

Algebraic Multigrid for Stabilized Finite Element Discretizations of the Navier Stokes Equations

by

Tolulope Olawale Okusanya

B.Sc (Mechanical Eng.), Rutgers University (1994)
S.M. (Aeronautics and Astronautics), M.I.T. (1996)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2002

© Massachusetts Institute of Technology 2002. All rights reserved.

Submitted by
Department of Aeronautics and Astronautics
May, 2002

Certified by David Darmofal
Associate Professor of Aeronautics and Astronautics

Certified by Jaume Peraire
Professor of Aeronautics and Astronautics

Certified by Mark Drela
Professor of Aeronautics and Astronautics

Certified by Carlos Cesnik
Associate Professor of Aerospace Engineering, University of Michigan

Accepted by Wallace E. Vander Velde
Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

Algebraic Multigrid for Stabilized Finite Element Discretizations of the Navier Stokes Equations

by

Tolulope Olawale Okusanya

Submitted to the Department of Aeronautics and Astronautics
on May, 2002, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Aeronautics and Astronautics

Abstract

A multilevel method for the solution of systems of equations generated by stabilized Finite Element discretizations of the Euler and Navier Stokes equations on generalized unstructured grids is described. The method is based on an elemental agglomeration multigrid which produces a hierarchical sequence of coarse subspaces. Linear combinations of the basis functions from a given space form the next subspace and the use of the Galerkin Coarse Grid Approximation (GCA) within an Algebraic Multigrid (AMG) context properly defines the hierarchical sequence. The multigrid coarse spaces constructed by the elemental agglomeration algorithm are based on a semi-coarsening scheme designed to reduce grid anisotropy. The multigrid transfer operators are induced by the graph of the coarse space mesh and proper consideration is given to the boundary conditions for an accurate representation of the coarse space operators. A generalized line implicit relaxation scheme is also described where the lines are constructed to follow the direction of strongest coupling. The solution algorithm is motivated by the decomposition of the system characteristics into acoustic and convective modes. Analysis of the application of elemental agglomeration AMG (AMGe) to stabilized numerical schemes shows that a characteristic length based rescaling of the numerical stabilization is necessary for a consistent multigrid representation.

Thesis Supervisor: David Darmofal

Title: Associate Professor of Aeronautics and Astronautics

Thesis Supervisor: Jaume Peraire

Title: Professor of Aeronautics and Astronautics

Acknowledgments

So many people have contributed to this thesis, whether directly or indirectly, that it's difficult to know where to start. I am just grateful to have met such a diverse and great group of people to have interacted with. Recognizing each and every person who has contributed would probably take far longer than it took for me to complete this thesis, so I apologize for this in advance.

On the financial front, this thesis would not have been possible without (a lot of) help especially having spent this amount of time at an expensive school in an expensive city. I am thus grateful for the financial support provided by the NASA Dryden Flight Research Center under research contract NAG4-157 and the Singapore-MIT Alliance.

I would like to thank Prof. David Darmofal for having provided the impetus for this thesis and for his sound research advice and ideas especially when it felt like things were spiraling into a black hole. I feel especially lucky to have had him as an advisor and for all his hard work in supporting this work. I would also like to especially thank Prof. Jaume Peraire for serving as my advisor and supporting me these past years. I am grateful to him for providing me with invaluable insight as well as allowing me the freedom to travel and pursue other activities. Thanks also go to Prof. Mark Drela for always having the simplest and most direct answer for everything and to my minor advisor Prof. Carlos Cesnik. I would also like to thank Dr. Tom Roberts and Dr. Dmitri Mavriplis for shouldering the responsibility of reading this thesis. A special thanks goes to Dr. Dmitri Mavriplis who went beyond the call of duty for his help in obtaining comparison data.

I would like to thank the cast of those who have gone before (and may possibly still be here in some form). Thanks goes to Bob Haimes for an honest perspective on everything and for reading my thesis. I really have no better way to describe Bob except that he is one of the more stable things around here. Here's to beer, good coffee, Jennifer Lopez and the other things that count in life. I would also like to especially thank Ed and Angie for friendship over the years and for some of the more colorful periods I've had. We've been through so much together that I feel we are a triumvirate of sorts. By the way, those incidents with users copying my dot files are completely not my fault. Forgive them, for they knew not what they did. To the crazy set of people I first met here at MIT - Graeme, Ray, Jim, Guy and Carmen (remember guys, that was no ORDINARY chicken). To other former CASL (that's how long it's been) members - Karen for introducing me to the quirkiest sides of New Zealand such as Vegemite (which I still hate by the way) and for company as an officemate; Ali for always being there to bounce ideas off as well as all the Kung-fu jokes; Imran for always weathering the Pakistan jokes with good grace as well as the most fantastic C++

naming scheme that this world has been privileged to behold. I would also like to thank Tatsuo for some of the more interesting discussions we've had about Japan as well as being my dealer for the good stuff.

A big thanks goes to Jean Sofronas for all her help in organization and quietly working behind the scenes. As much as advisors may hold your fate in their hands, she holds their fate in her hands. To the previous FDRL (again) and current set of ACDL (yet again!) members, I'd like to thank for all the wonderful times - Alex for complementing my (lack of) abstract mathematics and discussions on education and life; Larry and Adam for showing that it's only possible to have fun when you are a Master's student; Joe and Dave for the times up in Montreal (I can't wait for a second helping of those potatoes, Joe); Victor for some of the wackier stories about Ecuador and Tae-kwon-do; Joseph for haranguing the unenlightened masses about the intricacies of Buffy, guns and working out; and Sudheep for taking me back on memory lane regarding the British system.

Outside the lab, I have been fortunate to have met such a diverse set of people whom I'm proud to call friends. First of all, I would like to acknowledge Chris Lee whom I've known since 1995 and has been one of the few constants that I could always count upon. Here's to all the good times we've had and the never-ending search for the ideal woman. I'd also like to acknowledge two of the most interesting housemates that I've ever had, namely Dean Chen and Kevin Silva. Kevin is quite possibly one of the funniest humans on this earth while Dean and I go way back to our days in Japan together. Who could ever forget that night in Shibuya while waiting at the Hachiko and eating Baby Castellas.

I would also like to thank the MIT gamers who have come and gone especially Yue Zheng (for all the funny jokes), Julien Beasley (I'll make sure to let your wife know about your cross-dressing days), Shinpei Kuga (for the Japan perspective), Jef Perlman (there are still holes in the wall), Matt Tom (only you would make **akusokuzan** a way of life), Koichi Kunitake and Chi Nguyen (I've got your uppercut right here).

A big thanks goes to Johnson Chung and the MIT SKA Karate club for giving me a wonderful opportunity to focus my energies while learning something useful at the same time. There's nothing quite like 90 minutes of having your Gluteus Maximus handed back to you on a silver platter to make you realize that graduate school is not so bad. I guess it's all about putting things in perspective. I'd also like to thank all the friends I met through the club over the years.

