To the Graduate Council:

I am submitting herewith a dissertation written by Aiman Fateh Amer entitled "Uncertainty Analysis of Multilayer Feedforward Networks Using Generalized Adjoint and Model Adjustment Theories for Nuclear Reactor Physics Applications." I have examined the final 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 Nuclear Engineering.

L. F. Miller, Major Professor

We have read this dissertation
and recommend its acceptance:

Accepted for the Council:

Vice Provost
and Dean of The Graduate School

UNCERTAINTY ANALYSIS OF MULTILAYER FEEDFORWARD NETWORKS USING GENERALIZED ADJOINT AND MODEL ADJUSTMENT THEORIES FOR NUCLEAR REACTOR PHYSICS APPLICATIONS

A Dissertation
Presented for the
Doctor of Philosophy
Degree

The University of Tennessee, Knoxville

Aiman Fateh Amer

December 1991

DEDICATION

To my father and mother, my wife, and my brothers and sisters with love, amity, and sincerity.

ACKNOWLEDGEMENTS

In the course of my dissertation research, a number of people have assisted me generously. In particular, I am deeply indebted to Dr. L. F. Miller, who supported my efforts in the field of uncertainty analysis and for his detailed and scholarly comments on this dissertation as it progressed. With his consultation, criticism, and encouragement, I am also indebted to Dr. R. E. Uhrig, Dr. R. B. Perez, and Dr. Y. Kuo, the other members of my committee, for their time and patience in reading the drafts and for their helpful comments. I would also like to thank Dr. T. W. Kerlin for his comments, suggestions, and encouragement.

My gratitude also goes to Mr. Ahmed E. Al-Haddad and Mr. Raed Hajeer for their help and technical support and to Ms. Cindy Eldridge for her help in typing and editing the final manuscript.

ABSTRACT

Sensitivity and uncertainty analyses are leading characteristics of any system performance assessment. The role of sensitivity analysis is to provide a quantitative measure of the effect of system parameters upon key performance indices. It also limits the scope of uncertainty analysis by providing a ranking of the uncertain parameters with respect to their contribution to system output uncertainty. Uncertainty analysis provides a quantitative measure of the combined impact of parameter uncertainties on the output responses of a system. It also supports reliability studies, insures compliance with regulatory criteria, and helps identify important research and development needs.

A study of uncertainty analysis, based on sensitivity methodology, of multilayer feedforward neural networks using generalized adjoint and model adjustment theories is presented. Nuclear reactor physics and mathematical model applications for data field sets were evaluated in this study. These model applications include Reactor Noise Analysis (RNA) model, nuclear in-core fuel management, and Positron Emission Tomography (PET) model. The Oak Ridge National Laboratory (ORNL) large-scale sensitivity code system "GRESS," which incorporates the adjoint theory methodology, has been used in this research to study the effects of given input and weight vector parameters on the responses of interest, and to report model sensitivity functions required for uncertainty analysis. A general adjustment algorithm of multilayer feedforward networks using a model adjustment methodology was developed. Uncertainty and data adjustment analyses associated with model output predictions of neural networks relative to input data field and model parameter covariances were reported. It was concluded that the uncertainty analysis as well as the use of data adjustment method for improving the response uncertainties have shown a

considerable amount of improvement in the calculation of uncertainties of adjusted responses of interest. Hence, the adjustment algorithm can be a useful tool for improving the learning rate of multilayer feedforward networks.

TABLE OF CONTENTS

CHAPTER			PAGE
I.	INTRODUCTION		1
II.	LITERATURE REVIEW		6
	2.1	Introduction	6
	2.2	Differential Sensitivity Theory	6
	2.3	Response Surface Methodology	9
	2.4	Monte Carlo Method	10
	2.5	Fourier Amplitude Sensitivity Test (FAST) Method	11
	2.6	Existence of Solutions for General Mapping in Neural Networks	12
	2.7	Summary	13
III.	MATHEMATICAL BACKGROUND OF GENERAL SENSITIVITY AND MODEL ADJUSTMENT THEORIES		14
	3.1	Introduction	14
	3.2	General Sensitivity Theory (GST) Methods	14
		3.2.1 Graphical Method	15
		3.2.2 Statistical Method	15
		3.2.3 Perturbation Method	16
		3.2.3.1 Parameter Sensitivity	19
		3.2.3.2 Sensitivity Function	20
		3.2.3.3 Sensitivity Index	22
	3.3	Generalized Adjoint Theory (GAT) Methodology	23
	3.4	Model Adjustment Theory (MAT) Method	31
	3.5	Summary	40
IV.	DESCRIPTION OF AUTOMATED SYSTEMS PROCEDURE		41
	4.1	Introduction	41
	4.2	GRESS Methodology	41
	4.3	ADGEN Methodology	47
	4.4	Summary	49
V.	MULTILAYER FEEDFORWARD NEURAL NETWORKS		52
	5.1	Introduction	52
	5.2	Multilayer Feedforward Networks Fundamentals	52
	5.3	A Neural Network in Operation	58
	5.4	Learning and Training	59
	5.5	Summary	60

TABLE OF CONTENTS (Continued)

CHAPTER PAGE

VI. APPLICATION OF DATA ADJUSTMENT FOR UNCERTAINTY ANALYSIS OF NEURAL NETWORKS 61

6.1 Introduction 61
6.2 Data Adjustment for Multilayer Feedforward Networks 61
6.3 Description of the Adjustment Algorithm 65
6.4 Summary 67

VII. DISCUSSION OF SENSITIVITY, UNCERTAINTY, AND DATA ADJUSTMENT RESULTS 69

7.1 Introduction 69
7.2 Model Sensitivity Analysis 69
7.2.1 Sample Problems Description 70
7.2.2 Processing MFN.FOR with GRESS/ADGEN 71
7.2.3 Report of Sensitivities and Gradients 73
7.2.3.1 Sensitivity Analysis of RNA Model 73
7.2.3.2 Sensitivity Analysis of In-Core Fuel Management Model 74
7.2.3.3 Sensitivity Analysis of PET Model 87
7.2.4 Verification of Sensitivities 87
7.3 Uncertainty and Data Adjustment Analyses 94
7.4 Summary 96

VIII. CONCLUSIONS AND RECOMMENDATIONS 104

REFERENCES 107
APPENDICES 115

APPENDIX A Derivations of Some Equations in the Text 117
APPENDIX B 1. Input Data, Weight Vector Parameters, and Desired Outputs for RNA Model 123
2. Input Data, Weight Vector Parameters, and Desired Outputs for In-Core Fuel Management Model 131
3. Input Data, Weight Vector Parameters, and Desired Outputs for PET Model 149
APPENDIX C Some GRESS/ADGEN Utility Routines 161
APPENDIX D MFN.FOR Computer Code Program 165

VITA 180

LIST OF TABLES

TABLE		PAGE
1.	GRESS and ADGEN Terminology	50
2.	Data Sets Created During an ADGEN Application	72
3.	Gradient and Sensitivity Calculations of Selected Parameters of Interest of RNA Model for SIGC Output variable Using 9x5x1 Neural Network Architecture	75
4.	Gradient and Sensitivity Calculations of Selected Parameters of Interest of RNA Model for SIGC Output variable Using 19x3x1 Neural Network Architecture	78
5.	Gradient and Sensitivity Calculations of Selected Parameters of Interest of RNA Model for SIGC Output Variable Using 19x5x1 Neural Network Architecture	80
6.	Most Sensitive Parameters of RNA Model for the Three Neural Network Architectures	83
7.	Gradient and Sensitivity Calculations of Selected Parameters of Interest of In Core Fuel Management Model for SIGC Output Variable Using 31x15x31 Neural Network Architecture	84
8.	Most Sensitive Parameters of In-Core Fuel Management Model for 31x15x31 Neural Network Architecture	88
9.	Gradient and Sensitivity Calculations of Selected Parameters of Interest of PET Model for SIGC Output Variable Using 12x30x1 Neural Network Architecture	89
10.	Most Sensitive Input Data Parameters of PET Model for 12x30x1 Neural Network Architecture	93
11.	Correlation Matrix of Input Data Parameters In Standard Deviation Representation	95
12.	Effect of Adjustment on Response Values and their Uncertainties Using 9x5x1 Neural Network Architecture for RNA Model	97
13.	Effect of Adjustment on Response Values and their Uncertainties Using 19x3x1 Neural Network Architecture for RNA Model	98

LIST OF TABLES (Continued)

TABLE PAGE

14. Effect of Adjustment on Response Values and their Uncertainties Using 19x5x1 Neural Network Architecture for RNA Model 99

15. Effect of Adjustment on Response Values and their Uncertainties Using 31x15x31 Neural Network Architecture for In-Core Fuel Management Model 100

16. Effect of Adjustment on Response Values and their Uncertainties Using 12x30x1 Neural Network Architecture for PET Model 103

LIST OF FIGURES

FIGURE		PAGE
1.	Model source program and associated equations	43
2.	Gradient enhanced model and associated equations	44
3.	Flowchart of procedures for the use of GRESS system	46
4.	Flowchart of procedures for the use of ADGEN system	48
5.	A typical three-layer feedforward neural network architecture	54
6.	Topology of the generic processing element I_i [L]	55
7.	Sigmoidal function	57
8.	Topology of a multilayer feedforward ANS	62
9.	Flowchart of the adjustment algorithm	68

CHAPTER I

INTRODUCTION

The fields of sensitivity and uncertainty analyses have undergone significant growth in the past decade. Notable contributions to the theory of sensitivity and uncertainty methods have been made in various engineering and physics disciplines [1-13]. These disciplines include nuclear reactor physics, fusion reactor performance and parametric optimization studies, radiation transport physics and shield design studies, nuclear reactor system reliability and safety studies, reactor cross-section sensitivity analyses, and nuclear radioactive waste isolation and management programs.

Sensitivity analysis [14-26] attempts to determine the sensitivity (importance) of system response of interest and model sensitivity regions in comparison with system input data fields. Such an analysis requires (1) a definition of a system, (2) a description of the sensitivities of model response of interest relative to model input data fields, and (3) the characteristics of the most sensitive regions in a system under investigation. Uncertainty analysis [27-32], on the other hand, attempts to estimate the uncertainty (variation) in system response of interest (system output prediction) that results from the uncertainty in system input data field. Such an analysis, however, requires (1) a definition of a system, (2) a description of the uncertainties (variations) in model input data fields that are to be investigated, and (3) the characteristics of the system that is to be observed.

Although the Generalized Adjoint Theory (GAT) [33-35] is a well established sensitivity and uncertainty analysis methodology, the statistical [36-38] and perturbation techniques [39-40] have been the primary methods used to perform this type of analysis for Artificial Neural Systems (ANSs). In the statistical and perturbation approaches, the

perturbations (alterations and/or variations) of the input data fields are used to estimate the sensitivities and uncertainties associated with the calculated responses of interest (i.e., model sensitivities are calculated by direct numerical perturbations). This approach has serious drawbacks when applied to large-scale problems (e.g., nuclear reactor design, reactor safety, radioactive waste management, and neural networks). These drawbacks arise in the time-consuming nature of the statistical and perturbation calculations and of the inaccuracies that occur in handling all model data field parameters involved in a study.

An alternative approach to the statistical and perturbation methods is the adjoint theory methodology. The formulation of the adjoint theory allows systematic and quantitative screening of model data field parameter space which provides all the information necessary to perform a complete sensitivity and uncertainty analysis. Furthermore, the adjoint theory produces all model sensitivities for one model result of interest using only a single additional (i.e., adjoint) model calculation, regardless of how large the model data field or how complex the physical model including the nonlinear model problems.

The applicability of the adjoint theory has not enjoyed widespread use in computational systems. This is mainly because of the requirement of the large analytical calculational investment involved in the development of an adjoint model. This development (i.e., the calculations of very large numbers of partial derivatives in order to set up the adjoint model equations) requires considerable knowledge about the model equations and a long-term, high-level development effort.

To circumvent the development difficulties of using the adjoint method, two automated systems, GRadient-Enhanced Software System (GRESS) and ADjoint GENerator (ADGEN) [41-43], for solving the direct (forward) and adjoint (backward) model sensitivity problems were considered. These automated systems implement the adjoint theory

methodology and make use of the strengths of computer calculus to automate the costly and time-consuming calculations of derivatives in FORTRAN computer codes, and automatically generate derivatives, model sensitivities, and adjoint solutions of computer codes.

Model Adjustment Theory (MAT) [27,44,45] allows the simultaneous determination of model parameters (model fitting) and the adjustment of input data fields in order to obtain a set of adjusted data and parameters which are consistent with the model equations. The main advantage of this method, which uses the least squares principle, is the reduction of uncertainties due to the combination of information, namely the fitting (best fit estimates) and the adjustment procedures. Furthermore, the method estimates the uncertainties and covariances of residuals, model parameters, input data fields, and model responses of interest.

The proposed study will be devoted to uncertainty analysis, based on sensitivity methodology, of multilayer feedforward ANSs [46-48] using generalized adjoint and model adjustment theories. Nuclear reactor physics applications will be studied to evaluate the methodology. The ANS systems are being evaluated for use in nuclear power plants' reliability and maintainability applications [49-52]. Such applications are nuclear reactor diagnostics, regulations, and safety systems, along with nuclear in-core fuel management and automated optimal control of the next-generation nuclear power plants.

The adjoint theory as well as the capability of the automated systems procedure (GRESS and ADGEN) for performing model sensitivity and uncertainty analyses have been successfully tested and implemented when applied to various reactor physics studies [53-61]. Rall, Oblow, and Worley [62-64] were instrumental in the development of software packages to enhance FORTRAN coding programs with analytical differentiation of arithmetic statements. The procedure is embodied in a FORTRAN precompiler called EXAP (EXtended Arithmetic Processor) [43], which automatically processes computer models and

adds derivative-taking capabilities to the calculated responses of interest. These partial derivatives (the rate of change of model response of interest with respect to model input data field parameters) can be utilized to form the adjoint matrix for performing direct and/or adjoint calculations of model sensitivity functions. The calculational efficiency, capability, and applicability considerations of the adjoint theory demonstrate that the automated systems procedure, GRESS and ADGEN, are validated and their techniques are practical and feasible.

The originality of this piece of work lies in the fact that the GAT and MAT methodologies for evaluating sensitivities and uncertainties of various engineering and physics models have never been applied in the literature to ANS systems. This study will be accomplished by adapting the existing procedure of GRESS and ADGEN automated systems. It will develop a methodology for assessing the uncertainty analysis in model output predictions associated with any neural network and its input data field. It will also provide a new learning algorithm for various applications of neural networks.

The objectives of this dissertation are to determine model sensitivity and uncertainty analyses of feedforward neural networks using GAT and MAT methodologies and to develop a method for adjusting the input data and weight vector parameters of multilayer feedforward networks. The following tasks have been identified to accomplish these objectives:

(1) Develop a computer code for simulating a three-layer feedforward neural network.

(2) Obtain a GRESS version of the code for evaluating model derivative and sensitivity analyses. Three model applications [65-67] for data field sets evaluated in this study are Reactor Noise Analysis (RNA) model, nuclear in-core fuel management, and Positron Emission Tomography (PET) model.

(3) Develop a model adjustment method for multilayer feedforward networks. This methodology is based on a least squares principle that permits adjustment in model parameters and input data fields.

(4) Determine the uncertainty associated with model output predictions of neural networks relative to input data field and model parameter covariances.

(5) Develop a general adjustment algorithm of multilayer feedforward networks for various reactor physics applications.

Chapter II of this dissertation is devoted to the literature survey and review. Chapter III addresses the theoretical background of general sensitivity and model adjustment methods. The direct and adjoint sensitivity formalisms will be derived therein. A description of the automated systems procedure (GRESS and ADGEN) for performing model derivative and sensitivity studies will be outlined in Chapter IV. Chapter V contains an introductory presentation of the fundamental principles and mathematical concepts of multilayer feedforward networks as part of the ANS systems. Chapter VI introduces an application of the model adjustment theory to multilayer feedforward networks. A general adjustment algorithm for improving the learning rate of multilayer feedforward networks will be outlined therein. The discussion of results which includes a complete characterization of multilayer feedforward networks for nuclear reactor physics applications (i.e., model derivatives, sensitivities, and uncertainties) will be given in Chapter VII. Finally, Chapter VIII contains summary, conclusions, and recommendations for future work study.

CHAPTER II

LITERATURE REVIEW

2.1 INTRODUCTION

In dealing with models that depend upon uncertain parameters and variable data fields, the key performance indices determine the identification of input data fields that most affect the model output predictions (responses of interest). It is important, from the behavior of systems viewpoint, to evaluate quantitatively the effects of variations (alterations) in model input data fields relative to model output predictions using sensitivity and uncertainty analyses. This chapter will survey and review the literature that contributes to the theory, methods, and applications of the main approaches to model sensitivity and uncertainty analysis techniques. The main approaches are differential sensitivity theory, response surface methodology, Monte Carlo analysis, and Fourier Amplitude Sensitivity Test (FAST) method. The last section of this Chapter will discuss the existence of solutions for general mapping in neural networks.

2.2 DIFFERENTIAL SENSITIVITY THEORY

Differential sensitivity theory is based on expanding the model output predictions as Taylor series and then assessing the sensitivity (importance) of model variables from the functions (i.e., sensitivity functions) in these series representations. Oblow [19] derived sensitivity functions in terms of functional derivatives to interpret sensitivity model results from a differential viewpoint. Using this approach, sensitivity functions can be represented as "exact" differential rates of change of calculated responses of interest with respect to

differential changes in model input data fields used to calculate model output predictions. The physical insight gained with this approach focuses on a differential property of a system and eliminates the need of having made perturbations (or variations) to find out how the responses of interest are affected. In addition, the differential formulation allows simple model calculations to be analyzed easily since only functional derivatives have to be calculated to determine the sensitivities of interest.

Greenspan et al. [40] extended and generalized the first-order perturbation theory (FOPT) formulations (derived by Oblow) to account for the effects of the second-order perturbation theory (SOPT) formulation for source-driven (i.e., inhomogeneous) systems. They determined the applicability of the SOPT formulation by applying it to a shield optimization problem. They concluded that the SOPT formulation provides more accurate predictions and more efficient algorithms for optimization studies (applied to a shield design problem and optimization analysis of fusion reactor neutronic characteristics) than the FOPT formulation. Furthermore, they recommended that the SOPT formulation offers a more powerful approach for the potential application of perturbation theory in the following fields of study:

(1) Cross-section sensitivity studies by providing an adequate answer for the effects of model uncertainties in resonance cross sections in neutron deep penetration problems;

(2) Cross-section sensitivity analysis in altered (perturbed) systems by investigating the uncertainties in the responses of interest (model output predictions) caused by a given model input alteration (perturbation); and

(3) System design and analysis by developing a tool that is efficient for investigating the combined effects of multiple variations, especially when many

different combinations of model variations are to be investigated.

Oblow [2,8,12] explored the possibility of applying differential sensitivity theory to a large set of equations governing the physical behavior of nuclear reactor systems. Accordingly, he developed a sensitivity theory based on the differential formulation to address model sensitivity analysis of reactor thermal-hydraulics problems, radiation transport problems, and cross-section sensitivity studies. He argued that differential sensitivity theory has some significant advantages in evaluating model sensitivities and uncertainties. These advantages are (1) the differential formulation of the sensitivity theory is global in nature, that is, the sensitivity of model response of interest to all model input data fields can be studied without a prior choice of model data field parameters being made, and (2) the differential approach is fast and efficient since only one additional set of equations needs to be solved for each model output studied. Williams [5] developed a perturbation and sensitivity theory for the nonlinear reactor fuel burnup model equation describing the time-dependent behavior of the neutron and nuclide fields in reactor core physics. Chen [25], in his dissertation, developed a sensitivity and uncertainty analysis methodology for the transport of radionuclides in a saturated-unsaturated groundwater flow system based on a Generalized Perturbation Theory (GPT) formulation. Alsharif et al. [26] applied various methods of model sensitivity analysis (e.g., graphical, perturbation, and statistical approaches) to the model of solute radionuclide transport through planar fractured media coupled with diffusion into the adjacent matrix. They concluded that various methods of model sensitivity analysis (i.e., graphical, perturbation, statistical, and adjoint methods) can be applied either to analytical or numerical models and the appropriate method will be chosen by the user depending upon the number of model input data field parameters, simplicity of the method, cost efficiency, and level of information desired from the model sensitivity analysis. Cacuci [20,21,35] derived a

deterministic sensitivity theory for nonlinear systems by considering the forward and adjoint methods. Particularly, the adjoint theory was derived in the literature from different approaches. Pomraning and Stacey [16,17] used the variational methods, Greenspan et al. [39] used the perturbation theory including the generalized perturbation theory, and Oblow and Cacuci [19,35] used the differential sensitivity theory for setting up the model adjoint equation.

In the field of numerical computation, Oblow and Worley et al. [63,64] developed two automated systems procedure for evaluating model sensitivity and uncertainty analysis using computer calculus. These automated systems are the GRESS (Gradient-Enhanced Software System) and ADGEN (Adjoint Generator) methodologies for solving the direct and adjoint model sensitivity problems. The applications of GRESS and ADGEN automated systems to various engineering and physics disciplines will be found in References [53-61].

2.3 RESPONSE SURFACE METHODOLOGY

The response surface methodology [18] is based on using experimental design theory to select model input data field for a given system and then using model output predictions (associated with model input data field) to construct a response surface replacement for the system (i.e., the behavior of the model output prediction as a function of the model input data field), which is fit with a simple functional form for later use. This replacement system is then used as a representative for the original system in subsequent model sensitivity and uncertainty analyses. For the case of computer simulations, model parametric and sensitivity analyses are usually based on the response surface methodology where a set of reference input data field for a computer simulation is first chosen for study. To perform system

parametric analysis [25], a specified model input data field is varied over a certain value range while holding other parameters at their reference values. The computer code is then run to compute the changes made in the calculated responses of interest using the altered sets of model input data fields. To perform model sensitivity analysis, on the other hand, the relative sensitivity function of model response of interest to the specified model input data field is obtained from the ratio of percent change in the calculated responses of interest to percent change in the specified model input data field parameter about its reference value. It is assumed that the model response of interest is linear with respect to model input data field parameter near the reference point. Hence, the response surface can be constructed by assembling the calculated responses of interest from repetitive runs of the computer code with varied input data field parameters. Model sensitivities and uncertainties of the calculated responses of interest to model input data field parameters can then be derived from the functional form of the response surface. As an application to model sensitivity problems based on the response surface methodology, Dickens and Lennox [22] performed a sensitivity analysis on the problem of transient, two-dimensional transport of radionuclide contaminants in a steady saturated groundwater flow system. The simulation (using the Galerkin finite-element method) was repeated by varying model input data field value of the dispersivity while holding other field parameters fixed at their reference values.

2.4 MONTE CARLO METHOD

The Monte Carlo analysis [37] is based on randomly and repeatedly sampling model input data field distribution to calculate model output prediction distribution by constructing a Probability Distribution Function (PDF) (associated with model output prediction). This

can be implemented by rerunning the test model for each sample set of model input data field parameters. The probability distributions of model input data field and parameter correlations (combinations) are handled in the sampling procedure using statistical laws. As the number of sampling sets increases, the difference between the calculated and "true" model output distribution diminishes. Stucliffe [29] proposed a method of model uncertainty analysis which is represented by a probability distribution in the form of a histogram. In this approach, he discretized a probability distribution of model input data field into histograms and evaluated model output predictions for all possibilities of parameter correlations to form a probability tree. However, this method does not rely on random sampling and the probability distributions are propagated in probability trees by simple multiplication. The histogram probability distributions are not actually propagated, but rather mean or endpoint parameter values are used. He concluded that the histogram (probability tree) method for evaluating model uncertainties of nuclear waste package development program facilitates the separation of the probability distribution calculations from the evaluations of model output predictions. Helton, Iman, and Johnson, et al. [9] illustrated the application of model sensitivity and uncertainty techniques based on a Monte Carlo latin hypercube sampling (LHS) procedure and regression analysis to aggregation problems. They concluded (see References 9, 11, and 38) that Monte Carlo analysis, based on latin hypercube sampling procedure for evaluating model sensitivities and uncertainties of various aggregation problems is a reasonably straightforward and efficient approach.

2.5 FOURIER AMPLITUDE SENSITIVITY TEST (FAST) METHOD

The Fourier Amplitude Sensitivity Test (FAST) method [23,24] is based on the

utilization and implementation of Fourier series representation to approximate a given test model. This procedure is suited to the task of determining model sensitivity and uncertainty analysis of nonlinear mathematical, physical, and chemical systems subjected to variations of arbitrary size in either the system parameters or initial conditions. The FAST technique associates each uncertain parameter with a specific frequency in the Fourier transform space of the system. Model sensitivities are then determined by solving the system equations for discrete values of the Fourier transform variable and then computing the Fourier coefficients associated with each parameter frequency.

Cukier et al. [23] applied the FAST method to three models, namely a chemical laser model, a chemical reaction model, and an economic model. They concluded that the method allows arbitrarily large deviations in either system parameters or initial conditions. McRae et al. [24] have implemented the FAST method into a computational algorithm for determining model sensitivity and uncertainty analysis of a chemical engineering problem. They pointed out that the FAST method allows nonlinear sensitivities and uncertainties of mathematical, physical, and chemical systems, subjected to large parameter variations, to be determined in a practical and efficient manner. However, this method has not enjoyed widespread use in the literature mainly because of the implementation's difficulty and application's complexity associated with model sensitivity and uncertainty studies.

2.6 EXISTENCE OF SOLUTIONS FOR GENERAL MAPPING IN NEURAL NETWORKS

The capability of sufficiently complex multilayer feedforward networks to approximate an unknown mapping $f: R^n \rightarrow R^m$ arbitrarily well has been recently investigated by Cybenko

[68] for sigmoidal hidden layer activation functions and by Stinchcombe et al. [69] for non-sigmoidal hidden layer activation functions. Independently, they demonstrated that neural networks with only one hidden layer and arbitrary continuous sigmoidal and/or non-sigmoidal functions can approximate continuous mapping with arbitrary precision provided that no constraints on the number of nodes or size of weight vectors. Hornik et al. [70] have given conditions ensuring that multilayer feedforward networks with as few as a single hidden layer and an appropriately smooth hidden layer activation function are capable of arbitrarily accurate approximation to an arbitrary mapping function and its derivatives simultaneously.

2.7 SUMMARY

In this chapter the main approaches in the literature for evaluating model sensitivity and uncertainty analyses (i.e., differential sensitivity theory, response surface methodology, Monte Carlo analysis, and Fourier Amplitude Sensitivity Test (FAST) method) along with the existence of solutions for general mapping in multilayer feedforward networks have been discussed and reviewed.

CHAPTER III

MATHEMATICAL BACKGROUND OF GENERAL SENSITIVITY AND MODEL ADJUSTMENT THEORIES

3.1 INTRODUCTION

Sensitivity theory [14-21,35,53] has been used by researchers in many engineering and physics fields to assess confidence levels, importance of variations, and model uncertainties in the design of input data field parameters on calculated responses of interest. The importance of this theory lies in its methodological diversity and wide ranging applicability in determining model sensitivities and their regions (i.e., sensitivity regions) of analytic and calculated responses of interest relative to model input data fields.

This chapter will discuss several approaches to General Sensitivity Theory (GST) methods, but emphasis will be placed on the implementation of one particular approach, the Generalized Adjoint Theory (GAT) methodology. The second part of this chapter will describe the Model Adjustment Theory (MAT) method which uses model derivative (gradient) information as a basis to allow the simultaneous determination of model parameters and the adjustment of input data fields in order to obtain a set of adjusted data and parameters consistent with the model equations.

3.2 GENERAL SENSITIVITY THEORY (GST) METHODS

Four approaches for performing sensitivity analyses of a given physical test model are as follows: (1) graphical method, (2) statistical method, (3) perturbation method, and (4) adjoint method. These methods can be applied to both analytical and numerical design

models. The perturbation and adjoint approaches, however, require the analytical differentiability of the test model equations.

3.2.1 Graphical Method

In the graphical method [26], an input data field parameter variation is performed to analyze the sensitivities and important regions of model output predictions. A given test model can be executed for different values of a single input data field parameter, while keeping the rest of parameters fixed at their reference values. The calculated responses of interest are then plotted as "breakthrough" curves so that the sensitivity of model output prediction relative to model uncertainty (variation) of input data field parameters can be graphically analyzed. This illustration aids in determining model changes of the breakthrough curves as the values of input data field parameters are either increased or decreased gradually. The graphical method is considered to be the simplest method of model sensitivity analysis.

3.2.2 Statistical Method

In this approach, the test model is evaluated to obtain model sensitivities of performance measure values (i.e., responses of interest) and their probability distribution functions (PDFs) for all combinations (correlations) of several values of each of the input data field probability distributions. Consequently, the analytical propagation of input uncertainties through calculational models is a difficult task due to mapping probability density funcitons from a multidimensional space of input parameters to one-dimensional output distribution function [30]. To remedy this problem, the most common approach is to

randomly sample the input distributions and then calculate the model output of interest. That is, constructing a probability distribution of the model output of interest by rerunning the calculational test model for each sample set of input data field parameters. In the sampling procedure, the input data field probability distributions and any paramter correlations are handled in a statistical sense. The sampling procedure will then lead to an output distribution function (calculated output) representing the probability disturibution of model output of interest (acutal output) that results from the actual propagation of input probability distributions. As the number of sampling sets increases, the difference between the calculated and "true" output distribution functions diminishes.

Although the statistical method has the advantage of assessing model sensitivities to changes to each input data field probability distribution, this approach has a serious drawback. This drawback arises if the number of input data field parameters is large, then the number of runs (executions) of the test model needed to assure a large enough statistical sample will be accordingly very large. Hence, if this is the case, the statistical method of model sensitivity analysis will be surrounded with computational difficulties in the form of cost-inefficiency and time-consuming nature.

3.2.3 Perturbation Method

In the perturbation approach to model sensitivity analysis, each input data field parameter of the test model is perturbed (altered) by a small amount while keeping all other parameters at their preselected reference values, and quantifying the relative sensitivity effects on model output predictions. This approach can be applied to various systems described by model equations that can be differentiated analytically. The sensitivity analysis of a test model

from a perturbation theory viewpoint can be performed by computing three quantities: (1) parameter sensitivity, (2) sensitivity function, and (3) sensitivity index. The perturbation theory is based on establishing an analytic relationship between the response of interest and the input data field. The simplest form of this relationship is derived from a first-order Taylor series expansion of the response of interest about a particular calculated design point which is symbolized by the vector of all input data fields at a particular point ξ_0. This relationship can be written out to *nth* order to determine the finite changes in R (system response of interest) with respect to changes in $\xi(r,t)$ as [40]

$$
\begin{aligned}
R \equiv R[\,\xi(r,t)\,] &= R[\xi_0 + \delta\xi(r,t)\,] \\
&= R(\xi_0) + \sum_{i=1}^{I} \int_{r_i}\int_{t_i} \left(\frac{\partial R}{\partial \xi_i(r_i,t_i)}\right) \delta\xi_i(r_i,t_i)\, dr_i\, dt_i \\
&+ \frac{1}{2!}\sum_{i=1}^{I}\sum_{j=1}^{J} \int_{r_i}\int_{t_i}\int_{r_j}\int_{t_j} \left(\frac{\partial^2 R}{\partial\xi_i(r_i,t_i)\;\partial\xi_j(r_j,t_j)}\right) \\
&\times [\delta\xi_i(r_i,t_i)\;\delta\xi_j(r_j,t_j)]\, dr_i\, dt_i\, dr_j\, dt_j + \cdots \\
&+ \frac{1}{n!}\sum_{i=1}^{I}\sum_{j=1}^{J}\cdots\sum_{n=1}^{N} \int_{r_i}\int_{t_i}\int_{r_j}\int_{t_j}\cdots \\
&\times \int_{r_n}\int_{t_n} \left(\frac{\partial^n R}{\partial\xi_i(r_i,t_i)\;\partial\xi_j(r_j,t_j)\ldots\partial\xi_n(r_n,t_n)}\right) \\
&\times [\delta\xi_i(r_i,t_i)\;\delta\xi_j(r_j,t_j)\;\cdots\;\delta\xi_n(r_n,t_n)] \\
&\times dr_i\, dt_i\, dr_j\, dt_j \cdots dr_n\, dt_n\,,
\end{aligned}
\tag{3-1a}
$$

where

$$R \equiv \int_r\int_t F[\xi(r,t)]\, dr\, dt,$$

= system response of interest which is an explicit function of the J independent system input data field variables, $\xi = (\xi_1, \xi_2, \xi_3, ..., \xi_J)$, and

$$f(x) = \sum_{n=0}^{\infty} \frac{1}{n!} \left(\frac{d^n f(x)}{dx^n} \right)_{x=x_0} (x - x_0)^n = \text{Taylor series expansion.}$$

Eq. (3-1a) can be rewritten to second-order accuracy in vector form as

$$R \equiv R\,[\xi(r,t)] = R\,[\xi_0 + \delta\xi(r,t)] \approx R(\xi_0) + \int\int_t \left(\frac{dR}{d\xi(r,t)} \right)_{\xi = \xi_0} \cdot \delta\xi(r,t)\, dr\, dt, \qquad (3\text{-}1b)$$

where

$\xi(r,t)$ = $(\xi_0 + \delta\xi(r,t))$,
= vector of all input data field parameters,

ξ_0 = vector of all input data field parameters at a particular calculated design point,

$\delta\xi(r,t)$ = vector of finite changes in the input data field parameters as a function of the problem phase space (i.e., space and time),

$R(\xi_0)$ = response of interest evaluated at ξ_0,

$R[\xi_0 + \delta\xi\,(r,t)]$ = response of interest evaluated with the altered input data field set, and

$\left(\frac{dR}{d\xi(r,t)}\right)_{\xi = \xi_0}$ = vector of parameter sensitivity derivatives for each separate ξ of the input data field evaluated at $\xi = \xi_0$.

