To the Graduate Council:

I am submitting herewith a thesis written by Bobby Lamar Lepard, Jr., entitled "Implementation of Time Step Control Methods for CFD Algorithm Stability and Efficiency." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Engineering Science.

Dr. A. J. Baker, Major Professor

We have read this thesis and
recommend its accceptance:

Accepted for the Council:

Associate Vice Chancellor and
Dean of the Graduate School

IMPLEMENTATION OF TIME STEP CONTROL METHODS FOR CFD ALGORITHM STABILITY AND EFFICIENCY

A Thesis
Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville

Bobby Lamar Lepard, Jr.
August 1995

Dedication

This thesis is dedicated to my parents Bobby L., Sr., and Betty B. Lepard and to my wife Cullum E. Lepard

Acknowledgment

There are many people to whom I am grateful for assisting me in this effort. I have benefitted in both knowledge and experience from knowing many faculty and students in the Engineering Science and Mechanics Department at the University of Tennessee. I am especially grateful to my major professor, Dr. A. J. Baker, and to my thesis committee, Drs. R. D. Krieg and G. S.Iannelli, for their willingness to thoroughly review my work and for their suggestions and comments that greatly improved the quality of the document.

In addition, I would like to thank Messrs. Joe Orzechowski and Ed Schaub at the Computational Mechanics Corporation for their assistance with the *AKCESS.* * computer code. Their prompt responses to my questions about the code helped me greatly.

I am also appreciative of the help I received from Dr. Jim Freels, a co-worker of mine at the Oak Ridge National Laboratory. Since the underlying theory used in this thesis is

based on a portion of Dr. Freels' dissertation, he provided much insight to assist me in the proper implementation of the theory.

I am thankful for the reimbursement I received for tuition, books, and supplies through the Lockheed Martin Energy Systems, Inc., Educational Assistance Program.

Finally, I want to thank my wife, Cullum, who encouraged me to continue and who tolerated my absenses and my many stacks of books and papers in our home.

IMPLEMENTATION OF TIME STEP CONTROL METHODS FOR CFD ALGORITHM STABILITY AND EFFICIENCY

Bobby Lamar Lepard, Jr.
The University of Tennessee

Abstract

A time step control method has been implemented and evaluated for CFD algorithm stability, hence efficiency, via the *AKCESS.* * software platform which utilizes a "template" and "hook" format for implementing discrete algorithm variations. The time step control method monitors the extremum Newton iterate change in the solution field, at the beginning of each time step, and uses this information on line to adjust the time step size for the next time station. This method has been tested on two benchmark problems (driven cavity and thermal cavity) using weak statement finite element formulations based on vorticity-streamfunction, the continuity constraint (pressure relaxation) method, and the penalty method. A test matrix containing 35 cases is presented, demonstrating the versatility and effectiveness of the time step control method for both implicit and explicit time integration. Overall, the time step controller is a tool that enables a problem to converge to a steady-state solution with minimal user interaction (i.e., frustration) with the computer.

Table of Contents

1	**Introduction**	1
2	**Conservation Law Statements in Fluid Mechanics**	4
	2.1 The Continuity Equation	5
	2.2 The Momentum Equations	7
	2.3 The Energy Equation	11
	2.4 Dimensionless Forms	12
	2.5 Vorticity-Streamfunction Formulation	15
3	**Weak Statement Algorithm**	18
	3.1 Galerkin Weak Statement	20
	3.2 Finite Element Implementation	22
	3.3 Boundary Conditions	27
4	**Algorithm Stability Theory**	28
	4.1 Conditions for Stability	29
	4.2 On-Line Stability Monitoring	35
	4.3 On-Line Time Step Control	36
5	**Implementation of On-Line Stability Monitoring and Time Step Control**	38
	5.1 Original Time Step Control in the BOTTOM Hook	39
	5.2 Modified Time Step Control in the BOTTOM Hook	40

6 Discussion and Results 46
6.1 Vorticity-Streamfunction Performance 47
6.1.1 2D Driven Cavity, Re = 10 47
6.1.1.1 θ = 0.0 Cases 48
6.1.1.2 θ = 0.5 Cases 60
6.1.1.3 θ = 1.0 Cases 65
6.1.1.4 Summary, Re = 10 Driven Cavity 71
6.1.2 2D Driven Cavity, Re = 100 74
6.1.2.1 θ = 0.0 Cases 74
6.1.2.2 θ = 0.5 Cases 79
6.1.2.3 θ = 1.0 Cases 84
6.1.2.4 Summary, Re = 100 Driven Cavity 88
6.1.3 2D Thermal Cavity, Ra = 1000 90
6.1.3.1 θ = 0.0 Cases 90
6.1.3.2 θ = 0.5 Cases 99
6.1.3.3 θ = 1.0 Cases 103
6.1.3.4 Summary, Ra = 1000 Thermal Cavity . 110
6.2 Continuity Constraint Method Evaluation 114
6.2.1 2D Driven Cavity, Re = 100 114
6.2.2 2D Thermal Cavity, Ra = 1000 124
6.3 Penalty Method Problems 131
6.3.1 2D Driven Cavity, Re = 100 131
6.3.1.1 θ = 0.5 Cases 132
6.3.1.2 θ = 1.0 Cases 136
6.3.1.3 Summary, Re = 100 Driven Cavity . . . 140
6.3.2 2D Thermal Cavity, Ra = 1000 141
6.3.2.1 θ = 0.5 Cases 142
6.3.2.2 θ = 1.0 Cases 148
6.3.2.3 Summary, Ra = 1000 Thermal Cavity . . 153

7 Conclusion 154

References 156

Appendices 159

Appendix A--BOTTOM Hook FORTRAN Listing 160

Appendix B--Test Problem Descriptions 173

Vita 177

List of Tables

Table 5-1 Time Step Control Method User Input Parameters and Settings . . 45

Table 6-1 Case 1 Results 49

Table 6-2 Summary of Re = 10 Vorticity-Streamfunction Driven Cavity Cases . 73

Table 6-3 Case 10 Results 75

Table 6-4 Summary of Re = 100 Vorticity-Streamfunction Driven Cavity Cases . 89

Table 6-5 Summary of Ra = 1000 Vorticity-Streamfunction Thermal Cavity Cases 113

Table 6-6 Summary of Re = 100 Continuity Constraint Driven Cavity Cases . 124

Table 6-7 Summary of Ra = 1000 Continuity Constraint Thermal Cavity Cases . 130

Table 6-8 Summary of Re = 100 Penalty Driven Cavity Cases 141

Table 6-9 Summary of Ra = 1000 Penalty Thermal Cavity Cases 153

List of Figures

Fig. 4-1. Solution Convergence Characteristics Schematic 29

Fig. 6-1. Case 2 Time Step Size vs. Time Step 51

Fig. 6-2. Case 2 $\left|\frac{\delta\Omega}{\Omega_o}\right|_{max}$ Values vs. Time Step for Vorticity 52

Fig. 6-3. Case 2 $\left|\frac{\delta\Omega}{\Omega_o}\right|_{max}$ Values vs. Time Step for Streamfunction 52

Fig. 6-4. Case 2 $\frac{|\delta\Omega|_{max}}{|\Omega_o|_{max}}$ Values vs. Time Step for Vorticity 53

Fig. 6-5. Case 2 $\frac{|\delta\Omega|_{max}}{|\Omega_o|_{max}}$ Values vs. Time Step for Streamfunction 53

Fig. 6-6. Case 3 Time Step Size vs. Time Step 54

Fig. 6-7. Case 4 Time Step Size vs. Time Step 56

Fig. 6-8. Case 4 $\frac{|\delta\Omega|_{max}}{|\Omega_o|_{max}}$ Values vs. Time Step for Vorticity 57

Fig. 6-9. Case 4 $\frac{|\delta\Omega|_{max}}{|\Omega_o|_{max}}$ Values vs. Time Step for Streamfunction 58

Fig. 6-10. Case 5 Time Step Size vs. Time Step 61

Fig. 6-11. Case 5 $\frac{|\delta\Omega|_{max}}{|\Omega_o|_{max}}$ Values vs. Time Step for Vorticity 61

Fig. 6-12. Case 5 $\frac{|\delta\Omega|_{max}}{|\Omega_o|_{max}}$ Values vs. Time Step for Streamfunction 62

Fig. 6-13. Case 6 Time Step Size vs. Time Step 63

Fig. 6-14. Case 6 $\frac{|\delta\Omega|_{max}}{|\Omega_o|_{max}}$ Values vs. Time Step for Vorticity 64

Fig. 6-15. Case 6 $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ Values vs. Time Step for Streamfunction 64

Fig. 6-16. Case 7 Time Step Size vs. Time Step 66

Fig. 6-17. Case 7 $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ Values vs. Time Step for Vorticity 66

Fig. 6-18. Case 7 $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ Values vs. Time Step for Streamfunction 67

Fig. 6-19. Case 8 Time Step Size vs. Time Step 68

Fig. 6-20. Case 8 $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ Values vs. Time Step for Vorticity 68

Fig. 6-21. Case 8 $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ Values vs. Time Step for Streamfunction 69

Fig. 6-22. Case 9 Time Step Size vs. Time Step 70

Fig. 6-23. Case 9 $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ Values vs. Time Step for Vorticity 70

Fig. 6-24. Case 9 $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ Values vs. Time Step for Streamfunction 71

Fig. 6-25. Case 11 Time Step Size vs. Time Step 77

Fig. 6-26. Case 11 $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ Values vs. Time Step for Vorticity 77

Fig. 6-27. Case 11 $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ Values vs. Time Step for Streamfunction 78

Fig. 6-28. Case 12 Time Step Size vs. Time Step 80

Fig. 6-29. Case 12 $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ Values vs. Time Step for Vorticity 80

Fig. 6-30. Case 12 $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ Values vs. Time Step for Streamfunction 81

Fig. 6-31. Case 13 Time Step Size vs. Time Step 82

Fig. 6-32. Case 13 $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ Values vs. Time Step for Vorticity 82

Fig. 6-33. Case 13 $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ Values vs. Time Step for Streamfunction 83

Fig. 6-34. Case 14 Time Step Size vs. Time Step 85

Fig. 6-35. Case 14 $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ Values vs. Time Step for Vorticity 85

Fig. 6-36. Case 14 $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ Values vs. Time Step for Streamfunction 86

Fig. 6-37. Case 15 Time Step Size vs. Time Step 86

Fig. 6-38. Case 15 $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ Values vs. Time Step for Vorticity 87

Fig. 6-39. Case 15 $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ Values vs. Time Step for Streamfunction 87

Fig. 6-40. Case 16 Time Step Size vs. Time Step 92

Fig. 6-41. Case 17 Time Step Size vs. Time Step 92

Fig. 6-42. Case 17 $\frac{(\delta\Omega)_{max}}{(\Omega_0)_{max}}$ Values vs. Time Step for Vorticity 93

Fig. 6-43. Case 17 $\frac{(\delta\Omega)_{max}}{(\Omega_0)_{max}}$ Values vs. Time Step for Streamfunction 93

Fig. 6-44. Case 17 $\frac{(\delta\Omega)_{max}}{(\Omega_0)_{max}}$ Values vs. Time Step for Temperature 94

Fig. 6-45. Case 17 $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ Values vs. Time Step for Vorticity 94

Fig. 6-46. Case 18 Time Step Size vs. Time Step 97

Fig. 6-47. Case 18 $\frac{(\delta\Omega)_{max}}{(\Omega_0)_{max}}$ Values vs. Time Step for Vorticity 97

Fig. 6-48. Case 18 $\frac{(\delta\Omega)_{max}}{(\Omega_0)_{max}}$ Values vs. Time Step for Streamfunction 98

Fig. 6-49. Case 18 $\frac{(\delta\Omega)_{max}}{(\Omega_0)_{max}}$ Values vs. Time Step for Temperature 98

Fig. 6-50. Case 19 Time Step Size vs. Time Step 100

Fig. 6-51. Case 19 $\frac{(\delta\Omega)_{max}}{(\Omega_0)_{max}}$ Values vs. Time Step for Vorticity 101

Fig. 6-52. Case 19 $\frac{(\delta\Omega)_{max}}{(\Omega_0)_{max}}$ Values vs. Time Step for Streamfunction 102

Fig. 6-53. Case 19 $\frac{(\delta\Omega)_{max}}{(\Omega_0)_{max}}$ Values vs. Time Step for Temperature 102

Fig. 6-54. Case 20 Time Step Size vs. Time Step 104

Fig. 6-55. Case 20 $\frac{(\delta\Omega)_{max}}{(\Omega_0)_{max}}$ Values vs. Time Step for Vorticity 105

Fig. 6-56. Case 20 $\frac{(\delta\Omega)_{max}}{(\Omega_0)_{max}}$ Values vs. Time Step for Streamfunction 106

Fig. 6-57. Case 20 $\frac{(\delta\Omega)_{max}}{(\Omega_0)_{max}}$ Values vs. Time Step for Temperature 106

Fig. 6-58. Case 21 Time Step Size vs. Time Step 108

Fig. 6-59. Case 21 $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ Values vs. Time Step for Vorticity 108

Fig. 6-60. Case 21 $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ Values vs. Time Step for Streamfunction 109

Fig. 6-61. Case 21 $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ Values vs. Time Step for Temperature 109

Fig. 6-62. Case 22 Time Step Size vs. Time Step 110

Fig. 6-63. Case 22 $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ Values vs. Time Step for Vorticity 111

Fig. 6-64. Case 22 $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ Values vs. Time Step for Streamfunction 111

Fig. 6-65. Case 22 $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ Values vs. Time Step for Temperature 112

Fig. 6-66. Case 23 Time Step Size vs. Time Step 116

Fig. 6-67. Case 23 $\frac{(\delta\Omega)_{max}}{(\Omega_0)_{max}}$ Values vs. Time Step for Velocity (U1) 116

Fig. 6-68. Case 23 $\frac{(\delta\Omega)_{max}}{(\Omega_0)_{max}}$ Values vs. Time Step for Velocity (U2) 117

Fig. 6-69. Case 23 $\frac{(\delta\Omega)_{max}}{(\Omega_0)_{max}}$ Values Between Time Steps 80 and 100 for Velocity (U1) 117

Fig. 6-70. Case 24 Time Step Size vs. Time Step 118

Fig. 6-71. Case 24 $\frac{(\delta\Omega)_{max}}{(\Omega_0)_{max}}$ Values vs. Time Step for Velocity (U1) 118

Fig. 6-72. Case 24 $\frac{(\delta\Omega)_{max}}{(\Omega_0)_{max}}$ Values vs. Time Step for Velocity (U2) 119

Fig. 6-73. Case 24 Time Step Size Between Time Steps 820 and 900 120

Fig. 6-74. Case 24 $\frac{(\delta\Omega)_{max}}{(\Omega_0)_{max}}$ Values Between Time Steps 820 and 900 for Velocity (U1) 120

Fig. 6-75. Case 24 $\frac{(\delta\Omega)_{max}}{(\Omega_0)_{max}}$ Values Between Time Steps 820 and 900 for Velocity (U2) 121

Fig. 6-76. Case 25 Time Step Size vs. Time Step 122

Fig. 6-77. Case 25 $\frac{(\delta\Omega)_{max}}{(\Omega_0)_{max}}$ Values vs. Time Step for Velocity (U1) 123

Fig. 6-78. Case 25 $\frac{(\delta Q)_{max}}{(Q_o)_{max}}$ Values vs. Time Step for Velocity (U2) 123

Fig. 6-79. Case 26 Time Step Size vs. Time Step 126

Fig. 6-80. Case 26 $\frac{(\delta Q)_{max}}{(Q_o)_{max}}$ Values vs. Time Step for Velocity (U1) 126

Fig. 6-81. Case 26 $\frac{(\delta Q)_{max}}{(Q_o)_{max}}$ Values vs. Time Step for Velocity (U2) 127

Fig. 6-82. Case 26 $\frac{(\delta Q)_{max}}{(Q_o)_{max}}$ Values vs. Time Step for Temperature 127

Fig. 6-83. Case 27 Time Step Size vs. Time Step 128

Fig. 6-84. Case 27 $\frac{(\delta Q)_{max}}{(Q_o)_{max}}$ Values vs. Time Step for Velocity (U1) 128

Fig. 6-85. Case 27 $\frac{(\delta Q)_{max}}{(Q_o)_{max}}$ Values vs. Time Step for Velocity (U2) 129

Fig. 6-86. Case 27 $\frac{(\delta Q)_{max}}{(Q_o)_{max}}$ Values vs. Time Step for Temperature 129

Fig. 6-87. Case 28 Time Step Size vs. Time Step 132

Fig. 6-88. Case 28 $\frac{(\delta Q)_{max}}{(Q_o)_{max}}$ Values vs. Time Step for Velocity (U1) 133

Fig. 6-89. Case 28 $\frac{(\delta Q)_{max}}{(Q_o)_{max}}$ Values vs. Time Step for Velocity (U2) 133

Fig. 6-90. Case 29 Time Step Size vs. Time Step 134

Fig. 6-91. Case 29 $\frac{(\delta Q)_{max}}{(Q_o)_{max}}$ Values vs. Time Step for Velocity (U1) 135

Fig. 6-92. Case 29 $\frac{(\delta Q)_{max}}{(Q_o)_{max}}$ Values vs. Time Step for Velocity (U2) 135

Fig. 6-93. Case 30 Time Step Size vs. Time Step 137

Fig. 6-94. Case 30 $\frac{(\delta Q)_{max}}{(Q_o)_{max}}$ Values vs. Time Step for Velocity (U1) 137

Fig. 6-95. Case 30 $\frac{(\delta Q)_{max}}{(Q_o)_{max}}$ Values vs. Time Step for Velocity (U2) 138

Fig. 6-96. Case 31 Time Step Size vs. Time Step 139

Fig. 6-97. Case 31 $\frac{(\delta Q)_{max}}{(Q_o)_{max}}$ Values vs. Time Step for Velocity (U1) 139

Fig. 6-98. Case 31 $\frac{(\delta Q)_{max}}{(Q_o)_{max}}$ Values vs. Time Step for Velocity (U2) 140

Fig. 6-99. Case 32 Time Step Size vs. Time Step 143

Fig. 6-100. Case 32 $\frac{(\delta Q)_{max}}{(Q_s)_{max}}$ Values vs. Time Step for Velocity (U1) 143

Fig. 6-101. Case 32 $\frac{(\delta Q)_{max}}{(Q_s)_{max}}$ Values vs. Time Step for Velocity (U2) 144

Fig. 6-102. Case 32 $\frac{(\delta Q)_{max}}{(Q_s)_{max}}$ Values vs. Time Step for Temperature 144

Fig. 6-103. Case 33 Time Step Size vs. Time Step 145

Fig. 6-104. Case 33 $\frac{(\delta Q)_{max}}{(Q_s)_{max}}$ Values vs. Time Step for Velocity (U1) 146

Fig. 6-105. Case 33 $\frac{(\delta Q)_{max}}{(Q_s)_{max}}$ Values vs. Time Step for Velocity (U2) 146

Fig. 6-106. Case 33 $\frac{(\delta Q)_{max}}{(Q_s)_{max}}$ Values vs. Time Step for Temperature 147

Fig. 6-107. Case 34 Time Step Size vs. Time Step 148

Fig. 6-108. Case 34 $\frac{(\delta Q)_{max}}{(Q_s)_{max}}$ Values vs. Time Step for Velocity (U1) 149

Fig. 6-109. Case 34 $\frac{(\delta Q)_{max}}{(Q_s)_{max}}$ Values vs. Time Step for Velocity (U2) 149

Fig. 6-110. Case 34 $\frac{(\delta Q)_{max}}{(Q_s)_{max}}$ Values vs. Time Step for Temperature 150

Fig. 6-111. Case 35 Time Step Size vs. Time Step 151

Fig. 6-112. Case 35 $\frac{(\delta Q)_{max}}{(Q_s)_{max}}$ Values vs. Time Step for Velocity (U1) 151

Fig. 6-113. Case 35 $\frac{(\delta Q)_{max}}{(Q_s)_{max}}$ Values vs. Time Step for Velocity (U2) 152

Fig. 6-114. Case 35 $\frac{(\delta Q)_{max}}{(Q_s)_{max}}$ Values vs. Time Step for Temperature 152

Fig. B-1. Driven Cavity 174

Fig. B-2. Thermal Cavity 176

Chapter 1

Introduction

An important characteristic of ordinary differential equation (ODE) numerical integration algorithms is stability. ODE algorithm stability is determined by the behavior of a solution as $t \rightarrow \infty$, while keeping the integration time step size (Δt) constant. If the error generated in the ODE solution (excluding round-off error) is independent of the time step size, the numerical time-integration algorithm is termed "unconditionally" stable. If the error generated in the solution is dependent on the time step size, the ODE algorithm is "conditionally" stable (Burnette, 1987).

The computer central processing unit (CPU) time required to reach a steady-state solution is dependent on many factors, premier among which is the time step size of the numerical time-integration algorithm. Unconditionally stable methods will eventually reach the steady-state solution; however, choice of a too-large time step could cause the

solution to oscillate over many time steps with very slow decay in the oscillations. Conditionally stable methods may never reach a steady-state solution if an improper time step is used. Therefore, it is important that an automated, on-line stability and time step control method be implemented when coding a computational fluid dynamics (CFD) algorithm.

The goal of the research presented in this thesis is to implement a means to monitor the stability of a class of CFD algorithms for fluid flow problems. Specifically, it is desired to automatically adjust the time step to the largest possible value that will maintain a stable solution as the numerical time-integration algorithm approaches the steady state. Toward these ends, this thesis is organized as follows:

- Chapter 2 provides a synopsis of the governing principles, hence equations of fluid flow that are the basis for the ODE system eventually solved by a numerical time-integration algorithm.
- Chapter 3 summarizes the finite element approximation technique for the weak statement formulation of the governing equations leading to the numerical time-integration algorithm.
- Chapter 4 provides background theory on ODE algorithm stability, emphasizing aspects pertinent to the weak statement algorithms expressed in Chapter 3.
- Chapter 5 details the implementation of the stability theory discussed in Chapter 4 into the *AKCESS.* * software platform utilizing a "template" and "hook" format for implementing discrete algorithm variations.

- Chapter 6 presents the effects of the implemented stability/time step controller on the convergence and stability of select benchmark problems for several CFD algorithm forms.

The utopian goal would be to sufficiently automate the time step controller so that any problem would run stably and to steady state with minimal CPU time without any interaction by the code user. Such an achievement appears impractical for the real world that has such a wide variety of unique problems and circumstances to overcome. Therefore, a more realistic approach is to develop a time step controller that will maintain stability with a minimum of a user's involvement. Although much emphasis is placed on the cost of CPU time (with perhaps valid reason), it is the experience of the author that personnel costs, project schedules, and the amount of user aggravation can be much more important. In the current age of powerful, localized workstations, CPU time really does not have a cost (unless the purchase price of the workstation is considered). Thus, an effective time step controller should enable a problem solution to reach steady-state in a reliable, relatively efficient manner, but not necessarily in the shortest possible CPU time.

It is hoped that the results of this research will assist users of CFD codes in achieving efficient, cost-effective solutions of real problems with minimal interaction, therefore minimal frustration, with the computer.

Chapter 2

Conservation Law Statements in Fluid Mechanics

Before getting into the details of time step control, stability theory, and the like, it is important to understand some of the fundamentals underlying CFD systems. This understanding will begin by a discussion of the conservation law statements for fluid flow.

The flow of fluids is described by the solution of partial differential equations derived from select physical principles. These basic equations, presented in many fluid dynamics and continuum mechanics textbooks [e.g., (Knudzen and Katz, 1958), (White, 1974), (Fung, 1977), and (Malvern, 1969)], are

- the continuity equation, based on conservation of mass,
- the momentum equations, based on Newton's second law of motion, and
- the energy equation, based on the first law of thermodynamics.

A brief development of these equations is provided herein. Because the problems analyzed later in Chapter 6 are for two-dimensional (2D) flow, the discussions to follow will focus on the 2D form of these equations.

2.1 The Continuity Equation

The continuity equation is based on the principle of conservation of mass; i.e., for a particle or rigid body in motion, the mass of the body remains unchanged during the motion. Transforming to a system of particles, then using Reynolds transport theorem, the continuity equation for flow of a compressible fluid is

- Indicial notation:

$$\frac{D\rho}{Dt} + \rho\frac{\partial u_j}{\partial x_j} = \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j) = 0 \qquad (2\text{-}1a)$$

- Vector notation:

$$\frac{D\rho}{Dt} + \rho\nabla\cdot\mathbf{U} = \frac{\partial \rho}{\partial t} + \nabla\cdot(\rho\mathbf{U}) = 0 \qquad (2\text{-}1b)$$

- Scalar notation (2D):

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x} + \frac{\partial (\rho v)}{\partial y} = 0 \qquad (2\text{-}1c)$$

where ρ is the density; u_j or **U** is the velocity vector field, with u being the x-component of **U** and v being the y-component of **U**; and t is the time. The term $\frac{D(\star)}{Dt}$ is the material derivative representing $\frac{\partial(\star)}{\partial t} + u_j \frac{\partial(\star)}{\partial x_j}$.

If the fluid is "incompressible," i.e., its density is constant, the continuity equation forms reduce to

- Indicial notation:

$$\frac{\partial u_j}{\partial x_j} = 0 \qquad (2\text{-}2a)$$

- Vector notation:

$$\nabla \bullet \mathbf{U} = 0 \qquad (2\text{-}2b)$$

- Scalar notation (2D):

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0. \qquad (2\text{-}2c)$$

2.2 The Momentum Equations

The momentum equations are derived from Newton's second law of motion; i.e., the principle that states that the time rate of change of momentum of a fluid particle is equal to the external forces on the fluid particle. The general form of the momentum equations, obtained by transforming to a system of particles and using Reynolds transport theorem is

- Indicial notation:

$$\rho\left(\frac{\partial u_i}{\partial t} + u_j\frac{\partial u_i}{\partial x_j}\right) + \frac{\partial P}{\partial x_i} - \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \lambda\frac{\partial u_j}{\partial x_i}\right] - \rho b_i = 0 \qquad (2\text{-}3a)$$

- Vector notation:

$$\rho\frac{D\mathbf{U}}{Dt} + \nabla P - (\lambda + \mu)\nabla(\nabla\bullet\mathbf{U}) - \mu\nabla^2\mathbf{U} - \rho\boldsymbol{b} = 0 \qquad (2\text{-}3b)$$

- Scalar notation (2D):

$$\rho\frac{Du}{Dt} + \frac{\partial P}{\partial x} - \frac{\partial}{\partial x}\left[2\mu\frac{\partial u}{\partial x} + \lambda\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)\right] - \mu\frac{\partial}{\partial y}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) - \rho b_x = 0 \tag{2-3c}$$

$$\rho\frac{Dv}{Dt} + \frac{\partial P}{\partial y} - \frac{\partial}{\partial y}\left[2\mu\frac{\partial v}{\partial y} + \lambda\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)\right] - \mu\frac{\partial}{\partial x}\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right) - \rho b_y = 0,$$

where P is the pressure, μ is the first coefficient of viscosity (relating to the concept of one layer of a fluid "sliding" over another), λ is the second coefficient of viscosity (or bulk viscosity relating to the forces of expansion and contraction of a fluid), and $\boldsymbol{b}$ is an external force, e.g., gravity.

Applying Stokes' hypothesis [(White, 1974) and (Malvern, 1969)], i.e., $\lambda = -\frac{2\mu}{3}$, to Eq. (2-3) gives the usual form of the compressible Navier-Stokes equations

- Indicial notation:

$$\rho\left(\frac{\partial u_i}{\partial t} + u_j\frac{\partial u_i}{\partial x_j}\right) + \frac{\partial P}{\partial x_i} - \frac{\mu}{3}\frac{\partial}{\partial x_i}\left(\frac{\partial u_j}{\partial x_j}\right) + \mu\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}\right) - \rho b_i = 0 \qquad (2\text{-}4a)$$

- Vector notation:

$$\rho\frac{D\mathbf{U}}{Dt} + \nabla P - \frac{\mu}{3}\nabla(\nabla\bullet\mathbf{U}) - \mu\nabla^2\mathbf{U} - \rho\boldsymbol{b} = 0 \qquad (2\text{-}4b)$$

- Scalar notation (2D):

$$\rho\frac{Du}{Dt} + \frac{\partial P}{\partial x} - \frac{\mu}{3}\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) - \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) - \rho b_x = 0 \qquad (2\text{-}4c)$$

$$\rho\frac{Dv}{Dt} + \frac{\partial P}{\partial y} - \frac{\mu}{3}\frac{\partial}{\partial y}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) - \mu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) - \rho b_y = 0.$$

Further simplification of the Navier-Stokes equations can be made if the fluid is incompressible. Applying Eq. (2-2) to Eq. (2-4) gives the usual form of the incompressible Navier-Stokes (INS) equations

- Indicial notation:

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} + \frac{1}{\rho_0}\frac{\partial P}{\partial x_i} - \nu \frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}\right) - \frac{\rho}{\rho_0} b_i = 0 \qquad (2\text{-}5a)$$

- Vector notation:

$$\frac{D\mathbf{U}}{Dt} + \frac{1}{\rho_0}\nabla P - \nu \nabla^2 \mathbf{U} - \frac{\rho}{\rho_0}\boldsymbol{b} = 0 \qquad (2\text{-}5b)$$

- Scalar notation (2D):

$$\frac{Du}{Dt} + \frac{1}{\rho_0}\frac{\partial P}{\partial x} - \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) - \frac{\rho}{\rho_0} b_x = 0$$

$$\frac{Dv}{Dt} + \frac{1}{\rho_0}\frac{\partial P}{\partial y} - \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) - \frac{\rho}{\rho_0} b_y = 0, \qquad (2\text{-}5c)$$

where ρ_0 is the constant density and $\nu = \frac{\mu}{\rho_0}$ is the kinematic viscosity.

2.3 The Energy Equation

The energy equation is based on the first law of thermodynamics; i.e., the principle that states that for a body in motion, the total energy of the body remains unchanged during the motion. For constant thermodynamic properties, the energy equation, obtained by transforming to a system of particles and using the Reynolds transport theorem, is

- Indicial notation:

$$\rho c_p \left(\frac{\partial T}{\partial t} + u_j \frac{\partial T}{\partial x_j} \right) - k \frac{\partial^2 T}{\partial x_j^2} - \Phi - s = 0 \qquad \text{(2-6a)}$$

- Vector notation:

$$\rho c_p \frac{DT}{Dt} - k \nabla^2 T - \Phi - s = 0 \qquad \text{(2-6b)}$$

- Scalar notation (2D):

$$\rho c_p\left(\frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y}\right) - k\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right) - \Phi - s = 0 \tag{2-6c}$$

where T is the temperature, k is the thermal conductivity, s is the source (e.g., distributed heat generation), and Φ is the dissipation function defined in 2D as (White, 1979)

$$\Phi = \mu\left[2\frac{\partial^2 u}{\partial x^2} + 2\frac{\partial^2 v}{\partial y^2} + \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)^2\right] = \mu\frac{\partial u_i}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right). \tag{2-7}$$

2.4 Dimensionless Forms

The continuity, momentum, and energy (the Navier-Stokes) equations are dimensional in form. Converting these equations to dimensionless form can provide a means to scale results to correlate with experimental data and provide insight into the relative importance of the dimensionless parameters.

To convert the continuity, momentum, and energy equations to dimensionless form, the following relationships between dimensionless parameters (denoted by an asterisk) and dimensional parameters are used:

$$x^* = \frac{x}{L} \qquad \rho^* = \frac{\rho}{\rho_0} \qquad P^* = \frac{P + \rho_0 g z - P_0}{\rho_0 U_0^2}$$

$$U^* = \frac{U}{U_0} \qquad t^* = \frac{t U_0}{L} \qquad T^* = \frac{(T - T_0)}{(T_w - T_0)} \tag{2-8}$$

$$\Phi^* = \frac{\Phi L^2}{\mu_0 U_0^2} \qquad \nabla^* = \frac{\nabla}{L} \qquad s^* = \frac{sL}{U_0 \rho_0 c_{p_0}(T_w - T_0)}$$

Substituting the dimensionless variables into Eqs. (2-1), (2-2), (2-4), (2-5) and (2-6) gives the following dimensionless forms for the Navier-Stokes equations:

- Dimensionless continuity equation for compressible fluid:

$$\frac{\partial \rho^*}{\partial t^*} + \nabla^* \cdot (\rho^* \mathbf{U}^*) = 0 \tag{2-9}$$

- Dimensionless continuity equation for incompressible fluid:

$$\nabla^* \cdot \mathbf{U}^* = 0 \tag{2-10}$$

- Dimensionless momentum equations for compressible fluid:

$$\frac{D\mathbf{U}^*}{Dt^*} + \nabla^* P^* - \frac{1}{3Re}\nabla^*(\nabla^* \cdot \mathbf{U}^*) - \frac{1}{Re}\nabla^{*2}\mathbf{U}^* + \frac{Gr}{Re^2} T^* \mathbf{g}^* = 0 \tag{2-11}$$

- Dimensionless momentum equations for incompressible fluid:

$$\frac{D\mathbf{U}^*}{Dt^*} + \nabla^* P^* - \frac{1}{Re}\nabla^{*2}\mathbf{U}^* + \frac{Gr}{Re^2} T^* \mathbf{g}^* = 0 \tag{2-12}$$

- Dimensionless energy equation:

$$\frac{DT^*}{Dt^*} - \frac{1}{Pe}\nabla^{*2} T^* - \frac{Ec}{Re}\Phi^* - s^* = 0. \tag{2-13}$$

In Eqs. (2-11) and (2-12), $\mathbf{g}^*$ is the gravity unit vector. The introduced dimensionless groups *Re*, *Pe*, *Ec*, and *Gr* are the Reynolds, Peclet, Eckert, and Grashof numbers, respectively, defined as

$$Re = \frac{\rho_0 U_0 L}{\mu_0} \qquad Pe = \frac{\rho_0 U_0 L c_{p_0}}{k_0}$$

$$Ec = \frac{U_0^2}{c_{p_0}(T_w - T_0)} \qquad Gr = \frac{g\beta_0 \rho_0^2 L^3 (T_w - T_0)}{\mu_0}. \tag{2-14}$$

2.5 Vorticity-Streamfunction Formulation

The vorticity-streamfunction rearrangement for the incompressible Navier-Stokes (INS) equations eliminates the appearance of pressure and exactly enforces continuity via identification of a vector potential function. This transformation provides a convenient form of the INS equations for computing, that will be used for the benchmark driven cavity and thermal cavity problems (Chapter 6) to test out time step control methods. In 2D, the basic premise is to introduce two new varaibles; the first of which is the ***k***-component of a vector potential function

$$\mathbf{U} = \nabla \times \psi \mathbf{k} \tag{2-15}$$

Eq. (2-15) satisfies the incompressible continuity equation [Eq. (2-2)] as demonstrated in Eq. (2-16) [Recall that $\mathbf{i}\cdot\mathbf{k} = \mathbf{j}\cdot\mathbf{k} = 0$]:

$$\begin{aligned} \nabla\cdot\mathbf{U} &\equiv \nabla\cdot\nabla\times\psi\mathbf{k} \\ &= \left(\frac{\partial}{\partial x}\mathbf{i} + \frac{\partial}{\partial y}\mathbf{j}\right)\cdot\left(\frac{\partial v}{\partial x}\mathbf{k} + \frac{\partial u}{\partial y}\mathbf{k}\right) \equiv 0. \end{aligned} \tag{2-16}$$

The second variable is the **k**-component of the vorticity vector

$$\omega \equiv \nabla\times\mathbf{U}\cdot\mathbf{k} \tag{2-17}$$

From vector field theory, the curl of the incompressible momentum equations provides an alternative means to characterize **U**. Substituting Eqs. (2-15) and (2-17) into Eq. (2-12) gives the vorticity transport equation:

$$\frac{\partial\omega}{\partial t} + \nabla\times\psi\mathbf{k}\cdot\nabla\omega - \frac{1}{Re}\nabla^2\omega + \frac{Gr}{Re^2}\nabla\times T\mathbf{g}^*\cdot\mathbf{k} = 0. \tag{2-18}$$

The pressure term in Eq. (2-12) was eliminated based on the following identity (in 2D):

$$\nabla\times\nabla P = \epsilon_{ij3}\frac{\partial}{\partial x_i}\left(\frac{\partial P}{\partial x_j}\right) = \frac{\partial^2 P}{\partial x\partial y} - \frac{\partial^2 P}{\partial y\partial x} = 0. \tag{2-19}$$

Taking the curl of the vector potential function [Eq. (2-15)] and rearranging gives the kinematic compatibility equation that relates the streamfunction ψ to the vorticity ω:

$$\nabla^2 \psi + \omega = 0. \tag{2-20}$$

The resulting vorticity-streamfunction formulation of the INS equations is the combination of Eqs. (2-20), (2-18), and (2-13):

- Kinematic compatibility equation

$$\nabla^2 \psi + \omega = 0 \tag{2-21a}$$

- Vorticity transport equation

$$\frac{\partial \omega}{\partial t} + \nabla \times \psi \mathbf{k} \bullet \nabla \omega - \frac{1}{Re} \nabla^2 \omega + \frac{Gr}{Re^2} \nabla \times T\mathbf{g}^{*} \bullet \mathbf{k} = 0 \tag{2-21b}$$

- Energy equation

$$\frac{DT}{Dt} - \frac{1}{Pe} \nabla^2 T - \frac{Ec}{Re} \Phi - s = 0. \tag{2-21c}$$

Chapter 3

Weak Statement Algorithm

To solve the equations defined in Chapter 2, an approximation technique is required. The weak statement algorithm coupled with the finite element method (FEM) provides a well-ordered means to obtain approximate numerical solutions to equations representing physical problems. Any approximation can be expressed using a trial solution in the form of a Taylor series. Letting $q(\mathbf{x},t)$ represent the INS state-variable, e.g., (u, v, P, T), (ω, ψ, T), then its approximation q^N is expressible as

$$q(\mathbf{x},t) \approx q^N(\mathbf{x},t) = \sum_{i=1}^{N} \Phi_i(\mathbf{x}) Q_i(t), \tag{3-1}$$

where $\mathbf{x}$ represents all of the independent spatial variables in the system of equations and t is time. $Q_i(t)$ represents the approximation degrees of freedom, i.e., the unknown,

time-dependent expansion coefficients and $\Phi_i(\mathbf{x})$ represents the {known) trial function set supporting the approximation. The choice of $\Phi_i(\mathbf{x})$ ultimately determines how close the approximation can be to the exact solution, once the $Q_i(t)$ have been determined.

The difference between the exact solution and any approximate solution constitutes the error, $e^N(\mathbf{x},t)$, i.e.,

$$e^N(\mathbf{x},t) = q(\mathbf{x},t) - q^N(\mathbf{x},t). \tag{3-2}$$

In the sense of distributions, the weak statement (WS) seeks to minimize this error by requiring (Baker, 1991)

$$WS = \int_\Omega w_i(\mathbf{x})\, \mathcal{L}(q^N) dx = 0, \qquad for\ 1 \le i \le N, \tag{3-3}$$

where $\mathcal{L}(q^N)$ denotes any selected INS PDE system, $w_i(\mathbf{x})$ is the weight (or test) function set, and Ω is the problem domain. The Galerkin weak statement criterion requires the trial function set to be the same as the weight function set, hence

$$\Phi_i(\mathbf{x}) = w_i(\mathbf{x}). \tag{3-4}$$

Selecting the Galerkin criterion ensures that the approximation error is minimized, since the error is orthogonal to the trial function set supporting the approximate solution. It is this Galerkin method that is traditionally used to develop a finite element approximation, hence, a computable CFD algorithm.

3.1 Galerkin Weak Statement

The vorticity-streamfunction form of the INS equations was previously defined by Eqs. (2-21a), (2-21b), and (2-21c). Using it as an example, and for isothermal (i.e., nonvarying temperature) conditions, the energy equation [Eq. (2-21c)] and the buoyancy term in Eq. (2-21b) are ignored, leaving the following coupled PDE system:

$$\mathcal{L}(\omega) = \frac{\partial \omega}{\partial t} + (\nabla \times \psi \boldsymbol{k} \cdot \nabla)\,\omega - \frac{1}{Re}\nabla^2 \omega = 0 \qquad (3\text{-}5a)$$

$$\mathcal{L}(\psi) = \nabla^2 \psi + \omega = 0. \qquad (3\text{-}5b)$$

The state variable is now $q(\mathbf{x},t) = \{\omega, \psi\}^T$. For $\mathcal{L}(q)$ denoting the Eq. (3-5) system, the weak statement is

$$WS = \int_{\Omega} w_i(\mathbf{x})\, \mathcal{L}(q^N) d\tau = 0, \qquad 1 \le i \le N. \tag{3-6}$$

Using the Taylor series approximation given in Eq. (3-1), then $Q_i = \{\Omega_i, \Psi_i\}^T$ are the time-dependent expansion coefficients to be determined. Enforcing the Galerkin criterion [Eq. (3-4)] on Eq. (3-5a), the Galerkin weak statement for the vorticity transport equation is

$$\begin{aligned} GWS(\omega^N) &= \int_{\Omega} \Phi_i(\mathbf{x})\, \mathcal{L}(\omega^N) d\tau = 0, \qquad for\ 1 \le i \le N \\ &= \int_{\Omega} \Phi_i(\mathbf{x}) \left[\frac{\partial \omega^N}{\partial t} + (\nabla \times \psi \mathbf{k} \cdot \nabla)\, \omega^N - \frac{1}{Re} \nabla^2 \omega^N \right] d\tau \\ &= 0. \end{aligned} \tag{3-7}$$

Similarly, the Galerkin weak statement form for the kinematic compatibility equation is

$$\begin{aligned} GWS(\psi^N) &= \int_{\Omega} \Phi_i(\mathbf{x})\, \mathcal{L}(\psi^N) d\tau = 0, \qquad for\ 1 \le i \le N \\ &= \int_{\Omega} \Phi_i(\mathbf{x}) \left[\nabla^2 \psi^N + \omega^N \right] d\tau = 0. \end{aligned} \tag{3-8}$$

3.2 Finite Element Implementation

The finite element method uses a discretization Ω^h of the domain Ω made up of the union of a set of non-overlapping elements Ω_e such that

$$\Omega \approx \Omega^h = \bigcup_e \Omega_e, \tag{3-9}$$

Thereupon, an (any) approximation q^N is constructed as the union (non-overlapping sum) of piecewise-continuous functional expressions on the finite elements Ω_e of Ω^h in the form

$$q(\mathbf{x},t) \approx q^N(\mathbf{x},t) = q^h(\mathbf{x},t) = \bigcup_e q_e(\mathbf{x},t), \tag{3-10}$$

where superscript "h" denotes use of Ω^h. The generic form for q_e on any Ω_e is

$$q_e(\mathbf{x},t) = \{N_k(\mathbf{x})\}^T \{Q(t)\}_e, \tag{3-11}$$

where $\{N_k(\mathbf{x})\}$, called the finite element basis, constitutes the form of $\Phi_i(\mathbf{x})$ on Ω_e, the selection of which is usually a k^{th} degree polynomial.

Applying the approximation of Eqs. (3-10) and (3-11) to Eqs. (3-7) and (3-8), replacing the trial solution $\Phi_i(\mathbf{x})$ with $\{N_k(\mathbf{x})\}$, and rearranging gives

$$\begin{aligned} GWS(\omega^h) \;=\; S_e\Bigg[& \int_{\Omega_e} \{N_k\}\{N_k\}^T \frac{d\{\Omega\}_e}{dt} d\tau \\ & + \int_{\Omega_e} \{N_k\}\nabla\times\{N_k\}^T\{\Psi\}_e k\bullet\nabla\{N_k\}^T\{\Omega\}_e d\tau \\ & + \frac{1}{Re}\int_{\Omega_e} \nabla\{N_k\}\bullet\nabla\{N_k\}^T\{\Omega\}_e d\tau + \int_{\partial\Omega_e}\cdots \Bigg] \\ \equiv\; S_e\left([M]_e \frac{d\{\Omega\}_e}{dt} + [U]_e\{\Omega\}_e + \frac{1}{Re}[K]_e\{\Omega\}_e \right) \;=\; & \{0\} \end{aligned} \tag{3-12}$$

$$\begin{aligned} GWS(\psi^h) \;=\; S_e\Bigg[& \int_{\Omega_e} \nabla\{N_k\}\bullet\nabla\{N_k\}^T\{\Psi\}_e d\tau \\ & + \int_{\Omega_e} \{N_k\}\{N_k\}^T\{\Omega\}_e d\tau \Bigg] \\ \equiv\; S_e\left([D]_e\{\Psi\}_e + \{b(\Omega)\}_e \right) \;=\; & \{0\}, \end{aligned} \tag{3-13}$$

where S_e denotes an assembly, i.e., a summation of the expressions at the element level. The last term in Eq. (3-12) allows for insertion of boundary conditions of critical importance to be discussed.

Selecting a specific basis $\{N_k\}$, the integral calculus operations in Eq. (3-12) can be completed, creating a global matrix statement of the form

$$\begin{aligned} GWS^h &= [M]\frac{d\{\Omega\}}{dt} + [U+K]\{\Omega\} \\ &= [M]\{Q\}' + \{R(Q)\} = \{0\}. \end{aligned} \tag{3-14}$$

Forming a time Taylor series for $\{\Omega\}$ about t_n and t_{n+1}, and substituting Eq. (3-14) as required, yields a fully algebraic statement

$$\begin{aligned} \{F(\Omega)\} &= [M]\{\Omega_{n+1} - \Omega_n\} + \Delta t\Big(\theta\{R\}_{n+1} + (1-\theta)\{R\}_n\Big) \\ &= \{0\} \end{aligned} \tag{3-15}$$

where $0 \le \theta \le 1$ is the implicitness parameter in the Taylor series and $\{R\} = [U+K]\{\Omega\}$. For $\mathcal{L}(\psi)$, the WS^h directly produces the algebraic statement shown in Eq. (3-13)

$$\{F(\Psi)\} = [D]\{\Psi\} - \{b(\Omega)\} = \{0\}. \tag{3-16}$$

where $[D]$ is the discrete equivalent of the laplacian.

Eqs. (3-15) and (3-16) are a non-linear coupled algebraic equation system for the approximation coefficient set $\{\Omega, \Psi\}^T$, which depend on time. The generic solution statement is a Newton algorithm, i.e.,

$$[JAC]_{n+1}^{p+1} \{\delta Q\}^{p+1} = - \{F(Q)\}_{n+1} \tag{3-17}$$

where $\{Q\} = \{\Omega, \Psi\}^T$, $p = 0, 1, 2, \ldots$ in the iteration cycle until convergence, the jacobian $[JAC]$ is defined as

$$\begin{aligned} [JAC] &= \frac{\partial \{F(Q)\}}{\partial \{Q\}} \\ &= \begin{bmatrix} \dfrac{\partial \{F(\Omega)\}}{\partial \Omega} & , & \dfrac{\partial \{F(\Omega)\}}{\partial \Psi} \\ \dfrac{\partial \{F(\Psi)\}}{\partial \Omega} & , & \dfrac{\partial \{F(\Psi)\}}{\partial \Psi} \end{bmatrix} = \begin{bmatrix} J\Omega\Omega & , & J\Omega\Psi \\ J\Psi\Omega & , & J\Psi\Psi \end{bmatrix}, \end{aligned} \tag{3-18}$$

and

$$\begin{aligned} \{Q\}_{n+1}^{p+1} &= \{Q\}_{n+1}^{p} + \{\delta Q\}^{p+1} \\ &= \{Q\}_n + \sum_{i=1}^{p} \{\delta Q\}^{i+1}. \end{aligned} \tag{3-19}$$

The *GWS*[4] finite element method, therefore, produces a coupled system of equations that are solved by an iterative process. Generalizing for a non-isothermal statement, the Newton algorithm replacement of Eq. (3-17) is

$$\begin{bmatrix} J\Omega\Omega & , & J\Omega\Psi & , & J\Omega T \\ J\Psi\Omega & , & J\Psi\Psi & , & 0 \\ 0 & , & JT\Psi & , & JTT \end{bmatrix} \begin{Bmatrix} \delta\Omega \\ \delta\Psi \\ \delta T \end{Bmatrix}^{p+1} = - \begin{Bmatrix} F\Omega \\ F\Psi \\ FT \end{Bmatrix}_{n+1} . \qquad (3\text{-}20)$$

Recognizing the zeroes in Eq. (3-20), the temperature solution can be segregated (i.e., decoupled) from the vorticity and streamfunction solution process, thereby yielding a quasi-Newton algorithm. The algebraic statements are:

$$[JTT]\ \{\delta T\}^{p+1} = - \{FT\}_{n+1}$$

$$\begin{bmatrix} J\Omega\Omega & J\Omega\Psi \\ J\Omega\Psi & J\Psi\Psi \end{bmatrix} \begin{Bmatrix} \delta\Omega \\ \delta\Psi \end{Bmatrix}^{p+1} = - \begin{Bmatrix} F\Omega \\ F\Psi \end{Bmatrix}_{n+1} \qquad (3\text{-}22)$$

and the iterative convergence characteristics are therefore non-Newton. The Newton and quasi-Newton algorithms are both used for select test cases discussed in Chapter 6.

3.3 Boundary Conditions

The vorticity-streamfunction-temperature PDE system is elliptic boundary value; thus, select Dirichlet and/or Neumann data are required all around the boundary. Baker (1983) provides a detailed discussion of the boundary conditions appropriate for the selected benchmark cavity problems.

Chapter 4

Algorithm Stability Theory

One of the most important characteristics of ODE numerical time-integration algorithms is stability. ODE algorithm stability is determined by the behavior of a solution as $t \rightarrow \infty$ while keeping the time step (Δt) constant. As shown in Fig. 4-1(a), for $\Delta t >$ some limit, solution instability may exhibit oscillations that increase in amplitude as the time progresses until eventually the solution "blows up," hence becomes meaningless. On the other hand, stable solution processes can exhibit oscillations that decrease in amplitude (i.e., decay) as time progresses [Fig. 4-1(b)], or exhibit no oscillations at all [Fig. 4-1(c)]. It is these solution characteristics that are discussed in more detail later in this chapter, and are examined in specific test problems in Chapter 6.

