

To the Graduate Council:

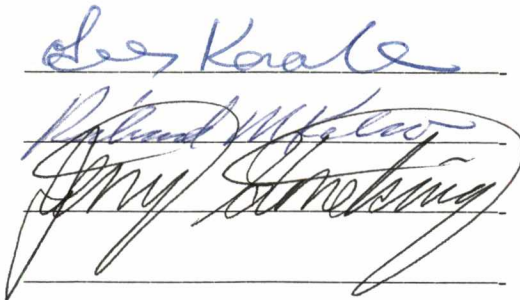
I am submitting herewith a dissertation written by Subrata Roy entitled "On Improved Methods for Monotone CFD Solution Accuracy". I have examined the final copy of this dissertation for form and content and recommend that it be accepted for the degree of Doctor of Philosophy, with a major in Engineering Science.



A. J. Baker, Ph.D

Major Professor

We have read this dissertation
and recommend its acceptance:



Accepted for the Council:



Associate Vice Chancellor

and Dean of The Graduate School

**On Improved Methods for Monotone CFD
Solution Accuracy**

A Dissertation

Presented for the

Doctor of Philosophy

Subrata Roy

December 1994

Copyright © Subrata Roy, 1994

All rights reserved

DEDICATION

This dissertation is dedicated to my parents

Mr. Nabaprakash Roy

and

Mrs. Dipali Roy

who have given me invaluable encouragement and blessings.

If I have not seen as far as others, it is because giants were standing on my shoulders.

--- Hal Abelson

If I have seen farther than others, it is because I was standing on the shoulder of giants.

--- Issac Newton

The eternal mystery of the world is its comprehensibility.

--- Albert Einstein

Classical thinking is not devoid of dispersion error.

Three Laws of Computational Mechanics

1. The computer solution is wrong! How wrong must be your premier determination.
2. You get out what you put in - especially on the boundaries.
3. There is no free lunch - the price of an adequate grid is inescapable.

--- Allen J. Baker

ACKNOWLEDGEMENTS

Research nowadays is a co-operative rather than an individual effort. I would like to express my special appreciation to my major professor, Dr. Allen J. Baker for being my friend, philosopher and guide for the past six years. Dr. Baker's guidance and support has helped fuel enthusiasm in my work and his invaluable suggestions has helped my dissertation become a reality. I must acknowledge the other committee members, Profs. Ohannes Karakasian, Richard M. Kelso and Jerry E. Stoneking, for their genuine help and concern during the period of dissertation writing.

I would also like to thank my colleagues, Drs. Joe Iannelli and Paul T. Williams and Messrs. Michael Barton, David Chaffin, Winston Holmes, Paul Manhart, Joe Orzakowski, Ed Schaub, Chris Wilson, Dee Wong, Kwai Wong and Jing Zhang, for their consistent support as a colleague and willingness to provide assistance during numerous intellectual and computational crises. I like to thank Ann Lacava of the Graduate School for her patience and cooperation.

My parents in India gave me continuous inspiration and blessings over the telephone and in the mail from thousands of miles away and I must say that worked. I appreciate Vivia for her endless love, devotion, understanding and encouragement that inspired me through the final years of my graduate school.

I also acknowledge the support and collaboration of Dr. Sukritranjan Bhattacharya and Dr. Baker for making my "coming to America" successful. Without their help I still might have been working as a pump "copier" in Calcutta. I express my sincere gratitude for financial assistance provided by the University of

Tennessee, Knoxville, Graduate School, National Science Foundation (NSF) MSS-9015912 and American Society of Heating, Refrigeration and Air-conditioning Engineers (ASHRAE) RP-464 to support completion of my Doctor of Philosophy degree.

ABSTRACT

The critical issues of improved solution accuracy, stability, monotonicity and efficiency for computational fluid dynamics (CFD) problems is the subject. The goal is to work towards a generalized algorithm that extracts the “best” solution out of an affordable “*optimal*” mesh. By optimal, we mean utilizing the *least number* of degrees of freedom possible for achieving $\|u - u_{h,p}\| \leq \varepsilon$, where ε is a small positive real number.

Computational fluid dynamics (CFD) solutions are especially prone to phase dispersion error yielding moderate to high distortion and ultimately instability. Specifically, the universally intrinsic zero phase velocity for “ $2\Delta x$ ” information induces an error cascading to smaller wave number modes that appears as mesh scale solution oscillations. Artificial viscosity and/or flux correction operators are designed to diffuse this error mode, but may produce unacceptable amounts of artificial dissipation especially in multidimensional embodiments. Alternatively, fluxvector splitting operators are not sufficient to prevent oscillations near strict local extrema. Introduction of non-linear correction factors called *limiters* may attain monotone solution at the expense of a very complicated procedure in multi-dimensions, and even then for a moderately coarse mesh the solution remains nodally inexact.

In this dissertation, three algorithms are derived to address the subject as follows:

a) A totally new, non-linear sub-grid embedding (SGM) weak statement finite element algorithm is derived, containing a local extremization step for which the theory predicts attainment of nodally accurate monotone solutions on arbitrary meshes. Further, concomitant accurate boundary fluxes can be established, with a minimal degrees of freedom solution, for linear and nonlinear parabolic benchmark problems and a Navier-Stokes problem.

b) The discrete approximate solution error, in general, can be expressed as a truncation of a Taylor series expansion. A Galerkin weak statement matrix perturbation (GMP) theory is derived yielding tridiagonal forms (in one-dimension) that can reduce, or annihilate in special cases, the Taylor series identified truncation error to high order. The theoretical analysis is via a von Neumann frequency analysis, and corresponding verification and benchmark solutions are documented in one and two dimensions. In concert, using an element-specific (local) Courant number, a *continuum* (total) time integration procedure is derived that can directly produce an accurate temporal solution *independent* of mesh measure and time step.

c) A discrete solution reconstruction (RCN) procedure is derived and validated for accuracy enhancement of dispersed (poor quality) solutions produced by “any” CFD algorithm. This algorithm theory is based on the preservation of l_p norms during solution evolution, and can yield an absolutely dispersion-free reconstruction of any transient or steady-state CFD solution.

TABLE OF CONTENTS

1	INTRODUCTION	1
1.1	General Preview :	1
1.2	Recent Advancements :	5
1.3	Present Approach	9
2	THE PROBLEM STATEMENT	12
2.1	Conservation Laws (CL)	12
2.2	The Continuum Navier-Stokes Equations	14
2.3	The Vorticity-Streamfunction Formulation	18
2.4	Continuity Constraint Method (CCM)	21
3	THE WEAK STATEMENT	26
3.1	Completion to a Newton algorithm	28
3.2	Spatial and temporal semi-discretizations	31
3.3	Finite element discrete weak statement algorithm	31

4	THEORETICAL ANALYSES	34
4.1	General Notions :	34
4.2	Asymptotic Convergence :	38
4.3	von Neumann Stability	43
5	SUB-GRID EMBEDDING, ORDER ADAPTIVE SEMI-DISCRETIZATION	47
5.1	Standard Finite Element	47
5.2	p -version Finite Elements	49
5.3	A Definition :	52
5.4	An Alternative Idea	53
5.5	Eigenvalue characterization	55
5.5.1	Element <i>Diffusion</i> or <i>Stiffness</i> matrix, $[D]_e$	56
5.5.2	Element <i>Interpolation (Mass)</i> matrix, $[M]_e$	58
5.5.3	Element <i>Convection</i> matrix, $[U]_e$	60
5.6	Local Embedding	61
5.7	Stability and Monotonicity for Steady State	68
5.8	Monotone Time Acceleration	72

5.9	Monotonicity and Stability for Transient Problems	74
5.10	Multidimensional SGM form	77
6	THE GALERKIN MATRIX PERTURBATION (GMP) ALGORITHM . .	80
6.1	Matrix perturbation terms	81
6.2	Matching the coefficients of analytical and algorithm amplification factors	82
6.3	Pertinent Examples	84
6.4	Comparison with TWS Class of Algorithm	88
6.5	Condition for Time Accurate Solution	92
6.6	Direct Time Integration	93
6.7	Multidimensional forms	95
6.7.1	Tensor product factorization algorithm:	95
6.7.2	The time-splitting (TS) algorithm:	96
7	THE RECONSTRUCTION (RCN) ALGORITHM	97
7.1	Linear RCN	100
7.2	Non-linear RCN	101
7.3	RCN Fortran Sample	102

8	RESULTS AND DISCUSSION	108
8.1	Synopsis	108
8.2	Verifications	110
8.2.1	Axisymmetric Steady State Conduction:	110
8.2.2	1d Unsteady Parabolic Problem:	116
8.2.3	Steady conduction with source:	120
	Case I : Both ends Dirichlet	120
	Case II : Left end Dirichlet, right end homogeneous Neumann	121
8.2.4	Linear Poisson equation on L-shaped two-dimensional domain: . . .	126
8.2.5	1d Steady Singular Perturbation Problem ($u \neq 0, \epsilon \rightarrow 0$) . . .	135
8.2.6	Parabolic perturbation verification in two dimensions:	151
8.2.7	1d Linear Stationary Wave ($u \neq 0, \epsilon \rightarrow 0$)	154
8.2.8	1d non-linear Burgers equation, stationary wave ($u \neq 0, \epsilon \rightarrow 0$)	157
8.2.9	Unsteady, $d = 1$ smooth and non-smooth wave convection, $C \leq 1173$	
8.2.10	Two-dimensional convection via a solid body rotation	182
8.3	The reconstruction (RCN) algorithm solutions:	188

8.3.1	Reconstruction for piecewise non-smooth and smooth wave propagation, $d = 1$	188
8.3.2	Steady advection-diffusion problem, $d = 1$	193
8.3.3	Solid body rotation, unsteady convection in $d = 2$ dimensions	196
8.4	Navier-Stokes Problems	201
8.4.1	Shear-Driven Cavity Flow	201
8.4.2	Close-coupled step wall diffuser	235
8.4.3	Airborne contaminant transport in a practical $d = 3$ room air flow field	264
9	CONCLUSIONS AND RECOMMENDATIONS	275
	Bibliography	278
	APPENDICES.	293
A		294
B		298
C		307
D		309

E	314
F	324
G	330
H	341
I	345
J	359
K	362
L	364
M	367
VITA.	369

LIST OF TABLES

8.1	Solution and Surface Flux Errors	114
8.2	Effect of ϕ on a uniform 9-node mesh	120
8.3	Solution comparison for various p on a 4 node mesh	122
8.4	1D Singular Perturbation Solution and Surface Flux Errors for $Re=50$, $c = 2.7$	139
8.5	Comparison element matrices $[D + U]_e$, $Re = 1000$, standard FE and SGM f_2	139
8.6	1d Singular perturbation solution comparison for $Re = 10^3$ and Nn - ode=13	140
8.7	Companion of boundary fluxes	143
8.8	1d Singular perturbation solution and surface flux errors, $Re=1000$.	145
8.9	Solution output data for 1D singular perturbation problem with $u =$ $1 - x$	148

8.10	Example element matrices for standard GWS and SGM f_2 computation	159
8.11	1d Stationary wave solution comparison for $Re = 10^6$ and $Nnode=10$	159
8.12	Element matrices for Burgers equation algorithms, 11 node mesh, $Re = 10^5$	163
8.13	Solution output for Burgers equation	167

LIST OF FIGURES

1.1	Traveling square wave on 101 node mesh, from Hirsch (1992, 2: 566)	2
4.1	Steady State solution for Burgers equation as function of $Re = 1/\epsilon$	36
5.1	Standard two dimensional elements	48
5.2	Combinations of h-p in a five degrees of freedom 1d element, Rachowicz, <i>et. al</i> (1989).	51
5.3	Increasing polynomials in a 2d p -element Zienkiewicz and Craig (1984).	51
5.4	Comparison of basis functions for standard, hierarchical and SGM elements in one dimension.	64
5.5	Two dimensional hierarchical p -element	65
5.6	Two dimensional locally extremized p -element	66
5.7	Two dimensional SGM $p = 1$ element	67
8.1	Convergence study, error for $d = 1$ steady conduction problem. . . .	115
8.2	Solution monotonicity and effect of ϕ for the transient evolution. . .	118

8.3	Effect of increasing p on solution energy in Case I for constant source $s = 1$	123
8.4	Effect of increasing p on solution energy in Case I for variable source $s = \text{radius}$	124
8.5	Effect of increasing p on solution energy in Case II.	125
8.6	Linear Poisson problem on L-shaped domain, 75 element (96 node) mesh, FE and SGM WS^h solutions.	128
8.7	Comparison among coarse to dense mesh linear basis and coarse mesh globally p -embedded solutions for the L-shaped region benchmark. . .	132
8.8	Convergence and error in energy norm for various FE algorithms vs. nodal degrees of freedom.	134
8.9	Parabolic perturbation steady state solution on a 13-node mesh, $\text{Re} =$ 1000, standard GWS and SGM (quadratic f_2) algorithm solutions. . .	136
8.10	Boundary fluxes from WS^h , parabolic perturbation ($\text{Re} = 1000$) steady state solution.	144
8.11	Verification of optimal SGM parabolic parameter c for different Re with various mesh size.	146

8.12 Comparison of GWS^h , TWS^h and SGM algorithm for $Re = 10^5$, U $= (1 - x)$ parabolic perturbation solution in one dimension.	149
8.13 2-D Parabolic perturbation on an $Nnode = 13^2$ uniform mesh, $Re = 20$.	152
8.14 2-D Parabolic perturbation on an $Nnode = 9^2$ uniform mesh, $Re = 2000$.	155
8.15 SGM $f_2(\mathbf{x})$ solution of a two dimensional parabolic perturbation (Re $= 20,000$) on an $Nnode = 9^2$ uniform mesh for $c_x = c_y = 124$	156
8.16 Linear stationary wave solution schematic.	156
8.17 1d Linear square wave steady solution, 10-node mesh, $Re = 10^6$	158
8.18 Nonlinear Burgers equation solution domain schematic	163
8.19 Monotonicity study for non-linear Burgers solution, $Re = 100,000$. . .	164
8.20 Gaussian wave propagation, two wavelengths at $C = 1$	174
8.21 Full sine wave propagation, two wavelengths at $C = 1$	175
8.22 Full sine wave propagation, two wavelengths at $C = 0.5$	177
8.23 20-node wide square wave propagation, three wavelengths at $C = 1$. .	178
8.24 20-node wide square wave propagation, three wavelengths at $C = 0.5$.	179
8.25 2-node wide square wave propagation, ten wavelengths at $C = 1.0$. .	181

8.26	2-node wide square wave propagation, ten wavelengths at $C = 0.5$. .	183
8.27	Rotating cone solutions, one revolution on a 33×33 mesh	184
8.28	Rotating cone GMP solution on a 17^2 mesh, after one revolution at $C_e = 1, \Delta t = \frac{2\pi}{16}$	189
8.29	1d 9-node wide square wave propagation, $C = 1.0$, GWS solutions, $t=90$ sec.	191
8.30	1d Sine wave propagation for $C = 1.0$, FE solutions at $t=20$ sec. . . .	192
8.31	Advection-diffusion ($Re = 1000$) steady state solutions on a 21-node mesh.	195
8.32	Rotating cone, initial condition and exact solution after one revolution.	198
8.33	Rotating cone verification CFD solution, $k = 1$ FE, $C = 1$, on 33^2 uniform mesh.	199
8.34	Rotating cone verification CNFD solution, $C = 1$, on 33^2 uniform mesh.	202
8.35	GWS ^h solution for driven cavity on a uniform Nnode $= 17 \times 17$ mesh, $Re = 100$	205
8.36	GWS ^h solution for driven cavity on a uniform Nnode $= 33 \times 33$ mesh, $Re = 100$	206

8.37 GWS ^h solution for driven cavity on a uniform Nnode = 65 × 65 mesh, Re = 100.	207
8.38 Convergence in the GWS ^h solution energy norm for streamfunction (PSI), vorticity (OMGA) and pressure (PRES), driven cavity, Re = 100.	208
8.39 GWS ^h solution energy computation for streamfunction and vorticity.	210
8.40 GWS ^h solution for driven cavity on a non-cartesian mesh, Nnode = 17 × 17, Re = 1000.	211
8.41 GWS ^h solution for driven cavity on a uniform mesh, Nnode = 33 × 33, Re = 1000.	212
8.42 TWS ^h solution for driven cavity on a non-cartesian mesh, Nnode = 17 × 17, $\beta = 0.1$, Re = 1000.	213
8.43 TWS ^h solution for driven cavity on a non-cartesian mesh, Nnode = 17 × 17, $\beta = 0.2$, Re = 1000.	214
8.44 SGM solution for driven cavity on a non-cartesian mesh, Nnode = 17 × 17, $c = 1.2$, Re = 1000.	216
8.45 Comparison GWS ^h and SGM streamfunction solution for driven cavity (Re = 10 ³) on 17 ² node mesh.	218

8.46 Comparison GWS ^h and SGM vorticity solution for driven cavity (Re = 10 ³) on 17 ² node mesh.	219
8.47 GWS ^h and SGM ^h solution comparison for driven cavity (Re = 2000) on different grids.	220
8.48 GWS ^h vorticity solution for driven cavity (Re = 3200) on a M = 16 ² (289 node) mesh.	221
8.49 SGM ^h vorticity solution for driven cavity (Re = 3200) on a M = 16 ² (289 node) mesh.	222
8.50 GWS ^h vorticity solution for driven cavity (Re = 3200) on a M = 32 ² (1089 node) mesh.	224
8.51 GWS ^h and SGM ^h vorticity solution comparison for driven cavity (Re = 3200).	225
8.52 GWS ^h streamfunction solution for driven cavity (Re = 3200) on a M = 16 ² mesh.	226
8.53 GWS ^h streamfunction solution for driven cavity (Re = 3200) on a M = 32 ² (1089 node) mesh.	227
8.54 SGM ^h streamfunction solution for driven cavity (Re = 3200) on a M = 16 ² (289 node) mesh.	228

8.55 SGM ^h streamfunction and velocity solution details for driven cavity, Re = 3200, rsgm _i ≥ 1.	229
8.56 GWS ^h and SGM ^h comparison pressure solutions for driven cavity (Re = 3200).	231
8.57 SGM ^h pressure solution for driven cavity (Re = 3200) on a M = 16 ² (289 node) mesh.	232
8.58 Nodal distribution of SGM ^h parameter RSGM for driven cavity (Re = 3200) on a 289 node mesh.	233
8.59 GWS ^h solution energy norm for driven cavity (Re = 3200) on a M = 16 ² (289 node) mesh.	234
8.60 GWS ^h solution energy norm for driven cavity (Re = 3200) on a M = 32 ² (1089 node) mesh.	236
8.61 SGM ^h solution energy norm for driven cavity (Re = 3200) on a M = 16 ² (289 node) mesh.	237
8.62 Flow field details for backward facing step, Armaly <i>et al.</i> (1983). . . .	239
8.63 Computational domain details for the 2-D backward facing step. . . .	241
8.64 GWS ^h solutions for 2-D backward facing step (100 ≤ Re ≤ 800) on a non-uniform 7 × 18 + 48 × 35 node mesh.	242

8.65	GWS ^h pressure solutions for 2-D backward facing step ($100 \leq Re \leq 800$) on a $7 \times 18 + 48 \times 35$ node non-uniform mesh.	244
8.66	Comparison pressure solutions for 2-D backward facing step at $Re = 800$	245
8.67	Comparison pressure solutions through the geometric centerline ($y = 0$) for backward facing step, $Re = 800$	247
8.68	Comparison V velocity solutions for 2-D backward facing step at $Re = 800$	248
8.69	Comparison 2-D backward facing step solutions for backward facing step, reattachment length vs. Reynolds number.	249
8.70	Comparison velocity vectors for 2-D backward facing step ($Re = 800$) on a non-uniform $7 \times 18 + 48 \times 35$ node mesh.	251
8.71	Comparison V velocity solutions through the geometric centerline ($y = 0$) for backward facing step, $Re = 800$	252
8.72	Comparison vorticity solutions along the geometric centerline ($y = 0$) for backward facing step, $Re = 800$	254
8.73	Comparison streamfunction solutions along the geometric centerline ($y = 0$) for backward facing step, $Re = 800$	255

8.74	GWS ^h planar velocity details for backward facing step, Re = 800. . .	256
8.75	p-WS ^h planar velocity details for backward facing step, Re = 800 . . .	256
8.76	Comparison U velocity solution perspective plots for Re = 800. . . .	258
8.77	Comparison V velocity solution perspective plots for Re = 800. . . .	260
8.78	Comparison pressure solution perspective plots for Re = 800.	261
8.79	Comparison pressure solution expanded perspective plots for Re = 800.	263
8.80	SGM ^h solution perspective details for 2-D backward facing step, Re = 800.	265
8.81	3-D room FE CFD velocity and temperature solution distribution, Williams (1993), a) velocity vectors, b) temperature.	267
8.82	Inert species concentration, a) FE CFD solution, Sc ^t = 1.0, β _t = 0.0, b) RCN solution from a).	268
8.83	Inert species concentration, a) FE CFD solution, Sc ^t = 1.0, β _t = 0.9, b) RCN solution from a).	271
8.84	Inert species concentration, a) FE CFD solution, Sc ^t = 0.1, β _t = 0.0, b) RCN solution from a).	272

8.85 Velocity vector reconstruction, a) FE CFD solution from Williams (1993), b) RCN solution from a).	274
J.1 Characteristic behaviour of monotone and non-monotone numerical solutions.	359

INTRODUCTION

1.1 General Preview :

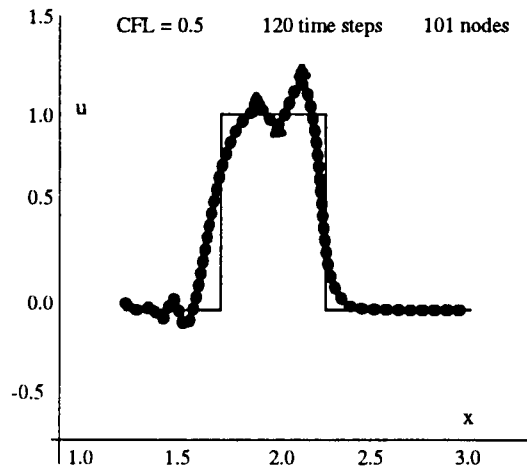
Consider the hyperbolic conservation law form

$$\mathcal{L}(q) = \frac{\partial q}{\partial t} + \frac{\partial \mathbf{f}}{\partial \mathbf{x}} - s = 0, \quad \text{on } \Omega \times t \subset \mathbb{R}^d \times \mathbb{R}^+ \quad (1.1)$$

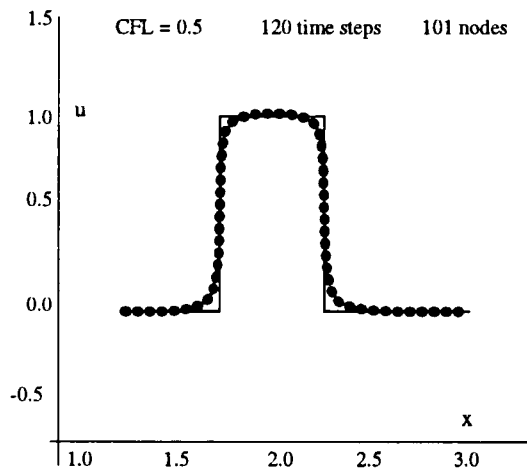
In (1.1), q is the state variable, $\mathbf{f} = \mathbf{u}q$, is the kinetic flux vector with convection velocity vector $\mathbf{u}$, and s is the source term. The boundary of the d -dimensional ($1 \leq d \leq 3$) problem domain Ω is the *union* of segments $\partial\Omega_i$, i.e., $\partial\Omega = \partial\Omega_1 \cup \partial\Omega_2 \cup \partial\Omega_3$.

Approximate (CFD) algorithms to this non-diffusive transient problem class are characterized by a phase lag dispersion error that can highly distort an approximation solution, *c.f.* Figure 1.1a). Specifically, the “ $2\Delta x$ wave” zero phase velocity induces a short wave-length error that cascades to smaller wave numbers. The theoretically esthetic remedies are to increase the density of degrees of freedom, *e.g.* number of mesh nodes, or algorithm order-of-accuracy. Conversely, the “practical” remedy is use of an artificial dissipation method, usually involving an added elliptic perturbation to (1.1) with a (linear or non-linear) numerical viscosity coefficient. A variation involves flux correction operators, to remove induced excessive dissipation via an antidiffusive contribution. In all instances, the goal is to induce a damping mechanism for control of

non-monotonicity, *i.e.*, oscillations, but solutions tend to suffer from excess diffusion, Figure 1.1b), even with a very dense mesh. Detailed comparisons of this phenomenon for the 1-D and 2-D linear convection verification problems are available, *c.f.*, Baker (1983, Ch. 4.13), Hirsch (1992, Ch. 17.3 and Ch. 21.3).



a) MacCormack finite difference (FD) scheme solution



b) Second-order FD total variation diminishing scheme, artificial dissipation

Figure 1.1 Traveling square wave on 101 node mesh, from Hirsch (1992, 2: 566)

The Fourier representation of the analytical solution to (1.1) is

$$q(\mathbf{x}, t) = \sum_{\zeta=-\infty}^{\infty} A_{\zeta} e^{i(\underline{\omega} \bullet \mathbf{x} - \alpha t)} \quad (1.2)$$

where $\omega_{\zeta} = 2\pi/\lambda_{\zeta}$ is the wavenumber of ζ th wave, λ_{ζ} is the wavelength of ζ th wave, $i = \sqrt{-1}$ and A_{ζ} is the expansion coefficient. For a linearity assumption for (1.1), one may focus on the generic ζ th mode only and consequently for (1.2), $\alpha = \mathbf{u} \bullet \underline{\omega}$. Hence, the exact solution $q(\mathbf{x}, t)$ finite Fourier generic mode $q_{\zeta}(\mathbf{x}, t)$ is

$$q_a(\mathbf{x}, t) = e^{i\underline{\omega} \bullet (\mathbf{x} - \mathbf{u}t)} \quad (1.3)$$

Assuming a semi-discrete solution $q^h(j\Delta\mathbf{x}, t)$ behaves in the same manner,

$$q^h(j\Delta\mathbf{x}, t) = e^{i\underline{\omega} \bullet (j\Delta\mathbf{x} - \mathbf{u}^h t)}, \text{ where } \mathbf{u}^h = \mathbf{u}_{\Re}^h + i\mathbf{u}_{\Im}^h \quad (1.4)$$

In (1.4) and hereafter, the superscript h denotes a spatial semi-discretization, and the subscripts $\Re$ and $\Im$ denote real and imaginary solution components respectively. Substituting $q_a(j\Delta\mathbf{x}, t) = e^{i\underline{\omega} \bullet (j\Delta\mathbf{x} - \mathbf{u}t)}$ in (1.4) yields

$$q^h(j\Delta\mathbf{x}, t) = q_a(j\Delta\mathbf{x}, t) e^{\underline{\omega} \bullet (\mathbf{u}_{\Im}^h + i(\mathbf{u} - \mathbf{u}_{\Re}^h))t} \quad (1.5)$$

The corresponding semi-discretization error, e^h , is

$$e^h = q_a - q^h = 1 - e^{\underline{\omega} \bullet (\mathbf{u}_{\Im}^h + i(\mathbf{u} - \mathbf{u}_{\Re}^h))t} \quad (1.6)$$

Thus the phase dispersion error of the approximation solution is related to $i(\mathbf{u} - \mathbf{u}_{\Re}^h)$ and the dissipation error is related to $\mathbf{u}_{\Im}^h$.

For the fully discrete solution, after elapsed time $n\Delta t$, the fully discrete Fourier component becomes,

$$Q_j^n(j\Delta\mathbf{x}, t + n\Delta t) \equiv e^{i\underline{\omega} \bullet (j\Delta\mathbf{x} - \mathbf{u}^h(t + n\Delta t))} \quad (1.7)$$

$$\begin{aligned}
&= e^{-i\omega \bullet \mathbf{u}^h n \Delta t} e^{i\omega \bullet (j\Delta \mathbf{x} - \mathbf{u}^h t)} \\
&= G^n q^h(j\Delta \mathbf{x}, t)
\end{aligned}$$

The solution for algorithm amplification factor G determines how the fully discrete solution grows or decays in time, and

$$G^n = e^{-i\omega \bullet (\mathbf{u}_{\mathfrak{R}}^h + i\mathbf{u}_{\mathfrak{S}}^h) n \Delta t} = G_{\mathfrak{R}} + iG_{\mathfrak{S}} \quad (1.8)$$

By definition, G^n obeys the following identity

$$G^n e^{i\omega \bullet \mathbf{u}^h n \Delta t} = 1 \quad (1.9)$$

Comparing to the analytical solution (1.3), one may determine that in one dimension

$$G_a^n = \frac{q(j\delta x, t + n\delta t)}{q(j\delta x, t)} = \frac{e^{i\omega(j\delta x - U(t + n\delta t))}}{e^{i\omega(j\delta x - Ut)}} = e^{-i\omega U(n\delta t)} \quad (1.10)$$

where δ denotes a virtual increment (as opposed to real increment Δ) and U is the magnitude of $\mathbf{u}$ in one dimension. For $n = 1$ and a virtual time increment δt ,

$$\begin{aligned}
G_a &= e^{-i\omega U \delta t} = e^{-i(\omega \delta x) \frac{U \delta t}{\delta x}} = e^{-imC} = \sum_{l=0}^{\infty} a_l m^l \\
&= 1 - imC - \frac{(mC)^2}{2} + i\frac{(mC)^3}{3!} + \frac{(mC)^4}{4!} - i\frac{(mC)^5}{5!} - \frac{(mC)^6}{6!} + i\frac{(mC)^7}{7!} + \dots
\end{aligned} \quad (1.11)$$

where $m = \omega \delta x$ is the virtual non-dimensional wavenumber with $C = U \delta t / \delta x$ the corresponding Courant number. In the last line of (1.11), G is expanded in a Laurent series on m with $a_l = f(C)$ coefficients.

The discrete approximation solution amplification factor G is (of course) algorithm specific, and may be computed via a Fourier modal analysis of the CFD discrete algorithm recursion relation. The order of accuracy of the numerical solution, hence algorithm stability, is characterized by G . As stated, the discrete algorithm “ $2\Delta x$ ” zero phase velocity induces a short wave-length error that cascades to smaller wave numbers. Utilization of artificial viscosity, via an additional dissipative flux vector term in (1.1), or “upwinding” the kinetic fluxvector, can diffuse this error mode, but typically with an unacceptable amount of purely numerical dissipation, c.f., Kreiss and Lorenz (1989), Lax (1954). An improved theory for phase-selective dissipation is the Taylor Weak Statement (TWS), Baker and Kim (1987), where the dependent variable is first expanded in a temporal Taylor series, and (1.1) is selectively substituted for higher order accuracy. The TWS theory encompasses many previous artificial dissipation methods, c.f., Baker and Kim (1987), Chaffin and Baker (1994).

1.2 Recent Advancements :

“Intelligent” algorithms for deriving discrete improved-accuracy solution methodologies in an *automatic* manner have been examined in “adaptive” forms constructed using finite element methods (FEM). Common procedures on adaptive refinements, c.f., Babuška, *et. al* (1981), Jensen (1992), Johnson and Hansbo (1992), Oden, *et. al* (1989), Zhu and Lin (1989), Zienkiewicz, *et. al* (1982), are: (a) the “ h -method,” which retains the same polynomial degree for the trial function while globally reducing mesh size successively to yield an “accurate” solution, *i.e.* using asymptotic

convergence; (b) the “ p -method,” which increases the degree of the trial function polynomial, usually locally on the same nodal mesh of measure h , to improve the asymptotic convergence rate, and (c) the “ r -method,” which retains fixed polynomial degree while redistributing a fixed number of nodes to reduce error, *i.e.*, “mesh optimization.” Here and afterwards, solution “*convergence*” means meeting a predesigned stopping criterion, *e.g.*, permissible extremum error distribution on the domain. Combinations like h - p , r - p and r - h are of course options and have been extensively examined, Babuška, *et. al* (1990), Bank, *et. al* (1983), Demkowicz, *et. al* (1992), Oden (1992), Rachowicz, *et. al* (1989).

Historically, McNeice and Mereal (1971) presented a mathematical theory for an optimal mesh for solving a self-adjoint partial differential equation system, where the discretization nodal distribution was considered an unknown. Melosh and Kellian (1976) analyzed the change in potential energy caused by introduction of an extra degree of freedom. Peano, *et. al* (1977-1978) proposed a similar approach based on a hierarchical (p - method) finite element approximation. In 1975, Babuška produced a satisfactory *a posteriori* error estimator for a one dimensional parabolic problem, which also extended to general higher dimensional elliptic boundary value problems.

In a recent study, Nguyen (1992) selected uniform refinement of p levels for a model elliptic boundary value problem. For a p -degree basis element, the order of the one dimensional element matrix is $((p + 1)^2 \times (p + 1)^2)$, while for an m -noded, h -version serendipity element the order is $m \times m$. Hence, the size of the matrix for p -elements increases rapidly as the degree (p) of approximation polynomial increases.

The element-order square and column matrices were computed by Gauss-Legendre numerical quadrature with $(p + 1)$ sampling points. Thus, the computational effort also increases directly with higher levels of p . To reduce these, Nyugen suggested, (a) a maximum level $p_{max} = 4$ be used, with more elements (than theoretically required) added to compensate for lost degrees of freedom for achieving a predesigned accuracy level, alternatively (b) *FE* mesh node numbering to minimize the front width, and (c) the lower p -level solution be used to initiate the next higher p -degree calculation.

One advantage of p -element methodology is that solutions on one refinement can be directly used for the next refinement level as an initial condition without extrapolation. For the h -method, this is generally not possible. Oden (1992) advocates that the h - p finite element method adopts the advantages from both p and h methods. Tamma and Saw (1989) suggest the process of p -adaptive refinement on an edge-by-edge basis, and propose enrichment of a local nature using numerical integration associated with mapping techniques. Tamma and Saw also chose Gauss quadrature for element integrals based on the maximum degree of the polynomials present within each element.

Accuracy and error estimates have been established for the linear model problems. For a linear elliptic boundary value problem, Jensen and Suri (1992) found the theoretical error bound to be $\mathcal{O}(p^{-2\alpha-3/2})$, where $\alpha(= \mathcal{O}(\pi/\omega))$ is not an integer, and in a polar coordinate system $\alpha = \min_i\{\alpha_i\} = \pi/\max_i\{w_i\}$, $\alpha > 1/2$ and w_i ($0 < w_i < 2\pi$, $w_i \neq \pi$). In another study, Zienkiewicz and Zhu (1992) determined superconvergence of the p -adaptive algorithm for linear elliptic problems. They used

a patch recovery technique that produced an $\mathcal{O}(h^{p+\alpha})$ convergence rate, for $\alpha = 1$ for odd p , and $\alpha = 2$ for even p , for elliptic boundary value problems.

Despite recent progress, implementing mesh refinement/solution adaptation procedures remains a tedious process, and is proven computationally expensive and time consuming. At any stage of refinement, how best to distribute the degrees of freedom over the solution domain, combining mesh refinement and/or increasing the order of local basis functions in order to minimize the error in some chosen norm, remains illusive, especially for eqn. (1.1).

A solution adaptive form of *FEM* typically utilizes *a posteriori* error estimation to guide the adaptive *h-p* refinement procedure, Babuška, *et. al* (1992), Oden, *et. al* (1989), Targowski, *et. al* (1992), Zienkiewicz and Craig (1984). The preferred choice for p enrichment is Legendre polynomials. For an exclusive p -method, the local polynomial degrees of freedom are enriched systematically, keeping the mesh (h) fixed, in order to achieve the “best” accurate solution on the given domain discretization. As compared to other adaptive methods, p -hierarchical elements have certain advantages, namely:

- (1) a potentially faster asymptotic convergence rate,
- (2) ability to conform mixed order interpolations,
- (3) an accurate initialization for an iterative solution procedure,
- (4) improved algebraic equation conditioning (as compared to the standard non-hierarchical elements),
- (5) facility to utilize previous solution vectors and matrices, and

(6) an *a posteriori* error analysis and adaptive polynomial refinement technique.

The disadvantages for h -, p - and h - p methods include,

- (1) significantly increased computer memory and storage space requirements,
- (2) higher computational resource demands for both local and global matrix operations,
- (3) exponential increase in numerical quadrature loop length for progressively higher levels of p , and hence
- (4) extensive logic requirements for involved problems, *e.g.*, (1.1).

1.3 Present Approach

In this dissertation, I present three methods for solution accuracy improvements to equation systems of the form (1.1). The first is an h - p finite element sub-grid enrichment algorithm for obtaining solution of linear elliptic, and linear and nonlinear parabolic problem statements characteristic of CFD requirements. The notion of a non-linear subgrid embedding (SGM), in concert with a local extremization, can produce nodally exact, monotone solutions for the entire range of verification problem statements on very coarse meshes. Lagrange polynomials are used to form basis functions for the p -element enrichment choice. Since the hierarchical family of p -elements degenerates under linear local extremization, *i.e.* reduces to the lower degree unperturbed form, a non-linear embedding is mandatory. The resultant non-linear embedded SGM Galerkin weak statement algorithm is verified absolutely stable, for a non-linear model problem of (1.1), when the standard Galerkin algorithm

diverges. For a particular selection of the subgrid embedding scale, the developed theory predicts the algorithm can always produce a monotone solution for all model problems considered. Of key importance, the SGM procedure utilizes the minimal storage and computing facility associated with a strictly linear basis algorithm.

Another alternative approach developed herein to improving algorithm accuracy is via perturbing the linear basis matrix statement of a finite element discretized Galerkin weak statement (GWS). The theory assumes that an arbitrary order of the Taylor series truncation error can be written in the form of linear basis rank FE matrices, and hence the resultant high-order accurate Galerkin linear algebra statement remains tridiagonal. This Galerkin matrix perturbation (GMP) method can be theoretically compared with a wide class of algorithms on a term by term basis. For an arbitrary n -th order accurate FE solution, the required perturbed matrices may be derived via matching the coefficients of the Laurent series expansions of the discrete algorithm amplification factor with that of the analytical solution. The elements of the perturbation matrices are represented as a power series, that enables one to incorporate an extended assembly stencil in a simple tridiagonal form. Hence, the GMP algorithm can be directly compared with a family of compact finite difference schemes, c.f., Lele (1992), although the later is far more complicated to implement. A select choice of the perturbation matrices also enable one to achieve a third or fourth-order accurate TWS stencil, as originally developed by Baker and Kim (1987).

Finally, a radically alternative approach has been developed, documented herein as an approximate solution *reconstruction* (RCN) algorithm for the low diffusive level

CFD problem class. The established RCN solution is always genuinely monotone, and is achieved as a post-processing operation which annihilates the discrete algorithm dispersion waves via enforcement of the requirement that the Sobolev l_p vector norms remain unperturbed. The RCN procedure can produce a monotone, dispersion-free solution reconstruction from *any* CFD algorithm solution. The reconstruction of the approximate solution does not, however, correct the overall transient phase lag inherent in the base CFD solution. The limiting assumption is that the sought monotone solution is predictable, *i.e.* no genuine solution randomness is present to confuse the theoretical reconstruction process.

For a scalar linear pure convection problem without source, the analytical solution is always exact preservation of the initial condition distribution, as it is transported throughout the domain. This allows (theoretical) relocation/translation of the initial distribution, in exact form, to the location where the CFD approximate solution l_∞ norm occurs. Adding a source to the problem definition destroys this elementary scenerio, however the developed RCN procedure is equally applicable, as is derived herein.

Numerical solutions are presented confirming theoretical expectations for specific algorithm performance for one- and two- dimensional verification and benchmark problems akin to Navier-Stokes systems, with both smooth and non-smooth solutions and including nonlinearity. The extension to select genuine Navier-Stokes problems and a three dimensional problem complete the envelope of verification/benchmark predictions.

THE PROBLEM STATEMENT

2.1 Conservation Laws (CL)

Consider the conservation law statement form of the Navier-Stokes equations

$$\mathcal{L}(q) = \frac{\partial q}{\partial t} + \frac{\partial}{\partial x_j}(f_j - f_j^v) - s = 0, \quad \text{on } \Omega \times t \subset \mathbb{R}^d \times \mathbb{R}^+[0, \infty) \quad (2.1)$$

In (2.1), f_j and f_j^v denote the kinetic and dissipative flux vectors respectively. In f_j , the product of q with the convection velocity u_j is common, while in f_j^v a diffusion coefficient ϵ typically exists that varies parametrically for problem type, and s is the source term. The boundary of the d -dimensional problem domain Ω is the *union* of segments $\partial\Omega_i$, i.e., $\partial\Omega = \partial\Omega_1 \cup \partial\Omega_2 \cup \partial\Omega_3$, completely encompassing Ω . For $\epsilon > 0$, well-posed boundary conditions are of the form

$$\begin{aligned} l(q) &= \epsilon \nabla q \cdot \hat{\mathbf{n}} + \alpha(q - q_r) = 0 & \text{on } \partial\Omega_1 \\ l(q) &= \epsilon \nabla q \cdot \hat{\mathbf{n}} + q_n = 0 & \text{on } \partial\Omega_2 \\ q(\mathbf{x}_b, t) &= q_b(\mathbf{x}_b, t) & \text{on } \partial\Omega_3 \end{aligned} \quad (2.2)$$

where $\hat{\mathbf{n}}$ is the outward pointing unit normal vector, α , q_r and q_n are given boundary flux data, both variable and fixed, and q_b is any fixed reference. The initial condition $q(x_i, t = 0) = q_0(x_i)$ closes system (2.1)-(2.2) as a well-posed statement.

As the parameter ϵ approaches zero, (2.1) exhibits the character of a singular perturbation problem. For Navier-Stokes systems, this amounts to the Reynolds number becoming large, as is the natural occurrence. Computational difficulty is anticipated since, for $\epsilon = 0$, the limit equation reduces to first order and all boundary conditions can no longer be enforced. The resultant incipient loss of a boundary condition typically leads to occurrence of a “boundary layer-type” solution. Thereby, even though the exact analytical solution is smooth, monotone and bounded, a spatially semi-discretization computational procedure usually leads to an oscillatory (*i.e.*, non-monotone) approximation that can eventually become unbounded in the presence of genuine Navier-Stokes non-linearity.

Artificial diffusion methods usually involve a perturbation to the dissipative flux vector via a linear or non-linear term containing a numerical viscosity coefficient. The associated theory seeks to induce a dissipation mechanism to numerically damp such non-monotonicity and/or instability of the semi-discrete scheme. In such a “parabolic” perturbation, the companion system, cast with an isotropic diffusion term multiplied by a single constant viscosity coefficient, replaces (2.1) with the form

$$\frac{\partial q}{\partial t} + \frac{\partial}{\partial x_j}(f_j - f_j^v) = \frac{\partial}{\partial x_j}(f_j^d) \quad (2.3)$$

where f_j and f_j^v remain the flux vectors as defined. Two possible forms for f_j^d , the parabolic-perturbation dissipation flux vector, *c.f.*, Kreiss and Lorenz (1989), Iannelli (1992), are

$$f_j^d \equiv h\beta_i \frac{\partial g_j(q)}{\partial x_i} \quad \text{or} \quad f_j^d \equiv h\tilde{\beta}_{ij} \frac{\partial q}{\partial x_i} \quad (2.4)$$

which intrinsically embed the mesh measure h , for dimensional consistency, and the viscosity coefficients β_i or $\tilde{\beta}_{ij}$ are theory-selectible. The formulation theoretically allows the artificial viscosity coefficient to vary cell wise, in concert with a discretization (h) procedure, to moderate the excess diffusion often associated with the elementary forms of (2.4).

Studies in this area, c.f., Lax (1954), have shown that for a general non-linear one-dimensional parabolic system

$$q_{,t} + f(q)_{,x} - s = \lambda q_{,xx} \quad (2.5)$$

the perturbed analytical solution, $q_\lambda(x, t)$, exists for a range of t , and converges towards a weak solution $q(x, t)$ of the corresponding hyperbolic system,

$$q_{,t} + f(q)_{,x} - s = 0 \quad (2.6)$$

as the artificial viscosity coefficient λ , in (2.5), approaches zero. In (2.5)-(2.6), subscripts "comma x " and "comma t " indicate the corresponding partial derivatives respectively.

2.2 The Continuum Navier-Stokes Equations

In the continuum, the general conservation law system describing continuity, momentum transport and energy conservation (including viscous dissipation as higher order) for laminar flow of an incompressible fluid is

Continuity :

$$\mathcal{L}(\rho_0) = \frac{\partial u_i}{\partial x_i} = 0 \quad \text{where } 1 \leq i \leq d \quad (2.7)$$

Momentum :

$$\mathcal{L}(u_i) = \frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j} \left(u_i u_j + P \delta_{ij} - \frac{1}{\text{Re}} \frac{\partial u_i}{\partial x_j} \right) - \frac{\text{Gr}}{\text{Re}^2} \Theta \hat{g}_i = 0 \quad (2.8)$$

Energy :

$$\mathcal{L}(\Theta) = \frac{\partial \Theta}{\partial t} + \frac{\partial}{\partial x_j} \left(u_j \Theta - \frac{1}{\text{Re Pr}} \frac{\partial \Theta}{\partial x_j} \right) - \frac{\text{Ec}}{\text{Re}} \varphi - s_\Theta = 0, \quad 1 \leq i, j \leq d \quad (2.9)$$

In (2.7)-(2.9), $u_i(x_j, t)$ is the velocity vector resolution, $P(x_j, t)$ is the hydrodynamic pressure (normalized by constant density ρ_0), $\Theta(x_j, t)$ is the potential temperature, δ_{ij} is the Kronecker delta, and the gravitational acceleration implicit in the Boussinesq approximation to thermal buoyancy has unit vector $\hat{g}_i$. The viscous dissipation function φ is defined as (Yih, 1969)

$$\varphi = \frac{\partial u_i}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (2.10)$$

The nondimensional groups in (2.7)-(2.9) are defined as follows:

$$\begin{aligned} \text{Reynolds number} \quad \text{Re} &= \frac{L_{ref} U_{ref}}{\nu_{ref}} \\ \text{Grashof number} \quad \text{Gr} &= \frac{g \beta L_{ref}^3 \Delta T}{\nu_{ref}^2} \end{aligned} \quad (2.11)$$

$$\text{Eckert number} \quad \text{Ec} = \frac{U_{ref}^2}{c_{ref} \Delta T} \quad (2.12)$$

$$\text{Prandtl number} \quad \text{Pr} = \frac{\nu_{ref} \rho_0 c_{ref}}{k_{ref}}$$

where L_{ref} and U_{ref} are select reference length and velocity scales respectively, ν_{ref} is the reference kinematic viscosity, β is the coefficient of thermal expansion, c_{ref} is the reference specific heat, k_{ref} is the diffusion coefficient and ΔT is the normalizing temperature difference.

For isothermal flow, the energy equation is eliminated and the buoyancy term in the momentum equation (2.8) vanishes. In forced convection, inertial and frictional effects can dominate buoyancy forces, hence the buoyancy term in (2.8) is a negligible influence. Under natural convection conditions, thermal buoyancy and inertial effects are of comparable magnitude, and (2.7)-(2.9) can be nondimensionalized independent of a Reynolds number as :

$$\text{Continuity : } \mathcal{L}(\rho_0) = \frac{\partial u_i}{\partial x_i} = 0 \quad (2.13)$$

$$\text{Momentum : } \mathcal{L}(u_i) = \frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j} \left(u_i u_j + P \delta_{ij} - \text{Pr} \frac{\partial u_i}{\partial x_j} \right) - \text{Ra Pr} \Theta \hat{g}_i = 0 \quad (2.14)$$

$$\text{Energy : } \mathcal{L}(\Theta) = \frac{\partial \Theta}{\partial t} + \frac{\partial}{\partial x_j} \left(u_j \Theta - \frac{1}{\text{Re Pr}} \frac{\partial \Theta}{\partial x_j} \right) = 0 \quad ; \text{ for } 1 \leq i, j \leq d \quad (2.15)$$

In (2.14), the Rayleigh number definition is $\text{Ra} = \text{Gr Pr}$, hence provides the measure of the buoyancy driving force. In all cases, the CFD problem is to find $u_i(x_j, t)$, $P(x_j, t)$, and $\Theta(x_j, t)$ satisfying (2.7)-(2.9) or (2.13)-(2.15).

The well-posed closure of the Navier-Stokes equation set requires specification of appropriate initial and boundary conditions on Ω and Γ respectively, c.f., Baker (1983). On bounding surfaces, the intrinsic requirement is

$$n_i u_i = u_n \quad (2.16)$$

and

$$s_i u_i = u_\tau \quad (2.17)$$

where $n_i = n(x_i)$ is the resolved unit outward normal vector, $s_i = s(x_i)$ is the resolved unit tangential vector aligned parallel to the boundary, u_n is the normal velocity com-

ponent (non-zero for a porous wall) and u_τ is the tangential wall velocity component. For the rigid no-slip wall, (2.16)-(2.17) reduces to $u_n = 0 = u_\tau$.

On segments of the boundary with inflow the inlet velocity distribution may be assumed known, hence

$$u_i(s, t) = f_i(s, t) \quad \text{on } s \subset \Gamma_2 \quad (2.18)$$

At outflow, the velocity distribution is generally not known *a priori*. One option is to assume the flow is sufficiently developed such that the normal efflux is axially non-varying, *i.e.*,

$$n_j u_{i,j} = 0 \quad \text{on } x_i \subset \Gamma_2 \quad (2.19)$$

The initial value problem definition for velocity is completed by the specification of the initial conditions:

$$u_i(x_i, t = 0) = u_i^0(x_i) \quad \text{in } \Omega \oplus \Gamma_i \quad (2.20)$$

and

$$u_{i,i}^0(x_i) = 0 \quad \text{in } \Omega \quad (2.21)$$

Similar considerations apply for the temperature. Defining the boundary as a non-overlapping sum of two segments Γ_D and Γ_F such that $\Gamma = \Gamma_D \cup \Gamma_F$ and $\Gamma_D \cap \Gamma_F = \emptyset$, at inflow, the temperature or heat flux field is known, hence

$$\Theta(x_i, t) = \Theta_0(x_i, t) \quad \text{on } x_i \subset \Gamma_D \quad (2.22)$$

or

$$\frac{\partial \Theta}{\partial x_j} n_j = F(x_i, t) \quad \text{on } x_i \subset \Gamma_F \quad (2.23)$$

The essential character of the Navier-Stokes equations is thus elliptic boundary value, to within the differential (solenoidal) constraint on the velocity field. Further, the absence of an algebraic equation of state or differential conservation law statement for pressure must be accommodated. Since no explicit pressure equation exists, boundary conditions for the pressure, if required, must be developed. Specifically, typically an outflow boundary specification is essential. For incompressible flows, pressure functions mathematically as a Lagrange multiplier, enforcing the incompressibility constraint by propagating at infinite wave-speed to maintain the velocity field in a continuous state of divergence-freeness.

2.3 The Vorticity-Streamfunction Formulation

The primitive variable equations (2.13)-(2.23) constitute a rigorous mathematical statement describing an incompressible thermal fluid flow. The fundamental conservation of mass required can be exactly enforced via vector field theory. Since the divergence of the curl of any vector function vanishes identically, the existence of a vector potential function Ψ is guaranteed, hence

$$\mathbf{u} = \nabla \times \Psi \quad (2.24)$$

is necessary and sufficient to automatically satisfy continuity (2.13). Substitution of the vector potential for the velocity in the momentum equation embeds the continuity condition but the pressure (gradient) remains present. Given that the curl of the gradient of a scalar is zero suggests taking the curl of the momentum equation to

eliminate the appearance of pressure. Hence, defining the vorticity vector

$$\underline{\Omega} = \nabla \times \mathbf{u} \quad (2.25)$$

the vorticity transport equation resulting from (2.14) is

$$\mathcal{L}(\underline{\Omega}) = \frac{\partial \underline{\Omega}}{\partial t} + \mathbf{u} \cdot \nabla \underline{\Omega} + \underline{\Omega} \cdot \nabla \mathbf{u} - \frac{1}{\text{Re}} \nabla^2 \underline{\Omega} - \frac{\text{Gr}}{\text{Re}^2} \nabla \times \Theta \hat{\mathbf{g}} = 0 \quad (2.26)$$

From (2.24) and (2.25), a compatibility relation exists between the streamfunction and vorticity vector fields in the form

$$\underline{\Omega} = -\nabla^2 \Psi + \nabla(\nabla \cdot \Psi) \quad (2.27)$$

Furthermore, since $\nabla \cdot \mathbf{u} = 0$, taking a curl on (2.25) a second compatibility condition relating velocity and vorticity is

$$\nabla^2 \mathbf{u} = -\nabla \times \underline{\Omega} \quad (2.28)$$

In two dimensions, the sole non-vanishing scalar components of the vector potential and vorticity lie perpendicular to the flow plane. Labelling these as streamfunction ψ and vorticity ω , then

$$\omega = \nabla \times \mathbf{u} \cdot \hat{\mathbf{k}} = \epsilon_{3jk} \frac{\partial u_k}{\partial x_j} = \frac{\partial u_2}{\partial x_1} - \frac{\partial u_1}{\partial x_2} \quad \text{in } \mathbb{R}^2 \quad (2.29)$$

where ϵ_{ijk} is the permutation tensor. Consequently, in two dimensions the conservation of momentum (2.14) is reduced to a scalar vorticity transport equation on $\underline{\Omega} \Rightarrow \omega \hat{\mathbf{k}}$, for $\mathbf{u} = \nabla \times \psi \hat{\mathbf{k}}$, as

$$\mathcal{L}(\omega) = \frac{\partial \omega}{\partial t} + \mathbf{u} \cdot \nabla \omega - \frac{1}{\text{Re}} \nabla^2 \omega - \frac{\text{Gr}}{\text{Re}^2} \nabla \times \Theta \hat{\mathbf{g}} \cdot \hat{\mathbf{k}} = 0 \quad (2.30)$$

$$\mathcal{L}(\psi) = -\nabla^2 \psi - \omega = 0 \quad (2.31)$$

and

$$\mathcal{L}(\mathbf{u}) = \nabla^2 \mathbf{u} - \nabla \times \omega \hat{\mathbf{k}} = 0 \quad (2.32)$$

Equation (2.30)-(2.32) constitute a well-posed NS problem statement. The fourth order biharmonic streamfunction approach, *c.f.*, Tuann and Olsen (1978), can replace the vorticity and velocity in (2.30) in terms of streamfunction only, via (2.31). It is of marginal practical interest, due to stringent requirements on the approximation subspace $\Psi^h \subset H_0^2$ (has to be at least ∇^2 complete), and also the corresponding interelement boundaries of the conforming elements must be C^1 continuous, *e.g.*, incomplete quintic polynomial basis functions for triangular elements. A review of both conforming and nonconforming element solution of the biharmonic streamfunction equation may be found in Gunzburger (1989).

The alternative velocity-vorticity formulation attention focuses on the vorticity transport equation (2.30) and the compatibility relation (2.28). In two dimensions, the velocity-vorticity relation is exactly (2.32), while in three dimensions the expanded form of (2.28) involves solving six equations,

$$\mathcal{L}(u_i) = \frac{\partial^2 u_i}{\partial x_j \partial x_j} + \epsilon_{ijk} \frac{\partial \omega_k}{\partial x_j} = 0 \quad \text{where } x_j \in \mathbb{R}^3, 1 \leq i, j \leq 3 \quad (2.33)$$

Vorticity boundary conditions stem from the definition (2.32) using the velocity boundary data. Agarwal (1981) calculated the flow field in a three dimensional lid-driven cavity for low Reynolds number ($\text{Re} \leq 400$) on a non-staggered grid. For two dimensional flow, Guj and Stella (1988) investigated the $\mathbf{u} - \underline{\Omega}$ method using the stag-

gered mesh ADI iteration technique for a lid-driven cavity and a step-wall diffuser. For a segregated variable scheme, Hafez and Soliman (1989) report a loss in stability and convergence behaviour due to the time lagged boundary condition evaluation. Farouk and Fusegi (1985) resolved this difficulty by employing a coupled line solver in two dimensions to obtain a stable algorithm for a wide range of flow parameters.

The inexact continuity enforcement or penalty method has also been developed, c.f., Zienkiewicz (1977), Temam (1968), Hughes, *et. al* (1979) and Baker (1983). In this method, the pressure term is replaced via a constrained minimization where the penalty term is a Lagrange multiplier approximation, *i.e.*, $p_{min}^{(\lambda)} = -\lambda \frac{\partial u_j^\lambda}{\partial x_j}$. Hence, as $\lambda \rightarrow \infty$, the solution to the momentum equation (2.14), “penalized” by a large penalty term, becomes implicitly compensated via driving velocity divergence $\frac{\partial u_j^\lambda}{\partial x_j} \rightarrow 0$.

2.4 Continuity Constraint Method (CCM)

The fundamental conclusions established in these formulations is the indispensable virtue of ensuring rigorous incompressibility (vector formulations) and the de-traction of a poorly conditioned penalty constraint algebraic statement. An alternative formulation, proposed by Noronha (1990) and implemented by Williams (1993), specifically resolves these issues; *i.e.*, yields construction of a well-conditioned constraint form with the unsteady flow solenoidal velocity attainment guaranteed as an intrinsic part of an iterative solution strategy.

This approach follows from an examination of the time semi-discrete conservative form of the momentum equations (neglecting buoyancy effects) over the time interval

$\Delta t = t_{n+1} - t_n$. The associated θ -implicit Taylor series is

$$\begin{aligned} u_i^{n+1} - u_i^n &= - \left[\theta \Delta t \frac{\partial}{\partial x_j} \left(u_i^{n+1} u_j^{n+1} + P^{n+1} \delta_{ij} - \frac{1}{\text{Re}} \frac{\partial u_i^{n+1}}{\partial x_j} \right) \right. \\ &\quad \left. + (1 - \theta) \Delta t \frac{\partial}{\partial x_j} \left(u_i^n u_j^n + P^n \delta_{ij} - \frac{1}{\text{Re}} \frac{\partial u_i^n}{\partial x_j} \right) \right] + \dots \end{aligned} \quad (2.34)$$

where P^{n+1} is the (unknown) pressure field corresponding to the divergence-free velocity field, *i.e.*,

$$\frac{\partial u_i^{n+1}}{\partial x_i} = 0 \quad (2.35)$$

In the absence of *a priori* knowledge of the pressure field P^{n+1} , an approximate distribution P^* must be identified. In general, the resulting associated velocity field $\tilde{u}_i^{n+1}$ fails to satisfy the continuity equation (22). The corresponding momentum balance is given by

$$\begin{aligned} \tilde{u}_i^{n+1} - u_i^n &= - \left[\theta \Delta t \frac{\partial}{\partial x_j} \left(\tilde{u}_i^{n+1} \tilde{u}_j^{n+1} + P^* \delta_{ij} - \frac{1}{\text{Re}} \frac{\partial \tilde{u}_i^{n+1}}{\partial x_j} \right) \right. \\ &\quad \left. + (1 - \theta) \Delta t \frac{\partial}{\partial x_j} \left(u_i^n u_j^n + P^n \delta_{ij} - \frac{1}{\text{Re}} \frac{\partial u_i^n}{\partial x_j} \right) \right] + \dots \end{aligned} \quad (2.36)$$

Subtracting (2.36) from (2.35) yields the error in the computed velocity field.

$$\begin{aligned} u_i^{n+1} - \tilde{u}_i^{n+1} &= -\theta \Delta t \frac{\partial}{\partial x_j} \left[\left(u_i^{n+1} u_j^{n+1} - \tilde{u}_i^{n+1} \tilde{u}_j^{n+1} \right) + \left(P^{n+1} - P^* \right) \delta_{ij} \right. \\ &\quad \left. - \frac{1}{\text{Re}} \frac{\partial}{\partial x_j} \left(u_i^{n+1} - \tilde{u}_i^{n+1} \right) \right] \end{aligned} \quad (2.37)$$

For $\theta = 0$, the curl of (2.37) yields the following identity,

$$\nabla \times \left(u_i^{n+1} - \tilde{u}_i^{n+1} \right) = 0 \quad (2.38)$$

which shows that the error in velocity for explicit time integration is irrotational. The vector relation (2.38) guarantees the existence of a scalar function Φ , the gradient of which is the precise measure of the error distribution in the approximate velocity field, *i.e.*,

$$\frac{\partial \Phi}{\partial x_i} = - (u_i^{n+1} - \tilde{u}_i^{n+1}) \quad (2.39)$$

The divergence of (2.39), subject to the incompressibility condition (2.35), yields a Poisson equation on scalar function Φ as

$$\mathcal{L}(\Phi) = \frac{\partial^2 \Phi}{\partial x_i^2} - \frac{\partial \tilde{u}_i^{n+1}}{\partial x_i} = 0 \quad (2.40)$$

Boundary conditions for the harmonic function follow naturally as the normal projection of (2.39) on to a boundary. For Dirichlet inflow and no-slip wall

$$\ell(\Phi) = \frac{\partial \Phi}{\partial x_i} n_i = (u_i^{n+1} - \tilde{u}_i^{n+1}) = 0 \quad (2.41)$$

and on the outflow boundary

$$\ell(\Phi) = \Phi = \text{constant} \quad (2.42)$$

Approximating the non-linear convective coefficients $u_i^{n+1} \approx \tilde{u}_i^{n+1}$ for the velocity error in (2.37) and substituting (2.39) yields a pressure correction relation for non-explicit time schemes in the form

$$\frac{\partial P^{n+1}}{\partial x_i} \approx \frac{\partial P^*}{\partial x_i} + \left(\frac{1}{\theta \Delta t} \right) \frac{\partial \Phi}{\partial x_i} + \frac{\partial}{\partial x_j} \left(\tilde{u}_j^{n+1} \frac{\partial \Phi}{\partial x_i} \right) \quad (2.43)$$

If the algorithm is developed from an explicit velocity field (but an implicit pressure), then the non-linear convective operator in (2.43) vanishes. In this case the pressure

correction is non-iterative and independent of the estimate P^* and is given by the identity

$$P^{n+1} - P^* \approx \left(\frac{1}{\theta \Delta t} \right) \Phi \quad (2.44)$$

For the implicit non-linear algorithm, an appropriate guess for the initial iteration is $P^* = P^n$. Subsequent approximations to the implicit pressure are then given by the summation of the iterative CCM function corrections as

$$P^{n+1} \rightarrow P^* \approx P^n + \left(\frac{1}{\theta \Delta t} \right) \sum_i \Phi^i \quad (2.45)$$

As the discrete divergence-free velocity field is attained the corresponding function Φ^i approaches zero in a suitable norm.

A broad comparison base exists between the CCM approach and several continuity-approximate algorithms documented in the literature, c.f., Williams (1993). The closest comparisons exist with the explicit algorithms of Amsden and Harlow (1970a,b), *i.e.*, the staggered grid SMAC scheme, and the equal order finite element algorithm of Ramaswamy and Kawahara (1987). In this context, the CCM algorithm is interpreted as an implicit generalization of the semi-discrete forms of these algorithms. The discrete weak statement algorithm employs equal order finite element approximations for all variables. In the fractional step method of Chorin (1968), an auxiliary velocity field is obtained from the implicit momentum equations written without the pressure gradient, *i.e.*, $P^* = 0$. Another comparison exists with the widely used class of pressure-correction schemes, namely the “SIMPLE” family of algorithms. The elementary SIMPLE algorithm by Patankar and Spalding (1972) was developed by

substituting the discrete pressure correction equation (2.44) into the discrete equation (2.40). In contrast with the CCM algorithm, SIMPLE does not enforce incompressibility directly.

Chapter 3

THE WEAK STATEMENT

Any approximate solution for (2.1) can be expressed in form

$$q(\mathbf{x}, t) \approx q^N(\mathbf{x}, t) = \sum_{j=1}^N \psi_j(\mathbf{x}) Q_j(t) \quad (3.1)$$

where $\psi_j(\mathbf{x})$, $1 \leq j \leq N$, is termed the “trial function” set. The superscript N in (3.1) and hereon denotes “approximation,” constituted as the sum of products of the assumed *known* function set $\psi_j(\mathbf{x})$ with the set of *unknown* coefficients $Q_j(t)$. The available choices for the trial functions are essentially limitless, *e.g.* trigonometric functions, Fourier series, Legendre polynomials, Chebyshev polynomials, Lagrange or Hermite interpolation polynomials, etc.

In (3.1), since q^N is an approximation, it cannot satisfy (2.1) identically. Hence, $\mathcal{L}(q^N) \neq 0$, and the corresponding approximation error (distribution) is

$$e^N = e^N(\mathbf{x}, t) = q - q^N \quad (3.2)$$

The ultimate design goal of an algorithm is to make this error absolutely minimum among the available options. Since e^N is a distribution, *i.e.*, $e^N = e^N(\mathbf{x}, t)$, then a constraint occurs when the integral

$$\int_{\Omega} w(\mathbf{x}, t) \mathcal{L}(q^N) d\tau \equiv 0, \quad \forall w(\mathbf{x}, t) \quad (3.3)$$

is satisfied for any “test function” $w(\mathbf{x}, t)$.

That (3.3) must hold for *any* test function can be made deterministic. Any given smooth function can be represented by its interpolation; hence, selecting another space of functions $\phi_i(\mathbf{x})$ eligible for supporting any such interpolation

$$w(\mathbf{x}, t) \approx w^M(\mathbf{x}, t) = \sum_{i=1}^M \phi_i(\mathbf{x}) W_i(t) \quad (3.4)$$

where superscript M denotes “interpolation”, and the time-dependence is again cast onto the *known* expansion coefficient set $W_i(t)$. This being the sole remaining arbitrariness, the *extremum* of (3.3) is deterministic, and defines the “*weak statement*”

$$\begin{aligned} \text{WS}^N &\equiv \frac{\partial}{\partial W_i} \int_{\Omega} w^M(\mathbf{x}, t) \mathcal{L}(q^N) d\tau, \quad \forall i \\ &= \int_{\Omega} \phi_i(\mathbf{x}) \mathcal{L}(q^N) d\tau = 0, \quad \text{for } 1 \leq i \leq M \approx N \\ &= \int_{\Omega} \phi_i \left(\frac{\partial q^N}{\partial t} - s \right) d\tau - \int_{\Omega} \frac{\partial \phi_i}{\partial x_j} (f_j - f_j^v)^N d\tau + \int_{\partial\Omega_\epsilon \cap \partial\Omega} \phi_i (f_j - f_j^v)^N \hat{n}_j d\sigma \\ &= \{0\} \end{aligned} \quad (3.5)$$

assuming the test function interpolation set ϕ_i is differentiable and Lebesgue-integrable.

Note then that no need exists to differentiate either flux vector, and the last expression exposes all appropriate boundary conditions of the well-posed problem statement (2.1)-(2.2). To within some boundary condition details, M in (3.3) is nominally equal to N in (3.1), such that (3.5) produces an ordinary differential equation (ODE) system of rank equal to the N unknowns $Q_j(t)$. For a finite element (FE) weak statement Galerkin construction of (3.5), the test and trial function sets con-

tain identical members, *i.e.*, $\phi_j(\mathbf{x}) \equiv \psi_j(\mathbf{x})$ Conversely, if one chooses $\phi_j(\mathbf{x})$ to be constants, then a “finite volume” (FV) weak statement is produced.

3.1 Completion to a Newton algorithm

The issue of stability and dispersion error control of semi-discrete approximation accrues to a companion to (2.1) established via a Taylor series time-semidiscrete analysis (“Taylor weak statement” (TWS) theory), Baker and Kim (1987). Retaining only the elementary term generated,

$$\mathcal{L}^c(q) \equiv \mathcal{L}(q) - \beta \Delta t \frac{\partial}{\partial x_j} \left(A_j A_k \frac{\partial q}{\partial x_k} \right) \quad (3.6)$$

is a representation “companion” replacement for (2.1). In (3.6), $A_j \equiv \partial f_j / \partial q$ is the kinematic flux vector jacobian, which is proven to directly contribute a tensor invariant, highly phase selective numerical dissipation mechanism, *c.f.*, Williams (1993) and Chaffin and Baker (1994).

Hence, the companion to GWS^N (3.3) may be written as

$$\text{TWS}^N = \int_{\Omega} \phi_i \mathcal{L}^c(q^N) d\Omega = 0 \quad (3.7)$$

which is satisfied for any (all!) given known functions $\phi_i(\mathbf{x})$ and yields a final form similar to (3.5).

In (3.5), one may approximate the time derivative via a multistep integration algorithm with an initial condition $q(t = 0) = q_0$. Herein, the newest value of $q(t_{n+k}) \equiv q_{n+k}$ is obtained from the previous K values of $q(t_{n+K-k}), k = 1, 2, \dots, K$

along with the present and previous values of the derivative $f'(t_{n+k}), k = 0, 1, \dots, K$.

$$q_{n+k} \equiv y(t_{n+k}) = \sum_{k=1}^K a^k q_{n+k-1} + \Delta t \sum_{k=0}^K b^k f(x_{n+k}) + \mathcal{O}(\Delta t^2 f'(\bar{t})) + \dots \quad (3.8)$$

For $t_n \leq \bar{t} \leq t_{n+1}$, (3.8) must be consistent with a Taylor series expansion on time t

$$q(t_1) = q(t_0) + \left. \frac{dq}{dx} \right|_{\bar{t}} \Delta t + \frac{1}{2!} \left. \frac{d^2 q}{dt^2} \right|_{\bar{t}} (\Delta t)^2 + \frac{1}{3!} \left. \frac{d^3 q}{dt^3} \right|_{\bar{t}} (\Delta t)^3 + \dots \quad (3.9)$$

The fundamental consistency requirements for (3.8) are

$$\sum_{k=1}^K a^k = 1, \quad \sum_{k=0}^K b^k = 1 \quad (3.10)$$

Several time-integration algorithms can be derived from (3.8)-(3.10) via a particular selection set of a^k and b^k . For CFD problems, a typical choice is one step ($K = 1$), *i.e.*, the Euler integration algorithm family (with $a^1 = 1$ and $0 \leq b^0, b^1 \leq 1$). The familiar form is

$$q_{n+1} = q_n + \Delta t \left[\theta \frac{dq_{n+1}}{dt} + (1 - \theta) \frac{dq_n}{dt} \right], \quad \text{for } 0 \leq \theta \leq 1 \quad (3.11)$$

Using (3.11) in (3.5), and approximating q as

$$q \approx \sum_i^N \phi_i(\mathbf{x}) Q_i(t), \text{ while } Q^n \equiv Q_i(t_n) \text{ and } Q^{n+1} \equiv Q_i(t_n + \Delta t) \quad (3.12)$$

global integrals produced in (3.5) take the forms

$$\mathbf{M} \equiv \int_{\Omega} \phi_j \phi_i d\Omega \quad (3.13)$$

$$\mathbf{U} \equiv \int_{\Omega} \phi_j \frac{\partial \phi_i}{\partial x} d\Omega$$

$$\mathbf{D} \equiv \int_{\Omega} \frac{\partial \phi_j}{\partial x} \frac{\partial \phi_i}{\partial x} d\Omega$$

$$\mathbf{B} \equiv \int_{\partial\Omega \cap \Omega} \phi_j \phi_i \hat{n} d\sigma$$

and the Galerkin (or Taylor) weak statement becomes the matrix ODE system

$$\text{GWS}^N \text{ or } \text{TWS}^N = \mathbf{M} \frac{dQ}{dt} + R = 0 \quad (3.14)$$

$$R = (\mathbf{U} + \mathbf{D} + \mathbf{B})Q(t) - s$$

The residual matrix R is a non-linear function of Q , and contains contribution from all terms except the time term, *i.e.*, convection, diffusion, source and boundary information. Substituting (3.14) into the θ -implicit, one-step time integration algorithm, yields the terminal matrix statement

$$F(Q) = \mathbf{M}(Q_{n+1} - Q_n) + \Delta t(\theta R_{n+1} + (1 - \theta)R_n) = 0 \quad (3.15)$$

where $t_{n+1} = t_n + \Delta t$.

The classic Newton algorithm may be used to express solution of equation system (3.13) as the iterative cycle,

$$Q_{n+1}^0 = Q_n ; F(Q_{n+1}^0) = F(Q_n) \quad (3.16)$$

do $p = 0, 1, 2, \dots$ until convergence

$$\left[\mathbf{M} + \theta \Delta t \frac{\partial R}{\partial Q} \right]_{n+1}^p \{\delta Q\}_{n+1}^{p+1} = -\{FQ\}_{n+1}^p$$

$$Q_{n+1}^{p+1} = Q_{n+1}^p + \delta Q_{n+1}^{p+1}$$

end do

Therefore, quasi-Newton approximations use simplified forms of the Jacobian

$$\left[\mathbf{M} + \theta \Delta t \frac{\partial R}{\partial Q} \right].$$

3.2 Spatial and temporal semi-discretizations

The spatial semi-discretization of the domain Ω introduces the computational mesh $\Omega^h = \bigcup_e \Omega_e$, where Ω_e denotes the generic computational (finite element) sub-domain. Using superscript “ h ” to denote use of a discretization, the form of (3.1) becomes

$$q^N(\mathbf{x}, t) \equiv q^h(\mathbf{x}, t) = \bigcup_e q_e(\mathbf{x}, t) \quad (3.17)$$

$$q_e(\mathbf{x}, t) = \{N_k(\boldsymbol{\eta})\}^T \{Q(t)\}_e \quad (3.18)$$

In (3.18), $\{N_k(\boldsymbol{\eta})\}$ is the finite element trial space *basis set* equivalent for ψ_j in (3.1), typically containing polynomials complete to k th degree and written in an intrinsic $(\boldsymbol{\eta})$ coordinate system. The (matrix) order of this trial space basis depends on the polynomial degree and the dimension d of Ω , hence Ω_e . Thereby, use of quadratic, cubic, etc. bases leads to rapid expansion of the basis matrix order, especially in multi-dimensions.

3.3 Finite element discrete weak statement algorithm

The *weak statement* WS^h associated with the semi-discrete approximation (3.5)-(3.18) is

$$\begin{aligned} WS^h &= \int_{\Omega^h} \phi_i \left(\frac{\partial q^h}{\partial t} + \frac{\partial}{\partial x_j} (f_j - f_j^v)^h - s \right) d\tau = 0, \quad \forall i \\ &= \int_{\Omega^h} \phi_i \left(\frac{\partial q^h}{\partial t} - s \right) d\tau - \int_{\Omega^h} \frac{\partial \phi_i}{\partial x_j} (f_j - f_j^v)^h d\tau \end{aligned} \quad (3.19)$$

$$+ \int_{\partial\Omega_e \cap \partial\Omega^h} \phi_i (f_j - f_j^v)^h \hat{n}_j d\sigma = \{0\}$$

Thereby in (3.19), the domain boundary surface integrals remain to permit exact enforcement of the surface normal flux boundary conditions. True interest in many engineering analyses lies in the associated boundary flux computation, *e.g.*, heat flux and drag in a turbulent flow, hence the identified surface integral is of significant practical use.

Independent of the dimension d of Ω , and for general variation of physical properties within the flux vectors, the semi-discrete finite element weak statement (3.19) always yields an ODE set. Introducing matrix notation, the form is, *c.f.*, (Baker and Pepper, 1991),

$$\text{GWS}^h \text{ or TWS}^h = [M]\{Q(t)\}' + \{R\} = \{0\} \quad (3.20)$$

$$[M] = S_e [M]_e$$

$$\{R\} = S_e (([U]_e + [D]_e + [B]_e)\{Q(t)\}_e - \{b(t)\}_e)$$

where $\{Q(t)\}'$ denotes $\frac{d\{Q\}}{dt}$. In the last two lines of (3.20), subscript e denotes element rank matrices, and S_e symbolizes the “assembly operation” carrying local matrix coefficients into the global arrays. On each finite element, $[D]_e$ contains the dissipative fluxvector contribution, $[M]_e$ is the “mass” matrix associated with interpolation, and $[U]_e$ is the kinetic fluxvector approximation matrix that carries the convection information on the domain. $[B]_e$ is the $d-1$ dimensional implimentation of the Robin boundary condition contribution and $\{b\}_e$ is the matrix of all known given data, *i.e.* s , q_r , q_n and q_b .

The first line in (3.20) uniformly represents the assembled global form of a finite element weak statement semi-discretization. Thereby, $\{Q\}' = -[M]^{-1}\{R\}$, for use in a time semi-discretization. Recalling the θ -implicit Euler time integration,

$$\{Q(t_{n+1})\} \equiv \{Q\}_{n+1} = \{Q\}_n + \Delta t \left[\theta \{Q\}'_{n+1} + (1 - \theta) \{Q\}'_n \right] + \mathcal{O}(\Delta t^{f(\theta)}) \quad (3.21)$$

and substituting (3.20) thereby yields the algebraic equation system associated with q^h as

$$\{F(Q)\} = [M]\{Q_{n+1} - Q_n\} + \Delta t(\theta\{R\}_{n+1} + (1 - \theta)\{R\}_n) = \{0\} \quad (3.22)$$

Here, subscript n again denotes time level, and the Newton station jacobian of (3.22) is

$$[JAC] = \frac{\partial\{FQ\}}{\partial\{Q\}} = [M] + \theta\Delta t \left(\frac{\partial\{R\}}{\partial\{Q\}} \right) \quad (3.23)$$

The corresponding Newton iteration statement for solution of (3.22) is

$$\begin{aligned} [JAC]\{\delta Q\}^{p+1} &= -\Delta t\{R\}_n, \text{ linear} \\ &= -\{F\}_{n+1}^p, \text{ non-linear} \end{aligned} \quad (3.24)$$

where for iteration index $p = 0, 1, 2, \dots$

$$\{Q\}_{n+1}^{p+1} \equiv \{Q\}_{n+1}^p + \{\delta Q\}^{p+1} = \{Q\}_n + \sum_{i=0}^p \{\delta Q\}^{i+1} \quad (3.25)$$

until convergence.

THEORETICAL ANALYSES

4.1 General Notions :

The general analytical solution to (1.1) or (2.1) in the continuum may be expressed as

$$q^a(x_j, t) = v^a(t)e^{i\omega_j x_j} \quad (4.1)$$

for $1 \leq j \leq d$, where d denotes the solution domain dimension, $\omega_j/2\pi$ is the Fourier frequency and $i = \sqrt{-1}$. For exposition, the $d = 1$ and $s = 0$, form of (2.1) is

$$\mathcal{L}(q) = \frac{\partial q}{\partial t} + \frac{\partial}{\partial x} \left(uq - \epsilon \frac{\partial q}{\partial x} \right) = 0, \quad \text{on } \Omega \times t \subset \mathbb{R}^{2+} = \mathbb{R}^1 \times \mathbb{R}^+ \quad (4.2)$$

where the *half space* $\mathbb{R}^{2+} = \mathbb{R}_x \times (0, \infty) = \{(x, t) : x \in \mathbb{R}_x \equiv \mathbb{R}^1; t > 0\}$. $\mathbb{R}^1$ is the one-dimensional euclidean space and corresponding dual pair $\mathbb{R}^+ = [0, \infty)$ is the set of real positive numbers (time). For $d = 1$, the form of (4.1) becomes $q^a(x, t) = v^a(t)e^{i\omega x}$, and substituting into (4.2) leads to the ODE

$$\frac{d}{dt}v^a(t) + (\omega^2\epsilon + i\omega u)v^a(t) = 0 \quad (4.3)$$

For the Fourier eigenvalue $\lambda^{\mathcal{F}} = \omega^2\epsilon + i\omega u$, the solution to (4.3) is

$$v^a(t) = e^{-\lambda^{\mathcal{F}}t} \quad (4.4)$$

which is stable for non-negative real part $\omega^2\epsilon$. The resulting continuum solution (4.1) is familiar as

$$q^a(x, t) = e^{-\omega^2\epsilon t} e^{i\omega(x - ut)} \quad (4.5)$$

and for $\epsilon = 0$, becomes

$$q^a(x, t) = e^{i\omega(x - ut)} \quad (4.6)$$

One specific problem of interest of the form (4.2) is for $u \equiv q$, which is the nonlinear viscous Burgers equation. At steady state, *i.e.*, $t \rightarrow \infty$, the one dimensional Burgers equation ($q = u$) is

$$\mathcal{L}(q) = q \frac{dq}{dx} - \frac{d}{dx} \left(\epsilon \frac{dq}{dx} \right) = 0 \quad (4.7)$$

or

$$\frac{d}{dx} \left(\frac{q^2}{2} - \epsilon \frac{dq}{dx} \right) = 0 \quad (4.8)$$

Integrating (4.8) with respect to x , once yields

$$\frac{dq}{a^2 - q^2} = \frac{dx}{2\epsilon} \quad (4.9)$$

which upon integrating again yields

$$\ln \left(\frac{a + q}{a - q} \right) = -\frac{x}{\epsilon} + b, \quad a, b \text{ integration constants} \quad (4.10)$$

Using the boundary conditions $q|_{x=-1} = 1$ and $q|_{x=1} = -1$ in (4.10), then

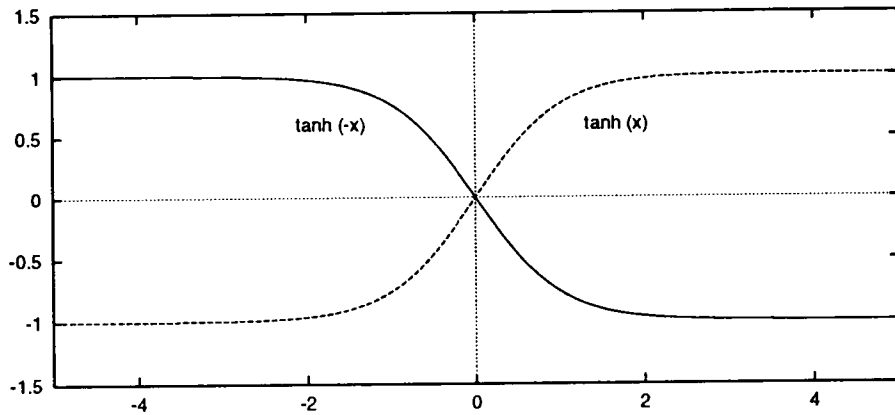
$$b = 0 \text{ and } a = \frac{1 + e^{-1/\epsilon}}{1 - e^{-1/\epsilon}} \rightarrow 1 \text{ as } \epsilon \rightarrow 0 \quad (4.11)$$

Hence, for $\epsilon \rightarrow 0$, the analytical solution (4.5) becomes

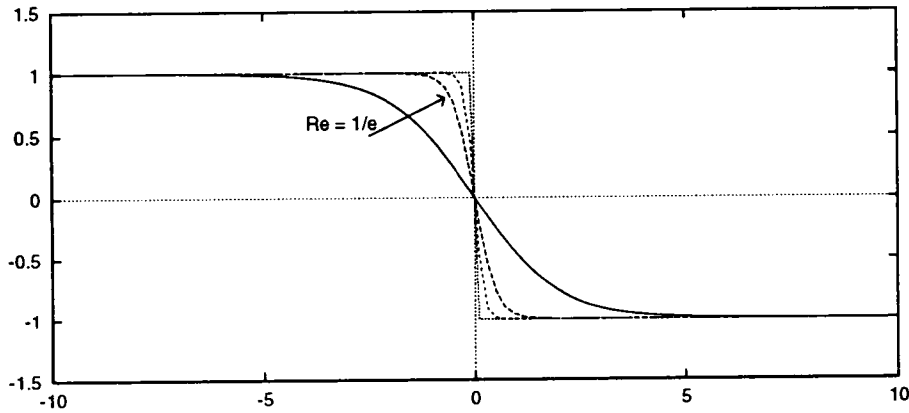
$$q = \tanh\left(\frac{-x}{2\epsilon}\right) + \text{constant} \quad (4.12)$$

where $\tanh$ is the hyperbolic tangent function and $\tanh^{-1}q = \frac{1}{2} \ln\left(\frac{1+q}{1-q}\right)$.

Figure 4.1 illustrates the solution for the steady-state Burgers equation (4.7). As $\text{Re} = 1/\epsilon \rightarrow \infty$, the slope of the solution near the origin ($x = 0$) approaches infinity, hence (4.12) resembles a shock phenomenon.



a) Hyperbolic function $\tanh$



b) Solution set for Burgers equation (4.7)

Figure 4.1 Steady State solution for Burgers equation as function of $\text{Re} = 1/\epsilon$

Equation (4.2) may also be expanded on a multidimensional domain, *e.g.* $\Omega \subset \mathbb{R}^3$, for a solution $q(\mathbf{x}, t)$ to

$$\mathcal{L}(q) = \frac{\partial q}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \left(\mathbf{u}q - \epsilon \frac{\partial q}{\partial \mathbf{x}} \right) = 0, \quad \text{on } \Omega \times t \subset \mathbb{R}^3 \times \mathbb{R}^+ \quad (4.13)$$

At steady state, assuming separation of variables is allowed, then

$q(x, y, z) = X(x)Y(y)Z(z)$ and (4.13) can be rewritten as

$$\begin{aligned} \frac{u}{X} \frac{dX}{dx} - \frac{1}{\text{Re } X} \frac{d^2 X}{dx^2} &= -\frac{v}{Y} \frac{dY}{dy} + \frac{1}{\text{Re } Y} \frac{d^2 Y}{dy^2} - \frac{w}{Z} \frac{dZ}{dz} + \frac{1}{\text{Re } Z} \frac{d^2 Z}{dz^2} \\ &= \lambda \text{ (a separation constant),} \end{aligned}$$

From (4.14), one gets the general equation for X as

$$\frac{d^2 X}{dx^2} - u \text{Re} \frac{dX}{dx} + \lambda \text{Re } X = 0 \quad (4.14)$$

$$\text{Solving, } X(x) = C_1 e^{m_1 x} + C_2 e^{m_2 x} \quad \text{where} \quad m = \frac{u \text{Re} \pm \sqrt{u^2 \text{Re}^2 - 4\lambda \text{Re}}}{2}$$

Using the similar procedure, the solutions for $Y(y)$ and $Z(z)$ are

$$\frac{d^2 Y}{dy^2} - v \frac{dY}{dy} + \sigma \text{Re } Y = 0 \quad (4.15)$$

$$\text{hence, } Y(y) = C_3 e^{\gamma_1 y} + C_4 e^{\gamma_2 y} \quad \text{where} \quad \gamma = \frac{v \text{Re} \pm \sqrt{v^2 \text{Re}^2 - 4\sigma \text{Re}}}{2}.$$

$$\text{Also } \frac{d^2 Z}{dz^2} - w \frac{dZ}{dz} + (\lambda + \sigma) \text{Re } Z = 0 \quad (4.16)$$

$$\text{hence, } Z(z) = C_5 e^{\zeta_1 z} + C_6 e^{\zeta_2 z} \quad \text{where} \quad \zeta = \frac{w \text{Re} \pm \sqrt{w^2 \text{Re}^2 - 4(\lambda + \sigma) \text{Re}}}{2},$$

and σ, λ are additional separation constants. Hence, steady state solution to (4.13) becomes

$$\begin{aligned}
q(x, y, z) &= X(x)Y(y)Z(z) \tag{4.17} \\
&= \left(C_1 e^{u \operatorname{Re} x} + C_2 \right) \left(C_3 e^{v \operatorname{Re} y} + C_4 \right) \left(C_5 e^{w \operatorname{Re} z} + C_6 \right), \text{ for } \lambda = \sigma = 0 \\
&= \left(C_1 e^{\operatorname{Re} x} + C_2 \right) \left(C_3 e^{\operatorname{Re} y} + C_4 \right) \left(C_5 e^{\operatorname{Re} z} + C_6 \right), \text{ for } u = v = w = 1
\end{aligned}$$

4.2 Asymptotic Convergence :

The assumption implicit in a discrete approximation numerical procedure is that the discrete solution will approach that of the continuum problem in the limit of mesh refinement. While the exact solution is rarely known, the convergence of the discrete solution to the exact solution can still be studied in terms of “asymptotic” convergence under discretization refinement in suitable norms.

For an integer $m \geq 1$, the Sobolev space W_p^m , of order m , p , on Ω of functions $v(\mathbf{x})$ is defined, c.f., (Oden and Reddy, 1976) as

$$W_p^m(\Omega) = \{v : D^\alpha v \in L_p(\Omega); \forall \alpha, |\alpha| \leq m\}, \quad p \geq 1 \tag{4.18}$$

where Ω is a bounded open set in $\mathbb{R}^d$, and $D^\alpha v$ stands for the α th weak derivative of v . A locally integrable (on $\Omega \subset \mathbb{R}^d$) function v has a *weak derivative* χ^α of order α , if for every test function $\phi \in \mathcal{D}(\Omega)$, there exists a locally integrable function χ^α such that

$$\int_\Omega \chi^\alpha \phi dx = (-1)^{|\alpha|} \int_\Omega v D^\alpha \phi dx \tag{4.19}$$

Then, the general Sobolev norm of order m associated with (4.18), c.f., Oden and Reddy (1976), is defined for each $v(\mathbf{x}) \in W_p^m$ as

$$H^m(v) = \|v\|_{W_p^m(\Omega)} = \left\{ \int_{\Omega} \sum_{|\alpha| \leq m} |D^\alpha v|^p d\Omega \right\}^{1/p} = \left\{ \sum_{|\alpha| \leq m} \|D^\alpha v\|_{L_p(\Omega)}^p \right\}^{1/p} < \infty \quad (4.20)$$

Hence, on a W_2^3 space, a function v exists with weak derivatives up to and including third order in L_2 . Let $\Omega = (a, b) \times (c, d) \times (e, f) \subset \mathbb{R}^3$ and $v(x, y, z) \in W_2^3(\Omega)$. Then the corresponding Sobolev norm of v is

$$\begin{aligned} \|v\|_{W_2^3(\Omega)} = & \left[\int_a^b \int_c^d \int_e^f \left(|v|^2 + \left| \frac{\partial v}{\partial x} \right|^2 + \left| \frac{\partial v}{\partial y} \right|^2 + \left| \frac{\partial v}{\partial z} \right|^2 \right. \right. \\ & + \left| \frac{\partial^2 v}{\partial x^2} \right|^2 + \left| \frac{\partial^2 v}{\partial y^2} \right|^2 + \left| \frac{\partial^2 v}{\partial z^2} \right|^2 \\ & \left. \left. + \left| \frac{\partial^2 v}{\partial x \partial y} \right|^2 + \left| \frac{\partial^2 v}{\partial y \partial z} \right|^2 + \left| \frac{\partial^2 v}{\partial z \partial x} \right|^2 \right) dx dy dz \right]^{1/2} \end{aligned} \quad (4.21)$$

In general, functions used in CFD approximations lie on W_p^1 which reduces (4.21) to

$$\|v\|_{W_p^1(\Omega)} = \left[\int_a^b \int_c^d \int_e^f \left(|v|^p + \left| \frac{\partial v}{\partial x} \right|^p + \left| \frac{\partial v}{\partial y} \right|^p + \left| \frac{\partial v}{\partial z} \right|^p \right) dx dy dz \right]^{1/p} \quad (4.22)$$

The highest derivative portion of (4.22), including the corresponding data (Re^{-1} , Pe^{-1}) and also the Robin boundary data, is referred to as the *energy* semi-norm H^E in FEM and is defined as

$$H^E = \|v\|_E = \frac{1}{2 \text{Re}} \int_{\Omega} (\nabla v) \cdot (\nabla v) d\Omega, \quad \text{where } \nabla = \frac{\partial}{\partial x} + \frac{\partial}{\partial y} + \frac{\partial}{\partial z} \quad (4.23)$$

Note that the definition (4.23) is made in the mean-square form, in place of the root mean-square form used in defining H^m . An additional norm measure in frequent

use is the L_2 - or H^0 norm defined as

$$H^0(v) = \|v\|_0 = \left(\int_{\Omega_e} v^2 dx \right)^{\frac{1}{2}} \quad (4.24)$$

For a linear elliptic boundary value problem of type $Aq = f$, where A is a properly elliptic differential operator of order $2m$ with smooth coefficients and $f \geq 0$, the FEM asymptotic error estimate, as given by Oden and Reddy (1976, pp. 337) for uniform mesh refinement, is

$$\|e^h\|_{H^s(\Omega)} \leq Ch^\mu \|q\|_{H^{k+1}(\Omega)}, \quad \mu = \min(k+1-s, 2(k+1-m)) \quad (4.25)$$

where $0 \leq s \leq m, k > 0$ and q is the exact solution to the m -th order variational statement.

The discrete approximation error for $q^h(x) \in \mathfrak{R}^1$ is bounded in energy in the form (Strang and Fix, 1973, Ch. 2)

$$\|e^h\|_{E(\Omega)} \leq Ch_e^{2(k+1-m)} \|q\|_{H^{k+1}}^2 \quad (4.26)$$

where $k \geq 1, C$ is a constant independent of h_e , the piecewise mesh measure, and $\|q\|_{H^{k+1}}$ is the Sobolev norm containing up to the $(k+1)$ st derivative of the exact solution (q). Hence, for sufficiently smooth solutions, the error in the finite element approximation, measured in energy, converges as the $2(k+1-m)$ power of mesh measure.

For the linear initial-value parabolic equation, *i.e.*, (4.2) with $u = 0$ and $\epsilon > 0$, the asymptotic error estimate in energy for an explicit Euler integration is (Oden and Reddy, 1976, Theorem 9.6, Ch. 9),

$$\|e^h(n\Delta t)\|_{H^m(\Omega)} \leq C_1 h_e^{(k+1-m)} \|q(n\Delta t)\|_{H^{k+1}(\Omega)} + C_5 \Delta t^{f(\theta)} \|q_0\|_{H^m(\Omega)} \quad (4.27)$$

where C_1 and C_5 are constants (> 0) independent of mesh measure h_e . The first term on the right side of (4.27) is identically that in the convergence statement for the elliptic equation, while the second term contains the H^0 norm of the interpolation of initial data onto the degrees of freedom of q^h .

Equation (4.2) degenerates to linear first order hyperbolic for $u \neq 0$ and $\epsilon \rightarrow 0$. Again, for an implicit ($\theta = 1$ in 3.21-3.23) Euler integration, the asymptotic estimate for smooth solutions, as given by Oden and Reddy (1976, Theorem 9.11, Ch.9), is

$$\begin{aligned} \|e^h(n\Delta t)\|_{H^m(\Omega)} &\leq C_1 h_e^{k+1} \|q(n\Delta t)\|_{H^{k+1}(\Omega)} + C_5 \Delta t^2 \|q_0\|_{H^m(\Omega)} \\ &\quad + C_6 h_e^2 \int_0^{n\Delta t} \|q\|_{H^{k+1}(\Omega)} d\tau \end{aligned} \quad (4.28)$$

where C_6 is another constant independent of h_e . The first two terms remain similar to those in (4.27), while the third term in (4.28) now dominates convergence, via accumulation of the $(k+1)$ st derivatives of the analytical solution over $n\Delta t$ multiplying h_e^2 .

The use of p -element methodology produces different estimates for accuracy and asymptotic convergence. For an elliptic boundary value problem, Jensen and Suri (1992) determine that

$$\|e_p^h\|_E = \|q - q_p^h\|_{E(\Omega)} \leq C p^{-2\alpha-3/2} \|q\|_{H^p}^2 \quad (4.29)$$

where C is a constant independent of q and p , and α is not an integer. Specifically, in a polar coordinate system the superscript $\alpha = \min_i \{\alpha_i\} = \pi / \max_i \{w_i\}$, $\alpha > 1/2$ and $0 < w_i < 2\pi$, $w_i \neq \pi$.

For a first order hyperbolic problem using a Petrov-Galerkin method, Jensen (1992) found

$$\|q - q_p^h\|_{L_2} \leq C p^{k-\kappa} \|q\|_{H^k} \quad (4.30)$$

where $q \in H^k(\Omega)$ for $k > \kappa$, $\kappa \approx 1/2$.

Zienkiewicz and Zhu (1992) found superconvergence of the p -adaptive algorithm using a patch recovery technique that yielded the asymptotic convergence estimate in the order

$$\mathcal{O}(h^{p+\beta}), \quad \beta = 1 \text{ for odd } p, \text{ and } \beta = 2 \text{ for even } p \quad (4.31)$$

for elliptic boundary value and parabolic ($d = 1$) problems.

The standard way of measuring error and convergence is via uniform mesh refinement studies in a suitable norm. However, in most practical (non-linear) situations, (uniform) mesh refinement studies are impossible to execute, hence subsequent solution improvement strategies need be better defined. Several mathematically elegant error estimation techniques are available, leading to improved solution quality by forcing the local *a posteriori* estimate of the *error distribution* to be more uniform, c.f., Babuška *et. al* (1992), Jensen and Suri (1992), Oden *et. al* (1989), Targowski *et. al* (1992), Zienkiewicz *et. al* (1982). But even with all these estimation theories, obtaining a “good” solution is often quite challenging.

Furthermore, error in boundary flux is not addressed in these analyses although the appropriate terms are present in the “energy” semi-norm. The Neumann (or Robin) boundary, *e.g.*, heat flux and drag in turbulent flow in CFD, can be calculated

from the solution field data or in the solution process. Use of the finite difference formulations for estimation is subject to error as an inverse power function of the mesh measure, *i.e.*, coarser grid brings forth coarser approximation of the solution slope. Alternatively, one may utilize the weak statement formulation WS^h itself to estimate fluxes where Dirichlet data are enforced. Equation (3.19) can be rewritten as

$$-\int_{\partial\Omega_e \cap \partial\Omega^h} \phi_i(f_j - f_j^v)^h \hat{n}_j d\sigma = \int_{\Omega^h} \phi_i \left(\frac{\partial q^h}{\partial t} - s \right) d\tau - \int_{\Omega^h} \frac{\partial \phi_i}{\partial x_j} (f_j - f_j^v)^h d\tau \quad (4.32)$$

Hence, the nonhomogeneous boundary condition for the problem statement (2.1) may be found within the WS^h via (3.20)-(3.22) as a boundary matrix, $[B]$,

$$[B] = \{R\} - ([U] + [D])\{Q\} + \{b\} \quad (4.33)$$

4.3 von Neumann Stability

The stability analysis of any numerical algorithm is usually cast as a von Neumann frequency analysis. For (1.1), the Fourier representation is given by (1.2)-(1.9). A Galerkin Weak Statement (GWS) for (1.1) does not produce a diffusion matrix $[D]_e$ in (3.20), while a TWS formulation does. The order of accuracy of a numerical algorithm may be determined from its amplification factor G (1.8, $n = 1$) by matching coefficients of powers in wavenumber of the Laurent series expansion with the analytical solutions G_a in (1.11). Using a Taylor series expansion one may readily see that the linear basis FE approximation to the spatial derivative about the point j is

of the form

$$\frac{q_{j+1} - q_{j-1}}{2\Delta x} = \left[\frac{\partial q}{\partial x} + \frac{\Delta x^2}{3} \frac{\partial^3 q}{\partial x^3} + \frac{\Delta x^4}{60} \frac{\partial^5 q}{\partial x^5} \right]_j + \mathcal{O}(\Delta x^6) \quad (4.34)$$

which is obviously second order accurate. Similarly, the θ -implicit temporal approximation, $0 \leq \theta \leq 1$, may also be represented in a Taylor series as

$$\{Q\}^{n+\Delta t} = \{Q\}^n + \Delta t \left(\theta \frac{d\{Q\}^{n+\Delta t}}{dt} + (1-\theta) \frac{d\{Q\}^n}{dt} \right) + \mathcal{O}(\Delta t^2, \Delta t^3) \quad (4.35)$$

The trapezoidal ($\theta = 1/2$) GWS form (4.35) is second order accurate for the linear basis FE construction. Higher order accuracy may be achieved by using a higher degree FE basis, or the TWS can maintain the third order accuracy of the linear FE basis form with added dissipation.

For a uniform mesh, Lele (1992) has reduced the Taylor series truncation error of the FD approximation to the first order spatial derivative terms, by including the $j \pm 3$, $j \pm 2$ terms in (4.34), as

$$\begin{aligned} & \beta \frac{dQ_{j-2}}{dx} + \alpha \frac{dQ_{j-1}}{dx} + \frac{dQ_j}{dx} + \alpha \frac{dQ_{j+1}}{dx} + \beta \frac{dQ_{j+2}}{dx} \\ &= a \frac{q_{j+1} - q_{j-1}}{2\Delta x} + b \frac{q_{j+2} - q_{j-2}}{4\Delta x} + c \frac{q_{j+3} - q_{j-3}}{6\Delta x} + \dots \end{aligned} \quad (4.36)$$

where the following relation must hold

$$a + b + c = 1 + 2\alpha + 2\beta \quad (4.37)$$

In one dimension ($\Omega \subset \mathbb{R}^{2+}$), the linear basis degree FEWS may be written as a recursion relation about generic node j for a space-time pair $(-\Delta x, \Delta x) \times (t + n\Delta t, t + (n+1)\Delta t)$ in the form

$$(b_1 Q_{j-1} + b_2 Q_j + b_3 Q_{j+1})_{t+(n+1)\Delta t} = (a_1 Q_{j-1} + a_2 Q_j + a_3 Q_{j+1})_{t+n\Delta t} \quad (4.38)$$

For the linear basis matrices and an elementwise constant fluid speed U , the coefficients in (4.38) are

$$a_1 = \frac{\Delta x}{6} + \frac{(1-\theta)\Delta t U}{2}, a_2 = \frac{2\Delta x}{3}, a_3 = \frac{\Delta x}{6} - \frac{(1-\theta)\Delta t U}{2} \quad (4.39)$$

and

$$b_1 = \frac{\Delta x}{6} - \frac{\theta\Delta t U}{2}, b_2 = \frac{2\Delta x}{3}, b_3 = \frac{\Delta x}{6} + \frac{\theta\Delta t U}{2} \quad (4.40)$$

From (1.7), the following relations hold

$$Q_{j-1}^n = e^{i\omega((j-1)\Delta x - U^h n \Delta t)} \quad (4.41)$$

$$Q_{j-1}^{n+1} = e^{i\omega((j-1)\Delta x - U^h (n+1) \Delta t)}$$

$$Q_j^n = e^{i\omega(j\Delta x - U^h n \Delta t)}$$

$$Q_j^{n+1} = e^{i\omega(j\Delta x - U^h (n+1) \Delta t)}$$

$$Q_{j+1}^n = e^{i\omega((j+1)\Delta x - U^h n \Delta t)}$$

$$Q_{j+1}^{n+1} = e^{i\omega((j+1)\Delta x - U^h (n+1) \Delta t)}$$

and (4.41), $U^h = U_{\mathfrak{R}}^h + iU_{\mathfrak{I}}^h$ is the phase velocity in one dimension, recall (1.3)-(1.5).

Substituting (4.41) in (4.38), and recalling the nondimensional wavenumber $m = \omega\delta x$, one may derive via (1.8)

$$\begin{aligned} G &= e^{-i\omega U^h \Delta t} = e^{-i\omega U_{\mathfrak{R}}^h \Delta t} e^{\omega U_{\mathfrak{I}}^h \Delta t} \\ &= \frac{a_1 e^{-im} + a_2 + a_3 e^{im}}{b_1 e^{-im} + b_2 + b_3 e^{im}} \end{aligned} \quad (4.42)$$

$$\begin{aligned}
&= 1 - C^2\theta m^2 + C^4\theta^3 m^4 - C^6\theta^5 m^6 + \frac{1}{90}C^2\theta m^6 + \mathcal{O}(m^8) \\
&\quad + i \left(-Cm + C^3\theta^2 m^3 - \left(C^5\theta^4 - \frac{C}{180} \right) m^5 + \mathcal{O}(m^7) \right) \\
&= G_{\Re} + iG_{\Im}
\end{aligned}$$

The Courant number definition is $C = U\Delta t/\Delta x$, and $0 \leq \theta \leq 1$ is the implicitness factor of the Euler time integration family.

Recall, the relative phase velocity for the semi-discrete approximation is defined as the ratio

$$\Phi^h = \frac{U^h}{U} \quad (4.43)$$

Hence, an alternative definition of the amplification factor (4.42) is

$$G = e^{-i\omega U^h \Delta t} = e^{-imC\Phi^h} \quad (4.44)$$

Using the following identities

$$2\cos m = e^{-im} + e^{im}, \text{ and } 2\sin m = e^{im} - e^{-im} \quad (4.45)$$

one may derive from (4.43)

$$G_{\Re} = \cos mC\Phi^h, \text{ and } G_{\Im} = -\sin mC\Phi^h \quad (4.46)$$

Hence, the relative phase velocity becomes

$$\Phi^h = \frac{\tan^{-1}(-G_{\Im}/G_{\Re})}{mC} = \frac{\tan^{-1}(G_{\Im}/G_{\Re})}{-mC} \quad (4.47)$$

(Note: in their Journal of Computers and Fluids paper, Pepper and Cooper (1980) derived the phase velocity as $\Phi(C) = \frac{\tan^{-1}(G_{\Im}/G_{\Re})}{\omega\Delta x}$ which is obviously incorrect, see (4.47)).

SUB-GRID EMBEDDING, AN ORDER ADAPTIVE SEMI-DISCRETIZATION

5.1 Standard Finite Element

The elements of the standard basis functions $\{N_k(\bullet)\}$ are:

One dimensional bases:	Linear	Quadratic
for $\zeta \in (-1, 1)$	$(1 - \zeta)/2$	$-\zeta(1 - \zeta)/2$
	$(1 + \zeta)/2$	$\zeta(1 + \zeta)/2$
		$(1 - \zeta)(1 + \zeta)$
Two dimensional linear bases:	η_1 direction	η_2 direction
for $\eta_1, \eta_2 \in (-1, 1)$	$(1 - \eta_1)/2$	$(1 - \eta_2)/2$
	$(1 + \eta_1)/2$	$(1 + \eta_2)/2$
Quad points	1d linear element points	quad bases
1	$1(12)X \otimes 2(41)Y$	$\left(\frac{1 - \eta_1}{2}\right) \left(\frac{1 - \eta_2}{2}\right)$
2	$2(12)X \otimes 1(23)Y$	$\left(\frac{1 + \eta_1}{2}\right) \left(\frac{1 - \eta_2}{2}\right)$
3	$2(23)X \otimes 1(34)Y$	$\left(\frac{1 + \eta_1}{2}\right) \left(\frac{1 + \eta_2}{2}\right)$
4	$2(34)X \otimes 1(41)Y$	$\left(\frac{1 - \eta_1}{2}\right) \left(\frac{1 + \eta_2}{2}\right)$

Figure 5.1a)-b) illustrates the formation of the bilinear and biquadratic bases for $\Omega \subset (\eta_1 \times \eta_2)$. The bases functions for the quadrilateral element are calculated at each element node as if the sides are “looking towards the center of the element”.