I would like to acknowledge the the MIT Japan Program and the MIT Japan Lunch Table for the wonderful experiences that I've had over the years. Pat Gercik and Mitsuko Barker are two of the most wonderful people I have met. I am honored to have had their

friendship and help extended to me. I would also like to thank some very special people that I have met through them; Dai Matsuoka for being my very first language partner and all the great conversation we've had, Yoshie and Soichi Saeki for being such great friends and last but not least, Mieko Kuroda for being one of the coolest people I know.

A very special thanks goes to Alain Hoang for the friendship we've had over the years and some of the interesting talks we've had about life, Capoeira and Wing Chung Ken; Alex Wong for things I cannot even begin to describe in words (yes, you are that "special"); and Sergei Aish for the wedding invite in Singapore, multimedia knowledge and friendship over the years. I consider these guys to be a special triad in my circle of friends.

I owe a special thanks to Andréa (Andy) Froncioni for starting me down the path of CFD (some would claim it's the path to the Dark Side). It is amazing how one Fortran homework can change the course of one's life. I am grateful for his initial guidance and patience and to all the victories and defeats we experienced together at Rutgers. I would also like to thank Prof. Richard Peskin, Prof. Richard Pelz and Prof. Mitsunori Denda of Rutgers University for their help and support in both academia and in general.

Finally, I would like to thank my family who have provided so much love, support and encouragement in all things. In particular I would like to thank my mother for always being the Rock of Gibraltar that she is, her sound advice on life in general and for instilling the love of learning in me; and my father for the wonderful support and academic advice. To Tayo, Ibukun and Gbenga for camaraderie and brotherhood. You guys are the greatest.

奥義

を極める

“To master the secret principles (of).”

Contents

Nomenclature	15
1 Introduction	19
1.1 Multigrid and Preconditioned Krylov Methods	21
1.2 Multigrid	23
1.3 AMG Approach for Navier-Stokes	29
1.4 Summary	32
2 AMG: Symmetric Elliptic Operators	35
2.1 Model Problem	36
2.2 Multigrid: Operation and Components	39
2.2.1 Multilevel Algorithm	40
2.2.2 Multigrid Smoother	42
2.2.3 Interpolation and Coarse Grid Operators	44
2.3 Coarse Space Agglomeration	47
2.3.1 Coarse Space Topology	48
2.3.2 Elemental Agglomeration Algorithm	49
2.3.3 Coarse Space Basis Functions	51
2.4 Consistency Scaling Formulation	55
2.5 Dirichlet Boundary Modification	57
2.6 Anisotropic Problems	58
2.7 Results	59
3 AMG: Convection-Diffusion Operators	63
3.1 Model Problem	64
3.2 Multistaging	70
3.3 Multigrid Smoother for Convection-Diffusion Operators	72
3.4 Implicit Line Construction	74
3.5 Fourier Analysis of Implicit Line Smoother	77

3.6	Consistency Scaling Issues for Stabilized Methods	88
3.7	Results: Elliptic Operator (revisited)	92
3.8	Results: Convection-Diffusion Operator	92
4	Euler Applications	99
4.1	FEM Discretization	101
4.2	AMG Extension to the Euler Equations	105
4.2.1	Newton Scheme	106
4.2.2	Implicit Line Creation Extension	107
4.2.3	Interpolation Operator Extension	109
4.3	Consistency Scaling Issues for the Stabilized Euler Equations	111
4.4	Results	111
4.4.1	Channel Flow	113
4.4.2	Airfoil Flow	114
5	Navier Stokes Applications	119
5.1	FEM Discretization	120
5.2	Limitations of Current FEM Implementation	124
5.3	AMG Extension to the Navier-Stokes Equations	125
5.3.1	Implicit Line Creation Extension	125
5.3.2	Dirichlet Boundary Condition Extension	125
5.4	Results	126
6	Conclusions	131
6.1	Summary	131
6.2	Future Work	133
A	Multigrid Preconditioning Matrix	135
B	Nodal Agglomeration	137
	References	141

List of Figures

1.1	Agglomeration Types	27
1.2	Diagnosis of multigrid breakdown for the Navier–Stokes equations and solutions.	29
2.1	Domain and Finite Element Discretization using Linear Triangles	36
2.2	Example Multigrid Cycles	41
2.3	Coarse Space Topology	48
2.4	Coarse space topology with exceptional macroelement bearing two coarse nodes and extra support node	49
2.5	Multilevel Elemental Agglomeration Example	51
2.6	Coarse Space Basis Function	52
2.7	Dirichlet Boundary Modification for $\mathbf{P}_k$	57
2.8	Semi-Coarsening	59
2.9	Agglomeration Multigrid Convergence History for Poisson Problem: Isotropic Mesh	60
2.10	Agglomeration Multigrid Convergence History for Poisson Problem: Anisotropic Mesh	61
3.1	Illustration of Linear Triangle Length Scale h_e	67
3.2	Element Mapping	69
3.3	Stability Contours	72
3.4	Example Implicit Lines	74
3.5	Scalar convection-diffusion model domain	78
3.6	Computational 9 point stencil for Fourier Analysis of Implicit Line Smoother	79
3.7	Determination of characteristic length h_e	81
3.8	Smooth (Θ_s) and rough (Θ_r , grayed) wavenumber sets in two dimension . .	85
3.9	Fourier footprints for implicit line Gauß-Seidel smoother using full coarsening ($\delta=1$)	87
3.10	Fourier footprints for implicit line Gauß-Seidel smoother using semi-coarsening ($\delta=10^{-2}$)	87

3.11	Comparison of scalar convection-diffusion equation eigenspectrum w/ and w/o τ scaling for $Pe = 1e6$	91
3.12	Computational domain for scalar convection diffusion boundary layer problem	93
3.13	Boundary Layer Problem Grid Level Dependency	94
3.14	Boundary Layer Problem Mesh Size Dependency	94
3.15	Multigrid Results for Scalar Convection-Diffusion: Peclet number dependency	95
4.1	Implicit lines construction using the reconstruction and rediscrretization scheme for inviscid flow over a NACA 0012 at Mach 0.1	110
4.2	τ scaling effect on line Jacobi iteration matrix eigenspectrum for channel problem at freestream Mach number $M = 0.1$	112
4.3	Channel Flow Geometry for Euler Equations	113
4.4	Mach Number Contours for $M = 0.1$ Compressible Euler 241×81 Bump Problem	113
4.5	Unstructured Mesh for NACA 0012 Foil	115
4.6	Mach Number contours for freestream $M = 0.1$ compressible Euler flow over a NACA 0012 airfoil at zero angle of attack	116
4.7	Non-linear Newton outer loop and linear multigrid convergence histories for compressible Euler flow over a NACA 0012 airfoil with 20,621 fine grid nodes at Mach 0.5 and zero angle of attack using a 5-stage scheme	116
4.8	Multigrid results by Pierce <i>et al</i> [1] for inviscid flow over NACA 0012 airfoil at freestream Mach number $M = 0.5$, $\alpha = 3^\circ$ on a 160×32 O-mesh using scalar, diagonal and block-Jacobi pre-conditioning	117
5.1	Unstructured Mesh for NACA 0005 Foil	127
5.2	Non-linear Newton outer loop and linear multigrid convergence histories for compressible Navier-Stokes flow over a NACA 0005 airfoil with 36,388 fine grid nodes at Mach 0.5, zero angle of attack and Reynolds number of 5000 using a 5-stage scheme	128
5.3	Multigrid results by Pierce <i>et al</i> [1] for viscous flow over NACA 0012 airfoil at freestream Mach number $M = 0.5$, $\alpha = 0^\circ$ and Reynolds number of 5000 on a 320×64 O-mesh using scalar, diagonal and block-Jacobi pre-conditioning	129
B.1	Injection	137

