

**An *h*-Adaptive Control-Volume Finite Element Method for Steady,
Two-Dimensional Fluid Flow and Heat Transfer**

by

David A. Venditti

Department of Mechanical Engineering
McGill University
Montreal, Quebec, Canada

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfilment of the requirements for the degree of
Master of Engineering

© David A. Venditti, Montreal, Canada, 1998.



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-50671-1

Canada

To my mother, and to the memory of my father

Numerical precision is the very soul of science.

- Sir D'Arcy Wentworth Thompson,
Scottish natural scientist, 1917

ABSTRACT

An unstructured-grid, h -adaptive control-volume finite element method (CVFEM) was formulated, implemented, and tested for the simulation of steady, viscous, incompressible fluid flow and heat transfer in arbitrarily shaped two-dimensional domains. The h -adaptive procedure is based on *a posteriori* error estimation using a superconvergent recovery or smoothing technique. The CVFEM is based on a co-located, equal-order formulation that deals directly with the primitive variables. A set of sufficient conditions is presented for guaranteeing that the algebraic approximations to the diffusion terms in the integral conservation equations contribute positively to the coefficients in the discretized equations. An additive-correction multigrid method, based on an adaptive volume-agglomeration technique, was implemented for the purposes of accelerating iterative convergence in the solution of the discretized equations.

An extended Richardson extrapolation technique for unstructured grids is proposed: A discrete solution is obtained from an h -adaptive analysis in which the estimated error is reduced to a predetermined level and uniformly distributed over each element. The resulting mesh is then used as a base grid in constructing a hierarchy of grids via uniform element subdivision.

A closed-form, analytical solution to the steady, two-dimensional, incompressible Navier-Stokes equations was obtained by prescribing a divergence-free velocity field and a consistent pressure field, substituting them into the momentum equations, and deriving the implied volumetric source terms. This analytical solution was used to test several numerical algorithms implemented in this work. Lid-driven, viscous flow in a triangular cavity was also used as a test problem.

RÉSUMÉ

Une méthode adaptative d'éléments finis à volume de contrôle, sur des maillages non-structurés (CVFEM), a été formulée, mise en oeuvre et testée pour la simulation des problèmes d'écoulement incompressible, stationnaire et visqueux et de transfert de chaleur dans des domaines de formes arbitraires en 2 dimensions. La méthode d'adaptation en h est basée sur une estimation d'erreur *a posteriori* en utilisant une technique de récupération superconvergente ou de lissage. La méthode CVFEM est basée sur une formulation d'ordre égal co-localisée qui résoud directement les variables primitives. Une série de conditions suffisantes est présentée afin de garantir que les approximations algébriques des termes de diffusion dans les équations de conservation sous forme intégrée contribuent de manière positive aux coefficients dans les équations discrétisées. Une méthode de correction additive de type multi-grilles, basée sur une technique d'agglomération de volumes adaptatifs, a été mise en oeuvre dans le but d'accélérer la convergence itérative dans la solution des équations discrétisées.

Une technique d'extrapolation de Richardson pour des maillages non-structurés est proposée: une solution discrète est obtenue au moyen d'une analyse des solutions adaptées en h dans laquelle l'erreur estimée est réduite à un niveau prédéterminé et distribuée également sur chaque élément. Le maillage résultant est alors utilisé comme maillage de base pour construire une hiérarchie de maillages au moyen d'une subdivision d'éléments uniformes.

Une solution analytique sous forme close pour les équations Navier-Stokes, incompressibles, stationnaires et à 2 dimensions a été obtenue en spécifiant un champ de vitesse à divergence nulle et un champ de pression régulier, en les substituant en équations de quantité de mouvement et en dérivant les termes sources volumétriques impliqués. Cette solution analytique a été utilisée pour tester plusieurs algorithmes numériques implantés dans ce travail. Un écoulement visqueux dans une cavité triangulaire a aussi été utilisé comme problème test.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis supervisor, Professor B. R. Baliga, for his unwavering support and guidance throughout the course of this work. I feel privileged to have worked with a man of his great integrity, and to have benefited from his vast knowledge and experience.

I would like to thank Jonathan Shewchuk for the use of *Triangle*, a Delaunay-based mesh generator that he developed. *Triangle* was used to create and refine many of the computational grids used in the present work.

During my studies, I have had the opportunity to meet many new friends. To eliminate the risk of forgetting anyone, I would like to simply thank all of them for making my stay at McGill that much more enjoyable.

The support of the Natural Sciences and Engineering Research Council of Canada (NSERC), in the form of a postgraduate scholarship to the author, is gratefully acknowledged. Use of the computing facilities at the Centre de Recherche en Calcul Appliqu  (CERCA) is also greatly appreciated.

I wish to express my thanks to Joseph Slanik and to my brother, Mike, for the use of their personal computers in finalizing the figures in this thesis.

Translation of the abstract from English to French was performed by Claire Poulin and Dr. J. Y. Tr panier. Their help in this regard is very much appreciated.

I would also like to express my special appreciation to Marta for keeping me sane, and for just being the lovely woman that she is.

And finally, a heartfelt thanks goes to my mother, for her perpetual and unconditional love and support. I hereby dedicate this thesis to her, and to the memory of my father.

TABLE OF CONTENTS

| | |
|--|------|
| ABSTRACT | iv |
| RÉSUMÉ | v |
| ACKNOWLEDGEMENTS | vi |
| TABLE OF CONTENTS | vii |
| LIST OF FIGURES | xi |
| LIST OF TABLES | xvii |
| NOMENCLATURE | xix |
| | |
| 1 INTRODUCTION | 1 |
| 1.1 Objectives | 1 |
| 1.2 Motivation and Background | 2 |
| 1.2.1 Grid-Adaptation | 3 |
| 1.2.2 Unstructured Grid Techniques | 3 |
| 1.2.3 The Control-Volume Finite Element Method | 4 |
| 1.2.4 Numerical Uncertainty | 5 |
| 1.3 Literature Review | 7 |
| 1.3.1 The Control-Volume Finite Element Method | 7 |
| 1.3.2 Unstructured Grid Generation | 11 |
| 1.3.2.1 Advancing-Front Method | 11 |
| 1.3.2.2 Delaunay-Based Methods..... | 12 |
| 1.3.2.3 Other Triangulation Methods..... | 12 |
| 1.3.3 Grid-Adaptation Strategies | 13 |
| 1.3.4 Error Estimation | 14 |

| | |
|--|-----------|
| 1.3.4.1 Review of A Posteriori Error Estimates | 15 |
| 1.3.4.2 The Zienkiewicz and Zhu Error Estimator | 16 |
| 1.4 Thesis Outline | 17 |
| 2 CONTROL-VOLUME FINITE ELEMENT METHOD FOR TWO-DIMENSIONAL, STEADY-STATE, INCOMPRESSIBLE FLUID FLOW AND HEAT TRANSFER | 21 |
| 2.1 Governing Equations | 22 |
| 2.2 Domain Discretization | 24 |
| 2.3 Integral Conservation Equation | 25 |
| 2.4 Interpolation Functions | 26 |
| 2.4.1 Diffusion Coefficient, Density and Source Term | 26 |
| 2.4.2 Mass Flow Rates | 27 |
| 2.4.3 Diffusion Term | 27 |
| 2.4.3.1 Sufficient Conditions for Positive Diffusion Contributions to the Coefficients in the Discretized Equations | 29 |
| 2.4.4 Advection Term | 30 |
| 2.4.4.1 Flow-Oriented Upwind Scheme (FLO) | 30 |
| 2.4.4.2 Mass-Weighted Upwind Scheme (MAW) | 32 |
| 2.4.5 Pressure | 34 |
| 2.5 Derivation of the Discretized Equations | 35 |
| 2.5.1 Diffusion Contribution | 35 |
| 2.5.2 Advection Contribution | 36 |
| 2.5.2.1 Flow-Oriented Upwind Scheme (FLO) | 36 |
| 2.5.2.2 Mass-Weighted Upwind Scheme (MAW) | 37 |
| 2.5.3 Source Contribution | 38 |
| 2.5.4 Discretized Advection-Diffusion Equation | 38 |
| 2.5.5 Discretized Momentum Equations | 39 |
| 2.5.6 Interpolation of Velocity Components in Mass-Flux Terms | 40 |
| 2.5.7 Discretized Equation for Pressure | 42 |
| 2.5.8 Boundary Conditions | 43 |
| 2.6 Solution Procedure | 47 |

| | |
|---|-----------|
| 3 ADDITIVE CORRECTION MULTIGRID WITH ADAPTIVE VOLUME AGGLOMERATION | 50 |
| 3.1 Introduction | 50 |
| 3.2 Grid Coarsening | 52 |
| 3.2.1 Volume-Agglomeration Procedure | 53 |
| 3.2.2 Solution-Adaptive Agglomeration | 55 |
| 3.3 Restriction and Prolongation | 55 |
| 3.4 Multigrid Cycle | 57 |
| 3.5 Relaxation Scheme | 58 |
| 4 A POSTERIORI ERROR ESTIMATION AND THE h-ADAPTIVE STRATEGY | 61 |
| 4.1 Recovery | 62 |
| 4.1.1 Global L_2 Projection | 63 |
| 4.1.2 Volume Averaging | 66 |
| 4.2 Error Estimation | 67 |
| 4.2.1 Error Estimator for Scalar Diffusion | 68 |
| 4.2.2 Strain-Rate Error Estimator | 72 |
| 4.2.3 Flow-Angle Indicator | 73 |
| 4.3 Refinement | 76 |
| 4.3.1 Global Error Criterion | 77 |
| 4.3.2 Local Error Criterion | 78 |
| 4.3.3 Element Refinement Parameter | 78 |
| 4.3.4 Multiple Error Estimators | 80 |
| 5 EXTENDED RICHARDSON EXTRAPOLATION | 82 |
| 5.1 The Deferred Approach to the Limit | 82 |
| 5.2 h^2 Extrapolation | 83 |
| 5.3 Generalized Richardson Extrapolation | 86 |
| 5.4 Unequal Grid Spacing and Non-Uniform Grids | 87 |
| 5.5 Extended Richardson Extrapolation | 90 |
| 5.5.1 Hierarchy of Grids | 90 |

| | |
|---|------------|
| 5.5.2 Extrapolation | 91 |
| 5.5.3 Mathematical Justification | 91 |
| 5.5.4 Order of Accuracy of the CVFEM | 92 |
| 6 RESULTS | 95 |
| 6.1 Verification of the Additive Correction Multigrid Solver | 96 |
| 6.1.1 Heat Conduction in an Elliptically Shaped Domain with an Asymmetrically Located Triangular Hole | 96 |
| 6.1.2 Solution to the Poisson Equation with Variable Diffusion Coefficient | 100 |
| 6.2 Analytical Solution to the Navier-Stokes Equations with Body Forces | 102 |
| 6.2.1 Derivation Procedure | 102 |
| 6.2.2 Analytical Flow Field | 102 |
| 6.2.3 Description of the Flow | 104 |
| 6.2.4 Simulations | 104 |
| 6.2.5 Global Energy Norm | 106 |
| 6.3 Verification of the h -Adaptive Procedure | 107 |
| 6.3.1 Recovery Schemes | 107 |
| 6.3.2 Error Estimators | 110 |
| 6.3.2.1 Poisson Equation on a Circular Domain with Prescribed Gaussian Distributions | 111 |
| 6.3.2.2 Effectivity Index | 112 |
| 6.4 Verification of the Extended Richardson Extrapolation Technique | 114 |
| 6.5 Lid-Driven Flow in a Triangular Cavity | 120 |
| 7 CONCLUSION | 164 |
| 7.1 Review of the Thesis and Its Main Contributions | 164 |
| 7.2 Recommendations for Future Work | 168 |
| REFERENCES | 170 |
| APPENDIX A Triangle: An Unstructured Delaunay-Based Mesh Generator ... | A-1 |
| APPENDIX B Derivation of the Sufficient Conditions for Positive Diffusion Contributions to the Coefficients in the Discretized Equations . | B-1 |

LIST OF FIGURES

| | | |
|-----|---|----|
| 1.1 | Discretization of a two-dimensional, planar calculation domain into three-node triangular elements and polygonal control volumes | 20 |
| 1.2 | Local rotated coordinate system (X, Y) used in the FLO scheme for a typical element. The X -axis is aligned with the element-average velocity, \bar{V}_{ave} | 20 |
| 2.1 | Typical patch of elements surrounding (a) an internal node; and (b) a boundary node. The shaded region represents the control volume associated with node i | 49 |
| 2.2 | A typical three-node triangular element and related nomenclature: (a) local rotated coordinate system (X, Y) used in the FLO scheme; and (b) convention for positive unit normal vectors in the derivation of the MAW scheme | 49 |
| 3.1 | Typical two-dimensional mesh. Node P and its neighbour-nodes are illustrated.... | 59 |
| 3.2 | Typical agglomeration pattern for obtaining the second level in the ACM cycle: The original finite element mesh (thin lines); corresponding fine-grid volumes (medium lines); and coarse-grid volumes (thick lines) are superimposed over the same domain | 59 |
| 3.3 | Cycling patterns used in the ACM: V-cycle; and hybrid V-cycle | 60 |
| 4.1 | The patch of elements surrounding node i , and the local node numbering used in the derivation of the global L_2 projection | 81 |
| 4.2 | A typical element and related nomenclature used in the derivation of the flow-angle indicator..... | 81 |
| 5.1 | Element subdivision technique used to construct the grid hierarchy in the extended Richardson extrapolation: (a) sample element from the base grid; (b) corresponding elements in the second grid after subdividing the base grid | |

| | |
|--|-----|
| element into four similar elements; (c) corresponding elements in the third grid | 94 |
| 5.2 Example of four grids in a hierarchy of grids used in the proposed extended Richardson extrapolation technique: (a) the base grid; (b) the second grid; (c) the third grid; (d) the seventh grid in the hierarchy | 94 |
| 6.1 Heat conduction in an elliptically shaped domain with an asymmetrically located triangular hole: (a) Final h -adapted mesh; and (b) surface plot of the computed temperature distribution | 129 |
| 6.2 Heat conduction in an elliptically shaped domain with an asymmetrically located triangular hole. Control volumes corresponding to 3 of the 6 grid levels in a typical ACM cycle: (a) the first grid level (the centroid-median tessellation of the original finite element mesh, shown in Fig. 6.1 (a)); (b) the second grid level; and (c) the sixth and final grid level | 130 |
| 6.3 Heat conduction in an elliptically shaped domain with an asymmetrically located triangular hole. Comparison of the convergence histories of the ACM and SOR-GS solvers on three different grids | 131 |
| 6.4 Solution to the Poisson equation on a Square domain. Contour plots of ϕ for the case of: (a) $\Gamma_\phi = 1$; and (b) $\Gamma_\phi = x + y$ | 132 |
| 6.5 Solution to the Poisson equation on a Square domain: (a) uniform, initial grid used for testing the adaptive volume-agglomeration algorithm; and (b) corresponding control-volume boundaries (that is, the centroid-median tessellation of grid (a)) | 132 |
| 6.6 Solution to the Poisson equation on a Square domain. Agglomeration patterns in the second, third, and fourth ACM grid levels for the case of $\Gamma_\phi = 1$ (left column); and $\Gamma_\phi = x + y$ (right column) | 133 |
| 6.7 Analytical solution to the Navier-Stokes equations with body forces: Streamline patterns for the case of (a) $n=2$; and (b) $n=3$. Other parameters in both cases are: $V_o < 1$; $u_o = v_o = 0$; and $l = 1$ | 134 |

-
- 6.8 Analytical solution to the Navier-Stokes equations with body forces: Surface plots of the velocity components, (a) u and (b) v , for the case of $n=2$; $V_o = -1$; $u_o \approx 1.1468$; $v_o \approx 0.5590$; and $l = 1$ 135
- 6.9 Analytical solution to the Navier-Stokes equations with body forces: (a) Surface plot of the pressure field; and (b) the streamline pattern, for the case of $n=2$; $V_o = -1$; $u_o \approx 1.1468$; $v_o \approx 0.5590$; and $l = 1$ 136
- 6.10 Analytical solution to the Navier-Stokes equations with body forces: (a) The base grid ($k=1$); and (b) the second grid ($k=2$), used in the simulations described in Section 6.2.4. There were 10 grids [$k=1,2,\dots,10$] in all 137
- 6.11 Verification of the h -adaptive procedure. Convergence histories of the relative error in the CVFEM gradients, \bar{e}_G^h , and the recovered gradients, \bar{e}_G^L (global L_2 projection) and \bar{e}_G^V (volume averaging), with respect to the characteristic grid-size parameter, h_c . These plots correspond to the following gradients: (a) $G = \partial u / \partial x$; (b) $G = \partial u / \partial y$; (c) $G = \partial v / \partial x$; (d) $G = \partial v / \partial y$, at the nodal location, $(x,y) = (0.125, 0.125)$. The Reynolds number in this case is $Re = 1$ 138
- 6.12 Verification of the h -adaptive procedure. Convergence histories of the relative error in the CVFEM gradients, \bar{e}_G^h , and the recovered gradients, \bar{e}_G^L (global L_2 projection) and \bar{e}_G^V (volume averaging), with respect to the characteristic grid-size parameter, h_c . These plots correspond to the following gradients: (a) $G = \partial u / \partial x$; (b) $G = \partial u / \partial y$; (c) $G = \partial v / \partial x$; (d) $G = \partial v / \partial y$, at the nodal location, $(x,y) = (0.375, 0.625)$. The Reynolds number in this case is $Re = 1$ 139
- 6.13 Verification of the h -adaptive procedure. Convergence histories of the relative error in the CVFEM gradients, \bar{e}_G^h , and the recovered gradients, \bar{e}_G^L (global L_2 projection) and \bar{e}_G^V (volume averaging), with respect to the characteristic grid-size parameter, h_c . These plots correspond to the gradient, $G = \partial u / \partial x$, at the nodal location, $(x,y) = (0.125, 0.125)$, for the following

| | |
|--|-----|
| Reynolds numbers: (a) $Re=0.001, 0.01, 0.1$, and 1 ; (b) $Re=10$; (c) $Re=100$; (d) $Re=1000$ | 140 |
| 6.14 Poisson equation on a circular domain with prescribed Gaussian distributions: (a) Surface plot of the computed ϕ distribution; and (b) the corresponding h -adapted mesh..... | 141 |
| 6.15 Poisson equation on a circular domain with prescribed Gaussian distributions: Comparison of initial coarse-grid solution (left column) and final h -adaptive solution (right column): (a) computational grids; (b) corresponding contour plots of the computed solution, ϕ ; and (c) bar chart of the relative error distribution, $ \bar{e}_\phi $, in the domain (note: the length of each bar is proportional to the relative error at a node; the y -axis points into the page). | 142 |
| 6.16 Verification of the strain-rate error estimator: Convergence histories of the effectivity index, κ , with respect to the characteristic grid-size, h_c , using the global L_2 projection (κ^L - filled markers); and volume averaging (κ^V - hollow markers), as the recovery schemes. Results are presented for the following Reynolds numbers: $Re=0.001, 0.01, 0.1, 1, 10, 100$, and 1000 | 143 |
| 6.17 Verification of the extended Richardson extrapolation technique: Convergence histories of the relative error in the velocity components (\bar{e}_v - hollow markers; \bar{e}_v - filled markers), with respect to the characteristic grid- size, h_c , at the nodal location, $(x, y)=(0.125, 0.125)$. Results are presented for the following Reynolds numbers: $Re=0.001, 0.01, 0.1, 1, 10, 100$, and 1000..... | 144 |
| 6.18 Verification of the extended Richardson extrapolation technique: Convergence histories of the relative error in the pressure, \bar{e}_p , with respect to the characteristic grid-size, h_c , at the nodal location, $(x, y)=(0.125, 0.125)$. Results are presented for the following Reynolds numbers: $Re=0.001, 0.01,$ $0.1, 1, 10, 100$, and 1000 | 145 |
| 6.19 Verification of the extended Richardson extrapolation technique: Convergence histories of the relative error in the global strain-rate energy | |

| | |
|---|-----|
| norm, $\bar{e}_{\ _c\ }$ with respect to the characteristic grid-size, h_c . Results are presented for the following Reynolds numbers: $Re=0.001, 0.01, 0.1, 1, 10, 100$, and 1000 | 146 |
| 6.20 Lid-driven flow in a triangular cavity: Schematic representation of the geometry, and related nomenclature | 147 |
| 6.21 Lid-driven flow in a triangular cavity: Computed streamline patterns for (a) $Re=50$; (b) $Re=100$; (c) $Re=400$; and (d) $Re=800$ | 148 |
| 6.22 Lid-driven flow in a triangular cavity. h -adaptive simulations for $Re=50$: (a) final refined mesh using the strain-rate error estimator (with a target error of $\bar{\eta}_c = 0.1$) alone to drive the adaptive procedure; (b) final refined mesh using the strain-rate error estimator ($\bar{\eta}_c = 0.1$) in combination with the flow-angle indicator ($\bar{\eta}_\psi = 0.15$) to drive the adaptive procedure; and (c) the computed streamline pattern obtained on grid (b). | 149 |
| 6.23 Lid-driven flow in a triangular cavity. h -adaptive simulations for $Re=800$: (a) final refined mesh using the strain-rate error estimator (with a target error of $\bar{\eta}_c = 0.1$) alone to drive the adaptive procedure; (b) final refined mesh using the strain-rate error estimator ($\bar{\eta}_c = 0.1$) in combination with the flow-angle indicator ($\bar{\eta}_\psi = 0.15$) to drive the adaptive procedure; and (c) the computed streamline pattern obtained on grid (b)..... | 150 |
| 6.24 Lid-driven flow in a triangular cavity. h -adaptive simulation for $Re=50$: Contour plots of (a) the x -component of velocity, u ; (b) the y -component of velocity, v ; (c) the pressure field, p ; and (d) a vector plot of the velocity field, $\bar{\mathbf{V}}$ | 151 |
| 6.25 Lid-driven flow in a triangular cavity. h -adaptive simulation for $Re=800$: Contour plots of (a) the x -component of velocity, u ; (b) the y -component of velocity, v ; (c) the pressure field, p ; and (d) a vector plot of the velocity field, $\bar{\mathbf{V}}$ | 152 |
| 6.26 Lid-driven flow in a triangular cavity. Results obtained in the present work are compared with the published solutions of Jyotsna and Vanka [152]: The x -component of velocity, u , along the centerline ($x=0$) of the cavity (horizontal | |

| | |
|---|-----|
| axis) is plotted versus y (vertical axis) for the case of (a) $Re=50$; and (b) $Re=800$. (Note: the “present extrapolated” results correspond to the 3-grid extrapolations using grids 3-5 in Tables 6.7 (b) and 6.8 (b), and the “present adaptive” results correspond to grid 5 in Tables 6.7 (b) and 6.8 (b)) | 153 |
| 6.27 Lid-driven flow in a triangular cavity. Final refined meshes resulting from h -adaptive simulations using the <i>strain-rate error estimator</i> , alone, to drive the adaptive procedure: h -adaptive simulations were performed for $Re=50$ (top row); and $Re=800$ (bottom row) with three different specified target errors of, $\bar{\eta}_e = 0.2, 0.15$, and 0.10 . The minimum-area constraint was set to $(\Omega_e)_{min} / \Omega = 5 \times 10^{-5}$ in these simulations | 154 |
| 6.28 Lid-driven flow in a triangular cavity. Final refined meshes resulting from h -adaptive simulations using the <i>flow-angle indicator</i> , alone, to drive the adaptive procedure: h -adaptive simulations were performed for $Re=50$ (top row); and $Re=800$ (bottom row) with three different specified target errors of, $\bar{\eta}_{\psi} = 0.3, 0.2$, and 0.1 . The minimum-area constraint was set to $(\Omega_e)_{min} / \Omega = 5 \times 10^{-5}$ in these simulations | 155 |
| B.1 The patch of elements surrounding a typical interior node, i , and related nomenclature. The dashed lines represent control-volume boundaries | B-8 |
| B.2 The k^{th} element surrounding node i , and related nomenclature | B-9 |
| B.3 The patch of elements surrounding a typical boundary node, i , and related nomenclature. The dashed lines represent control-volume boundaries | B-9 |

LIST OF TABLES

| | | |
|-----|---|-----|
| 2.1 | Specific forms of the general advection-diffusion equation | 23 |
| 4.1 | Specific norms represented by $\ \mathbf{u}\ $ and $\ \mathbf{e}_u\ $ | 77 |
| 6.1 | Heat conduction in an elliptically shaped domain with an asymmetrically located triangular hole: Specifications of the three grids used in the ACM iterative convergence tests | 156 |
| 6.2 | Heat conduction in an elliptically shaped domain with an asymmetrically located triangular hole: ACM performance is compared with that of the standard SOR-GS solver. CPU times and total number of work units are presented | 156 |
| 6.3 | Verification of the extended Richardson extrapolation technique. This table presents a series of extrapolations performed on the x -component of velocity, u , at the nodal location, $(x, y) = (0.125, 0.125)$, <i>with</i> the linear term included in the associated series expansion: In (a), the logarithm of the relative errors in the CVFEM values, u^h , are presented for each of the simulations described in Section 6.2.4; Below this, the logarithm of the relative errors in the extrapolated values, \bar{u}_o , corresponding to (b) 2-grid; (c) 3-grid; and (d) 4-grid extrapolations, are presented..... | 157 |
| 6.4 | Verification of the extended Richardson extrapolation technique. This table presents a series of extrapolations performed on the x -component of velocity, u , at the nodal location, $(x, y) = (0.125, 0.125)$, <i>without</i> the linear term included in the associated series expansion: In (a), the logarithm of the relative errors in the CVFEM values, u^h , are presented for each of the simulations described in Section 6.2.4; Below this, the logarithm of the relative errors in the extrapolated values, \bar{u}_o , corresponding to (b) 2-grid; (c) 3-grid; and (d) 4-grid extrapolations, are presented..... | 158 |

| | |
|--|-----|
| 6.5 Verification of the extended Richardson extrapolation technique. This table presents a series of extrapolations performed on the global strain-rate energy norm, $\ \boldsymbol{\varepsilon}\ $, with the linear term included in the associated series expansion: In (a), the logarithm of the relative errors in the CVFEM values, $\ \boldsymbol{\varepsilon}\ ^h$, are presented for each of the simulations described in Section 6.2.4; Below this, the logarithm of the relative errors in the extrapolated values, $\ \overline{\boldsymbol{\varepsilon}}\ _o$, corresponding to (b) 2-grid; (c) 3-grid; and (d) 4-grid extrapolations, are presented | 159 |
| 6.6 Verification of the extended Richardson extrapolation technique. This table presents a series of extrapolations performed on the global strain-rate energy norm, $\ \boldsymbol{\varepsilon}\ $, without the linear term included in the associated series expansion: In (a), the logarithm of the relative errors in the CVFEM values, $\ \boldsymbol{\varepsilon}\ ^h$, are presented for each of the simulations described in Section 6.2.4; Below this, the logarithm of the relative errors in the extrapolated values, $\ \overline{\boldsymbol{\varepsilon}}\ _o$, corresponding to (b) 2-grid; (c) 3-grid; and (d) 4-grid extrapolations, are presented | 160 |
| 6.7 Lid-driven flow in a triangular cavity: Computed locations of the primary and secondary eddy centres for $Re=50$. This table presents CVFEM values and corresponding extended Richardson extrapolations for (a) a hierarchy of grids constructed from a <i>uniform</i> initial grid; and (b) a hierarchy of grids constructed from a <i>solution-adapted</i> initial grid..... | 161 |
| 6.8 Lid-driven flow in a triangular cavity: Computed locations of the primary and secondary eddy centres for $Re=800$. This table presents CVFEM values and corresponding extended Richardson extrapolations for (a) a hierarchy of grids constructed from a <i>uniform</i> initial grid; and (b) a hierarchy of grids constructed from a <i>solution-adapted</i> initial grid..... | 162 |
| 6.9 Lid-driven flow in a triangular cavity: Computed locations of the primary and secondary eddy centres for $Re=50$ and 800. Present results are compared with the published solutions of Jyotsna and Vanka [152] | 163 |

NOMENCLATURE

| | |
|-----------------------------------|---|
| a_i^P, a_{nb}^P, b_i^P | coefficients in the discretized pressure equation |
| a_i^u, a_{nb}^u, b_i^u | coefficients in the discretized x -momentum equation |
| a_i^v, a_{nb}^v, b_i^v | coefficients in the discretized y -momentum equation |
| $a_i^\phi, a_{nb}^\phi, b_i^\phi$ | coefficients in the discretized equation for ϕ |
| a, b, c | integration points within an element |
| d^u, d^v | pressure-gradient coefficients in the <i>momentum interpolation scheme</i> |
| e_ϕ | absolute error (that is, the global truncation error) in the CVFEM approximation of ϕ |
| \bar{e}_ϕ | relative error (that is, the relative global truncation error) in the CVFEM approximation of ϕ |
| $\ e_{\bar{q}''}\ $ | global error norm associated with \bar{q}'' |
| $\ e_\epsilon\ $ | global error norm associated with ϵ |
| $\ e_{\bar{\psi}}\ $ | global error norm associated with $\bar{\psi}$ |
| G | represents a particular gradient or partial derivative |
| \tilde{G} | recovered gradients |
| \check{G} | CVFEM gradients |
| h | typical discretization length (for example, grid spacing or the length of the side of an element) |
| h_c | characteristic grid-size parameter |
| h | heat transfer coefficient |
| $\hat{i}, \hat{j}, \hat{k}$ | Cartesian unit vectors in the x -, y -, and z -coordinate directions, respectively |
| $\bar{\mathbf{J}}_A$ | advection flux |
| $\bar{\mathbf{J}}_D$ | diffusion flux |
| \dot{m} | mass flow rate |
| \bar{n} | unit normal vector |

| | |
|--------------------------|--|
| $\hat{\mathbf{N}}$ | normal vector |
| N_e | total number of elements in the computational domain |
| N_n | total number of nodes in the computational domain |
| p | pressure |
| $\bar{\mathbf{q}}''$ | heat flux vector |
| $\ \bar{\mathbf{q}}''\ $ | global energy norm associated with $\bar{\mathbf{q}}''$ |
| $\tilde{\mathbf{R}}$ | tangent vector |
| S_c, S_p | coefficients in the linearization of S , |
| S_u, S_v, S_ϕ | volumetric source terms in the x -momentum, y -momentum, and general advection-diffusion equation (equation for ϕ), respectively |
| u, v | x -component and y -component of velocity, respectively |
| \hat{u}, \hat{v} | pseudo-velocity components used in the <i>momentum interpolation scheme</i> |
| u'', v'' | velocity components used to interpolate the velocity in the mass-flux terms |
| $\hat{\mathbf{V}}$ | velocity vector |
| $\hat{\mathbf{V}}_{ave}$ | element-average velocity vector |
| x, y | Cartesian coordinates (global) |
| X, Y | local Flow-oriented coordinates in the <i>FLO scheme</i> |

Greek Symbols

| | |
|--------------------------------|---|
| $\beta,$ | element refinement parameter |
| ∂ | partial derivative operator |
| $\partial\Omega$ | domain boundary |
| $\bar{\nabla}$ | gradient operator |
| $\boldsymbol{\varepsilon}$ | strain-rate tensor |
| $\ \boldsymbol{\varepsilon}\ $ | global energy norm associated with $\boldsymbol{\varepsilon}$ |
| Γ_ϕ | diffusion coefficient associated with ϕ |
| μ | dynamic viscosity |
| ϕ | transported scalar |
| $\Phi_{II,i}$ | uniform correction to each fine-grid volume, i , within the coarse grid volume, I . |
| ρ | mass density |

| | |
|------------------|--|
| $\bar{\psi}$ | flow-angle gradient field |
| $\ \bar{\psi}\ $ | global energy norm associated with $\bar{\psi}$ |
| Σ | summation operator |
| π | ratio of the circumference of a circle to its diameter |
| θ | flow-angle scalar field |
| Ω | total area of the computational domain |
| Ω_e | area of an element |
| Ω_I | volume (area times unit depth) of control volume I |
| Λ | functional used in the <i>global L_2 projection</i> |
| η | ratio of the global error norm to global energy norm |
| $\bar{\eta}$ | user-specified target value of η |
| ξ_g | global error parameter (Chapter 4) |
| ξ_j | local error parameter (Chapter 4) |
| ξ | exponential variable used in the <i>FLO scheme</i> (Chapter 2) |

Subscripts

| | |
|-----------|--|
| a, b, c | refers to integration points within an element |
| ave | refers to element averaged quantity |
| e | refers, locally, to an element |
| i | refers, globally, to a node or its associated control volume |
| j | refers, globally, to an element |
| nb | refers to the nodes neighbouring a node |
| o | refers to the centroid of an element |
| s | refers to the elements surrounding a node |

Superscripts

| | |
|-----------------|--|
| p, u, v, ϕ | corresponds to pressure, u , v , and ϕ , respectively (Chapter 2) |
| p | order of the element-based polynomial interpolation functions (for example, $p=1$ corresponds to a linear function) (Chapter 4 and 5). |
| m | refers to mass-flux related velocity (Chapter 2) |
| m | order of accuracy of the numerical method (Chapter 5) |

Chapter 1

INTRODUCTION

1.1 Objectives

The main goal of the present study is to formulate, implement, and test an unstructured-grid h -adaptive control-volume finite element method (CVFEM) for the simulation of steady, viscous, incompressible fluid flow and heat transfer in arbitrarily shaped two-dimensional domains. The adaptive-grid procedure is based on *a posteriori* error estimation using a superconvergent recovery technique. The method uses an amalgamation of concepts and techniques developed by Zienkiewicz and Zhu [1-4], Zienkiewicz *et al.* [5-8], Babuska and Rheinboldt [9-10], Babuska *et al.* [11-14], Ainsworth and Oden [15], Ainsworth *et al.* [16], Onate and Bugeda [17], Wu *et al.* [18], and others [19-27].

The CVFEM presented in this thesis is based on concepts originally proposed by Baliga [28], Baliga and Patankar [29], and Prakash and Patankar [30], and later enhanced and/or extended by Raw [31], Hookey [32], Saabas [33], Masson [34], and others [35-38]. The co-located, equal-order formulation deals directly with the primitive dependent variables such as the velocity components, pressure, and temperature. The computational domains are discretized using unstructured grids composed of three-node triangular elements. Most of these meshes are conforming Delaunay triangulations [39,40] with specified minimum angle constraints.

Another aim of this thesis is to develop and implement an algebraic multigrid solver [41-50] for the purposes of accelerating convergence in the solution process. The restriction and prolongation operators are those implied by the additive correction multigrid (ACM) [41-43] method. The adaptive volume-agglomeration technique used to define the series

of grid levels in the ACM cycle is similar to those proposed by Raw [41] and Elias *et al.* [42].

A further goal is to derive a complex recirculating flow with closed-form analytical expressions for the velocity components and pressure, and then to use this analytical flow field for the purposes of testing and benchmarking several numerical algorithms developed in this work. This analytical flow field is an exact solution to the steady, two-dimensional, incompressible Navier-Stokes equations with appropriately added volumetric body force terms in the momentum equations. The derivation proceeds as follows: (i) prescribe a divergence-free velocity field; (ii) substitute this velocity field into the momentum equations; (iii) prescribe a consistent pressure field; and (iv) derive the implied volumetric source terms (volumetric body force terms). This analytical solution is used, in the present study, to test two recovery schemes and a strain-rate error estimator employed in the adaptive procedure. Since the exact solution to the continuous problem is known, the true error in the corresponding CVFEM simulation is also known and can be used to assess the effectiveness of the error estimator.

The analytical flow field is also used to verify an extended Richardson extrapolation scheme for unstructured grids. This scheme is based on the original Richardson extrapolation [51,52] for uniformly spaced structured grids, also known as h^2 extrapolation, the deferred approach to the limit, or iterated extrapolation [53]. In the extended scheme, an unstructured grid is uniformly refined via element subdivision such that each subsequent grid can be uniquely described by a single characteristic grid-size parameter, h_c , relative to the other grids. A discrete solution is then obtained on each of the newly refined grids. By using (i) this hierarchy of grids; (ii) the corresponding discrete solutions; and (iii) a truncated Taylor series expansion in h_c , an extrapolation can be performed to estimate the grid-independent numerical solution to the corresponding continuous problem.

1.2 Motivation and Background

The advent of high-speed digital computers coupled with the development of efficient and accurate numerical algorithms has expanded the application of computational fluid

dynamics (CFD) from the realm of pure research into virtually all areas of applied fluid dynamics and heat transfer, as well as other multidisciplinary applications [54–56,128,155]. The importance of CFD as a practical, and often necessary, tool in engineering design is becoming increasingly evident. Nevertheless, as the field of CFD approaches a mature state, there are still countless issues in algorithm development and mathematical modeling that need to be addressed. This thesis is primarily concerned with some of the algorithm development issues mentioned in the following subsections.

1.2.1 Grid-Adaptation

Adaptive-grid strategies based on *a posteriori* error estimation are becoming increasingly important in the numerical simulation process because they provide effective means of optimizing numerical solution methods, thereby enhancing the scope of their applications. Fluid flow and heat transfer phenomena in engineering and the environment often involve irregular geometries that include regions with large gradients. Simulating these regions typically requires high mesh densities in order to avoid excessive truncation errors, while more quiescent regions require relatively coarser mesh densities to achieve the same level of error. While uniformly increasing the mesh density generally yields a more accurate solution, it is at the unnecessary cost of increased computing time and memory requirements. Thus, the key to efficient, economic and accurate simulations is not merely the number of nodes and elements used in an analysis, but also their *placement* within the domain for any given problem. An effective way of *optimally* placing nodes and elements where needed is via automatic mesh adaptation during the solution process, where mesh optimality is defined in terms of minimizing and/or controlling the estimated truncation error.

1.2.2 Unstructured Grid Techniques

A traditional mode of discretizing the computational domain is to employ structured or block-structured grids that rely on regular arrays of quadrilateral or hexahedral cells in two or three dimensions, respectively. These grids are usually quite simple to generate for regular-shaped domains, however, they become exceedingly difficult to construct as the domains become more and more complicated. Unstructured grids employing triangles in

two dimensions or tetrahedra in three dimensions have emerged as a viable alternative to structured grids, primarily due to their well-suited ability to discretize arbitrarily complex geometries while providing a natural setting for the implementation of adaptive-grid techniques. Although adaptive strategies for structured grids have been successfully implemented by numerous researchers, their strategies have the inherent disadvantages of requiring that nodes be confined to the intersections of ordered gridlines; and in general, that non-automatic, and often tedious, grid generation procedures be used to discretize complex domains. Conversely, unstructured-grid implementations have the potential of providing a completely automatic grid generation facility for arbitrary configurations, eliminating the need for any human interaction. Furthermore, nodes in an unstructured mesh can be inserted, removed or repositioned without the constraint of having them ordered along structured sets of gridlines.

The efficiency of unstructured grid solvers has generally trailed the efficiency of their structured grid counterparts. Recently, however, new and efficient iterative solvers for unstructured grids, such as the algebraic multigrid method [41-50], have consistently delivered fast convergence rates with minimal memory overheads. The improvement of these new solution algorithms, combined with the benefits of adaptive meshing, have made unstructured grids the preferred choice for simulating many practical fluid flow and heat transfer problems in engineering [56]. In fact, a panel of grid generation experts at a NASA sponsored workshop (“Future Directions in Surface Modeling and Grid Generation”, Dec 5-7, 1989) predicted that unstructured grid methods will eventually replace overset or patched-structured grid techniques as the only surviving technology in the domain discretization process for practical CFD applications [57]. Whether this prediction materializes or not, the development of efficient and robust unstructured grid generation and solution algorithms remains a significant ongoing pursuit in the field of CFD.

1.2.3 The Control-Volume Finite Element Method

The control-volume finite element method (CVFEM) successfully combines attractive features traditionally associated with the finite volume method (FVM) and the finite element method (FEM). The CVFEM possesses the conservative property associated

with the FVM, in that its solutions satisfy local and global conservation requirements irrespective of the grid size. Furthermore, the CVFEM also provides the geometric flexibility traditionally associated with the FEM. This latter characteristic allows for CVFEM formulations to be particularly well suited for the implementation of unstructured h -adaptive procedures such as those more commonly employed in finite element analyses.

Following the views put forward by Finlayson [58] and Zienkiewicz [59], the FVM, FEM, and CVFEM can all be perceived as being particular cases of the method of weighted residuals. The formulation of a CVFEM for fluid flow and heat transfer typically involves five basic steps [60]: (i) discretization of the calculation domain into elements; (ii) construction of control-volume surfaces around each node in the finite element mesh; (iii) prescription of element-based interpolation functions for the dependent variables and thermophysical properties; (iv) use of the control-volume-based method of weighted residuals and an element-by-element procedure to derive and assemble algebraic approximations to the governing equations; and (v) prescription of a procedure to solve the discrete algebraic equations.

The CVFEM and other similar methods, which attempt to combine finite volume and finite element approaches, are becoming more and more popular in the simulation of fluid flow and heat transfer problems. Many such methods can be found in the published literature including the works of Baliga [60], McCormick [61], Jameson and Mavriplis [62], and Arminjon *et al.* [63]. The advantages and potential of these methods will likely ensure their continued development and application well into the future of CFD.

1.2.4 Numerical Uncertainty

The quantification of uncertainty in numerical simulations of fluid flow and heat transfer is of paramount importance to numerical analysts and researchers in CFD. It is generally accepted that qualitative assessments of accuracy based on intuition or experience should always be supplemented by a quantitative assessment. The editors of the ASME *Journal of Fluids Engineering* have even seen fit to publish a brief policy statement [64] requiring at least minimal attention to the quantification of numerical accuracy. More recently,

other journals have adopted similar policies (ASME Editorial Board [65], AIAA [66], Gresho and Taylor [67]). Following the taxonomy adopted by Boehm [68], Blottner [69], and Roache [53], there is an important distinction to be made between *validation* and *verification*. Validation can be described in simple terms as “solving the right equations”, while verification can be interpreted as “solving the equations right”. The task in the validation process is to determine whether or not a set of partial differential equations (PDEs) adequately describes the corresponding physical processes of interest. In the verification process, on the other hand, one must demonstrate convincingly that the numerical method is solving the specified set of PDEs correctly. Or, more precisely, that as some measure of the discretization (such as the grid spacing or element size) approaches zero, the numerical method produces a solution to the continuum PDEs.

The verification of structured-grid formulations has been commonly addressed in the published literature by reporting grid-convergence studies via Richardson extrapolation [53]. However, the original extrapolation procedure, proposed by Richardson [51,52], cannot be applied to unstructured grids as such. An attempt is made, in the present work, to address the verification issue for unstructured grid formulations. In particular, an *extended* Richardson extrapolation scheme for unstructured grids is described, and its applicability to various problems is demonstrated.

Another desirable feature in the verification process would be to have an exact analytical solution to the PDEs being solved for comparison with the discrete numerical solution. It is well known that there are only a hand full of closed-form analytical solutions to the Navier-Stokes equations, and they are all for simplified flow cases. However, if appropriate volumetric body-force terms are added to the momentum equations, closed-form analytical solutions can indeed be extracted from the resulting equations with relatively little effort. One can, therefore, in principle, propose expressions for the velocity components and pressure corresponding to flow fields of arbitrary complexity. Then, with the prescription of suitable volumetric source terms in the momentum equations, these proposed velocity and pressure fields become analytical solutions to the *full* Navier-Stokes equations with no simplifying assumptions. This procedure is illustrated and used, in this thesis, for the purposes of verifying different recovery techniques in the *h*-adaptive scheme, and for testing the extended Richardson

extrapolation scheme mentioned above. The process of obtaining analytical solutions to these body-force-driven flow problems is relatively straight forward, and the benefits of having these expressions for verification purposes is obvious.

1.3 Literature Review

1.3.1 The Control-Volume Finite Element Method

In the mid-1970s, the FVMs of Chorin [70], Patankar and Spalding [71], and Raithby [72] were successfully being applied to many types of complex fluid flow and heat transfer phenomena, however, their application was limited to problems involving regularly shaped calculation domains. Conversely, the FEMs of Zienkiewicz [73], Oden [74], and Taylor and Hood [75], generally lagged behind their FVM counterparts in terms of the complexity of the fluid flow phenomena being simulated. However, due to the use of finite element grids, they were able to handle highly complex irregularly shaped calculation domains. The motivation to develop CVFEMs arose out of the need to extend the capabilities of FVMs to problems involving irregular geometries.

The first CVFEM for advection-diffusion problems was likely proposed by Baliga [28], and Baliga and Patankar [29]. In their two-dimensional formulation, the calculation domain was discretized into three-node triangular elements. Control-volume surfaces were constructed around each node in the finite element mesh by joining the centroids of each triangle with the midpoints of their respective sides. Collectively, these newly created line segments within each element formed the faces of the control-volumes surrounding each node. This type of construction is also termed the median dual of the triangulation, or the centroid-median tessellation. An example of the domain discretization is presented in Fig. 1.1. The transported scalar within each element was interpolated using a flow-oriented upwind (FLO) function. This same function was used to interpolate the scalar in both the diffusion and advection terms of the governing equations. The function is derived from a simplified one-dimensional advection-diffusion equation written with respect to a local flow-oriented coordinate system, within the element, in which one of the axes is aligned with the element-average velocity vector,

\bar{V}_{ave} , as shown in Fig. 1.2. It automatically responds to an element-based Peclet number, which indicates the relative strengths of advection and diffusion within the element. Some of the key ideas behind this interpolation function were obtained from the skew upwind difference scheme (SUDS) of Raithby [72]. Similar ideas were also used in the streamline upwind Petrov-Galerkin (SUPG) method proposed by Brooks and Hughes [76] in the context of FEMs.

Although this CVFEM has been successfully applied to many practical engineering problems, it can encounter significant difficulties when simulating problems that involve high Peclet numbers [33,77,78], or when using grids that contain inappropriately stretched obtuse-angled triangles. These difficulties include numerical instabilities and/or physically unrealistic results caused by negative coefficients in the discretized equations.

In an effort to alleviate these problems, Prakash [79] proposed a donor-cell upwind scheme that equates the value of a scalar advected out of a control-volume across its boundaries to the value of the scalar within that control-volume. This procedure guarantees that the algebraic approximations to the advection terms in the integral conservation equations contribute positively to the coefficients in the discretized equations. However, this scheme takes little account of the direction of the flow and is therefore prone to considerable amounts of false diffusion.

Schneider and Raw [80] introduced a positive-coefficient, skewed, upwinding procedure, in the context of quadrilateral elements, which takes into account the local flow direction within an element, albeit in a rudimentary way compared with the FLO scheme. This procedure is nevertheless a more attractive approach to eliminating negative coefficients in the discretized equations compared with the donor-cell scheme of Prakash [79].

More recently, Saabas [33], Rousse [35], Masson *et al* [81], and Saabas and Baliga [77,78] have proposed mass-weighted (MAW) skew upwind schemes in the context of triangular elements, which are essentially adaptations of the positive-coefficient scheme of Schneider and Raw [80] for quadrilateral elements. A detailed derivation of the MAW scheme used in this thesis is given in Section 2.4.4.2.

The FLO scheme used in the present work is essentially the same as the one proposed by Baliga and Patankar [29], except that the FLO interpolation function is now used to interpolate the scalar variable in the advection terms only [60,81]. The scalar variable in the diffusion terms is interpolated linearly within each element. As pointed out by Patankar [82], it is not necessary to use the same interpolation function to approximate any particular dependent variable in *all* terms of the governing conservation equations. In fact, it is often advantageous not to do so. Details of the derivation are presented in Section 2.4.4.1. The strategy in the present CVFEM formulation is to use the FLO scheme whenever possible, and to use the MAW scheme only when the FLO scheme produces unacceptable large negative coefficients in the discretized equations.

It is well known that co-located, equal-order, primitive-variable formulations for incompressible fluid flow could admit physically unrealistic oscillations in the calculated pressure field unless special procedures are used [28,30,31,60,71,77,82-87]. For orthogonal, structured-grid formulations, one remedy is to use different grids for the velocity components and pressure. In this approach, the different grids are suitably displaced or staggered relative to one another [71,82,88]. Unfortunately, an equivalent grid-staggering approach for unstructured grids does not exist. Spurious pressure oscillations can be avoided on unstructured grids by using unequal-order formulations in which the pressure is stored at fewer nodes than the velocity components, and interpolated using lower-order interpolation functions [28,75,83,84].

Baliga [28] and Baliga and Patankar [83] proposed an unequal-order CVFEM for incompressible fluid flow. In their formulation, two different sets of control-volumes are used: one for the momentum equations, and the other for the continuity equation. Several disadvantages arose from this method: (i) mass conservation is not strictly satisfied over the momentum control-volumes; (ii) due to the two sets of control-volumes, the specification and calculation of geometric information is unwieldy; and (iii) accurate solutions of flows involving high Reynolds numbers or large pressure gradients would require excessively fine grids due to the lower-order interpolation functions for the pressure.

To avoid the above-stated problems associated with the unequal-order CVFEM of Baliga and Patankar [83], Prakash and Patankar [30] proposed a CVFEM based on a co-located, equal-order, velocity-pressure formulation. In this method, the velocity components and pressure are stored at the same nodes and interpolated over the same elements. Only one set of polygonal control-volumes is constructed around each node in the finite element mesh. Within each element, the pressure is interpolated linearly. When the velocity components are considered as transported scalars in the momentum equations, they are interpolated using the FLO scheme of Baliga and Patankar [29]. However, when the velocity components are considered as part of the mass-flux terms appearing in both the momentum and continuity equations, a special momentum interpolation scheme is used in order to avoid the possibility of spurious pressure oscillations. A detailed description of this interpolation scheme is given in Section 2.5.6. Similar interpolation schemes have been used by Rhee and Chow [89] and Peric *et al.* [90] in the context of FVMs, and by Rice and Schnipke [87] in the context of FEMs. Several enhancements of the CVFEM of Prakash and Patankar [30] have been presented in the works of Hookey [32], Hookey and Baliga [91], Prakash [85], Saabas and Baliga [77], and Masson *et al.* [81]. A detailed review of these works can be found in the treatise by Baliga [60]. The CVFEM used in this thesis is based on the formulations given by Saabas and Baliga [77], and Masson *et al.* [81]. Details of the current method are presented in Chapter 2.

Recently, Baliga [92] has discussed the conditions necessary for guaranteeing positive diffusion contributions to the discretized equations, and resulting implications regarding the momentum interpolation scheme of Prakash and Patankar [30]. Barth [57] has presented a formal proof in which he claims that Delaunay triangulation [93] in conjunction with linear interpolation of the scalar variable over each element is a sufficient condition for guaranteeing positive diffusion contributions to the coefficients in discretized equations. This proof, however, is only valid if the diffusion coefficient, Γ_s , is constant over the entire domain. If Γ_s is a variable quantity, but is made to be piecewise-constant over each element, then every element in the domain must be an acute-angled triangle in order to guarantee positive diffusion contributions. A more complete set of conditions is presented in Section 2.4.3.1 of this thesis, and the

corresponding proofs are provided in Appendix B. Implications of these conditions on the discretized equation for pressure, which is akin to a discretized Poisson equation with a variable diffusion coefficient, are discussed in Section 2.5.7.

1.3.2 Unstructured Grid Generation

Owing to the random way in which elements may be generated to fill a computational domain, the basic premise behind using unstructured grids is that they are inherently more automatic and amenable to discretizing complex geometries than structured or block-structured grids are. The problem of generating an unstructured grid is essentially one of designing automatic and robust algorithms that yield suitable element shapes and distributions for the desired type of numerical simulation.

The two most prevalent types of unstructured grid generation techniques are: the advancing-front method and Delaunay-based methods [56]. While the advancing-front method is somewhat heuristic in nature, Delaunay-based approaches are well-rooted in computational geometry principles. Both methods have been successfully incorporated into both two- and three-dimensional grid generation tools for arbitrary geometries. However, they have also encountered robustness difficulties and various problems related to grid quality, which have led to the development of hybrid methods that attempt to capitalize on the strengths of both methods.

1.3.2.1 Advancing-Front Method

In the advancing-front technique, an unstructured mesh is constructed by adding individual elements to an existing *front* of generated elements. The size and shape of a newly created element is usually determined from local field information on a background mesh. Advancing-front methods usually yield smooth, high quality triangulations in most regions of the domain, however, difficulties may arise where the fronts eventually merge. Heuristic edge-intersection checking operations are needed at the moving front, and this usually requires the use of quad/octree data structures for efficient implementations [56]. Detailed descriptions of some advancing-front methods are given in the works of Lo [94], Peraire *et al.* [95], and Gumbert *et al.* [96].

1.3.2.2 Delaunay-Based Methods

Given a set of points in a plane, a Delaunay construction represents a unique triangulation of these points, which possesses many well-defined properties [93]. The empty circumcircle property forms the basis of many Delaunay-based mesh generation schemes [97,98]. This property states that the circumcircle of a triangle in a Delaunay triangulation cannot contain any point besides the three forming vertices of the respective triangle. Thus, given an initial triangulation, a node may be inserted by first removing all existing triangles whose circumcircle contains the newly inserted point. Then, this new point is joined with each of the vertices on the boundary of the cavity created by the previously removed triangles. This type of point insertion algorithm may be used as the basis for a mesh generation strategy, and extends naturally to three dimensions by considering the circumsphere associated with a tetrahedron. The initial mesh point distribution can in principle be random, but most often it is generated using an automatic point placement strategy such as those proposed by Baker [99], Weatherill [100], Mavriplis [101], or Holmes and Snyder [102]. The main disadvantages of current Delaunay triangulation techniques relate to their inability to guarantee the integrity of the boundary. George *et al.* [103] and Weatherill *et al.* [104] have proposed edge and face swapping techniques for locally modifying Delaunay meshes to conform to the boundary discretizations. Several Delaunay triangulation techniques are evaluated in surveys by Fortune [105] and Su and Drysdale [106].

1.3.2.3 Other Triangulation Methods

Other triangulation techniques include Quad/octree-based methods, which are relatively inexpensive and produce good quality meshes in the interior of the domain. Details regarding this type of triangulation can be found in the work of Yerry and Shephard [107]. The main drawback of the technique is an irregular cell distribution near boundaries [56].

Examples of hybrid triangulation techniques combining the advantages of the advancing-front algorithm with the theoretical rigor of Delaunay triangulation, can be found in the works of Rebay [108] and Mueller *et al.* [109].

Highly *stretched* or *anisotropic* grids consisting of very large element aspect ratios are very useful for the efficient resolution of thin-boundary layers, wakes, and other viscous regions characteristic of high-Reynolds-number flows. Babuska and Aziz [110] have shown how the accuracy of a two-dimensional finite element approximation on a triangular element degrades as the maximum angle of the element increases. This implies that stretched obtuse-angle triangles containing one large angle and two small angles should be avoided, while stretched triangles with one small angle and two nearly right angles are preferred. Mavriplis [39,111], and Castro-Diaz *et al.* [112] were among the first to developed stretched grid generation techniques based on a Delaunay triangulation performed in a locally mapped space. Nakahashi [113], and Ward and Kallinderis [114] have proposed using structured or semi-structured meshes in regions where stretching is required, and fully unstructured isotropic meshes elsewhere. Other anisotropic grid generation techniques can be found in the works of Lohner [115], Pirzadeh [116,117] and Connell and Braaten [118].

Mixed-element meshes combine simplicial elements (triangles in two dimensions, and tetrahedra in three dimensions) with other types of elements. The most common alternate element types are quadrilaterals in two dimensions, and prisms, pyramids, and hexahedra in three dimensions. Some advantages of nonsimplicial meshes include: (i) their lower connectivity; and (ii) the possibility of employing regular arrays of hexahedra or prisms to discretize three-dimensional space, if desired (it is not possible to pack space with arrays of self-similar tetrahedra). This latter advantage may also enhance accuracy, owing to a cancellation of truncation errors that may not occur using nonsimilar tetrahedra. Examples of unstructured mixed-element grids can be found in the works of Blacker and Stephenson [119], Nakahashi [113], Ward and Kallinderis [114], Connell and Braaten [118], Merriam [120], and Tembulkar and Hanks [121].

1.3.3 Grid-Adaptation Strategies

In addition to their well-suited ability to discretize complex geometries, the second main advantage of unstructured grids is the natural setting they provide for the implementation of solution-adaptive meshing. Since there is no inherent structure assumed in the construction of the mesh, nodes may be added, deleted or displaced with only a local

topological and/or connectivity reconfiguration required for the affected region. The main goal of mesh adaptation is to obtain an optimal distribution of nodes in the computational domain that results in controlling and/or minimizing the truncation error in the numerical solution.

The class of problems to be solved often dictates the most suitable type of mesh-adapting strategy to be utilized. For most steady-state problems, the adaptation phase is a relatively small part of a lengthy solution process, thus relatively sophisticated strategies, such as complete mesh regeneration, can be employed. In this procedure, the refinement process is usually most important, while de-refinement has only a minor effect and can often be omitted in steady problems [56]. For most transient problems, however, mesh adaptation must be performed after every few time steps, thus efficiency is more important than optimality. Both mesh refinement and de-refinement are important for transient cases. Mesh movement may also be required for cases with moving boundaries.

The most common type of adaptive analysis is the *h*-version (or *h*-refinement) which controls the accuracy of a discrete solution by changing the size and quantity of elements in the computational domain via node insertion and/or removal. In the *p*-version, the mesh is kept fixed and the accuracy is controlled by hierarchically increasing or decreasing the order of the element-based polynomial approximations of the dependent variables. Examples of *p*-version formulations can be found in the works of Babuska *et al.* [122] and Zienkiewicz and Craig [123]. When the mesh is adapted via node movement, it is termed *r*-refinement. Examples of *r*-refinement strategies can be found in the works of Palmerio [124], Castro-Diaz *et al.* [112], Hooke [32], and Reuss and Stuble [125]. Combinations of these adaptive strategies are often used in practice. The *h-r*-version has been implemented by Baum *et al.* [126], and Castro-Diaz *et al.* [112]. Examples of the *h-p*-version can be found in the works of Zienkiewicz *et al.* [8], and Guo and Babuska [127].

1.3.4 Error Estimation

The success of any solution-adaptive scheme relies on its ability to detect and assess the solution error. The main problem is that an exact characterization of the true error would

require knowledge of the exact solution itself, which is obviously unavailable. Thus, the primary goal in *a posteriori* error estimation is to utilize the computed solution itself to somehow assess its accuracy.

1.3.4.1 Review of *A Posteriori* Error Estimates

This section gives a brief overview of some *a posteriori* error-estimation techniques used in the numerical simulation of fluid flow and heat transfer problems. A more thorough review is given in the monograph by Ainsworth and Oden [15]. Details can be found in the cited references.

Amongst the earliest *a posteriori* error estimates were those used in the solution of ordinary differential equations. A typical predictor-corrector algorithm [128] estimates the error by comparing two different solutions obtained from schemes with different orders of accuracy. This rough estimation of the error is then used to adjust the time step (or spatial increment) accordingly.

The first error estimates for finite element solutions of elliptic boundary-value problems were likely proposed by Babuska and Rheinboldt [9,10]. In their scheme, the error in an energy norm is calculated for each finite element. These values form the basis for adaptive meshing procedures designed to control and minimize the solution error.

A technique proposed by Ladeveze and Leguillon [129] involves utilizing element-by-element *complementary energy formulations* together with *equilibrated boundary data* to estimate the solution error. Similar ideas were put forward by Kelly [130].

The *element residual method* is another class of error estimation techniques. This method was developed by Demkowicz *et al.* [131] and simultaneously advanced by Bank and Weiser [132].

Extrapolation methods have been used effectively to obtain error estimates for both the *p*- and *h*-versions of the finite element method. For example, in the case of the *h*-version, generalized Richardson extrapolation techniques [51-53] use a series of successively finer grids, and knowledge of the order of accuracy of the numerical scheme, to extrapolate appropriately to a higher-order approximation of the exact solution, thereby yielding an estimate of the error. In the case of the *p*-version, a scheme put forward by Szabo [133]

involves using a sequence of hierarchical p -version approximations, and an extrapolation process, to obtain efficient *a posteriori* error estimations.

A relatively simple error-estimation technique, developed by Zienkiewicz and Zhu [1], has proven to be very effective and economical for many classes of elliptic problems. Their technique can be categorized as a *recovery-based method*. The method was originally proposed for linear elasticity problems, but has since been extended and applied to nonlinear problems including incompressible Navier-Stokes flows [18,22-24,134]. A more detailed review of the Zienkiewicz and Zhu error estimator is presented in the next section, as this was the method adopted for use in the present work.

Some of the most recent efforts in the subject of *a posteriori* error estimation have been in the development of new techniques to study the robustness, effectiveness, and performance of existing estimators. Representative examples of these efforts include the works of Babuska *et al.* [11,12,14], Zienkiewicz and Zhu [2,3,5,6], and Zhang and Zhu [27].

Other recent works in error estimation include those of Usmani [26], Reuss and Stubley [125], Zingg [135], Zingg *et al.* [136], and Wu and Currie [20,21].

1.3.4.2 The Zienkiewicz and Zhu Error Estimator and Recovery Process

The basic idea behind the Zienkiewicz and Zhu error estimator (also known as the Z^2 error estimator) is to obtain a higher-order projection of the solution gradients via a *recovery* or *smoothing* process. The solution gradients are then compared with the higher-order projection in order to assess the global and local error. The original Z^2 estimator [1] uses the global L_2 projection of Hinton and Campbell [19] to obtain nodal values of the recovered gradients. In this procedure, the recovered gradients are interpolated over each element using the same basis functions as the dependent variables themselves, and they are made to fit the original basis functions in a least-squares sense. This scheme is able to consistently achieve superconvergence of the nodal derivatives for linear three-node triangular elements (where superconvergence refers to the recovered gradients having a convergence rate of $O(h^{p+\alpha})$ with $\alpha \geq 1$, while the original solution gradients display the standard $O(h^p)$ convergence). However, it is unable to obtain the desired superconvergent

behavior for quadratic six-node triangular elements. In fact, in some applications, the local error of the recovered solution using quadratic elements actually increased [2]. In a subsequent paper, Zienkiewicz and Zhu proposed a *local* superconvergent patch recovery method [2] in an effort to overcome this deficiency. In this method, recovered gradients are interpolated over the *patch of elements* surrounding each node, using a single differentiable function. The order of this polynomial interpolation function is the same as the order of the element-based profiles used to interpolate the dependent variables. The recovered gradients are then made to fit the original basis functions of the FEM gradients in a *local* least squares sense. This new local patch-recovery scheme provides a dramatic improvement over the global L_2 projection of Hinton and Campbell [19] for the case of quadratic elements. However, for the case of three-node linear elements (as is used in the present CVFEM formulation for the diffusion terms and pressure), the two methods were shown to have similar convergence rates [2]. One disadvantage of the *local* patch-recovery scheme [2], as noted by Labb   and Garon [25], is the potential for rank deficiency in the system of equations obtained for patches of elements in highly irregular domains. Their proposed remedy requires special procedures at the boundary and no longer guarantees superconvergence in these regions. In the interest of robustness, two recovery schemes were implemented in the present formulation that are guaranteed to function irrespective of the complexity of the domain, and do not require any special procedures for extreme cases. The global L_2 projection of Hinton and Campbell [19] is such a scheme, and was implemented, along with a local averaging scheme that performs a volume-weighted average of the solution gradients over the patch of elements surrounding each node. As shown in Chapter 6, both of these schemes provide the desired superconvergent behavior at interior nodes, and are sufficiently convergent at the boundary nodes to successfully drive the adaptive process.