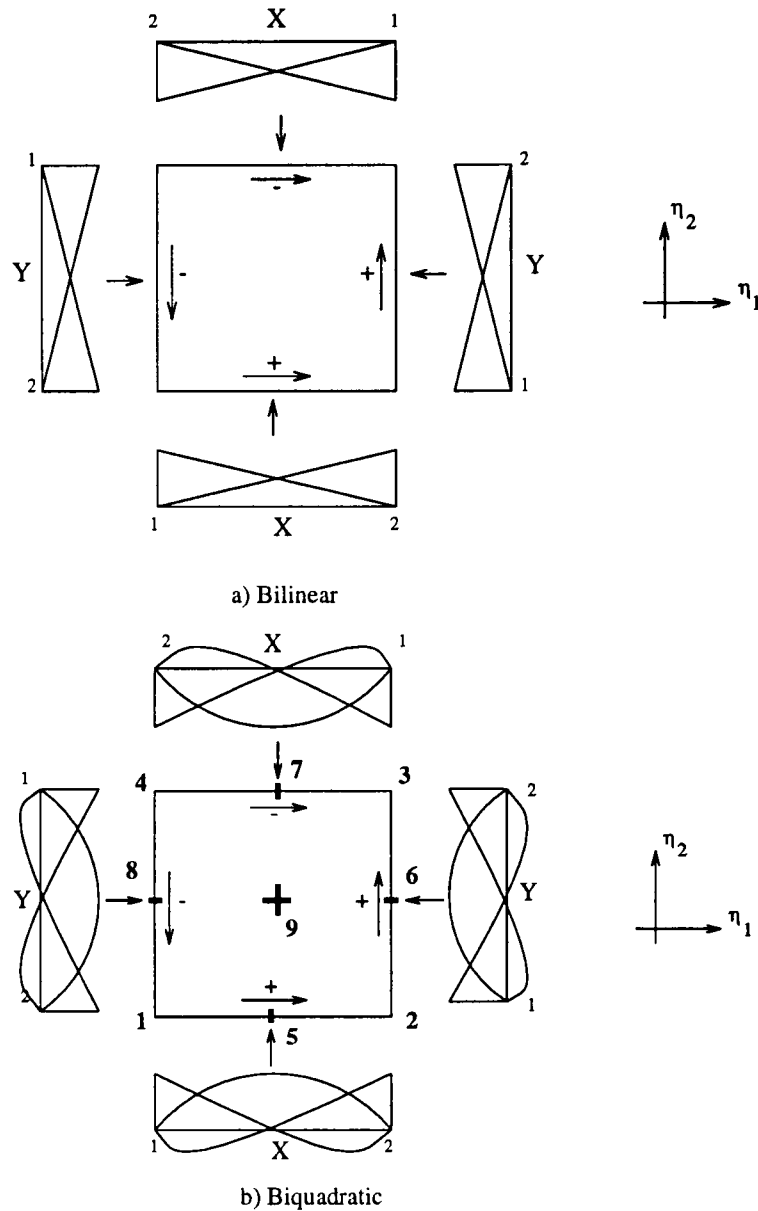


Figure 5.1 Standard two dimensional elements

5.2 p -version Finite Elements

p -version finite elements selectively enrich the trial function sets through addition of local higher order polynomials of degree $p - 1$. The corresponding shape functions N_p in one dimension are, Zienkiewicz, *et. al* (1970)

$$N_0 = \frac{1 - \zeta}{2}, \quad N_1 = \frac{1 + \zeta}{2}, \quad N_2 = (1 - \zeta)(1 + \zeta), \quad (5.1)$$

$$N_p = \frac{1}{(p-1)!} \frac{1}{2^{p-2}} \frac{d^{p-2}}{d\zeta^{p-2}} (1 - \zeta^2)^{p-1}$$

Two dimensional shape functions for rectangular elements calculated as the corner node functions are simply bilinear products of the form (5.1) with the constraint of zero at corners.

Finite elements in which successive polynomial additions are additive in the manner of additional terms in a Fourier series, are called "hierarchical" elements. Hierarchical functions are always zero at the (vertex) nodes of the linear basis on any d -dimensional finite element. This implies that the laplacian "stiffness" matrix $[D]$ in (3.20), the hierarchical basis at a certain level of polynomial addition, say p_1 , is a submatrix of the stiffness matrix corresponding to a higher level of polynomial addition p_2 ($p_2 > p_1$). Hence, matrix algebraic equations are of the type

$$\begin{bmatrix} D_{11} & D_{12} & \cdot & D_{1,n} \\ D_{21} & D_{22} & \cdot & D_{2,n} \\ \cdot & \cdot & \cdot & \cdot \\ D_{n,1} & D_{n,2} & \cdot & D_{n,n} \end{bmatrix} \begin{Bmatrix} q_1^{(1)} \\ q_2^{(1)} \\ \cdot \\ q_n^{(1)} \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \\ \cdot \\ f_n \end{Bmatrix} \quad (5.2)$$

and

$$\begin{bmatrix} D_{11} & D_{12} & \cdot & D_{1,n} & D_{1,n+1} \\ D_{21} & D_{22} & \cdot & D_{2,n} & D_{2,n+1} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ D_{n,1} & D_{n,2} & \cdot & D_{n,n} & D_{n,n+1} \\ D_{n+1,1} & D_{n+1,2} & \cdot & D_{n+1,n} & D_{n+1,n+1} \end{bmatrix} \begin{Bmatrix} q_1^{(2)} \\ q_2^{(2)} \\ \cdot \\ q_n^{(2)} \\ q_{n+1}^{(2)} \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \\ \cdot \\ f_n \\ f_{n+1} \end{Bmatrix} \quad (5.3)$$

Here, D_{ij} , $1 \leq i, j \leq n+1$ are the entries of the stiffness matrix $[D]$, evolved from the integral of derivatives of the corresponding shape functions, $q_i^{(1)}$, $1 \leq i \leq n$, are the nodal unknowns at the initial level of the polynomial distribution, $q_i^{(2)}$, $1 \leq i \leq n+1$, are the unknowns at the next level of polynomial embedding, and f_j , $1 \leq j \leq n+1$, are the elements of the "force" vector. A typical nine node two dimensional hierarchical element has four bilinear shape functions for its corner nodes (vertices), $(k-1)$, $k \geq 1$, shape functions corresponding to each midside node and $(k-1)^2$, $k > 1$, shape functions corresponding to the central node, Oden, *et. al* (1989). Figure 5.2 illustrates the one dimensional h - p element structure. The two dimensional hierarchical p element construction is shown in Figure 5.3.

In hierarchical p -version, the discretization process yields a system of algebraic equations for the initial (linear) approximation

$$[D^l] \{Q^l\} = \{f^l\} \quad (5.4)$$

where f is the given data. An improvement in the initial approximation can accrue by introduction of an additional *local* "nodeless" variable which is hierarchical.

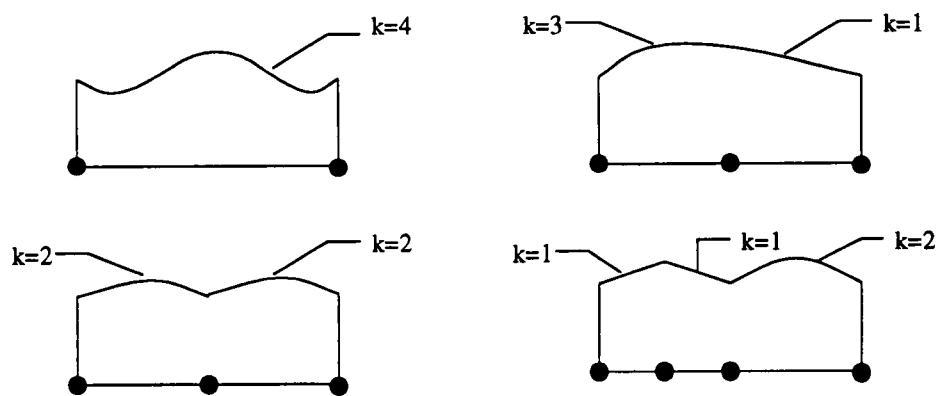


Figure 5.2 Combinations of h-p in a five degrees of freedom 1d element, Rachowicz, *et. al* (1989).

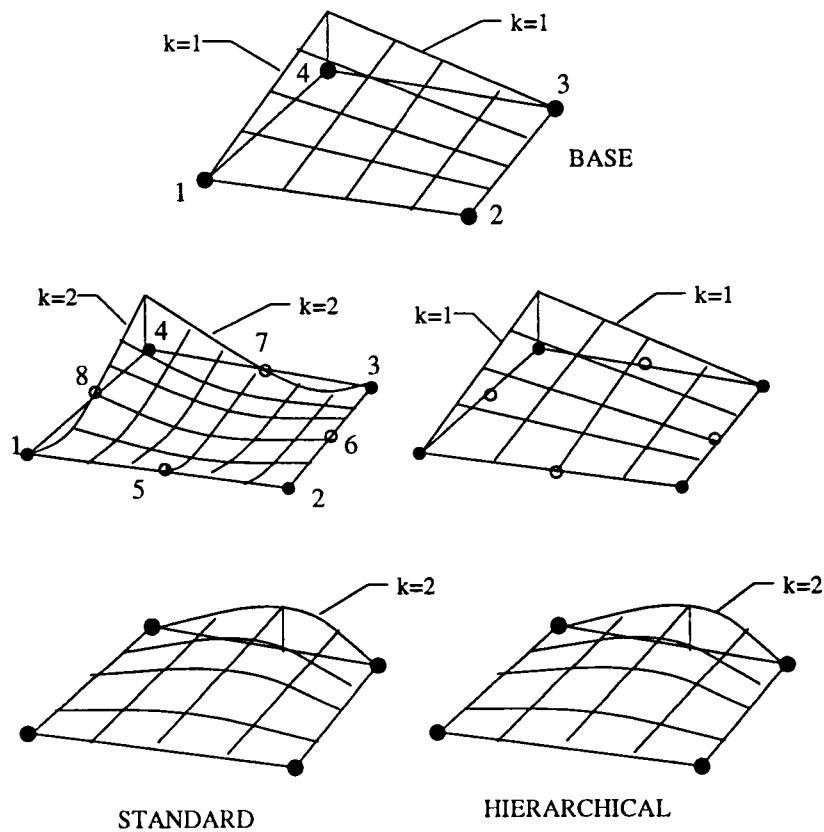


Figure 5.3 Increasing polynomials in a 2d p -element Zienkiewicz and Craig (1984).

This variable is then reduced from the **global matrix** using static condensation (5.9)-(5.10), Zienkiewicz and Craig (1984). Hence, the next level of approximation becomes

$$q^h = N_i Q_i + k_{n+1} N_{n+1} = q^l + k_{n+1} N_{n+1} \quad (5.5)$$

Rearrangement of the governing equation system with this introduction of hierarchical variables in the classical Galerkin weak statement process yields

$$k_{n+1} = \frac{(Q_{n+1} - D_{n+1,i} Q_i)}{D_{n+1,n+1}} \quad (5.6)$$

For each additional higher order hierarchical approximation, the associated coefficients can be evaluated following the similar procedure. Since the hierarchical functions are always zero at the linear basis vertex nodes, classical hierarchical elements do not allow local extremization and always reduce to the exact linear degree unperturbed form.

5.3 A Definition :

In standard p -version FE methodology, matrix condensation is a global operation, while in SGM (static) condensation is a local operation. A distinction between the basis degree in (5.30) and (5.33), and the local element matrix order in (5.31) and (5.34), needs to be notationally enforced. Herein, I elect to define p as the *extra* degrees of polynomial extremized (*i.e.*, matrix order reduced) on a **local** element via (5.10). Hence, with this definition,

$$k \geq 1, \quad p = 0 \text{ for FEM, including } p\text{-versions} \quad (5.7)$$

$$k = 1, \quad p \geq 1 \text{ for SGM}$$

Via (5.7), $p = 0$ for all standard FEM algorithms and the local matrix order remains $(k + 1)^d \times (k + 1)^d$. For SGM, independent of the choice for $p \geq 1$, the order of the local matrices always reduces to $(2^d \times 2^d)$, *i.e.*, linear, since the $k = 1$ basis always defines the terminal form order for all d .

5.4 An Alternative Idea

As the number of approximation degrees of freedom increases, especially for multi-dimensional solution domains, the element (and global) matrix orders associated with (3.20)-(3.24), and hence the computer resource requirements, increase rapidly. For example, in one dimension, the bandwidth of the global matrix is $2k + 1$, where k is the degree of the basis function, and it is $m + 1$ rows long for an m -element discretization.

A basic ingredient of the SGM algorithm development, regarding computer resource efficiency, is local static condensation, Baker (1983), Zienkiewicz and Craig (1984). This absolutely contains escalation of order, hence computer resource demands, via element matrix order reduction using a local extremization. A square element-level matrix $[A]_e$ of order (k^d) , where superscript d denotes the problem dimension, can be resolved into the form,

$$\begin{bmatrix} A_{(2^d)} & A_{(2^d, m)} \\ A_{(m, 2^d)} & A_{(m)} \end{bmatrix}_{(k+1)^d} \begin{Bmatrix} q^{(2^d)} \\ q^{(m)} \end{Bmatrix} = \begin{Bmatrix} F^{(2^d)} \\ F^{(m)} \end{Bmatrix} \quad (5.8)$$

In (5.8), the reducible part of the matrix vector system is denoted by the subscript/superscript m . Note that in one dimension for p -version, the parent matrix size in (5.8) is of order $(k * nel + 1)$, where nel is the total number of elements in the discretization, while it is only $(k + 1)$ for SGM. Matrix equation system (5.8) may be reduced to another element matrix $[A]^r$ of the order 2^d in the form

$$[A]_{(2^d)}^r \{\tilde{q}\}^r = \{\tilde{F}\}^r \quad (5.9)$$

The definitions for (5.8)-(5.9) are

$$[A]_{(2^d)}^r = [A]_{(2^d)} - [A]_{(2^d, m)} [A]_{(m)}^{-1} [A]_{(m, 2^d)} \quad (5.10)$$

with

$$\{\tilde{F}\}^r = \{\tilde{F}^{(2^d)}\} - [A]_{(2^d, m)} [A]_{(m)}^{-1} \{\tilde{F}^{(m)}\} \quad (5.11)$$

and $m = k^d - 2^d$ degrees of freedom on Ω_e have been reduced out in the process.

For example, in one dimension the element mass $[M]_e$ matrix formed by the quadratic Lagrange basis functions $\{N_2\}$ on Ω_e of measure h_e is

$$[M_2]_e = \int_{\Omega_e} \{N_2\} \{N_2\}^T dx = \frac{h_e}{30} \begin{bmatrix} 4 & -1 & 2 \\ -1 & 4 & 2 \\ 2 & 2 & 16 \end{bmatrix} \quad (5.12)$$

Via (5.8)-(5.10), (5.12) is reduced to an order-2 matrix as

$$[M_2]_e^r = \frac{h_e}{30} \left(\begin{bmatrix} 4 & -1 \\ -1 & 4 \end{bmatrix} - \frac{1}{16} \begin{bmatrix} 2 \\ 2 \end{bmatrix} \begin{bmatrix} 2 & 2 \end{bmatrix} \right) = \frac{h_e}{24} \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} \quad (5.13)$$

which differs appreciably from the linear basis form

$$[M_1]_e = \frac{h_e}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

5.5 Eigenvalue characterization

It is direct to characterize the matrices used in the solution algorithm (3.22), and what static condensation does to them. The eigenvalues λ associated with any matrix, *e.g.* $[M]_e$ and/or its static condensation $[M]_e^r$, can be computed by the solving the characteristic equation

$$\det|[M]_e - \lambda[I]| = 0 \quad (5.14)$$

where $[I]$ is the identity matrix of appropriate order. Although I illustrate only one dimensional matrices, the process and discussion is valid for higher dimensional problems.

In the following discussions, it is essential to recall the *Jordan Canonical form theorem* (Ortega, 1972). For a d -dimensional space, let $A \in L(C^d)$, $C^d = \Re^d + \Im^d$. Then A is similar to a block diagonal matrix

$$J = \begin{bmatrix} J_1 & & & \\ & J_2 & & \\ & & \ddots & \\ & & & J_m \end{bmatrix} \quad (5.15)$$

In (5.15), each J_i is either 1×1 matrix (i.e., λ_i , hence A is a **class M** matrix) or a matrix of the form

$$J_i = \begin{bmatrix} \lambda_i & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{bmatrix} \quad (5.16)$$

where λ_i is an eigenvalue. Again, a matrix $A \in L(\mathbb{R}^d)$ is called an **M-matrix** if $A^{-1} \geq 0$ and $a_{ij} \leq 0, i \neq j$. For $a_{ij} = a_{ji}$, an **M-matrix** is a **Stieltjes** matrix. A **Stieltjes** matrix is positive definite.

5.5.1 Element Diffusion or Stiffness matrix, $[D]_e$

For the linear basis, and $d = 1$

$$[D_1]_e = \int_{\Omega_e} \{N_1\}_{,x} \{N_1\}_{,x}^T dx = \frac{1}{h_e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (5.17)$$

where $\{N_1\} = \frac{1}{2}\{1 \mp \eta\}^T$. Hence, for a m element uniform discretization, ($h_e = h$), the global assembly over all the finite elements e , $1 \leq e \leq m+1$, yields the contribution to $\{R\}$ or $\{f\}$ in (3.24) via the product of

$$[D_1] = S_e([D]_e) = \frac{1}{h} \begin{bmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \circ \\ & -1 & 2 & -1 & \\ & & -1 & 2 & -1 \\ \circ & & & \ddots & \\ & & & & 1 \end{bmatrix} \quad (5.18)$$

with the nodal degree of freedom matrix $\{Q\}$. Note that (5.17) is a **Stieltjes** matrix.

Using (5.14), the eigenvalues (λ_i) of the $[D]_e$ matrix in (5.17) are $\{\lambda\} = \{2, 0\}^T$.

For quadratic lagrange basis polynomial on $d = 1$, the element diffusion matrix takes the form

$$[D_2]_e = \int_{\Omega_e} \{N_2\}_{,x} \{N_2\}_{,x}^T dx = \frac{1}{3h} \begin{bmatrix} 7 & 1 & -8 \\ 1 & 7 & -8 \\ -8 & -8 & 16 \end{bmatrix} \quad (5.19)$$

It can be reduced to an order-2 matrix, via last row/column condensation, (5.8)-(5.10), yielding

$$[D_2]_e^r = \frac{1}{3h} \left(\begin{bmatrix} 7 & 1 \\ 1 & 7 \end{bmatrix} - \frac{1}{16} \begin{bmatrix} -8 \\ -8 \end{bmatrix} \begin{bmatrix} -8 & -8 \end{bmatrix} \right) = \frac{1}{h} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (5.20)$$

which is identical to the linear basis matrix (5.17). Similarly, for $k = 3$,

$$\begin{aligned} [D_3]_e^r &= \frac{1}{40h} \begin{bmatrix} 148 & -13 \\ -13 & 148 \end{bmatrix} \\ &\quad - \begin{bmatrix} -189 & 54 \\ 54 & -189 \end{bmatrix} \begin{bmatrix} 432 & -297 \\ -297 & 432 \end{bmatrix}^{-1} \begin{bmatrix} -189 & 54 \\ 54 & -189 \end{bmatrix} \\ &= \frac{1}{h} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \end{aligned} \quad (5.21)$$

Hence, static condensation of *any* $k > 1$ element diffusion matrices $[D]_e$ always reproduces the base $k = 1$ matrix. At this point, I have avoided evaluating (5.11), as it would require an *unnecessary* condensation of the interpolated data on extremized

node points. The details of these computations is given in Chapter 8 and in the Appendices.

Hence, the global assembly of the element canonical form is

$$[D_1^{(c)}]^r = S_e([D_1]_e^{(c)}) = \frac{2}{h} \begin{bmatrix} 1 & & & \\ & 1 & & \circ \\ & & 1 & \\ & \circ & & \ddots \\ & & & & 1 \end{bmatrix} = S_e([D_k^{(c)}]_e^r) = [D_k^{(c)}]^r \quad (5.22)$$

In (5.22), the subscript k is the degree of the finite element basis $\{N_k\}$, $[D_k]_e$ is of the order $(k+1) \times (k+1)$, and the superscript r stands for static condensation to order 2×2 .

Note that even the usage of a higher degree basis function, and the subsequent reduction of matrix order by static condensation (5.10), does not change the structure of the assembled global matrix. This is an **excellent** matrix for numerical computation as its canonical assembly form resembles an identity matrix, and this assembly (5.22) does not change for the choice of larger degree k .

5.5.2 Element *Interpolation (Mass) matrix*, $[M]_e$

For the linear basis

$$[M_1]_e = \int_{\Omega_e} \{N_1\} \{N_1\}^T dx = \frac{h_e}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \quad (5.23)$$

The eigenvalues associated with $[M_1]_e$ are $\{\lambda\} = \{\frac{1}{2}, \frac{1}{6}\}$, and the matrix is positive and diagonally dominant. For a m element uniform discretization, the global assembly

over all finite elements e , $1 \leq e \leq m+1$, yields

$$[M_1] = S_e([M]_e) = \frac{h}{6} \begin{bmatrix} 2 & 1 & & \circ \\ 1 & 4 & 1 & \\ & 1 & 4 & 1 \\ & & & \ddots \\ & \circ & & 2 \end{bmatrix} \quad (5.24)$$

and the corresponding canonical assembled form is

$$[M_1^{(c)}]^r = S_e([M_1^{(c)}]_e) = \frac{h}{6} \begin{bmatrix} 3 & & & \\ & 4 & & \circ \\ & & 4 & \\ & \circ & & \ddots \\ & & & 3 \end{bmatrix} \quad (5.25)$$

Now, use of a higher degree basis function, and the subsequent reduction of matrix order by static condensation (5.10), changes the structure of the assembled global matrix. Recalling (5.13), reducing the mass matrix for quadratic lagrange basis polynomials, the eigenvalues become $\{\lambda\} = \{\frac{1}{6}, \frac{1}{12}\}^T$, and the corresponding canonical assembled form is

$$[M_2^{(c)}]^r = S_e([M_2^{(c)}]_e^r) = \frac{h}{12} \begin{bmatrix} 2 & & & \\ & 3 & & \circ \\ & & 3 & \\ & \circ & & \ddots \\ & & & 2 \end{bmatrix} \quad (5.26)$$

In (3.24), the interpolation matrix $[M]$ is associated with time and/or source (data) distribution. Hence, (5.26) may not be dependable matrices for numerical computation, as the reduced forms $[M_k]^r$ substantially changes the canonical assembled form which also loses normalization. This can seriously affect the time accuracy and/or the source level applied. In anticipation, I generally restrict use to linear form $[M_1]$. However, in a time acceleration procedure for monotone unsteady evolution to steady-state, the reduced quadratic basis form $[M_2]^r$ will find use.

5.5.3 Element Convection matrix, $[U]_e$

I uniformly discard consideration of higher (than linear) degree basis functions for convection matrices as follows. For constant velocity $u_e = 1$, the linear basis form is

$$[U_1]_e = u \int_{\Omega} \{N_1\} \{N_1\}_{,x}^T dx = \frac{1}{2} \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \quad (5.27)$$

and the global assembly over all the finite elements e , $1 \leq e \leq m+1$, yields

$$[U_1] = S_e([U_1]_e) = \frac{1}{2} \begin{bmatrix} -1 & 1 & \circ & & \\ -1 & 0 & 1 & & \\ & -1 & 0 & 1 & \\ & & & \ddots & \\ \circ & & & & -1 \end{bmatrix} \quad (5.28)$$

The associated eigenvalues of (5.27) are $\{\lambda\} = \{0, 0\}^T$, thus $[U_1]$ is a singular matrix (with zero eigenvalues). Hence, the assembly of the element canonical form is

$$[U_1^{(c)}]^r = S_e([U_1^{(c)}]^r_e) = \begin{bmatrix} 0 & & & & \\ & 0 & & \circ & \\ & & 0 & & \\ & \circ & & \ddots & \\ & & & & 0 \end{bmatrix} \quad (5.29)$$

This null matrix is useless for numerical computation. Hence, (5.29) indicates that one should exercise caution when condensing a higher degree basis functions (quadratic, or cubic) in forming the convection matrix $[U]_e$.

5.6 Local Embedding

With the reduction of element matrix order to the linear form via (5.10), local extremization allows a selective basis degree (p) choice for each of the element matrices. For example, the SGM algorithm can operate with a linear ($k = 1$) basis function for element convection matrices, linear or condensed quadratic ($k = 2$) basis function for the element mass matrix, while all higher degree ($k > 1$) basis functions may be chosen for condensed element diffusion matrices. For this to function requires a non-linear embedding. One example for any k is

$$\int_{\Omega_e} f_1(x) \{N_k\}_{,x} \{N_k\}_{,x}^T dx \quad (5.30)$$

where $f_1(x) = \{a, b\} \{N_1\}^T$ and $\{N_1\} \equiv \frac{1}{2} \{1 \mp \eta\}^T$ is a piecewise linear interpolation of embedded data with nodal values (a, b) . For basis degree $k = 2$, reducing the

matrix (5.30) by static condensation (5.10) yields

$$[D_2]^r = \frac{1}{3(a+b)h_e} \begin{bmatrix} (a^2 + b^2 + 4ab) & -(a^2 + b^2 + 4ab) \\ -(a^2 + b^2 + 4ab) & (a^2 + b^2 + 4ab) \end{bmatrix} \quad (5.31)$$

As a check, for $a = b = 1$, (5.31) becomes

$$[D_2]^r = \frac{1}{h_e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (5.32)$$

which is identical to the linear basis form. Thus, the linear embedding $f_1(x)$ is “consistent.”

An alternative embedding employs a piecewise quadratic interpolation of data $f_1(x)$. Hence,

$$\int_{\Omega_e} f_2(x) \{N_k\}_{,x} \{N_k\}_{,x}^T dx \quad (5.33)$$

and one specific identification is $f_2(x) = \{1, c, 1\} \{N_2\}^T$ and $\{N_2\}$ is the basis function of quadratic degree. In this instance, f_2 is piecewise quadratic with nodeless degree of freedom c , eligible for evaluation for non-unit value via the newly developed theory herein. Under local extremization, for basis degree $k = 2$, matrix (5.33) reduced by static condensation (5.10) yields

$$[D_2]^r = \frac{1}{3h_e} \begin{bmatrix} (2c+1) & -(2c+1) \\ -(2c+1) & (2c+1) \end{bmatrix} \quad (5.34)$$

As a special case, for $c = 1$, hence $f_2(x) = \text{constant}$, (5.34) becomes

$$[D_2]^r = \frac{1}{h_e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (5.35)$$

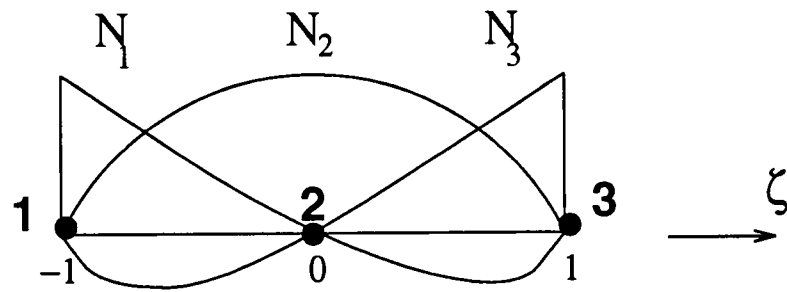
which is again identical to the linear basis form. Hence, this parabolic embedding is also *consistent*.

Figure 5.4 demonstrates comparison of standard, hierarchical and sub-grid embedded elements in one dimension. At element level, both standard and hierarchical cases retain an extra degree of freedom (node **2**), while for SGM, this extra degree of freedom is reduced. A study of extremized diffusion matrices, computed for SGM choice f_1 and f_2 , are given in Appendix E. Appendix H documents the definite integrable SGM basis functions, N_1^R and N_2^R , defined as

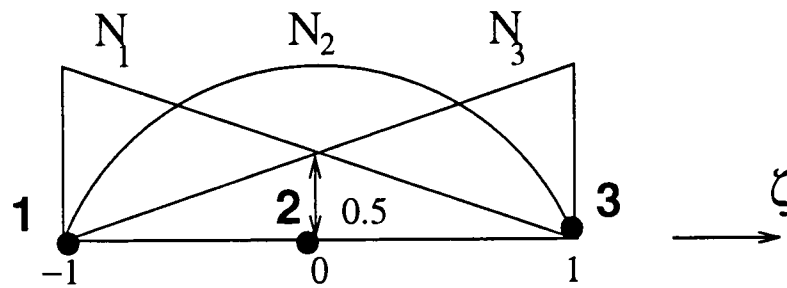
$$\int_{\Omega_e} \{N_1^R\}_{,x} \{N_2^R\}_{,x} dx = \int_{\Omega_e} f_2(x) \{N_k\}_{,x} \{N_k\}_{,x}^T dx \quad (5.36)$$

In two dimensions, the comparison element structures are documented in Figures 5.5-5.7. Figure 5.5 shows a hierarchical two dimensional element with two quadratic sides, one cubic side and a linear side. The locally extremized p -element, Figure 5.6, originally begins with one cubic and a quadratic side which are shown as extra local nodes, *e.g.*, 3 and 4 on side **12**. Finally, the new SGM element is documented for $f_2(\mathbf{x})$, Figure 5.7. Note that the directional subgrid embedding parameters, c_x and c_y , can be calculated exactly for a specific linear or nonlinear problem class (described later).

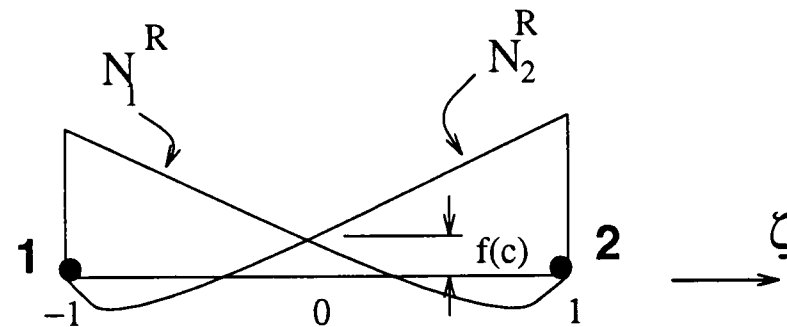
The combination of f_1 and f_2 in the form $\tilde{f}_2(x) = \{a, c, b\} \{N_2\}^T$ in (5.33), is also a candidate of specific potential use. For example, in the wall layer region of a turbulent boundary layer flow, the turbulent viscosity ν^t could be distributed using coefficients a and b , while c remains available for monotonicity control in the convection part of the conservation law system.



a) standard element $k=2$



b) hierarchical element $k=2, p=0$



c) SGM element $k=1, p=1$ embedded

Figure 5.4 Comparison of basis functions for standard, hierarchical and SGM elements in one dimension.

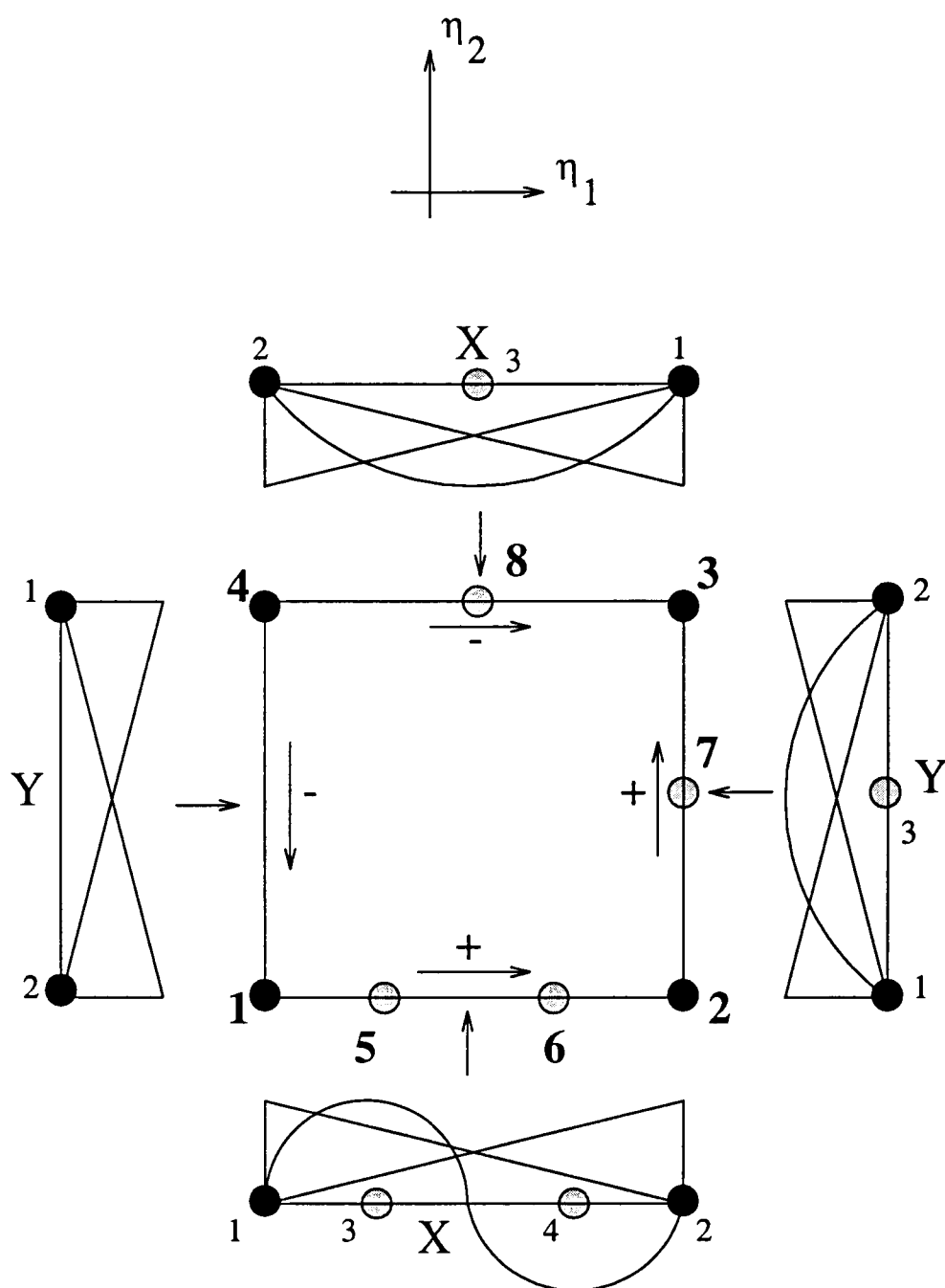


Figure 5.5 Two dimensional hierarchical p -element

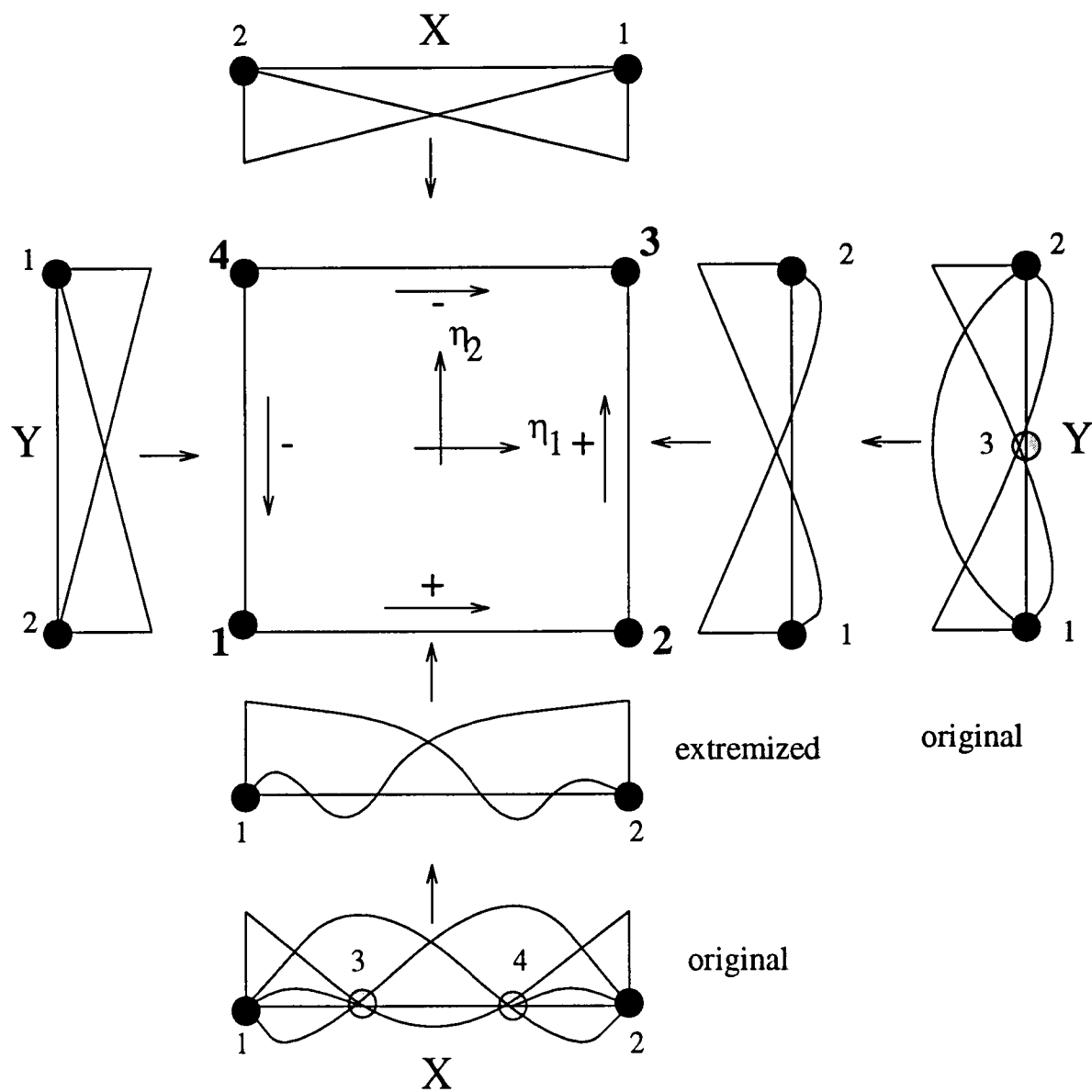


Figure 5.6 Two dimensional locally extremized p -element

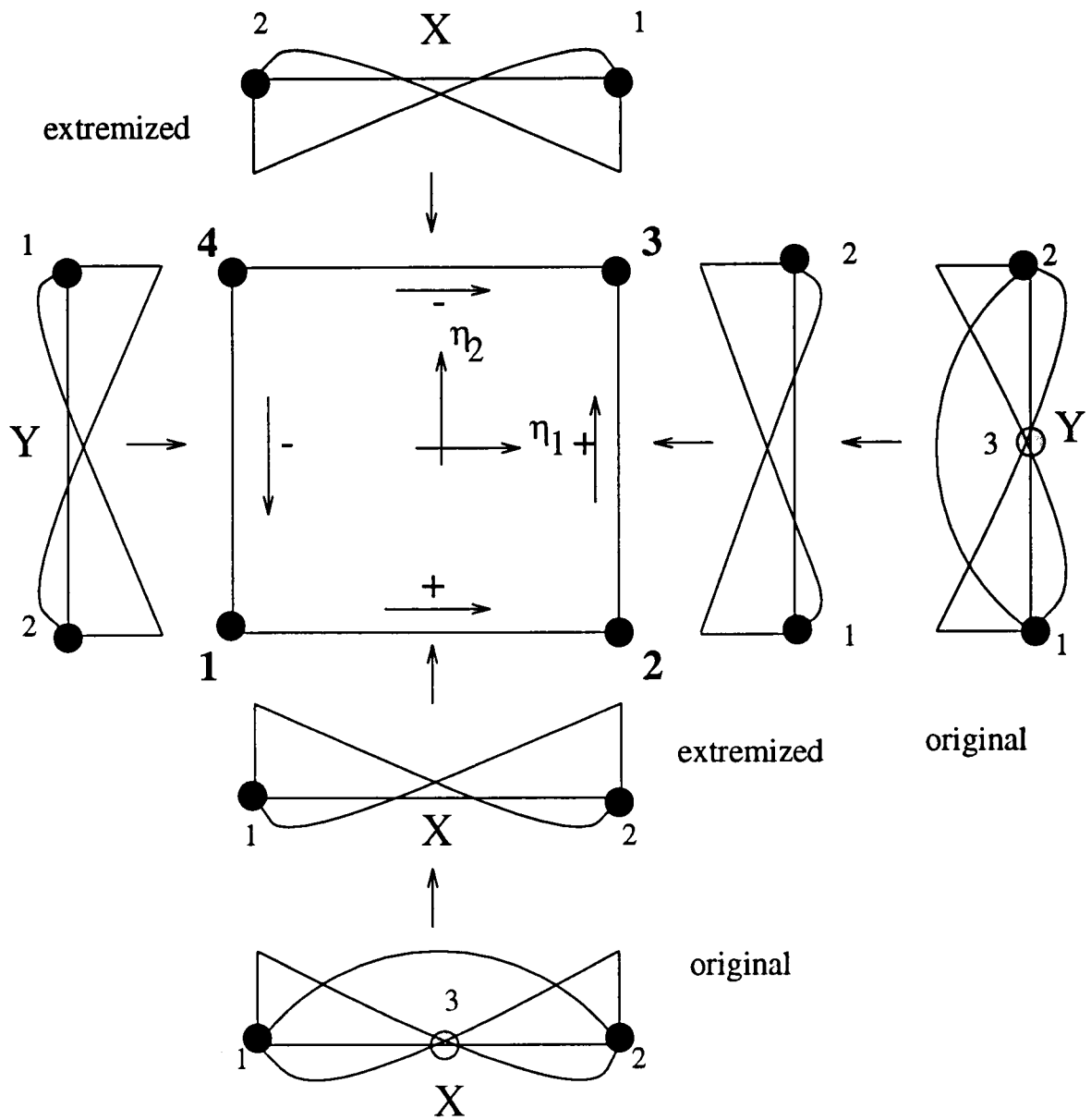


Figure 5.7 Two dimensional SGM $p = 1$ element

5.7 Stability and Monotonicity for Steady State

For the SGM choice $f_2(x)$, $k = 1$, $p = 1$, and a solution independent of time, the 1d steady advection-diffusion equation ($s = 0$, $u = U = \text{a constant}$) spatially uniform semi-discretized weak statement algorithm (3.20), assembled at node j with variable $Q_j = Q(j\Delta x)$, can be expressed as the recursion relation

$$\frac{(2c_e + 1)\epsilon}{3h_e} [-Q_{j-1} + 2Q_j - Q_{j+1}] + \frac{U}{2} [-Q_{j-1} + Q_{j+1}] = 0 \quad (5.37)$$

The consistency check results for (5.37), since $c = 1$ produces the standard linear finite element stencil

$$\frac{\epsilon}{h_e} [-Q_{j-1} + 2Q_j - Q_{j+1}] + \frac{U}{2} [-Q_{j-1} + Q_{j+1}] = 0 \quad (5.38)$$

An alternative form for (5.37) is

$$-\left(\frac{(2c_e + 1)\epsilon}{3h_e} + \frac{U}{2}\right) Q_{j-1} + \left(\frac{2(2c_e + 1)\epsilon}{3h_e}\right) Q_j - \left(\frac{(2c_e + 1)\epsilon}{3h_e} - \frac{U}{2}\right) Q_{j+1} = 0 \quad (5.39)$$

which for a nonuniform meshing becomes

$$\begin{aligned} & -\left(\frac{(2c_L + 1)\epsilon}{3h_L} + \frac{U}{2}\right) Q_{j-1} + \left(\frac{(2c_L + 1)\epsilon}{3h_L} + \frac{(2c_R + 1)\epsilon}{3h_R}\right) Q_j \\ & -\left(\frac{(2c_R + 1)\epsilon}{3h_R} - \frac{U}{2}\right) Q_{j+1} = 0 \end{aligned} \quad (5.40)$$

where subscripts "L" and "R" denote to the left and right of node j respectively.

An interpretation of the distinction between standard FEM and SGM accrues to denoting the (uniform) FEM mesh measure as Δx while h_e remains that for SGM. Hence, comparing (5.37) and (5.38) yields

$$\Delta x = \frac{h_e}{\left(\frac{2c_e + 1}{3}\right)} \quad (5.41)$$

One implication of (5.41) is that the FEM mesh measure needs be $\left(\frac{2c+1}{3}\right)$ times denser than the SGM mesh measure for an identical solution. Alternatively, a lesser number of nodes in SGM may produce comparable solution quality. Importantly, SGM assembled matrices are always tridiagonal (in 1-D), hence the minimal degrees of freedom are handled in the solution process.

For the SGM linear embedding with $f_1(x)$, $k = 1$, $p = 1$, the generic assembled recursion relation for arbitrary meshing is

$$\begin{aligned} & - \left(\frac{(a_L^2 + b_L^2 + 4a_L b_L)\epsilon}{3(a_L + b_L)h_L} + \frac{U}{2} \right) Q_{j-1} \\ & + \left(\frac{(a_L^2 + b_L^2 + 4a_L b_L)\epsilon}{3(a_L + b_L)h_L} + \frac{(a_R^2 + b_R^2 + 4a_R b_R)\epsilon}{3(a_R + b_R)h_R} \right) Q_j \\ & - \left(\frac{(a_R^2 + b_R^2 + 4a_R b_R)\epsilon}{3(a_R + b_R)h_R} - \frac{U}{2} \right) Q_{j+1} = 0 \end{aligned} \quad (5.42)$$

Thereby, for Δx the standard FEM mesh measure, the form comparable to (5.41) is

$$\Delta x = \frac{h_e}{\left(\frac{a^2 + b^2 + 4ab}{3(a+b)} \right)_e} \quad (5.43)$$

Hence, for given h_e , an adroit choice of the subgrid nodeless parameter c_e in $f_2(x)$ leads to $\left(\frac{2c+1}{3}\right)_e^{2(k+1-m)}$ lower error bound for asymptotic convergence, (4.27)-(4.29), while for $f_1(x)$, the comparison factor is $\left(\frac{a^2 + b^2 + 4ab}{3(a+b)}\right)_e^{2(k+1-m)}$.

It is clear from the recursion stencil forms, (5.40)-(5.41), that in place of decreasing h_e , one can increase the subgrid embedding parameter c_e in (5.39) to obtain the same effect. This coefficient operation in one-dimension certainly possesses an operator relationship with artificial diffusion. However, in SGM, ϵ remains the common

multiplier and there is no parabolic perturbation as with numerical viscosity methods, recall (2.3). For $f_1(x)$, this direct control over local mesh measure, is lost in (5.43), since a_e and b_e are assumed given data on Ω_e .

The $(m+1) \times (m+1)$ system matrix associated with an m finite element 1-D discretization has an assembly stencil identical to (5.39)-(5.42). The corresponding tridiagonal matrix has eigenvalues

$$\lambda^l = \frac{2(2c+1)\epsilon}{3h} - 2\sqrt{\left(\frac{(2c+1)\epsilon}{3h} + \frac{U}{2}\right)\left(\frac{(2c+1)\epsilon}{3h} - \frac{U}{2}\right)} \cos(l\pi h), \quad 1 \leq l \leq m \quad (5.44)$$

Thus, λ^l can be represented as a Pick function, Donoghue (1974), of the form $\Phi(\zeta) = \alpha(\zeta) + i\beta(\zeta)$, where $i = \sqrt{-1}$. Hence, the solution of the eigenvalue ODE system will be monotone if λ^l does not have any imaginary part, Iannelli (1992). This can be ensured by the constraint that the quadratic embedding SGM parameter c_e satisfy

$$\left| \frac{(2c_e+1)\epsilon}{3h_e} \right| \geq \left| \frac{U}{2} \right| \quad (5.45)$$

For any $c_e \geq 0$ and $\epsilon > 0$, λ^l is positive, hence the algorithm is always absolutely stable. Rearranging (5.45), for $u_e = U = a$ cellwise constant and all $\epsilon > 0$, the SGM solution will be strictly monotone for

$$c_e \geq \frac{u_e}{2} \left(\frac{3h_e}{2\epsilon_e} \right) - \frac{1}{2} \quad (5.46)$$

For a pure diffusion equation ($U = 0$), the SGM construction is strictly monotone for all $c_e \geq -1/2$.

For the SGM linear embedding $f_1(x)$, the ODE solution is monotone for $h_e = a$ constant provided

$$\left| \frac{(a^2 + b^2 + 4ab)\epsilon}{3(a+b)h} \right|_e \geq \left| \frac{U}{2} \right| \quad (5.47)$$

Stability of the algorithm for the steady form of (4.2) is also guaranteed, as the real part of the Fourier eigenvalue λ^l is nonnegative for all $(a, b)_e > 0$.

Equations (5.37)-(5.47) hold for the linear problem definition. For a non-linear problem, *e.g.*, viscous Burgers equation, the numerical linear algebra formulation (3.24)-(3.25) becomes $[J]\{\delta U\} + \{F\} = 0$, where

$$\{F\} = \{U\}_e [A3001]_e \{U\}_e + \{\epsilon\}_e [A3011]_e \{U\}_e \quad (5.48)$$

For SGM choice $f_2(x)$ the above relation becomes

$$\{F\} = \{U\}_e [A3001]_e \{U\}_e + \{\epsilon, r\}_e [A3011]_e \{U\}_e \quad (5.49)$$

where

$$r = \frac{2c+1}{3} \quad (5.50)$$

and from Baker (1983)

$$[A3001]_e = \frac{1}{6} \begin{bmatrix} \begin{Bmatrix} -2 \\ -1 \end{Bmatrix} & \begin{Bmatrix} 2 \\ 1 \end{Bmatrix} \\ \begin{Bmatrix} -1 \\ -2 \end{Bmatrix} & \begin{Bmatrix} 1 \\ 2 \end{Bmatrix} \end{bmatrix}, \quad [A3011]_e = \frac{1}{2h_e} \begin{bmatrix} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} & \begin{Bmatrix} -1 \\ -1 \end{Bmatrix} \\ \begin{Bmatrix} -1 \\ -1 \end{Bmatrix} & \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \end{bmatrix} \quad (5.51)$$

Assuming that (3.24) converges to the right solution, *i.e.*, $u^{p+1} = u^p$ and hence $\delta u = 0 \Rightarrow \{F\} \equiv \{0\}$. Therefore, the recursion relation for the assembled form is

$$\begin{aligned} & - \left(\frac{r_{j-1}\epsilon_{j-1}}{h_{j-1}} + \frac{r_j\epsilon_j}{h_j} + \frac{U_{j-1} + 2U_j}{6} \right) U_{j-1} \\ & + \left(\frac{2r_j\epsilon_j}{h_j} + \frac{r_{j-1}\epsilon_{j-1}}{h_{j-1}} + \frac{r_{j+1}\epsilon_{j+1}}{h_{j+1}} + \frac{U_{j-1} - U_{j+1}}{6} \right) U_j \\ & - \left(\frac{r_j\epsilon_j}{h_j} + \frac{r_{j+1}\epsilon_{j+1}}{h_{j+1}} - \frac{2U_j + U_{j+1}}{6} \right) U_{j+1} = 0 \end{aligned} \quad (5.52)$$

The solution to this recursion relation will be monotone if the eigenvalue of the system matrix formed by (5.52) is free of imaginary part. The condition for that is

$$\left(\frac{2U_j}{6} + \frac{r_j\epsilon_j}{h_j} \right) \left(\frac{2U_j}{6} - \frac{r_j\epsilon_j}{h_j} \right) \geq 0 \quad (5.53)$$

Hence, the monotonicity constraint for the non-linear parent differential equation is

$$r_j = \frac{2c_j + 1}{3} \geq \frac{h_j|U_j|}{3\epsilon_j} \quad (5.54)$$

The general form of monotonicity constraint in one dimension is

$$r_j = \frac{2c_j + 1}{3} \geq \frac{h_j|U_j|}{\mathcal{F}\epsilon_j} \quad (5.55)$$

where $\mathcal{F}$ is a constant whose value depends on the problem specification.

5.8 Monotone Time Acceleration

The classical element mass matrix $[M]_e$ associated with the time integral in (3.20) for an unsteady problem can adversely affect monotonicity of a transient solution.

This can be avoided by embedding f_2 into $[M_2]$ and condensing, yielding for $d = 1$

$$[M_2]_e^r = \frac{h_e}{24} \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} \quad (5.56)$$

Note the condensed mass matrix in (5.56) is not normalized, since the sum of all elements $\sum_{i=1}^n m_{ij}$ for a $n \times n$ matrix of order n is unity for a normalized matrix.

Therefore, introduce a normalization scalar $\phi > 0$ into (5.56) such that

$$[M_2]_e^r = \frac{h_e}{4\phi} \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} \quad (5.57)$$

which is normalized for $\phi = 1$ and equals (5.56) for $\phi = 6$.

The implication of ϕ is that it can be associated with the speed of the transient solution process. The companion recursion relation in a fully discretized scheme $Q_j^n = Q(j\Delta x, n\Delta t)$ on uniform mesh is

$$\begin{aligned} & - \left(\frac{h}{4\phi\theta\Delta t} + \frac{(2c+1)\epsilon}{3h} + \frac{U}{2} \right)_e Q_{j-1} + \left(\frac{6h}{4\phi\theta\Delta t} + \frac{2(2c+1)\epsilon}{3h} \right)_e Q_j \\ & - \left(\frac{h}{4\phi\theta\Delta t} + \frac{(2c+1)\epsilon}{3h} - \frac{U}{2} \right)_e Q_{j+1} = \text{known data} \end{aligned} \quad (5.58)$$

while for a nonuniform mesh

$$\begin{aligned} & - \left(\frac{h_L}{4\phi_L\theta\Delta t} + \frac{(2c_L+1)\epsilon}{3h_L} + \frac{U}{2} \right) Q_{j-1} \\ & + \left(\frac{3h_L}{4\phi_L\theta\Delta t} + \frac{(2c_L+1)\epsilon}{3h_L} + \frac{3h_R}{4\phi_R\theta\Delta t} + \frac{(2c_R+1)\epsilon}{3h_R} \right) Q_j \\ & - \left(\frac{h_R}{4\phi_R\theta\Delta t} + \frac{(2c_R+1)\epsilon}{3h_R} - \frac{U}{2} \right) Q_{j+1} = \text{known data} \end{aligned} \quad (5.59)$$

For $f_1(x)$, the companion recursion relation is

$$\begin{aligned}
& - \left(\frac{h_L}{4\phi_L\theta\Delta t} + \frac{(a_L^2 + b_L^2 + 4a_Lb_L)\epsilon}{3(a_L + b_L)h_L} + \frac{U}{2} \right) Q_{j-1} \\
& + \left(\frac{3h_L}{4\phi_L\theta\Delta t} + \frac{(a_L^2 + b_L^2 + 4a_Lb_L)\epsilon}{3(a_L + b_L)h_L} + \frac{3h_R}{4\phi_R\theta\Delta t} + \frac{(a_R^2 + b_R^2 + 4a_Rb_R)\epsilon}{3(a_R + b_R)h_R} \right) Q_j \\
& - \left(\frac{h_R}{4\phi_R\theta\Delta t} + \frac{(a_R^2 + b_R^2 + 4a_Rb_R)\epsilon}{3(a_R + b_R)h_R} - \frac{U}{2} \right) Q_{j+1} = \text{known data} \quad (5.60)
\end{aligned}$$

From (5.58)-(5.60), the speed of the transient solution process can be controlled by ϕ . Specifically, selection of ϕ can accelerate the monotone achievement of the steady state using a smaller number of timesteps. Note, that for $\phi = 1$ the matrix $[M_2]_e^r$ is fully normalized, hence the monotone transient evolution is time-accurate.

5.9 Monotonicity and Stability for Transient Problems

A Fourier modal analysis can determine how the error-induced perturbations propagate through a fully discrete approximation. Hence, via (1.7) in one dimension, expressing $Q_j^n(j\Delta x, t + n\Delta t)$ into its Fourier components of the form,

$$Q_j^n \equiv G^n e^{i\omega(j\Delta x - Ut)} \quad (5.61)$$

where $i = \sqrt{-1}$ and $\omega = 2\pi/\lambda$, $\lambda = \Re\Delta x$ is the associated wavelength. From (3.20) and (5.61), the algorithm specific amplification factor G is

$$G = \frac{Q_j^{n+1}}{Q_j^n} = \frac{G_{numr}}{G_{deno}} \quad (5.62)$$

where introducing the Courant number, $C = U\Delta t/\Delta x = U\Delta t/h$

$$G_{numr} = \frac{1}{4\phi} \left[6 - \left(e^{-i(\omega\Delta x)} + e^{i(\omega\Delta x)} \right) \right]$$

$$\begin{aligned}
& -\frac{(1-\theta)\epsilon(2c+1)\Delta t}{3h^2} \left[2 - \left(e^{-i(\omega\Delta x)} + e^{i(\omega\Delta x)} \right) \right] \\
& -(1-\theta)\frac{C}{2} \left(-e^{-i(\omega\Delta x)} + e^{i(\omega\Delta x)} \right)
\end{aligned} \tag{5.63}$$

and

$$\begin{aligned}
G_{deno} &= \frac{1}{4\phi} \left[6 - \left(e^{-i(\omega\Delta x)} + e^{i(\omega\Delta x)} \right) \right] \\
&+ \frac{\theta\epsilon(2c+1)\Delta t}{3h^2} \left[2 - \left(e^{-i(\omega\Delta x)} + e^{i(\omega\Delta x)} \right) \right] \\
&+ \theta\frac{C}{2} \left(-e^{-i(\omega\Delta x)} + e^{i(\omega\Delta x)} \right)
\end{aligned} \tag{5.64}$$

Using the identities

$$\begin{aligned}
\cos(\omega\Delta x) &= \frac{e^{-i(\omega\Delta x)} + e^{i(\omega\Delta x)}}{2} \\
i \sin(\omega\Delta x) &= \frac{-e^{-i(\omega\Delta x)} + e^{i(\omega\Delta x)}}{2}
\end{aligned} \tag{5.65}$$

one can readily verify that (5.62) becomes

$$G = \frac{\frac{1}{4\phi} [6 - 2\cos(\omega\Delta x)] - \frac{(1-\theta)\epsilon(2c+1)\Delta t}{3h^2} [2 - 2\cos(\omega\Delta x)] - i(1-\theta)C \sin(\omega\Delta x)}{\frac{1}{4\phi} [6 - 2\cos(\omega\Delta x)] + \frac{\theta\epsilon(2c+1)\Delta t}{3h^2} [2 - 2\cos(\omega\Delta x)] + i\theta C \sin(\omega\Delta x)} \tag{5.66}$$

The amplification factor G is an indicator of growth or decay of a solution perturbation. For $|G| < 1$, a solution perturbation decays, hence the algorithm is stable. For $|G| = 1$, the solution is neutrally stable, *i.e.* solution perturbations are neither damped nor amplified as evolution proceeds. Finally, if $|G| > 1$, the solution perturbation will amplify, hence eventually lead to divergence.

Case I ($\theta = 0$) :

For this explicit time integration procedure, and recognizing $\Delta x = h$, yields for

(5.66)

$$G = \frac{\frac{1}{4\phi} [6 - 2 \cos(\omega h)] - \frac{\epsilon(2c+1)\Delta t}{3h^2} [2 - 2 \cos(\omega h)] - iC \sin(\omega h)}{\frac{1}{4\phi} [6 - 2 \cos(\omega h)]} \quad (5.67)$$

Case II ($\theta = 0.5$) :

The *trapezoidal* semi-implicit time integration amplification factor is

$$G = \frac{\frac{1}{4\phi} [6 - 2 \cos(\omega h)] - \frac{\epsilon(2c+1)\Delta t}{6h^2} [2 - 2 \cos(\omega h)] - i\frac{C}{2} \sin(\omega h)}{\frac{1}{4\phi} [6 - 2 \cos(\omega h)] + \frac{\epsilon(2c+1)\Delta t}{6h^2} [2 - 2 \cos(\omega h)] + i\frac{C}{2} \sin(\omega h)} \quad (5.68)$$

Case III ($\theta = 1.0$) :

For the fully implicit time integration, (5.66) is

$$G = \frac{\frac{1}{4\phi} [6 - 2 \cos(\omega h)]}{\frac{1}{4\phi} [6 - 2 \cos(\omega h)] + \frac{\epsilon(2c+1)\Delta t}{3h^2} [2 - 2 \cos(\omega h)] + iC \sin(\omega h)} \quad (5.69)$$

The $(m+1) \times (m+1)$ system matrix associated with (5.58) for a m finite element discretization will yield a tridiagonal matrix structure for which the eigenvalues in closed form are

$$\lambda^l = \frac{3}{2\phi\theta\Delta t} + \frac{2(2c+1)\epsilon}{3h^2} - 2\sqrt{\left(\frac{1}{4\phi\theta\Delta t} + \frac{(2c+1)\epsilon}{3h^2} + \frac{U}{2h}\right)\left(\frac{1}{4\phi\theta\Delta t} + \frac{(2c+1)\epsilon}{3h^2} - \frac{U}{2h}\right)} \cos(l\pi h),$$

where $1 \leq l \leq m$ (5.70)

Hence, λ^l can be represented as a Pick function of the form $\Phi(\zeta) = \alpha(\zeta) + i\beta(\zeta)$, where $i = \sqrt{-1}$, and the solution of the ODE system will be monotone if λ^l does not have any imaginary part. This can be assured by a particular choice of the subgrid embedding c and renormalization ϕ such that

$$\left| \frac{1}{4\phi\theta} + \frac{(2c+1)\epsilon\Delta t}{3h^2} \right| \geq \left| \frac{C}{2} \right| \quad (5.71)$$

where θ is the implicitness factor, $0 \leq \theta \leq 1$, and $C = U\Delta t/h$ is the Courant number.

Again, for any c , and $\phi \geq 1$, λ^l is positive. Hence, the algorithm is absolutely stable for all time steps. Note that for a pure diffusion equation, $U = 0$ and the algorithm solution becomes strictly monotone for $\phi \geq 1$.

5.10 Multidimensional SGM form

In multidimensions ($d \geq 1$), $f_2(\mathbf{x})$ can be written as

$$f_2(\mathbf{x}) = \{S\}\{N_2\}^T \quad (5.72)$$

where, for $\mathbf{c} = (c_x, c_y, c_z)$, the column matrix $\{S\}$ is the following tensor product

$$\{S(\mathbf{x})\} = \{1, c_x, 1\} \otimes \{1, c_y, 1\} \otimes \{1, c_z, 1\} \quad (5.73)$$

In (5.73), for a given velocity field $\mathbf{u} = (u, v, w)$ and a diffusion field $(\epsilon_x, \epsilon_y, \epsilon_z)$, the condition for monotone solution for each component of $\mathbf{c}$ is

$$\begin{aligned} c_x &\geq \frac{1}{\mathcal{F}} \left(\frac{3uh_x}{2\epsilon_x} - 1 \right) \\ c_y &\geq \frac{1}{\mathcal{F}} \left(\frac{3vh_y}{2\epsilon_y} - 1 \right) \end{aligned} \quad (5.74)$$

$$c_z \geq \frac{1}{\mathcal{F}} \left(\frac{3wh_z}{2\epsilon_z} - 1 \right)$$

where $\mathcal{F} > 1$ and $c_x, c_y, c_z \geq 1$.

For example, in two dimensions for a master square element $(-1:1, -1:1)$ the standard diffusion matrix for bilinear basis is

$$[D_1]_e = \int_{\Omega_e} \nabla \{N_1^+\} \bullet \nabla \{N_1^+\}^T d\tau = \frac{1}{6} \begin{bmatrix} 4 & -1 & -2 & -1 \\ -1 & 4 & -1 & -2 \\ -2 & -1 & 4 & -1 \\ -1 & -2 & -1 & 4 \end{bmatrix} \quad (5.75)$$

while the SGM $f_2(\mathbf{x})$ choice diffusion matrix $[D]_e$ can be written as

$$[D_{\text{SGM}}]_e = \int_{\Omega_e} f_2(c_x, c_y) \nabla \{N_2^+\} \bullet \nabla \{N_2^+\}^T d\tau \quad (5.76)$$

In (5.75)-(5.76), $\{N_1^+\}$ and $\{N_2^+\}$ are bilinear and biquadratic basis functions, respectively, and $f_2(c_x, c_y) = \{1, 1, 1, 1, c_x, c_y, c_x, c_y, c_x c_y\} \{N_2^+\}$.

Via (5.9), the locally extremized form of (5.76) is

$$[D_{\text{SGM}}]_e = \frac{1}{6} \begin{bmatrix} D_{11} & D_{12} & D_{13} & D_{14} \\ D_{21} & D_{22} & D_{23} & D_{24} \\ D_{31} & D_{32} & D_{33} & D_{34} \\ D_{41} & D_{42} & D_{43} & D_{44} \end{bmatrix} \quad (5.77)$$

For $c_x = c_y = 1$, I called the algorithm as $p\text{-WS}^h$, while for $c_x, c_y \geq 1$, I called it SGM^h . The elements in (5.77) are of the form

$$\begin{aligned}
& (194176c_x + 456864c_x^3 + 313224c_x^2 + 64036) c_y^3 \\
& + (456864c_x^3 + 1088192c_x^2 + 793564c_x + 176830) c_y^2 \\
& + (313224c_x^3 + 793564c_x^2 + 625780c_x + 152182) c_y \\
D_{11} = & \frac{+64036c_x^3 + 176830c_x^2 + 152182c_x + 40077}{(1971360c_x^2 + 2184480c_x + 569160) c_y^2} \quad (5.78) \\
& + (2184480c_x^2 + 3227040c_x + 1077480) c_y \\
& + 569160c_x^2 + 1077480c_x + 479610
\end{aligned}$$

The other elements of the diffusion matrix in (5.77) are given in Appendix F. Using $c_x = c_y = c$, (5.78) yields (details in Appendix F)

$$D_{11} = \frac{90(148c^2 + 164c + 73)}{2c + 1} (656c^3 + 2032c^2 + 1838c + 549) \quad (5.79)$$

For $c_x = c_y = 1$, the matrix in (5.77) becomes

$$[D_{\text{SGM}}]_e = \frac{1}{66} \begin{bmatrix} 29 & -11 & -7 & -11 \\ -11 & 29 & -11 & -7 \\ -7 & -11 & 29 & -11 \\ -11 & -7 & -11 & 29 \end{bmatrix} \quad (5.80)$$

which is obviously different than linear basis GWS^h diffusion matrix $[D]_e$ in (5.75).

Chapter 6

THE GALERKIN MATRIX PERTURBATION (GMP) ALGORITHM

A Fourier modal analysis can determine how the error-induced perturbations propagate through a fully discrete approximation. The algorithm specific amplification factor G for an $n = 1$ step method is

$$G = \frac{Q_j^{t+\Delta t}}{Q_j^t} = e^{-i\omega U^h \Delta t} = e^{-imC^h} \quad (6.1)$$

In (6.1), the discrete Courant number $C^h = \Phi^h C$ depends on the specific solution algorithm where C is the algorithm independent Courant number and Φ^h is the relative phase velocity via (4.43) or (4.47).

Assume that an arbitrary order Taylor series truncation error for the spatial and temporal discretization (4.34)-(4.35) may be expressed in the form of FE perturbation matrices [APERT] and [APERX]. The temporal and convection one dimensional finite element matrices of linear basis order for (3.20b)-(3.20c) become

$$\begin{aligned} \frac{1}{\Delta x_e} [M_1]_e &= [A200L] - [APERT] \\ &= \frac{1}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}_{2 \times 2} - \frac{1}{6} \begin{bmatrix} \Lambda_1 & \Psi_1 \\ \Gamma_1 & \Upsilon_1 \end{bmatrix}_{2 \times 2} \end{aligned} \quad (6.2)$$

$$= \frac{1}{6} \begin{bmatrix} 2 - \Lambda_1 & 1 - \Psi_1 \\ 1 - \Gamma_1 & 2 - \Upsilon_1 \end{bmatrix}_{2 \times 2}$$

and

$$\begin{aligned} \frac{1}{U}[U]_e &= [A201L] - [APERX] \\ &= \frac{1}{2} \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}_{2 \times 2} - \frac{1}{2} \begin{bmatrix} \Lambda_2 & \Psi_2 \\ \Gamma_2 & \Upsilon_2 \end{bmatrix}_{2 \times 2} \\ &= \frac{1}{2} \begin{bmatrix} -(1 + \Lambda_2) & 1 - \Psi_2 \\ -(1 + \Gamma_2) & 1 - \Upsilon_2 \end{bmatrix}_{2 \times 2} \end{aligned} \quad (6.3)$$

6.1 Matrix perturbation terms

In (6.2)-(6.3), [A200L] and [A201L] remain the GWS linear basis constructions.

The interpolation matrix perturbation terms can be expressed as

$$\begin{aligned} \Lambda_1 &= t_1 \sum_{r=0}^{\alpha} a_r z^{-r}, \\ \Psi_1 &= t_2 \sum_{r=0}^{\beta} a_r z^r, \\ \Gamma_1 &= t_3 \sum_{r=0}^{\gamma} a_r z^{-r}, \\ \Upsilon_1 &= t_4 \sum_{r=0}^{\delta} a_r z^r \end{aligned} \quad (6.4)$$

and the convection matrix perturbation terms are

$$\Lambda_2 = t_5 \sum_{r=1}^{\alpha} a_r z^{-r},$$

$$\Psi_2 = t_6 \sum_{r=1}^{\beta} a_r z^r, \quad (6.5)$$

$$\Gamma_2 = t_7 \sum_{r=1}^{\gamma} a_r z^{-r},$$

$$\Upsilon_2 = t_8 \sum_{r=1}^{\delta} a_r z^r$$

where $z = e^{im}$ is the complex modulus. The summation limits $(\alpha, \beta, \gamma, \delta) \geq 0$ are integers, to be determined for desired order of accuracy (Appendix A). However, to maintain time accuracy of the algorithm, one necessary condition for (6.5) is

$$\Lambda_i + \Psi_i + \Gamma_i + \Upsilon_i = 0; \quad \text{for } i = 1, 2 \quad (6.6)$$

For (6.2)-(6.3) the fully discrete weak statement algorithm (3.20) assembly at mesh node j with variable $Q_j^n = Q(j\Delta x, t + n\Delta t)$ yields the recursion relation

$$\begin{aligned} & \left(\frac{\Delta x(1 - \Gamma_1)}{6\theta\Delta t} - \frac{(1 + \Gamma_2)U}{2} \right)_e Q_{j-1}^n \\ & + \left(\frac{\Delta x(4 - \Lambda_1 - \Upsilon_1)}{6\theta\Delta t} + \frac{(\Lambda_2 + \Upsilon_2)U}{2} \right)_e Q_j^n \\ & + \left(\frac{\Delta x(1 - \Psi_1)}{6\theta\Delta t} + \frac{(1 - \Psi_2)U}{2} \right)_e Q_{j+1}^n \\ & = \text{data from } (n - 1) \text{ timestep} \end{aligned} \quad (6.7)$$

6.2 Matching the coefficients of analytical and algorithm amplification factors

The amplification factor (6.1) for (6.7) is

$$G = \frac{Q_j^{t+\Delta t}}{Q_j^t} = \frac{G_N}{G_D} \quad (6.8)$$

where G_N and G_D can be resolved into spatial $G(x)$ and temporal $G(t)$ components.

$$G_N = G(t) - (1 - \theta)CG(x) \quad (6.9)$$

$$G_D = G(t) + \theta CG(x)$$

Then using relations (1.7) and (1.9), via (3.20) and (6.7) in (6.8), yields

$$\begin{aligned} G(t) &= \frac{(1 - \Gamma_1)e^{-im} + (4 - \Lambda_1 - \Upsilon_1) + (1 - \Psi_1)e^{im}}{6} \\ G(x) &= \frac{(1 - \Psi_2)e^{im} - (\Lambda_2 + \Upsilon_2) - (1 + \Gamma_2)e^{-im}}{2} \end{aligned} \quad (6.10)$$

Hence, (6.9) becomes

$$\begin{aligned} G_N &= \left(\frac{1 - \Gamma_1}{6} + (1 - \theta)C \frac{1 + \Gamma_2}{2} \right) e^{-im} \\ &\quad + \frac{4 - \Lambda_1 - \Upsilon_1}{6} + (1 - \theta) \frac{C}{2} (\Lambda_2 + \Upsilon_2) \\ &\quad + \left(\frac{1 - \Psi_1}{6} - (1 - \theta)C \frac{1 - \Psi_2}{2} \right) e^{im} \\ G_D &= \left(\frac{1 - \Gamma_1}{6} - \theta C \frac{1 + \Gamma_2}{2} \right) e^{-im} \\ &\quad + \frac{4 - \Lambda_1 - \Upsilon_1}{6} - \theta \frac{C}{2} (\Lambda_2 + \Upsilon_2) \\ &\quad + \left(\frac{1 - \Psi_1}{6} + \theta C \frac{1 - \Psi_2}{2} \right) e^{im} \end{aligned} \quad (6.11)$$

The GMP algorithm amplification factor (6.8)-(6.11) may now be expanded in a Laurent series on m (Appendix B). The procedure is very tedious; however, using "Macsyma" (1988), and for $\alpha = \beta = \gamma = \delta = 3$, one may obtain up to and including the second order term of the series in the form

$$G = \sum_{l=0}^{\infty} A_l m^l = 1 + (2iCt_8 + 4iCt_6 + 2iCt_5 - iC)m \quad (6.12)$$

$$\begin{aligned}
& - \left((12C^2t_8^2 + (48C^2t_6 + 24C^2t_5 - 12C^2)t_8 + 48C^2t_6^2 \right. \\
& + (48C^2t_5 - 24C^2)t_6 + 12C^2t_5^2 - 12C^2t_5 + 3C^2)\theta \\
& + (4Ct_4 + 8Ct_2 + 4Ct_1 + 6C)t_8 \\
& + (8Ct_4 + 16Ct_2 + 8Ct_1 + 18C)t_6 \\
& + (4Ct_4 + 8Ct_2 + 4Ct_1 - 12C)t_5 \\
& \left. - 2Ct_4 - 4Ct_2 - 2Ct_1 \right\} \frac{m^2}{3} + \mathcal{O}(m^3) \dots
\end{aligned}$$

The coefficients A_l of the series (6.12) are functionally of the form

$$A_l = f(C, \theta, t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, \alpha, \beta, \gamma, \delta) \quad (6.13)$$

where l corresponds to the power of m in the Laurent expansion. Comparing the sets a_l of (1.11) to A_l of (6.12), for a given degree of m a general correlation between C , θ and the elements of [APERT] and [APERX] may be derived (Appendix B) that can yield progressively higher order accurate algorithms without adding a diffusion term to the algorithm.

6.3 Pertinent Examples

A simplified example is pertinent to illustrate the process. Selecting

$$\theta = \frac{1}{2}, \alpha = \beta = \gamma = \delta = 1, \text{ and } t_5 = t_8 = 0 \quad (6.14)$$

then for the choice

$$t_1 = t_4, t_2 = t_3, \text{ and } t_6 = -t_7 \quad (6.15)$$

one may produce a fourth order accurate GMP algorithm by satisfying the relationships

$$t_1 + t_2 = 0 \quad (6.16)$$

$$t_6 = -t_7 = \frac{t_2 + t_1}{3} = 0 = t_5 = t_8$$

$$t_1 = t_4 = \frac{3C^2 + (4 - C^2)t_2}{C^2 + 2} = -t_2 = -t_3$$

$$i.e. \ t_1 = t_4 = \frac{C^2}{2} = -t_2 = -t_3$$

Hence, for $C = 0.8$, (6.16) yields

$$t_5 = t_6 = t_7 = t_8 = 0 = \Lambda_2 = \Psi_2 = \Gamma_2 = \Upsilon_2 \quad (6.17)$$

$$t_1 = t_4 = -t_2 = -t_3 = \frac{C^2}{2} = \frac{8}{25} = \Lambda_1 = \Upsilon_1 = -\Psi_1 = -\Gamma_1$$

and the corresponding $[M]_e$ and $[U]_e$ matrices become

$$[M]_e = \Delta x_e ([A200L] - [APERT]) \quad (6.18)$$

$$= \frac{\Delta x_e}{6} \begin{bmatrix} 2 - \frac{C^2}{2} & 1 + \frac{C^2}{2} \\ 1 + \frac{C^2}{2} & 2 - \frac{C^2}{2} \end{bmatrix} = \frac{\Delta x_e}{150} \begin{bmatrix} 42 & 33 \\ 33 & 42 \end{bmatrix}$$

$$[U]_e = U ([A201L] - [APERX]) = U([A201L]) \quad (6.19)$$

$$= \frac{U}{2} \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}$$

Using (6.16), the algorithm amplification factor (6.8) is

$$G = 1 - i(Cm) - \frac{(Cm)^2}{2} + i\frac{(Cm)^3}{3!} + \frac{(Cm)^4}{4!} \quad (6.20)$$

$$\begin{aligned}
& -i \frac{\frac{1}{6}(5C^5 + 5C^3 - 4C)m^5}{5!} - \frac{(5C^4 - 4C^2)m^6}{6!} \\
& -i \frac{\frac{1}{12}(35C^7 - 245C^5 + 238C^3 - 40C)m^7}{7!} - \mathcal{O}(m^8)
\end{aligned}$$

One can readily see that (6.20) matches exactly with (1.11) up to the fourth order term ($\mathcal{O}(m^4)$). A similar result may be obtained by using the TWS (Baker and Kim, 1987) algorithm, which can also be fourth order accurate.

As a special case (ref. Section 4), selecting $C = 1$ yields $t_1 = t_4 = \frac{1}{2} = -t_2 = -t_3$ and $t_6 = t_8 = \frac{t_1 + t_2}{3} = 0 = -t_5 = -t_7$. Hence

$$[M]_e = \Delta x_e ([A200L] - [APERT]) = \frac{\Delta x_e}{4} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad (6.21)$$

$$[U]_e = U ([A201L] - [APERX]) = \frac{U}{2} \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \quad (6.22)$$

for which the amplification factor of the approximation solution for (1.10), in terms of wave number $m = \omega \Delta x$ is

$$\begin{aligned}
G &= 1 - im - \frac{m^2}{2} + i \frac{m^3}{3!} + \frac{m^4}{4!} - i \frac{m^5}{5!} - \frac{m^6}{6!} + i \frac{m^7}{7!} + \frac{m^8}{8!} - i \frac{m^9}{9!} - \frac{m^{10}}{10!} \\
&+ i \frac{m^{11}}{11!} + \frac{m^{12}}{12!} - i \frac{m^{13}}{13!} - \frac{m^{14}}{14!} + i \frac{m^{15}}{15!} + \frac{m^{16}}{16!} - i \frac{m^{17}}{17!} - \frac{m^{18}}{18!} + \dots \\
&= e^{-im} = G_a
\end{aligned} \quad (6.23)$$

which is exact (matched) with the analytical solution amplification factor g_a to *eighteenth* order.

For another example, selecting $C = 2$ yields $t_1 = t_4 = 2 = -t_2 = -t_3$ and $t_6 = t_8 = \frac{t_1 + t_2}{3} = 0 = -t_5 = -t_7$, hence

$$[M]_e = \Delta x_e ([A200L] - [APERT]) = \frac{\Delta x_e}{2} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad (6.24)$$

$$[U]_e = U([A201L] - [APERX]) = \frac{U}{2} \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}$$

Here, again, the solution amplification factor, in terms of wave number m , is

$$\begin{aligned} G &= 1 - 2im - 2m^2 + i\frac{4m^3}{3} + \frac{2m^4}{3} - i\frac{4m^5}{15} - \frac{4m^6}{45} + i\frac{8m^7}{315} + \frac{2m^8}{315} \\ &\quad - i\frac{4m^9}{2835} - \frac{4m^{10}}{14175} + i\frac{8m^{11}}{155925} + \frac{4m^{12}}{467775} - i\frac{8m^{13}}{6081075} - \frac{8m^{14}}{42567525} + \dots \\ &= e^{-2im} = G_a \end{aligned} \quad (6.25)$$

which is exact (matched) to *fourteenth* order with the analytical solution amplification factor G_a . However, in this case, one may not obtain the exact solution numerically using (6.24) (Appendix C). In general, a von Neumann frequency analysis result is a necessary but *not sufficient* condition to guarantee the approximate solution accuracy. Specifically, matching the coefficients of equal powers of m in (1.11) and (6.12) only guarantees that the ratio $C^h/C = 1$, but does not guarantee $\Phi^h = 1$.

The superscripts $\alpha, \beta, \gamma, \delta$ in (6.2)-(6.3) for [APERT] and [APERX] are related to the stencil node locations (spread) in the mesh. For example, $\gamma = 1$ yields $z = e^{-2im}$ in (6.11)-(6.12) which brings in the $(j - 2)$ node in the assembly stencil in (6.9). Similarly, $\beta = 1$ includes $(j + 2)$ node (see Appendix B). Therefore, the procedure

(6.2)-(6.3) results can be compared directly with compact finite difference schemes, *c.f.*, Lele (1992), although the theory and implimentation of GMP is much simpler. Also note that the GMP assembly matrices in (6.19)-(6.21) retain the efficiency of a linear basis (or centered FD) algorithm form.

6.4 Comparison with TWS Class of Algorithm

This class of algorithms has been widely verified and documented in the literature, *c.f.*, Baker and Kim (1987), Chaffin and Baker (1994), Donea (1984), Genuchten and Gray (1978). Assuming sufficient continuity in the solution, the forward time semi-discretized Taylor series is

$$q^{n+\vartheta} = q^n + \vartheta \Delta t \frac{\partial q^n}{\partial t} + \frac{\vartheta^2 \Delta t^2}{2} \frac{\partial^2 q^n}{\partial t^2} + \frac{\vartheta^3 \Delta t^3}{6} \frac{\partial^3 q^n}{\partial t^3} + \frac{\vartheta^4 \Delta t^4}{24} \frac{\partial^4 q^n}{\partial t^4} + \frac{\vartheta^5 \Delta t^5}{120} \frac{\partial^5 q^n}{\partial t^5} + \mathcal{O}(\Delta t^6) \quad (6.26)$$

where ϑ is an upwind level. The corresponding backward time semi-discrete Taylor series is

$$q^{n+(1-\vartheta)} = q^{n+1} - \vartheta \Delta t \frac{\partial q^{n+1}}{\partial t} + \frac{\vartheta^2 \Delta t^2}{2} \frac{\partial^2 q^{n+1}}{\partial t^2} - \frac{\vartheta^3 \Delta t^3}{6} \frac{\partial^3 q^{n+1}}{\partial t^3} + \frac{\vartheta^4 \Delta t^4}{24} \frac{\partial^4 q^{n+1}}{\partial t^4} - \frac{\vartheta^5 \Delta t^5}{120} \frac{\partial^5 q^{n+1}}{\partial t^5} + \mathcal{O}(\Delta t^6) \quad (6.27)$$

Using (6.26), for $\vartheta = 1/2$ and 1, Chaffin and Baker (1994) derived

$$q^{n+1} - q^n - \Delta t \frac{\partial q^n}{\partial t} = \frac{\Delta t^2}{6} \frac{\partial^2 q^n}{\partial t^2} - \frac{\Delta t^4}{144} \frac{\partial^4 q^n}{\partial t^4} - \frac{\Delta t^5}{480} \frac{\partial^5 q^n}{\partial t^5} + \mathcal{O}(\Delta t^6) \quad (6.28)$$

Similarly, for $\vartheta = 1/3$ and $2/3$ in (6.26), I found

$$q^{n+1} - q^n - \Delta t \frac{\partial q^n}{\partial t} = \frac{\Delta t^2}{9} \frac{\partial^2 q^n}{\partial t^2} - \frac{\Delta t^4}{486} \frac{\partial^4 q^n}{\partial t^4} - \frac{\Delta t^5}{2430} \frac{\partial^5 q^n}{\partial t^5} + \mathcal{O}(\Delta t^6) \quad (6.29)$$

Finally, a weighted combination of (6.28) and (6.29) yields

$$q^{n+1} - q^n - \Delta t \frac{\partial q^n}{\partial t} = \frac{5\Delta t^2}{57} \frac{\partial^2 q^n}{\partial t^2} + \frac{\Delta t^5}{10} \frac{\partial^5 q^n}{\partial t^5} + \mathcal{O}(\Delta t^6) \quad (6.30)$$

Similarly, using (6.27), for $\vartheta = 1/2$ and 1, Chaffin and Baker (1994) derived

$$q^{n+1} - q^n - \Delta t \frac{\partial q^{n+1}}{\partial t} = -\frac{\Delta t^2}{6} \frac{\partial^2 q^{n+1}}{\partial t^2} + \frac{\Delta t^4}{144} \frac{\partial^4 q^{n+1}}{\partial t^4} - \frac{\Delta t^5}{480} \frac{\partial^5 q^{n+1}}{\partial t^5} + \mathcal{O}(\Delta t^6) \quad (6.31)$$

Again, for $\vartheta = 1/3$ and $2/3$ in (6.27), I found

$$q^{n+1} - q^n - \Delta t \frac{\partial q^{n+1}}{\partial t} = -\frac{\Delta t^2}{9} \frac{\partial^2 q^{n+1}}{\partial t^2} + \frac{\Delta t^4}{486} \frac{\partial^4 q^{n+1}}{\partial t^4} - \frac{\Delta t^5}{2430} \frac{\partial^5 q^{n+1}}{\partial t^5} + \mathcal{O}(\Delta t^6) \quad (6.32)$$

and a weighted combination of (6.31) and (6.32) yields

$$q^{n+1} - q^n - \Delta t \frac{\partial q^{n+1}}{\partial t} = -\frac{5\Delta t^2}{57} \frac{\partial^2 q^{n+1}}{\partial t^2} - \frac{\Delta t^5}{10} \frac{\partial^5 q^{n+1}}{\partial t^5} + \mathcal{O}(\Delta t^6) \quad (6.33)$$

Finally, multiplying (6.30) by $(1 - \theta)$ and (6.33) by θ and then adding them together (since $q^{n+\theta} \approx \theta q^{n+1} + (1 - \theta)q^n$), I get

$$\begin{aligned} q^{n+1} - q^n - \Delta t \left(\theta \frac{\partial q^{n+1}}{\partial t} + (1 - \theta) \frac{\partial q^n}{\partial t} \right) &= -\frac{5\Delta t^2}{57} \left(\theta \frac{\partial^2 q^{n+1}}{\partial t^2} - (1 - \theta) \frac{\partial^2 q^n}{\partial t^2} \right) \\ &\quad - \frac{\Delta t^5}{10} \left(\theta \frac{\partial^5 q^{n+1}}{\partial t^5} - (1 - \theta) \frac{\partial^5 q^n}{\partial t^5} \right) + \mathcal{O}(\Delta t^6) \end{aligned} \quad (6.34)$$

Via (4.2) in one dimension, one may write

$$\frac{\partial q}{\partial t} = - \left(U - \epsilon \frac{\partial}{\partial x} \right) \frac{\partial q}{\partial x} \quad (6.35)$$

$$\frac{\partial^2 q}{\partial t^2} = \frac{\partial}{\partial x} \left(U - \epsilon \frac{\partial}{\partial x} \right) \left(U - \epsilon \frac{\partial}{\partial x} \right) \frac{\partial q}{\partial x}$$

Hence, from (6.34) via (6.35)

$$\begin{aligned} q^{n+1} - q^n - \Delta t \left(U - \epsilon \frac{\partial}{\partial x} \right) \frac{\partial}{\partial x} (\theta q^{n+1} + (1 - \theta)q^n) \\ = -\frac{5\Delta t^2}{57} \left(U^2 - 2U\epsilon \frac{\partial}{\partial x} \right) \frac{\partial^2}{\partial x^2} (\theta q^{n+1} - (1 - \theta)q^n) + \mathcal{O}(\Delta t^5, \epsilon^2) \end{aligned} \quad (6.36)$$

Note that (6.36) is *naturally* $\mathcal{O}(\Delta t^5, \epsilon^2)$, i.e. *fifth-order* accurate in time.

Substituting different values of implicitness $0 \leq \theta \leq 1$ in (6.36) results in the following relations:

$$\theta = 0 : \quad (6.37)$$

$$q^{n+1} - q^n - \Delta t \left(U - \epsilon \frac{\partial}{\partial x} \right) \frac{\partial}{\partial x} q^n - \frac{5\Delta t^2}{57} \left(U^2 - 2U\epsilon \frac{\partial}{\partial x} \right) \frac{\partial^2}{\partial x^2} q^n = \mathcal{O}(\Delta t^5, \epsilon^2)$$

$$\theta = 1/2 :$$

$$\begin{aligned} q^{n+1} - q^n - \frac{\Delta t}{2} \left(U - \epsilon \frac{\partial}{\partial x} \right) \frac{\partial}{\partial x} (q^{n+1} + q^n) \\ + \frac{5\Delta t^2}{57} \left(U^2 - 2U\epsilon \frac{\partial}{\partial x} \right) \frac{\partial^2}{\partial x^2} (q^{n+1} - q^n) = \mathcal{O}(\Delta t^5, \epsilon^2) \end{aligned}$$

$$\theta = 1 :$$

$$q^{n+1} - q^n - \Delta t \left(U - \epsilon \frac{\partial}{\partial x} \right) \frac{\partial}{\partial x} q^{n+1} + \frac{5\Delta t^2}{57} \left(U^2 - 2U\epsilon \frac{\partial}{\partial x} \right) \frac{\partial^2}{\partial x^2} q^{n+1} = \mathcal{O}(\Delta t^5, \epsilon^2)$$

For trapezoidal condition ($\theta = 1/2$), Chaffin and Baker (1994) found

$$\begin{aligned} q^{n+1} - q^n - \frac{\Delta t}{2} \left(U - \epsilon \frac{\partial}{\partial x} \right) \frac{\partial}{\partial x} (q^{n+1} + q^n) \\ + \frac{\Delta t^2}{12} \left(U^2 - 2U\epsilon \frac{\partial}{\partial x} \right) \frac{\partial^2}{\partial x^2} (q^{n+1} - q^n) = \mathcal{O}(\Delta t^4, \epsilon^2) \end{aligned} \quad (6.38)$$

Defining $\mathbf{E}$ as

$$\mathbf{E} = \int_{\Omega} \frac{\partial \phi_j}{\partial x} \frac{\partial}{\partial x} \frac{\partial \phi_i}{\partial x} d\Omega \quad (6.39)$$

via (3.12) and (3.13), (6.37) and (6.38) may be written in a general form, Baker and Kim (1987)

$$(\mathbf{M} + (\theta - \alpha)(C\Delta x)\mathbf{U} + (\gamma + \beta\theta)(C\Delta x)^2\mathbf{D} + \mu\theta(C\Delta x)^3\mathbf{E}) Q^{n+1} = \quad (6.40)$$

$$(\mathbf{M} - (1 - \theta + \alpha)(C\Delta x)\mathbf{U} + (\gamma - (1 - \theta)\beta)(C\Delta x)^2\mathbf{D} - \mu(1 - \theta)(C\Delta x)^3\mathbf{E}) Q^n$$

where for (6.38), $\alpha = 0$, $\beta = \epsilon/(U^2\Delta t) = \epsilon$ (non-dimensional diffusion coefficient), $\gamma = -1/12$ and $\mu = \pm\epsilon/3$. Also, for (6.37) and $\theta = 1/2$, $\alpha = 0$, $\beta = \epsilon$, $\gamma = 5/57$ and so on. Heinrich and Zienkiewicz (1978) also proposed a similar Bubnov-Galerkin scheme with α and β as free perturbation parameters. Layton (1990) implemented this in a finite difference scheme for two dimensional transport problems.

In comparison the corresponding GMP algorithm general form can be written as

$$\begin{aligned} & ((\mathbf{M} - \mathbf{M}^\epsilon) + \theta [(C\Delta x)(\mathbf{U} - \mathbf{U}^\epsilon) + (C\Delta x)^2(\mathbf{D} - \mathbf{D}^\epsilon) + (C\Delta x)^3(\mathbf{E} - \mathbf{E}^\epsilon)]) Q^{n+1} \\ &= ((\mathbf{M} - \mathbf{M}^\epsilon) - (1 - \theta) [(C\Delta x)(\mathbf{U} - \mathbf{U}^\epsilon) - (C\Delta x)^2(\mathbf{D} - \mathbf{D}^\epsilon) \\ &\quad - (C\Delta x)^3(\mathbf{E} - \mathbf{E}^\epsilon)]) Q^n \end{aligned} \quad (6.41)$$

where upon full discretization

$$\mathbf{M} \Rightarrow [M], \quad \text{and} \quad \mathbf{M}^\epsilon \Rightarrow [\text{APERT}] \quad (6.42)$$

$$\mathbf{U} \Rightarrow [U], \quad \text{and} \quad \mathbf{U}^\epsilon \Rightarrow [\text{APERX}]$$

$$\mathbf{D} \Rightarrow [D], \quad \text{and} \quad \mathbf{D}^\epsilon \Rightarrow [\text{APERD}]$$

$$\mathbf{E} \Rightarrow [E], \quad \text{and} \quad \mathbf{E}^\epsilon \Rightarrow [\text{APERE}]$$

One may see that for no physical diffusion, i.e. $\epsilon = 0$, (6.40) is identical with GMP algorithm for $[\text{APERT}] = 0$ and $[\text{APERX}] C\Delta x (\theta\{Q\}^{n+1} + (1 - \theta)\{Q\}^n) = \alpha[U](Q^{n+1} - Q^n)$ in (6.3)-(6.2).

6.5 Condition for Time Accurate Solution

The necessary condition for the approximation solution of (1.1) to agree exactly with the continuum solution (1.2) is certainly

$$G \equiv G_a \quad (6.43)$$

Hence, the formulation (theoretical) goal is for equations (1.10) and (6.1) to match to extremum order. The general rational form of (6.1)-(6.2) is

$$G = \frac{a_1 e^{-im} + a_3 e^{im} + a_2}{b_1 e^{-im} + b_3 e^{im} + b_2} = \frac{a_1 + a_3 e^{2im} + a_2 e^{im}}{b_1 + b_3 e^{2im} + b_2 e^{im}} \quad (6.44)$$

where a_i and b_i , $1 \leq i \leq 3$, are specific CFD algorithm coefficients. From (1.6) and (3.20), one can write (Mulay, 1993),

$$\begin{aligned} & (a_1 + a_2 e^{im} + a_3 e^{2im}) e^{imC} - (b_1 + b_2 e^{im} + b_3 e^{2im}) \\ &= a_1 \sum_{n=0}^{\infty} \left(\frac{i^n}{n!} C^n \right) m^n + a_2 \sum_{n=0}^{\infty} \left(\frac{i^n}{n!} (C+1)^n \right) m^n + a_3 \sum_{n=0}^{\infty} \left(\frac{i^n}{n!} (C+2)^n \right) m^n - \\ & \quad b_1 - b_2 \sum_{n=0}^{\infty} \left(\frac{i^n}{n!} \right) m^n - b_3 \sum_{n=0}^{\infty} \left(\frac{i^n}{n!} 2^n \right) m^n \\ &= (a_1 + a_2 + a_3 - b_1 - b_2 - b_3) \\ & \quad + \sum_{n=0}^{\infty} \frac{i^n}{n!} [a_1 C^n + a_2 (C+1)^n + a_3 (C+2)^n - b_2 - 2^n b_3] \\ &= 0 \end{aligned} \quad (6.45)$$

i.e.

$$\det \begin{vmatrix} -1 & -2 & C & (C+1) & (C+2) \\ -1 & -2^2 & C^2 & (C+1)^2 & (C+2)^2 \\ -1 & -2^3 & C^3 & (C+1)^3 & (C+2)^3 \\ -1 & -2^4 & C^4 & (C+1)^4 & (C+2)^4 \\ -1 & -2^5 & C^5 & (C+1)^5 & (C+2)^5 \end{vmatrix} \quad (6.46)$$

$$= 4C^3(C+1)^2(C+2)(C-1)^2(C-2) = 0$$

The condition (6.46) yields the following roots for C

$$C = 0, \pm 1, \pm 2 \quad (6.47)$$

6.6 Direct Time Integration

For a time integration ranging from t to $t + n_e \Delta t_e$, where the subscript 'e' denotes a specific element, the weak statement algorithm (3.20) yields the following θ -implicit relation

$$S_e n_e (([M]_e + \theta \Delta t_e [U]_e) \{ \Delta Q \}_e - \Delta t_e (\{ b \}_e - [U]_e \{ Q^{n_e} \}_e)) = \{ 0 \} \quad (6.48)$$

where

$$\{ \Delta Q \}_e = \{ Q^{n_e+1} - Q^{n_e} \}_e, \quad n_e = \frac{T}{\Delta t_e} \quad \text{and} \quad \Delta t_e = \frac{C_e \Delta x_e}{|U_e|} \quad (6.49)$$

The condition for (6.48) to be true is

$$C_e = \frac{T|U_e|}{\Delta x_e} \quad (6.50)$$

Consequently, the weak statement (3.20)-(3.25) modified to total time T is

$$S_e \left(\left(\frac{T|U_e|}{C_e \Delta x_e} \Delta x_e [M] + \theta (n_e \Delta t_e) U_e [U] \right) \{ \Delta Q \}_e - (n_e \Delta t_e) (\{b\}_e - U_e [U] \{Q^{n_e}\}_e) \right) = \{0\} \quad (6.51)$$

Hence, for element specific C_e , taking the continuum total time T outside the element matrix assembly operator S_e , one may derive

$$S_e \left(\left(\frac{|U_e|}{C_e} [M] + \theta U_e [U] \right) \{ \Delta Q \}_e - (\{b\}_e - U_e [U] \{Q^{n_e}\}_e) \right) = \{0\} \quad (6.52)$$

The multiplier coefficient of the mass matrix $[M]$ is always positive (and equal to $\frac{|U_e|}{C_e}$) while for the convection matrix $[U]$ the multiplier is strictly U_e .

Then, for $C_e = 1$, for example, (6.52) in one-dimension becomes

$$S_e \left((|U_e| [A200] + \theta U_e [A201]) \{ \Delta Q \}_e - (\{b\}_e - U_e [A201] \{Q^{n_e}\}_e) \right) = \{0\} \quad (6.53)$$

Using (6.53), one can enforce that the element specific Courant number $C_e = 1$, and hence, the approximation solution $Q(n_e \Delta t_e, j \Delta x_e)$ from (3.20) becomes exactly equal to the analytical solution q in (1.2) at any time T .

If $C \neq 1$, then one may derive in one dimension

$$S_e \left(\left(\frac{|U_e|}{C_e} [A200] + \theta U_e [A201] \right) \{ \Delta Q \}_e - (\{b\}_e - U_e [A201] \{Q^{n_e}\}_e) \right) = \{0\} \quad (6.54)$$

It is direct to see the importance of (6.52)-(6.54) as the solution of this modified weak statement form is explicitly mesh independent. The implicit mesh dependence however plays a role in interpolating U_e . Also, if one is interested in finding the exact solution $Q = q$, the condition (6.50) yields

$$\Delta x_e = T |U_e| \quad (6.55)$$

which is a mesh generation constraint.

6.7 Multidimensional forms

6.7.1 Tensor product factorization algorithm:

The fluid convection contribution to (1.1) can be resolved into components parallel to the global coordinate system x_i . The multi-dimensional form of (3.20) is (Baker and Pepper, 1991)

$$S_e (([M]_e + \theta \Delta t_e [U]_e) \{ \Delta Q \}_e + \Delta t_e ([U]_e \{ Q^{n_e} \}_e)) = \{0\} \quad (6.56)$$

where boldface U_e indicates a vector field. Upon assembly over all Ω_e of Ω^h , (6.4) yields a system matrix $[S]$ which can be approximated for $d = 2$ as

$$[S] \equiv S_e ([M]_e + \theta \Delta t_e [U]_e) \approx [SX] \otimes [SY] \quad (6.57)$$

where the symbol $\otimes$ denotes the outer (*tensor*) matrix product. The product $[SX] \otimes [SY]$ in (6.57) is inexact with truncation error of order $(\theta \Delta t^2)$, which is the same order as the forward ($\theta = 0$) or backward ($\theta = 1$) Euler integration procedures in (1.10).

The definitions for forming (6.56) are

$$\begin{aligned} [SX] &\equiv S_e ([SX]_e) = S_e ([MX]_e + \theta \Delta t_e [UX]_e) \\ [SY] &\equiv S_e ([SY]_e) = S_e ([MY]_e + \theta \Delta t_e [UY]_e) \end{aligned} \quad (6.58)$$

From (6.57), via the modified weak statement (6.52)-(6.53), one may derive that for $C_e = 1$

$$\begin{aligned} n_e S_e ([SX]_e) &= T S_e (|U_e| [A200] + \theta U_e [A201]) \\ n_e S_e ([SY]_e) &= T S_e (|V_e| [A200] + \theta V_e [A201]) \end{aligned} \quad (6.59)$$

where U_e and V_e are element dependent velocity components in the x and y directions respectively.

The corresponding quasi-Newton approximate linear algebra solution procedure for (6.56) becomes the two-step sequence

$$[SX]\{P\} = -\Delta t\{R\}$$

$$[SY]\{\Delta Q\} = \{P^T\} \quad (6.60)$$

where $\{P\}$ is an intermediate array of data and $\{P^T\}$ is its row/column permutation.

6.7.2 The time-splitting (TS) algorithm:

In addition to (6.57), the residual in (6.60) may also be approximately resolved into directional components

$$\{R\} \approx \{RX\} \otimes \{RY\} = S_e(\{RX\}_e + \{RY\}_e) \quad (6.61)$$

$$\text{where} \quad \{RX\}_e = U_e[A201] \quad \text{and} \quad \{RY\}_e = V_e[A201] \quad (6.62)$$

The insertion of (6.61) into (6.60) is termed "time-splitting," and in two-dimensions the TS formulation is operationally

$$S_e([SX]_e)\{P\} = -\frac{1}{2}S_e(\{RX(Q)^{n_e}\}_e) \quad (6.63)$$

$$\{Q\}^{n_e+1/2} \equiv \{Q\}^{n_e} + \{P\}$$

$$S_e([SY]_e)\{\Delta Q\} = -\frac{1}{2}S_e(\{RY(Q)^{n_e+1/2}\}_e)$$

$$\{Q\}^{n_e+1} \equiv \{Q\}^{n_e+1/2} + \{\Delta Q\}$$

where $\{Q\}^{n_e+1/2}$ is interpreted as an intermediate solution approximation.

THE RECONSTRUCTION (RCN) ALGORITHM

The reconstruction (RCN) algorithm is developed as a post-processing operation applicable to *any* CFD solution, which would be an iteration of a non-linear problem specification or one with source. One problem class of specific interest for (2.1) is where q is an inert species mass fraction (Y_A), whereupon (2.1) can be written in the “Reynolds-averaged” PDE form in d dimensions as

$$\mathcal{L}(Y_A) = \frac{\partial Y_A}{\partial t} + \frac{\partial}{\partial x_j} \left(u_j Y_A - \frac{1}{\text{Re}} \left(\frac{1}{\text{Sc}} + \frac{\text{Re}^t}{\text{Sc}^t} \right) \frac{\partial Y_A}{\partial x_j} \right) - s_A = 0 \quad \text{on } \Omega \times t \subset \mathbb{R}^3 \times \mathbb{R}^+ \quad (7.1)$$

Hence, ϵ in (2.1) is associated with the dimensionless Reynolds number (Re), the binary diffusion Schmidt number (Sc), and their modeled “turbulent” counterparts turbulent Reynolds number (Re^t) and turbulent Schmidt number (Sc^t). Typically, Re and Re^t are set via the flowfield specification that produces the d -dimensional velocity field $u^h(x_j, t)$. Thereafter, solutions to (7.1) are parameterized via Sc (nominally equal to Prandtl number) and $\text{Sc}^t \leq \text{Sc}$, and the distribution of the source s_A . For smooth data, the analytical solution to (7.1) is smooth, monotone and bounded, but a CFD spatial semi-discretized computational procedure invariably leads to an oscillatory, *i.e.*, non-monotone error mode in the approximate solution.

The RCN theory utilizes Sobolev norm conservation principles, specifically, the computation and storage of the distribution and moments of the initial condition $q_0(\mathbf{x})$, as discretized on Ω^h , in several vector norms $l_{\mathcal{R}}$. A general class of $l_{\mathcal{R}}$ discrete norms is

$$l_{\mathcal{R}} \equiv \|Q\|_{\mathcal{R}} = \left| \left(\sum_{i=1}^N Q_i^{\mathcal{R}} \right)^{1/\mathcal{R}} \right| \quad (7.2)$$

for integer $\mathcal{R} \in [1, \infty)$, and N is the number of degrees of freedom in the system. For $\mathcal{R} \rightarrow \infty$, $l_{\mathcal{R}} \Rightarrow l_{\infty} \equiv \max (Q_i)$ for $1 \leq i \leq N$.

