University of Tennessee, Knoxville

TRACE: Tennessee Research and Creative Exchange

Doctoral Dissertations

Graduate School

12-2007

Theoretical Modeling and Experimental Validation of *In Vivo* Mechanics for Subjects Having Variable Cervical Spine Conditions

Fei Liu
University of Tennessee, Knoxville

Follow this and additional works at: https://trace.tennessee.edu/utk_graddiss

Part of the Biomedical Engineering and Bioengineering Commons

Recommended Citation

Liu, Fei, "Theoretical Modeling and Experimental Validation of *In Vivo* Mechanics for Subjects Having Variable Cervical Spine Conditions. " PhD diss., University of Tennessee, 2007.
https://trace.tennessee.edu/utk_graddiss/4188

This Dissertation is brought to you for free and open access by the Graduate School at TRACE: Tennessee Research and Creative Exchange. It has been accepted for inclusion in Doctoral Dissertations by an authorized administrator of TRACE: Tennessee Research and Creative Exchange. For more information, please contact trace@utk.edu.

To the Graduate Council:

I am submitting herewith a dissertation written by Fei Liu entitled "Theoretical Modeling and Experimental Validation of *In Vivo* Mechanics for Subjects Having Variable Cervical Spine Conditions." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Biomedical Engineering.

Richard D. Komistek, Major Professor

We have read this dissertation and recommend its acceptance:

Mohamed R. Mahfouz, William R. Hamel, Jack F. Wasserman, Ramon V. Leon, Joseph S. Cheng

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a dissertation written by Fei Liu entitled "Theoretical Modeling and Experimental Validation of *In Vivo* Mechanics for Subjects Having Variable Cervical Spine Conditions." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy with a major in Biomedical Engineering.

Richard D. Komistek
Major Professor

We have read this dissertation
and recommend its acceptance:

Mohamed R. Mahfouz

William R. Hamel

Jack F. Wasserman

Ramon V. Leon

Joseph S. Cheng

Accepted for the Council:

Carolyn R. Hodges
Vice Provost and Dean of the
Graduate School

(Original signatures are on file with official student records.)

Theoretical Modeling and Experimental Validation of *In Vivo* Mechanics for Subjects Having Variable Cervical Spine Conditions

A Dissertation
Presented for the Doctor of Philosophy Degree
The University of Tennessee, Knoxville

Fei Liu
December 2007

Copyright © 2007, by Fei Liu
All rights reserved

To my parents:

XiangLin Liu

JianXin Huang

Acknowledgements

Firstly, I would like to thank everyone involved for assisting and providing valuable support to me throughout this whole study. Particularly, though, I want to express my appreciation to Dr. Joseph S. Cheng and his research team at Vanderbilt University Medical Center for helping me complete IRB documents, cadaveric, and human experiments. I also wish to sincerely thank Dr. Jorge Ochoa and Archus Orthopedics, Inc. for financially supporting this project and helping me find the right surgeon at the first stage of this project. Additionally, I want to address Dr. Mohamed Mahfouz's kindness in his allowance for me to use SAAM. Next I must comment on how it was a great honor to work with Dr. Thomas Kane at the initial stage of this project. He gave invaluable suggestions for modeling techniques. This study would not have been possible without the help from all the staff and students in CMR lab: special thanks to Chris Little, Adrija Sharma, John Muller, Christopher Carr, and Mathew Anderle. I also would like to note my appreciation for the wonderful advices from Dr. William R. Hamel, Dr. Jack F. Wasserman, and Dr. Ramon V. Leon. Most importantly, this dissertation would never have been finished without all the supervision provided by my major professor Dr. Richard D. Komistek. Throughout the course of my Ph.D study, I was always thoroughly impressed by not only his experiences in the state-of-the-art biomechanical research area, but also his respectable personality. All that I have learned from Dr. Komistek will surely help me in both my future career and the rest of my life. Finally, I owe a special thanks to my parents who have always been here to support me.

Abstract

The objective of this study was to use the state-of-art 3D-to-2D registration technologies including fluoroscopic, CT and MRI methods to analyze 2D and 3D *in vivo* kinematics of the whole cervical spine under variable conditions; and use inverse dynamic model based on Kane's dynamics to predict their 2D and 3D *in vivo* interactive contact and muscular forces. Totally, forty patients (ten having normal cervical spines, ten having degenerative cervical spines, ten having anterior cervical decompression and fusion (ACDF), and ten having cervical artificial disc replacement (CADR)) were enrolled into 2D study and three patients (one having normal cervical spines, one having degenerative cervical spines, one having ACDF) were involved into 3D study. All of the patients had their symptoms, if any, at the C5-C6 level. Error analysis was performed on an entire cadaveric cervical spine. Two major mathematical models were derived using the principles governing Kane's dynamics. At the adjacent levels, both 2D and 3D study showed the ACDF group had relatively larger kinematic and kinetic data compared to the normal group, the degenerative group had relatively smaller kinematic and kinetic data. At the same time, 2D study demonstrated that the CADR group had similar kinematic and kinetic data compared to the normal group. Cadaveric error analysis demonstrated that the 3D-to-2D registration method and the inverse dynamic method had high accuracy and can be used in the cervical spine field.

Table of Contents

Chapter 1 Introduction 1
1.1. Biomechanical Anatomy and Functions of the Cervical Spine 1
1.1.1. Vertebra 2
1.1.2. Intervertebral Disc 5
1.1.3. Ligament 6
1.1.4. Muscle 7
1.2. Clinical Relevance 9
1.3. Medical Imaging 11
1.4. Kane's Dynamics 11
1.5. Research Objectives and Strategy 13
Chapter 2 Literature Reviews 16
2.1 Kinematic Analysis of the Cervical Spine 16
2.2 Kinetic Analysis of the Cervical Spine 17
2.3 Kinematics and Kinetics Comparison between Different Cervical Spines 19
Chapter 3 Materials and Methods 22
3.1 2D Dynamics Analysis Method 23
3.1.1 2D Kinematics Analysis Method 23
3.1.2 2D Kinetic Analysis Method 25
3.1.3 Statistical Analysis Method 30
3.2 3D Kinematic Analysis Method 30
3.2.1 Subject Demographics 30
3.2.2 Risks Specification and Protection 34
3.2.3 Human Experiment Procedure 35
3.2.4 3D Kinematic Analysis Method 41
3.3 3D Kinetic Analysis Method 45
3.3.1 Mathematical Modeling Description 45
3.3.2 Contact Point and Contact Force 50
3.3.3 Ligament Force 52
3.3.4 Muscular Forces 59
3.3.5 Description of 3D Mathematical Model 63
3.4 Cadaveric Error Analysis Experiment 71
3.4.1 Cadaveric Experimental Materials and Equipments 74
3.4.2 Experiment Procedures 79
3.4.3 Kinematic analysis method for cadaver experiment 80
3.4.4 Kinetic analysis method for cadaver experiment 89
Chapter 4 Results 90
4.1. 2D Kinematics 90
4.2. 2D Kinetics 92
4.3. 3D kinematics 96
4.3.1 Flexion-extension 99
4.3.2 Axial Rotation 110
4.3.3 Lateral Bending 114
4.4. 3D Kinetics 121
4.5. Error Analysis Experiment 133
4.4.1 Sensor Calibration 133

4.4.2 Kinematic Error Analysis 135
4.4.3 Kinetic Error Analysis 140
Chapter 5 Discussion 141
5.1 Kinematics Analysis 141
5.1.1 2D Kinematics 142
5.1.2 3D Kinematics 146
5.2 Kinetics Analysis 155
5.3 Error Analysis Experiment 160
Chapter 6 Limitations and Future Work 165
6.1 Limitations 165
6.2 Future work 167
Appendix 175
Matlab codes of 3D mathematical models 176
Publication 275
Abstracts and Presentations 275
Peer-reviewed journal articles 275
Vita 277

List of Tables

Table 3.1 Demographics of selected subjects. 33
Table 3.2 Description of Rigid Bodies in the 3D Mathematical Model. 48
Table 3.3 Description of the ligament attachment points on the vertebral bodies from C3 to C7 in the 3D Mathematical Model. 55
Table 3.4 Muscles included in the 3D Mathematical model. 59
Table 3.5 Equipments and Software used in the cadaveric experiment. 72
Table 4.1 Kinetic data for normal, degenerative, ACDF, and CADR groups. 94
Table 4.2 Average translation of the normal, degenerative and ACDF subjects during flexion-extension activity (Unit: mm). 107
Table 4.3 The overall range of axial rotation for three groups during axial rotation for one side with comparison to previous literature (Unit: Degrees) 112
Table 4.4 The overall range of coupled rotation of three groups during axial rotation for one side with comparison to previous literature (Unit: Degrees). 113
Table 4.5 Average intersegmental translations of normal, degenerative, and ACDF subjects during axial rotation (Unit: mm). 114
Table 4.6 The overall range of LB for three groups during lateral bending for one side with comparison to previous literature (Unit: Degrees) 120
Table 4.7 The overall range of coupled rotations between C4 and C6 for different groups (Unit: Degrees). 120
Table 4.8 Intersegmental translations of vertebrae for normal, degenerative and ACDF subjects during the lateral bending activity (Unit: mm). 121
Table 4.9 Contact forces predicted by the 3D mathematical model including 5 different types of ligaments but not muscular forces in the normal, degenerative, and ACDF cervical spines during the flexion-extension activity. 122
Table 4.10 The average errors for the drift testing. 134
Table 4.11 The results of the repeatability test. 134
Table 4.12 Error analysis results of the kinematic comparison in the cadaveric cervical spine experiment. 139
Table 5.1 Comparison of intersegmental rotation in the present study and Picket's. 144
Table 5.2 Mean and SD for flexion-extension of the cadaveric knee experiment by Drs. Mahfouz and Komistek[84]. 160
Table 5.3 Root Mean Square (RMS) of translations and rotation for flexion-extension in human error analysis [81]. 161
Table 5.4 Accuracy of the *in vivo* experiments in literature [93, 94]. 163
Table 5.5 3D in vivo error analysis results from Ishii's studies [20, 22]. 164

List of Figures

Figure 1.1 Anatomy of C1 and C2. 3
Figure 1.2 Anatomy of the typical cervical spine. 4
Figure 1.3 Load-Displacement curve of spinal ligaments [2] 7
Figure 1.4 Muscles in cervical spine (left: lateral view; right: anterior view). 8
Figure 1.5 Overlay 3D CAD models into 2D fluoroscopic images of different joints. 12
Figure 1.6 Flow chart of the research plan. 15
Figure 3.1 Flexion-extension fluoroscopic images of a disc replacement cervical spine in the sagittal plane. 24
Figure 3.2 Digitization method used to analyze the cervical spine. 24
Figure 3.3 Free body diagram of 2D mathematical model of the cervical spine with pseudo-muscular forces. 27
Figure 3.4 Switching Logic used in the mathematical model. 29
Figure 3.5 CT experiment for a subject. 37
Figure 3.6 MRI experiment for a subject. 37
Figure 3.7 Fluoroscope experiments for one subject (Top: flexion-extension; middle: lateral bending; bottom: axial rotation). 38
Figure 3.8 A CT segmentation image of the cervical spine in the axial plane. 43
Figure 3.9 A CT segmentation image of the cervical spine in the coronal and sagittal planes. 43
Figure 3.10 3D CAD models of a normal cervical spine from CT images after segmentation. 44
Figure 3.11 3D CAD models of a cervical spine with ACDF from CT images after segmentation. 44
Figure 3.12 The origin for the coordinate system of a vertebra (C2) was chosen automatically by using the bounding box method. 46
Figure 3.13 The origin for the coordinate system of a vertebra (C7) was chosen automatically by using the bounding box method. 47
Figure 3.14 Free body diagram of the 3D mathematical model of the cervical spine only including the contact forces. 49
Figure 3.15 Free body diagram of the 3D mathematical model of the cervical spine about ligament forces. 51
Figure 3.16 Modeling a contact point at the superior part of the vertebral body during the flexion-extension activity. 53
Figure 3.17 Modeling a contact point at the inferior part of the vertebral body during the flexion-extension activity. 53
Figure 3.18 Modeling of ligament forces for two adjacent levels (Sagittal View). 57
Figure 3.19 Modeling of ligament forces for two adjacent levels (Back View). 58
Figure 3.20 Modeling of muscular forces included in the 3D mathematical model for the activity from full extension position to neutral position. 60
Figure 3.21 Modeling of muscular forces included in the 3D mathematical model for the activity from the neutral position to the full flexion position. 61
Figure 3.22 A MRI images of the normal subject. 62
Figure 3.23 A MRI images of the degenerative subject. 62
Figure 3.24 A MRI image of the ACDF subject. 63
Figure 3.25 Euler translation diagram. 64

Figure 3.26 Experimental setup for the cadaver experiment. 73
Figure 3.27 Cervical spine specimen tested in the error analysis experiment. 74
Figure 3.28 Spine Simulator Machine. 75
Figure 3.29 Markers used for MicronTracker system in cadaveric experiment. 77
Figure 3.30 CT scan for the cadaveric cervical spine. 77
Figure 3.31 Flow chart of the experimental procedures in the cadaver experiment...79
Figure 3.32 Camera coordinate system (small one, right hand coordinate) and coordinate system in SAAM (large one, left hand coordinate). 82
Figure 3.33 An example of the original data directly from MicronTracker system. ..83
Figure 3.34 An example of 3D rotations (unit: degrees) and translations (unit: mm) calculated from MicronTracker system. 86
Figure 3.35 An example of 3D rotations (unit: degrees) and translations (unit: mm) captured from TableCurve 2D. 87
Figure 3.36 An example of data comparison for rotations and translations between SAAM and Micron Marker System. 88
Figure 3.37 Sensor outputs in the cadaveric cervical spine experiment. 89
Figure 4.1 Average intersegmental ROM in four different cervical spines. 91
Figure 4.2 Comparison of the average intersegmental rotations from 20º flexion to 15 º extension at the superior adjacent level C4-C5 of the different cervical spines. 91
Figure 4.3 Comparison of the average intersegmental rotations from 20º flexion to 15 º extension at the inferior adjacent level C6-C7 of the different cervical spines. 92
Figure 4.4 Comparison of the average intersegmental rotations from 20º flexion to 15 º extension at the level C3-C4 of the different cervical spines. 93
Figure 4.5 Anteroposterior force comparison between the normal and ACDF groups during the flexion-extension activity (upper: Normal; bottom: ACDF). 95
Figure 4.6 Compression force comparison between normal and ACDF during flexion-extension (upper: Normal; bottom: ACDF). 97
Figure 4.7 Pseudo-muscular force comparison between normal and ACDF during flexion-extension (upper: Normal; bottom: ACDF). 98
Figure 4.8 Flexion-extension activity of the entire cervical spine for the normal subject after registering 3D models onto 2D fluoroscopic images (from top-left to right-bottom: Full Extension, Position 2, Position 3 and Full Flexion). 100
Figure 4.9 Flexion-extension activity of the entire cervical spine for the degenerative subject after registering 3D models onto 2D fluoroscopic images (from top-left to right-bottom: Full Extension, Position 2, Position 3 and Full Flexion). 101
Figure 4.10 Flexion-extension activity of the entire cervical spine for the ACDF subject after registering 3D models onto 2D fluoroscopic images (from top-left to right-bottom: Full Extension, Position 2, Position 3 and Full Flexion). 102
Figure 4.11 Intersegmental ROM between vertebrae during the flexion-extension activity for the normal, degenerative, and ACDF subjects and compared to previous literature (Unit: degrees). 103
Figure 4.12 Flexion-extension activity of the entire cervical spine for the normal, degenerative and ACDF subjects after registering 3D models onto 2D fluoroscopic images (Full Extension-Original Position, Position 2, Position 3 and Full Flexion) (Unit: degrees). 105
Figure 4.13 Flexion-extension activity of the entire cervical spine for the normal, degenerative and ACDF subjects after registering 3D models onto 2D fluoroscopic images (Full Extension-Original Position and Position 2) (Unit: degrees). 105

Figure 4.14 Flexion-extension activity of the entire cervical spine for the normal, degenerative and ACDF subjects after registering 3D models onto 2D fluoroscopic images (Full Extension-Original Position and Position 3) (Unit: degrees). 106
Figure 4.15 Flexion-extension activity of the entire cervical spine for the normal, degenerative and ACDF subjects after registering 3D models onto 2D fluoroscopic images (Full Extension-Original Position and Full Flexion) (Unit: degrees).....106
Figure 4.16 Intersegmental translations for the normal, degenerative, and ACDF subjects during the flexion-extension activity. 108
Figure 4.17 Intersegmental translation between vertebrae for the normal subject (Full Extension-Original Position, Position 2, Position 3 and Full Flexion).....108
Figure 4.18 Intersegmental translation between vertebrae for the ACDF subject (Full Extension-Original Position, Position 2, Position 3 and Full Flexion).....109
Figure 4.19 Intersegmental translation between vertebrae for degenerative subjects (Full Extension-Original Position, Position 2, Position 3 and Full Flexion).....109
Figure 4.20 Axial rotation activity for the normal (top) and ACDF (bottom) subjects after registering 3D models onto a 2D fluoroscopic image. 111
Figure 4.21 Intersegmental translations of three subjects in AP, SI, and LT directions during axial rotation (Unit: mm).....115
Figure 4.22 Lateral bending activity for the normal subject after registering 3D models onto 2D fluoroscopic images (from top-left to right-bottom: Full Left, Position 2, Position 3 and Full Right).....117
Figure 4.23 Lateral bending activity for the ACDF subject after registering 3D models onto 2D fluoroscopic images (from top-left to right-bottom: Full Left, Position 2, Position 3 and Full Right).....118
Figure 4.24 Lateral bending activity for the degenerative subject after registering 3D models onto 2D fluoroscopic images (from top-left to right-bottom: Full Left, Position 2, Position 3 and Full Right).....119
Figure 4.25 Compression forces in the normal cervical spine.....124
Figure 4.26 Compression forces in the degenerative cervical spine.....124
Figure 4.27 Compression forces in the ACDF cervical spine. 125
Figure 4.28 Anteroposterior forces in the normal cervical spine.....125
Figure 4.29 Anteroposterior forces in the degenerative cervical spine. 126
Figure 4.30 Anteroposterior forces in the ACDF cervical spine. 126
Figure 4.31 Lateral forces in the normal cervical spine.....127
Figure 4.32 Lateral forces in the degenerative cervical spine. 128
Figure 4.33 Lateral forces in the ACDF cervical spine. 128
Figure 4.34 Compression forces in the normal (above), degenerative (middle), and ACDF (bottom) cervical spine after adding flexors and extensors to the mathematical models.....129
Figure 4.35 Anteroposterior forces in the normal (above), degenerative (middle), and ACDF (bottom) cervical spine after adding flexors and extensors to the mathematical models.....130
Figure 4.36 Lateral forces in the normal (above), degenerative (middle), and ACDF (bottom) cervical spine after adding flexors and extensors to the mathematical models. 131
Figure 4.37 Muscular forces in the normal (above), degenerative (middle), and ACDF (bottom) cervical spine after adding flexors and extensors to the mathematical models.....132

Figure 4.38 The original data of a drifting test by applying a 1000g load on a *FlexiForce*® Load/Force sensors for 2 hours. 134
Figure 4.39 The calibration of a *FlexiForce*® Load/Force sensor. 136
Figure 4.40 Flexion-extension activity of the cadaveric cervical spine after registering 3D models onto a 2D fluoroscopic image. 137
Figure 4.41 Axial rotation activity of the cadaveric cervical spine after registering 3D models onto a 2D fluoroscopic image. 138
Figure 4.42 Lateral bending activity of the cadaveric cervical spine after registering 3D models onto a 2D fluoroscopic image. 138
Figure 4.43 Kinetic comparison between predicted vertical forces and the reading from the *FlexiForce*® sensor. 140
Figure 5.1 The origin for a coordinate system was chosen at the posterior-inferior corner of a vertebral body in many previous studies. 147
Figure 5.2 Different setups of origins of a coordinate system at two vertebrae. 149
Figure 5.3 Shape comparison between artificial knee implant and vertebrae of the cervical spine. 162

Chapter 1 Introduction

The cervical spine serves an especially important purpose for the human body. It functions as a motion-controlling structure in the human body which provides support and stability to the head while transferring the weight and motion of the head to the trunk. It reduces internal and external impacts to the head and protects the spinal cord during daily activities such as walking and in extreme situations such as a car crash. The cervical spine is one of the most flexible hard tissues in the human body. Combined with a multitude of ligaments and muscles, it allows for accurate, swift, and complex physiological head motions in order to obtain a wide view of the environment.

Previously, few research studies have addressed *in vivo* three dimensional (3D) dynamics (kinematics and kinetics) of the whole cervical spine and compared the data between normal, pathologic, and postoperative subjects. The objective of this study was to determine the 2D and 3D *in vivo* dynamics for normal, pathological, and postoperative cervical spines and to see if surgical operations could cause significant motion and force variations in the postoperative cervical spines. This research study could provide important information for surgeons to design better surgical procedures and for biomedical engineers to design better prostheses, related to the cervical spine.

1.1. Biomechanical Anatomy and Functions of the Cervical Spine

The cervical spine is one of the most intricate joints in the human body. This research project analyzed the mechanics of the vertebrae and the associated surrounding ligaments and muscles. A better understanding of biomechanical

anatomy and function of the cervical spine is crucial in order to mathematically model these structures.

1.1.1. Vertebra

The cervical spine consists of seven vertebrae, commonly referred to as C1 through C7. Normally, C1 and C2 are also specifically called atlas and axis, respectively (Figure 1.1). They have different morphological structures compared to the other 5 vertebrae. Atlas looks like a ring of bone without a vertebral body and posterior processes. There is no intervertebral disc between atlas and axis. Atlas has an ellipse hollow structure called fovea dentis while axis has a tooth-like process called the den which projects vertically and upward. Both the fovea dentis and dens are located at the middle anterior position. The fovea dentis and dens together constitute a part of atlantoaxial joint which functions like a sleeve. Two articular facets are located at the left and right anterior positions between atlas and axis. However, they are located on the left and right posterior parts on the other levels of the cervical spine. According to the previous study, C1 and C2 contribute up to 50% of the total cervical spine motions because of their highly specialized structures [1].

C3 through C7 are the typical cervical vertebrae. In the transverse plane, they can be divided into anterior components, which include vertebral bodies and transverse processes, and posterior components, which include spinous processes and articular processes (Figure 1.2). On the vertebral bodies from C3 to C7, the vertebrae have the uncinate processes, which is different from the lumbar spine. The anterior and posterior edges on the inferior surface of each vertebral body hang downward to the corresponding edges, which have a slope in the same direction, on the superior

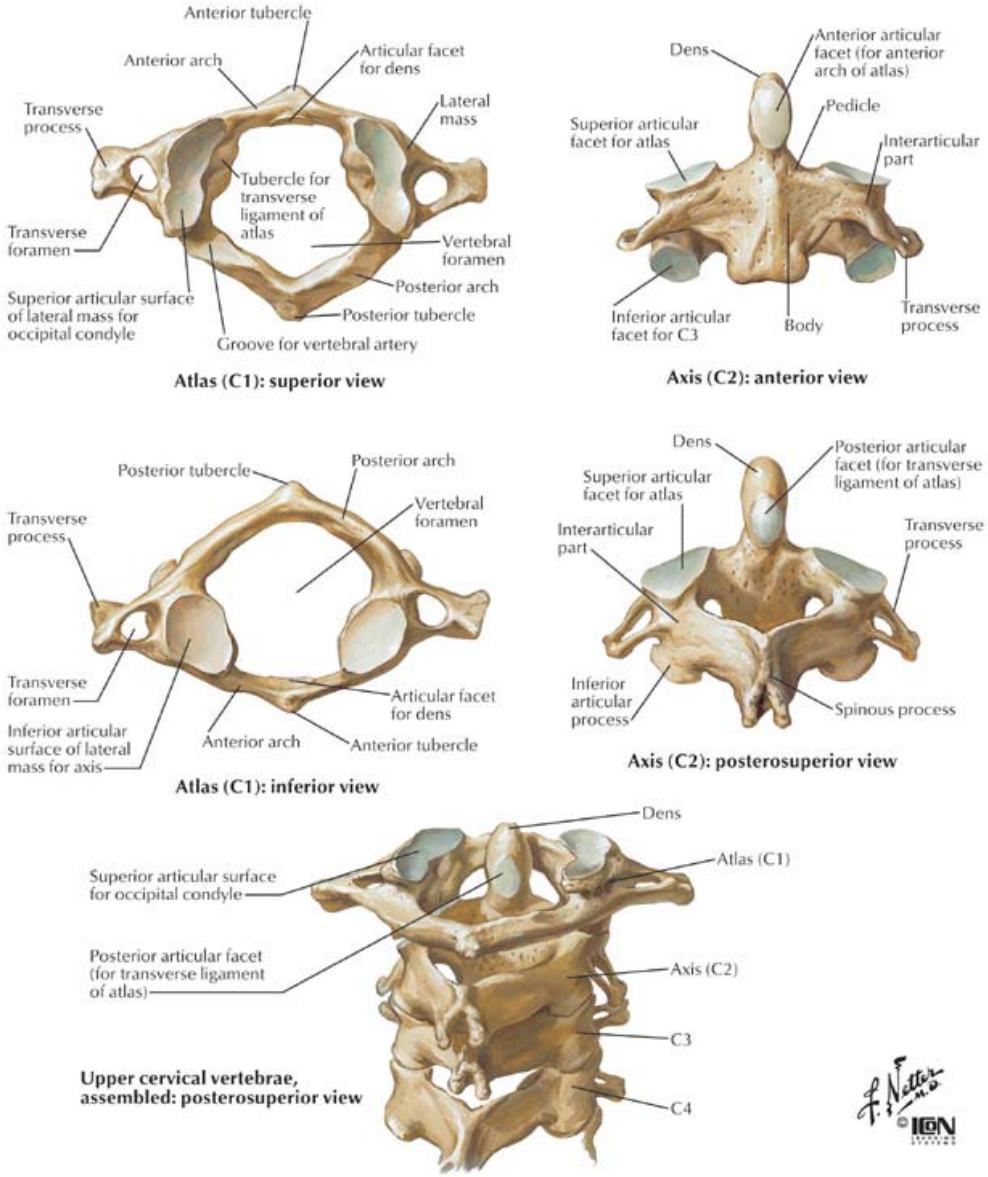


Figure 1.1 Anatomy of C1 and C2.
(http://www.backpain-guide.com/Chapter_Fig_folders/Ch05_Anatomy_Folder/3C1C2.html).

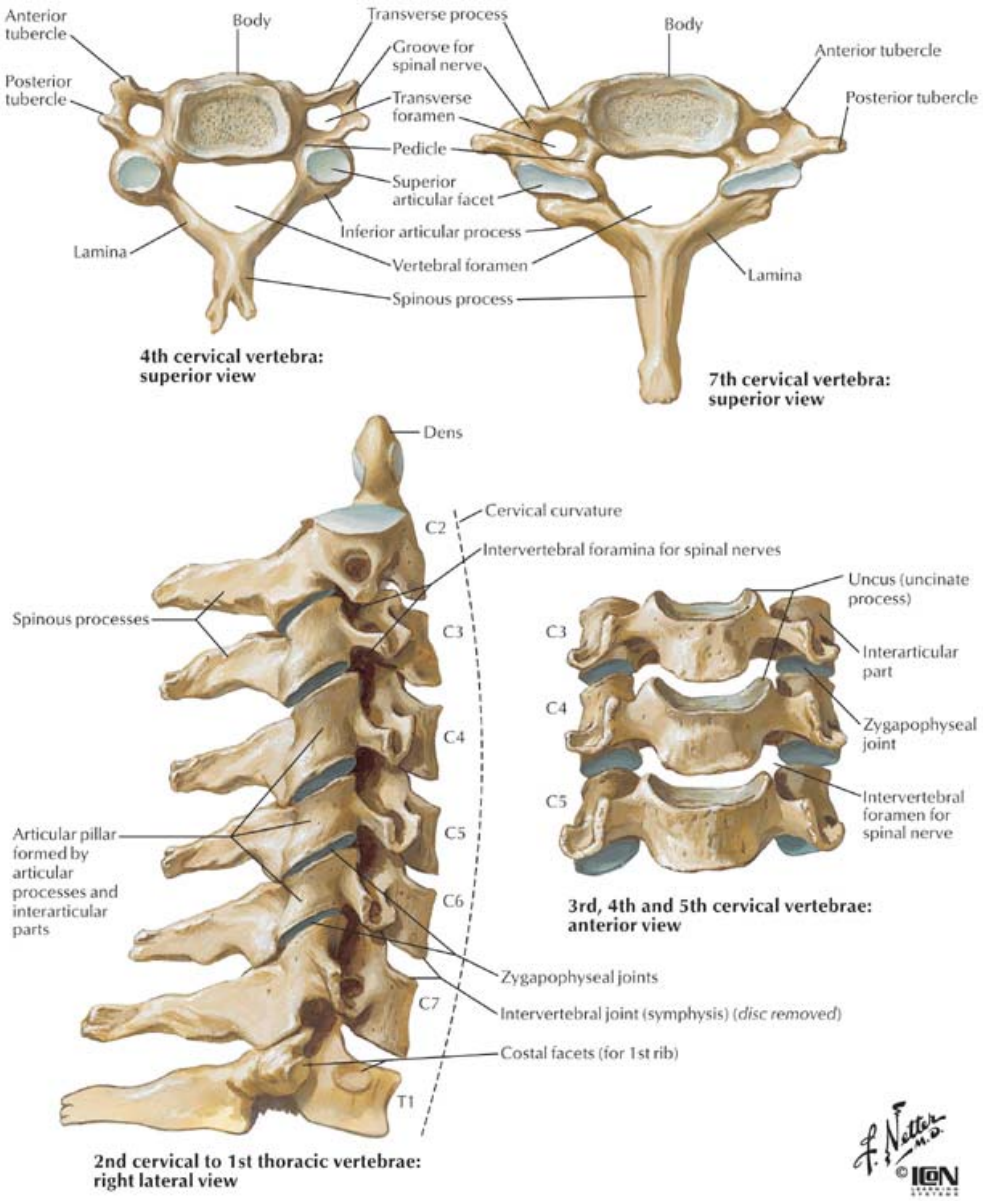


Figure 1.2 Anatomy of the typical cervical spine. (http://www.backpainguide.com/Chapter_Fig_folders/Ch05_Anatomy_Folder/2CervVert.html).

surface at the adjacent level. Contrarily, the left and right borders of the inferior surface slope upward while the corresponding borders of the superior surface hang upward in the same manner. These form two saddle structures, which guard smooth motions over all translational directions. At the same time, these structures are more likely to create coupling motions during axial rotation and lateral bending activities. Both vertebral bodies and facets share contact forces along the whole cervical spine, either through the intervertebral discs or directly. Vertebral bodies share most of the interactive contact forces. Compared to vertebral bodies, the major function of facets is to guide or constrain motions. In the cervical spine, facets are oriented at about 45º with respect to the horizontal plane, which allows them to slide between the superior and inferior parts of adjacent levels. All of these particular structures allow the cervical spine to be more flexible as compared to the lumbar and thoracic spines. Spinous and transverse processes serve as attachments for all of the major muscles surround the cervical spine.

1.1.2. Intervertebral Disc

The intervertebral disc consists of an annulus fibrosus and a nucleus pulposus. They are located between vertebral bodies from C2 to C7. An annulus is a strong radial tire–like structure made up of lamellae; a nucleus is mainly a gelatinous solid. When a load is applied to a nucleus, these physical properties transfer the pressure into an omni-directional force. The unique biological structure of intervertebral discs works to reduce impact between the trunk and skull, which profoundly enhances mechanical properties of the cervical spine. At a flexion position, there is a compression force at the anterior part and a tensile force at the posterior part. On the

other hand, at an extension position, a tensile force acts at the anterior part and a compression force acts at the posterior part.

1.1.3. Ligament

Ligaments are fibrous tissue structures connecting two or more bone and cartilages. They are pliant and flexible structures with viscoelastic properties. Some ligaments, such as the ligamentum flavum, can act as a curtain to fill the gap between adjacent vertebrae in order to protect the spinal cord. The vertebrae and intervertebral discs cannot maintain their stability and realize their various physiologic motions without ligaments acting as internal equilibratory components and muscles acting as external equilibratory components collectively. There are seven different types of ligaments from C3 to C7. They are the anterior longitudinal ligament, the posterior longitudinal ligament, the capsular ligament, the ligamentum flavum, the intertransverse ligament, the interspinous ligament, and the supraspinous ligament. Some ligaments between C1 and C2 are different from other portions of the cervical spine. Besides the anterior longitudinal ligaments, other ligaments include the anterior atlanto-dental ligament, the altantooccipital membrane, dentate ligaments, the cruciate ligament, the tectorial membrane, the posterior altantooccipital and atlanto-axial membranes, and the nuchal ligament.

Ligament length, cross-sectional dimension, coordinate of attachment, and material characteristics of ligaments are used as parametric input to mathematical models of the cervical spine in order to include these ligamentous constraint forces. Most parameters are presently quantified by using cadaveric experiments, and CT, MRI, and fluoroscopy radiological techniques. A load-displacement curve of a

ligament of the spine is plotted in Figure 1.3 [2]. This curve can be conventionally divided into three regions: the neutral zone (NZ), the elastic zone (EZ) and the plastic zone (PZ). The neutral zone (NZ) is the range of displacements near the neutral position due to a small force or torque. The elastic zone (EZ) is the range between the neutral zone and the plastic zone. The plastic zone (PZ) is the range from the elastic zone up, along the curve until failure. The neutral zone and the elastic zone together constitute the normal physiological range, while the plastic zone constitutes the range of trauma.

1.1.4. Muscle

Muscles surrounding the cervical spine can be categorized according to different standards (Figure 1.4). According to their position, they can be divided as anterior and posterior muscles. The major posterior muscles include trapezius, splenius capitis and cervicis, semispinalis capitis and cervicis, longissimus capitis

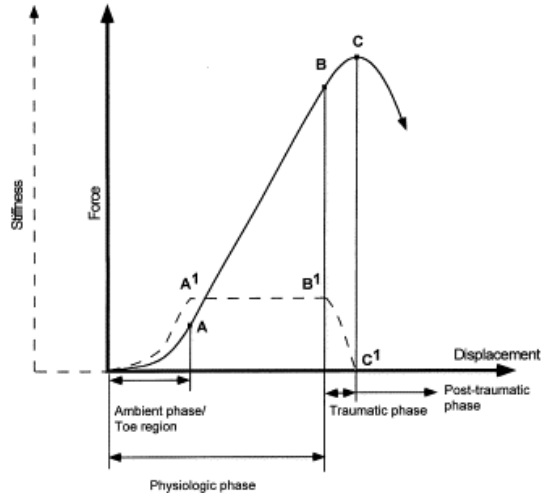


Figure 1.3 Load-Displacement curve of spinal ligaments [2].

and cervicis. The most prominent anterior muscle is the sternomastoid muscle. The deep anterior muscles are the longus colli, rectus capitis anterior and longus capitis. The intermediate anterior muscles, semispinalis cervicis, originate from the transverse processes of each vertebra and attach to the spinous processes of the correspective superior vertebra.

Muscles are attached to bones through tendons. They have the capability to actively create spinal motions by driving and controling motion of the vertebrae. At the same time, they have the same function as ligaments to maintain spinal stability under extension. Tendons have parallel-fibered collagenous tissue structures and a viscoelastic property which are not only strong enough to sustain tensile forces produced by contraction of muscles, but also flexible enough to angulate around the bone surface and to change directions of muscular forces. Thus, the direction of a muscular force cannot be determined by simply connecting a straight line between an origin and an insertion. On the other hand, even though muscles generally create forces by changing their length, tendons also could help muscles to tighten and generate forces without changing muscular lengths between origins and insertions.

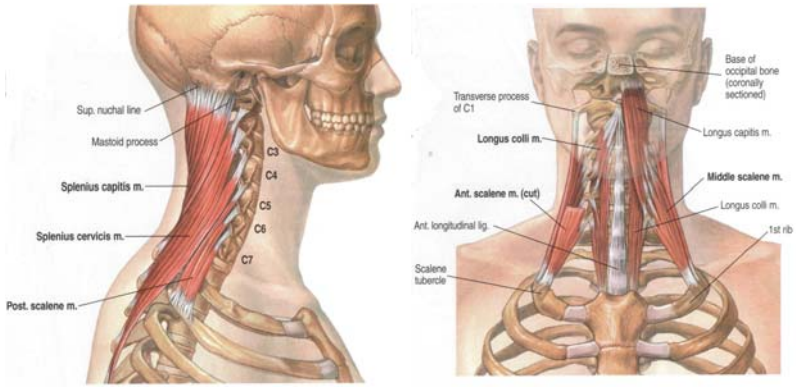


Figure 1.4 Muscles in cervical spine (left: lateral view; right: anterior view).

Thus, even though tendons have load-displacement curves as ligaments do, muscular forces should not be calculated by simply multiplying the distance by stiffness in some cases.

Under active situations, muscles have their maximum forces at 120%-130% of their original length and cannot provide forces around 50% of their original length [3]. In passive situations, muscles produce forces with respect to their extended length. According to Fung, muscles can be modeled as spring-dumping systems. There are three mechanical characteristics of muscles: force-velocity relationship, force-length relationship, and force-time relationship [3].

Muscles can sustain static stability with and without external load, when performing tasks. Nevertheless, when major muscles generate forces to perform movements, they could also cause unwanted movements of vertebrae. In order to balance these unwanted movements, some muscles and ligaments need to be embraced to resist them. Thus, muscles, which should not be activated, were indeed found to be active in some cases. Although electromyography studies cannot present a direct relationship with respect to muscular activities, they can provide valuable information pertaining to whether muscles are generating a force and the force ratios between tested muscles.

1.2. Clinical Relevance

Spending on health care in the U.S is steadily increasing. According to the World Health Report, the total health expenditure per capita was $5,711, which was about 15.2% of the Gross Domestic Product (GDP) per capita in 2003 (http://www.who.int/countries/usa/en/). This expenditure will grow 23% and

constitute 18.7% of GDP by 2010 according to the U.S Health Spending Projections for 2004–2014.

Neck pain is one of the most common epidemics in the U.S. Even though there is not an exact data of instances of neck pain in the U.S, chronic and persistent neck pain was reported in 13% of the Norwegian population[4], and the rate was as high as 26% in Sweden [5]. Maintaining and rebuilding the function of a pathologic cervical spine has a profound significance for patients. In the past decades, spinal fusion has been one of the most popular surgical spine operations [6, 7]. It has been demonstrated to be a successful surgical method to release spinal pain and to help restore subjects to normal lifestyles with an overall success rate around 70% [8]. According to previous reports, the rate of cervical spinal surgeries has risen rapidly and steadily in the last 25 years in the U.S [9-13]. From 1988 to 1990, the total number of cervical fusions was 26,000, which represents approximately a 120% increase compared to the period from 1979 to 1981 [10, 14, 15]. In 1999, the number of spinal fusions was about 350,000, among which there were 124,000 cases of the cervical fusion, which was increased 400% compared to the number reported 20 years ago. The size of the global spine arthroplasty market was approximately US$1.4 billion in 2004 and is projected to grow up to more than US$ 3 billion by 2010 [16].

Even though many new technical revolutions have been introduced into fusion surgeries, the average clinical success rate has not significantly increased during the last three decades [17]. It has been reported that about 2.9% of the cervical spines treated with anterior cervical decompression and fusion (ACDF) need a reoperation at their adjacent levels each year [18]. Disc replacement treatment, which contributes to motion preservation, was recently introduced into spinal surgery as a new alternative

to spinal fusion. Because of its advantages, some estimated that 70% of spinal surgeries would utilize some kind of disc replacement in the U.S by 2010[18].

1.3. Medical Imaging

The cervical spine has intricate anatomical structures and complex kinematic motions during normal physiological actions. Nowadays, radiological technologies enable a better understanding of *in vivo* pathological diagnosis and kinematic characteristics of the cervical spine. Both CT and MRI can help us to obtain *in vivo* cross-sectional images in the desired plane. In addition, MRI is a powerful tool in evaluating soft tissue contents. The images from MRI can be used to create 3D Computer Aided Design (CAD) models of soft tissues and to accurately determine muscular and ligament attachments. CT scanned images in the axial planes can be used to rebuild 3D CAD models of vertebrae. A disadvantage of MRI and CT is that they cannot determinate 3D motion characteristics under dynamic situations. Fluoroscopy is a relatively new medical method to help doctors and biomechanists obtain two dimensional (2D) dynamic information about the skeletal system. Researchers in the Center for Musculoskeletal Research (CMR) lab at the University of Tennessee, Knoxville have used 3D CAD models of implants or normal bones, derived from CT or MRI scans, and 2D fluoroscopic images to obtain 3D kinematic data of lower and upper extremity joints (Figure 1.5).

1.4. Kane's Dynamics

Currently, multibody dynamics has been widely utilized in the biomechanical field for reasonably simplifying physical systems by replacing them with equivalent

Figure 1.5 Overlay 3D CAD models into 2D fluoroscopic images of different joints.

dynamic models, whose mechanical properties have been well studied. In most cases, the dynamic equations governing motions of physical systems are highly nonlinear and cannot be analytically solved in a closed form. They must be numerically solved based on computer-aided techniques. Based on classic dynamics, Kane combined the advantages of Newton-Euler, Langrage, and D'Alembert conceptions and vector kinematics. Using Kane's dynamics, dynamic systems in the biomechanical field can then be solved to determine either their motions (direct approach) or their forces (inverse approach). The theory associated with Kane's dynamics can be used to solve complicated dynamic systems by using this generalized approach. The standard equation of Kane's dynamics is:

$$F_r + F_r^* = 0$$

For derivation purposes, Kane introduced generalization of the angular velocity and velocity equations, which are then differentiated for partial angular velocities and partial velocities. This generalization continues through the derivation of the motion equations, solving for the generalized active and inertia forces, which are produced through multiplying active forces and inertia forces by the corresponding partial angular velocity and partial velocity equations. Partial angular

velocities describe rotational properties of a body related to an applied torque. Partial velocities describe a body's translational properties related to applied forces. Normally, an n degrees of freedom (DOF) system has n generalized speeds, which are sequentially named "the first generalized speed", "the second generalized speed" and so on, through to the final generalized speed.

Differentiated with respect to the generalized speeds, partial angular velocities are derived from its corresponding angular velocity equation; and partial velocities are derived from its corresponding velocity equation. According to Kane's dynamics, if the angular velocity of a body (A) with respect to the Newtonian reference frame (A) is ${}^{N\to A}\vec{w} = A_1\vec{u}_1 + A_2\vec{u}_2 + A_3\vec{u}_3$, where $\vec{u}_1, \vec{u}_2, \vec{u}_3$ are general angular speeds and A_1, A_2, A_3 could be either constant or variable coefficients, the partial angular velocities with respect to U1, U2 and U3 are A1, A2, and A3, respectively. Similarly, if the velocity of a point (p) on body A in the Newtonian reference frame (N) is ${}^{N\to P}\vec{v} = B_1\vec{u}_4 + B_2\vec{u}_5 + B_3\vec{u}_6$, where $\vec{u}_4, \vec{u}_5, \vec{u}_6$ are generalized speeds and B_1, B_2, B_3 are either constant or variable coefficients, then the partial velocities with respect to U4, U5 and U6 are B1, B2, and B3, respectively.

1.5. Research Objectives and Strategy

Even though a substantial amount of data has been previously obtained through *in vitro* or *in vivo* experiments [19-21], few studies accurately describe 3D *in vivo* kinematic and kinetic characteristics of the entire cervical spine [22].

The objectives of this study were to thoroughly analyze 2D and 3D *in vivo* kinematics and kinetics of the normal, pathological and postoperative cervical spines -

- normal, degenerative, ACDF and cervical artificial disc replacement (CADR) -- as well as to compare them to see if operations could cause significant motion and force variations. The flow chart of the entire research procedure is shown as Figure 1.6. In Chapter 2, relevant literature is reviewed to give the fundamental background of this research. In Chapter 3, the materials and methods, both experimental and modeling, utilized in this study are extensively described and explained. Chapter 4 introduces and then compares results of 2D and 3D kinematics and kinetics of the normal, pathological, and postoperative cervical spines. The results from this research study are discussed and then compared with previous research studies in Chapter 5. Finally, limitations of the experiments and further work are discussed in Chapter 6.

In all, the following contributions are anticipated to derive from this research:

1. 2D *in vivo* derivation and application of dynamics (kinematics and kinetics) as related to the normal, pathologic, and postoperative cervical spines;
2. 3D *in vivo* derivation and application of dynamics (kinematics and kinetics) as related to the normal, pathologic, and postoperative cervical spines;
3. Development of 2D and 3D mathematical models based upon Kane's dynamics of the whole cervical spine including most major muscles, ligaments, and other characteristics of morphological structures;
4. Patient specific 2D and 3D mathematical models offering more accurate and more reliable kinematic and kinetic results;
5. Determination of this state-of-art system accuracy pertaining to the entire study by performing an experimental cadaveric error analysis.

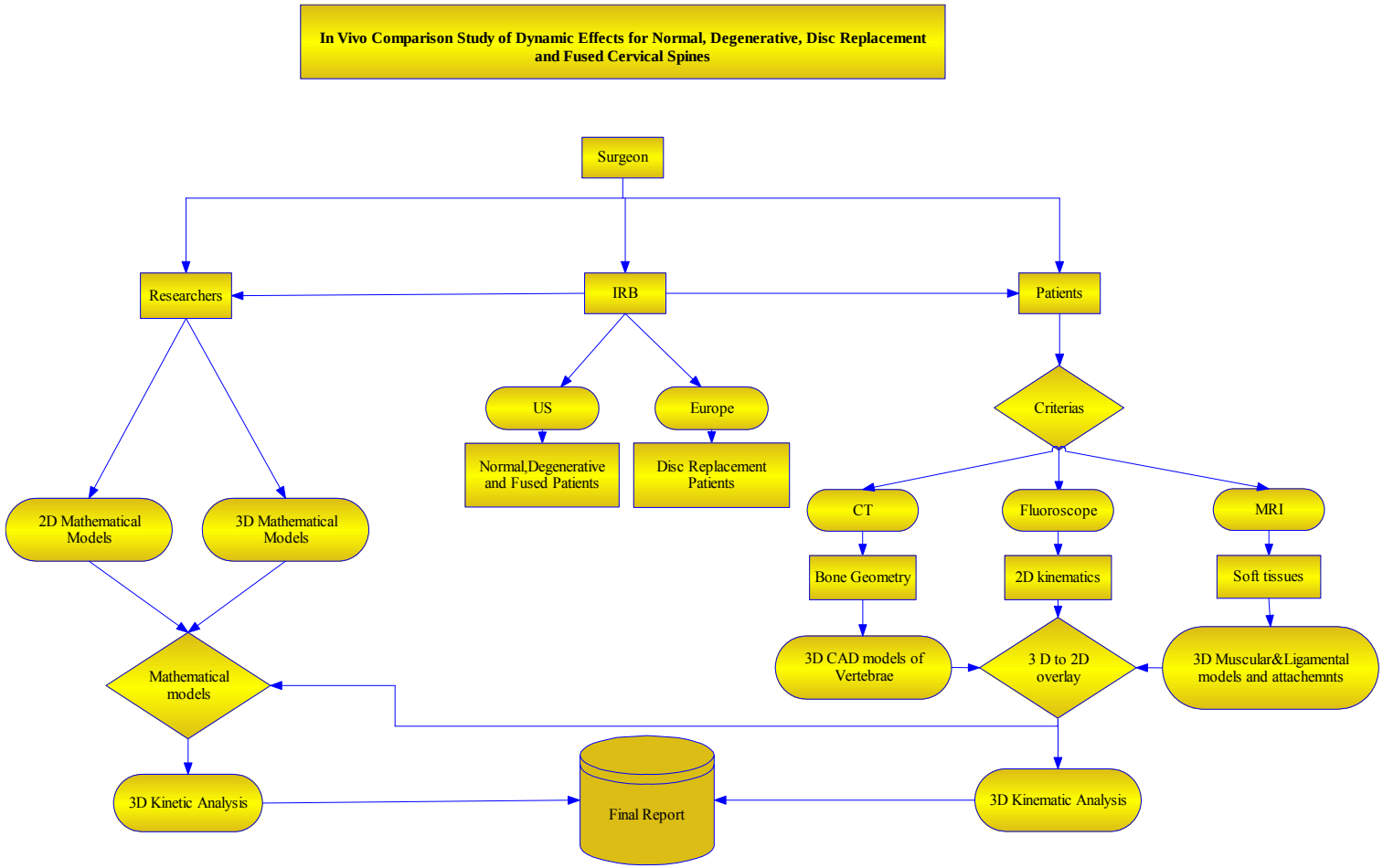


Figure 1.6 Flow chart of the research plan.

Chapter 2 Literature Reviews

In this chapter, previous studies about kinematics and kinetics of the cervical spine and the comparison between the normal, pathologic, and postoperative cervical spines have been presented and discussed in order to provide fundamental background in this field.

2.1 Kinematic Analysis of the Cervical Spine

The cervical spine exhibits three rotational motions: flexion-extension in the sagittal plane [23-28], lateral bending in the frontal plane [29], and axis rotation in the transverse plane [23, 29, 30]. Additionally, there are three translational movements in the 1, 2 and 3 directions based on Kane's dynamics (classically described as X, Y, and Z). Totally, there are 6 DOF for each vertebral body.

Kinematic studies of the cervical spine have been conducted as early as the middle 1800s [31]. Kinematic studies of the cervical spine can be divided into two major categories: *in vivo* and *in vitro*. In the *in vivo* studies, experiments were performed directly on live human subjects. The *in vivo* studies can be further divided into active studies [23, 25, 27, 29, 30] and passive studies [23, 24]. In the active studies, subjects perform movements up to the maximum physiological ranges without applying external loads. This type of experiment is used to study normal physiological motions. Alternatively, in the passive studies, external forces or torque are applied to push the examined cervical spines to their maximum positions, which helps analyzing injury

tolerance, cervical spinal trauma, and whiplash in automotive accidents. *In vitro* studies are conducted either on the cadaveric functional spinal unit (FSU) or the entire cervical spine specimen [1, 21, 32-34]. FSU includes two adjacent vertebrae, an intervertebral disc and/or major ligaments. It is easier to conduct biomechanical experiments by using FSU compared to using the whole cervical spine. In both *in vivo* and *in vitro* studies, medical images could be taken to document locations and orientations of the cervical spine. *In vitro* studies are easier to control and have a higher measurement of accuracy. However, *in vitro* studies are lack of basic muscular functions and physiological environments. So, *in vivo* studies can better describe the natural motions of the cervical spine because they encompass all of the soft tissues -- muscles and ligaments. Most recently, a series of MRI images were recorded when subjects placed their heads at different positions. 3D CAD models of vertebrae were rebuilt from these MRI images with respect to different head positions. Then, 3D *in vivo* kinematics of the whole cervical spine were quantified by using these 3D CAD models [20, 22, 35]. In these studies, the researchers only described the quasi-dynamic characteristics of the cervical spine, as opposed to the real dynamic situations.

2.2 Kinetic Analysis of the Cervical Spine

Many research studies have been performed to determine the kinetics of the cervical spine. Similar to kinematic analysis, kinetic analysis can be divided into *in vivo* and *in vitro* studies. *In vitro* kinetic experiments were performed by either using FSU or the whole cervical spine [21, 33, 36, 37]. The *in vitro* studies provided important and reliable references about different mechanical properties of the cervical spine. In the

biomechanical field, the main methods, which are available now to determine the kinetics of the spine under *in vivo* conditions are the utilization of telemetric implants and mathematical modeling approaches [38]. A telemetry implant, which is an experimental device to measure forces directly inside the human body, has been successfully used in hip, knee, and the lumbar spine [39-41]. However, it has not been utilized in the cervical spine field until now. In *in vivo* experiments, the number of subjects is very limited due to the expensive cost related to design, manufacture, and surgical procedures. On the other hand, a telemetric system is designed for a pathological joint. Therefore, it is basically impossible to experimentally obtain relevant information pertaining to kinetics of a normal joint. Also, there remains some technique problems associated with the telemetric implants, such as calibration, area measurements, and drift of signals, which can lead to unreliable results in many cases. Mathematical models can theoretically evaluate *in vivo* forces with reliable accuracy [38, 42]. This approach has been widely used in analyzing different joints in the human body. One of the challenges to use mathematical modeling is that only six unknowns can be solved for each rigid body in a 3D space domain according to dynamics principles. However, there are seven different types of ligaments, multiple muscles and contact forces acting on and/or across each vertebra. They represent more than six unknown forces on each vertebra. It is impossible to solve all these forces directly. The optimization technique [43, 44] and the reduction method [42] are two major mathematical methods used to solve an indeterminate system. The optimization technique is easy to induce results, which are mathematically correct, but may not be physiologically accurate. The reduction method simplifies these forces with well understanding anatomical and dynamic characteristics of

the human body in order to reduce the number of unknowns in the system.

2.3 Kinematics and Kinetics Comparison between Different Cervical Spines

A degenerative cervical spine refers to the cervical spine which breaks normal architectures of its components. Changes of normal architectures cause loss of range of motion (ROM) as well as induce neck pain. Except for aging, no other cause, such as wear and tear, have been proven to definitely and directly cause degeneration.

ACDF has been the golden standard in the treatment of cervical spondylosis, cervical herniation, and ossification, since Robinson and Smith first introduced it as a treatment for the cervical spine in the 1950s [45-49]. Numerous follow-up clinical studies have demonstrated its successful long-term outcome. It has been reported that the clinical success rate among subjects after this surgery was above 83% for three-level fusion and as high as 97% for single-level fusion after a one year follow-up [50]. At the same time, however, follow-up clinical studies have reported a high incidence of degeneration at the adjacent segments in a ACDF group, as compared with that in a normal group [18, 51, 52]. A previous study reported that each year after ACDF, about 2.9% of fusion cervical spines would need a new operation at adjacent levels, and that an estimated 25.6% of them would have a new disease in the first ten years [18]. Some studies have reported that ACDF operations significantly reduced the motion at pathological levels by about 81% for flexion, 61% for extension, 67% for right lateral bending, and 83% for left lateral bending, respectively [53]. Furthermore, *in vitro* biomechanics studies have shown that a loss of motion at the ACDF level created greater

segment motions, increased load or intradiscal pressure at its adjacent levels, which is believed to induce subsequent degeneration at those levels [54, 55]. A cadaver experiment has reported increased pressure of 73.2% and 45.3% at the adjacent levels after fusion, which was statistically higher than that found in the intact spines [55]. Other clinical and biomechanical studies addressed the opposite view that there was no obvious evidence of the development of adjacent degeneration after ACDF [56-58]. Even though the cause of adjacent degeneration were not completely understood, it is obvious that the cervical fusion does lead to the loss of motion at the ACDF levels and that it changes the normal biomechanical characteristics at its adjacent levels.

Artificial disc replacement (ADR) allows motion preservation and rebuilds normal motion patterns at the injured or degenerative joint discs. Despite that cervical artificial disc replacement (CADR) is still undergoing clinical trials in the U.S, more follow-up clinical and biomechanical studies have been undertaken to investigate its kinematic and kinetic outcomes [59-62]. Currently, an one year follow-up study has reported a satisfaction rate of over 85% after CADR [59]. Also, an *in vitro* biomechanical study was performed using six fresh cadaver cervical spines, and has indicated that the motion and pressure after CADR are equivalent to that of the intact cervical spines [60]. Another cadaveric study has addressed how the CADR and intact cadavers possess similar motion patterns and intradiscal pressure that are statistically different from those after ACDF [61]. An *in vivo* investigation of 2 years follow-up study has shown the potential of CADR to reproduce the preoperative motions of the injured disc for flexion-extension motion [62]. Because of its advantages, some estimate

that 70% of spinal surgeries will utilize some kind of disc replacement in the U.S by 2010 [16].

On the other hand, it could not be simply concluded that CADR is a better clinical alternative as compared with ACDF based on the above studies because respective radiographic and biomechanical results cannot be definitely associated with clinical symptoms. Nevertheless, if we could compare these *in vivo* data of normal, degenerative, ACDF, and CADR subjects collectively in a single study, we could better understand the clinical and biomechanical outcomes for each circumstance. Therefore, the objective of this study was to quantify the *in vivo* kinematic characteristics of these different cervical spines and to compare their *in vivo* kinetic data by deriving inverse dynamic models under dynamics situations, in order to determine which surgical technique has better postoperative results.

Chapter 3 Materials and Methods

During normal physiological activities, the cervical spine has more intricate anatomical structures and more complex motion patterns as compared to other joints such as the knee, hip, or shoulder. As seen in previous studies, it is difficult to accurately quantify 3D, *in vivo* dynamics (kinematics and kinetics) of the cervical spine, which is important to analyze postoperative results. Presently, a deeper understanding of 3D *in vivo* kinematic and kinetic characteristics of the cervical spines has become possible through the introduction of radiological technologies and cutting-edge computer modeling methods.

In total, three separate experiments were designed in this study: 1) a human experiment including 40 subjects (10 normal, 10 degenerative, 10 ACDF, and 10 CADR) was performed for 2D study; 2) a human experiment including 3 subjects (1 normal, 1 degenerative, and 1 ACDF) was done for 3D study; 3) a cadaveric cervical spine experiment analyzed the accuracy of the entire 3D study of the human experiment. Concurrently, two major mathematical models were derived based on Kane's dynamics. The details of the materials and experimental devices used for both human and cadaveric experiments were described in this chapter along with the experimental procedures utilized. Also, the 2D and 3D mathematical models were thoroughly discussed.

3.1 2D Dynamics Analysis Method

First, a fluoroscopic experiment involving forty subjects was performed in order to determine the 2D *in vivo* kinematic and kinetic characteristics for normal, degenerative, ACDF, and CADR spines. This human experiment and analysis led to a better understanding of the complex anatomic structures and dynamic characteristics of these four different cervical spines. And, this 2D study is the preliminary approach for further 3D analysis, allowing for baseline kinematic data to be derived and analyzed and for baseline kinetic data to be mathematically predicted.

3.1.1 2D Kinematics Analysis Method

Ten normal subjects, ten patients with degenerative symptoms, ten patients treated with ACDF, and ten patients having CADR performed a smooth full flexion to hyper-extension motion under fluoroscopic surveillance (IRB#0445) in the sagittal plane. The degenerative groups also had symptom at the C5-C6 level. Also, the ACDF and CADR surgeries were performed at the C5-C6 level. The minimum follow-up time was six months after an ACDF or CADR surgery. Experiments on the normal, degenerative and ACDF subjects were performed in the U.S. Experiments on the CADR patients were performed in Belgium. A high frequency pulsed fluoroscopy unit (Radiographic and Data Solutions, RADS, Minneapolis, MN), capable of capturing images at a rate of 30 frames per second, was used in this research. For each subject, 10 serial images were digitized from the fluoroscopic video (Figure 3.1). A local coordinate system was set up on each segment (Figure 3.2). The orientation angles of each vertebra throughout the

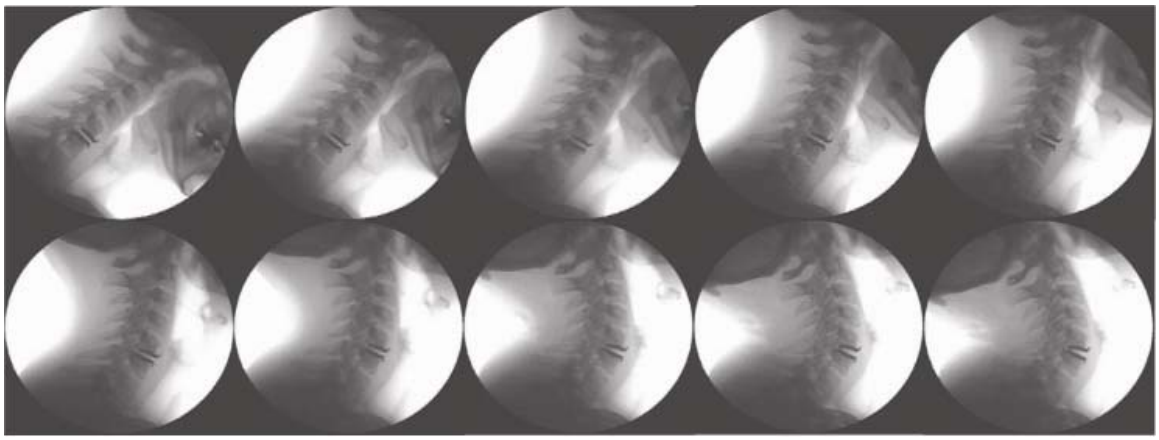


Figure 3.1 Flexion-extension fluoroscopic images of a disc replacement cervical spine in the sagittal plane.

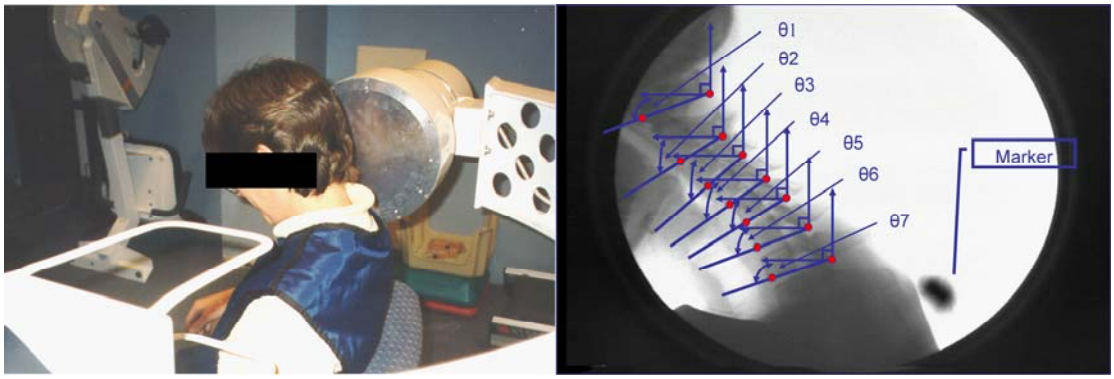


Figure 3.2 Digitization method used to analyze the cervical spine.

motion were obtained as well as coordinates of specific points. Both the inferior, which is inclined toward the concavity of the curvature, and the superior, which is inclined toward the convexity of the curvature edges of each segment, are presented as curves instead of straight lines. Because of these anatomic characteristics, the farthest points on the same edges were digitized. This method is more accurate and makes it easier to identify the edge of each vertebra in a sequence of images. Then, a straight line was drawn between these two points to measure the angle with respect to the horizontal axis. Cobb's angles were measured by subtracting the angle of each segment from the angle of its adjacent segment. At the same time, coordinates of important points, which includes contact points, the mass center of each vertebra, muscle attachments (transverse process), were digitized directly from fluoroscopic images according to the literature [63] by calculating the relative distances with respect to the origin of the inferior and left corner. Then, they were translated into actual distances by multiplying the rate between the size of standard marker and its pixels in the fluoroscopic images. Temporal equations of intersegmental motions were derived from the relative motions between adjacent vertebrae by curve fitting, and *in vivo* kinematic data were statistically analyzed and compared to describe the difference between the four different cervical spines. Then, the kinematical equations were input into the mathematical models to determine the 2D kinetic results.

3.1.2 2D Kinetic Analysis Method

A 2D inverse dynamic model of the entire cervical spine, including the pseudo-muscular forces, was derived based on Kane's dynamics and reduction modeling

technique. The equations of motion for this model were determined using the symbolic manipulation algorithm Autolev™ (Online Dynamics Inc, Sunnyvale, CA) and Matlab™ (The MathWorks, Inc. Natick, MA). The purpose of this research was to compare the force patterns at two adjacent segments (C6-C7 and C4-C5) and a segment located one level away from the symptom or postoperative segments (C3-C4). The anatomic, kinematic, and kinetic characteristics of the upper cervical spine (C0-C1-C2) were unique to the rest of the cervical spine. Based on these two reasons, only the kinetic data from C3 to C7 were calculated in order to simplify the mathematical model.

The skull weight was treated as the input externally applied load and was normalized to a value of 1.0. Even though the mathematical model did not calculate contact forces acting between the skull through the segment C2, kinematic data of these bodies were used as input to the model in order to translate the loading from the skull to C3. In the model, C3 through C7 were treated as Bodies, and Skull through C2 were treated as massless Frames, which had the same dynamic characteristics, but had no mass. In this way, the system included a whole dynamic chain from the skull through to C7. Specifically, segments from C7 to C3 were modeled as rigid bodies referred to as 'N', 'A', 'B', 'C', 'D' and the skull through C2 were modeled as the frames referred to as 'E', 'H' and 'K' while the C7 was assumed to be fixed and treated as the Newtonian reference frame (Figure 3.3). The coordinate was setup at the mass center of each vertebral body, with their reference axes oriented in anterior-posterior (AP) and superior-inferior (SI) directions referred to as 1 and 2 directions in free body diagram (FBD). 2D motions were allowed for the adjacent bodies, which included 2D translations and rotations. According to the biomechanical feature of the vertebral body, the contact point

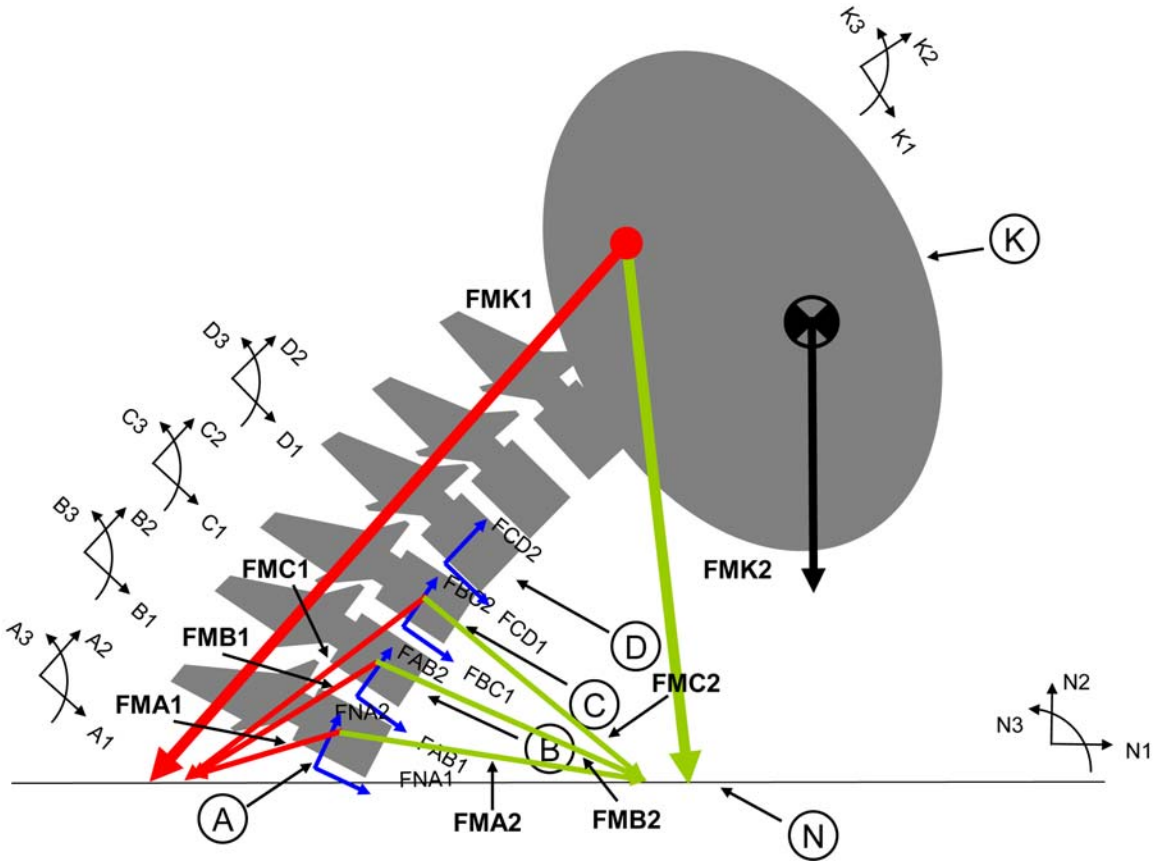


Figure 3.3 Free body diagram of 2D mathematical model of the cervical spine with pseudo-muscular forces.

between two bodies was modeled as a moving point, instead of a fixed point, which moved from the most anterior corner to the most posterior corner during the motion from flexion to extension. This mathematical model was then used to calculate the minimum resultant contact forces between the segments and the pseudo-muscular forces at each vertebral body. The force vector FNA represents the resultant contact force between the C6-C7, where FNA1 and FNA2 represent components of this force perpendicular and parallel to the vertebral body C6, respectively. In other words, FNA1 represents the compression force at the C6-C7 level; FNA2 represents the anteroposterior force at the same level. FAB, FBC and FCD were defined using the same approach and were related to C5-6, C4-5 and C3-C4, respectively. In the cervical spine, there are superior muscles, such as the sternocleidomastoid, anterior, middle and posterior scalene, splenius capitis and longissimus at each vertebral body, and inferior muscles, such as intertransversarius, rotatores, multifidus, and interspinales to connect adjacent levels [63, 64]. In addition, there was up to seven different groups of ligaments around each segment at the lower cervical spine. Reduction techniques were used in order to simplify all these muscles and ligaments into two groups: flexors and extensors, which had the same attachments but different functions during the flexion-extension activity. Most the muscles in the cervical spine are attached at the transverse and posterior processes. Because only two pseudo-muscular forces could be applied to each segment, the assumption was made that the attachment points were at the transverse processes of each vertebral body. There were a total of four unknowns -- two contact force components and two pseudo-muscular forces at each vertebral body in the mathematical model. Since, there are only three dynamic equation associated with each 2D rigid body, it was not possible to solve all the

unknowns. With the knowledge of the anatomical structure and musculoskeletal functions of the cervical spine, a technique was utilized to determine which pseudo-muscular forces were automatically turned on and off in this model. During the motion from flexion to the neutral position, FMA1, FMB1, FMC1 and FMK1 were used to simulate pseudo-muscular forces resulting in the generation of motion. For the motion from the neutral position to extension, FMA2, FMB2, FMC2 and FMK2 were utilized (Figure 3.4). Then, these two forces were combined as one single unknown force in the program and automatically switched its direction depending on the motion of the skull. In this way, the entire flexion-extension motion could be simulated more efficiently. The load, the skull weight, was assumed to be 8.1% of the body weight [65]. The orientation angles of pseudo-muscular forces were derived from the fluoroscopic images directly and in accordance with their attachments with respect to the horizontal directions.

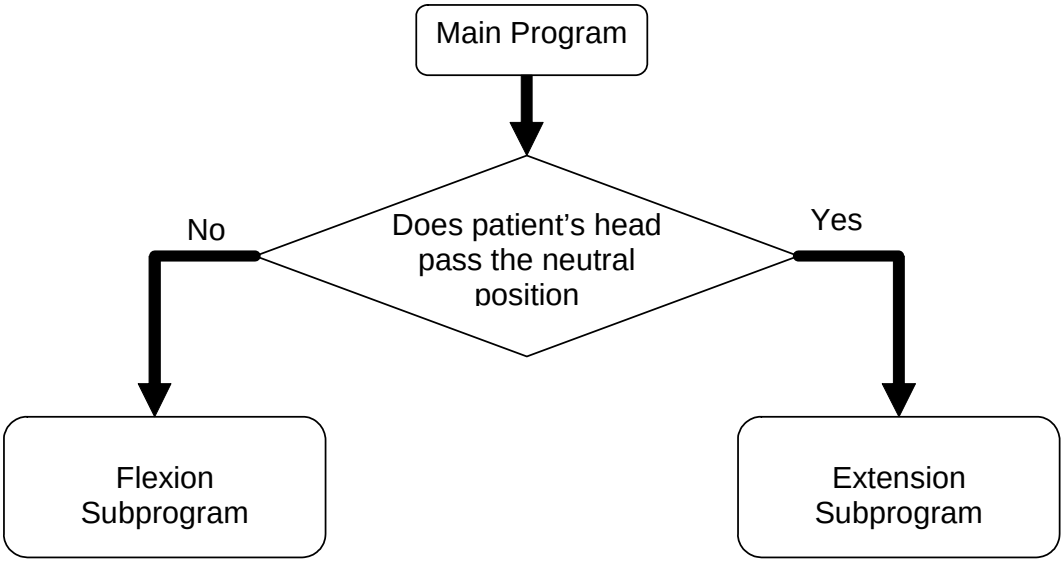


Figure 3.4 Switching Logic used in the mathematical model.

3.1.3 Statistical Analysis Method

Statistical analysis was performed to compare the means of the kinematic and kinetic characteristics among the groups. Since these groups were randomly selected, this study was an independent samples design and the corresponding statistical method was chosen. Because of the different preoperational situations and small sample size (10<<30) in each group [66], an unequal variance test was used to asses the assumption of normality before a *t* test of small samples was used to determine the statistical difference between groups.

3.2 3D Kinematic Analysis Method

After conducting the 2D kinematic and kinetic analysis of four different cervical spines, the 3D kinematic and kinetic characteristics of normal, pathological, and postoperative spines were extensively studied. A human experiment including fluoroscopic, CT, and MRI experiments was performed at Vanderbilt University Medical Center (VUMC) (IRB #060424). Since cervical disc replacement surgeries are still under clinical trial in the U.S, this 3D study only included normal, degenerative, and ACDF subjects. However, this method could also be applied to subjects having a disc replacement cervical spine.

3.2.1 Subject Demographics

In accordance with statistical data from the National Health Survey of 1960-1962, the range of heights for the subjects was limited to a range between 59-72 inches and the

weights to a range between 105-220 lbs in order to ensure that the experimental results properly represented the population. Many researchers have reported there is no statistical difference in CROM between male and female subjects [67, 68]. However, females in fact have a larger CROM compared to males [24, 69, 70]. Age has been widely accepted as one of the important factors affecting CROM [67, 68, 71, 72]. For example, it has been shown that people at an older age usually have a smaller CROM. So, gender and age were considered to be appropriate criteria during the process of subject selection. Recruited subjects were then grouped according to these factors, but weight and occupation were also used in order to eliminate potentially confounding variables which could affect in kinematic and kinetic comparison. Normal, degenerative, and ACDF subjects were brought to the study by Dr. Joseph S. Cheng, through VUMC. These subjects were recruited from a pool made up of the faculty, staff and student body of VUMC via e-mail and postal notification. Before the experiment, each potential subject was screened to ensure that he/she had satisfied the required clinical criteria. In addition, a serum pregnancy test was performed on females with child bearing potential. Detailed information of the study was thoroughly explained to the subjects who expressed interest.

The clinical/demographic information included gender, age, body weight, height, reasons for spinal surgery, postoperative time, and any other conditions that might affect experimental results. These data were either measured before the experiment or taken from the previous clinical documents of the participant; and the personal information was marked by VUMC. All of the collected data helped better understandings of the results

derived from this research study and more accurately led to a correlation of the dynamic characteristics with clinical outcomes.

The subjects in this study consisted of middle age females between the ages of 35 and 45 years old (Table 3.1). All of these subjects were at the time employed in the hospital as nurses. Also, they had similar heights but different weights. None were pregnant during the period of experiments and, with the exception of the normal control, the ACDF and degenerative subjects all had symptom at the C5-C6 level.

The normal subject exhibited no pathological symptoms in the cervical spine, had never suffered from serious neck pain, or experienced a loss of cervical spine motion during her life time. Her fluoroscopic, CT, and MRI images showed no evidence of any kind of abnormal bony structures or motion patterns.

One subject experienced cervical spondylosis and stenosis primarily at the C5-C6 level with neural effacement in her cervical spine. However, there were no discernable cord signal changes.

The third subject had a spinal fusion surgery without any disc replacement. Before the operation, this subject had a two-month history of neck pain, and this pain would occasionally spread into her left shoulder as well as the upper aspect of her arm. There was also an intermittent pain occurring in her fingertips. According to the Visual Analog Scale (VAS), she scored a 30 to 40 out of 100 while using pain medications. Therefore, an ACDF at the C5-C6 level was performed on the subject at VUMC. The Synthes ACDF trailers were used with a 7 mm graft, which was filled with 1 cc of demineralized bone matrix. A 16 mm anterior cervical plate was fixed onto the ACDF area through the use of locking screws. During her 1 year follow-up, the subject's

Table 3.1 Demographics of selected subjects.

Subject	Sex	Height	Weight	Age	Disease	Operation	Follow-up	Occupation
Normal	Female	67 in	160Lb	36	NA	NA	NA	Nurse
ACDF	Female	66 in	125Lb	36	Myeloradiculopathy	ACDF	39 Months	Nurse
Degenerative	Female	63 in	186Lb	44	Spondylosis	NA	NA	Nurse

anterior and lateral X-rays showed a strong callus formation and a fusion mass of the bone graft without any degenerative symptom at the adjacent levels. All instrumentation remained intact, and her total follow-up time was 39 months at the time of this experiment.

3.2.2 Risks Specification and Protection

For CT exams, the total number of examined subjects has increased from around 3 million in 1980 to about 57 million in 2000, according to the United States FDA Center for Devices and Radiological Health in the U.S [73-75]. Approximately 2-3 million of the 57 million CT exams are performed on children annually. An estimated 25~30% of all medical decisions are made by considering imaging findings, according to the World Health Organization. Just as other clinical studies, there were some possible risks in this experiment. CT scans can have a harmful effect on the human body, but the dosage of radiation received in a regular CT scan is not enough to produce detectable health problems. Overall, the benefit of CT scans overwhelms the risk for the subjects.

The average dosage received by the cervical spine during a CT scan is less than 200 mrem. Fluoroscopy "on-time" for each activity was limited to <10 seconds, and there were a maximum of six trials for each participant. Also, the typical exposure time for a given participant was restricted to less than 60 seconds. The exposure rate typically applied to the cervical spine was somewhere between 600 – 1,000 mrem/minute, depending on the size of the subject. Therefore, a subject participating in the fluoroscope experiment for less than 60 seconds (1 minute) was exposed to a maximum amount of 1,000 mrem of radiation. With the addition of <200 mrem of radiation exposure during a

CT scan, the total exposure for a subject was less than 1,200 mrem in the entire experiment. It has been demonstrated that exposures below 5,000–10,000 mrem, which includes occupational and environmental exposures, normally will not cause risks of health effects. So, at this maximum of 1,200 mrem, the subjects experienced very little risk.

Proper shielding was utilized during the fluoroscope experiment. It is known that the thyroid gland is sensitive to radiation with an annual thyroid dose limit of 50,000 mrem. With only a total exposure of <900 mrem for the fluoroscopy and CT scan procedures, the mrem level was far below the established limits for the participants. The risk on the thyroid was so small, in fact, that the subjects did not even need to wear a thyroid shield during the experiment. However, a lead apron was worn by the subjects to cover the area below the first rib in order to reduce any possible radiation exposure. Additionally, all the researchers and nurses were required to wear full lead aprons.

MRI uses radio waves and a rapidly changing and powerful magnetic field instead of X-rays. These magnet field and radio waves are capable of creating images of the human body. The procedure has been approved by the FDA and has been used by hospitals routinely. This regular MRI used in this experiment has been proven to be safe for subjects for this reason.

3.2.3 Human Experiment Procedure

At the beginning of the experiment, the study protocol was explained to subjects in full detail. Consent and HIPAA forms were signed by each subject, and all subjects were free to quit the experiment at any time without any reason. The entire experiment

for each subject was arranged to occur on the same day, or at most conducted within 48 hours apart.

First, CT and MRI scans were performed at UVMC. The CT scan was performed on a Philips EBW 64. Its minimum slice was 0.67mm thick with every 0.33mm overlapping; this allowed the collection of more detailed information from the images. The CT scan software used was Brilliance CT V2.0.2.6450, November 15, 2005, from Philips medical System, Netherlands. Following standard instructions, the CT scan was conducted from a part of the skull to the first rib at the minimum machine-allowed increment of 0.67 mm for the scan procedure; it was further broken down to 0.33mm in the workstation (Figure 3.5). MRI experiments were performed on the Philip Intera-Achieva 1.5T machine (Figure 3.6). There was no radiation during the testing. Also, for safety reasons, the subjects that had any metal devices (such as aneurysm clips in the brain, cochlear implants in the body, or any metal pieces close to important organs) were excluded from the MRI experiment and thus, could not participate in the study. Before the MRI test, subjects were not allowed to carry any metal objects, such as watches, credit cards, hairpins, pens, etc. After all the proper precautions were taken, T1, T2, and three dimensional scans were then performed.

No medication was taken by the subject before the fluoroscopic experiment. In this study, the fluoroscope experiments were performed by using an OEC 9800, which was a vascular unit fluoroscope machine with 12 inch X-ray and image tubes. Under fluoroscopic surveillance, each participant then performed three activities: flexion-extension, axial rotation, and lateral bending (Figure 3.7).

Before any of these activities, the RT modified the center of the image tube for

Figure 3.5 CT experiment for a subject.

Figure 3.6 MRI experiment for a subject.

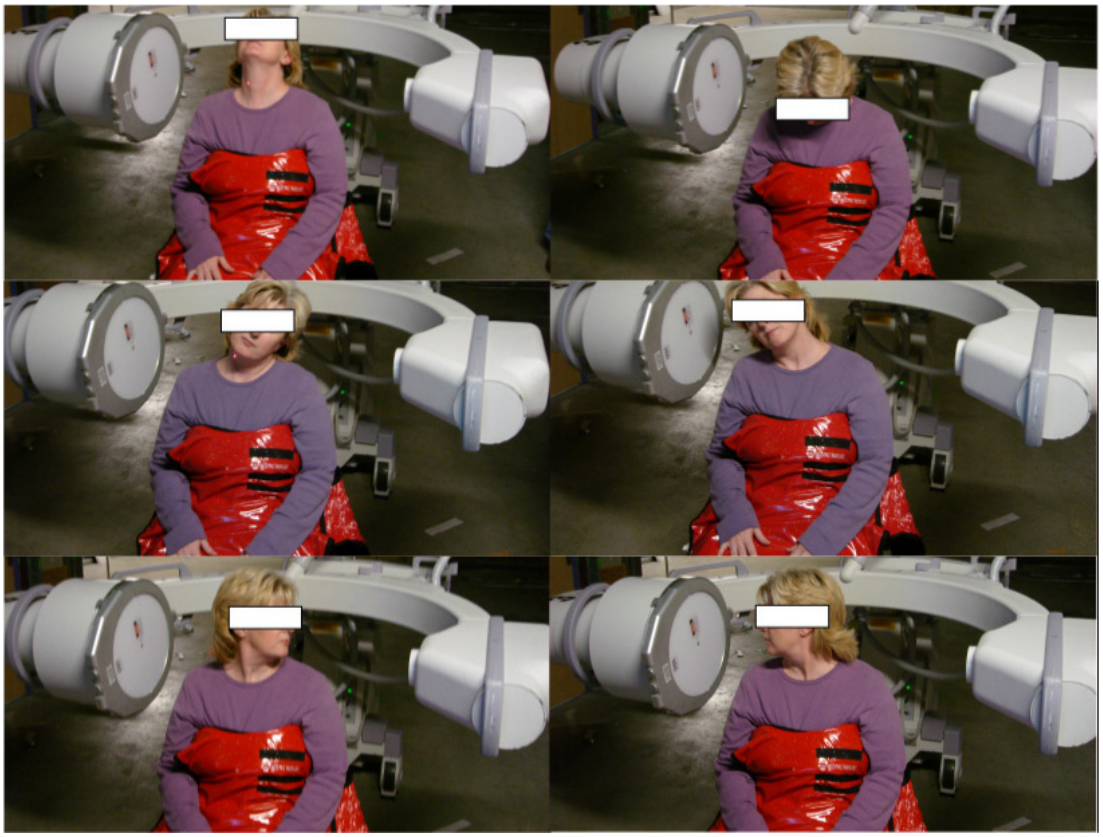


Figure 3.7 Fluoroscope experiments for one subject (Top: flexion-extension; middle: lateral bending; bottom: axial rotation).

each participant according to the length of his/her cervical spine in the neutral position. The purpose of this procedure was to capture a portion of the skull along with the whole portion of C1 through the first rib in the fluoroscopic images while ensuring the consistency of images in each experiment. Each activity was performed by the subject until an acceptable movement was achieved or until the fluoroscopy machine was turned on for 10 seconds. If no suitable fluoroscope video was recorded within those 10 seconds, one more trial was requested in order to reach a satisfactory result. All subjects attempted at most two trials for any of the required activities. If any activity was not achieved, the participant was dismissed and did not move on to the next procedure.

The participant was required to rest between each activity in order to relax their muscles and ligaments acting across the cervical spine segments. The start/end positions and speed to produce the required motions were done in concordance with the subject's comfort level. The only requirement was that they should perform the motions as smoothly as they could without intermission. In this study, no subjects chose to quit the experiments and all met the regulations provided above.

The total time of fluoroscopic experiments for each participant was limited to 30 minutes, and each participant was under fluoroscopic surveillance for less than 60 seconds. Only the area near the cervical spine was recorded on a fluoroscopic videotape from the fluoroscopy machine. Each of the study participants was asked to sit erectly on a chair, restricting the motions of their upper trunks and to place their cervical spines as close as possible to the image intensifier. Before turning on the fluoroscopic unit (no radiation), subjects were asked to perform the required activity to ensure they could do it correctly. Moreover, it allowed subjects to practice these activities and improve

performance.

Flexion-extension

1. During the flexion-extension activity, the subject began at the hyper-extension position (start position). The radiology technologist (RT) then started the fluoroscopic machine imaging in the sagittal plane once the participant was ready.
2. The participant then flexed his/her neck until it passed the neutral position and reached its maximum flexion position with her chin as close as to her chest (full flexion).
3. The RT stopped the fluoroscopy machine at the participant's full flexion position.

This activity was purposely performed without coupling motions of the head in the horizontal and frontal planes by the subjects.

Axial Rotation

1. During the axial rotation activity, the participants turned their heads as much as they could over their left shoulder (start position) and the RT started the fluoroscopic machine imaging in the sagittal plane after the participant once ready.
2. The participants then rotated their necks until it passed the neutral position and reached the participants' maximum degree of rotation over their right shoulder (end position).
3. The RT stopped the fluoroscopy machine when the participant reached his/her

maximum degree of rotation.

This activity was purposely performed without coupling motions of the head in the sagittal and frontal planes by all subjects.

Lateral Bending

1. During the lateral bending activity, each subject leaned their head as far as left as they could (start position) and the radiology technologist (RT) started the fluoroscopic machine imaging in the front plane once the participant was ready.
2. The participant then tilted his/her neck until it passed the neutral position and had been leaned as far to the right as possible (end position).
3. The RT stopped the fluoroscopy machine once the participant had leaned his/her head as far to the right as possible.

This activity was purposely performed without coupling motions of the head in the sagittal and horizontal planes by all subjects.

During the above activities, if any other motions were coupled with it, it might indicate abnormal structures or motions of the examined cervical spine in contrast to the natural motion functions of the normal cervical spine.

3.2.4 3D Kinematic Analysis Method

Firstly, the image processing functions (such as threshold, grow, shrink, fill and smooth) were used to roughly extract the profile of vertebrae from surrounding soft tissues in the commercial software -- Amira™ (Mercury Computer Systems, Inc,

Chelmsford, Massachusetts) (Figure 3.8 & Figure 3.9). Then, anatomic knowledge was utilized to further determine the profile of vertebrae on each CT image. The segmented CT images were then used to rebuild the 3D CAD model of each vertebra (Figure 3.10 & Figure 3.11).

Fluoroscopic image frames were captured at every second and analyzed to determine 3D *in vivo* kinematics of three activities: flexion-extension, axial rotation and lateral bending. These fluoroscope image frames and their corresponding CT and MRI data were loaded into a database for further analysis at CMR, the University of Tennessee, Knoxville, USA. The fluoroscopic images were projected onto an image plane, and 3D CAD models of each vertebra were added to the scene by using the custom developed software SAAM (CMR, The University of Tennessee, Knoxville, USA). The relative 3D poses of vertebrae were determined by registering appropriate 3D CAD models onto sequential 2D fluoroscopic images. The correct fit was achieved when the superimposed 3D CAD models covered only the silhouettes of the vertebra on the fluoroscopic image. 3D translational and rotational data of each vertebra were recorded and important parameters were calculated by a custom-designed program created in Matlab. After all the frames were analyzed for a single activity, kinematic curves were created in order to describe the 3D motions of the cervical spine. Finally, the kinematic data with its corresponding muscular and ligament information were inputted into 3D mathematical models in order to predict 3D *in vivo* kinetic data.

In this study, a new method was used to determine the origin of a coordinate system at a vertebra. The origins were automatically chosen by creating a bounding box which enclosed every part of the vertebral body while having the minimum volume

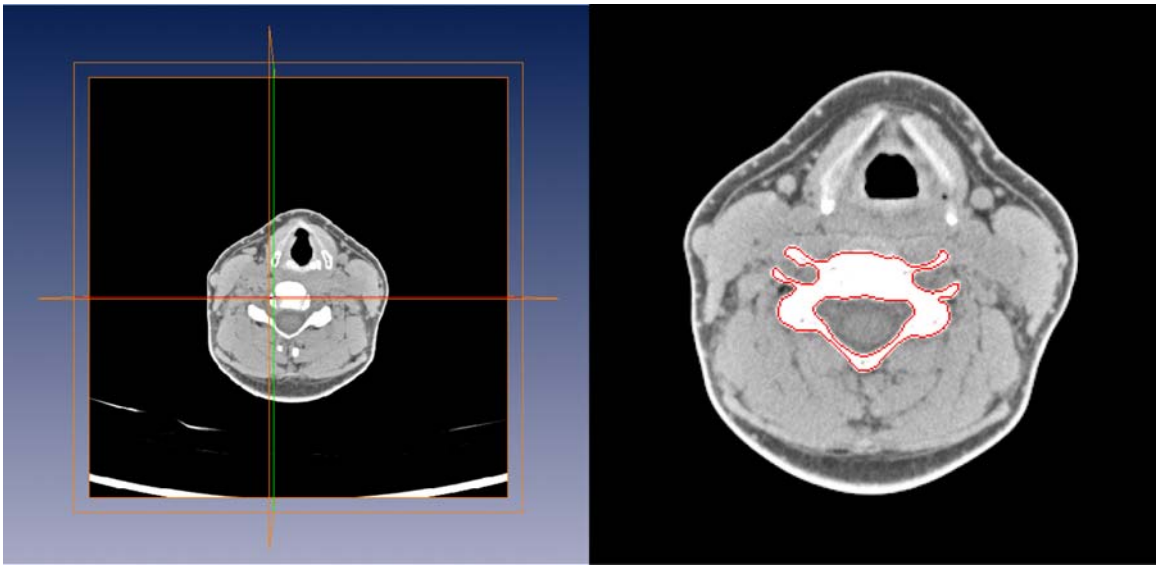


Figure 3.8 A CT segmentation image of the cervical spine in the axial plane.

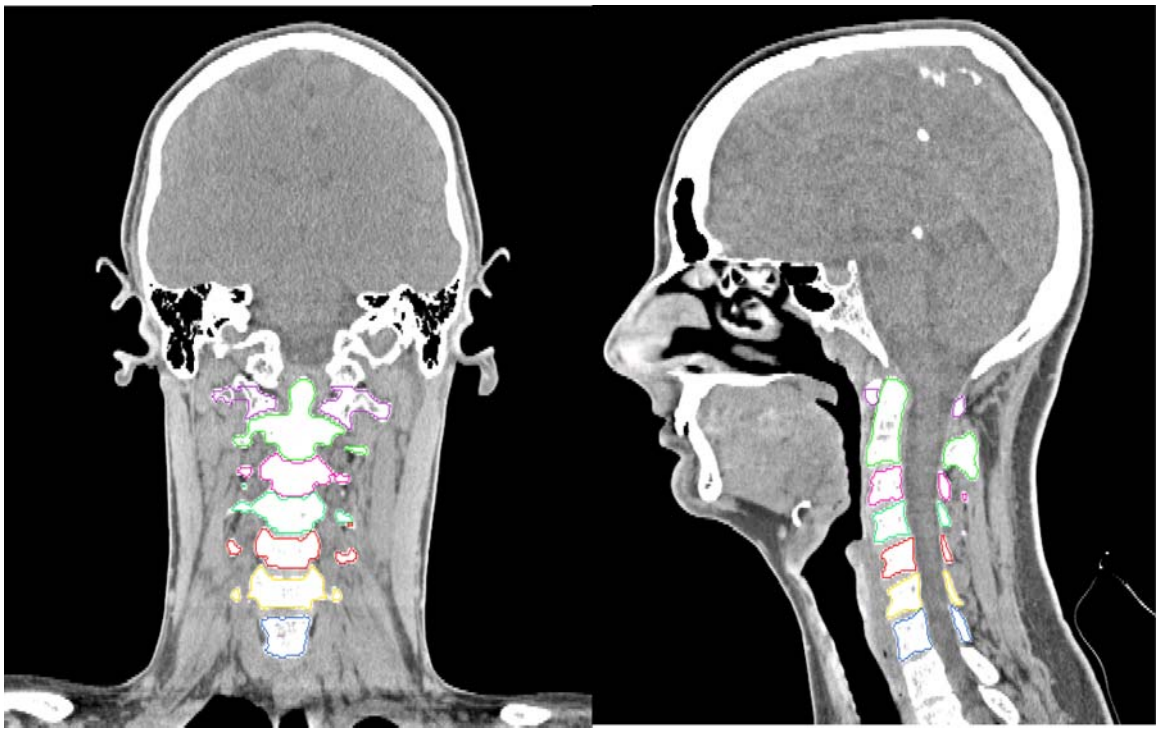


Figure 3.9 A CT segmentation image of the cervical spine in the coronal and sagittal planes.

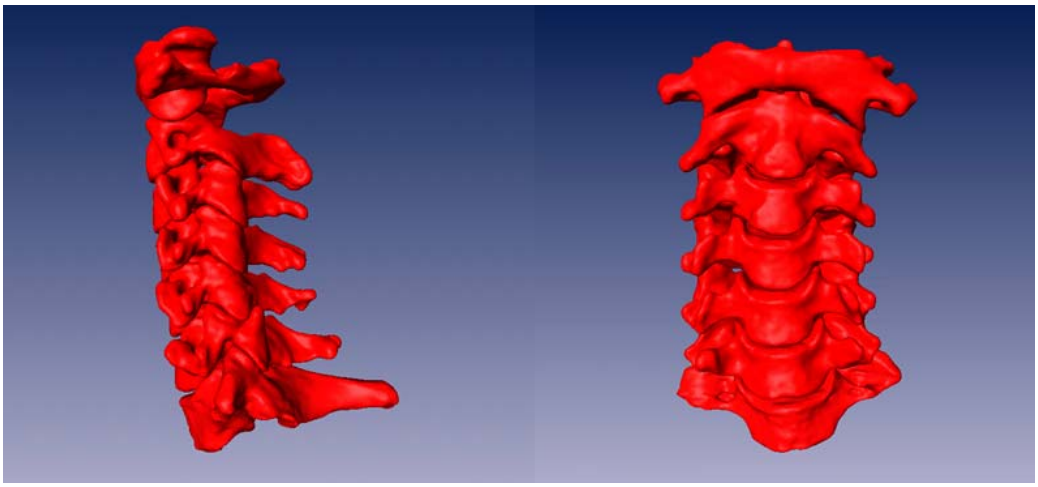


Figure 3.10 3D CAD models of a normal cervical spine from CT images after segmentation.

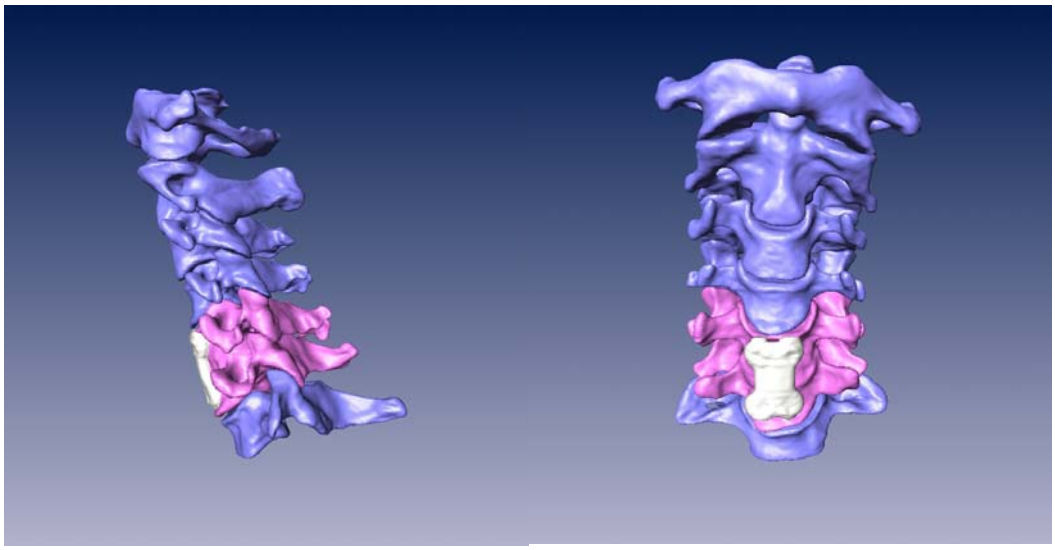


Figure 3.11 3D CAD models of a cervical spine with ACDF from CT images after segmentation.

(Figure 3.12 & Figure 3.13). The origin of the coordinate on a vertebra, initially defined at the beginning of the analysis, could be accurately moved to the center of the bounding box because the coordinates of its eight corners were already known by the computer.

3.3 3D Kinetic Analysis Method

3.3.1 Mathematical Modeling Description

In this 3D mathematical model, the skull weight (referred as 'SK' in the FBD) was treated as the only externally applied load during the motion activities. The skull weight and the cervical spine together were assumed to represent 8.1% of the body weight [65]. The skull, through C7, was each modeled as a rigid body. Specifically, C7 was assumed to be fixed and hence treated as the Newton reference frame. Segments from C7 to the skull were modeled as rigid bodies ('A','B', 'C', 'D', 'E', 'H', 'K', and 'S') (Table 3.2). On each vertebral body, the origin of the local coordinate system was set-up at its mass center, with their corresponding unit vectors oriented in anterior-posterior (AP) and superior-inferior (SI) directions and referred to as the 1 and 2 directions. Unit direction 3 was determined by using the right hand rule (Figure 3.14).

Inertial properties of each vertebra about the corresponding major axis (Z axis in SAAM and 3 in the mathematical model) during the flexion-extension were evaluated according to the reference frame [37]. Since only very small coupled motions were present about the other two axes, inertial properties about these two axes were assumed to equal to zero. It was also assumed that the mass of the neck was distributed throughout the vertebrae.

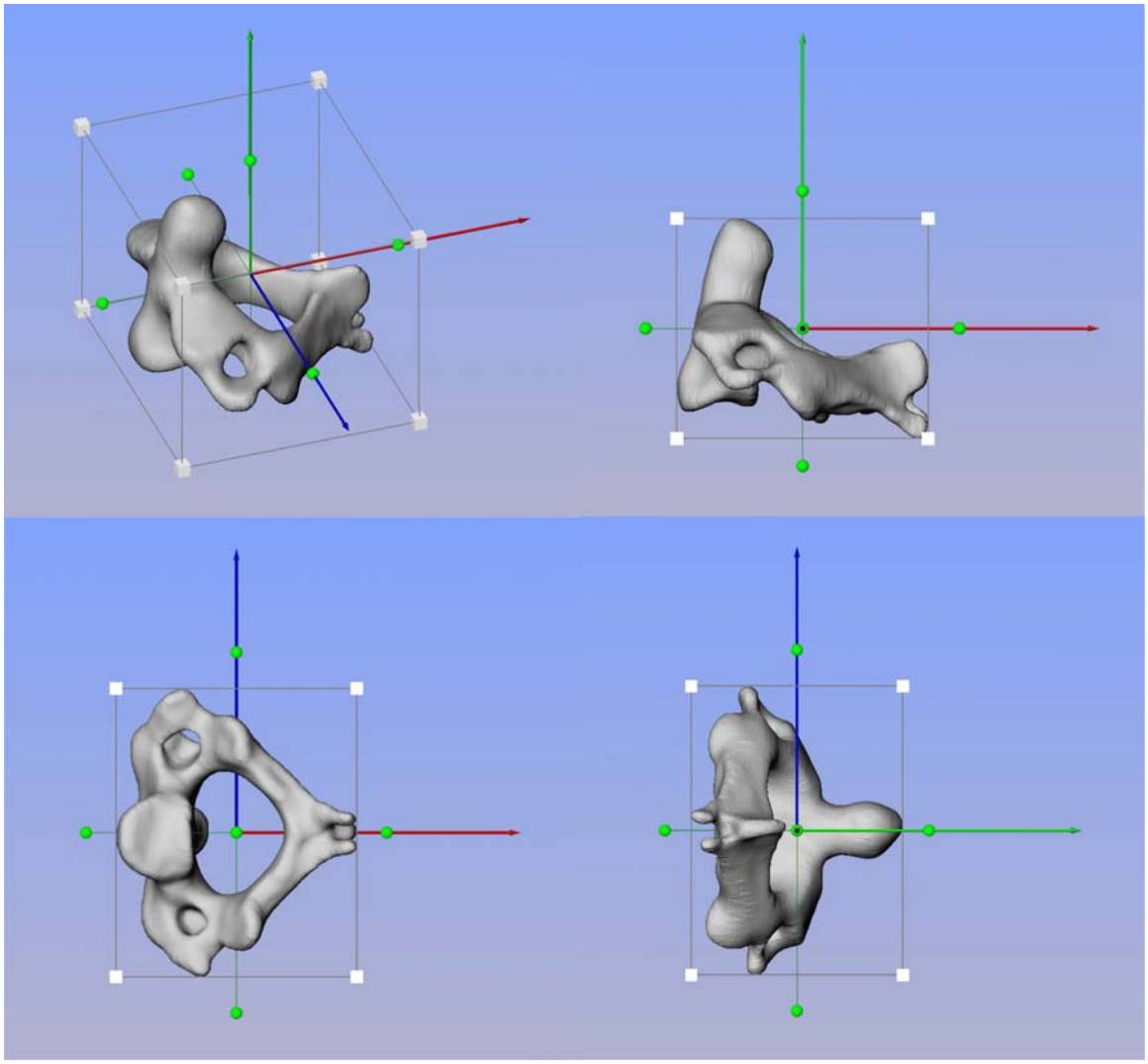


Figure 3.12 The origin for the coordinate system of a vertebra (C2) was chosen automatically by using the bounding box method.

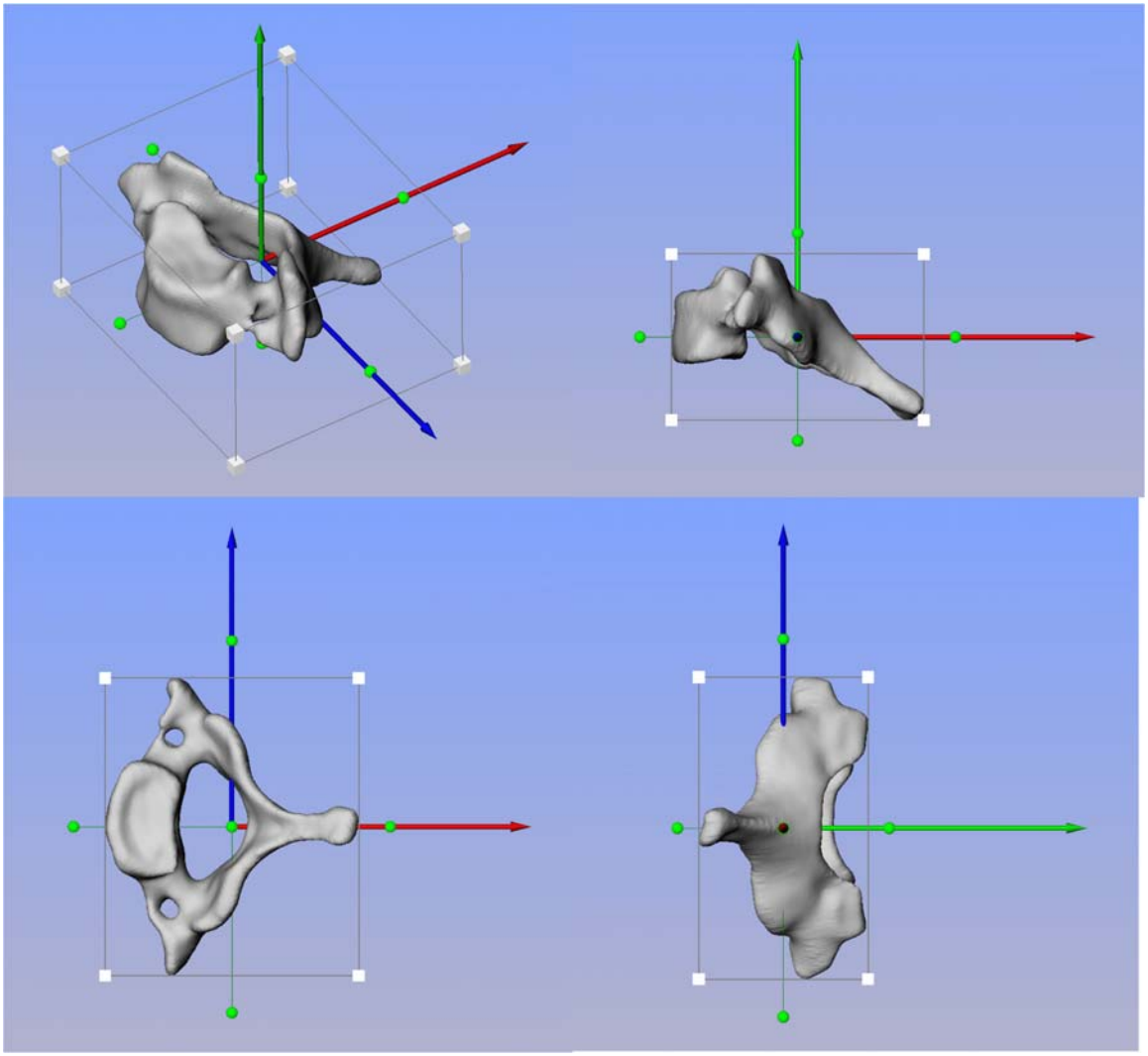


Figure 3.13 The origin for the coordinate system of a vertebra (C7) was chosen automatically by using the bounding box method.

Table 3.2 Description of Rigid Bodies in the 3D Mathematical Model.

ANATOMIC PART	SKULL	C1	C2	C3	C4	C5	C6	C7	T1
BODY	S	K	H	E	D	C	B	A	N
MASS CENTER	SO	KO	HO	EO	DO	CO	BO	AO	NO
CONTACT POINT	SK	KS,KH	HK,HE	EH,ED	DE,DC	CD,CB	BC,BA	AB,AN	NA

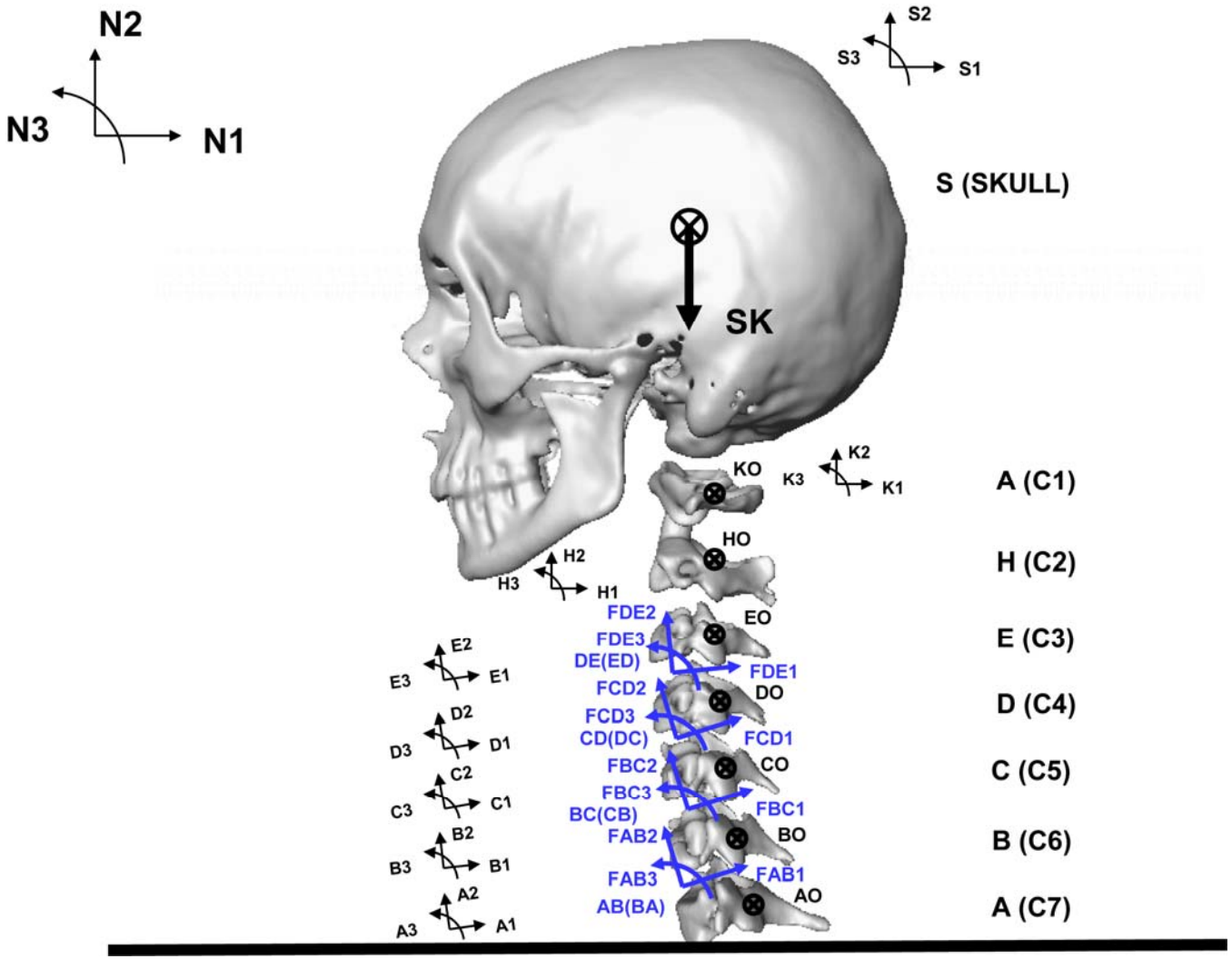


Figure 3.14 Free body diagram of the 3D mathematical model of the cervical spine only including the contact forces.

The mathematical model was not derived to calculate contact forces at the skull through C3 interaction (Figure 3.14), nor did it include the ligament and muscular forces acting across the skull through C2 vertebrae (Figure 3.15). The model did also input the kinematic data of the skull through C2 in order to translate the load from the skull to C3. Therefore, the system included the whole dynamic chain from the skull down to C7 without changing any dynamic characteristics of the cervical spine. The temporal functions of contact forces along the 1, 2, and 3 axis directions between adjacent bodies from C3 to C7 were then determined (Figure 3.14).

3.3.2 Contact Point and Contact Force

The contact point between A (C7) and B (C6) were referred as point 'AB' on body A and 'BA' on body B while the contact force between A and B acted at the interaction of the points AB on BA (and vice-versa). The three interactive force components were referred to as 'FAB1', 'FAB2', and 'FAB3'. 'FAB1' was the contact force along direction 1, parallel to the vertebra B in the B1> unit vector direction; 'FAB2' was the force along direction 2, perpendicular to the vertebra B in the B2> unit vector; and 'FAB3' was the force along direction 3. In other words, 'FAB1' was the anteroposterior force; 'FAB2' was the compression force; and 'FAB3' was the mediolateral force. The contact points and contact forces on other vertebral levels were labeled according to the same method and all contact points were modeled as moving points. Normally, for each vertebra, one contact point at the superior surface and one at the inferior surface were defined, but there was only one contact point at the superior surface on C7 and one at the inferior surface on C3. Firstly, let us only considered the

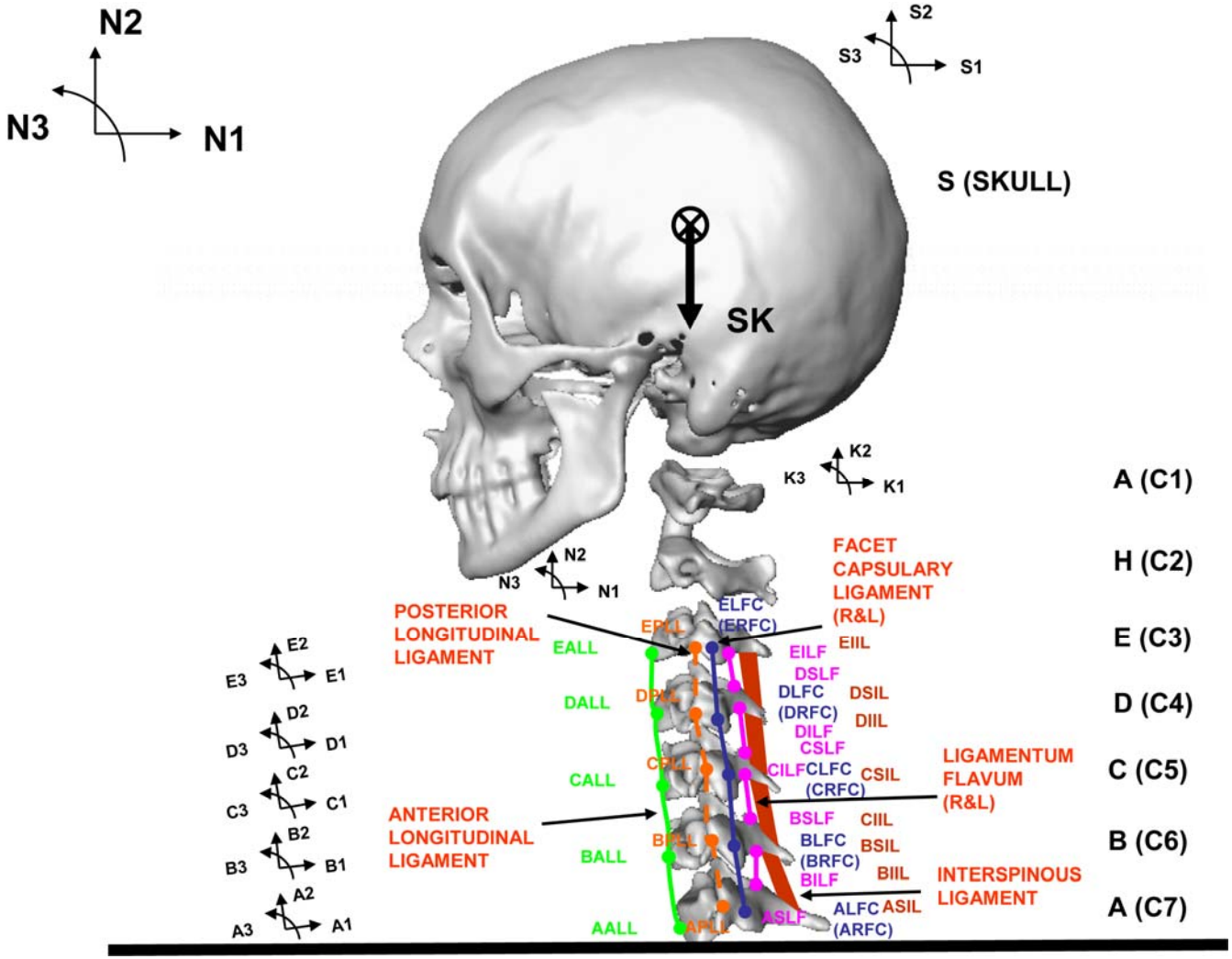


Figure 3.15 Free body diagram of the 3D mathematical model of the cervical spine about ligament forces.

inferior vertebral segment of two adjacent levels (Figure 3.16). The contact point was located at the superior part of this segment. During the flexion-extension activity, at the full flexion position, the contact point started at point B, which was at its anterior part of the vertebral body; at the full extension position, it was moved to point A, which was at the posterior part. The coordinates of these two points were digitized with respect to the origin of the coordinate at this vertebra. The temporal function of a contact point was created by using these coordinates with respect to the experimental time and was determined by the following formula:

$$\textit{Coordinates of the contact point} = A + t * (B - A) / T$$

T is the total experimental time that a subject performed the flexion-extension activity. The same method was used to determine the contact point at the inferior part, which moved from point D to point C (Figure 3.17).

3.3.3 Ligament Force

Due to the fact that C1 and C2 have different structures and functions, this research did not consider the ligament forces at these levels in the 3D mathematical model. During the flexion-extension activity, different ligaments provided different contributions to a particular posture of the cervical spine. At the full flexion position, the anterior longitudinal ligament is not in tension and therefore cannot restrain this movement; all the other six ligaments, though, are subjected to tension and hence provided motion restriction and maintained spinal stability. So, according to classic dynamics, the ligament force furthest from the anterior aspect had the largest lever arm. In turn, the farther the ligament was with respect to the posterior position, the larger the

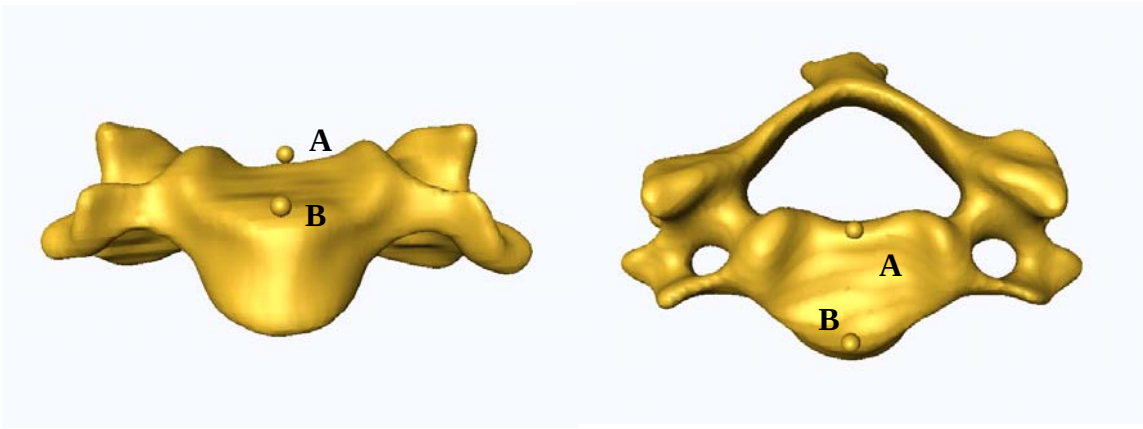


Figure 3.16 Modeling a contact point at the superior part of the vertebral body during the flexion-extension activity.

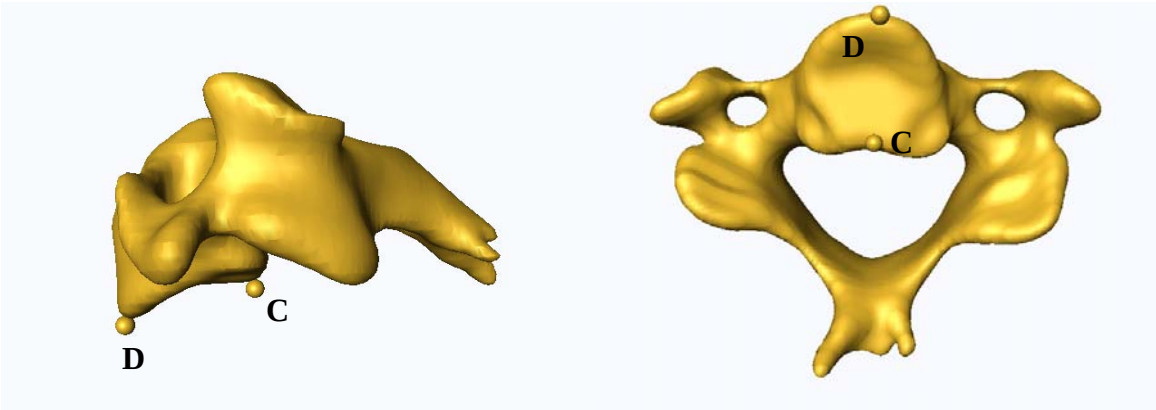


Figure 3.17 Modeling a contact point at the inferior part of the vertebral body during the flexion-extension activity.

contribution was in restricting this kind of motion at the full extension position given that all the ligament forces have the same magnitude. At the neutral position, though, all the ligaments provide motion restriction because they each had a small extension from their natural length. However, the ligament forces are very small since they are in the neutral zone (NZ).

Ligament forces were calculated using position vector information and then essentially, used as input variable restrictive forces in the mathematical model. In total, five different types of ligaments were considered for each vertebra in the 3D mathematical model: anterior longitudinal ligament (ALL), posterior longitudinal ligament (PLL), interspinous ligament (IL), ligament flavum (LF), and facet capsular ligament (FC) (Figure 3.15). The ligament attachment points from C3 to C7 were defined in Table 3.3. All of these contact points were digitized from 3D CAD models of the corresponding vertebrae according to the definitions in Yoganandan's study [76]. These data were evaluated through the use of MRI and CT images. The stiffness of these ligaments around the cervical spine was obtained from the previous study [76], which tested different types of cervical spine ligaments under in situ conditions through the use of cadaveric cervical spines. Although this research plotted the mean force-deformation curves of different cervical spine ligaments, only the slope of the most linear part of theses curve was numerically reported [76]. These data were directly used in the present study although many researchers have suggested that ligaments respond to the loading in a nonlinear function. The reason is that the examined activity in the present study was among normal daily activities, and the ligament forces should be in their physiologic phase. So, it is reasonable to use these data to predict ligament forces in the

Table 3.3 Description of the ligament attachment points on the vertebral bodies from C3 to C7 in the 3D Mathematical Model.

ANATOMIC PART	C3	C4	C5	C6	C7
BODY	E	D	C	B	A
ALL	EALL	DALL	CALL	BALL	AALL
PLL	EPLL	DPLL	CPLL	BPLL	APLL
LFC	ELFC	DLFC	CLFC	BLFC	ALFC
RFC	ERFC	DRFC	CRFC	BRFC	ARFC
LSLF	ELSLF	DLSLF	CLSLF	BLSLF	ALSLF
RSLF	ERSLF	DRSLF	CRSLF	BRSLF	ARSLF
LILF	ELILF	DLILF	CLILF	BLILF	ALILF
RILF	ERILF	DRILF	CRILF	BRILF	ARILF
SISL	ESISL	DSISL	CSISL	BSISL	ASISL
IISL	EIISL	DIISL	CIISL	BIISL	AIISL

1: ALL: the attachment point of anterior longitudinal ligament at a vertebral body;
2: PLL: the attachment point of posterior longitudinal ligament at a vertebral body;
3: LFC: the attachment point of facet capsular ligament at a vertebral body on the left side;
4: RFC: the attachment point of facet capsular ligament at a vertebral body on the right side;
5: LSLF: the attachment point of ligamentum flavum at the superior part of a vertebral body on the left side;
6: RSLF: the attachment point of ligamentum flavum at the superior part of a vertebral body on the right side;
7: LILF: the attachment point of ligamentum flavum at the inferior part of a vertebral body on the left side;
8: RILF: the attachment point of ligamentum flavum at the inferior part of a vertebral body on the right side;
9: SISL: the attachment point of interspinous ligament at the superior part of a vertebral body;
10: IISL: the attachment point of interspinous ligament at the inferior part of a vertebral body;

mathematical model. These ligaments were divided into the upper cervical levels including C3 and C4 and the lower cervical levels from C4 to C7; different parameters were used to calculate these ligament forces.

The ligament forces were named after the above ligaments (Figure 3.18 & Figure 3.19). There are two individual facet capsulary ligaments on the left and right sides of each vertebra, and the two different forces of these facet capsulary ligaments were represented in the mathematical model (Figure 3.15). Because it was difficult to pick up the attachment points of the ligament flavum by following the definition in Yoganandan's study, two attachment points on the left and right laminas were used. The ligament flavum forces were calculated according to the average of these two attachment points. The initial length of each ligament was determined at the neutral position. Then, the deformation was calculated by subtracting its current length from its initial length at each position with respect to time. Ligaments cannot apply a compressive force, but only a restrictive tensile force. So, if a deformation at any time became smaller than zero, no ligament force was deemed to be applied. An algorithm was designed for this ligament characteristic and described as the following:

Deformation = Current length – Initial length;
If Deformation >=0
Ligament force = Deformation × Stiffness
Else if Deformation < 0
Ligament force = 0
End

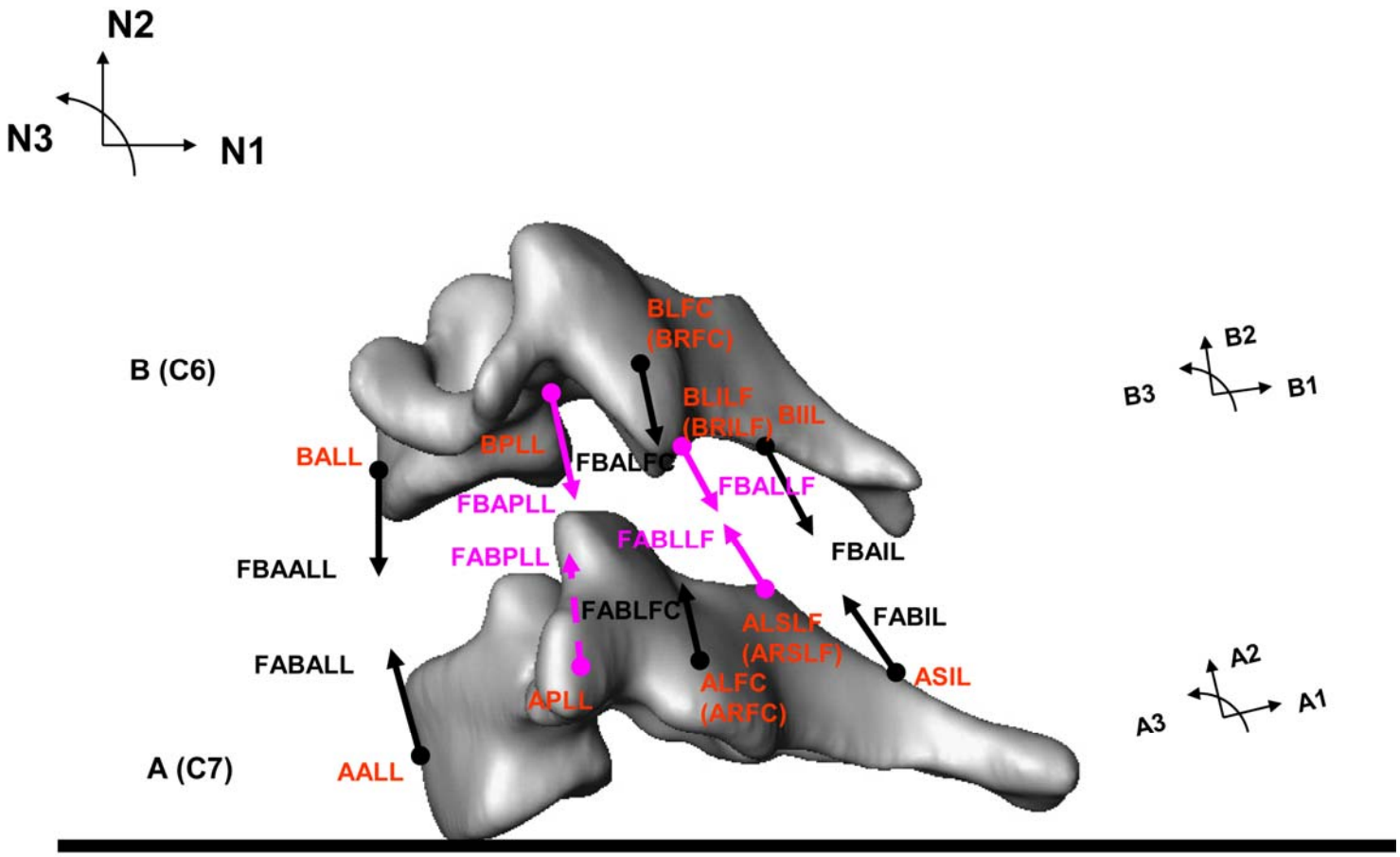


Figure 3.18 Modeling of ligament forces for two adjacent levels (Sagittal View).

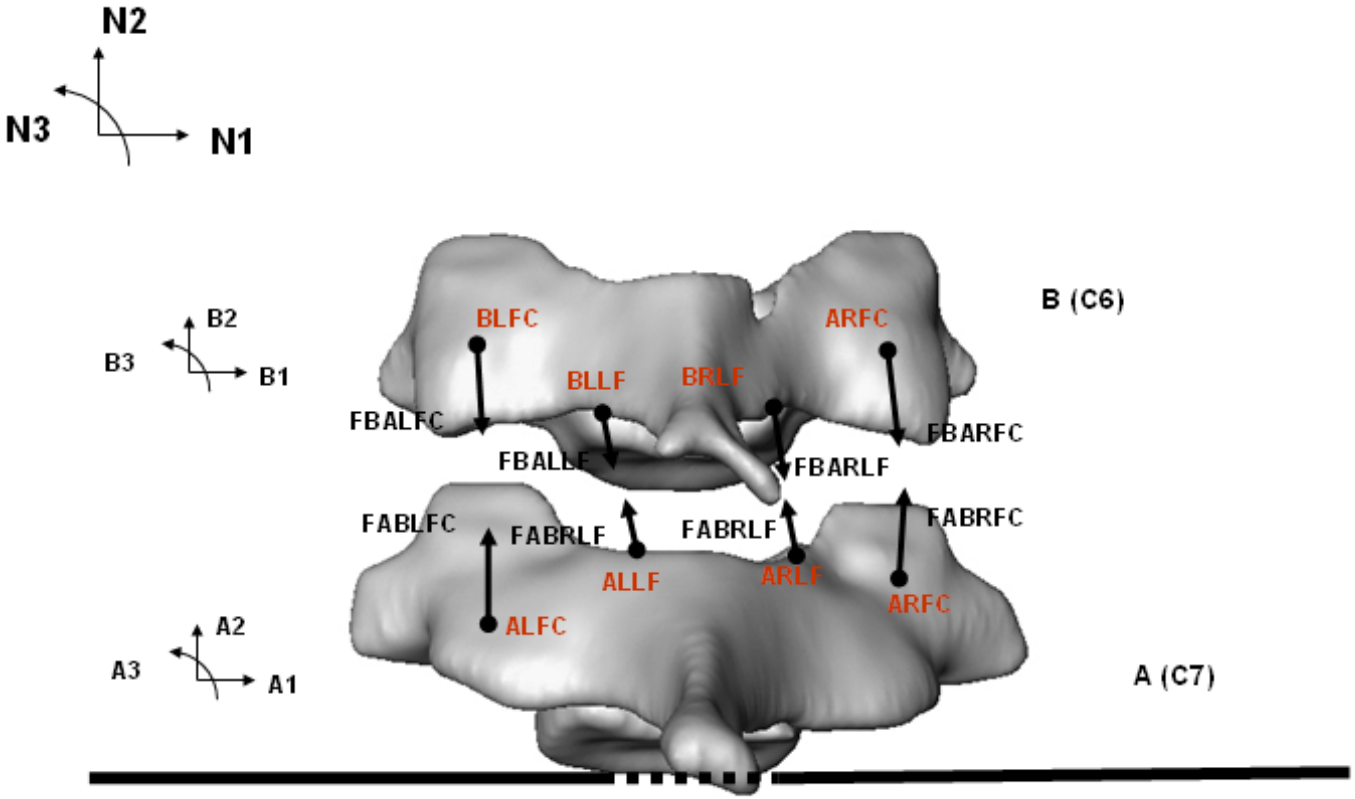


Figure 3.19 Modeling of ligament forces for two adjacent levels (Back View).

3.3.4 Muscular Forces

The cervical spine has complex muscular structures which help it perform fast and accurate motions. But in this mathematical model, only the major muscular forces were included. In total, two groups of muscles (flexors and extensors) were considered in the model; they are listed in Table 3.4. From the full extension to the full flexion position, different muscle groups were activated and included in the mathematical model as the major muscles depending on the movement. For example, when a subject performed a flexion-extension activity from full extension to neutral position, the force of the flexor muscle were considered (Figure 3.20); However, the extensor muscle were treated as the major muscles from the neutral to flexion position (Figure 3.21).

Muscular attachment points were identified either from 3D CAD models according to their anatomical data or by digitizing from MRI images. Figure 3.22 to Figure 3.24 are the MRI images in sagittal plane of the normal, degenerative, and ACDF cervical spine.

Table 3.4 Muscles included in the 3D Mathematical model.

Extensors (Neutral-Flexion)	Flexors (Extension-Neutral)
Trapezius	Sternocleidomastoid
Splenius Capitis and Cervicis	Longus Capitis and Colli
Semispinalis Capitis and Cervicis	Scalenes-Anterior,Middle, Posterior
Longissimus Capitis and Cervicis	

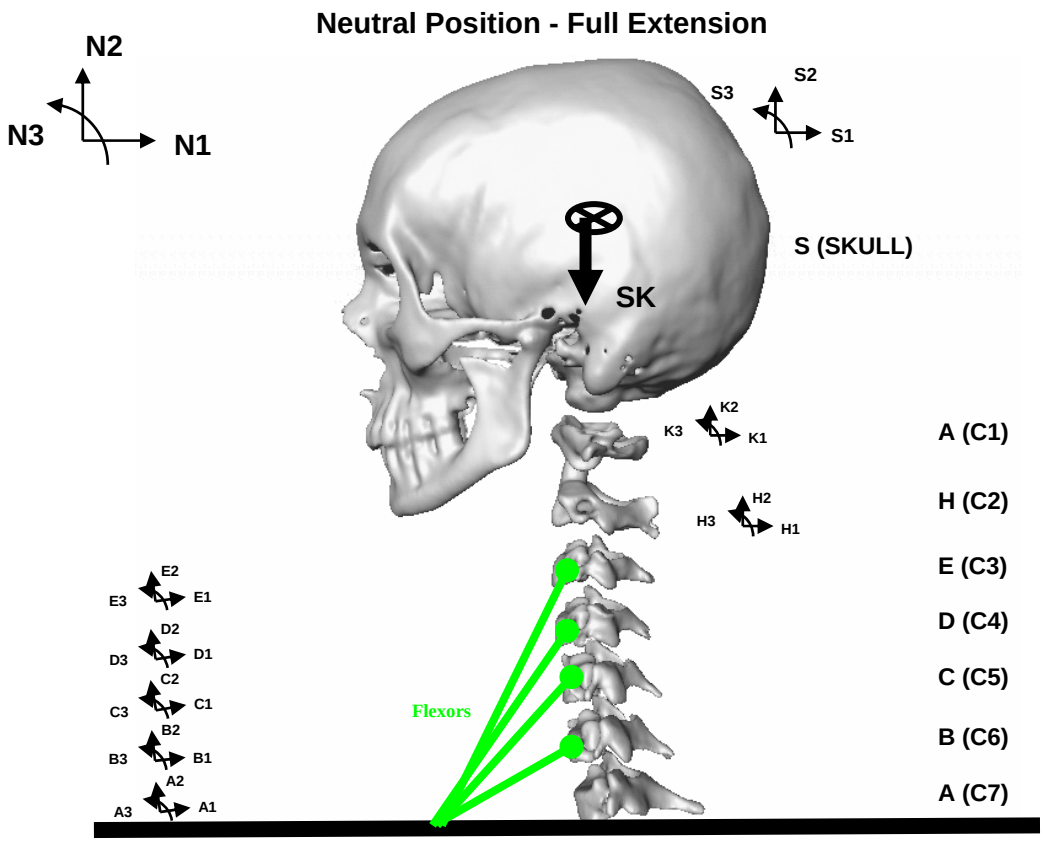


Figure 3.20 Modeling of muscular forces included in the 3D mathematical model for the activity from full extension position to neutral position.

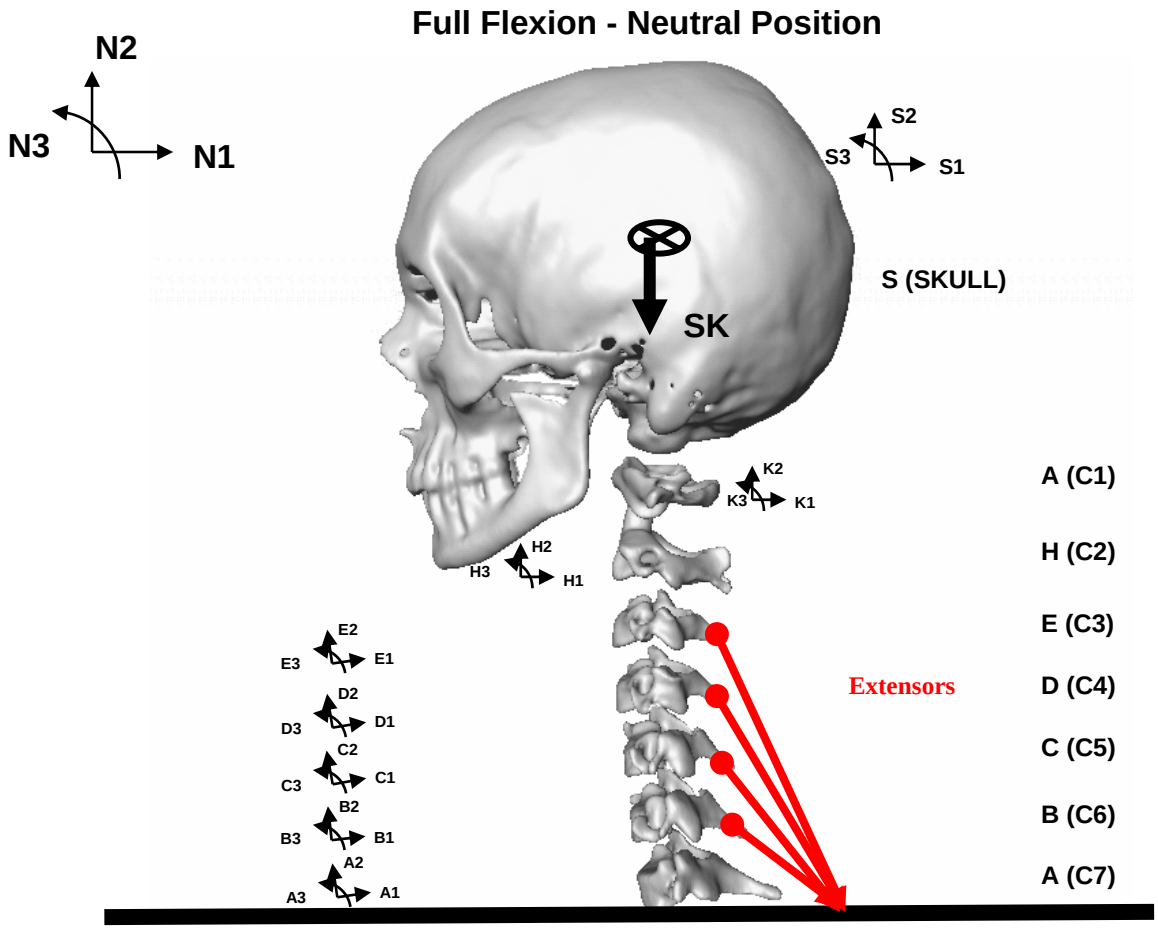


Figure 3.21 Modeling of muscular forces included in the 3D mathematical model for the activity from the neutral position to the full flexion position.

Figure 3.22 A MRI images of the normal subject.

Figure 3.23 A MRI images of the degenerative subject.

Figure 3.24 A MRI image of the ACDF subject.

3.3.5 Description of 3D Mathematical Model

This mathematical model had 24 DOF. The transformation matrix between Body A and Body N followed the Euler angles Z (θ_1)–Y (θ_2)-X (θ_3) (Figure 3.25) is derived as:

$$\begin{Bmatrix} A1> \\ A2> \\ A3> \end{Bmatrix} = \begin{bmatrix} C\theta_3 & 0 & -S\theta_3 \\ 0 & 1 & 0 \\ S\theta_3 & 0 & C\theta_3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\theta_2 & S\theta_2 \\ 0 & -S\theta_2 & C\theta_2 \end{bmatrix} \begin{bmatrix} C\theta_1 & S\theta_1 & 0 \\ -S\theta_1 & C\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} N1> \\ N2> \\ N3> \end{Bmatrix}$$

$$= \begin{bmatrix} (C\theta_1 \bullet C\theta_3 - S\theta_1 \bullet S\theta_2 \bullet S\theta_3) & (S\theta_1 \bullet C\theta_3 + S\theta_2 \bullet S\theta_3 \bullet C\theta_1) & (-S\theta_3 \bullet C\theta_2) \\ (-S\theta_1 \bullet C\theta_2) & (C\theta_1 \bullet C\theta_2) & (S\theta_2) \\ (S\theta_3 \bullet C\theta_1 + S\theta_1 \bullet S\theta_2 \bullet C\theta_3) & (S\theta_1 \bullet S\theta_3 - S\theta_2 \bullet C\theta_1 \bullet C\theta_3) & (C\theta_2 \bullet C\theta_3) \end{bmatrix} \begin{Bmatrix} N1> \\ N2> \\ N3> \end{Bmatrix}$$

Following the same rules, the Euler angles between Body B and Body A are Z (θ_4)–Y (θ_5)-X (θ_6);between Body C and Body B are Z (θ_7)–Y (θ_8)-X (θ_9);between Body D and Body C are Z (θ_{10})–Y (θ_{11})-X (θ_{12});between Body E and Body D are Z (θ_{13})–Y (θ_{14})-X (θ_{15});between Body H and Body E are Z (θ_{16})–Y (θ_{17})-X (θ_{18});between Body K

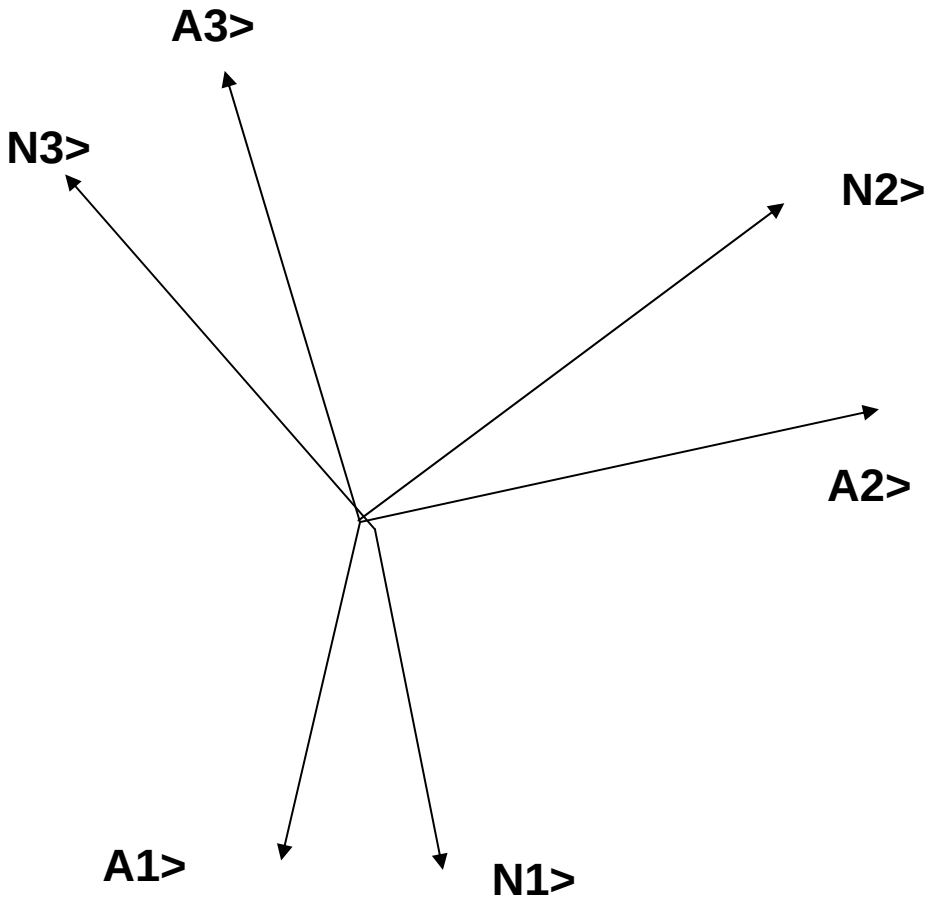


Figure 3.25 Euler translation diagram.

and Body H are Z (θ_{19})–Y (θ_{20})-X (θ_{21});between Body S and Body K are Z (θ_{22})–Y (θ_{23})-X (θ_{24}). The transformation matrix from the skull to the C7 is as the following:

$$\begin{Bmatrix} S1> \\ S2> \\ S3> \end{Bmatrix} =$$

$$\begin{bmatrix} (C\theta_{22} \bullet C\theta_{24} - S\theta_{22} \bullet S\theta_{23} \bullet S\theta_{24}) & (S\theta_{22} \bullet C\theta_{24} + S\theta_{23} \bullet S\theta_{24} \bullet C\theta_{22}) & (-S\theta_{24} \bullet C\theta_{23}) \\ (-S\theta_{22} \bullet C\theta_{24}) & (C\theta_{22} \bullet C\theta_{23}) & (S\theta_{23}) \\ (S\theta_{24} \bullet C\theta_{22} + S\theta_{22} \bullet S\theta_{23} \bullet C\theta_{24}) & (S\theta_{22} \bullet S\theta_{24} - S\theta_{23} \bullet C\theta_{22} \bullet C\theta_{24}) & (C\theta_{23} \bullet C\theta_{24}) \end{bmatrix}$$

$$\begin{bmatrix} (C\theta_{19} \bullet C\theta_{21} - S\theta_{19} \bullet S\theta_{20} \bullet S\theta_{21}) & (S\theta_{19} \bullet C\theta_{21} + S\theta_{20} \bullet S\theta_{21} \bullet C\theta_{19}) & (-S\theta_{21} \bullet C\theta_{20}) \\ (-S\theta_{19} \bullet C\theta_{20}) & (C\theta_{19} \bullet C\theta_{20}) & (S\theta_{20}) \\ (S\theta_{21} \bullet C\theta_{19} + S\theta_{19} \bullet S\theta_{20} \bullet C\theta_{21}) & (S\theta_{19} \bullet S\theta_{21} - S\theta_{20} \bullet C\theta_{19} \bullet C\theta_{21}) & (C\theta_{20} \bullet C\theta_{21}) \end{bmatrix}$$

$$\begin{bmatrix} (C\theta_{16} \bullet C\theta_{18} - S\theta_{16} \bullet S\theta_{17} \bullet S\theta_{18}) & (S\theta_{16} \bullet C\theta_{18} + S\theta_{17} \bullet S\theta_{18} \bullet C\theta_{16}) & (-S\theta_{18} \bullet C\theta_{17}) \\ (-S\theta_{16} \bullet C\theta_{17}) & (C\theta_{16} \bullet C\theta_{17}) & (S\theta_{17}) \\ (S\theta_{18} \bullet C\theta_{16} + S\theta_{16} \bullet S\theta_{17} \bullet C\theta_{18}) & (S\theta_{16} \bullet S\theta_{18} - S\theta_{17} \bullet C\theta_{16} \bullet C\theta_{18}) & (C\theta_{17} \bullet C\theta_{18}) \end{bmatrix}$$

$$\begin{bmatrix} (C\theta_{13} \bullet C\theta_{15} - S\theta_{13} \bullet S\theta_{14} \bullet S\theta_{15}) & (S\theta_{13} \bullet C\theta_{15} + S\theta_{14} \bullet S\theta_{15} \bullet C\theta_{13}) & (-S\theta_{15} \bullet C\theta_{14}) \\ (-S\theta_{13} \bullet C\theta_{14}) & (C\theta_{13} \bullet C\theta_{14}) & (S\theta_{14}) \\ (S\theta_{15} \bullet C\theta_{13} + S\theta_{13} \bullet S\theta_{14} \bullet C\theta_{15}) & (S\theta_{13} \bullet S\theta_{15} - S\theta_{14} \bullet C\theta_{13} \bullet C\theta_{15}) & (C\theta_{14} \bullet C\theta_{15}) \end{bmatrix}$$

$$\begin{bmatrix} (C\theta_{10} \bullet C\theta_{12} - S\theta_{10} \bullet S\theta_{11} \bullet S\theta_{12}) & (S\theta_{10} \bullet C\theta_{12} + S\theta_{11} \bullet S\theta_{12} \bullet C\theta_{10}) & (-S\theta_{12} \bullet C\theta_{11}) \\ (-S\theta_{10} \bullet C\theta_{11}) & (C\theta_{10} \bullet C\theta_{11}) & (S\theta_{11}) \\ (S\theta_{12} \bullet C\theta_{10} + S\theta_{10} \bullet S\theta_{11} \bullet C\theta_{12}) & (S\theta_{10} \bullet S\theta_{12} - S\theta_{11} \bullet C\theta_{10} \bullet C\theta_{12}) & (C\theta_{11} \bullet C\theta_{12}) \end{bmatrix}$$

$$\begin{bmatrix} (C\theta_{7} \bullet C\theta_{9} - S\theta_{7} \bullet S\theta_{8} \bullet S\theta_{9}) & (S\theta_{7} \bullet C\theta_{9} + S\theta_{8} \bullet S\theta_{9} \bullet C\theta_{7}) & (-S\theta_{9} \bullet C\theta_{8}) \\ (-S\theta_{7} \bullet C\theta_{8}) & (C\theta_{7} \bullet C\theta_{8}) & (S\theta_{8}) \\ (S\theta_{9} \bullet C\theta_{7} + S\theta_{7} \bullet S\theta_{8} \bullet C\theta_{9}) & (S\theta_{7} \bullet S\theta_{9} - S\theta_{8} \bullet C\theta_{7} \bullet C\theta_{9}) & (C\theta_{8} \bullet C\theta_{9}) \end{bmatrix}$$

$$\begin{bmatrix} (C\theta_{4} \bullet C\theta_{6} - S\theta_{4} \bullet S\theta_{5} \bullet S\theta_{6}) & (S\theta_{4} \bullet C\theta_{6} + S\theta_{5} \bullet S\theta_{6} \bullet C\theta_{4}) & (-S\theta_{6} \bullet C\theta_{5}) \\ (-S\theta_{4} \bullet C\theta_{5}) & (C\theta_{4} \bullet C\theta_{5}) & (S\theta_{5}) \\ (S\theta_{6} \bullet C\theta_{4} + S\theta_{4} \bullet S\theta_{5} \bullet C\theta_{6}) & (S\theta_{4} \bullet S\theta_{6} - S\theta_{5} \bullet C\theta_{4} \bullet C\theta_{6}) & (C\theta_{5} \bullet C\theta_{6}) \end{bmatrix}$$

$$\begin{bmatrix} (C\theta_{1} \bullet C\theta_{3} - S\theta_{1} \bullet S\theta_{2} \bullet S\theta_{3}) & (S\theta_{1} \bullet C\theta_{3} + S\theta_{2} \bullet S\theta_{3} \bullet C\theta_{1}) & (-S\theta_{3} \bullet C\theta_{2}) \\ (-S\theta_{1} \bullet C\theta_{2}) & (C\theta_{1} \bullet C\theta_{2}) & (S\theta_{2}) \\ (S\theta_{3} \bullet C\theta_{1} + S\theta_{1} \bullet S\theta_{2} \bullet C\theta_{3}) & (S\theta_{1} \bullet S\theta_{3} - S\theta_{2} \bullet C\theta_{1} \bullet C\theta_{3}) & (C\theta_{2} \bullet C\theta_{3}) \end{bmatrix} \begin{Bmatrix} N1> \\ N2> \\ N3> \end{Bmatrix}$$

$$= \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix} \begin{Bmatrix} N1> \\ N2> \\ N3> \end{Bmatrix}$$

$Cos = C; Sin = S$

In the 3D mathematical model, which included ligaments and muscles, F^A is the resultant external force acting on a rigid body A:

$$F^A = \sum_{i=1}^{M} F_i^{Mus} + F_{grav} + \sum_{j=1}^{L} F_j^{Lig} + F_{Comp} + F_{Ap} + F_{Lat}$$

Where,

M	the number of muscles included in the model
F_i^{Mus}	the ith muscular force
F_{grav}	the gravity force acting on the respective rigid body
L	the number of ligaments
F_j^{Lig}	the jth ligament force
F_{Comp}	the compression force on the respective rigid body
F_{Ap}	the anterior-posterior force on the respective rigid body
F_{Comp}	the lateral force on the respective rigid body

F^{A*} is the resultant inertia force acting on the rigid body A:

$$F^{A^*} = -M_A * a_A$$

Where, M_A the mass of the rigid body A

a_A the acceleration of the mass center of the rigid body A

T^A is the resultant external torque at the mass center of the rigid body A:

$$T^A = \sum_{i=1}^{M} (r_i^{Mus} \times F_i^{Mus}) + 0 \times F_{grav} + \sum_{j=1}^{L} (r_j^{Lig} \times F_j^{Lig}) + r^{Comp} \times F_{Comp} + r^{Ap} \times F_{Ap} + r^{Lat} \times F_{Lat}$$

Where, r_i^{Mus} the distance from the ith muscular force to the mass center of the rigid body A

r_j^{Lig} the distance from the jth ligament force to the mass center of the rigid body A

r^{Comp} the distance from the compression force to the mass center of the rigid body A

r^{Ap} the distance from the anterior-posterior force to the mass center of the rigid body A

r^{Lat} the distance from the lateral force to the mass center of the rigid body A

T^{A*} is the resultant inertia torque at the mass center of the rigid body A

$$T^{A^*} = (-I_A \bullet {}^N\omega^A) \times {}^N\omega^A - I_A \bullet {}^N\alpha^A$$

F_r is the constrained generalized active forces, and

$$F_r = \sum_{i=1}^{M} ({}^N\upsilon^A{}_{U_r} \bullet F^A + {}^N\omega^A{}_{U_r} \bullet T^A)$$

F_r^* is the constrained generalized inertia forces, and

$$F_r^* = \sum_{i=1}^{M} ({}^N\upsilon^A{}_{U_r} \bullet F^{A^*} + {}^N\omega^A{}_{U_r} \bullet T^{A^*})$$

Where, r the rth DOF and $r = 1,2,\ldots,n$

M the total number of rigid body in the system

U_r the rth generalized coordinate system

A one of eight rigid body, which represent eight vertebrae from C1 to C7 and the skull

N the Newtonian coordinate system

${}^N\upsilon^A{}_{U_r}$ the partial velocity of the mass center of the rigid body A, and

$${}^N\upsilon^A{}_{U_r} = \frac{\delta^N\upsilon^A}{\delta U_r}$$

${}^N\omega^A{}_{U_r}$ the partial angular velocity of the mass center of the rigid body A, and

$$^{N}\omega^{A}_{U_r} = \frac{\delta^{N}\omega^{A}}{\delta U_r}$$

According to the Kane's dynamics, assuming that the system has n DOF, the dynamical equation can be expressed as [77, 78]:

$$F_r + F_r^* = 0 \qquad r = 1,2,\ldots,n,$$

24 generalized speeds (U1, …, U24) were chosen in this 3D models.

$$^{B\to N}\vec{\omega} = {}^{A\to N}\vec{\omega} + {}^{B\to A}\vec{\omega} + U_1 * A_1 > + U_2 * A_2 > + U_3 * A_3 >$$

$$^{C\to N}\vec{\omega} = {}^{B\to N}\vec{\omega} + {}^{C\to B}\vec{\omega} + U_4 * B_1 > + U_5 * B_2 > + U_6 * B_3 >$$

$$^{D\to N}\vec{\omega} = {}^{C\to N}\vec{\omega} + {}^{D\to C}\vec{\omega} + U_7 * C_1 > + U_8 * C_2 > + U_9 * C_3 >$$

$$^{E\to N}\vec{\omega} = {}^{D\to N}\vec{\omega} + {}^{E\to D}\vec{\omega} + U_{10} * D_1 > + U_{11} * D_2 > + U_{12} * D_3 >$$

$$\vec{\upsilon}^{N}_{BA} = \vec{\upsilon}^{N}_{AB} + U_{13} * A_1 > + U_{14} * A_2 > + U_{15} * A_3 >$$

$$\vec{\upsilon}^{N}_{CB} = \vec{\upsilon}^{N}_{BC} + U_{16} * B_1 > + U_{17} * B_2 > + U_{18} * B_3 >$$

$$\vec{\upsilon}^{N}_{DC} = \vec{\upsilon}^{N}_{CD} + U_{19} * C_1 > + U_{20} * C_2 > + U_{21} * C_3 >$$

$$\vec{\upsilon}^{N}_{ED} = \vec{\upsilon}^{N}_{DE} + U_{22} * D_1 > + U_{23} * D_2 > + U_{24} * D_3 >$$

Setup $U_k = 0, \quad k = 1,\ldots,24$ to create 24 constraint equations in order to solve 24 unknown between Body A and Body E.