1.4 Thesis Outline

In this chapter, the objectives and motivation of the thesis were stated, and a review of the pertinent literature was provided. Reviews of other relevant topics are appropriately deferred to later chapters.

In Chapter 2, the formulation of a CVFEM for steady, two-dimensional, incompressible, viscous, Newtonian fluid flow and heat transfer is presented. First the governing equations are stated, and the domain discretization method is described. Then, appropriate element-based interpolation functions are specified for the dependent variables and thermofluid properties; algebraic approximations to the integral conservation equations are derived; and a solution procedure for solving the discrete algebraic equations is prescribed. In Section 2.4.3.1, sufficient conditions for guaranteeing positive diffusion contributions to the coefficients in the discretized equations are provided. These conditions supplement those stated by Barth [57]. The corresponding proofs are presented in Appendix B.

In Chapter 3, details of an additive-correction multigrid (ACM) method, used to accelerate iterative convergence in the CVFEM solution procedure, are presented. Following a brief introduction, the grid-coarsening algorithm used to create the hierarchy of grid levels in the ACM cycle is described. Then, the restriction and prolongation operators are presented in detail.

In Chapter 4, an h -adaptive strategy is presented. This procedure is used to optimize the CVFEM with respect to accuracy and computational effort, by adapting the computational mesh during the solution process to meet a specified mesh optimality criterion. First, the recovery schemes that were used to obtain higher-order projections of the CVFEM gradients are presented. Then, two error estimators and a flow-angle indicator are outlined. Their purpose is to use the higher-order projections and the original CVFEM gradients to estimate the global and local solution error. Finally, the refinement step is discussed. In this step, the estimated error is used as a guide to refine the current mesh appropriately.

In Chapter 5, an extended Richardson extrapolation technique for unstructured grids is proposed. This technique uses a series of discrete solutions, obtained on a hierarchy of unstructured grids, to approximate the solution to the analogous continuum problem, by performing an extrapolation to the limit of infinite grid density. First, the original Richardson extrapolation [51,52] technique for uniform, ordered grids, is reviewed. Then,

generalizations of the original technique [53,143,145] are outlined. Finally, the proposed extrapolation technique for unstructured grids is presented.

In Chapter 6, the numerical results of this thesis are presented. The aforementioned algorithms and techniques, implemented in the present work, are verified on several test problems. One of the test problems is an original exact solution to the steady, two-dimensional, incompressible Navier-Stokes equations with body forces. This closed-form, analytical solution is obtained by prescribing expressions for the pressure and divergence-free velocity fields, substituting them into the momentum equations, and deriving the implied volumetric source terms. Finally, lid-driven, viscous flow in a triangular cavity is simulated using the h -adaptive CVFEM. Results are compared with a published solution.

In Chapter 7, a brief review of the thesis and its contributions is presented, and some recommendations for future work are outlined.

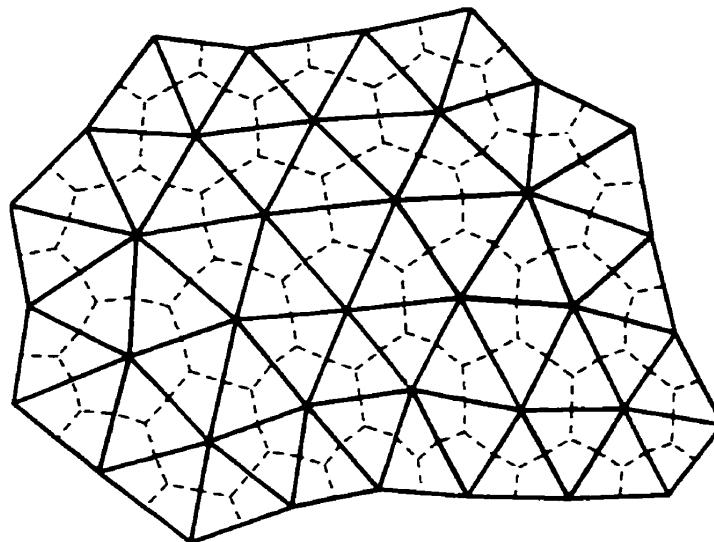


Figure 1.1: Discretization of a two-dimensional, planar calculation domain into three-node triangular elements and polygonal control volumes

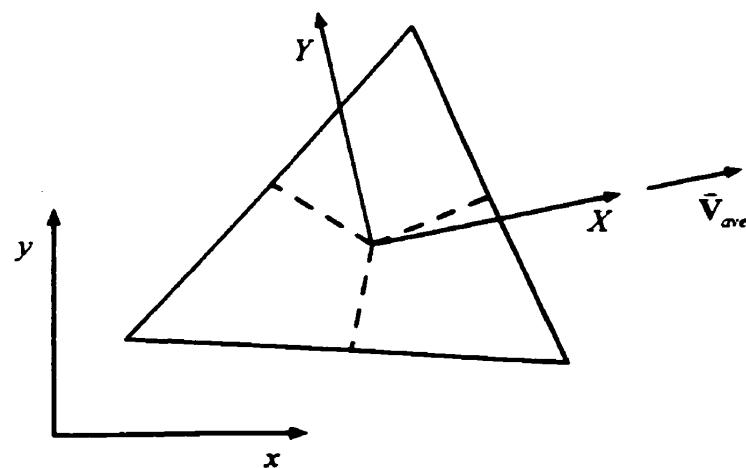


Figure 1.2: Local rotated coordinate system (X, Y) used in the FLO scheme for a typical element. The X -axis is aligned with the element-average velocity, \bar{V}_{ave} .

Chapter 2

CONTROL-VOLUME FINITE ELEMENT METHOD FOR TWO-DIMENSIONAL, STEADY-STATE, INCOMPRESSIBLE FLUID FLOW AND HEAT TRANSFER

A control-volume finite element method (CVFEM) is presented in this chapter for the simulation of steady, incompressible, Newtonian fluid flow and heat transfer in arbitrarily shaped two-dimensional (planar) domains. The co-located, equal-order formulation deals directly with the primitive dependent variables, such as the velocity components, pressure and temperature. This implies that all the dependent variables, including pressure, are stored at the same nodes in the finite element mesh, and interpolated over the same elements. The present method is based on concepts originally proposed by Baliga [28], Baliga and Patankar [29], and Prakash and Patankar [30], and later enhanced and/or extended by Raw [31], Hookey [32], Saabas [33], Masson [34], and others [35-38]. A brief review of CVFEMs is provided in Section 1.3.1 of this thesis. A more thorough review can be found in the monograph by Baliga [60].

The formulation of a CVFEM for fluid flow and heat transfer typically involves five basic steps [60]: (i) discretization of the calculation domain into elements; (ii) construction of control-volume surfaces around each node in the finite element mesh; (iii) prescription of element-based interpolation functions for the dependent variables and thermophysical properties; (iv) use of the control-volume-based method of weighted residuals and an element-by-element procedure to derive and assemble algebraic approximations to the governing equations; and (v) prescription of a procedure to solve

the discrete algebraic equations. The remainder of this chapter will elaborate on each of these steps.

2.1 Governing Equations

Steady, viscous flow of an incompressible, Newtonian fluid can be described, with respect to the Cartesian coordinate system (x,y) , by the following partial differential equations:

Continuity Equation:

$$\bar{\nabla} \cdot \rho \bar{\mathbf{V}} = 0. \quad (2.1)$$

x -Momentum Equation:

$$\bar{\nabla} \cdot \bar{\mathbf{J}}_u = -\frac{\partial p}{\partial x} + S_u, \quad (2.2)$$

$$\text{where, } \bar{\mathbf{J}}_u = \rho \bar{\mathbf{V}} u - \mu \bar{\nabla} u.$$

y -Momentum Equation:

$$\bar{\nabla} \cdot \bar{\mathbf{J}}_v = -\frac{\partial p}{\partial y} + S_v, \quad (2.3)$$

$$\text{where, } \bar{\mathbf{J}}_v = \rho \bar{\mathbf{V}} v - \mu \bar{\nabla} v.$$

Other Advection-Diffusion Equations (General Form):

$$\bar{\nabla} \cdot \bar{\mathbf{J}}_\phi = S_\phi, \quad (2.4)$$

$$\text{where, } \bar{\mathbf{J}}_\phi = \rho \bar{\mathbf{V}} \phi - \Gamma_\phi \bar{\nabla} \phi.$$

In these equations, ρ is the mass density; μ is the dynamic viscosity; p is the pressure; u and v are the x and y components of the velocity vector, $\bar{\mathbf{V}}$, respectively; S_u and S_v are body forces per unit volume acting on the fluid in the x and y coordinate directions, respectively; and $\bar{\mathbf{J}}_u$ and $\bar{\mathbf{J}}_v$ are the total (advection and diffusion) fluxes of the x and y components of momentum, respectively.

A large variety of other transport processes, such as heat or mass transfer, may be influenced by, or coupled to the flow field. Each of these processes can be described by the general advection-diffusion equation, given by Eq. (2.4) [82], where ϕ represents the corresponding specific scalar dependent variable; Γ_ϕ is the related diffusion coefficient; and S_ϕ is the appropriate volumetric generation rate or source term. The continuity and momentum equations, Eqs. (2.1) - (2.3), can also be cast in the general form of Eq. (2.4), by suitably defining ϕ , Γ_ϕ , and S_ϕ according to Table 2.1.

| Equation | ϕ | Γ_ϕ | S_ϕ |
|------------|--------|---------------|--|
| Continuity | 1 | 0 | 0 |
| x-Momentum | u | μ | $-\frac{\partial p}{\partial x} + S_u$ |
| y-Momentum | v | μ | $-\frac{\partial p}{\partial y} + S_v$ |

Table 2.1: Specific forms of the general advection-diffusion equation.

If the fluid is non-Newtonian, other constitutive relations can be implemented by including any additional shear-stress terms in S_u and S_v . Similarly, diffusion processes which cannot be described by a linear gradient law (such as Fourier's law or Fick's law) can be implemented by including any additional diffusion terms in the source term, S_ϕ .

Equation (2.4) is defined on a closed, bounded region, $\Omega \subset \mathbb{R}^2$, and is subject to the following generalized boundary conditions:

$$\psi \left[\phi, \frac{\partial \phi}{\partial n} \right] = 0 \quad \text{on} \quad \partial \Omega, \quad (2.5)$$

where $(\partial\phi/\partial n)|_{\partial\Omega}$ is the directional derivative of ϕ at the boundary, $\partial\Omega$, in the direction pointing outward and normal to $\partial\Omega$. In general, ψ is a nonlinear, algebraic operator, which can take on different functional forms over different portions of the boundary. For the fluid flow problems considered in this thesis, the momentum equations, Eqs. (2.2) and (2.3), are also subject to the boundary conditions given by Eq. (2.5), with ϕ replaced by u and v , respectively. Furthermore, the correct pressure field, p , is determined to within an arbitrary constant, when the resulting velocity field, obtained from the momentum equations, satisfies the continuity constraint given by Eq. (2.1). Details of the boundary treatment, in the CVFEM formulation, for specific forms of Eq. (2.5), will be presented in Section 2.5.8.

2.2 Domain Discretization

In the present CVFEM, the calculation domain is discretized using unstructured grids comprised of three-node triangular elements. One advantage offered by the use of such grids is the facility with which irregular, multiply-connected domains of arbitrary shape can be discretized without the need for multiblock composite grid schemes, or other special procedures, that may otherwise be required if structured grids are used. Another major advantage in the present unstructured formulation is that it allows for the implementation of unstructured, adaptive-grid schemes based on *a posteriori* error estimation, such as those developed by Zienkiewicz and Zhu [1-4], and others [5-27]. These schemes provide for a mechanism by which a grid can be refined or coarsened, both globally and locally, in an iterative fashion, with the ultimate aim of achieving an “optimal mesh” (defined by a prescribed *mesh optimality criteria*) for the particular problem being simulated. Details of the present *h*-adaptive scheme are given in Chapter 4.

All the grids in the present work are conforming Delaunay triangulations with specified minimum angle constraints. Most of these grids were created and refined using *Triangle*, an unstructured Delaunay triangulator developed by J. R. Shewchuk [40]. A brief description of *Triangle* and its capabilities are given in Appendix A.

Once the domain has been triangulated, each element is subdivided by joining its centroid to the midpoints of the sides of the element. Collectively, the resulting faces within each element form polygonal control-volumes surrounding each node in the finite element mesh, as shown in Fig. 1.1. These control-volumes do not overlap, and collectively they fill the entire calculation domain completely and exactly. This property facilitates the formulation of a discretization method that possesses the conservative property [82]. If the physical domain (two-dimensional, planar) has a curved boundary, the curve is approximated by piecewise-straight lines, corresponding to element edges, and the resulting polygonal region is taken to be the calculation domain.

2.3 Integral Conservation Equation

Integral conservation equations, corresponding to Eqs. (2.1) - (2.4), can be obtained for each node in the calculation domain, by applying the appropriate conservation principles to the control-volumes associated with each of the nodes. With respect to a typical node, i , which may be an internal node as shown in Fig. 2.1 (a), or a boundary node as shown in Fig. 2.1 (b), Eq. (2.4) can be integrated over the control volume surrounding node i , resulting in the following integral conservation equation:

$$\left[\int_a^o \bar{\mathbf{J}} \cdot \bar{\mathbf{n}} ds + \int_o^c \bar{\mathbf{J}} \cdot \bar{\mathbf{n}} ds - \int_{iac} S_\phi d\Omega \right] + [\text{similar contributions from other elements associated with node } i] + [\text{boundary contributions, if applicable}] = 0, \quad (2.6)$$

where, $\bar{\mathbf{J}} = \bar{\mathbf{J}}_A + \bar{\mathbf{J}}_D$,

$$\bar{\mathbf{J}}_A = \rho \bar{\mathbf{V}} \phi,$$

$$\bar{\mathbf{J}}_D = -\Gamma_\phi \bar{\nabla} \phi.$$

In this equation, $\bar{\mathbf{J}}_A$ and $\bar{\mathbf{J}}_D$ are the advection and diffusion fluxes, respectively, associated with the dependent variable ϕ , and $\bar{\mathbf{n}}$ is a unit vector directed outward and normal to the control-volume boundary. The Gauss divergence theorem is used in transforming the first two terms of Eq. (2.6) from volume integrals to surface integrals. The form of Eq. (2.6) emphasizes that it may be conveniently assembled using an element-by-element procedure.

2.4 Interpolation Functions

The derivation of algebraic approximations to the integral conservation equation, given by Eq. (2.6), requires the prescription of element-based interpolation functions for the dependent variables, thermofluid properties, and source terms.

2.4.1 Diffusion Coefficient, Density, and Source Term

In each element, the density, ρ , and diffusion coefficient, Γ_ϕ , are evaluated at the centroid, and assumed to prevail over that element.

The source term, S_ϕ , is linearized, if required, and expressed in the following general form:

$$S_\phi = S_c + S_p \phi. \quad (2.7)$$

Within each element, the values of S_c and S_p are evaluated at the *nodal* locations, and each of these three pairs of nodal values is assumed to prevail over the portion of the corresponding control-volume in that element. Element-based storage of S_c and S_p (three nodal pairs per element) is recommended to provide for cases where S_ϕ may be a discontinuous function of x and y . For such instances, S_ϕ is approximated as a piecewise-continuous function within each element, and element edges are placed along the surfaces of discontinuity. If the problems of interest involve continuous source terms only, then a node-based data structure is sufficient for the storage of S_c and S_p . In these cases, the interpolation of S_c and S_p , described above, is equivalent to having nodal values prevail over their respective control-volumes.

2.4.2 Mass Flow Rates

In the calculation of the mass flow rates across control-volume boundaries, the velocity is interpolated using special forms of the discretized momentum equations. Details of the interpolation scheme (often referred to as a *momentum interpolation scheme*) are presented in Section 2.5.6. This special momentum interpolation procedure, borrowed from the work of Prakash and Patankar [30], prevents the occurrence of spurious pressure oscillations that could otherwise afflict co-located, equal-order, primitive variable formulations for incompressible flow [30,60,82]. The resulting “mass conserving” velocity used to evaluate the mass flow rates is denoted by,

$$\bar{\mathbf{V}}^m = u^m \bar{\mathbf{i}} + v^m \bar{\mathbf{j}}. \quad (2.8)$$

where $\bar{\mathbf{i}}$ and $\bar{\mathbf{j}}$ are unit vectors in the x and y coordinate directions, respectively.

2.4.3 Diffusion Term

For the purposes of evaluating the diffusion term in Eq. (2.6), the dependent variable, ϕ , is interpolated linearly within each element:

$$\phi^D = A_\phi^D x + B_\phi^D y + C_\phi^D, \quad (2.9)$$

where the superscript, “ D ”, emphasizes that this function is used to derive the algebraic approximations to the *diffusion* flux. For each element, the coefficients, A_ϕ^D , B_ϕ^D , and C_ϕ^D , can be uniquely determined in terms of the three nodal values of x , y , and ϕ , by evaluating Eq. (2.9) at the element vertices. With reference to element 123 shown in Fig. 2.2, the resulting system of equations can be written as follows:

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix} \begin{Bmatrix} A_\phi^D \\ B_\phi^D \\ C_\phi^D \end{Bmatrix} = \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{Bmatrix}, \quad (2.10)$$

with solution,

$$\begin{Bmatrix} A_\phi^D \\ B_\phi^D \\ C_\phi^D \end{Bmatrix} = \frac{1}{2\Omega_e} \begin{bmatrix} y_2 - y_3 & y_3 - y_1 & y_1 - y_2 \\ x_3 - x_2 & x_1 - x_3 & x_2 - x_1 \\ x_2y_3 - x_3y_2 & x_3y_1 - x_1y_3 & x_1y_2 - x_2y_1 \end{bmatrix} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{Bmatrix}, \quad (2.11)$$

in which,

$$2\Omega_e = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} = (x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1), \quad (2.12)$$

where Ω_e is the volume (area times unit depth) of element 123. If these expressions for A_ϕ^D , B_ϕ^D , and C_ϕ^D are substituted back into Eq. (2.9), ϕ^D can be rewritten in terms of the linear shape functions, N_1^D , N_2^D , and N_3^D , as follows:

$$\phi^D = N_1^D \phi_1 + N_2^D \phi_2 + N_3^D \phi_3, \quad (2.13)$$

where,

$$\begin{Bmatrix} N_1^D \\ N_2^D \\ N_3^D \end{Bmatrix} = \begin{Bmatrix} L_1 \\ L_2 \\ L_3 \end{Bmatrix} = \frac{1}{2\Omega_e} \begin{bmatrix} y_2 - y_3 & x_3 - x_2 & x_2y_3 - x_3y_2 \\ y_3 - y_1 & x_1 - x_3 & x_3y_1 - x_1y_3 \\ y_1 - y_2 & x_2 - x_1 & x_1y_2 - x_2y_1 \end{bmatrix} \begin{Bmatrix} x \\ y \\ 1 \end{Bmatrix}. \quad (2.14)$$

In this expression, L_1 , L_2 , and L_3 , are the *area coordinates* [137] of triangle 123. Area coordinates are traditionally employed in the formulation of FEMs, and can also be used to formulate the present CVFEM. However, the current method will, arbitrarily, be presented here, using terms and nomenclature commonly employed in the formulation of FVMs.

2.4.3.1 Sufficient Conditions for Positive Diffusion Contribution to the Coefficients in the Discretized Equations

It is often assumed, in the published literature, that Delaunay triangulation, in conjunction with linear interpolation of the dependent variable, ϕ , is a sufficient condition for guaranteeing that the algebraic approximation to the diffusion transport term in Eq. (2.6) contributes positively to the coefficients in the discretized equations. Barth [57] has presented a formal proof of this claim, however, this proof is only valid if the diffusion coefficient, Γ_ϕ , is *constant* over the entire domain. Furthermore, the proof is not valid for elements that have at least one edge on the boundary of the domain.

The following is a summary of the sufficient conditions required to ensure positive diffusion contributions to the coefficients in the discretized equations. These statements are proven in Appendix B.

Case I: Constant Γ_ϕ

- Conditions:
- The mesh must be a Delaunay triangulation.
 - Every element with at least one of its edges on the boundary must be an acute-angle triangle (that is, each vertex angle $\leq \pi/2$ radians).
- Note: this is a sufficient condition that is overly restrictive. A less restrictive condition is given by Eq. B.37 in Appendix B, however, this latter condition may be impractical to achieve, in general.*

Case II: Variable Γ_ϕ - Piecewise constant over each element

- Conditions:
- Every element in the domain must be an acute-angle triangle.

Case III: Variable Γ_ϕ - Piecewise constant over each control-volume

- Conditions:
- (Same as in Case I)

Warning: This treatment for Γ_ϕ will result in a *non-conservative* CVFEM formulation, and is, therefore, *not recommended*.

It is noted here that in the case of variable Γ_ϕ with piecewise-linear distributions over each element, no conditions could be found on the size or shape of the elements that would guarantee positive diffusion contributions to the coefficients in the discretized equations. Instances in which Γ_ϕ cannot be represented by a scalar quantity (for example, in the case of non-isotropic or orthotropic materials) were not considered here.

2.4.4 Advection Term

The use of linear interpolation functions for ϕ in the diffusion transport terms is consistent with the elliptic nature of the diffusive process. However, for problems that involve moderate to strong advection relative to diffusion transport, linear interpolation of ϕ in the advection term could lead to physically unrealistic oscillatory solutions and cause iterative solution procedures to diverge [60,82]. In an attempt to overcome this difficulty, two different interpolation schemes, which take into account the direction of advection transport, were implemented for the purposes of obtaining algebraic approximations to the advection terms in Eq. (2.6): (1) a flow-oriented upwind scheme (FLO); and (2) a mass-weighted upwind scheme (MAW).

2.4.4.1 Flow-Oriented Upwind Scheme (FLO)

The FLO scheme was originally proposed by Baliga and Patankar [29]. It is based on a particular solution to a simplified advection-diffusion equation over the element of interest [28]. The FLO interpolation function automatically responds to an element-based Peclet number, which is an indicator of the relative strengths of advection and diffusion within the element. The function also takes into account the direction of the element-average velocity vector.

With reference to Fig. 2.2 (a), the interpolation function for ϕ can be written with respect to a local flow-oriented coordinate system (X, Y) as follows:

$$\phi^A = A_\phi^A \xi + B_\phi^A Y + C_\phi^A, \quad (2.15)$$

where,

$$\xi = \frac{\Gamma_\phi}{\rho U_{ave}^m} \left\{ \exp \left[-Pe \frac{(X_{max} - X)}{(X_{max} - X_{min})} \right] - 1 \right\},$$

$$\text{Pe} = \frac{\rho U_{\text{ave}}^m (X_{\max} - X_{\min})}{\Gamma_\phi},$$

$$X_{\max} = \max(X_1, X_2, X_3), \quad X_{\min} = \min(X_1, X_2, X_3),$$

$$U_{\text{ave}}^m = |\bar{\mathbf{V}}_{\text{ave}}^m|.$$

The element-average velocity, $\bar{\mathbf{V}}_{\text{ave}}^m$, is given by,

$$\bar{\mathbf{V}}_{\text{ave}}^m = u_{\text{ave}}^m \bar{\mathbf{i}} + v_{\text{ave}}^m \bar{\mathbf{j}}, \quad (2.16)$$

$$\text{where, } u_{\text{ave}}^m = \frac{u_1^m + u_2^m + u_3^m}{3}, \quad v_{\text{ave}}^m = \frac{v_1^m + v_2^m + v_3^m}{3}.$$

In this expression, $\bar{\mathbf{i}}$ and $\bar{\mathbf{j}}$ are the global Cartesian unit vectors, and the nodal values of u^m and v^m are obtained from Eq. (2.55) in Section 2.5.6. The superscript “A” in Eq. (2.15) indicates that this function is used to interpolate ϕ in the *advection* term. The coefficients A_ϕ^A , B_ϕ^A , and C_ϕ^A , in Eq. (2.15), are determined in the same manner as A_ϕ^D , B_ϕ^D , and C_ϕ^D were in Section 2.4.3. They are given by,

$$\begin{Bmatrix} A_\phi^A \\ B_\phi^A \\ C_\phi^A \end{Bmatrix} = \frac{1}{2\Omega_\xi} \begin{bmatrix} Y_2 - Y_3 & Y_3 - Y_1 & Y_1 - Y_2 \\ \xi_3 - \xi_2 & \xi_1 - \xi_3 & \xi_2 - \xi_1 \\ \xi_2 Y_3 - \xi_3 Y_2 & \xi_3 Y_1 - \xi_1 Y_3 & \xi_1 Y_2 - \xi_2 Y_1 \end{bmatrix} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{Bmatrix}, \quad (2.17)$$

$$\text{where, } 2\Omega_\xi = \begin{vmatrix} \xi_1 & Y_1 & 1 \\ \xi_2 & Y_2 & 1 \\ \xi_3 & Y_3 & 1 \end{vmatrix} = (\xi_2 - \xi_1)(Y_3 - Y_1) - (\xi_3 - \xi_1)(Y_2 - Y_1).$$

The local flow-oriented coordinate system, (X, Y) , is rotated with respect to the global (x, y) system, so that the X -axis is parallel to the element-average velocity vector, $\bar{\mathbf{V}}_{\text{ave}}^m$. The corresponding coordinate transformation is given by,

$$\begin{Bmatrix} X \\ Y \end{Bmatrix} = \frac{1}{U_{\text{ave}}^m} \begin{bmatrix} u_{\text{ave}}^m & v_{\text{ave}}^m \\ -v_{\text{ave}}^m & u_{\text{ave}}^m \end{bmatrix} \begin{Bmatrix} x - x_o \\ y - y_o \end{Bmatrix}, \quad (2.18)$$

where (x_o, y_o) are the global coordinates of the centroid of element 123.

In general, the FLO interpolation function, given by Eq. (2.15), is not C^0 continuous, in that the continuity of ϕ^A is not guaranteed along element boundaries except at the nodes. Along control-volume surfaces, however, ϕ^A and its derivatives are indeed continuous. In the limiting case, when advection transport within the element goes to zero ($Pe \rightarrow 0$), Eq. (2.15) reduces to a linear function and, thereby, becomes C^0 continuous. In the case of the opposite extreme, when advection transport within the element dominates relative to diffusion ($Pe \rightarrow \infty$), Eq. (2.15) reduces to a standard, one-dimensional upwind scheme along lines parallel to the X -axis, and remains linear in Y .

The FLO scheme has performed quite well in problems involving acute-angle triangular elements and relatively modest Reynolds or Peclet numbers [29, 78, 81]. However, the FLO interpolation function can lead to negative contributions to the coefficients in the discretized equations when large flow gradients occur within an element, or when obtuse-angle triangles are used in problems with high Reynolds or Peclet numbers. If negative coefficients are obtained in the discretized equation for a particular control-volume after all the corresponding elemental contributions have been assembled, this implies that an increase in ϕ at a node outside the control-volume, with other conditions remaining unchanged, would result in a decrease in ϕ within the control-volume. This is physically unrealistic in the context of steady, advection-diffusion problems without sources. Furthermore, negative coefficients in the discretized equations could be detrimental to iterative solution procedures causing them to diverge in some cases, or produce physically spurious numerical oscillations in the scalar fields [29, 60, 78, 81]. In an effort to overcome these difficulties, the MAW scheme was implemented as an alternative interpolation scheme for ϕ in the advection term of Eq. (2.6).

2.4.4.2 Mass-Weighted Upwind Scheme (MAW)

The MAW scheme [60] is essentially an adaptation of the positive-coefficient scheme of Schneider and Raw [80]. It ensures, at the element level (and therefore at the control-volume level), that the advection contribution to the coefficients in the discretized equations is positive. With reference to Fig. 2.2 (b), the MAW scheme defines a mass-weighted average of ϕ at the three integration points, r , s , and t , which lie on the control-

volume faces within element 123. The corresponding values, ϕ_r^A , ϕ_s^A , and ϕ_t^A , are expressed implicitly in the following manner:

$$\phi_r^A = \begin{cases} f_p \phi_t^A + (1 - f_p) \phi_1 & \text{where } f_p = \min[\max(\dot{m}_t / \dot{m}_r, 0), 1] \quad \text{if } \dot{m}_r > 0, \\ f_p \phi_s^A + (1 - f_p) \phi_2 & \text{where } f_p = \min[\max(\dot{m}_s / \dot{m}_r, 0), 1] \quad \text{if } \dot{m}_r < 0, \end{cases} \quad (2.19)$$

$$\phi_s^A = \begin{cases} f_p \phi_r^A + (1 - f_p) \phi_2 & \text{where } f_p = \min[\max(\dot{m}_r / \dot{m}_s, 0), 1] \quad \text{if } \dot{m}_s > 0, \\ f_p \phi_t^A + (1 - f_p) \phi_3 & \text{where } f_p = \min[\max(\dot{m}_t / \dot{m}_s, 0), 1] \quad \text{if } \dot{m}_s < 0, \end{cases} \quad (2.20)$$

$$\phi_t^A = \begin{cases} f_p \phi_s^A + (1 - f_p) \phi_3 & \text{where } f_p = \min[\max(\dot{m}_s / \dot{m}_t, 0), 1] \quad \text{if } \dot{m}_t > 0, \\ f_p \phi_r^A + (1 - f_p) \phi_1 & \text{where } f_p = \min[\max(\dot{m}_r / \dot{m}_t, 0), 1] \quad \text{if } \dot{m}_t < 0. \end{cases} \quad (2.21)$$

where the mass flow rates, \dot{m}_r , \dot{m}_s , and \dot{m}_t , are given by,

$$\dot{m}_r = \int_a^o \rho \bar{\mathbf{V}}^m \cdot \bar{\mathbf{n}}_r ds, \quad \dot{m}_s = \int_b^o \rho \bar{\mathbf{V}}^m \cdot \bar{\mathbf{n}}_s ds, \quad \dot{m}_t = \int_c^o \rho \bar{\mathbf{V}}^m \cdot \bar{\mathbf{n}}_t ds. \quad (2.22)$$

The superscript “A” in Eqs. (2.19)-(2.21) emphasizes that these values can be used to approximate ϕ in the *advection* term of Eq. (2.6). The unit normal vectors, $\bar{\mathbf{n}}_r$, $\bar{\mathbf{n}}_s$, and $\bar{\mathbf{n}}_t$, are as shown in Fig. 2.2 (b), and $\bar{\mathbf{V}}^m$ is the “mass conserving” velocity given by Eq. (2.55). The integrals in Eq. (2.22) can be evaluated by assuming $\bar{\mathbf{V}}^m$ to have a linear distribution over each element. The resulting expressions for the mass flow rates can be written as follows:

$$\begin{Bmatrix} \dot{m}_r \\ \dot{m}_s \\ \dot{m}_t \end{Bmatrix} = \frac{\rho}{2} \begin{Bmatrix} (u_o^m + u_a^m)(y_o - y_a) - (v_o^m + v_a^m)(x_o - x_a) \\ (u_o^m + u_b^m)(y_o - y_b) - (v_o^m + v_b^m)(x_o - x_b) \\ (u_o^m + u_c^m)(y_o - y_c) - (v_o^m + v_c^m)(x_o - x_c) \end{Bmatrix}. \quad (2.23)$$

To obtain explicit expressions for ϕ_r^A , ϕ_s^A , and ϕ_t^A , in terms of ϕ_1 , ϕ_2 , and ϕ_3 , the nominal values for the mass flow rates, given by Eq. (2.23), are first used to determine the appropriate implicit relations for ϕ_r^A , ϕ_s^A , and ϕ_t^A from Eqs. (2.19)-(2.21). These three relations can then be written in the following way:

$$\begin{bmatrix} 1 & a_{12} & a_{13} \\ a_{21} & 1 & a_{23} \\ a_{31} & a_{32} & 1 \end{bmatrix} \begin{Bmatrix} \phi_r^A \\ \phi_s^A \\ \phi_t^A \end{Bmatrix} = \begin{bmatrix} b_{11} & b_{12} & 0 \\ 0 & b_{22} & b_{23} \\ b_{31} & 0 & b_{33} \end{bmatrix} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{Bmatrix}, \quad (2.24)$$

from which the solution is readily obtained as,

$$\begin{Bmatrix} \phi_r^A \\ \phi_s^A \\ \phi_t^A \end{Bmatrix} = \begin{bmatrix} 1 & a_{12} & a_{13} \\ a_{21} & 1 & a_{23} \\ a_{31} & a_{32} & 1 \end{bmatrix}^{-1} \begin{bmatrix} b_{11} & b_{12} & 0 \\ 0 & b_{22} & b_{23} \\ b_{31} & 0 & b_{33} \end{bmatrix} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{Bmatrix}. \quad (2.25)$$

Eqs. (2.19)-(2.21) state that a node exterior to a control-volume can only contribute as much, or less, to the advective transport of ϕ out of the control-volume than it can to the advective transport of ϕ into the control-volume. Although this constraint guarantees positive elemental contributions to the coefficients in the discretized equations, it takes little account of the local flow direction, and gives a rather crude distribution of ϕ over the element. This may result in a significant amount of false diffusion in the numerical simulation. It is, therefore, recommended that the FLO scheme be used whenever possible, and that the MAW scheme be used only when the FLO scheme produces unacceptably large negative coefficients in the discretized equations.

2.4.5 Pressure

The pressure, p , is interpolated linearly within each element:

$$p = A_p x + B_p y + C_p. \quad (2.26)$$

The coefficients A_p , B_p , and C_p , are obtained in the same manner as A_ϕ^D , B_ϕ^D , and C_ϕ^D were in Section 2.4.3. They are given by,

$$\begin{Bmatrix} A_p \\ B_p \\ C_p \end{Bmatrix} = \frac{1}{2\Omega_e} \begin{bmatrix} y_2 - y_3 & y_3 - y_1 & y_1 - y_2 \\ x_3 - x_2 & x_1 - x_3 & x_2 - x_1 \\ x_2 y_3 - x_3 y_2 & x_3 y_1 - x_1 y_3 & x_1 y_2 - x_2 y_1 \end{bmatrix} \begin{Bmatrix} p_1 \\ p_2 \\ p_3 \end{Bmatrix}, \quad (2.27)$$

where Ω_e is given by Eq. (2.12). The pressure gradients can therefore be expressed as,

$$\left(\frac{\partial p}{\partial x} \right)_e = A_p, \quad \text{and} \quad \left(\frac{\partial p}{\partial y} \right)_e = B_p, \quad (2.28)$$

where the subscript “*e*” signifies that these are *element* pressure gradients.

2.5 Derivation of the Discretized Equations

A discretized, algebraic equation can be derived for each control-volume in the computational domain by substituting the appropriate interpolation functions, defined in Section 2.4, into the integral conservation equation, given by Eq. (2.6). Assembly of the coefficients in the discretized equations is achieved using an element-by-element procedure, as emphasized by the form of Eq. (2.6). In this section, elemental *diffusion*, *advection*, and *source* contributions are derived for the subvolume, *1aoc*, shown in Fig. 2.2. The final discrete equation for node 1 is obtained after similar contributions are assembled from the other elements associated with node 1, and after boundary contributions are added, if applicable.

2.5.1 Diffusion Contribution

The diffusion flux, $\bar{\mathbf{J}}_D$, can be expressed in the following form:

$$\bar{\mathbf{J}}_D = -\Gamma_s \bar{\nabla} \phi = -\Gamma_s \left[\frac{\partial \phi}{\partial x} \bar{\mathbf{i}} + \frac{\partial \phi}{\partial y} \bar{\mathbf{j}} \right]. \quad (2.29)$$

Using the linear interpolation function, ϕ^D , given by Eq. (2.9), to approximate ϕ , $\bar{\mathbf{J}}_D$ can be rewritten as,

$$\bar{\mathbf{J}}_D = -\Gamma_s [A_s^D \bar{\mathbf{i}} + B_s^D \bar{\mathbf{j}}], \quad (2.30)$$

where A_s^D and B_s^D are given by Eq. (2.11). The diffusion contribution, I_{1aoc}^D , for subvolume, *1aoc* of element 123 can be stated as,

$$I_{1aoc}^D = \int_a^o \bar{\mathbf{J}}_D \cdot \bar{\mathbf{n}} ds + \int_o^c \bar{\mathbf{J}}_D \cdot \bar{\mathbf{n}} ds, \quad (2.31)$$

where,

$$\bar{\mathbf{n}} ds = dy \bar{\mathbf{i}} - dx \bar{\mathbf{j}},$$

and,

$$\bar{\mathbf{J}}_D \cdot \bar{\mathbf{n}} \, ds = -\Gamma_\phi [A_\phi^D dy - B_\phi^D dx].$$

Therefore,

$$\int_a^o \bar{\mathbf{J}}_D \cdot \bar{\mathbf{n}} \, ds = -\Gamma_\phi [A_\phi^D (y_o - y_a) - B_\phi^D (x_o - x_a)]. \quad (2.32)$$

Similarly,

$$\int_o^c \bar{\mathbf{J}}_D \cdot \bar{\mathbf{n}} \, ds = -\Gamma_\phi [A_\phi^D (y_c - y_o) - B_\phi^D (x_c - x_o)]. \quad (2.33)$$

Thus, the diffusion contribution, I_{laoc}^D , is given by:

$$I_{laoc}^D = -\Gamma_\phi [A_\phi^D (y_c - y_o) - B_\phi^D (x_c - x_o)]. \quad (2.34)$$

If the expressions for A_ϕ^D and B_ϕ^D , given by Eq. (2.11), are substituted back into Eq. (2.34), I_{laoc}^D can be recast into the following form:

$$I_{laoc}^D = C_1^D \phi_1 + C_2^D \phi_2 + C_3^D \phi_3 .. \quad (2.35)$$

Explicit expressions for C_1^D , C_2^D , and C_3^D can be found in the work of Saabas [33].

2.5.2 Advection Contribution

The advection contribution, I_{laoc}^A , for subvolume $laoc$ of element 123 can be written as,

$$I_{laoc}^A = \int_a^o \bar{\mathbf{J}}_A \cdot \bar{\mathbf{n}} \, ds + \int_o^c \bar{\mathbf{J}}_A \cdot \bar{\mathbf{n}} \, ds, \quad (2.36)$$

where the advection flux, $\bar{\mathbf{J}}_A$, is given by,

$$\bar{\mathbf{J}}_A = \rho \bar{\mathbf{V}} \phi = \rho \phi [u \bar{\mathbf{i}} + v \bar{\mathbf{j}}]. \quad (2.37)$$

2.5.2.1 Flow-Oriented Upwind Scheme (FLO)

In the FLO scheme, the interpolation function, ϕ^A , given by Eq. (2.15), is used to approximate ϕ in Eq. (2.36). The first integral in Eq. (2.36) can, therefore, be written as,

$$\begin{aligned} \int_a^o \bar{\mathbf{J}}_A \cdot \bar{\mathbf{n}} \, ds &= \int_a^o \rho \phi^A [u^m \bar{\mathbf{i}} + v^m \bar{\mathbf{j}}] \cdot [dy \bar{\mathbf{i}} - dx \bar{\mathbf{j}}] \\ &= \int_{y_a}^{y_o} \rho \phi^A u^m dy - \int_{x_a}^{x_o} \rho \phi^A v^m dx, \end{aligned} \quad (2.38)$$

where u'' and v'' are the “mass conserving” velocity components obtained from Eq. (2.55). Using the three-point Newton-Cotes quadrature (also known as Simpson’s rule) to evaluate these integrals yields,

$$\int_a^o \bar{\mathbf{J}}_A \cdot \bar{\mathbf{n}} \, ds = \rho \left\{ \frac{(y_o - y_a)}{6} [\phi_a^A u_a'' + 4\phi_r^A u_r'' + \phi_o^A u_o''] - \frac{(x_o - x_a)}{6} [\phi_a^A v_a'' + 4\phi_r^A v_r'' + \phi_o^A v_o''] \right\}. \quad (2.39)$$

Similarly,

$$\int_o^c \bar{\mathbf{J}}_A \cdot \bar{\mathbf{n}} \, ds = \rho \left\{ \frac{(y_c - y_o)}{6} [\phi_c^A u_c'' + 4\phi_t^A u_t'' + \phi_o^A u_o''] - \frac{(x_c - x_o)}{6} [\phi_c^A v_c'' + 4\phi_t^A v_t'' + \phi_o^A v_o''] \right\}. \quad (2.40)$$

After substituting the appropriate expressions for ϕ^A , u'' , and v'' at the integration points, a , c , r , t , and o into Eqs. (2.39) and (2.40), the advection contribution, I_{laoc}^A , can be cast into the following form [33]:

$$\begin{aligned} I_{laoc}^A &= \int_a^o \bar{\mathbf{J}}_A \cdot \bar{\mathbf{n}} \, ds + \int_o^c \bar{\mathbf{J}}_A \cdot \bar{\mathbf{n}} \, ds \\ &= C_1^A \phi_1 + C_2^A \phi_2 + C_3^A \phi_3. \end{aligned} \quad (2.41)$$

2.5.2.2 Mass-Weighted Upwind Scheme (MAW)

In the MAW scheme, the surface integrals in Eq. (2.36) are expressed in the following manner:

$$\begin{aligned} \int_a^o \bar{\mathbf{J}}_A \cdot \bar{\mathbf{n}} \, ds &= \dot{m}_r \phi_r^A, \\ \int_c^o \bar{\mathbf{J}}_A \cdot \bar{\mathbf{n}} \, ds &= \dot{m}_t \phi_t^A, \end{aligned} \quad (2.42)$$

in which the mass flow rates, \dot{m}_r and \dot{m}_t , are given by Eq. (2.23), and the expressions for ϕ_r^A and ϕ_t^A are obtained from Eq. (2.25). The advection contribution, using the *MAW scheme*, for subvolume $laoc$ of element 123 can, therefore, be written as [33],

$$\begin{aligned}
 I_{1aoc}^A &= \int_a^o \bar{\mathbf{J}}_A \cdot \bar{\mathbf{n}} \, ds + \int_o^c \bar{\mathbf{J}}_A \cdot \bar{\mathbf{n}} \, ds \\
 &= \dot{m}_r \phi_r^A - \dot{m}_t \phi_t^A \\
 &= C_1^A \phi_1 + C_2^A \phi_2 + C_3^A \phi_3.
 \end{aligned} \tag{2.43}$$

2.5.3 Source Contribution

The volume integral of the source term, S_s , in the integral conservation equation (2.6), is approximated by linearizing S_s , if required, and expressing it in the form of Eq. (2.7).

The source contribution, I_{1aoc}^S , from subvolume 1aoc of element 123 (shown in Fig. 2.2) is given by,

$$\begin{aligned}
 I_{1aoc}^S &= \int_{1aoc} S_s \, d\Omega \\
 &= \int_{1aoc} (S_c + S_p \phi) \, d\Omega \\
 &= C_1^S \phi_1 + D^S,
 \end{aligned} \tag{2.44}$$

where,

$$\begin{aligned}
 C_1^S &= (S_p)_e^1 \Omega_{1aoc} = (S_p)_e^1 \frac{\Omega_e}{3}, \\
 D^S &= (S_c)_e^1 \Omega_{1aoc} = (S_c)_e^1 \frac{\Omega_e}{3}.
 \end{aligned} \tag{2.45}$$

In this expression, $(S_c)_e^1$ and $(S_p)_e^1$ are the values of S_c and S_p , respectively, evaluated at node 1 of element 123, and Ω_e is the volume of element 123, given by Eq. (2.12).

2.5.4 Discretized Advection-Diffusion Equation

The diffusion, advection, and source contributions derived in the last three subsections can be added together to arrive at the total elemental contribution, I_{1aoc}^t , from subvolume 1aoc of element 123 (Fig. 2.2) to the discretized equation for node 1. I_{1aoc}^t can be expressed as,

$$\begin{aligned}
I_{1aoc}^{\phi} &= \int_a^o (\bar{\mathbf{J}}_D + \bar{\mathbf{J}}_A) \cdot \bar{\mathbf{n}} ds + \int_o^c (\bar{\mathbf{J}}_D + \bar{\mathbf{J}}_A) \cdot \bar{\mathbf{n}} ds - \int_{1aoc} S_{\phi} d\Omega \\
&= I_{1aoc}^D + I_{1aoc}^A - I_{1aoc}^S \\
&= C_1^{\phi} \phi_1 + C_2^{\phi} \phi_2 + C_3^{\phi} \phi_3 + D^{\phi},
\end{aligned} \tag{2.46}$$

in which I_{1aoc}^D is obtained from Eq. (2.35); I_{1aoc}^A is obtained from either Eq. (2.41) or Eq. (2.43) depending on whether the FLO or MAW scheme is used, respectively, and I_{1aoc}^S is obtained from Eq. (2.44).

Expressions similar to Eq. (2.46) can be derived for each of the elements associated with the internal node, i , shown in Fig. 2.1 (a). Adding these contributions together yields the complete discretized equation for node i , which can be written in the following form:

$$a_i^{\phi} \phi_i = \sum_{nb} a_{nb}^{\phi} \phi_{nb} + b_i^{\phi}, \tag{2.47}$$

where the index, “ nb ”, refers to the *neighbouring* nodes surrounding node i .

2.5.5 Discretized Momentum Equations

The momentum equations (2.2)-(2.3) can be cast in the general form of Eq. (2.4) by making the appropriate specializations given in Table 2.1. The derivation of the discretized equations for u and v is identical to the derivation of the discretized ϕ -equation presented in Sections 2.5.1 – 2.5.4, except for the treatment of the pressure gradients, which is given in this section. The volume integral of the source and pressure-gradient terms in the x -momentum equation for subvolume $1aoc$ of element 123 (Fig. 2.2), is approximated as,

$$\begin{aligned}
\int_{1aoc} \left(S_u - \frac{\partial p}{\partial x} \right) d\Omega &= \int_{1aoc} \left(S_c^u + S_p^u u - \frac{\partial p}{\partial x} \right) d\Omega \\
&= \left[(S_c^u)_e^1 + (S_p^u)_e^1 u_1 - \left(\frac{\partial p}{\partial x} \right)_e \right] \frac{\Omega_e}{3},
\end{aligned} \tag{2.48}$$

where Ω_e is the element volume obtained from Eq. (2.12). $(S_c^u)_e^1$ and $(S_p^u)_e^1$ are the values of S_c^u and S_p^u , respectively, evaluated at node 1 of element 123, and $(\partial p / \partial x)_e$ is

the element pressure gradient given by Eq. (2.28). Similarly, the source and pressure-gradient contribution to the discretized y -momentum equation is approximated as,

$$\int_{\text{loc}} \left(S_v - \frac{\partial p}{\partial y} \right) d\Omega = \left[(S_c^v)_e^1 + (S_p^v)_e^1 v_1 - \left(\frac{\partial p}{\partial y} \right)_e \right] \frac{\Omega_e}{3}. \quad (2.49)$$

The complete discretized equations for u and v , after all diffusion, advection, source and pressure-gradient contributions associated with node 1 are assembled, can be expressed in the following forms:

$$a_i^u u_i = \sum_{nb} a_{nb}^u u_{nb} + b_i^u - \left(\overline{\frac{\partial p}{\partial x}} \right)_i \Omega_i, \quad (2.50)$$

$$a_i^v v_i = \sum_{nb} a_{nb}^v v_{nb} + b_i^v - \left(\overline{\frac{\partial p}{\partial y}} \right)_i \Omega_i, \quad (2.51)$$

where $(\overline{\partial p / \partial x})_i$ and $(\overline{\partial p / \partial y})_i$ are volume-averaged pressure gradients associated with the control-volume, Ω_i , surrounding node i .

2.5.6 Interpolation of the Velocity Components in the Mass-Flux Terms

When the velocity components are considered as transported scalars in the momentum equations, they are interpolated linearly in the diffusion terms as described in Section 2.4.3. In the advection terms they are interpolated using either the FLO scheme or the MAW scheme, as presented in Section 2.4.4. However, when the velocity is considered as part of the mass-flux terms appearing in the continuity, momentum, and advection-diffusion equations, a special *momentum interpolation scheme* is used. This interpolation scheme allows for the formulation of a co-located, equal-order CVFEM that does not suffer from spurious pressure oscillations in the simulation of incompressible fluid flow. In this scheme, the interpolation functions for the velocity components are defined using special forms of the discretized momentum equations. The main ideas behind this scheme are borrowed from the work of Prakash and Patankar [30]. Similar schemes have been used by Rhie and Chow [89], Peric *et al.* [90], and Rice and Schnipke [87].

Equations (2.50) and (2.51) can be rewritten in the following manner, after dividing through by a_i^u and a_i^v , respectively,

$$u_i = \hat{u}_i - d_i^u \left(\frac{\partial p}{\partial x} \right)_i; \quad v_i = \hat{v}_i - d_i^v \left(\frac{\partial p}{\partial y} \right)_i, \quad (2.52)$$

where,

$$\hat{u}_i = \frac{\sum_{nb} a_{nb}^u u_{nb} + b_i^u}{a_i^u}; \quad \hat{v}_i = \frac{\sum_{nb} a_{nb}^v v_{nb} + b_i^v}{a_i^v}, \quad (2.53)$$

and,

$$d_i^u = \frac{\Omega_i}{a_i^u}; \quad d_i^v = \frac{\Omega_i}{a_i^v}. \quad (2.54)$$

The interpolation functions used to approximate the velocity components in the mass-flux terms are defined as,

$$u'' = \hat{u} - d_e^u \left(\frac{\partial p}{\partial x} \right)_e; \quad v'' = \hat{v} - d_e^v \left(\frac{\partial p}{\partial y} \right)_e. \quad (2.55)$$

While u_i and v_i in Eq. (2.52) are defined at each node i in the finite element mesh, u'' and v'' are *interpolation functions* defined over each *element*. To evaluate these functions at specific locations within the element, the pseudo-velocities, \hat{u} and \hat{v} , are interpolated linearly from the corresponding nodal values. The pressure-gradient coefficients, d_e^u and d_e^v , are evaluated at the centroid (that is, evaluated as the arithmetic average of the three nodal values), and assumed to prevail over the corresponding element. Justification for this treatment of d^u and d^v is discussed in the next section. The pressure gradients, $(\partial p / \partial x)_e$ and $(\partial p / \partial y)_e$, are the constant elemental pressure gradients given by Eq. (2.28) and *not* the volume-averaged gradients of Eq. (2.52). This treatment of the pressure gradients is the key feature of the interpolation scheme, which prevents the occurrence of checkerboard-type pressure oscillations in the calculated pressure field. It is noted that this scheme is a consistent one, in that the element pressure gradients approach the corresponding volume-averaged nodal values as the element size approaches zero (assuming that the corresponding pressure gradient in the analogous continuum problem is continuous at this nodal location).

2.5.7 Discretization Equation for Pressure

The integral mass conservation equation applied to the control-volume surrounding node i in Fig. 2.1, can be written in the following manner,

$$\left[\int_a^o \rho \bar{\mathbf{V}}^m \cdot \bar{\mathbf{n}} ds + \int_o^c \rho \bar{\mathbf{V}}^m \cdot \bar{\mathbf{n}} ds \right] + [\text{similar contributions from other elements associated with node } i] + [\text{boundary contributions, if applicable}] = 0, \quad (2.56)$$

where $\bar{\mathbf{V}}^m = u^m \bar{\mathbf{i}} + v^m \bar{\mathbf{j}}$, and the “mass conserving” velocity components, u^m and v^m are given by Eq. (2.55). The first integral in Eq. (2.56) can be written as follows,

$$\begin{aligned} \int_a^o \rho \bar{\mathbf{V}}^m \cdot \bar{\mathbf{n}} ds &= \int_{y_a}^{y_o} \rho u^m dy - \int_{x_a}^{x_o} \rho v^m dx \\ &= \rho [u_r^m (y_o - y_a) - v_r^m (x_o - x_a)] \end{aligned} \quad (2.57)$$

Using the interpolation functions for u^m and v^m , given by Eq. (2.55), yields,

$$\int_a^o \rho \bar{\mathbf{V}}^m \cdot \bar{\mathbf{n}} ds = \rho \left\{ \left[\hat{u}_r - d_e^u \left(\frac{\partial p}{\partial x} \right)_e \right] (y_o - y_a) - \left[\hat{v}_r - d_e^v \left(\frac{\partial p}{\partial y} \right)_e \right] (x_o - x_a) \right\} \quad (2.58)$$

After deriving a similar expression for the second integral in Eq. (2.56), and substituting the expressions for $(\partial p / \partial x)_e$ and $(\partial p / \partial y)_e$, given by Eq. (2.28), into these expressions, the total contribution, I_{1aoc}^p , from subvolume 1aoc of element 123 (Fig. 2.2) to the discretized equation for p at node 1 can be represented by [33],

$$\begin{aligned} I_{1aoc}^p &= \int_a^o \rho \bar{\mathbf{V}}^m \cdot \bar{\mathbf{n}} ds + \int_o^c \rho \bar{\mathbf{V}}^m \cdot \bar{\mathbf{n}} ds \\ &= C_1^p p_1 + C_2^p p_2 + C_3^p p_3 + D^p. \end{aligned} \quad (2.59)$$

Expressions analogous to I_{1aoc}^p can be derived for each of the elements associated with node i (Fig. 2.1). Substituting these expressions back into Eq. (2.56) yields the complete discretized equation for node i , which can be expressed in the following form:

$$a_i^p p_i = \sum_{nb} a_{nb}^p p_{nb} + b_i^p. \quad (2.60)$$

As was mentioned in Section 2.5.6, the pressure-gradient coefficients, d_e^u and d_e^v , are evaluated, in each element, as the average of the three corresponding nodal values. Thus, d^u and d^v are assumed to *prevail* over their respective element. This is in contrast to previous CVFEM formulations [60,77,81], in which d^u and d^v were interpolated *linearly* within each element. Justification for the prevailing treatment of d^u and d^v in the present CVFEM formulation is discussed in the following paragraph. Further discussion can be found in the paper by Baliga [92].

It is first noted that the discretized equation for pressure, given by Eq. (2.60), is akin to a discretized *Poisson* equation. This is seen by observing that,

$$\begin{aligned}\bar{\nabla} \cdot \bar{\mathbf{V}}^m &= \bar{\nabla} \cdot (\hat{\bar{\mathbf{V}}} - d^u \bar{\nabla} p) = 0 \\ \Rightarrow \bar{\nabla} \cdot (d^u \bar{\nabla} p) &= \bar{\nabla} \cdot \hat{\bar{\mathbf{V}}},\end{aligned}\quad (2.61)$$

where $\hat{\bar{\mathbf{V}}} = \hat{u}\bar{\mathbf{i}} + \hat{v}\bar{\mathbf{j}}$ is the pseudo-velocity vector. This last expression can be written in this form when d^u is identically equal to d^v for all interior nodes, as is the case when $S_u = S_v$. Thus, it is evident from Eq. (2.61) that d^u is analogous to a diffusion coefficient. Therefore, the discussion in Section 2.4.3.1 pertaining to positive diffusion contributions to the coefficients in the discretized equations applies here as well. To ensure positive coefficients in the discretized pressure equation, the computational grid must be composed of acute-angle triangles, and the centroidal values d^u and d^v must be assumed to *prevail* over each element. If d^u and d^v are interpolated linearly within each element, positive coefficients cannot be guaranteed. The positive-coefficient property is highly desirable because it aids in avoiding spurious pressure oscillations, and results in a more stable and convergent iterative solution process [36,38,60,82]. It should be mentioned that if $S_u \neq S_v$, the prevailing d^u and d^v approach is still recommended, however, positive coefficients in the pressure equation are *not* guaranteed in this case.

2.5.8 Boundary Conditions

The final discretized ϕ -equation for each *internal* node, as derived in Sections 2.5.1 – 2.5.4, is given by Eq. (2.47). However, for those nodes lying on the *boundary* of the

domain, Eq. (2.47) is incomplete. To complete this equation, the diffusion and advection transport associated with ϕ across the boundary must be properly included. The following derivation pertains to the contribution from the boundary segment, $i-a$, shown in Fig. 2.1 (b), to the discretized equation for node i .

Dirichlet Conditions. Dirichlet boundary conditions can be expressed as,

$$\phi = \phi_o \quad \text{on} \quad \partial\Omega_o, \quad (2.62)$$

where ϕ_o is a specified distribution of ϕ over the portion of the boundary denoted by $\partial\Omega_o$. For a boundary node, i , lying on $\partial\Omega_o$, the coefficients of the corresponding discrete ϕ -equation are overwritten as follows:

$$a_i^o = 1; \quad a_{nb}^o = 0; \quad b_i^o = (\phi_o)|_i. \quad (2.63)$$

Thus, the resulting discrete ϕ -equation for node i takes on the following trivial form:

$$\phi_i = (\phi_o)|_i. \quad (2.64)$$

Neumann Conditions. In the case of Neumann boundary conditions, the outward normal derivative of ϕ is specified over the portion of the boundary denoted by $\partial\Omega_n$. This is written explicitly as,

$$\bar{\nabla}\phi \cdot \bar{n} = \left(\frac{\partial\phi}{\partial n} \right)_o \quad \text{on} \quad \partial\Omega_o. \quad (2.65)$$

Thus, the total advection-diffusion flux across $\partial\Omega_n$ is expressed as,

$$\begin{aligned} \bar{J}_o \cdot \bar{n} &= (\rho \bar{\nabla}\phi - \Gamma_o \bar{\nabla}\phi) \cdot \bar{n} \\ &= \rho V_n \phi - \Gamma_o \left(\frac{\partial\phi}{\partial n} \right)_o \quad \text{on} \quad \partial\Omega_o, \end{aligned} \quad (2.66)$$

where V_n is the component of velocity outward and normal to the boundary. The boundary contribution from surface $i-a$ to the discretized equation for node i is therefore given by,

$$\int_i^a \bar{J}_o \cdot \bar{n} ds = \int_i^a \left[\rho V_n \phi - \Gamma_o \left(\frac{\partial\phi}{\partial n} \right)_o \right] ds. \quad (2.67)$$

Mixed Neumann-Dirichlet Conditions. For this type of boundary condition (also known as a Robin condition) a linear combination of ϕ and its outward normal derivative, $\partial\phi/\partial n$, is specified on the portion of the boundary denoted by $\partial\Omega_r$. That is,

$$\bar{\nabla}\phi \cdot \bar{n} + \lambda_1\phi = \lambda_2 \quad \text{on} \quad \partial\Omega_r, \quad (2.68)$$

where λ_1 and λ_2 have specified distributions over $\partial\Omega_r$. In this case, the total advection-diffusion flux across $\partial\Omega_r$ is stated as,

$$\begin{aligned} \bar{\mathbf{J}}_\phi \cdot \bar{n} &= (\rho\bar{\nabla}\phi - \Gamma_\phi\bar{\nabla}\phi) \cdot \bar{n} \\ &= \rho V_n \phi - \Gamma_\phi(\lambda_2 - \lambda_1\phi) \\ &= (\rho V_n + \Gamma_\phi\lambda_1)\phi - \Gamma_\phi\lambda_2 \quad \text{on} \quad \partial\Omega_r, \end{aligned} \quad (2.69)$$

and the corresponding boundary contribution from surface $i-a$ to the discretized equation for node i is given by,

$$\int_i^a \bar{\mathbf{J}}_\phi \cdot \bar{n} ds = \int_i^a [(\rho V_n + \Gamma_\phi\lambda_1)\phi - \Gamma_\phi\lambda_2] ds. \quad (2.70)$$

Generalized Boundary Condition. The generalized boundary condition for the advection-diffusion equation is given by Eq. (2.5). It is rewritten here as,

$$\psi[\phi, \bar{\nabla}\phi \cdot \bar{n}] = 0 \quad \text{on} \quad \partial\Omega_g. \quad (2.5)$$

This condition is applied over the portion of the boundary denoted by $\partial\Omega_g$. This type of boundary condition can be handled, in general, by appropriately linearizing ψ and casting the linearized expression in the form of Eq. (2.68). An iterative procedure is required in this case. The parameters, λ_1 and λ_2 , are updated after each iteration until Eq. (2.5) is satisfied.

Special Treatment. The boundary treatments derived for the ϕ -equation (above) are also applicable to the momentum equations with ϕ appropriately replaced by either u or v .

For the mass conservation equation, Eq. (2.56), only boundaries that have mass flowing across them (inflow or outflow boundaries) have non-zero contributions. With respect to

the boundary surface $i-a$ (Fig. 2.1 (b)) the mass flow contribution to the discrete equation for node i can be expressed as,

$$\int_i^a \rho \bar{\mathbf{V}} \cdot \bar{\mathbf{n}} ds = \int_i^a \rho V_n ds. \quad (2.71)$$

It should be noted here that this last expression is computed using the latest available nodal values of the velocity, $\bar{\mathbf{V}}$, and *not* $\bar{\mathbf{V}}^m$. Only mass flow rates across control-volume faces in the *interior* of the domain are calculated using $\bar{\mathbf{V}}^m$.

At nodes that lie on boundaries with prescribed velocities (such as a solid wall or inflow boundary), the pressure-gradient coefficients d^u and d^v are set equal to zero, and, therefore,

$$u^m = \hat{u} = u; \quad v^m = \hat{v} = v. \quad (2.72)$$

At *outflow boundaries*, it is often assumed that the normal component of the *diffusion* flux associated with the scalar, ϕ (or velocity component u or v), is negligible. This corresponds to the following Neumann boundary condition on the outflow boundary denoted by $\partial\Omega_{\text{outflow}}$:

$$\bar{\nabla} \phi \cdot \bar{\mathbf{n}} = \frac{\partial \phi}{\partial n} = 0 \quad \text{on} \quad \partial\Omega_{\text{outflow}}, \quad (2.73)$$

thus, the boundary treatment for Neumann conditions, given by Eq. (2.67), applies here with $(\partial\phi/\partial n)_o = 0$.

In the case of incompressible Navier-Stokes flows, the level of pressure is unimportant, because only the pressure *gradient* appears in the momentum equations. Thus, the correct pressure field is determined to within an *arbitrary* constant when the corresponding velocity field, obtained from the momentum equations, satisfies the continuity equation. During a numerical simulation, however, the value of pressure at one node is often fixed to a convenient value. This tends to promote convergence of the iterative solution algorithm, and it prevents the occurrence of excessively large levels of pressure during the solution process.

2.6 Solution Procedure

An iterative sequential variable adjustment algorithm, similar to the *enhanced sequential solution algorithm* (ESSA) of Afshar and Baliga [138], is used to solve the nonlinear, coupled sets of discretized equations for u , v , p , and ϕ . This procedure consists of repeated outer and inner iteration loops. In each outer loop, the coefficients of the discretized equations for the velocity components are calculated and stored. Following this, an inner loop is performed, in which repeated solutions of the discretized equations are done sequentially, without recalculating the coefficients of the u and v equations. This inner loop enhances the *coupling* between the nominally linear equations before updating the *nonlinearities* of the u and v equations in the outer loop. The following is an outline of the main steps in the algorithm:

Beginning of Solution Procedure

1. Provide suitable guesses for the unknown nodal values of u , v , p , and any other scalar fields, ϕ , of interest. Also assign appropriate values of ρ , μ , S_u , S_v , Γ_ϕ , and S_ϕ .