Rearranging Eq. (3-1) yields a linear perturbation theory in the following form

$$\frac{\delta R}{R} \equiv \frac{R(\xi_0 + \delta\xi(r,t)) - R(\xi_0)}{R(\xi_0)} = \int\int_{r,t} S_\xi(r) \cdot \frac{\delta\xi(r,t)}{\xi(r,t)}\, dr\, dt, \tag{3-2}$$

where

$S_\xi(r)$ = vector of sensitivity functions, each element of which has the form $\left[\left(\frac{dR}{R}\right)/\left(\frac{d\xi_j}{\xi_j}\right)\right]$, and

$\frac{\delta\xi(r,t)}{\xi(r,t)}$ = fractional perturbation vector with elements $(\delta\xi_j/\xi_j)$ as a function of the problem phase space.

The derivatives and sensitivities of Eqs. (3-1) and (3-2) were taken as functional derivative form and integrated over all phase space (r,t). The calculated responses of interest caused by any arbitrary change in the input data field can be estimated using Eq. (3-2).

From the previous relationships (Eqs. (3-1) and (3-2)), the parameter sensitivity, sensitivity function, and sensitivity index can be explained as follows:

3.2.3.1 Parameter Sensitivity

Parameter sensitivity is defined as the sensitivity of the response of interest R, i.e.,

model output prediction, due to a perturbation (variation and/or alteration) in the input data field parameter ξ at a given point. The model sensitivity can change from one point to another and, for a given perturbation $d\xi$, the resulting change in R (i.e., dR) can be much larger than $d\xi$ (if $dR/d\xi$ is large) or much smaller than $d\xi$ (if $dR/d\xi$ is small). Hence, parameter sensitivity, denoted as R_ξ, is expressed as (see Eq. (3-1))

$$R_\xi \equiv \frac{\partial R}{\partial \xi(r,t)} . \tag{3-3}$$

3.2.3.2 Sensitivity Function

Sensitivity function, $S_\xi(r)$, is defined as the ratio of fractional change of the response function of interest, dR/R, to the fractional change of the input data field parameter, $d\xi/\xi$. That is

$$S_\xi(r) \equiv \left[\left(\frac{dR}{R}\right)/\left(\frac{d\xi}{\xi}\right)\right] . \tag{3-4a}$$

The sensitivity function, $S_\xi(r)$, is a dimensionless quantity and represents the fractional change in model output prediction to the fractional change in model input data field. With all other parameters fixed, Eq. (3-4a) can be written as

$$\frac{dR}{R} = S_\xi(r) \frac{d\xi(r,t)}{\xi(r,t)} , \tag{3-4b}$$

or

$$dR = S_\xi(r) \frac{R}{\xi(r,t)} d\xi(r,t) , \tag{3-4c}$$

or simply (see Eq. (3-4a))

$$dR = \left(\frac{\partial R}{\partial \xi(r,t)}\right) d\xi(r,t) = R_\xi \, d\xi(r,t) . \tag{3-4d}$$

Thus, the sensitivity function can be written as (see Eq. (3-4b))

$$S_\xi(r) = \left(\frac{\partial R}{\partial \xi(r,t)}\right) \frac{\xi(r,t)}{R} = R_\xi \frac{\xi(r,t)}{R} . \tag{3-4e}$$

This means that the expression of parameter sensitivity, R_ξ, can be used directly to obtain the sensitivity function, $S_\xi(r)$, for each of the input data field parameters. These functions can then be evaluated and tabulated or plotted to find model sensitivity effects of responses of interest on input data field parameters.

The derivation of sensitivity function is based on the assumption that only one parameter is uncertain while all others are fixed. Generally speaking, if all parameters are uncertain then the expression for the total differential of the functional response of interest should be determined. Further treatment of sensitivity function (Eq. (3-4b)) expression yields that the total percentage change of the functional response of interest is the sum of all the changes caused by individual data field parameters, which is the parameter percentage change times its own sensitivity function. That is

$$\frac{dR}{R} = S_{\xi}(r) \sum_{j=1}^{J} \frac{d\xi_j(r,t)}{\xi_j(r,t)} \quad . \tag{3-4f}$$

3.2.3.3 Sensitivity Index

Sensitivity index technique [26] can be applied on various design models. In this technique, the reference values and data field will be evaluated prior to conducting model sensitivity analysis. Then sensitivity indexes are calculated by substituting the maximum and minimum values of a data field parameter, while holding all other parameters at their reference values, to produce maximum and minimum values of model output predictions. That is R_{max} and R_{min} are model output predictions for a maximum and a minimum value of a data field parameter, ξ, respectively. Hence, the sensitivity index is given by

$$SI_{\xi} = 1 - (R_{\xi\ min}/R_{\xi\ max}) \quad . \tag{3-5}$$

Generally speaking, a sensitivity index less than 0.01 indicates that the test model is insensitive to changes in the input data field parameters and a sensitivity index approaching 1.0 indicates higher sensitivity.

If the statistical nature of the input data field parameters is considered, then it is possible to form an ensemble of finite perturbations of the input data field about ξ_0. It is known that most of the input data fields are experimentally derived, then the information about the standard deviation of the distribution function of ξ about its nominal value, ξ_0, and correlation between various input data fields can be put into Eq. (3-2) to formulate a first-order uncertainty theory [2]. This procedure can be done by squaring Eq. (3-2) and taking

its expectation value over the correlated ensemble of statistical input data field variations to obtain

$$\sigma_R^2 = S_\xi^{tr} U_x S_\xi , \tag{3-6}$$

where

σ_R = standard deviation in the response of interest.

U_x = Correlation matrix with elements along the diagonal representing the standard deviations in the input data field and off-diagonal elements representing the correlations between the input data field. In the case of uncorrelated input data field, U_x is a diagonal matrix of standard deviations, and

S_ξ and S_ξ^{tr} = total sensitivity vector and its transpose.

The derivation of Eq. (3-6) will be found in Appendix A.

3.3 GENERALIZED ADJOINT THEORY (GAT) METHODOLOGY

Adjoint theory similar to perturbation theory, is applied to analytical differentiability of model equations. Unlike the perturbation theory, where the first model derivative ($dR/d\xi$) is calculated directly, the adjoint theory utilizes the fact that the equation of the first model derivative is linear and appropriate adjoint equations can therefore be developed specifically to evaluate the quantity $dR/d\xi$.

The mathematical description of the Generalized Adjoint Theory (GAT) applied to model sensitivity analysis applications is given here as an aid to understanding the problem

of applying this theory to mathematical models of physical systems that are to be programmed in computer algorithms. The approach used here relies to a large extent on the approach suggested by Oblow [13]. The example to be considered is a general set of non-linear model equations given by the three line code, for example

$$x = a^2 + b^2,$$

$$u = ax + c^2 + v,$$

$$v = x^2 + xu + a^3 + d^3.$$

This FORTRAN computer model defines a relationship between a left-hand side solution vector, Y, and the program data vector, Z, given by right-hand side functions, H. These model vectors can be represented as

$$Y = \begin{pmatrix} x \\ u \\ v \end{pmatrix}_{(R\times 1)}, \quad Z = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}_{(R\times 1)}, \quad H = \begin{pmatrix} a^2 + b^2 \\ ax + c^2 + v \\ x^2 + xu + a^3 + d^3 \end{pmatrix}_{(R\times 1)}.$$

These relationships can be expressed mathematically as

$$Y = H(Y, Z), \tag{3-7}$$

where

Y = vector of the dependent variables being solved for,

Z = vector of the user-specified model data field or parameter set, and

H = vector of the mathematical model equations.

The particular form of Eq. (3-7) represents, in vector form, the model set of equations and storage operations. The left-hand side of Eq. (3-7) (i.e., Y) represents the stored value of the

variable calculated from the functional formula on the right-hand side (i.e., $H(Y,Z)$). A typical response of interest will be defined as a function of the solution of Eq. (3-7) (i.e., Y) is

$$R = G(Y) , \tag{3-8}$$

where R is a single number that represents a model output prediction and G represents the known functional dependence of R on Y.

The generalized sensitivity theory accounts for the rate of change in the results of interest, R, arising from changes in any model input data field parameters, ξ_j. The set of parameters of interest can be thought of as $\xi = \{\xi_1, \xi_2, ..., \xi_J\}$, a component of the total parameter set vector Z. The quantity of interest is then the numerical value of $dR/d\xi_j$ given analytically by

$$\frac{dR}{d\xi} = \frac{dG(Y)}{d\xi} = \sum_{j=1}^{J} \frac{\partial G}{\partial Y} \frac{dY}{d\xi_j} , \tag{3-9}$$

where

$$\frac{\partial G}{\partial Y} = \begin{bmatrix} \frac{\partial G_1}{\partial Y_1} & \frac{\partial G_1}{\partial Y_2} & \cdots & \frac{\partial G_1}{\partial Y_N} \\ \frac{\partial G_2}{\partial Y_1} & \frac{\partial G_2}{\partial Y_2} & \cdots & \frac{\partial G_2}{\partial Y_N} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \frac{\partial G_N}{\partial Y_1} & \frac{\partial G_N}{\partial Y_2} & \cdots & \frac{\partial G_N}{\partial Y_N} \end{bmatrix} , (N \times N)$$

and

$$\frac{d\boldsymbol{Y}}{d\xi_j} = \begin{bmatrix} \frac{dY_1}{d\xi_1} & \frac{dY_1}{d\xi_2} & \cdots & \frac{dY_1}{d\xi_J} \\ \frac{dY_2}{d\xi_1} & \frac{dY_2}{d\xi_2} & \cdots & \frac{dY_2}{d\xi_J} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \frac{dY_N}{d\xi_1} & \frac{dY_N}{d\xi_2} & \cdots & \frac{dY_N}{d\xi_J} \end{bmatrix}, \ (N \times J)$$

for all $j = 1, 2, ..., J$.

The functional dependence of R on $\boldsymbol{Y}$ through $G(\boldsymbol{Y})$ is defined analytically by the user, therefore, only $d\boldsymbol{Y}/d\xi_j$ needs to be generated in order to evaluate Eq. (3-9). The procedure needed to get $d\boldsymbol{Y}/d\xi_j$ is to differentiate Eq. (3-7) as follows:

$$\frac{d\boldsymbol{Y}}{d\xi} = \frac{d\boldsymbol{H}(\boldsymbol{Y},\boldsymbol{Z})}{d\xi} = \sum_{j=1}^{J} \left[\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{Y}} \frac{d\boldsymbol{Y}}{d\xi_j} + \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{Z}} \frac{d\boldsymbol{Z}}{d\xi_j} \right], \tag{3-10}$$

where

$$\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{Y}} = \begin{bmatrix} \frac{\partial H_1}{\partial Y_1} & \frac{\partial H_1}{\partial Y_2} & \cdots & \frac{\partial H_1}{\partial Y_N} \\ \frac{\partial Y_2}{\partial Y_1} & \frac{\partial Y_2}{\partial Y_2} & \cdots & \frac{\partial Y_2}{\partial Y_N} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \frac{\partial H_N}{\partial Y_1} & \frac{\partial H_N}{\partial Y_2} & \cdots & \frac{\partial H_N}{\partial Y_N} \end{bmatrix}, \ (N \times N)$$

$$\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{Z}} = \begin{bmatrix} \frac{\partial H_1}{\partial Z_1} & \frac{\partial H_1}{\partial Z_2} & \cdots & \frac{\partial H_1}{\partial Z_M} \\ \frac{\partial H_2}{\partial Z_1} & \frac{\partial H_2}{\partial Z_2} & \cdots & \frac{\partial H_2}{\partial Z_M} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \frac{\partial H_N}{\partial Z_1} & \frac{\partial H_N}{\partial Z_2} & \cdots & \frac{\partial H_N}{\partial Z_M} \end{bmatrix}, \ (N \times M)$$

and

$$\frac{dZ}{d\xi_j} = \begin{bmatrix} \frac{dZ_1}{d\xi_1} & \frac{dZ_1}{d\xi_2} & \cdots & \frac{dZ_1}{d\xi_J} \\ \frac{dZ_2}{d\xi_1} & \frac{dZ_2}{d\xi_2} & \cdots & \frac{dZ_2}{d\xi_J} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \frac{dZ_N}{d\xi_1} & \frac{dZ_N}{d\xi_2} & \cdots & \frac{dZ_N}{d\xi_J} \end{bmatrix}, \ (N \times J)$$

for all $j = 1, 2, ..., J$.

Rearranging Eq. (3-10), yields the following set of coupled equations that solve for $dY/d\xi_j$

$$\left(I - \frac{\partial H}{\partial Y}\right) \sum_{j=1}^{J} \frac{dY}{d\xi_j} = \sum_{j=1}^{J} \frac{\partial H}{\partial Z} \frac{dZ}{d\xi_j} \ , \tag{3-11a}$$

or

$$\sum_{j=1}^{J} \frac{dY}{d\xi_j} = \left(I - \frac{\partial H}{\partial Y}\right)^{-1} \sum_{j=1}^{J} \frac{\partial H}{\partial Z} \frac{dZ}{d\xi_j} \ , \tag{3-11b}$$

Eq. (3-11a) can be represented in more compact form as,

$$A \ Y'_j = S_j \ , \quad j = 1, 2, ..., J \tag{3-12}$$

where I is the identity matrix and A, Y'_j, and S_j are given by

$$A \equiv \left(I - \frac{\partial H(Y,Z)}{\partial Y}\right) , \tag{3-13}$$

$$Y'_j \equiv \frac{dY}{d\xi_j} , \tag{3-14}$$

and

$$S_j \equiv \frac{\partial H}{\partial Z} \frac{dZ}{d\xi_j} . \tag{3-15}$$

If Eq. (3-11b) were substituted directly into Eq. (3-9) for $dY/d\xi_j$ (i.e., Y'_j) then $dR/d\xi_j$ could be evaluated. This method of sensitivity analysis called the "direct" or "forward" approach and is a classical methodology which has received a great deal of attention in the literature [13]. Since Eq. (3-12) must be solved each time a new data field parameter ξ_j is defined, the computational expense puts this method out of reach as a practical sensitivity tool. As a remedy, the adjoint approach is far more applicable where use is made of the fact that Eq. (3-12) is linear in Y'_j, and appropriate adjoint equations can therefore be developed specifically to evaluate Eq. (3-9).

Defining the matrix adjoint of A as A^* and using the mathematical definition of an adjoint operator,

$$\int_V P\, A\, Q\, dV = \int_V Q\, A^*\, P\, dV , \tag{3-16a}$$

or inner product notation,

$$\langle P, AQ \rangle = \langle A^* P, Q \rangle , \tag{3-16b}$$

or simply in vector notation,

$$P^{tr} A Q = Q^{tr} A^* P , \tag{3-16c}$$

where P and Q are arbitrary vectors and A^* is defined as

$$A^* = A^{tr} . \tag{3-17}$$

Here the *tr* superscript represents the transpose of the vector or matrix.

The model adjoint formulation to Eq. (3-12) can then be set up as follows:

$$A^* Y^* = S^* , \tag{3-18}$$

where

$$A^* \equiv A^{tr} = \left(I - \frac{\partial H(Y,Z)}{\partial Y} \right)^{tr} . \tag{3-19}$$

Choosing S^* appropriately as (see Eq. (3-9))

$$S^* \equiv \left(\frac{dG(Y)}{dY} \right)^{tr} , \tag{3-20}$$

then Eq. (3-9) can be evaluated, using the definition of an adjoint operator and inner product technique, as follows:

$$\frac{dR}{d\xi_j} = \left(\frac{\partial G}{\partial Y}\right)\frac{dY}{d\xi_j} = \langle S^{*\,tr}\,,\, Y'_j\rangle = \langle (A^*\,Y^*)^{tr}\,,\, Y'_j\rangle = \langle Y^{*\,tr}\,A^{*\,tr}\,,\, Y'_j\rangle$$
$$= \langle Y^{*\,tr}\,,\, A\,Y'_j\rangle = \langle Y^{*\,tr}\,,\, S_j\rangle \qquad (3\text{-}21)$$
$$= Y^{*\,tr}\left(\frac{\partial H}{\partial Z}\,\frac{dZ}{d\xi_j}\right), \quad j = 1, 2, \ldots, J,$$

where Y^* is now the solution to (see Eq. (3-18))

$$\left(I - \frac{\partial H}{\partial Y}\right)^{tr} Y^* = \left(\frac{dG}{dY}\right)^{tr}, \qquad (3\text{-}22a)$$

or simply,

$$Y^* = \left[\left(I - \frac{\partial H}{\partial Y}\right)^{tr}\right]^{-1}\left(\frac{dG}{dY}\right)^{tr}. \qquad (3\text{-}22b)$$

The simplicity of the adjoint methodology lies in the fact that Eq. (3-22) needs to be solved only once to get any and all model sensitivities in the problem. This is a consequence result of Eq. (3-22) being independent of the definition of ξ_j. The particular choice of the data field parameter, ξ_j, is only reflected in the evaluation of Eq. (3-21), which involves only simple vector products. Hence, the adjoint approach reduces the computational effort needed to evaluate $dR/d\xi_j$ (Eq. (3-21)) from solving many coupled linear equations to the evaluation of several vector products [13].

It should be noted, however, that the direct (forward) and adjoint (backward) model equations (i.e., Eqs. (3-12) and (3-22)) are far easier to solve than the original model equations (i.e., Eq. (3-7)). Both Eqs. (3-12) and (3-22) are linear while Eq. (3-7) is nonlinear.

The direct and adjoint approaches require the results of the original model equations to be available in order to set up Eqs. (3-12) and (3-22), since the A matrix and the vectors S and S^* depend on Y.

In order to perform the GAT sensitivity analysis, the model user must first generate the matrices $\partial H/\partial Y$ and $\partial H/\partial Z$ from the original nonlinear computer model. This will be presented in Chapter IV as a part of GRESS and ADGEN automated systems procedure.

As a final frontier, the adjoint sensitivity theory has several strong points in dealing with various model design sensitivity and uncertainty applications. First, for a large input data field base, the use of adjoint functions to develop a sensitivity theory allows all data field parameters to be analyzed using only a single forward and adjoint run for each system response function being studied. Second, the adjoint functions offer strong physical insight into the processes that occur in any particular design problem. Third, the sensitivity functions can be used in formulating a linear perturbation theory and a theory for uncertainty analysis. Finally, the adjoint equations for nonlinear forward problems are linear, hence, making them easier to solve.

3.4 MODEL ADJUSTMENT THEORY (MAT) METHOD

Model Adjustment Theory (MAT) [27,44] is a general method for nonlinear model fitting and adjustment of input data and parameters of a given physical system represented by a mathematical model. The method is based on the least squares principle and allows the simultaneous determination of model parameters and the adjustment of input data fields in order to obtain a set of adjusted data and parameters which are consistent with the model equations. Following the procedure suggested by Stallman [44], the derivations of the

mathematical formulas for the least squares estimates and their variances used the Lagrange multiplier method. Due to the combination of information, namely model fitting and adjustment procedures, the main advantage of the least squares method lies in the reduction of model data uncertainties.

In formulating the MAT methodology, let x_m, $m = 1, 2, ..., M$, represents the "true" data of a set of physical system quantities and let ζ_m, $m = 1, 2, ..., M$, represents the corresponding "measurements" or "calculations" of the given system quantities. The measured or calculated data values are not absolutely accurate (i.e., exact). This can be represented mathematically as

$$\zeta_m = x_m + \epsilon_m , \quad m = 1, 2, ..., M , \tag{3-23}$$

where ϵ_m is the measuring error value.

If all the measuring biases are removed from the system then the expected value (i.e., mean value) of the measuring error, ϵ_m, is zero. That is

$$E [\epsilon_m] = 0 . \tag{3-24}$$

Furthermore, since the measured data, ζ_m, is a random variable then its expected (mean) value is the true data, x_m. In a statistical language, this can be represented as

$$\begin{aligned} E [\zeta_m] &= E [x_m + \epsilon_m] = E [x_m] + E [\epsilon_m] , \\ &= E [x_m] = x_m . \end{aligned} \tag{3-25}$$

The variances (uncertainties) and correlations of the measured data values (ζ_m's) can be

expressed statistically by a variance - covariance matrix i.e., uncertainty matrix of input data parameters

$$U_x = [U_{mn}] , \tag{3-26}$$

where

$$U_{mn} = cov\,(\zeta_m , \zeta_n) = cov\,(\epsilon_m , \epsilon_n) ,$$

which can be represented in a matrix form as

$$U_{mn} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_2^2 & \sigma_{23} & \cdots & \sigma_{2n} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \sigma_x^2 & \cdots & \sigma_{xy} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \sigma_{m1} & \sigma_{m2} & \sigma_{m3} & \cdots & \sigma_{mn}^2 \end{bmatrix}_{(m \times n)}$$

where σ_x^2 = variance of K measurements of quantity x,

$$= \frac{1}{K-1} \sum_{k=1}^{K} (x_k - \bar{x})^2 ,$$

$$= \frac{1}{K-1} \sum_{k=1}^{K} x_k^2 - K\bar{x}^2 ,$$

$$= \frac{1}{K-1} \left[\sum_{k=1}^{K} x_k^2 - \frac{1}{K} \left(\sum_{k=1}^{K} x_k \right)^2 \right] ,$$

$\bar{x}$ = mean (arithmetic average) of the K measurements of quantity x,

$$= \frac{1}{K} \sum_{k=1}^{K} x_k = \frac{x_1 + x_2 + \ldots + x_k}{K} ,$$

σ_{xy} = covariance of K measurements of two measured quantities x and y,

$$= \rho_{xy}\, \sigma_x\, \sigma_y ,$$

$$= \frac{1}{K-1}\left[\sum_{k=1}^{K} x_k\, y_k - \frac{1}{K}\left(\sum_{k=1}^{K} x_k\right)\left(\sum_{k=1}^{K} y_k\right)\right],$$

ρ_{xy} = correlation coefficient of x and y ,

$$= \sigma_{xy} / \sigma_x \sigma_y ,$$

$$= \frac{\sum_{k=1}^{K} (x_k - \bar{x})(y_k - \bar{y})}{\sqrt{\left[\sum_{k=1}^{K} (x_k - \bar{x})^2\right]\left[\sum_{k=1}^{K} (y_k - \bar{y})^2\right]}} ,$$

$$= \frac{K \sum_{k=1}^{K} x_k y_k - \left(\sum_{k=1}^{K} x_k\right)\left(\sum_{k=1}^{K} y_k\right)}{\sqrt{\left[K \sum_{k=1}^{K} x_k^2 - \left(\sum_{k=1}^{K} x_k\right)^2\right]\left[K \sum_{k=1}^{K} y_k^2 - \left(\sum_{k=1}^{K} y_k\right)^2\right]}} .$$

In the case of purely independent (uncorrelated) measurements, ρ_{xy} equal to zero, while for completely correlated measurements, ρ_{xy} has the value of either +1 or -1.

The mathematical model of a given physical system which restricts the true data values, x_m's, will be given as

$$\begin{aligned} &\Phi_1 (x_1, x_2, \ldots, x_M) = 0 , \\ &\Phi_2 (x_1, x_2, \ldots, x_M) = 0 , \\ &\cdot \\ &\cdot \\ &\cdot \\ &\Phi_K (x_1, x_2, \ldots, x_M) = 0 , \end{aligned} \qquad \text{(3-27a)}$$

or in a vector notation as

$$\Phi(X) = 0 , \tag{3-27b}$$

where

0 = zero vector or matrix of appropriate size.

The mathematical model (set of equations) of Eq. (3-27) may also contain a number of parameters, $p_1 , p_2 , ..., p_N$ whose values are determined uniquely by the values of x_m's (true values) through the set of equations (model fitting)

$$\begin{aligned} &\Phi_1 (x_1, x_2, ..., x_M; p_1, p_2, ..., p_N) = 0 , \\ &\Phi_2 (x_1, x_2, ..., x_M; p_1, p_2, ..., p_N) = 0 , \\ &\cdot \\ &\cdot \\ &\cdot \\ &\Phi_K (x_1, x_2, ..., x_M; p_1, p_2, ..., p_N) = 0 , \end{aligned} \tag{3-28a}$$

or in a vector notation as

$$\Phi(X,P) = 0 . \tag{3-28b}$$

Since the measured data (ζ_m's) are not exact, then Eqs. (3-27) and (3-28) are not exactly satisfied if x_m's are replaced by ζ_m's. The resulting differences between x_m's (actual or true values) and ζ_m's (measured or calculated values) are denoted as residuals r_m's. That is

$$\Phi(\zeta) = r , \tag{3-29a}$$

or

$$\Phi(\zeta, P_0) = r \ , \tag{3-29b}$$

where

P_0 = initial estimate of the parameter set P, and

r = residuals or deviations, $(\zeta - X)$.

The main idea of the adjustment procedure is to vary or adjust the measured data set, ζ, and the parameter set, P_0, to satisfy Eqs. (3-27) and (3-28), respectively (i.e., to make r in Eq. (3-29) vanish). Mathematically, this can be represented as

$$\Phi(\zeta + \Delta\zeta) = 0 \ , \tag{3-30a}$$

or

$$\Phi(\zeta + \Delta\zeta \ ; \ P_0 + \Delta P) = 0 \ , \tag{3-30b}$$

Defining the total variation in the form of chi-square as [44]

$$\chi^2 = \Delta\zeta^{tr} \ U_x^{-1} \ \Delta\zeta \ . \tag{3-31}$$

The least squares problem can then be solved by the Lagrange multiplier method. Defining the Lagrange function as [44]

$$L \ (\Delta\zeta, \Delta P) = \chi^2 + \Phi^{tr} \ (\Delta\zeta, \Delta P) \ \lambda \ , \tag{3-32}$$

where

λ = vector of Lagrange multipliers,

$= (\lambda_1, \lambda_2, ..., \lambda_K)^{tr}$.

The direction of the adjustment theory is to minimize the total variation, χ^2, using the Lagrange function and the following conditions for minimization

$$\frac{\partial L\ (\Delta\zeta_m, \Delta P_n)}{\partial \Delta\zeta_m} = 0\ , \tag{3-33}$$

and

$$\frac{\partial L\ (\Delta\zeta_m, \Delta P_n)}{\partial \Delta P_n} = 0\ , \tag{3-34}$$

for all $m = 1, 2, ..., M$ and $n = 1, 2, ..., N$.

Eqs. (3-33) and (3-34) together with Eq. (3-30b) are then used to calculate the unknown adjustments $\Delta\zeta_m$ and ΔP_n.

Applying Eqs. (3-33) and (3-34), the following set of equations are satisfied in matrix form

$$\frac{\partial L}{\partial \Delta\zeta} = \boldsymbol{U}_x^{-1}\ \Delta\zeta + \left(\frac{\partial \Phi}{\partial \Delta\zeta}\right)^{tr} \lambda = \mathbf{0}\ , \tag{3-35}$$

and

$$\frac{\partial L}{\partial \Delta \boldsymbol{P}} = \left(\frac{\partial \Phi}{\partial \Delta \boldsymbol{P}}\right)^{tr} \lambda = \mathbf{0}\ , \tag{3-36}$$

where $(\partial\Phi/\partial\Delta\zeta)$ and $(\partial\Phi/\partial\Delta\boldsymbol{P})$ are Jacobi matrices.

Expanding Eq. (3-30b) as Taylor series representation to second order of accuracy yields

$$\Phi\ (\zeta\ +\ \Delta\zeta\ ;\ \boldsymbol{P}_0\ +\ \Delta\boldsymbol{P}) = \Phi(\zeta,\ \boldsymbol{P}_0) + \left(\frac{\partial\Phi}{\partial\Delta\zeta}\right)\Delta\zeta + \left(\frac{\partial\Phi}{\partial\Delta\boldsymbol{P}}\right)\Delta\boldsymbol{P} = \boldsymbol{0}\ , \tag{3-37}$$

and noting that the first term on the right hand side of Eq. (3-37) is equal to $\boldsymbol{r}$ (Eq. (3-29b)). Thus, Eq. (3-37) can be rewritten as

$$\boldsymbol{r} + \left(\frac{\partial\Phi}{\partial\Delta\zeta}\right)\Delta\zeta + \left(\frac{\partial\Phi}{\partial\Delta\boldsymbol{P}}\right)\Delta\boldsymbol{P} = \boldsymbol{0}\ . \tag{3-38}$$

Rewriting Eqs. (3-35) and (3-38) for the adjustments of $\Delta\zeta$ and $\Delta\boldsymbol{P}$, yields the following

$$\Delta\zeta = -\ \boldsymbol{U}_x \left(\frac{\partial\Phi}{\partial\Delta\zeta}\right)^{tr} \lambda\ , \tag{3-39}$$

and

$$\Delta\boldsymbol{P} = -\left(\frac{\partial\Phi}{\partial\Delta\boldsymbol{P}}\right)^{-1}\left[\boldsymbol{r} + \left(\frac{\partial\Phi}{\partial\Delta\zeta}\right)\Delta\zeta\right], \tag{3-40}$$

where λ is determined from Eq. (3-36).

Finally, the adjustment theory reduces the nonlinear problem to simple set of linear equations (Eqs. (3-35), (3-36), and (3-38)). That is

$$\boldsymbol{U}_x^{-1}\ \Delta\zeta + \Phi_x^{tr}\ \lambda = \boldsymbol{0}\ , \tag{3-41a}$$

$$\Phi_p^{tr}\ \lambda = \mathbf{0}\ , \tag{3-41b}$$

$$r + \Phi_x\ \Delta\zeta + \Phi_p\ \Delta P = \mathbf{0}\ , \tag{3-41c}$$

where $\Phi_x = (\partial\Phi/\partial x)$ and $\Phi_p = (\partial\Phi/\partial P)$ are Jacobi matrices. Hence, the mathematical set of Eq. (3-41) is sufficient to determine the unknown values of $\Delta\zeta$, ΔP, and λ.

The following iteration process is for determining the solution of the unknown values:

1. Evaluate the residuals, r, from Eq. (3-29) and the variance-covariance matrix, U_x, from Eq. (3-26) using initial value set of measured data, ζ, and parameters, P_0.
2. Jacobi matrices Φ_x and Φ_p are calculated at initial value set of ζ and P_0.
3. Use the mathematical set of linear equations (Eq. (3-41)) to determine the unknown values of $\Delta\zeta$, ΔP, and λ.
4. Check if the adjustment values of $\Delta\zeta$ and ΔP are consistent with model Eqs. (3-35) and (3-36).

   If yes → finish

   If no → go to step 5.
5. Replace $\zeta = \zeta + \Delta\zeta$, $P = P_0 + \Delta P$, and repeat the iteration process.

In Chapter VI, an application of the MAT methodology to multilayer feedforward networks for determining the uncertainties and covariances of residuals, weight vector parameters, input data fields, and model responses of interest will be presented.

3.5 SUMMARY

Generalized sensitivity theory plays an important role in the formation of model adjoint equations. Also in model adjustment theory, the derivative information (Jacobi matrices) plays an important role in the iteration process to determine the adjustment of model input data and parameters of a given physical system.

CHAPTER IV

DESCRIPTION OF AUTOMATED SYSTEMS PROCEDURE

4.1 INTRODUCTION

This chapter reports on the development and availability of two systems, GRESS and ADGEN, that make use of computer calculus compilers to automate the implementation of general sensitivity theory (GST) capability into existing computer models. This automation removes the traditional limitations of the GST methods (direct and adjoint formalisms). These limitations require the following: (1) model equations must be extracted from the computer coding, (2) they must be differentiated with respect to all parameters of interest, and (3) direct or adjoint sets of equations must be set up for computational solution.

4.2 GRESS METHODOLOGY [63]

The GRadient-Enhanced Software System (GRESS) methodology is an automated system developed by Oblow et al. [63] at Oak Ridge National Laboratory (ORNL) for solving the direct (forward) model sensitivity problems.

The basic principle of GRESS is to read the model source program and search for model equations. These model equations are identified, in the FORTRAN source program, by the appearance of the assignment operator "=" (equal sign) symbol. Since all FORTRAN "equations" occur in the form of Eq. (3-7) (i.e., with a single dependent variable on the left-hand side of such an expression $Y = H(Y,Z)$), GRESS can search for and analyze each equation in terms of its functional dependence on Y and Z. The basic computer calculus

operations of GRESS are then used to compute the successive elements of $\partial H/\partial Y$ and $\partial H/\partial Z$ for each expression encountered (i.e., each component of $H(Y,Z)$). The resulting values of these derivatives are stored in vector form for each component of Y. The differentiation is carried out analytically using the chain-rule derivative propagation method embodied in calculus software for all permissible FORTRAN functions and operators and the results are computed and stored numerically using the current values of the independent and dependent variables. From the GRESS viewpoint, there are potentially three classes of variables: (1) independents (i.e., data field parameters of interest, $\xi = (\xi_1, \xi_2, ..., \xi_J)$), (2) dependents (i.e., Y and Z), and (3) constants.

The independent variables (parameters, ξ_j) for which GRESS generates partial derivatives in the enhanced FORTRAN source code will be selected by the user. The only theoretical restriction to this choice concerns the sequence of computations in the source code which must be continuous in differentiable form for model derivatives generated to possess practical meanings. GRESS takes advantage of the fact that in solving Eq. (3-11), the A matrix, i.e., $(I - \partial H/\partial Y)$, is lower triangular and the Y vector can be computed by forward substitution. The components of Y are solved successively as each equation is differentiated and that the matrix $(I - \partial H/\partial Y)$ doesn't have to be stored.