The Body A was assumed to be fixed. So,

P_NO_NA>=0>

P_NA_AN>=0>

Where P_NO_NA> is the position vector between the mass center NO of a Newtonian system and the contact point NA between the Newtonian system and Body A at Body N;

P_NA_AN> is the position vector between NA and the contact point AN between the Newtonian system and Body A at Body A.

At $BodyX \quad X = A,\ldots,S$, the position vectors of the mass center was derived as:

P_NO_XO>=XON1*N1>+XON2*N2>+XON3*N3>

Where XON1, XON2, XON3 are the distances between the mass center of Body X and the origin of the global coordinate.

At $BodyX \quad X = B,\ldots,K; BodyY \quad Y = A,\ldots,H; BodyZ \quad Z = C,\ldots,S$, the current level is Body X; Body Y and Z are the inferior and superior adjacent levels.

The position vectors of the contact points at Body X were derived as:

P_XO_XY>=XOXY1*X1>+XOXY2*X2>+XOXY3*X3>

P_XO_XZ>=XOXZ1*X1>+XOXZ2*X2>+XOXZ3*X3>

Where XOXY1, XOXY2, XOXY3 are the distances between the mass center of Body Y and the origin of the local coordinate of Body X.

XOXZ1, XOXZ2, XOXZ3 are the distances between the mass center of Body Z and the origin of the local coordinate of Body X.

The position vectors of the attachment points of ligament forces were derived as:

P_XO_XL>=XL1*X1>+XL2*X2>+XL3*X3>

FORCE_XYL= (MAG (P_XO_XL>-P_YO_YL>)-ORIGINAL_LENGTH)*STIFFNESS

Where XL1, XL2, XL3 are the distances between the mass center of Body X and the attachment point of a ligament. Five different types of ligaments and totally seven different ligaments were included at each vertebra from Body B to Body E;

FORCE_XYL is the ligament force calculated by multiplying the displacement and stiffness. This ligament force was acted at the attachment point XL.

The attachment points of muscular forces were added the mathematical model by following the above method.

The velocities of the mass centers were derived by using the following equation:

$$\upsilon_{XO}^{N} = \frac{dp_{XO}^{N}}{dt}$$

$$a_{XO}^{N} = \frac{d\upsilon_{XO}^{N}}{dt}$$

$$a_{XO}^{N} = \frac{d^2 p_{XO}^{N}}{dt^2}$$

Where p_{XO}^{N} is the coordinate of the mass center of Body X (X=A,…,S)
υ_{XO}^{N} is the velocity of the mass center of Body X (X=A,…,S)
a_{XO}^{N} is the acceleration of the mass center of Body X (X=A,…,S)

After known the velocity and acceleration of the mass center at each body, the velocities and accelerations of other points at the same vertebra were calculated based on the following equations:

$$\vec{\upsilon}_{XQ}^{X} = \vec{\upsilon}_{XO}^{X} + {}^{Y}\vec{\omega}^{X} \times \vec{p}_{QO}^{X}$$

$$\vec{a}_{XQ}^{X} = \vec{a}_{XO}^{X} + ({}^{Y}\vec{\omega}^{X} \times {}^{Y}\vec{\omega}^{X} \times \vec{p}_{QO}^{X}) + {}^{Y}\vec{\alpha}^{X} \times \vec{p}_{QO}^{X}$$

Where XQ is a point at Body X (X=A,…,S)
$\vec{\upsilon}_{XQ}^{X}$ is the velocity of the point XQ at Body X (X=A,…,S)

$\vec{a}^{X}_{XQ}$ is the acceleration of the point XQ at Body X (X=A,…,S)

The angular velocities and angular accelerations of Body X (X=A,…,S) were derived based on:

$$^{X}\omega^{Y} = X_1 \frac{dX_2}{dt} \bullet X_3 + X_2 \frac{dX_3}{dt} \bullet X_1 + X_3 \frac{dX_1}{dt} \bullet X_2$$

And,

$$^{X}\omega^{Y} = {}^{X}\omega^{Y1} + {}^{Y1}\omega^{Y2} + \cdots + {}^{Yn}\omega^{Y}$$

$$^{X}\alpha^{Y} = \frac{d\,{}^{X}\omega^{Y}}{dt}$$

Where, X is the reference vertebra, Y is the adjacent vertebra
X1, X2, X3 are a dextral set of perpendicular unit vectors and fixed in the rigid Body Y

For each sub-model, there are 500 input parameters to control the 3D kinematics, ligament and muscular attachments, and all these data came from CT or MRI images directly. At the same time, there are 36 outputs about compression, anteroposterior, lateral, and muscular forces from C3 to C7. For the whole mathematical model, there are 1500 input parameters and 108 outputs.

3.4 Cadaveric Error Analysis Experiment

Table 3.5 lists the major data which were recorded in the experiment and their corresponding experimental equipments or software. The experimental protocol involves two types of variables. The first type consists of experimental testing results derived

Table 3.5 Equipments and Software used in the cadaveric experiment.

EQUIPMENT Or SOFTWARE	FUNCTION
Spine Simulator	3D Motion Control at Fixed Levels
Microntracker System	3D Motion of Each Vertebral Level
Tekscan *Flexiforce*® System	Force between Selected Levels
Fluoroscope System	Fluoroscopic Video of the Entire Cervical Spine
CT & MRI	CT & MRI Scan of the Entire Cervical Spine
Arima	Rebuilding 3D CAD Models of the Entire Cervical Spine
SAAM	3D-to-2D Registration
3D Inverse Dynamic Model	3D Kinetic Data

from fluoroscopic and kinematic evaluation as well as inverse dynamics model. The second type is taken from absolute reference data recorded directly from a MicronTracker system and a custom-designed sensor system. MicronTracker system and FlexiForce® Load/Force Sensors had their own absolute errors. These errors were relatively small when compared to other available equipment and were also within an acceptable range for the biomechanics field. The 3D kinematic data derived from SAAM and 3D kinetic data calculated by using an inverse dynamic model were compared to the corresponding reference data recorded independently by a MicronTracker system and FlexiForce® Load/Force Sensors. The results of the comparison quantified how accurate the fluoroscopic and 3D-to-2D registration method and the inverse dynamic mathematical model were when they were used for the *in vivo* human experiments. The entire setup is showed as in Figure 3.26.

Figure 3.26 Experimental setup for the cadaver experiment.

3.4.1 Cadaveric Experimental Materials and Equipments

Cadaveric Cervical Spine

Two cadaveric cervical spines were harvested and used in the error analysis experiment. They were stored in the refrigerator until they were utilized for the error analysis experiment. According to the CT scan, one cadaveric cervical spine was found to have abnormal bony structures and serious fractures between C3 and C6. This spine was hence excluded from further experimentation. So, only one cadaveric cervical spine from C1 to T1 was then utilized for the following cadaver error analysis experiment (Figure 3.27). The donor of this cadaveric cervical spine was a Caucasian male about 27 years old whose height and weight were not listed. He had smoked about one packet of cigarettes per day for around ten years and drunk approximately 1/5 liter of liquor a day for around six years. This cadaveric cervical spine included the intact ligaments and facet structures, but, all of the muscles were dissected.

Figure 3.27 Cervical spine specimen tested in the error analysis experiment.

SpineSimulator

The SpineSimulator used in this experiment was located in Biomechanics Lab at VUMC (Figure 3.28). It was modified and improved from its original arrangement, which was invented and validated by Panjabi. The real-time control system was designed by using Labview™ (National Instruments Corporation, Austin, TX) in conjunction with hardware from National Instruments™ (National Instruments Corporation, Austin, TX). The machine utilized a pneumatic piston and servos motors in order to statically or continuously applies the force and moment loads while controlling their loading speeds. The load could only be applied through the superior connected part and it simulated the

Figure 3.28 Spine Simulator Machine.

normal dynamic situation in the cervical spine, to which the head provides the only load in the normal situation. Three force transducers and three speed transducers record the corresponding loads and motions respectively. By controlling the superior connected part, three types of motions (AR, FE, and LB) were analyzed over the entire spine. Due to the characteristics of the SpineSimulator, a MicronTracker system had to be introduced in order to record 3D motions of each vertebral level instead of just the top level.

MicronTracker System

The MicronTracker system included two optical pose tracking cameras and a set of black/white markers chosen from the database of Claron Technology company (www.clarontech.com). Its markers are designed with black and white color zones. The system could accurately record 3D coordinates of the points in a marker by detecting the edges between these black and white zones. In order to calculate 3D rotations and translations, three markers were attached to each vertebra in the experiment (Figure 3.29).

CT, MRI and Fluoroscope

Both CT and MRI experiments were performed on the cadaveric cervical spine at VUMC by utilizing the same CT (Philips EBW 64) and MRI (Philip Intera-Achieva 1.5T) machines used in the further human experiments (Figure 3.30). A standard 12 inch fluoroscope machine, different from the one used in the human experiment, was used in this experiment because VUMC provided different fluoroscope machines for a research

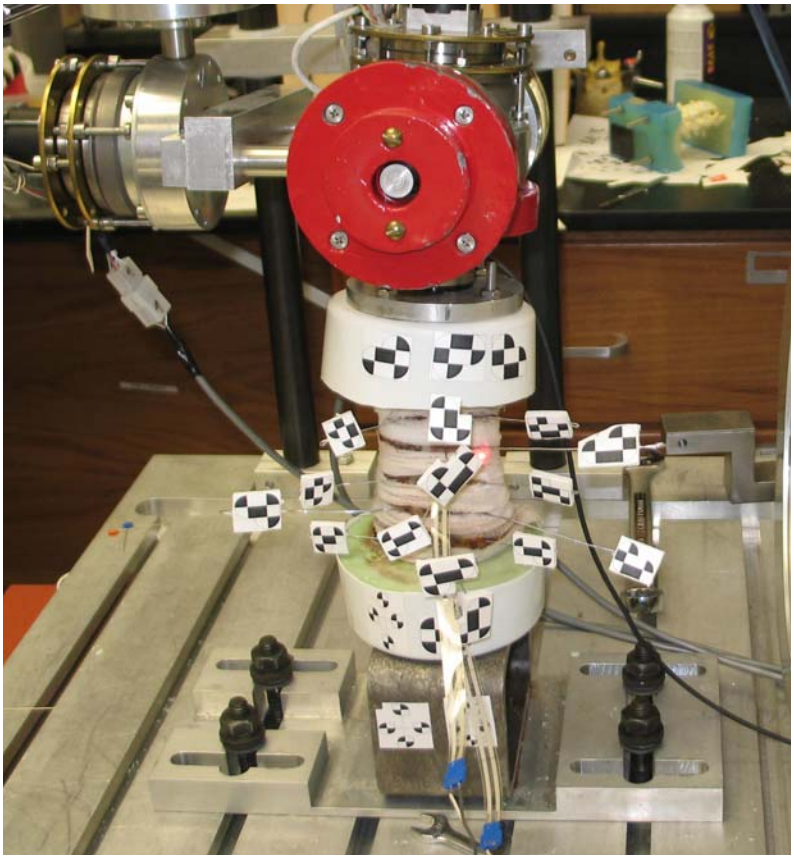


Figure 3.29 Markers used for MicronTracker system in cadaveric experiment.

Figure 3.30 CT scan for the cadaveric cervical spine.

purpose and a regular exam for a patient.

Force Sensor System

The whole system included a desktop computer, *FlexiForce*® Load/Force sensors, a National Instrument (NI) data acquisition card, a power supply, and an electrical circuit. In the electrical circuit, *FlexiForce*® Load/Force Sensors act as a resistor and have extremely high values at the unloaded situation. This electrical circuit was used to enlarge and modify the voltage output from sensors. After adding a load, the resistance of the sensor was reduced with respect to the applied load. The control system was programmed by using Labview software.

FlexiForce® Load/Force Sensor is a thin-film pressure sensor. When compared to other force sensors, it is ideal for measuring both static and dynamic forces between vertebrae without changing the natural motions and forces inside the cadaveric cervical spine. According to the manufacture's suggestions [79], *FlexiForce*® Load/Force has the following advantages over other pressure sensors for this experiment:

- High accuracy
- Linearity relationship between load and output
- Different force ranges
- Fast reaction once adding a load
- Continuous changing during applying and releasing a load
- Sensor output only related to the applied load and not related to the loading area

3.4.2 Experiment Procedures

The cadaver experiment followed the same procedures as those used in the 3D human experiment. The only difference, though, between the cadaveric experiment and the 3D human experiment was that the MicronTracker system and multi-sensor system were added in the cadaveric experiment as the absolute reference systems (Figure 3.31).

A bandage was carefully wrapped around the cadaveric cervical spine to prevent it from drying out. Water was sprayed on it after every specific time. A CT scan was performed on the Philips EBW 64-slice detector with an increment of 0.7mm with 0.33mm overlapping. A 3D CAD model of each vertebra was rebuilt up according to its CT images. All the CT images and 3D CAD models were thoroughly studied to ensure

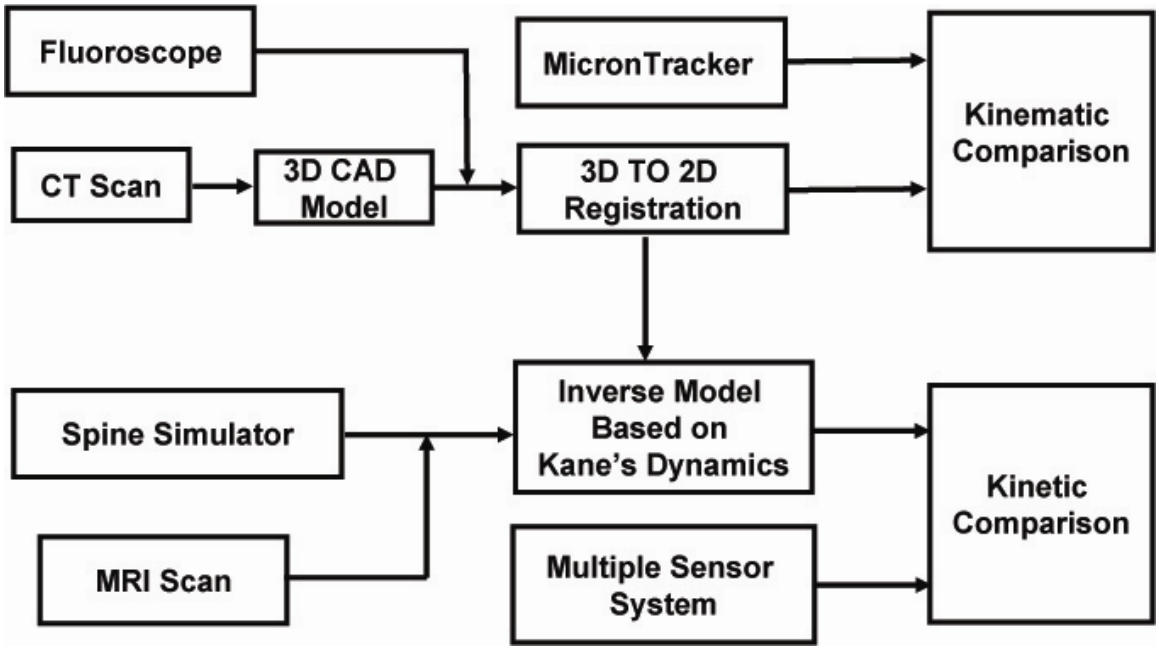


Figure 3.31 Flow chart of the experimental procedures in the cadaver experiment.

that the cadaveric cervical spine had no abnormal bony structures. C1, C2 and T1 were only the dynamics from C3 to C7 were studied. The potting material was left to solidify for more than 48 hours before the test. Then, the cadaver was fixed on the spine simulator machine. In the lab, the room temperature was kept between 70ºF and 75ºF. The cadaveric cervical spine was completely exposed under this room temperature. According to the recommendation of the manufacturer, the sensor system was conditioned and calibrated before experiment. Two FlexiForce® Sensors (range = 0 – 25 lb), were positioned between selected vertebrae in order to record real-time contact forces between vertebral bodies. Each vertebral level from C3 to C7 was attached with three markers of MicronTracker system. Their absolute 3D translation and orientation data were recorded with respect to the camera's coordinate.

A fluoroscopy unit was set up alongside the SpineSimulator. Three separate types of motions: axial rotation, flexion-extension and lateral bending were performed by controlling the motion of the superior connector (C1-C2) while under fluoroscopic surveillance. Synchronously, MicronTracker and multiple sensor systems captured the motion and force data independently. The kinematic data was obtained using the model-fitting technique and input into an inverse mathematical model based on Kane's dynamics. Finally, predicted kinematic and kinetic results were compared with the reference data.

3.4.3 Kinematic analysis method for cadaver experiment

Initially, CT images were downloaded to the server at CMR lab, the University of Tennessee, Knoxville, USA. The commercial software, Amira, was used to segment the

CT images. Anatomic knowledge was used to determine the profile of vertebras from their neighboring structures during the segmentation procedure. 3D CAD models from C1 to C7 were reconstructed from 603 CT images.

Following the same procedure as in the human experiment, 3D CAD models were registered onto 2D fluoroscopic images. Key frames were then captured for each second in order to obtain reliable results. For example, the total time elapse to perform axial rotation was 15 seconds; therefore, 15 images were captured for this activity.

Markers included in these figures were used with the MicronTracker system to help in determining the different motion patterns of the cervical spine during the prescribed activities. Through the use of the SAAM software package, which utilized the same origin and XYZ coordinate system specified in Chapter 3, the absolute 3D translations and rotations of each cadaveric vertebra were determined. A custom program was then created to calculate the 3D intersegmental rotations and translations from C3 to C7. The origin and coordinate system of 3D CAD models of the cadaveric vertebrae were defined the same as those in the human experiment.

The MicronTracker system recorded 3D coordinates of 12 points during the activities. This was done by attaching three markers, which contained four tracking points, to various locations on each vertebra. During an activity, the markers sometimes moved out of the observation view of the optical cameras, which were fixed on the ground in the MicronTracker system. As a result, the cameras might lose the track of certain points.

The 3D coordinates of each point were recorded by the MicronTracker system with respect to each camera coordinate. The origin of this coordinate system was located

in the middle of each camera. The setup of this coordinate system has been illustrated in Figure 3.32. Since the camera's origin and XYZ directions were not set up in the same way as in SAAM, an algorithm was designed to allow for the association of the results from SAAM and those from the MicronTracker system. The process has been outlined below.

Step 1: For each vertebra, three of the twelve available marker points were carefully selected. An example of the original coordinates of three points with respect to the MicronTracker camera system is depicted in Figure 3.33. Noise detected in the original data due to the vibration of MicronTracker system and SpineSimulator machine were filtered manually. The following selection criteria helped in choosing three points:

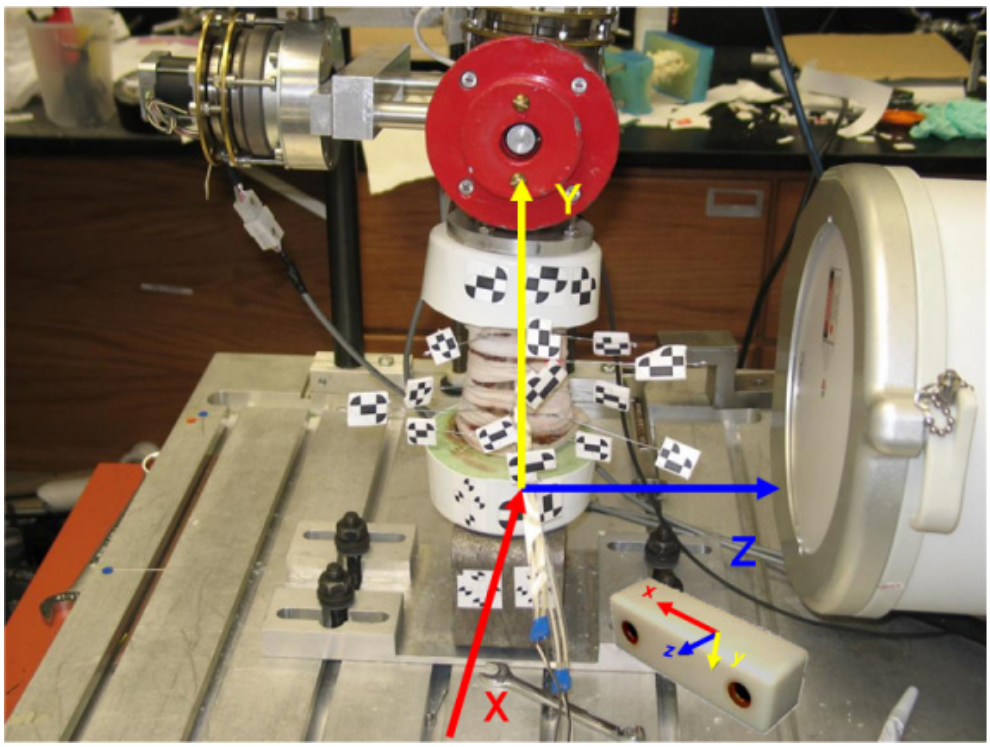


Figure 3.32 Camera coordinate system (small one, right hand coordinate) and coordinate system in SAAM (large one, left hand coordinate).

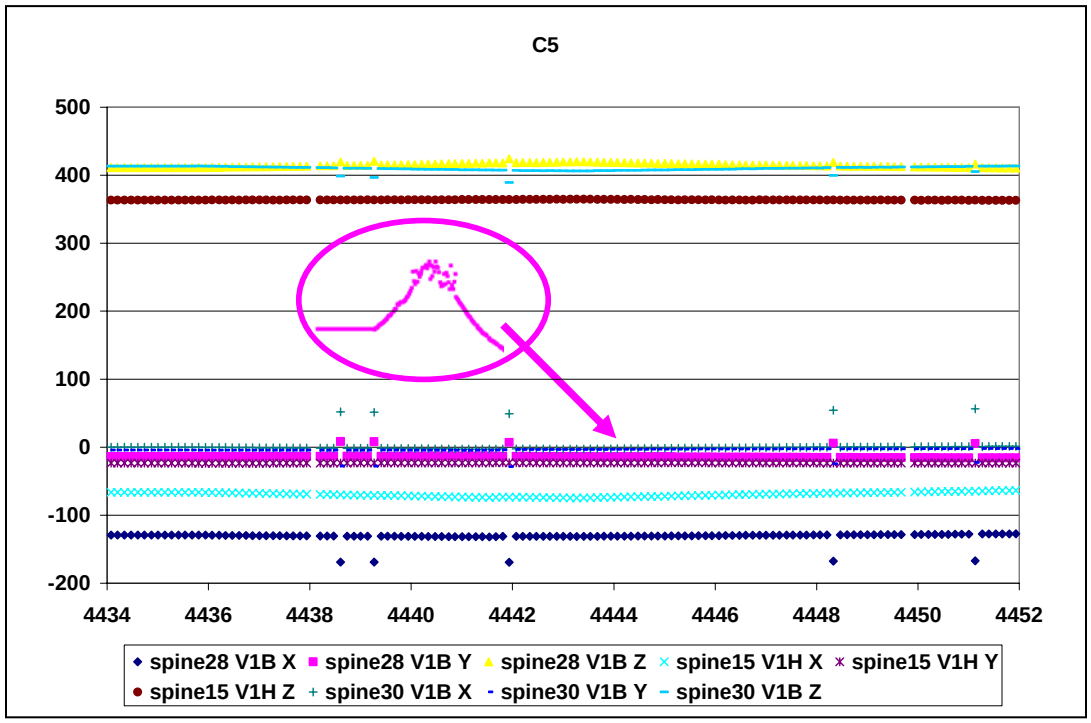


Figure 3.33 An example of the original data directly from MicronTracker system.

1) The selected points must have continuous tracking of their 3D coordinates by the cameras in the MicronTracker system during the entire activity. However, during different activities, alternate points may be chosen.
2) Three points were chosen on three different markers. This was done since the distance was larger between every two points using all three markers compared to using only one marker or two markers, and therefore, allowed for a reduction in the amount of vibration in the system.
3) Using these three points, a local coordinate system was constructed by the following procedures:

a) Two vectors $\overrightarrow{AB}$ and $\overrightarrow{AC}$ were created by connecting coordinates A and B and coordinates A and C, respectively, as shown in Figure 4.41. The direction of Z_l in the new local coordinate was then defined as:

$$Z_l = \frac{\overrightarrow{AC}}{\left|\overrightarrow{AC}\right|}$$

b) X_l is defined as:

$$X_l = \frac{\overrightarrow{AB} \times Z_l}{\left|\overrightarrow{AB} \times Z_l\right|}$$

c) Y_l is defined as:

$$Y_l = \frac{Z_l \times X_l}{\left|Z_l \times X_l\right|}$$

Following the same procedures listed above for each vertebra tested, the local coordinate system was created. The 4x4 transformation matrix was then calculated at each sequential position in order to describe the vertebra's 3D rotation and translation with respect to the coordinate system of the MicronTracker system.

Step 2: A reference marker was attached to the bottom of the tested cadaveric cervical spine, which was fixed to the SpineSimulator machine. Its local coordinate system was created by using the above method. Since this coordinate system did not move during any activity, it was treated as a global coordinate system having the same XYZ directions as those in SAAM.

Step 3: All transformation matrices calculated in Step 1 were transferred to new transformation matrices according to the reference coordinate system created in step 2.

Step 4: Relative 3D translations and rotations between adjacent vertebrae were calculated using their respective transformation matrix.

Step 5: If an alternate set of three points were chosen in step 1, 3D translational data may have produced different final results. In order to eliminate this difference, 3D relative translations and rotations were normalized with respect to their original value (Figure 3.34).

As mentioned before, fluoroscopic images were captured for each second. As an example, axial rotation took about 15 seconds, which allowed for 15 images to be captured and overlaid using SAAM. In contrast, the measurement rate of the MicronTracker system was 15~30HZ. In order to compare the result from the MicronTracker system with those from SAAM, a key position of the results from the MicronTracker system was extracted at every second using TableCurve 2D (SYSTAT Software Inc, CA) (Figure 3.35). This allowed for the comparison of the 3D relative kinematics at the C3-C4, C4-C5, C5-C6 and C6-C7 levels for every activity. In total, 60 individual positions were compared for the entire cadaveric cervical spine for the axial rotation activity.

3D kinematic results calculated from C3 to C7 using the MicronTracker and SAAM were separately compared for the flexion-extension, axial rotation, and lateral bending activities. A Box-and-Whisker plot was used to perform an analysis checking for outliers. Influential outliers were excluded from further analysis. Figure 3.36 shows an example of these comparison results.

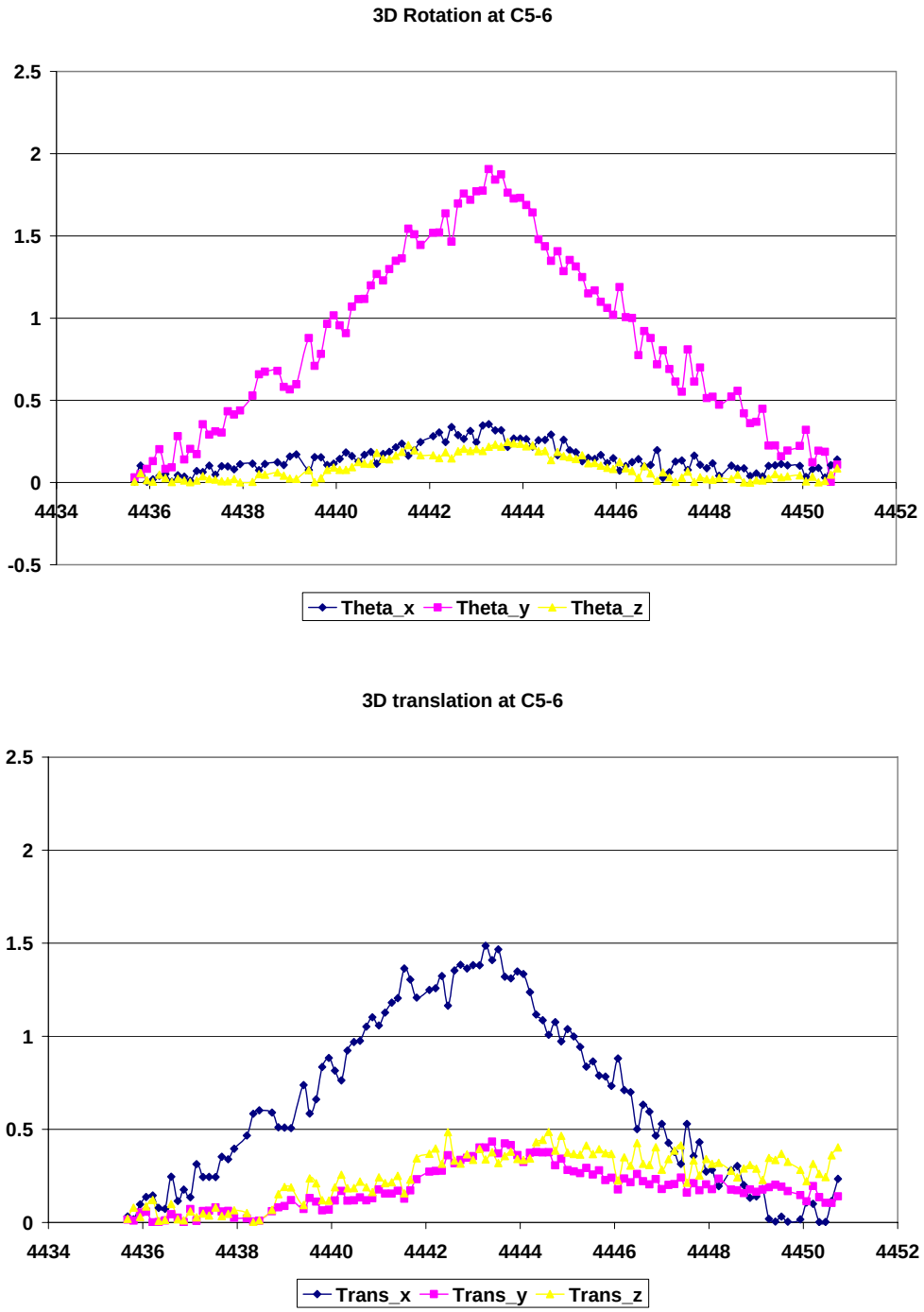
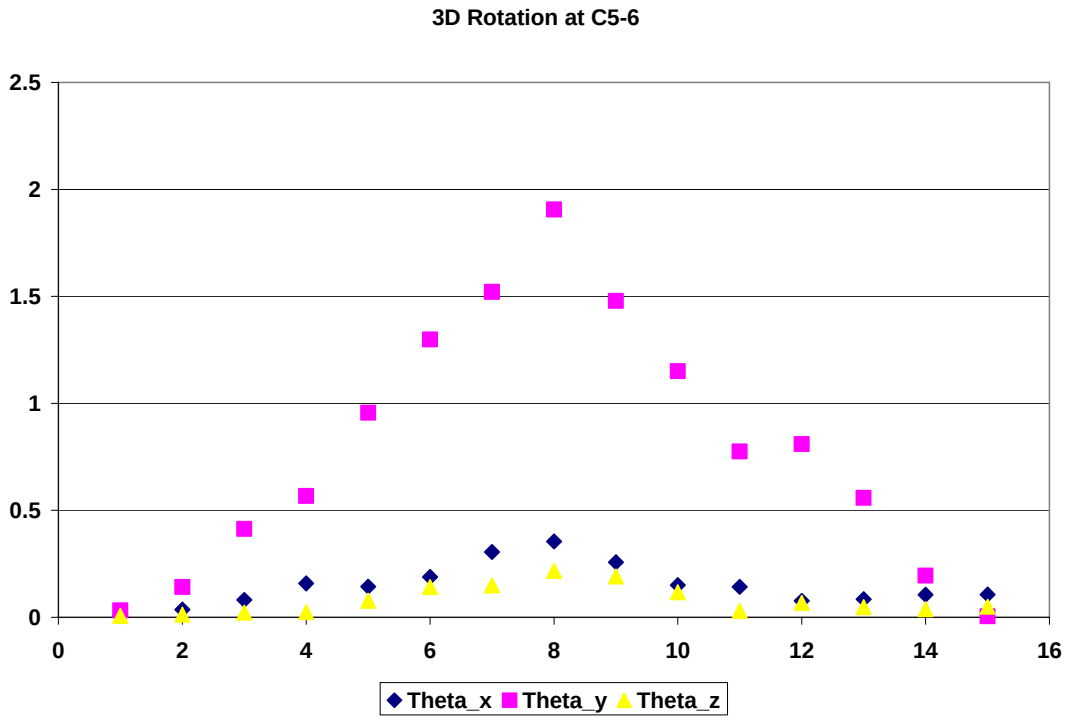


Figure 3.34 An example of 3D rotations (unit: degrees) and translations (unit: mm) calculated from MicronTracker system.



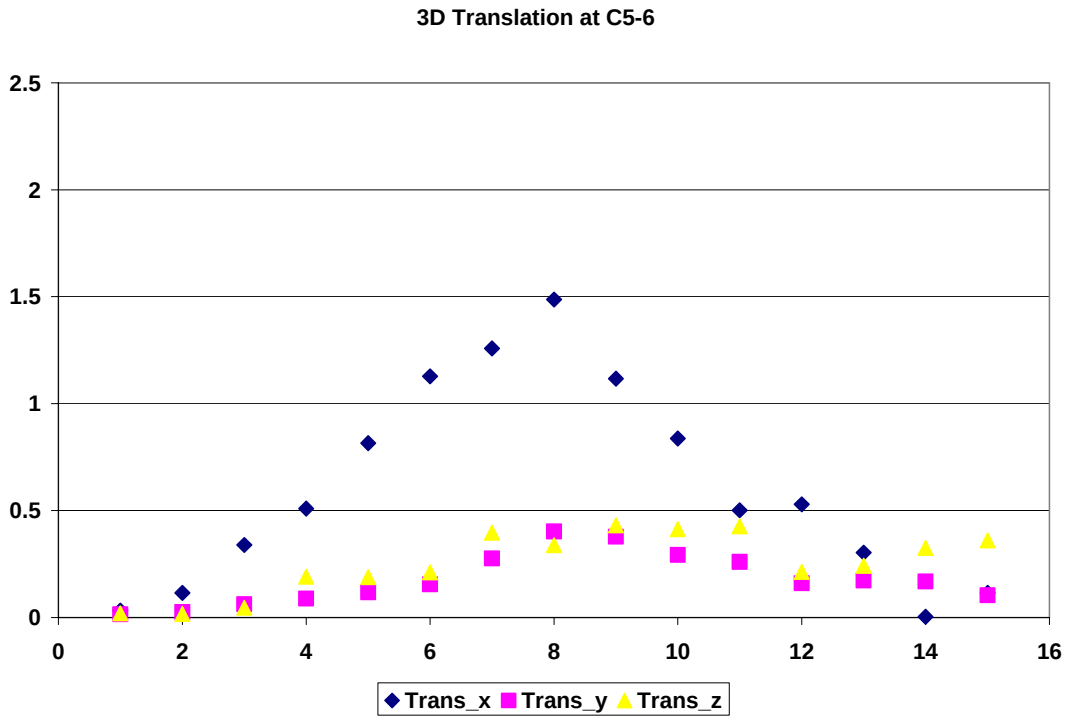


Figure 3.35 An example of 3D rotations (unit: degrees) and translations (unit: mm) captured from TableCurve 2D.

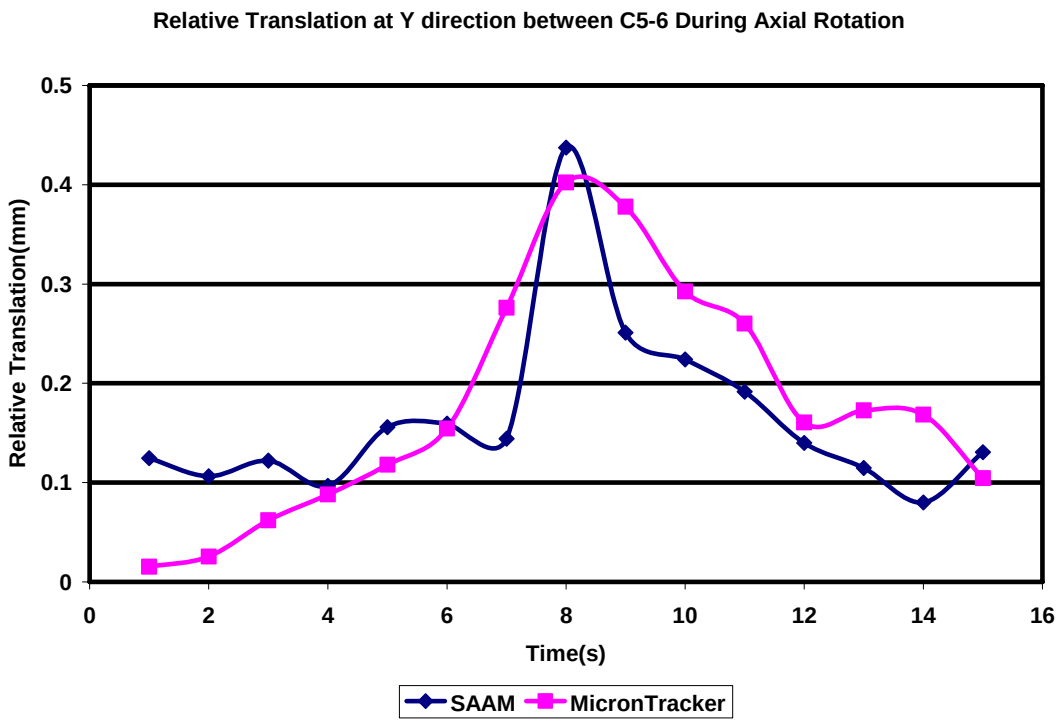


Figure 3.36 An example of data comparison for rotations and translations between SAAM and Micron Marker System.

3.4.4 Kinetic analysis method for cadaver experiment

Following the same method mentioned above, an inverse dynamic model of the cadaveric cervical spine was constructed. The weight of the cadaveric cervical spine was first measured and input into the model. Anthropometric data was also obtained by directly measuring 3D CAD models of this cadaveric spine using commercial CAD software. In the experiment, two sensors were inserted into two vertebral bodies at different levels. The real-time voltage readings from the sensors were plotted in Figure 3.37. These voltage values from the sensors were translated into force data using the linear conductance curve. The predicted forces extracted from the inverse dynamic model were directly compared to the reading from one sensor. *FlexiForce*® Load/Force sensors could not give the value of shear forces. Therefore, only the contact force in the vertical direction was compared with the predicted compression forces from the inverse dynamic model during the flexion-extension activity.

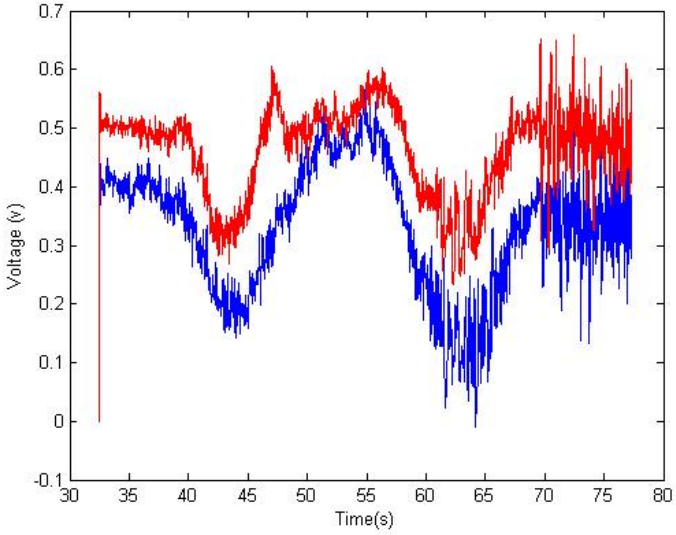


Figure 3.37 Sensor outputs in the cadaveric cervical spine experiment.

Chapter 4 Results

The relevant results of 2D and 3D *in vivo* kinematics and kinetics of normal, pathological and postoperative cervical spines are presented in this chapter. In addition, the findings from a comprehensive error analysis were included and discussed.

4.1. 2D Kinematics

The rotation of the entire cervical spine was defined as the angle between C1 relative to C7 (θ1- θ7 in Figure 3.2), which described only the motion of the cervical spine and excluded movements of the skull or trunk. The average ROM of the cervical spine was 80.6±6.4° for the normal group, 47.5±10.6° for the degenerative, 46.5±14.6° for the ACDF group, and 76.7±17.5° for the CADR group. The average intersegmental ROM of the cervical spine in the four different groups have been listed in Figure 4.1 and compared with those results that have been previously reported in the literature. During the entire flexion-extension activity, the average intersegmental ROM in the ACDF group was less than those in the normal and CADR groups at the adjacent levels. In order to better associate the intersegmental ROM and forces of each segment with respect to the rotation of the neck, the rotation of C1 (θ1 in Figure 3.2) was defined as the neck rotation. During the neck rotation from 20º of flexion to 15º of extension, the average ROM in the ACDF group at the adjacent C6-C7 and C4-C5 levels were 13.4º and 8.8º, respectively. In comparison, the values at the C6-C7 and C4-C5 levels were 3.7º and 4.8º in the normal cervical spine; 5.7º and 1.4º in the degenerative cervical spine; and 5.8º and 3.2º in the CADR cervical spine, respectively (Figure 4.2 & Figure 4.3). The difference

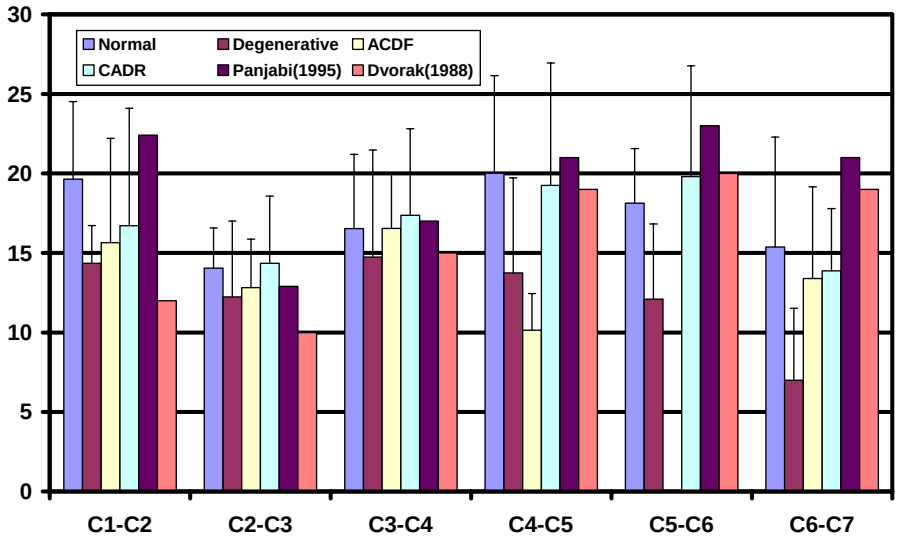


Figure 4.1 Average intersegmental ROM in four different cervical spines.

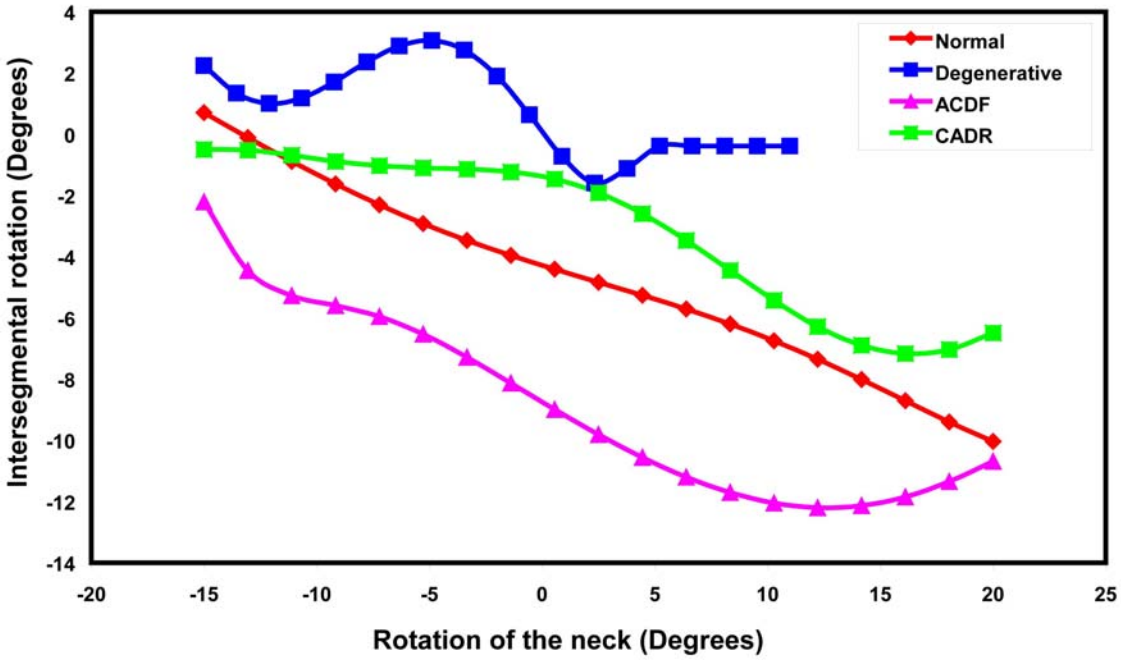


Figure 4.2 Comparison of the average intersegmental rotations from 20º flexion to 15 º extension at the superior adjacent level C4-C5 of the different cervical spines.

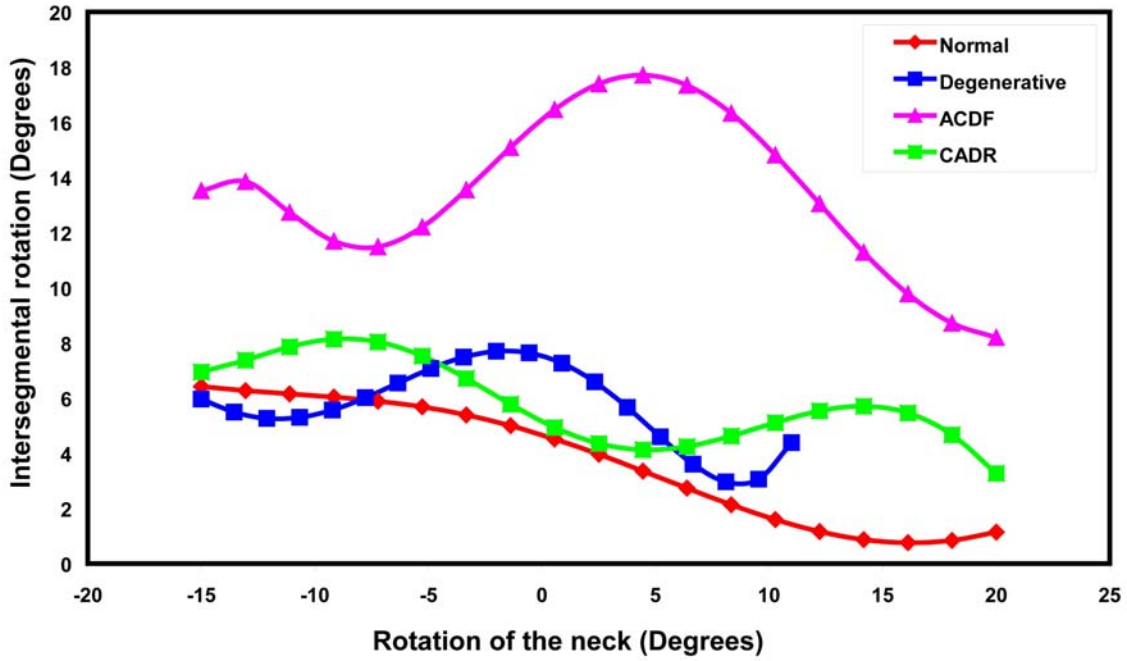


Figure 4.3 Comparison of the average intersegmental rotations from 20º flexion to 15 º extension at the inferior adjacent level C6-C7 of the different cervical spines.

between these four groups, at the C3-C4 level, was only about 1º on average (Figure 4.4).

4.2. 2D Kinetics

Both the maximum and average contact and pseudo-muscular forces of the normal, degenerative, ACDF, and CADR groups have been listed in Table 4.1. At the C4-C5 level, the anteroposterior forces ranged from 0.02 to 0.35 times skull weight (SW) in the normal group and from 0.01 to 0.94 SW in the ACDF group (Figure 4.5). The forces at the C6-C7 level in the normal group ranged from 0.02 to 0.62 SW and from 0.02 to 2.07 SW in the ACDF group. The compression forces at the C4-C5 and C6-C7 levels

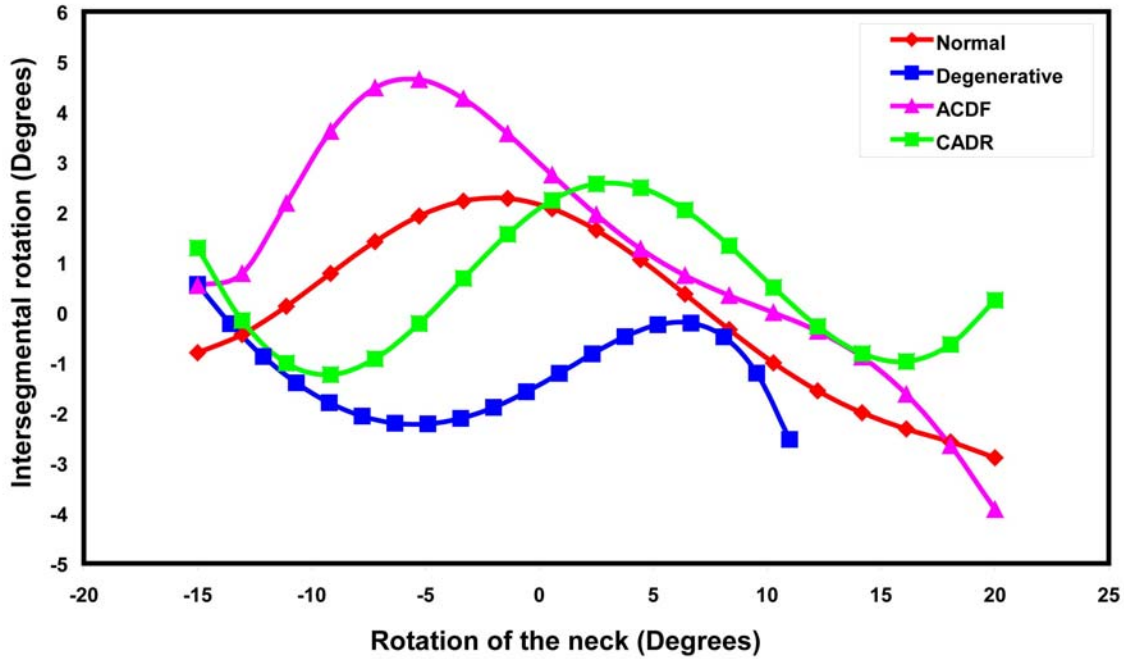


Figure 4.4 Comparison of the average intersegmental rotations from 20º flexion to 15 º extension at the level C3-C4 of the different cervical spines.

Table 4.1 Kinetic data for normal, degenerative, ACDF, and CADR groups.

		Normal		Degenerative		ACDF		CADR	
		Max	Ave	Max	Ave	Max	Ave	Max	Ave
Anteroposterior Forces	C6-C7	0.22	0.09	0.06	0.02	2.07	0.31	0.13	0.07
	C4-C5	0.35	0.13	0.11	0.05	0.94	0.15	0.36	0.10
	C3-C4	0.63	0.43	0.61	0.43	0.49	0.29	0.64	0.40
Compression Forces	C6-C7	3.09	1.97	2.24	1.79	2.63	2.03	2.79	1.84
	C4-C5	3.29	1.78	2.02	1.48	2.54	1.80	2.71	1.59
	C3-C4	2.93	1.72	2.07	1.56	2.63	1.95	2.38	1.58
Pseudo-muscular Forces	C6-C7	0.47	0.24	0.18	0.05	3.31	0.54	0.30	0.18
	C4-C5	0.97	0.51	0.62	0.50	2.1	0.63	0.96	0.50
	C3-C4	1.45	0.56	0.52	0.23	1.12	0.43	0.85	0.39

Units: Times of Skull Weight (SW).

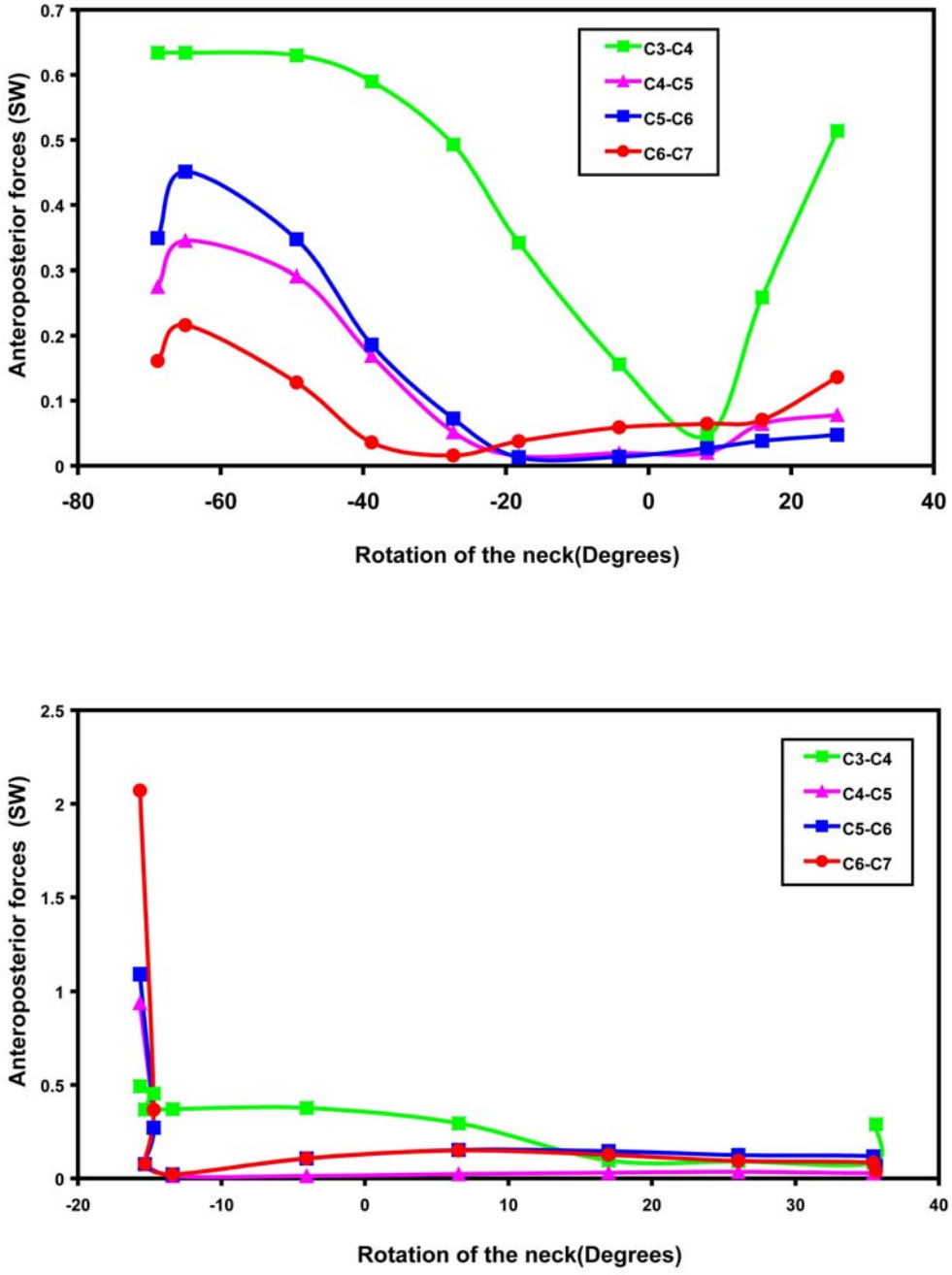


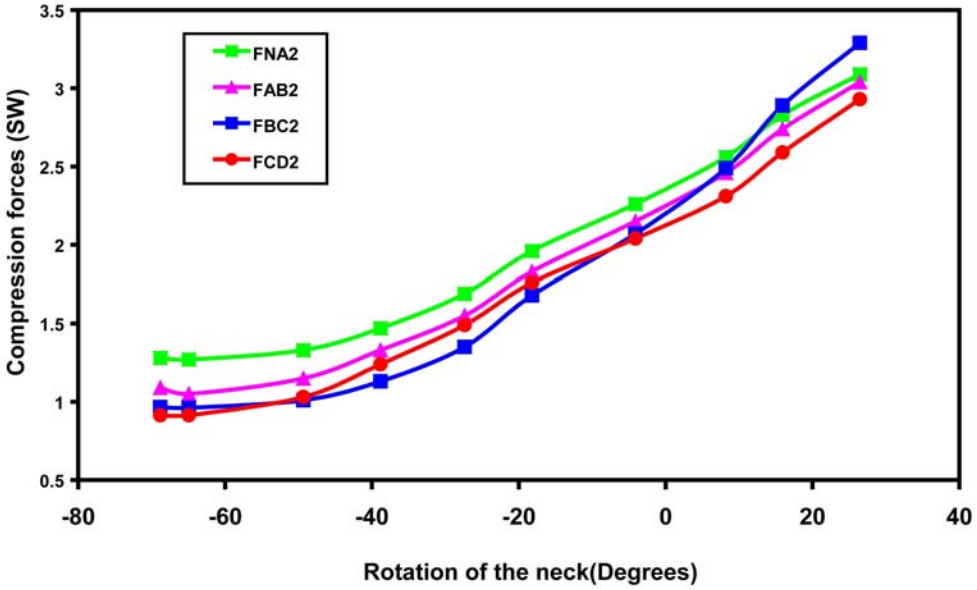
Figure 4.5 Anteroposterior force comparison between the normal and ACDF groups during the flexion-extension activity (upper: Normal; bottom: ACDF).

ranged from 0.90 to 3.29 SW and 1.40 to 3.09 SW, respectively, in the normal group and from 1.20 to 2.54 SW and 1.60 to 2.63 SW, respectively, in the ACDF group (Figure 4.6). The pseudo-muscular forces in the normal group ranged from 0 to 1.45 SW and from 0 to 3.31 SW in the ACDF group (Figure 4.7).

The normal, degenerative, and CADR groups had a smoother curve shape compared to the ACDF group, which had sudden changes at the beginning and end of the activity (Figure 4.5 & Figure 4.7). The results of the CADR were very similar to those of the normal group. The degenerative group had similar curve shapes compared to the normal, but had smaller magnitudes.

4.3. 3D kinematics

3D *in vivo* kinematics was determined for three different motions (FE, AR, and LB) by overlaying 3D CAD models onto 2D sequential fluoroscopic images. For AR and LB, the overall motion was defined as the change in specific rotation for only one side of the vertebral body, which was from the maximum side to the neutral position in the plane along which the motion primarily took place. Coupled rotations were defined as the rotations which did not take place in the main motion plane of interest. Thus for the AR activity, the main rotation was in the horizontal plane and the coupled rotations were in the frontal plane (LB) and sagittal plane (FE). Similarly, for the LB activity, main rotation occurred in the frontal plane along with coupled rotations in the horizontal plane (AR) and sagittal plane (FE). And all of the translations were defined as coupled translations pertaining only for the main rotation of interest, and then divided into AP, SI,



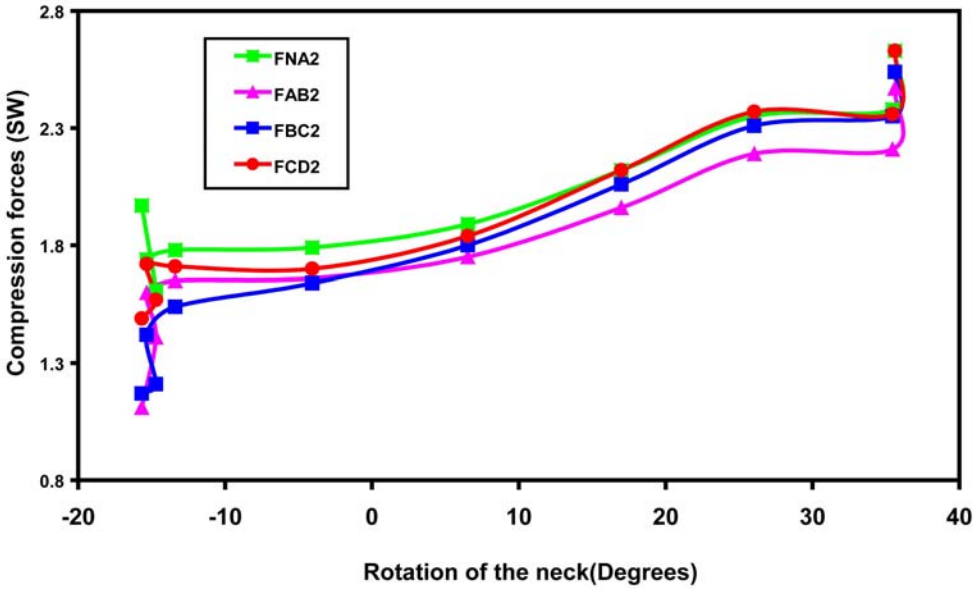
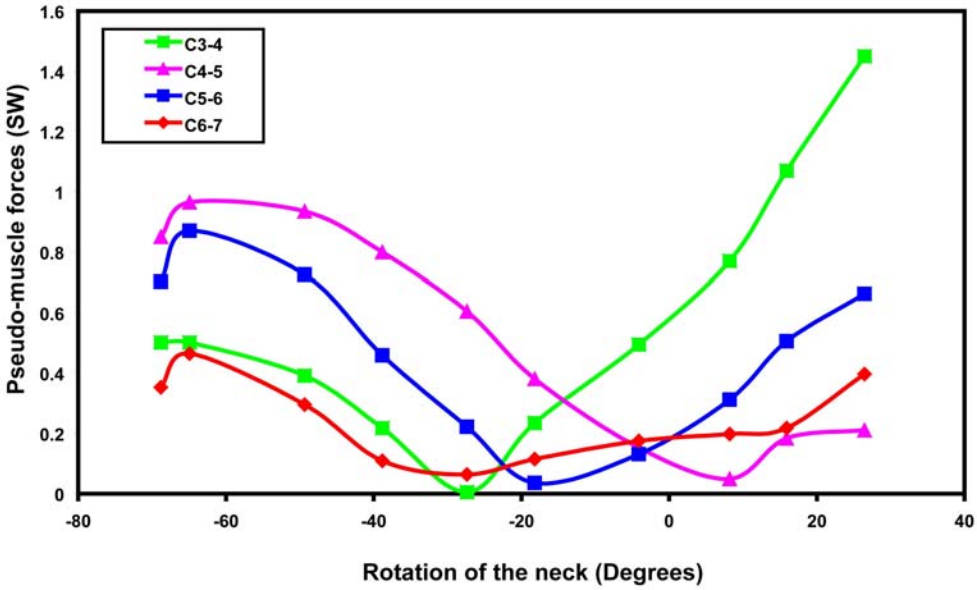


Figure 4.6 Compression force comparison between normal and ACDF during flexion-extension (upper: Normal; bottom: ACDF).



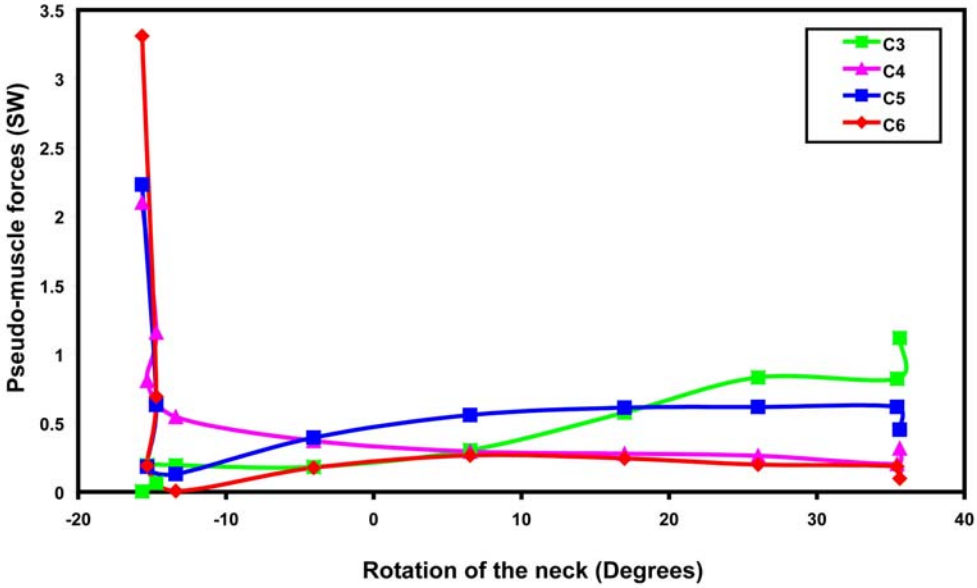


Figure 4.7 Pseudo-muscular force comparison between normal and ACDF during flexion-extension (upper: Normal; bottom: ACDF).

and LT directions. For FE, a main ROM was defined as the rotations from the full extension to the full flexion.

4.3.1 Flexion-extension

Three subjects, each representative of a different group, were found to have a similar ROM of their entire cervical spine: 95.0º in the normal group, 88.3º in the degenerative group, and 95.2º in the ACDF group, respectively. 3D *in vivo* intersegmental rotations and translations were determined by registering 3D CAD models of the cervical spine onto 2D sequential fluoroscopic images (Figure 4.8 through Figure 4.10). Position 1 was the full extension position (starting position); position 2 and position 3 represented 33% and 66% of the experimental time of the entire motion, respectively; and position 4 occurred at the full flexion position (end position). All the intersegmental rotations and translations were normalized with respect to the original position (full extension).

The intersegmental ROM of the FE activity for these three subjects were plotted and compared to previous studies in Figure 4.11. The intersegmental rotations of the C1-C2, C2-C3, and C3-C4 levels had no significant difference among these three subjects (Figure 4.11). In general, the degenerative subject had a relatively smaller ROM at every level, except at the C4-C5 level. This might be due to the degeneration that was present for this subject. Since the intersegmental ROM was decreased at the symptomatic level, the C4-C5 level might have required a larger ROM in order for the degenerative subject to reach the similar ROM of the entire cervical spine as in the other groups. There were

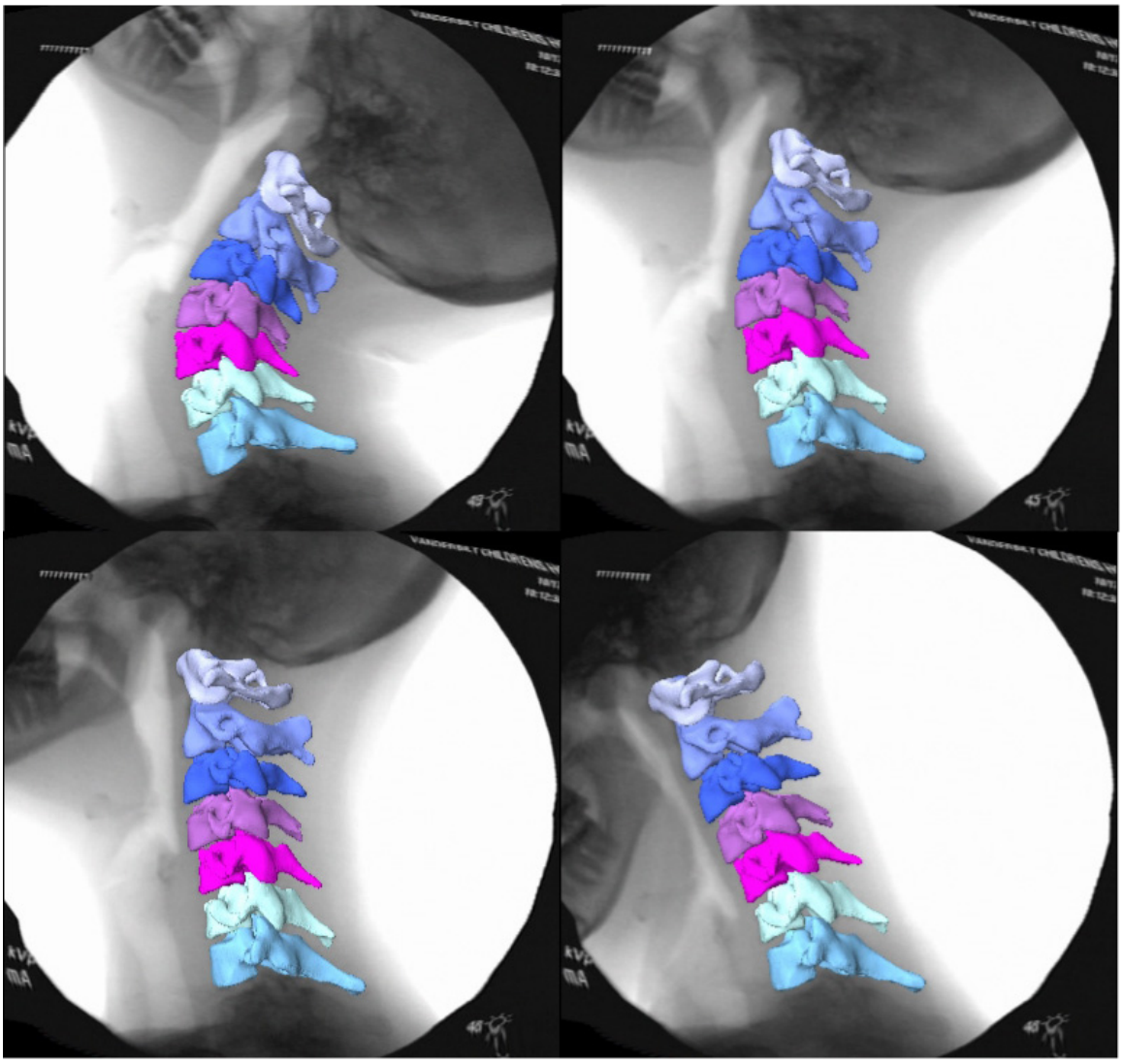


Figure 4.8 Flexion-extension activity of the entire cervical spine for the normal subject after registering 3D models onto 2D fluoroscopic images (from top-left to right-bottom: Full Extension, Position 2, Position 3 and Full Flexion).

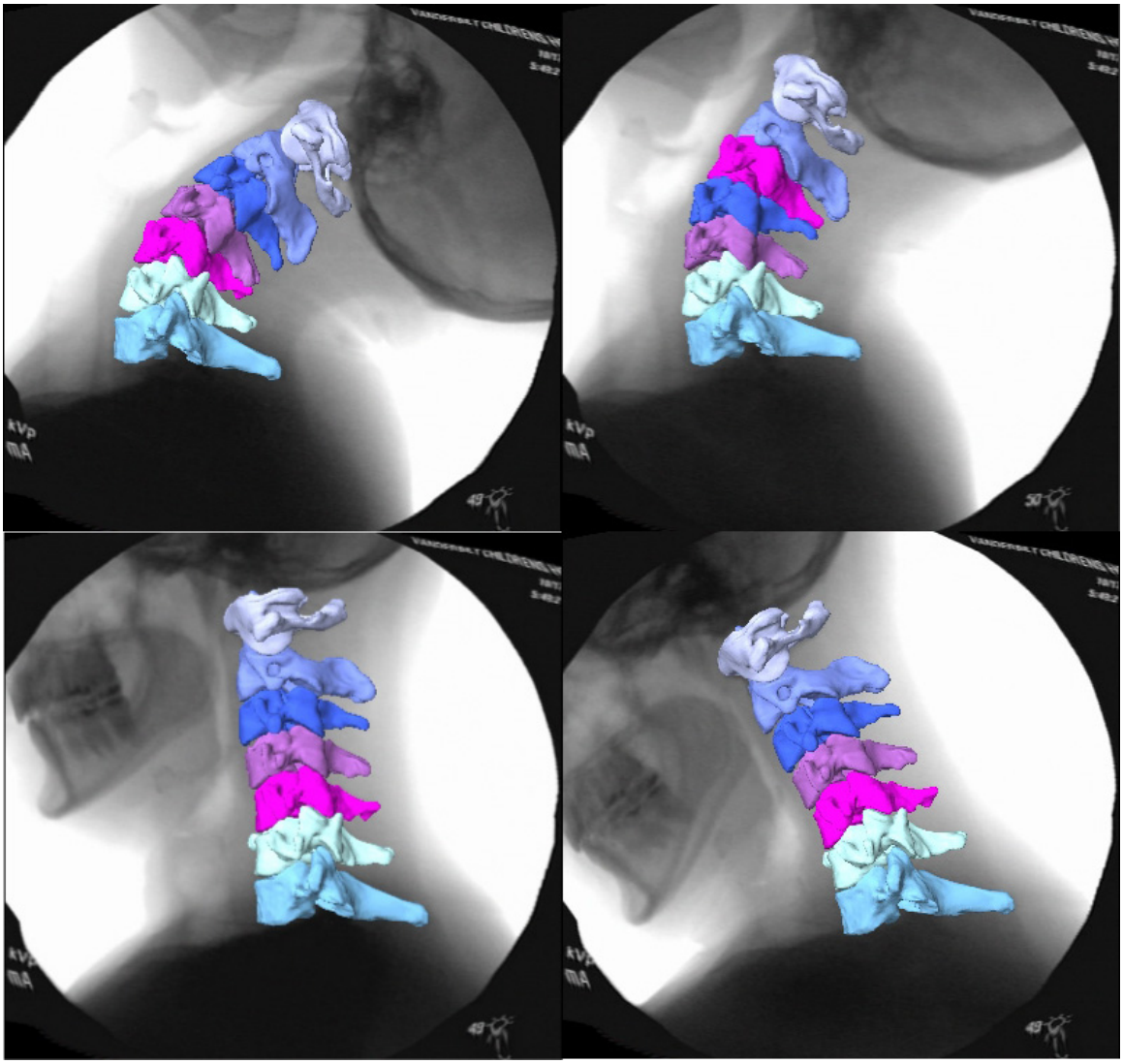


Figure 4.9 Flexion-extension activity of the entire cervical spine for the degenerative subject after registering 3D models onto 2D fluoroscopic images (from top-left to right-bottom: Full Extension, Position 2, Position 3 and Full Flexion).

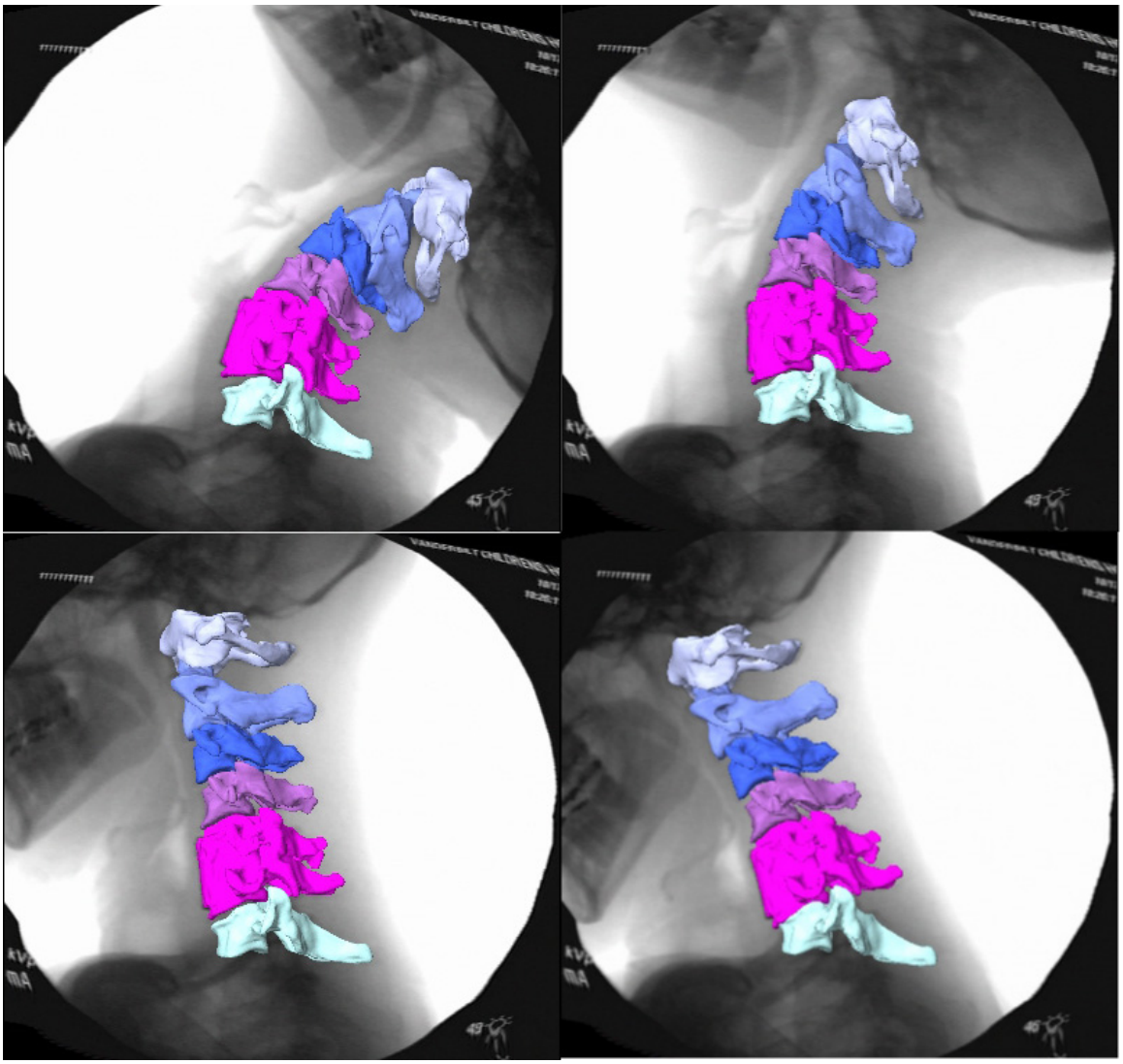


Figure 4.10 Flexion-extension activity of the entire cervical spine for the ACDF subject after registering 3D models onto 2D fluoroscopic images (from top-left to right-bottom: Full Extension, Position 2, Position 3 and Full Flexion).

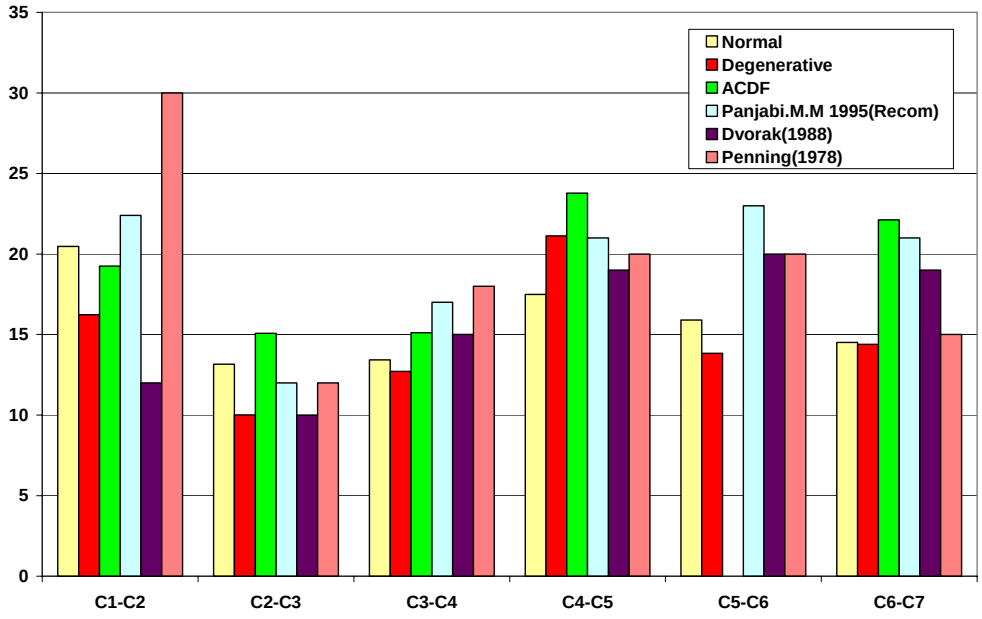


Figure 4.11 Intersegmental ROM between vertebrae during the flexion-extension activity for the normal, degenerative, and ACDF subjects and compared to previous literature (Unit: degrees).

abnormally large ROMs at adjacent levels in the ACDF subject as compared with the normal subject. For the ACDF subject, there was up to 6.3º (36.02%) more flexion-extension rotation at the superior level (C4-C5) and up to 7.6º (52.45%) more flexion-extension rotation at the inferior level (C6-C7).

During the flexion-extension activity, the magnitudes of the coupled motions pertaining to lateral bending (LB) and axial rotation (AR) between vertebrae were all less than 1º in all three subjects, except for the motion occurring at the C6-C7 level for the ACDF group. Compared to the original position of the ACDF spine, there was 0.6º LB rotation (5.77% of FE rotation) and 1.2º AR rotation (11.54% of FE rotation), with 10.4º

flexion-extension rotation, at position 2. At position 3, there was 0.3º LB rotation (1.76% of FE rotation) and 1.5º AR rotation (8.82% of FE rotation), with 17.0º flexion-extension rotation. At full flexion position, there was 1.9º LB rotation (8.60% of FE rotation) and 7.9º AR rotation (35.75% of FE rotation), with 22.1º flexion-extension rotation.

Figure 4.12 was plotted to detail the intersegmental flexion-extension rotations of the cervical spine for the normal, degenerative, and ACDF subjects at four different positions (Full Extension, Position 2, Position 3, and Full Flexion). Overall, all three groups had similar motion patterns. Further investigation of position 2 showed that the intersegmental FE at the C5-C6 level in the degenerative subject was larger than that in the normal subject, and also the intersegmental FE at adjacent levels in the degenerative subject was relatively smaller than those in the normal and ACDF subjects (Figure 4.13). However, at position 3, even though the degenerative subject still had a larger intersegmental motion when compared to the normal subject at the C5-C6 level, the intersegmental FE at the C4-C5 level in the degenerative subject was measurably increased and greater than that in the normal subject (Figure 4.14). Also at position 3, the ACDF subject experienced an increased amount of FE when compared to the normal subject at both adjacent levels (C4-C5 and C6-C7). Finally, at the full flexion position, both the degenerative and ACDF subject had larger motions at the C4-C5 level when compared to the normal subject (Figure 4.15). Nevertheless, the degenerative subject had a smaller FE at the C5-C6 level as compared with the normal. The ACDF subject still had a larger FE as compared with the normal at the adjacent C6-C7 level.

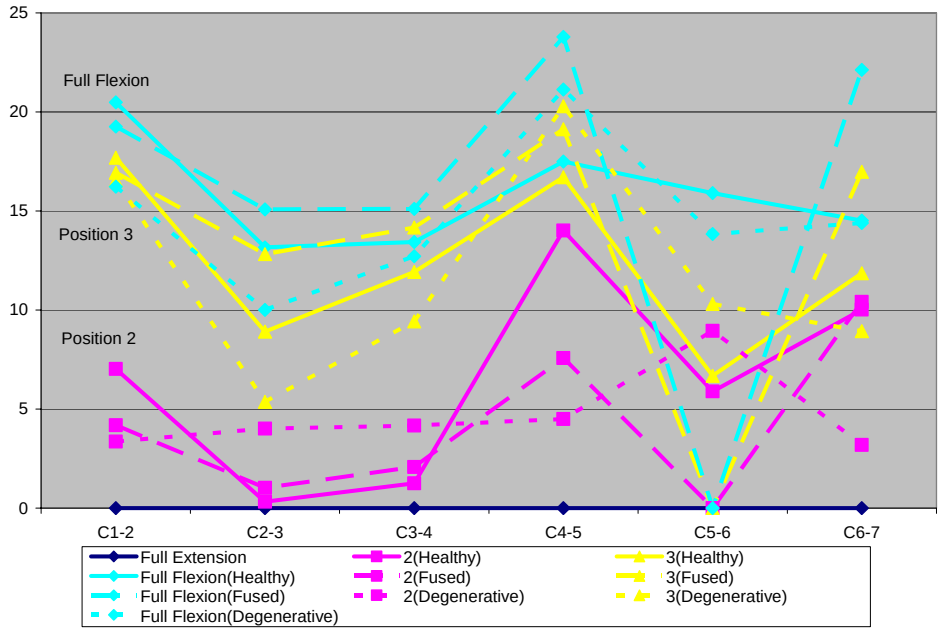


Figure 4.12 Flexion-extension activity of the entire cervical spine for the normal, degenerative and ACDF subjects after registering 3D models onto 2D fluoroscopic images (Full Extension-Original Position, Position 2, Position 3 and Full Flexion) (Unit: degrees).

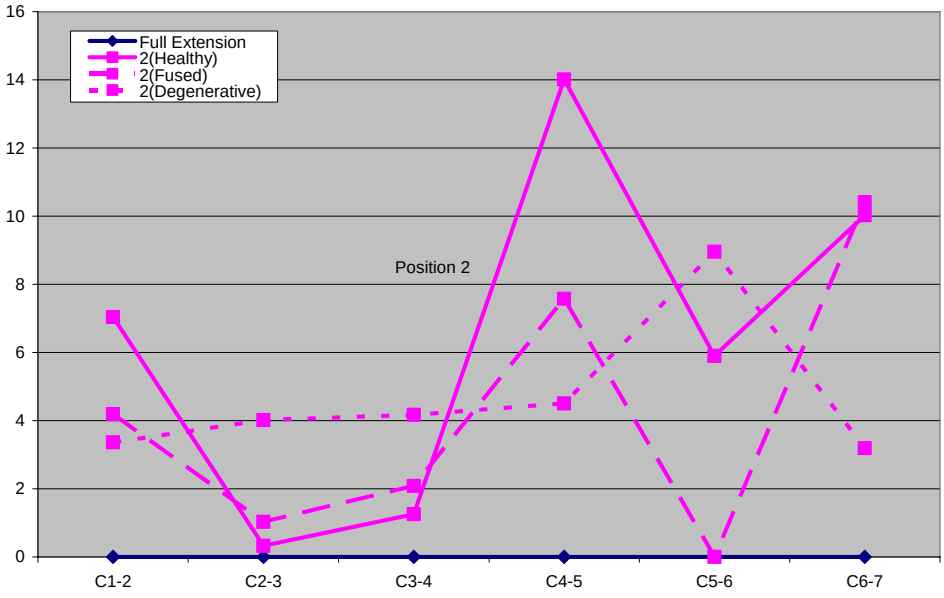


Figure 4.13 Flexion-extension activity of the entire cervical spine for the normal, degenerative and ACDF subjects after registering 3D models onto 2D fluoroscopic images (Full Extension-Original Position and Position 2) (Unit: degrees).

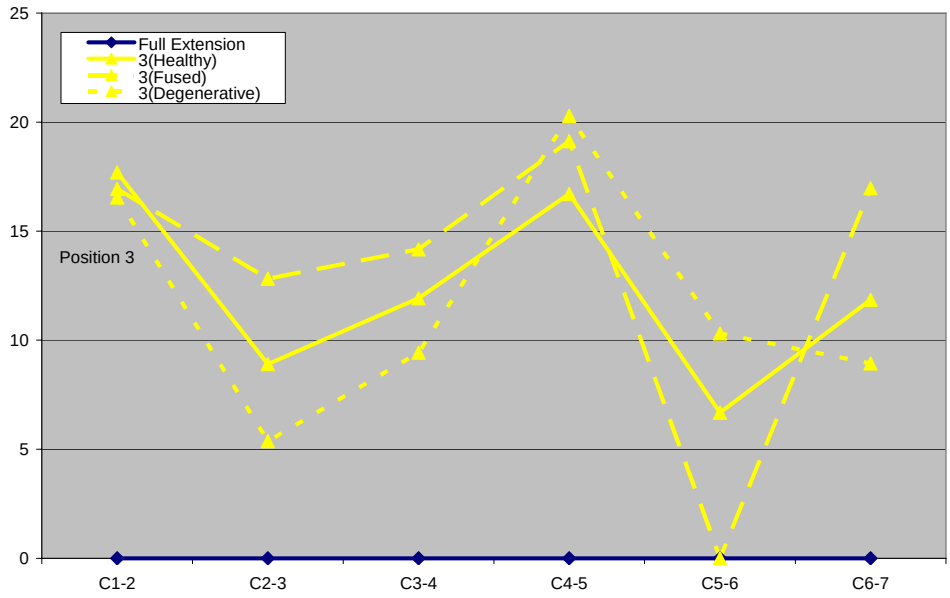


Figure 4.14 Flexion-extension activity of the entire cervical spine for the normal, degenerative and ACDF subjects after registering 3D models onto 2D fluoroscopic images (Full Extension-Original Position and Position 3) (Unit: degrees).

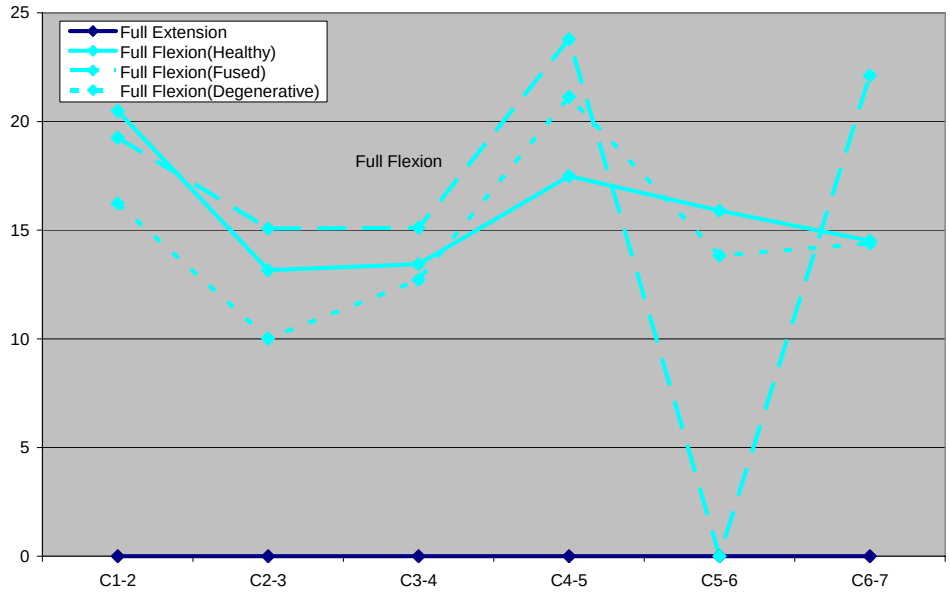


Figure 4.15 Flexion-extension activity of the entire cervical spine for the normal, degenerative and ACDF subjects after registering 3D models onto 2D fluoroscopic images (Full Extension-Original Position and Full Flexion) (Unit: degrees).

The average intersegmental translations of the entire cervical spine for all three groups have been listed in Table 4.2, and the intersegmental translations in the AP, SI and LT directions for all three subjects have been listed in Figure 4.16.

The maximum amount of intersegmental translation for all three subjects occurred at the C2-C3 level. At this level, the averages of all these three subjects were 25.7 mm in the AP directions and 29.8mm in the SI directions, respectively. Overall, despite the dissimilarities in the spine conditions, there were no significant differences in the translations between the degenerative and normal subjects. However, we did see larger intersegmental translations in both the AP and SI directions at the C4-C5 and C6-C7 levels for the ACDF subject. The intersegmental translations in the LT direction were very small for all three groups except at the C6-C7 level for the ACDF subject.

Intersegmental translations in the AP, SI and LT directions of the cervical spine for the normal, degenerative, and ACDF subjects at full extension, position 2, position 3, and full flexion were plotted in Figure 4.17- Figure 4.19. The normal and degenerative groups had similar motion patterns, but the ACDF subject experienced a different motion pattern at the C2-C3 level.

Table 4.2 Average translation of the normal, degenerative and ACDF subjects during flexion-extension activity (Unit: mm).

	C1-C2	C2-C3	C3-C4	C4-C5	C5-C6*	C6-C7
AP	18.20	25.67	23.48	19.62	11.33	8.59
SI	10.70	29.77	9.32	5.36	1.77	3.51
LT	0.15	0.26	0.22	0.09	0.10	0.67

Units: mm.
* Not include the ACDF subject.

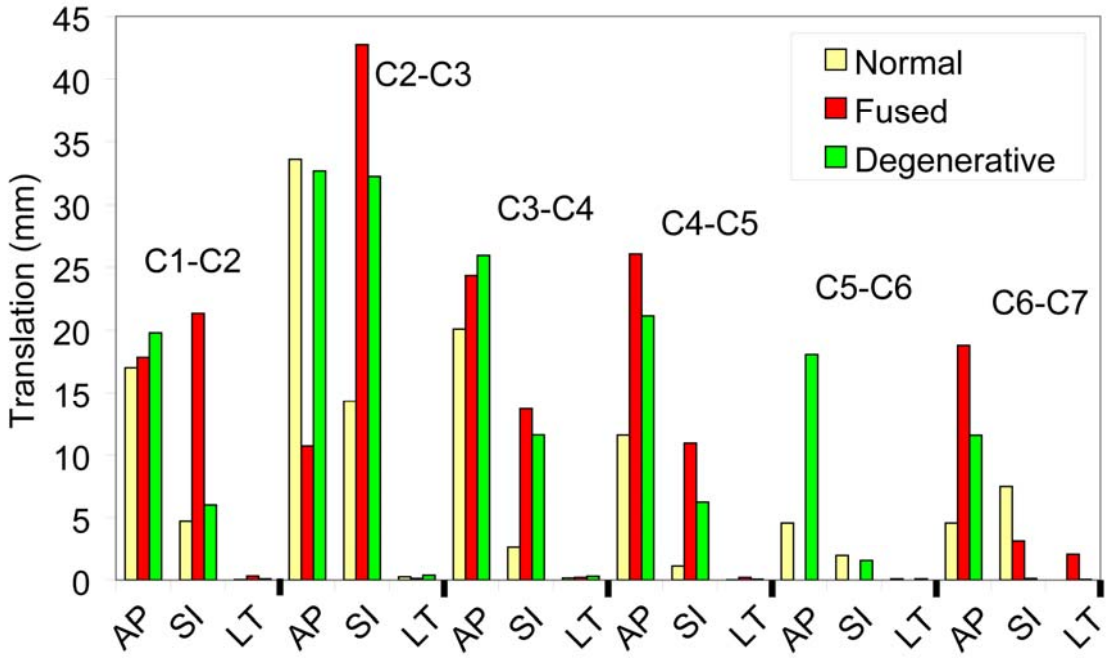


Figure 4.16 Intersegmental translations for the normal, degenerative, and ACDF subjects during the flexion-extension activity.

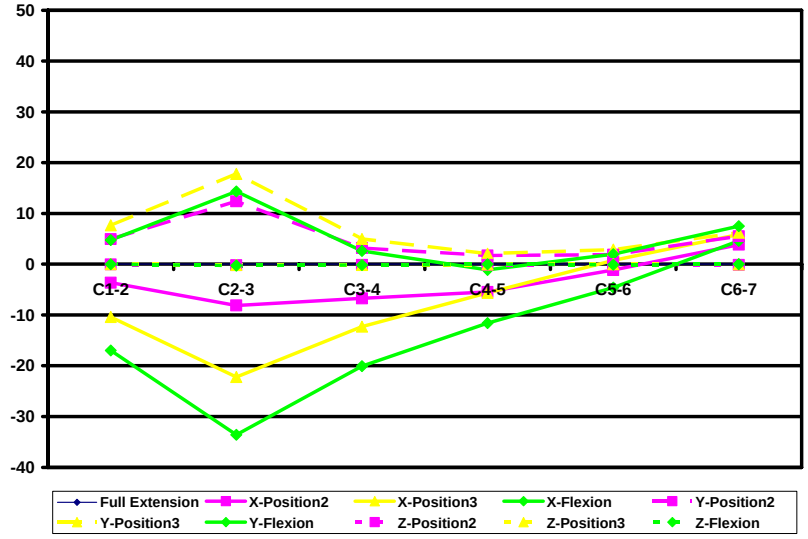


Figure 4.17 Intersegmental translation between vertebrae for the normal subject (Full Extension-Original Position, Position 2, Position 3 and Full Flexion) (Unit: mm).

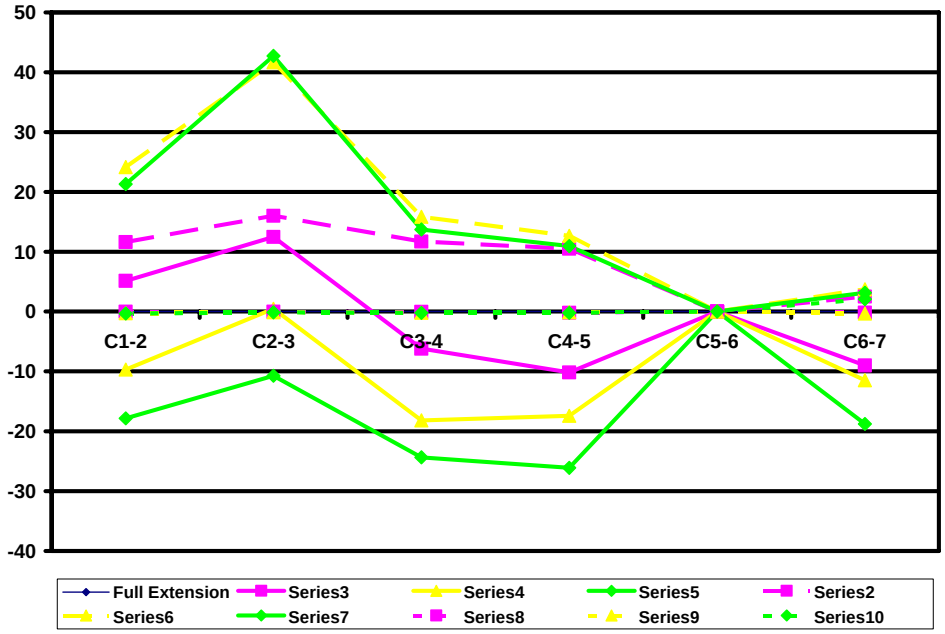


Figure 4.18 Intersegmental translation between vertebrae for the ACDF subject (Full Extension-Original Position, Position 2, Position 3 and Full Flexion) (Unit: mm).

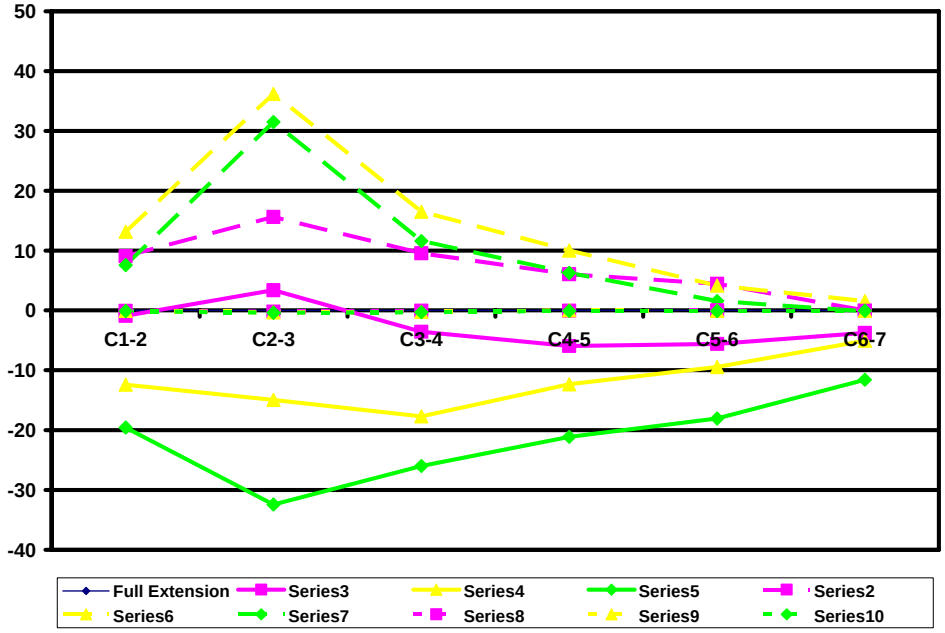


Figure 4.19 Intersegmental translation between vertebrae for degenerative subjects (Full Extension-Original Position, Position 2, Position 3 and Full Flexion) (Unit: mm).

4.3.2 Axial Rotation

During this activity, the fluoroscopic images were taken in the sagittal plane. In order to analyze the motion of the axial rotation under the natural conditions, the subjects were requested to perform the activity without lifting their chins. However, the skull blurred the C1 and C2 levels for the normal and degenerative subjects and the C1 to C3 levels for the ACDF subject on the fluoroscope images. Therefore, the 3D *in vivo* intersegmental rotations and translations were studied and compared from C3 to C7 between these three groups. This was done by registering 3D CAD models onto 2D fluoroscopic images (Figure 4.20).

The overall range of axial rotation between the adjacent vertebrae beginning at C3 and moving through C7 for the normal, degenerative, and ACDF subjects were reported as the variances between the starting position and the neutral position. In other words, this overall motion represented the axial rotation to one side. The data was then compared to the previous literature and listed in Table 4.3. The overall range of the coupled intersegmental motions was listed in Table 4.4. Overall, the intersegmental main and coupled motions of the normal subject were consistent with the previous literature. The degenerative subject had a relatively smaller overall range of both the main and coupled intersegmental motions. The ACDF subject had a significantly larger overall rangeat both the C4-C5 and C6-C7 levels as compared to the normal and degenerative subjects. The magnitudes of intersegmental AR were increased about 133.6% at the C4-5 level and 91.1% at the C5-C6 level as compared with the normal subject. At the superior level, C4-C5, the coupled LB and FE rotations were not much larger than those in the normal

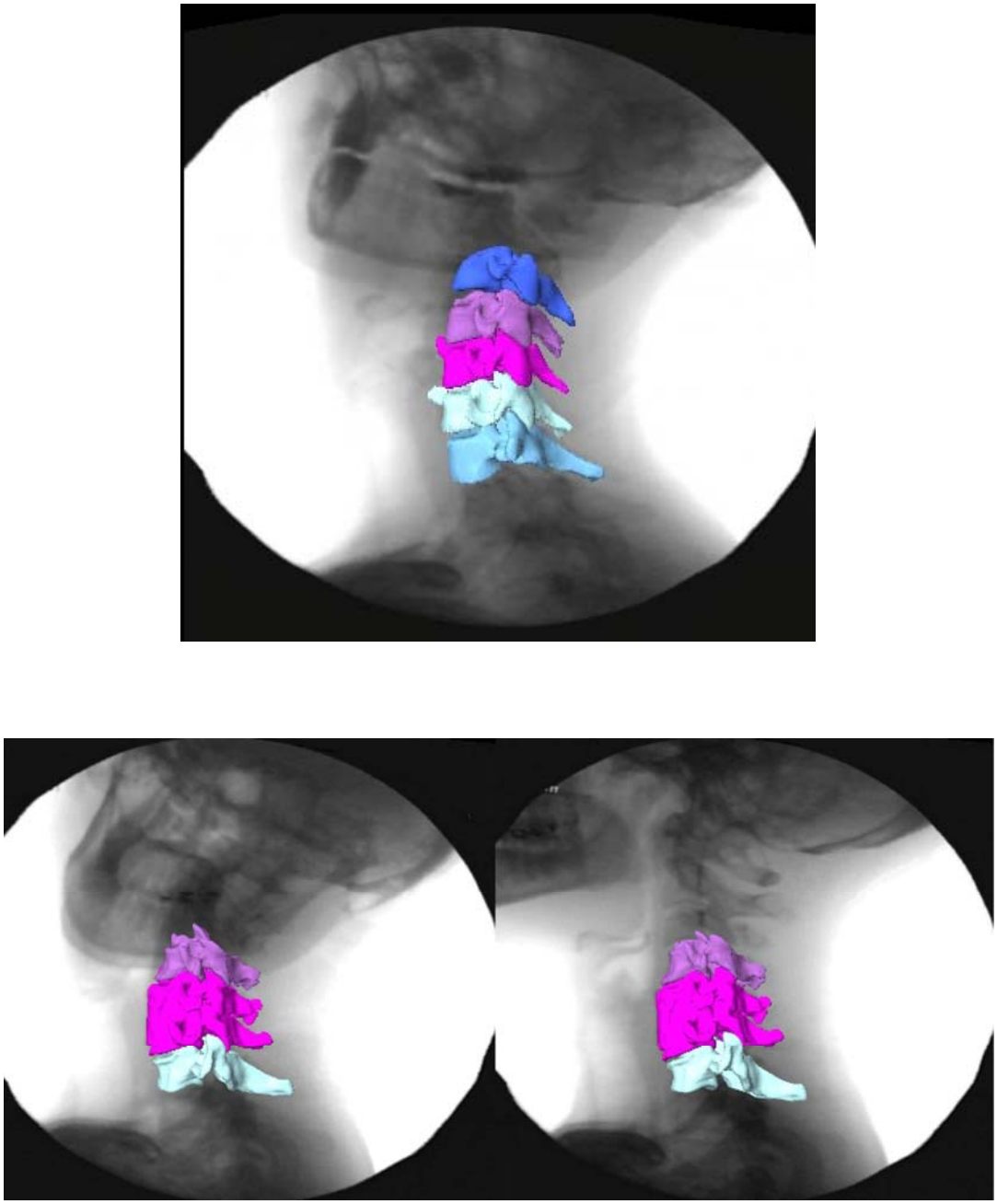


Figure 4.20 Axial rotation activity for the normal (top) and ACDF (bottom) subjects after registering 3D models onto a 2D fluoroscopic image.

Table 4.3 The overall range of axial rotation for three groups during axial rotation for one side with comparison to previous literature (Unit: Degrees)

Main AR	C4-C5	C5-C6	C6-C7
Normal	2.5	3.1	2.7
Degenerative	3.2	2.3	1.2
ACDF	5.9	NA	5.1
In vivo			
Dvorak(1987)	6.7	7.0	5.4
Penning(1987)	6.8	6.9	5.4
Mimura(1989)	2.1	2.7	3.2
Lai(1993)	3.5	3.0	3.0
Ishii(2004)	4.6	4.0	1.6
In vitro			
Lysell(1969)	5.2	4.0	2.9
Panjabi(2001)	3.4	2.6	1.5

Table 4.4 The overall range of coupled rotation of three groups during axial rotation for one side with comparison to previous literature (Unit: Degrees).

	C4-C5		C5-C6		C6-C7	
	Coupled LB	**Coupled FE**	**Coupled LB**	**Coupled FE**	**Coupled LB**	**Coupled FE**
Normal	6.2	2.6	2.7	2.5	3.1	1.2
Degenerative	1.8	2.2	1.9	1.1	0.6	1.3
ACDF	7.6	2.5	NA	NA	6.6	0.4
Mimura (1989)	6.2	2.1	4.0	2.1	2.7	2.5
Ishii (2004)	5.0	1.5	5.3	0.9	4.9	2.4

subject, which were approximately 23.0% and 5.7%, respectively. Nevertheless, the coupled LB and FE rotations at the inferior level, C6-C7, were significantly different from the results for the normal subject, which were about 112.9% and 64.1% greater, respectively. In general, the coupled intersegmental motions in the normal subject presented more smooth motion patterns and were consistent with previous literature. Comparatively, the ACDF subject exhibited different motion patterns, which coupled larger LB motions and smaller FE motions during the activity.

With respect to the coordinate system defined previously in 3D kinematics analysis, average intersegmental translations of these three subjects from C4 to C7 have been listed in Table 4.5. All three subjects experienced coupled intersegmental translations in all three directions. The maximum translation (4.0mm) was observed in the AP direction at the C6-C7 level in the normal subject (Figure 4.21).

4.3.3 Lateral Bending

During this activity, the subjects faced the image tube. The fluoroscopic images were taken in the frontal plane instead of the sagittal plane. In order to examine the lateral bending motion under natural conditions, the subjects were required to perform the

Table 4.5 Average intersegmental translations of normal, degenerative, and ACDF subjects during axial rotation (Unit: mm).

	C3-C4	C4-C5	C5-C6	C6-C7
AP	4.59	1.57	4.69	3.31
SI	1.56	1.66	1.00	2.18
LT	3.86	3.17	1.11	0.52

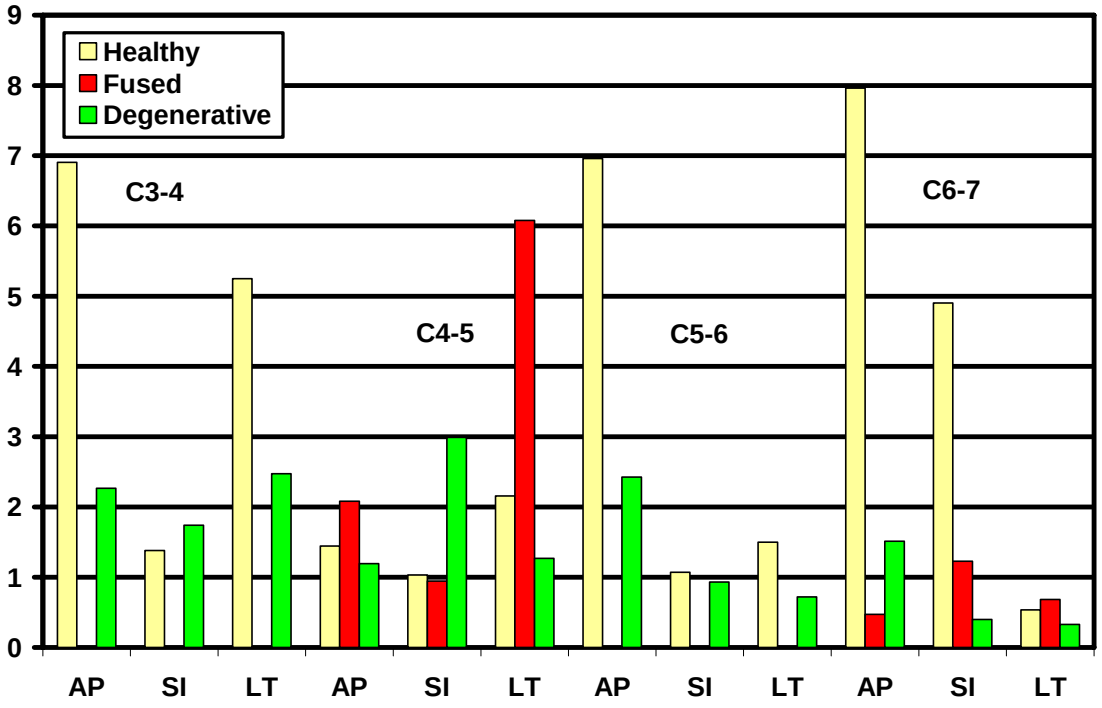


Figure 4.21 Intersegmental translations of three subjects in AP, SI, and LT directions during axial rotation (Unit: mm).

activity without lifting their chins. Because of the reason mentioned before, only 3D in vivo intersegmental rotations and translations were studied and compared only from the C4 through C7 levels among the groups (Figure 4.22 to Figure 4.24). The intersegmental rotation and translations were reported as those for the one side. Intersegmental rotation of LB between C4 and C7 for the normal, degenerative, and ACDF subjects as compared with previous literature has been plotted in Table 4.6 [19, 22, 27, 33]. The ACDF subject had a significantly larger intersegmental rotation at the C4-C56 level as compared to the normal and degenerative subjects. The magnitude at this level was about 1º larger than the sum of those at the C4-C5 and C5-C6 levels in the normal subject.

The coupled motions during the LB activity for three different groups have been listed in Table 4.7. After normalizing the values with respect to the starting position, it was found that the ACDF subject had 2.5º (28.74%) coupled FE rotation and 1.3º (14.94%) coupled AR rotation with 8.7º LB at the C4-C5 level. The magnitudes of these data are much larger than those in the normal and degenerative subjects as well as those in Ishii's study [22]. Values normalized with respect to the starting position also revealed that the degenerative subject had 0.5º (15.6%) coupled FE rotation and 0.2º (6.3%) coupled AR rotation with 3.2º LB at the C4-C5 level; there was 1.0º (28.6%) coupled FE rotation and 0.7º (20.0%) coupled AR rotation with 3.5º LB at the C5-C6 level. The intersegmental translations between C4 through C6 for all three groups have been listed in Table 4.8. The intersegmental translations in all three directions were also observed in the subjects. The maximum coupled translation (5.7mm) was observed in the LT

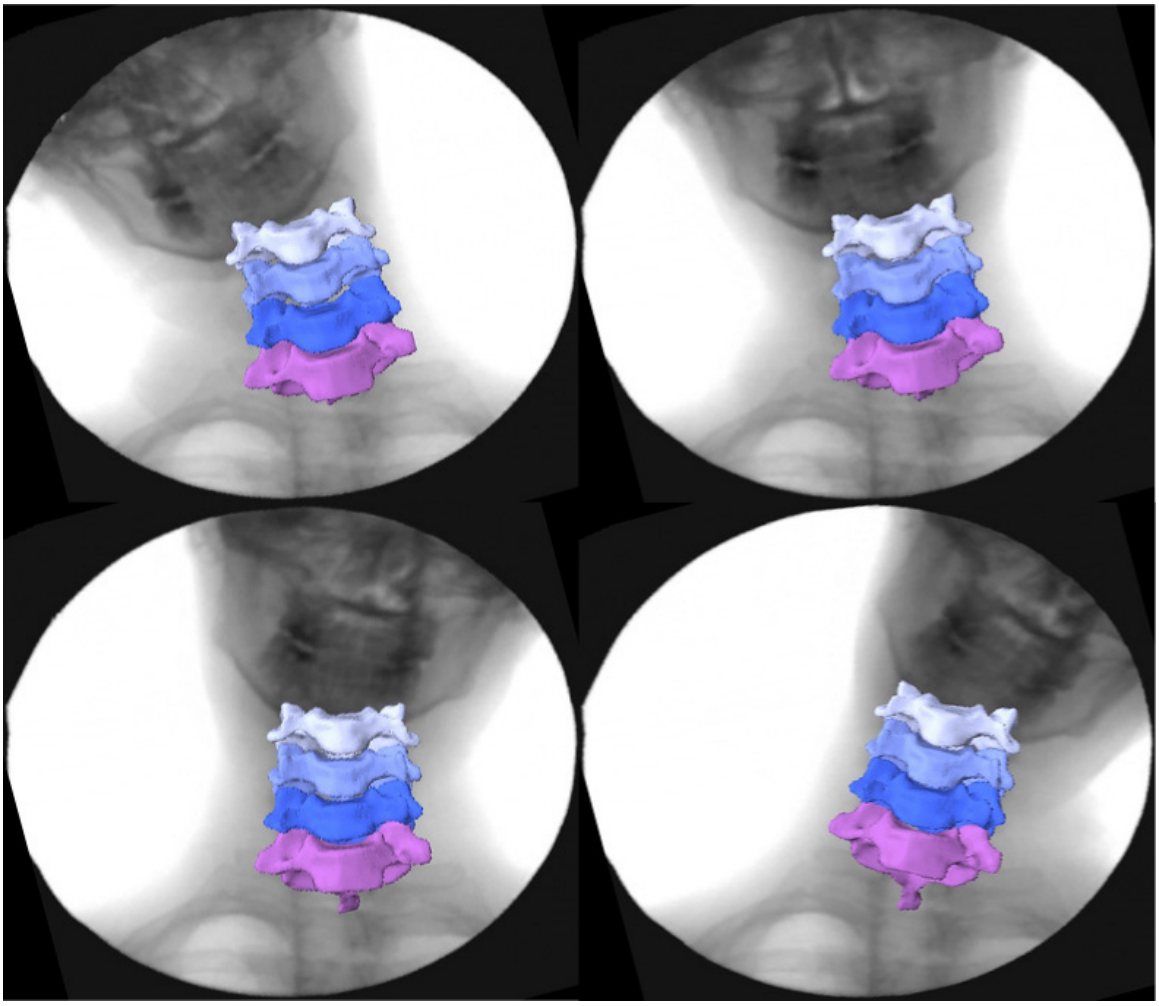


Figure 4.22 Lateral bending activity for the normal subject after registering 3D models onto 2D fluoroscopic images (from top-left to right-bottom: Full Left, Position 2, Position 3 and Full Right).

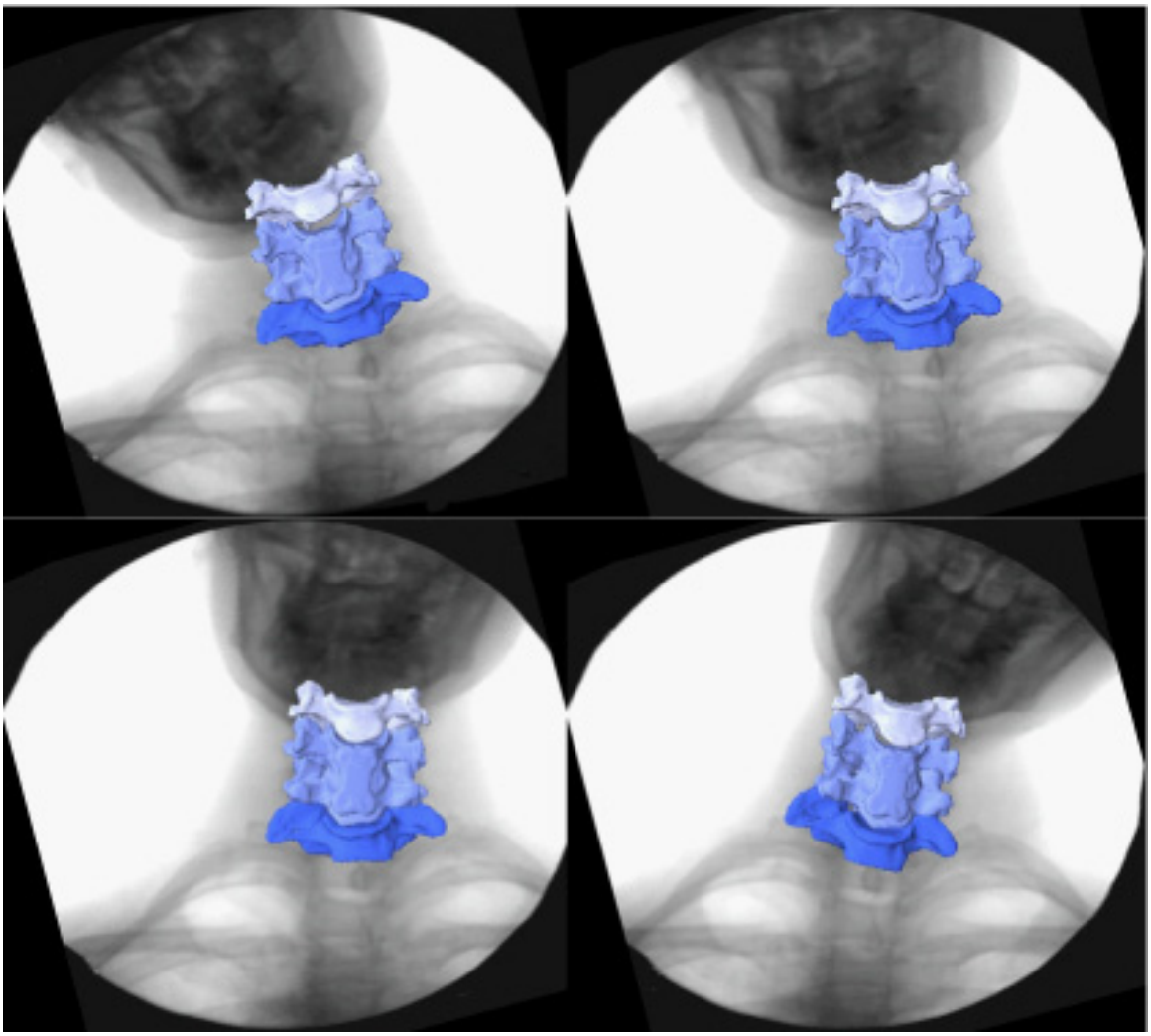


Figure 4.23 Lateral bending activity for the ACDF subject after registering 3D models onto 2D fluoroscopic images (from top-left to right-bottom: Full Left, Position 2, Position 3 and Full Right).

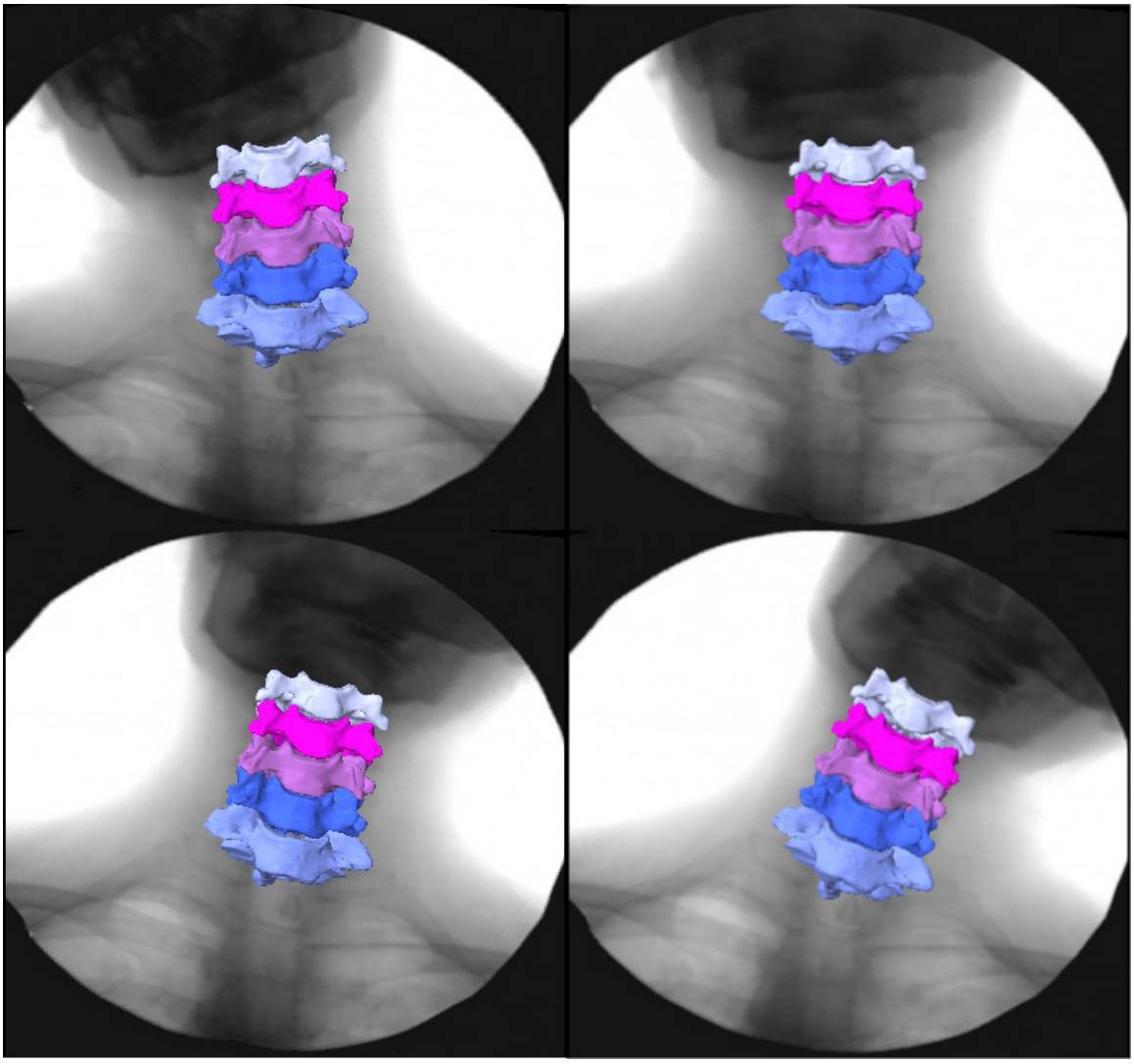


Figure 4.24 Lateral bending activity for the degenerative subject after registering 3D models onto 2D fluoroscopic images (from top-left to right-bottom: Full Left, Position 2, Position 3 and Full Right).

Table 4.6 The overall range of LB for three groups during lateral bending for one side with comparison to previous literature (Unit: Degrees)

Main LB rotation	C4-C5	C5-C6	C6-C7
Normal	2.8	5.0	6.3
Degenerative	3.2	3.5	2.0
ACDF	8.7	NA	2.2
In vivo			
Penning (1978)	3.0	3.0	3.0
Ishii (2006)	3.3	4.3	5.7
In vitro			
Moroney (1988)	4.7	4.7	4.7
Panjabi (2001)	4.7	3.3	2.7

Table 4.7 The overall range of coupled rotations between C4 and C6 for different groups (Unit: Degrees).

		LB	Coupled AR	Coupled FE
C4-C5	**Normal**	2.8	0.4	0.7
	Degenerative	3.2	0.5	0.2
	ACDF	8.7	2.5	1.3
C5-C6	**Normal**	5.0	0.0	1.7
	Degenerative	3.5	1.0	0.7
	ACDF	NA	NA	NA
C6-C7	**Normal**	6.3	0.7	1.2
	Degenerative	2.0	1.7	2.5
	ACDF	2.2	2.0	0.6

Table 4.8 Intersegmental translations of vertebrae for normal, degenerative and ACDF subjects during the lateral bending activity (Unit: mm).

		Coupled AP	Coupled SI	Coupled LT
C4-C5	**Normal**	1.6	0.1	1.0
	Degenerative	2.9	0.4	4.6
	ACDF	3.9	3.2	4.4
C5-C6	**Normal**	2.9	0.1	5.1
	Degenerative	2.4	1.5	3.1
	ACDF	NA	NA	NA
C6-C7	**Normal**	2.2	1.9	3.1
	Degenerative	3.9	2.3	2.1
	ACDF	3.3	3.2	5.7

direction at the C6-C7 level in the ACDF subject. At both the C4-C5 and C6-C7 levels, the ACDF had larger intersegmental translations than those of the normal subject.

4.4. 3D Kinetics

Two 3D mathematical models both including five different types of ligaments across the lower cervical spine, were derived based on Kane's dynamics. One did not include the muscular forces; the other did. During the flexion-extension activity, contact forces in the 1, 2 and 3 directions, shown in Figure 3.14 and Figure 3.14, were predicted and relabeled as anteroposterior forces, compression forces and lateral forces, respectively (Table 4.9). Anteroposterior and compression forces in this 3D study had the same directions as those in the 2D studies, respectively.

According to the first 3D mathematical model which did not include the muscular forces, the average anteroposterior forces was 0.53 SW and the average compression

Table 4.9 Contact forces predicted by the 3D mathematical model including 5 different types of ligaments but not muscular forces in the normal, degenerative, and ACDF cervical spines during the flexion-extension activity.

		Normal				Degenerative				ACDF			
		2D		3D		2D		3D		2D		3D	
		Max	Ave	Max	Ave	Max	Ave	Max	Ave	Max	Ave	Max	Ave
Antero -posterior forces	C6-C7	0.22	0.09	0.64	0.29	0.06	0.02	1.09	0.26	2.07	0.31	2.69	0.53
	C4-C5	0.35	0.13	0.47	0.28	0.11	0.05	0.35	0.14	0.94	0.15	1.81	0.45
	C3-C4	0.63	0.43	0.46	0.28	0.61	0.43	0.26	0.12	0.49	0.29	0.46	0.13
Compression Forces	C6-C7	3.09	1.97	2.83	1.74	2.24	1.79	2.10	1.59	2.63	2.03	4.78	2.21
	C4-C5	3.29	1.78	1.68	1.32	2.02	1.48	1.58	1.27	2.54	1.80	2.91	1.80
	C3-C4	2.93	1.72	1.56	1.24	2.07	1.56	1.51	1.15	2.63	1.95	1.44	1.08
Lateral forces	C6-C7	NA	NA	0.15	0.07	NA	NA	0.33	0.17	NA	NA	0.10	0.04
	C4-C5	NA	NA	0.13	0.06	NA	NA	0.33	0.16	NA	NA	0.16	0.07
	C3-C4	NA	NA	0.09	0.07	NA	NA	0.33	0.15	NA	NA	0.21	0.08

- **The maximum data was the maximum magnitude without considering the force direction.**
- **The average data was the average magnitude of all the forces without considering the force direction.**

force was 2.21 SW in the ACDF patient; 0.29 SW and 1.74 SW in the normal subject; and 0.26 SW and 1.59 SW in the degenerative subject, respectively.

The compression forces (direction 2 in FBD) at the C4-C5 and C6-C7 levels ranged from 0.79 to 1.68 SW and from 0.93 to 2.83 SW in the normal subject, respectively (Figure 4.25); from 0.79 to 1.58 SW and from 0.93 to 2.10 SW in the degenerative patient (Figure 4.26); from 0.81 to 2.91 SW and from 0.98 to 4.78 SW in the ACDF patient, respectively (Figure 4.27). At the C3-C4 level, the compression forces had no significant difference among these three subjects: from 0.72 to 1.56 SW in the normal subject; from 0.71 to 1.51 SW in the degenerative subject; and from 0.71 to 1.44 SW in the ACDF subject.

At the C4-C5 level, the anteroposterior forces (direction 1 in FBD) ranged from 0 to 0.47 SW in the normal subject, from -0.19 to 0.35 SW in the degenerative subject, and from -1.81 to 0.51 SW in the ACDF subject. The anteroposterior forces at the C6-C7 level ranged from -0.02 to 0.64 SW in the normal subject (Figure 4.28); from -1.09 to 0.37 SW in the degenerative patient (Figure 4.29); and from 2.69 to 0.52 SW in the ACDF patient (Figure 4.30). At the C3-C4 level, the anteroposterior forces had no significant difference among these three subjects: range from 0 to 0.46 SW in the normal subject; from -0.46 to 0.26 SW in the degenerative subject; and from 0.02 to 0.46 SW in the ACDF subject.

Relatively speaking, the lateral forces (direction 3 in FBD) among all three subjects were small. At the C4-C5 and C6-C7 levels, they ranged from -0.13 to 0.08 SW and from -0.05 to 0.15 SW in the normal subject (Figure 4.31); from 0 to 0.33 SW and

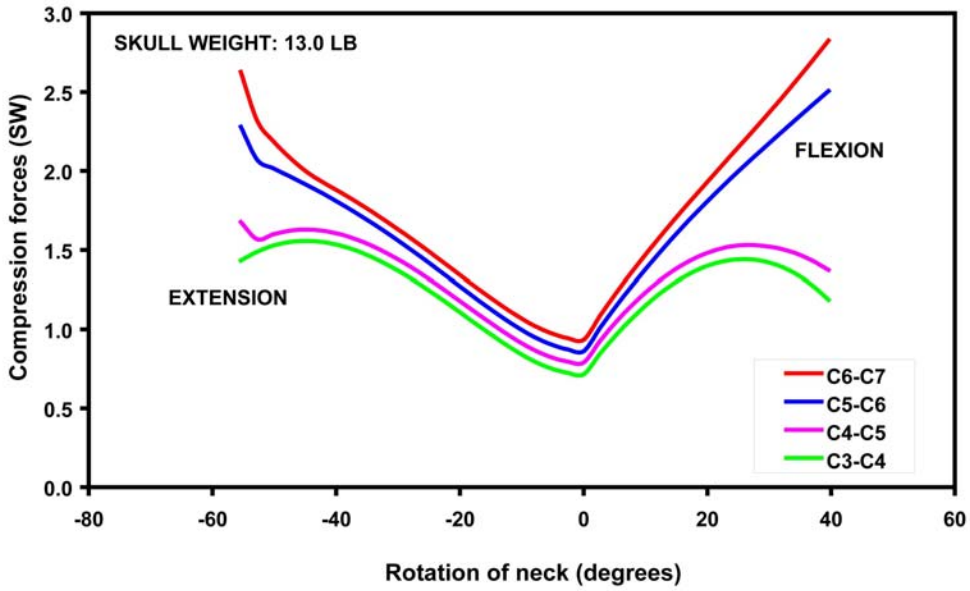


Figure 4.25 Compression forces in the normal cervical spine.

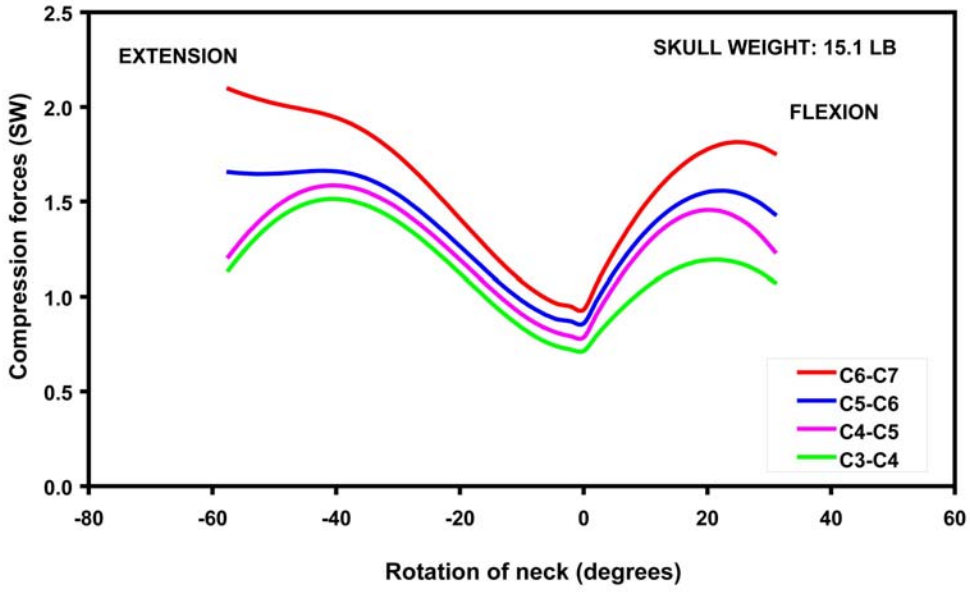


Figure 4.26 Compression forces in the degenerative cervical spine.

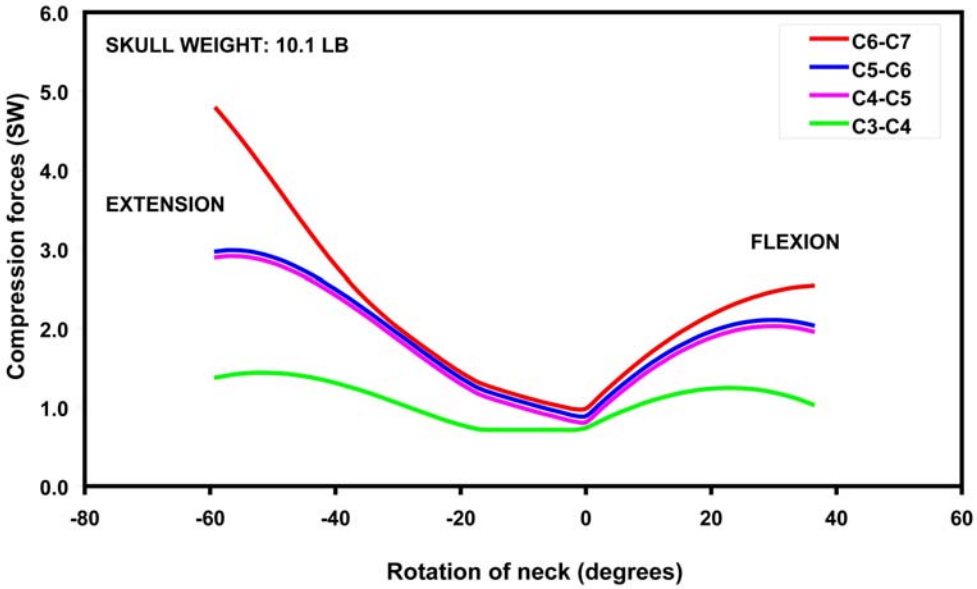


Figure 4.27 Compression forces in the ACDF cervical spine.

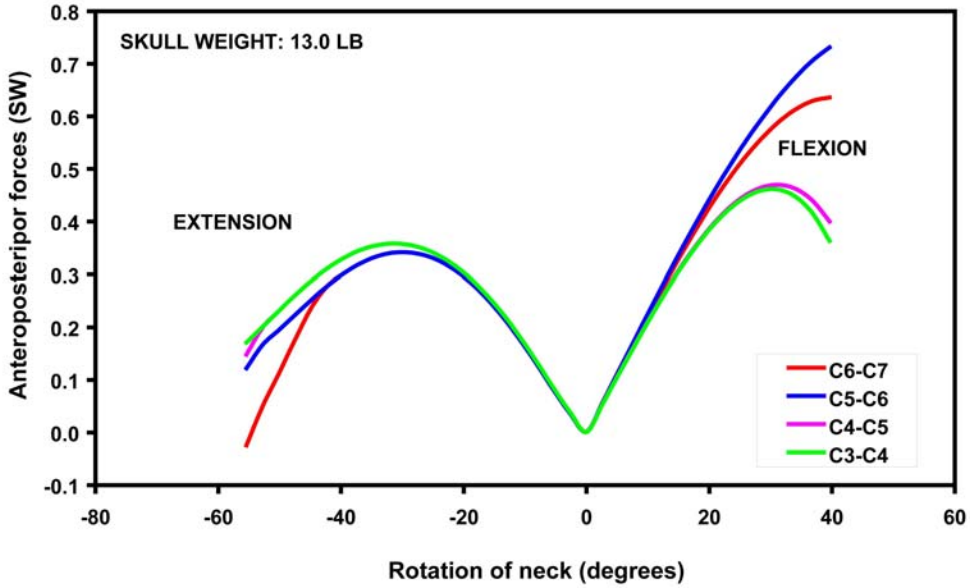


Figure 4.28 Anteroposterior forces in the normal cervical spine.

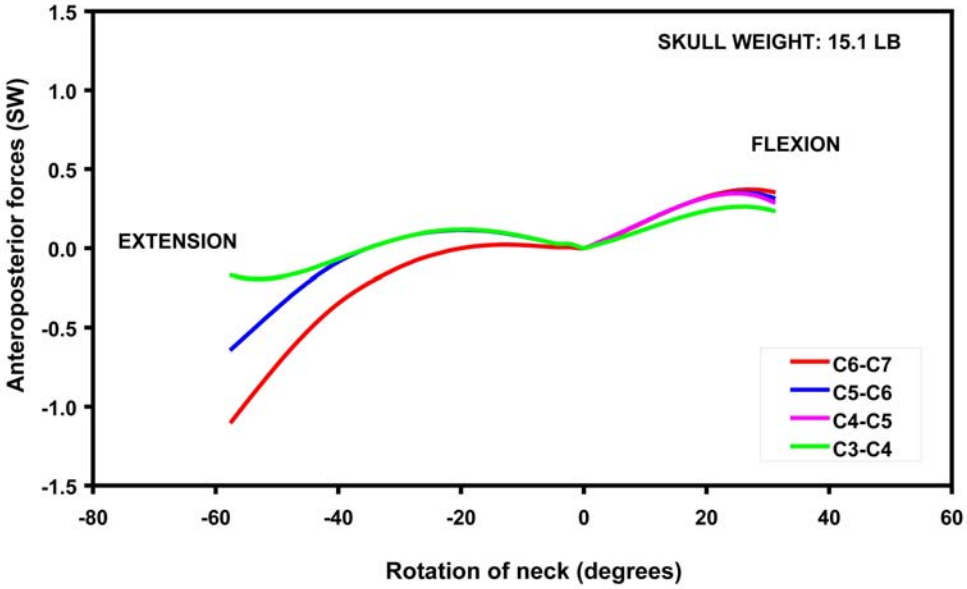


Figure 4.29 Anteroposterior forces in the degenerative cervical spine.

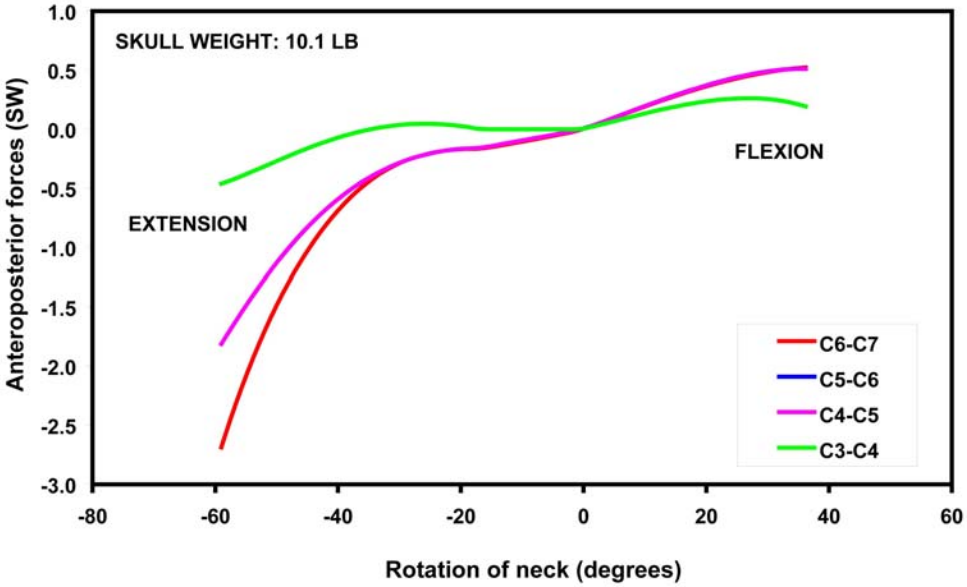


Figure 4.30 Anteroposterior forces in the ACDF cervical spine.

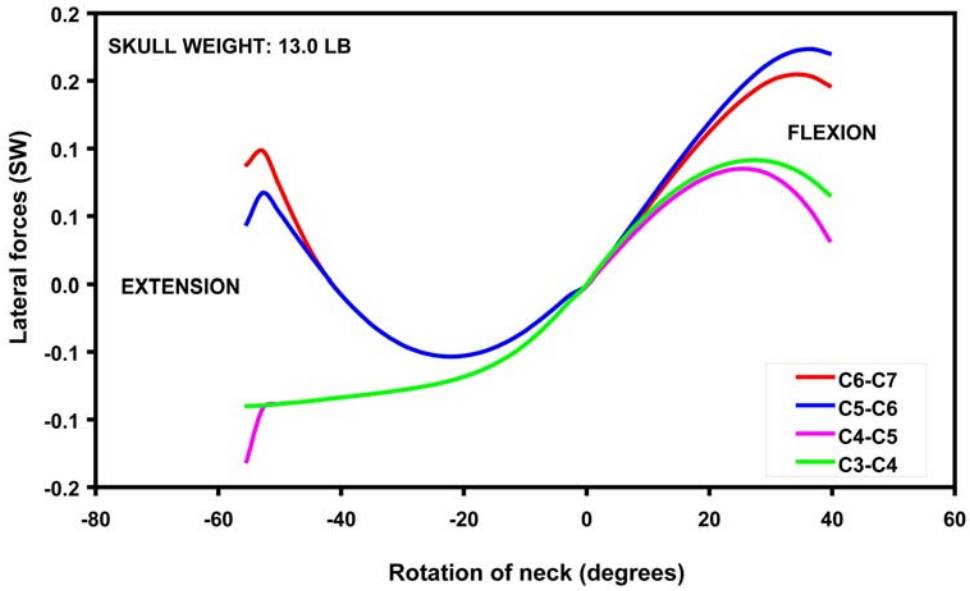


Figure 4.31 Lateral forces in the normal cervical spine.

from 0 to 0.33 SW in the degenerative patient (Figure 4.32); and from -0.16 to 0.04 SW and from -0.10 to 0.07 SW in the ACDF patient, respectively (Figure 4.33).

At the C3-C4 level, the lateral forces ranged from -0.09 to 0.09 SW in the normal subject; from 0.02 to 0.33 SW in the degenerative subject; and from -0.21 to 0.06 SW in the ACDF subject.

Two groups of muscles -- flexors and extensors -- were added to the above mathematical model, the compression forces were plotted in Figure 4.34; the anteroposterior forces were plotted in Figure 4.35; the lateral forces were plotted in Figure 4.36; the muscular forces (combination of flexors and extensors forces) were plotted in Figure 4.37.

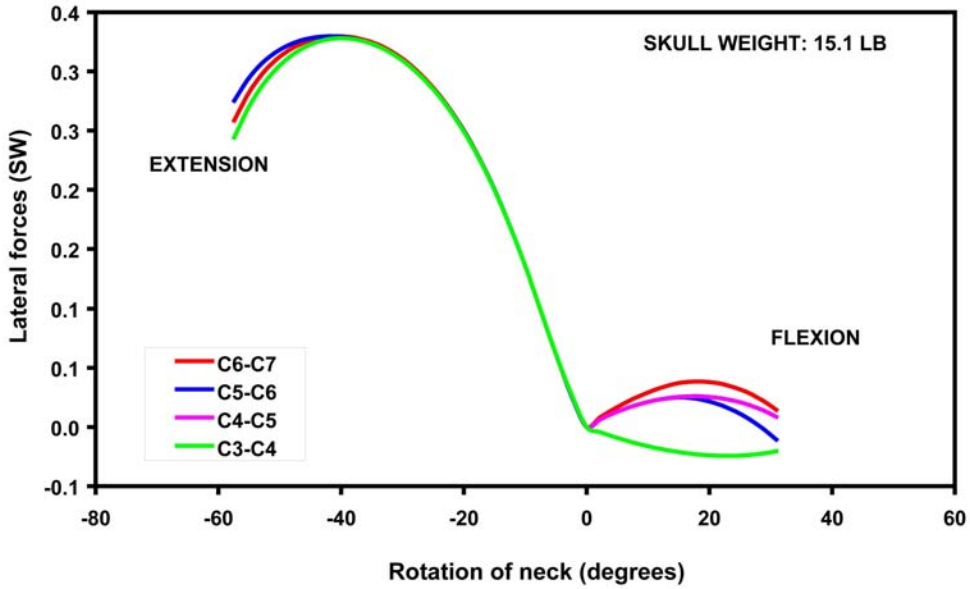


Figure 4.32 Lateral forces in the degenerative cervical spine.

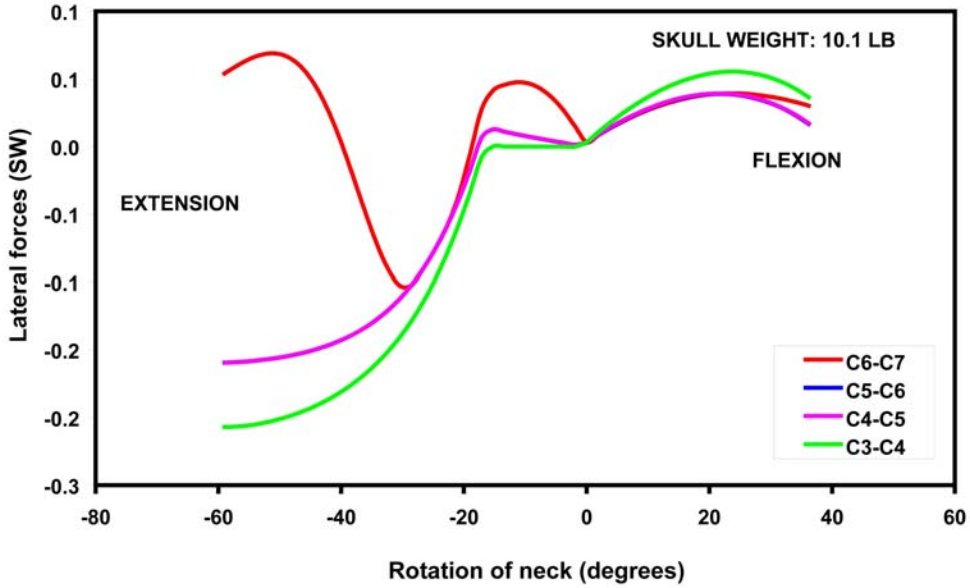


Figure 4.33 Lateral forces in the ACDF cervical spine.

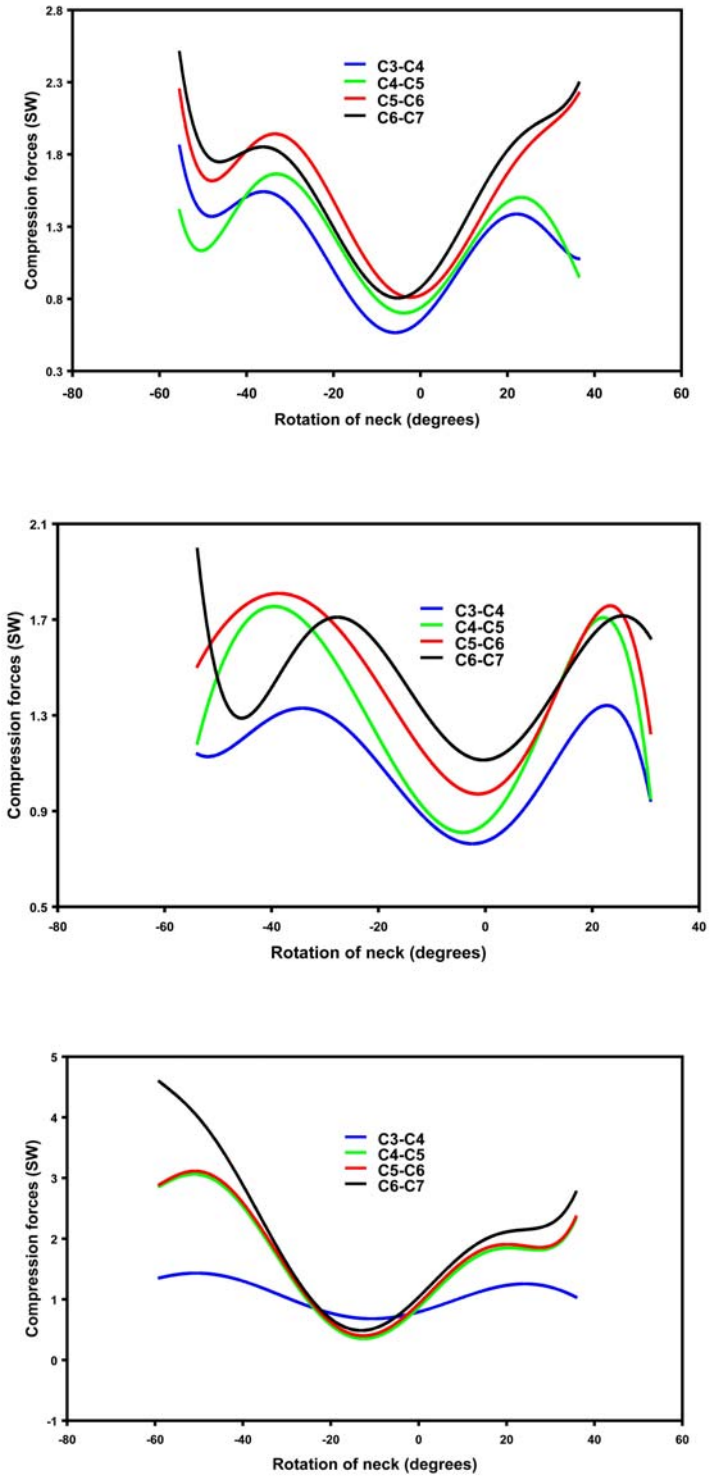


Figure 4.34 Compression forces in the normal (above), degenerative (middle), and ACDF (bottom) cervical spine after adding flexors and extensors to the mathematical models.