Beginning of Outer Loop

2. Calculate and assemble all coefficient contributions for the discretized u and v equations (Eq. (2.50) and (2.51), respectively) *excluding* contributions from the volume integral of the pressure gradients. Include boundary contributions at this stage, but *do not* overwrite coefficients to treat Dirichlet boundary conditions yet. *Store these coefficients for use in the inner loop.*

Beginning of Inner Loop

3. Calculate the nodal values of \hat{u} , \hat{v} , d^u , and d^v using Eqs. (2.53) and (2.54). Modify these values appropriately at nodes where u and/or v are specified (see Section 2.5.8, Eq. (2.72)).
4. Calculate and assemble coefficients in the discretized p equation (Eq. (2.60)), and solve for p .
5. Add the appropriate pressure-gradient contributions to the coefficients of the *incomplete u and v equations assembled in step 2*.

6. Under-relax the equation sets obtained in step 5, perform any Dirichlet boundary treatment, if needed, and then solve for new u and v distributions.
7. Calculate and assemble coefficients in the discretized equations for any other dependent variables that are *coupled* to the flow field. Under-relax these equation sets, apply boundary treatment (including Dirichlet treatment, if applicable), and solve them sequentially (Note: It may be advantageous to solve some or all of these auxiliary equations in the outer loop. This is determined heuristically).
8. Check for inner-loop convergence, return to step 3, if necessary, otherwise exit the inner loop.

End of Inner Loop

9. Check for outer-loop convergence, return to step 2, if necessary, otherwise exit the outer loop.

End of Outer Loop

10. Solve for other dependent variables of interest that do not influence the velocity field.

End of Solution Procedure

In this procedure, the discretized equations for u and v (and ϕ , if applicable), are appropriately under-relaxed, using an implicit under-relaxation procedure [82]. These sets of linearized equations are solved using a point-by-point successive over-relaxation (SOR) Gauss-Siedel method. An additive correction multigrid method, which is derived in Chapter 3, is used to accelerate convergence in the pressure equation.

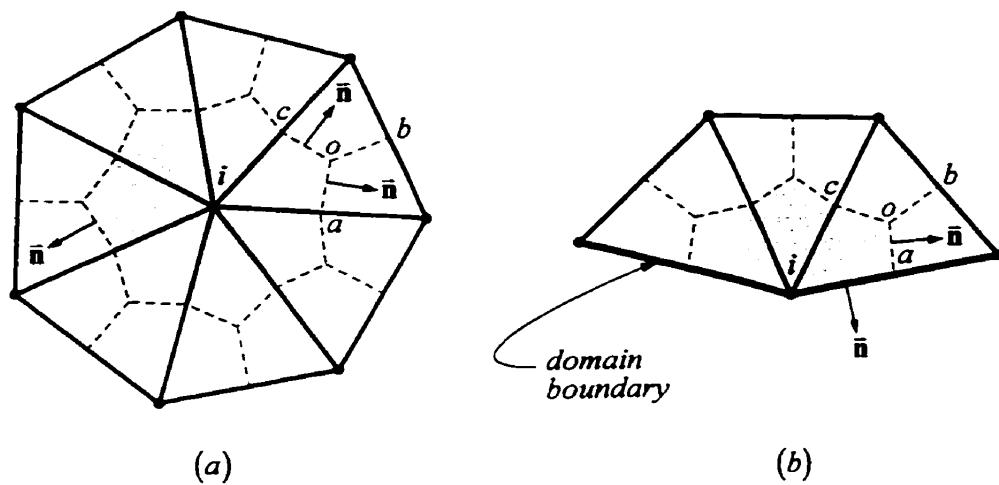


Figure 2.1: Typical patch of elements surrounding (a) an internal node; and (b) a boundary node. The shaded region represents the control volume associated with node i .

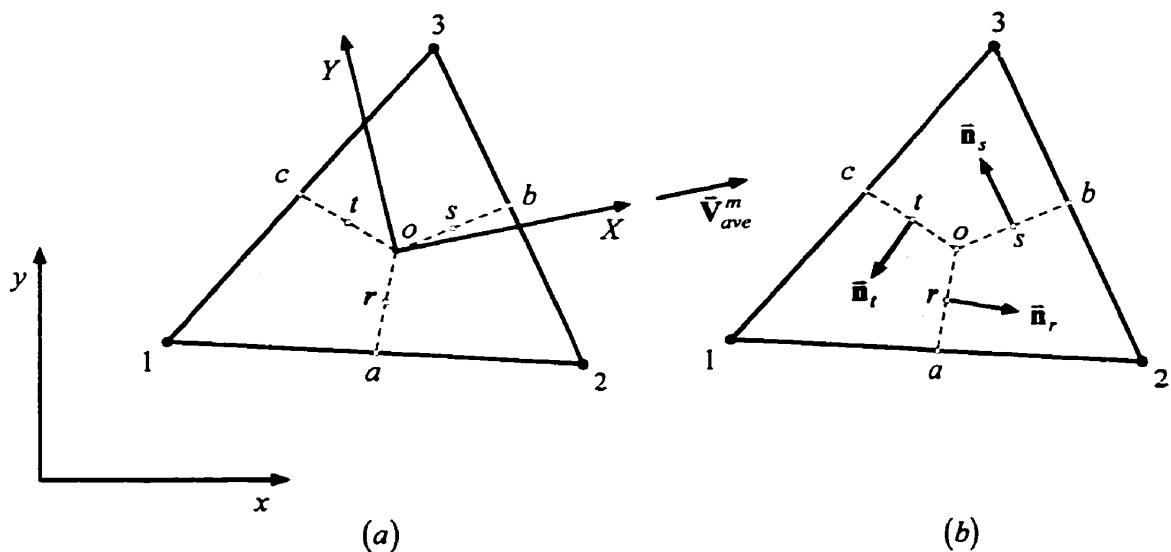


Figure 2.2: A typical three-node triangular element and related nomenclature: (a) local rotated coordinate system (X , Y) used in the FLO scheme; and (b) convention for positive unit normal vectors in the derivation of the MAW scheme.

Chapter 3

ADDITIVE CORRECTION MULTIGRID WITH ADAPTIVE VOLUME AGGLOMERATION

A linear equation solver using the *algebraic multigrid method* is presented in this chapter. It employs a grid-coarsening algorithm based on an adaptive volume-agglomeration technique similar to the ones proposed by Raw [41] and Elias *et al.* [42]. The prolongation and restriction operators are those implied by the additive correction multigrid (ACM) method [41-44,46-50]. Verification of the current ACM implementation is provided in Chapter 6.

3.1 Introduction

Classical point-by-point iterative solvers, such as the Gauss-Seidel (GS) point solver, are usually efficient at reducing the residuals of sparse linear equation sets in the first few sweeps (or iterations), but then “stall” (or suffer a drastic reduction in the rate of convergence) shortly thereafter. This is especially true for large, stiff, systems of discrete equations, such as those commonly encountered in the simulation of elliptic fluid flow problems. The initial rapid reduction in the residuals reflects the effectiveness of the GS solver at eliminating high-frequency components of the error distribution. *In this chapter, the error refers to the difference between the latest available estimate of the variable of interest (during the iterative solution procedure) and the exact solution to the discretized (algebraic) equations.* A high-frequency error component is a Fourier component that oscillates with a wavelength comparable to, or smaller than, the local grid spacing or element size. The “stalling”, which occurs after the first few sweeps of the GS solver, indicates its poor performance at reducing low-frequency, or “global”, error components.

This is to be expected since GS relies on local information supplied through each nodal stencil. The preferential reduction of high-frequency error components by GS can be turned from a disadvantage into an advantage by applying GS to an appropriate sequence of grids, from the original fine grid to a very coarse grid. On each grid, GS is applied for only a few sweeps, until the frequency components that are high with respect to that grid are efficiently removed. A typical multigrid strategy, therefore, consists of sweeping on the fine-grid equations until the high-frequency components have been annihilated. The solution is then transferred to a coarser grid where the remaining low-frequency components now manifest themselves as high-frequency components, so they can be efficiently eliminated with just a few sweeps of the base solver. This process is repeated recursively through the entire sequence of grids. On each grid level, a certain bandwidth of error is removed. When the coarsest grid of the sequence is reached, corrections are then transferred back to each successively finer grid. This entire process is repeated until overall convergence of the problem is achieved. A monograph on the philosophy and theory behind the multigrid method is provided by Briggs [49].

Several different types of multigrid methods have been proposed in the mathematical and engineering literature. A review has been put forward by Mavriplis [50]. These various types differ on how coarse grids and coarse-grid equations are generated (restriction), and on how corrections are made to the fine-grid solution, using improvements achieved on the next coarsest grid (prolongation). Other issues include the choice of base solver (for example, GS, Gauss-Jacobi (GJ), Incomplete Lower Upper Factorization (ILU)) and the type of cycle used to control the sequence in which grids are visited (for example, V-cycle, W-cycle, flexible cycle).

The current implementation uses an additive correction multigrid (ACM) method. The ACM is a natural choice for CVFEMs because it is based on physical conservation principles and has been developed within the engineering tradition of using block correction [82] to accelerate convergence. The ACM solver implemented in the present work has been primarily used for diffusion-type problems including the pressure equation (see Section 2.5.7) in the simulation of incompressible Navier-Stokes flows. As pointed out by Hutchinson [139], iterative convergence in the pressure equation can be

prohibitively slow if standard GS-type point-by-point solvers are used without the benefit of acceleration techniques such as the ACM.

In the remainder of this chapter, details of the current ACM scheme are presented. Verification of the method is provided in Section 6.1.

3.2 Grid Coarsening

As was stated in the previous section, motivation behind the development of many different types of multigrid techniques was based on the observation that simple iterative solvers, such as, GS or GJ, reduce high-frequency error components preferentially over low-frequency components. In multiple dimensions, however, there are other aspects that influence the convergence rates of these iterative solvers. To illustrate this consider the following discrete equation for ϕ at node P ,

$$a_P \phi_P = a_N \phi_N + a_S \phi_S + a_{NE} \phi_{NE} + \dots + b_P. \quad (3.1)$$

This equation could have been obtained, for example, from a standard CVFEM approximation on a two-dimensional mesh, such as the one shown in Fig. 3.1. Using GJ as an example iterative solver, the solution after the n^{th} iteration would be,

$$\phi_P^n = \frac{a_N \phi_N^{n-1} + a_S \phi_S^{n-1} + a_{NE} \phi_{NE}^{n-1} + \dots + b_P}{a_P}. \quad (3.2)$$

Consider the hypothetical case where $a_S = 100 a_{NE}$. Then upon examining Eq. (3.2), it is evident that a given change in ϕ_S would influence the new solution 100 times more than the same change in ϕ_{NE} . This implies that error components in large-coefficient directions are reduced more rapidly than error components in small coefficient directions. Thus, not only does GJ preferentially reduce high-frequency error components, it does so preferentially in large-coefficient directions. The main idea in the present grid-coarsening algorithm, therefore, is to first identify the direction(s) for which the base solver is able to efficiently reduce error components on the fine grid. The coarse grid is then created from the fine grid by coarsening only in those directions. Coarsening is achieved, in the

CVFEM context, by agglomerating fine-grid control-volumes (CVs) to form larger, coarse-grid CVs.

3.2.1 Volume-Agglomeration Procedure

The grid-coarsening algorithm presented here is similar to the one proposed by Raw [41]. In the remainder of this subsection, references are made to the following fine-grid equation for node i :

$$a_i \phi_i = \sum_{nb \text{ of } i} a_{nb} \phi_{nb} + b_i, \quad (3.3)$$

where the summation in this equation is performed over all neighbour nodes “ nb ” of i . The procedure involved in constructing a coarse grid from an existing fine grid consists of two *loops* outlined below.

The following terminology is used: (i) any reference to a particular fine-grid *node* also refers to its associated fine-grid CV (for example, if node i is said to have been agglomerated with another node, this actually means that their associated CVs have been agglomerated); (ii) a *block* refers to a coarse-grid CV, which is composed of agglomerated fine-grid CVs.

Loop I: Loop over all fine-grid nodes in the domain

Start of Loop I

- Determine whether the i^{th} node in the loop has already been agglomerated.
 - If it has, increment i by 1 and go back to the beginning of the loop.
 - If it has not, determine which *unagglomerated* neighbour-node of i has the largest corresponding neighbour coefficient, $a_{nb'}^{\max}$, where, $a_{nb'}^{\max} = \max_{nb'}(a_{nb'})$. The subscript, nb' , represents the neighbour-nodes of i which have *not yet been agglomerated*.
 - If $a_{nb'}^{\max}$ exists, then agglomerate node i with that neighbour node and designate both of these fine-grid nodes as having been agglomerated.

- If a_{nb}^{\max} does not exist (that is, if there are no unagglomerated neighbour-nodes left surrounding node i), then designate i as an *orphan node*.
- Increment i by 1 and repeat this process until all fine-grid nodes have been visited once.

End of Loop I

Loop II: Loop over all orphan nodes (designated as such in Loop I)

Start of Loop II

- For the k^{th} orphan node in the loop, determine which fine-grid neighbour node of k has the largest corresponding neighbour coefficient, $a_{nb \text{ of } k}^{\max}$, defined as,

$$a_{nb \text{ of } k}^{\max} = \max_{nb \text{ of } k} (a_{nb \text{ of } k}).$$

- Agglomerate node k with the coarse-grid *block* containing the fine-grid neighbour node associated with $a_{nb \text{ of } k}^{\max}$.
- Increment k by 1 and repeat this process until all orphan nodes have been visited once.

End of Loop II

One difference between the current agglomeration scheme and the one proposed by Raw [41], is that the current scheme, on average, uses fewer fine-grid CVs to form a single coarse-grid CV. In Raw's scheme, the size of the coarse-grid CVs can be controlled, to a certain extent, using user-specified parameters. The current scheme was purposefully designed to yield coarse-grid blocks with as few fine-grid CVs as possible. This results in more grid levels in the multigrid cycle, however, it also reduces the frequency bandwidth of error components that need to be damped on each grid. In principle this means that fewer sweeps of the base solver are required on each grid to achieve the same residual reduction after one complete cycle. Since the effectiveness of the base solver diminishes

rapidly as a function of the number of iterations, it is usually advantageous to minimize the total number of sweeps on each grid, if possible.

To create the entire grid hierarchy in the multigrid cycle, the grid-coarsening algorithm outlined above is first applied to the original fine grid to form a coarse grid. The procedure is then applied to the coarse grid to form yet another coarser grid, and the rest of the hierarchy is constructed recursively. The process is stopped, in the present implementation, once the coarsest grid has 20 control volumes or less.

3.2.2 Solution-Adaptive Agglomeration

If the ACM is being applied to a nonlinear problem where, for example, the coefficients of Eq. (3.3) are themselves functions of the unknown scalar variable, ϕ , then the optimal agglomeration pattern for each grid in the hierarchy will likely change during the iterative solution procedure as these coefficients are being updated. In such cases, it may be advantageous to reconstruct the grid hierarchy at several times during the nonlinear solution process to accommodate for the changing coefficients. To accomplish this, the grid-coarsening algorithm outlined in Section 3.2.1 can be reapplied to the original fine grid using the *updated* neighbour coefficients to guide in the directional volume agglomeration.

3.3 Restriction and Prolongation

In Section 3.2, an algorithm was presented for determining a grid hierarchy. Once this is established, the task is to define coarse-grid equations, the solutions of which are to be used to reduce low-frequency error components on the finer grids.

In the present scheme an additive correction multigrid (ACM) method is used to define the restriction and prolongation operators. In the ACM, the coarse-grid equation for a particular block of agglomerated fine-grid nodes is derived by appropriately adding the fine-grid equations so that the conserved quantity associated with the scalar variable on each fine-grid CV is also conserved over the associated coarse block. The solution to this coarse-grid equation is used as a uniform correction for each fine-grid node within the

associated block. The advantages to this approach include the following: (i) It is based on physically meaningful principles that are well suited for a CVFEM formulation; (ii) The resulting coarse-grid equations have well defined properties that aid in debugging and validating the corresponding implementation; (iii) The implicit restriction and prolongation operators in this approach are relatively simple, making the implementation of the method fairly straight forward.

The following is a detailed derivation of the restriction and prolongation operators implied by the ACM:

The residual, r_i , of the fine-grid equation for node i at an intermediate stage of the iterative solution process is given by,

$$r_i = a_i \phi_i - \sum_{nb \text{ of } i} a_{nb} \phi_{nb} - b_i. \quad (3.4)$$

Referring to Fig. 3.2, let Φ_{I/I_i} be the uniform correction for each fine-grid node i within the coarse block, I . Thus,

$$\bar{\phi}_i = \phi_i + \Phi_{I/I_i}, \quad (3.5)$$

where $\bar{\phi}_i$ is the corrected fine-grid solution at node i . The new residual, \bar{r}_i , is, therefore, given by,

$$\bar{r}_i = a_i \bar{\phi}_i - \sum_{nb \text{ of } i} a_{nb} \bar{\phi}_{nb} - b_i. \quad (3.6)$$

The expression for $\bar{\phi}_i$, given by Eq. (3.5), and similar expressions for the neighbour nodes, $\bar{\phi}_{nb}$, can be substituted into Eq. (3.6) to yield,

$$\bar{r}_i = a_i (\phi_i + \Phi_{I/I_i}) - \sum_{nb \text{ of } i} a_{nb} (\phi_{nb} + \Phi_{NB/nb}) - b_i. \quad (3.7)$$

Using Eq. (3.4), \bar{r}_i can be rewritten as,

$$\bar{r}_i = r_i + a_i \Phi_{I/I_i} - \sum_{nb \text{ of } i} a_{nb} \Phi_{NB/nb}, \quad (3.8)$$

where $\Phi_{NB/nb}$ is the correction corresponding to the coarse-grid block NB , which contains the fine-grid neighbour-node, nb , of node i . Note that block NB can either be block I itself, or a coarse-grid neighbour of block I .

Conservation over block I is imposed by requiring that the sum of the new fine-grid residuals in I be zero. This is written as,

$$\sum_{i \text{ in } I} \bar{r}_i = 0. \quad (3.9)$$

Substituting Eq. (3.8) into this summation gives,

$$\sum_{i \text{ in } I} r_i + \sum_{i \text{ in } I} a_i \Phi_{I/I} - \sum_{i \text{ in } I} \sum_{nb \text{ of } i} a_{nb} \Phi_{NB/nb} = 0. \quad (3.10)$$

Grouping together coefficients that multiply the same Φ , Eq. (3.10) can be recast into the following form:

$$A_I \Phi_I = \sum_{NB \text{ of } I} A_{NB} \Phi_{NB} + B_I, \quad (3.11)$$

where NB now refers to the coarse-grid neighbours of Block I (not including I itself).

The values of A_I , A_{NB} , and B_I follow directly from Eq. (3.10). For example,

$$B_I = - \sum_{i \text{ in } I} r_i. \quad (3.12)$$

In summary, Eq. (3.11) represents the coarse-grid equation. The coefficients A_I , A_{NB} , and B_I are obtained by *restricting* the fine-grid residuals and coefficients according to Eq. (3.10). Equation (3.5) represents the *prolongation* step in which the fine-grid solution is corrected using the coarse-grid solution.

3.4 Multigrid Cycle

With a grid hierarchy established (Section 3.2), and coarse-grid equations defined (Section 3.3), the next component of a complete multigrid algorithm addresses the sequence with which grid levels are visited. The current implementation uses a hybrid V-cycle. This is illustrated in Fig. 3.3, where the standard V-cycle and the hybrid cycle are

depicted. The V-cycle algorithm telescopes down to the coarsest grid, and then works its way back up to the finest grid. The hybrid cycle employs a series of V-type-cycles, as shown in Fig. 3.3. The main advantage of the hybrid cycle is that each grid level gets to pass its residuals down to the coarser grids several times and receive corrections several times. This approach gives the multigrid algorithm more opportunity to work on the coarser grid levels where the base solver sweeps are relatively inexpensive.

In the present implementation, the default number of sweeps of the base solver going downward in the cycle (pre-restriction) is 2. Going upward (post-prolongation), the default is set to 3 sweeps. The cycling algorithm, implemented in the present ACM solver, is capable of automatically increasing the number of post-prolongation sweeps, during the solution procedure, if the root-mean-square of the fine-grid residuals has not been reduced by a minimum predetermined factor after one complete ACM cycle.

3.5 Relaxation Scheme

The relaxation scheme (base iterative solver) must be capable of removing high-frequency error components efficiently (as discussed in Section 3.1), and preferentially in large-coefficient directions (Section 3.2). A base solver that possesses both of these properties is the successive over-relaxation Gauss-Seidel (SOR-GS) iterative point-by-point solver [140]. This is the relaxation scheme used in the present multigrid implementation. The value of the over-relaxation parameter used in the SOR-GS base solver is typically 1.4. The system of equations on the coarsest grid level is solved directly using a modified Lower-Upper (LU) Decomposition or Choleski algorithm [140].

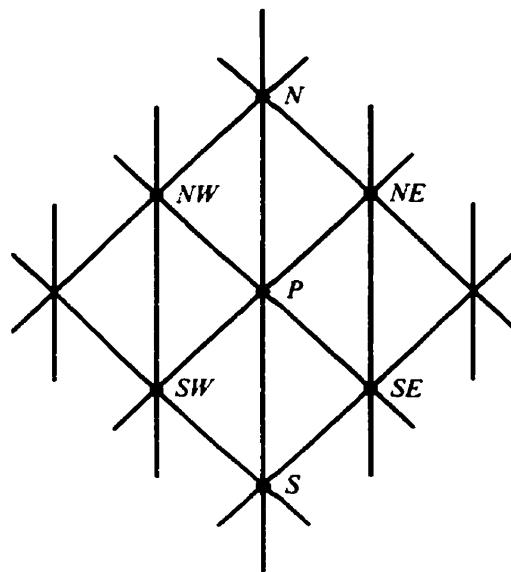


Figure 3.1: Typical two-dimensional mesh. Node P and its neighbour-nodes are illustrated.

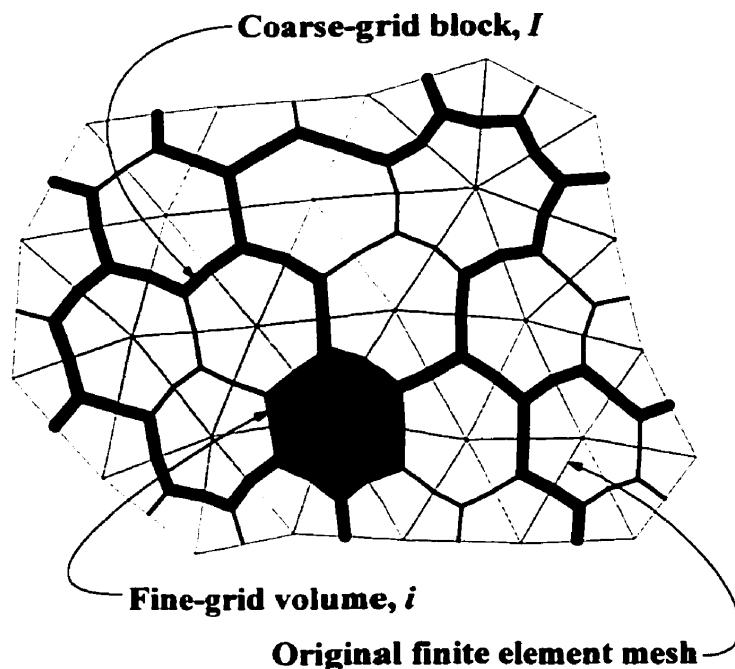


Figure 3.2: Typical agglomeration pattern for obtaining the second level in the ACM cycle: The original finite element mesh (thin lines); corresponding fine-grid volumes (medium lines); and coarse-grid volumes (thick lines) are superimposed over the same domain.

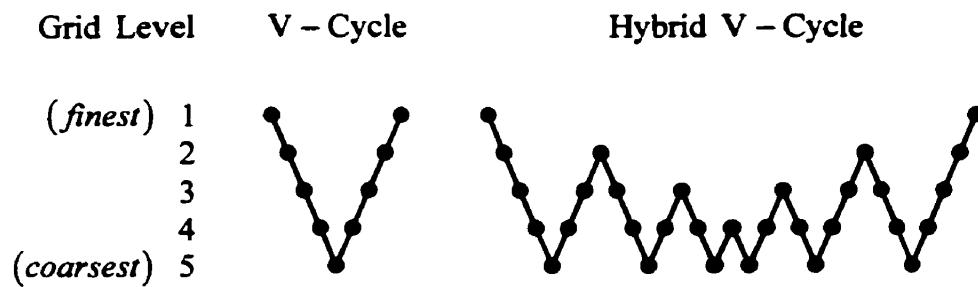


Figure 3.3: Cycling patterns used in the ACM: V-cycle; and hybrid V-cycle.

Chapter 4

***A POSTERIORI* ERROR ESTIMATION AND THE *h*-ADAPTIVE STRATEGY**

An *h*-adaptive strategy based on *a posteriori* error estimation is described in this chapter. Its ultimate purpose is to optimize the CVFEM derived in Chapter 2, with respect to accuracy and computational effort, for any particular problem. This is achieved, in the present context, by adapting the computational mesh during the solution process to meet specified mesh-optimality criteria. The iterative *h*-adaptive procedure has several distinctive steps, which can be summarized as follows:

1. **Preliminary Analysis.** A CVFEM solution is obtained on a suitably coarse mesh.
2. **Recovery.** A higher-order projection of the CVFEM solution is obtained using a *recovery* or *smoothing* process.
3. **Error Estimation.** Based on the preliminary solution and the higher-order projection, one or more error norms are used to estimate the solution error.
4. **Mesh Refinement.** The estimated error is used as a guide to refine the mesh, based on specified mesh-optimality criteria.
5. **Data Transfer.** The preliminary solution is transferred to the newly refined mesh by linear interpolation.

6. **Analysis.** A new CVFEM solution is obtained on the refined mesh, and the process is repeated from step 2 until the mesh-optimality criteria are satisfied.

Step 1 corresponds to a standard CVFEM fluid flow or heat transfer simulation. If complete mesh regeneration is used in the refinement step, the adaptive procedure will theoretically converge to an optimal mesh (defined by the mesh-optimality criteria), irrespective of the initial mesh-size distribution. However, if node-insertion is used to refine subsequent meshes, it is necessary to have a suitably coarse initial mesh that is capable of adequately capturing the essential features of the problem. It is also important that the adaptive scheme be designed to converge monotonically towards a final adapted mesh. Overshooting the optimal mesh density should be avoided (or at least minimized), especially if node-insertion is used and de-refinement is not possible. While complete mesh regeneration does not suffer from this limitation, node-insertion is usually more economical and is commonly employed for steady-state problems [56]. The refinement technique adopted for problems in this thesis is based on node-insertion [40], which has proven to be sufficient within the scope of this work. Special care is taken to achieve monotonic refinement towards a converged mesh, as described later in Section 4.3.3.

The remainder of this chapter will elaborate on the recovery, error estimation, and refinement steps in the adaptive procedure. Verification of the recovery schemes and error estimators are presented in Section 6.3.

4.1 Recovery

In Section 4.2, *a posteriori* error estimators are derived for various classes of problems. All of these error estimators involve expressions for the true error in the *gradients* of the dependent variables. Evaluation of the true error of a CVFEM simulation would require the exact solution to the associated continuum equations. Since this is rarely available, a higher-order projection of the CVFEM gradients is used in place of the exact values to approximate the true error. This projection process is termed *smoothing* or *recovery* [1-2,19].

A desirable goal in the recovery process is to obtain *superconvergent* values of the nodal gradients. Superconvergence refers to the recovered *gradients* having a convergence rate of $O(h^{p+\alpha})$ with $\alpha \geq 1$, while the original CVFEM gradients display the standard $O(h^p)$ convergence. In the remainder of this section, it will be assumed that $p=1$, which corresponds to a second-order accurate CVFEM formulation with piecewise-linear distributions of the dependent variables over each three-node triangular element. In general, the CVFEM presented in Chapter 2 is not strictly second-order accurate since other interpolation schemes (such as the FLO scheme outlined in Section 2.4.4.1, and the momentum interpolation scheme presented in Section 2.5.6) are used, besides linear interpolation. However, for low to moderate Peclet or Reynolds numbers, the CVFEM does indeed provide essentially $O(h^2)$ rates of convergence for the dependent variables, as shown in Chapter 6. It may, therefore, be argued that using this approach to derive the recovery scheme is justified for the current formulation and should still lead to a convergent adaptive procedure. This conjecture is verified in Chapter 6.

It should be noted that linear interpolation of the dependent variables over each element implies that the CVFEM gradients will be piecewise-*constant* over each element. Thus, the task in the present recovery process is to obtain piecewise-*linear* distributions of the recovered *gradients* over each element, while also achieving superconvergence of the recovered values at the nodes.

Two different recovery schemes are presented in this section: (1) the global L_2 projection of Hinton and Campbell [19]; and (2) a simple volume-weighted averaging process.

4.1.1 Global L_2 Projection

The global L_2 projection, as originally proposed by Hinton and Campbell [19], involves interpolating the recovered gradients over each element using the same basis functions as the dependent variables themselves. The nodal values of the recovered gradients are determined by performing a global least-squares-type fit to the original lower-order basis functions of the CVFEM gradients. The following is a detailed derivation of this procedure:

Over a typical element j , the interpolation function for the recovered gradients, \tilde{G}' , can be expressed in terms of the three unknown nodal values \tilde{G}_1' , \tilde{G}_2' , and \tilde{G}_3' as follows:

$$\tilde{G}' = \sum_{k=1}^3 N'_k \tilde{G}'_k, \quad (4.1)$$

where N'_k are the standard linear shape functions given by Eq. (2.14). Let \tilde{G} be the piecewise-linear function representing the set of all \tilde{G}' for $j=1, 2, \dots, N_e$, where N_e is the total number of elements in the domain. Also let \check{G} be the piecewise-constant function representing the set of all \check{G}' for $j=1, 2, \dots, N_e$ where \check{G}' are the constant CVFEM gradient distributions over each element j . The recovery problem thus becomes one of finding the nodal values \tilde{G}_i , which minimize the following functional,

$$\Lambda = \int_{\Omega} (\tilde{G} - \check{G})^2 d\Omega. \quad (4.2)$$

Hence, for Λ to be a minimum,

$$\frac{\partial \Lambda}{\partial \tilde{G}_i} = 0, \quad (4.3)$$

for each $i=1, 2, \dots, N_n$, where N_n is the total number of *nodes* in the domain. It should be noted that the sets, $\{\tilde{G}_i, i=1, 2, \dots, N_n\}$, and $\{\tilde{G}'_k, j=1, 2, \dots, N_e, k=1, 2, 3\}$, represent the same set of unknown nodal gradients, except that the former set uses node-based referencing, and the latter set uses element-based referencing (i.e. the k^{th} node of the j^{th} element).

Equation (4.3) represents a system of N_n simultaneous equations. Evaluation of the i^{th} equation yields,

$$\frac{\partial \Lambda}{\partial \tilde{G}_i} = 2 \int_{\Omega} (\tilde{G} - \check{G}) \frac{\partial \tilde{G}}{\partial \tilde{G}_i} d\Omega = 0. \quad (4.4)$$

The integrand in this expression is non-zero only over the patch of elements immediately surrounding node i . Thus, Eq. (4.4) may be rewritten as,

$$\frac{\partial \Lambda}{\partial \tilde{G}_i} = \sum_s \int_{\Omega} \left(\sum_{k=1}^3 N_k^s \tilde{G}_k^s - \tilde{G}^s \right) \frac{\partial \tilde{G}^s}{\partial \tilde{G}_i} d\Omega = 0, \quad (4.5)$$

where the index “*s*” refers to the elements *surrounding* node *i*. Referring to Fig. 4.1, the contribution, I_{123} , from element 123, to the equation for node *i* can be written as

$$I_{123} = \int_{\Omega_{123}} \left[(N_1 \tilde{G}_1 + N_2 \tilde{G}_2 + N_3 \tilde{G}_3 - \tilde{G}) N_i \right]_{123} d\Omega \quad (4.6)$$

This contribution corresponds to one of the terms in the summation over *s* in Eq. (4.5). The integrand in Eq. (4.6) contains second-order polynomial expressions in *x* and *y*, which can be integrated exactly over element 123 using the following area integration formula for triangles [137]:

$$\int_{\Omega} x^n y^m d\Omega = K_{n+m} \Omega (x_1^n y_1^m + x_2^n y_2^m + x_3^n y_3^m), \quad (4.7)$$

where Ω is the area of the triangle and the constant, K_{n+m} , is given by,

| $n + m$ | 1 | 2 | 3 | 4 | 5 |
|-----------|---|----------------|----------------|----------------|-----------------|
| K_{n+m} | 0 | $\frac{1}{12}$ | $\frac{1}{30}$ | $\frac{1}{30}$ | $\frac{2}{105}$ |

Upon evaluating the integral in Eq. (4.6), I_{123} can be recast into the following form:

$$I_{123} = C_1 \tilde{G}_1 + C_2 \tilde{G}_2 + C_3 \tilde{G}_3 + D, \quad (4.8)$$

After assembling similar contributions from the other elements surrounding node *i*, Eq. (4.5) may be rewritten in the following compact form:

$$a_i \tilde{G}_i = \sum_{nb} a_{nb} \tilde{G}_{nb} + b_i, \quad (4.9)$$

where the index “*nb*” refers to the nodes immediately *neighbouring* node *i*. The nodal values of the recovered gradients \tilde{G}_i are obtained by solving the system of simultaneous equations represented by Eq. (4.9).

The global L_2 projection is a robust and effective recovery scheme for essentially-second-order-accurate numerical methods using three-node triangular elements. This claim is verified in Chapter 6, where the scheme is shown to achieve superconvergence of the recovered gradients for low to moderate Reynolds-number flow simulations. Zienkiewicz and Zhu [2] have compared several different recovery techniques and have also found the global L_2 projection to yield superconvergent results for linear elements. In the case of quadratic elements, however, the global L_2 projection is unable to achieve superconvergent behavior. For this reason, Zienkiewicz and Zhu [2] have proposed a *local* superconvergent patch recovery technique to overcome this deficiency. A brief review of this technique is provided in Section 1.3.4.2. Since the present CVFEM formulation uses three-node triangular elements only, and since the *local* patch recovery scheme [2] suffers from other potential complications, such as possible rank deficiency of associated coefficient matrices [25], the global L_2 projection was chosen as the primary recovery scheme for the present adaptive procedure. A second recovery scheme was also implemented and is presented in the next section.

4.1.2 Volume-Averaging

A simple volume-weighted averaging procedure for obtaining nodal values of the recovered gradients is presented in this section. Its main advantages are its simplicity and efficiency. For each node in the domain, a local volume-weighted averaging of the piecewise-constant CVFEM gradients is performed over the patch of elements immediately surrounding that node. Thus, for a typical node i , the recovered gradient \tilde{G}_i is obtained from,

$$\tilde{G}_i = \frac{\sum_s \check{G}^s \Omega_s}{\sum_s \Omega_s}, \quad (4.10)$$

where the summations in this expression are carried out over the adjacent elements, s , immediately *surrounding* node i . The piecewise-constant CVFEM gradients are denoted by \check{G}^s , and the corresponding element volumes (area times unit thickness) are given by Ω_s .

This volume-averaging scheme, like the global L_2 projection, is also able to produce superconvergent nodal values of the recovered gradients, as will be shown in Chapter 6. In the present adaptive procedure, however, the volume-averaged gradients are only used as an initial guess in the iterative solution of the global L_2 projection. Wu *et al.* [18] found that simple volume-averaging alone was sufficient to drive their adaptive procedure, however, most h -adaptive finite element codes using a Zienkiewicz and Zhu-type error estimator [1-3], employ more sophisticated projection schemes for recovery purposes [2], such as the global L_2 projection.

4.2 Error Estimation

The ultimate goal of a CVFEM analysis is to obtain detailed field information about a real physical process by performing a computer simulation. The extent to which this simulation reflects reality depends on: (1) how well the mathematical model describes the physical process; and (2) how well the numerical solution approximates the exact solution to the associated mathematical model. The remainder of this section will deal with latter aspect. The task, therefore, is to determine how close the discrete CVFEM solution is to the analogous continuum solution of the partial differential equations under consideration. The absolute truncation error of a CVFEM solution is defined as the arithmetic difference between the exact solution to the continuum equations, and the solution yielded by the discrete CVFEM approximation. Since the exact solution is unavailable in most cases, the truncation error must be *estimated* in order to assess the accuracy of the CVFEM solution. The goal in *a posteriori* error estimation is to use the computed solution itself to somehow approximate its truncation error.

In this section, error estimators will be derived for various classes of problems. They will be used as a basis for adapting the grid during the solution process. A convenient way to assess the accuracy of a CVFEM solution, both globally and locally, is via the use of an *error norm*. In principle, any error norm, which depends on the truncation error, can be used in the adaptive procedure provided it vanishes as the total number of elements in the CVFEM approximation approaches infinity. A desirable error estimator, however, should be rooted in the physics of the problem and have a sound mathematical foundation.

4.2.1 Error Estimator for Scalar Diffusion

In this section, an error estimator for scalar diffusion is derived. This estimator was originally used for mesh-adaptive purposes in the context of linear elasticity problems [1-3], but has since been applied to heat conduction [141], electrostatics [25], acoustics [142], and advection-diffusion problems [26].

Steady-state diffusion processes that obey a linear gradient law, such as Fourier's law for heat conduction, or Fick's law for mass diffusion, are described by the following partial differential equation:

$$\bar{\nabla} \cdot \bar{\mathbf{q}}'' = S_\phi, \quad (4.11)$$

where, $\bar{\mathbf{q}}'' = -\Gamma_\phi \bar{\nabla} \phi.$

In this equation, $\bar{\mathbf{q}}''$ is the diffusion flux associated with the scalar variable ϕ ; Γ_ϕ is the corresponding diffusion coefficient; and S_ϕ is the appropriate volumetric generation rate or source term. Equation (4.3) is defined on a closed, bounded region, $\Omega \subset \mathbb{R}^2$, and is subject to the following boundary conditions:

$$\begin{aligned} \phi &= \phi_o && \text{on } \partial\Omega_s, \\ \bar{\mathbf{q}}'' \cdot \bar{\mathbf{n}} &= q''_o && \text{on } \partial\Omega_q, \\ \bar{\mathbf{q}}'' \cdot \bar{\mathbf{n}} &= \hbar(\phi - \phi_\infty) && \text{on } \partial\Omega_c, \end{aligned} \quad (4.12)$$

where $\bar{\mathbf{n}}$ is the unit vector pointed outward (with respect to Ω) and normal to the boundary $\partial\Omega$, and ϕ_o , q''_o , \hbar , and ϕ_∞ are specified values. The boundary needs to such that,

$$\begin{aligned} \partial\Omega &= \partial\Omega_s \cup \partial\Omega_q \cup \partial\Omega_c, \\ \partial\Omega_s \cap \partial\Omega_q &= \partial\Omega_q \cap \partial\Omega_c = \partial\Omega_s \cap \partial\Omega_c = \emptyset. \end{aligned} \quad (4.13)$$

A CVFEM approximation for scalar diffusion may be obtained from the general advection-diffusion formulation, presented in Chapter 2, by setting $\bar{\mathbf{V}} = 0$. The absolute truncation errors in the transported scalar and diffusion flux are,

$$\begin{aligned} e_\phi &= \phi - \phi^h, \\ \text{and } \bar{\mathbf{e}}_{\bar{\mathbf{q}}''} &= \bar{\mathbf{q}}'' - (\bar{\mathbf{q}}'')^h, \end{aligned} \quad (4.14)$$

respectively, where the values with superscript “ h ” denote the discrete computed solution. More precisely, ϕ^h represents the piecewise-linear element-based profiles satisfying the CVFEM approximation, and $(\bar{\mathbf{q}}'')^h$ represents the corresponding piecewise-constant profiles obtained by differentiating ϕ^h appropriately.

With respect to the Cartesian coordinate system (x,y) , the diffusion flux $\bar{\mathbf{q}}''$ may be rewritten as

$$\bar{\mathbf{q}}'' = -\Gamma_\phi \left(\frac{\partial \phi}{\partial x} \hat{\mathbf{i}} + \frac{\partial \phi}{\partial y} \hat{\mathbf{j}} \right). \quad (4.15)$$

A useful *energy norm* for scalar diffusion can be defined in the following manner [1,26,141]:

$$\|\bar{\mathbf{q}}''\| \equiv \left\{ \int_{\Omega} |\bar{\mathbf{q}}''|^2 d\Omega \right\}^{1/2} = \left\{ \int_{\Omega} \Gamma_\phi^2 \left[\left(\frac{\partial \phi}{\partial x} \right)^2 + \left(\frac{\partial \phi}{\partial y} \right)^2 \right] d\Omega \right\}^{1/2}, \quad (4.16)$$

with a corresponding *error norm* expressed as,

$$\|\bar{\mathbf{e}}_{\bar{\mathbf{q}}''}\| = \left\{ \int_{\Omega} |\bar{\mathbf{e}}_{\bar{\mathbf{q}}''}|^2 d\Omega \right\}^{1/2} = \left\{ \int_{\Omega} \Gamma_\phi^2 \left[\left(\frac{\partial \phi}{\partial x} - \frac{\partial \phi^h}{\partial x} \right)^2 + \left(\frac{\partial \phi}{\partial y} - \frac{\partial \phi^h}{\partial y} \right)^2 \right] d\Omega \right\}^{1/2}. \quad (4.17)$$

In these expressions, Γ_ϕ is assumed to be a known function of x and y . The error norm given by Eq. (4.17) can be interpreted as the energy norm of the *error* in the diffusion flux. To determine a *global* value for the error norm, the integral in Eq. (4.17) is evaluated over the entire domain, Ω . A *local* value for the error norm can be obtained by evaluating this integral over an individual element. A practical representation of the global error in the CVFEM solution is obtained by normalizing the error norm with respect to the energy norm as follows:

$$\eta_{\bar{\mathbf{q}}''} = \frac{\|\bar{\mathbf{e}}_{\bar{\mathbf{q}}''}\|}{\|\bar{\mathbf{q}}''\|}. \quad (4.18)$$

Huang and Usmani [141] have demonstrated the physical significance of $\eta_{\bar{q}''}$ by considering the diffusion problem given by Eq. (4.11) with the special case of homogeneous boundary conditions. Their derivation yielded the following final result:

$$\eta_{\bar{q}''}^2 = \frac{\|\bar{e}_{\bar{q}''}\|^2}{\|\bar{q}''\|^2} = \left| \frac{(\phi - \phi^h)_p}{\phi_p} \right|. \quad (4.19)$$

In this expression, the subscript “ p ” denotes an appropriate point in Ω , alluded to by the mean-value theorem, which Huang and Usmani invoked in their derivation [141]. Thus, Eq. (4.19) states that $\eta_{\bar{q}''}^2$ represents an average relative error in the scalar field. This desirable result provides justification for using the error norm in Eq. (4.17) as an error estimator for scalar diffusion problems.

Mathematical justification for using this estimator in the simulation of *advection-diffusion* problems may not exist. However, since the error norm is based on the diffusion flux, it has proven very effective at detecting regions with high gradients. It may be argued, therefore, that it is valid to use this error norm as a *gradient indicator* in advection-diffusion problems, with the understanding that it no longer provides for a strict assessment of the truncation error as such.

For the purposes of evaluating the energy norm, $\|\bar{q}''\|$, during the actual adaptive process, the exact gradients in Eq. (4.16) are replaced by the corresponding CVFEM values yielding:

$$\|\bar{q}''\| \approx \left\{ \int_{\Omega} |(\bar{q}'')^h|^2 d\Omega \right\}^{1/2} = \left\{ \int_{\Omega} \Gamma^2 \left[\left(\frac{\partial \phi^h}{\partial x} \right)^2 + \left(\frac{\partial \phi^h}{\partial y} \right)^2 \right] d\Omega \right\}^{1/2}. \quad (4.20)$$

For a typical element j , the linear interpolation function, ϕ_j^h , can be written as,

$$\phi_j^h = A_j^h x + B_j^h y + C_j^h. \quad (4.21)$$

Its derivatives, therefore, are:

$$\frac{\partial \phi_j^h}{\partial x} = A_j^h; \quad \frac{\partial \phi_j^h}{\partial y} = B_j^h. \quad (4.22)$$

Substituting these values back into Eq. (4.20), $\|\bar{\mathbf{q}}''\|$ may be evaluated as the sum of its respective element contributions, $\|\bar{\mathbf{q}}''\|_j$, as follows:

$$\|\bar{\mathbf{q}}''\| = \left\{ \sum_{j=1}^{N_e} \|\bar{\mathbf{q}}''\|_j^2 \right\}^{1/2} \approx \left\{ \sum_{j=1}^{N_e} \int_{\Omega_j} (\Gamma_s)^2 [(\mathbf{A}_j^h)^2 + (\mathbf{B}_j^h)^2] d\Omega \right\}^{1/2}, \quad (4.23)$$

where N_e is the total number of elements in the mesh.

To evaluate the error norm, $\|\bar{\mathbf{e}}_{\bar{\mathbf{q}}''}\|$, the exact gradients are replaced by the *recovered* gradients, as described in Section 4.1. The expression for $\|\bar{\mathbf{e}}_{\bar{\mathbf{q}}''}\|$, in Eq. (4.17), is therefore approximated by,

$$\|\bar{\mathbf{e}}_{\bar{\mathbf{q}}''}\| \approx \left\{ \int_{\Omega} \Gamma_s^2 \left[\left(\frac{\partial \tilde{\phi}}{\partial x} - \frac{\partial \phi^h}{\partial x} \right)^2 + \left(\frac{\partial \tilde{\phi}}{\partial y} - \frac{\partial \phi^h}{\partial y} \right)^2 \right] d\Omega \right\}^{1/2}, \quad (4.24)$$

where the “~” refers to the recovered values. These recovered gradients, are interpolated linearly over each element j , and can therefore be expressed as,

$$\frac{\partial \tilde{\phi}}{\partial x}_j = \tilde{A}_j^x x + \tilde{B}_j^x y + \tilde{C}_j^x; \quad (4.25)$$

and,

$$\frac{\partial \tilde{\phi}}{\partial y}_j = \tilde{A}_j^y x + \tilde{B}_j^y y + \tilde{C}_j^y. \quad (4.26)$$

Upon substituting these expressions back into Eq. (4.24), and splitting the integral into the sum of its N_e element contributions, the global error norm may be rewritten as:

$$\|\bar{\mathbf{e}}_{\bar{\mathbf{q}}''}\| \approx \left\{ \sum_{j=1}^{N_e} \int_{\Omega_j} (\Gamma_s)^2 [(\tilde{A}_j^x x + \tilde{B}_j^x y + \tilde{C}_j^x - A_j^h)^2 + (\tilde{A}_j^y x + \tilde{B}_j^y y + \tilde{C}_j^y - B_j^h)^2] d\Omega \right\}^{1/2}. \quad (4.27)$$

The integrands in this expression contain second-order polynomials in x and y . Thus, the integrals can be evaluated exactly using the triangle integration formula given by Eq. (4.7).

4.2.2 Strain-Rate Error Estimator

An error norm based on strain-rates is presented in this section. It is strictly derived for creeping-flow problems [7], but has successfully been applied to incompressible Navier-Stokes flows by Wu *et al.* [18], Hetu and Pelletier [22-23], and Pelletier *et al.* [24].

The continuum equations describing two-dimensional viscous flow of an incompressible Newtonian fluid are given by Eqs. (2.1)-(2.3). The corresponding CVFEM approximations to these equations were derived in Chapter 2.

The continuum strain-rate tensor $\boldsymbol{\varepsilon}$, in two-dimensions, may be written as follows:

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{1}{2} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \\ \frac{1}{2} \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) & \frac{\partial v}{\partial y} \end{bmatrix}. \quad (4.28)$$

An invariant of this tensor can be defined as,

$$|\boldsymbol{\varepsilon}| = \left\{ \left(\frac{\partial u}{\partial x} \right)^2 + \frac{1}{2} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \right\}^{1/2}. \quad (4.29)$$

The truncation error in the strain-rate tensor is represented by,

$$\boldsymbol{\epsilon}_{\varepsilon} = \boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^h, \quad (4.30)$$

where the superscript “ h ” denotes the CVFEM solution. An *energy norm*, analogous to the one given by Eq. (4.16) for scalar diffusion, may be defined as:

$$\|\boldsymbol{\varepsilon}\| = \left\{ \int_{\Omega} |\boldsymbol{\varepsilon}|^2 d\Omega \right\}^{1/2} = \left\{ \int_{\Omega} \left[\left(\frac{\partial u}{\partial x} \right)^2 + \frac{1}{2} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \right] d\Omega \right\}^{1/2}, \quad (4.31)$$

with a corresponding *error norm*,

$$\begin{aligned}\|\mathbf{e}_\epsilon\| &= \left\{ \int_{\Omega} |\mathbf{e}_\epsilon|^2 d\Omega \right\}^{\frac{1}{2}} \\ &= \left\{ \int_{\Omega} \left[\left(\frac{\partial u}{\partial x} - \frac{\partial u^h}{\partial x} \right)^2 + \frac{1}{2} \left(\frac{\partial u}{\partial y} - \frac{\partial u^h}{\partial y} + \frac{\partial v}{\partial x} - \frac{\partial v^h}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} - \frac{\partial v^h}{\partial y} \right)^2 \right] d\Omega \right\}^{\frac{1}{2}}.\end{aligned}\quad (4.32)$$

In order to evaluate these norms numerically, the exact gradients in the expression for $\|\boldsymbol{\varepsilon}\|$ are replaced by the CVFEM gradients, and the exact gradients in the expression for $\|\mathbf{e}_\epsilon\|$ are replaced by the higher-order recovered values, as was done in the previous section for the scalar diffusion norms ($\|\bar{\mathbf{q}}''\|$ and $\|\bar{\mathbf{e}}_{\bar{\mathbf{q}}''}\|$). Furthermore, the integral over the domain Ω can be split into the sum of the integrals over each element, so that,

$$\|\boldsymbol{\varepsilon}\| = \left\{ \sum_{j=1}^{N_e} \|\boldsymbol{\varepsilon}\|_j^2 \right\}^{\frac{1}{2}}; \quad \|\mathbf{e}_\epsilon\| = \left\{ \sum_{j=1}^{N_e} \|\mathbf{e}_\epsilon\|_j^2 \right\}^{\frac{1}{2}}. \quad (4.33)$$

A practical representation of the strain-rate error norm can be defined as,

$$\eta_\epsilon = \frac{\|\mathbf{e}_\epsilon\|}{\|\boldsymbol{\varepsilon}\|}. \quad (4.34)$$

Hetu and Pelletier [22] have derived a similar estimator from the variational formulation of Stokes flow, and Ainsworth *et al.* [16] have proven that this estimator is bounded and convergent for linear problems. For nonlinear problems, such as the Navier-Stokes equations, no known general theory can provide this guarantee, nevertheless, this error estimator should still prove useful for Navier-Stokes flows since it is sensitive to high strain-rates and shear-stresses by construction. Numerical simulations, which will be presented in Chapter 6, have verified this property.

4.2.3 Flow-Angle Indicator

One shortcoming of the strain-rate error norm given by Eq. (4.32) is that it may not be able to resolve certain flow features, such as stagnation points, wakes, or recirculation eddies, where the velocity magnitudes and computed gradients are typically very small.

To compensate in these regions, a *flow-angle indicator* [20-21] is implemented to work in conjunction with the strain-rate error estimator.

It is convenient, in the development of this indicator, to define a *flow-angle gradient field*, $\bar{\psi}$ as follows:

$$\bar{\psi} \equiv \bar{\nabla} \theta = \frac{\partial \theta}{\partial x} \mathbf{i} + \frac{\partial \theta}{\partial y} \mathbf{j}. \quad (4.35)$$

In this expression, θ represents the *flow-angle scalar field* defined at any point in the plane as the absolute angle swept by rotating the local velocity vector counter-clockwise to a fixed, predefined reference vector.

Using this definition, a *flow-angle norm* may be defined as,

$$\|\bar{\psi}\| \equiv \left\{ \int_{\Omega} |\bar{\psi}| d\Omega \right\}^{1/2} = \left\{ \int_{\Omega} \left[\left(\frac{\partial \theta}{\partial x} \right)^2 + \left(\frac{\partial \theta}{\partial y} \right)^2 \right] d\Omega \right\}^{1/2}, \quad (4.36)$$

and the corresponding *flow-angle error norm* as,

$$\|\bar{\epsilon}_{\bar{\psi}}\| \equiv \left\{ \int_{\Omega} |\bar{\epsilon}_{\bar{\psi}}| d\Omega \right\}^{1/2} = \left\{ \int_{\Omega} \left[\left(\frac{\partial \theta}{\partial x} - \frac{\partial \theta^h}{\partial x} \right)^2 + \left(\frac{\partial \theta}{\partial y} - \frac{\partial \theta^h}{\partial y} \right)^2 \right] d\Omega \right\}^{1/2}. \quad (4.37)$$

It is noted here that these last two norms are very similar to the scalar diffusion norms presented in Section 4.2.1. In fact, their definitions are identical if ϕ and Γ_s in Eqs. (4.16)-(4.17) are replaced by θ and 1, respectively. Thus, much of the programming infrastructure developed for the scalar diffusion error estimator can be utilized for the flow-angle indicator as well. A major difference, however, is in the calculation of the flow-angle scalar field, θ . Firstly, a piecewise-linear interpolation of θ , denoted by θ^h , is constructed over each element in the domain using information from the CVFEM solution. This is achieved, for element 123 in Fig. 4.2, by arbitrarily assigning $\theta_1^h = 0$ and computing θ_2^h and θ_3^h relative to θ_1^h . The level of θ^h is not relevant here since only its derivatives are needed to evaluate the norms in Eqs. (4.36) and (4.37). Thus, θ_2^h is computed as the angle swept by rotating the velocity vector $\bar{\mathbf{V}}_2^h$ through to $\bar{\mathbf{V}}_1^h$.

Similarly, θ_3^h is computed as the angle swept by rotating $\bar{\mathbf{V}}_3^h$ through to $\bar{\mathbf{V}}_1^h$. The sign of the angle is important. If the aforementioned rotation is counter-clockwise, the angle is taken as positive. If the rotation is clockwise, then the angle is made negative. Values of θ^h are kept in the range $-\pi \leq \theta^h \leq \pi$. The nodal values of θ^h can, therefore, be written explicitly as,

$$\begin{aligned}\theta_1^h &= 0, \\ \theta_2^h &= \begin{cases} \cos^{-1} \frac{\bar{\mathbf{V}}_1^h \cdot \bar{\mathbf{V}}_2^h}{\|\bar{\mathbf{V}}_1^h\| \|\bar{\mathbf{V}}_2^h\|} & \text{if } \bar{\mathbf{V}}_2^h \times \bar{\mathbf{V}}_1^h \cdot \bar{\mathbf{k}} > 0 \\ -\cos^{-1} \frac{\bar{\mathbf{V}}_1^h \cdot \bar{\mathbf{V}}_2^h}{\|\bar{\mathbf{V}}_1^h\| \|\bar{\mathbf{V}}_2^h\|} & \text{if } \bar{\mathbf{V}}_2^h \times \bar{\mathbf{V}}_1^h \cdot \bar{\mathbf{k}} < 0 \end{cases}, \\ \theta_3^h &= \begin{cases} \cos^{-1} \frac{\bar{\mathbf{V}}_1^h \cdot \bar{\mathbf{V}}_3^h}{\|\bar{\mathbf{V}}_1^h\| \|\bar{\mathbf{V}}_3^h\|} & \text{if } \bar{\mathbf{V}}_3^h \times \bar{\mathbf{V}}_1^h \cdot \bar{\mathbf{k}} > 0 \\ -\cos^{-1} \frac{\bar{\mathbf{V}}_1^h \cdot \bar{\mathbf{V}}_3^h}{\|\bar{\mathbf{V}}_1^h\| \|\bar{\mathbf{V}}_3^h\|} & \text{if } \bar{\mathbf{V}}_3^h \times \bar{\mathbf{V}}_1^h \cdot \bar{\mathbf{k}} < 0 \end{cases}.\end{aligned}\quad (4.38)$$

where $\bar{\mathbf{k}} = \bar{\mathbf{i}} \times \bar{\mathbf{j}}$, and $\bar{\mathbf{i}}$ and $\bar{\mathbf{j}}$ are the Cartesian unit vectors in the x - and y -coordinate directions, respectively. Using the three nodal values in Eq. (4.38), the linear interpolation function, θ^h , can be written in the form,

$$\begin{aligned}\theta^h &= N_1 \theta_1^h + N_2 \theta_2^h + N_3 \theta_3^h \\ &= A^\theta x + B^\theta y + C^\theta.\end{aligned}\quad (4.39)$$

Thus, the piecewise-constant elemental values of the CVFEM flow-angle gradients are given by,

$$\frac{\partial \theta^h}{\partial x} = A^\theta; \quad \frac{\partial \theta^h}{\partial y} = B^\theta. \quad (4.40)$$

The flow-angle norms, $\|\bar{\psi}\|$ and $\|\bar{\mathbf{e}}_{\bar{\psi}}\|$, are computed (as in the case of the diffusion norms - $\|\bar{\mathbf{q}}''\|$ and $\|\bar{\mathbf{e}}_{\bar{\mathbf{q}}''}\|$) by replacing the exact gradients in $\|\bar{\psi}\|$ with the corresponding CVFEM

values, and by replacing the exact gradients in $\|\bar{\mathbf{e}}_{\mathbf{v}}\|$ with the corresponding recovered values.

The flow angle indicator is sensitive to large streamline curvatures, or alternatively, to large gradients in the flow-angle scalar field, regardless of how small the velocity magnitude may be. This characteristic can be very useful for resolving flow features of potential significance, in regions where the magnitude of the velocity components and computed gradients may be very small, as will be illustrated in Section 6.5.

4.3 Refinement

In Section 4.1, recovery procedures were presented for obtaining a higher-order projection of the CVFEM gradients. The arithmetic difference between the higher-order projection and the original CVFEM values constitutes an *estimation* of the absolute error in the CVFEM gradients. In Section 4.2, several *error norms*, which depend on the absolute error, were derived for various classes of problems. These error norms attempt to characterize the accuracy of the CVFEM solution in both a global and local sense. In the present section, a procedure is presented for (i) processing the detailed information obtained from the error norms; and (ii) utilizing this processed information to determine how the mesh should be refined in order to meet specified mesh-optimality criteria.

The procedure outlined in this section is valid, in principle, for any energy norm and corresponding error norm that meet the basic requirements outlined in Section 4.2. Thus, the energy norm in this section will be referred to *generically* by $\|\mathbf{u}\|$ and the corresponding error norm by $\|\mathbf{e}_u\|$. Table 4.1 gives a list of the specific norms derived in Section 4.2 that may be represented by $\|\mathbf{u}\|$ and $\|\mathbf{e}_u\|$.

| | Energy Norm | Error Norm |
|----------------------------|---|--|
| Generic | $\ \mathbf{u}\ $ | $\ \mathbf{e}_u\ $ |
| Scalar Diffusion Estimator | $\ \bar{\mathbf{q}}''\ $ - Eq. (4.16) | $\ \bar{\mathbf{e}}_{\bar{\mathbf{q}}''}\ $ - Eq. (4.17) |
| Strain-Rate Estimator | $\ \boldsymbol{\varepsilon}\ $ - Eq. (4.31) | $\ \mathbf{e}_\varepsilon\ $ - Eq. (4.32) |
| Flow-Angle Indicator | $\ \bar{\psi}\ $ - Eq. (4.36) | $\ \bar{\mathbf{e}}_\psi\ $ - Eq. (4.37) |

Table 4.1: Specific norms represented by $\|\mathbf{u}\|$ and $\|\mathbf{e}_u\|$.

A mesh is said to be, “*acceptable*”, if the estimated error in the corresponding CVFEM solution satisfies some prescribed *global* and/or *local* conditions [17]. The next two sections establish global and local error criteria that are to be satisfied by the final converged mesh at the end of the h -adaptive procedure. These criteria are presented in a context which assumes that the grid-refinement step is performed using a node-insertion technique [40] (as in the present case), however they remain valid for adaptive procedures that are also capable of mesh coarsening.

4.3.1 Global Error Criterion

A practical representation of the global error in the CVFEM solution, with respect to a particular error estimator, $\|\mathbf{e}_u\|$, is given by,

$$\eta_u = \frac{\|\mathbf{e}_u\|}{\|\mathbf{u}\|}. \quad (4.41)$$

The global error criterion can therefore be established by specifying a target value of η_u , denoted by $\bar{\eta}_u$. A convenient way of monitoring whether this criterion has been met is by defining a global error parameter, ξ_g , as follows:

$$\xi_g = \frac{\eta_u}{\bar{\eta}_u} = \frac{\|\mathbf{e}_u\|}{\bar{\eta}_u \|\mathbf{u}\|}, \quad (4.42)$$

Note that the global error criterion is satisfied if $\xi_g \leq 1$, while further refinement is required if $\xi_g > 1$. Using this parameter alone would lead to uniform mesh refinement only, therefore, a local error criterion is also needed.

4.3.2 Local Error Criterion

A local error constraint can be imposed on each element in the domain by requiring that the square of the global error norm be distributed equally over all elements [1,17,26], that is,

$$\|\mathbf{e}_u\|_r = \sqrt{\frac{\|\mathbf{e}_u\|^2}{N_e}}, \quad (4.43)$$

where $\|\mathbf{e}_u\|_r$ is the *required* (or desired) error norm for each element, and N_e is the total number of elements in the domain. Using this expression, a local error parameter, ξ_j , can be defined for each element j , as,

$$\xi_j = \frac{\|\mathbf{e}_u\|_j}{\|\mathbf{e}_u\|_r} = \sqrt{N_e} \frac{\|\mathbf{e}_u\|_j}{\|\mathbf{e}_u\|}. \quad (4.44)$$

Note that the local error criterion is satisfied for element j if $\xi_j \leq 1$, and not satisfied if $\xi_j > 1$.

4.3.3 Element Refinement Parameter

The global and local error criteria can be expressed simultaneously by combining ξ_g and ξ_j into one *element refinement parameter*, β_j , which is then used to define a new element size, h'_j , in terms of the existing size, h_j , as,

$$h'_j = \frac{h_j}{\beta_j}. \quad (4.45)$$

Onate and Bugeda [17] have shown that the dependence of β_j on ξ_g and ξ_j should be consistent with the different convergence rates of the global and local error norms, in

order to avoid *oscillatory* convergence to an optimal mesh during the iterative mesh-adapting procedure. This is especially important when using a node-insertion technique to refine the mesh (and de-refinement is not possible). In such cases, overshooting the mesh density is irreversible and will incur unnecessary performance penalties.