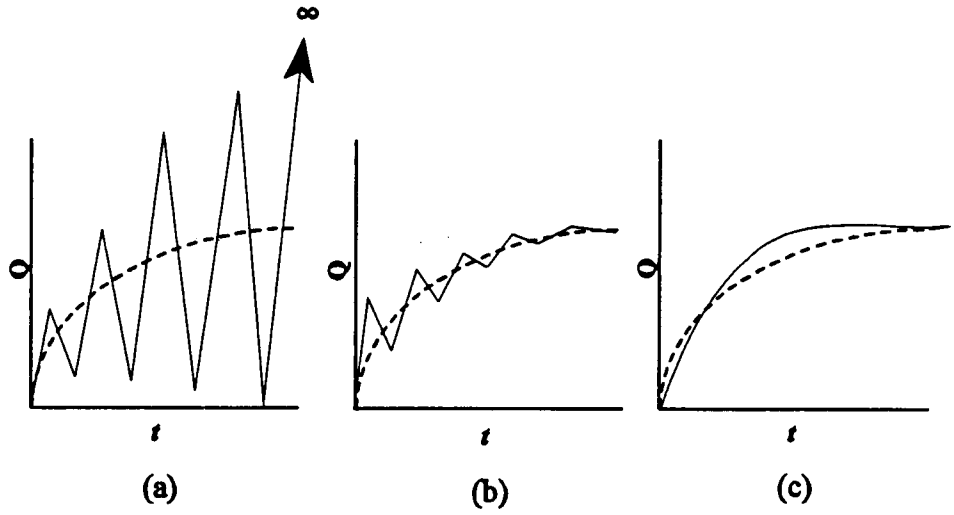


Fig. 4-1. Solution Convergence Characteristics Schematic

4.1 Conditions for Stability

To examine stable or unstable behavior, a multi-degree-of-freedom (DOF) system of ODEs of the form

$$[M]\frac{d\{Q(t)\}}{dt} + [K]\{Q(t)\} = \{b(t)\}, \qquad t > t_0, \tag{4-1}$$

is considered [(Burnette, 1987) and (Davies, 1980)]. The free response of the system, i.e., when the applied load $\{b(t)\}$ is zero, is governed as:

$$[M]\{Q(t)\}' + [K]\{Q(t)\} = 0. \tag{4-2}$$

Note that Eq. (4-2) is the same form as that resulting for the weak statement formulation of the vorticity equation in the vorticity-streamfunction algorithm described in Chapter 3.

A solution of the free response having the form

$$\{Q(t)\} = \{v\}\, e^{-\lambda_i t}, \tag{4-3}$$

provides considerable insight regarding the vorticity-streamfunction system. Substituting Eq. (4-3) into Eq. (4-2) gives

$$\Big([K] - \lambda_i[M]\Big)\{v\}\, e^{-\lambda_i t} = \{0\}, \tag{4-4}$$

which has nontrivial solutions $\{v\}$ only when the determinant of the coefficient matrix vanishes, i.e.,

$$\det\Big|\, [K] - \lambda_i[M] \,\Big| = 0. \tag{4-5}$$

The general solution to Eq. (4-2) is a sum of solutions of the form of Eq. (4-3), i.e.,

$$\{Q(t)\} = \sum_{j=1}^{J} Q_j \{v\}_j\, e^{-\lambda_j t}, \tag{4-6}$$

where J is the number of modes (i.e., eigenvectors) and Q_j is a set of coefficients determined from initial conditions. Similarly, the general solution of the forced problem [Eq. (4-1)] is also a sum of the modes:

$$\{Q(t)\} = \sum_{j=1}^{J} Q_j(t) \{v\}_j . \tag{4-7}$$

If Eq. (4-7) is substituted into the forced problem [Eq. (4-1)], the result is

$$[M]\left(\sum_{j=1}^{J} \frac{dQ_j(t)}{dt} \{v\}_j\right) + [K]\left(\sum_{j=1}^{J} Q_j(t) \{v\}_j\right) = \{b(t)\}. \tag{4-8}$$

Each of the modes $\{v\}_j$ satisfy the following orthogonality relations (Burnette, 1987):

$$\left.\begin{aligned} \{v\}_j^T [K] \{v\}_i &= \lambda_j \delta_{ij} \\ \{v\}_j^T [M] \{v\}_i &= \delta_{ij} \end{aligned}\right\}, \tag{4-9}$$

where δ_{ij} is the Kronecker delta. Substituting the orthogonality relations [Eq. (4-9)] into Eq. (4-8) yields

$$\frac{dQ_i(t)}{dt} + \lambda_i Q_i(t) = f_i(t), \qquad i = 1, 2, \ldots, J, \tag{4-10}$$

where $f_i(t) = \{v\}_i^T \{b(t)\}$. Eq. (4-10) is the effective 1-DOF system for each mode.

The recurrence relation corresponding to Eq. (4-10), i.e., the algebraic equation that relates the values $\{Q\}_n$ at two or more successive times, is determined using the θ-method (Burnette, 1987) whereby expressions of $\left\{\frac{dQ}{dt}\right\}$, $\{Q\}$, and $\{f\}$ are obtained by approximating $\{Q(t)\}$ and $\{f(t)\}$ by linear interpolation polynomials over the step. In particular,

$$\{Q\}_\theta = (1 - \theta)\{Q\}_{n-1} + \theta\{Q\}_n, \tag{4-11a}$$

$$\{f\}_\theta = (1 - \theta)\{f\}_{n-1} + \theta\{f\}_n, \text{ and} \tag{4-11b}$$

$$\left\{\frac{dQ}{dt}\right\}_\theta = \frac{1}{\Delta t_n}\frac{d\{Q\}_\theta}{d\theta} = \frac{\{Q\}_n - \{Q\}_{n-1}}{\Delta t_n}. \tag{4-11c}$$

Substituting Eqs. (4-11a) through (4-11c) into Eq. (4-10) gives:

$$\left(\frac{1}{\Delta t} + \theta\lambda_i\right)\{Q_i\}_n = \left(\frac{1}{\Delta t} - (1-\theta)\lambda_i\right)\{Q_i\}_{n-1} + (1-\theta)\{f_i\}_{n-1} + \theta\{f_i\}_n. \tag{4-12}$$

The free response of Eq. (4-12) is

$$\left(\frac{1}{\Delta t} + \theta\lambda_i\right)\{Q_i\}_n = \left(\frac{1}{\Delta t} - (1-\theta)\lambda_i\right)\{Q_i\}_{n-1}, \qquad i = 1, 2, \ldots, J. \tag{4-13}$$

Eq. (4-13) can be rearranged as follows:

$$\frac{\{Q_i\}_n}{\{Q_i\}_{n-1}} = \frac{1 - (1-\theta)\lambda_i\Delta t}{1 + \theta\lambda_i\Delta t}, \qquad i = 1, 2, \ldots, J. \tag{4-14}$$

Eq. (4-14) provides a means to determine the stability of each mode (hence the stability of the system of ODEs) based upon θ, the eigenvalue λ_i, the amplitude $\{Q_i\}$, and the time step size (Δt).

The requirement for stability is that the magnitude of the ratio of the amplitude values for successive time steps be less than unity, i.e.,

$$\left| \frac{\{Q_i\}_n}{\{Q_i\}_{n-1}} \right| < 1, \qquad i = 1, 2, \ldots, J. \tag{4-15}$$

Applying the inequality of Eq. (4-15) to Eq. (4-14) gives the following conditions for stability:

$$\textit{For } 0 \le \theta < \tfrac{1}{2}: \quad \lambda_i \Delta t < \frac{2}{1 - 2\theta}, \quad i = 1, 2, \ldots, J, \textit{ and} \tag{4-16a}$$

$$\textit{For } \theta \ge \tfrac{1}{2}: \quad \lambda_i \Delta t > \frac{-2}{2\theta - 1}, \quad i = 1, 2, \ldots, J. \tag{4-16b}$$

Closer examination of Eq. (4-16) provides insight into the stability of ODEs of the form given in Eq. (4-1). It is known that the eigenvalue λ_i is always greater than zero; thus, Eq (4-16b) is satisfied for all $\Delta t > 0$. It is concluded, therefore, that equations of the form given in Eq. (4-1) are unconditionally stable for $\theta \ge \frac{1}{2}$; i.e., the solution stability is independent of the time step size. However, for $0 \le \theta < \frac{1}{2}$, Eq. (4-16a) shows that Δt must be less than $\left(\frac{2}{1 - 2\theta} \right) \frac{1}{\lambda_i}$ for the solution to remain stable. Thus, for $0 \le \theta < \frac{1}{2}$,

equations of the form given by Eq. (4-1) are said to be conditionally stable; i.e., the solution stability is dependent upon the time step size.

4.2 On-Line Stability Monitoring

Because time step restrictions are needed for conditionally stable problems, a means to monitor solution stability is needed for proper selection and control of the time step size. One such method of examining the stability of nonlinear systems employs a Liapunov stability criterion [(Hahn, 1963) and (Lehnigk, 1966)]. Considering a system of nonlinear ODEs of the form,

$$\frac{dq}{dt} = f(q,t), \tag{4-17}$$

Liapunov determined that the system Eq. (4-17) is asymptotically stable if a positive definite functional $V(q)$ can be derived such that $\frac{dV}{dt} < 0$, for all q, $t > t_0$. In this thesis, an "on-line" stability monitor is achieved by applying this Liapunov stability criterion for the selected functional

$$V(Q^{n+1}) = \sqrt{\sum_{i=1}^{N} \{R(Q^{n+1})\}_i \cdot \{R(Q^{n+1})\}_i} \qquad (4\text{-}18)$$

where N is the number of nodes, $\{R(Q^{n+1})\}$ is the GWS^{th} algorithm extremum residual evaluated at the most current time station $n+1$ (Freels, 1992). Eq. (4-18) is a positive definite functional. Therefore, if the time rate of change of $V(Q^{n+1})$ is uniformly less than zero, i.e., the norm of the residual is always decreasing, then the Liapunov stability criterion is met.

Application of this stability monitor to an ODE time integration algorithm is presented in Chapter 5.

4.3 On-Line Time Step Control

A fundamental issue in the stability and convergence of an ODE numerical algorithm solution is determination of the maximum allowable time step. An "on-line" time step selection process has been proposed (Freels, 1992) using maximum iterate residuals as follows:

$$\Delta t^{n+1} = \frac{\Delta t^{n}\left[\epsilon_{set} \pm \delta\epsilon\right]}{\left|\frac{\delta Q}{Q_0}\right|_{max}} \tag{4-19}$$

where n is a time step index, ϵ_{set} is a fractional residual input set point, $\delta\epsilon$ allows a window to exist about ϵ_{set}, and $\left|\frac{\delta Q}{Q_0}\right|_{max}$ is the maximum fractional change in the solution field at the beginning of each time step [i.e., at p = 0 in the iteration loop, recall Eq. (3-19)]. As the maximum fractional change in the solution field grows larger, Eq. (4-19) causes the time step to decrease in size to preserve stability. Conversely, as the maximum fractional change in the solution field becomes smaller, Eq. (4-19) allows the time step to increase. Such a method essentially guarantees stable integrations; however, this guarantee comes at the expense of increased computational time due to the restrictions on the size of the time step.

Application of this time step selection process to an ODE numerical time-integration algorithm is presented in Chapter 5. Examination of the process for example problems is presented in Chapter 6.

Chapter 5

Implementation of On-Line Stability Monitoring and Time Step Control

The ultimate goal of this thesis research was to implement the on-line time step control and stability monitoring methods described in Sections 4.2 and 4.3 into a functioning CFD code. The platform chosen to perform this implementation was the *AKCESS* .* computer code (Computational Mechanics Corp., 1995). *AKCESS* .* is a software platform utilizing a "template" and "hook" format for implementing discrete finite element algorithm variations for computational simulation of fluid/thermal/chemical systems. As such, *AKCESS* .* was a suitable test bed for achieving the desired goal.

Implementation of the on-line time step control and stability monitoring methods was achieved by modifying a code "hook" (i.e., a user-written program for implementation of math and physics nuances into the solution process). The hook that controls the time

step process is the "BOTTOM" hook in the COMSLV directory. A discussion of the BOTTOM hook, both before and after the implementation of the stability monitoring and time step control methods, is provided in this chapter.

5.1 Original Time Step Control in the BOTTOM Hook

The original time step control method supplied with the *AKCESS* .* code has user capabilities to initialize and bound the time step size and to provide some control over time step changes. It does not monitor the characteristic of the solution process in selecting appropriate time step size. The *AKCESS* .* model file allows the user to input the desired time step control features to be implemented by the BOTTOM hook. The specific inputs are initial time step (t_{init}), time step multiplier (t_{mult}), maximum allowable time step (t_{max}), and a criterion to raise the maximum time step (t_{crit}).

For a given problem, the first time step size is the value input for t_{init}. If the initial time step input value is not preceded by a minus sign, the problem will run at a fixed time step size equal to t_{init}. If this input value is preceded by a minus sign, the second time step size is determined by multiplying the previous Δt by t_{mult}. This trend continues until the time step size reaches the value for the maximum time step size specified. If after a certain number of iterative cycles (determined by t_{crit}) at the maximum time step size

wherein $|\delta Q|_{max}$ is less than some input level, t_{max} is doubled, and Δt progresses towards this new limit using t_{mult}.

The original time step controller was designed to push Δt to large values as quickly as possible.

5.2 Modified Time Step Control in the BOTTOM Hook

Changes to the BOTTOM hook were made to implement the stability monitoring and time step control theory, Eqs. (4–18) and (4–19) of Sections 4.2 and 4.3. The modified BOTTOM hook allows user flexibility by implementing an "intelligent" time step selection process that is based upon the actual characteristics of the solution as it progresses towards steady-state convergence. Specifically, at time station n+1, the modified BOTTOM hook computes the value of $\frac{\delta Q}{Q_0}$ at each node for each state variable, where $\{Q\} = \{\Omega, \Psi, T, \text{etc.}\}$, $\delta Q = (Q^{n+1} - Q^n)$, and $Q_0 = Q^n$, and determines the largest values for each state variable. The values of $\left|\frac{\delta Q}{Q_0}\right|_{max}$ are compared with values of ϵ_{set}, $\epsilon_{set,high}$, and $\epsilon_{set,low}$ determined by the code for each state variable field based upon user input. These comparisons occur as follows:

- If $\left|\frac{\delta Q}{Q_0}\right|_{max}$ is both less than or equal to $\epsilon_{set,high}$ and greater than or equal to $\epsilon_{set,low}$ for all pertinent state variable fields, the size of the next time step is defined equal to the current time step.

- If $\left|\frac{\delta Q}{Q_0}\right|_{max}$ is less than $\epsilon_{set,low}$ for all pertinent state variable fields, the size of the next time step is increased by multiplying the current time step size by the ratio of ϵ_{set} and $\left|\frac{\delta Q}{Q_0}\right|_{max}$.

- If $\left|\frac{\delta Q}{Q_0}\right|_{max}$ is greater than $\epsilon_{set,high}$ for any pertinent state variable fields, the size of the next time step is decreased by multiplying the current time step size by the ratio of ϵ_{set} and $\left|\frac{\delta Q}{Q_0}\right|_{max}$.

A key consideration for the proper and effective operation of the modified BOTTOM hook is the choice of the ϵ_{set} value. Testing the described modifications revealed that a good way to make this choice is to allow the code to select the value of ϵ_{set} based upon the actual values of $\left|\frac{\delta Q}{Q_0}\right|_{max}$. Specifically, the user is allowed to determine the time station at which the time step control procedure is initiated (NSTEPJ). Prior to this time station, the problem advances at a fixed Δt equivalent to the user input initial time step size (t_{init}). The value of ϵ_{set} is calculated to be the average of the $\left|\frac{\delta Q}{Q_0}\right|_{max}$ values over any number of time steps (≥ 2) up to NSTEPJ, as input by the user.

The values of $\epsilon_{set,high}$ and $\epsilon_{set,low}$, which provide a window about each ϵ_{set}, are based upon user-specified input values. The $\epsilon_{set,high}$ values for each state variable field, which provide the upper bounds of the windows, are set at the minimum values as determined by the following two methods: (1) multiplying ϵ_{set} by the user input value of $\delta\epsilon_1$ ($\delta\epsilon_1 \geq 1.0$), and (2) adding to ϵ_{set} the average change in $\left|\frac{\delta Q}{Q_0}\right|_{max}$ per time step multiplied by a user-defined value. For problems that exhibit very slow changes in the values of $\left|\frac{\delta Q}{Q_0}\right|_{max}$, the second method provides the minimum value and, therefore, ensures that the value of $\epsilon_{set,high}$ is set low enough so that a request for a change in the size of the time step occurs within a reasonable number of time steps. Conversely, for problems that exhibit rapid changes in the values of $\left|\frac{\delta Q}{Q_0}\right|_{max}$, the first method provides the minimum value and, therefore, ensures that the value of $\epsilon_{set,high}$ is set low enough so that the change in time step size has a reasonable bound.

Similarly, the $\epsilon_{set,low}$ values for each state variable field, which provide the lower bounds of the windows, are set at the maximum value as determined by the following two methods: (1) multiplying ϵ_{set} by the user input value of $\delta\epsilon_2$ ($\delta\epsilon_2 \leq 1.0$), and (2) subtracting from ϵ_{set} the average change in $\left|\frac{\delta Q}{Q_0}\right|_{max}$ per time step multiplied by a user-defined value. The user has a choice to either keep the values of ϵ_{set}, $\epsilon_{set,high}$, and $\epsilon_{set,low}$ constant, or to allow these values to adjust to new values if the time step size changes over a user-specified number of consecutive time steps.

The preceding paragraphs have described the time step control options available to the user in the modified BOTTOM hook. There are three additional issues that deserve further comment. The first involves the use of $\left|\frac{\delta Q}{Q_0}\right|_{max}$ as the controlling solution characteristic for determining time step size. It has been found as discussed in Chapter 6 that the value of $\left|\frac{\delta Q}{Q_0}\right|_{max}$ can, at times, change in an erratic manner from one time station to the next. Erratic changes in $\left|\frac{\delta Q}{Q_0}\right|_{max}$ may lead to erratic changes in Δt. A possible solution is to use $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$, where the numerator and denominator are not necessarily determined at the same node within a state variable field, as the controlling solution characteristic in determining time step size. Changes in $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ tend to be much less erratic and, therefore, may be better suited for controlling the time step size. The user has the option of using either $\left|\frac{\delta Q}{Q_0}\right|_{max}$ or $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$, as desired.

The second issue adds complication to the time step control method, but is important to understand. The modified time step control method is applied separately and independently to any or all solution fields, as desired. For example, for the vorticity-streamfunction driven cavity problems discussed in Chapter 6, the time step control method is applied to the first-iterate fields for both vorticity and streamfunction. For the vorticity-streamfunction thermal cavity problems discussed in Chapter 6, the time step control method may be applied to the vorticity, streamfunction, and temperature first-iterate fields. An allowable time step size is determined separately based upon the ϵ_{set}, $\epsilon_{set,high}$, and $\epsilon_{set,low}$ values and upon $\left|\frac{\delta Q}{Q_0}\right|_{max}$ or $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ for each field, where, in this

particular case, $\{Q\} = \{\Omega, \Psi, T\}$. The overall time step size used for the next time station is taken to be the smallest size determined from tested applicable solution fields.

The third issue involves the ability to monitor the stability of the solution in a quantitative manner. Such a monitor, using the theory described in Section 4.2, has been implemented into the modified BOTTOM hook. This monitor provides a warning statement in the SOUT file at every time step in which the time rate of change of $V(Q^{n+1})$ [see Eq. (4-18)] is not less than zero.

A summary of user input parameters and settings in the BOTTOM hook is provided in Table 5–1. A FORTRAN listing of the entire BOTTOM hook is provided in Appendix A.

For best performance, it is desired that a problem run for several times steps (say 5 to 10) at a given time step size before adjusting the Δt to a new value. Using such a philosophy helps to assure that the iterative process remains stable and, therefore, achieves the desired steady-state convergence. The user input options described in the previous paragraphs provide the necessary controls to bound and manipulate the time step selection process to conceivably achieve whatever criteria the user may have.

Table 5-1. Time Step Control Method User Input Parameters and Settings

Parameter	Settings
NFLAG1	Set to 0 to use the original time step control method; set to 1 to use the new time step control method.
NDQSET	Set to 0 to base the new time step control method on the $\|\delta Q/Q\|_{max}$ values; set to 1 to base the new time step control method on the $\|\delta Q\|_{max}/\|Q\|_{max}$ values.
NSTEPJ	Set to the desired time step number to begin the new time step control method
NSTEPK	Set to the desired time step number to begin averaging the $\|\delta Q/Q\|_{max}$ or $\|\delta Q\|_{max}/\|Q\|_{max}$ values (depending on the value of NDQSET) to determine the value of ϵ_{set}.
EPSETHIF	Set to the desired value (> 1.0) to be used as a multiplier on ϵ_{set} to determine the upper bound value of $\epsilon_{set,high}$.
EPSETLOF	Set to the desired value (< 1.0) to be used as a multiplier on ϵ_{set} to determine the lower bound value of $\epsilon_{set,low}$.
EPSETF	Set to the desired value to be used as a multiplier on the average change in the $\|\delta Q/Q\|_{max}$ or $\|\delta Q\|_{max}/\|Q\|_{max}$ values at successive time steps between time step NSTEPK and NSTEPJ as an alternate means for determining $\epsilon_{set,high}$ and $\epsilon_{set,low}$.
NPLACEF	Set to the desired number of consecutive Δt changes required to reset the values of ϵ_{set}, $\epsilon_{set,high}$, and $\epsilon_{set,low}$ (i.e., if the new time step control method requests a change in the time step size in each active solution field NPLACEF consecutive times, new values of ϵ_{set}, $\epsilon_{set,high}$, and $\epsilon_{set,low}$ are computed and the values reset.
DTMINX	Set to the desired minimum Δt.
IGNORE(NP)	Set to 0 for each solution field to be used by the time step control method; set to NP for each solution field to be ignored by the time step control method [e.g., if solution field 3 is to be used by the time step controller, set IGNORE(3) = 0; if it is to be ignored, set IGNORE(3) = 3].

Chapter 6

Discussion and Results

This chapter documents the results of CFD experiments demonstrating the effectiveness of the time step controller described in Chapter 5. The approach is to solve well-known benchmark problems, e.g., the driven and the thermal cavity problems using several *GWS** algorithm forms, specifically vorticity-streamfunction, the continuity constraint (or "phi") method, and the penalty method. This chapter documents behavior of the time step size, as a function of developed user-control parameters, as solution time advances towards the problem steady state.

6.1 Vorticity-Streamfunction Performance

The vorticity-streamfunction formulation (see Sects. 2.5 and 3.1) was selected for the 2D driven and thermal cavity problems. Cases at Reynolds numbers of 10 and 100 for the driven cavity, and Rayleigh number of 1000 for the thermal cavity, were evaluated using explicit ($\theta = 0.0$) and implicit ($\theta = 0.5$ and $\theta = 1.0$) time integration. For the driven cavity cases, a uniform 16 × 16 element mesh of the unit square was used, while for the thermal cavity cases a non-uniform mesh was used. The problem convergence criterion $\frac{|\delta\varrho|_{max}}{|\varrho_0|_{max}} \leq 1 \times 10^{-4}$ was selected. Unless otherwise stated, all cases involving the new time step control method used the following factors (recall Table 5–1) to determine the values of ϵ_{set}, $\epsilon_{set,high}$, and $\epsilon_{set,low}$: NSTEPK = NSTEPJ - 2, EPSETF = 10.0, EPSETLOF = 0.8, EPSETHIF = 1.25, and NPLACEF = 100000.0.

6.1.1 2D Driven Cavity (Re = 10)

Several 2D driven cavity cases were run at Re = 10 to evaluate the effectiveness of the developed time step control algorithm. Cases for implicitness factor $\theta = 0.0$, $\theta = 0.5$, and $\theta = 1.0$ are presented herein.

6.1.1.1 $\theta = 0.0$ Cases

Several 2D vorticity-streamfunction driven cavity cases were run using explicit time integration ($\theta = 0.0$) at Re = 10. Case 1 used the *AKCESS* .* original time step controller with an initial time step of 0.002, a maximum time step of 1.0, and a time step size multiplier of 1.1. The solution diverged, as documented in Table 6–1.

The solution started off convergent. For the first 10 time steps, the magnitude of the $\frac{(\delta Q)_{max}}{(Q_0)_{max}}$ values in the vorticity solution field decreased. Beginning at time step 11, however, these values began to increase. The streamfunction solution field exhibited similar behavior; however, the magnitude of the $\frac{(\delta Q)_{max}}{(Q_0)_{max}}$ values did not begin to increase until time step 13.

The maximum time step size growth allowed was obviously too large. It also violated the stability criterion given in Eq. 4-16a. The original time step controller was unable to correct the instability condition in time, although it did attempt to reduce the time step size beginning at time step number 24. Unfortunately, the correction was too little, too late as divergent behavior had already begun as early as time step 11. Thus, the solution diverged.

Table 6–1. Case 1 Results

Time Step Number	Time Step Size	$\frac{(\delta\Omega)_{max}}{(\Omega_0)_{max}}$ for Vorticity	$\frac{(\delta\Omega)_{max}}{(\Omega_0)_{max}}$ for Streamfunction
1	2.000000E-03	-5.898784E+02	9.000000E-05
2	2.200000E-03	-4.851490E-02	-1.395796E-01
3	2.420000E-03	2.639059E-01	-6.024632E-02
4	2.662000E-03	1.660664E-01	-6.515153E-02
5	2.928200E-03	1.068979E-02	-5.738382E-02
6	3.221020E-03	7.567719E-02	-5.097472E-02
7	3.543122E-03	5.565470E-02	-4.546796E-02
8	3.897434E-03	4.394495E-02	-4.129435E-02
9	4.287178E-03	3.793960E-02	-4.110008E-02
10	4.715895E-03	3.322631E-02	-4.042782E-02
11	5.187485E-03	-3.780365E-02	-3.992940E-02
12	5.706233E-03	-5.819858E-01	-3.966466E-02
13	6.276857E-03	1.469642E-01	-4.074375E-02
14	6.904542E-03	-3.674974E-01	-4.824347E-02
15	7.594997E-03	1.565117E+00	-8.140001E-02
16	8.354496E-03	-2.836006E+00	-2.800283E-01
17	9.189946E-03	4.962454E+00	4.128799E-01
18	1.010894E-02	-5.862892E+00	-1.309928E+00
19	1.111983E-02	6.354085E+00	2.173729E+00
20	1.223182E-02	7.074303E+00	-7.093891E+00
21	1.345500E-02	-7.885747E+00	9.066779E+00

Table 6–1. (continued)

Time Step Number	Time Step Size	$\frac{(\delta Q)_{max}}{(Q_0)_{max}}$ for Vorticity	$\frac{(\delta Q)_{max}}{(Q_0)_{max}}$ for Streamfunction
22	1.480050E-02	2.454124E+01	-4.324342E+01
23	1.628055E-02	4.358099E+03	-5.136364E+03
24	1.480050E-02	-2.675523E+07	-2.175519E+07
25	1.345500E-02	-3.269576E+14	-4.365066E+14
26	1.223182E-02	2.938888E+29	2.978597E+29
27	1.111983E-02	-4.924604E+58	5.358876E+58
28	1.010894E-02	-2.228725E+117	-1.956132E+117

Case 1 was reexecuted using the new time step controller described in Chapter 5. This second case (Case 2) used the same initial and maximum time step sizes as Case 1, but added a minimum time step size of 0.001. The time step size was held constant for the first 50 time steps (i.e., NSTEPJ = 50) to allow sufficient time for the residuals to peak. Afterward, the time step size was based on the values of $\left|\frac{\delta Q}{Q_0}\right|_{max}$ in the vorticity and streamfunction first-iterate fields (within the bounds of the minimum and maximum time step sizes). This solution converged to steady state after 215 time steps in a total CPU time of 3228 s at $\theta = 0.0$.

A plot of the time step size versus time step for Case 2 is provided in Fig. 6–1. It is noted that although a maximum time step size of 1.0 was allowed, the controller never let the time step size exceed about 0.05. In general, the time step size oscillated between approximately 0.02 and the minimum input time step size of 0.001.

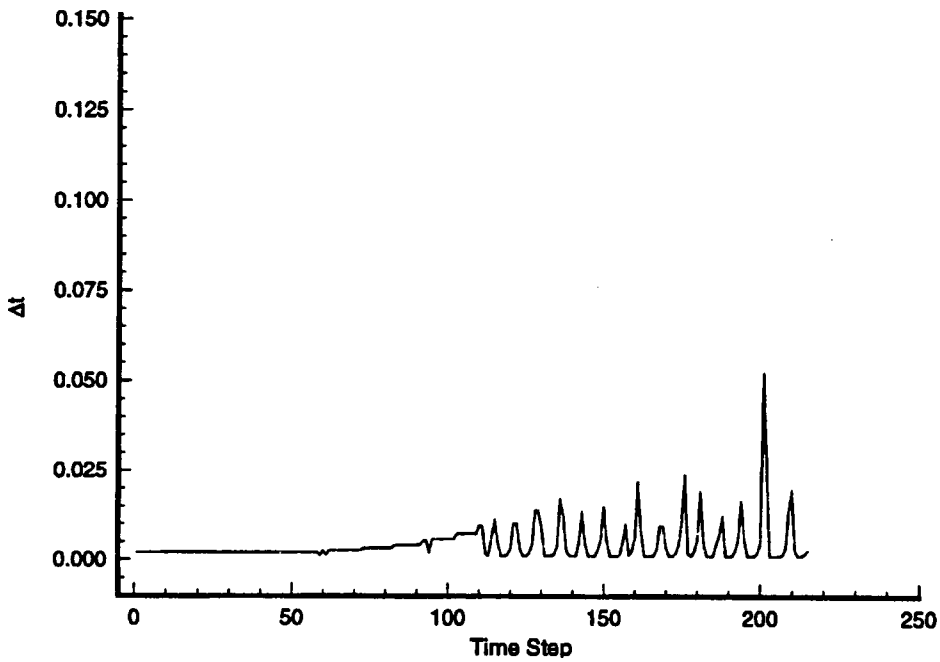


Fig. 6–1. Case 2 Time Step Size vs. Time Step

Plots of $\left|\frac{\delta Q}{Q_0}\right|_{max}$ and $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ for the vorticity and streamfunction first-iterate fields are provided in Figs. 6–2 through 6–5. The $\left|\frac{\delta Q}{Q_0}\right|_{max}$ values appear to be more erratic in nature than the $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ values. The magnitude of the oscillations in the $\left|\frac{\delta Q}{Q_0}\right|_{max}$ plots are significantly larger than those in the $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ plots.

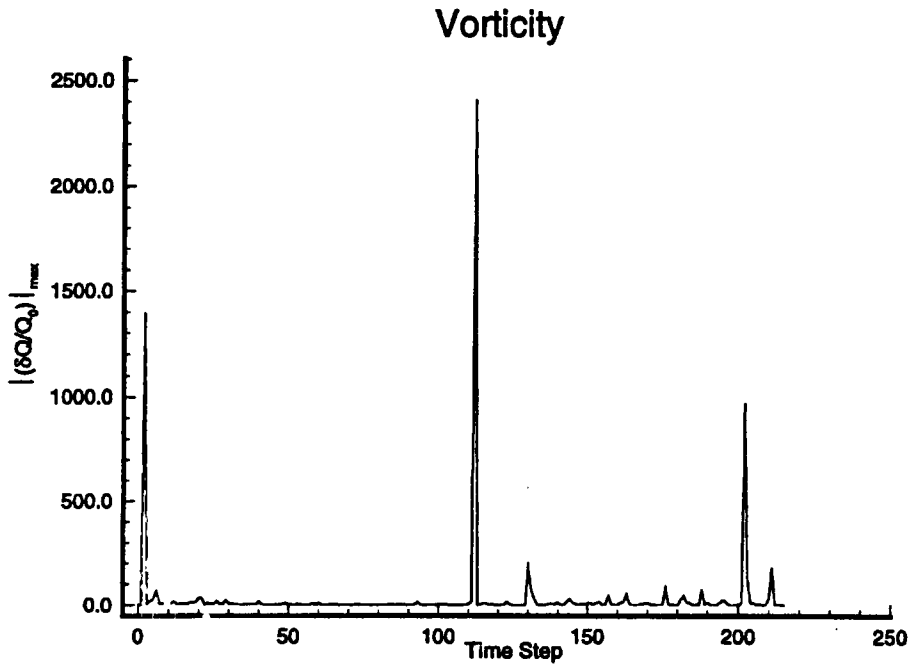


Fig. 6–2. Case 2 $|(\delta Q/Q)|_{max}$ Values vs. Time Step for Vorticity

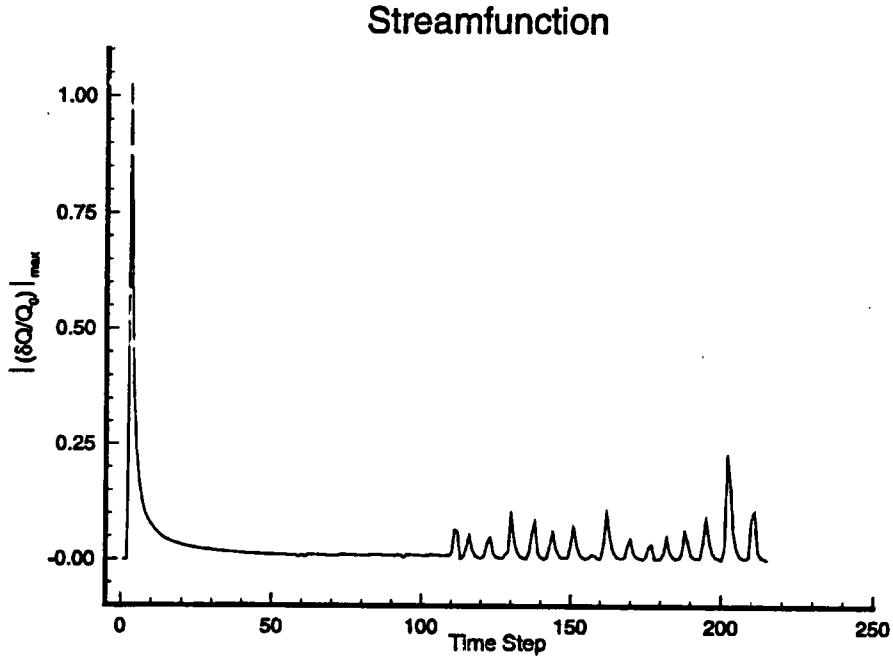


Fig. 6–3. Case 2 $|(\delta Q/Q)|_{max}$ Values vs. Time Step for Streamfunction

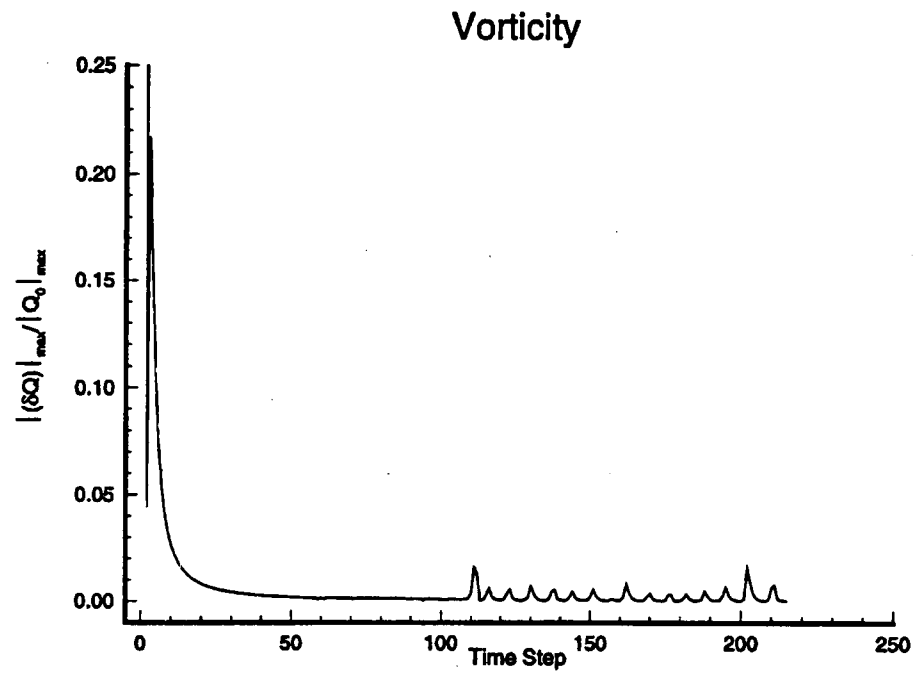


Fig. 6–4. Case 2 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Vorticity

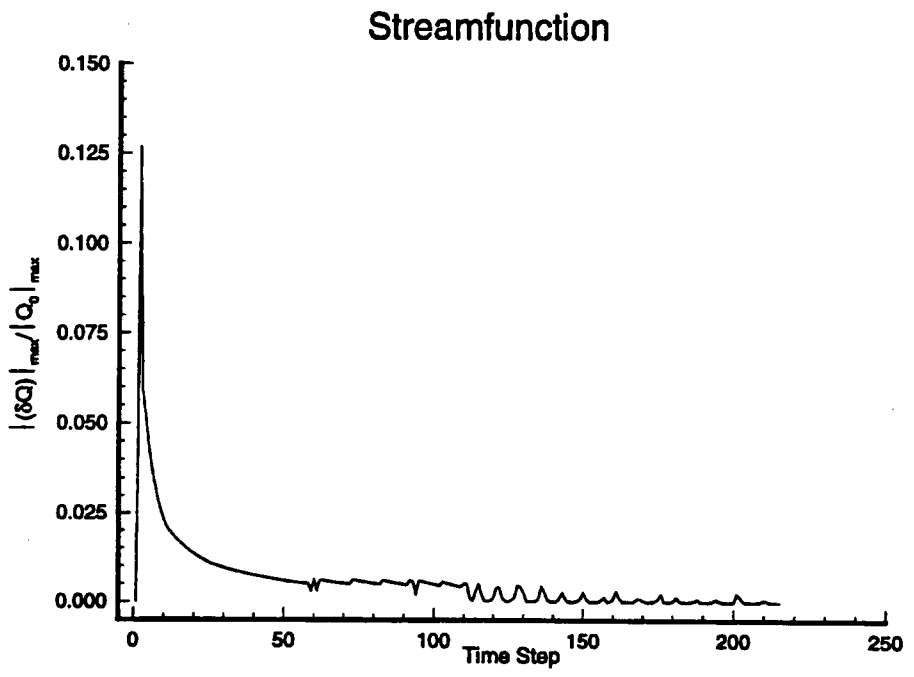


Fig. 6–5. Case 2 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Streamfunction

To determine the effects of basing the time step controller on the $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ values in the vorticity and streamfunction solution fields (in distinction to $\left|\frac{\delta\Omega}{\Omega_0}\right|_{max}$ as used in Case 2), another case was run. This new case (Case 3), using identical time step size and ϵ_{set} inputs as Case 2 (with the one exception noted in the previous sentence), converged in 224 time steps in a total CPU time of 3351 s.

Fig. 6–6 is the plot of the time step size versus time step for Case 3. Comparing Fig. 6–6 with Fig. 6–1 (the x-axis and y-axis scales are the same for ease of comparison), note no significant difference in the time step size behavior occurs, except for the magnitude of the large spike that appears near time step number 200, and the Case 3 data appear smoother than Case 2 between time steps 50 and 110. Other than these minor differences, the integration control functioning is essentially identical.

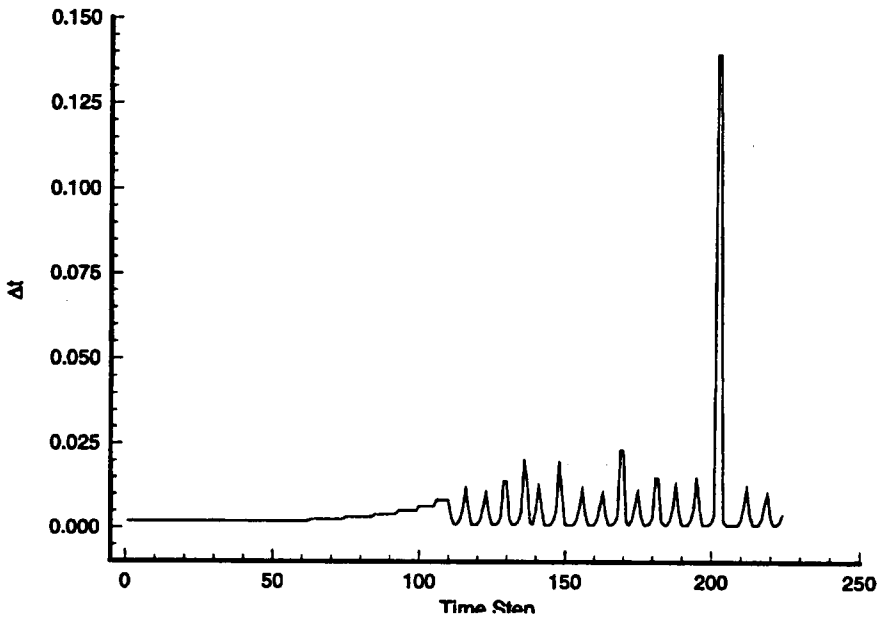


Fig. 6–6. Case 3 Time Step Size vs. Time Step

The solutions obtained for Cases 2 and 3, for all practical purposes, represent the most rapid achievement of steady state for the Re = 10 driven cavity for $\theta = 0.0$. Some initial and maximum time step size selections that would allow steady state to be achieved in fewer time steps may exist. However, the developed theory does not provide guidance here, hence the user must make numerous case studies to determine such values. Thus, the results shown in Cases 2 and 3 provide evidence that the goals cited in Chapter 1 have been met, i.e., the solution was obtained with minimal user interaction in some sense with minimal CPU time.

It is also possible to use the developed time step controller to achieve time-accurate solutions for problems, which often requires more user interaction and more CPU time. Specifically, any case can be run with tighter bounds on the allowed time step size to improve solution time accuracy. For example, Case 4 was run with an initial time step size of 0.0005, a minimum size of 0.00025, and a maximum size of 0.002. Using the $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ criterion for both vorticity and streamfunction solution fields, the steady-state solution converged in 475 time steps in a total CPU time of 7121 s. The total problem time was 0.77 non-D units.

The history of time step size versus time step is graphed in Fig. 6–7. Note that time step size increased between time steps 50 to 150 in a very methodical manner, from the initial value of 0.0005 to the maximum of 0.002.

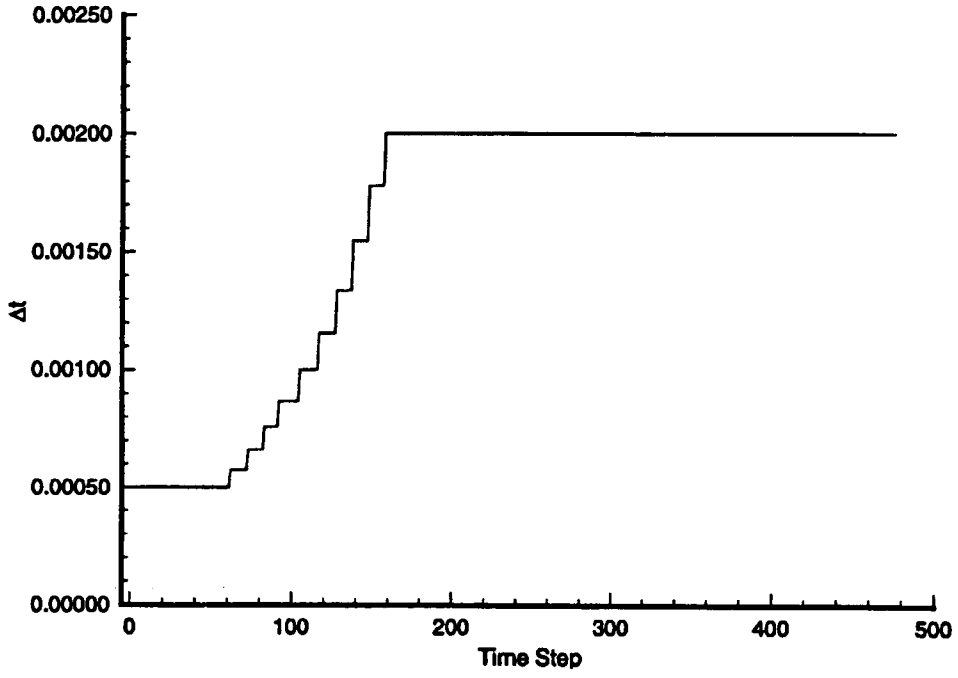
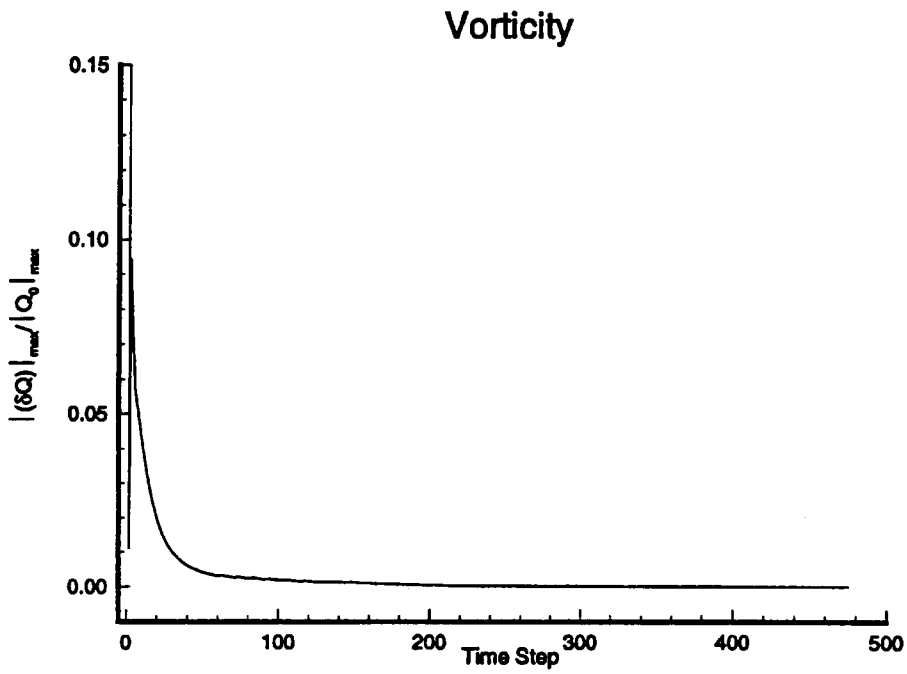


Fig. 6–7. Case 4 Time Step Size vs. Time Step

The behavior in Fig. 6–7 is clearly a function of $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ criterion in the streamfunction solution field, see Figs. 6–8 and 6–9. Beginning with time step number 50, the $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ values for streamfunction first iterates change relatively slowly, whereas the values of $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ for vorticity change relatively more quickly. Thus, the $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ values in the streamfunction field control the time step size, as constrained by the developed theory.



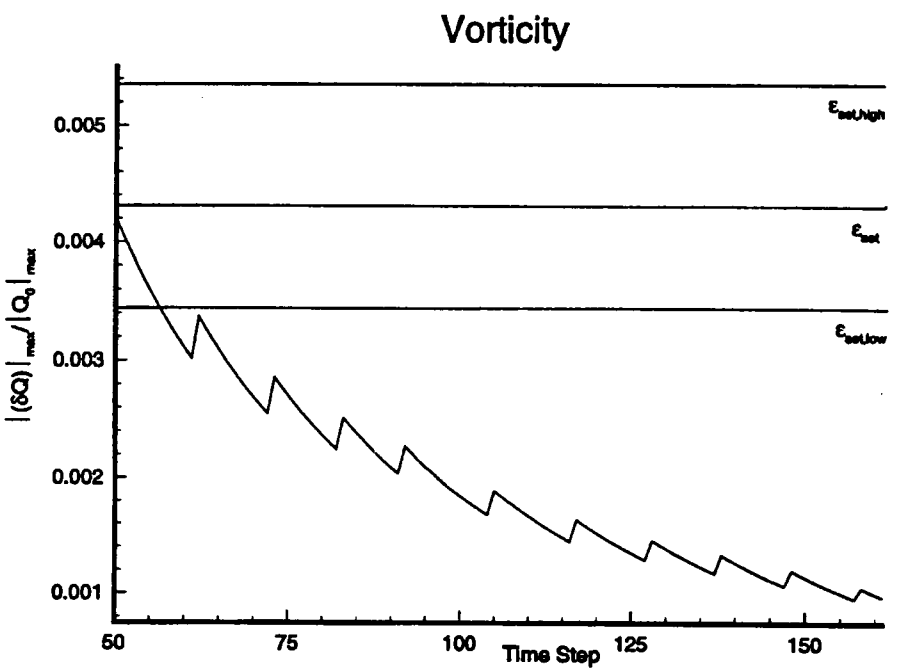


Fig. 6–8. Case 4 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Vorticity

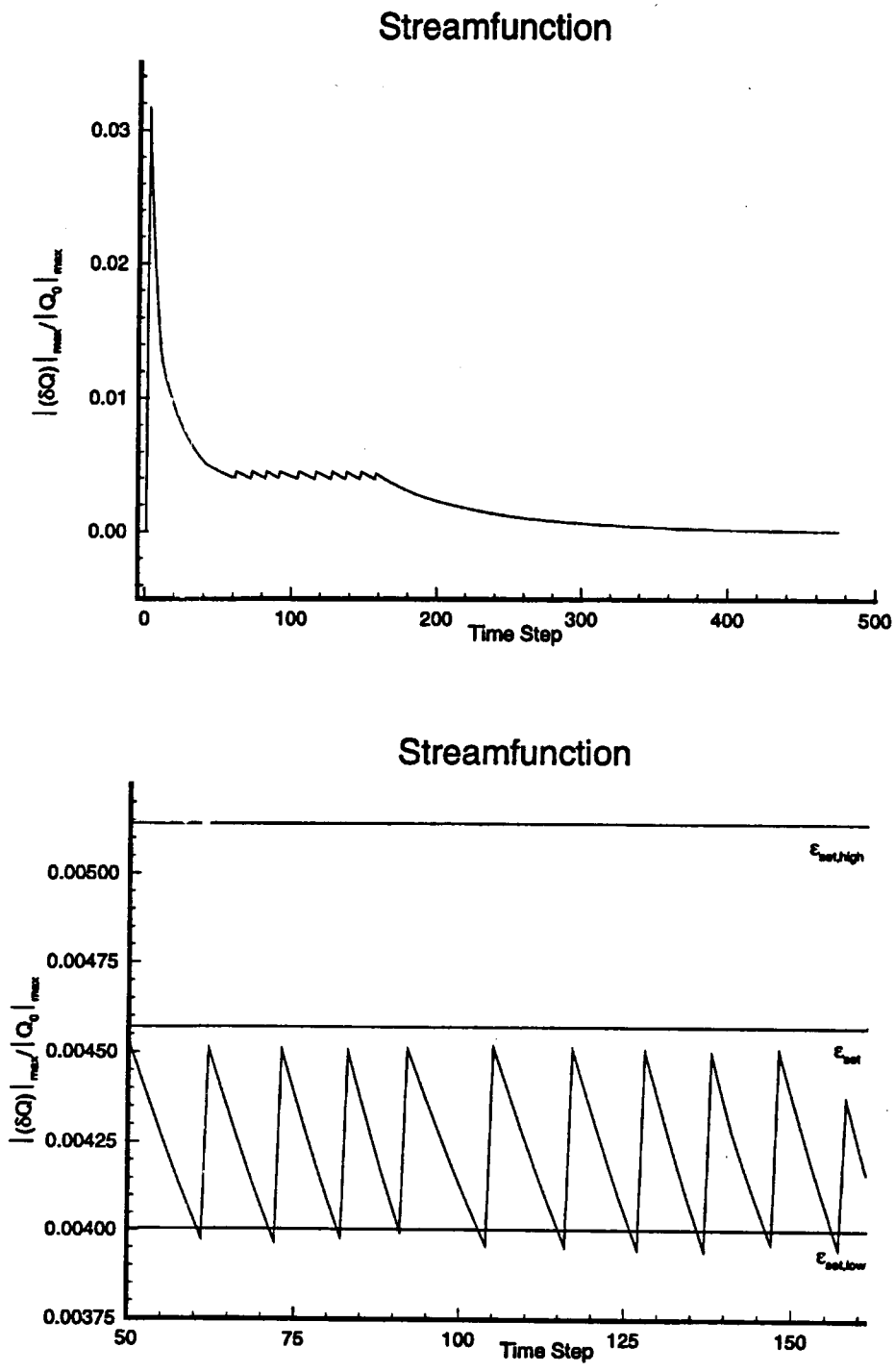


Fig. 6–9. Case 4 $|(\delta Q)|_{max} / |Q|_{max}$ Values vs. Time Step for Streamfunction

For this case, the time step size was held constant for the first 50 time steps to ensure that the solution field first-iterates had peaked before before initiating the time step controller. At time step number 50, the values for ϵ_{set} were determined by the algorithm for both the vorticity and streamfunction fields based upon the calculated average of the values of $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ at time step number 49 and 50 for each field. The value of $\epsilon_{set,high}$ for each field was defined to be 1.25 times the value of ϵ_{set} for each field, while the value of $\epsilon_{set,low}$ for each field was set at 0.8 times the value of ϵ_{set} for each field. As shown in Figs. 6–8 and 6–9, the $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ values in the vorticity and streamfunction solution fields reached their particular $\epsilon_{set,low}$ value after a few time steps. This condition prompted an increase in the size of the time step.