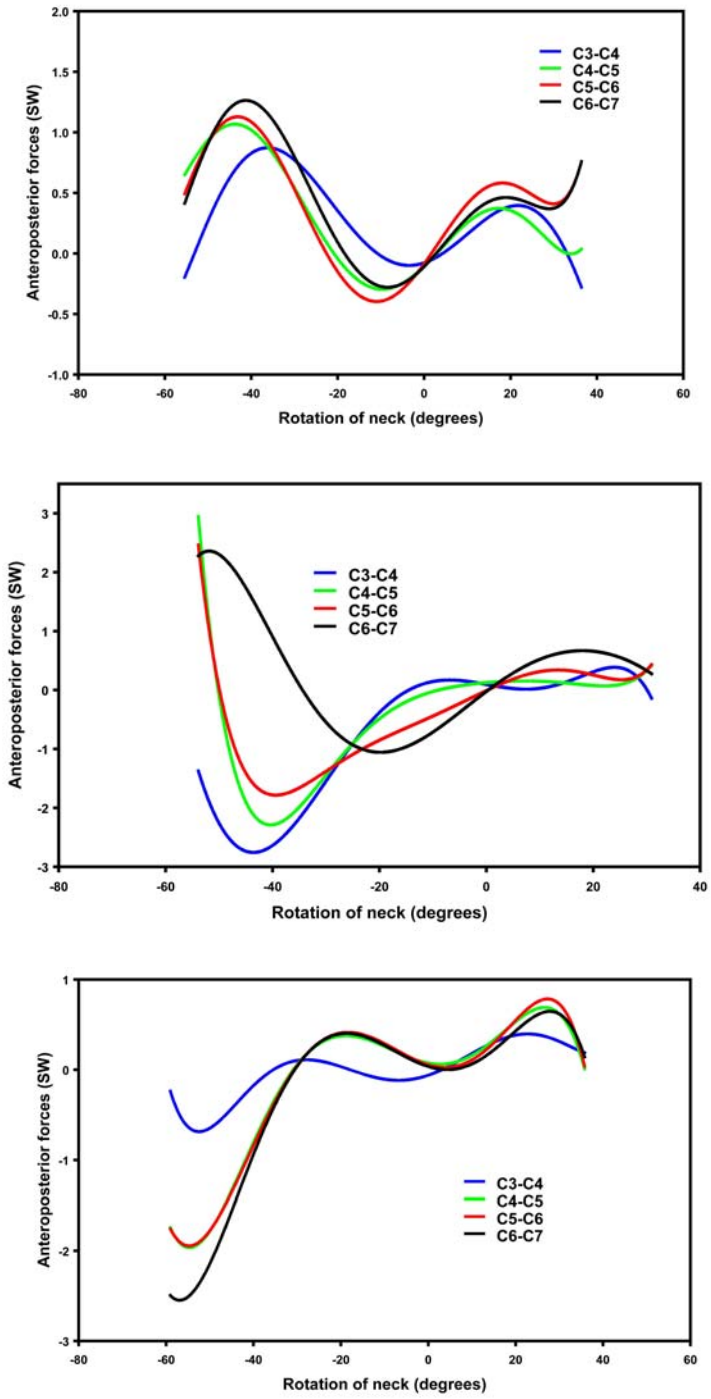


Figure 4.35 Anteroposterior forces in the normal (above), degenerative (middle), and ACDF (bottom) cervical spine after adding flexors and extensors to the mathematical models.

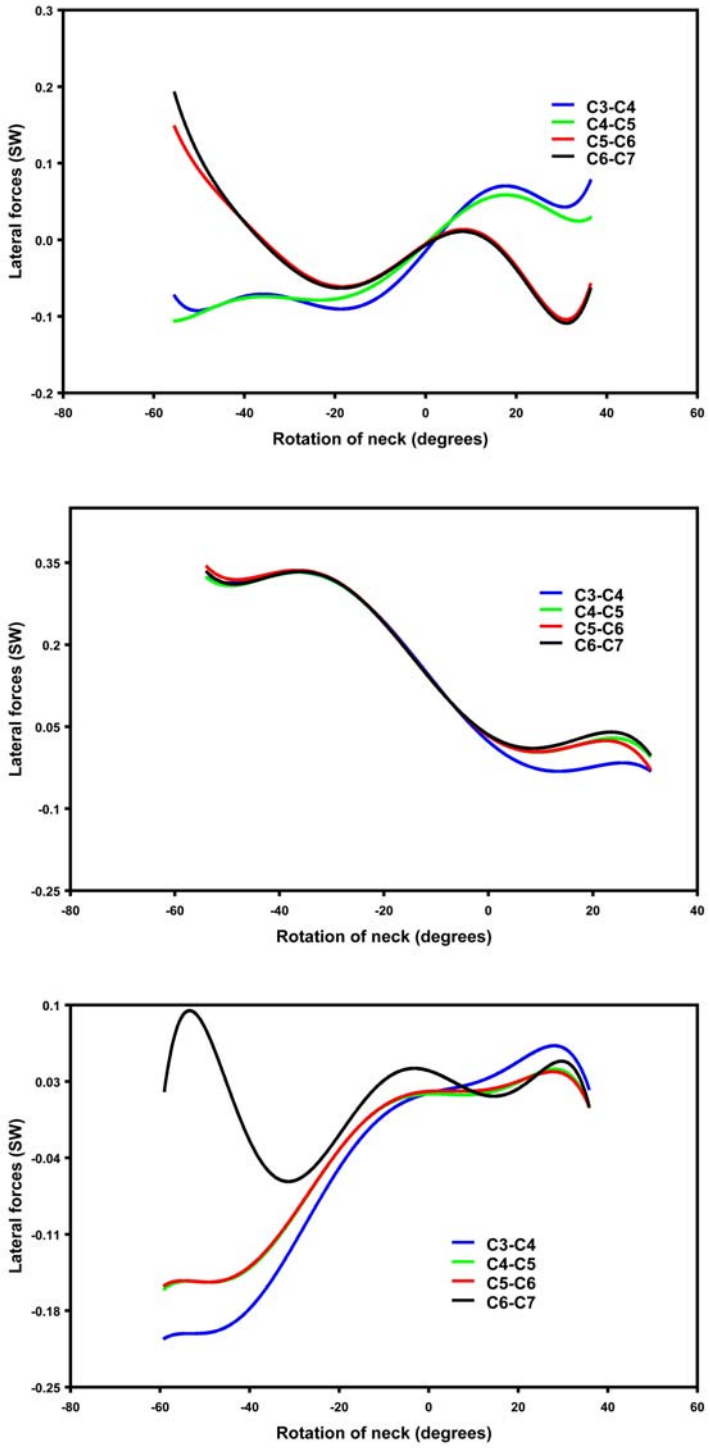


Figure 4.36 Lateral forces in the normal (above), degenerative (middle), and ACDF (bottom) cervical spine after adding flexors and extensors to the mathematical models.

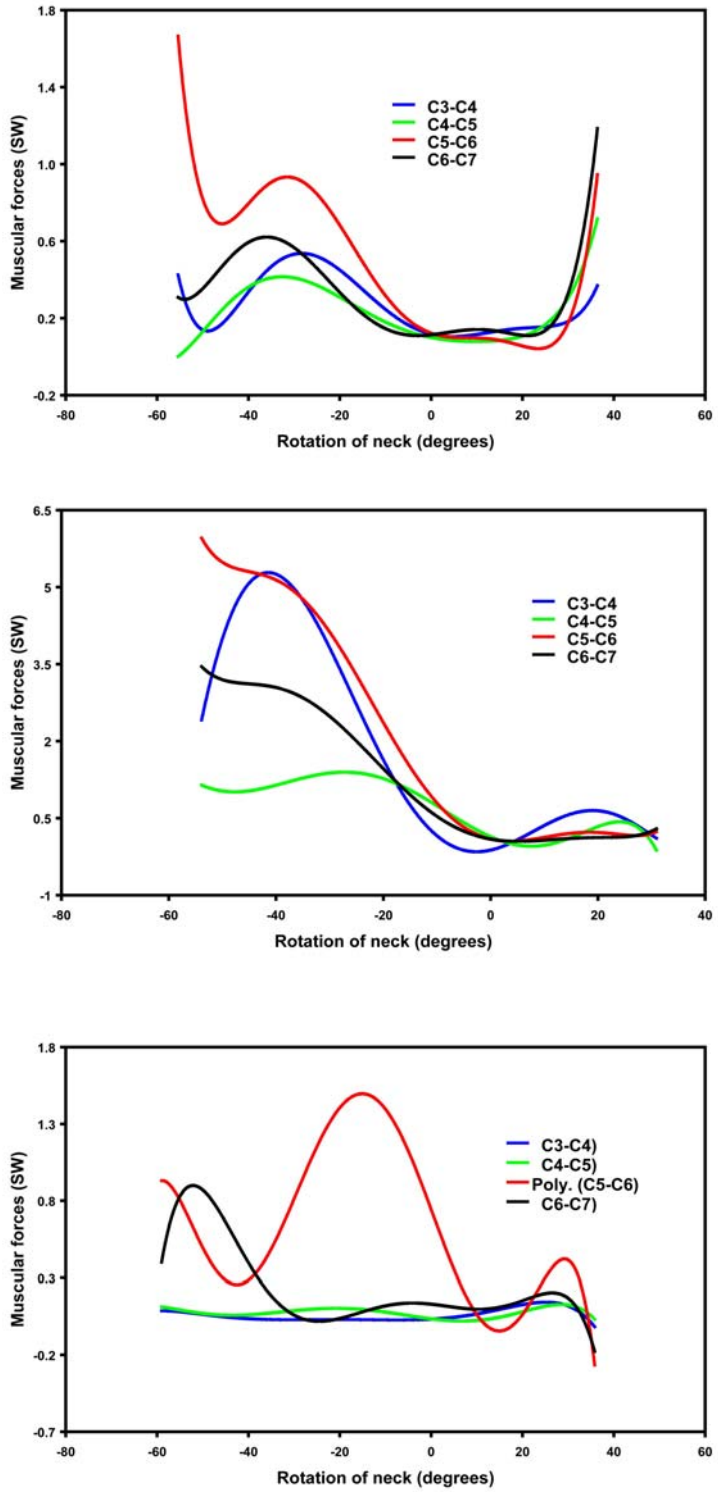


Figure 4.37 Muscular forces in the normal (above), degenerative (middle), and ACDF (bottom) cervical spine after adding flexors and extensors to the mathematical models.

4.5. Error Analysis Experiment

4.4.1 Sensor Calibration

The *FlexiForce*® Load/Force sensors were conditioned and calibrated before the experiment. A puck used in the calibration procedure was created using a rapid prototype (RP) machine. According to manufacturer requirements, a 2200g scale, which was assumed as 110 % of the predicted applied load on the sensor, was applied five times for about 60s before the experiment in order to condition the sensors and reduce the effects of hysteresis and drift. The reaction time was setup as 0.01 s for NI data acquisition system (Rate 1000HZ, Sample to read 10). The properties of the sensors such as drift and repeatability were tested. Each drift test took 2 hours to complete by applying a 1000g load on the sensors. There was a 0.5 hour resting time between two adjacent tests (Figure 3.33). In total, five tests were performed. The average rate of the difference between the first five minutes and the last five minutes during the tests was 1.3 % (Table 4.10).

Repeatability was tested to ensure that the sensor could output the same result when applying the same load under similar loading conditions. These tests were performed for three loading weights of 200g, 500g and 1000g. The testing time was 5 minutes for each test and every test was repeated 5 times. The coefficient of variation (STD/AVE) for 200g, 500g and 1000g was 5.97%, 10.46% and 1.98%, respectively (Table 4.11).

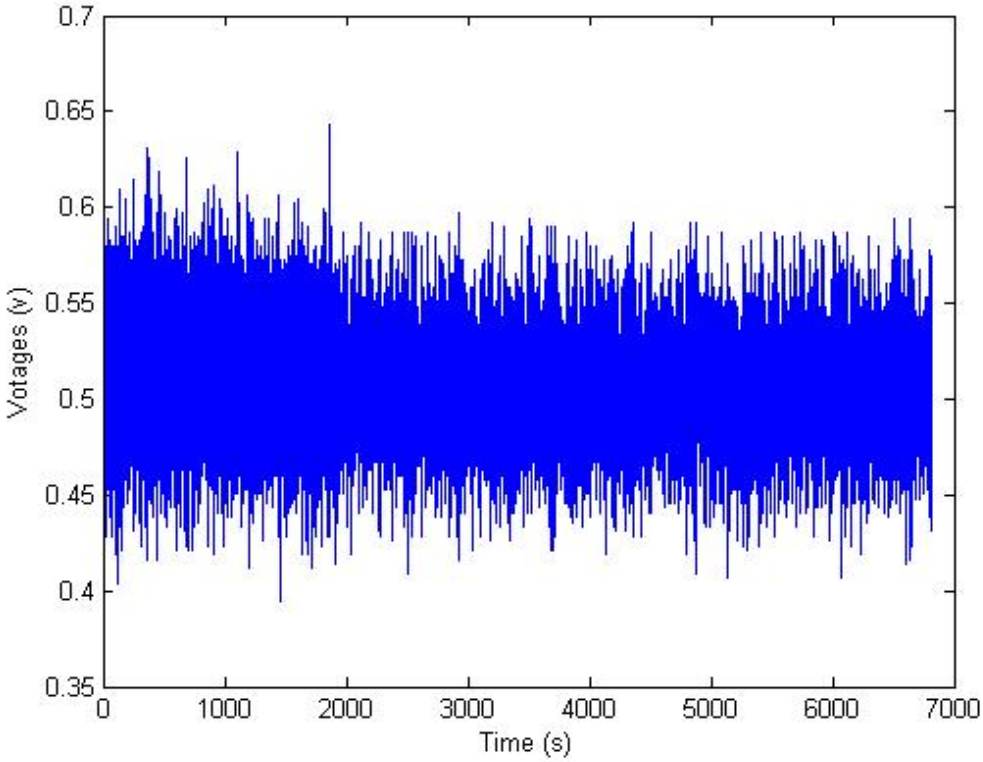


Figure 4.38 The original data of a drifting test by applying a 1000g load on a *FlexiForce*® Load/Force sensors for 2 hours.

Table 4.10 The average errors for the drift testing.

Time	1	2	3	4	5	AVE	STD
Error percentage	**0.028**	**0.008**	**0.006**	**0.006**	**0.018**	**0.013**	**0.010**

Table 4.11 The results of the repeatability test.

Time / Load	1	2	3	4	5	AVE	STD
200	0.12±0.006	0.11±0.003	0.104±0.003	0.104±0.002	0.11±0.03	0.1096	0.006542
500	0.26±0.05	0.24±0.16	0.26±0.01	0.23±0.008	0.20±0.10	0.238	0.0249
1000	0.50±0.13	NA	0.51±0.07	0.50±0.02	0.50±0.17	0.505	0.01

The linearity relationship between output voltages and the applied loads was tested 5 times by loading 20g, 50g, 100g, 200g, 500g, 1000g, 2000g and 2200g (Figure 4.39). Results showed that the sensor system had a good linear relationship between the voltage output and the applied force. During the calibration, it was indicated that the sensor had more stable and accurate readings at loads of 2000 g and 2200 g. The system output had a relatively large variation at loads of 20 g, 100 g, and 500 g as compared with other applied loads.

The above results suggested that the output of the sensor system was reliable and could be used for further cadaveric experiments.

4.4.2 Kinematic Error Analysis

The overlaid images for flexion-extension, axial rotation and lateral bending have been included in Figure 4.40 and Figure 4.42.

The 3D kinematic results of the error analysis are listed in Table 4.12. Overall, the fluoroscope and 3D-to-2D registration method had an average error of 0.5 mm for 3D translation and 0.5º for rotation in the cadaveric cervical spine experiment. Specifically, the root mean square (RMS) of rotation and RMS of translation were 0.5º and 0.6 mm for the flexion-extension activity, 0.5º and 0.5 mm for the axial rotation activity and 0.4º and 0.5 mm for the lateral bending activity, respectively. In the experiment, the MicronTracker system was treated as the absolute reference frame of the system. However, according to the user's manual, this system does have more than 0.25 mm of absolute error for translation as well as a degree variant error for rotation depending on the type of application. Therefore, the errors analysis reported in Table 4.12 included the

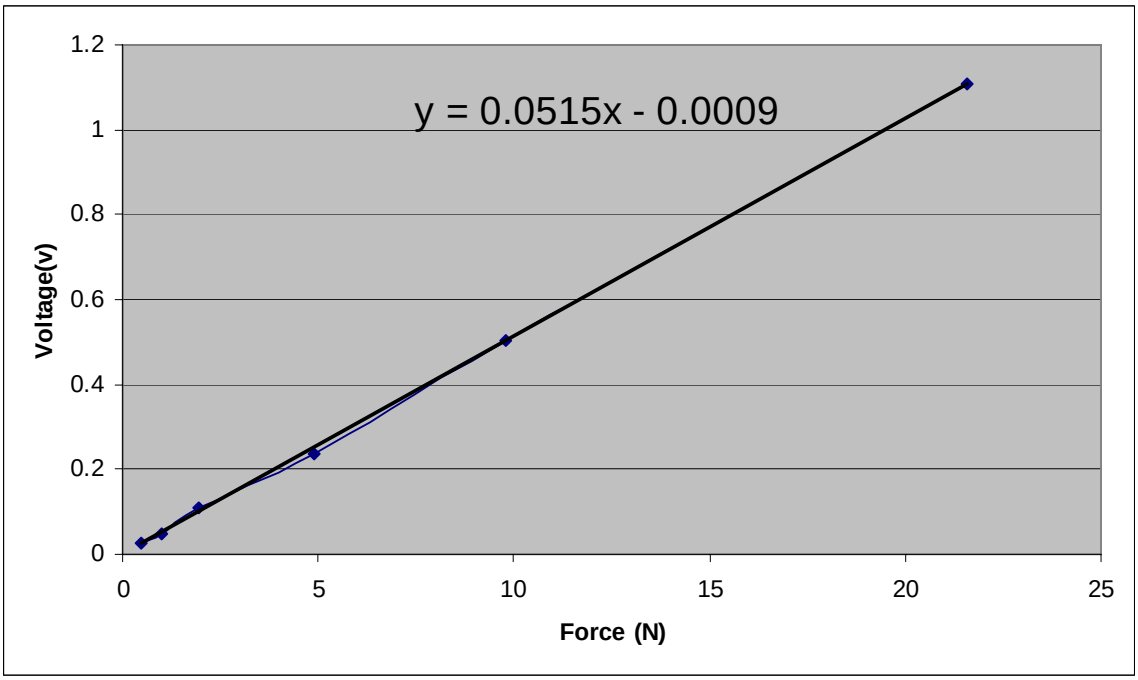


Figure 4.39 The calibration of a *FlexiForce*® Load/Force sensor.

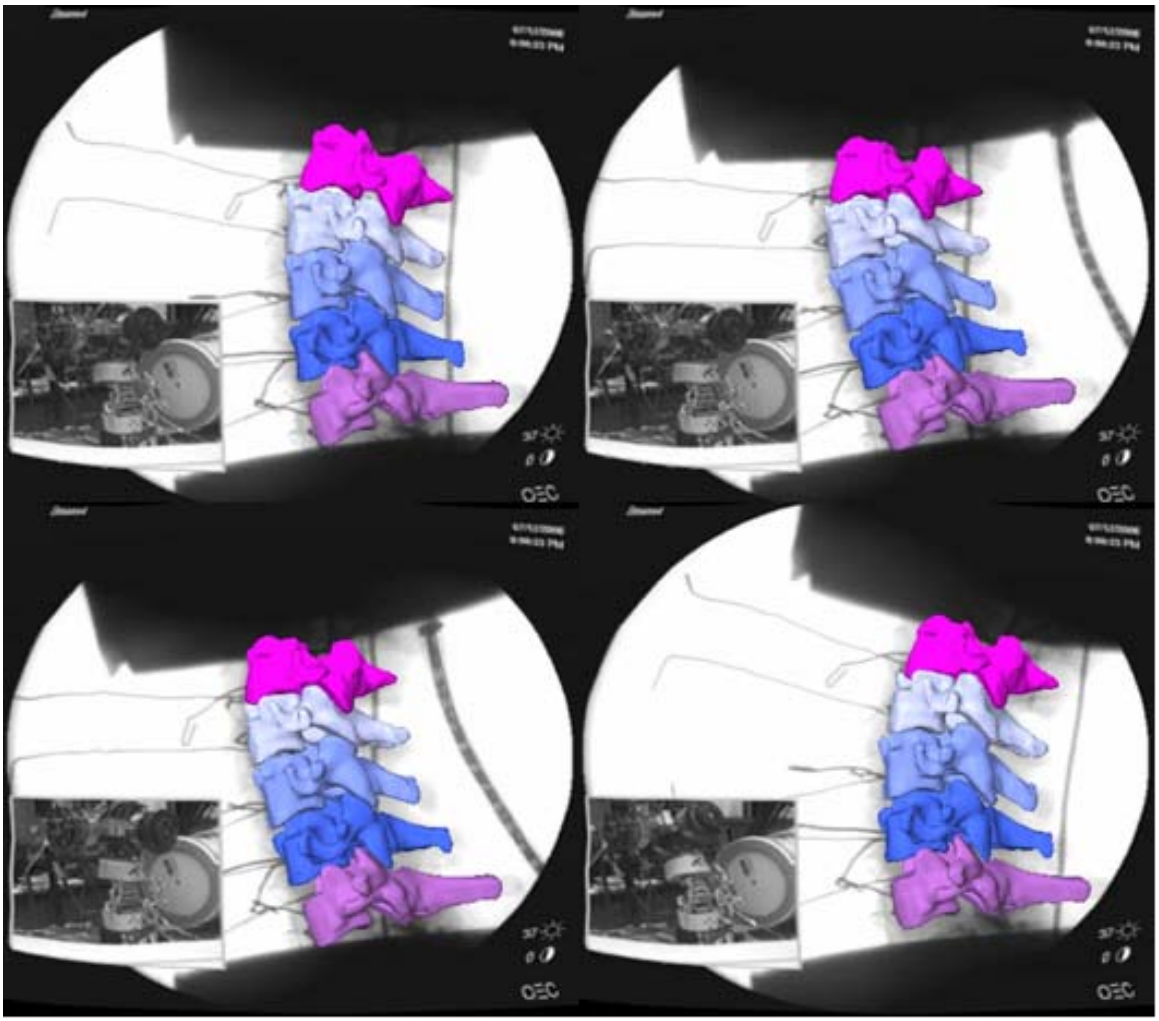


Figure 4.40 Flexion-extension activity of the cadaveric cervical spine after registering 3D models onto a 2D fluoroscopic image.

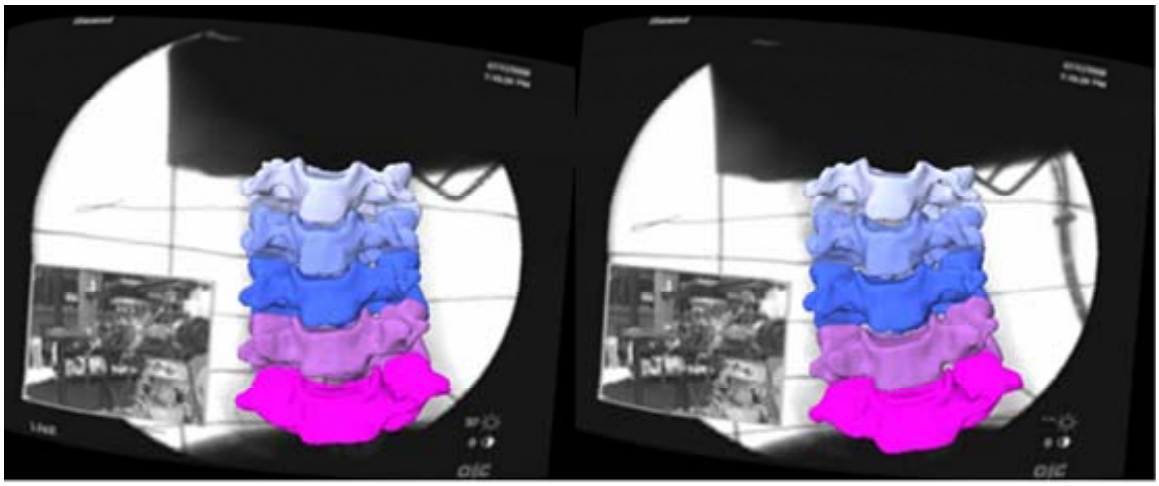


Figure 4.41 Axial rotation activity of the cadaveric cervical spine after registering 3D models onto a 2D fluoroscopic image.

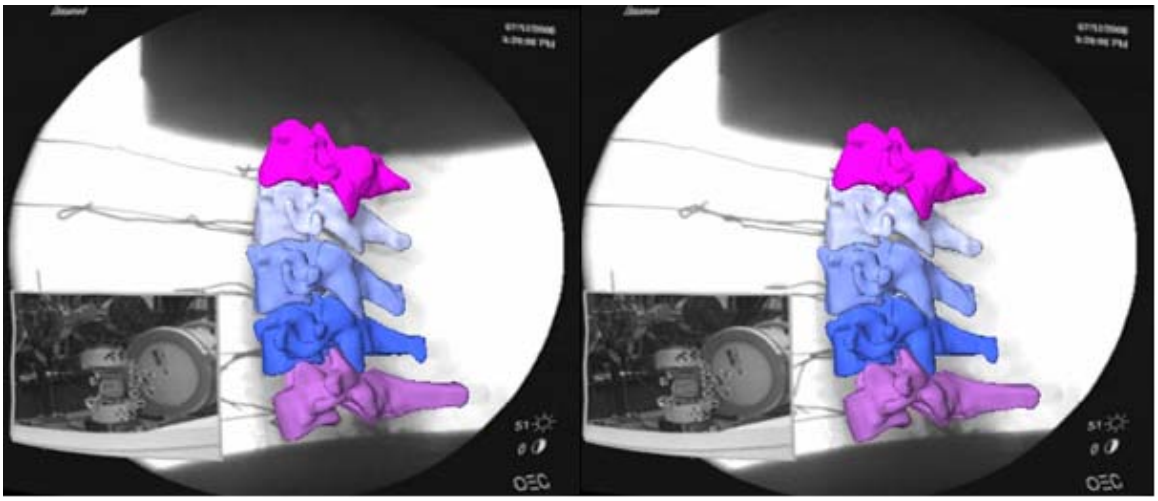


Figure 4.42 Lateral bending activity of the cadaveric cervical spine after registering 3D models onto a 2D fluoroscopic image.

Table 4.12 Error analysis results of the kinematic comparison in the cadaveric cervical spine experiment.

	Axial Rotation		Flexion Extension		Lateral Bending	
	AVE	STD	AVE	STD	AVE	STD
LB	0.54	0.42	0.44	0.39	0.43	0.53
AX	0.54	0.47	0.43	0.47	0.46	0.56
FE	0.49	0.40	0.56	0.42	0.43	0.47
AP	0.53	0.37	0.61	0.61	0.46	0.44
SI	0.38	0.34	0.67	0.55	0.46	0.48
LT	0.42	0.33	0.57	0.43	0.47	0.50
RMSR	0.52	NA	0.48	NA	0.44	NA
RMST	0.45	NA	0.62	NA	0.46	NA

Units: Rotation, degrees; Translation, mm.
LB: Lateral Bending; in SAAM, rotation with respect to X axis.
AX: Axial Rotation; in SAAM, rotation with respect to Y axis.
FE: Flexion-Extension rotation; in SAAM, rotation with respect to Z axis.
AP: Anteroposterior translation; in SAAM, translation with respect to X direction.
SI: Superoinferior translation; in SAAM, translation with respect to Y direction.
LT: Lateral Translation; in SAAM, translation with respect to Z direction.
RMSR: Root mean square of rotation in X, Y and Z.
RMST: Root mean square of translation in X, Y and Z.

errors for both translation and rotation from the MicronTracker system. The 3D kinematic method in this research study may be more accurate than the results listed in Table 4.11. Since the minimum increment of CT slices was 0.33 mm for the current CT machines at VUMC, it is hypothesized that the accuracy of this method could be improved if a thinner slice was obtained with a more accurate CT machine.

4.4.3 Kinetic Error Analysis

Direct comparison of the reading from *FlexiForce*® Load/Force sensors and the predicted forces from our inverse mathematical model produced an average error of 5.31% with a maximum error of 9.42% (Figure 4.43).

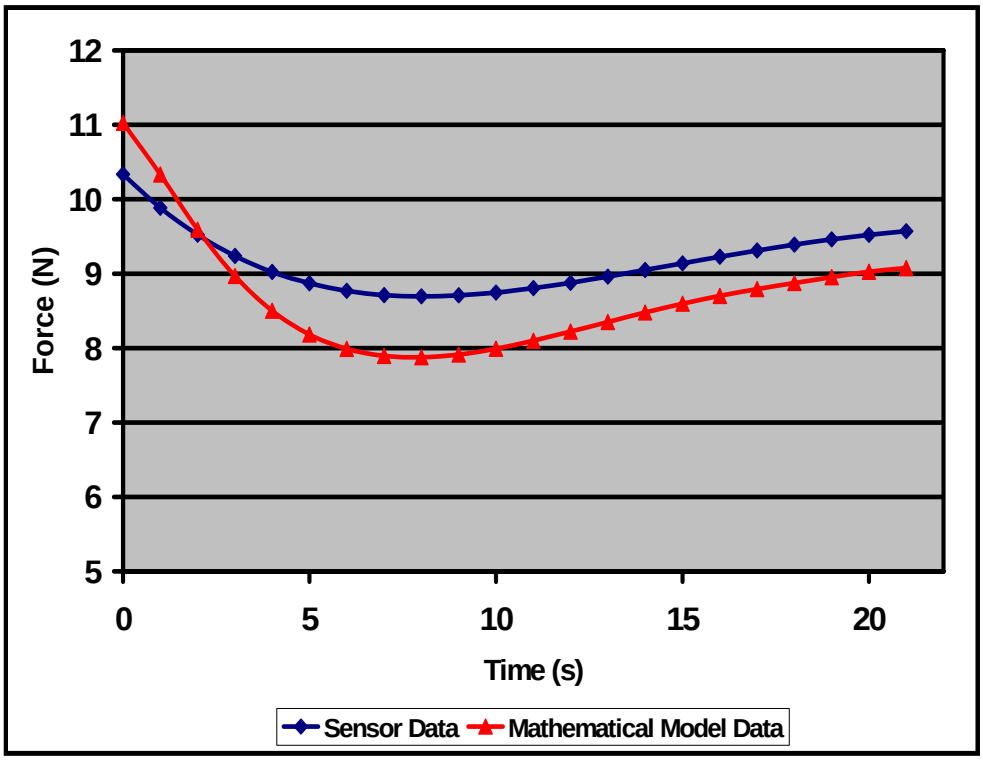


Figure 4.43 Kinetic comparison between predicted vertical forces and the reading from the *FlexiForce*® sensor.

Chapter 5 Discussion

Fluoroscopy and inverse dynamics methods have been successfully used in analyzing *3D in vivo* kinematic and kinetic characteristics of different joints in humans for more than a decade at CMR at the University of Tennessee, Knoxville, especially in the knee and hip fields[80, 81]. Cadaver and human experiments have been performed using artificial knee implants to validate the accuracy of these methods. Predicted kinematic results have also been obtained using fluoroscopic techniques and a 3D-to-2D registration method which allowed for comparison with experimental data from an Optotrak system. Predicted kinetic results were also calculated using a mathematical modeling method and compared with data directly from a telemetric knee implant [81-84].

However, these methods have not been widely used in the spine field, especially the cervical spine, likely due to the more complex anatomical structure and more intricate motion patterns of the cervical spine as compared to knee and hip joints. Therefore, this chapter has been presented in order to discuss the 2D and 3D *in vivo* kinematics and kinetics in the cervical spine from the human experiments and the results from an error analysis in the cadaveric experiment. The characteristics of relevant experimental and modeling techniques have been discussed in this chapter as well.

5.1 Kinematics Analysis

Even though numerous *in vivo* and *in vitro* studies have been performed to describe the clinical and biomechanical characteristics of the normal, degenerative,

ACDF, and CADR subjects, few of them have analyzed the *in vivo* motions among these groups in a single study. The 2D study was initially conducted to derive baseline data and provide the relationship between these groups. The ACDF and CADR surgeries were performed at the C5-C6 level, and this study analyzed the inferior (C6-C7) and superior (C4-C5) adjacent levels, and also the C3-C4 level.

5.1.1 2D Kinematics

Many studies have published the *in vivo* motion of the different cervical spines by using radiography, computerized tracking systems, or 3D movement analysis devices [24, 70, 85-88]. Normally, many above studies have also included the ROM of the skull as part of the cervical ROM (CROM). An article by Ordway *et al* measured the ROM between the occipital bone and C7 and defined this as the CROM [86]. Since we were only interested in the motion pattern of the cervical spine, this study did not include the motion of the skull or the trunk. In order to describe the movements of the cervical spine, two different types of rotations were defined in this study. Cervical spine rotation was used to identify the difference between θ1 and θ7, which represented the motion of the entire cervical spine. The value for the ROM of the cervical spine in this study was less than that in some previous studies, which had included the motion of the skull. The normal subjects had the largest average ROM of the cervical spine, followed by the CADR, degenerative, and ACDF subjects. Neck rotation was also defined in this study as the absolute angle of θ1. It helped us to easily determine the position and rotation of the neck. The contact forces at each level were plotted with respect to these angles in order to associate the forces with the neck rotation.

Much of the previous literature had reported the intersegmental ROM of the cervical spine for the normal, degenerative, ACDF, and CADR groups separately [23, 27, 60, 62, 63]. Picket *et.al* performed a follow-up study of up to two years with a total of 20 subjects (eight single CADR at C6-C7, eight implanted at C5-C6 and four with two level implantation) having either single level or two-level CADR and reported that the overall sagittal motion from C2 to C7 in the CADR was 56.1° after late follow-up [62]. In the present study, the overall sagittal motion from C1 to C7 was 76.7° and the average intersegmental ROM at the C1-C2 level was 16.7±7.4°. The ROM from C2 to C7, therefore, was similar to that in Picket's study. Additionally, Picket reported the observed sagittal rotation values after a minimum 6-month follow-up (Table 5.1). The present study found that CADR could preserve the normal intersegmental ROM, even at the implanted level. This result is contrary to Picket's study, which demonstrated CADR normally reduced the intersegmental ROM at the implanted level. Comparison of our results with the intersegmental ROM only for the patients having CADR at the C5-C6 level in Picket's study revealed that the intersegmental ROMs in this present study were larger at all levels. Our results also demonstrated that the intersegmental motion in the CADR subject was close to those in the normal CROM, which was similar to data reported by Panjabi and Dvorak [23, 63].

Table 5.1 Comparison of intersegmental rotation in the present study and Picket's.

Flex-Ext(Degree)	C1-C2	C2-C3	C3-C4	C4-C5	C5-C6	C6-C7
Liu	16.71±7.39	14.34±4.24	17.37±5.44	19.25±7.69	19.80±6.96	13.88±3.91
Picket	NA	6.8	11.4	15.3	11.9	10.6
Picket*	NA	6.6±2.52	12.66±3.46	16.38±4.90	9.46±4.18	13.13±3.97

* Only Implanted at C5-C6.

Many researchers had published the *in vivo* intersegmental ROM from C2 to C7 for the normal cervical spine [23, 71, 86, 89, 90]. As Picket mentioned, their results varied throughout a wide range. So, it is difficult to compare the *in vivo* intersegmental CROM from different studies. Therefore, this single study provided more significant information related to the normal, degenerative, ACDF, and CADR groups. In the present study, it was clear that the CADR group had no statistically different for intersegmental ROM from C3 to C7 as compared with the normal subjects ($P>0.05$). Overall, at these levels, the ACDF group had significantly less intersegmental ROM at almost every level when compared to the normal subjects ($P<0.05$). However, at the comparable neck ROM, say from 20° flexion to 15° extension, the ACDF group had a larger ROM at both adjacent levels (C6-C7 and C4-C5) but not at the C3-C4 level when compared with the normal group. In other words, when simply considering the entire flexion-extension activity, the intersegmental ROM of the normal group was typically larger than the corresponding intersegmental ROM of the ACDF group at both adjacent levels. This occurred because the normal group had a much larger ROM of the entire cervical spine, which was due to the larger ROM for the entire neck. However, if the ROM of the neck was limited to most daily activities such as walking, reading and eating, it was clear that both the adjacent levels in the ACDF group required a larger motion to reach the same neck position of the normal group. On the other hand, the average ROM at the C6-C7 and C4-C5 levels were very similar between the normal and ACDF group even though the CROM of the normal group was 80.56±6.40° as compared with 46.53±14.55° for the ACDF group. The intersegmental ROM at adjacent levels in the CADR group was comparable to those in the normal group. The degenerative group had

relatively smaller intersegmental angles compared to other groups, which may have been related to pain and discomfort in the neck leading to a loss of motion among the degenerative subjects.

It has been reported that CADR can restore normal motions of the cervical spine. In this study, the CADR group had similar average intersegmental ROM and motion patterns when compared to the normal group. At the C5-C6 level, which was the implanted level, the intersegmental ROM for the CADR group (19.80±6.96) was larger than the normal group (18.13±3.43). However, the intersegmental ROM at both adjacent levels in the CADR group was less than that of the normal group. Puttlitz *et al* evaluated the FE ROM at the implantation level (C4-C5) after CADR using 6 cadaveric cervical spines and reported a similar result with the ROM at the C4-C5 level being 14.0±3.4° in CADR and 9.4±2.2° in the intact cadavers [60]. Puttlitz did not report on the intersegmental ROM at levels adjacent to the implant. Even though there was no statistical difference between the two groups, further analysis should be given to this observation because it is assumed that a larger intersegmental motion at the implanted level could lead to larger contact forces in the transverse direction and smaller contact forces in the vertical direction. This force pattern could produce a loosening of an artificial disc implant in the future.

5.1.2 3D Kinematics

Coordinate System

In the spine field, previous studies setup the origin of a coordinate system to be at the posterior-inferior corner of a vertebral body [1, 19, 22] (Figure 5.1). Different

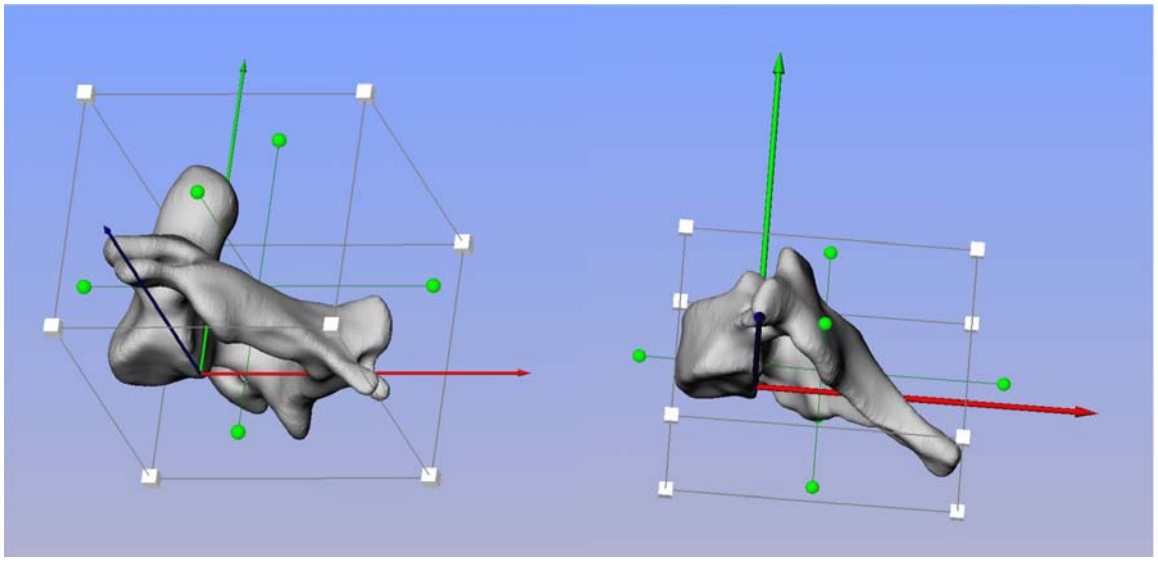


Figure 5.1 The origin for a coordinate system was chosen at the posterior-inferior corner of a vertebral body in many previous studies.

researchers could select different point as the origin of the same vertebra. It is also likely that the same researcher could pick a different point to define the same origin. In this study, the origins were automatically chosen by using the bounding box method (Figure 3.12 and Figure 3.13). Through the use of commercial 3D CAD softwares, a bounding box was created for a 3D CAD model of a vertebra. This bounding box enclosed the whole vertebra while maintaining a minimum volume. In effect, no matter where the location of the origin was initially on the vertebra, it was accurately moved to the center of the bounding box because the coordinates of its eight corners were known by the computer.

As a result, the coordinate system was more sensitive to abnormal structures as compared with the traditional methods. For instance, if there was a bony fracture at the end of a posterior process, the software would adjust the bounding box and automatically redefine the origin of the coordinate system. Moreover, the kinematic data describing the vertebra probably would be changed as well. However, it has been difficult for the traditional methods to observe this kind of abnormal structures. Because of the different methods to setup of the origins, the intersegmental translations in this study might be larger than previous studies. This might be due to fact that the origins in this study were further from the helix axes than the aforementioned traditional method. This reason is explained in Figure 5.2. Consider two cases: Case 1: where the origins are setup at O1 and O2 for two vertebrae; Case 2: where the origins are setup at O1' and O2', respectively. During the FE motion, there may have an intersegmental AP motion between these two vertebrae. However, it is barely noticeable in Case 1, although it is much more evident in Case 2. Because different origins were chosen in this study, the translations in the 3D kinematic analysis of FE, AR and LB were larger than those in the previous studies.

3D Flexion-extension

In the 2D kinematic study, the ACDF and normal groups had a similar intersegmental ROM at the adjacent levels for the entire flexion-extension motion. This might be due to the normal group having a relatively larger ROM of the entire cervical spine

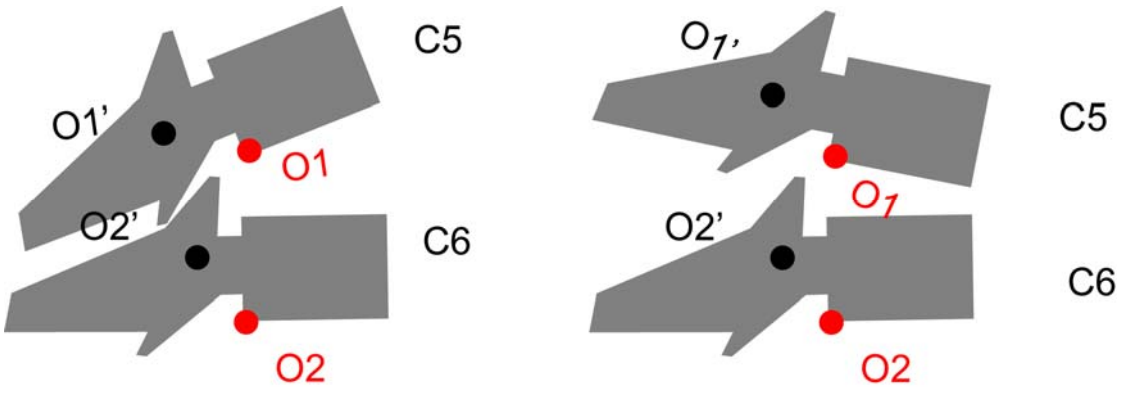


Figure 5.2 Different setups of origins of a coordinate system at two vertebrae.

when compared with the ACDF group. Considering the ROM of the neck from 20º of flexion to 15º of extension, the ACDF group exhibited a larger intersegmental ROM at the adjacent levels. At the inferior level (C5-C6), there was a statistical difference between the normal and ACDF groups, but at the superior level (C4-C5), the difference was not significant.

In the 3D study, subjects were randomly selected without having any knowledge of each subject's ROM of the cervical spine before experiments. In spite of this, all of the subjects were found to have a similar ROM of the cervical spine. Thus, it was unnecessary to further normalize the results in the 3D kinematic analysis when they were compared among groups.

In this 3D experiment, not only was the main FE motion analyzed, but also the coupled AR and LB in horizontal and frontal planes were thoroughly evaluated. Overall, the kinematic results in the sagittal plane matched those results in our previous 2D study.

At adjacent levels, the ACDF subject not only had a larger intersegmental ROM in the sagittal plane, but also in the other two planes as compared with the normal subject. Intersegmental translations in the AP, SI, and LT directions at the C6-C7 level in the ACDF patient were greatly increased as compared with the normal subject. Moreover, intersegmental translations in the AP and SI directions were also increased at the C4-C5 level in the ACDF patient. Relatively smaller intersegmental rotations were found at the symptom level (C5-C6) in the degenerative patient, which was consistent with the findings of Dvorak *et al* [91]. Contrary to Dvorak's study, findings revealed that the intersegmental rotations and translations at the adjacent levels in the degenerative patient were relatively larger compared to those in the normal subject. This could be attributed to the degenerative patient having an abnormally large ROM for her group which enabled her to attain a similar CROM as that in a normal subject. It is hypothesized that in order for the degenerative subject to reach this large CROM, adjacent levels may have to compensate for the loss of motion at the degenerative level [91].

Dvorak *et al* used X-rays to perform a human experiment having 44 asymptomatic subjects (average age=31.6 years old, 22 male *vs.* 22 female) [24]. The study reported the intersegmental translations of the superior and inferior corners on each vertebral body in the AP and SI directions. The published data contained values which were much smaller than those obtained in this 3D study. One reason for this difference was the different choose of the origins of the local coordinate systems in these two studies. Secondly, Dvorak *et al* reported that human interaction was needed in order to pick up the four corners of a vertebral body in the study[24]. Therefore, results were possibly dependent on the researcher. Different researchers produced varying results. In

this study, a computer automatically created a bounding box around a vertebra, even though it was still need human interaction to adjust the X-Y-Z directions. It is allowed for more accuracy compared to previous studies. Although this approach was different from the traditional analysis method having been set forth in the cervical spine field, it might be a better alternative. Current image processing techniques could finish the 3D-to-2D registration procedure with minimal human interaction. It is also possible that in the near future an automatic segmentation method could reconstruct 3D CAD models of the cervical spine quickly and accurately. Thus, it will be possible to completely automate the entire procedure of the 3D kinematic analysis in this study and become a powerful diagnostic method in the future.

3D Axial Rotation

Many research studies have documented the *in vivo* and *in vitro* kinematic characteristics of AR in the cervical spine [19, 20, 30, 35, 92-95]. The *in vitro* experiments have reported the mechanical relationship between the applied force/moment and displacement/rotation of the tested FSU or the whole cervical spine during AR [19, 95]. Lysell studied AR as early as 1969 [95]. More recently, Panjabi *et al* performed cadaveric analysis using sixteen complete cervical spine specimens[19]. Coupled motions were observed in Panjabi's study as well as a coupled translation in the lower cervical spine. All these data were relatively smaller compared to the results from previous *in vivo* studies referenced here and in the present study (Table 4.4). It could be assumed that the lack of muscular structures may be one reason for their differing results. Another reason would be the application of a pure moment. In his study, it mentioned

that a 1 Nm moment had been proven to recreate physiologic motions without creating injury to the specimen. However, this applied moment may not create the same ROM as a living subject would produce in normal activities.

In vitro data, from previous studies, have been found to provide important information concerning the AR. However, *in vivo* studies may offer more valuable information. Since the methods applied during *in vivo* studies were different, the results from various studies varied greatly. For example, early studies only reported the main intersegmental overall rotation in the horizontal plane during AR [30, 92]. Dvorak and Penning examined normal subjects using CT images and reported results (Table 4.3), which were larger than those in the present study. In Penning's study, the age of the normal subjects was in the mid-twenties. This might have allowed for a larger ROM compared to the subjects in the present study, whom were in their mid-thirties. Most recently, Ishii *et al* used MRI images to conduct a human experiment study having 10 normal subjects (5 male *vs.* 5 female; average age: 25.1 years). MRI images of the entire cervical spine at eleven different positions were recorded during the axial rotation. They reported the main overall rotation, coupled rotations and translations in three directions in detail. The normal subject in the present study had similar main and coupled motion as Ishii's study, though some differences did exist. He reported 108.7% coupled LB and 32.6% coupled EF at the C4-C5 level; 132.5% coupled LB and 22.5% coupled EF at the C5-C6 level; and 306.3% coupled LB and 150% coupled EF at the C6-C7 level. However, the data from Ishii's study only described the kinematic characteristics under a pseudo-dynamic situation. The present study provided the state-of-art method to explore the 3D *in vivo* kinematics in the cervical spine under a fully dynamic situation. Differing

from Ishii's study, middle age females were selected in this study. According to the suggestion from the surgeon, this age population was more likely to have degenerative symptoms in the cervical spine and need an operation, which would allow the present study to produce data more useful in the clinical field.

The ACDF patient demonstrated a larger main overall rotation at both adjacent levels. In the present study, the AR in the ACDF patient was 233.1% (11.75º *vs*. 5.03º) larger than that in the normal subject at the C4-C5 level and 191.1% (10.26º *vs*. 5.37º) larger at the C6-C7 level. The magnitudes of coupled LB in the ACDF patient were much larger than those in the normal subject. However, the percentages of the coupled LB with respect to the main AR in the ACDF subject were not much larger than those in the normal subject, at both the C4-C5 (128.9% *vs*. 244.9%) and C6-C7 (128.9% *vs*. 115.6%) levels. Both the magnitudes and percentages of the coupled FE in the ACDF subject were much smaller than those in the normal subject at both the C4-C5 (4.98 *vs*. 5.28; 42.4% *vs*. 104.9%) and C6-C7 (0.88 *vs*. 2.45; 8.6% *vs*. 45.6%) levels. The degenerative patient had relatively smaller intersegmental overall rotation for both the main and coupled motions compared to the normal and ACDF subjects. Also as mentioned above, the percentages of LB in the normal and ACDF subjects were larger than 100% at C4 through C7 levels. However, all the percentages of the coupled LB in the degenerative subject were less than 100% from C4 through C7 and were different from those in the normal and ACDF subjects.

3D Lateral Bending

Ishii *et al* also performed other human experiments (6 male *vs.* 6 female; average age: 23.6 years) using MRI [22]. Ishii *et al* recorded MRI images of the cervical spine at seven static positions during lateral bending. A 3D model of each vertebra was then reconstructed according to MRI images, and the intersegmental motions were analyzed using these static 3D models. They reported 33.33% coupled AR and 24.24% coupled FE at the C4-C5 level, and 27.91% coupled AR and 16.28% coupled FE at the C5-C6 level. Therefore, in the present study, in terms of the percentages of coupled motions during LB, the ACDF patient did not have increased coupled rotations at the adjacent level, C4-C5, when compared with Ishii's study. For the degenerative subject, the magnitudes and percentages of coupled motions were smaller at the C4-C5 level and similar at the C5-C6 level. Ishii reported small intersegmental translations in all three directions (<1mm), which were different from those in the present study. This might be due to the different methods used to setup the local coordinate systems.

Many *in vitro* studies have documented the motion patterns and mechanical properties of the cervical spine [19, 96, 97]. These studies are valuable references for further *in vivo* studies. Panjabi *et al* tested sixteen complete cadaveric cervical spines under pure moment loading [19]. During LB, a coupled motion was found at the C4-C5 level, resulting in about 4º (43.01%) of axial rotation coupled with 9.3º main rotation (LB). Another study tested ten cadaver cervical spines under three different situations: intact, injured and ACDF with an anterior metal plate [97]. The study reported up to 36.67% coupled motions during right lateral bending in the intact cadaver cervical spines. It also reported up to 41.3% increased motion for injured cervical spines and 45.9%

reduced motion for ACDF cervical spines when compared to the intact specimens. However, in the present *in vivo* study, decreased motion was observed in the degenerative subject. This difference might be that the *in vitro* study was not able to describe the real kinematics occurring inside the human body because it lacked essential soft tissues, which offer a significant degree of stabilization to the cervical spine. As a result, *in vivo* studies may be assumed to be different from *in vitro* studies but not significantly.

5.2 Kinetics Analysis

To the author's best knowledge, few studies have quantified *in vivo* contact forces and compared them among normal, pathological, and postoperative cervical spines. Since FE is the major activity for the cervical spine, the current mathematical model was designed to predict *in vivo* contact forces for this motion. On the other hand, after the current 3D kinematic method is improved and could obtain 3D kinematics of the entire cervical spine for AR and LB activities, this 3D mathematical model could be used to predict kinetics for these two activities as well.

Since the FE motion in the sagittal plane has lower out of plane coupled motions when compared with AR and LB, a 2D inverse dynamic model could be used to initially calculate the contact forces in the cervical spine. In the 2D study, the ACDF group statistically had the largest anteroposterior forces as well as the largest pseudo-muscular forces at the adjacent levels during the FE activity when compared to the normal, degenerative, and CADR groups. However, there was no statistical difference for the compression forces among these four groups, even though the magnitudes in the ACDF group were smaller when compared to the normal and CADR groups. At the C3-C4

level, there was also no significant difference among these groups. On average, the degenerative cervical spine had the smallest contact and pseudo-muscular forces at all levels during the entire FE activity. According to the present study, the forces at the adjacent levels after ACDF were statistically higher than the forces in the normal, degenerative, and CADR groups during the FE, especially anteroposterior forces. This could serve to be a contributor to the accelerated degeneration that has been reported at adjacent levels in previous studies.

Previous *in vitro* studies have reported an increased pressure at adjacent levels after ACDF [55, 61, 98]. Eck *et al* reported an increase of intradiscal pressures (IDP) at both the superior and inferior adjacent levels during flexion of 73.2% and 45.3%, respectively [55]. Also reported were increased IDPs without statistical significance during extension. More recently, Dmitriev *et al* compared IDPs under four different conditions in a single study by testing 10 cadaveric cervical spines [61]. They reported that the recorded IDPs were statistically similar at both adjacent levels between the intact condition and a total disc replacement reconstruction. These recorded IDPs were found to be statistically less than those in both the allograft dowel and allograft-anterior cervical plate conditions [61].

The normal group showed smooth force patterns during the FE activity (Figure 4.5). Comparatively, the ACDF group demonstrated variable force patterns at the beginning and end of the motion and had larger magnitudes than those for the other groups. Fluoroscopic images were captured directly from the fluoroscopic video with equal time intervals. After normalizing the motion with respect to the neck rotation, it was found that the ACDF group took more time to start and stop the motion as compared

to the normal group (Figure 4.5). The force patterns of the degenerative and CADR groups were similar to the normal group.

This 2D mathematical model provided a new approach to analyzing *in vivo* contact and pseudo-muscular forces. Nevertheless, the model does have limitations. First of all, ligaments and muscles are crucial components, but due to static determinacy requirements, they were not modeled separately and, instead, were included as a unified pseudo-muscular force. Also, accurate muscular attachments could not be determined using only 2D images. In addition, the directions of pseudo-muscular forces were simplified to be straight lines between the origin and insertion points which are much more complex in real life. Despite the above limitations of the 2D dynamic model, it still offered a new method to solve complex problems related to muscular forces by switching the pseudo-muscle groups during the motion.

This 2D model was further developed into a 3D model that includes more accurate kinematic data. The 3D mathematical model in this study more accurately predicted *in vivo* contact and muscular forces in the cervical spine by incorporating more than five major types of ligaments around the lower cervical spine. The material properties of these ligaments were gathered from the latest and most reliable cadaveric studies. The 3D CAD model of each vertebra was rebuilt by using CT images of each subject. These 3D models were used along with associated MRI images to directly determine the ligamentous attachment points. The 3D kinematics were found by applying the 3D-to-2D registration method and were then input into the 3D mathematical model.

In the 2D study, a total of forty subjects were analyzed with the normal, degenerative, CADF and ACDF groups each containing ten subjects. In the 3D study, three subjects were analyzed with only one subject in each of the three groups. As a result, the predicted forces in the 3D study were not assumed to be the same as those in the 2D study. However, the results from the 3D study were within a reasonable range of the results from the 2D study. In the 3D study, at the adjacent levels (C6-C7 and C5-C6), both compression and anteroposterior forces in the ACDF subject were much larger than those in the normal and degenerative subjects. In the 2D study, at the adjacent levels, the anteroposterior forces in the ACDF group were statistically larger than those in the normal group. This was consistent with the result from the 3D study. At these same levels, the compression forces in the ACDF group were a little larger than those in the normal group, even though there was no statistical difference. However, according to previous *in vitro* studies, the compression forces at the adjacent levels in the ACDF group should be larger than those in the normal group. So, the results from the 3D mathematical model were more consistent with those from *in vitro* studies. Following careful analysis of the 2D study, it was found that the ROM of the cervical spine in the ACDF group was much smaller than those in the normal group. Adversely, the 3D study resulted in the normal and ACDF subjects having a similar ROM of the cervical spine. Therefore, it was hypothesized that the compression forces in the ACDF group could be larger than those in the normal group if the ACDF group had the similar ROM as the normal group in the 2D study. In terms of the curve shapes representing the compression forces, results from the 3D study showed a more reasonable loading and unloading characteristic during the FE activity. Also according to the 3D study, there were no

significant differences at the C3-C4 level for both the compression and anteroposterior forces between the normal and ACDF subjects. In addition, the normal and ACDF subjects exhibited similar magnitudes of the lateral forces, The contact forces at each level were relatively smaller in magnitude for the degenerative subject as compared to the normal subject.

Moroney *et al* used a biomechanical model of the neck to predict the compression and forces at the C4 level [99]. In total, fourteen subjects were involved in his study. The applied external forces or moments were determined by evaluating the maximum physical forces and moments the subjects could sustain without causing injury. It was reported that the maximum compression and anteroposterior force during extension were approximately 1164 N and 135 N, respectively. The maximum lateral force during LB was about 125 N. Moroney's study predicted the maximum forces that would occur under extreme situations. In this study, the actual FE activity was studied without applying any external loads. Hence, the compression and forces were predicted to be less than the reported data in Moroney's study, if his results were correct. In the present 3D study at the C4-C5 level, the predicted maximum compression forces in the normal, degenerative and fused subjects were 96.5 N (1.7 SW), 106.1 N (1.6 SW) and 130.9 N (2.91 SW), respectively. Each of the maximum values occurred during extension and exhibited much smaller magnitudes than those reported in Moroney's study. In addition, the results of the lateral and anteroposterior forces among these three subjects were all much smaller than those in Moroney's study. Therefore, we were able to conclude that the results in the present 3D study were consistent with those in Moroney's study to an extent.

With the added knowledge of how forces change in response to cervical fusion, a strategy could be developed to minimize the risk of post-operative accelerated degeneration in the future. However, according to the findings of this research and when considering only the biomechanics of the cervical spine, it has been demonstrated that CADR could be a better alternative to ACDF for the reason that CADR can preserve the motion and force patterns of the normal cervical spine. Nevertheless, long-term follow-up results and clinical studies of CADR should be performed to further demonstrate whether or not CADR can replace ACDF as the next gold standard surgical method in the cervical spine field.

5.3 Error Analysis Experiment

Drs. Mahfouz and Komistek performed a cadaveric error analysis experiment for FE using artificial knee implants [84]. The results from this analysis, after eliminating the system error from the Optotrak system, are listed in Table 5.2. The means for both translations and rotations were much smaller than those in the present study (Table 4.11) except for the translation in the Z direction.

Table 5.2 Mean and SD for flexion-extension of the cadaveric knee experiment by Drs. Mahfouz and Komistek[84].

	Mean	SD
X translation	0.023 mm	0.473 mm
Y translation	0.086 mm	0.449 mm
Z translation	1.054 mm	3.031 mm
X rotation	0.068º	0.942º
Y rotation	0.001º	0.771º
Z rotation	0.253 º	0.841º

In addition, a human experiment for FE with artificial knee implants was performed by Drs. Mahfouz and Komistek [81]. Table 5.3 lists the RMS of translations and rotations in this human error analysis. All their results were much smaller than those in this study (Table 4.11) with exception to the Z translation once again.

Artificial knee implants were used in both of the above kinematics error analysis experiments, but obviously, the shape of a knee implant is much simpler than that of the cervical spine (Figure 5.3). The 3D CAD models in this cadaveric cervical spine experiment were constructed directly from CT images utilizing the same procedures followed in the human experiment, whereas the 3D CAD models of the artificial knee implants in the Mahfouz and Komistek study came either directly from companies or from laser scanning [81, 84]. These two approaches produced more accurate 3D CAD models than the method used for creating models in the present cadaveric cervical spine experiment. As was mentioned before, the thinner the CT slice increment (currently 0.33mm), the more accurate the current method can be not only in the cervical spine field but also in all the normal joint fields.

Table 5.3 Root Mean Square (RMS) of translations and rotation for flexion-extension in human error analysis [81].

	X translation	Y translation	Z translation	X rotation	Y rotation	Z rotation
RMS	0.080 mm	0.044 mm	1.367 mm	0.145º	0.157º	0.338º

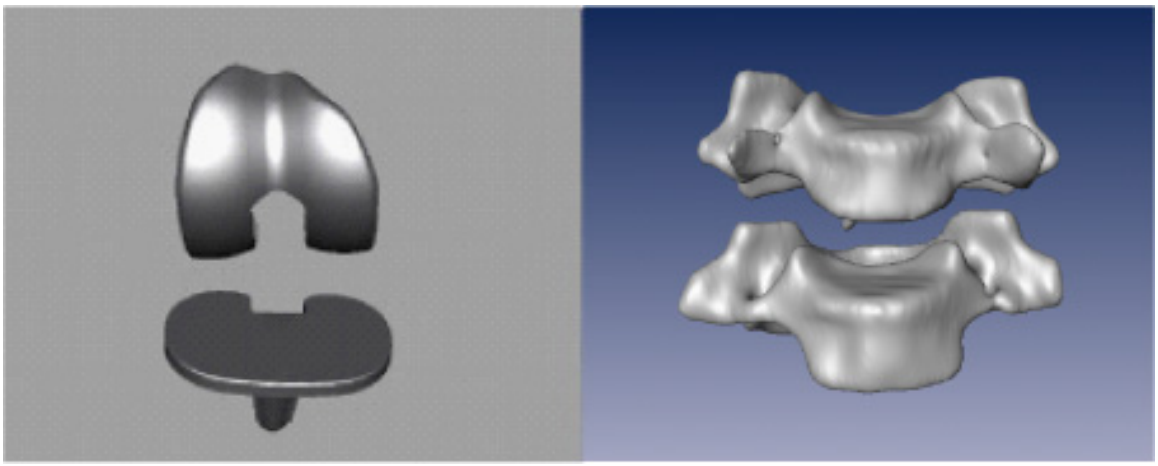


Figure 5.3 Shape comparison between artificial knee implant and vertebrae of the cervical spine.

Only 3D poses at stationary positions were studied for the error analysis experiments of the artificial knee implants [81, 84]. Additionally, only one image was analyzed in the human experiment [81]. The cadaveric cervical spine experiment simulated the actual conditions of the human experiments. Initially, it followed the actual loading patterns in the cervical spine for natural activities. During a usual motion, the load on the normal cervical spine comes directly from the head. In order to replicate this motion in the cadaveric experiment, the load was only applied to the top of the cadaveric cervical spine. Secondly, the real dynamic motions of the cervical spine during a motion were simulated in this experiment. This was accomplished by simultaneously recording the real-time fluoroscopic video and the MicronTracker data of FE, AR, and LB. This allowed for the predicted motion and forces to be compared directly with the measured data. Thus, the present cadaveric cervical spine experiment was able to better quantify the errors in the human experiment compared to most of the previous studies.

On the other hand, only two components of an artificial knee implant -- femoral and tibial -- were studied in the previous error analysis experiments [81, 84]. In the current cadaveric cervical spine experiment, five components from C3 to C7 were studied simultaneously, which made the experiment more sophisticated.

Considering the above factors, it is reasonable to assume that the results of a spine study would not be as accurate as those in the artificial knee implant field.

Some researchers performed error analysis with their own experimental methods when studying the 3D *in vivo* motion of the cervical spine [20, 22, 93, 94]. Lai and Mimura *et al* reported the accuracy of their system with the average errors shown in Table 5.4. Ishii and Mukai *et al* also performed two 3D *in vivo* human experiments which were very accurate [20, 22]. The results are listed in Table 5.5. Their error analysis study was performed using a MRI phantom which had markers with four high precision ceramic balls and used to determine the "gold standard". The whole phantom was then scanned at different positions and compared with the "gold standard". This system seemed have a higher accuracy than the present method, especially for rotation. Panjabi *et al* conducted a study and reported their error analysis for the cadaver experiment by testing a static rigid body [19].

Table 5.4 Accuracy of the *in vivo* experiments in literature [93, 94].

	Axial Rotation		Flexion Extension		Lateral Bending	
	Translation	Rotation	Translation	Rotation	Translation	Rotation
Error	1.0mm	1.5º	1.0mm	3.0º	1.0mm	4.5º

Table 5.5 3D in vivo error analysis results from Ishii's studies [20, 22].

	Rotation			Translation		
	Axial Rotation	Flexion Extension	Lateral Bending	Supero-inferior	Antero-posterior	Lateral
Mean	0.43º	0.24º	0.31º	0.52mm	0.51mm	0.41mm

In spite of the numerous studies mentioned, most previous error analysis experiments in the spine field were too simple to quantify the actual errors when used in the human experiments. So, the present cadaver experiment was more sophisticated than previous error analysis experiments and better simulated the actual situation in human experiments. As a result, the magnitudes from different experimental methods could not simply demonstrate that the present 3D kinematic method was less accurate than some current methods in the spine field.

Since the cadaver used in the experiment excluded all the muscular structures, the fluoroscopic images were clearer than those directly from the human experiment. This was especially true when the skull blurred the vertebrae close to it, such as C1, C2 and C3, during AR and LB activities in the human experiment. Hence, the errors in the human experiment were expected to be higher, but not significantly, than those in the cadaver experiment.

The cadaveric study revealed that the kinematics derived using fluoroscopy and the kinetics predicted using our mathematical model, were consistent with the experimentally derived results. The cadaver error analysis demonstrated that the current fluoroscopic and inverse dynamic modeling methods could be used to quantify 3D *in vivo* dynamics of the cervical spine.

Chapter 6 Limitations and Future Work

This project focused on analyzing the 2D and 3D *in vivo* kinematic properties of the normal, pathological and postoperative cervical spines. Furthermore, it developed 2D and 3D mathematical models that could be used to help us better understanding *in vivo* kinetic characteristics of these different cervical spines. The cadaveric error analysis experiment validated that the 3D-to-2D registration method and inverse dynamic modeling method based on Kane's dynamics were accurate when applied in the cervical spine field.

6.1 Limitations

However, this study does have some limitations and needs to be improved in the future work. Firstly, only three subjects, one for each group, were involved in the 3D *in vivo* kinematic and kinetic analysis. In spite of these three subjects having been randomly selected, the limited subject number cannot provide solid evidence that can only be achieved by involving more subjects in each group. Additionally, CADR subjects were not involved in this 3D study because no CADR clinical trials were being conducted at the time of this study at VUMC. However, 3D results for this group are very important for the whole study.

Secondly, the vertebrae were assumed as rigid bodies in this study. Since the intervertebral discs do not simply behave as rigid bodies, a more accurate method could be used to model these structures. Useful information concerning the behavior of rigid

bodies, fluid dynamic and finite element analysis (FEA) studies should be involved in this future work to provide more accurate results.

Thirdly, all the ligamentous and muscular attachment points were manually picked up from either 3D CAD models of the cervical spine or CT and MRI images according to previous research references. This was a very time-consuming process and does not currently allow the present method be used on a large subject population. One suggestion is to build up a database of the anatomical characteristics of the cervical spine, and to utilized automatic computation methods to complete these jobs automatically with minimum human interaction.

Finally, the cervical spine has very complex anatomical structures, which makes it difficult to overlay 3D CAD models onto 2D fluoroscopic images compared to other joints such as the knee and hip. Hence, more advanced techniques such as bi-plane fluoroscopic images should be utilized in the future.

Even though the above limitations were present in this research study, it does provide many unique and important research results related to the dynamic characteristics about the entire cervical spine. Most importantly, to the best of the author's knowledge, this study was the first to quantify 3D *in vivo* kinematics and kinetics of the entire cervical spine among different subjects under truly dynamic situations.

6.2 Future work

Even though this study has provided two reliable mathematical models and one complicated cadaveric error analysis experiment, many issues remain pertaining to mathematically modeling the cervical spine.

Firstly, a sensitive analysis should be included into in the future study. Choice of different major muscles and changing of muscular and ligament attachments may make results difference.

Secondly, a more complicated 3D mathematical model, which include FE, AR, and LB should be created. In this future model, a main program should be able to automatically switch to three different sub-programs –FE, AR, and LB- according to the rotation of the skull. For each sub-program, it should include two programs including different major muscles and ligaments. For sample, two programs should be included in the FE sub-program; one program describes the motion from an extension position to the neutral position; the other describes the motion from the neutral position to a flexion position.

Finally, a 3D *in vivo* error analysis study should be performed with telemetric implants to help us better understanding the accuracy of the 3D-to-2D kinematic method and the 3D *in vivo* mathematical model.

List of References

References

1. White, A.A., *clinical biomechanics of the spine*. c1990, Philadelphia:Lippincott.
2. Yoganandan, N., S. Kumaresan, and F.A. Pintar, *Biomechanics of the cervical spine Part 2. Cervical spine soft tissue responses and biomechanical modeling.* Clin Biomech (Bristol, Avon), 2001. 16(1): p. 1-27.
3. Knudson, D.V., *Fundamentals of Biomechanics* Jun 2003: Springer Verlag. 340
4. Bovim, G., H. Schrader, and T. Sand, *Neck pain in the general population.* Spine, 1994. 19(12): p. 1307-9.
5. Brattberg, G., M. Thorslund, and A. Wikman, *The prevalence of pain in a general population. The results of a postal survey in a county of Sweden.* Pain, 1989. 37(2): p. 215-222.
6. Hibbs, R.A., *An operation for progressive spinal deformity.* NY Med J 21 1911: p. 1013–1016.
7. Albee, F.H., *Transplantation of a portion of the tibia into the spine for Pott's disease.* J Am Med Assoc 57 1911: p. 855–858.
8. Bono, C.M. and S.R. Garfin, *History and evolution of disc replacement.* Spine J, 2004. 4(6 Suppl): p. 145S-150S.
9. Casey K. Lee, N.A.L., *A review of spinal fusion for degenerative disc disease: need for alternative treatment approach of disc arthroplasty?* . Spine 2004. 4: p. 173-176.
10. Davis, H., *Increasing rates of cervical and lumbar spine surgery in the United States, 1979-1990.* Spine, 1994. 19(10): p. 1117-23; discussion 1123-4.
11. Rutkow, I.M., *Orthopaedic operations in the United States, 1979 through 1983.* J Bone Joint Surg Am, 1986. 68(5): p. 716-9.
12. Taylor, V.M., et al., *Low back pain hospitalization. Recent United States trends and regional variations.* Spine, 1994. 19(11): p. 1207-12; discussion 13.
13. Errico, T.J., *Why a mechanical disc?* Spine J, 2004. 4(6 Suppl): p. 151S-157S.
14. Abraham, D.J. and H.N. Herkowitz, *Indications and trends in use in cervical spinal fusions.* Orthop Clin North Am, 1998. 29(4): p. 731-44.
15. Greenough, C.G., L.J. Taylor, and R.D. Fraser, *Anterior lumbar fusion: results, assessment techniques and prognostic factors.* Eur Spine J, 1994. 3(4): p. 225-30.
16. Singh, K., A.R. Vaccaro, and T.J. Albert, *Assessing the potential impact of total disc arthroplasty on surgeon practice patterns in North America.* Spine J, 2004. 4(6 Suppl): p. 195S-201S.
17. Lee, C.K. and N.A. Langrana, *A review of spinal fusion for degenerative disc disease: need for alternative treatment approach of disc arthroplasty?* Spine J, 2004. 4(6 Suppl): p. 173S-176S.
18. Hilibrand, A.S., et al., *Radiculopathy and myelopathy at segments adjacent to the site of a previous anterior cervical arthrodesis.* J Bone Joint Surg Am, 1999. 81(4): p. 519-28.
19. Panjabi, M.M., et al., *Mechanical properties of the human cervical spine as shown by three-dimensional load-displacement curves.* Spine, 2001. 26(24): p. 2692-700.

20. Ishii, T., et al., *Kinematics of the upper cervical spine in rotation: in vivo three-dimensional analysis.* Spine, 2004. 29(7): p. E139-44.
21. Panjabi, M., et al., *Three-dimensional movements of the upper cervical spine.* Spine, 1988. 13(7): p. 726-30.
22. Ishii, T., et al., *Kinematics of the cervical spine in lateral bending: in vivo three-dimensional analysis.* Spine, 2006. 31(2): p. 155-60.
23. Dvorak, J., et al., *Functional radiographic diagnosis of the cervical spine: flexion-extension.* Spine, 1988. 13(7): p. 748-55.
24. Dvorak, J., et al., *In vivo flexion-extension of the normal cervical spine.* J Orthop Res, 1991. 9(6): p. 828-34.
25. Duerinckx, A.J., et al., *MR imaging of cervical spine motion with HASTE.* Magn Reson Imaging, 1999. 17(3): p. 371-81.
26. Karhu, J.O., R.K. Parkkola, and S.K. Koskinen, *Evaluation of flexion-extension of the upper cervical spine in patients with rheumatoid arthritis: an MRI study with a dedicated positioning device compared to conventional radiographs.* Acta Radiol, 2005. 46(1): p. 55-66.
27. Penning, L., *Normal movements of the cervical spine.* AJR Am J Roentgenol, 1978. 130(2): p. 317-26.
28. Takeshima, T., et al., *Sagittal alignment of cervical flexion and extension: lateral radiographic analysis.* Spine, 2002. 27(15): p. E348-55.
29. Dvir, Z. and T. Prushansky, *Reproducibility and instrument validity of a new ultrasonography-based system for measuring cervical spine kinematics.* Clin Biomech (Bristol, Avon), 2000. 15(9): p. 658-64.
30. Penning, L. and J.T. Wilmink, *Rotation of the cervical spine. A CT study in normal subjects.* Spine, 1987. 12(8): p. 732-8.
31. EH, W., *Anatomisch-physiologische Untersuchung über einige Einrichtungen im Mechanismus der menschlichen Wirbelsäule.* Arch Anat Physiol, 1827.
32. Luan, F., et al., *Qualitative analysis of neck kinematics during low-speed rear-end impact.* Clin Biomech (Bristol, Avon), 2000. 15(9): p. 649-57.
33. Moroney, S.P., et al., *Load-displacement properties of lower cervical spine motion segments.* J Biomech, 1988. 21(9): p. 769-79.
34. Panjabi, M.M., et al., *Three-dimensional load-displacement curves due to forces on the cervical spine.* J Orthop Res, 1986. 4(2): p. 152-61.
35. Ishii, T., et al., *Kinematics of the subaxial cervical spine in rotation in vivo three-dimensional analysis.* Spine, 2004. 29(24): p. 2826-31.
36. Panjabi, M.M., et al., *Cervical spine ligament injury during simulated frontal impact.* Spine, 2004. 29(21): p. 2395-403.
37. Nightingale, R.W., et al., *Inertial properties and loading rates affect buckling modes and injury mechanisms in the cervical spine.* J Biomech, 2000. 33(2): p. 191-7.
38. Komistek, R.D., et al., *Knee mechanics: a review of past and present techniques to determine in vivo loads.* J Biomech, 2005. 38(2): p. 215-28.
39. Rohlmann, A., et al., *Loads on an internal spinal fixation device during sitting.* J Biomech, 2001. 34(8): p. 989-93.

40. Graichen, F., G. Bergmann, and A. Rohlmann, *Hip endoprosthesis for in vivo measurement of joint force and temperature.* J Biomech, 1999. 32(10): p. 1113-7.
41. Wasielewski, R.C., D.D. Galat, and R.D. Komistek, *Correlation of compartment pressure data from an intraoperative sensing device with postoperative fluoroscopic kinematic results in TKA patients.* J Biomech, 2005. 38(2): p. 333-9.
42. Komistek, R.D., et al., *Mathematical model of the lower extremity joint reaction forces using Kane's method of dynamics.* J Biomech, 1998. 31(2): p. 185-9.
43. Anderson, F.C. and M.G. Pandy, *Dynamic optimization of human walking.* J Biomech Eng, 2001. 123(5): p. 381-90.
44. Piazza, S.J. and S.L. Delp, *Three-dimensional dynamic simulation of total knee replacement motion during a step-up task.* J Biomech Eng, 2001. 123(6): p. 599-606.
45. Bartolomei, J.C., N. Theodore, and V.K. Sonntag, *Adjacent level degeneration after anterior cervical fusion: a clinical review.* Neurosurg Clin N Am, 2005. 16(4): p. 575-87, v.
46. Bohlman, H.H., et al., *Robinson anterior cervical discectomy and arthrodesis for cervical radiculopathy. Long-term follow-up of one hundred and twenty-two patients.* J Bone Joint Surg Am, 1993. 75(9): p. 1298-307.
47. Chiles, B.W., 3rd, et al., *Cervical spondylotic myelopathy: patterns of neurological deficit and recovery after anterior cervical decompression.* Neurosurgery, 1999. 44(4): p. 762-9; discussion 769-70.
48. Hilibrand, A.S., et al., *The success of anterior cervical arthrodesis adjacent to a previous fusion.* Spine, 1997. 22(14): p. 1574-9.
49. Smith, G.W. and R.A. Robinson, *The treatment of certain cervical-spine disorders by anterior removal of the intervertebral disc and interbody fusion.* J Bone Joint Surg Am, 1958. 40-A(3): p. 607-24.
50. Brodke, D.S. and T.A. Zdeblick, *Modified Smith-Robinson procedure for anterior cervical discectomy and fusion.* Spine, 1992. 17(10 Suppl): p. S427-30.
51. Hunter, L.Y., E.M. Braunstein, and R.W. Bailey, *Radiographic changes following anterior cervical fusion.* Spine, 1980. 5(5): p. 399-401.
52. Wu, W., et al., *Degenerative changes following anterior cervical discectomy and fusion evaluated by fast spin-echo MR imaging.* Acta Radiol, 1996. 37(5): p. 614-7.
53. Goffin, J., et al., *Long-term follow-up after interbody fusion of the cervical spine.* J Spinal Disord Tech, 2004. 17(2): p. 79-85.
54. Alund, M. and S.E. Larsson, *Three-dimensional analysis of neck motion. A clinical method.* Spine, 1990. 15(2): p. 87-91.
55. Eck, J.C., et al., *Biomechanical study on the effect of cervical spine fusion on adjacent-level intradiscal pressure and segmental motion.* Spine, 2002. 27(22): p. 2431-4.
56. Fuller, D.A., et al., *A kinematic study of the cervical spine before and after segmental arthrodesis.* Spine, 1998. 23(15): p. 1649-56.
57. Gore, D.R., et al., *Roentgenographic findings following anterior cervical fusion.* Skeletal Radiol, 1986. 15(7): p. 556-9.

58. Gore, D.R. and S.B. Sepic, *Anterior discectomy and fusion for painful cervical disc disease. A report of 50 patients with an average follow-up of 21 years.* Spine, 1998. 23(19): p. 2047-51.
59. Goffin, J., et al., *Intermediate follow-up after treatment of degenerative disc disease with the Bryan Cervical Disc Prosthesis: single-level and bi-level.* Spine, 2003. 28(24): p. 2673-8.
60. Puttlitz, C.M., et al., *Intervertebral disc replacement maintains cervical spine kinetics.* Spine, 2004. 29(24): p. 2809-14.
61. Dmitriev, A.E., et al., *Adjacent level intradiscal pressure and segmental kinematics following a cervical total disc arthroplasty: an in vitro human cadaveric model.* Spine, 2005. 30(10): p. 1165-72.
62. Pickett, G.E., J.P. Rouleau, and N. Duggal, *Kinematic analysis of the cervical spine following implantation of an artificial cervical disc.* Spine, 2005. 30(17): p. 1949-54.
63. Augustus A. White III, M.M.P., *Clinical biomechanics of the spine* 2nd ed ed. c1990, Philadelphia Lippincott.
64. Netter, F.H., *Netter's Atlas of Human Anatomy*. 3rd ed. 2002: Icon Learning Systems.
65. Winter, D.A., *Biomechanics And Motor Control of Human Movement* 3ed. 2004: John Wiley & Sons Inc.
66. Ajit C. Tamhane, D.D.D., *Statistics and Data Analysis: From Elementary to Intermediate*. 2000: Prentice Hall.
67. Castro, W.H., et al., *Noninvasive three-dimensional analysis of cervical spine motion in normal subjects in relation to age and sex. An experimental examination.* Spine, 2000. 25(4): p. 443-9.
68. Trott, P.H., et al., *Three-dimensional analysis of active cervical motion: the effect of age and gender.* Clin Biomech (Bristol, Avon), 1996. 11(4): p. 201-206.
69. Chen, J., et al., *Meta-analysis of normative cervical motion.* Spine, 1999. 24(15): p. 1571-8.
70. McClure, P., S. Siegler, and R. Nobilini, *Three-dimensional flexibility characteristics of the human cervical spine in vivo.* Spine, 1998. 23(2): p. 216-23.
71. Lind, B., et al., *Normal range of motion of the cervical spine.* Arch Phys Med Rehabil, 1989. 70(9): p. 692-5.
72. Nilsson, N., J. Hartvigsen, and H.W. Christensen, *Normal ranges of passive cervical motion for women and men 20-60 years old.* J Manipulative Physiol Ther, 1996. 19(5): p. 306-9.
73. Evens, R.G. and F.A. Mettler, *National CT use and radiation exposure: United States 1983.* AJR Am J Roentgenol, 1985. 144(5): p. 1077-81.
74. Linton, O.W. and F.A. Mettler, Jr., *National conference on dose reduction in CT, with an emphasis on pediatric patients.* AJR Am J Roentgenol, 2003. 181(2): p. 321-9.
75. C.D, B.D.J.a.E., *In response to Radiation risk of body CT: What to tell our patients and other questions.* . (R.E. Levatter) Radiology 2005. 234: p. 968-970.

76. Yoganandan, N., S. Kumaresan, and F.A. Pintar, *Geometric and mechanical properties of human cervical spine ligaments.* J Biomech Eng, 2000. 122(6): p. 623-9.
77. Thomas R.Kane, D.A.L., *Dynamics Online:Theory and Implementation with AUTOLEV™*, ed. 3. 2000: OnLine Dynamic,Inc.
78. Komistek, R.D., *Mathematical modeling of the human arm:an aid in the investigation of the role of muscle forces in the development of lateral epicondylitis*, in *Mechanical Department*. 1992, Memphis State University: Memphis.
79. Tekscan, C., *FlexiForce User Manual*
80. Dennis, D.A., et al., *In vivo anteroposterior femorotibial translation of total knee arthroplasty: a multicenter analysis.* Clin Orthop Relat Res, 1998. 356(356): p. 47-57.
81. Mahfouz, M.R., et al., *Effect of segmentation errors on 3D-to-2D registration of implant models in X-ray images.* J Biomech, 2005. 38(2): p. 229-39.
82. Hoff, W.A., et al., *Three-dimensional determination of femoral-tibial contact positions under in vivo conditions using fluoroscopy.* Clin Biomech (Bristol, Avon), 1998. 13(7): p. 455-472.
83. Sharma A, K.R., Outten JS,D'Lima DD, Colwell CWJr, *In-vivo quantification of the effect of slip and friction at the knee joint*, in *International Society for Technology in Arthroplasty*. 2006: New York,USA.
84. Mahfouz, M.R., et al., *A robust method for registration of three-dimensional knee implant models to two-dimensional fluoroscopy images.* IEEE Trans Med Imaging, 2003. 22(12): p. 1561-74.
85. Hino, H., et al., *Dynamic motion analysis of normal and unstable cervical spines using cineradiography. An in vivo study.* Spine, 1999. 24(2): p. 163-8.
86. Ordway, N.R., et al., *Cervical flexion, extension, protrusion, and retraction. A radiographic segmental analysis.* Spine, 1999. 24(3): p. 240-7.
87. Malmstrom, E.M., et al., *Primary and coupled cervical movements: the effect of age, gender, and body mass index. A 3-dimensional movement analysis of a population without symptoms of neck disorders.* Spine, 2006. 31(2): p. E44-50.
88. Lee, S.W., E.R. Draper, and S.P. Hughes, *Instantaneous center of rotation and instability of the cervical spine. A clinical study.* Spine, 1997. 22(6): p. 641-7; discussion 647-8.
89. Bhalla, S.K. and E.H. Simmons, *Normal ranges of intervertebral-joint motion of the cervical spine.* Can J Surg, 1969. 12(2): p. 181-7.
90. Holmes, A., et al., *The range and nature of flexion-extension motion in the cervical spine.* Spine, 1994. 19(22): p. 2505-10.
91. Dvorak, J., et al., *Clinical validation of functional flexion-extension radiographs of the cervical spine.* Spine, 1993. 18(1): p. 120-7.
92. Dvorak, J., J. Hayek, and R. Zehnder, *CT-functional diagnostics of the rotatory instability of the upper cervical spine. Part 2. An evaluation on healthy adults and patients with suspected instability.* Spine, 1987. 12(8): p. 726-31.
93. Mimura, M., et al., *Three-dimensional motion analysis of the cervical spine with special reference to the axial rotation.* Spine, 1989. 14(11): p. 1135-9.

94. Iai, H., et al., *Three-dimensional motion analysis of the upper cervical spine during axial rotation.* Spine, 1993. 18(16): p. 2388-92.
95. Lysell, E., *Motion in the cervical spine. An experimental study on autopsy specimens.* Acta Orthop Scand, 1969. 123: p. Suppl 123:1+.
96. Panjabi, M.M., et al., *Posture affects motion coupling patterns of the upper cervical spine.* J Orthop Res, 1993. 11(4): p. 525-36.
97. Schulte, K., C.R. Clark, and V.K. Goel, *Kinematics of the cervical spine following discectomy and stabilization.* Spine, 1989. 14(10): p. 1116-21.
98. Weinhoffer, S.L., et al., *Intradiscal pressure measurements above an instrumented fusion. A cadaveric study.* Spine, 1995. 20(5): p. 526-31.
99. Moroney, S.P., A.B. Schultz, and J.A. Miller, *Analysis and measurement of neck loads.* J Orthop Res, 1988. 6(5): p. 713-20.

Appendix

Matlab codes of 3D mathematical models

A Sample Function: Extension-Neutral Activity for the Normal Patient

```
function normen
% File  normen.m (normal muscular function for an extension-neutral activity in the normal patient)  created by Fei Liu on Thu Jul 26 2007
% %%%%%%%%%%%%%%% 3D MATHMATICAL MODEL OF NORMAL CERVICAL SPINE %%%%%%%%%%%%%%%%%
% %%%%%%%%%%%%%%% JULY 2007 FEI LIU %%%%%%%%%%%%%%%%%
% %%%%%%%%%%%%%%%%%
```

```
KC3                     =  0.001326145552560647;   % UNITS            Constant
KD1                     =  0;                      % UNITS            Constant
KD2                     =  0;                      % UNITS            Constant
KD3                     =  0.001286885245901639;   % UNITS            Constant
KE1                     =  0;                      % UNITS            Constant
KE2                     =  0;                      % UNITS            Constant
KE3                     =  0.001239669421487603;   % UNITS            Constant
KH1                     =  0;                      % UNITS            Constant
KH2                     =  0;                      % UNITS            Constant
KH3                     =  0.002165354330708661;   % UNITS            Constant
KK1                     =  0;                      % UNITS            Constant
KK2                     =  0;                      % UNITS            Constant
KK3                     =  0.001566831683168317;   % UNITS            Constant
KS1                     =  0;                      % UNITS            Constant
KS2                     =  0;                      % UNITS            Constant
KS3                     =  0.1;                    % UNITS            Constant
SW                      =  2.939085;               % UNITS            Constant
%-----------------------------+-------------------------+-------------------+-----------------

% Unit conversions
Pi       = 3.141592653589793;
DEGtoRAD = Pi/180.0;
RADtoDEG = 180.0/Pi;

% Reserve space and initialize matrices
z = zeros(5217,1);

% Evaluate constants
U1 = 0;
U2 = 0;
U3 = 0;
U4 = 0;
U5 = 0;
U6 = 0;
U7 = 0;
U8 = 0;
U9 = 0;
U10 = 0;
U11 = 0;
U12 = 0;
U13 = 0;
U14 = 0;
U15 = 0;
U16 = 0;
U17 = 0;
U18 = 0;
U19 = 0;
U20 = 0;
U21 = 0;
U22 = 0;
U23 = 0;
U24 = 0;
AON1 = -0.0026728 + 0.0033292*T + 4.760000000000001E-05*T^3 - 0.0007078*T^2;
AON2 = -0.05751899999999999 + 0.0002211*T^2 - 0.0007606000000000001*T - 1.86E-05*T^3;
AON3 = -0.86434;
AOAB1 = -0.01209 - 0.001911680911680912*T;
AOAB2 = 0.01044 - 0.0002108262108262106*T;
AOAB3 = 0.00105 - 2.991452991452993E-05*T;
BON1 = -0.012567 + 0.003425*T + 1.06E-05*T^3 - 0.0004329*T^2;
BON2 = -0.04323299999999999 + 0.001261*T + 2.0E-05*T^3 - 0.0002596*T^2;
BON3 = -0.86434;
BOBA1 = -0.009589686609686609;
BOBA2 = -0.005167094017094018;
BOBA3 = 0.0004870940170940171;
BOBC1 = -0.00888994301994302;
BOBC2 = 0.007934017094017094;
BOBC3 = 0.0003400854700854701;
CON1 = -0.016512 + 0.0005102999999999999*T + 0.0007865999999999999*T^2 - 0.0001154*T^3;
CON2 = -0.027922 + 0.0022666*T + 2.3E-05*T^3 - 0.000374*T^2;
CON3 = -0.86434;
COCB1 = -0.008285327635327636;
COCB2 = -0.006638461538461538;
COCB3 = -0.0005247008547008547;
COCD1 = -0.007903076923076924;
COCD2 = 0.006292991452991453;
COCD3 = -0.00087;
DON1 = -0.016591 + 0.0019449*T^2 - 0.0037191*T - 0.0002176*T^3;
DON2 = -0.015326 + 0.003221*T + 1.88E-05*T^3 - 0.0004602*T^2;
```

```
DON3 = -0.86434;
DODC1 = -0.007978183816766493;
DODC2 = -0.004916436727460349;
DODC3 = 7.940193373264239E-05;
DODE1 = -0.006568634048949009;
DODE2 = 0.008429335756107409;
DODE3 = 0.0006608116293155663;
EON1 = -0.009710900000000002 + 0.0017494*T^2 - 0.004681400000000001*T - 0.0002059*T^3;
EON2 = -0.0023381 + 0.0029449*T - 8.690000000000001E-05*T^2 - 2.03E-05*T^3;
EON3 = -0.86434;
EOED1 = -0.009795735468963816;
EOED2 = -0.004822139219777016;
EOED3 = -0.0004697265405926823;
EOEH1 = -0.007265103080063709;
EOEH2 = 0.008005395158938465;
EOEH3 = 0.0007962850797496465;
HON1 = 0.004818399999999999 + 0.001088*T^2 - 0.0063265*T - 0.0001424*T^3;
HON2 = 0.013992 + 0.0051273*T - 0.0001201*T^2 - 4.12E-05*T^3;
HON3 = -0.86434;
HOHE1 = -0.01109242434439285;
HOHE2 = -0.01440078964488414;
HOHE3 = -0.0009045023218251563;
HOHK1 = -0.009251317944231331;
HOHK2 = 0.006148708975480629;
HOHK3 = -0.0002898804316127151;
KON1 = 0.006843600000000001 + 0.0010594*T^2 - 0.0075755*T - 0.0001405*T^3;
KON2 = 0.023396 + 0.005086800000000001*T + 5.92E-05*T^2 - 6.759999999999999E-05*T^3;
KON3 = -0.86434;
z(562) = BOBC1 - BOBA1;
z(563) = BOBC2 - BOBA2;
z(564) = BOBC3 - BOBA3;
z(580) = -0.0217041 - BOBA1;
z(581) = 0.001172622 - BOBA2;
z(582) = 0.000203764 - BOBA3;
z(598) = -0.006617810000000001 - BOBA1;
z(599) = 0.002809858 - BOBA2;
z(600) = 0.000439106 - BOBA3;
z(616) = 0.01501072 - BOBA1;
z(617) = -0.008739810000000001 - BOBA2;
z(618) = 0.0009891419999999999 - BOBA3;
z(670) = 0.006931217 - BOBA1;
z(671) = -0.0040317 - BOBA2;
z(672) = -0.00687052 - BOBA3;
z(706) = 0.007283496 - BOBA1;
z(707) = -0.00421869 - BOBA2;
z(708) = 0.008922520000000002 - BOBA3;
z(724) = -21.06847638 - BOBA1;
z(725) = 7.784535433 - BOBA2;
z(786) = COCD1 - COCB1;
z(787) = COCD2 - COCB2;
z(788) = COCD3 - COCB3;
z(822) = -0.0204083 - COCB1;
z(823) = 4.9681E-05 - COCB2;
z(824) = -0.0012913 - COCB3;
z(858) = -0.005340810000000001 - COCB1;
z(859) = 0.00066872 - COCB2;
z(860) = -0.0009002400000000001 - COCB3;
z(894) = 0.01397208 - COCB1;
z(895) = -0.00841575 - COCB2;
z(896) = 5.641700000000001E-05 - COCB3;
z(1002) = 0.007522496 - COCB1;
z(1003) = -0.00404624 - COCB2;
z(1004) = -0.00816889 - COCB3;
z(1074) = 0.007558496 - COCB1;
z(1075) = -0.00353959 - COCB2;
z(1076) = 0.007263232000000001 - COCB3;
z(1110) = -17.86873228 - COCB1;
z(1111) = 8.772259843000001 - COCB2;
z(1226) = DODE1 - DODC1;
z(1227) = DODE2 - DODC2;
z(1228) = DODE3 - DODC3;
z(1280) = -0.0196764 - DODC1;
z(1281) = 0.0009485670000000001 - DODC2;
z(1282) = -0.00039917 - DODC3;
z(1334) = -0.00540409 - DODC1;
z(1335) = 0.001030031 - DODC2;
z(1336) = 0.000156039 - DODC3;
z(1388) = 0.01463621 - DODC1;
```

```
z(1389) = -0.00739011 - DODC2;
z(1390) = 0.001191484 - DODC3;
z(1550) = 0.008877487999999999 - DODC1;
z(1551) = -0.00297212 - DODC2;
z(1552) = -0.00680576 - DODC3;
z(1658) = 0.008493469 - DODC1;
z(1659) = -0.00334198 - DODC2;
z(1660) = 0.008434476 - DODC3;
z(1712) = -13.7861378 - DODC1;
z(1713) = 7.560169291 - DODC2;
z(1882) = EOEH1 - EOED1;
z(1883) = EOEH2 - EOED2;
z(1884) = EOEH3 - EOED3;
z(1954) = -0.0210342 - EOED1;
z(1955) = 0.001343886 - EOED2;
z(1956) = -2.0E-07 - EOED3;
z(2026) = -0.00699409 - EOED1;
z(2027) = 0.002091138 - EOED2;
z(2028) = 4.7299E-05 - EOED3;
z(2098) = 0.01525214 - EOED1;
z(2099) = -0.009746940000000001 - EOED2;
z(2100) = 0.000263142 - EOED3;
z(2170) = 0.007947417 - EOED1;
z(2171) = -0.00508017 - EOED2;
z(2172) = -0.00800262 - EOED3;
z(2242) = 0.008276457000000001 - EOED1;
z(2243) = -0.00641031 - EOED2;
z(2244) = 0.007242008 - EOED3;
z(2314) = -13.62237008 - EOED1;
z(2315) = 7.393145669 - EOED2;
z(2529) = HOHK1 - HOHE1;
z(2530) = HOHK2 - HOHE2;
z(2531) = HOHK3 - HOHE3;
z(2610) = KON1 - HON1;
z(2611) = KON2 - HON2;
z(2612) = KON3 - HON3;
z(5125) = BON1 - AON1;
z(5126) = BON2 - AON2;
z(5127) = BON3 - AON3;
z(5131) = CON1 - AON1;
z(5132) = CON2 - AON2;
z(5133) = CON3 - AON3;
z(5146) = DON1 - AON1;
z(5147) = DON2 - AON2;
z(5148) = DON3 - AON3;
z(5161) = EON1 - AON1;
z(5162) = EON2 - AON2;
z(5163) = EON3 - AON3;
z(5176) = HON1 - AON1;
z(5177) = HON2 - AON2;
z(5178) = HON3 - AON3;
z(5191) = KON1 - AON1;
z(5192) = KON2 - AON2;
z(5193) = KON3 - AON3;


%==========================================================================
function OpenOutputFilesAndWriteHeadings
FileIdentifier = fopen('normen.1', 'wt');   if( FileIdentifier == -1 ) error('Error: unable to open file normen.1'); end
fprintf( 1,          '%%      T            FABALL          FBCALL          FCDALL          FDEALL\n' );
fprintf( 1,          '%%    (UNITS)        (UNITS)         (UNITS)         (UNITS)         (UNITS)\n\n' );
fprintf(FileIdentifier, '%% FILE: normen.1\n%%\n' );
fprintf(FileIdentifier, '%%      T            FABALL          FBCALL          FCDALL          FDEALL\n' );
fprintf(FileIdentifier, '%%    (UNITS)        (UNITS)         (UNITS)         (UNITS)         (UNITS)\n\n' );
FileIdentifier = fopen('normen.2', 'wt');   if( FileIdentifier == -1 ) error('Error: unable to open file normen.2'); end
fprintf(FileIdentifier, '%% FILE: normen.2\n%%\n' );
fprintf(FileIdentifier, '%%      T            FABPLL          FBCPLL          FCDPLL          FDEPLL\n' );
fprintf(FileIdentifier, '%%    (UNITS)        (UNITS)         (UNITS)         (UNITS)         (UNITS)\n\n' );
FileIdentifier = fopen('normen.3', 'wt');   if( FileIdentifier == -1 ) error('Error: unable to open file normen.3'); end
fprintf(FileIdentifier, '%% FILE: normen.3\n%%\n' );
fprintf(FileIdentifier, '%%      T            FABIL           FBCIL           FCDIL           FDEIL\n' );
fprintf(FileIdentifier, '%%    (UNITS)        (UNITS)         (UNITS)         (UNITS)         (UNITS)\n\n' );
FileIdentifier = fopen('normen.4', 'wt');   if( FileIdentifier == -1 ) error('Error: unable to open file normen.4'); end
fprintf(FileIdentifier, '%% FILE: normen.4\n%%\n' );
fprintf(FileIdentifier, '%%      T            FABLLF          FBCLLF          FCDLLF          FDELLF\n' );
fprintf(FileIdentifier, '%%    (UNITS)        (UNITS)         (UNITS)         (UNITS)         (UNITS)\n\n' );
```

```
FileIdentifier = fopen('normen.5', 'wt');   if( FileIdentifier == -1 ) error('Error: unable to open file normen.5'); end
fprintf(FileIdentifier, '%% FILE: normen.5\n%%\n' );
fprintf(FileIdentifier, '%%       T            FABRLF          FBCRLF          FCDRLF          FDERLF\n' );
fprintf(FileIdentifier, '%%    (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)\n\n' );
FileIdentifier = fopen('normen.6', 'wt');   if( FileIdentifier == -1 ) error('Error: unable to open file normen.6'); end
fprintf(FileIdentifier, '%% FILE: normen.6\n%%\n' );
fprintf(FileIdentifier, '%%       T            FAB12           FBC12           FCD12           FDE12\n' );
fprintf(FileIdentifier, '%%    (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)\n\n' );
FileIdentifier = fopen('normen.7', 'wt');   if( FileIdentifier == -1 ) error('Error: unable to open file normen.7'); end
fprintf(FileIdentifier, '%% FILE: normen.7\n%%\n' );
fprintf(FileIdentifier, '%%       T            FAB22           FBC22           FCD22           FDE22\n' );
fprintf(FileIdentifier, '%%    (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)\n\n' );
FileIdentifier = fopen('normen.8', 'wt');   if( FileIdentifier == -1 ) error('Error: unable to open file normen.8'); end
fprintf(FileIdentifier, '%% FILE: normen.8\n%%\n' );
fprintf(FileIdentifier, '%%       T            FAB32           FBC32           FCD32           FDE32\n' );
fprintf(FileIdentifier, '%%    (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)\n\n' );
FileIdentifier = fopen('normen.9', 'wt');   if( FileIdentifier == -1 ) error('Error: unable to open file normen.9'); end
fprintf(FileIdentifier, '%% FILE: normen.9\n%%\n' );
fprintf(FileIdentifier, '%%       T            FESEMCER        FDSEMCER        FCSEMCER        FBSEMCER\n' );
fprintf(FileIdentifier, '%%    (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)\n\n' );
FileIdentifier = fopen('normen.10', 'wt');   if( FileIdentifier == -1 ) error('Error: unable to open file normen.10'); end
fprintf(FileIdentifier, '%% FILE: normen.10\n%%\n' );
fprintf(FileIdentifier, '%%       T            FESEMCER1       FDSEMCER1       FCSEMCER1       FBSEMCER1\n' );
fprintf(FileIdentifier, '%%    (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)\n\n' );
FileIdentifier = fopen('normen.11', 'wt');   if( FileIdentifier == -1 ) error('Error: unable to open file normen.11'); end
fprintf(FileIdentifier, '%% FILE: normen.11\n%%\n' );
fprintf(FileIdentifier, '%%       T          P_AO_BO[1]      P_AO_BO[2]      P_AO_BO[3]      A_B[1,1]        A_B[1,2]        A_B[1,3]
A_B[2,1]        A_B[2,2]        A_B[2,3]        A_B[3,1]        A_B[3,2]        A_B[3,3]\n' );
fprintf(FileIdentifier, '%%    (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)
(UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)\n\n' );
FileIdentifier = fopen('normen.12', 'wt');   if( FileIdentifier == -1 ) error('Error: unable to open file normen.12'); end
fprintf(FileIdentifier, '%% FILE: normen.12\n%%\n' );
fprintf(FileIdentifier, '%%       T          P_AO_CO[1]      P_AO_CO[2]      P_AO_CO[3]      A_C[1,1]        A_C[1,2]        A_C[1,3]
A_C[2,1]        A_C[2,2]        A_C[2,3]        A_C[3,1]        A_C[3,2]        A_C[3,3]\n' );
fprintf(FileIdentifier, '%%    (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)
(UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)\n\n' );
FileIdentifier = fopen('normen.13', 'wt');   if( FileIdentifier == -1 ) error('Error: unable to open file normen.13'); end
fprintf(FileIdentifier, '%% FILE: normen.13\n%%\n' );
fprintf(FileIdentifier, '%%       T          P_AO_DO[1]      P_AO_DO[2]      P_AO_DO[3]      A_D[1,1]        A_D[1,2]        A_D[1,3]
A_D[2,1]        A_D[2,2]        A_D[2,3]        A_D[3,1]        A_D[3,2]        A_D[3,3]\n' );
fprintf(FileIdentifier, '%%    (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)
(UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)\n\n' );
FileIdentifier = fopen('normen.14', 'wt');   if( FileIdentifier == -1 ) error('Error: unable to open file normen.14'); end
fprintf(FileIdentifier, '%% FILE: normen.14\n%%\n' );
fprintf(FileIdentifier, '%%       T          P_AO_EO[1]      P_AO_EO[2]      P_AO_EO[3]      A_E[1,1]        A_E[1,2]        A_E[1,3]        A_E[2,1]
A_E[2,2]        A_E[2,3]        A_E[3,1]        A_E[3,2]        A_E[3,3]\n' );
fprintf(FileIdentifier, '%%    (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)
(UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)\n\n' );
FileIdentifier = fopen('normen.15', 'wt');   if( FileIdentifier == -1 ) error('Error: unable to open file normen.15'); end
fprintf(FileIdentifier, '%% FILE: normen.15\n%%\n' );
fprintf(FileIdentifier, '%%       T          P_AO_HO[1]      P_AO_HO[2]      P_AO_HO[3]      A_H[1,1]        A_H[1,2]        A_H[1,3]
A_H[2,1]        A_H[2,2]        A_H[2,3]        A_H[3,1]        A_H[3,2]        A_H[3,3]\n' );
fprintf(FileIdentifier, '%%    (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)
(UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)\n\n' );
FileIdentifier = fopen('normen.16', 'wt');   if( FileIdentifier == -1 ) error('Error: unable to open file normen.16'); end
fprintf(FileIdentifier, '%% FILE: normen.16\n%%\n' );
fprintf(FileIdentifier, '%%       T          P_AO_KO[1]      P_AO_KO[2]      P_AO_KO[3]      A_K[1,1]        A_K[1,2]        A_K[1,3]
A_K[2,1]        A_K[2,2]        A_K[2,3]        A_K[3,1]        A_K[3,2]        A_K[3,3]\n' );
fprintf(FileIdentifier, '%%    (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)
(UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)\n\n' );
FileIdentifier = fopen('normen.17', 'wt');   if( FileIdentifier == -1 ) error('Error: unable to open file normen.17'); end
fprintf(FileIdentifier, '%% FILE: normen.17\n%%\n' );
fprintf(FileIdentifier, '%%       T          P_AO_SO[1]      P_AO_SO[2]      P_AO_SO[3]      A_S[1,1]        A_S[1,2]        A_S[1,3]        A_S[2,1]
A_S[2,2]        A_S[2,3]        A_S[3,1]        A_S[3,2]        A_S[3,3]\n' );
fprintf(FileIdentifier, '%%    (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)
(UNITS)         (UNITS)         (UNITS)         (UNITS)         (UNITS)\n\n' );


%==========================================================================
function DoCalculations
global   T;
global   KB1 KB2 KB3 KC1 KC2 KC3 KD1 KD2 KD3 KE1 KE2 KE3 KH1 KH2 KH3 KK1 KK2 KK3 KS1 KS2 KS3 SW;
global   AOAB1 AOAB2 AOAB3 AON1 AON2 AON3 BOBA1 BOBA2 BOBA3 BOBC1 BOBC2 BOBC3 BON1 BON2 BON3 COCB1 COCB2
COCB3 COCD1 COCD2 COCD3 CON1 CON2 CON3 DABALL DABIL DABLLF DABPLL DABRLF DBCALL DBCIL DBCLLF DBCPLL
DBCRLF DCDALL DCDIL DCDLLF DCDPLL DCDRLF DDEALL DDEIL DDELLF DDEPLL DDERLF DODC1 DODC2 DODC3 DODE1 DODE2
DODE3 DON1 DON2 DON3 EOED1 EOED2 EOED3 EOEH1 EOEH2 EOEH3 EON1 EON2 EON3 FAB1 FAB2 FAB3 FBC1 FBC2 FBC3
FBSEMCER FBSEMCER1 FBSEMCER_1 FBSEMCER_2 FBSEMCER_3 FCD1 FCD2 FCD3 FCSEMCER FCSEMCER1 FCSEMCER_1
```

```
FCSEMCER_2 FCSEMCER_3 FDE1 FDE2 FDE3 FDSEMCER FDSEMCER1 FDSEMCER_1 FDSEMCER_2 FDSEMCER_3 FESEMCER
FESEMCER1 FESEMCER_1 FESEMCER_2 FESEMCER_3 HOHE1 HOHE2 HOHE3 HOHK1 HOHK2 HOHK3 HON1 HON2 HON3 KON1 KON2
KON3 U1 U10 U11 U12 U13 U14 U15 U16 U17 U18 U19 U2 U20 U21 U22 U23 U24 U3 U4 U5 U6 U7 U8 U9 QNA_1 QNA_2 QNA_3 QNB_1
QNB_2 QNB_3 QNC_1 QNC_2 QNC_3 QND_1 QND_3 QNE_1 QNE_2 QNE_3 QNH_1 QNH_2 QNH_3 QNK_1 QNK_2 QNK_3 QNS_1 QNS_2
QNS_3 QNA_1p QNA_2p QNA_3p QNB_1p QNB_2p QNB_3p QNC_1p QNC_2p QNC_3p QND_1p QND_3p QNE_1p QNE_2p QNE_3p QNH_1p
QNH_2p QNH_3p QNK_1p QNK_2p QNK_3p QNS_1p QNS_2p QNS_3p QNA_1pp QNA_2pp QNA_3pp QNB_1pp QNB_2pp QNB_3pp QNC_1pp
QNC_2pp QNC_3pp QND_1pp QND_3pp QNE_1pp QNE_2pp QNE_3pp QNH_1pp QNH_2pp QNH_3pp QNK_1pp QNK_2pp QNK_3pp QNS_1pp
QNS_2pp QNS_3pp FAB12 FAB22 FAB32 FABALL FABIL FABLLF FABPLL FABRLF FBC12 FBC22 FBC32 FBCALL FBCIL FBCLLF FBCPLL
FBCRLF FCD12 FCD22 FCD32 FCDALL FCDIL FCDLLF FCDPLL FCDRLF FDE12 FDE22 FDE32 FDEALL FDEIL FDELLF FDEPLL FDERLF;
global   DEGtoRAD RADtoDEG z;

% Quantities which were specified
QNA_1 = 0.006455972903127025 + 0.001720894642466409*T + 2.443460952792061E-05*T^3 - 0.0005253441048502931*T^2;
QNA_1p = 0.001720894642466409 + 7.330382858376184E-05*T^2 - 0.001050688209700586*T;
QNA_1pp = -0.001050688209700586 + 0.0001466076571675237*T;
QNA_2 = 0.002192133540504878 + 0.006099925735720181*T + 0.000153588974175501*T^3 - 0.001877974275145899*T^2;
QNA_2p = 0.006099925735720181 + 0.000460766922526503*T^2 - 0.003755948550291797*T;
QNA_2pp = -0.003755948550291797 + 0.0009215338450530061*T;
QNA_3 = -0.4369431782367804 + 0.1648131865950765*T + 0.002274164015348611*T^3 - 0.03375815839207432*T^2;
QNA_3p = 0.1648131865950765 + 0.006822492046045833*T^2 - 0.06751631678414864*T;
QNA_3pp = -0.06751631678414864 + 0.01364498409209167*T;
QNB_1 = 0.006628760499074464 + 0.002415535684760152*T^2 - 0.007351326809400116*T - 0.0001937315469713706*T^3;
QNB_1p = -0.007351326809400116 + 0.004831071369520304*T - 0.000581194640914111 8*T^2;
QNB_1pp = 0.004831071369520304 - 0.001162389281828224*T;
QNB_2 = 0.0009354964790689607 + 0.008820894039579341*T + 0.000268780704807 1268*T^3 - 0.00307177948351002*T^2;
QNB_2p = 0.008820894039579341 + 0.0008063421144213803*T^2 - 0.006143558967020039*T;
QNB_2pp = -0.006143558967020039 + 0.001612684228842761*T;
QNB_3 = -0.3054500723915276 + 0.03751236161311412*T + 0.000298451302091 0303*T^3 - 0.006747442888210078*T^2;
QNB_3p = 0.03751236161311412 + 0.0008953539062730911*T^2 - 0.01349488577642016*T;
QNB_3pp = -0.01349488577642016 + 0.001790707812546182*T;
QNC_1 = 0.0008848819307611252 + 0.007209955139988576*T + 0.000137881010907 5521*T^3 - 0.001998401993533508*T^2;
QNC_1p = 0.007209955139988576 + 0.000413643032722 6562*T^2 - 0.003996803987067015*T;
QNC_1pp = -0.003996803987067015 + 0.0008272860654453124*T;
QNC_2 = 0.005492551156026155 + 3.839724354387525E-05*T^3 - 0.0009651670763528643*T - 0.0002146754979953026*T^2;
QNC_2p = -0.0009651670763528643 + 0.000115191730631 6258*T^2 - 0.0004293509959906051*T;
QNC_2pp = -0.0004293509959906051 + 0.0002303834612632515*T;
QNC_3 = -0.1622981671429527 + 0.02145533249476629*T^2 - 0.05694485750481899*T - 0.002539454061651749*T^3;
QNC_3p = -0.05694485750481899 + 0.04291066498953258*T - 0.007618362184955247*T^2;
QNC_3pp = 0.04291066498953258 - 0.0152367243699105*T;
QND_1 = 0.003822271061867582 + 0.003689626038716013*T + 6.45771823237902E-05*T^3 - 0.0009302604913129776*T^2;
QND_1p = 0.003689626038716013 + 0.0001937315469713706*T^2 - 0.001860520982625955*T;
QND_1pp = -0.001860520982625955 + 0.0003874630939427412*T;
QND_3 = 0.04634896361596141 + 0.05437747817513533*T^2 - 0.227154602147062*T - 0.004646066468808905*T^3;
QND_3p = -0.227154602147062 + 0.1087549563502707*T - 0.01393819940642671*T^2;
QND_3pp = 0.1087549563502707 - 0.02787639881285343*T;
QNE_1 = 0.006412339671827167 + 0.004729842272904634*T + 0.000198967534727353 6*T^3 - 0.00208392312688123*T^2;
QNE_1p = 0.004729842272904634 + 0.0005969026041820608*T^2 - 0.004167846253762459*T;
QNE_1pp = -0.004167846253762459 + 0.001193805208364122*T;
QNE_2 = 0.005286602304290824 + 0.001773254520026239*T^2 - 0.004530874738177 28*T - 0.000158824961931484*T^3;
QNE_2p = -0.004530874738177728 + 0.003546509040052478*T - 0.000476474885794452*T^2;
QNE_2pp = 0.003546509040052478 - 0.0009529497715889039*T;
QNE_3 = 0.2584309023428004 + 0.01283166066066231*T^2 - 0.1607413334501738*T - 0.0007853981633974482*T^3;
QNE_3p = -0.1607413334501738 + 0.02566332132132462*T - 0.002356194490192345*T^2;
QNE_3pp = 0.02566332132132462 - 0.00471238898038469*T;
QNH_1 = 0.006511823439190843 + 0.005206317158699085*T + 0.000244346095279 2061*T^3 - 0.002509783464367846*T^2;
QNH_1p = 0.005206317158699085 + 0.0007330382858376184*T^2 - 0.005019566928735692*T;
QNH_1pp = -0.005019566928735692 + 0.001466076571675237*T;
QNH_2 = 0.004710643651132695 + 0.005705481324769463*T + 0.000207694180987325 2*T^3 - 0.002303834612632515*T^2;
QNH_2p = 0.005705481324769463 + 0.0006230825429619757*T^2 - 0.00460766922526503*T;
QNH_2pp = -0.00460766922526503 + 0.001246165085923951*T;
QNH_3 = 0.5682966577418737 + 0.001855284994869972*T^3 - 0.1047197551196598*T - 0.01840100630377622*T^2;
QNH_3p = -0.1047197551196598 + 0.005565854984609917*T^2 - 0.03680201260755243*T;
QNH_3pp = -0.03680201260755243 + 0.01113170996921984*T;
QNK_1 = 0.006970845032465351 + 0.001759291886010284*T + 6.283185307179586E-05*T^3 - 0.0006928957130417488*T^2;
QNK_1p = 0.001759291886010284 + 0.0001884955592153876*T^2 - 0.001385791426083498*T;
QNK_1pp = -0.001385791426083498 + 0.0003769911184307751*T;
QNK_2 = 0.0003700098014227979 + 0.006063273821428301*T + 6.45771823237902E-05*T^3 - 0.001296779634231787*T^2;
QNK_2p = 0.006063273821428301 + 0.0001937315469713706*T^2 - 0.002593559268463574*T;
QNK_2pp = -0.002593559268463574 + 0.0003874630939427412*T;
QNK_3 = 0.9802641743826152 + 0.004188790204786391*T^3 - 0.119713878723543*T - 0.04000992777271801*T^2;
QNK_3p = -0.119713878723543 + 0.01256637061435917*T^2 - 0.08001985554543602*T;
QNK_3pp = -0.08001985554543602 + 0.02513274122871834*T;
QNS_1 = 0;
QNS_1p = 0;
QNS_1pp = 0;
QNS_2 = 0;
QNS_2p = 0;
QNS_2pp = 0;
```

```
QNS_3 = 0.5719443958785418 + 0.03882833986911785*T^2 - 0.2631781979082249*T - 0.004804891430740389*T^3;
QNS_3p = -0.2631781979082249 + 0.0776566797382357*T - 0.0144146742922211 7*T^2;
QNS_3pp = 0.0776566797382357 - 0.02882934858444233*T;

z(1) = cos(QNA_3);
z(2) = sin(QNA_3);
z(3) = cos(QNA_2);
z(4) = sin(QNA_2);
z(5) = cos(QNA_1);
z(6) = sin(QNA_1);
z(7) = cos(QNB_3);
z(8) = sin(QNB_3);
z(9) = cos(QNB_2);
z(10) = sin(QNB_2);
z(11) = cos(QNB_1);
z(12) = sin(QNB_1);
z(13) = cos(QNC_3);
z(14) = sin(QNC_3);
z(15) = cos(QNC_2);
z(16) = sin(QNC_2);
z(17) = cos(QNC_1);
z(18) = sin(QNC_1);
z(19) = cos(QND_3);
z(20) = sin(QND_3);
z(23) = cos(QND_1);
z(24) = sin(QND_1);
z(25) = cos(QNE_3);
z(26) = sin(QNE_3);
z(27) = cos(QNE_2);
z(28) = sin(QNE_2);
z(29) = cos(QNE_1);
z(30) = sin(QNE_1);
z(31) = cos(QNH_3);
z(32) = sin(QNH_3);
z(33) = cos(QNH_2);
z(34) = sin(QNH_2);
z(35) = cos(QNH_1);
z(36) = sin(QNH_1);
z(37) = cos(QNK_3);
z(38) = sin(QNK_3);
z(39) = cos(QNK_2);
z(40) = sin(QNK_2);
z(41) = cos(QNK_1);
z(42) = sin(QNK_1);
z(43) = cos(QNS_3);
z(44) = sin(QNS_3);
z(45) = cos(QNS_2);
z(46) = sin(QNS_2);
z(47) = cos(QNS_1);
z(48) = sin(QNS_1);
z(49) = z(1)*z(3);
z(50) = z(1)*z(4);
z(51) = z(2)*z(3);
z(52) = z(2)*z(4);
z(53) = z(6)*z(50) - z(2)*z(5);
z(54) = z(2)*z(6) + z(5)*z(50);
z(55) = z(1)*z(5) + z(6)*z(52);
z(56) = z(5)*z(52) - z(1)*z(6);
z(57) = z(3)*z(6);
z(58) = z(3)*z(5);
z(59) = z(1)*z(3)*QNA_2p - z(2)*z(4)*QNA_3p;
z(60) = z(2)*z(6)*QNA_1p + z(5)*z(50)*QNA_1p + z(6)*z(59) - z(1)*z(5)*QNA_3p;
z(61) = z(1)*z(4)*QNA_3p + z(2)*z(3)*QNA_2p;
z(62) = z(5)*z(52)*QNA_1p + z(6)*z(61) - z(1)*z(6)*QNA_1p - z(2)*z(5)*QNA_3p;
z(63) = z(3)*z(5)*QNA_1p - z(4)*z(6)*QNA_2p;
z(64) = z(2)*z(6)*QNA_3p + z(5)*z(61) - z(1)*z(5)*QNA_1p - z(6)*z(52)*QNA_1p;
z(65) = -z(3)*z(6)*QNA_1p - z(4)*z(5)*QNA_2p;
z(66) = z(1)*z(6)*QNA_3p + z(2)*z(5)*QNA_1p + z(5)*z(59) - z(6)*z(50)*QNA_1p;
z(67) = -z(1)*z(4)*QNA_2p - z(2)*z(3)*QNA_3p;
z(68) = z(1)*z(3)*QNA_3p - z(2)*z(4)*QNA_2p;
z(69) = z(54)*z(60) + z(56)*z(62) + z(58)*z(63);
z(70) = z(49)*z(66) + z(51)*z(64) - z(4)*z(65);
z(71) = z(53)*z(67) + z(55)*z(68) - z(3)*z(57)*QNA_2p;
z(72) = z(7)*z(49) + z(8)*z(51);
z(73) = z(7)*z(51) - z(8)*z(49);
z(74) = z(7)*z(53) + z(8)*z(55);
z(75) = z(7)*z(55) - z(8)*z(53);
z(76) = z(7)*z(54) + z(8)*z(56);
```

```
z(77) = z(7)*z(56) - z(8)*z(54);
z(78) = z(10)*z(12);
z(79) = z(10)*z(11);
z(80) = z(9)*z(12);
z(81) = z(9)*z(11);
z(82) = z(4)*z(10) + z(9)*z(72);
z(83) = z(11)*z(73) + z(72)*z(78) - z(4)*z(80);
z(84) = z(72)*z(79) - z(4)*z(81) - z(12)*z(73);
z(85) = z(9)*z(74) - z(10)*z(57);
z(86) = z(11)*z(75) + z(57)*z(80) + z(74)*z(78);
z(87) = z(57)*z(81) + z(74)*z(79) - z(12)*z(75);
z(88) = z(9)*z(76) - z(10)*z(58);
z(89) = z(11)*z(77) + z(58)*z(80) + z(76)*z(78);
z(90) = z(58)*z(81) + z(76)*z(79) - z(12)*z(77);
z(91) = z(7)*z(68) - z(7)*z(49)*QNB_3p - z(8)*z(51)*QNB_3p - z(8)*z(67);
z(92) = z(7)*z(51)*QNB_3p + z(7)*z(67) + z(8)*z(68) - z(8)*z(49)*QNB_3p;
z(93) = z(9)*z(12)*QNB_2p + z(10)*z(11)*QNB_1p;
z(94) = z(9)*z(11)*QNB_1p - z(10)*z(12)*QNB_2p;
z(95) = z(11)*z(91) + z(72)*z(93) + z(78)*z(92) - z(3)*z(80)*QNA_2p - z(12)*z(73)*QNB_1p - z(4)*z(94);
z(96) = z(7)*z(62) - z(7)*z(53)*QNB_3p - z(8)*z(55)*QNB_3p - z(8)*z(60);
z(97) = z(7)*z(55)*QNB_3p + z(7)*z(60) + z(8)*z(62) - z(8)*z(53)*QNB_3p;
z(98) = z(11)*z(96) + z(57)*z(94) + z(74)*z(93) + z(78)*z(97) + z(80)*z(63) - z(12)*z(75)*QNB_1p;
z(99) = z(7)*z(64) - z(7)*z(54)*QNB_3p - z(8)*z(56)*QNB_3p - z(8)*z(66);
z(100) = z(7)*z(56)*QNB_3p + z(7)*z(66) + z(8)*z(64) - z(8)*z(54)*QNB_3p;
z(101) = z(11)*z(99) + z(58)*z(94) + z(76)*z(93) + z(78)*z(100) + z(80)*z(65) - z(12)*z(77)*QNB_1p;
z(102) = -z(9)*z(12)*QNB_1p - z(10)*z(11)*QNB_2p;
z(103) = z(9)*z(11)*QNB_2p - z(10)*z(12)*QNB_1p;
z(104) = z(57)*z(102) + z(74)*z(103) + z(79)*z(97) + z(81)*z(63) - z(11)*z(75)*QNB_1p - z(12)*z(96);
z(105) = z(58)*z(102) + z(76)*z(103) + z(79)*z(100) + z(81)*z(65) - z(11)*z(77)*QNB_1p - z(12)*z(99);
z(106) = z(72)*z(103) + z(79)*z(92) - z(3)*z(81)*QNA_2p - z(11)*z(73)*QNB_1p - z(4)*z(102) - z(12)*z(91);
z(107) = z(9)*z(100) - z(9)*z(58)*QNB_2p - z(10)*z(76)*QNB_2p - z(10)*z(65);
z(108) = z(3)*z(10)*QNA_2p + z(4)*z(9)*QNB_2p + z(9)*z(92) - z(10)*z(72)*QNB_2p;
z(109) = z(9)*z(97) - z(9)*z(57)*QNB_2p - z(10)*z(74)*QNB_2p - z(10)*z(63);
z(110) = z(84)*z(95) + z(87)*z(98) + z(90)*z(101);
z(111) = z(82)*z(106) + z(85)*z(104) + z(88)*z(105);
z(112) = z(83)*z(108) + z(86)*z(109) + z(89)*z(107);
z(113) = z(7)*z(9);
z(114) = z(8)*z(9);
z(115) = z(7)*z(78) - z(8)*z(11);
z(116) = z(7)*z(11) + z(8)*z(78);
z(117) = z(7)*z(79) + z(8)*z(12);
z(118) = z(8)*z(79) - z(7)*z(12);
z(119) = z(13)*z(15);
z(120) = z(13)*z(16);
z(121) = z(14)*z(15);
z(122) = z(14)*z(16);
z(123) = z(18)*z(120) - z(14)*z(17);
z(124) = z(14)*z(18) + z(17)*z(120);
z(125) = z(13)*z(17) + z(18)*z(122);
z(126) = z(17)*z(122) - z(13)*z(18);
z(127) = z(15)*z(18);
z(128) = z(15)*z(17);
z(129) = z(10)*z(16) + z(113)*z(119) + z(114)*z(121);
z(130) = z(113)*z(123) + z(114)*z(125) - z(10)*z(127);
z(131) = z(113)*z(124) + z(114)*z(126) - z(10)*z(128);
z(132) = z(115)*z(119) + z(116)*z(121) - z(16)*z(80);
z(133) = z(80)*z(127) + z(115)*z(123) + z(116)*z(125);
z(134) = z(80)*z(128) + z(115)*z(124) + z(116)*z(126);
z(135) = z(117)*z(119) + z(118)*z(121) - z(16)*z(81);
z(136) = z(81)*z(127) + z(117)*z(123) + z(118)*z(125);
z(137) = z(81)*z(128) + z(117)*z(124) + z(118)*z(126);
z(138) = -z(7)*z(10)*QNB_2p - z(8)*z(9)*QNB_3p;
z(139) = z(13)*z(15)*QNC_2p - z(14)*z(16)*QNC_3p;
z(140) = z(14)*z(18)*QNC_1p + z(17)*z(120)*QNC_1p + z(18)*z(139) - z(13)*z(17)*QNC_3p;
z(141) = z(7)*z(9)*QNB_3p - z(8)*z(10)*QNB_2p;
z(142) = z(13)*z(16)*QNC_3p + z(14)*z(15)*QNC_2p;
z(143) = z(17)*z(122)*QNC_1p + z(18)*z(142) - z(13)*z(18)*QNC_1p - z(14)*z(17)*QNC_3p;
z(144) = z(15)*z(17)*QNC_1p - z(16)*z(18)*QNC_2p;
z(145) = z(113)*z(140) + z(114)*z(143) + z(123)*z(138) + z(125)*z(141) - z(9)*z(127)*QNB_2p - z(10)*z(144);
z(146) = z(8)*z(12)*QNB_1p + z(7)*z(93) - z(7)*z(11)*QNB_3p - z(8)*z(78)*QNB_3p;
z(147) = z(7)*z(78)*QNB_3p + z(8)*z(93) - z(7)*z(12)*QNB_1p - z(8)*z(11)*QNB_3p;
z(148) = z(80)*z(144) + z(115)*z(140) + z(116)*z(143) + z(123)*z(146) + z(125)*z(147) + z(127)*z(94);
z(149) = z(7)*z(12)*QNB_3p + z(8)*z(11)*QNB_1p + z(7)*z(103) - z(8)*z(79)*QNB_3p;
z(150) = z(7)*z(79)*QNB_3p + z(8)*z(12)*QNB_3p + z(8)*z(103) - z(7)*z(11)*QNB_1p;
z(151) = z(81)*z(144) + z(117)*z(140) + z(118)*z(143) + z(123)*z(149) + z(125)*z(150) + z(127)*z(102);
z(152) = -z(15)*z(18)*QNC_1p - z(16)*z(17)*QNC_2p;
z(153) = z(13)*z(18)*QNC_3p + z(14)*z(17)*QNC_1p + z(17)*z(139) - z(18)*z(120)*QNC_1p;
z(154) = z(14)*z(18)*QNC_3p + z(17)*z(142) - z(13)*z(17)*QNC_1p - z(18)*z(122)*QNC_1p;
```

```
z(155) = z(80)*z(152) + z(115)*z(153) + z(116)*z(154) + z(124)*z(146) + z(126)*z(147) + z(128)*z(94);
z(156) = z(81)*z(152) + z(117)*z(153) + z(118)*z(154) + z(124)*z(149) + z(126)*z(150) + z(128)*z(102);
z(157) = z(113)*z(153) + z(114)*z(154) + z(124)*z(138) + z(126)*z(141) - z(9)*z(128)*QNB_2p - z(10)*z(152);
z(158) = -z(13)*z(16)*QNC_2p - z(14)*z(15)*QNC_3p;
z(159) = z(13)*z(15)*QNC_3p - z(14)*z(16)*QNC_2p;
z(160) = z(117)*z(158) + z(118)*z(159) + z(119)*z(149) + z(121)*z(150) - z(15)*z(81)*QNC_2p - z(16)*z(102);
z(161) = z(9)*z(16)*QNB_2p + z(10)*z(15)*QNC_2p + z(113)*z(158) + z(114)*z(159) + z(119)*z(138) + z(121)*z(141);
z(162) = z(115)*z(158) + z(116)*z(159) + z(119)*z(146) + z(121)*z(147) - z(15)*z(80)*QNC_2p - z(16)*z(94);
z(163) = z(131)*z(145) + z(134)*z(148) + z(137)*z(151);
z(164) = z(129)*z(157) + z(132)*z(155) + z(135)*z(156);
z(165) = z(130)*z(161) + z(133)*z(162) + z(136)*z(160);
z(166) = z(19)*z(119) + z(20)*z(121);
z(167) = z(19)*z(121) - z(20)*z(119);
z(168) = z(19)*z(123) + z(20)*z(125);
z(169) = z(19)*z(125) - z(20)*z(123);
z(170) = z(19)*z(124) + z(20)*z(126);
z(171) = z(19)*z(126) - z(20)*z(124);
z(172) = z(23)*z(167) - z(16)*z(24);
z(173) = -z(16)*z(23) - z(24)*z(167);
z(174) = z(23)*z(169) + z(24)*z(127);
z(175) = z(23)*z(127) - z(24)*z(169);
z(176) = z(23)*z(171) + z(24)*z(128);
z(177) = z(23)*z(128) - z(24)*z(171);
z(178) = z(19)*z(159) - z(19)*z(119)*QND_3p - z(20)*z(121)*QND_3p - z(20)*z(158);
z(179) = z(23)*z(178) - z(15)*z(24)*QNC_2p - z(16)*z(23)*QND_1p - z(24)*z(167)*QND_1p;
z(180) = z(19)*z(143) - z(19)*z(123)*QND_3p - z(20)*z(125)*QND_3p - z(20)*z(140);
z(181) = z(23)*z(127)*QND_1p + z(23)*z(180) + z(24)*z(144) - z(24)*z(169)*QND_1p;
z(182) = z(19)*z(154) - z(19)*z(124)*QND_3p - z(20)*z(126)*QND_3p - z(20)*z(153);
z(183) = z(23)*z(128)*QND_1p + z(23)*z(182) + z(24)*z(152) - z(24)*z(171)*QND_1p;
z(184) = z(23)*z(144) - z(23)*z(169)*QND_1p - z(24)*z(127)*QND_1p - z(24)*z(180);
z(185) = z(23)*z(152) - z(23)*z(171)*QND_1p - z(24)*z(128)*QND_1p - z(24)*z(182);
z(186) = z(16)*z(24)*QND_1p - z(15)*z(23)*QNC_2p - z(23)*z(167)*QND_1p - z(24)*z(178);
z(187) = z(19)*z(126)*QND_3p + z(19)*z(153) + z(20)*z(154) - z(20)*z(124)*QND_3p;
z(188) = z(19)*z(121)*QND_3p + z(19)*z(158) + z(20)*z(159) - z(20)*z(119)*QND_3p;
z(189) = z(19)*z(125)*QND_3p + z(19)*z(140) + z(20)*z(143) - z(20)*z(123)*QND_3p;
z(190) = z(173)*z(179) + z(175)*z(181) + z(177)*z(183);
z(191) = z(166)*z(186) + z(168)*z(184) + z(170)*z(185);
z(192) = z(172)*z(188) + z(174)*z(189) + z(176)*z(187);
z(193) = z(20)*z(23);
z(194) = z(19)*z(23);
z(195) = z(20)*z(24);
z(196) = z(19)*z(24);
z(197) = z(25)*z(27);
z(198) = z(25)*z(28);
z(199) = z(26)*z(27);
z(200) = z(26)*z(28);
z(201) = z(30)*z(198) - z(26)*z(29);
z(202) = z(26)*z(30) + z(29)*z(198);
z(203) = z(25)*z(29) + z(30)*z(200);
z(204) = z(29)*z(200) - z(25)*z(30);
z(205) = z(27)*z(30);
z(206) = z(27)*z(29);
z(207) = z(19)*z(197) + z(20)*z(199);
z(208) = z(19)*z(201) + z(20)*z(203);
z(209) = z(19)*z(202) + z(20)*z(204);
z(210) = z(194)*z(199) - z(24)*z(28) - z(193)*z(197);
z(211) = z(24)*z(205) + z(194)*z(203) - z(193)*z(201);
z(212) = z(24)*z(206) + z(194)*z(204) - z(193)*z(202);
z(213) = z(195)*z(197) - z(23)*z(28) - z(196)*z(199);
z(214) = z(23)*z(205) + z(195)*z(201) - z(196)*z(203);
z(215) = z(23)*z(206) + z(195)*z(202) - z(196)*z(204);
z(216) = z(25)*z(27)*QNE_2p - z(26)*z(28)*QNE_3p;
z(217) = z(26)*z(30)*QNE_1p + z(29)*z(198)*QNE_1p + z(30)*z(216) - z(25)*z(29)*QNE_3p;
z(218) = z(25)*z(28)*QNE_3p + z(26)*z(27)*QNE_2p;
z(219) = z(29)*z(200)*QNE_1p + z(30)*z(218) - z(25)*z(30)*QNE_1p - z(26)*z(29)*QNE_3p;
z(220) = z(19)*z(203)*QND_3p + z(19)*z(217) + z(20)*z(219) - z(20)*z(201)*QND_3p;
z(221) = z(27)*z(29)*QNE_1p - z(28)*z(30)*QNE_2p;
z(222) = -z(19)*z(24)*QND_1p - z(20)*z(23)*QND_3p;
z(223) = z(19)*z(23)*QND_3p - z(20)*z(24)*QND_1p;
z(224) = z(23)*z(205)*QND_1p + z(24)*z(221) + z(194)*z(219) + z(203)*z(222) - z(193)*z(217) - z(201)*z(223);
z(225) = z(19)*z(24)*QND_3p + z(20)*z(23)*QND_1p;
z(226) = z(19)*z(23)*QND_1p - z(20)*z(24)*QND_3p;
z(227) = z(23)*z(221) + z(195)*z(217) + z(201)*z(225) - z(24)*z(205)*QND_1p - z(196)*z(219) - z(203)*z(226);
z(228) = -z(27)*z(30)*QNE_1p - z(28)*z(29)*QNE_2p;
z(229) = z(26)*z(30)*QNE_3p + z(29)*z(218) - z(25)*z(29)*QNE_1p - z(30)*z(200)*QNE_1p;
z(230) = z(25)*z(30)*QNE_3p + z(26)*z(29)*QNE_1p + z(29)*z(216) - z(30)*z(198)*QNE_1p;
z(231) = z(23)*z(206)*QND_1p + z(24)*z(228) + z(194)*z(229) + z(204)*z(222) - z(193)*z(230) - z(202)*z(223);
z(232) = z(23)*z(228) + z(195)*z(230) + z(202)*z(225) - z(24)*z(206)*QND_1p - z(196)*z(229) - z(204)*z(226);
```

```
z(233) = z(19)*z(204)*QND_3p + z(19)*z(230) + z(20)*z(229) - z(20)*z(202)*QND_3p;
z(234) = -z(25)*z(28)*QNE_2p - z(26)*z(27)*QNE_3p;
z(235) = z(25)*z(27)*QNE_3p - z(26)*z(28)*QNE_2p;
z(236) = z(24)*z(28)*QND_1p + z(195)*z(234) + z(197)*z(225) - z(23)*z(27)*QNE_2p - z(196)*z(235) - z(199)*z(226);
z(237) = z(19)*z(199)*QND_3p + z(19)*z(234) + z(20)*z(235) - z(20)*z(197)*QND_3p;
z(238) = z(194)*z(235) + z(199)*z(222) - z(23)*z(28)*QND_1p - z(24)*z(27)*QNE_2p - z(193)*z(234) - z(197)*z(223);
z(239) = z(209)*z(220) + z(212)*z(224) + z(215)*z(227);
z(240) = z(207)*z(233) + z(210)*z(231) + z(213)*z(232);
z(241) = z(208)*z(237) + z(211)*z(238) + z(214)*z(236);
z(242) = z(31)*z(197) + z(32)*z(199);
z(243) = z(31)*z(199) - z(32)*z(197);
z(244) = z(31)*z(201) + z(32)*z(203);
z(245) = z(31)*z(203) - z(32)*z(201);
z(246) = z(31)*z(202) + z(32)*z(204);
z(247) = z(31)*z(204) - z(32)*z(202);
z(248) = z(34)*z(36);
z(249) = z(34)*z(35);
z(250) = z(33)*z(36);
z(251) = z(33)*z(35);
z(252) = z(28)*z(34) + z(33)*z(242);
z(253) = z(35)*z(243) + z(242)*z(248) - z(28)*z(250);
z(254) = z(242)*z(249) - z(28)*z(251) - z(36)*z(243);
z(255) = z(33)*z(244) - z(34)*z(205);
z(256) = z(35)*z(245) + z(205)*z(250) + z(244)*z(248);
z(257) = z(205)*z(251) + z(244)*z(249) - z(36)*z(245);
z(258) = z(33)*z(246) - z(34)*z(206);
z(259) = z(35)*z(247) + z(206)*z(250) + z(246)*z(248);
z(260) = z(206)*z(251) + z(246)*z(249) - z(36)*z(247);
z(261) = z(31)*z(235) - z(31)*z(197)*QNH_3p - z(32)*z(199)*QNH_3p - z(32)*z(234);
z(262) = z(31)*z(199)*QNH_3p + z(31)*z(234) + z(32)*z(235) - z(32)*z(197)*QNH_3p;
z(263) = z(33)*z(36)*QNH_2p + z(34)*z(35)*QNH_1p;
z(264) = z(33)*z(35)*QNH_1p - z(34)*z(36)*QNH_2p;
z(265) = z(35)*z(261) + z(242)*z(263) + z(248)*z(262) - z(27)*z(250)*QNE_2p - z(36)*z(243)*QNH_1p - z(28)*z(264);
z(266) = z(31)*z(219) - z(31)*z(201)*QNH_3p - z(32)*z(203)*QNH_3p - z(32)*z(217);
z(267) = z(31)*z(203)*QNH_3p + z(31)*z(217) + z(32)*z(219) - z(32)*z(201)*QNH_3p;
z(268) = z(35)*z(266) + z(205)*z(264) + z(244)*z(263) + z(248)*z(267) + z(250)*z(221) - z(36)*z(245)*QNH_1p;
z(269) = z(31)*z(229) - z(31)*z(202)*QNH_3p - z(32)*z(204)*QNH_3p - z(32)*z(230);
z(270) = z(31)*z(204)*QNH_3p + z(31)*z(230) + z(32)*z(229) - z(32)*z(202)*QNH_3p;
z(271) = z(35)*z(269) + z(206)*z(264) + z(246)*z(263) + z(248)*z(270) + z(250)*z(228) - z(36)*z(247)*QNH_1p;
z(272) = -z(33)*z(36)*QNH_1p - z(34)*z(35)*QNH_2p;
z(273) = z(33)*z(35)*QNH_2p - z(34)*z(36)*QNH_1p;
z(274) = z(205)*z(272) + z(244)*z(273) + z(249)*z(267) + z(251)*z(221) - z(35)*z(245)*QNH_1p - z(36)*z(266);
z(275) = z(206)*z(272) + z(246)*z(273) + z(249)*z(270) + z(251)*z(228) - z(35)*z(247)*QNH_1p - z(36)*z(269);
z(276) = z(242)*z(273) + z(249)*z(262) - z(27)*z(251)*QNE_2p - z(35)*z(243)*QNH_1p - z(28)*z(272) - z(36)*z(261);
z(277) = z(33)*z(270) - z(33)*z(206)*QNH_2p - z(34)*z(246)*QNH_2p - z(34)*z(228);
z(278) = z(27)*z(34)*QNE_2p + z(28)*z(33)*QNH_2p + z(33)*z(262) - z(34)*z(242)*QNH_2p;
z(279) = z(33)*z(267) - z(33)*z(205)*QNH_2p - z(34)*z(244)*QNH_2p - z(34)*z(221);
z(280) = z(254)*z(265) + z(257)*z(268) + z(260)*z(271);
z(281) = z(252)*z(276) + z(255)*z(274) + z(258)*z(275);
z(282) = z(253)*z(278) + z(256)*z(279) + z(259)*z(277);
z(283) = z(31)*z(33);
z(284) = z(32)*z(33);
z(285) = z(31)*z(248) - z(32)*z(35);
z(286) = z(31)*z(35) + z(32)*z(248);
z(287) = z(31)*z(249) + z(32)*z(36);
z(288) = z(32)*z(249) - z(31)*z(36);
z(289) = z(37)*z(39);
z(290) = z(37)*z(40);
z(291) = z(38)*z(39);
z(292) = z(38)*z(40);
z(293) = z(42)*z(290) - z(38)*z(41);
z(294) = z(38)*z(42) + z(41)*z(290);
z(295) = z(37)*z(41) + z(42)*z(292);
z(296) = z(41)*z(292) - z(37)*z(42);
z(297) = z(39)*z(42);
z(298) = z(39)*z(41);
z(299) = z(34)*z(40) + z(283)*z(289) + z(284)*z(291);
z(300) = z(283)*z(293) + z(284)*z(295) - z(34)*z(297);
z(301) = z(283)*z(294) + z(284)*z(296) - z(34)*z(298);
z(302) = z(285)*z(289) + z(286)*z(291) - z(40)*z(250);
z(303) = z(250)*z(297) + z(285)*z(293) + z(286)*z(295);
z(304) = z(250)*z(298) + z(285)*z(294) + z(286)*z(296);
z(305) = z(287)*z(289) + z(288)*z(291) - z(40)*z(251);
z(306) = z(251)*z(297) + z(287)*z(293) + z(288)*z(295);
z(307) = z(251)*z(298) + z(287)*z(294) + z(288)*z(296);
z(308) = -z(31)*z(34)*QNH_2p - z(32)*z(33)*QNH_3p;
z(309) = z(37)*z(39)*QNK_2p - z(38)*z(40)*QNK_3p;
z(310) = z(38)*z(42)*QNK_1p + z(41)*z(290)*QNK_1p + z(42)*z(309) - z(37)*z(41)*QNK_3p;
```

```
z(311) = z(31)*z(33)*QNH_3p - z(32)*z(34)*QNH_2p;
z(312) = z(37)*z(40)*QNK_3p + z(38)*z(39)*QNK_2p;
z(313) = z(41)*z(292)*QNK_1p + z(42)*z(312) - z(37)*z(42)*QNK_1p - z(38)*z(41)*QNK_3p;
z(314) = z(39)*z(41)*QNK_1p - z(40)*z(42)*QNK_2p;
z(315) = z(283)*z(310) + z(284)*z(313) + z(293)*z(308) + z(295)*z(311) - z(33)*z(297)*QNH_2p - z(34)*z(314);
z(316) = z(32)*z(36)*QNH_1p + z(31)*z(263) - z(31)*z(35)*QNH_3p - z(32)*z(248)*QNH_3p;
z(317) = z(31)*z(248)*QNH_3p + z(32)*z(263) - z(31)*z(36)*QNH_1p - z(32)*z(35)*QNH_3p;
z(318) = z(250)*z(314) + z(285)*z(310) + z(286)*z(313) + z(293)*z(316) + z(295)*z(317) + z(297)*z(264);
z(319) = z(31)*z(36)*QNH_3p + z(32)*z(35)*QNH_1p + z(31)*z(273) - z(32)*z(249)*QNH_3p;
z(320) = z(31)*z(249)*QNH_3p + z(32)*z(36)*QNH_3p + z(32)*z(273) - z(31)*z(35)*QNH_1p;
z(321) = z(251)*z(314) + z(287)*z(310) + z(288)*z(313) + z(293)*z(319) + z(295)*z(320) + z(297)*z(272);
z(322) = -z(39)*z(42)*QNK_1p - z(40)*z(41)*QNK_2p;
z(323) = z(37)*z(42)*QNK_3p + z(38)*z(41)*QNK_1p + z(41)*z(309) - z(42)*z(290)*QNK_1p;
z(324) = z(38)*z(42)*QNK_3p + z(41)*z(312) - z(37)*z(41)*QNK_1p - z(42)*z(292)*QNK_1p;
z(325) = z(250)*z(322) + z(285)*z(323) + z(286)*z(324) + z(294)*z(316) + z(296)*z(317) + z(298)*z(264);
z(326) = z(251)*z(322) + z(287)*z(323) + z(288)*z(324) + z(294)*z(319) + z(296)*z(320) + z(298)*z(272);
z(327) = z(283)*z(323) + z(284)*z(324) + z(294)*z(308) + z(296)*z(311) - z(33)*z(298)*QNH_2p - z(34)*z(322);
z(328) = -z(37)*z(40)*QNK_2p - z(38)*z(39)*QNK_3p;
z(329) = z(37)*z(39)*QNK_3p - z(38)*z(40)*QNK_2p;
z(330) = z(287)*z(328) + z(288)*z(329) + z(289)*z(319) + z(291)*z(320) - z(39)*z(251)*QNK_2p - z(40)*z(272);
z(331) = z(33)*z(40)*QNH_2p + z(34)*z(39)*QNK_2p + z(283)*z(328) + z(284)*z(329) + z(289)*z(308) + z(291)*z(311);
z(332) = z(285)*z(328) + z(286)*z(329) + z(289)*z(316) + z(291)*z(317) - z(39)*z(250)*QNK_2p - z(40)*z(264);
z(333) = z(301)*z(315) + z(304)*z(318) + z(307)*z(321);
z(334) = z(299)*z(327) + z(302)*z(325) + z(305)*z(326);
z(335) = z(300)*z(331) + z(303)*z(332) + z(306)*z(330);
z(336) = z(43)*z(289) + z(44)*z(291);
z(337) = z(43)*z(291) - z(44)*z(289);
z(338) = z(43)*z(293) + z(44)*z(295);
z(339) = z(43)*z(295) - z(44)*z(293);
z(340) = z(43)*z(294) + z(44)*z(296);
z(341) = z(43)*z(296) - z(44)*z(294);
z(342) = z(46)*z(48);
z(343) = z(46)*z(47);
z(344) = z(45)*z(48);
z(345) = z(45)*z(47);
z(346) = z(40)*z(46) + z(45)*z(336);
z(347) = z(47)*z(337) + z(336)*z(342) - z(40)*z(344);
z(348) = z(336)*z(343) - z(40)*z(345) - z(48)*z(337);
z(349) = z(45)*z(338) - z(46)*z(297);
z(350) = z(47)*z(339) + z(297)*z(344) + z(338)*z(342);
z(351) = z(297)*z(345) + z(338)*z(343) - z(48)*z(339);
z(352) = z(45)*z(340) - z(46)*z(298);
z(353) = z(47)*z(341) + z(298)*z(344) + z(340)*z(342);
z(354) = z(298)*z(345) + z(340)*z(343) - z(48)*z(341);
z(355) = z(43)*z(329) - z(43)*z(289)*QNS_3p - z(44)*z(291)*QNS_3p - z(44)*z(328);
z(356) = z(43)*z(291)*QNS_3p + z(43)*z(328) + z(44)*z(329) - z(44)*z(289)*QNS_3p;
z(357) = z(45)*z(48)*QNS_2p + z(46)*z(47)*QNS_1p;
z(358) = z(45)*z(47)*QNS_1p - z(46)*z(48)*QNS_2p;
z(359) = z(47)*z(355) + z(336)*z(357) + z(342)*z(356) - z(39)*z(344)*QNK_2p - z(48)*z(337)*QNS_1p - z(40)*z(358);
z(360) = z(43)*z(313) - z(43)*z(293)*QNS_3p - z(44)*z(295)*QNS_3p - z(44)*z(310);
z(361) = z(43)*z(295)*QNS_3p + z(43)*z(310) + z(44)*z(313) - z(44)*z(293)*QNS_3p;
z(362) = z(47)*z(360) + z(297)*z(358) + z(338)*z(357) + z(342)*z(361) + z(344)*z(314) - z(48)*z(339)*QNS_1p;
z(363) = z(43)*z(324) - z(43)*z(294)*QNS_3p - z(44)*z(296)*QNS_3p - z(44)*z(323);
z(364) = z(43)*z(296)*QNS_3p + z(43)*z(323) + z(44)*z(324) - z(44)*z(294)*QNS_3p;
z(365) = z(47)*z(363) + z(298)*z(358) + z(340)*z(357) + z(342)*z(364) + z(344)*z(322) - z(48)*z(341)*QNS_1p;
z(366) = -z(45)*z(48)*QNS_1p - z(46)*z(47)*QNS_2p;
z(367) = z(45)*z(47)*QNS_2p - z(46)*z(48)*QNS_1p;
z(368) = z(297)*z(366) + z(338)*z(367) + z(343)*z(361) + z(345)*z(314) - z(47)*z(339)*QNS_1p - z(48)*z(360);
z(369) = z(298)*z(366) + z(340)*z(367) + z(343)*z(364) + z(345)*z(322) - z(47)*z(341)*QNS_1p - z(48)*z(363);
z(370) = z(336)*z(367) + z(343)*z(356) - z(39)*z(345)*QNK_2p - z(47)*z(337)*QNS_1p - z(40)*z(366) - z(48)*z(355);
z(371) = z(45)*z(364) - z(45)*z(298)*QNS_2p - z(46)*z(340)*QNS_2p - z(46)*z(322);
z(372) = z(39)*z(46)*QNK_2p + z(40)*z(45)*QNS_2p + z(45)*z(356) - z(46)*z(336)*QNS_2p;
z(373) = z(45)*z(361) - z(45)*z(297)*QNS_2p - z(46)*z(338)*QNS_2p - z(46)*z(314);
z(374) = z(348)*z(359) + z(351)*z(362) + z(354)*z(365);
z(375) = z(346)*z(370) + z(349)*z(368) + z(352)*z(369);
z(376) = z(347)*z(372) + z(350)*z(373) + z(353)*z(371);
z(377) = z(110) + z(82)*z(69) + z(85)*z(70) + z(88)*z(71);
z(378) = z(111) + z(83)*z(69) + z(86)*z(70) + z(89)*z(71);
z(379) = z(112) + z(84)*z(69) + z(87)*z(70) + z(90)*z(71);
z(380) = z(80)*z(132) + z(81)*z(135) - z(10)*z(129);
z(381) = z(113)*z(129) + z(115)*z(132) + z(117)*z(135);
z(382) = z(114)*z(129) + z(116)*z(132) + z(118)*z(135);
z(383) = z(163) + z(129)*z(377) + z(132)*z(378) + z(135)*z(379);
z(384) = z(80)*z(133) + z(81)*z(136) - z(10)*z(130);
z(385) = z(113)*z(130) + z(115)*z(133) + z(117)*z(136);
z(386) = z(114)*z(130) + z(116)*z(133) + z(118)*z(136);
z(387) = z(164) + z(130)*z(377) + z(133)*z(378) + z(136)*z(379);
z(388) = z(80)*z(134) + z(81)*z(137) - z(10)*z(131);
```

```
z(389) = z(113)*z(131) + z(115)*z(134) + z(117)*z(137);
z(390) = z(114)*z(131) + z(116)*z(134) + z(118)*z(137);
z(391) = z(165) + z(131)*z(377) + z(134)*z(378) + z(137)*z(379);
z(392) = z(119)*z(166) + z(123)*z(168) + z(124)*z(170);
z(393) = z(121)*z(166) + z(125)*z(168) + z(126)*z(170);
z(394) = z(127)*z(168) + z(128)*z(170) - z(16)*z(166);
z(395) = z(166)*z(380) + z(168)*z(384) + z(170)*z(388);
z(396) = z(166)*z(381) + z(168)*z(385) + z(170)*z(389);
z(397) = z(166)*z(382) + z(168)*z(386) + z(170)*z(390);
z(398) = z(190) + z(166)*z(383) + z(168)*z(387) + z(170)*z(391);
z(399) = z(119)*z(172) + z(123)*z(174) + z(124)*z(176);
z(400) = z(121)*z(172) + z(125)*z(174) + z(126)*z(176);
z(401) = z(127)*z(174) + z(128)*z(176) - z(16)*z(172);
z(402) = z(172)*z(380) + z(174)*z(384) + z(176)*z(388);
z(403) = z(172)*z(381) + z(174)*z(385) + z(176)*z(389);
z(404) = z(172)*z(382) + z(174)*z(386) + z(176)*z(390);
z(405) = z(191) + z(172)*z(383) + z(174)*z(387) + z(176)*z(391);
z(406) = z(119)*z(173) + z(123)*z(175) + z(124)*z(177);
z(407) = z(121)*z(173) + z(125)*z(175) + z(126)*z(177);
z(408) = z(127)*z(175) + z(128)*z(177) - z(16)*z(173);
z(409) = z(173)*z(380) + z(175)*z(384) + z(177)*z(388);
z(410) = z(173)*z(381) + z(175)*z(385) + z(177)*z(389);
z(411) = z(173)*z(382) + z(175)*z(386) + z(177)*z(390);
z(412) = z(192) + z(173)*z(383) + z(175)*z(387) + z(177)*z(391);
z(413) = z(19)*z(207) + z(195)*z(213) - z(193)*z(210);
z(414) = z(20)*z(207) + z(194)*z(210) - z(196)*z(213);
z(415) = z(207)*z(392) + z(210)*z(399) + z(213)*z(406);
z(416) = z(207)*z(393) + z(210)*z(400) + z(213)*z(407);
z(417) = z(207)*z(394) + z(210)*z(401) + z(213)*z(408);
z(418) = z(207)*z(395) + z(210)*z(402) + z(213)*z(409);
z(419) = z(207)*z(396) + z(210)*z(403) + z(213)*z(410);
z(420) = z(207)*z(397) + z(210)*z(404) + z(213)*z(411);
z(421) = z(23)*z(213) + z(24)*z(210);
z(422) = z(239) + z(207)*z(398) + z(210)*z(405) + z(213)*z(412);
z(423) = z(19)*z(208) + z(195)*z(214) - z(193)*z(211);
z(424) = z(20)*z(208) + z(194)*z(211) - z(196)*z(214);
z(425) = z(208)*z(392) + z(211)*z(399) + z(214)*z(406);
z(426) = z(208)*z(393) + z(211)*z(400) + z(214)*z(407);
z(427) = z(208)*z(394) + z(211)*z(401) + z(214)*z(408);
z(428) = z(208)*z(395) + z(211)*z(402) + z(214)*z(409);
z(429) = z(208)*z(396) + z(211)*z(403) + z(214)*z(410);
z(430) = z(208)*z(397) + z(211)*z(404) + z(214)*z(411);
z(431) = z(23)*z(214) + z(24)*z(211);
z(432) = z(240) + z(208)*z(398) + z(211)*z(405) + z(214)*z(412);
z(433) = z(19)*z(209) + z(195)*z(215) - z(193)*z(212);
z(434) = z(20)*z(209) + z(194)*z(212) - z(196)*z(215);
z(435) = z(209)*z(392) + z(212)*z(399) + z(215)*z(406);
z(436) = z(209)*z(393) + z(212)*z(400) + z(215)*z(407);
z(437) = z(209)*z(394) + z(212)*z(401) + z(215)*z(408);
z(438) = z(209)*z(395) + z(212)*z(402) + z(215)*z(409);
z(439) = z(209)*z(396) + z(212)*z(403) + z(215)*z(410);
z(440) = z(209)*z(397) + z(212)*z(404) + z(215)*z(411);
z(441) = z(23)*z(215) + z(24)*z(212);
z(442) = z(241) + z(209)*z(398) + z(212)*z(405) + z(215)*z(412);
DABALL = -5.04 + 1000*(0.00117724180595573 6+1.07820905544E-05*z(84)+4.334873076600001E-05*z(88)+6.204882408120001E-
05*z(83)+8.429802986879999E-05*z(85)-0.00114846376986*z(82)-4.554426324096E-06*z(86)-2.342031015720001E-06*z(89)-
7.914128555519999E-07*z(87)-4.069696866400001E-07*z(90))^0.5;
DBCALL = -2.72 + 1000*(0.000890653151714341 1+5.262409064E-07*z(137)+2.1565627842E-06*z(130)+3.0284135772E-
06*z(134)+8.3169536824E-06*z(135)+4.786244312520001E-05*z(132)-0.000885887568060000 1*z(129)-5.605300866E-05*z(131)-1.16514067164E-
07*z(133)-2.0246398568E-08*z(136))^0.5;
DCDALL = -2.79 + 1000*(0.000806388465784150 1+3.966232954000001E-08*z(175)+1.9550864568E-06*z(168)+2.4497691342E-
06*z(176)+3.87172798122E-05*z(172)-0.00080312374824*z(166)-5.081627064E-05*z(170)-1.6292762222E-05*z(173)-1.030896442E-06*z(177)-
9.425151425400001E-08*z(174))^0.5;
DDEALL = -1.58 + 1000*(0.000832463432263385+3.794268000000001E-10*z(212)+1.07287794924E-06*z(214)+3.99046959828E-
05*z(210)+5.28856769808E-05*z(208)-0.00082775466576*z(207)-1.6792443228E-05*z(213)-2.549531822724E-06*z(211)-7.87056E-09*z(209)-
1.59668E-10*z(215))^0.5;


%Ligament forces
if DABALL>=0
FABALL = 17.9*DABALL;
else
FABALL=0;
end

if DBCALL>=0
FBCALL = 17.9*DBCALL;
else
```

```
FBCALL=0;
end

if DCDALL>=0
FCDALL = 16*DCDALL;
else
FCDALL=0;
end

if DDEALL>=0
FDEALL = 16*DDEALL;
else
FDEALL=0;
end


z(443) = 0.000832463432263385 + 3.794268000000001E-10*z(212) + 1.07287794924E-06*z(214) + 3.99046959828E-05*z(210) + 5.28856769808E-
05*z(208) - 0.00082775466576*z(207) - 1.6792443228E-05*z(213) - 2.549531822724E-06*z(211) - 7.87056E-09*z(209) - 1.59668E-10*z(215);
z(444) = z(443)^0.5;
z(445) = 1/z(444);
z(446) = z(445)*FDEALL;
z(447) = 0.000806388465784150 1 + 3.966232954000001E-08*z(175) + 1.9550864568E-06*z(168) + 2.4497691342E-06*z(176) + 3.87172798122E-
05*z(172) - 0.00080312374824*z(166) - 5.081627064E-05*z(170) - 1.6292762222E-05*z(173) - 1.030896442E-06*z(177) - 9.425151425400001E-
08*z(174);
z(448) = z(447)^0.5;
z(449) = 1/z(448);
z(450) = z(449)*FCDALL;
z(451) = 0.0008906531517143411 + 5.262409064E-07*z(137) + 2.1565627842E-06*z(130) + 3.0284135772E-06*z(134) + 8.3169536824E-06*z(135) +
4.786244312520001E-05*z(132) - 0.0008858875680600001*z(129) - 5.605300866E-05*z(131) - 1.16514067164E-07*z(133) - 2.0246398568E-
08*z(136);
z(452) = z(451)^0.5;
z(453) = 1/z(452);
z(454) = z(453)*FBCALL;
z(455) = 0.001177241805955736 + 1.07820905544E-05*z(84) + 4.334873076600001E-05*z(88) + 6.204882408120001E-05*z(83) +
8.429802986879999E-05*z(85) - 0.00114846376986*z(82) - 4.554426324096E-06*z(86) - 2.342031015720001E-06*z(89) - 7.914128555519999E-
07*z(87) - 4.069696866400001E-07*z(90);
z(456) = z(455)^0.5;
z(457) = 1/z(456);
z(458) = z(457)*FABALL;
DABPLL = -6.79 + 1000*(0.000229566871167806+1.14897354172E-05*z(84)+1.38900552309E-05*z(88)+3.079808329858001E-
05*z(85)+7.352330639960001E-05*z(83)-0.000173162940022*z(82)-1.3076567737844E-05*z(86)-5.89758285762E-06*z(89)-2.043519406708E-
06*z(87)-9.216351923399999E-07*z(90))^0.5;
DBCPLL = -2.97 + 1000*(8.166539520760002E-05+7.9060157088E-07*z(137)+4.69036343172E-06*z(135)+5.059093131840001E-
06*z(134)+8.850923806400002E-06*z(130)+3.001383540996001E-05*z(132)-7.068893165220002E-05*z(129)-1.19152345488E-05*z(131)-
3.75801648352E-06*z(133)-5.872779286400001E-07*z(136))^0.5;
DCDPLL = -1.32 + 1000*(6.007137071068201E-05+2.809450987200001E-07*z(177)+1.66674930318E-06*z(173)+1.85455021488E-
06*z(176)+7.227646129600001E-06*z(168)+1.100239973022E-05*z(172)-5.772443582580001E-05*z(166)-9.729955963200001E-06*z(170)-
1.37760466064E-06*z(174)-2.0869280016E-07*z(175))^0.5;
DDEPLL = -1.84 + 1000*(8.3581891017127E-05+5.112161058200001E-07*z(209)+2.18270161902E-06*z(213)+1.440825903358E-
05*z(210)+2.260139590884E-05*z(208)-7.55933836562E-05*z(207)-4.307873930556001E-06*z(211)-6.525981647640001E-07*z(214)-
9.743887253800001E-08*z(212)-1.4760977322E-08*z(215))^0.5;


if DABPLL>=0
FABPLL = 23*DABPLL;
else
FABPLL =0;
end

if DBCPLL>=0
FBCPLL = 23*DBCPLL;
else
FBCPLL =0;
end

if DCDPLL>=0
FCDPLL = 25.4*DCDPLL;
else
FCDPLL =0;
end

if DDEPLL>=0
FDEPLL = 25.4*DDEPLL;
else
FDEPLL =0;
end
```

```
z(459) = 8.3581891017127E-05 + 5.112161058200001E-07*z(209) + 2.18270161902E-06*z(213) + 1.440825903358E-05*z(210) + 2.260139590884E-
05*z(208) - 7.55933836562E-05*z(207) - 4.307873930556001E-06*z(211) - 6.525981647640001E-07*z(214) - 9.743887253800001E-08*z(212) -
1.4760977322E-08*z(215);
z(460) = z(459)^0.5;
z(461) = 1/z(460);
z(462) = z(461)*FDEPLL;
z(463) = 6.007137071068201E-05 + 2.809450987200001E-07*z(177) + 1.66674930318E-06*z(173) + 1.85455021488E-06*z(176) +
7.227646129600001E-06*z(168) + 1.100239973022E-05*z(172) - 5.772443582580001E-05*z(166) - 9.729955963200001E-06*z(170) -
1.37760466064E-06*z(174) - 2.0869280016E-07*z(175);
z(464) = z(463)^0.5;
z(465) = 1/z(464);
z(466) = z(465)*FCDPLL;
z(467) = 8.166539520760002E-05 + 7.9060157088E-07*z(137) + 4.69036343172E-06*z(135) + 5.059093131840001E-06*z(134) +
8.850923806400002E-06*z(130) + 3.001383540996001E-05*z(132) - 7.068893165220002E-05*z(129) - 1.19152345488E-05*z(131) -
3.75801648352E-06*z(133) - 5.872779286400001E-07*z(136);
z(468) = z(467)^0.5;
z(469) = 1/z(468);
z(470) = z(469)*FBCPLL;
z(471) = 0.000229566871167806 + 1.14897354172E-05*z(84) + 1.38900552309E-05*z(88) + 3.079808329858001E-05*z(85) + 7.352330639960001E-
05*z(83) - 0.000173162940022*z(82) - 1.3076567737844E-05*z(86) - 5.89758285762E-06*z(89) - 2.043519406708E-06*z(87) -
9.216351923399999E-07*z(90);
z(472) = z(471)^0.5;
z(473) = 1/z(472);
z(474) = z(473)*FABPLL;
DABIL = -7.72 + 1000*(0.000495767730741664 1+4.236495186E-07*z(90)+4.38660737592E-06*z(87)+6.429091376E-
06*z(88)+6.656894062720001E-05*z(85)+0.0002397453988302*z(83)-0.0004117653648224*z(82)-3.87589597956E-05*z(86)-2.713356964164E-
05*z(84)-3.743260623E-06*z(89))^0.5;
DBCIL = -6.68 + 1000*(0.000499222747374265+1.5439866058E-07*z(134)+1.3973915256E-05*z(136)+3.82379502192E-
05*z(132)+0.00025560282237*z(130)-0.0004243594548768*z(129)-2.319991229184E-05*z(135)-2.3031719655E-05*z(133)-1.71349486732E-
06*z(131)-9.367749481600001E-08*z(137))^0.5;
DCDIL = -8.710000000000001 + 1000*(0.000473879991708172+9.057318815999999E-07*z(176)+1.10635858788E-
06*z(174)+0.00021090634929*z(172)-0.00041770279719*z(166)-3.400376187600001E-05*z(173)-2.19115772668E-06*z(168)-1.7938138976E-
06*z(170)-1.78374686672E-07*z(175)-1.4602827904E-07*z(177))^0.5;
DDEIL = -10.12 + 1000*(0.0005641770587438731+1.995436443336E-05*z(211)+1.99678736922E-05*z(214)+0.0002984411659824*z(208)-
0.0004670046645744*z(207)-3.12459915682E-05*z(213)-3.122485210216E-05*z(210)-8.057134372320001E-06*z(209)-5.390805954600001E-
07*z(215)-5.38715880648E-07*z(212))^0.5;


if DABIL>=0
FABIL = 6.36*DABIL;
else
FABIL =0;
end

if DBCIL>=0
FBCIL = 6.36*DBCIL;
else
FBCIL =0;
end

if DCDIL>=0
FCDIL = 7.74*DCDIL;
else
FCDIL =0;
end

if DDEIL>=0
FDEIL = 7.74*DDEIL;
else
FDEIL =0;
end


z(475) = 0.0005641770587438731 + 1.995436443336E-05*z(211) + 1.99678736922E-05*z(214) + 0.0002984411659824*z(208) -
0.0004670046645744*z(207) - 3.12459915682E-05*z(213) - 3.122485210216E-05*z(210) - 8.057134372320001E-06*z(209) - 5.390805954600001E-
07*z(215) - 5.38715880648E-07*z(212);
z(476) = z(475)^0.5;
z(477) = 1/z(476);
z(478) = z(477)*FDEIL;
z(479) = 0.000473879991708172 + 9.057318815999999E-07*z(176) + 1.10635858788E-06*z(174) + 0.00021090634929*z(172) -
0.00041770279719*z(166) - 3.400376187600001E-05*z(173) - 2.19115772668E-06*z(168) - 1.7938138976E-06*z(170) - 1.78374686672E-07*z(175) -
1.4602827904E-07*z(177);
z(480) = z(479)^0.5;
z(481) = 1/z(480);
z(482) = z(481)*FCDIL;
z(483) = 0.000499222747374265 + 1.5439866058E-07*z(134) + 1.3973915256E-05*z(136) + 3.82379502192E-05*z(132) + 0.00025560282237*z(130)
- 0.0004243594548768*z(129) - 2.319991229184E-05*z(135) - 2.3031719655E-05*z(133) - 1.71349486732E-06*z(131) - 9.367749481600001E-
08*z(137);
```

```
z(484) = z(483)^0.5;
z(485) = 1/z(484);
z(486) = z(485)*FBCIL;
z(487) = 0.0004957677307416641 + 4.236495186E-07*z(90) + 4.38660737592E-06*z(87) + 6.429091376E-06*z(88) + 6.656894062720001E-05*z(85)
+ 0.0002397453988302*z(83) - 0.0004117653648224*z(82) - 3.87589597956E-05*z(86) - 2.713356964164E-05*z(84) - 3.743260623E-06*z(89);
z(488) = z(487)^0.5;
z(489) = 1/z(488);
z(490) = z(489)*FABIL;
DABLLF = -13.79 +
4.00478818106526*(14.90504557971311+z(83)+1.704124810873825*z(84)+4.239214832571488*z(86)+6.26256645601244*z(88)+7.2241511748094*
z(87)-7.287972298080623*z(85)-6.207724860924508*z(90)-3.642764204425479*z(89)-1.719179750477466*z(82))^0.5;
DBCLLF = -8.69 +
6.098058152323575*(6.479676286527102+z(133)+1.420663527282674*z(130)+2.018884198663451*z(134)+2.478607770865086*z(135)+2.86815514
6848473*z(131)-2.691589896936082*z(137)-2.641201634438345*z(129)-1.85913242911938*z(132)-1.333206678579177*z(136))^0.5;
DCDLLF = -7.94 +
4.946015962570279*(10.79457055952518+z(174)+1.784859886888961*z(172)+2.289867165524945*z(175)+4.087092050049598*z(173)+5.58498744
9468074*z(170)-5.331235692885248*z(166)-4.281626084326089*z(177)-2.986921120277782*z(168)-1.869814174720715*z(176))^0.5;
DDELLF = -11.21 +
8.070651768951501*(5.005544742753542+z(211)+1.377998088211804*z(208)+1.575266182037215*z(212)+1.791391269873237*z(213)+2.17071378
7271991*z(209)-2.155739952053178*z(207)-1.803834328073255*z(215)-1.564399813392071*z(210)-1.145098110174906*z(214))^0.5;


if DABLLF>=0
FABLLF = 21.6*DABLLF;
else
FABLLF =0;
end

if DBCLLF>=0
FBCLLF = 21.6*DBCLLF;
else
FBCLLF =0;
end

if DCDLLF>=0
FCDLLF = 25*DCDLLF;
else
FCDLLF =0;
end

if DDELLF>=0
FDELLF = 25*DDELLF;
else
FDELLF =0;
end


z(491) = 0.000326038259026309 + 6.513541997568E-05*z(211) + 8.975648420136001E-05*z(208) + 0.00010260562434048*z(212) +
0.00011668302270396*z(213) + 0.00014139035418096*z(209) - 0.00014041502713533 6*z(207) - 0.0001174935065256*z(215) -
0.000101897838855168*z(210) - 7.458644631960001E-05*z(214);
z(492) = z(491)^0.5;
z(493) = 1/z(492);
z(494) = z(493)*FDELLF;
z(495) = 0.000264068377338018 + 2.4463073902E-05*z(174) + 4.366315931768E-05*z(172) + 5.6017189696E-05*z(175) + 9.998283486463999E-
05*z(173) + 0.00013662596071808*z(170) - 0.000130418412744032*z(166) - 0.0001047417353216*z(177) - 7.30692721048E-05*z(168) -
4.57414023392E-05*z(176);
z(496) = z(495)^0.5;
z(497) = 1/z(496);
z(498) = z(497)*FCDLLF;
z(499) = 0.000240955272014098 + 3.718631322912001E-05*z(133) + 5.282923891872001E-05*z(130) + 7.507486018482001E-05*z(134) +
9.217028493952001E-05*z(135) + 0.00010665611568042*z(131) - 0.0001000903049918*z(137) - 9.8216551279488E-05*z(129) -
6.913428084364801E-05*z(132) - 4.957704114880001E-05*z(136);
z(500) = z(499)^0.5;
z(501) = 1/z(500);
z(502) = z(501)*FBCLLF;
z(503) = 0.000239052015454762 + 1.60383283752E-05*z(83) + 2.733131330912E-05*z(84) + 6.798991953780001E-05*z(86) +
0.00010044109729304*z(88) + 0.00011586330877368*z(87) - 0.000116886892905978*z(85) - 9.95615297824E-05*z(90) - 5.842384850400001E-
05*z(89) - 2.757276937415 2E-05*z(82);
z(504) = z(503)^0.5;
z(505) = 1/z(504);
z(506) = z(505)*FABLLF;
DABRLF = -13.85 + 5.251179391887121*(9.96246228108277+z(83)+1.999895354633051*z(89)+2.575607348552605*z(86)-
5.447403833798548*z(87)-4.446741955619756*z(85)-4.229774242625199*z(90)-3.452785062635181*z(88)-2.114997783672183*z(84)-
1.726482865534088*z(82))^0.5;
DBCRLF = -11.45 + 6.637928716970378*(6.567131192781466+z(130)+1.281270258689456*z(136)+1.291407335358 66*z(133)-
2.757691478018384*z(132)-2.736044605511718*z(135)-2.649965414980761*z(134)-2.629164152786492*z(137)-2.135415683737382*z(129)-
2.051998112775773*z(131))^0.5;
```

```
DCDRLF = -9.199999999999999 + 5.814310484619823*(8.490728237839345+z(174)+1.439333735483592*z(176)+1.453070112742449*z(172)-
3.6929023983999*z(166)-3.667252644313695*z(173)-3.657992107368711*z(170)-3.632584829330726*z(177)-2.541448183412229*z(168)-
2.523796072986673*z(175))^0.5;
DDERLF = -11.39 + 8.20462494257233*(5.222438157703784+z(211)+1.677448684853561*z(214)+1.749405305380686*z(208)-
2.258686051931204*z(207)-2.165781672009162*z(213)-1.976379803287107*z(209)-1.895087257137169*z(215)-1.291116498266075*z(210)-
1.129743803341804*z(212))^0.5;


if DABRLF>=0
FABRLF = 21.6*DABRLF;
else
FABRLF =0;
end

if DBCRLF>=0
FBCRLF = 21.6*DBCRLF;
else
FBCRLF =0;
end

if DCDRLF>=0
FCDRLF = 25*DCDRLF;
else
FCDRLF =0;
end

if DDERLF>=0
FDERLF = 25*DDERLF;
else
FDERLF  =0;
end


z(507) = 0.000351552970448142 + 6.731587044828001E-05*z(211) + 0.000112918918353 24*z(214) + 0.00011776274089854*z(208) -
0.000152045417655138*z(207) - 0.000145791478452228*z(213) - 0.000133041726794672*z(209) - 0.000127569448289632*z(215) -
8.691263093091602E-05*z(210) - 7.604968750550401E-05*z(212);
z(508) = z(507)^0.5;
z(509) = 1/z(508);
z(510) = z(509)*FDERLF;
z(511) = 0.000287039311392858 + 3.380620641156E-05*z(174) + 4.865841335688E-05*z(176) + 4.912278816184E-05*z(172) -
0.000124843020738052*z(166) - 0.000123975899857008*z(173) - 0.000123662836233564*z(170) - 0.000122803912547856*z(177) -
8.5916721872718E-05*z(168) - 8.5319970984072E-05*z(175);
z(512) = z(511)^0.5;
z(513) = 1/z(512);
z(514) = z(513)*FCDRLF;
z(515) = 0.000289361575907074 + 4.406209765158E-05*z(130) + 5.645545525644E-05*z(136) + 5.690211611854E-05*z(133) -
0.000121509671197376*z(132) - 0.000120555864587136*z(135) - 0.000116763034888192*z(134) - 0.000115846487642112*z(137) -
9.409089438355199E-05*z(129) - 9.0415341225984E-05*z(131);
z(516) = z(515)^0.5;
z(517) = 1/z(516);
z(518) = z(517)*FBCRLF;
z(519) = 0.000274713751775278 + 2.757488500578E-05*z(83) + 5.514688442759999E-05*z(89) + 7.102207645638E-05*z(86) -
0.000150211534297 04*z(87) - 0.000122618398076592*z(85) - 0.0001166355383408*z(90) - 9.521015105184E-05*z(88) - 5.832082067224001E-
05*z(84) - 4.7607566481552E-05*z(82);
z(520) = z(519)^0.5;
z(521) = 1/z(520);
z(522) = z(521)*FABRLF;
z(574) = z(115)*z(564) - z(117)*z(563);
z(570) = z(117)*z(562) - z(113)*z(564);
z(566) = z(113)*z(563) - z(115)*z(562);
z(742) = z(129)*z(574) + z(132)*z(570) + z(135)*z(566);
z(807) = z(385)*z(788) - z(389)*z(787);
z(811) = z(742) + z(807);
z(746) = z(130)*z(574) + z(133)*z(570) + z(136)*z(566);
z(800) = z(389)*z(786) - z(381)*z(788);
z(815) = z(746) + z(800);
z(750) = z(131)*z(574) + z(134)*z(570) + z(137)*z(566);
z(793) = z(381)*z(787) - z(385)*z(786);
z(819) = z(750) + z(793);
z(1152) = z(166)*z(811) + z(168)*z(815) + z(170)*z(819);
z(1256) = z(403)*z(1228) - z(410)*z(1227);
z(1263) = z(1152) + z(1256);
z(1162) = z(172)*z(811) + z(174)*z(815) + z(176)*z(819);
z(1245) = z(410)*z(1226) - z(396)*z(1228);
z(1270) = z(1162) + z(1245);
z(1172) = z(173)*z(811) + z(175)*z(815) + z(177)*z(819);
z(1235) = z(396)*z(1227) - z(403)*z(1226);
z(1277) = z(1172) + z(1235);
```

```
z(1810) = z(209)*z(1263) + z(212)*z(1270) + z(215)*z(1277);
z(2038) = z(419)*z(2027) - z(429)*z(2026);
z(2095) = z(1810) + z(2038);
z(1794) = z(208)*z(1263) + z(211)*z(1270) + z(214)*z(1277);
z(2051) = z(439)*z(2026) - z(419)*z(2028);
z(2085) = z(1794) + z(2051);
z(1778) = z(207)*z(1263) + z(210)*z(1270) + z(213)*z(1277);
z(2064) = z(429)*z(2028) - z(439)*z(2027);
z(2075) = z(1778) + z(2064);
z(836) = z(389)*z(822) - z(381)*z(824);
z(851) = z(746) + z(836);
z(843) = z(385)*z(824) - z(389)*z(823);
z(847) = z(742) + z(843);
z(829) = z(381)*z(823) - z(385)*z(822);
z(855) = z(750) + z(829);
z(901) = z(381)*z(895) - z(385)*z(894);
z(927) = z(750) + z(901);
z(915) = z(385)*z(896) - z(389)*z(895);
z(919) = z(742) + z(915);
z(908) = z(389)*z(894) - z(381)*z(896);
z(923) = z(746) + z(908);
z(584) = z(113)*z(581) - z(115)*z(580);
z(588) = z(117)*z(580) - z(113)*z(582);
z(592) = z(115)*z(582) - z(117)*z(581);
z(2110) = z(419)*z(2099) - z(429)*z(2098);
z(2167) = z(1810) + z(2110);
z(2136) = z(429)*z(2100) - z(439)*z(2099);
z(2147) = z(1778) + z(2136);
z(2123) = z(439)*z(2098) - z(419)*z(2100);
z(2157) = z(1794) + z(2123);
z(602) = z(113)*z(599) - z(115)*z(598);
z(606) = z(117)*z(598) - z(113)*z(600);
z(610) = z(115)*z(600) - z(117)*z(599);
z(1397) = z(396)*z(1389) - z(403)*z(1388);
z(1439) = z(1172) + z(1397);
z(1418) = z(403)*z(1390) - z(410)*z(1389);
z(1425) = z(1152) + z(1418);
z(1407) = z(410)*z(1388) - z(396)*z(1390);
z(1432) = z(1162) + z(1407);
z(872) = z(389)*z(858) - z(381)*z(860);
z(887) = z(746) + z(872);
z(879) = z(385)*z(860) - z(389)*z(859);
z(883) = z(742) + z(879);
z(865) = z(381)*z(859) - z(385)*z(858);
z(891) = z(750) + z(865);
z(718) = z(115)*z(708) - z(117)*z(707);
z(710) = z(113)*z(707) - z(115)*z(706);
z(714) = z(117)*z(706) - z(113)*z(708);
z(1667) = z(396)*z(1659) - z(403)*z(1658);
z(1709) = z(1172) + z(1667);
z(1688) = z(403)*z(1660) - z(410)*z(1659);
z(1695) = z(1152) + z(1688);
z(1677) = z(410)*z(1658) - z(396)*z(1660);
z(1702) = z(1162) + z(1677);
z(1081) = z(381)*z(1075) - z(385)*z(1074);
z(1107) = z(750) + z(1081);
z(1095) = z(385)*z(1076) - z(389)*z(1075);
z(1099) = z(742) + z(1095);
z(1088) = z(389)*z(1074) - z(381)*z(1076);
z(1103) = z(746) + z(1088);
z(2254) = z(419)*z(2243) - z(429)*z(2242);
z(2311) = z(1810) + z(2254);
z(2280) = z(429)*z(2244) - z(439)*z(2243);
z(2291) = z(1778) + z(2280);
z(2267) = z(439)*z(2242) - z(419)*z(2244);
z(2301) = z(1794) + z(2267);
z(2790) = z(44)*z(45);
z(2789) = z(43)*z(343) + z(44)*z(48);
z(2792) = z(44)*z(343) - z(43)*z(48);
z(2870) = z(197)*z(2789) + z(199)*z(2792) - z(28)*z(345);
z(2873) = z(201)*z(2789) + z(203)*z(2792) + z(205)*z(345);
z(2876) = z(202)*z(2789) + z(204)*z(2792) + z(206)*z(345);
z(2899) = -0.07100000000000001*z(419)*z(2870) - 0.07100000000000001*z(429)*z(2873) - 0.07100000000000001*z(439)*z(2876);
z(2731) = z(197)*z(293) + z(199)*z(295) - z(28)*z(297);
z(2734) = z(201)*z(293) + z(203)*z(295) + z(205)*z(297);
z(2737) = z(202)*z(293) + z(204)*z(295) + z(206)*z(297);
z(2732) = z(197)*z(294) + z(199)*z(296) - z(28)*z(298);
z(2735) = z(201)*z(294) + z(203)*z(296) + z(205)*z(298);
```