It is evident upon examining the error norms in Eqs. (4.17), (4.32), and (4.37), that if the exact gradients are replaced by the piecewise-*linear* recovered values, the *global* error norm, $\|\mathbf{e}_u\|$, will exhibit $O(\bar{h})$ convergence where \bar{h} represents an average element size in the mesh. This is seen in the following expression [17]:

$$\|\mathbf{e}_u\| \rightarrow \left\{ \int_{\Omega} [O(h)]^2 d\Omega \right\}^{\frac{1}{2}} \rightarrow \left\{ [O(\bar{h})]^2 \right\}^{\frac{1}{2}} \rightarrow O(\bar{h}), \quad (4.46)$$

It is also evident that the *local* error norm, $\|\mathbf{e}_u\|_j$, over a typical element j , will exhibit $O(h_j^2)$ convergence as shown in the following:

$$\|\mathbf{e}_u\|_j \rightarrow \left\{ \int_{\Omega_j} [O(h_j)]^2 d\Omega \right\}^{\frac{1}{2}} \rightarrow \left\{ [O(h_j)]^2 \Omega_j \right\}^{\frac{1}{2}} \rightarrow O(h_j) \sqrt{\Omega_j} \rightarrow O(h_j^2), \quad (4.47)$$

These observations suggest the following definition of β_j [17]:

$$\beta_j = \xi_s \sqrt{\xi_j}. \quad (4.48)$$

The desired goal in defining β_j this way is to obtain *monotonic* convergence towards a final mesh by controlling the refinement process so that it is consistent with the different convergence rates of the global and local error norms.

Finally, it should be noted that the error norm, $\|\mathbf{e}_u\|$, is highly sensitive to discontinuities. In fact, the adaptive procedure will tend to refine the mesh indefinitely in the vicinity of a discontinuity. To avoid excessively dense meshes in these regions, a *minimum area constraint* can be placed on each element in the domain. This will allow the adaptive procedure to converge to a final mesh despite the presence of any discontinuities.

4.3.4 Multiple Error Estimators

In some problems, it may be advantageous to adapt on more than one variable, or to combine more than one error estimator or indicator in the adaptive procedure. One example is a coupled fluid flow and heat transfer problem, which may involve large velocity and temperature gradients in different regions of the domain. For such a problem, the scalar diffusion error estimator (4.17) could be used to adapt on the temperature field, while the strain-rate error estimator (4.32) could be used to adapt on the velocity field. Another combination that has been used extensively in the present thesis (see Chapter 6) is the strain-rate error estimator (4.32) in conjunction with the flow-angle indicator (4.37) for Navier-Stokes flows. The main difficulty when using this approach is to determine appropriate values of the target error, $\bar{\eta}_u$, for each estimator. In the present work, this is done heuristically, by performing a series of numerical experiments. For example, several different modest values of $\bar{\eta}_u$ can be prescribed for each error estimator, in a series of short simulations, until suitable values are found relative to each other. Then these values are appropriately reduced by the same factor for subsequent analyses.

Thus, in the present refinement scheme, if N_{est} is the total number of error estimates being considered, then N_{est} different target errors, $\bar{\eta}_u^k$, $k = 1, 2, \dots, N_{est}$, are specified. Each individual error estimate is performed in the usual way, as outlined in Sections 4.3.1-4.3.3. This will yield N_{est} different element refinement parameters, β_j^k , $k = 1, 2, \dots, N_{est}$, and N_{est} proposed new element sizes, $h_j'^k$, $k = 1, 2, \dots, N_{est}$, for each element j . In order to satisfy all of the element refinement criteria simultaneously with just *one* new element size, h_j' , we take,

$$h_j' = \min_k(h_j'^k) \quad (4.49)$$

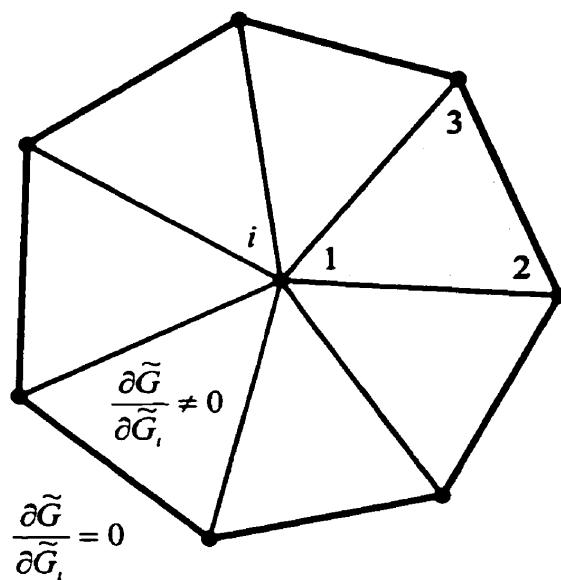


Figure 4.1: The patch of elements surrounding node i , and the local node numbering used in the derivation of the global L_2 projection.

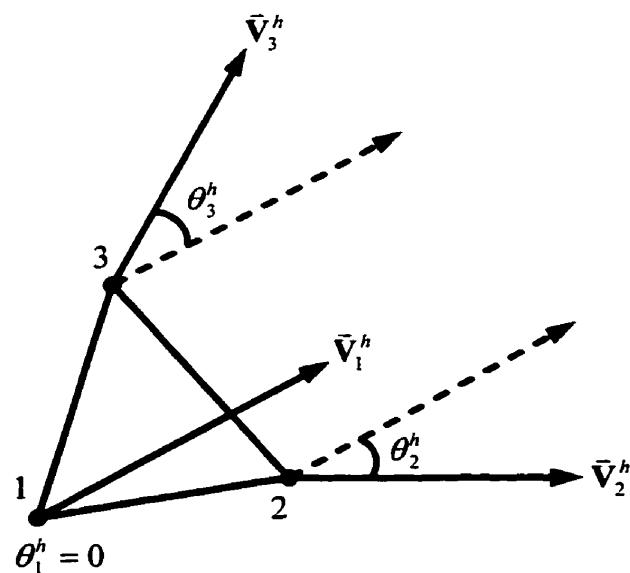


Figure 4.2: A typical element and related nomenclature used in the derivation of the flow-angle indicator.

Chapter 5

EXTENDED RICHARDSON EXTRAPOLATION FOR UNSTRUCTURED GRIDS

An extrapolation technique for *unstructured grids* is proposed in this chapter. The main purpose of this technique is to estimate grid-independent results by performing an extrapolation of results from two or more computational grids with different element densities. The corresponding discrete solutions obtained on these grids are used to extrapolate the analogous continuum solution, which, for a consistent numerical method, is obtained in the limit as the element density approaches infinity. The proposed extrapolation procedure is based on key concepts first introduced by Richardson in 1910 [51]. Sections 5.1 - 5.2 lay out the foundation for the original extrapolation procedure, as Richardson documented in 1927 [52]. Sections 5.3 – 5.4 present generalizations of this technique, and in Section 5.5, an extended Richardson extrapolation procedure is proposed for unstructured grids.

5.1 The Deferred Approach to the Limit

Richardson extrapolation was originally presented as a particular application of a more general concept, which Richardson referred to as, “the deferred approach to the limit” [52]. This statement refers to *postponing* the limiting process (in the definition of the derivative) that is implicitly invoked when performing a mathematical analysis using differential calculus. Richardson suggested that infinitesimal calculus may be cumbrous or unwieldy when applied to certain problems, and that in these cases, “it may be well to

return to the manner in which they did things before the calculus was invented". He was referring to "postponing the passage to the limit until after the problem had been solved for a moderate number of moderately small (finite) differences". In the context of simulating physical problems involving differential equations using the finite difference method, Richardson proposed a particular application of the deferred approach to the limit, which he termed h^2 extrapolation, and which is now known as Richardson extrapolation. In this procedure, two or more discrete solutions are obtained on uniform grids with different mesh spacing. Then, an extrapolation ("deferred approach to the limit") is performed using these discrete solutions to approximate the exact solution to the analogous continuum problem.

5.2 h^2 Extrapolation

In the original Richardson extrapolation method [52], discrete solutions, ϕ^h , obtained on uniformly spaced, ordered grids, are assumed to have a series representation in the grid spacing, h , as follows:

$$\phi^h(x, h) = \phi(x) + f_1(x)h + f_2(x)h^2 + f_3(x)h^3 + \dots, \quad (5.1)$$

where ϕ is the solution to the analogous continuum equation under consideration. If ϕ^h is assumed to be an infinitely differentiable function of h , the functions $f_1(x), f_2(x), f_3(x), \dots$, can be expressed in terms of the derivatives of ϕ^h through the following Taylor series expansion:

$$\phi^h(x, h) = \phi^h\Big|_{h=0} + \frac{\partial \phi^h}{\partial h}\Bigg|_{h=0} h + \frac{\partial^2 \phi^h}{\partial h^2}\Bigg|_{h=0} \frac{h^2}{2!} + \frac{\partial^3 \phi^h}{\partial h^3}\Bigg|_{h=0} \frac{h^3}{3!} + \dots, \quad (5.2)$$

where, for a consistent numerical method,

$$\phi^h\Big|_{h=0} = \lim_{h \rightarrow 0} \phi^h(x, h) = \phi(x) \quad (5.3)$$

It is noted that the existence of the Taylor series expression given by Eq. (5.2) is not a necessary assumption for Richardson extrapolation, however, it serves to motivate Eq. (5.1).

A second-order method is one in which $f_1(x) = 0$ identically. Furthermore, in the special case of a finite difference method using central-differencing, Richardson [52] found that all the coefficients of odd powers of h are absent from Eq. (5.1). This is illustrated here by deriving the central-difference approximation of the derivative: Firstly, an infinitely differentiable function $g(x)$ is expanded in h and $-h$ about x as,

$$g(x+h) = g(x) + \frac{dg}{dx}\Big|_x h + \frac{d^2g}{dx^2}\Big|_x \frac{h^2}{2!} + \frac{d^3g}{dx^3}\Big|_x \frac{h^3}{3!} + \frac{d^4g}{dx^4}\Big|_x \frac{h^4}{4!} + \frac{d^5g}{dx^5}\Big|_x \frac{h^5}{5!} + \dots, \quad (5.4)$$

and,

$$g(x-h) = g(x) - \frac{dg}{dx}\Big|_x h + \frac{d^2g}{dx^2}\Big|_x \frac{h^2}{2!} - \frac{d^3g}{dx^3}\Big|_x \frac{h^3}{3!} + \frac{d^4g}{dx^4}\Big|_x \frac{h^4}{4!} - \frac{d^5g}{dx^5}\Big|_x \frac{h^5}{5!} + \dots, \quad (5.5)$$

respectively. Subtracting Eq. (5.5) from Eq. (5.4) and solving for dg/dx yields:

$$\frac{dg}{dx}\Big|_x = \frac{g(x+h) - g(x-h)}{2h} - \frac{d^3g}{dx^3}\Big|_x \frac{h^2}{3!} - \frac{d^5g}{dx^5}\Big|_x \frac{h^4}{5!} - \dots \quad (5.6)$$

The central-difference approximation, $(dg/dx)^h$, at x is, therefore, defined as,

$$\left(\frac{dg}{dx}\right)^h = \frac{g(x+h) - g(x-h)}{2h}. \quad (5.7)$$

Setting $\frac{dg}{dx}\Big|_x = \phi(x)$; $\left(\frac{dg}{dx}\right)^h = \phi^h(x, h)$; and $f_n(x) = \frac{d^{n+1}g}{dx^{n+1}}\Big|_x \frac{1}{(n+1)!}$, $n = 2, 4, 6 \dots$,

and rearranging Eq. (5.6) gives,

$$\phi^h(x, h) = \phi(x) + f_2(x)h^2 + f_4(x)h^4 + \dots, \quad (5.8)$$

which, upon comparison with Eq. (5.1), demonstrates the absence of odd powers of h in the case of central-differencing. Richardson performed numerous “arithmetical” examples, which also confirmed the absence of these terms [52].

The “ h^2 extrapolation” process involves truncating Eq. (5.8) after the h^2 term; using two discrete solutions obtained on grids with unequal step sizes, h_1 and h_2 ; and eliminating $f_2(x)$ to arrive at an approximation for $\phi(x)$ in the form,

$$\phi(x) \approx \frac{\phi^h(x, h_1) h_1^2 - \phi^h(x, h_2) h_2^2}{h_2^2 - h_1^2}. \quad (5.9)$$

The value of $\phi(x)$ given by Eq. (5.9) is fourth-order accurate if $\phi^h(x, h_1)$ and $\phi^h(x, h_2)$ are obtained using a second-order accurate numerical scheme which *exclusively* employs central-differencing (as illustrated by Eq. (5.8)). In general, if uncentered second-order differences are used (for example, second order upwind schemes), then the coefficient multiplying h^3 in the expansion given by Eq. (5.1) may not be zero. In these cases, Eq. (5.9) is only third-order accurate. It should be noted that the anticipated fourth-order behavior of Eq. (5.9), in the case of central-differencing, is only realized in practice when h_1 and h_2 are chosen sufficiently small that they are within the *asymptotic region* of convergence with respect to grid spacing. The asymptotic region, for any particular problem, is the range of h for which the truncated terms in the expansion given by Eq. (5.8) (that is, $f_4(x)h^4$, $f_6(x)h^6$, ...) are negligible relative to $f_2(x)h^2$. This also implies *monotonic* truncation-error convergence with respect to h .

Richardson [52] also proposed a sixth-order extrapolation procedure for central-difference schemes in which Eq. (5.8) is truncated after the h^4 term. Three discrete solutions are required in this case to eliminate $f_2(x)$ and $f_4(x)$, and arrive at an approximation for $\phi(x)$.

It should also be noted that Richardson extrapolation applies not only to the nodal values of ϕ^h , but also to other local and global quantities computed from ϕ^h in post-processing. For example, if ϕ^h represents the computed velocity components in a primitive variable fluid flow formulation, Richardson extrapolation can be applied to local quantities such as the computed vorticity field, shear stress distribution, and stagnation point location(s), or to global quantities such as the total drag force and energy dissipation rate. The higher-order accuracy obtained by extrapolating nodal values of ϕ^h can also be expected from extrapolating the post-processed quantities, provided that consistent or higher-order methods are used in the evaluation of these post-processed quantities.

Although Richardson extrapolation was originally derived for central-difference schemes, it can also be applied to FVMs, FEMs, and CVFEMs, provided that Eq. (5.1) is indicative

of the order of the discretization (for example, $f_1 = 0$ for a second-order scheme; $f_1 = f_2 = 0$ for a third-order scheme; etc...). Numerous examples of its usage exist in the literature [36,53,135,143-148]. A significant deficiency of Richardson extrapolation, however, is that the extrapolated solution, in general, is not conservative in the sense of maintaining conservation properties [53]. In the context of finite difference methods, the extrapolated result may no longer satisfy a system of finite difference approximations. An additional disadvantage noted by Richardson [52] is that the accuracy of the extrapolation, in practice, does not necessarily apply to solution derivatives of arbitrary order (except in the limit as $h \rightarrow 0$). The extrapolation can introduce noise into the solution, which may decrease the accuracy of the higher-order solution derivatives.

5.3 Generalized Richardson Extrapolation

It was already suggested in the previous section that the extrapolation formula given by Eq. (5.9) could also be used for a *non-centered* second-order accurate difference scheme with the understanding that the extrapolated solution would, in general, be third-order accurate and *not* forth-order accurate. It follows, therefore, that a generalized extrapolation procedure can be devised for an arbitrary, consistent numerical method of order m in the following manner:

The discrete solution obtained from a truly m -order accurate method would display the following behavior,

$$\phi^h(x, h) = \phi(x) + f_m(x) h^m + f_{m+1}(x) h^{m+1} + \dots \quad (5.10)$$

An approximation for $\phi(x)$, which is at least $(m+1)$ -order accurate, can be obtained using the same extrapolation procedure as outlined in the previous subsection. Namely: truncate Eq. (5.10) after the h^m term; obtain two discrete solutions on two uniform grids with unequal step sizes h_1 and h_2 ; and eliminate $f_m(x)$ to arrive at,

$$\phi(x) \approx \frac{\phi^h(x, h_1) h_2^m - \phi^h(x, h_2) h_1^m}{h_2^m - h_1^m}. \quad (5.11)$$

An extrapolation to $\phi(x)$ from n different discrete solutions can also be conceived for an m -order accurate numerical method. For example, Eq. (5.10) can be truncated after the h^{m+n-2} term yielding,

$$\phi^h(x, h) = \phi(x) + f_m(x)h^m + f_{m+1}(x)h^{m+1} + \cdots + f_{m+n-2}(x)h^{m+n-2}. \quad (5.12)$$

If we let h_i , $i=1,2,\dots,n$, be the n different step sizes corresponding to the n different uniform grids upon which the n different discrete solutions were obtained, then upon substituting each of these solutions into Eq. (5.12), the following system of equations would result:

$$\begin{Bmatrix} \phi^h(x, h_1) \\ \phi^h(x, h_2) \\ \phi^h(x, h_3) \\ \vdots \\ \phi^h(x, h_n) \end{Bmatrix} = \begin{bmatrix} 1 & h_1^m & h_1^{m+1} & \cdots & h_1^{m+n-2} \\ 1 & h_2^m & h_2^{m+1} & \cdots & h_2^{m+n-2} \\ 1 & h_3^m & h_3^{m+1} & \cdots & h_3^{m+n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & h_n^m & h_n^{m+1} & \cdots & h_n^{m+n-2} \end{bmatrix} \begin{Bmatrix} \phi(x) \\ f_m(x) \\ f_{m+1}(x) \\ \vdots \\ f_{m+n-2}(x) \end{Bmatrix}, \quad (5.13)$$

where $\phi(x)$ is readily obtainable by using Cramer's rule [140], for example.

5.4 Unequal Grid Spacing and Non-Uniform Grids

Consider the case where numerical simulations are being performed on structured, two-dimensional Cartesian grids with uniform, but unequal grid spacing h_x and h_y in the x - and y -coordinate directions, respectively. Also consider the case where h_x and h_y may be specified *independent* of each other for any given simulation. Without knowledge of the order of the numerical method, if it is assumed that the discrete solution, ϕ^h , is a continuous and infinitely differentiable function of both h_x and h_y , then the following double Taylor series expansion is valid:

$$\begin{aligned} \phi^h(x, y, h_x, h_y) = & \phi^h\Big|_0 + \frac{\partial \phi^h}{\partial h_x}\Big|_0 h_x + \frac{\partial \phi^h}{\partial h_y}\Big|_0 h_y + \\ & \frac{\partial^2 \phi^h}{\partial h_x^2}\Big|_0 \frac{h_x^2}{2!} + \frac{\partial^2 \phi^h}{\partial h_x \partial h_y}\Big|_0 h_x h_y + \frac{\partial^2 \phi^h}{\partial h_y^2}\Big|_0 \frac{h_y^2}{2!} + \cdots, \end{aligned} \quad (5.14)$$

where, $\phi^h|_0 = \lim_{\substack{h_x \rightarrow 0 \\ h_y \rightarrow 0}} \phi^h(x, y, h_x, h_y) = \phi(x, y)$

Equation (5.14) can be rewritten in the following form (note: from this point forward the dependence of ϕ^h on x and y is implied but not explicitly written):

$$\phi^h(h_x, h_y) = \phi^h|_0 + f_1 h_x + f_2 h_y + f_3 h_x^2 + f_4 h_x h_y + f_5 h_y^2 + \dots, \quad (5.15)$$

This series expansion can form the basis of an extrapolation procedure for the aforementioned class of grids. For example, in the case of a second-order accurate numerical method (i.e. second-order accurate in both spatial dimensions), $f_1 = f_2 = 0$. To obtain an extrapolated result that is third-order accurate, Eq. (5.15) is truncated after the $f_5 h_y^2$ term. By using four different grids in which h_x and/or h_y are varied independently (in the asymptotic region), the following set of equations can be formed:

$$\begin{Bmatrix} \phi^h(h_{x1}, h_{y1}) \\ \phi^h(h_{x2}, h_{y2}) \\ \phi^h(h_{x3}, h_{y3}) \\ \phi^h(h_{x4}, h_{y3}) \end{Bmatrix} = \begin{bmatrix} 1 & h_{x1}^2 & h_{x1} h_{y1} & h_{y1}^2 \\ 1 & h_{x2}^2 & h_{x2} h_{y2} & h_{y2}^2 \\ 1 & h_{x3}^2 & h_{x3} h_{y3} & h_{y3}^2 \\ 1 & h_{x4}^2 & h_{x4} h_{y4} & h_{y4}^2 \end{bmatrix} \begin{Bmatrix} \phi^h|_0 \\ f_3 \\ f_4 \\ f_5 \end{Bmatrix}. \quad (5.16)$$

The extrapolated solution, $\phi^h|_0$, is readily obtainable by solving this system. The analogous extrapolation procedure for an m -order accurate numerical method follows in a straightforward manner. However, if a higher-order extrapolation is desired, or if the problem involves three or more dimensions, this approach quickly becomes impractical due to the large number of different grids and corresponding discrete solutions required. An alternative extrapolation procedure could be devised in which a series of different grids are constructed by varying h_x and h_y *proportionally*. That is, they are no longer varied independently, but instead are constrained to vary in proportion to a single grid-size parameter, h_c , in the following manner:

$$\begin{aligned} h_x &= \alpha_x h_c \\ h_y &= \alpha_y h_c \end{aligned} \quad (5.17)$$

Upon substituting these expressions back into Eq. (5.15), and collecting terms of like powers of h_c , Eq. (5.15) can be recast into,

$$\phi^h(\alpha_x h_c, \alpha_y h_c) = \phi^h(h_c) = \phi^h|_0 + f'_1 h_c + f'_2 h_c^2 + \dots \quad (5.18)$$

Thus, the original extrapolation technique outlined in Sections 5.2 and 5.3 is applicable once again.

Richardson extrapolation can also be applied to non-uniform grids. For example, consider the particular case of a one-dimensional grid with $N+1$ grid points distributed exponentially along the x -axis according to:

$$\begin{aligned} x_0 &= 0, \\ x_i &= x_{i-1} + \Delta x_{i-1}, \quad i = 1, 2, \dots, N, \end{aligned} \quad (5.19)$$

where, $\Delta x_i = \Delta x_0 \exp(\beta x_i)$, and β is a stretching parameter. The grid spacing, Δx_i , can, therefore, be expressed in terms of a characteristic grid parameter, h_c , in the following manner:

$$\Delta x_i = \alpha_i h_c. \quad (5.20)$$

If h_c is arbitrarily set to $h_c = \Delta x_0$, then the proportionality constants, α_i , $i = 0, 1, 2, \dots, N$, can be extracted from,

$$\alpha_i = \alpha(x_i) = \exp(\beta x_i) \quad (5.21)$$

Using the continuous function, $\alpha(x)$, a series of similarly stretched grids can be constructed by varying the characteristic grid parameter, h_c , in Eq. (5.20) (Note that h_c and N will have to be varied appropriately in order to span the same range of x). Since each of these grids is uniquely determined by the single grid parameter h_c , the corresponding discrete solution can be expanded in h_c yielding,

$$\phi^h(h_c) = \phi^h|_0 + f'_1 h_c + f'_2 h_c^2 + \dots \quad (5.22)$$

A standard Richardson extrapolation can now be performed using Eq. (5.22). It should be noted that the nodes on one particular grid might not occupy the same x -locations as the nodes on any other grid in the set. As discussed in Section 5.2, this is of no consequence

since any local quantity at a specific x -location can be computed on a particular grid using a consistent or higher order interpolation scheme. Then the same quantity computed on all other grids can be extrapolated using Eq. (5.22).

5.5 Extended Richardson Extrapolation

An extended Richardson extrapolation procedure for unstructured grids is presented in this section. It is based on concepts developed in the previous sections of this chapter. The following discussion pertains to two-dimensional, unstructured grids composed of triangular elements, such as the ones used in this thesis for the simulation of incompressible fluid flow and heat transfer. Similar extrapolation schemes, however, may also be devised for unstructured grids composed of quadrilateral elements without any other special considerations.

The main problem in devising a Richardson-type extrapolation procedure for unstructured, solution-adaptive grids is that for any given problem, different grids can vary drastically in terms of the local element shapes and patterns; the element density distributions; and the total number of elements, N_e . For an arbitrary set of such grids, it may no longer be valid to express the discrete solution, ϕ^h , as a Taylor series expansion in a single grid-length parameter related only to N_e . Merely specifying N_e does not uniquely determine an unstructured grid as it might for a structured grid with uniform grid spacing. What is needed is a process whereby a set of two or more unstructured grids can be constructed and *uniquely* characterized (with respect to each other) by specifying a *single* grid-length parameter, h_c , under a given set of practical guidelines.

5.5.1 Hierarchy of Grids

The proposed extrapolation procedure involves constructing a series of grids by subdividing each of the elements of a *base grid* into an integer number of similar elements. This refinement process is illustrated using Fig. 5.1. Figure 5.1 (a) shows a sample element from the base grid. Figure 5.1 (b) displays the corresponding elements in the second grid constructed by first subdividing the sides of element (a) into two equal

segments, and then forming the *four* similar triangles as shown. The elements depicted in Fig. 5.1 (c) are constructed by first subdividing the sides of element (a) into *three* equal segments, and then forming the *nine* similar triangles as shown. Similarly, the n^h grid is constructed by subdividing the sides of each base element into n equal segments to form n^2 similar triangles. Figure 5.2 shows an example of (a) a base grid; (b) the second grid in the hierarchy; (c) the third grid; and (d) the seventh grid. The base grid, in principle, may be any suitable grid within the asymptotic region of grid convergence for the particular problem being considered. To expedite local grid convergence economically (before commencing the extrapolation), a solution-adaptive scheme based on reducing and distributing the estimated error evenly over all elements (see Chapter 4) can be applied during the solution procedure. The final adapted mesh can then be used as the base grid in the aforementioned refinement process.

5.5.2 Extrapolation

Each of the grids in the hierarchy described in the previous subsection, can be assigned a characteristic grid length, h_c , which uniquely characterizes that grid with reference to the base grid and all other grids in the extrapolation set. For example, if the base grid is arbitrarily assigned the value $(h_c)_1 = 1$, then the k^h grid will have a characteristic length of,

$$(h_c)_k = \frac{1}{k}. \quad (5.23)$$

Assuming ϕ^h can be represented by the following series expansion in h_c ,

$$\phi^h(h_c) = \phi^h|_0 + f_1' h_c + f_2' h_c^2 + f_3' h_c^3 + \dots, \quad (5.24)$$

then the generalized Richardson extrapolation procedure outlined in Section 5.3 can be applied to Eq. (5.24) as well.

5.5.3 Mathematical Justification

Consider a base grid composed of N_e elements. Each element j can be assigned a characteristic element length δ_j equal to the length of a side, for example. Now consider

the case where each $\delta_j, j=1, 2, \dots, N_e$, is allowed to vary independently by subdividing the corresponding element according to the procedure outlined in Section 5.5.1. Then the computed solution, ϕ^δ , can be written as a discrete function of each characteristic element length, δ_j , as follows:

$$\phi^\delta = \phi^\delta(\delta_1, \delta_2, \dots, \delta_{N_e}). \quad (5.25)$$

Assuming a continuous function analogous to Eq. (5.25) exists and is infinitely differentiable with respect to each δ_j , then the following multi-variable Taylor series expansion may be written,

$$\phi^\delta(\delta_1, \delta_2, \dots, \delta_{N_e}) = \phi^\delta\Big|_0 + \frac{\partial \phi^\delta}{\partial \delta_1}\Big|_0 \delta_1 + \frac{\partial^2 \phi^\delta}{\partial \delta_1 \partial \delta_2}\Big|_0 \frac{\delta_1 \delta_2}{2!} + \frac{\partial^3 \phi^\delta}{\partial \delta_1 \partial \delta_2 \partial \delta_3}\Big|_0 \frac{\delta_1 \delta_2 \delta_3}{3!} + \dots, \quad (5.26)$$

where summation from $1, 2, \dots, N_e$ is implied over repeated indices. Now suppose each characteristic element length, δ_j , is constrained to vary proportionally. That is, they are no longer varied independently, but instead are *all* varied in proportion to a single characteristic grid-length parameter, h_c , as follows:

$$\delta_j = \alpha_j h_c, \quad j = 1, 2, \dots, N_e. \quad (5.27)$$

Substituting Eq. (5.27) back into Eq. (5.26) and collecting terms of like powers of h_c yields,

$$\phi^\delta(\alpha_1 h_c, \alpha_2 h_c, \dots, \alpha_{N_e} h_c) = \phi^h(h_c) = \phi^h\Big|_0 + f_1' h_c + f_2' h_c^2 + \dots, \quad (5.28)$$

which is identical to Eq. (5.24).

5.5.4 Order of Accuracy of the CVFEM

In the simulation of *diffusion dominated* problems, the present CVFEM employs linear interpolation functions to approximate the dependent variables over each triangular element (Section 2.4.3), which corresponds to a second-order accurate numerical method (assuming the diffusion coefficient, Γ , is also treated in a consistent manner). However, in the case of advection-diffusion-type problems, including Navier-Stokes problems, the

FLO scheme is usually employed to approximate the dependent variables in the advection terms (Section 2.4.4). In the limiting case of zero advection transport or negligibly small element Peclet number, the FLO scheme reduces to a linear interpolation function and thereby becomes second-order accurate. In the case of negligible diffusion transport, however, the FLO scheme reduces to a first-order accurate upwind scheme. In between these two extremes, the FLO scheme automatically adjusts to the relative strengths of advection and diffusion within each element, and may exhibit convergence rates anywhere from $O(h)$ to $O(h^2)$ or even greater in some cases (see Section 6.4). Thus, in the context of Richardson extrapolation, it is not always clear what the leading-order term in Eq. (5.24) should be (i.e. $f_1 h_c$ or $f_2 h_c^2$). Elkouh [36] used the entire series (including the $f_1 h_c$ term) in his extrapolations even though his numerical method was capable of achieving convergence rates of $O(h^2)$. The reasoning behind keeping the linear term in the series is that when the numerical method is second-order accurate, this term would come out to be zero anyway, provided enough higher-order terms are kept in the series. Thus, in the absence of knowing the true order of the numerical method, one may conclude that keeping the linear term is just “playing it safe”. In the present work, both scenarios were tried. That is, extrapolations were performed *with* the linear term, and *without* the linear term (See Section 6.4). It was found that the linear term actually contaminated the extrapolation in cases where the numerical method was indeed achieving convergence rates of $O(h^2)$. Thus, in the present work, efforts were made to determine the true rates of convergence of the CVFEM before the extrapolations were performed. In most cases, the CVFEM *did* indeed exhibit second-order convergence rates, and the linear term, therefore, was omitted in the corresponding extrapolations.

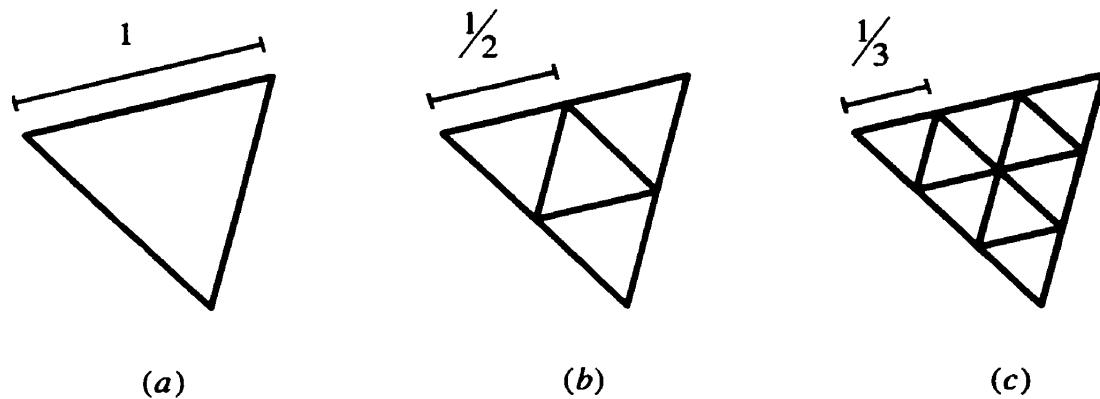


Figure 5.1: Element subdivision technique used to construct the grid hierarchy in the extended Richardson extrapolation: (a) sample element from the base grid; (b) corresponding elements in the second grid after subdividing the base grid element into four similar elements; (c) corresponding elements in the third grid.

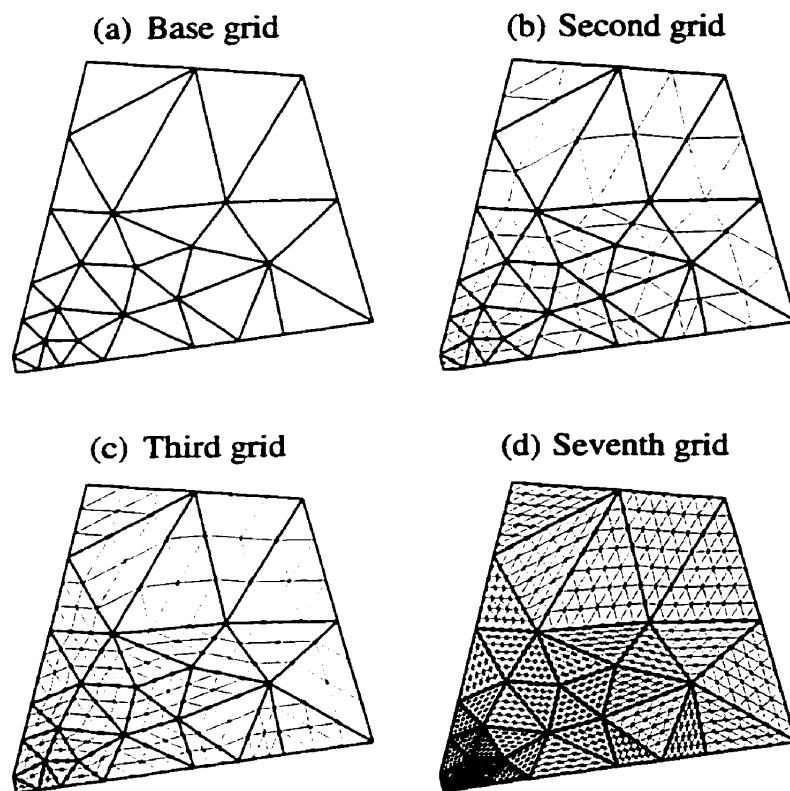


Figure 5.2: Example of four grids in a hierarchy of grids used in the proposed extended Richardson extrapolation technique: (a) the base grid; (b) the second grid; (c) the third grid; (d) the seventh grid in the hierarchy.

Chapter 6

RESULTS

The main results of the thesis are presented in this chapter.

In Section 6.1, the additive correction multigrid (ACM) method that was implemented to accelerate iterative convergence in the CVFEM solution procedure is verified and tested on two model problems. The efficiency of the ACM is demonstrated in relation to the point-by-point SOR Gauss-Seidel iterative solver by simulating a heat conduction problem on an irregular domain. The adaptive (directional) volume agglomeration technique is then tested on a Poisson equation with variable diffusion coefficient.

In Section 6.2, a particular solution to the steady, two-dimensional, Navier-Stokes equations is derived for the purposes of testing and verifying various algorithms in subsequent sections of the chapter. Analytical expressions for a divergence-free velocity field are specified and substituted into the momentum equations. Then, a consistent pressure field is chosen and the implied volumetric source terms are derived. The resulting flow pattern depicts a series of symmetrically located recirculation eddies superimposed over a uniform flow.

In Section 6.3, the analytical flow field, discussed in Section 6.2, is simulated using the CVFEM derived in Chapter 2. The numerical results are then compared with the corresponding exact analytical expressions in order to test the effectiveness of the error estimators and recovery schemes employed in the h -adaptive procedure (outlined in Chapter 4).

Section 6.4 presents the verification and testing of the extended Richardson extrapolation scheme for unstructured grids, proposed in Chapter 5. The aforementioned analytical flow field is simulated using the CVFEM on a series of successively finer grids, and the

results are then extrapolated to the limit of infinite grid density. The exact analytical expressions are subsequently used to determine the accuracy of the extrapolations.

Finally, in Section 6.5, lid-driven flow in a triangular cavity is simulated. The strain-rate error estimator (Section 4.2.2) and flow-angle indicator (Section 4.2.3) are used independently, and in combination, to illustrate their effectiveness in driving the solution-adaptive procedure. Results are compared with a solution available in the published literature [152].

The figures in this chapter are presented immediately following Section 6.5, and the tables are grouped together after the figures.

6.1 Verification of the Additive Correction Multigrid Solver

Details of an additive correction multigrid (ACM) method were presented in Chapter 3. This ACM was implemented in order to accelerate iterative convergence in the CVFEM solution procedure. The method employs a grid-coarsening algorithm based on an adaptive volume-agglomeration technique [41,42]. The current ACM implementation was tested on two model problems: (i) steady-state heat conduction in an elliptically shaped domain with an asymmetrically located triangular hole; and (ii) the Poisson equation with a spatially varying diffusion coefficient, on a square domain.

6.1.1 Heat Conduction in an Elliptically Shaped Domain with an Asymmetrically Located Triangular Hole

Problem formulation. Linear, steady-state heat conduction through an isotropic material is described, mathematically, by the Laplace equation, which can be obtained from Eq. (2.4) by setting $\bar{\mathbf{V}} = 0$ and $S_s = 0$, yielding,

$$\bar{\nabla} \cdot (\Gamma_s \bar{\nabla} \phi) = 0. \quad (6.1)$$

In this equation, ϕ represents the temperature field, and Γ_s represents the thermal conductivity. In the present case, Γ_s was taken to be constant. Equation (6.1) is defined

on the two-dimensional domain depicted in Fig. 6.1 (a). The outer boundary is a rotated ellipse given by,

$$\left(\frac{x'}{a}\right)^2 + \left(\frac{y'}{b}\right)^2 = 1, \quad (6.2)$$

where,

$$\begin{Bmatrix} x' \\ y' \end{Bmatrix} = \begin{bmatrix} \cos \varphi & \sin \varphi \\ -\sin \varphi & \cos \varphi \end{bmatrix} \begin{Bmatrix} x \\ y \end{Bmatrix},$$

and $a = 2$; $b = 1$; $\varphi = \pi / 6$. The inner boundary is an asymmetrically located equilateral triangle whose three vertex positions (x, y) are given by:

$$\left(\frac{\sqrt{3}}{2} + \frac{7}{20}, \frac{1}{2} \right); \left(\frac{\sqrt{3}}{2} - \frac{7}{40}, \frac{1}{2} \pm \frac{7\sqrt{3}}{40} \right)$$

Dirichlet conditions are applied on all the boundaries. The temperature on the outer boundary (the ellipse) is specified as $\phi_{outer} = 0$, and on the inner boundary (the triangular hole) the temperature is fixed at $\phi_{inner} = 1$.

Discussion. This problem was used to test the efficiency of the ACM implementation. The ACM iterative convergence rates were compared with those of the standard successive-over-relaxation Gauss-Seidel (SOR-GS) point solver for a series of different grid sizes. The grids were initially obtained by performing several h -adaptive CVFEM simulations. The scalar diffusion error estimator (Section 4.2.1) was used to drive the adaptive procedure in each case. The final adapted meshes were then used as fixed grids in a series of *non-adaptive* runs. Two non-adaptive runs were executed on each fixed grid – one using the ACM solver and the other using the standard SOR-GS solver. The CPU times and total *work units* were monitored in each run and compared. Here, a work unit is defined as the total computational effort required to complete one sweep of the finest grid in the ACM cycle using the base solver (which, in the present implementation, is SOR-GS itself).

Preliminary h -adaptive solutions. A representative example of a final adapted grid used in the iterative convergence tests is shown in Fig. 6.1 (a). A surface plot of the calculated temperature distribution is presented in Fig. 6.1 (b). This particular mesh, which contains

1062 nodes and 1974 elements, was obtained by specifying a relatively modest target error of $\bar{\eta}_{q^*} = 0.1$ (details of the h -adaptive procedure, including the definition of the target error, were presented earlier in Chapter 4). The h -adaptive procedure successfully refined the mesh in the vicinity of the three vertices of the triangular hole where the temperature field is non-differentiable and where the heat flux is, therefore, discontinuous. In order to arrive at a final converged mesh during the adaptive procedure, a minimum-area constraint was imposed on each element in the domain. As was discussed in Section 4.3.3, this area constraint is needed in order to avoid excessive mesh densities near the discontinuities. The minimum area of any element, expressed as a fraction of the total area of the domain, was specified as $(\Omega_e)_{\min} / \Omega = 5 \times 10^{-5}$ for the particular grid shown in Fig. 6.1 (a).

ACM grid levels. Figure 6.2 (a) shows the centroid-median tessellation (control volume surfaces) corresponding to the finite element mesh in Fig. 6.1 (a). In the context of the ACM, Fig. 6.2 (a) represents the finest grid level in the ACM cycle; (b) shows the second grid in the cycle; and (c) shows the *sixth* and final level in the ACM cycle for this particular run. Each control volume in grid (b) was formed by agglomerating two or more fine-grid volumes in grid (a). Similarly, the last grid (c) was constructed by successively agglomerating control volumes at each level (see Section 3.2, if required, for a review of this procedure).

Results. The actual convergence tests were performed on much finer grids than the one shown in Fig. 6.1 (a). Table 6.1 gives specifications of each of the three grids used in the tests. The coarsest grid contained approximately 26 thousand (26k) nodes and the finest grid contained over 104k nodes. These grids are too fine to be displayed as figures, but their element distribution patterns resemble the pattern of the coarse grid shown in Fig. 6.1 (a). The target errors, $\bar{\eta}_{q^*}$, and minimum-area constraints, $(\Omega_e)_{\min} / \Omega$, specified in the preliminary adaptive analyses for each grid are also given in Table 6.1.

In each run, the initial guessed temperature field was set to zero everywhere in the domain (except on the boundaries, where the aforementioned Dirichlet conditions were enforced) before the iterative solution procedure was started. As mentioned previously,

the convergence tests were *non-adaptive* runs using the three adapted grids in Table 6.1 as *fixed* grids. The iterative solutions to the discretized equations were considered converged when the maximum absolute residual of all the equations fell below 1×10^{-12} . In the ACM runs, the default number of sweeps of the base solver (Section 3.4) was set to 2 prior to restriction, and 3 just after prolongation. For the 104k run, however, the cycling algorithm automatically increased the number of post-prolongation sweeps to as many as 20 during the solution procedure. This was done in order to ensure that the root-mean-square of the fine-grid residuals was reduced by a factor of at least 2 after each complete ACM cycle. The over-relaxation parameter used in the SOR-GS base solver was 1.4.

All runs were performed on an SGI POWER Challenge with 8 R8000 CPUs (75MHz nominal clock speed) and 2 gigabytes of RAM. Each CPU is capable of 300 Mflops peak and 100-150 Mflops sustained. Each individual run was performed on a single devoted processor (that is, no parallelization).

In Table 6.2, ACM results are compared with the standard SOR-GS CPU times and total number of work units. The total number of grid levels in the ACM cycles for each run is also shown. CPU times using the AMC solver were nearly ten times less than the SOR-GS times for the 26k grid and the 54k grid. In the case of the 104k grid, the CPU time required by SOR-GS was well over ten times greater than that required by the ACM. These results suggest that the relative improvements in ACM performance over SOR-GS would only *increase* for even larger grids. The same trend was found by Raw [41], and Elias *et al.* [42]. Figure 6.3 gives a graphical representation of the convergence histories. The ACM and SOR-GS solvers are compared directly for each run. In this graph, the maximum absolute residual is plotted versus the number of work units (note that one full work unit is different for each run, in terms of the absolute number of computer operations, as is elaborated in Table 6.2). It is evident that the slopes of the ACM curves for the 54k and 104k grids are very similar, suggesting that the expected *linear* performance of the ACM solver is being approached with increasing grid size. The phrase “linear performance” refers to a linear relationship between the total computational effort required by the ACM solver and the total number of nodes (or equations) in the problem [41,56].

These convergence tests demonstrate the efficiency of the ACM in terms of CPU time and effort. The problem considered in the next subsection was designed to test the implementation of the directional volume agglomeration algorithm.

6.1.2 Solution to the Poisson Equation with Variable Diffusion Coefficient

Problem formulation. The Poisson equation can be obtained from Eq. (2.4) by setting $\bar{\mathbf{V}} = \mathbf{0}$, yielding,

$$\bar{\nabla} \cdot (\Gamma_\phi \bar{\nabla} \phi) = -S_\phi. \quad (6.3)$$

In this equation, ϕ may be interpreted as a specific scalar field associated with a steady, linear, diffusion transport process; Γ_ϕ represents the related diffusion coefficient; and S_ϕ is the appropriate volumetric source term. Equation (6.3) was solved, numerically, on the unit square defined by $(0 \leq x, y \leq 1)$. A value of $\phi = 0$ was specified on all boundaries, and the source term was taken to be $S_\phi = 10$. Two different cases, involving different prescriptions of Γ_ϕ , were considered: in the first case, the diffusion coefficient was assigned the constant value of $\Gamma_\phi = 1$; and in the other case, it was varied according to $\Gamma_\phi = x + y$. Contour plots of the corresponding CVFEM solutions for these two cases are shown in Fig. 6.4 (a) and (b), respectively.

Discussion. The adaptive volume-agglomeration algorithm was implemented to exploit the effectiveness of the ACM base solver (SOR-GS) in reducing high-frequency error components *preferentially in large coefficient directions* (Sections 3.1 and 3.2). To achieve this added effectiveness, the algorithm was especially designed to form coarse-grid volumes by agglomerating fine-grid volumes in these particular directions. The above-mentioned Poisson problem was specifically devised to test this particular algorithm. The CVFEM approximation corresponding to Eq. (6.3) was solved on the uniform grid shown in Fig. 6.5 (a). The control volumes associated with this grid are shown in Fig. 6.5 (b). For the case of $\Gamma_\phi = 1$, the resulting neighbour coefficients, a_{nb} , in Eq. (3.3) associated with each node in the domain, have no particular directional bias due to Γ_ϕ or due to the uniform grid (at least in the global sense). Thus, for this particular

case, the adaptive volume-agglomeration algorithm should be expected to yield a fairly uniform agglomeration pattern owing to the lack of any directional coefficient coupling. For the case of $\Gamma_\phi = x + y$, however, the neighbour coefficients, in general, will steadily increase in value along the direction of steepest ascent of Γ_ϕ (that is, parallel to $\bar{\nabla}\Gamma_\phi = \bar{i} + \bar{j}$). One would, therefore, expect the agglomeration algorithm to agglomerate control volumes in this direction in the absence of any other factors that may influence the values of a_{nb} .

Results. The centroid-median tessellation shown in Figure 6.5 (b) corresponds to the first grid in the ACM cycle, which was common for both treatments of Γ_ϕ . In Fig. 6.6, the second, third, and fourth grid levels in the ACM cycle are shown for the case of $\Gamma_\phi = 1$ (left column), and $\Gamma_\phi = x + y$ (right column). The same directional volume agglomeration algorithm was used to construct the grid levels in each case. As expected, the agglomeration patterns for the case of $\Gamma_\phi = 1$ are fairly uniform. The resulting volumes have relatively rounded shapes and lack any consistent directional bias. For the case of $\Gamma_\phi = x + y$, however, practically every volume in the second and third grid levels is elongated along the direction of the gradient of Γ_ϕ , as anticipated. In going from the third grid to the fourth grid, however, the volumes were no longer agglomerated in this direction. This suggests that the coefficient coupling due to the elongated shapes of the volumes in the third grid was greater than the coefficient bias caused by the gradient in Γ_ϕ .

An exhaustive study on the influence of directional volume agglomeration on ACM performance is beyond the scope of this thesis. However, its benefits have already been thoroughly demonstrated by several researchers [41-43,45,56]. In light of the favourable results presented here and in the previous subsection, the ACM was used, with confidence, as a tool to accelerate iterative convergence in many of the other problems considered in the present work.

6.2 Analytical Solution to the Navier-Stokes Equations with Body Forces

A complex, recirculating flow field, with closed-form analytical expressions for the velocity components and pressure, is presented in this section for the purposes of testing the recovery schemes and error estimators used in the h -adaptive strategy (details of the h -adaptive strategy were given in Chapter 4; testing and verification will be presented in Section 6.3). This analytical flow field will also be used to verify and test the extended Richardson extrapolation scheme (proposed in Chapter 5) in Section 6.4.

6.2.1 Derivation Procedure

The procedure used to derive the present analytical flow field can, in principle, be used to derive other flow fields of arbitrary complexity. Furthermore, it can be easily extended to deriving three-dimensional flows as well. The first step is to prescribe a suitable divergence-free velocity field (that is, a velocity field which satisfies the continuity equation for incompressible fluid flow). The next step is to substitute the corresponding velocity components into the x - and y -momentum equations resulting in two equations involving the pressure gradients. The key now is to prescribe appropriate body force terms in the momentum equations so that the pressure gradients can be successfully integrated yielding a consistent pressure field.

6.2.2 Analytical Flow Field

The steady, two-dimensional, incompressible Navier-Stokes equations for a constant-property fluid are given by Eqs. (2.1) - (2.3). They are rewritten here in nonconservative form as:

Continuity Equation:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0. \quad (6.4)$$

x-Momentum Equation:

$$\rho \left(u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right) = \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) - \frac{\partial p}{\partial x} + S_u. \quad (6.5)$$

y-Momentum Equation:

$$\rho \left(u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} \right) = \mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) - \frac{\partial p}{\partial y} + S_v. \quad (6.6)$$

The proposed velocity components and pressure field are given by,

$$u = u_o - V_o \sin\left(\frac{n\pi y}{l}\right) \left[\cos\left(\frac{n\pi x}{l}\right) - (-1)^n \right], \quad (6.7)$$

$$v = v_o + V_o \sin\left(\frac{n\pi x}{l}\right) \left[\cos\left(\frac{n\pi y}{l}\right) - (-1)^n \right], \quad (6.8)$$

$$\begin{aligned} p = p_o + \rho V_o^2 \left\{ \begin{array}{l} \frac{1}{2} \left[\sin^2\left(\frac{n\pi y}{l}\right) - \cos^2\left(\frac{n\pi x}{l}\right) \right] \\ + (-1)^n \left[\cos\left(\frac{n\pi x}{l}\right) + \cos\left(\frac{n\pi y}{l}\right) \right] \\ - \cos\left(\frac{n\pi x}{l}\right) \cos\left(\frac{n\pi y}{l}\right) \end{array} \right\}. \end{aligned} \quad (6.9)$$

These expressions are defined on the square domain given by ($-l \leq x, y \leq l$). The parameters u_o , v_o , V_o , p_o , and n are arbitrary (user-specified) constants that influence the nature and character of the flow (a description of the flow is given in Section 6.2.3).

It may be verified by direct substitution that the flow field defined by Eqs. (6.7)-(6.9) satisfies the Navier-Stokes equations, given by Eqs. (6.4)-(6.6), provided that the volumetric body force terms, S_u and S_v , are prescribed as,

$$S_u = \frac{n\pi\rho V_o^2}{l} \left\{ (-1)^n \cos\left(\frac{n\pi x}{l}\right) \sin\left(\frac{n\pi x}{l}\right) \cos\left(\frac{n\pi y}{l}\right) + \frac{v_o}{V_o} (-1)^n \cos\left(\frac{n\pi y}{l}\right) \right. \\ + \frac{u_o}{V_o} \sin\left(\frac{n\pi x}{l}\right) \sin\left(\frac{n\pi y}{l}\right) - \frac{v_o}{V_o} \cos\left(\frac{n\pi x}{l}\right) \cos\left(\frac{n\pi y}{l}\right) \\ \left. + \frac{n\pi\mu}{\rho V_o l} \sin\left(\frac{n\pi y}{l}\right) \left[(-1)^n - 2 \cos\left(\frac{n\pi x}{l}\right) \right] \right\}, \quad (6.10)$$

$$S_v = \frac{n\pi\rho V_o^2}{l} \left\{ (-1)^n \cos\left(\frac{n\pi x}{l}\right) \cos\left(\frac{n\pi y}{l}\right) \sin\left(\frac{n\pi y}{l}\right) - \frac{u_o}{V_o} (-1)^n \cos\left(\frac{n\pi x}{l}\right) \right. \\ + \frac{u_o}{V_o} \cos\left(\frac{n\pi x}{l}\right) \cos\left(\frac{n\pi y}{l}\right) - \frac{v_o}{V_o} \sin\left(\frac{n\pi x}{l}\right) \sin\left(\frac{n\pi y}{l}\right) \\ \left. - \frac{n\pi\mu}{\rho V_o l} \sin\left(\frac{n\pi x}{l}\right) \left[(-1)^n - 2 \cos\left(\frac{n\pi y}{l}\right) \right] \right\}. \quad (6.11)$$

6.2.3 Description of the Flow

The flow field defined by Eqs. (6.7)-(6.9) can be described, generally, as a set of n^2 similar vortices superimposed on a uniform velocity field given by $u_o \hat{i} + v_o \hat{j}$. The sign of V_o determines the rotational direction of the vortices. The vortices recirculate in a counterclockwise (CCW) sense when $V_o > 0$, and in a clockwise (CW) sense when $V_o < 0$. The values of u_o and v_o relative to the magnitude of V_o determine the strength and location of the vortices, and the value of p_o determines the arbitrary level of pressure in the domain.

Figure 6.7 (a) and (b) show the streamline patterns of the analytical flow field for the cases, $n = 2$ and $n = 3$, respectively. The vortices are recirculating in the CW sense indicating that $V_o < 0$, and in these particular cases there is no underlying uniform flow (that is, $u_o = v_o = 0$). The semi-length of the domain is $l = 1$ in this figure.

6.2.4 Simulations

In Sections 6.3 and 6.4, CVFEM simulations of the analytical flow field will be presented for $n = 2$, $l = 1$, and $V_o = -1$. The parameters u_o and v_o were chosen such that the

vortex in the first quadrant of the domain was centered at $(x, y) = (x_c, y_c) = (0.55, 0.40)$. This corresponds to $u_o \approx 1.1468$, and $v_o \approx 0.5590$. The level of pressure was such that $p = 0$ at the origin. Figure 6.8 shows surface plots of (a) the x -component of velocity and (b) the y -component of velocity, corresponding to these particular parameters. Figure 6.9 displays (a) the corresponding pressure field and (b) the streamline pattern.

The Reynolds number, Re , was defined as,

$$\text{Re} = \frac{\rho V_o l}{\mu}. \quad (6.12)$$

Simulations were carried out for $\text{Re}=0.001, 0.01, 0.1, 1, 10, 100$, and 1000 . The FLO scheme (Section 2.4.4.1) was employed for the first six Reynolds numbers ($0.001 \leq \text{Re} \leq 100$), however, for $\text{Re}=1000$, the MAW scheme (Section 2.4.4.2) had to be employed in order to obtain converged solutions. The source terms, S_u and S_v , were integrated using a third-order (quadratic), three-point Gauss quadrature [137]. The problem was simulated on 10 different grids per Reynolds number, except for $\text{Re}=1000$ where 8 different grids were used (yielding a total of 68 different CVFEM simulations). The grids were constructed using the extended Richardson refinement technique outlined in Section 5.5.1. The uniform grid shown in Fig. 6.10 (a) was used as the base grid in the refinement process. This was the coarsest of the ten grids and was assigned a characteristic grid size of $(h_c)_1 = 1$. The k^{th} grid in the hierarchy was constructed by subdividing the sides of each base grid element into k equal segments to form k^2 similar elements, as was described in Section 5.5.1. For example, Fig 6.10 (b) shows the second grid in the hierarchy. The k^{th} grid, therefore, has a characteristic grid size of $(h_c)_k = 1/k$, $k=1, 2\dots 10$. For $\text{Re}=1000$, the CVFEM was unable to obtain converged solutions on grids 1 and 2. Thus, only grids 3 to 10 were used for $\text{Re}=1000$. The base grid, shown in Fig. 6.10 (a), is comprised of 512 equal-sized elements and 289 nodes. The 10th grid contains 51200 elements and 25921 nodes.

Since V_o was held fixed in all the runs, the flow patterns for each Reynolds number were identical. However, the source terms (or applied body forces) were varied appropriately, in order to compensate for the different Reynolds numbers. Physically, this may be

interpreted in the following way: as the inertial forces increased relative to the viscous forces (that is, as the Reynolds number was increased), the applied body forces had to compensate for this relative change in order to maintain an identical flow pattern.

With the FLO scheme, the iterative solution procedure encountered convergence difficulties in the case of $\text{Re}=1000$, which is why the MAW scheme had to be employed. This suggests that the negative coefficients encountered in the FLO scheme may have become excessively large, or that the corresponding physical situation (if it exists) may be unstable. For example, turbulence, or other unsteady flow features, may occur at or around $\text{Re}=1000$ for this particular flow.

6.2.5 Global Energy Norm

The strain-rate energy norm, $\|\boldsymbol{\varepsilon}\|$, given by Eq. (4.31), was evaluated numerically for each of the simulations described in the previous subsection. These values were then used to test the extended Richardson extrapolation method in Section 6.4. In this subsection, the exact analytical value is derived, and will be used later to assess the quality of the extrapolations.

Over the square domain, $(-l \leq x, y \leq l)$, Eq. (4.31) may be rewritten as,

$$\|\boldsymbol{\varepsilon}\| = \left\{ \int_{-l}^l \int_{-l}^l \left[\left(\frac{\partial u}{\partial x} \right)^2 + \frac{1}{2} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \right] dx dy \right\}^{1/2}. \quad (6.13)$$

The partial derivatives in the integrand can be obtained by differentiating the expressions for the velocity components given by Eqs. (6.7) and (6.8), yielding,

$$\begin{aligned} \frac{\partial u}{\partial x} &= \frac{n\pi V_o}{l} \sin\left(\frac{n\pi x}{l}\right) \sin\left(\frac{n\pi y}{l}\right); & \frac{\partial u}{\partial y} &= -\frac{n\pi V_o}{l} \cos\left(\frac{n\pi y}{l}\right) \left[\cos\left(\frac{n\pi x}{l}\right) - (-1)^n \right]; \\ \frac{\partial v}{\partial x} &= -\frac{n\pi V_o}{l} \sin\left(\frac{n\pi x}{l}\right) \sin\left(\frac{n\pi y}{l}\right); & \frac{\partial v}{\partial y} &= \frac{n\pi V_o}{l} \cos\left(\frac{n\pi x}{l}\right) \left[\cos\left(\frac{n\pi y}{l}\right) - (-1)^n \right]. \end{aligned} \quad (6.14)$$

Substituting these expressions back into Eq. (6.13), and evaluating the double integral gives the following concise result:

$$\|\boldsymbol{\varepsilon}\| = |2n\pi V_o|. \quad (6.15)$$

For the present case of $n = 2$ and $V_o = -1$, the global strain-rate energy norm is given by,

$$\|\boldsymbol{\varepsilon}\| = 4\pi. \quad (6.16)$$

6.3 Verification of the h -Adaptive Procedure

The h -adaptive strategy, outlined in Chapter 4, was implemented in order to optimize the CVFEM with respect to accuracy and computational effort. The optimizing criteria are based on reducing and distributing the estimated error uniformly over all elements in the domain. The mechanism by which this is achieved, in the present context, is to appropriately adjust the local element density according to the requirements of any particular problem.

In this section, the key components of the adaptive algorithm are verified. Namely, the ability of the recovery schemes to obtain superconvergent, higher-order projections of the CVFEM solution gradients is shown; and the *effectivity* of the error estimators is verified by showing that they are *asymptotically exact* (that is, that the ratio of the estimated error to true error approaches unity in the limit of infinite grid density).

6.3.1 Recovery Schemes

Two different recovery schemes were implemented in the adaptive algorithm: the global L_2 projection of Hinton and Campbell [19] (Section 4.1.1); and a volume-weighted averaging scheme (Section 4.1.2). The purpose of the recovery process in the adaptive procedure is to obtain a higher-order projection of the CVFEM gradients. This higher-order projection is then used to assess the solution error by comparing it with the original CVFEM gradients. It is highly desirable that the recovered gradients be superconvergent relative to the CVFEM values (that is, the recovered nodal gradients have a convergence rate, with respect to grid size, of one order higher than the corresponding CVFEM values). In order to determine whether the recovery schemes possessed this property (and if they did, under what conditions), a complex recirculating flow, with a closed form analytical solution, was simulated on a series of different uniform grids. The grid-convergence histories of the true errors in both the recovered velocity gradients and the

corresponding CVFEM values were monitored for each run. Details of the analytical flow field were discussed in Section 6.2, and specific details pertaining to the simulations were given in Section 6.2.4.

Preliminary discussion. If a CVFEM is said to be strictly m -order accurate, then the element-based interpolation functions should be capable of representing a full $(m-1)$ -order polynomial exactly. Since these interpolation functions approximate the dependent variables themselves, the gradients of the dependent variables, by definition, will be approximated by an $(m-2)$ -order polynomial. Thus, one can only expect $(m-1)$ -order accuracy from the CVFEM gradients, in general. The goal of the recovery schemes is to obtain m -order accurate gradients.

In the case of the present CVFEM, the value of m is often ambiguous owing to the use of the FLO scheme to approximate the advection terms in the integral conservation equations (Section 2.4.4.1). The value of m has been shown to range from 1 to 2 (and even greater, in some cases), depending on the element-based Reynolds number (See Section 6.4). The MAW scheme, however, usually exhibits $O(h)$ convergence.

One method of determining the value of m is as follows: assume that the dependent variable, ϕ^h , has a series representation in some characteristic element size, h_c . That is,

$$\phi^h = \phi_o + f_m h_c^m + f_{m+1} h_c^{m+1} + f_{m+2} h_c^{m+2} + \dots, \quad (6.17)$$

where,

$$\phi_o = \lim_{h_c \rightarrow 0} \phi^h = \phi.$$

Using Eq. (6.17), let the relative error, \bar{e}_ϕ , be defined as,

$$\bar{e}_\phi \equiv \frac{\phi_o - \phi}{\phi_o} = f'_m h_c^m + f'_{m+1} h_c^{m+1} + f'_{m+2} h_c^{m+2} + \dots, \quad (6.18)$$

In these expressions, the coefficients, f_m, f_{m+1}, \dots , and f'_m, f'_{m+1}, \dots , are independent of h_c . For small enough values of h_c (within the asymptotic range), Eq. (6.18) can be truncated after the h_c^m term. Taking the logarithm of the resulting expression yields:

$$\log |\bar{e}_\phi| = m \log h_c + K. \quad (6.19)$$

If the exact solution to the analogous continuum problem, ϕ_o , is known, then m may be approximated by plotting $\log|\bar{e}_s|$ versus $\log h_c$, for several values of h_c (that is, several different grids). The value of m will be equal to the slope of the resulting curve.

In the present verification procedure, this method was used to determine the convergence rates of both the CVFEM gradients and the recovered gradients. The process by which the hierarchy of grids was obtained for the present verification procedure was outlined in Sections 5.5.1 and 6.2.4. Justification for characterizing each grid with a single characteristic grid length was discussed in Section 5.5.3.