The increased time step size then caused an increase in $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$, because the size of $(\delta Q)_{max}$ is obviously dependent on the size of the time step (i.e., the larger the Δt, the larger the size of $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$). At this point, it is noted that the $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ value for the streamfunction solution field is now back within the bounds of $\epsilon_{set,low}$ and $\epsilon_{set,high}$, whereas the $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ value for the vorticity solution field is still below the $\epsilon_{set,low}$ value. Therefore, the time step size remained fixed until $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ in both vorticity and streamfunction solution fields were outside the bounds of $\epsilon_{set,low}$ and $\epsilon_{set,high}$. This process continued until the maximum (or minimum) Δt was obtained. At that point, the Δt remained constant until the solution converged to steady state.

The solution characteristics displayed in Figs. 6–7 through 6–9 represent a very smooth, steady, methodical approach to time step control. If nothing else, this approach may provide the user with additional comfort as to the accuracy of the solution. Conversely, characteristics such as those displayed in Figs. 6–1 through 6–6 for Cases 2 and 3 do not give the user comfort about solution quality.

6.1.1.2 $\theta = 0.5$ Cases

The 2D vorticity-streamfunction driven cavity cases were rerun using the trapezoidal rule ($\theta = 0.5$) time integration for Re = 10. The first case (Case 5) used the original time step control method with an initial time step size of 0.0005, a maximum time step size of 1.0, and a time step multiplier of 1.1. This case converged in 61 time steps in a total CPU time of 1510 s and a total problem time of 1.67 non-D units.

The time step size and $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ for vorticity and streamfunction first-iterate fields for Case 5 are graphed in Figs. 6–10 through 6–12. These data confirm a very smooth progression towards convergence to steady state. Comparing to Case 1 results, note that the time step size for Case 5 exceed the maximum time step size in Case 1 by a factor of 10. This is allowable, stability-wise, since $\theta = 0.5$ is an implicit method.

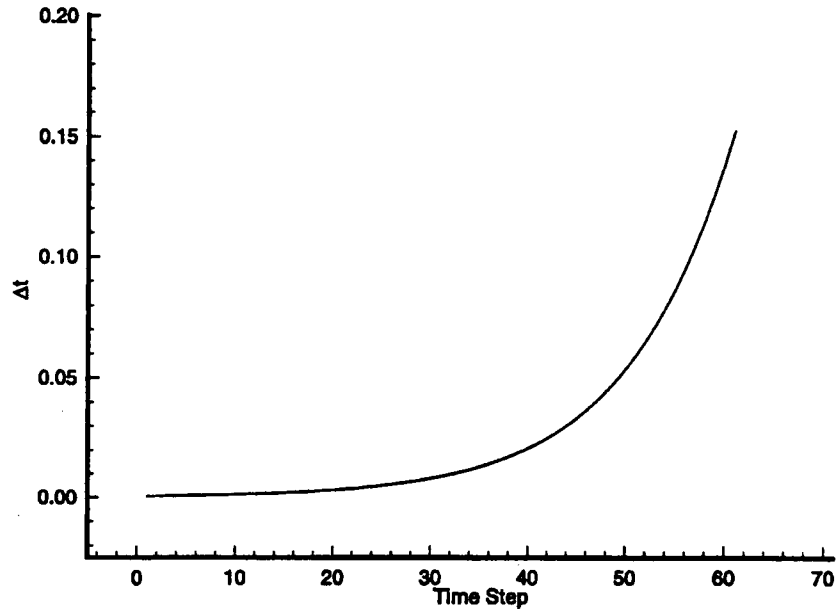


Fig. 6–10. Case 5 Time Step Size vs. Time Step

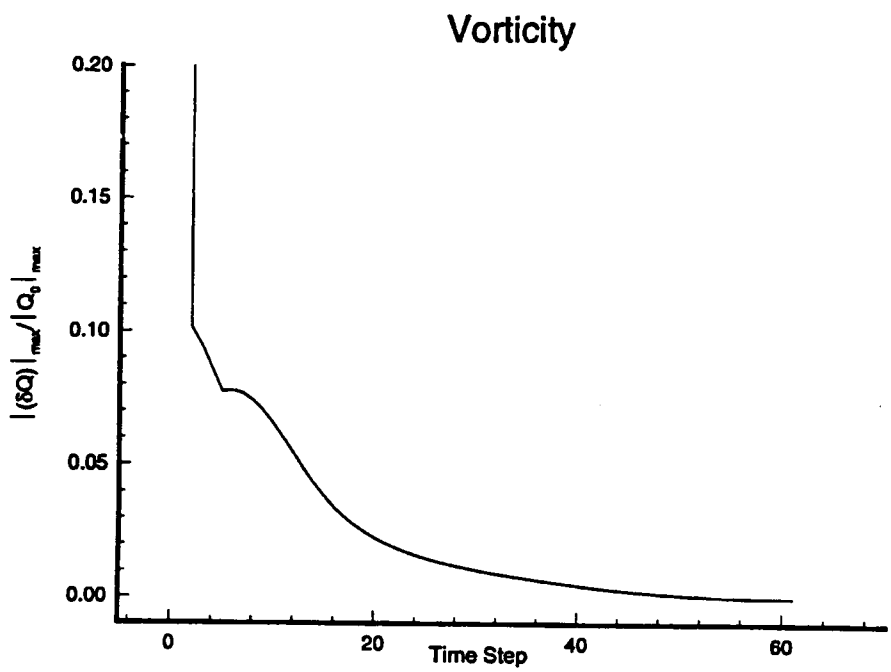


Fig. 6–11. Case 5 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Vorticity

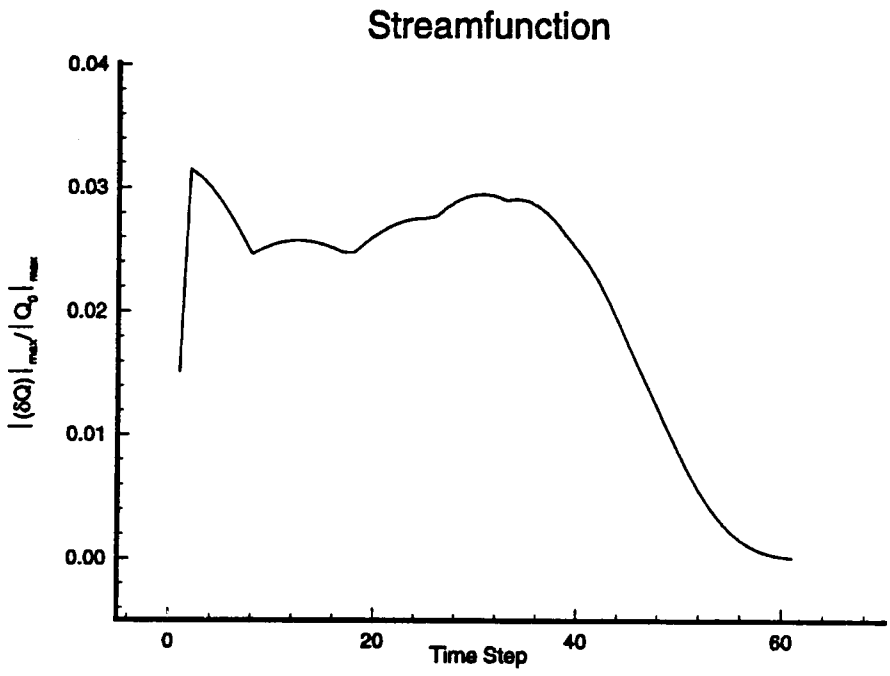


Fig. 6–12. Case 5 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Streamfunction

Although the Case 1 solution eventually diverged, convergence was possible at the larger time step for Case 5 because the Newton algorithm is unconditionally stable for $\theta \geq 0.5$, recall Eq. 4–16.

Case 6 used the same initial and maximum time step criterion, but employed time step control based on $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ for both the vorticity and streamfunction first-iterate fields. This case converged to steady state in 240 time steps in 5948 s of CPU time and a total problem time of 2.49 non-D units.

Time step history and $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ values for vorticity and streamfunction first-iterates for Case 6 are graphed in Figs. 6–13 through 6–15. These data also demonstrate a very smooth progression towards convergence to steady state.

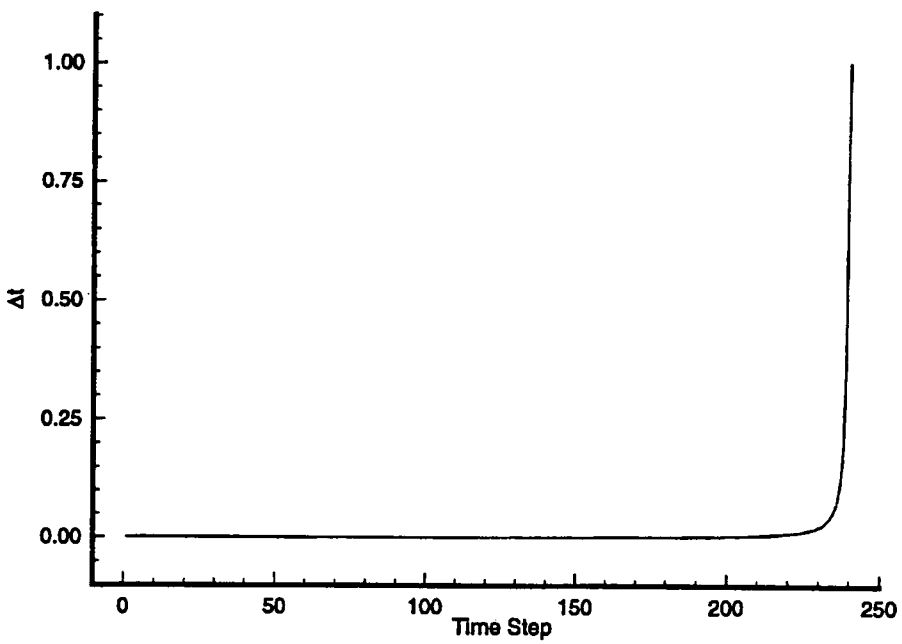


Fig. 6–13. Case 6 Time Step Size vs. Time Step

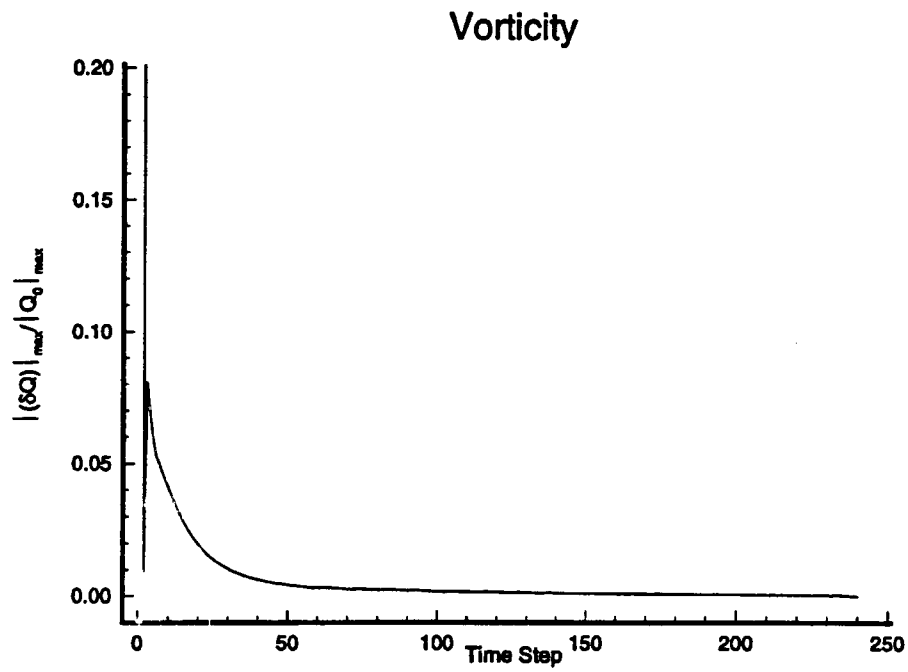


Fig. 6–14. Case 6 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Vorticity

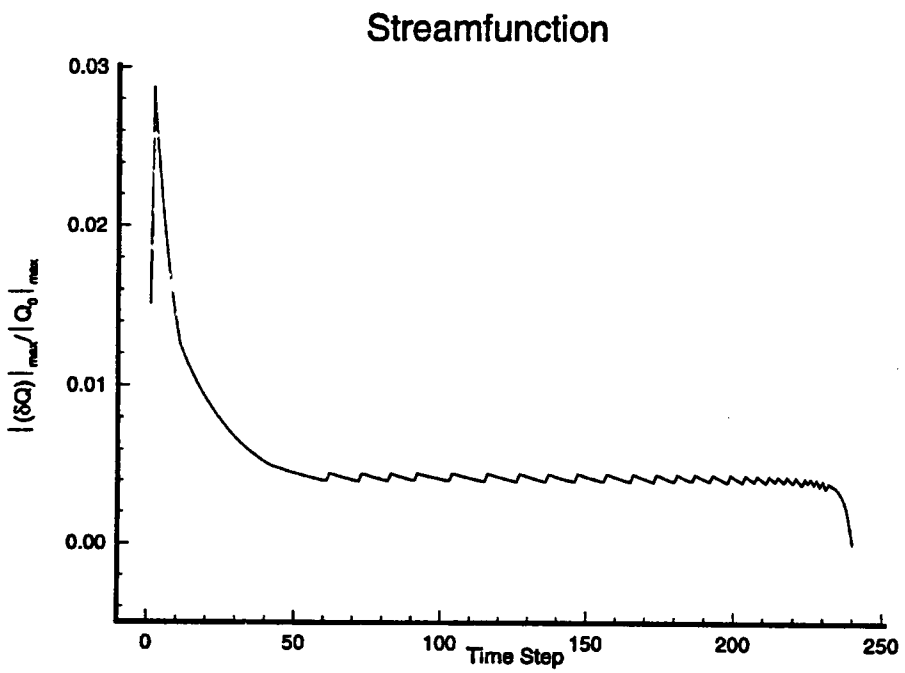


Fig. 6–15. Case 6 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Streamfunction

Note that the total problem time differs between Cases 5 and 6 (1.67 vs. 2.49 non-D units). Although time accurate solutions may be obtained via the $\theta = 0.5$ time integration, these solutions depend on the maximum time step size allowed and/or obtained by the solution. To quantify this phenomenon, Case 7 was run using the same initial, maximum, and minimum time step sizes as for Case 4 with explicit time integration. Case 7 converged in 477 time steps in a total CPU time of 7202 s and a total problem time of 0.77 non-D units. This total problem time is identical to the Case 4 total problem time. Thus, the time accuracy of the solution can be preserved using $\theta =$ 0.5 time integration.

Plots of the time step size and $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ for vorticity and streamfunction first-iterate fields for Case 7 are provided in Figs. 6–16 through 6–18. The time step history shows a methodical increase in Δt from the initial input value of 0.0005 to the maximum value of 0.002. Once the desired maximum Δt is reached, the solution converges to steady state in a smooth manner.

6.1.1.3 $\theta = 1.0$ Cases

Two 2D vorticity-streamfunction driven cavity cases were run using the $\theta = 1.0$ fully-implicit time integration at Re = 10. The first case (Case 8) used the original time step

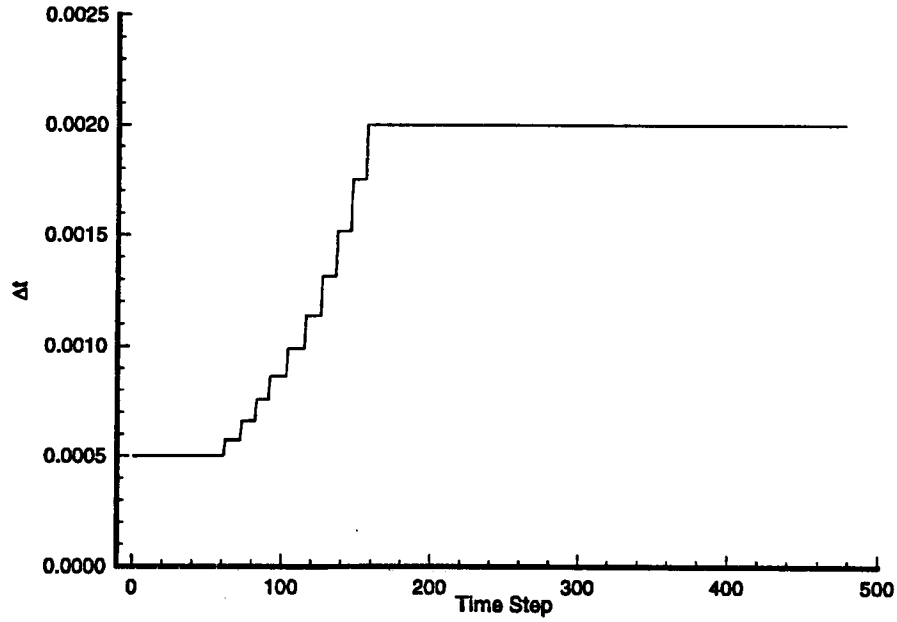


Fig. 6–16. Case 7 Time Step Size vs. Time Step

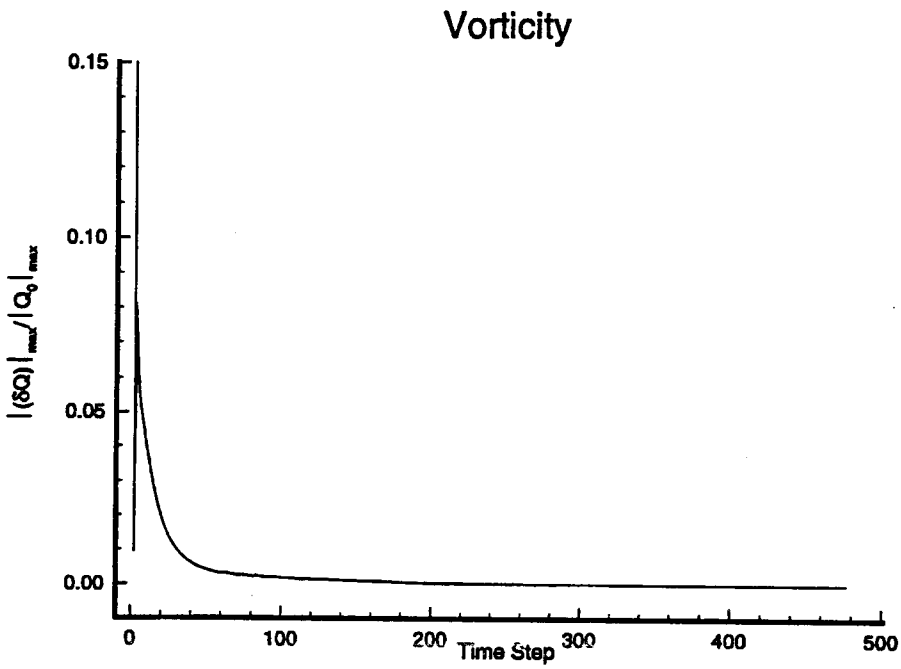


Fig. 6–17. Case 7 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Vorticity

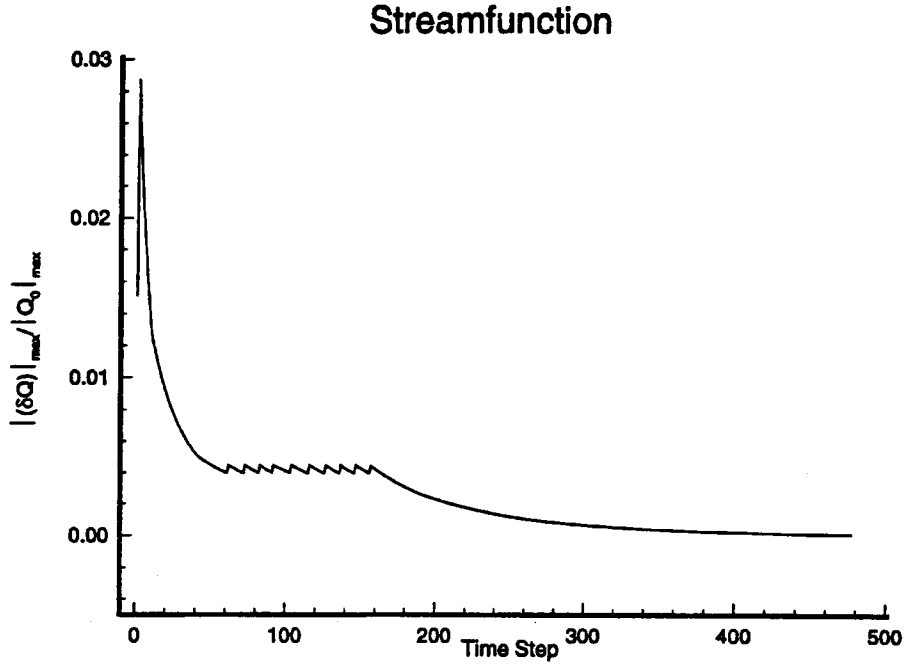


Fig. 6–18. Case 7 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Streamfunction

control method with an initial time step size of 0.1, a maximum time step size of 1.0, and a time step multiplier of 1.1. This case converged in 14 time steps in a total CPU time of 383 s.

Plots of time step history and $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ for vorticity and streamfunction first iterates for Case 8 are provided in Figs. 6–19 through 6–21. These plots indicate a very smooth and rapid progression towards steady-state convergence.

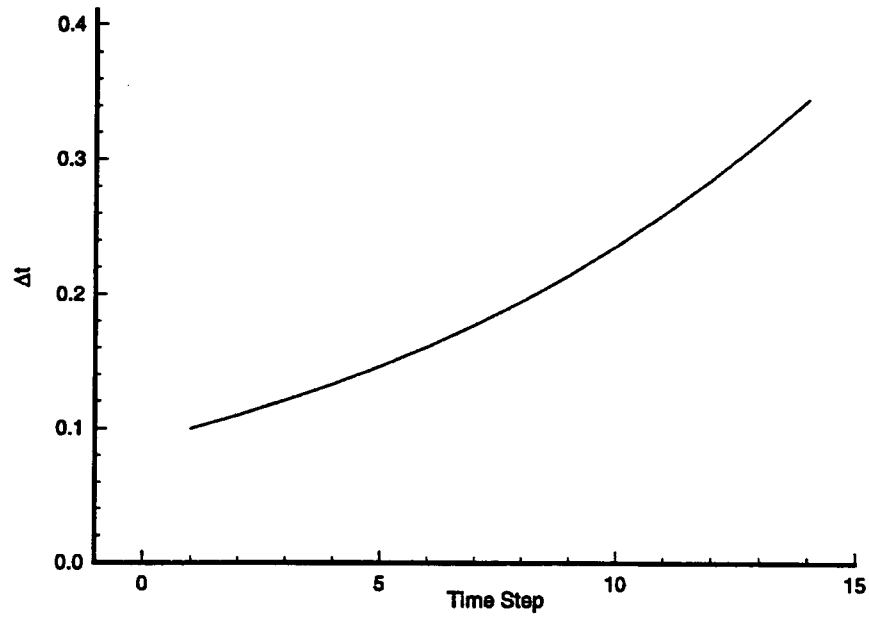


Fig. 6–19. Case 8 Time Step Size vs. Time Step

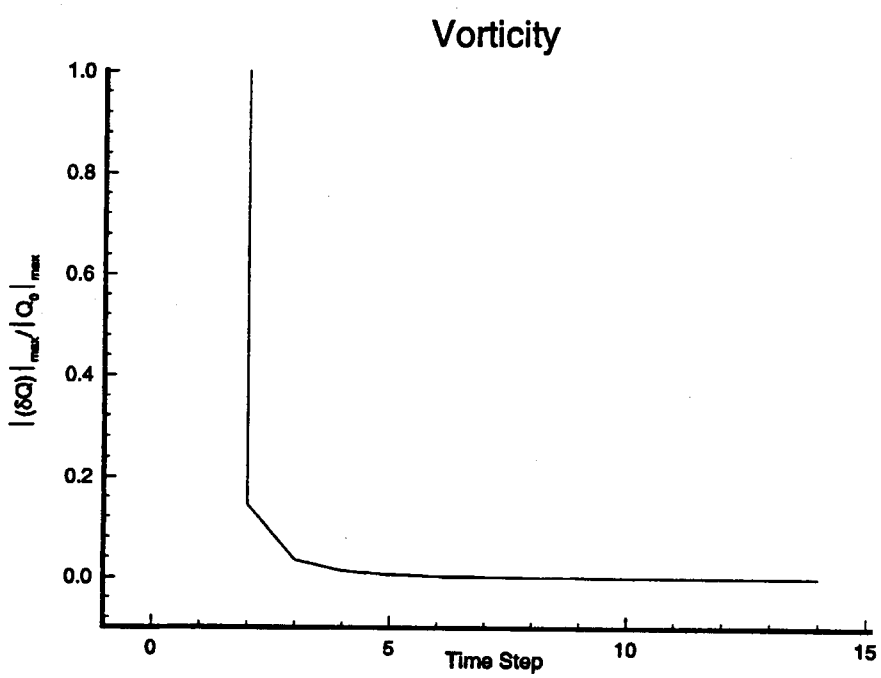


Fig. 6–20. Case 8 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Vorticity

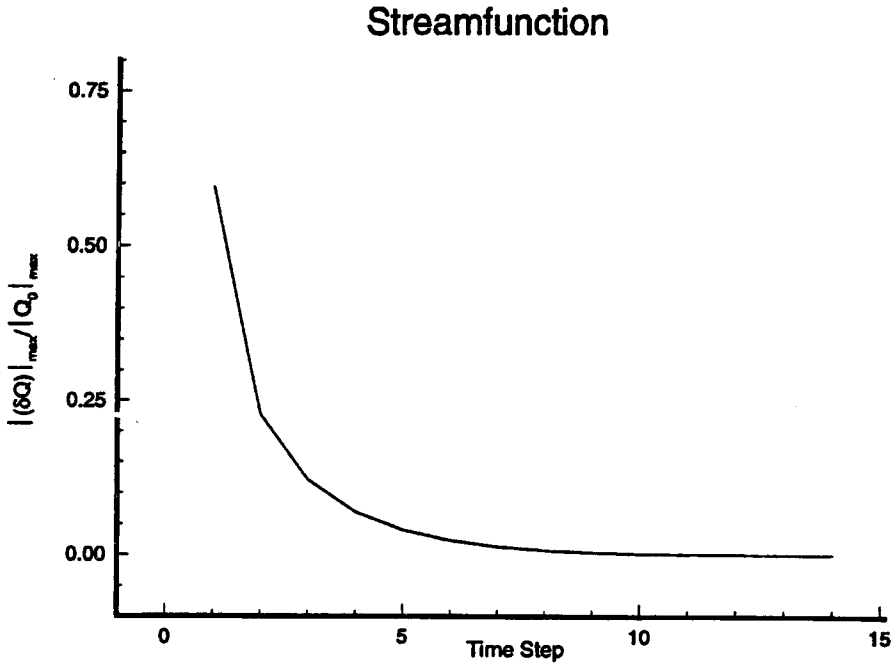


Fig. 6–21. Case 8 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Streamfunction

Case 9 used the new time step control method based on $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ for vorticity and streamfunction with the same initial and maximum time step sizes as the previous case and converged in 16 time steps. The total CPU time was 438 s. Plots of the time step size and $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ for vorticity and streamfunction for Case 9 are graphed in Figs. 6–22 through 6–24. Similar to Case 8, these plots demonstrate a smooth and rapid progression towards steady-state convergence.

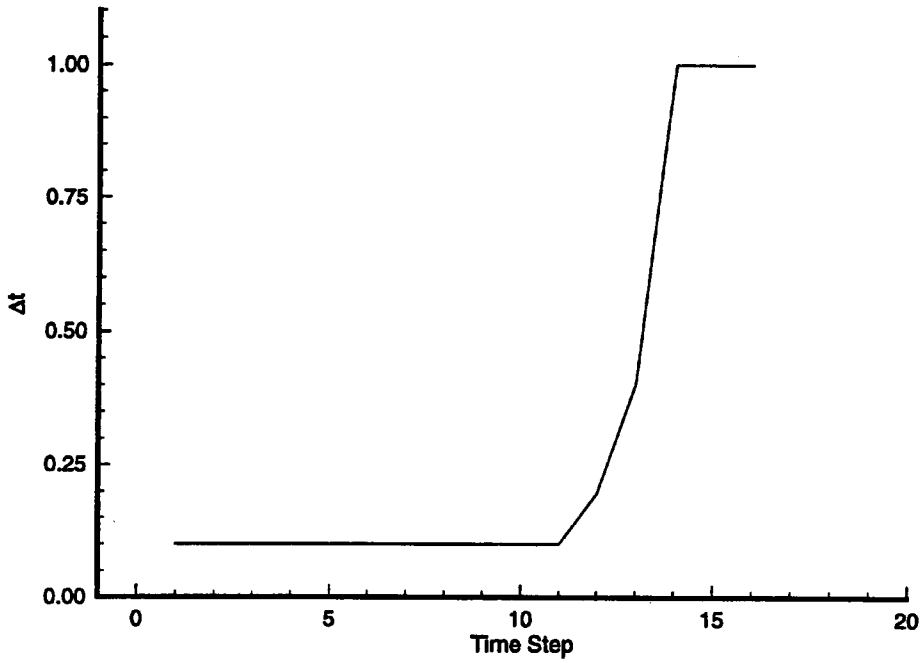


Fig. 6–22. Case 9 Time Step Size vs. Time Step

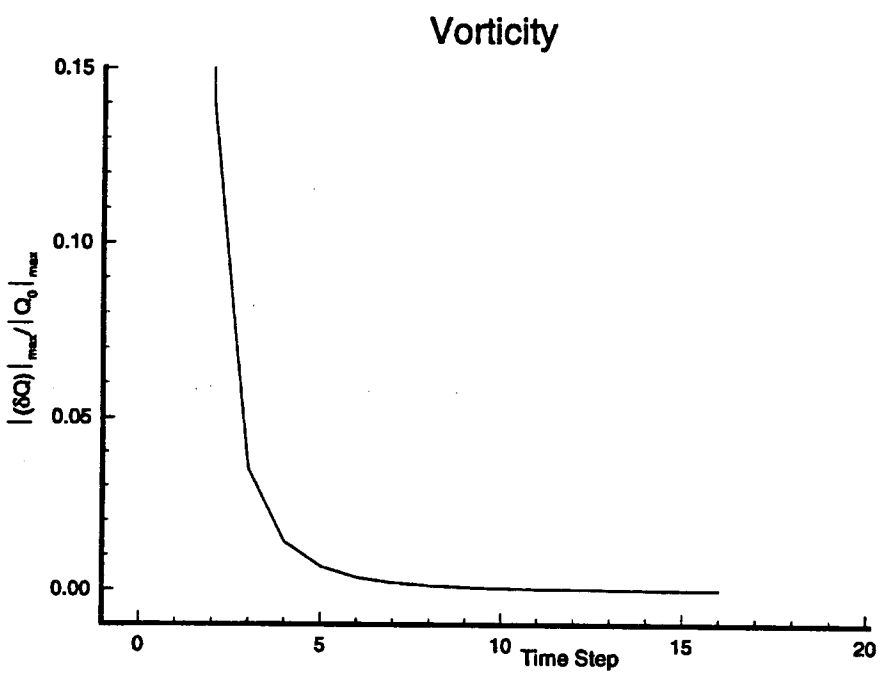


Fig. 6–23. Case 9 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Vorticity

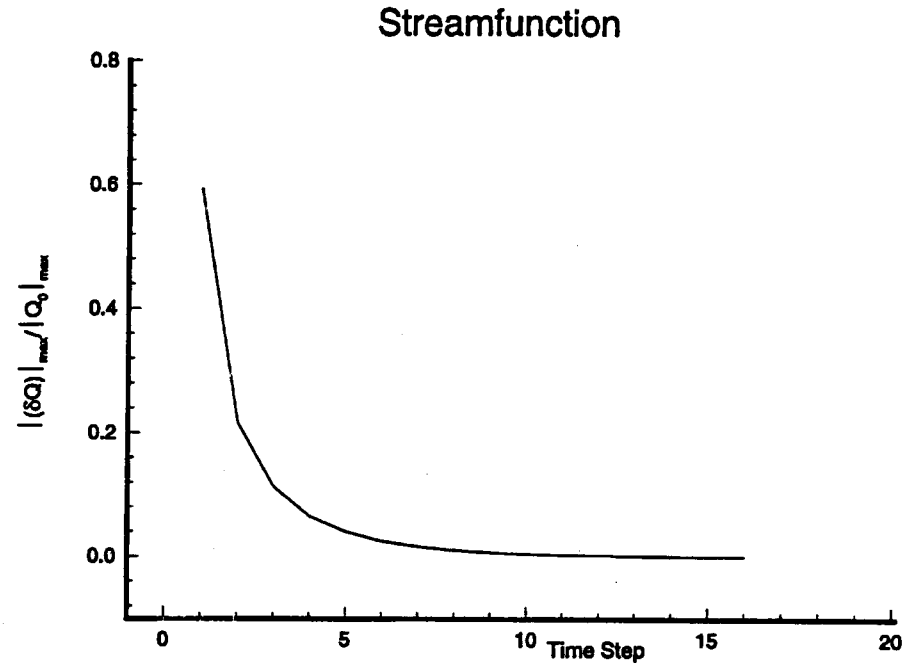


Fig. 6–24. Case 9 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Streamfunction

6.1.1.4 Summary, Re = 10 Driven Cavity

The conditionally stable CFD algorithm ($\theta = 0.0$) for the vorticity-streamfunction driven cavity cases requires limiting time step size to maintain a stable solution. The new time step control method, which selects the time step size based upon the characteristics of the solution, provides the necessary control to maintain stability of the solution. The original time step controller was not applicable to the $\theta = 0.0$ definition, hence would not maintain a stable solution.

Selecting an unconditionally stable integration ($\theta = 0.5$ and $\theta = 1.0$), the driven cavity cases will eventually converge regardless of the time step size for Re = 10. However, proper selection of the time step size will ensure that the convergence process occurs in an efficient manner, i.e., with minimal CPU time. Based on the $\theta = 0.5$ cases presented in Sect. 6.1.1.2, the original time step control method provides a more rapid convergence to steady-state convergence than does the new time step control method, as long as time accuracy is not desired. Time-accurate solutions are achievable using $\theta = 0.5$. Both methods can provide for efficient achievement for time-accurate evolution by selecting limiters appropriately.

The tests using the $\theta = 1.0$ implicit time integration demonstrate two major points. First, $\theta = 1.0$ allows the most rapid convergence to a steady-state solution since the algorithm contains a diffusion mechanism for limiting iterate oscillation. Of course, time-accuracy is lost as a consequence of $\theta = 1.0$ and large time steps. Second, both time step control methods are effective in controlling the time step size to achieve efficient steady-state convergence at Re = 10. The data summary for the Re = 10 2D vorticity-streamfunction driven cavity cases is given in Table 6-2.

Table 6-2. Summary of Re = 10 Vorticity-Streamfunction Driven Cavity Cases

Case No.	Method[1]	Initial Time Step Size	Maximum Time Step Size	Minimum Time Step Size	θ	Results Summary
1	Original (1.1)	0.002	1.0	N/A	0.0	Diverged
2	New $(\lvert(\delta Q)/Q\rvert_{max})^2$	0.002	1.0	0.001	0.0	Converged after 215 time steps in a total CPU time of 3228 s
3	New $(\lvert(\delta Q)\rvert_{max}/\lvert Q\rvert_{max})^2$	0.002	1.0	0.001	0.0	Converged after 224 time steps in a total CPU time of 3351 s
4	New $(\lvert(\delta Q)\rvert_{max}/\lvert Q\rvert_{max})^2$	0.0005	0.002	0.00025	0.0	Converged after 475 time steps in a total CPU time of 7121 s and a total problem time of 0.77
5	Original (1.1)	0.0005	1.0	N/A	0.5	Converged after 61 time steps in a total CPU time of 1510 s and a total problem time of 1.67
6	New $(\lvert(\delta Q)\rvert_{max}/\lvert Q\rvert_{max})^2$	0.0005	1.0	0.00025	0.5	Converged after 240 time steps in a total CPU time of 5948 s and a total problem time of 2.49
7	New $(\lvert(\delta Q)\rvert_{max}/\lvert Q\rvert_{max})^2$	0.0005	0.002	0.00025	0.5	Converged after 477 time steps in a total CPU time of 7202 s and a total problem time of 0.77
8	Original (1.1)	0.1	1.0	N/A	1.0	Converged after 14 time steps in a total CPU time of 383 s
9	New $(\lvert(\delta Q)\rvert_{max}/\lvert Q\rvert_{max})^2$	0.1	1.0	0.05	1.0	Converged after 16 time steps in a total CPU time of 438 s

Notes:

[1]If the original method is designated, the term in parentheses represents the time step multiplier. If the new method is designated, the term in the parentheses represents the values used to control the time step method.

[2]Both vorticity and streamfunction first-iterates were used.

6.1.2 2D Driven Cavity, Re = 100

The 2D vorticity-streamfunction driven cavity benchmark was executed for Re = 100 to validate the effectiveness of the time step controller. $\theta = 0.0$, $\theta = 0.5$, and $\theta = 1.0$ tests are documented in this section.

6.1.2.1 $\theta = 0.0$ Cases

Two 2D vorticity-streamfunction driven cavity cases were run using $\theta = 0.0$ explicit time integration at Re = 100. Case 10 used the original time step controller with an initial time step size of 0.001, a maximum time step size of 0.05, and a time step multiplier of 1.05. The solution diverged, as expected, see Table 6–3. As in Case 1, this controller is not sensitive to explicit integration stability criterion, hence is not of use for $\theta = 0.0$.

Case 11 was run using the new time step control method employing the same initial and maximum time step data, and selecting $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ for both vorticity and streamfunction first solutions for determining the size of the time step. This case converged after 667 time steps in a total CPU time of 9873 s.

Table 6-3. Case 10 Results

Time Step Number	Time Step Size	$\frac{(\delta\Omega)_{max}}{(\Omega_0)_{max}}$ for Vorticity	$\frac{(\delta\Omega)_{max}}{(\Omega_0)_{max}}$ for Streamfunction
111	5.000000E-02	1.618571E-01	-1.511037E-02
112	5.000000E-02	-3.213651E-01	-2.434729E-02
113	5.000000E-02	6.201753E-01	-4.191914E-02
114	5.000000E-02	-1.251515E+00	-7.730930E-02
115	5.000000E-02	2.303065E+00	-1.475853E-01
116	5.000000E-02	2.918834E+00	-1.904449E-01
117	5.000000E-02	-2.976649E+00	-2.945267E-01
118	5.000000E-02	2.962090E+00	3.710929E-01
119	5.000000E-02	-2.986832E+00	-5.864861E-01
120	5.000000E-02	2.974353E+00	7.291535E-01
121	5.000000E-02	-2.980804E+00	-1.166764E+00
122	5.000000E-02	-2.994681E+00	-1.446259E+00
123	5.000000E-02	2.990803E+00	-2.346933E+00
124	5.000000E-02	-2.967336E+00	-2.840568E+00
125	5.000000E-02	-4.7106737E+00	5.590531E+00
126	5.000000E-02	-3.431073E+01	-5.019201E+01
127	5.000000E-02	1.377769E+03	1.486110E+03
128	5.000000E-02	4.079955E+07	5.816701E+06

Table 6-3. (continued)

Time Step Number	Time Step Size	$\frac{(\delta Q)_{max}}{(Q_0)_{max}}$ for Vorticity	$\frac{(\delta Q)_{max}}{(Q_0)_{max}}$ for Streamfunction
129	4.761905E-02	1764507E+13	1.851886E+13
130	4.535147E-02	-6.806164E+26	-7.419084E+26
131	4.319188E-02	-5.566269E+53	-2.988644E+53
132	4.113512E-02	-1.363919E+107	-2.011868E+107

The history of time step size for Case 11 is graphed in Fig. 6–25. The time step size grew regularly to the maximum allowable value over 530 time steps. Thereafter, to the final time step number 667, the time step size remained at 0.05, except for about a dozen periods in which the time step size decreased to maintain solution stability.

Data for $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ for vorticity and streamfunction first-iterates are plotted in Figs. 6–26 and 6–27, respectively. At time step 544, $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ for the vorticity solution began to increase. Eventually, it reached the value of $\epsilon_{set,high}$ causing the time step size to decrease. The decreased time step size then allowed $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ for vorticity to fall below $\epsilon_{set,low}$ causing the time step size to increase back to the maximum. This pattern

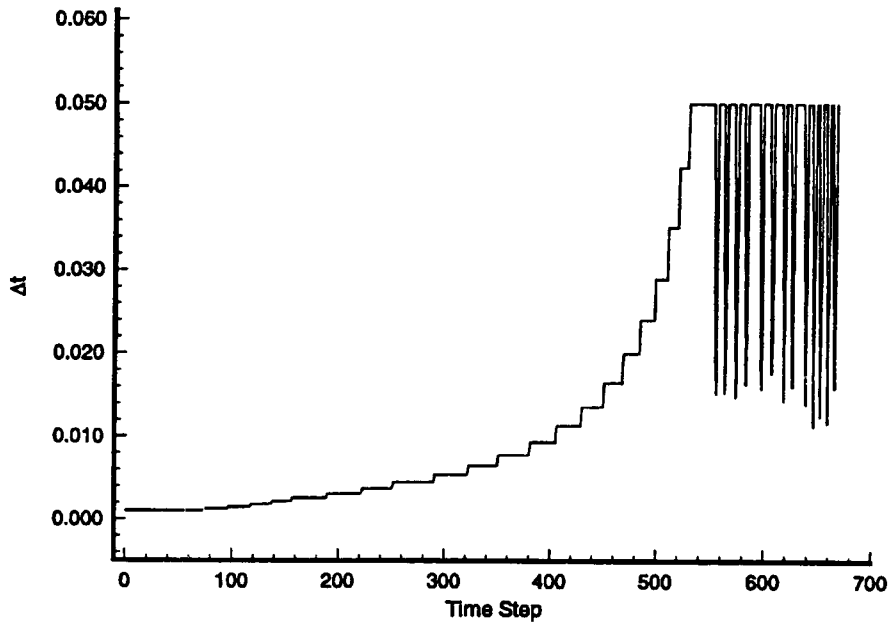


Fig. 6–25. Case 11 Time Step Size vs. Time Step

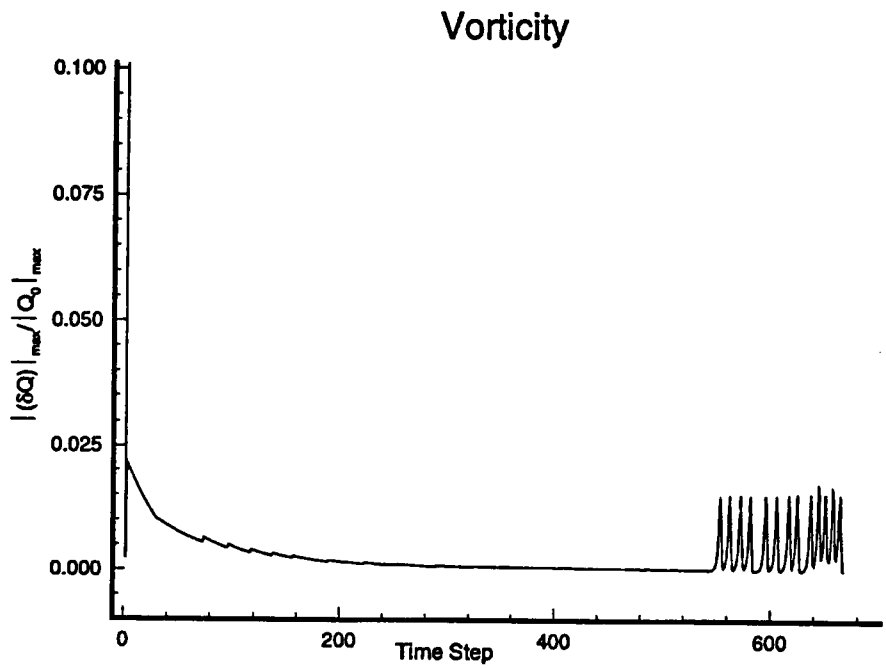


Fig. 6–26. Case 11 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Vorticity

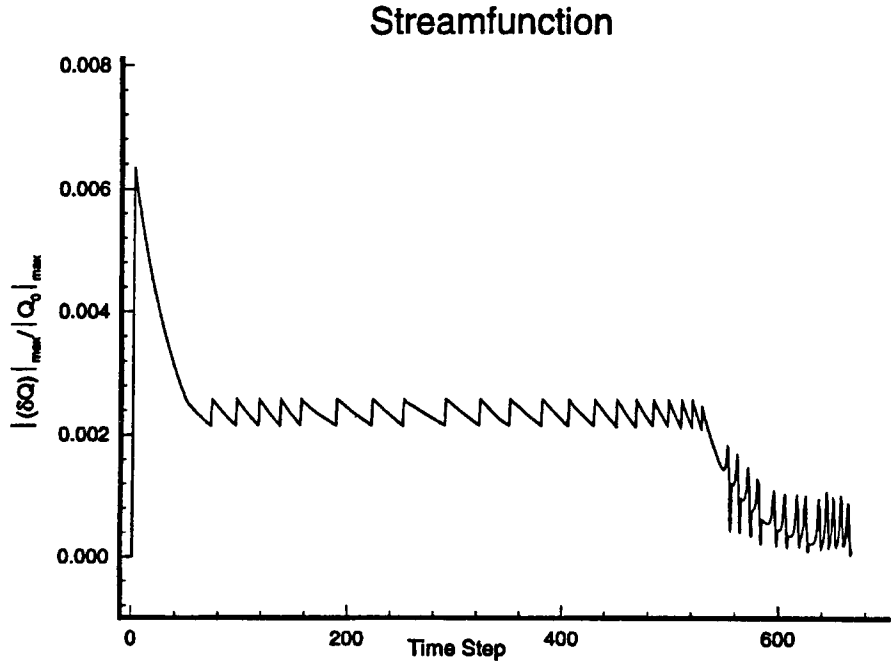


Fig. 6–27. Case 11 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Streamfunction

continued until eventually the steady-state convergence criterion was achieved for both the vorticity and streamfunction solutions.

Whereas the original time step control method allowed the solution to diverge, for the same initial and maximum time step size limits the new time step controller enabled convergence. Thus, for $\theta = 0.0$, the new time step control method provides the tool to guarantee achieving problem solution convergence with minimal user input and effort.

6.1.2.2 $\theta = 0.5$ Cases

The 2D vorticity-streamfunction driven cavity was run using the $\theta = 0.5$ time-integration algorithm at Re = 100. Case 12 used the original time step control method with an initial time step size of 0.01, a maximum time step size of 1.0, and a time step multiplier of 1.25. This case converged in 33 time steps in a total problem time of 16.40 non-D units and 515 s of CPU time.

The time step size history and $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ for vorticity and streamfunction first iterates for Case 12 are graphed in Figs. 6–28 through 6–30. The size of the time step increased to the maximum over the first 21 time steps, remained there for the next 10 steps, and then began to increase above the maximum limit because $|\delta Q|_{max}$ remained below the specified input for a sufficient number of iterative cycles. The $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ plots demonstrate a reasonably smooth curve, with some increasing values early and eventually decreasing values as steady-state convergence was reached.

Case 13 used the same initial and maximum time step sizes and employed the new time step control method based on $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ for both vorticity and streamfunction first-iterates. This case converged in 85 time steps in a total CPU time of 2171 s and a total problem time of 15.94 non-D units.