Additionally, the RCN procedure may require estimates of the first and second order spatial derivative distributions of the discrete CFD solution, for computing local monotone non-zero distributions and moments. Then, the discrete solution dispersion error modes may be damped using simple forms of scalar weighted averaging, *e.g.*,

for $2\Delta x$ waves :

$$\text{if } q_{i-1} > q_i < q_{i+1}, \text{ or } q_{i-1} < q_i > q_{i+1}, \text{ then } q_i \equiv \frac{q_{i+1} + q_{i-1}}{2}$$

for $4\Delta x$ waves :

$$\text{if } q_{i-2} > q_i < q_{i+2}, \text{ or } q_{i-2} < q_i > q_{i+2}, \text{ then } q_i \equiv \frac{q_{i+2} + q_{i+1} + q_{i-1} + q_{i-2}}{4} \quad (7.3)$$

Via (7.3), the following central difference formulae may be used to approximate discrete derivatives.

$$\begin{aligned}
\frac{dq^h}{dx} &= \frac{Q_{i+1} - Q_{i-1}}{2\Delta x} + \mathcal{O}(\Delta x)^2 \quad \cong \quad \frac{Q_{i+2} - Q_{i-2}}{4\Delta x} \\
\frac{d^2q^h}{dx^2} &= \frac{Q_{i-1} - 2Q_i + Q_{i+1}}{(\Delta x)^2} + \mathcal{O}(\Delta x)^2 \quad \cong \quad \frac{Q_{i-2} - 2Q_i + Q_{i+2}}{2(\Delta x)^2} \\
\frac{dq^h}{dy} &= \frac{Q_{i+1} - Q_{i-1}}{2\Delta y} + \mathcal{O}(\Delta y)^2 \quad \cong \quad \frac{Q_{i+2} - Q_{i-2}}{4\Delta y} \\
\frac{d^2q^h}{dy^2} &= \frac{Q_{i-1} - 2Q_i + Q_{i+1}}{(\Delta y)^2} + \mathcal{O}(\Delta y)^2 \quad \cong \quad \frac{Q_{i-2} - 2Q_i + Q_{i+2}}{2(\Delta y)^2} \\
\frac{dq^h}{dz} &= \frac{Q_{i+1} - Q_{i-1}}{2\Delta z} + \mathcal{O}(\Delta z)^2 \quad \cong \quad \frac{Q_{i+2} - Q_{i-2}}{4\Delta z} \\
\frac{d^2q^h}{dz^2} &= \frac{Q_{i-1} - 2Q_i + Q_{i+1}}{(\Delta z)^2} + \mathcal{O}(\Delta z)^2 \quad \cong \quad \frac{Q_{i-2} - 2Q_i + Q_{i+2}}{2(\Delta z)^2}
\end{aligned} \tag{7.4}$$

Note that first order derivatives do not change sign when the solution is monotone. Hence, care must be exercised to employ (7.3)-(7.4) only in the region containing dispersive error waves, not where the actual solution peak may exist. These issues are detailed later.

For the linear problem statement, γ_1 is defined as the reconstruction ratio of the sum of the nodal solution at any discrete time $t^n = t_0 + n\Delta t$ and the initial distribution. Hence,

$$\gamma_1 = \frac{\left| \sum_{j=1}^N Q_j^{t^n} \right|}{\left| \sum_{j=1}^N Q_j^t \right|} \equiv \frac{S^n}{S^0} \tag{7.5}$$

and N remains the number of nodes in Ω^h . For a conservative system, $\gamma_1 \sim 1$, while for a dissipative system $\gamma_1 < 1$. The presence of a source in the computational domain will produce $\gamma_1 > 1$.

For a non-linear problem, a second reconstruction ratio (γ_2) is required between the reconstructive iterations. The definition is

$$\gamma_2 = \frac{\left| \sum_{j=1}^M Q_{jR}^{t^n} \right|}{\left| \sum_{j=1}^N Q_j^{t^n} \right|} \equiv \frac{S_R^n}{S^n} \quad (7.6)$$

where the subscript R denotes the reconstructive iteration stage. However, the l_1 norm (algebraic sum) of the nodal solution Q_j at the final time $(t_0 + n\Delta t)$ does not change due to reconstruction. Thus, the solution data Q_{jR} after each iteration must be divided by γ_2 .

7.1 Linear RCN

For a linear problem statement, the RCN algorithm proceeds as follows:

Step 1. Characterize the distribution and moment of the initial condition by calculating the vector norms $l_1 = S^0 = \left| \sum_{j=1}^N Q_j^0 \right|$ and $l_\infty = |\max(Q_j)|$. Store the nonzero initial nodal distribution Q_j^0 and reference to the node(s) where l_∞ is located.

Step 2. For *any* CFD discrete solution, at time level $t_0 + n\Delta t$, labelled Q_j^n , determine the global *maxima*, i.e., l_∞ norm.

Step 3. Calculate the sum norm l_1 as $S^n = \left| \sum_{j=1}^N Q_j^n \right|$, hence form the reconstruction ratio, $\gamma_1 = \frac{S^n}{S^0}$.

Step 4. Relocate the initial condition fraction $\gamma_1 Q_j^0$ to the node(s) where the l_∞ norm of Q_j^n is located.

7.2 Non-linear RCN

For a non-linear problem statement, or in the presence of source s_A in (1.2), the RCN algorithm proceeds as follows:

Step 1. Label *any* CFD discrete solution at time $t_0 + n\Delta t$ as $Q_j^n, 1 \leq j \leq N$.

Step 2. Determine the l_1, l_2 and l_∞ norms of Q_j^n .

Step 3. Separate the solution domain Ω into the union of three nonoverlapping subdomains such that

$$(i) \Omega = \Omega^> \cup \Omega^\epsilon \cup \Omega^< \text{ and}$$

$$(ii) Q_{j_1}^> \subset \Omega^>, Q_{j_2}^\epsilon \subset \Omega^\epsilon, \text{ and } Q_{j_3}^< \subset \Omega^<,$$

where $Q_{j_1}^>$ denotes the nodal values of Q_j^n greater than a “close” approximation to the base value, *e.g.* $(b + \epsilon)$, $b = 0$ in general and $\epsilon \approx 0.05 \|Q\|_{l_\infty}$. Also, $Q_{j_2}^\epsilon$ denotes nodal values of Q_j^n for which $|Q_{j_2}| \leq \epsilon$, and $Q_{j_3}^<$ is associated with $Q_{j_3} < (b - \epsilon)$.

Step 4. Eliminate the dispersion error waves in Ω^ϵ and $\Omega^<$ via (7.3)-(7.4).

Step 5. Calculate the reconstruction ratio γ_2 via (7.6), and divide the reconstructed solution from Step 4 by γ_2 . Then redetermine the reconstructed solution $l_{\mathcal{R}_R}$ norms, and iterate between step 3, 4 and 5 until l_{1_R} exactly matches with the CFD solution l_1 norm.

7.3 RCN Fortran Sample

```
c
c   this reconstruction algorithm RCN is written by subrata roy,
c       esm, ut, knoxville, 4/01/93
c
c   for scalar dependent variable q of a non-linear / source problem
c   this is a code for reading n variables of size m in rows
c   and writing them in n columns of length m
c   ndim is the dimension of the solution domain
c   this program will print x, y and z coordinates
c
c   program vgr
c   implicit real*8 (a-h,o-z)
c   parameter (n=20,m=200)
c   dimension varbl(7,0:49,0:49,0:49),v(8,20000),mel(8,12000)
c   dimension sum(8),varmax(8),varmin(8),nmax(5)
c   dimension shape(5,49,49),ishape(49,49),nshape(49)
c   dimension gamma(5),store(20000)
c
c   open(unit=2,file='3drec.in',status='old')
c   open(unit=6,file='3drec.out',status='unknown')
c   open(unit=7,file='3drec.dat',status='unknown')
c
c   read(2,*)nvar,n1,n2,n3
c   for reading from the tecplot file
c   read(2,*)nnodes,neles
c
c   ndim=3
c   do 10 i=1,nvar
10  read(2,*)(v(i,l2),l2=1,nnodes)
c   read(2,31)((mel(i,j),i=1,8),j=1,neles)
c   l1=0
c   varmax(4)=0.d0
c   varmin(4)=0.d0
c   varmax(8)=0.d0
c   varmin(8)=0.d0
c   sum(4)=0.d0
c   sum(7)=0.d0
```

```

sum(8)=0.d0
do 11 k3=1,n3
do 11 k2=1,n2
do 11 k1=1,n1
varbl(4,0,0,0)=0.d0
varbl(4,k3,0,0)=0.d0
varbl(4,0,k2,0)=0.d0
varbl(4,0,0,k1)=0.d0
varbl(4,k3,k2,0)=0.d0
varbl(4,0,k2,k1)=0.d0
varbl(4,k3,0,k1)=0.d0
varbl(4,n3+1,n2+1,n1+1)=0.d0
varbl(4,k3,n2+1,n1+1)=0.d0
varbl(4,n3+1,k2,n1+1)=0.d0
varbl(4,n3+1,n2+1,k1)=0.d0
varbl(4,k3,k2,n1+1)=0.d0
varbl(4,n3+1,k2,k1)=0.d0
varbl(4,k3,n2+1,k1)=0.d0
l1=l1+1
c    if(v(4,l1).gt.100.d0)v(4,l1)=100.d0
store(l1)=0.d0
if(abs(v(4,l1)).ge.5.d0) store(l1)=v(4,l1)
varmax(4)=max(v(4,l1),varmax(4))
varmin(4)=min(v(4,l1),varmin(4))
sum(4)=sum(4)+v(4,l1)
11  varbl(4,k3,k2,k1)=v(4,l1)
l1=0

c    RCN procedure
c    4th variable is species

c    do 60 i=1,nnodes
c 60  if(store(i).ne.0.d0)v(4,i)=abs(v(4,i))

l1=0
c    wiping out 2deltax error wave
c    x direction
do 62 k3=1,n3
do 62 k2=1,n2
do 62 k1=2,n1-1

```

```

        l1=l1+1
        if(store(l1).ne.0.d0) then
            if(varbl(4,k3,k2,k1).lt.varbl(4,k3,k2,k1+1) .and.
+varbl(4,k3,k2,k1).lt.varbl(4,k3,k2,k1-1))
+varbl(4,k3,k2,k1)=(varbl(4,k3,k2,k1+1)+varbl(4,k3,k2,k1-1))/2.d0
            endif
62      continue
c      y direction
        l1=0
        do 63 k3=1,n3
        do 63 k2=2,n2-1
        do 63 k1=1,n1
            l1=l1+1
            if(store(l1).ne.0.d0) then
                if(varbl(4,k3,k2,k1).lt.varbl(4,k3,k2+1,k1) .and.
+varbl(4,k3,k2,k1).lt.varbl(4,k3,k2-1,k1))
+varbl(4,k3,k2,k1)=(varbl(4,k3,k2-1,k1)+varbl(4,k3,k2+1,k1))/2.d0
                endif
63      continue
c      z direction
        l1=0
        do 64 k3=2,n3-1
        do 64 k2=1,n2
        do 64 k1=1,n1
            l1=l1+1
            if(store(l1).ne.0.d0) then
                if(varbl(4,k3,k2,k1).lt.varbl(4,k3+1,k2,k1) .and.
+varbl(4,k3,k2,k1).lt.varbl(4,k3-1,k2,k1))
+varbl(4,k3,k2,k1)=(varbl(4,k3-1,k2,k1)+varbl(4,k3+1,k2,k1))/2.d0
                endif
64      continue

        l1=0
        do 65 k3=1,n3
        do 65 k2=1,n2
        do 65 k1=3,n1-2
            l1=l1+1
            if(store(l1).ne.0.d0) then
c      wiping out 4deltax error wave
c      x direction

```

```

        if((varbl(4,k3,k2,k1).lt.varbl(4,k3,k2,k1+1) .or.
+ varbl(4,k3,k2,k1).lt.varbl(4,k3,k2,k1+2)) .and.
+(varbl(4,k3,k2,k1).lt.varbl(4,k3,k2,k1-1) .or.
+ varbl(4,k3,k2,k1).lt.varbl(4,k3,k2,k1-2)))
+varbl(4,k3,k2,k1)=(varbl(4,k3,k2,k1+1)+varbl(4,k3,k2,k1-1)+
+varbl(4,k3,k2,k1+2)+varbl(4,k3,k2,k1-2))/4.d0
        endif
65      continue
        l1=0
        do 66 k3=1,n3
        do 66 k2=3,n2-2
        do 66 k1=1,n1
        l1=l1+1
        if(store(l1).ne.0.d0) then
c      y direction
        if((varbl(4,k3,k2,k1).lt.varbl(4,k3,k2+1,k1) .or.
+ varbl(4,k3,k2,k1).lt.varbl(4,k3,k2+2,k1)) .and.
+(varbl(4,k3,k2,k1).lt.varbl(4,k3,k2-1,k1) .or.
+ varbl(4,k3,k2,k1).lt.varbl(4,k3,k2-2,k1)))
+varbl(4,k3,k2,k1)=(varbl(4,k3,k2+1,k1)+varbl(4,k3,k2-1,k1)+
+varbl(4,k3,k2+2,k1)+varbl(4,k3,k2-2,k1))/4.d0
        endif
66      continue
        l1=0
        do 67 k3=3,n3-2
        do 67 k2=1,n2
        do 67 k1=1,n1
        l1=l1+1
        if(store(l1).ne.0.d0) then
c      z direction
        if((varbl(4,k3,k2,k1).lt.varbl(4,k3+1,k2,k1) .or.
+ varbl(4,k3,k2,k1).lt.varbl(4,k3+2,k2,k1)) .and.
+(varbl(4,k3,k2,k1).lt.varbl(4,k3-1,k2,k1) .or.
+ varbl(4,k3,k2,k1).lt.varbl(4,k3-2,k2,k1)))
+varbl(4,k3,k2,k1)=(varbl(4,k3+1,k2,k1)+varbl(4,k3-1,k2,k1)+
+varbl(4,k3+2,k2,k1)+varbl(4,k3-2,k2,k1))/4.d0
        endif
67      continue

```

:


```

      :
      :
      :
      :
      :

do 22 k3=1,n3
do 22 k2=1,n2
do 22 k1=1,n1
sum(7)=sum(7)+varbl(4,k3,k2,k1)
22  if(varbl(4,k3,k2,k1).lt.0.d0)varbl(4,k3,k2,k1)=0.d0
gamma(4)=sum(4)/sum(7)
c    you can change the output format as you like
l1=0
do 20 k3=1,n3
do 20 k2=1,n2
do 20 k1=1,n1
l1=l1+1
v(8,l1)=v(4,l1)
c    if(varbl(4,k3,k2,k1).gt.100.d0)varbl(4,k3,k2,k1)=100.d0
if(store(l1).eq.0.d0)v(8,l1)=0.d0
c    k3 = n3 is dirichlet plane
c 20  if(k3.eq.n3)v(8,l1)=varbl(4,k3,k2,k1)
20  if(k3.ne.n3 .and. store(l1).ne.0.d0)v(8,l1)=
*varbl(4,k3,k2,k1)*gamma(4)

do 23 i=1,nnodes
23  sum(8)=sum(8)+v(8,i)
gamma(5)=sum(4)/sum(8)

do 21 i=1,nnodes
v(8,i)=v(8,i)*gamma(5)
if(v(8,i).gt.100.d0)v(8,i)=100.d0
if(v(8,i).lt.0.d0)v(8,i)=0.d0
varmax(8)=max(varmax(8),v(8,i))
21  varmin(8)=min(varmin(8),v(8,i))
write(7,*)' FE solution max and min  :'
write(7,30)varmax(4),varmin(4)
write(7,*)' FE solution SIGMA_Q  :'
write(7,30) sum(4)

```

```

        write(7,*)' RCN solution max and min : '
        write(7,30)varmax(8),varmin(8)
        write(7,*)' RCN solutions SIGMA_Q  :'
        write(7,30) sum(7),sum(8)
        write(7,*)' Gammas : '
        write(7,30)gamma(4),gamma(5)
        write(6,31)nnodes,neles
        do 25 i=1,nvar+1
25      write(6,30)(v(i,lk),lk=1,nnodes)
c      write(6,30)(v(8,lk),lk=1,nnodes)
        close(2)
        close(6)
        close(7)

30      format(1p,6e13.5)
31      format(8i6)
32      format(14i5)

        stop
        end

c*****
c  end - of - program
c*****

```

RESULTS AND DISCUSSION

8.1 Synopsis

Computational experiments designed for investigating a specific algorithm are divided into three basic categories, *namely, verification, benchmarking and validation*. The intent is to quantitatively characterize the accuracy, monotonicity and limitations of the selected algorithm implementation. Verification experiments constitute generation of computational data for problem for which a closed-form solution exists. The comparison of model problem results to those produced by independent computational models of accepted-accuracy, *i.e.*, code-to-code results comparisons, are called benchmarking experiments. Validation of an algorithm and code involves comparison of computational simulations to genuine geometry experimental data. The results from all three classes of test cases are presented in this chapter, with the intent of exploring the accuracy, convergence and stability of the SGM, GMP and RCN algorithms described in Chapters 5, 6 and 7.

Documentary computational results presented herein verify theory and summarize performance for a range of elliptic, parabolic and hyperbolic verification and benchmark problems that model the character of solutions to the Navier-Stokes

equation system. The non-dissipative trapezoidal rule ($\theta = 0.5$) time integration algorithm is used for all unsteady solution processes. The problem domains lie on $\mathbb{R}^d$ for $d = 1, 2$ and 3 with the following characteristics.

a) For verification and validation of the SGM theory, Chapter 5, I utilized both piecewise constant and distributed SGM function approximations on one- and two-dimensional finite elements. For a one dimensional axisymmetric steady conduction problem, the linear SGM form results for the identification $f_1 \equiv r$ in (5.30), hence $a = R_j$, and $b = R_{j+1}$ for the j th element in the solution domain.

For the transient conduction problem, the parameter $\phi > 0$ is utilized to verify the efficiency and monotonicity of the time acceleration procedure. The quadratic SGM f_2 with parameter $c_e > 0$ is validated for selected steady-state singular perturbation and non-linear problems. For a two-dimensional elliptic problem on an L-shaped region, the SGM choice remains $f_p(\mathbf{x})$ as a local, variable distribution where $p = 1, 2, 3$ is utilized. The nonlinear Burgers equation benchmark, and the shear driven cavity and step wall diffuser validation problems, are tested with a nodal distribution of subgrid embedding parameter, $\text{rsgm}_i \geq \{1\}$.

b) The GMP theory of Chapter 6 is verified via implementation of one dimensional perturbed matrices on one and two dimensional computational domains. The analytical solution to (1.1) in one- and two-dimensions, for $s = 0$, is exact preservation of the initial condition translated on Ω over $n\Delta t$. Computational results verify theory and summarize performance for traveling wave propagation verification problems, for a range of Courant numbers ($C \leq 2$) and with four different initial conditions

including a gaussian wave, a sine wave, and two square waves of different initial wavelength. The two dimensional unsteady verification problem is solid body rotation of an axisymmetric initial condition, *i.e.*, the “cosine hill” pure convection problem.

c) For the RCN algorithm of Chapter 7, computational results are presented to verify theory and summarize performance for a range of verification problems for (1.1), and a genuine validation problem for mass transport (7.1). The problem domains lie on $\mathbb{R}^d$, for $d = 1, 2, 3$, and a linear ($k = 1$) or quadratic ($k = 2$) basis FE CFD algorithm (Baker, 1991) has generated the discrete approximate solutions for reconstruction. Also, a Crank-Nicolson finite difference (CNFD) solution is reconstructed for the two dimensional verification problem. The non-dissipative trapezoidal time integration algorithm has been used in generating all CFD unsteady solutions.

8.2 Verifications

8.2.1 Axisymmetric Steady State Conduction:

The governing equation is parabolic, and the canonical form of (2.1)-(2.2) in polar coordinates is

$$-\frac{1}{r} \frac{d}{dr} \left(r \epsilon \frac{dq}{dr} \right) - s = 0 \quad r_1 < r < r_2 \quad (8.1)$$

$$a_1 \alpha q + \epsilon \nabla q \cdot \hat{\mathbf{n}} - a_3 \alpha q_n = 0 \quad r = r_1$$

$$q(r_2) = q_b \quad r = r_2$$

For $a_1 = 1 = a_3$, (8.1b) defines solution-dependent convection at $r = r_1$, while setting $a_1 = 0$ and $a_3 = 1$ produces the fixed flux boundary condition αq_n .

For this problem, the linear SGM form f_1 occurs naturally for the identification $a = R_j$ and $b = R_{j+1}$ for the j th element in the solution domain. Therefore, (5.43) will result in

$$\Delta x = \frac{h_e}{\frac{R_j^2 + R_{j+1}^2 + 4R_j R_{j+1}}{3(R_j + R_{j+1})}} = \frac{3h_e}{R_j \left(1 + s + \frac{2s}{1+s}\right)} \quad (8.2)$$

where $s = \frac{R_{j+1}}{R_j}$ is the (geometrically) non-uniform meshing nodal attraction factor.

For constant ϵ , the exact solution to (8.1) is logarithmic in the form

$$q(r) = A + B \ln r \quad (8.3)$$

Therefore, piecewise continuous approximate solutions will clearly exhibit associated approximation error and quantizable convergence trends. For a Dirichlet boundary condition (8.1c), this problem can also clearly quantize flux boundary computation within the WS^h construction. In (8.1), for the Dirichlet boundary condition $Q_J = q_b$, at $r = r_2$, the specific WS^h algebraic rearrangement for direct flux computation becomes

$$\frac{\epsilon}{\Delta_e} \begin{bmatrix} 1 & -1 & & \bigcirc \\ -1 & 2 & -1 & \\ & & \ddots & \\ & & -1 & 1 & -\Lambda \\ \bigcirc & & & \Lambda & \Lambda^2 \end{bmatrix} \begin{Bmatrix} Q1 \\ Q2 \\ \vdots \\ QJ-1 \\ QNJ \end{Bmatrix} = \begin{Bmatrix} \alpha q_r \\ 0 \\ \vdots \\ 0 \\ QJ \end{Bmatrix} \quad (8.4)$$

where $\Lambda = \frac{\Delta_e}{\epsilon}$, for $e = j - 1$, QJ is the (Dirichlet) data at node $j = J$, and QNJ is the associated (unknown) normal boundary flux at node J .

In general, the system matrix algebra form in (3.20) can be written as

$$\begin{bmatrix} b_1 & c_1 & & \bigcirc \\ a_2 & b_2 & c_2 & \\ & & \ddots & \\ & & & a_{J-1} & b_{J-1} & c_{J-1} \\ \bigcirc & & & & a_J & b_J \end{bmatrix} \begin{Bmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_{J-1} \\ Q_J \end{Bmatrix} = \begin{Bmatrix} B_{x=0} \\ 0 \\ \vdots \\ 0 \\ B_{x=L} \end{Bmatrix} \quad (8.5)$$

While solving for the $\{Q\}$ in (8.5), one may enforce the Dirichlet boundary condition at $(x = 0)$ by setting $c_1 = 0$, hence $Q_1 = B_{x=0}$. Similarly, $a_n = 0$ in (8.5) will produce a Dirichlet boundary at $(x = L)$.

Once the solution field $\{Q\}$ is known, the boundary fluxes corresponding to the Dirichlet node(s) may be found in two ways. The simplest way is to use

$$\begin{aligned} Q_{N1} &= \left. \frac{dq}{dn} \right|_{x=0} = b_1 Q_1 + c_1 Q_2 \\ Q_{NJ} &= \left. \frac{dq}{dn} \right|_{x=L} = a_J Q_{J-1} + b_J Q_J \end{aligned} \quad (8.6)$$

Replacing (8.6) in (8.4), one may also opt for finding the (Dirichlet) boundary flux(es) via solving a system matrix algebra statement written as

$$\begin{bmatrix} \frac{1}{b_1} & -\frac{c_1}{b_1} & & \bigcirc \\ \frac{a_2}{b_1} & \left(b_2 - \frac{a_2 c_1}{b_1}\right) & c_2 & \\ & & \ddots & \\ & & & a_{J-1} & \left(b_{J-1} - \frac{c_{J-1} a_J}{b_J}\right) & \frac{c_{J-1}}{b_J} \\ \bigcirc & & & & -\frac{a_J}{b_J} & \frac{1}{b_J} \end{bmatrix} \begin{Bmatrix} Q_{N1} \\ Q_2 \\ \vdots \\ Q_{J-1} \\ Q_{NJ} \end{Bmatrix} = \begin{Bmatrix} Q_1 \\ 0 \\ \vdots \\ 0 \\ Q_J \end{Bmatrix} \quad (8.7)$$

Note that in (8.7) both ends of the domain are considered as containing Dirichlet boundary later.

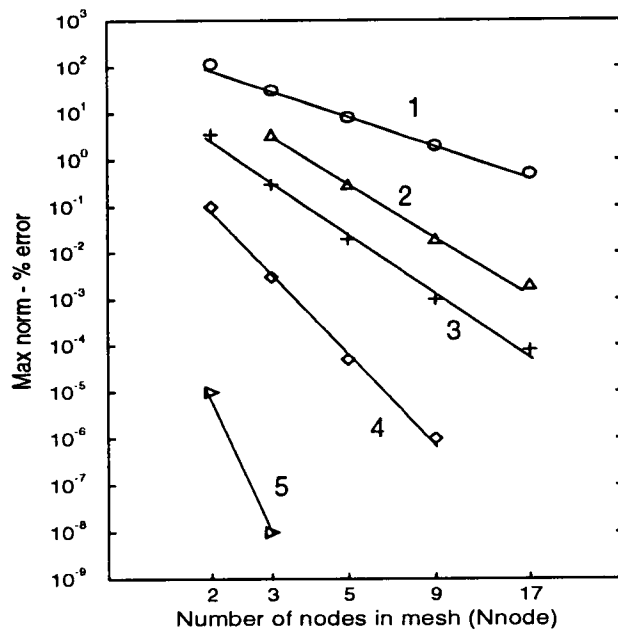
Table 8.1 summarizes normalized computational data for solutions to (8.1) for various combinations of k and p . Comparing the SGM ($k = 1, p \geq 1$) and FEM ($k \geq 1, p = 0$) data, via definition (5.7), both the point (max) and energy norm computed data confirm at least an order of magnitude better accuracy for the SGM solution on a given “Nnode” (number of nodes) mesh. For flux computed directly via the weak statement, the error is a one to two orders of magnitude improvement over second (and first) order accurate difference formulae predictions, and is obtained at no additional computational cost.

Figure 8.1 summarizes SGM algorithm asymptotic convergence for various (k, p) combinations. Data in Figures 8.1a and 8.1b both conform to the classical energy norm theory (4.7) for $k = 1, 2$ FEM solutions on a uniform mesh, as the corresponding (logarithmic) data are interpolated by straight lines of slope equal to $2k$. In distinction, for linear basis SGM embedding, the convergence rate (slope) increases rapidly as $p > 0$ increases. Specifically, for $k = 1$ and $p = 3$, the convergence trend slope is 10.8, which is nearly double that predicted for the classical theory (4.7).

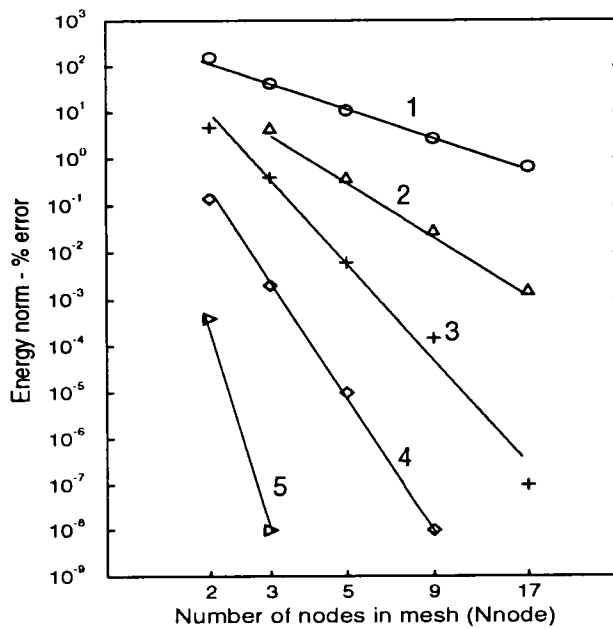
Figure 8.1c graphs the companion WS^h -computed boundary flux errors. While the $k = 2$ standard FEM algorithm results are significantly better than the $k = 1$ FEM solutions, the SGM $p \geq 1$ boundary flux computations are at least two orders of magnitude more accurate than either FEM result. The boundary flux error decreases uniformly as p increases, and the combination of h -SGM p minimizes error in the QNJ

Table 8.1 Solution and Surface Flux Errors

k	p	Nnode	$\frac{q_1 - q(r_1)}{q(r_1)}\%$	$\frac{\ e^h\ _E}{\ q\ _E}\%$	Dirichlet flux error, %		
			Point	Norm	WS^h	second order FD	first order FD
1	0	3	-31.300	42.100	-0.60	8.00	-15.0
1	0	9	-2.000	2.700	-0.04	0.30	-3.3
1	0	17	-.510	.680	-0.01	0.07	-1.6
2	0	3	-3.500	4.700	-0.07	7.60	-15.5
2	0	9	-.020	.030	$\sim 10^{-3}$	0.30	-3.3
2	0	17	-.010	.020	$\sim 10^{-4}$	0.07	-1.6
1	1	3	-.300	.400	-0.01	8.50	-15.1
1	1	9	-.001	.001	$\sim 10^{-5}$	0.30	-3.3
1	1	17	$\sim 10^{-5}$	$\sim 10^{-8}$	$\sim 10^{-8}$	0.07	-1.6
1	2	3	-.003	.004	$\sim 10^{-5}$	8.50	-15.1
1	2	9	$\sim 10^{-7}$	$\sim 10^{-8}$	$\sim 10^{-8}$	0.30	-3.3
1	3	3	$\sim 10^{-7}$	$\sim 10^{-7}$	$\sim 10^{-8}$	8.5	-15.1
1	4	2	$\sim 10^{-8}$	$\sim 10^{-8}$	$\sim 10^{-8}$	-	-38.6

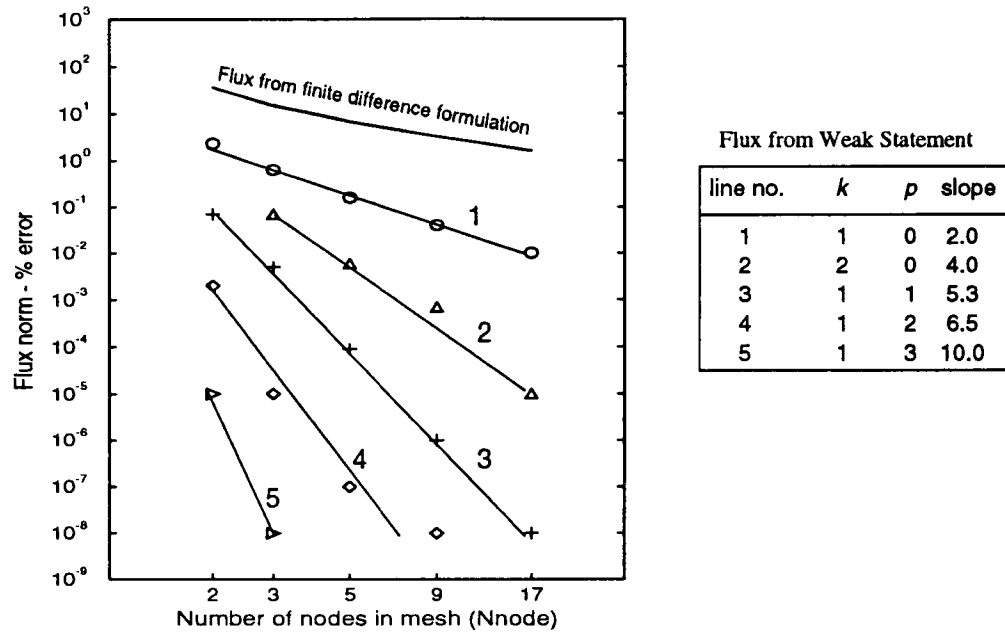


a) Max norm - % error



b) Energy norm - % error

Figure 8.1 Convergence study, error for $d = 1$ steady conduction problem.



c) Flux norm, % error

Figure 8.1 Convergence study, error for $d = 1$ steady conduction problem
(concluded).

computation. The flux computed via any finite difference formulation is totally dependent on the mesh measure h_e , hence its error is as much as five orders of magnitude higher than the WS^h prediction on the coarsest mesh, Figure 8.1c.

8.2.2 1d Unsteady Parabolic Problem:

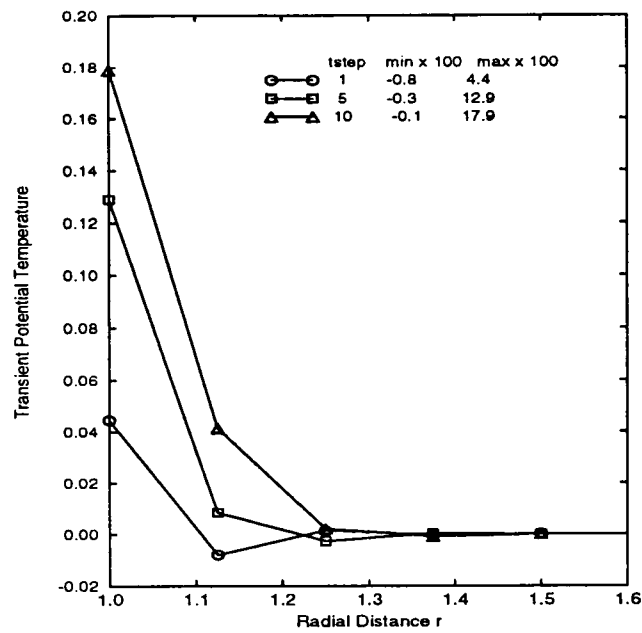
Unsteady radial conduction in the axisymmetric definition modifies (8.1a) to the form

$$\mathcal{L}(q) = \frac{\partial q}{\partial t} - \frac{1}{r} \frac{\partial}{\partial r} \left(r \epsilon \frac{\partial q}{\partial r} \right) = 0 \quad r_1 < r < r_2 \quad (8.8)$$

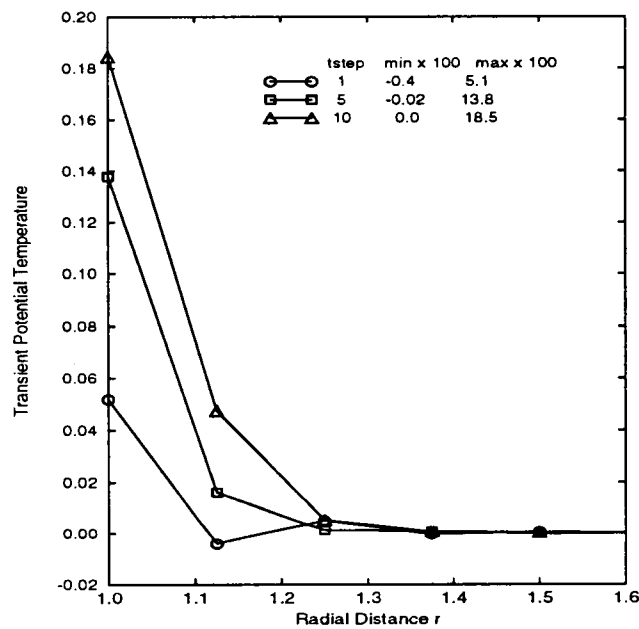
with initial and boundary conditions required for well-posedness. The SGM choice remains linear, *i.e.*, $f_1 \equiv r$, and the problem statement allows validation of the role of SGM-derived ϕ , recall (5.57)-(5.60). Solution monotonicity is a direct function of the initial condition, $q(r, 0) = q_0(r)$, and the selected flux (Robin) and Dirichlet boundary conditions are identical to those for the steady state problem.

Summaries of early-time solution accuracy and monotonicity are presented in Figure 8.2. The temporal solution evolution for both $k = 1$ and $k = 2$ standard FEM ($p = 0$) method is distinctly non-monotone on this meshing, Figure 8.2a and 8.2b, although nodal accuracy for surface temperature (at the Robin specification) is acceptable for engineering purposes. In distinction, the $p = 1$, $\phi = 1$ time-accurate SGM $f_1(r)$ solution procedure, Figure 8.2c, generates a monotone and accurate solution on this meshing.

For efficiency, selecting $\phi \geq 1$ can facilitate evolutionary monotone attainment of the steady state using a much smaller number of timesteps. As presented in Table 8.2, for standard FEM, $p = 0 = \phi$, the trapezoidal ($\theta = 0.5$ in (3.22)) rule time integration scheme solution becomes steady, *i.e.*, insensitive to time, after 1820 timesteps, with the initial timestep of 0.0001, yielding the data graphed in Figure 8.2a. In comparison, for the SGM time acceleration procedure, Figure 8.2d, with the same initial Δt and θ , only 82 timesteps are required for $p = 1$ and $\phi = 30$ to monotonely reach a time station with solution independent of timestep.

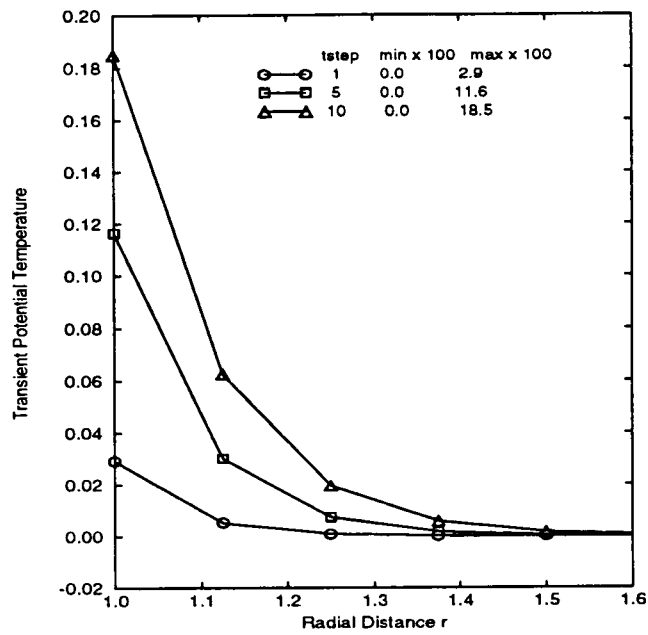


(a) FEM, $k = 1$, $p = 0$, $\phi = 0$,

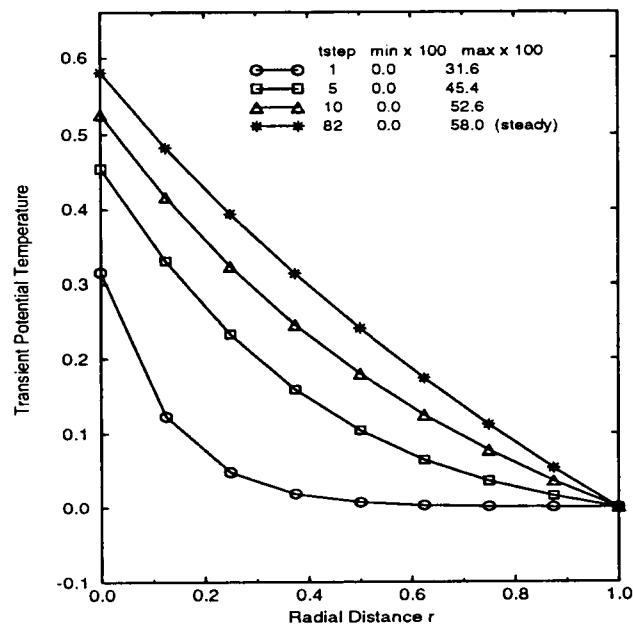


(b) FEM, $k = 2$, $p = 0$, $\phi = 0$,

Figure 8.2 Solution monotonicity and effect of ϕ for the transient evolution.



(c) SGM, $k = 1$, $p = 1$, $\phi = 1$,



(d) SGM, $k = 1$, $p = 1$, $\phi = 30$

Figure 8.2 Solution monotonicity and effect of ϕ (concluded).

Table 8.2 Effect of ϕ on a uniform 9-node mesh

k	p	ϕ	tstep	$Q_{min} \times 10^2$	$Q_{max} \times 10^2$
1	0	0	1820	0.0	58.0 (steady)
2	0	0	1820	0.0	58.1 (steady)
1	1	1	990	0.0	58.1 (steady)
1	1	30	82	0.0	58.1 (steady)

8.2.3 Steady conduction with source:

Consider the parabolic equation

$$\frac{1}{r} \frac{d}{dr} \left(r \frac{dq}{dr} \right) = -s \quad (8.9)$$

where $s(r)$ is a linear source term. The solution to (8.9) is

$$q = -\frac{r^2}{4} + a \ln r + b \quad (8.10)$$

and specific forms are:

Case I : Both ends Dirichlet

$q|_{r=1} = 0 \Rightarrow b = \frac{1}{4}$ and $q|_{r=2} = 0 \Rightarrow a = \frac{3}{4 \ln 2}$. Hence,

$$q = -\frac{r^2}{4} + \frac{3 \ln r}{4 \ln 2} + \frac{1}{4} \quad (8.11)$$

Case II : Left end Dirichlet, right end homogeneous Neumann

$q|_{r=1} = 0 \Rightarrow b = \frac{1}{4}$ and $\left. \frac{dq}{dr} \right|_{r=2} = 0 \Rightarrow a = 2$. Hence,

$$q = -\frac{r^2}{4} + 2\ln r + \frac{1}{4} \quad (8.12)$$

Upon discretization, the matrix algebra statement (3.20) becomes $S_e \left(\{RAD\}_e^T [D]_e \{Q\}_e - \{RAD\}_e^T [M]_e \{S\}_e \right) = \{0\}$ where $\{RAD\}_e$ contains the nodally discretized values of r and $[D]$ and $[M]$ are the FE hypermatrices. In Chapter 5.4, a discussion about the locally extremized p -element is given. For verification, I employ the quadratic degree ($k = 2$) basis function for computing the diffusion matrix $[D]$ and reduce $\{RAD\}_e^T [D]_e$ via (5.10). The resulting change in the data structure is noticeable as follows.

On element $e = 1$, choosing linear basis function for $[D]$ and premultiplying with the corresponding $\{RAD\}_{e=1}^T \equiv \{1, \frac{4}{3}\}$, results in the diffusion matrix

$$[D]_e = \begin{bmatrix} 3.5 & -3.5 \\ -3.5 & 3.5 \end{bmatrix} \quad (8.13)$$

In comparison, using the $k = 2$ basis function for $[D]$ and the corresponding data $\{RAD\}_{e=1}^T \equiv \{1, \frac{7}{6}, \frac{4}{3}\}$, and then reducing the square matrix $\{RAD\}_e^T [D]_e$ via (5.10) yields

$$[D]_e = \begin{bmatrix} 3.47619049 & -3.47619049 \\ -3.47619049 & 3.47619049 \end{bmatrix} \quad (8.14)$$

For a 4-node uniform mesh, the p -embedded ($p \geq 0$) solution comparison details are given in Table 8.3 and graphs are shown in Figures 8.3-8.5. For both Case I and II,

Table 8.3 Solution comparison for various p on a 4 node mesh

Case I. a) Both end Dirichlet, constant source (ddcs); $s=1$

maximum solution error for dd cs

p	Q2 (r=1.333)	Q3 (r=1.667)
0	0.1162250D+00	0.1078568D+00
1	0.1168287D+00	0.1082767D+00
2	0.1168316D+00	0.1082783D+00
3	0.1168318D+00	0.1082785D+00
4	0.1168317D+00	0.1082784D+00

Both end Dirichlet, variable source (ddvs); $s=\text{rad}(i)$

p	Q2 (r=1.333)	Q3 (r=1.667)
0	0.1704117E+00	0.1698951E+00
1	0.1712901D+00	0.1705475D+00
2	0.1712943D+00	0.1705499D+00
3	0.1712946D+00	0.1705502D+00
4	0.1712944D+00	0.1705500D+00

Solution energy norm (divided by 2π)

p	dd cs	dn cs*	dd vs	dn vs*
0	0.55788022E-01	0.34014404E+00	0.13019790E+00	0.93762982E+00
1	0.56038789D-01	0.34220827D+00	0.13076489D+00	0.94321562D+00
2	0.56039876D-01	0.34221880D+00	0.13076729D+00	0.94324375D+00
3	0.56039972D-01	0.34222248D+00	0.13076751D+00	0.94325393D+00
4	0.56039912D-01	0.34221925D+00	0.13076795D+00	0.94325398D+00

* dn cs : left end Dirichlet, right end Neumann, constant source
 dn vs : left end Dirichlet, right end Neumann, variable source

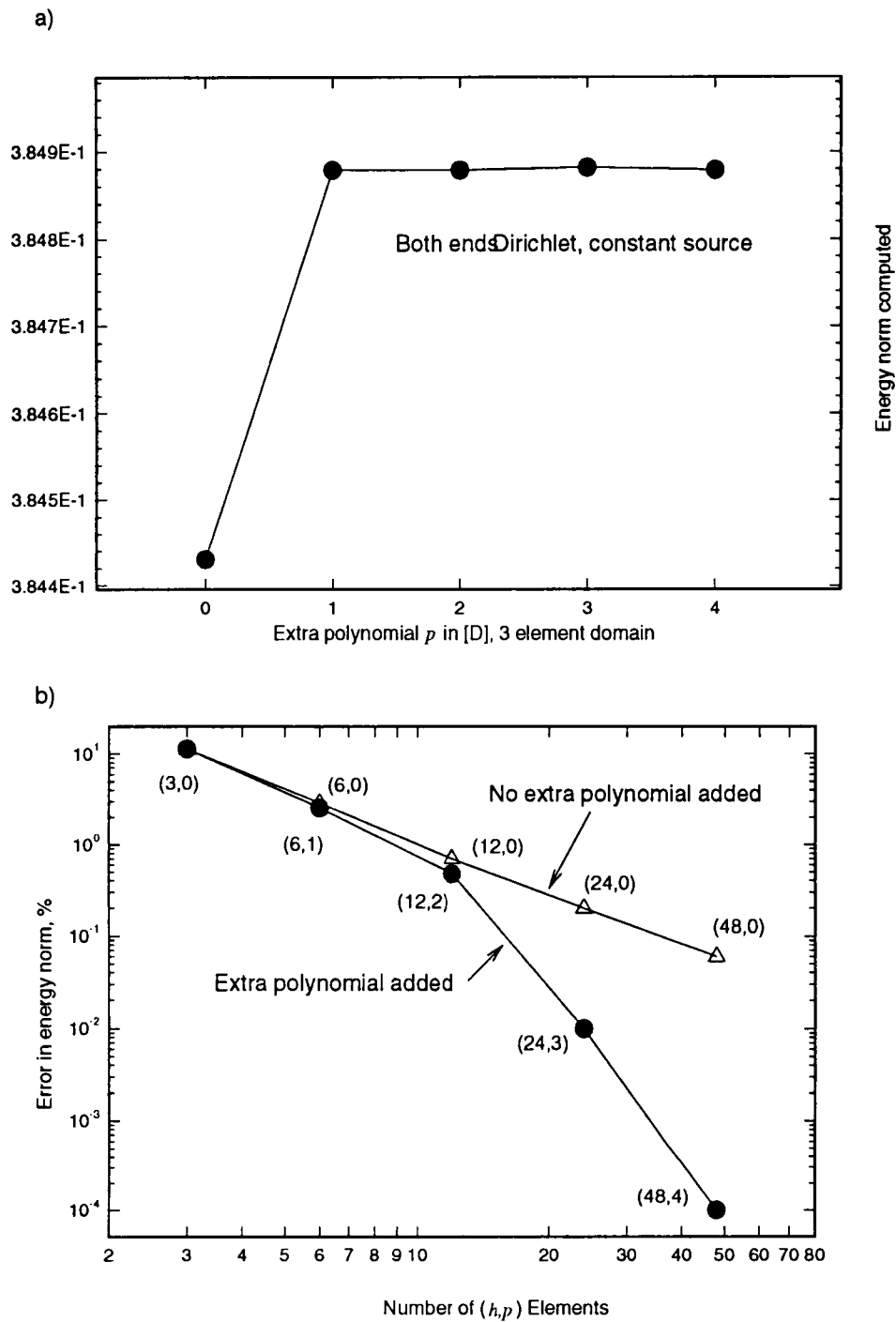


Figure 8.3 Effect of increasing p on solution energy in Case I for constant source $s = 1$.

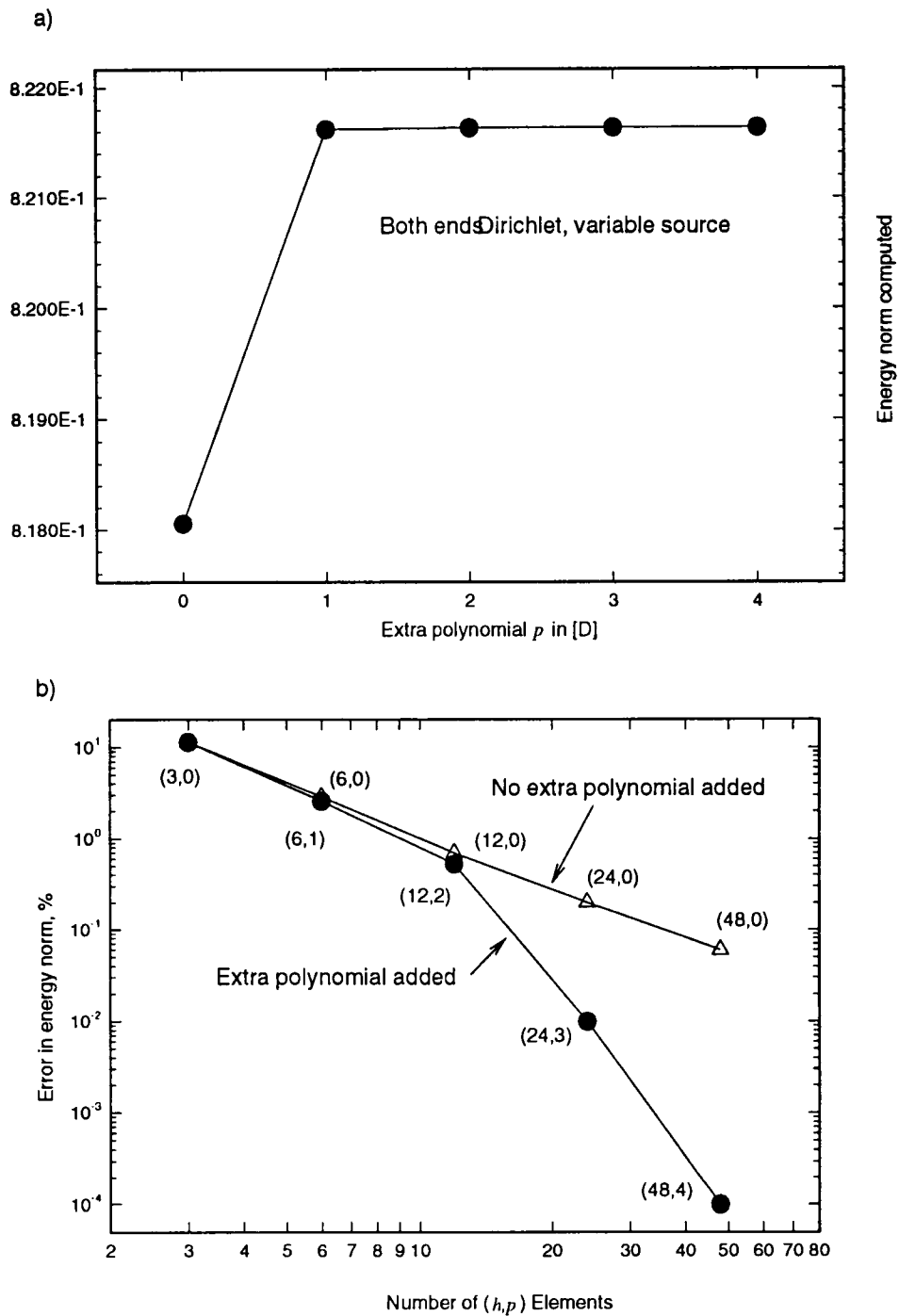


Figure 8.4 Effect of increasing p on solution energy in Case I for variable source $s =$ radius.

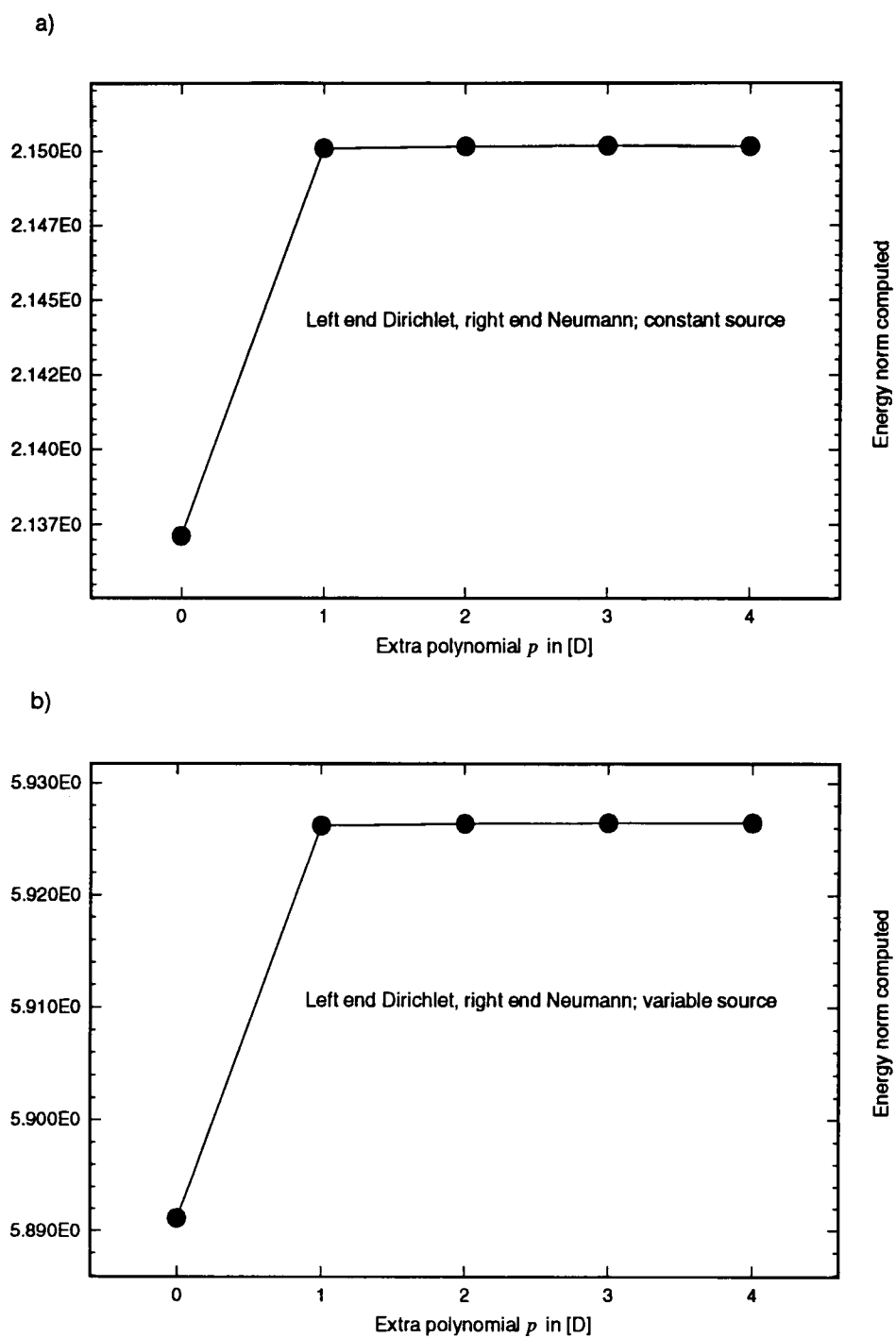


Figure 8.5 Effect of increasing p on solution energy in Case II.

these results clearly confirm a sharp decrease in computational error as p goes from 0 to 1 (linear to quadratic). Conversely, for $p \geq 2$, increasing p does not significantly affect the solution (error), Figures 8.3a, 8.4a and 8.5. Convergences in the solution energy norm error computations are compared between the standard linear basis ($p = 0$) and the companion combination h - p solution energy norms, Figures 8.3b, 8.4b. These results indicate that the slope of computational error decrease is significantly increased as the $(h$ - p) combination spans (12,2) to (48,4). Specifically, the solution error trend affirms usage of $p = 1$ as optimal for improved solution accuracy and computational storage efficiency.

8.2.4 Linear Poisson equation on L-shaped two-dimensional domain:

The elliptic boundary value problem definition is

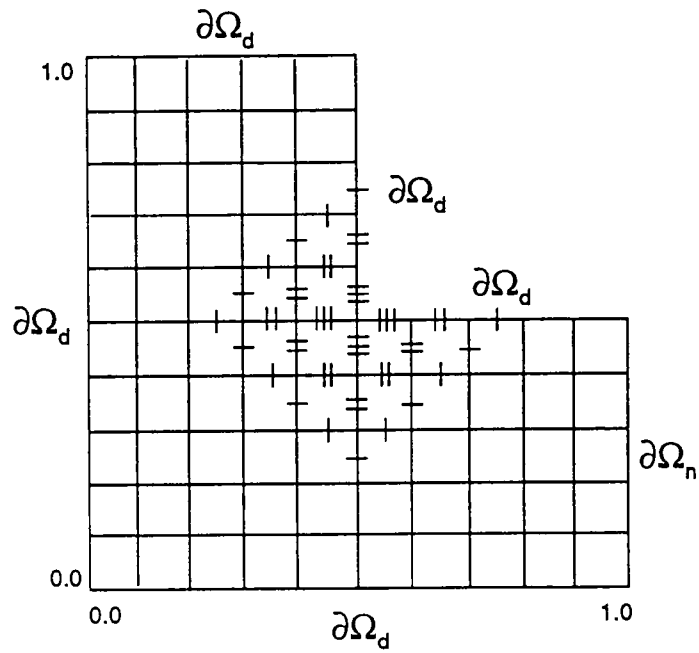
$$\begin{aligned} -\Delta q &= s && \text{in } \Omega \\ q &= 0 && \text{on } \partial\Omega_d \\ \frac{\partial q}{\partial n} &= 0 && \text{on } \partial\Omega_n \end{aligned} \tag{8.15}$$

which on an L-shaped domain Ω with $s = \text{constant}$ is a standard p -element benchmark problem, c.f., Demokowicz, *et. al* (1989), Zienkiewicz *et. al* (1982). The Dirichlet boundary $\partial\Omega_d$ fully encompasses $\partial\Omega$ except for one homogeneous Neumann boundary segment $\partial\Omega_n$, see page 128. The L-shaped domain introduces a geometric singularity at the convex corner. Hence, although the analytical solution to this linear problem is smooth, an approximate solution procedure can suffer large local error in this vicinity, dependent on meshing. Error reduction quantization is thereby facilitated via local

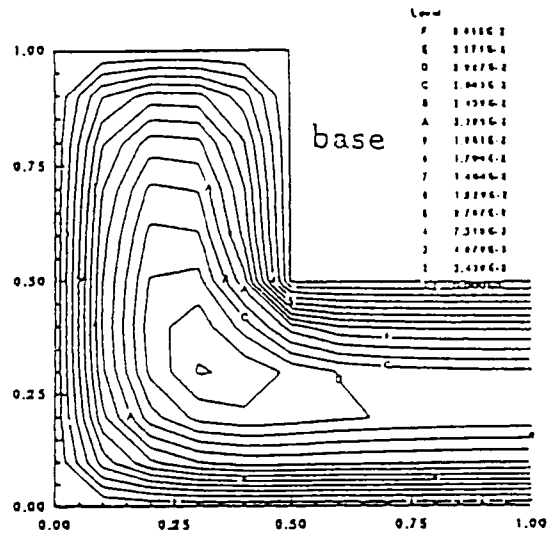
mesh refinement, increasing the FEM basis degree and/or reducing h and increasing k or p , in the near region. This two dimensional elliptic boundary value problem (bvp) with constant source $s = 1$ has been studied and documented extensively, c.f. Babuska, *et. al* (1990), Demkowicz, *et. al* (1992) and Zienkiewicz, *et. al* (1992). Using p -version or h - p version hierarchical elements, the results are convincing although the studies do not mention attendant costs associated with escalation of global matrix size. The investigation in this dissertation demonstrates that static condensation of a higher degree basis diffusion matrix (only) on a locally normalized coordinate scale has significant impact on coarse mesh reduction of solution energy and max norm errors while containing the size of the global matrix to that of the linear basis formulation.

Figure 8.6a, in addition to presenting the boundary conditions as defined, illustrates the SGM $f_p(x, y)$ embedding for determined-optimal p distributions on this mesh. Each stroke, perpendicular to an element side, signifies the degree of embedded polynomial p statically condensed on that element boundary segment. The two dimensional element matrices are formed via (5.1), where the $k = 1$ basis functions are the usual bilinear products of the one dimensional shape functions.

The base $k = 1$, $p = 0$ WS^h FEM solution isoline distribution is shown in Figure 8.6b. The comparison $p \geq 1$ SGM solutions are plotted in Figure 8.6c-e. In the "eyeball" norm, the enriched- basis SGM solutions obviously become smoother as local p is uniformly increased around the interior corner. The energy norm computed for the FEM and each SGM solution is also given in Figure 8.6b-e. The convergence

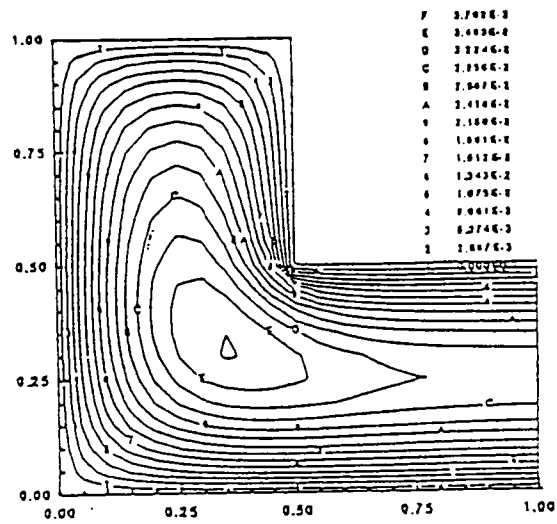


a) Boundary condition and p distribution

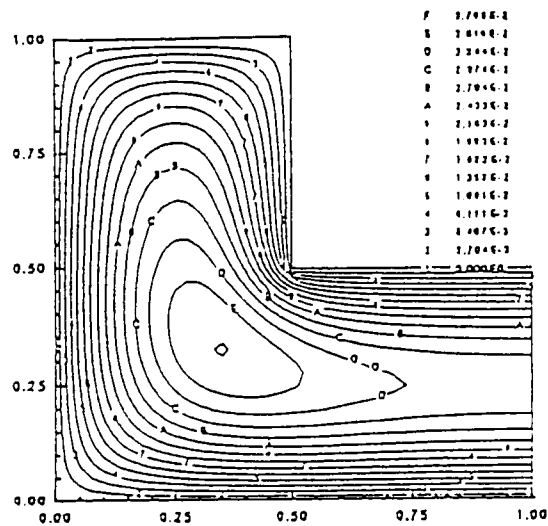


b) base FEM $k = 1$, $p = 0$, $\|q^h\|_E = 3.41$

Figure 8.6 Linear Poisson problem on L-shaped domain, 75 element (96 node) mesh, FE and SGM WS^h solutions.

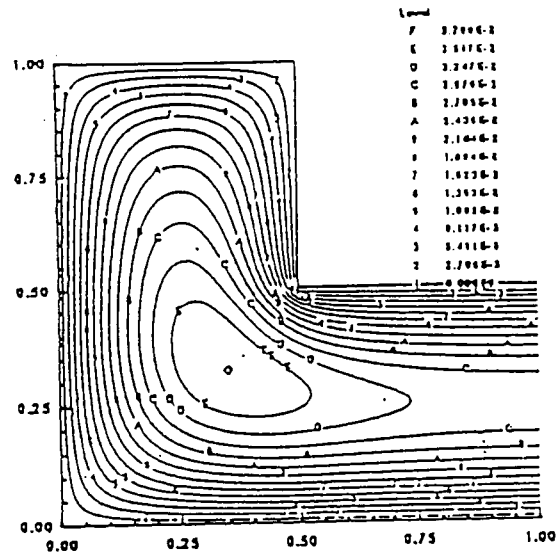


c) SGM $k = 1, p = 1, \|q^h\|_E = 3.762$

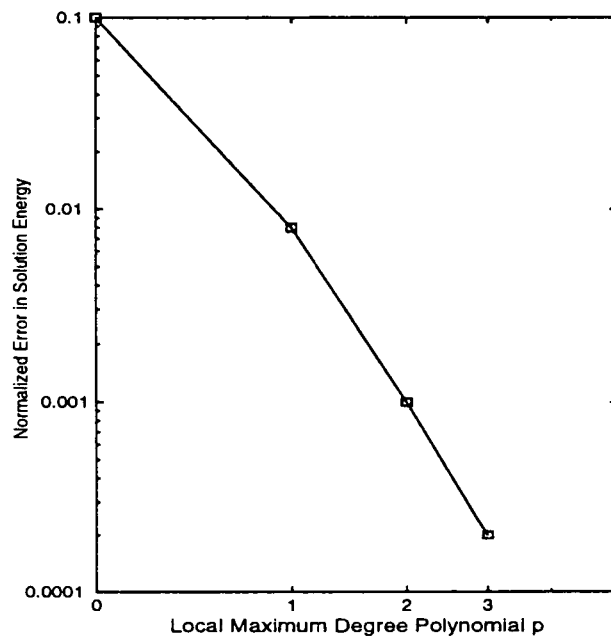


d) SGM $k = 1, p = 1, 2, \|q^h\|_E = 3.785$

Figure 8.6 Linear Poisson problem on L-shaped domain (continued).



e) optimal $k = 1$, $p = 1, 2, 3$, $\|q^h\|_E = 3.788$



f) Convergence of the energy norm

Figure 8.6 Linear Poisson problem on L-shaped domain (concluded).

facilitated via local mesh refinement, increasing the FEM basis degree and/or reducing h and increasing k or p , in the near region. This two dimensional elliptic boundary value problem (bvp) with constant source $s = 1$ has been studied and documented extensively, c.f. Babuska, *et. al* (1990), Demkowicz, *et. al* (1992) and Zienkiewicz, *et. al* (1992). Using p -version or h - p version hierarchical elements, the results are convincing although the studies do not mention attendant costs associated with escalation of global matrix size. The investigation in this dissertation demonstrates that static condensation of a higher degree basis diffusion matrix (only) on a locally normalized coordinate scale has significant impact on coarse mesh reduction of solution energy and max norm errors while containing the size of the global matrix to that of the linear basis formulation.

Figure 8.6a, in addition to presenting the boundary conditions as defined, illustrates the SGM $f_p(x, y)$ embedding for determined-optimal p distributions on this mesh. Each stroke, perpendicular to an element side, signifies the degree of embedded polynomial p statically condensed on that element boundary segment. The two dimensional element matrices are formed via (5.1), where the $k = 1$ basis functions are the usual bilinear products of the one dimensional shape functions.

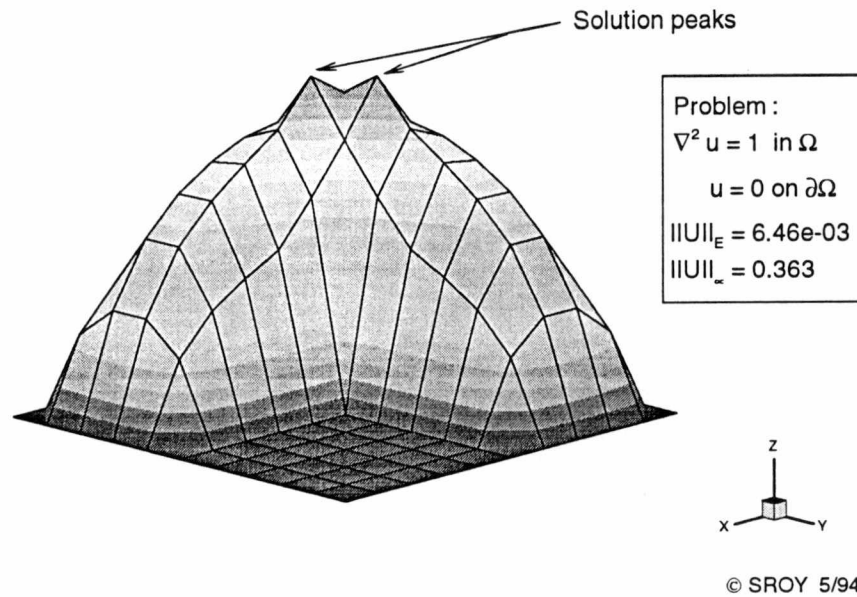
The base $k = 1$, $p = 0$ WS^h FEM solution isline distribution is shown in Figure 8.6b. The comparison $p \geq 1$ SGM solutions are plotted in Figure 8.6c-e. In the “eyeball” norm, the enriched- basis SGM solutions obviously become smoother as local p is uniformly increased around the interior corner. The energy norm computed for the FEM and each SGM solution is also given in Figure 8.6b-e. The convergence

trend with local p -enrichment for the approximation error measured in the energy norm is plotted in Figure 8.6f. The normalized error in solution energy, on this fixed 96-node mesh, is reduced from order 10^{-1} (for $p = 0$) to order 10^{-4} for p -symmetrically and uniformly distributed locally to $p_{max} = 3$, for no additional matrix solution computational cost.

An alternative to symmetrically and uniformly implementing higher order matrices on selected elements is to use a global p -embedding “all around” the computational domain. The local (elementwise) condensation process still allows implementation without any escalation of the global system matrix size. Figure 8.7 presents perspective view comparison of the sequence of base $k = 1$ solution on coarse to moderately dense uniform meshes. The linear element 96 node coarse mesh solution clearly contains peaks ($\|U\|_{\infty} = 0.363$) indicating that the accurate solution peak may lie somewhere between those two nodes, Figure 8.7a. The double peak remains present, Figure 8.7b, for the doubly-refined 341 node uniform mesh solution ($\|U\|_{\infty} = 0.369$). However, for a four-times refined dense (1281 node) mesh, the $k = 1$ FE solution produces the correct single peak ($\|U\|_{\infty} = 0.390$), Figure 8.7c, in agreement with the exact solution. The companion coarse mesh $p = 1$ (quadratic) embedded procedure uniformly on all elements, with locally condensed element diffusion matrices, exhibits excellent solution quality (one peak) with the max norm $\|U\|_{\infty} = 0.388$ nominally identical to the 1281 node $k = 1$ basis result, Figure 8.7d.

The comparison of GWS^h , uniformly p -embedded and SGM local p -embedded solution energy norm data is presented in Figure 8.8. The solution error in energy for

a) $p=0$ (linear element) on a 96 node mesh



b) $p=0$ (linear element) solution on a 341 node mesh

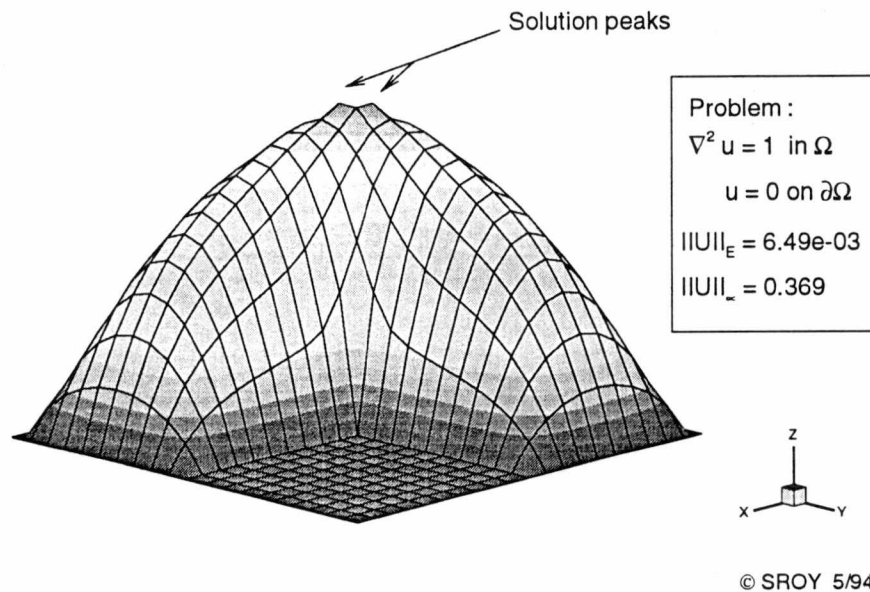
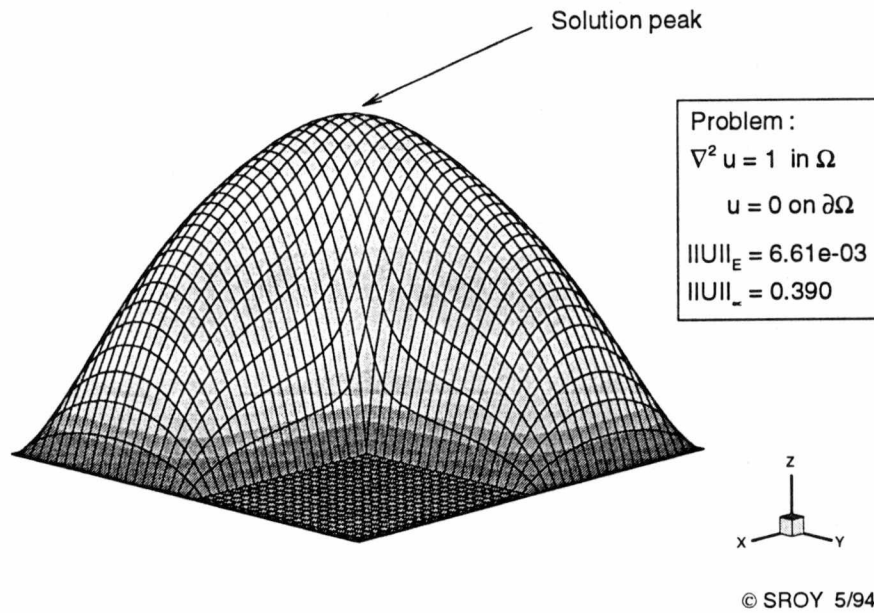


Figure 8.7 Comparison among coarse to dense mesh linear basis and coarse mesh globally p -embedded solutions for the L-shaped region benchmark.

c) $p=0$ (linear element) on a 1281 node mesh



d) $p=1$ (extremized element) on a 96 node mesh

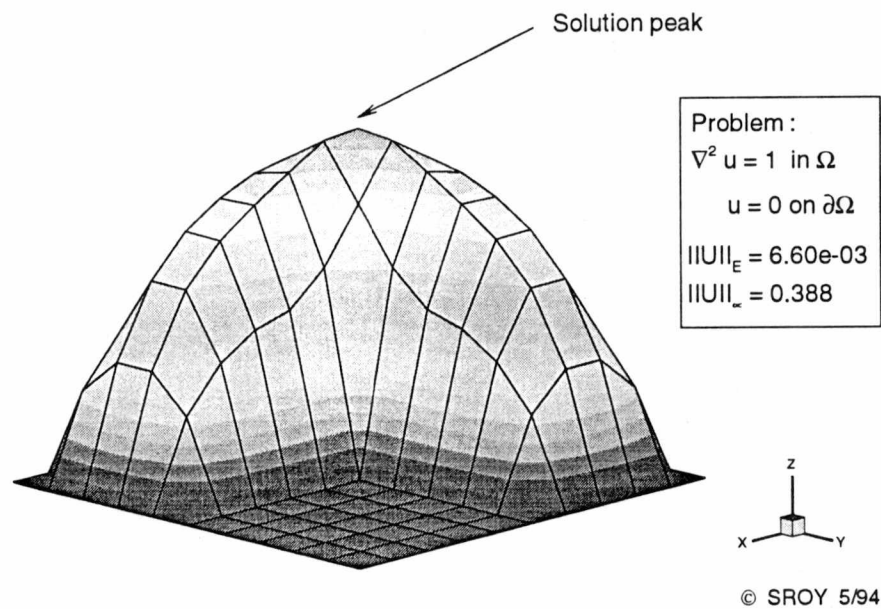


Figure 8.7 Comparison among coarse to dense mesh linear basis and coarse mesh globally p -embedded solutions (contd.)

$c = 1$ (or uniform $p = 1$ reduced) p -embedding algorithm diminishes twice as fast as the GWS solution energy norm, as expected for an essentially $k = 2$ GWS algorithm. Conversely, for the SGM theory with $c = 0.96$, the solution energy norm for the coarsest mesh (DOF = 121) is comparably accurate to a truly dense mesh (DOF = 6561) GWS^h solution.

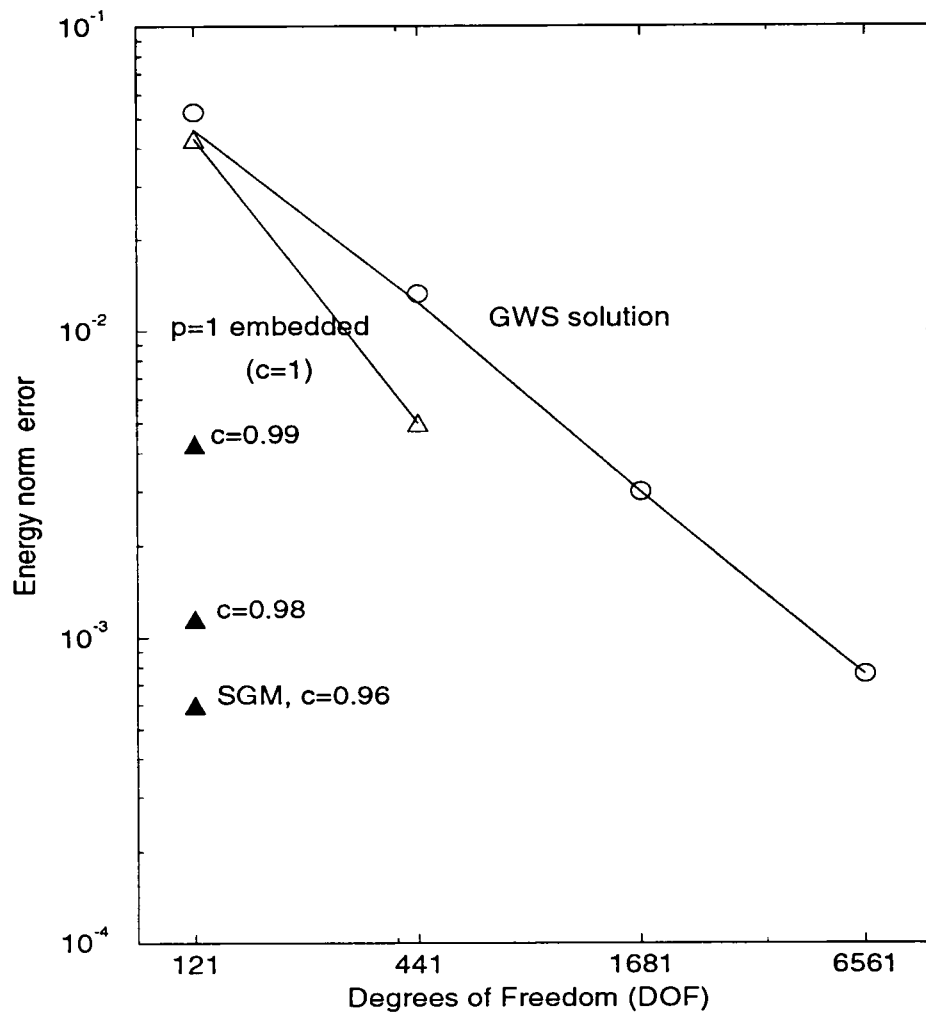


Figure 8.8 Convergence and error in energy norm for various FE algorithms vs. nodal degrees of freedom.

8.2.5 1d Steady Singular Perturbation Problem ($u \neq 0$, $\epsilon \rightarrow 0$)

The governing parabolic differential equation is

$$\frac{d}{dx} \left(uq - \epsilon \frac{dq}{dx} \right) = 0 \quad \text{on } 0 \leq x \leq \ell \quad (8.16)$$

which upon non-dimensionalization and for constant u becomes

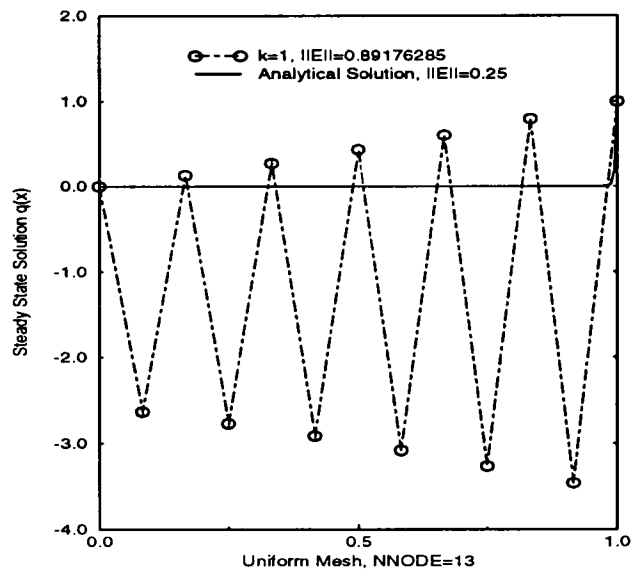
$$\frac{d}{dx} \left(q - \frac{1}{\text{Re}} \frac{dq}{dx} \right) = 0 \quad \text{on } 0 \leq x \leq 1 \quad (8.17)$$

where $\text{Re} = u\ell/\epsilon$ is the familiar “Reynolds number.” Solutions to (8.17) are parameterized by Re , and for the Dirichlet boundary conditions $q(0) = 0$ and $q(1) = 1$, the analytical solution to (8.17) is

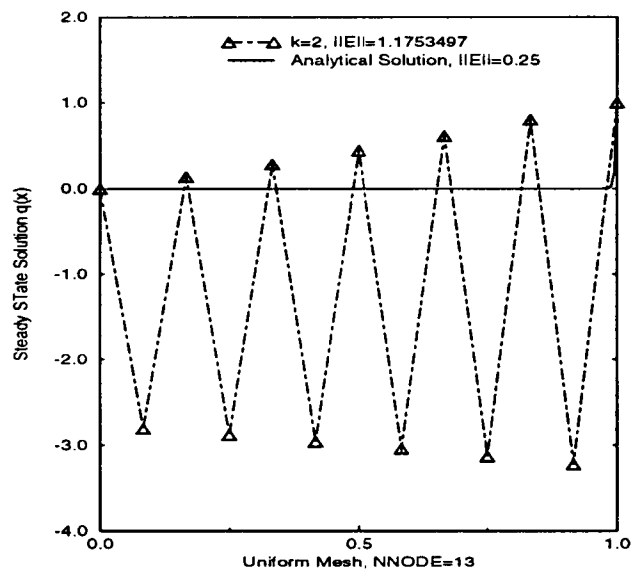
$$q(x) = \frac{e^{x\text{Re}} - 1}{e^{\text{Re}} - 1} \quad (8.18)$$

which is smooth but exhibits a sharp wall-layer character as Re becomes large, *i.e.*, $\epsilon \rightarrow 0$ in (8.16). As $\text{Re} \rightarrow \infty$, the analytical solution energy norm converges to exactly $\|q\|_E = 0.25$, and the associated domain end-point boundary fluxes for (8.18) equal exactly zero and unity.

In practical CFD situations, Re can range to 10^5 and larger, yielding a sharply varying wall (boundary) layer solution that is a challenge to accurately resolve numerically without use of local fine meshing and/or artificial diffusion. For large Re , the dominant error mechanism is dispersive, as clearly illustrated in Figure 8.9a-b for standard $k = 1$ and $k = 2$ GWS solutions on a uniform $N_{\text{node}}=13$ mesh. In particular, a uniform mesh containing 50 nodes, cannot yield a monotone solution for $\text{Re} \geq 50$ using either the $k = 1$ or $k = 2$ standard GWS algorithm.

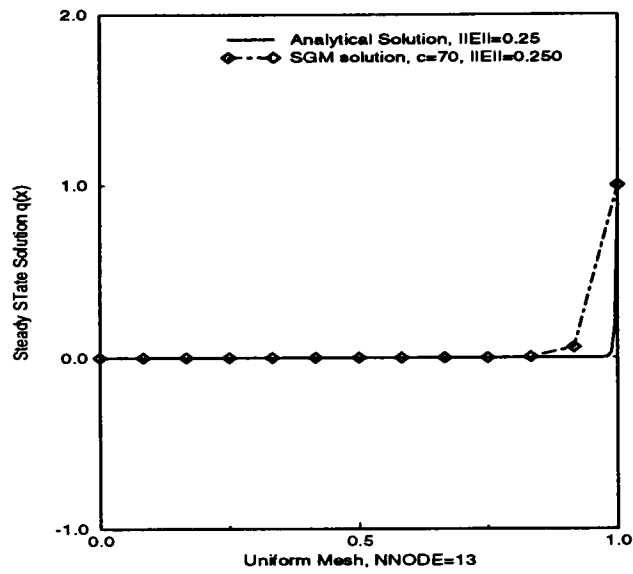


a) FEM $k = 1, p = 0$

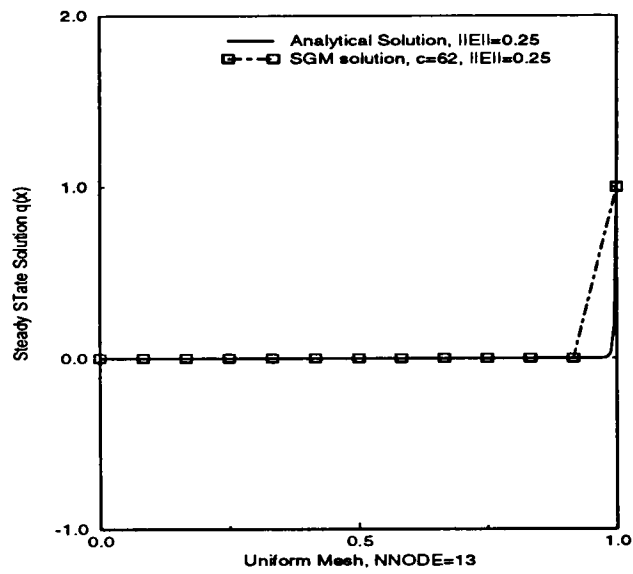


b) FEM $k = 2, p = 0$

Figure 8.9 Parabolic perturbation steady state solution on a 13-node mesh, $Re = 1000$, standard GWS and SGM (quadratic f_2) algorithm solutions.



c) SGM, $c = 70$, $p = 1$



d) SGM, $c = 62$, $p = 1$

Figure 8.9 Parabolic perturbation steady state solution on a 13-node mesh, $Re = 1000$, standard GWS and SGM (quadratic f_2) algorithm solutions (concluded).

The classical theory predicts that for increasing Nnode sufficiently, and/or $k \geq 1$ for the standard FE methodology, the solution accuracy steadily improves. However, for Re large enough, all GWS generated solutions remain non-monotone, hence convergence and error estimation is problematical. In distinction, an adroit selection of the SGM f_2 nodeless parameter c_e always yields a monotone, potentially nodally-exact solution for q on any mesh coupled with exact determination of boundary fluxes. On a 13 node uniform mesh, the SGM monotone theory (5.46) predicts $c = 2.625$ for $Re = 50$, which is indeed obtained, Table 8.4.

As Reynolds number increases, achieving a monotone, accurate approximate solution becomes increasingly difficult for all finite element (and finite difference) procedures without adding artificial diffusion. For example, a 100-node uniform mesh will not yield an acceptable artificial diffusion-free solution for $Re = 1000$, and a non-uniform mesh of 50 nodes cannot totally resolve the oscillations either. In distinction, the SGM quadratic f_2 choice, with $c = 70$, results in a monotone solution, slightly diffused, on a 13-node uniform mesh, Figure 8.9c. However, using (5.46), the theoretically predicted precise value of c for monotone solution is

$$c = \frac{U}{2} \left(\frac{3hRe}{2} \right) - \frac{1}{2} = \frac{1}{2} \left(\frac{3 \times 1000}{2 \times 12} \right) - \frac{1}{2} = 62 \quad (8.19)$$

The resultant SGM f_2 solution is absolutely monotone, and nodally-exact to round-off, Figure 8.9d.

Table 8.5 documents the FEM and SGM element matrices comparison. For the advection-diffusion problem at $Re = 1000$, the SGM element matrix (diffusion plus convection) is of pure upwind form. Corresponding assembled system matrices and

Table 8.4 1D Singular Perturbation Solution and Surface Flux Errors for $Re=50$,
 $c = 2.7$, (Appendix M)

k	p	Nnode $\left(\frac{h_R}{h_L}\right)$	$\frac{q_{R-1} - q_{R-1}^a}{q_R} \%$ Point	$\frac{\ e^h\ _E}{\ q\ _E} \%$ Norm	Dirichlet flux error, %		
					WS^h	2nd order FD	1st order FD
1	0	13 (1)	-36.7	$\sim 10^{-5}$	-35.1	45.6	-67.6
1	0	25 (1)	-14.5	$\sim 10^{-7}$	-2.0	-26.0	-51.0
1	0	49 (1)	-3.8	$\sim 10^{-5}$	-0.2	11.7	-34.3
1	0	13 (0.9)	-15.7	$\sim 10^{-7}$	-4.5	-28.7	-52.2
2	0	13 (1)	-18.9	$\sim 10^{-3}$	-31.0	-52.8	-71.9
2	0	25 (1)	-6.3	$\sim 10^{-5}$	-1.2	-31.9	-54.9
2	0	49 (1)	-1.2	$\sim 10^{-6}$	-0.1	-15.1	-36.7
2	0	13 (0.9)	-9.0	$\sim 10^{-3}$	-1.1	-34.9	-57.6
1	1	13 (1)	-.4	$\sim 10^{-8}$	$\sim 10^{-8}$	-64.6	-76.3

Table 8.5 Comparison element matrices $[D + U]_e$, $Re = 1000$, standard FE and SGM

f_2

Standard GWS			SGM f_2 $c = 62$
$k = 1$ ($c = 1$)	$k = 2$ ($c = 1$)		
$\begin{bmatrix} -0.488 & 0.488 \\ -0.512 & 0.512 \end{bmatrix}$	$\begin{bmatrix} -0.486 & 0.651 & -0.165 \\ -0.683 & 0.032 & 0.651 \\ 0.169 & -0.683 & 0.514 \end{bmatrix}$		

the solution print-outs, Table 8.6, show that for standard FEM $k = 1, 2$, diagonal terms of the system matrix approach zero as Reynolds number increases. Conversely, the pure upwind form of SGM matrices always yield the system matrix diagonal term as unity.

Table 8.6 1d Singular perturbation solution comparison for $Re = 10^3$ and $Nnode=13$

a) System matrix, GWS $k = 1$ ($r = 1 = c$, (5.55))

```

inte:  emat, el#  1
form:  system matrix [D + B]  column format
      0.000000000E+00 0.100000000E+01 0.000000000E+00
      -0.512000024E+00 0.239999890E-01 0.488000035E+00
      -0.512000024E+00 0.239999890E-01 0.488000035E+00
      -0.512000024E+00 0.239999890E-01 0.488000035E+00
      -0.512000024E+00 0.239999890E-01 0.488000035E+00
      -0.512000024E+00 0.239999890E-01 0.488000035E+00
      -0.512000024E+00 0.239999890E-01 0.488000035E+00
      -0.512000024E+00 0.239999890E-01 0.488000035E+00
      -0.512000024E+00 0.239999890E-01 0.488000035E+00
      -0.512000024E+00 0.239999890E-01 0.488000035E+00
      -0.512000024E+00 0.239999890E-01 0.488000035E+00
      -0.512000024E+00 0.239999890E-01 0.488000035E+00
      0.000000000E+00 0.100000000E+01 0.000000000E+00

```

Corresponding GWS system matrix condition number $K = 41.666$, Appendix L.

b) System matrix, $k = 2$ ($r = 1 = c$, (5.55))

```

form:  system matrix [D + B]  column format
0.000000000 0.000000000 1.000000000 0.000000000 0.000000000
0.000000000 -0.682666719 0.032000035 0.650666714 0.000000000
0.168666676 -0.682666659 0.028000087 0.650666714 -0.164666682
0.000000000 -0.682666719 0.032000035 0.650666714 0.000000000
0.168666676 -0.682666659 0.028000087 0.650666714 -0.164666682
0.000000000 -0.682666719 0.032000035 0.650666714 0.000000000
0.168666676 -0.682666659 0.028000087 0.650666714 -0.164666682
0.000000000 -0.682666719 0.032000035 0.650666714 0.000000000

```

Table 8.6 1d Singular perturbation solution (continued)

b) System matrix, GWS $k = 2$ (continued)

```
0.168666676 -0.682666659 0.028000087 0.650666714 -0.164666682
0.000000000 -0.682666719 0.032000035 0.650666714 0.000000000
0.168666676 -0.682666659 0.028000087 0.650666714 -0.164666682
0.000000000 -0.682666719 0.032000035 0.650666714 0.000000000
0.000000000 0.000000000 1.000000000 0.000000000 0.000000000
```

c) System matrix, SGM ($r = 125/3$ or $c = 62$, (5.50))

```
form:  system matrix [D + B]  column format
      0.000000000D+00 0.100000000D+01 0.000000000D+00
      -0.100000000D+01 0.100000000D+01-0.555111512D-15
      -0.100000000D+01 0.100000000D+01-0.555111512D-15
      -0.100000000D+01 0.100000000D+01-0.555111512D-15
      -0.100000000D+01 0.100000000D+01-0.555111512D-15
      -0.100000000D+01 0.100000000D+01-0.555111512D-15
      -0.100000000D+01 0.100000000D+01-0.555111512D-15
      -0.100000000D+01 0.100000000D+01-0.555111512D-15
      -0.100000000D+01 0.100000000D+01-0.555111512D-15
      -0.100000000D+01 0.100000000D+01-0.555111512D-15
      -0.100000000D+01 0.100000000D+01-0.555111512D-15
      -0.100000000D+01 0.100000000D+01-0.555111512D-15
      -0.100000000D+01 0.100000000D+01-0.555111512D-15
      0.000000000D+00 0.100000000D+01 0.000000000D+00
```

Corresponding SGM system matrix condition number $K = 1.0$, Appendix L.

d) Nodal solution, GWS $k = 1$ ($r = 1 = c$) on 13 node mesh

```
solv:  nodal solution (real variables)
      min =   -3.46000886   max =    1.00000000
0.0000000E+00-0.2630173E+01 0.1293527E+00-0.2765887E+01 0.2717414E+00
-0.2915279E+01 0.4284801E+00-0.3079726E+01 0.6010147E+00-0.3260746E+01
0.7909373E+00-0.3460009E+01 0.1000000E+01
norm:  f.e. soln. energy semi-norm:  e(qh)= 0.89176285E+00
```

Table 8.6 1d Singular perturbation solution (concluded)

e) Nodal solution, GWS $k = 2$ ($r = 1 = c$) on 13 node mesh

solv: nodal solution (real variables)

```

min =   -3.22475982   max =    1.00000000
0.0000000E+00-0.2809350E+01 0.1381649E+00-0.2880919E+01 0.2866446E+00
-0.2957829E+01 0.4462090E+00-0.3040482E+01 0.6176857E+00-0.3129305E+01
0.8019643E+00-0.3224760E+01 0.1000000E+01

```

norm: f.e. soln. energy semi-norm: $e(qh) = 0.11753497E+01$

f) Nodal solution, SGM f_2 ($r = 125/3$ or $c = 62$) on 13 node mesh

solv: nodal solution (real variables)

```

min =    0.00000000   max =    1.00000000
0.0000000D+00 0.1542349E-99 0.2778448E-99 0.5005208E-99 0.9016581E-99
0.1624283E-99 0.2926048D-91 0.5271099D-76 0.9495568D-61 0.1710569D-45
0.3081488D-30 0.5551115D-15 0.1000000D+01

```

norm: f.e. soln. energy semi-norm: $e(qh) = 0.25000000D+00$

The accurate prediction of the wall layer boundary flux is also very difficult for this problem. The error in boundary flux computation is at least two orders of magnitude higher for the standard GWS FEM solution, as compared to that for the f_2 SGM solution. Flux predictions from finite difference formulae always have higher error, than that from the WS^h , and the associated error does not decrease asymptotically unless the solution is monotone.

For $Re = 50$, computational data confirming these assertions are presented in the last three columns of Table 8.4. The standard FEM linear and quadratic WS^h solution, for $Re = 1000$, yields over 200% error in flux computation even on a highly non-uniform mesh. In distinction, for the theoretically-optimal SGM form, the boundary

flux calculated from the weak statement WS^h at both the left and right (Dirichlet) boundaries is accurate to machine precision, Figure 8.10. Comparison solution and end flux errors, for $Re = 1000$ on various Nnode meshes, are documented in Tables 8.7-8.8.

The quadratic f_2 SGM embedded parameter c can be theoretically determined to yield a monotone solution via (5.46). For example, the appropriate c for $Re=10,000$ on a uniform 5-node mesh ($h = 1/4$) is

$$c = \frac{1}{2} \left(\frac{3hRe}{2} - 1 \right) = \frac{1}{2} \left(\frac{30000}{8} - 1 \right) = 1874.5 \quad (8.20)$$

Figure 8.11a-b confirms the monotone, nodally exact SGM solutions attainable for a range of Re for different mesh measure h_e and corresponding selection of c via (5.46).

Table 8.7 Companion of boundary fluxes

Flux computation	Standard GWS				SGM f_2	
	$k = 1$		$k = 2$		$c = 62$ or $r = \frac{125}{3}$	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$	$x = 0$	$x = 1$
1st order FD	-3.16E-02	5.35E-02	-3.37E-02	5.07E-02	0.	1.80D-02
2nd order FD	6.39E-02	0.105786	6.83E-02	1.00E-01	0.	1.20D-02
WS^h	1.34665	-1.17648	1.94115	-1.71630	0.	1.00000

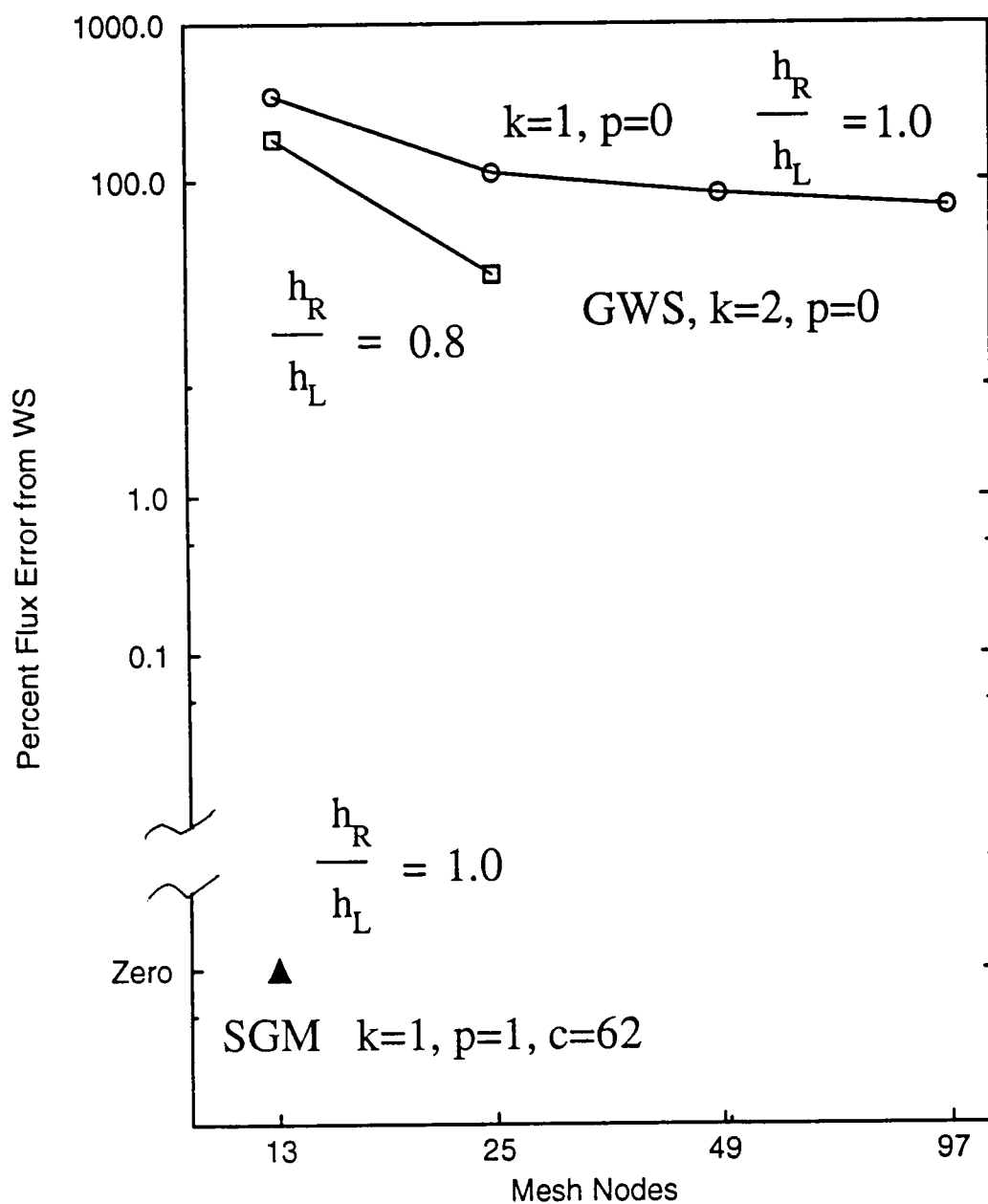
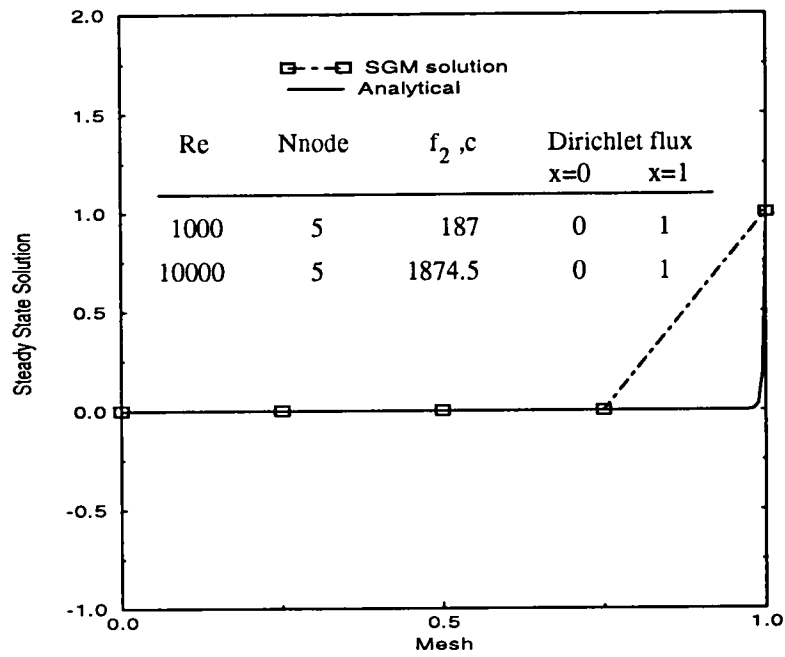


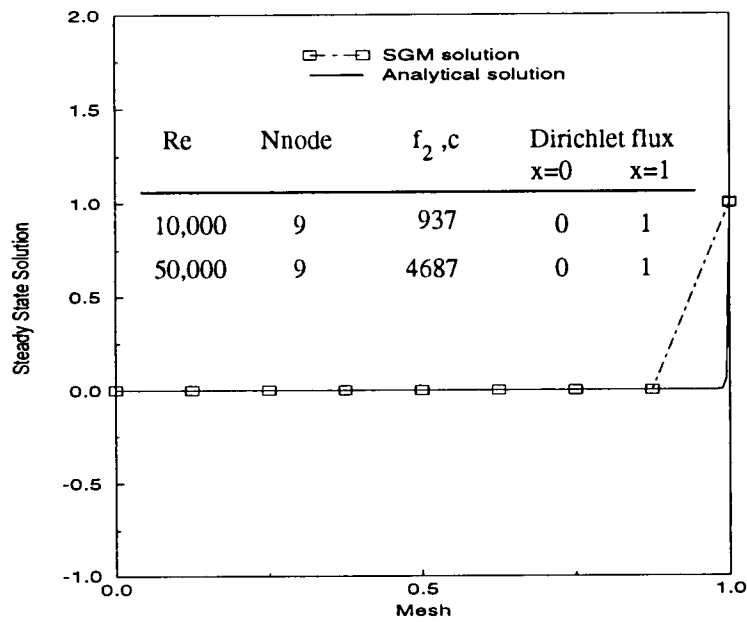
Figure 8.10 Boundary fluxes from WS^h , parabolic perturbation ($Re = 1000$) steady state solution.

Table 8.8 1d Singular perturbation solution and surface flux errors, Re=1000

k	p	Nnode $\left(\frac{h_R}{h_L}\right)$	$\frac{q_{R-1} - q_{R-1}^a}{q_R} \%$	$\frac{\ e^h\ _E}{\ q\ _E} \%$	Dirichlet flux error, %		
					WS^h	second order	first order
1	0	13 (1)	-346.0	257.000	-218.0	-89.4	-94.7
1	0	25 (1)	-112.0	22.100	-99.1	-90.1	-94.9
1	0	49 (1)	-82.5	.200	-82.5	-83.3	-91.2
1	0	97 (1)	-67.8	.002	-67.8	-70.4	-83.9
1	0	13 (0.8)	-184.0	1143.000	-126.0	-73.3	-84.6
1	0	25 (0.8)	-25.5	$\sim 10^{-4}$	-15.5	-15.9	-37.2
2	0	13 (1)	-132.0	370.000	-272.0	-89.9	-94.9
2	0	25 (1)	-78.0	432.000	-110.0	-91.7	-95.7
2	0	49 (1)	-42.9	1.990	-82.2	-86.9	-93.1
2	0	97 (1)	-35.9	.004	-66.6	-76.0	-86.9
2	0	13 (0.8)	-171.0	1798.000	-156.0	-75.0	-86.9
2	0	25 (0.8)	-24.0	.090	-19.1	-20.3	-43.1
1	1	13 (1)	0.00	0.000	0.0	-98.2	-98.8



a) 5-node mesh, $h_e = 1/4$,



b) 9-node mesh, $h_e = 1/8$

Figure 8.11 Verification of optimal SGM parabolic parameter c for different Re with various mesh size.