Results. Figures 6.11 to 6.13 present the results of this verification procedure. In these figures, the following notation is used: G is used to denote the gradient of a particular velocity component. The specific gradient that G represents in any given graph is indicated inside that graph. The relative error associated with G is denoted by \bar{e}_G . The superscripts, h , L , and V , correspond to the CVFEM, global L_2 projection, and volume-averaged values, respectively. For example, in the case of $G = \partial u / \partial x$, we have,

$$\bar{e}_G^h = \frac{\frac{\partial u}{\partial x} - \frac{\hat{\partial} u^h}{\partial x}}{\frac{\partial u}{\partial x}}; \quad \bar{e}_G^L = \frac{\frac{\partial u}{\partial x} - \frac{\hat{\partial} u^L}{\partial x}}{\frac{\partial u}{\partial x}}; \quad \bar{e}_G^V = \frac{\frac{\partial u}{\partial x} - \frac{\hat{\partial} u^V}{\partial x}}{\frac{\partial u}{\partial x}}, \quad (6.20)$$

where $\partial u / \partial x$, without any superscript, represents the exact gradient.

In Fig. 6.11, the convergence histories of the CVFEM and recovered gradients (with respect to the characteristic element size of the computational grid) are presented for a node located at $(x,y)=(0.125, 0.125)$. This node location exists on each grid (See Section 6.2.4). The logarithm of the relative error is plotted versus the logarithm of the characteristic grid size. Each data point corresponds to an individual simulation. These results were obtained for the case of $Re=1$. It is evident from Fig. 6.11 (a)-(d) that the CVFEM gradients exhibited $O(h_c)$ convergence (that is, $m=1$), as expected. In (a) and (d), both the global L_2 projection and the volume-averaged recovery schemes were successful in achieving superconvergence ($m=2$). In (b) and (c), the global L_2 projection was superconvergent, however, the volume-averaged values had a convergence rate of

slightly less than $O(h_c^2)$. It should be noted that while superconvergence is highly desirable, it is not an essential feature for driving the solution-adaptive process. As long as the recovered gradients achieve convergence rates of $O(h_c^\alpha)$ with $\alpha > 1$, then the adaptive procedure is still able to arrive at a final adapted mesh, however, more adaptive iterations may be required. In Fig. 6.12, convergence histories of the gradients are presented for a node located at $(x,y)=(0.375,0.625)$. Again, these results are for the case of $Re=1$. Both recovery schemes are superconvergent (or nearly superconvergent) at this particular nodal location as well.

In Fig. 6.13, convergence histories of $G = \partial u / \partial x$ for seven different Reynolds numbers are presented at the nodal location, $(x,y)=(0.125,0.125)$. In (a), the curves for $Re=0.001$, 0.01 , 0.1 , and 1 practically overlap. In each of these cases, superconvergence was obtained. In (b) and (c) convergence histories for $Re=10$ and $Re=100$, respectively, are shown. Again, essentially superconvergent behavior was obtained. In (d), however, the convergence rates of the recovered gradients, in the case of $Re=1000$, are actually *less* than the corresponding CVFEM convergence rate. This suggests that the grids used were not fine enough to be within the asymptotic region of grid convergence for the case of $Re=1000$. Simulating the problem for this particular Reynolds number proved to be very difficult. The FLO scheme was unable to yield converged solutions. The MAW scheme, which is less accurate than the FLO scheme, had to be used instead (See discussion in Section 6.2.4).

In general, both recovery schemes were able to produce superconvergent results for low to moderate Reynolds numbers ($Re \leq 100$). The other major components in the solution-adaptive algorithm that need to be verified are the error estimators. This is presented in the next subsection.

6.3.2 Error Estimators

Two error estimators were implemented in the present h -adaptive algorithm: an error estimator for scalar diffusion (Section 4.2.1); and a strain-rate error estimator for low to moderate Reynolds number Navier-Stokes flows (Section 4.2.2). Their function is to

process information obtained from the recovery schemes, and to use this processed information to provide some measure of the local and global solution error. These error estimators are tested in the present subsection. A *flow-angle indicator* was also implemented to work in conjunction with the strain-rate error estimator. Testing of the flow-angle indicator is deferred until Section 6.5.

6.3.2.1 Poisson Equation on a Circular Domain with Prescribed Gaussian Distributions

Problem formulation. The Poisson equation can be obtained from Eq. (2.4) by setting $\bar{\nabla} = 0$, yielding,

$$\bar{\nabla} \cdot (\Gamma_s \bar{\nabla} \phi) = -S_s. \quad (6.21)$$

In this equation, ϕ is the dependent scalar variable; Γ_s represents the related diffusion coefficient; and S_s is the appropriate volumetric source term. Equation (6.21) was solved, numerically, on a circle of unit radius centered at the origin. The source term, S_s , was prescribed as,

$$S_s = 4 \sum_{i=1}^n A_i \alpha_i \left\{ 1 - \alpha_i [(x - x_i)^2 + (y - y_i)^2] \right\} \exp \left\{ -\alpha_i [(x - x_i)^2 + (y - y_i)^2] \right\}. \quad (6.22)$$

An analytical solution to Eq. (6.21) is the superposition of n Gaussian distributions given by,

$$\phi = \phi_o + \sum_{i=1}^n A_i \exp \left\{ -\alpha_i [(x - x_i)^2 + (y - y_i)^2] \right\} \quad (6.23)$$

The following constants were used in the present case:

$$\begin{aligned} n &= 2; \phi_o = 1; \\ A_1 &= 1; \alpha_1 = 50; x_1 = 0.4; y_1 = 0.15; \\ A_2 &= 0.5; \alpha_2 = 15; x_2 = -0.35; y_2 = -0.15. \end{aligned} \quad (6.24)$$

In the CVFEM simulations, the exact analytical solution (Eq. (6.23)) was prescribed as Dirichlet conditions on the boundary.

Discussion and results. This problem was used to test the scalar diffusion error estimator (Section 4.2.1). A surface plot of the CVFEM solution is shown in Fig. 6.14 (a). The ϕ distribution consists of a sharp Gaussian peak centered at $(x,y)=(0.40,0.15)$ and a shallower Gaussian centered at $(x,y)=(-0.35,-0.15)$. Fig. 6.14 (b) shows the corresponding solution-adapted mesh. The error estimator was successful in driving the adaptive algorithm to refine the mesh in the vicinity of the Gaussian peaks, where the largest errors would be expected. The sharper peak required a denser distribution of elements in order to reduce the local error to the specified level. A global target error of $\bar{\eta}_{q^*} = 0.1$ was specified in this particular simulation. In Fig. 6.15, the adaptive-grid solution (right column) is compared with a coarse, uniform-grid solution (left column). The coarse grid (left) in row (a) was the starting grid in the adaptive analysis. The refined grid (right) was the final adapted mesh. In row (b), contour plots of the corresponding solutions are shown. The solution on the left is of rather poor quality. The contour lines are jagged and non-physical. The solution on the right is qualitatively very good. The contour lines are smooth and the Gaussian peaks are very well defined. In row (c), a quantitative comparison of the error distribution is shown. In these bar charts, the relative error, $|\bar{e}_\phi|$, at each node is plotted versus x (the y -axis points into the page). As expected, the largest error on the coarse, uniform mesh occurred in the vicinity of the sharp Gaussian peak. On the adapted mesh, however, the error was successfully distributed evenly throughout the domain.

6.3.2.2 Effectivity Index

In this subsection, the quality of the strain-rate error estimator (Section 4.2.2) is measured by monitoring its effectivity index, κ , using the analytical flow field derived in Section 6.2. The effectivity index is the ratio of the estimated error norm (evaluated using the recovered gradients) to the actual error norm (evaluated using the exact gradients). That is,

$$\kappa^* = \frac{\|\dot{\mathbf{e}}_t^*\|}{\|\dot{\mathbf{e}}_t\|}, \quad (6.25)$$

where $\|\mathbf{e}_\epsilon\|$ is given by Eq. (4.32). The superscript “•” refers, generically, to one of the recovery schemes. The following notation is adopted: if the global L_2 projection is used, then “•” is replaced by “ L ”; if the volume-averaging recovery scheme is used, then “•” is replaced by “ V ”. For example, in the case of the global L_2 projection, $\|\mathbf{e}_\epsilon^*\|$ would be given by,

$$\begin{aligned}\|\mathbf{e}_\epsilon^*\| &= \|\mathbf{e}_\epsilon^L\| \\ &= \left\{ \int_{\Omega} \left[\left(\frac{\partial u^L}{\partial x} - \frac{\partial u^h}{\partial x} \right)^2 + \frac{1}{2} \left(\frac{\partial u^L}{\partial y} - \frac{\partial u^h}{\partial y} + \frac{\partial v^L}{\partial x} - \frac{\partial v^h}{\partial x} \right)^2 + \left(\frac{\partial v^L}{\partial y} - \frac{\partial v^h}{\partial y} \right)^2 \right] d\Omega \right\}^{1/2}. \quad (6.26)\end{aligned}$$

The error estimator is said to be *asymptotically exact* if κ approaches unity as the exact error norm approaches zero (or as $h_c \rightarrow 0$, where h_c is a characteristic element size). This implies that if the error estimator is asymptotically exact, then the estimated error will always converge to the true error as the true error decreases (or as the grid density increases) within the asymptotic region of grid convergence [3].

Results. CVFEM simulations of the analytical flow field were performed for a wide range of Reynolds numbers. The details pertaining to these simulations were outlined in Section 6.2.4. In each simulation, κ^* was evaluated. These values are plotted in Fig. 6.16 versus (the negative logarithm of) the characteristic grid size, h_c . The curves corresponding to $Re = 0.001, 0.01, 0.1$, and 1 , practically overlap in this plot. For each of these Reynolds numbers, and for $Re = 10$, the strain-rate error estimator is shown to be asymptotically exact. For the case of $Re = 100$, the effectivity index is approaching unity, but it has not yet begun to *asymptote* to this value. This suggests that the corresponding CVFEM solutions have just barely entered their asymptotic region of grid convergence, and that finer grids would be required before κ^* reached near-unity values. Thus, as long as the initial grid in the h -adaptive procedure is sufficiently fine, the strain-rate error estimator should still be effective in driving the adaptive process for $Re = 100$. In the case of $Re = 1000$, however, the effectivity index is *diverging* from unity.

In summary, the strain-rate error norm was shown to be an effective error estimator for low to moderate Reynolds number flows. In the case of higher Reynolds numbers,

however, it did not exhibit asymptotic exactness for the range of grid sizes used. Thus, as anticipated (see discussion in Section 4.2.2), the strain-rate error norm, which was originally derived for creeping flows, may not be suitable for driving the adaptive process in high-Reynolds number flows.

6.4 Verification of the Extended Richardson Extrapolation Technique

An extended Richardson extrapolation technique for unstructured grids was outlined in Chapter 5. This technique uses a series of numerical solutions, obtained on grids with different element densities, to extrapolate to the limit of infinite grid density. Although the extrapolation, by necessity, is approximate, the goal is to obtain extrapolated values that are significantly more accurate than any of the corresponding CVFEM solutions.

The proposed extrapolation method was tested on the analytical flow field derived in Section 6.2. A series of simulations were performed for a wide range of Reynolds numbers. For each Reynolds number, the analytical flow field was simulated on 10 different grids (8 grids for $Re=1000$). These 10 grids form an ordered hierarchy with respect to their characteristic grid size, h_c . The value of h_c for each grid is given by $(h_c)_k = 1/k$, $k=1,2\dots 10$ ($k=3,4\dots 10$ for $Re=1000$). Further details pertaining to the simulations were outlined in Section 6.2.4.

Preliminary results. Before testing the actual extrapolation scheme, an effort was made to determine the local and global convergence rates (with respect to h_c) of the CVFEM for each Reynolds number. As was discussed in Section 6.3.1, the FLO scheme (Section 2.4.4.1), which was used to approximate the advection terms in the conservation equations for $Re=0.001 - 100$, can exhibit convergence rates ranging from $O(h_c)$ - $O(h_c^2)$ or even greater. The MAW scheme (Section 2.4.4.2), which had to be employed for $Re=1000$, generally exhibits $O(h_c)$. To perform a proper extrapolation, with as few grids as possible, it is highly advantageous to know the convergence rate (or order of accuracy) of the numerical method (see discussion in Section 5.5.4). This will be illustrated later.

Figure 6.17 shows the convergence histories of the velocity components, for all Reynolds numbers considered here, plotted versus (the logarithm of) the characteristic grid size, h_c . In this graph, $|\bar{e}_u|$ and $|\bar{e}_v|$ are defined as,

$$|\bar{e}_u| = \left| \frac{u - u^h}{u} \right|; \quad |\bar{e}_v| = \left| \frac{v - v^h}{v} \right|. \quad (6.27)$$

These relative errors are evaluated at the nodal location, $(x, y) = (0.125, 0.125)$. Each data point in Fig. 6.17 corresponds to an individual simulation on a particular grid. The hollow markers represent the values of $|\bar{e}_u|$ and the filled markers represent the values of $|\bar{e}_v|$. The curves for $Re = 0.001, 0.01$, and 0.1 practically overlap on this plot. The CVFEM velocity components for each of these Reynolds numbers exhibited convergence rates of $O(h_c^\alpha)$ with $2 \leq \alpha \leq 3$. In the case of $Re = 1$, the convergence rate was $O(h_c^\alpha)$ with $\alpha \approx 3$. For $Re = 10$, the rate of convergence was essentially $O(h_c^2)$, and for $Re = 100$, it was slightly less than $O(h_c^2)$, but it was approaching this convergence rate for smaller values of h_c . As expected, the MAW scheme exhibited $O(h_c)$ for $Re = 1000$.

Figure 6.18 shows the convergence histories for the pressure at the same nodal location. The convergence rates for $Re = 0.001 - 100$ were essentially $O(h_c^2)$, and for $Re = 1000$, it was approximately $O(h_c)$.

The local convergence rates of the velocity components and pressure, displayed in Figs. 6.17 and 6.18, were found to be essentially representative of the convergence rates observed at other nodal locations.

In Fig. 6.19, the global order of accuracy of the CVFEM is illustrated for each Reynolds number. The relative error in the global strain-rate energy norm, $|\bar{e}_{\|\epsilon\|}|$, is plotted versus (the logarithm of) the characteristic grid size, h_c . The strain-rate energy norm is given by Eq. (4.31), and its exact analytical value is derived in Section 6.2.5. The convergence rates for $Re = 0.001 - 10$ were approximately $O(h_c^2)$, and for $Re = 100$ the convergence rate was slightly less than this. For $Re = 1000$, the global energy norm was hardly converging

at all, suggesting that the solutions for this Reynolds number were not within the asymptotic region of grid convergence.

In summary, the CVFEM was able to achieve local and global convergence rates of $O(h_c^2)$ or better for $0.001 \leq Re \leq 10$. In the case of $Re=100$, it appears as though the CVFEM was *approaching* $O(h_c^2)$ convergence as the characteristic grid size, h_c , diminished. In the case of $Re=1000$, however, the convergence rates were observed to be $O(h_c)$ at best. In light of these results, and the discussion in Section 5.5.4, superior extrapolations may be expected, for the lower to moderate Reynolds numbers, if the leading order term in Eq. (5.24) is made to be $f_2 h_c^2$ instead of $f_1 h_c$ (assuming the CVFEM solutions are within the asymptotic region of grid convergence). The following results verify this conjecture.

Results and discussion. Tables 6.3 – 6.4 present data corresponding to a series of extended Richardson extrapolations performed on the x -component of velocity, u , at the nodal location $(x, y)=(0.125, 0.125)$. In Table 6.3 (a), the logarithm of the relative error in u^h is presented for each of the simulations performed. The ten different numbered columns correspond to the ten different grids used $[(h_c)_k, k=1,2\dots 10]$, 1 being the coarsest grid and 10 being the finest grid. Each row corresponds to one of the seven different Reynolds numbers considered. This data is a tabular representation of the results plotted in Fig. 6.17 for $|\bar{e}_u|$. It is useful to note that the magnitude of each number in this table gives, roughly, the corresponding number of correct digits in the value of u^h . For example, if the exact (continuum) value is $u_o \approx 0.939695$, and the CVFEM value is $u^h \approx 0.940424$, then we see that the CVFEM value is accurate to approximately 3 decimal places. Furthermore, we have,

$$\log \left| \frac{u_o - u^h}{u_o} \right| \approx \log \left| \frac{0.939695 - 0.940424}{0.939695} \right| \approx -3.11.$$

In Table 6.3 (b), linear extrapolations (with respect to h_c) are presented for each pair of adjacent grids shown in (a). These are referred to as 2-grid, linear extrapolations. The corresponding series expansion (see Section 5.5.2) is given by,

$$u^h = \bar{u}_o + f_1 h_c, \quad (6.28)$$

where \bar{u}_o denotes the extrapolated value. The extrapolation process associated with Eq. (6.28) involves using two discrete solutions, u_k^h and u_{k+1}^h , and their corresponding grid sizes, $(h_c)_k$ and $(h_c)_{k+1}$, to eliminate f_1 and solve for \bar{u}_o . In Table 6.3 (b), this is done for $k=1,2\dots 9$ ($k=3,4\dots 9$ in the case of $Re=1000$).

In Table 6.3 (c), a series of 3-grid extrapolations are presented. The corresponding series expansion in this case is given by,

$$u^h = \bar{u}_o + f_1 h_c + f_2 h_c^2, \quad (6.29)$$

The extrapolation process associated with Eq. (6.29) involves using three discrete solutions, u_k^h , u_{k+1}^h , and u_{k+2}^h , and their corresponding grid sizes, $(h_c)_k$, $(h_c)_{k+1}$, and $(h_c)_{k+2}$, to eliminate f_1 and f_2 , and solve for \bar{u}_o . In Table 6.3 (c), this is done for $k=1,2\dots 8$ ($k=3,4\dots 8$ in the case of $Re=1000$).

Similarly, in Table 6.3 (d), a series of 4-grid extrapolations are presented. The following series expansion applies in this case:

$$u^h = \bar{u}_o + f_1 h_c + f_2 h_c^2 + f_3 h_c^3. \quad (6.30)$$

In Table 6.4, analogous sets of extrapolations are presented for u at the same nodal location as in Table 6.3. However, in this case, the linear term, $f_1 h_c$, that was present in Eqs. (6.28)-(6.30) is *omitted*. Thus, the series expansions in Table 6.4 (b)-(d) corresponding to the 2-grid, 3-grid, and 4-grid extrapolations, are given by,

$$u^h = \bar{u}_o + f_2 h_c^2, \quad (6.31)$$

$$u^h = \bar{u}_o + f_2 h_c^2 + f_3 h_c^3, \quad (6.32)$$

and, $u^h = \bar{u}_o + f_2 h_c^2 + f_3 h_c^3 + f_4 h_c^4, \quad (6.33)$

respectively. Note that Table 6.4 (a) is identical to Table 6.3 (a), and is shown here again only to facilitate visual comparisons between the CVFEM values and the corresponding extrapolated values.

It is evident from Table 6.3 (b) that the 2-grid extrapolations associated with Eq. (6.28) often gave *worse* results than *either* of the two corresponding CVFEM solutions in that particular extrapolation. This is not surprising since we are trying to use a linear function, in this case, to extrapolate from curves that are essentially quadratic (i.e. $O(h_c^2)$). The only Reynolds number for which the 2-grid linear extrapolation gave significantly improved accuracy was for $\text{Re}=1000$. This is also to be expected since the CVFEM exhibited $O(h_c)$ convergence for this particular Reynolds number (see Fig. 6.17).

Table 6.3 (c) presents 3-grid extrapolations associated with Eq. (6.29). With the addition of the $f_2 h_c^2$ term in the corresponding series expansion, the 3-grid extrapolations consistently produced more accurate results than the corresponding CVFEM values. Furthermore, as the grids in the extrapolations become finer, the accuracy improved. The extrapolations corresponding to grids 4-6 show better-than-expected accuracy. This is believed to be an anomaly: It *may* have been caused by the extrapolation errors *accidentally* canceling out with the truncation errors of the corresponding CVFEM solutions.

Table 6.3 (d) presents the 4-grid extrapolations associated with Eq. (6.30). These values are generally more accurate than the corresponding 3-grid extrapolations, provided the grids in the extrapolations are sufficiently fine. If the 4-grid extrapolations involved grids that were too coarse (for example, grid 4 or coarser), the accuracy generally diminished relative to the corresponding 3-grid values.

In Table 6.4 (b), 2-grid extrapolations corresponding to Eq. (6.31) are presented. The effect of omitting the linear term, $f_1 h_c$, and keeping only the quadratic term, $f_2 h_c^2$, in Eq. (6.31) is quite substantial. In nearly all cases, the 2-grid *quadratic* extrapolation results in (b) were found to be significantly more accurate than the corresponding CVFEM values shown in (a). Perhaps an even more important observation is that the 2-grid *quadratic* extrapolations in Table 6.4 (b), are not only superior to the 2-grid *linear* extrapolations of Table 6.3 (b), they are also superior (in general) to the 3-grid extrapolations of Table 6.3 (c). This implies that the presence of the linear term in Eq.

(6.29) is actually *contaminating* the corresponding extrapolated result (albeit to a relatively small extent in this case).

The 3-grid and 4-grid extrapolations corresponding to Eqs. (6.32) and (6.33), respectively, are shown in Table 6.4 (c) and (d). In practically all cases, these extrapolations are significantly more accurate than the corresponding extrapolations shown in Table 6.3 (c) and (d).

Tables 6.5 and 6.6 present extended Richardson extrapolations of the global strain-rate energy norm, $\|\boldsymbol{\varepsilon}\|$, defined by Eq. (4.31). The exact (continuum) value, $\overline{\|\boldsymbol{\varepsilon}\|}_o$, for the present case was derived in Section 6.2.5. Tables 6.5 and 6.6 are analogous to Tables 6.3 and 6.4, respectively. The only difference being that $\|\boldsymbol{\varepsilon}\|$ is a global quantity, expressed as an integral over the entire domain, while the extrapolations for \boldsymbol{u} presented in Tables 6.3 and 6.4 were performed at a single nodal location. The previous discussion pertaining to the results presented in Tables 6.3 and 6.4 also applies to the results in Tables 6.5 and 6.6. Some of the observations are even more evident in the latter case: As was observed in Fig. 6.19, the convergence rates of the global energy norm for $Re=0.001-10$ are almost exactly $O(h_c^2)$, while the convergence rates of the velocity components, shown in Fig. 6.17, are only *essentially* $O(h_c^2)$. More precisely, the local convergence rates of \boldsymbol{u} for these Reynolds numbers were actually $O(h_c^\alpha)$ with α varying between $2 \leq \alpha \leq 3$. In light of the more stable convergence of $\|\boldsymbol{\varepsilon}\|$ relative to \boldsymbol{u} , one might expect even better extrapolation results for $\|\boldsymbol{\varepsilon}\|$. This was indeed the case as is shown in Tables 6.5 and 6.6.

In fact, some of the extrapolations in Table 6.6 give values of $\overline{\|\boldsymbol{\varepsilon}\|}_o$ to as many as 7 decimal places. But perhaps the most relevant result from these tables is the marked improvement in extrapolated results when the linear term is omitted from the related series expansions in h_c . A particularly striking observation is that the 2-grid *quadratic* extrapolations, shown in Table 6.6 (b), are generally more accurate than *any* of the corresponding *linear-term* extrapolations (that is, extrapolations which included the linear term in their associated series expansions) shown in Table 6.5. This is a significant observation, which illustrates the importance of knowing, or properly estimating, the

order of accuracy of the numerical method, and choosing the corresponding series expansion in h_c appropriately.

In summary, the results presented in Tables 6.3 to 6.6 confirm the earlier conjecture that the series expansion given by Eq. (5.24), corresponding to the extended Richardson extrapolation proposed in Section 5.5, should lead off with the quadratic term, $f_2 h_c^2$, and *not* the linear term, $f_1 h_c$, in order for the extrapolation method to be consistent with the present CVFEM, which exhibits essentially $O(h_c^2)$ convergence. Alternatively, results also suggest that if the numerical method is first order accurate (as was the case for $Re=1000$), then the leading order term should indeed be $f_1 h_c$.

These tests provide evidence that the proposed extended Richardson extrapolation technique, outlined in Section 5.5, is a reliable and effective method of using two or more discrete solutions to obtain a higher-order approximation of the related continuum solution. One must be careful, however, to ensure that the method is applied *only* in the asymptotic region of grid convergence. This also implies monotonic convergence of the discrete solution with respect to h_c . Furthermore, knowledge of the order of accuracy (or convergence rate with respect to h_c) of the numerical method is highly advantageous, so that the appropriate series expansion in h_c can be applied in the corresponding extrapolation procedure.

6.5 Lid-Driven Flow in a Triangular Cavity

Shear-driven flow in a triangular cavity was simulated using the h -adaptive CVFEM developed in Chapters 2 - 4. The strain-rate error estimator (Section 4.2.2) and flow-angle indicator (Section 4.2.3) were tested independently, and in combination, in order to demonstrate their individual strengths and weaknesses, and their combined effectiveness at driving the solution-adaptive procedure for this particular problem. The extended Richardson extrapolation for unstructured grids, proposed in Chapter 5, was used to approximate grid-independent results. These results are compared with the published solution of Jyotsna and Vanka [152].

Problem formulation. The steady, two-dimensional, incompressible Navier-Stokes equations, given by Eqs. (2.1) – (2.3), were solved numerically (using the CVFEM) on the triangular domain shown in Fig. 6.20. Externally applied volumetric body forces, such as gravity, were not considered in this problem (that is, $S_u = S_v = 0$). The top side of the isosceles triangle, shown in Fig. 6.20, represents a solid lid translating to the right with speed U . The height of the triangle was taken as $H = 2W$, where W is the width of the top side of the triangle. The Reynolds number was defined as,

$$\text{Re} = \frac{\rho U H}{\mu} \quad (6.34)$$

The origin, $(x, y) = (0,0)$, was placed at the geometric centre of the lid, with the y -axis pointing upward, out of the domain. Dirichlet conditions were applied on all boundaries: on the top side of the triangle (the translating lid), the velocity components were set to $u = U$, and $v = 0$; and on the two other sides of the triangle (solid, stationary walls), the velocity components were set to $u = 0$ and $v = 0$.

Preliminary discussion. Lid-driven flow in a *square* cavity has been the object of numerous numerical investigations in the published literature (for example, [149–151]). The attention this problem has received is likely attributable to its simple geometry and boundary conditions, and the interesting flow features exhibited at different Reynolds numbers [151]. This problem also serves as a convenient test problem for verifying numerical algorithms. On the other hand, there have been relatively fewer numerical studies pertaining to driven cavity flows in *complex* geometries, even though such geometries are commonly encountered in a variety of engineering applications [152]. In particular, *sharp corners* can be found in many of these applications. One possible idealization of the flow near a sharp corner is the shear-driven flow in a *triangular* cavity, which has been investigated here.

The primary reason for the apparent lack of interest in geometries other than the square is likely due to the difficulties encountered in representing these geometries, and their associated computational domains, using structured grids. Many of these difficulties are described in detail by Ribbens *et al.* [153]. In the present case, with the use of unstructured grids composed of triangular elements, no such difficulties arise. The two

upper corners of the domain, where the velocity is theoretically discontinuous, require no additional special treatment using the current h -adaptive CVFEM.

Streamline patterns and flow features. Lid-driven flow in a triangular cavity was simulated for $Re=50, 100, 400$, and 800 . The corresponding streamline patterns for these Reynolds numbers are shown in Fig. 6.21 (a), (b), (c), and (d), respectively. These solutions were obtained using the h -adaptive CVFEM with specified target errors of $\bar{\eta}_e = 0.1$ and $\bar{\eta}_\psi = 0.2$ prescribed for the strain-rate error estimator and flow-angle indicator, respectively. The minimum area constraint (see Section 4.3.3), expressed as a fraction of the total domain area, was specified as $(\Omega_e)_{\min} / \Omega = 5 \times 10^{-5}$. The total number of elements required to satisfy the above-mentioned adaptive criteria increased with Reynolds number, from 13662 elements for $Re=50$, to 16458 elements for $Re=800$. It is evident from Fig. 6.21 that these flows exhibit a series of similar eddies whose sizes decrease towards the lower vertex of the cavity. Only three eddies are shown per plot here, however, theoretically, there exists an infinite number of such eddies as one approaches the lower vertex, as will be discussed later. For each Reynolds number, the lowermost eddies are centered near the centerline of the cavity ($x=0$). In these regions inertial effects are negligible relative to viscous effects. Within the uppermost eddy (that is, the primary eddy adjacent to the translating lid), however, inertial effects are significant. The center of the primary eddy is initially observed to move to the right with increasing Reynolds number and later move downward and back towards the centerline, while the lowermost eddies remain relatively unaffected. In the case of $Re=400$ and 800 , the shape, size and location of the secondary eddy is also dependent on the Reynolds number, albeit to a lesser extent than the primary eddy. The apparent lack of dependence of the lowermost eddies on the Reynolds number, in terms of their shape and relative intensity, was predicted by Moffatt [154] in 1964. In his analytical investigation, Moffatt started with the assumption that inertial effects are negligible close enough to the sharp corner. He proceeded to derive some simple similarity solutions to the Stokes equations for a variety of different boundary conditions. Moffatt's most relevant finding pertaining to the present investigation, is that the flow near a sharp corner, induced by an arbitrary disturbance at a large distance away, consists of a sequence of eddies of decreasing sizes

and rapidly decreasing intensities as one moves towards the corner. Furthermore, he demonstrated that these “Stokes eddies” are all similar in shape and centered along the centerline. Resolving more than a few of these eddies in a Navier-Stokes numerical simulation is very difficult owing to the fact that the velocity magnitudes rapidly approach the lower limit of machine precision as one approaches the corner. Jyotsna and Vanka [152] were only able to resolve four eddies in their simulations of this problem. They used uniform grids composed of triangular elements without any adaptive refinement. In the present simulations, with the combined use of the strain-rate error estimator and the flow-angle indicator to drive the adaptive procedure, the current CVFEM was able to resolve five eddies as shown in Fig. 6.22 (c) and Fig. 6.23 (c) for $Re=50$ and 800 , respectively: The corresponding final adapted meshes are shown in (b) of each of these figures. In these simulations, the specified target errors were $\bar{\eta}_e = 0.1$ and $\bar{\eta}_\psi = 0.15$, and the minimum-area constraint was set to $(\Omega_e)_{min} / \Omega = 1 \times 10^{-5}$. Figures 6.22 (a) and 6.23 (a) show the final adapted meshes of the same simulation using *only* the strain-rate error estimator to drive the adaptive procedure with $\bar{\eta}_e = 0.1$. It is evident that the strain-rate error estimator alone is unable to properly resolve any of the lower eddies in the cavity: As was mentioned previously, this lack of effectiveness is due to the small velocity magnitudes and calculated gradients in these regions of the cavity. It should be noted, however, that the absolute error in the strain-rates is also very small in these regions. Thus, the strain-rate error estimator only functioned the way it was designed to. It detects little or no error in these regions and, therefore, does not drive the adaptive procedure to refine the mesh in that area. Regardless of the absolute error level, it may still be desirable to resolve certain flow features in regions of the domain with small velocity magnitudes and computed gradients. The flow-angle indicator is capable of functioning irrespective of the local velocity magnitudes, since it is based on the *angle* between the velocity vectors at adjacent nodes within an element (see Sections 4.2.2 and 4.2.3). Thus, it is evident from (b) and (c) in Figs. 6.22 and 6.23 that the flow-angle indicator in conjunction with the strain-rate error estimator, was successful at accurately resolving the lower eddies of the cavity while simultaneously refining close to the lid where the highest truncation error would be expected.

Flow field contour plots. Figures 6.24 and 6.25 present contour plots of the velocity components and pressure for the case of $Re=50$ and 800 , respectively. In plot (d) of each figure, the corresponding flow pattern is illustrated using a velocity vector plot. Each arrow in this plot stems from a node in the associated grid (note: only half the nodes are represented here so that there would not be an excessive number of arrows shown). The direction and relative magnitudes of these arrows represent the direction and relative magnitudes of the velocity vectors at these nodes. These adaptive-grid simulations were performed using both the strain-rate error estimator and flow-angle indicator. Specified target errors of $\bar{\eta}_e = 0.1$ and $\bar{\eta}_\psi = 0.15$, were prescribed in each case, with $(\Omega_e)_{min} / \Omega = 1 \times 10^{-5}$.

Extended Richardson extrapolations. Lid-driven flow in a triangular cavity was simulated on a series of different grids for $Re=50$ and 800 . These CVFEM solutions were then used to perform extended Richardson extrapolations (Chapter 5) for the purposes of approximating grid-independent results. The first sets of extrapolations were carried out using a hierarchy of *uniform* grids. The first grid in the hierarchy was composed of 4356 triangular elements. All of the elements in the domain were identical isosceles triangles, similar in shape to the cavity itself. Using this as the base grid, three other grids were constructed using the element subdivision technique described in Section 5.5.1. These four grids formed a hierarchy with respect to the characteristic grid size parameter, $(h_c)_k$, $k=1, 2, \dots, 4$, defined by Eq. (5.23). The finest grid in the hierarchy, corresponding to $(h_c)_4 = 1/4$, was composed of 35245 nodes and 69696 elements.

Solution-adaptive simulations were also used to define the base grids in two other sets of extended Richardson extrapolations. Two different hierarchies of five grids – one for each Reynolds number – were constructed using the aforementioned element subdivision technique (Section 5.5.1). The coarsest grids in these hierarchies resulted from prescribing target errors of $\bar{\eta}_e = 0.175$ and $\bar{\eta}_\psi = 0.4$, and an area constraint of $(\Omega_e)_{min} / \Omega = 2 \times 10^{-4}$, in the corresponding adaptive analyses. In the case of $Re=50$, the coarsest grid was composed of 1733 nodes and 3229 elements, and the finest grid

(corresponding to $(h_c)_5 = 1/5$) was composed of 40951 nodes and 80725 elements. For $Re=800$, the coarsest grid was composed of 1863 nodes and 3480 elements, and the finest grid contained 44111 nodes and 87000 elements.

Tables 6.7 and 6.8 present the computed locations of the primary and secondary eddy centres for $Re=50$ and $Re=800$, respectively. In these tables, section (a) gives the CVFEM values and corresponding extended Richardson extrapolations for the hierarchy of *uniform* grids described above. Section (b) of these tables, gives the analogous results for the hierarchy of grids based on the initial *solution-adapted* grid for each Reynolds number.

In light of the results presented in Section 6.4, the 2-grid and 3-grid extrapolations shown in Tables 6.7 and 6.8 were based on series expansions in h_c *without* the linear term, $f_1 h_c$, present (see Section 5.5.4). It was, therefore, assumed that the CVFEM exhibited essentially $O(h_c^2)$ convergence in this problem.

In general, the extrapolations based on the uniform-grid hierarchy (a), did not converge to grid-independent values as rapidly as the extrapolations based on the adaptive-grid hierarchy (b). This suggests that the solution-adaptive procedure used to arrive at the initial grid in (b) was successful at reducing and evenly distributing the measured truncation error relative to the initial grid in hierarchy (a). This allowed successive grids in hierarchy (b) to enter into the asymptotic region of grid convergence quicker than the corresponding grids in hierarchy (a).

Comparison of Results. In Fig. 6.26, the results obtained in this work are compared with the published solutions of Jyotsna and Vanka [152]. The x -component of velocity, u , along the centerline ($x=0$) of the cavity is plotted (horizontal axis), versus y (vertical axis), for (a) $Re=50$ and (b) $Re=800$. The “present adaptive-grid” u -profiles plotted in these graphs correspond to grid 5 in Tables 6.7 and 6.8 (b). The “present extrapolated” u -profiles correspond to the 3-grid extrapolation using grids 3-5 also shown in Tables 6.7 and 6.8 (b).

Table 6.9 presents comparisons of the computed eddy centre locations for the primary and secondary eddies for $Re=50$ and $Re=800$. The present values in this table were

obtained from the 3-grid extrapolations, using grids 3-5, shown in Tables 6.7 (b) and 6.8 (b), unless the corresponding CVFEM values exhibited oscillatory convergence with respect to h_c , in which case these values were obtained from grid 5.

Jyotsna and Vanka [152] used a control-volume finite element method, similar to the present method described in Chapter 2, to obtain their solutions to the lid-driven flow problem in a triangular cavity. There were two major differences between their method and the current CVFEM, however: (i) they did not employ any solution-adaptive grid-refinement techniques in their numerical simulations; and (ii) they used linear, element-based interpolation functions to approximate the velocity components in the advection terms of the related integral conservation equations. It is well known [60,82] that linear interpolation of the dependent variables in the advection terms could lead to physically unrealistic oscillatory solutions and cause iterative solution procedures to diverge, unless special care is taken. It is assumed here that Jyotsna and Vanka were indeed careful in this respect. Jyotsna and Vanka also performed Richardson extrapolations in their work, however, they only used *two* uniform grids in their extrapolations, and the finest grid they employed was comprised of only 49152 elements, compared with an h -adaptive grid with as many as 87000 elements in the present simulations. Furthermore with only two grids, they were unable to confirm whether their solutions were within the asymptotic region of grid convergence. If their solutions were suffering from oscillatory convergence with respect to grid size, then the 2-grid Richardson extrapolations they employed might not be applicable. In fact, any results obtained from such an extrapolation could conceivably be less accurate than the original solutions themselves. In spite of these uncertainties, most of their results were in good agreement with the present results.

Comparisons between the strain-rate error estimator and flow-angle indicator. The last sets of tests performed for the lid-driven flow in a triangular cavity were intended to illustrate the individual capabilities of the strain-rate error estimator (SREE) and the flow-angle indicator (FAI) in driving the h -adaptive procedure. A relatively coarse, uniform grid, composed of 1452 nodes and 2680 elements, was used as the *initial* grid in a series of h -adaptive simulations in which the SREE and FAI were used, independently,

for several different values of their associated target errors, $\bar{\eta}_e$ and $\bar{\eta}_\psi$. In each case, the minimum element-area-constraint was set to $(\Omega_e)_{\min} / \Omega = 5 \times 10^{-5}$.

Figure 6.27 (a)-(f) shows the final adapted meshes resulting from a series of simulations using the SREE only, to guide the adaptive refinement process. Grids (a)-(c) in this figure correspond to lid-driven cavity simulations for $Re=50$, and grids (d)-(f) correspond to simulations for $Re=800$. The prescribed value of the target error, total number of nodes, and total number of elements corresponding to each grid are indicated in the figure, next to the particular grid. The three values of $\bar{\eta}_e$ considered in these tests, for each Reynolds number, were $\bar{\eta}_e = 0.2, 0.15$, and 0.1 . In the case of $Re=50$, the total number of elements ranged from 4264 for $\bar{\eta}_e = 0.2$, to 11826 elements for $\bar{\eta}_e = 0.1$. In the case of $Re=800$ it ranged from 5074 elements for $\bar{\eta}_e = 0.2$, to 17407 elements for $\bar{\eta}_e = 0.1$.

As was discussed earlier, the SREE is sensitive to large strain-rates or shear stresses by construction. In the case of lid-driven flow in a triangular cavity, the largest strain-rates occur near the translating lid, and, in particular, in the vicinity of the two upper corners, where the velocity is mathematically discontinuous. In the lower portion of the cavity, however, the magnitude of the velocity, and the computed strain-rates, decrease rapidly as the lower corner is approached. As expected, the SREE was only able to refine the mesh near the lid. More precisely, it detected the regions of high strain-rates and refined the mesh appropriately. In fact, the final adapted mesh (produced by the SREE), itself, provides an illustration of the computed strain-rate distribution for the corresponding Reynolds number (albeit a rather crude one), especially in the case of grid (c) and grid (f). The main disadvantages in using the SREE alone, in the context of h -adaptive Navier-Stokes simulations, is that (i) there is no guarantee that a rigorous mathematical relationship exists between the truncation error in the computed solution (that is, in the u , v , and p distributions) and the energy norm of the error in the strain-rates, except in the limit as $Re \rightarrow 0$ (see Section 4.2.2); and (ii) the SREE may not be able to resolve certain flow features, such as the recirculating eddies in the bottom portion of the triangular cavity, where the velocity magnitudes and computed strain-rates are very small.

Figure 6.28 (a)-(f) is analogous to Fig. 6.27 (a)-(f), except that in this case, the FAI was used, independently, to drive the solution-adaptive procedure. The values of $\bar{\eta}_\psi$ used in these tests were $\bar{\eta}_\psi = 0.3, 0.2$, and 0.1 , for each Reynolds number. While the SREE enjoys a rigorous mathematical relationship to the truncation error in the context of simulating creeping flow problems ($Re \rightarrow 0$) [16,22], the FAI may not possess any such relationship, especially in the context of Navier-Stokes flows. However, one advantage the FAI has over the SREE is its ability to resolve flow features anywhere in the domain, irrespective of the local velocity magnitudes or computed strain-rates (see Section 4.2.3). The FAI was, therefore, successful in refining the grid in the bottom portion of the triangular cavity, where a sequence of “Stokes eddies” occur, as shown in Fig. 6.28 (a)-(f).

In the present work, the SREE and the FAI were combined (see Section 4.3.4) in the h -adaptive simulations of Navier-Stokes flows. This was done in an effort to reduce the truncation error in the computed solution, for low to moderate Reynolds numbers, while simultaneously resolving certain flow features of interest that may not have otherwise been resolved using the SREE alone.

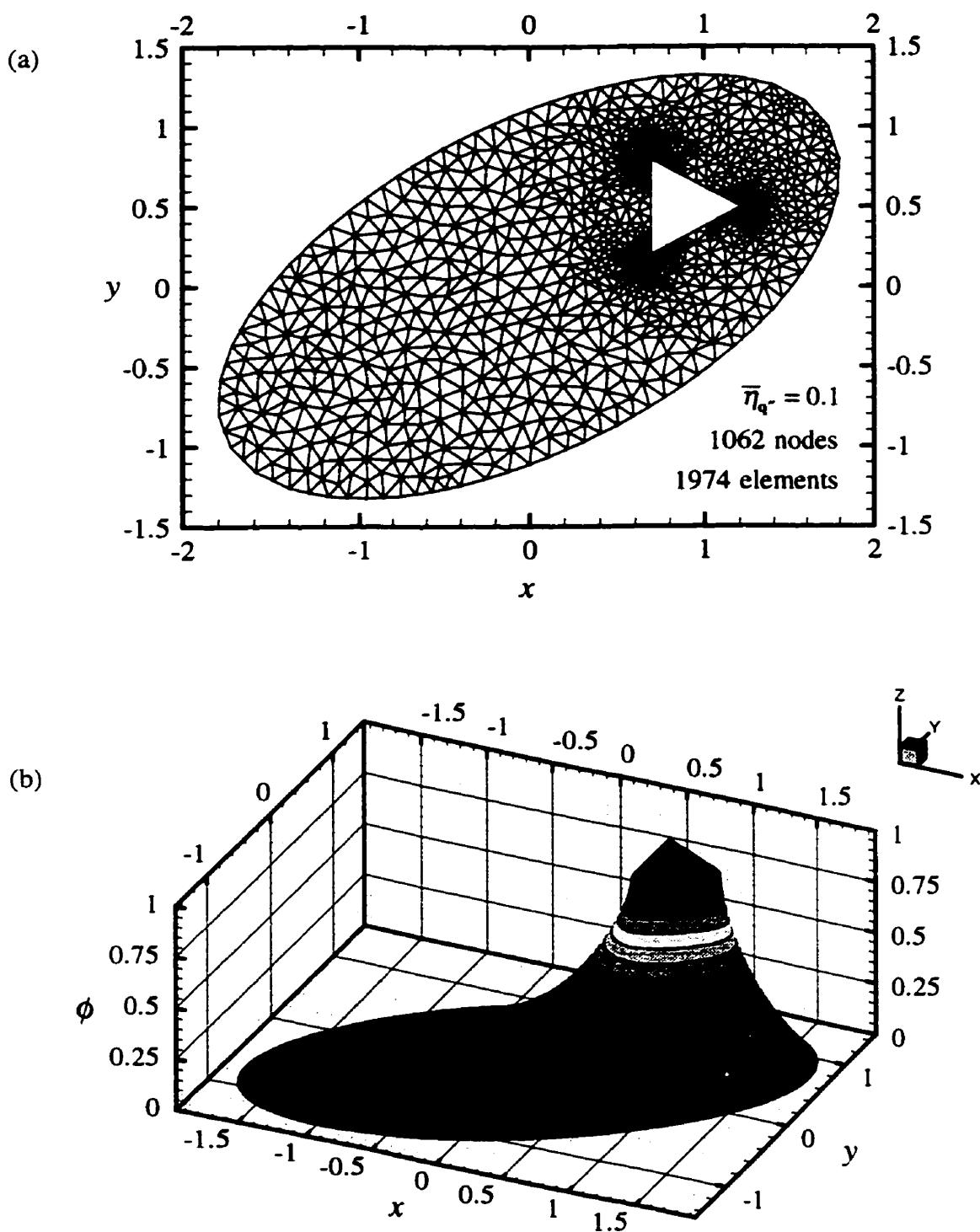


Figure 6.1: Heat conduction in an elliptically shaped domain with an asymmetrically located triangular hole: (a) Final h -adapted mesh; and (b) surface plot of the computed temperature distribution.

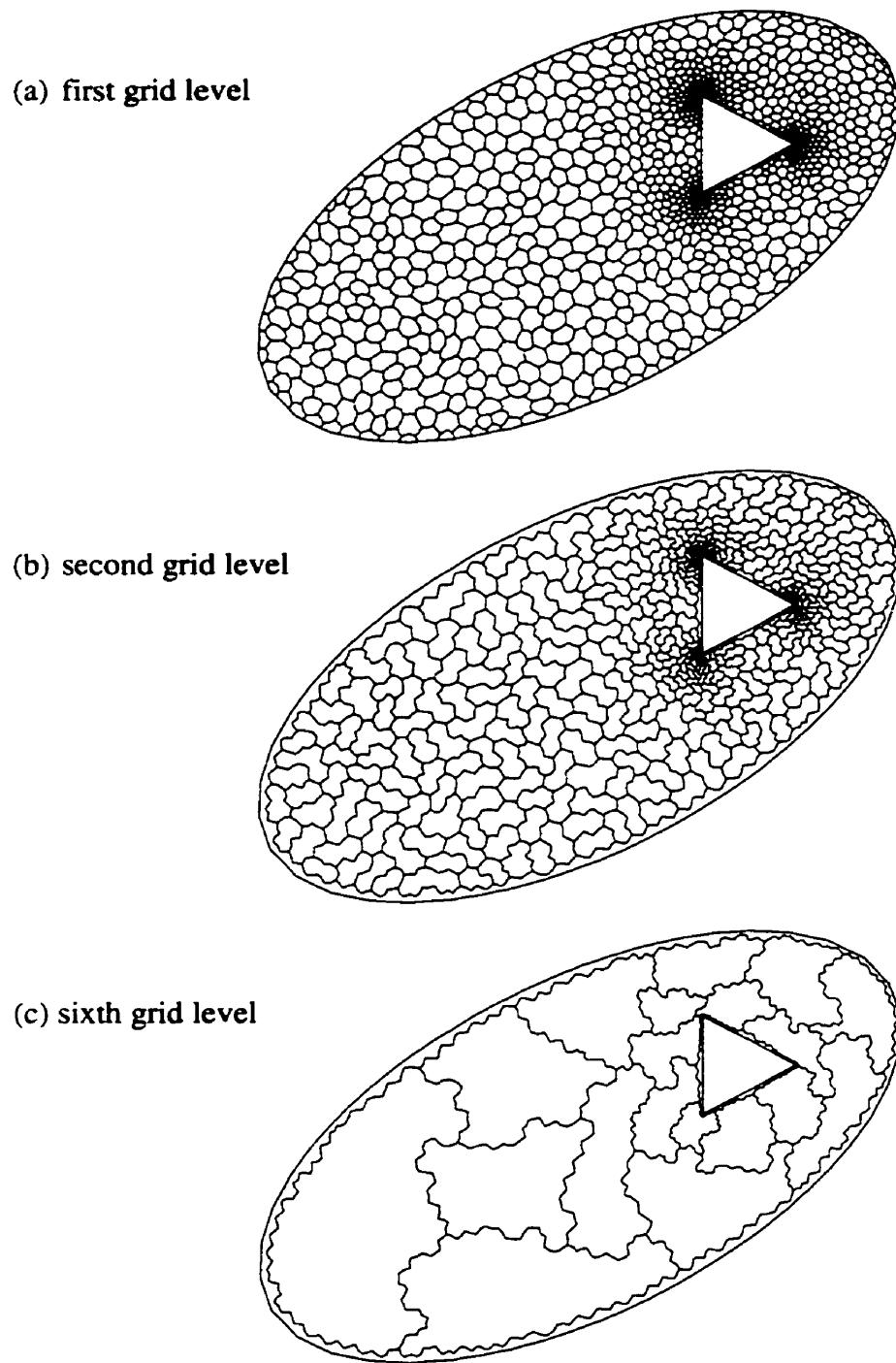


Figure 6.2: Heat conduction in an elliptically shaped domain with an asymmetrically located triangular hole. Control volumes corresponding to 3 of the 6 grid levels in a typical ACM cycle: (a) the first grid level (the centroid-median tessellation of the original finite element mesh, shown in Fig. 6.1 (a)); (b) the second grid level; and (c) the sixth and final grid level.

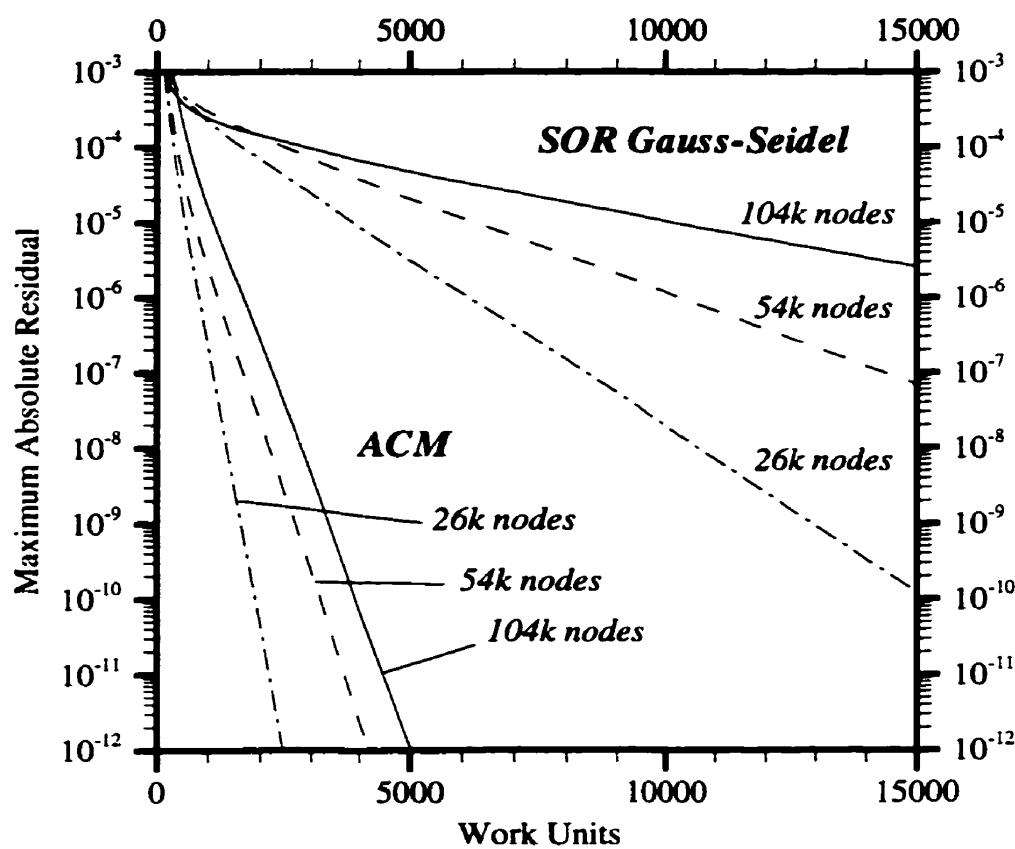


Figure 6.3: Heat conduction in an elliptically shaped domain with an asymmetrically located triangular hole. Comparison of the convergence histories of the ACM and SOR-GS solvers on three different grids.

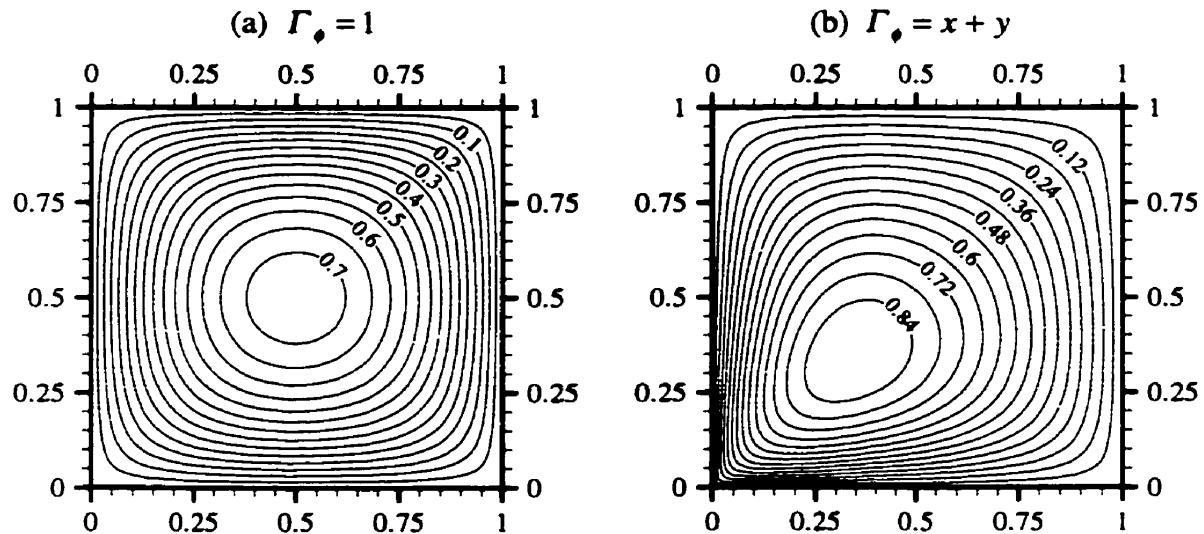


Figure 6.4: Solution to the Poisson equation on a Square domain. Contour plots of ϕ for the case of: (a) $\Gamma_\phi = 1$; and (b) $\Gamma_\phi = x + y$.

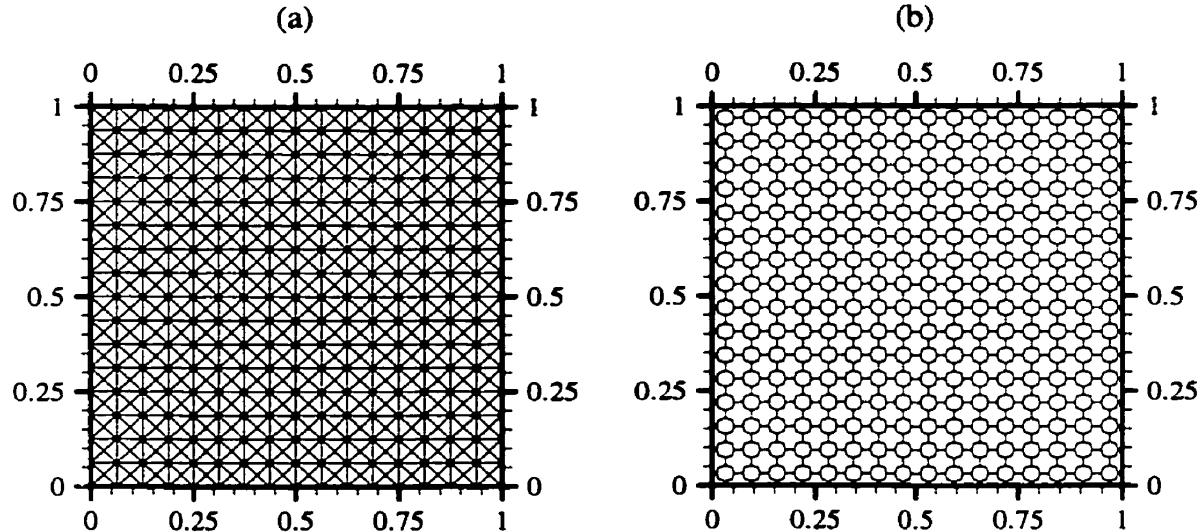


Figure 6.5: Solution to the Poisson equation on a Square domain: (a) uniform, initial grid used for testing the adaptive volume-agglomeration algorithm; and (b) corresponding control-volume boundaries (that is, the centroid-median tessellation of grid (a)).

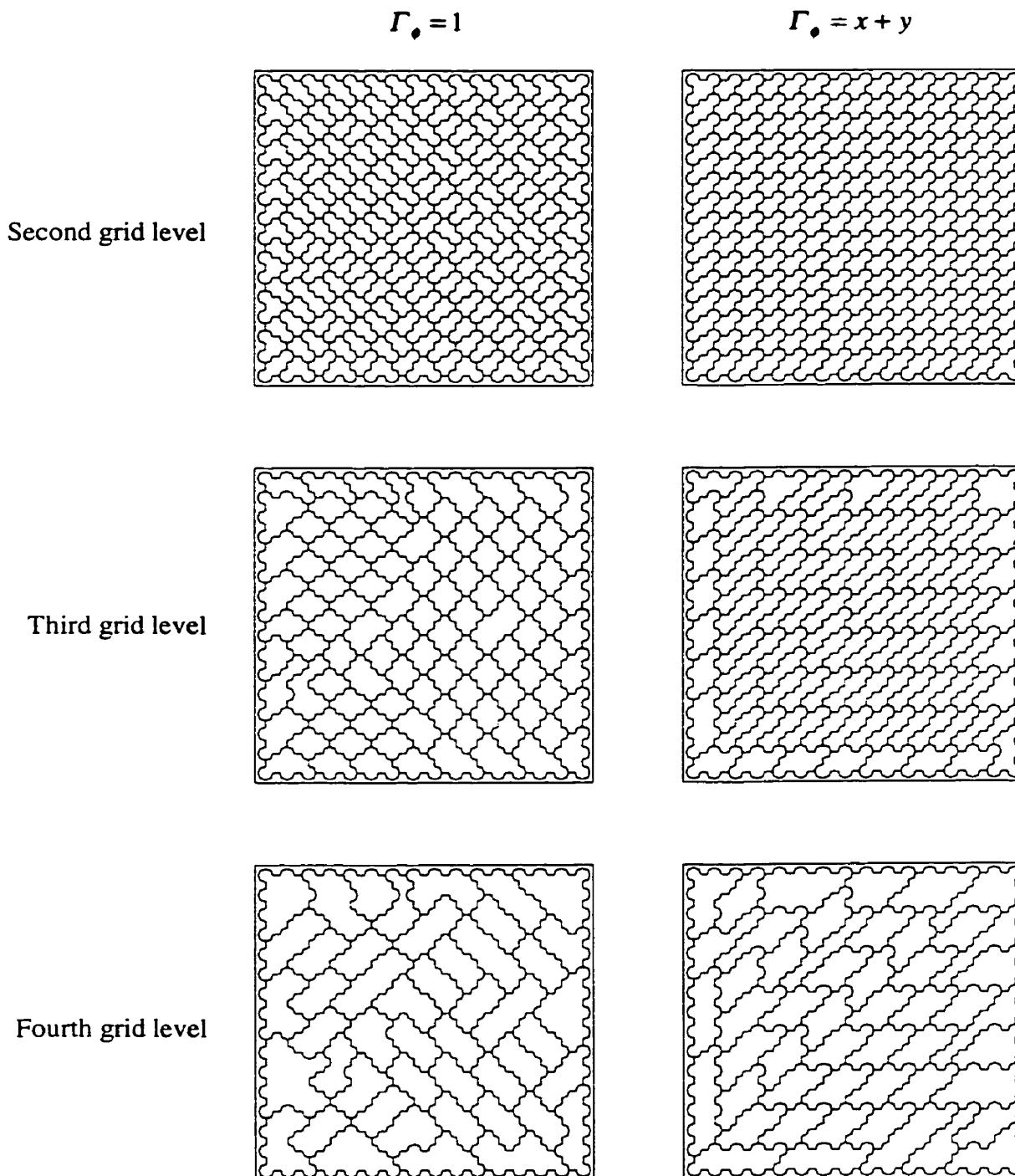


Figure 6.6: Solution to the Poisson equation on a Square domain. Agglomeration patterns in the second, third, and fourth ACM grid levels for the case of $\Gamma_\bullet = 1$ (left column); and $\Gamma_\bullet = x + y$ (right column).

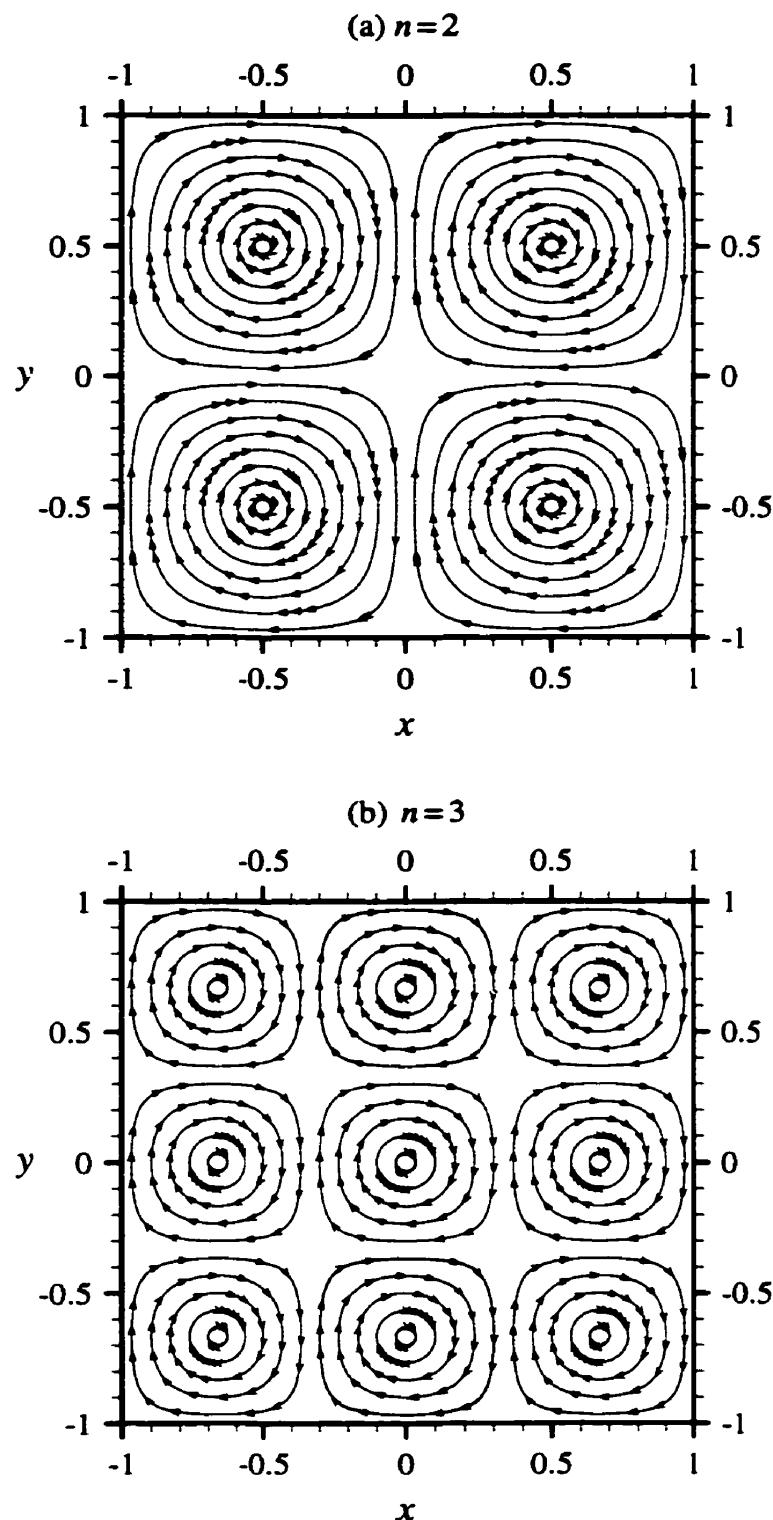


Figure 6.7: Analytical solution to the Navier-Stokes equations with body forces: Streamline patterns for the case of (a) $n=2$; and (b) $n=3$. Other parameters in both cases are: $V_o < 1$; $u_o = v_o = 0$; and $l = 1$.