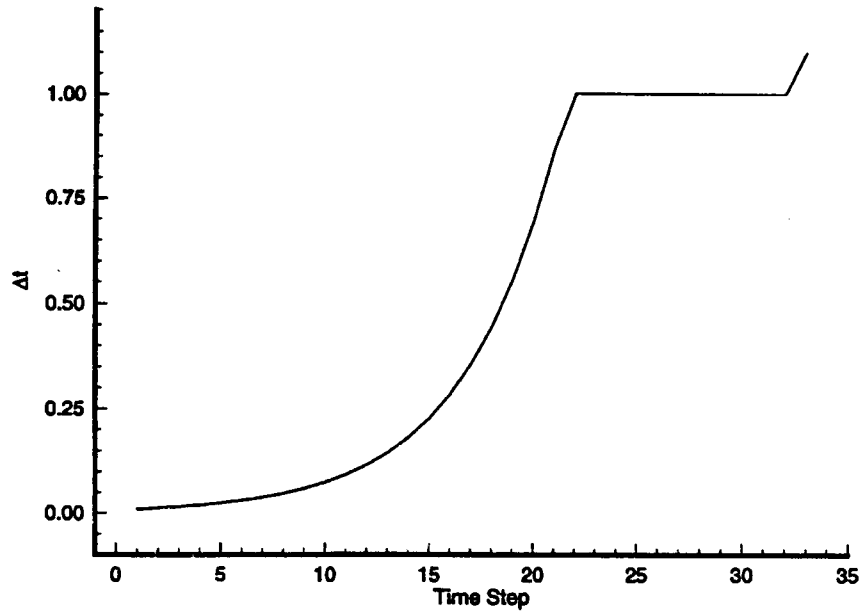


Fig. 6–28. Case 12 Time Step Size vs. Time Step

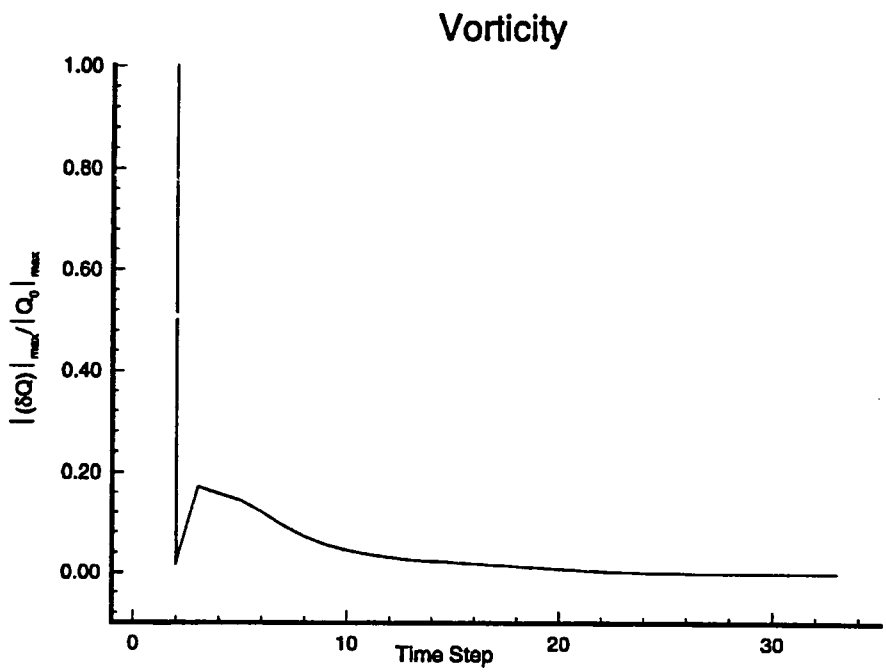


Fig. 6–29. Case 12 $|(\delta Q)|_{max} / |Q|_{max}$ Values vs. Time Step for Vorticity

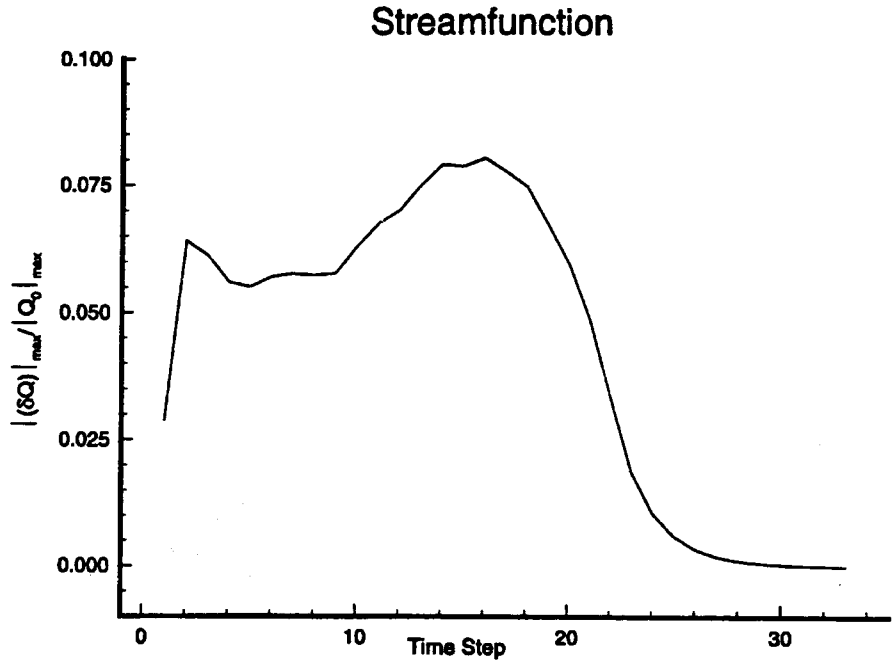


Fig. 6–30. Case 12 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Streamfunction

Time step size and $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ history for vorticity and streamfunction first-iterates are plotted for Case 13 in Figs. 6–31 through 6–33. The Δt values increased in a stepwise manner over 77 time step until the maximum size was reached. The vorticity and streamfunction first iterates were impacted by the changes in Δt, as noted by the sawtooth pattern. This pattern may seem to represent wasted CPU time; however, the apparent delays in problem convergence caused by the increases in time step size were offset by the greater negative slope that occurs for streamfunction over the final 10 time steps. If the time step size had not been increased, the slope would have been lesser in

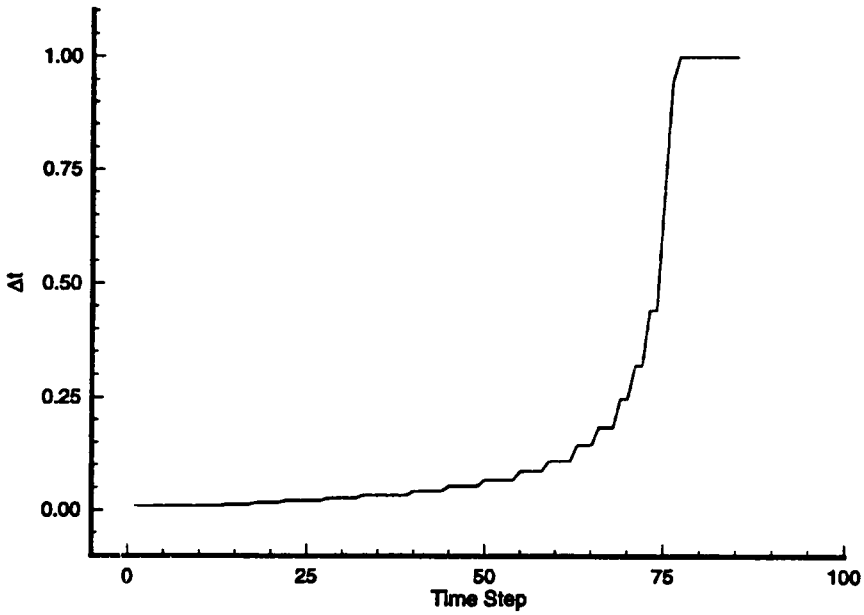


Fig. 6–31. Case 13 Time Step Size vs. Time Step

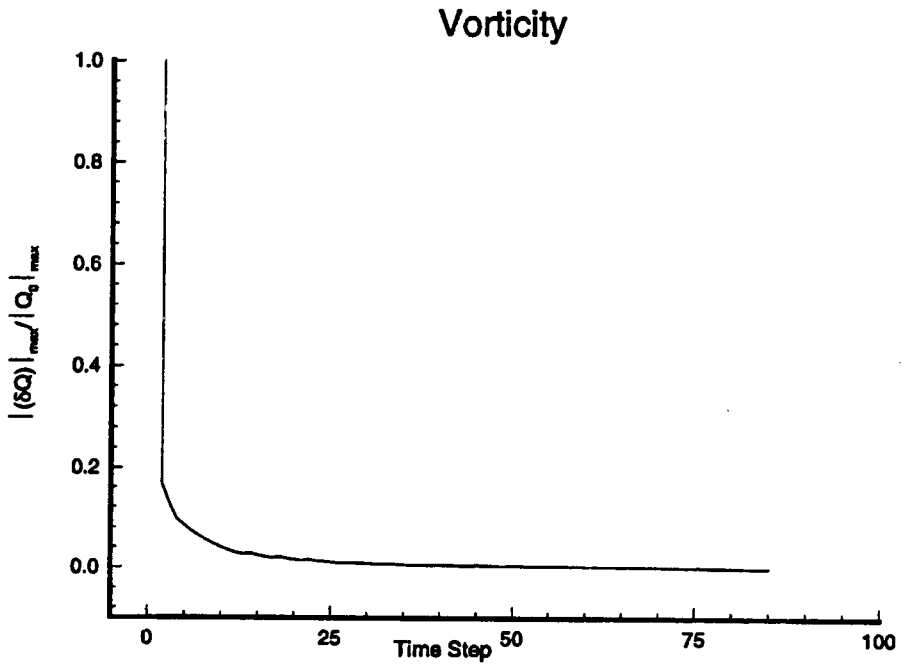


Fig. 6–32. Case 13 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Vorticity

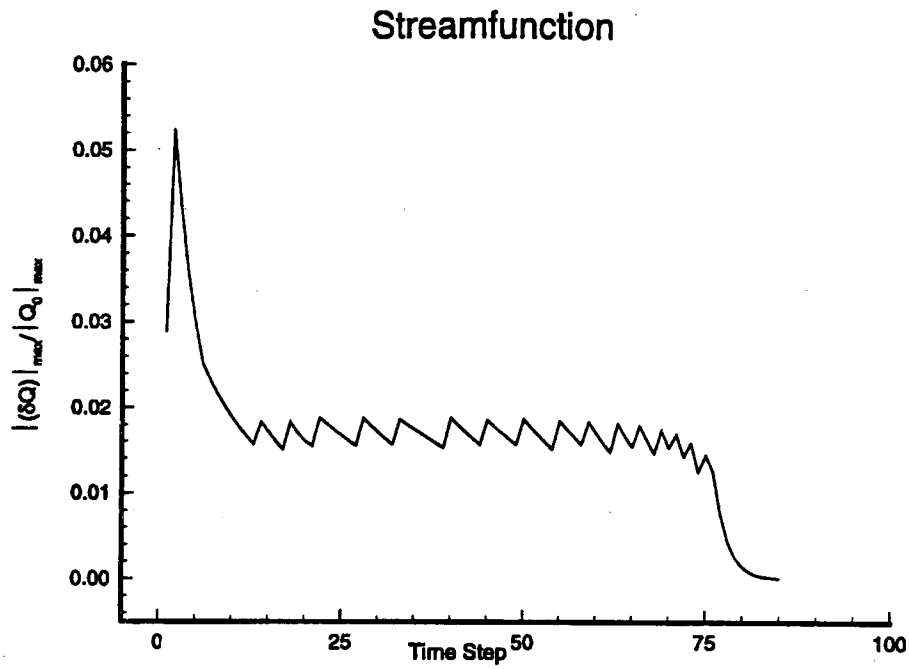


Fig. 6–33. Case 13 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Streamfunction

magnitude, thereby possibly increasing the CPU time necessary to achieve steady-state convergence. Once the maximum time step size was reached, the solution quickly converged to steady state.

Although the original method enabled the problem to achieve steady-state convergence in fewer time steps than the new method, both proved quite effective in reaching the steady-state solution using the $\theta = 0.5$ time-integration algorithm with minimal user interaction.

6.1.2.3 $\theta = 1.0$ Cases

Two 2D vorticity-streamfunction driven cavity cases were run using the $\theta = 1.0$ backwards implicit time integration at Re = 100. Case 14 used the original time step control method with an initial time step size of 0.01, a maximum time step size of 1.0, and a time step multiplier of 1.25. This case converged in 36 time steps in a total CPU time of 1215 s and a total problem time of 20.40 non-D units.

Time step size history and $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ for both vorticity and streamfunction first iterates are graphed for Case 14 in Figs. 6–34 through 6–36. These plots show reults that are quite similar to Case 12.

Case 15 used the same initial and maximum time step sizes as the previous case, but employed the new time step control method based on $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ for both vorticity and streamfunction first-iterates. This case reached steady-state convergence in 89 time steps in 2289 s CPU time for a total problem time of 18.54 non-D units.

Plots of the time step size and $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ for vorticity and streamfunction are provided for Case 15 in Figs. 6–37 through 6–39. The plots demonstrate results that are very similar to Case 13.

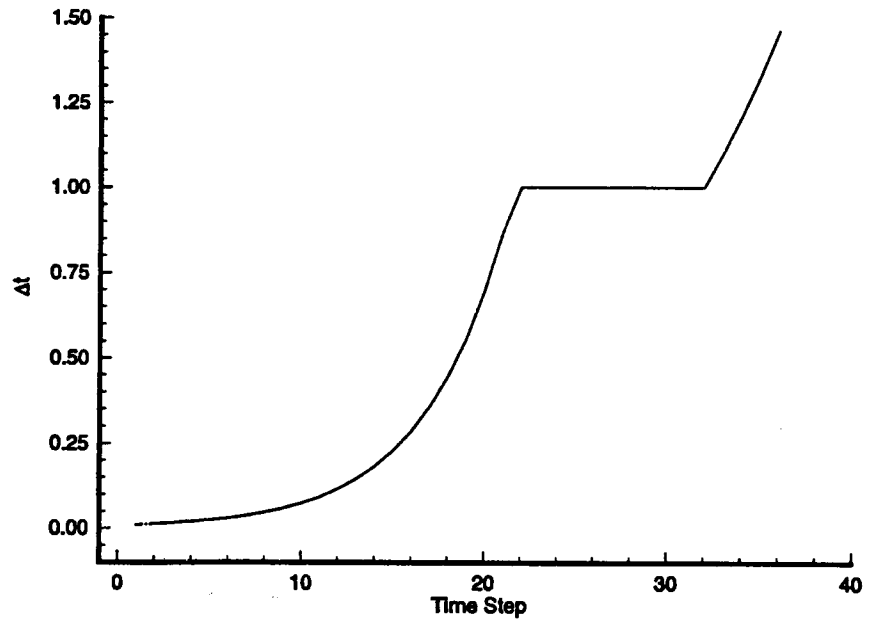


Fig. 6–34. Case 14 Time Step Size vs. Time Step

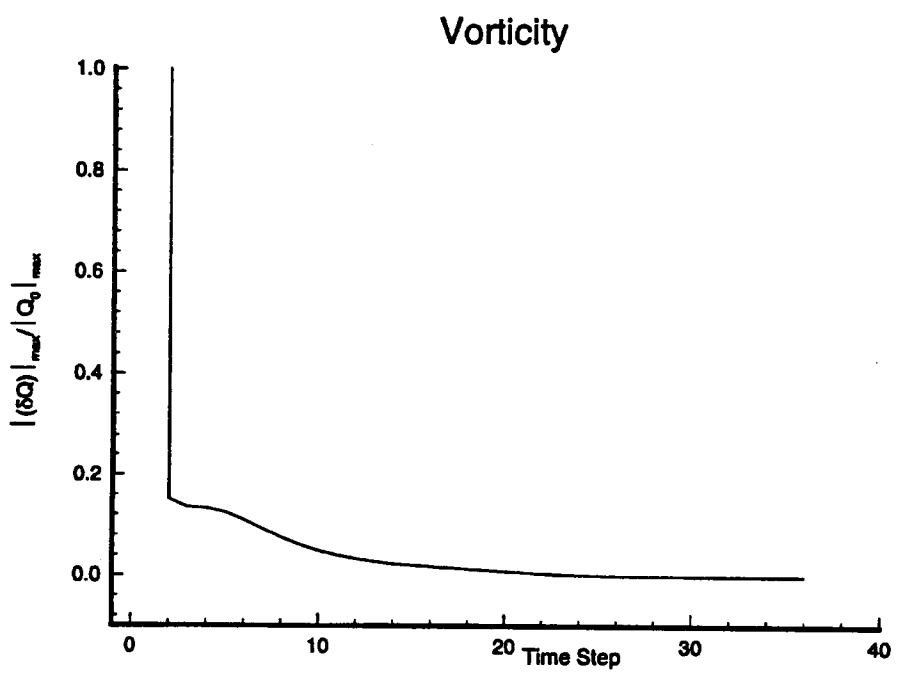


Fig. 6–35. Case 14 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Vorticity

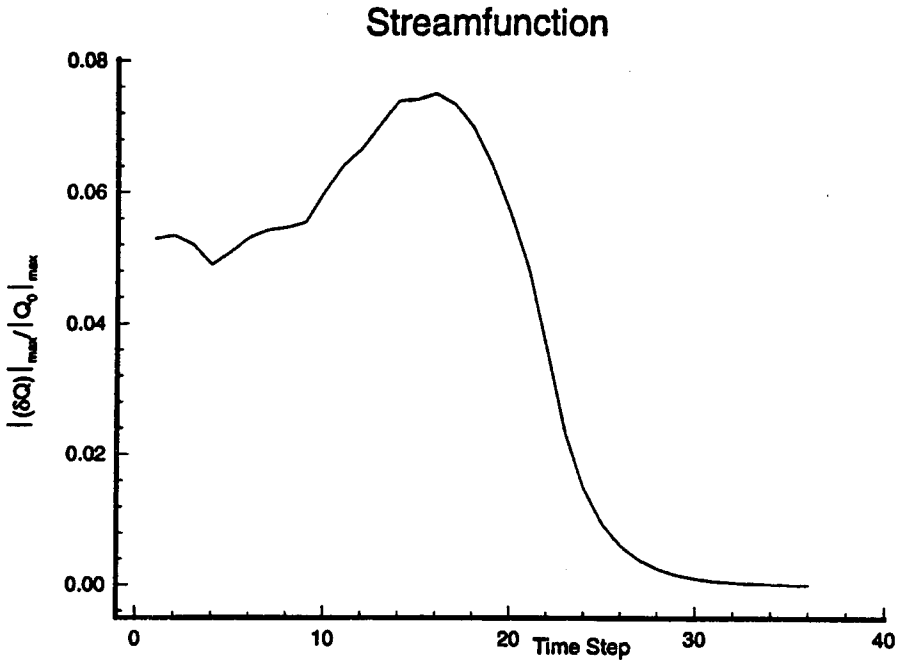


Fig. 6–36. Case 14 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Streamfunction

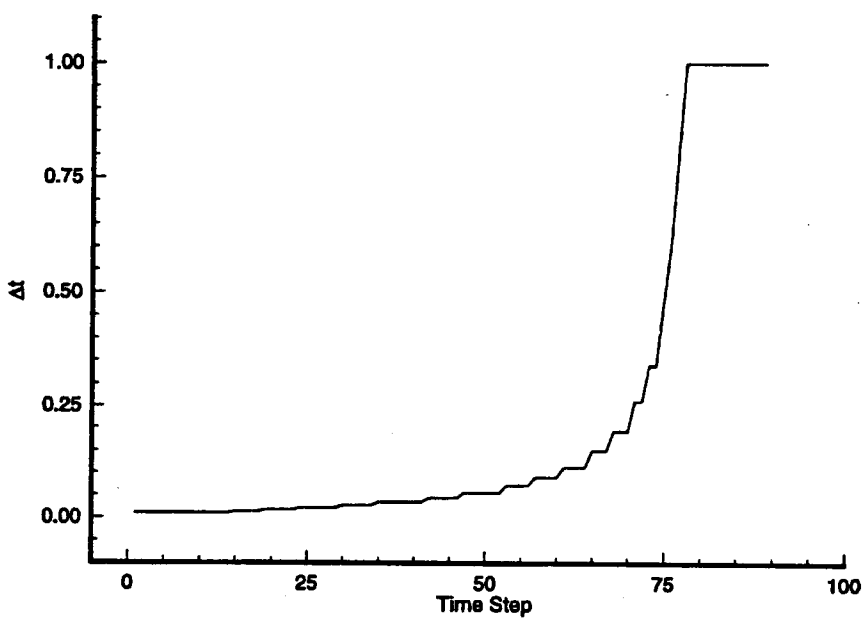


Fig. 6–37. Case 15 Time Step Size vs. Time Step

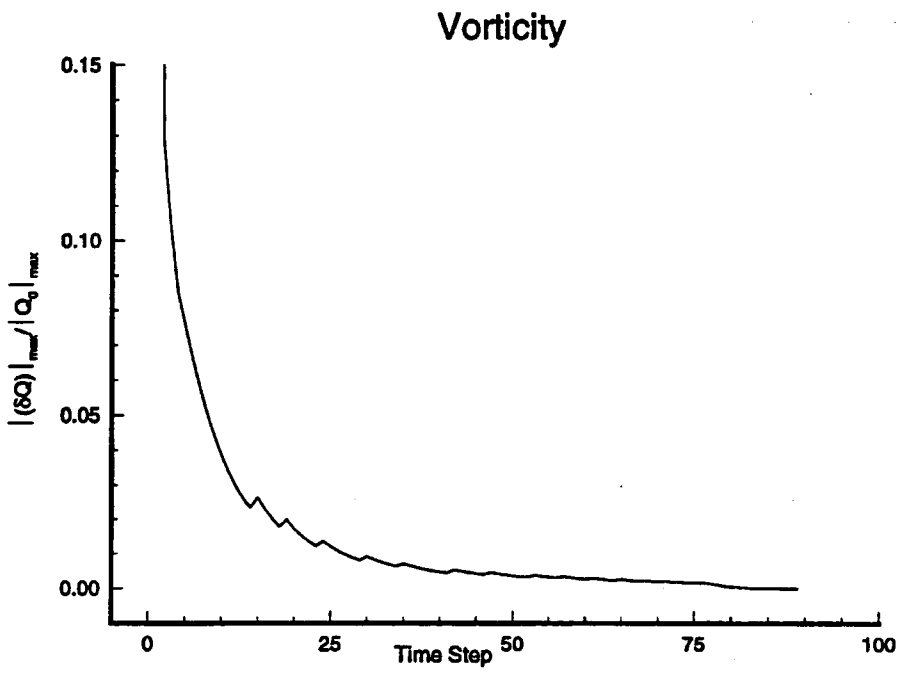


Fig. 6–38. Case 15 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Vorticity

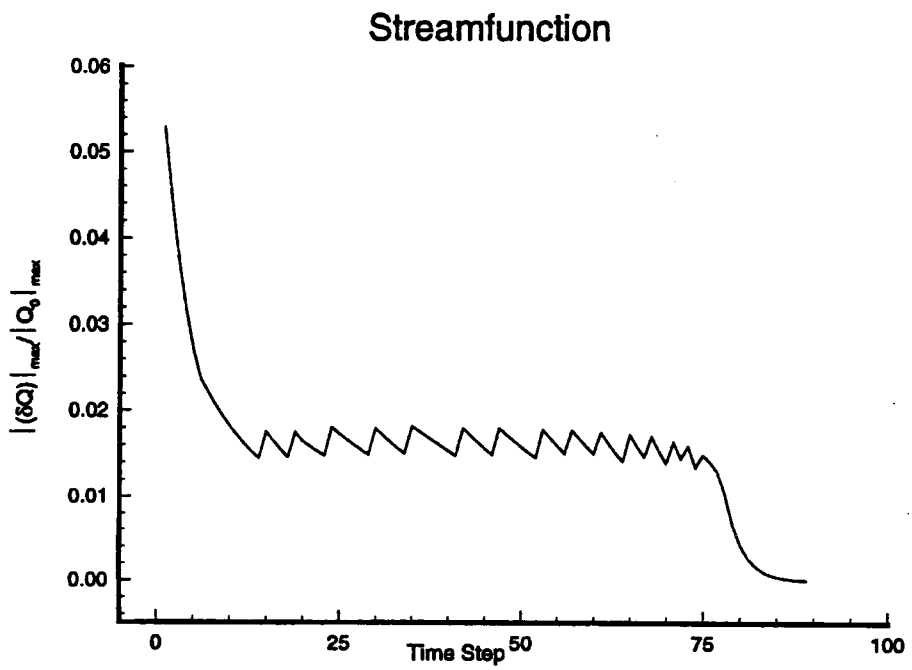


Fig. 6–39. Case 15 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Streamfunction

As was seen in the $\theta = 0.5$ cases, the original method enabled the problem to achieve steady-state convergence in fewer time steps than the new method. Nonetheless, both methods are effective in reaching the steady-state solution using the $\theta = 1.0$ time integration with minimal user interaction.

6.1.2.4 Summary, Re = 100 Driven Cavity

The Re = 100 vorticity-streamfunction driven cavity cases revealed results similar to the Re = 10 cases. For the conditionally stable algorithm ($\theta = 0.0$), the original time step control method failed to maintain the stability of the solution, while the new time step control method enabled the solution to converge to steady state. Thus, the new time step control method is essential for the efficient and effective solution of problems using explicit time integration.

For the unconditionally stable algorithms ($\theta = 0.5$ and $\theta = 1.0$), the original time step control method tends to reach convergence faster (i.e., in fewer time steps) than the new method. Nonetheless, both methods are effective in achieving convergence. A summary of the Re = 100 2D vorticity-streamfunction driven cavity cases is provided in Table 6-4.

Table 6-4. Summary of Re = 100 Vorticity-Streamfunction Driven Cavity Cases

Case No.	Method[1]	Initial Time Step Size	Maximum Time Step Size	Minimum Time Step Size	θ	Results Summary
10	Original (1.05)	0.001	0.05	N/A	0.0	Diverged
11	New $(\lvert(\delta Q)\rvert_{max}/\lvert Q\rvert_{max})^2$	0.001	0.05	0.0005	0.0	Converged after 667 time steps in a total CPU time of 9873 s
12	Original (1.25)	0.01	1.0	N/A	0.5	Converged after 33 time steps in a total CPU time of 515 s and a total problem time 16.40
13	New $(\lvert(\delta Q)\rvert_{max}/\lvert Q\rvert_{max})^2$	0.01	1.0	0.005	0.5	Converged after 85 time steps in a total CPU time of 2171 s and a total problem time of 15.94
14	Original (1.25)	0.01	1.0	N/A	1.0	Converged after 36 time steps in a total CPU time of 1215 s and a total problem time of 20.40
15	New $(\lvert(\delta Q)\rvert_{max}/\lvert Q\rvert_{max})^2$	0.01	1.0	0.005	1.0	Converged after 89 time steps in a total CPU time of 2289 s and a total problem time of 18.54

Notes:

[1]If the original method is designated, the term in parentheses represents the time step multiplier used in the case. If the new method is designated, the term in the parentheses represents the values used to control the time step method.

[2]Both the vorticity and streamfunction first-iterate fields were used.

6.1.3 2D Thermal Cavity, Ra = 1000

Several 2D thermal cavity cases were run at a Rayleigh (Ra) number of 1000 to evaluate the effectiveness of the new time step control method. Whereas the presented driven cavity cases used a Newton template, in which all variables were directly coupled, the thermal cavity cases employ a quasi-Newton template in which the energy equation is separate from the coupled vorticity-streamfunction solution. As before, executions using $\theta = 0.0$, $\theta = 0.5$, and $\theta = 1.0$ are provided for direct comparisons.

6.1.3.1 $\theta = 0.0$ Cases

Several 2D vorticity-streamfunction thermal cavity cases were run using explicit time integration at Ra = 1000. Case 16 used the new time step control method with an initial time step size of 0.0001, a maximum time step size of 0.0005, and a minimum time step size of 0.00005. To provide a variation from the previous cases and to satisfy curiosity, the value of $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ for the temperature first iterate only was used as the criterion to determine the time step size by the control method. This case ran over 400 time steps and then the solution diverged.

The time step size versus time step for case 16 is plotted in Fig. 6–40. Note that the time step size oscillated between time steps 20 and 350 before settling on a constant value. Apparently, this constant value was too large to maintain stability for vorticity and streamfunction, causing the problem to diverge. This case demonstrates the potential consequences of not basing the time step on all pertinent solution fields.

Thereafter, Case 17 was run with the same initial, minimum, and maximum time step sizes, but with the time step controller based on $\frac{|\delta\varrho|_{max}}{|\varrho_0|_{max}}$ from vorticity, streamfunction, and temperature first iterates. This case ran for 2000 time steps and neither converged nor diverged. The time step size, shown in Fig. 6–41, changed almost every time step because of oscillations in the $\frac{(\delta\varrho)_{max}}{(\varrho_0)_{max}}$ values in the vorticity, streamfunction, and temperature solution fields, see Figs. 6–42 through 6–44.

Unlike previous plots for the solution fields, Figs. 6–42 through 6–44 graph $\frac{(\delta\varrho)_{max}}{(\varrho_0)_{max}}$, not the $\frac{|\delta\varrho|_{max}}{|\varrho_0|_{max}}$ data actually used by the time step control method. Thus, focusing on vorticity, it is noticed that $\frac{(\delta\varrho)_{max}}{(\varrho_0)_{max}}$ generally oscillate about the line y = 0.0. The absolute values $\frac{|\delta\varrho|_{max}}{|\varrho_0|_{max}}$ create a relatively smooth curve, with minor oscillations (except for the period between time steps 400 and 900), as shown in Fig. 6–45.

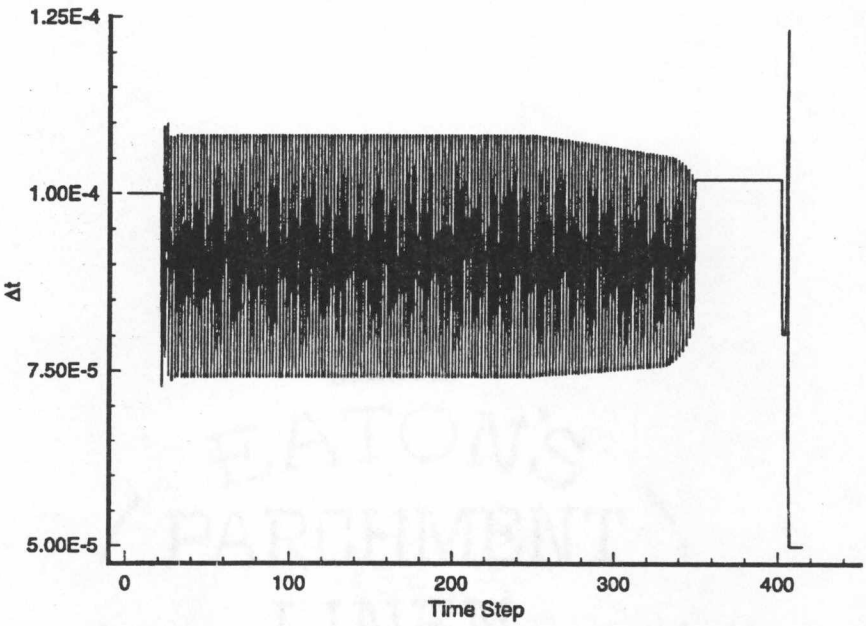


Fig. 6-40. Case 16 Time Step Size vs. Time Step

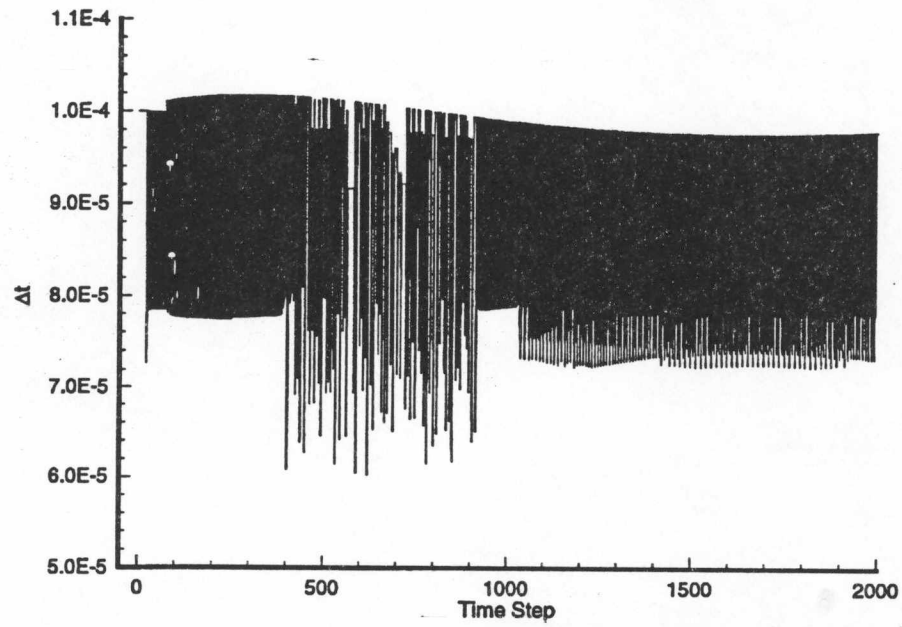


Fig 6–41. Case 17 Time Step Size vs. Time Step

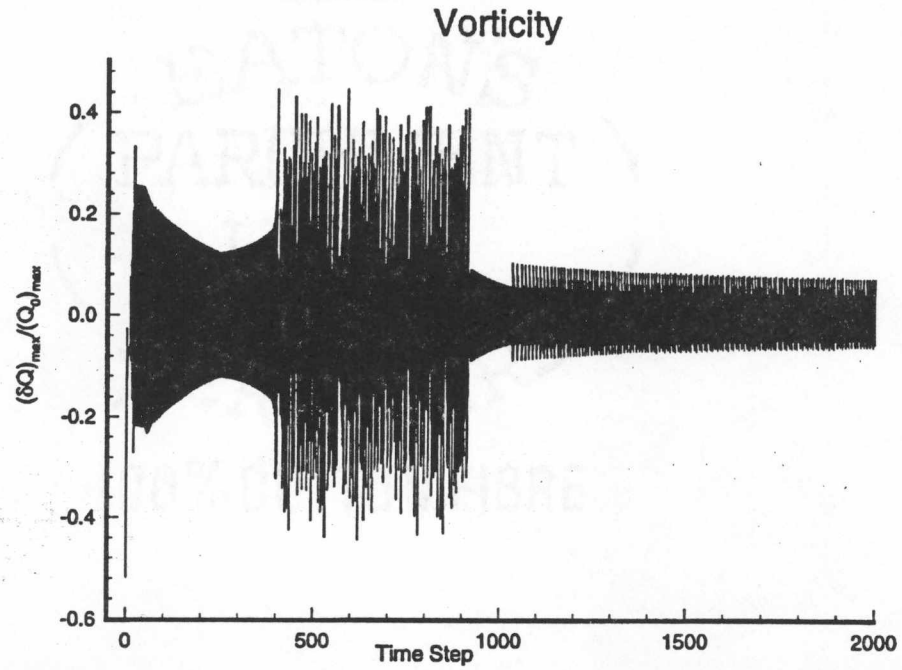


Fig. 6–42. Case 17 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Vorticity

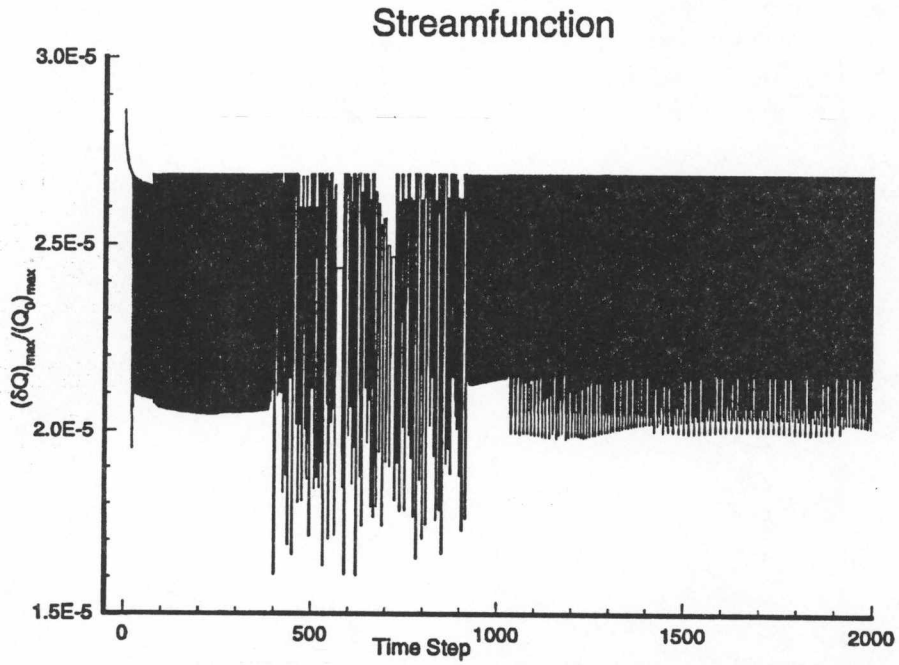


Fig. 6–43. Case 17 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Streamfunction

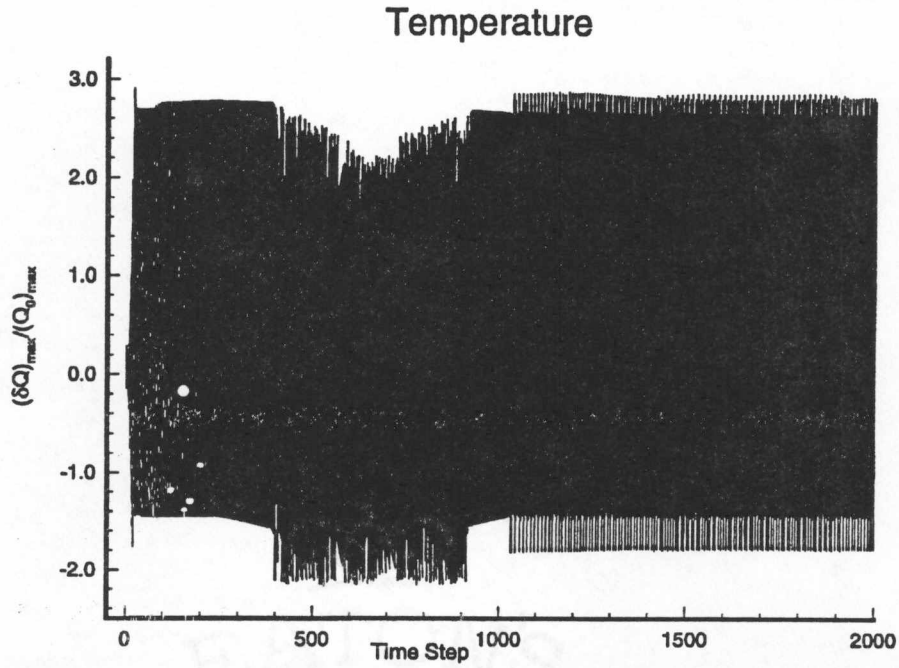


Fig. 6–44. Case 17 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Temperature

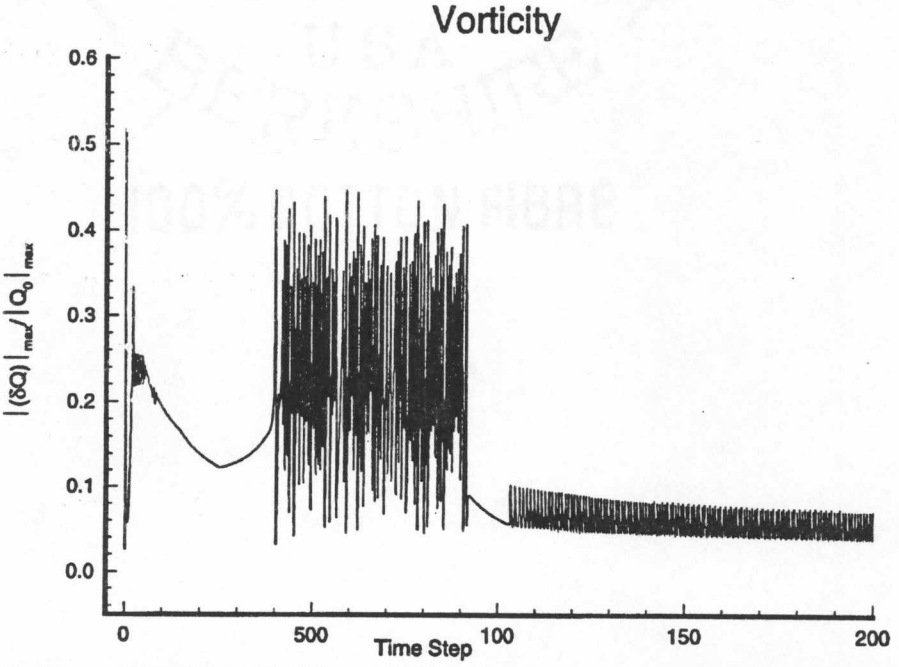


Fig. 6–45. Case 17 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Vorticity

If the solution time step control was based solely on $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ from the vorticity solution field, the time step size curve would be relatively smooth. Of course, as was learned from the results of Case 16, the solution may eventually diverge.

The $\frac{(\delta\Omega)_{max}}{(\Omega_0)_{max}}$ values in the streamfunction and temperature solution fields also oscillated, but not about the line y = 0. The streamfunction solution field remained positive at all times, generally oscillating about the line y = ~2.4×10^{-5}. The temperature solution field oscillated between positive and negative values, typically around the line y = ~0.6. Because these solution fields did not oscillate around the line y = 0.0, the absolute values of $\frac{(\delta\Omega)_{max}}{(\Omega_0)_{max}}$ for these fields changed significantly in magnitude with nearly each successive time step, especially in the temperature solution field. Therefore, the value of $\frac{|\delta\Omega|_{max}}{|\Omega_0|_{max}}$ is within the bounds of $\epsilon_{set,high}$ and $\epsilon_{set,low}$ at one time step, moves outside these bounds at the next time step, moves back within the bounds at the next, and so on. This characteristic causes the time step size to oscillate, as shown in Fig. 6–41.

It is difficult to determine if this case would ever converge to steady state. At best, it can be stated that the time step controller will prevent the solution from diverging but will not necessarily guarantee that it will converge. Clearly, this behavior is not efficient. Therefore, when such behavior occurs with $\theta = 0.0$, it is probably best to start over with a smaller time step size boundary.

Another case was run with smaller initial and maximum time step sizes (Case 18). This case used the same minimum time step size of 0.00005, but the initial and maximum time step sizes were reduced to 0.000075 and 0.0004, respectively. Fig. 6–46 shows the time step size for the first 2000 time steps for Case 18. Note the curve shows no oscillations and exhibits the desired "stair step" appearance. Figs. 6–47 through 6–49 show $\frac{(\delta Q)_{max}}{(Q_0)_{max}}$ for vorticity, streamfunction, and temperature, respectively, for Case 18. These data also show no oscillations.

The "sawtooth" pattern displayed by the streamfunction solution field coincides with the "stair steps" in the time step size plot. Initially, the values of $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ for the streamfunction solution field were decreasing, which caused the time step size to increase. However, around time step 320, these values began to increase, leading to successive reductions in the time step size. As the problem approached time step 2000, $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ began to decrease (although it is difficult to tell by the figure). It is concluded, therefore, that a time step size of about 7.5×10^{-5} is about the limit for this problem. The new time step control method will serve to keep the time step within this vicinity until the problem approaches convergence. Obviously, $\theta = 0.0$ is not a particularly good choice for this algorithm form.

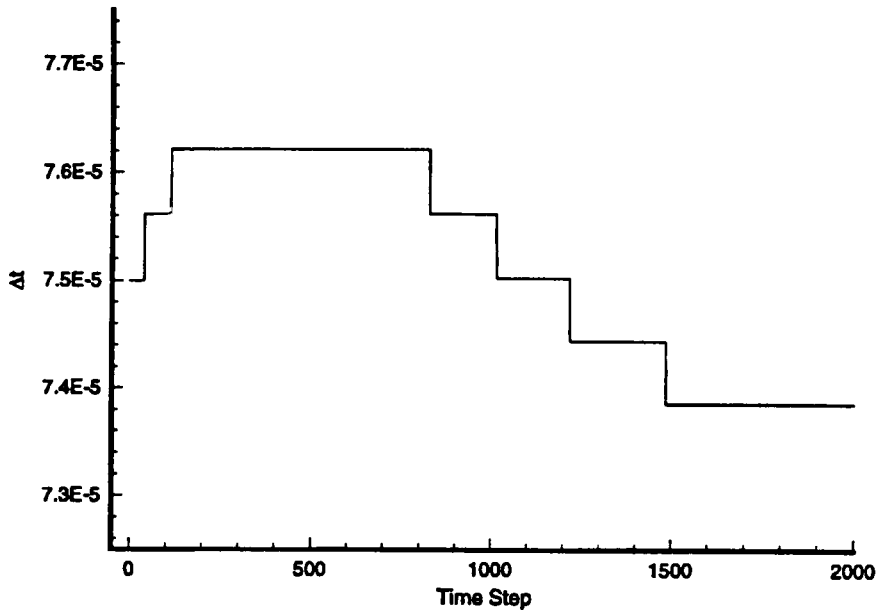


Fig. 6–46. Case 18 Time Step Size vs. Time Step

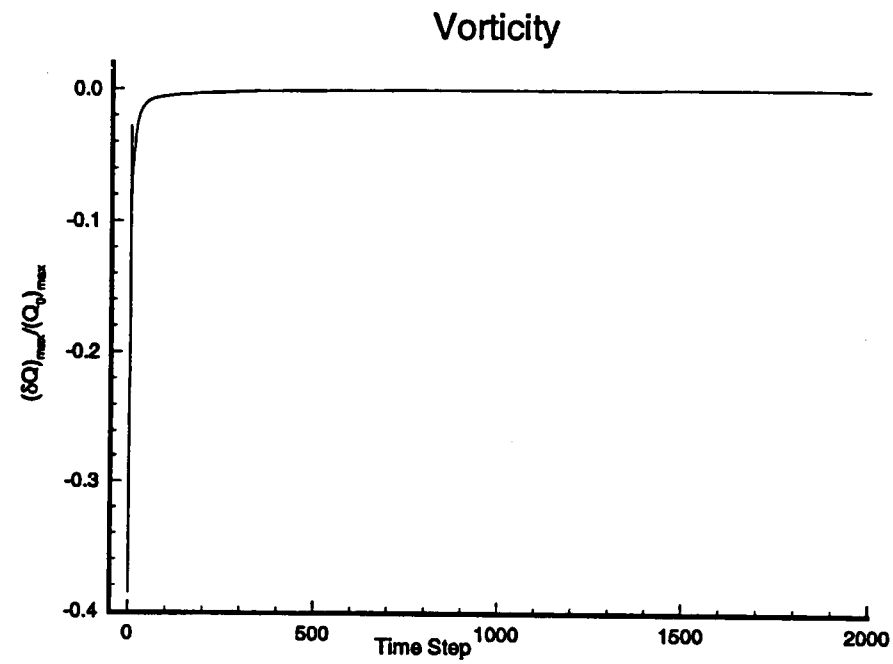


Fig. 6–47. Case 18 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Vorticity

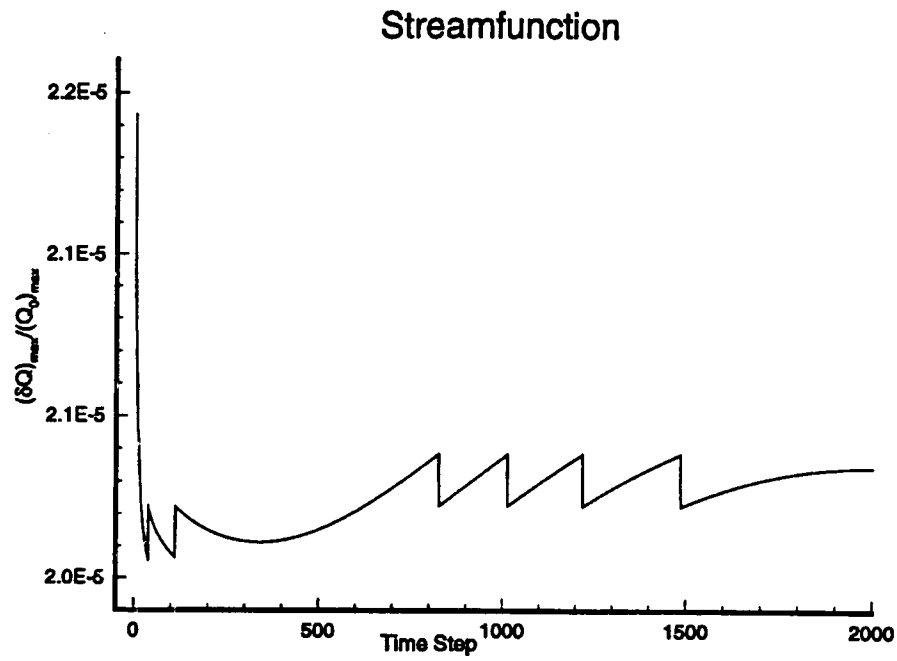


Fig. 6–48. Case 18 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Streamfunction

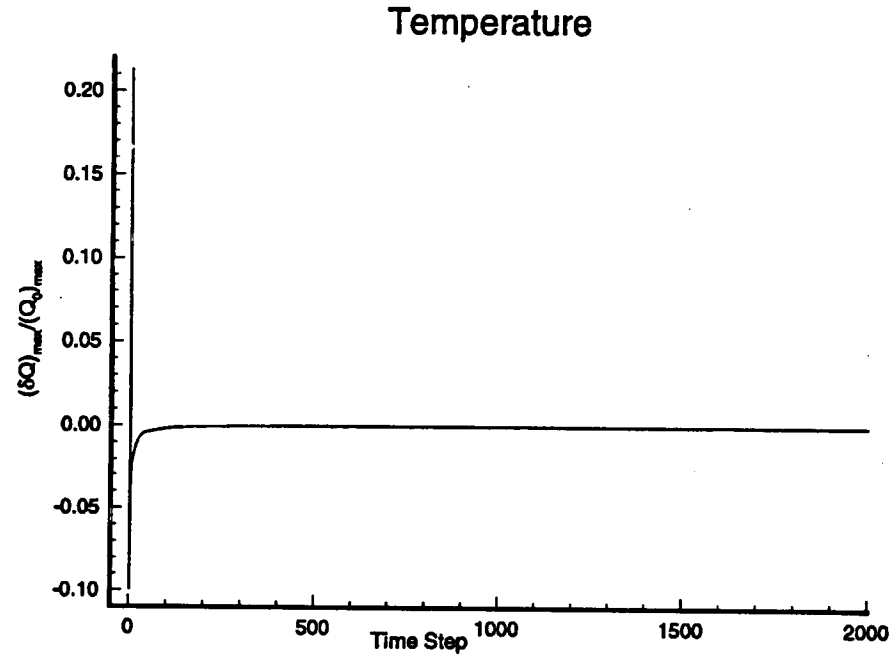


Fig. 6–49. Case 18 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Temperature

6.1.3.2 $\theta = 0.5$ Cases

Two 2D vorticity-streamfunction thermal cavity cases were run using the $\theta = 0.5$ implicit time integration at Ra = 1000. Case 19 used the original time step control method with an initial time step size of 0.01, a maximum time step size of 0.05, and a time step multiplier of 1.05. Based on this time step multiplier, the time step size quickly increased from its initial to its maximum value, see Fig. 6–50. The time step size remained at the maximum value until the time step number 2000 (the final input time station) was reached.

Figs. 6–51 through 6–53 show the values of $\frac{(\delta Q)_{max}}{(Q_0)_{max}}$ for the vorticity, streamfunction, and temperature first iterates. The vorticity and temperature solution fields exhibited oscillating convergence. Although the solution fields had not reached the steady-state convergence criterion by the 2000th time step, the solutions were clearly headed towards it.