The velocity inside the physical domain is, in general, not a uniform constant. Choosing a linearly varying velocity distribution $u = mx + g$, m and g are constants, the general solution to the parabolic perturbation problem (8.16) becomes

$$q(x) = e^{-h} \left[\int e^h r(x) dx + C \right], \quad \text{where} \quad h = - \int \text{Re} (mx + g) dx, \quad (8.21)$$

$$\text{and} \quad r(x) = (mx + g)q - \frac{1}{\text{Re}} \frac{dq}{dx} \quad (8.22)$$

Assuming $\text{Re} \gg 1$, $\left. \frac{dq}{dx} \right|_{x=0} = 0 \Rightarrow r(x) = 0$. Finally, for $q(x=0) = 0$ and $q(x=1) = 1$, the analytical solution to (8.16) is

$$q(x) = \frac{e^{\text{Re} (mx^2/2 + gx)} - 1}{e^{\text{Re} (m/2 + g)} - 1} \quad (8.23)$$

Specifically, for $m = -1$ and $g = 1$, $u = U = 1 - x$, (8.23) yields

$$q(x) = \frac{e^{\text{Re} (x - x^2/2)} - 1}{e^{\text{Re}/2} - 1} \quad (8.24)$$

Hence, for a large Reynolds number, the solution has a sharp singularity at $x = 1$. The solution output of a variable velocity parabolic perturbation problem (Figure 8.12), is shown in Table 8.9. Although the GWS solution oscillation, Figure 8.12a, is annihilated in companion TWS algorithm by the use of artificial diffusion $\beta = 0.05$, Figure 8.12b, and $\beta = 0.2$, Figure 8.12c, the solutions suffer from excessive diffusion. For $\text{Re} = 10^5$, the comparison SGM solution is computed from the nodewise distributed SGM^h parameter r_i (5.55) and is nodally exact, Figure 8.12d.

Table 8.9 Solution output data for 1D singular perturbation problem with $u = 1 - x$

```

$ Velocity distribution U = 1-x
1.000000E+00 8.750000E-01 7.500000E-01 6.250000E-01 5.000000E-01
3.750000E-01 2.500000E-01 1.250000E-01 0.000000E+00
$ Mesh (0,1) Nnode=9
0.000000E+00 1.250000E-01 2.500000E-01 3.750000E-01 5.000000E-01
6.250000E-01 7.500000E-01 8.750000E-01 1.000000E+00

Steady state solutions
-----
a) GWS, beta=0.0, rsgm_e = 0
$ Solution Q
0.000000E+00-2.755283E-01 2.766394E-02-3.113430E-01 7.629281E-02
-3.821176E-01 1.913907E-01-6.125788E-01 1.000000E+00

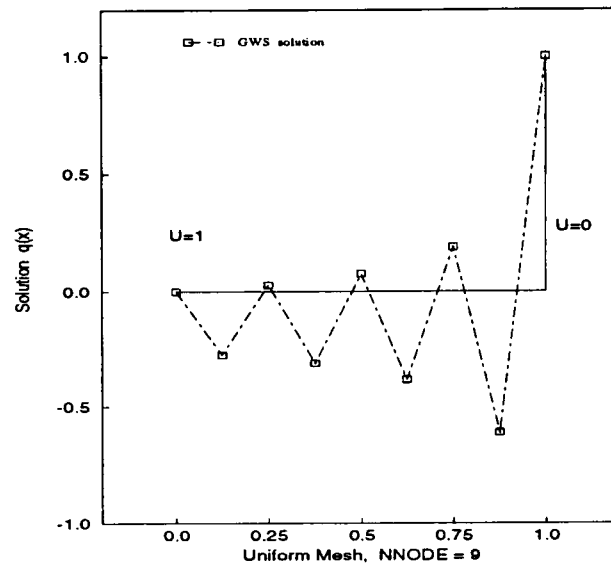
b) TWS, beta=0.05, rsgm_e = 0
$ Solution Q
0.000000E+00-2.762217E-06 1.401914E-04 2.618310E-03 1.938271E-02
8.519070E-02 2.567253E-01 5.731795E-01 1.000000E+00

c) TWS, beta=0.20, rsgm_e = 0
$ Solution Q
0.000000E+00-5.718773E-05 5.434172E-05-1.914627E-04 4.580691E-04
-1.869535E-03 1.453894E-02 2.887262E-01 1.000000E+00

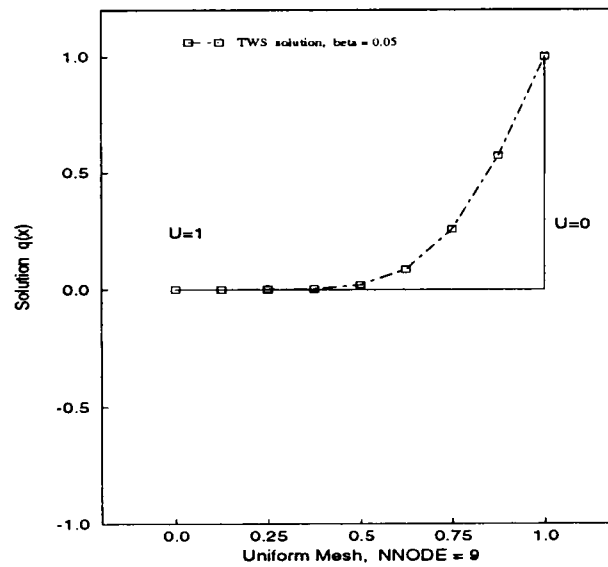
d) TWS, beta=0.25, rsgm_e = 0
$ Solution Q
0.000000E+00 6.626958E-06-1.351868E-05 6.439238E-05-3.893724E-04
6.957082E-03 9.675327E-02 4.206050E-01 1.000000E+00

e) SGM, beta=0.0, rsgm_e = {RSGM}
$ Solution Q
0.000000E+00 9.205837E-15 1.184624E-12 1.663812E-10 1.391074E-08
5.428368E-07 1.434577E-06 7.006777E-05 1.000000E+00
$ Distributed SGM (RSGM)
6.250000E+03 5.700000E+03 4.900000E+03 4.100000E+03 3.400000E+03
2.700000E+03 1.900000E+03 8.900000E+02 1.750000E+02

```

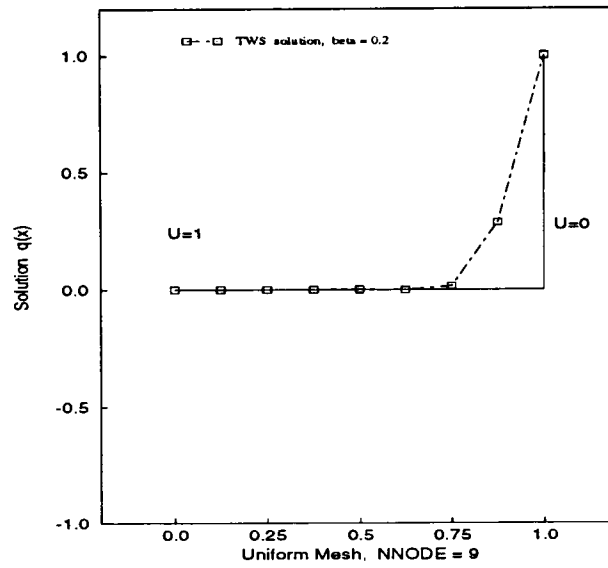


a) $GWS^h, k=1, p=0, \beta=0$

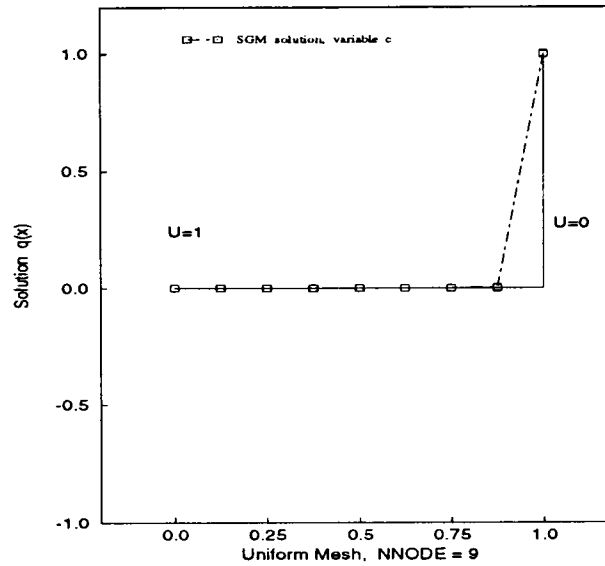


b) $TWS^h, k=1, p=0, \beta=0.05$

Figure 8.12 Comparison of GWS^h , TWS^h and SGM algorithm for $Re = 10^5$, $U = (1-x)$ parabolic perturbation solution in one dimension.



c) TWS^h, $k = 1$, $p = 0$, $\beta = 0.2$



d) SGM, $k = 1$, $p = 0$, $r_i \geq 1$

Figure 8.12 Comparison of GWS^h, TWS^h and SGM algorithm for $Re = 10^5$
(concluded).

8.2.6 Parabolic perturbation verification in two dimensions:

The $d = 2$ form of (8.16) is

$$\frac{\partial}{\partial x_j} \left(u_j q - \epsilon \frac{\partial q}{\partial x_j} \right) = 0 \quad 1 \leq x_j \leq \ell \text{ and } 1 \leq j \leq 2 \quad (8.25)$$

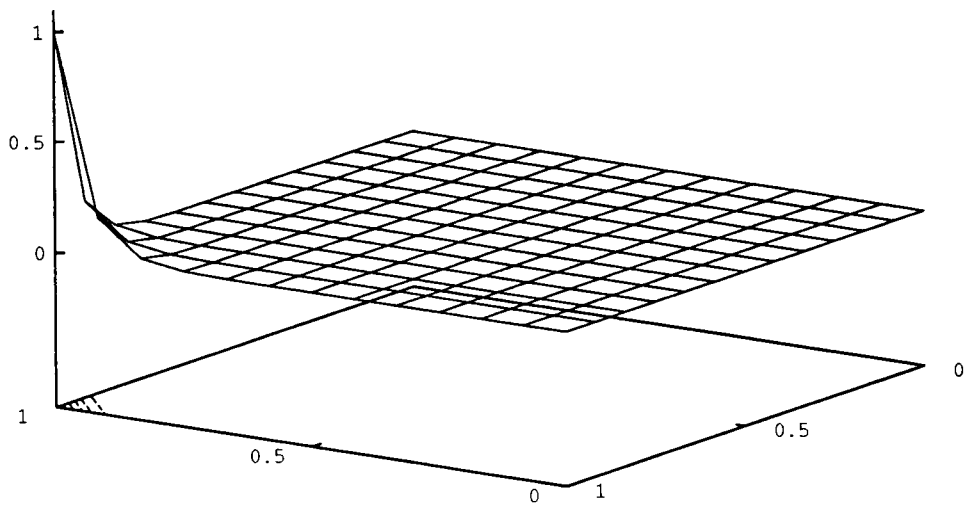
and the nondimensional analytical solution on $0 \leq x, y \leq 1$ for $u_1 = u_2 = 1$ and corresponding Dirichlet data $q(0, y) = 0 = q(x, 0)$, and $q(1, 1) = 1$, is

$$q(x) = \left[\frac{e^{x\text{Re}} - 1}{e^{\text{Re}} - 1} \right] \left[\frac{e^{y\text{Re}} - 1}{e^{\text{Re}} - 1} \right] \quad (8.26)$$

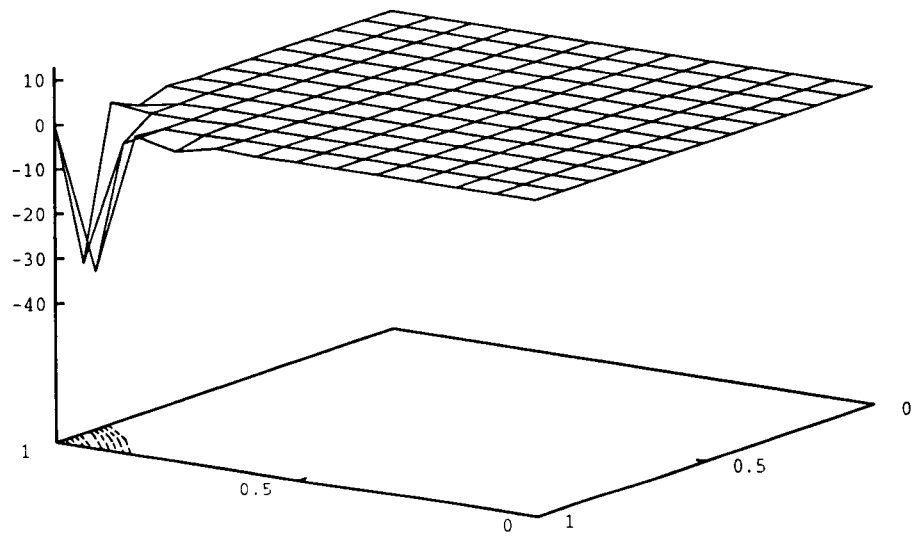
As $\epsilon \rightarrow 0$, i.e., Re becomes large, the analytical solution is smooth but essentially point-singular in the near region $(x, y) \sim 1$. For $\text{Re}=20$, the analytical solution (8.26) interpolated onto an $\text{Nnode} = 13 \times 13$ uniform mesh is shown in Figure 8.13a.

The error distributions for the standard GWS $k = 1$, $k = 2$ ($p = 0$) and the SGM $k = 1$, $p = 1$, computations are presented in Figure 8.13b-d. The local approximation dispersive error in linear standard GWS ($k = 1$, $p = 0$) solution exceeds 30% in the near vicinity of the singular point, Figure 8.13b, while for quadratic GWS ($k = 2$, $p = 0$) the approximation error in this region is almost 27%, Figure 8.13c. Both GWS solutions are clearly non-monotone on this mesh, even for the modest $\text{Re} = 20$. In distinction, for the SGM, $k = 1$, $p = 1$ quadratic f_2 choice, with (optimal) $c_x = c_y = 1.3$, the 13×13 node uniform mesh solution is monotone with local error $\sim 10^{-2}\%$, Figure 8.13d.

For $\text{Re} = 2000$ on a $\text{Nnode} = 9 \times 9$ uniform mesh, the comparison error distributions for the standard GWS ($k = 1$, $p = 0$) and the SGM ($k = 1$, $p = 1$), computations are presented in Figure 8.14. The standard GWS solution error

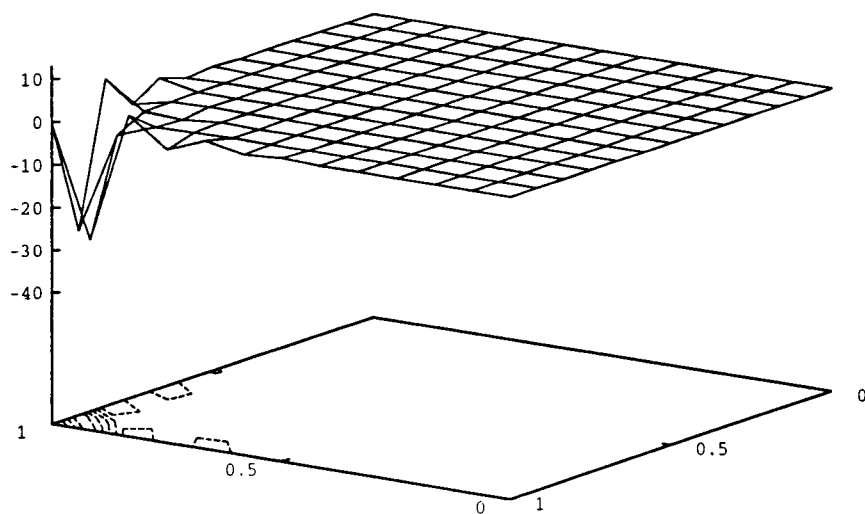


a) Analytical solution $q(x, y)$

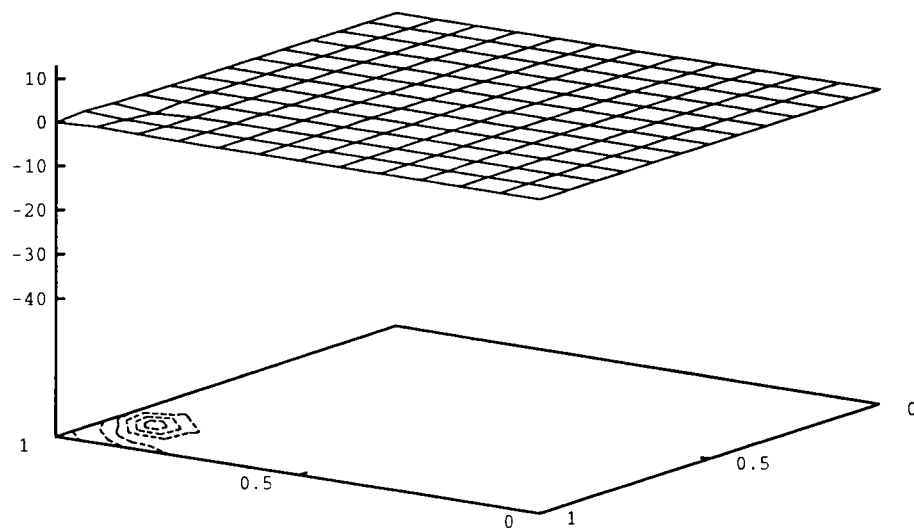


b) Normalized error, GWS, $k = 1$, $p = 0$

Figure 8.13 2-D Parabolic perturbation on an $N_{\text{node}} = 13^2$ uniform mesh, $\text{Re} = 20$.



c) Normalized error, GWS, $k = 2, p = 0$



d) Normalized error, SGM, $k = 1, p = 1$

Figure 8.13 2-D Parabolic perturbation (concluded).

extremum is nearly 55%, which is propagated throughout the entire domain, Figure 8.14a. Conversely, for optimal f_2 choice $c_x = c_y = 38$, the SGM solution error is $\mathcal{O}10^{-2}$, Figure 8.14b. Finally, the essentially nodally-exact SGM f_2 solution for $\text{Re} = 20,000$ on a 9×9 uniform mesh is shown in Figure 8.15, as obtained with directional subgrid embedding parameters $c_x = c_y = 124$. The extremum nodal error is again $\mathcal{O}10^{-2}$.

8.2.7 1d Linear Stationary Wave ($u \neq 0$, $\epsilon \rightarrow 0$)

The governing PDE remains

$$u \frac{dq}{dx} - \frac{d}{dx} \left(\epsilon \frac{dq}{dx} \right) = 0 \quad \text{on } 0 \leq x \leq \ell \quad (8.27)$$

Implementing Reynolds number $\text{Re} = u\ell/\epsilon$, the non-dimensional form of (8.27) becomes

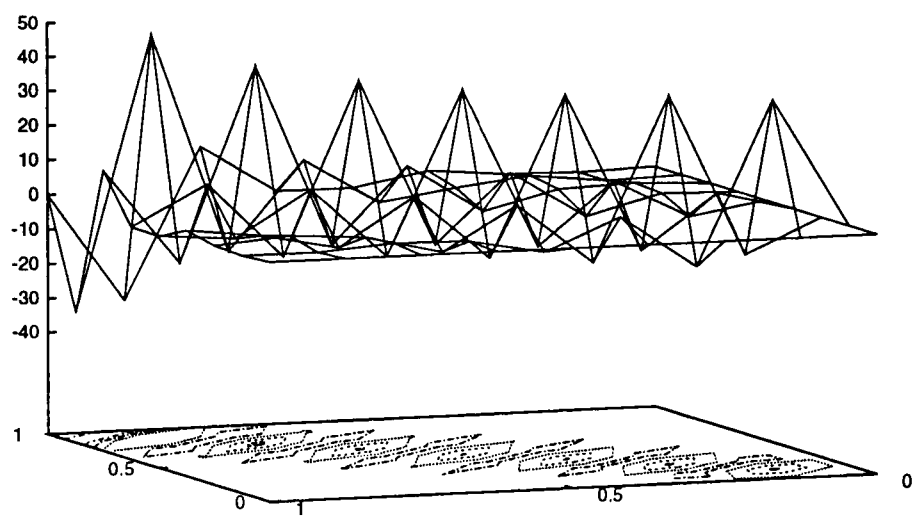
$$u \frac{dq}{dx} - \frac{d}{dx} \left(\frac{1}{\text{Re}} \frac{dq}{dx} \right) = 0 \quad \text{on } 0 \leq x \leq 1 \quad (8.28)$$

The analytical solution to (8.28) for the Dirichlet boundary conditions $q(0) = -1$ and $q(1) = 1$ is

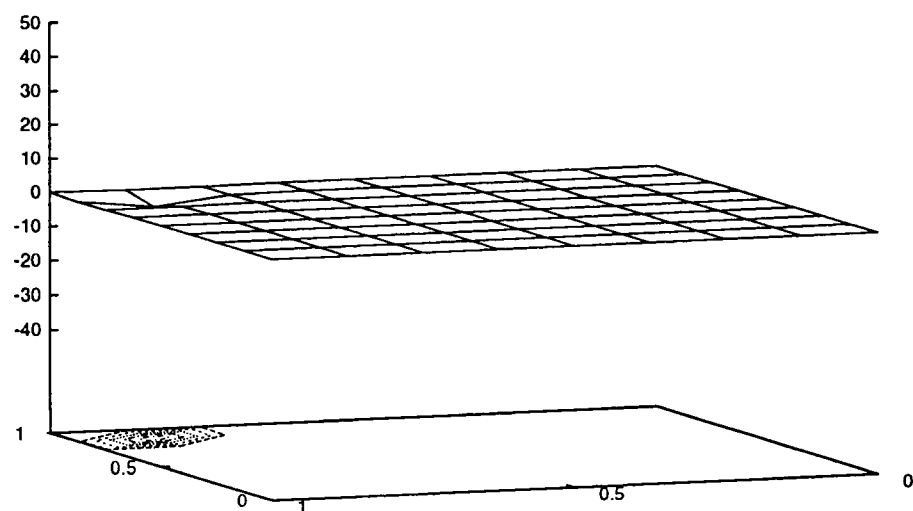
$$q(x) = \frac{1 + e^{u\text{Re}} - 2e^{xu\text{Re}}}{1 - e^{u\text{Re}}} \quad (8.29)$$

Figure 8.16 documents the solution schematic for the linear stationary wave, (8.29). As Re becomes large, $q(x)$ approaches a function with unity behind the front trailing $u = -1$ and $q = -1$ behind the $u = 1$ front. The slope of the steady state solution q at front coalescence is

$$\frac{dq}{dx} = -\frac{2u\text{Re} e^{xu\text{Re}}}{1 - e^{u\text{Re}}} \quad \text{as } \text{Re} \rightarrow \infty, \quad \frac{dq}{dx} \text{ approaches } +\text{Re}, \text{ i.e. } \infty \quad (8.30)$$



a) Normalized error, GWS, $k = 1$, $p = 0$



b) Normalized error, SGM, $k = 1$, $p = 1$, $f_2 c = 38$

Figure 8.14 2-D Parabolic perturbation on an $N_{\text{node}} = 9^2$ uniform mesh, $\text{Re} = 2000$.

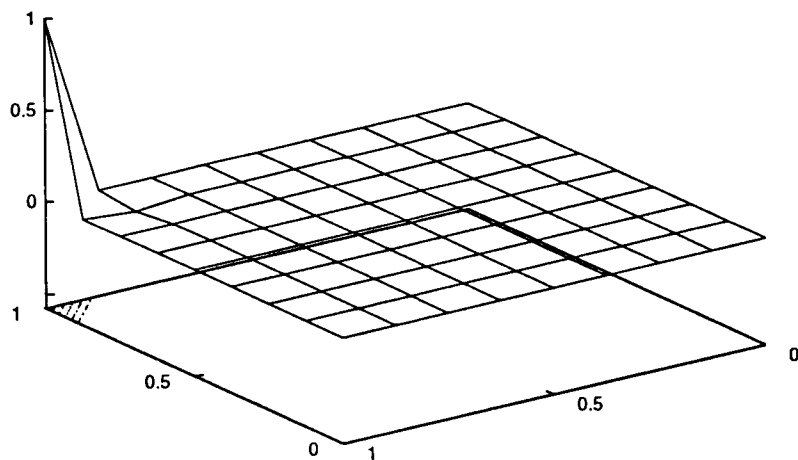


Figure 8.15 SGM $f_2(x)$ solution of a two dimensional parabolic perturbation (Re =20,000) on an Nnode = 9^2 uniform mesh for $c_x = c_y = 124$.

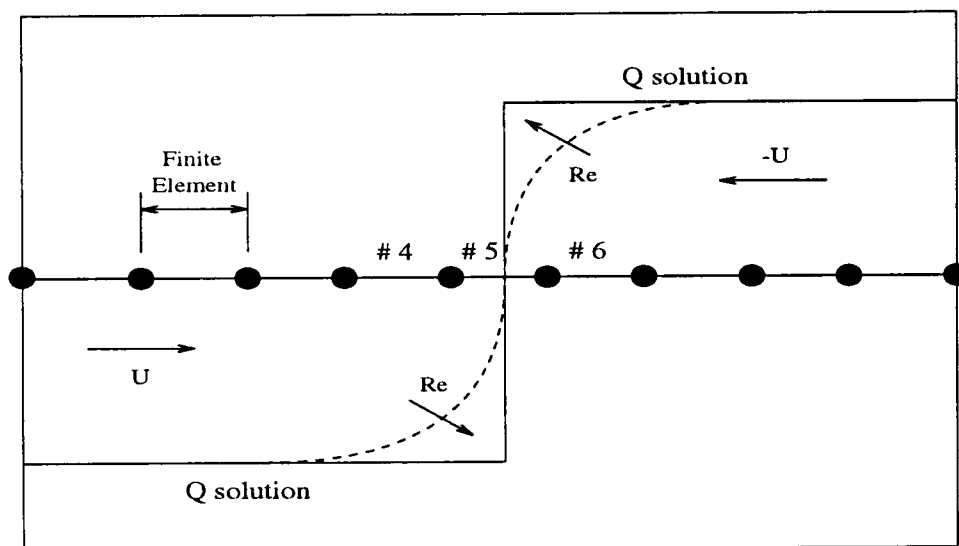


Figure 8.16 Linear stationary wave solution schematic.

Clearly, for large Re , as $|u| \rightarrow 0$ at the wave intersection, $|q|$ approaches 0 and the solution approaches a step-function in the region where u changes sign.

For this verification case, the standard GWS solution with $k = 1$, $p = 0$ for $\text{Re}=10^6$ is totally non-monotone on a uniform 10-node mesh, Figure 8.17a. In fact, the WS^h solution remains totally dispersive on meshes up to 1000 nodes. In distinction, the comparison SGM f_2 optimal choice r in (5.50), for $\text{Re} = 10^6$ and on the uniform 10 node mesh, yields an absolutely monotone nodally-exact solution, Figure 8.17b. For this problem definition, $\mathcal{F} = 6$ in (5.55) and the theoretical prediction of the optimal subgrid embedding parameter corresponding to $\text{Re} = 10^6$ and $h = 1/9$ is

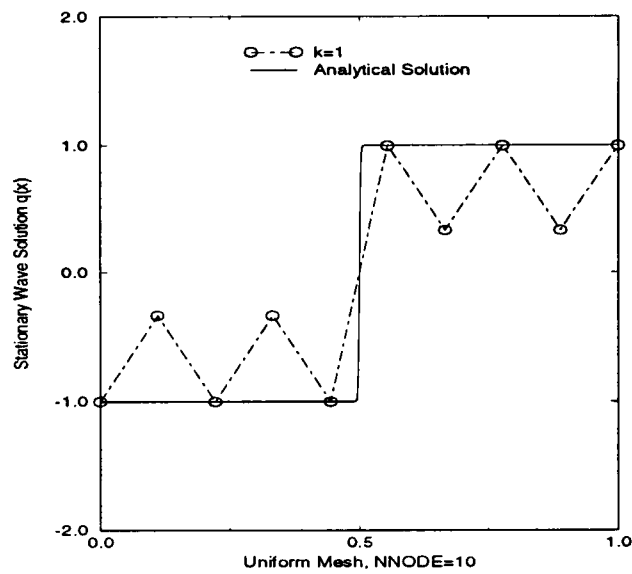
$$r = \frac{2c + 1}{3} = \frac{1000000}{6 \times 9} = 18518.519, \text{ hence } c = 27777.278 \quad (8.31)$$

Table 8.10-8.11 documents the comparison element and system matrices for standard GWS and SGM f_2 choice, and the corresponding nodal solutions to (8.27) for the computational domain shown in Figure 8.16. Via (8.30), the analytical end fluxes for (8.28) are exactly zero for any (large) Re . While the standard GWS solution predictions for these fluxes are highly erroneous (almost three orders of magnitude error), the SGM f_2 choice yields end fluxes accurate to machine precision.

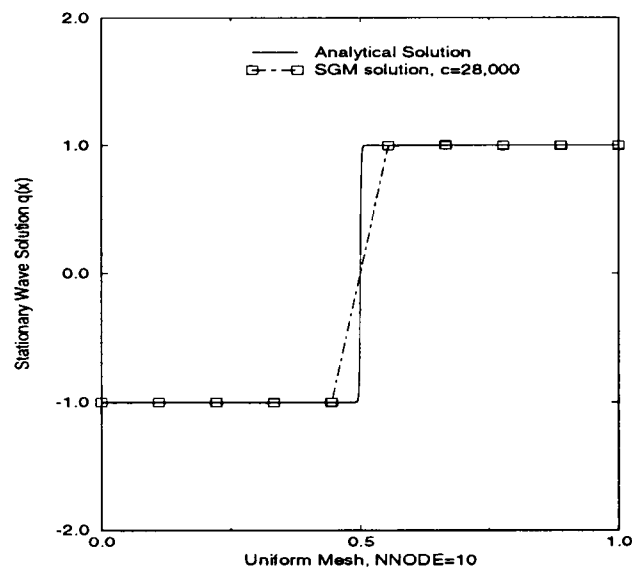
8.2.8 1d non-linear Burgers equation, stationary wave ($u \neq 0$, $\epsilon \rightarrow 0$)

The non-linear form of (8.27) results upon the definition of $q = u$, hence

$$u \frac{du}{dx} - \frac{d}{dx} \left(\epsilon \frac{du}{dx} \right) - s = 0 \quad (8.32)$$



a) GWS $k = 1, p = 0$



b) SGM f_2 $k = 1, p = 1, c = 27777.278$

Figure 8.17 1d Linear square wave steady solution, 10-node mesh, $Re = 10^6$.

Table 8.10 Example element matrices for standard GWS and SGM f_2 computation

Element number	Standard GWS ($c = 1$)	SGM f_2 ($c = 27777.278$)
# 4	$\frac{1}{2} \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}$	$\frac{1}{3} \begin{bmatrix} -1 & 1 \\ -2 & 2 \end{bmatrix}$
# 5	$\frac{1}{6} \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}$	$\frac{4}{3 \times 10^9} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$
# 6	$\frac{1}{2} \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$	$\frac{1}{3} \begin{bmatrix} 2 & -2 \\ 1 & -1 \end{bmatrix}$

Table 8.11 1d Stationary wave solution comparison for $Re = 10^6$ and $Nnode=10$

a) System matrix for standard GWS ($c = 1 = r$)

```

form:  system matrix [D + B]  column format
      0.000000000E+00 0.100000000E+01 0.000000000E+00
     -0.500009000E+00 0.180006027E-04 0.499991000E+00
     -0.500009000E+00 0.180006027E-04 0.499991000E+00
     -0.500009000E+00 0.180006027E-04 0.499991000E+00
     -0.500009000E+00 0.333351314E+00 0.166657671E+00
      0.166657671E+00 0.333351314E+00 -0.500009000E+00
      0.499991000E+00 0.180006027E-04 -0.500009000E+00
      0.499991000E+00 0.180006027E-04 -0.500009000E+00
      0.499991000E+00 0.180006027E-04 -0.500009000E+00
      0.000000000E+00 0.100000000E+01 0.000000000E+00

```

Table 8.11 1d Stationary wave solution (continued)

b) System matrix for SGM ($r = 18518.519$ or $c = 27777.278$)

form: system matrix [D + B] column format

```

0.000000000D+00 0.100000000D+01 0.000000000D+00
-0.666666668D+00 0.333333336D+00 0.333333332D+00
-0.666666668D+00 0.333333336D+00 0.333333332D+00
-0.666666668D+00 0.333333336D+00 0.333333332D+00
-0.666666668D+00 0.666666669D+00-0.133333336D-08
-0.133333336D-08 0.666666669D+00-0.666666668D+00
0.333333332D+00 0.333333336D+00-0.666666668D+00
0.333333332D+00 0.333333336D+00-0.666666668D+00
0.333333332D+00 0.333333336D+00-0.666666668D+00
0.000000000D+00 0.100000000D+01 0.000000000D+00

```

c) Solution for standard GWS algorithm ($c = 1 = r$)

```

min = -1.00004792 max = 1.00004792
-0.1000000E+01-0.3339844E+00-0.1000024E+01-0.3339844E+00-0.1000048E+01
0.1000048E+01 0.3333973E+00 0.1000024E+01 0.3334213E+00 0.1000000E+01

```

d) Solution for SGM algorithm ($r = 18518.519$ or $c = 27777.278$)

```

min = -1.00000000 max = 1.00000000
-0.1000000D+01-0.1000000D+01-0.1000000D+01-0.1000000D+01-0.1000000D+01
0.1000000D+01 0.1000000D+01 0.1000000D+01 0.1000000D+01 0.1000000D+01

```

e) Boundary Fluxes (real variables)

Standard FEM k=1		SGM f2, c=27777.278	
left end, x=0	right end, x=1	left end, x=0	right end, x=1
0.666986	-0.666705	-1.6667D-10	1.6667D-10

The nondimensional form of (8.32) is the $d = 1$ stationary viscous Burgers equation

$$u \frac{du}{dx} - \frac{d}{dx} \left(\frac{1}{\text{Re}} \frac{du}{dx} \right) - s = 0 \quad (8.33)$$

For $u = 1$ at $x = -1$ and $u = -1$ at $x = 1$, the $s = 0$ analytical solution to (8.33) is

$$u = \tanh \left(\frac{-x \text{Re}}{2} \right) \quad (8.34)$$

The corresponding slope of the solution is

$$\frac{du}{dx} = -\frac{\text{Re}}{2} \left(1 - \tanh^2 \left(\frac{\text{Re } x}{2} \right) \right) \text{ and as } x \rightarrow 0, \frac{du}{dx} \text{ approaches } -\frac{\text{Re}}{2} \quad (8.35)$$

Clearly, as $\text{Re} \rightarrow \infty$, the slope approaches $-\infty$. A detailed discussion of standard GWS solution procedures for the unsteady Burgers equation may be found in Baker (1983, Ch. 4.14).

The solution domain schematic is shown in Figure 8.18. As Re becomes large, (8.32) loses its parabolicity and becomes hyperbolic. Hence, the corresponding iterative algebraic solution process for the GWS^h of (8.33) becomes highly unstable, as the dispersive error mode totally corrupts the entire solution process. Figure 8.19a-b illustrates the nonlinear instability present for the $k = 1, 2$ and $p = 0$, GWS^h algorithm, for $\text{Re} = 10^5$ and stopping the iterative process at $\varepsilon < 0.001$. Although the quadratic basis standard GWS solution has somewhat reduced oscillations (25%), any GWS^h solution remains unsatisfactory even for a 2000 node uniform mesh. The iterative solution process is unstable, and all WS^h cases eventually diverge for standard GWS $k = 1, 2$ with $p = 0$ when iterative convergence is set below $\varepsilon < 10^{-4}$.

In distinction, the SGM f_2 optimal choice of uniform $c = 5500$, for an 11-node uniform mesh yields a monotone, essentially nodally exact solution. Figure 8.19c illustrates the approach to this solution for $\epsilon < 0.001$, while Figure 8.19d shows it at full (stable) convergence. Table 8.12 compares the standard GWS and SGM algorithm element matrices in the region where the analytical solution changes sign. The corresponding SGM solution iteration process is stable and convergent to $\epsilon \leq 10^{-18}$.

At this point, I introduce the solution adaptive SGM f_2 algorithm, (i.e., SGM^h) hence $c \Rightarrow c_e$, i.e., element-dependent data. Via (5.55), the nodally distributed SGM^h parameter r_j can be written as

$$r_j = \frac{2c_j + 1}{3} \geq \frac{h_j |U_j|}{\mathcal{F} \epsilon_j} \quad (8.36)$$

The *Weak Statement* WS^h is applied to (8.36), for implimenting the SGM^h algorithm. For a symmetric ramp initial condition, Figure 8.18, the SGM^h solutions are plotted in Figure 8.19d-f. The convergence criteria for this case is $\epsilon \leq 10^{-8}$ for which the standard GWS^h linear and quadratic solution processes are strongly divergent. On an 11 node uniform mesh, the SGM^h solution matches exactly the analytical solution, Figure 8.19d. The SGM^h nodal solution remains nodally exact for an 11 node, centrally attracted non-uniform mesh, Figure 8.19e.

Finally, to test the mesh sensitivity, I replaced the 11 node mesh with 10 nodes, hence no central node should appear in the step. Corresponding 10 node uniform mesh SGM^h solution is shown in Figure 8.19f. Here, unlike the previous cases, the central node is absent. Nevertheless, the corresponding SGM^h solution for $r_j \geq 1$ is

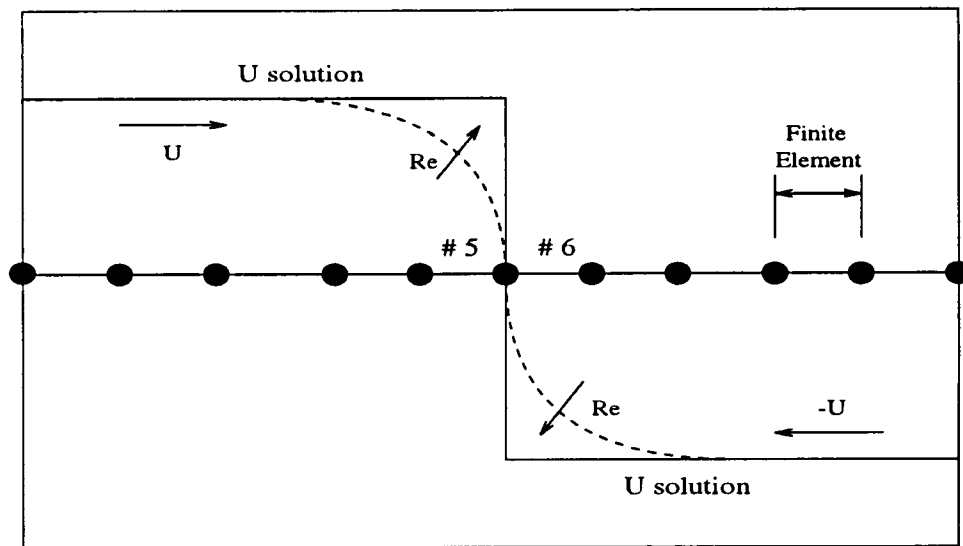
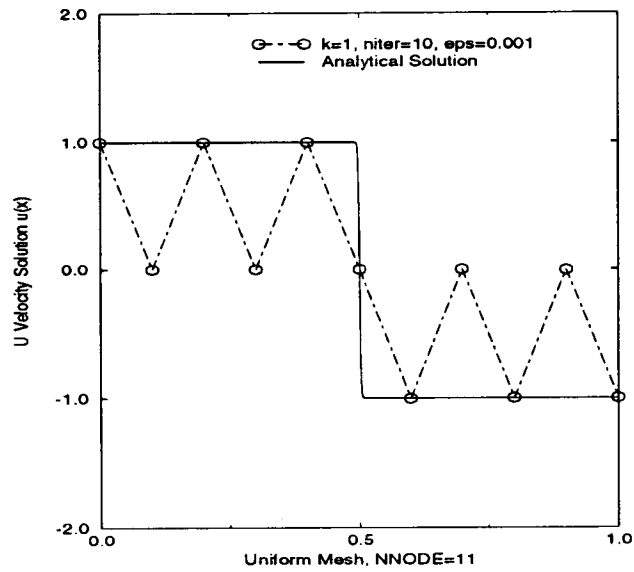


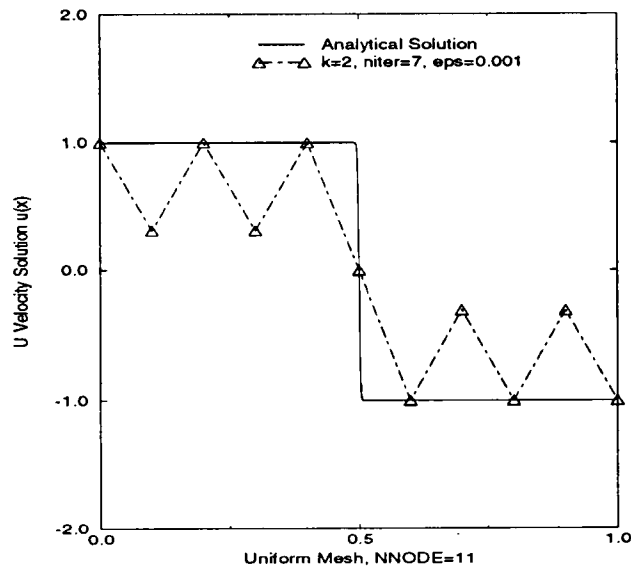
Figure 8.18 Nonlinear Burgers equation solution domain schematic

Table 8.12 Element matrices for Burgers equation algorithms, 11 node mesh, $Re = 10^5$

El. no.	It. no.	Standard GWS $k = 1 (r = 1)$	SGM f_2 $r = 100000/3$	Comment
5	1	-0.167 0.167	0.166678 -0.166678	GWS matrices are not Positive definite SGM f_2 matrices are Positive definite
		-0.083 0.083	-0.416678 0.416678	
6	1	0.083 -0.083	0.416678 -0.416678	
		0.167 -0.167	-0.166678 0.166678	
5	6	-0.325 0.325	$3.25D - 04$ $-3.25D - 04$	GWS diverges SGM converges
		-0.163 0.163	-0.4999 0.4999	
6	6	0.163 -0.163	0.4999 -0.4999	
		0.325 -0.325	$-3.25D - 04$ $3.25D - 04$	

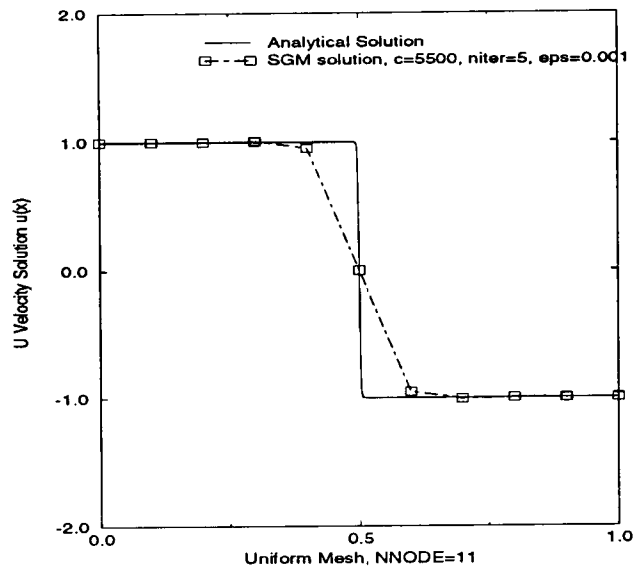


a) $k = 1, p = 0$, uniform 11-node mesh ($\varepsilon = 0.001$)

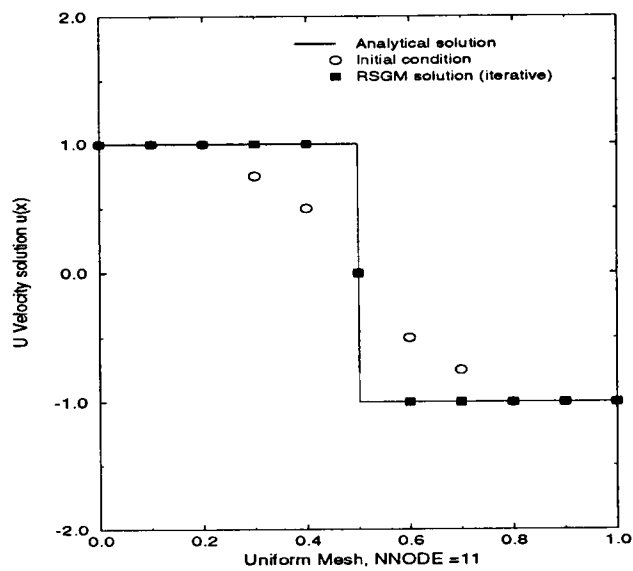


b) $k = 2, p = 0$, uniform 11-node mesh ($\varepsilon = 0.001$)

Figure 8.19 Monotonicity study for non-linear Burgers solution, $Re = 100,000$.

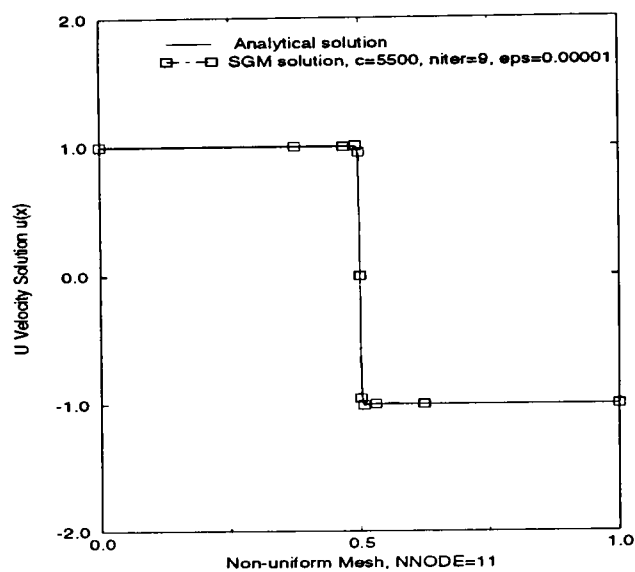


c) $k = 1$, $p = 1$, uniform 11-node mesh ($\epsilon = 0.001$)

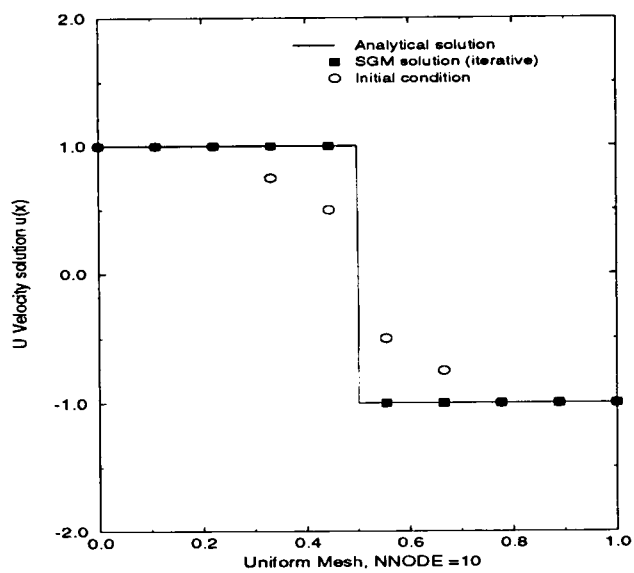


d) $k = 1$, $p = 1$, uniform 11-node mesh ($\epsilon = 10^{-8}$)

Figure 8.19 Non-linear Burgers solution, $Re = 100,000$ (continued).



e) $k = 1$, $p = 1$, non-uniform 11-node mesh ($\epsilon = 10^{-8}$)



f) $k = 1$, $p = 1$, uniform 10-node mesh ($\epsilon = 10^{-8}$)

Figure 8.19 Non-linear Burgers solution, $Re = 100,000$ (concluded).

nodally exact (to $\mathcal{O}10^{-8}$). A comparison of the GWS^h and SGM^h element matrices and the corresponding solutions are documented in Table 8.12. Also, the computed values of SGM^h parameter $r_j \equiv r_{sgm_j}$ for each node j during the solution iteration is listed in Table 8.13.

Table 8.13 Solution output for Burgers equation

a) Uniform 11 node mesh

```

-----
$ 1.000000E+00 X1 ( 1) 11 11 1 521 0
0.000000E+00 1.000000E-01 2.000000E-01 3.000000E-01 4.000000E-01
5.000000E-01 6.000000E-01 7.000000E-01 8.000000E-01 9.000000E-01
1.000000E+00

```

b) Initial condition :

```

-----
$ 0.000000E+00 U1 ( 1) 11 0 1 224 FSET 1
1.000000E+00 1.000000E+00 1.000000E+00 7.500000E-01 5.000000E-01
0.000000E+00-5.000000E-01-7.500000E-01-1.000000E+00-1.000000E+00
-1.000000E+00

```

c) Solution History Synopsis :

```

-----
1 1 2 2 3 1 RSGM SOLVER = >GAUSS-ELIM< INT = 1
2 2 2 1 3 1 U1 SOLVER = >GAUSS-ELIM< INT = 1
1 1 MAX DQ 0.00000E+00 -1.12411E+00
1 2 MAX DQ 0.00000E+00 7.55558E-01
:
1 24 MAX DQ 0.00000E+00 1.00716E-07
1 25 MAX DQ 0.00000E+00 -5.02370E-08

```

d) Final Solution :

```

-----
$ STEP 1, IT: 25, NITTOT 25, TH: 1.0, DT:1.000E+00, STA: 1.0
$ -1.12411E+00 U ( 1) 11 7 1 224 FSET 1
1.000000E+00 1.000000E+00 1.000000E+00 9.999999E-01 1.000001E+00
1.605001E-13 -1.000001E+00 -9.999999E-01 -1.000000E+00 -1.000000E+00
-1.000000E+00

```

Table 8.12 Solution output for Burgers equation (continued)

```
$ 3.33330E+03 RSGM( 1) 11 1 1 1 1.0000E+00 0
3.333300E+03 3.333300E+03 3.333300E+03 3.333298E+03 3.333311E+03
3.333311E+03 3.333311E+03 3.333298E+03 3.333300E+03 3.333300E+03
```

```
$ 5.02370E-08 Solution U Residual
0.000000E+00-4.830885E-10 1.932295E-09-1.014435E-08 5.023701E-08
1.543006E-14-5.023704E-08 1.014436E-08-1.932296E-09 4.830887E-10
0.000000E+00
```

e) Values of RSGM at selected iteration point

```
-----
Iteration # 1
$ -1.12411E+00 U1 ( 1) 11 7 1 224 FSET 1
1.000000E+00 9.982123E-01 1.007151E+00 1.038045E+00 1.624114E+00
-3.851849E-14-1.624114E+00-1.038045E+00-1.007151E+00-9.982123E-01
-1.000000E+00
$ 3.33330E+03 RSGM( 1) 11 10 1 1 1.0000E+00
3.333300E+03 3.333300E+03 3.333300E+03 2.499975E+03 1.666650E+03
1.666650E+03 1.666650E+03 2.499975E+03 3.333300E+03 3.333300E+03
3.333300E+03
```

```
Iteration # 2
$ -1.12411E+00 U1 ( 1) 11 7 1 224 FSET 1
1.000000E+00 1.004775E+00 9.810226E-01 1.099485E+00 8.685563E-01
-4.949895E-15-8.685563E-01-1.099485E+00-9.810226E-01-1.004775E+00
-1.000000E+00
$ 3.33330E+03 RSGM( 1) 11 10 1 1 1.0000E+00
3.333300E+03 3.327341E+03 3.357135E+03 3.460117E+03 5.413660E+03
5.413660E+03 5.413660E+03 3.460117E+03 3.357135E+03 3.327341E+03
3.333300E+03
```

```
Iteration # 4
$ -1.12411E+00 U1 ( 1) 11 7 1 224 FSET 1
1.000000E+00 1.000409E+00 9.983788E-01 1.008692E+00 9.643929E-01
-1.722826E-13-9.643929E-01-1.008692E+00-9.983788E-01-1.000409E+00
-1.000000E+00
```


Table 8.12 Solution output for Burgers equation (continued)

```
$ 3.33330E+03 RSGM( 1) 11 10 1 1 1.0000E+00
3.333300E+03 3.330352E+03 3.346151E+03 3.295812E+03 3.601244E+03
3.601244E+03 3.601244E+03 3.295812E+03 3.346151E+03 3.330352E+03
3.333300E+03
```

Iteration # 6

```
$ -1.12411E+00 U1 ( 1) 11 7 1 224 FSET 1
1.000000E+00 1.000091E+00 9.996373E-01 1.001919E+00 9.909107E-01
1.346332E-13-9.909107E-01-1.001919E+00-9.996373E-01-1.000091E+00
-1.000000E+00
```

```
$ 3.33330E+03 RSGM( 1) 11 10 1 1 1.0000E+00
3.333300E+03 3.332737E+03 3.335561E+03 3.321716E+03 3.395802E+03
3.395802E+03 3.395802E+03 3.321716E+03 3.335561E+03 3.332737E+03
3.333300E+03
```

:
:

Iteration # 20

```
$ -1.12411E+00 U1 ( 1) 11 7 1 224 FSET 1
1.000000E+00 1.000000E+00 1.000000E+00 9.999994E-01 1.000003E+00
-8.464560E-14-1.000003E+00-9.999994E-01-1.000000E+00-1.000000E+00
-1.000000E+00
```

```
$ 3.33330E+03 RSGM( 1) 11 10 1 1 1.0000E+00
3.333300E+03 3.333300E+03 3.333301E+03 3.333297E+03 3.333315E+03
3.333315E+03 3.333315E+03 3.333297E+03 3.333301E+03 3.333300E+03
3.333300E+03
```

Iteration # 22

```
$ -1.12411E+00 U1 ( 1) 11 7 1 224 FSET 1
1.000000E+00 1.000000E+00 1.000000E+00 9.999994E-01 1.000003E+00
2.603916E-14-1.000003E+00-9.999994E-01-1.000000E+00-1.000000E+00
-1.000000E+00
```

```
$ 3.33330E+03 RSGM( 1) 11 10 1 1 1.0000E+00
3.333300E+03 3.333300E+03 3.333300E+03 3.333298E+03 3.333312E+03
3.333312E+03 3.333312E+03 3.333298E+03 3.333300E+03 3.333300E+03
3.333300E+03
```

Table 8.12 Solution output for Burgers equation (continued)

Iteration # 25

```
$ -1.12411E+00 U1 ( 1) 11 7 1 224 FSET 1
1.000000E+00 1.000000E+00 1.000000E+00 9.999993E-01 1.000003E+00
1.286104E-13-1.000003E+00-9.999993E-01-1.000000E+00-1.000000E+00
-1.000000E+00
$ 3.33330E+03 RSGM( 1) 11 10 1 1 1.0000E+00
3.333300E+03 3.333300E+03 3.333300E+03 3.333298E+03 3.333311E+03
3.333311E+03 3.333311E+03 3.333298E+03 3.333300E+03 3.333300E+03
3.333300E+03
```

f) Uniform 10 node mesh

```
$ 1.00000E+00 X1 ( 1) 10 10 1 493 0
0.000000E+00 1.111111E-01 2.222222E-01 3.333333E-01 4.444444E-01
5.555556E-01 6.666667E-01 7.777778E-01 8.888889E-01 1.000000E+00
```

Initial condition :

```
$ STEP 0, IT: 1, NITTOT 1, TH: 1.0, DT:0.000E+00, STA: 0.0
$ 0.00000E+00 U1 ( 1) 10 0 1 223 FSET 1
1.000000E+00 1.000000E+00 1.000000E+00 7.500000E-01 5.000000E-01
-5.000000E-01-7.500000E-01-1.000000E+00-1.000000E+00-1.000000E+00
```

g) Solution History Synopsis :

```
1 1 2 2 3 1 RSGM SOLVER = >GAUSS-ELIM< INT = 1
2 2 2 1 3 1 U1 SOLVER = >GAUSS-ELIM< INT = 1
```

Table 8.12 Solution output for Burgers equation (continued)

```

1  1 MAX DQ    0.00000E+00  1.08036E+00
1  2 MAX DQ    0.00000E+00  6.96648E-01
      :
      :
      :
1 21 MAX DQ    0.00000E+00 -1.26236E-07
1 22 MAX DQ    0.00000E+00 -5.73794E-08

```

h) Final Solution :

```

$ STEP  1, IT:  22, NITTOT  22, TH: 1.0, DT:1.000E+00, STA: 1.0
$  1.08036E+00 U1 (    1)   10    5    1   223      FSET 1
  1.000000E+00 9.999999E-01 1.000001E+00 9.999999E-01 1.000003E+00
-1.000003E+00-9.999999E-01-1.000001E+00-9.999999E-01-1.000000E+00

$  1.85183E+03 RSGM(    1)   10    8    1    1  1.0000E+00 0
  1.851833E+03 1.851833E+03 1.851835E+03 1.851833E+03 1.851839E+03
  1.851839E+03 1.851833E+03 1.851835E+03 1.851833E+03 1.851833E+03

$   5.73794E-08 Solution U Residuals
  0.000000E+00 1.147619E-08-1.147602E-08 3.442813E-08-5.737938E-08
  5.737938E-08-3.442813E-08 1.147602E-08-1.147619E-08 0.000000E+00

```

i) Non-Uniform 10 node mesh :

```

$  1.00000E+00 X1 (    1)   10   10    1  493      0
  0.000000E+00 2.742857E-01 4.114286E-01 4.800000E-01 4.933333E-01
  5.066667E-01 5.200000E-01 5.885714E-01 7.257143E-01 1.000000E+00

```

Table 8.12 Solution output for Burgers equation (continued)

j) Initial condition :

```

$ STEP  0, IT:  1, NITTOT  1, TH: 1.0, DT:0.000E+00, STA: 0.0
$  0.00000E+00 U1 (  1)  10  0  1  223  FSET 1
  1.000000E+00 1.000000E+00 1.000000E+00 7.500000E-01 5.000000E-01
-5.000000E-01-7.500000E-01-1.000000E+00-1.000000E+00-1.000000E+00

```

k) Solution History Synopsis :

```

  1  1  2  2  3  1 RSGM SOLVER = >GAUSS-ELIM< INT = 1
  2  2  2  1  3  1 U1  SOLVER = >GAUSS-ELIM< INT = 1

  1  1 MAX DQ  0.00000E+00  1.13268E+00
  1  2 MAX DQ  0.00000E+00 -7.80190E-01
      :
      :
      :
  1 24 MAX DQ  0.00000E+00 -1.87405E-05
  1 25 MAX DQ  0.00000E+00 -1.17711E-05

```

l) Final Solution :

```

$ STEP  1, IT:  25, NITTOT  25, TH: 1.0, DT:1.000E+00
$  1.13268E+00 U  (  1)  10  5  1  223  FSET 1
  1.000000E+00 9.999995E-01 1.000001E+00 9.999984E-01 1.000008E+00
-1.000008E+00-9.999984E-01-1.000001E+00-9.999995E-01-1.000000E+00

$  4.57002E+03 RSGM Distribution
  4.571383E+03 3.428538E+03 1.714268E+03 6.825337E+02 2.222188E+02
  2.222188E+02 6.825337E+02 1.714268E+03 3.428538E+03 4.571383E+03

$  1.17711E-05 Solution U Residual
  0.000000E+00-7.826924E-07 8.348584E-07-2.584425E-06 1.177110E-05
-1.177110E-05 2.584425E-06-8.348584E-07 7.826924E-07 0.000000E+00

```

8.2.9 Unsteady, $d = 1$ smooth and non-smooth wave convection, $C \leq 1$

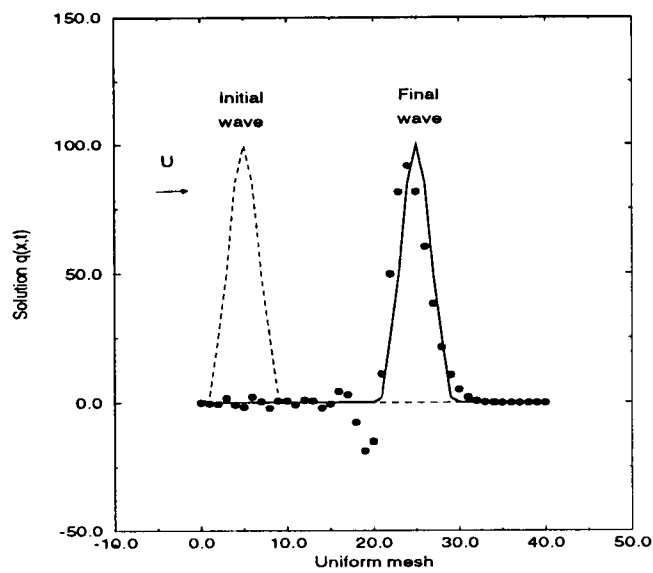
Translation of an isolated energy packet is a standard $d = 1$ verification problem that has been extensively studied, c.f. Baker (1983, Ch. 4.7) using present data for several FE and a finite difference (FD) Crank-Nicolson algorithm. Hirsch (1992) presents data for several FD algorithm, recall Figure 1.1. The one dimensional form of equation (1.1) is

$$\mathcal{L}(q) = \frac{\partial q}{\partial t} + U \frac{\partial q}{\partial x} = 0, \quad \text{on } \Omega \times t \subset \mathbb{R}^1 \times \mathbb{R}^+ \quad (8.37)$$

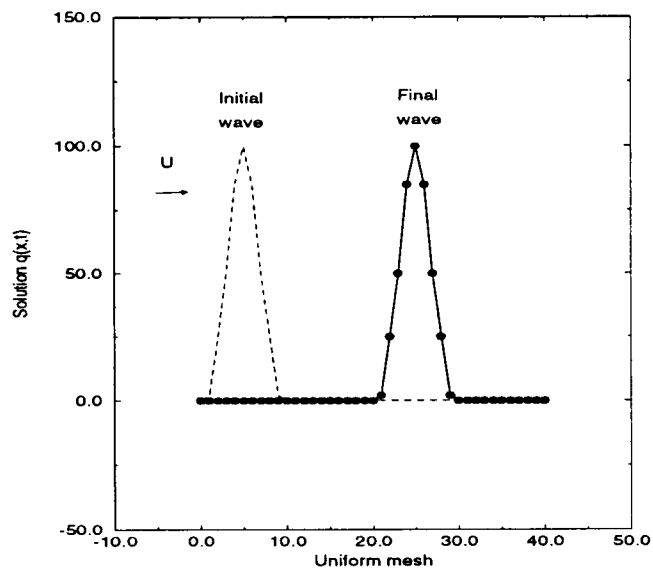
where U is a constant. The analytical solution $q(x, t)$ is preservation of the initial condition $q(x, t = t_0)$ as it is convected parallel to x .

The initial condition, the analytical final solution GWS and GMP solution data for a Gaussian distribution convection are compared for Courant number $C = 1$ after 20 timesteps in Figure 8.20. The initial wave is interpolated across 10 nodes, thus the analytical waveform has been translated 2 wavelengths. The GWS solution is non-dissipative and dispersive, as theoretically predicted, with trailing oscillatory maxima about 20% and with peak level reduced (via phase distortion) by almost 10%, Figure 8.20a. Also, the GWS peak lags the analytical solution by about 1 node, due to phase lag. The comparative GMP algorithm (6.20) solution is dispersion free and nodally exact to within round-off error (order 10^{-16}), Figure 8.20b.

For propagation of a full sine wave, 20 timesteps at $C = 1$ again translates q_0 by two wavelengths. The leading (negative) extremum of the GWS solution decays by nearly 40% and trailing dispersion error dominates the entire solution, Figure 8.21a. The companion GMP algorithm solution is totally devoid of $2\Delta x$ error oscillations

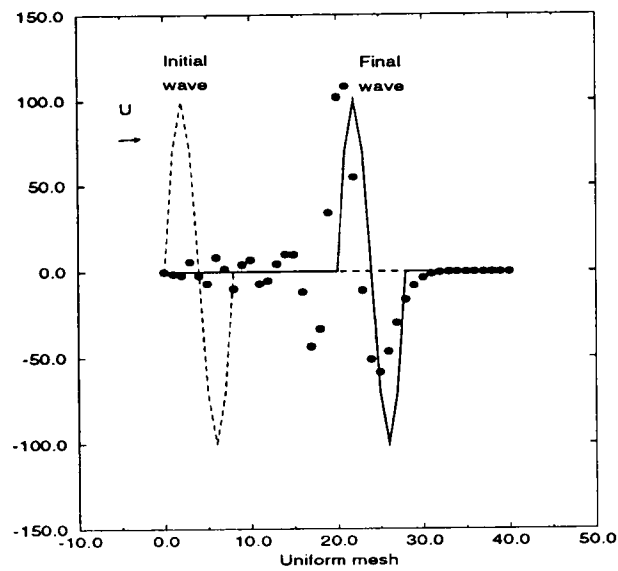


a) GWS nodal solution

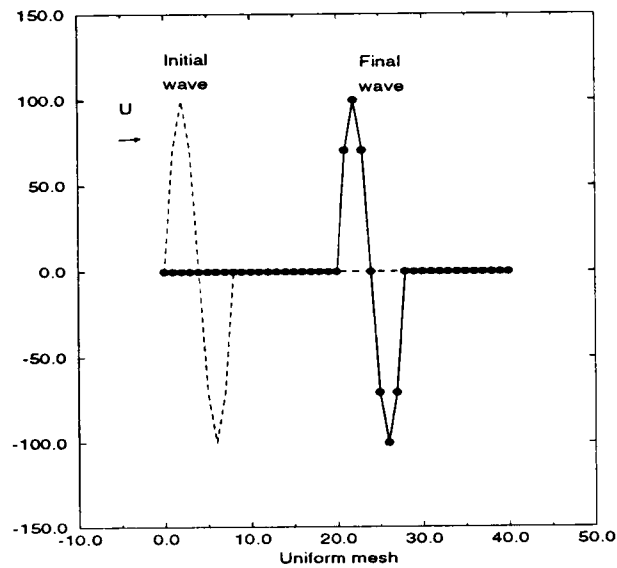


b) GMP nodal solution

Figure 8.20 Gaussian wave propagation, two wavelengths at $C = 1$.



a) GWS solution



b) GMP solution

Figure 8.21 Full sine wave propagation, two wavelengths at $C = 1$.

and the sine wave pattern remains nodally exact (to round-off), Figure 8.21b. Via (8.38)-(6.23), the GMP element matrices

$$[M]_e = \Delta x_e ([A200L] - [APERT]) = \frac{\Delta x_e}{4} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad (8.38)$$

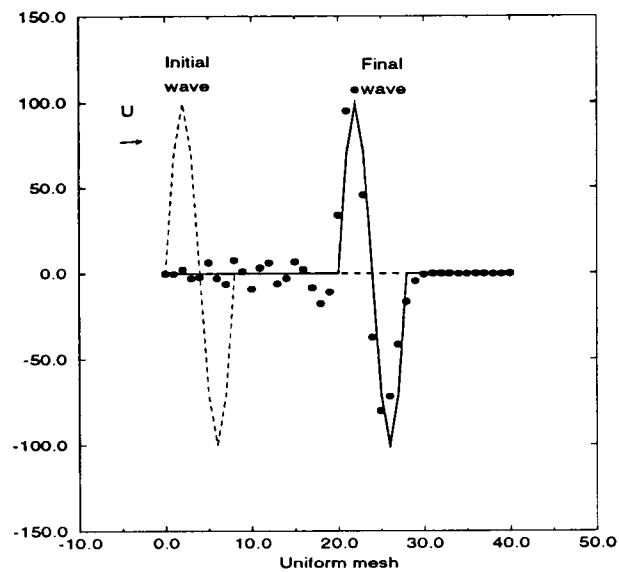
$$[U]_e = U ([A201L] - [APERX]) = \frac{U}{2} \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \quad (8.39)$$

were theoretically established to compute the $C = 1$ Gaussian and sine wave convection solution, Figures 8.20b-8.21b.

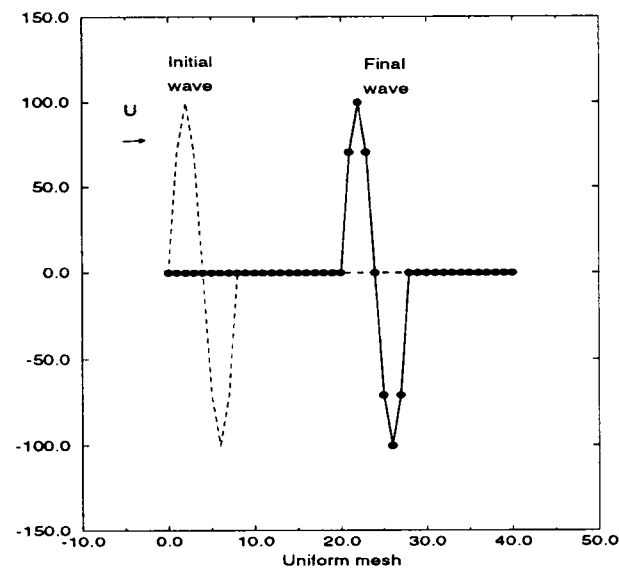
Figure 8.22 documents the companion $C = 0.5$ GWS and GMP solutions after 40 timesteps (two wavelengths displacement). The lower Courant number slightly improves the GWS solution, but it is still unacceptably polluted by dispersion error, Figure 8.22a. Conversely, for $\beta = \gamma = 1$ and $t_1 = t_4 = 1/8$, $t_2 = t_3 = -1/8$, the GMP algorithm solution remains nodally exact, Figure 8.22b, albeit to a lower order in round-off (order 10^{-6}). Solution output details are given in Appendix I.

CFD algorithms applied to convection of a (non-smooth) square wave form is detailed by Hirsch (1992, Ch. 21.3), recall Figure 1.1. For an initially 20-node wide square wave, the GWS and GMP solutions for $C = 1$ for propagation over 3 wavelengths are compared in Figure 8.23. The linear basis GWS solution dispersion error totally distorts the initial wave geometry. In distinction, the GMP algorithm solution, using (8.38), is nodewise exact to within round-off error ($\sim 10^{-16}$).

Solutions obtained for $C = 0.5$ after 120 timesteps are compared in Figure 8.24. In qualitative agreement with Figure 1.1a, the linear basis GWS solution, Figure 8.24a,

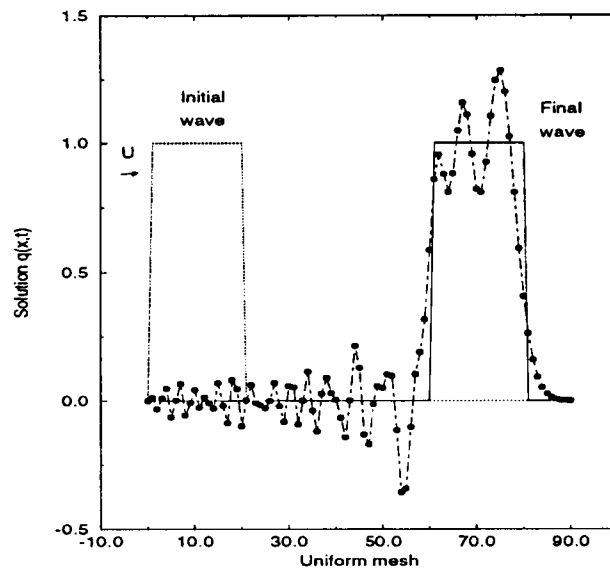


a) GWS solution

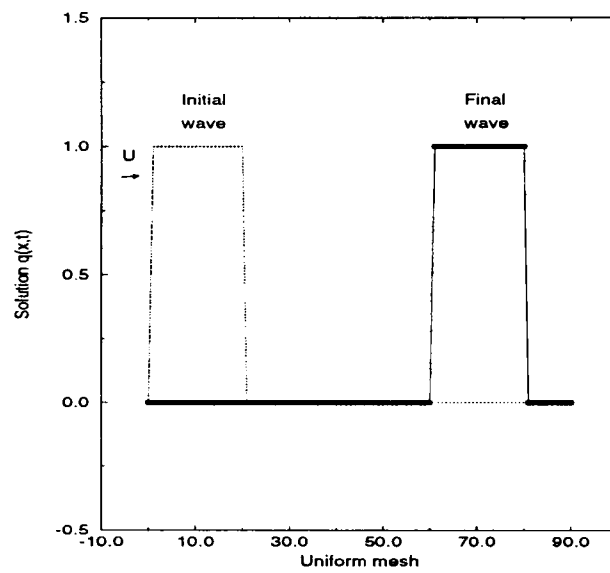


b) GMP solution

Figure 8.22 Full sine wave propagation, two wavelengths at $C = 0.5$.

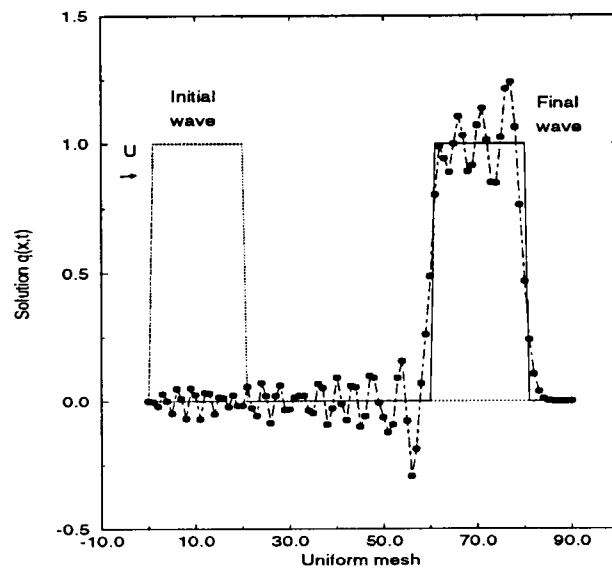


a) GWS solution

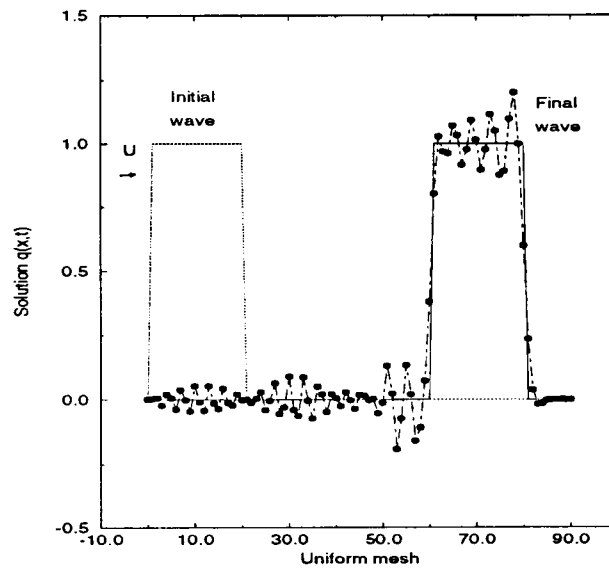


b) GMP solution

Figure 8.23 20-node wide square wave propagation, three wavelengths at $C = 1$.

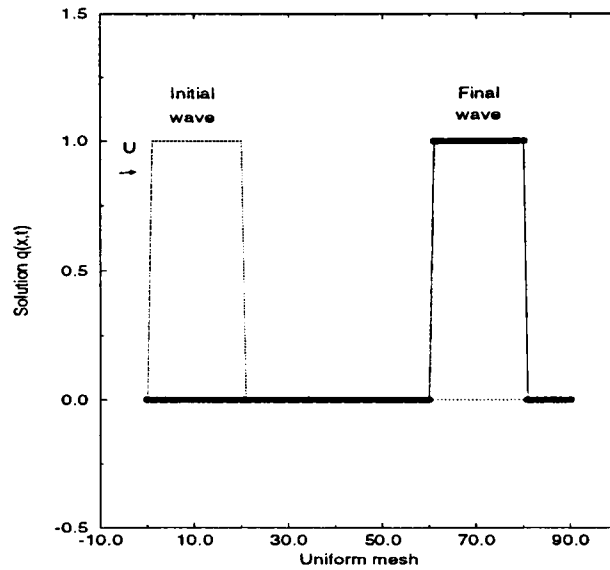


a) GWS solution



b) GMP solution ($\beta = \gamma = 0$)

Figure 8.24 20-node wide square wave propagation, three wavelengths at $C = 0.5$.

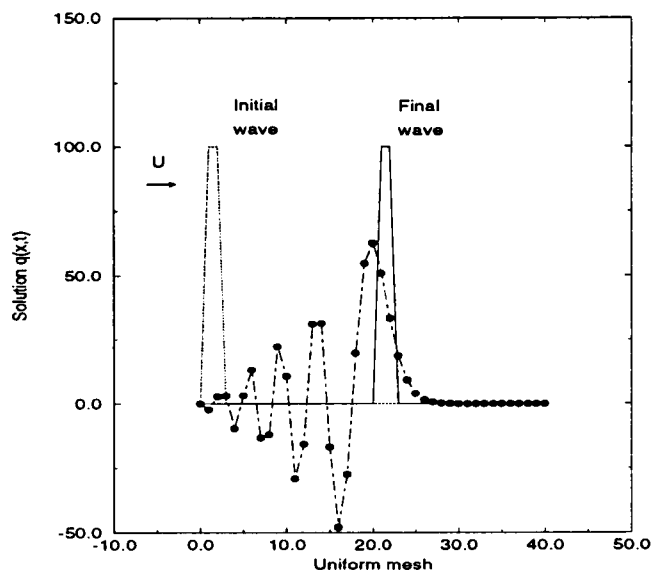


c) GMP solution ($\beta = \gamma = 2$)

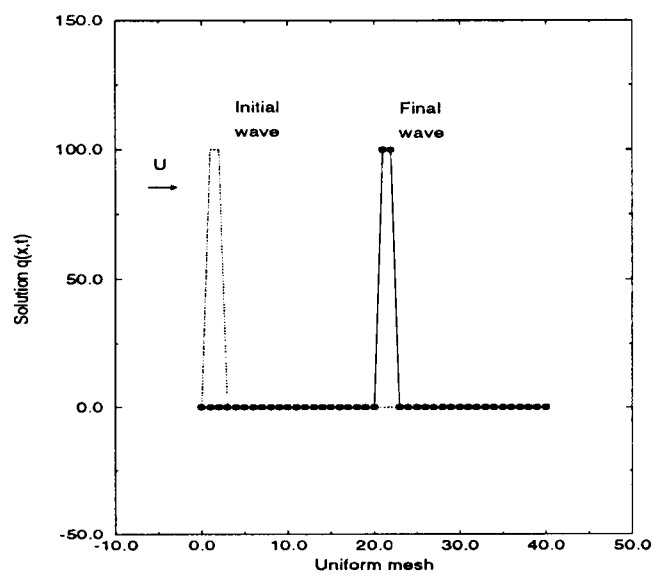
Figure 8.24 20-node wide square wave propagation, three wavelengths at $C = 0.5$
(concluded).

is very dispersive with an error extremum of 22%. For the data selection in (6.16), the GMP algorithm is only fourth order accurate, hence it produces the dispersive solution shown in Figure 8.24b. However, using $\beta = \gamma = 2$, which yields tenth order accuracy, the GMP solution is again nodally exact to a lower order ($\sim 10^{-8}$) in round-off, Figure 8.24c. Unlike Figure 1.1b, this excellent non-unit Courant number GMP solution does not suffer from any artificial diffusion.

The most challenging definition for this problem is the minimum ($2\Delta x$ wide) initial wavelength square wave. Figure 8.25 summarizes comparative solutions for this square wave propagation over 10 initial wavelengths at $C = 1$ and $C = 0.5$. Clearly, the linear basis GWS solution is totally distorted by dispersion error, with



a) GWS solution



b) GMP solution

Figure 8.25 2-node wide square wave propagation, ten wavelengths at $C = 1.0$

trailing wake and peak loss error of nearly 50%, Figure 8.25a. As predicted by the Fourier modal analysis (8.39), the corresponding eighteenth-order accurate GMP solution remains nodally exact (to order 10^{-16}) and dispersion-free, Figure 8.25b. The companion $C = 0.5$ solutions attest to GMP algorithm relative robustness, Figure 8.26a-b.

8.2.10 Two-dimensional convection via a solid body rotation

Originally developed for atmospheric dispersion prediction CFD assessments, this is an extremely informative non-diffusive 2-d verification test case for evaluating any algorithm. A finite difference (FD), a finite volume (FV), and several finite element (FE) algorithm tests are reported by Baker, *et. al* (1993), and a FE adaptive h - p mesh refinement strategy result is published by Demkowicz, *et. al* (1989). The solution to this unsteady, linear pure convection verification problem satisfies

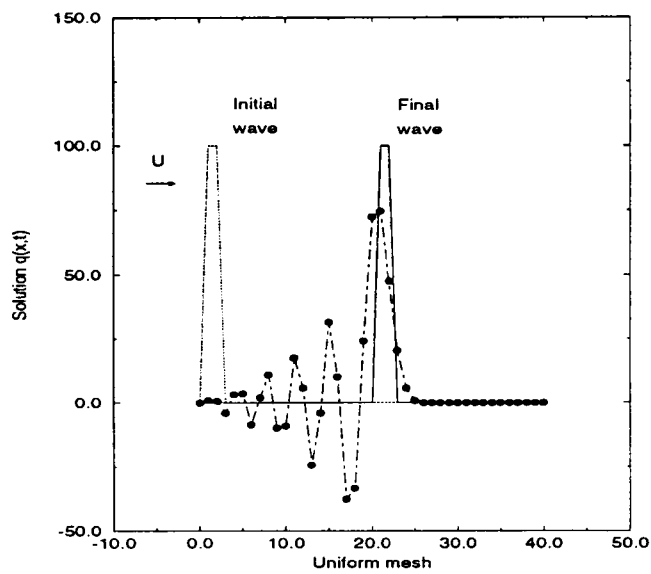
$$\begin{aligned} \frac{\partial q}{\partial t} + \mathbf{u} \cdot \nabla q &= 0 & \text{in } \Omega \\ q &= 0 & \text{on } \partial\Omega \end{aligned} \quad (8.40)$$

where the imposed solid body rotational velocity field is $\mathbf{u} = \pm r\dot{\theta} \hat{e}_\theta$. The “rotating cone” initial condition is an elevated cosine-hill, see Figure 8.27a, with specification

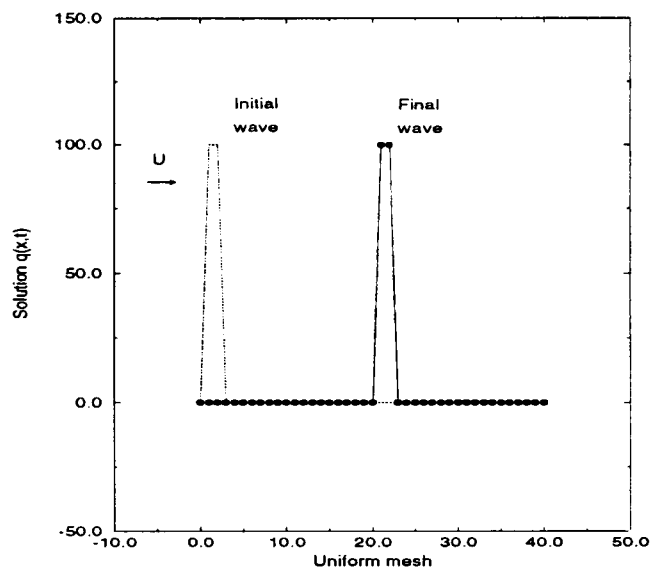
$$q_0 = \begin{cases} \frac{1}{2}(1 + \cos 4\rho\pi) & \text{if } \rho \leq 0.125 \\ 0 & \text{otherwise} \end{cases} \quad (8.41)$$

where $1 \leq j \leq 2$ and

$$\rho^2 = (x_j - x_j^c)^2 \quad 0 \leq x_j \leq 1 \quad (8.42)$$

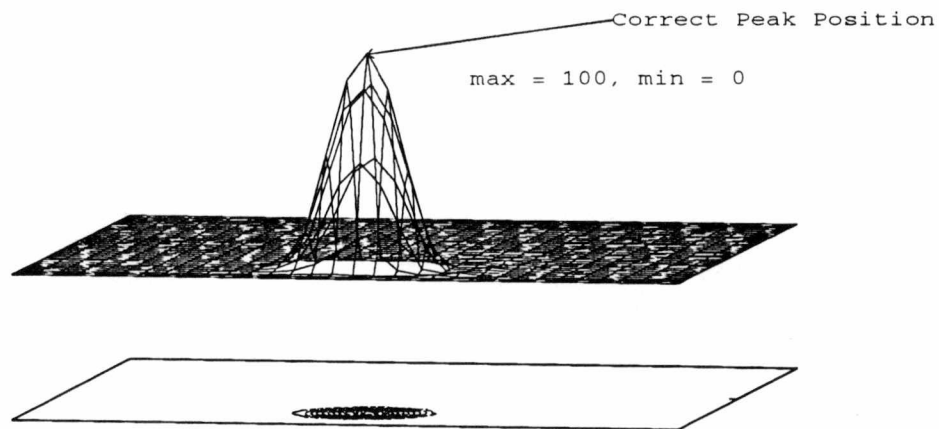


a) GWS solution

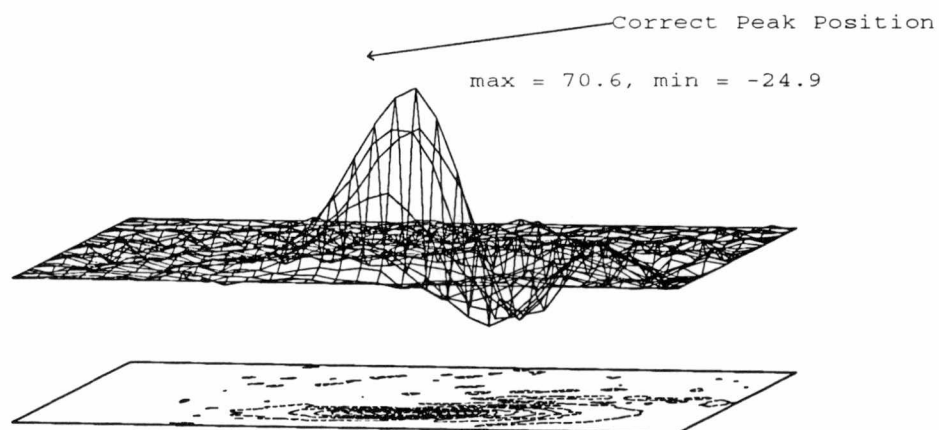


b) GMP solution ($\beta = \gamma = 2$)

Figure 8.26 2-node wide square wave propagation, ten wavelengths at $C = 0.5$



a) initial condition and exact final solution



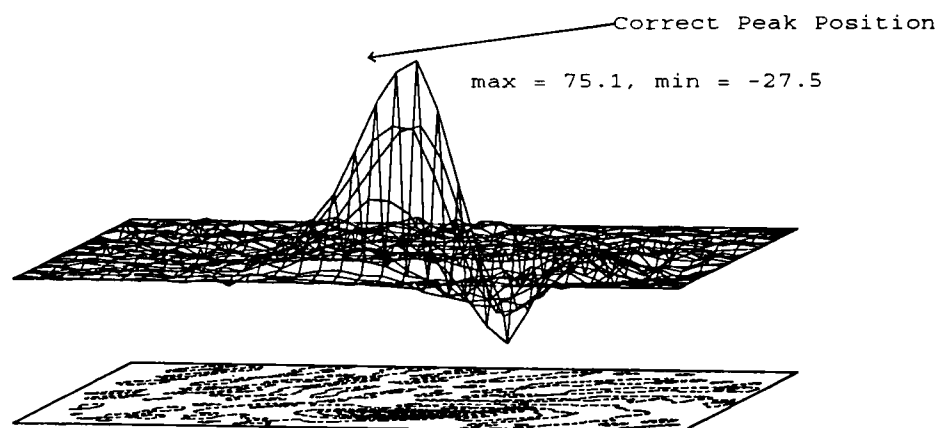
b) $k = 1$ FE GWS, $C = 1$

Figure 8.27 Rotating cone solutions, one revolution on a 33×33 mesh

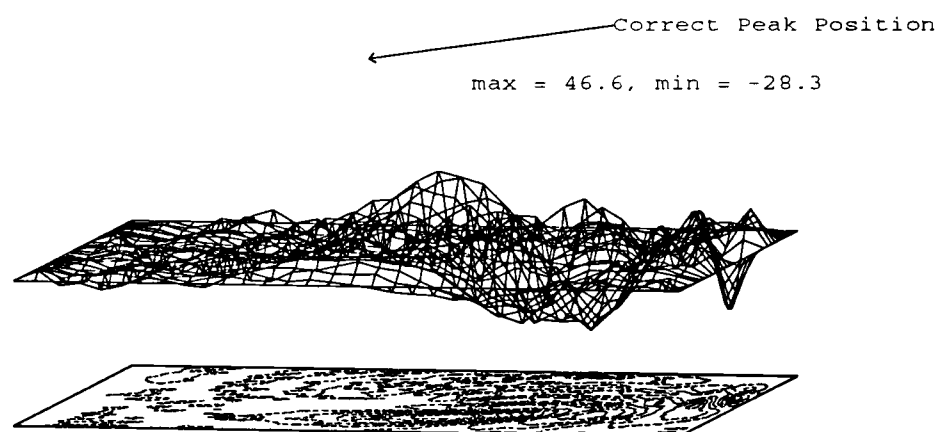
In (8.42), x_j^c denotes the mesh centroidal coordinate of the axi-symmetrical initial distribution. The analytical solution is again exact preservation of the initial condition, as the distribution is circularly convected around the z -plane.

The full-rotation solutions obtained on a uniform 33×33 node mesh for $k = 1$ and $k = 2$ FE GWS algorithms, using the (non-diffusive) trapezoidal time integration scheme and for $C = 1.0$ at the distribution centroid are shown in Figure 8.27b-c. Both GWS solutions suffer large dispersive error accumulation on this mesh at this large Courant number. The $k = 1$ FE solution has trailing wake error extremum exceeding 25%, Figure 8.27b, and the solution peak is reduced by nearly 30% and severely lags behind the correct location. Although an improvement in the peak level ($\sim 5\%$) accrues to the quadratic basis solution, Figure 8.27c, the dispersion error extremum is nominally worsened by 3%. Therefore, increasing basis degree k of an FE implementation is of marginal value for this problem.

For comparison, the Crank-Nicolson finite difference (CNFD) solution, Figure 8.27d, after one full revolution is poorer than either FE result, with solution peak level loss to 45% and a trailing wake error extremum of 28%. The “best appearing” solution in the literature is the h - p adaptive FE GWS mesh solution obtained after 400 timesteps for one full revolution, Demkowicz, *et. al* (1989), reproduced as Figure 8.27e. The surface perspective plot appears smooth, but nearly 2% of the peak level is lost even for this low Courant number ($C \sim 1/6$). No results are reported for unit Courant number, which severely aggravates short wavelength error oscillation creation throughout the entire solution, *c.f.*, Figure 8.27b-d.

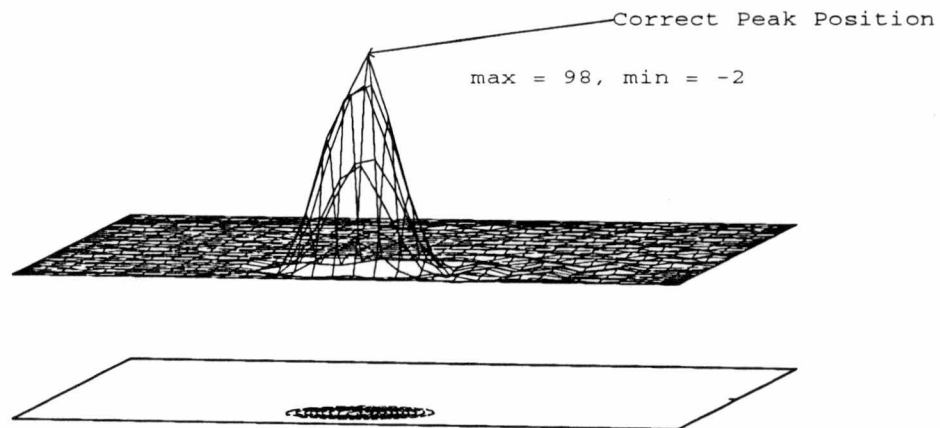


c) $k = 2$ FE GWS, $C = 1$

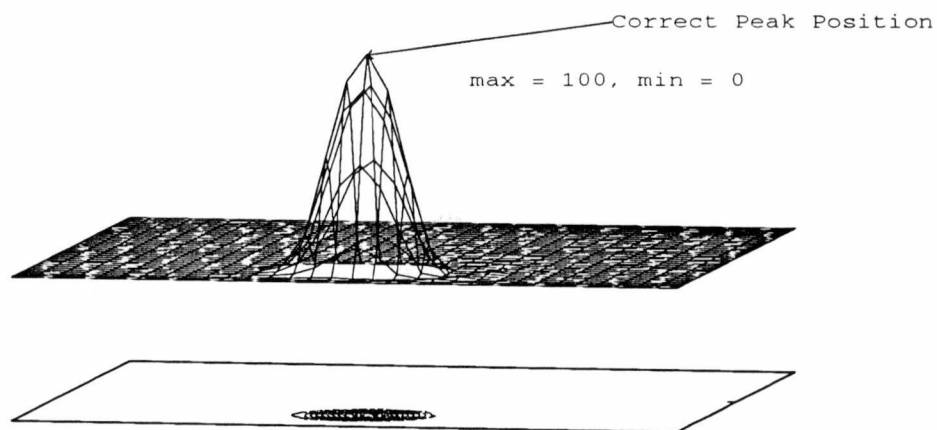


d) Crank-Nicolson FD, $C = 1$

Figure 8.27 Rotating cone solutions, one revolution on a 33×33 mesh (contd.)



e) h - p FE solution, $C \sim 1/6$, $\Delta t = \frac{2\pi}{400}$



f) GMP solution, $C_e = 1$, $\Delta t = \frac{2\pi}{32}$

Figure 8.27 Rotating cone solutions, one revolution on a 33×33 mesh (contd.)

In distinction to each of these solutions, using the time-splitting technique (6.54)-(6.62), the linear basis modified weak statement GMP solution, for $C_e = 1$ after one revolution, yields the essentially nodally exact solution with negligible dispersion error or phase lag (to round-off, order 10^{-4}), Figure 8.27f. The GMP method utilized only 32 timesteps to complete one full revolution on this mesh. Most importantly, the peak travelled the correct distance, and the solution is certainly the “best” among those available for comparison.

The mesh independence of the GMP weak statement formulation (6.51) for $C_e = 1$ is also verified on a 17×17 node uniform mesh using $\Delta t = 2\pi/16$. The solution after one full revolution, Figure 8.28, again confirms essentially exact preservation of the initial condition without dispersion error or phase lag to round-off (order 10^{-4}). Via (6.49), the GMP mesh measure becomes directly proportional to the solution timestep for a particular C_e in (6.51). Hence, for a large timestep, the GMP algorithm will produce a nodally accurate (to round-off) solution only on a very coarse mesh. Detailed time-accurate solutions on 17^2 and 33^2 node meshes are in Appendix I.

8.3 The reconstruction (RCN) algorithm solutions:

8.3.1 Reconstruction for piecewise non-smooth and smooth wave propagation, $d = 1$

The governing hyperbolic form of (2.1) for $u = 1$ and $\epsilon = 0$ is

$$\frac{\partial q}{\partial t} + \frac{\partial q}{\partial x} = 0 \quad (8.43)$$

For $q_0(x) = q(x, t_0)$ initial condition, the analytical solution is

$$q(x, t) = q_0 e^{i\omega(x - t)} \quad (8.44)$$

i.e., exact preservation of the initial distribution translated parallel to the x axis.

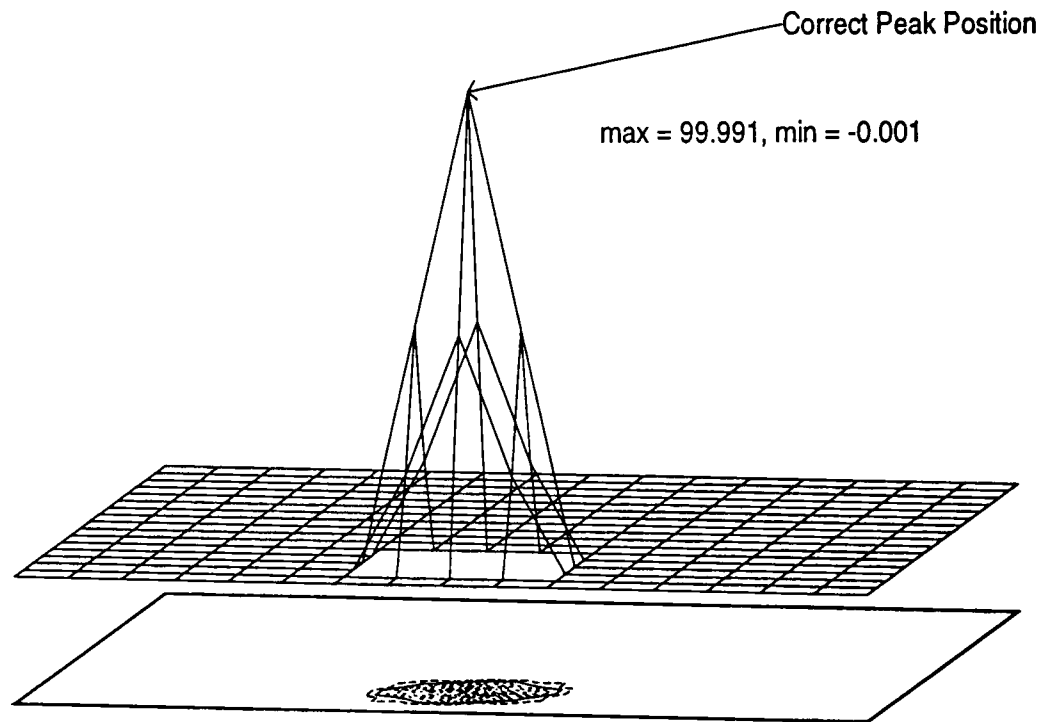


Figure 8.28 Rotating cone GMP solution on a 17^2 mesh, after one revolution at

$$C_e = 1, \Delta t = \frac{2\pi}{16}$$

A 9-node wide square wave, with $q_{j_0} = 100$ for $2 \leq j \leq 10$ and $q_{j_0} = 0$ elsewhere, is propagated to $n\Delta t = 90$ sec. For $C = 1$, this amounts to 90 timesteps, and the resultant FE solutions for linear ($k = 1$) and quadratic ($k = 2$) bases are compared to the exact solution in Figure 8.29a-b respectively. As observed in Section 8.2.9, dispersion error dominates both solutions, which become quite distorted images of the initial wave, recall Figure 1.1. The derived reconstruction (RCN) procedure can produce a completely dispersion-free restoration of the initial wave geometry for both solutions, Figure 8.29c. Note that the RCN algorithm does not affect the discrete FE CFD solution phase lag (1 node). With reference to Section 3, the RCN procedure for this linear problem is as follows.

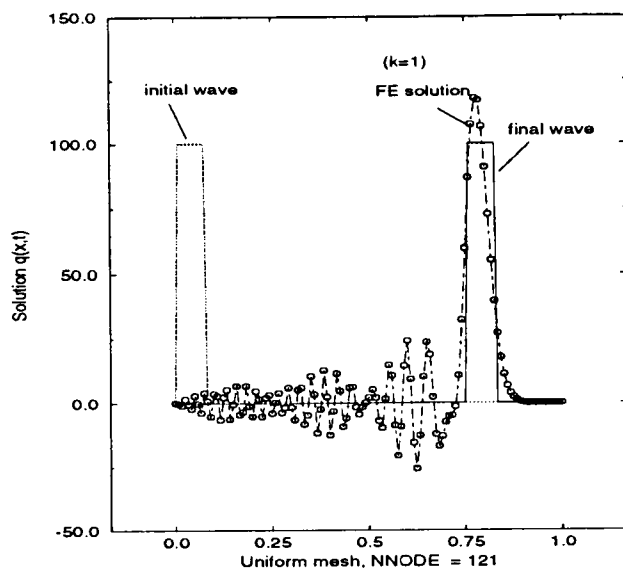
Step 1. l_1 norm of the initial condition is $S^0 = 900$, and $l_\infty = 100$. The non-zero distribution, $q_0 = 100$, is a set of 9 nodes symmetric about the central fifth node, which is the reference location.

Step 2. After 90 sec., the discrete solution l_∞ is located at node 94.

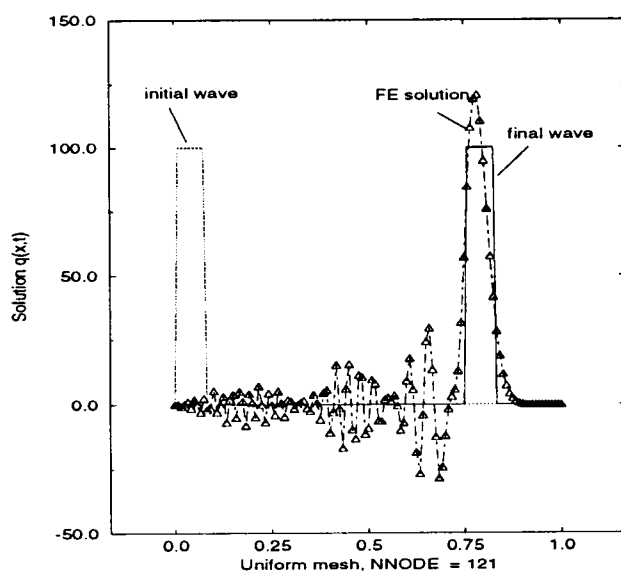
Step 3. Corresponding sum-norm $l_1 \equiv S^n = 903$, hence the reconstruction ratio $\gamma_1 = \frac{903}{900} \approx 1$.

Step 4. The reconstructed solution is the 9-node wide set of $\gamma_1 q_0 = 100$ distribution, symmetric about the central reference point located at node 94.

The RCN procedure is equally applicable to initial conditions with non-positive data, *e.g.*, a complete sine wave. As shown in Figure 8.30a, after propagating two full wavelengths, the $k = 1$ FE solution l_∞ norm is located at node 22 and the leading discrete solution negative extremum is lost by 50% while the trailing edge dispersion

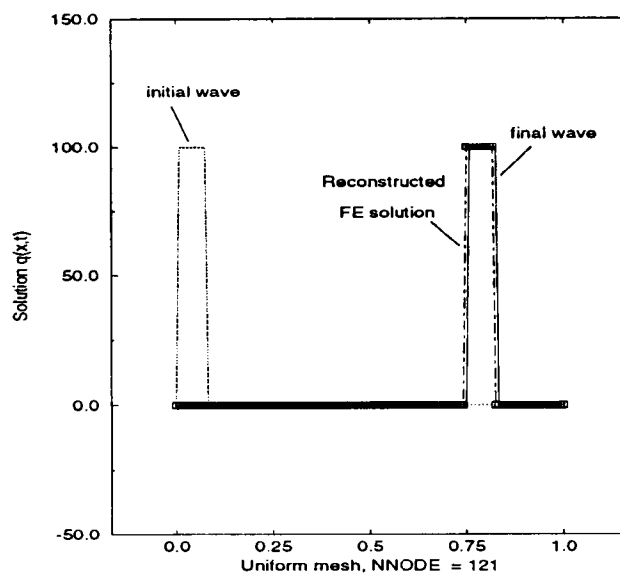


a) $k = 1$, linear basis



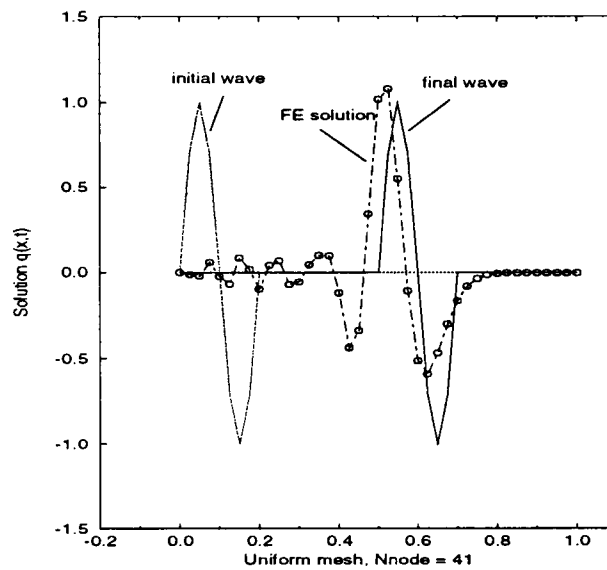
b) $k = 2$, quadratic basis

Figure 8.29 1d 9-node wide square wave propagation, $C = 1.0$, GWS solutions, $t=90$ sec.



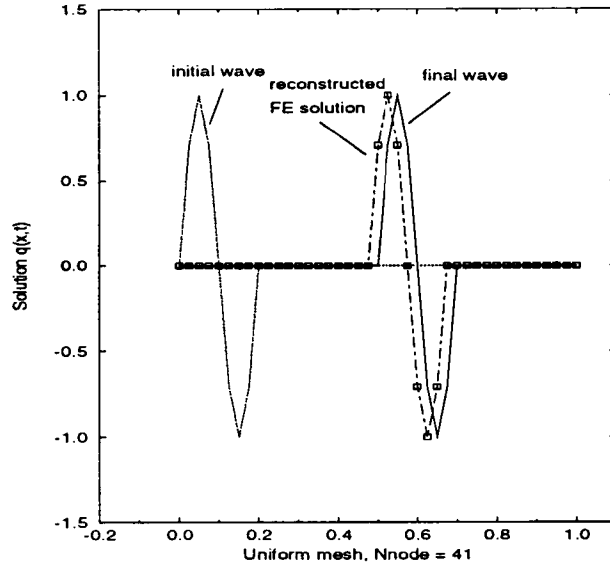
c) FE solution, $k = 1$ or 2, reconstructed

Figure 8.29 1d 9-node wide square wave propagation, $C = 1.0$, solutions (contd.).



a) $k = 1$, linear basis

Figure 8.30 1d Sine wave propagation for $C = 1.0$, FE solutions at $t=20$ sec.



b) FE solution, $k = 1$, reconstructed

Figure 8.30 1d Sine wave propagation for $C = 1.0$, solutions (continued).

error dominates the remainder of the solution. The resultant RCN solution is devoid of all dispersive error oscillations, Figure 8.30b, and the sine wave pattern is fully restored, but with the phase lag inherited from the original CFD solution.