A simple example of this procedure is illustrated in Figs. 1 and 2. The analytical goal of this example aims to calculate $dR/d\xi_j$, where x and y are the main dependent variables of interest and ξ_j is the independent variable (i.e., data field parameter of interest) in the problem. The three given equations (i.e., $x = 2\xi_j$, $y = xe^{2x}$, *and* $R = xy^2$), in the GRESS procedure, are processed to produce model derivatives with respect to a single parameter of interest (ξ_j). An additional analytical equation is needed to produce the derivative for each dependent variable (i.e., there are three such new equations, one for x, y, and R). These

SOURCE PROGRAM	ANALYTICAL EQUATIONS
READ(5) ZJ	ξ_j = value set externally
X = 2.0 * ZJ	$x = 2\xi_j$
Y = X*EXP(2.0*X)	$y = xe^{2x}$
R = X*Y**2.0	$R = xy^2$

Figure 1. Model source program and associated equations. (Ref. 63)

GRADIENT ENHANCED SOURCE PROGRAM	ANALYTICAL EQUATIONS
READ (5) ZJ	ξ_j = value set externally
DZDZJ = 1.0	$d\xi_j/d\xi_j = 1$
X = 2.0*ZJ	$x = 2\xi_j$
DXDZJ = 2.0*DZDZJ	$dx/d\xi_j = 2(d\xi_j/d\xi_j)$
Y = X*EXP(2.0*X)	$y = xe^{2x}$
DYDZJ = (2.0*X+1.0)* EXP(2.0*X)*DXDZJ	$dy/d\xi_j = (2x+1)e^{2x}(dx/d\xi_j)$
R = X*Y**2.0	$R = xy^2$
DRDZJ = 2.0*X*Y*DYDZJ+ DXDZJ*Y**2.0	$dR/d\xi_j = 2xy\ (dy/d\xi_j) + (dx/d\xi_j)y^2$

Figure 2. Gradient enhanced model and associated equations. (Ref. 63)

equations are produced by the GRESS software analytically (using the chain-rule derivative propagation method) with numerical values being computed and stored at execution time. It is instructive to note that chain rule propagation, in the GRESS language, is the key to the success of the method, since current values of all the dependent-variable derivative vectors (i.e., $dx/d\xi_j$ and $dy/d\xi_j$) are needed to propagate all derivative values (i.e., $dR/d\xi_j$) explicitly through the complete system of model equations (see Fig. 2).

The application of GRESS to an existing FORTRAN model is illustrated in Fig. 3. It consists of an automated precompilation step in which the automated code translation necessary to compute derivatives is performed using computer calculus. This step consists primarily of a rearrangement of the program data structure and a substitution of calls to GRESS interpretive software in place of all arithmetic lines of coding. All arithmetic operations of this original model are precompiled into a pseudomachine code (the GRESS P-code) for use during program execution. The two output files of this step are the enhanced model and the binary P-code file. These two files and a set of GRESS software subroutines supporting the enhanced model are compiled and run as a normal FORTRAN program to produce both the reference model results and gradient (derivative) information. The gradients and reference results are used to calculate model sensitivities and uncertainties for various engineering and physics applications.

In summary, GRESS accomplishes its task appropriately by reading the FORTRAN source code text, redefining variable storage locations, searching for arithmetic statements by enhancing these for gradient calculation, and then generating a new FORTRAN source program that can produce derivatives for any real variables. These derivatives are available for model sensitivity and uncertainty studies.

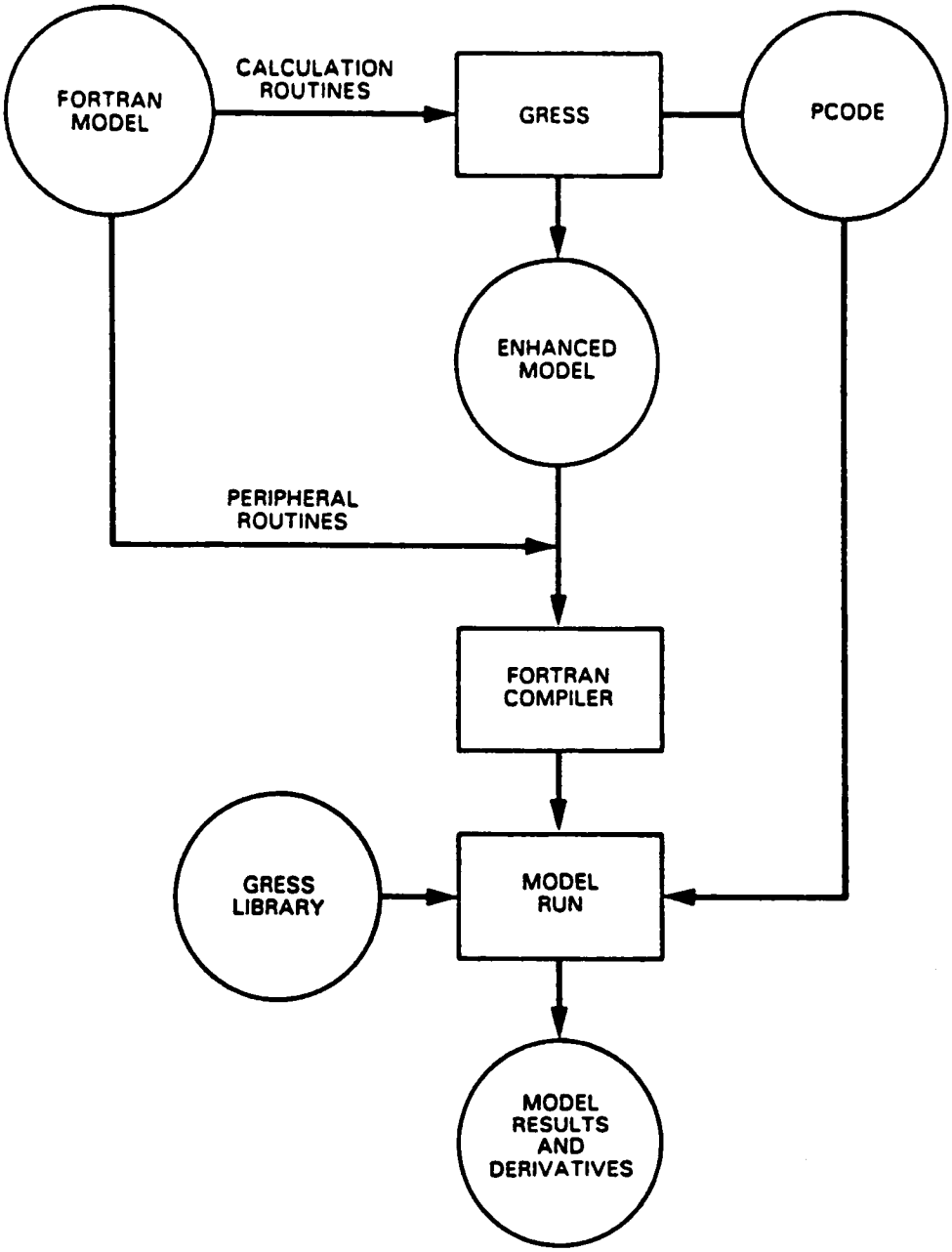


Figure 3. Flowchart of procedures for the use of GRESS system. (Ref. 13)

4.3 ADGEN METHODOLOGY [64]

The ADjoint GENerator (ADGEN) methodology is an automated system developed by Worley et al. [64] at ORNL for solving the adjoint (backward) model sensitivity problems.

The adjoint problem is defined by Eqs. (3-18) through (3-22). The calculation of the adjoint solution vector Y^* from Eq. (3-22) is not a function of the selection of input data field parameter ξ_j and hence need only be performed once to determine the derivatives of model response of interest with respect to any input data field parameter of interest. The matrix $\partial H/\partial Z$ must also be determined since it is independent of the parameter of interest (ξ_j). Nevertheless, the only parameter dependent operation required to calculate the derivative $dR/d\xi_j$ is the simple matrix multiplication operation, $(Y^*)^{tr}\ (\partial H/\partial Z)\ (dZ/d\xi_j)$, in which the vector $dZ/d\xi_j$ is a function of ξ_j. Here, ADGEN uses the GRESS precompiler to calculate all required derivatives of the $\partial H/\partial Y$ and $\partial H/\partial Z$ matrices.

The ADGEN system (as well as GRESS) only recognizes real-variable store operations as valid equations (i.e., the left-hand side variable in a FORTRAN equation must be real), since continuous derivatives are to be calculated. In spite of this similarity, ADGEN system (unlike GRESS) requires that the matrix $(I - \partial H/\partial Y)^{tr}$ (i.e., all derivatives that constitute such a matrix) be stored and includes a post processor solver routine to calculate the adjoint solution. The adjoint equation, Eq. (3-22) is subsequently solved by the ADGEN solution module using the adjoint matrix (i.e., $(I - \partial H/\partial Y)^{tr}$) and $(dG/dY)^{tr}$ as input.

The application of ADGEN to an existing FORTRAN model is illustrated in Fig. 4. It consists of a step in which the reference model is read as input to EXAP (Extended Arithmetic Processor). This step consists primarily of a rearrangement of the program data structure and a substitution of calls to the interpretive software. The two output files of this

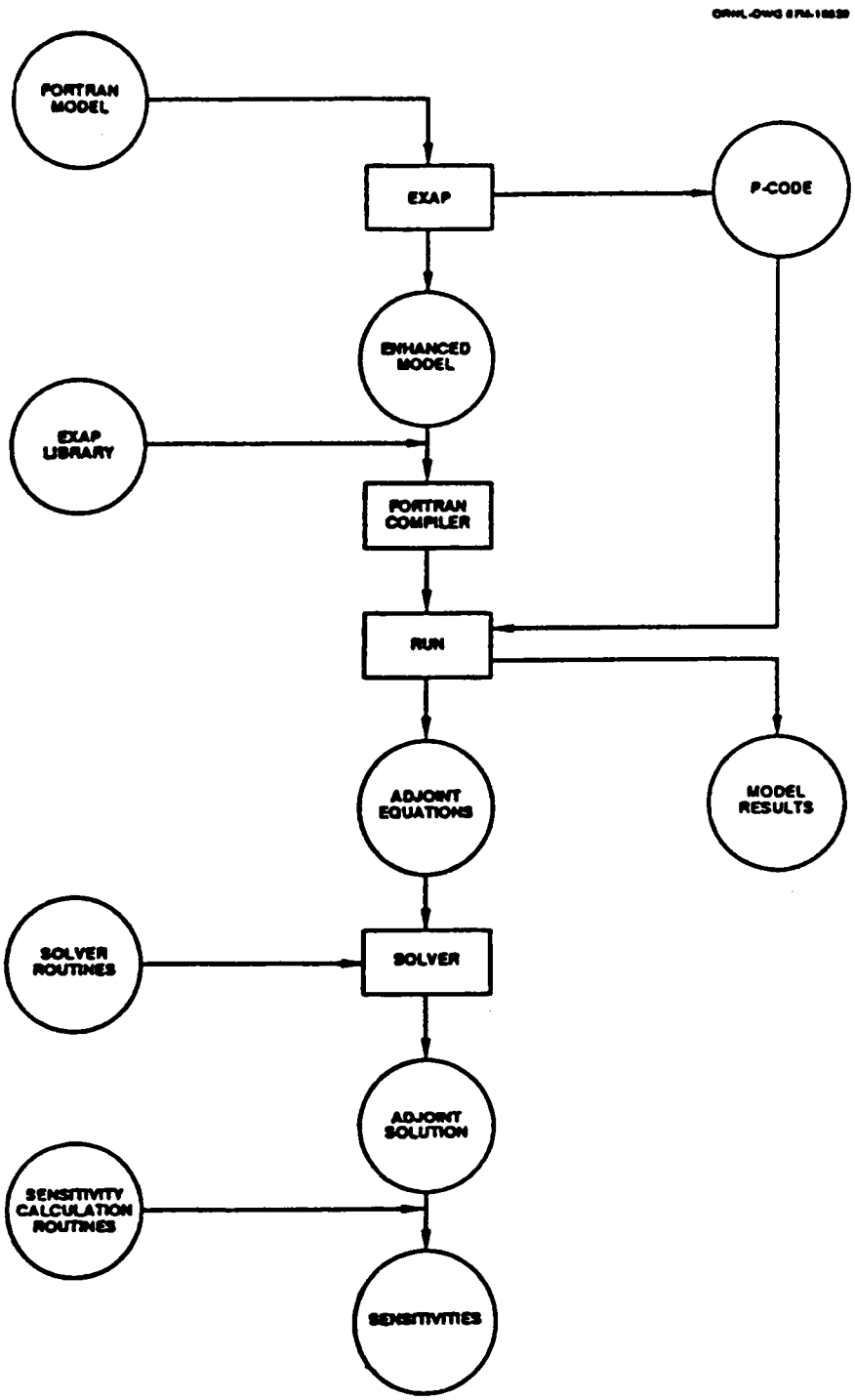


Figure 4. Flowchart of procedures of the use of ADGEN system. (Ref. 42)

step are thus the enhanced model and the binary P-code file. These two files and a set of ADGEN software subroutines supporting the enhanced model are compiled and run as a normal FORTRAN program to produce both the reference model results and the adjoint matrix.

A list of GRESS and ADGEN automated systems terminology is listed in Table 1.

4.4 SUMMARY

The availability of both GRESS and ADGEN systems allow the analyst to compute model derivatives and sensitivities for a wide range of engineering and physics applications. GRESS is more suited for small data bases and for restricted direct access storage availability. ADGEN, on the other hand, is more suited for calculating model sensitivities to a large number of input data field parameters. In both systems, the first derivatives and model sensitivities are analytically exact and can be printed out and/or saved by simple user instructions in addition to normal model results.

Table 1.

GRESS and ADGEN Terminology (Ref. 64)

Name	Description
ADGEN	ADjoint GENerator. That part of GRESS that automates calculation of derivatives and sensitivities using the adjoint theory.
ADJOINT	Option flag in the EXAP precompiler for producing the adjoint matrix and calculating derivatives and sensitivities using the adjoint sensitivity method.
Adjoint Matrix	A direct access output data set containing partial derivatives identified by row and column number information. The data set is created by restructuring the forward matrix for reading sequentially from the last record to the first record, and by replacing the virtual addresses with column and row number information.
CHAIN	Option flag in the EXAP precompiler for calculating derivatives and sensitivities by the direct sensitivity method using the chain rule.
Dependent Variable	Any program calculated real number variable whose value is at least partially determined by one or more independent variables.
Enhanced Model	The reference model (source code) enhanced for gradient calculation.
EXAP	Extended Arithmetic Processor. The FORTRAN precompiler in GRESS that enhances the FORTRAN source computer code with (1) the capability to calculate derivatives and sensitivities by the direct sensitivity method of chain rule application or (2) the capability to produce the adjoint matrix for subsequent calculation of derivatives and sensitivities by the adjoint method.
Forward Matrix	An output data set from an adjoint application containing partial derivatives identified by virtual addresses.
Forward solution	The results that are obtained by running the computer code prior to precompilation.
GRESS	GRadient Enhanced Software System. The overall software system consisting of the EXAP precompiler, run-time library routines, and the BREDUCE and BSOLVE routines required by ADGEN.

Table 1. (Continued)

Name	Description
Parameter	Any real number variable, input or calculated, that is explicitly declared to be a parameter.
precompilation	A line-by-line translation of a FORTRAN source program into an enhanced FORTRAN program prepared for either CHAIN or ADJOINT calculation.
Psuedo-code (P-Code)	A sequence of integers that specifies the mathematical relationship between the terms on the LHS and the terms on the RHS for each equation.
Reference Model	The user's source code prior to enhancement with EXAP.
Response	Any program calculated real number selected by the user for gradient and sensitivity calculations.

CHAPTER V

MULTILAYER FEEDFORWARD NEURAL NETWORKS

5.1 INTRODUCTION

This chapter introduces the fundamental principles and the mathematical concepts used to describe and analyze the multilayer feedforward networks - a part of the ANS systems. Broadly speaking, the ANS systems consist of three elements: (1) an organized topology (geometry) of interconnected processing elements, (2) a method of encoding information, and (3) a method of recalling information. These elemental characteristics play an important role in the foundations, applications, and implementations of ANS systems.

5.2 MULTILAYER FEEDFORWARD NETWORKS FUNDAMENTALS [46]

A neural network is an information (or data) processing system that is intensely parallel, non-algorithmic, and non-digital system. It is not a computer model nor is it programmed like it. It consists of a number of very simple and highly interconnected processing elements in an architecture inspired by the structure of cerebral cortex of the brain [46]. These processing elements are connected by a large number of weighted links, over which given signals can pass. Each processing element receives many signals over its incoming connections. The processing element produces only a single output (outgoing) signal. That output signal transmits over the processing element outgoing connection which then splits into a very large number of smaller connections, each of which terminates at a different position. Each branch of the single outgoing connection transmits the same signal. Most of these

outgoing branches will terminate at the incoming connection of some other processing elements in the network; others may terminate outside the network and hence, generate control or response patterns. A typical multilayer feedforward neural network architecture system, which illustrates the interfield connections among the processing elements, is shown in Fig. 5.

The processing element receives input stimuli along its input (incoming) connections and translates them (i.e., input stimuli) into an output response through transmitting along the processing element's output (outgoing) connection. The place where the input connection meets the processing element is called the "synapse" [46]. The translation of input stimulus pattern to output response signal is called the "transfer function (TF)" of the processing element. The TF consists of a three-step process. First, the processing element computes the net weighted input stimulus pattern which received along its input connections. This process consists of computing the activation value $I_i[L]$ as follows (see Fig. 6 which displays a topology of the generic processing element $I_i[L]$)

$$I_i[L] = \sum_{h=1}^{n} a_h[L-1]\, w_{hi}[L] , \qquad \begin{matrix} h=1, 2, ..., n , \\ i=1, 2, ..., p , \end{matrix} \tag{5-1}$$

where

$I_i[L]$ = net weighted input received by the *ith* processing element in layer [L] from a total of n processing elements or inputs in the network,

$a_h[L-1]$ = incoming signal from the *hth* processing element in layer [L-1] of input vector $A = (a_1, a_2, ..., a_h, ..., a_n)$,

$w_{hi}[L]$ = weight of the connection joining the *hth* processing element in layer [L-1] to the *ith* processing element in layer [L] which represents a weight vector $W = (w_{1i}, ..., w_{hi}, ..., w_{ni})$, and

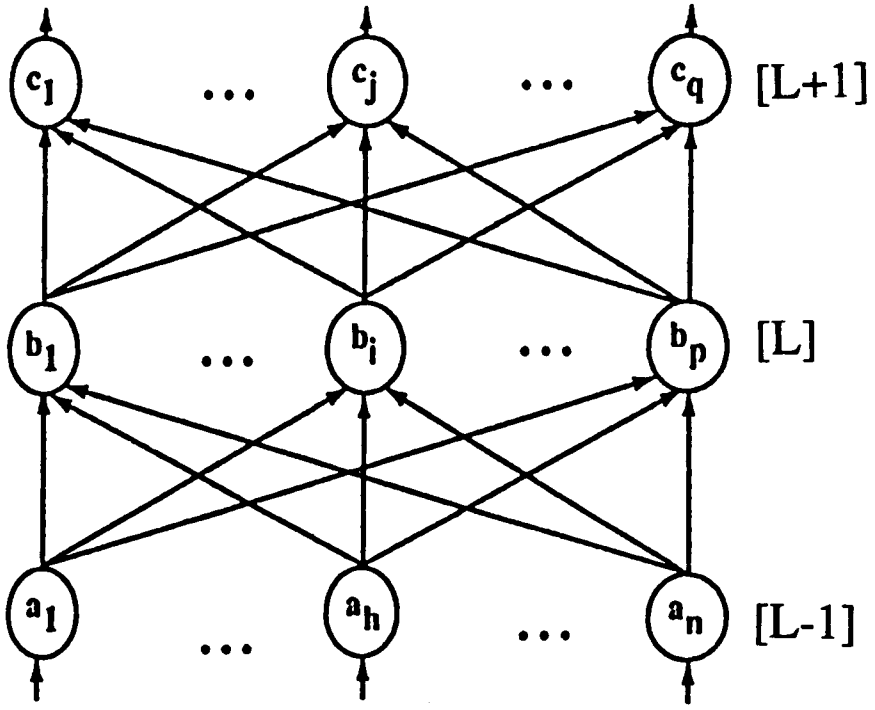


Figure 5. A typical three-layer feedforward neural network architecture. (Ref. 48)

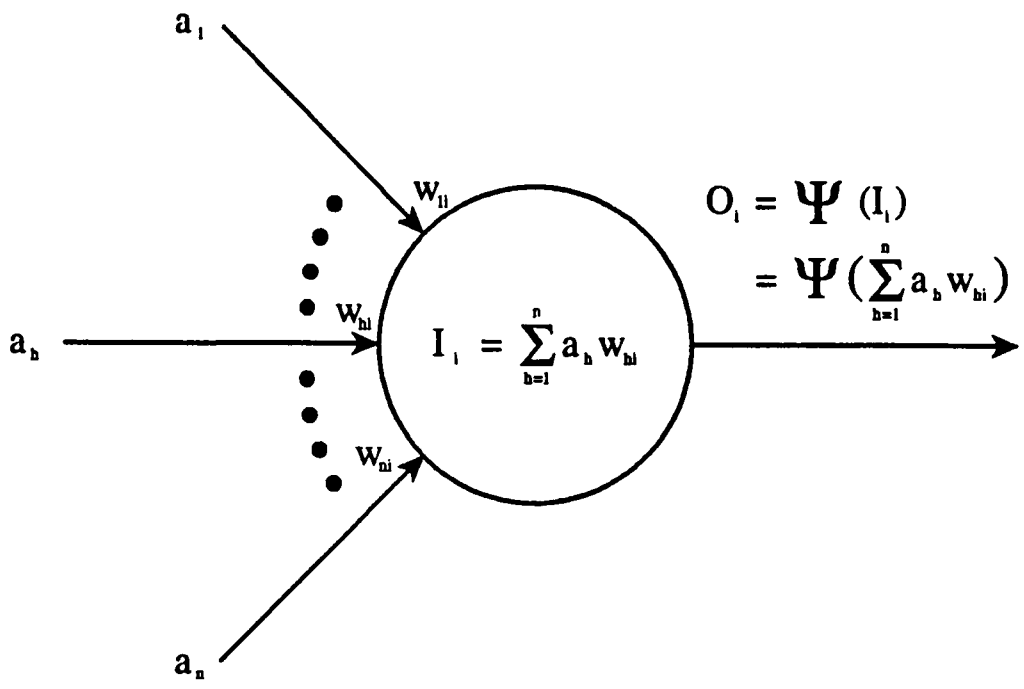


Figure 6. Topology of the generic processing element I_i [L]. (Ref. 48)

[L-1], [L], and [L+1] represent the input, middle (hidden), and output layers, respectively.

From Eq. (5-1), it is noticed that the overall net stimulus pattern is considered (i.e., not the value of any particular stimulus pattern). Second, is to convert the net input to an activation level function for the processing element. In most networks the activation function is a "sigmoidal" function, shown in Fig. 7, which is expressed by an S-shaped curve (other activation functions to be considered are hyperbolic tangent or sine). The general form of sigmoidal function is

$$\Psi(\eta) = (1+e^{-\alpha\eta})^{-1} , \tag{5-2}$$

where $\alpha = 1$ for the common used sigmoidal function. This function has the property that its derivative is easy to compute, being merely

$$\begin{aligned} \frac{d\Psi(\eta)}{d\eta} = \Psi'(\eta) &= e^{-\eta} (1+e^{-\eta})^{-2} , \\ &= [(1-e^{-\eta})-1] (1+e^{-\eta})^{-2} , \\ &= (1+e^{-\eta})^{-1} - (1+e^{-\eta})^{-2} , \\ &= (1+e^{-\eta})^{-1} [1 - (1+e^{-\eta})^{-1}] , \\ &= \Psi(\eta) [1-\Psi(\eta)] . \end{aligned} \tag{5-3}$$

It is important to know that the activation function, $\Psi(\eta)$, is monotonically increasing (i.e., everywhere differentiable with a positive slope), and bounded with both lower and upper limits. Hence in the case of $\Psi(\eta)$, the minimum value of $\Psi(\eta)$ as $\eta \rightarrow -\infty$ is 0 (zero), and the maximum value of $\Psi(\eta)$ as $\eta \rightarrow +\infty$ is 1. The third step is to convert the activation level function of the processing element to an output signal. This can be done by setting the output signal as follows:

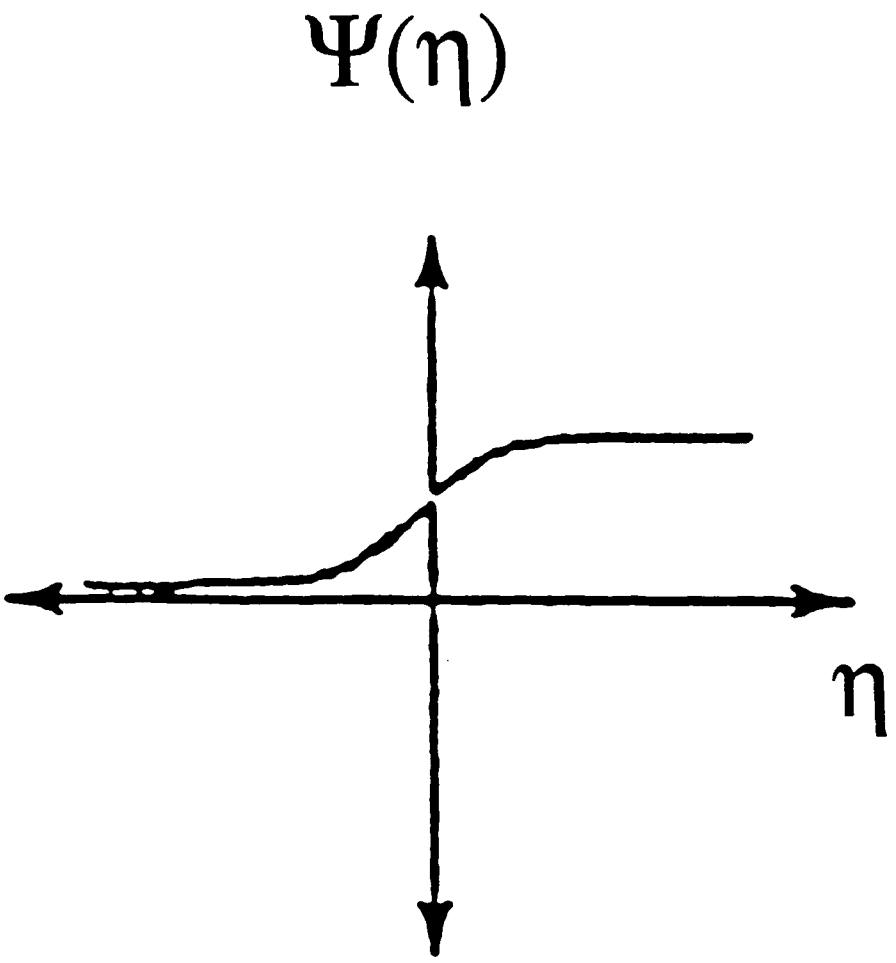


Figure 7. Sigmoidal function. (Ref. 48)

$$O_i[L] = \Psi(I_i[L]) = \Psi\left(\sum_{h=1}^{n} a_h w_{hi}\right),$$
$$= \left(1 + \exp\left[-\sum_{h=1}^{n} a_h\, w_{hi}\right]\right)^{-1}. \qquad (5\text{-}4)$$

5.3 A NEURAL NETWORK IN OPERATION [46]

When a neural network is presented with an input stimulus pattern, each of the processing elements on the input layer of the network receives a small piece of the input pattern. The processing elements collectively distribute their small part of the input pattern, more or less unchanged, to the middle-layer processing elements. Each of the middle-layer processing elements receives the entire input pattern, but the incoming pattern (to middle-layer processing elements) is modified by its passage through the weighted connections leading to the middle layer. Since the weights on the connections are different for each middle-layer processing element, then each of these processing elements (i.e., middle-layer processing elements) sees a somewhat different version of the input pattern than its neighbors. As a result, some combination of middle-layer processing elements will become active to certain varying degrees, depending on the exact collection (or summation) of weights on their input connections. This results in a variety of output response signals transmitting from the middle-layer processing elements. This variety of output response ranging from not output at all to, possibly, an extremely strong output.

Normally, all the middle-layer processing elements transmit their output signals to all of the processing elements in the output layer. Thus, each of the output layer processing elements receives the complete pattern of output activity from the middle-layer processing

elements. This activity pattern, as was the case with middle-layer processing elements, is modified by its passage through the weighted connections between the middle and output-layer processing elements. Thus, the processing elements in the output layer, each sees somewhat different versions of the middle-layer activity. The result, however, is some combination of output-layer processing elements become active to greater or lesser degrees, causing them to output a signal pattern. This pattern of output-layer processing element responses is the network's overall response to the original input stimulus pattern.

5.4 LEARNING AND TRAINING [46]

Neural networks are not programmed to solve a given problem rather they learn how to solve it. Learning is achieved by changing and/or modifying the weights on the interconnections between layers in the network. Each processing element's output is determined (with the assumption that the transfer function is fixed) by only two things: (1) the incoming signal and (2) the weights on the input connections to the processing element. Hence, if the processing element is to learn to respond correctly to the incoming signal pattern, then the only possible element that can be used to improve the processing element's performance is the weight on the connection. Therefore, learning in neural networks consists of making "systematic" changes to these weights in order to improve the network's response performance to acceptable levels.

Training, on the other hand, is the procedure by which the network is able to learn, and learning is the end result of that procedure. Training is done by various examples and it can be performed in three different kinds of methods. First, is the supervised training (most common method). In this technique, the network is provided with an input stimulus

pattern along with the corresponding desired output pattern. The learning law for such networks is to compute an error (i.e., the difference between the network's "actual" and desired outputs). This error is then used to modify the weights on the interconnections among the processing elements and thus train the network. Second, is the graded (or reinforcement) training. This type of training is similar to supervised training except that the desired output pattern is not provided, only a "grade" on how well the network is doing. There are a number of schemes for this type of training and they vary from giving a brief message e.g., "you succeeded" or "you failed" to more informative e.g., "too high" or "too low" performance feedback. Third, is the unsupervised (or self-organization) training. In this procedure, the network is only provided with a series of input stimulus patterns with no information or feedback presented to regulate its performance levels. This type of network training is often used for categorization or statistical modeling applications, since the network's specific responses of interest can not be predetermined by the designer.

5.5 SUMMARY

An introduction to the fundamental principles and the mathematical concepts which are used to describe and analyze the theory of multilayer feedforward networks was presented and outlined.

CHAPTER VI

APPLICATION OF DATA ADJUSTMENT FOR UNCERTAINTY ANALYSIS OF NEURAL NETWORKS

6.1 INTRODUCTION

An application of model adjustment theory (MAT) method, presented in Chapter III, for uncertainty analysis of multilayer feedforward networks will be developed in this chapter. The determination of uncertainties and covariances along with a description of the adjustment algorithm for improving the learning rate of such networks will be discussed and outlined.

6.2 DATA ADJUSTMENT FOR MULTILAYER FEEDFORWARD NETWORKS

In this section, the formalism of data adjustment for obtaining the uncertainty matrix of the adjusted parameters (input and/or weight vectors) in multilayer feedforward networks, will be presented.

A multilayer feedforward topology, shown in Fig. 8, is a three-layer ANS with feedforward connections from the [L-1] processing elements to the [L] processing elements and feedforward connections from the [L] processing elements to the [L+1] processing elements. Following Eqs. (5-1) - (5-4), the activation value of the *ith* [L] processing element is given by

$$b_i = \Psi\left(\sum_{h=1}^{n} a_h v_{hi} + \theta_i\right), \quad \begin{matrix} h=1,2,\ldots, n, \\ i=1,2,\ldots, p, \end{matrix} \tag{6-1}$$

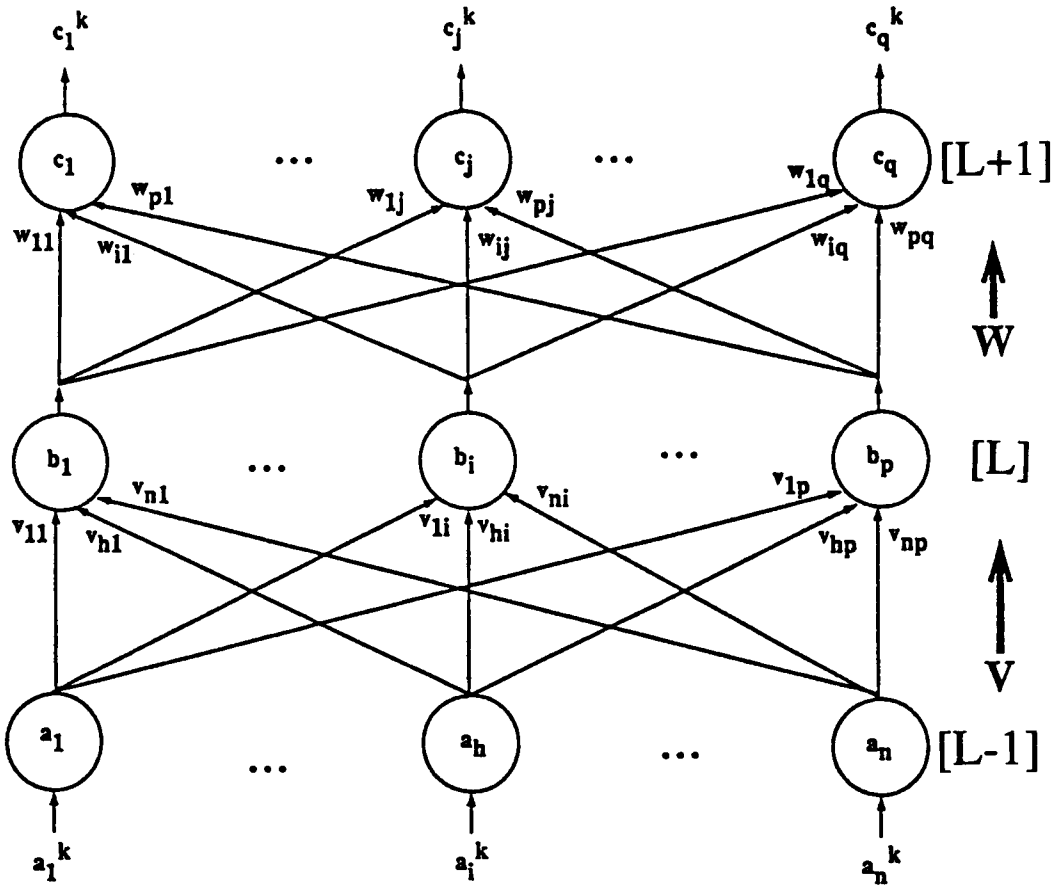


Figure 8. Topology of a multilayer feedforward ANS.

where

a_h = incoming signal from the *hth* processing element of an input vector $A = (a_1, a_2, ..., a_h, ..., a_n)$ in layer [L-1],

v_{hi} = weight on connection joining the *hth* processing element in layer [L-1] to the *ith* processing element in layer [L] which is represented by a weight vector $V = (v_{1i}, v_{2i}, ..., v_{hi}, ..., v_{ni})$,

θ_i = *ith* processing element's threshold value in layer [L], and

$\Psi(\eta)$ = sigmoidal threshold function = $(1+e^{-\eta})^{-1}$.