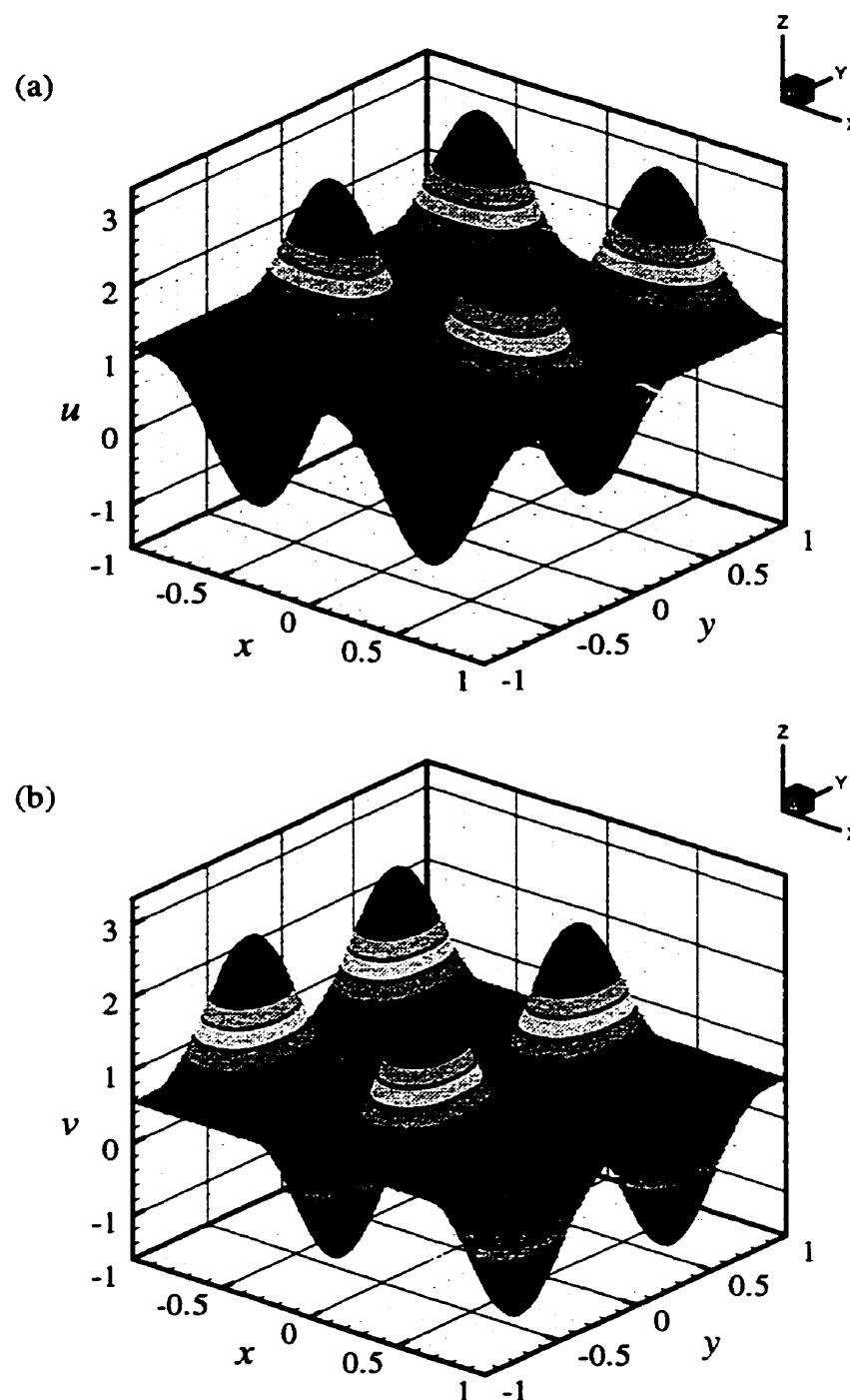


Figure 6.8: Analytical solution to the Navier-Stokes equations with body forces: Surface plots of the velocity components, (a) u and (b) v , for the case of $n=2$; $V_o = -1$; $u_o = 1.1468$; $v_o = 0.5590$; and $l=1$.

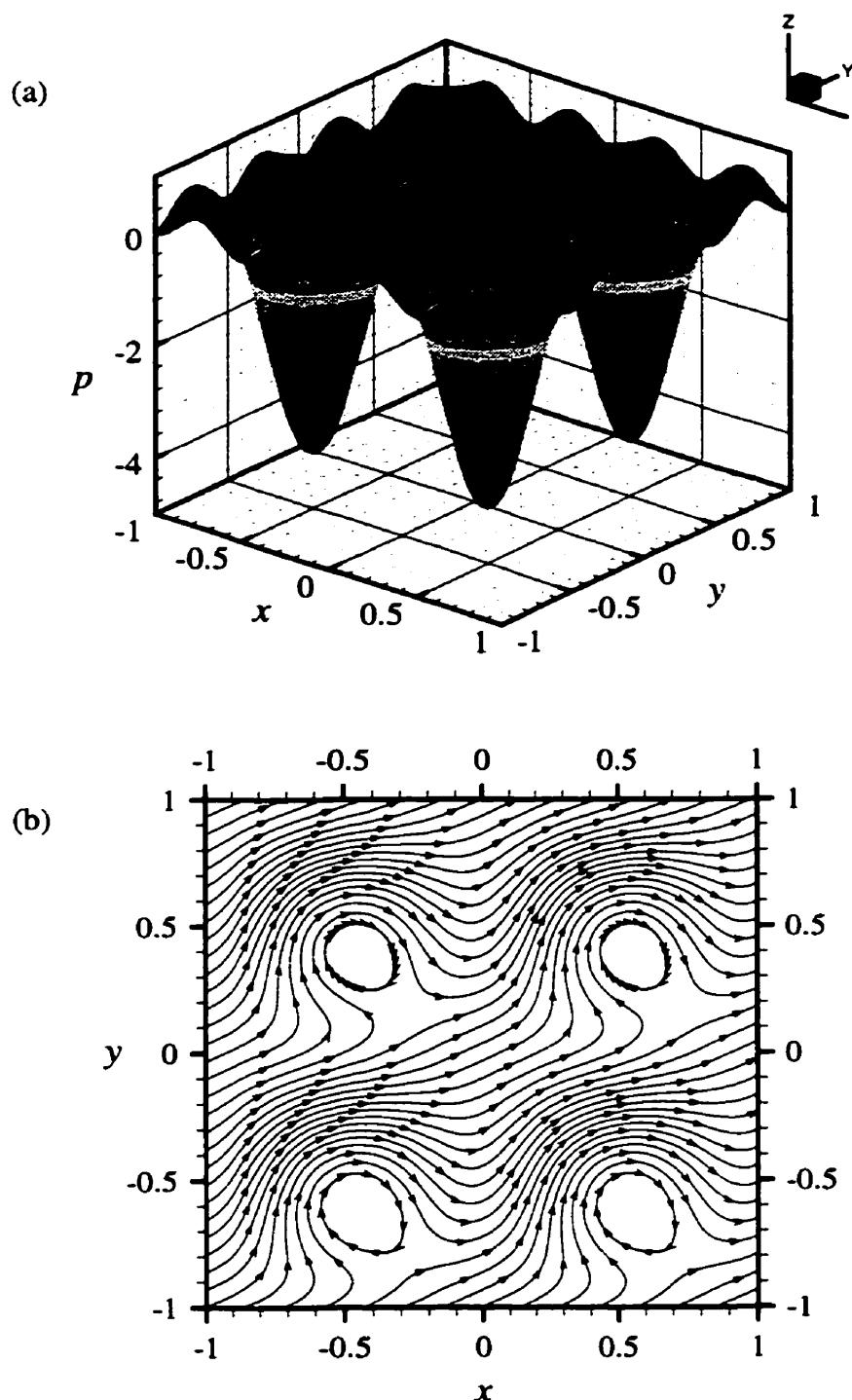


Figure 6.9: Analytical solution to the Navier-Stokes equations with body forces: (a) Surface plot of the pressure field; and (b) the streamline pattern, for the case of $n=2$; $V_o = -1$; $u_o \approx 1.1468$; $v_o \approx 0.5590$; and $l = 1$.

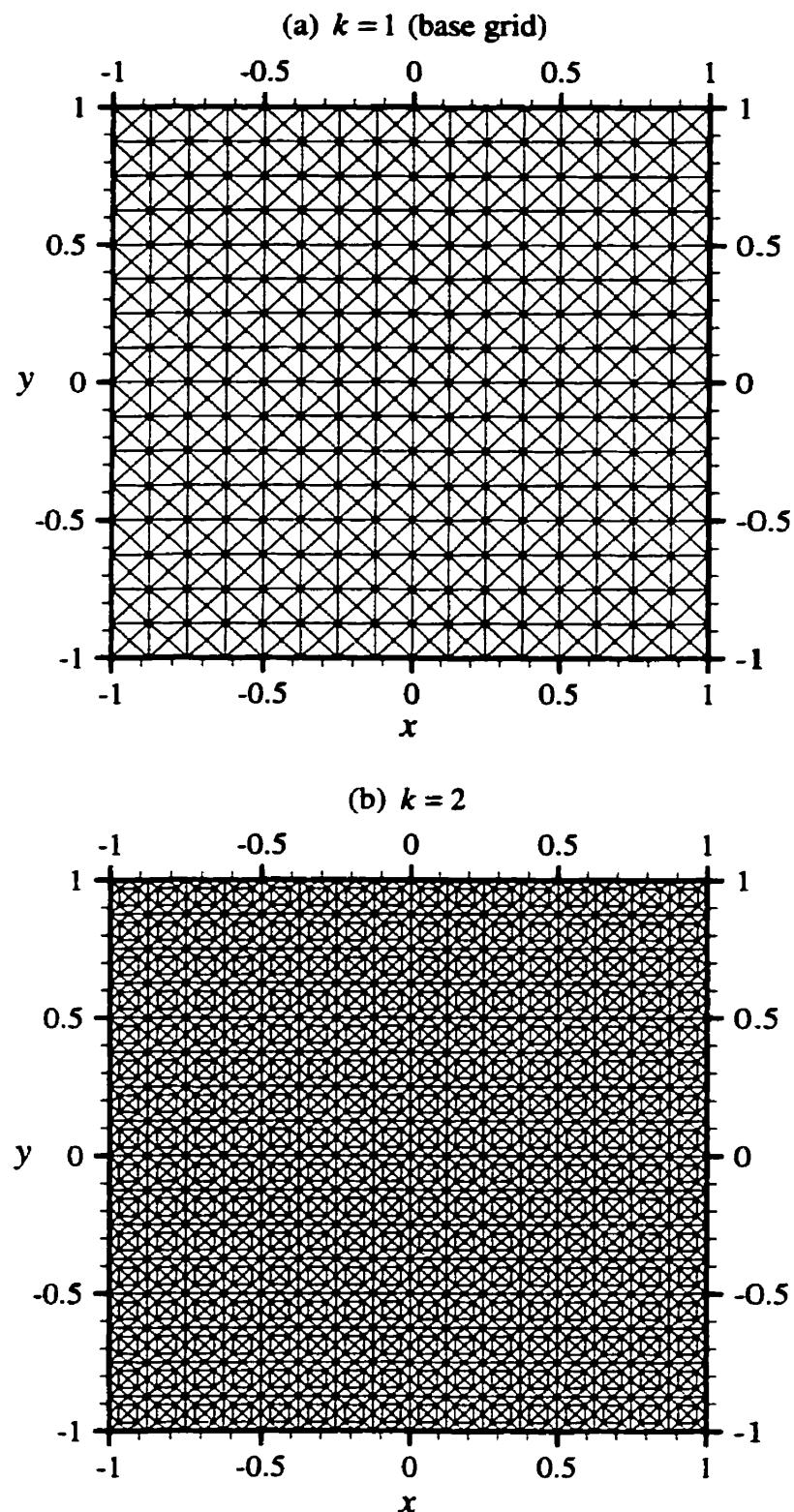


Figure 6.10: Analytical solution to the Navier-Stokes equations with body forces: (a) The base grid ($k = 1$); and (b) the second grid ($k = 2$), used in the simulations described in Section 6.2.4. There were 10 grids [$k = 1, 2, \dots, 10$] in all.

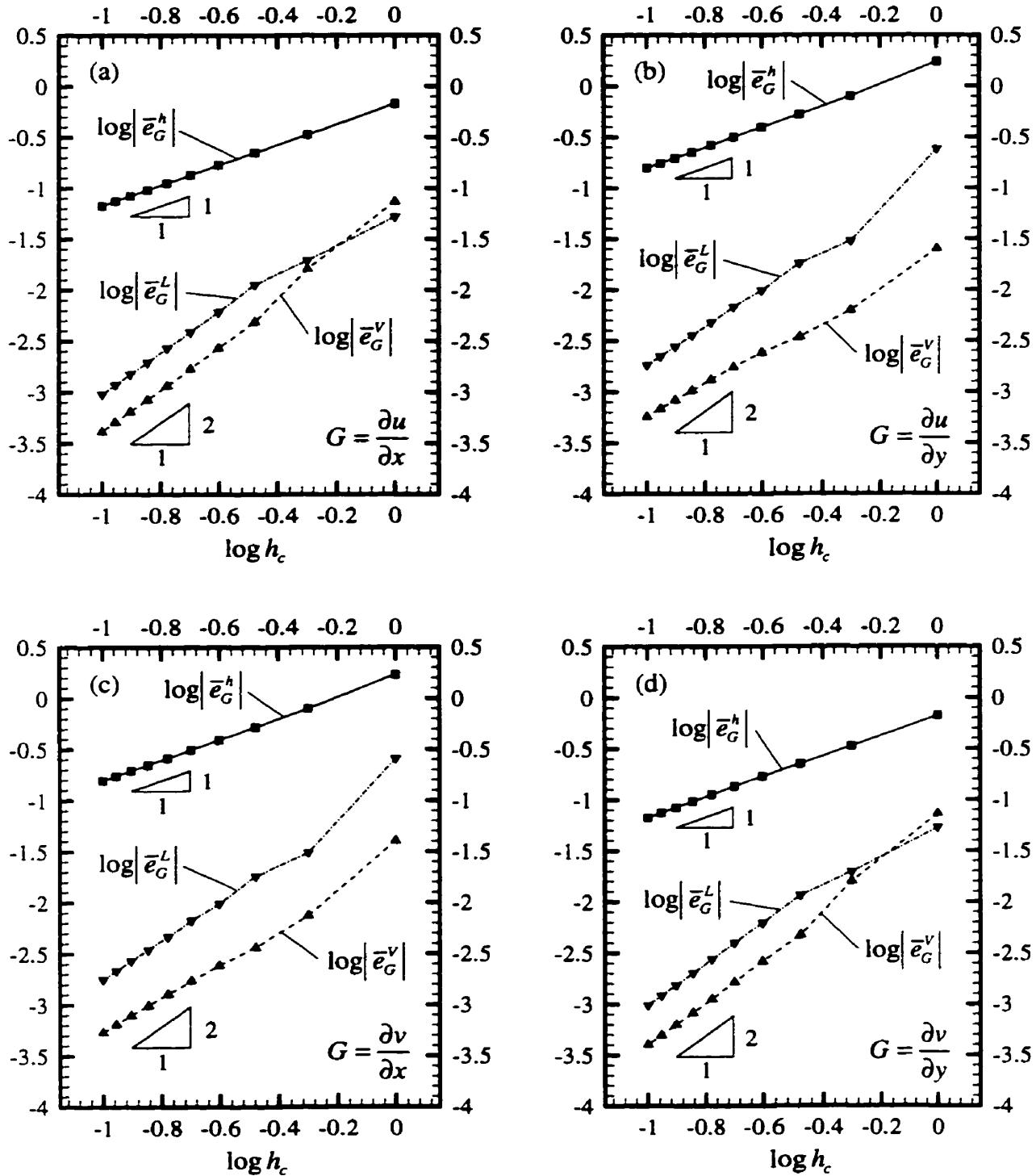


Figure 6.11: Verification of the h -adaptive procedure. Convergence histories of the relative error in the CVFEM gradients, \bar{e}_G^h , and the recovered gradients, \bar{e}_G^L (global L_2 projection) and \bar{e}_G^V (volume averaging), with respect to the to the characteristic grid-size parameter, h_c . These plots correspond to the following gradients: (a) $G = \partial u / \partial x$; (b) $G = \partial u / \partial y$; (c) $G = \partial v / \partial x$; (d) $G = \partial v / \partial y$, at the nodal location, $(x, y) = (0.125, 0.125)$. The Reynolds number in this case is $Re = 1$.

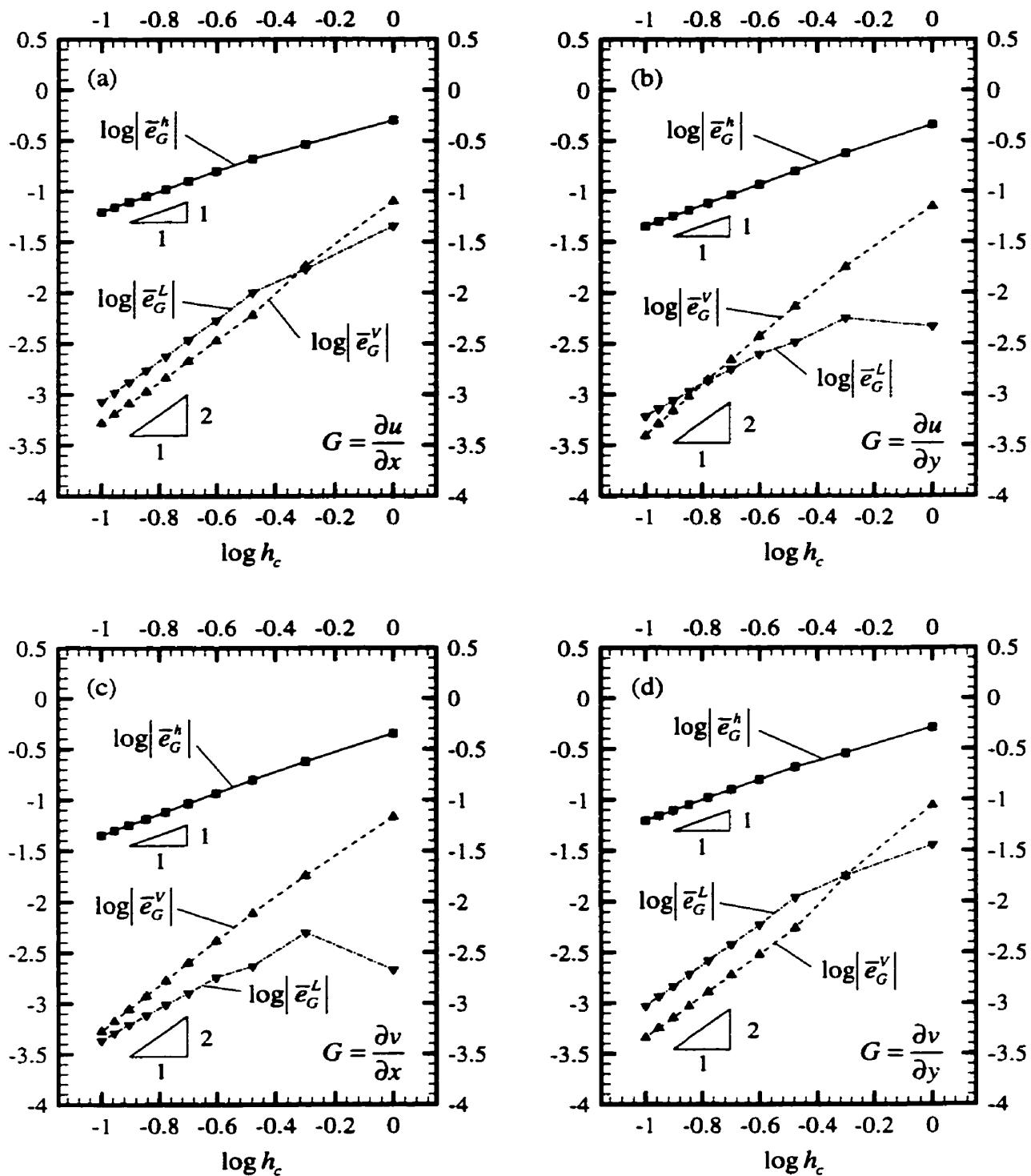


Figure 6.12: Verification of the h -adaptive procedure. Convergence histories of the relative error in the CVFEM gradients, \bar{e}_G^h , and the recovered gradients, \bar{e}_G^L (global L_2 projection) and \bar{e}_G^V (volume averaging), with respect to the characteristic grid-size parameter, h_c . These plots correspond to the following gradients: (a) $G = \frac{\partial u}{\partial x}$; (b) $G = \frac{\partial u}{\partial y}$; (c) $G = \frac{\partial v}{\partial x}$; (d) $G = \frac{\partial v}{\partial y}$, at the nodal location, $(x, y) = (0.375, 0.625)$. The Reynolds number in this case is $Re = 1$.

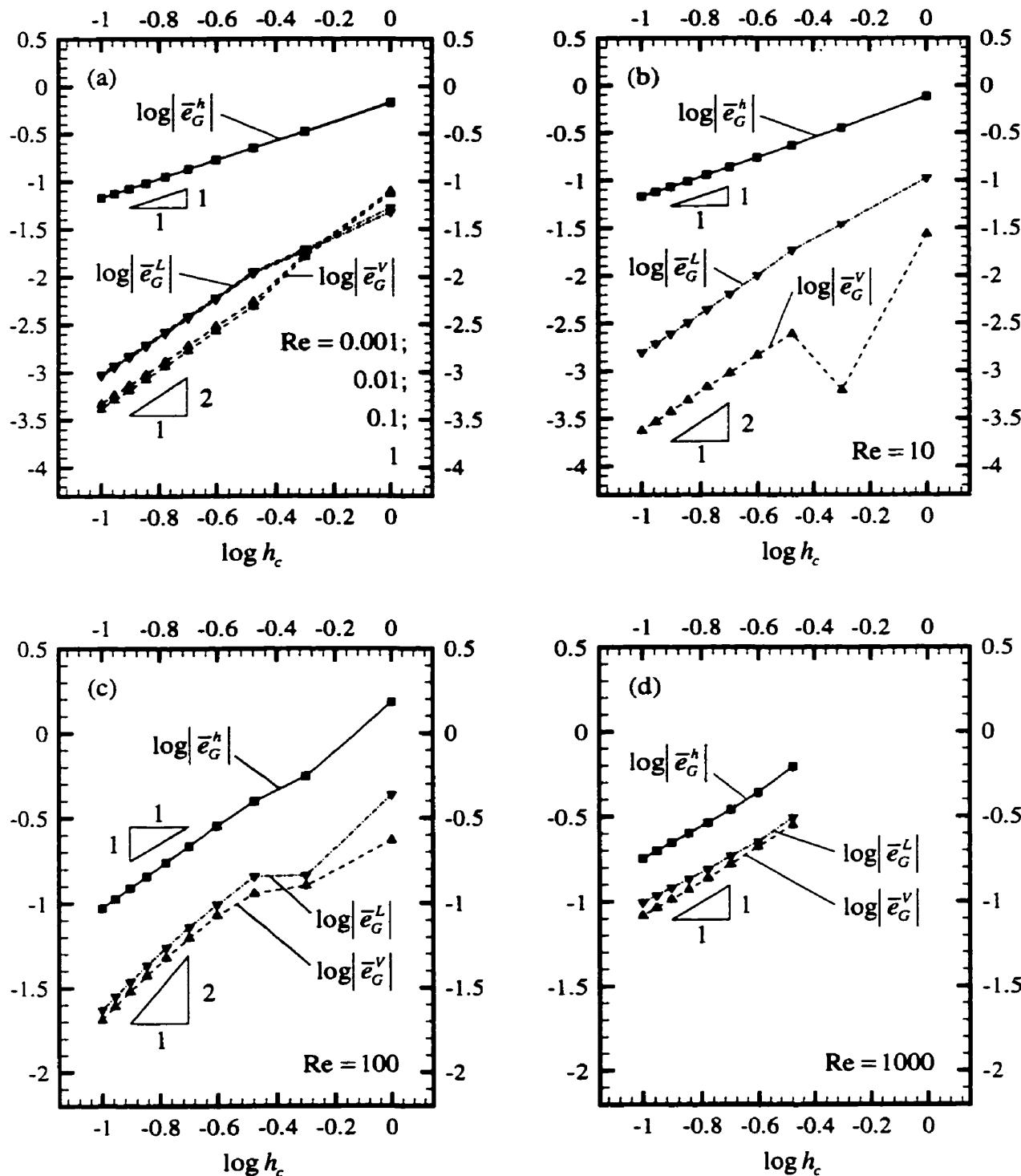


Figure 6.13: Verification of the h -adaptive procedure. Convergence histories of the relative error in the CVFEM gradients, \bar{e}_G^h , and the recovered gradients, \bar{e}_G^L (global L_2 projection) and \bar{e}_G^V (volume averaging), with respect to the characteristic grid-size parameter, h_c . These plots correspond to the gradient, $G = \partial u / \partial x$, at the nodal location, $(x, y) = (0.125, 0.125)$, for the following Reynolds numbers: (a) $Re = 0.001, 0.01, 0.1$, and 1; (b) $Re = 10$; (c) $Re = 100$; (d) $Re = 1000$.

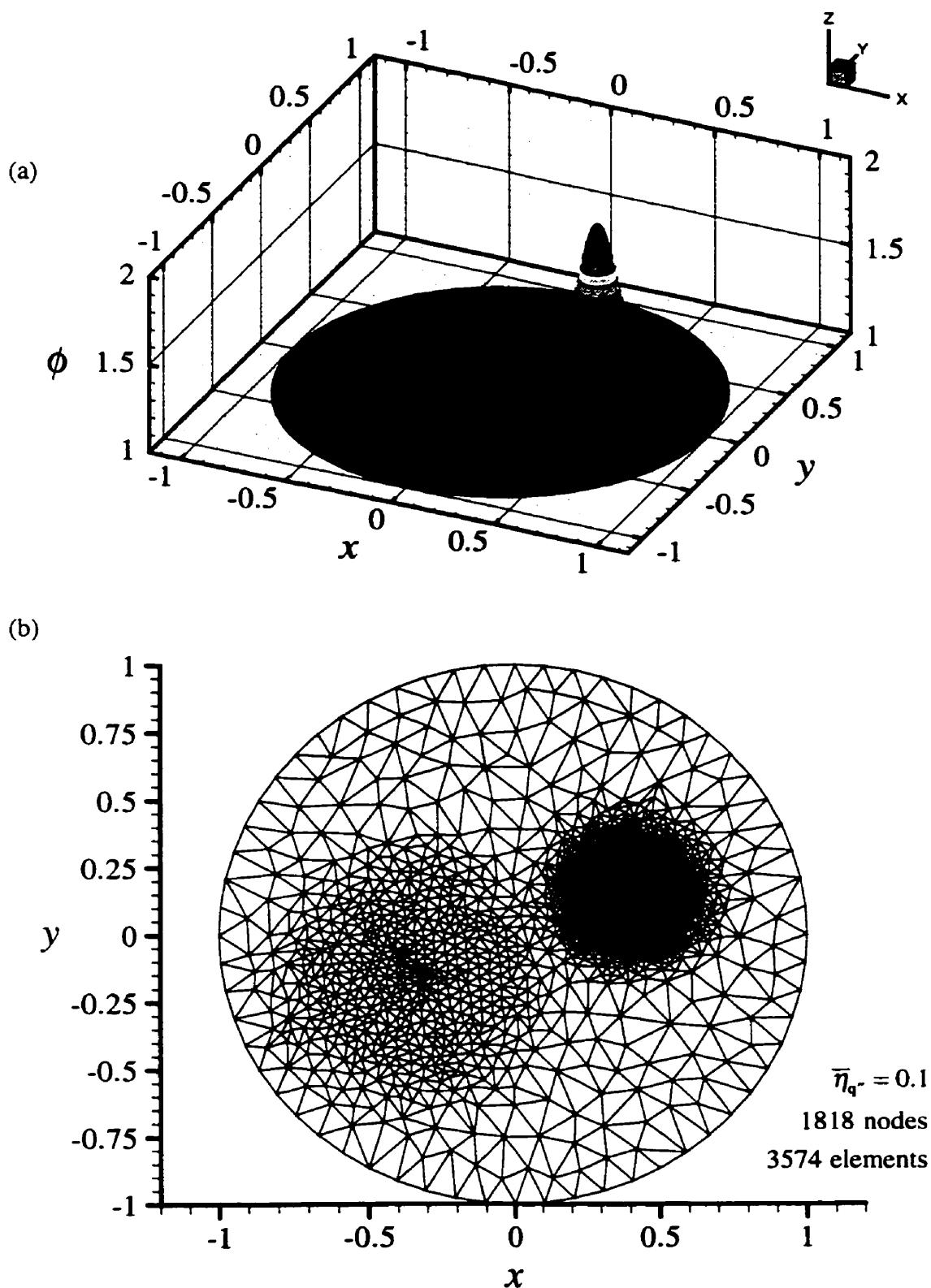


Figure 6.14: Poisson equation on a circular domain with prescribed Gaussian distributions: (a) Surface plot of the computed ϕ distribution; and (b) the corresponding h -adapted mesh.

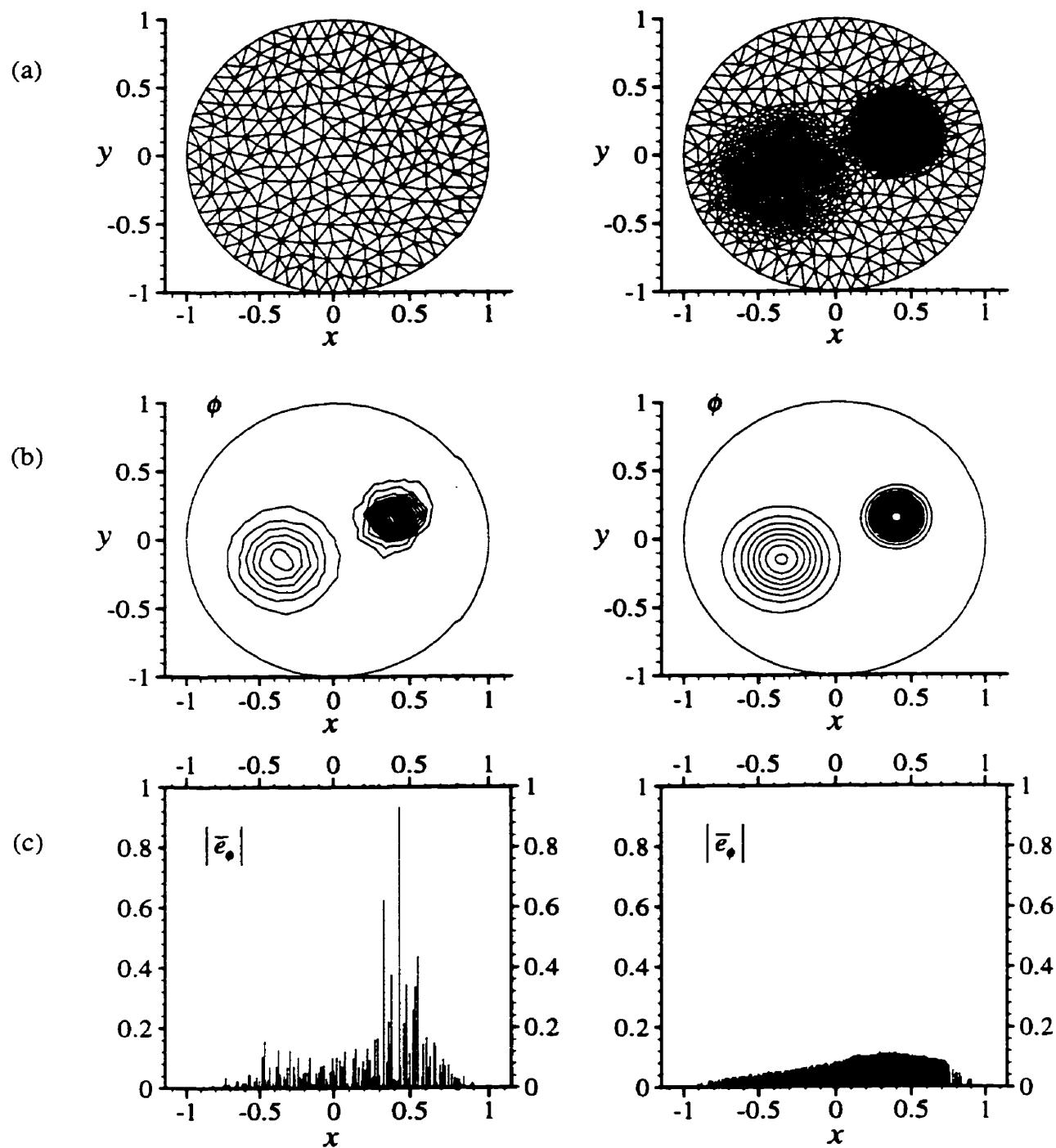


Figure 6.15: Poisson equation on a circular domain with prescribed Gaussian distributions: Comparison of initial coarse-grid solution (left column) and final h -adaptive solution (right column): (a) computational grids; (b) corresponding contour plots of the computed solution, ϕ ; and (c) bar chart of the relative error distribution, $|\bar{e}_\bullet|$, in the domain (note: the length of each bar is proportional to the relative error at a node; the y-axis points into the page).

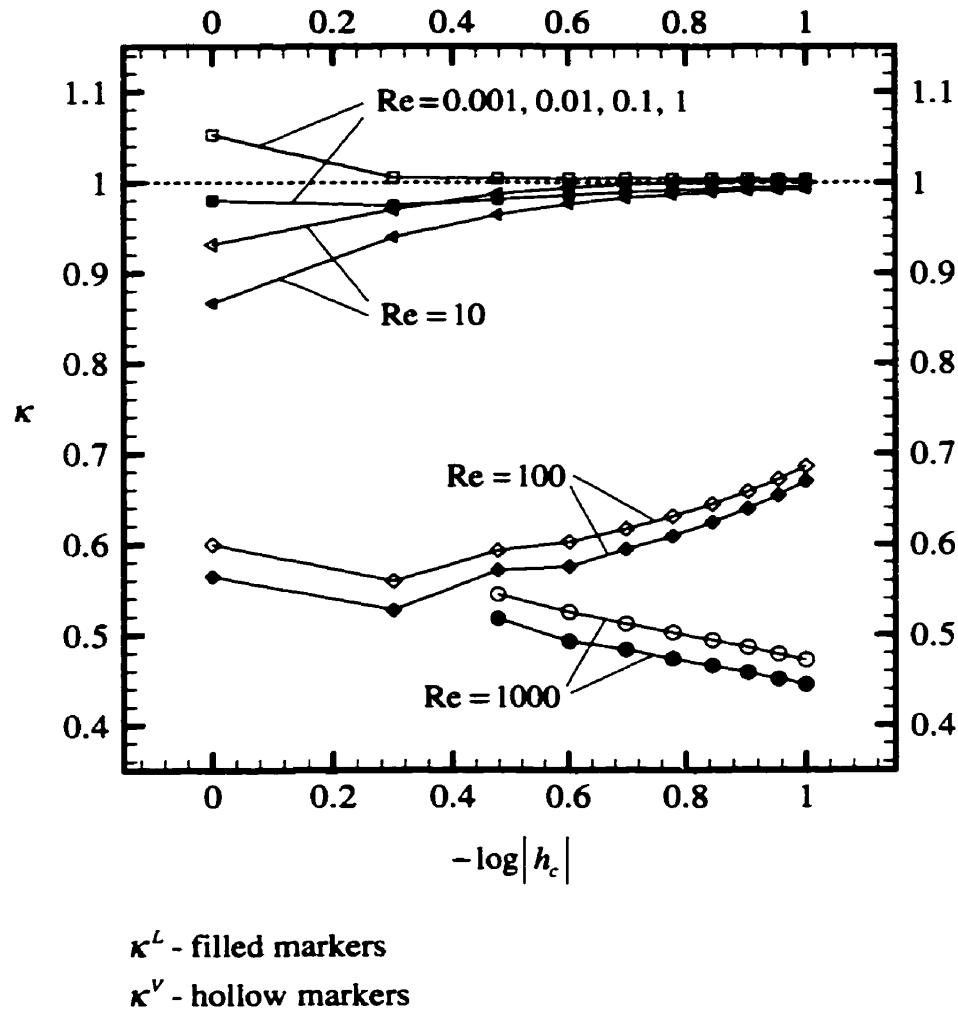


Figure 6.16: Verification of the strain-rate error estimator: Convergence histories of the effectivity index, κ , with respect to the characteristic grid-size, h_c , using the global L_2 projection (κ^L -filled markers); and volume averaging (κ^V -hollow markers), as the recovery schemes. Results are presented for the following Reynolds numbers: $Re=0.001, 0.01, 0.1, 1, 10, 100, and }1000.$

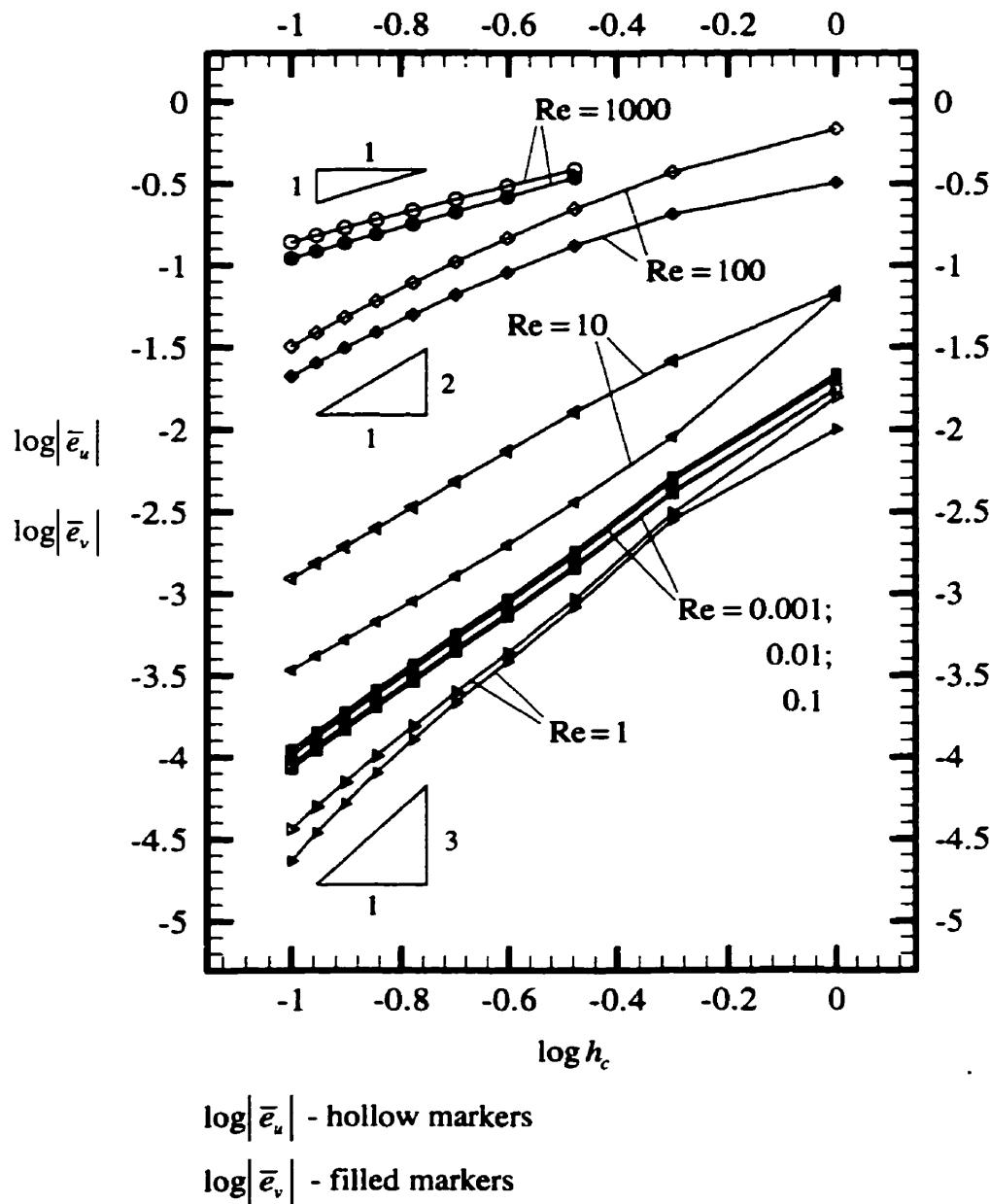


Figure 6.17: Verification of the extended Richardson extrapolation technique: Convergence histories of the relative error in the velocity components (\bar{e}_u - hollow markers; \bar{e}_v - filled markers), with respect to the characteristic grid-size, h_c , at the nodal location, $(x, y) = (0.125, 0.125)$. Results are presented for the following Reynolds numbers: $Re = 0.001, 0.01, 0.1, 1, 10, 100$, and 1000 .

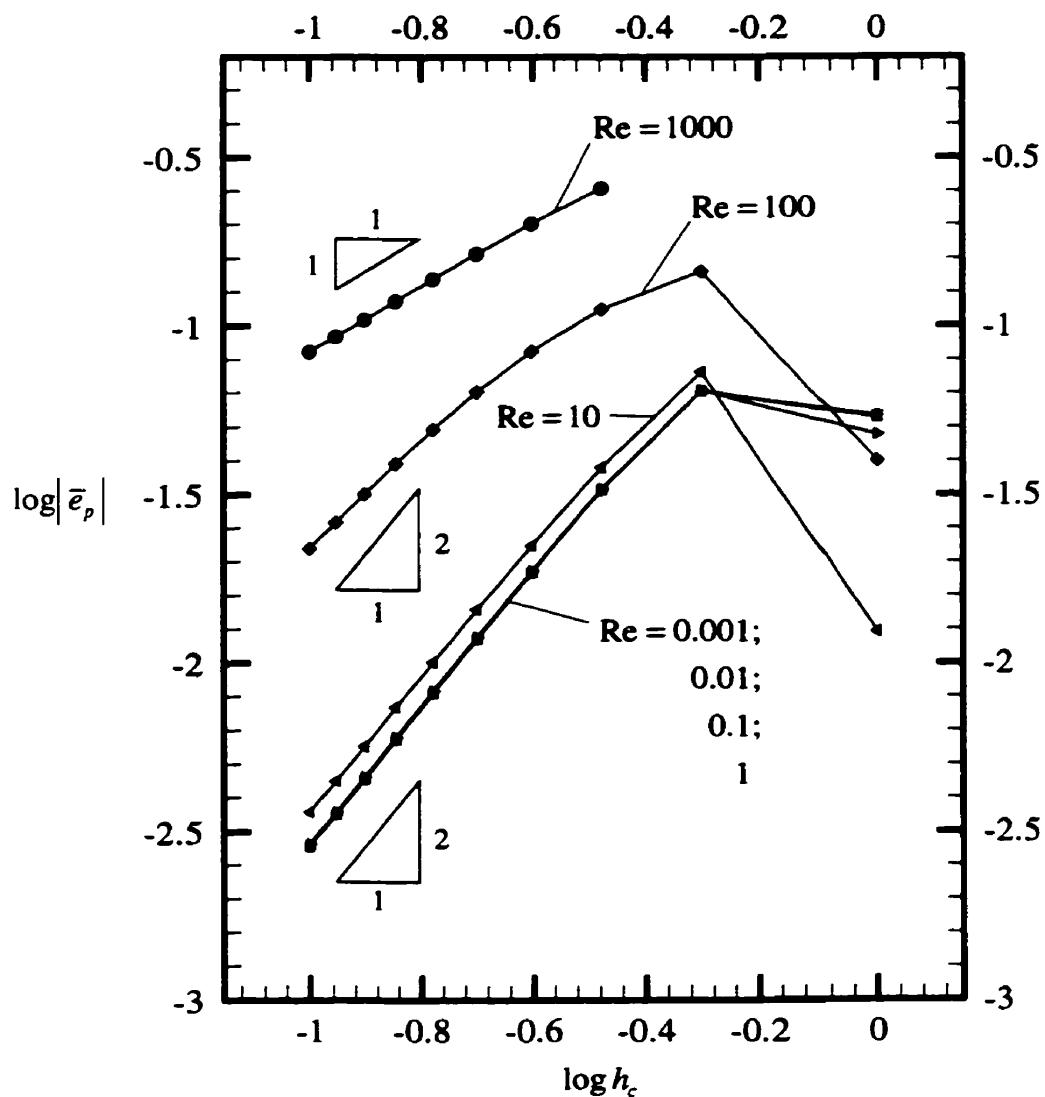


Figure 6.18: Verification of the extended Richardson extrapolation technique: Convergence histories of the relative error in the pressure, \bar{e}_p , with respect to the characteristic grid-size, h_c , at the nodal location, $(x, y) = (0.125, 0.125)$. Results are presented for the following Reynolds numbers: $Re = 0.001, 0.01, 0.1, 1, 10, 100$, and 1000 .

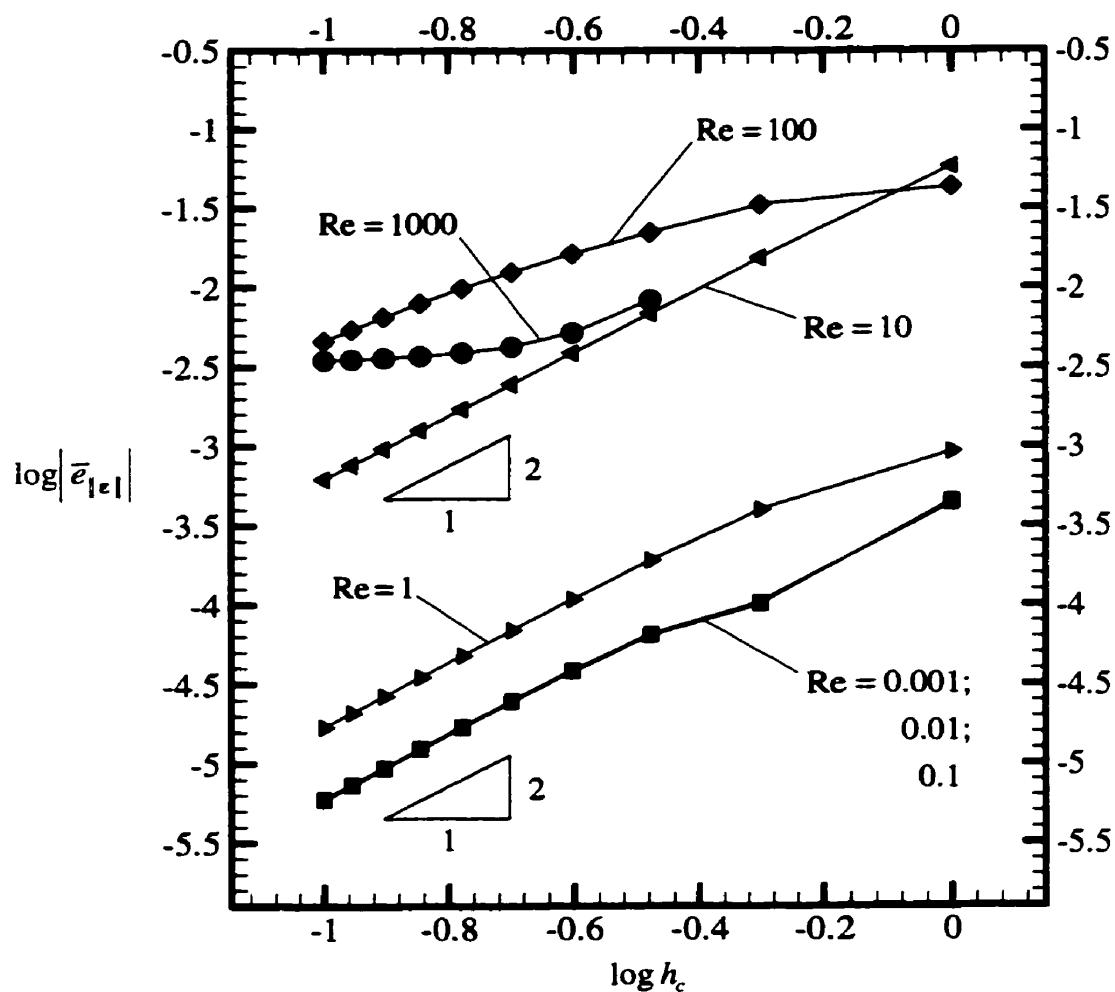


Figure 6.19: Verification of the extended Richardson extrapolation technique: Convergence histories of the relative error in the global strain-rate energy norm, $\bar{e}_{|\epsilon|}$ with respect to the characteristic grid-size, h_c . Results are presented for the following Reynolds numbers: $Re=0.001, 0.01, 0.1, 1, 10, 100$, and 1000 .

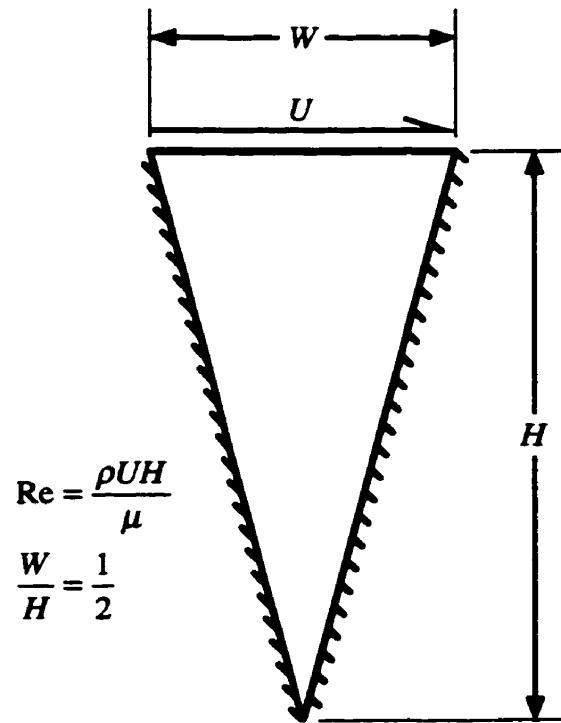


Figure 6.20: Lid-driven flow in a triangular cavity: Schematic representation of the geometry, and related nomenclature.

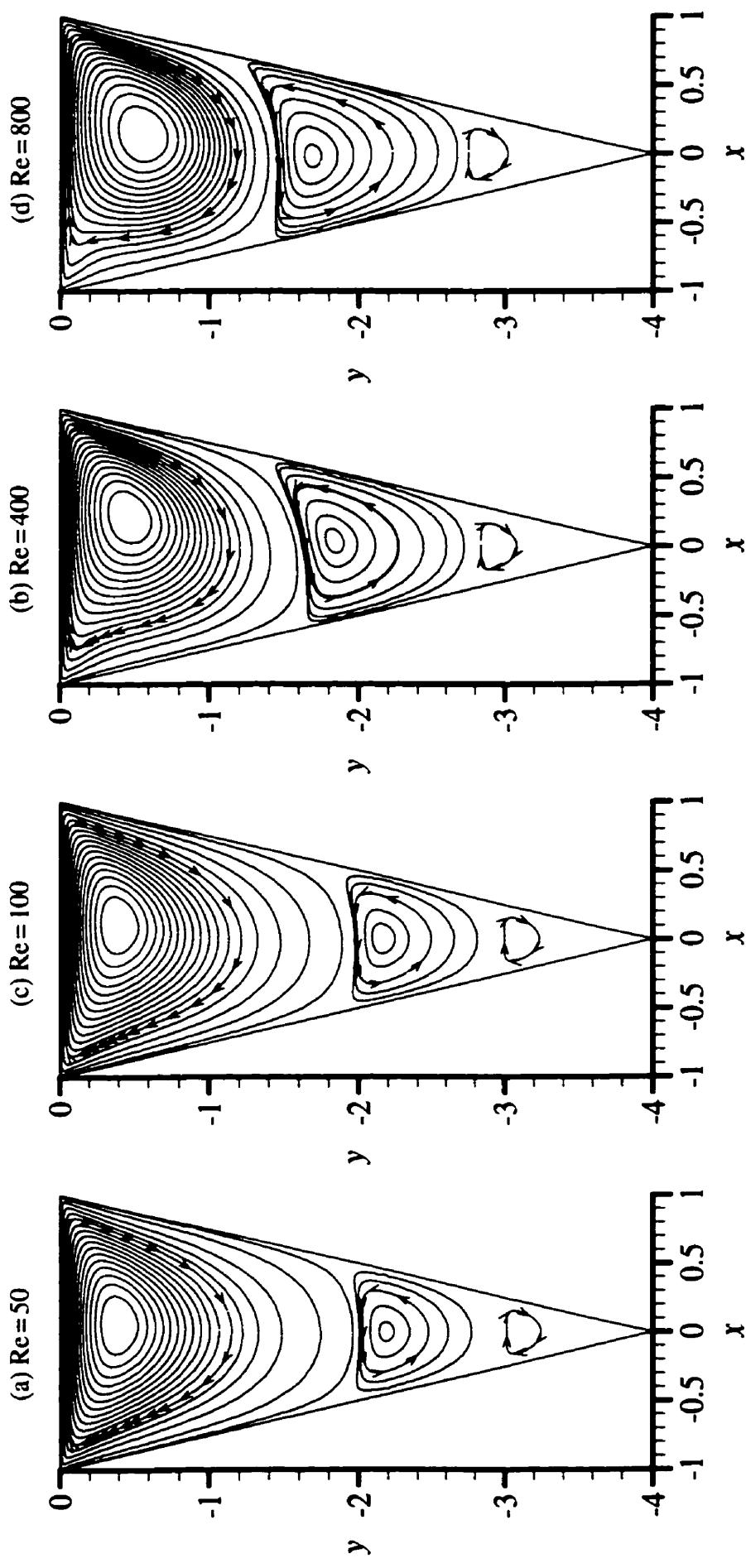


Figure 6.21: Lid-driven flow in a triangular cavity: Computed streamline patterns for (a) $Re = 50$; (b) $Re = 100$; (c) $Re = 400$; and (d) $Re = 800$.

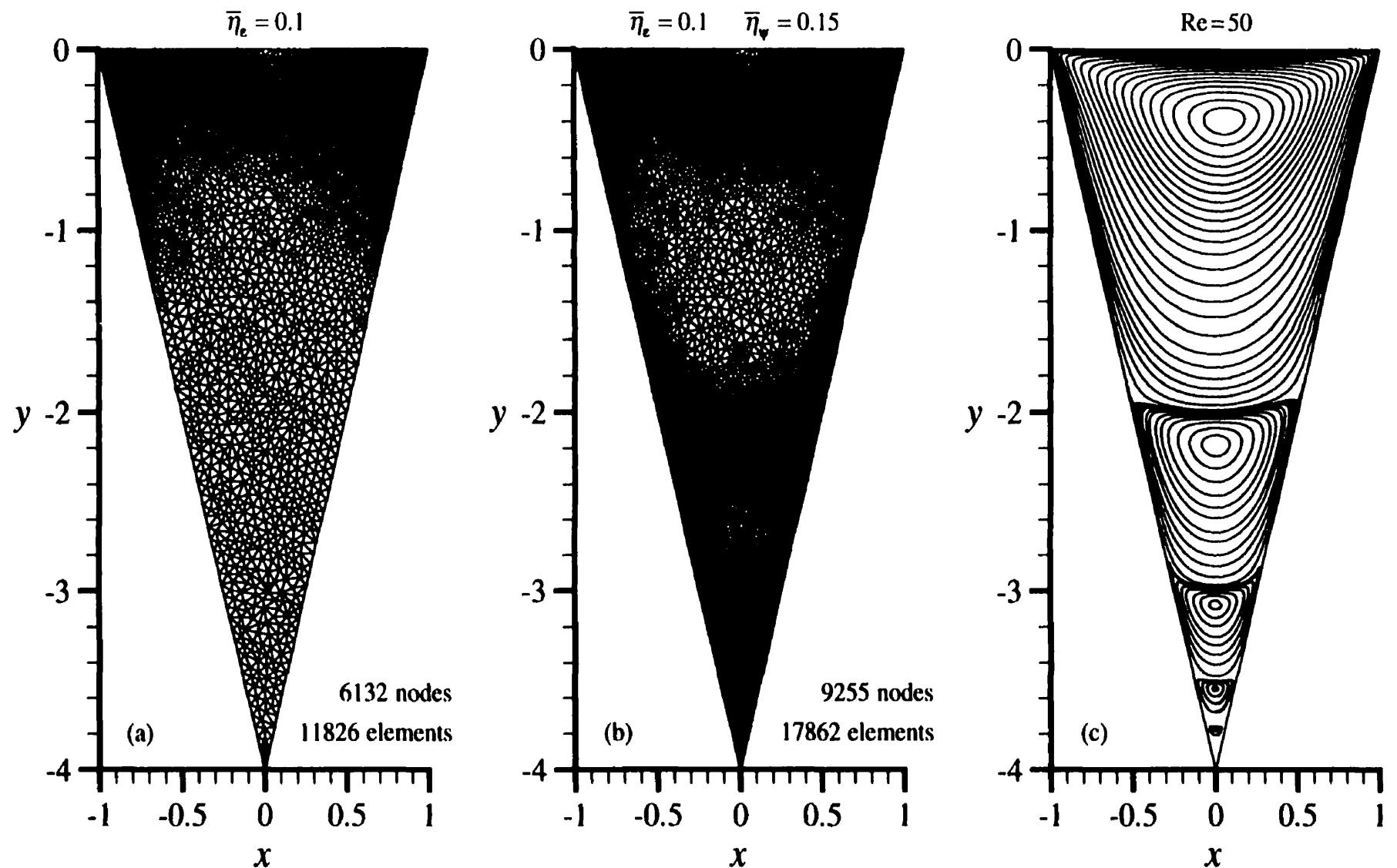


Figure 6.22: Lid-driven flow in a triangular cavity. h -adaptive simulations for $Re=50$: (a) final refined mesh using the strain-rate error estimator (with a target error of $\bar{\eta}_e = 0.1$) alone to drive the adaptive procedure; (b) final refined mesh using the strain-rate error estimator ($\bar{\eta}_e = 0.1$) in combination with the flow-angle indicator ($\bar{\eta}_\psi = 0.15$) to drive the adaptive procedure; and (c) the computed streamline pattern obtained on grid (b).

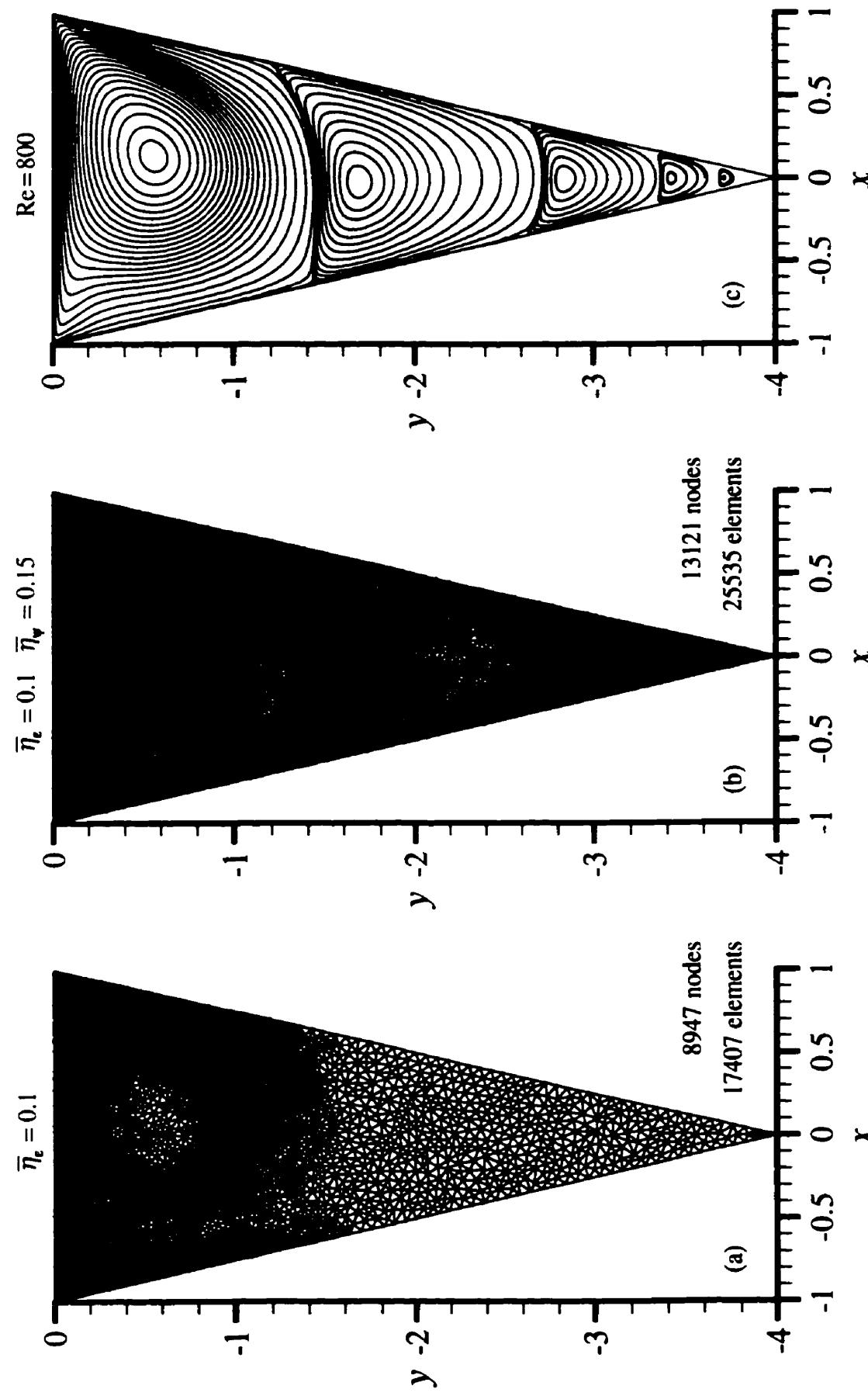


Figure 6.23: Lid-driven flow in a triangular cavity. h -adaptive simulations for $Re = 800$: (a) final refined mesh using the strain-rate error estimator (with a target error of $\bar{\eta}_e = 0.1$) alone to drive the adaptive procedure; (b) final refined mesh using the strain-rate error estimator ($\bar{\eta}_e = 0.1$) in combination with the flow-angle indicator ($\bar{\eta}_\nabla = 0.15$) to drive the adaptive procedure; and (c) the computed streamline pattern obtained on grid (b).