z(2738) = z(202)*z(294) + z(204)*z(296) + z(206)*z(298);
z(2774) = 0.0005*z(419)*z(2731) + 0.0005*z(429)*z(2734) + 0.0005*z(439)*z(2737) - 0.00893*z(419)*z(2732) - 0.00893*z(429)*z(2735) - 0.00893*z(439)*z(2738);
z(2730) = z(28)*z(40) + z(197)*z(289) + z(199)*z(291);
z(2733) = z(201)*z(289) + z(203)*z(291) - z(40)*z(205);
z(2736) = z(202)*z(289) + z(204)*z(291) - z(40)*z(206);
z(2761) = -0.01653*z(419)*z(2732) - 0.01653*z(429)*z(2735) - 0.01653*z(439)*z(2738) - 0.0005*z(419)*z(2730) - 0.0005*z(429)*z(2733) - 0.0005*z(439)*z(2736);
z(2748) = 0.00893*z(419)*z(2730) + 0.00893*z(429)*z(2733) + 0.00893*z(439)*z(2736) + 0.01653*z(419)*z(2731) + 0.01653*z(429)*z(2734) + 0.01653*z(439)*z(2737);
z(2778) = z(299)*z(346) + z(300)*z(349) + z(301)*z(352);
z(1920) = z(429)*z(1884) - z(439)*z(1883);
z(1931) = z(1778) + z(1920);
z(1907) = z(439)*z(1882) - z(419)*z(1884);
z(1941) = z(1794) + z(1907);
z(1894) = z(419)*z(1883) - z(429)*z(1882);
z(1951) = z(1810) + z(1894);
z(2404) = z(252)*z(1931) + z(255)*z(1941) + z(258)*z(1951);
z(2567) = z(253)*z(419)*z(2531) + z(256)*z(429)*z(2531) + z(259)*z(439)*z(2531) - z(254)*z(419)*z(2530) - z(257)*z(429)*z(2530) - z(260)*z(439)*z(2530);
z(2581) = z(2404) + z(2567);
z(2648) = z(254)*z(419)*HOHK2 + z(257)*z(429)*HOHK2 + z(260)*z(439)*HOHK2 - z(253)*z(419)*HOHK3 - z(256)*z(429)*HOHK3 - z(259)*z(439)*HOHK3;
z(2701) = z(2581) + z(2648);
z(2781) = z(302)*z(346) + z(303)*z(349) + z(304)*z(352);
z(2426) = z(253)*z(1931) + z(256)*z(1941) + z(259)*z(1951);
z(2554) = z(254)*z(419)*z(2529) + z(257)*z(429)*z(2529) + z(260)*z(439)*z(2529) - z(252)*z(419)*z(2531) - z(255)*z(429)*z(2531) - z(258)*z(439)*z(2531);
z(2594) = z(2426) + z(2554);
z(2635) = z(252)*z(419)*HOHK3 + z(255)*z(429)*HOHK3 + z(258)*z(439)*HOHK3 - z(254)*z(419)*HOHK1 - z(257)*z(429)*HOHK1 - z(260)*z(439)*HOHK1;
z(2714) = z(2594) + z(2635);
z(2784) = z(305)*z(346) + z(306)*z(349) + z(307)*z(352);
z(2448) = z(254)*z(1931) + z(257)*z(1941) + z(260)*z(1951);
z(2541) = z(252)*z(419)*z(2530) + z(255)*z(429)*z(2530) + z(258)*z(439)*z(2530) - z(253)*z(419)*z(2529) - z(256)*z(429)*z(2529) - z(259)*z(439)*z(2529);
z(2607) = z(2448) + z(2541);
z(2622) = z(253)*z(419)*HOHK1 + z(256)*z(429)*HOHK1 + z(259)*z(439)*HOHK1 - z(252)*z(419)*HOHK2 - z(255)*z(429)*HOHK2 - z(258)*z(439)*HOHK2;
z(2727) = z(2607) + z(2622);
z(2787) = z(43)*z(45);
z(2687) = z(28)*z(419)*z(2611) + z(199)*z(419)*z(2612) + z(203)*z(429)*z(2612) + z(204)*z(439)*z(2612) - z(205)*z(429)*z(2611) - z(206)*z(439)*z(2611);
z(2674) = z(205)*z(429)*z(2610) + z(206)*z(439)*z(2610) - z(28)*z(419)*z(2610) - z(197)*z(419)*z(2612) - z(201)*z(429)*z(2612) - z(202)*z(439)*z(2612);
z(2661) = z(197)*z(419)*z(2611) + z(201)*z(429)*z(2611) + z(202)*z(439)*z(2611) - z(199)*z(419)*z(2610) - z(203)*z(429)*z(2610) - z(204)*z(439)*z(2610);
z(2802) = z(346)*z(2774) + z(349)*z(2761) + z(352)*z(2748) + z(2778)*z(2701) + z(2781)*z(2714) + z(2784)*z(2727) + z(2787)*z(2687) + z(2790)*z(2674) - z(46)*z(2661);
z(2912) = z(2899) + z(2802);
z(2791) = z(43)*z(47) + z(44)*z(342);
z(2779) = z(299)*z(347) + z(300)*z(350) + z(301)*z(353);
z(2782) = z(302)*z(347) + z(303)*z(350) + z(304)*z(353);
z(2785) = z(305)*z(347) + z(306)*z(350) + z(307)*z(353);
z(2788) = z(43)*z(342) - z(44)*z(47);
z(2827) = z(347)*z(2774) + z(350)*z(2761) + z(353)*z(2748) + z(344)*z(2661) + z(2779)*z(2701) + z(2782)*z(2714) + z(2785)*z(2727) + z(2788)*z(2687) + z(2791)*z(2674);
z(2868) = z(28)*z(46) + z(197)*z(2787) + z(199)*z(2790);
z(2871) = z(201)*z(2787) + z(203)*z(2790) - z(46)*z(205);
z(2874) = z(202)*z(2787) + z(204)*z(2790) - z(46)*z(206);
z(2886) = 0.07100000000000001*z(419)*z(2868) + 0.07100000000000001*z(429)*z(2871) + 0.07100000000000001*z(439)*z(2874);
z(2780) = z(299)*z(348) + z(300)*z(351) + z(301)*z(354);
z(2783) = z(302)*z(348) + z(303)*z(351) + z(304)*z(354);
z(2786) = z(305)*z(348) + z(306)*z(351) + z(307)*z(354);
z(2852) = z(348)*z(2774) + z(351)*z(2761) + z(354)*z(2748) + z(345)*z(2661) + z(2780)*z(2701) + z(2783)*z(2714) + z(2786)*z(2727) + z(2789)*z(2687) + z(2792)*z(2674);
z(2925) = z(2886) + z(2852);
z(556) = z(117)*BOBA2 - z(115)*BOBA3;
z(552) = z(113)*BOBA3 - z(117)*BOBA1;
z(548) = z(115)*BOBA1 - z(113)*BOBA2;
z(771) = z(389)*COCB2 - z(385)*COCB3;
z(775) = z(742) + z(771);
z(764) = z(381)*COCB3 - z(389)*COCB1;
z(779) = z(746) + z(764);
z(757) = z(385)*COCB1 - z(381)*COCB2;
z(783) = z(750) + z(757);
z(1848) = z(439)*EOED2 - z(429)*EOED3;
z(1859) = z(1778) + z(1848);

z(1835) = z(419)*EOED3 - z(439)*EOED1;
z(1869) = z(1794) + z(1835);
z(1822) = z(429)*EOED1 - z(419)*EOED2;
z(1879) = z(1810) + z(1822);
z(2486) = z(254)*z(419)*HOHE2 + z(257)*z(429)*HOHE2 + z(260)*z(439)*HOHE2 - z(253)*z(419)*HOHE3 - z(256)*z(429)*HOHE3 - z(259)*z(439)*HOHE3;
z(2500) = z(2404) + z(2486);
z(2473) = z(252)*z(419)*HOHE3 + z(255)*z(429)*HOHE3 + z(258)*z(439)*HOHE3 - z(254)*z(419)*HOHE1 - z(257)*z(429)*HOHE1 - z(260)*z(439)*HOHE1;
z(2513) = z(2426) + z(2473);
z(2460) = z(253)*z(419)*HOHE1 + z(256)*z(429)*HOHE1 + z(259)*z(439)*HOHE1 - z(252)*z(419)*HOHE2 - z(255)*z(429)*HOHE2 - z(258)*z(439)*HOHE2;
z(2526) = z(2448) + z(2460);
z(1202) = z(410)*DODC2 - z(403)*DODC3;
z(1209) = z(1152) + z(1202);
z(1191) = z(396)*DODC3 - z(410)*DODC1;
z(1216) = z(1162) + z(1191);
z(1181) = z(403)*DODC1 - z(396)*DODC2;
z(1223) = z(1172) + z(1181);
z(678) = z(117)*z(670) - z(113)*z(672);
z(674) = z(113)*z(671) - z(115)*z(670);
z(682) = z(115)*z(672) - z(117)*z(671);
z(1289) = z(396)*z(1281) - z(403)*z(1280);
z(1331) = z(1172) + z(1289);
z(1310) = z(403)*z(1282) - z(410)*z(1281);
z(1317) = z(1152) + z(1310);
z(1299) = z(410)*z(1280) - z(396)*z(1282);
z(1324) = z(1162) + z(1299);
z(2195) = z(439)*z(2170) - z(419)*z(2172);
z(2229) = z(1794) + z(2195);
z(2182) = z(419)*z(2171) - z(429)*z(2170);
z(2239) = z(1810) + z(2182);
z(2208) = z(429)*z(2172) - z(439)*z(2171);
z(2219) = z(1778) + z(2208);
z(1016) = z(389)*z(1002) - z(381)*z(1004);
z(1031) = z(746) + z(1016);
z(1009) = z(381)*z(1003) - z(385)*z(1002);
z(1035) = z(750) + z(1009);
z(1023) = z(385)*z(1004) - z(389)*z(1003);
z(1027) = z(742) + z(1023);
z(1569) = z(410)*z(1550) - z(396)*z(1552);
z(1594) = z(1162) + z(1569);
z(1559) = z(396)*z(1551) - z(403)*z(1550);
z(1601) = z(1172) + z(1559);
z(1580) = z(403)*z(1552) - z(410)*z(1551);
z(1587) = z(1152) + z(1580);
z(1966) = z(419)*z(1955) - z(429)*z(1954);
z(2023) = z(1810) + z(1966);
z(1992) = z(429)*z(1956) - z(439)*z(1955);
z(2003) = z(1778) + z(1992);
z(1979) = z(439)*z(1954) - z(419)*z(1956);
z(2013) = z(1794) + z(1979);
z(624) = z(117)*z(616) - z(113)*z(618);
z(628) = z(115)*z(618) - z(117)*z(617);
z(620) = z(113)*z(617) - z(115)*z(616);
z(1364) = z(403)*z(1336) - z(410)*z(1335);
z(1371) = z(1152) + z(1364);
z(1353) = z(410)*z(1334) - z(396)*z(1336);
z(1378) = z(1162) + z(1353);
z(1343) = z(396)*z(1335) - z(403)*z(1334);
z(1385) = z(1172) + z(1343);
z(2941) = 4.7299E-05*z(462)*(z(2095)+44.21104040254551*z(2085)+114.2537897207129*z(207)*z(2075)+114.2537897207129*z(208)*z(2085)+114.2537897207129*z(209)*z(2095)-147.8697224042791*z(2075)-21.77701431319901*z(210)*z(2075)-21.77701431319901*z(211)*z(2085)-21.77701431319901*z(212)*z(2095)-3.298991522019493*z(213)*z(2075)-3.298991522019493*z(214)*z(2085)-3.298991522019493*z(215)*z(2095)) + 4.9681E-05*z(454)*(z(851)+436.869225659709*z(129)*z(847)+436.869225659709*z(130)*z(851)+436.869225659709*z(131)*z(855)-410.7868199110324*z(847)-25.99182786175802*z(855)-23.60302731426501*z(132)*z(847)-23.60302731426501*z(133)*z(851)-23.60302731426501*z(134)*z(855)-4.101447233348765*z(135)*z(847)-4.101447233348765*z(136)*z(851)-4.101447233348765*z(137)*z(855)) + 5.641700000000001E-05*z(486)*(z(927)+247.6572664267862*z(919)+24.25456865838311*z(132)*z(919)+24.25456865838311*z(133)*z(923)+24.25456865838311*z(134)*z(927)-149.1704628037648*z(923)-269.1738305829803*z(129)*z(919)-269.1738305829803*z(130)*z(923)-269.1738305829803*z(131)*z(927)-14.71584805998192*z(135)*z(919)-14.71584805998192*z(136)*z(923)-14.71584805998192*z(137)*z(927)) + 0.000203764*z(458)*(z(584)+5.754804577844959*z(588)+129.8428574232936*z(82)*z(592)+129.8428574232936*z(83)*z(588)+129.8428574232936*z(84)*z(584)-106.5158713020946*z(592)-9.530554955733102*z(85)*z(592)-9.530554955733102*z(86)*z(588)-9.530554955733102*z(87)*z(584)-4.900914783769459*z(88)*z(592)-4.900914783769459*z(89)*z(588)-4.900914783769459*z(90)*z(584)) + 0.000263142*z(478)*(z(2167)+57.96163288262611*z(2147)-37.04060925279887*z(2157)-58.17953804409786*z(207)*z(2147)-58.17953804409786*z(208)*z(2157)-58.17953804409786*z(209)*z(2167)-3.892632114979745*z(213)*z(2147)-3.892632114979745*z(214)*z(2157)-3.892632114979745*z(215)*z(2167)-3.889998555912777*z(210)*z(2147)-3.889998555912777*z(211)*z(2157)-3.889998555912777*z(212)*z(2167))

```
+
0.000439106*z(474)*(z(602)+6.399042600192209*z(606)+29.7948559117844*z(82)*z(610)+29.7948559117844*z(83)*z(606)+29.7948559117844*z(
84)*z(602)-15.07109900570705*z(610)-5.299196549352549*z(85)*z(610)-5.299196549352549*z(86)*z(606)-5.299196549352549*z(87)*z(602)-
2.389958233319518*z(88)*z(610)-2.389958233319518*z(89)*z(606)-2.389958233319518*z(90)*z(602)) + 6.128E-
05*z(482)*(19.44327676240209*z(1439)+238.8415469973891*z(1425)-120.5957898172324*z(1432)-232.8573759791123*z(166)*z(1425)-
232.8573759791123*z(172)*z(1432)-232.8573759791123*z(173)*z(1439)-1.221507832898173*z(168)*z(1425)-1.221507832898173*z(174)*z(1432)-
1.221507832898173*z(175)*z(1439)-z(170)*z(1425)-z(176)*z(1432)-z(177)*z(1439)) +
0.000439106*z(470)*(1.522912463049924*z(887)+15.07109900570705*z(129)*z(883)+15.07109900570705*z(130)*z(887)+15.07109900570705*z(13
1)*z(891)-12.16291738213552*z(883)-2.050165563667998*z(891)-6.399042600192209*z(132)*z(883)-6.399042600192209*z(133)*z(887)-
6.399042600192209*z(134)*z(891)-z(135)*z(883)-z(136)*z(887)-z(137)*z(891)) +
0.003268181*z(522)*(2.228608513420769*z(718)+2.730118068736096*z(710)-1.290837318985699*z(714)-2.575607348552605*z(85)*z(718)-
2.575607348552605*z(86)*z(714)-2.575607348552605*z(87)*z(710)-1.999895354633051*z(88)*z(718)-1.999895354633051*z(89)*z(714)-
1.999895354633051*z(90)*z(710)-z(82)*z(718)-z(83)*z(714)-z(84)*z(710)) +
0.00334198*z(514)*(2.523796072986673*z(1709)+2.541448183412228*z(1695)-z(1702)-2.199101730112089*z(166)*z(1695)-
2.199101730112089*z(172)*z(1702)-2.199101730112089*z(173)*z(1709)-2.178312856450368*z(170)*z(1695)-2.178312856450368*z(176)*z(1702)-
2.178312856450368*z(177)*z(1709)-1.513417495017923*z(168)*z(1695)-1.513417495017923*z(174)*z(1702)-1.513417495017923*z(175)*z(1709))
+ 0.00353959*z(518)*(2.051998112775774*z(1107)+2.135415683737382*z(1099)-z(1103)-2.270871202596911*z(132)*z(1099)-
2.270871202596911*z(133)*z(1103)-2.270871202596911*z(134)*z(1107)-2.253045691732658*z(135)*z(1099)-2.253045691732658*z(136)*z(1103)-
2.253045691732658*z(137)*z(1107)-1.758446882266025*z(129)*z(1099)-1.758446882266025*z(130)*z(1103)-1.758446882266025*z(131)*z(1107))
+ 0.005250594000000001*z(510)*(1.3792740402324*z(2311)+1.576289654084852*z(2291)-1.220873295478569*z(2301)-
1.749405305380686*z(207)*z(2291)-1.749405305380686*z(208)*z(2301)-1.749405305380686*z(209)*z(2311)-1.677448684853561*z(213)*z(2291)-
1.677448684853561*z(214)*z(2301)-1.677448684853561*z(215)*z(2311)-z(210)*z(2291)-z(211)*z(2301)-z(212)*z(2311)) -
9.800000000000001*SW*(z(2790)*z(2912)+z(2791)*z(2827)+z(2792)*z(2925)) - 1.4*SW*(z(114)*z(556)+z(116)*z(552)+z(118)*z(548)) -
1.4*SW*(z(121)*z(775)+z(125)*z(779)+z(126)*z(783)) - 1.4*SW*(z(199)*z(1859)+z(203)*z(1869)+z(204)*z(1879)) -
1.4*SW*(z(284)*z(2500)+z(286)*z(2513)+z(288)*z(2526)) - 1.4*SW*(z(20)*z(1209)+z(194)*z(1216)-z(196)*z(1223)) -
1.4*SW*(z(2674)+z(284)*z(2701)+z(286)*z(2714)+z(288)*z(2727)) -
0.001989028*z(506)*(2.026969957185118*z(678)+3.454209794935014*z(674)+z(82)*z(682)+z(83)*z(678)+z(84)*z(674)+4.239214832571488*z(85)
*z(682)+4.239214832571488*z(86)*z(678)+4.239214832571488*z(87)*z(674)-3.484725705218831*z(682)-3.642764204425479*z(88)*z(682)-
3.642764204425479*z(89)*z(678)-3.642764204425479*z(90)*z(674)) - 4.9681E-
05*z(450)*(8.034661138060828*z(1331)+396.0548298142147*z(1317)+z(168)*z(1317)+z(174)*z(1324)+z(175)*z(1331)-
19.09315432459089*z(1324)-410.7868199110324*z(166)*z(1317)-410.7868199110324*z(172)*z(1324)-410.7868199110324*z(173)*z(1331)-
25.99182786175802*z(170)*z(1317)-25.99182786175802*z(176)*z(1324)-25.99182786175802*z(177)*z(1331)) -
0.00508017*z(494)*(z(2229)+1.575266182037215*z(2239)+1.26191682561804*z(210)*z(2219)+1.26191682561804*z(211)*z(2229)+1.261916825618
04*z(212)*z(2239)+1.738918973183969*z(207)*z(2219)+1.738918973183969*z(208)*z(2229)+1.738918973183969*z(209)*z(2239)-
1.564399813392072*z(2219)-1.445018572213135*z(213)*z(2219)-1.445018572213135*z(214)*z(2229)-1.445018572213135*z(215)*z(2239)) -
0.00404624*z(502)*(z(1031)+2.01888419866345*z(1035)+1.135663974455297*z(132)*z(1027)+1.135663974455297*z(133)*z(1031)+1.13566397445
5297*z(134)*z(1035)+1.613396387757523*z(129)*z(1027)+1.613396387757523*z(130)*z(1031)+1.613396387757523*z(131)*z(1035)-
1.85913242911938*z(1027)-1.514074795365574*z(135)*z(1027)-1.514074795365574*z(136)*z(1031)-1.514074795365574*z(137)*z(1035)) -
0.00297212*z(498)*(z(1594)+2.289867165524945*z(1601)+1.384676594484745*z(168)*z(1587)+1.384676594484745*z(174)*z(1594)+1.3846765944
84745*z(175)*z(1601)+2.471453709809833*z(166)*z(1587)+2.471453709809833*z(172)*z(1594)+2.471453709809833*z(173)*z(1601)-
2.986921120277781*z(1587)-2.589087923771584*z(170)*z(1587)-2.589087923771584*z(176)*z(1594)-2.589087923771584*z(177)*z(1601)) -
z(446)*(2.0E-
07*z(2023)+0.0210342*z(2003)+0.0009485670000000001*z(210)*z(2003)+0.0009485670000000001*z(211)*z(2013)+0.0009485670000000001*z(212
)*z(2023)-0.001343886*z(2013)-0.0196764*z(207)*z(2003)-0.0196764*z(208)*z(2013)-0.0196764*z(209)*z(2023)-0.00039917*z(213)*z(2003)-
0.00039917*z(214)*z(2013)-0.00039917*z(215)*z(2023)) -
0.00021415*z(490)*(40.8116273639972*z(624)+64.04720989960309*z(82)*z(628)+64.04720989960309*z(83)*z(624)+64.04720989960309*z(84)*z(
620)-70.09441979920617*z(628)-4.618921316833995*z(620)-10.35433107634836*z(85)*z(628)-10.35433107634836*z(86)*z(624)-
10.35433107634836*z(87)*z(620)-z(88)*z(628)-z(89)*z(624)-z(90)*z(620)) -
0.000156039*z(466)*(34.63294432802056*z(1371)+4.285595267849704*z(168)*z(1371)+4.285595267849704*z(174)*z(1378)+4.285595267849704*
z(175)*z(1385)-6.601112542377225*z(1378)-z(1385)-34.22740468728971*z(166)*z(1371)-34.22740468728971*z(172)*z(1378)-
34.22740468728971*z(173)*z(1385)-5.76932689904447*z(170)*z(1371)-5.76932689904447*z(176)*z(1378)-5.76932689904447*z(177)*z(1385));
z(3143) = KB3*z(379);
z(3142) = KB2*z(378);
z(3149) = z(378)*z(3143) - z(379)*z(3142);
z(3141) = KB1*z(377);
z(3148) = z(379)*z(3141) - z(377)*z(3143);
z(3147) = z(377)*z(3142) - z(378)*z(3141);
z(3198) = KC3*z(391);
z(3197) = KC2*z(387);
z(3204) = z(387)*z(3198) - z(391)*z(3197);
z(3196) = KC1*z(383);
z(3203) = z(391)*z(3196) - z(383)*z(3198);
z(3202) = z(383)*z(3197) - z(387)*z(3196);
z(3249) = KD3*z(412);
z(3248) = KD2*z(405);
z(3255) = z(405)*z(3249) - z(412)*z(3248);
z(3247) = KD1*z(398);
z(3254) = z(412)*z(3247) - z(398)*z(3249);
z(3253) = z(398)*z(3248) - z(405)*z(3247);
z(3320) = KE3*z(442);
z(3319) = KE2*z(432);
z(3326) = z(432)*z(3320) - z(442)*z(3319);
z(3318) = KE1*z(422);
z(3325) = z(442)*z(3318) - z(422)*z(3320);
z(3324) = z(422)*z(3319) - z(432)*z(3318);
z(3336) = z(252)*z(419) + z(255)*z(429) + z(258)*z(439);
z(3352) = z(281) + z(253)*z(422) + z(256)*z(432) + z(259)*z(442);
z(3365) = z(282) + z(254)*z(422) + z(257)*z(432) + z(260)*z(442);
```

z(3396) = KH3*z(3365);
z(3395) = KH2*z(3352);
z(3402) = z(3352)*z(3396) - z(3365)*z(3395);
z(3349) = z(253)*z(419) + z(256)*z(429) + z(259)*z(439);
z(3339) = z(280) + z(252)*z(422) + z(255)*z(432) + z(258)*z(442);
z(3394) = KH1*z(3339);
z(3401) = z(3365)*z(3394) - z(3339)*z(3396);
z(3362) = z(254)*z(419) + z(257)*z(429) + z(260)*z(439);
z(3400) = z(3339)*z(3395) - z(3352)*z(3394);
z(3412) = z(419)*z(2730) + z(429)*z(2733) + z(439)*z(2736);
z(3428) = z(334) + z(300)*z(280) + z(303)*z(281) + z(306)*z(282) + z(2731)*z(422) + z(2734)*z(432) + z(2737)*z(442);
z(3441) = z(335) + z(301)*z(280) + z(304)*z(281) + z(307)*z(282) + z(2732)*z(422) + z(2735)*z(432) + z(2738)*z(442);
z(3481) = KK3*z(3441);
z(3480) = KK2*z(3428);
z(3487) = z(3428)*z(3481) - z(3441)*z(3480);
z(3425) = z(419)*z(2731) + z(429)*z(2734) + z(439)*z(2737);
z(3415) = z(333) + z(299)*z(280) + z(302)*z(281) + z(305)*z(282) + z(2730)*z(422) + z(2733)*z(432) + z(2736)*z(442);
z(3479) = KK1*z(3415);
z(3486) = z(3441)*z(3479) - z(3415)*z(3481);
z(3438) = z(419)*z(2732) + z(429)*z(2735) + z(439)*z(2738);
z(3485) = z(3415)*z(3480) - z(3428)*z(3479);
z(3497) = z(419)*z(2868) + z(429)*z(2871) + z(439)*z(2874);
z(2869) = z(197)*z(2788) + z(199)*z(2791) - z(28)*z(344);
z(2872) = z(201)*z(2788) + z(203)*z(2791) + z(205)*z(344);
z(2875) = z(202)*z(2788) + z(204)*z(2791) + z(206)*z(344);
z(3513) = z(375) + z(347)*z(333) + z(350)*z(334) + z(353)*z(335) + z(2779)*z(280) + z(2782)*z(281) + z(2785)*z(282) + z(2869)*z(422) +
z(2872)*z(432) + z(2875)*z(442);
z(3526) = z(376) + z(348)*z(333) + z(351)*z(334) + z(354)*z(335) + z(2780)*z(280) + z(2783)*z(281) + z(2786)*z(282) + z(2870)*z(422) +
z(2873)*z(432) + z(2876)*z(442);
z(3563) = KS3*z(3526);
z(3562) = KS2*z(3513);
z(3569) = z(3513)*z(3563) - z(3526)*z(3562);
z(3510) = z(419)*z(2869) + z(429)*z(2872) + z(439)*z(2875);
z(3500) = z(374) + z(346)*z(333) + z(349)*z(334) + z(352)*z(335) + z(2778)*z(280) + z(2781)*z(281) + z(2784)*z(282) + z(2868)*z(422) +
z(2871)*z(432) + z(2874)*z(442);
z(3561) = KS1*z(3500);
z(3568) = z(3526)*z(3561) - z(3500)*z(3563);
z(3523) = z(419)*z(2870) + z(429)*z(2873) + z(439)*z(2876);
z(3567) = z(3500)*z(3562) - z(3513)*z(3561);
z(3100) = z(1)*z(3)*QNA_3pp - 2*z(1)*z(4)*QNA_2p*QNA_3p - z(2)*z(3)*QNA_2p^2 - z(2)*z(3)*QNA_3p^2 - z(2)*z(4)*QNA_2pp;
z(3101) = 2*z(2)*z(4)*QNA_2p*QNA_3p - z(1)*z(3)*QNA_2p^2 - z(1)*z(3)*QNA_3p^2 - z(1)*z(4)*QNA_2pp - z(2)*z(3)*QNA_3pp;
z(3102) = z(8)*z(49)*QNB_3p^2 + z(7)*z(3100) - z(7)*z(51)*QNB_3p^2 - 2*z(7)*QNB_3p*z(67) - 2*z(8)*QNB_3p*z(68) - z(7)*z(49)*QNB_3pp -
z(8)*z(51)*QNB_3pp - z(8)*z(3101);
z(3103) = 2*z(9)*z(11)*QNB_1p*QNB_2p + z(9)*z(12)*QNB_2pp + z(10)*z(11)*QNB_1pp - z(10)*z(12)*QNB_1p^2 - z(10)*z(12)*QNB_2p^2;
z(3104) = 2*z(7)*QNB_3p*z(68) + z(7)*z(51)*QNB_3pp + z(7)*z(3101) + z(8)*z(3100) - z(7)*z(49)*QNB_3p^2 - z(8)*z(51)*QNB_3p^2 -
2*z(8)*QNB_3p*z(67) - z(8)*z(49)*QNB_3pp;
z(3105) = z(9)*z(11)*QNB_1pp - 2*z(10)*z(11)*QNB_1p*QNB_2p - z(9)*z(12)*QNB_1p^2 - z(9)*z(12)*QNB_2p^2 - z(10)*z(12)*QNB_2pp;
z(3106) = z(4)*z(80)*QNA_2p^2 + 2*z(92)*z(93) + z(11)*z(3102) + z(72)*z(3103) + z(78)*z(3104) - z(11)*z(73)*QNB_1p^2 - 2*z(3)*QNA_2p*z(94)
- 2*z(12)*QNB_1p*z(91) - z(3)*z(80)*QNA_2pp - z(12)*z(73)*QNB_1pp - z(4)*z(3105);
z(3107) = 2*z(1)*z(3)*QNA_2p*QNA_3p + z(1)*z(4)*QNA_3pp + z(2)*z(3)*QNA_2pp - z(2)*z(4)*QNA_2p^2 - z(2)*z(4)*QNA_3p^2;
z(3108) = 2*z(2)*z(6)*QNA_1p*QNA_3p + 2*z(5)*QNA_1p*z(61) + z(5)*z(52)*QNA_1pp + z(6)*z(3107) - z(1)*z(5)*QNA_1p^2 -
z(1)*z(5)*QNA_3p^2 - z(6)*z(52)*QNA_1p^2 - z(1)*z(6)*QNA_1pp - z(2)*z(5)*QNA_3pp;
z(3109) = z(1)*z(3)*QNA_2pp - 2*z(2)*z(3)*QNA_2p*QNA_3p - z(1)*z(4)*QNA_2p^2 - z(1)*z(4)*QNA_3p^2 - z(2)*z(4)*QNA_3pp;
z(3110) = z(2)*z(5)*QNA_1p^2 + z(2)*z(5)*QNA_3p^2 + 2*z(1)*z(6)*QNA_1p*QNA_3p + 2*z(5)*QNA_1p*z(59) + z(2)*z(6)*QNA_1pp +
z(5)*z(50)*QNA_1pp + z(6)*z(3109) - z(6)*z(50)*QNA_1p^2 - z(1)*z(5)*QNA_3pp;
z(3111) = z(8)*z(53)*QNB_3p^2 + z(7)*z(3108) - z(7)*z(55)*QNB_3p^2 - 2*z(7)*QNB_3p*z(60) - 2*z(8)*QNB_3p*z(62) - z(7)*z(53)*QNB_3pp -
z(8)*z(55)*QNB_3pp - z(8)*z(3110);
z(3112) = 2*z(7)*QNB_3p*z(62) + z(7)*z(55)*QNB_3pp + z(7)*z(3110) + z(8)*z(3108) - z(7)*z(53)*QNB_3p^2 - z(8)*z(55)*QNB_3p^2 -
2*z(8)*QNB_3p*z(60) - z(8)*z(53)*QNB_3pp;
z(3113) = z(3)*z(5)*QNA_1pp - 2*z(4)*z(5)*QNA_1p*QNA_2p - z(3)*z(6)*QNA_1p^2 - z(3)*z(6)*QNA_2p^2 - z(4)*z(6)*QNA_2pp;
z(3114) = 2*z(63)*z(94) + 2*z(93)*z(97) + z(11)*z(3111) + z(57)*z(3105) + z(74)*z(3103) + z(78)*z(3112) + z(80)*z(3113) - z(11)*z(75)*QNB_1p^2
- 2*z(12)*QNB_1p*z(96) - z(12)*z(75)*QNB_1pp;
z(3115) = z(1)*z(6)*QNA_1p^2 + z(1)*z(6)*QNA_3p^2 + 2*z(2)*z(5)*QNA_1p*QNA_3p + z(2)*z(6)*QNA_3pp + z(5)*z(3107) -
z(5)*z(52)*QNA_1p^2 - 2*z(6)*QNA_1p*z(61) - z(1)*z(5)*QNA_1pp - z(6)*z(52)*QNA_1pp;
z(3116) = 2*z(1)*z(5)*QNA_1p*QNA_3p + z(1)*z(6)*QNA_3pp + z(2)*z(5)*QNA_1pp + z(5)*z(3109) - z(2)*z(6)*QNA_1p^2 -
z(2)*z(6)*QNA_3p^2 - z(5)*z(50)*QNA_1p^2 - 2*z(6)*QNA_1p*z(59) - z(6)*z(50)*QNA_1pp;
z(3117) = z(8)*z(54)*QNB_3p^2 + z(7)*z(3115) - z(7)*z(56)*QNB_3p^2 - 2*z(7)*QNB_3p*z(66) - 2*z(8)*QNB_3p*z(64) - z(7)*z(54)*QNB_3pp -
z(8)*z(56)*QNB_3pp - z(8)*z(3116);
z(3118) = 2*z(7)*QNB_3p*z(64) + z(7)*z(56)*QNB_3pp + z(7)*z(3116) + z(8)*z(3115) - z(7)*z(54)*QNB_3p^2 - z(8)*z(56)*QNB_3p^2 -
2*z(8)*QNB_3p*z(66) - z(8)*z(54)*QNB_3pp;
z(3119) = 2*z(4)*z(6)*QNA_1p*QNA_2p - z(3)*z(5)*QNA_1p^2 - z(3)*z(5)*QNA_2p^2 - z(3)*z(6)*QNA_1pp - z(4)*z(5)*QNA_2pp;
z(3120) = 2*z(65)*z(94) + 2*z(93)*z(100) + z(11)*z(3117) + z(58)*z(3105) + z(76)*z(3103) + z(78)*z(3118) + z(80)*z(3119) -
z(11)*z(77)*QNB_1p^2 - 2*z(12)*QNB_1p*z(99) - z(12)*z(77)*QNB_1pp;
z(3121) = z(95)*z(106) + z(98)*z(104) + z(101)*z(105) + z(84)*z(3106) + z(87)*z(3114) + z(90)*z(3120);
z(3122) = z(60)*z(66) + z(62)*z(64) + z(63)*z(65) + z(54)*z(3110) + z(56)*z(3108) + z(58)*z(3113);
z(3123) = z(64)*z(68) + z(66)*z(67) + z(49)*z(3116) + z(51)*z(3115) - z(3)*QNA_2p*z(65) - z(4)*z(3119);
z(3124) = z(4)*z(57)*QNA_2p^2 + z(60)*z(67) + z(62)*z(68) + z(53)*z(3101) + z(55)*z(3100) - z(3)*QNA_2p*z(63) - z(3)*z(57)*QNA_2pp;
z(3125) = z(69)*z(108) + z(70)*z(109) + z(71)*z(107) + z(3121) + z(82)*z(3122) + z(85)*z(3123) + z(88)*z(3124);

z(3126) = z(3125) + z(138)*U1 + z(141)*U2 - z(9)*QNB_2p*U3;
z(3144) = KB1*z(3126);
z(3127) = z(9)*z(11)*QNB_2pp - 2*z(9)*z(12)*QNB_1p*QNB_2p - z(10)*z(11)*QNB_1p^2 - z(10)*z(11)*QNB_2p^2 - z(10)*z(12)*QNB_1pp;
z(3128) = 2*z(10)*z(12)*QNB_1p*QNB_2p - z(9)*z(11)*QNB_1p^2 - z(9)*z(11)*QNB_2p^2 - z(9)*z(12)*QNB_1pp - z(10)*z(11)*QNB_2pp;
z(3129) = z(4)*z(81)*QNA_2p^2 + z(12)*z(73)*QNB_1p^2 + 2*z(92)*z(103) + z(72)*z(3127) + z(79)*z(3104) - 2*z(3)*QNA_2p*z(102) - 2*z(11)*QNB_1p*z(91) - z(3)*z(81)*QNA_2pp - z(11)*z(73)*QNB_1pp - z(4)*z(3128) - z(12)*z(3102);
z(3130) = z(12)*z(75)*QNB_1p^2 + 2*z(63)*z(102) + 2*z(97)*z(103) + z(57)*z(3128) + z(74)*z(3127) + z(79)*z(3112) + z(81)*z(3113) - 2*z(11)*QNB_1p*z(96) - z(11)*z(75)*QNB_1pp - z(12)*z(3111);
z(3131) = z(12)*z(77)*QNB_1p^2 + 2*z(65)*z(102) + 2*z(100)*z(103) + z(58)*z(3128) + z(76)*z(3127) + z(79)*z(3118) + z(81)*z(3119) - 2*z(11)*QNB_1p*z(99) - z(11)*z(77)*QNB_1pp - z(12)*z(3117);
z(3132) = z(104)*z(109) + z(105)*z(107) + z(106)*z(108) + z(82)*z(3129) + z(85)*z(3130) + z(88)*z(3131);
z(3133) = z(69)*z(95) + z(70)*z(98) + z(71)*z(101) + z(3132) + z(83)*z(3122) + z(86)*z(3123) + z(89)*z(3124);
z(3134) = z(3133) + z(94)*U3 + z(146)*U1 + z(147)*U2;
z(3145) = KB2*z(3134);
z(3135) = 2*z(3)*z(9)*QNA_2p*QNB_2p + z(3)*z(10)*QNA_2pp + z(4)*z(9)*QNB_2pp + z(9)*z(3104) - z(4)*z(10)*QNA_2p^2 - z(4)*z(10)*QNB_2p^2 - z(9)*z(72)*QNB_2p^2 - 2*z(10)*QNB_2p*z(92) - z(10)*z(72)*QNB_2pp;
z(3136) = z(10)*z(57)*QNB_2p^2 + z(9)*z(3112) - z(9)*z(74)*QNB_2p^2 - 2*z(9)*QNB_2p*z(63) - 2*z(10)*QNB_2p*z(97) - z(9)*z(57)*QNB_2pp - z(10)*z(74)*QNB_2pp - z(10)*z(3113);
z(3137) = z(10)*z(58)*QNB_2p^2 + z(9)*z(3118) - z(9)*z(76)*QNB_2p^2 - 2*z(9)*QNB_2p*z(65) - 2*z(10)*QNB_2p*z(100) - z(9)*z(58)*QNB_2pp - z(10)*z(76)*QNB_2pp - z(10)*z(3119);
z(3138) = z(95)*z(108) + z(98)*z(109) + z(101)*z(107) + z(83)*z(3135) + z(86)*z(3136) + z(89)*z(3137);
z(3139) = z(69)*z(106) + z(70)*z(104) + z(71)*z(105) + z(3138) + z(84)*z(3122) + z(87)*z(3123) + z(90)*z(3124);
z(3140) = z(3139) + z(102)*U3 + z(149)*U1 + z(150)*U2;
z(3146) = KB3*z(3140);
z(3153) = z(13)*z(15)*QNC_2pp - 2*z(14)*z(15)*QNC_2p*QNC_3p - z(13)*z(16)*QNC_2p^2 - z(13)*z(16)*QNC_3p^2 - z(14)*z(16)*QNC_3pp;
z(3154) = z(14)*z(17)*QNC_1p^2 + z(14)*z(17)*QNC_3p^2 + 2*z(13)*z(18)*QNC_1p*QNC_3p + 2*z(17)*QNC_1p*z(139) + z(14)*z(18)*QNC_1pp + z(17)*z(120)*QNC_1pp + z(18)*z(3153) - z(18)*z(120)*QNC_1p^2 - z(13)*z(17)*QNC_3pp;
z(3155) = 2*z(13)*z(15)*QNC_2p*QNC_3p + z(13)*z(16)*QNC_3pp + z(14)*z(15)*QNC_2pp - z(14)*z(16)*QNC_2p^2 - z(14)*z(16)*QNC_3p^2;
z(3156) = 2*z(14)*z(18)*QNC_1p*QNC_3p + 2*z(17)*QNC_1p*z(142) + z(17)*z(122)*QNC_1pp + z(18)*z(3155) - z(13)*z(17)*QNC_1p^2 - z(13)*z(17)*QNC_3p^2 - z(18)*z(122)*QNC_1p^2 - z(13)*z(18)*QNC_1pp - z(14)*z(17)*QNC_3pp;
z(3157) = 2*z(8)*z(10)*QNB_2p*QNB_3p - z(7)*z(9)*QNB_2p^2 - z(7)*z(9)*QNB_3p^2 - z(7)*z(10)*QNB_2pp - z(8)*z(9)*QNB_3pp;
z(3158) = z(7)*z(9)*QNB_3pp - 2*z(7)*z(10)*QNB_2p*QNB_3p - z(8)*z(9)*QNB_2p^2 - z(8)*z(9)*QNB_3p^2 - z(8)*z(10)*QNB_2pp;
z(3159) = z(15)*z(17)*QNC_1pp - 2*z(16)*z(17)*QNC_1p*QNC_2p - z(15)*z(18)*QNC_1p^2 - z(15)*z(18)*QNC_2p^2 - z(16)*z(18)*QNC_2pp;
z(3160) = z(10)*z(127)*QNB_2p^2 + 2*z(138)*z(140) + 2*z(141)*z(143) + z(113)*z(3154) + z(114)*z(3156) + z(123)*z(3157) + z(125)*z(3158) - 2*z(9)*QNB_2p*z(144) - z(9)*z(127)*QNB_2pp - z(10)*z(3159);
z(3161) = z(8)*z(11)*QNB_1p^2 + z(8)*z(11)*QNB_3p^2 + 2*z(7)*z(12)*QNB_1p*QNB_3p + z(8)*z(12)*QNB_1pp + z(7)*z(3103) - z(7)*z(78)*QNB_3p^2 - 2*z(8)*QNB_3p*z(93) - z(7)*z(11)*QNB_3pp - z(8)*z(78)*QNB_3pp;
z(3162) = 2*z(8)*z(12)*QNB_1p*QNB_3p + 2*z(7)*QNB_3p*z(93) + z(7)*z(78)*QNB_3pp + z(8)*z(3103) - z(7)*z(11)*QNB_1p^2 - z(7)*z(11)*QNB_3p^2 - z(8)*z(78)*QNB_3p^2 - z(7)*z(12)*QNB_1pp - z(8)*z(11)*QNB_3pp;
z(3163) = 2*z(94)*z(144) + 2*z(140)*z(146) + 2*z(143)*z(147) + z(80)*z(3159) + z(115)*z(3154) + z(116)*z(3156) + z(123)*z(3161) + z(125)*z(3162) + z(127)*z(3105);
z(3164) = 2*z(7)*z(11)*QNB_1p*QNB_3p + z(7)*z(12)*QNB_3pp + z(8)*z(11)*QNB_1pp + z(7)*z(3127) - z(7)*z(79)*QNB_3p^2 - z(8)*z(12)*QNB_1p^2 - z(8)*z(12)*QNB_3p^2 - 2*z(8)*QNB_3p*z(103) - z(8)*z(79)*QNB_3pp;
z(3165) = z(7)*z(12)*QNB_1p^2 + z(7)*z(12)*QNB_3p^2 + 2*z(8)*z(11)*QNB_1p*QNB_3p + 2*z(7)*QNB_3p*z(103) + z(7)*z(79)*QNB_3pp + z(8)*z(12)*QNB_3pp + z(8)*z(3127) - z(8)*z(79)*QNB_3p^2 - z(7)*z(11)*QNB_1pp;
z(3166) = 2*z(102)*z(144) + 2*z(140)*z(149) + 2*z(143)*z(150) + z(81)*z(3159) + z(117)*z(3154) + z(118)*z(3156) + z(123)*z(3164) + z(125)*z(3165) + z(127)*z(3128);
z(3167) = z(145)*z(157) + z(148)*z(155) + z(151)*z(156) + z(131)*z(3160) + z(134)*z(3163) + z(137)*z(3166);
z(3168) = z(160)*z(379) + z(161)*z(377) + z(162)*z(378) + z(3167) + z(129)*z(3125) + z(132)*z(3133) + z(135)*z(3139);
z(3169) = z(80)*z(162) + z(81)*z(160) + z(132)*z(94) + z(135)*z(102) - z(9)*z(129)*QNB_2p - z(10)*z(161);
z(3170) = z(113)*z(161) + z(115)*z(162) + z(117)*z(160) + z(129)*z(138) + z(132)*z(146) + z(135)*z(149);
z(3171) = z(114)*z(161) + z(116)*z(162) + z(118)*z(160) + z(129)*z(141) + z(132)*z(147) + z(135)*z(150);
z(3172) = z(3168) + z(158)*U4 + z(159)*U5 + z(3169)*U3 + z(3170)*U1 + z(3171)*U2 - z(15)*QNC_2p*U6;
z(3199) = KC1*z(3172);
z(3173) = 2*z(13)*z(17)*QNC_1p*QNC_3p + z(13)*z(18)*QNC_3pp + z(14)*z(17)*QNC_1pp + z(17)*z(3153) - z(14)*z(18)*QNC_1p^2 - z(14)*z(18)*QNC_3p^2 - z(17)*z(120)*QNC_1p^2 - 2*z(18)*QNC_1p*z(139) - z(18)*z(120)*QNC_1pp;
z(3174) = z(13)*z(18)*QNC_1p^2 + z(13)*z(18)*QNC_3p^2 + 2*z(14)*z(17)*QNC_1p*QNC_3p + z(14)*z(18)*QNC_3pp + z(17)*z(3155) - z(17)*z(122)*QNC_1p^2 - 2*z(18)*QNC_1p*z(142) - z(13)*z(17)*QNC_1pp - z(18)*z(122)*QNC_1pp;
z(3175) = 2*z(16)*z(18)*QNC_1p*QNC_2p - z(15)*z(17)*QNC_1p^2 - z(15)*z(17)*QNC_2p^2 - z(15)*z(18)*QNC_1pp - z(16)*z(17)*QNC_2pp;
z(3176) = z(10)*z(128)*QNB_2p^2 + 2*z(138)*z(153) + 2*z(141)*z(154) + z(113)*z(3173) + z(114)*z(3174) + z(124)*z(3157) + z(126)*z(3158) - 2*z(9)*QNB_2p*z(152) - z(9)*z(128)*QNB_2pp - z(10)*z(3175);
z(3177) = 2*z(94)*z(152) + 2*z(146)*z(153) + 2*z(147)*z(154) + z(80)*z(3175) + z(115)*z(3173) + z(116)*z(3174) + z(124)*z(3161) + z(126)*z(3162) + z(128)*z(3105);
z(3178) = 2*z(102)*z(152) + 2*z(149)*z(153) + 2*z(150)*z(154) + z(81)*z(3175) + z(117)*z(3173) + z(118)*z(3174) + z(124)*z(3164) + z(126)*z(3165) + z(128)*z(3128);
z(3179) = z(155)*z(162) + z(156)*z(160) + z(157)*z(161) + z(129)*z(3176) + z(132)*z(3177) + z(135)*z(3178);
z(3180) = z(145)*z(377) + z(148)*z(378) + z(151)*z(379) + z(3179) + z(130)*z(3125) + z(133)*z(3133) + z(136)*z(3139);
z(3181) = z(80)*z(148) + z(81)*z(151) + z(133)*z(94) + z(136)*z(102) - z(9)*z(130)*QNB_2p - z(10)*z(145);
z(3182) = z(113)*z(145) + z(115)*z(148) + z(117)*z(151) + z(130)*z(138) + z(133)*z(146) + z(136)*z(149);
z(3183) = z(114)*z(145) + z(116)*z(148) + z(118)*z(151) + z(130)*z(141) + z(133)*z(147) + z(136)*z(150);
z(3184) = z(3180) + z(140)*U4 + z(143)*U5 + z(144)*U6 + z(3181)*U3 + z(3182)*U1 + z(3183)*U2;
z(3200) = KC2*z(3184);
z(3185) = 2*z(14)*z(16)*QNC_2p*QNC_3p - z(13)*z(15)*QNC_2p^2 - z(13)*z(15)*QNC_3p^2 - z(13)*z(16)*QNC_2pp - z(14)*z(15)*QNC_3pp;
z(3186) = z(13)*z(15)*QNC_3pp - 2*z(13)*z(16)*QNC_2p*QNC_3p - z(14)*z(15)*QNC_2p^2 - z(14)*z(15)*QNC_3p^2 - z(14)*z(16)*QNC_2pp;
z(3187) = 2*z(9)*z(15)*QNB_2p*QNC_2p + 2*z(138)*z(158) + 2*z(141)*z(159) + z(9)*z(16)*QNB_2pp + z(10)*z(15)*QNC_2pp + z(113)*z(3185) + z(114)*z(3186) + z(119)*z(3157) + z(121)*z(3158) - z(10)*z(16)*QNB_2p^2 - z(10)*z(16)*QNC_2p^2;
z(3188) = z(16)*z(80)*QNC_2p^2 + 2*z(146)*z(158) + 2*z(147)*z(159) + z(115)*z(3185) + z(116)*z(3186) + z(119)*z(3161) + z(121)*z(3162) - 2*z(15)*QNC_2p*z(94) - z(15)*z(80)*QNC_2pp - z(16)*z(3105);

z(3189) = z(16)*z(81)*QNC_2p^2 + 2*z(149)*z(158) + 2*z(150)*z(159) + z(117)*z(3185) + z(118)*z(3186) + z(119)*z(3164) + z(121)*z(3165) - 2*z(15)*QNC_2p*z(102) - z(15)*z(81)*QNC_2pp - z(16)*z(3128);
z(3190) = z(145)*z(161) + z(148)*z(162) + z(151)*z(160) + z(130)*z(3187) + z(133)*z(3188) + z(136)*z(3189);
z(3191) = z(155)*z(378) + z(156)*z(379) + z(157)*z(377) + z(3190) + z(131)*z(3125) + z(134)*z(3133) + z(137)*z(3139);
z(3192) = z(80)*z(155) + z(81)*z(156) + z(134)*z(94) + z(137)*z(102) - z(9)*z(131)*QNB_2p - z(10)*z(157);
z(3193) = z(113)*z(157) + z(115)*z(155) + z(117)*z(156) + z(131)*z(138) + z(134)*z(146) + z(137)*z(149);
z(3194) = z(114)*z(157) + z(116)*z(155) + z(118)*z(156) + z(131)*z(141) + z(134)*z(147) + z(137)*z(150);
z(3195) = z(3191) + z(152)*U6 + z(153)*U4 + z(154)*U5 + z(3192)*U3 + z(3193)*U1 + z(3194)*U2;
z(3201) = KC3*z(3195);
z(3208) = z(20)*z(119)*QND_3p^2 + z(19)*z(3186) - z(19)*z(121)*QND_3p^2 - 2*z(19)*QND_3p*z(158) - 2*z(20)*QND_3p*z(159) - z(19)*z(119)*QND_3pp - z(20)*z(121)*QND_3pp - z(20)*z(3185);
z(3209) = z(16)*z(24)*QNC_2p^2 + z(16)*z(24)*QND_1p^2 + z(23)*z(3208) - 2*z(15)*z(23)*QNC_2p*QND_1p - z(23)*z(167)*QND_1p^2 - 2*z(24)*QND_1p*z(178) - z(15)*z(24)*QNC_2pp - z(16)*z(23)*QND_1pp - z(24)*z(167)*QND_1pp;
z(3210) = z(20)*z(123)*QND_3p^2 + z(19)*z(3156) - z(19)*z(125)*QND_3p^2 - 2*z(19)*QND_3p*z(140) - 2*z(20)*QND_3p*z(143) - z(19)*z(123)*QND_3pp - z(20)*z(125)*QND_3pp - z(20)*z(3154);
z(3211) = 2*z(23)*QND_1p*z(144) + z(23)*z(127)*QND_1pp + z(23)*z(3210) + z(24)*z(3159) - z(23)*z(169)*QND_1p^2 - z(24)*z(127)*QND_1p^2 - 2*z(24)*QND_1p*z(180) - z(24)*z(169)*QND_1pp;
z(3212) = z(20)*z(124)*QND_3p^2 + z(19)*z(3174) - z(19)*z(126)*QND_3p^2 - 2*z(19)*QND_3p*z(153) - 2*z(20)*QND_3p*z(154) - z(19)*z(124)*QND_3pp - z(20)*z(126)*QND_3pp - z(20)*z(3173);
z(3213) = 2*z(23)*QND_1p*z(152) + z(23)*z(128)*QND_1pp + z(23)*z(3212) + z(24)*z(3175) - z(23)*z(171)*QND_1p^2 - z(24)*z(128)*QND_1p^2 - 2*z(24)*QND_1p*z(182) - z(24)*z(171)*QND_1pp;
z(3214) = z(179)*z(186) + z(181)*z(184) + z(183)*z(185) + z(173)*z(3209) + z(175)*z(3211) + z(177)*z(3213);
z(3215) = z(187)*z(391) + z(188)*z(383) + z(189)*z(387) + z(3214) + z(166)*z(3168) + z(168)*z(3180) + z(170)*z(3191);
z(3216) = z(119)*z(188) + z(123)*z(189) + z(124)*z(187) + z(166)*z(158) + z(168)*z(140) + z(170)*z(153);
z(3217) = z(121)*z(188) + z(125)*z(189) + z(126)*z(187) + z(166)*z(159) + z(168)*z(143) + z(170)*z(154);
z(3218) = z(127)*z(189) + z(128)*z(187) + z(168)*z(144) + z(170)*z(152) - z(15)*z(166)*QNC_2p - z(16)*z(188);
z(3219) = z(166)*z(3169) + z(168)*z(3181) + z(170)*z(3192) + z(380)*z(188) + z(384)*z(189) + z(388)*z(187);
z(3220) = z(166)*z(3170) + z(168)*z(3182) + z(170)*z(3193) + z(381)*z(188) + z(385)*z(189) + z(389)*z(187);
z(3221) = z(166)*z(3171) + z(168)*z(3183) + z(170)*z(3194) + z(382)*z(188) + z(386)*z(189) + z(390)*z(187);
z(3222) = z(3215) + z(3216)*U4 + z(3217)*U5 + z(3218)*U6 + z(3219)*U3 + z(3220)*U1 + z(3221)*U2 + z(19)*QND_3p*U8 - z(20)*QND_3p*U7;
z(3250) = KD1*z(3222);
z(3223) = z(16)*z(23)*QNC_2p^2 + z(16)*z(23)*QND_1p^2 + z(24)*z(167)*QND_1p^2 + 2*z(15)*z(24)*QNC_2p*QND_1p + z(16)*z(24)*QND_1pp - 2*z(23)*QND_1p*z(178) - z(15)*z(23)*QNC_2pp - z(23)*z(167)*QND_1pp - z(24)*z(3208);
z(3224) = z(24)*z(169)*QND_1p^2 + z(23)*z(3159) - z(23)*z(127)*QND_1p^2 - 2*z(23)*QND_1p*z(180) - 2*z(24)*QND_1p*z(144) - z(23)*z(169)*QND_1pp - z(24)*z(127)*QND_1pp - z(24)*z(3210);
z(3225) = z(24)*z(171)*QND_1p^2 + z(23)*z(3175) - z(23)*z(128)*QND_1p^2 - 2*z(23)*QND_1p*z(182) - 2*z(24)*QND_1p*z(152) - z(23)*z(171)*QND_1pp - z(24)*z(128)*QND_1pp - z(24)*z(3212);
z(3226) = z(184)*z(189) + z(185)*z(187) + z(186)*z(188) + z(166)*z(3223) + z(168)*z(3224) + z(170)*z(3225);
z(3227) = z(179)*z(383) + z(181)*z(387) + z(183)*z(391) + z(3226) + z(172)*z(3168) + z(174)*z(3180) + z(176)*z(3191);
z(3228) = z(119)*z(179) + z(123)*z(181) + z(124)*z(183) + z(172)*z(158) + z(174)*z(140) + z(176)*z(153);
z(3229) = z(121)*z(179) + z(125)*z(181) + z(126)*z(183) + z(172)*z(159) + z(174)*z(143) + z(176)*z(154);
z(3230) = z(127)*z(181) + z(128)*z(183) + z(174)*z(144) + z(176)*z(152) - z(15)*z(172)*QNC_2p - z(16)*z(179);
z(3231) = z(172)*z(3169) + z(174)*z(3181) + z(176)*z(3192) + z(380)*z(179) + z(384)*z(181) + z(388)*z(183);
z(3232) = z(172)*z(3170) + z(174)*z(3182) + z(176)*z(3193) + z(381)*z(179) + z(385)*z(181) + z(389)*z(183);
z(3233) = z(172)*z(3171) + z(174)*z(3183) + z(176)*z(3194) + z(382)*z(179) + z(386)*z(181) + z(390)*z(183);
z(3234) = z(3227) + z(222)*U8 + z(3228)*U4 + z(3229)*U5 + z(3230)*U6 + z(3231)*U3 + z(3232)*U1 + z(3233)*U2 + z(23)*QND_1p*U9 - z(223)*U7;
z(3251) = KD2*z(3234);
z(3235) = 2*z(19)*QND_3p*z(159) + z(19)*z(121)*QND_3pp + z(19)*z(3185) + z(20)*z(3186) - z(19)*z(119)*QND_3p^2 - z(20)*z(121)*QND_3p^2 - 2*z(20)*QND_3p*z(158) - z(20)*z(119)*QND_3pp;
z(3236) = 2*z(19)*QND_3p*z(143) + z(19)*z(125)*QND_3pp + z(19)*z(3154) + z(20)*z(3156) - z(19)*z(123)*QND_3p^2 - z(20)*z(125)*QND_3p^2 - 2*z(20)*QND_3p*z(140) - z(20)*z(123)*QND_3pp;
z(3237) = 2*z(19)*QND_3p*z(154) + z(19)*z(126)*QND_3pp + z(19)*z(3173) + z(20)*z(3174) - z(19)*z(124)*QND_3p^2 - z(20)*z(126)*QND_3p^2 - 2*z(20)*QND_3p*z(153) - z(20)*z(124)*QND_3pp;
z(3238) = z(179)*z(188) + z(181)*z(189) + z(183)*z(187) + z(172)*z(3235) + z(174)*z(3236) + z(176)*z(3237);
z(3239) = z(184)*z(387) + z(185)*z(391) + z(186)*z(383) + z(3238) + z(173)*z(3168) + z(175)*z(3180) + z(177)*z(3191);
z(3240) = z(119)*z(186) + z(123)*z(184) + z(124)*z(185) + z(173)*z(158) + z(175)*z(140) + z(177)*z(153);
z(3241) = z(121)*z(186) + z(125)*z(184) + z(126)*z(185) + z(173)*z(159) + z(175)*z(143) + z(177)*z(154);
z(3242) = z(127)*z(184) + z(128)*z(185) + z(175)*z(144) + z(177)*z(152) - z(15)*z(173)*QNC_2p - z(16)*z(186);
z(3243) = z(173)*z(3169) + z(175)*z(3181) + z(177)*z(3192) + z(380)*z(186) + z(384)*z(184) + z(388)*z(185);
z(3244) = z(173)*z(3170) + z(175)*z(3182) + z(177)*z(3193) + z(381)*z(186) + z(385)*z(184) + z(389)*z(185);
z(3245) = z(173)*z(3171) + z(175)*z(3183) + z(177)*z(3194) + z(382)*z(186) + z(386)*z(184) + z(390)*z(185);
z(3246) = z(3239) + z(225)*U7 + z(3240)*U4 + z(3241)*U5 + z(3242)*U6 + z(3243)*U3 + z(3244)*U1 + z(3245)*U2 - z(226)*U8 - z(24)*QND_1p*U9;
z(3252) = KD3*z(3246);
z(3259) = z(25)*z(27)*QNE_2pp - 2*z(26)*z(27)*QNE_2p*QNE_3p - z(25)*z(28)*QNE_2p^2 - z(25)*z(28)*QNE_3p^2 - z(26)*z(28)*QNE_3pp;
z(3260) = z(26)*z(29)*QNE_1p^2 + z(26)*z(29)*QNE_3p^2 + 2*z(25)*z(30)*QNE_1p*QNE_3p + 2*z(29)*QNE_1p*z(216) + z(26)*z(30)*QNE_1pp + z(29)*z(198)*QNE_1pp + z(30)*z(3259) - z(30)*z(198)*QNE_1p^2 - z(25)*z(29)*QNE_3pp;
z(3261) = 2*z(25)*z(27)*QNE_2p*QNE_3p + z(25)*z(28)*QNE_3pp + z(26)*z(27)*QNE_2pp - z(26)*z(28)*QNE_2p^2 - z(26)*z(28)*QNE_3p^2;
z(3262) = 2*z(26)*z(30)*QNE_1p*QNE_3p + 2*z(29)*QNE_1p*z(218) + z(29)*z(200)*QNE_1pp + z(30)*z(3261) - z(25)*z(29)*QNE_1p^2 - z(25)*z(29)*QNE_3p^2 - z(30)*z(200)*QNE_1p^2 - z(25)*z(30)*QNE_1pp - z(26)*z(29)*QNE_3pp;
z(3263) = 2*z(19)*QND_3p*z(219) + z(19)*z(203)*QND_3pp + z(19)*z(3260) + z(20)*z(3262) - z(19)*z(201)*QND_3p^2 - z(20)*z(203)*QND_3p^2 - 2*z(20)*QND_3p*z(217) - z(20)*z(201)*QND_3pp;
z(3264) = z(27)*z(29)*QNE_1pp - 2*z(28)*z(29)*QNE_1p*QNE_2p - z(27)*z(30)*QNE_1p^2 - z(27)*z(30)*QNE_2p^2 - z(28)*z(30)*QNE_2pp;
z(3265) = 2*z(20)*z(24)*QND_1p*QND_3p - z(19)*z(23)*QND_1p^2 - z(19)*z(23)*QND_3p^2 - z(19)*z(24)*QND_1pp - z(20)*z(23)*QND_3pp;
z(3266) = z(19)*z(23)*QND_3pp - 2*z(19)*z(24)*QND_1p*QND_3p - z(20)*z(23)*QND_1p^2 - z(20)*z(23)*QND_3p^2 - z(20)*z(24)*QND_1pp;
z(3267) = 2*z(219)*z(222) + 2*z(23)*QND_1p*z(221) + z(23)*z(205)*QND_1pp + z(24)*z(3264) + z(194)*z(3262) + z(203)*z(3265) - z(24)*z(205)*QND_1p^2 - 2*z(217)*z(223) - z(193)*z(3260) - z(201)*z(3266);

z(3268) = 2*z(19)*z(23)*QND_1p*QND_3p + z(19)*z(24)*QND_3pp + z(20)*z(23)*QND_1pp - z(20)*z(24)*QND_1p^2 - z(20)*z(24)*QND_3p^2;
z(3269) = z(19)*z(23)*QND_1pp - 2*z(20)*z(23)*QND_1p*QND_3p - z(19)*z(24)*QND_1p^2 - z(19)*z(24)*QND_3p^2 - z(20)*z(24)*QND_3pp;
z(3270) = 2*z(217)*z(225) + z(23)*z(3264) + z(195)*z(3260) + z(201)*z(3268) - z(23)*z(205)*QND_1p^2 - 2*z(219)*z(226) -
2*z(24)*QND_1p*z(221) - z(24)*z(205)*QND_1pp - z(196)*z(3262) - z(203)*z(3269);
z(3271) = z(220)*z(233) + z(224)*z(231) + z(227)*z(232) + z(209)*z(3263) + z(212)*z(3267) + z(215)*z(3270);
z(3272) = z(236)*z(412) + z(237)*z(398) + z(238)*z(405) + z(3271) + z(207)*z(3215) + z(210)*z(3227) + z(213)*z(3239);
z(3273) = z(19)*z(237) + z(195)*z(236) + z(213)*z(225) - z(20)*z(207)*QND_3p - z(193)*z(238) - z(210)*z(223);
z(3274) = z(19)*z(207)*QND_3p + z(20)*z(237) + z(194)*z(238) + z(210)*z(222) - z(196)*z(236) - z(213)*z(226);
z(3275) = z(207)*z(3216) + z(210)*z(3228) + z(213)*z(3240) + z(392)*z(237) + z(399)*z(238) + z(406)*z(236);
z(3276) = z(207)*z(3217) + z(210)*z(3229) + z(213)*z(3241) + z(393)*z(237) + z(400)*z(238) + z(407)*z(236);
z(3277) = z(207)*z(3218) + z(210)*z(3230) + z(213)*z(3242) + z(394)*z(237) + z(401)*z(238) + z(408)*z(236);
z(3278) = z(207)*z(3219) + z(210)*z(3231) + z(213)*z(3243) + z(395)*z(237) + z(402)*z(238) + z(409)*z(236);
z(3279) = z(207)*z(3220) + z(210)*z(3232) + z(213)*z(3244) + z(396)*z(237) + z(403)*z(238) + z(410)*z(236);
z(3280) = z(207)*z(3221) + z(210)*z(3233) + z(213)*z(3245) + z(397)*z(237) + z(404)*z(238) + z(411)*z(236);
z(3281) = z(23)*z(210)*QND_1p + z(23)*z(236) + z(24)*z(238) - z(24)*z(213)*QND_1p;
z(3282) = z(3272) + z(234)*U10 + z(235)*U11 + z(3273)*U7 + z(3274)*U8 + z(3275)*U4 + z(3276)*U5 + z(3277)*U6 + z(3278)*U3 + z(3279)*U1 +
z(3280)*U2 + z(3281)*U9 - z(27)*QNE_2p*U12;
z(3321) = KE1*z(3282);
z(3283) = 2*z(25)*z(29)*QNE_1p*QNE_3p + z(25)*z(30)*QNE_3pp + z(26)*z(29)*QNE_1pp + z(29)*z(3259) - z(26)*z(30)*QNE_1p^2 -
z(26)*z(30)*QNE_3p^2 - z(29)*z(198)*QNE_1p^2 - 2*z(30)*QNE_1p*z(216) - z(30)*z(198)*QNE_1pp;
z(3284) = z(25)*z(30)*QNE_1p^2 + z(25)*z(30)*QNE_3p^2 + 2*z(26)*z(29)*QNE_1p*QNE_3p + z(26)*z(30)*QNE_3pp + z(29)*z(3261) -
z(29)*z(200)*QNE_1p^2 - 2*z(30)*QNE_1p*z(218) - z(25)*z(29)*QNE_1pp - z(30)*z(200)*QNE_1pp;
z(3285) = 2*z(19)*QND_3p*z(229) + z(19)*z(204)*QND_3pp + z(19)*z(3283) + z(20)*z(3284) - z(19)*z(202)*QND_3p^2 - z(20)*z(204)*QND_3p^2
- 2*z(20)*QND_3p*z(230) - z(20)*z(202)*QND_3pp;
z(3286) = 2*z(28)*z(30)*QNE_1p*QNE_2p - z(27)*z(29)*QNE_1p^2 - z(27)*z(29)*QNE_2p^2 - z(27)*z(30)*QNE_1pp - z(28)*z(29)*QNE_2pp;
z(3287) = 2*z(222)*z(229) + 2*z(23)*QND_1p*z(228) + z(23)*z(206)*QND_1pp + z(24)*z(3286) + z(194)*z(3284) + z(204)*z(3265) -
z(24)*z(206)*QND_1p^2 - 2*z(223)*z(230) - z(193)*z(3283) - z(202)*z(3266);
z(3288) = 2*z(225)*z(230) + z(23)*z(3286) + z(195)*z(3283) + z(202)*z(3268) - z(23)*z(206)*QND_1p^2 - 2*z(226)*z(229) -
2*z(24)*QND_1p*z(228) - z(24)*z(206)*QND_1pp - z(196)*z(3284) - z(204)*z(3269);
z(3289) = z(231)*z(238) + z(232)*z(236) + z(233)*z(237) + z(207)*z(3285) + z(210)*z(3287) + z(213)*z(3288);
z(3290) = z(220)*z(398) + z(224)*z(405) + z(227)*z(412) + z(3289) + z(208)*z(3215) + z(211)*z(3227) + z(214)*z(3239);
z(3291) = z(19)*z(220) + z(195)*z(227) + z(214)*z(225) - z(20)*z(208)*QND_3p - z(193)*z(224) - z(211)*z(223);
z(3292) = z(19)*z(208)*QND_3p + z(20)*z(220) + z(194)*z(224) + z(211)*z(222) - z(196)*z(227) - z(214)*z(226);
z(3293) = z(208)*z(3216) + z(211)*z(3228) + z(214)*z(3240) + z(392)*z(220) + z(399)*z(224) + z(406)*z(227);
z(3294) = z(208)*z(3217) + z(211)*z(3229) + z(214)*z(3241) + z(393)*z(220) + z(400)*z(224) + z(407)*z(227);
z(3295) = z(208)*z(3218) + z(211)*z(3230) + z(214)*z(3242) + z(394)*z(220) + z(401)*z(224) + z(408)*z(227);
z(3296) = z(208)*z(3219) + z(211)*z(3231) + z(214)*z(3243) + z(395)*z(220) + z(402)*z(224) + z(409)*z(227);
z(3297) = z(208)*z(3220) + z(211)*z(3232) + z(214)*z(3244) + z(396)*z(220) + z(403)*z(224) + z(410)*z(227);
z(3298) = z(208)*z(3221) + z(211)*z(3233) + z(214)*z(3245) + z(397)*z(220) + z(404)*z(224) + z(411)*z(227);
z(3299) = z(23)*z(211)*QND_1p + z(23)*z(227) + z(24)*z(224) - z(24)*z(214)*QND_1p;
z(3300) = z(3290) + z(217)*U10 + z(219)*U11 + z(221)*U12 + z(3291)*U7 + z(3292)*U8 + z(3293)*U4 + z(3294)*U5 + z(3295)*U6 + z(3296)*U3 +
z(3297)*U1 + z(3298)*U2 + z(3299)*U9;
z(3322) = KE2*z(3300);
z(3301) = 2*z(26)*z(28)*QNE_2p*QNE_3p - z(25)*z(27)*QNE_2p^2 - z(25)*z(27)*QNE_3p^2 - z(25)*z(28)*QNE_2pp - z(26)*z(27)*QNE_3pp;
z(3302) = z(25)*z(27)*QNE_3pp - 2*z(25)*z(28)*QNE_2p*QNE_3p - z(26)*z(27)*QNE_2p^2 - z(26)*z(27)*QNE_3p^2 - z(26)*z(28)*QNE_2pp;
z(3303) = 2*z(19)*QND_3p*z(235) + z(19)*z(199)*QND_3pp + z(19)*z(3301) + z(20)*z(3302) - z(19)*z(197)*QND_3p^2 - z(20)*z(199)*QND_3p^2
- 2*z(20)*QND_3p*z(234) - z(20)*z(197)*QND_3pp;
z(3304) = z(24)*z(28)*QND_1p^2 + z(24)*z(28)*QNE_2p^2 + 2*z(222)*z(235) + z(194)*z(3302) + z(199)*z(3265) -
2*z(23)*z(27)*QND_1p*QNE_2p - 2*z(223)*z(234) - z(23)*z(28)*QND_1pp - z(24)*z(27)*QNE_2pp - z(193)*z(3301) - z(197)*z(3266);
z(3305) = z(23)*z(28)*QND_1p^2 + z(23)*z(28)*QNE_2p^2 + 2*z(24)*z(27)*QND_1p*QNE_2p + 2*z(225)*z(234) + z(24)*z(28)*QND_1pp +
z(195)*z(3301) + z(197)*z(3268) - 2*z(226)*z(235) - z(23)*z(27)*QNE_2pp - z(196)*z(3302) - z(199)*z(3269);
z(3306) = z(220)*z(237) + z(224)*z(238) + z(227)*z(236) + z(208)*z(3303) + z(211)*z(3304) + z(214)*z(3305);
z(3307) = z(231)*z(405) + z(232)*z(412) + z(233)*z(398) + z(3306) + z(209)*z(3215) + z(212)*z(3227) + z(215)*z(3239);
z(3308) = z(19)*z(233) + z(195)*z(232) + z(215)*z(225) - z(20)*z(209)*QND_3p - z(193)*z(231) - z(212)*z(223);
z(3309) = z(19)*z(209)*QND_3p + z(20)*z(233) + z(194)*z(231) + z(212)*z(222) - z(196)*z(232) - z(215)*z(226);
z(3310) = z(209)*z(3216) + z(212)*z(3228) + z(215)*z(3240) + z(392)*z(233) + z(399)*z(231) + z(406)*z(232);
z(3311) = z(209)*z(3217) + z(212)*z(3229) + z(215)*z(3241) + z(393)*z(233) + z(400)*z(231) + z(407)*z(232);
z(3312) = z(209)*z(3218) + z(212)*z(3230) + z(215)*z(3242) + z(394)*z(233) + z(401)*z(231) + z(408)*z(232);
z(3313) = z(209)*z(3219) + z(212)*z(3231) + z(215)*z(3243) + z(395)*z(233) + z(402)*z(231) + z(409)*z(232);
z(3314) = z(209)*z(3220) + z(212)*z(3232) + z(215)*z(3244) + z(396)*z(233) + z(403)*z(231) + z(410)*z(232);
z(3315) = z(209)*z(3221) + z(212)*z(3233) + z(215)*z(3245) + z(397)*z(233) + z(404)*z(231) + z(411)*z(232);
z(3316) = z(23)*z(212)*QND_1p + z(23)*z(232) + z(24)*z(231) - z(24)*z(215)*QND_1p;
z(3317) = z(3307) + z(228)*U12 + z(229)*U11 + z(230)*U10 + z(3308)*U7 + z(3309)*U8 + z(3310)*U4 + z(3311)*U5 + z(3312)*U6 + z(3313)*U3 +
z(3314)*U1 + z(3315)*U2 + z(3316)*U9;
z(3323) = KE3*z(3317);
z(3366) = z(32)*z(197)*QNH_3p^2 + z(31)*z(3302) - z(31)*z(199)*QNH_3p^2 - 2*z(31)*QNH_3p*z(234) - 2*z(32)*QNH_3p*z(235) -
z(31)*z(197)*QNH_3pp - z(32)*z(199)*QNH_3pp - z(32)*z(3301);
z(3367) = 2*z(33)*z(35)*QNH_1p*QNH_2p + z(33)*z(36)*QNH_2pp + z(34)*z(35)*QNH_1pp - z(34)*z(36)*QNH_1p^2 - z(34)*z(36)*QNH_2p^2;
z(3368) = 2*z(31)*QNH_3p*z(235) + z(31)*z(199)*QNH_3pp + z(31)*z(3301) + z(32)*z(3302) - z(31)*z(197)*QNH_3p^2 - z(32)*z(199)*QNH_3p^2
- 2*z(32)*QNH_3p*z(234) - z(32)*z(197)*QNH_3pp;
z(3369) = z(33)*z(35)*QNH_1pp - 2*z(34)*z(35)*QNH_1p*QNH_2p - z(33)*z(36)*QNH_1p^2 - z(33)*z(36)*QNH_2p^2 - z(34)*z(36)*QNH_2pp;
z(3370) = z(28)*z(250)*QNE_2p^2 + 2*z(262)*z(263) + z(35)*z(3366) + z(242)*z(3367) + z(248)*z(3368) - z(35)*z(243)*QNH_1p^2 -
2*z(27)*QNE_2p*z(264) - 2*z(36)*QNH_1p*z(261) - z(27)*z(250)*QNE_2pp - z(36)*z(243)*QNH_1pp - z(28)*z(3369);
z(3371) = z(32)*z(201)*QNH_3p^2 + z(31)*z(3262) - z(31)*z(203)*QNH_3p^2 - 2*z(31)*QNH_3p*z(217) - 2*z(32)*QNH_3p*z(219) -
z(31)*z(201)*QNH_3pp - z(32)*z(203)*QNH_3pp - z(32)*z(3260);
z(3372) = 2*z(31)*QNH_3p*z(219) + z(31)*z(203)*QNH_3pp + z(31)*z(3260) + z(32)*z(3262) - z(31)*z(201)*QNH_3p^2 - z(32)*z(203)*QNH_3p^2
- 2*z(32)*QNH_3p*z(217) - z(32)*z(201)*QNH_3pp;

z(3373) = 2*z(221)*z(264) + 2*z(263)*z(267) + z(35)*z(3371) + z(205)*z(3369) + z(244)*z(3367) + z(248)*z(3372) + z(250)*z(3264) - z(35)*z(245)*QNH_1p^2 - 2*z(36)*QNH_1p*z(266) - z(36)*z(245)*QNH_1pp;
z(3374) = z(32)*z(202)*QNH_3p^2 + z(31)*z(3284) - z(31)*z(204)*QNH_3p^2 - 2*z(31)*QNH_3p*z(230) - 2*z(32)*QNH_3p*z(229) - z(31)*z(202)*QNH_3pp - z(32)*z(204)*QNH_3pp - z(32)*z(3283);
z(3375) = 2*z(31)*QNH_3p*z(229) + z(31)*z(204)*QNH_3pp + z(31)*z(3283) + z(32)*z(3284) - z(31)*z(202)*QNH_3p^2 - z(32)*z(204)*QNH_3p^2 - 2*z(32)*QNH_3p*z(230) - z(32)*z(202)*QNH_3pp;
z(3376) = 2*z(228)*z(264) + 2*z(263)*z(270) + z(35)*z(3374) + z(206)*z(3369) + z(246)*z(3367) + z(248)*z(3375) + z(250)*z(3286) - z(35)*z(247)*QNH_1p^2 - 2*z(36)*QNH_1p*z(269) - z(36)*z(247)*QNH_1pp;
z(3377) = z(265)*z(276) + z(268)*z(274) + z(271)*z(275) + z(254)*z(3370) + z(257)*z(3373) + z(260)*z(3376);
z(3257) = z(432) + z(201)*U10 + z(203)*U11 + z(205)*U12 + z(423)*U7 + z(424)*U8 + z(425)*U4 + z(426)*U5 + z(427)*U6 + z(428)*U3 + z(429)*U1 + z(430)*U2 + z(431)*U9;
z(3258) = z(442) + z(202)*U10 + z(204)*U11 + z(206)*U12 + z(433)*U7 + z(434)*U8 + z(435)*U4 + z(436)*U5 + z(437)*U6 + z(438)*U3 + z(439)*U1 + z(440)*U2 + z(441)*U9;
z(3388) = z(3282) + (z(258)*z(280)+z(259)*z(281)+z(260)*z(282))*z(3257) - (z(255)*z(280)+z(256)*z(281)+z(257)*z(282))*z(3258);
z(3256) = z(422) + z(197)*U10 + z(199)*U11 + z(413)*U7 + z(414)*U8 + z(415)*U4 + z(416)*U5 + z(417)*U6 + z(418)*U3 + z(419)*U1 + z(420)*U2 + z(421)*U9 - z(28)*U12;
z(3389) = z(3300) + (z(252)*z(280)+z(253)*z(281)+z(254)*z(282))*z(3258) - (z(258)*z(280)+z(259)*z(281)+z(260)*z(282))*z(3256);
z(3390) = z(3317) + (z(255)*z(280)+z(256)*z(281)+z(257)*z(282))*z(3256) - (z(252)*z(280)+z(253)*z(281)+z(254)*z(282))*z(3257);
z(3391) = z(3377) + z(252)*z(3388) + z(255)*z(3389) + z(258)*z(3390);
z(3397) = KH1*z(3391);
z(3378) = z(33)*z(35)*QNH_2pp - 2*z(33)*z(36)*QNH_1p*QNH_2p - z(34)*z(35)*QNH_1p^2 - z(34)*z(35)*QNH_2p^2 - z(34)*z(36)*QNH_1pp;
z(3379) = 2*z(34)*z(36)*QNH_1p*QNH_2p - z(33)*z(35)*QNH_1p^2 - z(33)*z(35)*QNH_2p^2 - z(33)*z(36)*QNH_1pp - z(34)*z(35)*QNH_2pp;
z(3380) = z(28)*z(251)*QNE_2p^2 + z(36)*z(243)*QNH_1p^2 + 2*z(262)*z(273) + z(242)*z(3378) + z(249)*z(3368) - 2*z(27)*QNE_2p*z(272) - 2*z(35)*QNH_1p*z(261) - z(27)*z(251)*QNE_2pp - z(35)*z(243)*QNH_1pp - z(28)*z(3379) - z(36)*z(3366);
z(3381) = z(36)*z(245)*QNH_1p^2 + 2*z(221)*z(272) + 2*z(267)*z(273) + z(205)*z(3379) + z(244)*z(3378) + z(249)*z(3372) + z(251)*z(3264) - 2*z(35)*QNH_1p*z(266) - z(35)*z(245)*QNH_1pp - z(36)*z(3371);
z(3382) = z(36)*z(247)*QNH_1p^2 + 2*z(228)*z(272) + 2*z(270)*z(273) + z(206)*z(3379) + z(246)*z(3378) + z(249)*z(3375) + z(251)*z(3286) - 2*z(35)*QNH_1p*z(269) - z(35)*z(247)*QNH_1pp - z(36)*z(3374);
z(3383) = z(274)*z(279) + z(275)*z(277) + z(276)*z(278) + z(252)*z(3380) + z(255)*z(3381) + z(258)*z(3382);
z(3392) = z(3383) + z(253)*z(3388) + z(256)*z(3389) + z(259)*z(3390);
z(3398) = KH2*z(3392);
z(3384) = 2*z(27)*z(33)*QNE_2p*QNH_2p + z(27)*z(34)*QNE_2pp + z(28)*z(33)*QNH_2pp + z(33)*z(3368) - z(28)*z(34)*QNE_2p^2 - z(28)*z(34)*QNH_2p^2 - z(33)*z(242)*QNH_2p^2 - 2*z(34)*QNH_2p*z(262) - z(34)*z(242)*QNH_2pp;
z(3385) = z(34)*z(205)*QNH_2p^2 + z(33)*z(3372) - z(33)*z(244)*QNH_2p^2 - 2*z(33)*QNH_2p*z(221) - 2*z(34)*QNH_2p*z(267) - z(33)*z(205)*QNH_2pp - z(34)*z(244)*QNH_2pp - z(34)*z(3264);
z(3386) = z(34)*z(206)*QNH_2p^2 + z(33)*z(3375) - z(33)*z(246)*QNH_2p^2 - 2*z(33)*QNH_2p*z(228) - 2*z(34)*QNH_2p*z(270) - z(33)*z(206)*QNH_2pp - z(34)*z(246)*QNH_2pp - z(34)*z(3286);
z(3387) = z(265)*z(278) + z(268)*z(279) + z(271)*z(277) + z(253)*z(3384) + z(256)*z(3385) + z(259)*z(3386);
z(3393) = z(3387) + z(254)*z(3388) + z(257)*z(3389) + z(260)*z(3390);
z(3399) = KH3*z(3393);
z(3442) = z(37)*z(39)*QNK_2pp - 2*z(38)*z(39)*QNK_2p*QNK_3p - z(37)*z(40)*QNK_2p^2 - z(37)*z(40)*QNK_3p^2 - z(38)*z(40)*QNK_3pp;
z(3443) = z(38)*z(41)*QNK_1p^2 + z(38)*z(41)*QNK_3p^2 + 2*z(37)*z(42)*QNK_1p*QNK_3p + 2*z(41)*QNK_1p*z(309) + z(38)*z(42)*QNK_1pp + z(41)*z(290)*QNK_1pp + z(42)*z(3442) - z(42)*z(290)*QNK_1p^2 - z(37)*z(41)*QNK_3pp;
z(3444) = 2*z(37)*z(39)*QNK_2p*QNK_3p + z(37)*z(40)*QNK_3pp + z(38)*z(39)*QNK_2pp - z(38)*z(40)*QNK_2p^2 - z(38)*z(40)*QNK_3p^2;
z(3445) = 2*z(38)*z(42)*QNK_1p*QNK_3p + 2*z(41)*QNK_1p*z(312) + z(41)*z(292)*QNK_1pp + z(42)*z(3444) - z(37)*z(41)*QNK_1p^2 - z(37)*z(41)*QNK_3p^2 - z(42)*z(292)*QNK_1p^2 - z(37)*z(42)*QNK_1pp - z(38)*z(41)*QNK_3pp;
z(3446) = 2*z(32)*z(34)*QNH_2p*QNH_3p - z(31)*z(33)*QNH_2p^2 - z(31)*z(33)*QNH_3p^2 - z(31)*z(34)*QNH_2pp - z(32)*z(33)*QNH_3pp;
z(3447) = z(31)*z(33)*QNH_3pp - 2*z(31)*z(34)*QNH_2p*QNH_3p - z(32)*z(33)*QNH_2p^2 - z(32)*z(33)*QNH_3p^2 - z(32)*z(34)*QNH_2pp;
z(3448) = z(39)*z(41)*QNK_1pp - 2*z(40)*z(41)*QNK_1p*QNK_2p - z(39)*z(42)*QNK_1p^2 - z(39)*z(42)*QNK_2p^2 - z(40)*z(42)*QNK_2pp;
z(3449) = z(34)*z(297)*QNH_2p^2 + 2*z(308)*z(310) + 2*z(311)*z(313) + z(283)*z(3443) + z(284)*z(3445) + z(293)*z(3446) + z(295)*z(3447) - 2*z(33)*QNH_2p*z(314) - z(33)*z(297)*QNH_2pp - z(34)*z(3448);
z(3450) = z(32)*z(35)*QNH_1p^2 + z(32)*z(35)*QNH_3p^2 + 2*z(31)*z(36)*QNH_1p*QNH_3p + z(32)*z(36)*QNH_1pp + z(31)*z(3367) - z(31)*z(248)*QNH_3p^2 - 2*z(32)*QNH_3p*z(263) - z(31)*z(35)*QNH_3pp - z(32)*z(248)*QNH_3pp;
z(3451) = 2*z(32)*z(36)*QNH_1p*QNH_3p + 2*z(31)*QNH_3p*z(263) + z(31)*z(248)*QNH_3pp + z(32)*z(3367) - z(31)*z(35)*QNH_1p^2 - z(31)*z(35)*QNH_3p^2 - z(32)*z(248)*QNH_3p^2 - z(31)*z(36)*QNH_1pp - z(32)*z(35)*QNH_3pp;
z(3452) = 2*z(264)*z(314) + 2*z(310)*z(316) + 2*z(313)*z(317) + z(250)*z(3448) + z(285)*z(3443) + z(286)*z(3445) + z(293)*z(3450) + z(295)*z(3451) + z(297)*z(3369);
z(3453) = 2*z(31)*z(35)*QNH_1p*QNH_3p + z(31)*z(36)*QNH_3pp + z(32)*z(35)*QNH_1pp + z(31)*z(3378) - z(31)*z(249)*QNH_3p^2 - z(32)*z(36)*QNH_1p^2 - z(32)*z(36)*QNH_3p^2 - 2*z(32)*QNH_3p*z(273) - z(32)*z(249)*QNH_3pp;
z(3454) = z(31)*z(36)*QNH_1p^2 + z(31)*z(36)*QNH_3p^2 + 2*z(32)*z(35)*QNH_1p*QNH_3p + 2*z(31)*QNH_3p*z(273) + z(31)*z(249)*QNH_3pp + z(32)*z(36)*QNH_3pp + z(32)*z(3378) - z(32)*z(249)*QNH_3p^2 - z(31)*z(35)*QNH_1pp;
z(3455) = 2*z(272)*z(314) + 2*z(310)*z(319) + 2*z(313)*z(320) + z(251)*z(3448) + z(287)*z(3443) + z(288)*z(3445) + z(293)*z(3453) + z(295)*z(3454) + z(297)*z(3379);
z(3456) = z(315)*z(327) + z(318)*z(325) + z(321)*z(326) + z(301)*z(3449) + z(304)*z(3452) + z(307)*z(3455);
z(3473) = z(281)*(z(305)*z(333)+z(306)*z(334)+z(307)*z(335)) + z(3377) - z(282)*(z(302)*z(333)+z(303)*z(334)+z(304)*z(335));
z(3474) = z(282)*(z(299)*z(333)+z(300)*z(334)+z(301)*z(335)) + z(3383) - z(280)*(z(305)*z(333)+z(306)*z(334)+z(307)*z(335));
z(3475) = z(280)*(z(302)*z(333)+z(303)*z(334)+z(304)*z(335)) + z(3387) - z(281)*(z(299)*z(333)+z(300)*z(334)+z(301)*z(335));
z(3470) = z(3282) + (z(258)*z(280)+z(259)*z(281)+z(260)*z(282)+z(2736)*z(333)+z(2737)*z(334)+z(2738)*z(335))*z(3257) - (z(255)*z(280)+z(256)*z(281)+z(257)*z(282)+z(2733)*z(333)+z(2734)*z(334)+z(2735)*z(335))*z(3258);
z(3471) = z(3300) + (z(252)*z(280)+z(253)*z(281)+z(254)*z(282)+z(2730)*z(333)+z(2731)*z(334)+z(2732)*z(335))*z(3258) - (z(258)*z(280)+z(259)*z(281)+z(260)*z(282)+z(2736)*z(333)+z(2737)*z(334)+z(2738)*z(335))*z(3256);
z(3472) = z(3317) + (z(255)*z(280)+z(256)*z(281)+z(257)*z(282)+z(2733)*z(333)+z(2734)*z(334)+z(2735)*z(335))*z(3256) - (z(252)*z(280)+z(253)*z(281)+z(254)*z(282)+z(2730)*z(333)+z(2731)*z(334)+z(2732)*z(335))*z(3257);
z(3476) = z(3456) + z(299)*z(3473) + z(302)*z(3474) + z(305)*z(3475) + z(2730)*z(3470) + z(2733)*z(3471) + z(2736)*z(3472);
z(3482) = KK1*z(3476);
z(3457) = 2*z(37)*z(41)*QNK_1p*QNK_3p + z(37)*z(42)*QNK_3pp + z(38)*z(41)*QNK_1pp + z(41)*z(3442) - z(38)*z(42)*QNK_1p^2 - z(38)*z(42)*QNK_3p^2 - z(41)*z(290)*QNK_1p^2 - 2*z(42)*QNK_1p*z(309) - z(42)*z(290)*QNK_1pp;

z(3458) = z(37)*z(42)*QNK_1p^2 + z(37)*z(42)*QNK_3p^2 + 2*z(38)*z(41)*QNK_1p*QNK_3p + z(38)*z(42)*QNK_3pp + z(41)*z(3444) - z(41)*z(292)*QNK_1p^2 - 2*z(42)*QNK_1p*z(312) - z(37)*z(41)*QNK_1pp - z(42)*z(292)*QNK_1pp;
z(3459) = 2*z(40)*z(42)*QNK_1p*QNK_2p - z(39)*z(41)*QNK_1p^2 - z(39)*z(41)*QNK_2p^2 - z(39)*z(42)*QNK_1pp - z(40)*z(41)*QNK_2pp;
z(3460) = z(34)*z(298)*QNH_2p^2 + 2*z(308)*z(323) + 2*z(311)*z(324) + z(283)*z(3457) + z(284)*z(3458) + z(294)*z(3446) + z(296)*z(3447) - 2*z(33)*QNH_2p*z(322) - z(33)*z(298)*QNH_2pp - z(34)*z(3459);
z(3461) = 2*z(264)*z(322) + 2*z(316)*z(323) + 2*z(317)*z(324) + z(250)*z(3459) + z(285)*z(3457) + z(286)*z(3458) + z(294)*z(3450) + z(296)*z(3451) + z(298)*z(3369);
z(3462) = 2*z(272)*z(322) + 2*z(319)*z(323) + 2*z(320)*z(324) + z(251)*z(3459) + z(287)*z(3457) + z(288)*z(3458) + z(294)*z(3453) + z(296)*z(3454) + z(298)*z(3379);
z(3463) = z(325)*z(332) + z(326)*z(330) + z(327)*z(331) + z(299)*z(3460) + z(302)*z(3461) + z(305)*z(3462);
z(3477) = z(3463) + z(300)*z(3473) + z(303)*z(3474) + z(306)*z(3475) + z(2731)*z(3470) + z(2734)*z(3471) + z(2737)*z(3472);
z(3483) = KK2*z(3477);
z(3464) = 2*z(38)*z(40)*QNK_2p*QNK_3p - z(37)*z(39)*QNK_2p^2 - z(37)*z(39)*QNK_3p^2 - z(37)*z(40)*QNK_2pp - z(38)*z(39)*QNK_3pp;
z(3465) = z(37)*z(39)*QNK_3pp - 2*z(37)*z(40)*QNK_2p*QNK_3p - z(38)*z(39)*QNK_2p^2 - z(38)*z(39)*QNK_3p^2 - z(38)*z(40)*QNK_2pp;
z(3466) = 2*z(33)*z(39)*QNH_2p*QNK_2p + 2*z(308)*z(328) + 2*z(311)*z(329) + z(33)*z(40)*QNH_2pp + z(34)*z(39)*QNK_2pp + z(283)*z(3464) + z(284)*z(3465) + z(289)*z(3446) + z(291)*z(3447) - z(34)*z(40)*QNH_2p^2 - z(34)*z(40)*QNK_2p^2;
z(3467) = z(40)*z(250)*QNK_2p^2 + 2*z(316)*z(328) + 2*z(317)*z(329) + z(285)*z(3464) + z(286)*z(3465) + z(289)*z(3450) + z(291)*z(3451) - 2*z(39)*QNK_2p*z(264) - z(39)*z(250)*QNK_2pp - z(40)*z(3369);
z(3468) = z(40)*z(251)*QNK_2p^2 + 2*z(319)*z(328) + 2*z(320)*z(329) + z(287)*z(3464) + z(288)*z(3465) + z(289)*z(3453) + z(291)*z(3454) - 2*z(39)*QNK_2p*z(272) - z(39)*z(251)*QNK_2pp - z(40)*z(3379);
z(3469) = z(315)*z(331) + z(318)*z(332) + z(321)*z(330) + z(300)*z(3466) + z(303)*z(3467) + z(306)*z(3468);
z(3478) = z(3469) + z(301)*z(3473) + z(304)*z(3474) + z(307)*z(3475) + z(2732)*z(3470) + z(2735)*z(3471) + z(2738)*z(3472);
z(3484) = KK3*z(3478);
z(3527) = z(44)*z(289)*QNS_3p^2 + z(43)*z(3465) - z(43)*z(291)*QNS_3p^2 - 2*z(43)*QNS_3p*z(328) - 2*z(44)*QNS_3p*z(329) - z(43)*z(289)*QNS_3pp - z(44)*z(291)*QNS_3pp - z(44)*z(3464);
z(3528) = 2*z(45)*z(47)*QNS_1p*QNS_2p + z(45)*z(48)*QNS_2pp + z(46)*z(47)*QNS_1pp - z(46)*z(48)*QNS_1p^2 - z(46)*z(48)*QNS_2p^2;
z(3529) = 2*z(43)*QNS_3p*z(329) + z(43)*z(291)*QNS_3pp + z(43)*z(3464) + z(44)*z(3465) - z(43)*z(289)*QNS_3p^2 - z(44)*z(291)*QNS_3p^2 - 2*z(44)*QNS_3p*z(328) - z(44)*z(289)*QNS_3pp;
z(3530) = z(45)*z(47)*QNS_1pp - 2*z(46)*z(47)*QNS_1p*QNS_2p - z(45)*z(48)*QNS_1p^2 - z(45)*z(48)*QNS_2p^2 - z(46)*z(48)*QNS_2pp;
z(3531) = z(40)*z(344)*QNK_2p^2 + 2*z(356)*z(357) + z(47)*z(3527) + z(336)*z(3528) + z(342)*z(3529) - z(47)*z(337)*QNS_1p^2 - 2*z(39)*QNK_2p*z(358) - 2*z(48)*QNS_1p*z(355) - z(39)*z(344)*QNK_2pp - z(48)*z(337)*QNS_1pp - z(40)*z(3530);
z(3532) = z(44)*z(293)*QNS_3p^2 + z(43)*z(3445) - z(43)*z(295)*QNS_3p^2 - 2*z(43)*QNS_3p*z(310) - 2*z(44)*QNS_3p*z(313) - z(43)*z(293)*QNS_3pp - z(44)*z(295)*QNS_3pp - z(44)*z(3443);
z(3533) = 2*z(43)*QNS_3p*z(313) + z(43)*z(295)*QNS_3pp + z(43)*z(3443) + z(44)*z(3445) - z(43)*z(293)*QNS_3p^2 - z(44)*z(295)*QNS_3p^2 - 2*z(44)*QNS_3p*z(310) - z(44)*z(293)*QNS_3pp;
z(3534) = 2*z(314)*z(358) + 2*z(357)*z(361) + z(47)*z(3532) + z(297)*z(3530) + z(338)*z(3528) + z(342)*z(3533) + z(344)*z(3448) - z(47)*z(339)*QNS_1p^2 - 2*z(48)*QNS_1p*z(360) - z(48)*z(339)*QNS_1pp;
z(3535) = z(44)*z(294)*QNS_3p^2 + z(43)*z(3458) - z(43)*z(296)*QNS_3p^2 - 2*z(43)*QNS_3p*z(323) - 2*z(44)*QNS_3p*z(324) - z(43)*z(294)*QNS_3pp - z(44)*z(296)*QNS_3pp - z(44)*z(3457);
z(3536) = 2*z(43)*QNS_3p*z(324) + z(43)*z(296)*QNS_3pp + z(43)*z(3457) + z(44)*z(3458) - z(43)*z(294)*QNS_3p^2 - z(44)*z(296)*QNS_3p^2 - 2*z(44)*QNS_3p*z(323) - z(44)*z(294)*QNS_3pp;
z(3537) = 2*z(322)*z(358) + 2*z(357)*z(364) + z(47)*z(3535) + z(298)*z(3530) + z(340)*z(3528) + z(342)*z(3536) + z(344)*z(3459) - z(47)*z(341)*QNS_1p^2 - 2*z(48)*QNS_1p*z(363) - z(48)*z(341)*QNS_1pp;
z(3538) = z(359)*z(370) + z(362)*z(368) + z(365)*z(369) + z(348)*z(3531) + z(351)*z(3534) + z(354)*z(3537);
z(3555) = z(334)*(z(352)*z(374)+z(353)*z(375)+z(354)*z(376)) + z(3456) - z(335)*(z(349)*z(374)+z(350)*z(375)+z(351)*z(376));
z(3556) = z(335)*(z(346)*z(374)+z(347)*z(375)+z(348)*z(376)) + z(3463) - z(333)*(z(352)*z(374)+z(353)*z(375)+z(354)*z(376));
z(3557) = z(333)*(z(349)*z(374)+z(350)*z(375)+z(351)*z(376)) + z(3469) - z(334)*(z(346)*z(374)+z(347)*z(375)+z(348)*z(376));
z(3552) = z(281)*(z(305)*z(333)+z(306)*z(334)+z(307)*z(335)+z(2784)*z(374)+z(2785)*z(375)+z(2786)*z(376)) + z(3377) - z(282)*(z(302)*z(333)+z(303)*z(334)+z(304)*z(335)+z(2781)*z(374)+z(2782)*z(375)+z(2783)*z(376));
z(3553) = z(282)*(z(299)*z(333)+z(300)*z(334)+z(301)*z(335)+z(2778)*z(374)+z(2779)*z(375)+z(2780)*z(376)) + z(3383) - z(280)*(z(305)*z(333)+z(306)*z(334)+z(307)*z(335)+z(2784)*z(374)+z(2785)*z(375)+z(2786)*z(376));
z(3554) = z(280)*(z(302)*z(333)+z(303)*z(334)+z(304)*z(335)+z(2781)*z(374)+z(2782)*z(375)+z(2783)*z(376)) + z(3387) - z(281)*(z(299)*z(333)+z(300)*z(334)+z(301)*z(335)+z(2778)*z(374)+z(2779)*z(375)+z(2780)*z(376));
z(3549) = z(3282) +
(z(258)*z(280)+z(259)*z(281)+z(260)*z(282)+z(2736)*z(333)+z(2737)*z(334)+z(2738)*z(335)+z(2874)*z(374)+z(2875)*z(375)+z(2876)*z(376))*z(3257) -
(z(255)*z(280)+z(256)*z(281)+z(257)*z(282)+z(2733)*z(333)+z(2734)*z(334)+z(2735)*z(335)+z(2871)*z(374)+z(2872)*z(375)+z(2873)*z(376))*z(3258);
z(3550) = z(3300) +
(z(252)*z(280)+z(253)*z(281)+z(254)*z(282)+z(2730)*z(333)+z(2731)*z(334)+z(2732)*z(335)+z(2868)*z(374)+z(2869)*z(375)+z(2870)*z(376))*z(3258) -
(z(258)*z(280)+z(259)*z(281)+z(260)*z(282)+z(2736)*z(333)+z(2737)*z(334)+z(2738)*z(335)+z(2874)*z(374)+z(2875)*z(375)+z(2876)*z(376))*z(3256);
z(3551) = z(3317) +
(z(255)*z(280)+z(256)*z(281)+z(257)*z(282)+z(2733)*z(333)+z(2734)*z(334)+z(2735)*z(335)+z(2871)*z(374)+z(2872)*z(375)+z(2873)*z(376))*z(3256) -
(z(252)*z(280)+z(253)*z(281)+z(254)*z(282)+z(2730)*z(333)+z(2731)*z(334)+z(2732)*z(335)+z(2868)*z(374)+z(2869)*z(375)+z(2870)*z(376))*z(3257);
z(3558) = z(3538) + z(346)*z(3555) + z(349)*z(3556) + z(352)*z(3557) + z(2778)*z(3552) + z(2781)*z(3553) + z(2784)*z(3554) + z(2868)*z(3549) + z(2871)*z(3550) + z(2874)*z(3551);
z(3564) = KS1*z(3558);
z(3539) = z(45)*z(47)*QNS_2pp - 2*z(45)*z(48)*QNS_1p*QNS_2p - z(46)*z(47)*QNS_1p^2 - z(46)*z(47)*QNS_2p^2 - z(46)*z(48)*QNS_1pp;
z(3540) = 2*z(46)*z(48)*QNS_1p*QNS_2p - z(45)*z(47)*QNS_1p^2 - z(45)*z(47)*QNS_2p^2 - z(45)*z(48)*QNS_1pp - z(46)*z(47)*QNS_2pp;
z(3541) = z(40)*z(345)*QNK_2p^2 + z(48)*z(337)*QNS_1p^2 + 2*z(356)*z(367) + z(336)*z(3539) + z(343)*z(3529) - 2*z(39)*QNK_2p*z(366) - 2*z(47)*QNS_1p*z(355) - z(39)*z(345)*QNK_2pp - z(47)*z(337)*QNS_1pp - z(40)*z(3540) - z(48)*z(3527);
z(3542) = z(48)*z(339)*QNS_1p^2 + 2*z(314)*z(366) + 2*z(361)*z(367) + z(297)*z(3540) + z(338)*z(3539) + z(343)*z(3533) + z(345)*z(3448) - 2*z(47)*QNS_1p*z(360) - z(47)*z(339)*QNS_1pp - z(48)*z(3532);

z(3543) = z(48)*z(341)*QNS_1p^2 + 2*z(322)*z(366) + 2*z(364)*z(367) + z(298)*z(3540) + z(340)*z(3539) + z(343)*z(3536) + z(345)*z(3459) - 2*z(47)*QNS_1p*z(363) - z(47)*z(341)*QNS_1pp - z(48)*z(3535);
z(3544) = z(368)*z(373) + z(369)*z(371) + z(370)*z(372) + z(346)*z(3541) + z(349)*z(3542) + z(352)*z(3543);
z(3559) = z(3544) + z(347)*z(3555) + z(350)*z(3556) + z(353)*z(3557) + z(2779)*z(3552) + z(2782)*z(3553) + z(2785)*z(3554) + z(2869)*z(3549) + z(2872)*z(3550) + z(2875)*z(3551);
z(3565) = KS2*z(3559);
z(3545) = 2*z(39)*z(45)*QNK_2p*QNS_2p + z(39)*z(46)*QNK_2pp + z(40)*z(45)*QNS_2pp + z(45)*z(3529) - z(40)*z(46)*QNK_2p^2 - z(40)*z(46)*QNS_2p^2 - z(45)*z(336)*QNS_2p^2 - 2*z(46)*QNS_2p*z(356) - z(46)*z(336)*QNS_2pp;
z(3546) = z(46)*z(297)*QNS_2p^2 + z(45)*z(3533) - z(45)*z(338)*QNS_2p^2 - 2*z(45)*QNS_2p*z(314) - 2*z(46)*QNS_2p*z(361) - z(45)*z(297)*QNS_2pp - z(46)*z(338)*QNS_2pp - z(46)*z(3448);
z(3547) = z(46)*z(298)*QNS_2p^2 + z(45)*z(3536) - z(45)*z(340)*QNS_2p^2 - 2*z(45)*QNS_2p*z(322) - 2*z(46)*QNS_2p*z(364) - z(45)*z(298)*QNS_2pp - z(46)*z(340)*QNS_2pp - z(46)*z(3459);
z(3548) = z(359)*z(372) + z(362)*z(373) + z(365)*z(371) + z(347)*z(3545) + z(350)*z(3546) + z(353)*z(3547);
z(3560) = z(3548) + z(348)*z(3555) + z(351)*z(3556) + z(354)*z(3557) + z(2780)*z(3552) + z(2783)*z(3553) + z(2786)*z(3554) + z(2870)*z(3549) + z(2873)*z(3550) + z(2876)*z(3551);
z(3566) = KS3*z(3560);
z(2777) = 0.0005*z(334) + 0.0005*z(300)*z(280) + 0.0005*z(303)*z(281) + 0.0005*z(306)*z(282) + 0.0005*z(2731)*z(422) + 0.0005*z(2734)*z(432) + 0.0005*z(2737)*z(442) - 0.00893*z(335) - 0.00893*z(301)*z(280) - 0.00893*z(304)*z(281) - 0.00893*z(307)*z(282) - 0.00893*z(2732)*z(422) - 0.00893*z(2735)*z(432) - 0.00893*z(2738)*z(442);
z(2764) = -0.01653*z(335) - 0.0005*z(333) - 0.01653*z(301)*z(280) - 0.01653*z(304)*z(281) - 0.01653*z(307)*z(282) - 0.01653*z(2732)*z(422) - 0.01653*z(2735)*z(432) - 0.01653*z(2738)*z(442) - 0.0005*z(299)*z(280) - 0.0005*z(302)*z(281) - 0.0005*z(305)*z(282) - 0.0005*z(2730)*z(422) - 0.0005*z(2733)*z(432) - 0.0005*z(2736)*z(442);
z(2751) = 0.00893*z(333) + 0.01653*z(334) + 0.00893*z(299)*z(280) + 0.00893*z(302)*z(281) + 0.00893*z(305)*z(282) + 0.00893*z(2730)*z(422) + 0.00893*z(2733)*z(432) + 0.00893*z(2736)*z(442) + 0.01653*z(300)*z(280) + 0.01653*z(303)*z(281) + 0.01653*z(306)*z(282) + 0.01653*z(2731)*z(422) + 0.01653*z(2734)*z(432) + 0.01653*z(2737)*z(442);
z(4206) = z(197)*z(310) + z(199)*z(313) + z(293)*z(234) + z(295)*z(235) - z(27)*z(297)*QNE_2p - z(28)*z(314);
z(4207) = z(201)*z(310) + z(203)*z(313) + z(205)*z(314) + z(293)*z(217) + z(295)*z(219) + z(297)*z(221);
z(4208) = z(202)*z(310) + z(204)*z(313) + z(206)*z(314) + z(293)*z(230) + z(295)*z(229) + z(297)*z(228);
z(4209) = z(197)*z(323) + z(199)*z(324) + z(294)*z(234) + z(296)*z(235) - z(27)*z(298)*QNE_2p - z(28)*z(322);
z(4210) = z(201)*z(323) + z(203)*z(324) + z(205)*z(322) + z(294)*z(217) + z(296)*z(219) + z(298)*z(221);
z(4211) = z(202)*z(323) + z(204)*z(324) + z(206)*z(322) + z(294)*z(230) + z(296)*z(229) + z(298)*z(228);
z(4212) = 0.0005*z(280)*z(315) + 0.0005*z(281)*z(318) + 0.0005*z(282)*z(321) + 0.0005*z(422)*z(4206) + 0.0005*z(432)*z(4207) + 0.0005*z(442)*z(4208) + 0.0005*z(3463) + 0.0005*z(300)*z(3377) + 0.0005*z(303)*z(3383) + 0.0005*z(306)*z(3387) + 0.0005*z(2731)*z(3272) + 0.0005*z(2734)*z(3290) + 0.0005*z(2737)*z(3307) - 0.00893*z(280)*z(327) - 0.00893*z(281)*z(325) - 0.00893*z(282)*z(326) - 0.00893*z(422)*z(4209) - 0.00893*z(432)*z(4210) - 0.00893*z(442)*z(4211) - 0.00893*z(3469) - 0.00893*z(301)*z(3377) - 0.00893*z(304)*z(3383) - 0.00893*z(307)*z(3387) - 0.00893*z(2732)*z(3272) - 0.00893*z(2735)*z(3290) - 0.00893*z(2738)*z(3307);
z(4213) = z(27)*z(40)*QNE_2p + z(28)*z(39)*QNK_2p + z(197)*z(328) + z(199)*z(329) + z(289)*z(234) + z(291)*z(235);
z(4214) = z(201)*z(328) + z(203)*z(329) + z(289)*z(217) + z(291)*z(219) - z(39)*z(205)*QNK_2p - z(40)*z(221);
z(4215) = z(202)*z(328) + z(204)*z(329) + z(289)*z(230) + z(291)*z(229) - z(39)*z(206)*QNK_2p - z(40)*z(228);
z(4216) = -0.01653*z(280)*z(327) - 0.01653*z(281)*z(325) - 0.01653*z(282)*z(326) - 0.01653*z(422)*z(4209) - 0.01653*z(432)*z(4210) - 0.01653*z(442)*z(4211) - 0.0005*z(280)*z(331) - 0.0005*z(281)*z(332) - 0.0005*z(282)*z(330) - 0.0005*z(422)*z(4213) - 0.0005*z(432)*z(4214) - 0.0005*z(442)*z(4215) - 0.01653*z(3469) - 0.0005*z(3456) - 0.01653*z(301)*z(3377) - 0.01653*z(304)*z(3383) - 0.01653*z(307)*z(3387) - 0.01653*z(2732)*z(3272) - 0.01653*z(2735)*z(3290) - 0.01653*z(2738)*z(3307) - 0.0005*z(299)*z(3377) - 0.0005*z(302)*z(3383) - 0.0005*z(305)*z(3387) - 0.0005*z(2730)*z(3272) - 0.0005*z(2733)*z(3290) - 0.0005*z(2736)*z(3307);
z(4217) = 0.00893*z(280)*z(331) + 0.00893*z(281)*z(332) + 0.00893*z(282)*z(330) + 0.00893*z(422)*z(4213) + 0.00893*z(432)*z(4214) + 0.00893*z(442)*z(4215) + 0.01653*z(280)*z(315) + 0.01653*z(281)*z(318) + 0.01653*z(282)*z(321) + 0.01653*z(422)*z(4206) + 0.01653*z(432)*z(4207) + 0.01653*z(442)*z(4208) + 0.00893*z(3456) + 0.01653*z(3463) + 0.00893*z(299)*z(3377) + 0.00893*z(302)*z(3383) + 0.00893*z(305)*z(3387) + 0.00893*z(2730)*z(3272) + 0.00893*z(2733)*z(3290) + 0.00893*z(2736)*z(3307) + 0.01653*z(300)*z(3377) + 0.01653*z(303)*z(3383) + 0.01653*z(306)*z(3387) + 0.01653*z(2731)*z(3272) + 0.01653*z(2734)*z(3290) + 0.01653*z(2737)*z(3307);
z(4177) = (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(199)*z(422) + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(203)*z(432) + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(204)*z(442) + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(284)*z(280) + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(286)*z(281) + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(288)*z(282) + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(291)*z(333) + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(295)*z(334) + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(296)*z(335) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(197)*z(422) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(201)*z(432) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(202)*z(442) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(283)*z(280) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(285)*z(281) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(287)*z(282) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(289)*z(333) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(293)*z(334) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(294)*z(335) + z(197)*z(3272)*z(2611) + z(201)*z(3290)*z(2611) + z(202)*z(3307)*z(2611) + z(283)*z(3377)*z(2611) + z(285)*z(3383)*z(2611) + z(287)*z(3387)*z(2611) + z(289)*z(3456)*z(2611) + z(293)*z(3463)*z(2611) + z(294)*z(3469)*z(2611) + z(217)*z(432)*z(2611) + z(230)*z(442)*z(2611) + z(234)*z(422)*z(2611) + z(280)*z(308)*z(2611) + z(281)*z(316)*z(2611) + z(282)*z(319)*z(2611) + z(310)*z(334)*z(2611) + z(323)*z(335)*z(2611) + z(328)*z(333)*z(2611) - z(199)*z(3272)*z(2610) - z(203)*z(3290)*z(2610) - z(204)*z(3307)*z(2610) - z(284)*z(3377)*z(2610) - z(286)*z(3383)*z(2610) - z(288)*z(3387)*z(2610) - z(291)*z(3456)*z(2610) - z(295)*z(3463)*z(2610) - z(296)*z(3469)*z(2610) - z(219)*z(432)*z(2610) - z(229)*z(442)*z(2610) - z(235)*z(422)*z(2610) - z(280)*z(311)*z(2610) - z(281)*z(317)*z(2610) - z(282)*z(320)*z(2610) - z(313)*z(334)*z(2610) - z(324)*z(335)*z(2610) - z(329)*z(333)*z(2610);
z(528) = z(70)*AOAB3 - z(71)*AOAB2;
z(527) = z(71)*AOAB1 - z(69)*AOAB3;
z(526) = z(69)*AOAB2 - z(70)*AOAB1;
z(525) = z(4)*z(69)*AON2 + z(51)*z(69)*AON3 + z(55)*z(70)*AON3 + z(56)*z(71)*AON3 - z(57)*z(70)*AON2 - z(58)*z(71)*AON2;
z(524) = z(57)*z(70)*AON1 + z(58)*z(71)*AON1 - z(4)*z(69)*AON1 - z(49)*z(69)*AON3 - z(53)*z(70)*AON3 - z(54)*z(71)*AON3;
z(523) = z(49)*z(69)*AON2 + z(53)*z(70)*AON2 + z(54)*z(71)*AON2 - z(51)*z(69)*AON1 - z(55)*z(70)*AON1 - z(56)*z(71)*AON1;
z(544) = z(82)*z(528) + z(85)*z(527) + z(88)*z(526) + z(113)*z(525) + z(114)*z(524) - z(10)*z(523);
z(576) = z(378)*z(564) - z(379)*z(563);
z(577) = z(544) + z(576);
z(545) = z(80)*z(523) + z(83)*z(528) + z(86)*z(527) + z(89)*z(526) + z(115)*z(525) + z(116)*z(524);

z(572) = z(379)*z(562) - z(377)*z(564);
z(578) = z(545) + z(572);
z(546) = z(81)*z(523) + z(84)*z(528) + z(87)*z(527) + z(90)*z(526) + z(117)*z(525) + z(118)*z(524);
z(568) = z(377)*z(563) - z(378)*z(562);
z(579) = z(546) + z(568);
z(744) = z(129)*z(577) + z(132)*z(578) + z(135)*z(579);
z(809) = z(387)*z(788) - z(391)*z(787);
z(813) = z(744) + z(809);
z(748) = z(130)*z(577) + z(133)*z(578) + z(136)*z(579);
z(802) = z(391)*z(786) - z(383)*z(788);
z(817) = z(748) + z(802);
z(752) = z(131)*z(577) + z(134)*z(578) + z(137)*z(579);
z(795) = z(383)*z(787) - z(387)*z(786);
z(821) = z(752) + z(795);
z(1154) = z(166)*z(813) + z(168)*z(817) + z(170)*z(821);
z(1258) = z(405)*z(1228) - z(412)*z(1227);
z(1265) = z(1154) + z(1258);
z(1164) = z(172)*z(813) + z(174)*z(817) + z(176)*z(821);
z(1248) = z(412)*z(1226) - z(398)*z(1228);
z(1272) = z(1164) + z(1248);
z(1174) = z(173)*z(813) + z(175)*z(817) + z(177)*z(821);
z(1238) = z(398)*z(1227) - z(405)*z(1226);
z(1279) = z(1174) + z(1238);
z(1812) = z(209)*z(1265) + z(212)*z(1272) + z(215)*z(1279);
z(1897) = z(422)*z(1883) - z(432)*z(1882);
z(1953) = z(1812) + z(1897);
z(1780) = z(207)*z(1265) + z(210)*z(1272) + z(213)*z(1279);
z(1923) = z(432)*z(1884) - z(442)*z(1883);
z(1933) = z(1780) + z(1923);
z(1796) = z(208)*z(1265) + z(211)*z(1272) + z(214)*z(1279);
z(1910) = z(442)*z(1882) - z(422)*z(1884);
z(1943) = z(1796) + z(1910);
z(3576) = 0.000210826210826210 6*z(71) + z(3123)*AOAB3 - 2.991452991452993E-05*z(70) - z(3124)*AOAB2;
z(3577) = 2.991452991452993E-05*z(69) + z(3124)*AOAB1 - 0.001911680911680912*z(71) - z(3122)*AOAB3;
z(3578) = 0.001911680911680912*z(70) + z(3122)*AOAB2 - 0.000210826210826210 6*z(69) - z(3123)*AOAB1;
z(3570) = 5.579999999999999E-05*(-13.63082437275986+7.924731182795699*T-T^2)*z(4)*z(69) + z(4)*z(3122)*AON2 + z(51)*z(3122)*AON3 + z(55)*z(3123)*AON3 + z(56)*z(3124)*AON3 + z(62)*z(70)*AON3 + z(64)*z(71)*AON3 + z(68)*z(69)*AON3 + z(3)*QNA_2p*z(69)*AON2 - 5.579999999999999E-05*(-13.63082437275986+7.924731182795699*T-T^2)*z(57)*z(70) - 5.579999999999999E-05*(-13.63082437275986+7.924731182795699*T-T^2)*z(58)*z(71) - z(57)*z(3123)*AON2 - z(58)*z(3124)*AON2 - z(63)*z(70)*AON2 - z(65)*z(71)*AON2;
z(3571) = 0.0001428*(-23.31372549019608+9.913165266106441*T-T^2)*z(4)*z(69) + z(57)*z(3123)*AON1 + z(58)*z(3124)*AON1 + z(63)*z(70)*AON1 + z(65)*z(71)*AON1 - 0.0001428*(-23.31372549019608+9.913165266106441*T-T^2)*z(57)*z(70) - 0.0001428*(-23.31372549019608+9.913165266106441*T-T^2)*z(58)*z(71) - z(4)*z(3122)*AON1 - z(49)*z(3122)*AON3 - z(53)*z(3123)*AON3 - z(54)*z(3124)*AON3 - z(60)*z(70)*AON3 - z(66)*z(71)*AON3 - z(67)*z(69)*AON3 - z(3)*QNA_2p*z(69)*AON1;
z(3572) = 5.579999999999999E-05*(-13.63082437275986+7.924731182795699*T-T^2)*z(49)*z(69) + 5.579999999999999E-05*(-13.63082437275986+7.924731182795699*T-T^2)*z(53)*z(70) + 5.579999999999999E-05*(-13.63082437275986+7.924731182795699*T-T^2)*z(54)*z(71) + 0.0001428*(-23.31372549019608+9.913165266106441*T-T^2)*z(51)*z(69) + 0.0001428*(-23.31372549019608+9.913165266106441*T-T^2)*z(55)*z(70) + 0.0001428*(-23.31372549019608+9.913165266106441*T-T^2)*z(56)*z(71) + z(49)*z(3122)*AON2 + z(53)*z(3123)*AON2 + z(54)*z(3124)*AON2 + z(60)*z(70)*AON2 + z(66)*z(71)*AON2 + z(67)*z(69)*AON2 - z(51)*z(3122)*AON1 - z(55)*z(3123)*AON1 - z(56)*z(3124)*AON1 - z(62)*z(70)*AON1 - z(64)*z(71)*AON1 - z(68)*z(69)*AON1;
z(3579) = z(82)*z(3576) + z(85)*z(3577) + z(88)*z(3578) + z(113)*z(3570) + z(114)*z(3571) + z(107)*z(526) + z(108)*z(528) + z(109)*z(527) + z(138)*z(525) + z(141)*z(524) - z(10)*z(3572) - z(9)*QNB_2p*z(523);
z(3603) = z(3133)*z(564) - z(3139)*z(563);
z(3585) = z(80)*z(3572) + z(83)*z(3576) + z(86)*z(3577) + z(89)*z(3578) + z(115)*z(3570) + z(116)*z(3571) + z(94)*z(523) + z(95)*z(528) + z(98)*z(527) + z(101)*z(526) + z(146)*z(525) + z(147)*z(524);
z(3604) = z(3139)*z(562) - z(3125)*z(564);
z(3591) = z(81)*z(3572) + z(84)*z(3576) + z(87)*z(3577) + z(90)*z(3578) + z(117)*z(3570) + z(118)*z(3571) + z(102)*z(523) + z(104)*z(527) + z(105)*z(526) + z(106)*z(528) + z(149)*z(525) + z(150)*z(524);
z(3605) = z(3125)*z(563) - z(3133)*z(562);
z(3606) = z(160)*z(579) + z(161)*z(577) + z(162)*z(578) + z(129)*(z(3579)+z(3603)) + z(132)*(z(3585)+z(3604)) + z(135)*(z(3591)+z(3605));
z(3657) = z(3180)*z(788) - z(3191)*z(787);
z(3627) = z(145)*z(577) + z(148)*z(578) + z(151)*z(579) + z(130)*(z(3579)+z(3603)) + z(133)*(z(3585)+z(3604)) + z(136)*(z(3591)+z(3605));
z(3658) = z(3191)*z(786) - z(3168)*z(788);
z(3639) = z(155)*z(578) + z(156)*z(579) + z(157)*z(577) + z(131)*(z(3579)+z(3603)) + z(134)*(z(3585)+z(3604)) + z(137)*(z(3591)+z(3605));
z(3659) = z(3168)*z(787) - z(3180)*z(786);
z(3660) = z(187)*z(821) + z(188)*z(813) + z(189)*z(817) + z(166)*(z(3606)+z(3657)) + z(168)*(z(3627)+z(3658)) + z(170)*(z(3639)+z(3659));
z(3747) = z(3227)*z(1228) - z(3239)*z(1227);
z(3699) = z(179)*z(813) + z(181)*z(817) + z(183)*z(821) + z(172)*(z(3606)+z(3657)) + z(174)*(z(3627)+z(3658)) + z(176)*(z(3639)+z(3659));
z(3748) = z(3239)*z(1226) - z(3215)*z(1228);
z(3720) = z(184)*z(817) + z(185)*z(821) + z(186)*z(813) + z(173)*(z(3606)+z(3657)) + z(175)*(z(3627)+z(3658)) + z(177)*(z(3639)+z(3659));
z(3749) = z(3215)*z(1227) - z(3227)*z(1226);
z(3750) = z(236)*z(1279) + z(237)*z(1265) + z(238)*z(1272) + z(207)*(z(3660)+z(3747)) + z(210)*(z(3699)+z(3748)) + z(213)*(z(3720)+z(3749));
z(3873) = z(3290)*z(1884) - z(3307)*z(1883);
z(3807) = z(220)*z(1265) + z(224)*z(1272) + z(227)*z(1279) + z(208)*(z(3660)+z(3747)) + z(211)*(z(3699)+z(3748)) + z(214)*(z(3720)+z(3749));
z(3874) = z(3307)*z(1882) - z(3272)*z(1884);
z(3837) = z(231)*z(1272) + z(232)*z(1279) + z(233)*z(1265) + z(209)*(z(3660)+z(3747)) + z(212)*(z(3699)+z(3748)) + z(215)*(z(3720)+z(3749));
z(3875) = z(3272)*z(1883) - z(3290)*z(1882);
z(3876) = z(277)*z(1953) + z(278)*z(1933) + z(279)*z(1943) + z(252)*(z(3750)+z(3873)) + z(255)*(z(3807)+z(3874)) + z(258)*(z(3837)+z(3875));