On the other hand, the activation value of the *jth* [L+1] processing element is given by

$$c_j = \Psi\left(\sum_{j=1}^{p} b_i w_{ij} + \Gamma_j\right), \quad \begin{matrix} i=1,2,..., p, \\ j=1,2,..., q, \end{matrix} \tag{6-2}$$

where

b_i = activation value of the *ith* [L] processing element,

w_{ij} = weight on connection joining the *ith* processing element in layer [L] to the *jth* processing element in layer [L+1] which is represented by a weight vector $W = (w_{1j}, w_{2j}, ..., w_{ij}, ..., w_{pj})$, and

Γ_j = *jth* processing element's threshold value in layer [L+1].

Eq. (6-2) can be rewritten exclusively in an expanded form as

$$c_j = \left(1 + \exp\left[- \left(\sum_{i=1}^{p} b_i w_{ij} + \Gamma_j \right) \right]\right)^{-1}, \tag{6-3a}$$

or

$$c_j = \left(1+\exp\left[-\left(\sum_{i=1}^{p} \left[1+\exp\left(-\left[\sum_{h=1}^{n} a_h \, v_{hi} + \theta_i \right] \right) \right]^{-1} w_{ij} + \Gamma_j \right) \right] \right)^{-1}. \qquad (6\text{-}3b)$$

It is noticed from Eq. (6-3) that the input parameters, a_h's, and the weight vector parameters (including the threshold values), v_{hi}'s, w_{ij}'s, θ_i's, and Γ_j's, are all functions of the output activation value, c_j. Hence, relying on the calculational procedure suggested by Stallman [44] which will be found in Appendix A, the adjustment of weight vector parameters and input parameters are given as

$$\Delta Q = -\left[\left(\frac{\partial C}{\partial Q} \right)^{tr} U_r^{-1} \left(\frac{\partial C}{\partial Q} \right) \right]^{-1} \left[\left(\frac{\partial C}{\partial Q} \right)^{tr} U_r^{-1} \, r \right], \qquad (6\text{-}4)$$

and

$$\Delta A = -\left[U_x \left(\frac{\partial C}{\partial A} \right)^{tr} U_r^{-1} \right] \left[r + \left(\frac{\partial C}{\partial Q} \right) \Delta Q \right], \qquad (6\text{-}5)$$

where

Q = total weight vector in neural network,

= $(V, W, \theta, \Gamma)^{tr}$,

U_r = uncertainty matrix of residuals ,

$$= \left[\left(\frac{\partial C}{\partial A} \right) U_x \left(\frac{\partial C}{\partial A} \right)^{tr} \right], \qquad (6\text{-}6)$$

and

U_x = uncertainty matrix of input parameters, (see Eq. (3-26)).

The uncertainty matrix of the adjusted responses, i.e., the calculations of the responses with the adjusted weight vector parameters and/or input parameters, will be given as

$$U_R = U_x - \left(U_x \left(\frac{\partial C}{\partial A}\right)^{tr} \right) U_r^{-1} \left(U_x \left(\frac{\partial C}{\partial A}\right)^{tr} \right)^{tr} + \left(U_x \left(\frac{\partial C}{\partial A}\right)^{tr} \right) U_r^{-1} \left(\frac{\partial C}{\partial Q}\right) U_w^{-1} \left(\frac{\partial C}{\partial Q}\right)^{tr} U_r^{-1} \times \left(U_x \left(\frac{\partial C}{\partial A}\right)^{tr} \right)^{tr}, \tag{6-7}$$

where

U_w = uncertainty matrix of weight vector parameters,

$$= \left[\left(\frac{\partial C}{\partial Q}\right)^{tr} U_r^{-1} \left(\frac{\partial C}{\partial Q}\right) \right]. \tag{6-8}$$

The adjustment procedure is suggested to improve (reduce) the uncertainties of calculated responses of interest.

6.3 DESCRIPTION OF THE ADJUSTMENT ALGORITHM

The adjustment algorithm is the encoding algorithm that adjusts the weight vector and/or input data parameters based on the model adjustment method and uses the three-layer feedforward topology shown in Fig. 8. The algorithm is outlined as follows:

1. Assign random values in the range of [-0.3, +0.3] to all v_{hi}, θ_i, w_{ij}, and Γ_j.

2. For each pattern pair (A_k, C_k), $k=1,2,...,NT$ (number of training cycles), do the following:

   a. Calculate the activation value function of the *ith* [L] processing element using the equation

$$b_i = \left[1 + \exp\left(- \left[\sum_{h=1}^{NI} a_h v_{hi} + \theta_i \right] \right) \right]^{-1}$$

   for each $h = 1,2,..., NI$ (number of input signals) and $i = 1,2,..., NH$ (number of hidden or middle layers).

   b. Calculate the activation value function of the *jth* [L+1] processing element using the equation

$$c_j = \left[1 + \exp\left(- \left[\sum_{i=1}^{NH} b_i w_{ij} + \Gamma_j \right] \right) \right]^{-1}$$

   for each $i = 1,2,..., NH$ and $j = 1,2,..., NO$ (number of output layers).

   c. Compute the difference between desired output and calculated output (i.e., residual or deviation) using the following relation

$$r_j = c_j^{desired} - c_j^{calculated}$$

   for each $j = 1,2,..., NO$.

   d. Adjust all weights and threshold values using the equation

$$\Delta Q_m = - \left[\left(\frac{\partial c_j}{\partial Q_m} \right)^{tr} U_{r_{ij}}^{-1} \left(\frac{\partial c_j}{\partial Q_m} \right) \right]^{-1} \left[\left(\frac{\partial c_j}{\partial Q_m} \right)^{tr} U_{r_{ij}}^{-1} r_j \right]$$

   for each $m = 1,2,..., NW$ (number of weight vector parameters) and

$j = 1,2,..., NO.$

e. Adjust the input data using the equation (optional step)

$$\Delta a_h = -\left[U_{x_{hh}} \left(\frac{\partial c_j}{\partial a_h} \right)^{tr} U_{r_{jj}}^{-1} \right] \left[r_j + \left(\frac{\partial c_j}{\partial Q_m} \right) \Delta Q_m \right]$$

for each $h = 1,2,..., NI$, $j = 1,2,..., NO$, and $m = 1,2,..., NW$.

3. Repeat step (2) until the error correction value r_j, for each $j = 1,2,..., NO$, and each $k = 1,2,..., NT$, is either sufficiently low or zero.

The flowchart of this adjustment procedure is shown in Fig. 9.

6.4 SUMMARY

An application of model adjustment method for uncertainty analysis in neural networks has been presented along with the adjustment algorithm. The weight adjustment (learning algorithm) procedure is derived here by employing the model adjustment method which reduces the uncertainties of calculated responses of interest. This method requires the availability of sensitivity functions, $\partial R/\partial \xi_j, j = 1, 2, ..., J$, and the uncertainties of model input data field parameters, U_x, prior to evaluating a new set of weight vector parameters. The availability of sensitivity functions prior to adjusting weight vector parameters requires, in its turn, the availability of the "reset" sensitivity option in GRESS system. The current version of GRESS system doesn't have the reset capability. In other schemes such as the Backpropagation Neural Networks (BNNs) [48], the weight adjustment procedure is derived by computing the change in the error surface function (cost function) with respect to the change in each weight. This learning algorithm moves the weight vector in a direction where the error surface function is decreasing rapidly.

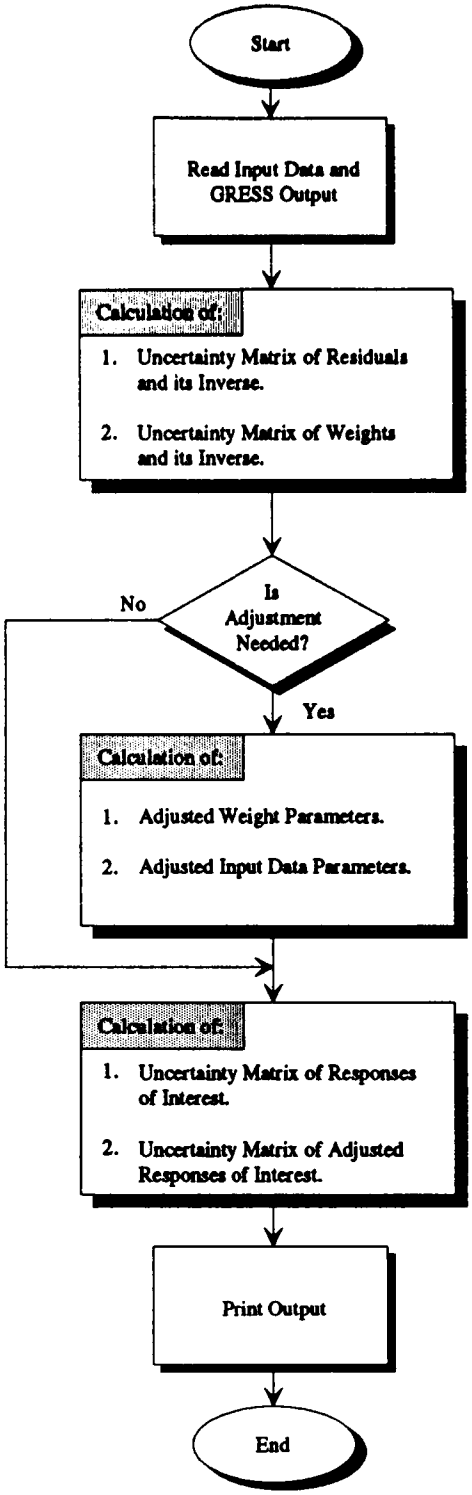


Figure 9. Flowchart of the adjustment algorithm.

CHAPTER VII

DISCUSSION OF SENSITIVITY, UNCERTAINTY, AND DATA ADJUSTMENT RESULTS

7.1 INTRODUCTION

This chapter discusses the results of sensitivity, uncertainty, and data adjustment analyses of multilayer feedforward networks for various reactor physics and mathematical model applications. These applications include reactor noise analysis (RNA) model, nuclear in-core fuel management, and second order time domain model of blood flow for Positron Emission Tomography (PET) studies. The role of uncertainties in computer codes for this study is based on the adjoint sensitivity analysis. The adjoint approach is more suited for calculating sensitivities to a large number of model data parameters.

7.2 MODEL SENSITIVITY ANALYSIS

In this section the results from a preliminary sensitivity analysis of multilayer feedforward network computer model (i.e., MFN.FOR) will be presented. The sample problems description and the steps in applying the GRESS/ADGEN option for adjoint sensitivity analysis are reviewed. A report of the gradients (derivatives) and model sensitivities for one response of interest (i.e., output activation value) with respect to all input data field parameters and weight vector parameters are presented and verified by the GRESS/CHAIN option. Finally, the results of the most sensitive input data field and weight vector parameters are presented.

7.2.1 Sample Problems Description

MFN.FOR is a computer model that is used to simulate a three-layer feedforward topology. This computer model uses the data adjustment encoding algorithm described in Chapter VI.

Three model data applications will be considered for the sensitivity analysis study. The reactor noise analysis (RNA) model [65] uses the Oak Ridge National Laboratory (ORNL) subcriticality measurement data of the Cf-252 element. For the nuclear in-core fuel management application, the training patterns are taken from the input-output of Reload-Power-Mapping (RPM) Code [66]. The input data field consists of 31 values of reactivity, k_∞. Each value is associated with one reactor core assembly. The output space consists of relative powers of reactor core assemblies at the beginning of cycle (BOC). The function of neural networks is to model the relationship between the reactivity, k_∞, and the relative power of core assembly at BOC. Finally, the Positron Emission Tomography (PET) model [67] is a newly developed imaging technique for measuring the concentration of positron emitting radioisotopes within an object. Each positron (antielectron) combines with an electron and produces a pair of 511 keV gamma ray (γ) annihilation photons in opposite directions. The gamma ray is penetrating the object and can then be measured externally. The isotope location can be presented as a gray scale image of a cross-sectional of the object with the intensity of each picture element (or pixel) proportional to the isotope concentration of that position in the object. The quantity that is measured in PET is tissue concentration of positron emitter.

The complete input data set, weight vector parameters, and computer model output produced are provided in Appendix B.

7.2.2 Processing MFN.FOR with GRESS/ADGEN

There are four steps in applying ADGEN with matrix reduction algorithm [61] to an exiting FORTRAN-77 program. These are:

1. Precompilation with EXAP,
2. Creation of the adjoint matrix with Forward Reduction technique,
3. Application of Back Reduction to reduce storage requirements, and
4. Solving the adjoint matrix for selected results of interest,

Precompilation with EXAP did not require any changes to the MFN.FOR source code for the selected sample problems. To perform the ADGEN application, three subroutine calls were added to the enhanced MFN.FOR source code (i.e., MFN_EH.FOR). The first executed in the enhanced version code is a call to subroutine SETRXX with the character string "ADJOINT" as the argument. This "switches" the run-time library into ADGEN mode. To identify potential responses of interest for gradient and model sensitivity calculations, a call to subroutine POTRXX is required. For this application the calculated values of SIGC (i.e., c_j, the activation value of the *jth* [L+1] processing element) were selected. Finally, a call to subroutine CLEARXX is inserted as the last executed line in the code to complete the creation of the adjoint matrix, the parameter dictionary, and the list of potential responses.

When the enhanced code is executed, five data sets are crated as shown in Table 2. These represent the parameter dictionary, the response dictionary, and the adjoint matrix (with three data sets). Once the adjoint matrix exists the Back Reduction algorithm is applied by executing a program named BREDUCE. The output from BREDUCE is simply a reduced form of the adjoint matrix as represented in Table 2.

Table 2.

Data Sets Created During an ADGEN Application

Name	Description
Parameter Dictionary	List of parameter names
Response Dictionary	List of response names
Adjoint Matrix:	(3 data sets)
- Number of Pairs - Column Numbers - Derivative Values	Non-zero derivative count for a row for right-hand-side real variables single precision partial derivative.

The final step is to calculate and report gradients and model sensitivities using the ADGEN utility BSOLVE. Some GRESS/ADGEN utility routines and computer code program using model adjustment method will be presented in Appendices C and D, respectively.

7.2.3 Report of Sensitivities and Gradients

The calculated values of output variable SIGC (activation value of output node) were selected as results of interest for gradient and sensitivity calculations. These calculations can be represented mathematically as

$$Gradient \equiv \left(\frac{dc_j}{da_h}\right), \left(\frac{dc_j}{dQ_m}\right), \tag{7-1}$$

and

$$Sensitivity \equiv \left(\frac{dc_j}{da_h}\right)\frac{a_h}{c_j}, \left(\frac{dc_j}{dQ_m}\right)\frac{Q_m}{c_j}, \tag{7-2}$$

for all $h = 1,2,...,NI$, $j = 1,2,...,NO$, and $m = 1,2...,NW$, and where a_h, Q_m, and c_j are defined in Chapter VI.

7.2.3.1 Sensitivity Analysis of RNA Model

For the RNA model, a group of twenty-six data sets from measurements on subcritical assemblies were analyzed with several neural network architectures. Three neural network topologies were evaluated with this data. One uses nine inputs with five hidden layer nodes

and one single output node (i.e., 9x5x1 input, hidden, and output nodes), another uses nineteen inputs with three hidden layer nodes and one single output node (i.e., 19x3x1 input, hidden, and output nodes), and the third uses nineteen inputs with five hidden layer nodes and one single output node (i.e., 19x5x1 input, hidden, and output nodes).

All REAL variables input via FORTRAN READ statements were automatically included in the parameter dictionary as parameters of interest. A total of 65, 83, and 125 input and weight vector parameters for each data set analyzed representing the three neural network architectures (i.e., 9x5x1, 19x3x1, and 19x5x1, respectively) were evaluated.

Tables 3, 4, and 5 list selected parameters of interest and their gradient and sensitivity information for the SIGC output variable. Table 6 lists the most sensitive parameters of RNA model of the three neural network architectures. The parameters with negative sensitivities indicate that a 1 percent increase in such a parameter value will cause a corresponding sensitivity value percent decrease in the response of interest, SIGC. For example, a 1% increase in an input parameter value of $a(1) = 0.582$ will cause a value of 0.00705 decrease in SIGC (see the first entry of Table 3).

7.2.3.2 Sensitivity Analysis of Nuclear In-Core Fuel Management Model

For the nuclear in-core fuel management model, a group of fifteen data sets from relative power measurements of core assemblies at BOC were analyzed with a neural network architecture consisting of thirty-one inputs, fifteen hidden layer nodes, and thirty-one output nodes (i.e., 31x15x31 input, hidden, and output nodes).

A total of 1007 input and weight vector parameters for each data set analyzed representing the neural network 31x15x31 architecture were evaluated. Table 7 lists selected

Table 3.

Gradient and Sensitivity Calculations of Selected Parameters of Interest of RNA Model for SIGC Output Variable Using 9x5x1 Neural Network Architecture

Parameter	Gradient	Sensitivity
a(1)*	-1.014E-02	-7.050E-03
a(2)*	-4.996E-02	-1.832E-02
a(3)*	-8.048E-03	-5.114E-04
a(4)*	-6.243E-02	-3.564E-03
a(5)*	-7.538E-03	-1.621E-05
a(6)*	-9.017E-03	3.641E-06
a(7)*	4.346E-02	5.918E-04
a(8)*	-2.096E-02	4.810E-05
a(9)*	2.764E-02	-1.611E-02
θ(1)*	-1.358E-02	1.623E-02
θ(2)*	1.788E-02	-2.135E-02
θ(3)*	2.143E-02	-1.993E-02
θ(4)*	2.178E-02	1.369E-02
θ(5)*	1.771E-02	-1.354E-02
Γ(1)*	1.363E-01	1.312E-01
v(1,1)	-7.906E-03	-7.363E-03
v(2,1)	-4.170E-03	-1.383E-03
v(3,1)	-7.227E-04	-4.912E-04
v(4,1)	-6.493E-04	-6.880E-04
v(5,1)	-2.445E-05	-2.388E-05
v(6,1)	4.591E-06	-3.293E-06
v(7,1)	-1.548E-04	6.239E-05
v(8,1)	2.608E-05	-3.107E-05
v(9,1)	6.629E-03	-2.710E-03
v(1,2)	1.041E-02	-2.810E-03

Table 3. (Continued)

Parameter	Gradient	Sensitivity
v(2,2)	5.489E-03	-3.026E-03
v(3,2)	9.512E-04	4.920E-04
v(4,2)	8.547E-04	-1.411E-04
v(5,2)	3.218E-05	1.234E-07
v(6,2)	-6.043E-06	5.128E-06
v(7,2)	2.038E-04	-5.722E-05
v(8,2)	-3.433E-05	3.453E-05
v(9,2)	-8.725E-03	-6.706E-03
v(1,3)	1.247E-02	-7.168E-04
v(2,3)	6.579E-03	-3.982E-03
v(3,3)	1.140E-03	-6.007E-04
v(4,3)	1.024E-03	-1.013E-03
v(5,3)	3.856E-05	-8.402E-06
v(6,3)	-7.244E-06	-5.051E-06
v(7,3)	2.443E-04	1.732E-04
v(8,3)	-4.115E-05	1.696E-05
v(9,3)	-1.045E-02	8.577E-04
v(1,4)	1.267E-02	1.157E-02
v(2,4)	6.687E-03	-5.812E-03
v(3,4)	1.158E-03	3.270E-04
v(4,4)	1.041E-03	-7.804E-04
v(5,4)	3.921E-05	4.666E-05
v(6,4)	-7.362E-06	7.508E-06
v(7,4)	2.483E-04	1.814E-04
v(8,4)	-4.182E-05	2.845E-05
v(9,4)	-1.063E-02	-1.247E-03
v(1,5)	1.030E-02	-7.736E-03
v(2,5)	5.436E-03	-4.116E-03

Table 3. (Continued)

Parameter	Gradient	Sensitivity
v(3,5)	9.420E-04	-2.385E-04
v(4,5)	8.464E-04	-9.416E-04
v(5,5)	3.187E-05	-3.070E-05
v(6,5)	-5.985E-06	-6.522E-07
v(7,5)	2.018E-04	2.321E-04
v(8,5)	-3.400E-05	-7.935E-07
v(9,5)	-8.641E-03	-6.304E-03
w(1,1)*	6.063E-02	-2.922E-02
w(2,1)*	2.344E-02	2.579E-02
w(3,1)*	3.710E-02	3.516E-02
w(4,1)*	9.084E-02	7.801E-02
w(5,1)*	2.396E-02	2.566E-02

*Indicates parameters that were also verified by GRESS/CHAIN option.

Table 4.

Gradient and Sensitivity Calculations of Selected Parameters of Interest of RNA Model for SIGC Output Variable Using 19x3x1 Neural Network Architecture

Parameter	Gradient	Sensitivity
a(1)*	3.424E-02	1.415E-02
a(2)*	8.707E-03	5.087E-03
a(3)*	2.772E-02	7.796E-03
a(4)*	1.211E-02	2.067E-03
a(5)*	-1.808E-02	-1.078E-03
a(6)*	-1.132E-02	-7.868E-04
a(7)*	-3.567E-03	-3.595E-05
a(8)*	4.565E-02	7.719E-04
a(9)*	1.758E-02	3.503E-04
a(10)*	1.642E-02	2.258E-04
a(11)*	3.371E-02	4.468E-04
a(12)*	1.585E-02	6.689E-05
a(13)*	-4.908E-02	-2.325E-04
a(14)*	7.610E-03	-6.651E-05
a(15)*	2.039E-02	1.141E-04
a(16)*	2.482E-03	8.311E-06
a(17)*	-2.780E-02	-4.548E-04
a(18)*	-3.901E-02	1.123E-04
a(19)*	-2.860E-03	2.337E-03
$\theta(1)$*	-3.964E-02	6.637E-02
$\theta(2)$*	1.066E-02	-1.786E-02
$\theta(3)$*	3.418E-04	-4.455E-04
$\Gamma(1)$*	2.405E-01	2.119E-01
v(1,1)	-9.792E-03	1.049E-02
v(5,1)	-1.411E-03	-1.344E-03

Table 4. (Continued)

Parameter	Gradient	Sensitivity
v(9,1)	-4.717E-04	1.440E-04
v(11,1)	-3.048E-04	4.358E-04
v(15,1)	-1.324E-04	7.581E-05
v(19,1)	1.934E-02	-6.225E-03
v(1,2)	2.635E-03	3.554E-03
v(5,2)	3.797E-04	2.753E-04
v(9,2)	1.269E-04	2.117E-04
v(11,2)	8.203E-05	1.253E-05
v(15,2)	3.563E-05	3.838E-05
v(19,2)	-5.205E-03	8.317E-03
v(1,3)	8.553E-05	1.102E-04
v(5,3)	1.216E-05	-8.986E-06
v(9,3)	4.067E-06	-5.493E-06
v(11,3)	2.628E-06	-1.484E-06
v(15,3)	1.142E-06	-.312E-07
v(19,3)	-1.668E-04	2.444E-04
w(1,1)*	5.217E-02	-8.476E-02
w(2,1)*	9.945E-02	3.045E-02
w(3,1)*	1.192E-01	1.134E-03

*Indicates parameters that were also verified by GRESS/CHAIN option.

Table 5.

Gradient and Sensitivity Calculations of Selected Parameters of Interest of RNA Model for SIGC Output Variable Using 19x5x1 Neural Network Architectures

Parameter	Gradient	Sensitivity
a(1)	6.326E-03	2.375E-03
a(2)	-1.228E-03	-6.516E-04
a(3)	-1.256E-02	-3.207E-03
a(4)	-1.496E-02	-2.320E-03
a(5)	9.734E-03	5.267E-04
a(6)	7.498E-03	4.730E-04
a(7)	5.402E-03	4.934E-05
a(8)	3.186E-02	4.891E-04
a(9)	-8.159E-03	-1.476E-04
a(10)	-8.181E-03	-1.021E-04
a(11)	1.564E-02	1.828E-04
a(12)	2.652E-02	1.016E-04
a(13)	-1.601E-03	-6.888E-06
a(14)	-5.146E-02	4.053E-04
a(15)	-2.277E-02	-1.156E-04
a(16)	2.514E-02	7.642E-05
a(17)	-3.905E-02	-5.799E-04
a(18)	-3.449E-02	9.017E-05
a(19)	-4.694E-02	3.482E-02
$\theta(1)$	-2.835E-02	4.308E-02
$\theta(2)$	-1.519E-02	2.309E-02
$\theta(3)$	-4.981E-03	5.894E-03
$\theta(4)$	1.859E-02	1.487E-02
$\theta(5)$	-1.677E-02	1.632E-02
$\Gamma(1)$	2.251E-01	2.756E-01

Table 5. (Continued)

Parameter	Gradient	Sensitivity
v(1,1)*	-7.001E-03	-8.298E-03
v(5,1)*	-1.009E-03	-1.254E-03
v(9,1)*	-3.373E-04	1.753E-04
v(11,1)*	-2.179E-04	1.015E-04
v(15,1)*	-9.467E-05	4.614E-05
v(19,1)*	1.383E-02	1.943E-02
v(1,2)*	-3.752E-03	1.289E-03
v(5,2)*	-5.408E-04	-2.640E-06
v(9,2)*	-1.809E-04	-1.768E-04
v(11,2)*	-1.168E-04	1.059E-04
v(15,2)*	-5.074E-05	4.190E-08
v(19,2)*	7.414E-03	1.742E-03
v(1,3)*	-1.230E-03	8.998E-05
v(5,3)*	-1.773E-04	4.914E-05
v(9,3)*	-5.927E-05	6.186E-06
v(11,3)*	-3.830E-05	1.119E-05
v(15,3)*	-1.664E-05	-1.228E-05
v(19,3)*	2.431E-03	-2.883E-03
v(1,4)*	4.592E-03	5.336E-03
v(5,4)*	6.619E-04	1.002E-03
v(9,4)*	2.213E-04	3.305E-05
v(11,4)*	1.429E-04	-2.074E-04
v(15,4)*	6.210E-05	-8.169E-05
v(19,4)*	-9.074E-03	4.139E-03
v(1,5)*	-4.143E-03	3.958E-03
v(5,5)*	-5.972E-04	7.321E-04
v(9,5)*	-1.996E-04	-1.853E-04
v(11,5)*	-1.290E-04	1.716E-04

Table 5. (Continued)

Parameter	Gradient	Sensitivity
v(15,5)*	-5.603E-05	-6.783E-05
v(19,5)*	8.186E-03	1.239E-02
w(1,1)	6.066E-02	-5.898E-02
w(2,1)	4.967E-02	-2.963E-02
w(3,1)	7.335E-02	-1.123E-02
w(4,1)	1.438E-01	7.829E-02
w(5,1)	3.636E-02	-3.041E-02

*Indicates parameters that were also verified by GRESS/CHAIN option.

Table 6.

Most Sensitive Parameters of RNA Model for the Three Neural Network Architectures

Model Data	Neural Network Topology	Most Sensitive Parameters	Gradient	Sensitivity
RNA	9x5x1	$\theta(1)$	-1.358E-02	1.623E-02
		$\theta(4)$	2.178E-02	1.369E-02
		$\Gamma(1)$	1.363E-01	1.312E-01
		v(1,4)	1.267E-02	1.157E-02
		w(4,1)	9.084E-02	7.801E-02
	19x3x1	a(1)	3.424E-02	1.415E-02
		$\theta(1)$	-3.964E-02	6.637E-02
		$\Gamma(1)$	2.405E-01	2.119E-01
		v(1,1)	-9.792E-02	1.049E-02
		w(2,1)	9.945E-02	3.045E-02
	19x5x1	a(19)	-4.694E-02	3.482E-02
		$\theta(1)$	-2.835E-02	4.308E-02
		$\Gamma(1)$	2.251E-01	2.756E-01
		v(19,1)	1.383E-02	1.943E-02
		w(4,1)	1.438E-01	7.829E-02

Table 7.

Gradient and Sensitivity Calculations of Selected Parameters of Interest of In-Core Fuel Management Model for SIGC Output Variable Using 31x15x31 Neural Network Architecture

Parameter	Gradient	Sensitivity
a(1)*	2.001E-03	2.088E-02
a(2)*	-7.346E-03	-8.355E-02
a(3)*	-6.221E-03	-6.491E-02
a(4)*	-4.213E-03	-4.395E-02
a(5)*	1.048E-02	1.093E-01
a(6)*	-5.994E-03	-6.818E-02
a(7)*	-3.561E-03	-3.715E-02
a(8)*	-4.079E-03	-5.826E-02
a(9)*	-7.452E-03	-8.476E-02
a(10)*	2.697E-03	3.853E-02
a(11)*	6.197E-03	6.466E-02
a(12)*	1.027E-02	1.467E-01
a(13)*	-8.980E-03	-9.369E-02
a(14)*	-1.268E-03	-1.443E-02
a(15)*	9.898E-03	1.414E-01
a(16)*	1.443E-02	1.641E-01
a(17)*	4.701E-03	4.905E-02
a(18)*	5.566E-03	5.808E-02
a(19)*	-8.816E-03	-1.003E-01
a(20)*	-1.337E-02	-1.909E-01
a(21)*	-8.601E-03	-1.228E-01
a(22)*	-5.265E-03	-5.494E-02
a(23)*	1.224E-02	1.392E-01
a(24)*	3.958E-04	5.654E-03
a(25)*	-9.799E-03	-1.115E-01

Table 7. (Continued)

Parameter	Gradient	Sensitivity
a(26)*	1.456E-02	2.080E-01
a(27)*	-8.532E-03	-9.705E-02
a(28)*	3.824E-03	4.350E-02
a(29)*	-1.678E-04	-2.397E-03
a(30)*	-1.704E-02	-1.937E-01
a(31)*	2.148E-03	2.241E-02
$\theta(1)$	5.687E-03	-4.864E-02
$\theta(5)$	-8.303E-04	4.545E-03
$\theta(10)$	-5.330E-03	-3.484E-02
$\theta(15)$	-3.794E-03	2.361E-02
$\Gamma(1)$	1.032E-01	-5.598E-01
$\Gamma(5)$	-8.336E-03	7.129E-02
$\Gamma(10)$	2.598E-04	-1.729E-03
$\Gamma(15)$	2.395E-04	1.078E-03
$\Gamma(20)$	3.016E-04	2.078E-03
$\Gamma(25)$	6.843E-03	-2.815E-03
$\Gamma(31)$	-5.385E-03	2.333E-02
v(1,1)	6.938E-03	3.376E-02
v(6,1)	7.564E-03	5.840E-02
v(8,1)	9.497E-03	7.701E-02
v(1,2)	-1.017E-02	-3.766E-02
v(5,3)	3.169E-04	1.317E-03
v(10,4)	4.000E-04	8.995E-05
v(15,5)	-1.386E-03	9.871E-03
v(20,6)	5.037E-04	-3.626E-03
v(25,7)	-3.158E-04	1.401E-03
v(31,8)	1.494E-05	4.212E-05
v(1,9)	8.348E-03	-4.480E-02
v(5,10)	-6.503E-03	-6.036E-02

Table 7. (Continued)

Parameter	Gradient	Sensitivity
v(10,11)	5.114E-03	-3.408E-03
v(15,12)	1.236E-02	4.311E-02
v(20,13)	-1.059E-03	-7.652E-03
v(25,14)	-7.163E-03	-1.199E-02
v(31,15)	-5.045E-03	7.661E-03
w(1,1)	6.122E-02	1.195E-01
w(3,1)	1.029E-01	6.906E-01
w(5,1)	9.924E-02	-1.824E-01
w(7,1)	1.029E-01	-6.746E-01
w(9,1)	8.697E-02	3.712E-01
w(11,1)	6.623E-03	3.831E-01
w(13,1)	1.022E-01	-6.875E-02
w(15,1)	7.041E-03	-3.482E-02
w(1,2)	-6.335E-03	-2.011E-02
w(3,3)	-4.628E-03	-3.639E-03
w(5,4)	1.134E-02	-1.462E-02
w(7,5)	1.236E-02	8.226E-02
w(9,6)	4.012E-04	2.182E-03
w(11,7)	-2.897E-04	-2.004E-03
w(13,8)	1.495E-05	-7.334E-06
w(15,9)	1.430E-02	1.951E-02
w(15,10)	-8.910E-03	3.215E-02
w(15,15)	-5.045E-03	1.462E-02
w(15,20)	-1.087E-02	-2.584E-02
w(15,25)	3.455E-04	-1.346E-03
w(15,31)	2.922E-04	1.824E-03

*Indicates parameters that were also verified by GRESS/CHAIN option.

parameters of interest and their gradient and sensitivity information for the SIGC output variable. Table 8 lists the most sensitive parameters of nuclear in-core fuel management model for the given neural network architecture.

7.2.3.3 Sensitivity Analysis of PET Model

For the PET model, a group of twenty data sets were analyzed with a neural network architecture consisting of twelve inputs, thirty hidden layer nodes, and one single output node (i.e., 12x30x1 input, hidden, and output nodes).

A total of 433 input and weight vector parameters for each data set analyzed representing the neural network 12x30x1 architecture were evaluated. Table 9 lists selected parameters of interest and their gradient and sensitivity information for the SIGC output variable. Table 10 lists the most sensitive parameters of PET model for the given neural network architecture.