8.3.2 Steady advection-diffusion problem, $d = 1$

The governing steady parabolic differential form of (1.1) is

$$\frac{d}{dx} \left(uq - \epsilon \frac{dq}{dx} \right) = 0 \quad \text{on } 0 \leq x \leq \ell \quad (8.45)$$

which upon non-dimensionalization and for constant u becomes

$$\frac{d}{dx} \left(q - \frac{1}{\text{Re}} \frac{dq}{dx} \right) = 0 \quad \text{on } 0 \leq x \leq 1 \quad (8.46)$$

where $\text{Re} = u\ell/\epsilon$ is the familiar “Reynolds number”. Solutions to (8.45) are parameterized by Re , and for the Dirichlet boundary conditions $q(0) = 0$ and $q(1) = 1$, the

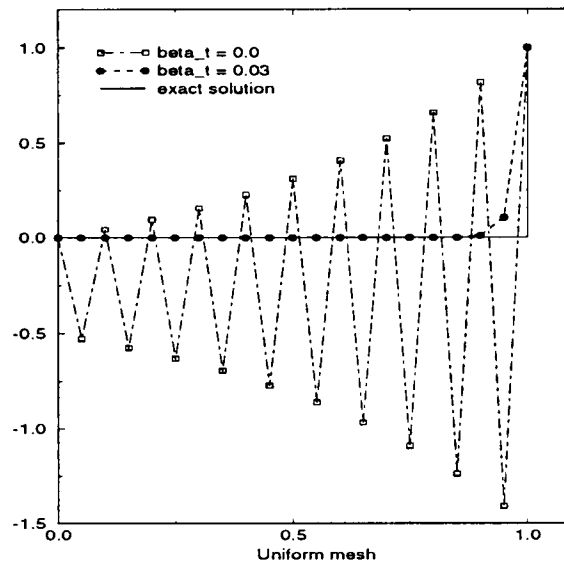
analytical solution to (8.46) is

$$q(x) = \frac{e^{x\text{Re}} - 1}{e^{\text{Re}} - 1} \quad (8.47)$$

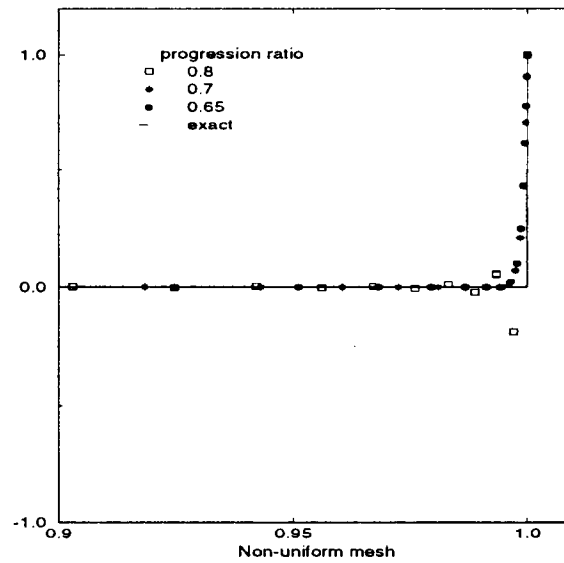
which is smooth but exhibits a sharp wall-layer character as Re becomes large, *i.e.*, $\epsilon \rightarrow 0$ in (8.45).

In practical situations, Re can range to 10^5 and larger, yielding a wall-layer point near the right boundary. For large Re , the dominant error mechanism is dispersive, as clearly illustrated for standard $k = 1$ FE algorithm on a uniform, Figure 8.31a, and a nonuniform, Figure 8.31b, $\text{Nnode}=21$ mesh. The classical theory, predicts that increasing Nnode sufficiently, and/or $k \geq 1$ in the standard FE methodology, steadily improves solution accuracy; however, these solutions remain non-monotone. In particular, at $\text{Re} = 1000$, a uniform mesh containing 100 nodes cannot yield a monotone solution using either the $k = 1$ or $k = 2$ FEM standard algorithms.

Using artificial dissipation via the TWS, *e.g.*, $\beta_t > 0$ in (3.6), smoothening of the dispersion error mode is possible, but at the expense of artificial diffusion. Figure 8.31a also illustrates one such CFD result, for $\beta_t = 0.03$ in (3.6), on a uniform mesh. However, as shown in Figure 8.31c, the RCN algorithm can produce an accurate monotone solution from non-artificially diffused FE solution both uniform and non-uniform mesh. Conversely, applying RCN to the TWS solution reproduces the inaccurate, artificially-diffused solution.

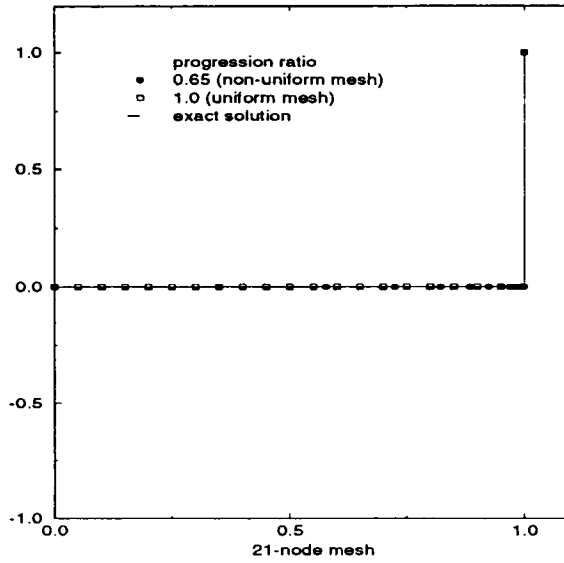


a) FE $k = 1$ uniform mesh



b) FE $k = 1$ non-uniform mesh

Figure 8.31 Advection-diffusion ($Re = 1000$) steady state solutions on a 21-node mesh.



c) FE solution, $k = 1$, reconstructed

Figure 8.31 Advection-diffusion ($Re = 1000$) steady state solutions (contd.).

8.3.3 Solid body rotation, unsteady convection in $d = 2$ dimensions

Originally developed for atmospheric dispersion CFD assessment, this extremely challenging 2-d verification is the solution to the unsteady, $d = 2$ linear pure convection form of (2.1) with $\epsilon = 0 = s$

$$\frac{\partial q}{\partial t} + \mathbf{u} \cdot \nabla q = 0 \quad \text{in } \Omega \quad (8.48)$$

$$q = 0 \quad \text{on } \partial\Omega \text{ (inlet only)}$$

For the “rotating cone,” the imposed solid body rotational velocity field is $\mathbf{u} = r\dot{\theta} \hat{e}_\theta$, and the initial condition is the cosine-hill,

$$q_0 = \begin{cases} \frac{1}{2}(1 + \cos 4\rho\pi) & \text{if } \rho \leq 0.125 \\ 0 & \text{otherwise} \end{cases} \quad (8.49)$$

see Figure 8.32, where $1 \leq j \leq 2$ and

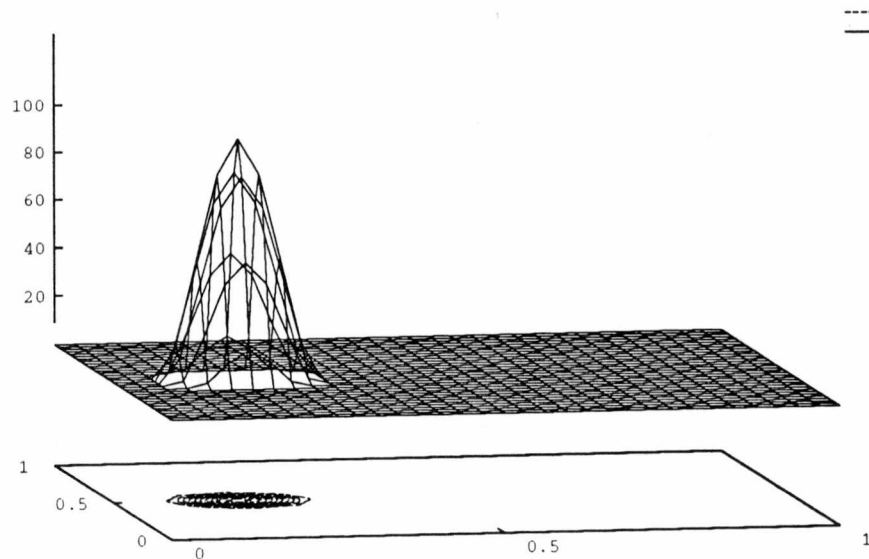
$$\rho^2 = (x_j - x_j^c)^2 \quad 0 \leq x_j \leq 1 \quad (8.50)$$

where x_j^c denotes the centroidal coordinate of the axi-symmetrical initial distribution. The analytical solution is again exact preservation of the initial condition, as it is convected around the z -plane, in the form

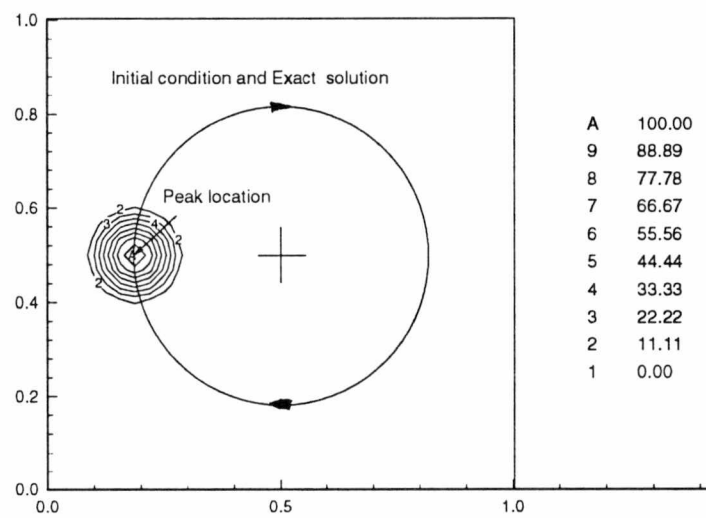
$$q(\mathbf{x}, t) = q_0 e^{i\omega(\mathbf{x} - \mathbf{u}t)} \quad (8.51)$$

For this hyperbolic problem, increasing basis degree k is of marginal value, while decreasing the non-dimensional time-step (Courant number) C serves to lower solution dispersion error levels for all k . Detailed discussions on comparative issues were discussed previously in this Chapter. Figures 8.33-8.34 graph perspective and contour plots of CFD solutions for the linear basis ($k = 1$) FE and Crank Nicolson finite difference (CNFD) algorithms (a,b) respectively, on a 33×33 node uniform mesh for $C = 1.0$ and trapezoidal integration scheme. Figure 8.33a presents the perspective view of the $k = 1$ FE solution and the contour plot clearly shows the large dispersive error accumulation on this mesh, Figure 8.33b. The $k = 1$ solution maximum is decreased to 74%, and the extremum wake dispersion is -33%. Companion presentations of the resultant reconstructed solution, Figure 8.33c and 8.33d, as obtained via the linear RCN algorithm, confirms attainment of a dispersion free solution with phase lag (~ 18 degrees) inherent to the base $k = 1$ FE solution.

These data for the Crank-Nicolson finite difference (CNFD) solution are shown in Figure 8.34. The CNFD scheme, after one full revolution at $C = 1.0$, creates an

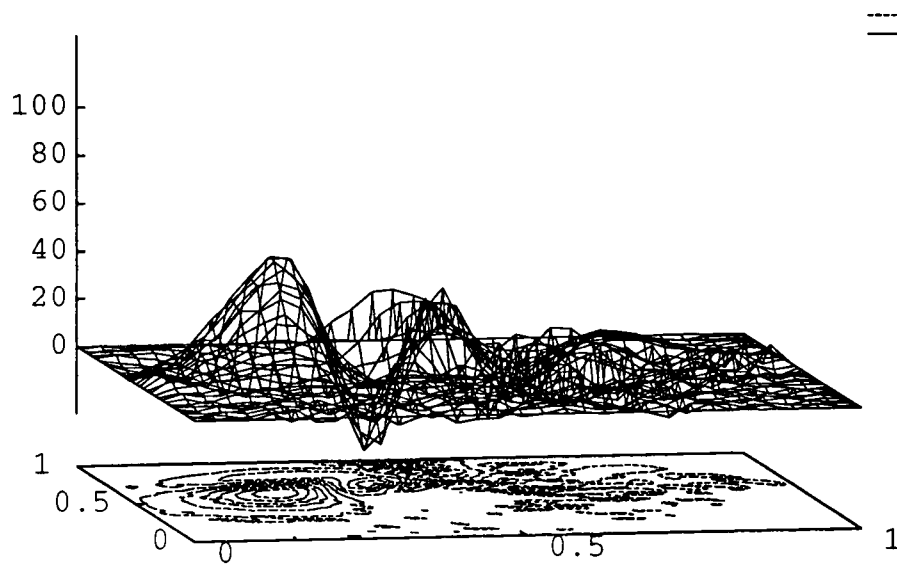


a) Perspective view

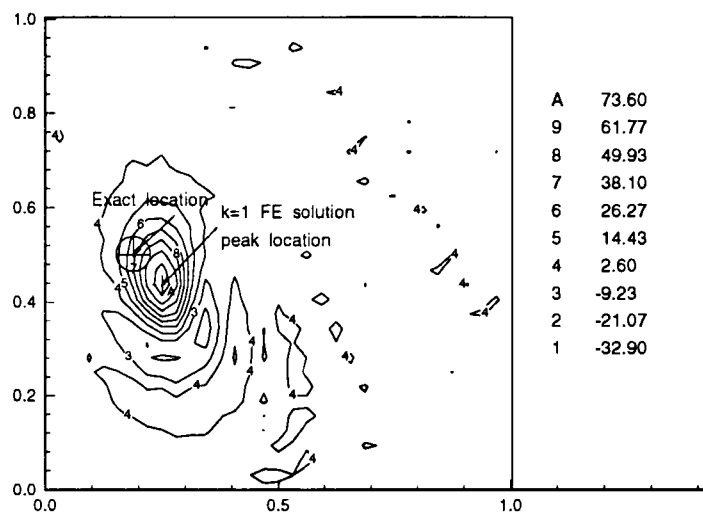


b) Contour plot

Figure 8.32 Rotating cone, initial condition and exact solution after one revolution.

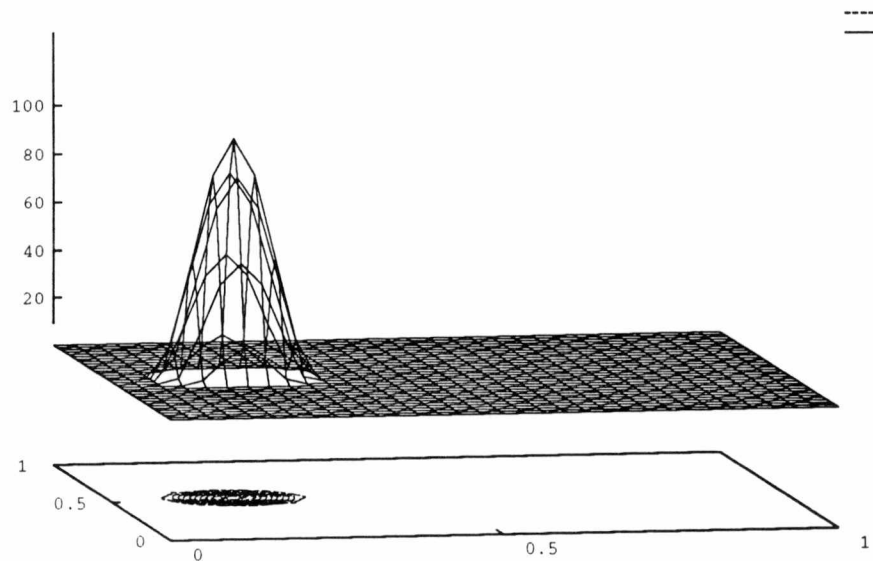


a) $k = 1$ FE solution perspective plot

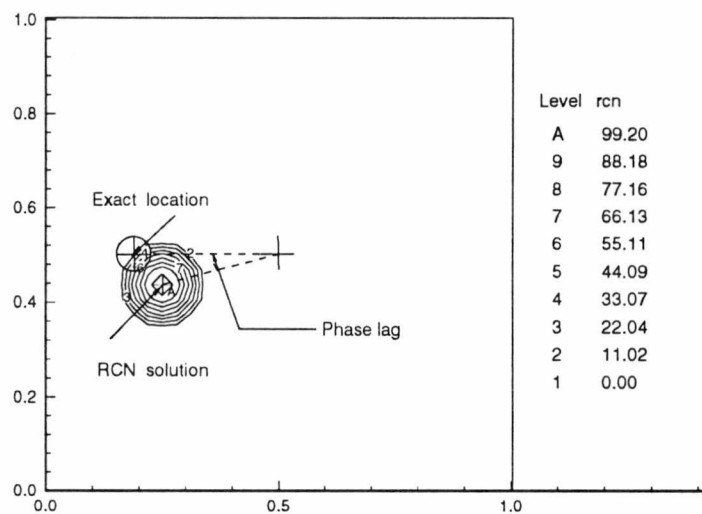


b) Contour plot for data in a)

Figure 8.33 Rotating cone verification CFD solution, $k = 1$ FE, $C = 1$, on 33^2 uniform mesh.



c) Reconstructed solution perspective plot



d) Contour plot for data in c)

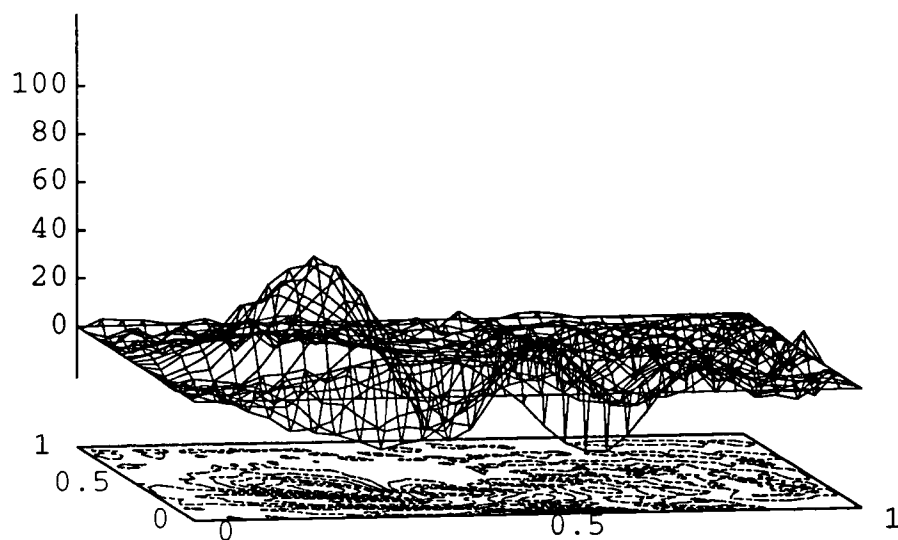
Figure 8.33 Rotating cone verification CFD solution (concluded)

even more distorted image of the cosine hill distribution, Figures 8.34a and 8.34b, with peak level decreased to 47% and the trailing wake extremum at -28% . The companion RCN solution for $\gamma = 0.98$ yields a dispersion free reconstruction of the initial condition centered where the FD solution l_∞ -norm is located, Figure 8.34c-d. The phase lag inherent to the CNFD discrete solution is unaltered, and is twice that (~ 35 degrees) of the $k = 1$ FE solution.

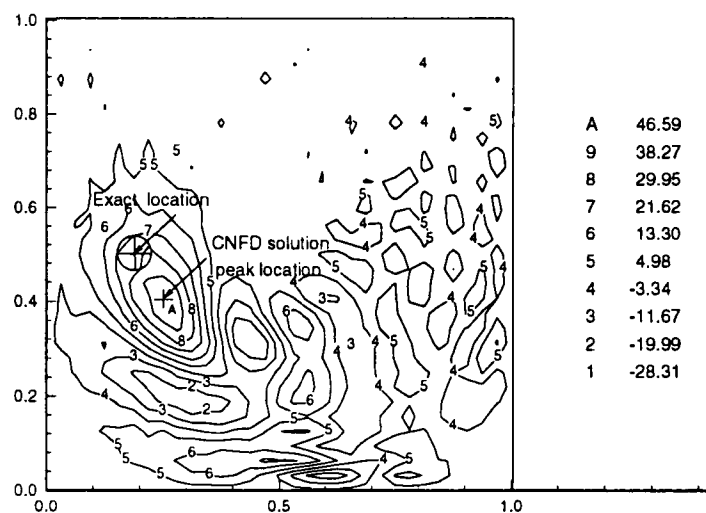
8.4 Navier-Stokes Problems

8.4.1 Shear-Driven Cavity Flow

The laminar incompressible flow in a square cavity whose top lid moves with a constant velocity in its own plane has served as a model problem in testing and validation of numerous numerical techniques, despite the boundary condition singularities at its two corners. For moderately high values of the Reynolds number Re , published results are available for this flow problem from a number of sources, *c.f.*, Ghia, *et. al* (1982), Rubin and Khosla (1977), Smith and Kidd (1975), Jiang (1990). These results, however, are obtained using moderate to very dense meshes. For example, Jiang used a 51×51 mesh using Least Square Finite Element Methods (LSFEM) while Ghia, *et. al*, used a 129×129 mesh as well as a 257×257 mesh using multigrid approach for $Re=400$ onwards. Ghia, *et. al*, mentioned that for “ $Re=100$, all results agree well with one another as well as with the present investigators, indicating that for this value of Re , the coarser grids employed by the previous investigators were quite adequate. As Re increases, however, the inadequacy of coarse meshes gradually

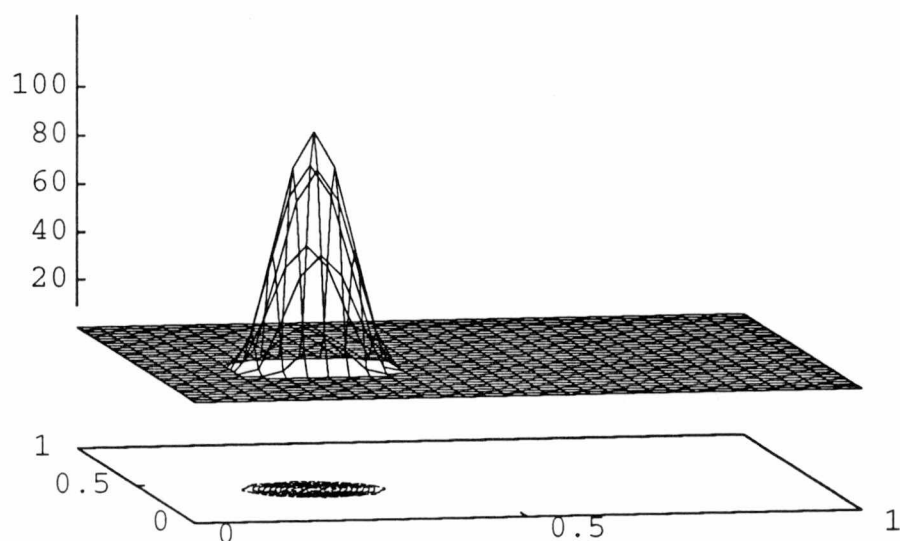


a) CNFD solution perspective plot

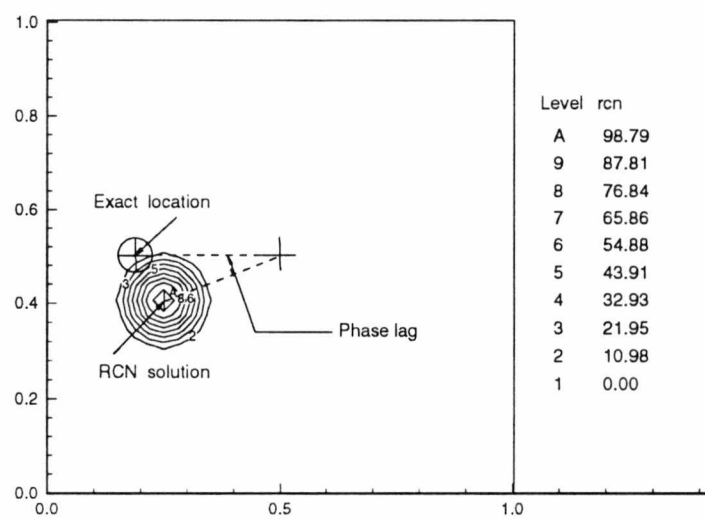


b) Contour plot for data in a)

Figure 8.34 Rotating cone verification CNFD solution, $C = 1$, on 33^2 uniform mesh.



c) RCN solution perspective plot



d) Contour plot for data in c)

Figure 8.34 Rotating cone verification CNFD solution, $C = 1$ (contd.).

becomes apparent. This is particularly evident in the solutions reported by Nallasamy and Prasad (1977). Nevertheless, the fourth-order accurate spline method of Rubin and Khosla (1977) remains satisfactory with a 17×17 mesh at $Re = 1000$ but the corresponding computer time is quite large.”

In this dissertation, I have studied the driven cavity problem via (2.30)-(2.32) using GWS^h , TWS^h and SGM^h algorithms for a range of Reynolds numbers ($100 \leq Re \leq 3200$) on coarse ($Nnode = 17 \times 17$) to moderately dense ($Nnode = 65 \times 65$) meshes. The solution comparisons are documented in this section.

Figure 8.35 summarizes the vorticity, streamfunction GWS^h algorithm solution including pressure and velocity vector distributions for $Re=100$, on a uniform, $Nnode = 17 \times 17$ mesh. All solution fields appear nondispersive (except for pressure) and comparable (in the “eye ball norm”) to GWS^h solutions obtained on uniform $Nnode = 33 \times 33$, Figure 8.36, and $Nnode = 65 \times 65$ meshes, Figure 8.37. The available theory, Oden and Reddy (1976), suggests estimating GWS^h solution trends in the energy norm, which are plotted in Figure 8.38. These data confirm the theoretical prediction of asymptotic convergence with mesh refinement for vorticity, streamfunction and pressure for this moderate Reynolds number ($Re=100$). For linear basis ($k = 1$ in 3.18), the slope of solution error convergence agrees well with the predicted slope, (4.26), as the mesh becomes moderately dense (1089 nodes) to dense (4225 nodes), while for the very coarse (289 nodes) mesh, the solution resolution fails to adhere to the convergence line. The solution energy norms actually computed for the vorticity Ω^h and the streamfunction Ψ^h on the three different meshes are also plotted in

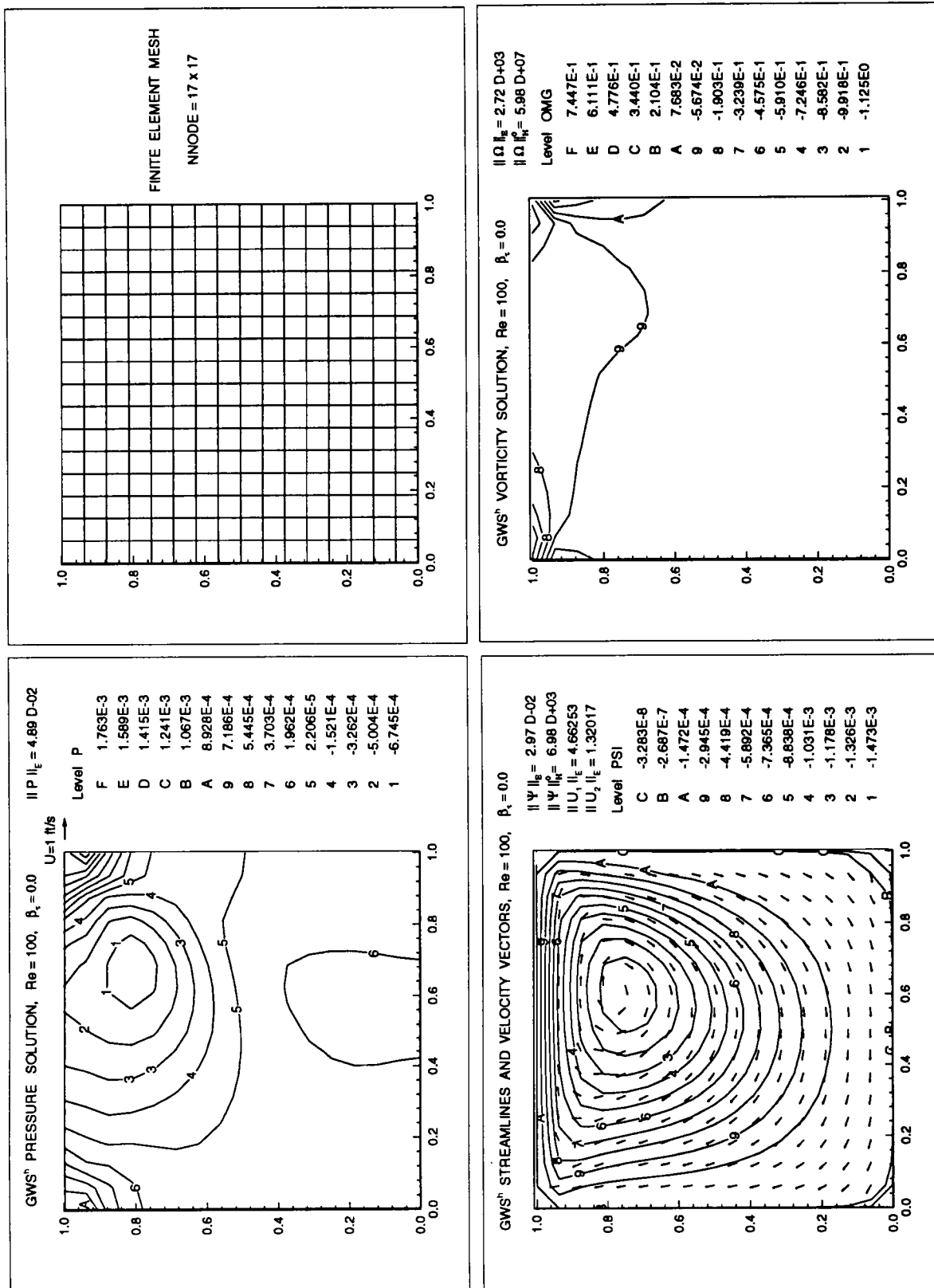


Figure 8.35 GWS^h solution for driven cavity on a uniform Nnode = 17×17 mesh, $Re = 100$.

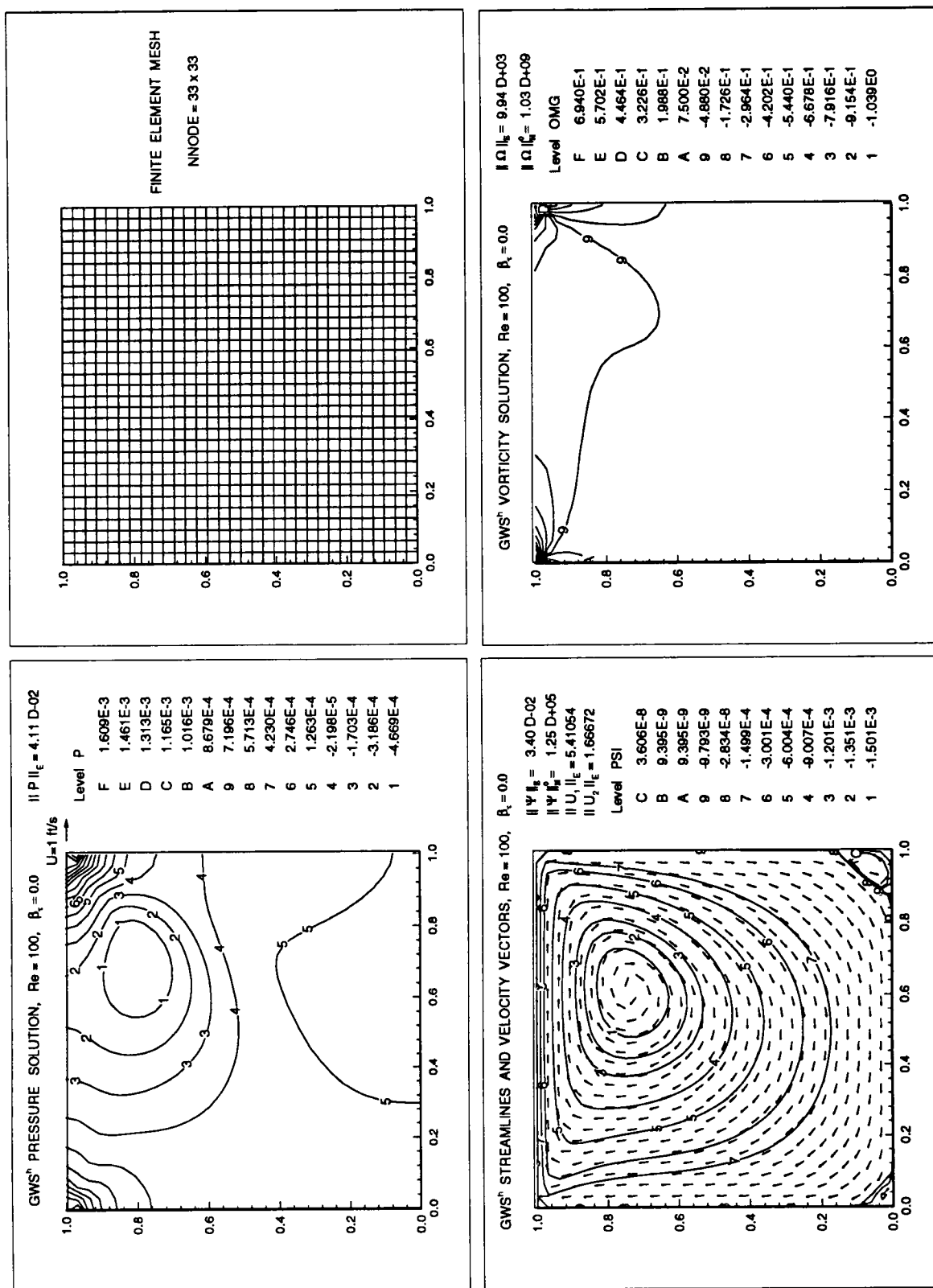


Figure 8.36 GWS^h solution for driven cavity on a uniform Nnode = 33 × 33 mesh, $Re = 100$.

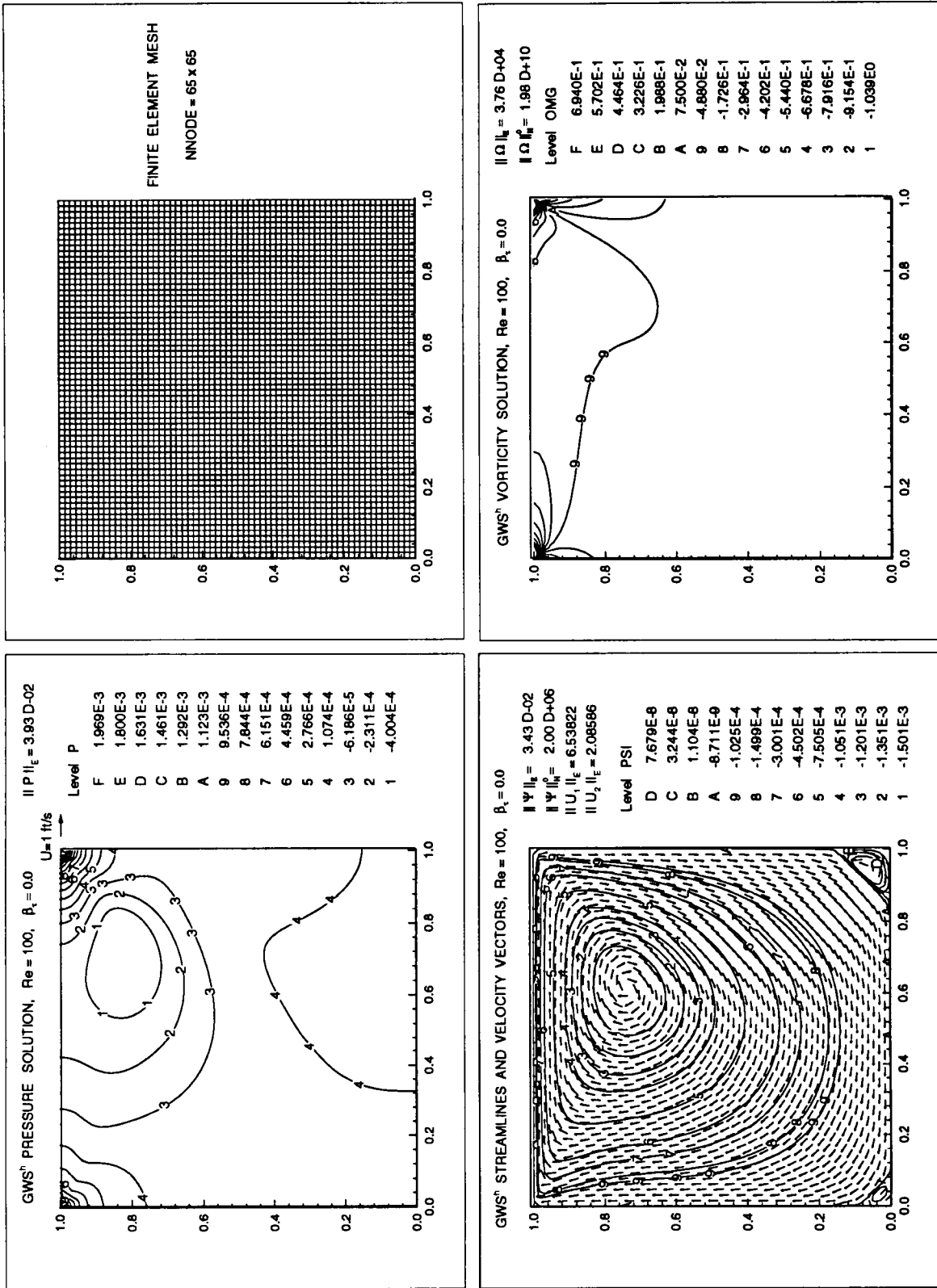


Figure 8.37 GWS^h solution for driven cavity on a uniform Nnode = 65 × 65 mesh, Re = 100.

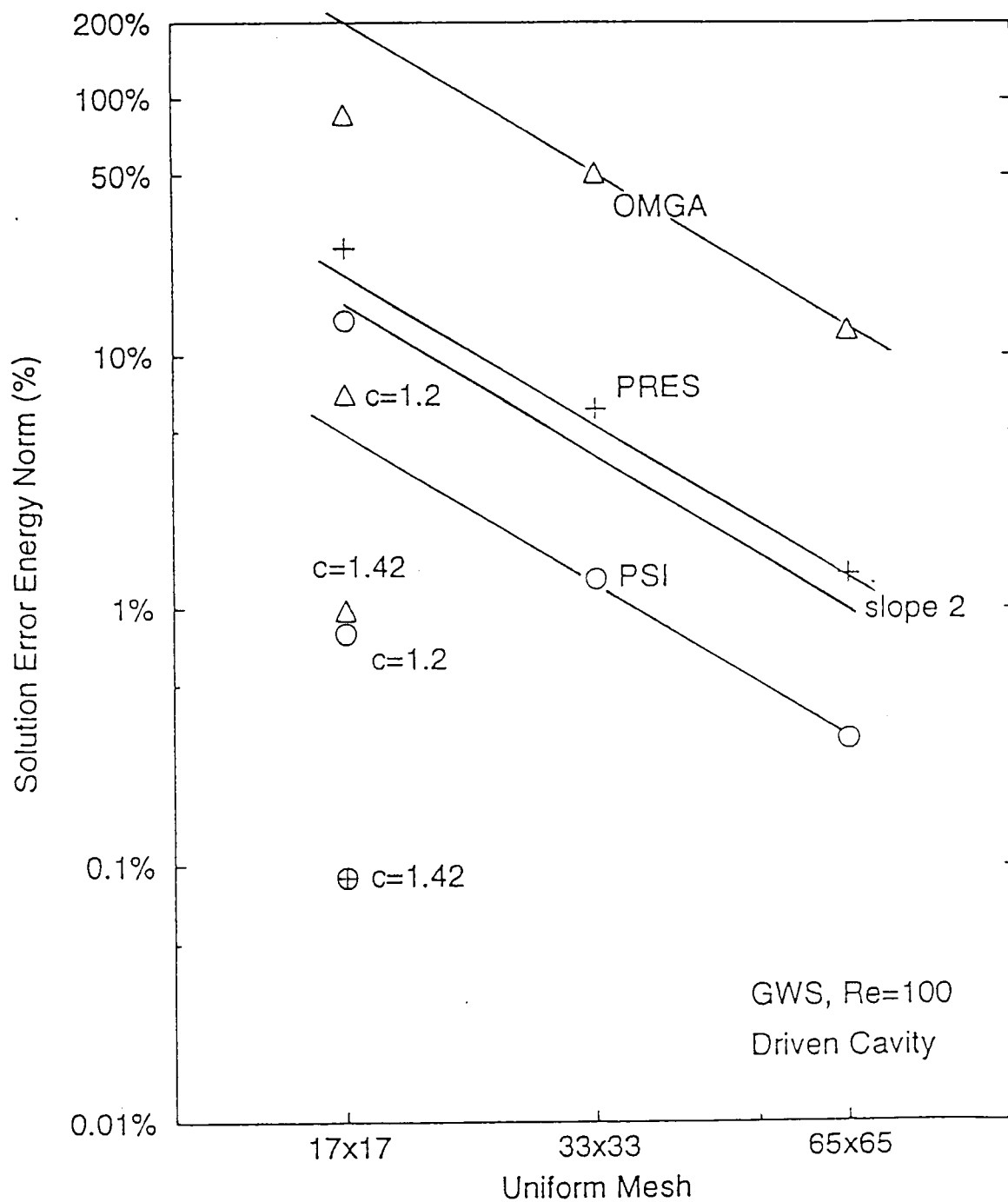
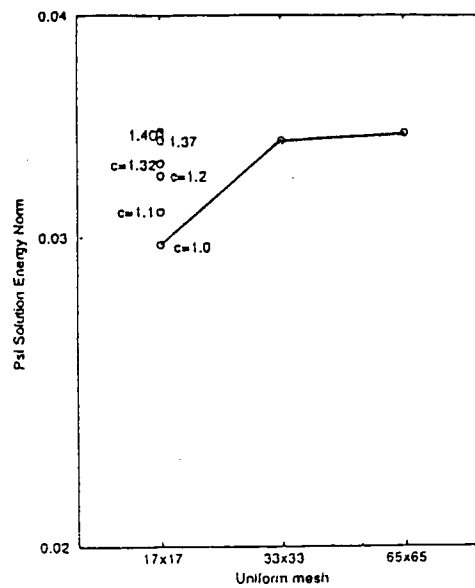


Figure 8.38 Convergence in the GWS^h solution energy norm for streamfunction (PSI), vorticity (OMGA) and pressure (PRES), driven cavity, Re = 100.

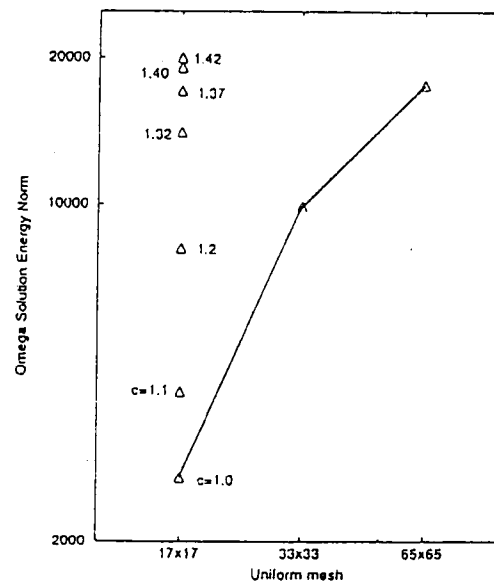
Figure 8.39a-b. The energy norms reaches a near plateau for the Ψ^h solution at 4225 node mesh in Figure 8.39a, while the Ω^h energy keeps on increasing, Figure 8.39b. Using the knowledge of geometric progression, Appendix K, the streamfunction solution energy norm for an infinite number of discretization can be estimated as $E(\Psi^{h/2^\infty}) = 3.44\text{E-}02$, and the corresponding vorticity solution energy norm estimate is $E(\Omega^{h/2^\infty}) = 2.02\text{E+}04$.

The application of the SGM theory, with a uniform parameter $c > 1$ to the problem can reduce the error in the $N_{\text{node}} = 17 \times 17$ mesh solution energy to a minimum level ($\approx 0.1\%$) comparable to that obtained on the $N_{\text{node}} = 65 \times 65$ GWS^h solution. As shown in Figure 8.38, application of uniform SGM $c > 1$ on the coarsest mesh yields a steady improvement of the solution energy norm for both CL variables, indicating convergence of the physical solution. The computation of c was done using domain average of u and h for Re . Note that for higher Reynolds number this scalar estimation of c will not hold simply because of a wider variation of the local velocity vectors u and v . Then a nodewise distributed directional SGM^h parameter $r_{\text{sgm};i,j}$ (for i node and j direction) would be necessary, which follows.

As Reynolds number increases, the impact of the lid boundary condition singularity increases sharply near the top two corner regions. For $Re = 1000$, the comparison of two GWS^h and two TWS^h algorithm solutions is summarized in Figures 8.40-8.43. For the GWS^h algorithm vorticity-streamfunction solution on a coarse ($N_{\text{node}} = 17 \times 17$), non-uniform, modestly non-cartesian mesh, Figure 8.40, both vorticity



a) Streamfunction Ψ energy norm



b) Vorticity Ω energy norm

Figure 8.39 GWS^h solution energy computation for streamfunction and vorticity.

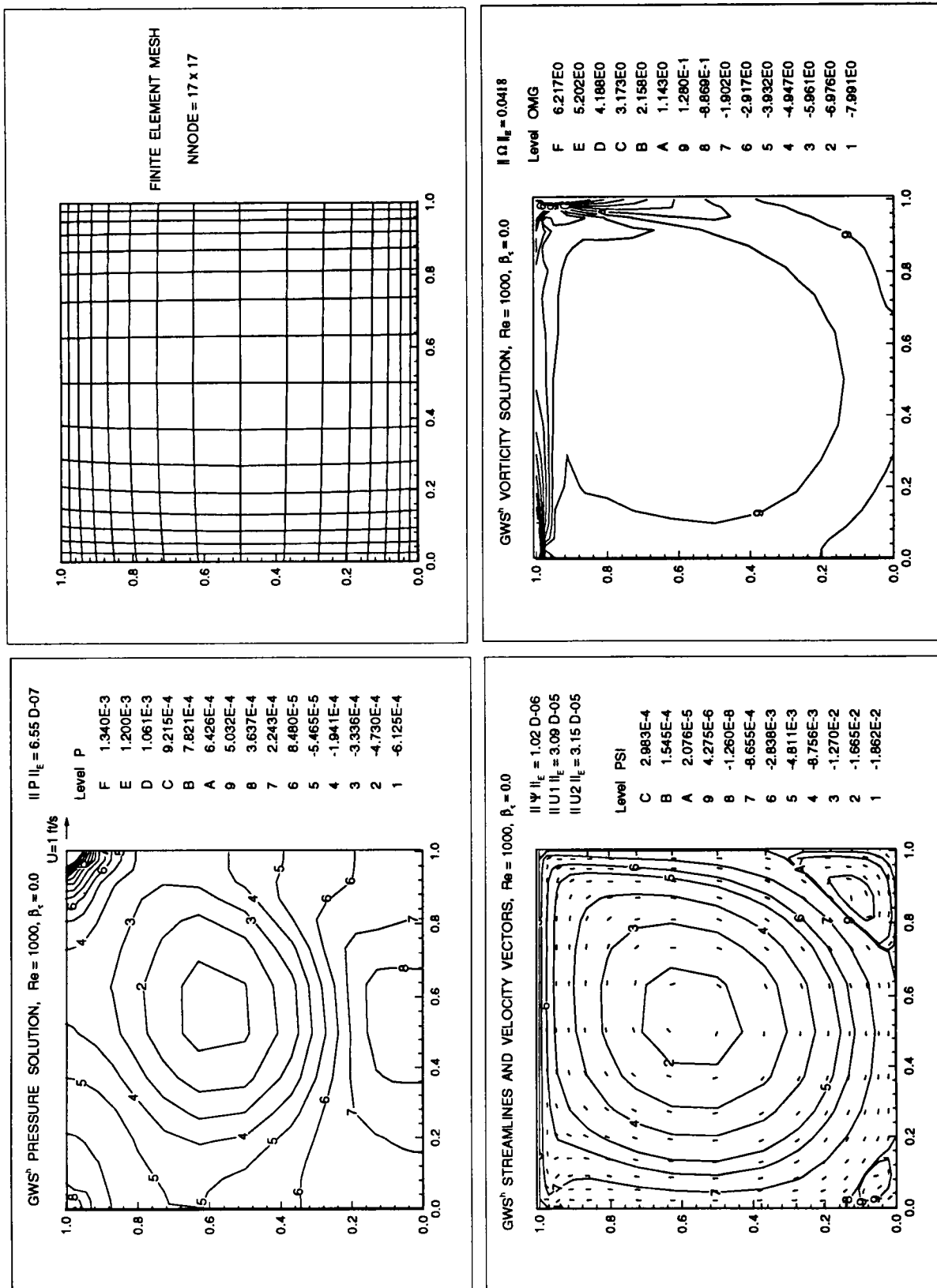


Figure 8.40 GWS^h solution for driven cavity on a non-cartesian mesh Nnode = 17 × 17 mesh, $Re = 1000$.

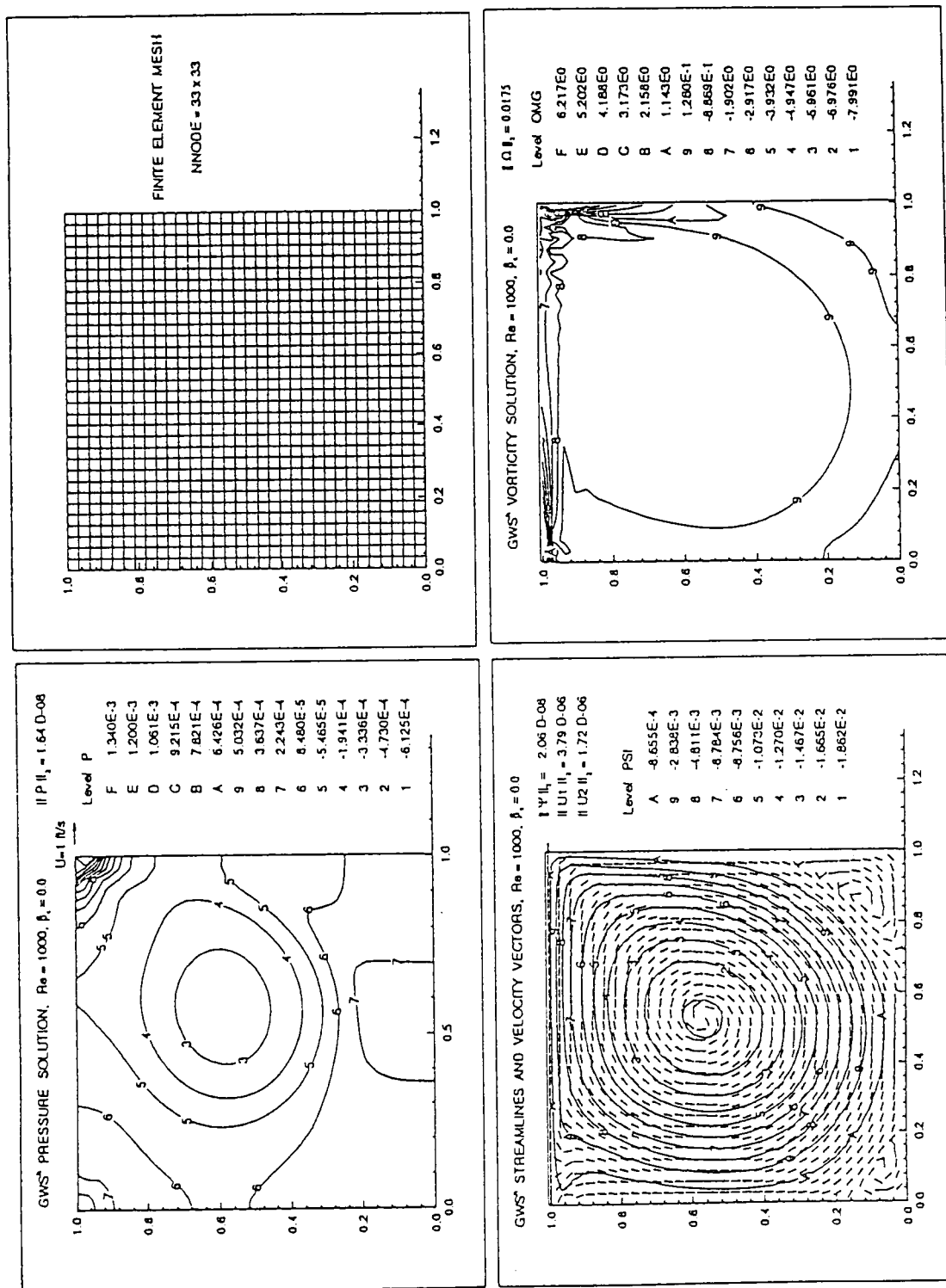


Figure 8.41 GWS^h solution for driven cavity on a uniform mesh, $Nnode = 33 \times 33$, $Re = 1000$.

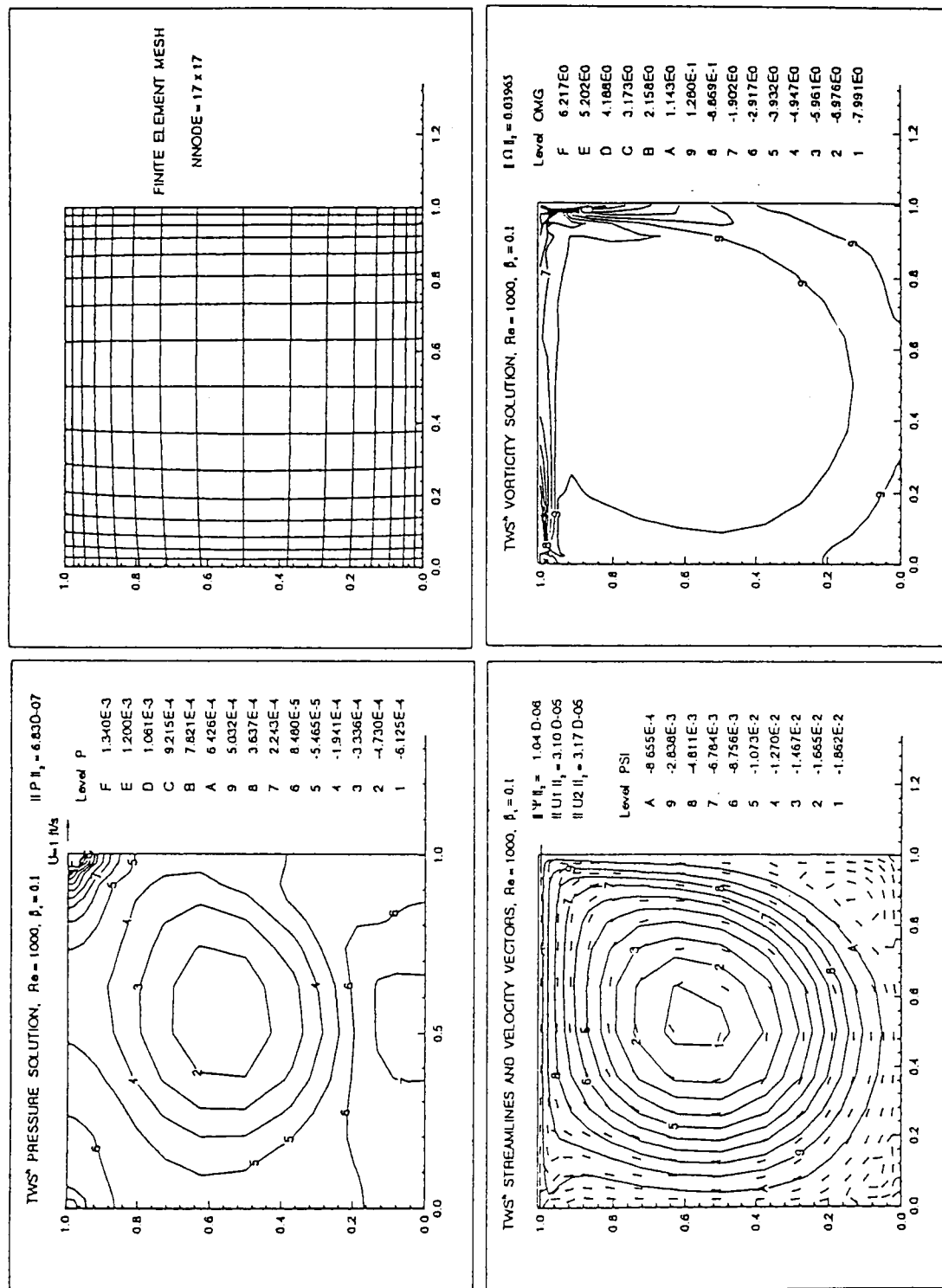


Figure 8.42 TWS^h solution for driven cavity on a non-cartesian mesh, $Nnode = 17 \times 17$, $\beta = 0.1$, $Re = 1000$.

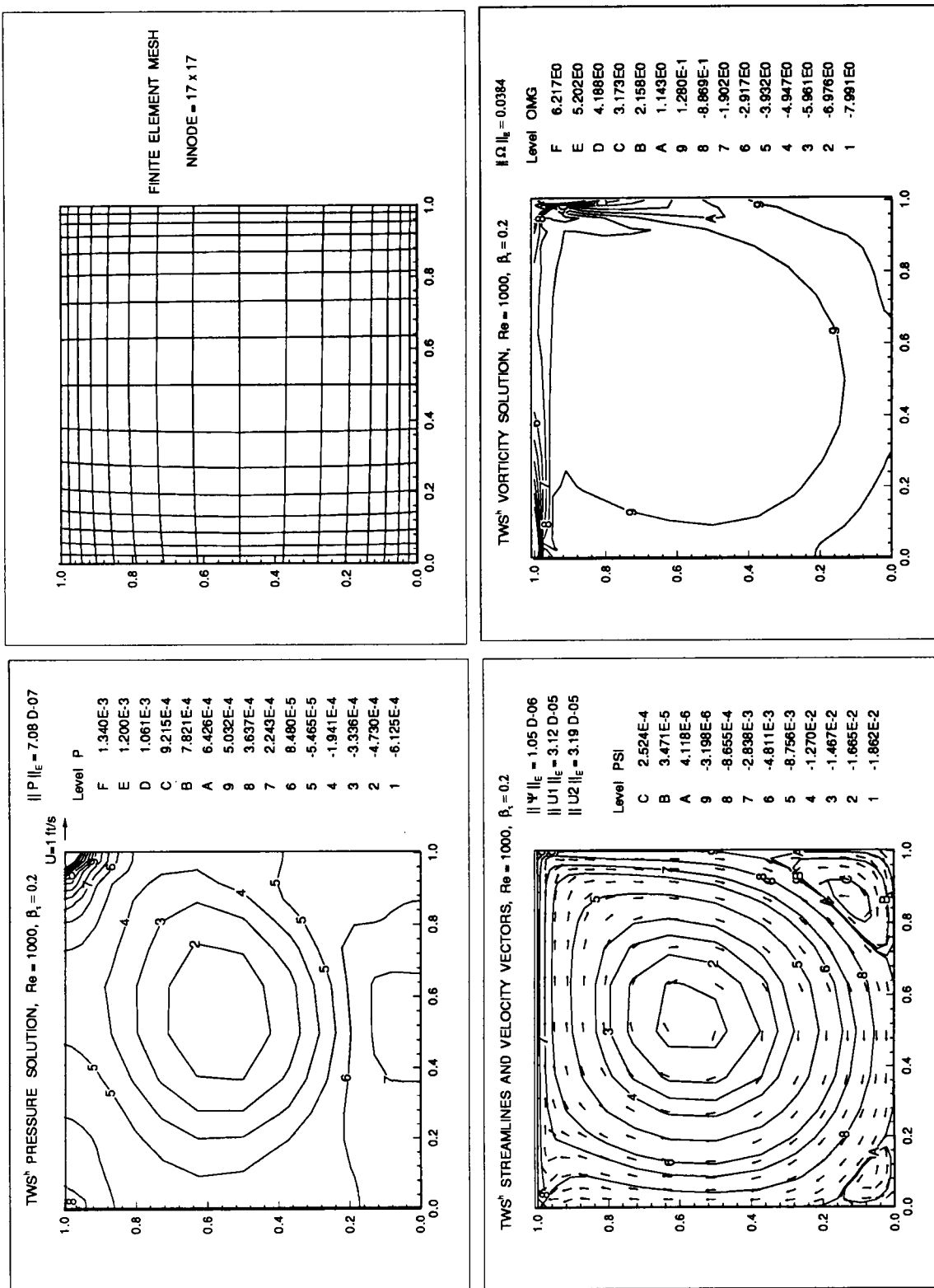


Figure 8.43 TWS^b solution for driven cavity on a non-cartesian mesh Nnode = 17×17 mesh, $\beta = 0.2$, $Re = 1000$.

and pressure solutions contain strong $2\Delta x$ dispersion error waves propagating parallel to the lid, which are especially evident at the two corners (0,1) and (1,1). Conversely, the streamfunction solution is essentially smooth. Ghia, *et. al* (1982) have documented that for $Re = 1000$, the vorticity solution becomes smooth on a very dense 129×129 node uniform mesh. However, increasing discretization to a uniform 33×33 mesh does not improve GWS^h vorticity solution monotonicity, Figure 8.41.

Use of the artificial diffusion mechanism ($\beta > 0$) of the TWS^h algorithm, for $Re = 1000$ on a coarse (17×17 node modestly) non-uniform non-cartesian mesh, produces the comparative solutions documented in Figures 8.42- 8.43. For $\beta = 0.1$, the dispersion error wave mode along the lid and near the two corners is reduced considerably (though not completely), Figure 8.42. The streamfunction solution remains non-oscillatory. Use of the higher $\beta = 0.2$ level does not significantly improve the vorticity solution monotonicity, Figure 8.43, although the pressure distribution becomes modestly smoother near the upper right corner.

One can conclude, therefore, that neither the GWS^h nor TWS^h $k = 1$ basis algorithms produce a satisfactory solution for the driven cavity on a 17×17 mesh at $Re = 1000$. Therefore, this represents an excellent opportunity to assess the comparative performance of the SGM algorithm. The comparison SGM solution, for a uniform $c = 1.2$, based on an average u and h on the same non-cartesian mesh ($Nnode = 17 \times 17$) is shown in Figure 8.44. The monotonicity of the vorticity solution therein is comparable to the $\beta = 0.1$ TWS^h solution, Figure 8.42, except that the trace of $2\Delta x$ error wave near the upper right hand corner of the cavity is somewhat reduced.

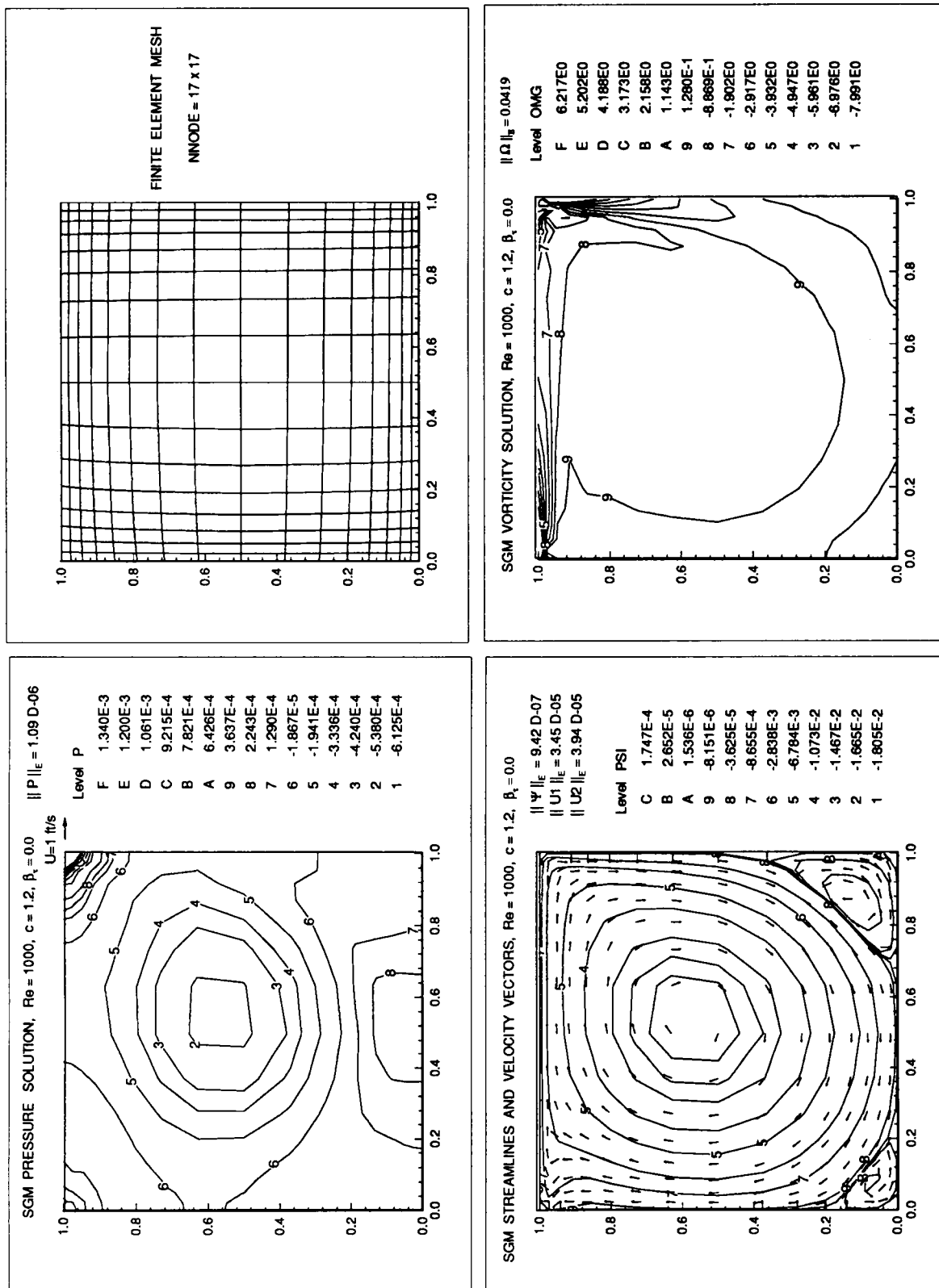


Figure 8.44 SGM solution for driven cavity on a non-cartesian mesh Nnode = 17×17 mesh, $c = 1.2$, $Re = 1000$.

The direct comparison of the GWS^h and SGM algorithm streamfunction and vorticity solutions for $Re = 1000$ on the 17×17 node mesh is presented in Figures 8.45-8.46. In Figure 8.45, both streamfunction solutions are essentially non-oscillatory (ENO) and nominally identical (in the eye-ball norm). However, in the perspective presentation of Figure 8.46, the GWS^h and SGM vorticity solution comparison clearly shows a distinct improvement in the $2\Delta x$ error mode (for $c = 1.2$) although a “trace of wiggle” still exists at the stagnation corner.

For $Re = 2000$, on the same 17×17 nodal mesh, the vorticity solution perspective plot, Figure 8.47a, confirms that the GWS^h dispersion error mode dominates near the lid. In comparison, this error mode is substantially reduced in the SGM solution for uniform $c = 2.0$, Figure 8.47c, although a trace of dispersion error is now evident all along the lid. Hence, both solutions remain poor on this mesh; furthermore, the SGM vorticity distributions is quite diffused when compared to the non-uniform, denser mesh GWS^h solution, Figure 8.47b. The corner sharp singularities present in Figure 8.47b are virtually absent in both Figure 8.47a and Figure 8.47c. This indicates the need for implementation of a nodally distributed embedding parameter c_j according to the theory (5.74), r_j in (5.55).

For this verification, I pushed the Reynolds number to $Re = 3200$. The GWS^h vorticity solution contours and the perspective plots, Figure 8.48a-b, on the base 17×17 coarse non-uniform mesh is totally trashy near the lid. The comparison SGM^h vorticity solution, with optimal nodally distributed parameter $r_{sgm_j} \geq 1$ on the same coarse mesh, contains solution contours and perspective, Figure 8.49a-b,

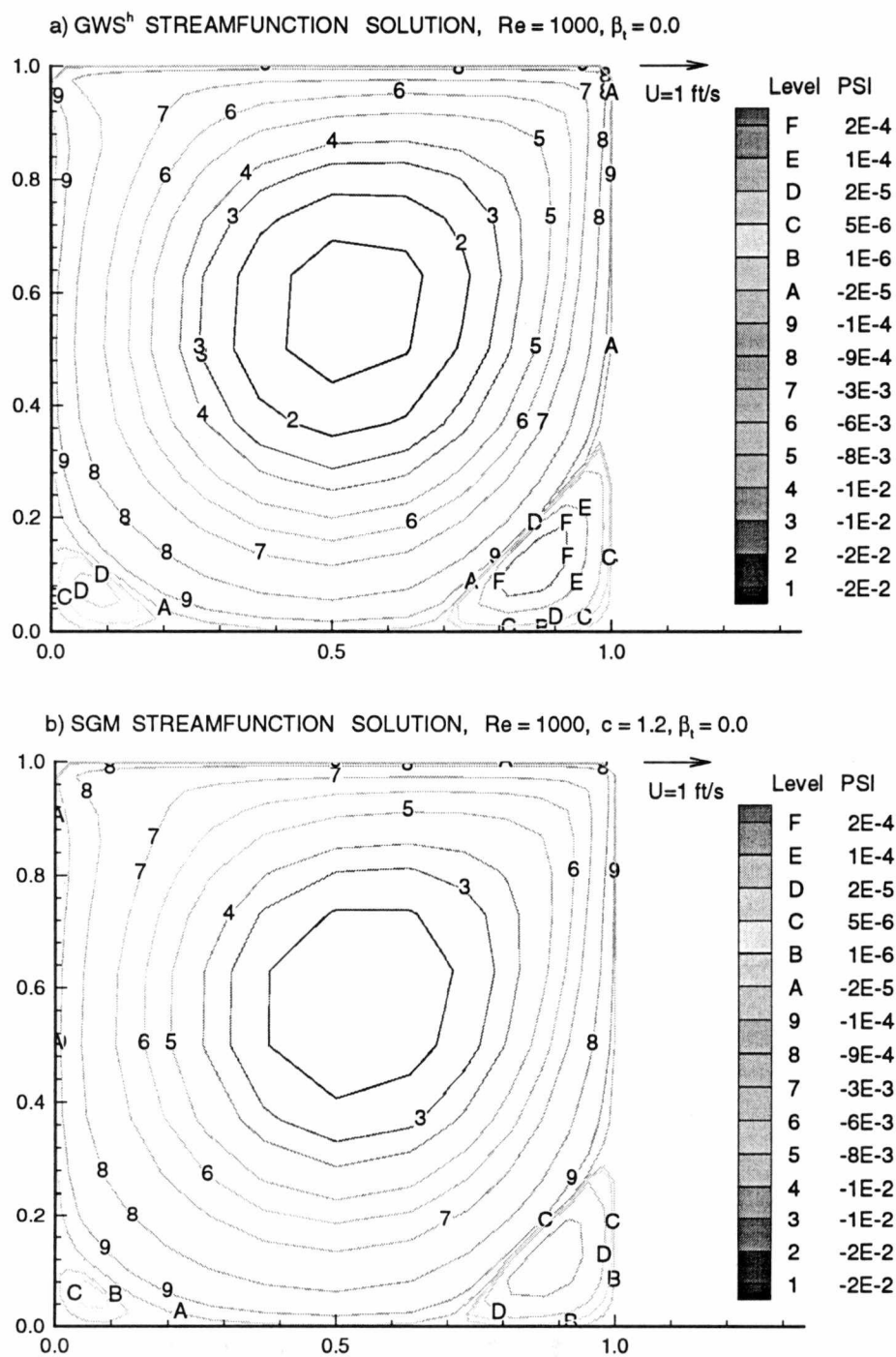
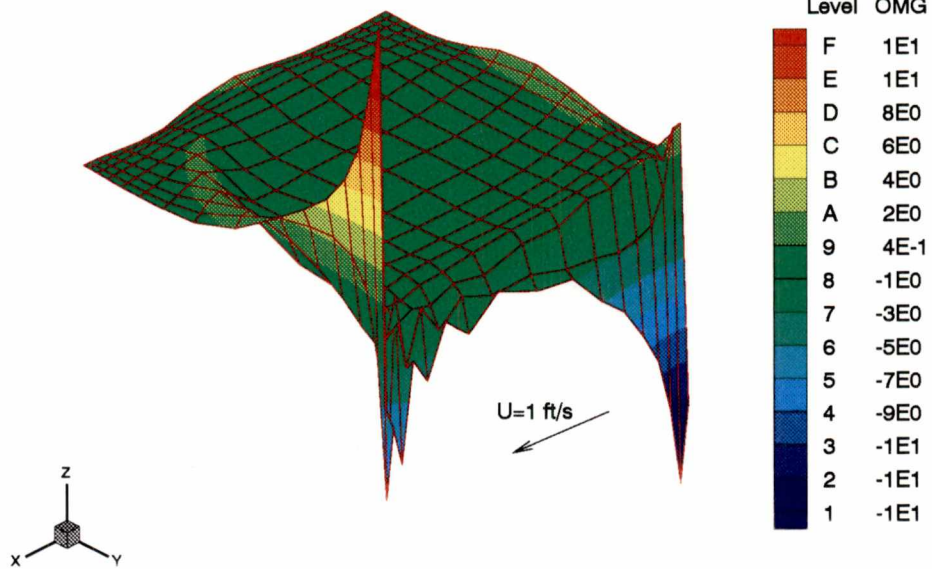


Figure 8.45 Comparison GWS^h and SGM streamfunction solution for driven cavity ($Re = 10^3$) on 17^2 node mesh.

a) GWS^h VORTICITY SOLUTION, $Re = 1000$, $\beta_t = 0.0$



b) SGM VORTICITY SOLUTION, $Re = 1000$, $c = 1.2$, $\beta_t = 0.0$

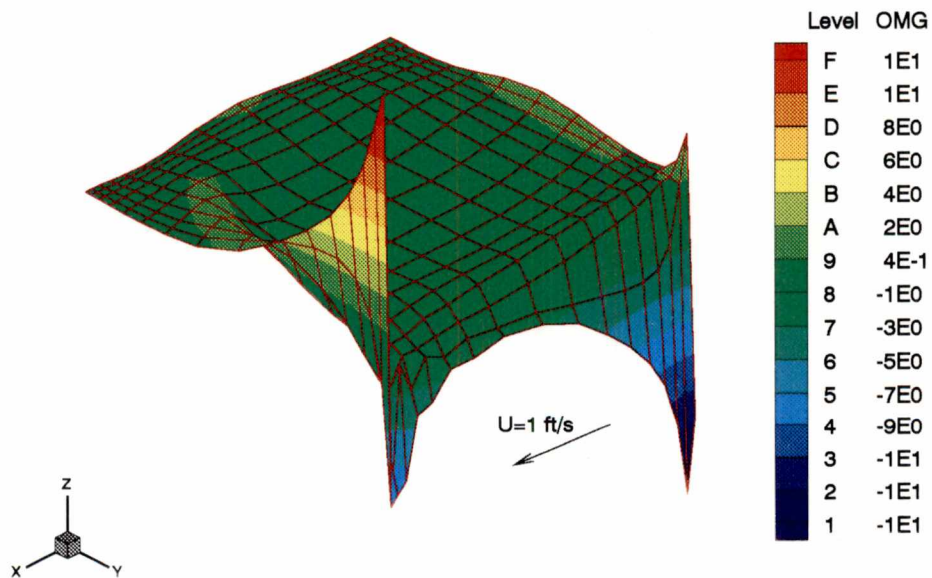


Figure 8.46 Comparison GWS^h and SGM vorticity solution for driven cavity ($Re = 10^3$) on 17^2 node mesh.

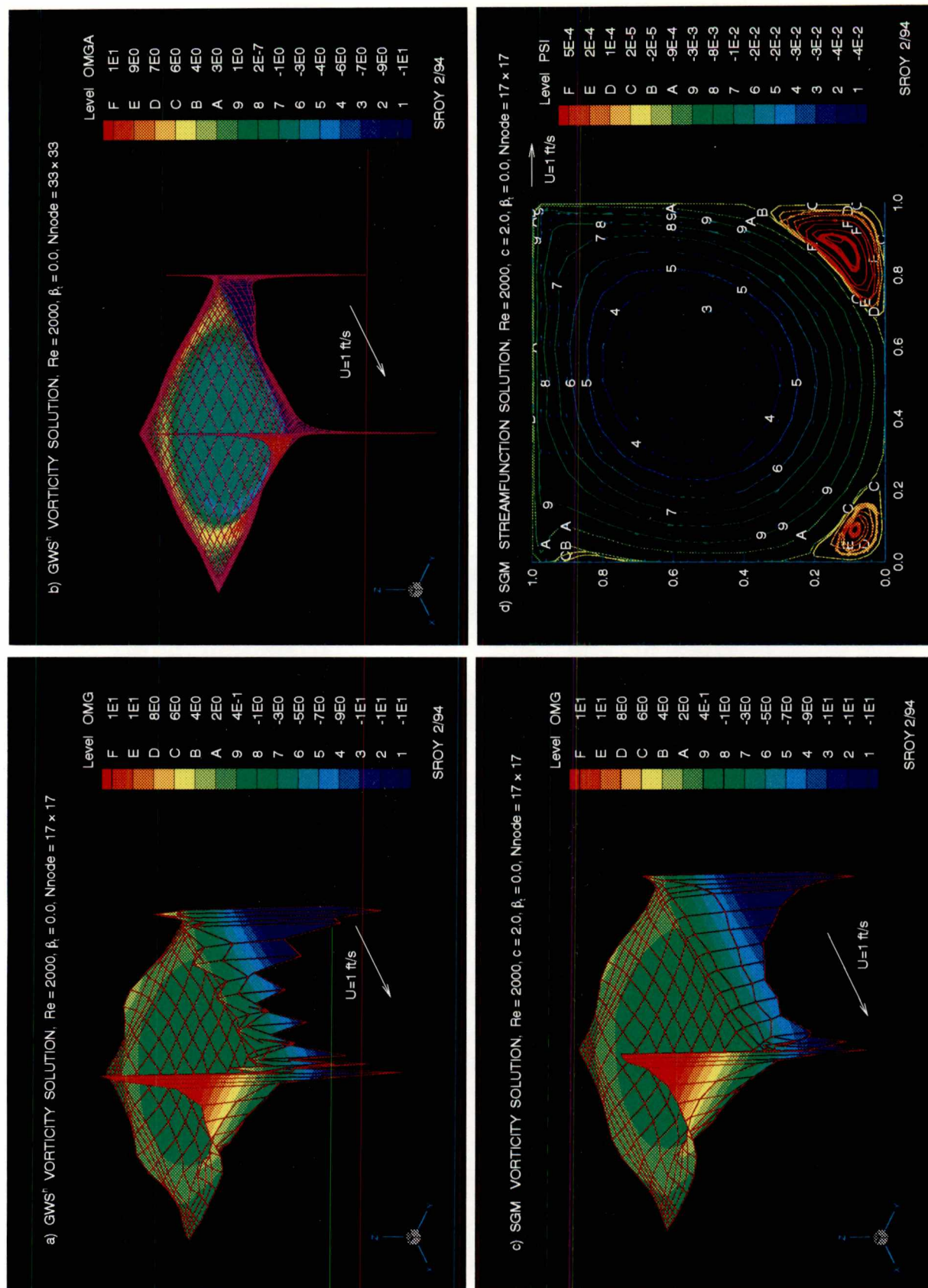


Figure 8.47 GWS^h and SGM^h solution comparison for driven cavity ($Re = 2000$) on different grids.

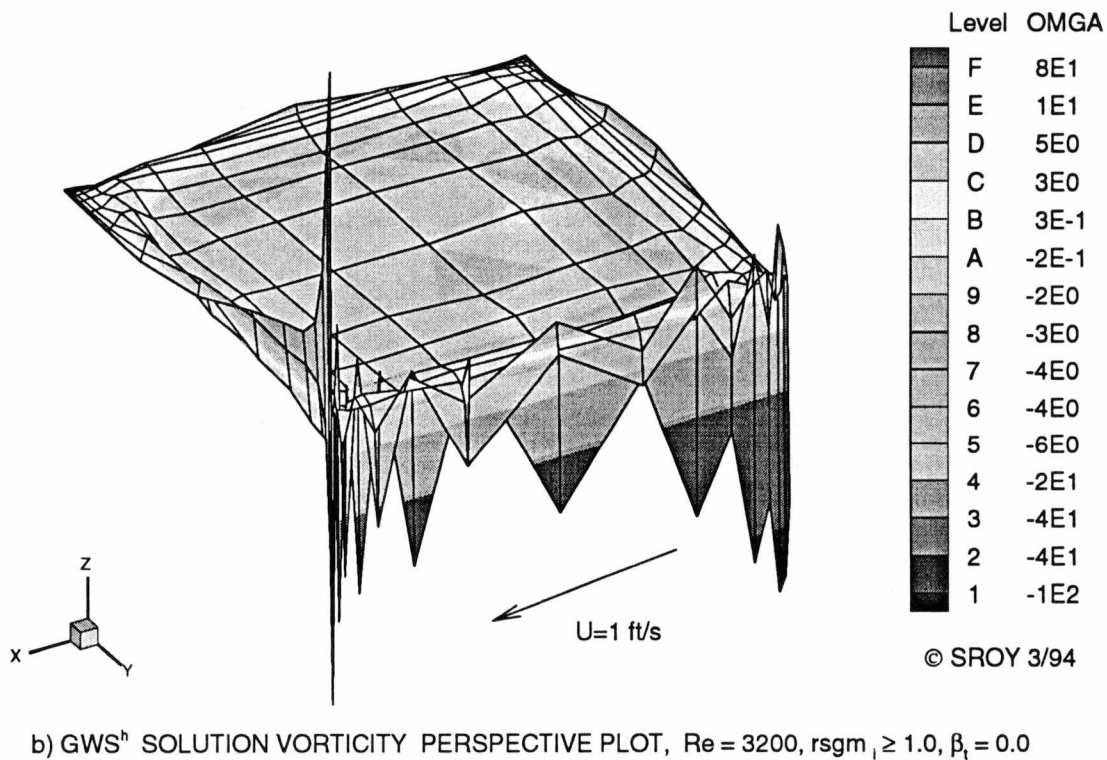
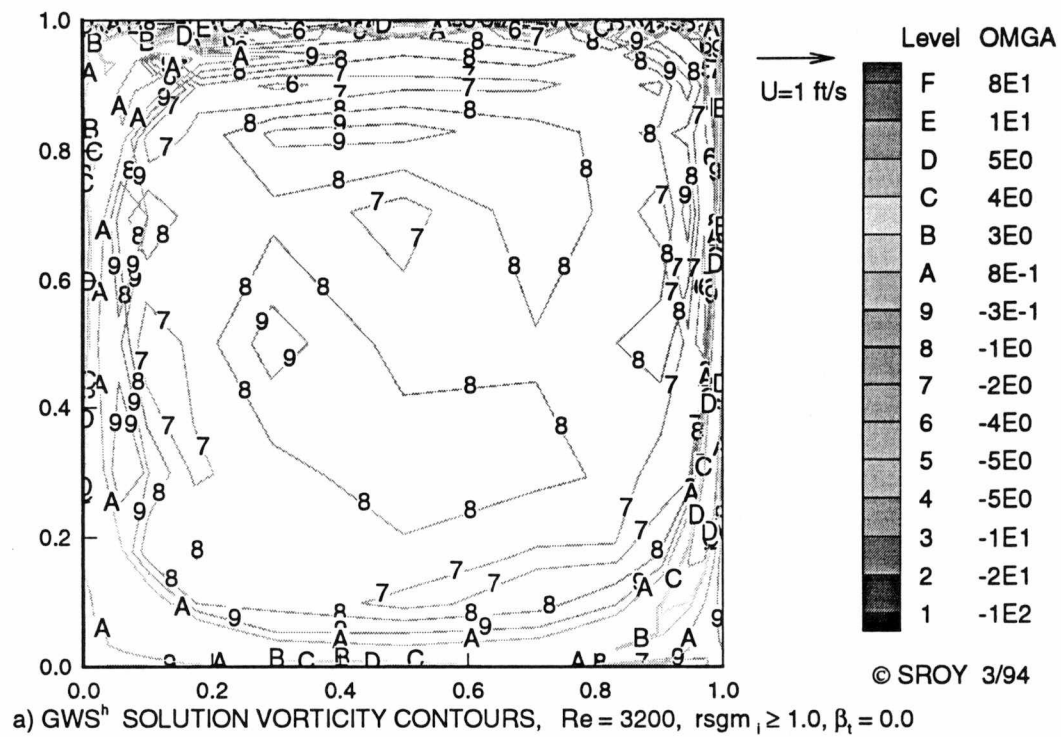


Figure 8.48 GWS^h vorticity solution for driven cavity ($Re = 3200$) on a $M=16^2$ (289 node) mesh.

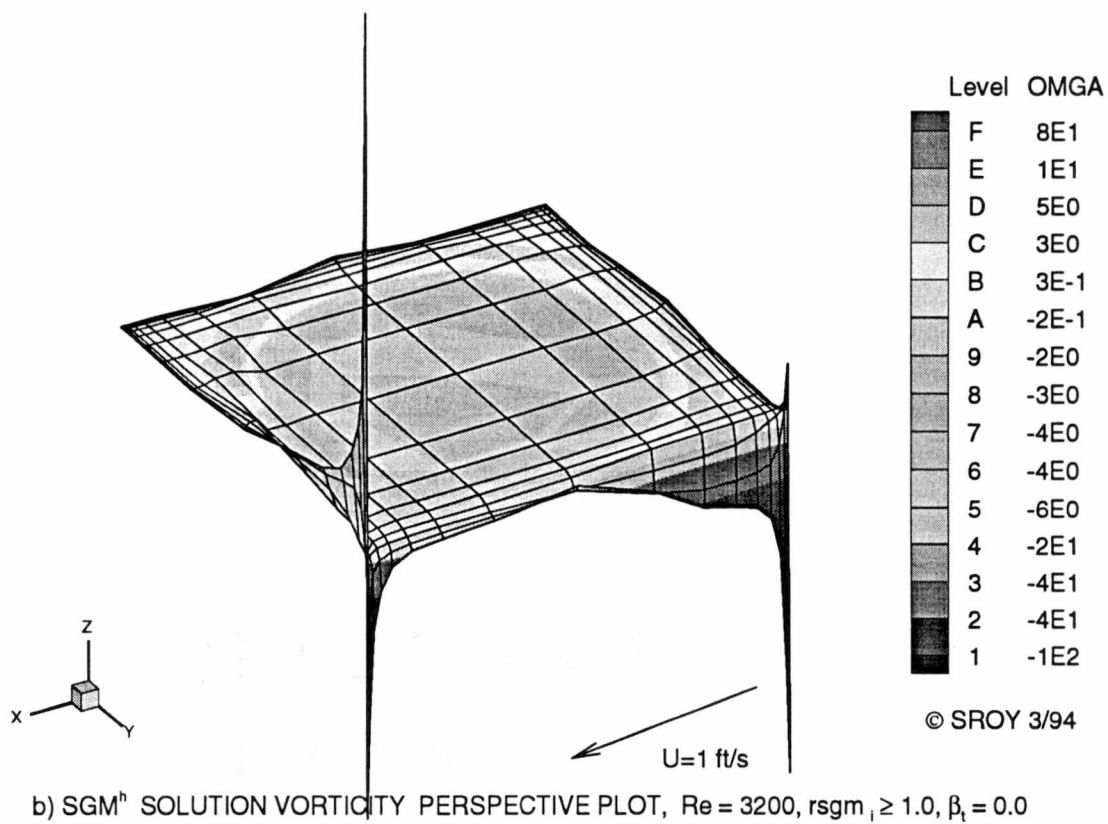
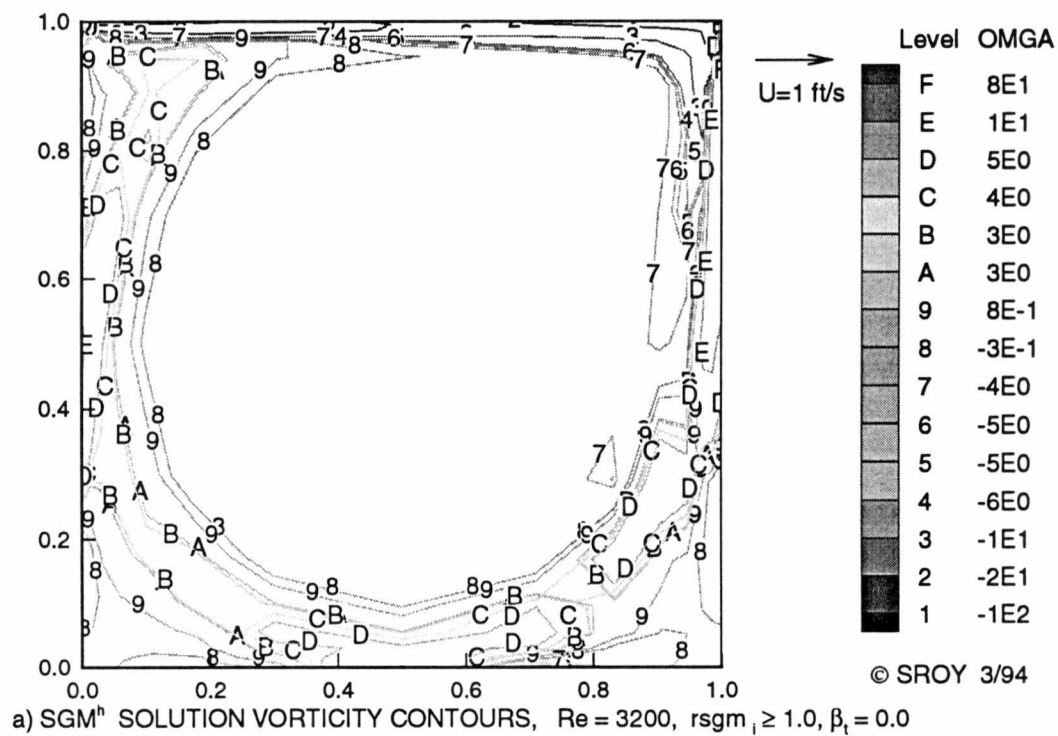


Figure 8.49 SGM^h vorticity solution for driven cavity ($Re = 3200$) on a $M=16^2$ (289 node) mesh.

verifying attainment of a monotone solution with very sharp extrema (singularities) at the (0,1) and (1,1) corners. In fact, the solution resolution is comparable to that obtained via GWS^h on a double-density (1089 node) highly non-uniform mesh, Figure 8.50a-b. The computation time for this GWS^h solution evolution was nearly six times that required for the SGM^h solution, Figure 8.49. Figure 8.51 directly compares these GWS^h and SGM^h vorticity solution perspective plots in color with corresponding meshes overlaid.

Although the GWS^h streamfunction solution for $Re = 3200$ is essentially monotone on the 289 node mesh, it fails to correctly locate the four vortex regions of the benchmark solution. The contour plot, Figure 8.52a, clearly shows only three under-developed recirculation regions in the lower right (1,0), lower left (0,0) and upper left (0,1) corners. The GWS^h solution on the 1089 non-uniform mesh does capture these secondary vortices more accurately, Figure 8.53, as does the SGM^h solution on the coarse mesh ($N_{node} = 289$), Figure 8.54.

A fully detailed presentation of the SGM^h streamfunction solution distribution in comparison to benchmark, is given in Figure 8.55. The arrowheads secondary vortex region graphs indicate the reattachment points documented by Ghia, Ghia and Shin (1982, Figure 3) on a $N_{node} = 129 \times 129$ uniform mesh. The agreement of the 17×17 SGM solution is excellent. The u - and v -velocity profiles through the geometric center, Figure 8.55, is another available measure of accuracy. The SGM data agree closely with those generated by the 57 times denser mesh solution of Ghia, *et. al* (1982, Figure 2a and 2b), shown as circles in Figure 8.55.

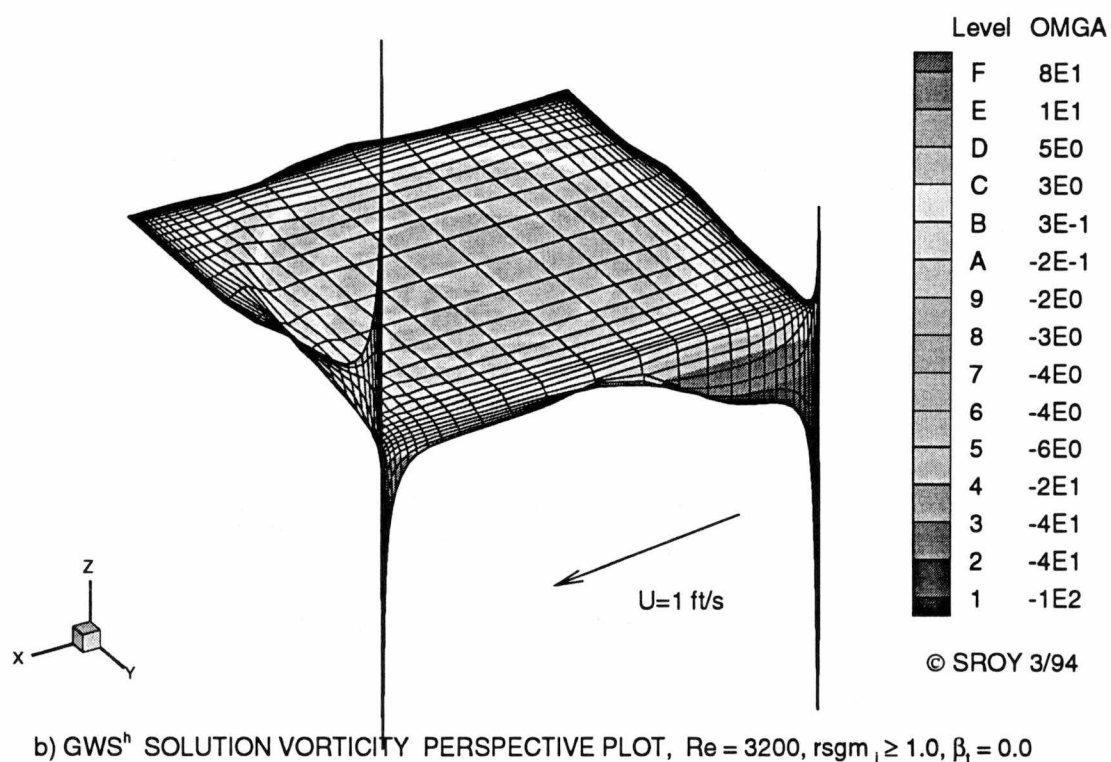
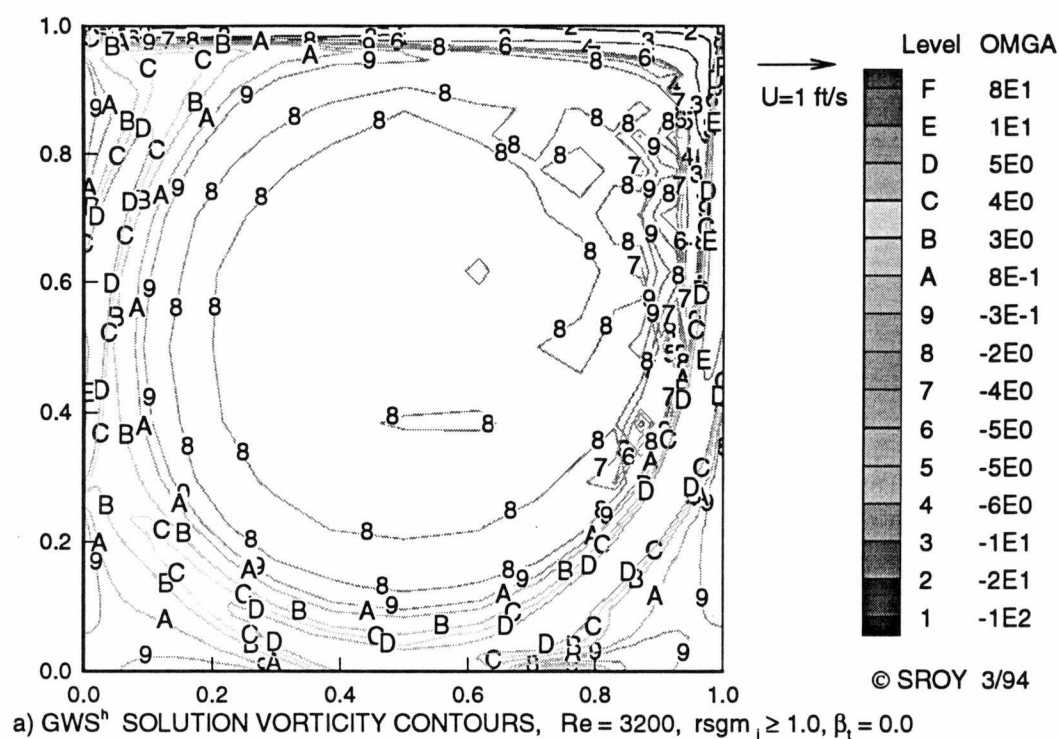


Figure 8.50 GWS^h vorticity solution for driven cavity ($Re = 3200$) on a $M=32^2$ (1089 node) mesh.

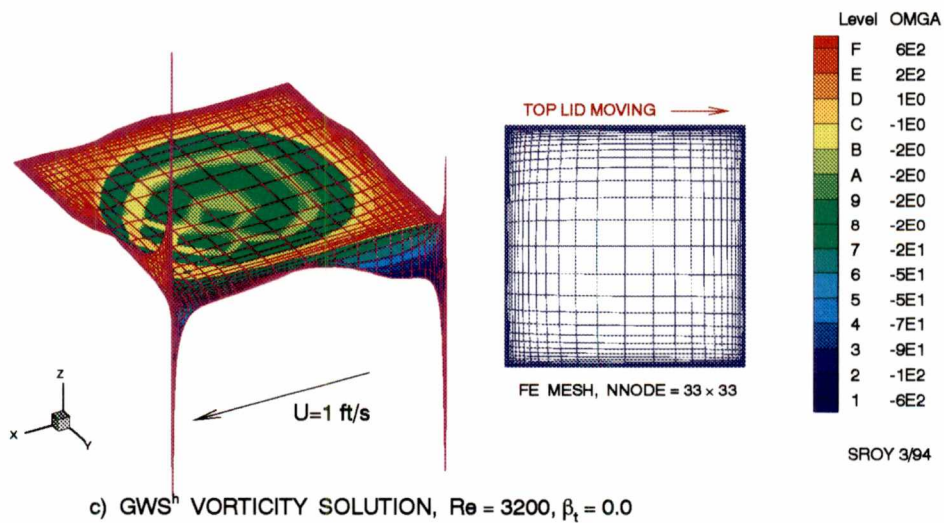
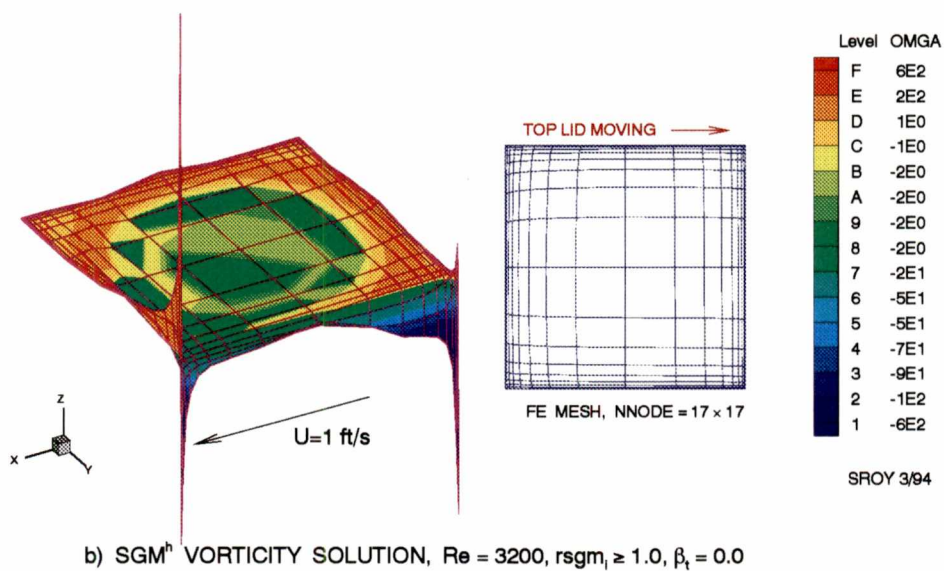
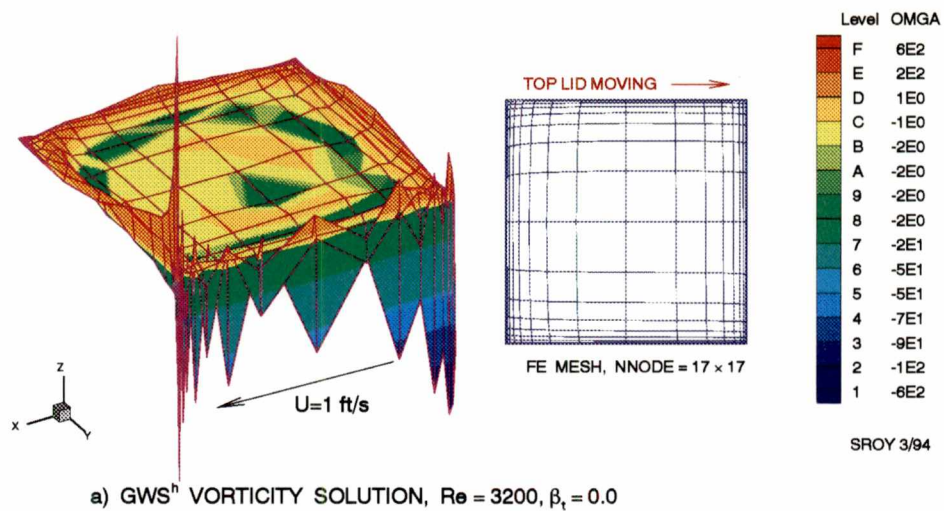


Figure 8.51 GWS^h and SGM^h vorticity solution comparison for driven cavity ($Re = 3200$).