Nomenclature

Roman Symbols

<u>Symbol</u>	<u>Description</u>	<u>Section</u>
$\mathbf{A}_1, \mathbf{A}_2$	Inviscid flux Jacobians $\frac{\partial \mathbf{F}_1}{\partial \mathbf{U}}, \frac{\partial \mathbf{F}_2}{\partial \mathbf{U}}$	4.1
$\mathbf{A}_h$	Fine grid matrix	2.1
$\mathbf{A}_H, \mathbf{A}_k$	Coarse space representation of fine grid matrix $\mathbf{A}_h$	2.2.1
$\ \mathbf{A}\ $	Contraction number of matrix $\mathbf{A}$	2.2.2
$c_{i,j}$	Line construction weighting matrix coefficient	3.4
c	Speed of sound	
e	Internal energy	4.1
E	Specific total energy	4.1
f	Source function	2.1
$\mathbf{F}_1, \mathbf{F}_2$	Inviscid flux of conserved variables $\mathbf{U}$	4.1
$\mathbf{F}_1^v, \mathbf{F}_2^v$	Viscous flux of conserved variables $\mathbf{U}$	5.1
g_D	Specified data on Dirichlet boundary	2.1
g_N	Specified data on Neumann boundary	2.1
h, h_e	Characteristic element size	2.1
$H^1(\Omega)$	Sobolev space of functions with square integrable first derivatives	2.1
$H_0^1(\Omega; \Gamma_D)$	Subset space of $H^1(\Omega; \Gamma)$ which vanish on Γ_D	2.1

$H^{1h}(\Omega_h)$	Space spanned by piecewise continuous polynomial basis functions	2.1
$\mathbf{K}_{ij}$	Viscous flux Jacobians $\mathbf{F}_{i,\mathbf{U},j}^v$	5.1
$\mathbf{M}_k$	Preconditioning matrix	2.2.2
m_P	Order of polynomial interpolated exactly by $\mathbf{P}$	2.2.3
m_R	Order of polynomial interpolated exactly by $\mathbf{R}$	2.2.3
$2m$	Order of governing PDE	2.2.3
M	Mach number	4.1
$\mathbf{n}$	Outward unit normal to boundary Γ	2.1
n_e	Number of elements in triangulation	2.1
n_p	Number of vertices in triangulation	2.1
N_i, N_j	Nodal basis functions associated with nodes i and j	2.1
p	Pressure	4.1
Pe, Pe_h	Peclet number	3.1
Pr	Prandtl number	5.1
$\mathbf{P}_k, \mathbf{P}_h$	Prolongation operator	2.2
$\mathbf{R}_k, \mathbf{R}_h$	Restriction operator	2.2
Re	Reynolds Number	5.1
$\mathcal{S}^h$	Finite Element trial function space	2.1
$\mathcal{S}(\cdot, \cdot, \cdot, \cdot)$	Multigrid smoother	2.2
$\mathbf{S}_k$	Iteration matrix	2.2.2
TEC	Total Element Complexity	4.4.2
TVC	Total Vertex Complexity	4.4.2
$\mathbf{u}$	Velocity vector	4.1

$\mathbf{U}$	Vector of conserved variables $(\rho, \rho u_1, \rho u_2, \rho E)^T$	4.1
$\mathbf{V}$	Vector of entropy variables	4.1
$\mathcal{V}^h$	Finite Element test or weight function space	2.1
$\vec{\mathbf{V}}$	Convection velocity field	3.1
$\mathbf{x}, \mathbf{y}, x_i$	Cartesian coordinates	2.1

Greek Symbols

<u>Symbol</u>	<u>Description</u>	<u>Section</u>
α_i	Multistage coefficients	3.2
δ_{ij}	Kronecker delta function	2.1
δ	Mesh aspect ratio	3.5
δ_s	Initial anisotropic mesh aspect ratio	2.1
ϵ	Asymptotic multigrid convergence rate	2.7
$\bar{\epsilon}$	Average multigrid convergence rate	2.7
γ	Ratio of specific heats	4.1
γ_k	Multigrid cycle index	2.2.1
Γ	Governing PDE domain boundary	2.1
Γ_D	Dirichlet boundary of domain Ω	2.1
Γ^e	Triangle element boundary	2.1
Γ_{ff}	Farfield boundary of domain Ω	4.1

Γ_N	Neumann boundary of domain Ω	2.1
Γ_w	Impermeable solid wall boundary of domain Ω	4.1
Γ^+	Inflow boundary of domain Ω	3.1
Γ^-	Outflow boundary of domain Ω	3.1
ξ, η	Parametric space coordinates	3.1
μ	Viscosity (diffusivity) coefficient	2.1
ν_1	Number of pre-smoothing sweeps	2.2.1
ν_2	Number of post-smoothing sweeps	2.2.1
ω	Iterative scheme relaxation factor	2.2.2
Ω	Governing PDE domain	2.1
Ω_h, Ω_H	Computational domain	2.1
ρ	Density	4.1
$\rho(\mathbf{A})$	Spectral radius of matrix $\mathbf{A}$	2.2.2
σ	Diagonal scaling matrix for restriction operator $\mathbf{R}$	2.4
τ, τ_e	Stabilization parameter	3.1
τ	Stabilization matrix	4.1
τ_i	Inviscid stabilization matrix	5.1
τ_v	Viscous stabilization matrix contribution	5.1
θ_x, θ_y	Fourier angles	3.5

Chapter 1

Introduction

Rapid advances in unstructured mesh methods for computational fluid dynamics (CFD) have been made in recent years and, for the computation of inviscid flows [2–4], have achieved a considerable level of maturity. Viscous flow technology is also rapidly developing and the use of unstructured grids has been started [5–7]. Unstructured meshes offer a practical means for computation and have the advantage of providing both flexible approximations of the domain geometry and easy adaptation/refinement of the mesh.

Accurate and efficient solutions to the compressible Navier-Stokes equations, especially in the turbulent high Reynolds number limit, remains a challenging problem due in part to the myriad of associated length scales required to properly resolve flow features. This is especially true in the boundary layer regions which are characterized by strong gradients in the normal direction and relatively weak streamwise variations. In order to accurately resolve the boundary layer in a computationally efficient manner, grid anisotropy is employed. This introduces two problems that lead to the severe deterioration of many existing numerical algorithms. The first is the increased stiffness of the discrete problem and the second is the increase in the number of required mesh points which strains existing computational resources for problems of practical interest. Hence, the efficiency of current solution methods remains a critical problem.