7.2.4 Verification of Sensitivities

The capability of the enhanced version of MFN.FOR to correctly calculate analytic first derivatives and model sensitivities using computer calculus was tested by comparison of the ADGEN results with derivatives and sensitivities calculated by the GRESS/CHAIN option. All sensitivities calculated with the CHAIN option agreed with those calculated by ADGEN to seven significant digits. This close agreement is due to the fact that both CHAIN and ADGEN systems are calculating analytic derivatives. Sensitivities that are verified using the GRESS/CHAIN option are flagged with an asterisk (*).

Table 8.

Most Sensitive Parameters of In-Core Fuel Management Model for 31x15x31 Neural Network Architecture

Model Data	Neural Network Topology	Most Sensitive Parameters	Gradient	Sensitivity
In-core Fuel Management	31x15x31	a(5)	1.048E-02	1.093E-01
		a(12)	1.027E-02	1.467E-01
		a(15)	9.898E-03	1.414E-01
		a(16)	1.443E-02	1.641E-01
		a(23)	1.224E-02	1.392E-01
		a(26)	1.456E-02	2.080E-01
		v(8,1)	9.497E-03	7.701E-02
		w(1,1)	6.122E-02	1.195E-01
		w(3,1)	1.029E-01	6.906E-01
		w(9,1)	8.697E-02	3.712E-01
		w(11,1)	6.623E-03	3.831E-01
		w(7,5)	1.236E-02	8.226E-02

Table 9.

Gradient and Sensitivity Calculations of Selected Parameters of Interest of PET Model for SIGC Output Variable Using 12x30x1 Neural Network Architecture

# Data Set Pattern	Input Data Parameters	Gradient	Sensitivity
1	a(1)*	-6.163E-03	-1.287E-03
	a(2)*	6.828E-03	1.374E-02
	a(3)*	5.918E-04	5.88E-04
	a(4)*	-4.692E-03	-4.228E-03
	a(5)*	-4.010E-03	-1.712E-04
	a(6)*	6.913E-03	2.466E-03
	a(7)*	7.082E-03	4.765E-03
	a(8)*	-1.216E-02	-2.701E-04
	a(9)*	4.551E-03	9.145E-04
	a(10)*	1.814E-03	5.140E-04
	a(11)*	-4.123E-03	-7.566E-04
	a(12)*	1.029E-02	2.350E-03
2	a(1)	-1.286E-02	-6.287E-04
	a(2)	2.442E-02	1.974E-02
	a(3)	3.107E-04	2.932E-02
	a(4)	2.081E-04	6.681E-04
	a(5)	3.840E-03	1.529E-03
	a(6)	1.100E-02	3.696E-03
	a(7)	7.609E-04	3.349E-03
	a(8)	-1.1362E-02	-1.846E-03
	a(9)	1.769E-02	5.002E-03
	a(10)	1.313E-02	3.028E-03
	a(11)	-1.148E-02	-1.350E-03
	a(12)	2.842E-02	6.975E-03
3	a(1)	-2.592E-02	-5.743E-03
	a(2)	6.416E-02	6.159E-02
	a(3)	5.822E-03	3.347E-03
	a(4)	1.314E-02	3.579E-03
	a(5)	1.529E-02	5.589E-03
	a(6)	1.457E-02	1.882E-03
	a(7)	1.020E-02	3.036E-03
	a(8)	-2.287E-02	-8.418E-03
	a(9)	4.470E-02	1.919E-02
	a(10)	4.013E-02	2.921E-03
	a(11)	-3.248E-02	-6.772E-03
	a(12)	7.473E-02	1.895E-02

Table 9. (Continued)

# Data Set Pattern	Input Data Parameters	Gradient	Sensitivity
4	a(1)	-7.928E-02	-3.025E=03
	a(2)	-2.395E-01	-1.207E-02
	a(3)	7.622E-02	2.570E-02
	a(4)	-1.249E-02	-3.172E-03
	a(5)	2.688E-02	2.998E-03
	a(6)	3.720E-02	4.726E-03
	a(7)	3.731E-02	5.787E-03
	a(8)	1.738E-02	3.986E-03
	a(9)	-1.476E-02	-2.591E-03
	a(10)	-1.271E-01	-9.987E-03
	a(11)	1.192E-01	1.614E-02
	a(12)	1.847E-01	-3.224E-02
5	a(1)	-4.134E-02	-9.953E-04
	a(2)	-1.141E-01	-5.195E-02
	a(3)	3.872E-02	1.787E-02
	a(4)	1.172E-03	3.564E-04
	a(5)	2.029E-02	5.827E-03
	a(6)	1.485E-02	2.856E-03
	a(7)	1.092E-02	1.821E-03
	a(8)	1.020E-02	2.224E-03
	a(9)	-7.807E-04	-1.532E-04
	a(10)	-5.422E-02	-1.240E-02
	a(11)	5.914E-02	7.476E-03
	a(12)	-9.038E-02	-2.193E-02
6	a(1)	-5.888E-02	-6.462E-04
	a(2)	-1.725E-01	-2.353E-02
	a(3)	5.314E-02	1.207E-02
	a(4)	-5.700E-04	-5.397E-04
	a(5)	2.496E-02	4.519E-03
	a(6)	2.729E-02	2.826E-04
	a(7)	2.669E-02	7.815E-03
	a(8)	1.219E-02	1.535E-03
	a(9)	-8.529E-03	-6.741E-04
	a(10)	-8.992E-02	-2.009E-02
	a(11)	8.809E-02	1.267E-02
	a(12)	-1.337E-01	-2.209E-02

Table 9. (Continued)

# Data Set Pattern	Input Data Parameters	Gradient	Sensitivity
7	a(1)	-7.887E-02	7.978E-04
	a(2)	-2.374E-01	3.202E-03
	a(3)	7.415E-02	1.674E-2
	a(4)	-1.194E-02	-2.435E-03
	a(5)	3.079E-02	5.522E-03
	a(6)	3.770E-02	3.497E-03
	a(7)	3.980E-02	6.291E-03
	a(8)	1.560E-02	1.938E-03
	a(9)	-1.427E-02	-1.726E-04
	a(10)	-1.262E-01	-2.587E-02
	a(11)	1.223E-01	2.060E-02
	a(12)	-1.845E-01	-2.129E-02
8	a(1)	-1.823E-02	-3.320E-04
	a(2)	-5.745E-02	4.578E-04
	a(3)	1.665E-02	2.474E-03
	a(4)	3.093E-04	4.535E-04
	a(5)	1.484E-02	2.314E-03
	a(6)	6.672E-04	2.097E-04
	a(7)	4.109E-04	5.086E-04
	a(8)	1.286E-02	2.026E-03
	a(9)	1.110E-04	2.153E-04
	a(10)	-2.506E-02	-3.911E-03
	a(11)	2.686E-02	3.074E-03
	a(12)	-4.324E-02	-6.321E-03
9	a(1)	-6.634E-03	2.056E-04
	a(2)	-2.237E-02	-4.497E-04
	a(3)	6.690E-03	1.484E-03
	a(4)	7.602E-03	1.997E-03
	a(5)	1.224E-02	1.612E-03
	a(6)	7.080E-04	9.087E-5
	a(7)	-1.891E-03	-2.838E-04
	a(8)	1.289E-02	1.818E-03
	a(9)	2.136E-04	2.369E-04
	a(10)	-4.998E-03	-7.930E-04
	a(11)	9.737E-03	1.285E-04
	a(12)	-1.591E-02	1.935E-03

Table 9. (Continued)

# Data Set Pattern	Input Data Parameters	Gradient	Sensitivity
10	a(1)	-3.706E-03	-7.576E-05
	a(2)	-1.456E-02	-2.756E-03
	a(3)	4.717E-03	1.093E-03
	a(4)	7.969E-03	9.637E-04
	a(5)	1.351E-02	2.137E-03
	a(6)	-5.461E-04	-7.678E-05
	a(7)	-4.434E-03	-6.494E-04
	a(8)	1.380E-02	1.933E-03
	a(9)	4.678E-03	8.502E-04
	a(10)	4.236E-04	9.347E-05
	a(11)	6.806E-03	8.711E-04
	a(12)	-1.801E-02	-1.658E-03

*Indicates parameters that were also verified by GRESS/ADGEN option.

Table 10.

Most Sensitive Input Data Parameters of PET Model for 12x30x1 Neural Network Architecture

Model Data	Neural Network Topology	# of Data Set	Most Sensitive Input Parameters	Gradient	Sensitivity
Pet Model	12x30x1	1	a(2)	6.828E-03	1.374E-02
		2	a(2)	2.441E-02	1.973E-02
		3	a(2)	6.416E-02	6.159E-02
		4	a(3)	7.622E-02	2.570E-02
		5	a(3)	3.871E-02	1.787E-02
		6	a(11)	8.809E-02	1.267E-02
		7	a(11)	1.223E-01	2.060E-02
		8	a(11)	2.686E-02	3.074E-03
		9	a(4)	7.602E-03	1.996E-03
		10	a(5)	1.351E-02	2.136E-03

Although GRESS is an efficient automated software system for calculating sensitivities of model results of interest to input data parameters in existing FORTRAN computer models, it lacks the ability of the "reset" sensitivity option. This option has to be considered for some applications such as the sensitivity calculations in the adjustment algorithm.

7.3 UNCERTAINTY AND DATA ADJUSTMENT ANALYSES

As an example for the application of uncertainty and data adjustment analyses to ANS systems, the sensitivity analysis study of neural network's responses for the aforementioned models (i.e., RNA, in-core fuel management, and PET models) were considered. The responses of interest were the activation values of output nodes, c_j's, of the considered neural network architectures.

The uncertainty matrix of input parameters and their correlations, U_x, is given in Table 11. The sensitivity functions represented as

$$(S)_{nj} = \partial R_n / \partial \xi_j \ , \tag{7-3}$$

where $\boldsymbol{R} = (R_1, R_2, ..., R_N)$ and $\xi = (\xi_1, \xi_2, ..., \xi_J)$ are the response of interest and input data field vectors, respectively, were calculated using the GRESS/ADGEN adjoint sensitivity code system. The sensitivities with respect to input data parameters combined with the input data uncertainty matrix were used in order to calculate the uncertainty matrix of the residuals, U_r (Eq. (6-6)). The sensitivities with respect to weight vector parameters combined with the inverse uncertainty matrix of residuals were used in order to calculate the uncertainty matrix of the weight vectors, U_w (Eq. (6-8)).

The uncertainty matrix of the responses, U_R, was then calculated using Eq. (6-7) in

Table 11.

Correlation Matrix of Input Data Parameters In Standard Deviation Representation

Correlation Matrix (U_x)

0.033	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.033	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.033	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.033	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.033	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.033	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.033	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.033	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.033	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	σ_r*

* $\sigma_r = \sqrt{\frac{1}{K-1} \sum_{j=1}^{K} (r_j - c_j)^2}$ = standard deviation of residuals

order to obtain the model standard deviations in U_R. The adjustment procedure of the weight vector parameters was then followed using Eq. (6-4) in order to obtain a new set of adjusted weight vector parameters. The sensitivity matrix of the new set of adjusted weight vector parameters were recalculated (with respect to adjusted weight parameters) using the GRESS/ADGEN sensitivity code system. The uncertainty matrix of the adjusted responses, U_R (Eq. (6-7)), utilizing the uncertainty matrix of adjusted weight vector parameters, U_w (Eq. (6-8)), was recalculated to obtain model standard deviations in the adjusted responses.

For the RNA model, the fractional deviations of reference and adjusted responses for the three neural network topologies (i.e., 9x5x1, 19x3x1, and 19x5x1, respectively) are given in Tables 12, 13, and 14, respectively. The fractional deviations of reference and adjusted responses for reactor in-core fuel management model using the 31x15x31 neural network architecture are given in Table 15. Three selected data sets are reported therein. Finally, the results of the fractional deviations of reference and adjusted responses for the PET model using 12x30x1 neural network architecture are given in Table 16.

It is seen that calculating the uncertainties of the responses of interest with the adjusted weight vector parameters yields promising results. From these results, a considerable improvement in the uncertainty of the responses using model adjustment method is achieved.

7.4 SUMMARY

The usefulness of the uncertainty analysis based on sensitivity methodology as well as the use of model adjustment method for improving (reducing) uncertainties was demonstrated in this chapter.

Table 12.

Effect of Adjustment on Response Values and their Uncertainties
Using 9x5x1 Neural Network Architecture for RNA Model

# Data Set Pattern	Fractional Deviations	
	Reference Model Data (%)	Adjusted Model Data (%)
1	21.05	14.94
2	21.41	15.20
3	22.12	15.70
4	26.31	18.68
5	20.83	14.79
6	20.03	14.40
7	11.28	8.01
8	20.41	14.46
9	22.57	16.03
10	19.53	13.86
11	19.97	14.18
12	18.65	13.25
13	17.42	12.37
14	18.72	13.30
15	18.38	13.05
16	17.73	12.58
17	17.54	12.45
18	18.15	12.88
19	17.74	12.60
20	17.72	12.58
21	15.88	11.27
22	15.87	11.26
23	15.77	11.19
24	15.88	11.27
25	14.83	10.53
26	16.01	11.37

Table 13.

Effect of Adjustment on Response Values and their Uncertainties Using 19x3x1 Neural Network Architecture for RNA Model

# Data Set Pattern	Fractional Deviations	
	Reference Model Data (%)	Adjusted Model Data (%)
1	24.34	17.70
2	24.80	18.04
3	25.65	18.66
4	31.22	22.70
5	24.24	17.62
6	23.92	17.39
7	12.32	8.96
8	23.44	17.05
9	26.20	19.06
10	22.49	16.36
11	23.13	16.82
12	21.47	15.62
13	19.97	14.52
14	21.59	15.70
15	21.15	15.38
16	20.35	14.80
17	20.13	14.62
18	20.85	15.16
19	20.34	14.80
20	20.32	14.77
21	18.13	13.20
22	18.12	13.18
23	18.01	13.09
24	18.14	13.20
25	16.89	12.28
26	18.29	13.30

Table 14.

Effect of Adjustment Response Values and their Uncertainties
Using 19x5x1 Neural Network Architecture for RNA Model

# Data Set Pattern	Fractional Deviations	
	Reference Model Data (%)	Adjusted Model Data (%)
1	33.34	23.63
2	33.94	24.06
3	35.08	24.86
4	42.42	30.07
5	33.20	23.55
6	32.77	23.25
7	17.11	12.13
8	32.14	22.78
9	35.82	25.42
10	30.87	21.89
11	31.72	22.48
12	29.51	20.95
13	27.49	19.50
14	29.67	21.05
15	29.07	20.63
16	28.01	19.87
17	27.71	19.66
18	28.66	20.34
19	27.99	19.86
20	27.96	19.84
21	25.01	17.73
22	25.00	17.72
23	24.84	17.61
24	25.02	17.75
25	23.37	16.55
26	25.22	17.90

Table 15.

Effect of Adjustment on Selected Response Values and their Uncertainties Using 31x15x31 Neural Network Architecture of In-Core Fuel Management Model

# Data Set Pattern	Fractional Deviations	
	Reference Model Data (%)	Adjusted Model Data (%)
6	21.17	10.39
	19.47	9.56
	13.01	6.39
	6.21	3.05
	4.81	2.36
	4.36	2.14
	5.35	2.63
	10.05	4.93
	17.82	8.75
	13.21	6.48
	6.93	3.40
	5.23	2.57
	4.60	2.25
	5.89	2.90
	10.94	5.37
	11.75	5.77
	8.54	4.20
	7.00	2.94
	5.48	2.69
	8.62	3.68
	13.45	6.60
	8.24	4.04
	6.32	3.10
	6.88	3.38
	9.22	4.52
	19.15	9.40
	6.95	3.41
	7.89	3.87
	13.06	6.41
	10.48	5.14
	18.94	9.30

Table 15. (Continued)

# Data Set Pattern	Fractional Deviations	
	Reference Model Data (%)	Adjusted Model Data (%)
7	24.77	12.16
	22.64	11.11
	14.45	7.09
	6.58	3.23
	4.98	2.45
	4.34	2.13
	5.25	2.57
	9.28	4.56
	20.16	9.89
	14.03	6.88
	7.33	3.60
	5.16	2.53
	4.61	2.26
	5.52	2.71
	9.91	4.86
	12.48	6.13
	9.11	4.47
	6.21	3.05
	5.38	2.64
	6.96	3.42
	11.80	5.79
	8.71	4.27
	6.40	3.14
	6.70	3.28
	8.57	4.21
	17.03	8.36
	6.95	3.41
	7.84	3.85
	12.29	6.03
	10.42	5.11
	18.72	9.19

Table 15. (Continued)

# Data Set Pattern	Fractional Deviations	
	Reference Model Data (%)	Adjusted Model Data (%)
9	37.17	18.25
	29.51	14.48
	19.70	9.67
	11.76	5.77
	5.71	2.80
	4.46	2.20
	3.58	1.76
	5.03	2.47
	23.77	11.67
	19.01	9.33
	12.34	6.06
	7.00	3.44
	4.20	2.06
	3.78	1.86
	5.61	2.75
	21.07	10.36
	13.95	6.85
	6.46	3.17
	5.55	2.72
	5.14	2.52
	9.18	4.51
	10.38	5.10
	7.75	3.80
	6.23	3.06
	8.83	4.33
	15.14	7.43
	8.32	4.08
	8.01	3.93
	13.96	6.85
	13.29	6.53
	25.51	12.52

Table 16.

Effect of Adjustment on Response Values and Their Uncertainties using 12x30x1 Neural Network Architecture for PET Model

# Data Set Pattern	Fractional Deviations	
	Reference Model Data (%)	Adjusted Model Data (%)
1	13.31	9.17
2	15.76	10.82
3	18.26	12.61
4	18.82	12.95
5	12.50	8.63
6	19.37	13.38
7	18.55	12.72
8	43.03	29.71
9	65.22	45.03
10	29.33	20.14
11	41.34	28.54
12	35.12	24.25
13	37.98	25.89
14	46.46	32.08
15	42.75	29.47
16	50.31	34.52
17	57.16	39.40
18	47.45	32.76
19	49.29	34.04
20	33.08	22.66

CHAPTER VIII

CONCLUSIONS AND RECOMMENDATIONS

A sensitivity is a measure of the effect of a given input data field parameter on a required response of interest. The sensitivity of a given physical model is defined as the first order derivative of the response of interest with respect to an input data field parameter. If the required response of interest is denoted as $R\ (\xi_1\ ,\ \xi_2\ ,\ ...,\ \xi_J)$ and the input data field parameter by $\xi_j\ (j\ =\ 1\ ,\ 2\ ,\ ...,\ J)$, then the sensitivity is usually defined as $\partial R/\partial \xi_j$. The sensitivity function can also be defined in relative terms, i.e., $\partial R/\partial \xi_j \bullet \xi_j/R$. This definition clearly demonstrates the physical meaning of the sensitivity as the measure of the importance of the response of interest relative to the input data field parameter. The higher the relative sensitivity is, the more important the input parameter in consideration. Thus, one of the important aspects of model sensitivity analysis is to identify the most important input data field parameters in a system.

Sensitivity analysis is also used for uncertainty analysis. The ingredients of the uncertainty analysis based on sensitivity methodology are model sensitivities and the uncertainty (variance-covariance) matrix of the input parameters. When these elements are available, uncertainty analysis based on sensitivity methodology is a straightforward approach. There are three major advantages to sensitivity-uncertainty analysis. These are (1) the identification of important model parameters in a system, (2) the evaluation of the uncertainty of model response of interest is done in a systematic way, and (3) the changes in the uncertainties of model parameters can be incorporated into model equations to obtain a new uncertainty analysis to the responses of interest in a system under investigation.

For a large number of model parameters, the approach to the sensitivity-uncertainty

analysis methodology was the use of the generalized adjoint theory. The formulation of the adjoint theory allows systematic and quantitative screening of model data field parameter space which provides all the information necessary to perform the response uncertainty analysis. In addition, the adjoint theory produces all model sensitivities for one model result of interest using only a single additional (i.e., adjoint) model calculation, regardless of how large the model data field or how complex the physical model including the nonlinear problems.

Model adjustment method, based on the least-squares approach, is a procedure suggested in order to improve (reduce) uncertainties of model data field parameters. The adjusted model data parameters with their reduced uncertainties can be used in order to improve the uncertainties of calculated responses of interest.

The preparation needed for the evaluation of uncertainty and data adjustment analyses of neural networks for reactor physics applications was followed two directions. First, the sensitivity profile was obtained using the GRESS/ADGEN system, and second, the uncertainties of model data parameters and their correlations were evaluated.

The uncertainty analysis based on sensitivity methodology as well as the use of data adjustment method for improving the response uncertainties have shown a considerable amount of improvement in the calculation of uncertainties of adjusted responses of interest. Hence, the adjustment algorithm can be a useful tool for improving the learning rate of multilayer feedforward networks.

The advantage of the adjustment algorithm over the backpropagation algorithm lies in the fact that the former learning algorithm has the capability of assessing model uncertainties of weight vector parameters and calculated responses of interest, provided that sensitivity functions and input data field parameter uncertainties are available. The potential

application of the adjustment algorithm, however, can be used for listing all model data and weight vector parameters that are potentially important contributors to predictive uncertainty, and consequently perform uncertainty analysis by combining the sensitivities (associated with input data and wight vector parameters) with the appropriate model parameter covariances.

For future work study, several areas of research can be investigated. Among these are the utilization of high order sensitivities for certain cases of nonlinear problems. For these cases, the *nth* order sensitivity is defined as $\partial^n R/\partial \xi^n_j$. Other areas are the reset capability of sensitivity analysis in the GRESS system, the utilization of other methods, such as the bias factor and the bias operator, for improving the uncertainty of the evaluated responses, and finally, the utilization of the probabilistic methods for calculating the sensitivity and uncertainty analyses of neural networks.

REFERENCES

REFERENCES

1. Weisbin, C.R., et al., "Sensitivity and Uncertainty Analysis of Reactor Performance Parameters," Advances in Nuclear Science and Technology, Vol, 14, Plenum Press, New York, NY (1982).
2. Oblow, E.M., "Sensitivity Theory for Reactor Thermal-Hydraulics Problems," Nuclear Science and Engineering, 68, 322-337 (1978).
3. Williams, M. L. and C. R. Weisbin, "Sensitivity and Uncertainty Analysis for Linear Functionals of the Time Dependent Nuclide Density Fields," ORNL/TM-5799, Oak Ridge National Laboratory, Oak Ridge, TN, (1977).
4. Weisbin, C.R., et al., "Application of FORSS Sensitivity and Uncertainty Methodology to Fast Reactor Benchmark Analysis," ORNL/TM-5563, Oak Ridge National Laboratory, Oak Ridge, TN (1976).
5. Williams, M.L., "Perturbation and Sensitivity Theory for Reactor Burnup Analysis," Ph.D. Dissertation, University of Tennessee, Knoxville, (1979).
6. Greenspan, E., "A Method for the Optimization of Fusion Reactor Neutronic Characteristics," in Proc. Conf. Math. Models and Computational Techniques for Analysis of Nuclear Systems, P. 162, Ann Arbor, Michigan (1973).
7. Goldstein, H., "A Survey of Cross-Section Sensitivity Analysis as Applied to Radiation Shielding," in Proc. Fifth Int. Conf. Reactor Shielding, Knoxville, TN (1977).
8. Oblow, E. M., et al., "General Sensitivity Theory for Radiation Transport," ORNL/TM-4110, Oak Ridge National Laboratory, Oak Ridge, TN (1974).
9. Helton, J. C., et al., "Uncertainty and Sensitivity Analysis of a Dry Containment Test Problem for the MAEROS Aerosol Model," Nuclear Sci. and Eng., 102, 22-42 (1989).

10. Iman, R. L. and J. C. Helton, "Overview of Methods for Uncertainty and Sensitivity Analysis in Probabilistic Risk Assessment," in Proc. of the ANS/ENS Int. Topical Meeting on Probabilistic Safety Methods and Applications, San Francisco, CA (1985).

11. Helton, J. C. and J. D. Johnson, "Station Blackout Sequence at a Mark 1 BWR," Reliability Engineering and System Safety, 26, 295-328 (1989).

12. Oblow, E.M., "Reactor Cross-Section Sensitivity Studies Using Transport Theory," ORNL/TM-4437, Oak Ridge National Laboratory, Oak Ride, TN (1974).

13. Oblow, E. M., F. G. Pin, and R. Q. Wright, "Sensitivity Analysis Using Computer Calculus: A Nuclear Waste Isolation Application," Nuclear Science and Engineering, 94, 46-65 (1986).

14. Tomovic, R. and M. Vukobratovic, General Sensitivity Theory, Elsevier, New York, NY (1972).

15. Frank, P. M., Introduction to System Sensitivity Theory, Academic Press, New York, NY (1978).

16. Pomraning, G. C., "Variational Principle for Eigenvalue Equation," Journal of Mathematical Physics, 8, 149, (1967).

17. Stacey, W. M., Variational Methods in Nuclear Reactor Physics, Academic Press, New York, NY (1974).

18. Myers, R. H., Response Surface Methodology, Allyn and Bacon, Boston, MA (1971).

19. Oblow, E. M., "Sensitivity Theory from a Differential Viewpoint," Nuclear Science and Engineering, 59, 187 (1976).

20. Cacuci, D. G., "Sensitivity Theory for Nonlinear Systems. I. Nonlinear Functional Analysis Approach," Journal of Mathematical Physics, 22, 2794-2802 (1981).

21. Cacuci, D. G., "Sensitivity Theory for Nonlinear Systems. II. Extensions to Additional Classes of Responses," Journal of Mathematical Physics, 22, 2803-2812 (1981).

22. Dickens, J. F., and W. C. Lennox, "Numerical Simulation of Water Movement in Steady Groundwater Flow Systems," Water Resour. Res., Vol. 12, (1976).

23. Cukier, R. I., H. B. Levine, and K. E. Shuler, "Nonlinear Sensitivity Analysis of Multiparameter Model Systems," Journal of Computational Physics 26, 1-42, Academic Press, Inc. (1978).

24. McRae, G. J., J. W. Tilden, and J. H. Seinfeld, "Global Sensitivity Analysis - A Computational Implementation of the Fourier Amplitude Sensitivity Test (FAST)," Computers and Chemical Engineering Vol. 6, Pergamon Press Ltd. (1982).

25. Chen, H. H., Transient Simulation and Sensitivity Analysis for Transport of Radionuclides in a Saturated-Unsaturated Groundwater Flow System, Ph. D. Dissertation, University of Tennessee, Knoxville (1980).

26. Alsharif, M. M., and I. Ahmad, "Sensitivity Analysis for Radionuclide Transport Through Fractured Media with Matrix Diffusion," PAI Corp., Oak Ridge, TN (1989).

27. Ronen, Y., Uncertainty Analysis, CRC Press, Inc., Boca Raton, Florida (1988).

28. Thomas, R. E., "Uncertainty Analysis," ONWI-380 Office of Nuclear Waste Isolation, Battelle Memorial Institute, Columbus, Ohio (1982).

29. Stucliffe, W. G., "Uncertainty Analysis: An Illustration from Nuclear Waste Package Development," Nuclear and Chemical Waste Management, 5, 131-140 (1984).

30. Worley, B. A., "Deterministic Uncertainty Analysis," ORNL/TM-6428, Oak Ridge National Laboratory, Oak Ridge, TN (1987).

31. Beers, Y., Introduction to the Theory of Error, Addison-Wesley Publishing Company, Inc., 2nd Edition, Reading, MA (1962).

32. Boas, M. L., Mathematical Methods in Physical Sciences, John Wiley and Sons, Inc., 2nd Edition, New York, NY (1983).

33. Lewins, J., Importance: The Adjoint Function, Pergamon Press (1965).

34. Greenspan, H., C. N. Kelber, and D. Okrnet, Computing Methods in Reactor Physics, Gordon and Breach, Section 3.10.2, (1968).

35. Cacuci, D. G., Deterministic Sensitivity Theory for Nonlinear Systems: the Forward and Adjoint Methods. Report No. ORNL/TM-10058, Oak Ridge National Laboratory, Oak Ridge, TN (1986).

36. Hicks, C. R., Fundamental Concepts in the Design of Experiments, Holt, Rinehart, and Winston, New York, NY (1973).

37. Rubinstein, R. V., Simulation and Monte Carlo Method, John Wiley & Sons, New York, NY (1981).

38. Iman, R. L. and W. J. Conover, "A Distribution-Free Approach to Inducing Rank Correlation Among Input Variables," Communications in Statistics B22(3), (1982).

39. Greenspan, E., "Developments in Perturbation Theory," Advances in Nuclear Science and Technology, Vol. 9, Academic Press, New York, NY (1976).

40. Greenspan, E., D. Gilai, and E. M. Oblow, "Second-Order Generalized Perturbation Theory for Source-Driven Systems," Nuclear Science and Engineering, 68, 1-9 (1978).

41. Horwedel, J. E., B. A. Worley, E. M. Oblow, F. G. Pin, and R. Q. Wright." GRESS Version 0.0 User's Manual," ORNL/TM-10835, Oak Ridge National Laboratory, Oak Ridge, TN (1988).

42. Pin, F. G., E. M. Oblow, J. E. Horwedel, and J. L. Lucius, "ADGEN: An Automated Adjoint Code Generator for Large-Scale Sensitivity Analysis," Paper presented at American Nuclear Society Conference, Los Angeles, CA (1987).

43. Horwedel, J. E., F. G. Pin, B. A. Worley, and E. M. Oblow, "EXAP - Precompiler for the GRESS and ADGEN Automated Sensitivity Calculation Systems." Published in the Proc. of the Annual Meeting of ANS, San Diego, CA (1988).

44. Stallman, F. W., "Theory and Practice of General Adjustment and Model Fitting Procedures," ORNL/TM-7896, Oak Ridge National Laboratory, Oak Ridge, TN (1981).

45. Hahn, G. J., Statistical methods in Engineering, John Wiley and Sons, New York, NY (1967).

46. Caudill, M., K. Mahaffey, and C. Butler, Understanding Neural Networks: Computer Explorations, MIT Press, Vol. I and Vol. II, MA (1991).

47. Hecht-Nielsen, R., Neurocomputing, Addison-Wesley Publishing Company, MA (1990).

48. Simpson, P. K., Artificial Neural Systems, Foundations, Paradigms, Applications, and Implementations, Pergamon Press, (1990).

49. Uhrig, R. E., "Use of Neural Networks in Nuclear Power Plant Diagnostics," Tran. of the Int. Conf. on Reliability, Availability, and Maintainability of Nuclear Power Plants, Madrid, Spain, April 9-14, 1989, pp. 310-312 (1989).

50. Uhrig, R. E., "Use of Neural Networks in Nuclear Power Plants," Proc. of the 7th Power Plant Dynamics, Control, and Testing Symposium, May 15-17, 1989, Knoxville, TN, PP. 2601-26.05 (1989).

51. Uhrig, R. E. and L. F. Miller, "Neural Networks for In-Core Fuel Management," ANS 1989 Annual Meeting, Atlanta, GA, June 4-8, 1989. Trans. of the ANS, 59, p 59, (1989).

52. Uhrig, R.E., "Opportunities for Automation and control of the Next Generation of Nuclear Power Plants," Nuclear Technology, Vol. 88, pp. 157-165, (1989).

53. Oblow, E. M., "An Automated Procedure for Sensitivity Analysis Using Computer Calculus," ORNL/TM-8776, Oak Ridge National Laboratory, Oak Ridge, TN (1983).

54. Pin, F. G., E. M. Oblow, and R. Q. Wright, "Automated Sensitivity Analysis Using the GRESS Language," ORNL/TM-6148, Oak Ridge National Laboratory, Oak Ridge, TN (1986).

55. Pin, F. G. and E. M. Oblow, "Sensitivity Analysis of Predictive Models with an Automated Adjoint Generator," Proc. of 9th Annual DOE Low-Level Waste Management Conference, Denver, CO (1987).

56. Pin, F. G., B. A. Worley, E. M. Oblow, R. Q. Wright, and W. V. Harper, "An Automated Sensitivity Analysis Procedure for the Performance Assessment of Nuclear Waste Isolation Systems," Nuclear and Chemical Waste Management, 6, 255 (1986).

57. Worley, B. A., R. Q. Wright, F. G. Pin, and W. V. Harper, "Application of Automated Procedure for Adding a Comprehensive sensitivity Calculation Capability to the ORIGEN 2 Point Depletion and Radiative Decay Code." Nuclear Science and Engineering, 94, 180 (1986).

58. Worley, B. A., R. Q. Wright, and F. G. Pin, "A Finite-Line Heat Transfer Code with Automated Sensitivity-Calculation Capability," ORNL/TM-9975, Oak Ridge National Laboratory, Oak Ridge, TN (1986).

59. Worley, B. A. and J. E. Horwedel, "A Waste Package Performance Assessment Code with Automated Sensitivity-Calculation Capability," ORNL/TM-9976, Oak Ridge National Laboratory, Oak Ridge, TN (1986).

60. King, J. L., A Large Computer Code Application of the ADGEN/EXAP System, M. S. Thesis, University of Tennessee, Knoxville (1988).

61. Horwedel, J. E., "Matrix Reduction Algorithms for GRESS and ADGEN," Oak Ridge National Laboratory, ORNL/TM-11261, Oak Ridge, TN (1989).

62. Rall, L. B., Automatic Differentiation: Techniques and Applications, Springer-Verlag, New York, NY (1981).

63. Oblow, E. M., "GRESS-GRadient-Enhanced Software System," ORNL/TM-9658, Oak Ridge National Laboratory, Oak Ridge, TN (1985).

64. Worley, B. A., et al., "ADGEN-ADjoint GENerator for Computer Models," ORNL/TM-11037, Oak Ridge National Laboratory, Oak Ridge, TN (1989).

65. Miller, L. F., "An Evaluation of Neural Networks for Identification of System Parameters in Reactor Noise Signals," In a Symposium on Nuclear Reactor Surveillance and Diagnostics, Gatlinburg, TN (1991).