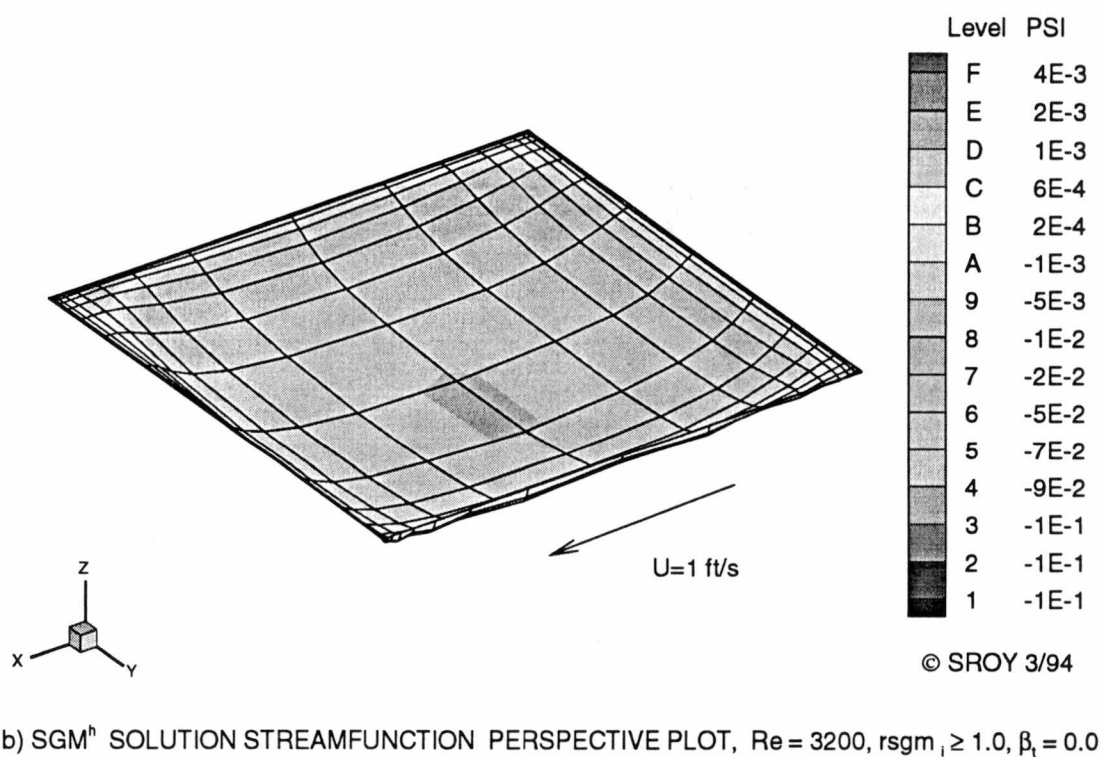
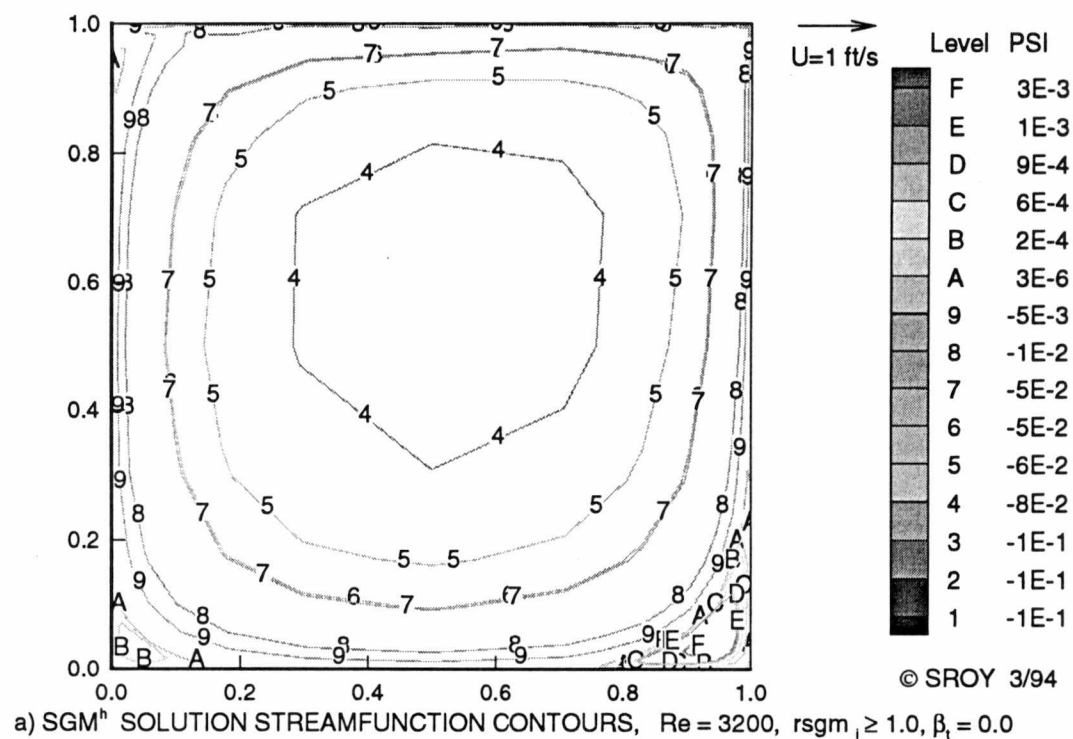
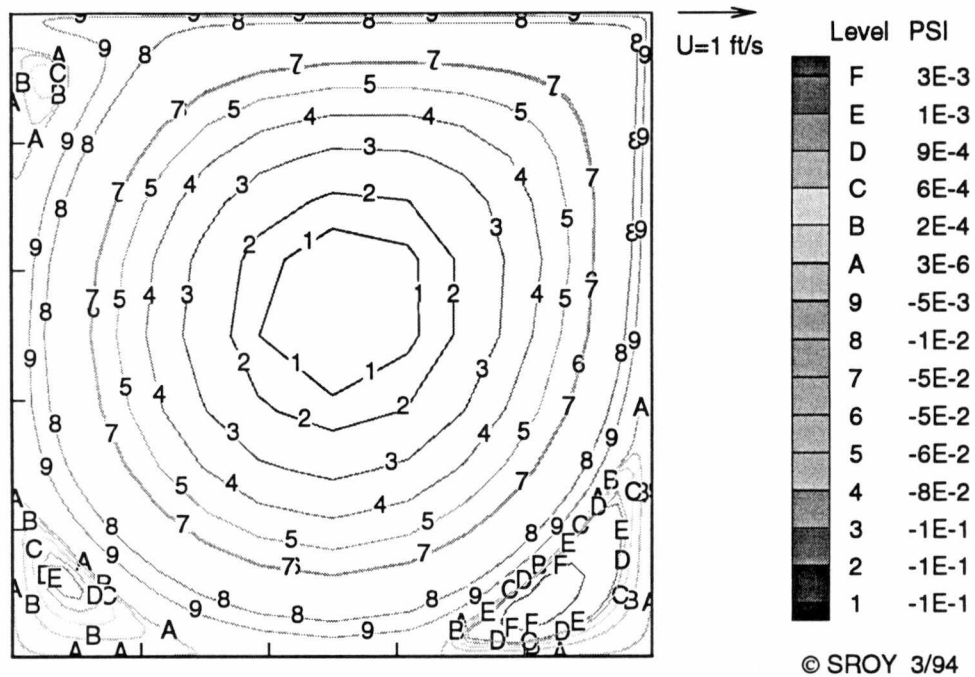
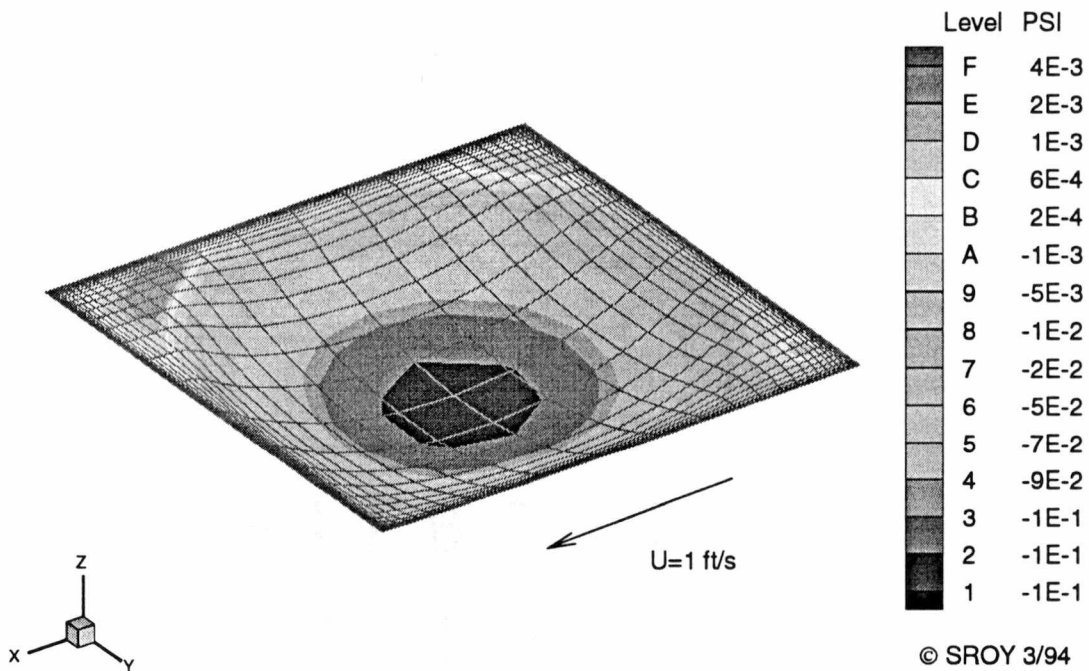


Figure 8.52 GWS^h streamfunction solution for driven cavity ($Re = 3200$) on a $M=16^2$ mesh.

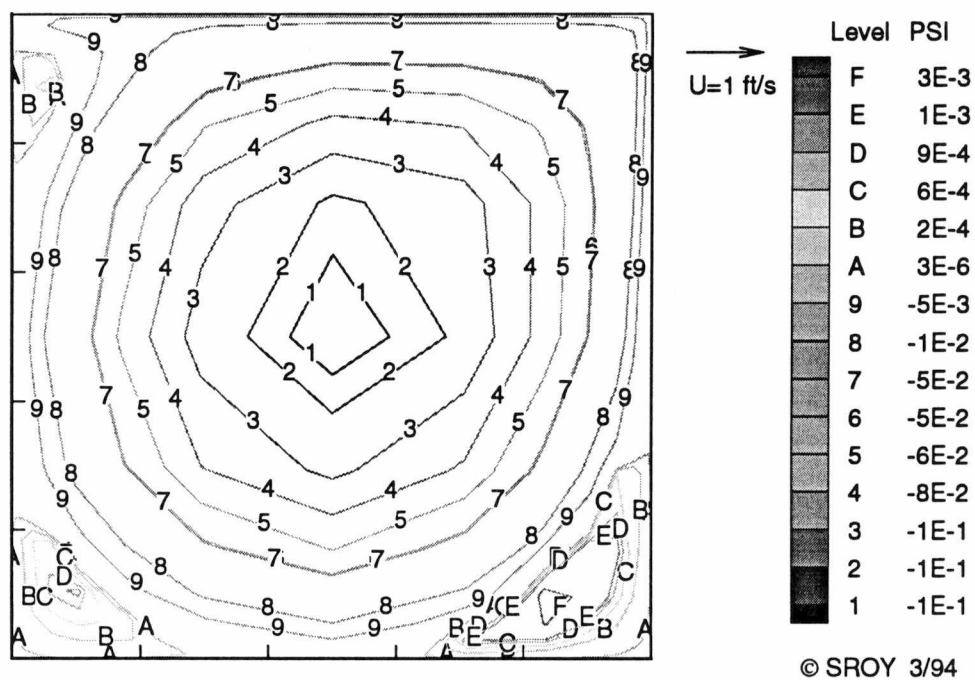


a) GWS^h SOLUTION STREAMFUNCTION CONTOURS, $Re = 3200$, $rsgm_1 \geq 1.0$, $\beta_1 = 0.0$

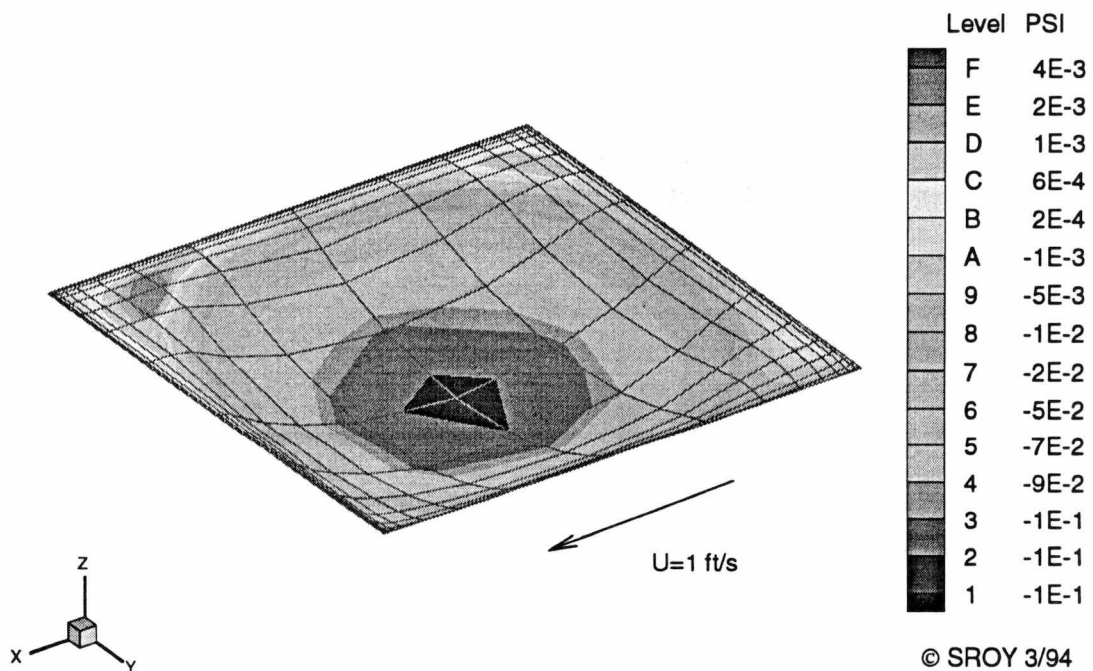


b) GWS^h SOLUTION STREAMFUNCTION PERSPECTIVE PLOT, $Re = 3200$, $rsgm_1 \geq 1.0$, $\beta_1 = 0.0$

Figure 8.53 GWS^h solution for driven cavity ($Re = 3200$) on a $M=32^2$ (1089 node) mesh.



a) SGM^h SOLUTION STREAMFUNCTION CONTOURS, $Re = 3200$, $rsgm_1 \geq 1.0$, $\beta_1 = 0.0$



b) SGM^h SOLUTION STREAMFUNCTION PERSPECTIVE PLOT, $Re = 3200$, $rsgm_1 \geq 1.0$, $\beta_1 = 0.0$

Figure 8.54 SGM^h solution for driven cavity ($Re = 3200$) on a $M=16^2$ (289 node) mesh.

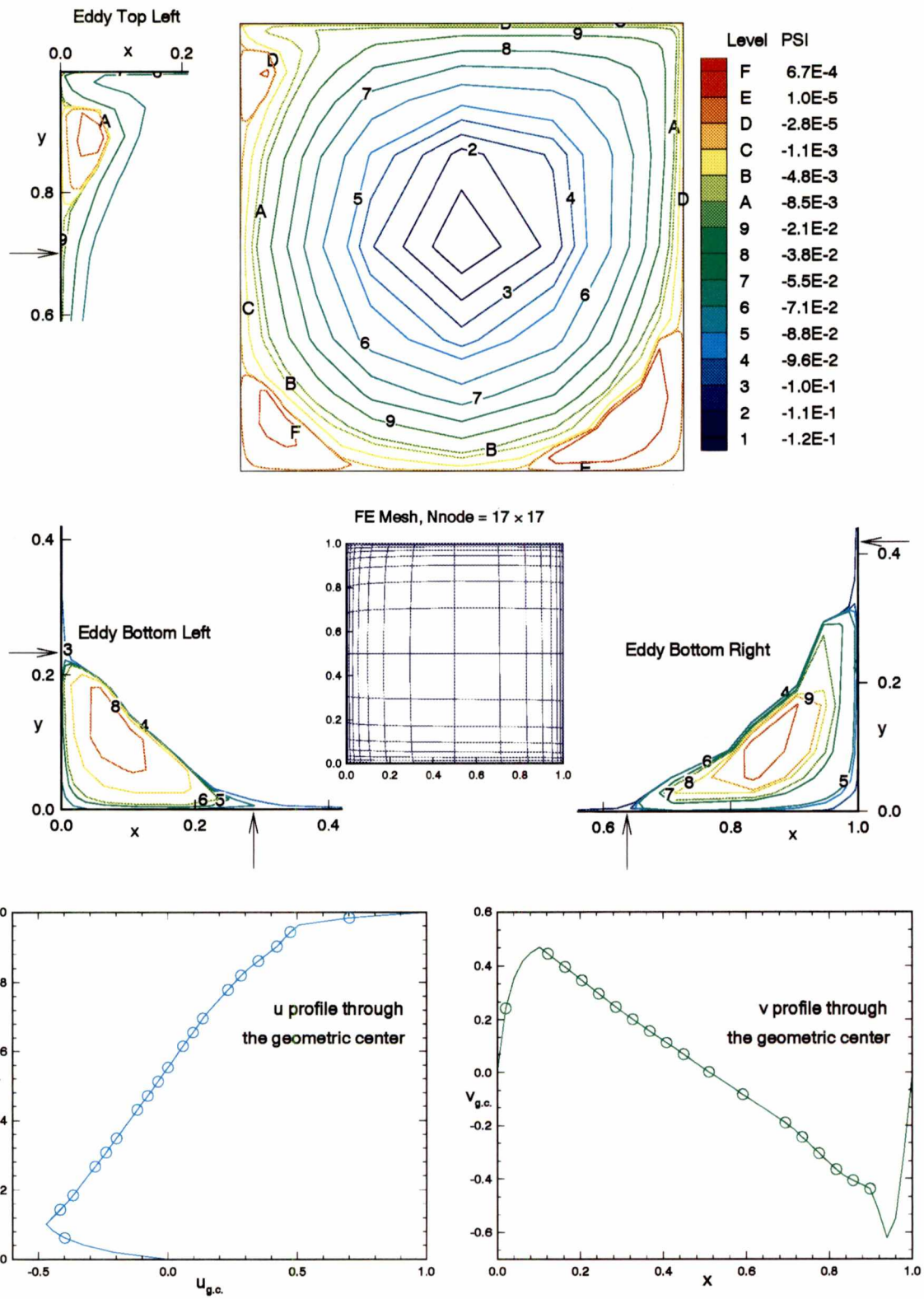
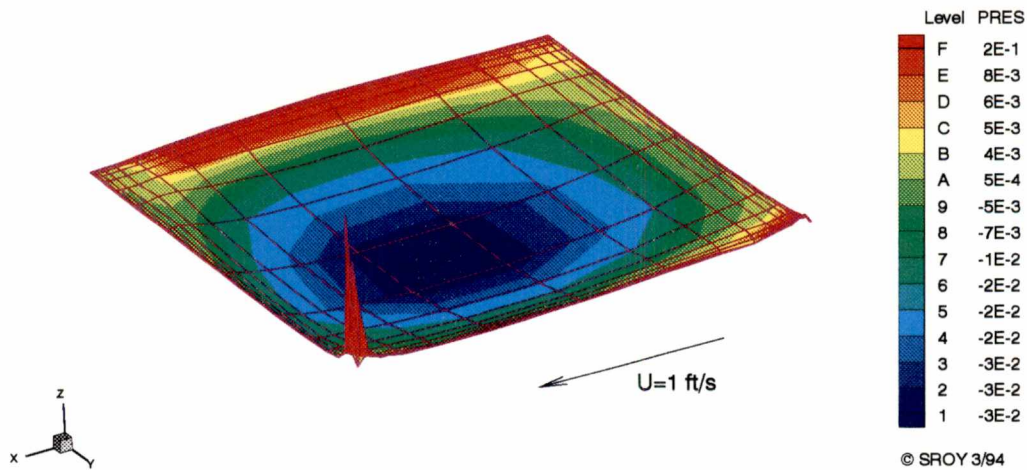


Figure 8.55 SGM^h streamfunction and velocity solution details for driven cavity, $Re = 3200$, $rsgm_1 \geq 1$.

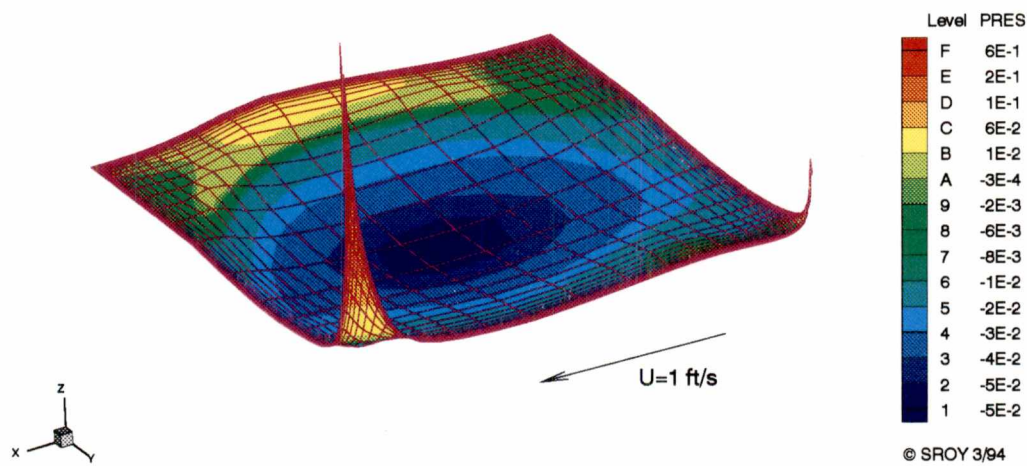
Pressure is an experimentally measureable variable, while vorticity and stream-functions are simply CL variables. The comparison of pressure distributions on coarse ($N_{\text{node}} = 289$) to dense ($N_{\text{node}} = 1089$) mesh GWS^h solutions and the coarse ($N_{\text{node}} = 289$) mesh SGM^h solution is given in Figure 8.56. The SGM^h solution clearly exhibits the “best” local extrema resolution. Figure 8.57 graphs pressure contours and perspective plots of the SGM^h solution. The distribution near the lid corners are clearly detailed, as is the effect of the coarse centroidal region mesh.

The computed nodal distributions of the SGM^h theory $f_2(\mathbf{x})$ directional parameter $RSGM_x$ and $RSGM_y$ are plotted in perspective in Figure 8.58a-b. To fully visualize these data, I have selected the local u (v)-velocity scales as the elevation (z) reference in Tecplot™ (1993). These distributions clearly indicate that the highest level (red) $RSGM_x$ ($RSGM_y$) is computed where the scalar magnitude of u (v) is the highest. Note that $RSGM_x$ and $RSGM_y$ range (1,9) and (1,3) for this meshing.

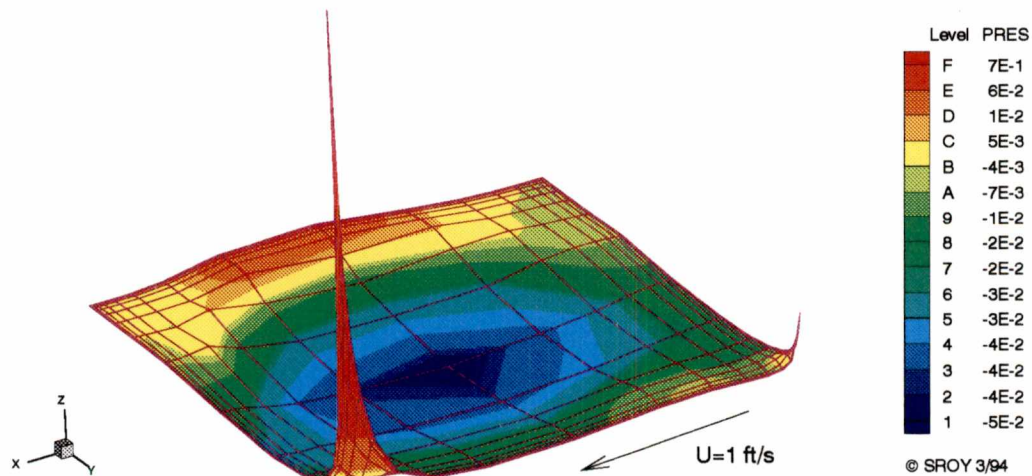
The “global sum” of solution energy is, in general, referred to as the “energy semi-norm”. It is understandable that although this norm converges in the global sense, the local solution energy distribution may be highly erroneous due to dispersion error. Hence, I computed the distributed local solution energy on the element level (DNRM), and the area specific distribution of this local solution energy (DSNR). For a monotone solution, the distribution of the energy should reflect the nature of solution extrema near the lid. Figure 8.59a-b documents the poor distributions of DNRM and DSNR for the GWS^h solution on a 289 node coarse mesh. The comparable GWS^h algorithm solution DNRM and DSNR distributions on the 4 times denser mesh ($N_{\text{node}} = 1089$)



a) GWS^h SOLUTION PRESSURE PERSPECTIVE PLOT, $Re = 3200$, on 289 node mesh.

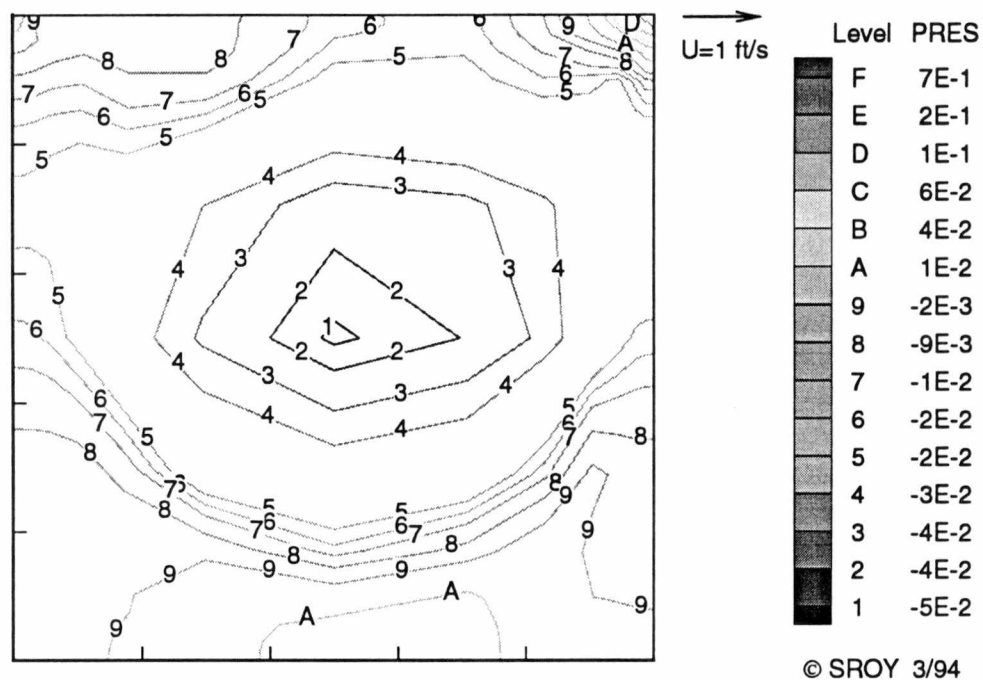


b) GWS^h SOLUTION PRESSURE PERSPECTIVE PLOT, $Re = 3200$, on 1089 node mesh.

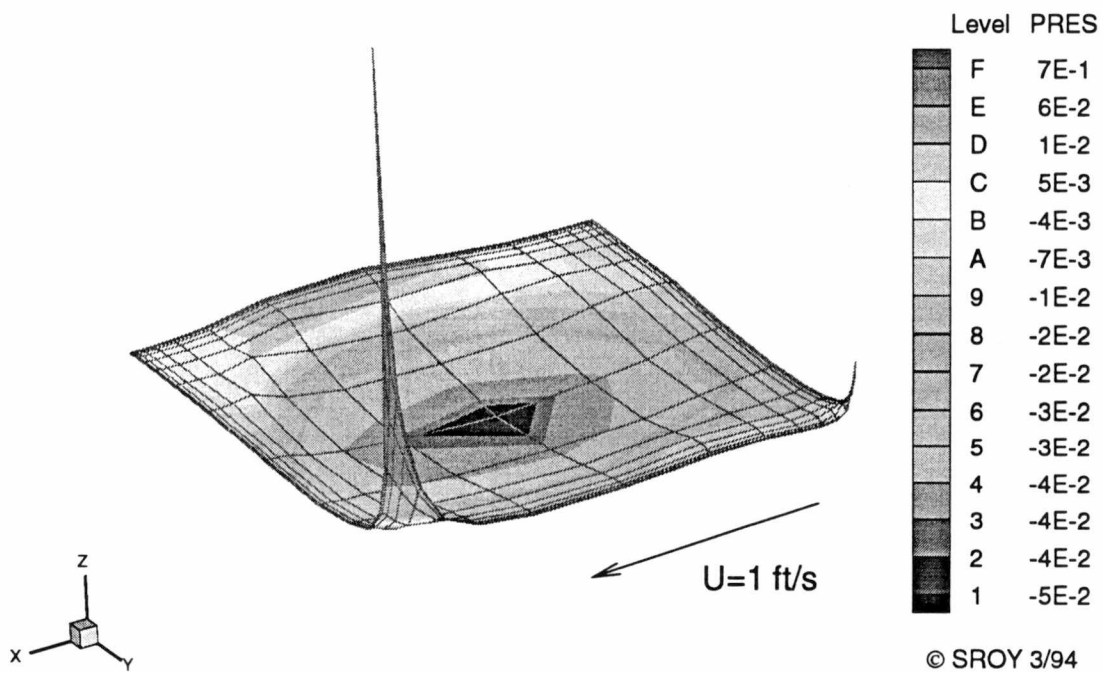


c) SGM^h SOLUTION PRESSURE PERSPECTIVE PLOT, $Re = 3200$, $rsgm_1 \geq 1.0$ on 289 node mesh.

Figure 8.56 GWS^h and SGM^h comparison pressure solutions for driven cavity ($Re = 3200$).

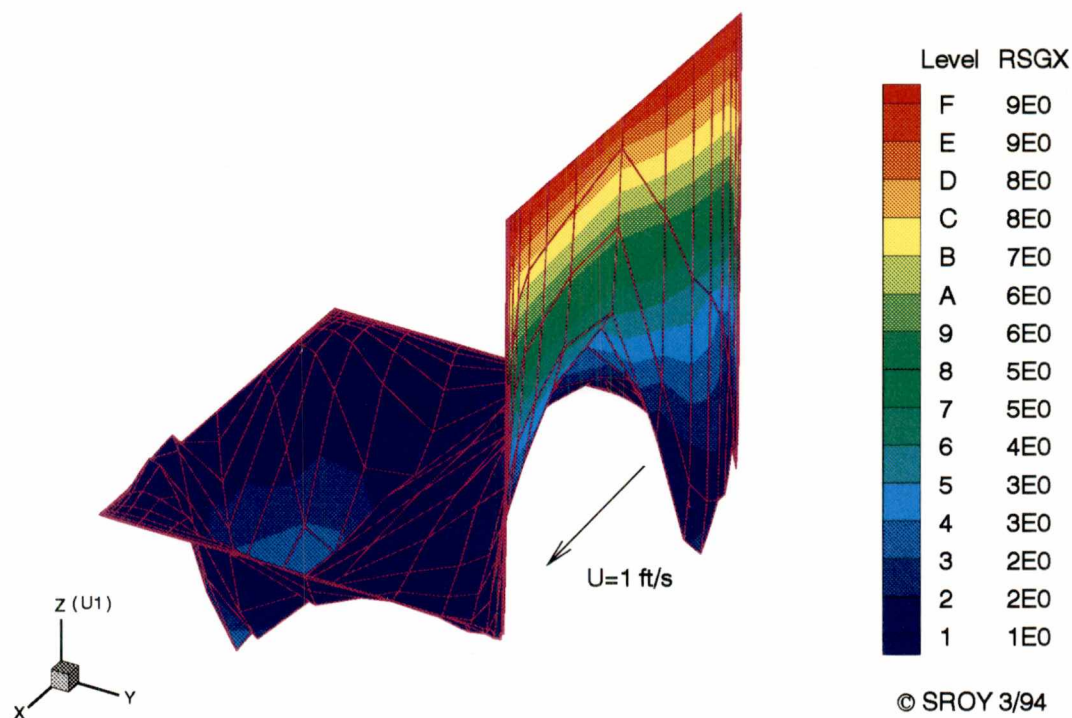


a) SGM^h SOLUTION PRESSURE CONTOURS, $Re = 3200$, $rsgm_i \geq 1.0$, $\beta_i = 0.0$

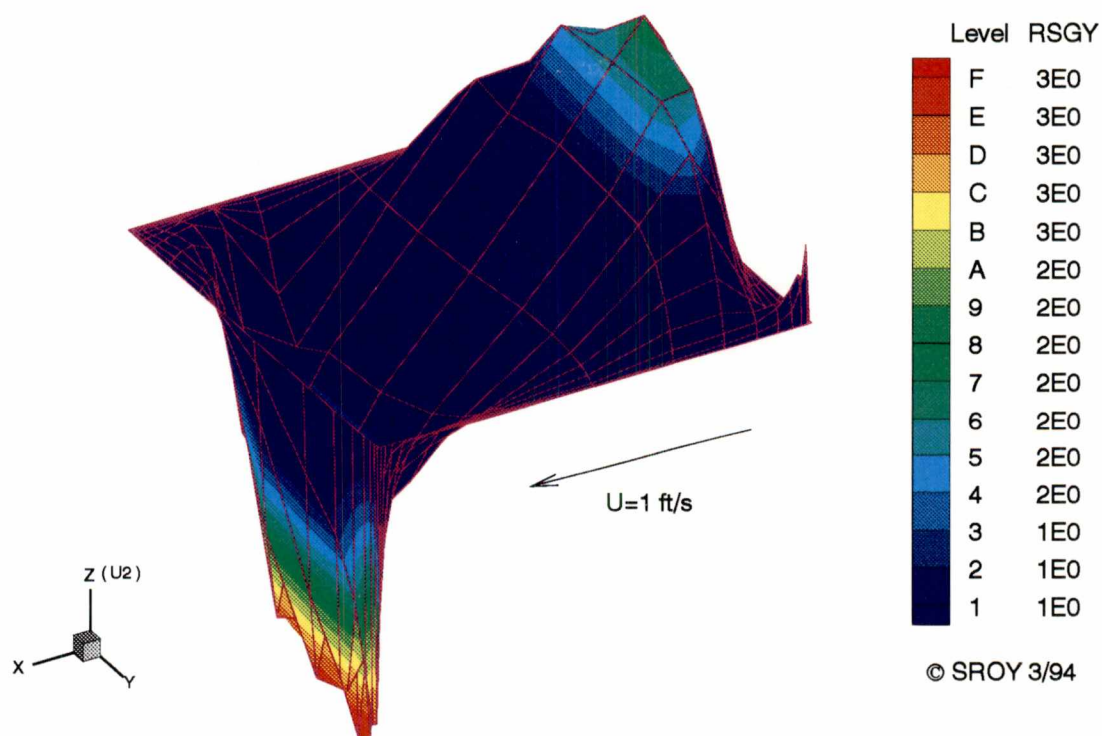


b) SGM^h SOLUTION PRESSURE PERSPECTIVE PLOT, $Re = 3200$, $rsgm_i \geq 1.0$, $\beta_i = 0.0$

Figure 8.57 SGM^h pressure solution for driven cavity ($Re = 3200$) on a $M=16^2$ (289 node) mesh.

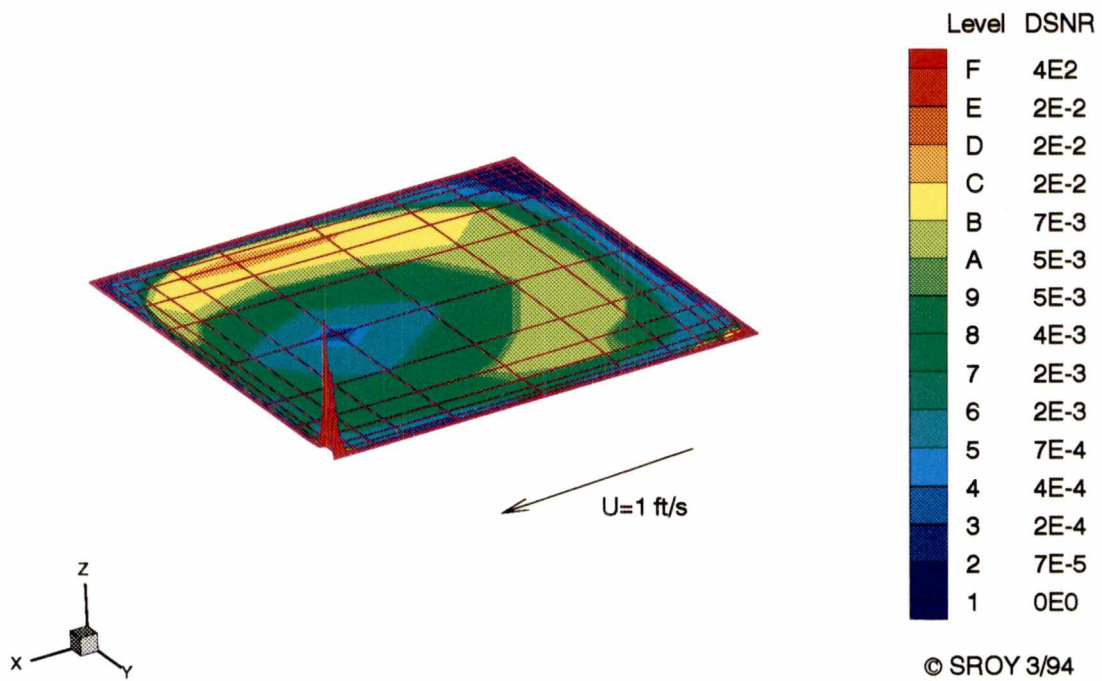


a) $RSGM_x$ PERSPECTIVE VIEW, $Re = 3200$, $rsgm_x \geq 1$.

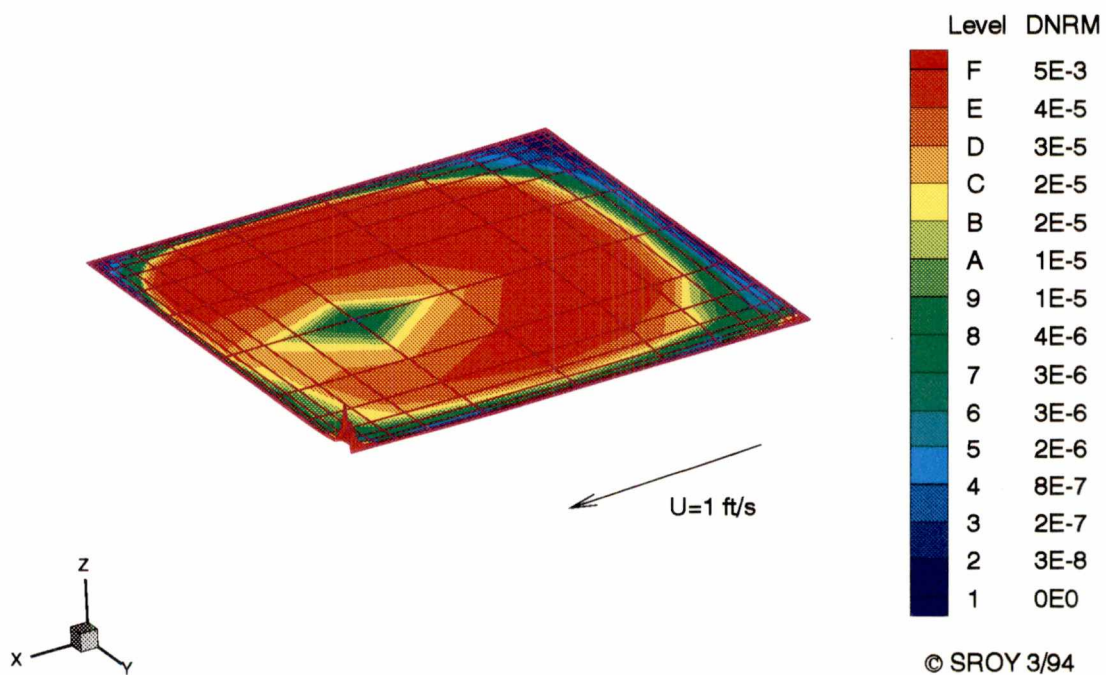


b) $RSGM_y$ PERSPECTIVE VIEW, $Re = 3200$, $rsgm_y \geq 1$.

Figure 8.58 Nodal distribution of SGM^h parameter $RSGM_i$ for driven cavity ($Re = 3200$) on a $M = 16^2$ (289 node) mesh.



a) GWS^h SOLUTION AREA SPECIFIC DISTRIBUTED NORM PERSPECTIVE PLOT



b) GWS^h SOLUTION DISTRIBUTED NORM PERSPECTIVE PLOT

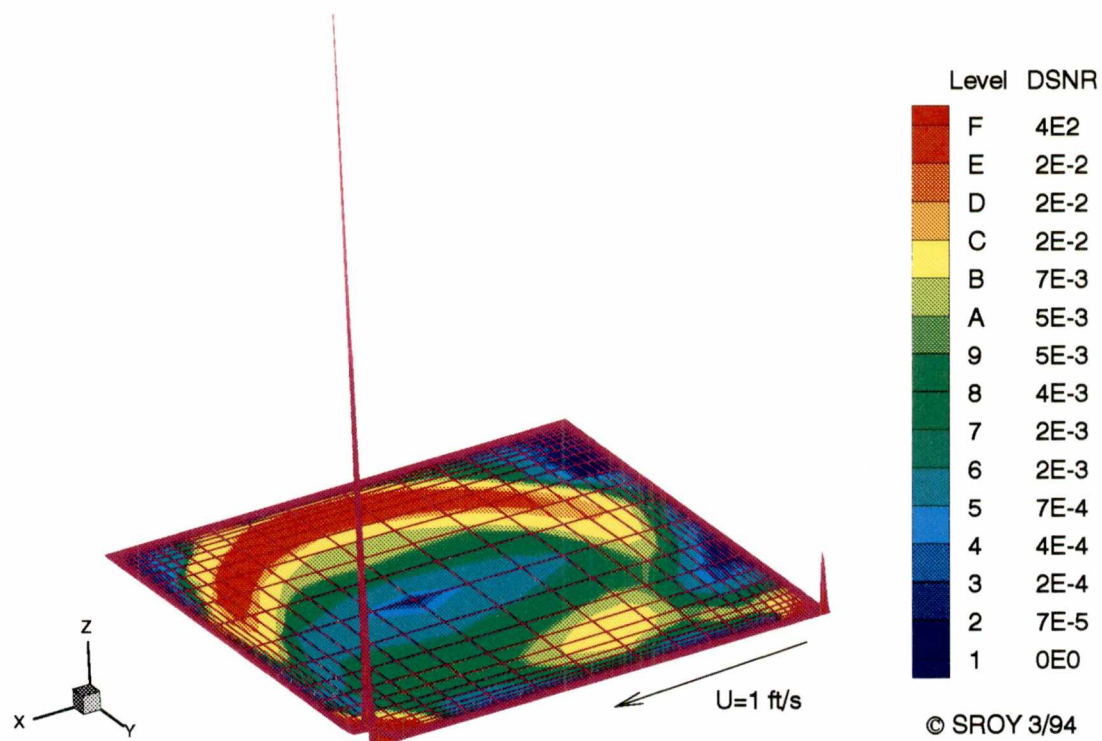
Figure 8.59 GWS^h solution energy norm for driven cavity ($Re = 3200$) on a $M=16^2$ mesh.

are plotted in Figure 8.60a-b. The companion coarse mesh SGM^h solution energy norm distributions, Figure 8.61, clearly show the sharp increase in DNRM near the two local extrema. The area specific DSNR captures the energy distribution on a much better scale, as it is not grid size dependent. The SGM^h solution is clearly comparable to the fine mesh GWS^h solution.

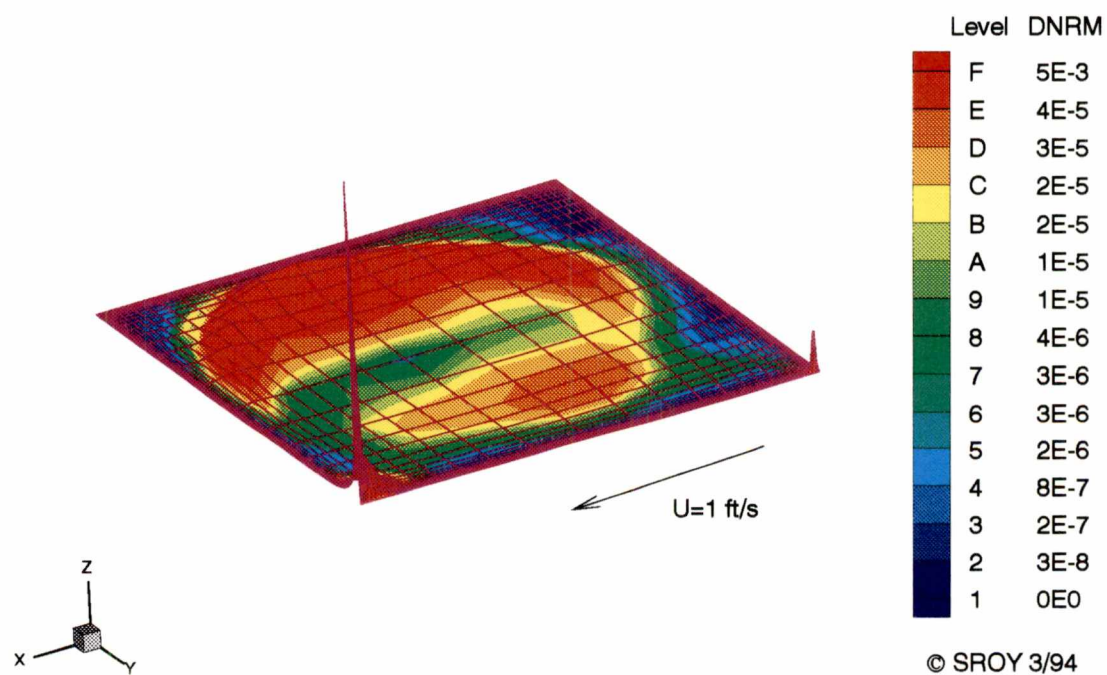
8.4.2 Close-coupled step wall diffuser

This benchmark geometry has been studied and reported widely, *c.f.*, Armaly, *et al.* (1983), Kim and Moin (1985), Guj and Stella (1988), Sohn (1988), Ghia, *et al.* (1989), Thangam and Knight (1990), Gartling (1990), Ikohagi and Shin (1991), Cabuk, *et al.* (1992), Schaub (1992) and Williams (1993). Armaly, *et al.* (1983) report laser-Doppler anemometer measurements of velocity distribution and reattachment lengths downstream of a close-coupled backward facing step mounted in a wide open air-driven flow channel. The data include laminar, transitional and turbulent flow measurements of air for a range of Reynolds number ($70 < Re < 8000$). These experimental data have been used widely by researchers for validation of their computational models.

The aspect ratio of the test section was chosen to ensure that the onset flow was laminar and fully developed, hence essentially one-dimensional, before the step cross-section for all Reynolds numbers. The test section was a 200 mm straight channel approach to the backward facing step and a 500 mm long channel downstream of the step. The crosssection of the channel along with the primary and secondary

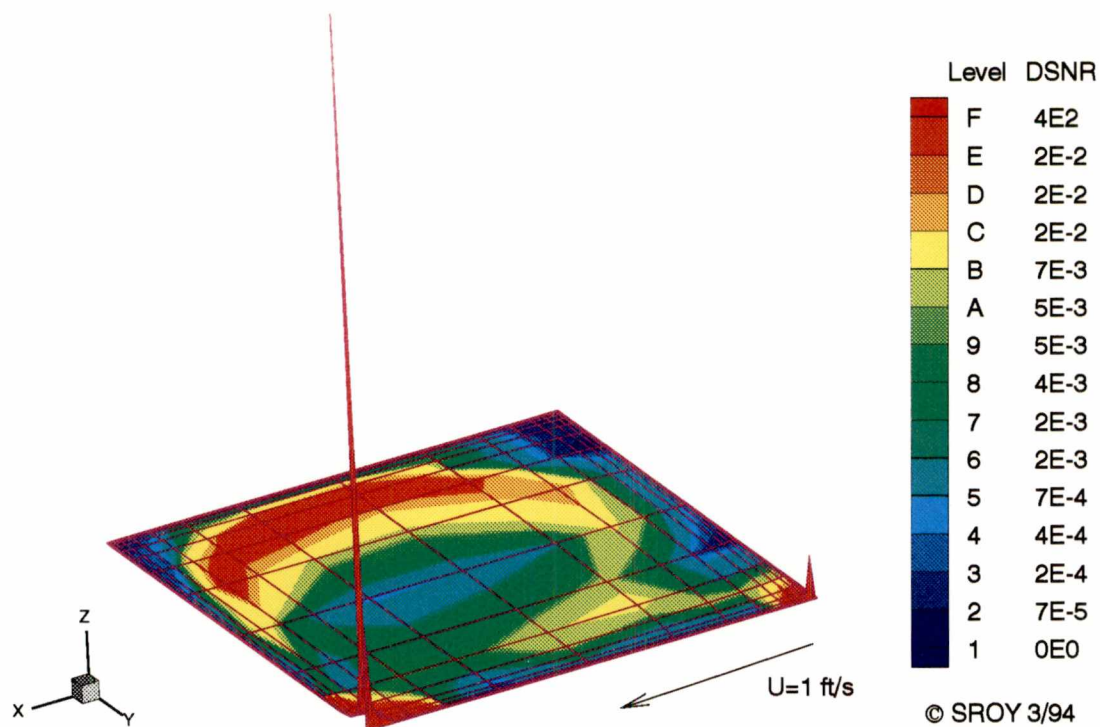


a) GWS^h SOLUTION AREA SPECIFIC DISTRIBUTED NORM PERSPECTIVE PLOT

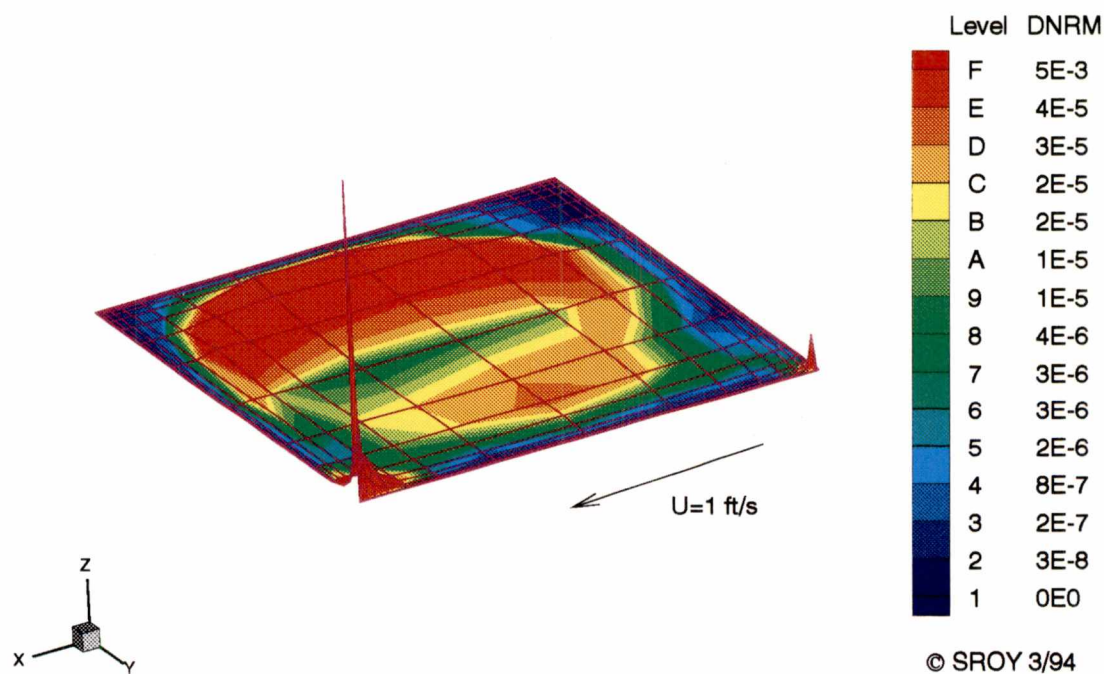


b) GWS^h SOLUTION DISTRIBUTED NORM PERSPECTIVE PLOT

Figure 8.60 GWS^h solution energy norm for driven cavity ($Re = 3200$) on a $M=32^2$ (1089 node) mesh.



a) SGM^h SOLUTION AREA SPECIFIC DISTRIBUTED NORM PERSPECTIVE PLOT



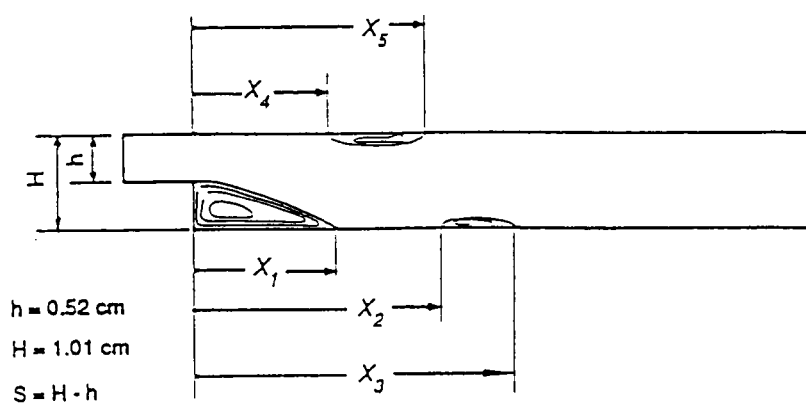
b) SGM^h SOLUTION DISTRIBUTED NORM PERSPECTIVE PLOT

Figure 8.61 SGM^h solution energy norm for driven cavity ($\text{Re} = 3200$) on a $M=16^2$ (289 node) mesh.

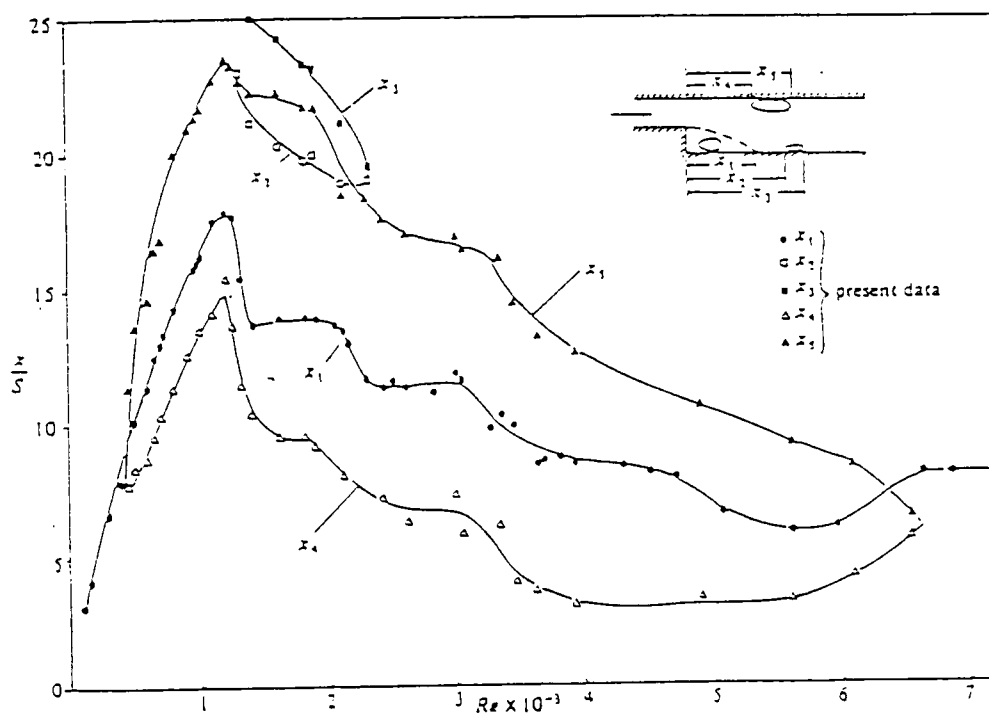
recirculation regions are shown in Figure 8.62a. The upstream channel height was 5.2 mm, and the downstream channel height was 10.1 mm, Figure 8.62a, hence creating a step height $S = 4.9$ mm and an expansion ratio of $H/h = 1.9423$. The width of the channel was 180 mm.

These experimental data confirm that the flow downstream of the step was essentially two-dimensional only for $Re < 400$, and became fully turbulent flow ($Re > 6500$). The reattachment lengths of primary and secondary recirculation regions on the duct centerline were measured by scanning the position of the zero-mean velocity line at the lower and upper wall in the streamwise direction at constant and known elevations. The experimental locations for these regions, as given by Armaly, *et. al* (1983), are reproduced in Figure 8.62b. Here, the reattachment coordinates, normalized by the step height S , are plotted as a function of the Reynolds number based on the hydraulic diameter D_H of the channel. For this channel, $D_H = 2h$, where h is channel inlet height.

This dissertation focus was to compute two dimensional laminar steady-state solutions for $100 \leq Re \leq 800$. Specifically, there is some controversy regarding existence of a steady-state for $Re = 800$, and no 2-D results for $Re > 400$ have agreed with the experimental data for center plane recirculation reattachments. The velocity vectors, pressure, vorticity and streamfunction solutions are computed using the continuity constraint method (CCM, Section 2.4) for two dimensional isothermal flow. I implemented SGM^h (and $p-WS^h$) algorithms (Section 5.10) in the CCM formulation, and validated the SGM^h (and $p-WS^h$) solutions against the standard GWS^h and TWS^h



a) Flowfield eddy locations and reattachment coordinates



b) Location of recirculation reattachments of the flow

Figure 8.62 Flow field details for backward facing step, Armaly *et al.* (1983).

algorithm solutions. Other results reported in the literature are also compared. The graded finite element mesh details for these computational experiments are plotted in Figure 8.63. The inflow duct is 20 mm long and is divided into 6×17 bilinear elements. The downstream channel is 160 mm long and is divided into 47×35 bilinear elements.

The prescribed fully-developed laminar inlet U velocity profile was parabolic and the V inlet velocity was held at zero. No-slip boundary conditions were enforced on all walls. The pressure and the continuity constraint function Φ (see Section 2.4) were both fixed at zero at the outflow boundary. The outlet velocity boundary condition is homogeneous Neumann. Hence, in this solution process, the y -momentum equation ($i = 2$ in (2.14) and (2.36)) produces data that are less significant (smaller numbers around zero) than the x -momentum ($i = 1$) counterpart. Schaub (1992) and Williams (1993) found that the CCM solution process for GWS^h and TWS^h algorithms lost stability robustness for $Re \geq 400$, hence they rotated their solution domain by an angle (30°) to promote equal significance in the U and V solution data. For this dissertation project, I have used an unrotated domain to study the relative robustness and stability of the SGM^h algorithm for the CCM formulation.

Figure 8.64 documents the resultant benchmark velocity vector distributions for $100 \leq Re \leq 800$ using the standard GWS^h algorithm. The arrows adjacent to duct walls in Figure 8.64 indicate the primary and secondary recirculation reattachment coordinates as reported by Armaly, *et al.* (1983). The solution for primary reattachment length for the lower Reynolds numbers ($100 \leq Re \leq 400$), Figure 8.64a-c,

2-D Finite element mesh, $M = 6 \times 17 + 48 \times 34$, $N_{node} = 7 \times 18 + 48 \times 35$

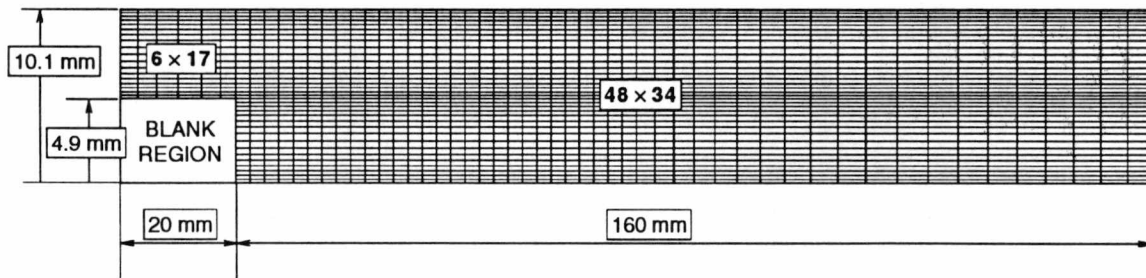
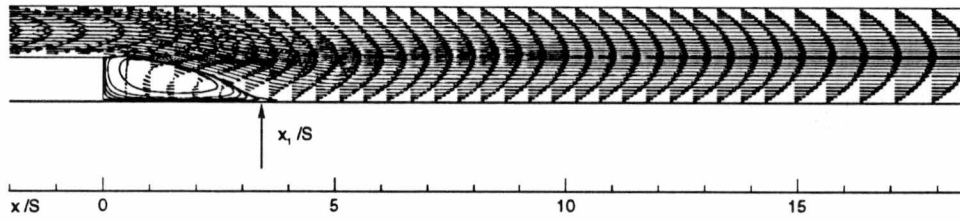


Figure 8.63 Computational domain details for the 2-D backward facing step.

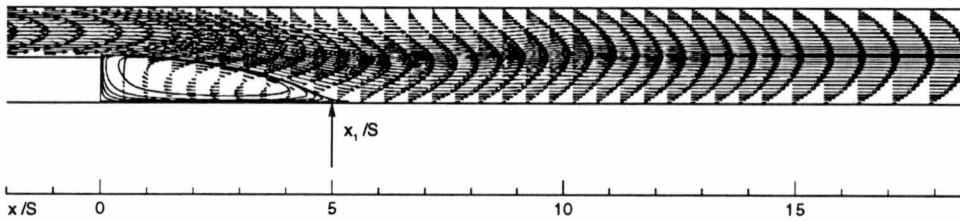
agrees exactly with the experimental data. However, agreement is poor for $Re > 400$ as solutions becomes periodic and not steady in nature, Gartling (1993).

In both two and three dimensions, dispersion error control plays an important role for maintaining solution process stability for $Re > 400$. Ghia, *et al.* (1989) conjecture that the “abrupt change” in the flow structure from two to three dimensional flow is due to a Taylor-Görtler vortex instability that causes a spanwise-periodic counter-rotating pairs of vortices with axes aligned with main flowstream direction. As the Reynolds number increases, the 2-D solution becomes very sensitive to the initial flow conditions requiring a Re continuation procedure, *i.e.*, using lower Re solution as the initial condition. The resultant instability in the flowfield can be observed near

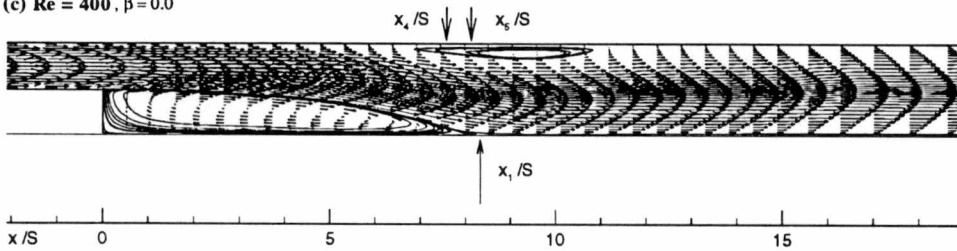
(a) $Re = 100$



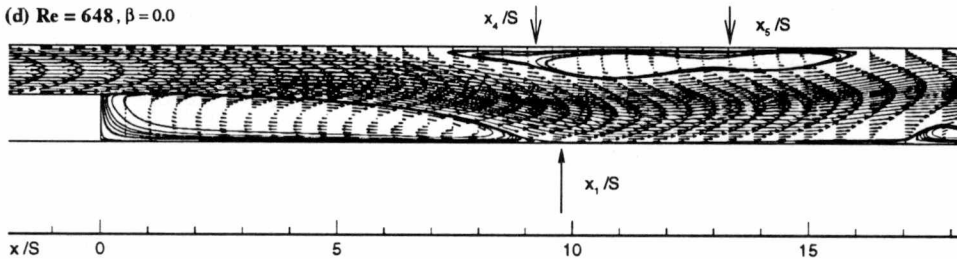
(b) $Re = 200$



(c) $Re = 400, \beta = 0.0$



(d) $Re = 648, \beta = 0.0$



(e) $Re = 800, \beta = 0.0$

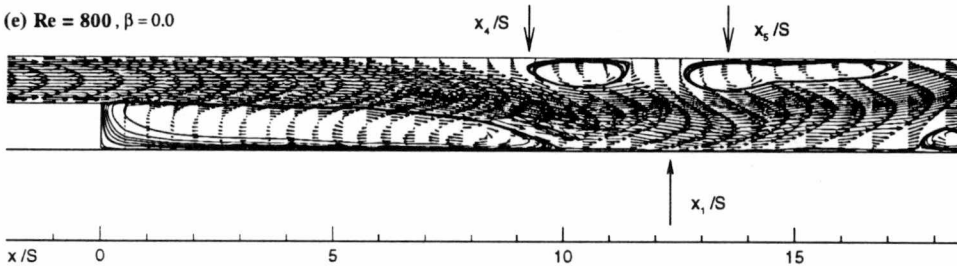


Figure 8.64 GWS^h solutions for 2-D backward facing step ($100 \leq Re \leq 800$) on a non-uniform $7 \times 18 + 48 \times 35$ node mesh (arrows indicate reattachment coordinates, Armaly, et. al (1983)).

the top wall where the secondary vortex structure (x_4 to x_5 in Figure 8.62a) separates in a cluster of bubbles, Figure 8.64c-e. Finally, at $Re = 800$, the GWS^h solution becomes dispersive with an unsteady separated bubble cluster at the secondary vortex region, Figure 8.64e. Formation of downstream complementary recirculation regions, x_2 - x_3 in Figure 8.62a, is also noticeable.

Williams (1993) observed that the “dispersion error manifests itself most significantly in the pressure solution. The visual inspection of the corresponding velocity solutions showed no readily discernible differences.” GWS^h pressure solutions for $100 \leq Re \leq 800$, Figure 8.65, confirm that observation. The pressure solutions for $Re \leq 400$, Figure 8.65a-b, are smooth and monotone. For $Re = 400$ and 648, the pressure solution contour wiggles confirm the presence of localized dispersion error near the step, Figure 8.65c-d. The GWS^h pressure solution becomes totally dispersive error dominated throughout the entire channel (especially near the primary reattachment point x_1/S) at $Re = 800$, Figure 8.65e.

Since $Re = 800$ is clearly the case of the worst dispersion error for the range of laminar flow Reynolds numbers, the application of SGM^h in controlling the dispersion error, hence improving solution quality, would be an excellent validation of the algorithm. Figure 8.66 presents the comparison of pressure solutions obtained for $Re = 800$ via the GWS^h , two TWS^h , a quadratic p - WS^h and the SGM^h algorithms. The GWS^h solution, Figure 8.66a, is obviously the most dispersed solution. For $\beta > 0$, TWS^h can smooth out to an extent these solution oscillations, Figure 8.66b-c, at the

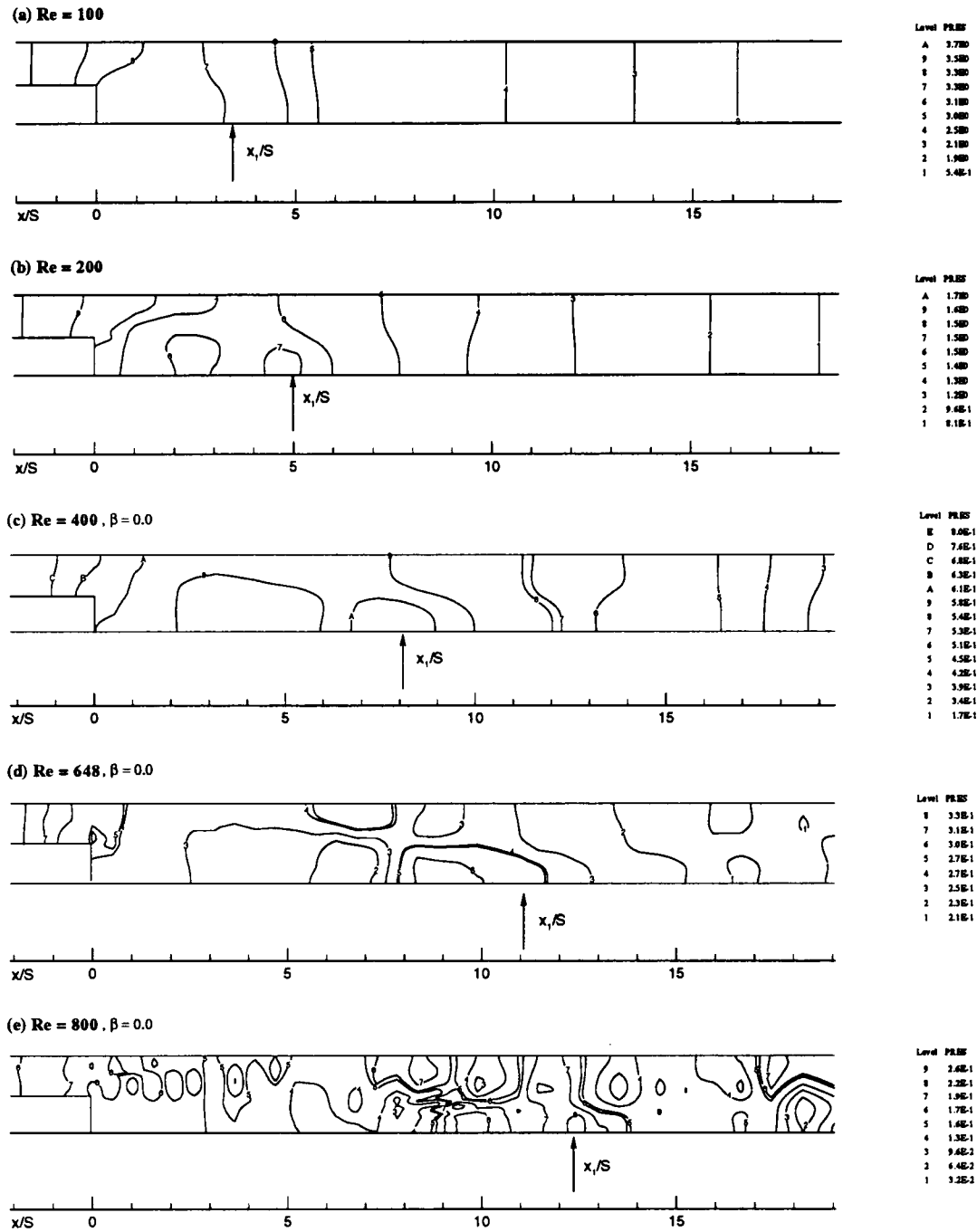


Figure 8.65 GWS^h pressure solution for 2-D backward facing step ($100 \leq Re \leq 800$) on a $7 \times 18 + 48 \times 35$ node non-uniform mesh.

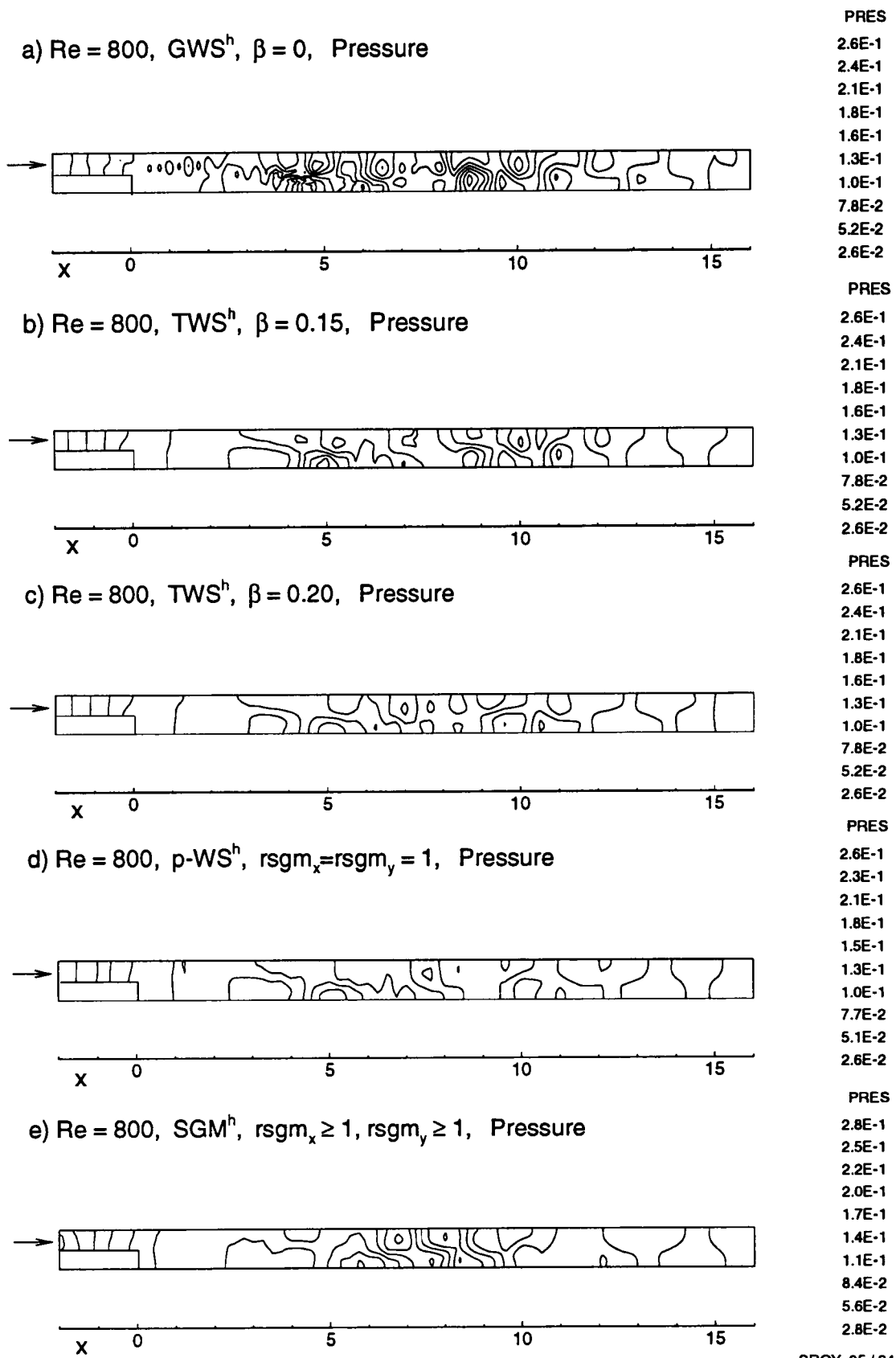


Figure 8.66 Comparison pressure solution contours for 2-D backward facing step at $Re = 800$.

expense of added artificial diffusion, thus lowering the Reynolds number implicitly. The comparison quadratic p -WS^h ($c_x = c_y = 1 = \text{rsgm}_x = \text{rsgm}_y$ in (5.76)) pressure solution, Figure 8.66d, shows the relatively more dispersion-free nature of this algorithm solution, without explicitly adding artificial diffusion. Finally, the SGM^h solution shows a truly monotone non-dispersive pressure solution, Figure 8.66e. The corresponding nodal distribution of subgrid embedding parameters ($\text{rsgm}_x, \text{rsgm}_y \geq 1$) were computed through the solution iteration process via (5.74). Further, the SGM^h solution did achieve a genuine steady-state, as a consequence of iterating the CCM algorithm solution as a steady state problem.

Figure 8.67 compares pressure solution distributions on the geometric centerline ($y = 0$) of the channel for $\text{Re} = 800$. These data clearly show the presence of rampant $2\Delta x$ oscillations in the GWS^h solution, Figure 8.67a. Using artificial dissipation ($\beta = 0.15$), in TWS^h, reduces most of these small scale oscillations to a considerable extent, Figure 8.67b. A higher level of $\beta = 0.20$ essentially over-diffuses the solution, Figure 8.67c. The quadratic p -WS^h solution, Figure 8.67d, shows an apparently dissipated distribution free of local extrema. Conversely, the SGM^h solution, Figure 8.67e, appears almost monotone with local extrema, hence appears quite different from the other four WS^h solutions.

The V velocity component solution isovels, for $\text{Re} = 800$, corresponding to GWS^h, TWS^h, p -WS^h and SGM^h algorithms are plotted in Figure 8.68. Again, the dispersive nature of the GWS^h solution is clearly evident, Figure 8.68a, while the TWS^h solutions, Figure 8.68b-c, confirm a dissipated but locally non-monotone V solution

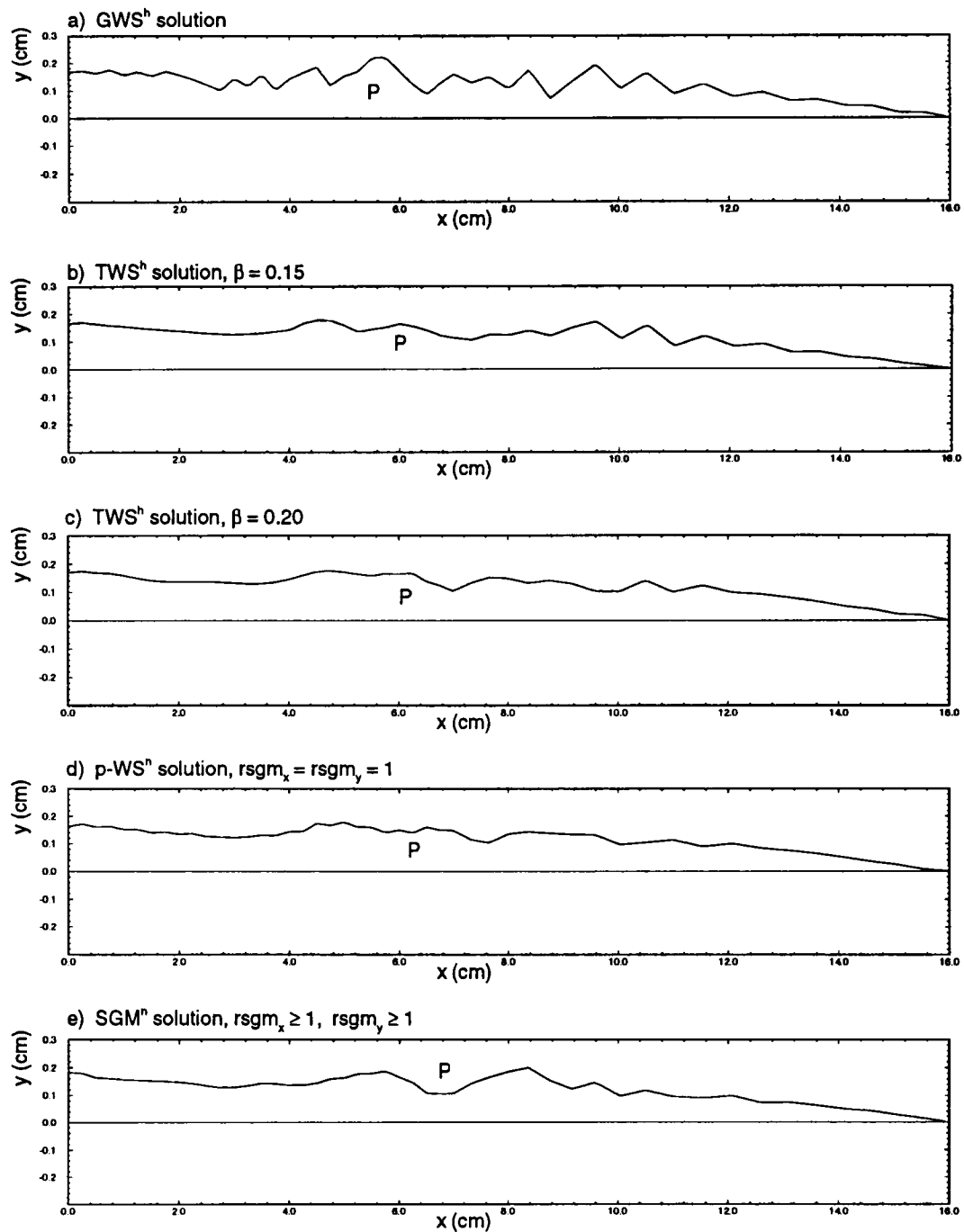


Figure 8.67 Comparison pressure solution through the geometric centerline ($y = 0$) for backward facing step, $Re = 800$.

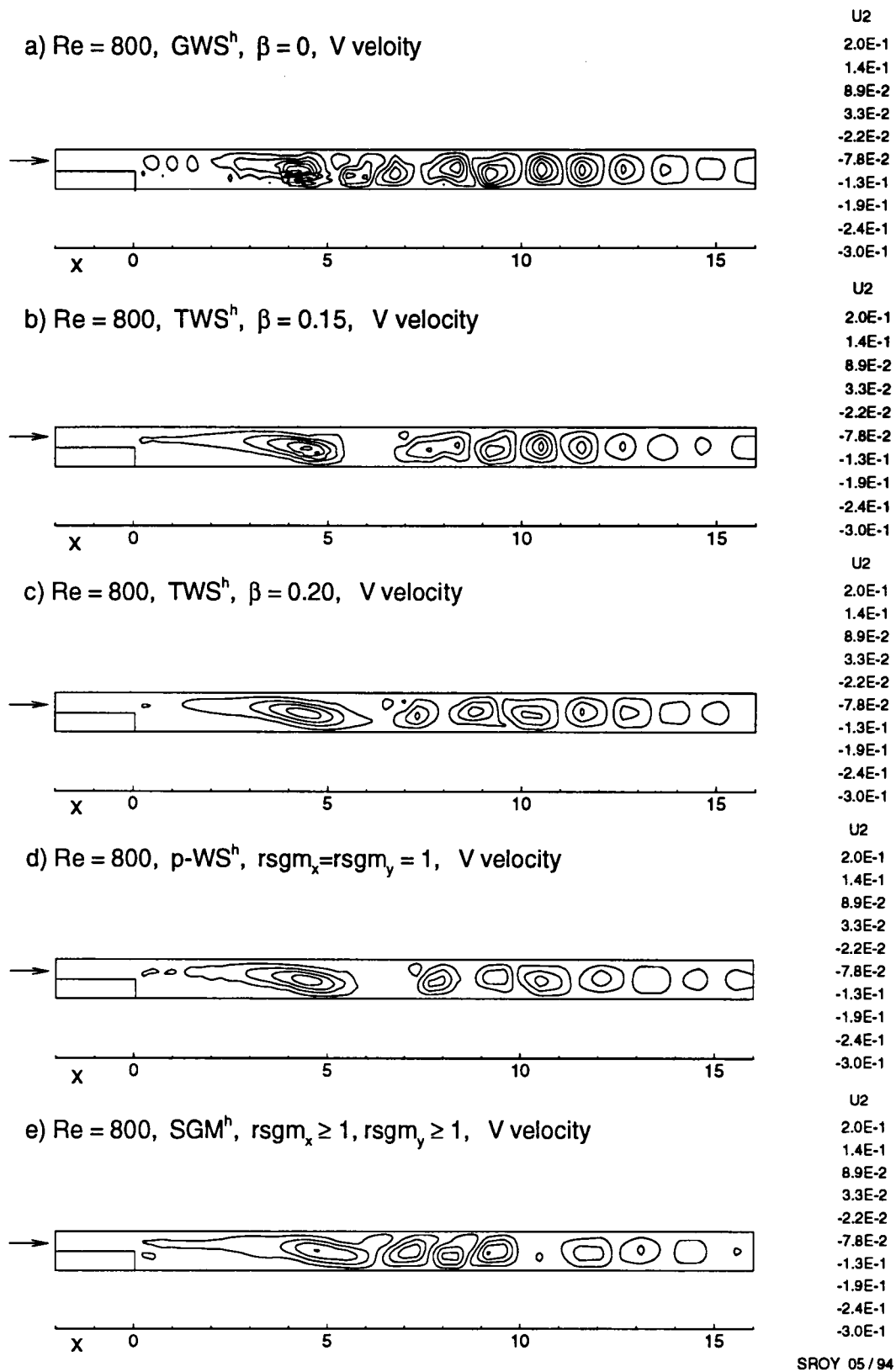


Figure 8.68 Comparison V velocity solution contours for 2-D backward facing step at $Re = 800$.

distribution. The comparison p -WS^h and SGM^h solution contours, Figure 8.68d-e, document distinct improvements in the local monotonicity of the V solution. The V solution distribution near the geometric step is least dispersive in SGM^h solution contours, Figure 8.68e.

A benchmark comparison of SGM^h solution primary recirculation zone reattachment length with other results from the literature, is plotted on the scale of Reynolds number based on the channel hydraulic diameter, Figure 8.69. The comparison

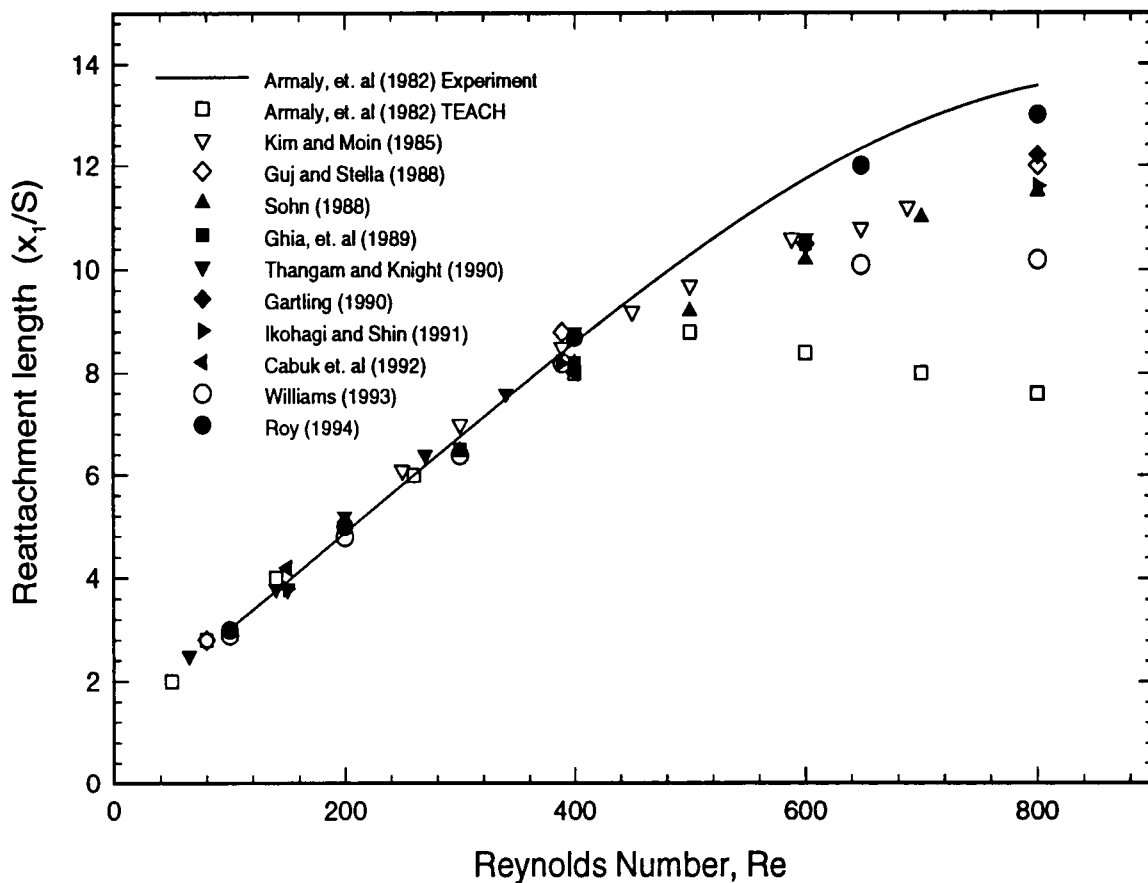


Figure 8.69 Comparison 2-D backward facing step solutions for backward facing step, reattachment length vs. Reynolds number.

unequivocally establishes SGM^h computed solution in closest agreement with the experimental data, Armaly, *et. al* (1983). The velocity vectors at $Re = 800$, computed from various comparison algorithms, are plotted on an x -scale normalized by the step height S , Figure 8.70a-e. The arrows labelled x_1/S , x_4/S , and x_5/S indicate the measured primary and secondary reattachment locations from Armaly, *et. al* (1983). The comparison confirms that the GWS^h , TWS^h $\beta = 0.15$, TWS^h $\beta = 0.20$, and $p-WS^h$ algorithm solution errors are respectively $\sim 20\%$, $\sim 19\%$, $\sim 16\%$ and $\sim 14\%$, Figure 8.70a-d, for primary reattachment, with essentially no agreement for secondary reattachment. The SGM^h computed solution primary reattachment length (x_1/S), Figure 8.70e, is only $\sim 2\%$ in error for primary reattachment. The secondary recirculation reattachment length is within 95% agreement with data, however the x_4 intercept is displaced $0.15(x_1/S)$ downstream.

The distribution of the V velocity solution along the geometric centerline ($y = 0$) is another indicator of solution monotonicity, Figure 8.71. The oscillatory behavior of GWS^h solution, Figure 8.71a, is dissipated in the TWS^h $\beta = 0.15$, Figure 8.71b, and $\beta = 0.20$, Figure 8.71c, solutions. The quadratic $p-WS^h$ solution, Figure 8.71d, appears rather similar to $\beta = 0.20$ TWS^h with a trace of low level local "wiggles". The locally monotone SGM^h solution, Figure 8.71e, confirms a periodicity noticeably different than the GWS^h , TWS^h and $p-WS^h$ solutions. In particular, the SGM^h solution primary recirculation coordinates are $\sim 12\%$ more accurate than the comparison solutions, Figure 8.69.

Comparison $Re = 800$ vorticity solution distributions on the geometric centerline

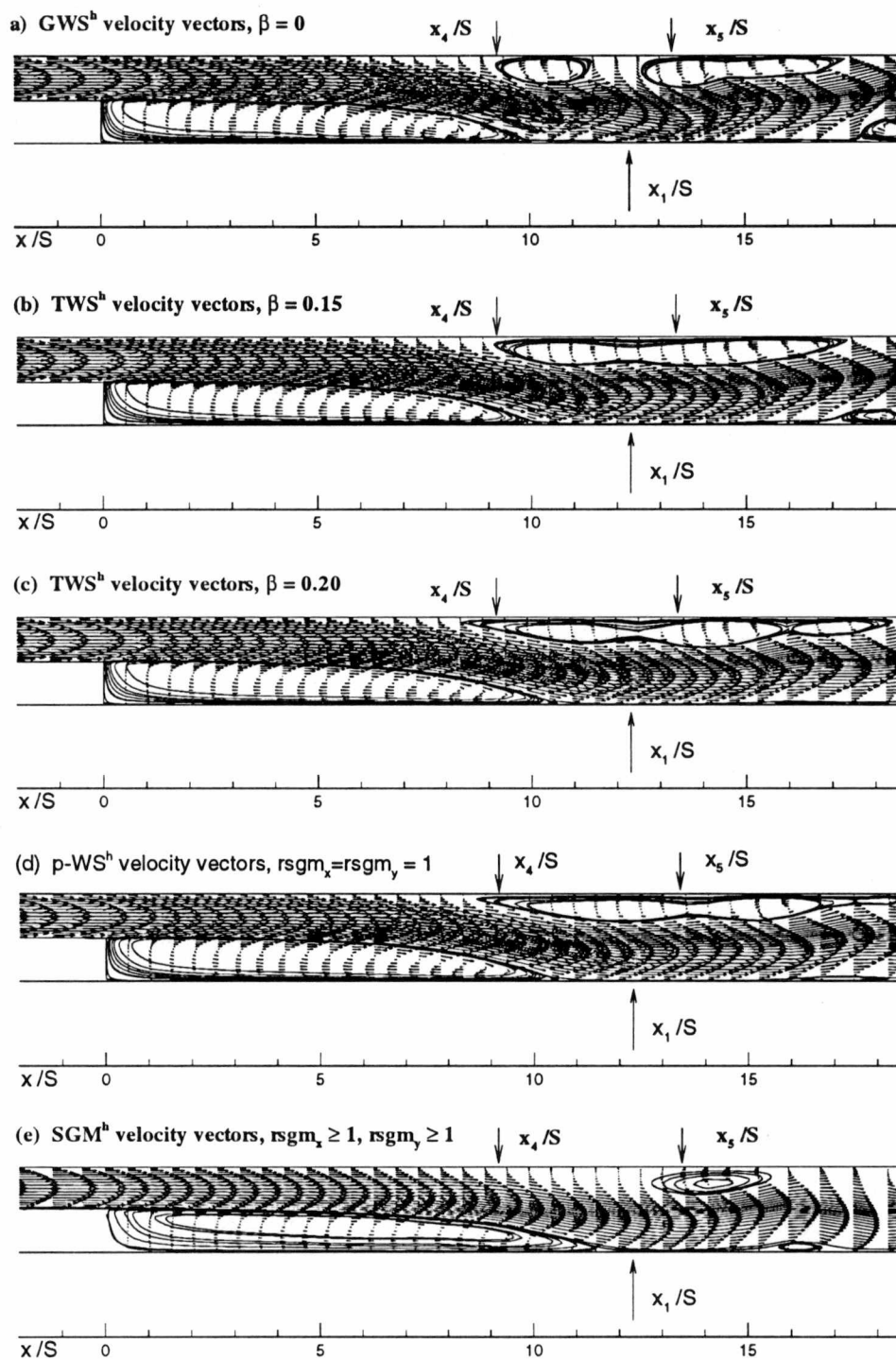


Figure 8.70 Comparison velocity vectors for 2-D backward facing step ($Re = 800$) on a non-uniform $7 \times 18 + 48 \times 35$ node mesh, arrows indicate measured reattachment lengths from Armaly, et. al (1983).

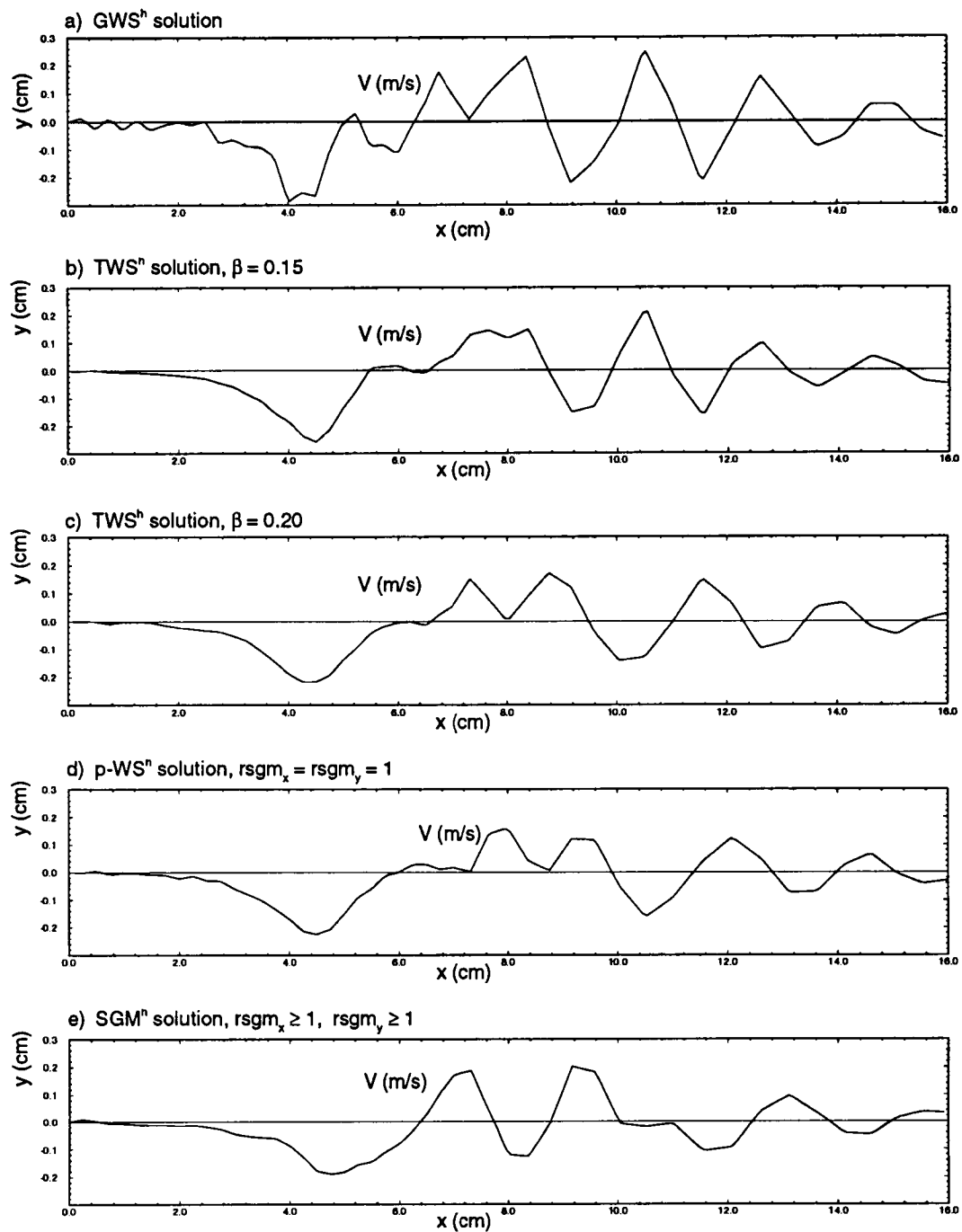


Figure 8.71 Comparison V velocity through the geometric centerline ($y = 0$) for backward facing step, $Re = 800$.

($y = 0$), computed via (2.30), are shown in Figure 8.72a-e. The GWS^h vorticity solution is quite dispersive throughout the solution domain, Figure 8.72a. Companion TWS^h solutions, though less oscillatory, are not free of a $2\Delta x$ error wave near the geometric singularity (0,0) location, Figure 8.72b-c. In fact, these solutions are diffused by the use of artificial dissipation ($\beta = 0.15$ and $=0.20$) near the near corner singularity. The $p-WS^h$ solution, though appearing non-oscillatory, Figure 8.72d, also does not clearly define the vorticity in the singularity region. The SGM^h algorithm solution, for nodally distributed $rsgm_x \geq 1$ and $rsgm_y \geq 1$, yields a non-dissipative, monotone solution, Figure 8.72e, and the step corner vorticity singularity is best predicted among the various methods.

The streamfunction distributions are computed via (2.32) and the corresponding solution distributions along the geometric centerline ($y = 0$) are plotted in Figure 8.73. Although streamfunction is an extremely well behaved variable, even for high Reynolds number, here again, for GWS^h the distribution shows both small scale and large scale oscillations through out the solution field, Figure 8.73a. Using artificial diffusion ($\beta = 0.15$ and $= 0.20$) reduces these solution oscillations, Figure 8.73b-c. The quadratic $p-WS^h$ and SGM^h solutions, Figure 8.73d-e, both show non-dispersive streamfunction distributions along the centerline; however, the SGM^h solution periodicity again appears totally distinct from the other solutions.

A direct comparison between the GWS^h and SGM^h U and V velocity component solutions ($Re = 800$) completes this planar study, Figure 8.74-8.75. Figure 8.74a plots the GWS^h U velocity component distribution at several stations $x = \text{constant}$. The

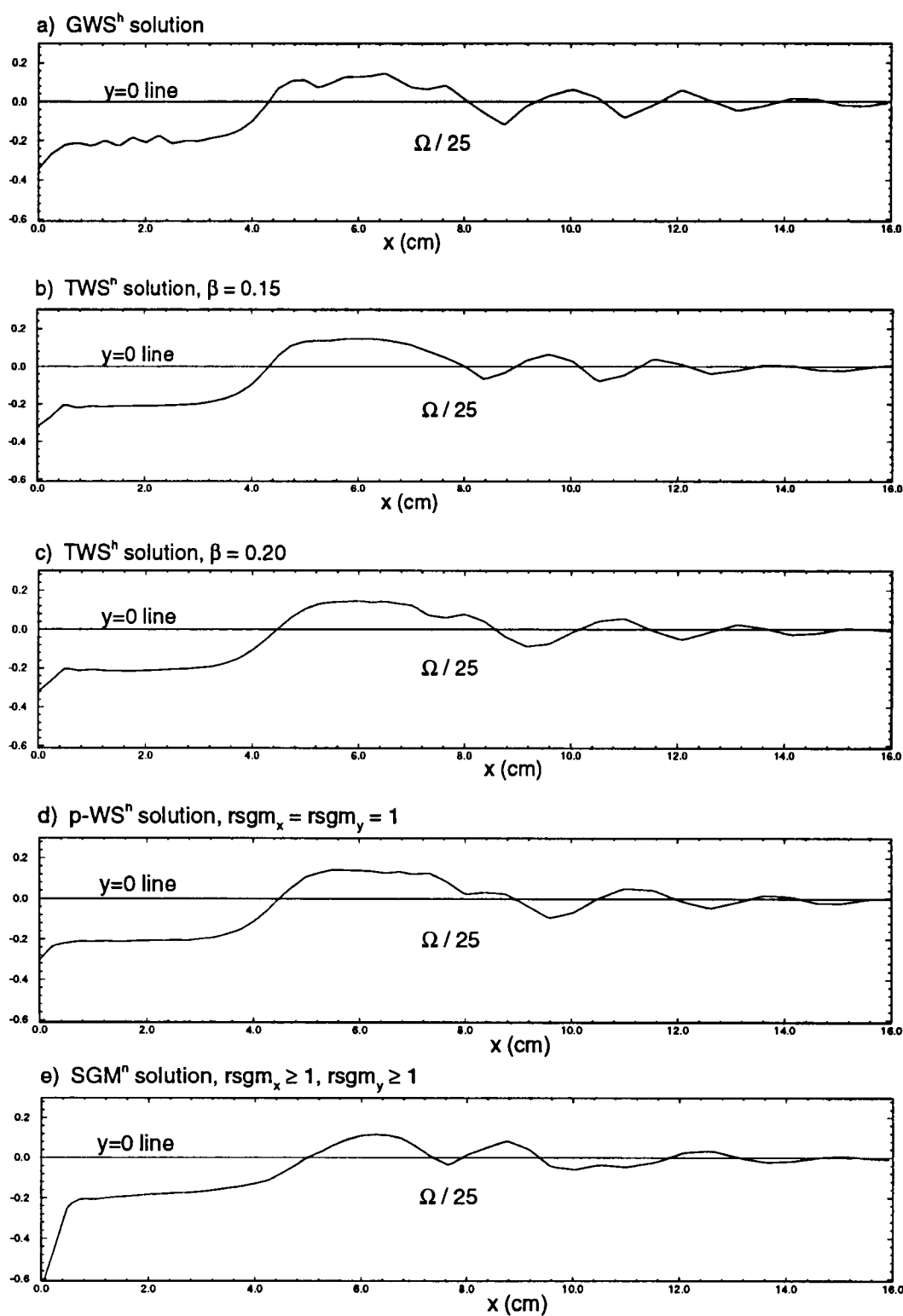


Figure 8.72 Comparison vorticity distribution through the geometric centerline ($y = 0$) for backward facing step, $Re = 800$.

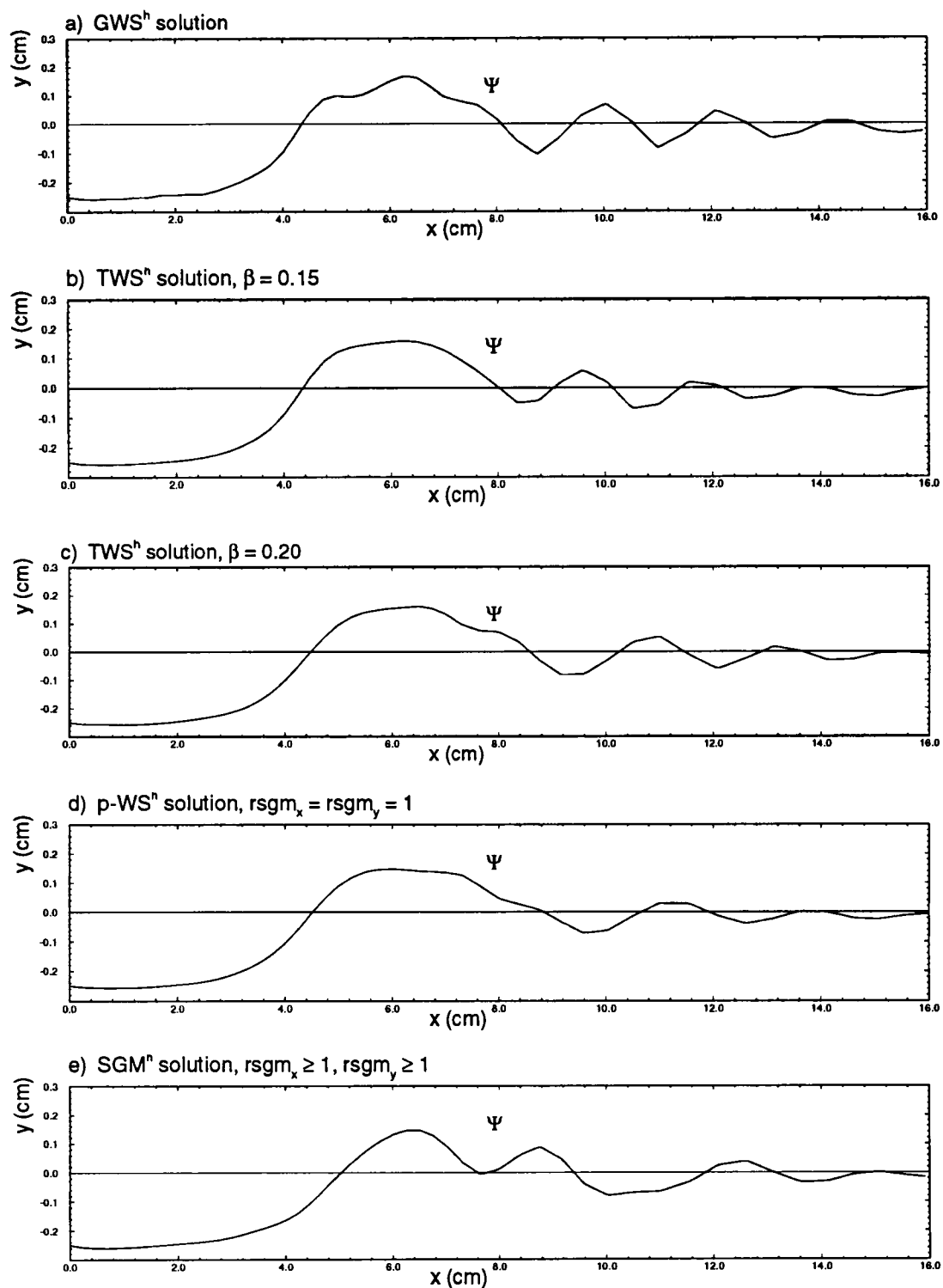


Figure 8.73 Comparison streamfunction distribution through the geometric centerline ($y = 0$) for backward facing step, $Re = 800$.