A brief review of the state-of-the-art in viscous flow technology is made by considering two solution algorithms for practical aerodynamic applications. The first is by Pierce *et al* [8] which is a structured grid solver that employs a conservative cell-centered semi-discrete Finite Volume discretization with a characteristic based matrix artificial dissipation. Turbu-

lence is accounted for by the implementation of the Baldwin-Lomax and Spalart-Allmaras one-equation turbulence models. The solution scheme is a J-coarsened non-linear multigrid scheme with a point implicit block-Jacobi preconditioner. For the range of problems tested, the convergence rate is roughly of the order of 0.94. The second flow solver considered is by Mavriplis [9] which is an unstructured grid solver that employs a conservative vertex-centered Finite Volume discretization with a matrix-based artificial dissipation and the Spalart-Allmaras one-equation turbulence model. The solution scheme is a semi-coarsening non-linear multigrid scheme with a hybrid point/line implicit block-Jacobi preconditioner and for the range of problems tested, the convergence rates ranged from 0.78 to 0.965. These convergence rates are a far cry from the ideal multigrid convergence rate of 0.1 which has been theoretically proven and demonstrated for elliptic symmetric operators.

While these schemes represent an improvement over standard multigrid implementations that typically achieve convergence rates of 0.99, it is still not fully clear why optimal rates are not achieved. However, one conjecture that can be made is due to the common point between these schemes which is that they are *non-compact* schemes i.e they have extended stencils. The computational costs of evaluating and/or storing the flux Jacobians or non-linear residuals for a non-compact formulation typically results in these schemes employing a reduced order approximation that may lead to a convergence slowdown. Mavriplis [9] has conducted a careful study of the use of lower order approximations in multigrid solutions for Finite-Volume discretizations of the Euler and Navier-Stokes equations. He concludes that any improvement in the multigrid components will have little effect and the only way to achieve better convergence rates is through better full Jacobian approximations. In contrast, the Finite Element Method (FEM) offers better alternatives. FEM formulations offer advantages which are crucial to the development of an efficient solution scheme. Some of these advantages include

- **Compact scheme:** FEM is a compact scheme which results in a nearest neighbor stencil. This enables an exact derivative for the Jacobian matrix to be taken without any approximations.
- **Higher order formulations:** FEM formulations allow for an easy extension to higher order formulations.
- **Variational structure:** FEM features a rich variational structure for mathematical analysis such as error estimation.

- **Grid distortion:** FEM formulations also allow for more accurate interpolation within the computational cells in the presence of strong mesh anisotropy.

However, a current problem with FEM discretizations is that existing solution methods are typically slower and more memory intensive than for Finite Volume discretizations. However, the properties of the FEM method, in particular the compactness for higher order discretizations, we believe are critical to the future development of an accurate and efficient method for solution of the Navier-Stokes equations. In this thesis, we will present an Algebraic Multigrid method for solving stabilized FEM discretizations of the Navier-Stokes equations. Specifically, the contributions of this thesis include:

1. Development of a fast solution method for Euler and Navier-Stokes equations for non-trivial flows.
2. Implementation of Algebraic Multigrid within a stabilized Finite Element context.
3. Construction of improved multigrid components for convection-dominated flows.

1.1 Multigrid and Preconditioned Krylov Methods

Discretization of the governing partial differential equations on the mesh gives rise to large non-linear systems of equations such that for 3D problems, the solution of these large discrete problems is rendered intractable for direct solution methods. As a result, iterative solution methods based on Krylov subspace methods [10–12] and/or multilevel methods [9, 13], which include multigrid and domain decomposition methods, are attractive.

Subspace methods can be very efficient methods but suffer from a dependence of the convergence on the scaling of the eigenvalues with mesh size. This shortcoming may be ameliorated by the use of a suitable preconditioner [14, 15] and have been shown to be effective for inviscid calculations of the Euler equations. On the other hand, for elliptic operators, multigrid methods can provide mesh independent convergence rates [16–22] and offer good scaling of the compute time as well as data storage requirements. Even though there has been no generalized extension of the mesh independent convergence proof for systems with hyperbolic components, multigrid methods have been effectively applied to the Euler equations [23, 24] and have remained a popular approach for the Navier-Stokes equations [13, 25, 26]. Reusken [27] shows a grid independent convergence proof for a Finite

Difference discretization of a 2D model convection-diffusion problem under some simplified conditions using linear algebra arguments. For some simple cases, Roberts *et al* demonstrate Textbook Multigrid Efficiency (TME) in the computation of the steady incompressible Euler equations [28] and Thomas *et al* have also demonstrated TME for the steady compressible Navier-Stokes equations [29]. TME here, is defined by Brandt [30] as *solving a discrete PDE problem in a computational work which is only a small (less than 10) multiple of the operation count in the discretized system of equations itself*

In general, problems for which subspace methods perform well such as elliptic problems, are also problems on which standard multigrid algorithms perform well. For tougher problems such as the Navier-Stokes equations, both methods suffer a significant increase in the work required for convergence. However, multigrid remains a more attractive option due to the scaling of the computational resources required. For example, in a viscous computation using an implicit discretization, the size of the system matrix which is typically sparse, is a constant multiplied by N , where N is the number of unknowns.

For comparison, a much favored Krylov subspace method for this type of computation is GMRES, with an Incomplete LU (ILU) preconditioner [7, 31–33]. Any Krylov subspace method needs to store a set of search space vectors which is typically of the order of 40 to 50 while the cost of the ILU factorization is of $\mathcal{O}(N)$. The storage requirements for multigrid methods however are bounded by the complexity of the coarsening procedure such that even in the worst case scenario of a 2:1 coarsening ratio on the multigrid coarse spaces, the total storage for all the coarse space matrices is less than that for the fine mesh matrix. Let $\bar{r} \leq 1$ be the ratio of the total multigrid coarse space matrix storage to the fine mesh matrix storage. Also, let KN be the storage requirements for the Krylov search space vectors. A direct comparison of the storage cost then gives

$$\text{MULTIGRID storage cost} = \begin{cases} CN & \text{Fine grid sparse matrix} \\ + \bar{r}CN & \text{Coarse grid matrices} \\ + N & \text{Fine grid solution array} \\ + \bar{r}N & \text{Coarse grid solution arrays} \end{cases}$$

$$\text{GMRES/ILU cost} = \left\{ \begin{array}{ll} CN & \text{Sparse matrix} \\ + N & \text{Solution array} \\ + CN & \text{ILU} \\ + KN & \text{Krylov vectors} \end{array} \right.$$

$$\begin{aligned} \frac{\text{MULTIGRID}}{\text{GMRES/ILU}} &= \frac{(C+1)(1+\bar{r})}{(2C+K+1)} \\ &\leq \frac{2(C+1)}{2(C+1)+(K-1)} \\ &< 1 \end{aligned}$$

Hence, a direct comparison of the computational resources shows that a multilevel approach is more attractive provided that the convergence rates are compared on an equal basis. Krylov subspace methods may also be implemented in matrix free form [34] but the more efficient preconditioners such as SSOR and ILU can no longer be used. Multilevel methods, however, may be used as preconditioners for Krylov subspace iterative solvers which provides a powerful and flexible framework for computation. Another added benefit of this is the possibility of reducing the number of Krylov search space vectors.