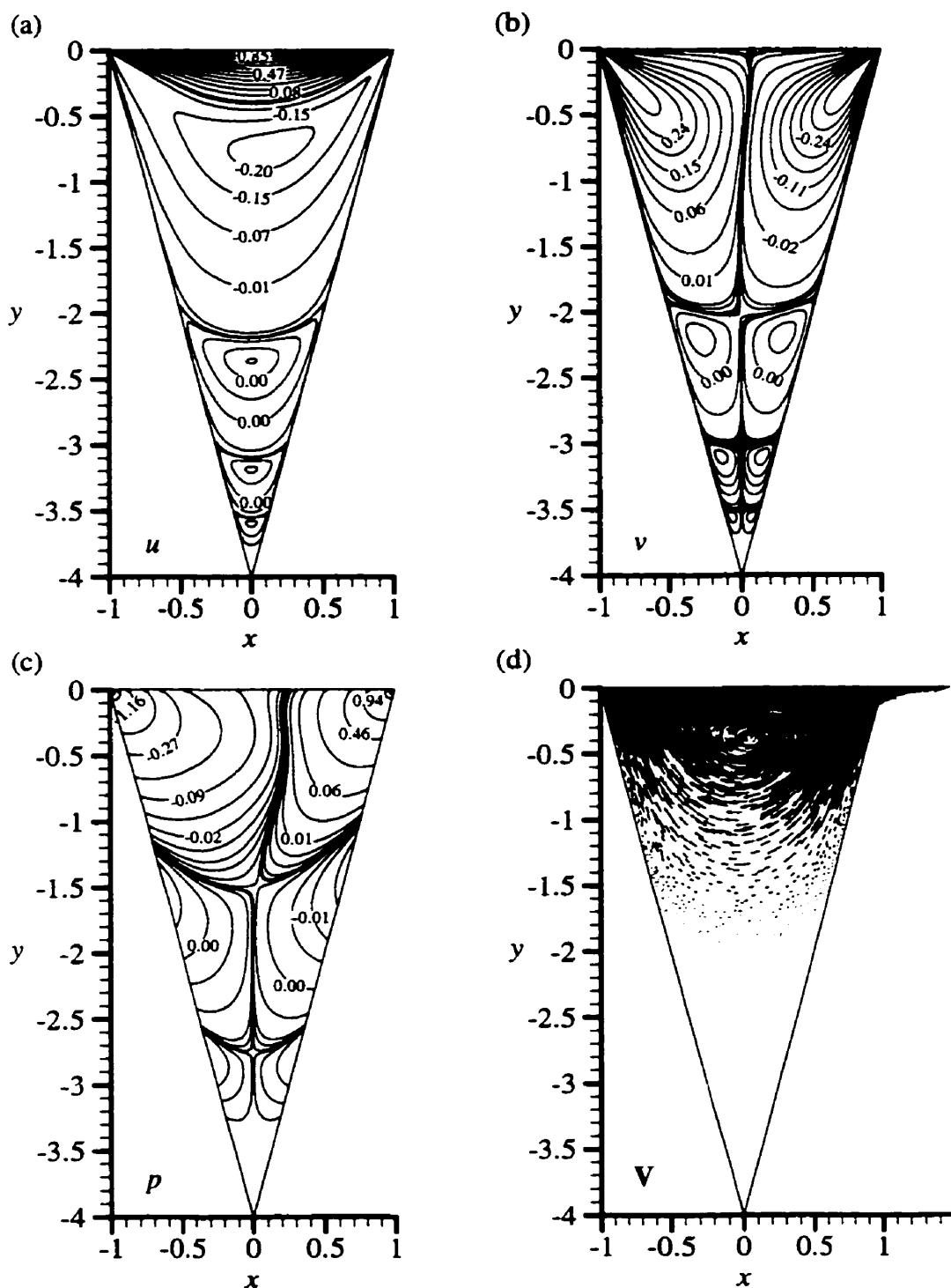


Figure 6.24: Lid-driven flow in a triangular cavity. h -adaptive simulation for $Re=50$: Contour plots of (a) the x -component of velocity, u ; (b) the y -component of velocity, v ; (c) the pressure field, p ; and (d) a vector plot of the velocity field, \mathbf{V} .

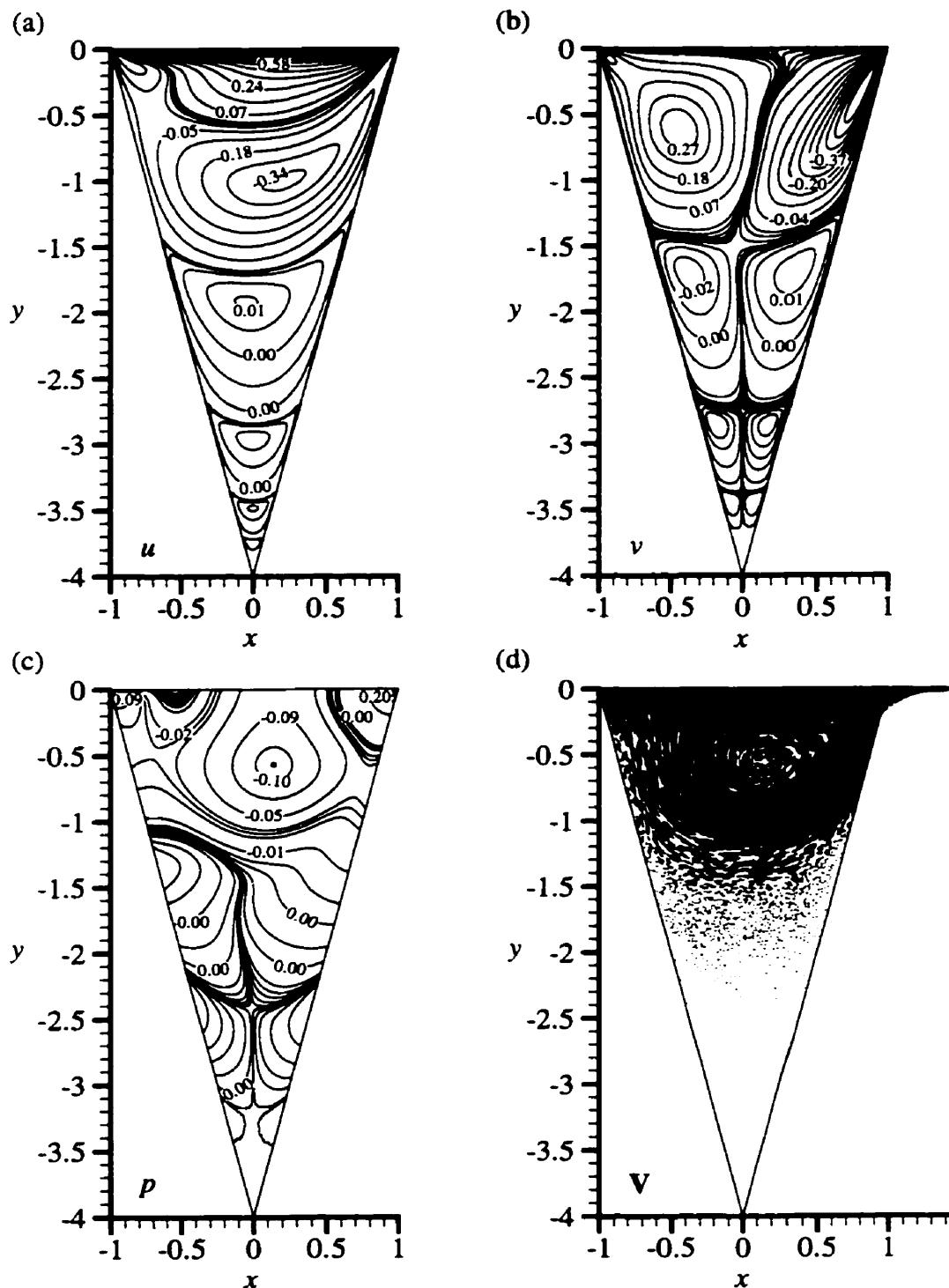


Figure 6.25: Lid-driven flow in a triangular cavity. h -adaptive simulation for $Re=800$: Contour plots of (a) the x -component of velocity, u ; (b) the y -component of velocity, v ; (c) the pressure field, p ; and (d) a vector plot of the velocity field, \mathbf{V} .

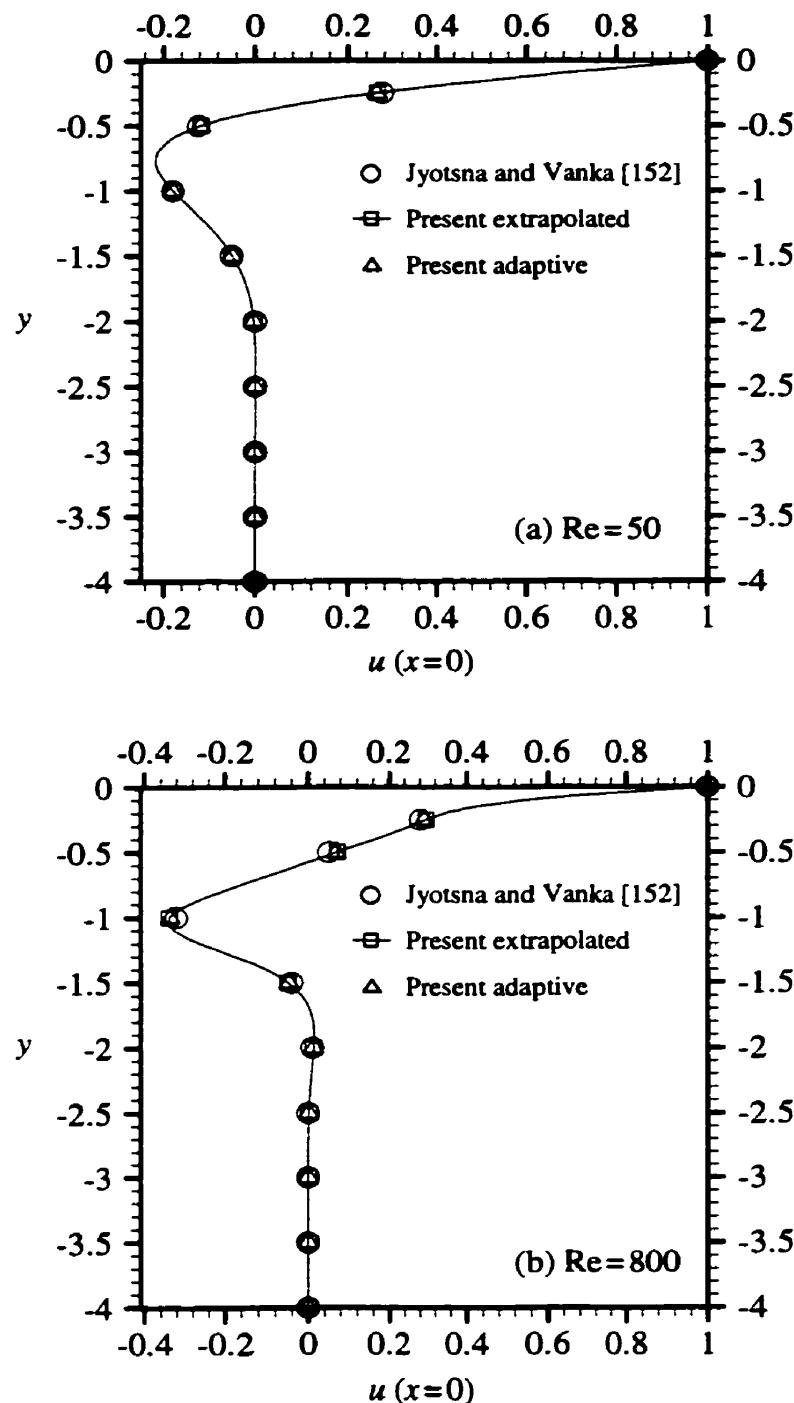


Figure 6.26: Lid-driven flow in a triangular cavity. Results obtained in the present work are compared with the published solutions of Jyotsna and Vanka [152]: The x -component of velocity, u , along the centerline ($x=0$) of the cavity (horizontal axis) is plotted versus y (vertical axis) for the case of (a) $Re=50$; and (b) $Re=800$. (Note: the “present extrapolated” results correspond to the 3-grid extrapolations using grids 3-5 in Tables 6.7 (b) and 6.8 (b), and the “present adaptive” results correspond to grid 5 in Tables 6.7 (b) and 6.8 (b)).

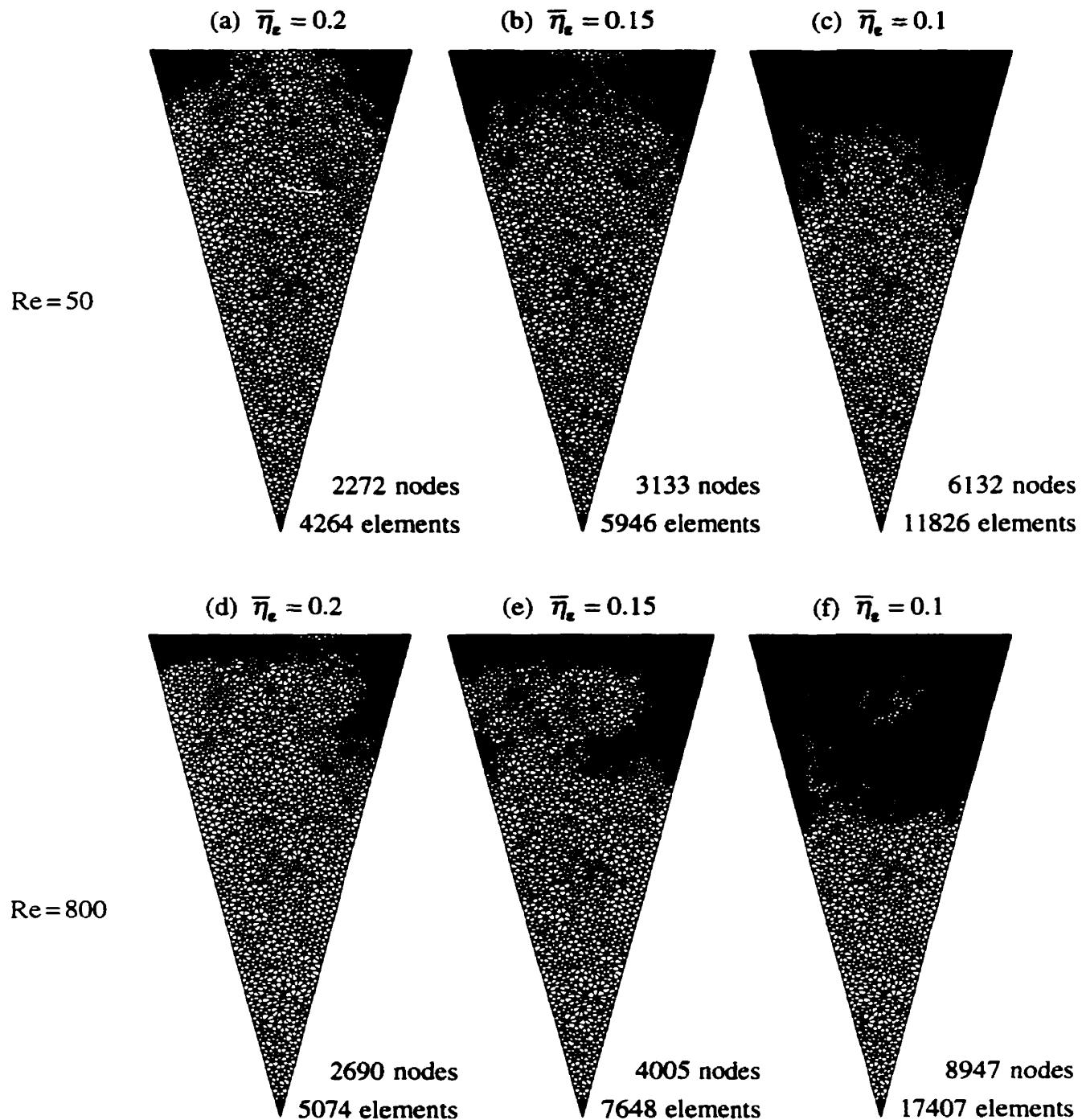


Figure 6.27: Lid-driven flow in a triangular cavity. Final refined meshes resulting from h -adaptive simulations using the *strain-rate error estimator*, alone, to drive the adaptive procedure: h -adaptive simulations were performed for $Re=50$ (top row); and $Re=800$ (bottom row) with three different specified target errors of, $\bar{\eta}_e = 0.2, 0.15$, and 0.10 . The minimum-area constraint was set to $(\Omega_e)_{\min} / \Omega = 5 \times 10^{-5}$ in these simulations.

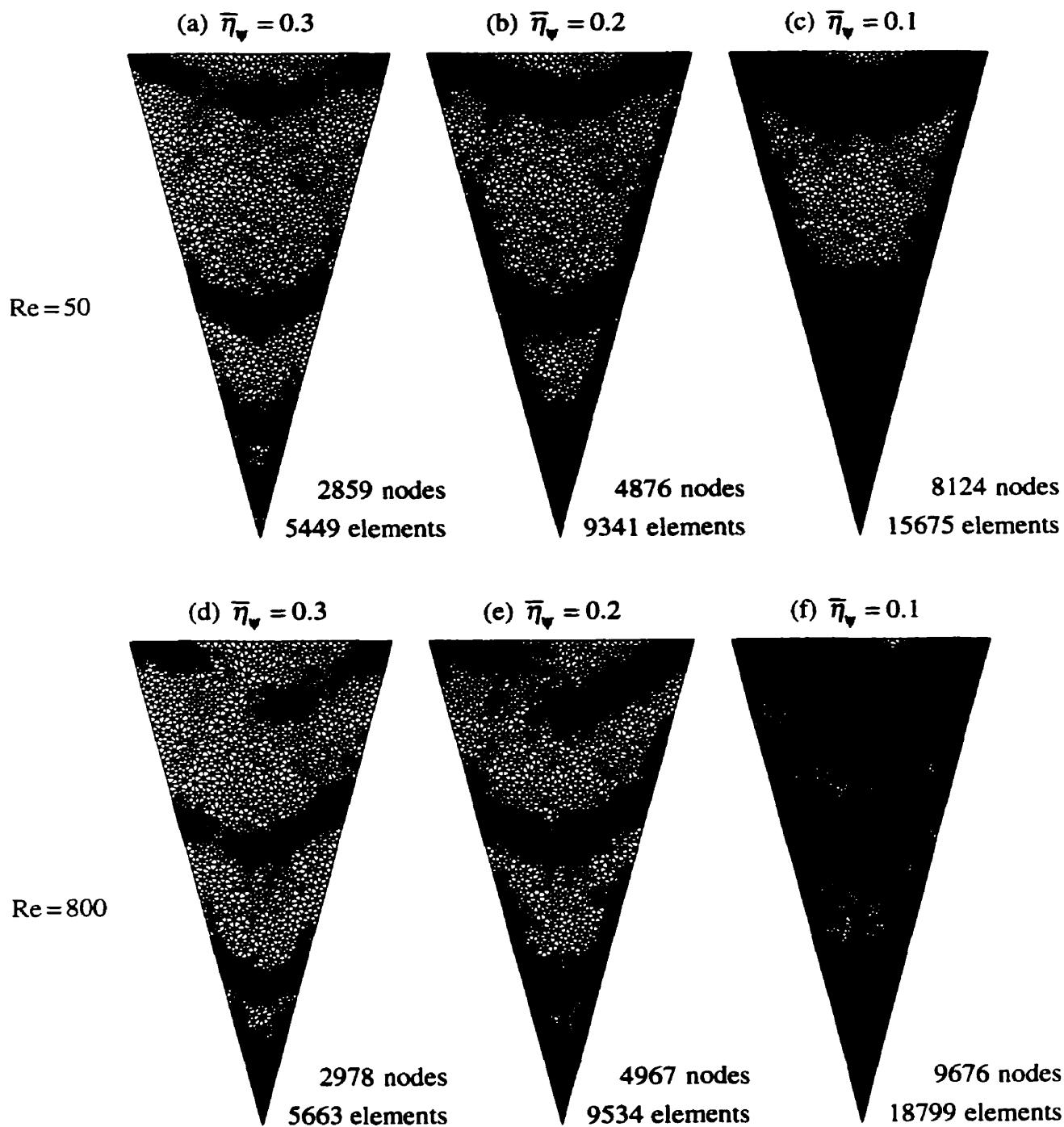


Figure 6.28: Lid-driven flow in a triangular cavity. Final refined meshes resulting from h -adaptive simulations using the *flow-angle indicator*, alone, to drive the adaptive procedure: h -adaptive simulations were performed for $Re = 50$ (top row); and $Re = 800$ (bottom row) with three different specified target errors of, $\bar{\eta}_v = 0.3, 0.2$, and 0.1 . The minimum-area constraint was set to $(\Omega_e)_{\min} / \Omega = 5 \times 10^{-5}$ in these simulations.

| Grid | Total number of nodes | Total number of elements | Target error in preliminary adaptive analysis, $\bar{\eta}_k$ | Specified minimum element area, $(\Omega_e)_{\min} / \Omega$ |
|------|-----------------------|--------------------------|---|--|
| 26k | 26129 | 51629 | 0.01 | 1×10^{-5} |
| 54k | 54200 | 107376 | 0.01 | 1×10^{-6} |
| 104k | 104558 | 207768 | 0.005 | 1×10^{-6} |

Table 6.1: Heat conduction in an elliptically shaped domain with an asymmetrically located triangular hole: Specifications of the three grids used in the ACM iterative convergence tests.

| Grid | ACM | | | SOR-GS | |
|------|-------------------------------|----------------------|----------------------------|----------------------|----------------------------|
| | Number of levels in ACM cycle | Total CPU time (min) | Total number of work units | Total CPU time (min) | Total number of work units |
| 26k | 10 | 1.25 | 2475 | 10.00 | 19800 |
| 54k | 11 | 3.03 | 4185 | 25.17 | 34800 |
| 104k | 12 | 17.94 | 5010 | 246.01 | 68700 |

- One work unit represents the computational effort required to sweep the finest grid in the ACM cycle once using the base solver. Thus, a work unit is different, in terms of the absolute number of computer operations, for each of the three fine grids presented in the table.

Table 6.2: Heat conduction in an elliptically shaped domain with an asymmetrically located triangular hole: ACM performance is compared with that of the standard SOR-GS solver. CPU times and total number of work units are presented.

Data presented in this table: $\log|(u_o - u^h)/u_o|$

| (a) | CVFEM values from 10 different grids | | | | | | | | | |
|-------|--------------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | Re | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0.001 | -1.75 | -2.38 | -2.83 | -3.12 | -3.34 | -3.52 | -3.68 | -3.81 | -3.94 | -4.05 |
| 0.01 | -1.75 | -2.38 | -2.83 | -3.12 | -3.34 | -3.52 | -3.68 | -3.82 | -3.94 | -4.05 |
| 0.1 | -1.76 | -2.39 | -2.85 | -3.14 | -3.36 | -3.54 | -3.70 | -3.84 | -3.96 | -4.07 |
| 1 | -1.81 | -2.51 | -3.04 | -3.36 | -3.61 | -3.81 | -4.00 | -4.16 | -4.30 | -4.44 |
| 10 | -1.17 | -1.59 | -1.90 | -2.13 | -2.32 | -2.47 | -2.60 | -2.72 | -2.82 | -2.91 |
| 100 | -0.17 | -0.43 | -0.65 | -0.83 | -0.98 | -1.11 | -1.22 | -1.32 | -1.41 | -1.50 |
| 1000 | - | - | -0.42 | -0.51 | -0.59 | -0.66 | -0.72 | -0.77 | -0.82 | -0.86 |

Data presented in this table: $\log|(u_o - \bar{u}_o)/u_o|$

| (b) | 2-Grid extrapolation | | | | | | | | | Corresponding series expansion: $u^h = \bar{u}_o + f_1 h_c$ | |
|-------|----------------------|--------|-------|-------|-------|-------|-------|-------|-------|---|------|
| | Re | Grids: | 1-2 | 2-3 | 3-4 | 4-5 | 5-6 | 6-7 | 7-8 | 8-9 | 9-10 |
| 0.001 | | -2.03 | -2.40 | -2.86 | -3.13 | -3.31 | -3.47 | -3.61 | -3.74 | -3.85 | |
| 0.01 | | -2.03 | -2.40 | -2.86 | -3.13 | -3.31 | -3.47 | -3.62 | -3.74 | -3.85 | |
| 0.1 | | -2.03 | -2.41 | -2.87 | -3.15 | -3.32 | -3.49 | -3.63 | -3.75 | -3.86 | |
| 1 | | -2.02 | -2.47 | -2.99 | -3.31 | -3.50 | -3.67 | -3.82 | -3.96 | -4.08 | |
| 10 | | -1.79 | -1.86 | -2.07 | -2.25 | -2.42 | -2.56 | -2.68 | -2.78 | -2.88 | |
| 100 | | -1.20 | -1.12 | -1.11 | -1.18 | -1.26 | -1.33 | -1.41 | -1.48 | -1.55 | |
| 1000 | | - | - | -1.13 | -1.37 | -1.50 | -1.58 | -1.66 | -1.70 | -1.76 | |
| (c) | 3-Grid extrapolation | | | | | | | | | $u^h = \bar{u}_o + f_1 h_c + f_2 h_c^2$ | |
| | Re | Grids: | 1-3 | 2-4 | 3-5 | 4-6 | 5-7 | 6-8 | 7-9 | 8-10 | |
| 0.001 | | -2.89 | -2.92 | -3.64 | -5.34 | -4.37 | -4.42 | -4.58 | -4.67 | | |
| 0.01 | | -2.89 | -2.92 | -3.64 | -5.32 | -4.37 | -4.42 | -4.58 | -4.67 | | |
| 0.1 | | -2.92 | -2.92 | -3.62 | -5.19 | -4.36 | -4.42 | -4.58 | -4.67 | | |
| 1 | | -3.47 | -2.87 | -3.53 | -4.64 | -4.29 | -4.37 | -4.54 | -4.64 | | |
| 10 | | -1.90 | -2.46 | -2.99 | -3.60 | -3.82 | -4.01 | -4.22 | -4.38 | | |
| 100 | | -0.84 | -1.10 | -1.32 | -1.47 | -1.62 | -1.76 | -1.90 | -2.04 | | |
| 1000 | | - | - | -2.37 | -2.08 | -1.88 | -2.00 | -1.94 | -2.06 | | |
| (d) | 4-Grid extrapolation | | | | | | | | | $u^h = \bar{u}_o + f_1 h_c + f_2 h_c^2 + f_3 h_c^3$ | |
| | Re | Grids: | 1-4 | 2-5 | 3-6 | 4-7 | 5-8 | 6-9 | 7-10 | | |
| 0.001 | | -2.69 | -3.39 | -3.65 | -4.03 | -4.52 | -5.53 | -5.03 | | | |
| 0.01 | | -2.69 | -3.39 | -3.65 | -4.03 | -4.52 | -5.55 | -5.03 | | | |
| 0.1 | | -2.70 | -3.39 | -3.65 | -4.03 | -4.53 | -5.54 | -5.03 | | | |
| 1 | | -2.71 | -3.38 | -3.60 | -4.06 | -4.54 | -5.85 | -5.08 | | | |
| 10 | | -3.37 | -3.22 | -3.29 | -4.74 | -5.02 | -4.79 | -5.60 | | | |
| 100 | | -1.24 | -1.57 | -1.72 | -1.96 | -2.22 | -2.54 | -2.92 | | | |
| 1000 | | - | - | -1.68 | -1.71 | -2.30 | -1.86 | -2.57 | | | |

Table 6.3: Verification of the extended Richardson extrapolation technique. This table presents a series of extrapolations performed on the x -component of velocity, u , at the nodal location, $(x,y) = (0.125, 0.125)$, with the linear term included in the associated series expansion: In (a), the logarithm of the relative errors in the CVFEM values, u^h , are presented for each of the simulations described in Section 6.2.4; Below this, the logarithm of the relative errors in the extrapolated values, \bar{u}_o , corresponding to (b) 2-grid; (c) 3-grid; and (d) 4-grid extrapolations, are presented.

Data presented in this table: $\log |(u_o - u^h)/u_o|$

| (a) | CVFEM values from 10 different grids | | | | | | | | | |
|-------|--------------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 0.001 | -1.75 | -2.38 | -2.83 | -3.12 | -3.34 | -3.52 | -3.68 | -3.81 | -3.94 | -4.05 |
| 0.01 | -1.75 | -2.38 | -2.83 | -3.12 | -3.34 | -3.52 | -3.68 | -3.82 | -3.94 | -4.05 |
| 0.1 | -1.76 | -2.39 | -2.85 | -3.14 | -3.36 | -3.54 | -3.70 | -3.84 | -3.96 | -4.07 |
| 1 | -1.81 | -2.51 | -3.04 | -3.36 | -3.61 | -3.81 | -4.00 | -4.16 | -4.30 | -4.44 |
| 10 | -1.17 | -1.59 | -1.90 | -2.13 | -2.32 | -2.47 | -2.60 | -2.72 | -2.82 | -2.91 |
| 100 | -0.17 | -0.43 | -0.65 | -0.83 | -0.98 | -1.11 | -1.22 | -1.32 | -1.41 | -1.50 |
| 1000 | - | - | -0.42 | -0.51 | -0.59 | -0.66 | -0.72 | -0.77 | -0.82 | -0.86 |

Data presented in this table: $\log |(u_o - \bar{u}_o)/u_o|$

| (b) | 2-Grid extrapolation | | | | | | | | | $u^h = \bar{u}_o + f_2 h_c^2$ |
|-------|----------------------|-------|-------|-------|-------|-------|-------|-------|-------|---|
| | Grids: | 1-2 | 2-3 | 3-4 | 4-5 | 5-6 | 6-7 | 7-8 | 8-9 | |
| 0.001 | | -3.53 | -3.16 | -3.80 | -4.14 | -4.24 | -4.37 | -4.50 | -4.60 | -4.70 |
| 0.01 | | -3.52 | -3.15 | -3.80 | -4.14 | -4.24 | -4.37 | -4.50 | -4.60 | -4.70 |
| 0.1 | | -3.42 | -3.15 | -3.79 | -4.14 | -4.23 | -4.37 | -4.50 | -4.60 | -4.70 |
| 1 | | -2.95 | -3.09 | -3.73 | -4.09 | -4.21 | -4.35 | -4.49 | -4.59 | -4.69 |
| 10 | | -1.93 | -2.67 | -3.27 | -3.73 | -3.97 | -4.18 | -4.35 | -4.49 | -4.61 |
| 100 | | -0.57 | -0.99 | -1.30 | -1.54 | -1.76 | -1.96 | -2.15 | -2.32 | -2.47 |
| 1000 | | - | - | -0.68 | -0.80 | -0.88 | -0.94 | -1.00 | -1.05 | -1.09 |
| (c) | 3-Grid extrapolation | | | | | | | | | $u^h = \bar{u}_o + f_2 h_c^2 + f_3 h_c^3$ |
| | Grids: | 1-3 | 2-4 | 3-5 | 4-6 | 5-7 | 6-8 | 7-9 | 8-10 | |
| 0.001 | | -3.10 | -3.85 | -5.41 | -4.38 | -4.73 | -4.93 | -5.00 | -5.11 | |
| 0.01 | | -3.10 | -3.85 | -5.40 | -4.38 | -4.73 | -4.93 | -5.00 | -5.11 | |
| 0.1 | | -3.10 | -3.84 | -5.32 | -4.39 | -4.73 | -4.93 | -5.00 | -5.11 | |
| 1 | | -3.14 | -3.80 | -4.87 | -4.41 | -4.75 | -4.95 | -5.02 | -5.12 | |
| 10 | | -3.50 | -3.45 | -3.94 | -4.96 | -5.22 | -5.42 | -5.27 | -5.30 | |
| 100 | | -1.21 | -1.67 | -2.03 | -2.44 | -2.93 | -3.79 | -3.67 | -3.49 | |
| 1000 | | - | - | -0.92 | -1.00 | -1.07 | -1.13 | -1.17 | -1.22 | |
| (d) | 4-Grid extrapolation | | | | | | | | | $u^h = \bar{u}_o + f_2 h_c^2 + f_3 h_c^3 + f_4 h_c^4$ |
| | Grids: | 1-4 | 2-5 | 3-6 | 4-7 | 5-8 | 6-9 | 7-10 | | |
| 0.001 | | -3.48 | -4.26 | -4.15 | -5.79 | -5.42 | -5.13 | -5.37 | | |
| 0.01 | | -3.48 | -4.26 | -4.15 | -5.79 | -5.42 | -5.13 | -5.37 | | |
| 0.1 | | -3.48 | -4.26 | -4.14 | -5.80 | -5.42 | -5.13 | -5.37 | | |
| 1 | | -3.48 | -4.32 | -4.13 | -5.87 | -5.45 | -5.13 | -5.36 | | |
| 10 | | -3.44 | -4.85 | -4.02 | -5.79 | -5.91 | -5.12 | -5.34 | | |
| 100 | | -1.87 | -2.38 | -3.77 | -2.98 | -2.99 | -3.13 | -3.30 | | |
| 1000 | | - | - | -1.06 | -1.13 | -1.21 | -1.25 | -1.32 | | |

Table 6.4: Verification of the extended Richardson extrapolation technique. This table presents a series of extrapolations performed on the x -component of velocity, u , at the nodal location, $(x,y) = (0.125, 0.125)$, without the linear term included in the associated series expansion: In (a), the logarithm of the relative errors in the CVFEM values, u^h , are presented for each of the simulations described in Section 6.2.4; Below this, the logarithm of the relative errors in the extrapolated values, \bar{u}_o , corresponding to (b) 2-grid; (c) 3-grid; and (d) 4-grid extrapolations, are presented.

Data presented in this table: $\log \left| (\|\boldsymbol{\varepsilon}\|_o - \|\boldsymbol{\varepsilon}\|^h) / \|\boldsymbol{\varepsilon}\|_o \right|$

| (a) | CVFEM values from 10 different grids | | | | | | | | | |
|-------|--------------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Re | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 0.001 | -3.34 | -3.99 | -4.19 | -4.42 | -4.61 | -4.77 | -4.91 | -5.03 | -5.14 | -5.23 |
| 0.01 | -3.34 | -3.99 | -4.19 | -4.42 | -4.61 | -4.77 | -4.91 | -5.03 | -5.14 | -5.23 |
| 0.1 | -3.36 | -3.98 | -4.18 | -4.41 | -4.60 | -4.76 | -4.90 | -5.02 | -5.13 | -5.22 |
| 1 | -3.03 | -3.40 | -3.72 | -3.96 | -4.16 | -4.32 | -4.45 | -4.57 | -4.68 | -4.77 |
| 10 | -1.23 | -1.81 | -2.16 | -2.41 | -2.60 | -2.76 | -2.90 | -3.01 | -3.12 | -3.21 |
| 100 | -1.36 | -1.48 | -1.65 | -1.79 | -1.90 | -2.00 | -2.10 | -2.18 | -2.26 | -2.34 |
| 1000 | - | - | -2.08 | -2.28 | -2.37 | -2.41 | -2.43 | -2.44 | -2.45 | -2.46 |

Data presented in this table: $\log \left| (\|\boldsymbol{\varepsilon}\|_o - \overline{\|\boldsymbol{\varepsilon}\|}_o) / \|\boldsymbol{\varepsilon}\|_o \right|$

| (b) | 2-Grid extrapolation | | | | | | | | Corresponding series expansion: $\ \boldsymbol{\varepsilon}\ ^h = \ \boldsymbol{\varepsilon}\ _o + f_1 h_c$ | |
|-------|----------------------|-------|-------|-------|-------|-------|-------|-------|---|-------|
| Re | Grids: | 1-2 | 2-3 | 3-4 | 4-5 | 5-6 | 6-7 | 7-8 | 8-9 | 9-10 |
| 0.001 | | -3.18 | -4.96 | -4.40 | -4.53 | -4.67 | -4.81 | -4.94 | -5.05 | -5.15 |
| 0.01 | | -3.18 | -4.96 | -4.40 | -4.53 | -4.67 | -4.81 | -4.94 | -5.05 | -5.15 |
| 0.1 | | -3.19 | -4.88 | -4.39 | -4.52 | -4.66 | -4.81 | -4.93 | -5.04 | -5.14 |
| 1 | | -3.89 | -3.65 | -3.86 | -4.06 | -4.23 | -4.37 | -4.50 | -4.61 | -4.71 |
| 10 | | -1.56 | -1.99 | -2.29 | -2.51 | -2.68 | -2.83 | -2.95 | -3.06 | -3.16 |
| 100 | | -1.63 | -3.77 | -2.91 | -2.60 | -2.48 | -2.44 | -2.45 | -2.47 | -2.50 |
| 1000 | | - | - | -2.37 | -3.35 | -2.67 | -2.58 | -2.53 | -2.54 | -2.54 |
| (c) | 3-Grid extrapolation | | | | | | | | $\ \boldsymbol{\varepsilon}\ ^h = \ \boldsymbol{\varepsilon}\ _o + f_1 h_c + f_2 h_c^2$ | |
| Re | Grids: | 1-3 | 2-4 | 3-5 | 4-6 | 5-7 | 6-8 | 7-9 | | |
| 0.001 | | -3.46 | -4.16 | -4.84 | -5.34 | -6.29 | -6.86 | -6.61 | | |
| 0.01 | | -3.46 | -4.16 | -4.84 | -5.34 | -6.29 | -6.86 | -6.61 | | |
| 0.1 | | -3.46 | -4.16 | -4.84 | -5.34 | -6.30 | -6.85 | -6.60 | | |
| 1 | | -3.57 | -4.25 | -4.97 | -5.55 | -6.47 | -6.22 | -6.28 | | |
| 10 | | -2.82 | -3.86 | -4.46 | -5.06 | -5.94 | -6.38 | -6.15 | | |
| 100 | | -1.92 | -2.64 | -2.36 | -2.30 | -2.37 | -2.46 | -2.56 | | |
| 1000 | | - | - | -2.12 | -2.26 | -2.41 | -2.40 | -2.59 | | |
| (d) | 4-Grid extrapolation | | | | | | | | $\ \boldsymbol{\varepsilon}\ ^h = \ \boldsymbol{\varepsilon}\ _o + f_1 h_c + f_2 h_c^2 + f_3 h_c^3$ | |
| Re | Grids: | 1-4 | 2-5 | 3-6 | 4-7 | 5-8 | 6-9 | 7-10 | | |
| 0.001 | | -4.63 | -4.67 | -5.26 | -5.31 | -5.91 | -6.34 | -6.35 | | |
| 0.01 | | -4.63 | -4.67 | -5.26 | -5.31 | -5.91 | -6.33 | -6.36 | | |
| 0.1 | | -4.63 | -4.67 | -5.26 | -5.31 | -5.91 | -6.34 | -6.35 | | |
| 1 | | -4.79 | -4.71 | -5.30 | -5.34 | -5.98 | -6.44 | -6.42 | | |
| 10 | | -3.49 | -4.47 | -4.76 | -5.05 | -5.52 | -5.89 | -6.10 | | |
| 100 | | -3.04 | -2.24 | -2.26 | -2.47 | -2.68 | -2.89 | -3.12 | | |
| 1000 | | - | - | -2.47 | -2.76 | -2.39 | -3.87 | -2.49 | | |

Table 6.5: Verification of the extended Richardson extrapolation technique. This table presents a series of extrapolations performed on the global strain-rate energy norm, $\|\boldsymbol{\varepsilon}\|$, with the linear term included in the associated series expansion: In (a), the logarithm of the relative errors in the CVFEM values, $\|\boldsymbol{\varepsilon}\|^h$, are presented for each of the simulations described in Section 6.2.4; Below this, the logarithm of the relative errors in the extrapolated values, $\overline{\|\boldsymbol{\varepsilon}\|}_o$, corresponding to (b) 2-grid; (c) 3-grid; and (d) 4-grid extrapolations, are presented.

Data presented in this table: $\log \left(\left| \|\boldsymbol{\epsilon}\|_o - \overline{\|\boldsymbol{\epsilon}\|}_o \right) / \|\boldsymbol{\epsilon}\|_o \right|$

| (a) | | CVFEM values from 10 different grids | | | | | | | | | |
|-------|-------|--------------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|----|
| Re | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 0.001 | -3.34 | -3.99 | -4.19 | -4.42 | -4.61 | -4.77 | -4.91 | -5.03 | -5.14 | -5.23 | |
| 0.01 | -3.34 | -3.99 | -4.19 | -4.42 | -4.61 | -4.77 | -4.91 | -5.03 | -5.14 | -5.23 | |
| 0.1 | -3.36 | -3.98 | -4.18 | -4.41 | -4.60 | -4.76 | -4.90 | -5.02 | -5.13 | -5.22 | |
| 1 | -3.03 | -3.40 | -3.72 | -3.96 | -4.16 | -4.32 | -4.45 | -4.57 | -4.68 | -4.77 | |
| 10 | -1.23 | -1.81 | -2.16 | -2.41 | -2.60 | -2.76 | -2.90 | -3.01 | -3.12 | -3.21 | |
| 100 | -1.36 | -1.48 | -1.65 | -1.79 | -1.90 | -2.00 | -2.10 | -2.18 | -2.26 | -2.34 | |
| 1000 | - | - | -2.08 | -2.28 | -2.37 | -2.41 | -2.43 | -2.44 | -2.45 | -2.46 | |

Data presented in this table: $\log \left(\left| \|\boldsymbol{\epsilon}\|_o - \overline{\|\boldsymbol{\epsilon}\|}_o \right) / \|\boldsymbol{\epsilon}\|_o \right|$

| (b) | | 2-Grid extrapolation | | | | | | | | Corresponding series expansion: $\ \boldsymbol{\epsilon}\ ^h = \ \boldsymbol{\epsilon}\ _o + f_2 h_c^2$ | |
|-------|--------|----------------------|-------|-------|-------|-------|-------|-------|-------|---|--|
| Re | Grids: | 1-2 | 2-3 | 3-4 | 4-5 | 5-6 | 6-7 | 7-8 | 8-9 | 9-10 | |
| 0.001 | | -3.54 | -4.47 | -5.32 | -6.34 | -6.34 | -6.34 | -6.42 | -6.51 | -6.61 | |
| 0.01 | | -3.54 | -4.47 | -5.32 | -6.34 | -6.34 | -6.34 | -6.42 | -6.51 | -6.61 | |
| 0.1 | | -3.54 | -4.47 | -5.32 | -6.35 | -6.34 | -6.33 | -6.42 | -6.51 | -6.61 | |
| 1 | | -3.65 | -4.59 | -5.60 | -6.38 | -6.07 | -6.17 | -6.30 | -6.42 | -6.54 | |
| 10 | | -2.95 | -4.15 | -4.95 | -6.04 | -6.07 | -6.05 | -6.14 | -6.26 | -6.38 | |
| 100 | | -1.52 | -1.88 | -2.06 | -2.23 | -2.41 | -2.58 | -2.74 | -2.89 | -3.03 | |
| 1000 | | - | - | -2.94 | -2.59 | -2.51 | -2.49 | -2.48 | -2.49 | -2.50 | |
| (c) | | 3-Grid extrapolation | | | | | | | | $\ \boldsymbol{\epsilon}\ ^h = \ \boldsymbol{\epsilon}\ _o + f_2 h_c^2 + f_3 h_c^3$ | |
| Re | Grids: | 1-3 | 2-4 | 3-5 | 4-6 | 5-7 | 6-8 | 7-9 | 8-10 | | |
| 0.001 | | -4.54 | -4.94 | -5.48 | -5.81 | -6.32 | -6.63 | -6.83 | -7.09 | | |
| 0.01 | | -4.54 | -4.94 | -5.48 | -5.81 | -6.32 | -6.63 | -6.83 | -7.09 | | |
| 0.1 | | -4.54 | -4.94 | -5.48 | -5.81 | -6.32 | -6.63 | -6.83 | -7.09 | | |
| 1 | | -4.63 | -4.98 | -5.53 | -5.86 | -6.41 | -6.73 | -6.93 | -7.21 | | |
| 10 | | -3.71 | -4.66 | -5.10 | -5.53 | -6.01 | -6.40 | -6.73 | -7.12 | | |
| 100 | | -2.04 | -2.20 | -2.48 | -2.81 | -3.14 | -3.49 | -3.93 | -4.68 | | |
| 1000 | | - | - | -2.42 | -2.43 | -2.47 | -2.46 | -2.52 | -2.51 | | |
| (d) | | 4-Grid extrapolation | | | | | | | | $\ \boldsymbol{\epsilon}\ ^h = \ \boldsymbol{\epsilon}\ _o + f_2 h_c^2 + f_3 h_c^3 + f_4 h_c^4$ | |
| Re | Grids: | 1-4 | 2-5 | 3-6 | 4-7 | 5-8 | 6-9 | 7-10 | | | |
| 0.001 | | -5.09 | -6.67 | -6.44 | -6.31 | -7.33 | -7.60 | -7.57 | | | |
| 0.01 | | -5.09 | -6.67 | -6.44 | -6.31 | -7.33 | -7.61 | -7.58 | | | |
| 0.1 | | -5.09 | -6.67 | -6.44 | -6.31 | -7.33 | -7.60 | -7.57 | | | |
| 1 | | -5.11 | -6.64 | -6.49 | -6.30 | -7.28 | -7.66 | -7.54 | | | |
| 10 | | -4.91 | -5.70 | -6.39 | -6.07 | -6.58 | -6.96 | -6.99 | | | |
| 100 | | -2.24 | -2.69 | -3.45 | -4.75 | -3.83 | -3.78 | -3.86 | | | |
| 1000 | | - | - | -2.44 | -2.51 | -2.44 | -2.62 | -2.51 | | | |

Table 6.6: Verification of the extended Richardson extrapolation technique. This table presents a series of extrapolations performed on the global strain-rate energy norm, $\|\boldsymbol{\epsilon}\|$, without the linear term included in the associated series expansion: In (a), the logarithm of the relative errors in the CVFEM values, $\|\boldsymbol{\epsilon}\|^h$, are presented for each of the simulations described in Section 6.2.4; Below this, the logarithm of the relative errors in the extrapolated values, $\overline{\|\boldsymbol{\epsilon}\|}_o$, corresponding to (b) 2-grid; (c) 3-grid; and (d) 4-grid extrapolations, are presented.

Locations of Eddy Centres for $Re = 50$

(a) CVFEM values - Uniform initial grid

| Grid | Primary eddy | | Secondary eddy | |
|------|--------------|-----------|----------------|-----------|
| | x_1 | y_1 | x_2 | y_2 |
| 1 | 0.058061 | -0.403141 | 0.003016 | -2.186134 |
| 2 | 0.058811 | -0.398483 | 0.003119 | -2.181086 |
| 3 | 0.058891 | -0.397239 | 0.003176 | -2.179789 |
| 4 | 0.059027 | -0.396499 | 0.003335 | -2.179175 |

| 2-Grid extrapolations | | Associated series expansion: $\phi^h = \bar{\phi}_o + f_2 h_c^2$ | | | |
|-----------------------|--|--|-----------|----------|-----------|
| Grids | | x_1 | y_1 | x_2 | y_2 |
| 1-2 | | 0.059061 | -0.396931 | 0.003153 | -2.179403 |
| 2-3 | | 0.058954 | -0.396244 | 0.003222 | -2.178751 |
| 3-4 | | 0.059202 | -0.395548 | 0.003540 | -2.178387 |

| 3-Grid extrapolations | | $\phi^h = \bar{\phi}_o + f_2 h_c^2 + f_3 h_c^3$ | | | |
|-----------------------|--|---|-----------|----------|-----------|
| Grids | | x_1 | y_1 | x_2 | y_2 |
| 1-3 | | 0.058928 | -0.396072 | 0.003239 | -2.178588 |
| 2-4 | | 0.059340 | -0.395161 | 0.003717 | -2.178185 |

(b) CVFEM values - Solution-adapted initial grid

| Grid | Primary eddy | | Secondary eddy | |
|------|--------------|-----------|----------------|-----------|
| | x_1 | y_1 | x_2 | y_2 |
| 1 | 0.058811 | -0.395168 | 0.002865 | -2.175905 |
| 2 | 0.058896 | -0.395202 | 0.003298 | -2.177861 |
| 3 | 0.059013 | -0.395027 | 0.003340 | -2.177999 |
| 4 | 0.059059 | -0.394913 | 0.003304 | -2.177990 |
| 5 | 0.059084 | -0.394836 | 0.003339 | -2.177995 |

| 2-Grid extrapolations | | Associated series expansion: $\phi^h = \bar{\phi}_o + f_2 h_c^2$ | | | |
|-----------------------|--|--|-----------|----------|-----------|
| Grids | | x_1 | y_1 | x_2 | y_2 |
| 1-2 | | 0.058924 | -0.395213 | 0.003443 | -2.178513 |
| 2-3 | | 0.059106 | -0.394887 | 0.003374 | -2.178109 |
| 3-4 | | 0.059119 | -0.394766 | 0.003257 | -2.177978 |
| 4-5 | | 0.059128 | -0.394700 | 0.003401 | -2.178004 |

| 3-Grid extrapolations | | $\phi^h = \bar{\phi}_o + f_2 h_c^2 + f_3 h_c^3$ | | | |
|-----------------------|--|---|-----------|----------|-----------|
| Grids | | x_1 | y_1 | x_2 | y_2 |
| 1-3 | | 0.059151 | -0.394806 | 0.003356 | -2.178008 |
| 2-4 | | 0.059126 | -0.394699 | 0.003192 | -2.177906 |
| 3-5 | | 0.059136 | -0.394642 | 0.003528 | -2.178027 |

Table 6.7: Lid-driven flow in a triangular cavity: Computed locations of the primary and secondary eddy centres for $Re=50$. This table presents CVFEM values and corresponding extended Richardson extrapolations for (a) a hierarchy of grids constructed from a *uniform* initial grid; and (b) a hierarchy of grids constructed from a *solution-adapted* initial grid.

Locations of Eddy Centres for $Re = 800$

(a)

| CVFEM values - Uniform initial grid | | | | |
|-------------------------------------|--------------|-----------|----------------|-----------|
| Grid | Primary eddy | | Secondary eddy | |
| | x_1 | y_1 | x_2 | y_2 |
| 1 | 0.160827 | -0.553147 | 0.000533 | -1.713230 |
| 2 | 0.140940 | -0.562443 | -0.016262 | -1.689898 |
| 3 | 0.135692 | -0.564315 | -0.020837 | -1.684971 |
| 4 | 0.133406 | -0.564981 | -0.022787 | -1.682986 |

| 2-Grid extrapolations | | Associated series expansion: $\phi^h = \bar{\phi}_o + f_2 h_c^2$ | | | |
|-----------------------|--|--|-----------|-----------|-----------|
| Grids | | x_1 | y_1 | x_2 | y_2 |
| 1-2 | | 0.134311 | -0.565542 | -0.021860 | -1.682121 |
| 2-3 | | 0.131493 | -0.565813 | -0.024498 | -1.681029 |
| 3-4 | | 0.130468 | -0.565836 | -0.025294 | -1.680434 |

| 3-Grid extrapolations | | $\phi^h = \bar{\phi}_o + f_2 h_c^2 + f_3 h_c^3$ | | | |
|-----------------------|--|---|-----------|-----------|-----------|
| Grids | | x_1 | y_1 | x_2 | y_2 |
| 1-3 | | 0.130789 | -0.565881 | -0.025157 | -1.680757 |
| 2-4 | | 0.129898 | -0.565849 | -0.025736 | -1.680103 |

(b)

| CVFEM values - Solution-adapted initial grid | | | | |
|--|--------------|-----------|----------------|-----------|
| Grid | Primary eddy | | Secondary eddy | |
| | x_1 | y_1 | x_2 | y_2 |
| 1 | 0.168251 | -0.551833 | -0.002078 | -1.702145 |
| 2 | 0.141339 | -0.562498 | -0.018022 | -1.686600 |
| 3 | 0.134891 | -0.564385 | -0.022269 | -1.682946 |
| 4 | 0.132429 | -0.564978 | -0.024012 | -1.681427 |
| 5 | 0.131203 | -0.565249 | -0.024880 | -1.680721 |

| 2-Grid extrapolations | | Associated series expansion: $\phi^h = \bar{\phi}_o + f_2 h_c^2$ | | | |
|-----------------------|--|--|-----------|-----------|-----------|
| Grids | | x_1 | y_1 | x_2 | y_2 |
| 1-2 | | 0.132368 | -0.566053 | -0.023336 | -1.681418 |
| 2-3 | | 0.129733 | -0.565894 | -0.025667 | -1.680022 |
| 3-4 | | 0.129262 | -0.565741 | -0.026253 | -1.679476 |
| 4-5 | | 0.129025 | -0.565731 | -0.026422 | -1.679464 |

| 3-Grid extrapolations | | $\phi^h = \bar{\phi}_o + f_2 h_c^2 + f_3 h_c^3$ | | | |
|-----------------------|--|---|-----------|-----------|-----------|
| Grids | | x_1 | y_1 | x_2 | y_2 |
| 1-3 | | 0.129075 | -0.565854 | -0.026250 | -1.679673 |
| 2-4 | | 0.129001 | -0.565656 | -0.026578 | -1.679172 |
| 3-5 | | 0.128817 | -0.565722 | -0.026570 | -1.679454 |

Table 6.8: Lid-driven flow in a triangular cavity: Computed locations of the primary and secondary eddy centres for $Re=800$. This table presents CVFEM values and corresponding extended Richardson extrapolations for (a) a hierarchy of grids constructed from a uniform initial grid; and (b) a hierarchy of grids constructed from a solution-adapted initial grid.

| | Re = 50 | | | | Re = 800 | | | |
|-------------------------|--------------|--------|----------------|--------|--------------|--------|----------------|--------|
| | Primary eddy | | Secondary eddy | | Primary eddy | | Secondary eddy | |
| | x_1 | y_1 | x_2 | y_2 | x_1 | y_1 | x_2 | y_2 |
| Present | 0.059 | -0.395 | 0.003 | -2.178 | 0.129 | -0.566 | -0.027 | -1.679 |
| Jyotsna and Vanka [152] | 0.050 | -0.395 | 0.002 | -2.182 | 0.153 | -0.545 | 0.000 | -1.695 |

- These values were obtained from the 3-grid extrapolations (grids 3-5) shown in Tables 6.7 (b) and 6.8 (b), unless the corresponding CVFEM values exhibited oscillatory convergence with respect to h_c , in which case these values were obtained from grid 5.

Table 6.9: Lid-driven flow in a triangular cavity: Computed locations of the primary and secondary eddy centres for Re=50 and 800. Present results are compared with the published solutions of Jyotsna and Vanka [152].

Chapter 7

CONCLUSION

In this chapter, a brief review of the thesis and its contributions is presented first. Then, some recommendations for possible enhancements and extensions of this work are outlined.

7.1 Review of the Thesis and Its Main Contributions

In Chapter 1, the objectives and motivation of the thesis were stated, and a review of the pertinent literature was provided. Some of the pioneering work of the mid-1970s in the development of the control-volume finite element method (CVFEM) was summarized, as were the major milestones in its evolution to the present day. The key ideas behind several unstructured grid generation techniques were mentioned, including two of the most prevalent types: advancing-front methods and Delaunay-based methods. A few of the more commonly employed grid-adaptation strategies were also discussed. Then a succinct review of some *a posteriori* error estimation techniques was provided, followed by a discussion of the Zienkiewicz and Zhu error estimator and recovery process, as this is the method employed in the present work. Reviews of other relevant topics were deferred to later Chapters.

In Chapter 2, the formulation of a control-volume finite element method for the simulation of steady, two-dimensional, incompressible, viscous, Newtonian fluid flow and heat transfer was presented. This CVFEM is based on concepts originally proposed by Baliga [28], Baliga and Patankar [29], and Prakash and Patankar [30], and later enhanced and/or extended by Masson *et al.* [81], and others [31-38]. The present CVFEM implementation is based on a co-located, equal-order formulation, in which the dependent

variables are stored at the same nodes in the finite element mesh, and interpolated over the same elements. Furthermore, this method deals directly with the primitive variables, such as the velocity components and pressure. In order to avoid the possibility of obtaining spurious pressure oscillations in the simulation of incompressible fluid flow problems, a special *momentum interpolation* scheme was implemented. The main ideas behind this scheme are borrowed from the work of Prakash and Patankar [30].

One of the contributions made in the present work was to provide a complete proof of the sufficient conditions required for guaranteeing positive *diffusion* contributions to the coefficients in the discretized equations. It is often assumed that Delaunay triangulation in conjunction with linear element-based interpolation of the dependent variables is a sufficient condition for guaranteeing that the algebraic approximations to the diffusion terms in the integral conservation equations contribute positively to the coefficients in the discretized equations. Barth [57] presented a formal proof of this claim, however, his proof is only valid if the diffusion coefficient, Γ_e , is *constant* over the entire domain. Furthermore, his proof is not valid for elements that have at least one edge on the boundary of the domain. In Section 2.4.3.1, a more complete set of sufficient conditions is summarized, and the corresponding proofs are provided in Appendix B.

In Chapter 3, an algebraic multigrid procedure was presented. This multigrid procedure was implemented for the purposes of accelerating convergence in the iterative solution of the discretized equations. In particular, it was employed to accelerate convergence in the iterative solution of the discretized pressure equations during the simulation of incompressible, Newtonian fluid flow. The prolongation and restriction operators are those implied by the additive-correction multigrid method [41–44,46–50]. Grid coarsening was achieved using an adaptive volume-agglomeration technique similar to those proposed originally by Raw [41] and Elias *et al.* [42].

In Chapter 4, an *h*-adaptive strategy based on *a posteriori* error estimation was presented. Its ultimate purpose is to optimize the CVFEM with respect to accuracy and computational effort, for any particular problem. This is achieved, in the present context, by locally refining the computational mesh during the solution process to meet a specified mesh optimality criterion. The key features of the iterative solution-adaptive

procedure include: (i) a recovery step, in which a higher-order projection of the CVFEM gradients is obtained; (ii) an error estimation step, whereby the higher-order projection and the original CVFEM solution are used to approximate the solution error; and (iii) a mesh refinement step, where the estimated error is used as a guide to refine the mesh based on a mesh-optimality criterion. In the current scheme, a node-insertion technique is used to refine the computational grid [40]. The present mesh-adaptive strategy is based on an amalgamation of ideas and concepts borrowed from Zienkiewicz and Zhu [1-4], and others [5-27]. This work constitutes the first attempt at incorporating unstructured-grid, solution-adaptive techniques into a CVFEM implementation in Professor B. R. Baliga's computational fluid dynamics and heat transfer group at McGill University.

In Chapter 5, an extended Richardson extrapolation technique for unstructured grids was proposed. This technique is based on key concepts first introduced by Richardson in 1910 [51] and 1927 [52]. In the classical Richardson extrapolation, two or more discrete solutions are obtained on uniform, ordered grids, with different mesh spacing. Then, an extrapolation is performed, using these discrete solutions, to approximate the exact solution to the analogous continuum problem.

In the proposed extended Richardson extrapolation procedure, a discrete solution is obtained on an unstructured grid with an arbitrary distribution of element shapes and sizes. It was demonstrated that it is advantageous if this initial grid is obtained using the aforementioned h -adaptive CVFEM, in which the estimated solution error is effectively reduced and uniformly distributed over each element (see Chapter 4): This initial grid is then used as a base grid in constructing a hierarchy of grids via uniform element subdivision. Each grid in the hierarchy is constructed by subdividing the elements of the *base* (or initial) grid into an integer number of similar elements. For example, the second grid in the hierarchy is constructed by subdividing the elements of the base grid into 4 similar elements, and the k^{th} grid in the hierarchy is constructed by subdividing the elements of the base grid into k^2 similar elements. By constructing the hierarchy of grids in this manner, each of the grids can be uniquely characterized with respect to the base grid, and all other grids, by specifying a single grid-size parameter, h_c . A standard extrapolation can now be performed with respect to h_c in order to approximate the exact solution to the analogous continuum problem, which, for a consistent numerical method,

occurs in the limit as $h_c \rightarrow 0$. To the author's knowledge, this is the first time this type of element subdivision technique has been used in the context of applying Richardson extrapolation to a series of unstructured-grid simulations. Many concepts and ideas in Chapter 5 were borrowed and/or extended from the works of Roache [53, 143], Celik and Zhang [145], Ferziger [146], Churchill *et al.* [147], and from discussions with the author's supervisor, Professor B. R. Baliga.

In Chapter 6, the numerical results of the thesis were presented. In Section 6.1, the additive correction multigrid solver that was outlined in Chapter 3 was tested on two model problems. Its efficiency, in terms of CPU time and effort, was demonstrated relative to the SOR Gauss-Seidel iterative point-by-point solver. The directional volume-agglomeration algorithm was also tested, and its advantage was demonstrated, by using a diffusion-type test problem in which the coefficients in the CVFEM discretized equations were given a directional bias by prescribing a non-uniform variation of the diffusion coefficient.

In Section 6.2, a particular solution to the steady, two-dimensional, incompressible Navier-Stokes equations was derived for the purposes of testing various numerical algorithms in subsequent sections of Chapter 6. Closed-form analytical expressions for a divergence-free velocity field and a consistent pressure field were prescribed and substituted into the momentum equations to derive the appropriate body-force terms. The resulting flow pattern depicts a series of recirculating eddies superimposed over a uniform base flow. To the author's knowledge this is an original solution to the Navier-Stokes equations, with the particular set of body force terms prescribed.

In Section 6.3, the aforementioned analytical flow field was simulated using the CVFEM presented in Chapter 2. The recovery schemes and error estimators employed in the h -adaptive procedure (presented in Chapter 4) were tested, and their capabilities and limitations were established, by comparing the numerical results with the corresponding analytical values. In Section 6.4, the proposed extended Richardson extrapolation technique was also tested and verified using results from CVFEM simulations of the analytical flow field.

Finally, in Section 6.5, lid-driven flow in a triangular cavity was simulated using the h -adaptive CVFEM. The strain-rate error estimator (Section 4.2.2) and flow-angle indicator (Section 4.2.3) were used independently, and in combination, in order to illustrate their effectiveness in driving the solution-adaptive procedure for this problem. It was also shown that the results obtained in this work compared favourably with the published solution of Jyotsna and Vanka [152].