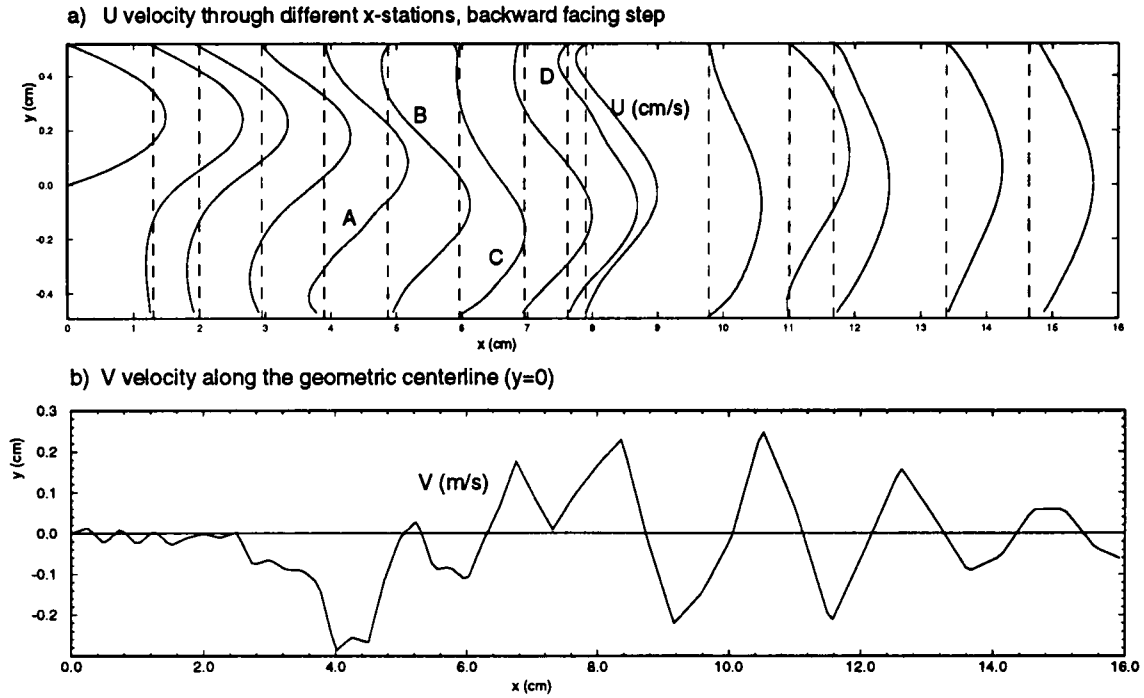


Figure 8.74 GWS^h planar velocity details for backward facing step, $Re = 800$.

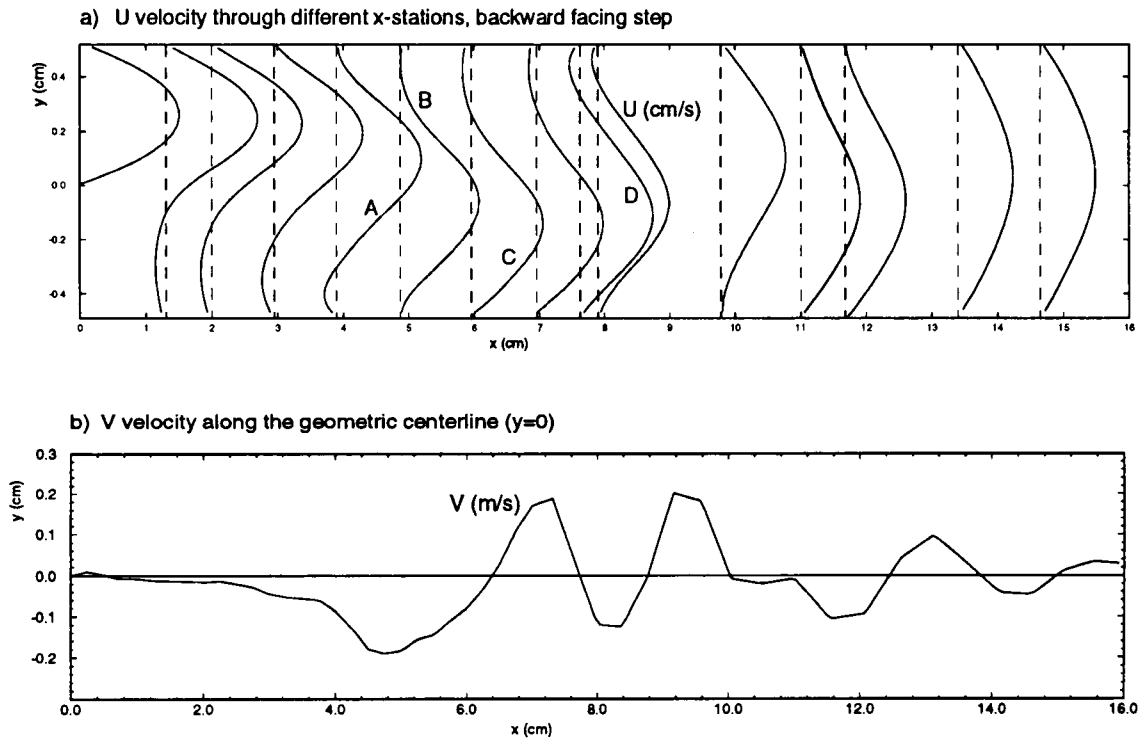


Figure 8.75 SGM^h planar velocity details for backward facing step, $Re = 800$.

inlet parabolic fully developed profile slowly changes its maximum locus as the flow turns downward, hits the floor and then returns towards the geometric centerline ($y = 0$). The four x -planes labelled, A, B, C, and D clearly show the U velocity undulations in the primary and secondary recirculation regions. Again, the V velocity component along the geometric centerline is dominantly dispersive, Figure 8.74b.

Figure 8.75a-b compares SGM^h U and V velocity solution distributions at the same x -stations. The U velocity distributions at all stations, including A, B, C, and D are very smooth, Figure 8.75a. The most noticeable difference, however, is in the V velocity component distribution again showing a $2\Delta x$ error-free, non-diffusive, locally monotone solution characteristic.

The true nature of these solution distribution distinctions can be best realized using a three dimensional perspective plotting presentation. The comparison U velocity perspective plots for GWS^h, TWS^h and p -WS^h algorithms, Figure 8.76, clearly show the $2\Delta x$ error mode as “color diamonds” in the blue (-0.15 - 0.016), yellow (1.0 - 1.2) and lemon yellow (0.84 - 1.0) ranges. The GWS^h solution for $Re = 800$ is clearly the most dispersive, while the two TWS^h solutions show selective improvement in this error mode. The larger $\beta = 0.20$ TWS^h solution, Figure 8.76c, does dissipate essentially all local small scale error waves. The quadratic p -WS^h ($rsgm_x = 1 = rsgm_y$) solution, Figure 8.76d, is artificial dissipation free, showing increasing length of yellow level, while evidencing modest non-monotonicity in the orange-red diamonds. The new artificial diffusion-free SGM^h ($rsgm_x, rsgm_y \geq 1$) algorithm solution, Figure 8.76e, shows no dispersion error diamonds, *i.e.*, the solution is genuinely

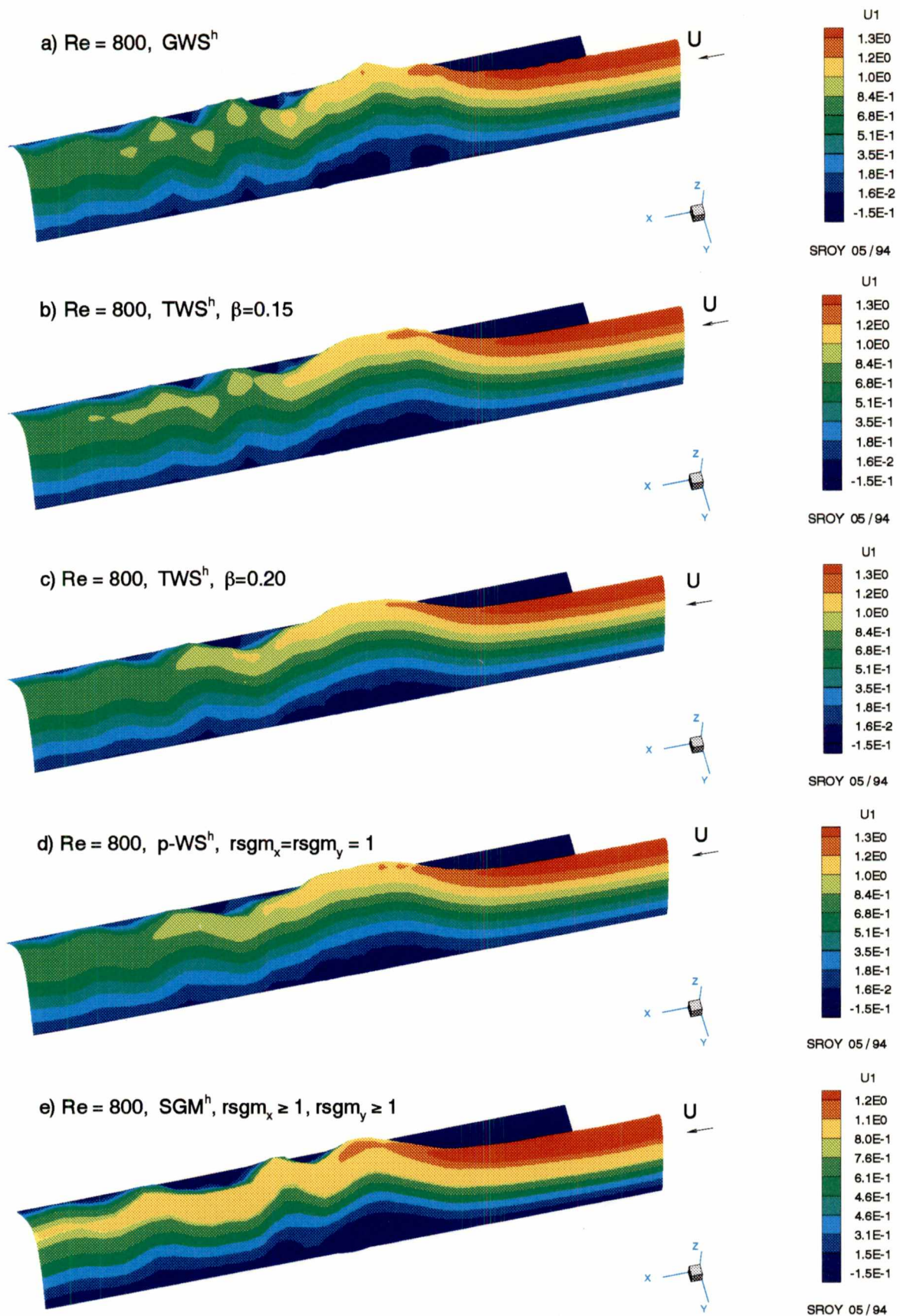


Figure 8.76 Comparison U velocity solution perspective view for $Re = 800$.

monotone. As a consequence, the ranges of red (1.2 - 1.3) and yellow (0.84 - 1.0) levels have fully extended into the solution domain, producing the vital agreement with primary recirculation data.

The companion V velocity comparison solutions with mesh overlays are presented in Figure 8.77. The red (0.12 - 0.09) and yellow (0.09 - 0.04) regions are dominant for all solutions with the three peaks of the SGM^h solution related to the detachment and reattachment locations of recirculation regions. The extra peaks in the other solutions are due to the dispersion error mode. The GWS^h solution, Figure 8.77a, shows the most dispersive peaks and troughs, as expected, while the TWS^h $\beta = 0.15$ solution shows retainment of all error undulations although the extrema have reduced, Figure 8.77b. Using $\beta = 0.20$ in TWS^h , Figure 8.77c, or $p\text{-}WS^h$ ($rsgm_x = rsgm_y = 1$), Figure 8.77d, does not make the solution domain free of short wavelength error waves. However, the SGM^h solution drastically alters the error distribution mode, while clearly retaining the three necessary peaks at detachment and reattachment locations.

Perspective presentations of the comparative pressure solutions are plotted in Figure 8.78a-e. Clearly, the GWS^h pressure solution is full of oscillations and wiggles, Figure 8.78a. The short wavelength $2\Delta x$ structures are evident near the step corner and at approximately one-third and two-third of the downstream distances. Use of artificial diffusion, $\beta = 0.15$ and $= 0.20$ in TWS^h , gradually reduces the error mode amplitude, Figure 8.78b-c, but the dispersion error mode always remains, as visible in green and yellow “diamonds” near the mid-length of the channel. The quadratic

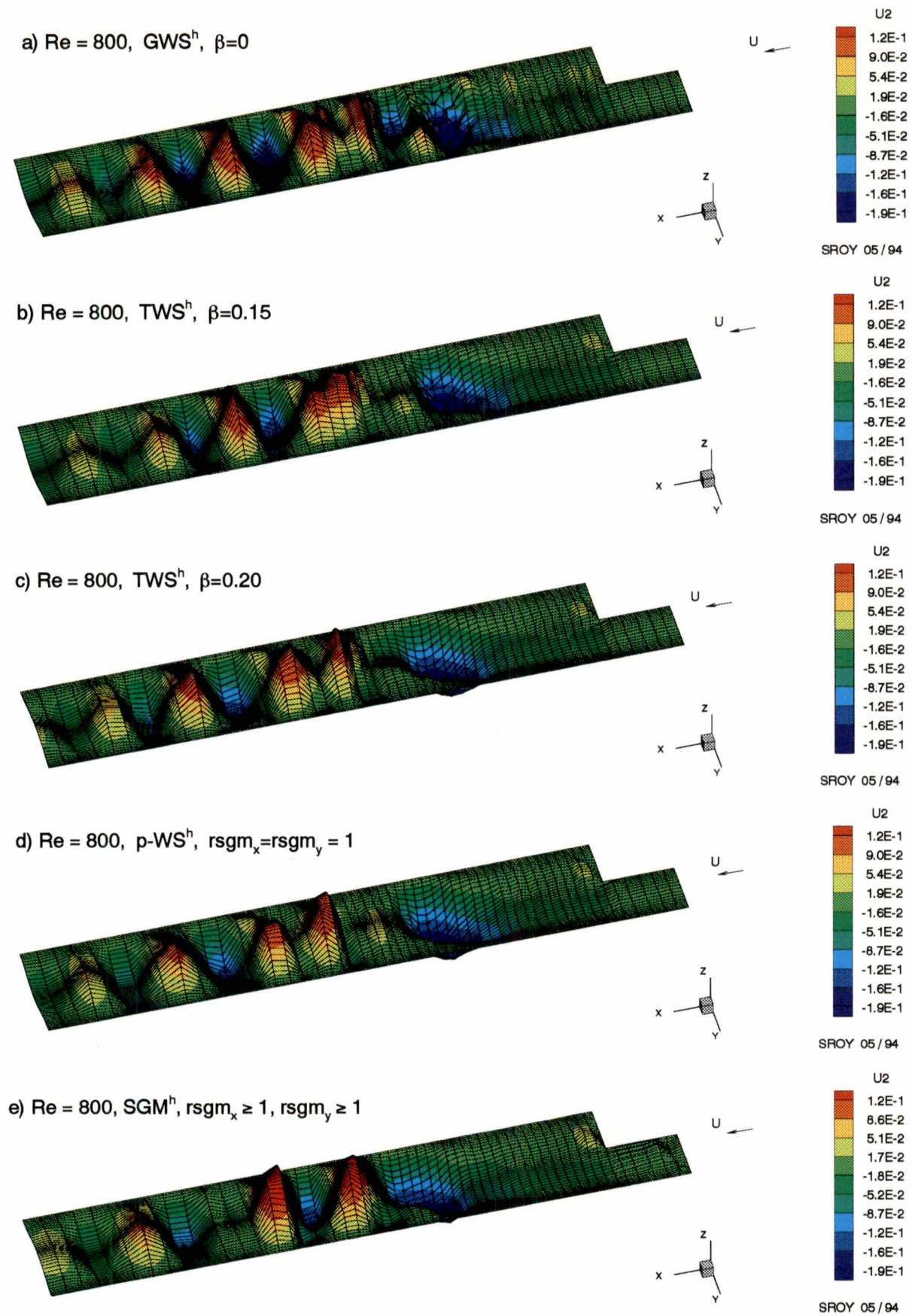


Figure 8.77 Comparison V velocity solution perspective plots for $Re = 800$.

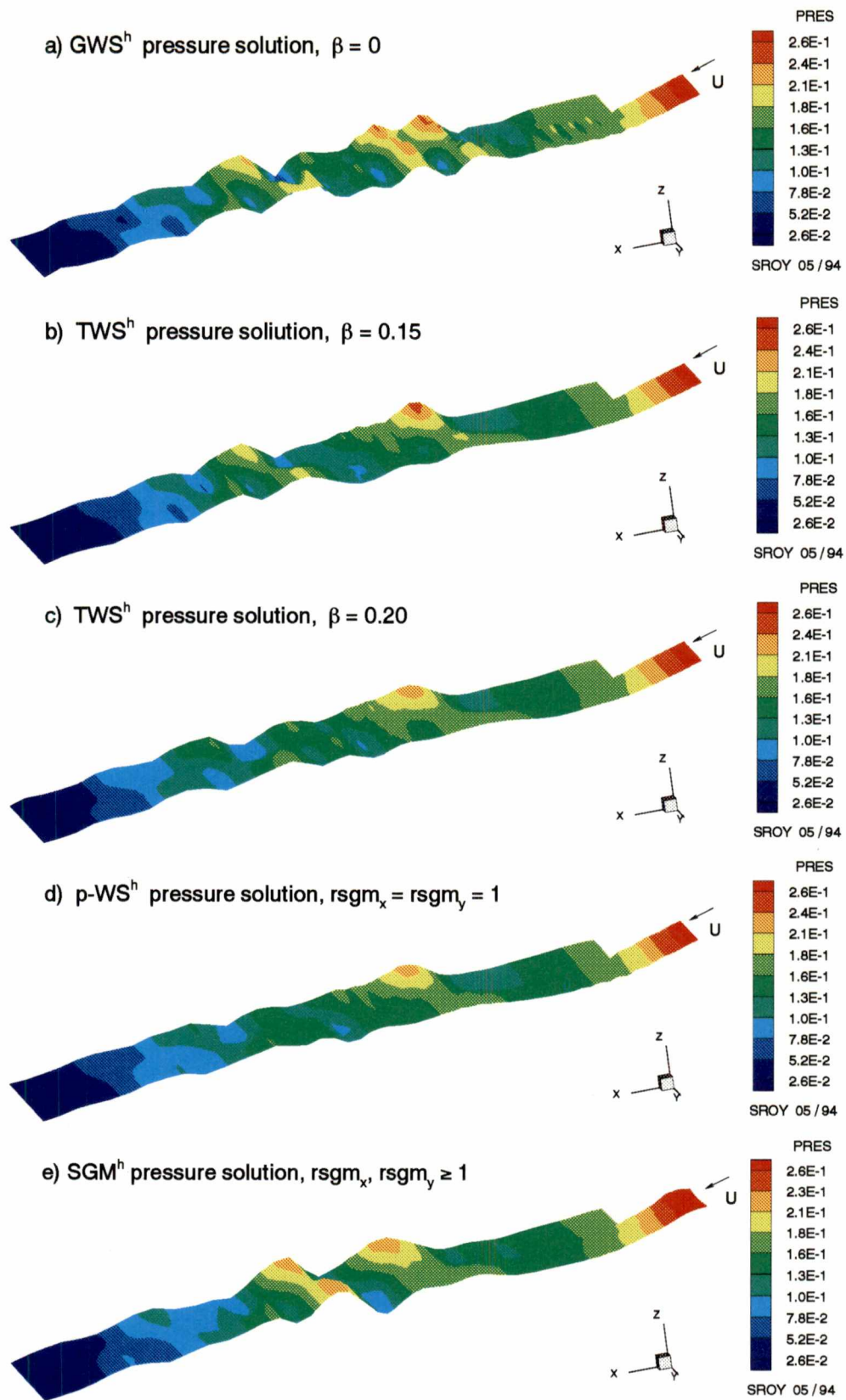


Figure 8.78 Comparison pressure solution perspective plots for $\text{Re} = 800$.

p -WS^h algorithm pressure solution, Figure 8.78d, is clearly smoother than the GWS^h and TWS^h solutions, Figure 8.78a-c, however, it is not dispersion-free (greenish yellow spikes near the orange peak (0.24 - 0.21) at the mid-section of the flow field). The SGM^h pressure solution, Figure 8.78e, is monotone and dispersion-free, and distinctly different than the other four solutions. Both the primary lower reattachment (x_1) as well as and upper wall vortex ($x_4 - x_5$), a secondary bubble ($x_2 - x_3$) solution features are also observed, recall Figure 8.62a. The three orange peaks downstream represent these two lower and one upper reattachment points, Figure 8.78e.

Because of the presence of three recirculation regions at $Re = 800$, the pressure distribution in the central region of the channel is a critical indicator of accuracy and monotonicity. Expanded views of pressure solutions in this region, shown enclosed, are shown with mesh overlays in Figure 8.79. The linear basis GWS^h solution, Figure 8.79b, is clearly highly dispersive, with rampant peaks (red, orange and yellow) and troughs (light blue and navy blue). The TWS^h solution for $\beta = 0.20$ is expanded in Figure 8.79c, confirms overdamping, however, footprints of $2\Delta x$ error oscillations clearly remain in greenish-yellow (0.16 - 0.18) patches. The p -WS^h pressure solution, Figure 8.79d, is clearly a minor modification to the TWS^h solution. Again, the SGM^h pressure solution is non-dispersive, the three local pressure peaks (reattachment locations) are vividly documented in red (0.24 - 0.26), and its appearance is totally distinct from the other solutions.

An important data set is the SGM-theory, non-linearly determined directional parameters ($rsgm_x$ and $rsgm_y$). The solution perspective details are summarized in

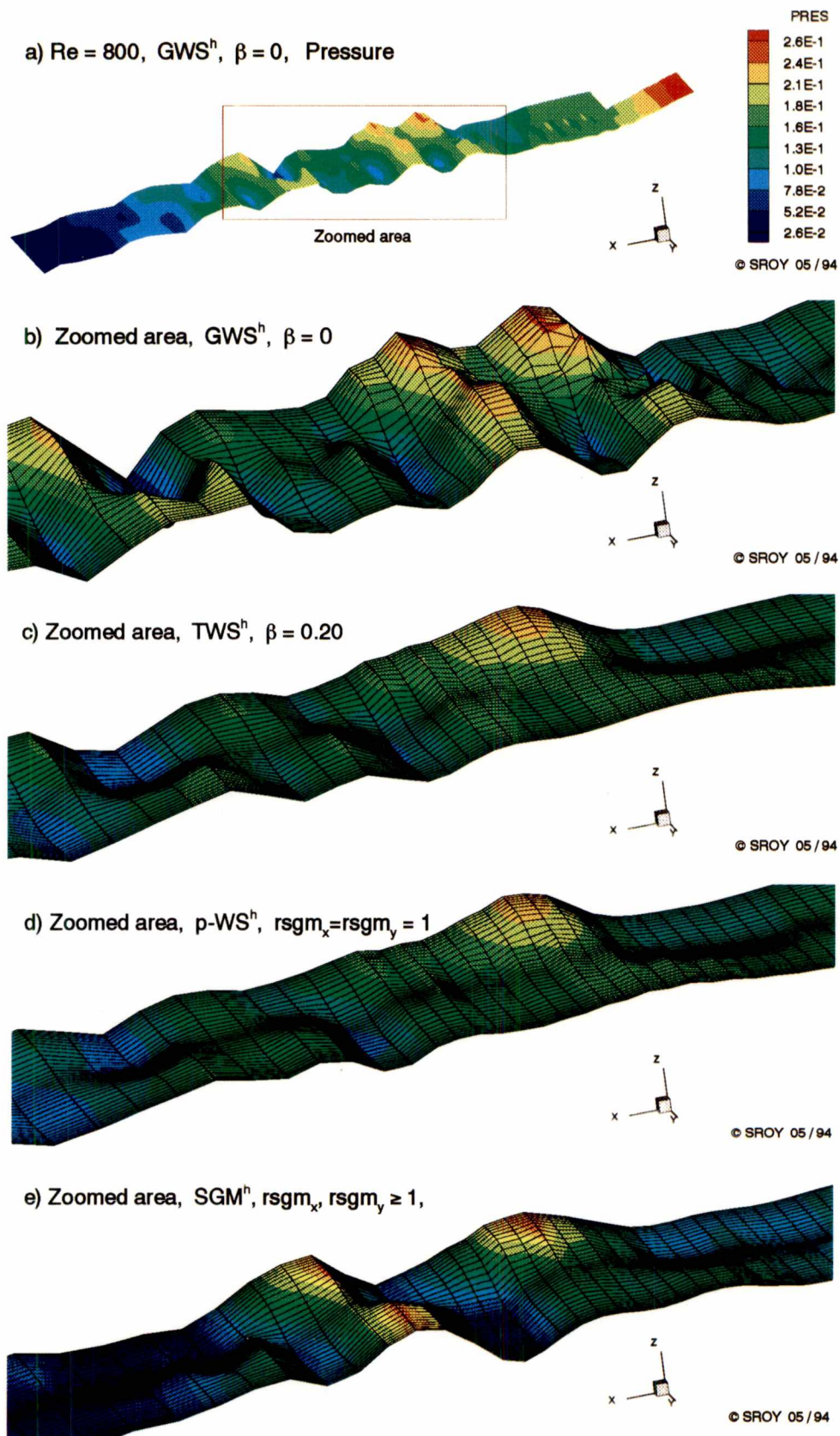


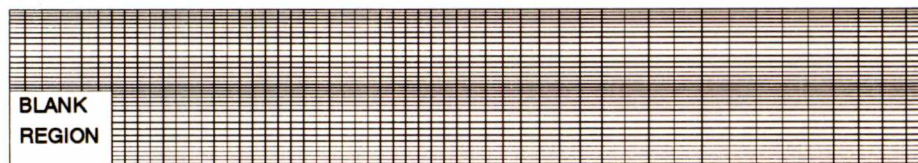
Figure 8.79 Comparison pressure solution expanded perspective plots for $Re = 800$.

Figure 8.80. The local finite element mesh is shown in Figure 8.80a. Figure 8.80b-c present the corresponding SGM^h U and V velocity solutions in perspective. The nodally distributed SGM^h directional parameters, $\text{rsgm}_x \geq 1$ and $\text{rsgm}_y \geq 1$, Figure 8.80d-e, are computed during the solution iteration as a function of the corresponding directional nodal velocity components, U and V, and the local directional mesh measures, h_x and h_y . The level of the SGM^h monotone-constraint parameter sets are noticeably higher where velocities are larger and the local mesh is coarse. In particular, the data presented in Figure 8.80d could be used to solution adapt the mesh, to reduce the local variation in rsgm_x to a smaller bound.

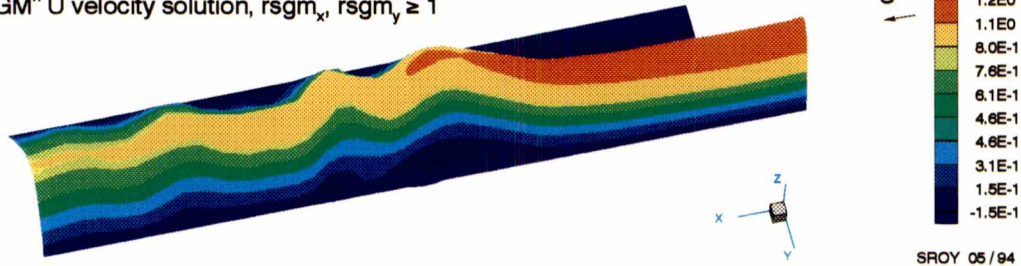
8.4.3 Airborne contaminant transport in a practical $d = 3$ room air flow field

A specific application is to reconstruct monotone species mass fraction distributions in 3-dimensions, via the RCN procedure, Chapter 7, from (dispersive) CFD solutions of the “Reynolds-averaged” PDE system (7.1). This RCN validation employs $d = 3$ velocity vector and temperature distributions obtained using a linear basis FE GWS^h algorithm without artificial diffusion, Baker, *et. al* (1993). In comparison to the available full scale 3-D experimental data reported by Spitler (1990), the CFD simulation predictions confirmed that a low turbulence level ($\text{Re}^t/\text{Re} \approx 10^{-3}$) produced best agreement, hence dispersive error mechanisms are present. Further, for $\text{Re} = 15,000$ and Archimedes number $\text{Ar} = \text{Gr}/\text{Re}^2 = 4.3$, the negative buoyancy

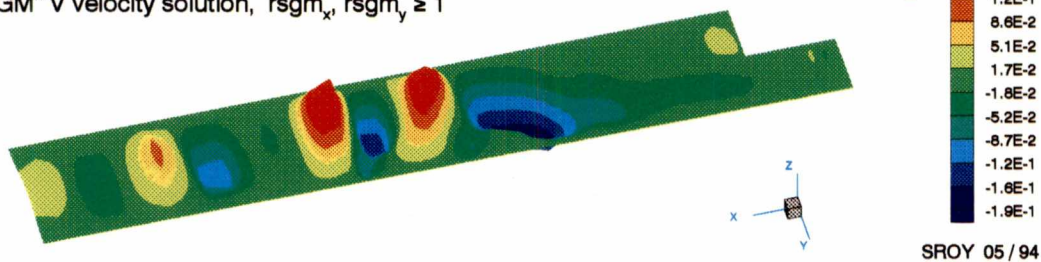
a) Finite element mesh, $7 \times 18 + 48 \times 35$



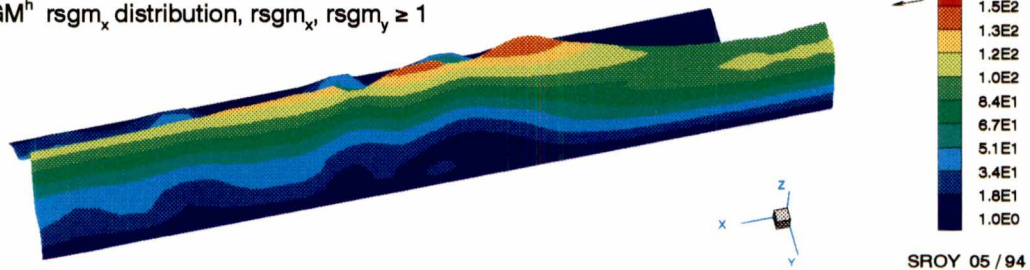
b) SGM^h U velocity solution, $rsgm_x, rsgm_y \geq 1$



c) SGM^h V velocity solution, $rsgm_x, rsgm_y \geq 1$



d) SGM^h $rsgm_x$ distribution, $rsgm_x, rsgm_y \geq 1$



e) SGM^h $rsgm_y$ distribution, $rsgm_x, rsgm_y \geq 1$

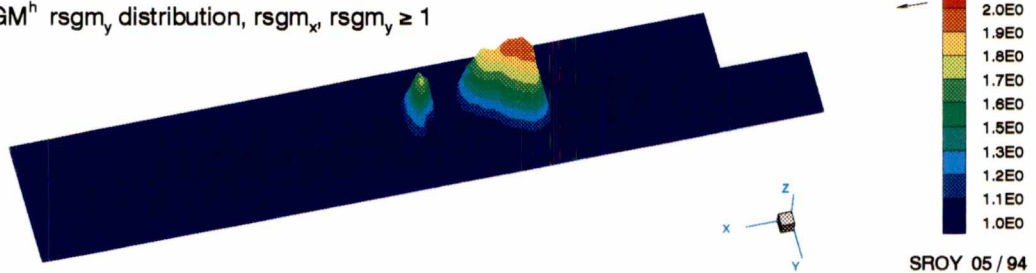


Figure 8.80 SGM^h solution perspective details for 2-D backward facing step, $Re = 800$.

effects accelerated the jet to a 3-D stagnation point on the floor, which further aggravates this error process.

Figure 8.81 summarizes the GWS^h predicted velocity vector and temperature distributions in the full scale room. The velocity field is graphed as unit vectors colored by local speed, Figure 8.81a, on a “cutting plane” bisecting the supply jet and on the end-walls and floor. This presentation confirms the dominating character of the negative-buoyant cold jet accelerating to the floor with a resounding “splat.” The bouyancy-induced maximum local speed (red, $u \cong 2.2$ ft/s), is located well away from the supply plane, and is 65% larger than the supply mean flow speed ($u \cong 1.3$ ft/s). The resulting velocity field is highly vortical, hence three-dimensional, and the associated temperature distribution is stably stratified with the cold jet floor impingement foot print clearly visible, Figure 8.81b.

For the CFD mass transport simulation, the boundary conditions are vanishing normal derivative for Y_A on all walls, the ceiling and the floor and at the room exhaust outflow plane. The initial condition is $Y_A(t_0) = 0$ everywhere, except for a concentric “cosine-hill” source s_A centered in the supply duct at the inflow plane. The summary essence of the $k = 1$ GWS^h solution for mass fraction distribution (in %) at $\Delta t = 25$ sec. is presented as flooded color contours on room far-wall planes and the floor, Figure 8.82a. For $Sc^t = 1.0 = Sc$ in (7.1), this non-artificially diffused ($\beta_t = 0$ in (16)) solution predicts that the negatively buoyant cold air jet carries the species extremum level directly to the floor stagnation point with little lateral spread. With no artificial diffusion, and minimal physical diffusion effects ($Re^t Sc^t / Re Sc \cong 10^{-3}$),

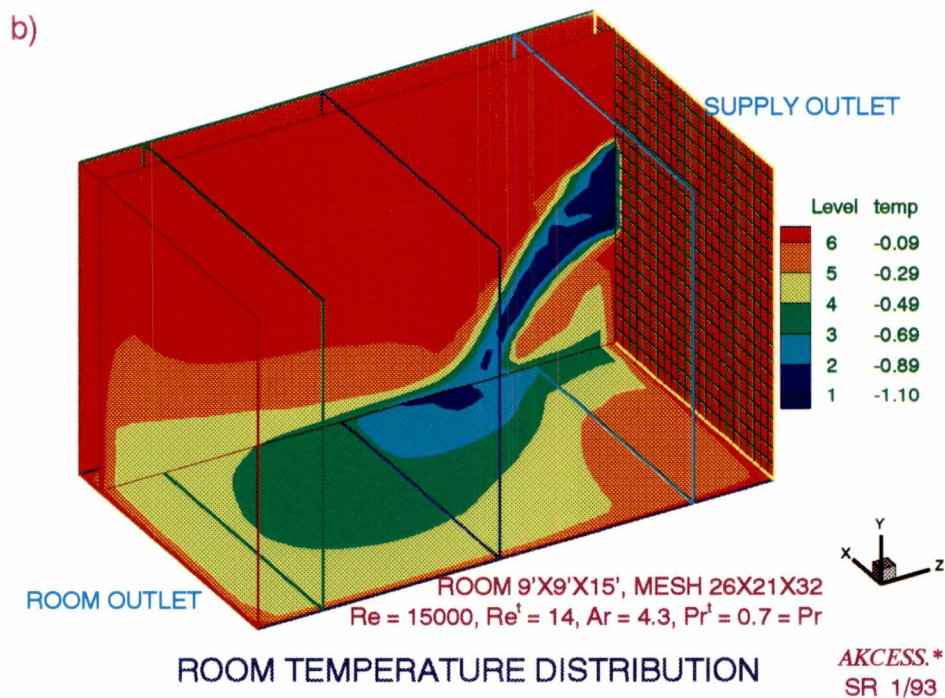
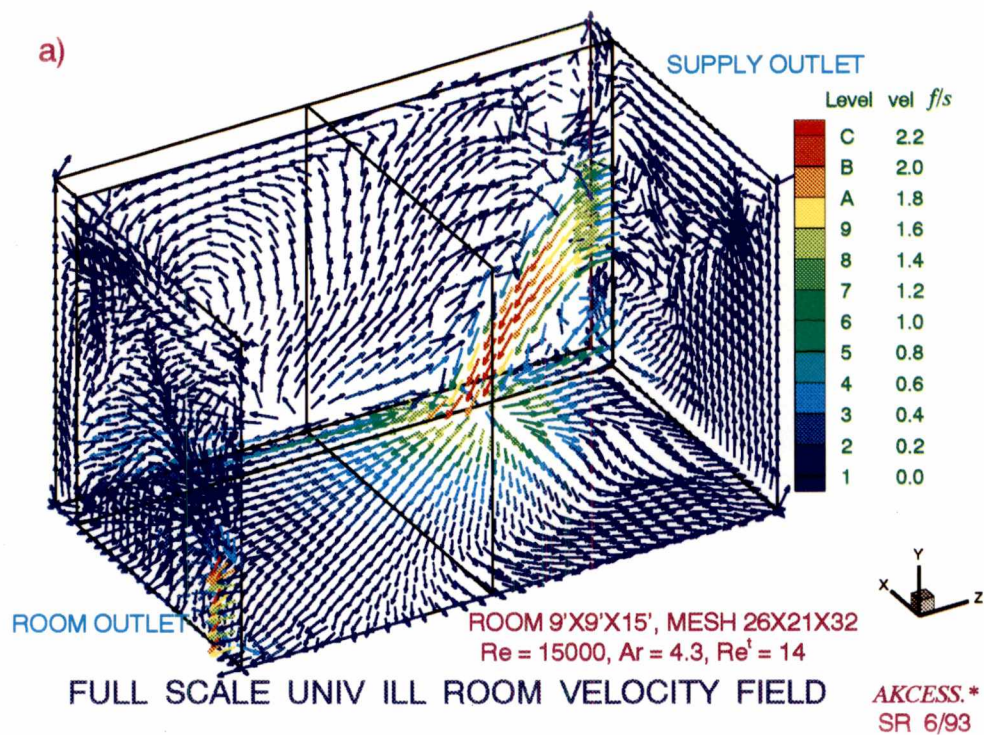
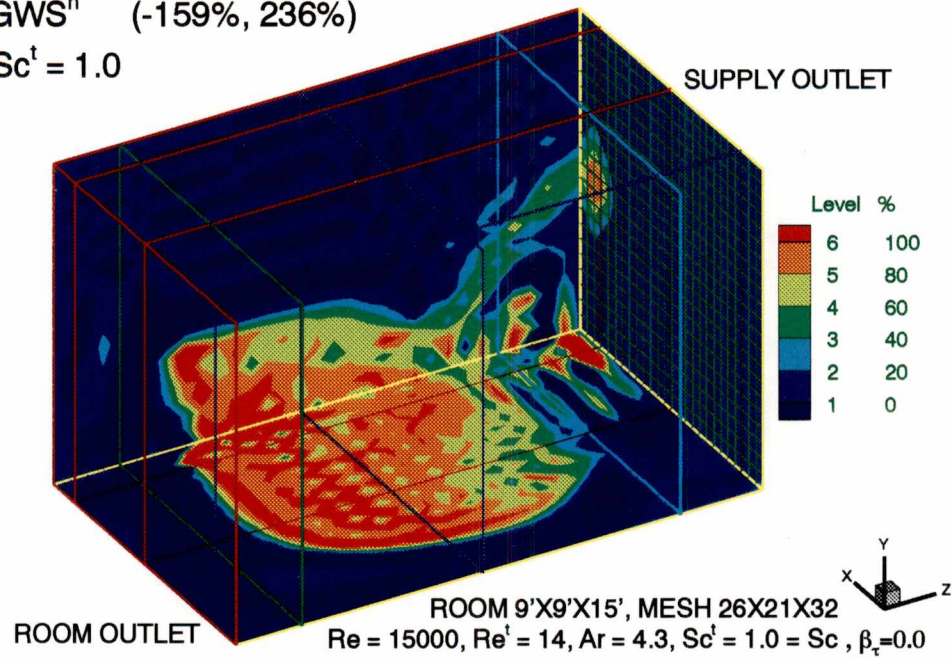


Figure 8.81 3-D room FE CFD velocity and temperature solution distribution, Williams (1993), a) Velocity vector, b) Temperature.

a) GWS^h (-159%, 236%)

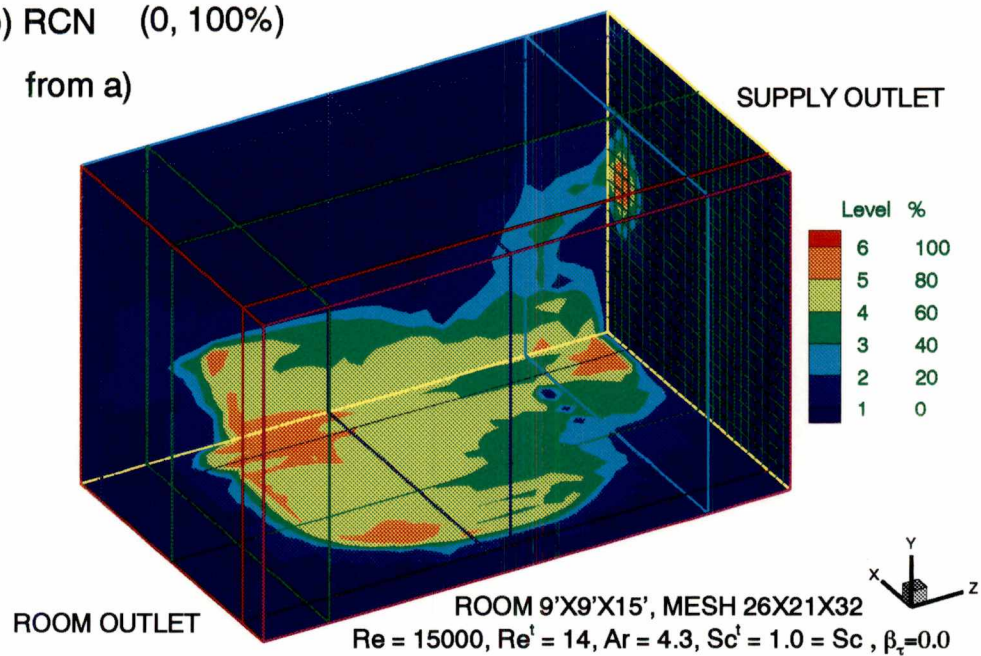
$Sc^t = 1.0$



SPECIES MASS FRACTION DISTRIBUTION, t=25 sec AKCESS.*
SR 6/93

b) RCN (0, 100%)

from a)



SPECIES MASS FRACTION DISTRIBUTION, t=25 sec AKCESS.*
SR 6/93

Figure 8.82 Inert species concentration, a) FE CFD solution, $Sc^t = 1.0$, $\beta_t = 0.0$,
b) RCN solution from a).

the high levels of species mass fraction is predicted to spread directly across the floor in a highly dispersed solution. The submerged 3-D vortex under the jet is also predicted to concentrate an extremum contaminant level (red) in the rear corner floor area, directly underneath the supply outlet.

This non-artificially diffused GWS^h solution is informative, but is distressingly suspect since dispersion error is rampant throughout. Specifically, the predicted range $-159 \leq Y_A \leq 236(\%)$ clearly exceeds the admissible range $(0, 100\%)$. This character of the 3-D problem is fully analogous to the 1-D advection-diffusion (Peclet) benchmark discussed earlier, as in both cases a stagnation point strongly activates the dispersive error characteristic of the discrete GWS^h algorithm. Thereby, this is a very appropriate test case for the RCN algorithm, as none of the artificially diffused (or non-diffused) solutions (to be described) produce an acceptably dispersion-free solution that retains definitive character with realizable extrema. Further, since the non-zero source s_A is present in (7.1), the non-linear RCN algorithm is required.

A suite of three distinct solutions for mass fraction distributions at $\Delta t = 25\text{sec.}$ are reconstructed using the non-linear RCN procedure. The base GWS^h solution also exhibits lower level dispersion error “noise everywhere (the “ripples” in the blue levels), as well as the gross dispersion distortion across the floor, Figure 8.82a. The non-linear RCN reconstruction (detailed in Section 7.2) of the solution in Figure 8.82a annihilates almost all (high and low level) dispersion error, although evidence of some (probably non-physical) local extrema in the floor plume remains, Figure 8.82b. The realizable local extrema ($\sim 99\%$) underneath the supply flow remains, as well as an

extrema at this level along the floor-backwall juncture. This RCN solution is bounded on $(0, 100\%)$ in agreement with mathematical realizability.

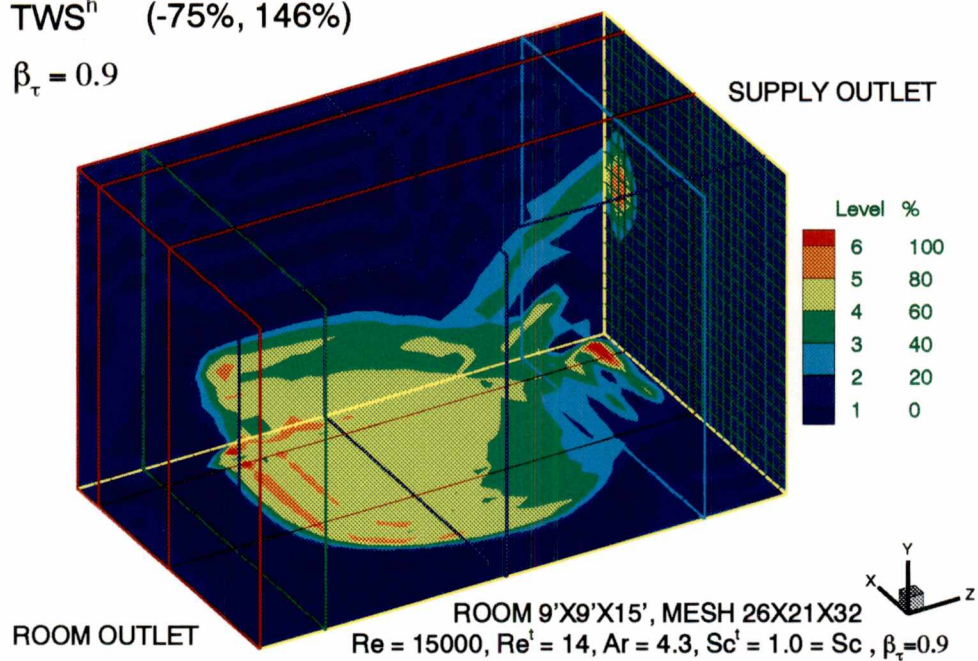
The TWS^h algorithm with dispersion error modulator ($\beta_t = 0.9$ in (3.6)) significantly reduces mass fraction solution oscillations, as precisely documented. Figure 8.83a summarizes the resultant TWS^h mass fraction distribution at $\Delta t = 25$ sec. In comparison to Figure 8.82a, the rampant floor-level dispersion oscillations are almost totally removed; however, some non-physical local extrema bands remain, and a non-monotone characteristic is still observable in the backwall levels (blue). The extremum range for Y_A is reduced to $(146, -75\%)$, and the juncture region concentration front and local pocket beneath the supply flow remain predicted. Of interest, this solution gives no significant evidence of enhanced (artificial) lateral diffusion, compare Figure 8.82a-8.83a.

The resultant RCN solution, Figure 8.83b, with range $0 \leq Y_A \leq 100\%$, is now essentially monotone everywhere, with retained extrema predicted at the backwall-floor juncture (100%) and underneath the supply flow (98%). Hence, the essential extrema features predicted in either Figure 8.82a and 8.83a remains present in the “realizable” monotone RCN solution.

The alternative (to $\beta_t > 0$) CFD experiment is to increase the physical diffusion level via reducing Sc^t by, say, an order of magnitude. Figure 8.84a presents the $\Delta t = 25$ sec. GWS^h solution for $Sc^t = 0.1$ and $\beta_t = 0$, and now only the backwall low level dispersion error remains (blue). This solution extrema are 103% and -12%, and the overall solution floor footprint is substantially more diffused, with all evidence

a) TWS^h (-75%, 146%)

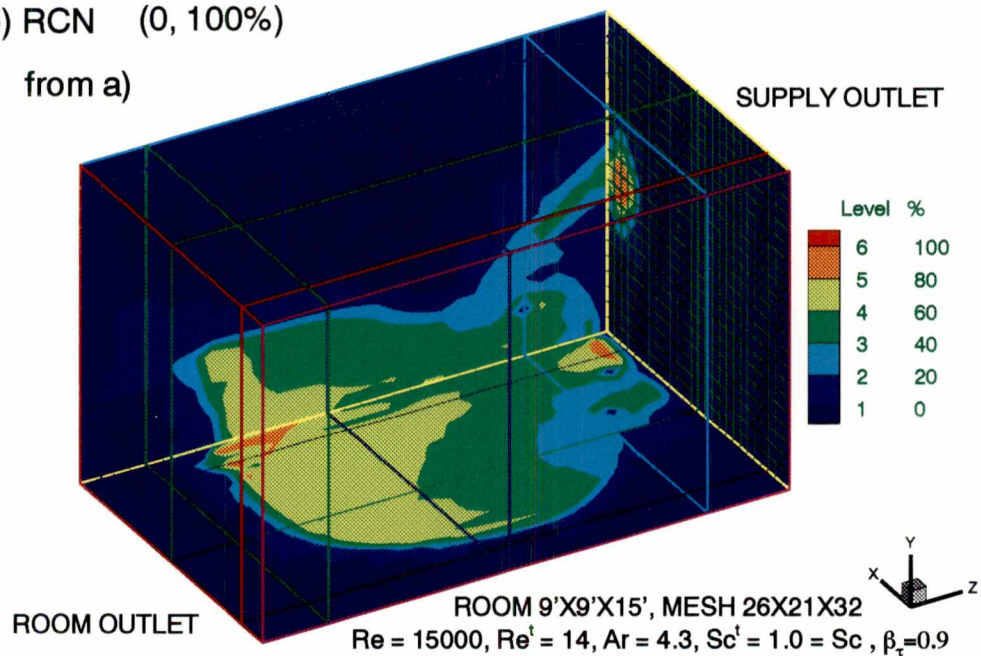
$$\beta_t = 0.9$$



SPECIES MASS FRACTION DISTRIBUTION, t=25 sec AKCESS.*
SR 6/93

b) RCN (0, 100%)

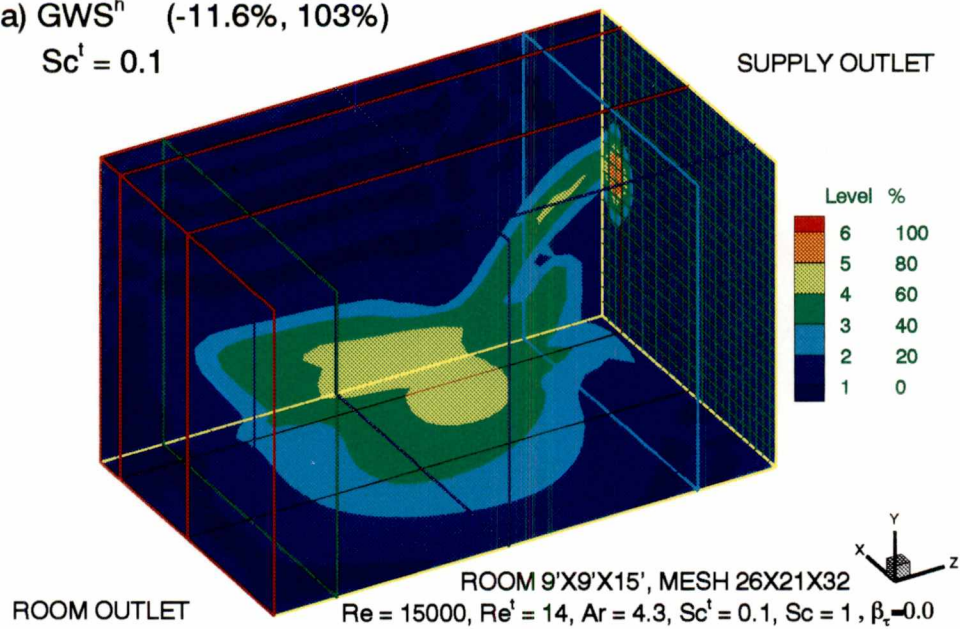
from a)



SPECIES MASS FRACTION DISTRIBUTION, t=25 sec AKCESS.*
SR 9/93

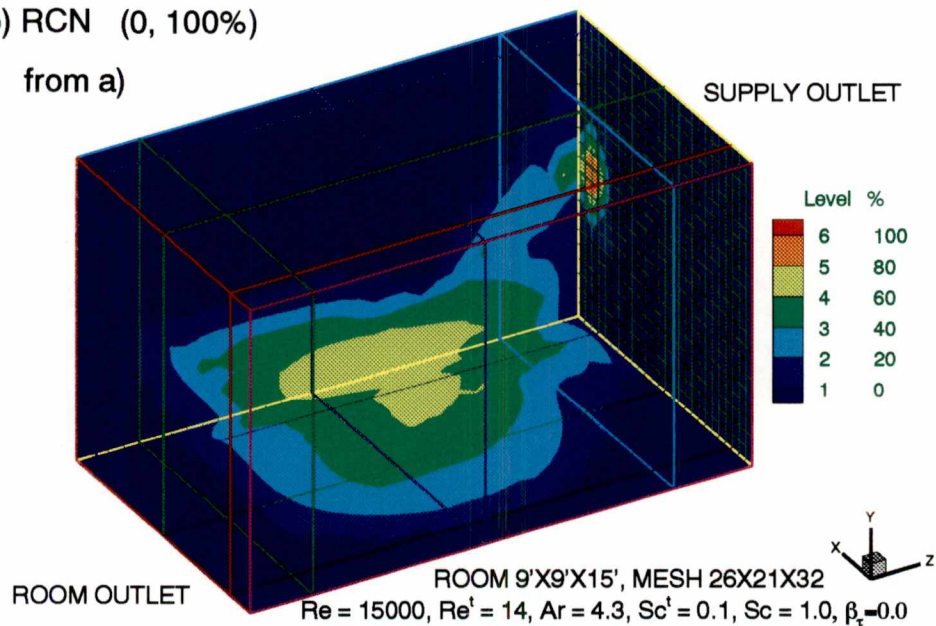
Figure 8.83 Inert species concentration, a) FE CFD solution, Sc^t = 1.0, $\beta_t = 0.9$, b) RCN solution from a).

a) GWS^h (-11.6%, 103%)
 $Sc^t = 0.1$



SPECIES MASS FRACTION DISTRIBUTION, $t=25$ sec AKCESS.*
 SR 6/93

b) RCN (0, 100%)
 from a)



SPECIES MASS FRACTION DISTRIBUTION, $t=25$ sec AKCESS.*
 SR 9/93

Figure 8.84 Inert species concentration, a) FE CFD solution, $Sc^t = 0.1, \beta = 0.0$,
 b) RCN solution from a).

of significant dispersive error absent (except in the blue). The prediction of the extremum floor-backwall juncture level and the rear corner pocket are also absent.

The RCN solution is now absolutely monotone on $0 \leq Y_A \leq 100\%$, Figure 8.84b, and differs from the parent solution only in elimination of the low level (blue) oscillations. Clearly, both the parent and RCN solutions at $Sc^t = 0.1$ have lost the definitive prediction of the local extrema existing in the less-diffused GWS^h and TWS^h solutions and their reconstructions.

The RCN procedure is equally applicable to filtering oscillations present in any CFD solution field, *e.g.*, the velocity vector field of this problem. As clearly evident in Figure 8.81a, on the rear and supply planes a $2\Delta x$ error mode is present in the GWS^h -predicted velocity field in the supply flow vicinity. The GWS^h FE algorithm velocity vector resolution (u , v and w) was reconstructed using the non-linear RCN algorithm. The GWS^h and resultant RCN monotone velocity vector fields are plotted in Figure 8.85. Although the resultant solution is devoid of any $2\Delta x$ structure, the extent of peak velocity region (red) in the RCN solution has also been noticeably reduced in the region above the floor “splat”. The true value in the RCN procedure in this application is thus as a filter to be used intermittantly in the actual CFD solution process to restore monotonicity.

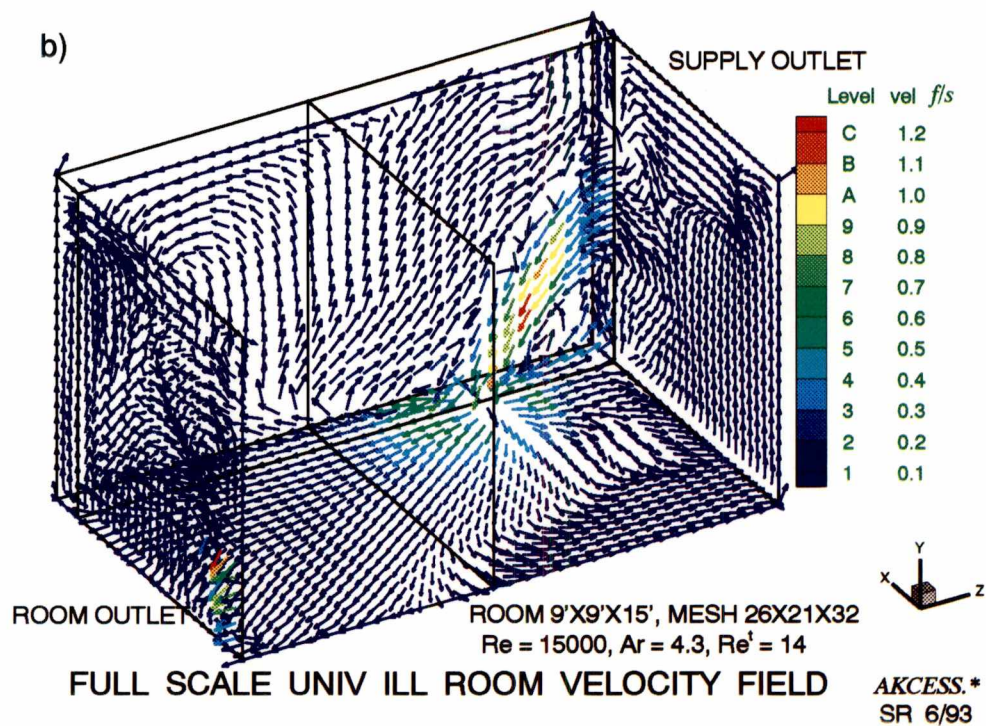
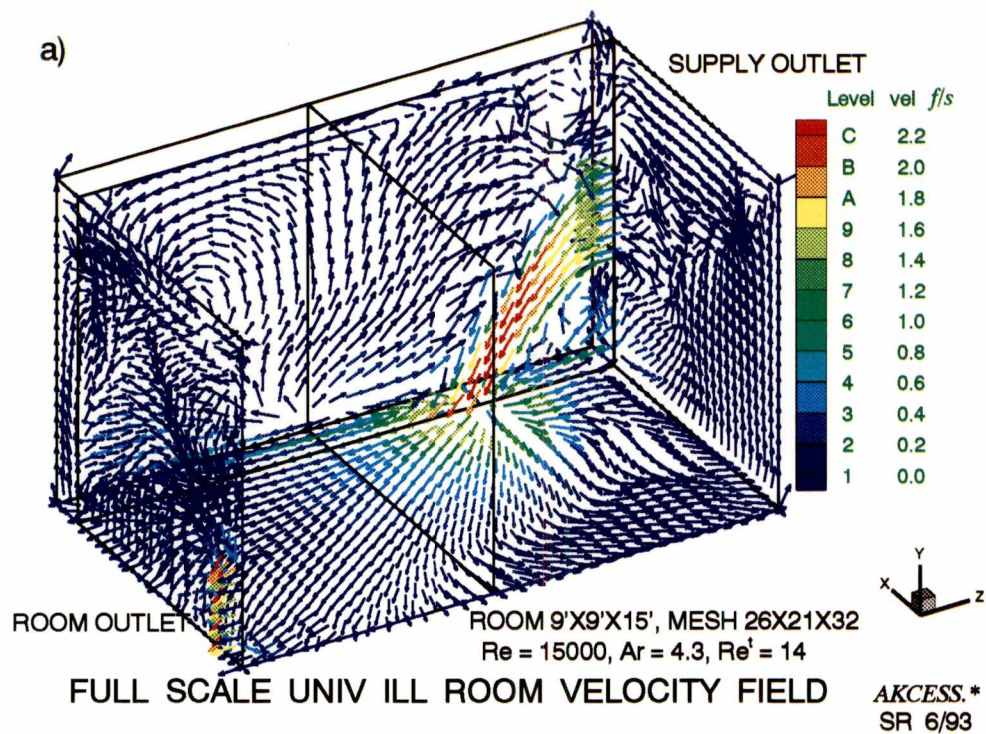


Figure 8.85 Velocity vector reconstruction, a) FE CFD solution from Williams (1993),
 b) RCN solution from a).

CONCLUSIONS AND RECOMMENDATIONS

A p -hierarchical, sub-grid embedding (SGM) procedure has been developed, validated and benchmarked for finite element semi-discrete approximate constructions of Galerkin weak statements. In comparison to other reported higher-order methods, advantages include better conditioning of the system matrix, and a much improved stability of the algorithm. The local extremization facilitates embedding of arbitrary (hierarchical) basis degree polynomials for any (all) matrices associated with the problem, hence results in minimal data storage requirements for the linear algebra statement size. Further, the methodology does permit retention of lexicographic ordering, for any SGM choice, hence exhibits the significant efficiency of the strictly linear basis (or centered FD) algorithm.

For the verification/benchmarks presented, SGM solution monotonicity can be guaranteed via a particular selection of parameters a , b , c and all $\phi > 0$. A proper selection of the f_2 choice c is confirmed to invariably yield an essentially nodally exact SGM solution on any given (coarse) mesh. The parameter ϕ is proven of use in enhancing monotone steady or periodic stage of the transient solution. Finally, boundary fluxes computed from the SGM weak statement WS^h (FEM in general)

definition are confirmed up to several orders of magnitude more accurate than these computed via FD or standard FE basis methods on coarse meshes.

The SGM^h algorithm is also highly parallelizable, as it is based on an elementwise and/or nodewise distributed parameter that can be exactly calculated in one dimension.

A weak statement perturbation (GMP) technique based on Fourier modal analysis of the approximation solution amplification factor has been developed for generating progressively higher order accurate algorithm constructions for non-diffusive (pure convection) applications. The GMP theory is complete and easy to incorporate in one-dimension, or in two-dimensions using time-splitting. It exhibits the efficiency of a strictly linear basis FE GWS (or centered FD) algorithm. The approximation solution process for the modified weak statement, using a local element Courant number definition, is mesh and time-step independent for the model problems analyzed.

A reconstruction (RCN) algorithm, applicable for dispersion error annihilation in *any* CFD solution, has been developed, verified and validated for representative verification and benchmark problems for heat and mass transport, CFD applications. The RCN procedure can not correct the phase lag inherent in the parent CFD solution. However, the RCN procedure has generated dispersion-free, essentially monotone solutions in the non-linear mode, even when the parent CFD solution is highly polluted by dispersive error. Definitive benchmarks have been presented, and a practical 3-D application has provided definitive interpretation of a set of CFD experiments on mass transport.

The algorithms presented in this dissertation provide an excellent point of departure for future research and development of benchmark-quality solutions and validation-quality experimental data for three dimensional flow fields.

Bibliography

Bibliography

- Angeloni, P., R. Boccellato, E. Bonacina, A. Pasini and A. Peano, 1978, Accuracy assessment by finite element p -version software, *ISMES*, Bergamo, Italy, pp. 6-35 - 6-48.
- Armaly, B.F., F. Durst, J.C.F. Pereira and B. Schönung, 1983, Experimental and theoretical investigation of backward-facing step flow, *Journal of Fluid Mechanics*, **127**, pp. 473-496.
- Babuška, I., 1975, The selfadaptive approach in the finite element method, *Mathematics of Finite Elements and Applications* (Edited by J.R. Whiteman), Academic Press, London.
- Babuška, I., B.Q. Guo and E.P. Stephan, 1990, On the exponential convergence of the h - p version for boundary element Galerkin methods on polygons, *Math. Methods Appl. Sci.*, **12**, pp. 413-427.
- Babuška, I., A. Craig, J. Mandel and J. Pitkäranta, 1991, Efficient Preconditioning for the p -version finite element method in two dimensions, *SIAM Journal for Numerical Analysis*, **28**, pp. 624-661.
- Babuška, I., L. Planck and R. Rodriguez, 1992, Basic problems of a posteriori error estimation, *Computer Methods in Applied Mechanics and Engineering*, **101**, pp. 97-112.
- Babuška, I. and W.C. Rheinboldt, 1981, A posteriori error analysis of finite element solutions for one-dimensional problems, *Siam J. Numerical Analysis*, **18**, pp. 565-589.

Babuška, I., B.A. Szabo and I.N. Katz, 1981, The p -version of the Finite Element Method, *Siam J. Num. Anal.*, **18**, pp. 515-546.

Babuška, I. and M. Suri, 1987, The p -version of the finite element method for constraint boundary conditions, *Institute for Physical Science and Technology*, Technical Note BN-1064.

Baker, A.J., 1983, Finite Element Computational Fluid Mechanics, Hemisphere Publishing Corporation, New York.

Baker, A.J. and D.W. Pepper, 1991, Finite Element 1-2-3, McGraw-Hill, Inc., New York.

Baker, A.J. and J.W. Kim, 1987, *Int. J. for Num. Methods in Fluids*, **7**, pp. 489-520.

Baker, A.J., D.J. Chaffin, S. Roy, E.G. Schaub, and P.T. Williams, Aug 1992, Step wall benchmark problem prediction via three finite element algorithms, ASME News.

Baker, A.J., P.T. Williams, D.J. Chaffin and S. Roy, 1993, Recent advances in finite element CFD modeling for environmental flow sciences, *Advances in Hydro-Science and -Engineering* (Edited by Sam S.Y. Wang), **1**, pp. 32-45.

Baker, A.J., Subrata Roy and R.M. Kelso, 1994, CFD experiment characterization of airborne contaminant transport for two practical 3-D room air flow fields, *Building and Environment*, **29**, No. 3, pp. 253-259.

- Bank, R.E., A.H. Sherman and A. Weiser, 1983, Refinement algorithms and data structures for regular mesh refinement, *Scientific Computing* (Edited by R. Stepleman, et al.), IMACS, North Holland, pp. 1-17.
- Cabuk, H., C.H. Sung and V. Modi, 1992, Explicit Runge-Kutta method for three-dimensional internal incompressible flows, *AIAA Journal* **30**, pp. 2024-2031.
- Chaffin, D.J. and A.J. Baker, 1993, Taylor-Galerkin finite element methods for Computational Fluid Dynamics, *International Journal for Numerical Methods in Fluids* (in review).
- Choi, H.G. and J.Y. Yoo, 1994, Streamline upwind scheme for the segregated formulation of the Navier-Stokes equation, *Numerical Heat Transfer*, **25**, Part B, pp. 145-161.
- Ciarlet, Philip G., 1978, The Finite Element Methods for Elliptic Problems, North Holland, Amsterdam.
- Daube, O., 1992, Resolution of the 2D Navier-Stokes equations in velocity-vorticity form by means of an influence matrix technique, *Journal of Computational Physics*, **103**, pp. 402-414.
- Demkowicz, L., A. Karafiat and J.T. Oden, 1992, Solution of elastic scattering problems in linear acoustics using h - p boundary element method, *Computer Methods in Applied Mechanics and Engineering*, **101**, pp. 251-282.
- Demkowicz, L., J.T. Oden, W. Rachowicz and O. Hardy, 1989, Toward a universal h - p adaptive finite element strategy, Part 1. Constrained approximation and data

structure, *Computer Methods in Applied Mechanics and Engineering*, **77**, pp. 79-112.

de Vahl Davis, G., 1983, Natural Convection of Air in a Square Cavity: A Benchmark Numerical Solution, *International Journal for Numerical Methods in Fluids*, **3**, pp. 249-264.

Donoghue, W.F., 1974, Monotone matrix functions and analytic continuation, Springer-Verlag, Berlin.

Evans, G. and S. Paolucci, 1990, The thermoconvective instability of plane Poiseuille flow heated from below: a proposed benchmark solution for open boundary flows, *International Journal for Numerical Methods in Fluids*, **11**, pp. 1001-1013.

Farouk, B. and T. Fusegi, 1985, A coupled solution of the Vorticity-Velocity formulation of the incompressible Navier-Stokes equations, *International Journal for Numerical Methods in Fluids*, **5**, pp. 1017-1034.

Gartling, D.K., 1990, A test problem for outflow boundary conditions-flow over a backward-facing step, *International Journal for Numerical Methods in Fluids*, **11**, pp. 953-967.

Genuchten, M.T.V. and W.G. Gray, 1978, Analysis of some dispersion corrected numerical schemes for solution of the transport equation, *International Journal for Numerical Methods in Engineering*, **12**, pp. 387-404.

Geist, G.A., 1989, Reduction of a general matrix to tridiagonal form, ORNL/TM-10991.

- Ghia, U., K.N. Ghia and C.T. Shin, 1982, High-Re solutions for incompressible flow using the Navier-Stokes Equations and a multigrid method, *Journal of Computational Physics*, **48**, pp. 387-411.
- Ghia, K.N., G.A. Osswald and U. Ghia, 1989, Analysis of incompressible massively separated viscous flows using unsteady Navier-Stokes equations, *International Journal for Numerical Methods in Fluids*, **9**, pp. 1025-1050.
- Gresho, P.M., D.K. Gartling, K.A. Cliffe, T.J. Garratt, A. Spence, K.H. Winters, J.H. Goodrich, and J.R. Torczynski, 1993, Is the steady viscous incompressible 2D flow over a backward-facing step at $Re = 800$ stable? *International Journal of Numerical methods in Fluids*, **17**, pp. 501-541.
- Gresho, P.M. and R.L. Sani, 1987, On pressure boundary conditions for the incompressible Navier-Stokes equations, *International Journal for Numerical Methods in Fluids*, **7**, pp. 1111-1145.
- Gresho, P.M. and R.L. Lee, 1981, Don't suppress the wiggles - they're telling you something! *Computers and Fluids*, **9**, pp. 223-253.
- Guj, G. and F. Stella, 1988, Numerical solutions of high-Re recirculating flows in Vorticity-Velocity form, *International Journal for Numerical Methods in Fluids*, **8**, pp. 405-416.
- Gunzburger, M.D., 1989, Finite Element Methods for Viscous Incompressible Flows, New York, Academic Press, Inc.
- Haras, Z. and S. Ta'asan, 1993, *Finite difference schemes for long-time integration*, NASA CR-191471, ICASE Report No. 93-25.

- Harten, A. and H. Tal-Ezer, 1981, On a fourth order accurate implicit finite difference scheme for hyperbolic conservation laws: I. nonstiff strongly dynamic problems, *Mathematics of Computation*, **36**, No. 154, pp. 353-373.
- Hirsch, C., 1992, Numerical Computation of Internal and External Flows, **1,2**, Computational Methods for Inviscid and Viscous Flows, John Wiley & Sons Ltd., West Sussex, England.
- Heinrich, J.C. and O.C. Zienkiewicz, 1978, Finite Elements in Fluids, III, Wiley-Interscience, New York.
- Hughes, T.J.R., W.K. Liu and A. Brooks, 1979, Finite element analysis of incompressible viscous flows by the penalty method, *Journal of Computational Physics*, **30**, pp. 1-60.
- Huynh, H.T., 1993, Accurate upwind methods for the Euler equations, NASA TM 106388.
- Huynh, H.T., 1993, Accurate monotone cubic interpolation, *SIAM Journal of Numerical Analysis*, **30**, No.1, pp. 57-100.
- Iannelli, G.S., 1992, The method of lines in computational fluid dynamics, *First Int. Sym. on Method of Lines, Surfaces and Dimensional reduction in Comp. Math. and Mech.*, London, UK.
- Ikohagi, T., B.R. Shin and H. Daiguji, 1992, Application of an implicit time-marching scheme to a three-dimensional incompressible flow problem in curvilinear coordinate system, *Computers and Fluids* **21**, pp. 163-175.

- Janik, T. and S. Jensen, 1993, Stability and performance of the p -version of the finite element method for first order hyperbolic problems, *Appl. Numer. Math.*, submitted.
- Jensen, S., 1992, p -version of mixed finite element methods for Stokes-like problems, *Computer Methods in Applied Mechanics and Engineering*, **101**, pp. 27-41.
- Jensen, S. and M. Suri, 1992, On the L_2 error for the p -version of the finite element method over polygonal domains, *Computer Methods in Applied Mechanics and Engineering*, **97**, pp. 233-244.
- Jiang, B.N., 1989, A least-squares finite element method for incompressible Navier-Stokes problems, NASA TM 102385, ICOMP-89-28.
- Jiang, B.N. and L.A. Povinelli, 1991, Optimal least-squares finite element method for elliptic problems, NASA TM 105382, ICOMP-91-29.
- Johnson, C. and P. Hansbo, 1992, Adaptive finite element methods in computational mechanics, *Computer Methods in Applied Mechanics and Engineering*, **101**, pp. 143-181.
- Jorgenson, P. and E. Turkel, 1993, Central difference TVD schemes for time dependent and steady state problems, *Journal of Computational Physics* **107**, pp. 297-308.
- Kelso, R.M., S. Roy, and A.J. Baker, 1993, A CFD study of thermal comfort distribution in a space with mixing and displacement ventilation, INDOOR AIR '93, Helsinki, Finland.

- Kim, J. and P. Moin, 1985, Application of a fractional-step method to incompressible Navier-Stokes equations, *Journal of Computational Physics* **59**, pp. 308-323.
- Kreiss, H.O. and J. Lorenz, 1989, Initial-Boundary Value Problems and the Navier-Stokes equations, Academic Press, New York, NY.
- Ladyzhenskaya, A., 1956, On the construction of discontinuous solutions of quasi-linear hyperbolic equations as limits to the solutions of the respective parabolic equations when the viscosity coefficient is approaching zero, *Doklady Akad. Nauk USSR*, **3**, pp. 291-295.
- Lax, P.D., 1954, Weak solutions of nonlinear hyperbolic equations and their numerical computation, *Comm. Pure Appl. Math.* **7**, pp. 159-193.
- Layton, W., 1990, On the principal axes of diffusion in difference schemes for 2d transport problems, *Journal of Computational Physics*, **90**, pp. 336-347.
- Lele, S.K., 1992, *Compact finite difference schemes with spectral-like resolution*, J. Comp. Phys., **103**, pp. 16-42.
- Leone, J.M., 1990, Open boundary condition symposium benchmark solution: stratified flow over a backward-facing step, *International Journal for Numerical Methods in Fluids*, **11**, pp.969-984.
- MacsymaTM Reference Manual, 1988, Version 13, Symbolics, Inc., Cambridge, MA.
- Markatos, N.C. and K.A. Pericleous, 1984, Laminar and Turbulent Natural Convection in an Enclosed Cavity, *Int. J. Heat Mass Transfer*, **27**, pp. 755-772.

- McNiece, B.M. and F.V. Marcal, 1971, Optimisation of finite element grids based on maximum potential energy, *Technical Report No. 7*, Brown University Providence.
- Melosh, R.J. and D.E. Killian, 1976, Finite element analysis to attain a pre-specified accuracy, *Proc. 3rd Nat. Congress on Computing in Structures*.
- Mulay, S.B., 1993, Private communications.
- Nguyen, Steve H., 1992, Use of a p -version finite element formulation for compressible lubrication calculations (to be published).
- Noronha, W.P., 1989, PhD Dissertation, University of Tennessee, Knoxville.
- Oden, J.T., 1992, Optimal hp -finite element methods, *TICOM Report 92-09*, The Texas Institute for Computational Mechanics, The University of Texas at Austin, Austin, TX.
- Oden, J.T., L. Demkowicz, W. Rachowicz and T.A. Westermann, 1989, Toward a universal h - p adaptive finite element strategy, Part 2. A posteriori error estimation, *Computer Methods in Applied Mechanics and Engineering*, **77**, pp. 113-180.
- Oden, J.T. and J.N. Reddy, 1976, An Introduction to the Mathematical Theory of Finite Elements, Wiley-Interscience, New York.
- Oh, H.-S. and I. Babuška, 1992, The p -version of the finite element method for the elliptic boundary value problems with interfaces, *Computer Methods in Applied Mechanics and Engineering*, **97**, pp. 211-231.
- Oleinik, O.A., 1959, Uniqueness and stability of the generalized solutions of the Cauchy problem for quasi-linear equation, *Uspehi Mat Nauk* **14**, **2(86)**, pp. 165-170 (english translation in *Amer. Math. Soc. Transl.* (2), **33**, pp. 285-290).

- Ortega, J., 1972, Numerical Analysis A Second Course, Academic Press, New York.
- Pasini, A., A. Peano, R. Riccioni and L. Sardella, 1977, A self-adaptive finite element analysis, *Proc. VIth Int. Finite Element Congress*, pp. 17-43, Baden Baden.
- Peano, A., A. Pasini, R. Riccioni and L. Sardella, 1979, Adaptive approximations in finite element structural analysis, *Computers & Structures*, **10**, pp. 333-342.
- Peano, A. and R. Riccioni, 1978, Automated discretization error control in finite element analysis, *Proc. 2nd World Congress on Finite Element Methods*, Dorset, England.
- Pepper, D.W., C.D. Kern and P.E. Long, 1979, Modelling the dispersion of atmospheric pollution using cubic splines and chapeau functions, *Atmospheric Environment*, **13**, pp. 223- 237.
- Rachowicz, W., J.T. Oden and L. Demkowicz, 1989, Toward a universal *h-p* adaptive finite element strategy, Part 3. Design of *h-p* meshes, *Computer Methods in Applied Mechanics and Engineering*, **77**, pp. 181-212.
- Roy, S., R.M. Kelso, and A.J. Baker, 1994, An efficient CFD algorithm for prediction of contaminant dispersion in room air motion, (Accepted for publication) *ASHRAE Transactions*.
- Roy, S. and A.J. Baker, 1993, A non-linear, sub-grid embedding weak statement algorithm for optimal adaptive CFD applications, *Journal of Computational Physics* (in review).

- Roy, S. and A.J. Baker, 1993, A weak statement perturbation CFD algorithm with high-order phase accuracy for hyperbolic problems, *Computer Methods in Applied Mechanics and Engineering* (in review).
- Roy, S. and A.J. Baker, 1993, A reconstruction algorithm for phase error annihilation in heat and mass transport CFD methods, *International Journal for Numerical Methods in Fluids* (in review).
- Roy, S., A.J. Baker, and R.M. Kelso, 1993, Airborne contaminant CFD modeling studies for two practical 3D room air flow fields, INDOOR AIR '93, Helsinki, Finland.
- Rubin, S.G. and P.K. Khosla, 1977, *Journal of Computational Physics*, **9**, 163.
- Scapolla, T. and L. Della Croce, 1992, On the robustness of hierarchic finite elements for Reissner-Mindlin plates, *Computer Methods in Applied Mechanics and Engineering*, **101**, pp. 43-60.
- Schaub, E.G., 1993, M.S. Thesis, University of Tennessee, Knoxville.
- Sewell, G., An adaptive computer program for the solution of $\text{div}(p(x,y) \text{ grad } u) = f(x,y,u)$ on a polygonal region, pp. 543-553.
- Smith, R.E. and A. Kidd, 1975, Comparative study of two numerical techniques for the solution of viscous flow in a driven cavity, NASA SP-378, pp. 61-82.
- Sohn, J., 1988, Evaluation of FIDAP on some classical laminar and turbulent benchmarks, *International Journal of Numerical Methods in Fluids* **8**, pp. 1469-1490.
- Spitler, J.D., 1990, Ph.D. Dissertation, University of Illinois, Urbana.

- Strang, G. and G.J. Fix, 1973, *An Analysis of the Finite Element Method*, Prentice Hall, Englewood Cliffs, New Jersey.
- Suri, M., 1990, The p -version of the finite element method for elliptic problems of order 2l, *RAIRO Modél. Math. Anal. Numér.*, **24**, pp. 265-304.
- Tamma, Kumar K. and Kong C. Saw, 1989, Hierarchical p -version finite elements and adaptive *a posteriori* computational formulations for two-dimensional thermal analysis, *Computers & Structures*, **32**, pp. 1183-1194.
- Targowski, R., G. Guerlement and D. Lamblin, 1992, Geometric interpretation for error estimation in finite element analysis, *Computer Methods in Applied Mechanics and Engineering*, **101**, pp. 197-206.
- TecplotTM Version 6.0 User's Manual, 1993, Bellevue, WA: Amtec Engineering, Inc.
- Temam, R., 1968, Une méthode d'approximation de la solution des équations de Navier-Stokes, [A method of approximation of the solution for Navier-Stokes equations], *Bulletin of Soc. Math. France*, **96**, pp. 115-152.
- Temam, R., 1977, *Navier-Stokes Equations*, North-Holland, Amsterdam.
- Temam, R., 1983, *Navier-Stokes Equations and Non-linear Functional Analysis*, Philadelphia: Society for Industrial and Applied Mechanics.
- Thangam, S. and D.D. Knight, 1990, A computational scheme in generalized coordinates for viscous incompressible flows, *Computers and Fluids* **18**, pp. 317-327.
- Vichenevetsy, R. and J.B. Bowles, 1982, *Fourier analysis of numerical approximations of hyperbolic equations*, SIAM, 140pp.

- Wachspress, E.L., 1982, Mathematical foundations for numerical solution of Navier-Stokes equations, *Topical Conference of the Mathematics and Computation Division of the American Nuclear Society*, KAPL-4152, Knolls Atomic Power Laboratory, Schenectady, New York.
- Wachspress, E.L., 1975, A Rational Finite Element Basis, Academic Press, New York.
- Whiteman, J.R., 1982, Finite elements for singularities in two- and three-dimensions, *The Mathematics of Finite Elements and Applications IV* (Edited by J. Whiteman), pp. 36-55, Academic Press, New York.
- Wilkinson, J.H, The perfidious polynomial, *Studies in Numerical Analysis*.
- Williams, P.T., 1993, Ph.D. Dissertation, University of Tennessee, Knoxville.
- Winterscheidt, D., 1993, A p -version finite element method for steady incompressible fluid flow and convective heat transfer, NASA TM 106260, ICOMP-93-23.
- Zhu, Q.D. and Q. Lin, 1989, Superconvergence Theory of FEM, Hunan Science Press.
- Zienkiewicz, O.C., 1977, The finite element method, 3d ed., McGraw-Hill Book Co., New York.
- Zienkiewicz, O.C. and A.W. Craig, 1984, A-posteriori error estimation and adaptive mesh refinement in the finite element method, *The Mathematical Basis of Finite Element Methods* (Edited by David F. Griffiths), Clarendon Press, Oxford.
- Zienkiewicz, O.C., J.P. de S.R. Gago and D.W. Kelly, 1983, The hierarchical concept in finite element analysis, *Computers & Structures*, **16**, pp. 53-65.

Zienkiewicz, O.C., D.W. Kelly, J. Gago and I. Babuska, 1982, Hierarchical finite element approaches, error estimates and adaptive refinement, *The Mathematics of Finite Elements and Applications IV* (Edited by J. Whiteman), pp. 311-346, Academic Press, New York.

Zienkiewicz, O.C. and J.Z. Zhu, 1992, The superconvergent patch recovery (SPR) and adaptive finite element refinement, *Computer Methods in Applied Mathematics and Engineering*, **101**, pp. 207-224.

APPENDICES

Appendix A

Consider the standard FE linear basis one-dimensional matrices [A200] and [A201] in (1.4), *i.e.*

$$[M]_e = \Delta x_e [A200L] = \frac{\Delta x_e}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}_{2 \times 2} \quad (A.1)$$

and

$$[U]_e = U_e [A201L] = \frac{U_e}{2} \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}_{2 \times 2} \quad (A.2)$$

The corresponding amplification factor (6.1) for algorithm (3.20), (6.10)-(6.11) becomes

$$G = \frac{\left(\frac{1}{6} + (1 - \theta)C\frac{1}{2}\right)e^{-im} + \frac{4}{6} + \left(\frac{1}{6} - (1 - \theta)C\frac{1}{2}\right)e^{im}}{\left(\frac{1}{6} - \theta C\frac{1}{2}\right)e^{-im} + \frac{4}{6} + \left(\frac{1}{6} + \theta C\frac{1}{2}\right)e^{im}} \quad (A.3)$$

In general, for linear basis FE and FD, via (6.2), equation (6.10) can be written as

$$f(z) = \frac{a_1 + a_2 z + a_3 z^2}{b_1 + b_2 z + b_3 z^2} \quad (A.4)$$

where, a_i and b_i , $1 \leq i$, are approximation coefficients and $z = e^{im}$.

DEFINITION : If $p(x) = \sum_{i \geq k} c_i x^i$ is a Laurent series (*i.e.* k could be a negative integer) which is not identically zero then, by the *order* of $p(x)$ we mean the *least* index i such that $c_i \neq 0$. We denote the order of $p(x)$ by $\mathcal{O}(p(x))$. For example, if

$p(x) = -3x^3 + x^4 - x^5 + \dots$ then $\mathcal{O}(p(x)) = 3$, if $p(x) = 5x^{-2} + 1 + x^7 + \dots$ then $\mathcal{O}(p(x)) = -2$, if $p(x) = 1 - x + x^5 + \dots$ then $\mathcal{O}(p(x)) = 0$ etc.

If $\mathcal{O}(p(x)) = r$ then one can write $p(x) = x^r q(x)$, where $q(x)$ is a power-series (*i.e.* nonnegative components only) of order zero. For example $p(x) = x^{-2}(5 + x^2 + x^9 + \dots)$ in the second example above.

If $q(x)$ is a power-series of order zero, then its reciprocal is also a power-series of order zero. Therefore, if $p(x)$ is of order r then its reciprocal is $x^{-r} q^*(x)$, where $q^*(x)$ denotes the reciprocal of $q(x)$. Also, if $\sigma(x)$ is another Laurent series of order s , then the order of $p(x)\sigma(x)$ is exactly $r + s$, and the order of $p(x)/\sigma(x)$ is exactly $r - s$.

Note that if the Taylor expansions of $f(e^{im})$ in (A.4) and e^{imC} match up to (and including) the $(r-1)$ th power of x then, the order of $e^{imC} - f(e^{im})$ will be $\geq r$ (Mulay, 1993).

THEOREM : Assume that $C \neq 0, \pm 1, \pm 2$. Then, for (A.4) there does not exist any value of a_j, b_j , such that $\mathcal{O}(e^{imC} - f(e^{im})) \geq 6$. In other words, for any choice of function $f(z)$ we must have $\mathcal{O}(e^{imC} - f(e^{im})) \leq 5$.

PROOF : We have

$$\begin{aligned} e^{imC} - f(e^{im}) &= \frac{b_1 e^{imC} + b_2 e^{im(C+1)} + b_3 e^{im(C+2)} - a_1 - a_2 e^{im} - a_3 e^{i2m}}{b_1 + b_2 e^{im} + b_3 e^{i2m}} \\ &= \frac{N(m)}{D(m)} \end{aligned} \quad (\text{A.5})$$

where $N(m)$ and $D(m)$ denotes the numerator and the denominator. Clearly both $N(m)$ and $D(m)$ are power series, in particular, $\mathcal{O}(D(m)) \geq 0$. Since $\mathcal{O}(e^{imC} - f(e^{im})) = \mathcal{O}(N(m)) - \mathcal{O}(D(m))$, it follows that $\mathcal{O}(e^{imC} - f(e^{im})) \leq \mathcal{O}(N(m))$.

Also, we can see that

$$N(x) = (a_1 + a_2 + a_3 - b_1 - b_2 - b_3) + \sum_{n=1}^{\infty} \frac{i^n}{n!} u_n m^n \quad (\text{A.6})$$

where $u_n = a_1 C^n + a_2 (C+1)^n + a_3 (C+2)^n - b_2 - 2^n b_3$.

Now suppose that $\mathcal{O}(N(x)) \geq 6$. Then, we must have $u_1 = \dots = u_5 = 0$, as well as $(a_1 + a_2 + a_3 - b_1 - b_2 - b_3) = 0$. Now, $u_1 = \dots = u_5 = 0$ yields five homogeneous linear equations in a_1, a_2, a_3, b_2, b_3 . The determinant of this system can be easily computed. Upto a sign, via (A.1), it turns out to be $4(C-2)(C-1)^2 C^3 (C+1)^2 (C+2)$. Since, by assumption $C \neq 0, \pm 1, \pm 2$, this determinant is nonzero, forcing $a_1 = a_2 = a_3 = b_2 = b_3 = 0$. But then $b_1 = 0$ too. This is obviously absurd. In conclusion, we must always have $\mathcal{O}(N(x)) \leq 5$. Consequently, for (A.4), $\mathcal{O}(e^{imC} - f(e^{im}))$ can never exceed 5. $\square$

QED

For the GMP method (6.2)-(6.6), equation (A.4) for algorithm (6.12), is of the form

$$\mathcal{F}(z) = \frac{a_1 + a_2 z + \dots + a_P z^{P-1}}{b_1 + b_2 z + \dots + b_P z^{P-1}} \quad (\text{A.7})$$

where $P-3 = \max \text{degree}(\alpha, \beta, \gamma, \delta)$. Equation (6.10) directly relates P to the width of the computational stencil. For example, $\gamma = 1$, $\beta = 1$, and $\alpha = \delta = 0$, (A.7) takes the form

$$\mathcal{F}(z) = \frac{a_1 + a_2 z + a_3 z^2 + a_4 z^3 + a_5 z^4}{b_1 + b_2 z + b_3 z^2 + b_4 z^3 + b_5 z^4} \quad (\text{A.8})$$

which is equivalent to the $P = 4$ pentadiagonal (quadratic FE basis, or five point centered FD) form. Again, for $\gamma = -(1+m+m^2)$, $\beta = (1+m+m^2)$ is equivalent to the $P = 8$, 9-wide diagonal (quartic FE basis, or nine point centered FD) algorithm

$P = 4$, for septadiagonal $P = 6$, and so on. It can be also be shown by induction that

$$\mathcal{O}(e^{imC} - \mathcal{F}(e^{im})) \leq (2P - 1) \quad (\text{A.9})$$

Appendix B

For $\beta = 2 = \gamma$, and $\Lambda_1 : la1, \Lambda_2 : la2, \Psi_1 : ps1, \Psi_2 : ps2, \Gamma_1 : ga1, \Gamma_2 : ga2, \Upsilon_1 : up1, \Upsilon_2 : up2, a_1 : a, a_2 : b$, in (27), this appendix, generated using MacsymaTM (1988) symbolic logic program, verifies the analytical process leading to the determination of the algorithm order-of-accuracy.

Note: In MacsymaTM, %i = $\sqrt{-1}$ and %e(x) = exp (x).

```
(d1) sys$user6:[roy]gmp.general;1
(c2) z:%e^(m%i)$
(c3) x1:z^-1$
(c4) x2:z$
(c5) theta:1/2$
(c6) la1:t1*(1+a*z^-1+b*z^-2)$
(c7) ps1:t2*(1+a*z^-1+b*z^-2)$
(c8) ga1:t3*(1+a*z^-1+b*z^-2)$
(c9) up1:t4*(1+a*z^-1+b*z^-2)$
(c10) la2:t5*(1+a*z^-1+b*z^-2)$
(c11) ps2:t6*(1+a*z^-1+b*z^-2)$
(c12) ga2:t7*(1+a*z^-1+b*z^-2)$
(c13) up2:t8*(1+a*z^-1+b*z^-2)$
(c14) ga1:-(up1+la1+ps1)$
(c15) ga2:-(up2+la2+ps2)$
(c16) ght:((1-ga1)*x1+(4-la1-up1)+(1-ps1)*x2)/6$
(c17) ghx:((1-ps2)*x2-(la2+up2)-(1+ga2)*x1)/2$
(c18) ghn:ght-(1-theta)*c*ghx$
(c19) ghd:ght+theta*c*ghx$
(c20) gh:ratsimp(ghn/ghd)$
(c21) th:taylor(gh,m,0,6)$
(c23) ta:taylor(z^-c,m,0,9)$
(c24) term1:coeff(th,m,1);
(d24)/r/ ((%i b + %i a + %i) c t8 + (2 %i b + 2 %i a + 2 %i) c t6
          + (%i b + %i a + %i) c t5 - 2 %i c)/2
(c25) ta1:coeff(ta,m,1);
(d25)/r/ - %i c
(c26) solve([term1-ta1],[t6]);
(d26) [t6 = - (t8 + t5) / 2]
```

```

(c27) t6:rhs(%[1])$
(c28) th:ev(th,eval)$
(c29) term1:coeff(th,m,1);
(d29)/r/

```

$$- \frac{1}{2} c$$

Note: "gh" in (c21) is the same as G in (24). "term1" in (c29) is A_1 in (6.12), "term2" in (c30) is A_2 in (6.12).

```

(c30) term2:coeff(th,m,2);
(d30)/r/ ((3 b + 3 a + 3) c t8 + (27 b + 15 a + 3) c t5 + (2 b + 2 a + 2) c t4
+ (4 b + 4 a + 4) c t2 + (2 b + 2 a + 2) c t1 - 6 c )/12
(c31) ta2:coeff(ta,m,2);
(d31)/r/
(c32) solve([term2-ta2],[t5]);
(d32) [t5 = - ((3 b + 3 a + 3) t8 + (2 b + 2 a + 2) t4 + (4 b + 4 a + 4) t2
+ (2 b + 2 a + 2) t1)/(27 b + 15 a + 3)]
(c33) t5:rhs(%[1])$
(c34) th:ev(th,eval)$
(c35) term2:coeff(th,m,2);
(d35)/r/

```

$$- \frac{c}{2}$$

Note: "term3" in (c37) is A_3 in (6.12), "term4" in (c42) is A_4 in (6.12).

```

(c36) term3:coeff(th,m,3);
(d36)/r/ ((60 %i b + (66 %i a + 12 %i) b + 18 %i a + 6 %i a) c t8
+ (31 %i b + (30 %i a - 2 %i) b + 7 %i a - 2 %i a - %i) c t4
+ (80 %i b + (88 %i a + 16 %i) b + 24 %i a + 8 %i a) c t2
+ (- 41 %i b + (- 46 %i a - 10 %i) b - 13 %i a - 6 %i a - %i) c t1
+ (27 %i b + 15 %i a + 3 %i) c )/(108 b + 60 a + 12)
(c37) ta3:coeff(ta,m,3);
(d37)/r/
(c38) solve([term3-ta3],[t8]);

```

$$\frac{\frac{1}{2} c}{6}$$

```

(d38) [t8 = - ((31 b2 + (30 a - 2) b + 7 a2 - 2 a - 1) t4
+ (80 b2 + (88 a + 16) b + 24 a2 + 8 a) t2
+ (- 41 b2 + (- 46 a - 10) b - 13 a2 - 6 a - 1) t1 + (9 b + 5 a + 1) c2)
/(60 b2 + (66 a + 12) b + 18 a2 + 6 a)]

(c39) t8:rhs(%[1])$
(c40) th:ev(th,eval)$
(c41) term3:coeff(th,m,3);

                                     3
                                     %i c
(d41)/r/ -----
                                     6

(c42) term4:coeff(th,m,4);

(d42)/r/ (((10 b3 + (21 a + 12) b2 + (14 a2 + 14 a + 2) b + 3 a3 + 4 a2 + a) c
+ (80 b3 + 126 a b2 + (70 a2 + 16 a + 16) b + 12 a3 + 2 a2 + 2 a) c) t4
+ ((20 b3 + (42 a + 24) b2 + (28 a2 + 28 a + 4) b + 6 a3 + 8 a2 + 2 a) c
+ (- 20 b3 + (- 42 a - 24) b2 + (- 28 a2 - 28 a - 4) b - 6 a3 - 8 a2 - 2 a) c)
t2 + (((10 b3 + (21 a + 12) b2 + (14 a2 + 14 a + 2) b + 3 a3 + 4 a2 + a) c
+ (- 400 b3 + (- 498 a + 96) b2 + (- 218 a2 + 64 a + 16) b - 36 a3 + 2 a
+ 2 a) c) t1 + (30 b2 + (33 a + 6) b + 9 a3 + 3 a) c
+ (30 b2 + (15 a - 18) b + 3 a3 - 3 a) c )
/(720 b2 + (792 a + 144) b + 216 a2 + 72 a)
(c43) ta4:coeff(ta,m,4);

                                     4
                                     c
(d43)/r/ -----
                                     24

(c46) solve([term4-ta4],[t4]);

(d46) [t4 = - (((20 b3 + (42 a + 24) b2 + (28 a2 + 28 a + 4) b + 6 a3 + 8 a2
+ 2 a) c - 20 b3 + (- 42 a - 24) b2 + (- 28 a2 - 28 a - 4) b - 6 a3 - 8 a2
- 2 a) t2 + (((10 b3 + (21 a + 12) b2 + (14 a2 + 14 a + 2) b + 3 a3 + 4 a2 + a)
c - 400 b2 + (96 - 498 a) b + (- 218 a2 + 64 a + 16) b - 36 a3 + 2 a + 2 a)

```

```

      2      2      2
t1 + (30 b + (15 a - 18) b + 3 a - 3 a) c )
      3      2      2      3      2      2      3
/((10 b + (21 a + 12) b + (14 a + 14 a + 2) b + 3 a + 4 a + a) c + 80 b
      2      2      3      2
+ 126 a b + (70 a + 16 a + 16) b + 12 a + 2 a + 2 a)]
(c47) t4:rhs(%[1])$
(c48) th:ev(th,eval)$
(c49) term4:coeff(th,m,4);

```

```

      4
      c
(d49)/r/  --
      24

```

Note: "term5" in (c60) is A_5 in (6.12) and "term6" in (c56) is A_6 in (6.12)

```

(c50) term5:coeff(th,m,5);
      4      3
(d50)/r/ (((50 %i b + (155 %i a + 110 %i) b
      2      2      3      2
+ (175 %i a + 235 %i a + 70 %i) b + (85 %i a + 160 %i a + 85 %i a + 10 %i)
      4      3      2      5
b + 15 %i a + 35 %i a + 25 %i a + 5 %i a) c
      4      3      2      2
+ (950 %i b + (1955 %i a - 430 %i) b + (1675 %i a - 5 %i a + 610 %i) b
      3      2      4      3      2
+ (625 %i a + 100 %i a + 445 %i a + 70 %i) b + 75 %i a - 25 %i a + 25 %i a
      3      4      3
+ 5 %i a) c + (- 1000 %i b + (- 2110 %i a + 320 %i) b
      2      2
+ (- 1850 %i a - 230 %i a - 680 %i) b
      3      2      4      3
+ (- 710 %i a - 260 %i a - 530 %i a - 80 %i) b - 90 %i a - 10 %i a
      2      4      3
- 50 %i a - 10 %i a) c) t2 + ((- 200 %i b + (- 520 %i a - 240 %i) b
      2      3      2
+ (- 490 %i a - 400 %i a - 40 %i) b + (- 200 %i a - 220 %i a - 40 %i a) b
      4      3      2      5      4      3
- 30 %i a - 40 %i a - 10 %i a) c + (2200 %i b + (3080 %i a - 1680 %i) b
      2      2      3      2
+ (1550 %i a - 2080 %i a - 40 %i) b + (340 %i a - 820 %i a - 40 %i a) b
      4      3      2      3      4
+ 30 %i a - 100 %i a - 10 %i a) c + (16000 %i b
      3      2      2
+ (20840 %i a - 12480 %i) b + (11540 %i a - 9760 %i a + 2240 %i) b
      3      2      4      3      2
+ (3100 %i a - 2920 %i a + 440 %i a) b + 360 %i a - 220 %i a + 20 %i a) c)

```

```

t1 + (- 50 %i b + (105 %i a + 420 %i) b + (65 %i a + 260 %i a - 10 %i) b
      2          5          3          2
+ 25 %i a - 5 %i a) c + (520 %i b + (1272 %i a + 1164 %i) b
      2          3          2          3
+ (443 %i a + 428 %i a - 76 %i) b + 21 %i a + 43 %i a - 8 %i a) c
      3          2          2          3
+ (160 %i b + 252 %i a b + (140 %i a + 32 %i a + 32 %i) b + 24 %i a
      2          3          2
+ 4 %i a + 4 %i a) c)/((3600 b + (7560 a + 4320) b
      2          3          2          2          3
+ (5040 a + 5040 a + 720) b + 1080 a + 1440 a + 360 a) c + 28800 b
      2          2          3          2
+ 45360 a b + (25200 a + 5760 a + 5760) b + 4320 a + 720 a + 720 a)
(c51) ta5:coeff(ta,m,5);

```

```

      5
      %i c
(d51)/r/ - -----
      120
(c52) solve([term5-ta5],[t1]);
(d52) [t1 = (((50 b + (155 a + 110) b + (175 a + 235 a + 70) b
      4          3          2          2
+ (85 a + 160 a + 85 a + 10) b + 15 a + 35 a + 25 a + 5 a) c
      4          3          2          2          4
+ (950 b + (1955 a - 430) b + (1675 a - 5 a + 610) b
      3          2          4          3          2
+ (625 a + 100 a + 445 a + 70) b + 75 a - 25 a + 25 a + 5 a) c - 1000 b
      3          2          2
+ (320 - 2110 a) b + (- 1850 a - 230 a - 680) b
      3          2          4          3          2
+ (- 710 a - 260 a - 530 a - 80) b - 90 a - 10 a - 50 a - 10 a) t2
      3          2          2          3          2          6
+ (30 b + (63 a + 36) b + (42 a + 42 a + 6) b + 9 a + 12 a + 3 a) c
      3          2          2          3          2          4
+ (190 b + (483 a + 420) b + (275 a + 308 a + 38) b + 36 a + 31 a + a) c
      3          2          2          3          2
+ (520 b + (1272 a + 1164) b + (443 a + 428 a - 76) b + 21 a + 43 a - 8 a)
      2          3          2          2          3          2
c + 160 b + 252 a b + (140 a + 32 a + 32) b + 24 a + 4 a + 4 a)
      4          3          2          2
/((200 b + (520 a + 240) b + (490 a + 400 a + 40) b
      3          2          4          3          2          4
+ (200 a + 220 a + 40 a) b + 30 a + 40 a + 10 a) c
      4          3          2          2
+ (- 2200 b + (1680 - 3080 a) b + (- 1550 a + 2080 a + 40) b
      3          2          4          3          2          2          4
+ (- 340 a + 820 a + 40 a) b - 30 a + 100 a + 10 a) c - 16000 b

```

```

      3      2      2
+ (12480 - 20840 a) b + (- 11540 a + 9760 a - 2240) b
      3      2      4      3      2
+ (- 3100 a + 2920 a - 440 a) b - 360 a + 220 a - 20 a )]
(c53) t1:rhs(%[1])$
(c54) th:ev(th,eval)$
(c55) term5:coeff(th,m,5);

```

```

      5
      %i c
(d55)/r/ - -----
      120

```

Finally, in (d62) the approximation solution G of (6.12), expanded in Laurent series up to and including the sixth term, for particular choices of t_i matches exactly up to the sixth term of G_a in (1.11). Hence, the GMP algorithm for $\beta = 2 = \gamma$ is *at least* sixth order accurate for any $a_1, a_2 > 0$.

```

(c56) term6:coeff(th,m,6);
(d56)/r/ (((500 b + (1600 a + 700) b + (2005 a + 1620 a + 220) b
      3      2      2      4      3      2
+ (1235 a + 1395 a + 360 a + 20) b + (375 a + 530 a + 195 a + 20 a) b
      5      4      3      2      7      5      4
+ 45 a + 75 a + 35 a + 5 a ) c + (- 21000 b + (- 51000 a - 7800) b
      2      3      3      2      2
+ (- 48030 a - 6480 a + 10920) b + (- 23250 a - 4770 a + 7560 a - 2280) b
      4      3      2      5      4      3      2
+ (- 5850 a - 2100 a + 2430 a - 480 a) b - 630 a - 450 a + 150 a - 30 a )
      5      5      4      2      3
c + (220500 b + (349200 a - 252900) b + (268245 a - 230220 a + 102780) b
      3      2      2
+ (110115 a - 99045 a + 57240 a + 180) b
      4      3      2      5      4      3
+ (23175 a - 22230 a + 7155 a + 180 a) b + 2205 a - 1125 a + 315 a
      2      3      5      4
+ 45 a ) c + (- 200000 b + (- 299800 a + 260000) b
      2      3      3      2
+ (- 222220 a + 235080 a - 113920) b + (- 88100 a + 102420 a - 65160 a
      2      4      3      2      5      4
+ 2080) b + (- 17700 a + 23800 a - 9780 a + 280 a) b - 1620 a + 1500 a
      3      2      4      3
- 500 a - 20 a ) c) t2 + (- 100 b + (- 260 a - 120) b
      2      2      3      2      4      3
+ (- 245 a - 200 a - 20) b + (- 100 a - 110 a - 20 a) b - 15 a - 20 a
      2      10      4      3      2      2
- 5 a ) c + (120 b + (276 a + 96) b + (228 a + 120 a - 24) b
      3      2      4      3      9
+ (84 a + 60 a - 12 a) b + 12 a + 12 a ) c

```

```

      4      3      2      2
+ (1100 b + (1540 a - 840) b + (775 a - 1040 a - 20) b
      3      2      4      3      2      8
+ (170 a - 410 a - 20 a) b + 15 a - 50 a - 5 a ) c
      4      3      2      2
+ (680 b + (1210 a + 1560) b + (145 a + 280 a - 1040) b
      3      2      4      3      2      7
+ (- 175 a + 40 a - 170 a) b - 60 a - 35 a - 5 a ) c
      4      3      2      2
+ (8000 b + (10420 a - 6240) b + (5770 a - 4880 a + 1120) b
      3      2      4      3      2      6
+ (1550 a - 1460 a + 220 a) b + 180 a - 110 a + 10 a ) c
      4      3      2      2
+ (- 37320 b + (- 23472 a + 26808) b + (- 5856 a + 9120 a + 768) b
      3      2      4      3      2      5
+ (- 738 a - 690 a + 264 a) b + 66 a - 84 a + 30 a ) c
      4      3      2      2
+ (- 68280 b + (90 a + 77040) b + (3255 a + 10320 a - 1560) b
      3      2      4      3      2      3
+ (225 a + 1110 a - 330 a) b - 90 a + 135 a - 45 a ) c
      4      3      2      2
+ (- 3200 b + (296 a + 2496) b + (2228 a + 1760 a + 1856) b
      3      2      4      3      2
+ (604 a - 520 a + 248 a) b + 72 a - 28 a + 20 a ) c
      4      3      2      2
/((72000 b + (187200 a + 86400) b + (176400 a + 144000 a + 14400) b
      3      2      4      3      2      4
+ (72000 a + 79200 a + 14400 a) b + 10800 a + 14400 a + 3600 a ) c
      4      3
+ (- 792000 b + (- 1108800 a + 604800) b
      2      2      3      2
+ (- 558000 a + 748800 a + 14400) b + (- 122400 a + 295200 a + 14400 a) b
      4      3      2      2      4      3
- 10800 a + 36000 a + 3600 a ) c - 5760000 b + (- 7502400 a + 4492800) b
      2      2
+ (- 4154400 a + 3513600 a - 806400) b
      3      2      4      3      2
+ (- 1116000 a + 1051200 a - 158400 a) b - 129600 a + 79200 a - 7200 a )
(c57) ta6:=coeff(ta,m,6);

      6
      c
(d57)/r/ - ----
      720

(c60) solve([term6-ta6],[t2]);

      4      3      2      2
(d60) [t2 = - ((120 b + (276 a + 96) b + (228 a + 120 a - 24) b
      3      2      4      3      8
+ (84 a + 60 a - 12 a) b + 12 a + 12 a ) c

```


$$\begin{aligned}
& + (680 b^4 + (1210 a + 1560) b^3 + (145 a^2 + 280 a - 1040) b^2 \\
& + (-175 a^3 + 40 a^2 - 170 a) b - 60 a^4 - 35 a^3 - 5 a^2) c^6 \\
& + (-37320 b^4 + (26808 - 23472 a) b^3 + (-5856 a + 9120 a + 768) b^2 \\
& + (-738 a^3 - 690 a^2 + 264 a) b + 66 a^4 - 84 a^3 + 30 a^2) c^4 \\
& + (-68280 b^4 + (90 a + 77040) b^3 + (3255 a^2 + 10320 a - 1560) b^2 \\
& + (225 a^3 + 1110 a^2 - 330 a) b - 90 a^4 + 135 a^3 - 45 a^2) c^2 - 3200 b^4 \\
& + (296 a^3 + 2496) b^2 + (2228 a^2 + 1760 a + 1856) b \\
& + (604 a^3 - 520 a^2 + 248 a) b + 72 a^4 - 28 a^3 + 20 a^2) \\
& / ((500 b^5 + (1600 a + 700) b^4 + (2005 a^2 + 1620 a + 220) b^3 \\
& + (1235 a^3 + 1395 a^2 + 360 a + 20) b^2 + (375 a^4 + 530 a^3 + 195 a^2 + 20 a) b \\
& + 45 a^5 + 75 a^4 + 35 a^3 + 5 a^2) c^6 + (-21000 b^5 + (-51000 a - 7800) b^4 \\
& + (-48030 a^2 - 6480 a + 10920) b^3 + (-23250 a^3 - 4770 a^2 + 7560 a - 2280) b^2 \\
& + (-5850 a^4 - 2100 a^3 + 2430 a^2 - 480 a) b - 630 a^5 - 450 a^4 + 150 a^3 - 30 a^2) \\
& c^4 + (220500 b^3 + (349200 a - 252900) b^2 + (268245 a - 230220 a + 102780) b \\
& + (110115 a - 99045 a + 57240 a + 180) b \\
& + (23175 a^4 - 22230 a^3 + 7155 a^2 + 180 a) b + 2205 a^5 - 1125 a^4 + 315 a^3 \\
& + 45 a^2) c^2 - 200000 b^2 + (260000 - 299800 a) b \\
& + (-222220 a^2 + 235080 a - 113920) b^3 + (-88100 a^3 + 102420 a^2 - 65160 a \\
& + 2080) b^4 + (-17700 a^2 + 23800 a - 9780 a + 280 a) b - 1620 a^5 + 1500 a^4 \\
& - 500 a^3 - 20 a^2)] \\
(c61) t2:rhs(%[1])$ \\
(c62) th:ev(th,eval); \\
(d62)/r/
\end{aligned}$$

$$\frac{c^6 m^6 + 6 i c^5 m^5 - 30 c^4 m^4 - 120 i c^3 m^3 + 360 c^2 m^2 + 720 i c m - 720}{720}$$

720

(c63) term6:coeff(th,m,6);

6

c

(d63)/r/

- ---

720

(c65) quit();

Appendix C

Via weak statement algorithm (6.2), the θ -implicit relation for (8.37) is

$$([M] + \theta \Delta t [U]) \{Q^{n+1}\} - ([M] + \theta \Delta t [U]) \{Q^n\} + \Delta t [U] \{Q^n\} = \{0\} \quad (C.1)$$

In one dimensional finite element form, for $C = 1$ and $\theta = 1/2$, (C.1) yields

$$\left(\frac{1}{4} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + \frac{1}{4} \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \right) \{Q^{n+1}\} = \left(\frac{1}{4} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} - \frac{1}{4} \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \right) \{Q^n\} \quad (C.2)$$

Hence, the global assembly form of (C.2) is

$$\begin{bmatrix} 0 & 1 & & & & & \\ 0 & 1 & 1 & & & & \\ & 0 & 1 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 0 & 1 & 1 & \\ & & & & 0 & 1 \end{bmatrix} \begin{Bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ \vdots \\ Q_M \\ Q_{M+1} \end{Bmatrix}^{n+1} = \begin{bmatrix} 1 & 0 & & & & & \\ 1 & 1 & 0 & & & & \\ & 1 & 1 & 0 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & 1 & 0 & \\ & & & & 1 & 0 \end{bmatrix} \begin{Bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ \vdots \\ Q_M \\ Q_{M+1} \end{Bmatrix}^n \quad (C.3)$$

It is easy to see that the solution of (C.3) is

$$Q_j^{n+1} = Q_{j-1}^n, \text{ i.e. if } \{Q^n\} = \{0, a, b, 0, 0, \dots, 0, c, 0\}^T \quad (C.4)$$

$$\text{then } \{Q^{n+1}\} = \{0, 0, a, b, 0, \dots, 0, 0, c\}^T$$

which is exact for $C = 1$.