1.2 Multigrid

The multigrid method essentially considers a decomposition of the solution error into rough error components, which cannot be resolved on a coarser grid without aliasing, and the complementary smooth error component which can be resolved on the coarser grid. Given a hierarchical sequence of successively coarser grids, a recursive partitioning of the solution error may be made amongst these grids such that the associated error components on each coarse grid effectively form a basis for the smooth error component on the finest grid. The partitioning of the error is achieved through a set of interpolation operators for the transfer of error components between the spaces. The crux of multigrid methods is the elimination of the rough error modes by means of a relaxation scheme on the current mesh. Hence, the efficiency of the multigrid performance is dependent on the synergy of the relaxation scheme with the coarsening algorithms.

Structured meshes have the advantage that the coarse spaces may be naturally defined

using the fine mesh. However, unstructured meshes do not have a natural grid hierarchy and as such introduce additional complexities such as proper identification of the coarse grid problems. Some of the geometric methods for the construction of coarse spaces for unstructured meshes include the a-priori generation of independent coarse grids which are overset [35] and the subsequent construction of piecewise interpolants between the grids. This could be advantageous since the same grid generator can be used for all the grids. However, since no relationship exists between the fine grid nodes and the coarse grid nodes, the work involved in computing the interpolants will be $O(N^2)$ [18]. This can be reduced to $O(N)$ using fast graph traversal algorithms [36]. In contrast to this, one may consider the first mesh as the coarsest mesh and simply refine the elements in order to obtain a finer mesh. This leads to a set of coarse spaces which are nested and for which interpolation operators may be easily defined. One serious drawback of this method is the dependence of the fine grid distribution on the coarse levels. Another approach is based on a nodal decimation technique which involves selection of a vertex subset and retriangulation. The selection process is typically based on the fine grid geometry and depends on some pattern in the fine grid [37, 38].

For numerical discretizations of the Euler and Navier-Stokes equations, one fundamental cause of degradation in standard multigrid algorithms is the decoupling of error modes in one or more coordinate directions. Navier-Stokes computations using standard Geometric Multigrid formulations suffer an appreciable degradation in performance [39] with convergence rates of the order of 0.99. One means to combat this is the use of directional coarsening algorithms in the construction of the coarse grids. Calculations in the inviscid regions of the mesh use a full coarsening technique which gives a 4:1 element count reduction in 2D. However, depending on the chosen relaxation scheme, alleviation of the stiffness due to stretched grids in viscous flow calculations requires semi-coarsening techniques [6] which typically gives a 2:1 element count reduction in 2D within the boundary layer. These methods are typically used within a Geometric Multigrid (GMG) context where the coarse grid equations are based on a rediscretization of the governing PDE on these coarse spaces. Extensive analysis of Geometric Multigrid as applied to elliptic problems has been done by Brandt [16, 17, 40] from which theoretical properties such as mesh independent convergence have been shown. These methods have also been extended to computational flow calculations for transonic potential flow [41, 42] and are now routine in inviscid Euler computations [24, 35, 43]. These applications of Geometric Multigrid to the Euler applications

have met with varying success and in general, the best achievable convergence rates have been of the order of 0.7.

In contrast to rediscrctizing the PDE as in GMG, Algebraic Multigrid (AMG) uses an algebraic definition for the coarse space operators [44]. Under AMG, the coarse spaces may be generated in a purely algebraic fashion or with the help of geometric constructs. In the classic definition of AMG [44], the construction is algebraic which allows for automatic construction of the coarse spaces and does not require geometric information. Classic AMG makes use of an abstraction of traditional multigrid principles in an algebraic context leading to a redefinition of such geometric concepts as grids. Let us consider the linear set of n algebraic equations

$$\mathbf{A}u = b, \quad \{\mathbf{A} \in \mathbb{R}^{n \times n}; u, b \in \mathbb{R}^n\}$$

where $\mathbf{A} = \{a_{ij}\}_{n \times n}$, $u = (u_1, u_2, \dots, u_n)^T$ and M is an M -matrix. A 2-grid problem involves a definition of the coarse space equations

$$\bar{\mathbf{A}}\bar{u} = \bar{b}, \quad \{\bar{\mathbf{A}} \in \mathbb{R}^{N \times N}; \bar{u}, \bar{b} \in \mathbb{R}^N\}$$

through an algebraic transformation where $\bar{\mathbf{A}} = \{\bar{a}_{ij}\}_{N \times N}$, $\bar{u} = (\bar{u}_1, \bar{u}_2, \dots, \bar{u}_N)^T$. These equations now formally play the role of the coarse grid equations as in traditional multigrid methods. This implies that we may view the fine grid Ω as being represented by the set of unknowns $\{u_i : \forall i \in [1..n]\}$ from which we may now construct a graph $G = (V, E)$ based on the matrix. The vertices V are represented by the unknown variable index i and the edges E are defined using the matrix coefficient entries to determine connectivity. In this manner, it is possible to construct the coarse grid $\bar{\Omega}$ by considering the coarse grid variables to be a partitioned subset of the fine grid. Let the fine grid variables be partitioned into two disjoint subsets: The first contains variables contained in the coarse level which we denote as C-variables (or C-nodes). The complementary set is denoted as F-variables (or F-nodes). Classic AMG now considers the construction of a Maximal Independent Subset (MIS) of the C-nodes by using the matrix stencil to define the set S of strongly connected nodes for a given node i [44, 45]. For an M -matrix $\{a_{ij}\}$, a point i is *strongly connected to* j , or *strongly depends on* j if

$$-a_{ij} \geq \theta \max_{k \neq i} \{-a_{ik}\}, \quad 0 < \theta \leq 1 \quad (1.1)$$

An MIS partitioning of a set of vertices in a graph is a subset of vertices which is *independent* in the sense that no two vertices in the subset are connected by an edge, and *maximal* if the addition of a vertex results in a dependent subset. This MIS set is now chosen to represent the coarse grid.