Case 20 used the new time step control method based on $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ for vorticity, streamfunction, and temperature first iterates with the same initial and maximum time step sizes as the previous case. The minimum time step size was 0.005, and this case reached steady-state convergence in 1202 time steps and a total CPU time of 40,517 s.

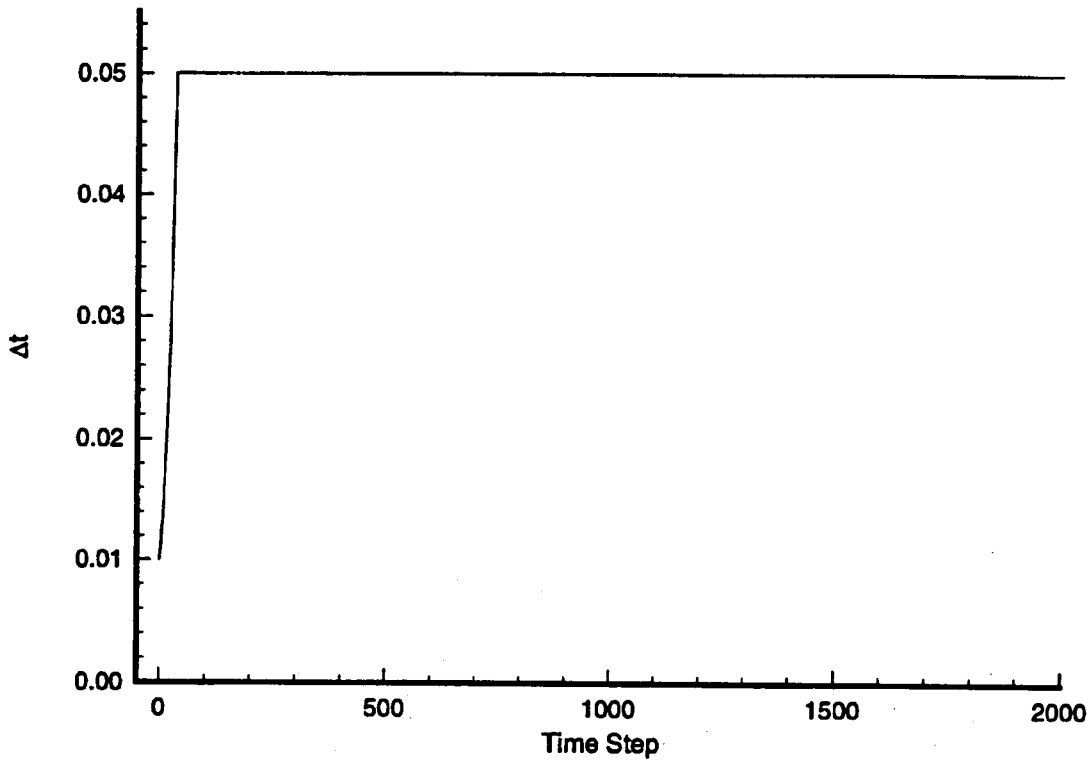
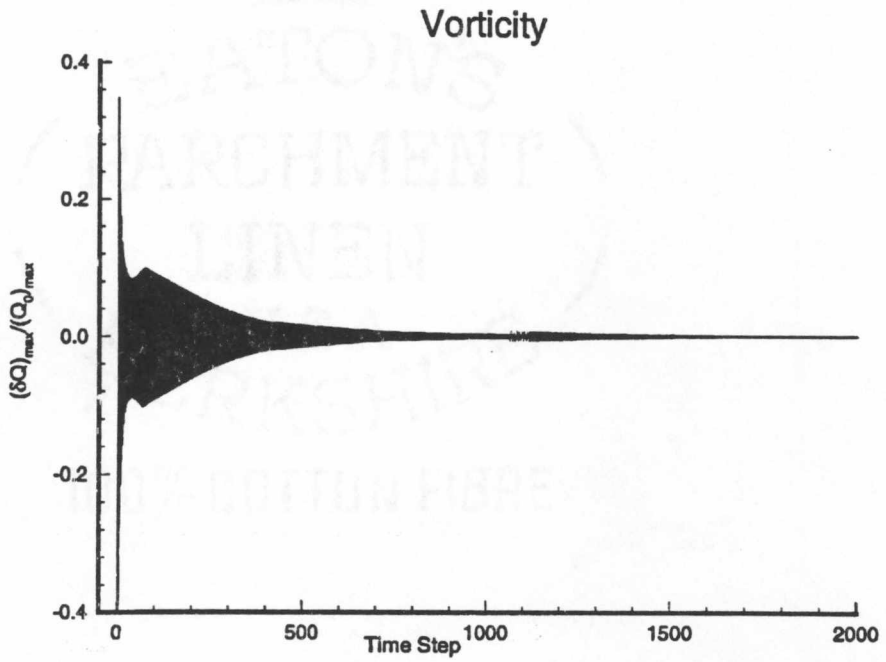


Fig. 6–50. Case 19 Time Step Size vs. Time Step



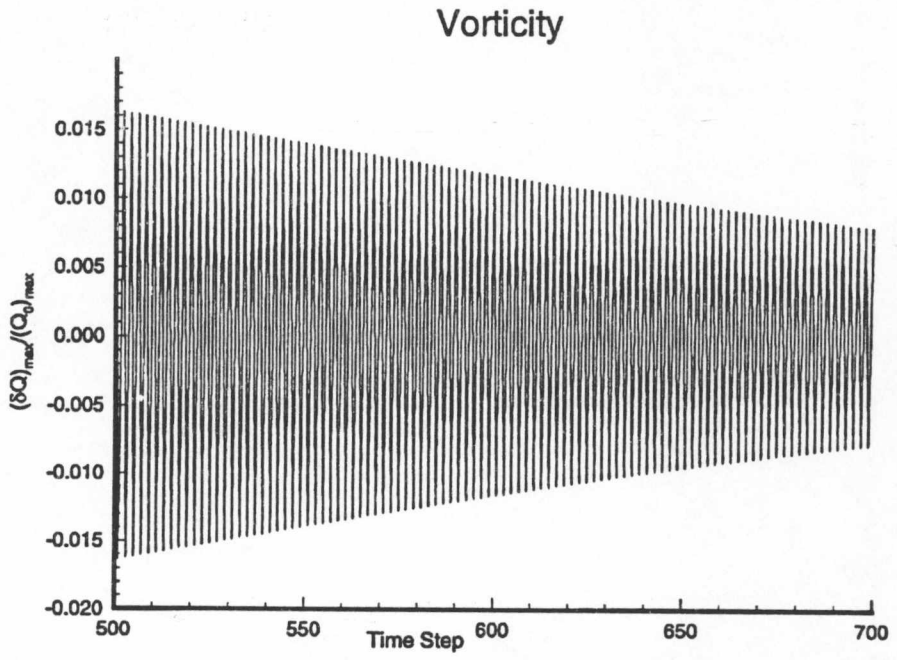


Fig. 6–51. Case 19 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Vorticity

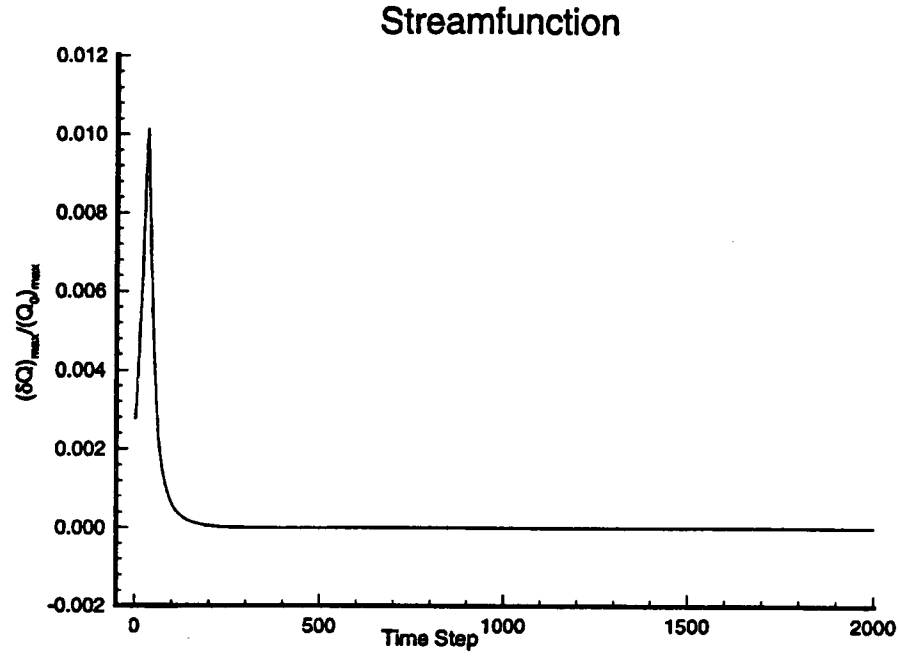


Fig. 6–52. Case 19 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Streamfunction

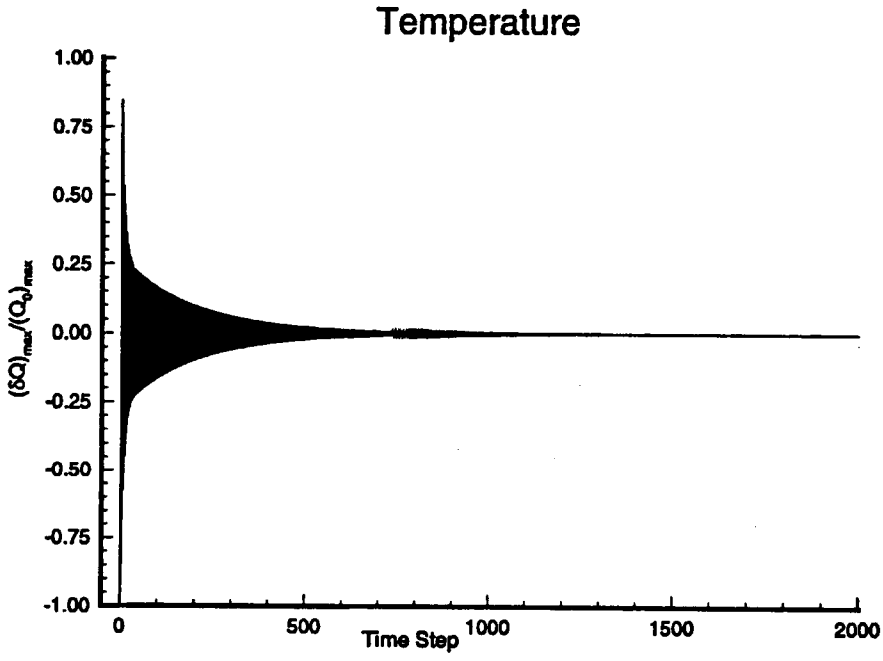


Fig. 6–53. Case 19 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Temperature

Fig. 6–54 shows the time step size history. The time step size fluctuated quite erratically, especially during the first 400 time steps, and again around time steps 550 and 675. Eventually, the time step size reached the maximum value and remained there until the problem solution converged.

Figs. 6–55 through 6–57 show $\frac{(\delta Q)_{max}}{(Q_0)_{max}}$ for vorticity, streamfunction, and temperature first iterates. All three solution fields exhibited erratic, fluctuating behavior during the first 200 or so time steps, after which a smooth oscillating behavior emerged until the steady-state convergence criterion was eventually reached.

For these two cases, the new time step control method enabled the steady state solution to be obtained in fewer time steps than the original method.

6.1.3.3 $\theta = 1.0$ Cases

Two 2D vorticity-streamfunction thermal cavity cases were run using the $\theta = 1.0$ implicit time integration at Ra = 1000. The first case (Case 21) used the original time step control method with an initial time step size of 0.025, a maximum time step size of 1.0, and a time step multiplier of 1.5. Based upon this time step multiplier, the time step size

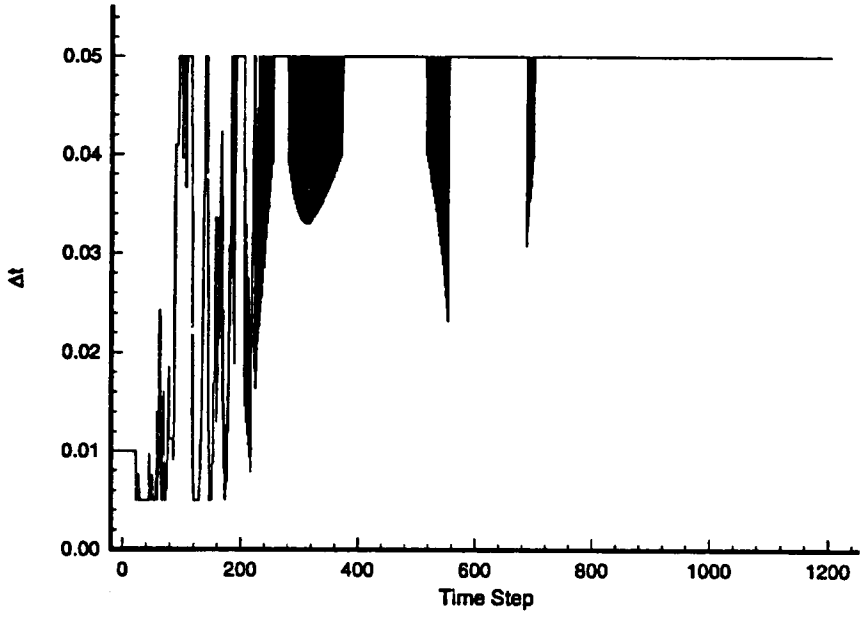
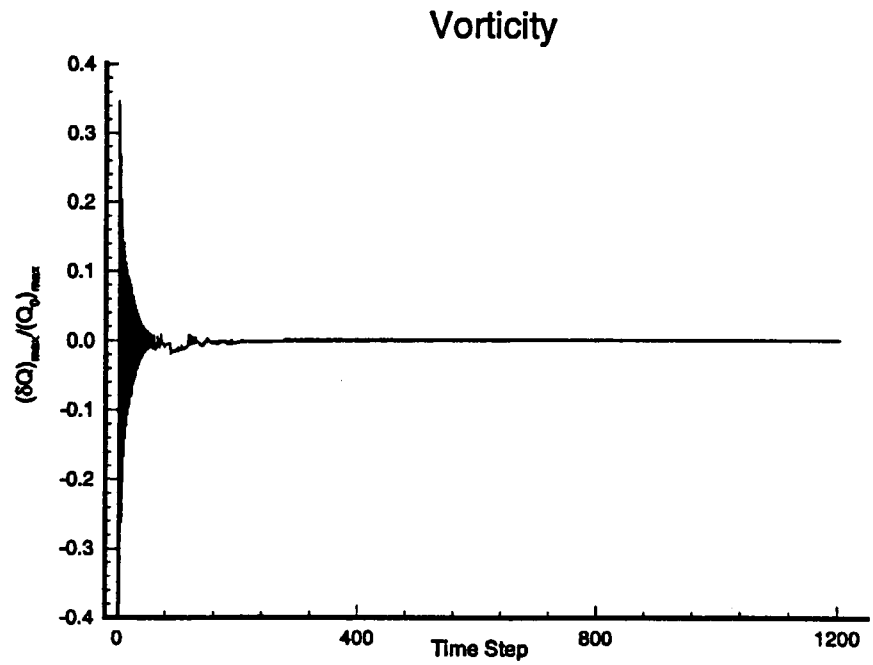


Fig. 6–54. Case 20 Time Step Size vs. Time Step



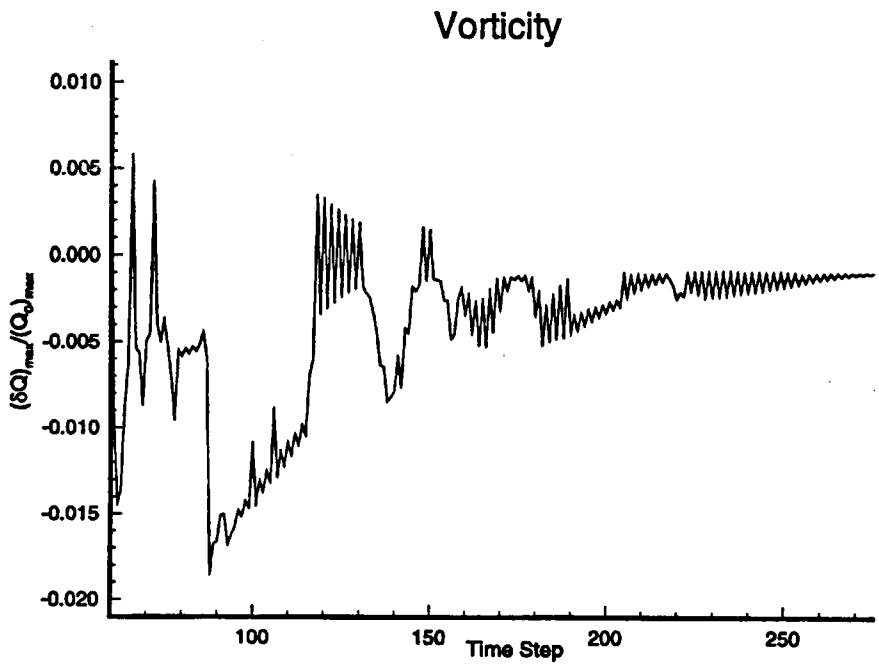


Fig. 6–55. Case 20 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Vorticity

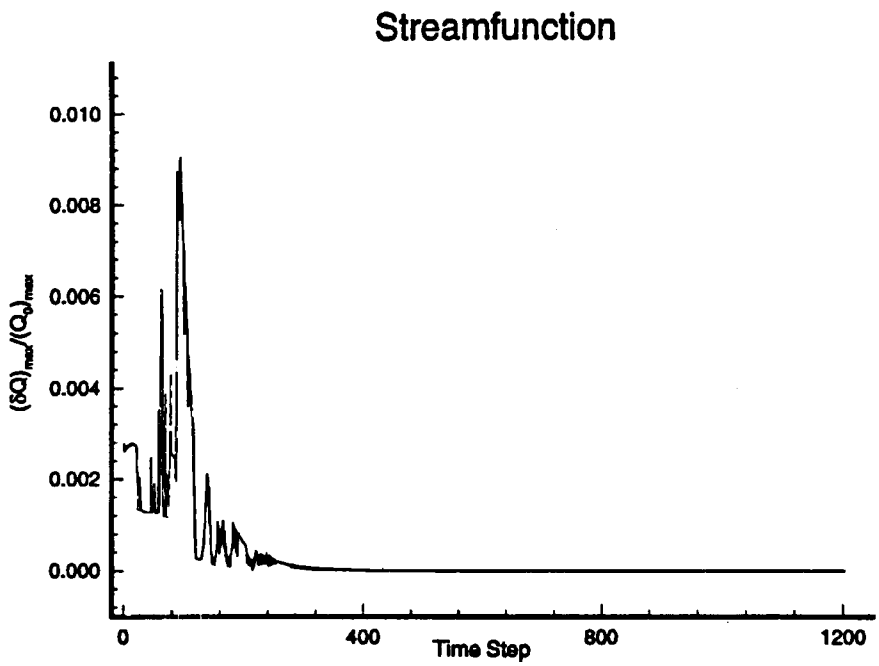


Fig. 6–56. Case 20 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Streamfunction

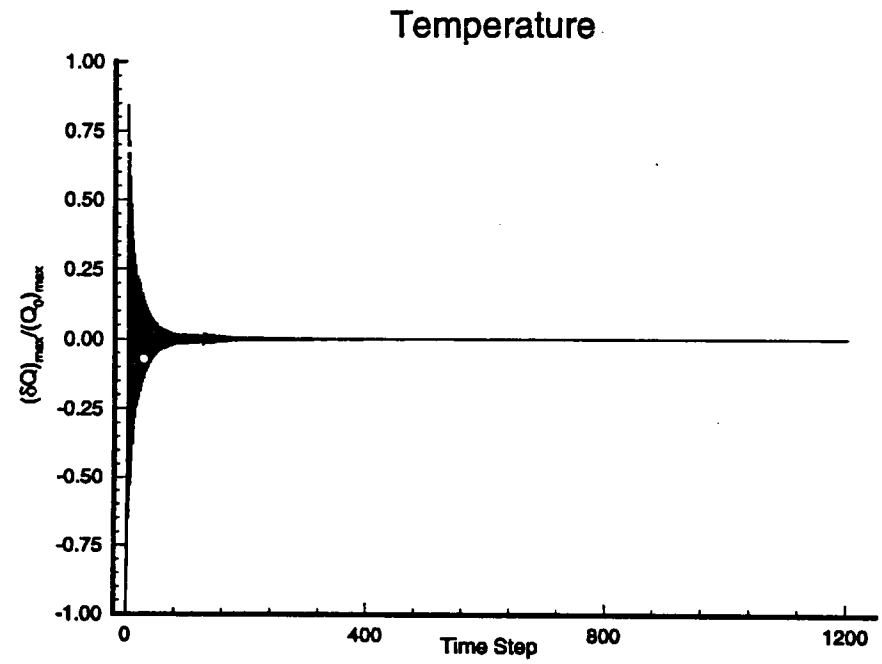


Fig. 6–57. Case 20 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Temperature

quickly increased from its initial value to the maximum value (and beyond, similar to Case 12) as shown in Fig. 6–58.

The time step size remained at its maximum value for about 10 time steps, and then started to increase above the maximum time step until the problem solution fields reached the convergence criterion at time step number 28 after 1795 s of CPU time.

Figs. 6–59 through 6–61 show $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ for vorticity, streamfunction, and temperature solution fields. The actu$\frac{(\delta Q)_{max}}{(Q_0)_{max}}$ values displayed no oscillating behavior. The convergence of the solution field was very smooth.

Case 22 used the new time step control method based on $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ for vorticity, streamfunction, and temperature first iterates with the same initial and maximum time step sizes as the previous case, as well as a minimum time step size of 0.005. This case reached steady-state convergence in 115 time steps in a total CPU time of 7853 s.

Fig. 6–62 shows the time step size at each time step for Case 22. Note the time step size gradually increased to its maximum value, although the path between the initial and maximum time step was not smooth. Minor fluctuations occurred between time step 20 and time step 85.

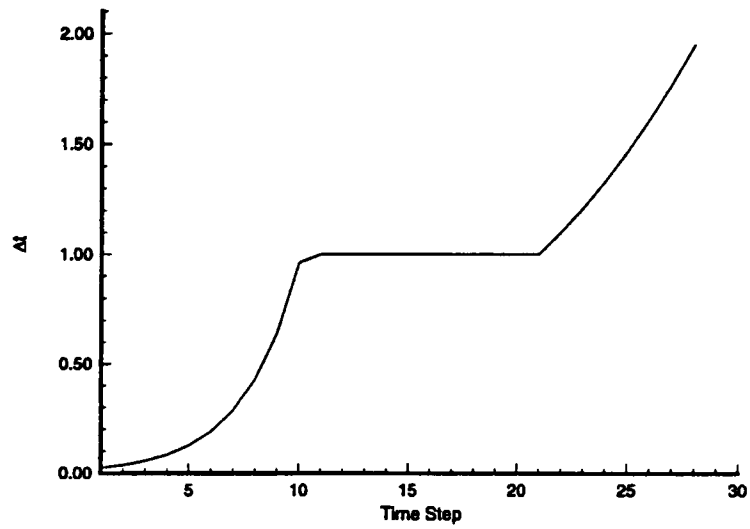


Fig 6–58. Case 21 Time Step Size vs. Time Step

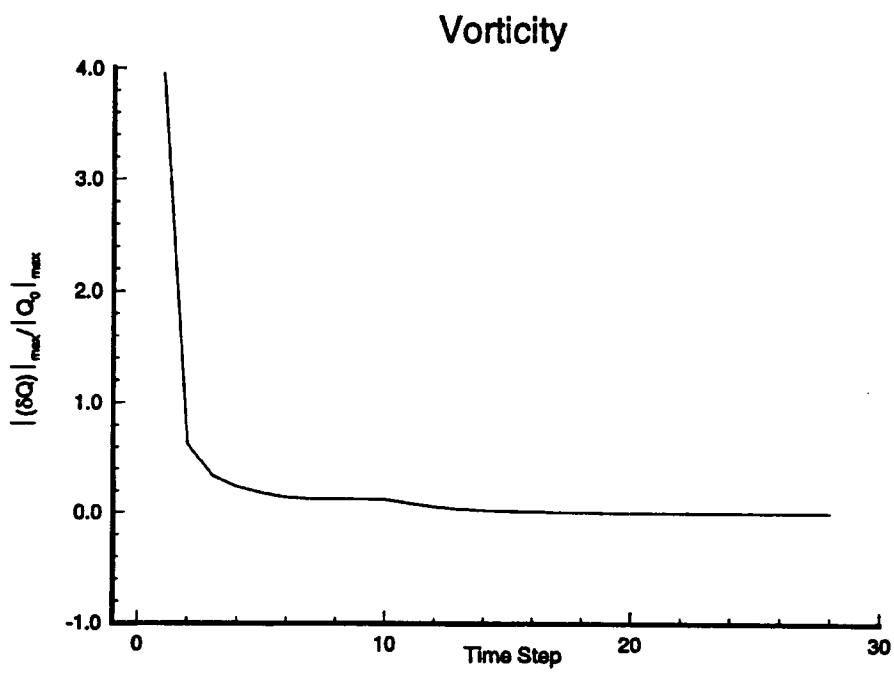


Fig. 6–59. Case 21 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Vorticity

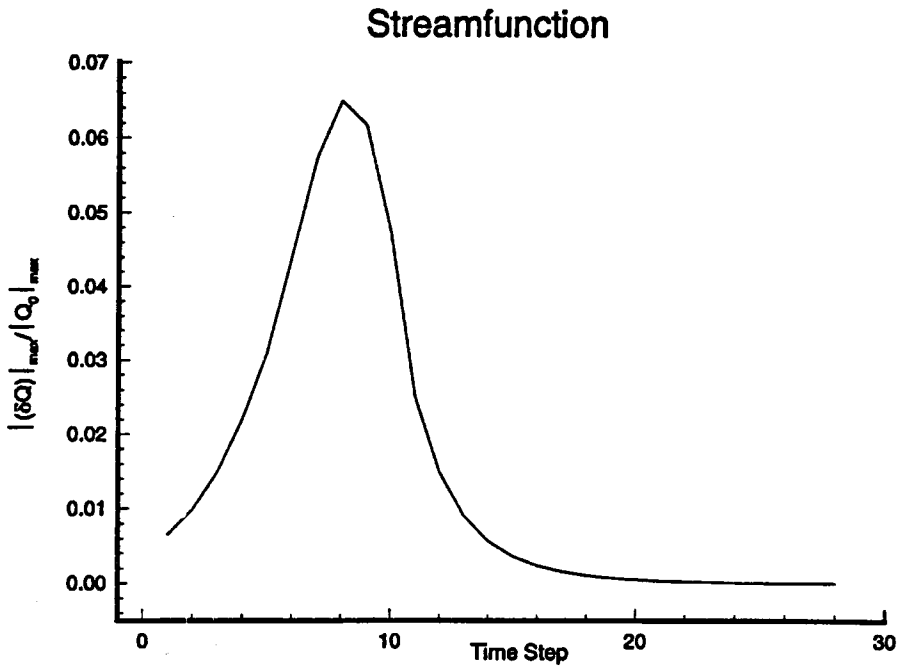


Fig. 6–60. Case 21 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Streamfunction

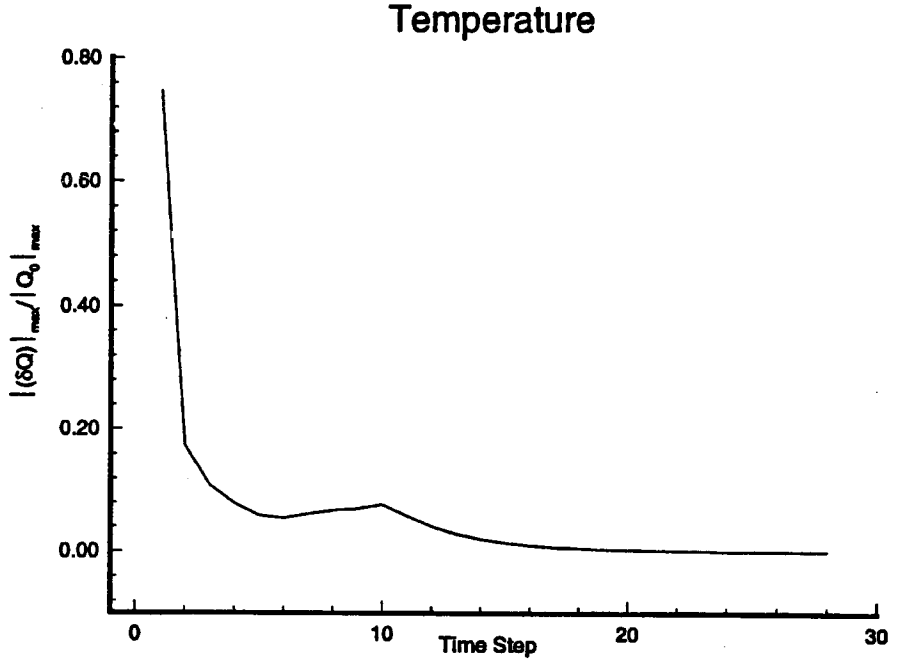


Fig. 6–61. Case 21 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Temperature

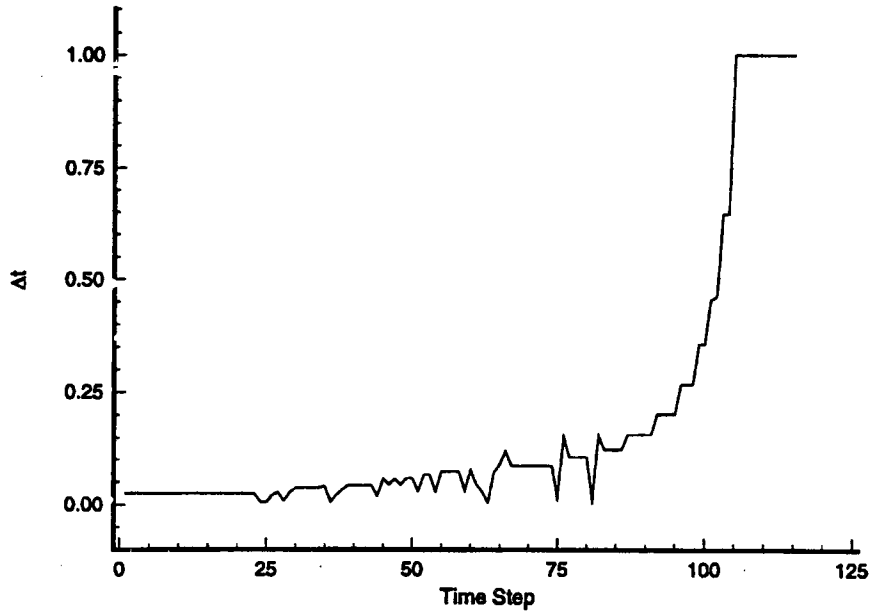


Fig 6–62. Case 22 Time Step Size vs. Time Step

Figs. 6–63 through 6–65 show $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ for vorticity, streamfunction, and temperature first iterates. These data show a general progression towards steady-state convergence, although it is not smooth.

6.1.3.4 Summary, Ra = 1000 Thermal Cavity

The Ra = 1000 thermal cavity cases present results that are very similar to the driven cavity cases. The new time step control method proved to be essential for achieving

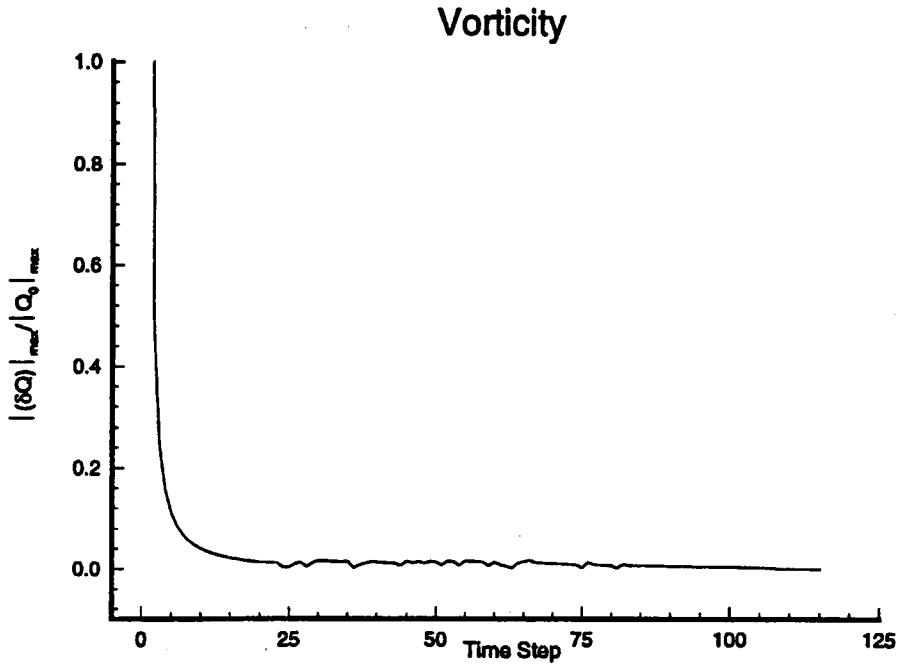


Fig. 6–63. Case 22 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Vorticity

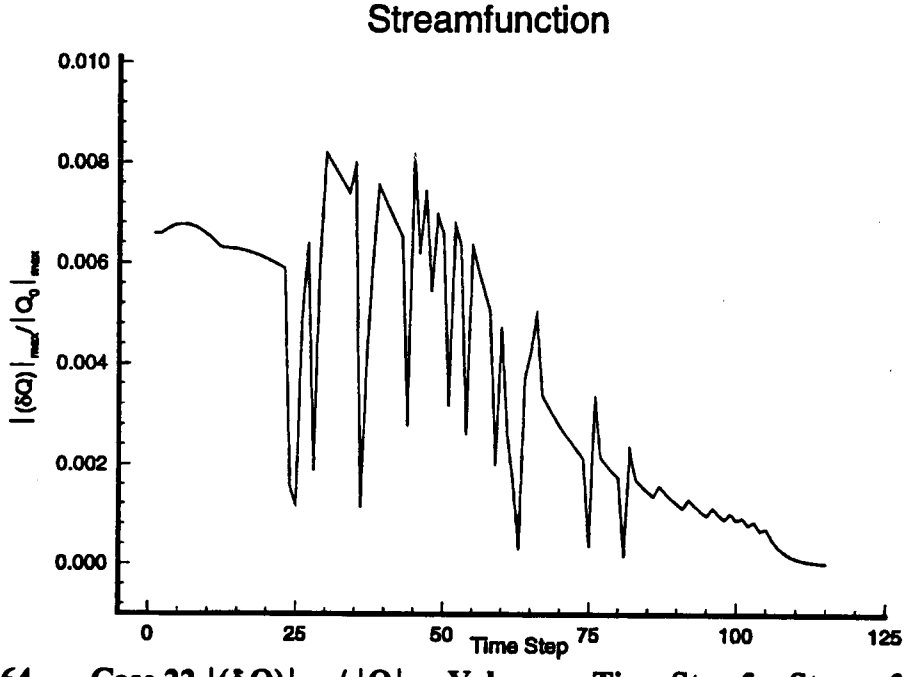


Fig. 6–64. Case 22 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Streamfunction

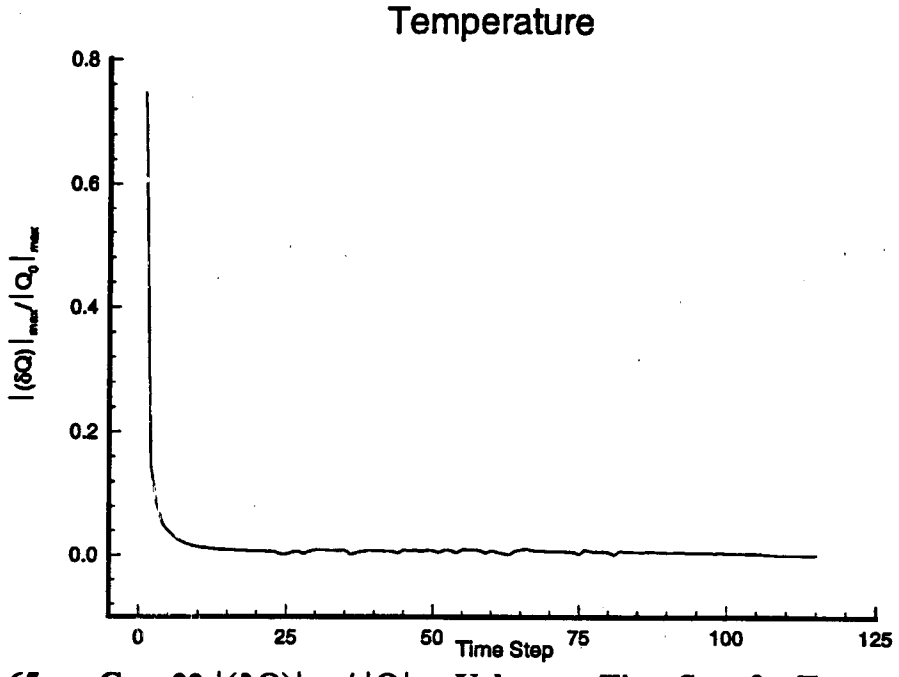


Fig. 6–65. Case 22 $|(\delta Q)|_{max}/|Q|_{max}$ Values vs. Time Step for Temperature

solution convergence using explicit time integration. For $\theta = 0.5$ and $\theta = 1.0$ implicit time integration, both the original and new time step control methods were effective. In some cases, the new time step control method enabled the problem solution to achieve steady-state convergence in fewer time steps, while in other cases the original time step control method enabled the solution to converge quicker. A summary of the Ra = 1000 vorticity-streamfunction thermal cavity cases is provided in Table 6–5.

Table 6–5. Summary of Ra = 1000 Vorticity-Streamfunction Thermal Cavity Cases

Case No.	Method[1]	Initial Time Step Size	Maximum Time Step Size	Minimum Time Step Size	θ	Results Summary
16	Original ($\|(\delta Q)\|_{max}/\|Q\|_{max})^2$	0.0001	0.0005	0.00005	0.0	Diverged
17	New ($\|(\delta Q)\|_{max}/\|Q\|_{max})^3$	0.0001	0.0005	0.00005	0.0	Neither converged nor diverged after 2000 time steps
18	New ($\|(\delta Q)\|_{max}/\|Q\|_{max})^3$	0.00005	0.0004	0.000025	0.0	Neither converged nor diverged after 2000 time steps, but was headed towards convergence
19	Original (1.05)	0.01	0.05	N/A	0.5	Neither converged nor diverged after 2000 time steps, but was headed towards convergence
20	New ($\|(\delta Q)\|_{max}/\|Q\|_{max})^3$	0.01	0.05	0.005	0.5	Converged after 1202 time steps in a total CPU time of 40,517 s
21	Original (1.5)	0.025	1.0	N/A	1.0	Converged after 28 time steps in a total CPU time of 1795 s
22	New ($\|(\delta Q)\|_{max}/\|Q\|_{max})^3$	0.025	1.0	0.005	1.0	Converged after 115 time steps in a total CPU time of 7853 s

Notes:

[1]If the original method is designated, the term in parentheses represents the time step multiplier used in the case. If the new method is designated, the term in the parentheses represents the values used to control the time step method.

[2]Temperature first iterates only was used.

[3]Vorticity, streamfunction, and temperature first iterates were used.

6.2 Continuity Constraint Method Evaluation

The continuity constraint method (Williams, 1993) was used to solve a sample of 2D driven cavity and thermal cavity problems. Data at Re = 100 for the driven cavity problem and at Ra = 1000 for the thermal cavity problem are provided using the $\theta = 0.5$ implicit time integration. For all cases, a nonuniform 16 × 16 element mesh was used. The steady-state convergence criterion was $\frac{|\delta Q|_{max}}{|Q_0|_{max}} \leq 1 \times 10^{-5}$. Unless otherwise stated, all cases involving the new time step control method used the following inputs to determine the values of ϵ_{set}, $\epsilon_{set,high}$, and $\epsilon_{set,low}$: NSTEPK = NSTEPJ - 2, EPSETF = 10.0, EPSETLOF = 0.8, EPSETHIF = 1.25, and NPLACEF = 100000.0.

6.2.1 2D Driven Cavity, Re = 100, $\theta = 0.5$

Several 2D continuity constraint driven cavity cases were run at Re = 100 to evaluate the effectiveness of the time step control method. These cases employ a Newton template.

Case 23 used the original time step control method with an initial time step size of 0.001, a maximum time step size of 0.1, and a time step multiplier of 1.1. This case converged to steady state in 533 time steps in a total CPU time of 17,048 s.

Plots of the time step size and $\frac{(\delta Q)_{max}}{(Q_0)_{max}}$ for the velocity first-iterate fields (U1 and U2) for Case 23 are provided in Figs. 6–66 through 6–68. The time step size increased to the maximum input value in 70 time steps. During this time, the velocity solution fields exhibited large oscillations. Afterwards, the velocity solution fields exhibited smaller oscillations as the solution approached steady-state convergence.

Unlike some of the vorticity-streamfunction cases presented in Sect. 6.1, that oscillated at each successive time step, the continuity constraint problems exhibit oscillations based on a 3-time-step-rotation because, for this case, the pressure field is computed and updated every third time step. This characteristic is shown in Fig. 6–69 which shows $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ for the U1 first iterates between time steps 80 and 100.

Case 24 used the same initial and maximum time step sizes as the previous case, but employed the new time step control method based on $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ for the velocity first-iterate fields (U1 and U2). This case neither converged nor diverged after 1500 time steps.

The time step size history and $\frac{(\delta Q)_{max}}{(Q_0)_{max}}$ for the velocity first-iterates for Case 24 are provided in Figs. 6–70 through 6–72. It is clear from these plots that the problem solution was not getting anywhere. The time step size decreased to its minimum value numerous times as the $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ values reached the $\epsilon_{set,high}$ value.

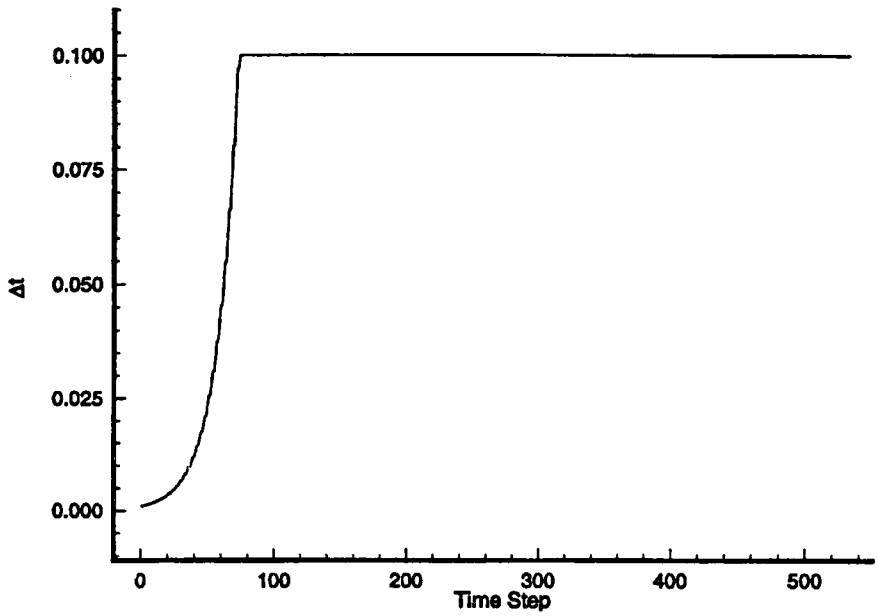


Fig. 6–66. Case 23 Time Step Size vs. Time Step

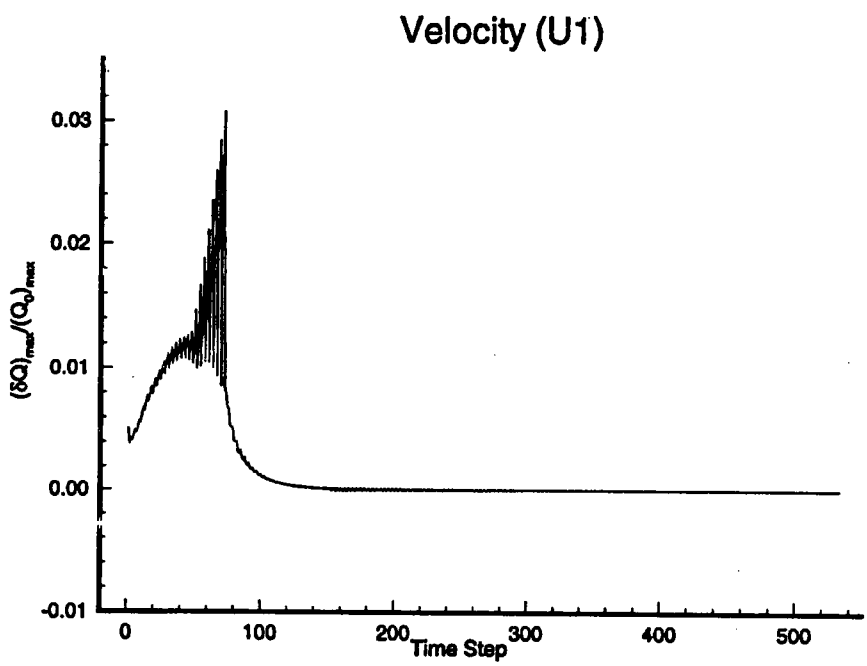


Fig. 6–67. Case 23 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Velocity (U1)

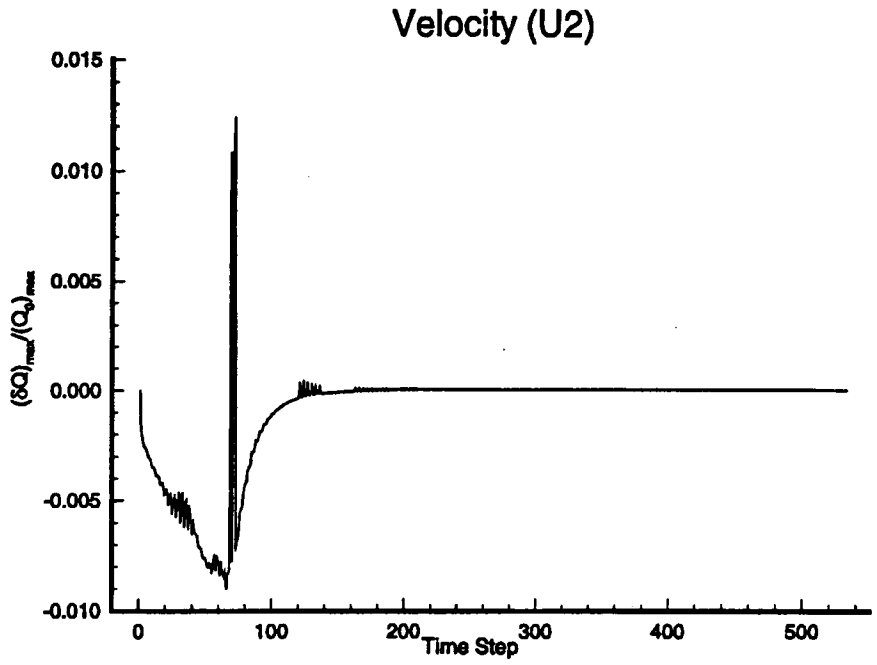


Fig. 6–68. Case 23 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Velocity (U2)

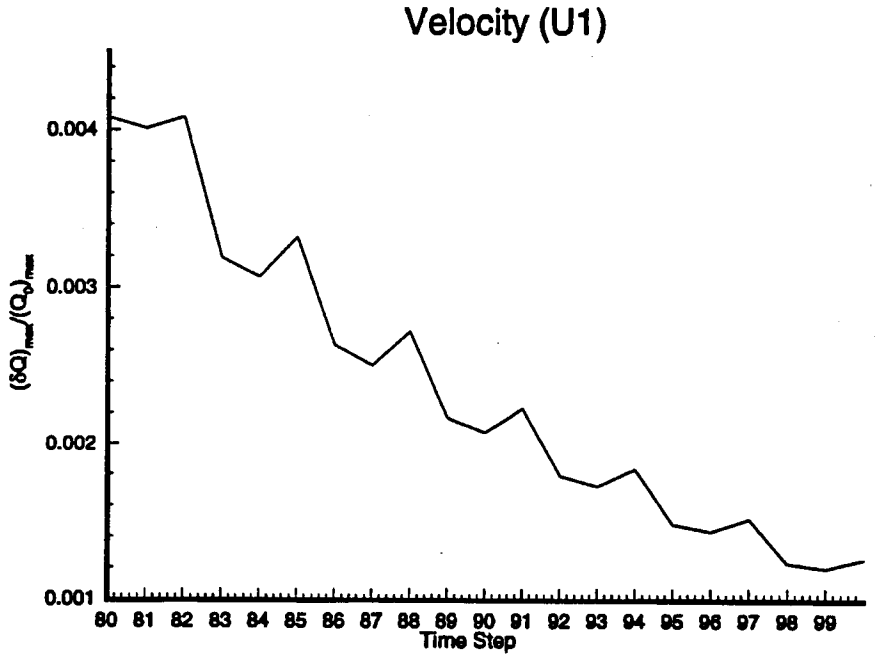


Fig. 6–69. Case 23 $(\delta Q)_{max}/Q_{max}$ Values Between Time Steps 80 and 100 for Velocity (U1)

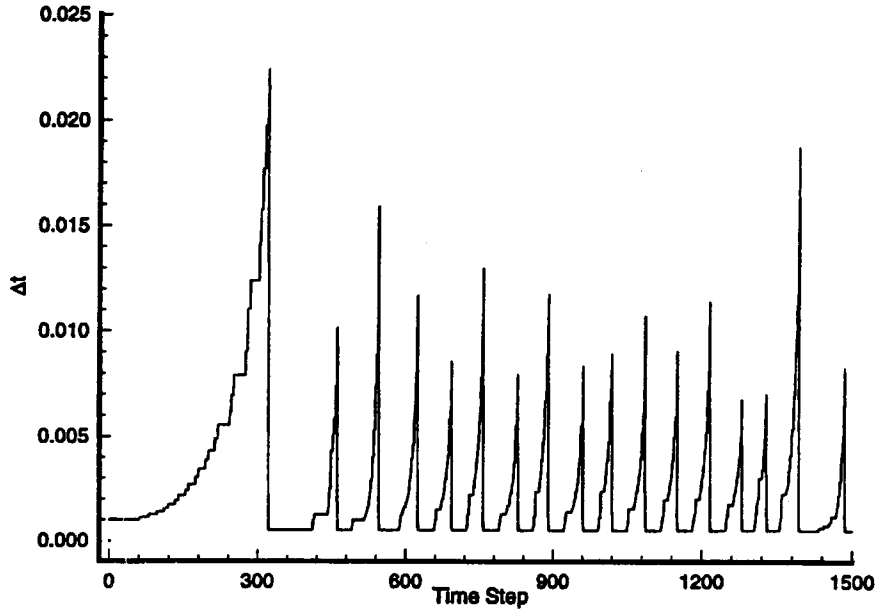


Fig. 6–70. Case 24 Time Step Size vs. Time Step

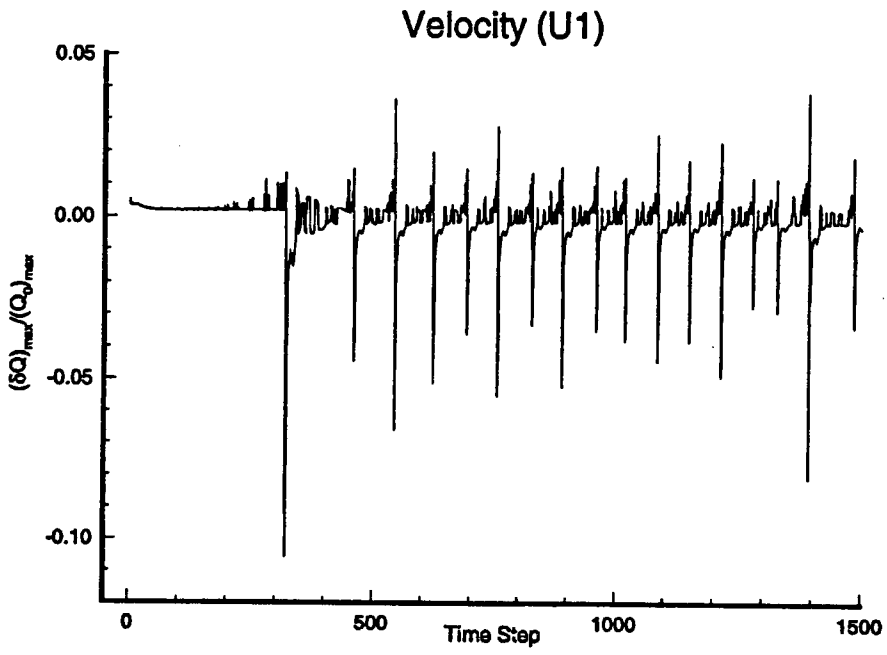


Fig. 6–71. Case 24 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Velocity (U1)

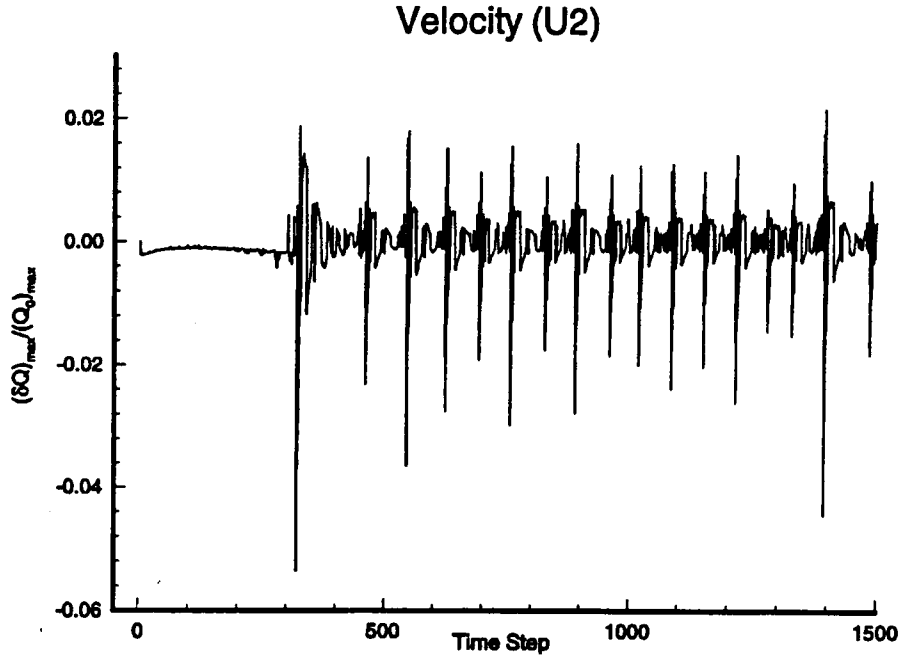


Fig. 6–72. Case 24 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Velocity (U2)

This phenomenon is more easily seen in Figs. 6–73 through 6–75 which show the time step size and $\frac{(\delta Q)_{max}}{(Q_0)_{max}}$ values for the velocity first-iterates between time steps 820 and 900. The U1 field reached the $\epsilon_{set,high}$ value at time step 827. As a result, the time step size immediately dropped to its minimum value, as it was designed to do. The drop in time step size caused the $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ value for the U1 field to drop to a value that was less than the $\epsilon_{set,low}$ value. The time step size gradually increased over the next 50 or so time steps until the $\epsilon_{set,high}$ value was exceeded once more at time step 889. This process continued until the run was manually stopped after 1500 time steps.