Similarly, for $C = 2$, the exact one dimensional solution is

$$Q_j^{n+1} = Q_{j-2}^n, \text{ i.e. if } \{Q^n\} = \{0, a, b, 0, 0, \dots, 0, c, 0, 0\}^T \quad (\text{C.5})$$

$$\text{then } \{Q^{n+1}\} = \{0, 0, 0, a, b, \dots, 0, 0, 0, c\}^T$$

This is not possible to extract from the tridiagonal matrix form (C.3), since the residual matrix (right hand side) has to be at least four-wide, i.e.,

$$\begin{bmatrix} 0 & 1 & & & \\ 0 & 1 & 1 & & \\ & 0 & 1 & 1 & \\ & & \ddots & \ddots & \ddots \\ & & & 0 & 1 & 1 \\ & & & & 0 & 1 \end{bmatrix} \begin{Bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ \vdots \\ Q_M \\ Q_{M+1} \end{Bmatrix}^{n+1} = \begin{bmatrix} 1 & 0 & 0 & & \\ 1 & 1 & 0 & 0 & \\ & 1 & 1 & 0 & 0 \\ & & \ddots & \ddots & \ddots \\ & & & 1 & 1 & 0 & 0 \\ & & & & 1 & 0 & 0 \end{bmatrix} \begin{Bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ \vdots \\ Q_M \\ Q_{M+1} \end{Bmatrix}^n \quad (\text{C.6})$$

such that (C.5) can be satisfied. This is impossible to achieve within the context of linear basis (tridiagonal) matrix equations.

Appendix D

```

                                     h
      *Details of Burgers Equation solution, via SGM *
*****
* for 21 nodes, burgers solution with re=10e5, non-uniform mesh, mar 7,94
*****

$  INITIAL CONDITION
$  0.00000E+00 U1 (      1)      21      0      1      264      FSET 1
1.000000E+00 1.000000E+00 1.000000E+00 1.000000E+00 1.000000E+00
1.000000E+00 1.000000E+00 1.000000E+00 7.500000E-01 5.000000E-01
0.000000E+00-5.000000E-01-7.500000E-01-1.000000E+00-1.000000E+00
-1.000000E+00-1.000000E+00-1.000000E+00-1.000000E+00-1.000000E+00
-1.000000E+00
$  MESH
0.000000E+00 2.409412E-01 3.614118E-01 4.216471E-01 4.517647E-01
4.668235E-01 4.743529E-01 4.781176E-01 4.800000E-01 4.900000E-01
5.000000E-01 5.100000E-01 5.200000E-01 5.218824E-01 5.256471E-01
5.331765E-01 5.482353E-01 5.783529E-01 6.385882E-01 7.590588E-01
1.000000E+00

-----
* iteration history (linear convergence due to uncoupled templates)
-----

1      1      2      1      3      1  U1      SOLVER = >GAUSS-ELIM< INT =      1
1  1 MAX DQ  -1.35714E+00
1  2 MAX DQ   9.89012E-01
1  3 MAX DQ  -2.12825E-01
1  4 MAX DQ   1.16888E-01
1  5 MAX DQ   5.49010E-02
1  6 MAX DQ  -2.81900E-02
1  7 MAX DQ   1.38945E-02
1  8 MAX DQ   6.99527E-03
1  9 MAX DQ  -3.48537E-03
1 10 MAX DQ   1.74572E-03
1 11 MAX DQ   8.72098E-04
1 12 MAX DQ   4.36239E-04
1 13 MAX DQ  -2.18072E-04
1 14 MAX DQ  -1.09048E-04
1 15 MAX DQ  -5.45210E-05
1 16 MAX DQ   2.72619E-05
1 17 MAX DQ  -1.36306E-05
1 18 MAX DQ   6.81539E-06

```

```

1 19 MAX DQ -3.40761E-06
1 20 MAX DQ -1.70428E-06
1 21 MAX DQ 8.51909E-07
1 22 MAX DQ 4.26360E-07
1 23 MAX DQ -2.12987E-07
1 24 MAX DQ 1.06542E-07
1 25 MAX DQ 5.37598E-08

```

* SOLUTION OUTPUT AS NODAL DATA

```

$ Iteration No.: 25
$ U1 Solution
1.000000E+00 1.000000E+00 1.000000E+00 1.000000E+00 1.000000E+00
1.000000E+00 1.000000E+00 1.000000E+00 1.000000E+00 1.000000E+00
3.257469E-09-1.000000E+00-1.000000E+00-1.000000E+00-1.000000E+00
-1.000000E+00-1.000000E+00-1.000000E+00-1.000000E+00-1.000000E+00
-1.000000E+00
$ RSGM Distribution
8.031373E+03 6.023529E+03 3.011765E+03 1.505882E+03 7.529412E+02
3.764706E+02 1.882353E+02 9.411765E+01 1.980392E+02 3.333333E+02
3.333333E+02 3.333333E+02 1.980392E+02 9.411765E+01 1.882353E+02
3.764706E+02 7.529412E+02 1.505882E+03 3.011765E+03 6.023529E+03
8.031373E+03

```

* SECTION OF THE FORTRAN

```

C:: ITHOOK INS.F
C:
C>>> ITHOOK USER BEGIN
C:
croy for variable csgm
c    REYNLD=1.0/RZ(ICOEFN+04)
    CSGM=RZ(ICOEFN+08)
    MX1=LVADD(01)
    MU1=LVADD(04)
    MUMG=LVADD(06)
    MRGM=LVADD(10)
    MDET=LVADD(11)
    MDTJ=LVADD(12)
    IF(CSGM.NE.1.0)THEN
    DETUNI=(RZ(MX1+NNODE)-RZ(MX1+1))/(NNODE-1)
    INODEE=NNODE/2
    INODEE=INODEE*2
    DO 10 I=1,NNODE
    RZ(MUMG+I)=ABS(RZ(MU1+I))

```

```

      IF(NNODE.EQ.INODEE) RZ(MDET+I)=RZ(MDTJ+I)
      IF(NNODE.NE.INODEE) RZ(MDET+I)=RZ(MDTJ+I)*2.
      IF(RZ(MRGM+I).LT.9.9E-1)
        RZ(MRGM+I)= 0.5*(RZ(MRGM+I-1)+RZ(MRGM+I+1))
10    CONTINUE
      ENDIF

```

```

-----
TITLE      **** INPUT FILE FOR AKCESS.* ****
1D BURGERS SOLUTION SROY 3/94
INTEGRATION FACTORS

```

```

      :
      1.E-09          $ PROBLEM_CONVERGENCE_CRITERIA
      0.              $ MAXIMUM_CHANGE_IN_Q__(DQ)
      1              $ MAXIMUM_NUMBER_OF_STEPS
      25             $ MAXIMUM_NUMBER_OF_ITERATIONS_PER_STEP

```

```

      :
GLOBAL SCALARS

```

```

      [ 0. ]          $ TDELTA

```

```

CELL SCALARS

```

```

      1 1            #
[ 1., -1., 0.00001, 0. $ ONE, -ONE, DIFF, DUM1
  0., 0., 0., 0.0, 0.0 $ -SCOF,NROB,CSGM,BETA,GAMA
  0., 0., 0., 0., 0.   $ HALF,SIXTH,DELTA,DELTA2,DELTA2
  1E05, 0.33333 ] $ REY,THRD

```

```

      :
      :
BOUNDARY CONDITIONS

```

```

DIRI_Q 1 [ 1 21 ]    $ FIXEDSET 1 # Q ALL AROUND

```

```

INITIAL CONDITIONS

```

```

[21*0.]              $ RSGM    8
[8*1. 0.75 0.5 0. -0.5 -0.75 8*-1. ] $ U1    2
[21*0.]              $ U2    3
[21*0.]              $ UMAG    4

```

```

      :
      :
[21*0.]              $ RSGL 26
[21*0.]              $ U1L 27
[21*0.]              $ F1 28
[21*0.]              $ F2 29
[21*0.]              $ G1 30
[21*0.]              $ G2 31
[1(0. 8R0.5 0.48 4R1 0.52 8R2.0 1.)] $ X1 32
[21*1.]              $ ONES 33

```

```

      T T T

```

```

END

```

```

-----
* TEMPLATE FOR AKCESS
-----

```

```

:
:
BOUNDARY CONDITIONS

```

```

    U1
DIRI_Q  D
NEUM_Q  [-3 3]

```

```

TITLE          **** TEMPLATE TEMP.T.OP ****
      DISTRIBUTED SGM COMPUTATION F(UE,DETE,RE) 2/18/94

```

RESIDUALS

```

    RSGM  2  #  VARBL, SET NO., # OF TERMS --- SPATIAL SET (U)
    (-,REY,THRD)()(AJRO)(0;1)(A3000)(UMAG)

```

JACOBIANS

```

    RSGM  RSGM  2  1 T VARBL, VARDIF, SET, # OF TERMS, ALL DIRECTIONS
    ()()()(0;1)(A200)()

```

GROUP FREQUENCY

```

    1
SOLUTION TYPE
    Q
    GAUSS_ELIMINATION
    IMPLICIT_EULER

```

END

```

TITLE          **** TEMPLATE FILE TEMP.TWS123.ITER ****
      FE.1-2-3 ITERATIVE TWS ALGORITHM WITH SGM, (10/93)

```

RESIDUALS

```

    U1  2  #                                SPATIAL TERMS IN {R (Q)}E

    ()()(U1)(0;0)(A3001)(U1)
    +(BETA)()(0;-1)(A211)(U1)
    +(DIFF)()(RSGM)(0;-1)(A3011)(U1)
    +(-SCOF)()(0;1)(A200)(SORC)

```

```

    U1  3  #                                BOUNDARY TERMS IN {R (Q)}E

    (NROB)(NUSL)()(0;1)(A200)(U1)
    +(-,NROB)(NUSL)()(0;1)(A200)(QREF)

```

JACOBIANS


```

U1 U1 2 1 T          SPATIAL TERMS [DIFF]E + [CONV]E

  ( ) (U1) (0;0) (A3001) ( )
+ ( ) (U1) (0;0) (A3100) ( )
+ (BETA) ( ) (0;-1) (A211) ( )
+ (DIFF) ( ) (RSGM) (0;-1) (A3011) ( )

U1 U1 3 1 T          BOUNDARY TERMS [HBC]E

  (NROB) (NUSL) ( ) (0;1) (A200) ( )

GROUP FREQUENCY
  1
SOLUTION TYPE
  DELTA_Q
  GAUSS_ELIMINATION
  IMPLICIT_EULER

END

```

Appendix E

1D matrices, Macsyma (1988) output

Quadratic basis:

sroy 3/16/93

a200 : [m] - matrix, k2 (lagrange polynomial)

indefinite integral (mmatx,x)

$$\begin{aligned}
 & \left[\begin{array}{cccccc} 5 & & 4 & & 2 & 3 & & 3 & 2 & & 4 \end{array} \right] \\
 & \left[24 x^5 - 90 l e x^4 + 130 l e^2 x^3 - 90 l e^3 x^2 + 30 l e^4 x \right] \\
 & \left[\begin{array}{c} \hline 4 \\ 30 l e^4 \end{array} \right] \\
 \text{col 1} = & \left[\begin{array}{cccccc} 5 & & 4 & & 2 & 3 & & 3 & 2 \end{array} \right] \\
 & \left[24 x^5 - 75 l e x^4 + 80 l e^2 x^3 - 30 l e^3 x^2 \right] \\
 & \left[\begin{array}{c} \hline 4 \\ 15 l e^4 \end{array} \right] \\
 & \left[\begin{array}{cccccc} 5 & & 4 & & 2 & 3 & & 3 & 2 \end{array} \right] \\
 & \left[24 x^5 - 60 l e x^4 + 50 l e^2 x^3 - 15 l e^3 x^2 \right] \\
 & \left[\begin{array}{c} \hline 4 \\ 30 l e^4 \end{array} \right] \\
 & \left[\begin{array}{cccccc} 5 & & 4 & & 2 & 3 & & 3 & 2 \end{array} \right] \\
 & \left[24 x^5 - 75 l e x^4 + 80 l e^2 x^3 - 30 l e^3 x^2 \right] \\
 & \left[\begin{array}{c} \hline 4 \\ 15 l e^4 \end{array} \right] \\
 \text{col 2} = & \left[\begin{array}{cccccc} 5 & & 4 & & 2 & 3 \end{array} \right] \\
 & \left[4 (12 x^5 - 30 l e x^4 + 20 l e^2 x^3) \right] \\
 & \left[\begin{array}{c} \hline 4 \\ 15 l e^4 \end{array} \right] \\
 & \left[\begin{array}{cccccc} 5 & & 4 & & 2 & 3 \end{array} \right] \\
 & \left[24 x^5 - 45 l e x^4 + 20 l e^2 x^3 \right] \\
 & \left[\begin{array}{c} \hline 4 \\ 15 l e^4 \end{array} \right] \\
 & \left[\begin{array}{cccccc} 5 & & 4 & & 2 & 3 & & 3 & 2 \end{array} \right] \\
 & \left[24 x^5 - 60 l e x^4 + 50 l e^2 x^3 - 15 l e^3 x^2 \right] \\
 & \left[\begin{array}{c} \hline 4 \\ 30 l e^4 \end{array} \right]
 \end{aligned}$$

$$\text{col 3} = \begin{bmatrix} \frac{24x^5 - 45le x^4 + 20le^2 x^3}{48x^5 - 60le x^4 + 20le^2 x^3} \\ \frac{15le}{60le} \end{bmatrix}$$

definite integral (mmatx,x,0,le);

$$\begin{bmatrix} 2le & le & le \\ \frac{15}{le} & \frac{8le}{le} & \frac{30}{le} \\ 15 & 15 & 15 \\ le & le & 2le \\ - & - & - \\ 30 & 15 & 15 \end{bmatrix}$$

a211 : [D] - matrix, k2 (lagrange polynomial)

indefinite integral (kmatx,x)

$$\text{matrix} \left(\begin{bmatrix} 4 \left(\frac{x^3}{2le} - \frac{x^2}{le} + x \right) - 4 \left(x - \frac{x^2}{2le} \right) + x & 4 \left(\frac{x^3}{2le} - \frac{x^2}{le} + x \right) \\ 3le & 3le \end{bmatrix} + \begin{bmatrix} 2(4x^3 - 9le x^2 + 6le^2 x) & 64x^3 - 120le x^2 + 72le^2 x \\ 32x^3 - 48le x^2 + 18le^2 x & 64x^3 - 120le x^2 + 72le^2 x \end{bmatrix}, \begin{bmatrix} \frac{2(4x^3 - 9le x^2 + 6le^2 x)}{6le} & \frac{64x^3 - 120le x^2 + 72le^2 x}{6le} \end{bmatrix} \right)$$

$$\begin{aligned}
& 16 \left(\frac{x^3}{3 \text{ le}} - \frac{x^2}{2 \text{ le}} + x \right) \\
& - \frac{64 x^3 - 72 \text{ le } x^2 + 24 \text{ le}^2 x}{6 \text{ le}^4} + \frac{16 (2 x^3 - 3 \text{ le } x^2)}{6 \text{ le}^4} + \frac{16 x^3}{6 \text{ le}^4}, \\
& - \frac{32 x^3 - 48 \text{ le } x^2 + 18 \text{ le}^2 x}{6 \text{ le}^4}, \left[\frac{4 x^3 - 2 x^2}{6 \text{ le}^4} + x \right] \\
& - \frac{64 x^3 - 72 \text{ le } x^2 + 24 \text{ le}^2 x}{6 \text{ le}^4}, \frac{2 (4 x^3 - 3 \text{ le } x^2)}{3 \text{ le}^4} + \frac{4 x^3}{3 \text{ le}^4} + \frac{4 x^2}{3 \text{ le}^4}
\end{aligned}$$

definite integral (kmatx,x,0,le);

$$\begin{bmatrix}
7 & 8 & 1 \\
\frac{7}{3 \text{ le}} & -\frac{8}{3 \text{ le}} & \frac{1}{3 \text{ le}} \\
8 & 16 & 8 \\
-\frac{8}{3 \text{ le}} & \frac{16}{3 \text{ le}} & -\frac{8}{3 \text{ le}} \\
1 & 8 & 7 \\
\frac{1}{3 \text{ le}} & -\frac{8}{3 \text{ le}} & \frac{7}{3 \text{ le}}
\end{bmatrix}$$

$$\begin{aligned}
& [m] : [m] - [m]^{-1} [m] [m] \\
& \quad \quad \quad a \quad \quad b \quad \quad d \quad \quad c \\
& \quad \quad \quad 24 x^5 - 90 \text{ le } x^4 + 130 \text{ le}^2 x^3 - 90 \text{ le}^3 x^2 + 30 \text{ le}^4 x \\
& \text{matrix} \left(\frac{(24 x^5 - 75 \text{ le } x^4 + 80 \text{ le}^2 x^3 - 30 \text{ le}^2 x^2)}{60 \text{ le}^4 (12 x^4 - 30 \text{ le } x^5 + 20 \text{ le}^4 x^2)} \right)
\end{aligned}$$

$$\begin{aligned}
& \begin{array}{ccccccc} 5 & & 4 & & 2 & 3 & & 3 & 2 \\ 24 x^5 & - 60 l e x^4 & + 50 l e x^3 & - 15 l e x^2 \end{array} \\
& \hline
& \begin{array}{ccccccc} & & 4 & & & & & & \\ & & 30 l e & & & & & & \\ & 5 & & 4 & & 2 & 3 & & 3 & 2 & & 5 & & 4 & & 2 & 3 \\ (24 x^5 & - 75 l e x^4 & + 80 l e x^3 & - 30 l e x^2 & x) & (24 x^5 & - 45 l e x^4 & + 20 l e x^3 & x) \end{array} \\
& \hline
& \begin{array}{ccccccc} & & 4 & & 5 & & 4 & & 2 & 3 \\ & & 60 l e & (12 x^4 & - 30 l e x^3 & + 20 l e x^2 & x) \end{array} \\
& \begin{array}{ccccccc} 5 & & 4 & & 2 & 3 & & 3 & 2 \\ 24 x^5 & - 60 l e x^4 & + 50 l e x^3 & - 15 l e x^2 \end{array} \\
& \hline
& \begin{array}{ccccccc} & & 4 & & & & & & \\ & & 30 l e & & & & & & \\ & 5 & & 4 & & 2 & 3 & & 3 & 2 & & 5 & & 4 & & 2 & 3 \\ (24 x^5 & - 75 l e x^4 & + 80 l e x^3 & - 30 l e x^2 & x) & (24 x^5 & - 45 l e x^4 & + 20 l e x^3 & x) \end{array} \\
& \hline
& \begin{array}{ccccccc} & & 4 & & 5 & & 4 & & 2 & 3 \\ & & 60 l e & (12 x^4 & - 30 l e x^3 & + 20 l e x^2 & x) \end{array} \\
& \begin{array}{ccccccc} 5 & & 4 & & 2 & 3 & & 5 & & 4 & & 2 & 3 & 2 \\ 48 x^5 & - 60 l e x^4 & + 20 l e x^3 & & (24 x^5 & - 45 l e x^4 & + 20 l e x^3 & x) \end{array} \\
& \hline
& \begin{array}{ccccccc} & & 4 & & 4 & & 5 & & 4 & & 2 & 3 \\ & & 60 l e & & 60 l e & (12 x^4 & - 30 l e x^3 & + 20 l e x^2 & x) \end{array}
\end{aligned}$$

integrate([m] ,x,0,le)\$

$$\begin{bmatrix} l e & l e \\ -- & -- \\ 8 & 24 \\ l e & l e \\ - & -- \\ 24 & 8 \end{bmatrix}$$

integration with a row-column premultiplier {a , b} . {z1 , z2}

$$\begin{aligned}
& \begin{array}{ccccccc} & & 2 & & 2 & & \\ & & a & l e & & b & l e & & 7 & a & l e \end{array} \\
& \hline
& \begin{array}{ccccccc} & & 4 & b & l e & 4 & a & l e & & 60 & & 60 \end{array} \\
& \begin{array}{ccccccc} 225 & (- & & + & &) & & & & & & \end{array} \\
& \begin{array}{ccccccc} & & 15 & & 15 & & & & & & & \end{array} \\
& \hline
& \begin{array}{ccccccc} & & 2 & & & & & & & & & \\ & & a & b & l e & & b & l e & & a & l e \end{array} \\
& \hline
& \begin{array}{ccccccc} & & 4 & b & l e & 4 & a & l e & & 60 & & 60 \end{array} \\
& \begin{array}{ccccccc} 225 & (- & & + & &) & & & & & & \end{array} \\
& \begin{array}{ccccccc} & & 15 & & 15 & & & & & & & \end{array}
\end{aligned}$$

$$\begin{aligned}
& \frac{32x^3 - 48le x^2 + 18le^2 x}{4} \\
& - \frac{(64x^3 - 120le x^2 + 72le^2 x)(64x^3 - 72le x^2 + 24le^2 x)}{16 \left(\frac{x^3}{2le} - \frac{x^2}{le} + x \right)} \\
& + \frac{36le^8 \left(\frac{x^3}{le^2} - \frac{16(2x^3 - 3le x^2)}{3le^4} + \frac{16x^3}{3le^4} \right)}{(64x^3 - 72le x^2 + 24le^2 x)} + \frac{4x^3 - 2x^2}{3le^2} + x \\
& + \frac{16 \left(\frac{x^3}{2le} - \frac{x^2}{le} + x \right)}{36le^8 \left(\frac{x^3}{le^2} - \frac{16(2x^3 - 3le x^2)}{3le^4} + \frac{16x^3}{3le^4} \right)} + \frac{2(4x^3 - 3le x^2)}{3le^4} + \frac{4x^3}{3le^4}
\end{aligned}$$

integrate([D],x,0,le)\$

$$\begin{bmatrix} 1 & 1 \\ - & - \\ le & le \\ 1 & 1 \\ - & - \\ le & le \end{bmatrix}$$

integration with a row-column premultiplier {a , b} . {z1 , z2}^t

$$\begin{aligned}
& \begin{bmatrix} & & & 2b & 2a^2 \\ & & & (-\frac{\quad}{3le} - \frac{\quad}{le}) \\ & b & 11a & 3le & le \\ & \frac{\quad}{2le} + \frac{\quad}{6le} - \frac{\quad}{\quad} \\ & 8b & 8a \\ & \frac{\quad}{3le} + \frac{\quad}{3le} \end{bmatrix} \\
\text{col 1} = & \begin{bmatrix} & & & 2b & 2a & 2b & 2a \\ & & & (-\frac{\quad}{le} - \frac{\quad}{3le}) & (-\frac{\quad}{3le} - \frac{\quad}{le}) \\ b & a & & & & & \\ \frac{\quad}{6le} + \frac{\quad}{6le} - \frac{\quad}{\quad} \\ 6le & 6le & & 8b & 8a \\ & \frac{\quad}{3le} + \frac{\quad}{3le} \end{bmatrix} \\
& \begin{bmatrix} & & & 2b & 2a & 2b & 2a \\ & & & (-\frac{\quad}{le} - \frac{\quad}{3le}) & (-\frac{\quad}{3le} - \frac{\quad}{le}) \\ b & a & & & & & \\ \frac{\quad}{6le} + \frac{\quad}{6le} - \frac{\quad}{\quad} \\ 6le & 6le & & 8b & 8a \\ & \frac{\quad}{3le} + \frac{\quad}{3le} \end{bmatrix} \\
\text{col 2} = & \begin{bmatrix} & & & 2b & 2a^2 \\ & & & (-\frac{\quad}{le} - \frac{\quad}{3le}) \\ 11b & a & & le & 3le \\ \frac{\quad}{6le} + \frac{\quad}{2le} - \frac{\quad}{\quad} \\ 6le & 2le & & 8b & 8a \\ & \frac{\quad}{3le} + \frac{\quad}{3le} \end{bmatrix}
\end{aligned}$$

integration with a row-column premultiplier $\{1, c, 1\} \cdot \{z_1, z_2, z_3\}^t$

$$\begin{aligned}
& \begin{bmatrix} 1 & 1 \\ -- & -- \\ le & le \\ 1 & 1 \\ - & - \\ le & le \end{bmatrix} \\
(2c+1) & \frac{\quad}{3le}
\end{aligned}$$

Cubic basis:

A200 : [M] - matrix, k3 (Lagrange Polynomial)
indefinite integral (mmatx,x)

$$\begin{aligned}
& \begin{matrix} 7 & 6 & 2 & 5 & 3 & 4 & 4 & 3 \\ m = & (1215 x^7 - 5670 le x^6 + 10962 le x^5 - 11340 le x^4 + 6755 le x^3 \\ & 11 \end{matrix}
\end{aligned}$$

$$- 2310 \text{ le } x^5 + 420 \text{ le } x^6) / (420 \text{ le}),$$

definite integral (mmatx,x,0,le);

$$\begin{bmatrix} 8 \text{ le} & 33 \text{ le} & 3 \text{ le} & 19 \text{ le} \\ \hline 105 & 560 & 140 & 1680 \\ 33 \text{ le} & 27 \text{ le} & 27 \text{ le} & 3 \text{ le} \\ \hline 560 & 70 & 560 & 140 \\ 3 \text{ le} & 27 \text{ le} & 27 \text{ le} & 33 \text{ le} \\ \hline 140 & 560 & 70 & 560 \\ 19 \text{ le} & 3 \text{ le} & 33 \text{ le} & 8 \text{ le} \\ \hline 1680 & 140 & 560 & 105 \end{bmatrix}$$

A211 : [D] - matrix, k3 (Lagrange Polynomial)

indefinite integral (kmatx,x)

$$\begin{aligned} & \frac{81}{11} \left(\frac{x^5}{5 \text{ le}} - \frac{x^4}{4 \text{ le}} + \frac{3 \text{ le } x^2}{9} + \frac{x^3}{3 \text{ le}} + \frac{4 x}{81} \right) \\ & + \frac{3 (216 x^5 - 675 \text{ le } x^4 + 800 \text{ le } x^3 - 450 \text{ le } x^2 + 120 \text{ le } x)}{40 \text{ le}^6} \\ & + \frac{27 (24 x^5 - 90 \text{ le } x^4 + 130 \text{ le } x^3 - 90 \text{ le } x^2 + 30 \text{ le } x)}{40 \text{ le}^6}, \end{aligned}$$

definite integral (kmatx,x,0,le);

$$\begin{bmatrix} 37 & 189 & 27 & 13 \\ - & - & - & - \\ 10 \text{ le} & 40 \text{ le} & 20 \text{ le} & 40 \text{ le} \\ [& & &] \\ 189 & 54 & 297 & 27 \\ - & - & - & - \\ 40 \text{ le} & 5 \text{ le} & 40 \text{ le} & 20 \text{ le} \\ [& & &] \\ 27 & 297 & 54 & 189 \\ - & - & - & - \\ 20 \text{ le} & 40 \text{ le} & 5 \text{ le} & 40 \text{ le} \\ [& & &] \\ 13 & 27 & 189 & 37 \\ - & - & - & - \\ 40 \text{ le} & 20 \text{ le} & 40 \text{ le} & 10 \text{ le} \end{bmatrix}$$

$$[M] : \begin{matrix} r & & -1 \\ [M] & : & [M] & - & [M] & [M] & [M] \\ & A & B & D & C \end{matrix}$$

integrate([M],x,0,le)\$

$$\begin{bmatrix} 1 \text{ le} & 1 \text{ le} \\ - & - \\ 15 & 60 \\ [&] \\ 1 \text{ le} & 1 \text{ le} \\ - & - \\ 60 & 15 \end{bmatrix}$$

integration with a row-column premultiplier {a , b} . {z1 , z2} ^T

$$\begin{aligned} m &= \frac{r}{11} - \left(\frac{3 b \text{ le}}{1120} + \frac{9 a \text{ le}}{160} \right) \left(\left(\frac{3 b \text{ le}}{1120} + \frac{9 a \text{ le}}{160} \right) \left(\frac{297 b \text{ le}}{1120} + \frac{27 a \text{ le}}{224} \right) \right. \\ &\quad \left. \left(\frac{27 b \text{ le}}{224} + \frac{297 a \text{ le}}{1120} \right) \left(\frac{297 b \text{ le}}{1120} + \frac{27 a \text{ le}}{224} \right) \right. \\ &\quad \left. + \left(\frac{27 b \text{ le}}{1120} - \frac{27 a \text{ le}}{1120} \right) \left(\frac{27 b \text{ le}}{1120} + \frac{27 a \text{ le}}{1120} \right) \right) \\ &\quad + \left(\frac{3 b \text{ le}}{1120} - \frac{27 a \text{ le}}{1120} \right) \left(\frac{27 b \text{ le}}{1120} + \frac{27 a \text{ le}}{1120} \right) \dots \end{aligned}$$

For the diffusion matrix

$$[D] : [D] - [D] [D] [D] [D]$$

A B D C

$$D = \frac{(13122 x^5 - 40095 l e x^4 + 46170 l e^2 x^3 - 24570 l e^3 x^2 + 5940 l e^4 x) ((6561 x^5 - 18225 l e x^4 + 18630 l e^2 x^3 - 8370 l e^3 x^2 + 1485 l e^4 x) (39366 x^5 - 98415 l e x^4 + 86670 l e^2 x^3 - 31590 l e^3 x^2 + 4860 l e^4 x) / (7200 l e^6) - (486 x^5 - 1215 l e x^4 + 990 l e^2 x^3 - 270 l e^3 x^2))}{10 l e^6}$$

integrate([D], x, 0, le)\$

$$\begin{bmatrix} 1 & 1 \\ -- & - -- \\ l e & l e \\ [&] \\ [1 & 1 \\ - -- & - \\ l e & l e \end{bmatrix}$$

integration with a row-column premultiplier {a, b} . {z1, z2}

$$D = \frac{540 b^3 + 4571 a b^2 + 1154 a^2 b - 11341 a^3}{180 (11 b^2 + 38 a b + 11 a^2)}$$

Appendix F

Two dimensional Diffusion Matrix Macsyma (1988) output

(c64) b2kkr(4,4) = [D]

$$b2kkr(4,4) = [D]^r = [D]^a - [D]^b [D]^{-1} [D]^d [D]^c$$

```

      3      2      3
(d64) matrix([((194176 cx  + 456864 cx  + 313224 cx + 64036) cy
      3      2      2
+ (456864 cx  + 1088192 cx  + 793564 cx + 176830) cy
      3      2      3      2
+ (313224 cx  + 793564 cx  + 625780 cx + 152182) cy + 64036 cx  + 176830 cx
      2      2
+ 152182 cx + 40077)/((1971360 cx  + 2184480 cx + 569160) cy
      2      2
+ (2184480 cx  + 3227040 cx + 1077480) cy + 569160 cx  + 1077480 cx + 479610),
      3      2      3
- ((87616 cx  + 146624 cx  + 78984 cx + 15076) cy
      3      2      2
+ (222784 cx  + 403312 cx  + 236364 cx + 49990) cy
      3      2      3      2
+ (165784 cx  + 342484 cx  + 224660 cx + 52822) cy + 35516 cx  + 83730 cx
      2      2
+ 61642 cx + 15987)/((1971360 cx  + 2184480 cx + 569160) cy
      2      2
+ (2184480 cx  + 3227040 cx + 1077480) cy + 569160 cx  + 1077480 cx + 479610),
      3      2      3
- ((18944 cx  + 87456 cx  + 68456 cx + 13444) cy
      3      2      2
+ (87456 cx  + 281568 cx  + 214716 cx + 43110) cy
      3      2      3      2
+ (68456 cx  + 214716 cx  + 176460 cx + 37718) cy + 13444 cx  + 43110 cx
      2      2
+ 37718 cx + 8103)/((1971360 cx  + 2184480 cx + 569160) cy
      2      2
+ (2184480 cx  + 3227040 cx + 1077480) cy + 569160 cx  + 1077480 cx + 479610),
      3      2      3
- ((87616 cx  + 222784 cx  + 165784 cx + 35516) cy
      3      2      2

```

$$\begin{aligned}
& + (146624 \text{ cx} + 403312 \text{ cx} + 342484 \text{ cx} + 83730) \text{ cy} \\
& + (78984 \text{ cx} + 236364 \text{ cx} + 224660 \text{ cx} + 61642) \text{ cy} + 15076 \text{ cx} + 49990 \text{ cx} \\
& + 52822 \text{ cx} + 15987) / ((1971360 \text{ cx} + 2184480 \text{ cx} + 569160) \text{ cy} \\
& + (2184480 \text{ cx} + 3227040 \text{ cx} + 1077480) \text{ cy} + 569160 \text{ cx} + 1077480 \text{ cx} \\
& + 479610)], [- ((87616 \text{ cx} + 146624 \text{ cx} + 78984 \text{ cx} + 15076) \text{ cy} \\
& + (222784 \text{ cx} + 403312 \text{ cx} + 236364 \text{ cx} + 49990) \text{ cy} \\
& + (165784 \text{ cx} + 342484 \text{ cx} + 224660 \text{ cx} + 52822) \text{ cy} + 35516 \text{ cx} + 83730 \text{ cx} \\
& + 61642 \text{ cx} + 15987) / ((1971360 \text{ cx} + 2184480 \text{ cx} + 569160) \text{ cy} \\
& + (2184480 \text{ cx} + 3227040 \text{ cx} + 1077480) \text{ cy} + 569160 \text{ cx} + 1077480 \text{ cx} + 479610), \\
& ((194176 \text{ cx} + 456864 \text{ cx} + 313224 \text{ cx} + 64036) \text{ cy} \\
& + (456864 \text{ cx} + 1088192 \text{ cx} + 793564 \text{ cx} + 176830) \text{ cy} \\
& + (313224 \text{ cx} + 793564 \text{ cx} + 625780 \text{ cx} + 152182) \text{ cy} + 64036 \text{ cx} + 176830 \text{ cx} \\
& + 152182 \text{ cx} + 40077) / ((1971360 \text{ cx} + 2184480 \text{ cx} + 569160) \text{ cy} \\
& + (2184480 \text{ cx} + 3227040 \text{ cx} + 1077480) \text{ cy} + 569160 \text{ cx} + 1077480 \text{ cx} + 479610), \\
& - ((87616 \text{ cx} + 222784 \text{ cx} + 165784 \text{ cx} + 35516) \text{ cy} \\
& + (146624 \text{ cx} + 403312 \text{ cx} + 342484 \text{ cx} + 83730) \text{ cy} \\
& + (78984 \text{ cx} + 236364 \text{ cx} + 224660 \text{ cx} + 61642) \text{ cy} + 15076 \text{ cx} + 49990 \text{ cx} \\
& + 52822 \text{ cx} + 15987) / ((1971360 \text{ cx} + 2184480 \text{ cx} + 569160) \text{ cy} \\
& + (2184480 \text{ cx} + 3227040 \text{ cx} + 1077480) \text{ cy} + 569160 \text{ cx} + 1077480 \text{ cx} + 479610), \\
& - ((18944 \text{ cx} + 87456 \text{ cx} + 68456 \text{ cx} + 13444) \text{ cy} \\
& + (87456 \text{ cx} + 281568 \text{ cx} + 214716 \text{ cx} + 43110) \text{ cy} \\
& + (68456 \text{ cx} + 214716 \text{ cx} + 176460 \text{ cx} + 37718) \text{ cy} + 13444 \text{ cx} + 43110 \text{ cx} \\
& + 37718 \text{ cx} + 8103) / ((1971360 \text{ cx} + 2184480 \text{ cx} + 569160) \text{ cy} \\
& + (2184480 \text{ cx} + 3227040 \text{ cx} + 1077480) \text{ cy} + 569160 \text{ cx} + 1077480 \text{ cx}
\end{aligned}$$

$$\begin{aligned}
& + 479610)], [- ((18944 \text{ cx}^3 + 87456 \text{ cx}^2 + 68456 \text{ cx} + 13444) \text{ cy}^2 \\
& + (87456 \text{ cx}^3 + 281568 \text{ cx}^2 + 214716 \text{ cx} + 43110) \text{ cy}^3 \\
& + (68456 \text{ cx}^2 + 214716 \text{ cx} + 176460 \text{ cx} + 37718) \text{ cy}^2 + 13444 \text{ cx}^3 + 43110 \text{ cx}^2 \\
& + 37718 \text{ cx} + 8103)/((1971360 \text{ cx}^2 + 2184480 \text{ cx} + 569160) \text{ cy}^2 \\
& + (2184480 \text{ cx}^3 + 3227040 \text{ cx}^2 + 1077480) \text{ cy}^3 + 569160 \text{ cx}^3 + 1077480 \text{ cx}^2 + 479610), \\
& - ((87616 \text{ cx}^3 + 222784 \text{ cx}^2 + 165784 \text{ cx} + 35516) \text{ cy}^2 \\
& + (146624 \text{ cx}^3 + 403312 \text{ cx}^2 + 342484 \text{ cx} + 83730) \text{ cy}^3 \\
& + (78984 \text{ cx}^2 + 236364 \text{ cx} + 224660 \text{ cx} + 61642) \text{ cy}^2 + 15076 \text{ cx}^3 + 49990 \text{ cx}^2 \\
& + 52822 \text{ cx} + 15987)/((1971360 \text{ cx}^2 + 2184480 \text{ cx} + 569160) \text{ cy}^2 \\
& + (2184480 \text{ cx}^3 + 3227040 \text{ cx}^2 + 1077480) \text{ cy}^3 + 569160 \text{ cx}^3 + 1077480 \text{ cx}^2 + 479610), \\
& ((194176 \text{ cx}^3 + 456864 \text{ cx}^2 + 313224 \text{ cx} + 64036) \text{ cy}^2 \\
& + (456864 \text{ cx}^3 + 1088192 \text{ cx}^2 + 793564 \text{ cx} + 176830) \text{ cy}^3 \\
& + (313224 \text{ cx}^2 + 793564 \text{ cx} + 625780 \text{ cx} + 152182) \text{ cy}^2 + 64036 \text{ cx}^3 + 176830 \text{ cx}^2 \\
& + 152182 \text{ cx} + 40077)/((1971360 \text{ cx}^2 + 2184480 \text{ cx} + 569160) \text{ cy}^2 \\
& + (2184480 \text{ cx}^3 + 3227040 \text{ cx}^2 + 1077480) \text{ cy}^3 + 569160 \text{ cx}^3 + 1077480 \text{ cx}^2 + 479610), \\
& - ((87616 \text{ cx}^3 + 146624 \text{ cx}^2 + 78984 \text{ cx} + 15076) \text{ cy}^2 \\
& + (222784 \text{ cx}^3 + 403312 \text{ cx}^2 + 236364 \text{ cx} + 49990) \text{ cy}^3 \\
& + (165784 \text{ cx}^2 + 342484 \text{ cx} + 224660 \text{ cx} + 52822) \text{ cy}^2 + 35516 \text{ cx}^3 + 83730 \text{ cx}^2 \\
& + 61642 \text{ cx} + 15987)/((1971360 \text{ cx}^2 + 2184480 \text{ cx} + 569160) \text{ cy}^2 \\
& + (2184480 \text{ cx}^3 + 3227040 \text{ cx}^2 + 1077480) \text{ cy}^3 + 569160 \text{ cx}^3 + 1077480 \text{ cx}^2 \\
& + 479610)], [- ((87616 \text{ cx}^3 + 222784 \text{ cx}^2 + 165784 \text{ cx} + 35516) \text{ cy}^2 \\
& + (146624 \text{ cx}^3 + 403312 \text{ cx}^2 + 342484 \text{ cx} + 83730) \text{ cy}^3 \\
& + (78984 \text{ cx}^2 + 236364 \text{ cx} + 224660 \text{ cx} + 61642) \text{ cy}^2 + 15076 \text{ cx}^3 + 49990 \text{ cx}^2 \\
& + 52822 \text{ cx} + 15987)/((1971360 \text{ cx}^2 + 2184480 \text{ cx} + 569160) \text{ cy}^2
\end{aligned}$$

$$\begin{aligned}
& + (2184480 \text{ cx}^3 + 3227040 \text{ cx}^2 + 1077480 \text{ cx} + 569160) \text{ cy}^3 + 1077480 \text{ cx}^3 + 479610), \\
& - ((18944 \text{ cx}^3 + 87456 \text{ cx}^2 + 68456 \text{ cx} + 13444) \text{ cy}^2 \\
& + (87456 \text{ cx}^3 + 281568 \text{ cx}^2 + 214716 \text{ cx} + 43110) \text{ cy}^3 \\
& + (68456 \text{ cx}^3 + 214716 \text{ cx}^2 + 176460 \text{ cx} + 37718) \text{ cy}^3 + 13444 \text{ cx}^3 + 43110 \text{ cx}^2 \\
& + 37718 \text{ cx} + 8103)/((1971360 \text{ cx}^2 + 2184480 \text{ cx} + 569160) \text{ cy}^2 \\
& + (2184480 \text{ cx}^3 + 3227040 \text{ cx}^2 + 1077480 \text{ cx} + 569160) \text{ cy}^3 + 1077480 \text{ cx}^3 + 479610), \\
& - ((87616 \text{ cx}^3 + 146624 \text{ cx}^2 + 78984 \text{ cx} + 15076) \text{ cy}^2 \\
& + (222784 \text{ cx}^3 + 403312 \text{ cx}^2 + 236364 \text{ cx} + 49990) \text{ cy}^3 \\
& + (165784 \text{ cx}^3 + 342484 \text{ cx}^2 + 224660 \text{ cx} + 52822) \text{ cy}^3 + 35516 \text{ cx}^3 + 83730 \text{ cx}^2 \\
& + 61642 \text{ cx} + 15987)/((1971360 \text{ cx}^2 + 2184480 \text{ cx} + 569160) \text{ cy}^2 \\
& + (2184480 \text{ cx}^3 + 3227040 \text{ cx}^2 + 1077480 \text{ cx} + 569160) \text{ cy}^3 + 1077480 \text{ cx}^3 + 479610), \\
& ((194176 \text{ cx}^3 + 456864 \text{ cx}^2 + 313224 \text{ cx} + 64036) \text{ cy}^2 \\
& + (456864 \text{ cx}^3 + 1088192 \text{ cx}^2 + 793564 \text{ cx} + 176830) \text{ cy}^3 \\
& + (313224 \text{ cx}^3 + 793564 \text{ cx}^2 + 625780 \text{ cx} + 152182) \text{ cy}^3 + 64036 \text{ cx}^3 + 176830 \text{ cx}^2 \\
& + 152182 \text{ cx} + 40077)/((1971360 \text{ cx}^2 + 2184480 \text{ cx} + 569160) \text{ cy}^2 \\
& + (2184480 \text{ cx}^3 + 3227040 \text{ cx}^2 + 1077480 \text{ cx} + 569160) \text{ cy}^3 + 1077480 \text{ cx}^3 + 479610))]
\end{aligned}$$

(c65) cx:c\$

(c66) cy:c\$

(c67) ev(d64,eval)\$

(c68) 90*((148*c^2+164*c+73)/(2*c+1))*%;

(d68) col 1 =

$$\begin{aligned}
& \begin{bmatrix} 656 c^3 + 2032 c^2 + 1838 c + 549 \\ - (2 c + 3) (148 c^2 + 164 c + 73) \\ - (4 c + 1) (16 c^2 + 118 c + 111) \\ - (2 c + 3) (148 c^2 + 164 c + 73) \end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
& \begin{bmatrix} - (2c + 3) (148c^2 + 164c + 73) \\ 656c^3 + 2032c^2 + 1838c + 549 \\ - (2c + 3) (148c^2 + 164c + 73) \\ - (4c + 1) (16c^2 + 118c + 111) \end{bmatrix} \\
\text{col 2} = & \begin{bmatrix} - (2c + 3) (148c^2 + 164c + 73) \\ 656c^3 + 2032c^2 + 1838c + 549 \\ - (2c + 3) (148c^2 + 164c + 73) \\ - (4c + 1) (16c^2 + 118c + 111) \end{bmatrix} \\
& \begin{bmatrix} - (4c + 1) (16c^2 + 118c + 111) \\ - (2c + 3) (148c^2 + 164c + 73) \\ 656c^3 + 2032c^2 + 1838c + 549 \\ - (2c + 3) (148c^2 + 164c + 73) \end{bmatrix} \\
\text{col 3} = & \begin{bmatrix} - (2c + 3) (148c^2 + 164c + 73) \\ 656c^3 + 2032c^2 + 1838c + 549 \\ - (2c + 3) (148c^2 + 164c + 73) \\ - (4c + 1) (16c^2 + 118c + 111) \end{bmatrix} \\
& \begin{bmatrix} - (2c + 3) (148c^2 + 164c + 73) \\ - (4c + 1) (16c^2 + 118c + 111) \\ - (2c + 3) (148c^2 + 164c + 73) \\ 656c^3 + 2032c^2 + 1838c + 549 \end{bmatrix} \\
\text{col 4} = & \begin{bmatrix} - (2c + 3) (148c^2 + 164c + 73) \\ - (4c + 1) (16c^2 + 118c + 111) \\ - (2c + 3) (148c^2 + 164c + 73) \\ 656c^3 + 2032c^2 + 1838c + 549 \end{bmatrix}
\end{aligned}$$

Note: the common premultiplier for (d68) is the ratio $\frac{90(148c^2 + 164c + 73)}{2c + 1}$.


```
(c74) c:1$
(c75) ev(d70,eval)$
(c77) ratsimp(%);
```

```
(d77)
[ 29  1  7  1 ]
[ -- - - - ]
[ 66  6 66  6 ]
[      ]
[  1 29  1  7 ]
[ - - - - ]
[  6 66  6 66 ]
[      ]
[  7  1 29  1 ]
[ - - - - ]
[ 66  6 66  6 ]
[      ]
[  1  7  1 29 ]
[ - - - - ]
[  6 66  6 66 ]
```

```
(c78) eigenvalues(%);
```

```
(d78)
      2 6 6
[ -, --, --, 0 ]
      3 11 11
```

```
non-zero maximum eigenvalue
ratio of ----- = 1.22
non-zero minimum eigenvalue
```

```
standard bilinear b2kk (-1:1,-1:1)
```

```
(D80)
[ 2  1  1  1 ]
[ - - - - ]
[ 3  6  3  6 ]
[      ]
[  1 2  1  1 ]
[ - - - - ]
[  6 3  6  3 ]
[      ]
[  1 1  2  1 ]
[ - - - - ]
[  3 6  3  6 ]
[      ]
[  1 1  1  2 ]
[ - - - - ]
[  6 3  6  3 ]
```

```
(D81)
      2
[ -, 1, 1, 0 ]
      3
```

```
non-zero maximum eigenvalue
ratio of ----- = 1.5
non-zero minimum eigenvalue
```

Appendix G

Computer output for axi-symmetric pure diffusion problem (8.1), using $1 \leq k \leq 2$ and $0 \leq p \leq 4$ via (5.7). Comparison of the standard GWS^h ($k = 1, 2$ and $p = 0$) system matrices with the higher order p - WS^h ($p \geq 1$, $k = 1$) system matrices show a small perturbation of the values of matrix elements can have significant impact on the solution accuracy.

SROY 2/5/93

```

*****
      finite elements 1-2-3  code: learn finite elements
*****
titl:      ***** code exercise 4.2, m=2, k=1 *****
type:      k  n  nnode1  refl  npri  ntran  naxi  nztr
           1  1      2    1.0    1      0      1      0
prin:      nbug1 nbug2 nbug3 nbug4 nbug5 nbug6
           1      1      1      3      0      0
grid:      ni  npri  ntran
           1      1      0
           x1      xr1      pr1      xr2      pr2      xr3      pr3
           1.0000  2.0000  1.0000
           nem1  nem2  nem3
           2
node coordinates (x , integer map)  exponent e: -2
           100  150  200
matl:      macro nodal conductivities
           1  0.10000E+02  0.10000E+02
robn:      nrob:      1
           jrob:      1
           h          tref
           20.00000  1500.00000
diri:      nfix:      1  knum:  0  isor:  0
           jfix:      3
           306.85281
form:      system matrix [k + h]; column format * Referred to as [S] later
           0.00  45.00  -25.00
           -25.00  60.00  -35.00
           0.00  1.00  0.00
data matrix {b} row format
min =      0.00000000  max =30000.00000000
30000.00  0.00  306.85
solv:      nodal solution (real variables)
min = 306.85281372  max = 996.86572266
996.87  594.36  306.85
norm:      f.e. soln. energy semi-norm:  e(qh)= 0.13409104E+08

```

stop

```
titl:      ***** code exercise 4.2, m=1, k=2 *****
type:      k  n  nnode1  refl  npri  ntran  naxi  nztr
           2  1      3  1.0    1      0      1      0
prin:      nbug1 nbug2 nbug3 nbug4 nbug5 nbug6
           1      1      1      3      0      0
grid:      ni  npri  ntran
           1      1      0
           xl      xr1      pr1      xr2      pr2      xr3      pr3
           1.0000  2.0000  1.0000
           nem1  nem2  nem3
           1
           node coordinates (x , integer map)  exponent e: -2
           100  150  200
matl:      macro nodal conductivities
           1  0.10000E+02  0.10000E+02
robn:      nrob:      1
           jrob:      1
           h      tref
           20.00000  1500.00000
diri:      nfix:      1  knum:  0  isor:  0
           jfix:      3
           306.85281
form:      system matrix [k + h]; column format
           0.00      0.00      48.33  -33.33      5.00
           0.00  -33.33      80.00  -46.67      0.00
           0.00      0.00      1.00      0.00      0.00
           data matrix {b} row format
           min =      0.000000000  max =30000.000000000
           30000.00      0.00  306.85
solv:      nodal solution (real variables)
           min =  306.85281372  max =  999.64794922
           999.65  595.52  306.85
norm:      f.e. soln. energy semi-norm:  e(qh)= 0.13459373E+08
stop
```

```
titl:      ***** code exercise 4.2, m=1, p=1 *****
type:      k  n  nnode1  refl  npri  ntran  naxi  nztr
           1  1      2  1.0    1      0      1      0
prin:      nbug1 nbug2 nbug3 nbug4 nbug5 nbug6
           1      1      1      3      0      0
grid:      ni  npri  ntran
           1      1      0
           xl      xr1      pr1      xr2      pr2      xr3      pr3
           1.0000  2.0000  1.0000
           nem1  nem2  nem3
           1
```

```

node coordinates (x , integer map) exponent e: -2
  100 200
matl: macro nodal conductivities
      1 0.10000D+02 0.10000D+02
robn: nrob: 1
      jrob: 1
           h      tref
           20.00000 1500.00000
diri: nfix: 1 knum: 0 isor: 0
      jfix: 2
           306.85282
form: system matrix [k + h]; column format * Referred to as [V] later
           0.00 34.44 -14.44 -----
           0.00 1.00 0.00
data matrix {b} row format
min = 306.85282000 max = 30000.00000000
30000.00 306.85
solv: nodal solution (real variables)
min = 306.85282000 max = 999.64795677
999.65 306.85
norm: f.e. soln. energy semi-norm: e(qh)= 0.23452335D+08
stop

titl: ***** code exercise 4.2, m=2, p=1 *****
type: k n nnode1 refl npri ntran naxi nztr
      1 1 2 1.0 1 0 1 0
prin: nbug1 nbug2 nbug3 nbug4 nbug5 nbug6
      1 1 1 3 0 0
grid: ni npri ntran
      1 1 0
           xl xr1 pr1 xr2 pr2 xr3 pr3
           1.0000 2.0000 1.0000
           nem1 nem2 nem3
           2
node coordinates (x , integer map) exponent e: -2
  100 150 200
matl: macro nodal conductivities
      1 0.10000D+02 0.10000D+02
robn: nrob: 1
      jrob: 1
           h      tref
           20.00000 1500.00000
diri: nfix: 1 knum: 0 isor: 0
      jfix: 3
           306.85282
form: system matrix [k + h]; column format * Compare with [S]
           0.00 44.67 -24.67 -----
          -24.67 59.43 -34.76

```

```

      0.00    1.00    0.00
data matrix {b} row format
min =    0.00000000    max =30000.00000000
30000.00    0.00  306.85
solv: nodal solution (real variables)
min =  306.85282000    max =  999.97043706
      999.97  594.54  306.85
norm: f.e. soln. energy semi-norm:  e(qh)= 0.23464610D+08
stop

```

```

titl:      ***** code exercise 4.2, m=4, p=1 *****
type:      k n nnode1 refl npri ntran naxi nztr
           1 1      2 1.0      1      0      1      0
prin: nbug1 nbug2 nbug3 nbug4 nbug5 nbug6
           1      1      1      3      0      0
grid:      ni npri ntran
           1      1      0
           x1      xr1      pr1      xr2      pr2      xr3      pr3
           1.0000  2.0000  1.0000
node coordinates (x , integer map) exponent e: -2
           100 125 150 175 200
matl: macro nodal conductivities
           1 0.10000D+02 0.10000D+02
robn: nrob: 1
      jrob: 1
           h      tref
           20.00000 1500.00000
diri: nfix: 1 knum: 0 isor: 0
      jfix: 5
           306.85282
form: system matrix [k + h]; column format
           0.00  64.81 -44.81
           -44.81  99.66 -54.85
           -54.85 119.72 -64.87
           -64.87 139.76 -74.89
           0.00  1.00  0.00
data matrix {b} row format
min =    0.00000000    max =30000.00000000
30000.00    0.00    0.00    0.00  306.85
solv: nodal solution (real variables)
min =  306.85282000    max =  999.99795194
      1000.00  776.86  594.54  440.38  306.85
norm: f.e. soln. energy semi-norm:  e(qh)= 0.23465658D+08
stop

```

```

titl:      ***** code exercise 4.2, m=8, p=1 *****
type:      k n nnode1 refl npri ntran naxi nztr
           1 1      2 1.0      1      0      1      0

```

```

prin:  nbug1  nbug2  nbug3  nbug4  nbug5  nbug6
        1      1      1      3      0      0
grid:   ni  npri  ntran
        1      1      0
        xl    xr1    pr1    xr2    pr2    xr3    pr3
        1.0000  2.0000  1.0000
node coordinates (x , integer map) exponent e: -2
        100  112  125  137  150  162  175  187  200
matl:   macro nodal conductivities
        1  0.10000D+02  0.10000D+02
robn:   nrob:    1
        jrob:    1
        h        tref
        20.00000 1500.00000
diri:   nfix:    1  knum:  0  isor:  0
        jfix:    9
        306.85282
form:   system matrix [k + h]; column format
        0.00  104.90  -84.90
        -84.90  179.81  -94.91
        -94.91  199.83  -104.92
        -104.92  219.85  -114.93
        -114.93  239.86  -124.93
        -124.93  259.87  -134.94
        -134.94  279.88  -144.94
        -144.94  299.89  -154.95
        0.00    1.00    0.00
data matrix {b} row format
min =    0.00000000    max =30000.00000000
30000.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
306.85
solv:   nodal solution (real variables)
min =  306.85282000    max =  999.99986838
1000.00  882.22  776.86  681.55  594.53  514.49  440.38  371.39
306.85
norm:   f.e. soln. energy semi-norm:  e(qh)= 0.23465731D+08
stop

titl:   ***** code exercise 4.2, m=16, p=1 *****
type:   k  n  nnode1  refl  npri  ntran  naxi  nztr
        1  1      2  1.0    1      0      1      0
prin:   nbug1  nbug2  nbug3  nbug4  nbug5  nbug6
        1      1      1      3      0      0
grid:   ni  npri  ntran
        1      1      0
        xl    xr1    pr1    xr2    pr2    xr3    pr3
        1.0000  2.0000  1.0000
node coordinates (x , integer map) exponent e: -2

```

```

100 106 112 118 125 131 137 143 150 156 162 168 175 181 187
193 200
matl: macro nodal conductivities
      1 0.10000D+02 0.10000D+02
robn: nrob: 1
      jrob: 1
           h      tref
           20.00000 1500.00000
diri: nfix: 1 knum: 0 isor: 0
      jfix: 17
           306.85282
form: system matrix [k + h]; column format
      0.00 184.95 -164.95
      -164.95 339.90 -174.95
      -174.95 359.91 -184.95
      -184.95 379.91 -194.96
      -194.96 399.92 -204.96
      -204.96 419.92 -214.96
      -214.96 439.92 -224.96
      -224.96 459.93 -234.96
      -234.96 479.93 -244.97
      -244.97 499.93 -254.97
      -254.97 519.94 -264.97
      -264.97 539.94 -274.97
      -274.97 559.94 -284.97
      -284.97 579.94 -294.97
      -294.97 599.94 -304.97
      -304.97 619.95 -314.97
      0.00 1.00 0.00
data matrix {b} row format
min = 0.000000000 max = 30000.000000000
30000.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
306.85
solv: nodal solution (real variables)
min = 306.85282000 max = 999.99999193
1000.00 939.38 882.22 828.15 776.86 728.07 681.55 637.09
594.53 553.71 514.49 476.75 440.38 405.29 371.39 338.60
306.85
norm: f.e. soln. energy semi-norm: e(qh)= 0.23465736D+08
stop

titl: ***** code exercise 4.2, m=1, p=2 *****
type: k n nnode1 refl npri ntran naxi nztr
      1 1 2 1.0 1 0 0 0
prin: nbug1 nbug2 nbug3 nbug4 nbug5 nbug6
      1 1 1 3 0 0
grid: ni npri ntran

```

```

      1      1      0
      xl      xr1      pr1      xr2      pr2      xr3      pr3
      1.0000      2.0000      1.0000
node coordinates (x , integer map) exponent e: -2
      100      200
matl: macro nodal conductivities
      1      0.10000D+02      0.10000D+02
robn: nrob:      1
      jrob:      1
           h      tref
      20.00000      1500.00000
diri: nfix:      1      knum:      0      isor:      0
      jfix:      2
      306.85282
form: system matrix [k + h]; column format * Compare with [V]
      0.00      34.43      -14.43      -----
      0.00      1.00      0.00
data matrix {b} row format
min =      306.85282000      max =30000.00000000
      30000.00      306.85
solv: nodal solution (real variables)
min =      306.85282000      max =      999.98931925
      999.99      306.85
norm: f.e. soln. energy semi-norm:      e(qh)=      0.23465329D+08
stop

titl:      ***** code exercise 4.2, m=2, p=2 *****
type:      k      n      nnode1      refl      npri      ntran      naxi      nztr
      1      1      2      1.0      1      0      1      0
prin: nbug1      nbug2      nbug3      nbug4      nbug5      nbug6
      1      1      1      3      0      0
grid:      ni      npri      ntran
      1      1      0
           xl      xr1      pr1      xr2      pr2      xr3      pr3
      1.0000      2.0000      1.0000
node coordinates (x , integer map) exponent e: -2
      100      150      200
matl: macro nodal conductivities
      1      0.10000D+02      0.10000D+02
robn: nrob:      1
      jrob:      1
           h      tref
      20.00000      1500.00000
diri: nfix:      1      knum:      0      isor:      0
      jfix:      3
      306.85282
form: system matrix [k + h]; column format * Compare with [S]
      0.00      44.66      -24.66      -----

```



```

-24.66  59.42 -34.76
  0.00   1.00   0.00
data matrix {b} row format
min =    0.00000000  max =30000.00000000
30000.00   0.00  306.85
solv: nodal solution (real variables)
min = 306.85282000  max = 999.99971329
1000.00  594.53  306.85
norm: f.e. soln. energy semi-norm:  e(qh)= 0.23465725D+08
stop

titl:      ***** code exercise 4.2, m=4, p=2 *****
type:      k  n  nnode1  refl  npri  ntran  naxi  nztr
          1  1      2   1.0    1      0      1      0
prin: nbug1 nbug2 nbug3 nbug4 nbug5 nbug6
          1      1      1      3      0      0
grid:      ni  npri  ntran
          1      1      0
          x1      xr1      pr1      xr2      pr2      xr3      pr3
          1.0000  2.0000  1.0000
node coordinates (x , integer map)  exponent e: -2
          100 125 150 175 200
matl:      macro nodal conductivities
          1  0.10000D+02  0.10000D+02
robn: nrob:  1
      jrob:  1
          h      tref
          20.00000 1500.00000
diri: nfix:  1  knum:  0  isor:  0
      jfix:  5
          306.85282
form: system matrix [k + h]; column format
          0.00  64.81 -44.81
          -44.81  99.66 -54.85
          -54.85 119.72 -64.87
          -64.87 139.76 -74.89
          0.00   1.00   0.00
data matrix {b} row format
min =    0.00000000  max =30000.00000000
30000.00   0.00   0.00   0.00  306.85
solv: nodal solution (real variables)
min = 306.85282000  max = 999.99999473
1000.00  776.86  594.53  440.38  306.85
norm: f.e. soln. energy semi-norm:  e(qh)= 0.23465736D+08
stop

titl:      ***** code exercise 4.2, m=8, p=2 *****
type:      k  n  nnode1  refl  npri  ntran  naxi  nztr

```

```

      1 1      2 1.0      1      0      1      0
prin: nbug1 nbug2 nbug3 nbug4 nbug5 nbug6
      1      1      1      3      0      0
grid:  ni npri ntran
      1      1      0
      xl      xr1      pr1      xr2      pr2      xr3      pr3
      1.0000  2.0000  1.0000
node coordinates (x , integer map) exponent e: -2
      100 112 125 137 150 162 175 187 200
matl:  macro nodal conductivities
      1 0.10000D+02 0.10000D+02
robn:  nrob:      1
      jrob:      1
           h      tref
           20.00000 1500.00000
diri:  nfix:      1 knum: 0 isor: 0
      jfix:      9
           306.85282
form:  system matrix [k + h]; column format
           0.00 104.90 -84.90
           -84.90 179.81 -94.91
           -94.91 199.83 -104.92
           -104.92 219.85 -114.93
           -114.93 239.86 -124.93
           -124.93 259.87 -134.94
           -134.94 279.88 -144.94
           -144.94 299.89 -154.95
           0.00 1.00 0.00
data matrix {b} row format
min = 0.00000000 max = 30000.00000000
30000.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
      306.85
solv:  nodal solution (real variables)
min = 306.85282000 max = 1000.00000014
      1000.00 882.22 776.86 681.55 594.53 514.49 440.38 371.39
      306.85
norm:  f.e. soln. energy semi-norm: e(qh)= 0.23465736D+08
stop

titl:  ***** code exercise 4.2, m=1, p=3 *****
type:  k n nnode refl npri ntran naxi nztr
      1 1      2 1.0      1      0      0      0
prin:  nbug1 nbug2 nbug3 nbug4 nbug5 nbug6
      1      1      1      3      0      0
grid:  ni npri ntran
      1      1      0
      xl      xr1      pr1      xr2      pr2      xr3      pr3
      1.0000  2.0000  1.0000

```

```

node coordinates (x , integer map) exponent e: -2
  100 200
matl: macro nodal conductivities
      1 0.10000D+02 0.10000D+02
robn: nrob: 1
      jrob: 1
           h      tref
           20.00000 1500.00000
diri: nfix: 1 knum: 0 isor: 0
      jfix: 2
           306.85282
form: system matrix [k + h]; column format * Compare with [V]
      0.00 34.43 -14.43 -----
      0.00 1.00 0.00
data matrix {b} row format
min = 306.85282000 max = 30000.00000000
30000.00 306.85
solv: nodal solution (real variables)
      min = 306.85282000 max = 999.99990877
      1000.00 306.85
norm: f.e. soln. energy semi-norm: e(qh)= 0.23465729D+08
stop

titl: ***** code exercise 4.2, m=2, p=3 *****
type: k n nnode1 refl npri ntran naxi nztr
      1 1 2 1.0 1 0 1 0
prin: nbug1 nbug2 nbug3 nbug4 nbug5 nbug6
      1 1 1 3 0 0
grid: ni npri ntran
      1 1 0
           xl xr1 pr1 xr2 pr2 xr3 pr3
           1.0000 2.0000 1.0000
node coordinates (x , integer map) exponent e: -2
  100 150 200
matl: macro nodal conductivities
      1 0.10000D+02 0.10000D+02
robn: nrob: 1
      jrob: 1
           h      tref
           20.00000 1500.00000
diri: nfix: 1 knum: 0 isor: 0
      jfix: 3
           306.85282
form: system matrix [k + h]; column format * Compare with [S]
      0.00 44.66 -24.66 -----
      -24.66 59.42 -34.76
      0.00 1.00 0.00
data matrix {b} row format

```

```

min =      0.00000000    max =30000.00000000
30000.00    0.00  306.85
solv: nodal solution (real variables)
min = 306.85282000    max = 1000.00106661
1000.00  594.54  306.85
norm: f.e. soln. energy semi-norm:  e(qh)= 0.23465764D+08
stop

titl:      ***** code exercise 4.2, m=1, p=4 *****
type:      k  n  nnode1  refl  npri  ntran  naxi  nztr
          1  1      2  1.0    1      0      1      0
prin: nbug1  nbug2  nbug3  nbug4  nbug5  nbug6
          1      1      1      3      0      0
grid:      ni  npri  ntran
          1      1      0
          xl      xr1      pr1      xr2      pr2      xr3      pr3
          1.0000  2.0000  1.0000
node coordinates (x , integer map)  exponent e: -2
100  200
matl: macro nodal conductivities
      1  0.10000D+02  0.10000D+02
robn: nrob:      1
      jrob:      1
           h      tref
      20.00000 1500.00000
diri: nfix:      1  knum: 0  isor: 0
      jfix:      2
      306.85282
form: system matrix [k + h]; column format * Compare with [V]
      0.00  34.43  -14.43
      0.00  1.00  0.00
data matrix {b} row format
min = 306.85282000    max =30000.00000000
30000.00  306.85
solv: nodal solution (real variables)
min = 306.85282000    max = 1000.00006164
1000.00  306.85
norm: f.e. soln. energy semi-norm:  e(qh)= 0.23465738D+08
stop

```

Appendix H

In search of the SGM basis functions N_1^R and N_2^R , Figure 5.4, in one dimension, I have used MacsymaTM (1988) to derive a set of integrals, via (5.36), that are finite and definite integrable. The symbolic output is given below.

```
(c3) z1:1-x/le$
(c4) z2:x/le$
(c5) x1:z1*(2*z1-1)$
(c6) x2:4*z1*z2$
(c7) x3:z2*(2*z2-1)$
(c8) d1:diff(x1,x)$
(c9) d2:diff(x2,x)$
(c10) d3:diff(x3,x)$
(c11) factr:x1+c*x2+x3$
(c12) kmat:factr*matrix(d1*[d1,d3],d3*[d1,d3])$
(c13) kmxab:factr*matrix(d1*[d2],d3*[d2])$
(c14) kmxba:factr*matrix(d2*[d1,d3])$
(c15) kmxbb:factr*matrix([d2*d2])$
(c16) kmat1:integrate(kmat,x)$
(c17) kmxab1:integrate(kmxab,x)$
(c18) kmxba1:integrate(kmxba,x)$
(c19) kmxbb1:integrate(kmxbb,x)$
(c20) dinv:invert(kmxbb1)$
(c21) bxdinv:kmxab1 . dinv$
(c22) bxc:bxdinv . kmxba1$
(c23) kmatr:kmatrix1-bxc$
(c24) a11:kmatr[1,1]$
(c25) a12:kmatr[1,2]$
(c26) a21:kmatr[2,1]$
(c27) a22:kmatr[2,2]$
(c28) ax1:diff(a11,x)$
(c29) ax2:diff(a12,x)$
(c30) ax3:diff(a21,x)$
(c31) ax4:diff(a22,x)$
(c32) dn1dx:sqrt(ax1)$
(c33) dn2dx:sqrt(ax4)$
(c35) n1r:integrate(dn1dx,x)$
(c36) n2r:integrate(dn2dx,x)$
```

```

(c37) n1r;
/
[
3      2      8
(d39) (i sqrt(- ((576 c - 1728 c + 1728 c - 576) x
]
/

3      2      7
+ (- 3456 c + 10368 c - 10368 c + 3456) 1e x
3      2      2 6
+ (8400 c - 26304 c + 27408 c - 9504) 1e x
3      2      3 5
+ (- 10320 c + 36480 c - 42000 c + 15840) 1e x
3      2      4 4
+ (6400 c - 29620 c + 40680 c - 17460) 1e x
3      2      5 3
+ (- 1600 c + 13600 c - 24960 c + 12960) 1e x
2      6 2      7      8
+ (- 2800 c + 9000 c - 6300) 1e x + (1800 - 1500 c) 1e x - 225 1e )
2      8      2      7
/((2304 c - 4608 c + 2304) x + (- 11520 c + 23040 c - 11520) 1e x
2      2 6      2      3 5
+ (24000 c - 49920 c + 25920) 1e x + (- 26880 c + 61440 c - 34560) 1e x
2      4 4      2      5 3
+ (17200 c - 47040 c + 30240) 1e x + (- 6000 c + 22800 c - 18000) 1e x
2      6 2      7      8
+ (900 c - 6600 c + 7200) 1e x + (900 c - 1800) 1e x + 225 1e )) abs(x) dx)
2
/1e
(c41) grind(d39);
('integrate(sqrt(-(576*c^3-1728*c^2+1728*c-576)*x^8
+(-3456*c^3+10368*c^2-10368*c+3456)*1e*x^7
+(8400*c^3-26304*c^2+27408*c-9504)*1e^2*x^6
+(-10320*c^3+36480*c^2-42000*c+15840)*1e^3*x^5
+(6400*c^3-29620*c^2+40680*c-17460)*1e^4*x^4
+(-1600*c^3+13600*c^2-24960*c+12960)*1e^5*x^3
+(-2800*c^2+9000*c-6300)*1e^6*x^2+(1800-1500*c)*1e^7*x
-225*1e^8)
/((2304*c^2-4608*c+2304)*x^8
+(-11520*c^2+23040*c-11520)*1e*x^7
+(24000*c^2-49920*c+25920)*1e^2*x^6
+(-26880*c^2+61440*c-34560)*1e^3*x^5
+(17200*c^2-47040*c+30240)*1e^4*x^4
+(-6000*c^2+22800*c-18000)*1e^5*x^3
+(900*c^2-6600*c+7200)*1e^6*x^2+(900*c-1800)*1e^7*x
+225*1e^8)) *abs(x),x)) /1e^2$

```

```

(c43) n2r;
/
[
      3      2      10
(d43) i sqrt(- ((576 c - 1728 c + 1728 c - 576) x
]
/
      3      2      9
+ (- 3456 c + 10368 c - 10368 c + 3456) le x

      3      2      2 8
+ (8400 c - 26304 c + 27408 c - 9504) le x

      3      2      3 7
+ (- 10320 c + 36480 c - 42000 c + 15840) le x

      3      2      4 6
+ (6400 c - 29620 c + 40680 c - 17460) le x

      3      2      5 5
+ (- 1600 c + 13600 c - 24960 c + 12960) le x

      2      6 4      7 3      8 2
+ (- 2800 c + 9000 c - 6300) le x + (1800 - 1500 c) le x - 225 le x )

      2      4 8      2      5 7
/((2304 c - 4608 c + 2304) le x + (- 11520 c + 23040 c - 11520) le x

      2      6 6      2      7 5
+ (24000 c - 49920 c + 25920) le x + (- 26880 c + 61440 c - 34560) le x

      2      8 4      2      9 3
+ (17200 c - 47040 c + 30240) le x + (- 6000 c + 22800 c - 18000) le x

      2      10 2      11      12
+ (900 c - 6600 c + 7200) le x + (900 c - 1800) le x + 225 le )) dx
(c44) grind(%);
'integrate(sqrt(-((576*c^3-1728*c^2+1728*c-576)*x^10
+(-3456*c^3+10368*c^2-10368*c+3456)*le*x^9
+(8400*c^3-26304*c^2+27408*c-9504)*le^2*x^8
+(-10320*c^3+36480*c^2-42000*c+15840)*le^3*x^7
+(6400*c^3-29620*c^2+40680*c-17460)*le^4*x^6
+(-1600*c^3+13600*c^2-24960*c+12960)*le^5*x^5
+(-2800*c^2+9000*c-6300)*le^6*x^4+(1800-1500*c)*le^7*x^3
-225*le^8*x^2)
/((2304*c^2-4608*c+2304)*le^4*x^8
+(-11520*c^2+23040*c-11520)*le^5*x^7
+(24000*c^2-49920*c+25920)*le^6*x^6
+(-26880*c^2+61440*c-34560)*le^7*x^5
+(17200*c^2-47040*c+30240)*le^8*x^4
+(-6000*c^2+22800*c-18000)*le^9*x^3
+(900*c^2-6600*c+7200)*le^10*x^2+(900*c-1800)*le^11*x
+225*le^12)),x)$
(c45) quit();

```

Hence, the final indefinite integral expressions for N_1^R and N_2^R are

$$N_1^R(x) = \int_{h_*} \frac{|x|}{h^2} \left(-\frac{A}{B} \right)^{1/2} dx, \quad N_2^R(x) = \int_{h_*} \frac{|x|}{h^2} \left(-\frac{C_1 C_2}{D^2} \right)^{1/2} dx \quad (\text{H.1})$$

$$A = 576(c-1)^3 x^8 - 3456(c-1)^3 h x^7 + 48(c-1)^2 (175c-198) h^2 x^6 \quad (\text{H.2})$$

$$-240(c-1)^2 (43c-66) h^3 x^5 + 20(c-1) (320c^2 - 1161c + 873) h^4 x^4$$

$$-160(c-1) (10c^2 - 75c + 81) h^5 x^3 - 100(28c^2 - 90c + 63) h^6 x^2$$

$$-300(5c-6) h^7 x - 225 h^8$$

$$B = 2304(c-1)^2 x^8 - 11520(c-1)^2 h x^7 + 960(c-1) (25c-27) h^2 x^6 \quad (\text{H.3})$$

$$-3840(c-1) (7c-9) h^3 x^5 + 80(215c^2 - 588c + 378) h^4 x^4$$

$$-1200(5c^2 - 19c + 15) h^5 x^3 + 300(c-6) (3c-4) h^6 x^2$$

$$+900(c-2) h^7 x + 225 h^8$$

$$C_1 = 4cx^2 - 4x^2 - 4chx + 4hx - h^2 \quad (\text{H.4})$$

$$C_2 = (12cx^3 - 12x^3 - 30chx^2 + 30hx^2 + 20cxh^2 - 30xh^2 + 15h^3)^2 \quad (\text{H.5})$$

$$D = 48cx^4 - 48x^4 - 120chx^3 + 120hx^3 + 100ch^2x^2 \quad (\text{H.6})$$

$$-120h^2x^2 - 30ch^3x + 60h^3x - 15h^4$$

Appendix I

 * Matrix perturbation (GMP) algorithm and GWS 1d wave solution comparison

```

****
C=1
*****
      finite elements 1-2-3  code: learn finite elements
*****
titl:  ***** 1d sine wave, m=40, k=1 *****
type:   k  n  nnodel  refl  npri  ntran  naxi  nztr
        1  1      2   1.0    1     0     0     0
prin:  nbug1 nbug2 nbug3 nbug4 nbug5 nbug6
        0     0     0     1     0     0
grid:   ni  npri  ntran
        1    1     0
          xl    xr1    pr1    xr2    pr2    xr3    pr3
        0.0000 40.0000  1.0000
      nem1 nem2 nem3
        40
node coordinates (x real variables)
min =  0.00000000  max =  40.00000000
0.0000000E+00 0.1000000E+01 0.2000000E+01 0.3000000E+01 0.4000000E+01
0.5000000E+01 0.6000000E+01 0.7000000E+01 0.8000000E+01 0.9000000E+01
0.1000000E+02 0.1100000E+02 0.1200000E+02 0.1300000E+02 0.1400000E+02
0.1500000E+02 0.1600000E+02 0.1700000E+02 0.1800000E+02 0.1900000E+02
0.2000000E+02 0.2100000E+02 0.2200000E+02 0.2300000E+02 0.2400000E+02
0.2500000E+02 0.2600000E+02 0.2700000E+02 0.2800000E+02 0.2900000E+02
0.3000000E+02 0.3100000E+02 0.3200000E+02 0.3300000E+02 0.3400000E+02
0.3500000E+02 0.3600000E+02 0.3700000E+02 0.3800000E+02 0.3900000E+02
0.4000000E+02
uvel:  ucon,  selection,  peclet
        1.00000  1  0.00000
        0.100000E+01
init:  nfix:  41  knum:  0  isor:  0
      jfix:   1   2   3   4   5   6   7   8   9  10
      jfix:  11  12  13  14  15  16  17  18  19  20
      jfix:  21  22  23  24  25  26  27  28  29  30
      jfix:  31  32  33  34  35  36  37  38  39  40
      jfix:  41
          0.00000  0.70700  1.00000  0.70700  0.00000 -0.70700
        -1.00000 -0.70700  0.00000  0.00000  0.00000  0.00000
          0.00000  0.00000  0.00000  0.00000  0.00000  0.00000
  
```

```

0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
diri: nfix: 1 knum: 0 isor: 0
      jfix: 1
          0.00000

```

```

inte:
      tzero      tstop      tstep      theta      dqmin      dqmax      tstepi ntsm
      0.0000100.0000 1.0000      0.50      0.0010200.0000 1.0000      20
inte: reached maximum number of timesteps
      time=      20.00000 step= 20 final deltat=      1.00000

```

GMP algorithm solution

```

inte: nodal solution
      min = -1.00000000 max = 1.00000000
      0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
      0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
      0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
      0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
      0.0000000E+00 0.7070000E+00 0.1000000E+01 0.7070000E+00 0.0000000E+00
-0.7070000E+00-0.1000000E+01-0.7070000E+00 0.0000000E+00 0.0000000E+00
      0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
      0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
      0.0000000E+00

```

standard GWS solution

```

      min = -0.58768523 max = 1.08030605
      0.0000000E+00-0.1058629E-01-0.2112842E-01 0.5882384E-01-0.2260192E-01
-0.6791861E-01 0.8376834E-01 0.1719403E-01-0.9775721E-01 0.4212544E-01
      0.6903373E-01-0.6887330E-01-0.5356379E-01 0.4617809E-01 0.1011991E+00
      0.9793513E-01-0.1181284E+00-0.4384882E+00-0.3368071E+00 0.3423955E+00
      0.1017695E+01 0.1080306E+01 0.5492032E+00-0.1092530E+00-0.5115044E+00
-0.5876852E+00-0.4679665E+00-0.3000229E+00-0.1643855E+00-0.7954089E-01
-0.3472143E-01-0.1388418E-01-0.5145293E-02-0.1783476E-02-0.5825800E-03
-0.1804533E-03-0.5329325E-04-0.1505756E-04-0.4110215E-05-0.1051823E-05
-0.3051183E-06

```

stop

titl: ***** 1d 2-node square wave, m=40, k=1 *****

```

:
:

```

```

:
init: nfix: 41 knum: 0 isor: 0
      jfix: 1 2 3 4 5 6 7 8 9 10
      jfix: 11 12 13 14 15 16 17 18 19 20
      jfix: 21 22 23 24 25 26 27 28 29 30
      jfix: 31 32 33 34 35 36 37 38 39 40
      jfix: 41
          0.00000 100.00000 100.00000 0.00000 0.00000 0.00000
          0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
          0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
          0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
          0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
          0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
          0.00000 0.00000 0.00000 0.00000 0.00000
      diri: nfix: 1 knum: 0 isor: 0
            jfix: 1
              0.00000
      inte:
          tzero tstop tstep theta dqmin dqmax tstepi ntsm
          0.0000100.0000 1.0000 0.50 0.0010200.0000 1.0000 20
      inte: reached maximum number of timesteps
            time= 20.00000 step= 20 final deltat= 1.00000

```

GMP algorithm solution

```

inte: nodal solution
      min = 0.00000000 max = 100.00000000
          0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
          0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
          0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
          0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
          0.0000000E+00 0.1000000E+03 0.1000000E+03 0.0000000E+00 0.0000000E+00
          0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
          0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
          0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
          0.0000000E+00

```

standard GWS solution

```

      min = -47.91314697 max = 62.62605667
          0.0000000E+00-0.2162952E+01 0.2866905E+01 0.3144287E+01-0.9643810E+01
          0.3259303E+01 0.1301086E+02-0.1319363E+02-0.1184224E+02 0.2225233E+02
          0.1065273E+02-0.2904177E+02-0.1567047E+02 0.3100444E+02 0.3133294E+02
          -0.1681408E+02-0.4791315E+02-0.2746890E+02 0.1970964E+02 0.5479765E+02
          0.6262606E+02 0.5095157E+02 0.3341944E+02 0.1868440E+02 0.9197583E+01
          0.4073698E+01 0.1649118E+01 0.6175731E+00 0.2159961E+00 0.7110505E-01

```

0.2217450E-01 0.6586676E-02 0.1872188E-02 0.5112514E-03 0.1345940E-03
 0.3426334E-04 0.8457653E-05 0.2027735E-05 0.4752965E-06 0.1058883E-06
 0.2734491E-07

stop

titl: ***** 1d 9-node square wave, m=40, k=1 *****

:
:
:

init: nfix: 41 knum: 0 isor: 0
 jfix: 1 2 3 4 5 6 7 8 9 10
 jfix: 11 12 13 14 15 16 17 18 19 20
 jfix: 21 22 23 24 25 26 27 28 29 30
 jfix: 31 32 33 34 35 36 37 38 39 40
 jfix: 41
 0.00000 100.00000 100.00000 100.00000 100.00000 100.00000
 100.00000 100.00000 100.00000 100.00000 0.00000 0.00000
 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
 0.00000 0.00000 0.00000 0.00000 0.00000

diri: nfix: 1 knum: 0 isor: 0
 jfix: 1
 0.00000

inte:
 tzero tstop tstep theta dqmin dqmax tstepi ntsm
 0.0000100.0000 1.0000 0.50 0.0010200.0000 1.0000 20
 inte: reached maximum number of timesteps
 time= 20.00000 step= 20 final deltat= 1.00000

GMP algorithm solution

inte: nodal solution
 min = 0.00000000 max = 100.00000000
 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
 0.0000000E+00 0.1000000E+03 0.1000000E+03 0.1000000E+03 0.1000000E+03
 0.1000000E+03 0.1000000E+03 0.1000000E+03 0.1000000E+03 0.1000000E+03
 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
 0.0000000E+00

standard GWS solution

```

min = -21.00515747   max = 126.97774506
  0.0000000E+00  0.3344820E+01-0.2422112E+01-0.6266707E+01  0.9394121E+01
  0.7827988E+00-0.7629393E+01  0.1337755E+01-0.1588784E+01  0.7801028E+01
  0.9456984E+01-0.2100516E+02-0.1457919E+02  0.2438894E+02  0.1795300E+02
-0.1057807E+02-0.1195452E+02-0.1038749E+02-0.9781525E+01  0.2386402E+02
  0.7324693E+02  0.9018022E+02  0.7737587E+02  0.7563702E+02  0.9898688E+02
  0.1241331E+03  0.1269777E+03  0.1056378E+03  0.7355589E+02  0.4408456E+02
  0.2326671E+02  0.1101128E+02  0.4741345E+01  0.1879519E+01  0.6926250E+00
  0.2391908E+00  0.7796194E-01  0.2408936E-01  0.7140450E-02  0.1968417E-02
  0.6072538E-03

```

C = 0.5

titl: ***** 1d sine wave, m=40, k=1, C=0.5 *****

:

:

:

inte:

```

      tzero      tstop      tstep      theta      dqmin      dqmax      tstepi ntsm
      0.0000100.0000  0.5000          0.50   0.0010200.0000  0.5000      40

```

inte: reached maximum number of timesteps

```

      time=      20.00000 step=  40 final deltat=      0.50000

```

GMP algorithm solution

inte: nodal solution

```

min = 0.00000000   max = 1.00000000
  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
  0.0000000E+00  0.7070000E+00  0.1000000E+01  0.7070000E+00  0.0000000E+00
-0.7070000E+00-0.1000000E+01-0.7070000E+00  0.0000000E+00  0.0000000E+00
  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
  0.0000000E+00

```

standard GWS solution

```

min = -0.79974413   max = 1.07274234
  0.0000000E+00-0.2793150E-02  0.2595257E-01-0.2702394E-01-0.2062271E-01
  0.6473882E-01-0.2923647E-01-0.5984495E-01  0.7870099E-01  0.1419986E-01
-0.8910570E-01  0.3409457E-01  0.6261398E-01-0.5880002E-01-0.2718577E-01
  0.6834569E-01  0.2616111E-01-0.8352390E-01-0.1762191E+00-0.1088203E+00

```

```

0.3408290E+00 0.9527810E+00 0.1072742E+01 0.4591035E+00-0.3718196E+00
-0.7997441E+00-0.7163568E+00-0.4135656E+00-0.1642138E+00-0.4301585E-01
-0.5733303E-02 0.4341857E-03 0.3145552E-03 0.2935377E-04-0.8693845E-05
-0.1768602E-05 0.2261508E-06 0.6851248E-07-0.7717039E-08-0.2251325E-08
0.3497044E-09

```

```

titl: ***** 1d 2-node square wave, m=40, k=1 *****

```

```

:
:
:

```

```

inte:

```

```

tzero      tstop      tstep      theta      dqmin      dqmax      tstepi ntsm
0.0000100.0000 0.5000      0.50      0.0010200.0000 0.5000      20

```

```

inte: reached maximum number of timesteps

```

```

time=      20.00000 step= 40 final deltat=      1.00000

```

```

*****

```

```

GMP algorithm solution

```

```

*****

```

```

inte: nodal solution

```

```

min =      0.00000000 max = 100.00000000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00 0.1000000E+03 0.1000000E+03 0.0000000E+00 0.0000000E+00
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00

```

```

*****

```

```

standard GWS solution

```

```

*****

```

```

min = -37.66547775 max = 74.61215973
0.0000000E+00 0.9453082E+00 0.6554949E+00-0.3935153E+01 0.3277736E+01
0.3562588E+01-0.8605086E+01 0.1900926E+01 0.1092422E+02-0.9868397E+01
-0.9203817E+01 0.1750138E+02 0.5645607E+01-0.2428185E+02-0.4020144E+01
0.3143132E+02 0.1000576E+02-0.3766548E+02-0.3330590E+02 0.2426318E+02
0.7228424E+02 0.7461216E+02 0.4745000E+02 0.2033545E+02 0.5721041E+01
0.8480701E+00-0.3998509E-01-0.4291908E-01-0.4792931E-02 0.1125345E-02
0.2683643E-03-0.2794802E-04-0.1032718E-04 0.9718087E-06 0.3403943E-06
-0.4620000E-07-0.8982710E-08 0.2181580E-08 0.1120741E-09-0.8265878E-10
0.3665301E-11

```

```

titl: ***** 1d 9-node square wave, m=40, k=1 *****

```

```

:

```

```

:
:
inte:
      tzero      tstop      tstep      theta      dqmin      dqmax      tstepi ntsm
      0.0000100.0000  0.5000      0.50      0.0010200.0000  0.5000      20
inte: reached maximum number of timesteps
      time=      20.00000 step= 40 final deltat=      1.00000

*****
GMP algorithm solution
*****
inte: nodal solution
min =      0.00000000 max = 100.00000000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00 0.1000000E+03 0.1000000E+03 0.1000000E+03 0.1000000E+03
0.1000000E+03 0.1000000E+03 0.1000000E+03 0.1000000E+03 0.1000000E+03
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00

*****
standard GWS solution
*****
min = -22.06464005 max = 122.97941589
0.0000000E+00-0.1912431E+01-0.3648746E+00 0.6046462E+01-0.5794923E+01
-0.3471548E+01 0.9734783E+01-0.3008314E+01-0.6205368E+01 0.4231848E+01
-0.1043617E+00 0.3401774E+01 0.2993134E+01-0.1626273E+02-0.3645260E+01
0.2721183E+02 0.9705849E+01-0.2206464E+02-0.8701237E+01 0.2139544E+02
0.4229312E+02 0.7518185E+02 0.1074649E+03 0.1043458E+03 0.8455638E+02
0.9075620E+02 0.1168840E+03 0.1229794E+03 0.9466992E+02 0.5284787E+02
0.2111709E+02 0.5684563E+01 0.8087190E+00-0.4456435E-01-0.4192732E-01
-0.4538506E-02 0.1102424E-02 0.2584677E-03-0.2711922E-04-0.9995689E-05
0.7299391E-06

```

Two dimensional rotating cone, Section 8.2.10, solutions given below show the time accuracy of the GMP solution for $C_e = 1$, $\theta = 1/2$ on a $N_{node} = 17^2$ mesh.

Initial Condition :

```

min =      0.00000000 max = 100.00000000
exponent =      0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0

```

.....

[illegible]

variable values continued

353

.....

Half Turn :

[illegible]

variable values continued

356

variable values continued

357

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Appendix J

A numerical algorithm is considered to be **monotone** if the corresponding solution does not lead to any non-physical oscillation, Figure J.1. A numerical algorithm is *monotonicity preserving* if the solution maintains the following rules as a function of time t , Hirsch (1992) : (a) no new spatial local extrema is created, (b) the local minimum is non-decreasing, and (c) the local maximum is non-increasing. Hence, a monotonicity preserving algorithm always produces $\{Q\}^{n+1}$ to be monotone if $\{Q\}^n$ is monotone.

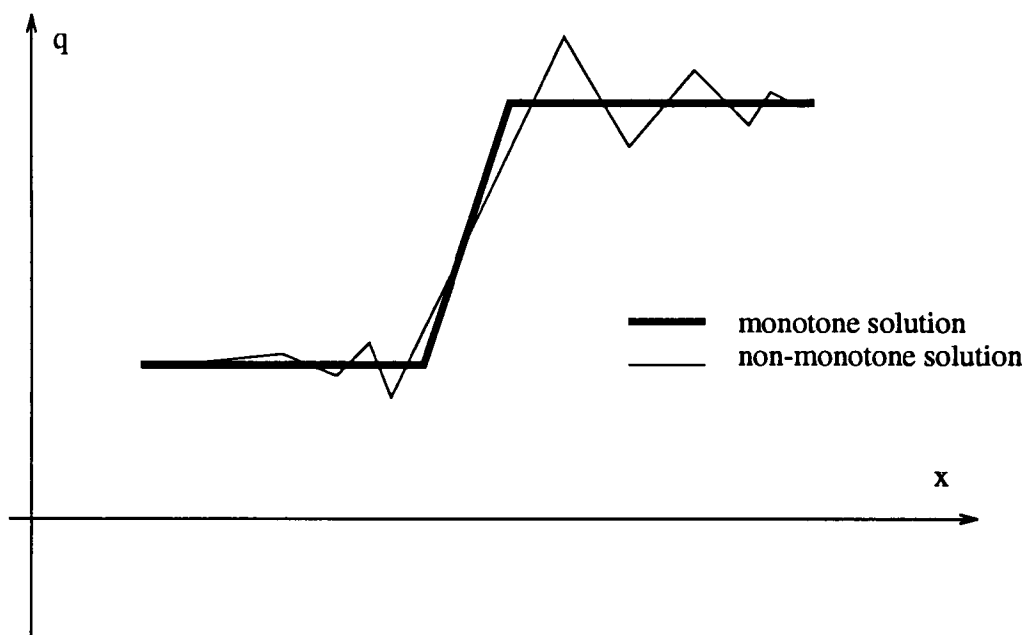


Figure J.1 Characteristic behaviour of monotone and non-monotone numerical solutions.

For a scalar conservation equation, $q_x + f_x = 0$, if the general form of the numerical procedure can be written as $Q_i^{n+1} = H(Q_{i-k}^n, Q_{i-k+1}^n, \dots, Q_{i+k}^n)$, then the algorithm is monotone if H is a monotone increasing function of its arguments, i.e.,

$$\frac{\partial H}{\partial Q_j}(Q_{i-k}, Q_{i-k+1}, \dots, Q_{i+k}) \geq 0 \quad (\text{J.1})$$

Non-oscillatory properties of a solution can be expressed by

- (i) the entropy condition,
- (ii) the monotonicity condition, and
- (iii) the total variation diminishing (TVD) condition.

Definition 1. Two functions u and v that satisfy *entropy condition*

$$\frac{\partial |u - z|}{\partial t} + \frac{\partial}{\partial x} [f(u) - f(z)] \operatorname{sgn} (u - z) \leq 0 \quad (\text{J.2})$$

have the following bound in L_1 norm

$$\|u(x, t) - v(x, t)\|_{L_1} \leq \|u(x, t_0) - v(x, t_0)\|_{L_1} \quad (\text{J.3})$$

Definition 2. If any linear scheme can be written in the form

$$Q_i^{n+1} = \sum_k b_k Q_{i+k}^n \quad (\text{J.4})$$

then the *monotonicity condition* for (J.4) is $b_k \geq 0$.

Definition 3. The total variation (TV) of a function q is defined as

$$\text{Total variation (TV)} = \int \left| \frac{\partial q}{\partial x} \right| dx \quad (\text{J.5})$$

which upon full discretization becomes

$$\text{TV}(Q^n) = \sum_i |Q_{i+1}^n - Q_i^n| \quad (\text{J.6})$$

A numerical algorithm is total variation diminishing (TVD) if

$$\text{TV}(Q^{n+1}) \leq \text{TV}(Q^n) \quad (\text{J.7})$$

Lemma 1. All monotone schemes are TVD.

Lemma 2. All TVD schemes are monotonicity preserving.

Appendix K

Consider a second order elliptic PDE. Hence, $m = 1$ in (4.26) and the convergence in energy semi-norm obeys the inequality,

$$\begin{aligned} E(e^h, e^h) &\equiv E(T, T) - E(T^h, T^h) \\ &\leq Ch^{2k} \end{aligned} \quad (\text{K.1})$$

for a dependent variable T . In (K.1), $E(T^h, T^h)$ is the only data measureable directly from the discretized solution, since in general, the analytical energy semi-norm is not available. However, with proper interpretation, (K.1) always expresses asymptotic convergence for the problem class.

Using the knowledge of geometric series, one may estimate the analytical energy semi-norm from (K.1), and two or more discrete solutions, in the following manner. Since

$$E(T^{h/2}, T^{h/2}) - E(T^h, T^h) = C(h/2)^2 = Ch^2/4 \quad (\text{K.2})$$

if one now continues refining the mesh from h to $h/2^n$ via $h/2, h/4, \dots, h/2^{n-1}$, then

$$\begin{aligned} E(T^{2/h^n}) &= E(T^{2/h^{n-1}}) + \Delta E^n \\ &= E(T^{2/h^{n-2}}) + \Delta E^{n-1} + \Delta E^n \\ &= E(T^h) + \Delta E^1 + \Delta E^2 + \dots + \Delta E^{n-1} + \Delta E^n \end{aligned} \quad (\text{K.3})$$

Since, via (K.2) $\frac{\Delta E^n}{\Delta E^{n-1}} = \frac{1}{4}$, (K.3) becomes

$$E(T^{2/h^n}) = E(T^h) + \Delta E^1 \left(\frac{1 - 1/4^n}{1 - 1/4} \right) \quad (\text{K.4})$$

Finally, as $n \rightarrow \infty$, the corresponding approximation energy semi-norm should match the analytical solution energy, the value of which can be derived as

$$\begin{aligned} E(T^{2/h^\infty}) &= \liminf \left[E(T^h) + \Delta E^1 \left(\frac{1 - 1/4^n}{1 - 1/4} \right) \right] \\ &= E(T^h) + \Delta E^1 \left(\frac{1}{1 - 1/4} \right) \\ &= E(T^h) + \frac{4}{3} \Delta E^1 \end{aligned} \quad (\text{K.5})$$

where $\Delta E^1 = E(T^{h/2}) - E(T^h)$. Equation (K.5) can be used to accurately estimate $E(T^{2/h^\infty})$ provided two discrete solutions T^h , $T^{h/2}$, etc., exist that lie on the asymptotic convergence curve.

Appendix L

Condition number of a matrix is defined as

$$K = \frac{\rho_{max}}{\rho_{min}}, \quad \text{where } \rho \text{ are matrix eigenvalues} \quad (L.1)$$

1D Singular Perturbation problem, Section 8.2.5

System Matrix from SGM algorithm :

```
(C2) SGMA:MATRIX([1,0,0,0,0,0,0,0,0,0,0,0,0,0],
                  [-1,1,0,0,0,0,0,0,0,0,0,0,0,0],
                  [0,-1,1,0,0,0,0,0,0,0,0,0,0,0],
                  [0,0,-1,1,0,0,0,0,0,0,0,0,0,0],
                  [0,0,0,-1,1,0,0,0,0,0,0,0,0,0],
                  [0,0,0,0,-1,1,0,0,0,0,0,0,0,0],
                  [0,0,0,0,0,-1,1,0,0,0,0,0,0,0],
                  [0,0,0,0,0,0,-1,1,0,0,0,0,0,0],
                  [0,0,0,0,0,0,0,-1,1,0,0,0,0,0],
                  [0,0,0,0,0,0,0,0,-1,1,0,0,0,0],
                  [0,0,0,0,0,0,0,0,0,-1,1,0,0,0],
                  [0,0,0,0,0,0,0,0,0,0,-1,1,0,0],
                  [0,0,0,0,0,0,0,0,0,0,0,-1,1,0],
                  [0,0,0,0,0,0,0,0,0,0,0,0,-1,1,0],
                  [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1]))$
```

```
(C3) EIGENVALUES(%)$
```

```
CL_MACSYMA:[SHARE]eigen.fas;1 being loaded.
```

```
(C4) %;
```

```
(D4)                                     [[1], [13]]
                                     Eigenvalue 1; multiplicity 13
```

```
eigenvalue span( 1,1) :
SGM system matrix condition number : 1
```

365

```

(C25) RECTFORM(D10[6]);

(D25)                                0.706903 %I + 0.024
(C26) RECTFORM(D10[7]);

(D26)                                0.024 - 0.865776 %I
(C27) RECTFORM(D10[8]);

(D27)                                0.865776 %I + 0.024
(C28) RECTFORM(D10[9]);

(D28)                                0.024 - 0.499856 %I
(C29) RECTFORM(D10[10]);

(D29)                                0.499856 %I + 0.024
(C30) RECTFORM(D10[11]);

(D30)                                0.024
(C31) ABS(D26);

(D31)                                0.866109
(C32) D10[12];

(D32)                                1
      Eigenvalue span (0.024, 1)

(C34) GWSCONDNO:1/0.024;
      Condition number for the system matrix from GWS algorithm

(D34)                                41.66666

```

Appendix M

Uncertainty Principle

Heisenberg's inequality :

In quantum mechanics, the position x and the momentum p of a particle do not have defined values simultaneously. The uncertainty, or statistical spread in their measured value satisfies the relation

$$\Delta x \Delta p \geq h/2, \quad h \text{ is Planck's constant} \quad (\text{M.1})$$

Zero is an exact number. As the left hand side in (M.1) approaches zero, simultaneous determination of the location (Δx) and the velocity of a particle (included in Δp) becomes impossible. This effect can be observed when some particles are kept at a temperature very close to absolute zero (0° Kelvin) and the corresponding velocities of the particles are determinable very accurately. Hence, the closer it will get to 0°K , the harder it will become to determine the location of a particle. Bose condensate liquid helium is an example. At this condition for a certain critical number of bosons (particles with spin of integer numbers) in a critical space, scientists speculate the formation of a "superatom" by conglomeration of a number of atoms.

Similar inequalities apply to other pairs of dynamic variables. In SGM formulation, the parameter r is determined from (5.55) following the inequality

$$r \geq \Delta x |u| \text{Re}/\mathcal{F} \quad (\text{M.2})$$

So, as $Re \rightarrow 0$ (pure diffusion) or $u \rightarrow \infty$ (pure convection), the r becomes indeterminate. Hence it is increasingly difficult to predict r for a pure diffusion or a pure convection problem.

For low Re , a correction to r in (5.55) is proposed as

$$r \geq \frac{\Delta x |u| Re}{\mathcal{F}} + \frac{m}{Re} \quad \text{where } m \text{ is an integer.} \quad (\text{M.3})$$

Hence, for low Re , corrected form of c in (5.46) is

$$c \geq \frac{|u|}{2} \left(\frac{3h Re}{2} \right) - 1 + \frac{3m}{2 Re} \quad (\text{M.4})$$

For example, at $Re = 50$ and $u = 1$, using $m = 19$ one gets $c = 2.7$ which is used in exercise Table 8.4.

VITA

Subrata Roy (সুপ্রভা রায়), a native of India, graduated from *Jadavpur University*, Calcutta, with a Bachelor of Mechanical Engineering degree in July 1984. He worked four years for private industry in Calcutta. In the fall of 1988, he accepted an assistantship at *The University of Tennessee, Knoxville, Tennessee, USA* and graduated with a Master of Science degree (Engineering Science) in May 1990. Subrata joined the Engineering Science and Mechanics Ph.D. program the following August, and completed the degree in December 1994.

The author is a National Scholar of his country. His undergraduate research work has been published as a paper entitled *Computer Aided Water Turbine Design* (CAWTDEN) in the proceedings of 4th Canadian CAD/CAM & Robotics Conference organized by CIM in Toronto on June 1985. Afterwards, he coauthored several articles in journals and conference proceedings.

Subrata is a free-lance writer for the most widely circulated regional newspaper in India and served as an artist for All India Radio and Door-darshan (TV), Calcutta. He founded a drama group *SAAYAN* in Calcutta in 1984. His first anthology of poems was published in 1986. Subrata was elected as the President of the India Student Association (ISA) in Knoxville and served as the Chairman of the INDIA WEEK 1990 organizing committee in the US.