The construction of the interpolation operators has given rise to different algorithms. The original paper by Ruge and Stüben [44] makes use of the assumption that smooth error components are required to be in the range of the interpolation operators, and as such gives rise to the definition for *algebraic smoothness*. Given a simple relaxation scheme G , an algebraically smooth error e is defined as an error which is slow to converge using the scheme G i.e

$$\|Ge\| \sim \|e\|$$

This assumes that for these simple schemes, the residual defined for these smooth errors is small:

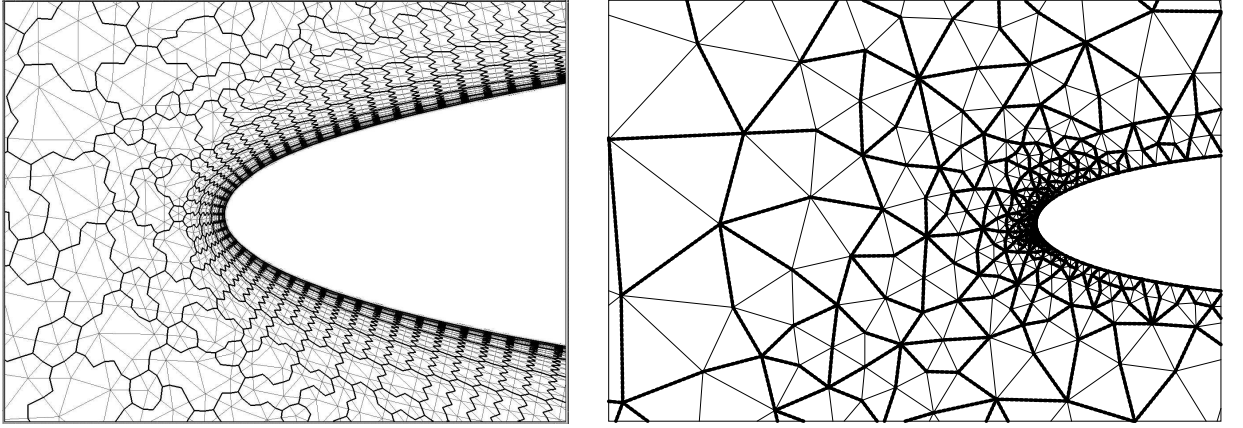
$$\mathbf{A}e = r \sim 0$$

This now provides the basis for the construction of the interpolation operators [44]. This method is efficient for M-matrices [46] and has been extended to the case of general matrices with both positive and negative off-diagonal entries [45]. Brezina *et al* [47] however, make use of a different assumption in the construction of the interpolation operators. This assumption states that the interpolation must be able to approximate an eigenvector of the governing matrix $\mathbf{A}$ with error bound proportional to the size of the associated eigenvalue. They show improved convergence for AMG when applied to certain classes of problems such as 2D linear elasticity.

Classic AMG can be efficient when applied to a wide range of problems such as scalar elliptic problems but can also suffer from many deficiencies. Some of these include:

- Standard AMG cannot be applied to non-linear systems since the underlying principles are based on a linear algebraic formulation.
- Extension of the standard AMG algorithm to block systems of equations is not clear and often leads to poor convergence rates.

A hybrid variant of the classic AMG method is Agglomeration Multigrid which operates by agglomeration of the finite element subspace on the fine grid. One such agglomeration



(a) Nodal Agglomeration

(b) Element Agglomeration

Figure 1.1: Agglomeration Types

technique is the nodal agglomeration technique [48–51] which results in the Additive Correction Multigrid (ACM) method. This involves the agglomeration of connected nodes of the mesh graph as shown in Fig. 1.1(a). However, a known problem with typical nodal agglomeration methods is their inability to accurately represent higher order differential operators on coarse meshes due to low accuracy multigrid transfer operators [36]. In the case of ACM, the transfer operators turn out to be simple injection, i.e. zeroth order operators which are inadequate for even the simplest elliptic problems. Another effective agglomeration method is elemental agglomeration [18, 38, 47, 52] for which higher order transfer operators may be defined. This involves the agglomeration of neighboring elements into macroelements as shown in Fig. 1.1(b). As can be observed, the coarse space elements are not standard elements and appropriate basis functions need to be defined. Chan *et al* [18] present an elemental agglomeration coarsening technique based on the underlying graph of the fine grid that does not involve geometry. This technique produces a set of node-nested coarse spaces

which is retriangulated based on fixed patterns in the agglomerated macroelement. This method offers great potential since the interpolation operators can be based on integers and lead to savings in storage and CPU time. Also, the algorithm recovers the natural structure of the coarse grids if the fine grid is regular. However, since the elemental agglomeration algorithm is purely topology-based, it cannot distinguish between anisotropic and isotropic mesh regions.

The construction of the multigrid transfer operators has been the focus of much research in AMG where the coarse space matrices are created algebraically using these operators. These construction techniques may be classified into matrix based and grid based algorithms. As discussed earlier, the standard interpolation used in classic AMG is a matrix-based method which uses a fine/coarse node partitioning of the global matrix [18, 53]. In contrast to this method is elemental AMG (AMGe) interpolation which constructs a local matrix-based interpolation from the elemental stiffness matrices [47, 52]. An element-free extension of this algorithm which attempts to capture the benefits of AMGe without access to the elemental stiffness matrix has also been reported by Henson *et al* [54]. This leads to significant storage saving and both versions of the AMGe interpolation operators have been shown to exhibit excellent convergence rates for elliptic and elasticity problems.

In the context of structured meshes, mesh stencil-based interpolants [46, 55] can be easily defined. For unstructured meshes, the use of the nodal basis functions has remained a popular choice for the construction of the interpolants [35, 56]. Leclercg *et al* [57] describe the construction and Fourier analysis of an upwind transfer operator for a Finite Volume discretization of the Euler equations and demonstrated the robustness of the interpolant. Extension of the mesh stencil-based interpolant to unstructured meshes for AMG is described by Chan *et al* [18] where several interpolants of varying complexities based on elemental agglomeration are discussed. An energy minimization approach is taken by Wan *et al* [58] which exploits the properties of the underlying PDEs while allowing general computational domains. A smoothed aggregation technique as described by Vaněk *et al* [59, 60] starts with a piecewise constant basis, which has high energy, and then smooths the basis through the application of a relaxation method. This minimizes the energy of the basis locally. A review of these algorithms may be found in [61].

The success of AMG formulations has been mostly limited to scalar elliptic applications where similar mesh independent convergence rates as in GMG have been obtained, as well as 2D elasticity problems [38, 47, 62]. The early attempts at an AMG approach for the Navier-

Stokes equations by Webster [51] and Raw [50] showed some promise but were limited by various factors such as degradation of the convergence rates with large grid sizes. Recent developments by Mavriplis [9,36,63] have shown significant improvements for more practical applications.

1.3 AMG Approach for Navier-Stokes

In this thesis, we consider the development of a multigrid methodology for the solution of convection-diffusion based problems, using stabilized Finite Element discretizations with the final objective of efficiently computing high Reynolds number Navier-Stokes flows.

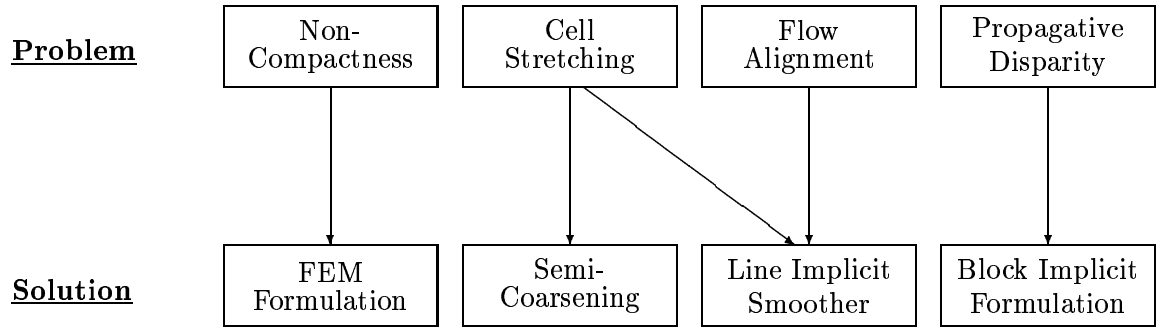


Figure 1.2: Diagnosis of multigrid breakdown for the Navier-Stokes equations and solutions.