z(4029) = z(3383)*z(2531) + z(253)*z(3272)*z(2531) + z(256)*z(3290)*z(2531) + z(259)*z(3307)*z(2531) + z(265)*z(422)*z(2531) +
z(268)*z(432)*z(2531) + z(271)*z(442)*z(2531) - z(3387)*z(2530) - z(254)*z(3272)*z(2530) - z(257)*z(3290)*z(2530) - z(260)*z(3307)*z(2530) -
z(274)*z(432)*z(2530) - z(275)*z(442)*z(2530) - z(276)*z(422)*z(2530);
z(4030) = z(3387)*HOHK2 + z(254)*z(3272)*HOHK2 + z(257)*z(3290)*HOHK2 + z(260)*z(3307)*HOHK2 + z(305)*z(3456)*HOHK2 +
z(306)*z(3463)*HOHK2 + z(307)*z(3469)*HOHK2 + z(274)*z(432)*HOHK2 + z(275)*z(442)*HOHK2 + z(276)*z(422)*HOHK2 +
z(321)*z(334)*HOHK2 + z(326)*z(335)*HOHK2 + z(330)*z(333)*HOHK2 - z(3383)*HOHK3 - z(253)*z(3272)*HOHK3 - z(256)*z(3290)*HOHK3 -
z(259)*z(3307)*HOHK3 - z(302)*z(3456)*HOHK3 - z(303)*z(3463)*HOHK3 - z(304)*z(3469)*HOHK3 - z(265)*z(422)*HOHK3 -
z(268)*z(432)*HOHK3 - z(271)*z(442)*HOHK3 - z(318)*z(334)*HOHK3 - z(325)*z(335)*HOHK3 - z(332)*z(333)*HOHK3;
z(4031) = z(3876) + z(4029) + z(4030);
z(3948) = z(265)*z(1933) + z(268)*z(1943) + z(271)*z(1953) + z(253)*(z(3750)+z(3873)) + z(256)*(z(3807)+z(3874)) + z(259)*(z(3837)+z(3875));
z(4069) = z(3387)*z(2529) + z(254)*z(3272)*z(2529) + z(257)*z(3290)*z(2529) + z(260)*z(3307)*z(2529) + z(274)*z(432)*z(2529) +
z(275)*z(442)*z(2529) + z(276)*z(422)*z(2529) - z(3377)*z(2531) - z(252)*z(3272)*z(2531) - z(255)*z(3290)*z(2531) - z(258)*z(3307)*z(2531) -
z(277)*z(442)*z(2531) - z(278)*z(422)*z(2531) - z(279)*z(432)*z(2531);
z(4070) = z(3377)*HOHK3 + z(252)*z(3272)*HOHK3 + z(255)*z(3290)*HOHK3 + z(258)*z(3307)*HOHK3 + z(299)*z(3456)*HOHK3 +
z(300)*z(3463)*HOHK3 + z(301)*z(3469)*HOHK3 + z(277)*z(442)*HOHK3 + z(278)*z(422)*HOHK3 + z(279)*z(432)*HOHK3 +
z(315)*z(334)*HOHK3 + z(327)*z(335)*HOHK3 + z(331)*z(333)*HOHK3 - z(3387)*HOHK1 - z(254)*z(3272)*HOHK1 - z(257)*z(3290)*HOHK1 -
z(260)*z(3307)*HOHK1 - z(305)*z(3456)*HOHK1 - z(306)*z(3463)*HOHK1 - z(307)*z(3469)*HOHK1 - z(274)*z(432)*HOHK1 -
z(275)*z(442)*HOHK1 - z(276)*z(422)*HOHK1 - z(321)*z(334)*HOHK1 - z(326)*z(335)*HOHK1 - z(330)*z(333)*HOHK1;
z(4071) = z(3948) + z(4069) + z(4070);
z(3984) = z(274)*z(1943) + z(275)*z(1953) + z(276)*z(1933) + z(254)*(z(3750)+z(3873)) + z(257)*(z(3807)+z(3874)) + z(260)*(z(3837)+z(3875));
z(4109) = z(3377)*z(2530) + z(252)*z(3272)*z(2530) + z(255)*z(3290)*z(2530) + z(258)*z(3307)*z(2530) + z(277)*z(442)*z(2530) +
z(278)*z(422)*z(2530) + z(279)*z(432)*z(2530) - z(3383)*z(2529) - z(253)*z(3272)*z(2529) - z(256)*z(3290)*z(2529) - z(259)*z(3307)*z(2529) -
z(265)*z(422)*z(2529) - z(268)*z(432)*z(2529) - z(271)*z(442)*z(2529);
z(4110) = z(3383)*HOHK1 + z(253)*z(3272)*HOHK1 + z(256)*z(3290)*HOHK1 + z(259)*z(3307)*HOHK1 + z(302)*z(3456)*HOHK1 +
z(303)*z(3463)*HOHK1 + z(304)*z(3469)*HOHK1 + z(265)*z(422)*HOHK1 + z(268)*z(432)*HOHK1 + z(271)*z(442)*HOHK1 +
z(318)*z(334)*HOHK1 + z(325)*z(335)*HOHK1 + z(332)*z(333)*HOHK1 - z(3377)*HOHK2 - z(252)*z(3272)*HOHK2 - z(255)*z(3290)*HOHK2 -
z(258)*z(3307)*HOHK2 - z(299)*z(3456)*HOHK2 - z(300)*z(3463)*HOHK2 - z(301)*z(3469)*HOHK2 - z(277)*z(442)*HOHK2 -
z(278)*z(422)*HOHK2 - z(279)*z(432)*HOHK2 - z(315)*z(334)*HOHK2 - z(327)*z(335)*HOHK2 - z(331)*z(333)*HOHK2;
z(4111) = z(3984) + z(4109) + z(4110);
z(4149) = 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(28)*z(422) + 4.049999999999887E-05*(-
1+8.854320987654567*T-1.955555555555609*T^2)*z(34)*z(280) + 4.049999999999887E-05*(-1+8.854320987654567*T-
1.955555555555609*T^2)*z(40)*z(333) + z(28)*z(3272)*z(2611) + z(34)*z(3377)*z(2611) + z(40)*z(3456)*z(2611) + z(199)*z(3272)*z(2612) +
z(203)*z(3290)*z(2612) + z(204)*z(3307)*z(2612) + z(284)*z(3377)*z(2612) + z(286)*z(3383)*z(2612) + z(288)*z(3387)*z(2612) +
z(291)*z(3456)*z(2612) + z(295)*z(3463)*z(2612) + z(296)*z(3469)*z(2612) + z(219)*z(432)*z(2612) + z(229)*z(442)*z(2612) +
z(235)*z(422)*z(2612) + z(280)*z(311)*z(2612) + z(281)*z(317)*z(2612) + z(282)*z(320)*z(2612) + z(313)*z(334)*z(2612) + z(324)*z(335)*z(2612)
+ z(329)*z(333)*z(2612) + z(27)*QNE_2p*z(422)*z(2611) + z(33)*QNH_2p*z(280)*z(2611) + z(39)*QNK_2p*z(333)*z(2611) -
4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(205)*z(432) - 4.049999999999887E-05*(-
1+8.854320987654567*T-1.955555555555609*T^2)*z(206)*z(442) - 4.049999999999887E-05*(-1+8.854320987654567*T-
1.955555555555609*T^2)*z(250)*z(281) - 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(251)*z(282) -
4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(297)*z(334) - 4.049999999999887E-05*(-
1+8.854320987654567*T-1.955555555555609*T^2)*z(298)*z(335) - z(205)*z(3290)*z(2611) - z(206)*z(3307)*z(2611) - z(250)*z(3383)*z(2611) -
z(251)*z(3387)*z(2611) - z(297)*z(3463)*z(2611) - z(298)*z(3469)*z(2611) - z(221)*z(432)*z(2611) - z(228)*z(442)*z(2611) - z(264)*z(281)*z(2611)
- z(272)*z(282)*z(2611) - z(314)*z(334)*z(2611) - z(322)*z(335)*z(2611);
z(4163) = (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(28)*z(422) + (0.001249+5.719999999999987E-05*T-
5.699999999999868E-06*T^2)*z(34)*z(280) + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(40)*z(333) +
z(205)*z(3290)*z(2610) + z(206)*z(3307)*z(2610) + z(250)*z(3383)*z(2610) + z(251)*z(3387)*z(2610) + z(297)*z(3463)*z(2610) +
z(298)*z(3469)*z(2610) + z(221)*z(432)*z(2610) + z(228)*z(442)*z(2610) + z(264)*z(281)*z(2610) + z(272)*z(282)*z(2610) +
z(314)*z(334)*z(2610) + z(322)*z(335)*z(2610) - (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(205)*z(432) -
(0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(206)*z(442) - (0.001249+5.719999999999987E-05*T-5.699999999999868E-
06*T^2)*z(250)*z(281) - (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(251)*z(282) - (0.001249+5.719999999999987E-
05*T-5.699999999999868E-06*T^2)*z(297)*z(334) - (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(298)*z(335) -
z(28)*z(3272)*z(2610) - z(34)*z(3377)*z(2610) - z(40)*z(3456)*z(2610) - z(197)*z(3272)*z(2612) - z(201)*z(3290)*z(2612) - z(202)*z(3307)*z(2612)
- z(283)*z(3377)*z(2612) - z(285)*z(3383)*z(2612) - z(287)*z(3387)*z(2612) - z(289)*z(3456)*z(2612) - z(293)*z(3463)*z(2612) -
z(294)*z(3469)*z(2612) - z(217)*z(432)*z(2612) - z(230)*z(442)*z(2612) - z(234)*z(422)*z(2612) - z(280)*z(308)*z(2612) - z(281)*z(316)*z(2612) -
z(282)*z(319)*z(2612) - z(310)*z(334)*z(2612) - z(323)*z(335)*z(2612) - z(328)*z(333)*z(2612) - z(27)*QNE_2p*z(422)*z(2610) -
z(33)*QNH_2p*z(280)*z(2610) - z(39)*QNK_2p*z(333)*z(2610);
z(2664) = z(197)*z(422)*z(2611) + z(201)*z(432)*z(2611) + z(202)*z(442)*z(2611) + z(283)*z(280)*z(2611) + z(285)*z(281)*z(2611) +
z(287)*z(282)*z(2611) + z(289)*z(333)*z(2611) + z(293)*z(334)*z(2611) + z(294)*z(335)*z(2611) - z(199)*z(422)*z(2610) - z(203)*z(432)*z(2610) -
z(204)*z(442)*z(2610) - z(284)*z(280)*z(2610) - z(286)*z(281)*z(2610) - z(288)*z(282)*z(2610) - z(291)*z(333)*z(2610) - z(295)*z(334)*z(2610) -
z(296)*z(335)*z(2610);
z(4297) = z(299)*z(359) + z(300)*z(362) + z(301)*z(365) + z(347)*z(331) + z(350)*z(315) + z(353)*z(327);
z(2406) = z(252)*z(1933) + z(255)*z(1943) + z(258)*z(1953);
z(2570) = z(281)*z(2531) + z(253)*z(422)*z(2531) + z(256)*z(432)*z(2531) + z(259)*z(442)*z(2531) - z(282)*z(2530) - z(254)*z(422)*z(2530) -
z(257)*z(432)*z(2530) - z(260)*z(442)*z(2530);
z(2583) = z(2406) + z(2570);
z(2651) = z(282)*HOHK2 + z(254)*z(422)*HOHK2 + z(257)*z(432)*HOHK2 + z(260)*z(442)*HOHK2 + z(305)*z(333)*HOHK2 +
z(306)*z(334)*HOHK2 + z(307)*z(335)*HOHK2 - z(281)*HOHK3 - z(253)*z(422)*HOHK3 - z(256)*z(432)*HOHK3 - z(259)*z(442)*HOHK3 -
z(302)*z(333)*HOHK3 - z(303)*z(334)*HOHK3 - z(304)*z(335)*HOHK3;
z(2703) = z(2583) + z(2651);
z(4298) = z(302)*z(359) + z(303)*z(362) + z(304)*z(365) + z(347)*z(332) + z(350)*z(318) + z(353)*z(325);
z(2428) = z(253)*z(1933) + z(256)*z(1943) + z(259)*z(1953);
z(2557) = z(282)*z(2529) + z(254)*z(422)*z(2529) + z(257)*z(432)*z(2529) + z(260)*z(442)*z(2529) - z(280)*z(2531) - z(252)*z(422)*z(2531) -
z(255)*z(432)*z(2531) - z(258)*z(442)*z(2531);
z(2596) = z(2428) + z(2557);
z(2638) = z(280)*HOHK3 + z(252)*z(422)*HOHK3 + z(255)*z(432)*HOHK3 + z(258)*z(442)*HOHK3 + z(299)*z(333)*HOHK3 +
z(300)*z(334)*HOHK3 + z(301)*z(335)*HOHK3 - z(282)*HOHK1 - z(254)*z(422)*HOHK1 - z(257)*z(432)*HOHK1 - z(260)*z(442)*HOHK1 -
z(305)*z(333)*HOHK1 - z(306)*z(334)*HOHK1 - z(307)*z(335)*HOHK1;
z(2716) = z(2596) + z(2638);

z(4299) = z(305)*z(359) + z(306)*z(362) + z(307)*z(365) + z(347)*z(330) + z(350)*z(321) + z(353)*z(326);
z(2450) = z(254)*z(1933) + z(257)*z(1943) + z(260)*z(1953);
z(2544) = z(280)*z(2530) + z(252)*z(422)*z(2530) + z(255)*z(432)*z(2530) + z(258)*z(442)*z(2530) - z(281)*z(2529) - z(253)*z(422)*z(2529) - z(256)*z(432)*z(2529) - z(259)*z(442)*z(2529);
z(2609) = z(2450) + z(2544);
z(2625) = z(281)*HOHK1 + z(253)*z(422)*HOHK1 + z(256)*z(432)*HOHK1 + z(259)*z(442)*HOHK1 + z(302)*z(333)*HOHK1 + z(303)*z(334)*HOHK1 + z(304)*z(335)*HOHK1 - z(280)*HOHK2 - z(252)*z(422)*HOHK2 - z(255)*z(432)*HOHK2 - z(258)*z(442)*HOHK2 - z(299)*z(333)*HOHK2 - z(300)*z(334)*HOHK2 - z(301)*z(335)*HOHK2;
z(2729) = z(2609) + z(2625);
z(4300) = z(44)*z(48)*QNS_1p + z(43)*z(357) - z(43)*z(47)*QNS_3p - z(44)*z(342)*QNS_3p;
z(2690) = z(28)*z(422)*z(2611) + z(34)*z(280)*z(2611) + z(40)*z(333)*z(2611) + z(199)*z(422)*z(2612) + z(203)*z(432)*z(2612) + z(204)*z(442)*z(2612) + z(284)*z(280)*z(2612) + z(286)*z(281)*z(2612) + z(288)*z(282)*z(2612) + z(291)*z(333)*z(2612) + z(295)*z(334)*z(2612) + z(296)*z(335)*z(2612) - z(205)*z(432)*z(2611) - z(206)*z(442)*z(2611) - z(250)*z(281)*z(2611) - z(251)*z(282)*z(2611) - z(297)*z(334)*z(2611) - z(298)*z(335)*z(2611);
z(4301) = z(43)*z(342)*QNS_3p + z(44)*z(357) - z(43)*z(48)*QNS_1p - z(44)*z(47)*QNS_3p;
z(2677) = z(205)*z(432)*z(2610) + z(206)*z(442)*z(2610) + z(250)*z(281)*z(2610) + z(251)*z(282)*z(2610) + z(297)*z(334)*z(2610) + z(298)*z(335)*z(2610) - z(28)*z(422)*z(2610) - z(34)*z(280)*z(2610) - z(40)*z(333)*z(2610) - z(197)*z(422)*z(2612) - z(201)*z(432)*z(2612) - z(202)*z(442)*z(2612) - z(283)*z(280)*z(2612) - z(285)*z(281)*z(2612) - z(287)*z(282)*z(2612) - z(289)*z(333)*z(2612) - z(293)*z(334)*z(2612) - z(294)*z(335)*z(2612);
z(4302) = z(359)*z(2777) + z(362)*z(2764) + z(365)*z(2751) + z(347)*z(4212) + z(350)*z(4216) + z(353)*z(4217) + z(344)*z(4177) + z(2779)*z(4031) + z(2782)*z(4071) + z(2785)*z(4111) + z(2788)*z(4149) + z(2791)*z(4163) + z(358)*z(2664) + z(4297)*z(2703) + z(4298)*z(2716) + z(4299)*z(2729) + z(4300)*z(2690) + z(4301)*z(2677);
z(4303) = z(252)*z(4297) + z(253)*z(4298) + z(254)*z(4299) + z(2779)*z(278) + z(2782)*z(265) + z(2785)*z(276);
z(4304) = z(255)*z(4297) + z(256)*z(4298) + z(257)*z(4299) + z(2779)*z(279) + z(2782)*z(268) + z(2785)*z(274);
z(4305) = z(258)*z(4297) + z(259)*z(4298) + z(260)*z(4299) + z(2779)*z(277) + z(2782)*z(271) + z(2785)*z(275);
z(2385) = z(207)*z(252) + z(208)*z(255) + z(209)*z(258);
z(2407) = z(207)*z(253) + z(208)*z(256) + z(209)*z(259);
z(2429) = z(207)*z(254) + z(208)*z(257) + z(209)*z(260);
z(3878) = z(207)*z(278) + z(208)*z(279) + z(209)*z(277) + z(252)*z(237) + z(255)*z(220) + z(258)*z(233);
z(3950) = z(207)*z(265) + z(208)*z(268) + z(209)*z(271) + z(253)*z(237) + z(256)*z(220) + z(259)*z(233);
z(3986) = z(207)*z(276) + z(208)*z(274) + z(209)*z(275) + z(254)*z(237) + z(257)*z(220) + z(260)*z(233);
z(4306) = z(2385)*z(4297) + z(2407)*z(4298) + z(2429)*z(4299) + z(2779)*z(3878) + z(2782)*z(3950) + z(2785)*z(3986);
z(2386) = z(210)*z(252) + z(211)*z(255) + z(212)*z(258);
z(2408) = z(210)*z(253) + z(211)*z(256) + z(212)*z(259);
z(2430) = z(210)*z(254) + z(211)*z(257) + z(212)*z(260);
z(3879) = z(210)*z(278) + z(211)*z(279) + z(212)*z(277) + z(252)*z(238) + z(255)*z(224) + z(258)*z(231);
z(3951) = z(210)*z(265) + z(211)*z(268) + z(212)*z(271) + z(253)*z(238) + z(256)*z(224) + z(259)*z(231);
z(3987) = z(210)*z(276) + z(211)*z(274) + z(212)*z(275) + z(254)*z(238) + z(257)*z(224) + z(260)*z(231);
z(4307) = z(2386)*z(4297) + z(2408)*z(4298) + z(2430)*z(4299) + z(2779)*z(3879) + z(2782)*z(3951) + z(2785)*z(3987);
z(2387) = z(213)*z(252) + z(214)*z(255) + z(215)*z(258);
z(2409) = z(213)*z(253) + z(214)*z(256) + z(215)*z(259);
z(2431) = z(213)*z(254) + z(214)*z(257) + z(215)*z(260);
z(3880) = z(213)*z(278) + z(214)*z(279) + z(215)*z(277) + z(252)*z(236) + z(255)*z(227) + z(258)*z(232);
z(3952) = z(213)*z(265) + z(214)*z(268) + z(215)*z(271) + z(253)*z(236) + z(256)*z(227) + z(259)*z(232);
z(3988) = z(213)*z(276) + z(214)*z(274) + z(215)*z(275) + z(254)*z(236) + z(257)*z(227) + z(260)*z(232);
z(4308) = z(2387)*z(4297) + z(2409)*z(4298) + z(2431)*z(4299) + z(2779)*z(3880) + z(2782)*z(3952) + z(2785)*z(3988);
z(1765) = z(166)*z(207) + z(172)*z(210) + z(173)*z(213);
z(1781) = z(166)*z(208) + z(172)*z(211) + z(173)*z(214);
z(1797) = z(166)*z(209) + z(172)*z(212) + z(173)*z(215);
z(2388) = z(252)*z(1765) + z(255)*z(1781) + z(258)*z(1797);
z(2410) = z(253)*z(1765) + z(256)*z(1781) + z(259)*z(1797);
z(2432) = z(254)*z(1765) + z(257)*z(1781) + z(260)*z(1797);
z(3752) = z(166)*z(237) + z(172)*z(238) + z(173)*z(236) + z(207)*z(188) + z(210)*z(179) + z(213)*z(186);
z(3809) = z(166)*z(220) + z(172)*z(224) + z(173)*z(227) + z(208)*z(188) + z(211)*z(179) + z(214)*z(186);
z(3839) = z(166)*z(233) + z(172)*z(231) + z(173)*z(232) + z(209)*z(188) + z(212)*z(179) + z(215)*z(186);
z(3881) = z(252)*z(3752) + z(255)*z(3809) + z(258)*z(3839) + z(1765)*z(278) + z(1781)*z(279) + z(1797)*z(277);
z(3953) = z(253)*z(3752) + z(256)*z(3809) + z(259)*z(3839) + z(1765)*z(265) + z(1781)*z(268) + z(1797)*z(271);
z(3989) = z(254)*z(3752) + z(257)*z(3809) + z(260)*z(3839) + z(1765)*z(276) + z(1781)*z(274) + z(1797)*z(275);
z(4309) = z(2388)*z(4297) + z(2410)*z(4298) + z(2432)*z(4299) + z(2779)*z(3881) + z(2782)*z(3953) + z(2785)*z(3989);
z(1766) = z(168)*z(207) + z(174)*z(210) + z(175)*z(213);
z(1782) = z(168)*z(208) + z(174)*z(211) + z(175)*z(214);
z(1798) = z(168)*z(209) + z(174)*z(212) + z(175)*z(215);
z(2389) = z(252)*z(1766) + z(255)*z(1782) + z(258)*z(1798);
z(2411) = z(253)*z(1766) + z(256)*z(1782) + z(259)*z(1798);
z(2433) = z(254)*z(1766) + z(257)*z(1782) + z(260)*z(1798);
z(3753) = z(168)*z(237) + z(174)*z(238) + z(175)*z(236) + z(207)*z(189) + z(210)*z(181) + z(213)*z(184);
z(3810) = z(168)*z(220) + z(174)*z(224) + z(175)*z(227) + z(208)*z(189) + z(211)*z(181) + z(214)*z(184);
z(3840) = z(168)*z(233) + z(174)*z(231) + z(175)*z(232) + z(209)*z(189) + z(212)*z(181) + z(215)*z(184);
z(3882) = z(252)*z(3753) + z(255)*z(3810) + z(258)*z(3840) + z(1766)*z(278) + z(1782)*z(279) + z(1798)*z(277);
z(3954) = z(253)*z(3753) + z(256)*z(3810) + z(259)*z(3840) + z(1766)*z(265) + z(1782)*z(268) + z(1798)*z(271);
z(3990) = z(254)*z(3753) + z(257)*z(3810) + z(260)*z(3840) + z(1766)*z(276) + z(1782)*z(274) + z(1798)*z(275);
z(4310) = z(2389)*z(4297) + z(2411)*z(4298) + z(2433)*z(4299) + z(2779)*z(3882) + z(2782)*z(3954) + z(2785)*z(3990);
z(1767) = z(170)*z(207) + z(176)*z(210) + z(177)*z(213);
z(1783) = z(170)*z(208) + z(176)*z(211) + z(177)*z(214);
z(1799) = z(170)*z(209) + z(176)*z(212) + z(177)*z(215);
z(2390) = z(252)*z(1767) + z(255)*z(1783) + z(258)*z(1799);
z(2412) = z(253)*z(1767) + z(256)*z(1783) + z(259)*z(1799);
z(2434) = z(254)*z(1767) + z(257)*z(1783) + z(260)*z(1799);

z(3754) = z(170)*z(237) + z(176)*z(238) + z(177)*z(236) + z(207)*z(187) + z(210)*z(183) + z(213)*z(185);
z(3811) = z(170)*z(220) + z(176)*z(224) + z(177)*z(227) + z(208)*z(187) + z(211)*z(183) + z(214)*z(185);
z(3841) = z(170)*z(233) + z(176)*z(231) + z(177)*z(232) + z(209)*z(187) + z(212)*z(183) + z(215)*z(185);
z(3883) = z(252)*z(3754) + z(255)*z(3811) + z(258)*z(3841) + z(1767)*z(278) + z(1783)*z(279) + z(1799)*z(277);
z(3955) = z(253)*z(3754) + z(256)*z(3811) + z(259)*z(3841) + z(1767)*z(265) + z(1783)*z(268) + z(1799)*z(271);
z(3991) = z(254)*z(3754) + z(257)*z(3811) + z(260)*z(3841) + z(1767)*z(276) + z(1783)*z(274) + z(1799)*z(275);
z(4311) = z(2390)*z(4297) + z(2412)*z(4298) + z(2434)*z(4299) + z(2779)*z(3883) + z(2782)*z(3955) + z(2785)*z(3991);
z(1145) = z(129)*z(166) + z(130)*z(168) + z(131)*z(170);
z(1155) = z(129)*z(172) + z(130)*z(174) + z(131)*z(176);
z(1165) = z(129)*z(173) + z(130)*z(175) + z(131)*z(177);
z(1768) = z(207)*z(1145) + z(210)*z(1155) + z(213)*z(1165);
z(1784) = z(208)*z(1145) + z(211)*z(1155) + z(214)*z(1165);
z(1800) = z(209)*z(1145) + z(212)*z(1155) + z(215)*z(1165);
z(2391) = z(252)*z(1768) + z(255)*z(1784) + z(258)*z(1800);
z(2413) = z(253)*z(1768) + z(256)*z(1784) + z(259)*z(1800);
z(2435) = z(254)*z(1768) + z(257)*z(1784) + z(260)*z(1800);
z(3662) = z(129)*z(188) + z(130)*z(189) + z(131)*z(187) + z(166)*z(161) + z(168)*z(145) + z(170)*z(157);
z(3701) = z(129)*z(179) + z(130)*z(181) + z(131)*z(183) + z(172)*z(161) + z(174)*z(145) + z(176)*z(157);
z(3722) = z(129)*z(186) + z(130)*z(184) + z(131)*z(185) + z(173)*z(161) + z(175)*z(145) + z(177)*z(157);
z(3755) = z(207)*z(3662) + z(210)*z(3701) + z(213)*z(3722) + z(1145)*z(237) + z(1155)*z(238) + z(1165)*z(236);
z(3812) = z(208)*z(3662) + z(211)*z(3701) + z(214)*z(3722) + z(1145)*z(220) + z(1155)*z(224) + z(1165)*z(227);
z(3842) = z(209)*z(3662) + z(212)*z(3701) + z(215)*z(3722) + z(1145)*z(233) + z(1155)*z(231) + z(1165)*z(232);
z(3884) = z(252)*z(3755) + z(255)*z(3812) + z(258)*z(3842) + z(1768)*z(278) + z(1784)*z(279) + z(1800)*z(277);
z(3956) = z(253)*z(3755) + z(256)*z(3812) + z(259)*z(3842) + z(1768)*z(265) + z(1784)*z(268) + z(1800)*z(271);
z(3992) = z(254)*z(3755) + z(257)*z(3812) + z(260)*z(3842) + z(1768)*z(276) + z(1784)*z(274) + z(1800)*z(275);
z(4312) = z(2391)*z(4297) + z(2413)*z(4298) + z(2435)*z(4299) + z(2779)*z(3884) + z(2782)*z(3956) + z(2785)*z(3992);
z(1146) = z(132)*z(166) + z(133)*z(168) + z(134)*z(170);
z(1156) = z(132)*z(172) + z(133)*z(174) + z(134)*z(176);
z(1166) = z(132)*z(173) + z(133)*z(175) + z(134)*z(177);
z(1769) = z(207)*z(1146) + z(210)*z(1156) + z(213)*z(1166);
z(1785) = z(208)*z(1146) + z(211)*z(1156) + z(214)*z(1166);
z(1801) = z(209)*z(1146) + z(212)*z(1156) + z(215)*z(1166);
z(2392) = z(252)*z(1769) + z(255)*z(1785) + z(258)*z(1801);
z(2414) = z(253)*z(1769) + z(256)*z(1785) + z(259)*z(1801);
z(2436) = z(254)*z(1769) + z(257)*z(1785) + z(260)*z(1801);
z(3663) = z(132)*z(188) + z(133)*z(189) + z(134)*z(187) + z(166)*z(162) + z(168)*z(148) + z(170)*z(155);
z(3702) = z(132)*z(179) + z(133)*z(181) + z(134)*z(183) + z(172)*z(162) + z(174)*z(148) + z(176)*z(155);
z(3723) = z(132)*z(186) + z(133)*z(184) + z(134)*z(185) + z(173)*z(162) + z(175)*z(148) + z(177)*z(155);
z(3756) = z(207)*z(3663) + z(210)*z(3702) + z(213)*z(3723) + z(1146)*z(237) + z(1156)*z(238) + z(1166)*z(236);
z(3813) = z(208)*z(3663) + z(211)*z(3702) + z(214)*z(3723) + z(1146)*z(220) + z(1156)*z(224) + z(1166)*z(227);
z(3843) = z(209)*z(3663) + z(212)*z(3702) + z(215)*z(3723) + z(1146)*z(233) + z(1156)*z(231) + z(1166)*z(232);
z(3885) = z(252)*z(3756) + z(255)*z(3813) + z(258)*z(3843) + z(1769)*z(278) + z(1785)*z(279) + z(1801)*z(277);
z(3957) = z(253)*z(3756) + z(256)*z(3813) + z(259)*z(3843) + z(1769)*z(265) + z(1785)*z(268) + z(1801)*z(271);
z(3993) = z(254)*z(3756) + z(257)*z(3813) + z(260)*z(3843) + z(1769)*z(276) + z(1785)*z(274) + z(1801)*z(275);
z(4313) = z(2392)*z(4297) + z(2414)*z(4298) + z(2436)*z(4299) + z(2779)*z(3885) + z(2782)*z(3957) + z(2785)*z(3993);
z(1147) = z(135)*z(166) + z(136)*z(168) + z(137)*z(170);
z(1157) = z(135)*z(172) + z(136)*z(174) + z(137)*z(176);
z(1167) = z(135)*z(173) + z(136)*z(175) + z(137)*z(177);
z(1770) = z(207)*z(1147) + z(210)*z(1157) + z(213)*z(1167);
z(1786) = z(208)*z(1147) + z(211)*z(1157) + z(214)*z(1167);
z(1802) = z(209)*z(1147) + z(212)*z(1157) + z(215)*z(1167);
z(2393) = z(252)*z(1770) + z(255)*z(1786) + z(258)*z(1802);
z(2415) = z(253)*z(1770) + z(256)*z(1786) + z(259)*z(1802);
z(2437) = z(254)*z(1770) + z(257)*z(1786) + z(260)*z(1802);
z(3664) = z(135)*z(188) + z(136)*z(189) + z(137)*z(187) + z(166)*z(160) + z(168)*z(151) + z(170)*z(156);
z(3703) = z(135)*z(179) + z(136)*z(181) + z(137)*z(183) + z(172)*z(160) + z(174)*z(151) + z(176)*z(156);
z(3724) = z(135)*z(186) + z(136)*z(184) + z(137)*z(185) + z(173)*z(160) + z(175)*z(151) + z(177)*z(156);
z(3757) = z(207)*z(3664) + z(210)*z(3703) + z(213)*z(3724) + z(1147)*z(237) + z(1157)*z(238) + z(1167)*z(236);
z(3814) = z(208)*z(3664) + z(211)*z(3703) + z(214)*z(3724) + z(1147)*z(220) + z(1157)*z(224) + z(1167)*z(227);
z(3844) = z(209)*z(3664) + z(212)*z(3703) + z(215)*z(3724) + z(1147)*z(233) + z(1157)*z(231) + z(1167)*z(232);
z(3886) = z(252)*z(3757) + z(255)*z(3814) + z(258)*z(3844) + z(1770)*z(278) + z(1786)*z(279) + z(1802)*z(277);
z(3958) = z(253)*z(3757) + z(256)*z(3814) + z(259)*z(3844) + z(1770)*z(265) + z(1786)*z(268) + z(1802)*z(271);
z(3994) = z(254)*z(3757) + z(257)*z(3814) + z(260)*z(3844) + z(1770)*z(276) + z(1786)*z(274) + z(1802)*z(275);
z(4314) = z(2393)*z(4297) + z(2415)*z(4298) + z(2437)*z(4299) + z(2779)*z(3886) + z(2782)*z(3958) + z(2785)*z(3994);
z(4242) = 0.0005*z(197)*z(4206) + 0.0005*z(201)*z(4207) + 0.0005*z(202)*z(4208) + 0.0005*z(2731)*z(234) + 0.0005*z(2734)*z(217) +
0.0005*z(2737)*z(230) - 0.00893*z(197)*z(4209) - 0.00893*z(201)*z(4210) - 0.00893*z(202)*z(4211) - 0.00893*z(2732)*z(234) -
0.00893*z(2735)*z(217) - 0.00893*z(2738)*z(230);
z(4243) = -0.01653*z(197)*z(4209) - 0.01653*z(201)*z(4210) - 0.01653*z(202)*z(4211) - 0.01653*z(2732)*z(234) - 0.01653*z(2735)*z(217) -
0.01653*z(2738)*z(230) - 0.0005*z(197)*z(4213) - 0.0005*z(201)*z(4214) - 0.0005*z(202)*z(4215) - 0.0005*z(2730)*z(234) - 0.0005*z(2733)*z(217)
- 0.0005*z(2736)*z(230);
z(4244) = 0.00893*z(197)*z(4213) + 0.00893*z(201)*z(4214) + 0.00893*z(202)*z(4215) + 0.00893*z(2730)*z(234) + 0.00893*z(2733)*z(217) +
0.00893*z(2736)*z(230) + 0.01653*z(197)*z(4206) + 0.01653*z(201)*z(4207) + 0.01653*z(202)*z(4208) + 0.01653*z(2731)*z(234) +
0.01653*z(2734)*z(217) + 0.01653*z(2737)*z(230);
z(2739) = 0.00893*z(197)*z(2730) + 0.00893*z(201)*z(2733) + 0.00893*z(202)*z(2736) + 0.01653*z(197)*z(2731) + 0.01653*z(201)*z(2734) +
0.01653*z(202)*z(2737);
z(2753) = -0.01653*z(197)*z(2732) - 0.01653*z(201)*z(2735) - 0.01653*z(202)*z(2738) - 0.0005*z(197)*z(2730) - 0.0005*z(201)*z(2733) -
0.0005*z(202)*z(2736);

z(2765) = 0.0005*z(197)*z(2731) + 0.0005*z(201)*z(2734) + 0.0005*z(202)*z(2737) - 0.00893*z(197)*z(2732) - 0.00893*z(201)*z(2735) - 0.00893*z(202)*z(2738);
z(4178) = (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(197)*z(199) + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(201)*z(203) + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(202)*z(204) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(197)^2 + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(201)^2 + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(202)^2 + 2*z(197)*z(234)*z(2611) + 2*z(201)*z(217)*z(2611) + 2*z(202)*z(230)*z(2611) - z(197)*z(235)*z(2610) - z(199)*z(234)*z(2610) - z(201)*z(219)*z(2610) - z(202)*z(229)*z(2610) - z(203)*z(217)*z(2610) - z(204)*z(230)*z(2610);
z(3887) = z(217)*z(1884) - z(230)*z(1883);
z(3888) = z(230)*z(1882) - z(234)*z(1884);
z(3889) = z(234)*z(1883) - z(217)*z(1882);
z(1885) = z(197)*z(1883) - z(201)*z(1882);
z(1911) = z(201)*z(1884) - z(202)*z(1883);
z(1898) = z(202)*z(1882) - z(197)*z(1884);
z(3890) = z(252)*z(3887) + z(255)*z(3888) + z(258)*z(3889) + z(277)*z(1885) + z(278)*z(1911) + z(279)*z(1898);
z(4032) = z(197)*z(265)*z(2531) + z(201)*z(268)*z(2531) + z(202)*z(271)*z(2531) + z(253)*z(234)*z(2531) + z(256)*z(217)*z(2531) + z(259)*z(230)*z(2531) - z(197)*z(276)*z(2530) - z(201)*z(274)*z(2530) - z(202)*z(275)*z(2530) - z(254)*z(234)*z(2530) - z(257)*z(217)*z(2530) - z(260)*z(230)*z(2530);
z(4033) = z(197)*z(276)*HOHK2 + z(201)*z(274)*HOHK2 + z(202)*z(275)*HOHK2 + z(254)*z(234)*HOHK2 + z(257)*z(217)*HOHK2 + z(260)*z(230)*HOHK2 - z(197)*z(265)*HOHK3 - z(201)*z(268)*HOHK3 - z(202)*z(271)*HOHK3 - z(253)*z(234)*HOHK3 - z(256)*z(217)*HOHK3 - z(259)*z(230)*HOHK3;
z(4034) = z(3890) + z(4032) + z(4033);
z(3959) = z(253)*z(3887) + z(256)*z(3888) + z(259)*z(3889) + z(265)*z(1911) + z(268)*z(1898) + z(271)*z(1885);
z(4072) = z(197)*z(276)*z(2529) + z(201)*z(274)*z(2529) + z(202)*z(275)*z(2529) + z(254)*z(234)*z(2529) + z(257)*z(217)*z(2529) + z(260)*z(230)*z(2529) - z(197)*z(278)*z(2531) - z(201)*z(279)*z(2531) - z(202)*z(277)*z(2531) - z(252)*z(234)*z(2531) - z(255)*z(217)*z(2531) - z(258)*z(230)*z(2531);
z(4073) = z(197)*z(278)*HOHK3 + z(201)*z(279)*HOHK3 + z(202)*z(277)*HOHK3 + z(252)*z(234)*HOHK3 + z(255)*z(217)*HOHK3 + z(258)*z(230)*HOHK3 - z(197)*z(276)*HOHK1 - z(201)*z(274)*HOHK1 - z(202)*z(275)*HOHK1 - z(254)*z(234)*HOHK1 - z(257)*z(217)*HOHK1 - z(260)*z(230)*HOHK1;
z(4074) = z(3959) + z(4072) + z(4073);
z(3995) = z(254)*z(3887) + z(257)*z(3888) + z(260)*z(3889) + z(274)*z(1898) + z(275)*z(1885) + z(276)*z(1911);
z(4112) = z(197)*z(278)*z(2530) + z(201)*z(279)*z(2530) + z(202)*z(277)*z(2530) + z(252)*z(234)*z(2530) + z(255)*z(217)*z(2530) + z(258)*z(230)*z(2530) - z(197)*z(265)*z(2529) - z(201)*z(268)*z(2529) - z(202)*z(271)*z(2529) - z(253)*z(234)*z(2529) - z(256)*z(217)*z(2529) - z(259)*z(230)*z(2529);
z(4113) = z(197)*z(265)*HOHK1 + z(201)*z(268)*HOHK1 + z(202)*z(271)*HOHK1 + z(253)*z(234)*HOHK1 + z(256)*z(217)*HOHK1 + z(259)*z(230)*HOHK1 - z(197)*z(278)*HOHK2 - z(201)*z(279)*HOHK2 - z(202)*z(277)*HOHK2 - z(252)*z(234)*HOHK2 - z(255)*z(217)*HOHK2 - z(258)*z(230)*HOHK2;
z(4114) = z(3995) + z(4112) + z(4113);
z(4150) = 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(28)*z(197) + z(28)*z(234)*z(2611) + z(197)*z(235)*z(2612) + z(199)*z(234)*z(2612) + z(201)*z(219)*z(2612) + z(202)*z(229)*z(2612) + z(203)*z(217)*z(2612) + z(204)*z(230)*z(2612) + z(27)*z(197)*QNE_2p*z(2611) - 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(201)*z(205) - 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(202)*z(206) - z(201)*z(221)*z(2611) - z(202)*z(228)*z(2611) - z(205)*z(217)*z(2611) - z(206)*z(230)*z(2611);
z(4164) = (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(28)*z(197) + z(201)*z(221)*z(2610) + z(202)*z(228)*z(2610) + z(205)*z(217)*z(2610) + z(206)*z(230)*z(2610) - (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(201)*z(205) - (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(202)*z(206) - 2*z(197)*z(234)*z(2612) - 2*z(201)*z(217)*z(2612) - 2*z(202)*z(230)*z(2612) - z(28)*z(234)*z(2610) - z(27)*z(197)*QNE_2p*z(2610);
z(2652) = z(197)^2*z(2611) + z(201)^2*z(2611) + z(202)^2*z(2611) - z(197)*z(199)*z(2610) - z(201)*z(203)*z(2610) - z(202)*z(204)*z(2610);
z(2394) = z(252)*z(1911) + z(255)*z(1898) + z(258)*z(1885);
z(2558) = z(197)*z(253)*z(2531) + z(201)*z(256)*z(2531) + z(202)*z(259)*z(2531) - z(197)*z(254)*z(2530) - z(201)*z(257)*z(2530) - z(202)*z(260)*z(2530);
z(2571) = z(2394) + z(2558);
z(2639) = z(197)*z(254)*HOHK2 + z(201)*z(257)*HOHK2 + z(202)*z(260)*HOHK2 - z(197)*z(253)*HOHK3 - z(201)*z(256)*HOHK3 - z(202)*z(259)*HOHK3;
z(2691) = z(2571) + z(2639);
z(2416) = z(253)*z(1911) + z(256)*z(1898) + z(259)*z(1885);
z(2545) = z(197)*z(254)*z(2529) + z(201)*z(257)*z(2529) + z(202)*z(260)*z(2529) - z(197)*z(252)*z(2531) - z(201)*z(255)*z(2531) - z(202)*z(258)*z(2531);
z(2584) = z(2416) + z(2545);
z(2626) = z(197)*z(252)*HOHK3 + z(201)*z(255)*HOHK3 + z(202)*z(258)*HOHK3 - z(197)*z(254)*HOHK1 - z(201)*z(257)*HOHK1 - z(202)*z(260)*HOHK1;
z(2704) = z(2584) + z(2626);
z(2438) = z(254)*z(1911) + z(257)*z(1898) + z(260)*z(1885);
z(2532) = z(197)*z(252)*z(2530) + z(201)*z(255)*z(2530) + z(202)*z(258)*z(2530) - z(197)*z(253)*z(2529) - z(201)*z(256)*z(2529) - z(202)*z(259)*z(2529);
z(2597) = z(2438) + z(2532);
z(2613) = z(197)*z(253)*HOHK1 + z(201)*z(256)*HOHK1 + z(202)*z(259)*HOHK1 - z(197)*z(252)*HOHK2 - z(201)*z(255)*HOHK2 - z(202)*z(258)*HOHK2;
z(2717) = z(2597) + z(2613);
z(2678) = z(28)*z(197)*z(2611) + z(197)*z(199)*z(2612) + z(201)*z(203)*z(2612) + z(202)*z(204)*z(2612) - z(201)*z(205)*z(2611) - z(202)*z(206)*z(2611);
z(2665) = z(201)*z(205)*z(2610) + z(202)*z(206)*z(2610) - z(28)*z(197)*z(2610) - z(197)^2*z(2612) - z(201)^2*z(2612) - z(202)^2*z(2612);
z(4315) = z(347)*z(4242) + z(350)*z(4243) + z(353)*z(4244) + z(2739)*z(365) + z(2753)*z(362) + z(2765)*z(359) + z(344)*z(4178) + z(2779)*z(4034) + z(2782)*z(4074) + z(2785)*z(4114) + z(2788)*z(4150) + z(2791)*z(4164) + z(358)*z(2652) + z(4297)*z(2691) + z(4298)*z(2704) + z(4299)*z(2717) + z(4300)*z(2678) + z(4301)*z(2665);
z(4247) = 0.0005*z(199)*z(4206) + 0.0005*z(203)*z(4207) + 0.0005*z(204)*z(4208) + 0.0005*z(2731)*z(235) + 0.0005*z(2734)*z(219) + 0.0005*z(2737)*z(229) - 0.00893*z(199)*z(4209) - 0.00893*z(203)*z(4210) - 0.00893*z(204)*z(4211) - 0.00893*z(2732)*z(235) - 0.00893*z(2735)*z(219) - 0.00893*z(2738)*z(229);

z(4248) = -0.01653*z(199)*z(4209) - 0.01653*z(203)*z(4210) - 0.01653*z(204)*z(4211) - 0.01653*z(2732)*z(235) - 0.01653*z(2735)*z(219) - 0.01653*z(2738)*z(229) - 0.0005*z(199)*z(4213) - 0.0005*z(203)*z(4214) - 0.0005*z(204)*z(4215) - 0.0005*z(2730)*z(235) - 0.0005*z(2733)*z(219) - 0.0005*z(2736)*z(229);
z(4249) = 0.00893*z(199)*z(4213) + 0.00893*z(203)*z(4214) + 0.00893*z(204)*z(4215) + 0.00893*z(2730)*z(235) + 0.00893*z(2733)*z(219) + 0.00893*z(2736)*z(229) + 0.01653*z(199)*z(4206) + 0.01653*z(203)*z(4207) + 0.01653*z(204)*z(4208) + 0.01653*z(2731)*z(235) + 0.01653*z(2734)*z(219) + 0.01653*z(2737)*z(229);
z(2740) = 0.00893*z(199)*z(2730) + 0.00893*z(203)*z(2733) + 0.00893*z(204)*z(2736) + 0.01653*z(199)*z(2731) + 0.01653*z(203)*z(2734) + 0.01653*z(204)*z(2737);
z(2754) = -0.01653*z(199)*z(2732) - 0.01653*z(203)*z(2735) - 0.01653*z(204)*z(2738) - 0.0005*z(199)*z(2730) - 0.0005*z(203)*z(2733) - 0.0005*z(204)*z(2736);
z(2766) = 0.0005*z(199)*z(2731) + 0.0005*z(203)*z(2734) + 0.0005*z(204)*z(2737) - 0.00893*z(199)*z(2732) - 0.00893*z(203)*z(2735) - 0.00893*z(204)*z(2738);
z(4179) = (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(199)^2 + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(203)^2 + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(204)^2 + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(197)*z(199) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(201)*z(203) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(202)*z(204) + z(197)*z(235)*z(2611) + z(199)*z(234)*z(2611) + z(201)*z(219)*z(2611) + z(202)*z(229)*z(2611) + z(203)*z(217)*z(2611) + z(204)*z(230)*z(2611) - 2*z(199)*z(235)*z(2610) - 2*z(203)*z(219)*z(2610) - 2*z(204)*z(229)*z(2610);
z(3892) = z(219)*z(1884) - z(229)*z(1883);
z(3893) = z(229)*z(1882) - z(235)*z(1884);
z(3894) = z(235)*z(1883) - z(219)*z(1882);
z(1886) = z(199)*z(1883) - z(203)*z(1882);
z(1912) = z(203)*z(1884) - z(204)*z(1883);
z(1899) = z(204)*z(1882) - z(199)*z(1884);
z(3895) = z(252)*z(3892) + z(255)*z(3893) + z(258)*z(3894) + z(277)*z(1886) + z(278)*z(1912) + z(279)*z(1899);
z(4035) = z(199)*z(265)*z(2531) + z(203)*z(268)*z(2531) + z(204)*z(271)*z(2531) + z(253)*z(235)*z(2531) + z(256)*z(219)*z(2531) + z(259)*z(229)*z(2531) - z(199)*z(276)*z(2530) - z(203)*z(274)*z(2530) - z(204)*z(275)*z(2530) - z(254)*z(235)*z(2530) - z(257)*z(219)*z(2530) - z(260)*z(229)*z(2530);
z(4036) = z(199)*z(276)*HOHK2 + z(203)*z(274)*HOHK2 + z(204)*z(275)*HOHK2 + z(254)*z(235)*HOHK2 + z(257)*z(219)*HOHK2 + z(260)*z(229)*HOHK2 - z(199)*z(265)*HOHK3 - z(203)*z(268)*HOHK3 - z(204)*z(271)*HOHK3 - z(253)*z(235)*HOHK3 - z(256)*z(219)*HOHK3 - z(259)*z(229)*HOHK3;
z(4037) = z(3895) + z(4035) + z(4036);
z(3961) = z(253)*z(3892) + z(256)*z(3893) + z(259)*z(3894) + z(265)*z(1912) + z(268)*z(1899) + z(271)*z(1886);
z(4075) = z(199)*z(276)*z(2529) + z(203)*z(274)*z(2529) + z(204)*z(275)*z(2529) + z(254)*z(235)*z(2529) + z(257)*z(219)*z(2529) + z(260)*z(229)*z(2529) - z(199)*z(278)*z(2531) - z(203)*z(279)*z(2531) - z(204)*z(277)*z(2531) - z(252)*z(235)*z(2531) - z(255)*z(219)*z(2531) - z(258)*z(229)*z(2531);
z(4076) = z(199)*z(278)*HOHK3 + z(203)*z(279)*HOHK3 + z(204)*z(277)*HOHK3 + z(252)*z(235)*HOHK3 + z(255)*z(219)*HOHK3 + z(258)*z(229)*HOHK3 - z(199)*z(276)*HOHK1 - z(203)*z(274)*HOHK1 - z(204)*z(275)*HOHK1 - z(254)*z(235)*HOHK1 - z(257)*z(219)*HOHK1 - z(260)*z(229)*HOHK1;
z(4077) = z(3961) + z(4075) + z(4076);
z(3997) = z(254)*z(3892) + z(257)*z(3893) + z(260)*z(3894) + z(274)*z(1899) + z(275)*z(1886) + z(276)*z(1912);
z(4115) = z(199)*z(278)*z(2530) + z(203)*z(279)*z(2530) + z(204)*z(277)*z(2530) + z(252)*z(235)*z(2530) + z(255)*z(219)*z(2530) + z(258)*z(229)*z(2530) - z(199)*z(265)*z(2529) - z(203)*z(268)*z(2529) - z(204)*z(271)*z(2529) - z(253)*z(235)*z(2529) - z(256)*z(219)*z(2529) - z(259)*z(229)*z(2529);
z(4116) = z(199)*z(265)*HOHK1 + z(203)*z(268)*HOHK1 + z(204)*z(271)*HOHK1 + z(253)*z(235)*HOHK1 + z(256)*z(219)*HOHK1 + z(259)*z(229)*HOHK1 - z(199)*z(278)*HOHK2 - z(203)*z(279)*HOHK2 - z(204)*z(277)*HOHK2 - z(252)*z(235)*HOHK2 - z(255)*z(219)*HOHK2 - z(258)*z(229)*HOHK2;
z(4117) = z(3997) + z(4115) + z(4116);
z(4151) = 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(28)*z(199) + z(28)*z(235)*z(2611) + 2*z(199)*z(235)*z(2612) + 2*z(203)*z(219)*z(2612) + 2*z(204)*z(229)*z(2612) + z(27)*z(199)*QNE_2p*z(2611) - 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(203)*z(205) - 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(204)*z(206) - z(203)*z(221)*z(2611) - z(204)*z(228)*z(2611) - z(205)*z(219)*z(2611) - z(206)*z(229)*z(2611);
z(4165) = (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(28)*z(199) + z(203)*z(221)*z(2610) + z(204)*z(228)*z(2610) + z(205)*z(219)*z(2610) + z(206)*z(229)*z(2610) - (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(203)*z(205) - (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(204)*z(206) - z(28)*z(235)*z(2610) - z(197)*z(235)*z(2612) - z(199)*z(234)*z(2612) - z(201)*z(219)*z(2612) - z(202)*z(229)*z(2612) - z(203)*z(217)*z(2612) - z(204)*z(230)*z(2612) - z(27)*z(199)*QNE_2p*z(2610);
z(2653) = z(197)*z(199)*z(2611) + z(201)*z(203)*z(2611) + z(202)*z(204)*z(2611) - z(199)^2*z(2610) - z(203)^2*z(2610) - z(204)^2*z(2610);
z(2395) = z(252)*z(1912) + z(255)*z(1899) + z(258)*z(1886);
z(2559) = z(199)*z(253)*z(2531) + z(203)*z(256)*z(2531) + z(204)*z(259)*z(2531) - z(199)*z(254)*z(2530) - z(203)*z(257)*z(2530) - z(204)*z(260)*z(2530);
z(2572) = z(2395) + z(2559);
z(2640) = z(199)*z(254)*HOHK2 + z(203)*z(257)*HOHK2 + z(204)*z(260)*HOHK2 - z(199)*z(253)*HOHK3 - z(203)*z(256)*HOHK3 - z(204)*z(259)*HOHK3;
z(2692) = z(2572) + z(2640);
z(2417) = z(253)*z(1912) + z(256)*z(1899) + z(259)*z(1886);
z(2546) = z(199)*z(254)*z(2529) + z(203)*z(257)*z(2529) + z(204)*z(260)*z(2529) - z(199)*z(252)*z(2531) - z(203)*z(255)*z(2531) - z(204)*z(258)*z(2531);
z(2585) = z(2417) + z(2546);
z(2627) = z(199)*z(252)*HOHK3 + z(203)*z(255)*HOHK3 + z(204)*z(258)*HOHK3 - z(199)*z(254)*HOHK1 - z(203)*z(257)*HOHK1 - z(204)*z(260)*HOHK1;
z(2705) = z(2585) + z(2627);
z(2439) = z(254)*z(1912) + z(257)*z(1899) + z(260)*z(1886);
z(2533) = z(199)*z(252)*z(2530) + z(203)*z(255)*z(2530) + z(204)*z(258)*z(2530) - z(199)*z(253)*z(2529) - z(203)*z(256)*z(2529) - z(204)*z(259)*z(2529);
z(2598) = z(2439) + z(2533);
z(2614) = z(199)*z(253)*HOHK1 + z(203)*z(256)*HOHK1 + z(204)*z(259)*HOHK1 - z(199)*z(252)*HOHK2 - z(203)*z(255)*HOHK2 - z(204)*z(258)*HOHK2;

z(2718) = z(2598) + z(2614);
z(2679) = z(28)*z(199)*z(2611) + z(199)^2*z(2612) + z(203)^2*z(2612) + z(204)^2*z(2612) - z(203)*z(205)*z(2611) - z(204)*z(206)*z(2611);
z(2666) = z(203)*z(205)*z(2610) + z(204)*z(206)*z(2610) - z(28)*z(199)*z(2610) - z(197)*z(199)*z(2612) - z(201)*z(203)*z(2612) - z(202)*z(204)*z(2612);
z(4316) = z(347)*z(4247) + z(350)*z(4248) + z(353)*z(4249) + z(2740)*z(365) + z(2754)*z(362) + z(2766)*z(359) + z(344)*z(4179) + z(2779)*z(4037) + z(2782)*z(4077) + z(2785)*z(4117) + z(2788)*z(4151) + z(2791)*z(4165) + z(358)*z(2653) + z(4297)*z(2692) + z(4298)*z(2705) + z(4299)*z(2718) + z(4300)*z(2679) + z(4301)*z(2666);
z(4237) = 0.00893*z(27)*z(2732)*QNE_2p + 0.0005*z(205)*z(4207) + 0.0005*z(206)*z(4208) + 0.0005*z(2734)*z(221) + 0.0005*z(2737)*z(228) + 0.00893*z(28)*z(4209) - 0.0005*z(27)*z(2731)*QNE_2p - 0.00893*z(205)*z(4210) - 0.00893*z(206)*z(4211) - 0.00893*z(2735)*z(221) - 0.00893*z(2738)*z(228) - 0.0005*z(28)*z(4206);
z(4238) = 0.0005*z(27)*z(2730)*QNE_2p + 0.01653*z(27)*z(2732)*QNE_2p + 0.0005*z(28)*z(4213) + 0.01653*z(28)*z(4209) - 0.01653*z(205)*z(4210) - 0.01653*z(206)*z(4211) - 0.01653*z(2735)*z(221) - 0.01653*z(2738)*z(228) - 0.0005*z(205)*z(4214) - 0.0005*z(206)*z(4215) - 0.0005*z(2733)*z(221) - 0.0005*z(2736)*z(228);
z(4239) = 0.00893*z(205)*z(4214) + 0.00893*z(206)*z(4215) + 0.00893*z(2733)*z(221) + 0.00893*z(2736)*z(228) + 0.01653*z(205)*z(4207) + 0.01653*z(206)*z(4208) + 0.01653*z(2734)*z(221) + 0.01653*z(2737)*z(228) - 0.01653*z(27)*z(2731)*QNE_2p - 0.00893*z(27)*z(2730)*QNE_2p - 0.01653*z(28)*z(4206) - 0.00893*z(28)*z(4213);
z(2741) = 0.00893*z(205)*z(2733) + 0.00893*z(206)*z(2736) + 0.01653*z(205)*z(2734) + 0.01653*z(206)*z(2737) - 0.01653*z(28)*z(2731) - 0.00893*z(28)*z(2730);
z(2752) = 0.0005*z(28)*z(2730) + 0.01653*z(28)*z(2732) - 0.01653*z(205)*z(2735) - 0.01653*z(206)*z(2738) - 0.0005*z(205)*z(2733) - 0.0005*z(206)*z(2736);
z(2767) = 0.0005*z(205)*z(2734) + 0.0005*z(206)*z(2737) + 0.00893*z(28)*z(2732) - 0.00893*z(205)*z(2735) - 0.00893*z(206)*z(2738) - 0.0005*z(28)*z(2731);
z(4180) = (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(203)*z(205) + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(204)*z(206) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(201)*z(205) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(202)*z(206) + z(28)*z(235)*z(2610) + z(201)*z(221)*z(2611) + z(202)*z(228)*z(2611) + z(205)*z(217)*z(2611) + z(206)*z(230)*z(2611) + z(27)*z(199)*QNE_2p*z(2610) - (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(28)*z(199) - 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(28)*z(197) - z(28)*z(234)*z(2611) - z(203)*z(221)*z(2610) - z(204)*z(228)*z(2610) - z(205)*z(219)*z(2610) - z(206)*z(229)*z(2610) - z(27)*z(197)*QNE_2p*z(2611);
z(3897) = z(221)*z(1884) - z(228)*z(1883);
z(3898) = z(228)*z(1882) + z(27)*QNE_2p*z(1884);
z(3899) = -z(221)*z(1882) - z(27)*QNE_2p*z(1883);
z(1887) = -z(28)*z(1883) - z(205)*z(1882);
z(1913) = z(205)*z(1884) - z(206)*z(1883);
z(1900) = z(28)*z(1884) + z(206)*z(1882);
z(3900) = z(252)*z(3897) + z(255)*z(3898) + z(258)*z(3899) + z(277)*z(1887) + z(278)*z(1913) + z(279)*z(1900);
z(4038) = z(28)*z(276)*z(2530) + z(205)*z(268)*z(2531) + z(206)*z(271)*z(2531) + z(256)*z(221)*z(2531) + z(259)*z(228)*z(2531) + z(27)*z(254)*QNE_2p*z(2530) - z(28)*z(265)*z(2531) - z(205)*z(274)*z(2530) - z(206)*z(275)*z(2530) - z(257)*z(221)*z(2530) - z(260)*z(228)*z(2530) - z(27)*z(253)*QNE_2p*z(2531);
z(4039) = z(28)*z(265)*HOHK3 + z(205)*z(274)*HOHK2 + z(206)*z(275)*HOHK2 + z(257)*z(221)*HOHK2 + z(260)*z(228)*HOHK2 + z(27)*z(253)*QNE_2p*HOHK3 - z(28)*z(276)*HOHK2 - z(205)*z(268)*HOHK3 - z(206)*z(271)*HOHK3 - z(256)*z(221)*HOHK3 - z(259)*z(228)*HOHK3 - z(27)*z(254)*QNE_2p*HOHK2;
z(4040) = z(3900) + z(4038) + z(4039);
z(3963) = z(253)*z(3897) + z(256)*z(3898) + z(259)*z(3899) + z(265)*z(1913) + z(268)*z(1900) + z(271)*z(1887);
z(4078) = z(28)*z(278)*z(2531) + z(205)*z(274)*z(2529) + z(206)*z(275)*z(2529) + z(257)*z(221)*z(2529) + z(260)*z(228)*z(2529) + z(27)*z(252)*QNE_2p*z(2531) - z(28)*z(276)*z(2529) - z(205)*z(279)*z(2531) - z(206)*z(277)*z(2531) - z(255)*z(221)*z(2531) - z(258)*z(228)*z(2531) - z(27)*z(254)*QNE_2p*z(2529);
z(4079) = z(28)*z(276)*HOHK1 + z(205)*z(279)*HOHK3 + z(206)*z(277)*HOHK3 + z(255)*z(221)*HOHK3 + z(258)*z(228)*HOHK3 + z(27)*z(254)*QNE_2p*HOHK1 - z(28)*z(278)*HOHK3 - z(205)*z(274)*HOHK1 - z(206)*z(275)*HOHK1 - z(257)*z(221)*HOHK1 - z(260)*z(228)*HOHK1 - z(27)*z(252)*QNE_2p*HOHK3;
z(4080) = z(3963) + z(4078) + z(4079);
z(3999) = z(254)*z(3897) + z(257)*z(3898) + z(260)*z(3899) + z(274)*z(1900) + z(275)*z(1887) + z(276)*z(1913);
z(4118) = z(28)*z(265)*z(2529) + z(205)*z(279)*z(2530) + z(206)*z(277)*z(2530) + z(255)*z(221)*z(2530) + z(258)*z(228)*z(2530) + z(27)*z(253)*QNE_2p*z(2529) - z(28)*z(278)*z(2530) - z(205)*z(268)*z(2529) - z(206)*z(271)*z(2529) - z(256)*z(221)*z(2529) - z(259)*z(228)*z(2529) - z(27)*z(252)*QNE_2p*z(2530);
z(4119) = z(28)*z(278)*HOHK2 + z(205)*z(268)*HOHK1 + z(206)*z(271)*HOHK1 + z(256)*z(221)*HOHK1 + z(259)*z(228)*HOHK1 + z(27)*z(252)*QNE_2p*HOHK2 - z(28)*z(265)*HOHK1 - z(205)*z(279)*HOHK2 - z(206)*z(277)*HOHK2 - z(255)*z(221)*HOHK2 - z(258)*z(228)*HOHK2 - z(27)*z(253)*QNE_2p*HOHK1;
z(4120) = z(3999) + z(4118) + z(4119);
z(4152) = z(203)*z(221)*z(2612) + z(204)*z(228)*z(2612) + z(205)*z(219)*z(2612) + z(206)*z(229)*z(2612) - 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(28)^2 - 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(205)^2 - 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(206)^2 - 2*z(205)*z(221)*z(2611) - 2*z(206)*z(228)*z(2611) - z(28)*z(235)*z(2612) - 2*z(27)*z(28)*QNE_2p*z(2611) - z(27)*z(199)*QNE_2p*z(2612);
z(4166) = z(28)*z(234)*z(2612) + 2*z(205)*z(221)*z(2610) + 2*z(206)*z(228)*z(2610) + z(27)*z(197)*QNE_2p*z(2612) + 2*z(27)*z(28)*QNE_2p*z(2610) - (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(28)^2 - (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(205)^2 - (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(206)^2 - z(201)*z(221)*z(2612) - z(202)*z(228)*z(2612) - z(205)*z(217)*z(2612) - z(206)*z(230)*z(2612);
z(2654) = z(28)*z(199)*z(2610) + z(201)*z(205)*z(2611) + z(202)*z(206)*z(2611) - z(28)*z(197)*z(2611) - z(203)*z(205)*z(2610) - z(204)*z(206)*z(2610);
z(2396) = z(252)*z(1913) + z(255)*z(1900) + z(258)*z(1887);
z(2560) = z(28)*z(254)*z(2530) + z(205)*z(256)*z(2531) + z(206)*z(259)*z(2531) - z(28)*z(253)*z(2531) - z(205)*z(257)*z(2530) - z(206)*z(260)*z(2530);
z(2573) = z(2396) + z(2560);
z(2641) = z(28)*z(253)*HOHK3 + z(205)*z(257)*HOHK2 + z(206)*z(260)*HOHK2 - z(28)*z(254)*HOHK2 - z(205)*z(256)*HOHK3 - z(206)*z(259)*HOHK3;
z(2693) = z(2573) + z(2641);
z(2418) = z(253)*z(1913) + z(256)*z(1900) + z(259)*z(1887);

z(2547) = z(28)*z(252)*z(2531) + z(205)*z(257)*z(2529) + z(206)*z(260)*z(2529) - z(28)*z(254)*z(2529) - z(205)*z(255)*z(2531) - z(206)*z(258)*z(2531);
z(2586) = z(2418) + z(2547);
z(2628) = z(28)*z(254)*HOHK1 + z(205)*z(255)*HOHK3 + z(206)*z(258)*HOHK3 - z(28)*z(252)*HOHK3 - z(205)*z(257)*HOHK1 - z(206)*z(260)*HOHK1;
z(2706) = z(2586) + z(2628);
z(2440) = z(254)*z(1913) + z(257)*z(1900) + z(260)*z(1887);
z(2534) = z(28)*z(253)*z(2529) + z(205)*z(255)*z(2530) + z(206)*z(258)*z(2530) - z(28)*z(252)*z(2530) - z(205)*z(256)*z(2529) - z(206)*z(259)*z(2529);
z(2599) = z(2440) + z(2534);
z(2615) = z(28)*z(252)*HOHK2 + z(205)*z(256)*HOHK1 + z(206)*z(259)*HOHK1 - z(28)*z(253)*HOHK1 - z(205)*z(255)*HOHK2 - z(206)*z(258)*HOHK2;
z(2719) = z(2599) + z(2615);
z(2680) = z(203)*z(205)*z(2612) + z(204)*z(206)*z(2612) - z(28)*z(199)*z(2612) - z(28)^2*z(2611) - z(205)^2*z(2611) - z(206)^2*z(2611);
z(2667) = z(28)*z(197)*z(2612) + z(28)^2*z(2610) + z(205)^2*z(2610) + z(206)^2*z(2610) - z(201)*z(205)*z(2612) - z(202)*z(206)*z(2612);
z(4317) = z(347)*z(4237) + z(350)*z(4238) + z(353)*z(4239) + z(2741)*z(365) + z(2752)*z(362) + z(2767)*z(359) + z(344)*z(4180) + z(2779)*z(4040) + z(2782)*z(4080) + z(2785)*z(4120) + z(2788)*z(4152) + z(2791)*z(4166) + z(358)*z(2654) + z(4297)*z(2693) + z(4298)*z(2706) + z(4299)*z(2719) + z(4300)*z(2680) + z(4301)*z(2667);
z(4252) = 0.0005*z(413)*z(4206) + 0.0005*z(423)*z(4207) + 0.0005*z(433)*z(4208) + 0.0005*z(2731)*z(3273) + 0.0005*z(2734)*z(3291) + 0.0005*z(2737)*z(3308) - 0.00893*z(413)*z(4209) - 0.00893*z(423)*z(4210) - 0.00893*z(433)*z(4211) - 0.00893*z(2732)*z(3273) - 0.00893*z(2735)*z(3291) - 0.00893*z(2738)*z(3308);
z(4253) = -0.01653*z(413)*z(4209) - 0.01653*z(423)*z(4210) - 0.01653*z(433)*z(4211) - 0.01653*z(2732)*z(3273) - 0.01653*z(2735)*z(3291) - 0.01653*z(2738)*z(3308) - 0.0005*z(413)*z(4213) - 0.0005*z(423)*z(4214) - 0.0005*z(433)*z(4215) - 0.0005*z(2730)*z(3273) - 0.0005*z(2733)*z(3291) - 0.0005*z(2736)*z(3308);
z(4254) = 0.00893*z(413)*z(4213) + 0.00893*z(423)*z(4214) + 0.00893*z(433)*z(4215) + 0.00893*z(2730)*z(3273) + 0.00893*z(2733)*z(3291) + 0.00893*z(2736)*z(3308) + 0.01653*z(413)*z(4206) + 0.01653*z(423)*z(4207) + 0.01653*z(433)*z(4208) + 0.01653*z(2731)*z(3273) + 0.01653*z(2734)*z(3291) + 0.01653*z(2737)*z(3308);
z(2742) = 0.00893*z(413)*z(2730) + 0.00893*z(423)*z(2733) + 0.00893*z(433)*z(2736) + 0.01653*z(413)*z(2731) + 0.01653*z(423)*z(2734) + 0.01653*z(433)*z(2737);
z(2755) = -0.01653*z(413)*z(2732) - 0.01653*z(423)*z(2735) - 0.01653*z(433)*z(2738) - 0.0005*z(413)*z(2730) - 0.0005*z(423)*z(2733) - 0.0005*z(433)*z(2736);
z(2768) = 0.0005*z(413)*z(2731) + 0.0005*z(423)*z(2734) + 0.0005*z(433)*z(2737) - 0.00893*z(413)*z(2732) - 0.00893*z(423)*z(2735) - 0.00893*z(433)*z(2738);
z(4181) = (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(199)*z(413) + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(203)*z(423) + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(204)*z(433) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(197)*z(413) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(201)*z(423) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(202)*z(433) + z(197)*z(3273)*z(2611) + z(201)*z(3291)*z(2611) + z(202)*z(3308)*z(2611) + z(413)*z(234)*z(2611) + z(423)*z(217)*z(2611) + z(433)*z(230)*z(2611) - z(199)*z(3273)*z(2610) - z(203)*z(3291)*z(2610) - z(204)*z(3308)*z(2610) - z(413)*z(235)*z(2610) - z(423)*z(219)*z(2610) - z(433)*z(229)*z(2610);
z(1251) = -z(193)*z(1228) - z(195)*z(1227);
z(1239) = z(195)*z(1226) - z(19)*z(1228);
z(1229) = z(19)*z(1227) + z(193)*z(1226);
z(1805) = z(209)*z(1251) + z(212)*z(1239) + z(215)*z(1229);
z(1888) = z(413)*z(1883) - z(423)*z(1882);
z(1946) = z(1805) + z(1888);
z(1773) = z(207)*z(1251) + z(210)*z(1239) + z(213)*z(1229);
z(1914) = z(423)*z(1884) - z(433)*z(1883);
z(1926) = z(1773) + z(1914);
z(1789) = z(208)*z(1251) + z(211)*z(1239) + z(214)*z(1229);
z(1901) = z(433)*z(1882) - z(413)*z(1884);
z(1936) = z(1789) + z(1901);
z(3771) = -z(223)*z(1228) - z(225)*z(1227);
z(3772) = z(225)*z(1226) + z(20)*QND_3p*z(1228);
z(3773) = z(223)*z(1226) - z(20)*QND_3p*z(1227);
z(3774) = z(207)*z(3771) + z(210)*z(3772) + z(213)*z(3773) + z(236)*z(1229) + z(237)*z(1251) + z(238)*z(1239);
z(3912) = z(3291)*z(1884) - z(3308)*z(1883);
z(3822) = z(208)*z(3771) + z(211)*z(3772) + z(214)*z(3773) + z(220)*z(1251) + z(224)*z(1239) + z(227)*z(1229);
z(3913) = z(3308)*z(1882) - z(3273)*z(1884);
z(3852) = z(209)*z(3771) + z(212)*z(3772) + z(215)*z(3773) + z(231)*z(1239) + z(232)*z(1229) + z(233)*z(1251);
z(3914) = z(3273)*z(1883) - z(3291)*z(1882);
z(3915) = z(277)*z(1946) + z(278)*z(1926) + z(279)*z(1936) + z(252)*(z(3774)+z(3912)) + z(255)*(z(3822)+z(3913)) + z(258)*(z(3852)+z(3914));
z(4047) = z(253)*z(3273)*z(2531) + z(256)*z(3291)*z(2531) + z(259)*z(3308)*z(2531) + z(413)*z(265)*z(2531) + z(423)*z(268)*z(2531) + z(433)*z(271)*z(2531) - z(254)*z(3273)*z(2530) - z(257)*z(3291)*z(2530) - z(260)*z(3308)*z(2530) - z(413)*z(276)*z(2530) - z(423)*z(274)*z(2530) - z(433)*z(275)*z(2530);
z(4048) = z(254)*z(3273)*HOHK2 + z(257)*z(3291)*HOHK2 + z(260)*z(3308)*HOHK2 + z(413)*z(276)*HOHK2 + z(423)*z(274)*HOHK2 + z(433)*z(275)*HOHK2 - z(253)*z(3273)*HOHK3 - z(256)*z(3291)*HOHK3 - z(259)*z(3308)*HOHK3 - z(413)*z(265)*HOHK3 - z(423)*z(268)*HOHK3 - z(433)*z(271)*HOHK3;
z(4049) = z(3915) + z(4047) + z(4048);
z(3969) = z(265)*z(1926) + z(268)*z(1936) + z(271)*z(1946) + z(253)*(z(3774)+z(3912)) + z(256)*(z(3822)+z(3913)) + z(259)*(z(3852)+z(3914));
z(4087) = z(254)*z(3273)*z(2529) + z(257)*z(3291)*z(2529) + z(260)*z(3308)*z(2529) + z(413)*z(276)*z(2529) + z(423)*z(274)*z(2529) + z(433)*z(275)*z(2529) - z(252)*z(3273)*z(2531) - z(255)*z(3291)*z(2531) - z(258)*z(3308)*z(2531) - z(413)*z(278)*z(2531) - z(423)*z(279)*z(2531) - z(433)*z(277)*z(2531);
z(4088) = z(252)*z(3273)*HOHK3 + z(255)*z(3291)*HOHK3 + z(258)*z(3308)*HOHK3 + z(413)*z(278)*HOHK3 + z(423)*z(279)*HOHK3 + z(433)*z(277)*HOHK3 - z(254)*z(3273)*HOHK1 - z(257)*z(3291)*HOHK1 - z(260)*z(3308)*HOHK1 - z(413)*z(276)*HOHK1 - z(423)*z(274)*HOHK1 - z(433)*z(275)*HOHK1;
z(4089) = z(3969) + z(4087) + z(4088);
z(4005) = z(274)*z(1936) + z(275)*z(1946) + z(276)*z(1926) + z(254)*(z(3774)+z(3912)) + z(257)*(z(3822)+z(3913)) + z(260)*(z(3852)+z(3914));

z(4127) = z(252)*z(3273)*z(2530) + z(255)*z(3291)*z(2530) + z(258)*z(3308)*z(2530) + z(413)*z(278)*z(2530) + z(423)*z(279)*z(2530) + z(433)*z(277)*z(2530) - z(253)*z(3273)*z(2529) - z(256)*z(3291)*z(2529) - z(259)*z(3308)*z(2529) - z(413)*z(265)*z(2529) - z(423)*z(268)*z(2529) - z(433)*z(271)*z(2529);
z(4128) = z(253)*z(3273)*HOHK1 + z(256)*z(3291)*HOHK1 + z(259)*z(3308)*HOHK1 + z(413)*z(265)*HOHK1 + z(423)*z(268)*HOHK1 + z(433)*z(271)*HOHK1 - z(252)*z(3273)*HOHK2 - z(255)*z(3291)*HOHK2 - z(258)*z(3308)*HOHK2 - z(413)*z(278)*HOHK2 - z(423)*z(279)*HOHK2 - z(433)*z(277)*HOHK2;
z(4129) = z(4005) + z(4127) + z(4128);
z(4153) = 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(28)*z(413) + z(28)*z(3273)*z(2611) + z(199)*z(3273)*z(2612) + z(203)*z(3291)*z(2612) + z(204)*z(3308)*z(2612) + z(413)*z(235)*z(2612) + z(423)*z(219)*z(2612) + z(433)*z(229)*z(2612) + z(27)*z(413)*QNE_2p*z(2611) - 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(205)*z(423) - 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(206)*z(433) - z(205)*z(3291)*z(2611) - z(206)*z(3308)*z(2611) - z(423)*z(221)*z(2611) - z(433)*z(228)*z(2611);
z(4167) = (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(28)*z(413) + z(205)*z(3291)*z(2610) + z(206)*z(3308)*z(2610) + z(423)*z(221)*z(2610) + z(433)*z(228)*z(2610) - (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(205)*z(423) - (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(206)*z(433) - z(28)*z(3273)*z(2610) - z(197)*z(3273)*z(2612) - z(201)*z(3291)*z(2612) - z(202)*z(3308)*z(2612) - z(413)*z(234)*z(2612) - z(423)*z(217)*z(2612) - z(433)*z(230)*z(2612) - z(27)*z(413)*QNE_2p*z(2610);
z(2655) = z(197)*z(413)*z(2611) + z(201)*z(423)*z(2611) + z(202)*z(433)*z(2611) - z(199)*z(413)*z(2610) - z(203)*z(423)*z(2610) - z(204)*z(433)*z(2610);
z(2399) = z(252)*z(1926) + z(255)*z(1936) + z(258)*z(1946);
z(2561) = z(253)*z(413)*z(2531) + z(256)*z(423)*z(2531) + z(259)*z(433)*z(2531) - z(254)*z(413)*z(2530) - z(257)*z(423)*z(2530) - z(260)*z(433)*z(2530);
z(2576) = z(2399) + z(2561);
z(2642) = z(254)*z(413)*HOHK2 + z(257)*z(423)*HOHK2 + z(260)*z(433)*HOHK2 - z(253)*z(413)*HOHK3 - z(256)*z(423)*HOHK3 - z(259)*z(433)*HOHK3;
z(2696) = z(2576) + z(2642);
z(2421) = z(253)*z(1926) + z(256)*z(1936) + z(259)*z(1946);
z(2548) = z(254)*z(413)*z(2529) + z(257)*z(423)*z(2529) + z(260)*z(433)*z(2529) - z(252)*z(413)*z(2531) - z(255)*z(423)*z(2531) - z(258)*z(433)*z(2531);
z(2589) = z(2421) + z(2548);
z(2629) = z(252)*z(413)*HOHK3 + z(255)*z(423)*HOHK3 + z(258)*z(433)*HOHK3 - z(254)*z(413)*HOHK1 - z(257)*z(423)*HOHK1 - z(260)*z(433)*HOHK1;
z(2709) = z(2589) + z(2629);
z(2443) = z(254)*z(1926) + z(257)*z(1936) + z(260)*z(1946);
z(2535) = z(252)*z(413)*z(2530) + z(255)*z(423)*z(2530) + z(258)*z(433)*z(2530) - z(253)*z(413)*z(2529) - z(256)*z(423)*z(2529) - z(259)*z(433)*z(2529);
z(2602) = z(2443) + z(2535);
z(2616) = z(253)*z(413)*HOHK1 + z(256)*z(423)*HOHK1 + z(259)*z(433)*HOHK1 - z(252)*z(413)*HOHK2 - z(255)*z(423)*HOHK2 - z(258)*z(433)*HOHK2;
z(2722) = z(2602) + z(2616);
z(2681) = z(28)*z(413)*z(2611) + z(199)*z(413)*z(2612) + z(203)*z(423)*z(2612) + z(204)*z(433)*z(2612) - z(205)*z(423)*z(2611) - z(206)*z(433)*z(2611);
z(2668) = z(205)*z(423)*z(2610) + z(206)*z(433)*z(2610) - z(28)*z(413)*z(2610) - z(197)*z(413)*z(2612) - z(201)*z(423)*z(2612) - z(202)*z(433)*z(2612);
z(4318) = z(347)*z(4252) + z(350)*z(4253) + z(353)*z(4254) + z(2742)*z(365) + z(2755)*z(362) + z(2768)*z(359) + z(344)*z(4181) + z(2779)*z(4049) + z(2782)*z(4089) + z(2785)*z(4129) + z(2788)*z(4153) + z(2791)*z(4167) + z(358)*z(2655) + z(4297)*z(2696) + z(4298)*z(2709) + z(4299)*z(2722) + z(4300)*z(2681) + z(4301)*z(2668);
z(4257) = 0.0005*z(414)*z(4206) + 0.0005*z(424)*z(4207) + 0.0005*z(434)*z(4208) + 0.0005*z(2731)*z(3274) + 0.0005*z(2734)*z(3292) + 0.0005*z(2737)*z(3309) - 0.00893*z(414)*z(4209) - 0.00893*z(424)*z(4210) - 0.00893*z(434)*z(4211) - 0.00893*z(2732)*z(3274) - 0.00893*z(2735)*z(3292) - 0.00893*z(2738)*z(3309);
z(4258) = -0.01653*z(414)*z(4209) - 0.01653*z(424)*z(4210) - 0.01653*z(434)*z(4211) - 0.01653*z(2732)*z(3274) - 0.01653*z(2735)*z(3292) - 0.01653*z(2738)*z(3309) - 0.0005*z(414)*z(4213) - 0.0005*z(424)*z(4214) - 0.0005*z(434)*z(4215) - 0.0005*z(2730)*z(3274) - 0.0005*z(2733)*z(3292) - 0.0005*z(2736)*z(3309);
z(4259) = 0.00893*z(414)*z(4213) + 0.00893*z(424)*z(4214) + 0.00893*z(434)*z(4215) + 0.00893*z(2730)*z(3274) + 0.00893*z(2733)*z(3292) + 0.00893*z(2736)*z(3309) + 0.01653*z(414)*z(4206) + 0.01653*z(424)*z(4207) + 0.01653*z(434)*z(4208) + 0.01653*z(2731)*z(3274) + 0.01653*z(2734)*z(3292) + 0.01653*z(2737)*z(3309);
z(2743) = 0.00893*z(414)*z(2730) + 0.00893*z(424)*z(2733) + 0.00893*z(434)*z(2736) + 0.01653*z(414)*z(2731) + 0.01653*z(424)*z(2734) + 0.01653*z(434)*z(2737);
z(2756) = -0.01653*z(414)*z(2732) - 0.01653*z(424)*z(2735) - 0.01653*z(434)*z(2738) - 0.0005*z(414)*z(2730) - 0.0005*z(424)*z(2733) - 0.0005*z(434)*z(2736);
z(2769) = 0.0005*z(414)*z(2731) + 0.0005*z(424)*z(2734) + 0.0005*z(434)*z(2737) - 0.00893*z(414)*z(2732) - 0.00893*z(424)*z(2735) - 0.00893*z(434)*z(2738);
z(4182) = (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(199)*z(414) + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(203)*z(424) + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(204)*z(434) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(197)*z(414) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(201)*z(424) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(202)*z(434) + z(197)*z(3274)*z(2611) + z(201)*z(3292)*z(2611) + z(202)*z(3309)*z(2611) + z(414)*z(234)*z(2611) + z(424)*z(217)*z(2611) + z(434)*z(230)*z(2611) - z(199)*z(3274)*z(2610) - z(203)*z(3292)*z(2610) - z(204)*z(3309)*z(2610) - z(414)*z(235)*z(2610) - z(424)*z(219)*z(2610) - z(434)*z(229)*z(2610);
z(1249) = z(194)*z(1228) + z(196)*z(1227);
z(1240) = -z(20)*z(1228) - z(196)*z(1226);
z(1230) = z(20)*z(1227) - z(194)*z(1226);
z(1803) = z(209)*z(1249) + z(212)*z(1240) + z(215)*z(1230);
z(1889) = z(414)*z(1883) - z(424)*z(1882);
z(1944) = z(1803) + z(1889);
z(1771) = z(207)*z(1249) + z(210)*z(1240) + z(213)*z(1230);
z(1915) = z(424)*z(1884) - z(434)*z(1883);
z(1924) = z(1771) + z(1915);

```
z(1787) = z(208)*z(1249) + z(211)*z(1240) + z(214)*z(1230);
z(1902) = z(434)*z(1882) - z(414)*z(1884);
z(1934) = z(1787) + z(1902);
z(3761) = z(222)*z(1228) + z(226)*z(1227);
z(3762) = -z(226)*z(1226) - z(19)*QND_3p*z(1228);
z(3763) = z(19)*QND_3p*z(1227) - z(222)*z(1226);
z(3764) = z(207)*z(3761) + z(210)*z(3762) + z(213)*z(3763) + z(236)*z(1230) + z(237)*z(1249) + z(238)*z(1240);
z(3902) = z(3292)*z(1884) - z(3309)*z(1883);
z(3818) = z(208)*z(3761) + z(211)*z(3762) + z(214)*z(3763) + z(220)*z(1249) + z(224)*z(1240) + z(227)*z(1230);
z(3903) = z(3309)*z(1882) - z(3274)*z(1884);
z(3848) = z(209)*z(3761) + z(212)*z(3762) + z(215)*z(3763) + z(231)*z(1240) + z(232)*z(1230) + z(233)*z(1249);
z(3904) = z(3274)*z(1883) - z(3292)*z(1882);
z(3905) = z(277)*z(1944) + z(278)*z(1924) + z(279)*z(1934) + z(252)*(z(3764)+z(3902)) + z(255)*(z(3818)+z(3903)) + z(258)*(z(3848)+z(3904));
z(4041) = z(253)*z(3274)*z(2531) + z(256)*z(3292)*z(2531) + z(259)*z(3309)*z(2531) + z(414)*z(265)*z(2531) + z(424)*z(268)*z(2531) +
z(434)*z(271)*z(2531) - z(254)*z(3274)*z(2530) - z(257)*z(3292)*z(2530) - z(260)*z(3309)*z(2530) - z(414)*z(276)*z(2530) - z(424)*z(274)*z(2530)
- z(434)*z(275)*z(2530);
z(4042) = z(254)*z(3274)*HOHK2 + z(257)*z(3292)*HOHK2 + z(260)*z(3309)*HOHK2 + z(414)*z(276)*HOHK2 + z(424)*z(274)*HOHK2 +
z(434)*z(275)*HOHK2 - z(253)*z(3274)*HOHK3 - z(256)*z(3292)*HOHK3 - z(259)*z(3309)*HOHK3 - z(414)*z(265)*HOHK3 -
z(424)*z(268)*HOHK3 - z(434)*z(271)*HOHK3;
z(4043) = z(3905) + z(4041) + z(4042);
z(3965) = z(265)*z(1924) + z(268)*z(1934) + z(271)*z(1944) + z(253)*(z(3764)+z(3902)) + z(256)*(z(3818)+z(3903)) + z(259)*(z(3848)+z(3904));
z(4081) = z(254)*z(3274)*z(2529) + z(257)*z(3292)*z(2529) + z(260)*z(3309)*z(2529) + z(414)*z(276)*z(2529) + z(424)*z(274)*z(2529) +
z(434)*z(275)*z(2529) - z(252)*z(3274)*z(2531) - z(255)*z(3292)*z(2531) - z(258)*z(3309)*z(2531) - z(414)*z(278)*z(2531) - z(424)*z(279)*z(2531)
- z(434)*z(277)*z(2531);
z(4082) = z(252)*z(3274)*HOHK3 + z(255)*z(3292)*HOHK3 + z(258)*z(3309)*HOHK3 + z(414)*z(278)*HOHK3 + z(424)*z(279)*HOHK3 +
z(434)*z(277)*HOHK3 - z(254)*z(3274)*HOHK1 - z(257)*z(3292)*HOHK1 - z(260)*z(3309)*HOHK1 - z(414)*z(276)*HOHK1 -
z(424)*z(274)*HOHK1 - z(434)*z(275)*HOHK1;
z(4083) = z(3965) + z(4081) + z(4082);
z(4001) = z(274)*z(1934) + z(275)*z(1944) + z(276)*z(1924) + z(254)*(z(3764)+z(3902)) + z(257)*(z(3818)+z(3903)) + z(260)*(z(3848)+z(3904));
z(4121) = z(252)*z(3274)*z(2530) + z(255)*z(3292)*z(2530) + z(258)*z(3309)*z(2530) + z(414)*z(278)*z(2530) + z(424)*z(279)*z(2530) +
z(434)*z(277)*z(2530) - z(253)*z(3274)*z(2529) - z(256)*z(3292)*z(2529) - z(259)*z(3309)*z(2529) - z(414)*z(265)*z(2529) - z(424)*z(268)*z(2529)
- z(434)*z(271)*z(2529);
z(4122) = z(253)*z(3274)*HOHK1 + z(256)*z(3292)*HOHK1 + z(259)*z(3309)*HOHK1 + z(414)*z(265)*HOHK1 + z(424)*z(268)*HOHK1 +
z(434)*z(271)*HOHK1 - z(252)*z(3274)*HOHK2 - z(255)*z(3292)*HOHK2 - z(258)*z(3309)*HOHK2 - z(414)*z(278)*HOHK2 -
z(424)*z(279)*HOHK2 - z(434)*z(277)*HOHK2;
z(4123) = z(4001) + z(4121) + z(4122);
z(4154) = 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(28)*z(414) + z(28)*z(3274)*z(2611) +
z(199)*z(3274)*z(2612) + z(203)*z(3292)*z(2612) + z(204)*z(3309)*z(2612) + z(414)*z(235)*z(2612) + z(424)*z(219)*z(2612) +
z(434)*z(229)*z(2612) + z(27)*z(414)*QNE_2p*z(2611) - 4.049999999999887E-05*(-1+8.854320987654567*T-
1.955555555555609*T^2)*z(205)*z(424) - 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(206)*z(434) -
z(205)*z(3292)*z(2611) - z(206)*z(3309)*z(2611) - z(424)*z(221)*z(2611) - z(434)*z(228)*z(2611);
z(4168) = (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(28)*z(414) + z(205)*z(3292)*z(2610) + z(206)*z(3309)*z(2610) +
z(424)*z(221)*z(2610) + z(434)*z(228)*z(2610) - (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(205)*z(424) -
(0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(206)*z(434) - z(28)*z(3274)*z(2610) - z(197)*z(3274)*z(2612) -
z(201)*z(3292)*z(2612) - z(202)*z(3309)*z(2612) - z(414)*z(234)*z(2612) - z(424)*z(217)*z(2612) - z(434)*z(230)*z(2612) -
z(27)*z(414)*QNE_2p*z(2610);
z(2656) = z(197)*z(414)*z(2611) + z(201)*z(424)*z(2611) + z(202)*z(434)*z(2611) - z(199)*z(414)*z(2610) - z(203)*z(424)*z(2610) -
z(204)*z(434)*z(2610);
z(2397) = z(252)*z(1924) + z(255)*z(1934) + z(258)*z(1944);
z(2562) = z(253)*z(414)*z(2531) + z(256)*z(424)*z(2531) + z(259)*z(434)*z(2531) - z(254)*z(414)*z(2530) - z(257)*z(424)*z(2530) -
z(260)*z(434)*z(2530);
z(2574) = z(2397) + z(2562);
z(2643) = z(254)*z(414)*HOHK2 + z(257)*z(424)*HOHK2 + z(260)*z(434)*HOHK2 - z(253)*z(414)*HOHK3 - z(256)*z(424)*HOHK3 -
z(259)*z(434)*HOHK3;
z(2694) = z(2574) + z(2643);
z(2419) = z(253)*z(1924) + z(256)*z(1934) + z(259)*z(1944);
z(2549) = z(254)*z(414)*z(2529) + z(257)*z(424)*z(2529) + z(260)*z(434)*z(2529) - z(252)*z(414)*z(2531) - z(255)*z(424)*z(2531) -
z(258)*z(434)*z(2531);
z(2587) = z(2419) + z(2549);
z(2630) = z(252)*z(414)*HOHK3 + z(255)*z(424)*HOHK3 + z(258)*z(434)*HOHK3 - z(254)*z(414)*HOHK1 - z(257)*z(424)*HOHK1 -
z(260)*z(434)*HOHK1;
z(2707) = z(2587) + z(2630);
z(2441) = z(254)*z(1924) + z(257)*z(1934) + z(260)*z(1944);
z(2536) = z(252)*z(414)*z(2530) + z(255)*z(424)*z(2530) + z(258)*z(434)*z(2530) - z(253)*z(414)*z(2529) - z(256)*z(424)*z(2529) -
z(259)*z(434)*z(2529);
z(2600) = z(2441) + z(2536);
z(2617) = z(253)*z(414)*HOHK1 + z(256)*z(424)*HOHK1 + z(259)*z(434)*HOHK1 - z(252)*z(414)*HOHK2 - z(255)*z(424)*HOHK2 -
z(258)*z(434)*HOHK2;
z(2720) = z(2600) + z(2617);
z(2682) = z(28)*z(414)*z(2611) + z(199)*z(414)*z(2612) + z(203)*z(424)*z(2612) + z(204)*z(434)*z(2612) - z(205)*z(424)*z(2611) -
z(206)*z(434)*z(2611);
z(2669) = z(205)*z(424)*z(2610) + z(206)*z(434)*z(2610) - z(28)*z(414)*z(2610) - z(197)*z(414)*z(2612) - z(201)*z(424)*z(2612) -
z(202)*z(434)*z(2612);
z(4319) = z(347)*z(4257) + z(350)*z(4258) + z(353)*z(4259) + z(2743)*z(365) + z(2756)*z(362) + z(2769)*z(359) + z(344)*z(4182) +
z(2779)*z(4043) + z(2782)*z(4083) + z(2785)*z(4123) + z(2788)*z(4154) + z(2791)*z(4168) + z(358)*z(2656) + z(4297)*z(2694) + z(4298)*z(2707) +
z(4299)*z(2720) + z(4300)*z(2682) + z(4301)*z(2669);
```

z(4262) = 0.0005*z(415)*z(4206) + 0.0005*z(425)*z(4207) + 0.0005*z(435)*z(4208) + 0.0005*z(2731)*z(3275) + 0.0005*z(2734)*z(3293) + 0.0005*z(2737)*z(3310) - 0.00893*z(415)*z(4209) - 0.00893*z(425)*z(4210) - 0.00893*z(435)*z(4211) - 0.00893*z(2732)*z(3275) - 0.00893*z(2735)*z(3293) - 0.00893*z(2738)*z(3310);
z(4263) = -0.01653*z(415)*z(4209) - 0.01653*z(425)*z(4210) - 0.01653*z(435)*z(4211) - 0.01653*z(2732)*z(3275) - 0.01653*z(2735)*z(3293) - 0.01653*z(2738)*z(3310) - 0.0005*z(415)*z(4213) - 0.0005*z(425)*z(4214) - 0.0005*z(435)*z(4215) - 0.0005*z(2730)*z(3275) - 0.0005*z(2733)*z(3293) - 0.0005*z(2736)*z(3310);
z(4264) = 0.00893*z(415)*z(4213) + 0.00893*z(425)*z(4214) + 0.00893*z(435)*z(4215) + 0.00893*z(2730)*z(3275) + 0.00893*z(2733)*z(3293) + 0.00893*z(2736)*z(3310) + 0.01653*z(415)*z(4206) + 0.01653*z(425)*z(4207) + 0.01653*z(435)*z(4208) + 0.01653*z(2731)*z(3275) + 0.01653*z(2734)*z(3293) + 0.01653*z(2737)*z(3310);
z(2744) = 0.00893*z(415)*z(2730) + 0.00893*z(425)*z(2733) + 0.00893*z(435)*z(2736) + 0.01653*z(415)*z(2731) + 0.01653*z(425)*z(2734) + 0.01653*z(435)*z(2737);
z(2757) = -0.01653*z(415)*z(2732) - 0.01653*z(425)*z(2735) - 0.01653*z(435)*z(2738) - 0.0005*z(415)*z(2730) - 0.0005*z(425)*z(2733) - 0.0005*z(435)*z(2736);
z(2770) = 0.0005*z(415)*z(2731) + 0.0005*z(425)*z(2734) + 0.0005*z(435)*z(2737) - 0.00893*z(415)*z(2732) - 0.00893*z(425)*z(2735) - 0.00893*z(435)*z(2738);
z(4183) = (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(199)*z(415) + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(203)*z(425) + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(204)*z(435) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(197)*z(415) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(201)*z(425) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(202)*z(435) + z(197)*z(3275)*z(2611) + z(201)*z(3293)*z(2611) + z(202)*z(3310)*z(2611) + z(415)*z(234)*z(2611) + z(425)*z(217)*z(2611) + z(435)*z(230)*z(2611) - z(199)*z(3275)*z(2610) - z(203)*z(3293)*z(2610) - z(204)*z(3310)*z(2610) - z(415)*z(235)*z(2610) - z(425)*z(219)*z(2610) - z(435)*z(229)*z(2610);
z(803) = z(123)*z(788) - z(124)*z(787);
z(796) = z(124)*z(786) - z(119)*z(788);
z(789) = z(119)*z(787) - z(123)*z(786);
z(1148) = z(166)*z(803) + z(168)*z(796) + z(170)*z(789);
z(1252) = z(399)*z(1228) - z(406)*z(1227);
z(1259) = z(1148) + z(1252);
z(1158) = z(172)*z(803) + z(174)*z(796) + z(176)*z(789);
z(1241) = z(406)*z(1226) - z(392)*z(1228);
z(1266) = z(1158) + z(1241);
z(1168) = z(173)*z(803) + z(175)*z(796) + z(177)*z(789);
z(1231) = z(392)*z(1227) - z(399)*z(1226);
z(1273) = z(1168) + z(1231);
z(1806) = z(209)*z(1259) + z(212)*z(1266) + z(215)*z(1273);
z(1890) = z(415)*z(1883) - z(425)*z(1882);
z(1947) = z(1806) + z(1890);
z(1774) = z(207)*z(1259) + z(210)*z(1266) + z(213)*z(1273);
z(1916) = z(425)*z(1884) - z(435)*z(1883);
z(1927) = z(1774) + z(1916);
z(1790) = z(208)*z(1259) + z(211)*z(1266) + z(214)*z(1273);
z(1903) = z(435)*z(1882) - z(415)*z(1884);
z(1937) = z(1790) + z(1903);
z(3668) = z(140)*z(788) - z(153)*z(787);
z(3669) = z(153)*z(786) - z(158)*z(788);
z(3670) = z(158)*z(787) - z(140)*z(786);
z(3671) = z(166)*z(3668) + z(168)*z(3669) + z(170)*z(3670) + z(187)*z(789) + z(188)*z(803) + z(189)*z(796);
z(3776) = z(3228)*z(1228) - z(3240)*z(1227);
z(3706) = z(172)*z(3668) + z(174)*z(3669) + z(176)*z(3670) + z(179)*z(803) + z(181)*z(796) + z(183)*z(789);
z(3777) = z(3240)*z(1226) - z(3216)*z(1228);
z(3728) = z(173)*z(3668) + z(175)*z(3669) + z(177)*z(3670) + z(184)*z(796) + z(185)*z(789) + z(186)*z(803);
z(3778) = z(3216)*z(1227) - z(3228)*z(1226);
z(3779) = z(236)*z(1273) + z(237)*z(1259) + z(238)*z(1266) + z(207)*(z(3671)+z(3776)) + z(210)*(z(3706)+z(3777)) + z(213)*(z(3728)+z(3778));
z(3917) = z(3293)*z(1884) - z(3310)*z(1883);
z(3824) = z(220)*z(1259) + z(224)*z(1266) + z(227)*z(1273) + z(208)*(z(3671)+z(3776)) + z(211)*(z(3706)+z(3777)) + z(214)*(z(3728)+z(3778));
z(3918) = z(3310)*z(1882) - z(3275)*z(1884);
z(3854) = z(231)*z(1266) + z(232)*z(1273) + z(233)*z(1259) + z(209)*(z(3671)+z(3776)) + z(212)*(z(3706)+z(3777)) + z(215)*(z(3728)+z(3778));
z(3919) = z(3275)*z(1883) - z(3293)*z(1882);
z(3920) = z(277)*z(1947) + z(278)*z(1927) + z(279)*z(1937) + z(252)*(z(3779)+z(3917)) + z(255)*(z(3824)+z(3918)) + z(258)*(z(3854)+z(3919));
z(4050) = z(253)*z(3275)*z(2531) + z(256)*z(3293)*z(2531) + z(259)*z(3310)*z(2531) + z(415)*z(265)*z(2531) + z(425)*z(268)*z(2531) + z(435)*z(271)*z(2531) - z(254)*z(3275)*z(2530) - z(257)*z(3293)*z(2530) - z(260)*z(3310)*z(2530) - z(415)*z(276)*z(2530) - z(425)*z(274)*z(2530) - z(435)*z(275)*z(2530);
z(4051) = z(254)*z(3275)*HOHK2 + z(257)*z(3293)*HOHK2 + z(260)*z(3310)*HOHK2 + z(415)*z(276)*HOHK2 + z(425)*z(274)*HOHK2 + z(435)*z(275)*HOHK2 - z(253)*z(3275)*HOHK3 - z(256)*z(3293)*HOHK3 - z(259)*z(3310)*HOHK3 - z(415)*z(265)*HOHK3 - z(425)*z(268)*HOHK3 - z(435)*z(271)*HOHK3;
z(4052) = z(3920) + z(4050) + z(4051);
z(3971) = z(265)*z(1927) + z(268)*z(1937) + z(271)*z(1947) + z(253)*(z(3779)+z(3917)) + z(256)*(z(3824)+z(3918)) + z(259)*(z(3854)+z(3919));
z(4090) = z(254)*z(3275)*z(2529) + z(257)*z(3293)*z(2529) + z(260)*z(3310)*z(2529) + z(415)*z(276)*z(2529) + z(425)*z(274)*z(2529) + z(435)*z(275)*z(2529) - z(252)*z(3275)*z(2531) - z(255)*z(3293)*z(2531) - z(258)*z(3310)*z(2531) - z(415)*z(278)*z(2531) - z(425)*z(279)*z(2531) - z(435)*z(277)*z(2531);
z(4091) = z(252)*z(3275)*HOHK3 + z(255)*z(3293)*HOHK3 + z(258)*z(3310)*HOHK3 + z(415)*z(278)*HOHK3 + z(425)*z(279)*HOHK3 + z(435)*z(277)*HOHK3 - z(254)*z(3275)*HOHK1 - z(257)*z(3293)*HOHK1 - z(260)*z(3310)*HOHK1 - z(415)*z(276)*HOHK1 - z(425)*z(274)*HOHK1 - z(435)*z(275)*HOHK1;
z(4092) = z(3971) + z(4090) + z(4091);
z(4007) = z(274)*z(1937) + z(275)*z(1947) + z(276)*z(1927) + z(254)*(z(3779)+z(3917)) + z(257)*(z(3824)+z(3918)) + z(260)*(z(3854)+z(3919));
z(4130) = z(252)*z(3275)*z(2530) + z(255)*z(3293)*z(2530) + z(258)*z(3310)*z(2530) + z(415)*z(278)*z(2530) + z(425)*z(279)*z(2530) + z(435)*z(277)*z(2530) - z(253)*z(3275)*z(2529) - z(256)*z(3293)*z(2529) - z(259)*z(3310)*z(2529) - z(415)*z(265)*z(2529) - z(425)*z(268)*z(2529) - z(435)*z(271)*z(2529);

z(4131) = z(253)*z(3275)*HOHK1 + z(256)*z(3293)*HOHK1 + z(259)*z(3310)*HOHK1 + z(415)*z(265)*HOHK1 + z(425)*z(268)*HOHK1 +
z(435)*z(271)*HOHK1 - z(252)*z(3275)*HOHK2 - z(255)*z(3293)*HOHK2 - z(258)*z(3310)*HOHK2 - z(415)*z(278)*HOHK2 -
z(425)*z(279)*HOHK2 - z(435)*z(277)*HOHK2;
z(4132) = z(4007) + z(4130) + z(4131);
z(4155) = 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(28)*z(415) + z(28)*z(3275)*z(2611) +
z(199)*z(3275)*z(2612) + z(203)*z(3293)*z(2612) + z(204)*z(3310)*z(2612) + z(415)*z(235)*z(2612) + z(425)*z(219)*z(2612) +
z(435)*z(229)*z(2612) + z(27)*z(415)*QNE_2p*z(2611) - 4.049999999999887E-05*(-1+8.854320987654567*T-
1.955555555555609*T^2)*z(205)*z(425) - 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(206)*z(435) -
z(205)*z(3293)*z(2611) - z(206)*z(3310)*z(2611) - z(425)*z(221)*z(2611) - z(435)*z(228)*z(2611);
z(4169) = (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(28)*z(415) + z(205)*z(3293)*z(2610) + z(206)*z(3310)*z(2610) +
z(425)*z(221)*z(2610) + z(435)*z(228)*z(2610) - (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(205)*z(425) -
(0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(206)*z(435) - z(28)*z(3275)*z(2610) - z(197)*z(3275)*z(2612) -
z(201)*z(3293)*z(2612) - z(202)*z(3310)*z(2612) - z(415)*z(234)*z(2612) - z(425)*z(217)*z(2612) - z(435)*z(230)*z(2612) -
z(27)*z(415)*QNE_2p*z(2610);
z(2657) = z(197)*z(415)*z(2611) + z(201)*z(425)*z(2611) + z(202)*z(435)*z(2611) - z(199)*z(415)*z(2610) - z(203)*z(425)*z(2610) -
z(204)*z(435)*z(2610);
z(2400) = z(252)*z(1927) + z(255)*z(1937) + z(258)*z(1947);
z(2563) = z(253)*z(415)*z(2531) + z(256)*z(425)*z(2531) + z(259)*z(435)*z(2531) - z(254)*z(415)*z(2530) - z(257)*z(425)*z(2530) -
z(260)*z(435)*z(2530);
z(2577) = z(2400) + z(2563);
z(2644) = z(254)*z(415)*HOHK2 + z(257)*z(425)*HOHK2 + z(260)*z(435)*HOHK2 - z(253)*z(415)*HOHK3 - z(256)*z(425)*HOHK3 -
z(259)*z(435)*HOHK3;
z(2697) = z(2577) + z(2644);
z(2422) = z(253)*z(1927) + z(256)*z(1937) + z(259)*z(1947);
z(2550) = z(254)*z(415)*z(2529) + z(257)*z(425)*z(2529) + z(260)*z(435)*z(2529) - z(252)*z(415)*z(2531) - z(255)*z(425)*z(2531) -
z(258)*z(435)*z(2531);
z(2590) = z(2422) + z(2550);
z(2631) = z(252)*z(415)*HOHK3 + z(255)*z(425)*HOHK3 + z(258)*z(435)*HOHK3 - z(254)*z(415)*HOHK1 - z(257)*z(425)*HOHK1 -
z(260)*z(435)*HOHK1;
z(2710) = z(2590) + z(2631);
z(2444) = z(254)*z(1927) + z(257)*z(1937) + z(260)*z(1947);
z(2537) = z(252)*z(415)*z(2530) + z(255)*z(425)*z(2530) + z(258)*z(435)*z(2530) - z(253)*z(415)*z(2529) - z(256)*z(425)*z(2529) -
z(259)*z(435)*z(2529);
z(2603) = z(2444) + z(2537);
z(2618) = z(253)*z(415)*HOHK1 + z(256)*z(425)*HOHK1 + z(259)*z(435)*HOHK1 - z(252)*z(415)*HOHK2 - z(255)*z(425)*HOHK2 -
z(258)*z(435)*HOHK2;
z(2723) = z(2603) + z(2618);
z(2683) = z(28)*z(415)*z(2611) + z(199)*z(415)*z(2612) + z(203)*z(425)*z(2612) + z(204)*z(435)*z(2612) - z(205)*z(425)*z(2611) -
z(206)*z(435)*z(2611);
z(2670) = z(205)*z(425)*z(2610) + z(206)*z(435)*z(2610) - z(28)*z(415)*z(2610) - z(197)*z(415)*z(2612) - z(201)*z(425)*z(2612) -
z(202)*z(435)*z(2612);
z(4320) = z(347)*z(4262) + z(350)*z(4263) + z(353)*z(4264) + z(2744)*z(365) + z(2757)*z(362) + z(2770)*z(359) + z(344)*z(4183) +
z(2779)*z(4052) + z(2782)*z(4092) + z(2785)*z(4132) + z(2788)*z(4155) + z(2791)*z(4169) + z(358)*z(2657) + z(4297)*z(2697) + z(4298)*z(2710) +
z(4299)*z(2723) + z(4300)*z(2683) + z(4301)*z(2670);
z(4267) = 0.0005*z(416)*z(4206) + 0.0005*z(426)*z(4207) + 0.0005*z(436)*z(4208) + 0.0005*z(2731)*z(3276) + 0.0005*z(2734)*z(3294) +
0.0005*z(2737)*z(3311) - 0.00893*z(416)*z(4209) - 0.00893*z(426)*z(4210) - 0.00893*z(436)*z(4211) - 0.00893*z(2732)*z(3276) -
0.00893*z(2735)*z(3294) - 0.00893*z(2738)*z(3311);
z(4268) = -0.01653*z(416)*z(4209) - 0.01653*z(426)*z(4210) - 0.01653*z(436)*z(4211) - 0.01653*z(2732)*z(3276) - 0.01653*z(2735)*z(3294) -
0.01653*z(2738)*z(3311) - 0.0005*z(416)*z(4213) - 0.0005*z(426)*z(4214) - 0.0005*z(436)*z(4215) - 0.0005*z(2730)*z(3276) -
0.0005*z(2733)*z(3294) - 0.0005*z(2736)*z(3311);
z(4269) = 0.00893*z(416)*z(4213) + 0.00893*z(426)*z(4214) + 0.00893*z(436)*z(4215) + 0.00893*z(2730)*z(3276) + 0.00893*z(2733)*z(3294) +
0.00893*z(2736)*z(3311) + 0.01653*z(416)*z(4206) + 0.01653*z(426)*z(4207) + 0.01653*z(436)*z(4208) + 0.01653*z(2731)*z(3276) +
0.01653*z(2734)*z(3294) + 0.01653*z(2737)*z(3311);
z(2745) = 0.00893*z(416)*z(2730) + 0.00893*z(426)*z(2733) + 0.00893*z(436)*z(2736) + 0.01653*z(416)*z(2731) + 0.01653*z(426)*z(2734) +
0.01653*z(436)*z(2737);
z(2758) = -0.01653*z(416)*z(2732) - 0.01653*z(426)*z(2735) - 0.01653*z(436)*z(2738) - 0.0005*z(416)*z(2730) - 0.0005*z(426)*z(2733) -
0.0005*z(436)*z(2736);
z(2771) = 0.0005*z(416)*z(2731) + 0.0005*z(426)*z(2734) + 0.0005*z(436)*z(2737) - 0.00893*z(416)*z(2732) - 0.00893*z(426)*z(2735) -
0.00893*z(436)*z(2738);
z(4184) = (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(199)*z(416) + (0.001249+5.719999999999987E-05*T-
5.699999999999868E-06*T^2)*z(203)*z(426) + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(204)*z(436) +
4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(197)*z(416) + 4.049999999999887E-05*(-
1+8.854320987654567*T-1.955555555555609*T^2)*z(201)*z(426) + 4.049999999999887E-05*(-1+8.854320987654567*T-
1.955555555555609*T^2)*z(202)*z(436) + z(197)*z(3276)*z(2611) + z(201)*z(3294)*z(2611) + z(202)*z(3311)*z(2611) + z(416)*z(234)*z(2611) +
z(426)*z(217)*z(2611) + z(436)*z(230)*z(2611) - z(199)*z(3276)*z(2610) - z(203)*z(3294)*z(2610) - z(204)*z(3311)*z(2610) -
z(416)*z(235)*z(2610) - z(426)*z(219)*z(2610) - z(436)*z(229)*z(2610);
z(804) = z(125)*z(788) - z(126)*z(787);
z(797) = z(126)*z(786) - z(121)*z(788);
z(790) = z(121)*z(787) - z(125)*z(786);
z(1149) = z(166)*z(804) + z(168)*z(797) + z(170)*z(790);
z(1253) = z(400)*z(1228) - z(407)*z(1227);
z(1260) = z(1149) + z(1253);
z(1159) = z(172)*z(804) + z(174)*z(797) + z(176)*z(790);
z(1242) = z(407)*z(1226) - z(393)*z(1228);
z(1267) = z(1159) + z(1242);
z(1169) = z(173)*z(804) + z(175)*z(797) + z(177)*z(790);
z(1232) = z(393)*z(1227) - z(400)*z(1226);
z(1274) = z(1169) + z(1232);

z(1807) = z(209)*z(1260) + z(212)*z(1267) + z(215)*z(1274);
z(1891) = z(416)*z(1883) - z(426)*z(1882);
z(1948) = z(1807) + z(1891);
z(1775) = z(207)*z(1260) + z(210)*z(1267) + z(213)*z(1274);
z(1917) = z(426)*z(1884) - z(436)*z(1883);
z(1928) = z(1775) + z(1917);
z(1791) = z(208)*z(1260) + z(211)*z(1267) + z(214)*z(1274);
z(1904) = z(436)*z(1882) - z(416)*z(1884);
z(1938) = z(1791) + z(1904);
z(3673) = z(143)*z(788) - z(154)*z(787);
z(3674) = z(154)*z(786) - z(159)*z(788);
z(3675) = z(159)*z(787) - z(143)*z(786);
z(3676) = z(166)*z(3673) + z(168)*z(3674) + z(170)*z(3675) + z(187)*z(790) + z(188)*z(804) + z(189)*z(797);
z(3781) = z(3229)*z(1228) - z(3241)*z(1227);
z(3708) = z(172)*z(3673) + z(174)*z(3674) + z(176)*z(3675) + z(179)*z(804) + z(181)*z(797) + z(183)*z(790);
z(3782) = z(3241)*z(1226) - z(3217)*z(1228);
z(3730) = z(173)*z(3673) + z(175)*z(3674) + z(177)*z(3675) + z(184)*z(797) + z(185)*z(790) + z(186)*z(804);
z(3783) = z(3217)*z(1227) - z(3229)*z(1226);
z(3784) = z(236)*z(1274) + z(237)*z(1260) + z(238)*z(1267) + z(207)*(z(3676)+z(3781)) + z(210)*(z(3708)+z(3782)) + z(213)*(z(3730)+z(3783));
z(3922) = z(3294)*z(1884) - z(3311)*z(1883);
z(3826) = z(220)*z(1260) + z(224)*z(1267) + z(227)*z(1274) + z(208)*(z(3676)+z(3781)) + z(211)*(z(3708)+z(3782)) + z(214)*(z(3730)+z(3783));
z(3923) = z(3311)*z(1882) - z(3276)*z(1884);
z(3856) = z(231)*z(1267) + z(232)*z(1274) + z(233)*z(1260) + z(209)*(z(3676)+z(3781)) + z(212)*(z(3708)+z(3782)) + z(215)*(z(3730)+z(3783));
z(3924) = z(3276)*z(1883) - z(3294)*z(1882);
z(3925) = z(277)*z(1948) + z(278)*z(1928) + z(279)*z(1938) + z(252)*(z(3784)+z(3922)) + z(255)*(z(3826)+z(3923)) + z(258)*(z(3856)+z(3924));
z(4053) = z(253)*z(3276)*z(2531) + z(256)*z(3294)*z(2531) + z(259)*z(3311)*z(2531) + z(416)*z(265)*z(2531) + z(426)*z(268)*z(2531) +
z(436)*z(271)*z(2531) - z(254)*z(3276)*z(2530) - z(257)*z(3294)*z(2530) - z(260)*z(3311)*z(2530) - z(416)*z(276)*z(2530) - z(426)*z(274)*z(2530)
- z(436)*z(275)*z(2530);
z(4054) = z(254)*z(3276)*HOHK2 + z(257)*z(3294)*HOHK2 + z(260)*z(3311)*HOHK2 + z(416)*z(276)*HOHK2 + z(426)*z(274)*HOHK2 +
z(436)*z(275)*HOHK2 - z(253)*z(3276)*HOHK3 - z(256)*z(3294)*HOHK3 - z(259)*z(3311)*HOHK3 - z(416)*z(265)*HOHK3 -
z(426)*z(268)*HOHK3 - z(436)*z(271)*HOHK3;
z(4055) = z(3925) + z(4053) + z(4054);
z(3973) = z(265)*z(1928) + z(268)*z(1938) + z(271)*z(1948) + z(253)*(z(3784)+z(3922)) + z(256)*(z(3826)+z(3923)) + z(259)*(z(3856)+z(3924));
z(4093) = z(254)*z(3276)*z(2529) + z(257)*z(3294)*z(2529) + z(260)*z(3311)*z(2529) + z(416)*z(276)*z(2529) + z(426)*z(274)*z(2529) +
z(436)*z(275)*z(2529) - z(252)*z(3276)*z(2531) - z(255)*z(3294)*z(2531) - z(258)*z(3311)*z(2531) - z(416)*z(278)*z(2531) - z(426)*z(279)*z(2531)
- z(436)*z(277)*z(2531);
z(4094) = z(252)*z(3276)*HOHK3 + z(255)*z(3294)*HOHK3 + z(258)*z(3311)*HOHK3 + z(416)*z(278)*HOHK3 + z(426)*z(279)*HOHK3 +
z(436)*z(277)*HOHK3 - z(254)*z(3276)*HOHK1 - z(257)*z(3294)*HOHK1 - z(260)*z(3311)*HOHK1 - z(416)*z(276)*HOHK1 -
z(426)*z(274)*HOHK1 - z(436)*z(275)*HOHK1;
z(4095) = z(3973) + z(4093) + z(4094);
z(4009) = z(274)*z(1938) + z(275)*z(1948) + z(276)*z(1928) + z(254)*(z(3784)+z(3922)) + z(257)*(z(3826)+z(3923)) + z(260)*(z(3856)+z(3924));
z(4133) = z(252)*z(3276)*z(2530) + z(255)*z(3294)*z(2530) + z(258)*z(3311)*z(2530) + z(416)*z(278)*z(2530) + z(426)*z(279)*z(2530) +
z(436)*z(277)*z(2530) - z(253)*z(3276)*z(2529) - z(256)*z(3294)*z(2529) - z(259)*z(3311)*z(2529) - z(416)*z(265)*z(2529) - z(426)*z(268)*z(2529)
- z(436)*z(271)*z(2529);
z(4134) = z(253)*z(3276)*HOHK1 + z(256)*z(3294)*HOHK1 + z(259)*z(3311)*HOHK1 + z(416)*z(265)*HOHK1 + z(426)*z(268)*HOHK1 +
z(436)*z(271)*HOHK1 - z(252)*z(3276)*HOHK2 - z(255)*z(3294)*HOHK2 - z(258)*z(3311)*HOHK2 - z(416)*z(278)*HOHK2 -
z(426)*z(279)*HOHK2 - z(436)*z(277)*HOHK2;
z(4135) = z(4009) + z(4133) + z(4134);
z(4156) = 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(28)*z(416) + z(28)*z(3276)*z(2611) +
z(199)*z(3276)*z(2612) + z(203)*z(3294)*z(2612) + z(204)*z(3311)*z(2612) + z(416)*z(235)*z(2612) + z(426)*z(219)*z(2612) +
z(436)*z(229)*z(2612) + z(27)*z(416)*QNE_2p*z(2611) - 4.049999999999887E-05*(-1+8.854320987654567*T-
1.955555555555609*T^2)*z(205)*z(426) - 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(206)*z(436) -
z(205)*z(3294)*z(2611) - z(206)*z(3311)*z(2611) - z(426)*z(221)*z(2611) - z(436)*z(228)*z(2611);
z(4170) = (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(28)*z(416) + z(205)*z(3294)*z(2610) + z(206)*z(3311)*z(2610) +
z(426)*z(221)*z(2610) + z(436)*z(228)*z(2610) - (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(205)*z(426) -
(0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(206)*z(436) - z(28)*z(3276)*z(2610) - z(197)*z(3276)*z(2612) -
z(201)*z(3294)*z(2612) - z(202)*z(3311)*z(2612) - z(416)*z(234)*z(2612) - z(426)*z(217)*z(2612) - z(436)*z(230)*z(2612) -
z(27)*z(416)*QNE_2p*z(2610);
z(2658) = z(197)*z(416)*z(2611) + z(201)*z(426)*z(2611) + z(202)*z(436)*z(2611) - z(199)*z(416)*z(2610) - z(203)*z(426)*z(2610) -
z(204)*z(436)*z(2610);
z(2401) = z(252)*z(1928) + z(255)*z(1938) + z(258)*z(1948);
z(2564) = z(253)*z(416)*z(2531) + z(256)*z(426)*z(2531) + z(259)*z(436)*z(2531) - z(254)*z(416)*z(2530) - z(257)*z(426)*z(2530) -
z(260)*z(436)*z(2530);
z(2578) = z(2401) + z(2564);
z(2645) = z(254)*z(416)*HOHK2 + z(257)*z(426)*HOHK2 + z(260)*z(436)*HOHK2 - z(253)*z(416)*HOHK3 - z(256)*z(426)*HOHK3 -
z(259)*z(436)*HOHK3;
z(2698) = z(2578) + z(2645);
z(2423) = z(253)*z(1928) + z(256)*z(1938) + z(259)*z(1948);
z(2551) = z(254)*z(416)*z(2529) + z(257)*z(426)*z(2529) + z(260)*z(436)*z(2529) - z(252)*z(416)*z(2531) - z(255)*z(426)*z(2531) -
z(258)*z(436)*z(2531);
z(2591) = z(2423) + z(2551);
z(2632) = z(252)*z(416)*HOHK3 + z(255)*z(426)*HOHK3 + z(258)*z(436)*HOHK3 - z(254)*z(416)*HOHK1 - z(257)*z(426)*HOHK1 -
z(260)*z(436)*HOHK1;
z(2711) = z(2591) + z(2632);
z(2445) = z(254)*z(1928) + z(257)*z(1938) + z(260)*z(1948);
z(2538) = z(252)*z(416)*z(2530) + z(255)*z(426)*z(2530) + z(258)*z(436)*z(2530) - z(253)*z(416)*z(2529) - z(256)*z(426)*z(2529) -
z(259)*z(436)*z(2529);
z(2604) = z(2445) + z(2538);

z(2619) = z(253)*z(416)*HOHK1 + z(256)*z(426)*HOHK1 + z(259)*z(436)*HOHK1 - z(252)*z(416)*HOHK2 - z(255)*z(426)*HOHK2 - z(258)*z(436)*HOHK2;
z(2724) = z(2604) + z(2619);
z(2684) = z(28)*z(416)*z(2611) + z(199)*z(416)*z(2612) + z(203)*z(426)*z(2612) + z(204)*z(436)*z(2612) - z(205)*z(426)*z(2611) - z(206)*z(436)*z(2611);
z(2671) = z(205)*z(426)*z(2610) + z(206)*z(436)*z(2610) - z(28)*z(416)*z(2610) - z(197)*z(416)*z(2612) - z(201)*z(426)*z(2612) - z(202)*z(436)*z(2612);
z(4321) = z(347)*z(4267) + z(350)*z(4268) + z(353)*z(4269) + z(2745)*z(365) + z(2758)*z(362) + z(2771)*z(359) + z(344)*z(4184) + z(2779)*z(4055) + z(2782)*z(4095) + z(2785)*z(4135) + z(2788)*z(4156) + z(2791)*z(4170) + z(358)*z(2658) + z(4297)*z(2698) + z(4298)*z(2711) + z(4299)*z(2724) + z(4300)*z(2684) + z(4301)*z(2671);
z(4272) = 0.0005*z(417)*z(4206) + 0.0005*z(427)*z(4207) + 0.0005*z(437)*z(4208) + 0.0005*z(2731)*z(3277) + 0.0005*z(2734)*z(3295) + 0.0005*z(2737)*z(3312) - 0.00893*z(417)*z(4209) - 0.00893*z(427)*z(4210) - 0.00893*z(437)*z(4211) - 0.00893*z(2732)*z(3277) - 0.00893*z(2735)*z(3295) - 0.00893*z(2738)*z(3312);
z(4273) = -0.01653*z(417)*z(4209) - 0.01653*z(427)*z(4210) - 0.01653*z(437)*z(4211) - 0.01653*z(2732)*z(3277) - 0.01653*z(2735)*z(3295) - 0.01653*z(2738)*z(3312) - 0.0005*z(417)*z(4213) - 0.0005*z(427)*z(4214) - 0.0005*z(437)*z(4215) - 0.0005*z(2730)*z(3277) - 0.0005*z(2733)*z(3295) - 0.0005*z(2736)*z(3312);
z(4274) = 0.00893*z(417)*z(4213) + 0.00893*z(427)*z(4214) + 0.00893*z(437)*z(4215) + 0.00893*z(2730)*z(3277) + 0.00893*z(2733)*z(3295) + 0.00893*z(2736)*z(3312) + 0.01653*z(417)*z(4206) + 0.01653*z(427)*z(4207) + 0.01653*z(437)*z(4208) + 0.01653*z(2731)*z(3277) + 0.01653*z(2734)*z(3295) + 0.01653*z(2737)*z(3312);
z(2746) = 0.00893*z(417)*z(2730) + 0.00893*z(427)*z(2733) + 0.00893*z(437)*z(2736) + 0.01653*z(417)*z(2731) + 0.01653*z(427)*z(2734) + 0.01653*z(437)*z(2737);
z(2759) = -0.01653*z(417)*z(2732) - 0.01653*z(427)*z(2735) - 0.01653*z(437)*z(2738) - 0.0005*z(417)*z(2730) - 0.0005*z(427)*z(2733) - 0.0005*z(437)*z(2736);
z(2772) = 0.0005*z(417)*z(2731) + 0.0005*z(427)*z(2734) + 0.0005*z(437)*z(2737) - 0.00893*z(417)*z(2732) - 0.00893*z(427)*z(2735) - 0.00893*z(437)*z(2738);
z(4185) = (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(199)*z(417) + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(203)*z(427) + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(204)*z(437) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(197)*z(417) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(201)*z(427) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(202)*z(437) + z(197)*z(3277)*z(2611) + z(201)*z(3295)*z(2611) + z(202)*z(3312)*z(2611) + z(417)*z(234)*z(2611) + z(427)*z(217)*z(2611) + z(437)*z(230)*z(2611) - z(199)*z(3277)*z(2610) - z(203)*z(3295)*z(2610) - z(204)*z(3312)*z(2610) - z(417)*z(235)*z(2610) - z(427)*z(219)*z(2610) - z(437)*z(229)*z(2610);
z(805) = z(127)*z(788) - z(128)*z(787);
z(798) = z(16)*z(788) + z(128)*z(786);
z(791) = -z(16)*z(787) - z(127)*z(786);
z(1150) = z(166)*z(805) + z(168)*z(798) + z(170)*z(791);
z(1254) = z(401)*z(1228) - z(408)*z(1227);
z(1261) = z(1150) + z(1254);
z(1160) = z(172)*z(805) + z(174)*z(798) + z(176)*z(791);
z(1243) = z(408)*z(1226) - z(394)*z(1228);
z(1268) = z(1160) + z(1243);
z(1170) = z(173)*z(805) + z(175)*z(798) + z(177)*z(791);
z(1233) = z(394)*z(1227) - z(401)*z(1226);
z(1275) = z(1170) + z(1233);
z(1808) = z(209)*z(1261) + z(212)*z(1268) + z(215)*z(1275);
z(1892) = z(417)*z(1883) - z(427)*z(1882);
z(1949) = z(1808) + z(1892);
z(1776) = z(207)*z(1261) + z(210)*z(1268) + z(213)*z(1275);
z(1918) = z(427)*z(1884) - z(437)*z(1883);
z(1929) = z(1776) + z(1918);
z(1792) = z(208)*z(1261) + z(211)*z(1268) + z(214)*z(1275);
z(1905) = z(437)*z(1882) - z(417)*z(1884);
z(1939) = z(1792) + z(1905);
z(3678) = z(144)*z(788) - z(152)*z(787);
z(3679) = z(152)*z(786) + z(15)*QNC_2p*z(788);
z(3680) = -z(144)*z(786) - z(15)*QNC_2p*z(787);
z(3681) = z(166)*z(3678) + z(168)*z(3679) + z(170)*z(3680) + z(187)*z(791) + z(188)*z(805) + z(189)*z(798);
z(3786) = z(3230)*z(1228) - z(3242)*z(1227);
z(3710) = z(172)*z(3678) + z(174)*z(3679) + z(176)*z(3680) + z(179)*z(805) + z(181)*z(798) + z(183)*z(791);
z(3787) = z(3242)*z(1226) - z(3218)*z(1228);
z(3732) = z(173)*z(3678) + z(175)*z(3679) + z(177)*z(3680) + z(184)*z(798) + z(185)*z(791) + z(186)*z(805);
z(3788) = z(3218)*z(1227) - z(3230)*z(1226);
z(3789) = z(236)*z(1275) + z(237)*z(1261) + z(238)*z(1268) + z(207)*(z(3681)+z(3786)) + z(210)*(z(3710)+z(3787)) + z(213)*(z(3732)+z(3788));
z(3927) = z(3295)*z(1884) - z(3312)*z(1883);
z(3828) = z(220)*z(1261) + z(224)*z(1268) + z(227)*z(1275) + z(208)*(z(3681)+z(3786)) + z(211)*(z(3710)+z(3787)) + z(214)*(z(3732)+z(3788));
z(3928) = z(3312)*z(1882) - z(3277)*z(1884);
z(3858) = z(231)*z(1268) + z(232)*z(1275) + z(233)*z(1261) + z(209)*(z(3681)+z(3786)) + z(212)*(z(3710)+z(3787)) + z(215)*(z(3732)+z(3788));
z(3929) = z(3277)*z(1883) - z(3295)*z(1882);
z(3930) = z(277)*z(1949) + z(278)*z(1929) + z(279)*z(1939) + z(252)*(z(3789)+z(3927)) + z(255)*(z(3828)+z(3928)) + z(258)*(z(3858)+z(3929));
z(4056) = z(253)*z(3277)*z(2531) + z(256)*z(3295)*z(2531) + z(259)*z(3312)*z(2531) + z(417)*z(265)*z(2531) + z(427)*z(268)*z(2531) + z(437)*z(271)*z(2531) - z(254)*z(3277)*z(2530) - z(257)*z(3295)*z(2530) - z(260)*z(3312)*z(2530) - z(417)*z(276)*z(2530) - z(427)*z(274)*z(2530) - z(437)*z(275)*z(2530);
z(4057) = z(254)*z(3277)*HOHK2 + z(257)*z(3295)*HOHK2 + z(260)*z(3312)*HOHK2 + z(417)*z(276)*HOHK2 + z(427)*z(274)*HOHK2 + z(437)*z(275)*HOHK2 - z(253)*z(3277)*HOHK3 - z(256)*z(3295)*HOHK3 - z(259)*z(3312)*HOHK3 - z(417)*z(265)*HOHK3 - z(427)*z(268)*HOHK3 - z(437)*z(271)*HOHK3;
z(4058) = z(3930) + z(4056) + z(4057);
z(3975) = z(265)*z(1929) + z(268)*z(1939) + z(271)*z(1949) + z(253)*(z(3789)+z(3927)) + z(256)*(z(3828)+z(3928)) + z(259)*(z(3858)+z(3929));

z(4096) = z(254)*z(3277)*z(2529) + z(257)*z(3295)*z(2529) + z(260)*z(3312)*z(2529) + z(417)*z(276)*z(2529) + z(427)*z(274)*z(2529) +
z(437)*z(275)*z(2529) - z(252)*z(3277)*z(2531) - z(255)*z(3295)*z(2531) - z(258)*z(3312)*z(2531) - z(417)*z(278)*z(2531) - z(427)*z(279)*z(2531)
- z(437)*z(277)*z(2531);
z(4097) = z(252)*z(3277)*HOHK3 + z(255)*z(3295)*HOHK3 + z(258)*z(3312)*HOHK3 + z(417)*z(278)*HOHK3 + z(427)*z(279)*HOHK3 +
z(437)*z(277)*HOHK3 - z(254)*z(3277)*HOHK1 - z(257)*z(3295)*HOHK1 - z(260)*z(3312)*HOHK1 - z(417)*z(276)*HOHK1 -
z(427)*z(274)*HOHK1 - z(437)*z(275)*HOHK1;
z(4098) = z(3975) + z(4096) + z(4097);
z(4011) = z(274)*z(1939) + z(275)*z(1949) + z(276)*z(1929) + z(254)*(z(3789)+z(3927)) + z(257)*(z(3828)+z(3928)) + z(260)*(z(3858)+z(3929));
z(4136) = z(252)*z(3277)*z(2530) + z(255)*z(3295)*z(2530) + z(258)*z(3312)*z(2530) + z(417)*z(278)*z(2530) + z(427)*z(279)*z(2530) +
z(437)*z(277)*z(2530) - z(253)*z(3277)*z(2529) - z(256)*z(3295)*z(2529) - z(259)*z(3312)*z(2529) - z(417)*z(265)*z(2529) - z(427)*z(268)*z(2529)
- z(437)*z(271)*z(2529);
z(4137) = z(253)*z(3277)*HOHK1 + z(256)*z(3295)*HOHK1 + z(259)*z(3312)*HOHK1 + z(417)*z(265)*HOHK1 + z(427)*z(268)*HOHK1 +
z(437)*z(271)*HOHK1 - z(252)*z(3277)*HOHK2 - z(255)*z(3295)*HOHK2 - z(258)*z(3312)*HOHK2 - z(417)*z(278)*HOHK2 -
z(427)*z(279)*HOHK2 - z(437)*z(277)*HOHK2;
z(4138) = z(4011) + z(4136) + z(4137);
z(4157) = 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(28)*z(417) + z(28)*z(3277)*z(2611) +
z(199)*z(3277)*z(2612) + z(203)*z(3295)*z(2612) + z(204)*z(3312)*z(2612) + z(417)*z(235)*z(2612) + z(427)*z(219)*z(2612) +
z(437)*z(229)*z(2612) + z(27)*z(417)*QNE_2p*z(2611) - 4.049999999999887E-05*(-1+8.854320987654567*T-
1.955555555555609*T^2)*z(205)*z(427) - 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(206)*z(437) -
z(205)*z(3295)*z(2611) - z(206)*z(3312)*z(2611) - z(427)*z(221)*z(2611) - z(437)*z(228)*z(2611);
z(4171) = (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(28)*z(417) + z(205)*z(3295)*z(2610) + z(206)*z(3312)*z(2610) +
z(427)*z(221)*z(2610) + z(437)*z(228)*z(2610) - (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(205)*z(427) -
(0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(206)*z(437) - z(28)*z(3277)*z(2610) - z(197)*z(3277)*z(2612) -
z(201)*z(3295)*z(2612) - z(202)*z(3312)*z(2612) - z(417)*z(234)*z(2612) - z(427)*z(217)*z(2612) - z(437)*z(230)*z(2612) -
z(27)*z(417)*QNE_2p*z(2610);
z(2659) = z(197)*z(417)*z(2611) + z(201)*z(427)*z(2611) + z(202)*z(437)*z(2611) - z(199)*z(417)*z(2610) - z(203)*z(427)*z(2610) -
z(204)*z(437)*z(2610);
z(2402) = z(252)*z(1929) + z(255)*z(1939) + z(258)*z(1949);
z(2565) = z(253)*z(417)*z(2531) + z(256)*z(427)*z(2531) + z(259)*z(437)*z(2531) - z(254)*z(417)*z(2530) - z(257)*z(427)*z(2530) -
z(260)*z(437)*z(2530);
z(2579) = z(2402) + z(2565);
z(2646) = z(254)*z(417)*HOHK2 + z(257)*z(427)*HOHK2 + z(260)*z(437)*HOHK2 - z(253)*z(417)*HOHK3 - z(256)*z(427)*HOHK3 -
z(259)*z(437)*HOHK3;
z(2699) = z(2579) + z(2646);
z(2424) = z(253)*z(1929) + z(256)*z(1939) + z(259)*z(1949);
z(2552) = z(254)*z(417)*z(2529) + z(257)*z(427)*z(2529) + z(260)*z(437)*z(2529) - z(252)*z(417)*z(2531) - z(255)*z(427)*z(2531) -
z(258)*z(437)*z(2531);
z(2592) = z(2424) + z(2552);
z(2633) = z(252)*z(417)*HOHK3 + z(255)*z(427)*HOHK3 + z(258)*z(437)*HOHK3 - z(254)*z(417)*HOHK1 - z(257)*z(427)*HOHK1 -
z(260)*z(437)*HOHK1;
z(2712) = z(2592) + z(2633);
z(2446) = z(254)*z(1929) + z(257)*z(1939) + z(260)*z(1949);
z(2539) = z(252)*z(417)*z(2530) + z(255)*z(427)*z(2530) + z(258)*z(437)*z(2530) - z(253)*z(417)*z(2529) - z(256)*z(427)*z(2529) -
z(259)*z(437)*z(2529);
z(2605) = z(2446) + z(2539);
z(2620) = z(253)*z(417)*HOHK1 + z(256)*z(427)*HOHK1 + z(259)*z(437)*HOHK1 - z(252)*z(417)*HOHK2 - z(255)*z(427)*HOHK2 -
z(258)*z(437)*HOHK2;
z(2725) = z(2605) + z(2620);
z(2685) = z(28)*z(417)*z(2611) + z(199)*z(417)*z(2612) + z(203)*z(427)*z(2612) + z(204)*z(437)*z(2612) - z(205)*z(427)*z(2611) -
z(206)*z(437)*z(2611);
z(2672) = z(205)*z(427)*z(2610) + z(206)*z(437)*z(2610) - z(28)*z(417)*z(2610) - z(197)*z(417)*z(2612) - z(201)*z(427)*z(2612) -
z(202)*z(437)*z(2612);
z(4322) = z(347)*z(4272) + z(350)*z(4273) + z(353)*z(4274) + z(2746)*z(365) + z(2759)*z(362) + z(2772)*z(359) + z(344)*z(4185) +
z(2779)*z(4058) + z(2782)*z(4098) + z(2785)*z(4138) + z(2788)*z(4157) + z(2791)*z(4171) + z(358)*z(2659) + z(4297)*z(2699) + z(4298)*z(2712) +
z(4299)*z(2725) + z(4300)*z(2685) + z(4301)*z(2672);
z(4277) = 0.0005*z(418)*z(4206) + 0.0005*z(428)*z(4207) + 0.0005*z(438)*z(4208) + 0.0005*z(2731)*z(3278) + 0.0005*z(2734)*z(3296) +
0.0005*z(2737)*z(3313) - 0.00893*z(418)*z(4209) - 0.00893*z(428)*z(4210) - 0.00893*z(438)*z(4211) - 0.00893*z(2732)*z(3278) -
0.00893*z(2735)*z(3296) - 0.00893*z(2738)*z(3313);
z(4278) = -0.01653*z(418)*z(4209) - 0.01653*z(428)*z(4210) - 0.01653*z(438)*z(4211) - 0.01653*z(2732)*z(3278) - 0.01653*z(2735)*z(3296) -
0.01653*z(2738)*z(3313) - 0.0005*z(418)*z(4213) - 0.0005*z(428)*z(4214) - 0.0005*z(438)*z(4215) - 0.0005*z(2730)*z(3278) -
0.0005*z(2733)*z(3296) - 0.0005*z(2736)*z(3313);
z(4279) = 0.00893*z(418)*z(4213) + 0.00893*z(428)*z(4214) + 0.00893*z(438)*z(4215) + 0.00893*z(2730)*z(3278) + 0.00893*z(2733)*z(3296) +
0.00893*z(2736)*z(3313) + 0.01653*z(418)*z(4206) + 0.01653*z(428)*z(4207) + 0.01653*z(438)*z(4208) + 0.01653*z(2731)*z(3278) +
0.01653*z(2734)*z(3296) + 0.01653*z(2737)*z(3313);
z(2747) = 0.00893*z(418)*z(2730) + 0.00893*z(428)*z(2733) + 0.00893*z(438)*z(2736) + 0.01653*z(418)*z(2731) + 0.01653*z(428)*z(2734) +
0.01653*z(438)*z(2737);
z(2760) = -0.01653*z(418)*z(2732) - 0.01653*z(428)*z(2735) - 0.01653*z(438)*z(2738) - 0.0005*z(418)*z(2730) - 0.0005*z(428)*z(2733) -
0.0005*z(438)*z(2736);
z(2773) = 0.0005*z(418)*z(2731) + 0.0005*z(428)*z(2734) + 0.0005*z(438)*z(2737) - 0.00893*z(418)*z(2732) - 0.00893*z(428)*z(2735) -
0.00893*z(438)*z(2738);
z(4186) = (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(199)*z(418) + (0.001249+5.719999999999987E-05*T-
5.699999999999868E-06*T^2)*z(203)*z(428) + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(204)*z(438) +
4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(197)*z(418) + 4.049999999999887E-05*(-
1+8.854320987654567*T-1.955555555555609*T^2)*z(201)*z(428) + 4.049999999999887E-05*(-1+8.854320987654567*T-
1.955555555555609*T^2)*z(202)*z(438) + z(197)*z(3278)*z(2611) + z(201)*z(3296)*z(2611) + z(202)*z(3313)*z(2611) + z(418)*z(234)*z(2611) +
z(428)*z(217)*z(2611) + z(438)*z(230)*z(2611) - z(199)*z(3278)*z(2610) - z(203)*z(3296)*z(2610) - z(204)*z(3313)*z(2610) -
z(418)*z(235)*z(2610) - z(428)*z(219)*z(2610) - z(438)*z(229)*z(2610);
z(573) = z(80)*z(564) - z(81)*z(563);

z(569) = z(10)*z(564) + z(81)*z(562);
z(565) = -z(10)*z(563) - z(80)*z(562);
z(741) = z(129)*z(573) + z(132)*z(569) + z(135)*z(565);
z(806) = z(384)*z(788) - z(388)*z(787);
z(810) = z(741) + z(806);
z(745) = z(130)*z(573) + z(133)*z(569) + z(136)*z(565);
z(799) = z(388)*z(786) - z(380)*z(788);
z(814) = z(745) + z(799);
z(749) = z(131)*z(573) + z(134)*z(569) + z(137)*z(565);
z(792) = z(380)*z(787) - z(384)*z(786);
z(818) = z(749) + z(792);
z(1151) = z(166)*z(810) + z(168)*z(814) + z(170)*z(818);
z(1255) = z(402)*z(1228) - z(409)*z(1227);
z(1262) = z(1151) + z(1255);
z(1161) = z(172)*z(810) + z(174)*z(814) + z(176)*z(818);
z(1244) = z(409)*z(1226) - z(395)*z(1228);
z(1269) = z(1161) + z(1244);
z(1171) = z(173)*z(810) + z(175)*z(814) + z(177)*z(818);
z(1234) = z(395)*z(1227) - z(402)*z(1226);
z(1276) = z(1171) + z(1234);
z(1809) = z(209)*z(1262) + z(212)*z(1269) + z(215)*z(1276);
z(1893) = z(418)*z(1883) - z(428)*z(1882);
z(1950) = z(1809) + z(1893);
z(1777) = z(207)*z(1262) + z(210)*z(1269) + z(213)*z(1276);
z(1919) = z(428)*z(1884) - z(438)*z(1883);
z(1930) = z(1777) + z(1919);
z(1793) = z(208)*z(1262) + z(211)*z(1269) + z(214)*z(1276);
z(1906) = z(438)*z(1882) - z(418)*z(1884);
z(1940) = z(1793) + z(1906);
z(3611) = z(94)*z(564) - z(102)*z(563);
z(3612) = z(102)*z(562) + z(9)*QNB_2p*z(564);
z(3613) = -z(94)*z(562) - z(9)*QNB_2p*z(563);
z(3614) = z(129)*z(3611) + z(132)*z(3612) + z(135)*z(3613) + z(160)*z(565) + z(161)*z(573) + z(162)*z(569);
z(3683) = z(3181)*z(788) - z(3192)*z(787);
z(3632) = z(130)*z(3611) + z(133)*z(3612) + z(136)*z(3613) + z(145)*z(573) + z(148)*z(569) + z(151)*z(565);
z(3684) = z(3192)*z(786) - z(3169)*z(788);
z(3644) = z(131)*z(3611) + z(134)*z(3612) + z(137)*z(3613) + z(155)*z(569) + z(156)*z(565) + z(157)*z(573);
z(3685) = z(3169)*z(787) - z(3181)*z(786);
z(3686) = z(187)*z(818) + z(188)*z(810) + z(189)*z(814) + z(166)*(z(3614)+z(3683)) + z(168)*(z(3632)+z(3684)) + z(170)*(z(3644)+z(3685));
z(3791) = z(3231)*z(1228) - z(3243)*z(1227);
z(3712) = z(179)*z(810) + z(181)*z(814) + z(183)*z(818) + z(172)*(z(3614)+z(3683)) + z(174)*(z(3632)+z(3684)) + z(176)*(z(3644)+z(3685));
z(3792) = z(3243)*z(1226) - z(3219)*z(1228);
z(3734) = z(184)*z(814) + z(185)*z(818) + z(186)*z(810) + z(173)*(z(3614)+z(3683)) + z(175)*(z(3632)+z(3684)) + z(177)*(z(3644)+z(3685));
z(3793) = z(3219)*z(1227) - z(3231)*z(1226);
z(3794) = z(236)*z(1276) + z(237)*z(1262) + z(238)*z(1269) + z(207)*(z(3686)+z(3791)) + z(210)*(z(3712)+z(3792)) + z(213)*(z(3734)+z(3793));
z(3932) = z(3296)*z(1884) - z(3313)*z(1883);
z(3830) = z(220)*z(1262) + z(224)*z(1269) + z(227)*z(1276) + z(208)*(z(3686)+z(3791)) + z(211)*(z(3712)+z(3792)) + z(214)*(z(3734)+z(3793));
z(3933) = z(3313)*z(1882) - z(3278)*z(1884);
z(3860) = z(231)*z(1269) + z(232)*z(1276) + z(233)*z(1262) + z(209)*(z(3686)+z(3791)) + z(212)*(z(3712)+z(3792)) + z(215)*(z(3734)+z(3793));
z(3934) = z(3278)*z(1883) - z(3296)*z(1882);
z(3935) = z(277)*z(1950) + z(278)*z(1930) + z(279)*z(1940) + z(252)*(z(3794)+z(3932)) + z(255)*(z(3830)+z(3933)) + z(258)*(z(3860)+z(3934));
z(4059) = z(253)*z(3278)*z(2531) + z(256)*z(3296)*z(2531) + z(259)*z(3313)*z(2531) + z(418)*z(265)*z(2531) + z(428)*z(268)*z(2531) +
z(438)*z(271)*z(2531) - z(254)*z(3278)*z(2530) - z(257)*z(3296)*z(2530) - z(260)*z(3313)*z(2530) - z(418)*z(276)*z(2530) - z(428)*z(274)*z(2530)
- z(438)*z(275)*z(2530);
z(4060) = z(254)*z(3278)*HOHK2 + z(257)*z(3296)*HOHK2 + z(260)*z(3313)*HOHK2 + z(418)*z(276)*HOHK2 + z(428)*z(274)*HOHK2 +
z(438)*z(275)*HOHK2 - z(253)*z(3278)*HOHK3 - z(256)*z(3296)*HOHK3 - z(259)*z(3313)*HOHK3 - z(418)*z(265)*HOHK3 -
z(428)*z(268)*HOHK3 - z(438)*z(271)*HOHK3;
z(4061) = z(3935) + z(4059) + z(4060);
z(3977) = z(265)*z(1930) + z(268)*z(1940) + z(271)*z(1950) + z(253)*(z(3794)+z(3932)) + z(256)*(z(3830)+z(3933)) + z(259)*(z(3860)+z(3934));
z(4099) = z(254)*z(3278)*z(2529) + z(257)*z(3296)*z(2529) + z(260)*z(3313)*z(2529) + z(418)*z(276)*z(2529) + z(428)*z(274)*z(2529) +
z(438)*z(275)*z(2529) - z(252)*z(3278)*z(2531) - z(255)*z(3296)*z(2531) - z(258)*z(3313)*z(2531) - z(418)*z(278)*z(2531) - z(428)*z(279)*z(2531)
- z(438)*z(277)*z(2531);
z(4100) = z(252)*z(3278)*HOHK3 + z(255)*z(3296)*HOHK3 + z(258)*z(3313)*HOHK3 + z(418)*z(278)*HOHK3 + z(428)*z(279)*HOHK3 +
z(438)*z(277)*HOHK3 - z(254)*z(3278)*HOHK1 - z(257)*z(3296)*HOHK1 - z(260)*z(3313)*HOHK1 - z(418)*z(276)*HOHK1 -
z(428)*z(274)*HOHK1 - z(438)*z(275)*HOHK1;
z(4101) = z(3977) + z(4099) + z(4100);
z(4013) = z(274)*z(1940) + z(275)*z(1950) + z(276)*z(1930) + z(254)*(z(3794)+z(3932)) + z(257)*(z(3830)+z(3933)) + z(260)*(z(3860)+z(3934));
z(4139) = z(252)*z(3278)*z(2530) + z(255)*z(3296)*z(2530) + z(258)*z(3313)*z(2530) + z(418)*z(278)*z(2530) + z(428)*z(279)*z(2530) +
z(438)*z(277)*z(2530) - z(253)*z(3278)*z(2529) - z(256)*z(3296)*z(2529) - z(259)*z(3313)*z(2529) - z(418)*z(265)*z(2529) - z(428)*z(268)*z(2529)
- z(438)*z(271)*z(2529);
z(4140) = z(253)*z(3278)*HOHK1 + z(256)*z(3296)*HOHK1 + z(259)*z(3313)*HOHK1 + z(418)*z(265)*HOHK1 + z(428)*z(268)*HOHK1 +
z(438)*z(271)*HOHK1 - z(252)*z(3278)*HOHK2 - z(255)*z(3296)*HOHK2 - z(258)*z(3313)*HOHK2 - z(418)*z(278)*HOHK2 -
z(428)*z(279)*HOHK2 - z(438)*z(277)*HOHK2;
z(4141) = z(4013) + z(4139) + z(4140);
z(4158) = 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(28)*z(418) + z(28)*z(3278)*z(2611) +
z(199)*z(3278)*z(2612) + z(203)*z(3296)*z(2612) + z(204)*z(3313)*z(2612) + z(418)*z(235)*z(2612) + z(428)*z(219)*z(2612) +
z(438)*z(229)*z(2612) + z(27)*z(418)*QNE_2p*z(2611) - 4.049999999999887E-05*(-1+8.854320987654567*T-