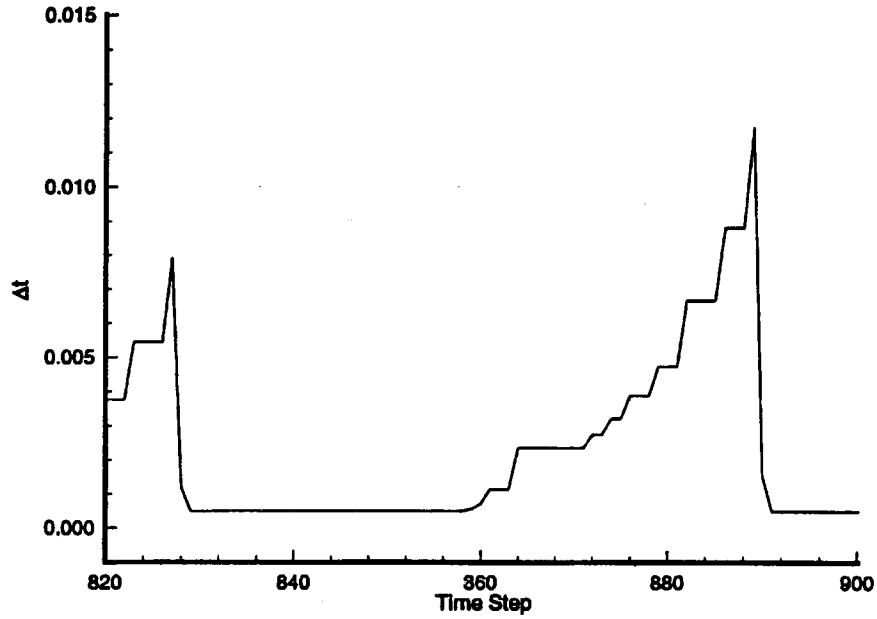


Fig. 6–73. Case 24 Time Step Size Between Time Steps 820 and 900

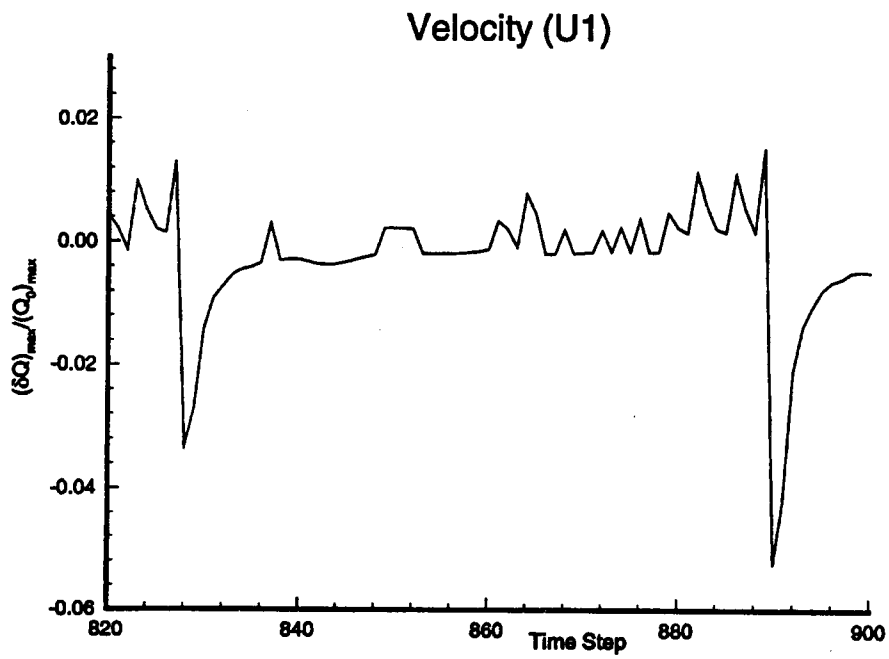


Fig. 6–74. Case 24 $(\delta Q)_{max}/Q_{max}$ Values Between Time Steps 820 and 900 for Velocity (U1)

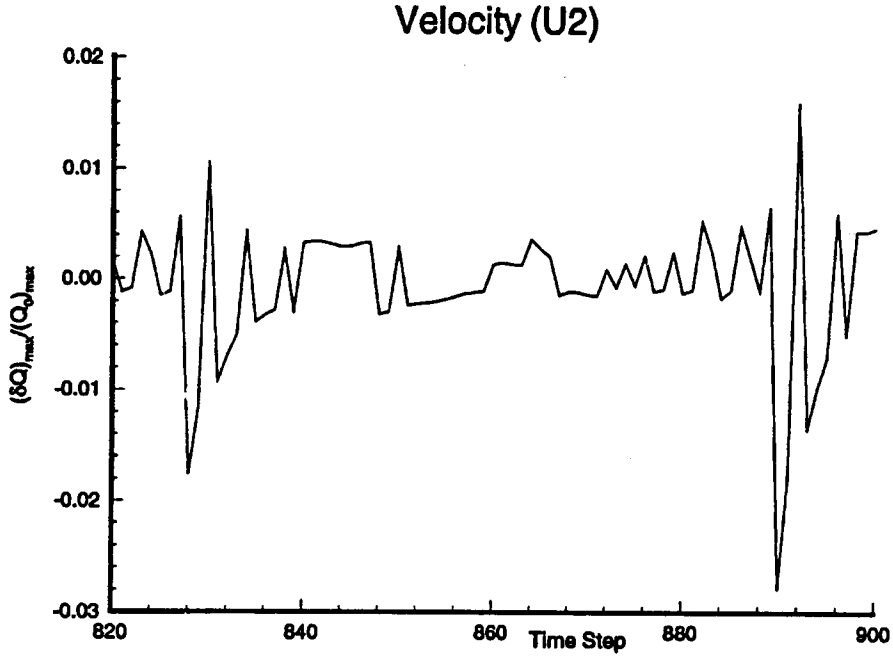


Fig. 6–75. Case 24 $(\delta Q)_{max}/Q_{max}$ Values Between Time Steps 820 and 900 for Velocity (U2)

The problems displayed in the Case 24 results are not the result of a problem with the new time step control method, but are instead caused by the input values used for $\epsilon_{set,high}$ and $\epsilon_{set,low}$. Another case (Case 25) was run with the same time step size boundaries as the previous case, but with new boundaries on the ϵ_{set} values based on the following input parameters: EPSETHIF = 100.0, EPSETLOF = 0.9, and EPSETF = 10000.0 (compared with EPSETHIF = 10.0, EPSETLOF = 0.8, and EPSETF = 500.0 which were used in Case 24). By increasing the upper bound on ϵ_{set} using these new values, larger oscillations in the $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ values can occur without significant impact on the time step size.

This case converged after 668 time steps in a total CPU time of 21,041 s. The time step size and $\frac{(\delta Q)_{max}}{(Q_0)_{max}}$ for the velocity first iterates for Case 25 are graphed in Figs. 6–76 through 6–78. These plots demonstrate a very methodical increase in the time step size to its maximum value and demonstrate converging behavior in the velocity solution fields that are similar to those in Case 23.

A summary of the Re = 100 continuity constraint driven cavity problems is provided in Table 6–6.

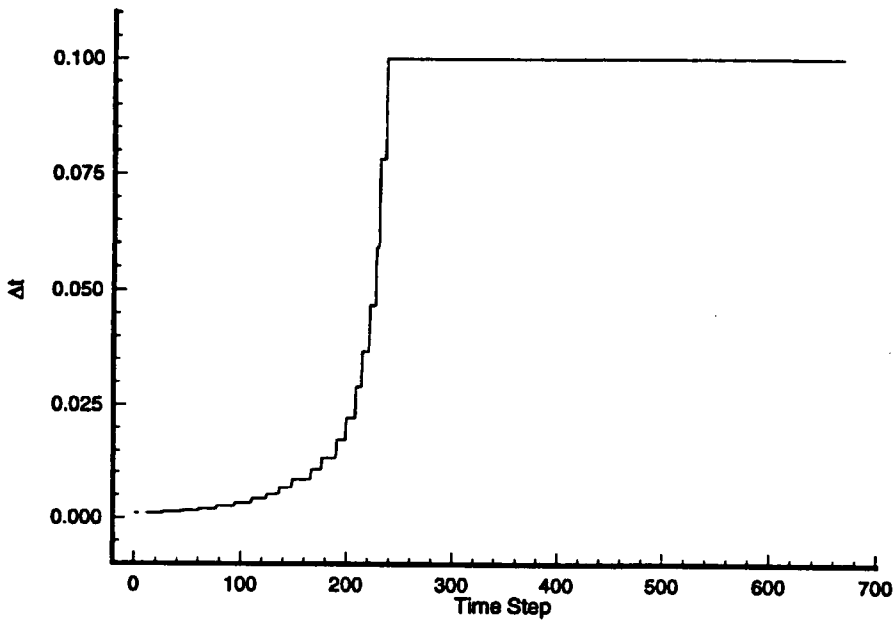


Fig. 6–76. Case 25 Time Step Size vs. Time Step

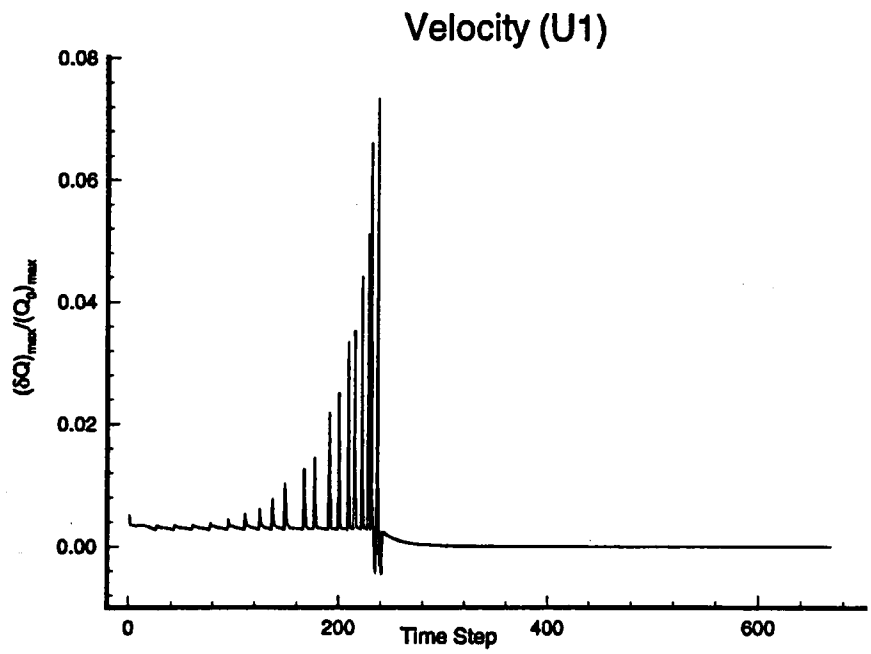


Fig. 6–77. Case 25 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Velocity (U1)

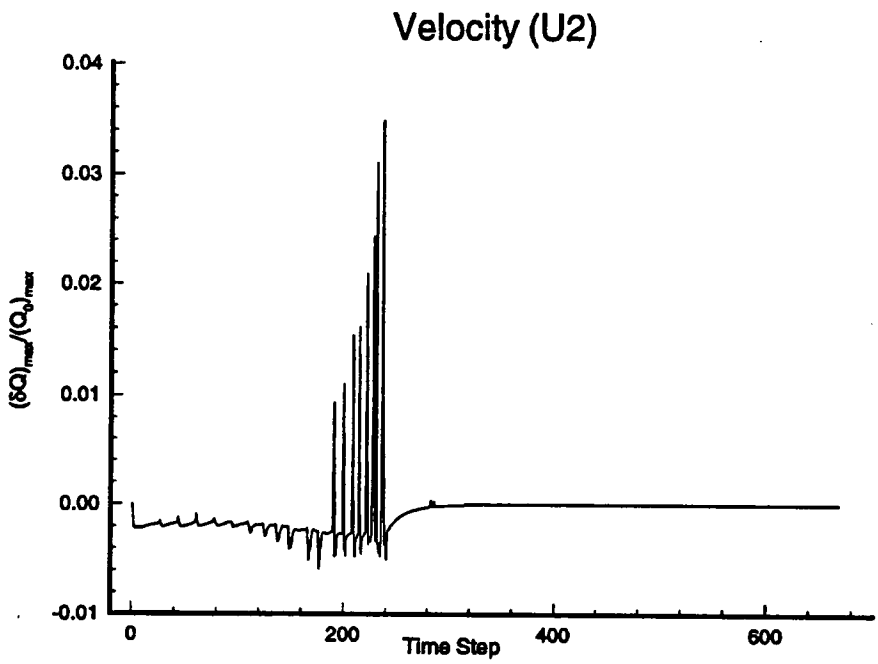


Fig. 6–78. Case 25 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Velocity (U2)

Table 6–6. Summary of Re = 100 Continuity Constraint Driven Cavity Cases

Case No.	Method[1]	Initial Time Step Size	Maximum Time Step Size	Minimum Time Step Size	θ	Results Summary
23	Original (1.1)	0.001	0.1	N/A	0.5	Converged after 533 time steps in a total CPU time of 17,048 s
24	New $(\lvert(\delta Q)\rvert_{max}/\lvert Q\rvert_{max})^{2,3}$	0.001	0.1	0.0005	0.5	Neither converged nor diverged after 1500 time steps
25	New $(\lvert(\delta Q)\rvert_{max}/\lvert Q\rvert_{max})^{2,4}$	0.001	0.1	0.0005	0.5	Converged after 668 time steps in a total CPU time of 21,041 s

Notes:

[1]If the original method is designated, the term in parentheses represents the time step multiplier used in the case. If the new method is designated, the term in the parentheses represents the values used to control the time step method.

[2]Both velocity solution fields (U1 and U2) were used.

[3]Input Values of EPSETHIF = 10.0, EPSETLOF = 0.8, and EPSETF = 500.0 were used.

[4]Input values of EPSETHIF = 100.0, EPSETLOF = 0.9, and EPSETF = 10000.0 were used.

6.2.2 2D Thermal Cavity, Ra = 1000

Two 2D continuity constraint thermal cavity cases were run at θ = 0.5 and a Rayleigh number of 1000 to evaluate the effectiveness of the time step control method. These

cases used a Newton template in which the energy equation solution is coupled with the velocity solutions.

Case 26 used the original time step control method with an initial time step size of 0.01, a maximum time step size of 1.0, and a time step multiplier of 1.1. This case converged in 216 time steps with a total CPU time of 16,208 s.

The time step size history and $\frac{(\delta Q)_{max}}{(Q_0)_{max}}$ values for the velocity (U1 and U2) and temperature first-iterate fields for Case 26 are plotted in Figs. 6–79 through 6–82. These plots demonstrate a very methodical increase in the time step size to its maximum value and demonstrate reasonably controlled oscillating behavior in the solution fields that enable the problem to reach the steady-state convergence.

Case 27 used the new time step control method based on $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ for the velocity (U1 and U2) and temperature first-iterate fields with the same initial and maximum time step sizes as the previous case, as well as a minimum time step size of 0.005. This case reached convergence in 417 time steps with a total CPU time of 26,515 s.

Plots of Δt and $\frac{(\delta Q)_{max}}{(Q_0)_{max}}$ for the velocity (U1 and U2) and temperature first iterates for Case 27 are shown in Figs. 6–83 through 6–86. These data demonstrate a very methodical increase in the time step size to its maximum value and demonstrate

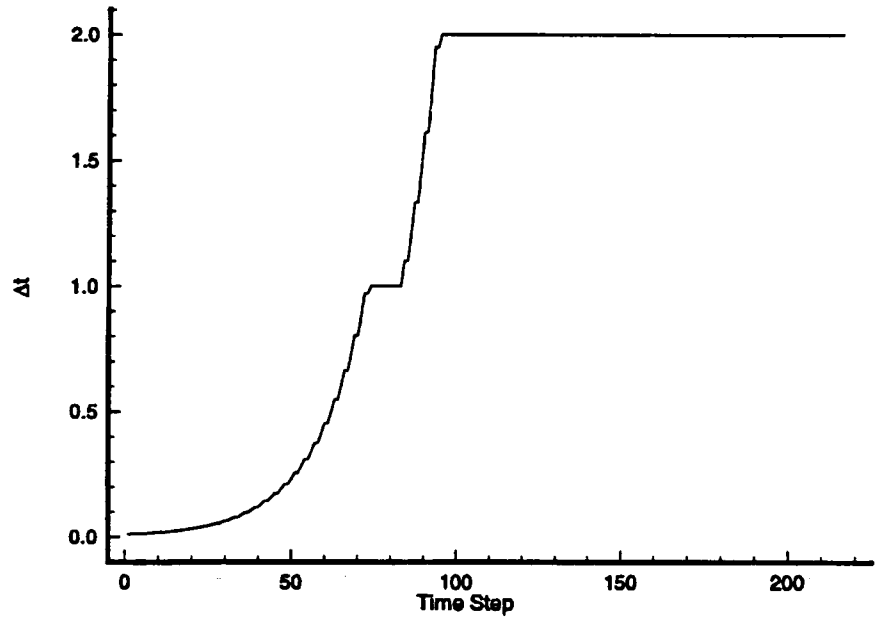


Fig 6–79. Case 26 Time Step Size vs. Time Step

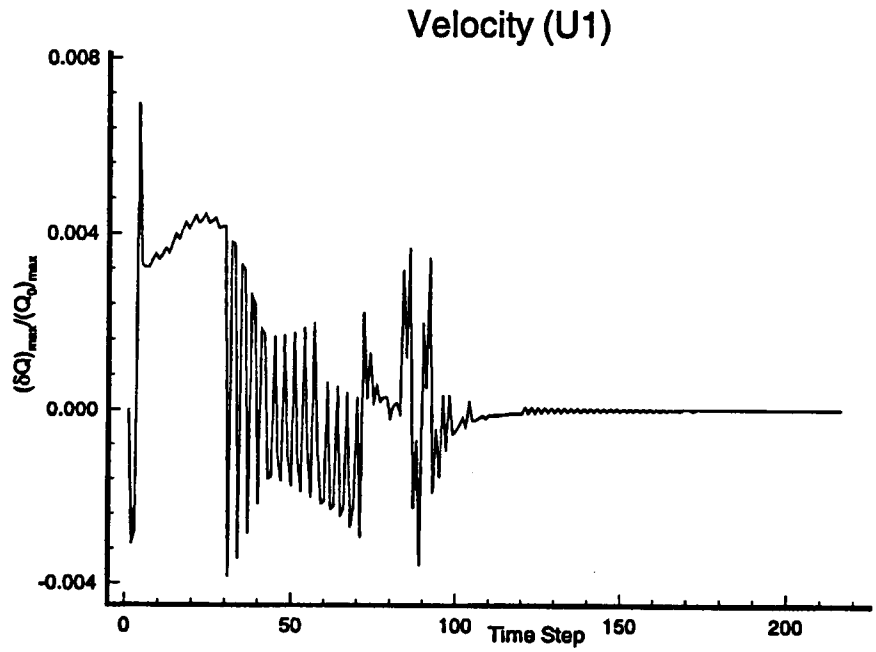


Fig. 6–80. Case 26 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Velocity (U1)

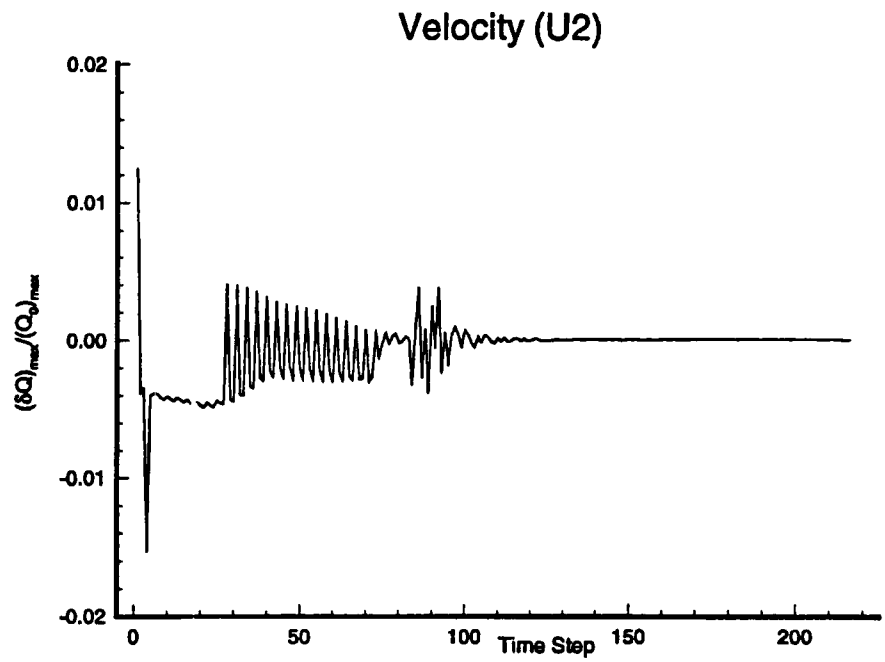


Fig. 6–81. Case 26 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Velocity (U2)

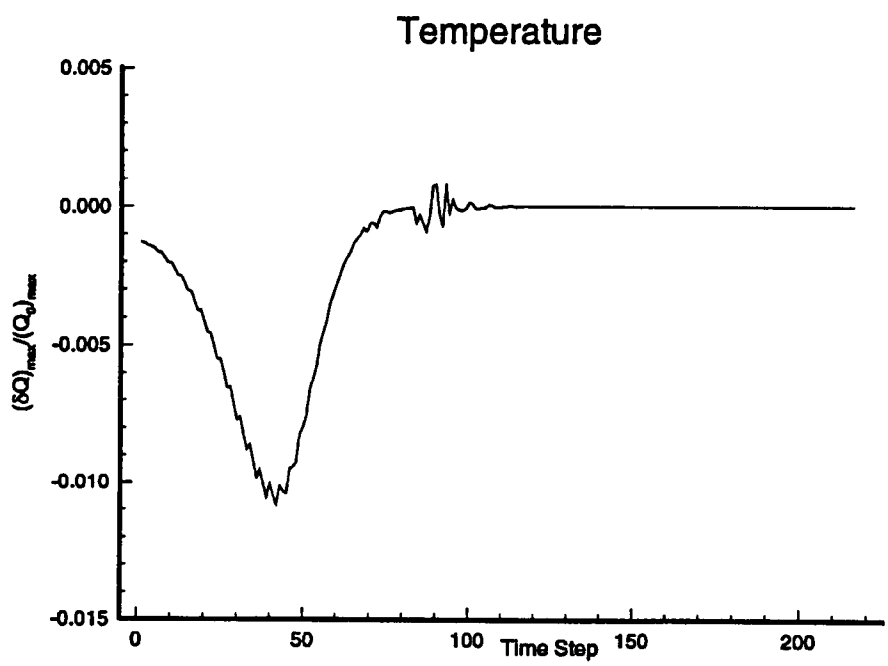


Fig. 6–82. Case 26 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Temperature

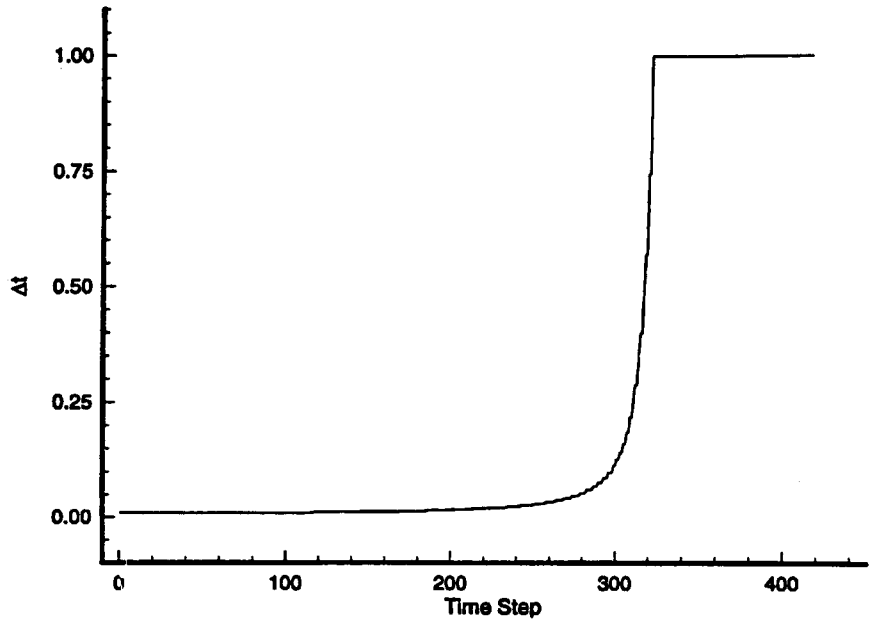


Fig. 6–83. Case 27 Time Step Size vs. Time Step

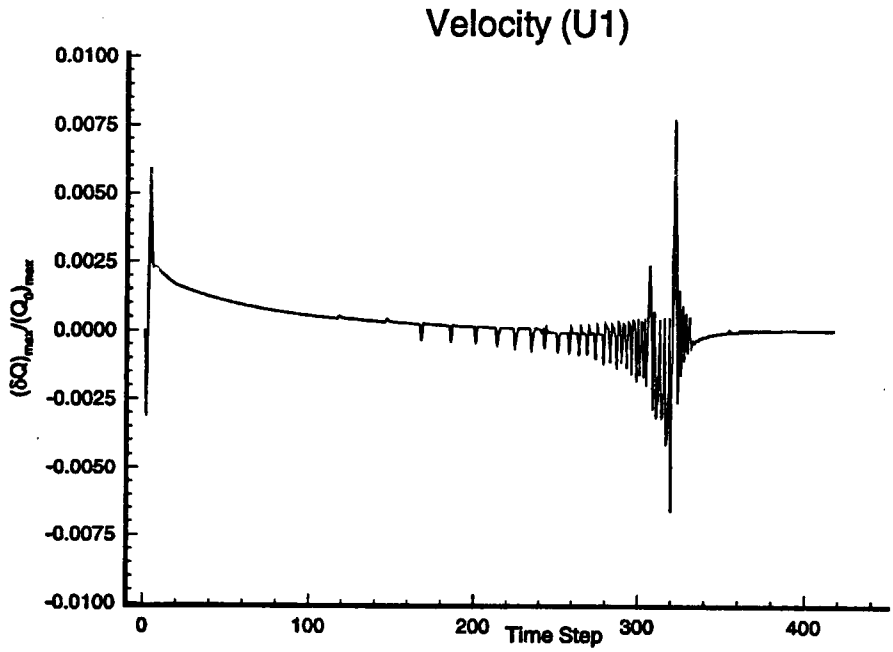


Fig. 6–84. Case 27 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Velocity (U1)

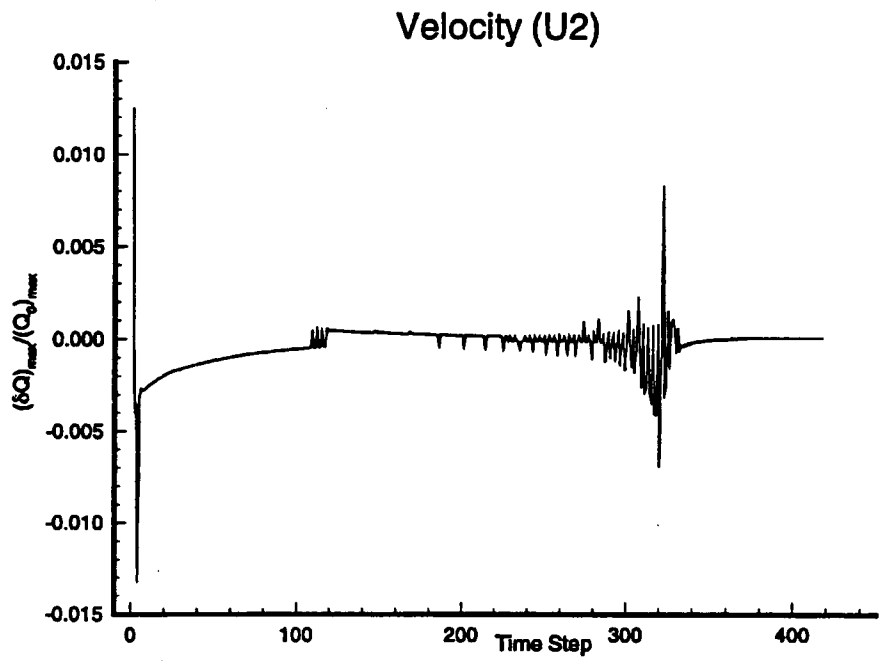


Fig. 6–85. Case 27 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Velocity (U2)

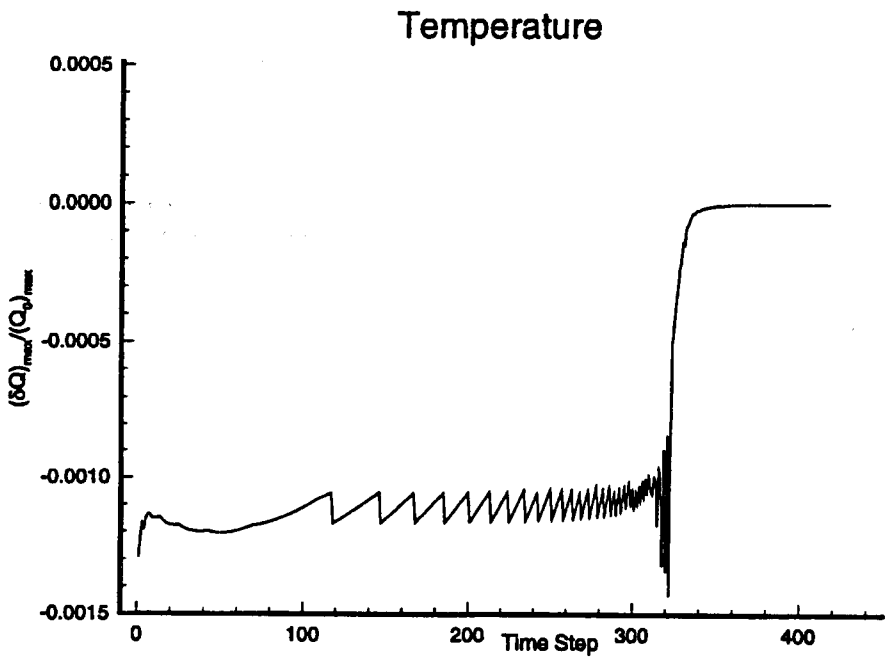


Fig. 6–86. Case 27 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Temperature

controlled converging behavior in the velocity solution fields that is similar to that in the previous case.

A summary of the Ra = 1000 continuity constraint thermal cavity problems is provided in Table 6–7.

Table 6–7. Summary of Ra = 1000 Continuity Constraint Thermal Cavity Cases

Case No.	Method[1]	Initial Time Step Size	Maximum Time Step Size	Minimum Time Step Size	θ	Results Summary
26	Original (1.1)	0.01	1.0	N/A	0.5	Converged after 216 time steps in a total CPU time of 16,208 s.
27	New[2] ($\|(\delta Q)\|_{max} / \|Q\|_{max})^{2,3}$	0.01	1.0	0.005	0.5	Converged after 417 time steps in a total CPU time of 26,515 s.

Notes:

[1]If the original method is designated, the term in parentheses represents the time step multiplier used in the case. If the new method is designated, the term in the parentheses represents the values used to control the time step method.

[2]The velocity (U1 and U2) and temperature solution fields were used.

[3]Input values of EPSETHIF = 100.0, EPSETLOF = 0.9, and EPSETF = 10000.0 were used.

6.3 Penalty Method Results

The penalty method (Zienkiewicz, 1974) algorithm was used to solve a sample of 2D driven cavity and thermal cavity problems at Re = 100 and Ra = 1000, respectively, using $\theta = 0.5$ and $\theta = 1.0$ time integration. For all cases, a 16 × 16 uniform element mesh was used. The steady-state convergence criterion was $\frac{|\delta Q|_{max}}{|Q_0|_{max}} \leq 1 \times 10^{-5}$. Unless otherwise stated, all cases involving the new time step control method used the following factors to determine the values of ϵ_{set}, $\epsilon_{set,high}$, and $\epsilon_{set,low}$: NSTEPK = NSTEPJ - 2, EPSETF = 10.0, EPSETLOF = 0.8, EPSETHIF = 1.25, and NPLACEF = 100000.0.

6.3.1 2D Driven Cavity, Re = 100

Several 2D penalty driven cavity cases were run at Re = 100 to evaluate the effectiveness of the time step control method. Cases at $\theta = 0.5$ and $\theta = 1.0$ are presented in the following sections.

6.3.1.1 $\theta = 0.5$ Cases

Two 2D penalty driven cavity cases were run using $\theta = 0.5$ at Re = 100. Case 28 used the original time step control method with an initial time step size of 0.1, a maximum time step size of 0.5, and a time step multiplier of 1.1. This case converged in 295 time steps with a total CPU time of 4649 s.

Plots of the time step size and the $\frac{(\delta Q)_{max}}{(Q_0)_{max}}$ values for the velocity first-iterate fields (U1 and U2) for Case 28 are provided in Figs. 6–87 through 6–89. These plots exhibit oscillating convergence that effectively reaches the steady-state solution.

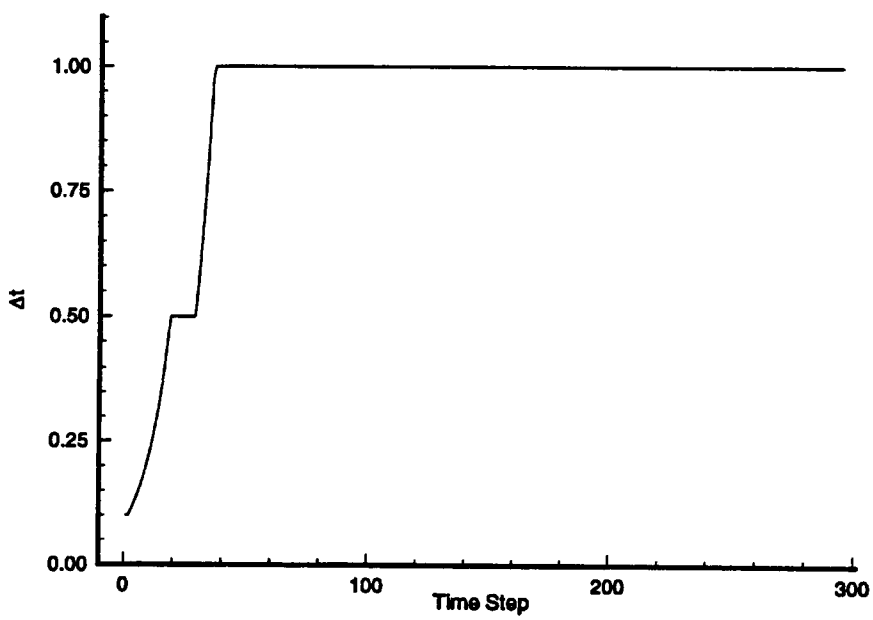


Fig. 6–87. Case 28 Time Step Size vs. Time Step

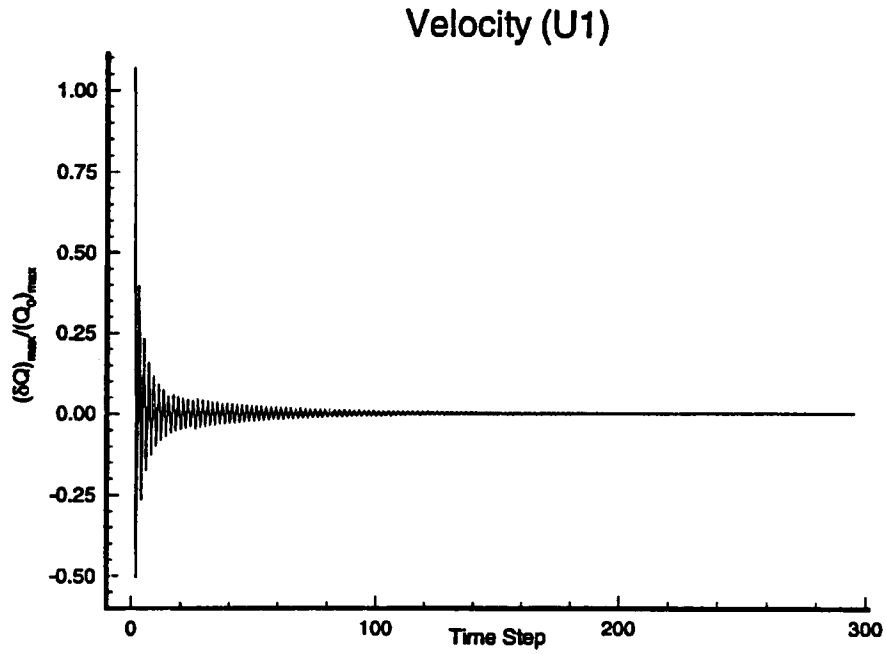


Fig. 6–88. Case 28 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Velocity (U1)

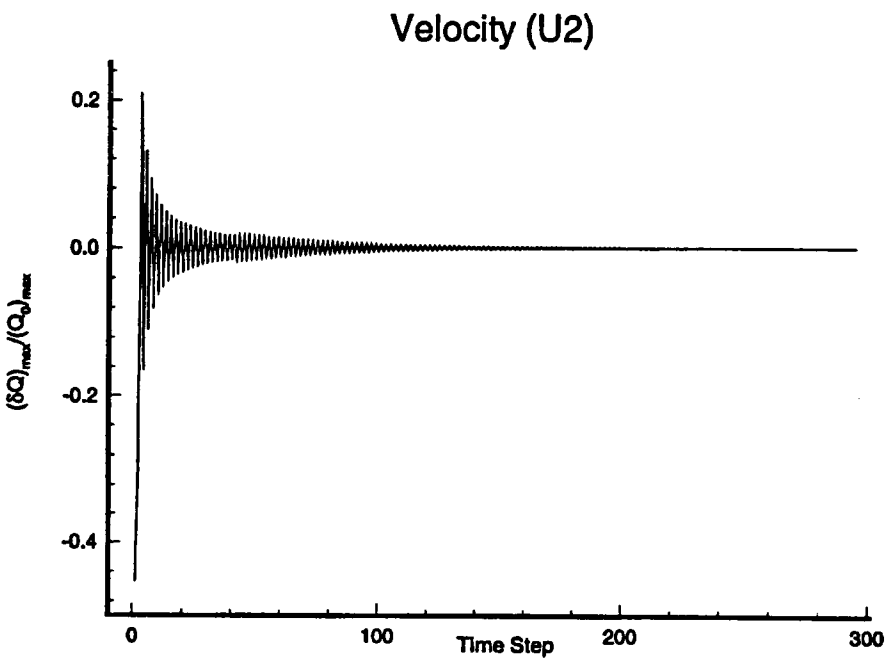


Fig. 6–89. Case 28 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Velocity (U2)

The next case (Case 29) used the same initial and maximum time step sizes as the previous case, but employed the new time step control method based on the values of $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ in the velocity first-iterate fields (U1 and U2). This case converged in 147 time steps in a total CPU time of 2112 s.

Time step size history and $\frac{(\delta Q)_{max}}{(Q_0)_{max}}$ for the velocity first iterates (U1 and U2) for Case 29 are plotted in Figs. 6–90 through 6–92. These plots also demonstrate oscillating convergence similar to the previous case.

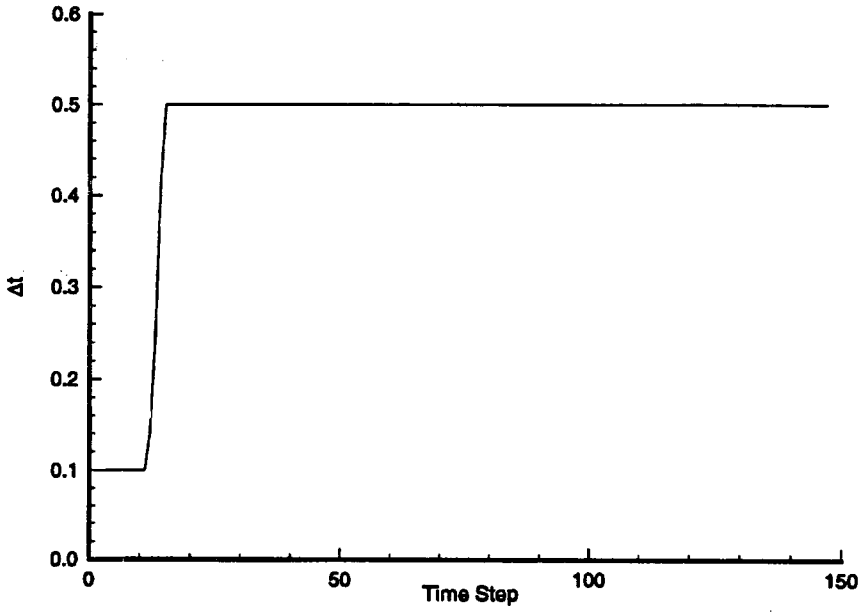


Fig. 6–90. Case 29 Time Step Size vs. Time Step

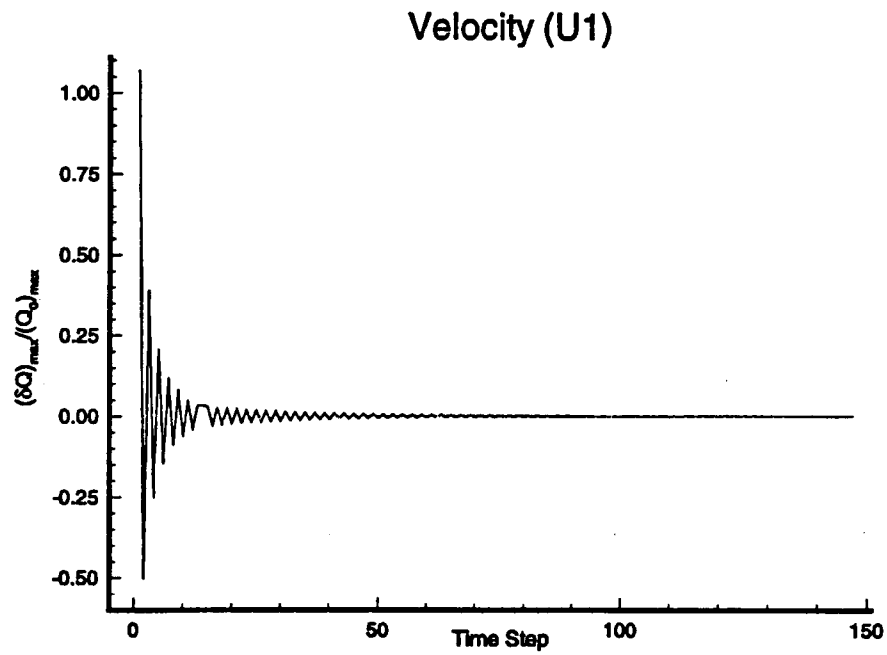


Fig. 6–91. Case 29 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Velocity (U1)

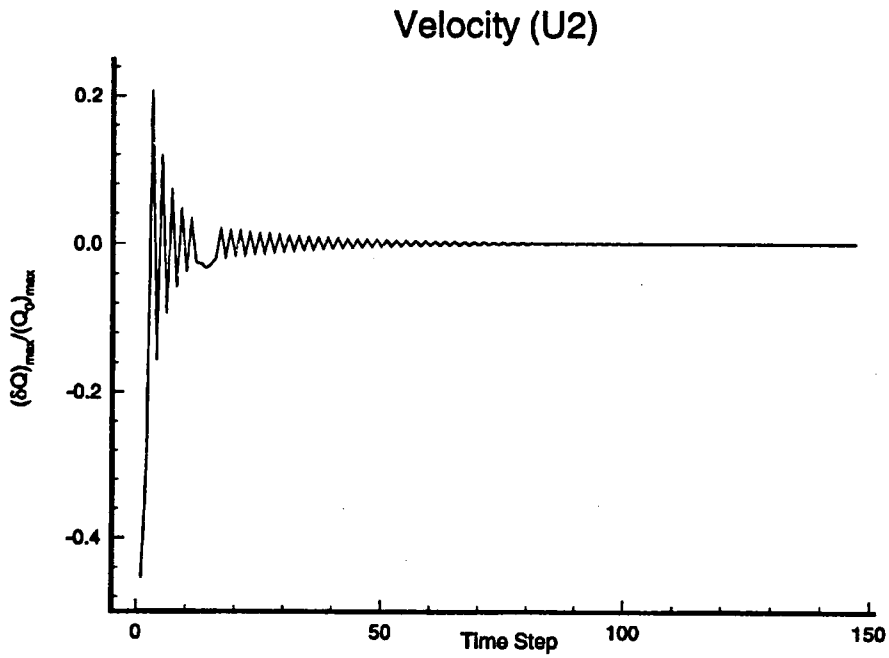


Fig. 6–92. Case 29 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Velocity (U2)

For these two cases, the new time step control method achieved steady-state convergence in the solution fields in fewer time steps than occurred in the case using the original method. The number of time steps required to achieve convergence differed in the two methods because of the difference in the maximum time step size that was reached. The new method maintained the time step size within the specified input minimum and maximum limits, whereas the original method allowed the time step size to exceed the maximum limit. The larger time step size in Case 28 actually caused the problem to converge using a greater number of time steps. These cases demonstrate that a larger time step does not always provide the most efficient solving of a problem.

6.3.1.2 $\theta = 1.0$ Cases

Two 2D penalty driven cavity cases were run using $\theta = 1.0$ at Re = 100. Case 30 used the original time step control method with an initial time step size of 2.0, a maximum time step size of 10.0, and a time step multiplier of 1.1. This case converged in 10 time steps with a total CPU time of 321 s.