Fig. 1.2 delineates the various issues which multigrid methods face when applied to Navier-Stokes flows. For each category, a modification of standard multigrid components is proposed to effectively deal with the issues. For both inviscid and viscous calculations, discrete stiffness can arise from the disparity in the propagation speeds of the acoustic and convective modes [39]. Hence, explicit schemes which use standard scalar preconditioners are severely limited by stability constraints. Flow misalignment with the grid serves to decouple convective error modes in the transverse direction which also leads to a slowdown in convergence rate of standard multigrid schemes. Viscous computations however, introduce an extra source of discrete stiffness due to the use of high aspect ratio cells within the boundary layer. This effect serves to collapse the convective eigenvalues onto the origin while

decoupling the acoustic modes from the streamwise coordinate direction [13]. Motivated by Fourier analysis, Pierce and Giles develop a preconditioned multigrid methodology based on point block implicit smoothing and semi-coarsening. While this leaves some error modes lightly damped, the resulting algorithm is a significant improvement over previous multigrid performances.

Motivated by the inherent problems of the application of standard multigrid to the Euler/Navier-Stokes equations, Brandt advocated that a hyperbolic/elliptic splitting of the advective/acoustic subsystems is a means for obtaining optimal multigrid performance [64]. Based on a local mode analysis for the Euler equations, Brandt shows that the convergence rate for standard multigrid algorithms which utilize full coarsening and scalar preconditioning is limited by the error correction from the coarse spaces. By splitting the acoustic and hyperbolic subsystems into components which are treated separately, Brandt and Yavneh demonstrated optimal multigrid convergence rates for the Euler equations [65]. Extension of this principal idea has been done by a number of researchers such as Thomas *et al* [29] for the incompressible Navier-Stokes equations applied to high Reynolds number wakes and boundary layers as well as the compressible Navier-Stokes equations [66]. Nishikawa *et al* [67] utilize the Van Leer-Lee Roe (VLR) preconditioner to obtain a decomposition of the discrete residual into hyperbolic and elliptic components for a cell-centered Finite Volume discretization of the Euler equations on a structured grid. A full coarsening approach for the elliptic subsystem and a semi-coarsening approach for the hyperbolic subsystem is taken and the formulation is shown to be $O(N)$ with respect to the number of unknowns. However, the current implementations of Brandt's characteristic splitting idea are not yet mature and the applications are still limited in scope.

In the approach this thesis takes towards addressing the hyperbolic and elliptic characteristics of the flow equations, no attempt is made at formulating a discretization which distinguishes between these components. Rather, the multigrid components are specifically designed to separately deal with these components as applied to standard Finite Volume/Finite Element discretizations. A stabilized FEM discretization is chosen to provide accurate yet compact discretizations of convection-dominated flows. This formulation has been made use of by Rannacher *et al* [68] where a multigrid solution scheme for stabilized FEM discretizations is implemented. This is based on a preconditioned GMRES formulation using a defect-correction multigrid as the preconditioner, an injection prolongation operator, an L^2 -projection for the restriction operator and an ILU Gauß-Seidel smoother

with node renumbering in the streamwise direction.

The multigrid implementation in this thesis is based on an Algebraic Multigrid (AMG) formulation for fast and automatic construction of the multigrid components. For strongly advection-dominated flows, line implicit relaxation schemes have been shown to possess good smoothing properties using a Fourier analysis of a structured, constant spacing Finite Difference discretization of a model 2D convection-diffusion equation [27, 46, 69]. Following this reasoning, the choice of a line implicit relaxation scheme is made to deal with the convective modes where the implicit lines are constructed to follow the direction of strongest coupling. Point implicit relaxation methods based on this idea have been implemented for structured meshes [13, 70] and also successfully applied to unstructured mesh formulations [9]. These are typically problems for which a primary flow direction can be identified *a-priori* such that the points can be sorted in this primary direction. Strongly advected flows exhibit characteristic directions along which information is propagated. Hence, the convergence rate for these point implicit methods, which are sensitive to sorting such as Gauß-Seidel, is highly dependent on the flow direction and will probably suffer if there are localized flow regions for which no choice of a preferential direction can be made *a-priori*. A hybrid method by Mavriplis [71] for unstructured meshes makes use of implicit lines within the boundary layer and point implicit relaxation elsewhere. These lines are constructed by linking up nodes where the coupling between the nodes is based on the edge length. This helps to reduce some of the problems associated with the stiffness introduced by the stretched mesh but does not fully address all the issues related to the degradation of the multigrid algorithm. The proposed relaxation scheme is a generalized line implicit relaxation scheme where the nodal coupling is derived from the discretization of the governing differential equation.

The acoustic modes are dealt with by means of the multigrid coarse space which is effective in handling elliptic error modes. An agglomeration multigrid approach which enables the construction of higher order interpolants was chosen as opposed to a pure algebraic methodology. The construction of the coarse spaces is through elemental agglomeration due to the difficulty of nodal agglomeration to accurately represent the second-order derivative terms of the Navier-Stokes equations. The coarse space elements are generalized polygons which obviates the need for retriangulation procedures [38]. The interpolation operator is an extension of an interpolant developed by Chan *et al* [18] which is a linear interpolation on the agglomerated element boundary and constant interpolation over the interior. The current modification extends the linear interpolation over the agglomerated element inte-

rior. The agglomeration algorithm makes use of semi-coarsening to further remove grid induced stiffness.

1.4 Summary

This thesis deals with research into the application of an algebraic multigrid formulation to stabilized Finite Element discretizations of the Euler and Navier-Stokes equations. The core of this dissertation is the development of the multigrid components to effectively deal with the elliptic and hyperbolic characteristics of these equations without the need to resort to specialized discretizations. The discussion in the chapters is ordered by increasing complexity starting with the basic formulation of the multigrid components for scalar elliptic operators and culminating in the final application to select Euler and Navier-Stokes examples.

The first chapter on symmetric elliptic operators begins with the application of AMG to a simple Galerkin FEM discretization of the Poisson equation. This can be viewed as the first step towards addressing the acoustic subsystem of the Euler and Navier-Stokes equations. This chapter focuses on the construction of the interpolation operators and the multigrid coarse spaces. The generalized multigrid algorithm is described and the construction of the multigrid components, especially the elemental agglomeration algorithm, within the AMG context is outlined. Numerical studies into the behavior of the basic multigrid algorithm is made for the Poisson problem on isotropic meshes and the robustness of the algorithm is demonstrated. In addition, the algorithm is applied to the Poisson problem on anisotropic meshes to demonstrate possible issues which arise in the application of standard multigrid algorithms to viscous computations on stretched grids.