1.955555555555609*T^2)*z(205)*z(428) - 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(206)*z(438) - z(205)*z(3296)*z(2611) - z(206)*z(3313)*z(2611) - z(428)*z(221)*z(2611) - z(438)*z(228)*z(2611);
z(4172) = (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(28)*z(418) + z(205)*z(3296)*z(2610) + z(206)*z(3313)*z(2610) + z(428)*z(221)*z(2610) + z(438)*z(228)*z(2610) - (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(205)*z(428) - (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(206)*z(438) - z(28)*z(3278)*z(2610) - z(197)*z(3278)*z(2612) - z(201)*z(3296)*z(2612) - z(202)*z(3313)*z(2612) - z(418)*z(234)*z(2612) - z(428)*z(217)*z(2612) - z(438)*z(230)*z(2612) - z(27)*z(418)*QNE_2p*z(2610);
z(2660) = z(197)*z(418)*z(2611) + z(201)*z(428)*z(2611) + z(202)*z(438)*z(2611) - z(199)*z(418)*z(2610) - z(203)*z(428)*z(2610) - z(204)*z(438)*z(2610);
z(2403) = z(252)*z(1930) + z(255)*z(1940) + z(258)*z(1950);
z(2566) = z(253)*z(418)*z(2531) + z(256)*z(428)*z(2531) + z(259)*z(438)*z(2531) - z(254)*z(418)*z(2530) - z(257)*z(428)*z(2530) - z(260)*z(438)*z(2530);
z(2580) = z(2403) + z(2566);
z(2647) = z(254)*z(418)*HOHK2 + z(257)*z(428)*HOHK2 + z(260)*z(438)*HOHK2 - z(253)*z(418)*HOHK3 - z(256)*z(428)*HOHK3 - z(259)*z(438)*HOHK3;
z(2700) = z(2580) + z(2647);
z(2425) = z(253)*z(1930) + z(256)*z(1940) + z(259)*z(1950);
z(2553) = z(254)*z(418)*z(2529) + z(257)*z(428)*z(2529) + z(260)*z(438)*z(2529) - z(252)*z(418)*z(2531) - z(255)*z(428)*z(2531) - z(258)*z(438)*z(2531);
z(2593) = z(2425) + z(2553);
z(2634) = z(252)*z(418)*HOHK3 + z(255)*z(428)*HOHK3 + z(258)*z(438)*HOHK3 - z(254)*z(418)*HOHK1 - z(257)*z(428)*HOHK1 - z(260)*z(438)*HOHK1;
z(2713) = z(2593) + z(2634);
z(2447) = z(254)*z(1930) + z(257)*z(1940) + z(260)*z(1950);
z(2540) = z(252)*z(418)*z(2530) + z(255)*z(428)*z(2530) + z(258)*z(438)*z(2530) - z(253)*z(418)*z(2529) - z(256)*z(428)*z(2529) - z(259)*z(438)*z(2529);
z(2606) = z(2447) + z(2540);
z(2621) = z(253)*z(418)*HOHK1 + z(256)*z(428)*HOHK1 + z(259)*z(438)*HOHK1 - z(252)*z(418)*HOHK2 - z(255)*z(428)*HOHK2 - z(258)*z(438)*HOHK2;
z(2726) = z(2606) + z(2621);
z(2686) = z(28)*z(418)*z(2611) + z(199)*z(418)*z(2612) + z(203)*z(428)*z(2612) + z(204)*z(438)*z(2612) - z(205)*z(428)*z(2611) - z(206)*z(438)*z(2611);
z(2673) = z(205)*z(428)*z(2610) + z(206)*z(438)*z(2610) - z(28)*z(418)*z(2610) - z(197)*z(418)*z(2612) - z(201)*z(428)*z(2612) - z(202)*z(438)*z(2612);
z(4323) = z(347)*z(4277) + z(350)*z(4278) + z(353)*z(4279) + z(2747)*z(365) + z(2760)*z(362) + z(2773)*z(359) + z(344)*z(4186) + z(2779)*z(4061) + z(2782)*z(4101) + z(2785)*z(4141) + z(2788)*z(4158) + z(2791)*z(4172) + z(358)*z(2660) + z(4297)*z(2700) + z(4298)*z(2713) + z(4299)*z(2726) + z(4300)*z(2686) + z(4301)*z(2673);
z(4282) = 0.0005*z(419)*z(4206) + 0.0005*z(429)*z(4207) + 0.0005*z(439)*z(4208) + 0.0005*z(2731)*z(3279) + 0.0005*z(2734)*z(3297) + 0.0005*z(2737)*z(3314) - 0.00893*z(419)*z(4209) - 0.00893*z(429)*z(4210) - 0.00893*z(439)*z(4211) - 0.00893*z(2732)*z(3279) - 0.00893*z(2735)*z(3297) - 0.00893*z(2738)*z(3314);
z(4283) = -0.01653*z(419)*z(4209) - 0.01653*z(429)*z(4210) - 0.01653*z(439)*z(4211) - 0.01653*z(2732)*z(3279) - 0.01653*z(2735)*z(3297) - 0.01653*z(2738)*z(3314) - 0.0005*z(419)*z(4213) - 0.0005*z(429)*z(4214) - 0.0005*z(439)*z(4215) - 0.0005*z(2730)*z(3279) - 0.0005*z(2733)*z(3297) - 0.0005*z(2736)*z(3314);
z(4284) = 0.00893*z(419)*z(4213) + 0.00893*z(429)*z(4214) + 0.00893*z(439)*z(4215) + 0.00893*z(2730)*z(3279) + 0.00893*z(2733)*z(3297) + 0.00893*z(2736)*z(3314) + 0.01653*z(419)*z(4206) + 0.01653*z(429)*z(4207) + 0.01653*z(439)*z(4208) + 0.01653*z(2731)*z(3279) + 0.01653*z(2734)*z(3297) + 0.01653*z(2737)*z(3314);
z(4187) = (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(199)*z(419) + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(203)*z(429) + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(204)*z(439) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(197)*z(419) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(201)*z(429) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(202)*z(439) + z(197)*z(3279)*z(2611) + z(201)*z(3297)*z(2611) + z(202)*z(3314)*z(2611) + z(419)*z(234)*z(2611) + z(429)*z(217)*z(2611) + z(439)*z(230)*z(2611) - z(199)*z(3279)*z(2610) - z(203)*z(3297)*z(2610) - z(204)*z(3314)*z(2610) - z(419)*z(235)*z(2610) - z(429)*z(219)*z(2610) - z(439)*z(229)*z(2610);
z(3616) = z(146)*z(564) - z(149)*z(563);
z(3617) = z(149)*z(562) - z(138)*z(564);
z(3618) = z(138)*z(563) - z(146)*z(562);
z(3619) = z(129)*z(3616) + z(132)*z(3617) + z(135)*z(3618) + z(160)*z(566) + z(161)*z(574) + z(162)*z(570);
z(3688) = z(3182)*z(788) - z(3193)*z(787);
z(3634) = z(130)*z(3616) + z(133)*z(3617) + z(136)*z(3618) + z(145)*z(574) + z(148)*z(570) + z(151)*z(566);
z(3689) = z(3193)*z(786) - z(3170)*z(788);
z(3646) = z(131)*z(3616) + z(134)*z(3617) + z(137)*z(3618) + z(155)*z(570) + z(156)*z(566) + z(157)*z(574);
z(3690) = z(3170)*z(787) - z(3182)*z(786);
z(3691) = z(187)*z(819) + z(188)*z(811) + z(189)*z(815) + z(166)*(z(3619)+z(3688)) + z(168)*(z(3634)+z(3689)) + z(170)*(z(3646)+z(3690));
z(3796) = z(3232)*z(1228) - z(3244)*z(1227);
z(3714) = z(179)*z(811) + z(181)*z(815) + z(183)*z(819) + z(172)*(z(3619)+z(3688)) + z(174)*(z(3634)+z(3689)) + z(176)*(z(3646)+z(3690));
z(3797) = z(3244)*z(1226) - z(3220)*z(1228);
z(3736) = z(184)*z(815) + z(185)*z(819) + z(186)*z(811) + z(173)*(z(3619)+z(3688)) + z(175)*(z(3634)+z(3689)) + z(177)*(z(3646)+z(3690));
z(3798) = z(3220)*z(1227) - z(3232)*z(1226);
z(3799) = z(236)*z(1277) + z(237)*z(1263) + z(238)*z(1270) + z(207)*(z(3691)+z(3796)) + z(210)*(z(3714)+z(3797)) + z(213)*(z(3736)+z(3798));
z(3937) = z(3297)*z(1884) - z(3314)*z(1883);
z(3832) = z(220)*z(1263) + z(224)*z(1270) + z(227)*z(1277) + z(208)*(z(3691)+z(3796)) + z(211)*(z(3714)+z(3797)) + z(214)*(z(3736)+z(3798));
z(3938) = z(3314)*z(1882) - z(3279)*z(1884);
z(3862) = z(231)*z(1270) + z(232)*z(1277) + z(233)*z(1263) + z(209)*(z(3691)+z(3796)) + z(212)*(z(3714)+z(3797)) + z(215)*(z(3736)+z(3798));
z(3939) = z(3279)*z(1883) - z(3297)*z(1882);
z(3940) = z(277)*z(1951) + z(278)*z(1931) + z(279)*z(1941) + z(252)*(z(3799)+z(3937)) + z(255)*(z(3832)+z(3938)) + z(258)*(z(3862)+z(3939));
z(4062) = z(253)*z(3279)*z(2531) + z(256)*z(3297)*z(2531) + z(259)*z(3314)*z(2531) + z(419)*z(265)*z(2531) + z(429)*z(268)*z(2531) + z(439)*z(271)*z(2531) - z(254)*z(3279)*z(2530) - z(257)*z(3297)*z(2530) - z(260)*z(3314)*z(2530) - z(419)*z(276)*z(2530) - z(429)*z(274)*z(2530) - z(439)*z(275)*z(2530);

z(4063) = z(254)*z(3279)*HOHK2 + z(257)*z(3297)*HOHK2 + z(260)*z(3314)*HOHK2 + z(419)*z(276)*HOHK2 + z(429)*z(274)*HOHK2 +
z(439)*z(275)*HOHK2 - z(253)*z(3279)*HOHK3 - z(256)*z(3297)*HOHK3 - z(259)*z(3314)*HOHK3 - z(419)*z(265)*HOHK3 -
z(429)*z(268)*HOHK3 - z(439)*z(271)*HOHK3;
z(4064) = z(3940) + z(4062) + z(4063);
z(3979) = z(265)*z(1931) + z(268)*z(1941) + z(271)*z(1951) + z(253)*(z(3799)+z(3937)) + z(256)*(z(3832)+z(3938)) + z(259)*(z(3862)+z(3939));
z(4102) = z(254)*z(3279)*z(2529) + z(257)*z(3297)*z(2529) + z(260)*z(3314)*z(2529) + z(419)*z(276)*z(2529) + z(429)*z(274)*z(2529) +
z(439)*z(275)*z(2529) - z(252)*z(3279)*z(2531) - z(255)*z(3297)*z(2531) - z(258)*z(3314)*z(2531) - z(419)*z(278)*z(2531) - z(429)*z(279)*z(2531)
- z(439)*z(277)*z(2531);
z(4103) = z(252)*z(3279)*HOHK3 + z(255)*z(3297)*HOHK3 + z(258)*z(3314)*HOHK3 + z(419)*z(278)*HOHK3 + z(429)*z(279)*HOHK3 +
z(439)*z(277)*HOHK3 - z(254)*z(3279)*HOHK1 - z(257)*z(3297)*HOHK1 - z(260)*z(3314)*HOHK1 - z(419)*z(276)*HOHK1 -
z(429)*z(274)*HOHK1 - z(439)*z(275)*HOHK1;
z(4104) = z(3979) + z(4102) + z(4103);
z(4015) = z(274)*z(1941) + z(275)*z(1951) + z(276)*z(1931) + z(254)*(z(3799)+z(3937)) + z(257)*(z(3832)+z(3938)) + z(260)*(z(3862)+z(3939));
z(4142) = z(252)*z(3279)*z(2530) + z(255)*z(3297)*z(2530) + z(258)*z(3314)*z(2530) + z(419)*z(278)*z(2530) + z(429)*z(279)*z(2530) +
z(439)*z(277)*z(2530) - z(253)*z(3279)*z(2529) - z(256)*z(3297)*z(2529) - z(259)*z(3314)*z(2529) - z(419)*z(265)*z(2529) - z(429)*z(268)*z(2529)
- z(439)*z(271)*z(2529);
z(4143) = z(253)*z(3279)*HOHK1 + z(256)*z(3297)*HOHK1 + z(259)*z(3314)*HOHK1 + z(419)*z(265)*HOHK1 + z(429)*z(268)*HOHK1 +
z(439)*z(271)*HOHK1 - z(252)*z(3279)*HOHK2 - z(255)*z(3297)*HOHK2 - z(258)*z(3314)*HOHK2 - z(419)*z(278)*HOHK2 -
z(429)*z(279)*HOHK2 - z(439)*z(277)*HOHK2;
z(4144) = z(4015) + z(4142) + z(4143);
z(4159) = 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(28)*z(419) + z(28)*z(3279)*z(2611) +
z(199)*z(3279)*z(2612) + z(203)*z(3297)*z(2612) + z(204)*z(3314)*z(2612) + z(419)*z(235)*z(2612) + z(429)*z(219)*z(2612) +
z(439)*z(229)*z(2612) + z(27)*z(419)*QNE_2p*z(2611) - 4.049999999999887E-05*(-1+8.854320987654567*T-
1.955555555555609*T^2)*z(205)*z(429) - 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(206)*z(439) -
z(205)*z(3297)*z(2611) - z(206)*z(3314)*z(2611) - z(429)*z(221)*z(2611) - z(439)*z(228)*z(2611);
z(4173) = (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(28)*z(419) + z(205)*z(3297)*z(2610) + z(206)*z(3314)*z(2610) +
z(429)*z(221)*z(2610) + z(439)*z(228)*z(2610) - (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(205)*z(429) -
(0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(206)*z(439) - z(28)*z(3279)*z(2610) - z(197)*z(3279)*z(2612) -
z(201)*z(3297)*z(2612) - z(202)*z(3314)*z(2612) - z(419)*z(234)*z(2612) - z(429)*z(217)*z(2612) - z(439)*z(230)*z(2612) -
z(27)*z(419)*QNE_2p*z(2610);
z(4324) = z(347)*z(4282) + z(350)*z(4283) + z(353)*z(4284) + z(2748)*z(365) + z(2761)*z(362) + z(2774)*z(359) + z(344)*z(4187) +
z(2779)*z(4064) + z(2782)*z(4104) + z(2785)*z(4144) + z(2788)*z(4159) + z(2791)*z(4173) + z(358)*z(2661) + z(4297)*z(2701) + z(4298)*z(2714) +
z(4299)*z(2727) + z(4300)*z(2687) + z(4301)*z(2674);
z(4287) = 0.0005*z(420)*z(4206) + 0.0005*z(430)*z(4207) + 0.0005*z(440)*z(4208) + 0.0005*z(2731)*z(3280) + 0.0005*z(2734)*z(3298) +
0.0005*z(2737)*z(3315) - 0.00893*z(420)*z(4209) - 0.00893*z(430)*z(4210) - 0.00893*z(440)*z(4211) - 0.00893*z(2732)*z(3280) -
0.00893*z(2735)*z(3298) - 0.00893*z(2738)*z(3315);
z(4288) = -0.01653*z(420)*z(4209) - 0.01653*z(430)*z(4210) - 0.01653*z(440)*z(4211) - 0.01653*z(2732)*z(3280) - 0.01653*z(2735)*z(3298) -
0.01653*z(2738)*z(3315) - 0.0005*z(420)*z(4213) - 0.0005*z(430)*z(4214) - 0.0005*z(440)*z(4215) - 0.0005*z(2730)*z(3280) -
0.0005*z(2733)*z(3298) - 0.0005*z(2736)*z(3315);
z(4289) = 0.00893*z(420)*z(4213) + 0.00893*z(430)*z(4214) + 0.00893*z(440)*z(4215) + 0.00893*z(2730)*z(3280) + 0.00893*z(2733)*z(3298) +
0.00893*z(2736)*z(3315) + 0.01653*z(420)*z(4206) + 0.01653*z(430)*z(4207) + 0.01653*z(440)*z(4208) + 0.01653*z(2731)*z(3280) +
0.01653*z(2734)*z(3298) + 0.01653*z(2737)*z(3315);
z(2749) = 0.00893*z(420)*z(2730) + 0.00893*z(430)*z(2733) + 0.00893*z(440)*z(2736) + 0.01653*z(420)*z(2731) + 0.01653*z(430)*z(2734) +
0.01653*z(440)*z(2737);
z(2762) = -0.01653*z(420)*z(2732) - 0.01653*z(430)*z(2735) - 0.01653*z(440)*z(2738) - 0.0005*z(420)*z(2730) - 0.0005*z(430)*z(2733) -
0.0005*z(440)*z(2736);
z(2775) = 0.0005*z(420)*z(2731) + 0.0005*z(430)*z(2734) + 0.0005*z(440)*z(2737) - 0.00893*z(420)*z(2732) - 0.00893*z(430)*z(2735) -
0.00893*z(440)*z(2738);
z(4188) = (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(199)*z(420) + (0.001249+5.719999999999987E-05*T-
5.699999999999868E-06*T^2)*z(203)*z(430) + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(204)*z(440) +
4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(197)*z(420) + 4.049999999999887E-05*(-
1+8.854320987654567*T-1.955555555555609*T^2)*z(201)*z(430) + 4.049999999999887E-05*(-1+8.854320987654567*T-
1.955555555555609*T^2)*z(202)*z(440) + z(197)*z(3280)*z(2611) + z(201)*z(3298)*z(2611) + z(202)*z(3315)*z(2611) + z(420)*z(234)*z(2611) +
z(430)*z(217)*z(2611) + z(440)*z(230)*z(2611) - z(199)*z(3280)*z(2610) - z(203)*z(3298)*z(2610) - z(204)*z(3315)*z(2610) -
z(420)*z(235)*z(2610) - z(430)*z(219)*z(2610) - z(440)*z(229)*z(2610);
z(575) = z(116)*z(564) - z(118)*z(563);
z(571) = z(118)*z(562) - z(114)*z(564);
z(567) = z(114)*z(563) - z(116)*z(562);
z(743) = z(129)*z(575) + z(132)*z(571) + z(135)*z(567);
z(808) = z(386)*z(788) - z(390)*z(787);
z(812) = z(743) + z(808);
z(747) = z(130)*z(575) + z(133)*z(571) + z(136)*z(567);
z(801) = z(390)*z(786) - z(382)*z(788);
z(816) = z(747) + z(801);
z(751) = z(131)*z(575) + z(134)*z(571) + z(137)*z(567);
z(794) = z(382)*z(787) - z(386)*z(786);
z(820) = z(751) + z(794);
z(1153) = z(166)*z(812) + z(168)*z(816) + z(170)*z(820);
z(1257) = z(404)*z(1228) - z(411)*z(1227);
z(1264) = z(1153) + z(1257);
z(1163) = z(172)*z(812) + z(174)*z(816) + z(176)*z(820);
z(1246) = z(411)*z(1226) - z(397)*z(1228);
z(1271) = z(1163) + z(1246);
z(1173) = z(173)*z(812) + z(175)*z(816) + z(177)*z(820);
z(1236) = z(397)*z(1227) - z(404)*z(1226);
z(1278) = z(1173) + z(1236);
z(1811) = z(209)*z(1264) + z(212)*z(1271) + z(215)*z(1278);
z(1895) = z(420)*z(1883) - z(430)*z(1882);

```
z(1952) = z(1811) + z(1895);
z(1779) = z(207)*z(1264) + z(210)*z(1271) + z(213)*z(1278);
z(1921) = z(430)*z(1884) - z(440)*z(1883);
z(1932) = z(1779) + z(1921);
z(1795) = z(208)*z(1264) + z(211)*z(1271) + z(214)*z(1278);
z(1908) = z(440)*z(1882) - z(420)*z(1884);
z(1942) = z(1795) + z(1908);
z(3621) = z(147)*z(564) - z(150)*z(563);
z(3622) = z(150)*z(562) - z(141)*z(564);
z(3623) = z(141)*z(563) - z(147)*z(562);
z(3624) = z(129)*z(3621) + z(132)*z(3622) + z(135)*z(3623) + z(160)*z(567) + z(161)*z(575) + z(162)*z(571);
z(3693) = z(3183)*z(788) - z(3194)*z(787);
z(3636) = z(130)*z(3621) + z(133)*z(3622) + z(136)*z(3623) + z(145)*z(575) + z(148)*z(571) + z(151)*z(567);
z(3694) = z(3194)*z(786) - z(3171)*z(788);
z(3648) = z(131)*z(3621) + z(134)*z(3622) + z(137)*z(3623) + z(155)*z(571) + z(156)*z(567) + z(157)*z(575);
z(3695) = z(3171)*z(787) - z(3183)*z(786);
z(3696) = z(187)*z(820) + z(188)*z(812) + z(189)*z(816) + z(166)*(z(3624)+z(3693)) + z(168)*(z(3636)+z(3694)) + z(170)*(z(3648)+z(3695));
z(3801) = z(3233)*z(1228) - z(3245)*z(1227);
z(3716) = z(179)*z(812) + z(181)*z(816) + z(183)*z(820) + z(172)*(z(3624)+z(3693)) + z(174)*(z(3636)+z(3694)) + z(176)*(z(3648)+z(3695));
z(3802) = z(3245)*z(1226) - z(3221)*z(1228);
z(3738) = z(184)*z(816) + z(185)*z(820) + z(186)*z(812) + z(173)*(z(3624)+z(3693)) + z(175)*(z(3636)+z(3694)) + z(177)*(z(3648)+z(3695));
z(3803) = z(3221)*z(1227) - z(3233)*z(1226);
z(3804) = z(236)*z(1278) + z(237)*z(1264) + z(238)*z(1271) + z(207)*(z(3696)+z(3801)) + z(210)*(z(3716)+z(3802)) + z(213)*(z(3738)+z(3803));
z(3942) = z(3298)*z(1884) - z(3315)*z(1883);
z(3834) = z(220)*z(1264) + z(224)*z(1271) + z(227)*z(1278) + z(208)*(z(3696)+z(3801)) + z(211)*(z(3716)+z(3802)) + z(214)*(z(3738)+z(3803));
z(3943) = z(3315)*z(1882) - z(3280)*z(1884);
z(3864) = z(231)*z(1271) + z(232)*z(1278) + z(233)*z(1264) + z(209)*(z(3696)+z(3801)) + z(212)*(z(3716)+z(3802)) + z(215)*(z(3738)+z(3803));
z(3944) = z(3280)*z(1883) - z(3298)*z(1882);
z(3945) = z(277)*z(1952) + z(278)*z(1932) + z(279)*z(1942) + z(252)*(z(3804)+z(3942)) + z(255)*(z(3834)+z(3943)) + z(258)*(z(3864)+z(3944));
z(4065) = z(253)*z(3280)*z(2531) + z(256)*z(3298)*z(2531) + z(259)*z(3315)*z(2531) + z(420)*z(265)*z(2531) + z(430)*z(268)*z(2531) +
z(440)*z(271)*z(2531) - z(254)*z(3280)*z(2530) - z(257)*z(3298)*z(2530) - z(260)*z(3315)*z(2530) - z(420)*z(276)*z(2530) - z(430)*z(274)*z(2530)
- z(440)*z(275)*z(2530);
z(4066) = z(254)*z(3280)*HOHK2 + z(257)*z(3298)*HOHK2 + z(260)*z(3315)*HOHK2 + z(420)*z(276)*HOHK2 + z(430)*z(274)*HOHK2 +
z(440)*z(275)*HOHK2 - z(253)*z(3280)*HOHK3 - z(256)*z(3298)*HOHK3 - z(259)*z(3315)*HOHK3 - z(420)*z(265)*HOHK3 -
z(430)*z(268)*HOHK3 - z(440)*z(271)*HOHK3;
z(4067) = z(3945) + z(4065) + z(4066);
z(3981) = z(265)*z(1932) + z(268)*z(1942) + z(271)*z(1952) + z(253)*(z(3804)+z(3942)) + z(256)*(z(3834)+z(3943)) + z(259)*(z(3864)+z(3944));
z(4105) = z(254)*z(3280)*z(2529) + z(257)*z(3298)*z(2529) + z(260)*z(3315)*z(2529) + z(420)*z(276)*z(2529) + z(430)*z(274)*z(2529) +
z(440)*z(275)*z(2529) - z(252)*z(3280)*z(2531) - z(255)*z(3298)*z(2531) - z(258)*z(3315)*z(2531) - z(420)*z(278)*z(2531) - z(430)*z(279)*z(2531)
- z(440)*z(277)*z(2531);
z(4106) = z(252)*z(3280)*HOHK3 + z(255)*z(3298)*HOHK3 + z(258)*z(3315)*HOHK3 + z(420)*z(278)*HOHK3 + z(430)*z(279)*HOHK3 +
z(440)*z(277)*HOHK3 - z(254)*z(3280)*HOHK1 - z(257)*z(3298)*HOHK1 - z(260)*z(3315)*HOHK1 - z(420)*z(276)*HOHK1 -
z(430)*z(274)*HOHK1 - z(440)*z(275)*HOHK1;
z(4107) = z(3981) + z(4105) + z(4106);
z(4017) = z(274)*z(1942) + z(275)*z(1952) + z(276)*z(1932) + z(254)*(z(3804)+z(3942)) + z(257)*(z(3834)+z(3943)) + z(260)*(z(3864)+z(3944));
z(4145) = z(252)*z(3280)*z(2530) + z(255)*z(3298)*z(2530) + z(258)*z(3315)*z(2530) + z(420)*z(278)*z(2530) + z(430)*z(279)*z(2530) +
z(440)*z(277)*z(2530) - z(253)*z(3280)*z(2529) - z(256)*z(3298)*z(2529) - z(259)*z(3315)*z(2529) - z(420)*z(265)*z(2529) - z(430)*z(268)*z(2529)
- z(440)*z(271)*z(2529);
z(4146) = z(253)*z(3280)*HOHK1 + z(256)*z(3298)*HOHK1 + z(259)*z(3315)*HOHK1 + z(420)*z(265)*HOHK1 + z(430)*z(268)*HOHK1 +
z(440)*z(271)*HOHK1 - z(252)*z(3280)*HOHK2 - z(255)*z(3298)*HOHK2 - z(258)*z(3315)*HOHK2 - z(420)*z(278)*HOHK2 -
z(430)*z(279)*HOHK2 - z(440)*z(277)*HOHK2;
z(4147) = z(4017) + z(4145) + z(4146);
z(4160) = 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(28)*z(420) + z(28)*z(3280)*z(2611) +
z(199)*z(3280)*z(2612) + z(203)*z(3298)*z(2612) + z(204)*z(3315)*z(2612) + z(420)*z(235)*z(2612) + z(430)*z(219)*z(2612) +
z(440)*z(229)*z(2612) + z(27)*z(420)*QNE_2p*z(2611) - 4.049999999999887E-05*(-1+8.854320987654567*T-
1.955555555555609*T^2)*z(205)*z(430) - 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(206)*z(440) -
z(205)*z(3298)*z(2611) - z(206)*z(3315)*z(2611) - z(430)*z(221)*z(2611) - z(440)*z(228)*z(2611);
z(4174) = (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(28)*z(420) + z(205)*z(3298)*z(2610) + z(206)*z(3315)*z(2610) +
z(430)*z(221)*z(2610) + z(440)*z(228)*z(2610) - (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(205)*z(430) -
(0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(206)*z(440) - z(28)*z(3280)*z(2610) - z(197)*z(3280)*z(2612) -
z(201)*z(3298)*z(2612) - z(202)*z(3315)*z(2612) - z(420)*z(234)*z(2612) - z(430)*z(217)*z(2612) - z(440)*z(230)*z(2612) -
z(27)*z(420)*QNE_2p*z(2610);
z(2662) = z(197)*z(420)*z(2611) + z(201)*z(430)*z(2611) + z(202)*z(440)*z(2611) - z(199)*z(420)*z(2610) - z(203)*z(430)*z(2610) -
z(204)*z(440)*z(2610);
z(2405) = z(252)*z(1932) + z(255)*z(1942) + z(258)*z(1952);
z(2568) = z(253)*z(420)*z(2531) + z(256)*z(430)*z(2531) + z(259)*z(440)*z(2531) - z(254)*z(420)*z(2530) - z(257)*z(430)*z(2530) -
z(260)*z(440)*z(2530);
z(2582) = z(2405) + z(2568);
z(2649) = z(254)*z(420)*HOHK2 + z(257)*z(430)*HOHK2 + z(260)*z(440)*HOHK2 - z(253)*z(420)*HOHK3 - z(256)*z(430)*HOHK3 -
z(259)*z(440)*HOHK3;
z(2702) = z(2582) + z(2649);
z(2427) = z(253)*z(1932) + z(256)*z(1942) + z(259)*z(1952);
z(2555) = z(254)*z(420)*z(2529) + z(257)*z(430)*z(2529) + z(260)*z(440)*z(2529) - z(252)*z(420)*z(2531) - z(255)*z(430)*z(2531) -
z(258)*z(440)*z(2531);
z(2595) = z(2427) + z(2555);
z(2636) = z(252)*z(420)*HOHK3 + z(255)*z(430)*HOHK3 + z(258)*z(440)*HOHK3 - z(254)*z(420)*HOHK1 - z(257)*z(430)*HOHK1 -
z(260)*z(440)*HOHK1;
z(2715) = z(2595) + z(2636);
```

z(2449) = z(254)*z(1932) + z(257)*z(1942) + z(260)*z(1952);
z(2542) = z(252)*z(420)*z(2530) + z(255)*z(430)*z(2530) + z(258)*z(440)*z(2530) - z(253)*z(420)*z(2529) - z(256)*z(430)*z(2529) - z(259)*z(440)*z(2529);
z(2608) = z(2449) + z(2542);
z(2623) = z(253)*z(420)*HOHK1 + z(256)*z(430)*HOHK1 + z(259)*z(440)*HOHK1 - z(252)*z(420)*HOHK2 - z(255)*z(430)*HOHK2 - z(258)*z(440)*HOHK2;
z(2728) = z(2608) + z(2623);
z(2688) = z(28)*z(420)*z(2611) + z(199)*z(420)*z(2612) + z(203)*z(430)*z(2612) + z(204)*z(440)*z(2612) - z(205)*z(430)*z(2611) - z(206)*z(440)*z(2611);
z(2675) = z(205)*z(430)*z(2610) + z(206)*z(440)*z(2610) - z(28)*z(420)*z(2610) - z(197)*z(420)*z(2612) - z(201)*z(430)*z(2612) - z(202)*z(440)*z(2612);
z(4325) = z(347)*z(4287) + z(350)*z(4288) + z(353)*z(4289) + z(2749)*z(365) + z(2762)*z(362) + z(2775)*z(359) + z(344)*z(4188) + z(2779)*z(4067) + z(2782)*z(4107) + z(2785)*z(4147) + z(2788)*z(4160) + z(2791)*z(4174) + z(358)*z(2662) + z(4297)*z(2702) + z(4298)*z(2715) + z(4299)*z(2728) + z(4300)*z(2688) + z(4301)*z(2675);
z(4292) = 0.0005*z(421)*z(4206) + 0.0005*z(431)*z(4207) + 0.0005*z(441)*z(4208) + 0.0005*z(2731)*z(3281) + 0.0005*z(2734)*z(3299) + 0.0005*z(2737)*z(3316) - 0.00893*z(421)*z(4209) - 0.00893*z(431)*z(4210) - 0.00893*z(441)*z(4211) - 0.00893*z(2732)*z(3281) - 0.00893*z(2735)*z(3299) - 0.00893*z(2738)*z(3316);
z(4293) = -0.01653*z(421)*z(4209) - 0.01653*z(431)*z(4210) - 0.01653*z(441)*z(4211) - 0.01653*z(2732)*z(3281) - 0.01653*z(2735)*z(3299) - 0.01653*z(2738)*z(3316) - 0.0005*z(421)*z(4213) - 0.0005*z(431)*z(4214) - 0.0005*z(441)*z(4215) - 0.0005*z(2730)*z(3281) - 0.0005*z(2733)*z(3299) - 0.0005*z(2736)*z(3316);
z(4294) = 0.00893*z(421)*z(4213) + 0.00893*z(431)*z(4214) + 0.00893*z(441)*z(4215) + 0.00893*z(2730)*z(3281) + 0.00893*z(2733)*z(3299) + 0.00893*z(2736)*z(3316) + 0.01653*z(421)*z(4206) + 0.01653*z(431)*z(4207) + 0.01653*z(441)*z(4208) + 0.01653*z(2731)*z(3281) + 0.01653*z(2734)*z(3299) + 0.01653*z(2737)*z(3316);
z(2750) = 0.00893*z(421)*z(2730) + 0.00893*z(431)*z(2733) + 0.00893*z(441)*z(2736) + 0.01653*z(421)*z(2731) + 0.01653*z(431)*z(2734) + 0.01653*z(441)*z(2737);
z(2763) = -0.01653*z(421)*z(2732) - 0.01653*z(431)*z(2735) - 0.01653*z(441)*z(2738) - 0.0005*z(421)*z(2730) - 0.0005*z(431)*z(2733) - 0.0005*z(441)*z(2736);
z(2776) = 0.0005*z(421)*z(2731) + 0.0005*z(431)*z(2734) + 0.0005*z(441)*z(2737) - 0.00893*z(421)*z(2732) - 0.00893*z(431)*z(2735) - 0.00893*z(441)*z(2738);
z(4189) = (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(199)*z(421) + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(203)*z(431) + (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(204)*z(441) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(197)*z(421) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(201)*z(431) + 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(202)*z(441) + z(197)*z(3281)*z(2611) + z(201)*z(3299)*z(2611) + z(202)*z(3316)*z(2611) + z(421)*z(234)*z(2611) + z(431)*z(217)*z(2611) + z(441)*z(230)*z(2611) - z(199)*z(3281)*z(2610) - z(203)*z(3299)*z(2610) - z(204)*z(3316)*z(2610) - z(421)*z(235)*z(2610) - z(431)*z(219)*z(2610) - z(441)*z(229)*z(2610);
z(1250) = z(24)*z(1228) - z(23)*z(1227);
z(1247) = z(23)*z(1226);
z(1237) = z(24)*z(1226);
z(1804) = z(209)*z(1250) + z(212)*z(1247) - z(215)*z(1237);
z(1896) = z(421)*z(1883) - z(431)*z(1882);
z(1945) = z(1804) + z(1896);
z(1772) = z(207)*z(1250) + z(210)*z(1247) - z(213)*z(1237);
z(1922) = z(431)*z(1884) - z(441)*z(1883);
z(1925) = z(1772) + z(1922);
z(1788) = z(208)*z(1250) + z(211)*z(1247) - z(214)*z(1237);
z(1909) = z(441)*z(1882) - z(421)*z(1884);
z(1935) = z(1788) + z(1909);
z(3766) = QND_1p*(z(23)*z(1228)+z(24)*z(1227));
z(3767) = z(24)*QND_1p*z(1226);
z(3768) = z(23)*QND_1p*z(1226);
z(3769) = z(207)*z(3766) + z(237)*z(1250) + z(238)*z(1247) - z(210)*z(3767) - z(213)*z(3768) - z(236)*z(1237);
z(3907) = z(3299)*z(1884) - z(3316)*z(1883);
z(3820) = z(208)*z(3766) + z(220)*z(1250) + z(224)*z(1247) - z(211)*z(3767) - z(214)*z(3768) - z(227)*z(1237);
z(3908) = z(3316)*z(1882) - z(3281)*z(1884);
z(3850) = z(209)*z(3766) + z(231)*z(1247) + z(233)*z(1250) - z(212)*z(3767) - z(215)*z(3768) - z(232)*z(1237);
z(3909) = z(3281)*z(1883) - z(3299)*z(1882);
z(3910) = z(277)*z(1945) + z(278)*z(1925) + z(279)*z(1935) + z(252)*(z(3769)+z(3907)) + z(255)*(z(3820)+z(3908)) + z(258)*(z(3850)+z(3909));
z(4044) = z(253)*z(3281)*z(2531) + z(256)*z(3299)*z(2531) + z(259)*z(3316)*z(2531) + z(421)*z(265)*z(2531) + z(431)*z(268)*z(2531) + z(441)*z(271)*z(2531) - z(254)*z(3281)*z(2530) - z(257)*z(3299)*z(2530) - z(260)*z(3316)*z(2530) - z(421)*z(276)*z(2530) - z(431)*z(274)*z(2530) - z(441)*z(275)*z(2530);
z(4045) = z(254)*z(3281)*HOHK2 + z(257)*z(3299)*HOHK2 + z(260)*z(3316)*HOHK2 + z(421)*z(276)*HOHK2 + z(431)*z(274)*HOHK2 + z(441)*z(275)*HOHK2 - z(253)*z(3281)*HOHK3 - z(256)*z(3299)*HOHK3 - z(259)*z(3316)*HOHK3 - z(421)*z(265)*HOHK3 - z(431)*z(268)*HOHK3 - z(441)*z(271)*HOHK3;
z(4046) = z(3910) + z(4044) + z(4045);
z(3967) = z(265)*z(1925) + z(268)*z(1935) + z(271)*z(1945) + z(253)*(z(3769)+z(3907)) + z(256)*(z(3820)+z(3908)) + z(259)*(z(3850)+z(3909));
z(4084) = z(254)*z(3281)*z(2529) + z(257)*z(3299)*z(2529) + z(260)*z(3316)*z(2529) + z(421)*z(276)*z(2529) + z(431)*z(274)*z(2529) + z(441)*z(275)*z(2529) - z(252)*z(3281)*z(2531) - z(255)*z(3299)*z(2531) - z(258)*z(3316)*z(2531) - z(421)*z(278)*z(2531) - z(431)*z(279)*z(2531) - z(441)*z(277)*z(2531);
z(4085) = z(252)*z(3281)*HOHK3 + z(255)*z(3299)*HOHK3 + z(258)*z(3316)*HOHK3 + z(421)*z(278)*HOHK3 + z(431)*z(279)*HOHK3 + z(441)*z(277)*HOHK3 - z(254)*z(3281)*HOHK1 - z(257)*z(3299)*HOHK1 - z(260)*z(3316)*HOHK1 - z(421)*z(276)*HOHK1 - z(431)*z(274)*HOHK1 - z(441)*z(275)*HOHK1;
z(4086) = z(3967) + z(4084) + z(4085);
z(4003) = z(274)*z(1935) + z(275)*z(1945) + z(276)*z(1925) + z(254)*(z(3769)+z(3907)) + z(257)*(z(3820)+z(3908)) + z(260)*(z(3850)+z(3909));
z(4124) = z(252)*z(3281)*z(2530) + z(255)*z(3299)*z(2530) + z(258)*z(3316)*z(2530) + z(421)*z(278)*z(2530) + z(431)*z(279)*z(2530) + z(441)*z(277)*z(2530) - z(253)*z(3281)*z(2529) - z(256)*z(3299)*z(2529) - z(259)*z(3316)*z(2529) - z(421)*z(265)*z(2529) - z(431)*z(268)*z(2529) - z(441)*z(271)*z(2529);

z(4125) = z(253)*z(3281)*HOHK1 + z(256)*z(3299)*HOHK1 + z(259)*z(3316)*HOHK1 + z(421)*z(265)*HOHK1 + z(431)*z(268)*HOHK1 +
z(441)*z(271)*HOHK1 - z(252)*z(3281)*HOHK2 - z(255)*z(3299)*HOHK2 - z(258)*z(3316)*HOHK2 - z(421)*z(278)*HOHK2 -
z(431)*z(279)*HOHK2 - z(441)*z(277)*HOHK2;
z(4126) = z(4003) + z(4124) + z(4125);
z(4161) = 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(28)*z(421) + z(28)*z(3281)*z(2611) +
z(199)*z(3281)*z(2612) + z(203)*z(3299)*z(2612) + z(204)*z(3316)*z(2612) + z(421)*z(235)*z(2612) + z(431)*z(219)*z(2612) +
z(441)*z(229)*z(2612) + z(27)*z(421)*QNE_2p*z(2611) - 4.049999999999887E-05*(-1+8.854320987654567*T-
1.955555555555609*T^2)*z(205)*z(431) - 4.049999999999887E-05*(-1+8.854320987654567*T-1.955555555555609*T^2)*z(206)*z(441) -
z(205)*z(3299)*z(2611) - z(206)*z(3316)*z(2611) - z(431)*z(221)*z(2611) - z(441)*z(228)*z(2611);
z(4175) = (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(28)*z(421) + z(205)*z(3299)*z(2610) + z(206)*z(3316)*z(2610) +
z(431)*z(221)*z(2610) + z(441)*z(228)*z(2610) - (0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(205)*z(431) -
(0.001249+5.719999999999987E-05*T-5.699999999999868E-06*T^2)*z(206)*z(441) - z(28)*z(3281)*z(2610) - z(197)*z(3281)*z(2612) -
z(201)*z(3299)*z(2612) - z(202)*z(3316)*z(2612) - z(421)*z(234)*z(2612) - z(431)*z(217)*z(2612) - z(441)*z(230)*z(2612) -
z(27)*z(421)*QNE_2p*z(2610);
z(2663) = z(197)*z(421)*z(2611) + z(201)*z(431)*z(2611) + z(202)*z(441)*z(2611) - z(199)*z(421)*z(2610) - z(203)*z(431)*z(2610) -
z(204)*z(441)*z(2610);
z(2398) = z(252)*z(1925) + z(255)*z(1935) + z(258)*z(1945);
z(2569) = z(253)*z(421)*z(2531) + z(256)*z(431)*z(2531) + z(259)*z(441)*z(2531) - z(254)*z(421)*z(2530) - z(257)*z(431)*z(2530) -
z(260)*z(441)*z(2530);
z(2575) = z(2398) + z(2569);
z(2650) = z(254)*z(421)*HOHK2 + z(257)*z(431)*HOHK2 + z(260)*z(441)*HOHK2 - z(253)*z(421)*HOHK3 - z(256)*z(431)*HOHK3 -
z(259)*z(441)*HOHK3;
z(2695) = z(2575) + z(2650);
z(2420) = z(253)*z(1925) + z(256)*z(1935) + z(259)*z(1945);
z(2556) = z(254)*z(421)*z(2529) + z(257)*z(431)*z(2529) + z(260)*z(441)*z(2529) - z(252)*z(421)*z(2531) - z(255)*z(431)*z(2531) -
z(258)*z(441)*z(2531);
z(2588) = z(2420) + z(2556);
z(2637) = z(252)*z(421)*HOHK3 + z(255)*z(431)*HOHK3 + z(258)*z(441)*HOHK3 - z(254)*z(421)*HOHK1 - z(257)*z(431)*HOHK1 -
z(260)*z(441)*HOHK1;
z(2708) = z(2588) + z(2637);
z(2442) = z(254)*z(1925) + z(257)*z(1935) + z(260)*z(1945);
z(2543) = z(252)*z(421)*z(2530) + z(255)*z(431)*z(2530) + z(258)*z(441)*z(2530) - z(253)*z(421)*z(2529) - z(256)*z(431)*z(2529) -
z(259)*z(441)*z(2529);
z(2601) = z(2442) + z(2543);
z(2624) = z(253)*z(421)*HOHK1 + z(256)*z(431)*HOHK1 + z(259)*z(441)*HOHK1 - z(252)*z(421)*HOHK2 - z(255)*z(431)*HOHK2 -
z(258)*z(441)*HOHK2;
z(2721) = z(2601) + z(2624);
z(2689) = z(28)*z(421)*z(2611) + z(199)*z(421)*z(2612) + z(203)*z(431)*z(2612) + z(204)*z(441)*z(2612) - z(205)*z(431)*z(2611) -
z(206)*z(441)*z(2611);
z(2676) = z(205)*z(431)*z(2610) + z(206)*z(441)*z(2610) - z(28)*z(421)*z(2610) - z(197)*z(421)*z(2612) - z(201)*z(431)*z(2612) -
z(202)*z(441)*z(2612);
z(4326) = z(347)*z(4292) + z(350)*z(4293) + z(353)*z(4294) + z(2750)*z(365) + z(2763)*z(362) + z(2776)*z(359) + z(344)*z(4189) +
z(2779)*z(4046) + z(2782)*z(4086) + z(2785)*z(4126) + z(2788)*z(4161) + z(2791)*z(4175) + z(358)*z(2663) + z(4297)*z(2695) + z(4298)*z(2708) +
z(4299)*z(2721) + z(4300)*z(2689) + z(4301)*z(2676);
z(4327) = z(4302) + z(4303)*U22 + z(4304)*U23 + z(4305)*U24 + z(4306)*U19 + z(4307)*U20 + z(4308)*U21 + z(4309)*U16 + z(4310)*U17 +
z(4311)*U18 + z(4312)*U13 + z(4313)*U14 + z(4314)*U15 + z(4315)*U10 + z(4316)*U11 + z(4317)*U12 + z(4318)*U7 + z(4319)*U8 + z(4320)*U4
+ z(4321)*U5 + z(4322)*U6 + z(4323)*U3 + z(4324)*U1 + z(4325)*U2 + z(4326)*U9;
z(2902) = -0.07100000000000001*z(376) - 0.07100000000000001*z(348)*z(333) - 0.07100000000000001*z(351)*z(334) -
0.07100000000000001*z(354)*z(335) - 0.07100000000000001*z(2780)*z(280) - 0.07100000000000001*z(2783)*z(281) -
0.07100000000000001*z(2786)*z(282) - 0.07100000000000001*z(2870)*z(422) - 0.07100000000000001*z(2873)*z(432) -
0.07100000000000001*z(2876)*z(442);
z(2817) = z(346)*z(2777) + z(349)*z(2764) + z(352)*z(2751) + z(2778)*z(2703) + z(2781)*z(2716) + z(2784)*z(2729) + z(2787)*z(2690) +
z(2790)*z(2677) - z(46)*z(2664);
z(2915) = z(2902) + z(2817);
z(2805) = z(252)*z(2778) + z(253)*z(2781) + z(254)*z(2784);
z(2806) = z(255)*z(2778) + z(256)*z(2781) + z(257)*z(2784);
z(2807) = z(258)*z(2778) + z(259)*z(2781) + z(260)*z(2784);
z(2808) = z(2385)*z(2778) + z(2407)*z(2781) + z(2429)*z(2784);
z(2809) = z(2386)*z(2778) + z(2408)*z(2781) + z(2430)*z(2784);
z(2810) = z(2387)*z(2778) + z(2409)*z(2781) + z(2431)*z(2784);
z(2811) = z(2388)*z(2778) + z(2410)*z(2781) + z(2432)*z(2784);
z(2812) = z(2389)*z(2778) + z(2411)*z(2781) + z(2433)*z(2784);
z(2813) = z(2390)*z(2778) + z(2412)*z(2781) + z(2434)*z(2784);
z(2814) = z(2391)*z(2778) + z(2413)*z(2781) + z(2435)*z(2784);
z(2815) = z(2392)*z(2778) + z(2414)*z(2781) + z(2436)*z(2784);
z(2816) = z(2393)*z(2778) + z(2415)*z(2781) + z(2437)*z(2784);
z(2890) = 0.07100000000000001*z(28)*z(2870) - 0.07100000000000001*z(205)*z(2873) - 0.07100000000000001*z(206)*z(2876);
z(2795) = z(346)*z(2767) + z(349)*z(2752) + z(352)*z(2741) + z(2778)*z(2693) + z(2781)*z(2706) + z(2784)*z(2719) + z(2787)*z(2680) +
z(2790)*z(2667) - z(46)*z(2654);
z(2903) = z(2890) + z(2795);
z(2891) = -0.07100000000000001*z(197)*z(2870) - 0.07100000000000001*z(201)*z(2873) - 0.07100000000000001*z(202)*z(2876);
z(2793) = z(346)*z(2765) + z(349)*z(2753) + z(352)*z(2739) + z(2778)*z(2691) + z(2781)*z(2704) + z(2784)*z(2717) + z(2787)*z(2678) +
z(2790)*z(2665) - z(46)*z(2652);
z(2904) = z(2891) + z(2793);
z(2892) = -0.07100000000000001*z(199)*z(2870) - 0.07100000000000001*z(203)*z(2873) - 0.07100000000000001*z(204)*z(2876);
z(2794) = z(346)*z(2766) + z(349)*z(2754) + z(352)*z(2740) + z(2778)*z(2692) + z(2781)*z(2705) + z(2784)*z(2718) + z(2787)*z(2679) +
z(2790)*z(2666) - z(46)*z(2653);
z(2905) = z(2892) + z(2794);

z(2893) = -0.07100000000000001*z(413)*z(2870) - 0.07100000000000001*z(423)*z(2873) - 0.07100000000000001*z(433)*z(2876);
z(2796) = z(346)*z(2768) + z(349)*z(2755) + z(352)*z(2742) + z(2778)*z(2696) + z(2781)*z(2709) + z(2784)*z(2722) + z(2787)*z(2681) + z(2790)*z(2668) - z(46)*z(2655);
z(2906) = z(2893) + z(2796);
z(2894) = -0.07100000000000001*z(414)*z(2870) - 0.07100000000000001*z(424)*z(2873) - 0.07100000000000001*z(434)*z(2876);
z(2797) = z(346)*z(2769) + z(349)*z(2756) + z(352)*z(2743) + z(2778)*z(2694) + z(2781)*z(2707) + z(2784)*z(2720) + z(2787)*z(2682) + z(2790)*z(2669) - z(46)*z(2656);
z(2907) = z(2894) + z(2797);
z(2895) = -0.07100000000000001*z(415)*z(2870) - 0.07100000000000001*z(425)*z(2873) - 0.07100000000000001*z(435)*z(2876);
z(2798) = z(346)*z(2770) + z(349)*z(2757) + z(352)*z(2744) + z(2778)*z(2697) + z(2781)*z(2710) + z(2784)*z(2723) + z(2787)*z(2683) + z(2790)*z(2670) - z(46)*z(2657);
z(2908) = z(2895) + z(2798);
z(2896) = -0.07100000000000001*z(416)*z(2870) - 0.07100000000000001*z(426)*z(2873) - 0.07100000000000001*z(436)*z(2876);
z(2799) = z(346)*z(2771) + z(349)*z(2758) + z(352)*z(2745) + z(2778)*z(2698) + z(2781)*z(2711) + z(2784)*z(2724) + z(2787)*z(2684) + z(2790)*z(2671) - z(46)*z(2658);
z(2909) = z(2896) + z(2799);
z(2897) = -0.07100000000000001*z(417)*z(2870) - 0.07100000000000001*z(427)*z(2873) - 0.07100000000000001*z(437)*z(2876);
z(2800) = z(346)*z(2772) + z(349)*z(2759) + z(352)*z(2746) + z(2778)*z(2699) + z(2781)*z(2712) + z(2784)*z(2725) + z(2787)*z(2685) + z(2790)*z(2672) - z(46)*z(2659);
z(2910) = z(2897) + z(2800);
z(2898) = -0.07100000000000001*z(418)*z(2870) - 0.07100000000000001*z(428)*z(2873) - 0.07100000000000001*z(438)*z(2876);
z(2801) = z(346)*z(2773) + z(349)*z(2760) + z(352)*z(2747) + z(2778)*z(2700) + z(2781)*z(2713) + z(2784)*z(2726) + z(2787)*z(2686) + z(2790)*z(2673) - z(46)*z(2660);
z(2911) = z(2898) + z(2801);
z(2900) = -0.07100000000000001*z(420)*z(2870) - 0.07100000000000001*z(430)*z(2873) - 0.07100000000000001*z(440)*z(2876);
z(2803) = z(346)*z(2775) + z(349)*z(2762) + z(352)*z(2749) + z(2778)*z(2702) + z(2781)*z(2715) + z(2784)*z(2728) + z(2787)*z(2688) + z(2790)*z(2675) - z(46)*z(2662);
z(2913) = z(2900) + z(2803);
z(2901) = -0.07100000000000001*z(421)*z(2870) - 0.07100000000000001*z(431)*z(2873) - 0.07100000000000001*z(441)*z(2876);
z(2804) = z(346)*z(2776) + z(349)*z(2763) + z(352)*z(2750) + z(2778)*z(2695) + z(2781)*z(2708) + z(2784)*z(2721) + z(2787)*z(2689) + z(2790)*z(2676) - z(46)*z(2663);
z(2914) = z(2901) + z(2804);
z(4194) = z(2915) + z(2805)*U22 + z(2806)*U23 + z(2807)*U24 + z(2808)*U19 + z(2809)*U20 + z(2810)*U21 + z(2811)*U16 + z(2812)*U17 + z(2813)*U18 + z(2814)*U13 + z(2815)*U14 + z(2816)*U15 + z(2903)*U12 + z(2904)*U10 + z(2905)*U11 + z(2906)*U7 + z(2907)*U8 + z(2908)*U4 + z(2909)*U5 + z(2910)*U6 + z(2911)*U3 + z(2912)*U1 + z(2913)*U2 + z(2914)*U9;
z(2889) = 0.07100000000000001*z(374) + 0.07100000000000001*z(346)*z(333) + 0.07100000000000001*z(349)*z(334) + 0.07100000000000001*z(352)*z(335) + 0.07100000000000001*z(2778)*z(280) + 0.07100000000000001*z(2781)*z(281) + 0.07100000000000001*z(2784)*z(282) + 0.07100000000000001*z(2868)*z(422) + 0.07100000000000001*z(2871)*z(432) + 0.07100000000000001*z(2874)*z(442);
z(2867) = z(348)*z(2777) + z(351)*z(2764) + z(354)*z(2751) + z(345)*z(2664) + z(2780)*z(2703) + z(2783)*z(2716) + z(2786)*z(2729) + z(2789)*z(2690) + z(2792)*z(2677);
z(2928) = z(2889) + z(2867);
z(2855) = z(252)*z(2780) + z(253)*z(2783) + z(254)*z(2786);
z(2856) = z(255)*z(2780) + z(256)*z(2783) + z(257)*z(2786);
z(2857) = z(258)*z(2780) + z(259)*z(2783) + z(260)*z(2786);
z(2858) = z(2385)*z(2780) + z(2407)*z(2783) + z(2429)*z(2786);
z(2859) = z(2386)*z(2780) + z(2408)*z(2783) + z(2430)*z(2786);
z(2860) = z(2387)*z(2780) + z(2409)*z(2783) + z(2431)*z(2786);
z(2861) = z(2388)*z(2780) + z(2410)*z(2783) + z(2432)*z(2786);
z(2862) = z(2389)*z(2780) + z(2411)*z(2783) + z(2433)*z(2786);
z(2863) = z(2390)*z(2780) + z(2412)*z(2783) + z(2434)*z(2786);
z(2864) = z(2391)*z(2780) + z(2413)*z(2783) + z(2435)*z(2786);
z(2865) = z(2392)*z(2780) + z(2414)*z(2783) + z(2436)*z(2786);
z(2866) = z(2393)*z(2780) + z(2415)*z(2783) + z(2437)*z(2786);
z(2877) = 0.07100000000000001*z(197)*z(2868) + 0.07100000000000001*z(201)*z(2871) + 0.07100000000000001*z(202)*z(2874);
z(2843) = z(348)*z(2765) + z(351)*z(2753) + z(354)*z(2739) + z(345)*z(2652) + z(2780)*z(2691) + z(2783)*z(2704) + z(2786)*z(2717) + z(2789)*z(2678) + z(2792)*z(2665);
z(2916) = z(2877) + z(2843);
z(2878) = 0.07100000000000001*z(199)*z(2868) + 0.07100000000000001*z(203)*z(2871) + 0.07100000000000001*z(204)*z(2874);
z(2844) = z(348)*z(2766) + z(351)*z(2754) + z(354)*z(2740) + z(345)*z(2653) + z(2780)*z(2692) + z(2783)*z(2705) + z(2786)*z(2718) + z(2789)*z(2679) + z(2792)*z(2666);
z(2917) = z(2878) + z(2844);
z(2879) = 0.07100000000000001*z(205)*z(2871) + 0.07100000000000001*z(206)*z(2874) - 0.07100000000000001*z(28)*z(2868);
z(2845) = z(348)*z(2767) + z(351)*z(2752) + z(354)*z(2741) + z(345)*z(2654) + z(2780)*z(2693) + z(2783)*z(2706) + z(2786)*z(2719) + z(2789)*z(2680) + z(2792)*z(2667);
z(2918) = z(2879) + z(2845);
z(2880) = 0.07100000000000001*z(413)*z(2868) + 0.07100000000000001*z(423)*z(2871) + 0.07100000000000001*z(433)*z(2874);
z(2846) = z(348)*z(2768) + z(351)*z(2755) + z(354)*z(2742) + z(345)*z(2655) + z(2780)*z(2696) + z(2783)*z(2709) + z(2786)*z(2722) + z(2789)*z(2681) + z(2792)*z(2668);
z(2919) = z(2880) + z(2846);
z(2881) = 0.07100000000000001*z(414)*z(2868) + 0.07100000000000001*z(424)*z(2871) + 0.07100000000000001*z(434)*z(2874);
z(2847) = z(348)*z(2769) + z(351)*z(2756) + z(354)*z(2743) + z(345)*z(2656) + z(2780)*z(2694) + z(2783)*z(2707) + z(2786)*z(2720) + z(2789)*z(2682) + z(2792)*z(2669);
z(2920) = z(2881) + z(2847);
z(2882) = 0.07100000000000001*z(415)*z(2868) + 0.07100000000000001*z(425)*z(2871) + 0.07100000000000001*z(435)*z(2874);
z(2848) = z(348)*z(2770) + z(351)*z(2757) + z(354)*z(2744) + z(345)*z(2657) + z(2780)*z(2697) + z(2783)*z(2710) + z(2786)*z(2723) + z(2789)*z(2683) + z(2792)*z(2670);
z(2921) = z(2882) + z(2848);

z(2883) = 0.07100000000000001*z(416)*z(2868) + 0.07100000000000001*z(426)*z(2871) + 0.07100000000000001*z(436)*z(2874);
z(2849) = z(348)*z(2771) + z(351)*z(2758) + z(354)*z(2745) + z(345)*z(2658) + z(2780)*z(2698) + z(2783)*z(2711) + z(2786)*z(2724) + z(2789)*z(2684) + z(2792)*z(2671);
z(2922) = z(2883) + z(2849);
z(2884) = 0.07100000000000001*z(417)*z(2868) + 0.07100000000000001*z(427)*z(2871) + 0.07100000000000001*z(437)*z(2874);
z(2850) = z(348)*z(2772) + z(351)*z(2759) + z(354)*z(2746) + z(345)*z(2659) + z(2780)*z(2699) + z(2783)*z(2712) + z(2786)*z(2725) + z(2789)*z(2685) + z(2792)*z(2672);
z(2923) = z(2884) + z(2850);
z(2885) = 0.07100000000000001*z(418)*z(2868) + 0.07100000000000001*z(428)*z(2871) + 0.07100000000000001*z(438)*z(2874);
z(2851) = z(348)*z(2773) + z(351)*z(2760) + z(354)*z(2747) + z(345)*z(2660) + z(2780)*z(2700) + z(2783)*z(2713) + z(2786)*z(2726) + z(2789)*z(2686) + z(2792)*z(2673);
z(2924) = z(2885) + z(2851);
z(2887) = 0.07100000000000001*z(420)*z(2868) + 0.07100000000000001*z(430)*z(2871) + 0.07100000000000001*z(440)*z(2874);
z(2853) = z(348)*z(2775) + z(351)*z(2762) + z(354)*z(2749) + z(345)*z(2662) + z(2780)*z(2702) + z(2783)*z(2715) + z(2786)*z(2728) + z(2789)*z(2688) + z(2792)*z(2675);
z(2926) = z(2887) + z(2853);
z(2888) = 0.07100000000000001*z(421)*z(2868) + 0.07100000000000001*z(431)*z(2871) + 0.07100000000000001*z(441)*z(2874);
z(2854) = z(348)*z(2776) + z(351)*z(2763) + z(354)*z(2750) + z(345)*z(2663) + z(2780)*z(2695) + z(2783)*z(2708) + z(2786)*z(2721) + z(2789)*z(2689) + z(2792)*z(2676);
z(2927) = z(2888) + z(2854);
z(4196) = z(2928) + z(2855)*U22 + z(2856)*U23 + z(2857)*U24 + z(2858)*U19 + z(2859)*U20 + z(2860)*U21 + z(2861)*U16 + z(2862)*U17 + z(2863)*U18 + z(2864)*U13 + z(2865)*U14 + z(2866)*U15 + z(2916)*U10 + z(2917)*U11 + z(2918)*U12 + z(2919)*U7 + z(2920)*U8 + z(2921)*U4 + z(2922)*U5 + z(2923)*U6 + z(2924)*U3 + z(2925)*U1 + z(2926)*U2 + z(2927)*U9;
z(4372) = z(4327) +
z(4194)*(z(376)+z(348)*z(333)+z(351)*z(334)+z(354)*z(335)+z(2780)*z(280)+z(2783)*z(281)+z(2786)*z(282)+z(2870)*z(3256)+z(2873)*z(3257)+z(2876)*z(3258)) -
z(4196)*(z(374)+z(346)*z(333)+z(349)*z(334)+z(352)*z(335)+z(2778)*z(280)+z(2781)*z(281)+z(2784)*z(282)+z(2868)*z(3256)+z(2871)*z(3257)+z(2874)*z(3258));
z(4197) = z(299)*z(370) + z(300)*z(368) + z(301)*z(369) + z(348)*z(331) + z(351)*z(315) + z(354)*z(327);
z(4198) = z(302)*z(370) + z(303)*z(368) + z(304)*z(369) + z(348)*z(332) + z(351)*z(318) + z(354)*z(325);
z(4199) = z(305)*z(370) + z(306)*z(368) + z(307)*z(369) + z(348)*z(330) + z(351)*z(321) + z(354)*z(326);
z(4200) = z(43)*z(48)*QNS_3p + z(44)*z(47)*QNS_1p + z(43)*z(367) - z(44)*z(343)*QNS_3p;
z(4201) = z(43)*z(343)*QNS_3p + z(44)*z(48)*QNS_3p + z(44)*z(367) - z(43)*z(47)*QNS_1p;
z(4202) = z(197)*z(4200) + z(199)*z(4201) + z(2789)*z(234) + z(2792)*z(235) - z(27)*z(345)*QNE_2p - z(28)*z(366);
z(4203) = z(201)*z(4200) + z(203)*z(4201) + z(205)*z(366) + z(345)*z(221) + z(2789)*z(217) + z(2792)*z(219);
z(4204) = z(202)*z(4200) + z(204)*z(4201) + z(206)*z(366) + z(345)*z(228) + z(2789)*z(230) + z(2792)*z(229);
z(4205) = -0.07100000000000001*z(280)*z(4197) - 0.07100000000000001*z(281)*z(4198) - 0.07100000000000001*z(282)*z(4199) - 0.07100000000000001*z(333)*z(370) - 0.07100000000000001*z(334)*z(368) - 0.07100000000000001*z(335)*z(369) - 0.07100000000000001*z(422)*z(4202) - 0.07100000000000001*z(432)*z(4203) - 0.07100000000000001*z(442)*z(4204) - 0.07100000000000001*z(3548) - 0.07100000000000001*z(348)*z(3456) - 0.07100000000000001*z(351)*z(3463) - 0.07100000000000001*z(354)*z(3469) - 0.07100000000000001*z(2780)*z(3377) - 0.07100000000000001*z(2783)*z(3383) - 0.07100000000000001*z(2786)*z(3387) - 0.07100000000000001*z(2870)*z(3272) - 0.07100000000000001*z(2873)*z(3290) - 0.07100000000000001*z(2876)*z(3307);
z(4218) = z(299)*z(372) + z(300)*z(373) + z(301)*z(371) + z(346)*z(331) + z(349)*z(315) + z(352)*z(327);
z(4219) = z(302)*z(372) + z(303)*z(373) + z(304)*z(371) + z(346)*z(332) + z(349)*z(318) + z(352)*z(325);
z(4220) = z(305)*z(372) + z(306)*z(373) + z(307)*z(371) + z(346)*z(330) + z(349)*z(321) + z(352)*z(326);
z(4221) = -z(43)*z(46)*QNS_2p - z(44)*z(45)*QNS_3p;
z(4222) = z(43)*z(45)*QNS_3p - z(44)*z(46)*QNS_2p;
z(4223) = z(371)*z(2751) + z(372)*z(2777) + z(373)*z(2764) + z(346)*z(4212) + z(349)*z(4216) + z(352)*z(4217) + z(2778)*z(4031) + z(2781)*z(4071) + z(2784)*z(4111) + z(2787)*z(4149) + z(2790)*z(4163) + z(4218)*z(2703) + z(4219)*z(2716) + z(4220)*z(2729) + z(4221)*z(2690) + z(4222)*z(2677) - z(46)*z(4177) - z(45)*QNS_2p*z(2664);
z(4224) = z(252)*z(4218) + z(253)*z(4219) + z(254)*z(4220) + z(2778)*z(278) + z(2781)*z(265) + z(2784)*z(276);
z(4225) = z(255)*z(4218) + z(256)*z(4219) + z(257)*z(4220) + z(2778)*z(279) + z(2781)*z(268) + z(2784)*z(274);
z(4226) = z(258)*z(4218) + z(259)*z(4219) + z(260)*z(4220) + z(2778)*z(277) + z(2781)*z(271) + z(2784)*z(275);
z(4227) = z(2385)*z(4218) + z(2407)*z(4219) + z(2429)*z(4220) + z(2778)*z(3878) + z(2781)*z(3950) + z(2784)*z(3986);
z(4228) = z(2386)*z(4218) + z(2408)*z(4219) + z(2430)*z(4220) + z(2778)*z(3879) + z(2781)*z(3951) + z(2784)*z(3987);
z(4229) = z(2387)*z(4218) + z(2409)*z(4219) + z(2431)*z(4220) + z(2778)*z(3880) + z(2781)*z(3952) + z(2784)*z(3988);
z(4230) = z(2388)*z(4218) + z(2410)*z(4219) + z(2432)*z(4220) + z(2778)*z(3881) + z(2781)*z(3953) + z(2784)*z(3989);
z(4231) = z(2389)*z(4218) + z(2411)*z(4219) + z(2433)*z(4220) + z(2778)*z(3882) + z(2781)*z(3954) + z(2784)*z(3990);
z(4232) = z(2390)*z(4218) + z(2412)*z(4219) + z(2434)*z(4220) + z(2778)*z(3883) + z(2781)*z(3955) + z(2784)*z(3991);
z(4233) = z(2391)*z(4218) + z(2413)*z(4219) + z(2435)*z(4220) + z(2778)*z(3884) + z(2781)*z(3956) + z(2784)*z(3992);
z(4234) = z(2392)*z(4218) + z(2414)*z(4219) + z(2436)*z(4220) + z(2778)*z(3885) + z(2781)*z(3957) + z(2784)*z(3993);
z(4235) = z(2393)*z(4218) + z(2415)*z(4219) + z(2437)*z(4220) + z(2778)*z(3886) + z(2781)*z(3958) + z(2784)*z(3994);
z(4236) = 0.07100000000000001*z(27)*z(2870)*QNE_2p + 0.07100000000000001*z(28)*z(4202) - 0.07100000000000001*z(205)*z(4203) - 0.07100000000000001*z(206)*z(4204) - 0.07100000000000001*z(2873)*z(221) - 0.07100000000000001*z(2876)*z(228);
z(4240) = z(346)*z(4237) + z(349)*z(4238) + z(352)*z(4239) + z(2741)*z(371) + z(2752)*z(373) + z(2767)*z(372) + z(2778)*z(4040) + z(2781)*z(4080) + z(2784)*z(4120) + z(2787)*z(4152) + z(2790)*z(4166) + z(4218)*z(2693) + z(4219)*z(2706) + z(4220)*z(2719) + z(4221)*z(2680) + z(4222)*z(2667) - z(46)*z(4180) - z(45)*QNS_2p*z(2654);
z(4241) = -0.07100000000000001*z(197)*z(4202) - 0.07100000000000001*z(201)*z(4203) - 0.07100000000000001*z(202)*z(4204) - 0.07100000000000001*z(2870)*z(234) - 0.07100000000000001*z(2873)*z(217) - 0.07100000000000001*z(2876)*z(230);
z(4245) = z(346)*z(4242) + z(349)*z(4243) + z(352)*z(4244) + z(2739)*z(371) + z(2753)*z(373) + z(2765)*z(372) + z(2778)*z(4034) + z(2781)*z(4074) + z(2784)*z(4114) + z(2787)*z(4150) + z(2790)*z(4164) + z(4218)*z(2691) + z(4219)*z(2704) + z(4220)*z(2717) + z(4221)*z(2678) + z(4222)*z(2665) - z(46)*z(4178) - z(45)*QNS_2p*z(2652);
z(4246) = -0.07100000000000001*z(199)*z(4202) - 0.07100000000000001*z(203)*z(4203) - 0.07100000000000001*z(204)*z(4204) - 0.07100000000000001*z(2870)*z(235) - 0.07100000000000001*z(2873)*z(219) - 0.07100000000000001*z(2876)*z(229);
z(4250) = z(346)*z(4247) + z(349)*z(4248) + z(352)*z(4249) + z(2740)*z(371) + z(2754)*z(373) + z(2766)*z(372) + z(2778)*z(4037) + z(2781)*z(4077) + z(2784)*z(4117) + z(2787)*z(4151) + z(2790)*z(4165) + z(4218)*z(2692) + z(4219)*z(2705) + z(4220)*z(2718) + z(4221)*z(2679) + z(4222)*z(2666) - z(46)*z(4179) - z(45)*QNS_2p*z(2653);

z(4251) = -0.07100000000000001*z(413)*z(4202) - 0.07100000000000001*z(423)*z(4203) - 0.07100000000000001*z(433)*z(4204) - 0.07100000000000001*z(2870)*z(3273) - 0.07100000000000001*z(2873)*z(3291) - 0.07100000000000001*z(2876)*z(3308);
z(4255) = z(346)*z(4252) + z(349)*z(4253) + z(352)*z(4254) + z(2742)*z(371) + z(2755)*z(373) + z(2768)*z(372) + z(2778)*z(4049) + z(2781)*z(4089) + z(2784)*z(4129) + z(2787)*z(4153) + z(2790)*z(4167) + z(4218)*z(2696) + z(4219)*z(2709) + z(4220)*z(2722) + z(4221)*z(2681) + z(4222)*z(2668) - z(46)*z(4181) - z(45)*QNS_2p*z(2655);
z(4256) = -0.07100000000000001*z(414)*z(4202) - 0.07100000000000001*z(424)*z(4203) - 0.07100000000000001*z(434)*z(4204) - 0.07100000000000001*z(2870)*z(3274) - 0.07100000000000001*z(2873)*z(3292) - 0.07100000000000001*z(2876)*z(3309);
z(4260) = z(346)*z(4257) + z(349)*z(4258) + z(352)*z(4259) + z(2743)*z(371) + z(2756)*z(373) + z(2769)*z(372) + z(2778)*z(4043) + z(2781)*z(4083) + z(2784)*z(4123) + z(2787)*z(4154) + z(2790)*z(4168) + z(4218)*z(2694) + z(4219)*z(2707) + z(4220)*z(2720) + z(4221)*z(2682) + z(4222)*z(2669) - z(46)*z(4182) - z(45)*QNS_2p*z(2656);
z(4261) = -0.07100000000000001*z(415)*z(4202) - 0.07100000000000001*z(425)*z(4203) - 0.07100000000000001*z(435)*z(4204) - 0.07100000000000001*z(2870)*z(3275) - 0.07100000000000001*z(2873)*z(3293) - 0.07100000000000001*z(2876)*z(3310);
z(4265) = z(346)*z(4262) + z(349)*z(4263) + z(352)*z(4264) + z(2744)*z(371) + z(2757)*z(373) + z(2770)*z(372) + z(2778)*z(4052) + z(2781)*z(4092) + z(2784)*z(4132) + z(2787)*z(4155) + z(2790)*z(4169) + z(4218)*z(2697) + z(4219)*z(2710) + z(4220)*z(2723) + z(4221)*z(2683) + z(4222)*z(2670) - z(46)*z(4183) - z(45)*QNS_2p*z(2657);
z(4266) = -0.07100000000000001*z(416)*z(4202) - 0.07100000000000001*z(426)*z(4203) - 0.07100000000000001*z(436)*z(4204) - 0.07100000000000001*z(2870)*z(3276) - 0.07100000000000001*z(2873)*z(3294) - 0.07100000000000001*z(2876)*z(3311);
z(4270) = z(346)*z(4267) + z(349)*z(4268) + z(352)*z(4269) + z(2745)*z(371) + z(2758)*z(373) + z(2771)*z(372) + z(2778)*z(4055) + z(2781)*z(4095) + z(2784)*z(4135) + z(2787)*z(4156) + z(2790)*z(4170) + z(4218)*z(2698) + z(4219)*z(2711) + z(4220)*z(2724) + z(4221)*z(2684) + z(4222)*z(2671) - z(46)*z(4184) - z(45)*QNS_2p*z(2658);
z(4271) = -0.07100000000000001*z(417)*z(4202) - 0.07100000000000001*z(427)*z(4203) - 0.07100000000000001*z(437)*z(4204) - 0.07100000000000001*z(2870)*z(3277) - 0.07100000000000001*z(2873)*z(3295) - 0.07100000000000001*z(2876)*z(3312);
z(4275) = z(346)*z(4272) + z(349)*z(4273) + z(352)*z(4274) + z(2746)*z(371) + z(2759)*z(373) + z(2772)*z(372) + z(2778)*z(4058) + z(2781)*z(4098) + z(2784)*z(4138) + z(2787)*z(4157) + z(2790)*z(4171) + z(4218)*z(2699) + z(4219)*z(2712) + z(4220)*z(2725) + z(4221)*z(2685) + z(4222)*z(2672) - z(46)*z(4185) - z(45)*QNS_2p*z(2659);
z(4276) = -0.07100000000000001*z(418)*z(4202) - 0.07100000000000001*z(428)*z(4203) - 0.07100000000000001*z(438)*z(4204) - 0.07100000000000001*z(2870)*z(3278) - 0.07100000000000001*z(2873)*z(3296) - 0.07100000000000001*z(2876)*z(3313);
z(4280) = z(346)*z(4277) + z(349)*z(4278) + z(352)*z(4279) + z(2747)*z(371) + z(2760)*z(373) + z(2773)*z(372) + z(2778)*z(4061) + z(2781)*z(4101) + z(2784)*z(4141) + z(2787)*z(4158) + z(2790)*z(4172) + z(4218)*z(2700) + z(4219)*z(2713) + z(4220)*z(2726) + z(4221)*z(2686) + z(4222)*z(2673) - z(46)*z(4186) - z(45)*QNS_2p*z(2660);
z(4281) = -0.07100000000000001*z(419)*z(4202) - 0.07100000000000001*z(429)*z(4203) - 0.07100000000000001*z(439)*z(4204) - 0.07100000000000001*z(2870)*z(3279) - 0.07100000000000001*z(2873)*z(3297) - 0.07100000000000001*z(2876)*z(3314);
z(4285) = z(346)*z(4282) + z(349)*z(4283) + z(352)*z(4284) + z(2748)*z(371) + z(2761)*z(373) + z(2774)*z(372) + z(2778)*z(4064) + z(2781)*z(4104) + z(2784)*z(4144) + z(2787)*z(4159) + z(2790)*z(4173) + z(4218)*z(2701) + z(4219)*z(2714) + z(4220)*z(2727) + z(4221)*z(2687) + z(4222)*z(2674) - z(46)*z(4187) - z(45)*QNS_2p*z(2661);
z(4286) = -0.07100000000000001*z(420)*z(4202) - 0.07100000000000001*z(430)*z(4203) - 0.07100000000000001*z(440)*z(4204) - 0.07100000000000001*z(2870)*z(3280) - 0.07100000000000001*z(2873)*z(3298) - 0.07100000000000001*z(2876)*z(3315);
z(4290) = z(346)*z(4287) + z(349)*z(4288) + z(352)*z(4289) + z(2749)*z(371) + z(2762)*z(373) + z(2775)*z(372) + z(2778)*z(4067) + z(2781)*z(4107) + z(2784)*z(4147) + z(2787)*z(4160) + z(2790)*z(4174) + z(4218)*z(2702) + z(4219)*z(2715) + z(4220)*z(2728) + z(4221)*z(2688) + z(4222)*z(2675) - z(46)*z(4188) - z(45)*QNS_2p*z(2662);
z(4291) = -0.07100000000000001*z(421)*z(4202) - 0.07100000000000001*z(431)*z(4203) - 0.07100000000000001*z(441)*z(4204) - 0.07100000000000001*z(2870)*z(3281) - 0.07100000000000001*z(2873)*z(3299) - 0.07100000000000001*z(2876)*z(3316);
z(4295) = z(346)*z(4292) + z(349)*z(4293) + z(352)*z(4294) + z(2750)*z(371) + z(2763)*z(373) + z(2776)*z(372) + z(2778)*z(4046) + z(2781)*z(4086) + z(2784)*z(4126) + z(2787)*z(4161) + z(2790)*z(4175) + z(4218)*z(2695) + z(4219)*z(2708) + z(4220)*z(2721) + z(4221)*z(2689) + z(4222)*z(2676) - z(46)*z(4189) - z(45)*QNS_2p*z(2663);
z(4296) = z(4205) + z(4223) + z(4224)*U22 + z(4225)*U23 + z(4226)*U24 + z(4227)*U19 + z(4228)*U20 + z(4229)*U21 + z(4230)*U16 + z(4231)*U17 + z(4232)*U18 + z(4233)*U13 + z(4234)*U14 + z(4235)*U15 + (z(4236)+z(4240))*U12 + (z(4241)+z(4245))*U10 + (z(4246)+z(4250))*U11 + (z(4251)+z(4255))*U7 + (z(4256)+z(4260))*U8 + (z(4261)+z(4265))*U4 + (z(4266)+z(4270))*U5 + (z(4271)+z(4275))*U6 + (z(4276)+z(4280))*U3 + (z(4281)+z(4285))*U1 + (z(4286)+z(4290))*U2 + (z(4291)+z(4295))*U9;
z(2842) = z(347)*z(2777) + z(350)*z(2764) + z(353)*z(2751) + z(344)*z(2664) + z(2779)*z(2703) + z(2782)*z(2716) + z(2785)*z(2729) + z(2788)*z(2690) + z(2791)*z(2677);
z(2830) = z(252)*z(2779) + z(253)*z(2782) + z(254)*z(2785);
z(2831) = z(255)*z(2779) + z(256)*z(2782) + z(257)*z(2785);
z(2832) = z(258)*z(2779) + z(259)*z(2782) + z(260)*z(2785);
z(2833) = z(2385)*z(2779) + z(2407)*z(2782) + z(2429)*z(2785);
z(2834) = z(2386)*z(2779) + z(2408)*z(2782) + z(2430)*z(2785);
z(2835) = z(2387)*z(2779) + z(2409)*z(2782) + z(2431)*z(2785);
z(2836) = z(2388)*z(2779) + z(2410)*z(2782) + z(2432)*z(2785);
z(2837) = z(2389)*z(2779) + z(2411)*z(2782) + z(2433)*z(2785);
z(2838) = z(2390)*z(2779) + z(2412)*z(2782) + z(2434)*z(2785);
z(2839) = z(2391)*z(2779) + z(2413)*z(2782) + z(2435)*z(2785);
z(2840) = z(2392)*z(2779) + z(2414)*z(2782) + z(2436)*z(2785);
z(2841) = z(2393)*z(2779) + z(2415)*z(2782) + z(2437)*z(2785);
z(2818) = z(347)*z(2765) + z(350)*z(2753) + z(353)*z(2739) + z(344)*z(2652) + z(2779)*z(2691) + z(2782)*z(2704) + z(2785)*z(2717) + z(2788)*z(2678) + z(2791)*z(2665);
z(2819) = z(347)*z(2766) + z(350)*z(2754) + z(353)*z(2740) + z(344)*z(2653) + z(2779)*z(2692) + z(2782)*z(2705) + z(2785)*z(2718) + z(2788)*z(2679) + z(2791)*z(2666);
z(2820) = z(347)*z(2767) + z(350)*z(2752) + z(353)*z(2741) + z(344)*z(2654) + z(2779)*z(2693) + z(2782)*z(2706) + z(2785)*z(2719) + z(2788)*z(2680) + z(2791)*z(2667);
z(2821) = z(347)*z(2768) + z(350)*z(2755) + z(353)*z(2742) + z(344)*z(2655) + z(2779)*z(2696) + z(2782)*z(2709) + z(2785)*z(2722) + z(2788)*z(2681) + z(2791)*z(2668);
z(2822) = z(347)*z(2769) + z(350)*z(2756) + z(353)*z(2743) + z(344)*z(2656) + z(2779)*z(2694) + z(2782)*z(2707) + z(2785)*z(2720) + z(2788)*z(2682) + z(2791)*z(2669);
z(2823) = z(347)*z(2770) + z(350)*z(2757) + z(353)*z(2744) + z(344)*z(2657) + z(2779)*z(2697) + z(2782)*z(2710) + z(2785)*z(2723) + z(2788)*z(2683) + z(2791)*z(2670);
z(2824) = z(347)*z(2771) + z(350)*z(2758) + z(353)*z(2745) + z(344)*z(2658) + z(2779)*z(2698) + z(2782)*z(2711) + z(2785)*z(2724) + z(2788)*z(2684) + z(2791)*z(2671);

z(2825) = z(347)*z(2772) + z(350)*z(2759) + z(353)*z(2746) + z(344)*z(2659) + z(2779)*z(2699) + z(2782)*z(2712) + z(2785)*z(2725) + z(2788)*z(2685) + z(2791)*z(2672);
z(2826) = z(347)*z(2773) + z(350)*z(2760) + z(353)*z(2747) + z(344)*z(2660) + z(2779)*z(2700) + z(2782)*z(2713) + z(2785)*z(2726) + z(2788)*z(2686) + z(2791)*z(2673);
z(2828) = z(347)*z(2775) + z(350)*z(2762) + z(353)*z(2749) + z(344)*z(2662) + z(2779)*z(2702) + z(2782)*z(2715) + z(2785)*z(2728) + z(2788)*z(2688) + z(2791)*z(2675);
z(2829) = z(347)*z(2776) + z(350)*z(2763) + z(353)*z(2750) + z(344)*z(2663) + z(2779)*z(2695) + z(2782)*z(2708) + z(2785)*z(2721) + z(2788)*z(2689) + z(2791)*z(2676);
z(4195) = z(2842) + z(2830)*U22 + z(2831)*U23 + z(2832)*U24 + z(2833)*U19 + z(2834)*U20 + z(2835)*U21 + z(2836)*U16 + z(2837)*U17 + z(2838)*U18 + z(2839)*U13 + z(2840)*U14 + z(2841)*U15 + z(2818)*U10 + z(2819)*U11 + z(2820)*U12 + z(2821)*U7 + z(2822)*U8 + z(2823)*U4 + z(2824)*U5 + z(2825)*U6 + z(2826)*U3 + z(2827)*U1 + z(2828)*U2 + z(2829)*U9;
z(4370) = z(4296) +
z(4196)*(z(375)+z(347)*z(333)+z(350)*z(334)+z(353)*z(335)+z(2779)*z(280)+z(2782)*z(281)+z(2785)*z(282)+z(2869)*z(3256)+z(2872)*z(3257)+z(2875)*z(3258)) -
z(4195)*(z(376)+z(348)*z(333)+z(351)*z(334)+z(354)*z(335)+z(2780)*z(280)+z(2783)*z(281)+z(2786)*z(282)+z(2870)*z(3256)+z(2873)*z(3257)+z(2876)*z(3258));
z(4328) = z(27)*z(46)*QNE_2p + z(28)*z(45)*QNS_2p + z(197)*z(4221) + z(199)*z(4222) + z(2787)*z(234) + z(2790)*z(235);
z(4329) = z(201)*z(4221) + z(203)*z(4222) + z(2787)*z(217) + z(2790)*z(219) - z(45)*z(205)*QNS_2p - z(46)*z(221);
z(4330) = z(202)*z(4221) + z(204)*z(4222) + z(2787)*z(230) + z(2790)*z(229) - z(45)*z(206)*QNS_2p - z(46)*z(228);
z(4331) = 0.07100000000000001*z(280)*z(4218) + 0.07100000000000001*z(281)*z(4219) + 0.07100000000000001*z(282)*z(4220) + 0.07100000000000001*z(333)*z(372) + 0.07100000000000001*z(334)*z(373) + 0.07100000000000001*z(335)*z(371) + 0.07100000000000001*z(422)*z(4328) + 0.07100000000000001*z(432)*z(4329) + 0.07100000000000001*z(442)*z(4330) + 0.07100000000000001*z(3538) + 0.07100000000000001*z(346)*z(3456) + 0.07100000000000001*z(349)*z(3463) + 0.07100000000000001*z(352)*z(3469) + 0.07100000000000001*z(2778)*z(3377) + 0.07100000000000001*z(2781)*z(3383) + 0.07100000000000001*z(2784)*z(3387) + 0.07100000000000001*z(2868)*z(3272) + 0.07100000000000001*z(2871)*z(3290) + 0.07100000000000001*z(2874)*z(3307);
z(4332) = z(368)*z(2764) + z(369)*z(2751) + z(370)*z(2777) + z(348)*z(4212) + z(351)*z(4216) + z(354)*z(4217) + z(345)*z(4177) + z(2780)*z(4031) + z(2783)*z(4071) + z(2786)*z(4111) + z(2789)*z(4149) + z(2792)*z(4163) + z(366)*z(2664) + z(4197)*z(2703) + z(4198)*z(2716) + z(4199)*z(2729) + z(4200)*z(2690) + z(4201)*z(2677);
z(4333) = z(252)*z(4197) + z(253)*z(4198) + z(254)*z(4199) + z(2780)*z(278) + z(2783)*z(265) + z(2786)*z(276);
z(4334) = z(255)*z(4197) + z(256)*z(4198) + z(257)*z(4199) + z(2780)*z(279) + z(2783)*z(268) + z(2786)*z(274);
z(4335) = z(258)*z(4197) + z(259)*z(4198) + z(260)*z(4199) + z(2780)*z(277) + z(2783)*z(271) + z(2786)*z(275);
z(4336) = z(2385)*z(4197) + z(2407)*z(4198) + z(2429)*z(4199) + z(2780)*z(3878) + z(2783)*z(3950) + z(2786)*z(3986);
z(4337) = z(2386)*z(4197) + z(2408)*z(4198) + z(2430)*z(4199) + z(2780)*z(3879) + z(2783)*z(3951) + z(2786)*z(3987);
z(4338) = z(2387)*z(4197) + z(2409)*z(4198) + z(2431)*z(4199) + z(2780)*z(3880) + z(2783)*z(3952) + z(2786)*z(3988);
z(4339) = z(2388)*z(4197) + z(2410)*z(4198) + z(2432)*z(4199) + z(2780)*z(3881) + z(2783)*z(3953) + z(2786)*z(3989);
z(4340) = z(2389)*z(4197) + z(2411)*z(4198) + z(2433)*z(4199) + z(2780)*z(3882) + z(2783)*z(3954) + z(2786)*z(3990);
z(4341) = z(2390)*z(4197) + z(2412)*z(4198) + z(2434)*z(4199) + z(2780)*z(3883) + z(2783)*z(3955) + z(2786)*z(3991);
z(4342) = z(2391)*z(4197) + z(2413)*z(4198) + z(2435)*z(4199) + z(2780)*z(3884) + z(2783)*z(3956) + z(2786)*z(3992);
z(4343) = z(2392)*z(4197) + z(2414)*z(4198) + z(2436)*z(4199) + z(2780)*z(3885) + z(2783)*z(3957) + z(2786)*z(3993);
z(4344) = z(2393)*z(4197) + z(2415)*z(4198) + z(2437)*z(4199) + z(2780)*z(3886) + z(2783)*z(3958) + z(2786)*z(3994);
z(4345) = 0.07100000000000001*z(197)*z(4328) + 0.07100000000000001*z(201)*z(4329) + 0.07100000000000001*z(202)*z(4330) + 0.07100000000000001*z(2868)*z(234) + 0.07100000000000001*z(2871)*z(217) + 0.07100000000000001*z(2874)*z(230);
z(4346) = z(348)*z(4242) + z(351)*z(4243) + z(354)*z(4244) + z(2739)*z(369) + z(2753)*z(368) + z(2765)*z(370) + z(345)*z(4178) + z(2780)*z(4034) + z(2783)*z(4074) + z(2786)*z(4114) + z(2789)*z(4150) + z(2792)*z(4164) + z(366)*z(2652) + z(4197)*z(2691) + z(4198)*z(2704) + z(4199)*z(2717) + z(4200)*z(2678) + z(4201)*z(2665);
z(4347) = 0.07100000000000001*z(199)*z(4328) + 0.07100000000000001*z(203)*z(4329) + 0.07100000000000001*z(204)*z(4330) + 0.07100000000000001*z(2868)*z(235) + 0.07100000000000001*z(2871)*z(219) + 0.07100000000000001*z(2874)*z(229);
z(4348) = z(348)*z(4247) + z(351)*z(4248) + z(354)*z(4249) + z(2740)*z(369) + z(2754)*z(368) + z(2766)*z(370) + z(345)*z(4179) + z(2780)*z(4037) + z(2783)*z(4077) + z(2786)*z(4117) + z(2789)*z(4151) + z(2792)*z(4165) + z(366)*z(2653) + z(4197)*z(2692) + z(4198)*z(2705) + z(4199)*z(2718) + z(4200)*z(2679) + z(4201)*z(2666);
z(4349) = 0.07100000000000001*z(205)*z(4329) + 0.07100000000000001*z(206)*z(4330) + 0.07100000000000001*z(2871)*z(221) + 0.07100000000000001*z(2874)*z(228) - 0.07100000000000001*z(27)*z(2868)*QNE_2p - 0.07100000000000001*z(28)*z(4328);
z(4350) = z(348)*z(4237) + z(351)*z(4238) + z(354)*z(4239) + z(2741)*z(369) + z(2752)*z(368) + z(2767)*z(370) + z(345)*z(4180) + z(2780)*z(4040) + z(2783)*z(4080) + z(2786)*z(4120) + z(2789)*z(4152) + z(2792)*z(4166) + z(366)*z(2654) + z(4197)*z(2693) + z(4198)*z(2706) + z(4199)*z(2719) + z(4200)*z(2680) + z(4201)*z(2667);
z(4351) = 0.07100000000000001*z(413)*z(4328) + 0.07100000000000001*z(423)*z(4329) + 0.07100000000000001*z(433)*z(4330) + 0.07100000000000001*z(2868)*z(3273) + 0.07100000000000001*z(2871)*z(3291) + 0.07100000000000001*z(2874)*z(3308);
z(4352) = z(348)*z(4252) + z(351)*z(4253) + z(354)*z(4254) + z(2742)*z(369) + z(2755)*z(368) + z(2768)*z(370) + z(345)*z(4181) + z(2780)*z(4049) + z(2783)*z(4089) + z(2786)*z(4129) + z(2789)*z(4153) + z(2792)*z(4167) + z(366)*z(2655) + z(4197)*z(2696) + z(4198)*z(2709) + z(4199)*z(2722) + z(4200)*z(2681) + z(4201)*z(2668);
z(4353) = 0.07100000000000001*z(414)*z(4328) + 0.07100000000000001*z(424)*z(4329) + 0.07100000000000001*z(434)*z(4330) + 0.07100000000000001*z(2868)*z(3274) + 0.07100000000000001*z(2871)*z(3292) + 0.07100000000000001*z(2874)*z(3309);
z(4354) = z(348)*z(4257) + z(351)*z(4258) + z(354)*z(4259) + z(2743)*z(369) + z(2756)*z(368) + z(2769)*z(370) + z(345)*z(4182) + z(2780)*z(4043) + z(2783)*z(4083) + z(2786)*z(4123) + z(2789)*z(4154) + z(2792)*z(4168) + z(366)*z(2656) + z(4197)*z(2694) + z(4198)*z(2707) + z(4199)*z(2720) + z(4200)*z(2682) + z(4201)*z(2669);
z(4355) = 0.07100000000000001*z(415)*z(4328) + 0.07100000000000001*z(425)*z(4329) + 0.07100000000000001*z(435)*z(4330) + 0.07100000000000001*z(2868)*z(3275) + 0.07100000000000001*z(2871)*z(3293) + 0.07100000000000001*z(2874)*z(3310);
z(4356) = z(348)*z(4262) + z(351)*z(4263) + z(354)*z(4264) + z(2744)*z(369) + z(2757)*z(368) + z(2770)*z(370) + z(345)*z(4183) + z(2780)*z(4052) + z(2783)*z(4092) + z(2786)*z(4132) + z(2789)*z(4155) + z(2792)*z(4169) + z(366)*z(2657) + z(4197)*z(2697) + z(4198)*z(2710) + z(4199)*z(2723) + z(4200)*z(2683) + z(4201)*z(2670);
z(4357) = 0.07100000000000001*z(416)*z(4328) + 0.07100000000000001*z(426)*z(4329) + 0.07100000000000001*z(436)*z(4330) + 0.07100000000000001*z(2868)*z(3276) + 0.07100000000000001*z(2871)*z(3294) + 0.07100000000000001*z(2874)*z(3311);
z(4358) = z(348)*z(4267) + z(351)*z(4268) + z(354)*z(4269) + z(2745)*z(369) + z(2758)*z(368) + z(2771)*z(370) + z(345)*z(4184) + z(2780)*z(4055) + z(2783)*z(4095) + z(2786)*z(4135) + z(2789)*z(4156) + z(2792)*z(4170) + z(366)*z(2658) + z(4197)*z(2698) + z(4198)*z(2711) + z(4199)*z(2724) + z(4200)*z(2684) + z(4201)*z(2671);
z(4359) = 0.07100000000000001*z(417)*z(4328) + 0.07100000000000001*z(427)*z(4329) + 0.07100000000000001*z(437)*z(4330) + 0.07100000000000001*z(2868)*z(3277) + 0.07100000000000001*z(2871)*z(3295) + 0.07100000000000001*z(2874)*z(3312);

z(4360) = z(348)*z(4272) + z(351)*z(4273) + z(354)*z(4274) + z(2746)*z(369) + z(2759)*z(368) + z(2772)*z(370) + z(345)*z(4185) +
z(2780)*z(4058) + z(2783)*z(4098) + z(2786)*z(4138) + z(2789)*z(4157) + z(2792)*z(4171) + z(366)*z(2659) + z(4197)*z(2699) + z(4198)*z(2712) +
z(4199)*z(2725) + z(4200)*z(2685) + z(4201)*z(2672);
z(4361) = 0.07100000000000001*z(418)*z(4328) + 0.07100000000000001*z(428)*z(4329) + 0.07100000000000001*z(438)*z(4330) +
0.07100000000000001*z(2868)*z(3278) + 0.07100000000000001*z(2871)*z(3296) + 0.07100000000000001*z(2874)*z(3313);
z(4362) = z(348)*z(4277) + z(351)*z(4278) + z(354)*z(4279) + z(2747)*z(369) + z(2760)*z(368) + z(2773)*z(370) + z(345)*z(4186) +
z(2780)*z(4061) + z(2783)*z(4101) + z(2786)*z(4141) + z(2789)*z(4158) + z(2792)*z(4172) + z(366)*z(2660) + z(4197)*z(2700) + z(4198)*z(2713) +
z(4199)*z(2726) + z(4200)*z(2686) + z(4201)*z(2673);
z(4363) = 0.07100000000000001*z(419)*z(4328) + 0.07100000000000001*z(429)*z(4329) + 0.07100000000000001*z(439)*z(4330) +
0.07100000000000001*z(2868)*z(3279) + 0.07100000000000001*z(2871)*z(3297) + 0.07100000000000001*z(2874)*z(3314);
z(4364) = z(348)*z(4282) + z(351)*z(4283) + z(354)*z(4284) + z(2748)*z(369) + z(2761)*z(368) + z(2774)*z(370) + z(345)*z(4187) +
z(2780)*z(4064) + z(2783)*z(4104) + z(2786)*z(4144) + z(2789)*z(4159) + z(2792)*z(4173) + z(366)*z(2661) + z(4197)*z(2701) + z(4198)*z(2714) +
z(4199)*z(2727) + z(4200)*z(2687) + z(4201)*z(2674);
z(4365) = 0.07100000000000001*z(420)*z(4328) + 0.07100000000000001*z(430)*z(4329) + 0.07100000000000001*z(440)*z(4330) +
0.07100000000000001*z(2868)*z(3280) + 0.07100000000000001*z(2871)*z(3298) + 0.07100000000000001*z(2874)*z(3315);
z(4366) = z(348)*z(4287) + z(351)*z(4288) + z(354)*z(4289) + z(2749)*z(369) + z(2762)*z(368) + z(2775)*z(370) + z(345)*z(4188) +
z(2780)*z(4067) + z(2783)*z(4107) + z(2786)*z(4147) + z(2789)*z(4160) + z(2792)*z(4174) + z(366)*z(2662) + z(4197)*z(2702) + z(4198)*z(2715) +
z(4199)*z(2728) + z(4200)*z(2688) + z(4201)*z(2675);
z(4367) = 0.07100000000000001*z(421)*z(4328) + 0.07100000000000001*z(431)*z(4329) + 0.07100000000000001*z(441)*z(4330) +
0.07100000000000001*z(2868)*z(3281) + 0.07100000000000001*z(2871)*z(3299) + 0.07100000000000001*z(2874)*z(3316);
z(4368) = z(348)*z(4292) + z(351)*z(4293) + z(354)*z(4294) + z(2750)*z(369) + z(2763)*z(368) + z(2776)*z(370) + z(345)*z(4189) +
z(2780)*z(4046) + z(2783)*z(4086) + z(2786)*z(4126) + z(2789)*z(4161) + z(2792)*z(4175) + z(366)*z(2663) + z(4197)*z(2695) + z(4198)*z(2708) +
z(4199)*z(2721) + z(4200)*z(2689) + z(4201)*z(2676);
z(4369) = z(4331) + z(4332) + z(4333)*U22 + z(4334)*U23 + z(4335)*U24 + z(4336)*U19 + z(4337)*U20 + z(4338)*U21 + z(4339)*U16 +
z(4340)*U17 + z(4341)*U18 + z(4342)*U13 + z(4343)*U14 + z(4344)*U15 + (z(4345)+z(4346))*U10 + (z(4347)+z(4348))*U11 +
(z(4349)+z(4350))*U12 + (z(4351)+z(4352))*U7 + (z(4353)+z(4354))*U8 + (z(4355)+z(4356))*U4 + (z(4357)+z(4358))*U5 + (z(4359)+z(4360))*U6 +
(z(4361)+z(4362))*U3 + (z(4363)+z(4364))*U1 + (z(4365)+z(4366))*U2 + (z(4367)+z(4368))*U9;
z(4371) = z(4369) +
z(4195)*(z(374)+z(346)*z(333)+z(349)*z(334)+z(352)*z(335)+z(2778)*z(280)+z(2781)*z(281)+z(2784)*z(282)+z(2868)*z(3256)+z(2871)*z(3257)+z(
2874)*z(3258)) -
z(4194)*(z(375)+z(347)*z(333)+z(350)*z(334)+z(353)*z(335)+z(2779)*z(280)+z(2782)*z(281)+z(2785)*z(282)+z(2869)*z(3256)+z(2872)*z(3257)+z(
2875)*z(3258));
z(3592) = z(3133)*BOBA1 - z(3125)*BOBA2;
z(3593) = z(94)*BOBA1 + z(9)*QNB_2p*BOBA2;
z(3594) = z(146)*BOBA1 - z(138)*BOBA2;
z(3595) = z(147)*BOBA1 - z(141)*BOBA2;
z(3596) = z(3591) + z(3592) + z(3593)*U3 + z(3594)*U1 + z(3595)*U2;
z(3097) = z(377) + z(113)*U1 + z(114)*U2 - z(10)*U3;
z(554) = z(377)*BOBA3 - z(379)*BOBA1;
z(560) = z(545) + z(554);
z(551) = -z(10)*BOBA3 - z(81)*BOBA1;
z(553) = z(114)*BOBA3 - z(118)*BOBA1;
z(3574) = z(560) + U14 + z(551)*U3 + z(552)*U1 + z(553)*U2;
z(3098) = z(378) + z(80)*U3 + z(115)*U1 + z(116)*U2;
z(558) = z(379)*BOBA2 - z(378)*BOBA3;
z(559) = z(544) + z(558);
z(555) = z(81)*BOBA2 - z(80)*BOBA3;
z(557) = z(118)*BOBA2 - z(116)*BOBA3;
z(3573) = z(559) + U13 + z(555)*U3 + z(556)*U1 + z(557)*U2;
z(3598) = z(3596) + z(3097)*z(3574) - z(3098)*z(3573);
z(3586) = z(3125)*BOBA3 - z(3139)*BOBA1;
z(3587) = -z(102)*BOBA1 - z(9)*QNB_2p*BOBA3;
z(3588) = z(138)*BOBA3 - z(149)*BOBA1;
z(3589) = z(141)*BOBA3 - z(150)*BOBA1;
z(3590) = z(3585) + z(3586) + z(3587)*U3 + z(3588)*U1 + z(3589)*U2;
z(3099) = z(379) + z(81)*U3 + z(117)*U1 + z(118)*U2;
z(550) = z(378)*BOBA1 - z(377)*BOBA2;
z(561) = z(546) + z(550);
z(547) = z(10)*BOBA2 + z(80)*BOBA1;
z(549) = z(116)*BOBA1 - z(114)*BOBA2;
z(3575) = z(561) + U15 + z(547)*U3 + z(548)*U1 + z(549)*U2;
z(3599) = z(3590) + z(3099)*z(3573) - z(3097)*z(3575);
z(3580) = z(3139)*BOBA2 - z(3133)*BOBA3;
z(3581) = z(102)*BOBA2 - z(94)*BOBA3;
z(3582) = z(149)*BOBA2 - z(146)*BOBA3;
z(3583) = z(150)*BOBA2 - z(147)*BOBA3;
z(3584) = z(3579) + z(3580) + z(3581)*U3 + z(3582)*U1 + z(3583)*U2;
z(3597) = z(3584) + z(3098)*z(3575) - z(3099)*z(3574);
z(3607) = z(3191)*COCB2 - z(3180)*COCB3;
z(3608) = z(153)*COCB2 - z(140)*COCB3;
z(3609) = z(154)*COCB2 - z(143)*COCB3;
z(3610) = z(152)*COCB2 - z(144)*COCB3;
z(3615) = z(3192)*COCB2 - z(3181)*COCB3;
z(3620) = z(3193)*COCB2 - z(3182)*COCB3;
z(3625) = z(3194)*COCB2 - z(3183)*COCB3;
z(3626) = z(3606) + z(3607) + z(160)*U15 + z(161)*U13 + z(162)*U14 + z(3608)*U4 + z(3609)*U5 + z(3610)*U6 + (z(3614)+z(3615))*U3 +
(z(3619)+z(3620))*U1 + (z(3624)+z(3625))*U2;
z(3151) = z(387) + z(123)*U4 + z(125)*U5 + z(127)*U6 + z(384)*U3 + z(385)*U1 + z(386)*U2;

```
z(759) = z(387)*COCB1 - z(383)*COCB2;
z(785) = z(752) + z(759);
z(753) = z(123)*COCB1 - z(119)*COCB2;
z(754) = z(125)*COCB1 - z(121)*COCB2;
z(755) = z(16)*COCB2 + z(127)*COCB1;
z(756) = z(384)*COCB1 - z(380)*COCB2;
z(782) = z(749) + z(756);
z(758) = z(386)*COCB1 - z(382)*COCB2;
z(784) = z(751) + z(758);
z(3602) = z(785) + U18 + z(131)*U13 + z(134)*U14 + z(137)*U15 + z(753)*U4 + z(754)*U5 + z(755)*U6 + z(782)*U3 + z(783)*U1 + z(784)*U2;
z(3152) = z(391) + z(124)*U4 + z(126)*U5 + z(128)*U6 + z(388)*U3 + z(389)*U1 + z(390)*U2;
z(766) = z(383)*COCB3 - z(391)*COCB1;
z(781) = z(748) + z(766);
z(760) = z(119)*COCB3 - z(124)*COCB1;
z(761) = z(121)*COCB3 - z(126)*COCB1;
z(762) = -z(16)*COCB3 - z(128)*COCB1;
z(763) = z(380)*COCB3 - z(388)*COCB1;
z(778) = z(745) + z(763);
z(765) = z(382)*COCB3 - z(390)*COCB1;
z(780) = z(747) + z(765);
z(3601) = z(781) + U17 + z(130)*U13 + z(133)*U14 + z(136)*U15 + z(760)*U4 + z(761)*U5 + z(762)*U6 + z(778)*U3 + z(779)*U1 + z(780)*U2;
z(3651) = z(3626) + z(3151)*z(3602) - z(3152)*z(3601);
z(3628) = z(3168)*COCB3 - z(3191)*COCB1;
z(3629) = z(158)*COCB3 - z(153)*COCB1;
z(3630) = z(159)*COCB3 - z(154)*COCB1;
z(3631) = -z(152)*COCB1 - z(15)*QNC_2p*COCB3;
z(3633) = z(3169)*COCB3 - z(3192)*COCB1;
z(3635) = z(3170)*COCB3 - z(3193)*COCB1;
z(3637) = z(3171)*COCB3 - z(3194)*COCB1;
z(3638) = z(3627) + z(3628) + z(145)*U13 + z(148)*U14 + z(151)*U15 + z(3629)*U4 + z(3630)*U5 + z(3631)*U6 + (z(3632)+z(3633))*U3 +
(z(3634)+z(3635))*U1 + (z(3636)+z(3637))*U2;
z(773) = z(391)*COCB2 - z(387)*COCB3;
z(777) = z(744) + z(773);
z(767) = z(124)*COCB2 - z(123)*COCB3;
z(768) = z(126)*COCB2 - z(125)*COCB3;
z(769) = z(128)*COCB2 - z(127)*COCB3;
z(770) = z(388)*COCB2 - z(384)*COCB3;
z(774) = z(741) + z(770);
z(772) = z(390)*COCB2 - z(386)*COCB3;
z(776) = z(743) + z(772);
z(3600) = z(777) + U16 + z(129)*U13 + z(132)*U14 + z(135)*U15 + z(767)*U4 + z(768)*U5 + z(769)*U6 + z(774)*U3 + z(775)*U1 + z(776)*U2;
z(3150) = z(383) + z(119)*U4 + z(121)*U5 + z(380)*U3 + z(381)*U1 + z(382)*U2 - z(16)*U6;
z(3653) = z(3638) + z(3152)*z(3600) - z(3150)*z(3602);
z(3640) = z(3180)*COCB1 - z(3168)*COCB2;
z(3641) = z(140)*COCB1 - z(158)*COCB2;
z(3642) = z(143)*COCB1 - z(159)*COCB2;
z(3643) = z(144)*COCB1 + z(15)*QNC_2p*COCB2;
z(3645) = z(3181)*COCB1 - z(3169)*COCB2;
z(3647) = z(3182)*COCB1 - z(3170)*COCB2;
z(3649) = z(3183)*COCB1 - z(3171)*COCB2;
z(3650) = z(3639) + z(3640) + z(155)*U14 + z(156)*U15 + z(157)*U13 + z(3641)*U4 + z(3642)*U5 + z(3643)*U6 + (z(3644)+z(3645))*U3 +
(z(3646)+z(3647))*U1 + (z(3648)+z(3649))*U2;
z(3652) = z(3650) + z(3150)*z(3601) - z(3151)*z(3600);
z(3661) = z(3239)*DODC2 - z(3227)*DODC3;
z(3665) = z(223)*DODC3 + z(225)*DODC2;
z(3667) = -z(222)*DODC3 - z(226)*DODC2;
z(3672) = z(3240)*DODC2 - z(3228)*DODC3;
z(3677) = z(3241)*DODC2 - z(3229)*DODC3;
z(3682) = z(3242)*DODC2 - z(3230)*DODC3;
z(3687) = z(3243)*DODC2 - z(3231)*DODC3;
z(3692) = z(3244)*DODC2 - z(3232)*DODC3;
z(3697) = z(3245)*DODC2 - z(3233)*DODC3;
z(3666) = QND_1p*(z(23)*DODC3+z(24)*DODC2);
z(3698) = z(3660) + z(3661) + z(187)*U18 + z(188)*U16 + z(189)*U17 + z(3662)*U13 + z(3663)*U14 + z(3664)*U15 + z(3665)*U7 + z(3667)*U8 +
(z(3671)+z(3672))*U4 + (z(3676)+z(3677))*U5 + (z(3681)+z(3682))*U6 + (z(3686)+z(3687))*U3 + (z(3691)+z(3692))*U1 + (z(3696)+z(3697))*U2 -
z(3666)*U9;
z(3206) = z(405) + z(24)*U9 + z(194)*U8 + z(399)*U4 + z(400)*U5 + z(401)*U6 + z(402)*U3 + z(403)*U1 + z(404)*U2 - z(193)*U7;
z(1184) = z(405)*DODC1 - z(398)*DODC2;
z(1225) = z(1174) + z(1184);
z(1175) = -z(19)*DODC2 - z(193)*DODC1;
z(1176) = z(194)*DODC1 - z(20)*DODC2;
z(1183) = z(24)*DODC1;
z(1177) = z(399)*DODC1 - z(392)*DODC2;
z(1219) = z(1168) + z(1177);
z(1178) = z(400)*DODC1 - z(393)*DODC2;
z(1220) = z(1169) + z(1178);
z(1179) = z(401)*DODC1 - z(394)*DODC2;
z(1221) = z(1170) + z(1179);
```

z(1180) = z(402)*DODC1 - z(395)*DODC2;
z(1222) = z(1171) + z(1180);
z(1182) = z(404)*DODC1 - z(397)*DODC2;
z(1224) = z(1173) + z(1182);
z(3656) = z(1225) + U21 + z(173)*U16 + z(175)*U17 + z(177)*U18 + z(1165)*U13 + z(1166)*U14 + z(1167)*U15 + z(1175)*U7 + z(1176)*U8 + z(1183)*U9 + z(1219)*U4 + z(1220)*U5 + z(1221)*U6 + z(1222)*U3 + z(1223)*U1 + z(1224)*U2;
z(3207) = z(412) + z(23)*U9 + z(195)*U7 + z(406)*U4 + z(407)*U5 + z(408)*U6 + z(409)*U3 + z(410)*U1 + z(411)*U2 - z(196)*U8;
z(1194) = z(398)*DODC3 - z(412)*DODC1;
z(1218) = z(1164) + z(1194);
z(1185) = z(19)*DODC3 - z(195)*DODC1;
z(1186) = z(20)*DODC3 + z(196)*DODC1;
z(1187) = z(392)*DODC3 - z(406)*DODC1;
z(1212) = z(1158) + z(1187);
z(1188) = z(393)*DODC3 - z(407)*DODC1;
z(1213) = z(1159) + z(1188);
z(1189) = z(394)*DODC3 - z(408)*DODC1;
z(1214) = z(1160) + z(1189);
z(1190) = z(395)*DODC3 - z(409)*DODC1;
z(1215) = z(1161) + z(1190);
z(1192) = z(397)*DODC3 - z(411)*DODC1;
z(1217) = z(1163) + z(1192);
z(1193) = z(23)*DODC1;
z(3655) = z(1218) + U20 + z(172)*U16 + z(174)*U17 + z(176)*U18 + z(1155)*U13 + z(1156)*U14 + z(1157)*U15 + z(1185)*U7 + z(1186)*U8 + z(1212)*U4 + z(1213)*U5 + z(1214)*U6 + z(1215)*U3 + z(1216)*U1 + z(1217)*U2 - z(1193)*U9;
z(3741) = z(3698) + z(3206)*z(3656) - z(3207)*z(3655);
z(3700) = z(3215)*DODC3 - z(3239)*DODC1;
z(3704) = -z(225)*DODC1 - z(20)*QND_3p*DODC3;
z(3705) = z(226)*DODC1 + z(19)*QND_3p*DODC3;
z(3718) = z(24)*QND_1p*DODC1;
z(3707) = z(3216)*DODC3 - z(3240)*DODC1;
z(3709) = z(3217)*DODC3 - z(3241)*DODC1;
z(3711) = z(3218)*DODC3 - z(3242)*DODC1;
z(3713) = z(3219)*DODC3 - z(3243)*DODC1;
z(3715) = z(3220)*DODC3 - z(3244)*DODC1;
z(3717) = z(3221)*DODC3 - z(3245)*DODC1;
z(3719) = z(3699) + z(3700) + z(179)*U16 + z(181)*U17 + z(183)*U18 + z(3701)*U13 + z(3702)*U14 + z(3703)*U15 + z(3704)*U7 + z(3705)*U8 + z(3718)*U9 + (z(3706)+z(3707))*U4 + (z(3708)+z(3709))*U5 + (z(3710)+z(3711))*U6 + (z(3712)+z(3713))*U3 + (z(3714)+z(3715))*U1 + (z(3716)+z(3717))*U2;
z(1204) = z(412)*DODC2 - z(405)*DODC3;
z(1211) = z(1154) + z(1204);
z(1195) = z(193)*DODC3 + z(195)*DODC2;
z(1196) = z(23)*DODC2 - z(24)*DODC3;
z(1197) = -z(194)*DODC3 - z(196)*DODC2;
z(1198) = z(406)*DODC2 - z(399)*DODC3;
z(1205) = z(1148) + z(1198);
z(1199) = z(407)*DODC2 - z(400)*DODC3;
z(1206) = z(1149) + z(1199);
z(1200) = z(408)*DODC2 - z(401)*DODC3;
z(1207) = z(1150) + z(1200);
z(1201) = z(409)*DODC2 - z(402)*DODC3;
z(1208) = z(1151) + z(1201);
z(1203) = z(411)*DODC2 - z(404)*DODC3;
z(1210) = z(1153) + z(1203);
z(3654) = z(1211) + U19 + z(166)*U16 + z(168)*U17 + z(170)*U18 + z(1145)*U13 + z(1146)*U14 + z(1147)*U15 + z(1195)*U7 + z(1196)*U9 + z(1197)*U8 + z(1205)*U4 + z(1206)*U5 + z(1207)*U6 + z(1208)*U3 + z(1209)*U1 + z(1210)*U2;
z(3205) = z(398) + z(19)*U7 + z(20)*U8 + z(392)*U4 + z(393)*U5 + z(394)*U6 + z(395)*U3 + z(396)*U1 + z(397)*U2;
z(3743) = z(3719) + z(3207)*z(3654) - z(3205)*z(3656);
z(3721) = z(3227)*DODC1 - z(3215)*DODC2;
z(3725) = z(20)*QND_3p*DODC2 - z(223)*DODC1;
z(3726) = z(222)*DODC1 - z(19)*QND_3p*DODC2;
z(3727) = z(23)*QND_1p*DODC1;
z(3729) = z(3228)*DODC1 - z(3216)*DODC2;
z(3731) = z(3229)*DODC1 - z(3217)*DODC2;
z(3733) = z(3230)*DODC1 - z(3218)*DODC2;
z(3735) = z(3231)*DODC1 - z(3219)*DODC2;
z(3737) = z(3232)*DODC1 - z(3220)*DODC2;
z(3739) = z(3233)*DODC1 - z(3221)*DODC2;
z(3740) = z(3720) + z(3721) + z(184)*U17 + z(185)*U18 + z(186)*U16 + z(3722)*U13 + z(3723)*U14 + z(3724)*U15 + z(3725)*U7 + z(3726)*U8 + z(3727)*U9 + (z(3728)+z(3729))*U4 + (z(3730)+z(3731))*U5 + (z(3732)+z(3733))*U6 + (z(3734)+z(3735))*U3 + (z(3736)+z(3737))*U1 + (z(3738)+z(3739))*U2;
z(3742) = z(3740) + z(3205)*z(3655) - z(3206)*z(3654);
z(3751) = z(3307)*EOED2 - z(3290)*EOED3;
z(3758) = z(230)*EOED2 - z(217)*EOED3;
z(3759) = z(229)*EOED2 - z(219)*EOED3;
z(3760) = z(228)*EOED2 - z(221)*EOED3;
z(3765) = z(3309)*EOED2 - z(3292)*EOED3;
z(3770) = z(3316)*EOED2 - z(3299)*EOED3;
z(3775) = z(3308)*EOED2 - z(3291)*EOED3;

z(3780) = z(3310)*EOED2 - z(3293)*EOED3;
z(3785) = z(3311)*EOED2 - z(3294)*EOED3;
z(3790) = z(3312)*EOED2 - z(3295)*EOED3;
z(3795) = z(3313)*EOED2 - z(3296)*EOED3;
z(3800) = z(3314)*EOED2 - z(3297)*EOED3;
z(3805) = z(3315)*EOED2 - z(3298)*EOED3;
z(3806) = z(3750) + z(3751) + z(236)*U21 + z(237)*U19 + z(238)*U20 + z(3752)*U16 + z(3753)*U17 + z(3754)*U18 + z(3755)*U13 + z(3756)*U14 + z(3757)*U15 + z(3758)*U10 + z(3759)*U11 + z(3760)*U12 + (z(3764)+z(3765))*U8 + (z(3769)+z(3770))*U9 + (z(3774)+z(3775))*U7 + (z(3779)+z(3780))*U4 + (z(3784)+z(3785))*U5 + (z(3789)+z(3790))*U6 + (z(3794)+z(3795))*U3 + (z(3799)+z(3800))*U1 + (z(3804)+z(3805))*U2;
z(1825) = z(432)*EOED1 - z(422)*EOED2;
z(1881) = z(1812) + z(1825);
z(1813) = z(201)*EOED1 - z(197)*EOED2;
z(1814) = z(203)*EOED1 - z(199)*EOED2;
z(1815) = z(28)*EOED2 + z(205)*EOED1;
z(1817) = z(424)*EOED1 - z(414)*EOED2;
z(1872) = z(1803) + z(1817);
z(1824) = z(431)*EOED1 - z(421)*EOED2;
z(1873) = z(1804) + z(1824);
z(1816) = z(423)*EOED1 - z(413)*EOED2;
z(1874) = z(1805) + z(1816);
z(1818) = z(425)*EOED1 - z(415)*EOED2;
z(1875) = z(1806) + z(1818);
z(1819) = z(426)*EOED1 - z(416)*EOED2;
z(1876) = z(1807) + z(1819);
z(1820) = z(427)*EOED1 - z(417)*EOED2;
z(1877) = z(1808) + z(1820);
z(1821) = z(428)*EOED1 - z(418)*EOED2;
z(1878) = z(1809) + z(1821);
z(1823) = z(430)*EOED1 - z(420)*EOED2;
z(1880) = z(1811) + z(1823);
z(3746) = z(1881) + U24 + z(209)*U19 + z(212)*U20 + z(215)*U21 + z(1797)*U16 + z(1798)*U17 + z(1799)*U18 + z(1800)*U13 + z(1801)*U14 + z(1802)*U15 + z(1813)*U10 + z(1814)*U11 + z(1815)*U12 + z(1872)*U8 + z(1873)*U9 + z(1874)*U7 + z(1875)*U4 + z(1876)*U5 + z(1877)*U6 + z(1878)*U3 + z(1879)*U1 + z(1880)*U2;
z(1838) = z(422)*EOED3 - z(442)*EOED1;
z(1871) = z(1796) + z(1838);
z(1826) = z(197)*EOED3 - z(202)*EOED1;
z(1827) = z(199)*EOED3 - z(204)*EOED1;
z(1828) = -z(28)*EOED3 - z(206)*EOED1;
z(1830) = z(414)*EOED3 - z(434)*EOED1;
z(1862) = z(1787) + z(1830);
z(1837) = z(421)*EOED3 - z(441)*EOED1;
z(1863) = z(1788) + z(1837);
z(1829) = z(413)*EOED3 - z(433)*EOED1;
z(1864) = z(1789) + z(1829);
z(1831) = z(415)*EOED3 - z(435)*EOED1;
z(1865) = z(1790) + z(1831);
z(1832) = z(416)*EOED3 - z(436)*EOED1;
z(1866) = z(1791) + z(1832);
z(1833) = z(417)*EOED3 - z(437)*EOED1;
z(1867) = z(1792) + z(1833);
z(1834) = z(418)*EOED3 - z(438)*EOED1;
z(1868) = z(1793) + z(1834);
z(1836) = z(420)*EOED3 - z(440)*EOED1;
z(1870) = z(1795) + z(1836);
z(3745) = z(1871) + U23 + z(208)*U19 + z(211)*U20 + z(214)*U21 + z(1781)*U16 + z(1782)*U17 + z(1783)*U18 + z(1784)*U13 + z(1785)*U14 + z(1786)*U15 + z(1826)*U10 + z(1827)*U11 + z(1828)*U12 + z(1862)*U8 + z(1863)*U9 + z(1864)*U7 + z(1865)*U4 + z(1866)*U5 + z(1867)*U6 + z(1868)*U3 + z(1869)*U1 + z(1870)*U2;
z(3867) = z(3806) + z(3257)*z(3746) - z(3258)*z(3745);
z(3808) = z(3272)*EOED3 - z(3307)*EOED1;
z(3815) = z(234)*EOED3 - z(230)*EOED1;
z(3816) = z(235)*EOED3 - z(229)*EOED1;
z(3817) = -z(228)*EOED1 - z(27)*QNE_2p*EOED3;
z(3819) = z(3274)*EOED3 - z(3309)*EOED1;
z(3821) = z(3281)*EOED3 - z(3316)*EOED1;
z(3823) = z(3273)*EOED3 - z(3308)*EOED1;
z(3825) = z(3275)*EOED3 - z(3310)*EOED1;
z(3827) = z(3276)*EOED3 - z(3311)*EOED1;
z(3829) = z(3277)*EOED3 - z(3312)*EOED1;
z(3831) = z(3278)*EOED3 - z(3313)*EOED1;
z(3833) = z(3279)*EOED3 - z(3314)*EOED1;
z(3835) = z(3280)*EOED3 - z(3315)*EOED1;
z(3836) = z(3807) + z(3808) + z(220)*U19 + z(224)*U20 + z(227)*U21 + z(3809)*U16 + z(3810)*U17 + z(3811)*U18 + z(3812)*U13 + z(3813)*U14 + z(3814)*U15 + z(3815)*U10 + z(3816)*U11 + z(3817)*U12 + (z(3818)+z(3819))*U8 + (z(3820)+z(3821))*U9 + (z(3822)+z(3823))*U7 + (z(3824)+z(3825))*U4 + (z(3826)+z(3827))*U5 + (z(3828)+z(3829))*U6 + (z(3830)+z(3831))*U3 + (z(3832)+z(3833))*U1 + (z(3834)+z(3835))*U2;
z(1851) = z(442)*EOED2 - z(432)*EOED3;
z(1861) = z(1780) + z(1851);
z(1839) = z(202)*EOED2 - z(201)*EOED3;
z(1840) = z(204)*EOED2 - z(203)*EOED3;

z(1841) = z(206)*EOED2 - z(205)*EOED3;
z(1843) = z(434)*EOED2 - z(424)*EOED3;
z(1852) = z(1771) + z(1843);
z(1850) = z(441)*EOED2 - z(431)*EOED3;
z(1853) = z(1772) + z(1850);
z(1842) = z(433)*EOED2 - z(423)*EOED3;
z(1854) = z(1773) + z(1842);
z(1844) = z(435)*EOED2 - z(425)*EOED3;
z(1855) = z(1774) + z(1844);
z(1845) = z(436)*EOED2 - z(426)*EOED3;
z(1856) = z(1775) + z(1845);
z(1846) = z(437)*EOED2 - z(427)*EOED3;
z(1857) = z(1776) + z(1846);
z(1847) = z(438)*EOED2 - z(428)*EOED3;
z(1858) = z(1777) + z(1847);
z(1849) = z(440)*EOED2 - z(430)*EOED3;
z(1860) = z(1779) + z(1849);
z(3744) = z(1861) + U22 + z(207)*U19 + z(210)*U20 + z(213)*U21 + z(1765)*U16 + z(1766)*U17 + z(1767)*U18 + z(1768)*U13 + z(1769)*U14 + z(1770)*U15 + z(1839)*U10 + z(1840)*U11 + z(1841)*U12 + z(1852)*U8 + z(1853)*U9 + z(1854)*U7 + z(1855)*U4 + z(1856)*U5 + z(1857)*U6 + z(1858)*U3 + z(1859)*U1 + z(1860)*U2;
z(3869) = z(3836) + z(3258)*z(3744) - z(3256)*z(3746);
z(3838) = z(3290)*EOED1 - z(3272)*EOED2;
z(3845) = z(217)*EOED1 - z(234)*EOED2;
z(3846) = z(219)*EOED1 - z(235)*EOED2;
z(3847) = z(221)*EOED1 + z(27)*QNE_2p*EOED2;
z(3849) = z(3292)*EOED1 - z(3274)*EOED2;
z(3851) = z(3299)*EOED1 - z(3281)*EOED2;
z(3853) = z(3291)*EOED1 - z(3273)*EOED2;
z(3855) = z(3293)*EOED1 - z(3275)*EOED2;
z(3857) = z(3294)*EOED1 - z(3276)*EOED2;
z(3859) = z(3295)*EOED1 - z(3277)*EOED2;
z(3861) = z(3296)*EOED1 - z(3278)*EOED2;
z(3863) = z(3297)*EOED1 - z(3279)*EOED2;
z(3865) = z(3298)*EOED1 - z(3280)*EOED2;
z(3866) = z(3837) + z(3838) + z(231)*U20 + z(232)*U21 + z(233)*U19 + z(3839)*U16 + z(3840)*U17 + z(3841)*U18 + z(3842)*U13 + z(3843)*U14 + z(3844)*U15 + z(3845)*U10 + z(3846)*U11 + z(3847)*U12 + (z(3848)+z(3849))*U8 + (z(3850)+z(3851))*U9 + (z(3852)+z(3853))*U7 + (z(3854)+z(3855))*U4 + (z(3856)+z(3857))*U5 + (z(3858)+z(3859))*U6 + (z(3860)+z(3861))*U3 + (z(3862)+z(3863))*U1 + (z(3864)+z(3865))*U2;
z(3868) = z(3866) + z(3256)*z(3745) - z(3257)*z(3744);
z(3877) = z(3387)*HOHE2 + z(254)*z(3272)*HOHE2 + z(257)*z(3290)*HOHE2 + z(260)*z(3307)*HOHE2 + z(274)*z(432)*HOHE2 + z(275)*z(442)*HOHE2 + z(276)*z(422)*HOHE2 - z(3383)*HOHE3 - z(253)*z(3272)*HOHE3 - z(256)*z(3290)*HOHE3 - z(259)*z(3307)*HOHE3 - z(265)*z(422)*HOHE3 - z(268)*z(432)*HOHE3 - z(271)*z(442)*HOHE3;
z(3891) = z(197)*z(276)*HOHE2 + z(201)*z(274)*HOHE2 + z(202)*z(275)*HOHE2 + z(254)*z(234)*HOHE2 + z(257)*z(217)*HOHE2 + z(260)*z(230)*HOHE2 - z(197)*z(265)*HOHE3 - z(201)*z(268)*HOHE3 - z(202)*z(271)*HOHE3 - z(253)*z(234)*HOHE3 - z(256)*z(217)*HOHE3 - z(259)*z(230)*HOHE3;
z(3896) = z(199)*z(276)*HOHE2 + z(203)*z(274)*HOHE2 + z(204)*z(275)*HOHE2 + z(254)*z(235)*HOHE2 + z(257)*z(219)*HOHE2 + z(260)*z(229)*HOHE2 - z(199)*z(265)*HOHE3 - z(203)*z(268)*HOHE3 - z(204)*z(271)*HOHE3 - z(253)*z(235)*HOHE3 - z(256)*z(219)*HOHE3 - z(259)*z(229)*HOHE3;
z(3901) = z(28)*z(265)*HOHE3 + z(205)*z(274)*HOHE2 + z(206)*z(275)*HOHE2 + z(257)*z(221)*HOHE2 + z(260)*z(228)*HOHE2 + z(27)*z(253)*QNE_2p*HOHE3 - z(28)*z(276)*HOHE2 - z(205)*z(268)*HOHE3 - z(206)*z(271)*HOHE3 - z(256)*z(221)*HOHE3 - z(259)*z(228)*HOHE3 - z(27)*z(254)*QNE_2p*HOHE2;
z(3906) = z(254)*z(3274)*HOHE2 + z(257)*z(3292)*HOHE2 + z(260)*z(3309)*HOHE2 + z(414)*z(276)*HOHE2 + z(424)*z(274)*HOHE2 + z(434)*z(275)*HOHE2 - z(253)*z(3274)*HOHE3 - z(256)*z(3292)*HOHE3 - z(259)*z(3309)*HOHE3 - z(414)*z(265)*HOHE3 - z(424)*z(268)*HOHE3 - z(434)*z(271)*HOHE3;
z(3911) = z(254)*z(3281)*HOHE2 + z(257)*z(3299)*HOHE2 + z(260)*z(3316)*HOHE2 + z(421)*z(276)*HOHE2 + z(431)*z(274)*HOHE2 + z(441)*z(275)*HOHE2 - z(253)*z(3281)*HOHE3 - z(256)*z(3299)*HOHE3 - z(259)*z(3316)*HOHE3 - z(421)*z(265)*HOHE3 - z(431)*z(268)*HOHE3 - z(441)*z(271)*HOHE3;
z(3916) = z(254)*z(3273)*HOHE2 + z(257)*z(3291)*HOHE2 + z(260)*z(3308)*HOHE2 + z(413)*z(276)*HOHE2 + z(423)*z(274)*HOHE2 + z(433)*z(275)*HOHE2 - z(253)*z(3273)*HOHE3 - z(256)*z(3291)*HOHE3 - z(259)*z(3308)*HOHE3 - z(413)*z(265)*HOHE3 - z(423)*z(268)*HOHE3 - z(433)*z(271)*HOHE3;
z(3921) = z(254)*z(3275)*HOHE2 + z(257)*z(3293)*HOHE2 + z(260)*z(3310)*HOHE2 + z(415)*z(276)*HOHE2 + z(425)*z(274)*HOHE2 + z(435)*z(275)*HOHE2 - z(253)*z(3275)*HOHE3 - z(256)*z(3293)*HOHE3 - z(259)*z(3310)*HOHE3 - z(415)*z(265)*HOHE3 - z(425)*z(268)*HOHE3 - z(435)*z(271)*HOHE3;
z(3926) = z(254)*z(3276)*HOHE2 + z(257)*z(3294)*HOHE2 + z(260)*z(3311)*HOHE2 + z(416)*z(276)*HOHE2 + z(426)*z(274)*HOHE2 + z(436)*z(275)*HOHE2 - z(253)*z(3276)*HOHE3 - z(256)*z(3294)*HOHE3 - z(259)*z(3311)*HOHE3 - z(416)*z(265)*HOHE3 - z(426)*z(268)*HOHE3 - z(436)*z(271)*HOHE3;
z(3931) = z(254)*z(3277)*HOHE2 + z(257)*z(3295)*HOHE2 + z(260)*z(3312)*HOHE2 + z(417)*z(276)*HOHE2 + z(427)*z(274)*HOHE2 + z(437)*z(275)*HOHE2 - z(253)*z(3277)*HOHE3 - z(256)*z(3295)*HOHE3 - z(259)*z(3312)*HOHE3 - z(417)*z(265)*HOHE3 - z(427)*z(268)*HOHE3 - z(437)*z(271)*HOHE3;
z(3936) = z(254)*z(3278)*HOHE2 + z(257)*z(3296)*HOHE2 + z(260)*z(3313)*HOHE2 + z(418)*z(276)*HOHE2 + z(428)*z(274)*HOHE2 + z(438)*z(275)*HOHE2 - z(253)*z(3278)*HOHE3 - z(256)*z(3296)*HOHE3 - z(259)*z(3313)*HOHE3 - z(418)*z(265)*HOHE3 - z(428)*z(268)*HOHE3 - z(438)*z(271)*HOHE3;
z(3941) = z(254)*z(3279)*HOHE2 + z(257)*z(3297)*HOHE2 + z(260)*z(3314)*HOHE2 + z(419)*z(276)*HOHE2 + z(429)*z(274)*HOHE2 + z(439)*z(275)*HOHE2 - z(253)*z(3279)*HOHE3 - z(256)*z(3297)*HOHE3 - z(259)*z(3314)*HOHE3 - z(419)*z(265)*HOHE3 - z(429)*z(268)*HOHE3 - z(439)*z(271)*HOHE3;
z(3946) = z(254)*z(3280)*HOHE2 + z(257)*z(3298)*HOHE2 + z(260)*z(3315)*HOHE2 + z(420)*z(276)*HOHE2 + z(430)*z(274)*HOHE2 + z(440)*z(275)*HOHE2 - z(253)*z(3280)*HOHE3 - z(256)*z(3298)*HOHE3 - z(259)*z(3315)*HOHE3 - z(420)*z(265)*HOHE3 - z(430)*z(268)*HOHE3 - z(440)*z(271)*HOHE3;

z(3947) = z(3876) + z(3877) + z(277)*U24 + z(278)*U22 + z(279)*U23 + z(3878)*U19 + z(3879)*U20 + z(3880)*U21 + z(3881)*U16 + z(3882)*U17 + z(3883)*U18 + z(3884)*U13 + z(3885)*U14 + z(3886)*U15 + (z(3890)+z(3891))*U10 + (z(3895)+z(3896))*U11 + (z(3900)+z(3901))*U12 + (z(3905)+z(3906))*U8 + (z(3910)+z(3911))*U9 + (z(3915)+z(3916))*U7 + (z(3920)+z(3921))*U4 + (z(3925)+z(3926))*U5 + (z(3930)+z(3931))*U6 + (z(3935)+z(3936))*U3 + (z(3940)+z(3941))*U1 + (z(3945)+z(3946))*U2;
z(2463) = z(281)*HOHE1 + z(253)*z(422)*HOHE1 + z(256)*z(432)*HOHE1 + z(259)*z(442)*HOHE1 - z(280)*HOHE2 - z(252)*z(422)*HOHE2 - z(255)*z(432)*HOHE2 - z(258)*z(442)*HOHE2;
z(2528) = z(2450) + z(2463);
z(2451) = z(197)*z(253)*HOHE1 + z(201)*z(256)*HOHE1 + z(202)*z(259)*HOHE1 - z(197)*z(252)*HOHE2 - z(201)*z(255)*HOHE2 - z(202)*z(258)*HOHE2;
z(2516) = z(2438) + z(2451);
z(2452) = z(199)*z(253)*HOHE1 + z(203)*z(256)*HOHE1 + z(204)*z(259)*HOHE1 - z(199)*z(252)*HOHE2 - z(203)*z(255)*HOHE2 - z(204)*z(258)*HOHE2;
z(2517) = z(2439) + z(2452);
z(2453) = z(28)*z(252)*HOHE2 + z(205)*z(256)*HOHE1 + z(206)*z(259)*HOHE1 - z(28)*z(253)*HOHE1 - z(205)*z(255)*HOHE2 - z(206)*z(258)*HOHE2;
z(2518) = z(2440) + z(2453);
z(2455) = z(253)*z(414)*HOHE1 + z(256)*z(424)*HOHE1 + z(259)*z(434)*HOHE1 - z(252)*z(414)*HOHE2 - z(255)*z(424)*HOHE2 - z(258)*z(434)*HOHE2;
z(2519) = z(2441) + z(2455);
z(2462) = z(253)*z(421)*HOHE1 + z(256)*z(431)*HOHE1 + z(259)*z(441)*HOHE1 - z(252)*z(421)*HOHE2 - z(255)*z(431)*HOHE2 - z(258)*z(441)*HOHE2;
z(2520) = z(2442) + z(2462);
z(2454) = z(253)*z(413)*HOHE1 + z(256)*z(423)*HOHE1 + z(259)*z(433)*HOHE1 - z(252)*z(413)*HOHE2 - z(255)*z(423)*HOHE2 - z(258)*z(433)*HOHE2;
z(2521) = z(2443) + z(2454);
z(2456) = z(253)*z(415)*HOHE1 + z(256)*z(425)*HOHE1 + z(259)*z(435)*HOHE1 - z(252)*z(415)*HOHE2 - z(255)*z(425)*HOHE2 - z(258)*z(435)*HOHE2;
z(2522) = z(2444) + z(2456);
z(2457) = z(253)*z(416)*HOHE1 + z(256)*z(426)*HOHE1 + z(259)*z(436)*HOHE1 - z(252)*z(416)*HOHE2 - z(255)*z(426)*HOHE2 - z(258)*z(436)*HOHE2;
z(2523) = z(2445) + z(2457);
z(2458) = z(253)*z(417)*HOHE1 + z(256)*z(427)*HOHE1 + z(259)*z(437)*HOHE1 - z(252)*z(417)*HOHE2 - z(255)*z(427)*HOHE2 - z(258)*z(437)*HOHE2;
z(2524) = z(2446) + z(2458);
z(2459) = z(253)*z(418)*HOHE1 + z(256)*z(428)*HOHE1 + z(259)*z(438)*HOHE1 - z(252)*z(418)*HOHE2 - z(255)*z(428)*HOHE2 - z(258)*z(438)*HOHE2;
z(2525) = z(2447) + z(2459);
z(2461) = z(253)*z(420)*HOHE1 + z(256)*z(430)*HOHE1 + z(259)*z(440)*HOHE1 - z(252)*z(420)*HOHE2 - z(255)*z(430)*HOHE2 - z(258)*z(440)*HOHE2;
z(2527) = z(2449) + z(2461);
z(3872) = z(2528) + z(254)*U22 + z(257)*U23 + z(260)*U24 + z(2429)*U19 + z(2430)*U20 + z(2431)*U21 + z(2432)*U16 + z(2433)*U17 + z(2434)*U18 + z(2435)*U13 + z(2436)*U14 + z(2437)*U15 + z(2516)*U10 + z(2517)*U11 + z(2518)*U12 + z(2519)*U8 + z(2520)*U9 + z(2521)*U7 + z(2522)*U4 + z(2523)*U5 + z(2524)*U6 + z(2525)*U3 + z(2526)*U1 + z(2527)*U2;
z(2476) = z(280)*HOHE3 + z(252)*z(422)*HOHE3 + z(255)*z(432)*HOHE3 + z(258)*z(442)*HOHE3 - z(282)*HOHE1 - z(254)*z(422)*HOHE1 - z(257)*z(432)*HOHE1 - z(260)*z(442)*HOHE1;
z(2515) = z(2428) + z(2476);
z(2464) = z(197)*z(252)*HOHE3 + z(201)*z(255)*HOHE3 + z(202)*z(258)*HOHE3 - z(197)*z(254)*HOHE1 - z(201)*z(257)*HOHE1 - z(202)*z(260)*HOHE1;
z(2503) = z(2416) + z(2464);
z(2465) = z(199)*z(252)*HOHE3 + z(203)*z(255)*HOHE3 + z(204)*z(258)*HOHE3 - z(199)*z(254)*HOHE1 - z(203)*z(257)*HOHE1 - z(204)*z(260)*HOHE1;
z(2504) = z(2417) + z(2465);
z(2466) = z(28)*z(254)*HOHE1 + z(205)*z(255)*HOHE3 + z(206)*z(258)*HOHE3 - z(28)*z(252)*HOHE3 - z(205)*z(257)*HOHE1 - z(206)*z(260)*HOHE1;
z(2505) = z(2418) + z(2466);
z(2468) = z(252)*z(414)*HOHE3 + z(255)*z(424)*HOHE3 + z(258)*z(434)*HOHE3 - z(254)*z(414)*HOHE1 - z(257)*z(424)*HOHE1 - z(260)*z(434)*HOHE1;
z(2506) = z(2419) + z(2468);
z(2475) = z(252)*z(421)*HOHE3 + z(255)*z(431)*HOHE3 + z(258)*z(441)*HOHE3 - z(254)*z(421)*HOHE1 - z(257)*z(431)*HOHE1 - z(260)*z(441)*HOHE1;
z(2507) = z(2420) + z(2475);
z(2467) = z(252)*z(413)*HOHE3 + z(255)*z(423)*HOHE3 + z(258)*z(433)*HOHE3 - z(254)*z(413)*HOHE1 - z(257)*z(423)*HOHE1 - z(260)*z(433)*HOHE1;
z(2508) = z(2421) + z(2467);
z(2469) = z(252)*z(415)*HOHE3 + z(255)*z(425)*HOHE3 + z(258)*z(435)*HOHE3 - z(254)*z(415)*HOHE1 - z(257)*z(425)*HOHE1 - z(260)*z(435)*HOHE1;
z(2509) = z(2422) + z(2469);
z(2470) = z(252)*z(416)*HOHE3 + z(255)*z(426)*HOHE3 + z(258)*z(436)*HOHE3 - z(254)*z(416)*HOHE1 - z(257)*z(426)*HOHE1 - z(260)*z(436)*HOHE1;
z(2510) = z(2423) + z(2470);
z(2471) = z(252)*z(417)*HOHE3 + z(255)*z(427)*HOHE3 + z(258)*z(437)*HOHE3 - z(254)*z(417)*HOHE1 - z(257)*z(427)*HOHE1 - z(260)*z(437)*HOHE1;
z(2511) = z(2424) + z(2471);
z(2472) = z(252)*z(418)*HOHE3 + z(255)*z(428)*HOHE3 + z(258)*z(438)*HOHE3 - z(254)*z(418)*HOHE1 - z(257)*z(428)*HOHE1 - z(260)*z(438)*HOHE1;
z(2512) = z(2425) + z(2472);
z(2474) = z(252)*z(420)*HOHE3 + z(255)*z(430)*HOHE3 + z(258)*z(440)*HOHE3 - z(254)*z(420)*HOHE1 - z(257)*z(430)*HOHE1 - z(260)*z(440)*HOHE1;

z(2514) = z(2427) + z(2474);
z(3871) = z(2515) + z(253)*U22 + z(256)*U23 + z(259)*U24 + z(2407)*U19 + z(2408)*U20 + z(2409)*U21 + z(2410)*U16 + z(2411)*U17 + z(2412)*U18 + z(2413)*U13 + z(2414)*U14 + z(2415)*U15 + z(2503)*U10 + z(2504)*U11 + z(2505)*U12 + z(2506)*U8 + z(2507)*U9 + z(2508)*U7 + z(2509)*U4 + z(2510)*U5 + z(2511)*U6 + z(2512)*U3 + z(2513)*U1 + z(2514)*U2;
z(4020) = z(3947) + z(3872)*(z(281)+z(253)*z(3256)+z(256)*z(3257)+z(259)*z(3258)) - z(3871)*(z(282)+z(254)*z(3256)+z(257)*z(3257)+z(260)*z(3258));
z(3949) = z(3377)*HOHE3 + z(252)*z(3272)*HOHE3 + z(255)*z(3290)*HOHE3 + z(258)*z(3307)*HOHE3 + z(277)*z(442)*HOHE3 + z(278)*z(422)*HOHE3 + z(279)*z(432)*HOHE3 - z(3387)*HOHE1 - z(254)*z(3272)*HOHE1 - z(257)*z(3290)*HOHE1 - z(260)*z(3307)*HOHE1 - z(274)*z(432)*HOHE1 - z(275)*z(442)*HOHE1 - z(276)*z(422)*HOHE1;
z(3960) = z(197)*z(278)*HOHE3 + z(201)*z(279)*HOHE3 + z(202)*z(277)*HOHE3 + z(252)*z(234)*HOHE3 + z(255)*z(217)*HOHE3 + z(258)*z(230)*HOHE3 - z(197)*z(276)*HOHE1 - z(201)*z(274)*HOHE1 - z(202)*z(275)*HOHE1 - z(254)*z(234)*HOHE1 - z(257)*z(217)*HOHE1 - z(260)*z(230)*HOHE1;
z(3962) = z(199)*z(278)*HOHE3 + z(203)*z(279)*HOHE3 + z(204)*z(277)*HOHE3 + z(252)*z(235)*HOHE3 + z(255)*z(219)*HOHE3 + z(258)*z(229)*HOHE3 - z(199)*z(276)*HOHE1 - z(203)*z(274)*HOHE1 - z(204)*z(275)*HOHE1 - z(254)*z(235)*HOHE1 - z(257)*z(219)*HOHE1 - z(260)*z(229)*HOHE1;
z(3964) = z(28)*z(276)*HOHE1 + z(205)*z(279)*HOHE3 + z(206)*z(277)*HOHE3 + z(255)*z(221)*HOHE3 + z(258)*z(228)*HOHE3 + z(27)*z(254)*QNE_2p*HOHE1 - z(28)*z(278)*HOHE3 - z(205)*z(274)*HOHE1 - z(206)*z(275)*HOHE1 - z(257)*z(221)*HOHE1 - z(260)*z(228)*HOHE1 - z(27)*z(252)*QNE_2p*HOHE3;
z(3966) = z(252)*z(3274)*HOHE3 + z(255)*z(3292)*HOHE3 + z(258)*z(3309)*HOHE3 + z(414)*z(278)*HOHE3 + z(424)*z(279)*HOHE3 + z(434)*z(277)*HOHE3 - z(254)*z(3274)*HOHE1 - z(257)*z(3292)*HOHE1 - z(260)*z(3309)*HOHE1 - z(414)*z(276)*HOHE1 - z(424)*z(274)*HOHE1 - z(434)*z(275)*HOHE1;
z(3968) = z(252)*z(3281)*HOHE3 + z(255)*z(3299)*HOHE3 + z(258)*z(3316)*HOHE3 + z(421)*z(278)*HOHE3 + z(431)*z(279)*HOHE3 + z(441)*z(277)*HOHE3 - z(254)*z(3281)*HOHE1 - z(257)*z(3299)*HOHE1 - z(260)*z(3316)*HOHE1 - z(421)*z(276)*HOHE1 - z(431)*z(274)*HOHE1 - z(441)*z(275)*HOHE1;
z(3970) = z(252)*z(3273)*HOHE3 + z(255)*z(3291)*HOHE3 + z(258)*z(3308)*HOHE3 + z(413)*z(278)*HOHE3 + z(423)*z(279)*HOHE3 + z(433)*z(277)*HOHE3 - z(254)*z(3273)*HOHE1 - z(257)*z(3291)*HOHE1 - z(260)*z(3308)*HOHE1 - z(413)*z(276)*HOHE1 - z(423)*z(274)*HOHE1 - z(433)*z(275)*HOHE1;
z(3972) = z(252)*z(3275)*HOHE3 + z(255)*z(3293)*HOHE3 + z(258)*z(3310)*HOHE3 + z(415)*z(278)*HOHE3 + z(425)*z(279)*HOHE3 + z(435)*z(277)*HOHE3 - z(254)*z(3275)*HOHE1 - z(257)*z(3293)*HOHE1 - z(260)*z(3310)*HOHE1 - z(415)*z(276)*HOHE1 - z(425)*z(274)*HOHE1 - z(435)*z(275)*HOHE1;
z(3974) = z(252)*z(3276)*HOHE3 + z(255)*z(3294)*HOHE3 + z(258)*z(3311)*HOHE3 + z(416)*z(278)*HOHE3 + z(426)*z(279)*HOHE3 + z(436)*z(277)*HOHE3 - z(254)*z(3276)*HOHE1 - z(257)*z(3294)*HOHE1 - z(260)*z(3311)*HOHE1 - z(416)*z(276)*HOHE1 - z(426)*z(274)*HOHE1 - z(436)*z(275)*HOHE1;
z(3976) = z(252)*z(3277)*HOHE3 + z(255)*z(3295)*HOHE3 + z(258)*z(3312)*HOHE3 + z(417)*z(278)*HOHE3 + z(427)*z(279)*HOHE3 + z(437)*z(277)*HOHE3 - z(254)*z(3277)*HOHE1 - z(257)*z(3295)*HOHE1 - z(260)*z(3312)*HOHE1 - z(417)*z(276)*HOHE1 - z(427)*z(274)*HOHE1 - z(437)*z(275)*HOHE1;
z(3978) = z(252)*z(3278)*HOHE3 + z(255)*z(3296)*HOHE3 + z(258)*z(3313)*HOHE3 + z(418)*z(278)*HOHE3 + z(428)*z(279)*HOHE3 + z(438)*z(277)*HOHE3 - z(254)*z(3278)*HOHE1 - z(257)*z(3296)*HOHE1 - z(260)*z(3313)*HOHE1 - z(418)*z(276)*HOHE1 - z(428)*z(274)*HOHE1 - z(438)*z(275)*HOHE1;
z(3980) = z(252)*z(3279)*HOHE3 + z(255)*z(3297)*HOHE3 + z(258)*z(3314)*HOHE3 + z(419)*z(278)*HOHE3 + z(429)*z(279)*HOHE3 + z(439)*z(277)*HOHE3 - z(254)*z(3279)*HOHE1 - z(257)*z(3297)*HOHE1 - z(260)*z(3314)*HOHE1 - z(419)*z(276)*HOHE1 - z(429)*z(274)*HOHE1 - z(439)*z(275)*HOHE1;
z(3982) = z(252)*z(3280)*HOHE3 + z(255)*z(3298)*HOHE3 + z(258)*z(3315)*HOHE3 + z(420)*z(278)*HOHE3 + z(430)*z(279)*HOHE3 + z(440)*z(277)*HOHE3 - z(254)*z(3280)*HOHE1 - z(257)*z(3298)*HOHE1 - z(260)*z(3315)*HOHE1 - z(420)*z(276)*HOHE1 - z(430)*z(274)*HOHE1 - z(440)*z(275)*HOHE1;
z(3983) = z(3948) + z(3949) + z(265)*U22 + z(268)*U23 + z(271)*U24 + z(3950)*U19 + z(3951)*U20 + z(3952)*U21 + z(3953)*U16 + z(3954)*U17 + z(3955)*U18 + z(3956)*U13 + z(3957)*U14 + z(3958)*U15 + (z(3959)+z(3960))*U10 + (z(3961)+z(3962))*U11 + (z(3963)+z(3964))*U12 + (z(3965)+z(3966))*U8 + (z(3967)+z(3968))*U9 + (z(3969)+z(3970))*U7 + (z(3971)+z(3972))*U4 + (z(3973)+z(3974))*U5 + (z(3975)+z(3976))*U6 + (z(3977)+z(3978))*U3 + (z(3979)+z(3980))*U1 + (z(3981)+z(3982))*U2;
z(2489) = z(282)*HOHE2 + z(254)*z(422)*HOHE2 + z(257)*z(432)*HOHE2 + z(260)*z(442)*HOHE2 - z(281)*HOHE3 - z(253)*z(422)*HOHE3 - z(256)*z(432)*HOHE3 - z(259)*z(442)*HOHE3;
z(2502) = z(2406) + z(2489);
z(2477) = z(197)*z(254)*HOHE2 + z(201)*z(257)*HOHE2 + z(202)*z(260)*HOHE2 - z(197)*z(253)*HOHE3 - z(201)*z(256)*HOHE3 - z(202)*z(259)*HOHE3;
z(2490) = z(2394) + z(2477);
z(2478) = z(199)*z(254)*HOHE2 + z(203)*z(257)*HOHE2 + z(204)*z(260)*HOHE2 - z(199)*z(253)*HOHE3 - z(203)*z(256)*HOHE3 - z(204)*z(259)*HOHE3;
z(2491) = z(2395) + z(2478);
z(2479) = z(28)*z(253)*HOHE3 + z(205)*z(257)*HOHE2 + z(206)*z(260)*HOHE2 - z(28)*z(254)*HOHE2 - z(205)*z(256)*HOHE3 - z(206)*z(259)*HOHE3;
z(2492) = z(2396) + z(2479);
z(2481) = z(254)*z(414)*HOHE2 + z(257)*z(424)*HOHE2 + z(260)*z(434)*HOHE2 - z(253)*z(414)*HOHE3 - z(256)*z(424)*HOHE3 - z(259)*z(434)*HOHE3;
z(2493) = z(2397) + z(2481);
z(2488) = z(254)*z(421)*HOHE2 + z(257)*z(431)*HOHE2 + z(260)*z(441)*HOHE2 - z(253)*z(421)*HOHE3 - z(256)*z(431)*HOHE3 - z(259)*z(441)*HOHE3;
z(2494) = z(2398) + z(2488);
z(2480) = z(254)*z(413)*HOHE2 + z(257)*z(423)*HOHE2 + z(260)*z(433)*HOHE2 - z(253)*z(413)*HOHE3 - z(256)*z(423)*HOHE3 - z(259)*z(433)*HOHE3;
z(2495) = z(2399) + z(2480);
z(2482) = z(254)*z(415)*HOHE2 + z(257)*z(425)*HOHE2 + z(260)*z(435)*HOHE2 - z(253)*z(415)*HOHE3 - z(256)*z(425)*HOHE3 - z(259)*z(435)*HOHE3;
z(2496) = z(2400) + z(2482);
z(2483) = z(254)*z(416)*HOHE2 + z(257)*z(426)*HOHE2 + z(260)*z(436)*HOHE2 - z(253)*z(416)*HOHE3 - z(256)*z(426)*HOHE3 - z(259)*z(436)*HOHE3;
z(2497) = z(2401) + z(2483);
z(2484) = z(254)*z(417)*HOHE2 + z(257)*z(427)*HOHE2 + z(260)*z(437)*HOHE2 - z(253)*z(417)*HOHE3 - z(256)*z(427)*HOHE3 - z(259)*z(437)*HOHE3;

z(2498) = z(2402) + z(2484);
z(2485) = z(254)*z(418)*HOHE2 + z(257)*z(428)*HOHE2 + z(260)*z(438)*HOHE2 - z(253)*z(418)*HOHE3 - z(256)*z(428)*HOHE3 - z(259)*z(438)*HOHE3;
z(2499) = z(2403) + z(2485);
z(2487) = z(254)*z(420)*HOHE2 + z(257)*z(430)*HOHE2 + z(260)*z(440)*HOHE2 - z(253)*z(420)*HOHE3 - z(256)*z(430)*HOHE3 - z(259)*z(440)*HOHE3;
z(2501) = z(2405) + z(2487);
z(3870) = z(2502) + z(252)*U22 + z(255)*U23 + z(258)*U24 + z(2385)*U19 + z(2386)*U20 + z(2387)*U21 + z(2388)*U16 + z(2389)*U17 + z(2390)*U18 + z(2391)*U13 + z(2392)*U14 + z(2393)*U15 + z(2490)*U10 + z(2491)*U11 + z(2492)*U12 + z(2493)*U8 + z(2494)*U9 + z(2495)*U7 + z(2496)*U4 + z(2497)*U5 + z(2498)*U6 + z(2499)*U3 + z(2500)*U1 + z(2501)*U2;
z(4022) = z(3983) + z(3870)*(z(282)+z(254)*z(3256)+z(257)*z(3257)+z(260)*z(3258)) - z(3872)*(z(280)+z(252)*z(3256)+z(255)*z(3257)+z(258)*z(3258));
z(3985) = z(3383)*HOHE1 + z(253)*z(3272)*HOHE1 + z(256)*z(3290)*HOHE1 + z(259)*z(3307)*HOHE1 + z(265)*z(422)*HOHE1 + z(268)*z(432)*HOHE1 + z(271)*z(442)*HOHE1 - z(3377)*HOHE2 - z(252)*z(3272)*HOHE2 - z(255)*z(3290)*HOHE2 - z(258)*z(3307)*HOHE2 - z(277)*z(442)*HOHE2 - z(278)*z(422)*HOHE2 - z(279)*z(432)*HOHE2;
z(3996) = z(197)*z(265)*HOHE1 + z(201)*z(268)*HOHE1 + z(202)*z(271)*HOHE1 + z(253)*z(234)*HOHE1 + z(256)*z(217)*HOHE1 + z(259)*z(230)*HOHE1 - z(197)*z(278)*HOHE2 - z(201)*z(279)*HOHE2 - z(202)*z(277)*HOHE2 - z(252)*z(234)*HOHE2 - z(255)*z(217)*HOHE2 - z(258)*z(230)*HOHE2;
z(3998) = z(199)*z(265)*HOHE1 + z(203)*z(268)*HOHE1 + z(204)*z(271)*HOHE1 + z(253)*z(235)*HOHE1 + z(256)*z(219)*HOHE1 + z(259)*z(229)*HOHE1 - z(199)*z(278)*HOHE2 - z(203)*z(279)*HOHE2 - z(204)*z(277)*HOHE2 - z(252)*z(235)*HOHE2 - z(255)*z(219)*HOHE2 - z(258)*z(229)*HOHE2;
z(4000) = z(28)*z(278)*HOHE2 + z(205)*z(268)*HOHE1 + z(206)*z(271)*HOHE1 + z(256)*z(221)*HOHE1 + z(259)*z(228)*HOHE1 + z(27)*z(252)*QNE_2p*HOHE2 - z(28)*z(265)*HOHE1 - z(205)*z(279)*HOHE2 - z(206)*z(277)*HOHE2 - z(255)*z(221)*HOHE2 - z(258)*z(228)*HOHE2 - z(27)*z(253)*QNE_2p*HOHE1;
z(4002) = z(253)*z(3274)*HOHE1 + z(256)*z(3292)*HOHE1 + z(259)*z(3309)*HOHE1 + z(414)*z(265)*HOHE1 + z(424)*z(268)*HOHE1 + z(434)*z(271)*HOHE1 - z(252)*z(3274)*HOHE2 - z(255)*z(3292)*HOHE2 - z(258)*z(3309)*HOHE2 - z(414)*z(278)*HOHE2 - z(424)*z(279)*HOHE2 - z(434)*z(277)*HOHE2;
z(4004) = z(253)*z(3281)*HOHE1 + z(256)*z(3299)*HOHE1 + z(259)*z(3316)*HOHE1 + z(421)*z(265)*HOHE1 + z(431)*z(268)*HOHE1 + z(441)*z(271)*HOHE1 - z(252)*z(3281)*HOHE2 - z(255)*z(3299)*HOHE2 - z(258)*z(3316)*HOHE2 - z(421)*z(278)*HOHE2 - z(431)*z(279)*HOHE2 - z(441)*z(277)*HOHE2;
z(4006) = z(253)*z(3273)*HOHE1 + z(256)*z(3291)*HOHE1 + z(259)*z(3308)*HOHE1 + z(413)*z(265)*HOHE1 + z(423)*z(268)*HOHE1 + z(433)*z(271)*HOHE1 - z(252)*z(3273)*HOHE2 - z(255)*z(3291)*HOHE2 - z(258)*z(3308)*HOHE2 - z(413)*z(278)*HOHE2 - z(423)*z(279)*HOHE2 - z(433)*z(277)*HOHE2;
z(4008) = z(253)*z(3275)*HOHE1 + z(256)*z(3293)*HOHE1 + z(259)*z(3310)*HOHE1 + z(415)*z(265)*HOHE1 + z(425)*z(268)*HOHE1 + z(435)*z(271)*HOHE1 - z(252)*z(3275)*HOHE2 - z(255)*z(3293)*HOHE2 - z(258)*z(3310)*HOHE2 - z(415)*z(278)*HOHE2 - z(425)*z(279)*HOHE2 - z(435)*z(277)*HOHE2;
z(4010) = z(253)*z(3276)*HOHE1 + z(256)*z(3294)*HOHE1 + z(259)*z(3311)*HOHE1 + z(416)*z(265)*HOHE1 + z(426)*z(268)*HOHE1 + z(436)*z(271)*HOHE1 - z(252)*z(3276)*HOHE2 - z(255)*z(3294)*HOHE2 - z(258)*z(3311)*HOHE2 - z(416)*z(278)*HOHE2 - z(426)*z(279)*HOHE2 - z(436)*z(277)*HOHE2;
z(4012) = z(253)*z(3277)*HOHE1 + z(256)*z(3295)*HOHE1 + z(259)*z(3312)*HOHE1 + z(417)*z(265)*HOHE1 + z(427)*z(268)*HOHE1 + z(437)*z(271)*HOHE1 - z(252)*z(3277)*HOHE2 - z(255)*z(3295)*HOHE2 - z(258)*z(3312)*HOHE2 - z(417)*z(278)*HOHE2 - z(427)*z(279)*HOHE2 - z(437)*z(277)*HOHE2;
z(4014) = z(253)*z(3278)*HOHE1 + z(256)*z(3296)*HOHE1 + z(259)*z(3313)*HOHE1 + z(418)*z(265)*HOHE1 + z(428)*z(268)*HOHE1 + z(438)*z(271)*HOHE1 - z(252)*z(3278)*HOHE2 - z(255)*z(3296)*HOHE2 - z(258)*z(3313)*HOHE2 - z(418)*z(278)*HOHE2 - z(428)*z(279)*HOHE2 - z(438)*z(277)*HOHE2;
z(4016) = z(253)*z(3279)*HOHE1 + z(256)*z(3297)*HOHE1 + z(259)*z(3314)*HOHE1 + z(419)*z(265)*HOHE1 + z(429)*z(268)*HOHE1 + z(439)*z(271)*HOHE1 - z(252)*z(3279)*HOHE2 - z(255)*z(3297)*HOHE2 - z(258)*z(3314)*HOHE2 - z(419)*z(278)*HOHE2 - z(429)*z(279)*HOHE2 - z(439)*z(277)*HOHE2;
z(4018) = z(253)*z(3280)*HOHE1 + z(256)*z(3298)*HOHE1 + z(259)*z(3315)*HOHE1 + z(420)*z(265)*HOHE1 + z(430)*z(268)*HOHE1 + z(440)*z(271)*HOHE1 - z(252)*z(3280)*HOHE2 - z(255)*z(3298)*HOHE2 - z(258)*z(3315)*HOHE2 - z(420)*z(278)*HOHE2 - z(430)*z(279)*HOHE2 - z(440)*z(277)*HOHE2;
z(4019) = z(3984) + z(3985) + z(274)*U23 + z(275)*U24 + z(276)*U22 + z(3986)*U19 + z(3987)*U20 + z(3988)*U21 + z(3989)*U16 + z(3990)*U17 + z(3991)*U18 + z(3992)*U13 + z(3993)*U14 + z(3994)*U15 + (z(3995)+z(3996))*U10 + (z(3997)+z(3998))*U11 + (z(3999)+z(4000))*U12 + (z(4001)+z(4002))*U8 + (z(4003)+z(4004))*U9 + (z(4005)+z(4006))*U7 + (z(4007)+z(4008))*U4 + (z(4009)+z(4010))*U5 + (z(4011)+z(4012))*U6 + (z(4013)+z(4014))*U3 + (z(4015)+z(4016))*U1 + (z(4017)+z(4018))*U2;
z(4021) = z(4019) + z(3871)*(z(280)+z(252)*z(3256)+z(255)*z(3257)+z(258)*z(3258)) - z(3870)*(z(281)+z(253)*z(3256)+z(256)*z(3257)+z(259)*z(3258));
z(4068) = z(4031) + z(277)*U24 + z(278)*U22 + z(279)*U23 + z(3878)*U19 + z(3879)*U20 + z(3880)*U21 + z(3881)*U16 + z(3882)*U17 + z(3883)*U18 + z(3884)*U13 + z(3885)*U14 + z(3886)*U15 + z(4034)*U10 + z(4037)*U11 + z(4040)*U12 + z(4043)*U8 + z(4046)*U9 + z(4049)*U7 + z(4052)*U4 + z(4055)*U5 + z(4058)*U6 + z(4061)*U3 + z(4064)*U1 + z(4067)*U2;
z(4025) = z(2729) + z(254)*U22 + z(257)*U23 + z(260)*U24 + z(2429)*U19 + z(2430)*U20 + z(2431)*U21 + z(2432)*U16 + z(2433)*U17 + z(2434)*U18 + z(2435)*U13 + z(2436)*U14 + z(2437)*U15 + z(2717)*U10 + z(2718)*U11 + z(2719)*U12 + z(2720)*U8 + z(2721)*U9 + z(2722)*U7 + z(2723)*U4 + z(2724)*U5 + z(2725)*U6 + z(2726)*U3 + z(2727)*U1 + z(2728)*U2;
z(4024) = z(2716) + z(253)*U22 + z(256)*U23 + z(259)*U24 + z(2407)*U19 + z(2408)*U20 + z(2409)*U21 + z(2410)*U16 + z(2411)*U17 + z(2412)*U18 + z(2413)*U13 + z(2414)*U14 + z(2415)*U15 + z(2704)*U10 + z(2705)*U11 + z(2706)*U12 + z(2707)*U8 + z(2708)*U9 + z(2709)*U7 + z(2710)*U4 + z(2711)*U5 + z(2712)*U6 + z(2713)*U3 + z(2714)*U1 + z(2715)*U2;
z(4191) = z(4068) + z(4025)*(z(281)+z(253)*z(3256)+z(256)*z(3257)+z(259)*z(3258)) - z(4024)*(z(282)+z(254)*z(3256)+z(257)*z(3257)+z(260)*z(3258));
z(4190) = z(4177) + z(4178)*U10 + z(4179)*U11 + z(4180)*U12 + z(4181)*U7 + z(4182)*U8 + z(4183)*U4 + z(4184)*U5 + z(4185)*U6 + z(4186)*U3 + z(4187)*U1 + z(4188)*U2 + z(4189)*U9;
z(4176) = z(4163) + z(4164)*U10 + z(4165)*U11 + z(4166)*U12 + z(4167)*U7 + z(4168)*U8 + z(4169)*U4 + z(4170)*U5 + z(4171)*U6 + z(4172)*U3 + z(4173)*U1 + z(4174)*U2 + z(4175)*U9;
z(4162) = z(4149) + z(4150)*U10 + z(4151)*U11 + z(4152)*U12 + z(4153)*U7 + z(4154)*U8 + z(4155)*U4 + z(4156)*U5 + z(4157)*U6 + z(4158)*U3 + z(4159)*U1 + z(4160)*U2 + z(4161)*U9;
z(4108) = z(4071) + z(265)*U22 + z(268)*U23 + z(271)*U24 + z(3950)*U19 + z(3951)*U20 + z(3952)*U21 + z(3953)*U16 + z(3954)*U17 + z(3955)*U18 + z(3956)*U13 + z(3957)*U14 + z(3958)*U15 + z(4074)*U10 + z(4077)*U11 + z(4080)*U12 + z(4083)*U8 + z(4086)*U9 + z(4089)*U7 + z(4092)*U4 + z(4095)*U5 + z(4098)*U6 + z(4101)*U3 + z(4104)*U1 + z(4107)*U2;

z(4023) = z(2703) + z(252)*U22 + z(255)*U23 + z(258)*U24 + z(2385)*U19 + z(2386)*U20 + z(2387)*U21 + z(2388)*U16 + z(2389)*U17 +
z(2390)*U18 + z(2391)*U13 + z(2392)*U14 + z(2393)*U15 + z(2691)*U10 + z(2692)*U11 + z(2693)*U12 + z(2694)*U8 + z(2695)*U9 + z(2696)*U7
+ z(2697)*U4 + z(2698)*U5 + z(2699)*U6 + z(2700)*U3 + z(2701)*U1 + z(2702)*U2;
z(4193) = z(4108) + z(4023)*(z(282)+z(254)*z(3256)+z(257)*z(3257)+z(260)*z(3258)) -
z(4025)*(z(280)+z(252)*z(3256)+z(255)*z(3257)+z(258)*z(3258));
z(4148) = z(4111) + z(274)*U23 + z(275)*U24 + z(276)*U22 + z(3986)*U19 + z(3987)*U20 + z(3988)*U21 + z(3989)*U16 + z(3990)*U17 +
z(3991)*U18 + z(3992)*U13 + z(3993)*U14 + z(3994)*U15 + z(4114)*U10 + z(4117)*U11 + z(4120)*U12 + z(4123)*U8 + z(4126)*U9 + z(4129)*U7
+ z(4132)*U4 + z(4135)*U5 + z(4138)*U6 + z(4141)*U3 + z(4144)*U1 + z(4147)*U2;
z(4192) = z(4148) + z(4024)*(z(280)+z(252)*z(3256)+z(255)*z(3257)+z(258)*z(3258)) -
z(4023)*(z(281)+z(253)*z(3256)+z(256)*z(3257)+z(259)*z(3258));
z(4373) = z(113)*z(3149) + z(115)*z(3148) + z(117)*z(3147) + z(381)*z(3204) + z(385)*z(3203) + z(389)*z(3202) + z(396)*z(3255) + z(403)*z(3254)
+ z(410)*z(3253) + z(419)*z(3326) + z(429)*z(3325) + z(439)*z(3324) + z(3336)*z(3402) + z(3349)*z(3401) + z(3362)*z(3400) + z(3412)*z(3487) +
z(3425)*z(3486) + z(3438)*z(3485) + z(3497)*z(3569) + z(3510)*z(3568) + z(3523)*z(3567) + z(113)*z(3144) + z(115)*z(3145) + z(117)*z(3146) +
z(381)*z(3199) + z(385)*z(3200) + z(389)*z(3201) + z(396)*z(3250) + z(403)*z(3251) + z(410)*z(3252) + z(419)*z(3321) + z(429)*z(3322) +
z(439)*z(3323) + z(3336)*z(3397) + z(3349)*z(3398) + z(3362)*z(3399) + z(3412)*z(3482) + z(3425)*z(3483) + z(3438)*z(3484) + z(3497)*z(3564) +
z(3510)*z(3565) + z(3523)*z(3566) + SW*(z(2827)*z(4372)+z(2912)*z(4370)+z(2925)*z(4371)) +
0.1428571428571429*SW*(z(548)*z(3598)+z(552)*z(3599)+z(556)*z(3597)) +
0.1428571428571429*SW*(z(775)*z(3651)+z(779)*z(3653)+z(783)*z(3652)) +
0.1428571428571429*SW*(z(1209)*z(3741)+z(1216)*z(3743)+z(1223)*z(3742)) +
0.1428571428571429*SW*(z(1859)*z(3867)+z(1869)*z(3869)+z(1879)*z(3868)) +
0.1428571428571429*SW*(z(2500)*z(4020)+z(2513)*z(4022)+z(2526)*z(4021)) -
0.1428571428571429*SW*(z(34)*z(2661)*z(4191)+z(34)*z(2701)*z(4190)-z(2661)*z(4190)-z(2674)*z(4176)-z(2687)*z(4162)-z(2701)*z(4191)-
z(2714)*z(4193)-z(2727)*z(4192)-z(250)*z(2661)*z(4193)-z(250)*z(2714)*z(4190)-z(251)*z(2661)*z(4192)-z(251)*z(2727)*z(4190)-
z(283)*z(2687)*z(4191)-z(283)*z(2701)*z(4162)-z(284)*z(2674)*z(4191)-z(284)*z(2701)*z(4176)-z(285)*z(2687)*z(4193)-z(285)*z(2714)*z(4162)-
z(286)*z(2674)*z(4193)-z(286)*z(2714)*z(4176)-z(287)*z(2687)*z(4192)-z(287)*z(2727)*z(4162)-z(288)*z(2674)*z(4192)-z(288)*z(2727)*z(4176));
z(4397) = z(2941) - z(4373);
z(731) = z(113)*BOBA3 + z(117)*z(724);
z(727) = z(113)*z(725) - z(115)*z(724);
z(735) = -z(115)*BOBA3 - z(117)*z(725);
z(2929) = z(80)*z(731) + z(81)*z(727) - z(10)*z(735);
z(2344) = -z(205)*EOED3 - z(206)*z(2315);
z(2331) = z(206)*z(2314) - z(28)*EOED3;
z(2318) = -z(28)*z(2315) - z(205)*z(2314);
z(3027) = z(197)*z(2344) + z(201)*z(2331) + z(202)*z(2318);
z(2351) = -z(429)*EOED3 - z(439)*z(2315);
z(2362) = z(1778) + z(2351);
z(2338) = z(419)*EOED3 + z(439)*z(2314);
z(2372) = z(1794) + z(2338);
z(2325) = z(419)*z(2315) - z(429)*z(2314);
z(2382) = z(1810) + z(2325);
z(2938) = z(197)*z(2362) + z(201)*z(2372) + z(202)*z(2382);
z(4551) = z(3027)/z(2938);
z(2031) = -z(28)*z(2027) - z(205)*z(2026);
z(2044) = z(28)*z(2028) + z(206)*z(2026);
z(2057) = z(205)*z(2028) - z(206)*z(2027);
z(2103) = -z(28)*z(2099) - z(205)*z(2098);
z(2129) = z(205)*z(2100) - z(206)*z(2099);
z(2116) = z(28)*z(2100) + z(206)*z(2098);
z(2247) = -z(28)*z(2243) - z(205)*z(2242);
z(2273) = z(205)*z(2244) - z(206)*z(2243);
z(2260) = z(28)*z(2244) + z(206)*z(2242);
z(2188) = z(28)*z(2172) + z(206)*z(2170);
z(2175) = -z(28)*z(2171) - z(205)*z(2170);
z(2201) = z(205)*z(2172) - z(206)*z(2171);
z(1959) = -z(28)*z(1955) - z(205)*z(1954);
z(1985) = z(205)*z(1956) - z(206)*z(1955);
z(1972) = z(28)*z(1956) + z(206)*z(1954);
z(3030) = 4.7299E-
05*z(462)*(z(2031)+44.21104040254551*z(2044)+114.2537897207129*z(207)*z(2057)+114.2537897207129*z(208)*z(2044)+114.2537897207129*z(
209)*z(2031)-147.8697224042791*z(2057)-21.77701431319901*z(210)*z(2057)-21.77701431319901*z(211)*z(2044)-
21.77701431319901*z(212)*z(2031)-3.298991522019493*z(213)*z(2057)-3.298991522019493*z(214)*z(2044)-3.298991522019493*z(215)*z(2031))
+ 0.000263142*z(478)*(z(2103)+57.96163288262611*z(2129)-37.04060925279887*z(2116)-58.17953804409786*z(207)*z(2129)-
58.17953804409786*z(208)*z(2116)-58.17953804409786*z(209)*z(2103)-3.892632114979745*z(213)*z(2129)-3.892632114979745*z(214)*z(2116)-
3.892632114979745*z(215)*z(2103)-3.889998555912777*z(210)*z(2129)-3.889998555912777*z(211)*z(2116)-3.889998555912777*z(212)*z(2103))
+ 0.005250594000000001*z(510)*(1.3792740402324*z(2247)+1.576289654084852*z(2273)-1.220873295478569*z(2260)-
1.749405305380686*z(207)*z(2273)-1.749405305380686*z(208)*z(2260)-1.749405305380686*z(209)*z(2247)-1.677448684853561*z(213)*z(2273)-
1.677448684853561*z(214)*z(2260)-1.677448684853561*z(215)*z(2247)-z(210)*z(2273)-z(211)*z(2260)-z(212)*z(2247)) -
9.800000000000001*SW*(z(2790)*z(2903)+z(2791)*z(2820)+z(2792)*z(2918)) - 1.4*SW*(z(199)*z(1841)+z(203)*z(1828)+z(204)*z(1815)) -
1.4*SW*(z(284)*z(2492)+z(286)*z(2505)+z(288)*z(2518)) - 1.4*SW*(z(2667)+z(284)*z(2693)+z(286)*z(2706)+z(288)*z(2719)) -
0.00508017*z(494)*(z(2188)+1.575266182037215*z(2175)+1.26191682561804*z(210)*z(2201)+1.26191682561804*z(211)*z(2188)+1.261916825618
04*z(212)*z(2175)+1.738918973183969*z(207)*z(2201)+1.738918973183969*z(208)*z(2188)+1.738918973183969*z(209)*z(2175)-
1.564399813392072*z(2201)-1.445018572213135*z(213)*z(2201)-1.445018572213135*z(214)*z(2188)-1.445018572213135*z(215)*z(2175)) -
z(446)*(2.0E-
07*z(1959)+0.0210342*z(1985)+0.0009485670000000001*z(210)*z(1985)+0.0009485670000000001*z(211)*z(1972)+0.0009485670000000001*z(212
)*z(1959)-0.001343886*z(1972)-0.0196764*z(207)*z(1985)-0.0196764*z(208)*z(1972)-0.0196764*z(209)*z(1959)-0.00039917*z(213)*z(1985)-
0.00039917*z(214)*z(1972)-0.00039917*z(215)*z(1959));
z(3329) = z(205)*z(255) + z(206)*z(258) - z(28)*z(252);
z(3342) = z(205)*z(256) + z(206)*z(259) - z(28)*z(253);

z(3355) = z(205)*z(257) + z(206)*z(260) - z(28)*z(254);
z(3405) = z(205)*z(2733) + z(206)*z(2736) - z(28)*z(2730);
z(3418) = z(205)*z(2734) + z(206)*z(2737) - z(28)*z(2731);
z(3431) = z(205)*z(2735) + z(206)*z(2738) - z(28)*z(2732);
z(3490) = z(205)*z(2871) + z(206)*z(2874) - z(28)*z(2868);
z(3503) = z(205)*z(2872) + z(206)*z(2875) - z(28)*z(2869);
z(3516) = z(205)*z(2873) + z(206)*z(2876) - z(28)*z(2870);
z(4384) = z(205)*z(3325) + z(206)*z(3324) + z(3329)*z(3402) + z(3342)*z(3401) + z(3355)*z(3400) + z(3405)*z(3487) + z(3418)*z(3486) +
z(3431)*z(3485) + z(3490)*z(3569) + z(3503)*z(3568) + z(3516)*z(3567) + z(205)*z(3322) + z(206)*z(3323) + z(3329)*z(3397) + z(3342)*z(3398) +
z(3355)*z(3399) + z(3405)*z(3482) + z(3418)*z(3483) + z(3431)*z(3484) + z(3490)*z(3564) + z(3503)*z(3565) + z(3516)*z(3566) +
SW*(z(2820)*z(4372)+z(2903)*z(4370)+z(2918)*z(4371)) + 0.1428571428571429*SW*(z(1815)*z(3868)+z(1828)*z(3869)+z(1841)*z(3867)) +
0.1428571428571429*SW*(z(2492)*z(4020)+z(2505)*z(4022)+z(2518)*z(4021)) - z(28)*z(3326) - z(28)*z(3321) -
0.1428571428571429*SW*(z(34)*z(2654)*z(4191)+z(34)*z(2693)*z(4190)-z(2654)*z(4190)-z(2667)*z(4176)-z(2680)*z(4162)-z(2693)*z(4191)-
z(2706)*z(4193)-z(2719)*z(4192)-z(250)*z(2654)*z(4193)-z(250)*z(2706)*z(4190)-z(251)*z(2654)*z(4192)-z(251)*z(2719)*z(4190)-
z(283)*z(2680)*z(4191)-z(283)*z(2693)*z(4162)-z(284)*z(2667)*z(4191)-z(284)*z(2693)*z(4176)-z(285)*z(2680)*z(4193)-z(285)*z(2706)*z(4162)-
z(286)*z(2667)*z(4193)-z(286)*z(2706)*z(4176)-z(287)*z(2680)*z(4192)-z(287)*z(2719)*z(4162)-z(288)*z(2667)*z(4192)-z(288)*z(2719)*z(4176));
z(4408) = z(3030) - z(4384);
z(4563) = z(4551)*z(4397) - z(4408);
z(3028) = z(199)*z(2344) + z(203)*z(2331) + z(204)*z(2318);
z(2939) = z(199)*z(2362) + z(203)*z(2372) + z(204)*z(2382);
z(4552) = z(3028) - z(2939)*z(4551);
z(2352) = -z(430)*EOED3 - z(440)*z(2315);
z(2363) = z(1779) + z(2352);
z(2339) = z(420)*EOED3 + z(440)*z(2314);
z(2373) = z(1795) + z(2339);
z(2326) = z(420)*z(2315) - z(430)*z(2314);
z(2383) = z(1811) + z(2326);
z(2952) = z(199)*z(2363) + z(203)*z(2373) + z(204)*z(2383);
z(2951) = z(197)*z(2363) + z(201)*z(2373) + z(202)*z(2383);
z(4421) = z(2951)/z(2938);
z(4422) = z(2952) - z(2939)*z(4421);
z(4672) = z(4552)/z(4422);
z(2039) = z(420)*z(2027) - z(430)*z(2026);
z(2096) = z(1811) + z(2039);
z(2052) = z(440)*z(2026) - z(420)*z(2028);
z(2086) = z(1795) + z(2052);
z(2065) = z(430)*z(2028) - z(440)*z(2027);
z(2076) = z(1779) + z(2065);
z(837) = z(390)*z(822) - z(382)*z(824);
z(852) = z(747) + z(837);
z(844) = z(386)*z(824) - z(390)*z(823);
z(848) = z(743) + z(844);
z(830) = z(382)*z(823) - z(386)*z(822);
z(856) = z(751) + z(830);
z(902) = z(382)*z(895) - z(386)*z(894);
z(928) = z(751) + z(902);
z(916) = z(386)*z(896) - z(390)*z(895);
z(920) = z(743) + z(916);
z(909) = z(390)*z(894) - z(382)*z(896);
z(924) = z(747) + z(909);
z(585) = z(114)*z(581) - z(116)*z(580);
z(589) = z(118)*z(580) - z(114)*z(582);
z(593) = z(116)*z(582) - z(118)*z(581);
z(2111) = z(420)*z(2099) - z(430)*z(2098);
z(2168) = z(1811) + z(2111);
z(2137) = z(430)*z(2100) - z(440)*z(2099);
z(2148) = z(1779) + z(2137);
z(2124) = z(440)*z(2098) - z(420)*z(2100);
z(2158) = z(1795) + z(2124);
z(603) = z(114)*z(599) - z(116)*z(598);
z(607) = z(118)*z(598) - z(114)*z(600);
z(611) = z(116)*z(600) - z(118)*z(599);
z(1398) = z(397)*z(1389) - z(404)*z(1388);
z(1440) = z(1173) + z(1398);
z(1419) = z(404)*z(1390) - z(411)*z(1389);
z(1426) = z(1153) + z(1419);
z(1408) = z(411)*z(1388) - z(397)*z(1390);
z(1433) = z(1163) + z(1408);
z(873) = z(390)*z(858) - z(382)*z(860);
z(888) = z(747) + z(873);
z(880) = z(386)*z(860) - z(390)*z(859);
z(884) = z(743) + z(880);
z(866) = z(382)*z(859) - z(386)*z(858);
z(892) = z(751) + z(866);
z(719) = z(116)*z(708) - z(118)*z(707);
z(711) = z(114)*z(707) - z(116)*z(706);
z(715) = z(118)*z(706) - z(114)*z(708);
z(1668) = z(397)*z(1659) - z(404)*z(1658);

z(1710) = z(1173) + z(1668);
z(1689) = z(404)*z(1660) - z(411)*z(1659);
z(1696) = z(1153) + z(1689);
z(1678) = z(411)*z(1658) - z(397)*z(1660);
z(1703) = z(1163) + z(1678);
z(1082) = z(382)*z(1075) - z(386)*z(1074);
z(1108) = z(751) + z(1082);
z(1096) = z(386)*z(1076) - z(390)*z(1075);
z(1100) = z(743) + z(1096);
z(1089) = z(390)*z(1074) - z(382)*z(1076);
z(1104) = z(747) + z(1089);
z(2255) = z(420)*z(2243) - z(430)*z(2242);
z(2312) = z(1811) + z(2255);
z(2281) = z(430)*z(2244) - z(440)*z(2243);
z(2292) = z(1779) + z(2281);
z(2268) = z(440)*z(2242) - z(420)*z(2244);
z(2302) = z(1795) + z(2268);
z(679) = z(118)*z(670) - z(114)*z(672);
z(675) = z(114)*z(671) - z(116)*z(670);
z(683) = z(116)*z(672) - z(118)*z(671);
z(1290) = z(397)*z(1281) - z(404)*z(1280);
z(1332) = z(1173) + z(1290);
z(1311) = z(404)*z(1282) - z(411)*z(1281);
z(1318) = z(1153) + z(1311);
z(1300) = z(411)*z(1280) - z(397)*z(1282);
z(1325) = z(1163) + z(1300);
z(2196) = z(440)*z(2170) - z(420)*z(2172);
z(2230) = z(1795) + z(2196);
z(2183) = z(420)*z(2171) - z(430)*z(2170);
z(2240) = z(1811) + z(2183);
z(2209) = z(430)*z(2172) - z(440)*z(2171);
z(2220) = z(1779) + z(2209);
z(1017) = z(390)*z(1002) - z(382)*z(1004);
z(1032) = z(747) + z(1017);
z(1010) = z(382)*z(1003) - z(386)*z(1002);
z(1036) = z(751) + z(1010);
z(1024) = z(386)*z(1004) - z(390)*z(1003);
z(1028) = z(743) + z(1024);
z(1570) = z(411)*z(1550) - z(397)*z(1552);
z(1595) = z(1163) + z(1570);
z(1560) = z(397)*z(1551) - z(404)*z(1550);
z(1602) = z(1173) + z(1560);
z(1581) = z(404)*z(1552) - z(411)*z(1551);
z(1588) = z(1153) + z(1581);
z(1967) = z(420)*z(1955) - z(430)*z(1954);
z(2024) = z(1811) + z(1967);
z(1993) = z(430)*z(1956) - z(440)*z(1955);
z(2004) = z(1779) + z(1993);
z(1980) = z(440)*z(1954) - z(420)*z(1956);
z(2014) = z(1795) + z(1980);
z(625) = z(118)*z(616) - z(114)*z(618);
z(629) = z(116)*z(618) - z(118)*z(617);
z(621) = z(114)*z(617) - z(116)*z(616);
z(1365) = z(404)*z(1336) - z(411)*z(1335);
z(1372) = z(1153) + z(1365);
z(1354) = z(411)*z(1334) - z(397)*z(1336);
z(1379) = z(1163) + z(1354);
z(1344) = z(397)*z(1335) - z(404)*z(1334);
z(1386) = z(1173) + z(1344);
z(2954) = 4.7299E-
05*z(462)*(z(2096)+44.21104040254551*z(2086)+114.2537897207129*z(207)*z(2076)+114.2537897207129*z(208)*z(2086)+114.2537897207129*z(
209)*z(2096)-147.8697224042791*z(2076)-21.77701431319901*z(210)*z(2076)-21.77701431319901*z(211)*z(2086)-
21.77701431319901*z(212)*z(2096)-3.298991522019493*z(213)*z(2076)-3.298991522019493*z(214)*z(2086)-3.298991522019493*z(215)*z(2096))
+ 4.9681E-05*z(454)*(z(852)+436.869225659709*z(129)*z(848)+436.869225659709*z(130)*z(852)+436.869225659709*z(131)*z(856)-
410.7868199110324*z(848)-25.99182786175802*z(856)-23.60302731426501*z(132)*z(848)-23.60302731426501*z(133)*z(852)-
23.60302731426501*z(134)*z(856)-4.101447233348765*z(135)*z(848)-4.101447233348765*z(136)*z(852)-4.101447233348765*z(137)*z(856)) +
5.641700000000001E-
05*z(486)*(z(928)+247.6572664267862*z(920)+24.25456865838311*z(132)*z(920)+24.25456865838311*z(133)*z(924)+24.25456865838311*z(134)
*z(928)-149.1704628037648*z(924)-269.1738305829803*z(129)*z(920)-269.1738305829803*z(130)*z(924)-269.1738305829803*z(131)*z(928)-
14.71584805998192*z(135)*z(920)-14.71584805998192*z(136)*z(924)-14.71584805998192*z(137)*z(928)) +
0.000203764*z(458)*(z(585)+5.754804577844959*z(589)+129.8428574232936*z(82)*z(593)+129.8428574232936*z(83)*z(589)+129.8428574232936
*z(84)*z(585)-106.5158713020946*z(593)-9.530554955733102*z(85)*z(593)-9.530554955733102*z(86)*z(589)-9.530554955733102*z(87)*z(585)-
4.900914783769459*z(88)*z(593)-4.900914783769459*z(89)*z(589)-4.900914783769459*z(90)*z(585)) +
0.000263142*z(478)*(z(2168)+57.9616328826611*z(2148)-37.04060925279887*z(2158)-58.17953804409786*z(207)*z(2148)-
58.17953804409786*z(208)*z(2158)-58.17953804409786*z(209)*z(2168)-3.892632114979745*z(213)*z(2148)-3.892632114979745*z(214)*z(2158)-
3.892632114979745*z(215)*z(2168)-3.889998555912777*z(210)*z(2148)-3.889998555912777*z(211)*z(2158)-3.889998555912777*z(212)*z(2168))
+
0.000439106*z(474)*(z(603)+6.399042600192209*z(607)+29.7948559117844*z(82)*z(611)+29.7948559117844*z(83)*z(607)+29.7948559117844*z(