The time step size and $\frac{(\delta Q)_{max}}{(Q_0)_{max}}$ values for the velocity first-iterate fields (U1 and U2) for Case 30 are graphed in Figs. 6–93 through 6–95. These plots indicate a very smooth and robust progression towards steady-state convergence.

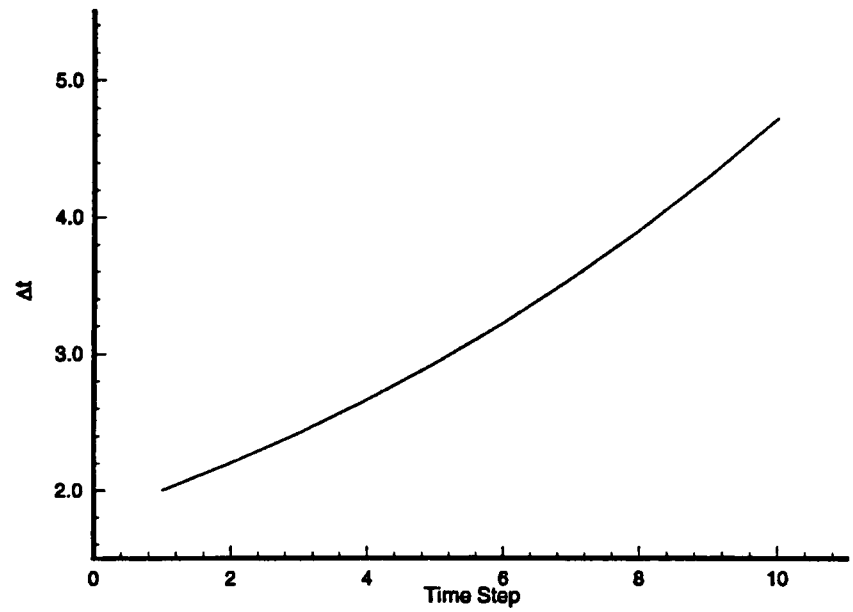


Fig. 6–93. Case 30 Time Step Size vs. Time Step

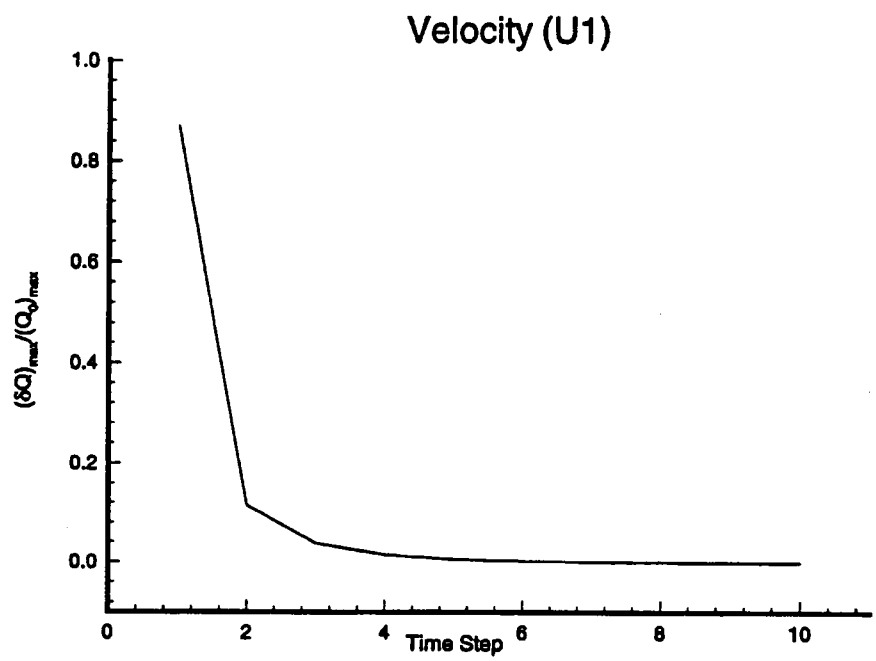


Fig. 6–94. Case 30 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Velocity (U1)

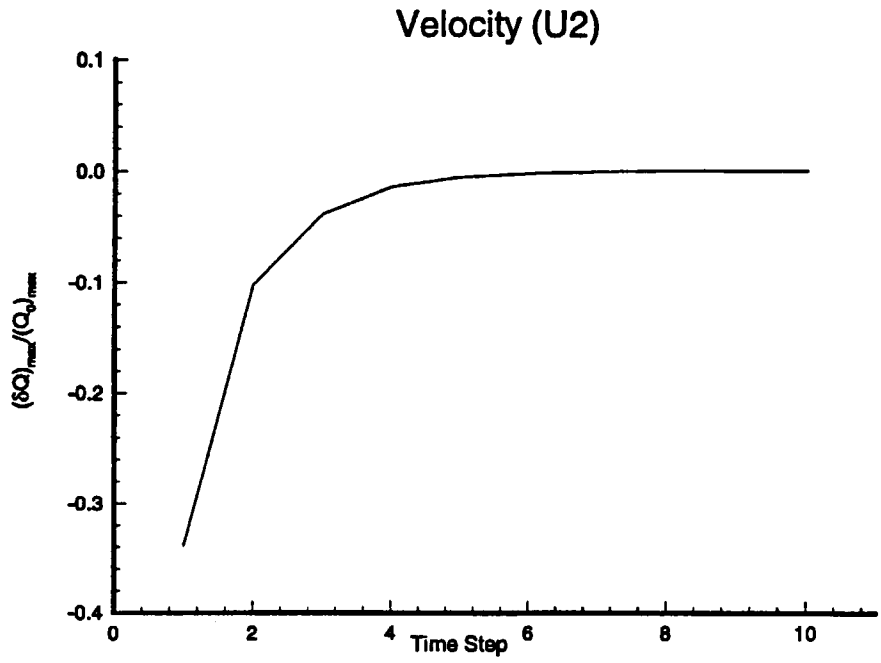


Fig. 6–95. Case 30 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Velocity (U2)

The next case (Case 31) used the new time step control method based on $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ for the velocity (U1 and U2) first iterates with the same initial and maximum time step sizes as the previous case. This case converged in 8 time steps in a total CPU time of 268 s.

Time step history and $\frac{(\delta Q)_{max}}{(Q_0)_{max}}$ for the velocity (U1 and U2) first-iterate fields for Case 31 are plotted in Figs. 6–96 through 6–98. These plots also demonstrate a very smooth and robust progression towards steady-state convergence.

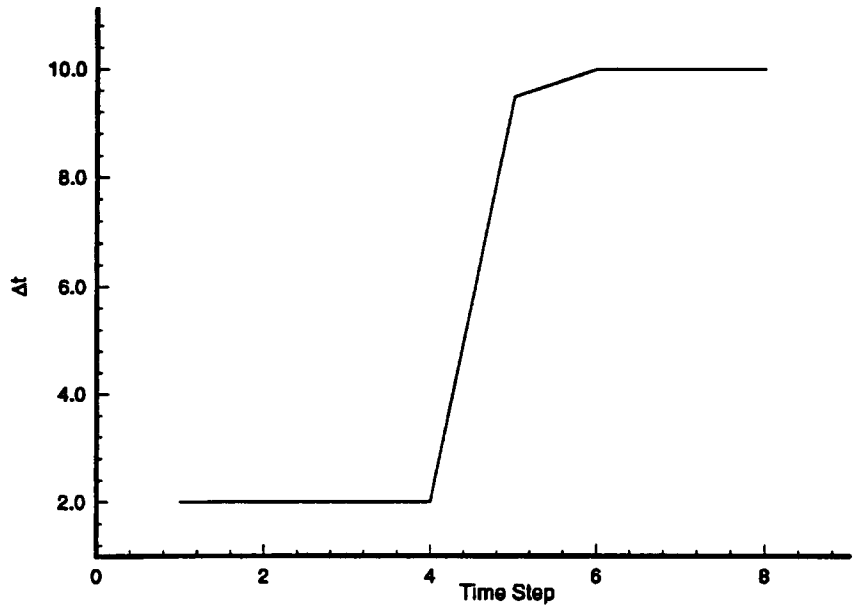


Fig. 6–96. Case 31 Time Step Size vs. Time Step

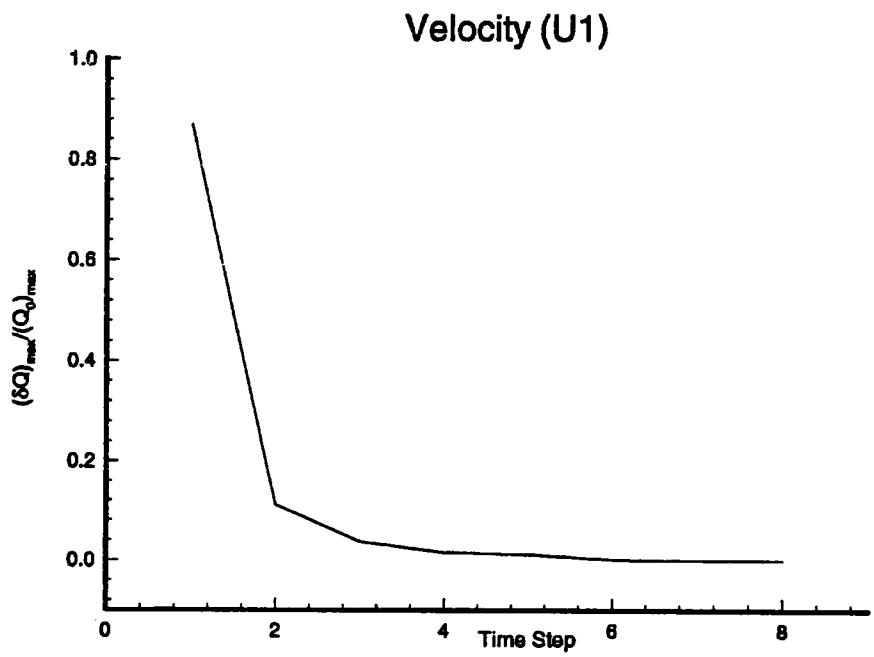


Fig. 6–97. Case 31 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Velocity (U1)

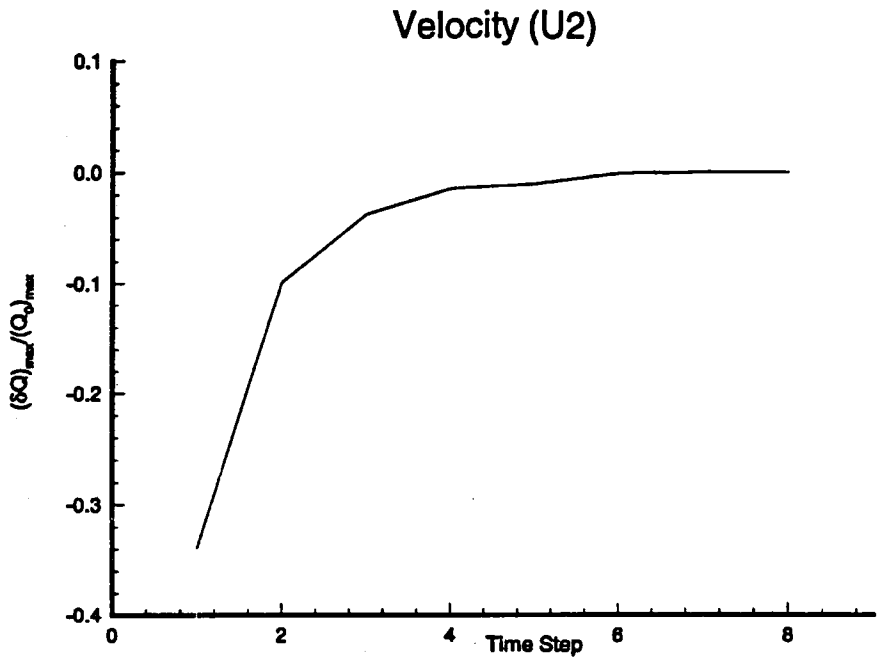


Fig. 6–98. Case 31 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Velocity (U2)

6.3.1.3 Summary, Re = 100 Driven Cavity

The results from the Re = 100 penalty algorithm driven cavity cases evaluate the effectiveness of both time step control methods. Both methods allowed the problem to achieve convergence in a "reasonable" number of time steps. The penalty method is so effective for the driven cavity problems on a uniform Cartesian mesh that perhaps any time step control method would be effective! A summary of the Re = 100 penalty driven cavity cases is provided in Table 6–8.

Table 6–8. Summary of Re = 100 Penalty Driven Cavity Cases

Case No.	Method[1]	Initial Time Step Size	Maximum Time Step Size	Minimum Time Step Size	θ	Results Summary
28	Original (1.1)	0.1	0.5	N/A	0.5	Converged after 295 time steps in a total CPU time of 4649 s
29	New $(\|(\delta Q)\|_{max} / \|Q\|_{max})^2$	0.1	0.5	0.05	0.5	Converged after 147 time steps in a total CPU time of 2112 s
30	Original (1.1)	2.0	10.0	1.0	1.0	Converged after 10 time steps in a total CPU time of 321 s
31	New $(\|(\delta Q)\|_{max} / \|Q\|_{max})^2$	2.0	10.0	1.0	1.0	Converged after 8 time steps in a total CPU time of 268 s

Notes:
[1]If the original method is designated, the term in parentheses represents the time step multiplier used in the case. If the new method is designated, the term in the parentheses represents the values used to control the time step method.

[2]The velocity (U1 and U2) first-iterate fields were used.

6.3.2 2D Thermal Cavity, Ra = 1000

Several 2D thermal cavity cases were run at a Rayleigh (Ra) number of 1000 to evaluate the effectiveness of the time step control method. Whereas the driven cavity cases used a Newton template, in which all variables were directly coupled, the thermal cavity cases presented here employ a quasi-Newton template in which the energy equation solution is

separate from the coupled velocity solutions. Cases at $\theta = 0.5$ and $\theta = 1.0$ are provided in the following sections.

6.3.2.1 $\theta = 0.5$ Cases

Two 2D penalty thermal cavity cases were run using the $\theta = 0.5$ implicit time-integration algorithm at Ra = 1000. Case 32 used the original time step control method with an initial time step size of 0.1, a maximum time step size of 0.5, and a time step multiplier of 1.1. This case converged in 939 time steps in a total CPU time of 25,140 s.

Plots of the time step size and $\frac{(\delta Q)_{max}}{(Q_0)_{max}}$ values for the velocity (U1 and U2) and temperature first-iterate fields at each time step for Case 32 are provided in Figs. 6–99 through 6–102. These plots demonstrate an oscillating, but deliberate, progression towards steady-state convergence.

Case 33 used the new time step controller based on $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ for the velocity (U1 and U2) and temperature first iterates with the same initial and maximum time step sizes as the previous case. A minimum time step size of 0.05 was selected. This case reached convergence in 397 time steps with a total CPU time of 8199 s.

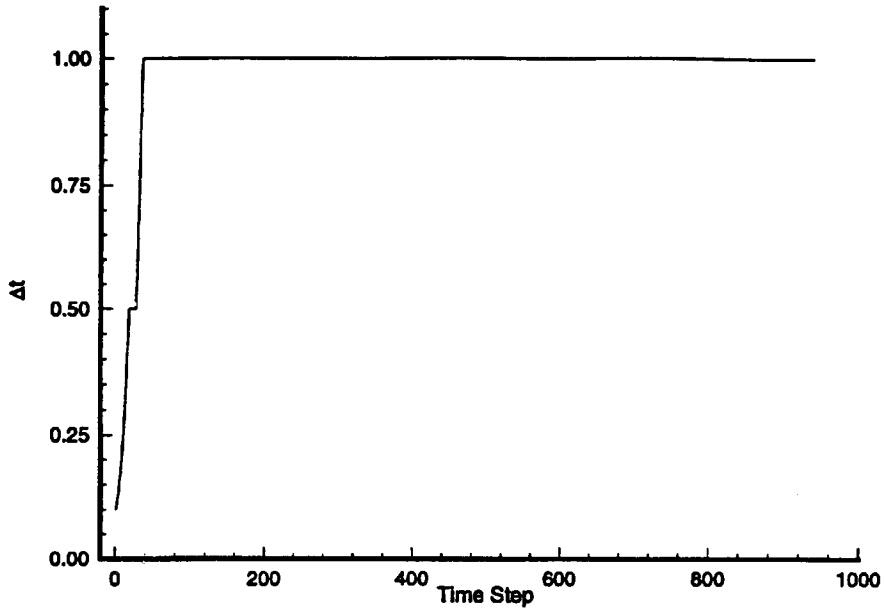


Fig. 6–99. Case 32 Time Step Size vs. Time Step

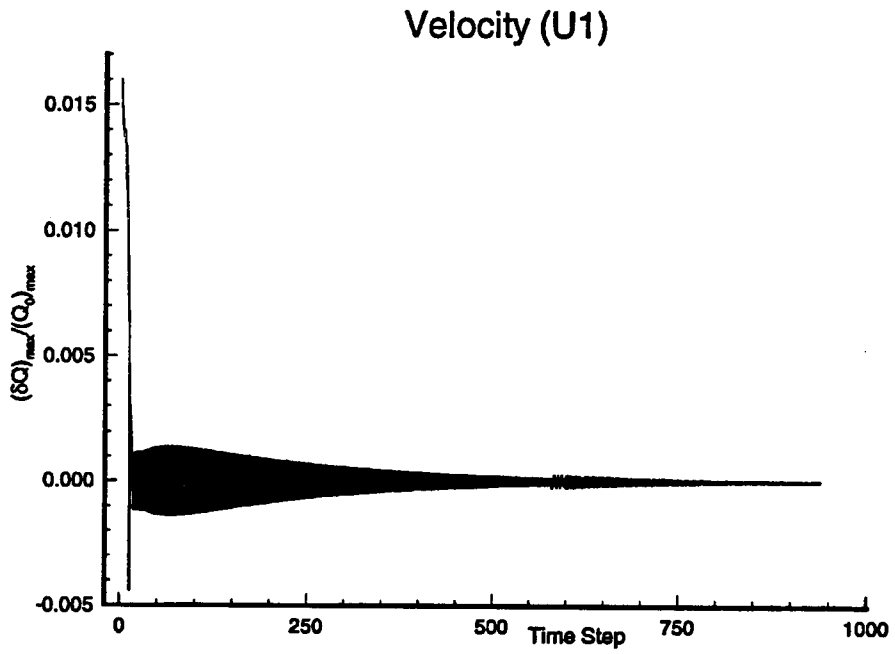


Fig. 6–100. Case 32 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Velocity (U1)

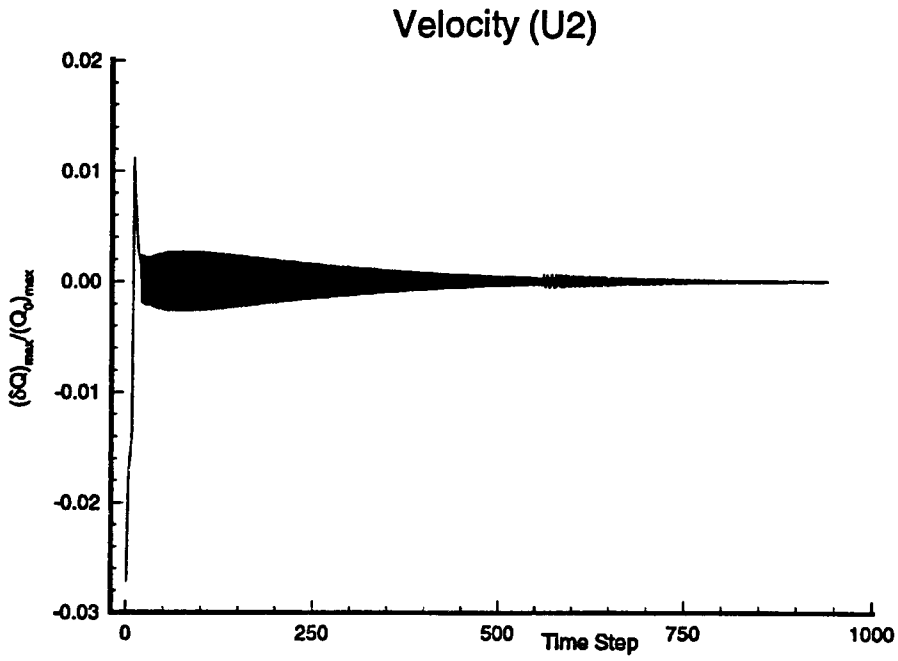


Fig. 6–101. Case 32 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Velocity (U2)

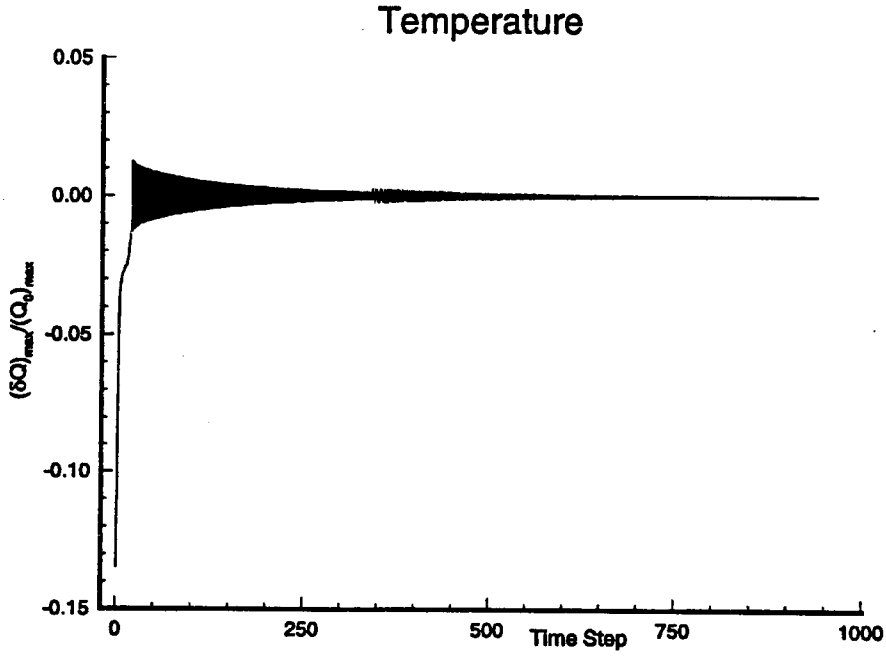


Fig. 6–102. Case 32 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Temperature

Time step size history and $\frac{(\delta Q)_{max}}{(Q_0)_{max}}$ for the velocity (U1 and U2) and temperature first-iterate fields for Case 33 are graphed in Figs. 6–103 through 6–106. These plots also demonstrate an oscillating, but deliberate progression towards steady-state convergence, similar to that shown in the previous case.

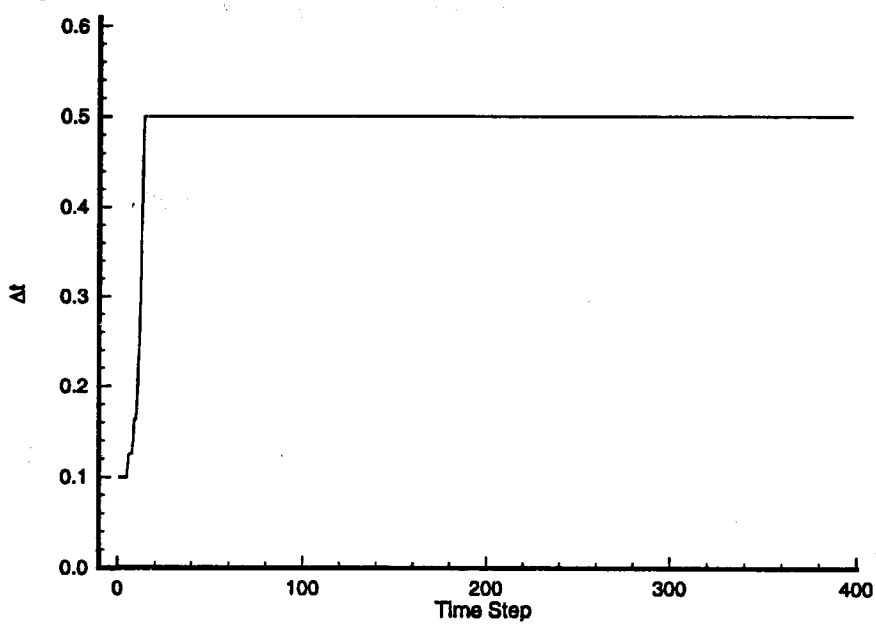


Fig. 6–103. Case 33 Time Step Size vs. Time Step

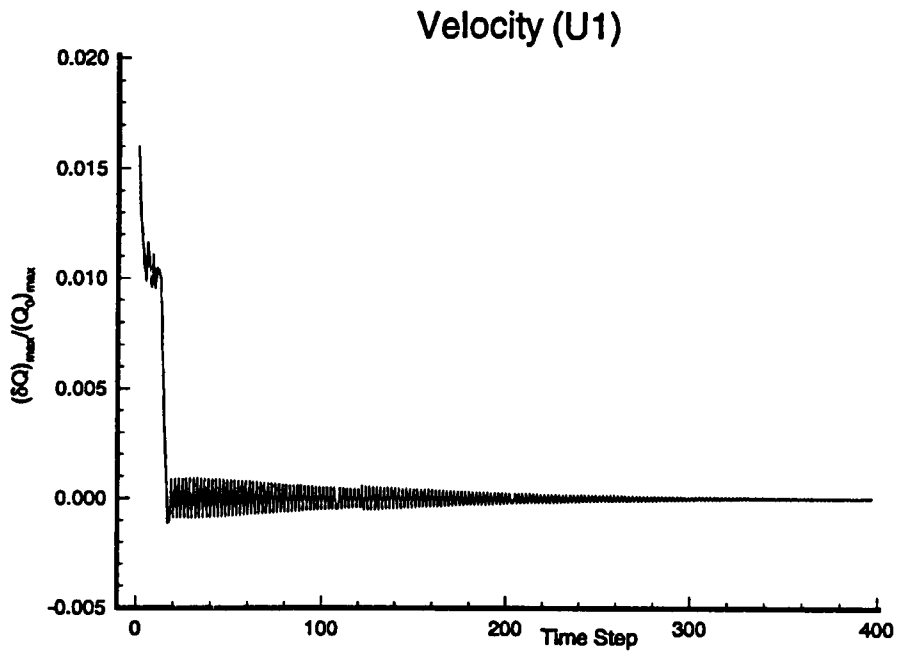


Fig. 6–104. Case 33 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Velocity (U1)

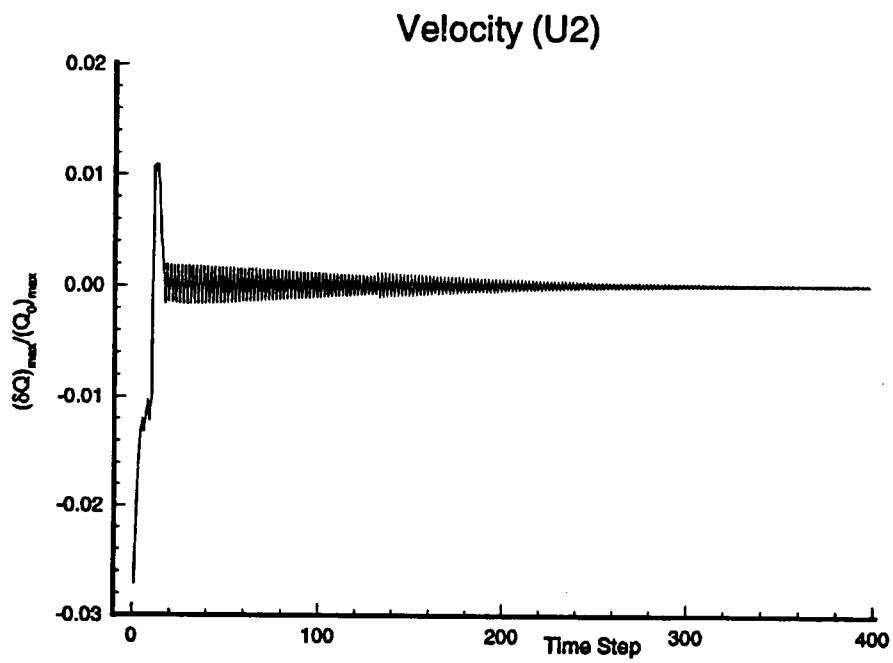


Fig. 6–105. Case 33 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Velocity (U2)

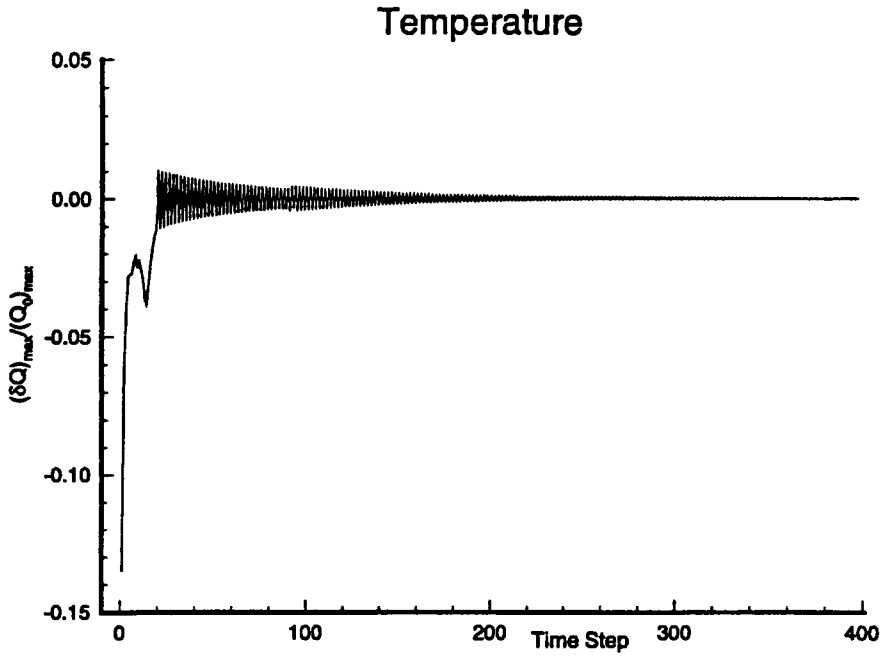


Fig. 6–106. Case 33 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Temperature

The new time step control method enabled steady-state convergence to be obtained in fewer time steps than the original method. The difference in the number of time steps for the two cases is again attributed to the difference in the maximum time step size obtained using the two methods. As stated in Sect. 6.3.1.1, larger time steps do not always provide the most efficient convergence to a steady-state solution.

6.3.2.2 $\theta = 1.0$ Cases

Two 2D penalty thermal cavity cases were run using the $\theta = 1.0$ time integration at $Ra = 1000$. Case 34 used the original time step control method with an initial time step size of 2.0, a maximum time step size of 10.0, and a time step multiplier of 1.1. This case converged in 11 time steps with a total CPU time of 629 s.

Plots of the time step size and $\frac{(\delta Q)_{max}}{(Q_0)_{max}}$ values for the velocity (U1 and U2) and temperature first-iterate fields at each time step for Case 34 are given in Figs. 6–107 through 6–110. These plots demonstrate a very smooth and robust progression towards steady-state convergence.

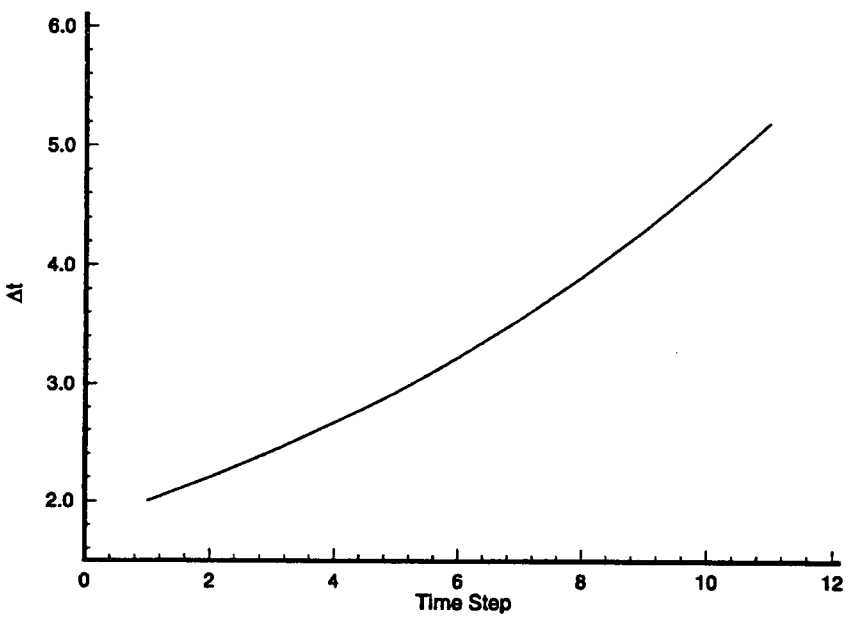


Fig 6–107. Case 34 Time Step Size vs. Time Step

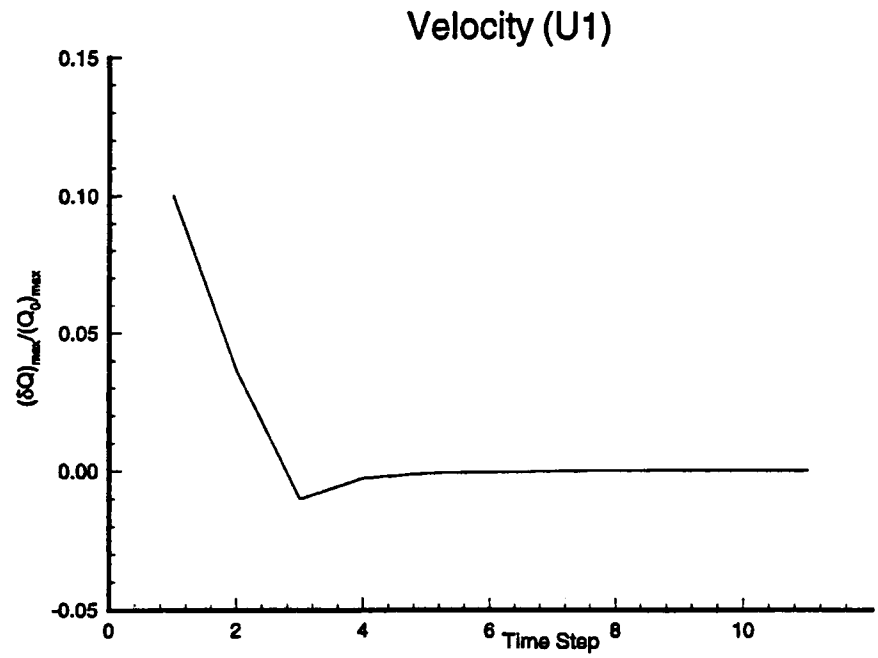


Fig. 6–108. Case 34 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Velocity (U1)

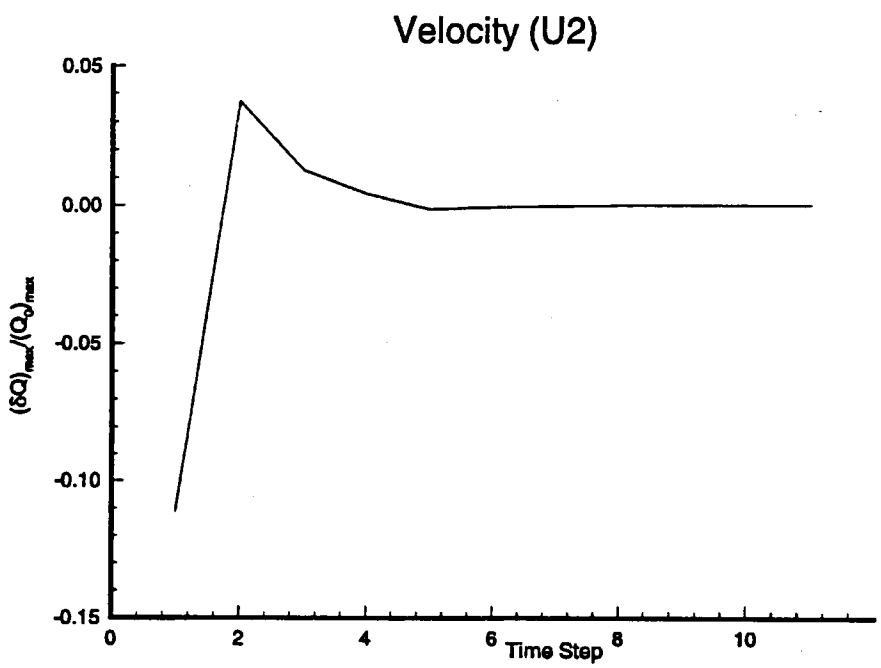


Fig. 6–109. Case 34 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Velocity (U2)

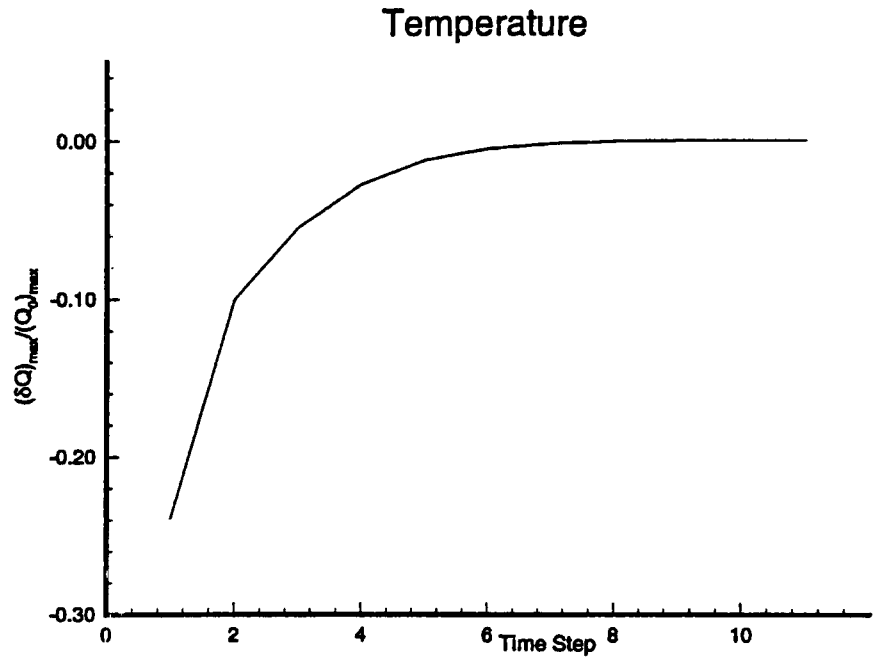


Fig. 6–110. Case 34 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Temperature

Case 35 used the new time step control method based on $\frac{|\delta Q|_{max}}{|Q_0|_{max}}$ for velocity (U1 and U2) and temperature with the same initial and maximum time step sizes as the previous case, as well as a minimum time step size of 1.0. This case reached convergence in 9 time steps with a total CPU time of 543 s.

Time step size history and $\frac{(\delta Q)_{max}}{(Q_0)_{max}}$ for the velocity (U1 and U2) and temperature first iterates for Case 35 are graphed in Figs. 6–111 through 6–114. These plots also demonstrate a very smooth and robust progression towards steady-state convergence.

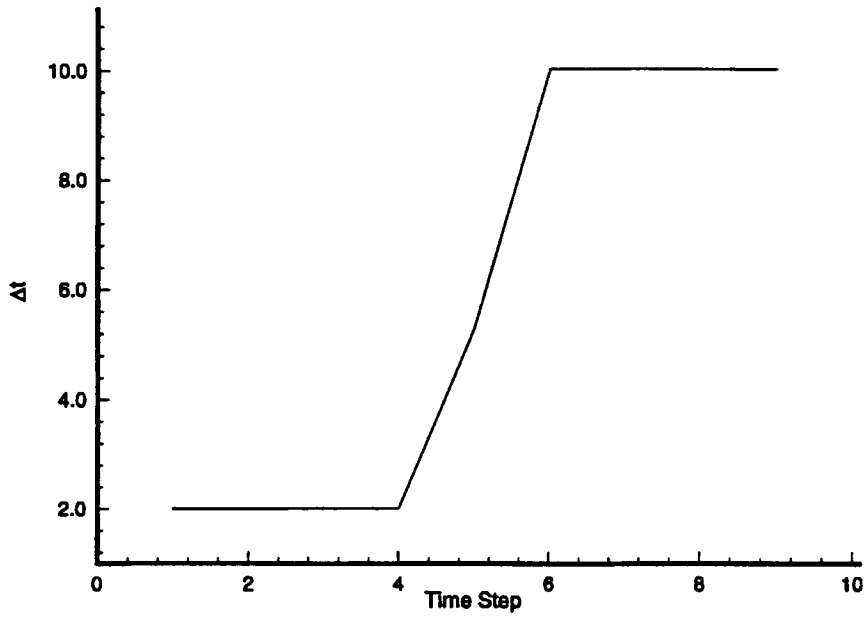


Fig 6–111. Case 35 Time Step Size vs. Time Step

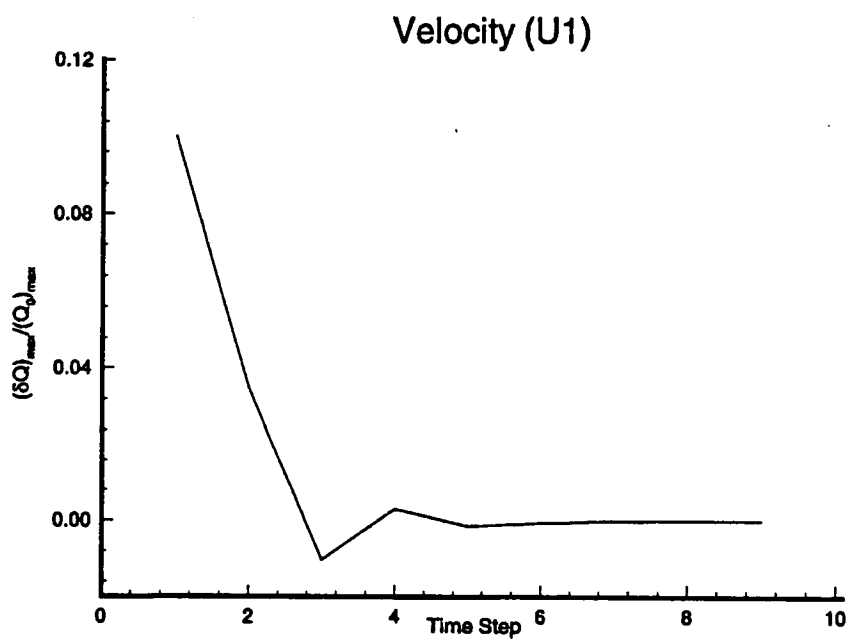


Fig. 6–112. Case 35 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Velocity (U1)

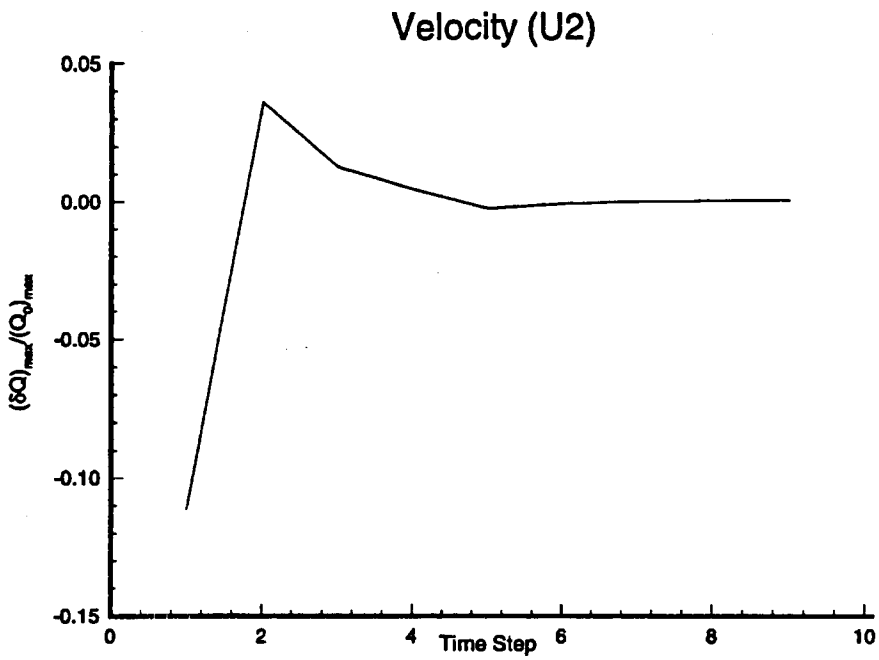


Fig. 6–113. Case 35 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Velocity (U2)

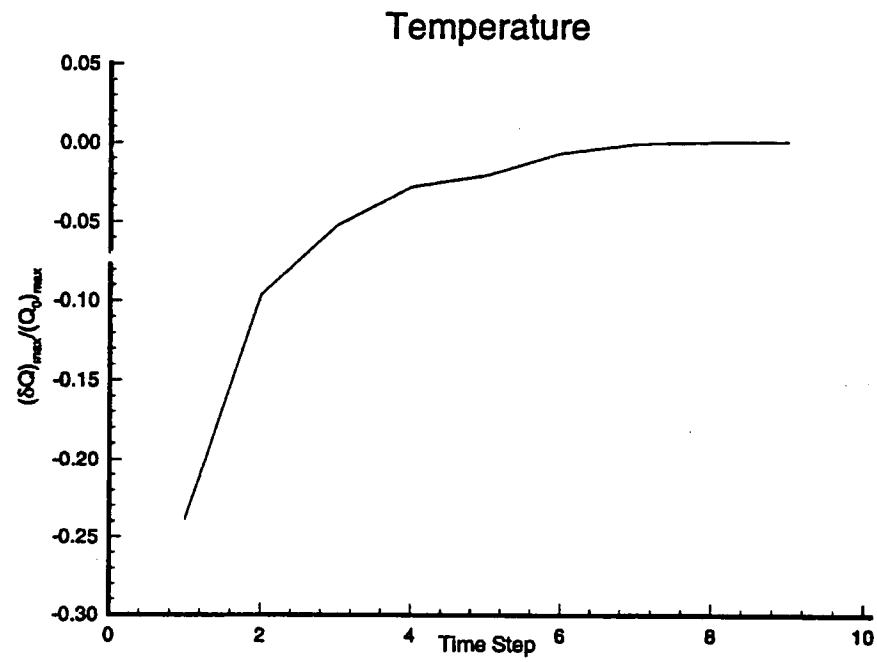


Fig. 6–114. Case 35 $(\delta Q)_{max}/Q_{max}$ Values vs. Time Step for Temperature

6.3.2.3 Summary, Ra = 1000 Thermal Cavity

The results from the Ra = 1000 penalty thermal cavity cases evaluate the effectiveness of both the original and the new time step control methods. These results were similar to those presented for the penalty algorithm driven cavity cases in Sect. 6.3.1. A summary of the Ra = 1000 penalty thermal cavity cases in provided in Table 6–9.

Table 6–9. Summary of Ra = 1000 Penalty Thermal Cavity Cases

Case No.	Method[1]	Initial Time Step Size	Maximum Time Step Size	Minimum Time Step Size	θ	Results Summary
32	Original (1.1)	0.1	0.5	N/A	0.5	Converged after 939 time steps in a total CPU time of 25,140 s
33	New $(\lvert(\delta Q)\rvert_{max}/\lvert Q\rvert_{max})^2$	0.1	0.5	0.05	0.5	Converged after 397 time steps in a total CPU time of 8199 s
34	Original (1.1)	2.0	10.0	N/A	1.0	Converged after 11 time steps in a total CPU time of 629 s
35	New $(\lvert(\delta Q)\rvert_{max}/\lvert Q\rvert_{max})^2$	2.0	10.0	1.0	1.0	Converged after 9 time steps in a total CPU time of 543 s

Notes:

[1]If the original method is designated, the term in parentheses represents the time step multiplier used in the case. If the new method is designated, the term in the parentheses represents the values used to control the time step method.

[2]The velocity (U1 and U2) and temperature solution fields were used.

Chapter 7

Conclusions

A total of 35 cases were run for the driven cavity and thermal cavity problems using CFD $WS^{\dagger}$ algorithms based on a vorticity-streamfunction form, the continuity constraint method, and the penalty method. These cases compared the original time step control method with the new time step control method. These cases confirmed the theory presented in Chapter 4, specifically:

- Conditionally stable problems, such as the $\theta = 0.0$ vorticity-streamfunction driven cavity and thermal cavity problems, require that the time step size be carefully selected to maintain a stable solution. The original time step control method, which was not designed for this, fails to provide the necessary time step size control to maintain solution stability. The new time step control method, however, does provide the necessary time step size control to maintain stability

of the solution. Thus, the new time step control method is essential for obtaining steady-state convergence using explicit time integration.

- Unconditionally stable problems, such as the $\theta = 0.5$ vorticity-streamfunction, continuity constraint, and penalty cavity problems and the $\theta = 1.0$ vorticity-streamfunction and penalty cavity problems do not require that the time step size be carefully selected to prevent divergence of the solution. However, if the time step size is too large, the Newton or quasi-Newton procedure may oscillate indefinitely.

- In some cases, the new time step control method, which tends to maintain tighter controls on the time step size, actually enables the solution to converge quicker (i.e., in fewer time steps) than does the original method. Larger time steps do not always provide the most effective and efficient solution of a problem.

Overall, the variety of input parameters in the new time step control method enables the user to tailor the method the fit the problem being solved. Such versatility may prove to be a valuable tool for more comprehensive CFD problems.

References

References

Baker, A. J. (1983), *Finite Element Computational Fluid Mechanics*, Hemisphere Publishing Corp.

Burnette, D. S. (1987), *Finite Element Analysis: From Concepts to Applications*, Addison-Wesley Publishing Co.

Computational Mechanics Corporation (1995), *AKCESS.4CFD, for Finite Element CFD Modeling in AKCESS.*, User Guide*, Draft.

Davies, A. J. (1980), *The Finite Element Method: A First Approach*, Clarendon Press, Oxford.

Freels, J. D. (1992), *A Taylor Weak Statement Finite Element Algorithm for Real-Gas Compressible Navier-Stokes Equations*, PhD Dissertation, The University of Tennessee, Knoxville.

Fung, Y. C. (1977), *A First Course in Continuum Mechanics*, 2nd Edition, Prentice-Hall, Inc.

Hahn, W. (1963), *Theory and Application of Liapunov's Direct Method*, Prentice-Hall, Inc.

Knudzen, J. G. and D. L. Katz (1958), *Fluid Dynamics and Heat Transfer*, McGraw-Hill, Inc.

Lehnigk, S. H. (1966), *Stability Theorems for Linear Motions (With an Introduction to Liapunov's Direct Method)*, Prentice-Hall, Inc.

Malvern, L. E. (1969), *Introduction to the Mechanics of a Continuous Medium*, Prentice-Hall, Inc.