66. Miller, L. F., Personal Communications.

67. Freeman, L. M., Freeman and Johnson's Clinical Radionuclide Imaging, Vol. 3, Grune and Stratton, Inc., Orlando, Florida (1986).

68. Cybenko, G., "Approximations by Superpositions of a Sigmoidal Function," Mathematics of Control, Signals and Systems, 2, 803-314 (1989).

69. Stinchcombe, M., and H. White, "Universal Approximation Using Feedforward networks with Non-Sigmoid Hidden Layer Activation Functions," In Proceedings of Int. Joint Conf. on Neural Networks, Washington, D.C., pp. I; 613-617, NY (1989).

70. Hornik, K., M. Stinchcombe, and H. White, "Universal Approximation of an Unknown Mapping and Its Derivatives Using Multilayer Feedforward Networks," Neural Networks, Vol. 3, pp. 551-560, Pergamon Press (1990).

APPENDICES

APPENDIX A

DERIVATION OF SOME EQUATIONS IN THE TEXT

APPENDIX A

1. Derivation of Eq. (3-6)

The statistical distribution of all calculated responses of interest due to the distribution of the collection of J data vectors is represented by the response variance given as

$$V = \frac{1}{J} \sum_{j=1}^{J} (R_j - R)^2 , \tag{A.1.1}$$

where

R_j = calculated response,

R = desired or expected response.

Expanding R_j in a first-order Taylor series about the expected response R, yields

$$R_j = R + \int_r \int_t \frac{\partial R}{\partial \xi_j(r,t)} \, \delta \xi_j(r,t) \, dr dt , \tag{A.1.2}$$

where

$\xi_j(r,t)$ = input parameter data field vector for all $j = 1, 2, ..., J$.

Substituting Eq. (A.1.2) into Eq. (A.1.1) results in

$$V = \frac{1}{J} \sum_{j=1}^{J} \left(\int_r \int_t \frac{\partial R}{\partial \xi_j(r,t)} \, \delta \xi_j(r,t) \, dr dt \right)^2 . \tag{A.1.3}$$

Eq. (A.1.3) can be put in another form

$$V = \int_r \int_t \left(\frac{1}{J} \sum_{j=1}^{J} \left(\frac{\partial R}{\partial \xi_j} \right)^{tr} D \, D^{-1} \, \delta \xi_j \right)^2 dr dt , \tag{A.1.4}$$

where D is a diagonal matrix of the components of the data vector $\xi(r,t)$.

Multiplying Eq. (A.1.4) by R^2 and then dividing it by R^2, yields

$$V = R^2 \int_r \int_t \left[\frac{1}{J} \sum_{j=1}^{J} \left(\frac{D}{R} \frac{\partial R}{\partial \xi} \right)^{tr} \left(D^{-1} \partial \xi_j \cdot \partial \xi_j^{tr} D^{-1} \right) \left(\frac{D}{R} \frac{\partial R}{\partial \xi} \right) \right] drdt \, . \qquad \text{(A.1.5)}$$

Finally, the response variance is defined to be

$$\sigma_R^2 = \frac{V}{R^2} = \int_r \int_t S_\xi^{tr}(r,t) \; U_x \; S_\xi \; (r,t) \; drdt \, , \qquad \text{(A.1.6)}$$

where

$$S_\xi(r,t) = \left(\frac{D}{R} \frac{\partial R}{\partial \xi} \right) \qquad = \text{total sensitivity vector, and}$$

$$U_x \quad = \quad (D^{-1} \, \delta\xi_j \bullet \delta\xi_j^{tr} \, D^{-1}) \qquad = \text{correlation matrix}$$

2. Derivation of Eqs. (6-4), (6-5), and (6-7)

Following the set of linear equations in Chapter III, i.e., Eq. (3-41), and using the nomenclature of neural network application, Eq. (3-41) can be written as

$$U_x^{-1}\,\Delta A + C_A^{tr}\,\lambda = 0\,, \tag{A.2.1.a}$$

$$C_Q^{tr}\,\lambda = 0\,, \tag{A.2.1b}$$

and

$$r + C_A\,\Delta A + C_Q \Delta Q = 0\,, \tag{A.2.1c}$$

where $C_A = (\partial C/\partial A)$ and $C_Q = (\partial C/\partial Q)$ are Jacobi matrices.

Solving Eq. (A.2.1a) for ΔA, yields

$$\Delta A = -U_x\,C_A^{tr}\,\lambda\,. \tag{A.2.2}$$

Substituting Eq. (A.2.2) into Eq. (A.2.1c), yields

$$r - C_A\,U_x\,C_A^{tr}\,\lambda + C_Q\,\Delta Q = 0\,. \tag{A.2.3}$$

Defining the uncertainty matrix of residuals as

$$U_r = C_A\,U_x\,C_A^{tr} \tag{A.2.4}$$

Thus, Eq. (A.2.3) can be rewritten as

$$r - U_r\lambda + C_Q\Delta Q = 0\,. \tag{A.2.5}$$

Eqs. (A.2.1b) and (A.2.5) can be combined into a large symmetric system. That is

$$0\Delta Q - C_Q^{tr}\,\lambda = 0\,, \tag{A.2.6a}$$

$$-C_Q\Delta Q + U_r\,\lambda = r\,, \tag{A.2.6b}$$

or in matrix form as

$$\begin{pmatrix} 0 & -C_Q^{tr} \\ -C_Q & U_r \end{pmatrix} \begin{pmatrix} \Delta Q \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ r \end{pmatrix}. \tag{A.2.6c}$$

To convert this system to least squares fitting algorithm, the matrix U_r need to be diagonalized to obtain a system of uncorrelated residuals. Using Choleski decomposition, U_r can be factored into

$$U_r = U^{tr}\,U\,, \tag{A.2.7}$$

where U is an upper triangular matrix. Applying the transformation

$$T = \begin{pmatrix} I & 0 \\ 0 & U^{-1} \end{pmatrix}, \tag{A.2.7}$$

where I is the identity matrix. Multiplying the system of Eq. (A.2.6c) by T^{tr}, yields

$$T^{tr}\begin{pmatrix} 0 & -C_Q \\ -C_Q & U_r \end{pmatrix} TT^{-1}\begin{pmatrix} \Delta Q \\ \lambda \end{pmatrix} = T^{tr}\begin{pmatrix} 0 \\ r \end{pmatrix}. \tag{A.2.9}$$

Defining

$$(U^{tr})^{-1} = U^{*}\,, \tag{A.2.10a}$$

$$-U^{*}\, C_Q = G\,, \tag{A.2.10b}$$

$$U\lambda = \mu\,, \tag{A.2.10c}$$

$$U^{*}\, r = \rho\,, \tag{A.2.10d}$$

we have from Eq (A.2.9) the following system

$$\begin{pmatrix} 0 & G^{tr} \\ G & I \end{pmatrix} \begin{pmatrix} \Delta Q \\ \mu \end{pmatrix} = \begin{pmatrix} 0 \\ \rho \end{pmatrix}, \tag{A.2.11}$$

that is

$$G^{tr}\, \mu = 0\,, \tag{A.2.12}$$

$$G\Delta Q + \mu = \rho\,. \tag{A.2.13}$$

Multiplying Eq. (A.2.13) with G^{tr} and using Eq. (A.2.12) leads to the following set of equations

$$G^{tr}\, G\, \Delta Q = G^{tr}\, \rho \tag{A.2.14a}$$

or

$$C_Q^{tr}\, U_r^{-1}\, C_Q\, \Delta Q = -C_Q^{tr}\, U_r^{-1}\, r\,, \tag{A.2.14b}$$

or simply

$$\Delta Q = -\left(C_Q^{tr}\, U_r^{-1}\, C_Q\right)\left(C_Q^{tr}\, U_r^{-1}\, r\right). \tag{A.2.14c}$$

This system determines the values of ΔQ.

Going back to Eq. (A.2.5) and solving for λ, yields

$$\lambda = -U_r^{-1} \left(r + C_Q \, \Delta Q \right) . \tag{A.2.15}$$

Substituting Eq. (A.2.15) into Eq. (A.2.2) for λ, yields

$$\Delta A = -\left(U_x \, C_A^{tr} \, U_r^{-1} \right) \left(r + C_Q \, \Delta Q \right) . \tag{A.2.16}$$

This system determines the values of ΔA.

Finally, the uncertainty matrix of the adjusted data is decreased from the original uncertainty matrix, U_x, by the uncertainty matrix of the adjustments, $U_{\Delta A}$. That is

$$U_R = U_x - U_{\Delta A} , \tag{A.2.17a}$$

$$= U_x - U_{Ar} \, U_r^{-1} \, U_{Ar}^{tr} + U_{Ar} \, U_r^{-1} \, C_Q \, U_w^{-1} \, C_Q^{tr} \, U_r^{-1} \, U_{Ar}^{tr} , \tag{A.2.17b}$$

where

$U_{Ar} = U_x \, C_A^{tr}$, and

$U_w = C_Q^{tr} \, U_r^{-1} \, C_Q$ = uncertainty matrix of weight vector parameters.

APPENDIX B

MODEL DATA PARAMETERS OF SAMPLE PROBLEMS

INPUT DATA, WEIGHT PARAMETERS, AND MODEL OUTPUT FOR RNA MODEL

```
(   .582E+00   .307E+00   .532E-01   .478E-01   .180E-02  -.338E-03   .114E-01
-.192E-02  -.488E+00 )
(   .395E+00   .429E+00   .474E-01   .106E+00   .938E-02   .175E-01   .776E-02
-.103E-01  -.228E+00 )
(   .595E+00   .300E+00   .661E-01   .381E-01   .555E-02   .463E-02   .323E-02
-.107E-01  -.465E+00 )
(   .494E+00   .349E+00   .284E-01   .998E-01   .215E-01   .343E-02   .892E-02
-.399E-02  -.268E+00 )
(   .410E+00   .421E+00   .585E-01   .851E-01   .178E-01   .756E-02   .280E-02
-.184E-02  -.385E+00 )
(   .351E+00   .447E+00   .623E-01   .129E+00  -.191E-02   .170E-01   .610E-02
-.897E-02  -.260E+00 )
(  -.950E-01   .735E+00   .694E-02   .333E+00   .121E-01   .780E-02  -.318E-01
 .268E-01   .291E-02 )
(   .281E+00   .483E+00   .638E-01   .137E+00   .244E-01   .421E-02  -.570E-02
 .106E-01  -.180E+00 )
(   .246E+00   .501E+00   .509E-01   .163E+00   .138E-01   .647E-02   .469E-02
 .118E-01  -.143E+00 )
(   .288E+00   .478E+00   .382E-01   .174E+00  -.533E-02   .172E-01   .527E-02
 .293E-02  -.200E+00 )
(   .338E+00   .433E+00   .627E-01   .131E+00   .156E-01   .119E-01  -.540E-02
 .109E-01  -.203E+00 )
(   .362E+00   .452E+00   .104E+00   .466E-01   .211E-01   .455E-02   .886E-02
 .130E-02  -.308E+00 )
(   .362E+00   .399E+00   .121E+00   .583E-01   .248E-01   .146E-01   .141E-01
 .512E-02  -.311E+00 )
(   .452E+00   .367E+00   .115E+00   .430E-01   .266E-01  -.312E-02  -.585E-02
 .476E-02  -.470E+00 )
(   .447E+00   .341E+00   .119E+00   .511E-01   .190E-01   .998E-02   .208E-02
 .810E-02  -.608E+00 )
(   .446E+00   .340E+00   .120E+00   .480E-01   .245E-01   .826E-02   .473E-02
 .706E-02  -.755E+00 )
(   .445E+00   .332E+00   .116E+00   .535E-01   .280E-01   .117E-01   .102E-01
 .320E-02  -.689E+00 )
(   .452E+00   .326E+00   .123E+00   .623E-01   .231E-01   .761E-02   .318E-02
 .258E-02  -.624E+00 )
(   .448E+00   .334E+00   .119E+00   .559E-01   .248E-01   .133E-01   .511E-02
 .494E-03  -.727E+00 )
(   .453E+00   .347E+00   .115E+00   .506E-01   .189E-01   .743E-02   .483E-02
 .315E-02  -.698E+00 )
(   .290E+00   .355E+00   .160E+00   .745E-01   .586E-01   .237E-01   .292E-01
 .833E-02  -.465E+00 )
(   .256E+00   .379E+00   .174E+00   .992E-01   .462E-01   .214E-01   .203E-01
 .336E-02  -.349E+00 )
(   .278E+00   .390E+00   .205E+00   .672E-01   .711E-02   .485E-01   .203E-01
-.126E-01  -.214E+00 )
(   .271E+00   .418E+00   .166E+00   .566E-01   .386E-01   .446E-01   .122E-01
-.593E-02  -.248E+00 )
(   .267E+00   .366E+00   .178E+00   .947E-01   .492E-01   .169E-01   .198E-01
 .714E-02  -.567E+00 )
(   .288E+00   .353E+00   .169E+00   .891E-01   .470E-01   .245E-01   .212E-01
 .761E-02  -.635E+00 )
```

```
(   .247E+00   .349E+00   .168E+00   .102E+00   .356E-01   .415E-01   .602E-02
 .101E-01   .119E-01   .821E-02   .769E-02   .252E-02   .283E-02  -.522E-02   .3
34E-02   .200E-02   .977E-02  -.172E-02  -.488E+00 )
(   .218E+00   .310E+00   .211E+00   .924E-01   .432E-01   .374E-01   .349E-01
 .227E-01  -.841E-02   .940E-02  -.213E-01   .372E-01   .331E-02   .240E-01   .1
55E-02  -.229E-02  -.162E-01   .244E-02  -.228E+00 )
(   .273E+00   .308E+00   .190E+00   .855E-01   .484E-01   .294E-01   .170E-01
 .740E-02   .152E-01   .100E-01   .362E-02   .133E-01   .529E-02  -.401E-02   .4
44E-02   .726E-02  -.281E-02  -.972E-02  -.465E+00 )
(   .354E+00   .338E+00   .164E+00   .784E-01   .381E-01   .224E-01   .824E-02
 .529E-02   .122E-01  -.993E-03  -.924E-02   .155E-01  -.235E-02   .126E-01  -.2
49E-02  -.207E-02  -.795E-02  -.210E-01  -.268E+00 )
(   .217E+00   .256E+00   .239E+00   .117E+00   .527E-01   .427E-01   .213E-01
 .215E-01   .515E-02   .822E-02  -.852E-03   .233E-01  -.361E-02  -.601E-03   .3
11E-03   .170E-02   .429E-02  -.483E-02  -.385E+00 )
(   .203E+00   .295E+00   .208E+00   .147E+00   .361E-01   .389E-01   .428E-01
-.331E-02   .141E-01  -.808E-02  -.362E-02   .199E-01   .150E-01  -.136E-02   .3
87E-02   .505E-02  -.644E-02  -.523E-02  -.260E+00 )
(  -.950E+00   .386E+00   .610E+00   .164E+01   .545E-01  -.374E+00   .219E+01
 .305E+00  -.913E-01  -.247E+00  -.253E+00  -.908E+00   .598E+00  -.570E+00   .8
26E+00  -.758E+00  -.152E+01   .439E-01   .291E-02 )
(   .177E+00   .324E+00   .219E+00   .122E+00   .323E-01   .404E-01   .167E-01
 .200E-01   .272E-01   .213E-02   .414E-02  -.216E-01   .250E-01   .991E-02  -.1
93E-01  -.816E-03   .130E-01   .713E-02  -.180E+00 )
(   .211E+00   .259E+00   .260E+00   .523E-01   .778E-01   .244E-01   .201E-01
-.537E-02  -.141E-02   .351E-01   .256E-01   .627E-01  -.207E-01  -.264E-01   .5
62E-02   .672E-02  -.241E-01   .331E-01  -.143E+00 )
(   .165E+00   .306E+00   .215E+00   .119E+00   .610E-01   .614E-01  -.852E-03
 .694E-02   .133E-01   .117E-01  -.179E-01   .327E-01   .167E-01   .412E-02  -.4
58E-02   .245E-02  -.518E-02   .121E-01  -.200E+00 )
(   .174E+00   .280E+00   .222E+00   .106E+00   .553E-01   .724E-01  -.428E-03
 .240E-01   .218E-02   .783E-02   .151E-01   .311E-01  -.795E-02   .345E-02   .2
59E-01  -.347E-01   .135E-01   .883E-02  -.203E+00 )
(   .151E+00   .253E+00   .220E+00   .134E+00   .926E-01   .368E-01   .600E-01
 .642E-02   .885E-02  -.308E-02   .146E-01  -.417E-02   .366E-02  -.136E-01   .2
39E-01   .140E-01  -.556E-02   .647E-02  -.308E+00 )
(   .875E-01   .286E+00   .236E+00   .157E+00   .720E-01   .603E-01   .424E-01
 .205E-01   .242E-01  -.229E-02   .158E-01   .234E-02   .107E-01  -.344E-02   .4
43E-02  -.246E-01   .233E-01  -.112E-01  -.311E+00 )
(   .162E+00   .249E+00   .217E+00   .132E+00   .633E-01   .739E-01   .276E-01
 .333E-01   .147E-01   .376E-02   .325E-01  -.780E-02   .594E-03  -.406E-02  -.1
96E-02   .100E-01  -.654E-02   .513E-04  -.470E+00 )
(   .171E+00   .232E+00   .236E+00   .126E+00   .527E-01   .620E-01   .473E-01
 .216E-01   .821E-02   .112E-01   .979E-02   .277E-02   .797E-02   .748E-02  -.2
71E-02  -.173E-02   .721E-02   .289E-03  -.608E+00 )
(   .157E+00   .250E+00   .229E+00   .114E+00   .656E-01   .599E-01   .392E-01
 .221E-01   .168E-01   .994E-02   .113E-01   .469E-02   .817E-02   .608E-02   .9
71E-03   .290E-02   .411E-02  -.839E-03  -.755E+00 )
(   .143E+00   .277E+00   .201E+00   .141E+00   .758E-01   .625E-01   .315E-01
 .293E-01   .165E-01   .796E-02   .532E-02   .467E-02   .233E-02   .958E-03   .4
59E-02   .219E-02  -.912E-02   .293E-02  -.689E+00 )
(   .158E+00   .285E+00   .206E+00   .122E+00   .821E-01   .474E-01   .413E-01
 .191E-01   .114E-01   .630E-02   .125E-01   .383E-02   .252E-02   .128E-02   .2
53E-02  -.415E-02   .110E-02   .168E-02  -.624E+00 )
(   .157E+00   .265E+00   .204E+00   .122E+00   .894E-01   .542E-01   .351E-01
 .230E-01   .170E-01   .921E-02   .837E-02   .511E-02   .672E-02  -.256E-02   .2
81E-02  -.654E-03   .320E-02   .435E-03  -.727E+00 )
(   .142E+00   .268E+00   .227E+00   .130E+00   .620E-01   .593E-01   .312E-01
```

```
 .302E-01    .128E-01    .954E-02    .102E-01    .756E-02   -.593E-05    .133E-02    .3
73E-02   -.401E-04    .471E-02   -.412E-03   -.698E+00 )
(    .939E-01    .165E+00    .204E+00    .121E+00    .103E+00    .814E-01    .581E-01
 .412E-01    .271E-01    .283E-01    .495E-02    .178E-01    .644E-02    .825E-02    .1
57E-01    .493E-02    .155E-01    .313E-02   -.465E+00 )
(    .484E-01    .211E+00    .222E+00    .141E+00    .106E+00    .830E-01    .356E-01
 .524E-01    .373E-01    .942E-02    .285E-01    .126E-01    .649E-02    .655E-02   -.1
17E-01    .375E-02    .136E-01   -.592E-02   -.349E+00 )
(    .543E-01    .232E+00    .198E+00    .146E+00    .879E-01    .972E-01    .715E-01
 .239E-01    .195E-01    .445E-02    .707E-02    .839E-02   -.441E-02    .114E-01    .2
27E-01    .292E-01   -.391E-01    .262E-01   -.214E+00 )
(    .590E-01    .252E+00    .182E+00    .151E+00    .983E-01    .100E+00    .463E-01
 .232E-01    .198E-01    .240E-01   -.416E-02    .214E-01    .407E-03    .146E-01    .1
74E-01    .155E-01   -.408E-01    .175E-01   -.248E+00 )
(    .417E-01    .193E+00    .200E+00    .143E+00    .926E-01    .105E+00    .623E-01
 .322E-01    .341E-01    .290E-01    .203E-01    .826E-02    .538E-02    .746E-02    .2
19E-02    .183E-01    .400E-03    .320E-02   -.567E+00 )
(    .113E+00    .173E+00    .180E+00    .146E+00    .106E+00    .762E-01    .509E-01
 .342E-01    .397E-01    .193E-01    .179E-01    .132E-01    .212E-02    .150E-01    .2
01E-02    .111E-02    .128E-01   -.122E-02   -.635E+00 )
```

-0.642902
1.361118
1.022979
1.692582
0.579003
-2.517487
0.928771
-0.805857
0.768598
1.093238
1.011317
-0.442471
0.659066
-2.003797
1.306296
-1.282921
-1.202446
2.426941
4.966881
4.899883
0.517179
-0.420430
-2.946064
-0.030611
-1.477202
-0.838056
-0.980248
-7.342701
-3.377675
0.539176
3.815460
2.442983
1.275891
3.559356
1.840054
1.920741
3.957998
-2.081368
-4.646481
-1.492457
3.255080
2.955075
-1.160217
0.703113
-0.388975
1.108122
0.580277
-2.360415
5.112047
-2.435104
-1.608999
-0.247660
-0.570108
0.018476
0.532820
2.147652

-3.702103
-0.646079
-1.134801
0.655065
0.750294
0.999669
0.457093
-0.611723
0.126812
-0.271204
-0.779493
-0.661671
0.790107
1.281494
0.796109
0.509392
0.659533
1.287562
-0.393837
-5.008869
-3.644758
-2.674735
-2.940714
-1.983848
-2.522937
-2.332721
-2.397513
-1.947201
0.188583
0.760460
2.207224
3.023837
3.371582
2.272437
3.115202
3.912821
3.300253
0.443464
6.325577
-4.896277
0.058115
-0.282325
0.176744
0.038244
0.137954
0.602310
-0.661437
-1.059965
-0.265080
-1.214995
0.417882
-1.379616
-0.884392
0.026950
-0.854291
-0.134259
3.124037
3.157974
8.709432
-5.986748
-0.364643
-0.933671
-3.009505
-0.811155

```
Outputs for Input 1:   (  0.464833  )
Outputs for Input 2:   (  0.456628  )
Outputs for Input 3:   (  0.441819  )
Outputs for Input 4:   (  0.365411  )
Outputs for Input 5:   (  0.466831  )
Outputs for Input 6:   (  0.472935  )
Outputs for Input 7:   (  0.906115  )
Outputs for Input 8:   (  0.482257  )
Outputs for Input 9:   (  0.432683  )
Outputs for Input 10:  (  0.502036  )
Outputs for Input 11:  (  0.488662  )
Outputs for Input 12:  (  0.525149  )
Outputs for Input 13:  (  0.563755  )
Outputs for Input 14:  (  0.522413  )
Outputs for Input 15:  (  0.533157  )
Outputs for Input 16:  (  0.553410  )
Outputs for Input 17:  (  0.559249  )
Outputs for Input 18:  (  0.540658  )
Outputs for Input 19:  (  0.553591  )
Outputs for Input 20:  (  0.554326  )
Outputs for Input 21:  (  0.619606  )
Outputs for Input 22:  (  0.619985  )
Outputs for Input 23:  (  0.623851  )
Outputs for Input 24:  (  0.619448  )
Outputs for Input 25:  (  0.664160  )
Outputs for Input 26:  (  0.614439  )
```

INPUT DATA, WEIGHT PARAMETERS, AND MODEL OUTPUT FOR IN-CORE FUEL MANAGEMENT MODEL

(1.136364 1.315789 1.136364 1.428571 1.315789 1.428571 1.315789 1.428571 1.3157
89 1.428571 1.136364 1.428571 1.315789 1.136364 1.428571 1.136364 1.428571 1.315
789 1.136364 1.315789 1.428571 1.315789 1.136364 1.315789 1.315789 1.428571 1.13
6364 1.315789 1.428571 1.428571 1.428571
0.222040 0.251300 0.228620 0.252840 0.235480 0.197260 0.122080 0.080500 0.26502
0 0.260820 0.215180 0.225960 0.166320 0.099400 0.072240 0.216160 0.223300 0.1789
20 0.119980 0.091000 0.062300 0.178920 0.122500 0.103460 0.076020 0.047180 0.091
000 0.083440 0.062020 0.070140 0.045220)
(1.136364 1.428571 1.315789 1.428571 1.136364 1.315789 1.428571 1.315789 1.1363
64 1.428571 1.136364 1.428571 1.315789 1.136364 1.428571 1.315789 1.428571 1.315
789 1.136364 1.315789 1.428571 1.136364 1.136364 1.315789 1.315789 1.428571 1.31
5789 1.315789 1.428571 1.428571 1.428571
0.288820 0.326900 0.314300 0.266840 0.182700 0.146720 0.106260 0.062160 0.28322
0 0.311220 0.226240 0.202720 0.139720 0.084980 0.059780 0.283080 0.235760 0.1710
80 0.108500 0.080500 0.053760 0.156100 0.116900 0.097580 0.070000 0.042140 0.103
600 0.084840 0.060200 0.070280 0.044520)
(1.428571 1.315789 1.136364 1.428571 1.136364 1.315789 1.428571 1.315789 1.3157
89 1.136364 1.428571 1.315789 1.428571 1.136364 1.136364 1.428571 1.428571 1.315
789 1.428571 1.315789 1.136364 1.136364 1.428571 1.315789 1.315789 1.136364 1.31
5789 1.315789 1.428571 1.428571 1.428571
0.204960 0.185360 0.165760 0.222320 0.170100 0.158620 0.108220 0.052220 0.18228
0 0.177240 0.232260 0.197540 0.172620 0.091840 0.042560 0.230720 0.238420 0.2045
40 0.173180 0.098980 0.040320 0.190400 0.194460 0.146580 0.088340 0.034020 0.160
160 0.120680 0.078960 0.095340 0.058240)
(1.428571 1.315789 1.428571 1.136364 1.428571 1.136364 1.315789 1.315789 1.3157
89 1.136364 1.428571 1.136364 1.428571 1.136364 1.315789 1.428571 1.428571 1.136
364 1.428571 1.315789 1.315789 1.136364 1.428571 1.315789 1.315789 1.136364 1.31
5789 1.315789 1.428571 1.428571 1.428571
0.303380 0.270200 0.252140 0.196140 0.177940 0.124040 0.086100 0.050400 0.24864
0 0.217420 0.225960 0.157640 0.145880 0.082880 0.049280 0.254240 0.233380 0.1611
40 0.153860 0.095620 0.049000 0.184800 0.181580 0.141540 0.089040 0.036960 0.157
360 0.122780 0.082320 0.100240 0.062160)
(1.428571 1.315789 1.428571 1.136364 1.428571 1.136364 1.428571 1.315789 1.3157
89 1.136364 1.428571 1.136364 1.428571 1.136364 1.315789 1.428571 1.428571 1.136
364 1.428571 1.315789 1.315789 1.136364 1.428571 1.315789 1.136364 1.315789 1.31
5789 1.428571 1.315789 1.428571 1.315789
0.300720 0.268100 0.250740 0.195860 0.179200 0.127820 0.098420 0.055020 0.24682
0 0.216440 0.225680 0.158200 0.147700 0.085540 0.051100 0.253540 0.233380 0.1617
00 0.153300 0.091840 0.048580 0.185920 0.184380 0.141400 0.073780 0.037520 0.165
900 0.135800 0.072660 0.104300 0.054880)
(1.136364 1.136364 1.136364 1.428571 1.428571 1.428571 1.428571 1.315789 1.1363
64 1.136364 1.428571 1.315789 1.428571 1.428571 1.315789 1.136364 1.136364 1.315
789 1.428571 1.315789 1.315789 1.136364 1.428571 1.315789 1.428571 1.315789 1.31
5789 1.428571 1.315789 1.428571 1.315789
0.050120 0.058100 0.091840 0.184100 0.237160 0.254940 0.205240 0.110040 0.06370
0 0.089740 0.166320 0.206920 0.236320 0.189980 0.102340 0.091560 0.122500 0.1772
40 0.200060 0.149240 0.082600 0.128520 0.172340 0.160860 0.122780 0.062440 0.158
200 0.139720 0.086940 0.105420 0.058240)
(1.136364 1.136364 1.136364 1.428571 1.315789 1.428571 1.428571 1.315789 1.1363
64 1.136364 1.315789 1.428571 1.428571 1.428571 1.315789 1.136364 1.136364 1.315
789 1.428571 1.315789 1.428571 1.136364 1.428571 1.315789 1.428571 1.315789 1.31
5789 1.428571 1.315789 1.428571 1.315789
0.044520 0.051520 0.081200 0.162400 0.215740 0.252420 0.208460 0.114100 0.05642
0 0.079520 0.144900 0.216440 0.241360 0.195440 0.108220 0.084420 0.116340 0.1800
40 0.204960 0.155960 0.095200 0.126560 0.173880 0.164220 0.127260 0.067620 0.160
160 0.142100 0.089040 0.107240 0.059360)
(1.219512 1.333333 1.333333 1.219512 1.666667 1.219512 1.666667 1.666667 1.6666
67 1.333333 1.333333 1.219512 1.666667 1.219512 1.666667 1.219512 1.219512 1.666

667 1.333333 1.666667 1.219512 1.333333 1.333333 1.666667 1.219512 1.333333 1.21
9512 1.666667 1.333333 1.333333 1.219512
 0.242760 0.261100 0.233380 0.192780 0.192920 0.153720 0.125160 0.079800 0.27146
0 0.228760 0.197540 0.175000 0.165760 0.111020 0.072240 0.188580 0.175420 0.1871
80 0.158480 0.112700 0.056280 0.175000 0.165480 0.146300 0.084420 0.043260 0.132
860 0.116900 0.066920 0.078680 0.041020)
(1.219512 1.219512 1.219512 1.219512 1.666667 1.333333 1.666667 1.666667 1.2195
12 1.219512 1.219512 1.219512 1.666667 1.666667 1.666667 1.219512 1.219512 1.666
667 1.333333 1.666667 1.333333 1.333333 1.333333 1.666667 1.333333 1.333333 1.33
3333 1.666667 1.333333 1.333333 1.219512
 0.027160 0.031220 0.047600 0.089460 0.191940 0.246540 0.308560 0.220640 0.03486
0 0.049420 0.084140 0.156940 0.263620 0.289240 0.196560 0.061880 0.093800 0.1731
80 0.197120 0.213080 0.118580 0.109900 0.140140 0.176400 0.123200 0.067900 0.130
900 0.136080 0.076440 0.081340 0.040880)
(1.219512 1.666667 1.219512 1.219512 1.219512 1.333333 1.219512 1.666667 1.6666
67 1.666667 1.219512 1.666667 1.219512 1.666667 1.666667 1.333333 1.219512 1.219
512 1.333333 1.666667 1.333333 1.219512 1.333333 1.666667 1.333333 1.666667 1.33
3333 1.333333 1.333333 1.333333 1.219512
 0.593600 0.688800 0.396060 0.205240 0.135800 0.095200 0.076300 0.064120 0.68432
0 0.465780 0.214060 0.161140 0.095620 0.093100 0.063560 0.291620 0.153020 0.1044
40 0.085120 0.080780 0.044380 0.100940 0.080780 0.077700 0.050960 0.033460 0.059
360 0.045080 0.028000 0.027020 0.014280)
(1.219512 1.666667 1.219512 1.219512 1.219512 1.333333 1.219512 1.666667 1.6666
67 1.666667 1.219512 1.666667 1.219512 1.666667 1.666667 1.333333 1.219512 1.219
512 1.333333 1.666667 1.333333 1.219512 1.333333 1.666667 1.333333 1.666667 1.33
3333 1.333333 1.333333 1.333333 1.219512
 0.559580 0.582680 0.407120 0.267260 0.186760 0.112280 0.085960 0.058940 0.56224
0 0.417900 0.259140 0.195440 0.117880 0.091980 0.055440 0.267820 0.190680 0.1345
40 0.086380 0.070280 0.035000 0.132580 0.085260 0.068600 0.039760 0.024920 0.052
360 0.035000 0.020300 0.019600 0.011480)
(1.219512 1.666667 1.219512 1.219512 1.219512 1.333333 1.219512 1.666667 1.6666
67 1.666667 1.219512 1.666667 1.219512 1.666667 1.666667 1.333333 1.219512 1.219
512 1.333333 1.666667 1.333333 1.219512 1.333333 1.666667 1.333333 1.666667 1.33
3333 1.333333 1.333333 1.333333 1.219512
 0.593600 0.688800 0.396060 0.205240 0.135800 0.095200 0.076300 0.064120 0.68432
0 0.465780 0.214060 0.161140 0.095620 0.093100 0.063560 0.291620 0.153020 0.1044
40 0.085120 0.080780 0.044380 0.100940 0.080780 0.077700 0.050960 0.033460 0.059
360 0.045080 0.028000 0.027020 0.014280)
(1.219512 1.333333 1.219512 1.219512 1.219512 1.333333 1.219512 1.666667 1.3333
33 1.666667 1.219512 1.666667 1.219512 1.666667 1.666667 1.333333 1.219512 1.219
512 1.333333 1.666667 1.666667 1.219512 1.333333 1.666667 1.333333 1.666667 1.33
3333 1.333333 1.666667 1.333333 1.219512
 0.121660 0.134820 0.133280 0.117180 0.137900 0.152600 0.172480 0.175000 0.14896
0 0.168420 0.126980 0.169400 0.158900 0.221200 0.182420 0.135800 0.109480 0.1261
40 0.160720 0.208460 0.161420 0.103600 0.122780 0.162540 0.136080 0.109340 0.110
740 0.105140 0.093800 0.069300 0.043400)
(1.219512 1.333333 1.219512 1.219512 1.219512 1.333333 1.219512 1.666667 1.3333
33 1.666667 1.219512 1.666667 1.219512 1.666667 1.666667 1.333333 1.219512 1.219
512 1.333333 1.666667 1.666667 1.219512 1.333333 1.666667 1.333333 1.666667 1.33
3333 1.333333 1.666667 1.333333 1.219512
 0.121660 0.134820 0.133280 0.117180 0.137900 0.152600 0.172480 0.175000 0.14896
0 0.168420 0.126980 0.169400 0.158900 0.221200 0.182420 0.135800 0.109480 0.1261
40 0.160720 0.208460 0.161420 0.103600 0.122780 0.162540 0.136080 0.109340 0.110
740 0.105140 0.093800 0.069300 0.043400)
(1.219512 1.333333 1.219512 1.219512 1.219512 1.333333 1.219512 1.666667 1.3333
33 1.666667 1.219512 1.666667 1.219512 1.333333 1.666667 1.333333 1.219512 1.219
512 1.333333 1.666667 1.666667 1.219512 1.333333 1.666667 1.333333 1.666667 1.33
3333 1.333333 1.666667 1.333333 1.219512
 0.165900 0.183260 0.175980 0.141960 0.145880 0.133560 0.120680 0.128520 0.20174
0 0.223160 0.154280 0.180880 0.137760 0.137900 0.132720 0.176120 0.132300 0.1369
20 0.153580 0.173600 0.133700 0.121380 0.136500 0.169400 0.130060 0.099260 0.123
760 0.114520 0.098140 0.076440 0.047180)