84)*z(603)-15.07109900570705*z(611)-5.299196549352549*z(85)*z(611)-5.299196549352549*z(86)*z(607)-5.299196549352549*z(87)*z(603)-
2.389958233319518*z(88)*z(611)-2.389958233319518*z(89)*z(607)-2.389958233319518*z(90)*z(603)) + 6.128E-
05*z(482)*(19.44327676240209*z(1440)+238.8415469973891*z(1426)-120.5957898172324*z(1433)-232.8573759791123*z(166)*z(1426)-
232.8573759791123*z(172)*z(1433)-232.8573759791123*z(173)*z(1440)-1.221507832898173*z(168)*z(1426)-1.221507832898173*z(174)*z(1433)-
1.221507832898173*z(175)*z(1440)-z(170)*z(1426)-z(176)*z(1433)-z(177)*z(1440)) +
0.000439106*z(470)*(1.522912463049924*z(888)+15.07109900570705*z(129)*z(884)+15.07109900570705*z(130)*z(888)+15.07109900570705*z(13
1)*z(892)-12.16291738213552*z(884)-2.050165563667998*z(892)-6.399042600192209*z(132)*z(884)-6.399042600192209*z(133)*z(888)-
6.399042600192209*z(134)*z(892)-z(135)*z(884)-z(136)*z(888)-z(137)*z(892)) +
0.003268181*z(522)*(2.228608513420769*z(719)+2.730118068736096*z(711)-1.290837318985699*z(715)-2.575607348552605*z(85)*z(719)-
2.575607348552605*z(86)*z(715)-2.575607348552605*z(87)*z(711)-1.999895354633051*z(88)*z(719)-1.999895354633051*z(89)*z(715)-
1.999895354633051*z(90)*z(711)-z(82)*z(719)-z(83)*z(715)-z(84)*z(711)) +
0.00334198*z(514)*(2.523796072986673*z(1710)+2.541448183412228*z(1696)-z(1703)-2.199101730112089*z(166)*z(1696)-
2.199101730112089*z(172)*z(1703)-2.199101730112089*z(173)*z(1710)-2.178312856450368*z(170)*z(1696)-2.178312856450368*z(176)*z(1703)-
2.178312856450368*z(177)*z(1710)-1.513417495017923*z(168)*z(1696)-1.513417495017923*z(174)*z(1703)-1.513417495017923*z(175)*z(1710))
+ 0.00353959*z(518)*(2.051998112775774*z(1108)+2.135415683737382*z(1100)-z(1104)-2.270871202596911*z(132)*z(1100)-
2.270871202596911*z(133)*z(1104)-2.270871202596911*z(134)*z(1108)-2.253045691732658*z(135)*z(1100)-2.253045691732658*z(136)*z(1104)-
2.253045691732658*z(137)*z(1108)-1.758446882266025*z(129)*z(1100)-1.758446882266025*z(130)*z(1104)-1.758446882266025*z(131)*z(1108))
+ 0.005250594000000001*z(510)*(1.379274040232 4*z(2312)+1.576289654084852*z(2292)-1.220873295478569*z(2302)-
1.749405305380686*z(207)*z(2292)-1.749405305380686*z(208)*z(2302)-1.749405305380686*z(209)*z(2312)-1.677448684853561*z(213)*z(2292)-
1.677448684853561*z(214)*z(2302)-1.677448684853561*z(215)*z(2312)-z(210)*z(2292)-z(211)*z(2302)-z(212)*z(2312)) -
9.800000000000001*SW*(z(2790)*z(2913)+z(2791)*z(2828)+z(2792)*z(2926)) - 1.4*SW*(z(114)*z(557)+z(116)*z(553)+z(118)*z(549)) -
1.4*SW*(z(121)*z(776)+z(125)*z(780)+z(126)*z(784)) - 1.4*SW*(z(199)*z(1860)+z(203)*z(1870)+z(204)*z(1880)) -
1.4*SW*(z(284)*z(2501)+z(286)*z(2514)+z(288)*z(2527)) - 1.4*SW*(z(20)*z(1210)+z(194)*z(1217)-z(196)*z(1224)) -
1.4*SW*(z(2675)+z(284)*z(2702)+z(286)*z(2715)+z(288)*z(2728)) -
0.001989028*z(506)*(2.026969957185118*z(679)+3.454209794935014*z(675)+z(82)*z(683)+z(83)*z(679)+z(84)*z(675)+4.239214832571488*z(85)
*z(683)+4.239214832571488*z(86)*z(679)+4.239214832571488*z(87)*z(675)-3.484725705218831*z(683)-3.642764204425479*z(88)*z(683)-
3.642764204425479*z(89)*z(679)-3.642764204425479*z(90)*z(675)) - 4.9681E-
05*z(450)*(8.034661138060828*z(1332)+396.0548298142147*z(1318)+z(168)*z(1318)+z(174)*z(1325)+z(175)*z(1332)-
19.09315432459089*z(1325)-410.7868199110324*z(166)*z(1318)-410.7868199110324*z(172)*z(1325)-410.7868199110324*z(173)*z(1332)-
25.99182786175802*z(170)*z(1318)-25.99182786175802*z(176)*z(1325)-25.99182786175802*z(177)*z(1332)) -
0.00508017*z(494)*(z(2230)+1.575266182037215*z(2240)+1.26191682561804*z(210)*z(2220)+1.26191682561804*z(211)*z(2230)+1.261916825618
04*z(212)*z(2240)+1.738918973183969*z(207)*z(2220)+1.738918973183969*z(208)*z(2230)+1.738918973183969*z(209)*z(2240)-
1.564399813392072*z(2220)-1.445018572213135*z(213)*z(2220)-1.445018572213135*z(214)*z(2230)-1.445018572213135*z(215)*z(2240)) -
0.00404624*z(502)*(z(1032)+2.01888419866345*z(1036)+1.135663974455297*z(132)*z(1028)+1.135663974455297*z(133)*z(1032)+1.13566397445
5297*z(134)*z(1036)+1.613396387757523*z(129)*z(1028)+1.613396387757523*z(130)*z(1032)+1.613396387757523*z(131)*z(1036)-
1.85913242911938*z(1028)-1.514074795365574*z(135)*z(1028)-1.514074795365574*z(136)*z(1032)-1.514074795365574*z(137)*z(1036)) -
0.00297212*z(498)*(z(1595)+2.289867165524945*z(1602)+1.384676594484745*z(168)*z(1588)+1.384676594484745*z(174)*z(1595)+1.3846765944
84745*z(175)*z(1602)+2.471453709809833*z(166)*z(1588)+2.471453709809833*z(172)*z(1595)+2.471453709809833*z(173)*z(1602)-
2.986921120277781*z(1588)-2.589087923771584*z(170)*z(1588)-2.589087923771584*z(176)*z(1595)-2.589087923771584*z(177)*z(1602)) -
z(446)*(2.0E-
07*z(2024)+0.0210342*z(2004)+0.000948567000000000001*z(210)*z(2004)+0.000948567000000000001*z(211)*z(2014)+0.000948567000000000001*z(212
)*z(2024)-0.001343886*z(2014)-0.0196764*z(207)*z(2004)-0.0196764*z(208)*z(2014)-0.0196764*z(209)*z(2024)-0.00039917*z(213)*z(2004)-
0.00039917*z(214)*z(2014)-0.00039917*z(215)*z(2024)) -
0.00021415*z(490)*(40.8116273639972*z(625)+64.04720989960309*z(82)*z(629)+64.04720989960309*z(83)*z(625)+64.04720989960309*z(84)*z(
621)-70.09441979920617*z(629)-4.618921316833995*z(621)-10.35433107634836*z(85)*z(629)-10.35433107634836*z(86)*z(625)-
10.35433107634836*z(87)*z(621)-z(88)*z(629)-z(89)*z(625)-z(90)*z(621)) -
0.000156039*z(466)*(34.63294432802056*z(1372)+4.285595267849704*z(168)*z(1372)+4.285595267849704*z(174)*z(1379)+4.285595267849704*
z(175)*z(1386)-6.601112542377225*z(1379)-z(1386)-34.22740468728971*z(166)*z(1372)-34.22740468728971*z(172)*z(1379)-
34.22740468728971*z(173)*z(1386)-5.76932689904447*z(170)*z(1372)-5.76932689904447*z(176)*z(1379)-5.76932689904447*z(177)*z(1386));
z(3337) = z(252)*z(420) + z(255)*z(430) + z(258)*z(440);
z(3350) = z(253)*z(420) + z(256)*z(430) + z(259)*z(440);
z(3363) = z(254)*z(420) + z(257)*z(430) + z(260)*z(440);
z(3413) = z(420)*z(2730) + z(430)*z(2733) + z(440)*z(2736);
z(3426) = z(420)*z(2731) + z(430)*z(2734) + z(440)*z(2737);
z(3439) = z(420)*z(2732) + z(430)*z(2735) + z(440)*z(2738);
z(3498) = z(420)*z(2868) + z(430)*z(2871) + z(440)*z(2874);
z(3511) = z(420)*z(2869) + z(430)*z(2872) + z(440)*z(2875);
z(3524) = z(420)*z(2870) + z(430)*z(2873) + z(440)*z(2876);
z(4374) = z(114)*z(3149) + z(116)*z(3148) + z(118)*z(3147) + z(382)*z(3204) + z(386)*z(3203) + z(390)*z(3202) + z(397)*z(3255) + z(404)*z(3254)
+ z(411)*z(3253) + z(420)*z(3326) + z(430)*z(3325) + z(440)*z(3324) + z(3337)*z(3402) + z(3350)*z(3401) + z(3363)*z(3400) + z(3413)*z(3487) +
z(3426)*z(3486) + z(3439)*z(3485) + z(3498)*z(3569) + z(3511)*z(3568) + z(3524)*z(3567) + z(114)*z(3144) + z(116)*z(3145) + z(118)*z(3146) +
z(382)*z(3199) + z(386)*z(3200) + z(390)*z(3201) + z(397)*z(3250) + z(404)*z(3251) + z(411)*z(3252) + z(420)*z(3321) + z(430)*z(3322) +
z(440)*z(3323) + z(3337)*z(3397) + z(3350)*z(3398) + z(3363)*z(3399) + z(3413)*z(3482) + z(3426)*z(3483) + z(3439)*z(3484) + z(3498)*z(3564) +
z(3511)*z(3565) + z(3524)*z(3566) + SW*(z(2828)*z(4372)+z(2913)*z(4370)+z(2926)*z(4371)) +
0.142857142857142 9*SW*(z(549)*z(3598)+z(553)*z(3599)+z(557)*z(3597)) +
0.142857142857142 9*SW*(z(776)*z(3651)+z(780)*z(3653)+z(784)*z(3652)) +
0.142857142857142 9*SW*(z(1210)*z(3741)+z(1217)*z(3743)+z(1224)*z(3742)) +
0.142857142857142 9*SW*(z(1860)*z(3867)+z(1870)*z(3869)+z(1880)*z(3868)) +
0.142857142857142 9*SW*(z(2501)*z(4020)+z(2514)*z(4022)+z(2527)*z(4021)) -
0.142857142857142 9*SW*(z(34)*z(2662)*z(4191)+z(34)*z(2702)*z(4190)-z(2662)*z(4190)-z(2675)*z(4176)-z(2688)*z(4162)-z(2702)*z(4191)-
z(2715)*z(4193)-z(2728)*z(4192)-z(250)*z(2662)*z(4193)-z(250)*z(2715)*z(4190)-z(251)*z(2662)*z(4192)-z(251)*z(2728)*z(4190)-
z(283)*z(2688)*z(4191)-z(283)*z(2702)*z(4162)-z(284)*z(2675)*z(4191)-z(284)*z(2702)*z(4176)-z(285)*z(2688)*z(4193)-z(285)*z(2715)*z(4162)-
z(286)*z(2675)*z(4193)-z(286)*z(2715)*z(4176)-z(287)*z(2688)*z(4192)-z(287)*z(2728)*z(4162)-z(288)*z(2675)*z(4192)-z(288)*z(2728)*z(4176));
z(4398) = z(2954) - z(4374);
z(4433) = z(4421)*z(4397) - z(4398);
z(4683) = z(4563) - z(4672)*z(4433);
z(3029) = z(205)*z(2331) + z(206)*z(2318) - z(28)*z(2344);
z(2940) = z(205)*z(2372) + z(206)*z(2382) - z(28)*z(2362);

```
z(4553) = z(3029) - z(2940)*z(4551);
z(2953) = z(205)*z(2373) + z(206)*z(2383) - z(28)*z(2363);
z(4423) = z(2953) - z(2940)*z(4421);
z(4673) = z(4553) - z(4423)*z(4672);
z(2337) = z(418)*EOED3 + z(438)*z(2314);
z(2371) = z(1793) + z(2337);
z(2324) = z(418)*z(2315) - z(428)*z(2314);
z(2381) = z(1809) + z(2324);
z(2350) = -z(428)*EOED3 - z(438)*z(2315);
z(2361) = z(1777) + z(2350);
z(2966) = z(205)*z(2371) + z(206)*z(2381) - z(28)*z(2361);
z(2964) = z(197)*z(2361) + z(201)*z(2371) + z(202)*z(2381);
z(4434) = z(2964)/z(2938);
z(4436) = z(2966) - z(2940)*z(4434);
z(2965) = z(199)*z(2361) + z(203)*z(2371) + z(204)*z(2381);
z(4435) = z(2965) - z(2939)*z(4434);
z(4564) = z(4435)/z(4422);
z(4565) = z(4436) - z(4423)*z(4564);
z(4772) = z(4673)/z(4565);
z(2037) = z(418)*z(2027) - z(428)*z(2026);
z(2094) = z(1809) + z(2037);
z(2050) = z(438)*z(2026) - z(418)*z(2028);
z(2084) = z(1793) + z(2050);
z(2063) = z(428)*z(2028) - z(438)*z(2027);
z(2074) = z(1777) + z(2063);
z(835) = z(388)*z(822) - z(380)*z(824);
z(850) = z(745) + z(835);
z(842) = z(384)*z(824) - z(388)*z(823);
z(846) = z(741) + z(842);
z(828) = z(380)*z(823) - z(384)*z(822);
z(854) = z(749) + z(828);
z(900) = z(380)*z(895) - z(384)*z(894);
z(926) = z(749) + z(900);
z(914) = z(384)*z(896) - z(388)*z(895);
z(918) = z(741) + z(914);
z(907) = z(388)*z(894) - z(380)*z(896);
z(922) = z(745) + z(907);
z(583) = -z(10)*z(581) - z(80)*z(580);
z(587) = z(10)*z(582) + z(81)*z(580);
z(591) = z(80)*z(582) - z(81)*z(581);
z(2109) = z(418)*z(2099) - z(428)*z(2098);
z(2166) = z(1809) + z(2109);
z(2135) = z(428)*z(2100) - z(438)*z(2099);
z(2146) = z(1777) + z(2135);
z(2122) = z(438)*z(2098) - z(418)*z(2100);
z(2156) = z(1793) + z(2122);
z(601) = -z(10)*z(599) - z(80)*z(598);
z(605) = z(10)*z(600) + z(81)*z(598);
z(609) = z(80)*z(600) - z(81)*z(599);
z(1396) = z(395)*z(1389) - z(402)*z(1388);
z(1438) = z(1171) + z(1396);
z(1417) = z(402)*z(1390) - z(409)*z(1389);
z(1424) = z(1151) + z(1417);
z(1406) = z(409)*z(1388) - z(395)*z(1390);
z(1431) = z(1161) + z(1406);
z(871) = z(388)*z(858) - z(380)*z(860);
z(886) = z(745) + z(871);
z(878) = z(384)*z(860) - z(388)*z(859);
z(882) = z(741) + z(878);
z(864) = z(380)*z(859) - z(384)*z(858);
z(890) = z(749) + z(864);
z(717) = z(80)*z(708) - z(81)*z(707);
z(709) = -z(10)*z(707) - z(80)*z(706);
z(713) = z(10)*z(708) + z(81)*z(706);
z(1666) = z(395)*z(1659) - z(402)*z(1658);
z(1708) = z(1171) + z(1666);
z(1687) = z(402)*z(1660) - z(409)*z(1659);
z(1694) = z(1151) + z(1687);
z(1676) = z(409)*z(1658) - z(395)*z(1660);
z(1701) = z(1161) + z(1676);
z(1080) = z(380)*z(1075) - z(384)*z(1074);
z(1106) = z(749) + z(1080);
z(1094) = z(384)*z(1076) - z(388)*z(1075);
z(1098) = z(741) + z(1094);
z(1087) = z(388)*z(1074) - z(380)*z(1076);
z(1102) = z(745) + z(1087);
z(2253) = z(418)*z(2243) - z(428)*z(2242);
z(2310) = z(1809) + z(2253);
```

z(2279) = z(428)*z(2244) - z(438)*z(2243);
z(2290) = z(1777) + z(2279);
z(2266) = z(438)*z(2242) - z(418)*z(2244);
z(2300) = z(1793) + z(2266);
z(677) = z(10)*z(672) + z(81)*z(670);
z(673) = -z(10)*z(671) - z(80)*z(670);
z(681) = z(80)*z(672) - z(81)*z(671);
z(1288) = z(395)*z(1281) - z(402)*z(1280);
z(1330) = z(1171) + z(1288);
z(1309) = z(402)*z(1282) - z(409)*z(1281);
z(1316) = z(1151) + z(1309);
z(1298) = z(409)*z(1280) - z(395)*z(1282);
z(1323) = z(1161) + z(1298);
z(2194) = z(438)*z(2170) - z(418)*z(2172);
z(2228) = z(1793) + z(2194);
z(2181) = z(418)*z(2171) - z(428)*z(2170);
z(2238) = z(1809) + z(2181);
z(2207) = z(428)*z(2172) - z(438)*z(2171);
z(2218) = z(1777) + z(2207);
z(1015) = z(388)*z(1002) - z(380)*z(1004);
z(1030) = z(745) + z(1015);
z(1008) = z(380)*z(1003) - z(384)*z(1002);
z(1034) = z(749) + z(1008);
z(1022) = z(384)*z(1004) - z(388)*z(1003);
z(1026) = z(741) + z(1022);
z(1568) = z(409)*z(1550) - z(395)*z(1552);
z(1593) = z(1161) + z(1568);
z(1558) = z(395)*z(1551) - z(402)*z(1550);
z(1600) = z(1171) + z(1558);
z(1579) = z(402)*z(1552) - z(409)*z(1551);
z(1586) = z(1151) + z(1579);
z(1965) = z(418)*z(1955) - z(428)*z(1954);
z(2022) = z(1809) + z(1965);
z(1991) = z(428)*z(1956) - z(438)*z(1955);
z(2002) = z(1777) + z(1991);
z(1978) = z(438)*z(1954) - z(418)*z(1956);
z(2012) = z(1793) + z(1978);
z(623) = z(10)*z(618) + z(81)*z(616);
z(627) = z(80)*z(618) - z(81)*z(617);
z(619) = -z(10)*z(617) - z(80)*z(616);
z(1363) = z(402)*z(1336) - z(409)*z(1335);
z(1370) = z(1151) + z(1363);
z(1352) = z(409)*z(1334) - z(395)*z(1336);
z(1377) = z(1161) + z(1352);
z(1342) = z(395)*z(1335) - z(402)*z(1334);
z(1384) = z(1171) + z(1342);
z(2967) = 4.7299E-
05*z(462)*(z(2094)+44.21104040254551*z(2084)+114.2537897207129*z(207)*z(2074)+114.2537897207129*z(208)*z(2084)+114.2537897207129*z(
209)*z(2094)-147.8697224042791*z(2074)-21.77701431319901*z(210)*z(2074)-21.77701431319901*z(211)*z(2084)-
21.77701431319901*z(212)*z(2094)-3.298991522019493*z(213)*z(2074)-3.298991522019493*z(214)*z(2084)-3.298991522019493*z(215)*z(2094))
+ 4.9681E-05*z(454)*(z(850)+436.869225659709*z(129)*z(846)+436.869225659709*z(130)*z(850)+436.869225659709*z(131)*z(854)-
410.7868199110324*z(846)-25.99182786175802*z(854)-23.60302731426501*z(132)*z(846)-23.60302731426501*z(133)*z(850)-
23.60302731426501*z(134)*z(854)-4.101447233348765*z(135)*z(846)-4.101447233348765*z(136)*z(850)-4.101447233348765*z(137)*z(854)) +
5.641700000000001E-
05*z(486)*(z(926)+247.6572664267862*z(918)+24.25456865838311*z(132)*z(918)+24.25456865838311*z(133)*z(922)+24.25456865838311*z(134)
*z(926)-149.1704628037648*z(922)-269.1738305829803*z(129)*z(918)-269.1738305829803*z(130)*z(922)-269.1738305829803*z(131)*z(926)-
14.71584805998192*z(135)*z(918)-14.71584805998192*z(136)*z(922)-14.71584805998192*z(137)*z(926)) +
0.000203764*z(458)*(z(583)+5.754804577844959*z(587)+129.8428574232936*z(82)*z(591)+129.8428574232936*z(83)*z(587)+129.8428574232936
*z(84)*z(583)-106.5158713020946*z(591)-9.530554955733102*z(85)*z(591)-9.530554955733102*z(86)*z(587)-9.530554955733102*z(87)*z(583)-
4.900914783769459*z(88)*z(591)-4.900914783769459*z(89)*z(587)-4.900914783769459*z(90)*z(583)) +
0.000263142*z(478)*(z(2166)+57.96163288262611*z(2146)-37.04060925279887*z(2156)-58.17953804409786*z(207)*z(2146)-
58.17953804409786*z(208)*z(2156)-58.17953804409786*z(209)*z(2166)-3.892632114979745*z(213)*z(2146)-3.892632114979745*z(214)*z(2156)-
3.892632114979745*z(215)*z(2166)-3.889998555912777*z(210)*z(2146)-3.889998555912777*z(211)*z(2156)-3.889998555912777*z(212)*z(2166))
+
0.000439106*z(474)*(z(601)+6.399042600192209*z(605)+29.7948559117844*z(82)*z(609)+29.7948559117844*z(83)*z(605)+29.7948559117844*z(
84)*z(601)-15.07109900570705*z(609)-5.299196549352549*z(85)*z(609)-5.299196549352549*z(86)*z(605)-5.299196549352549*z(87)*z(601)-
2.389958233319518*z(88)*z(609)-2.389958233319518*z(89)*z(605)-2.389958233319518*z(90)*z(601)) + 6.128E-
05*z(482)*(19.44327676240209*z(1438)+238.8415469973891*z(1424)-120.5957898172324*z(1431)-232.8573759791123*z(166)*z(1424)-
232.8573759791123*z(172)*z(1431)-232.8573759791123*z(173)*z(1438)-1.221507832898173*z(168)*z(1424)-1.221507832898173*z(174)*z(1431)-
1.221507832898173*z(175)*z(1438)-z(170)*z(1424)-z(176)*z(1431)-z(177)*z(1438)) +
0.000439106*z(470)*(1.522912463049924*z(886)+15.07109900570705*z(129)*z(882)+15.07109900570705*z(130)*z(886)+15.07109900570705*z(13
1)*z(890)-12.16291738213552*z(882)-2.050165563667998*z(890)-6.399042600192209*z(132)*z(882)-6.399042600192209*z(133)*z(886)-
6.399042600192209*z(134)*z(890)-z(135)*z(882)-z(136)*z(886)-z(137)*z(890)) +
0.003268181*z(522)*(2.228608513420769*z(717)+2.730118068736096*z(709)-1.290837318985699*z(713)-2.575607348552605*z(85)*z(717)-
2.575607348552605*z(86)*z(713)-2.575607348552605*z(87)*z(709)-1.999895354633051*z(88)*z(717)-1.999895354633051*z(89)*z(713)-
1.999895354633051*z(90)*z(709)-z(82)*z(717)-z(83)*z(713)-z(84)*z(709)) +
0.00334198*z(514)*(2.523796072986673*z(1708)+2.541448183412228*z(1694)-z(1701)-2.199101730112089*z(166)*z(1694)-
2.199101730112089*z(172)*z(1701)-2.199101730112089*z(173)*z(1708)-2.178312856450368*z(170)*z(1694)-2.178312856450368*z(176)*z(1701)-

2.178312856450368*z(177)*z(1708)-1.513417495017923*z(168)*z(1694)-1.513417495017923*z(174)*z(1701)-1.513417495017923*z(175)*z(1708))
+ 0.00353959*z(518)*(2.051998112775774*z(1106)+2.135415683737382*z(1098)-z(1102)-2.270871202596911*z(132)*z(1098)-
2.270871202596911*z(133)*z(1102)-2.270871202596911*z(134)*z(1106)-2.253045691732658*z(135)*z(1098)-2.253045691732658*z(136)*z(1102)-
2.253045691732658*z(137)*z(1106)-1.758446882266025*z(129)*z(1098)-1.758446882266025*z(130)*z(1102)-1.758446882266025*z(131)*z(1106))
+ 0.005250594000000001*z(510)*(1.3792740402324*z(2310)+1.576289654084852*z(2290)-1.220873295478569*z(2300)-
1.749405305380686*z(207)*z(2290)-1.749405305380686*z(208)*z(2300)-1.749405305380686*z(209)*z(2310)-1.677448684853561*z(213)*z(2290)-
1.677448684853561*z(214)*z(2300)-1.677448684853561*z(215)*z(2310)-z(210)*z(2290)-z(211)*z(2300)-z(212)*z(2310)) -
9.800000000000001*SW*(z(2790)*z(2911)+z(2791)*z(2826)+z(2792)*z(2924)) - 1.4*SW*(z(114)*z(555)+z(116)*z(551)+z(118)*z(547)) -
1.4*SW*(z(121)*z(774)+z(125)*z(778)+z(126)*z(782)) - 1.4*SW*(z(199)*z(1858)+z(203)*z(1868)+z(204)*z(1878)) -
1.4*SW*(z(284)*z(2499)+z(286)*z(2512)+z(288)*z(2525)) - 1.4*SW*(z(20)*z(1208)+z(194)*z(1215)-z(196)*z(1222)) -
1.4*SW*(z(2673)+z(284)*z(2700)+z(286)*z(2713)+z(288)*z(2726)) -
0.001989028*z(506)*(2.026969957185118*z(677)+3.454209794935014*z(673)+z(82)*z(681)+z(83)*z(677)+z(84)*z(673)+4.239214832571488*z(85)
*z(681)+4.239214832571488*z(86)*z(677)+4.239214832571488*z(87)*z(673)-3.484725705218831*z(681)-3.642764204425479*z(88)*z(681)-
3.642764204425479*z(89)*z(677)-3.642764204425479*z(90)*z(673)) - 4.9681E-
05*z(450)*(8.034661138060828*z(1330)+396.0548298142147*z(1316)+z(168)*z(1316)+z(174)*z(1323)+z(175)*z(1330)-
19.09315432459089*z(1323)-410.7868199110324*z(166)*z(1316)-410.7868199110324*z(172)*z(1323)-410.7868199110324*z(173)*z(1330)-
25.99182786175802*z(170)*z(1316)-25.99182786175802*z(176)*z(1323)-25.99182786175802*z(177)*z(1330)) -
0.00508017*z(494)*(z(2228)+1.575266182037215*z(2238)+1.26191682561804*z(210)*z(2218)+1.26191682561804*z(211)*z(2228)+1.261916825618
04*z(212)*z(2238)+1.738918973183969*z(207)*z(2218)+1.738918973183969*z(208)*z(2228)+1.738918973183969*z(209)*z(2238)-
1.564399813392072*z(2218)-1.445018572213135*z(213)*z(2218)-1.445018572213135*z(214)*z(2228)-1.445018572213135*z(215)*z(2238)) -
0.00404624*z(502)*(z(1030)+2.01888419866345*z(1034)+1.135663974455297*z(132)*z(1026)+1.135663974455297*z(133)*z(1030)+1.13566397445
5297*z(134)*z(1034)+1.613396387757523*z(129)*z(1026)+1.613396387757523*z(130)*z(1030)+1.613396387757523*z(131)*z(1034)-
1.85913242911938*z(1026)-1.514074795365574*z(135)*z(1026)-1.514074795365574*z(136)*z(1030)-1.514074795365574*z(137)*z(1034)) -
0.00297212*z(498)*(z(1593)+2.289867165524945*z(1600)+1.384676594484745*z(168)*z(1586)+1.384676594484745*z(174)*z(1593)+1.3846765944
84745*z(175)*z(1600)+2.471453709809833*z(166)*z(1586)+2.471453709809833*z(172)*z(1593)+2.471453709809833*z(173)*z(1600)-
2.986921120277781*z(1586)-2.589087923771584*z(170)*z(1586)-2.589087923771584*z(176)*z(1593)-2.589087923771584*z(177)*z(1600)) -
z(446)*(2.0E-
07*z(2022)+0.0210342*z(2002)+0.000948567000000000001*z(210)*z(2002)+0.000948567000000000001*z(211)*z(2012)+0.000948567000000000001*z(212
)*z(2022)-0.001343886*z(2012)-0.0196764*z(207)*z(2002)-0.0196764*z(208)*z(2012)-0.0196764*z(209)*z(2022)-0.00039917*z(213)*z(2002)-
0.00039917*z(214)*z(2012)-0.00039917*z(215)*z(2022)) -
0.00021415*z(490)*(40.8116273639972*z(623)+64.04720989960309*z(82)*z(627)+64.04720989960309*z(83)*z(623)+64.04720989960309*z(84)*z(
619)-70.09441979920617*z(627)-4.618921316833995*z(619)-10.35433107634836*z(85)*z(627)-10.35433107634836*z(86)*z(623)-
10.35433107634836*z(87)*z(619)-z(88)*z(627)-z(89)*z(623)-z(90)*z(619)) -
0.000156039*z(466)*(34.63294432802056*z(1370)+4.285595267849704*z(168)*z(1370)+4.285595267849704*z(174)*z(1377)+4.285595267849704*
z(175)*z(1384)-6.601112542377225*z(1377)-z(1384)-34.22740468728971*z(166)*z(1370)-34.22740468728971*z(172)*z(1377)-
34.22740468728971*z(173)*z(1384)-5.76932689904447*z(170)*z(1370)-5.76932689904447*z(176)*z(1377)-5.76932689904447*z(177)*z(1384));
z(3335) = z(252)*z(418) + z(255)*z(428) + z(258)*z(438);
z(3348) = z(253)*z(418) + z(256)*z(428) + z(259)*z(438);
z(3361) = z(254)*z(418) + z(257)*z(428) + z(260)*z(438);
z(3411) = z(418)*z(2730) + z(428)*z(2733) + z(438)*z(2736);
z(3424) = z(418)*z(2731) + z(428)*z(2734) + z(438)*z(2737);
z(3437) = z(418)*z(2732) + z(428)*z(2735) + z(438)*z(2738);
z(3496) = z(418)*z(2868) + z(428)*z(2871) + z(438)*z(2874);
z(3509) = z(418)*z(2869) + z(428)*z(2872) + z(438)*z(2875);
z(3522) = z(418)*z(2870) + z(428)*z(2873) + z(438)*z(2876);
z(4375) = z(80)*z(3148) + z(81)*z(3147) + z(380)*z(3204) + z(384)*z(3203) + z(388)*z(3202) + z(395)*z(3255) + z(402)*z(3254) + z(409)*z(3253) +
z(418)*z(3326) + z(428)*z(3325) + z(438)*z(3324) + z(3335)*z(3402) + z(3348)*z(3401) + z(3361)*z(3400) + z(3411)*z(3487) + z(3424)*z(3486) +
z(3437)*z(3485) + z(3496)*z(3569) + z(3509)*z(3568) + z(3522)*z(3567) + z(80)*z(3145) + z(81)*z(3146) + z(380)*z(3199) + z(384)*z(3200) +
z(388)*z(3201) + z(395)*z(3250) + z(402)*z(3251) + z(409)*z(3252) + z(418)*z(3321) + z(428)*z(3322) + z(438)*z(3323) + z(3335)*z(3397) +
z(3348)*z(3398) + z(3361)*z(3399) + z(3411)*z(3482) + z(3424)*z(3483) + z(3437)*z(3484) + z(3496)*z(3564) + z(3509)*z(3565) + z(3522)*z(3566)
+ SW*(z(2826)*z(4372)+z(2911)*z(4370)+z(2924)*z(4371)) + 0.1428571428571429*SW*(z(547)*z(3598)+z(551)*z(3599)+z(555)*z(3597)) +
0.1428571428571429*SW*(z(774)*z(3651)+z(778)*z(3653)+z(782)*z(3652)) +
0.1428571428571429*SW*(z(1208)*z(3741)+z(1215)*z(3743)+z(1222)*z(3742)) +
0.1428571428571429*SW*(z(1858)*z(3867)+z(1868)*z(3869)+z(1878)*z(3868)) +
0.1428571428571429*SW*(z(2499)*z(4020)+z(2512)*z(4022)+z(2525)*z(4021)) - z(10)*z(3149) - z(10)*z(3144) -
0.1428571428571429*SW*(z(34)*z(2660)*z(4191)+z(34)*z(2700)*z(4190)-z(2660)*z(4190)-z(2673)*z(4176)-z(2686)*z(4162)-z(2700)*z(4191)-
z(2713)*z(4193)-z(2726)*z(4192)-z(250)*z(2660)*z(4193)-z(250)*z(2713)*z(4190)-z(251)*z(2660)*z(4192)-z(251)*z(2726)*z(4190)-
z(283)*z(2686)*z(4191)-z(283)*z(2700)*z(4162)-z(284)*z(2673)*z(4191)-z(284)*z(2700)*z(4176)-z(285)*z(2686)*z(4193)-z(285)*z(2713)*z(4162)-
z(286)*z(2673)*z(4193)-z(286)*z(2713)*z(4176)-z(287)*z(2686)*z(4192)-z(287)*z(2726)*z(4162)-z(288)*z(2673)*z(4192)-z(288)*z(2726)*z(4176));
z(4399) = z(2967) - z(4375);
z(4446) = z(4434)*z(4397) - z(4399);
z(4575) = z(4446) - z(4564)*z(4433);
z(4782) = z(4683) - z(4772)*z(4575);
z(1741) = -z(403)*DODC3 - z(410)*z(1713);
z(1748) = z(1152) + z(1741);
z(1720) = z(396)*z(1713) - z(403)*z(1712);
z(1762) = z(1172) + z(1720);
z(1730) = z(396)*DODC3 + z(410)*z(1712);
z(1755) = z(1162) + z(1730);
z(2935) = z(19)*z(1748) + z(195)*z(1762) - z(193)*z(1755);
z(4554) = z(2935)*z(4551);
z(1742) = -z(404)*DODC3 - z(411)*z(1713);
z(1749) = z(1153) + z(1742);
z(1721) = z(397)*z(1713) - z(404)*z(1712);
z(1763) = z(1173) + z(1721);
z(1731) = z(397)*DODC3 + z(411)*z(1712);
z(1756) = z(1163) + z(1731);
z(2948) = z(19)*z(1749) + z(195)*z(1763) - z(193)*z(1756);

```
z(4424) = z(2948) - z(2935)*z(4421);
z(4674) = -z(4554) - z(4424)*z(4672);
z(1740) = -z(402)*DODC3 - z(409)*z(1713);
z(1747) = z(1151) + z(1740);
z(1719) = z(395)*z(1713) - z(402)*z(1712);
z(1761) = z(1171) + z(1719);
z(1729) = z(395)*DODC3 + z(409)*z(1712);
z(1754) = z(1161) + z(1729);
z(2961) = z(19)*z(1747) + z(195)*z(1761) - z(193)*z(1754);
z(4437) = z(2961) - z(2935)*z(4434);
z(4566) = z(4437) - z(4424)*z(4564);
z(4773) = z(4674) - z(4566)*z(4772);
z(1737) = -z(399)*DODC3 - z(406)*z(1713);
z(1744) = z(1148) + z(1737);
z(1716) = z(392)*z(1713) - z(399)*z(1712);
z(1758) = z(1168) + z(1716);
z(1726) = z(392)*DODC3 + z(406)*z(1712);
z(1751) = z(1158) + z(1726);
z(2971) = z(19)*z(1744) + z(195)*z(1758) - z(193)*z(1751);
z(2347) = -z(425)*EOED3 - z(435)*z(2315);
z(2358) = z(1774) + z(2347);
z(2334) = z(415)*EOED3 + z(435)*z(2314);
z(2368) = z(1790) + z(2334);
z(2321) = z(415)*z(2315) - z(425)*z(2314);
z(2378) = z(1806) + z(2321);
z(2974) = z(197)*z(2358) + z(201)*z(2368) + z(202)*z(2378);
z(4447) = z(2974)/z(2938);
z(4450) = z(2971) - z(2935)*z(4447);
z(2975) = z(199)*z(2358) + z(203)*z(2368) + z(204)*z(2378);
z(4448) = z(2975) - z(2939)*z(4447);
z(4576) = z(4448)/z(4422);
z(4578) = z(4450) - z(4424)*z(4576);
z(2976) = z(205)*z(2368) + z(206)*z(2378) - z(28)*z(2358);
z(4449) = z(2976) - z(2940)*z(4447);
z(4577) = z(4449) - z(4423)*z(4576);
z(4684) = z(4577)/z(4565);
z(4685) = z(4578) - z(4566)*z(4684);
z(4853) = z(4773)/z(4685);
z(2034) = z(415)*z(2027) - z(425)*z(2026);
z(2091) = z(1806) + z(2034);
z(2047) = z(435)*z(2026) - z(415)*z(2028);
z(2081) = z(1790) + z(2047);
z(2060) = z(425)*z(2028) - z(435)*z(2027);
z(2071) = z(1774) + z(2060);
z(832) = z(124)*z(822) - z(119)*z(824);
z(839) = z(123)*z(824) - z(124)*z(823);
z(825) = z(119)*z(823) - z(123)*z(822);
z(897) = z(119)*z(895) - z(123)*z(894);
z(911) = z(123)*z(896) - z(124)*z(895);
z(904) = z(124)*z(894) - z(119)*z(896);
z(2106) = z(415)*z(2099) - z(425)*z(2098);
z(2163) = z(1806) + z(2106);
z(2132) = z(425)*z(2100) - z(435)*z(2099);
z(2143) = z(1774) + z(2132);
z(2119) = z(435)*z(2098) - z(415)*z(2100);
z(2153) = z(1790) + z(2119);
z(1393) = z(392)*z(1389) - z(399)*z(1388);
z(1435) = z(1168) + z(1393);
z(1414) = z(399)*z(1390) - z(406)*z(1389);
z(1421) = z(1148) + z(1414);
z(1403) = z(406)*z(1388) - z(392)*z(1390);
z(1428) = z(1158) + z(1403);
z(868) = z(124)*z(858) - z(119)*z(860);
z(875) = z(123)*z(860) - z(124)*z(859);
z(861) = z(119)*z(859) - z(123)*z(858);
z(1663) = z(392)*z(1659) - z(399)*z(1658);
z(1705) = z(1168) + z(1663);
z(1684) = z(399)*z(1660) - z(406)*z(1659);
z(1691) = z(1148) + z(1684);
z(1673) = z(406)*z(1658) - z(392)*z(1660);
z(1698) = z(1158) + z(1673);
z(1077) = z(119)*z(1075) - z(123)*z(1074);
z(1091) = z(123)*z(1076) - z(124)*z(1075);
z(1084) = z(124)*z(1074) - z(119)*z(1076);
z(2250) = z(415)*z(2243) - z(425)*z(2242);
z(2307) = z(1806) + z(2250);
z(2276) = z(425)*z(2244) - z(435)*z(2243);
z(2287) = z(1774) + z(2276);
```

z(2263) = z(435)*z(2242) - z(415)*z(2244);
z(2297) = z(1790) + z(2263);
z(1285) = z(392)*z(1281) - z(399)*z(1280);
z(1327) = z(1168) + z(1285);
z(1306) = z(399)*z(1282) - z(406)*z(1281);
z(1313) = z(1148) + z(1306);
z(1295) = z(406)*z(1280) - z(392)*z(1282);
z(1320) = z(1158) + z(1295);
z(2191) = z(435)*z(2170) - z(415)*z(2172);
z(2225) = z(1790) + z(2191);
z(2178) = z(415)*z(2171) - z(425)*z(2170);
z(2235) = z(1806) + z(2178);
z(2204) = z(425)*z(2172) - z(435)*z(2171);
z(2215) = z(1774) + z(2204);
z(1012) = z(124)*z(1002) - z(119)*z(1004);
z(1005) = z(119)*z(1003) - z(123)*z(1002);
z(1019) = z(123)*z(1004) - z(124)*z(1003);
z(1565) = z(406)*z(1550) - z(392)*z(1552);
z(1590) = z(1158) + z(1565);
z(1555) = z(392)*z(1551) - z(399)*z(1550);
z(1597) = z(1168) + z(1555);
z(1576) = z(399)*z(1552) - z(406)*z(1551);
z(1583) = z(1148) + z(1576);
z(1962) = z(415)*z(1955) - z(425)*z(1954);
z(2019) = z(1806) + z(1962);
z(1988) = z(425)*z(1956) - z(435)*z(1955);
z(1999) = z(1774) + z(1988);
z(1975) = z(435)*z(1954) - z(415)*z(1956);
z(2009) = z(1790) + z(1975);
z(1360) = z(399)*z(1336) - z(406)*z(1335);
z(1367) = z(1148) + z(1360);
z(1349) = z(406)*z(1334) - z(392)*z(1336);
z(1374) = z(1158) + z(1349);
z(1339) = z(392)*z(1335) - z(399)*z(1334);
z(1381) = z(1168) + z(1339);
z(2977) = 4.7299E-
05*z(462)*(z(2091)+44.21104040254551*z(2081)+114.2537897207129*z(207)*z(2071)+114.2537897207129*z(208)*z(2081)+114.2537897207129*z(
209)*z(2091)-147.8697224042791*z(2071)-21.77701431319901*z(210)*z(2071)-21.77701431319901*z(211)*z(2081)-
21.77701431319901*z(212)*z(2091)-3.298991522019493*z(213)*z(2071)-3.298991522019493*z(214)*z(2081)-3.298991522019493*z(215)*z(2091))
+ 4.9681E-05*z(454)*(z(832)+436.869225659709*z(129)*z(839)+436.869225659709*z(130)*z(832)+436.869225659709*z(131)*z(825)-
410.7868199110324*z(839)-25.99182786175802*z(825)-23.60302731426501*z(132)*z(839)-23.60302731426501*z(133)*z(832)-
23.60302731426501*z(134)*z(825)-4.101447233348765*z(135)*z(839)-4.101447233348765*z(136)*z(832)-4.101447233348765*z(137)*z(825)) +
5.641700000000001E-
05*z(486)*(z(897)+247.6572664267862*z(911)+24.25456865838311*z(132)*z(911)+24.25456865838311*z(133)*z(904)+24.25456865838311*z(134)
*z(897)-149.1704628037648*z(904)-269.1738305829803*z(129)*z(911)-269.1738305829803*z(130)*z(904)-269.1738305829803*z(131)*z(897)-
14.71584805998192*z(135)*z(911)-14.71584805998192*z(136)*z(904)-14.71584805998192*z(137)*z(897)) +
0.000263142*z(478)*(z(2163)+57.96163288262611*z(2143)-37.04060925279887*z(2153)-58.17953804409786*z(207)*z(2143)-
58.17953804409786*z(208)*z(2153)-58.17953804409786*z(209)*z(2163)-3.892632114979745*z(213)*z(2143)-3.892632114979745*z(214)*z(2153)-
3.892632114979745*z(215)*z(2163)-3.889998555912777*z(210)*z(2143)-3.889998555912777*z(211)*z(2153)-3.889998555912777*z(212)*z(2163))
+ 6.128E-05*z(482)*(19.44327676240209*z(1435)+238.8415469973891*z(1421)-120.5957898172324*z(1428)-232.8573759791123*z(166)*z(1421)-
232.8573759791123*z(172)*z(1428)-232.8573759791123*z(173)*z(1435)-1.221507832898173*z(168)*z(1421)-1.221507832898173*z(174)*z(1428)-
1.221507832898173*z(175)*z(1435)-z(170)*z(1421)-z(176)*z(1428)-z(177)*z(1435)) +
0.000439106*z(470)*(1.522912463049924*z(868)+15.07109900570705*z(129)*z(875)+15.07109900570705*z(130)*z(868)+15.07109900570705*z(13
1)*z(861)-12.16291738213552*z(875)-2.050165563667998*z(861)-6.399042600192209*z(132)*z(875)-6.399042600192209*z(133)*z(868)-
6.399042600192209*z(134)*z(861)-z(135)*z(875)-z(136)*z(868)-z(137)*z(861)) +
0.00334198*z(514)*(2.523796072986673*z(1705)+2.541448183412228*z(1691)-z(1698)-2.199101730112089*z(166)*z(1691)-
2.199101730112089*z(172)*z(1698)-2.199101730112089*z(173)*z(1705)-2.178312856450368*z(170)*z(1691)-2.178312856450368*z(176)*z(1698)-
2.178312856450368*z(177)*z(1705)-1.513417495017923*z(168)*z(1691)-1.513417495017923*z(174)*z(1698)-1.513417495017923*z(175)*z(1705))
+ 0.00353959*z(518)*(2.051998112775774*z(1077)+2.135415683737382*z(1091)-z(1084)-2.270871202596911*z(132)*z(1091)-
2.270871202596911*z(133)*z(1084)-2.270871202596911*z(134)*z(1077)-2.253045691732658*z(135)*z(1091)-2.253045691732658*z(136)*z(1084)-
2.253045691732658*z(137)*z(1077)-1.758446882266025*z(129)*z(1091)-1.758446882266025*z(130)*z(1084)-1.758446882266025*z(131)*z(1077))
+ 0.005250594000000001*z(510)*(1.3792740402324*z(2307)+1.576289654084852*z(2287)-1.220873295478569*z(2297)-
1.749405305380686*z(207)*z(2287)-1.749405305380686*z(208)*z(2297)-1.749405305380686*z(209)*z(2307)-1.677448684853561*z(213)*z(2287)-
1.677448684853561*z(214)*z(2297)-1.677448684853561*z(215)*z(2307)-z(210)*z(2287)-z(211)*z(2297)-z(212)*z(2307)) -
9.800000000000001*SW*(z(2790)*z(2908)+z(2791)*z(2823)+z(2792)*z(2921)) - 1.4*SW*(z(121)*z(767)+z(125)*z(760)+z(126)*z(753)) -
1.4*SW*(z(199)*z(1855)+z(203)*z(1865)+z(204)*z(1875)) - 1.4*SW*(z(284)*z(2496)+z(286)*z(2509)+z(288)*z(2522)) -
1.4*SW*(z(20)*z(1205)+z(194)*z(1212)-z(196)*z(1219)) - 1.4*SW*(z(2670)+z(284)*z(2697)+z(286)*z(2710)+z(288)*z(2723)) - 4.9681E-
05*z(450)*(8.034661138060828*z(1327)+396.0548298142147*z(1313)+z(168)*z(1313)+z(174)*z(1320)+z(175)*z(1327)-
19.09315432459089*z(1320)-410.7868199110324*z(166)*z(1313)-410.7868199110324*z(172)*z(1320)-410.7868199110324*z(173)*z(1327)-
25.99182786175802*z(170)*z(1313)-25.99182786175802*z(176)*z(1320)-25.99182786175802*z(177)*z(1327)) -
0.00508017*z(494)*(z(2225)+1.575266182037215*z(2235)+1.26191682561804*z(210)*z(2215)+1.26191682561804*z(211)*z(2225)+1.261916825618
04*z(212)*z(2235)+1.738918973183969*z(207)*z(2215)+1.738918973183969*z(208)*z(2225)+1.738918973183969*z(209)*z(2235)-
1.564399813392072*z(2215)-1.445018572213135*z(213)*z(2215)-1.445018572213135*z(214)*z(2225)-1.445018572213135*z(215)*z(2235)) -
0.00404624*z(502)*(z(1012)+2.01888419866345*z(1005)+1.135663974455297*z(132)*z(1019)+1.135663974455297*z(133)*z(1012)+1.13566397445
5297*z(134)*z(1005)+1.613396387757523*z(129)*z(1019)+1.613396387757523*z(130)*z(1012)+1.613396387757523*z(131)*z(1005)-
1.85913242911938*z(1019)-1.514074795365574*z(135)*z(1019)-1.514074795365574*z(136)*z(1012)-1.514074795365574*z(137)*z(1005)) -
0.00297212*z(498)*(z(1590)+2.289867165524945*z(1597)+1.384676594484745*z(168)*z(1583)+1.384676594484745*z(174)*z(1590)+1.3846765944
84745*z(175)*z(1597)+2.471453709809833*z(166)*z(1583)+2.471453709809833*z(172)*z(1590)+2.471453709809833*z(173)*z(1597)-

2.986921120277781*z(1583)-2.589087923771584*z(170)*z(1583)-2.589087923771584*z(176)*z(1590)-2.589087923771584*z(177)*z(1597)) -
z(446)*(2.0E-
07*z(2019)+0.0210342*z(1999)+0.0009485670000000001*z(210)*z(1999)+0.0009485670000000001*z(211)*z(2009)+0.0009485670000000001*z(212
)*z(2019)-0.001343886*z(2009)-0.0196764*z(207)*z(1999)-0.0196764*z(208)*z(2009)-0.0196764*z(209)*z(2019)-0.00039917*z(213)*z(1999)-
0.00039917*z(214)*z(2009)-0.00039917*z(215)*z(2019)) -
0.000156039*z(466)*(34.63294432802056*z(1367)+4.285595267849704*z(168)*z(1367)+4.285595267849704*z(174)*z(1374)+4.285595267849704*
z(175)*z(1381)-6.601112542377225*z(1374)-z(1381)-34.22740468728971*z(166)*z(1367)-34.22740468728971*z(172)*z(1374)-
34.22740468728971*z(173)*z(1381)-5.76932689904447*z(170)*z(1367)-5.76932689904447*z(176)*z(1374)-5.76932689904447*z(177)*z(1381));
z(3332) = z(252)*z(415) + z(255)*z(425) + z(258)*z(435);
z(3345) = z(253)*z(415) + z(256)*z(425) + z(259)*z(435);
z(3358) = z(254)*z(415) + z(257)*z(425) + z(260)*z(435);
z(3408) = z(415)*z(2730) + z(425)*z(2733) + z(435)*z(2736);
z(3421) = z(415)*z(2731) + z(425)*z(2734) + z(435)*z(2737);
z(3434) = z(415)*z(2732) + z(425)*z(2735) + z(435)*z(2738);
z(3493) = z(415)*z(2868) + z(425)*z(2871) + z(435)*z(2874);
z(3506) = z(415)*z(2869) + z(425)*z(2872) + z(435)*z(2875);
z(3519) = z(415)*z(2870) + z(425)*z(2873) + z(435)*z(2876);
z(4376) = z(119)*z(3204) + z(123)*z(3203) + z(124)*z(3202) + z(392)*z(3255) + z(399)*z(3254) + z(406)*z(3253) + z(415)*z(3326) + z(425)*z(3325)
+ z(435)*z(3324) + z(3332)*z(3402) + z(3345)*z(3401) + z(3358)*z(3400) + z(3408)*z(3487) + z(3421)*z(3486) + z(3434)*z(3485) + z(3493)*z(3569)
+ z(3506)*z(3568) + z(3519)*z(3567) + z(119)*z(3199) + z(123)*z(3200) + z(124)*z(3201) + z(392)*z(3250) + z(399)*z(3251) + z(406)*z(3252) +
z(415)*z(3321) + z(425)*z(3322) + z(435)*z(3323) + z(3332)*z(3397) + z(3345)*z(3398) + z(3358)*z(3399) + z(3408)*z(3482) + z(3421)*z(3483) +
z(3434)*z(3484) + z(3493)*z(3564) + z(3506)*z(3565) + z(3519)*z(3566) + SW*(z(2823)*z(4372)+z(2908)*z(4370)+z(2921)*z(4371)) +
0.1428571428571429*SW*(z(753)*z(3652)+z(760)*z(3653)+z(767)*z(3651)) +
0.1428571428571429*SW*(z(1205)*z(3741)+z(1212)*z(3743)+z(1219)*z(3742)) +
0.1428571428571429*SW*(z(1855)*z(3867)+z(1865)*z(3869)+z(1875)*z(3868)) +
0.1428571428571429*SW*(z(2496)*z(4020)+z(2509)*z(4022)+z(2522)*z(4021)) -
0.1428571428571429*SW*(z(34)*z(2657)*z(4191)+z(34)*z(2697)*z(4190)-z(2657)*z(4190)-z(2670)*z(4176)-z(2683)*z(4162)-z(2697)*z(4191)-
z(2710)*z(4193)-z(2723)*z(4192)-z(250)*z(2657)*z(4193)-z(250)*z(2710)*z(4190)-z(251)*z(2657)*z(4192)-z(251)*z(2723)*z(4190)-
z(283)*z(2683)*z(4191)-z(283)*z(2697)*z(4162)-z(284)*z(2670)*z(4191)-z(284)*z(2697)*z(4176)-z(285)*z(2683)*z(4193)-z(285)*z(2710)*z(4162)-
z(286)*z(2670)*z(4193)-z(286)*z(2710)*z(4176)-z(287)*z(2683)*z(4192)-z(287)*z(2723)*z(4162)-z(288)*z(2670)*z(4192)-z(288)*z(2723)*z(4176));
z(4400) = z(2977) - z(4376);
z(4459) = z(4447)*z(4397) - z(4400);
z(4587) = z(4459) - z(4576)*z(4433);
z(4694) = z(4587) - z(4684)*z(4575);
z(4862) = z(4782) - z(4853)*z(4694);
z(2936) = z(20)*z(1748) + z(194)*z(1755) - z(196)*z(1762);
z(4555) = z(2936)*z(4551);
z(2949) = z(20)*z(1749) + z(194)*z(1756) - z(196)*z(1763);
z(4425) = z(2949) - z(2936)*z(4421);
z(4675) = -z(4555) - z(4425)*z(4672);
z(2962) = z(20)*z(1747) + z(194)*z(1754) - z(196)*z(1761);
z(4438) = z(2962) - z(2936)*z(4434);
z(4567) = z(4438) - z(4425)*z(4564);
z(4774) = z(4675) - z(4567)*z(4772);
z(2972) = z(20)*z(1744) + z(194)*z(1751) - z(196)*z(1758);
z(4451) = z(2972) - z(2936)*z(4447);
z(4579) = z(4451) - z(4425)*z(4576);
z(4686) = z(4579) - z(4567)*z(4684);
z(4854) = z(4774) - z(4686)*z(4853);
z(1738) = -z(400)*DODC3 - z(407)*z(1713);
z(1745) = z(1149) + z(1738);
z(1727) = z(393)*DODC3 + z(407)*z(1712);
z(1752) = z(1159) + z(1727);
z(1717) = z(393)*z(1713) - z(400)*z(1712);
z(1759) = z(1169) + z(1717);
z(2982) = z(20)*z(1745) + z(194)*z(1752) - z(196)*z(1759);
z(2348) = -z(426)*EOED3 - z(436)*z(2315);
z(2359) = z(1775) + z(2348);
z(2335) = z(416)*EOED3 + z(436)*z(2314);
z(2369) = z(1791) + z(2335);
z(2322) = z(416)*z(2315) - z(426)*z(2314);
z(2379) = z(1807) + z(2322);
z(2984) = z(197)*z(2359) + z(201)*z(2369) + z(202)*z(2379);
z(4460) = z(2984)/z(2938);
z(4464) = z(2982) - z(2936)*z(4460);
z(2985) = z(199)*z(2359) + z(203)*z(2369) + z(204)*z(2379);
z(4461) = z(2985) - z(2939)*z(4460);
z(4588) = z(4461)/z(4422);
z(4591) = z(4464) - z(4425)*z(4588);
z(2986) = z(205)*z(2369) + z(206)*z(2379) - z(28)*z(2359);
z(4462) = z(2986) - z(2940)*z(4460);
z(4589) = z(4462) - z(4423)*z(4588);
z(4695) = z(4589)/z(4565);
z(4697) = z(4591) - z(4567)*z(4695);
z(2981) = z(19)*z(1745) + z(195)*z(1759) - z(193)*z(1752);
z(4463) = z(2981) - z(2935)*z(4460);
z(4590) = z(4463) - z(4424)*z(4588);
z(4696) = z(4590) - z(4566)*z(4695);

```
z(4783) = z(4696)/z(4685);
z(4784) = z(4697) - z(4686)*z(4783);
z(4917) = z(4854)/z(4784);
z(2035) = z(416)*z(2027) - z(426)*z(2026);
z(2092) = z(1807) + z(2035);
z(2048) = z(436)*z(2026) - z(416)*z(2028);
z(2082) = z(1791) + z(2048);
z(2061) = z(426)*z(2028) - z(436)*z(2027);
z(2072) = z(1775) + z(2061);
z(833) = z(126)*z(822) - z(121)*z(824);
z(840) = z(125)*z(824) - z(126)*z(823);
z(826) = z(121)*z(823) - z(125)*z(822);
z(898) = z(121)*z(895) - z(125)*z(894);
z(912) = z(125)*z(896) - z(126)*z(895);
z(905) = z(126)*z(894) - z(121)*z(896);
z(2107) = z(416)*z(2099) - z(426)*z(2098);
z(2164) = z(1807) + z(2107);
z(2133) = z(426)*z(2100) - z(436)*z(2099);
z(2144) = z(1775) + z(2133);
z(2120) = z(436)*z(2098) - z(416)*z(2100);
z(2154) = z(1791) + z(2120);
z(1394) = z(393)*z(1389) - z(400)*z(1388);
z(1436) = z(1169) + z(1394);
z(1415) = z(400)*z(1390) - z(407)*z(1389);
z(1422) = z(1149) + z(1415);
z(1404) = z(407)*z(1388) - z(393)*z(1390);
z(1429) = z(1159) + z(1404);
z(869) = z(126)*z(858) - z(121)*z(860);
z(876) = z(125)*z(860) - z(126)*z(859);
z(862) = z(121)*z(859) - z(125)*z(858);
z(1664) = z(393)*z(1659) - z(400)*z(1658);
z(1706) = z(1169) + z(1664);
z(1685) = z(400)*z(1660) - z(407)*z(1659);
z(1692) = z(1149) + z(1685);
z(1674) = z(407)*z(1658) - z(393)*z(1660);
z(1699) = z(1159) + z(1674);
z(1078) = z(121)*z(1075) - z(125)*z(1074);
z(1092) = z(125)*z(1076) - z(126)*z(1075);
z(1085) = z(126)*z(1074) - z(121)*z(1076);
z(2251) = z(416)*z(2243) - z(426)*z(2242);
z(2308) = z(1807) + z(2251);
z(2277) = z(426)*z(2244) - z(436)*z(2243);
z(2288) = z(1775) + z(2277);
z(2264) = z(436)*z(2242) - z(416)*z(2244);
z(2298) = z(1791) + z(2264);
z(1286) = z(393)*z(1281) - z(400)*z(1280);
z(1328) = z(1169) + z(1286);
z(1307) = z(400)*z(1282) - z(407)*z(1281);
z(1314) = z(1149) + z(1307);
z(1296) = z(407)*z(1280) - z(393)*z(1282);
z(1321) = z(1159) + z(1296);
z(2192) = z(436)*z(2170) - z(416)*z(2172);
z(2226) = z(1791) + z(2192);
z(2179) = z(416)*z(2171) - z(426)*z(2170);
z(2236) = z(1807) + z(2179);
z(2205) = z(426)*z(2172) - z(436)*z(2171);
z(2216) = z(1775) + z(2205);
z(1013) = z(126)*z(1002) - z(121)*z(1004);
z(1006) = z(121)*z(1003) - z(125)*z(1002);
z(1020) = z(125)*z(1004) - z(126)*z(1003);
z(1566) = z(407)*z(1550) - z(393)*z(1552);
z(1591) = z(1159) + z(1566);
z(1556) = z(393)*z(1551) - z(400)*z(1550);
z(1598) = z(1169) + z(1556);
z(1577) = z(400)*z(1552) - z(407)*z(1551);
z(1584) = z(1149) + z(1577);
z(1963) = z(416)*z(1955) - z(426)*z(1954);
z(2020) = z(1807) + z(1963);
z(1989) = z(426)*z(1956) - z(436)*z(1955);
z(2000) = z(1775) + z(1989);
z(1976) = z(436)*z(1954) - z(416)*z(1956);
z(2010) = z(1791) + z(1976);
z(1361) = z(400)*z(1336) - z(407)*z(1335);
z(1368) = z(1149) + z(1361);
z(1350) = z(407)*z(1334) - z(393)*z(1336);
z(1375) = z(1159) + z(1350);
z(1340) = z(393)*z(1335) - z(400)*z(1334);
z(1382) = z(1169) + z(1340);
```

z(2987) = 4.7299E-
05*z(462)*(z(2092)+44.21104040254551*z(2082)+114.2537897207129*z(207)*z(2072)+114.2537897207129*z(208)*z(2082)+114.2537897207129*z(
209)*z(2092)-147.8697224042791*z(2072)-21.77701431319901*z(210)*z(2072)-21.77701431319901*z(211)*z(2082)-
21.77701431319901*z(212)*z(2092)-3.298991522019493*z(213)*z(2072)-3.298991522019493*z(214)*z(2082)-3.298991522019493*z(215)*z(2092))
+ 4.9681E-05*z(454)*(z(833)+436.869225659709*z(129)*z(840)+436.869225659709*z(130)*z(833)+436.869225659709*z(131)*z(826)-
410.7868199110324*z(840)-25.99182786175802*z(826)-23.60302731426501*z(132)*z(840)-23.60302731426501*z(133)*z(833)-
23.60302731426501*z(134)*z(826)-4.101447233348765*z(135)*z(840)-4.101447233348765*z(136)*z(833)-4.101447233348765*z(137)*z(826)) +
5.641700000000001E-
05*z(486)*(z(898)+247.6572664267862*z(912)+24.25456865838311*z(132)*z(912)+24.25456865838311*z(133)*z(905)+24.25456865838311*z(134)
*z(898)-149.1704628037648*z(905)-269.1738305829803*z(129)*z(912)-269.1738305829803*z(130)*z(905)-269.1738305829803*z(131)*z(898)-
14.71584805998192*z(135)*z(912)-14.71584805998192*z(136)*z(905)-14.71584805998192*z(137)*z(898)) +
0.000263142*z(478)*(z(2164)+57.96163288262611*z(2144)-37.04060925279887*z(2154)-58.17953804409786*z(207)*z(2144)-
58.17953804409786*z(208)*z(2154)-58.17953804409786*z(209)*z(2164)-3.892632114979745*z(213)*z(2144)-3.892632114979745*z(214)*z(2154)-
3.892632114979745*z(215)*z(2164)-3.889998555912777*z(210)*z(2144)-3.889998555912777*z(211)*z(2154)-3.889998555912777*z(212)*z(2164))
+ 6.128E-05*z(482)*(19.44327676240209*z(1436)+238.8415469973891*z(1422)-120.5957898172324*z(1429)-232.8573759791123*z(166)*z(1422)-
232.8573759791123*z(172)*z(1429)-232.8573759791123*z(173)*z(1436)-1.221507832898173*z(168)*z(1422)-1.221507832898173*z(174)*z(1429)-
1.221507832898173*z(175)*z(1436)-z(170)*z(1422)-z(176)*z(1429)-z(177)*z(1436)) +
0.000439106*z(470)*(1.522912463049924*z(869)+15.07109900570705*z(129)*z(876)+15.07109900570705*z(130)*z(869)+15.07109900570705*z(13
1)*z(862)-12.16291738213552*z(876)-2.050165563667998*z(862)-6.399042600192209*z(132)*z(876)-6.399042600192209*z(133)*z(869)-
6.399042600192209*z(134)*z(862)-z(135)*z(876)-z(136)*z(869)-z(137)*z(862)) +
0.00334198*z(514)*(2.523796072986673*z(1706)+2.541448183412228*z(1692)-z(1699)-2.199101730112089*z(166)*z(1692)-
2.199101730112089*z(172)*z(1699)-2.199101730112089*z(173)*z(1706)-2.178312856450368*z(170)*z(1692)-2.178312856450368*z(176)*z(1699)-
2.178312856450368*z(177)*z(1706)-1.513417495017923*z(168)*z(1692)-1.513417495017923*z(174)*z(1699)-1.513417495017923*z(175)*z(1706))
+ 0.00353959*z(518)*(2.051998112775774*z(1078)+2.135415683737382*z(1092)-z(1085)-2.270871202596911*z(132)*z(1092)-
2.270871202596911*z(133)*z(1085)-2.270871202596911*z(134)*z(1078)-2.253045691732658*z(135)*z(1092)-2.253045691732658*z(136)*z(1085)-
2.253045691732658*z(137)*z(1078)-1.758446882266025*z(129)*z(1092)-1.758446882266025*z(130)*z(1085)-1.758446882266025*z(131)*z(1078))
+ 0.005250594000000001*z(510)*(1.3792740402324*z(2308)+1.576289654084852*z(2288)-1.220873295478569*z(2298)-
1.749405305380686*z(207)*z(2288)-1.749405305380686*z(208)*z(2298)-1.749405305380686*z(209)*z(2308)-1.677448684853561*z(213)*z(2288)-
1.677448684853561*z(214)*z(2298)-1.677448684853561*z(215)*z(2308)-z(210)*z(2288)-z(211)*z(2298)-z(212)*z(2308)) -
9.800000000000001*SW*(z(2790)*z(2909)+z(2791)*z(2824)+z(2792)*z(2922)) - 1.4*SW*(z(121)*z(768)+z(125)*z(761)+z(126)*z(754)) -
1.4*SW*(z(199)*z(1856)+z(203)*z(1866)+z(204)*z(1876)) - 1.4*SW*(z(284)*z(2497)+z(286)*z(2510)+z(288)*z(2523)) -
1.4*SW*(z(20)*z(1206)+z(194)*z(1213)-z(196)*z(1220)) - 1.4*SW*(z(2671)+z(284)*z(2698)+z(286)*z(2711)+z(288)*z(2724)) - 4.9681E-
05*z(450)*(8.034661138060828*z(1328)+396.0548298142147*z(1314)+z(168)*z(1314)+z(174)*z(1321)+z(175)*z(1328)-
19.09315432459089*z(1321)-410.7868199110324*z(166)*z(1314)-410.7868199110324*z(172)*z(1321)-410.7868199110324*z(173)*z(1328)-
25.99182786175802*z(170)*z(1314)-25.99182786175802*z(176)*z(1321)-25.99182786175802*z(177)*z(1328)) -
0.00508017*z(494)*(z(2226)+1.575266182037215*z(2236)+1.26191682561804*z(210)*z(2216)+1.26191682561804*z(211)*z(2226)+1.261916825618
04*z(212)*z(2236)+1.738918973183969*z(207)*z(2216)+1.738918973183969*z(208)*z(2226)+1.738918973183969*z(209)*z(2236)-
1.564399813392072*z(2216)-1.445018572213135*z(213)*z(2216)-1.445018572213135*z(214)*z(2226)-1.445018572213135*z(215)*z(2236)) -
0.00404624*z(502)*(z(1013)+2.01888419866345*z(1006)+1.135663974455297*z(132)*z(1020)+1.135663974455297*z(133)*z(1013)+1.13566397445
5297*z(134)*z(1006)+1.613396387757523*z(129)*z(1020)+1.613396387757523*z(130)*z(1013)+1.613396387757523*z(131)*z(1006)-
1.85913242911938*z(1020)-1.514074795365574*z(135)*z(1020)-1.514074795365574*z(136)*z(1013)-1.514074795365574*z(137)*z(1006)) -
0.00297212*z(498)*(z(1591)+2.289867165524945*z(1598)+1.384676594484745*z(168)*z(1584)+1.384676594484745*z(174)*z(1591)+1.3846765944
84745*z(175)*z(1598)+2.471453709809833*z(166)*z(1584)+2.471453709809833*z(172)*z(1591)+2.471453709809833*z(173)*z(1598)-
2.986921120277781*z(1584)-2.589087923771584*z(170)*z(1584)-2.589087923771584*z(176)*z(1591)-2.589087923771584*z(177)*z(1598)) -
z(446)*(2.0E-
07*z(2020)+0.0210342*z(2000)+0.0009485670000000001*z(210)*z(2000)+0.0009485670000000001*z(211)*z(2010)+0.0009485670000000001*z(212
)*z(2020)-0.001343886*z(2010)-0.0196764*z(207)*z(2000)-0.0196764*z(208)*z(2010)-0.0196764*z(209)*z(2020)-0.00039917*z(213)*z(2000)-
0.00039917*z(214)*z(2010)-0.00039917*z(215)*z(2020)) -
0.000156039*z(466)*(34.63294432802056*z(1368)+4.285595267849704*z(168)*z(1368)+4.285595267849704*z(174)*z(1375)+4.285595267849704*
z(175)*z(1382)-6.601112542377225*z(1375)-z(1382)-34.22740468728971*z(166)*z(1368)-34.22740468728971*z(172)*z(1375)-
34.22740468728971*z(173)*z(1382)-5.76932689904447*z(170)*z(1368)-5.76932689904447*z(176)*z(1375)-5.76932689904447*z(177)*z(1382));
z(3333) = z(252)*z(416) + z(255)*z(426) + z(258)*z(436);
z(3346) = z(253)*z(416) + z(256)*z(426) + z(259)*z(436);
z(3359) = z(254)*z(416) + z(257)*z(426) + z(260)*z(436);
z(3409) = z(416)*z(2730) + z(426)*z(2733) + z(436)*z(2736);
z(3422) = z(416)*z(2731) + z(426)*z(2734) + z(436)*z(2737);
z(3435) = z(416)*z(2732) + z(426)*z(2735) + z(436)*z(2738);
z(3494) = z(416)*z(2868) + z(426)*z(2871) + z(436)*z(2874);
z(3507) = z(416)*z(2869) + z(426)*z(2872) + z(436)*z(2875);
z(3520) = z(416)*z(2870) + z(426)*z(2873) + z(436)*z(2876);
z(4377) = z(121)*z(3204) + z(125)*z(3203) + z(126)*z(3202) + z(393)*z(3255) + z(400)*z(3254) + z(407)*z(3253) + z(416)*z(3326) + z(426)*z(3325)
+ z(436)*z(3324) + z(3333)*z(3402) + z(3346)*z(3401) + z(3359)*z(3400) + z(3409)*z(3487) + z(3422)*z(3486) + z(3435)*z(3485) + z(3494)*z(3569)
+ z(3507)*z(3568) + z(3520)*z(3567) + z(121)*z(3199) + z(125)*z(3200) + z(126)*z(3201) + z(393)*z(3250) + z(400)*z(3251) + z(407)*z(3252) +
z(416)*z(3321) + z(426)*z(3322) + z(436)*z(3323) + z(3333)*z(3397) + z(3346)*z(3398) + z(3359)*z(3399) + z(3409)*z(3482) + z(3422)*z(3483) +
z(3435)*z(3484) + z(3494)*z(3564) + z(3507)*z(3565) + z(3520)*z(3566) + SW*(z(2824)*z(4372)+z(2909)*z(4370)+z(2922)*z(4371)) +
0.1428571428571429*SW*(z(754)*z(3652)+z(761)*z(3653)+z(768)*z(3651)) +
0.1428571428571429*SW*(z(1206)*z(3741)+z(1213)*z(3743)+z(1220)*z(3742)) +
0.1428571428571429*SW*(z(1856)*z(3867)+z(1866)*z(3869)+z(1876)*z(3868)) +
0.1428571428571429*SW*(z(2497)*z(4020)+z(2510)*z(4022)+z(2523)*z(4021)) -
0.1428571428571429*SW*(z(34)*z(2658)*z(4191)+z(34)*z(2698)*z(4190)-z(2658)*z(4190)-z(2671)*z(4176)-z(2684)*z(4162)-z(2698)*z(4191)-
z(2711)*z(4193)-z(2724)*z(4192)-z(250)*z(2658)*z(4193)-z(250)*z(2711)*z(4190)-z(251)*z(2658)*z(4192)-z(251)*z(2724)*z(4190)-
z(283)*z(2684)*z(4191)-z(283)*z(2698)*z(4162)-z(284)*z(2671)*z(4191)-z(284)*z(2698)*z(4176)-z(285)*z(2684)*z(4193)-z(285)*z(2711)*z(4162)-
z(286)*z(2671)*z(4193)-z(286)*z(2711)*z(4176)-z(287)*z(2684)*z(4192)-z(287)*z(2724)*z(4162)-z(288)*z(2671)*z(4192)-z(288)*z(2724)*z(4176));
z(4401) = z(2987) - z(4377);
z(4472) = z(4460)*z(4397) - z(4401);
z(4599) = z(4472) - z(4588)*z(4433);
z(4705) = z(4599) - z(4695)*z(4575);
z(4792) = z(4705) - z(4783)*z(4694);

```
z(4925) = z(4862) - z(4917)*z(4792);
z(2937) = z(23)*z(1762) + z(24)*z(1755);
z(4556) = z(2937)*z(4551);
z(2950) = z(23)*z(1763) + z(24)*z(1756);
z(4426) = z(2950) - z(2937)*z(4421);
z(4676) = -z(4556) - z(4426)*z(4672);
z(2963) = z(23)*z(1761) + z(24)*z(1754);
z(4439) = z(2963) - z(2937)*z(4434);
z(4568) = z(4439) - z(4426)*z(4564);
z(4775) = z(4676) - z(4568)*z(4772);
z(2973) = z(23)*z(1758) + z(24)*z(1751);
z(4452) = z(2973) - z(2937)*z(4447);
z(4580) = z(4452) - z(4426)*z(4576);
z(4687) = z(4580) - z(4568)*z(4684);
z(4855) = z(4775) - z(4687)*z(4853);
z(2983) = z(23)*z(1759) + z(24)*z(1752);
z(4465) = z(2983) - z(2937)*z(4460);
z(4592) = z(4465) - z(4426)*z(4588);
z(4698) = z(4592) - z(4568)*z(4695);
z(4785) = z(4698) - z(4687)*z(4783);
z(4918) = z(4855) - z(4785)*z(4917);
z(1718) = z(394)*z(1713) - z(401)*z(1712);
z(1760) = z(1170) + z(1718);
z(1728) = z(394)*DODC3 + z(408)*z(1712);
z(1753) = z(1160) + z(1728);
z(2993) = z(23)*z(1760) + z(24)*z(1753);
z(2349) = -z(427)*EOED3 - z(437)*z(2315);
z(2360) = z(1776) + z(2349);
z(2336) = z(417)*EOED3 + z(437)*z(2314);
z(2370) = z(1792) + z(2336);
z(2323) = z(417)*z(2315) - z(427)*z(2314);
z(2380) = z(1808) + z(2323);
z(2994) = z(197)*z(2360) + z(201)*z(2370) + z(202)*z(2380);
z(4473) = z(2994)/z(2938);
z(4478) = z(2993) - z(2937)*z(4473);
z(2995) = z(199)*z(2360) + z(203)*z(2370) + z(204)*z(2380);
z(4474) = z(2995) - z(2939)*z(4473);
z(4600) = z(4474)/z(4422);
z(4604) = z(4478) - z(4426)*z(4600);
z(2996) = z(205)*z(2370) + z(206)*z(2380) - z(28)*z(2360);
z(4475) = z(2996) - z(2940)*z(4473);
z(4601) = z(4475) - z(4423)*z(4600);
z(4706) = z(4601)/z(4565);
z(4709) = z(4604) - z(4568)*z(4706);
z(1739) = -z(401)*DODC3 - z(408)*z(1713);
z(1746) = z(1150) + z(1739);
z(2991) = z(19)*z(1746) + z(195)*z(1760) - z(193)*z(1753);
z(4476) = z(2991) - z(2935)*z(4473);
z(4602) = z(4476) - z(4424)*z(4600);
z(4707) = z(4602) - z(4566)*z(4706);
z(4793) = z(4707)/z(4685);
z(4795) = z(4709) - z(4687)*z(4793);
z(2992) = z(20)*z(1746) + z(194)*z(1753) - z(196)*z(1760);
z(4477) = z(2992) - z(2936)*z(4473);
z(4603) = z(4477) - z(4425)*z(4600);
z(4708) = z(4603) - z(4567)*z(4706);
z(4794) = z(4708) - z(4686)*z(4793);
z(4863) = z(4794)/z(4784);
z(4864) = z(4795) - z(4785)*z(4863);
z(4966) = z(4918)/z(4864);
z(2036) = z(417)*z(2027) - z(427)*z(2026);
z(2093) = z(1808) + z(2036);
z(2049) = z(437)*z(2026) - z(417)*z(2028);
z(2083) = z(1792) + z(2049);
z(2062) = z(427)*z(2028) - z(437)*z(2027);
z(2073) = z(1776) + z(2062);
z(834) = z(16)*z(824) + z(128)*z(822);
z(841) = z(127)*z(824) - z(128)*z(823);
z(827) = -z(16)*z(823) - z(127)*z(822);
z(899) = -z(16)*z(895) - z(127)*z(894);
z(913) = z(127)*z(896) - z(128)*z(895);
z(906) = z(16)*z(896) + z(128)*z(894);
z(2108) = z(417)*z(2099) - z(427)*z(2098);
z(2165) = z(1808) + z(2108);
z(2134) = z(427)*z(2100) - z(437)*z(2099);
z(2145) = z(1776) + z(2134);
z(2121) = z(437)*z(2098) - z(417)*z(2100);
z(2155) = z(1792) + z(2121);
```

z(1395) = z(394)*z(1389) - z(401)*z(1388);
z(1437) = z(1170) + z(1395);
z(1416) = z(401)*z(1390) - z(408)*z(1389);
z(1423) = z(1150) + z(1416);
z(1405) = z(408)*z(1388) - z(394)*z(1390);
z(1430) = z(1160) + z(1405);
z(870) = z(16)*z(860) + z(128)*z(858);
z(877) = z(127)*z(860) - z(128)*z(859);
z(863) = -z(16)*z(859) - z(127)*z(858);
z(1665) = z(394)*z(1659) - z(401)*z(1658);
z(1707) = z(1170) + z(1665);
z(1686) = z(401)*z(1660) - z(408)*z(1659);
z(1693) = z(1150) + z(1686);
z(1675) = z(408)*z(1658) - z(394)*z(1660);
z(1700) = z(1160) + z(1675);
z(1079) = -z(16)*z(1075) - z(127)*z(1074);
z(1093) = z(127)*z(1076) - z(128)*z(1075);
z(1086) = z(16)*z(1076) + z(128)*z(1074);
z(2252) = z(417)*z(2243) - z(427)*z(2242);
z(2309) = z(1808) + z(2252);
z(2278) = z(427)*z(2244) - z(437)*z(2243);
z(2289) = z(1776) + z(2278);
z(2265) = z(437)*z(2242) - z(417)*z(2244);
z(2299) = z(1792) + z(2265);
z(1287) = z(394)*z(1281) - z(401)*z(1280);
z(1329) = z(1170) + z(1287);
z(1308) = z(401)*z(1282) - z(408)*z(1281);
z(1315) = z(1150) + z(1308);
z(1297) = z(408)*z(1280) - z(394)*z(1282);
z(1322) = z(1160) + z(1297);
z(2193) = z(437)*z(2170) - z(417)*z(2172);
z(2227) = z(1792) + z(2193);
z(2180) = z(417)*z(2171) - z(427)*z(2170);
z(2237) = z(1808) + z(2180);
z(2206) = z(427)*z(2172) - z(437)*z(2171);
z(2217) = z(1776) + z(2206);
z(1014) = z(16)*z(1004) + z(128)*z(1002);
z(1007) = -z(16)*z(1003) - z(127)*z(1002);
z(1021) = z(127)*z(1004) - z(128)*z(1003);
z(1567) = z(408)*z(1550) - z(394)*z(1552);
z(1592) = z(1160) + z(1567);
z(1557) = z(394)*z(1551) - z(401)*z(1550);
z(1599) = z(1170) + z(1557);
z(1578) = z(401)*z(1552) - z(408)*z(1551);
z(1585) = z(1150) + z(1578);
z(1964) = z(417)*z(1955) - z(427)*z(1954);
z(2021) = z(1808) + z(1964);
z(1990) = z(427)*z(1956) - z(437)*z(1955);
z(2001) = z(1776) + z(1990);
z(1977) = z(437)*z(1954) - z(417)*z(1956);
z(2011) = z(1792) + z(1977);
z(1362) = z(401)*z(1336) - z(408)*z(1335);
z(1369) = z(1150) + z(1362);
z(1351) = z(408)*z(1334) - z(394)*z(1336);
z(1376) = z(1160) + z(1351);
z(1341) = z(394)*z(1335) - z(401)*z(1334);
z(1383) = z(1170) + z(1341);
z(2997) = 4.7299E-
05*z(462)*(z(2093)+44.21104040254551*z(2083)+114.2537897207129*z(207)*z(2073)+114.2537897207129*z(208)*z(2083)+114.2537897207129*z(
209)*z(2093)-147.8697224042791*z(2073)-21.77701431319901*z(210)*z(2073)-21.77701431319901*z(211)*z(2083)-
21.77701431319901*z(212)*z(2093)-3.298991522019493*z(213)*z(2073)-3.298991522019493*z(214)*z(2083)-3.298991522019493*z(215)*z(2093))
+ 4.9681E-05*z(454)*(z(834)+436.869225659709*z(129)*z(841)+436.869225659709*z(130)*z(834)+436.869225659709*z(131)*z(827)-
410.7868199110324*z(841)-25.99182786175802*z(827)-23.60302731426501*z(132)*z(841)-23.60302731426501*z(133)*z(834)-
23.60302731426501*z(134)*z(827)-4.101447233348765*z(135)*z(841)-4.101447233348765*z(136)*z(834)-4.101447233348765*z(137)*z(827)) +
5.641700000000001E-
05*z(486)*(z(899)+247.6572664267862*z(913)+24.25456865838311*z(132)*z(913)+24.25456865838311*z(133)*z(906)+24.25456865838311*z(134)
*z(899)-149.1704628037648*z(906)-269.1738305829803*z(129)*z(913)-269.1738305829803*z(130)*z(906)-269.1738305829803*z(131)*z(899)-
14.71584805998192*z(135)*z(913)-14.71584805998192*z(136)*z(906)-14.71584805998192*z(137)*z(899)) +
0.000263142*z(478)*(z(2165)+57.96163288262611*z(2145)-37.04060925279887*z(2155)-58.17953804409786*z(207)*z(2145)-
58.17953804409786*z(208)*z(2155)-58.17953804409786*z(209)*z(2165)-3.892632114979745*z(213)*z(2145)-3.892632114979745*z(214)*z(2155)-
3.892632114979745*z(215)*z(2165)-3.889998555912777*z(210)*z(2145)-3.889998555912777*z(211)*z(2155)-3.889998555912777*z(212)*z(2165))
+ 6.128E-05*z(482)*(19.44327676240209*z(1437)+238.8415469973891*z(1423)-120.5957898172324*z(1430)-232.8573759791123*z(166)*z(1423)-
232.8573759791123*z(172)*z(1430)-232.8573759791123*z(173)*z(1437)-1.221507832898173*z(168)*z(1423)-1.221507832898173*z(174)*z(1430)-
1.221507832898173*z(175)*z(1437)-z(170)*z(1423)-z(176)*z(1430)-z(177)*z(1437)) +
0.000439106*z(470)*(1.522912463049924*z(870)+15.07109900570705*z(129)*z(877)+15.07109900570705*z(130)*z(870)+15.07109900570705*z(13
1)*z(863)-12.16291738213552*z(877)-2.050165563667998*z(863)-6.399042600192209*z(132)*z(877)-6.399042600192209*z(133)*z(870)-
6.399042600192209*z(134)*z(863)-z(135)*z(877)-z(136)*z(870)-z(137)*z(863)) +
0.00334198*z(514)*(2.523796072986673*z(1707)+2.541448183412228*z(1693)-z(1700)-2.199101730112089*z(166)*z(1693)-

2.199101730112089*z(172)*z(1700)-2.199101730112089*z(173)*z(1707)-2.178312856450368*z(170)*z(1693)-2.178312856450368*z(176)*z(1700)-
2.178312856450368*z(177)*z(1707)-1.513417495017923*z(168)*z(1693)-1.513417495017923*z(174)*z(1700)-1.513417495017923*z(175)*z(1707))
+ 0.00353959*z(518)*(2.051998112775774*z(1079)+2.135415683737382*z(1093)-z(1086)-2.270871202596911*z(132)*z(1093)-
2.270871202596911*z(133)*z(1086)-2.270871202596911*z(134)*z(1079)-2.253045691732658*z(135)*z(1093)-2.253045691732658*z(136)*z(1086)-
2.253045691732658*z(137)*z(1079)-1.758446882266025*z(129)*z(1093)-1.758446882266025*z(130)*z(1086)-1.758446882266025*z(131)*z(1079))
+ 0.005250594000000001*z(510)*(1.3792740402324*z(2309)+1.576289654084852*z(2289)-1.220873295478569*z(2299)-
1.749405305380686*z(207)*z(2289)-1.749405305380686*z(208)*z(2299)-1.749405305380686*z(209)*z(2309)-1.677448684853561*z(213)*z(2289)-
1.677448684853561*z(214)*z(2299)-1.677448684853561*z(215)*z(2309)-z(210)*z(2289)-z(211)*z(2299)-z(212)*z(2309)) -
9.800000000000001*SW*(z(2790)*z(2910)+z(2791)*z(2825)+z(2792)*z(2923)) - 1.4*SW*(z(121)*z(769)+z(125)*z(762)+z(126)*z(755)) -
1.4*SW*(z(199)*z(1857)+z(203)*z(1867)+z(204)*z(1877)) - 1.4*SW*(z(284)*z(2498)+z(286)*z(2511)+z(288)*z(2524)) -
1.4*SW*(z(20)*z(1207)+z(194)*z(1214)-z(196)*z(1221)) - 1.4*SW*(z(2672)+z(284)*z(2699)+z(286)*z(2712)+z(288)*z(2725)) - 4.9681E-
05*z(450)*(8.034661138060828*z(1329)+396.0548298142147*z(1315)+z(168)*z(1315)+z(174)*z(1322)+z(175)*z(1329)-
19.09315432459089*z(1322)-410.7868199110324*z(166)*z(1315)-410.7868199110324*z(172)*z(1322)-410.7868199110324*z(173)*z(1329)-
25.99182786175802*z(170)*z(1315)-25.99182786175802*z(176)*z(1322)-25.99182786175802*z(177)*z(1329)) -
0.00508017*z(494)*(z(2227)+1.575266182037215*z(2237)+1.26191682561804*z(210)*z(2217)+1.26191682561804*z(211)*z(2227)+1.261916825618
04*z(212)*z(2237)+1.738918973183969*z(207)*z(2217)+1.738918973183969*z(208)*z(2227)+1.738918973183969*z(209)*z(2237)-
1.564399813392072*z(2217)-1.445018572213135*z(213)*z(2217)-1.445018572213135*z(214)*z(2227)-1.445018572213135*z(215)*z(2237)) -
0.00404624*z(502)*(z(1014)+2.01888419866345*z(1007)+1.135663974455297*z(132)*z(1021)+1.135663974455297*z(133)*z(1014)+1.13566397445
5297*z(134)*z(1007)+1.613396387757523*z(129)*z(1021)+1.613396387757523*z(130)*z(1014)+1.613396387757523*z(131)*z(1007)-
1.85913242911938*z(1021)-1.514074795365574*z(135)*z(1021)-1.514074795365574*z(136)*z(1014)-1.514074795365574*z(137)*z(1007)) -
0.00297212*z(498)*(z(1592)+2.289867165524945*z(1599)+1.384676594484745*z(168)*z(1585)+1.384676594484745*z(174)*z(1592)+1.3846765944
84745*z(175)*z(1599)+2.471453709809833*z(166)*z(1585)+2.471453709809833*z(172)*z(1592)+2.471453709809833*z(173)*z(1599)-
2.986921120277781*z(1585)-2.589087923771584*z(170)*z(1585)-2.589087923771584*z(176)*z(1592)-2.589087923771584*z(177)*z(1599)) -
z(446)*(2.0E-
07*z(2021)+0.0210342*z(2001)+0.000948567000000000001*z(210)*z(2001)+0.000948567000000000001*z(211)*z(2011)+0.000948567000000000001*z(212
)*z(2021)-0.001343886*z(2011)-0.0196764*z(207)*z(2001)-0.0196764*z(208)*z(2011)-0.0196764*z(209)*z(2021)-0.00039917*z(213)*z(2001)-
0.00039917*z(214)*z(2011)-0.00039917*z(215)*z(2021)) -
0.000156039*z(466)*(34.63294432802056*z(1369)+4.285595267849704*z(168)*z(1369)+4.285595267849704*z(174)*z(1376)+4.285595267849704*
z(175)*z(1383)-6.601112542377225*z(1376)-z(1383)-34.22740468728971*z(166)*z(1369)-34.22740468728971*z(172)*z(1376)-
34.22740468728971*z(173)*z(1383)-5.76932689904447*z(170)*z(1369)-5.76932689904447*z(176)*z(1376)-5.76932689904447*z(177)*z(1383));
z(3334) = z(252)*z(417) + z(255)*z(427) + z(258)*z(437);
z(3347) = z(253)*z(417) + z(256)*z(427) + z(259)*z(437);
z(3360) = z(254)*z(417) + z(257)*z(427) + z(260)*z(437);
z(3410) = z(417)*z(2730) + z(427)*z(2733) + z(437)*z(2736);
z(3423) = z(417)*z(2731) + z(427)*z(2734) + z(437)*z(2737);
z(3436) = z(417)*z(2732) + z(427)*z(2735) + z(437)*z(2738);
z(3495) = z(417)*z(2868) + z(427)*z(2871) + z(437)*z(2874);
z(3508) = z(417)*z(2869) + z(427)*z(2872) + z(437)*z(2875);
z(3521) = z(417)*z(2870) + z(427)*z(2873) + z(437)*z(2876);
z(4378) = z(127)*z(3203) + z(128)*z(3202) + z(394)*z(3255) + z(401)*z(3254) + z(408)*z(3253) + z(417)*z(3326) + z(427)*z(3325) + z(437)*z(3324)
+ z(3334)*z(3402) + z(3347)*z(3401) + z(3360)*z(3400) + z(3410)*z(3487) + z(3423)*z(3486) + z(3436)*z(3485) + z(3495)*z(3569) +
z(3508)*z(3568) + z(3521)*z(3567) + z(127)*z(3200) + z(128)*z(3201) + z(394)*z(3250) + z(401)*z(3251) + z(408)*z(3252) + z(417)*z(3321) +
z(427)*z(3322) + z(437)*z(3323) + z(3334)*z(3397) + z(3347)*z(3398) + z(3360)*z(3399) + z(3410)*z(3482) + z(3423)*z(3483) + z(3436)*z(3484) +
z(3495)*z(3564) + z(3508)*z(3565) + z(3521)*z(3566) + SW*(z(2825)*z(4372)+z(2910)*z(4370)+z(2923)*z(4371)) +
0.1428571428571429*SW*(z(755)*z(3652)+z(762)*z(3653)+z(769)*z(3651)) +
0.1428571428571429*SW*(z(1207)*z(3741)+z(1214)*z(3743)+z(1221)*z(3742)) +
0.1428571428571429*SW*(z(1857)*z(3867)+z(1867)*z(3869)+z(1877)*z(3868)) +
0.1428571428571429*SW*(z(2498)*z(4020)+z(2511)*z(4022)+z(2524)*z(4021)) - z(16)*z(3204) - z(16)*z(3199) -
0.1428571428571429*SW*(z(34)*z(2659)*z(4191)+z(34)*z(2699)*z(4190)-z(2659)*z(4190)-z(2672)*z(4176)-z(2685)*z(4162)-z(2699)*z(4191)-
z(2712)*z(4193)-z(2725)*z(4192)-z(250)*z(2659)*z(4193)-z(250)*z(2712)*z(4190)-z(251)*z(2659)*z(4192)-z(251)*z(2725)*z(4190)-
z(283)*z(2685)*z(4191)-z(283)*z(2699)*z(4162)-z(284)*z(2672)*z(4191)-z(284)*z(2699)*z(4176)-z(285)*z(2685)*z(4193)-z(285)*z(2712)*z(4162)-
z(286)*z(2672)*z(4193)-z(286)*z(2712)*z(4176)-z(287)*z(2685)*z(4192)-z(287)*z(2725)*z(4162)-z(288)*z(2672)*z(4192)-z(288)*z(2725)*z(4176));
z(4402) = z(2997) - z(4378);
z(4485) = z(4473)*z(4397) - z(4402);
z(4611) = z(4485) - z(4600)*z(4433);
z(4716) = z(4611) - z(4706)*z(4575);
z(4802) = z(4716) - z(4793)*z(4694);
z(4871) = z(4802) - z(4863)*z(4792);
z(4973) = z(4925) - z(4966)*z(4871);
z(1130) = -z(385)*COCB3 - z(389)*z(1111);
z(1134) = z(742) + z(1130);
z(1123) = z(381)*COCB3 + z(389)*z(1110);
z(1138) = z(746) + z(1123);
z(1116) = z(381)*z(1111) - z(385)*z(1110);
z(1142) = z(750) + z(1116);
z(2932) = z(119)*z(1134) + z(123)*z(1138) + z(124)*z(1142);
z(4557) = z(2932)*z(4551);
z(1131) = -z(386)*COCB3 - z(390)*z(1111);
z(1135) = z(743) + z(1131);
z(1124) = z(382)*COCB3 + z(390)*z(1110);
z(1139) = z(747) + z(1124);
z(1117) = z(382)*z(1111) - z(386)*z(1110);
z(1143) = z(751) + z(1117);
z(2945) = z(119)*z(1135) + z(123)*z(1139) + z(124)*z(1143);
z(4427) = z(2945) - z(2932)*z(4421);
z(4677) = -z(4557) - z(4427)*z(4672);
z(1129) = -z(384)*COCB3 - z(388)*z(1111);
z(1133) = z(741) + z(1129);

```
z(1122) = z(380)*COCB3 + z(388)*z(1110);
z(1137) = z(745) + z(1122);
z(1115) = z(380)*z(1111) - z(384)*z(1110);
z(1141) = z(749) + z(1115);
z(2958) = z(119)*z(1133) + z(123)*z(1137) + z(124)*z(1141);
z(4440) = z(2958) - z(2932)*z(4434);
z(4569) = z(4440) - z(4427)*z(4564);
z(4776) = z(4677) - z(4569)*z(4772);
z(1126) = -z(123)*COCB3 - z(124)*z(1111);
z(1119) = z(119)*COCB3 + z(124)*z(1110);
z(1112) = z(119)*z(1111) - z(123)*z(1110);
z(2968) = z(119)*z(1126) + z(123)*z(1119) + z(124)*z(1112);
z(4453) = z(2968) - z(2932)*z(4447);
z(4581) = z(4453) - z(4427)*z(4576);
z(4688) = z(4581) - z(4569)*z(4684);
z(4856) = z(4776) - z(4688)*z(4853);
z(1127) = -z(125)*COCB3 - z(126)*z(1111);
z(1120) = z(121)*COCB3 + z(126)*z(1110);
z(1113) = z(121)*z(1111) - z(125)*z(1110);
z(2978) = z(119)*z(1127) + z(123)*z(1120) + z(124)*z(1113);
z(4466) = z(2978) - z(2932)*z(4460);
z(4593) = z(4466) - z(4427)*z(4588);
z(4699) = z(4593) - z(4569)*z(4695);
z(4786) = z(4699) - z(4688)*z(4783);
z(4919) = z(4856) - z(4786)*z(4917);
z(1128) = -z(127)*COCB3 - z(128)*z(1111);
z(1121) = z(128)*z(1110) - z(16)*COCB3;
z(1114) = -z(16)*z(1111) - z(127)*z(1110);
z(2988) = z(119)*z(1128) + z(123)*z(1121) + z(124)*z(1114);
z(4479) = z(2988) - z(2932)*z(4473);
z(4605) = z(4479) - z(4427)*z(4600);
z(4710) = z(4605) - z(4569)*z(4706);
z(4796) = z(4710) - z(4688)*z(4793);
z(4865) = z(4796) - z(4786)*z(4863);
z(4967) = z(4919) - z(4865)*z(4966);
z(2345) = -z(423)*EOED3 - z(433)*z(2315);
z(2357) = z(1773) + z(2345);
z(2332) = z(413)*EOED3 + z(433)*z(2314);
z(2367) = z(1789) + z(2332);
z(2319) = z(413)*z(2315) - z(423)*z(2314);
z(2377) = z(1805) + z(2319);
z(3001) = z(197)*z(2357) + z(201)*z(2367) + z(202)*z(2377);
z(4486) = z(3001)/z(2938);
z(4492) = z(2932)*z(4486);
z(3002) = z(199)*z(2357) + z(203)*z(2367) + z(204)*z(2377);
z(4487) = z(3002) - z(2939)*z(4486);
z(4612) = z(4487)/z(4422);
z(4617) = -z(4492) - z(4427)*z(4612);
z(3003) = z(205)*z(2367) + z(206)*z(2377) - z(28)*z(2357);
z(4488) = z(3003) - z(2940)*z(4486);
z(4613) = z(4488) - z(4423)*z(4612);
z(4717) = z(4613)/z(4565);
z(4721) = z(4617) - z(4569)*z(4717);
z(1734) = z(193)*DODC3 - z(195)*z(1713);
z(1714) = z(19)*z(1713) + z(193)*z(1712);
z(1724) = z(19)*DODC3 + z(195)*z(1712);
z(2998) = z(19)*z(1734) + z(195)*z(1714) - z(193)*z(1724);
z(4489) = z(2998) - z(2935)*z(4486);
z(4614) = z(4489) - z(4424)*z(4612);
z(4718) = z(4614) - z(4566)*z(4717);
z(4803) = z(4718)/z(4685);
z(4806) = z(4721) - z(4688)*z(4803);
z(2999) = z(20)*z(1734) + z(194)*z(1724) - z(196)*z(1714);
z(4490) = z(2999) - z(2936)*z(4486);
z(4615) = z(4490) - z(4425)*z(4612);
z(4719) = z(4615) - z(4567)*z(4717);
z(4804) = z(4719) - z(4686)*z(4803);
z(4872) = z(4804)/z(4784);
z(4874) = z(4806) - z(4786)*z(4872);
z(3000) = z(23)*z(1714) + z(24)*z(1724);
z(4491) = z(3000) - z(2937)*z(4486);
z(4616) = z(4491) - z(4426)*z(4612);
z(4720) = z(4616) - z(4568)*z(4717);
z(4805) = z(4720) - z(4687)*z(4803);
z(4873) = z(4805) - z(4785)*z(4872);
z(4926) = z(4873)/z(4864);
z(4927) = z(4874) - z(4865)*z(4926);
z(5002) = z(4967)/z(4927);
```

z(2032) = z(413)*z(2027) - z(423)*z(2026);
z(2090) = z(1805) + z(2032);
z(2045) = z(433)*z(2026) - z(413)*z(2028);
z(2080) = z(1789) + z(2045);
z(2058) = z(423)*z(2028) - z(433)*z(2027);
z(2070) = z(1773) + z(2058);
z(2104) = z(413)*z(2099) - z(423)*z(2098);
z(2162) = z(1805) + z(2104);
z(2130) = z(423)*z(2100) - z(433)*z(2099);
z(2142) = z(1773) + z(2130);
z(2117) = z(433)*z(2098) - z(413)*z(2100);
z(2152) = z(1789) + z(2117);
z(1391) = z(19)*z(1389) + z(193)*z(1388);
z(1413) = -z(193)*z(1390) - z(195)*z(1389);
z(1401) = z(195)*z(1388) - z(19)*z(1390);
z(1661) = z(19)*z(1659) + z(193)*z(1658);
z(1683) = -z(193)*z(1660) - z(195)*z(1659);
z(1671) = z(195)*z(1658) - z(19)*z(1660);
z(2248) = z(413)*z(2243) - z(423)*z(2242);
z(2306) = z(1805) + z(2248);
z(2274) = z(423)*z(2244) - z(433)*z(2243);
z(2286) = z(1773) + z(2274);
z(2261) = z(433)*z(2242) - z(413)*z(2244);
z(2296) = z(1789) + z(2261);
z(1283) = z(19)*z(1281) + z(193)*z(1280);
z(1305) = -z(193)*z(1282) - z(195)*z(1281);
z(1293) = z(195)*z(1280) - z(19)*z(1282);
z(2189) = z(433)*z(2170) - z(413)*z(2172);
z(2224) = z(1789) + z(2189);
z(2176) = z(413)*z(2171) - z(423)*z(2170);
z(2234) = z(1805) + z(2176);
z(2202) = z(423)*z(2172) - z(433)*z(2171);
z(2214) = z(1773) + z(2202);
z(1563) = z(195)*z(1550) - z(19)*z(1552);
z(1553) = z(19)*z(1551) + z(193)*z(1550);
z(1575) = -z(193)*z(1552) - z(195)*z(1551);
z(1960) = z(413)*z(1955) - z(423)*z(1954);
z(2018) = z(1805) + z(1960);
z(1986) = z(423)*z(1956) - z(433)*z(1955);
z(1998) = z(1773) + z(1986);
z(1973) = z(433)*z(1954) - z(413)*z(1956);
z(2008) = z(1789) + z(1973);
z(1359) = -z(193)*z(1336) - z(195)*z(1335);
z(1347) = z(195)*z(1334) - z(19)*z(1336);
z(1337) = z(19)*z(1335) + z(193)*z(1334);
z(3004) = 4.7299E-
05*z(462)*(z(2090)+44.21104040254551*z(2080)+114.2537897207129*z(207)*z(2070)+114.2537897207129*z(208)*z(2080)+114.2537897207129*z(
209)*z(2090)-147.8697224042791*z(2070)-21.77701431319901*z(210)*z(2070)-21.77701431319901*z(211)*z(2080)-
21.77701431319901*z(212)*z(2090)-3.298991522019493*z(213)*z(2070)-3.298991522019493*z(214)*z(2080)-3.298991522019493*z(215)*z(2090))
+ 0.000263142*z(478)*(z(2162)+57.96163288262611*z(2142)-37.04060925279887*z(2152)-58.17953804409786*z(207)*z(2142)-
58.17953804409786*z(208)*z(2152)-58.17953804409786*z(209)*z(2162)-3.892632114979745*z(213)*z(2142)-3.892632114979745*z(214)*z(2152)-
3.892632114979745*z(215)*z(2162)-3.889998555912777*z(210)*z(2142)-3.889998555912777*z(211)*z(2152)-3.889998555912777*z(212)*z(2162))
+ 6.128E-05*z(482)*(19.44327676240209*z(1391)+238.8415469973891*z(1413)-120.5957898172324*z(1401)-232.8573759791123*z(166)*z(1413)-
232.8573759791123*z(172)*z(1401)-232.8573759791123*z(173)*z(1391)-1.221507832898173*z(168)*z(1413)-1.221507832898173*z(174)*z(1401)-
1.221507832898173*z(175)*z(1391)-z(170)*z(1413)-z(176)*z(1401)-z(177)*z(1391)) +
0.00334198*z(514)*(2.523796072986673*z(1661)+2.541448183412228*z(1683)-z(1671)-2.199101730112089*z(166)*z(1683)-
2.199101730112089*z(172)*z(1671)-2.199101730112089*z(173)*z(1661)-2.178312856450368*z(170)*z(1683)-2.178312856450368*z(176)*z(1671)-
2.178312856450368*z(177)*z(1661)-1.513417495017923*z(168)*z(1683)-1.513417495017923*z(174)*z(1671)-1.513417495017923*z(175)*z(1661))
+ 0.005250594000000001*z(510)*(1.3792740402324*z(2306)+1.576289654084852*z(2286)-1.220873295478569*z(2296)-
1.749405305380686*z(207)*z(2286)-1.749405305380686*z(208)*z(2296)-1.749405305380686*z(209)*z(2306)-1.677448684853561*z(213)*z(2286)-
1.677448684853561*z(214)*z(2296)-1.677448684853561*z(215)*z(2306)-z(210)*z(2286)-z(211)*z(2296)-z(212)*z(2306)) -
9.800000000000001*SW*(z(2790)*z(2906)+z(2791)*z(2821)+z(2792)*z(2919)) - 1.4*SW*(z(199)*z(1854)+z(203)*z(1864)+z(204)*z(1874)) -
1.4*SW*(z(284)*z(2495)+z(286)*z(2508)+z(288)*z(2521)) - 1.4*SW*(z(20)*z(1195)+z(194)*z(1185)-z(196)*z(1175)) -
1.4*SW*(z(2668)+z(284)*z(2696)+z(286)*z(2709)+z(288)*z(2722)) - 4.9681E-
05*z(450)*(8.034661138060828*z(1283)+396.0548298142147*z(1305)+z(168)*z(1305)+z(174)*z(1293)+z(175)*z(1283)-
19.09315432459089*z(1293)-410.7868199110324*z(166)*z(1305)-410.7868199110324*z(172)*z(1293)-410.7868199110324*z(173)*z(1283)-
25.99182786175802*z(170)*z(1305)-25.99182786175802*z(176)*z(1293)-25.99182786175802*z(177)*z(1283)) -
0.00508017*z(494)*(z(2224)+1.575266182037215*z(2234)+1.26191682561804*z(210)*z(2214)+1.26191682561804*z(211)*z(2224)+1.261916825618
04*z(212)*z(2234)+1.738918973183969*z(207)*z(2214)+1.738918973183969*z(208)*z(2224)+1.738918973183969*z(209)*z(2234)-
1.564399813392072*z(2214)-1.445018572213135*z(213)*z(2214)-1.445018572213135*z(214)*z(2224)-1.445018572213135*z(215)*z(2234)) -
0.00297212*z(498)*(z(1563)+2.289867165524945*z(1553)+1.384676594484745*z(168)*z(1575)+1.384676594484745*z(174)*z(1563)+1.3846765944
84745*z(175)*z(1553)+2.471453709809833*z(166)*z(1575)+2.471453709809833*z(172)*z(1563)+2.471453709809833*z(173)*z(1553)-
2.986921120277781*z(1575)-2.589087923771584*z(170)*z(1575)-2.589087923771584*z(176)*z(1563)-2.589087923771584*z(177)*z(1553)) -
z(446)*(2.0E-
07*z(2018)+0.0210342*z(1998)+0.000948567000000000001*z(210)*z(1998)+0.000948567000000000001*z(211)*z(2008)+0.000948567000000000001*z(212
)*z(2018)-0.001343886*z(2008)-0.0196764*z(207)*z(1998)-0.0196764*z(208)*z(2008)-0.0196764*z(209)*z(2018)-0.00039917*z(213)*z(1998)-
0.00039917*z(214)*z(2008)-0.00039917*z(215)*z(2018)) -
0.000156039*z(466)*(34.63294432802056*z(1359)+4.285595267849704*z(168)*z(1359)+4.285595267849704*z(174)*z(1347)+4.285595267849704*

z(175)*z(1337)-6.601112542377225*z(1347)-z(1337)-34.22740468728971*z(166)*z(1359)-34.22740468728971*z(172)*z(1347)-
34.22740468728971*z(173)*z(1337)-5.76932689904447*z(170)*z(1359)-5.76932689904447*z(176)*z(1347)-5.76932689904447*z(177)*z(1337));
z(3330) = z(252)*z(413) + z(255)*z(423) + z(258)*z(433);
z(3343) = z(253)*z(413) + z(256)*z(423) + z(259)*z(433);
z(3356) = z(254)*z(413) + z(257)*z(423) + z(260)*z(433);
z(3406) = z(413)*z(2730) + z(423)*z(2733) + z(433)*z(2736);
z(3419) = z(413)*z(2731) + z(423)*z(2734) + z(433)*z(2737);
z(3432) = z(413)*z(2732) + z(423)*z(2735) + z(433)*z(2738);
z(3491) = z(413)*z(2868) + z(423)*z(2871) + z(433)*z(2874);
z(3504) = z(413)*z(2869) + z(423)*z(2872) + z(433)*z(2875);
z(3517) = z(413)*z(2870) + z(423)*z(2873) + z(433)*z(2876);
z(4379) = z(19)*z(3255) + z(195)*z(3253) + z(413)*z(3326) + z(423)*z(3325) + z(433)*z(3324) + z(3330)*z(3402) + z(3343)*z(3401) +
z(3356)*z(3400) + z(3406)*z(3487) + z(3419)*z(3486) + z(3432)*z(3485) + z(3491)*z(3569) + z(3504)*z(3568) + z(3517)*z(3567) + z(19)*z(3250) +
z(195)*z(3252) + z(413)*z(3321) + z(423)*z(3322) + z(433)*z(3323) + z(3330)*z(3397) + z(3343)*z(3398) + z(3356)*z(3399) + z(3406)*z(3482) +
z(3419)*z(3483) + z(3432)*z(3484) + z(3491)*z(3564) + z(3504)*z(3565) + z(3517)*z(3566) +
SW*(z(2821)*z(4372)+z(2906)*z(4370)+z(2919)*z(4371)) + 0.1428571428571429*SW*(z(1175)*z(3742)+z(1185)*z(3743)+z(1195)*z(3741)) +
0.1428571428571429*SW*(z(1854)*z(3867)+z(1864)*z(3869)+z(1874)*z(3868)) +
0.1428571428571429*SW*(z(2495)*z(4020)+z(2508)*z(4022)+z(2521)*z(4021)) - z(193)*z(3254) - z(193)*z(3251) -
0.1428571428571429*SW*(z(34)*z(2655)*z(4191)+z(34)*z(2696)*z(4190)-z(2655)*z(4190)-z(2668)*z(4176)-z(2681)*z(4162)-z(2696)*z(4191)-
z(2709)*z(4193)-z(2722)*z(4192)-z(250)*z(2655)*z(4193)-z(250)*z(2709)*z(4190)-z(251)*z(2655)*z(4192)-z(251)*z(2722)*z(4190)-
z(283)*z(2681)*z(4191)-z(283)*z(2696)*z(4162)-z(284)*z(2668)*z(4191)-z(284)*z(2696)*z(4176)-z(285)*z(2681)*z(4193)-z(285)*z(2709)*z(4162)-
z(286)*z(2668)*z(4193)-z(286)*z(2709)*z(4176)-z(287)*z(2681)*z(4192)-z(287)*z(2722)*z(4162)-z(288)*z(2668)*z(4192)-z(288)*z(2722)*z(4176));
z(4403) = z(3004) - z(4379);
z(4498) = z(4486)*z(4397) - z(4403);
z(4623) = z(4498) - z(4612)*z(4433);
z(4727) = z(4623) - z(4717)*z(4575);
z(4812) = z(4727) - z(4803)*z(4694);
z(4880) = z(4812) - z(4872)*z(4792);
z(4933) = z(4880) - z(4926)*z(4871);
z(5008) = z(4973) - z(5002)*z(4933);
z(2933) = z(121)*z(1134) + z(125)*z(1138) + z(126)*z(1142);
z(4558) = z(2933)*z(4551);
z(2946) = z(121)*z(1135) + z(125)*z(1139) + z(126)*z(1143);
z(4428) = z(2946) - z(2933)*z(4421);
z(4678) = -z(4558) - z(4428)*z(4672);
z(2959) = z(121)*z(1133) + z(125)*z(1137) + z(126)*z(1141);
z(4441) = z(2959) - z(2933)*z(4434);
z(4570) = z(4441) - z(4428)*z(4564);
z(4777) = z(4678) - z(4570)*z(4772);
z(2969) = z(121)*z(1126) + z(125)*z(1119) + z(126)*z(1112);
z(4454) = z(2969) - z(2933)*z(4447);
z(4582) = z(4454) - z(4428)*z(4576);
z(4689) = z(4582) - z(4570)*z(4684);
z(4857) = z(4777) - z(4689)*z(4853);
z(2979) = z(121)*z(1127) + z(125)*z(1120) + z(126)*z(1113);
z(4467) = z(2979) - z(2933)*z(4460);
z(4594) = z(4467) - z(4428)*z(4588);
z(4700) = z(4594) - z(4570)*z(4695);
z(4787) = z(4700) - z(4689)*z(4783);
z(4920) = z(4857) - z(4787)*z(4917);
z(2989) = z(121)*z(1128) + z(125)*z(1121) + z(126)*z(1114);
z(4480) = z(2989) - z(2933)*z(4473);
z(4606) = z(4480) - z(4428)*z(4600);
z(4711) = z(4606) - z(4570)*z(4706);
z(4797) = z(4711) - z(4689)*z(4793);
z(4866) = z(4797) - z(4787)*z(4863);
z(4968) = z(4920) - z(4866)*z(4966);
z(4493) = z(2933)*z(4486);
z(4618) = -z(4493) - z(4428)*z(4612);
z(4722) = z(4618) - z(4570)*z(4717);
z(4807) = z(4722) - z(4689)*z(4803);
z(4875) = z(4807) - z(4787)*z(4872);
z(4928) = z(4875) - z(4866)*z(4926);
z(5003) = z(4968) - z(4928)*z(5002);
z(2346) = -z(424)*EOED3 - z(434)*z(2315);
z(2355) = z(1771) + z(2346);
z(2333) = z(414)*EOED3 + z(434)*z(2314);
z(2365) = z(1787) + z(2333);
z(2320) = z(414)*z(2315) - z(424)*z(2314);
z(2375) = z(1803) + z(2320);
z(3008) = z(197)*z(2355) + z(201)*z(2365) + z(202)*z(2375);
z(4499) = z(3008)/z(2938);
z(4506) = z(2933)*z(4499);
z(3009) = z(199)*z(2355) + z(203)*z(2365) + z(204)*z(2375);
z(4500) = z(3009) - z(2939)*z(4499);
z(4624) = z(4500)/z(4422);
z(4630) = -z(4506) - z(4428)*z(4624);
z(3010) = z(205)*z(2365) + z(206)*z(2375) - z(28)*z(2355);

```
z(4501) = z(3010) - z(2940)*z(4499);
z(4625) = z(4501) - z(4423)*z(4624);
z(4728) = z(4625)/z(4565);
z(4733) = z(4630) - z(4570)*z(4728);
z(1736) = z(196)*z(1713) - z(194)*DODC3;
z(1715) = z(20)*z(1713) - z(194)*z(1712);
z(1725) = z(20)*DODC3 - z(196)*z(1712);
z(3005) = z(19)*z(1736) + z(195)*z(1715) - z(193)*z(1725);
z(4502) = z(3005) - z(2935)*z(4499);
z(4626) = z(4502) - z(4424)*z(4624);
z(4729) = z(4626) - z(4566)*z(4728);
z(4813) = z(4729)/z(4685);
z(4817) = z(4733) - z(4689)*z(4813);
z(3006) = z(20)*z(1736) + z(194)*z(1725) - z(196)*z(1715);
z(4503) = z(3006) - z(2936)*z(4499);
z(4627) = z(4503) - z(4425)*z(4624);
z(4730) = z(4627) - z(4567)*z(4728);
z(4814) = z(4730) - z(4686)*z(4813);
z(4881) = z(4814)/z(4784);
z(4884) = z(4817) - z(4787)*z(4881);
z(3007) = z(23)*z(1715) + z(24)*z(1725);
z(4504) = z(3007) - z(2937)*z(4499);
z(4628) = z(4504) - z(4426)*z(4624);
z(4731) = z(4628) - z(4568)*z(4728);
z(4815) = z(4731) - z(4687)*z(4813);
z(4882) = z(4815) - z(4785)*z(4881);
z(4934) = z(4882)/z(4864);
z(4936) = z(4884) - z(4866)*z(4934);
z(4505) = z(2932)*z(4499);
z(4629) = -z(4505) - z(4427)*z(4624);
z(4732) = z(4629) - z(4569)*z(4728);
z(4816) = z(4732) - z(4688)*z(4813);
z(4883) = z(4816) - z(4786)*z(4881);
z(4935) = z(4883) - z(4865)*z(4934);
z(4974) = z(4935)/z(4927);
z(4975) = z(4936) - z(4928)*z(4974);
z(5027) = z(5003)/z(4975);
z(2033) = z(414)*z(2027) - z(424)*z(2026);
z(2088) = z(1803) + z(2033);
z(2046) = z(434)*z(2026) - z(414)*z(2028);
z(2078) = z(1787) + z(2046);
z(2059) = z(424)*z(2028) - z(434)*z(2027);
z(2068) = z(1771) + z(2059);
z(2105) = z(414)*z(2099) - z(424)*z(2098);
z(2160) = z(1803) + z(2105);
z(2131) = z(424)*z(2100) - z(434)*z(2099);
z(2140) = z(1771) + z(2131);
z(2118) = z(434)*z(2098) - z(414)*z(2100);
z(2150) = z(1787) + z(2118);
z(1392) = z(20)*z(1389) - z(194)*z(1388);
z(1411) = z(194)*z(1390) + z(196)*z(1389);
z(1402) = -z(20)*z(1390) - z(196)*z(1388);
z(1662) = z(20)*z(1659) - z(194)*z(1658);
z(1681) = z(194)*z(1660) + z(196)*z(1659);
z(1672) = -z(20)*z(1660) - z(196)*z(1658);
z(2249) = z(414)*z(2243) - z(424)*z(2242);
z(2304) = z(1803) + z(2249);
z(2275) = z(424)*z(2244) - z(434)*z(2243);
z(2284) = z(1771) + z(2275);
z(2262) = z(434)*z(2242) - z(414)*z(2244);
z(2294) = z(1787) + z(2262);
z(1284) = z(20)*z(1281) - z(194)*z(1280);
z(1303) = z(194)*z(1282) + z(196)*z(1281);
z(1294) = -z(20)*z(1282) - z(196)*z(1280);
z(2190) = z(434)*z(2170) - z(414)*z(2172);
z(2222) = z(1787) + z(2190);
z(2177) = z(414)*z(2171) - z(424)*z(2170);
z(2232) = z(1803) + z(2177);
z(2203) = z(424)*z(2172) - z(434)*z(2171);
z(2212) = z(1771) + z(2203);
z(1564) = -z(20)*z(1552) - z(196)*z(1550);
z(1554) = z(20)*z(1551) - z(194)*z(1550);
z(1573) = z(194)*z(1552) + z(196)*z(1551);
z(1961) = z(414)*z(1955) - z(424)*z(1954);
z(2016) = z(1803) + z(1961);
z(1987) = z(424)*z(1956) - z(434)*z(1955);
z(1996) = z(1771) + z(1987);
z(1974) = z(434)*z(1954) - z(414)*z(1956);
```

z(2006) = z(1787) + z(1974);
z(1357) = z(194)*z(1336) + z(196)*z(1335);
z(1348) = -z(20)*z(1336) - z(196)*z(1334);
z(1338) = z(20)*z(1335) - z(194)*z(1334);
z(3011) = 4.7299E-
05*z(462)*(z(2088)+44.21104040254551*z(2078)+114.2537897207129*z(207)*z(2068)+114.2537897207129*z(208)*z(2078)+114.2537897207129*z(
209)*z(2088)-147.8697224042791*z(2068)-21.77701431319901*z(210)*z(2068)-21.77701431319901*z(211)*z(2078)-
21.77701431319901*z(212)*z(2088)-3.298991522019493*z(213)*z(2068)-3.298991522019493*z(214)*z(2078)-3.298991522019493*z(215)*z(2088))
+ 0.000263142*z(478)*(z(2160)+57.96163288262611*z(2140)-37.04060925279887*z(2150)-58.17953804409786*z(207)*z(2140)-
58.17953804409786*z(208)*z(2150)-58.17953804409786*z(209)*z(2160)-3.892632114979745*z(213)*z(2140)-3.892632114979745*z(214)*z(2150)-
3.892632114979745*z(215)*z(2160)-3.889998555912777*z(210)*z(2140)-3.889998555912777*z(211)*z(2150)-3.889998555912777*z(212)*z(2160))
+ 6.128E-05*z(482)*(19.44327676240209*z(1392)+238.8415469973891*z(1411)-120.5957898172324*z(1402)-232.8573759791123*z(166)*z(1411)-
232.8573759791123*z(172)*z(1402)-232.8573759791123*z(173)*z(1392)-1.221507832898173*z(168)*z(1411)-1.221507832898173*z(174)*z(1402)-
1.221507832898173*z(175)*z(1392)-z(170)*z(1411)-z(176)*z(1402)-z(177)*z(1392)) +
0.00334198*z(514)*(2.523796072986673*z(1662)+2.541448183412228*z(1681)-z(1672)-2.199101730112089*z(166)*z(1681)-
2.199101730112089*z(172)*z(1672)-2.199101730112089*z(173)*z(1662)-2.178312856450368*z(170)*z(1681)-2.178312856450368*z(176)*z(1672)-
2.178312856450368*z(177)*z(1662)-1.513417495017923*z(168)*z(1681)-1.513417495017923*z(174)*z(1672)-1.513417495017923*z(175)*z(1662))
+ 0.005250594000000001*z(510)*(1.3792740402324*z(2304)+1.576289654084852*z(2284)-1.220873295478569*z(2294)-
1.749405305380686*z(207)*z(2284)-1.749405305380686*z(208)*z(2294)-1.749405305380686*z(209)*z(2304)-1.677448684853561*z(213)*z(2284)-
1.677448684853561*z(214)*z(2294)-1.677448684853561*z(215)*z(2304)-z(210)*z(2284)-z(211)*z(2294)-z(212)*z(2304)) -
9.800000000000001*SW*(z(2790)*z(2907)+z(2791)*z(2822)+z(2792)*z(2920)) - 1.4*SW*(z(199)*z(1852)+z(203)*z(1862)+z(204)*z(1872)) -
1.4*SW*(z(284)*z(2493)+z(286)*z(2506)+z(288)*z(2519)) - 1.4*SW*(z(20)*z(1197)+z(194)*z(1186)-z(196)*z(1176)) -
1.4*SW*(z(2669)+z(284)*z(2694)+z(286)*z(2707)+z(288)*z(2720)) - 4.9681E-
05*z(450)*(8.034661138060828*z(1284)+396.0548298142147*z(1303)+z(168)*z(1303)+z(174)*z(1294)+z(175)*z(1284)-
19.09315432459089*z(1294)-410.7868199110324*z(166)*z(1303)-410.7868199110324*z(172)*z(1294)-410.7868199110324*z(173)*z(1284)-
25.99182786175802*z(170)*z(1303)-25.99182786175802*z(176)*z(1294)-25.99182786175802*z(177)*z(1284)) -
0.00508017*z(494)*(z(2222)+1.575266182037215*z(2232)+1.26191682561804*z(210)*z(2212)+1.26191682561804*z(211)*z(2222)+1.261916825618
04*z(212)*z(2232)+1.738918973183969*z(207)*z(2212)+1.738918973183969*z(208)*z(2222)+1.738918973183969*z(209)*z(2232)-
1.564399813392072*z(2212)-1.445018572213135*z(213)*z(2212)-1.445018572213135*z(214)*z(2222)-1.445018572213135*z(215)*z(2232)) -
0.00297212*z(498)*(z(1564)+2.289867165524945*z(1554)+1.384676594484745*z(168)*z(1573)+1.384676594484745*z(174)*z(1564)+1.3846765944
84745*z(175)*z(1554)+2.471453709809833*z(166)*z(1573)+2.471453709809833*z(172)*z(1564)+2.471453709809833*z(173)*z(1554)-
2.986921120277781*z(1573)-2.589087923771584*z(170)*z(1573)-2.589087923771584*z(176)*z(1564)-2.589087923771584*z(177)*z(1554)) -
z(446)*(2.0E-
07*z(2016)+0.0210342*z(1996)+0.000948567000000000001*z(210)*z(1996)+0.000948567000000000001*z(211)*z(2006)+0.000948567000000000001*z(212
)*z(2016)-0.001343886*z(2006)-0.0196764*z(207)*z(1996)-0.0196764*z(208)*z(2006)-0.0196764*z(209)*z(2016)-0.00039917*z(213)*z(1996)-
0.00039917*z(214)*z(2006)-0.00039917*z(215)*z(2016)) -
0.000156039*z(466)*(34.63294432802056*z(1357)+4.285595267849704*z(168)*z(1357)+4.285595267849704*z(174)*z(1348)+4.285595267849704*
z(175)*z(1338)-6.601112542377225*z(1348)-z(1338)-34.22740468728971*z(166)*z(1357)-34.22740468728971*z(172)*z(1348)-
34.22740468728971*z(173)*z(1338)-5.76932689904447*z(170)*z(1357)-5.76932689904447*z(176)*z(1348)-5.76932689904447*z(177)*z(1338));
z(3331) = z(252)*z(414) + z(255)*z(424) + z(258)*z(434);
z(3344) = z(253)*z(414) + z(256)*z(424) + z(259)*z(434);
z(3357) = z(254)*z(414) + z(257)*z(424) + z(260)*z(434);
z(3407) = z(414)*z(2730) + z(424)*z(2733) + z(434)*z(2736);
z(3420) = z(414)*z(2731) + z(424)*z(2734) + z(434)*z(2737);
z(3433) = z(414)*z(2732) + z(424)*z(2735) + z(434)*z(2738);
z(3492) = z(414)*z(2868) + z(424)*z(2871) + z(434)*z(2874);
z(3505) = z(414)*z(2869) + z(424)*z(2872) + z(434)*z(2875);
z(3518) = z(414)*z(2870) + z(424)*z(2873) + z(434)*z(2876);
z(4380) = z(20)*z(3255) + z(194)*z(3254) + z(414)*z(3326) + z(424)*z(3325) + z(434)*z(3324) + z(3331)*z(3402) + z(3344)*z(3401) +
z(3357)*z(3400) + z(3407)*z(3487) + z(3420)*z(3486) + z(3433)*z(3485) + z(3492)*z(3569) + z(3505)*z(3568) + z(3518)*z(3567) + z(20)*z(3250) +
z(194)*z(3251) + z(414)*z(3321) + z(424)*z(3322) + z(434)*z(3323) + z(3331)*z(3397) + z(3344)*z(3398) + z(3357)*z(3399) + z(3407)*z(3482) +
z(3420)*z(3483) + z(3433)*z(3484) + z(3492)*z(3564) + z(3505)*z(3565) + z(3518)*z(3566) +
SW*(z(2822)*z(4372)+z(2907)*z(4370)+z(2920)*z(4371)) + 0.1428571428571429*SW*(z(1176)*z(3742)+z(1186)*z(3743)+z(1197)*z(3741)) +
0.1428571428571429*SW*(z(1852)*z(3867)+z(1862)*z(3869)+z(1872)*z(3868)) +
0.1428571428571429*SW*(z(2493)*z(4020)+z(2506)*z(4022)+z(2519)*z(4021)) - z(196)*z(3253) - z(196)*z(3252) -
0.1428571428571429*SW*(z(34)*z(2656)*z(4191)+z(34)*z(2694)*z(4190)-z(2656)*z(4190)-z(2669)*z(4176)-z(2682)*z(4162)-z(2694)*z(4191)-
z(2707)*z(4193)-z(2720)*z(4192)-z(250)*z(2656)*z(4193)-z(250)*z(2707)*z(4190)-z(251)*z(2656)*z(4192)-z(251)*z(2720)*z(4190)-
z(283)*z(2682)*z(4191)-z(283)*z(2694)*z(4162)-z(284)*z(2669)*z(4191)-z(284)*z(2694)*z(4176)-z(285)*z(2682)*z(4193)-z(285)*z(2707)*z(4162)-
z(286)*z(2669)*z(4193)-z(286)*z(2707)*z(4176)-z(287)*z(2682)*z(4192)-z(287)*z(2720)*z(4162)-z(288)*z(2669)*z(4192)-z(288)*z(2720)*z(4176));
z(4404) = z(3011) - z(4380);
z(4511) = z(4499)*z(4397) - z(4404);
z(4635) = z(4511) - z(4624)*z(4433);
z(4738) = z(4635) - z(4728)*z(4575);
z(4822) = z(4738) - z(4813)*z(4694);
z(4889) = z(4822) - z(4881)*z(4792);
z(4941) = z(4889) - z(4934)*z(4871);
z(4980) = z(4941) - z(4974)*z(4933);
z(5032) = z(5008) - z(5027)*z(4980);
z(2934) = z(127)*z(1138) + z(128)*z(1142) - z(16)*z(1134);
z(4559) = z(2934)*z(4551);
z(2947) = z(127)*z(1139) + z(128)*z(1143) - z(16)*z(1135);
z(4429) = z(2947) - z(2934)*z(4421);
z(4679) = -z(4559) - z(4429)*z(4672);
z(2960) = z(127)*z(1137) + z(128)*z(1141) - z(16)*z(1133);
z(4442) = z(2960) - z(2934)*z(4434);
z(4571) = z(4442) - z(4429)*z(4564);
z(4778) = z(4679) - z(4571)*z(4772);
z(2970) = z(127)*z(1119) + z(128)*z(1112) - z(16)*z(1126);

```
z(4455) = z(2970) - z(2934)*z(4447);
z(4583) = z(4455) - z(4429)*z(4576);
z(4690) = z(4583) - z(4571)*z(4684);
z(4858) = z(4778) - z(4690)*z(4853);
z(2980) = z(127)*z(1120) + z(128)*z(1113) - z(16)*z(1127);
z(4468) = z(2980) - z(2934)*z(4460);
z(4595) = z(4468) - z(4429)*z(4588);
z(4701) = z(4595) - z(4571)*z(4695);
z(4788) = z(4701) - z(4690)*z(4783);
z(4921) = z(4858) - z(4788)*z(4917);
z(2990) = z(127)*z(1121) + z(128)*z(1114) - z(16)*z(1128);
z(4481) = z(2990) - z(2934)*z(4473);
z(4607) = z(4481) - z(4429)*z(4600);
z(4712) = z(4607) - z(4571)*z(4706);
z(4798) = z(4712) - z(4690)*z(4793);
z(4867) = z(4798) - z(4788)*z(4863);
z(4969) = z(4921) - z(4867)*z(4966);
z(4494) = z(2934)*z(4486);
z(4619) = -z(4494) - z(4429)*z(4612);
z(4723) = z(4619) - z(4571)*z(4717);
z(4808) = z(4723) - z(4690)*z(4803);
z(4876) = z(4808) - z(4788)*z(4872);
z(4929) = z(4876) - z(4867)*z(4926);
z(5004) = z(4969) - z(4929)*z(5002);
z(4507) = z(2934)*z(4499);
z(4631) = -z(4507) - z(4429)*z(4624);
z(4734) = z(4631) - z(4571)*z(4728);
z(4818) = z(4734) - z(4690)*z(4813);
z(4885) = z(4818) - z(4788)*z(4881);
z(4937) = z(4885) - z(4867)*z(4934);
z(4976) = z(4937) - z(4929)*z(4974);
z(5028) = z(5004) - z(4976)*z(5027);
z(2353) = -z(431)*EOED3 - z(441)*z(2315);
z(2356) = z(1772) + z(2353);
z(2340) = z(421)*EOED3 + z(441)*z(2314);
z(2366) = z(1788) + z(2340);
z(2327) = z(421)*z(2315) - z(431)*z(2314);
z(2376) = z(1804) + z(2327);
z(3015) = z(197)*z(2356) + z(201)*z(2366) + z(202)*z(2376);
z(4512) = z(3015)/z(2938);
z(4520) = z(2934)*z(4512);
z(3016) = z(199)*z(2356) + z(203)*z(2366) + z(204)*z(2376);
z(4513) = z(3016) - z(2939)*z(4512);
z(4636) = z(4513)/z(4422);
z(4643) = -z(4520) - z(4429)*z(4636);
z(3017) = z(205)*z(2366) + z(206)*z(2376) - z(28)*z(2356);
z(4514) = z(3017) - z(2940)*z(4512);
z(4637) = z(4514) - z(4423)*z(4636);
z(4739) = z(4637)/z(4565);
z(4745) = z(4643) - z(4571)*z(4739);
z(1735) = -z(24)*DODC3 - z(23)*z(1713);
z(1732) = z(23)*z(1712);
z(1722) = z(24)*z(1712);
z(3012) = z(19)*z(1735) - z(193)*z(1732) - z(195)*z(1722);
z(4515) = z(3012) - z(2935)*z(4512);
z(4638) = z(4515) - z(4424)*z(4636);
z(4740) = z(4638) - z(4566)*z(4739);
z(4823) = z(4740)/z(4685);
z(4828) = z(4745) - z(4690)*z(4823);
z(3013) = z(20)*z(1735) + z(194)*z(1732) + z(196)*z(1722);
z(4516) = z(3013) - z(2936)*z(4512);
z(4639) = z(4516) - z(4425)*z(4636);
z(4741) = z(4639) - z(4567)*z(4739);
z(4824) = z(4741) - z(4686)*z(4823);
z(4890) = z(4824)/z(4784);
z(4894) = z(4828) - z(4788)*z(4890);
z(3014) = z(24)*z(1732) - z(23)*z(1722);
z(4517) = z(3014) - z(2937)*z(4512);
z(4640) = z(4517) - z(4426)*z(4636);
z(4742) = z(4640) - z(4568)*z(4739);
z(4825) = z(4742) - z(4687)*z(4823);
z(4891) = z(4825) - z(4785)*z(4890);
z(4942) = z(4891)/z(4864);
z(4945) = z(4894) - z(4867)*z(4942);
z(4518) = z(2932)*z(4512);
z(4641) = -z(4518) - z(4427)*z(4636);
z(4743) = z(4641) - z(4569)*z(4739);
z(4826) = z(4743) - z(4688)*z(4823);
```

z(4892) = z(4826) - z(4786)*z(4890);
z(4943) = z(4892) - z(4865)*z(4942);
z(4981) = z(4943)/z(4927);
z(4983) = z(4945) - z(4929)*z(4981);
z(4519) = z(2933)*z(4512);
z(4642) = -z(4519) - z(4428)*z(4636);
z(4744) = z(4642) - z(4570)*z(4739);
z(4827) = z(4744) - z(4689)*z(4823);
z(4893) = z(4827) - z(4787)*z(4890);
z(4944) = z(4893) - z(4866)*z(4942);
z(4982) = z(4944) - z(4928)*z(4981);
z(5009) = z(4982)/z(4975);
z(5010) = z(4983) - z(4976)*z(5009);
z(5043) = z(5028)/z(5010);
z(2040) = z(421)*z(2027) - z(431)*z(2026);
z(2089) = z(1804) + z(2040);
z(2053) = z(441)*z(2026) - z(421)*z(2028);
z(2079) = z(1788) + z(2053);
z(2066) = z(431)*z(2028) - z(441)*z(2027);
z(2069) = z(1772) + z(2066);
z(1291) = z(24)*z(1280);
z(1301) = z(23)*z(1280);
z(1304) = z(24)*z(1282) - z(23)*z(1281);
z(2112) = z(421)*z(2099) - z(431)*z(2098);
z(2161) = z(1804) + z(2112);
z(2138) = z(431)*z(2100) - z(441)*z(2099);
z(2141) = z(1772) + z(2138);
z(2125) = z(441)*z(2098) - z(421)*z(2100);
z(2151) = z(1788) + z(2125);
z(1355) = z(23)*z(1334);
z(1345) = z(24)*z(1334);
z(1358) = z(24)*z(1336) - z(23)*z(1335);
z(1561) = z(24)*z(1550);
z(1574) = z(24)*z(1552) - z(23)*z(1551);
z(1571) = z(23)*z(1550);
z(2256) = z(421)*z(2243) - z(431)*z(2242);
z(2305) = z(1804) + z(2256);
z(2282) = z(431)*z(2244) - z(441)*z(2243);
z(2285) = z(1772) + z(2282);
z(2269) = z(441)*z(2242) - z(421)*z(2244);
z(2295) = z(1788) + z(2269);
z(1399) = z(24)*z(1388);
z(1409) = z(23)*z(1388);
z(1412) = z(24)*z(1390) - z(23)*z(1389);
z(2197) = z(441)*z(2170) - z(421)*z(2172);
z(2223) = z(1788) + z(2197);
z(2184) = z(421)*z(2171) - z(431)*z(2170);
z(2233) = z(1804) + z(2184);
z(2210) = z(431)*z(2172) - z(441)*z(2171);
z(2213) = z(1772) + z(2210);
z(1679) = z(23)*z(1658);
z(1669) = z(24)*z(1658);
z(1682) = z(24)*z(1660) - z(23)*z(1659);
z(1968) = z(421)*z(1955) - z(431)*z(1954);
z(2017) = z(1804) + z(1968);
z(1994) = z(431)*z(1956) - z(441)*z(1955);
z(1997) = z(1772) + z(1994);
z(1981) = z(441)*z(1954) - z(421)*z(1956);
z(2007) = z(1788) + z(1981);
z(3018) = 4.7299E-
05*z(462)*(z(2089)+44.21104040254551*z(2079)+114.2537897207129*z(207)*z(2069)+114.2537897207129*z(208)*z(2079)+114.2537897207129*z(
209)*z(2089)-147.8697224042791*z(2069)-21.77701431319901*z(210)*z(2069)-21.77701431319901*z(211)*z(2079)-
21.77701431319901*z(212)*z(2089)-3.298991522019493*z(213)*z(2069)-3.298991522019493*z(214)*z(2079)-3.298991522019493*z(215)*z(2089))
+ 4.9681E-
05*z(450)*(8.034661138060828*z(1291)+19.09315432459089*z(1301)+z(175)*z(1291)+25.99182786175802*z(170)*z(1304)+25.99182786175802*z(
176)*z(1301)+410.7868199110324*z(166)*z(1304)+410.7868199110324*z(172)*z(1301)-396.0548298142147*z(1304)-
410.7868199110324*z(173)*z(1291)-25.99182786175802*z(177)*z(1291)-z(168)*z(1304)-z(174)*z(1301)) +
0.000263142*z(478)*(z(2161)+57.96163288262611*z(2141)-37.04060925279887*z(2151)-58.17953804409786*z(207)*z(2141)-
58.17953804409786*z(208)*z(2151)-58.17953804409786*z(209)*z(2161)-3.892632114979745*z(213)*z(2141)-3.892632114979745*z(214)*z(2151)-
3.892632114979745*z(215)*z(2161)-3.889998555912777*z(210)*z(2141)-3.889998555912777*z(211)*z(2151)-3.889998555912777*z(212)*z(2161))
+
0.000156039*z(466)*(6.601112542377225*z(1355)+4.285595267849704*z(175)*z(1345)+5.76932689904447*z(170)*z(1358)+5.76932689904447*z(1
76)*z(1355)+34.22740468728971*z(166)*z(1358)+34.22740468728971*z(172)*z(1355)-34.63294432802056*z(1358)-z(1345)-
34.22740468728971*z(173)*z(1345)-5.76932689904447*z(177)*z(1345)-4.285595267849704*z(168)*z(1358)-4.285595267849704*z(174)*z(1355)) +
0.00297212*z(498)*(2.289867165524945*z(1561)+2.986921120277781*z(1574)+1.384676594484745*z(175)*z(1561)+2.471453709809833*z(173)*z(
1561)+2.589087923771584*z(170)*z(1574)+2.589087923771584*z(176)*z(1571)-z(1571)-2.589087923771584*z(177)*z(1561)-
2.471453709809833*z(166)*z(1574)-2.471453709809833*z(172)*z(1571)-1.384676594484745*z(168)*z(1574)-1.384676594484745*z(174)*z(1571))
+ 0.005250594000000001*z(510)*(1.3792740402324*z(2305)+1.576289654084852*z(2285)-1.220873295478569*z(2295)-