7.2 Recommendations for Future Work

An Adaptive Hybrid Scheme for Approximating the Advection Term

In Section 2.4.4, two different interpolation schemes for approximating ϕ in the advection terms of the integral conservation equations were presented: the flow-oriented upwind scheme (FLO); and the mass-weighted upwind scheme (MAW). The FLO interpolation function automatically responds to an element-based Peclet number, and takes into account the direction of the element-average velocity vector. It has performed quite well in problems involving acute-angle triangular elements and relatively modest Reynolds or Peclet numbers [29,78,81]. The FLO scheme, however, can lead to negative contributions to the coefficients in the discretized equations when obtuse-angle triangular elements are used and/or when high Reynolds or Peclet number flows are solved with large cross-flow gradients of the dependent variables within an element. The MAW scheme was presented as an alternative to the FLO scheme, whenever the FLO scheme produces unacceptably large negative coefficients in the discretized equations. Although the MAW scheme guarantees positive coefficients, it is less accurate than the FLO scheme.

A formulation that would allow for automatic switching between the FLO and MAW schemes within any given element would be highly desirable. One such hybrid scheme could make use of the FLO scheme, by default, and switch to MAW in only those elements where FLO produced negative advection contributions to the coefficients in discretized equations. This hybrid scheme would likely benefit from the increased accuracy of the FLO scheme, and provide for a more stable and convergent iterative

solution procedure. A further enhancement to this type of hybrid scheme for h -adaptive CVFEM formulations would be to incorporate adaptive criteria based on the elimination of negative coefficients. During this adaptive solution procedure, an initial patch of elements that are required to be “MAW elements”, could be refined, subdivided, or reshaped in such a way as to produce a new patch of “FLO elements”, in which the FLO scheme no longer produced negative coefficient contributions. It is felt that the implementation of such a scheme might prove to be a fruitful endeavor in future work.

Improved Error Estimators for High-Reynolds Number Flows

The strain-rate error estimator, presented in Section 4.2.2, was strictly derived for creeping-low problems [7]. Since it is sensitive to high strain-rates and shear stresses by construction, it has also proven useful for low to moderate-Reynolds number flows [18,22]. However, as was shown in Section 6.3, the strain-rate error estimator is inappropriate for high-Reynolds number flows, where inertial forces dominate compared to viscous forces. It would be beneficial to develop and incorporate other error estimators that are better suited for such flows. Examples of other error estimators used for Navier-Stokes problems can be found in the works of Hetu and Pelletier [23], and Pelletier *et al.* [24].

Directional Mesh Adaptation

Efforts to obtain efficient numerical solutions of practical Navier-Stokes problems have necessitated the development of anisotropic, or directionally-stretched grid-adaptation techniques [56,95,111,112]. In high-Reynolds number viscous flows, proper resolution of thin boundary layers, for example, could require mesh spacing several orders of magnitude larger in the streamwise direction compared to the direction normal to the boundary. This would result in extremely large element aspect ratios in these regions. This author believes that the development and incorporation of an automatic, anisotropic mesh-adaptation algorithm into a CVFEM implementation would be a worthwhile undertaking.

REFERENCES

- [1] Zienkiewicz, O. C. and Zhu, J. Z., A Simple Error Estimator and Adaptive Procedure For Practical Engineering Analysis, *Int. J. Num. Methods in Eng.*, Vol. 24, pp. 337-357, 1987.
- [2] Zienkiewicz, O. C. and Zhu, J. Z., The Superconvergent Patch Recovery and A Posteriori Error Estimates. Part 1: The Recovery Technique, *Int. J. Num. Methods in Eng.*, Vol. 33, pp. 1331-1364, 1992.
- [3] Zienkiewicz, O. C. and Zhu, J. Z., The Superconvergent Patch Recovery and A Posteriori Error Estimates. Part 2: Error Estimates and Adaptivity, *Int. J. Num. Methods in Eng.*, Vol. 33, pp. 1365-1382, 1992.
- [4] Zienkiewicz, O. C. and Zhu, J. Z., Adaptivity and Mesh Generation, *Int. J. Num. Methods in Eng.*, Vol. 32, pp. 783-810, 1991.
- [5] Zienkiewicz, O. C., Liu, Y. C., Huang, G. C., Error Estimates and Convergence Rates for Various Incompressible Elements, *Int. J. Num. Methods in Eng.*, Vol. 28, pp. 2191-2202, 1989.
- [6] Zienkiewicz, O. C., Zhu, J. Z., Wu, J., Superconvergent Patch Recovery Techniques – Some Further Tests, *Comm. in Num. Methods in Eng.*, Vol. 9, pp. 251-258, 1993.
- [7] Zienkiewicz, O. C., Liu, Y. C., Huang, G. C., Error Estimates and Adaptivity in Flow Formulation for Forming Problems, *Int. J. Num. Methods in Eng.*, Vol. 25, pp. 23-42, 1988.
- [8] Zienkiewicz, O. C., Zhu, J. Z., Gong, N. G., Effective and Practical $h\text{-}p$ -Version Adaptive Analysis Procedures for the Finite Element Method, *Int. J. Num. Methods in Eng.*, Vol. 28, pp. 879-891, 1989.
- [9] Babuska, I. and Rheinboldt, W. C., A Posteriori Error Estimators in the Finite Element Method, *Int. J. Num. Methods in Eng.*, Vol. 12, pp. 1597-1615, 1978.

- [10] Babuska, I. and Rheinboldt, W. C., Error Estimates for Adaptive Finite Element Computations, *Siam J. Num. Anal.*, Vol. 15, pp. 736-754, 1978.
- [11] Babuska, I., Strouboulis, T., Upadhyay, C. S., A Model Study of the Quality of A Posteriori Error Estimators for Finite Element Solutions of Linear Elliptic Problems, with Particular Reference to the Behavior Near the Boundary, *Int. J. Num. Methods in Eng.*, Vol. 40, pp. 2521-2577, 1997.
- [12] Babuska, I., Strouboulis, T., Upadhyay, C. S., $\eta\%$ -Superconvergence of finite element approximations in the interior of general meshes of triangles, *Comput. Methods in Appl. Mech. And Engrg.*, Vol. 122, pp. 273-305, 1995.
- [13] Babuska, I. M. and Rodriguez, R., The Problem of the Selection of an A Posteriori Error Indicator Based on Smoothing Techniques, *Int. J. Num. Methods in Eng.*, Vol. 36, pp. 539-567, 1993.
- [14] Babuska, I. M., Duran, R., Rodriguez, R., Analysis of the Efficiency of an A Posteriori Error Estimator for Linear Triangular Finite Elements, *Siam J. Num. Anal.*, Vol. 29, pp. 947-964, 1992.
- [15] Ainsworth, M. and Oden, J. T., A Posteriori Error Estimation in Finite Element Analysis, *Comput. Methods in Appl. Mech. And Engrg.*, Vol. 142, pp. 1-88, 1997.
- [16] Ainsworth, M., Zhu, J. Z., Craig, A. W., Zienkiewicz, O. C., Analysis of the Zienkiewicz-Zhu A-Posteriori Error Estimator in the Finite Element Method, *Int. J. Num. Methods in Eng.*, Vol. 28, pp. 2161-2174, 1989.
- [17] Onate, E. and Bugeda, G., Mesh Optimality Criteria for Adaptive Finite Element Computations, Ch. 7, in *The Mathematics of Finite Elements and Applications* (Ed. Whiteman, J. R.), pp. 121-135, Wiley & Sons, Chichester, England, 1994.
- [18] Wu, J., Zhu, Z., Szmelter, J., Zienkiewicz, O. C., Error Estimation and Adaptivity in Navier-Stokes Incompressible Flows, *Computational Mechanics*, Vol. 6, pp. 259-270, 1990.
- [19] Hinton, E. and Campbell, J. S., Local and Global Smoothing of Discontinuous Finite Element Functions Using A Least Squares Method, *Int. J. Num. Methods in Eng.*, Vol. 8, pp. 461-480, 1974.

- [20] Wu, H. and Currie, I. G., Numerical Simulation of Viscous Incompressible Flows Using Posteriori Error Estimation, *CFD '97 Proceedings, 5th Annual Conference of the CFD Society of Canada*, pp. 5.25-5.30, Victoria, B.C., 1997.
- [21] Wu, H. and Currie, I. G., Adaptive Grid Computation of Viscous Flows Using an Object-Oriented Data Structure, *CFD '98 Proceedings, 6th Annual Conference of the CFD Society of Canada*, pp. XII-31 – XII-36, Quebec City, Quebec, 1998.
- [22] Hetu, J. F. and Pelletier, D. H., Adaptive Remeshing for Viscous Incompressible Flows, *AIAA Journal*, Vol. 30, No. 8, 1992.
- [23] Hetu, J. F. and Pelletier, D. H., Fast, Adaptive Finite Element Scheme for Viscous Incompressible Flows, *AIAA Journal*, Vol. 30, No. 11, 1992.
- [24] Pelletier, D. H., Hetu, J. F., Ilinca, F., Adaptive Finite Element Method for Thermal Flow Problems, *AIAA Journal*, Vol. 32, No. 4, 1994.
- [25] Labbé, P. and Garon, A., A Robust Implementation of Zienkiewicz and Zhu's Local Patch Recovery Method, *Communications in Numerical Methods in Engineering*, Vol. 11, pp. 427-434, 1995.
- [26] Usmani, A. S., Solution of Advection Problems Using *h*-Adaptive FEM with Discontinuity Capturing SUPG Method, *Int. Symp. on Advances in Computational Heat Transfer: Book of Abstracts*, Cesme, Turkey, 1997.
- [27] Zhang, Z. and Zhu, J., Analysis of the Superconvergent Patch Recovery Technique and A Posteriori Error Estimator in the Finite Element Method (I), *Comput. Methods Appl. Mech. Engrg.*, Vol 123, pp. 173-187, 1995.
- [28] Baliga, B. R., A Control-Volume Based Finite Element Method for Convective Heat and Mass Transfer, Ph.D. Thesis, University of Minnesota, Minneapolis, U.S.A., 1978.
- [29] Baliga, B. R. and Patankar, S. V., A New Finite Element Formulation for Convection-Diffusion Problems, *Numer. Heat Transfer*, Vol. 3, pp. 393-409, 1980.
- [30] Prakash C. and Patankar, S. V., A Control-Volume-Based Finite-Element Method for Solving the Navier-Stokes Equations Using Equal-Order Velocity-Pressure Interpolation, *Numer. Heat Transfer*, Vol. 8, pp. 259-280, 1985.

- [31] Raw, M., A New Control-Volume-Based Finite Element Procedure for the Numerical Solution of the Fluid Flow and Scalar Transport Equations, Ph.D. Thesis, University of Waterloo, Waterloo, Canada, 1985.
- [32] Hookey, N. A., A CVFEM for Two-Dimensional Viscous Compressible Fluid Flow, Ph.D. Thesis, McGill University, Montreal, Canada, 1989.
- [33] Saabas, H. J., A Control-Volume Finite Element Method for Three-Dimensional, Incompressible, Viscous Fluid Flow, Ph.D. Thesis, McGill University, Montreal, Canada, 1991.
- [34] Masson, C., Numerical Prediction of Gas-Solid Particle Flows Over a Wide Range of Concentration in Irregular Geometries, Ph.D. Thesis, McGill University, Montreal, Canada, 1993.
- [35] Rousse, D. R., Numerical Prediction of Multidimensional Conduction, Convection, and Radiation Heat Transfer in Participating Media, Ph.D. Thesis, McGill University, Montreal, Canada, 1994.
- [36] Elkouh, N., Laminar Natural Convection and Interfacial Heat Flux Distributions in Pure Ice-Water Systems, Ph.D. Thesis, McGill University, Montreal, Canada, 1996.
- [37] Afshar, M., Investigation and Improvement of FVMs and CVFEMs for Fluid Flow, M.Eng. Thesis, McGill University, Montreal, Canada, 1992.
- [38] Afshar, M., Numerical Predictions of Fully-Developed, Turbulent, Single-Phase and Bubbly Two-Phase Flows in Straight Ducts, Ph.D. Thesis, McGill University, Montreal, Canada, 1998.
- [39] Mavriplis, D. J., Adaptive Mesh Generation for Viscous Flows Using Delaunay Triangulation, *Journal of Comp. Physics*, Vol. 90, pp. 271-291, 1990.
- [40] Shewchuk, J. R., Triangle: Engineering a 2D Quality Mesh Generator and Delauney Triangulator, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1996.
- [41] Raw, M., Coupled Algebraic Multigrid for the Solution of the Discretized 3D Navier-Stokes Equations, *CFD '94 Proceedings, 2nd Annual Conference of the CFD Society of Canada*, pp. 335-344, Toronto, Ontario, 1994.

- [42] Elias, S. R., Stuble, G. D., Raithby, G. D., An Adaptive Agglomeration Method for Additive Correction Multigrid, *Int. J. Num. Methods in Eng.*, Vol. 40, pp. 887-903, 1997.
- [43] Hutchinson, B. R., Galpin, P. F., Raithby, G. D., Applications of Additive Correction Multigrid to the Coupled Fluid Flow Equations, *Numer. Heat Transfer*, Vol. 13, pp. 133-147, 1988.
- [44] Griebel, M., Neunhoeffer, T., Regler, H., Algebraic Multigrid Methods for the Solution of the Navier-Stokes Equations in Complicated Geometries, *Int. J. Num. Methods in Fluids*, Vol. 26, pp. 281-301, 1998.
- [45] Koobus, B., Lallemand, M., Dervieux, A., Unstructured Volume-Agglomeration MG: Solution of the Poisson Equation, *Int. J. Num. Methods in Fluids*, Vol. 18, pp. 27-42, 1994.
- [46] Chang, Q., Wong, Y., Fu, H., On the Algebraic Multigrid Method, *J. Comp. Physics*, Vol. 125, pp. 279-292, 1996.
- [47] Brandt, A., Multi-Level Adaptive Solutions to Boundary Value Problems, *Math. Comput.*, Vol. 31, pp. 330-390, 1977.
- [48] Brandt, A., Guide to Multigrid Development, in *Multigrid Methods* (eds. Hackbusch, W and Trottenberg, U.), Lecture Notes in Mathematics (eds. Dold, A. and Eckmann, B.), Springer-Verlag, Berlin, Germany, pp. 220-312, 1982.
- [49] Briggs, W. L., *A Multigrid Tutorial*, Society for Industrial and Applied Mathematics, Philadelphia, 1987.
- [50] Mavriplis, D. J., Multigrid Techniques for Unstructured Meshes, *ICASE Report No. 95-27, NASA Contractor Report 195070*, 61 pages, 1995.
- [51] Richardson, L. F., The Approximate Arithmetic Solution by Finite Differences of Physical Problems Involving Differential Equations with an Application to Stresses in a Masonry Dam, *Trans. Roy. Soc. London, Series A*, Vol. 210, pp. 307-357, 1910.
- [52] Richardson, L. F., The Deferred Approach to the Limit, *Trans. Roy. Soc. London, Series A*, Vol. 226, pp. 299-361, 1927.
- [53] Roache, P. J., Quantification of Uncertainty in Computational Fluid Dynamics, *Annu. Rev. Fluid Mech.*, Vol. 29, pp. 123-160, 1997.

- [54] Bauchau, O. A. and Ahmad, J. U., Advanced CFD and CSD Methods for Multidisciplinary Applications in Rotorcraft Problems, *AIAA Paper 96-4151-CP*, pp. 1441-1451, 1996.
- [55] Kenny, D. P., The Design of Modern Gas Turbine Design: Beyond CFD, (Invited paper), *CFD '98 Proceedings, 6th Annual Conference of the CFD Society of Canada*, pp. I-3 – I-7, Quebec City, Quebec, 1998.
- [56] Mavriplis, D. J., Unstructured Grid Techniques, *Annu. Rev. Fluid Mech.*, Vol. 29, pp. 473-514, 1997.
- [57] Barth, T. J., On Unstructured Grids and Solvers, *Computational Fluid Dynamics, von Karman Inst. Lecture Series*, 1990-03, 1990.
- [58] Finlayson, B. A., *The Method of Weighted Residuals and Variational Principles*, Academic Press, New York, 1972.
- [59] Zienkiewicz, O. C., Why Finite Elements, in *Finite Elements in Fluids* (eds. Gallagher, R. H. et al), Vol. 1, Ch. 1, John Wiley and Sons, New York, 1975.
- [60] Baliga, B. R., Control-Volume Finite Element Methods for Fluid Flow and Heat Transfer, Ch. 3, in *Advances in Numerical Heat Transfer* (Eds. W. J. Minkowycz and E. M. Sparrow), pp. 97-135, Taylor & Francis, Washington D. C., 1997.
- [61] McCormick, S. F., Multilevel Adaptive Methods for Partial Differential Equations, in *Frontiers in Applied Mathematics*, Vol. 6, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1989.
- [62] Jameson, A. and Mavriplis, D., Finite Volume Solution of the Two-Dimensional Euler Equations on a Regular Mesh, *AIAA Paper 85-0435*, 1985.
- [63] Arminjon, P. and Madrane, A., A Staggered Lax-Friedrichs Type Finite Volume/Finite Element Method for the Two-Dimensional Navier-Stokes Equations, *CFD '98 Proceedings, 6th Annual Conference of the CFD Society of Canada*, pp. VIII-103 – VIII-108, Quebec city, Quebec, 1998.
- [64] Roache, P. J., Ghia, K., White, F., Editorial Policy Statement on the Control of Numerical Accuracy, *ASME J. Fluids Engrg.*, Vol. 108, pp.2, 1986.
- [65] ASME Editorial Board, *Journal of Heat Transfer* editorial policy statement on numerical accuracy, *ASME J. Heat Transfer*, Vol. 116, pp. 797-798, 1994.

- [66] AIAA, Policy Statement on Numerical Accuracy and Experimental Uncertainty, *AIAA J.*, Vol. 32, pp. 3, 1994.
- [67] Gresho, P. M. and Taylor, C., (editorial), *Int. J. Num. Meth. Fluids*, Vol. 19, pp. iii, 1994.
- [68] Boehm, B. W., *Software Engineering Economics*, Prentice-Hall, New York, 1981.
- [69] Blottner, F. G., Accurate Navier-Stokes Results for the Hypersonic Flow over a Spherical Nose tip, *AIAA J. Spacecraft Rockets*, Vol. 27, pp. 113-122, 1990.
- [70] Chorin A. J., Numerical Solution of the Navier-Stokes Equations, *Math. Comput.* Vol. 22, pp. 745-762, 1968.
- [71] Patankar, S. V. and Spalding, D. B., A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-Dimensional Parabolic Flows, *Int. J. Heat Mass Tranfer*, Vol. 15, pp. 1787-1806, 1972.
- [72] Raithby, G. D., Skew Upstream Differencing Schemes for Problems Involving Fluid Flow, *Comp. Meth. App. Mech. And Engrg.*, Vol. 9, pp. 153-164, 1976.
- [73] Zienkiewicz, O. C., *The Finite Element Method*, 3rd Ed., McGraw-Hill, New York, 1977.
- [74] Oden, J. T., Finite-Element Analogue of the Navier-Stokes Equations, *J. Eng. Mech. Div., ASCE*, Vol. 96, No. EM4, pp. 529-534, 1970.
- [75] Taylor, C. and Hood, P., A Numerical Solution of the Navier-Stokes Equations Using the Finite Element Technique, *Computers and Fluids*, Vol. 1, pp. 73-100, 1973.
- [76] Brookes A. N. and Hughes, T. J. R., Streamline Upwind / Petrov-Galerkin Formulations for Convection Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equations, *Comp. Meths. Appl. Mech. Eng.*, Vol. 32, pp. 199-259, 1982.
- [77] Saabas, H. J. and Baliga, B. R., A Co-Located Equal-Order Control-Volume Finite Element Method for Multidimensional, Incompressible Fluid Flow – Part I: Formulation, *Numer. Heat Transfer, Part B*, Vol. 26, pp. 381-407, 1994.

- [78] Saabas, H. J. and Baliga, B. R., A Co-Located Equal-Order Control-Volume Finite Element Method for Multidimensional, Incompressible Fluid Flow – Part II: Verification, *Numer. Heat Transfer, Part B*, Vol. 26, pp. 409-424, 1994.
- [79] Prakash, C., Examination of the Upwind (Donor-Cell) Formulation in Control-Volume Finite-Element Methods for Fluid Flow and Heat Transfer, *Numer. Heat Transfer*, Vol. 11, pp. 401-416, 1987.
- [80] Schneider, G. E. and Raw, M. J., A Skewed Positive Influence Coefficient Upwind Procedure for Control-Volume-Based Finite Element Convection-Diffusion Computation, *Numer. Heat Transfer*, Vol. 9, pp. 1-26, 1986.
- [81] Masson, C., Saabas, H. J., Baliga, B. R., Co-Located Equal-Order Control-Volume Finite Element Method for Two-Dimensional Axisymmetric Incompressible Fluid Flow, *Int. J. Numer. Methods in Fluids*, Vol. 18, pp. 1-26, 1994.
- [82] Patankar, S. V., *Numerical Heat Transfer and Fluid Flow*, Hemisphere/McGraw-Hill, New York, 1980.
- [83] Baliga, B. R. and Patankar, S. V., A Control-Volume Finite Element Method for Two-Dimensional Incompressible Fluid Flow and Heat Transfer, *Numer. Heat Transfer*, Vol. 6, pp. 245-261, 1983.
- [84] Baliga, B. R. and Patankar, S. V., Elliptic Systems: Finite-Element Method II, in *Handbook of Numerical Heat Transfer*, (eds. Minkowycz, W. J. *et al.*), Chap. 11, John Wiley and Sons, New York, pp. 421-461, 1988.
- [85] Prakash, C., An Improved Control-Volume Finite-Element Method for Heat and Mass Transfer, and for Fluid Flow Using Equal-Order Velocity-Pressure Interpolation, *Numer. Heat Transfer*, Vol. 9, pp. 253-276, 1986.
- [86] Schneider, G. E., Raithby, G. D., Yovanovich, M. M., Finite-Element Solution Procedures for Solving the Incompressible Navier-Stokes Equations Using Equal-Order Variable Interpolation, *Numer. Heat Transfer*, Vol. 1, pp. 433-451, 1978.
- [87] Rice, J. G. and Schnipke, R. J., An Equal-Order Velocity-Pressure Formulation that Does Not Exhibit Spurious Pressure Modes, *Comp. Meth. in App. Mech. and Eng.*, Vol. 58, pp. 135-149, 1986.
- [88] Harlow, F. H. and Welch, J. E., Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with a Free Surface, *Phys. of Fluids*, Vol. 8, pp. 2182-2189, 1965.

- [89] Rhee, C. M. and Chow, W. L., Numerical Study of the Turbulent Flow Past an Airfoil with Trailing Edge Separation, *AIAA Journal*, Vol. 21, pp. 1525-1532, 1983.
- [90] Peric, M., Kessler, R., Scheurer, G., Comparison of Finite-Volume Numerical Methods with Staggered and Co-Located Grids, *Computers and Fluids*, Vol. 16, pp. 389-403, 1988.
- [91] Hookey, N. A., and Baliga, B. R., Evaluation and Enhancements of Some Control-Volume Finite-Element Methods – Part II: Fluid Flow Problems, *Numer Heat Transfer*, Vol. 14, pp. 273-293, 1988.
- [92] Baliga, B. R., Some Comments on the Computation of Fluid Flow and Heat Transfer in Irregular Geometries, (Invited paper), *CFD '98 Proceedings, 6th Annual Conference of the CFD Society of Canada*, pp. I-9 – I-14, Quebec City, Quebec, 1998.
- [93] Preparata, F. P. and Shamos, M. I., *Computational Geometry, An Introduction*, Texts Monogr. Comput. Sci., New York: Springer-Verlag, 1985.
- [94] Lo, S. H., A New Mesh Generation Scheme for Arbitrary Planar Domains, *Int. J. Numer. Methods Eng.* Vol. 21, pp. 1403-1426, 1985.
- [95] Peraire, J., Vehdati, M., Morgan, K., Zienkiewicz, O. C., Adaptive Remeshing for Compressible Flow Computations, *J. Comput. Phys.*, Vol. 72, pp. 449-466, 1987.
- [96] Gumbert, C., Lohner, R., Parikh, P., Pirzadeh, S., A Package for Unstructured Grid Generation and Finite Element Flow Solvers, *AIAA Pap. 89-2175*, 1989.
- [97] Bowyer, A., Computing Dirichlet Tessellations, *Comput. J.*, Vol. 24, pp. 162-166, 1981.
- [98] Watson, D. F., Computing the n-Dimensional Delaunay Tessellation with Application to Voronoi Polytopes, *Comput. J.*, Vol. 24, pp. 167-171, 1981.
- [99] Baker, T. J., Three Dimensional Mesh Generation by Triangulation of Arbitrary Point Sets, *AIAA Pap. 87-1124*, 1987.
- [100] Weatherill, N. P., The Generation of Unstructured Grids Using Dirichlet Tessellations, *Princeton Univ. Dept. Mech. Aerospace Eng. Rep. MAE 1715*, 1985.

- [101] Mavriplis, D. J., Accurate Multigrid Solution of the Euler Equations on Unstructured and Adaptive Meshes, *AIAA J.* Vol. 28, pp. 213-221, 1990.
- [102] Holmes, D.G., and Snyder, D. D., The Generation of Unstructured Meshes Using Delaunay Triangulation, *Numer. Grid Gener. Comput. Fluid Mech. Proc. Int. Conf. Numer. Grid Gener. Comput. Fluid Dyn.*, 2nd, Miami, (eds. Sengupta, S. et al.), UK: Pineridge, 1988.
- [103] George, P. L., Hecht, F., Saltel, E., Automatic Mesh Generator with Specified Boundary, *Comput. Methods Appl. Mech. Eng.*, Vol. 33, pp. 975-995, 1991.
- [104] Weatherill, N. P., Hassan, O., Marcum, D. L., Calculation of Steady Compressible Flowfields with the Finite Element Method, *AIAA Paper 93-0341*, 1993.
- [105] Fortune, S., Voronoi Diagrams and Delaunay Triangulations, in *Computing in Euclidean Geometry, Lecture Notes Series on Computing* (eds. Du, D. Z. and Hwang, F.), World Scientific, Singapore, Vol. 1, pp. 193-233, 1992.
- [106] Su, P. and Drysdale, R. L. S., A Comparison of Sequential Delaunay Triangulation Algorithms, *Proc. of the Eleventh Annual Symp. on Comp. Geom.*, Association for Computing Machinery, pp. 61-70, 1995.
- [107] Yerry, M. A. and Shephard, M. S., Automatic Three-Dimensional Mesh Generation by the Modified-Octree Technique, *Int. J. Numer. Methods Eng.*, Vol. 20, pp. 1965-1990, 1984.
- [108] Rebay, S., Efficient Unstructured Mesh Generation by Means of Delaunay Triangulation and Bowyer/Watson Algorithm, *J. Comput. Phys.*, Vol. 106, pp. 125-138, 1993.
- [109] Mueller, J. D., Roe, P. L., Deconinck, H., A Frontal Approach for Node Generation in Delaunay Triangulations, *VKI Lect. Ser., AGARD Publ.*, R-787, 1992.
- [110] Babuska, I. and Aziz, A. K., On the Angle Condition in the Finite-Element Method, *SIAM J. Numer. Anal.*, Vol. 13, 1976.
- [111] Mavriplis, D. J., Unstructured and Adaptive Mesh Generation for High-Reynolds Number Viscous Flows, *Proc. of the International Conf. on Numer. Grid Generation: CFD and Related Fields*, 3rd, Barcelona, Spain, (eds. Arcilla, A. S. et al.), pp. 79-92, New York: North-Holland, 1991.

- [112] Castro-Diaz, M. J., Hecht, F., Mohammadi, B., Anisotropic Unstructured Mesh Adaptation for Flow Simulations, *Int. Mesh Roundtable, 4th*, Albuquerque, NM, pp. 73-85, 1995.
- [113] Nakahashi, K., FDM-FEM Approach for Viscous Flow Computations over Multiple Bodies, *AIAA Pap. 87-0604*, 1987.
- [114] Ward, S. and Kallinderis, Y., Hybrid Prismatic/Tetrahedral Grid Generation for Complex 3D Geometries, *AIAA Pap. 93-0669*, 1993.
- [115] Lohner, R., Matching Semi-Structured and Unstructured Grids for Navier-Stokes Calculations, *AIAA Pap. 93-3348*, 1993.
- [116] Pirzadeh, S., Unstructured Viscous Grid Generation by the Advancing-Layers Method, *AIAA J.*, Vol. 32, pp. 1735-1737, 1994.
- [117] Pirzadeh, S., Viscous Unstructured Three-Dimensional Grids by the Advancing-Layers Method, *AIAA Pap. 94-0417*, 1994.
- [118] Connell, S. D. and Braaten, M. E., Semi-Structured Mesh Generation for Three-Dimensional Navier-Stokes Calculations, *AIAA J.*, Vol. 33, pp. 1017-1024, 1995.
- [119] Blacker, T. D., Stephenson, M. B., Paving a New Approach to Automated Quadrilateral Mesh Generation, *Int. J. Numer. Methods Eng.*, Vol. 32, pp. 811-847, 1991.
- [120] Merriam, M. L., An Efficient Advancing Front Algorithm for Delaunay Triangulation, *AIAA Pap. 91-0792*, 1991.
- [121] Tembulkar, J. M. and Hanks, B. W., On Generating Quadrilateral Elements from a Triangular Mesh, *Comput. Struct.*, Vol. 42, pp. 665-667, 1992.
- [122] Babuska, I., Szabo, B. A., Katz, I. N., The p -Version of the Finite Element Method, *SIAM J. Numer. Anal.*, Vol. 18, pp. 512-545, 1981.
- [123] Zienkiewicz, O. C., and Craig, A. W., Adaptive Mesh Refinement and *A Posteriori* Error Estimation for the p -Version of the Finite Element Method (eds. Babuska, I. et al.), *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, Wiley, New York, pp. 25-59, 1986.
- [124] Palmerio, B., An Attraction-Repulsion Mesh Adaptation Model for Flow Solution on Unstructured Grids, *Comput. Fluids*, Vol. 23, pp. 487-506, 1994.

- [125] Reuss, S. and Stuble, G. D., Grid Adaptation Based on Flux Errors for One-Dimensional Transport, *CFD '98 Proceedings, 6th Annual Conference of the CFD Society of Canada*, pp. XII-43 – XII-48, Quebec city, Quebec, 1998.
- [126] Baum, J. D., Luo, H., Lohner, R., A New ALE Adaptive Unstructured Methodology for the Simulation of Moving Bodies, *AIAA Pap. 94-0414*, 1994.
- [127] Guo, B. and Babuska, I., The $h\text{-}p$ -version of the Finite Element Method. Part 1: The Basic Approximation Results. Part 2: General Results and Applications, *Comp. Mech.*, Vol. 1, pp. 21-41, 203-226, 1986.
- [128] Anderson, J. D., *Computational Fluid Dynamics: The Basics with Applications*, McGraw-Hill, New York, 1995.
- [129] Ladeveze, P. and Leguillon, D., Error Estimate Procedure in the Finite Element Method and Applications, *SIAM J. Numer. Anal.*, Vol. 20, pp. 485-509, 1983.
- [130] Kelly, D. W., The Self-Equilibration of Residuals and Complementary A Posteriori Error Estimates in the Finite Element Method, *Int. J. Numer. Methods Engrg.*, Vol. 20, pp. 1491-1506, 1984.
- [131] Demkowicz, L., Oden, J. T., Strouboulis, T., Adaptive Finite Elements for Flow Problems with Moving Boundaries, Part 1: Variational Principles and A Posteriori Error Estimates, *Comput. Methods Appl. Mech. Engrg.*, Vol. 46, pp. 217-251, 1984.
- [132] Bank, R. E. and Weiser, A., Some A Posteriori Error Estimators for Elliptic Partial Differential Equations, *Math. Comput.*, Vol. 44, pp. 283-301, 1985.
- [133] Szabo, B. A., Mesh Design for the p -Version of the Finite Element Method, *Comput. Methods Appl. Mech. Engrg.*, Vol. 55, pp. 181-197, 1986.
- [134] Venditti, D. A. and Baliga, B. R., An h -Adaptive Strategy for CVFEM Simulations of Viscous Incompressible Flow, *CFD '98 Proceedings, 6th Annual Conference of the CFD Society of Canada*, pp. VIII-65 – VIII-70, Quebec city, Quebec, 1998.
- [135] Zingg, D. W., Grid Studies for Thin-Layer Navier-Stokes Computations of Airfoil Flowfields, *AIAA J.*, Vol. 30, pp. 2561-2564, 1992.
- [136] Zingg, D. W., De Rango, S., Nemec, M., An Investigation of Numerical Errors in Aerodynamic Flow Computations, *CFD '98 Proceedings, 6th Annual Conference of the CFD Society of Canada*, pp. II-31 – II-36, Quebec city, Quebec, 1998.

- [137] Cook, R. D., Malkus, D. S., Plesha, M. E., *Concepts and Applications of Finite Element Analysis, 3rd edition*, John Wiley & Sons, New York, 1989.
- [138] Afshar, M. and Baliga, B. R., A Sequential Solution Algorithm for the Solution of the Discretized Momentum and Continuity Equations, *Proc. 8th Int. Conf. on Num. Methods in Laminar and Turbulent Flow*, Swansea, U.K., Vol. 8, Part 1, pp. 668-679, 1993.
- [139] Hutchinson, B. R., Numerical Simulation of River Flows, Ph.D. Thesis, University of Waterloo, Waterloo, Ontario, Canada, 1986.
- [140] Tuma, J. J., *Handbook of Numerical Calculations in Engineering*, McGraw-Hill, New York, 1989.
- [141] Huang, H. C. and Usmani, A. S., *Finite Element Analysis for Heat Transfer, Theory and Software*, Springer-Verlag, London, Great Britain, 1994.
- [142] Bouillard, P., Allard, J. F., Warzee, G., Superconvergent Patch Recovery Technique for the Finite Element Method in Acoustics, *Comm. in Num. Methods in Engineering*, Vol. 12, pp. 581-594, 1996.
- [143] Roache, P. J., Perspective: A Method for Uniform Reporting of Grid Refinement Studies, *Journal of Fluids Engineering*, Vol. 116, pp. 405-413, 1994.
- [144] Roache, P. J., *Computational Fluid Dynamics*, Hermosa Publisher, Albuquerque, New Mexico, 1972.
- [145] Celik, I. and Zhang, W. M., Calculation of Numerical Uncertainty Using Richardson Extrapolation: Application to Some Turbulent Flow Calculations, *Journal of Fluids Engineering*, Vol. 117, pp. 439-445, 1995.
- [146] Ferziger, J. H., Estimation and Reduction of Numerical Error, *Proc. of the Symp. on Quantification of Uncertainty in Computational Fluid Dynamics*, (eds. Celik et al.), ASME Fluids Engineering Division Spring Meeting, Washington DC., pp.1-8, 1993.
- [147] Churchill, S. W., Chao, P., Ozoe, H., Extrapolation of Finite Difference Calculations of Laminar Natural Convection in Enclosures to Zero Grid Size, *Numerical Heat Transfer*, Vol. 4, pp. 39-51, 1981.

- [148] Caruso, S. C., Ferziger, J. H., Oliger, J., Adaptive Grid Techniques for Elliptic Flow Problems, Rept. No. TF-23, Thermosciences Div., Stanford University, Stanford, CA, 1985.
- [149] Smith, R. E. and Kidd, A., Comparative Study of Two Numerical Techniques for the Solution of Viscous Flow in a Driven Cavity, *NASA SP-378*, pp. 61-82, 1975.
- [150] Ghia, K. N., Hankey, W. L., Hodge, J. K., Study of Incompressible Navier-Stokes Equations in Primitive Variables Using Implicit Numerical Technique, *AIAA Paper No. 77-648*, 1977; *AIAA J.*, Vol. 17, 1979.
- [151] Ghia, U., Ghia, K. N., Shin, C. T., High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method, *J. Comp. Physics*, Vol. 48, pp. 387-411, 1982.
- [152] Jyotsna, R. and Vanka, S. P., Multigrid Calculation of Steady, Viscous Flow in a Triangular Cavity, *Journal of Computational Physics*, Vol. 122, pp. 107-117, 1995.
- [153] Ribbens, C. J., Watson, L. T., Wang, C. Y., Steady Viscous Flow in a Triangular Cavity, *J. Comp. Physics*, Vol. 112, pp. 173-181, 1994.
- [154] Moffatt, H. K., Viscous and Resistive Eddies Near a Sharp Corner, *J. Fluid Mech.*, Vol. 18, pp. 1-18, 1964.
- [155] Jameson, A., Essential Elements of Computational Algorithms for Aerodynamic Analysis and Design, *Inst. for Comp. Appl. in Science and Engrg.*, NASA/CR-97-206268, ICASE Report No. 97-68, 1997.

Appendix A

Triangle: An Unstructured Delaunay-Based Mesh Generator

Most of the meshes used by the present h -adaptive CVFEM were generated and refined using *Triangle* - a two-dimensional, unstructured, Delaunay-based mesh generator developed by J. R. Shewchuk [40].

Capabilities. Given a set of points defining a piecewise-linear approximation of a planar domain boundary, *Triangle* is capable of automatically generating a constrained, or conforming Delaunay triangulation of the corresponding domain. This domain may contain holes or concavities. A minimum-vertex-angle constraint can be placed on all triangles in the domain up to a maximum of 33.8 degrees. Global mesh refinement is controlled by specifying a fixed maximum-area-constraint, which applies to every element in the domain. Local mesh refinement is controlled by specifying a maximum-area-constraint for each individual triangle in the domain. Mesh refinement is achieved via node insertion. *Triangle* is not capable of mesh coarsening, node movement, or element stretching.

Algorithms. The reader is referred to a paper by Shewchuk [40] in which he discusses many of the key implementation decisions, including the choice of triangulation algorithms and data structures, the steps taken to create and refine a mesh, and other pertinent issues.

Availability. Further information on *Triangle*, as well as the C source code for the program itself, is freely provided by Shewchuk on the Web at:

<http://www.cs.cmu.edu/~quake/triangle.html>

Appendix B

Derivation of the Sufficient Conditions for Positive Diffusion Contributions to the Coefficients in the Discretized Equations

In Section 2.4.3.1, a summary of the sufficient conditions required to ensure positive diffusion contributions to the coefficients in the discretized equations was presented. Those statements are proven in this appendix.

The notation used in the following derivation is represented in Figs. B.1 – B.3. In Fig. B.1, the patch of elements surrounding a typical interior node i is shown. There are d_i neighbour nodes surrounding node i . Local node numbering and the labeling of specific points within the k^{th} element surrounding node i is shown in Fig. B.2. The following normal vectors are defined with respect to element k in Fig. B.2:

$$\bar{\mathbf{N}}_r^k = (y_o - y_a)\bar{\mathbf{i}} - (x_o - x_a)\bar{\mathbf{j}}, \quad (\text{B.1})$$

$$\bar{\mathbf{N}}_s^k = (y_o - y_b)\bar{\mathbf{i}} - (x_o - x_b)\bar{\mathbf{j}}, \quad (\text{B.2})$$

$$\bar{\mathbf{N}}_t^k = (y_o - y_c)\bar{\mathbf{i}} - (x_o - x_c)\bar{\mathbf{j}}, \quad (\text{B.3})$$

$$\begin{aligned} \bar{\mathbf{N}}_o^k &= (y_k - y_i)\bar{\mathbf{i}} - (x_k - x_i)\bar{\mathbf{j}} \\ &= 2[(y_b - y_c)\bar{\mathbf{i}} - (x_b - x_c)\bar{\mathbf{j}}] \\ &= 2(\bar{\mathbf{N}}_s^k - \bar{\mathbf{N}}_t^k), \end{aligned} \quad (\text{B.4})$$

$$\begin{aligned} \bar{\mathbf{N}}_b^k &= (y_{k+1} - y_k)\bar{\mathbf{i}} - (x_{k+1} - x_k)\bar{\mathbf{j}} \\ &= 2[(y_c - y_a)\bar{\mathbf{i}} - (x_c - x_a)\bar{\mathbf{j}}] \\ &= 2(\bar{\mathbf{N}}_r^k - \bar{\mathbf{N}}_t^k), \end{aligned} \quad (\text{B.5})$$

$$\begin{aligned}
\bar{\mathbf{N}}_c^k &= (y_i - y_{k+1})\bar{\mathbf{i}} - (x_i - x_{k+1})\bar{\mathbf{j}} \\
&= 2[(y_a - y_b)\bar{\mathbf{i}} - (x_a - x_b)\bar{\mathbf{j}}] \\
&= 2(\bar{\mathbf{N}}_s^k - \bar{\mathbf{N}}_r^k).
\end{aligned} \tag{B.6}$$

In the diffusion terms, the dependent variable, ϕ , is interpolated linearly over each element. Within the k^{th} element surrounding node i , we have,

$$\phi_k = A_k x + B_k y + C_k. \tag{B.7}$$

The gradient is, therefore, given by,

$$\bar{\nabla}\phi_k = A_k \bar{\mathbf{i}} + B_k \bar{\mathbf{j}}, \tag{B.8}$$

where, $A_k = \frac{1}{2\Omega_k} [(y_k - y_{k+1})\phi_i + (y_{k+1} - y_i)\phi_k + (y_i - y_k)\phi_{k+1}],$ (B.9)

and, $B_k = -\frac{1}{2\Omega_k} [(x_k - x_{k+1})\phi_i + (x_{k+1} - x_i)\phi_k + (x_i - x_k)\phi_{k+1}].$ (B.10)

Note that $2\Omega_k = (x_i - x_{k+1})(y_k - y_{k+1}) - (x_k - x_{k+1})(y_i - y_{k+1})$, where Ω_k is the volume (area times unit depth) of the k^{th} element. Substituting Eqs. (B.9) and (B.10) into Eq. (B.8) and rearranging yields,

$$\begin{aligned}
\bar{\nabla}\phi_k &= \frac{1}{2\Omega_k} \left\{ \left[(y_k - y_{k+1})\bar{\mathbf{i}} - (x_k - x_{k+1})\bar{\mathbf{j}} \right] \phi_i \right. \\
&\quad + \left[(y_{k+1} - y_i)\bar{\mathbf{i}} - (x_{k+1} - x_i)\bar{\mathbf{j}} \right] \phi_k \\
&\quad \left. + \left[(y_i - y_k)\bar{\mathbf{i}} - (x_i - x_k)\bar{\mathbf{j}} \right] \phi_{k+1} \right\}.
\end{aligned} \tag{B.11}$$

Using the expressions given by Eqs. (B.4)–(B.6), Eq. (B.11) can be recast into the following form:

$$\bar{\nabla}\phi_k = -\frac{1}{2\Omega_k} [\bar{\mathbf{N}}_b^k \phi_i + \bar{\mathbf{N}}_c^k \phi_k + \bar{\mathbf{N}}_a^k \phi_{k+1}]. \tag{B.12}$$

The total diffusion contribution, I_{Ω_i} , to the rate of transport of ϕ out of the control volume, Ω_i , surrounding node i is given by,

$$I_{\Omega_i} = \int_{\Omega_i} (\bar{\nabla} \cdot \bar{\mathbf{J}}_D) d\Omega = \oint_{\partial\Omega_i} \bar{\mathbf{J}}_D \cdot d\bar{\mathbf{N}}, \quad (\text{B.13})$$

where $\bar{\mathbf{J}}_D = -\Gamma_\phi \bar{\nabla} \phi$ is the diffusion flux associated with the scalar variable, ϕ . In the expressions that follow, the periodic index, $k = [1, 2, \dots, d_i, 1, 2, \dots]$, will be adopted. Using Eq. (B.12) and (B.5), the algebraic approximation to Eq. (B.13) can be written as,

$$I_{\Omega_i} = \sum_{k=1}^{d_i} \bar{\mathbf{J}}_D^k \cdot (\bar{\mathbf{N}}_r^k - \bar{\mathbf{N}}_t^k) = \frac{1}{2} \sum_{k=1}^{d_i} \bar{\mathbf{J}}_D^k \cdot \bar{\mathbf{N}}_b^k, \quad (\text{B.14})$$

where,

$$\bar{\mathbf{J}}_D^k = -\Gamma_\phi^k \bar{\nabla} \phi_k. \quad (\text{B.15})$$

Note that in writing Eq. (B.14), we have approximated Γ_ϕ as being piecewise-constant over each element. Using Eq. (B.12) and (B.15), Eq. (B.14) can be rewritten as,

$$I_{\Omega_i} = \frac{1}{2} \sum_{k=1}^{d_i} \frac{\Gamma_\phi^k}{2\Omega_k} [\bar{\mathbf{N}}_b^k \phi_i + \bar{\mathbf{N}}_c^k \phi_k + \bar{\mathbf{N}}_a^k \phi_{k+1}] \cdot \bar{\mathbf{N}}_b^k. \quad (\text{B.16})$$

By expanding out the dot product in this last expression, and by making use of the periodicity of k , I_{Ω_i} can be recast as,

$$\begin{aligned} I_{\Omega_i} &= \frac{1}{2} \sum_{k=1}^{d_i} \frac{\Gamma_\phi^k}{2\Omega_k} \left[\left| \bar{\mathbf{N}}_b^k \right|^2 \phi_i + (\bar{\mathbf{N}}_c^k \cdot \bar{\mathbf{N}}_b^k) \phi_k + (\bar{\mathbf{N}}_a^k \cdot \bar{\mathbf{N}}_b^k) \phi_{k+1} \right], \\ &= \frac{1}{2} \sum_{k=1}^{d_i} \left[\Gamma_\phi^k \frac{\left| \bar{\mathbf{N}}_b^k \right|^2}{2\Omega_k} \phi_i + \left(\Gamma_\phi^k \frac{(\bar{\mathbf{N}}_c^k \cdot \bar{\mathbf{N}}_b^k)}{2\Omega_k} + \Gamma_\phi^{k-1} \frac{(\bar{\mathbf{N}}_a^{k-1} \cdot \bar{\mathbf{N}}_b^k)}{2\Omega_{k-1}} \right) \phi_k \right], \quad (\text{B.17}) \\ &= \alpha_i \phi_i - \sum_{k=1}^{d_i} \alpha_k \phi_k, \end{aligned}$$

where,

$$\begin{aligned} \alpha_i &= \frac{1}{2} \sum_{k=1}^{d_i} \Gamma_\phi^k \frac{\left| \bar{\mathbf{N}}_b^k \right|^2}{2\Omega_k}, \\ \alpha_k &= -\frac{1}{2} \left(\Gamma_\phi^k \frac{(\bar{\mathbf{N}}_c^k \cdot \bar{\mathbf{N}}_b^k)}{2\Omega_k} + \Gamma_\phi^{k-1} \frac{(\bar{\mathbf{N}}_a^{k-1} \cdot \bar{\mathbf{N}}_b^k)}{2\Omega_{k-1}} \right). \end{aligned} \quad (\text{B.18})$$

It is clear that α_c is always positive. Thus, in order to establish sufficient conditions for positive diffusion contributions to the coefficients in the discretized equations, the only remaining task is to find conditions such that $\alpha_k \geq 0$, or equivalently,

$$\Gamma_a^k \frac{-(\bar{\mathbf{N}}_a^k \cdot \bar{\mathbf{N}}_b^k)}{2\Omega_k} + \Gamma_b^{k-1} \frac{-(\bar{\mathbf{N}}_a^{k-1} \cdot \bar{\mathbf{N}}_b^{k-1})}{2\Omega_{k-1}} \geq 0. \quad (\text{B.19})$$

For convenience, the following tangent vectors are defined:

$$\bar{\mathbf{R}}_a^k = (x_k - x_i)\bar{\mathbf{i}} + (y_k - y_i)\bar{\mathbf{j}}, \quad (\text{B.20})$$

$$\bar{\mathbf{R}}_b^k = (x_{k+1} - x_k)\bar{\mathbf{i}} + (y_{k+1} - y_k)\bar{\mathbf{j}}, \quad (\text{B.21})$$

$$\bar{\mathbf{R}}_c^k = (x_i - x_{k+1})\bar{\mathbf{i}} + (y_i - y_{k+1})\bar{\mathbf{j}}. \quad (\text{B.22})$$

By referring to Fig. B.2, it is evident that these tangent vectors are related to the normal vectors, given by Eqs. (B.4)-(B.6), in the following respect:

$$\begin{aligned} \bar{\mathbf{N}}_a^k \times \bar{\mathbf{R}}_a^k &= (l_a^k)^2 \bar{\mathbf{k}}; & |\bar{\mathbf{N}}_a^k| &= |\bar{\mathbf{R}}_a^k| = l_a^k; \\ \bar{\mathbf{N}}_b^k \times \bar{\mathbf{R}}_b^k &= (l_b^k)^2 \bar{\mathbf{k}}; & |\bar{\mathbf{N}}_b^k| &= |\bar{\mathbf{R}}_b^k| = l_b^k; \\ \bar{\mathbf{N}}_c^k \times \bar{\mathbf{R}}_c^k &= (l_c^k)^2 \bar{\mathbf{k}}; & |\bar{\mathbf{N}}_c^k| &= |\bar{\mathbf{R}}_c^k| = l_c^k, \end{aligned} \quad (\text{B.23})$$

where $\bar{\mathbf{k}} = \bar{\mathbf{i}} \times \bar{\mathbf{j}}$, and l_a^k , l_b^k , and l_c^k represent the lengths of the corresponding sides of element k . With reference to Figs. B.1 and B.2, the following relationships also hold:

$$\begin{aligned} 2\Omega_k &= (x_i - x_{k+1})(y_k - y_{k+1}) - (x_k - x_{k+1})(y_i - y_{k+1}) \\ &= -(\bar{\mathbf{N}}_b^k \cdot \bar{\mathbf{R}}_c^k) \\ &= -l_b^k l_c^k \cos\left(\frac{\pi}{2} + \delta_k\right) \\ &= l_b^k l_c^k \sin(\delta_k), \end{aligned} \quad (\text{B.24})$$

$$\begin{aligned}
2\Omega_{k-1} &= (x_i - x_k)(y_{k-1} - y_k) - (x_{k-1} - x_k)(y_i - y_k) \\
&= -(\bar{\mathbf{N}}_a^{k-1} \cdot \bar{\mathbf{R}}_b^{k-1}) \\
&= -l_a^{k-1} l_b^{k-1} \cos\left(\frac{\pi}{2} + \gamma_{k-1}\right) \\
&= l_a^{k-1} l_b^{k-1} \sin(\gamma_{k-1}),
\end{aligned} \tag{B.25}$$

$$\begin{aligned}
\bar{\mathbf{N}}_b^k \cdot \bar{\mathbf{N}}_c^k &= \bar{\mathbf{R}}_b^k \cdot \bar{\mathbf{R}}_c^k \\
&= -l_b^k l_c^k \cos(\delta_k),
\end{aligned} \tag{B.26}$$

$$\begin{aligned}
\bar{\mathbf{N}}_a^{k-1} \cdot \bar{\mathbf{N}}_b^{k-1} &= \bar{\mathbf{R}}_a^{k-1} \cdot \bar{\mathbf{R}}_b^{k-1} \\
&= -l_a^{k-1} l_b^{k-1} \cos(\gamma_{k-1}).
\end{aligned} \tag{B.27}$$

Substituting Eqs. (B.24)-(B.27) into Eq. (B.19) yields,

$$\Gamma_\phi^k \frac{-(-l_b^k l_c^k \cos(\delta_k))}{l_b^k l_c^k \sin(\delta_k)} + \Gamma_\phi^{k-1} \frac{-(-l_a^{k-1} l_b^{k-1} \cos(\gamma_{k-1}))}{l_a^{k-1} l_b^{k-1} \sin(\gamma_{k-1})} \geq 0, \tag{B.28}$$

which, upon simplification, gives,

$$\Gamma_\phi^k \cot(\delta_k) + \Gamma_\phi^{k-1} \cot(\gamma_{k-1}) \geq 0. \tag{B.29}$$

Without placing any further restrictions on Γ_ϕ , the following sufficient condition guarantees that Eq. (B.29) is satisfied:

$$\begin{aligned}
\cot(\delta_k) &\geq 0 \quad \text{and} \quad \cot(\gamma_{k-1}) \geq 0 \\
\Rightarrow \delta_k &\leq \frac{\pi}{2} \quad \text{and} \quad \gamma_{k-1} \leq \frac{\pi}{2}.
\end{aligned} \tag{B.30}$$

Conclusion I: If Γ_ϕ is approximated as being piecewise-constant over each element, then, with respect to the conservation of ϕ , the sufficient condition required to guarantee positive *diffusion* contributions *from* control-volumes associated with interior nodes in the domain *to* the coefficients in the discretized equations is: all vertex angles in the domain (including vertex angles associated with nodes on the boundary) must be less than $\pi/2$ radians. That is, all elements in the domain must be acute-angle triangles.

If Γ_ϕ is constant over the entire domain, then the following sufficient condition guarantees that Eq. (B.29) is satisfied:

$$\begin{aligned} \cot(\delta_k) + \cot(\gamma_{k-1}) &\geq 0 \\ \Rightarrow \delta_k + \gamma_{k-1} &\leq \pi. \end{aligned} \quad (\text{B.31})$$

Equation (B.31) states that elements k and $(k-1)$ satisfy the empty circumcircle property (with respect to each other) associated with Delaunay triangulations [93]. It is noted here that Eq. (B.31) would also be satisfied if Γ_ϕ was taken as piecewise-constant over the control volume, Ω_i . However, this treatment of Γ_ϕ would lead to a non-conservative CVFEM formulation (if Γ_ϕ was not constant) and is, therefore, *not* recommended.

Conclusion II: If Γ_ϕ is constant over the entire domain, then the sufficient condition required to guarantee positive diffusion contributions from control-volumes associated with interior nodes in the domain to the coefficients in the discretized equations is: the mesh must be a Delaunay triangulation.

Boundary nodes. For a boundary node i , as shown in Fig. B.3, the index k is no longer periodic. In this case, the algebraic approximation to Eq. (B.13) can be written as,

$$\begin{aligned} I_{\Omega_i} &= \frac{1}{2} \sum_{k=1}^{d_i-1} \frac{\Gamma_\phi^k}{2\Omega_k} \left[\left| \bar{\mathbf{N}}_b^k \right|^2 \phi_i + (\bar{\mathbf{N}}_c^k \cdot \bar{\mathbf{N}}_b^k) \phi_k + (\bar{\mathbf{N}}_a^k \cdot \bar{\mathbf{N}}_b^k) \phi_{k+1} \right] \\ &= \frac{1}{2} \sum_{k=1}^{d_i-1} \Gamma_\phi^k \frac{\left| \bar{\mathbf{N}}_b^k \right|^2}{2\Omega_k} \phi_i + \frac{1}{2} \Gamma_\phi^1 \frac{(\bar{\mathbf{N}}_c^1 \cdot \bar{\mathbf{N}}_b^1)}{2\Omega_1} \phi_1 \\ &\quad + \frac{1}{2} \sum_{k=2}^{d_i-1} \left(\Gamma_\phi^k \frac{(\bar{\mathbf{N}}_c^k \cdot \bar{\mathbf{N}}_b^k)}{2\Omega_k} + \Gamma_\phi^{k-1} \frac{(\bar{\mathbf{N}}_a^{k-1} \cdot \bar{\mathbf{N}}_b^k)}{2\Omega_{k-1}} \right) \phi_k + \frac{1}{2} \Gamma_\phi^{d_i-1} \frac{(\bar{\mathbf{N}}_a^{d_i-1} \cdot \bar{\mathbf{N}}_b^{d_i-1})}{2\Omega_{d_i-1}} \phi_{d_i} \\ &= \alpha_1 \phi_i - \alpha_1 \phi_1 - \sum_{k=2}^{d_i-1} \alpha_k \phi_k - \alpha_{d_i} \phi_{d_i}, \end{aligned} \quad (\text{B.32})$$

where,

$$\begin{aligned}\alpha_i &= \frac{1}{2} \sum_{k=1}^{d_i-1} \Gamma_s^k \frac{|\bar{\mathbf{N}}_b^k|^2}{2\Omega_k}, \\ \alpha_1 &= -\frac{1}{2} \Gamma_s^1 \frac{(\bar{\mathbf{N}}_c^1 \cdot \bar{\mathbf{N}}_b^1)}{2\Omega_1}, \\ \alpha_k &= -\frac{1}{2} \left[\Gamma_s^k \frac{(\bar{\mathbf{N}}_c^k \cdot \bar{\mathbf{N}}_b^k)}{2\Omega_k} + \Gamma_s^{k-1} \frac{(\bar{\mathbf{N}}_c^{k-1} \cdot \bar{\mathbf{N}}_b^{k-1})}{2\Omega_{k-1}} \right], \quad \text{for } k = [2, 3, \dots, (d_i - 1)], \\ \alpha_{d_i} &= -\frac{1}{2} \Gamma_s^{d_i-1} \frac{(\bar{\mathbf{N}}_c^{d_i-1} \cdot \bar{\mathbf{N}}_b^{d_i-1})}{2\Omega_{d_i-1}}.\end{aligned}\tag{B.33}$$

It is evident that α_i is always positive, and the coefficients α_k , are identical to those given by Eq. (B.18). Thus, assuming Eq. (B.29) holds in the present case for $k = [2, 3, \dots, (d_i - 1)]$, then in order to establish sufficient conditions for positive diffusion contributions from the control-volumes associated with the boundary nodes to the coefficients in the discretized equations, the task is to find conditions such that $\alpha_{d_i} + \alpha_1 \geq 0$, or equivalently,

$$\Gamma_s^{d_i-1} \frac{-(\bar{\mathbf{N}}_c^{d_i-1} \cdot \bar{\mathbf{N}}_b^{d_i-1})}{2\Omega_{d_i-1}} + \Gamma_s^1 \frac{-(\bar{\mathbf{N}}_c^1 \cdot \bar{\mathbf{N}}_b^1)}{2\Omega_1} \geq 0.\tag{B.34}$$

Using Eqs. (B.24) and (B.26) with $k = d_i - 1$, and Eqs. (B.25) and (B.27) with $k = 2$, Eq. (B.34) can be recast as,

$$\Gamma_s^{d_i-1} \cot(\delta_{d_i-1}) + \Gamma_s^1 \cot(\gamma_1) \geq 0.\tag{B.35}$$

If Γ_s is taken as piecewise-constant over each element, then the following condition guarantees that Eq. (B.35) is satisfied:

$$\begin{aligned}\cot(\delta_{d_i-1}) &\geq 0 \quad \text{and} \quad \cot(\gamma_1) \geq 0 \\ \Rightarrow \quad \delta_{d_i-1} &\leq \frac{\pi}{2} \quad \text{and} \quad \gamma_1 \leq \frac{\pi}{2}.\end{aligned}\tag{B.36}$$

Conclusion III: Based on conclusion I and Eq. (B.36), if Γ_s is approximated as being piecewise-constant over each element, then the sufficient condition required to guarantee

positive diffusion contributions to the coefficients in the discretized equations is: all vertex angles in the domain must be less than $\pi/2$ radians. That is, all elements in the domain must be acute-angle triangles.

If Γ_s is *constant over the domain*, then the following sufficient condition guarantees that Eq. (B.35) is satisfied:

$$\begin{aligned} \cot(\delta_{d,-1}) + \cot(\gamma_1) &\geq 0 \\ \Rightarrow \delta_{d,-1} + \gamma_1 &\leq \pi. \end{aligned} \quad (\text{B.37})$$

Conclusion IV: Based on conclusion II and Eq. (B.37), if Γ_s is *constant over the entire domain*, then the sufficient conditions required to guarantee positive diffusion contributions to the coefficients in the discretized equations are: (i) the mesh must be a Delaunay triangulation; and (ii) Eq. (B.37) must hold for each node, i , on the boundary.

It may not be practical (or possible) to guarantee that the condition given by Eq. (B.37) is satisfied in general. Thus, the following set of stricter conditions may be adopted for the case of constant Γ_s : (i) the mesh must be a Delaunay triangulation; and (ii) all elements that have at least one edge on the boundary must be acute-angle triangles.

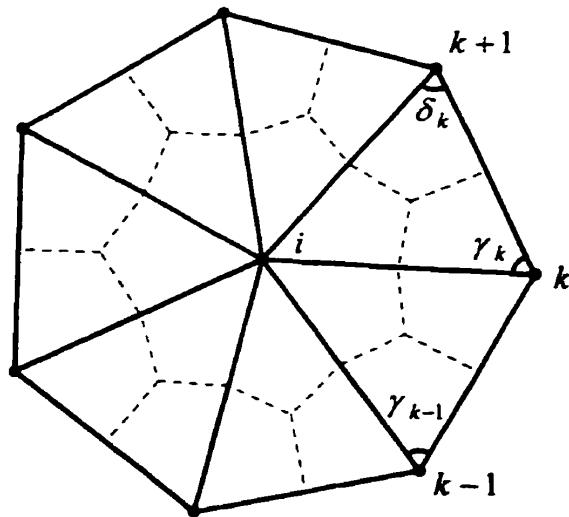


Figure B.1: The patch of elements surrounding a typical interior node, i , and related nomenclature. The dashed lines represent control-volume boundaries.

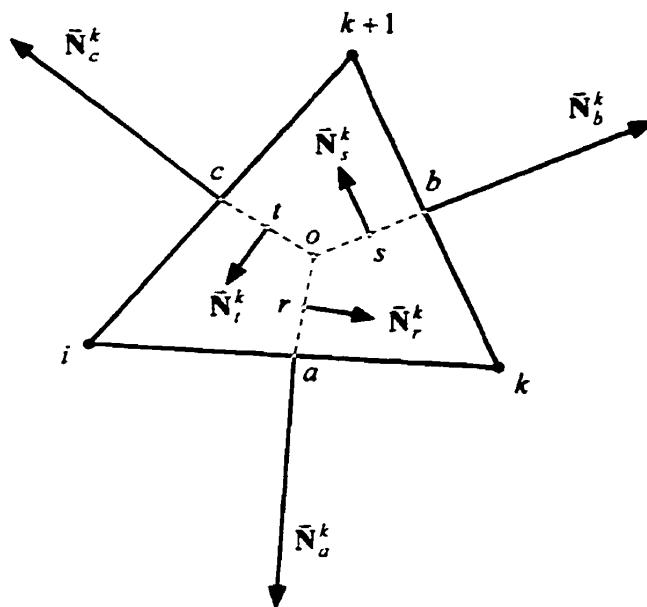


Figure B.2: The k^{th} element surrounding node i , and related nomenclature.

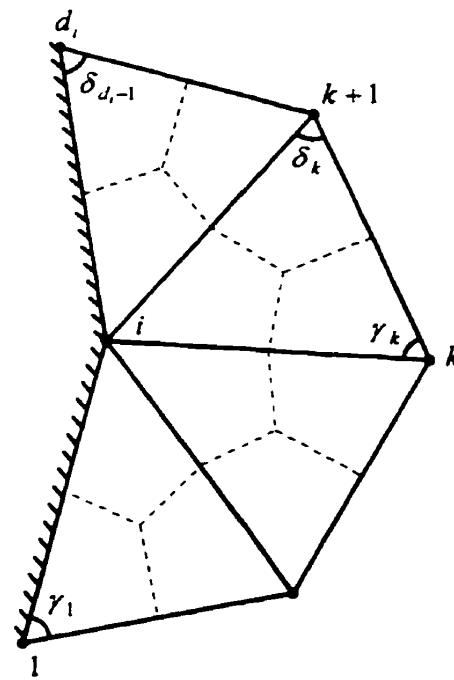


Figure B.3: The patch of elements surrounding a typical boundary node, i , and related nomenclature. The dashed lines represent control-volume boundaries.