0.006219
-0.277148
-0.111815
0.203755
-0.120606
0.101124
0.139169
-0.208793
-0.449587
-0.175588
-0.073410
-0.126438
0.120251
-0.066825
-0.156948
-0.012383
0.090757
0.137840
-0.103617
-0.192998
0.100447
-0.105489
-0.127986
-0.117704
0.022683
-0.080005
-0.025765
-0.055031
0.256091
0.062549
0.150090
-0.238848
3.382089
0.150121
-0.383005
-0.664271
-0.223363
-0.952091
0.199109
2.748838
1.227172
-0.804419
0.657088
-0.932362
0.782029
0.643208
0.084212
0.098164
-0.687901
-0.774562
0.230641

-2.128252
-0.140912
-0.651663
0.111343
0.001083
0.705842
-0.067435
-0.796356
-2.310412
-0.245424
-0.041895
0.045786
1.047770
0.677752
-0.648892
-0.091000
-0.589323
-0.329973
-0.019857
0.705779
0.357472
-0.161513
-0.150813
-0.256055
0.012235
0.363768
0.196506
0.124677
-0.447078
-0.269442
0.147795
-0.481106
-0.229024
-0.237437
0.153880
-0.164482
0.059294
0.054469
-0.249400
-0.612185
-0.216841
-0.035802
-0.741445
-3.695652
-1.577791
0.347656
0.627546
0.705618
0.539058
0.397098
-2.817461
-1.095741
-0.051460
-0.081526
0.708956
2.495193
-0.058289
-1.199164
-1.409426
1.060630
0.342675
0.351893
1.741476
0.197195
0.408052
0.306363

0.266385
-0.331714
-0.229478
0.777331
1.546793
-0.437728
-0.796604
0.143725
-1.566138
-0.875387
0.515023
0.253623
0.470124
0.811348
0.059629
-0.722339
-0.942783
0.540229
-0.333209
0.622480
0.334946
-0.578329
-0.343972
-0.421354
0.762637
0.579381
-0.007999
-0.124357
0.107177
0.578603
-0.022783
-0.136653
-0.589784
0.000278
0.588422
0.243610
-0.084912
-0.398415
0.097556
0.958043
0.303783
0.730705
-0.419288
0.091070
0.467104
-0.512781
0.479094
-0.308391
0.110942
-0.062065
0.128721
-1.197061
-0.418879
0.373363
0.711956
0.084762
-0.082786
-0.573021
-0.872765
-0.090612
0.021902
-0.552549
0.096026
-0.302736
0.080460
-0.118896

-0.662353
0.489424
0.613515
0.007233
-1.329412
-0.402195
-0.150849
0.550689
0.065869
0.596992
0.082971
-1.159251
-0.858963
0.153341
-0.650444
0.629984
0.781761
-0.231629
-0.253547
-0.396017
0.507167
0.315112
0.052113
0.201180
0.190171
0.307651
-0.018290
-0.226750
-0.337028
-0.121043
0.607966
0.109271
-0.089884
-0.397741
0.757915
-0.115745
0.445485
-0.022992
-0.116254
-0.234124
0.245311
-0.605780
0.376081
-0.811343
1.071075
-0.243722
0.242218
-0.959487
-0.771461
0.525938
-0.070748
-0.431950
1.076119
-0.357417
-0.374891
-0.403378
1.122822
-0.277217
0.051634
-0.418569
0.375214
0.197308
-0.132862
0.215651
0.128011
-0.700972

-3.404886
-1.337877
-0.141629
0.548780
0.589189
0.209644
1.104777
-2.270652
-0.038334
-0.020402
0.648841
0.225584
2.827379
0.675567
-1.001771
-2.045699
0.822559
0.754246
1.244155
2.407833
0.091880
0.567676
1.278492
0.202260
0.475923
0.113120
0.714800
1.985734
-0.693936
-1.388798
-0.350767
-2.290042
-0.851901
0.031336
0.239175
0.354367
0.304920
0.281955
-1.870752
-0.420994
-0.060521
-0.045534
0.126021
1.265041
0.033581
-0.566570
-0.954880
0.421303
0.202922
0.218035
1.250115
-0.011704
0.208451
0.341602
0.110624
-0.060606
-0.194625
0.291050
1.116378
-0.384983
-0.650076
0.372457
-4.348724
-1.692276
1.020920
1.089237

1.213944
1.480104
0.006499
-2.771130
-2.115432
0.946101
-0.712524
1.321641
0.304593
-1.372577
-0.951594
-0.417843
1.375840
0.992478
-0.317132
1.532639
0.688544
1.046268
-0.338543
-0.146403
-0.804917
-0.331863
1.340168
1.852938
0.157779
-0.273182
-1.104594
-4.930356
-1.652903
-0.928549
0.229938
0.274261
-0.259867
1.210617
-3.948893
-0.190867
-0.463978
0.606470
-0.149285
3.392603
0.889077
-1.232204
-2.662086
0.647442
0.408323
1.481953
3.457588
-0.064989
0.401561
1.392123
-0.009260
0.476368
-0.202305
0.511012
2.731400
-1.181825
-2.078845
-0.187214
-2.320087
-1.410781
1.171915
0.205774
1.067774
0.478547
0.040867
-2.233328

-0.655124
-0.066741
0.072132
0.468404
1.128502
-0.610182
-0.780129
0.093534
0.882217
-0.089824
-0.328010
1.110102
0.485354
-0.047196
-0.396264
0.282613
-0.226456
-0.242015
0.287324
1.292637
0.156560
0.126407
0.313990
1.610229
0.810193
-0.068611
0.262760
-0.345358
0.488240
-0.193612
1.231326
-0.192239
-0.006392
-0.475952
0.469833
-0.831889
0.056128
0.421213
0.776274
0.085914
-0.191369
-0.204273
-1.208241
0.157403
-0.338369
-0.164718
-0.108905
-0.204038
0.071701
0.080873
-1.273284
0.237785
0.584510
-0.155428
-1.830607
-0.618952
0.529765
-0.582566
0.442773
0.043499
0.009977
-1.780459
-0.021426
-0.026677
0.607540
-0.057358

-0.304393
-0.323223
-0.177535
-0.295224
0.096367
0.019777
-0.040234
1.635250
-0.124175
0.144732
-0.073153
0.156289
0.203658
0.091348
0.045921
1.581251
-0.056232
-0.166020
-0.147605
0.594968
0.114819
-0.823047
-0.615549
-0.056993
-0.448823
0.577560
-0.947855
-0.611845
-0.993064
-0.743743
-0.630113
0.406542
-0.555761
-0.218861
0.710675
-0.099550
-0.451872
-0.508805
0.112234
-0.488691
-0.083856
-0.743662
-0.514086
-1.151931
-1.003995
-0.594530
0.517129
-0.607879
-0.063220
-0.496784
0.138242
0.090186
-0.460405
0.137246
-0.152839
-0.248158
-0.294353
-0.130450
-1.203733
-1.266292
-0.749916
0.306993
-0.062428
0.091918
0.071575
-0.342174

-0.103202
0.037636
0.253199
-0.235747
-0.209321
-0.312663
-0.199722
-0.337463
-1.248872
0.408575
-0.450697
0.245629
-0.086764
-0.347679
-0.428690
0.297970
-0.223429
-0.261490
0.063034
-0.484814
0.157643
0.145019
0.251423
-1.014997
0.244193
-0.557632
-0.196123
0.077434
-0.127337
-0.653304
0.301370
0.079685
-0.016136
0.110486
-0.675368
0.203008
0.009590
0.487029
-0.517066
0.608394
-0.889363
-0.219279
-0.200526
-0.249070
-0.671533
0.580357
0.073811
-0.098288
0.269101
-0.680419
0.356901
0.125576
0.419074
0.171169
0.376786
-0.858252
-0.860208
-0.267260
-0.401252
-0.480015
0.421600
-0.064765
-0.647669
0.006185
-0.898345
0.446137

0.085811
0.407678
0.555406
0.033980
-0.927190
-0.723048
-0.330371
0.934406
-0.070067
-0.823792
-0.403773
-0.275459
-0.679548
-0.342845
-0.623208
-0.449536
-0.527726
-0.845143
-0.519005
0.334636
-0.551473
-0.198950
0.067378
-0.115214
-0.492234
-0.330487
-0.020644
-0.567931
-0.542468
-0.247354
-0.297739
-0.799536
-0.915608
-0.348211
0.135060
0.051079
-0.054872
0.153782
-0.136930
-0.408052
0.130832
0.062101
-0.317513
0.332868
-0.468033
-0.169576
-0.212205
-0.762826
0.073323
-0.510604
0.160275
-0.071002
0.183644
-0.447526
-0.136186
0.141024
0.034820
-0.353899
-0.405792
-0.048518
-0.062157
0.249834
-0.565520
0.366092
-0.717598
0.345636

-0.084581
-0.134134
-0.665270
0.264971
0.102392
0.091754
0.206390
-0.306639
0.068048
0.106941
0.480106
-0.106295
0.240448
-0.759467
-0.468250
-0.198886
0.181944
-0.454735
0.571274
0.161818
-0.323479
0.003594
-0.200923
0.337929
0.280339
0.358683
0.592806
0.197376
-1.140539
-0.655959
-0.308156
-0.337293
-0.507447
0.438523
-0.105681
-0.606496
-0.011358
-0.754830
0.449180
0.075569
0.282009
0.635056
0.020477
-1.073003
-0.573150
0.152983
-0.135264
-0.244407
-0.599671
-0.269117
0.511084
-0.468236
0.299978
-0.602864
-0.470800
-0.777839
-0.426336
-0.222723
0.051333
0.014574
0.183819
-0.400758
-0.433885
-0.510509
-0.404295
0.525514

-0.210223
0.045460
-0.553328
-0.388959
-0.129434
-0.484415
0.114738
-0.031840
-0.233282
-0.030472
-0.310774
-0.729479
-0.316655
0.404471
0.202104
-0.088060
-0.547514
-0.107181
-0.180184
0.371147
-0.267284
0.075054
-0.436272
-0.307548
-0.252951
-0.201343
-0.500223
-0.158956
0.496018
-0.294512
-0.099132
0.348592
0.052680
-0.055047
0.523842
0.268155
-0.244282
-1.086316
-0.060946
-0.190948
-0.322896
-0.392615
0.078077
0.062212
-0.516186
0.009468
-0.045892
0.344876
0.128565
0.360597
0.689819
-0.219351
-1.082256
-0.258641
-0.284014
-0.734213
-0.459510
0.246338
-0.259804
-0.550303
-0.273466
-0.378737
0.369845
0.050579
0.092274
0.556488

-0.031675
-1.292768
0.008360
-0.179827
-0.618456
-0.545008
-0.497322
-0.052085
0.006669
-0.260986
-0.234379
-0.182879
-0.322326
0.354064
-0.406497
-0.121869
-0.369990
-0.358408
-0.158062
-0.403301
-0.563283
-0.360228
0.398765
-0.211653
-0.105387
0.546808
-0.160854
-0.118548
0.465301
0.040370
-0.446059
-0.948650
0.000188
-0.243703
-0.589576
-0.512897
-0.281898
0.392428
-0.498885
-0.066484
0.274717
0.178344
-0.024600
0.378229
0.411961
-0.476723
-0.968879
-0.068112
-0.138308
-0.602863
-0.690567
0.014976
0.140403
-0.448541
-0.280159
0.144870
0.232443
-0.048753
0.242052
0.313708
-0.177319
-1.250975
0.126987
-0.229556
-0.750369
-0.900700

0.113467
-0.244173
-0.528931
-0.385878
-0.266602
0.221994
0.006836
-0.015475
0.384488
-0.051710
-1.304821
0.069611
-0.064384
-0.702765
-0.621799
-0.100387
0.215055
-0.165770
-0.030369
0.733889
-0.144494
0.041074
0.327108
0.099363
-0.568350
-1.059375
-0.013359
-0.213974
-1.038837
-0.743841
-0.033171
0.177257
-0.307289
0.063747
0.383032
0.076805
-0.022902
0.331409
0.133731
-0.545793
-1.018049
-0.145773
-0.195151
-1.234676
-0.862684
-0.193253
-0.141383
-0.208857
-0.260905
0.255528
0.118335
-0.034104
0.178620
0.287815
-0.291817
-1.249209
0.056172
-0.190565
-1.243538
-1.033601
0.005935
-0.124286
-0.152201
-0.180223
0.479404
0.010108

-0.093822
0.294106
-0.096835
-0.325998
-1.103171
-0.094586
-0.173139
-1.334823
-1.048109
-0.035172
-0.084935
-0.030766
-0.383806
0.278739
-0.115465
-0.224351
0.175391
-0.171200
-0.232769
-1.297433
-0.078094
0.007482
-0.083640
-0.086855
-0.160409
-0.028873
0.146607
-0.073631
0.133688
-0.172003
-0.133930
0.083028
-0.467361
0.035269
0.105243
-0.016922
0.398957
0.527753
0.251146
-0.442518
-0.653676
-1.073532
-1.320551
-1.318585
0.454371
0.215142
-0.459678
-0.797922
-1.114803
-1.467773
-1.321244
-0.260962
-0.557732
-0.867176
-1.121606
-1.274380
-1.245679
-0.624957
-0.917453
-1.054912
-1.264574
-1.274731
-0.989360
-1.050270
-1.157936
-1.183867
-1.291658

INPUT DATA, WEIGHT PARAMETERS, AND MODEL OUTPUT FOR PET MODEL

(-0.02311 0.014996 0.16549 0.1959 0.09824 0.095731 0.11199 0.1052 0.0827
06 0.11831 0.098475 0.090739 0.789)
(0.011444 -0.00501 0.093396 0.09219 0.098205 0.01976 0.077813 0.099043 0.121
92 0.098115 0.071957 0.091914 0.666)
(-0.00552 -0.00736 0.12322 0.11129 0.097875 0.050623 0.086268 0.067796 0.0660
07 0.11182 0.091919 0.062965 0.575)
(0.012533 0.23691 0.24037 0.15823 0.14945 0.10011 0.086806 0.11351 0.102
13 0.11907 0.065805 0.12632 0.558)
(0.016015 0.14825 0.18155 0.094746 0.12393 0.11017 0.11474 0.10978 0.142
39 0.17286 0.10028 0.12014 0.838)
(0.020614 0.027217 0.18217 0.13722 0.060266 0.068648 0.083798 0.1239 0.0948
42 0.042461 0.07312 0.094323 0.542)
(0.005861 0.072821 0.12135 0.050569 0.096704 0.055302 0.15639 0.067239 0.0419
06 0.11936 0.076839 0.088267 0.566)
(0.013951 0.23076 0.26934 0.091647 0.11368 0.095883 0.12565 0.038708 0.0806
91 0.065816 0.033549 0.070064 0.244)
(0.045227 0.43562 0.21539 0.19508 0.092417 0.077218 0.14566 0.048063 0.04
35 0.061355 0.039723 0.049459 0.161)
(0.072199 0.31276 0.1854 0.088769 0.11911 0.042069 0.09698 0.11995 0.139
91 0.023717 0.067926 0.082613 0.358)
(0.0050230 0.0573930 0.2729400 0.1000100 0.1258800 0.2313900 0.1257800
0.1094700 0.1301200 0.1362600 0.1147300 0.1047300 0.2538)
(0.0022930 0.0807280 0.1040400 0.1251800 0.1569600 0.1342900 0.1492900
0.1352500 0.1309000 0.0887650 0.1276600 0.1308900 0.2986)
(0.0000340 0.0431810 0.2262400 0.1419800 0.1559400 0.1359500 0.0944600
0.1244100 0.1167700 0.1481600 0.1683000 0.1043300 0.2764)
(-0.0005800 -0.0029300 0.1707000 0.0667560 0.1455400 0.1445700 0.1481700
0.0694890 0.0966800 0.1010600 0.1168200 0.1170800 0.2258)
(0.0101630 0.0115940 0.1260900 0.1434800 0.0840090 0.1605600 0.0735540
0.1332600 0.1065400 0.1477500 0.0787560 0.0922040 0.2456)
(0.0043020 0.0242580 0.1112300 0.0850760 0.1005200 0.1294200 0.1101700
0.0998460 0.0816100 0.0875900 0.0971840 0.1221700 0.2087)
(-0.0032900 -0.0169000 0.1433500 0.1348800 0.0794160 0.0759190 0.0731530
0.1270900 0.0989890 0.0984980 0.0681890 0.1042900 0.1837)
(-0.0008400 0.0066810 0.2162300 0.1194700 0.1339400 0.1216900 0.0953030
0.1520000 0.1501200 0.0702560 0.0837550 0.1073900 0.2213)
(0.0021450 0.1504600 0.2560200 0.1136300 0.1276600 0.2091700 0.1184000
0.0696610 0.0975740 0.1304400 0.1426600 0.0825450 0.2128)
(-0.0023300 0.0407910 0.1144900 0.1206200 0.1209500 0.2009100 0.1314200
0.1070300 0.1817700 0.1125300 0.0859090 0.1527200 0.3174)

-0.023573
0.615020
-0.286914
0.077125
-0.486154
-0.055264
-0.122267
-0.831621
0.113817
-0.340678
-0.805455
-0.819326
0.483743
0.870212
0.617098
-0.458173
-0.329923
-0.006928
-1.036150
-0.881832
-1.218825
-1.518529
-1.307985
0.049971
0.351515
0.296982
-1.020904
-0.263772
0.754068
0.778905
1.056520
1.477756
1.097199
0.135267
0.326522
0.815566
-0.228846
-0.691078
0.316133
0.470145
0.080456
-1.077832
-0.368648
-0.628418
-0.386350
-0.059855
-1.059182
-1.150640
0.012444
-0.783793
0.627287
-1.105237
-0.577292
0.294553
-0.548522
0.067132
0.070371
-0.171495
-0.819388

-0.608769
-0.588328
-0.689968
0.496788
0.515700
0.636441
-0.207040
-1.130931
-0.417749
-0.302684
-0.405844
-0.061783
0.656201
0.749399
0.527067
0.419528
-0.010680
-0.079603
-1.117349
0.183305
-0.225839
-0.704790
0.314693
-0.606346
-0.586991
-0.011879
0.900671
0.865533
-0.554556
-0.100330
-0.593890
-0.039357
-0.909131
0.503016
-1.366282
-1.298825
-1.236948
0.308103
-1.006139
-0.762582
-0.724879
-0.067601
0.572649
-0.880322
-0.499330
0.609697
0.972786
0.900088
0.314263
0.081493
0.025315
-0.036756
-0.726816
-0.681105
-0.400337
-0.710776
-0.217484
-1.193214
-0.457490
-1.075973
-0.764025
-0.193847
0.190375
-0.390790
0.450235
-0.725029

-1.409304
0.237857
-0.011900
-0.099428
0.104627
-0.368293
-0.226463
0.357980
0.880576
0.845475
-0.424160
-0.839609
-0.716189
0.635433
-0.455462
-0.185394
-1.058614
0.538597
-0.644788
-0.109141
-0.325569
0.476543
-0.848206
-0.019508
-0.800071
0.758519
-0.601407
-0.133712
-0.442883
-0.637936
-1.119176
0.014392
-0.203588
0.608535
-0.877174
-0.337291
0.605110
0.764934
0.447206
-0.802902
-0.554232
0.396604
-0.720923
-0.632920
0.876993
0.009169
0.355288
-0.192074
-0.063090
0.066384
-0.667564
0.295015
1.039674
0.322689
-0.235935
-0.173856
-0.897084
-1.039862
-0.200194
0.796753
-0.185622
-0.242593
0.513647
1.002619
-0.514743
0.047486

-0.400597
0.112875
-0.046687
-0.791335
0.456116
0.810648
-0.580771
0.620544
0.312718
-0.173984
0.872080
-0.391038
-0.603860
0.832335
-0.229757
-0.584941
0.560926
0.730141
-0.872425
-0.483086
0.719042
-0.655909
0.613123
-0.638396
0.651324
-0.219235
0.279826
-0.511643
-0.805802
-1.022122
-1.457029
-0.122381
-0.428897
-1.260479
-0.202979
-0.910305
-0.795513
-0.112878
-1.012046
-0.225206
-0.880352
-1.156199
-0.907425
-0.979535
-0.076771
-1.094861
-1.312989
0.224863
0.522366
-0.577722
-1.838215
2.088627
-1.226538
-0.478515
-0.495474
-0.820223
-1.762128
0.750532
-1.564302
-0.562292
-0.811628
-0.782277
-0.038300
0.554203
0.613351
0.868376

0.696585
-1.613438
-1.335765
-0.648490
-0.126509
-0.618061
-0.687133
-0.721643
-0.976111
-0.837110
-0.244130
-0.674941
-0.336031
-0.831126
-0.667265
-0.619229
-0.859263
-0.826277
-0.699411
-0.695937
-0.733720
-0.117881
-0.283970
0.550356

-0.237775
-0.506808
-0.635839
0.057407
0.683502
0.822134
0.292130
-0.829275
-0.152960
0.110776
-0.031930
0.451977
0.927618
1.129525
2.203521
2.316883
1.527735
0.163411
-1.513634
-0.605708
-0.368567
-1.540687
0.332900
-1.174999
-1.300241
0.061524
2.322346
0.880966
-3.747646
-0.747875
-1.896701
-2.463448
-3.405471
-2.368600
-3.858805
-4.283789
-4.192776
0.411638
-0.718206
-0.869943
0.143464
0.409780
1.804333
0.537520
0.778444
2.041608
2.708208
2.633802
2.194851
0.135937
0.131085
-0.067018
-0.234708
-0.513450
-0.215299
-0.911916
-0.119648
-1.083606
-0.236783
-0.955199
-0.909158
-0.055183
0.396048
-0.521806
0.521107
-0.654400

-0.077968
0.075644
0.012950
0.272916
-0.619388
-0.497636
-0.746009
-1.191305
-0.196047
-0.658336
-1.143470
-1.267910
-0.311657
0.265988
1.698885
-0.389728
-0.565565
-0.615183
-2.801042
-1.660974
-2.468640
-2.357812
-2.839036
-1.221766
0.758585
0.408938
-1.432549
-0.765916
0.777043
0.805094
1.960107
1.831218
2.209538
0.547694
1.358153
1.806737
-0.055624
-0.618957
0.017081
0.709135
0.278861
-0.942115
-0.526920
-0.547808
-0.190634
0.177861
-0.896209
-1.277777
-0.178500
-1.689140
0.097466
-1.062248
-0.522778
0.675496
-0.492416
0.581011
0.151453
0.257397
-0.629707

-1.343541
0.156242
-0.165504
-0.034706
0.076290
-0.266184
-0.424779
0.379147
-0.439968
1.478482
-0.086109
-1.269597
-1.940333
-0.787224
-1.308870
-0.950355
-1.663247
-0.356501
-1.779572
-0.223744
-1.687400
-0.038180
-0.820654
-0.123999
-0.985949
0.380914
-0.494141
-0.149091
-0.420263
-0.770148
-1.249688
-0.102373
-0.955423
0.211105
-1.062730
-0.446634
0.599510
0.635581
0.527732
-0.919028
-0.471925
0.359116
-0.641303
-0.684633
0.495731
0.190124
0.114455
-0.452676
-0.543246
-0.381535
-1.136211
-0.084927
0.506035
-0.052751
-0.667305
-0.072821
-1.007905
-1.199381
-0.005945
0.978508
0.495355
0.438810
1.318380
1.513719
0.210336
0.791837

0.774554
0.021692
-0.482936
-1.146373
0.160210
0.664977
-0.423689
0.681539
0.364277
-0.298622
0.903944
-0.324511
-0.402292
0.613541
-0.847628
-1.067392
0.515403
0.677112
-0.151223
-0.001841
1.279292
-0.415281
1.067487
-0.089920
1.424340
-0.240168
-0.329593
-0.754856
-0.352817
-0.919348
-1.497531
-0.557927
-0.346378
-1.132575
-0.027239
-0.898670
-0.990172
-0.239725
-1.546454
-0.892242
-0.670672
-0.969984
-0.281877
-0.794743
0.589525
-0.858624
-0.637098
0.570172
0.944957
-1.028605
-1.782918
2.840168
-0.856376
0.224526
0.628382
-2.870072
-4.910017
3.133625
-0.632063
-0.179389
-1.985322
-0.799259
0.337061
-0.348422
1.258979
0.802207

1.302608
-0.635923
0.249113
-1.786907
1.231998
-2.166760
-1.870682
-1.962490
-2.226356
-2.159963
3.675834
-2.491839
-1.866900
-2.161566
-1.701452
-1.747328
-2.012321
-2.146996
-1.300005
-1.810441
-1.749029
-1.722445
-1.713583
4.333969

APPENDIX C

SOME GRESS/ADGEN UTILITY ROUTINES

Name: CLEARXX
CLEADXX

Function: To clear the forward matrix buffers.

Application: Adjoint

Arguments: NONE

How to use it: Generally, CALL CLEARXX should be inserted in the main program immediately before the STOP statement. CLEARXX must be the last executed call before ending the run. If the program exits at some point other than in the main program, it will be necessary to insert CALL CLEARXX at that point. Use CLEADXX if responses are double precision and you are using G_float option on the FORTRAN compiler.

Example:

1) Normal exit from a main program with a STOP statement

```
PROGRAM MAIN
  .
  .
CALL CLEARXX
STOP
END
```

2) Possible exit from subroutine and no STOP statement

```
PROGRAM MAIN
  .
  .
  .
CALL THIS
CALL CLEARXX
END
SUBROUTINE THIS
  .
  .
  .
IF(UNHAPPY) CALL CLEARXX
IF(UNHAPPY) STOP
RETURN
END
```

UTILITY ROUTINE

Name: POTRXX (X, 'CHAR')
POTDXX (X, 'CHAR')

Function: To declare a variable as a potential response of interest during an adjoint application run. A list of potential responses will be printed at the end of the run with row number and name.

Application: Adjoint

Arguments:
1) X: program variable to be declared
2) name or description of variable

Argument Type:
1) REAL*4 or REAL*8
2) CHARACTER*n (n < 24)

How to use it: Insert CALL POTRXX immediately following the line defining the variable. CLEARXX must be called at the end of the run. Use POTDXX if X is REAL*8. POTDXX is used exactly the same way as POTRXX.

Examples:

1) Declare X to be a potential response of interest.

```
      .
      .
      .
X = 2.0 * Y + Z
CALL POTRXX(X,'  X  ')
      .
      .
      .
```

2) Declare D(5) to be a response of interest.

```
      .
      .
      .
D(J) = B(J)**2
IF(J.EQ.5) CALL POTRXX(D(J),' D of 5 ')
      .
      .
      .
```

UTILITY ROUTINE

Name: SETRXX (CHAR)

Function: To specify the purpose for the run (i.e., chain rule propagation of derivatives or adjoint matrix generation).

Application: CHAIN or Adjoint

Arguments: 'CHAIN' or 'ADJOINT'

Argument Type: CHARACTER*5 or CHARACTER*7

How to use it: CALL SETRXX must be the first executed line in the unenhanced code, or the line immediately following CALL INITXX in the enhanced code. If CALL SETRXX is NOT made, no derivatives will be calculated.

Examples:

1) Enhanced code ready for chain rule propagation of derivatives.

```
      .
      .
      .
   DATA X /4.0/
   CALL INITXX(D00001)
   CALL SETRXX('CHAIN')
      .
      .
      .
```

2) Unenhanced code preparing for adjoint matrix generation.

```
      .
      .
      .
   DATA X /4.0/
   CALL SETRXX('ADJOINT')
      .
      .
      .
```