White, F. M. (1974), *Viscous Fluid Flow*, McGraw-Hill, Inc.

White, F. M. (1979), *Fluid Mechanics*, McGraw-Hill, Inc.

Williams, P. T. (1993), *Continuity Constraint Method: A Finite-Element Computational Fluid Dynamics Algorithm for Incompressible Navier-Stokes Fluid Flows*, PhD Dissertation, The University of Tennessee, Knoxville.

Zienkiewicz, O. C. (1974), Constrained Variational Principles and Penalty Function Methods in Finite Element Analysis, in G. A. Watson (ed.), *Lecture Notes in Mathematics: Conference on the Numerical Solution of Differential Equations*, Springer-Verlag, Berlin.

Appendices

Appendix A

BOTTOM Hook FORTRAN Listing

```
      SUBROUTINE CNVTST (NTMOD,NGO,IQNOWB,IVARBF,KVARBF)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*1 FMOUT(80)
      CHARACTER*4 VARBS(500)
      CHARACTER*5 CDNALN,CDOWN,CALONG
      CHARACTER*8 VGMAX,QRATCN,DQMN,DQMA,QMXR,DELO,DELN
      COMMON /GVARBS/ VARBS
      COMMON /FMTOUT/ FMOUT
      COMMON /READS / IINP,IOUT,ITYPE,KOUNT
      COMMON /IORDER/ MORDER(20)
      COMMON /GROUP / NGROUP,NGRSET,NGPTST,NG,NVGJ,NGRSUM,MVFB,MVJB,KDSP
      COMMON /SOLVAR/ ISOLV,INTMTD,NAMSOL,IISOL(7)
      COMMON /DATSET/ ITRM,IDAT4,IDAT5,INT1,INT2,INT3,INSET2,
     1 KTRLI,KTRLO,ISCR,MZ(10)
      COMMON /CNVPRM/ TLAST,HPO,HPPO,HPS,HPSL,QMXSAV,QMXSST
      COMMON /LOCAT / INC,LOCIRZ(2),NIZ,NRZ,NINPUT,NIPRNT,IABNCD
      COMMON /ARYS  / RSCALR(100),ISCALR(100),NKIRZ,KIRZ(200,2,2)
      COMMON /GLOBAL/ RARRAY(100),IARRAY(100)
C:
      COMMON /MESDUM/ NSLINE,NSEGI,KRXLC,NSGBAS(4),NOPADD(32),NOSADD(16)
      COMMON /SPRLEL/ NSEGB,NSEGE,NSEGIL
      COMMON /VADRES/ ICOORD,ILF,ILGJ,IQLAST,IFIXRR
      COMMON /GETLOC/ IBNDY,IMCHE,IMTRL,ISCALE,IDETJ,IDETC,IETKJ,IPRES,
     * IUMAG,IUIJK,IRHO,IEDGE,IOUTLN,IGTUMG,INLX2,IDETE
      COMMON /CQSIZE/ LCOLS,KROWS,MPLANS,NVAR,IONES,NTRAN,ICIR,ICCQ(13)
      COMMON /CSTEPR/ NSTEPS,NITERS,NJACAD,NSHERS(17)
      COMMON /CPRINT/ NPTMOD,NSTRT,NSTOP,NPHERS(97)
      COMMON /CINTEG/ XP,HP,EPS,THETA,DTLIM,EPN,HRATIO,RATBAS,DQMXRF,
     * OTHERS(11)
C:    NINT4-NINT7 FOR FUTURE DEVELOPMENT
      COMMON /PRCOMS/ NITCNT,NSTPSV,NSTTST,NINT4,NINT5,NINT6,NINT7
      COMMON /QLOCS1/ MDDD1,MDDD2
      COMMON /RSV/ RATCON(20),DELTA1(20),DELOLD(20),DQMNDQ(20),DQMAL(20)
      COMMON /RESIDV/ VDG(20)
      COMMON /RESID0/ DTLIMX,DTLIMS,RATEMX,EPNSAV,AMAXLM
      COMMON /RESID3/ NITLST,NCON,NSTEPC,NCNCNT,NTTCNT
      COMMON /RESID4/ QMXRAT(20),NONCE
      COMMON /DCDQ  /DONPS,DQNPS
      EQUIVALENCE (KIRZ(08,1,2),IVRMAX),(KIRZ(08,2,2),KVRMAX)
      COMMON /HTHDEL/HTHMLT(20),HOMMLT(20),DELTAQ(20),NFXX(20),NFXL(20)
      COMMON /RZA   / RZ(01000)
      COMMON /IZA   / IZ(0100)
      COMMON /XLEP1 / DQQMRA(20),EPSET(20),EPSETHI(20),EPSETLO(20)
      COMMON /XLEP2 / V1O(20),V1N(20),STEVIND(20),HPP(20),DQQDIF(20)
      COMMON /XLEP3 / EPSETSUM(20),EPDIFSUM(20),EPDIF(20),DQQMRAO(20)
      COMMON /XLEP4 / QMXRATA(20)
      COMMON /ILEP1 / NPLACE(20),IGNORE(20)
C:
      EQUIVALENCE (KIRZ(10,1,1),IPRINT),(KIRZ(10,2,1),KPRINT)
      EQUIVALENCE (KIRZ(22,1,1),LVARBF),(KIRZ(22,2,1),MVARBF)
      EQUIVALENCE (KIRZ(01,1,2),ICOEFN),(KIRZ(01,2,2),KCOEFN)
      EQUIVALENCE (KIRZ(04,1,2),IQNOW ),(KIRZ(04,2,2),KQNOW )
C:
      EQUIVALENCE ( RSCALR(01), TFINAL )
      EQUIVALENCE ( RSCALR(21), QMX1ST )
      EQUIVALENCE ( RSCALR(22), GLOBMX )
      EQUIVALENCE ( RSCALR(23), CNVGNC )
      EQUIVALENCE ( RSCALR(24), BRB    )
      EQUIVALENCE ( RSCALR(25), QMXLST )
C:
      EQUIVALENCE ( IARRAY(03), NNODE  )
      EQUIVALENCE ( IARRAY(04), NITTOT )
      EQUIVALENCE ( IARRAY(14), IFLITR )
      EQUIVALENCE ( IARRAY(17), NDIM   )
```

```
      EQUIVALENCE ( IARRAY(18), NSTEP  )
      EQUIVALENCE ( IARRAY(19), NITER  )
      EQUIVALENCE ( IARRAY(20), IRK    )
      EQUIVALENCE ( IARRAY(24), NHMAX  )
      EQUIVALENCE ( IARRAY(25), IINODE )
      EQUIVALENCE ( IARRAY(28), NSMAX  )
      EQUIVALENCE ( IARRAY(32), NGOSS  )
C
C
C
C
C:
      EQUIVALENCE ( RARRAY(22), HOMTHT )
      EQUIVALENCE ( RARRAY(23), HMULT  )
      EQUIVALENCE ( RARRAY(24), DQMAXX )
      EQUIVALENCE ( RARRAY(25), EPSITR )
      EQUIVALENCE ( RARRAY(26), EPSPHI )
C:
      DATA  HPX,HPQ,NTLIM /0.0,.90,0/
      DATA  VGMAX,QRATCN,DQMN,DQMA,QMXR,DELO,DELN,DQMR1/
     * 'RESMAX','RATIO','DELQMAX','Q-MAX','DQM/QMAX','DQ-OLD',
     * 'DQ-NEW','DQ/QMAX'/
      DATA  STEV1,STEV2,STEV3,STEV9,EPSE,EPSH,EPSL,EPSU,EPSO/
     * 'V1OLD','V1NEW','STEVIND','HPP','EPSET','EPSETHI',
     * 'EPSETLO','EPSETSU','EPSETOL'/
      DATA  CDOWN,CALONG /'DOWN ','ALONG'/
C:
 9215 FORMAT (9H BOTTOM  ,10I5)
 9400 FORMAT (' EPSET(',I1,')=',F10.7,' EPSETHI(',I1,')=',F10.7,
     1 ' EPSETLO(',I1,')=',F10.7)
 9600 FORMAT (1H ,I3,1H,,2I4,5H, VAR ,
     1 1P,E11.4,5H, DLN ,E11.4,4H, HP,E10.3,4H, DL ,E10.3)
 9610 FORMAT (1X,A8,1X,I3,I3,1P,5E11.4:/(16X,5E11.4))
 9611 FORMAT (I3,I3,F6.3,1P,6E10.3)
 9612 FORMAT (I3,I3,'    EPS=',F8.6,' EPSITR=',F8.6)
 9613 FORMAT (I3,I3,' RATTST=',F8.6,'    EPN=',F8.6,' QMX1ST=',
     1 1PE10.3,'  DQMAXX=',1PE10.3)
 9614 FORMAT (I3,I3,' HPPO=',1PE13.6,' HP=',1PE13.6,'  NSTEPD=',
     1 I4,'   NVGJ=',I2)
 9616 FORMAT (I3,I3,'  NSMAX=',I3,' NSTTST=',I3,' NGPTST=',I3,
     * ' NGRSET=',I2,' NGRSPR=',I2,' NGRSPT=',I2)
 9617 FORMAT (I3,I3,' NSTEPD=',I3,' NSTEPJ=',I3,' NRATBS=',I3,
     * '   NCON=',I2,' NRESET=',I2,'    IRK=',I2)
 9618 FORMAT (I3,I3,' NCNCNT=',I3,' NGROUP=',I3,'  NRATE=',I3,
     * ' IWCHDQ=',I2,'  NTMOD=',I2,'   NVGJ=',I2)
 9619 FORMAT (' NSTEPJ=',I4,' NSTEPK=',I4,' NGROUP=',I2,
     * ' NGRSPR=',I2,'  NGRSET=',I2,' IWCHDQ=',I2)
 9625 FORMAT (I4,',',1PE13.6,',',1PE13.6,',',1PE13.6,
     * ',',1PE13.6,',',1PE13.6,',')
 9626 FORMAT (I4,',')
 9627 FORMAT (1PE13.6,',')
 9630 FORMAT ('   STEP ',A5,1P,E12.4,I4,2I2,2X,3E12.4)
 9640 FORMAT (' PROGRAM STOPPED: MAX Q =',1P,E14.5,', LIMIT = ',E14.5)
 9660 FORMAT (' XDTLIM ',I4,I2,2I5,1P,2E13.5,0P,2F9.5)
 9670 FORMAT (' STEP SIZE LESS THAN ',1P,4E13.5)
 9671 FORMAT (' STEP SIZE LESS THAN ZERO ',2I2,1P,4E13.5)
 9680 FORMAT (' NS NCN DT EPI ',3I5,1P,2E14.6)
C:
C:: CNVTST
C:
      ZERO = 0.0
      if (hp.lt.zero.or.xp.gt.tfinal)
     * WRITE (*,9671) 3,NCON,HPO,HP,XP,TFINAL
      NSEGI = 1
```

```
      NSEGIL = NSEGI
      SMALL = 1.0E-15
      ZP0001 = 0.0001
      ZP01 = 0.01
      ZPT1 = 0.1
      ZPT2 = 0.2
      ZPT25 = 0.25
      ZPT5 = 0.5
      ZPT8 = 0.8
      ZPT9 = 0.9
      ONEP = 1.0
      ONE3 = 1.0E3
      ONE25 = 1.025
      ONEP05 = 1.05
      ONEP1 = 1.1
      ONEP25 = 1.25
      ONEP4 = 1.4
      ONEP5 = 1.5
      THREE = 3.0
      FOUR = 4.0
      TENP = 10.0
      HUND = 100.0
      HPCHCK = ZPT2*DTLIM
      RATTST = ZERO
      EPNRAT=ZERO
      IF (NITTOT.LE.NITERS) NSTTST=0
      IF (NITERS.EQ.1) NSTTST=1
      NRATBS = RATBAS
      NITCNT = NITER
      NGO =0
      NGOSS =0
      NCON=1
C:
C:  ************************************************************
C:  *                                                          *
C:  *            NEW TIMESTEPPING SETUP PARAMETERS             *
C:  *                                                          *
C:
C:      IF NFLAG1=0, OLD TIMESTEPPING METHOD IS SELECTED; IF
C:      NFLAG1=1, NEW TIMESTEPPING METHOD IS SELECTED
      NFLAG1 = 0
C:
C:      SET NDQSET=0 IF TIME STEP CONTROL IS TO BE BASED ON
C:      (DQ/Q)MAX;  SET NDQSET=1 IF TIME STEP CONTROL IS TO BE
C:      BASED ON (DQ)MAX/(Q)MAX
      NDQSET = 0
C:
C:      SET NFLAG2=1 IF MINIMAL OUTPUT IS DESIRED
      NFLAG2 = 1
C:
      IF (NFLAG1.NE.1) GO TO 1100
C:
C:      MARK THE FIRST TIME STEP AS NSTEPI
      IF (NITTOT.EQ.1) NSTEPI = NSTEP
C:
C:      SET NSTEPJ TO THE DESIRED TIMESTEP TO BEGIN THE NEW
C:      TIMESTEPPING METHOD
      IF (NITTOT.EQ.1) NSTEPJ = NSTEPI + 20
C:
C:      SET NSTEPK TO THE DESIRED TIMESTEP TO BEGIN AVERAGING
C:      THE DQ/QMAX VALUES TO DETERMINE THE VALUE OF EPSET
      IF (NITTOT.EQ.1) NSTEPK = NSTEPJ - 2
C:
C:      SET THE FACTOR FOR THE APPROXIMATE NUMBER OF TIMESTEPS
```

```
C:      AT WHICH DELTA-T WILL BE ADJUSTED (BASED ON THE AVERAGE
C:      CHANGE IN DQ/QMAX FOR EACH VALUE OF NP)
      EPSETF = TENP
C:
C:      SET THE VALUE AT WHICH EPSET WILL BE MULTIPLIED TO
C:      DETERMINE THE MAXIMUM VALUE OF EPSETHI
      EPSETHIF = ONEP25
C:
C:      SET THE VALUE AT WHICH EPSET WILL BE MULTIPLIED TO
C:      DETERMINE THE MINIMUM VALUE OF EPSETLO
      EPSETLOF = ZPT8
C:
C:      SET NEPSET = 0 TO AVOID PRINTING THE VALUES OF EPSET,
C:      EPSETHI, AND EPSETLO AT EACH TIMESTEP
      NEPSET = 0
C:
      NPLACE1 = 0
C:
C:      SET THE NUMBER OF CONSECUTIVE TIME STEP DELTA-T CHANGE
C:      REQUESTS BEFORE THE VALUE OF EPSET, EPSETHI, AND EPSETLO
C:      IS ADJUSTED
      NPLACEF = 100000
C:
C:      SET THE MINIMUM TIME STEP SIZE
      DTMINX = 0.005
C:
C:      SET THE NP VALUES THAT SHOULD BE IGNORED BY THE NEW
C:      TIMESTEPPING METHOD (IF IGNORE(NP) = NP, THE DQ/QMAX
C:      VALUE WILL BE IGNORED IN DETERMINING THE NEW VALUES
C:      FOR HPP(NP)).
      IGNORE(1) = 0
      IGNORE(2) = 0
      IGNORE(3) = 3
      IGNORE(4) = 4
      IGNORE(5) = 5
      IGNORE(6) = 6
      IGNORE(7) = 7
      IGNORE(8) = 8
      IGNORE(9) = 9
      IGNORE(10) = 10
C:
 1100 CONTINUE
C:                                                                *
C:  *
C:  ************************************************************
C:
      NHMAX = 0
      RATE = ONEP
      QMXITR = ONE3*EPS
      NGRPST = IABS(MORDER(NVGJ))
      NGPMOD = 0
      IF (NGRPST.GT.0) NGPMOD = MOD(NSTEP,NGRPST)
      IF (NGPMOD.NE.0) GO TO 2510
      IWCHDQ = 0
      IF (DQMXRF.LE.1.0) IWCHDQ = 1
      IF (DQMXRF.EQ.2.0) IWCHDQ = 2
      IF (ISOLV.NE.1.AND.ISOLV.NE.3) IWCHDQ = 0
      IGTDET = 0
      IF ((ISOLV.GT.2.OR.NAMSOL.EQ.1).AND.IDETC.GT.0) IGTDET = 1
      IF (MDDD2.GT.0.AND.IDETC.GT.0) IGTDET = 2
      DO 2500 NPP=1,KVARBF
      NP = IZ(IVARBF+NPP)
      IF (NITTOT.EQ.1) QMXRAT(NP)=ZERO
      IF (NITTOT.EQ.1) V1O(NP)=ZERO
```

```
      DQMAL(NP) = 1.0
      IQLOC = (NP-1)*NNODE
      NNMAX = 0
      NQMAX = 0
      NDMAX = 0
      DELTQP = ZERO
      DRMAXX = ZERO
      DQQMAX = ZERO
      QQMAX = RZ(IVRMAX+NP)
      RATCNN = ONEP
C:
C:  ***************************************************************
C:  *                                                             *
C:  *                    MAXIMUM Qs, DQs, AND DQ/Qs               *
C:  *                                                             *
      DO 2000 I=1,NNODE
      IF (NFLAG1.EQ.1) GO TO 2050
      IF (IGTDET.EQ.0) GO TO 2050
      IF (IGTDET.EQ.1) RZ(ILF+IQLOC+I) = RZ(ILF+IQLOC+I)*RZ(IDETC+I)
      IF (IGTDET.EQ.2.AND.NP.EQ.1)
     *RZ(ILF+IQLOC+I) = RZ(ILF+IQLOC+I)*RZ(MDDD2+I)
      IF (IGTDET.EQ.2.AND.NP.GE.2)
     *RZ(ILF+IQLOC+I) = RZ(ILF+IQLOC+I)*RZ(IDETC+I)
 2050 CONTINUE
C:
C:      FIND LARGEST DELTAQ AT EACH NODE
      DQM  = ABS(RZ(ILF+IQLOC+I))
      IF (DQM.GT.DRMAXX) NNMAX=I
      IF (DQM.GT DRMAXX) DRMAXX = DQM
C:
C:      FIND LARGEST Q AT EACH NODE
      QM = ABS(RZ(IQNOWB+IQLOC+I))
      IF (QM.GT.ABS(QQMAX)) NQMAX = I
      IF (QM.GT.ABS(QQMAX)) QQMAX = QM
C:
C:      FIND LARGEST DELTAQ/Q AT EACH NODE
      IF (QM.NE.ZERO) DQQRAT  = DQM/QM
      IF (QM.LT.EPS) DQQRAT = EPS
      IF (DQQRAT.GT.DQQMAX) NDMAX=I
      IF (DQQRAT.GT.DQQMAX) DQQMAX = DQQRAT
C:
C:
 2000 CONTINUE
      NFXL(NP) = NFXX(NP)
      IF (NITER.EQ.1) NFXX(NP) = NNMAX
      IF (NNMAX.GT.0) DELTQP=RZ(ILF+IQLOC+NNMAX)
C:
C:      (DQ)MAX FOR VARIABLE NP
      DQMNDQ(NP)=DELTQP
      IF (QQMAX.LE.QMXITR) QQMAX=QMXITR
      IF (NFLAG1.EQ.1) GO TO 2001
      IF (DELTQP.EQ.ZERO.AND.IWCHDQ.EQ.1) DELTQP = ZPT9*EPS
      IF (DELTQP.EQ.ZERO.AND.IWCHDQ.EQ.2) DELTQP = ZPT9*EPS*QQMAX
 2001 CONTINUE
C:      (DQ)MAX/(Q)MAX FOR VARIABLE NP
      QMXRAT(NP) = DELTQP/QQMAX
      QMXRATA(NP) = ABS(QMXRAT(NP))
C:      (Q)MAX FOR VARIABLE NP
      DQMAL(NP) = QQMAX
C:      (DQ/Q)MAX FOR VARIABLE NP
      DQQMRA(NP) = DQQMAX
C:  *                                                             *
C:  ***************************************************************
C:
```

```
C:
C:  ***********************************************************************
C:  *                                                                     *
C:  *      INDIVIDUAL STABILITY EVALUATION TERMS FOR VARIABLE NP          *
C:  *                                                                     *
      IF (NITER.GT.1) GO TO 2110
      SUMG1SQ = ZERO
      DO 2100 I=1,NNODE
      G1SQ = ((RZ(ILGJ+IQLOC+I))**2)
      SUMG1SQ = SUMG1SQ + G1SQ
 2100 CONTINUE
      IF (NSTEPD.LT.5) V1O(NP) = V1N(NP)
      V1N(NP) = ((SUMG1SQ)**0.5)
      STEVIND(NP) = (V1N(NP) - V1O(NP))/HP
      IF (NP.GT.NGRSPR.OR.STEVIND(NP).LE.ZERO) GO TO 2110
      IF (NSTEP.GT.NSTEPI) WRITE (ISOT,*) NSTEP,NITER,NVGJ,NP,
     * '*** WARNING: INSTABILITY ***'
      IF (NSTEP.GT.NSTEPI) WRITE (*,*) NSTEP,NITER,NVGJ,NP,
     * '*** WARNING: INSTABILITY ***'
 2110 CONTINUE
C:  *                                                                     *
C:  ***********************************************************************
C:
      IF (N1TER.EQ.1) DELTA1(NP)=DELTQP
      IF (IWCHDQ.EQ.1) DELTAQ(NP) = DELTQP
      IF (IWCHDQ.EQ.2) DELTAQ(NP) = QMXRAT(NP)
      IF (NITER.EQ.1.AND.NITERS.GT.1) DELOLD(NP) = DELTAQ(NP)
      IF (NITER.EQ.1) GO TO 2400
      DELNP = DELTA1(NP)
      RATCNN = 0.0
      IF (DELNP.NE.0.0) RATCNN = ONEP-ABS(DELTQP/DELNP)
      IF (RATCNN.GT.ONEP) RATCNN = ZERO
      IF (RATCNN.EQ.ZERO) RATCNN = ONEP
      IF (RATCNN.LT.RATTST.AND.ABS(DELTQP).LT.ZP0001) RATCNN = ONEP
      IF (RATCNN.LT.RATE.AND.ABS(DELTQP).GT.ZPT2*ABS(DELOLD(NP)))
     * RATE = RATCNN
      RATCON(NP)=RATCNN
 2400 CONTINUE
C:
 2500 CONTINUE
C:
 2510 CONTINUE
C:  *                                                                   *
C:  *********************************************************************
C:
      IF (NITTOT.GT.1) GO TO 3000
      HPPO = HP
      DTLIMX = DTLIM
 3000 CONTINUE
      OSHPR = ONEP/HRATIO
      AMAXQ = ZERO
      IF (IRK.EQ.0.AND.NITER.EQ.1) QMX1ST = ONEP
      IF (QMX1ST.EQ.ZERO) QMX1ST = EPS
      DO 3200 NPP=1,KVARBF
      NP = IZ(IVARBF+NPP)
      DQN = ABS(DQMAL(NP))
C:      FIND LARGEST VALUE OF ALL VARIABLES
      IF (DQN.GT.AMAXQ) AMAXQ=DQN
      QDMAXX = ABS(DELTAQ(NP))
      IF (QDMAXX.NE.0.0) RZ(IVRMAX+NP)=QDMAXX
      IF (QDMAXX.LE.DQMAXX) GO TO 3200
      IF (ISOLV.EQ.1.OR.ISOLV.EQ.3) DQMAXX = QDMAXX
      RATER = ONEP-ABS(DQMNDQ(NP)/QMX1ST)
 3200 CONTINUE
```

```
      IF (IRK.EQ.1.AND.NSTEP.NE.NSTPSV) QMXSAV = QMXSST
      if (dqmaxx.eq.zero) dqmaxx = zpt9*epsitr
      IF (DQMAXX.EQ.ZERO) QMXSST = EPS
      IF(NITER.EQ.1.AND.NSTEP.NE.NSTPSV.AND.DQMAXX.NE.ZERO)QMXSST=DQMAXX
      IF (IRK.EQ.0.AND.NITER.EQ.1) QMXLST = QMX1ST
      IF (IRK.EQ.0.AND.NITER.EQ.1.AND.DQMAXX.NE.ZERO) QMX1ST = DQMAXX
      IF (IRK.EQ.1) QMX1ST = QMXSAV
      IF (NSEGIL.NE.NSEGI) EPSITR = HUND*EPS
      IF (NSEGIL.NE.NSEGI) EPSPHI = HUND*EPSITR
      NSTPSV = NSTEP
      IF (QMX1ST.EQ.ZERO.AND.DQMAXX.NE.ZERO) QMX1ST = DQMAXX
      IF (QMX1ST.GT.GLOBMX) GLOBMX=QMX1ST
      ARB = GLOBMX/QMX1ST
      BRB = ONEP
      RATTST = ZPT5 - ZPT1*BRB
      CDNALN = CDOWN
      IF (NITERS.EQ.2) RATTST = ZPT5*ZPT5
      RATTS2 = ONEP4*RATTST
      DTLADD=ZPT1
      IF (BRB.GT.CNVGNC) DTLADD=ZERO
      DTLIMY = DTLIM+DTLADD
      EPMULT = ZPT25
      NRESET = 0
      DQTEST = ONEP5
      EPNMX  = DQTEST*EPN
      IF (NFLAG1.EQ.1) GO TO 4005
      IF (NSMAX.GE.0) GO TO 9000
 4005 CONTINUE
      NITSAV = NITER
      NRATE = 0
      RATEMN = ZERO
      IF (NSTTST.EQ.0) GO TO 4020
      IF (NITER.EQ.1.AND.NITERS.GT.1.AND.DQMAXX.GE.EPSITR) GO TO 4020
      IF (NVGJ.EQ.1) RATEMN = ONEP
      DO 4010 NPP=1,KVARBF
      NP = IZ(IVARBF+NPP)
      DQNP = ABS(DELTAQ(NP))
      DONP = ABS(DELOLD(NP))
      IF (NVGJ.EQ.1.AND.RATEMN.GT.RATCON(NP)) RATEMN=RATCON(NP)
      IF (RATCON(NP).EQ.ONEP) NRATE=NRATE+1
      IF (DQNP.GT.0.8*QMX1ST) NRESET = -1
      IF (DQNP.LT.QMX1ST) GO TO 4010
      IF (DQNP.GT.DONP) EPMULT = ZPT1*DONP/DQNP
      IF (DQNP.GT.DONP) DQNPS = DQNP
      IF (DQNP.GT.DONP) DONPS = DONP
      IF (EPMULT.LT.ZPT25) EPMULT = ZPT25
 4010 CONTINUE
 4020 CONTINUE
C:
C: ***************************************************************
C: *                                                             *
C: *    DETERMINE THE VALUES FOR EPSET, EPSETHI, AND EPSETLO     *
C: *                                                             *
C:
      IF (NSTEP.LE.NSTEPK-1.OR.NSTEP.GT.NSTEPJ-1.OR.NITER.GT.1
     * .OR.NVGJ.NE.NGROUP) GO TO 4520
      IF (NDQSET.EQ.1) GO TO 4510
      I = 0
 4505 CONTINUE
      I = I + 1
      IF (NSTEP.EQ.NSTEPK)  EPSETSUM(I) = ZERO
      IF (NSTEP.EQ.NSTEPK)  EPDIFSUM(I) = ZERO
      IF (NSTEP.EQ.NSTEPK)  DQQMRAO(I) = DQQMRA(I)
      EPSETSUM(I) = DQQMRA(I) + EPSETSUM(I)
```

```
      DQQDIF(I) = DQQMRA(I) - DQQMRAO(I)
      IF (DQQDIF(I).LT.ZERO) DQQDIF(I) = -DQQDIF(I)
      EPDIFSUM(I) = DQQDIF(I) + EPDIFSUM(I)
      DQQMRAO(I) = DQQMRA(I)
C:      IF (NSTEP.NE.NSTEPJ-1) GO TO 4506
      SNSTEP = NSTEP-(NSTEPK-1)
      PNSTEP = NSTEP-NSTEPK
c     EPSET(I) = EPSETSUM(I)/(NSTEP - (NSTEPK-1))
      IF (SNSTEP.NE.ZERO) EPSET(I) = EPSETSUM(I)/SNSTEP
      IF (EPSET(I).LT.SMALL) EPSET(I) = ONEP
c     EPDIF(I) = EPDIFSUM(I)/(NSTEP - NSTEPK)
      IF (PNSTEP.NE.ZERO) EPDIF(I) = EPDIFSUM(I)/PNSTEP
      EPSETHI(I) = (EPSET(I) + EPSETF*EPDIF(I))
      EPSETLO(I) = (EPSET(I) - EPSETF*EPDIF(I))
      IF (EPSETHI(I).GT.EPSET(I)*EPSETHIF)
     * EPSETHI(I) = EPSETHIF*EPSET(I)
      IF (EPSETLO(I).LT.EPSET(I)*EPSETLOF)
     * EPSETLO(I) = EPSETLOF*EPSET(I)
 4506 CONTINUE
      IF (I.LT.NP) GO TO 4505
      GO TO 4520
 4510 CONTINUE
      I = 0
 4515 CONTINUE
      I = I + 1
      IF (NSTEP.EQ.NSTEPK)  EPSETSUM(I) = ZERO
      IF (NSTEP.EQ.NSTEPK)  EPDIFSUM(I) = ZERO
      IF (NSTEP.EQ.NSTEPK)  DQQMRAO(I) = QMXRATA(I)
      EPSETSUM(I) = QMXRATA(I) + EPSETSUM(I)
      DQQDIF(I) = QMXRATA(I) - DQQMRAO(I)
      IF (DQQDIF(I).LT.ZERO) DQQDIF(I) = -DQQDIF(I)
      EPDIFSUM(I) = DQQDIF(I) + EPDIFSUM(I)
      DQQMRAO(I) = QMXRATA(I)
C:      IF (NSTEP.NE.NSTEPJ-1) GO TO 4516
      SNSTEP = NSTEP-(NSTEPK-1)
      PNSTEP = NSTEP-NSTEPK
c     EPSET(I) = EPSETSUM(I)/(NSTEP - (NSTEPK-1))
      IF (SNSTEP.NE.ZERO) EPSET(I) = EPSETSUM(I)/SNSTEP
      IF (EPSET(I).LT.SMALL) EPSET(I) = ONEP
c     EPDIF(I) = EPDIFSUM(I)/(NSTEP - NSTEPK)
      IF (PNSTEP.NE.ZERO) EPDIF(I) = EPDIFSUM(I)/PNSTEP
      EPSETHI(I) = (EPSET(I) + EPSETF*EPDIF(I))
      EPSETLO(I) = (EPSET(I) - EPSETF*EPDIF(I))
      IF (EPSETHI(I).GT.EPSET(I)*EPSETHIF)
     * EPSETHI(I) = EPSETHIF*EPSET(I)
      IF (EPSETLO(I).LT.EPSET(I)*EPSETLOF)
     * EPSETLO(I) = EPSETLOF*EPSET(I)
 4516 CONTINUE
      IF (I.LT.NP) GO TO 4515
 4520 CONTINUE
C:
C:      ******************************************************
C
C
C
C
C
C
C:
      IF (NITMOT.EQ.1) HPPO = HP
      TSTITR=ONEP5
      IF (QMXLST.LT.EPSITR) TSTITR = ONEP1
      IF (QMXLST.LT.TENP*EPS) TSTITR = ONEP
      IF (IRK.EQ.1) TSTITR=THREE
```

```
      IF (NFLAG1.EQ.1) GO TO 4600
      IF (IRK.EQ.0.AND.QMXLST.LT.EPSITR.AND.NSTTST.EQ.1)
     * EPN=TSTITR*QMXLST
      IF (NITTOT.GT.1.AND.NSEGIL.NE.NSEGI) EPN=QMX1ST
      IF (NSEGIL.NE.NSEGI) NSEGIL=NSEGI
      IF (EPN.LE.ZERO) EPN=ONEP
      EPNLO = ZPT9*EPN
      EPNHI = TSTITR*EPN
      IF (QMXLST.LT.TENP*EPS) EPNLO = ZPT8*EPN
      IF (QMXLST.LT.TENP*EPS) EPNHI = ONEP1*EPN
4600  CONTINUE
      EPNRAT=TSTITR*QMX1ST
      IF (NFLAG1.EQ.1) GO TO 4605
      IF ((HP.GT.HPCHCK.OR.NSTEP.GT.5).AND.DQMAXX.GT.EPNRAT) GO TO 4640
      IF (NITER.GT.1.AND.NRESET.GT.0.AND.IRK.EQ.0) GO TO 4640
4605  CONTINUE
      IF (NITER.GT.1) GO TO 4690
      IF (NFLAG1.NE.1) GO TO 4610
      IF (NVGJ.NE.NGROUP) GO TO 4608
      DO 4604 I=1,NP
      IF (NSTEP.LT.NSTEPJ) GO TO 4602
      IF (IGNORE(I).EQ.I) GO TO 4601
      IF (NDQSET.EQ.1) GO TO 4606
      IF (DQQMRA(I).LE.EPSETHI(I).AND.DQQMRA(I).GE.
     * EPSETLO(I)) GO TO 4603
      HPP(I) = HP*(EPSET(I)/DQQMRA(I))
      NPLACE(I) = NPLACE(I) + 1
      NPLACE1 = 1
      GO TO 4607
4606  CONTINUE
      IF (QMXRATA(I).LE.EPSETHI(I).AND.QMXRATA(I).GE.
     * EPSETLO(I)) GO TO 4603
      HPP(I) = HP*(EPSET(I)/QMXRATA(I))
      NPLACE(I) = NPLACE(I) + 1
      NPLACE1 = 1
4607  CONTINUE
      GO TO 4604
4601  HPP(I) = DTLIM
      GO TO 4604
4602  CONTINUE
      HPP(I) = HP
      NPLACE(I) = 0
      GO TO 4604
4603  CONTINUE
      NPLACE(I) = 0
4604  CONTINUE
4608  CONTINUE
      IF (NPLACE1.EQ.0) GO TO 4680
      NSTEPD = 4
      NCON = 1
      GO TO 4690
4610  CONTINUE
      IF (DQMAXX.GT.EPNHI) GO TO 4620
      NCON=1
      NSTEPD=1
      IF (DQMAXX.LE.EPNLO) HPPO=HRATIO*HP
      IF (DQMAXX.LE.EPNLO) NSTEPD = 2
      GO TO 4690
4620  CONTINUE
      HPPO=OSHPR*HP
      NSTTST = 1
      NSTEPD = 3
      GO TO 4690
4640  CONTINUE
```

```
      IF (NGPTST.EQ.1) GO TO 4690
      NSTTST = 1
      NSTEPD = 5
      IF (IRK.EQ.0) EPN=TSTITR*QMX1ST
      IF (NFLAG1.NE.1) HPPO = EPMULT*HP
      IF (NFLAG1.EQ.1) HPPO = ZPT5*HP
      IF (NSMAX.NE.-2.AND.NSTEP.GT.NSTEPS/2) DTLIM = HPPO
      IF (DTLIM.LT.DTLIMS) DTLIM = DTLIMS
      NCNCNT = 0
      IF (NFLAG1.NE.1) DTLIMX = DTLIM
      HPO    =HPPO
      HP     =HPO
      NCON   =0
      NSTEPJ = NSTEP + 3
      NITTOT=NITTOT-NITER
      NITSAV=0
      NTTCNT=0
      IF (HRATIO.GT.ONEP) HRATIO = ONEP+ZPT9*(HRATIO-ONEP)
      IF (HRATIO.GT.ONEP5) HRATIO=ONEP5
      IF (HRATIO.LT.ONEP05) HRATIO=ONEP05
      WRITE (IOUT,9630)CDNALN,HP,NSTEP,NITER,NRESET,DQMAXX,EPNRAT,QMX1ST
      DQMAXX=MAX(0.1*DQMAXX,EPSITR)
      IF (ISCR.EQ.ITRM)
     *WRITE (    *,9630)CDNALN,HP,NSTEP,NITER,NGPTST,DQMAXX,EPNRAT,QMX1ST
      DO 4670 NPP=1,KVARBF
      NP = IZ(IVARBF+NPP)
      DELTAQ(NP) = DQMAXX
 4670 CONTINUE
      GO TO 4690
 4680 CONTINUE
      NCON = 1
      NSTEPD = 0
 4690 CONTINUE
      IF (NVGJ.NE.NGROUP.OR.NFLAG1.NE.1) GO TO 4810
      HPPO = 1000000.0
      DO 4800 I=1,NP
      IF (HPP(I).LT.HPPO) HPPO = HPP(I)
 4800 CONTINUE
 4810 CONTINUE
      IF (NITER.GT.1.OR.NVGJ.NE.NGROUP) GO TO 4850
      DO 4820 I=1,NP
      IF (NPLACE(I).LT.NPLACEF) GO TO 4820
      EPSET(I) = DQQMRA(I)
      EPSETHI(I) = EPSETHIF*EPSET(I)
      EPSETLO(I) = EPSETLOF*EPSET(I)
      NEPSET = 1
      NPLACE(I) = 0
 4820 CONTINUE
 4850 CONTINUE
      IF (NITER.EQ.1.AND.DQMAXX.LT.EPSITR) GO TO 6810
      IF (NITERS.GT.1.AND.NITER.NE.2) GO TO 6900
      IF (IRK.EQ.1.OR.NSTTST.NE.1.OR.HP.NE.DTLIM.OR.DTLIM.EQ.ZP0001)
     * GO TO 6900
 6810 CONTINUE
 6900 CONTINUE
C:
C: END CONVN
C:
      IF (HPPO.GT.DTLIMX) HPPO = DTLIMX
      IF (HPPO.LT.DTMINX) HPPO = DTMINX
C:
      IF (NFLAG1.EQ.1) GO TO 8760
      IF (NSMAX.GE.0) GO TO 8800
      IF (NTMOD.NE.0.AND.NCON.EQ.1.AND.NRATE.NE.NGRSET) GO TO 8800
```

```
 8760 CONTINUE
C:       IF (NITER.GT.2.OR.NITERS.EQ.1.OR.NVGJ.NE.NGROUP) GO TO 8800
      IF (NITER.GT.2.OR.NVGJ.NE.NGROUP) GO TO 8800
C:
      ISOT = IOUT
      NGRSPR=NGRSUM+KVARBF
      IF (NGRSPR.LE.5) GO TO 8780
      NGRSPT=NGRSPR+1
      DO 8770 I=1,NGRSPR
      IF (NGRSPR.EQ.1) GO TO 8780
c     write (*,*) ' ngrspr ',i,ngrspr,deltaq(ngrspt-i)
      IF (DELTAQ(NGRSPT-I).NE.0.0) GO TO 8780
      NGRSPR=NGRSPR-1
 8770 CONTINUE
 8780 CONTINUE
      IF (NITER.LE.1) WRITE (ISOT,9610) VGMAX,NSTEP,NITCNT,
     * (VDG(I),I=1,NGRSPR)
      IF(NITER.GT.1)WRITE(ISOT,9610)  QRATCN,NSTEP,NITCNT,
     * (RATCON(I),I=1,NGRSPR)
      WRITE (ISOT,9610) DQMN,NSTEP,NITCNT,(DQMNDQ(I),I=1,NGRSPR)
      WRITE (ISOT,9610) DQMA,NSTEP,NITCNT,( DQMAL(I),I=1,NGRSPR)
      WRITE (ISOT,9610) QMXR,NSTEP,NITCNT,(QMXRAT(I),I=1,NGRSPR)
      IF (NITER.GT.1)
     * WRITE (ISOT,9610) DELO,NSTEP,NITCNT,(DELOLD(I),I=1,NGRSPR)
      WRITE (ISOT,9610) DELN,NSTEP,NITCNT,(DELTAQ(I),I=1,NGRSPR)
      WRITE (ISOT,9610) DQMR1,NSTEP,NITCNT,(DQQMRA(I),I=1,NGRSPR)
      IF (NFLAG2.EQ.1) GO TO 4785
      IF (NITER.LE.1) WRITE (*,9610) VGMAX,NSTEP,NITCNT,
     * (VDG(I),I=1,NGRSPR)
      IF(NITER.GT.1)WRITE(*,9610)  QRATCN,NSTEP,NITCNT,
     * (RATCON(I),I=1,NGRSPR)
      WRITE (*,9610) DQMN,NSTEP,NITCNT,(DQMNDQ(I),I=1,NGRSPR)
      WRITE (*,9610) DQMA,NSTEP,NITCNT,( DQMAL(I),I=1,NGRSPR)
      WRITE (*,9610) QMXR,NSTEP,NITCNT,(QMXRAT(I),I=1,NGRSPR)
      IF (NITER.GT.1)
     * WRITE (*,9610) DELO,NSTEP,NITCNT,(DELOLD(I),I=1,NGRSPR)
      WRITE (*,9610) DELN,NSTEP,NITCNT,(DELTAQ(I),I=1,NGRSPR)
 4785 CONTINUE
      IF (NITER.LE.1)
     * WRITE (*,9610) DQMR1,NSTEP,NITCNT,(DQQMRA(I),I=1,NGRSPR)
      IF (NITER.LE.1)
     * WRITE (*,9610) STEV1,NSTEP,NITCNT,(V1O(I),I=1,NGRSPR)
      IF (NITER.LE.1)
     * WRITE (*,9610) STEV2,NSTEP,NITCNT,(V1N(I),I=1,NGRSPR)
      IF (NITER.LE.1)
     * WRITE (*,9610) STEV3,NSTEP,NITCNT,(STEVIND(I),I=1,NGRSPR)
      IF (NITER.LE.1.AND.NFLAG1.EQ.1)
     * WRITE (*,9610) STEV9,NSTEP,NITCNT,(HPP(I),I=1,NGRSPR)
      IF (NITER.LE.1.AND.(NSTEP.EQ.NSTEPJ-1.OR.NEPSET.EQ.1)
     * .AND.NFLAG1.EQ.1)
     * WRITE (*,9610) EPSE,NSTEP,NITCNT,(EPSET(I),I=1,NGRSPR)
      IF (NITER.LE.1.AND.(NSTEP.EQ.NSTEPJ-1.OR.NEPSET.EQ.1)
     * .AND.NFLAG1.EQ.1)
     * WRITE (*,9610) EPSH,NSTEP,NITCNT,(EPSETHI(I),I=1,NGRSPR)
      IF (NITER.LE.1.AND.(NSTEP.EQ.NSTEPJ-1.OR.NEPSET.EQ.1)
     * .AND.NFLAG1.EQ.1)
     * WRITE (*,9610) EPSL,NSTEP,NITCNT,(EPSETLO(I),I=1,NGRSPR)
      IF (NITER.LE.1)
     * WRITE (*,9614) NSTEP,NITCNT,HPPO,HP,NSTEPD,NVGJ
      IF (NFLAG2.EQ.1) GO TO 8800
      WRITE (*,9612) NSTEP,NITCNT,EPS,EPSITR
      WRITE (*,9613) NSTEP,NITCNT,RATTST,EPN,QMX1ST,DQMAXX
      WRITE (*,9616) NSTEP,NITCNT,NSMAX,NSTTST,NGPTST,NGRSET,
     * NGRSPR,NGRSPT
```

```
      WRITE (*,9617) NSTEP,NITCNT,NSTEPD,NSTEPJ,NRATBS,NCON,
     * NRESET,IRK
      WRITE (*,9618) NSTEP,NITCNT,NCNCNT,NGROUP,NRATE,IWCHDQ,
     * NTMOD,NVGJ
 8800 CONTINUE
      IF (NVGJ.EQ.1.AND.RATEMN.EQ.ONEP.AND.NSTEP.GT.5.AND.IFLITR.EQ.0)
     * TFINAL=XP
      IF (HPO.LT.SMALL) WRITE (*,9670) SMALL,HP,XP,TFINAL
      IF (HPO.LT.SMALL) TFINAL = XP
      IF (HPO.LT.SMALL) STOP 'BAD RUN'
      NITER = NITSAV
C:
C: END CONVN
C:
      IF (NIPRNT.EQ.0.OR.NITER.NE.0) GO TO 8950
      NKVSET = NGRSET*NNODE
      DO 8940 I=1,NKVSET
 8940 RZ(IQNOWB+I)=RZ(IQLAST+I)
      NGO = 2
      NGOSS = 2
      IF (DQMAXX.LE.EPSITR) NGO = 1
      IF (DQMAXX.LE.EPSITR) NGOSS = 1
 8950 CONTINUE
      IF (NCON.NE.1) GO TO 9000
      IF (XP+HPPO.GT.TFINAL.AND.NITER.GE.NITERS) HPPO =ONEP1*(TFINAL-XP)
      IF (HPPO.LE.ZERO) HPPO = ZP0001
 9000 CONTINUE
      IF (HP.NE.DTLIMY.AND.NITCNT.EQ.NITERS.AND.NVGJ.EQ.NGROUP) HPO=HPPO
      IF (NITERS.GT.01.AND.DQMAXX.LT.EPSITR.AND.NVGJ.EQ.NGROUP) HPO=HPPO
      IF (NVGJ.NE.1.OR.NITER.GT.1.OR.NSMAX.EQ.-2) GO TO 9050
      IF (NSMAX.GT.0) GO TO 9050
      IF (HP.EQ.DTLIM) NCNCNT = NCNCNT+1
      IF (NCNCNT.LT.10) GO TO 9050
      IF (NTTCNT.GE.NRATBS) GO TO 9050
      NTTCNT = NTTCNT+1
      IF (NFLAG1.EQ.1) GO TO 9005
      DTLIM = 2.0*DTLIM
      DTLIMX = DTLIM
 9005 CONTINUE
      WRITE (*,9680) NSTEP,NCNCNT,NTTCNT,DTLIM,EPSITR
      NCNCNT = 0
      IF (HRATIO.GT.ONEP1) HRATIO = ONEP1
 9050 CONTINUE
      IF (EPN.GT.EPNSAV) EPN=EPNSAV
      if (hp.lt.zero.or.xp.gt.tfinal)
     * WRITE (*,9671) 4,NCON,XP,HP,HPO,HPPO
      IF (NITERS.LE.10.OR.NITER.LT.NITERS-3) RETURN
      IF (NITER.EQ.NITERS.AND.DQMAXX.GT.EPSITR)
     1 HPO=ZPT25*HP
      TENEPS = TEN*EPSITR
      IF (NITER.EQ.NITERS-3.AND.DQMAXX.GT.TENEPS)
     1 HPO=ZPT5*HP
      HPPO=HPO
      RETURN
      END
```

Appendix B

Test Problem Descriptions

Two benchmark problems were used to test the time step control method (see Chapter 6): the driven cavity and the thermal cavity. A description of these problems is provided herein for convenience.

B.1 Driven Cavity

The driven cavity problem considers the shear-driven recirculation created by a lid moving at uniform velocity $u = 1$ ($v = 0$) along the top of a unit square cavity, as shown in Fig. B–1. The velocity boundary condition is no-slip on the three stationary walls and on the lid. All walls are adiabatic.

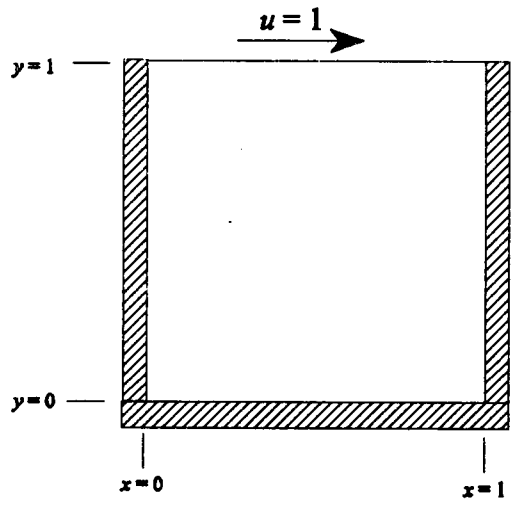


Fig. B–1. Driven Cavity

The test cases in Chapter 6 considered very low Reynolds numbers. Thus, a uniform 16×16 element mesh was sufficient for solution accuracy. Problems with higher Reynolds numbers exhibit singularities in the upper corners; thus, finer discretization may be required in the corners.

B.2 Thermal Cavity

The thermal cavity problem considers the bouyancy-driven flow created by a temperature difference between two walls, as shown in Fig. B–2. The side walls have Dirichlet boundary conditions with the left wall at $T = T_{hot}$ and the right wall at $T = T_{cold}$. The upper and lower walls are adiabatic. The velocity boundary condition is no-slip on all four walls.

Cases were run using 16×16 elements with both uniform and nonuniform discretizations. As the Rayleigh number is increased, finer discretization near the walls is necessary for solution accuracy.

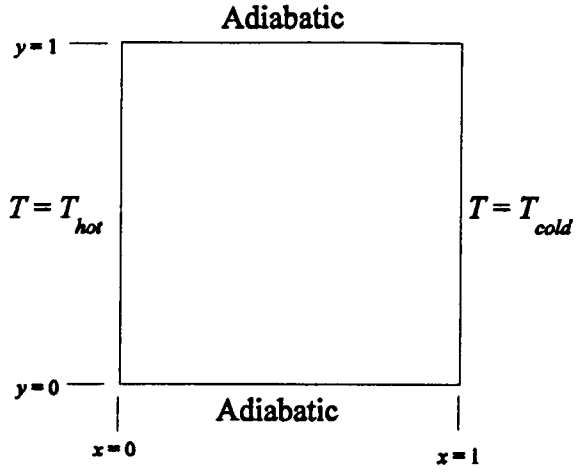


Fig. B–2. Thermal Cavity

Vita

Lamar Lepard was born in Aiken, South Carolina on February 9, 1964 the son of Bobby L., Sr., and Betty B. Lepard. He attended schools in the Aiken County Public School System where he graduated from Aiken High School Magna Cum Laude in May 1982. He served as Student Body President during his senior year and as Junior Class President, and participated as goalkeeper for the varsity soccer team and as first-chair trumpet for the varsity band.

In September 1982, he entered the Georgia Institute of Technology in Atlanta, Georgia and received a Bachelor of Nuclear Engineering degree with honors in June 1987. During his stay at Georgia Tech, he participated in the Cooperative Education Program, working 7 quarters with the Georgia Power Company at the Vogtle Electric Generating Plant. He also served as a resident assistant at the Woodruff Residence Hall for 3 quarters.

He began full-time employment in June 1987 with the Tennessee Valley Authority as a thermal/hydraulics analyst in the Nuclear Technology Branch in Knoxville, Tennessee and as a safety analyst at the Sequoia Nuclear Plant in Soddy-Daisy, Tennessee. In

December 1988, he accepted a position with Martin Marietta Energy Systems, Inc., at the Oak Ridge National Laboratory as a reactor safety analyst in the Research Reactors Division.

After a short period of time, he entered the Master's program as an evening student in Engineering Science and Mechanics with emphasis in Computational Fluid Mechanics at the University of Tennessee, Knoxville, in June 1989. The master's degree was received in August 1995.

He is presently working at the Oak Ridge National Laboratory, now operated by Lockheed Martin Energy Systems, Inc., as the Technical Assistant to the Director of Reactor Operations.

He is married to his high school sweetheart, the former Martha Cullum Ergle of Aiken, South Carolina. The Lepard family resides in Knoxville, Tennessee.