1.749405305380686*z(207)*z(2285)-1.749405305380686*z(208)*z(2295)-1.749405305380686*z(209)*z(2305)-1.677448684853561*z(213)*z(2285)-
1.677448684853561*z(214)*z(2295)-1.677448684853561*z(215)*z(2305)-z(210)*z(2285)-z(211)*z(2295)-z(212)*z(2305)) -
9.800000000000001*SW*(z(2790)*z(2914)+z(2791)*z(2829)+z(2792)*z(2927)) - 1.4*SW*(z(199)*z(1853)+z(203)*z(1863)+z(204)*z(1873)) -
1.4*SW*(z(284)*z(2494)+z(286)*z(2507)+z(288)*z(2520)) - 1.4*SW*(z(2676)+z(284)*z(2695)+z(286)*z(2708)+z(288)*z(2721)) -
1.4*SW*(z(20)*z(1196)-z(194)*z(1193)-z(196)*z(1183)) - 6.128E-
05*z(482)*(19.44327676240209*z(1399)+120.5957898172324*z(1409)+z(170)*z(1412)+z(176)*z(1409)+1.221507832898173*z(168)*z(1412)+1.2215
07832898173*z(174)*z(1409)+232.8573759791123*z(166)*z(1412)+232.8573759791123*z(172)*z(1409)-238.8415469973891*z(1412)-
232.8573759791123*z(173)*z(1399)-1.221507832898173*z(175)*z(1399)-z(177)*z(1399)) -
0.00508017*z(494)*(z(2223)+1.575266182037215*z(2233)+1.26191682561804*z(210)*z(2213)+1.26191682561804*z(211)*z(2223)+1.261916825618
04*z(212)*z(2233)+1.738918973183969*z(207)*z(2213)+1.738918973183969*z(208)*z(2223)+1.738918973183969*z(209)*z(2233)-
1.564399813392072*z(2213)-1.445018572213135*z(213)*z(2213)-1.445018572213135*z(214)*z(2223)-1.445018572213135*z(215)*z(2233)) -
0.00334198*z(514)*(z(1679)+2.523796072986673*z(1669)+1.513417495017923*z(168)*z(1682)+1.513417495017923*z(174)*z(1679)+2.1783128564
50368*z(170)*z(1682)+2.178312856450368*z(176)*z(1679)+2.199101730112089*z(166)*z(1682)+2.199101730112089*z(172)*z(1679)-
2.541448183412228*z(1682)-2.199101730112089*z(173)*z(1669)-2.178312856450368*z(177)*z(1669)-1.513417495017923*z(175)*z(1669)) -
z(446)*(2.0E-
07*z(2017)+0.0210342*z(1997)+0.0009485670000000001*z(210)*z(1997)+0.0009485670000000001*z(211)*z(2007)+0.0009485670000000001*z(212
)*z(2017)-0.001343886*z(2007)-0.0196764*z(207)*z(1997)-0.0196764*z(208)*z(2007)-0.0196764*z(209)*z(2017)-0.00039917*z(213)*z(1997)-
0.00039917*z(214)*z(2007)-0.00039917*z(215)*z(2017));
z(3338) = z(252)*z(421) + z(255)*z(431) + z(258)*z(441);
z(3351) = z(253)*z(421) + z(256)*z(431) + z(259)*z(441);
z(3364) = z(254)*z(421) + z(257)*z(431) + z(260)*z(441);
z(3414) = z(421)*z(2730) + z(431)*z(2733) + z(441)*z(2736);
z(3427) = z(421)*z(2731) + z(431)*z(2734) + z(441)*z(2737);
z(3440) = z(421)*z(2732) + z(431)*z(2735) + z(441)*z(2738);
z(3499) = z(421)*z(2868) + z(431)*z(2871) + z(441)*z(2874);
z(3512) = z(421)*z(2869) + z(431)*z(2872) + z(441)*z(2875);
z(3525) = z(421)*z(2870) + z(431)*z(2873) + z(441)*z(2876);
z(4381) = z(23)*z(3253) + z(24)*z(3254) + z(421)*z(3326) + z(431)*z(3325) + z(441)*z(3324) + z(3338)*z(3402) + z(3351)*z(3401) +
z(3364)*z(3400) + z(3414)*z(3487) + z(3427)*z(3486) + z(3440)*z(3485) + z(3499)*z(3569) + z(3512)*z(3568) + z(3525)*z(3567) + z(23)*z(3252) +
z(24)*z(3251) + z(421)*z(3321) + z(431)*z(3322) + z(441)*z(3323) + z(3338)*z(3397) + z(3351)*z(3398) + z(3364)*z(3399) + z(3414)*z(3482) +
z(3427)*z(3483) + z(3440)*z(3484) + z(3499)*z(3564) + z(3512)*z(3565) + z(3525)*z(3566) +
SW*(z(2829)*z(4372)+z(2914)*z(4370)+z(2927)*z(4371)) + 0.1428571428571429*SW*(z(1853)*z(3867)+z(1863)*z(3869)+z(1873)*z(3868)) +
0.1428571428571429*SW*(z(2494)*z(4020)+z(2507)*z(4022)+z(2520)*z(4021)) - 0.1428571428571429*SW*(z(1193)*z(3743)-z(1183)*z(3742)-
z(1196)*z(3741)) - 0.1428571428571429*SW*(z(34)*z(2663)*z(4191)+z(34)*z(2695)*z(4190)-z(2663)*z(4190)-z(2676)*z(4176)-z(2689)*z(4162)-
z(2695)*z(4191)-z(2708)*z(4193)-z(2721)*z(4192)-z(250)*z(2663)*z(4193)-z(250)*z(2708)*z(4190)-z(251)*z(2663)*z(4192)-
z(251)*z(2721)*z(4190)-z(283)*z(2689)*z(4191)-z(283)*z(2695)*z(4162)-z(284)*z(2676)*z(4191)-z(284)*z(2695)*z(4176)-z(285)*z(2689)*z(4193)-
z(285)*z(2708)*z(4162)-z(286)*z(2676)*z(4193)-z(286)*z(2708)*z(4176)-z(287)*z(2689)*z(4192)-z(287)*z(2721)*z(4162)-z(288)*z(2676)*z(4192)-
z(288)*z(2721)*z(4176));
z(4405) = z(3018) - z(4381);
z(4524) = z(4512)*z(4397) - z(4405);
z(4647) = z(4524) - z(4636)*z(4433);
z(4749) = z(4647) - z(4739)*z(4575);
z(4832) = z(4749) - z(4823)*z(4694);
z(4898) = z(4832) - z(4890)*z(4792);
z(4949) = z(4898) - z(4942)*z(4871);
z(4987) = z(4949) - z(4981)*z(4933);
z(5014) = z(4987) - z(5009)*z(4980);
z(5047) = z(5032) - z(5043)*z(5014);
z(2930) = z(113)*z(735) + z(115)*z(731) + z(117)*z(727);
z(4560) = z(2930)*z(4551);
z(736) = -z(116)*BOBA3 - z(118)*z(725);
z(732) = z(114)*BOBA3 + z(118)*z(724);
z(728) = z(114)*z(725) - z(116)*z(724);
z(2943) = z(113)*z(736) + z(115)*z(732) + z(117)*z(728);
z(4430) = z(2943) - z(2930)*z(4421);
z(4680) = -z(4560) - z(4430)*z(4672);
z(734) = -z(80)*BOBA3 - z(81)*z(725);
z(730) = z(81)*z(724) - z(10)*BOBA3;
z(726) = -z(10)*z(725) - z(80)*z(724);
z(2956) = z(113)*z(734) + z(115)*z(730) + z(117)*z(726);
z(4443) = z(2956) - z(2930)*z(4434);
z(4572) = z(4443) - z(4430)*z(4564);
z(4779) = z(4680) - z(4572)*z(4772);
z(4456) = z(2930)*z(4447);
z(4584) = -z(4456) - z(4430)*z(4576);
z(4691) = z(4584) - z(4572)*z(4684);
z(4859) = z(4779) - z(4691)*z(4853);
z(4469) = z(2930)*z(4460);
z(4596) = -z(4469) - z(4430)*z(4588);
z(4702) = z(4596) - z(4572)*z(4695);
z(4789) = z(4702) - z(4691)*z(4783);
z(4922) = z(4859) - z(4789)*z(4917);
z(4482) = z(2930)*z(4473);
z(4608) = -z(4482) - z(4430)*z(4600);
z(4713) = z(4608) - z(4572)*z(4706);
z(4799) = z(4713) - z(4691)*z(4793);
z(4868) = z(4799) - z(4789)*z(4863);
z(4970) = z(4922) - z(4868)*z(4966);

```
z(4495) = z(2930)*z(4486);
z(4620) = -z(4495) - z(4430)*z(4612);
z(4724) = z(4620) - z(4572)*z(4717);
z(4809) = z(4724) - z(4691)*z(4803);
z(4877) = z(4809) - z(4789)*z(4872);
z(4930) = z(4877) - z(4868)*z(4926);
z(5005) = z(4970) - z(4930)*z(5002);
z(4508) = z(2930)*z(4499);
z(4632) = -z(4508) - z(4430)*z(4624);
z(4735) = z(4632) - z(4572)*z(4728);
z(4819) = z(4735) - z(4691)*z(4813);
z(4886) = z(4819) - z(4789)*z(4881);
z(4938) = z(4886) - z(4868)*z(4934);
z(4977) = z(4938) - z(4930)*z(4974);
z(5029) = z(5005) - z(4977)*z(5027);
z(4521) = z(2930)*z(4512);
z(4644) = -z(4521) - z(4430)*z(4636);
z(4746) = z(4644) - z(4572)*z(4739);
z(4829) = z(4746) - z(4691)*z(4823);
z(4895) = z(4829) - z(4789)*z(4890);
z(4946) = z(4895) - z(4868)*z(4942);
z(4984) = z(4946) - z(4930)*z(4981);
z(5011) = z(4984) - z(4977)*z(5009);
z(5044) = z(5029) - z(5011)*z(5043);
z(2342) = -z(201)*EOED3 - z(202)*z(2315);
z(2329) = z(197)*EOED3 + z(202)*z(2314);
z(2316) = z(197)*z(2315) - z(201)*z(2314);
z(3019) = z(197)*z(2342) + z(201)*z(2329) + z(202)*z(2316);
z(4525) = z(3019)/z(2938);
z(4534) = z(2930)*z(4525);
z(3020) = z(199)*z(2342) + z(203)*z(2329) + z(204)*z(2316);
z(4526) = z(3020) - z(2939)*z(4525);
z(4648) = z(4526)/z(4422);
z(4656) = -z(4534) - z(4430)*z(4648);
z(3021) = z(205)*z(2329) + z(206)*z(2316) - z(28)*z(2342);
z(4527) = z(3021) - z(2940)*z(4525);
z(4649) = z(4527) - z(4423)*z(4648);
z(4750) = z(4649)/z(4565);
z(4757) = z(4656) - z(4572)*z(4750);
z(4528) = z(2935)*z(4525);
z(4650) = -z(4528) - z(4424)*z(4648);
z(4751) = z(4650) - z(4566)*z(4750);
z(4833) = z(4751)/z(4685);
z(4839) = z(4757) - z(4691)*z(4833);
z(4529) = z(2936)*z(4525);
z(4651) = -z(4529) - z(4425)*z(4648);
z(4752) = z(4651) - z(4567)*z(4750);
z(4834) = z(4752) - z(4686)*z(4833);
z(4899) = z(4834)/z(4784);
z(4904) = z(4839) - z(4789)*z(4899);
z(4530) = z(2937)*z(4525);
z(4652) = -z(4530) - z(4426)*z(4648);
z(4753) = z(4652) - z(4568)*z(4750);
z(4835) = z(4753) - z(4687)*z(4833);
z(4900) = z(4835) - z(4785)*z(4899);
z(4950) = z(4900)/z(4864);
z(4954) = z(4904) - z(4868)*z(4950);
z(4531) = z(2932)*z(4525);
z(4653) = -z(4531) - z(4427)*z(4648);
z(4754) = z(4653) - z(4569)*z(4750);
z(4836) = z(4754) - z(4688)*z(4833);
z(4901) = z(4836) - z(4786)*z(4899);
z(4951) = z(4901) - z(4865)*z(4950);
z(4988) = z(4951)/z(4927);
z(4991) = z(4954) - z(4930)*z(4988);
z(4532) = z(2933)*z(4525);
z(4654) = -z(4532) - z(4428)*z(4648);
z(4755) = z(4654) - z(4570)*z(4750);
z(4837) = z(4755) - z(4689)*z(4833);
z(4902) = z(4837) - z(4787)*z(4899);
z(4952) = z(4902) - z(4866)*z(4950);
z(4989) = z(4952) - z(4928)*z(4988);
z(5015) = z(4989)/z(4975);
z(5017) = z(4991) - z(4977)*z(5015);
z(4533) = z(2934)*z(4525);
z(4655) = -z(4533) - z(4429)*z(4648);
z(4756) = z(4655) - z(4571)*z(4750);
z(4838) = z(4756) - z(4690)*z(4833);
```

```
z(4903) = z(4838) - z(4788)*z(4899);
z(4953) = z(4903) - z(4867)*z(4950);
z(4990) = z(4953) - z(4929)*z(4988);
z(5016) = z(4990) - z(4976)*z(5015);
z(5033) = z(5016)/z(5010);
z(5034) = z(5017) - z(5011)*z(5033);
z(5052) = z(5044)/z(5034);
z(2029) = z(197)*z(2027) - z(201)*z(2026);
z(2042) = z(202)*z(2026) - z(197)*z(2028);
z(2055) = z(201)*z(2028) - z(202)*z(2027);
z(2101) = z(197)*z(2099) - z(201)*z(2098);
z(2127) = z(201)*z(2100) - z(202)*z(2099);
z(2114) = z(202)*z(2098) - z(197)*z(2100);
z(2245) = z(197)*z(2243) - z(201)*z(2242);
z(2271) = z(201)*z(2244) - z(202)*z(2243);
z(2258) = z(202)*z(2242) - z(197)*z(2244);
z(2186) = z(202)*z(2170) - z(197)*z(2172);
z(2173) = z(197)*z(2171) - z(201)*z(2170);
z(2199) = z(201)*z(2172) - z(202)*z(2171);
z(1957) = z(197)*z(1955) - z(201)*z(1954);
z(1983) = z(201)*z(1956) - z(202)*z(1955);
z(1970) = z(202)*z(1954) - z(197)*z(1956);
z(3022) = 4.7299E-
05*z(462)*(z(2029)+44.21104040254551*z(2042)+114.2537897207129*z(207)*z(2055)+114.2537897207129*z(208)*z(2042)+114.2537897207129*z(
209)*z(2029)-147.8697224042791*z(2055)-21.77701431319901*z(210)*z(2055)-21.77701431319901*z(211)*z(2042)-
21.77701431319901*z(212)*z(2029)-3.298991522019493*z(213)*z(2055)-3.298991522019493*z(214)*z(2042)-3.298991522019493*z(215)*z(2029))
+ 0.000263142*z(478)*(z(2101)+57.96163288262611*z(2127)-37.04060925279887*z(2114)-58.17953804409786*z(207)*z(2127)-
58.17953804409786*z(208)*z(2114)-58.17953804409786*z(209)*z(2101)-3.892632114979745*z(213)*z(2127)-3.892632114979745*z(214)*z(2114)-
3.892632114979745*z(215)*z(2101)-3.889998555912777*z(210)*z(2127)-3.889998555912777*z(211)*z(2114)-3.889998555912777*z(212)*z(2101))
+ 0.005250594000000001*z(510)*(1.3792740402324*z(2245)+1.576289654084852*z(2271)-1.220873295478569*z(2258)-
1.749405305380686*z(207)*z(2271)-1.749405305380686*z(208)*z(2258)-1.749405305380686*z(209)*z(2245)-1.677448684853561*z(213)*z(2271)-
1.677448684853561*z(214)*z(2258)-1.677448684853561*z(215)*z(2245)-z(210)*z(2271)-z(211)*z(2258)-z(212)*z(2245)) -
9.800000000000001*SW*(z(2790)*z(2904)+z(2791)*z(2818)+z(2792)*z(2916)) - 1.4*SW*(z(199)*z(1839)+z(203)*z(1826)+z(204)*z(1813)) -
1.4*SW*(z(284)*z(2490)+z(286)*z(2503)+z(288)*z(2516)) - 1.4*SW*(z(2665)+z(284)*z(2691)+z(286)*z(2704)+z(288)*z(2717)) -
0.00508017*z(494)*(z(2186)+1.575266182037215*z(2173)+1.2619168256180 4*z(210)*z(2199)+1.2619168256180 4*z(211)*z(2186)+1.261916825618
04*z(212)*z(2173)+1.738918973183969*z(207)*z(2199)+1.738918973183969*z(208)*z(2186)+1.738918973183969*z(209)*z(2173)-
1.564399813392072*z(2199)-1.445018572213135*z(213)*z(2199)-1.445018572213135*z(214)*z(2186)-1.445018572213135*z(215)*z(2173)) -
z(446)*(2.0E-
07*z(1957)+0.0210342*z(1983)+0.0009485670000000001*z(210)*z(1983)+0.0009485670000000001*z(211)*z(1970)+0.0009485670000000001*z(212
)*z(1957)-0.001343886*z(1970)-0.0196764*z(207)*z(1983)-0.0196764*z(208)*z(1970)-0.0196764*z(209)*z(1957)-0.00039917*z(213)*z(1983)-
0.00039917*z(214)*z(1970)-0.00039917*z(215)*z(1957));
z(3327) = z(197)*z(252) + z(201)*z(255) + z(202)*z(258);
z(3340) = z(197)*z(253) + z(201)*z(256) + z(202)*z(259);
z(3353) = z(197)*z(254) + z(201)*z(257) + z(202)*z(260);
z(3403) = z(197)*z(2730) + z(201)*z(2733) + z(202)*z(2736);
z(3416) = z(197)*z(2731) + z(201)*z(2734) + z(202)*z(2737);
z(3429) = z(197)*z(2732) + z(201)*z(2735) + z(202)*z(2738);
z(3488) = z(197)*z(2868) + z(201)*z(2871) + z(202)*z(2874);
z(3501) = z(197)*z(2869) + z(201)*z(2872) + z(202)*z(2875);
z(3514) = z(197)*z(2870) + z(201)*z(2873) + z(202)*z(2876);
z(4382) = z(197)*z(3326) + z(201)*z(3325) + z(202)*z(3324) + z(3327)*z(3402) + z(3340)*z(3401) + z(3353)*z(3400) + z(3403)*z(3487) +
z(3416)*z(3486) + z(3429)*z(3485) + z(3488)*z(3569) + z(3501)*z(3568) + z(3514)*z(3567) + z(197)*z(3321) + z(201)*z(3322) + z(202)*z(3323) +
z(3327)*z(3397) + z(3340)*z(3398) + z(3353)*z(3399) + z(3403)*z(3482) + z(3416)*z(3483) + z(3429)*z(3484) + z(3488)*z(3564) + z(3501)*z(3565)
+ z(3514)*z(3566) + SW*(z(2818)*z(4372)+z(2904)*z(4370)+z(2916)*z(4371)) +
0.1428571428571429*SW*(z(1813)*z(3868)+z(1826)*z(3869)+z(1839)*z(3867)) +
0.1428571428571429*SW*(z(2490)*z(4020)+z(2503)*z(4022)+z(2516)*z(4021)) -
0.1428571428571429*SW*(z(34)*z(2652)*z(4191)+z(34)*z(2691)*z(4190)-z(2652)*z(4190)-z(2665)*z(4176)-z(2678)*z(4162)-z(2691)*z(4191)-
z(2704)*z(4193)-z(2717)*z(4192)-z(250)*z(2652)*z(4193)-z(250)*z(2704)*z(4190)-z(251)*z(2652)*z(4192)-z(251)*z(2717)*z(4190)-
z(283)*z(2678)*z(4191)-z(283)*z(2691)*z(4162)-z(284)*z(2665)*z(4191)-z(284)*z(2691)*z(4176)-z(285)*z(2678)*z(4193)-z(285)*z(2704)*z(4162)-
z(286)*z(2665)*z(4193)-z(286)*z(2704)*z(4176)-z(287)*z(2678)*z(4192)-z(287)*z(2717)*z(4162)-z(288)*z(2665)*z(4192)-z(288)*z(2717)*z(4176));
z(4406) = z(3022) - z(4382);
z(4537) = z(4525)*z(4397) - z(4406);
z(4659) = z(4537) - z(4648)*z(4433);
z(4760) = z(4659) - z(4750)*z(4575);
z(4842) = z(4760) - z(4833)*z(4694);
z(4907) = z(4842) - z(4899)*z(4792);
z(4957) = z(4907) - z(4950)*z(4871);
z(4994) = z(4957) - z(4988)*z(4933);
z(5020) = z(4994) - z(5015)*z(4980);
z(5037) = z(5020) - z(5033)*z(5014);
z(5055) = z(5047) - z(5052)*z(5037);
z(2931) = z(114)*z(735) + z(116)*z(731) + z(118)*z(727);
z(4561) = z(2931)*z(4551);
z(2944) = z(114)*z(736) + z(116)*z(732) + z(118)*z(728);
z(4431) = z(2944) - z(2931)*z(4421);
z(4681) = -z(4561) - z(4431)*z(4672);
z(2957) = z(114)*z(734) + z(116)*z(730) + z(118)*z(726);
z(4444) = z(2957) - z(2931)*z(4434);
```

```
z(4573) = z(4444) - z(4431)*z(4564);
z(4780) = z(4681) - z(4573)*z(4772);
z(4457) = z(2931)*z(4447);
z(4585) = -z(4457) - z(4431)*z(4576);
z(4692) = z(4585) - z(4573)*z(4684);
z(4860) = z(4780) - z(4692)*z(4853);
z(4470) = z(2931)*z(4460);
z(4597) = -z(4470) - z(4431)*z(4588);
z(4703) = z(4597) - z(4573)*z(4695);
z(4790) = z(4703) - z(4692)*z(4783);
z(4923) = z(4860) - z(4790)*z(4917);
z(4483) = z(2931)*z(4473);
z(4609) = -z(4483) - z(4431)*z(4600);
z(4714) = z(4609) - z(4573)*z(4706);
z(4800) = z(4714) - z(4692)*z(4793);
z(4869) = z(4800) - z(4790)*z(4863);
z(4971) = z(4923) - z(4869)*z(4966);
z(4496) = z(2931)*z(4486);
z(4621) = -z(4496) - z(4431)*z(4612);
z(4725) = z(4621) - z(4573)*z(4717);
z(4810) = z(4725) - z(4692)*z(4803);
z(4878) = z(4810) - z(4790)*z(4872);
z(4931) = z(4878) - z(4869)*z(4926);
z(5006) = z(4971) - z(4931)*z(5002);
z(4509) = z(2931)*z(4499);
z(4633) = -z(4509) - z(4431)*z(4624);
z(4736) = z(4633) - z(4573)*z(4728);
z(4820) = z(4736) - z(4692)*z(4813);
z(4887) = z(4820) - z(4790)*z(4881);
z(4939) = z(4887) - z(4869)*z(4934);
z(4978) = z(4939) - z(4931)*z(4974);
z(5030) = z(5006) - z(4978)*z(5027);
z(4522) = z(2931)*z(4512);
z(4645) = -z(4522) - z(4431)*z(4636);
z(4747) = z(4645) - z(4573)*z(4739);
z(4830) = z(4747) - z(4692)*z(4823);
z(4896) = z(4830) - z(4790)*z(4890);
z(4947) = z(4896) - z(4869)*z(4942);
z(4985) = z(4947) - z(4931)*z(4981);
z(5012) = z(4985) - z(4978)*z(5009);
z(5045) = z(5030) - z(5012)*z(5043);
z(4535) = z(2931)*z(4525);
z(4657) = -z(4535) - z(4431)*z(4648);
z(4758) = z(4657) - z(4573)*z(4750);
z(4840) = z(4758) - z(4692)*z(4833);
z(4905) = z(4840) - z(4790)*z(4899);
z(4955) = z(4905) - z(4869)*z(4950);
z(4992) = z(4955) - z(4931)*z(4988);
z(5018) = z(4992) - z(4978)*z(5015);
z(5035) = z(5018) - z(5012)*z(5033);
z(5053) = z(5045) - z(5035)*z(5052);
z(2343) = -z(203)*EOED3 - z(204)*z(2315);
z(2330) = z(199)*EOED3 + z(204)*z(2314);
z(2317) = z(199)*z(2315) - z(203)*z(2314);
z(3023) = z(197)*z(2343) + z(201)*z(2330) + z(202)*z(2317);
z(4538) = z(3023)/z(2938);
z(4548) = z(2931)*z(4538);
z(3024) = z(199)*z(2343) + z(203)*z(2330) + z(204)*z(2317);
z(4539) = z(3024) - z(2939)*z(4538);
z(4660) = z(4539)/z(4422);
z(4669) = -z(4548) - z(4431)*z(4660);
z(3025) = z(205)*z(2330) + z(206)*z(2317) - z(28)*z(2343);
z(4540) = z(3025) - z(2940)*z(4538);
z(4661) = z(4540) - z(4423)*z(4660);
z(4761) = z(4661)/z(4565);
z(4769) = z(4669) - z(4573)*z(4761);
z(4541) = z(2935)*z(4538);
z(4662) = -z(4541) - z(4424)*z(4660);
z(4762) = z(4662) - z(4566)*z(4761);
z(4843) = z(4762)/z(4685);
z(4850) = z(4769) - z(4692)*z(4843);
z(4542) = z(2936)*z(4538);
z(4663) = -z(4542) - z(4425)*z(4660);
z(4763) = z(4663) - z(4567)*z(4761);
z(4844) = z(4763) - z(4686)*z(4843);
z(4908) = z(4844)/z(4784);
z(4914) = z(4850) - z(4790)*z(4908);
z(4543) = z(2937)*z(4538);
```

z(4664) = -z(4543) - z(4426)*z(4660);
z(4764) = z(4664) - z(4568)*z(4761);
z(4845) = z(4764) - z(4687)*z(4843);
z(4909) = z(4845) - z(4785)*z(4908);
z(4958) = z(4909)/z(4864);
z(4963) = z(4914) - z(4869)*z(4958);
z(4544) = z(2932)*z(4538);
z(4665) = -z(4544) - z(4427)*z(4660);
z(4765) = z(4665) - z(4569)*z(4761);
z(4846) = z(4765) - z(4688)*z(4843);
z(4910) = z(4846) - z(4786)*z(4908);
z(4959) = z(4910) - z(4865)*z(4958);
z(4995) = z(4959)/z(4927);
z(4999) = z(4963) - z(4931)*z(4995);
z(4545) = z(2933)*z(4538);
z(4666) = -z(4545) - z(4428)*z(4660);
z(4766) = z(4666) - z(4570)*z(4761);
z(4847) = z(4766) - z(4689)*z(4843);
z(4911) = z(4847) - z(4787)*z(4908);
z(4960) = z(4911) - z(4866)*z(4958);
z(4996) = z(4960) - z(4928)*z(4995);
z(5021) = z(4996)/z(4975);
z(5024) = z(4999) - z(4978)*z(5021);
z(4546) = z(2934)*z(4538);
z(4667) = -z(4546) - z(4429)*z(4660);
z(4767) = z(4667) - z(4571)*z(4761);
z(4848) = z(4767) - z(4690)*z(4843);
z(4912) = z(4848) - z(4788)*z(4908);
z(4961) = z(4912) - z(4867)*z(4958);
z(4997) = z(4961) - z(4929)*z(4995);
z(5022) = z(4997) - z(4976)*z(5021);
z(5038) = z(5022)/z(5010);
z(5040) = z(5024) - z(5012)*z(5038);
z(4547) = z(2930)*z(4538);
z(4668) = -z(4547) - z(4430)*z(4660);
z(4768) = z(4668) - z(4572)*z(4761);
z(4849) = z(4768) - z(4691)*z(4843);
z(4913) = z(4849) - z(4789)*z(4908);
z(4962) = z(4913) - z(4868)*z(4958);
z(4998) = z(4962) - z(4930)*z(4995);
z(5023) = z(4998) - z(4977)*z(5021);
z(5039) = z(5023) - z(5011)*z(5038);
z(5048) = z(5039)/z(5034);
z(5049) = z(5040) - z(5035)*z(5048);
z(5056) = z(5053)/z(5049);
z(2030) = z(199)*z(2027) - z(203)*z(2026);
z(2043) = z(204)*z(2026) - z(199)*z(2028);
z(2056) = z(203)*z(2028) - z(204)*z(2027);
z(2102) = z(199)*z(2099) - z(203)*z(2098);
z(2128) = z(203)*z(2100) - z(204)*z(2099);
z(2115) = z(204)*z(2098) - z(199)*z(2100);
z(2246) = z(199)*z(2243) - z(203)*z(2242);
z(2272) = z(203)*z(2244) - z(204)*z(2243);
z(2259) = z(204)*z(2242) - z(199)*z(2244);
z(2187) = z(204)*z(2170) - z(199)*z(2172);
z(2174) = z(199)*z(2171) - z(203)*z(2170);
z(2200) = z(203)*z(2172) - z(204)*z(2171);
z(1958) = z(199)*z(1955) - z(203)*z(1954);
z(1984) = z(203)*z(1956) - z(204)*z(1955);
z(1971) = z(204)*z(1954) - z(199)*z(1956);
z(3026) = 4.7299E-
05*z(462)*(z(2030)+44.21104040254551*z(2043)+114.2537897207129*z(207)*z(2056)+114.2537897207129*z(208)*z(2043)+114.2537897207129*z(
209)*z(2030)-147.8697224042791*z(2056)-21.77701431319901*z(210)*z(2056)-21.77701431319901*z(211)*z(2043)-
21.77701431319901*z(212)*z(2030)-3.298991522019493*z(213)*z(2056)-3.298991522019493*z(214)*z(2043)-3.298991522019493*z(215)*z(2030))
+ 0.000263142*z(478)*(z(2102)+57.96163288262611*z(2128)-37.04060925279887*z(2115)-58.17953804409786*z(207)*z(2128)-
58.17953804409786*z(208)*z(2115)-58.17953804409786*z(209)*z(2102)-3.892632114979745*z(213)*z(2128)-3.892632114979745*z(214)*z(2115)-
3.892632114979745*z(215)*z(2102)-3.889998555912777*z(210)*z(2128)-3.889998555912777*z(211)*z(2115)-3.889998555912777*z(212)*z(2102))
+ 0.005250594000000001*z(510)*(1.3792740402324*z(2246)+1.576289654084852*z(2272)-1.220873295478569*z(2259)-
1.749405305380686*z(207)*z(2272)-1.749405305380686*z(208)*z(2259)-1.749405305380686*z(209)*z(2246)-1.677448684853561*z(213)*z(2272)-
1.677448684853561*z(214)*z(2259)-1.677448684853561*z(215)*z(2246)-z(210)*z(2272)-z(211)*z(2259)-z(212)*z(2246)) -
9.800000000000001*SW*(z(2790)*z(2905)+z(2791)*z(2819)+z(2792)*z(2917)) - 1.4*SW*(z(199)*z(1840)+z(203)*z(1827)+z(204)*z(1814)) -
1.4*SW*(z(284)*z(2491)+z(286)*z(2504)+z(288)*z(2517)) - 1.4*SW*(z(2666)+z(284)*z(2692)+z(286)*z(2705)+z(288)*z(2718)) -
0.00508017*z(494)*(z(2187)+1.575266182037215*z(2174)+1.26191682561804*z(210)*z(2200)+1.26191682561804*z(211)*z(2187)+1.261916825618
04*z(212)*z(2174)+1.738918973183969*z(207)*z(2200)+1.738918973183969*z(208)*z(2187)+1.738918973183969*z(209)*z(2174)-
1.564399813392072*z(2200)-1.445018572213135*z(213)*z(2200)-1.445018572213135*z(214)*z(2187)-1.445018572213135*z(215)*z(2174)) -
z(446)*(2.0E-
07*z(1958)+0.0210342*z(1984)+0.0009485670000000001*z(210)*z(1984)+0.0009485670000000001*z(211)*z(1971)+0.0009485670000000001*z(212

)*z(1958)-0.001343886*z(1971)-0.0196764*z(207)*z(1984)-0.0196764*z(208)*z(1971)-0.0196764*z(209)*z(1958)-0.00039917*z(213)*z(1984)-
0.00039917*z(214)*z(1971)-0.00039917*z(215)*z(1958));
z(3328) = z(199)*z(252) + z(203)*z(255) + z(204)*z(258);
z(3341) = z(199)*z(253) + z(203)*z(256) + z(204)*z(259);
z(3354) = z(199)*z(254) + z(203)*z(257) + z(204)*z(260);
z(3404) = z(199)*z(2730) + z(203)*z(2733) + z(204)*z(2736);
z(3417) = z(199)*z(2731) + z(203)*z(2734) + z(204)*z(2737);
z(3430) = z(199)*z(2732) + z(203)*z(2735) + z(204)*z(2738);
z(3489) = z(199)*z(2868) + z(203)*z(2871) + z(204)*z(2874);
z(3502) = z(199)*z(2869) + z(203)*z(2872) + z(204)*z(2875);
z(3515) = z(199)*z(2870) + z(203)*z(2873) + z(204)*z(2876);
z(4383) = z(199)*z(3326) + z(203)*z(3325) + z(204)*z(3324) + z(3328)*z(3402) + z(3341)*z(3401) + z(3354)*z(3400) + z(3404)*z(3487) +
z(3417)*z(3486) + z(3430)*z(3485) + z(3489)*z(3569) + z(3502)*z(3568) + z(3515)*z(3567) + z(199)*z(3321) + z(203)*z(3322) + z(204)*z(3323) +
z(3328)*z(3397) + z(3341)*z(3398) + z(3354)*z(3399) + z(3404)*z(3482) + z(3417)*z(3483) + z(3430)*z(3484) + z(3489)*z(3564) + z(3502)*z(3565)
+ z(3515)*z(3566) + SW*(z(2819)*z(4372)+z(2905)*z(4370)+z(2917)*z(4371)) +
0.1428571428571429*SW*(z(1814)*z(3868)+z(1827)*z(3869)+z(1840)*z(3867)) +
0.1428571428571429*SW*(z(2491)*z(4020)+z(2504)*z(4022)+z(2517)*z(4021)) -
0.1428571428571429*SW*(z(34)*z(2653)*z(4191)+z(34)*z(2692)*z(4190)-z(2653)*z(4190)-z(2666)*z(4176)-z(2679)*z(4162)-z(2692)*z(4191)-
z(2705)*z(4193)-z(2718)*z(4192)-z(250)*z(2653)*z(4193)-z(250)*z(2705)*z(4190)-z(251)*z(2653)*z(4192)-z(251)*z(2718)*z(4190)-
z(283)*z(2679)*z(4191)-z(283)*z(2692)*z(4162)-z(284)*z(2666)*z(4191)-z(284)*z(2692)*z(4176)-z(285)*z(2679)*z(4193)-z(285)*z(2705)*z(4162)-
z(286)*z(2666)*z(4193)-z(286)*z(2705)*z(4176)-z(287)*z(2679)*z(4192)-z(287)*z(2718)*z(4162)-z(288)*z(2666)*z(4192)-z(288)*z(2718)*z(4176));
z(4407) = z(3026) - z(4383);
z(4550) = z(4538)*z(4397) - z(4407);
z(4671) = z(4550) - z(4660)*z(4433);
z(4771) = z(4671) - z(4761)*z(4575);
z(4852) = z(4771) - z(4843)*z(4694);
z(4916) = z(4852) - z(4908)*z(4792);
z(4965) = z(4916) - z(4958)*z(4871);
z(5001) = z(4965) - z(4995)*z(4933);
z(5026) = z(5001) - z(5021)*z(4980);
z(5042) = z(5026) - z(5038)*z(5014);
z(5051) = z(5042) - z(5048)*z(5037);
z(5058) = z(5055) - z(5056)*z(5051);
z(4562) = z(2929)*z(4551);
z(2942) = z(80)*z(732) + z(81)*z(728) - z(10)*z(736);
z(4432) = z(2942) - z(2929)*z(4421);
z(4682) = -z(4562) - z(4432)*z(4672);
z(2955) = z(80)*z(730) + z(81)*z(726) - z(10)*z(734);
z(4445) = z(2955) - z(2929)*z(4434);
z(4574) = z(4445) - z(4432)*z(4564);
z(4781) = z(4682) - z(4574)*z(4772);
z(4458) = z(2929)*z(4447);
z(4586) = -z(4458) - z(4432)*z(4576);
z(4693) = z(4586) - z(4574)*z(4684);
z(4861) = z(4781) - z(4693)*z(4853);
z(4471) = z(2929)*z(4460);
z(4598) = -z(4471) - z(4432)*z(4588);
z(4704) = z(4598) - z(4574)*z(4695);
z(4791) = z(4704) - z(4693)*z(4783);
z(4924) = z(4861) - z(4791)*z(4917);
z(4484) = z(2929)*z(4473);
z(4610) = -z(4484) - z(4432)*z(4600);
z(4715) = z(4610) - z(4574)*z(4706);
z(4801) = z(4715) - z(4693)*z(4793);
z(4870) = z(4801) - z(4791)*z(4863);
z(4972) = z(4924) - z(4870)*z(4966);
z(4497) = z(2929)*z(4486);
z(4622) = -z(4497) - z(4432)*z(4612);
z(4726) = z(4622) - z(4574)*z(4717);
z(4811) = z(4726) - z(4693)*z(4803);
z(4879) = z(4811) - z(4791)*z(4872);
z(4932) = z(4879) - z(4870)*z(4926);
z(5007) = z(4972) - z(4932)*z(5002);
z(4510) = z(2929)*z(4499);
z(4634) = -z(4510) - z(4432)*z(4624);
z(4737) = z(4634) - z(4574)*z(4728);
z(4821) = z(4737) - z(4693)*z(4813);
z(4888) = z(4821) - z(4791)*z(4881);
z(4940) = z(4888) - z(4870)*z(4934);
z(4979) = z(4940) - z(4932)*z(4974);
z(5031) = z(5007) - z(4979)*z(5027);
z(4523) = z(2929)*z(4512);
z(4646) = -z(4523) - z(4432)*z(4636);
z(4748) = z(4646) - z(4574)*z(4739);
z(4831) = z(4748) - z(4693)*z(4823);
z(4897) = z(4831) - z(4791)*z(4890);
z(4948) = z(4897) - z(4870)*z(4942);
z(4986) = z(4948) - z(4932)*z(4981);

```
z(5013) = z(4986) - z(4979)*z(5009);
z(5046) = z(5031) - z(5013)*z(5043);
z(4536) = z(2929)*z(4525);
z(4658) = -z(4536) - z(4432)*z(4648);
z(4759) = z(4658) - z(4574)*z(4750);
z(4841) = z(4759) - z(4693)*z(4833);
z(4906) = z(4841) - z(4791)*z(4899);
z(4956) = z(4906) - z(4870)*z(4950);
z(4993) = z(4956) - z(4932)*z(4988);
z(5019) = z(4993) - z(4979)*z(5015);
z(5036) = z(5019) - z(5013)*z(5033);
z(5054) = z(5046) - z(5036)*z(5052);
z(4549) = z(2929)*z(4538);
z(4670) = -z(4549) - z(4432)*z(4660);
z(4770) = z(4670) - z(4574)*z(4761);
z(4851) = z(4770) - z(4693)*z(4843);
z(4915) = z(4851) - z(4791)*z(4908);
z(4964) = z(4915) - z(4870)*z(4958);
z(5000) = z(4964) - z(4932)*z(4995);
z(5025) = z(5000) - z(4979)*z(5021);
z(5041) = z(5025) - z(5013)*z(5038);
z(5050) = z(5041) - z(5036)*z(5048);
z(5057) = z(5054) - z(5050)*z(5056);
z(5059) = z(5058)/z(5057);
z(5060) = (z(5051)-z(5050)*z(5059))/z(5049);
z(5061) = (z(5037)-z(5035)*z(5060)-z(5036)*z(5059))/z(5034);
z(5062) = (z(5014)-z(5011)*z(5061)-z(5012)*z(5060)-z(5013)*z(5059))/z(5010);
z(5063) = (z(4980)-z(4976)*z(5062)-z(4977)*z(5061)-z(4978)*z(5060)-z(4979)*z(5059))/z(4975);
z(5064) = (z(4933)-z(4928)*z(5063)-z(4929)*z(5062)-z(4930)*z(5061)-z(4931)*z(5060)-z(4932)*z(5059))/z(4927);
z(5065) = (z(4871)-z(4865)*z(5064)-z(4866)*z(5063)-z(4867)*z(5062)-z(4868)*z(5061)-z(4869)*z(5060)-z(4870)*z(5059))/z(4864);
z(5066) = (z(4792)-z(4785)*z(5065)-z(4786)*z(5064)-z(4787)*z(5063)-z(4788)*z(5062)-z(4789)*z(5061)-z(4790)*z(5060)-z(4791)*z(5059))/z(4784);
z(5067) = (z(4694)-z(4686)*z(5066)-z(4687)*z(5065)-z(4688)*z(5064)-z(4689)*z(5063)-z(4690)*z(5062)-z(4691)*z(5061)-z(4692)*z(5060)-
z(4693)*z(5059))/z(4685);
z(5068) = (z(4575)-z(4566)*z(5067)-z(4567)*z(5066)-z(4568)*z(5065)-z(4569)*z(5064)-z(4570)*z(5063)-z(4571)*z(5062)-z(4572)*z(5061)-
z(4573)*z(5060)-z(4574)*z(5059))/z(4565);
z(5069) = (z(4433)-z(4423)*z(5068)-z(4424)*z(5067)-z(4425)*z(5066)-z(4426)*z(5065)-z(4427)*z(5064)-z(4428)*z(5063)-z(4429)*z(5062)-
z(4430)*z(5061)-z(4431)*z(5060)-z(4432)*z(5059))/z(4422);
z(5070) =
(z(4397)+z(2929)*z(5059)+z(2930)*z(5061)+z(2931)*z(5060)+z(2932)*z(5064)+z(2933)*z(5063)+z(2934)*z(5062)+z(2935)*z(5067)+z(2936)*z(5066)
+z(2937)*z(5065)+z(2939)*z(5069)+z(2940)*z(5068))/z(2938);
FESEMCER_1 = -z(5070);
FESEMCER_2 = z(5069);
FESEMCER_3 = z(5068);
FDSEMCER_1 = z(5067);
FDSEMCER_2 = z(5066);
FDSEMCER_3 = z(5065);
FCSEMCER_1 = z(5064);
FCSEMCER_2 = z(5063);
FCSEMCER_3 = z(5062);
FBSEMCER_1 = z(5061);
FBSEMCER_2 = z(5060);
FBSEMCER_3 = z(5059);
FESEMCER = 0.0510204081632653*(FESEMCER_1^2+FESEMCER_2^2+FESEMCER_3^2)^0.5/SW;
FDSEMCER = 0.0510204081632653*(FDSEMCER_1^2+FDSEMCER_2^2+FDSEMCER_3^2)^0.5/SW;
FCSEMCER = 0.0510204081632653*(FCSEMCER_1^2+FCSEMCER_2^2+FCSEMCER_3^2)^0.5/SW;
FBSEMCER = 0.0510204081632653*(FBSEMCER_1^2+FBSEMCER_2^2+FBSEMCER_3^2)^0.5/SW;
FESEMCER1 = (FESEMCER_1^2+FESEMCER_2^2+FESEMCER_3^2)^0.5;
FDSEMCER1 = (FDSEMCER_1^2+FDSEMCER_2^2+FDSEMCER_3^2)^0.5;
FCSEMCER1 = (FCSEMCER_1^2+FCSEMCER_2^2+FCSEMCER_3^2)^0.5;
FBSEMCER1 = (FBSEMCER_1^2+FBSEMCER_2^2+FBSEMCER_3^2)^0.5;
z(3057) = z(197)*z(1770) + z(201)*z(1786) + z(202)*z(1802);
z(3060) = 0.000203764*(1+129.8428574232936*z(84)-9.530554955733102*z(87)-4.900914783769459*z(90))*z(458) +
0.000439106*(1+29.7948559117844*z(84)-5.299196549352549*z(87)-2.389958233319518*z(90))*z(474) + 4.7299E-
05*(z(1802)+44.21104040254551*z(1786)+114.2537897207129*z(207)*z(1770)+114.2537897207129*z(208)*z(1786)+114.2537897207129*z(209)*z(
1802)-147.8697224042791*z(1770)-21.77701431319901*z(210)*z(1770)-21.77701431319901*z(211)*z(1786)-21.77701431319901*z(212)*z(1802)-
3.298991522019493*z(213)*z(1770)-3.298991522019493*z(214)*z(1786)-3.298991522019493*z(215)*z(1802))*z(462) + 4.9681E-
05*(z(136)+436.869225659709*z(129)*z(135)+436.869225659709*z(130)*z(136)+436.869225659709*z(131)*z(137)-410.7868199110324*z(135)-
25.99182786175802*z(137)-23.60302731426501*z(132)*z(135)-23.60302731426501*z(133)*z(136)-23.60302731426501*z(134)*z(137)-
4.101447233348765*z(135)^2-4.101447233348765*z(136)^2-4.101447233348765*z(137)^2)*z(454) + 5.641700000000001E-
05*(z(137)+247.6572664267862*z(135)+24.25456865838311*z(132)*z(135)+24.25456865838311*z(133)*z(136)+24.25456865838311*z(134)*z(137)
-149.1704628037648*z(136)-269.1738305829803*z(129)*z(135)-269.1738305829803*z(130)*z(136)-269.1738305829803*z(131)*z(137)-
14.71584805998192*z(135)^2-14.71584805998192*z(136)^2-14.71584805998192*z(137)^2)*z(486) +
0.000263142*(z(1802)+57.96163288262611*z(1770)-37.04060925279887*z(1786)-58.17953804409786*z(207)*z(1770)-
58.17953804409786*z(208)*z(1786)-58.17953804409786*z(209)*z(1802)-3.892632114979745*z(213)*z(1770)-3.892632114979745*z(214)*z(1786)-
3.892632114979745*z(215)*z(1802)-3.889998555912777*z(210)*z(1770)-3.889998555912777*z(211)*z(1786)-
3.889998555912777*z(212)*z(1802))*z(478) + 6.128E-05*(19.44327676240209*z(1167)+238.8415469973891*z(1147)-120.5957898172324*z(1157)-
232.8573759791123*z(166)*z(1147)-232.8573759791123*z(172)*z(1157)-232.8573759791123*z(173)*z(1167)-1.221507832898173*z(168)*z(1147)-
1.221507832898173*z(174)*z(1157)-1.221507832898173*z(175)*z(1167)-z(170)*z(1147)-z(176)*z(1157)-z(177)*z(1167))*z(482) +
```

0.000439106*(1.522912463049924*z(136)+15.07109900570705*z(129)*z(135)+15.07109900570705*z(130)*z(136)+15.07109900570705*z(131)*z(13
7)-12.16291738213552*z(135)-2.050165563667998*z(137)-6.399042600192209*z(132)*z(135)-6.399042600192209*z(133)*z(136)-
6.399042600192209*z(134)*z(137)-z(135)^2-z(136)^2-z(137)^2)*z(470) + 0.00334198*(2.523796072986673*z(1167)+2.541448183412228*z(1147)-
z(1157)-2.199101730112089*z(166)*z(1147)-2.199101730112089*z(172)*z(1157)-2.199101730112089*z(173)*z(1167)-
2.178312856450368*z(170)*z(1147)-2.178312856450368*z(176)*z(1157)-2.178312856450368*z(177)*z(1167)-1.513417495017923*z(168)*z(1147)-
1.513417495017923*z(174)*z(1157)-1.513417495017923*z(175)*z(1167))*z(514) +
0.00353959*(2.051998112775774*z(137)+2.135415683737382*z(135)-z(136)-2.270871202596911*z(132)*z(135)-
2.270871202596911*z(133)*z(136)-2.270871202596911*z(134)*z(137)-2.253045691732658*z(135)^2-2.253045691732658*z(136)^2-
2.253045691732658*z(137)^2-1.758446882266025*z(129)*z(135)-1.758446882266025*z(130)*z(136)-1.758446882266025*z(131)*z(137))*z(518) +
0.005250594000000001*(1.3792740402324*z(1802)+1.576289654084852*z(1770)-1.220873295478569*z(1786)-1.749405305380686*z(207)*z(1770)-
1.749405305380686*z(208)*z(1786)-1.749405305380686*z(209)*z(1802)-1.677448684853561*z(213)*z(1770)-1.677448684853561*z(214)*z(1786)-
1.677448684853561*z(215)*z(1802)-z(210)*z(1770)-z(211)*z(1786)-z(212)*z(1802))*z(510) - 1.4*SW*z(118) -
9.800000000000001*SW*(z(2790)*z(2816)+z(2791)*z(2841)+z(2792)*z(2866)) - 2.8*SW*(z(284)*z(2393)+z(286)*z(2415)+z(288)*z(2437)) -
1.4*SW*(z(121)*z(135)+z(125)*z(136)+z(126)*z(137)) - 1.4*SW*(z(199)*z(1770)+z(203)*z(1786)+z(204)*z(1802)) -
1.4*SW*(z(20)*z(1147)+z(194)*z(1157)-z(196)*z(1167)) - 0.003268181*(-
2.730118068736096+z(84)+1.999895354633051*z(90)+2.575607348552605*z(87))*z(522) -
0.001989028*(3.454209794935014+z(84)+4.239214832571488*z(87)-3.642764204425479*z(90))*z(506) - 0.00021415*(-
4.618921316833995+64.04720989960309*z(84)-10.35433107634836*z(87)-z(90))*z(490) - 4.9681E-
05*(8.034661138060828*z(1167)+396.0548298142147*z(1147)+z(168)*z(1147)+z(174)*z(1157)+z(175)*z(1167)-19.09315432459089*z(1157)-
410.7868199110324*z(166)*z(1147)-410.7868199110324*z(172)*z(1157)-410.7868199110324*z(173)*z(1167)-25.99182786175802*z(170)*z(1147)-
25.99182786175802*z(176)*z(1157)-25.99182786175802*z(177)*z(1167))*z(450) -
0.00508017*(z(1786)+1.575266182037215*z(1802)+1.26191682561804*z(210)*z(1770)+1.26191682561804*z(211)*z(1786)+1.26191682561804*z(2
12)*z(1802)+1.738918973183969*z(207)*z(1770)+1.738918973183969*z(208)*z(1786)+1.738918973183969*z(209)*z(1802)-
1.564399813392072*z(1770)-1.445018572213135*z(213)*z(1770)-1.445018572213135*z(214)*z(1786)-1.445018572213135*z(215)*z(1802))*z(494)
-
0.00404624*(z(136)+2.01888419866345*z(137)+1.135663974455297*z(132)*z(135)+1.135663974455297*z(133)*z(136)+1.135663974455297*z(134)
*z(137)+1.613396387757523*z(129)*z(135)+1.613396387757523*z(130)*z(136)+1.613396387757523*z(131)*z(137)-1.85913242911938*z(135)-
1.514074795365574*z(135)^2-1.514074795365574*z(136)^2-1.514074795365574*z(137)^2)*z(502) -
0.00297212*(z(1157)+2.289867165524945*z(1167)+1.384676594484745*z(168)*z(1147)+1.384676594484745*z(174)*z(1157)+1.384676594484745*
z(175)*z(1167)+2.471453709809833*z(166)*z(1147)+2.471453709809833*z(172)*z(1157)+2.471453709809833*z(173)*z(1167)-
2.986921120277781*z(1147)-2.589087923771584*z(170)*z(1147)-2.589087923771584*z(176)*z(1157)-2.589087923771584*z(177)*z(1167))*z(498)
- (2.0E-
07*z(1802)+0.0210342*z(1770)+0.0009485670000000001*z(210)*z(1770)+0.0009485670000000001*z(211)*z(1786)+0.0009485670000000001*z(212
)*z(1802)-0.001343886*z(1786)-0.0196764*z(207)*z(1770)-0.0196764*z(208)*z(1786)-0.0196764*z(209)*z(1802)-0.00039917*z(213)*z(1770)-
0.00039917*z(214)*z(1786)-0.00039917*z(215)*z(1802))*z(446) -
0.000156039*(34.63294432802056*z(1147)+4.285595267849704*z(168)*z(1147)+4.285595267849704*z(174)*z(1157)+4.285595267849704*z(175)*
z(1167)-6.601112542377225*z(1157)-z(1167)-34.22740468728971*z(166)*z(1147)-34.22740468728971*z(172)*z(1157)-
34.22740468728971*z(173)*z(1167)-5.76932689904447*z(170)*z(1147)-5.76932689904447*z(176)*z(1157)-
5.76932689904447*z(177)*z(1167))*z(466);
z(4387) = SW*(z(34)*z(2393)*z(4190)-z(3598)-7*z(2816)*z(4370)-7*z(2841)*z(4372)-7*z(2866)*z(4371)-z(135)*z(3651)-z(136)*z(3653)-
z(137)*z(3652)-z(1147)*z(3741)-z(1157)*z(3743)-z(1167)*z(3742)-z(1770)*z(3867)-z(1786)*z(3869)-z(1802)*z(3868)-z(2393)*z(4020)-
z(2393)*z(4191)-z(2415)*z(4022)-z(2415)*z(4193)-z(2437)*z(4021)-z(2437)*z(4192)-z(250)*z(2415)*z(4190)-z(251)*z(2437)*z(4190)-
z(283)*z(2393)*z(4162)-z(284)*z(2393)*z(4176)-z(285)*z(2415)*z(4162)-z(286)*z(2415)*z(4176)-z(287)*z(2437)*z(4162)-z(288)*z(2437)*z(4176));
z(4411) = z(3060) + 0.1428571428571429*z(4387);
z(3051) = z(119)*z(135) + z(123)*z(136) + z(124)*z(137);
z(3052) = z(121)*z(135) + z(125)*z(136) + z(126)*z(137);
z(3053) = z(127)*z(136) + z(128)*z(137) - z(16)*z(135);
z(3054) = z(19)*z(1147) + z(195)*z(1167) - z(193)*z(1157);
z(3055) = z(20)*z(1147) + z(194)*z(1157) - z(196)*z(1167);
z(3056) = z(23)*z(1167) + z(24)*z(1157);
z(3058) = z(199)*z(1770) + z(203)*z(1786) + z(204)*z(1802);
z(3059) = z(205)*z(1786) + z(206)*z(1802) - z(28)*z(1770);
z(5075) = z(117)/z(113);
z(3037) = z(197)*z(1768) + z(201)*z(1784) + z(202)*z(1800);
z(3040) = 0.0009986300000000001*(-21.7338754093107 5+26.49359622683075*z(82)-1.944648167990146*z(85)-z(88))*z(458) + 0.001049445*(-
6.306009366855816+12.46668477147445*z(82)-2.21727579815998*z(85)-z(88))*z(474) + 4.7299E-
05*(z(1800)+44.21104040254551*z(1784)+114.2537897207129*z(207)*z(1768)+114.2537897207129*z(208)*z(1784)+114.2537897207129*z(209)*z(
1800)-147.8697224042791*z(1768)-21.77701431319901*z(210)*z(1768)-21.77701431319901*z(211)*z(1784)-21.77701431319901*z(212)*z(1800)-
3.298991522019493*z(213)*z(1768)-3.298991522019493*z(214)*z(1784)-3.298991522019493*z(215)*z(1800))*z(462) + 5.641700000000001E-
05*(z(131)+247.6572664267862*z(129)+24.25456865838311*z(129)*z(132)+24.25456865838311*z(130)*z(133)+24.25456865838311*z(131)*z(134)
-149.1704628037648*z(130)-269.1738305829803*z(129)^2-269.1738305829803*z(130)^2-269.1738305829803*z(131)^2-
14.71584805998192*z(129)*z(135)-14.71584805998192*z(130)*z(136)-14.71584805998192*z(131)*z(137))*z(486) +
0.000263142*(z(1800)+57.96163288262611*z(1768)-37.04060925279887*z(1784)-58.17953804409786*z(207)*z(1768)-
58.17953804409786*z(208)*z(1784)-58.17953804409786*z(209)*z(1800)-3.892632114979745*z(213)*z(1768)-3.892632114979745*z(214)*z(1784)-
3.892632114979745*z(215)*z(1800)-3.889998555912777*z(210)*z(1768)-3.889998555912777*z(211)*z(1784)-
3.889998555912777*z(212)*z(1800))*z(478) + 6.128E-05*(19.44327676240209*z(1165)+238.8415469973891*z(1145)-120.5957898172324*z(1155)-
232.8573759791123*z(166)*z(1145)-232.8573759791123*z(172)*z(1155)-232.8573759791123*z(173)*z(1165)-1.221507832898173*z(168)*z(1145)-
1.221507832898173*z(174)*z(1155)-1.221507832898173*z(175)*z(1165)-z(170)*z(1145)-z(176)*z(1155)-z(177)*z(1165))*z(482) +
0.00334198*(2.523796072986673*z(1165)+2.541448183412228*z(1145)-z(1155)-2.199101730112089*z(166)*z(1145)-
2.199101730112089*z(172)*z(1155)-2.199101730112089*z(173)*z(1165)-2.178312856450368*z(170)*z(1145)-2.178312856450368*z(176)*z(1155)-
2.178312856450368*z(177)*z(1165)-1.513417495017923*z(168)*z(1145)-1.513417495017923*z(174)*z(1155)-
1.513417495017923*z(175)*z(1165))*z(514) + 0.00353959*(2.051998112775774*z(131)+2.135415683737382*z(129)-z(130)-
2.270871202596911*z(129)*z(132)-2.270871202596911*z(130)*z(133)-2.270871202596911*z(131)*z(134)-2.253045691732658*z(129)*z(135)-
2.253045691732658*z(130)*z(136)-2.253045691732658*z(131)*z(137)-1.758446882266025*z(129)^2-1.758446882266025*z(130)^2-
1.758446882266025*z(131)^2)*z(518) +
0.00404624*(1.85913242911938*z(129)+1.514074795365574*z(129)*z(135)+1.514074795365574*z(130)*z(136)+1.514074795365574*z(131)*z(137)
-2.01888419866345*z(131)-z(130)-1.613396387757523*z(129)^2-1.613396387757523*z(130)^2-1.613396387757523*z(131)^2-
1.135663974455297*z(129)*z(132)-1.135663974455297*z(130)*z(133)-1.135663974455297*z(131)*z(134))*z(502) +

0.005250594000000001*(1.3792740402324*z(1800)+1.576289654084852*z(1768)-1.220873295478569*z(1784)-1.749405305380686*z(207)*z(1768)-
1.749405305380686*z(208)*z(1784)-1.749405305380686*z(209)*z(1800)-1.677448684853561*z(213)*z(1768)-1.677448684853561*z(214)*z(1784)-
1.677448684853561*z(215)*z(1800)-z(210)*z(1768)-z(211)*z(1784)-z(212)*z(1800))*z(510) - 1.4*SW*z(114) -
9.800000000000001*SW*(z(2790)*z(2814)+z(2791)*z(2839)+z(2792)*z(2864)) - 2.8*SW*(z(284)*z(2391)+z(286)*z(2413)+z(288)*z(2435)) -
1.4*SW*(z(121)*z(129)+z(125)*z(130)+z(126)*z(131)) - 1.4*SW*(z(199)*z(1768)+z(203)*z(1784)+z(204)*z(1800)) -
1.4*SW*(z(20)*z(1145)+z(194)*z(1155)-z(196)*z(1165)) - 0.003268181*(-
2.228608513420769+z(82)+1.999895354633051*z(88)+2.575607348552605*z(85))*z(522) - 0.001989028*(-
3.484725705218831+z(82)+4.239214832571488*z(85)-3.642764204425479*z(88))*z(506) - 0.00021415*(-
70.09441979920617+64.04720989960309*z(82)-10.35433107634836*z(85)-z(88))*z(490) -
0.000439106*(2.050165563667998*z(131)+12.16291738213552*z(129)+z(129)*z(135)+z(130)*z(136)+z(131)*z(137)+6.399042600192209*z(129)*z(
132)+6.399042600192209*z(130)*z(133)+6.399042600192209*z(131)*z(134)-1.522912463049924*z(130)-15.07109900570705*z(129)^2-
15.07109900570705*z(130)^2-15.07109900570705*z(131)^2)*z(470) - 4.9681E-
05*(8.034661138060828*z(1165)+396.0548298142147*z(1145)+z(168)*z(1145)+z(174)*z(1155)+z(175)*z(1165)-19.09315432459089*z(1155)-
410.7868199110324*z(166)*z(1145)-410.7868199110324*z(172)*z(1155)-410.7868199110324*z(173)*z(1165)-25.99182786175802*z(170)*z(1145)-
25.99182786175802*z(176)*z(1155)-25.99182786175802*z(177)*z(1165))*z(450) -
0.00508017*(z(1784)+1.575266182037215*z(1800)+1.26191682561804*z(210)*z(1768)+1.26191682561804*z(211)*z(1784)+1.26191682561804*z(2
12)*z(1800)+1.738918973183969*z(207)*z(1768)+1.738918973183969*z(208)*z(1784)+1.738918973183969*z(209)*z(1800)-
1.564399813392072*z(1768)-1.445018572213135*z(213)*z(1768)-1.445018572213135*z(214)*z(1784)-1.445018572213135*z(215)*z(1800))*z(494)
-
0.00297212*(z(1155)+2.289867165524945*z(1165)+1.384676594484745*z(168)*z(1145)+1.384676594484745*z(174)*z(1155)+1.384676594484745*
z(175)*z(1165)+2.471453709809833*z(166)*z(1145)+2.471453709809833*z(172)*z(1155)+2.471453709809833*z(173)*z(1165)-
2.986921120277781*z(1145)-2.589087923771584*z(170)*z(1145)-2.589087923771584*z(176)*z(1155)-2.589087923771584*z(177)*z(1165))*z(498)
- (2.0E-
07*z(1800)+0.0210342*z(1768)+0.000948567000000000001*z(210)*z(1768)+0.000948567000000000001*z(211)*z(1784)+0.000948567000000000001*z(212
)*z(1800)-0.001343886*z(1784)-0.0196764*z(207)*z(1768)-0.0196764*z(208)*z(1784)-0.0196764*z(209)*z(1800)-0.00039917*z(213)*z(1768)-
0.00039917*z(214)*z(1784)-0.00039917*z(215)*z(1800))*z(446) -
0.000156039*(34.63294432802056*z(1145)+4.285595267849704*z(168)*z(1145)+4.285595267849704*z(174)*z(1155)+4.285595267849704*z(175)*
z(1165)-6.601112542377225*z(1155)-z(1165)-34.22740468728971*z(166)*z(1145)-34.22740468728971*z(172)*z(1155)-
34.22740468728971*z(173)*z(1165)-5.76932689904447*z(170)*z(1145)-5.76932689904447*z(176)*z(1155)-
5.76932689904447*z(177)*z(1165))*z(466) - 4.9681E-
05*(25.99182786175802*z(131)+410.7868199110324*z(129)+4.101447233348765*z(129)*z(135)+4.101447233348765*z(130)*z(136)+4.1014472333
48765*z(131)*z(137)+23.60302731426501*z(129)*z(132)+23.60302731426501*z(130)*z(133)+23.60302731426501*z(131)*z(134)-z(130)-
436.869225659709*z(129)^2-436.869225659709*z(130)^2-436.869225659709*z(131)^2)*z(454);
z(4385) = SW*(z(34)*z(2391)*z(4190)-z(3597)-7*z(2814)*z(4370)-7*z(2839)*z(4372)-7*z(2864)*z(4371)-z(129)*z(3651)-z(130)*z(3653)-
z(131)*z(3652)-z(1145)*z(3741)-z(1155)*z(3743)-z(1165)*z(3742)-z(1768)*z(3867)-z(1784)*z(3869)-z(1800)*z(3868)-z(2391)*z(4020)-
z(2391)*z(4191)-z(2413)*z(4022)-z(2413)*z(4193)-z(2435)*z(4021)-z(2435)*z(4192)-z(250)*z(2413)*z(4190)-z(251)*z(2435)*z(4190)-
z(283)*z(2391)*z(4162)-z(284)*z(2391)*z(4176)-z(285)*z(2413)*z(4162)-z(286)*z(2413)*z(4176)-z(287)*z(2435)*z(4162)-z(288)*z(2435)*z(4176));
z(4409) = z(3040) + 0.1428571428571429*z(4385);
z(3031) = z(119)*z(129) + z(123)*z(130) + z(124)*z(131);
z(3032) = z(121)*z(129) + z(125)*z(130) + z(126)*z(131);
z(3033) = z(127)*z(130) + z(128)*z(131) - z(16)*z(129);
z(3034) = z(19)*z(1145) + z(195)*z(1165) - z(193)*z(1155);
z(3035) = z(20)*z(1145) + z(194)*z(1155) - z(196)*z(1165);
z(3036) = z(23)*z(1165) + z(24)*z(1155);
z(3038) = z(199)*z(1768) + z(203)*z(1784) + z(204)*z(1800);
z(3039) = z(205)*z(1784) + z(206)*z(1800) - z(28)*z(1768);
z(5078) = z(3057)*z(5070) - z(4411) - z(81)*z(5059) - z(117)*z(5061) - z(118)*z(5060) - z(3051)*z(5064) - z(3052)*z(5063) - z(3053)*z(5062) -
z(3054)*z(5067) - z(3055)*z(5066) - z(3056)*z(5065) - z(3058)*z(5069) - z(3059)*z(5068) - z(5075)*(z(10)*z(5059)+z(3037)*z(5070)-z(4409)-
z(113)*z(5061)-z(114)*z(5060)-z(3031)*z(5064)-z(3032)*z(5063)-z(3033)*z(5062)-z(3034)*z(5067)-z(3035)*z(5066)-z(3036)*z(5065)-
z(3038)*z(5069)-z(3039)*z(5068));
z(5076) = z(118) - z(114)*z(5075);
z(5071) = z(115)/z(113);
z(5072) = z(116) - z(114)*z(5071);
z(5079) = z(5076)/z(5072);
z(3047) = z(197)*z(1769) + z(201)*z(1785) + z(202)*z(1801);
z(3050) = 0.0009986300000000001*(1.174230696053593+26.49359622683075*z(83)-1.944648167990146*z(86)-z(89))*z(458) +
0.001049445*(2.677470472487839+12.46668477147445*z(83)-2.21727579815998*z(86)-z(89))*z(474) + 4.7299E-
05*(z(1801)+44.21104040254551*z(1785)+114.2537897207129*z(207)*z(1769)+114.2537897207129*z(208)*z(1785)+114.2537897207129*z(209)*z(
1801)-147.8697224042791*z(1769)-21.77701431319901*z(210)*z(1769)-21.77701431319901*z(211)*z(1785)-21.77701431319901*z(212)*z(1801)-
3.298991522019493*z(213)*z(1769)-3.298991522019493*z(214)*z(1785)-3.298991522019493*z(215)*z(1801))*z(462) + 4.9681E-
05*(z(133)+436.869225659709*z(129)*z(132)+436.869225659709*z(130)*z(133)+436.869225659709*z(131)*z(134)-410.7868199110324*z(132)-
25.99182786175802*z(134)-23.60302731426501*z(132)^2-23.60302731426501*z(133)^2-23.60302731426501*z(134)^2-
4.101447233348765*z(132)*z(135)-4.101447233348765*z(133)*z(136)-4.101447233348765*z(134)*z(137))*z(454) +
0.000263142*(z(1801)+57.96163288262611*z(1769)-37.04060925279887*z(1785)-58.17953804409786*z(207)*z(1769)-
58.17953804409786*z(208)*z(1785)-58.17953804409786*z(209)*z(1801)-3.892632114979745*z(213)*z(1769)-3.892632114979745*z(214)*z(1785)-
3.892632114979745*z(215)*z(1801)-3.889998555912777*z(210)*z(1769)-3.889998555912777*z(211)*z(1785)-
3.889998555912777*z(212)*z(1801))*z(478) + 6.128E-05*(19.44327676240209*z(1166)+238.8415469973891*z(1146)-120.5957898172324*z(1156)-
232.8573759791123*z(166)*z(1146)-232.8573759791123*z(172)*z(1156)-232.8573759791123*z(173)*z(1166)-1.221507832898173*z(168)*z(1146)-
1.221507832898173*z(174)*z(1156)-1.221507832898173*z(175)*z(1166)-z(170)*z(1146)-z(176)*z(1156)-z(177)*z(1166))*z(482) +
0.000439106*(1.522912463049924*z(133)+15.07109900570705*z(129)*z(132)+15.07109900570705*z(130)*z(133)+15.07109900570705*z(131)*z(13
4)-12.16291738213552*z(132)-2.050165563667998*z(134)-6.399042600192209*z(132)^2-6.399042600192209*z(133)^2-
6.399042600192209*z(134)^2-z(132)*z(135)-z(133)*z(136)-z(134)*z(137))*z(470) +
0.00334198*(2.523796072986673*z(1166)+2.541448183412228*z(1146)-z(1156)-2.199101730112089*z(166)*z(1146)-
2.199101730112089*z(172)*z(1156)-2.199101730112089*z(173)*z(1166)-2.178312856450368*z(170)*z(1146)-2.178312856450368*z(176)*z(1156)-
2.178312856450368*z(177)*z(1166)-1.513417495017923*z(168)*z(1146)-1.513417495017923*z(174)*z(1156)-
1.513417495017923*z(175)*z(1166))*z(514) + 0.00353959*(2.051998112775774*z(134)+2.135415683737382*z(132)-z(133)-
2.270871202596911*z(132)^2-2.270871202596911*z(133)^2-2.270871202596911*z(134)^2-2.253045691732658*z(132)*z(135)-
2.253045691732658*z(133)*z(136)-2.253045691732658*z(134)*z(137)-1.758446882266025*z(129)*z(132)-1.758446882266025*z(130)*z(133)-

1.758446882266025*z(131)*z(134))*z(518) +
0.00404624*(1.85913242911938*z(132)+1.514074795365574*z(132)*z(135)+1.514074795365574*z(133)*z(136)+1.514074795365574*z(134)*z(137)
-2.01888419866345*z(134)-z(133)-1.613396387757523*z(129)*z(132)-1.613396387757523*z(130)*z(133)-1.613396387757523*z(131)*z(134)-
1.135663974455297*z(132)^2-1.135663974455297*z(133)^2-1.135663974455297*z(134)^2)*z(502) +
0.005250594000000001*(1.3792740402324*z(1801)+1.576289654084852*z(1769)-1.220873295478569*z(1785)-1.749405305380686*z(207)*z(1769)-
1.749405305380686*z(208)*z(1785)-1.749405305380686*z(209)*z(1801)-1.677448684853561*z(213)*z(1769)-1.677448684853561*z(214)*z(1785)-
1.677448684853561*z(215)*z(1801)-z(210)*z(1769)-z(211)*z(1785)-z(212)*z(1801))*z(510) - 1.4*SW*z(116) -
9.800000000000001*SW*(z(2790)*z(2815)+z(2791)*z(2840)+z(2792)*z(2865)) - 2.8*SW*(z(284)*z(2392)+z(286)*z(2414)+z(288)*z(2436)) -
1.4*SW*(z(121)*z(132)+z(125)*z(133)+z(126)*z(134)) - 1.4*SW*(z(199)*z(1769)+z(203)*z(1785)+z(204)*z(1801)) -
1.4*SW*(z(20)*z(1146)+z(194)*z(1156)-z(196)*z(1166)) -
0.003268181*(1.290837318985699+z(83)+1.999895354633051*z(89)+2.575607348552605*z(86))*z(522) -
0.001989028*(2.026969957185118+z(83)+4.239214832571488*z(86)-3.642764204425479*z(89))*z(506) -
0.00021415*(40.8116273639972+64.04720989960309*z(83)-10.35433107634836*z(86)-z(89))*z(490) - 4.9681E-
05*(8.034661138060828*z(1166)+396.0548298142147*z(1146)+z(168)*z(1146)+z(174)*z(1156)+z(175)*z(1166)-19.09315432459089*z(1156)-
410.7868199110324*z(166)*z(1146)-410.7868199110324*z(172)*z(1156)-410.7868199110324*z(173)*z(1166)-25.99182786175802*z(170)*z(1146)-
25.99182786175802*z(176)*z(1156)-25.99182786175802*z(177)*z(1166))*z(450) -
0.00508017*(z(1785)+1.575266182037215*z(1801)+1.26191682561804*z(210)*z(1769)+1.26191682561804*z(211)*z(1785)+1.26191682561804*z(2
12)*z(1801)+1.738918973183969*z(207)*z(1769)+1.738918973183969*z(208)*z(1785)+1.738918973183969*z(209)*z(1801)-
1.564399813392072*z(1769)-1.445018572213135*z(213)*z(1769)-1.445018572213135*z(214)*z(1785)-1.445018572213135*z(215)*z(1801))*z(494)
-
0.00297212*(z(1156)+2.289867165524945*z(1166)+1.384676594484745*z(168)*z(1146)+1.384676594484745*z(174)*z(1156)+1.384676594484745*
z(175)*z(1166)+2.471453709809833*z(166)*z(1146)+2.471453709809833*z(172)*z(1156)+2.471453709809833*z(173)*z(1166)-
2.986921120277781*z(1146)-2.589087923771584*z(170)*z(1146)-2.589087923771584*z(176)*z(1156)-2.589087923771584*z(177)*z(1166))*z(498)
- (2.0E-
07*z(1801)+0.0210342*z(1769)+0.000948567000000000001*z(210)*z(1769)+0.000948567000000000001*z(211)*z(1785)+0.000948567000000000001*z(212
)*z(1801)-0.001343886*z(1785)-0.0196764*z(207)*z(1769)-0.0196764*z(208)*z(1785)-0.0196764*z(209)*z(1801)-0.00039917*z(213)*z(1769)-
0.00039917*z(214)*z(1785)-0.00039917*z(215)*z(1801))*z(446) -
0.000156039*(34.63294432802056*z(1146)+4.285595267849704*z(168)*z(1146)+4.285595267849704*z(174)*z(1156)+4.285595267849704*z(175)*
z(1166)-6.601112542377225*z(1156)-z(1166)-34.22740468728971*z(166)*z(1146)-34.22740468728971*z(172)*z(1156)-
34.22740468728971*z(173)*z(1166)-5.76932689904447*z(170)*z(1146)-5.76932689904447*z(176)*z(1156)-
5.76932689904447*z(177)*z(1166))*z(466) - 5.641700000000001E-
05*(149.1704628037648*z(133)+14.71584805998192*z(132)*z(135)+14.71584805998192*z(133)*z(136)+14.71584805998192*z(134)*z(137)+269.17
38305829803*z(129)*z(132)+269.1738305829803*z(130)*z(133)+269.1738305829803*z(131)*z(134)-247.6572664267862*z(132)-z(134)-
24.25456865838311*z(132)^2-24.25456865838311*z(133)^2-24.25456865838311*z(134)^2)*z(486);
z(4386) = SW*(z(34)*z(2392)*z(4190)-z(3599)-7*z(2815)*z(4370)-7*z(2840)*z(4372)-7*z(2865)*z(4371)-z(132)*z(3651)-z(133)*z(3653)-
z(134)*z(3652)-z(1146)*z(3741)-z(1156)*z(3743)-z(1166)*z(3742)-z(1769)*z(3867)-z(1785)*z(3869)-z(1801)*z(3868)-z(2392)*z(4020)-
z(2392)*z(4191)-z(2414)*z(4022)-z(2414)*z(4193)-z(2436)*z(4021)-z(2436)*z(4192)-z(250)*z(2414)*z(4190)-z(251)*z(2436)*z(4190)-
z(283)*z(2392)*z(4162)-z(284)*z(2392)*z(4176)-z(285)*z(2414)*z(4162)-z(286)*z(2414)*z(4176)-z(287)*z(2436)*z(4162)-z(288)*z(2436)*z(4176));
z(4410) = z(3050) + 0.1428571428571429*z(4386);
z(3041) = z(119)*z(132) + z(123)*z(133) + z(124)*z(134);
z(3042) = z(121)*z(132) + z(125)*z(133) + z(126)*z(134);
z(3043) = z(127)*z(133) + z(128)*z(134) - z(16)*z(132);
z(3044) = z(19)*z(1146) + z(195)*z(1166) - z(193)*z(1156);
z(3045) = z(20)*z(1146) + z(194)*z(1156) - z(196)*z(1166);
z(3046) = z(23)*z(1166) + z(24)*z(1156);
z(3048) = z(199)*z(1769) + z(203)*z(1785) + z(204)*z(1801);
z(3049) = z(205)*z(1785) + z(206)*z(1801) - z(28)*z(1769);
z(5074) = z(3047)*z(5070) - z(4410) - z(80)*z(5059) - z(115)*z(5061) - z(116)*z(5060) - z(3041)*z(5064) - z(3042)*z(5063) - z(3043)*z(5062) -
z(3044)*z(5067) - z(3045)*z(5066) - z(3046)*z(5065) - z(3048)*z(5069) - z(3049)*z(5068) - z(5071)*(z(10)*z(5059)+z(3037)*z(5070)-z(4409)-
z(113)*z(5061)-z(114)*z(5060)-z(3031)*z(5064)-z(3032)*z(5063)-z(3033)*z(5062)-z(3034)*z(5067)-z(3035)*z(5066)-z(3036)*z(5065)-
z(3038)*z(5069)-z(3039)*z(5068));
z(5081) = z(5078) - z(5079)*z(5074);
z(5077) = z(81) + z(10)*z(5075);
z(5073) = z(80) + z(10)*z(5071);
z(5080) = z(5077) - z(5073)*z(5079);
z(5082) = z(5081)/z(5080);
z(5083) = (z(5074)-z(5073)*z(5082))/z(5072);
z(5084) = (z(10)*z(5059)+z(10)*z(5082)+z(3037)*z(5070)-z(4409)-z(114)*z(5060)-z(114)*z(5083)-z(3031)*z(5064)-z(3032)*z(5063)-
z(3033)*z(5062)-z(3034)*z(5067)-z(3035)*z(5066)-z(3036)*z(5065)-z(3038)*z(5069)-z(3039)*z(5068))/z(113) - z(5061);
FAB1 = z(5084);
FAB2 = z(5083);
FAB3 = z(5082);
z(3078) = z(197)*z(1767) + z(201)*z(1783) + z(202)*z(1799);
z(3081) = 5.641700000000001E-05*(1+24.25456865838311*z(134)-269.1738305829803*z(131)-14.71584805998192*z(137))*z(486) +
0.000203764*(-6.337233269861212+106.5158713020946*z(131)-5.754804577844959*z(134)-z(137))*z(454) + 0.000439106*(-
2.050165563667998+15.07109900570705*z(131)-6.399042600192209*z(134)-z(137))*z(470) + 4.7299E-
05*(z(1799)+44.21104040254551*z(1783)+114.2537897207129*z(207)*z(1767)+114.2537897207129*z(208)*z(1783)+114.2537897207129*z(209)*z(
1799)-147.8697224042791*z(1767)-21.77701431319901*z(210)*z(1767)-21.77701431319901*z(211)*z(1783)-21.77701431319901*z(212)*z(1799)-
3.298991522019493*z(213)*z(1767)-3.298991522019493*z(214)*z(1783)-3.298991522019493*z(215)*z(1799))*z(462) +
0.000263142*(z(1799)+57.96163288262611*z(1767)-37.04060925279887*z(1783)-58.17953804409786*z(207)*z(1767)-
58.17953804409786*z(208)*z(1783)-58.17953804409786*z(209)*z(1799)-3.892632114979745*z(213)*z(1767)-3.892632114979745*z(214)*z(1783)-
3.892632114979745*z(215)*z(1799)-3.889998555912777*z(210)*z(1767)-3.889998555912777*z(211)*z(1783)-
3.889998555912777*z(212)*z(1799))*z(478) + 6.128E-05*(19.44327676240209*z(177)+238.8415469973891*z(170)-120.5957898172324*z(176)-
232.8573759791123*z(166)*z(170)-232.8573759791123*z(172)*z(176)-232.8573759791123*z(173)*z(177)-1.221507832898173*z(168)*z(170)-
1.221507832898173*z(174)*z(176)-1.221507832898173*z(175)*z(177)-z(170)^2-z(176)^2-z(177)^2)*z(482) +
0.00334198*(2.523796072986673*z(177)+2.541448183412228*z(170)-z(176)-2.199101730112089*z(166)*z(170)-
2.199101730112089*z(172)*z(176)-2.199101730112089*z(173)*z(177)-2.178312856450368*z(170)^2-2.178312856450368*z(176)^2-
2.178312856450368*z(177)^2-1.513417495017923*z(168)*z(170)-1.513417495017923*z(174)*z(176)-1.513417495017923*z(175)*z(177))*z(514) +

0.005250594000000001*(1.3792740402324*z(1799)+1.576289654084852*z(1767)-1.220873295478569*z(1783)-1.749405305380686*z(207)*z(1767)-
1.749405305380686*z(208)*z(1783)-1.749405305380686*z(209)*z(1799)-1.677448684853561*z(213)*z(1767)-1.677448684853561*z(214)*z(1783)-
1.677448684853561*z(215)*z(1799)-z(210)*z(1767)-z(211)*z(1783)-z(212)*z(1799))*z(510) - 1.4*SW*z(126) -
9.800000000000001*SW*(z(2790)*z(2813)+z(2791)*z(2838)+z(2792)*z(2863)) - 2.8*SW*(z(284)*z(2390)+z(286)*z(2412)+z(288)*z(2434)) -
1.4*SW*(z(199)*z(1767)+z(203)*z(1783)+z(204)*z(1799)) - 1.4*SW*(z(20)*z(170)+z(176)*z(194)-z(177)*z(196)) - 0.006224181*(-
1.166937786674263+z(131)+1.281270258689456*z(137)+1.291407335535866*z(134))*z(518) -
0.004595169000000001*(1.777712636901929+z(134)+1.420663527282674*z(131)-1.333206678579177*z(137))*z(502) - 4.9681E-
05*(8.034661138060828*z(177)+396.0548298142147*z(170)+z(168)*z(170)+z(174)*z(176)+z(175)*z(177)-19.09315432459089*z(176)-
410.7868199110324*z(166)*z(170)-410.7868199110324*z(172)*z(176)-410.7868199110324*z(173)*z(177)-25.99182786175802*z(170)^2-
25.99182786175802*z(176)^2-25.99182786175802*z(177)^2)*z(450) -
0.00508017*(z(1783)+1.575266182037215*z(1799)+1.26191682561804*z(210)*z(1767)+1.26191682561804*z(211)*z(1783)+1.26191682561804*z(2
12)*z(1799)+1.738918973183969*z(207)*z(1767)+1.738918973183969*z(208)*z(1783)+1.738918973183969*z(209)*z(1799)-
1.564399813392072*z(1767)-1.445018572213135*z(213)*z(1767)-1.445018572213135*z(214)*z(1783)-1.445018572213135*z(215)*z(1799))*z(494)
-
0.00297212*(z(176)+2.289867165524945*z(177)+1.384676594484745*z(168)*z(170)+1.384676594484745*z(174)*z(176)+1.384676594484745*z(17
5)*z(177)+2.471453709809833*z(166)*z(170)+2.471453709809833*z(172)*z(176)+2.471453709809833*z(173)*z(177)-2.986921120277781*z(170)-
2.589087923771584*z(170)^2-2.589087923771584*z(176)^2-2.589087923771584*z(177)^2)*z(498) - (2.0E-
07*z(1799)+0.0210342*z(1767)+0.000948567000000000001*z(210)*z(1767)+0.000948567000000000001*z(211)*z(1783)+0.000948567000000000001*z(212
)*z(1799)-0.001343886*z(1783)-0.0196764*z(207)*z(1767)-0.0196764*z(208)*z(1783)-0.0196764*z(209)*z(1799)-0.00039917*z(213)*z(1767)-
0.00039917*z(214)*z(1783)-0.00039917*z(215)*z(1799))*z(446) -
0.000156039*(34.63294432802056*z(170)+4.285595267849704*z(168)*z(170)+4.285595267849704*z(174)*z(176)+4.285595267849704*z(175)*z(17
7)-6.601112542377225*z(176)-z(177)-34.22740468728971*z(166)*z(170)-34.22740468728971*z(172)*z(176)-34.22740468728971*z(173)*z(177)-
5.76932689904447*z(170)^2-5.76932689904447*z(176)^2-5.76932689904447*z(177)^2)*z(466);
z(4390) = SW*(z(34)*z(2390)*z(4190)-z(3652)-7*z(2813)*z(4370)-7*z(2838)*z(4372)-7*z(2863)*z(4371)-z(170)*z(3741)-z(176)*z(3743)-
z(177)*z(3742)-z(1767)*z(3867)-z(1783)*z(3869)-z(1799)*z(3868)-z(2390)*z(4020)-z(2390)*z(4191)-z(2412)*z(4022)-z(2412)*z(4193)-
z(2434)*z(4021)-z(2434)*z(4192)-z(250)*z(2412)*z(4190)-z(251)*z(2434)*z(4190)-z(283)*z(2390)*z(4162)-z(284)*z(2390)*z(4176)-
z(285)*z(2412)*z(4162)-z(286)*z(2412)*z(4176)-z(287)*z(2434)*z(4162)-z(288)*z(2434)*z(4176));
z(4414) = z(3081) + 0.1428571428571429*z(4390);
z(3075) = z(19)*z(170) + z(177)*z(195) - z(176)*z(193);
z(3076) = z(20)*z(170) + z(176)*z(194) - z(177)*z(196);
z(3077) = z(23)*z(177) + z(24)*z(176);
z(3079) = z(199)*z(1767) + z(203)*z(1783) + z(204)*z(1799);
z(3080) = z(205)*z(1783) + z(206)*z(1799) - z(28)*z(1767);
z(5089) = z(124)/z(119);
z(3064) = z(197)*z(1765) + z(201)*z(1781) + z(202)*z(1797);
z(3067) = 0.000203764*(-100.156553601166+106.5158713020946*z(129)-5.754804577844959*z(132)-z(135))*z(454) + 0.000439106*(-
12.16291738213552+15.07109900570705*z(129)-6.399042600192209*z(132)-z(135))*z(470) +
0.000830224*(16.82928944477635+1.648193740484496*z(132)-18.29142496482877*z(129)-z(135))*z(486) + 4.7299E-
05*(z(1797)+44.21104040254551*z(1781)+114.2537897207129*z(207)*z(1765)+114.2537897207129*z(208)*z(1781)+114.2537897207129*z(209)*z(
1797)-147.8697224042791*z(1765)-21.77701431319901*z(210)*z(1765)-21.77701431319901*z(211)*z(1781)-21.77701431319901*z(212)*z(1797)-
3.298991522019493*z(213)*z(1765)-3.298991522019493*z(214)*z(1781)-3.298991522019493*z(215)*z(1797))*z(462) +
0.000263142*(z(1797)+57.96163288262611*z(1765)-37.04060925279887*z(1781)-58.17953804409786*z(207)*z(1765)-
58.17953804409786*z(208)*z(1781)-58.17953804409786*z(209)*z(1797)-3.892632114979745*z(213)*z(1765)-3.892632114979745*z(214)*z(1781)-
3.892632114979745*z(215)*z(1797)-3.889998555912777*z(210)*z(1765)-3.889998555912777*z(211)*z(1781)-
3.889998555912777*z(212)*z(1797))*z(478) + 6.128E-05*(19.44327676240209*z(173)+238.8415469973891*z(166)-120.5957898172324*z(172)-
232.8573759791123*z(166)^2-232.8573759791123*z(172)^2-232.8573759791123*z(173)^2-1.221507832898173*z(166)*z(168)-
1.221507832898173*z(172)*z(174)-1.221507832898173*z(173)*z(175)-z(166)*z(170)-z(172)*z(176)-z(173)*z(177))*z(482) +
0.00297212*(2.986921120277781*z(166)+2.589087923771584*z(166)*z(170)+2.589087923771584*z(172)*z(176)+2.589087923771584*z(173)*z(177
)-2.289867165524945*z(173)-z(172)-2.471453709809833*z(166)^2-2.471453709809833*z(172)^2-2.471453709809833*z(173)^2-
1.384676594484745*z(166)*z(168)-1.384676594484745*z(172)*z(174)-1.384676594484745*z(173)*z(175))*z(498) +
0.00334198*(2.523796072986673*z(173)+2.541448183412228*z(166)-z(172)-2.199101730112089*z(166)^2-2.199101730112089*z(172)^2-
2.199101730112089*z(173)^2-2.178312856450368*z(166)*z(170)-2.178312856450368*z(172)*z(176)-2.178312856450368*z(173)*z(177)-
1.513417495017923*z(166)*z(168)-1.513417495017923*z(172)*z(174)-1.513417495017923*z(173)*z(175))*z(514) +
0.005250594000000001*(1.3792740402324*z(1797)+1.576289654084852*z(1765)-1.220873295478569*z(1781)-1.749405305380686*z(207)*z(1765)-
1.749405305380686*z(208)*z(1781)-1.749405305380686*z(209)*z(1797)-1.677448684853561*z(213)*z(1765)-1.677448684853561*z(214)*z(1781)-
1.677448684853561*z(215)*z(1797)-z(210)*z(1765)-z(211)*z(1781)-z(212)*z(1797))*z(510) - 1.4*SW*z(121) -
9.800000000000001*SW*(z(2790)*z(2811)+z(2791)*z(2836)+z(2792)*z(2861)) - 2.8*SW*(z(284)*z(2388)+z(286)*z(2410)+z(288)*z(2432)) -
1.4*SW*(z(199)*z(1765)+z(203)*z(1781)+z(204)*z(1797)) - 1.4*SW*(z(20)*z(166)+z(172)*z(194)-z(173)*z(196)) - 0.006224181*(-
1.214375995813746+z(129)+1.281270258689456*z(135)+1.291407335535866*z(132))*z(518) - 0.004595169000000001*(-
1.637044469963999+z(132)+1.420663527282674*z(129)-1.333206678579177*z(135))*z(502) - 4.9681E-
05*(8.034661138060828*z(173)+396.0548298142147*z(166)+z(166)*z(168)+z(172)*z(174)+z(173)*z(175)-19.09315432459089*z(172)-
410.7868199110324*z(166)^2-410.7868199110324*z(172)^2-410.7868199110324*z(173)^2-25.99182786175802*z(166)*z(170)-
25.99182786175802*z(172)*z(176)-25.99182786175802*z(173)*z(177))*z(450) -
0.00508017*(z(1781)+1.575266182037215*z(1797)+1.26191682561804*z(210)*z(1765)+1.26191682561804*z(211)*z(1781)+1.26191682561804*z(2
12)*z(1797)+1.738918973183969*z(207)*z(1765)+1.738918973183969*z(208)*z(1781)+1.738918973183969*z(209)*z(1797)-
1.564399813392072*z(1765)-1.445018572213135*z(213)*z(1765)-1.445018572213135*z(214)*z(1781)-1.445018572213135*z(215)*z(1797))*z(494)
- (2.0E-
07*z(1797)+0.0210342*z(1765)+0.000948567000000000001*z(210)*z(1765)+0.000948567000000000001*z(211)*z(1781)+0.000948567000000000001*z(212
)*z(1797)-0.001343886*z(1781)-0.0196764*z(207)*z(1765)-0.0196764*z(208)*z(1781)-0.0196764*z(209)*z(1797)-0.00039917*z(213)*z(1765)-
0.00039917*z(214)*z(1781)-0.00039917*z(215)*z(1797))*z(446) -
0.000156039*(34.63294432802056*z(166)+4.285595267849704*z(166)*z(168)+4.285595267849704*z(172)*z(174)+4.285595267849704*z(173)*z(17
5)-6.601112542377225*z(172)-z(173)-34.22740468728971*z(166)^2-34.22740468728971*z(172)^2-34.22740468728971*z(173)^2-
5.76932689904447*z(166)*z(170)-5.76932689904447*z(172)*z(176)-5.76932689904447*z(173)*z(177))*z(466);
z(4388) = SW*(z(34)*z(2388)*z(4190)-z(3651)-7*z(2811)*z(4370)-7*z(2836)*z(4372)-7*z(2861)*z(4371)-z(166)*z(3741)-z(172)*z(3743)-
z(173)*z(3742)-z(1765)*z(3867)-z(1781)*z(3869)-z(1797)*z(3868)-z(2388)*z(4020)-z(2388)*z(4191)-z(2410)*z(4022)-z(2410)*z(4193)-
z(2432)*z(4021)-z(2432)*z(4192)-z(250)*z(2410)*z(4190)-z(251)*z(2432)*z(4190)-z(283)*z(2388)*z(4162)-z(284)*z(2388)*z(4176)-
z(285)*z(2410)*z(4162)-z(286)*z(2410)*z(4176)-z(287)*z(2432)*z(4162)-z(288)*z(2432)*z(4176));
z(4412) = z(3067) + 0.1428571428571429*z(4388);

```
z(3061) = z(19)*z(166) + z(173)*z(195) - z(172)*z(193);
z(3062) = z(20)*z(166) + z(172)*z(194) - z(173)*z(196);
z(3063) = z(23)*z(173) + z(24)*z(172);
z(3065) = z(199)*z(1765) + z(203)*z(1781) + z(204)*z(1797);
z(3066) = z(205)*z(1781) + z(206)*z(1797) - z(28)*z(1765);
z(5092) = z(3078)*z(5070) - z(4414) - z(124)*z(5064) - z(126)*z(5063) - z(128)*z(5062) - z(3075)*z(5067) - z(3076)*z(5066) - z(3077)*z(5065) -
z(3079)*z(5069) - z(3080)*z(5068) - z(5089)*(z(16)*z(5062)+z(3064)*z(5070)-z(4412)-z(119)*z(5064)-z(121)*z(5063)-z(3061)*z(5067)-
z(3062)*z(5066)-z(3063)*z(5065)-z(3065)*z(5069)-z(3066)*z(5068));
z(5090) = z(126) - z(121)*z(5089);
z(5085) = z(123)/z(119);
z(5086) = z(125) - z(121)*z(5085);
z(5093) = z(5090)/z(5086);
z(3071) = z(197)*z(1766) + z(201)*z(1782) + z(202)*z(1798);
z(3074) = 4.9681E-05*(1+436.869225659709*z(130)-23.60302731426501*z(133)-4.101447233348765*z(136))*z(454) +
0.000439106*(1.522912463049924+15.07109900570705*z(130)-6.399042600192209*z(133)-z(136))*z(470) + 0.000830224*(-
10.13672213763996+1.648193740484496*z(133)-18.29142496482877*z(130)-z(136))*z(486) + 4.7299E-
05*(z(1798)+44.21104040254551*z(1782)+114.2537897207129*z(207)*z(1766)+114.2537897207129*z(208)*z(1782)+114.2537897207129*z(209)*z(
1798)-147.8697224042791*z(1766)-21.77701431319901*z(210)*z(1766)-21.77701431319901*z(211)*z(1782)-21.77701431319901*z(212)*z(1798)-
3.298991522019493*z(213)*z(1766)-3.298991522019493*z(214)*z(1782)-3.298991522019493*z(215)*z(1798))*z(462) +
0.000156039*(z(175)+6.601112542377225*z(174)+5.76932689904447*z(168)*z(170)+5.76932689904447*z(174)*z(176)+5.76932689904447*z(175)*
z(177)+34.22740468728971*z(166)*z(168)+34.22740468728971*z(172)*z(174)+34.22740468728971*z(173)*z(175)-34.63294432802056*z(168)-
4.285595267849704*z(168)^2-4.285595267849704*z(174)^2-4.285595267849704*z(175)^2)*z(466) +
0.000263142*(z(1798)+57.96163288262611*z(1766)-37.04060925279887*z(1782)-58.17953804409786*z(207)*z(1766)-
58.17953804409786*z(208)*z(1782)-58.17953804409786*z(209)*z(1798)-3.892632114979745*z(213)*z(1766)-3.892632114979745*z(214)*z(1782)-
3.892632114979745*z(215)*z(1798)-3.889998555912777*z(210)*z(1766)-3.889998555912777*z(211)*z(1782)-
3.889998555912777*z(212)*z(1798))*z(478) + 4.9681E-
05*(19.09315432459089*z(174)+25.99182786175802*z(168)*z(170)+25.99182786175802*z(174)*z(176)+25.99182786175802*z(175)*z(177)+410.78
68199110324*z(166)*z(168)+410.7868199110324*z(172)*z(174)+410.7868199110324*z(173)*z(175)-396.0548298142147*z(168)-
8.034661138060828*z(175)-z(168)^2-z(174)^2-z(175)^2)*z(450) + 6.128E-05*(19.44327676240209*z(175)+238.8415469973891*z(168)-
120.5957898172324*z(174)-232.8573759791123*z(166)*z(168)-232.8573759791123*z(172)*z(174)-232.8573759791123*z(173)*z(175)-
1.221507832898173*z(168)^2-1.221507832898173*z(174)^2-1.221507832898173*z(175)^2-z(168)*z(170)-z(174)*z(176)-z(175)*z(177))*z(482) +
0.00297212*(2.986921120277781*z(168)+2.589087923771584*z(168)*z(170)+2.589087923771584*z(174)*z(176)+2.589087923771584*z(175)*z(177
)-2.289867165524945*z(175)-z(174)-2.471453709809833*z(166)*z(168)-2.471453709809833*z(172)*z(174)-2.471453709809833*z(173)*z(175)-
1.384676594484745*z(168)^2-1.384676594484745*z(174)^2-1.384676594484745*z(175)^2)*z(498) +
0.00334198*(2.523796072986673*z(175)+2.541448183412228*z(168)-z(174)-2.199101730112089*z(166)*z(168)-
2.199101730112089*z(172)*z(174)-2.199101730112089*z(173)*z(175)-2.178312856450368*z(168)*z(170)-2.178312856450368*z(174)*z(176)-
2.178312856450368*z(175)*z(177)-1.513417495017923*z(168)^2-1.513417495017923*z(174)^2-1.513417495017923*z(175)^2)*z(514) +
0.005250594000000001*(1.3792740402324*z(1798)+1.576289654084852*z(1766)-1.220873295478569*z(1782)-1.749405305380686*z(207)*z(1766)-
1.749405305380686*z(208)*z(1782)-1.749405305380686*z(209)*z(1798)-1.677448684853561*z(213)*z(1766)-1.677448684853561*z(214)*z(1782)-
1.677448684853561*z(215)*z(1798)-z(210)*z(1766)-z(211)*z(1782)-z(212)*z(1798))*z(510) - 1.4*SW*z(125) -
9.800000000000001*SW*(z(2790)*z(2812)+z(2791)*z(2837)+z(2792)*z(2862)) - 2.8*SW*(z(284)*z(2389)+z(286)*z(2411)+z(288)*z(2433)) -
1.4*SW*(z(199)*z(1766)+z(203)*z(1782)+z(204)*z(1798)) - 1.4*SW*(z(20)*z(168)+z(174)*z(194)-z(175)*z(196)) -
0.00404624*(1+1.135663974455297*z(133)+1.613396387757523*z(130)-1.514074795365574*z(136))*z(502) -
0.00353959*(1+1.758446882266025*z(130)+2.253045691732658*z(136)+2.270871202596911*z(133))*z(518) -
0.00508017*(z(1782)+1.575266182037215*z(1798)+1.26191682561804*z(210)*z(1766)+1.26191682561804*z(211)*z(1782)+1.26191682561804*z(2
12)*z(1798)+1.738918973183969*z(207)*z(1766)+1.738918973183969*z(208)*z(1782)+1.738918973183969*z(209)*z(1798)-
1.564399813392072*z(1766)-1.445018572213135*z(213)*z(1766)-1.445018572213135*z(214)*z(1782)-1.445018572213135*z(215)*z(1798))*z(494)
- (2.0E-
07*z(1798)+0.0210342*z(1766)+0.000948567000000000001*z(210)*z(1766)+0.000948567000000000001*z(211)*z(1782)+0.000948567000000000001*z(212
)*z(1798)-0.001343886*z(1782)-0.0196764*z(207)*z(1766)-0.0196764*z(208)*z(1782)-0.0196764*z(209)*z(1798)-0.00039917*z(213)*z(1766)-
0.00039917*z(214)*z(1782)-0.00039917*z(215)*z(1798))*z(446);
z(4389) = SW*(z(34)*z(2389)*z(4190)-z(3653)-7*z(2812)*z(4370)-7*z(2837)*z(4372)-7*z(2862)*z(4371)-z(168)*z(3741)-z(174)*z(3743)-
z(175)*z(3742)-z(1766)*z(3867)-z(1782)*z(3869)-z(1798)*z(3868)-z(2389)*z(4020)-z(2389)*z(4191)-z(2411)*z(4022)-z(2411)*z(4193)-
z(2433)*z(4021)-z(2433)*z(4192)-z(250)*z(2411)*z(4190)-z(251)*z(2433)*z(4190)-z(283)*z(2389)*z(4162)-z(284)*z(2389)*z(4176)-
z(285)*z(2411)*z(4162)-z(286)*z(2411)*z(4176)-z(287)*z(2433)*z(4162)-z(288)*z(2433)*z(4176));
z(4413) = z(3074) + 0.1428571428571429*z(4389);
z(3068) = z(19)*z(168) + z(175)*z(195) - z(174)*z(193);
z(3069) = z(20)*z(168) + z(174)*z(194) - z(175)*z(196);
z(3070) = z(23)*z(175) + z(24)*z(174);
z(3072) = z(199)*z(1766) + z(203)*z(1782) + z(204)*z(1798);
z(3073) = z(205)*z(1782) + z(206)*z(1798) - z(28)*z(1766);
z(5088) = z(3071)*z(5070) - z(4413) - z(123)*z(5064) - z(125)*z(5063) - z(127)*z(5062) - z(3068)*z(5067) - z(3069)*z(5066) - z(3070)*z(5065) -
z(3072)*z(5069) - z(3073)*z(5068) - z(5085)*(z(16)*z(5062)+z(3064)*z(5070)-z(4412)-z(119)*z(5064)-z(121)*z(5063)-z(3061)*z(5067)-
z(3062)*z(5066)-z(3063)*z(5065)-z(3065)*z(5069)-z(3066)*z(5068));
z(5095) = z(5092) - z(5093)*z(5088);
z(5091) = z(128) + z(16)*z(5089);
z(5087) = z(127) + z(16)*z(5085);
z(5094) = z(5091) - z(5087)*z(5093);
z(5096) = z(5095)/z(5094);
z(5097) = (z(5088)-z(5087)*z(5096))/z(5086);
z(5098) = (z(16)*z(5062)+z(16)*z(5096)+z(3064)*z(5070)-z(4412)-z(121)*z(5063)-z(121)*z(5097)-z(3061)*z(5067)-z(3062)*z(5066)-
z(3063)*z(5065)-z(3065)*z(5069)-z(3066)*z(5068))/z(119) - z(5064);
FBC1 = z(5098);
FBC2 = z(5097);
FBC3 = z(5096);
z(3082) = z(197)*z(207) + z(201)*z(208) + z(202)*z(209);
z(3085) = 0.000263142*(z(209)+57.96163288262611*z(207)-37.04060925279887*z(208)-58.17953804409786*z(207)^2-
58.17953804409786*z(208)^2-58.17953804409786*z(209)^2-3.892632114979745*z(207)*z(213)-3.892632114979745*z(208)*z(214)-
```

3.892632114979745*z(209)*z(215)-3.889998555912777*z(207)*z(210)-3.889998555912777*z(208)*z(211)-
3.889998555912777*z(209)*z(212))*z(478) +
0.00508017*(1.564399813392072*z(207)+1.445018572213135*z(207)*z(213)+1.445018572213135*z(208)*z(214)+1.445018572213135*z(209)*z(215
)-1.575266182037215*z(209)-z(208)-1.738918973183969*z(207)^2-1.738918973183969*z(208)^2-1.738918973183969*z(209)^2-
1.26191682561804*z(207)*z(210)-1.26191682561804*z(208)*z(211)-1.26191682561804*z(209)*z(212))*z(494) +
0.005250594000000001*(1.3792740402324*z(209)+1.576289654084852*z(207)-1.220873295478569*z(208)-1.749405305380686*z(207)^2-
1.749405305380686*z(208)^2-1.749405305380686*z(209)^2-1.677448684853561*z(207)*z(213)-1.677448684853561*z(208)*z(214)-
1.677448684853561*z(209)*z(215)-z(207)*z(210)-z(208)*z(211)-z(209)*z(212))*z(510) - 1.4*SW*z(20) -
9.800000000000001*SW*(z(2790)*z(2808)+z(2791)*z(2833)+z(2792)*z(2858)) - 2.8*SW*(z(284)*z(2385)+z(286)*z(2407)+z(288)*z(2429)) -
1.4*SW*(z(199)*z(207)+z(203)*z(208)+z(204)*z(209)) - 0.005057811*(-
1.679277655887102+z(168)+1.439333735483592*z(170)+1.453070112742449*z(166))*z(514) - 0.004115425*(-
2.15712544876896 1+z(168)+1.784859886888961*z(166)-1.869814174720716*z(170))*z(498) - 0.00066872*(8.081244766120349+z(168)-
7.986616222036129*z(166)-1.346213661921283*z(170))*z(466) - 6.128E-05*(-
238.8415469973891+z(170)+1.221507832898173*z(168)+232.8573759791123*z(166))*z(482) - 4.9681E-05*(396.0548298142147+z(168)-
410.7868199110324*z(166)-25.99182786175802*z(170))*z(450) - (2.0E-
07*z(209)+0.0210342*z(207)+0.0009485670000000001*z(207)*z(210)+0.0009485670000000001*z(208)*z(211)+0.0009485670000000001*z(209)*z(
212)-0.001343886*z(208)-0.0196764*z(207)^2-0.0196764*z(208)^2-0.0196764*z(209)^2-0.00039917*z(207)*z(213)-0.00039917*z(208)*z(214)-
0.00039917*z(209)*z(215))*z(446) - 4.7299E-
05*(147.8697224042791*z(207)+3.298991522019493*z(207)*z(213)+3.298991522019493*z(208)*z(214)+3.298991522019493*z(209)*z(215)+21.777
01431319901*z(207)*z(210)+21.77701431319901*z(208)*z(211)+21.77701431319901*z(209)*z(212)-44.21104040254551*z(208)-z(209)-
114.2537897207129*z(207)^2-114.2537897207129*z(208)^2-114.2537897207129*z(209)^2)*z(462);
z(4391) = SW*(z(34)*z(2385)*z(4190)-z(3741)-7*z(2808)*z(4370)-7*z(2833)*z(4372)-7*z(2858)*z(4371)-z(207)*z(3867)-z(208)*z(3869)-
z(209)*z(3868)-z(2385)*z(4020)-z(2385)*z(4191)-z(2407)*z(4022)-z(2407)*z(4193)-z(2429)*z(4021)-z(2429)*z(4192)-z(250)*z(2407)*z(4190)-
z(251)*z(2429)*z(4190)-z(283)*z(2385)*z(4162)-z(284)*z(2385)*z(4176)-z(285)*z(2407)*z(4162)-z(286)*z(2407)*z(4176)-z(287)*z(2429)*z(4162)-
z(288)*z(2429)*z(4176));
z(4415) = z(3085) + 0.1428571428571429*z(4391);
z(3090) = z(197)*z(213) + z(201)*z(214) + z(202)*z(215);
z(3093) = 1.4*SW*z(196) + 4.7299E-
05*(z(215)+44.21104040254551*z(214)+114.2537897207129*z(207)*z(213)+114.2537897207129*z(208)*z(214)+114.2537897207129*z(209)*z(215)
-147.8697224042791*z(213)-21.77701431319901*z(210)*z(213)-21.77701431319901*z(211)*z(214)-21.77701431319901*z(212)*z(215)-
3.298991522019493*z(213)^2-3.298991522019493*z(214)^2-3.298991522019493*z(215)^2)*z(462) +
0.000263142*(z(215)+57.96163288262611*z(213)-37.04060925279887*z(214)-58.17953804409786*z(207)*z(213)-
58.17953804409786*z(208)*z(214)-58.17953804409786*z(209)*z(215)-3.892632114979745*z(213)^2-3.892632114979745*z(214)^2-
3.892632114979745*z(215)^2-3.889998555912777*z(210)*z(213)-3.889998555912777*z(211)*z(214)-3.889998555912777*z(212)*z(215))*z(478) +
0.005250594000000001*(1.3792740402324*z(215)+1.576289654084852*z(213)-1.220873295478569*z(214)-1.749405305380686*z(207)*z(213)-
1.749405305380686*z(208)*z(214)-1.749405305380686*z(209)*z(215)-1.677448684853561*z(213)^2-1.677448684853561*z(214)^2-
1.677448684853561*z(215)^2-z(210)*z(213)-z(211)*z(214)-z(212)*z(215))*z(510) -
9.800000000000001*SW*(z(2790)*z(2810)+z(2791)*z(2835)+z(2792)*z(2860)) - 2.8*SW*(z(284)*z(2387)+z(286)*z(2409)+z(288)*z(2431)) -
1.4*SW*(z(199)*z(213)+z(203)*z(214)+z(204)*z(215)) - 0.005057811*(-
1.667613914398937+z(175)+1.439333735483592*z(177)+1.453070112742449*z(173))*z(514) -
0.004115425*(1.653719846674402+z(175)+1.784859886888961*z(173)-1.869814174720716*z(177))*z(498) - 6.128E-05*(-
19.44327676240209+z(177)+1.221507832898173*z(175)+232.8573759791123*z(173))*z(482) - 4.9681E-05*(8.034661138060828+z(175)-
410.7868199110324*z(173)-25.99182786175802*z(177))*z(450) - 0.000156039*(-1+4.285595267849704*z(175)-34.22740468728971*z(173)-
5.76932689904447*z(177))*z(466) -
0.00508017*(z(214)+1.575266182037215*z(215)+1.26191682561804*z(210)*z(213)+1.26191682561804*z(211)*z(214)+1.26191682561804*z(212)*z
(215)+1.738918973183969*z(207)*z(213)+1.738918973183969*z(208)*z(214)+1.738918973183969*z(209)*z(215)-1.564399813392072*z(213)-
1.445018572213135*z(213)^2-1.445018572213135*z(214)^2-1.445018572213135*z(215)^2)*z(494) - (2.0E-
07*z(215)+0.0210342*z(213)+0.0009485670000000001*z(210)*z(213)+0.0009485670000000001*z(211)*z(214)+0.0009485670000000001*z(212)*z(
215)-0.001343886*z(214)-0.0196764*z(207)*z(213)-0.0196764*z(208)*z(214)-0.0196764*z(209)*z(215)-0.00039917*z(213)^2-0.00039917*z(214)^2-
0.00039917*z(215)^2)*z(446);
z(4393) = SW*(z(34)*z(2387)*z(4190)-z(3742)-7*z(2810)*z(4370)-7*z(2835)*z(4372)-7*z(2860)*z(4371)-z(213)*z(3867)-z(214)*z(3869)-
z(215)*z(3868)-z(2387)*z(4020)-z(2387)*z(4191)-z(2409)*z(4022)-z(2409)*z(4193)-z(2431)*z(4021)-z(2431)*z(4192)-z(250)*z(2409)*z(4190)-
z(251)*z(2431)*z(4190)-z(283)*z(2387)*z(4162)-z(284)*z(2387)*z(4176)-z(285)*z(2409)*z(4162)-z(286)*z(2409)*z(4176)-z(287)*z(2431)*z(4162)-
z(288)*z(2431)*z(4176));
z(4417) = z(3093) + 0.1428571428571429*z(4393);
z(3091) = z(199)*z(213) + z(203)*z(214) + z(204)*z(215);
z(3092) = z(205)*z(214) + z(206)*z(215) - z(28)*z(213);
z(5102) = z(195)/z(19);
z(3083) = z(199)*z(207) + z(203)*z(208) + z(204)*z(209);
z(3084) = z(205)*z(208) + z(206)*z(209) - z(28)*z(207);
z(5104) = z(196)*z(5066) + z(3090)*z(5070) - z(4417) - z(23)*z(5065) - z(195)*z(5067) - z(3091)*z(5069) - z(3092)*z(5068) -
z(5102)*(z(3082)*z(5070)-z(4415)-z(19)*z(5067)-z(20)*z(5066)-z(3083)*z(5069)-z(3084)*z(5068));
z(3086) = z(197)*z(210) + z(201)*z(211) + z(202)*z(212);
z(5099) = z(193)/z(19);
z(3089) =
(0.001343886*z(211)+0.00039917*z(210)*z(213)+0.00039917*z(211)*z(214)+0.00039917*z(212)*z(215)+0.0196764*z(207)*z(210)+0.0196764*z(20
8)*z(211)+0.0196764*z(209)*z(212)-0.0210342*z(210)-2.0E-07*z(212)-0.0009485670000000001*z(210)^2-0.0009485670000000001*z(211)^2-
0.0009485670000000001*z(212)^2)*z(446) + 4.7299E-
05*(z(212)+44.21104040254551*z(211)+114.2537897207129*z(207)*z(210)+114.2537897207129*z(208)*z(211)+114.2537897207129*z(209)*z(212)
-147.8697224042791*z(210)-21.77701431319901*z(210)^2-21.77701431319901*z(211)^2-21.77701431319901*z(212)^2-
3.298991522019493*z(210)*z(213)-3.298991522019493*z(211)*z(214)-3.298991522019493*z(212)*z(215))*z(462) +
0.000263142*(z(212)+57.96163288262611*z(210)-37.04060925279887*z(211)-58.17953804409786*z(207)*z(210)-
58.17953804409786*z(208)*z(211)-58.17953804409786*z(209)*z(212)-3.892632114979745*z(210)*z(213)-3.892632114979745*z(211)*z(214)-
3.892632114979745*z(212)*z(215)-3.889998555912777*z(210)^2-3.889998555912777*z(211)^2-3.889998555912777*z(212)^2)*z(478) +
0.00508017*(1.564399813392072*z(210)+1.445018572213135*z(210)*z(213)+1.445018572213135*z(211)*z(214)+1.445018572213135*z(212)*z(215
)-1.575266182037215*z(212)-z(211)-1.738918973183969*z(207)*z(210)-1.738918973183969*z(208)*z(211)-1.738918973183969*z(209)*z(212)-
1.26191682561804*z(210)^2-1.26191682561804*z(211)^2-1.26191682561804*z(212)^2)*z(494) +
0.005250594000000001*(1.3792740402324*z(212)+1.576289654084852*z(210)-1.220873295478569*z(211)-1.749405305380686*z(207)*z(210)-

1.749405305380686*z(208)*z(211)-1.749405305380686*z(209)*z(212)-1.677448684853561*z(210)*z(213)-1.677448684853561*z(211)*z(214)-
1.677448684853561*z(212)*z(215)-z(210)^2-z(211)^2-z(212)^2)*z(510) - 1.4*SW*z(194) -
9.800000000000001*SW*(z(2790)*z(2809)+z(2791)*z(2834)+z(2792)*z(2859)) - 2.8*SW*(z(284)*z(2386)+z(286)*z(2408)+z(288)*z(2430)) -
1.4*SW*(z(199)*z(210)+z(203)*z(211)+z(204)*z(212)) - 0.00066872*(-1.54030236870439+z(174)-7.986616222036129*z(172)-
1.346213661921283*z(176))*z(466) - 6.128E-05*(120.5957898172324+z(176)+1.221507832898173*z(174)+232.8573759791123*z(172))*z(482) -
4.9681E-05*(-19.09315432459089+z(174)-410.7868199110324*z(172)-25.99182786175802*z(176))*z(450) -
0.00334198*(1+1.513417495017923*z(174)+2.178312856450368*z(176)+2.199101730112089*z(172))*z(514) -
0.00297212*(1+1.384676594484745*z(174)+2.471453709809833*z(172)-2.589087923771584*z(176))*z(498);
z(4392) = SW*(z(34)*z(2386)*z(4190)-z(3743)-7*z(2809)*z(4370)-7*z(2834)*z(4372)-7*z(2859)*z(4371)-z(210)*z(3867)-z(211)*z(3869)-
z(212)*z(3868)-z(2386)*z(4020)-z(2386)*z(4191)-z(2408)*z(4022)-z(2408)*z(4193)-z(2430)*z(4021)-z(2430)*z(4192)-z(250)*z(2408)*z(4190)-
z(251)*z(2430)*z(4190)-z(283)*z(2386)*z(4162)-z(284)*z(2386)*z(4176)-z(285)*z(2408)*z(4162)-z(286)*z(2408)*z(4176)-z(287)*z(2430)*z(4162)-
z(288)*z(2430)*z(4176));
z(4416) = z(3089) + 0.1428571428571429*z(4392);
z(3087) = z(199)*z(210) + z(203)*z(211) + z(204)*z(212);
z(3088) = z(205)*z(211) + z(206)*z(212) - z(28)*z(210);
z(5101) = z(193)*z(5067) + z(3086)*z(5070) + z(5099)*(z(3082)*z(5070)-z(4415)-z(19)*z(5067)-z(20)*z(5066)-z(3083)*z(5069)-z(3084)*z(5068)) -
z(4416) - z(24)*z(5065) - z(194)*z(5066) - z(3087)*z(5069) - z(3088)*z(5068);
z(5100) = z(194) + z(20)*z(5099);
z(5103) = -z(196) - z(20)*z(5102);
z(5113) = z(5100)/z(5103);
z(5115) = z(5101) - z(5113)*z(5104);
z(5114) = z(24) - z(23)*z(5113);
z(5120) = z(5115)/z(5114);
z(5122) = (z(5104)-z(23)*z(5120))/z(5103);
z(5124) = (z(3082)*z(5070)-z(4415)-z(20)*z(5066)-z(20)*z(5122)-z(3083)*z(5069)-z(3084)*z(5068))/z(19) - z(5067);
FCD1 = z(5124);
FCD2 = z(5122);
FCD3 = z(5120);
z(3096) = 4.7299E-05*(1+114.2537897207129*z(209)-21.77701431319901*z(212)-3.298991522019493*z(215))*z(462) + 0.000263142*(1-
58.17953804409786*z(209)-3.892632114979745*z(215)-3.889998555912777*z(212))*z(478) - 1.4*SW*z(204) -
9.800000000000001*SW*(z(2790)*z(2807)+z(2791)*z(2832)+z(2792)*z(2857)) - 2.8*SW*(z(258)*z(284)+z(259)*z(286)+z(260)*z(288)) -
0.006410752*(1.248312210486383+z(212)+1.377998088211804*z(209)-1.145098110174906*z(215))*z(494) - 0.005250594000000001*(-
1.3792740402324+z(212)+1.677448684853561*z(215)+1.749405305380686*z(209))*z(510) - (2.0E-07+0.0009485670000000001*z(212)-
0.0196764*z(209)-0.00039917*z(215))*z(446);
z(4396) = SW*(z(34)*z(258)*z(4190)-z(3868)-7*z(2807)*z(4370)-7*z(2832)*z(4372)-7*z(2857)*z(4371)-z(258)*z(4020)-z(258)*z(4191)-
z(259)*z(4022)-z(259)*z(4193)-z(260)*z(4021)-z(260)*z(4192)-z(250)*z(259)*z(4190)-z(251)*z(260)*z(4190)-z(258)*z(283)*z(4162)-
z(258)*z(284)*z(4176)-z(259)*z(285)*z(4162)-z(259)*z(286)*z(4176)-z(260)*z(287)*z(4162)-z(260)*z(288)*z(4176));
z(4420) = z(3096) + 0.1428571428571429*z(4396);
z(5109) = z(202)/z(197);
z(3094) = 0.000156039*(-44.82270458026518+34.63294432802056*z(207)-6.601112542377225*z(210)-z(213))*z(462) - 1.4*SW*z(199) -
9.800000000000001*SW*(z(2790)*z(2805)+z(2791)*z(2830)+z(2792)*z(2855)) - 2.8*SW*(z(252)*z(284)+z(253)*z(286)+z(254)*z(288)) -
0.006410752*(-1.23970120822019+z(210)+1.377998088211804*z(207)-1.145098110174906*z(213))*z(494) - 0.005250594000000001*(-
1.576289654084852+z(210)+1.677448684853561*z(213)+1.749405305380686*z(207))*z(510) - 0.001023622*(-
14.90016822616161+z(210)+1.000677007723554*z(213)+14.95618499797777*z(207))*z(478) -
0.00039917*(52.69484179672821+2.376348422977679*z(210)-49.29328356339404*z(207)-z(213))*z(446);
z(4394) = SW*(z(34)*z(252)*z(4190)-z(3867)-7*z(2805)*z(4370)-7*z(2830)*z(4372)-7*z(2855)*z(4371)-z(252)*z(4020)-z(252)*z(4191)-
z(253)*z(4022)-z(253)*z(4193)-z(254)*z(4021)-z(254)*z(4192)-z(250)*z(253)*z(4190)-z(251)*z(254)*z(4190)-z(252)*z(283)*z(4162)-
z(252)*z(284)*z(4176)-z(253)*z(285)*z(4162)-z(253)*z(286)*z(4176)-z(254)*z(287)*z(4162)-z(254)*z(288)*z(4176));
z(4418) = z(3094) + 0.1428571428571429*z(4394);
z(5112) = z(202)*z(5070) - z(4420) - z(204)*z(5069) - z(206)*z(5068) - z(5109)*(z(28)*z(5068)+z(197)*z(5070)-z(4418)-z(199)*z(5069));
z(5110) = z(204) - z(199)*z(5109);
z(5105) = z(201)/z(197);
z(5106) = z(203) - z(199)*z(5105);
z(5116) = z(5110)/z(5106);
z(3095) = 0.000156039*(13.40138042412474+34.63294432802056*z(208)-6.601112542377225*z(211)-z(214))*z(462) - 1.4*SW*z(203) -
9.800000000000001*SW*(z(2790)*z(2806)+z(2791)*z(2831)+z(2792)*z(2856)) - 2.8*SW*(z(255)*z(284)+z(256)*z(286)+z(257)*z(288)) -
0.005250594000000001*(1.220873295478569+z(211)+1.677448684853561*z(214)+1.749405305380686*z(208))*z(510) -
0.001023622*(9.522011054862048+z(211)+1.000677007723554*z(214)+14.95618499797777*z(208))*z(478) -
0.00508017*(1+1.261916825561804*z(211)+1.738918973183969*z(208)-1.445018572213135*z(214))*z(494) - 0.00039917*(-
3.366700904376581+2.376348422977679*z(211)-49.29328356339404*z(208)-z(214))*z(446);
z(4395) = SW*(z(34)*z(255)*z(4190)-z(3869)-7*z(2806)*z(4370)-7*z(2831)*z(4372)-7*z(2856)*z(4371)-z(255)*z(4020)-z(255)*z(4191)-
z(256)*z(4022)-z(256)*z(4193)-z(257)*z(4021)-z(257)*z(4192)-z(250)*z(256)*z(4190)-z(251)*z(257)*z(4190)-z(255)*z(283)*z(4162)-
z(255)*z(284)*z(4176)-z(256)*z(285)*z(4162)-z(256)*z(286)*z(4176)-z(257)*z(287)*z(4162)-z(257)*z(288)*z(4176));
z(4419) = z(3095) + 0.1428571428571429*z(4395);
z(5108) = z(201)*z(5070) - z(4419) - z(203)*z(5069) - z(205)*z(5068) - z(5105)*(z(28)*z(5068)+z(197)*z(5070)-z(4418)-z(199)*z(5069));
z(5118) = z(5112) - z(5116)*z(5108);
z(5111) = z(206) + z(28)*z(5109);
z(5107) = z(205) + z(28)*z(5105);
z(5117) = z(5111) - z(5107)*z(5116);
z(5119) = z(5118)/z(5117);
z(5121) = (z(5108)-z(5107)*z(5119))/z(5106);
z(5123) = z(5070) + (z(28)*z(5068)+z(28)*z(5119)-z(4418)-z(199)*z(5069)-z(199)*z(5121))/z(197);
FDE1 = z(5123);
FDE2 = z(5121);
FDE3 = z(5119);
FAB12 = 0.0510204081632653*FAB1/SW;
FBC12 = 0.0510204081632653*FBC1/SW;
FCD12 = 0.0510204081632653*FCD1/SW;

```
FDE12 = 0.0510204081632653*FDE1/SW;
FAB22 = 0.0510204081632653*FAB2/SW;
FBC22 = 0.0510204081632653*FBC2/SW;
FCD22 = 0.0510204081632653*FCD2/SW;
FDE22 = 0.0510204081632653*FDE2/SW;
FAB32 = 0.0510204081632653*FAB3/SW;
FBC32 = 0.0510204081632653*FBC3/SW;
FCD32 = 0.0510204081632653*FCD3/SW;
FDE32 = 0.0510204081632653*FDE3/SW;
z(5128) = z(49)*z(5125) + z(51)*z(5126) - z(4)*z(5127);
z(5129) = z(53)*z(5125) + z(55)*z(5126) + z(57)*z(5127);
z(5130) = z(54)*z(5125) + z(56)*z(5126) + z(58)*z(5127);
z(5134) = z(49)*z(5131) + z(51)*z(5132) - z(4)*z(5133);
z(5135) = z(53)*z(5131) + z(55)*z(5132) + z(57)*z(5133);
z(5136) = z(54)*z(5131) + z(56)*z(5132) + z(58)*z(5133);
z(5137) = z(4)*z(16) + z(49)*z(119) + z(51)*z(121);
z(5138) = z(49)*z(123) + z(51)*z(125) - z(4)*z(127);
z(5139) = z(49)*z(124) + z(51)*z(126) - z(4)*z(128);
z(5140) = z(53)*z(119) + z(55)*z(121) - z(16)*z(57);
z(5141) = z(53)*z(123) + z(55)*z(125) + z(57)*z(127);
z(5142) = z(53)*z(124) + z(55)*z(126) + z(57)*z(128);
z(5143) = z(54)*z(119) + z(56)*z(121) - z(16)*z(58);
z(5144) = z(54)*z(123) + z(56)*z(125) + z(58)*z(127);
z(5145) = z(54)*z(124) + z(56)*z(126) + z(58)*z(128);
z(5149) = z(49)*z(5146) + z(51)*z(5147) - z(4)*z(5148);
z(5150) = z(53)*z(5146) + z(55)*z(5147) + z(57)*z(5148);
z(5151) = z(54)*z(5146) + z(56)*z(5147) + z(58)*z(5148);
z(5152) = z(19)*z(49) + z(20)*z(51);
z(5153) = z(51)*z(194) - z(4)*z(24) - z(49)*z(193);
z(5154) = z(49)*z(195) - z(4)*z(23) - z(51)*z(196);
z(5155) = z(19)*z(53) + z(20)*z(55);
z(5156) = z(24)*z(57) + z(55)*z(194) - z(53)*z(193);
z(5157) = z(23)*z(57) + z(53)*z(195) - z(55)*z(196);
z(5158) = z(19)*z(54) + z(20)*z(56);
z(5159) = z(24)*z(58) + z(56)*z(194) - z(54)*z(193);
z(5160) = z(23)*z(58) + z(54)*z(195) - z(56)*z(196);
z(5164) = z(49)*z(5161) + z(51)*z(5162) - z(4)*z(5163);
z(5165) = z(53)*z(5161) + z(55)*z(5162) + z(57)*z(5163);
z(5166) = z(54)*z(5161) + z(56)*z(5162) + z(58)*z(5163);
z(5167) = z(4)*z(28) + z(49)*z(197) + z(51)*z(199);
z(5168) = z(49)*z(201) + z(51)*z(203) - z(4)*z(205);
z(5169) = z(49)*z(202) + z(51)*z(204) - z(4)*z(206);
z(5170) = z(53)*z(197) + z(55)*z(199) - z(28)*z(57);
z(5171) = z(53)*z(201) + z(55)*z(203) + z(57)*z(205);
z(5172) = z(53)*z(202) + z(55)*z(204) + z(57)*z(206);
z(5173) = z(54)*z(197) + z(56)*z(199) - z(28)*z(58);
z(5174) = z(54)*z(201) + z(56)*z(203) + z(58)*z(205);
z(5175) = z(54)*z(202) + z(56)*z(204) + z(58)*z(206);
z(5179) = z(49)*z(5176) + z(51)*z(5177) - z(4)*z(5178);
z(5180) = z(53)*z(5176) + z(55)*z(5177) + z(57)*z(5178);
z(5181) = z(54)*z(5176) + z(56)*z(5177) + z(58)*z(5178);
z(5182) = z(4)*z(34) + z(49)*z(283) + z(51)*z(284);
z(5183) = z(49)*z(285) + z(51)*z(286) - z(4)*z(250);
z(5184) = z(49)*z(287) + z(51)*z(288) - z(4)*z(251);
z(5185) = z(53)*z(283) + z(55)*z(284) - z(34)*z(57);
z(5186) = z(53)*z(285) + z(55)*z(286) + z(57)*z(250);
z(5187) = z(53)*z(287) + z(55)*z(288) + z(57)*z(251);
z(5188) = z(54)*z(283) + z(56)*z(284) - z(34)*z(58);
z(5189) = z(54)*z(285) + z(56)*z(286) + z(58)*z(250);
z(5190) = z(54)*z(287) + z(56)*z(288) + z(58)*z(251);
z(5194) = z(49)*z(5191) + z(51)*z(5192) - z(4)*z(5193);
z(5195) = z(53)*z(5191) + z(55)*z(5192) + z(57)*z(5193);
z(5196) = z(54)*z(5191) + z(56)*z(5192) + z(58)*z(5193);
z(5197) = z(4)*z(40) + z(49)*z(289) + z(51)*z(291);
z(5198) = z(49)*z(293) + z(51)*z(295) - z(4)*z(297);
z(5199) = z(49)*z(294) + z(51)*z(296) - z(4)*z(298);
z(5200) = z(53)*z(289) + z(55)*z(291) - z(40)*z(57);
z(5201) = z(53)*z(293) + z(55)*z(295) + z(57)*z(297);
z(5202) = z(53)*z(294) + z(55)*z(296) + z(57)*z(298);
z(5203) = z(54)*z(289) + z(56)*z(291) - z(40)*z(58);
z(5204) = z(54)*z(293) + z(56)*z(295) + z(58)*z(297);
z(5205) = z(54)*z(294) + z(56)*z(296) + z(58)*z(298);
z(5206) = z(346)*z(5197) + z(349)*z(5198) + z(352)*z(5199);
z(5207) = z(346)*z(5200) + z(349)*z(5201) + z(352)*z(5202);
z(5208) = z(346)*z(5203) + z(349)*z(5204) + z(352)*z(5205);
z(5209) = z(347)*z(5197) + z(350)*z(5198) + z(353)*z(5199);
z(5210) = z(347)*z(5200) + z(350)*z(5201) + z(353)*z(5202);
z(5211) = z(347)*z(5203) + z(350)*z(5204) + z(353)*z(5205);
```

```
z(5212) = z(348)*z(5197) + z(351)*z(5198) + z(354)*z(5199);
z(5213) = z(348)*z(5200) + z(351)*z(5201) + z(354)*z(5202);
z(5214) = z(348)*z(5203) + z(351)*z(5204) + z(354)*z(5205);
z(5215) = 0.0005*z(5199) + 0.00893*z(5198) + 0.07100000000000001*z(5209) + z(49)*z(5191) + z(51)*z(5192) - 0.01653*z(5197) - z(4)*z(5193);
z(5216) = 0.0005*z(5202) + 0.00893*z(5201) + 0.07100000000000001*z(5210) + z(53)*z(5191) + z(55)*z(5192) + z(57)*z(5193) - 0.01653*z(5200);
z(5217) = 0.0005*z(5205) + 0.00893*z(5204) + 0.07100000000000001*z(5211) + z(54)*z(5191) + z(56)*z(5192) + z(58)*z(5193) - 0.01653*z(5203);



%==========================================================================
function Output = PrintUserOutput
global   T;
global   KB1 KB2 KB3 KC1 KC2 KC3 KD1 KD2 KD3 KE1 KE2 KE3 KH1 KH2 KH3 KK1 KK2 KK3 KS1 KS2 KS3 SW;
global   AOAB1 AOAB2 AOAB3 AON1 AON2 AON3 BOBA1 BOBA2 BOBA3 BOBC1 BOBC2 BOBC3 BON1 BON2 BON3 COCB1 COCB2
COCB3 COCD1 COCD2 COCD3 CON1 CON2 CON3 DABALL DABIL DABLLF DABPLL DABRLF DBCALL DBCIL DBCLLF DBCPLL
DBCRLF DCDALL DCDIL DCDLLF DCDPLL DCDRLF DDEALL DDEIL DDELLF DDEPLL DDERLF DODC1 DODC2 DODC3 DODE1 DODE2
DODE3 DON1 DON2 DON3 EOED1 EOED2 EOED3 EOEH1 EOEH2 EOEH3 EON1 EON2 EON3 FAB1 FAB2 FAB3 FBC1 FBC2 FBC3
FBSEMCER FBSEMCER1 FBSEMCER_1 FBSEMCER_2 FBSEMCER_3 FCD1 FCD2 FCD3 FCSEMCER FCSEMCER1 FCSEMCER_1
FCSEMCER_2 FCSEMCER_3 FDE1 FDE2 FDE3 FDSEMCER FDSEMCER1 FDSEMCER_1 FDSEMCER_2 FDSEMCER_3 FESEMCER
FESEMCER1 FESEMCER_1 FESEMCER_2 FESEMCER_3 HOHE1 HOHE2 HOHE3 HOHK1 HOHK2 HOHK3 HON1 HON2 HON3 KON1 KON2
KON3 U1 U10 U11 U12 U13 U14 U15 U16 U17 U18 U19 U2 U20 U21 U22 U23 U24 U3 U4 U5 U6 U7 U8 U9 QNA_1 QNA_2 QNA_3 QNB_1
QNB_2 QNB_3 QNC_1 QNC_2 QNC_3 QND_1 QND_3 QNE_1 QNE_2 QNE_3 QNH_1 QNH_2 QNH_3 QNK_1 QNK_2 QNK_3 QNS_1 QNS_2
QNS_3 QNA_1p QNA_2p QNA_3p QNB_1p QNB_2p QNB_3p QNC_1p QNC_2p QNC_3p QND_1p QND_3p QNE_1p QNE_2p QNE_3p QNH_1p
QNH_2p QNH_3p QNK_1p QNK_2p QNK_3p QNS_1p QNS_2p QNS_3p QNA_1pp QNA_2pp QNA_3pp QNB_1pp QNB_2pp QNB_3pp QNC_1pp
QNC_2pp QNC_3pp QND_1pp QND_3pp QNE_1pp QNE_2pp QNE_3pp QNH_1pp QNH_2pp QNH_3pp QNK_1pp QNK_2pp QNK_3pp QNS_1pp
QNS_2pp QNS_3pp FAB12 FAB22 FAB32 FABALL FABIL FABLLF FABPLL FABRLF FBC12 FBC22 FBC32 FBCALL FBCIL FBCLLF FBCPLL
FBCRLF FCD12 FCD22 FCD32 FCDALL FCDIL FCDLLF FCDPLL FCDRLF FDE12 FDE22 FDE32 FDEALL FDEIL FDELLF FDEPLL FDERLF;
global   DEGtoRAD RADtoDEG z;

Output(1)=T;  Output(2)=FABALL;  Output(3)=FBCALL;  Output(4)=FCDALL;  Output(5)=FDEALL;
Output(6)=T;  Output(7)=FABPLL;  Output(8)=FBCPLL;  Output(9)=FCDPLL;  Output(10)=FDEPLL;
Output(11)=T;  Output(12)=FABIL;  Output(13)=FBCIL;  Output(14)=FCDIL;  Output(15)=FDEIL;
Output(16)=T;  Output(17)=FABLLF;  Output(18)=FBCLLF;  Output(19)=FCDLLF;  Output(20)=FDELLF;
Output(21)=T;  Output(22)=FABRLF;  Output(23)=FBCRLF;  Output(24)=FCDRLF;  Output(25)=FDERLF;
Output(26)=T;  Output(27)=FAB12;  Output(28)=FBC12;  Output(29)=FCD12;  Output(30)=FDE12;
Output(31)=T;  Output(32)=FAB22;  Output(33)=FBC22;  Output(34)=FCD22;  Output(35)=FDE22;
Output(36)=T;  Output(37)=FAB32;  Output(38)=FBC32;  Output(39)=FCD32;  Output(40)=FDE32;
Output(41)=T;  Output(42)=FESEMCER;  Output(43)=FDSEMCER;  Output(44)=FCSEMCER;  Output(45)=FBSEMCER;
Output(46)=T;  Output(47)=FESEMCER1;  Output(48)=FDSEMCER1;  Output(49)=FCSEMCER1;  Output(50)=FBSEMCER1;
Output(51)=T;  Output(52)=z(5128);  Output(53)=z(5129);  Output(54)=z(5130);  Output(55)=z(82);  Output(56)=z(83);  Output(57)=z(84);
Output(58)=z(85);  Output(59)=z(86);  Output(60)=z(87);  Output(61)=z(88);  Output(62)=z(89);  Output(63)=z(90);
Output(64)=T;  Output(65)=z(5134);  Output(66)=z(5135);  Output(67)=z(5136);  Output(68)=z(5137);  Output(69)=z(5138);  Output(70)=z(5139);
Output(71)=z(5140);  Output(72)=z(5141);  Output(73)=z(5142);  Output(74)=z(5143);  Output(75)=z(5144);  Output(76)=z(5145);
Output(77)=T;  Output(78)=z(5149);  Output(79)=z(5150);  Output(80)=z(5151);  Output(81)=z(5152);  Output(82)=z(5153);  Output(83)=z(5154);
Output(84)=z(5155);  Output(85)=z(5156);  Output(86)=z(5157);  Output(87)=z(5158);  Output(88)=z(5159);  Output(89)=z(5160);
Output(90)=T;  Output(91)=z(5164);  Output(92)=z(5165);  Output(93)=z(5166);  Output(94)=z(5167);  Output(95)=z(5168);  Output(96)=z(5169);
Output(97)=z(5170);  Output(98)=z(5171);  Output(99)=z(5172);  Output(100)=z(5173);  Output(101)=z(5174);  Output(102)=z(5175);
Output(103)=T;  Output(104)=z(5179);  Output(105)=z(5180);  Output(106)=z(5181);  Output(107)=z(5182);  Output(108)=z(5183);
Output(109)=z(5184);  Output(110)=z(5185);  Output(111)=z(5186);  Output(112)=z(5187);  Output(113)=z(5188);  Output(114)=z(5189);
Output(115)=z(5190);
Output(116)=T;  Output(117)=z(5194);  Output(118)=z(5195);  Output(119)=z(5196);  Output(120)=z(5197);  Output(121)=z(5198);
Output(122)=z(5199);  Output(123)=z(5200);  Output(124)=z(5201);  Output(125)=z(5202);  Output(126)=z(5203);  Output(127)=z(5204);
Output(128)=z(5205);
Output(129)=T;  Output(130)=z(5215);  Output(131)=z(5216);  Output(132)=z(5217);  Output(133)=z(5206);  Output(134)=z(5209);
Output(135)=z(5212);  Output(136)=z(5207);  Output(137)=z(5210);  Output(138)=z(5213);  Output(139)=z(5208);  Output(140)=z(5211);
Output(141)=z(5214);
FileIdentifier = fopen('all');
WriteOutput( 1,              Output(1:5) );
WriteOutput( FileIdentifier(1), Output(1:5) );
WriteOutput( FileIdentifier(2), Output(6:10) );
WriteOutput( FileIdentifier(3), Output(11:15) );
WriteOutput( FileIdentifier(4), Output(16:20) );
WriteOutput( FileIdentifier(5), Output(21:25) );
WriteOutput( FileIdentifier(6), Output(26:30) );
WriteOutput( FileIdentifier(7), Output(31:35) );
WriteOutput( FileIdentifier(8), Output(36:40) );
WriteOutput( FileIdentifier(9), Output(41:45) );
WriteOutput( FileIdentifier(10), Output(46:50) );
WriteOutput( FileIdentifier(11), Output(51:63) );
WriteOutput( FileIdentifier(12), Output(64:76) );
WriteOutput( FileIdentifier(13), Output(77:89) );
WriteOutput( FileIdentifier(14), Output(90:102) );
WriteOutput( FileIdentifier(15), Output(103:115) );
WriteOutput( FileIdentifier(16), Output(116:128) );
WriteOutput( FileIdentifier(17), Output(129:141) );
```

```
%==============================================================================
function WriteOutput( fileIdentifier, Output )
numberOfOutputQuantities = length( Output );
if numberOfOutputQuantities > 0,
  for i=1:numberOfOutputQuantities,
    fprintf( fileIdentifier, ' %- 17.9E', Output(i) );
  end
  fprintf( fileIdentifier, '\n' );
end


%==============================================================================
function CloseOutputFilesAndTerminate
FileIdentifier = fopen('all');
fclose( FileIdentifier(1) );
fclose( FileIdentifier(2) );
fclose( FileIdentifier(3) );
fclose( FileIdentifier(4) );
fclose( FileIdentifier(5) );
fclose( FileIdentifier(6) );
fclose( FileIdentifier(7) );
fclose( FileIdentifier(8) );
fclose( FileIdentifier(9) );
fclose( FileIdentifier(10) );
fclose( FileIdentifier(11) );
fclose( FileIdentifier(12) );
fclose( FileIdentifier(13) );
fclose( FileIdentifier(14) );
fclose( FileIdentifier(15) );
fclose( FileIdentifier(16) );
fclose( FileIdentifier(17) );
fprintf( 1, '\n Output is in the files normen.i  (i=1, ..., 17)\n' );
fprintf( 1, ' The output quantities and associated files are listed in the file normen.dir\n' );
fprintf( 1, '\n To load and plot columns 1 and 2 with a solid line and columns 1 and 3 with a dashed line, enter:\n' );
fprintf( 1, '    someName = load( "normen.1" );\n' );
fprintf( 1, '    plot( someName(:,1), someName(:,2), "-", someName(:,1), someName(:,3), "--" )\n\n' );
```

Publication

Abstracts and Presentations

F Liu, RD Komistek *et al,* *In Vivo* 3D Quantification of Kinetics Characteristics in the Normal, Degenerative and Fused Cervical Spine, Science Exhibit, AAOS, San Diego 2007.
F Liu, RD Komistek *et al,* *In Vivo* Evaluation of Dynamic Characteristics of the Normal, Degenerative, Fused and Disc Replacement Cervical Spine, Poster, ORS, San Diego 2007 .
JS,Cheng, F Liu *et al*, Comparison of Cervical Spine Kinematics Using a Fluoroscopic Model for Adjacent Segment Degeneration, Oral, AANS/CNS Section, Phoenix 2007.
JS,Cheng, F Liu *et al*, Comparison of Cervical Spine Kinematics Using an *In Vivo* 3D Biomechanical Model for Adjacent Segment Degeneration, Poster, AANS/CNS Section, Phoenix 2007.
A Sharm, F Liu *et al,* In Vivo Contact Areas and Stresses for Multiple TKA Types, Oral, ISTA, Paris 2007.
DA Dennis, F Liu et *al,* Kinematic Difference Between Subjects Having Low and High Flexion at the Same Flexion Angles: A Multicenter Study, Oral, ISTA, Paris 2007.
D Glaser, F Liu *et al,* Experimental and Theoretical Analysis of Different Bearing Surfaces for THA, Poster, ISTA, Paris 2007.
D Glaser, F Liu *et al,* *In Vivo* Comparison of Hip Mechanics for Subjects Implanted With a MIS or Traditional Surgical Technique - Extended Study, Oral, ISTA, Paris 2007.
A Sharm, F Liu *et al,* Kinetic Performance Comparison for Traditional and High Flexion TKA, Oral, AAOS, San Diego 2007.
F Liu, RD Komistek *et al,* *In Vivo* 3D Dynamic Determination of Normal, Degenerative and Fused Cervical Spines, Oral, ISTA, New York 2006.
F Liu, RD Komistek *et al,* *In Vivo* Dynamic Comparison of Normal, Degenerative, Fused And Disc Replacement Cervical Spines, Science Exhibit, AAOS, Chicago 2006.
F Liu, RD Komistek *et al,* Automatically Image Controlled Mobile Fluoroscope System, Oral, WCB, Munich 2006.
F Liu, RD Komistek *et al,* *In Vivo* Kinetic Analysis of Normal, Degenerative, Fused and Disc Replacement Cervical Spine, Oral, WCB, Munich 2006.
F Liu, RD Komistek *et al,* *In Vivo* Kinematic and Kinetic Comparison of Normal and Fused Cervical Spines, Oral, ISTA, Kyoto 2005.

Peer-reviewed journal articles

Fei Liu, JS,Cheng *et al,* *In Vivo* Evaluation of Dynamics Characteristics of the Normal, Degenerative, Fused and Disc Replacement Cervical Spines, Spine, 2007 (In press).
Adrija Sharm, Fei Liu *et al,* *In Vivo* Patellofemoral Forces in High Flexion Total Knee Arthroplasty, Journal of Biomechanics, 2007 (Accepted).

JS,Cheng, Fei Liu *et al,* Comparison of Cervical Spine Kinematics Using a Fluoroscopic Model for Adjacent Segment Degeneration, Journal of neurosurgery, Spine, 2007 (In press).

Wang Zhen, Fei Liu *et al,* A New Custom-Made Artificial Articular Cartilage of Femoral Condyle Based on Rapid Prototyping Technique: A Case Report, Chinese Journal Of Surgery, 2004 Vol.42 No.12 P.746-749.

Wang Zhen, Fei Liu *et al,* Fabrication of Custom-Made Artificial Semi-Knee Joint Based on Rapid Prototyping Technique: Computer-Assisted Design and Manufacturing, Chinese Journal of Reparative and Reconstructive Surgery, 2004 Vol.18 No.5 P.347-351.

Yong Teng, Fei Liu *et al,* Fabrication of Custom-Made Artificial Semi-Knee Joint Based on Rapid Prototyping Technique: Three-Dimensional Reconstruction of Femoral Condyle, Chinese Journal of Reparative and Reconstructive Surgery, 2004 Vol.18 No.4 P.257-360.

Yong Teng, Fei Liu *et al,* Custom-Made Implant Study Based on Rapid Prototyping Technique, Orthopedic Journal of China, 2004,12(14):1077-1079.

Fei Liu, Dichen Li *et al,* Rapid Fabrication of Costumed Artificial Human Joint Based on Rapid Prototyping, China Mechanical Engineering, 2003, 14(16):1422～1424.

Fei Liu, Dichen Li *et al,* Research on the Apply of Reverse Engineering to Costumed Artificial Joint Based on CT Images, Journal of Xi'an Jiaotong University, Vol.36, No.9, 2002.

Yong Teng, Fei Liu *et al,* Three-Dimensional Reconstruction of Bone Contour for Manufacturing Artificial Bioactive Bone, Journal of the Fourth Military Medical Unversity,Vol.22,No.22,Nov.30 2001.

Vita

Fei Liu was born in Changsha, Hunan, China, on October 5th, 1975. He never thought that he would leave his hometown and went to a university when he was 19 years ago. His dream was to be a good worker in his hometown. After graduating from Xi'an Technology University, China with his B.S in 1998 and Xi'an Jiaotong University, China with his M.S in 2002 both from Department of Mechanical Engineering, he worked as a Visiting Scholar at Hong Kong University of Science and Technology in 2002, where he had one of his happiest times there in his life. After a year or so, he decided to come to US to start his Ph.D program in biomedical field in fall 2003. He graduated with a Ph.D major in Biomedical Engineering and minor in Statistics, in Dec, 2007.

After having been worked so hard for more than ten years in the universities, he will probably spend most of his time in enjoying life, hiking, swimming, playing tennis, painting, and playing piano. However, whatever he will do now and in the future, he will never forget using his precious degrees to help other people having a better life.