APPENDIX D

MFN.FOR COMPUTER CODE PROGRAM

```
C********************** ADJUSTMENT ALGORITHM ***************************
C THIS IS A THREE-LAYER FEEDFORWARD NEURAL NETWORK CODE PROGRAM BASED  *
C ON THE "ADJUSTMENT ALGORITHM" THAT USES MODEL ADJUSTMENT THEORY (MAT)*
C WHICH REDUCES THE UNCERTAINTIES IN CALCULATED RESPONSES OF INTEREST. *
C THIS METHOD REQUIRES THE AVAILABILITY OF SENSITIVITY FUNCTIONS AND   *
C UNCERTAINTY MATRIX OF MODEL DATA FIELD PARAMETERS PRIOR TO EVALUATING*
C A NEW SET OF WEIGHT VECTOR PARAMETERS.                               *
C***********************************************************************
C NI          - # INPUT PATTERN NODES                                  *
C NH          - # HIDDEN LAYER NODES                                   *
C NO          - # OUTPUT LAYER NODES                                   *
C NT          - # TRAINING SETS                                        *
C A(H)        - INPUT DATA VALUES TO Hth PE IN LAYER (L-1)             *
C V(H,I)      - WEIGHT ON CONNECTION JOINING Hth (L-1) & Ith (L) PEs   *
C THETA(I)    - Ith (L) PE's THRESHOLD VALUE                           *
C W(I,J)      - WEIGHT ON CONNECTION JOINING Ith (L) & Jth (L+1) PEs   *
C GAMMA(J)    - Jth (L+1) PE's THRESHOLD VALUE                         *
C B(I)        - ACTIVATION FUNCTION OF Ith (L) PE                      *
C C(J)        - ACTIVATION FUNCTION OF Jth (L+1) PE                    *
C DEV(J)      - DEVIATION BETWEEN CALCULATED AND DESIRED OUTPUTS       *
C Q(M)        - TOTAL WEIGHT VECTOR PARAMETERS                         *
C DESOUT(J)   - DESIRED OUTPUTS                                        *
C***********************************************************************
C
      INTEGER*4 SEED
      INTEGER H
      REAL SUMA,SUMB,SUMC,SUME
      PARAMETER (NI=9,NH=5,NO=1,NT=100,NELEM=65,NREM=56)
      DIMENSION A(NI),B(NH),C(NO),DEV(NO),STD(NO)
      DIMENSION THETA(NH),GAMMA(NO),V(NI,NH),W(NH,NO)
      DIMENSION R1(NH),R2(NO),GA(NO,NI),VX(NI,NI),VR(NO,NO)
      DIMENSION VRI(NO,NO),VAR(NI,NO),VRT(NO,NI),VA(NI,NI)
      DIMENSION VWI(NREM,NREM),F(NI,NO),DELTAQ(NREM),DELTAA(NI)
      DIMENSION STDW(NREM),GW(NO,NREM),GWT(NREM,NO),DESOUT(NO)
      DIMENSION GAT(NI,NO),STDA(NI),Q(NREM),VW(NREM,NREM)
      DIMENSION P(NO,NI),P1(NO,NO),P2(NREM,NO),P3(NREM,NREM)
      DIMENSION P4(NREM,NO),P5(NREM,NO),P6(NI,NO),P7(NI,NO)
      DIMENSION P8(NI,NI),P9(NI,NREM),P10(NI,NREM),P11(NI,NO)
      DIMENSION P12(NI,NO),P13(NI,NI),VX1(NI,NI),VX2(NI,NI)
      DIMENSION INDXR(NO),INDXW(NREM)
      EQUIVALENCE (Q(1),THETA(1)),(Q(NH+1),GAMMA(1))
      EQUIVALENCE (Q(NH+NO+1),V(1,1)),(Q(NH+NO+NI*NH+1),W(1,1))
C
C***********************************************************************
C READ INPUT DATA, WEIGHT VECTOR PARAMETERS, AND DESIRE OUTPUTS        *
C***********************************************************************
C
C
      PRINT *, 'ENTER INPUT DATA VALUES: '
      READ *, (A(H), H=1,NI)
      PRINT *, 'ENTER WEIGHT PARAMETERS V(H,I)'s: '
      READ *, ((V(H,I), I=1,NH), H=1,NI)
      PRINT *, 'ENTER THETA(I)'s: '
      READ *, (THETA(I), I=1,NH)
      PRINT *, 'ENTER WEIGHT PARAMETERS W(I,J)'s: '
      READ *, ((W(I,J), J=1,NO), I=1,NH)
      PRINT *, 'ENTER GAMMA(J)'s: '
```

```
      READ *, (GAMMA(J), J=1,NO)
      PRINT *, 'ENTER DESIRED OUTPUTS: '
      READ *, (DESOUT(J), J=1,NO)
C
C*************************************************************************
C READ SENSITIVITY FUNCTIONS OF INPUTS AND WEIGHT PARAMETERS            *
C*************************************************************************
C
      PRINT *, 'ENTER SENSITIVITY FUNCTIONS OF INPUTS: '
      READ *, ((GA(I,J), J=1,NI), I=1,NO)
      PRINT *, 'ENTER ESNSITIVITY FUNCTIONS OF WEIGHTS: '
      READ *, ((GW(I,J), J=1,NREM), I=1,NO)
C
C******READ UNCERTAINTY MATRIX OF INPUTS*********************************
C
      PRINT *, 'ENTER UNCERTAINTY MATRIX OF INPUTS: '
      READ *, ((VX(I,J), J=1,NI), I=1,NI)
C
C******INITIALIZATION OF WEIGHTS (OPTION)********************************
C
      DO 10 I=1,NH
      THETA(I)=2.0*RAN(SEED)-1.0
C      PRINT *,'THETA(I) IS', THETA(I)
10    CONTINUE
      DO 20 J=1,NO
      GAMMA(J)=2.0*RAN(SEED)-1.0
C      PRINT *,'GAMMA(J) IS', GAMMA(J)
20    CONTINUE
      DO 35 H=1,NI
      DO 30 I=1,NH
      V(H,I)=2.0*RAN(SEED)-1.0
30    CONTINUE
35    CONTINUE
      DO 45 I=1,NH
      DO 40 J=1,NO
      W(I,J)=2.0*RAN(SEED)-1.0
C      PRINT *,'W(I,J) IS', W(I,J)
40    CONTINUE
45    CONTINUE
C
C******ADJUSTMENT ALGORITHM WITH MAT PROCEDURE**************************
C
      DO 1000 K=1,NT
      DO 110 I=1,NH
      SUMB=0.0
      DO 100 H=1,NI
      SUMB=SUMB+A(H)*V(H,I)
C      PRINT *,'A(H) IS', A(H) ,'V(H,I)=',V(H,I),'SUMB=',SUMB
100   CONTINUE
      B(I)=SUMB+THETA(I)
      X=B(I)
C      PRINT *,'X=B(I)=',X,'EXP(-X)=',EXP(-X)
      R1(I)=1.0/(1.0+EXP(-X))
C      PRINT *,' '
C      PRINT *, ' RESULT1 IS: ', R1(I)
110   CONTINUE
      DO 130 J=1,NO
C
      SUMC=0.0
      DO 120 I=1,NH
      SUMC=SUMC+R1(I)*W(I,J)
120   CONTINUE
      C(J)=SUMC+GAMMA(J)
      Y=C(J)
      R2(J)=1.0/(1.0+EXP(-Y))
      DEV(J)=DESOUT(J)-R2(J)
```

```
C        PRINT *,
C        PRINT *, ' RESULT2 IS: ', R2(J)
C        PRINT *,' '
C        PRINT *, ' ERROR DEV(J) IS: ', DEV(J)
130     CONTINUE
C
C******CALCULATION OF DEREV OF OUTPUT WRT INPUT & ITS TRANSPOSE*********
C
        DO 170 H=1,NI
        DO 160 J=1,NO
        GAT(H,J)=GA(J,H)
C        PRINT *,'GAT IS', GAT(H,J)
160     CONTINUE
170     CONTINUE
C
C******CALCULATION OF DEREV OF OUTPUT WRT WEIGHT & ITS TRANSPOSE********
C
        DO 190 I=1,NREM
        DO 180 J=1,NO
        GWT(I,J)=GW(J,I)
C        PRINT *,'GWT IS', GWT(I,J)
180     CONTINUE
190     CONTINUE
C
C******CALCULATION OF UNCERTAINTY OF RESPONSES & ITS INVERSION**********
C
        CALL MATMPY(NO,NI,NI,1,1,1,GA,VX,P)
        CALL MATMPY(NO,NI,NO,1,1,1,P,GAT,P1)
        DO 300 I=1,NO
        DO 250 J=1,NO
        VR(I,J)=P1(I,J)
        STD(J)=SQRT(ABS(VR(J,J)))
250     CONTINUE
300     CONTINUE
        IF(NO.EQ.1) THEN
        DO 111 I=1,NO
        DO 222 J=1,NO
        VRI(I,J)=1.0/VR(I,J)
C        PRINT *,'VRI IS', VRI(I,J)
222     CONTINUE
111     CONTINUE
        ELSE
C        CALL MINV(VR,VRI,NO,NO,DET)
         DO 22 I=1,NO
         DO 23 J=1,NO
           VRI(I,J)=0.0
23       CONTINUE
           VRI(I,I)=1.0
22       CONTINUE
         CALL LUDCMP(VR,NO,NO,INDXR,D)
         DO 24 J=1,NO
          CALL LUBKSB(VR,NO,NO,INDXR,VRI(1,J))
24       CONTINUE
        ENDIF
C
C******CALCULATION OF UNCERTAINTY OF WEIGHTS & ITS INVERSION***********
C
C
C
        CALL MATMPY(NREM,NO,NO,1,1,1,GWT,VRI,P2)
        CALL MATMPY(NREM,NO,NREM,1,1,1,P2,GW,P3)
C        PRINT *,'P2 = GWT . VRI = ',P2
C        PRINT *,'P3 = P2 . GW = ',P3
        DO 500 I=1,NREM
        DO 400 J=1,NREM
        VW(I,J)=P3(I,J)
```

```
C        IF (VW(I,J) .EQ. 0.0) VW(I,J)=1.0E-26
C        PRINT *,'VW (',I,',',J,')= ',VW(I,J)
400     CONTINUE
        STDW(I)=SQRT(ABS(VW(I,I)))
500     CONTINUE
C        CALL MINV(VW,VWI,NREM,NREM,DET)
         DO 33 I=1,NREM
         DO 34 J=1,NREM
          VWI(I,J)=0.0
34       CONTINUE
          VWI(I,I)=1.0
33       CONTINUE
         CALL LUDCMP(VW,NREM,NREM,INDXW,D)
         DO 37 J=1,NREM
          CALL LUBKSB(VW,NREM,NREM,INDXW,VWI(1,J))
37       CONTINUE
C
C******CALCULATION OF ADJUSTED WEIGHTS**********************************
C
       CALL MATMPY(NREM,NREM,NO,1,1,1,VWI,GWT,P4)
       CALL MATMPY(NREM,NO,NO,1,1,1,P4,VRI,P5)
       DO 700 I=1,NREM
       DO 600 J=1,NO
       DELTAQ(I)=-1*P5(I,J)*DEV(J)
C       PRINT *,'DELTAQ ADJ IS', DELTAQ(I)
600    CONTINUE
       Q(I)=Q(I)+DELTAQ(I)
C       PRINT *,'ADJ WEIGHTS ARE',Q(I)
700    CONTINUE
       DO 13 I=1,NH
       THETA(I)=Q(I)
C       PRINT *,'THETA ADJ IS', THETA(I)
13     CONTINUE
       DO 14 J=1,NO
       GAMMA(J)=Q(NH+J)
C       PRINT *,'GAMMA ADJ IS', GAMMA(J)
14     CONTINUE
       DO 15 H=1,NI
       DO 15 I=1,NH
       KK=H+I-1
       V(H,I)=Q(NH+NO+KK)
C       PRINT *,'V(H,I) ADJ IS', V(H,I)
15     CONTINUE
       NN=NH+NO+NI*NH
       DO 16 I=1,NH
       DO 16 J=1,NO
       KK=I+J-1
       W(I,J)=Q(NN+KK)
C       PRINT *,'W(I,J) ADJ IS', W(I,J)
16     CONTINUE
C
C*******CALCULATION OF ADJUSTED INPUTS*********************************
C
       CALL MATMPY(NI,NI,NO,1,1,1,VX,GAT,P6)
       CALL MATMPY(NI,NO,NO,1,1,1,P6,VRI,P7)
       DO 900 H=1,NI
       DO 800 J=1,NO
        VAR(H,J)=P6(H,J)
        F(H,J)=P7(H,J)
800    CONTINUE
900    CONTINUE
       DO 255 H=1,NI
       SUMA=0.0
       DO 245 J=1,NO
       DO 240 I=1,NREM
       SUMA=SUMA+GW(J,I)*DELTAQ(I)
```

```
240     CONTINUE
        SUMA=SUMA+DEV(J)
        SUMA=-1*F(H,J)*SUMA
245     CONTINUE
        DELTAA(H)=SUMA
C        PRINT *,'DELTAA ADJ IS', DELTAA(H)
        A(H)=A(H)+DELTAA(H)
C        PRINT *,'ADJ INPUTS ARE',A(H)
255     CONTINUE
        DO 280 J=1,NO
        DO 270 H=1,NI
        VRT(J,H)=VAR(H,J)
270     CONTINUE
280     CONTINUE
C
C*******CALCULATION OF UNCERTAINTY OF ADJ INPUTS************************
C
        CALL MATMPY(NI,NO,NI,1,1,1,F,VRT,P8)
        DO 950 I=1,NI
        DO 850 J=1,NI
         VX1(I,J)=P8(I,J)
850     CONTINUE
950     CONTINUE
        CALL MATMPY(NI,NO,NREM,1,1,1,F,GW,P9)
        CALL MATMPY(NI,NREM,NREM,1,1,1,P9,VWI,P10)
        CALL MATMPY(NI,NREM,NO,1,1,1,P10,GWT,P11)
        CALL MATMPY(NI,NO,NO,1,1,1,P11,VRI,P12)
        CALL MATMPY(NI,NO,NI,1,1,1,P12,VRT,P13)
        DO 955 I=1,NI
        DO 855 J=1,NI
         VX2(I,J)=P13(I,J)
855     CONTINUE
955     CONTINUE
        DO 340 H=1,NI
        VA(H,H)=VX(H,H)-VX1(H,H)+VX2(H,H)
        STDA(H)=SQRT(ABS(VA(H,H)))
340     CONTINUE
        DO 200 J=1,NO
C        IF(K.EQ.1) GOTO 210
        IF(ABS(DEV(J)).LE.0.01) GOTO 210
200     CONTINUE
        GOTO 1000
210     NTRAIN=K
        LAYER=J
        ILAYER=I
        LAYERH=H
        DO 1 I=1,NREM
        PRINT *,'ADJ WEIGHTS ARE=',Q(I)
1       CONTINUE
        DO 2 H=1,NI
        PRINT *,' '
        PRINT *,'ADJ INPUTS ARE=',A(H)
2       CONTINUE
        GOTO 5000
1000    CONTINUE
        PRINT *,' THERE WERE NO VALUES OF DEV(J)<=0.01 '
        GOTO 6000
5000    PRINT *,' THE VALUE OF DEV(J) THAT IS <=0.01 '
        PRINT *,' IS: ',DEV(LAYER), ' AND ITS K VALUE IS: ',NTRAIN
        PRINT *,' THE DESIRED OUTPUT IS: ',DESOUT
        PRINT *,' '
        PRINT *,' THE VARIANCE-COVARIANCE MATRIX OF RESIDUALS IS'
        DO 333 I=1,NO
        PRINT *, (VR(I,J), J=1,NO)
        PRINT *,'THE STANDARD DEVIATIONS OF RESIDUALS ARE',STD(I)
333     CONTINUE
```

```
      PRINT *,' '
      PRINT *,' THE VARIANCE-COVARIANCE MATRIX OF WEIGHTS IS'
      DO 444 I=1,NREM
      PRINT *, (VWI(I,J), J=1,NREM)
      PRINT *,'THE STANDARD DEVIATIONS OF WEIGHTS ARE',STDW(I)
444   CONTINUE
      PRINT *,' '
      PRINT *,' THE VARIANCE-COVARIANCE MATRIX OF OUTPUTS IS'
      DO 555 I=1,NI
      PRINT *, (VA(I,J), J=1,NI)
      PRINT *,'THE STANDARD DEVIATIONS OF OUTPUTS ARE',STDA(I)
555   CONTINUE
       STOP
       END
C*************************************************************************
C SUBROUTINE OF MATRIX MULTIPLICATION                                     *
C*************************************************************************
      SUBROUTINE MATMPY(M,N,K,X,Y,Z,A,B,C)
      INTEGER M,N,K,R,S,I,X,Y,Z
      REAL A(M,N),B(N,K),C(M,K),SUM
      DO 100 R=X,M
      DO 200 S=Y,K
       SUM=0.0
       DO 300 I=Z,N
        SUM=SUM+A(R,I)*B(I,S)
300    CONTINUE
       C(R,S)=SUM
200    CONTINUE
100    CONTINUE
       RETURN
       END
C*************************************************************************
C SUBROUTINE OF MATRIX INVERSE                                            *
C*************************************************************************

      subroutine minv(u,t,nd,n,det)
c****** t= inverse of u; both are n*n
c     det= determinant value
      dimension u(n,n),t(n,n)
      common hh(56,56)
      do 1 i=1,n
      do 1 j=1,n
 1    hh(i,j)=u(i,j)
      do 2 i=1,n
      do 2 j=1,n
      t(i,j)=0.0
 2    if (i.eq.j) t(i,j)=1.0
      det=1.0
      do 9 i=1,n
      m= i
      do 3 j=i,n
 3    if (abs(hh(j,i)).gt.abs(hh(m,i))) m=j
      if (m.eq.i) go to 6
      do 4 j=i,n
      s= hh(i,j)
      hh(i,j)=hh(m,j)
 4    hh(m,j)=s
      do 5 j=1,n
      s=t(i,j)
      t(i,j)=t(m,j)
 5    t(m,j)=s
      det=-1.0*det
 6    det=hh(i,i)*det
      if(det .eq. 0.0) go to 10
      s=1.0/hh(i,i)
      do 7 j=1,n
```

```
        if (j.gt.i) hh(i,j)=s*nh(i,j)
  7     t(i,j)=s*t(i,j)
        do 9 j=1,n
        if (j.eq.i) go to 9
        s=hh(j,i)
        do 8 k=1,n
        if (k.gt.i) hh(j,k)=hh(j,k)-s*hh(i,k)
  8     t(j,k)=t(j,k)-s*t(i,k)
  9     continue
  10    print*,  'det  = 0.0'
        return
        end
C*************************************************************************
C SUBROUTINE OF MATRIX INVERSE                                           *
C*************************************************************************
        SUBROUTINE LUDCMP(A,N,NP,INDX,D)
        PARAMETER (NMAX=100,TINY=1.0E-20)
        DIMENSION A(NP,NP),INDX(N),VV(NMAX)
        D=1.0
        DO 12 I=1,N
        AAMAX=0.0
        DO 11 J=1,N
        IF(ABS(A(I,J)) .GT. AAMAX)  AAMAX=ABS(A(I,J))
11      CONTINUE
        IF (AAMAX .EQ. 0.0)  PAUSE 'SINGULAR MATRIX'
        VV(I)=1.0/AAMAX
12      CONTINUE
        DO 19 J=1,N
        DO 14 I=1,J-1
        SUM=A(I,J)
        DO 13 K=1,I-1
        SUM=SUM-A(I,K)*A(K,J)
13      CONTINUE
        A(I,J)=SUM
14      CONTINUE
        AAMAX=0.0
        DO 16 I=J,N
        SUM=A(I,J)
        DO 15 K=1,J-1
        SUM=SUM-A(I,K)*A(K,J)
15      CONTINUE
        A(I,J)=SUM
        DUMM=VV(I)*ABS(SUM)
        IF (DUMM .GE. AAMAX) THEN
        IMAX=I
        AAMAX=DUMM
        ENDIF
16      CONTINUE
        IF (J .NE. IMAX)  THEN
        DO 17 K=1,N
        DUMM=A(IMAX,K)
        A(IMAX,K)=A(J,K)
        A(J,K)=DUMM
17      CONTINUE
        D=-D
        VV(IMAX)=VV(J)
        ENDIF
        INDX(J)=IMAX
        IF (A(J,J) .EQ. 0.0) A(J,J) = TINY
        IF (J .NE. N)  THEN
        DUMM =1.0/A(J,J)
        DO 18 I=J+1,N
        A(I,J)=A(I,J)*DUMM
18      CONTINUE
        ENDIF
19      CONTINUE
```

```
      RETURN
      END
      SUBROUTINE LUBKSB(A,N,NP,INDX,B)
      DIMENSION A(NP,NP),INDX(N),B(N)
      II=0
      DO 12 I=1,N
      LL=INDX(I)
      SUM=B(LL)
      B(LL)=B(I)
C      PRINT *,'II=',II,'SUM=',SUM
      IF (II .NE. 0)  THEN
      DO 11 J=II,I-1
      SUM=SUM-A(I,J)*B(J)
11    CONTINUE
      ELSE IF (SUM .NE. 0.0)  THEN
      II=I
      ENDIF
      B(I) =SUM
12    CONTINUE
      DO 14 I=N,1,-1
      SUM=B(I)
      DO 13 J=I+1,N
      SUM=SUM-A(I,J)*B(J)
13    CONTINUE
      B(I)=SUM/A(I,I)
14    CONTINUE
      RETURN
      END
```

```
C********************* UNCERTAINTY ALGORITHM ***************************
C THIS PROGRAM DETERMINES THE UNCERTAINTY ANALYSIS OF MULTILAYER FEED- *
C FORWARD NEURAL NETWORK.                                               *
C***********************************************************************
C
      INTEGER H
      PARAMETER (NI=10,NO=1,NELEM=65,NREM=56)
      DIMENSION GA(NO,NI),VX(NI,NI),VR(NO,NO)
      DIMENSION VRI(NO,NO),VAR(NI,NO),VRT(NO,NI),VA(NI,NI)
      DIMENSION VWI(NREM,NREM),F(NI,NO),STDR(NO),STDO(NO)
      DIMENSION STDW(NREM),GW(NO,NREM),GWT(NREM,NO)
      DIMENSION GAT(NI,NO),STDA(NI),VW(NREM,NREM),VC(NO,NO)
      DIMENSION P(NO,NI),P1(NO,NO),P2(NREM,NO),P3(NREM,NREM)
      DIMENSION P6(NI,NO),P7(NI,NO)
      DIMENSION P8(NI,NI),P9(NI,NREM),P10(NI,NREM),P11(NI,NO)
      DIMENSION P12(NI,NO),P13(NI,NI),VX1(NI,NI),VX2(NI,NI)
      DIMENSION P14(NO,NI),P15(NO,NO),P16(NO,NO),P17(NO,NO)
      DIMENSION P18(NO,NO),P19(NO,NREM),P20(NO,NREM),P21(NO,NO)
      DIMENSION VCR(NO,NO),VCRT(NO,NO),VC1(NO,NO),VC0(NO,NREM)
      DIMENSION VCQT(NREM,NO),VC2(NO,NO),VCO(NO,NO),INDX(NREM)
C
C*** READ GRADIENT INFORMATION OF INPUTS & WEIGHT PARAMETERS ***********
C
      PRINT *, 'ENTER GRADIENT CALCULATION OF INPUTS: '
      READ *, ((GA(I,J), J=1,NI), I=1,NO)
      PRINT *, 'ENTER GRADIENT CALCULATION OF WEIGHTS: '
      READ *, ((GW(I,J), J=1,NREM), I=1,NO)
      PRINT *, 'ENTER UNCERTAINTY MATRIX OF INPUTS: '
      READ *, ((VX(I,J), J=1,NI), I=1,NI)
      PRINT *, 'ENTER UNCERTAINTY MATRIX OF OUTPUTS: '
      READ *, ((VC(I,J), J=1,NO), I=1,NO)
C
C*** CALCULATION OF VAR.COV. MATRIX OF RESIDUALS ************************
C
      DO 170 H=1,NI
      DO 160 J=1,NO
      GAT(H,J)=GA(J,H)
C      PRINT *,'GAT IS', GAT(H,J)
160   CONTINUE
170   CONTINUE
      DO 190 I=1,NREM
      DO 180 J=1,NO
      GWT(I,J)=GW(J,I)
C      PRINT *,'GWT IS', GWT(I,J)
180   CONTINUE
190   CONTINUE
      CALL MATMPY(NO,NI,NI,1,1,1,GA,VX,P)
      CALL MATMPY(NO,NI,NO,1,1,1,P,GAT,P1)
      DO 300 I=1,NO
      DO 250 J=1,NO
      VR(I,J)=P1(I,J)
250   CONTINUE
      STDR(I)=SQRT(ABS(VR(I,I)))
300   CONTINUE
C
C*** CALCULATION OF INVERSION OF VR (VAR.COV. MATRIX OF RESIDUALS) *****
C
```

```
      IF(NO.EQ.1) THEN
      DO 111 I=1,NO
      DO 222 J=1,NO
      VRI(I,J)=1.0/VR(I,J)
C      PRINT *,'VRI IS', VRI(I,J)
222   CONTINUE
111   CONTINUE
      ELSE
C      CALL MINV(VR,VRI,NO,NO,DET)
      DO 22 I=1,NO
      DO 23 J=1,NO
         VRI(I,J)=0.0
23    CONTINUE
         VRI(I,I)=1.0
22    CONTINUE
      CALL LUDCMP(VR,NO,NO,INDXR,D)
      DO 24 J=1,NO
      CALL LUBKSB(VR,NO,NO,INDXR,VRI(I,J))
24    CONTINUE
      ENDIF
C
C*** CALCULATION OF VAR.COV. MATRIX OF WEIGHT PARAMETERS ***************
C
      CALL MATMPY(NREM,NO,NO,1,1,1,GWT,VRI,P2)
      CALL MATMPY(NREM,NO,NREM,1,1,1,P2,GW,P3)
C      PRINT *,'P2 = GWT . VRI = ',P2
C      PRINT *,'P3 = P2 . GW = ',P3
      DO 500 I=1,NREM
      DO 400 J=1,NREM
      VW(I,J)=P3(I,J)
C      IF (VW(I,J) .EQ. 0.0) VW(I,J)=1.0E-26
C      PRINT *,'VW (',I,',',J,')= ',VW(I,J)
400   CONTINUE
500   CONTINUE
C      CALL MINV(VW,VWI,NREM,NREM,DET)
      DO 12 I=1,NREM
      DO 11 J=1,NREM
        VWI(I,J)=0.0
11    CONTINUE
        VWI(I,I)=1.0
12    CONTINUE
      CALL LUDCMP(VW,NREM,NREM,INDX,D)
      DO 13 J=1,NREM
      CALL LUBKSB(VW,NREM,NREM,INDX,VWI(1,J))
13    CONTINUE
     DO 5 I=1,NREM
      STDW(I)=SQRT(ABS(VW(I,I)))
5     CONTINUE
C
C*** CALCULATION OF VAR.COV. MATRIX OF INPUTS ***************************
C
      CALL MATMPY(NI,NI,NO,1,1,1,VX,GAT,P6)
      CALL MATMPY(NI,NO,NO,1,1,1,P6,VRI,P7)
      DO 900 H=1,NI
      DO 800 J=1,NO
       VAR(H,J)=P6(H,J)
       F(H,J)=P7(H,J)
800   CONTINUE
900   CONTINUE
      DO 280 J=1,NO
      DO 270 H=1,NI
      VRT(J,H)=VAR(H,J)
270   CONTINUE
280   CONTINUE
      CALL MATMPY(NI,NO,NI,1,1,1,F,VRT,P8)
      DO 950 I=1,NI
```

```
      DO 850 J=1,NI
       VX1(I,J)=P8(I,J)
850   CONTINUE
950   CONTINUE
      CALL MATMPY(NI,NO,NREM,1,1,1,F,GW,P9)
      CALL MATMPY(NI,NREM,NREM,1,1,1,P9,VWI,P10)
      CALL MATMPY(NI,NREM,NO,1,1,1,P10,GWT,P11)
      CALL MATMPY(NI,NO,NO,1,1,1,P11,VRI,P12)
      CALL MATMPY(NI,NO,NI,1,1,1,P12,VRT,P13)
      DO 955 I=1,NI
      DO 855 J=1,NI
       VX2(I,J)=P13(I,J)
855   CONTINUE
955   CONTINUE
      DO 340 H=1,NI
      VA(H,H)=VX(H,H)-VX1(H,H)+VX2(H,H)
      STDA(H)=SQRT(ABS(VA(H,H)))
340   CONTINUE
C*** CALCULATION OF VAR.COV. MATRIX OF OUTPUTS **************************
C
      CALL MATMPY(NO,NI,NI,1,1,1,GA,VX,P14)
      CALL MATMPY(NO,NI,NO,1,1,1,P14,GAT,P15)
      CALL MATMPY(NO,NO,NO,1,1,1,P15,VRI,P16)
      DO 10 I=1,NO
      DO 20 J=1,NO
       VCR(I,J)=P15(I,J)
       VCRT(J,I)=VCR(I,J)
20    CONTINUE
10    CONTINUE
      CALL MATMPY(NO,NO,NO,1,1,1,P16,VCRT,P17)
      DO 30 I=1,NO
      DO 40 J=1,NO
      VC1(I,J)=P17(I,J)
40    CONTINUE
30    CONTINUE
      CALL MATMPY(NO,NO,NO,1,1,1,VCR,VRI,P18)
      CALL MATMPY(NO,NO,NREM,1,1,1,P18,GW,P19)
      DO 50 J=1,NO
      DO 60 I=1,NREM
       VCQ(J,I)=GW(J,I)-P19(J,I)
       VCQT(I,J)=VCQ(J,I)
60    CONTINUE
50    CONTINUE
      CALL MATMPY(NO,NREM,NREM,1,1,1,VCQ,VWI,P20)
      CALL MATMPY(NO,NREM,NO,1,1,1,P20,VCQT,P21)
      DO 70 I=1,NO
      DO 80 J=1,NO
       VC2(I,J)=P21(I,J)
       VCO(I,J)=VC(I,J)-VC1(I,J)+VC2(I,J)
80    CONTINUE
       STDO(I)=SQRT(ABS(VCO(I,I)))
70    CONTINUE
C*** PRINT THE UNCERTAINTY RESULTS **************************************
C
      PRINT *,' '
      PRINT *,' THE VARIANCE-COVARIANCE MATRIX OF RESIDUALS IS'
      DO 333 I=1,NO
      PRINT *, (VR(I,J), J=1,NO)
      PRINT *,'THE STANDARD DEVIATIONS OF RESIDUALS ARE',STDR(I)
333   CONTINUE
      PRINT *,' '
      PRINT *,' THE VARIANCE-COVARIANCE MATRIX OF WEIGHTS IS'
      DO 444 I=1,NREM
      PRINT *, (VW(I,J), J=1,NREM)
      PRINT *,'THE STANDARD DEVIATIONS OF WEIGHTS ARE',STDW(I)
444   CONTINUE
```

```
      PRINT *,' '
      PRINT *,' THE VARIANCE-COVARIANCE MATRIX OF INPUTS IS'
      DO 555 I=1,NI
      PRINT *, (VA(I,J), J=1,NI)
      PRINT *,'THE STANDARD DEVIATIONS OF INPUTS ARE',STDA(I)
555   CONTINUE
      PRINT *,' '
      PRINT *,' THE VARIANCE-COVARIANCE MATRIX OF OUTPUTS IS'
      DO 666 I=1,NO
      PRINT *, (VCO(I,J), J=1,NO)
      PRINT *,' THE STANDARD DEVIATIONS OF OUTPUTS ARE',STDO(I)
666   CONTINUE
      STOP
      END
C**** SUBROUTINE FOR MATRIX MULTIPLICATION ******************************
C
      SUBROUTINE MATMPY(M,N,K,X,Y,Z,A,B,C)
      INTEGER M,N,K,R,S,I,X,Y,Z
      REAL A(M,N),B(N,K),C(M,K),SUM
      DO 100 R=X,M
      DO 200 S=Y,K
       SUM=0.0
       DO 300 I=Z,N
        SUM=SUM+A(R,I)*B(I,S)
300    CONTINUE
        C(R,S)=SUM
200    CONTINUE
100    CONTINUE
       RETURN
       END
C**** SUBROUTINE FOR MATRIX INVERSE *************************************
C
      SUBROUTINE MINV(A,AINV,ND,N,DET)
      REAL A(ND,ND),AINV(ND,ND),DET
      INTEGER ND,N,IPASS
       DET=1.0
       DO 1 I=1,N
       DO 1 J=1,N
          IF(I .EQ. J) THEN
           AINV(I,I)=1.0
       ELSE
           AINV(I,J)=0.0
       ENDIF
1      CONTINUE
       DO 9 IPASS=1,N
        IMX=IPASS
       DO 2 IROW=IPASS,N
         IF(ABS(A(IROW,IPASS)) .GT. ABS(A(IMX,IPASS))) THEN
        IMX=IROW
       ENDIF
2      CONTINUE
          IF(IMX .NE. IPASS) THEN
        DO 3 ICOL=1,N
           TEMP=AINV(IPASS,ICOL)
           AINV(IPASS,ICOL)=AINV(IMX,ICOL)
           AINV(IMX,ICOL)=TEMP
           IF(ICOL .GE. IPASS) THEN
         TEMP=A(IPASS,ICOL)
         A(IPASS,ICOL)=A(IMX,ICOL)
         A(IMX,ICOL)=TEMP
         ENDIF
3       CONTINUE
        ENDIF
            PIVOT=A(IPASS,IPASS)
C            COUNTER=COUNTER+1
C            PRINT *,'PIVOT =',PIVOT ,'COUNTER = ',COUNTER
```

```
            IF (PIVOT .EQ. 0.0) PIVOT = 1.0E-20
C            PRINT *,'PIVOT=',PIVOT
            DET=DET*PIVOT
C            PRINT *,'DET IS',DET
            IF (DET .EQ. 0.0)
C            PRINT *,'DET=',DET
          IF(DET .EQ. 0.0) PAUSE 'SINGULAR MATRIX'
         DO 6 ICOL=1,N
          AINV(IPASS,ICOL)=AINV(IPASS,ICOL)/PIVOT
C          PRINT *,'AINV(',IPASS,',',ICOL,')= ',AINV(IPASS,ICOL)
          IF(ICOL .GE. IPASS) THEN
          A(IPASS,ICOL)=A(IPASS,ICOL)/PIVOT
          ENDIF
6        CONTINUE
          DO 8 IROW=1,N
          IF(IROW .NE. IPASS) THEN
          FACTOR=A(IROW,IPASS)
         ENDIF
          DO 7 ICOL=1,N
          IF(IROW .NE. IPASS) THEN
        AINV(IROW,ICOL)=AINV(IROW,ICOL)-FACTOR*AINV(IPASS,ICOL)
        PRINT *,'AINV=',AINV(IROW,ICOL)
        A(IROW,ICOL)=A(IROW,ICOL)-FACTOR*A(IPASS,ICOL)
         ENDIF
7        CONTINUE
8        CONTINUE
9        CONTINUE
        RETURN
         END
C**** SUBROUTINE FOR MATRIX INVERSE ************************************
        SUBROUTINE LUDCMP(A,N,NP,INDX,D)
        PARAMETER (NMAX=100,TINY=1.0E-20)
        DIMENSION A(NP,NP),INDX(N),VV(NMAX)
        D=1.0
        DO 12 I=1,N
         AAMAX=0.0
        DO 11 J=1,N
        IF (ABS(A(I,J)) .GT. AAMAX) AAMAX=ABS(A(I,J))
11      CONTINUE
        IF (AAMAX .EQ. 0.0) PAUSE 'SINGULAR MATRIX'
        VV(I)=1.0/AAMAX
12      CONTINUE
        DO 19 J=1,N
         DO 14 I=1,J-1
          SUM=A(I,J)
           DO 13 K=1,I-1
            SUM=SUM-A(I,K)*A(K,J)
13       CONTINUE
         A(I,J)=SUM
14       CONTINUE
         AAMAX=0.0
        DO 16 I=J,N
           SUM=A(I,J)
        DO 15 K=1,J-1
           SUM=SUM-A(I,K)*A(K,J)
15       CONTINUE
         A(I,J)=SUM
         DUM=VV(I)*ABS(SUM)
         IF (DUM .GE. AAMAX) THEN
            IMAX=I
            AAMAX=DUM
        ENDIF
16      CONTINUE
        IF (J .NE. IMAX) THEN
        DO 17 K=1,N
           DUM=A(IMAX,K)
```

```
            A(IMAX,K)=A(J,K)
            A(J,K)=DUM
17          CONTINUE
            D=-D
            VV(IMAX)=VV(J)
            ENDIF
            INDX(J)=IMAX
            IF (A(J,J).EQ.0.0) A(J,J)=TINY
            IF (J.NE.N) THEN
               DUM=1.0/A(J,J)
            DO 18 I=J+1,N
               A(I,J)=A(I,J)*DUM
18          CONTINUE
            ENDIF
19          CONTINUE
            RETURN
            END
            SUBROUTINE LUBKSB(A,N,NP,INDX,B)
            DIMENSION A(NP,NP),INDX(N),B(N)
            II=0
            DO 12 I=1,N
             LL=INDX(I)
             SUM=B(LL)
             B(LL)=B(I)
            IF (II.NE.0) THEN
            DO 11 J=II,I-1
               SUM=SUM-A(I,J)*B(J)
11          CONTINUE
            ELSE IF (SUM.NE.0.0) THEN
              II=I
             ENDIF
            B(I)=SUM
12          CONTINUE
            DO 14 I=N,1,-1
               SUM=B(I)
            DO 13 J=I+1,N
              SUM=SUM-A(I,J)*B(J)
13          CONTINUE
              B(I)=SUM/A(I,I)
14          CONTINUE
            RETURN
            END
```

VITA

Aiman Fateh Amer was born in Kuwait City, Kuwait on November 10, 1961. He attended elementary and high schools in that city and was graduated from Abdullah Al-Salem High School in June 1979. The following September he accepted a plant operator position at the Petrochemical Industries Company (PIC) for two years. In September 1983, he entered the University of Tennessee, Knoxville, Department of Engineering, and in August 1986 he received a Bachelor of Science degree in Nuclear Engineering.

Beginning in the Fall of 1986, he accepted a teaching assistantship with the Nuclear Engineering Department of the University of Tennessee, Knoxville, in conjunction with studying for a master of Science degree in Nuclear Engineering. This degree was awarded in December 1988. After joining the Fusion Physics Division of the Oak Ridge National Laboratory (ORNL) and working toward his Ph.D., he was awarded the Ph.D. degree in December 1991.

The author is a member of the American Institute of Physics (AIP), the American Nuclear Society (ANS), the American Society of Mechanical Engineers (ASME), and the Institute of Electrical and Electronics Engineers (IEEE).

He is married to the former Thanaa A. Miswadeh of Jordan.