The second chapter on convection-diffusion operators introduces hyperbolic characteristics by means of a convective component. The model problem considered is the linear, stationary convection-diffusion equation and a stabilized FEM discretization for this problem is implemented. The application of the multigrid algorithm described in the first chapter is shown to be inadequate for these problems especially in the strongly convective limit. A modification of the multigrid smoother to a line implicit relaxation scheme is described which effectively deals with the hyperbolic characteristics. The application of standard AMG algorithms to stabilized numerical schemes is shown to require a characteristic length based rescaling of the numerical stabilization for a consistent multigrid representation. This

has not been addressed in the literature and a solution for this problem is proposed. Numerical studies into the behavior of the modified multigrid algorithm is done and improved convergence rates are demonstrated.

The next two chapters on 2D Euler and Navier-Stokes applications begins with a description of a GLS/FEM discretization of the equations using symmetrizing entropy variables. The extension of the fully modified multigrid algorithm for the system of equations as well as a solution strategy for the non-linear equations is described. Numerical studies for several test cases are performed and improved convergence rates are demonstrated. Several issues related to the formulation of the stabilization terms are also discussed.

The concluding chapter summarizes the main results of the thesis and provides some suggestions for possible future work.

Chapter 2

AMG: Symmetric Elliptic Operators

The multigrid technique has been applied with great success to elliptic PDEs. For symmetric elliptic operators, efficient multigrid methods exist with mesh independent convergence rates [46]. Theoretical bounds for the convergence rate of the multigrid algorithm applied to these problems have been rigorously proven and provide a point for comparison of new multigrid techniques. This class of problems also provides a basis for the extension of multigrid to harder problems such as convection-diffusion.

In this chapter, we present an Algebraic Multigrid methodology as applied to a Finite Element discretization of the Poisson equation. The generalized multigrid algorithm is described and the construction of the multigrid components within the AMG context is outlined. The main contribution in this chapter is the development of an elemental agglomeration algorithm with improved accuracy interpolation between grid levels. Brandt [17] has shown that the accuracy of the multigrid interpolation operators is important in the construction of a multigrid algorithm with grid independent convergence rates. Furthermore, the behavior of standard multigrid algorithms is known to degrade appreciably in the presence of anisotropy which may be either introduced through variable coefficients or grid stretching [45, 72]. The behavior of the proposed multigrid algorithm when applied to elliptic anisotropic problems is tested and the results analyzed.

2.1 Model Problem

The model problem considered is the symmetric two dimensional diffusion problem represented by the differential equation:

$$\nabla \cdot (\mu(x, y) \nabla u) = -f \quad \text{in } \Omega, \quad (2.1)$$

$$u = g_D \quad \text{on } \Gamma_D, \quad (2.2)$$

$$\mu(x, y) \nabla u \cdot \mathbf{n} = g_N \quad \text{on } \Gamma_N, \quad (2.3)$$

The domain Ω is a bounded domain in $\mathbb{R}^2$ with boundary Γ which is made up of a Dirichlet boundary Γ_D and a Neumann boundary $\Gamma_N = \Gamma \setminus \Gamma_D$. The function $f = f(x, y)$ is a given source function while g_N and g_D are data defined on the boundary Γ with $\mathbf{n}$ being the outward unit normal to Γ . The variational weak formulation of Eq. 2.1 can be stated as:

Find $u \in H^1(\Omega; \Gamma_D)$ such that

$$\int_{\Omega} \mu(x, y) \nabla u \cdot \nabla v \, d\Omega = \int_{\Omega} v f \, d\Omega \quad \forall v \in H_0^1(\Omega; \Gamma_D) \quad (2.4)$$

where $H^1(\Omega; \Gamma_D)$ is the Sobolev space of functions with square integrable first derivatives that satisfy the Dirichlet boundary conditions and $H_0^1(\Omega; \Gamma_D)$ is the space of functions which vanish on Γ_D . A unique solution for Eq. 2.4 is guaranteed if $\mu(x, y)$ is a strictly positive function and $f(x, y)$ is square integrable, subject to admissible boundary conditions [73].

This figure was broken... original no longer exists.

Figure 2.1: Domain and Finite Element Discretization using Linear Triangles

Let us now consider a partitioning of the domain Ω as in Fig. 2.1 into n_e non-overlapping triangular elements Ω^e with boundary Γ^e and n_p vertices such that:

$$\Omega_h = \bigcup_{e=1}^{n_e} \Omega^e \quad (2.5)$$

$$\emptyset = \bigcap_{e=1}^{n_e} \Omega^e \quad (2.6)$$

For a domain with curved boundaries, the Finite Element partitioning of the domain may create a region $|\Omega - \Omega_h| \neq \emptyset$ as shown in Fig. 2.1 which can be minimized by refinement through a reduction in the characteristic element size h_e . Within each element, the solution and geometry is approximated by a set of k th-order interpolation polynomials $\mathcal{P}_k$ which are $\mathcal{C}^0$ continuous across elements and belong to the space $H^{1h} \subset H^1$ defined as:

$$H^{1h} = \{U^h \mid U^h \in \mathcal{C}^0(\Omega_h), U^h|_{\Omega^e} \in \mathcal{P}_k(\Omega^e) \quad \forall \Omega^e \in \Omega_h\} \quad (2.7)$$

The approximate (or trial) Finite Element solution can now be described as belonging to the space $\mathcal{S}^h \subset H^{1h}$ which is the finite dimensional subspace of piecewise k th-order $\mathcal{C}^0$ continuous functions defined on Ω_h :

$$\mathcal{S}^h = \{V^h \mid V^h \in H^{1h}, V^h = g_D \text{ on } \Gamma_D\} \quad (2.8)$$

The classical Galerkin Finite Element Method makes use of a weighted residual formulation based on Eq. 2.4. We introduce the space $\mathcal{V}^h$ of weighting (test) functions which is the same as the trial function space $\mathcal{S}^h$, up to the Dirichlet boundary conditions:

$$\mathcal{V}^h = \{W^h \mid W^h \in H^{1h}, W^h = 0 \text{ on } \Gamma_D\} \quad (2.9)$$

Any function u^h in H^{1h} can be written as a linear combination of a set of nodal basis functions $N_i(x, y)$ such that

$$u^h(x, y) = \sum_i u_i N_i(x, y). \quad (2.10)$$

The coefficients u_i are nodal values $u^h(x_i, y_i)$ and the nodal basis functions are such that

$N_i \in H^{1h}$ and

$$N_i(x_j, y_j) = \delta_{ij}, \quad (2.11)$$

where δ_{ij} represents the Kronecker function. The Galerkin Finite Element formulation for the discrete problem now reduces to:

Find $u_h \in \mathcal{S}^h$ such that

$$B(u_h, w_h)_{gal} + B(u_h, w_h)_{bc} = 0, \quad \forall w_h \in \mathcal{V}^h \quad (2.12)$$

where the forms $B(\cdot, \cdot)_{gal}$ and $B(\cdot, \cdot)_{bc}$ account for the Galerkin and boundary condition terms respectively, and are defined as

$$\begin{aligned} B(u_h, w_h)_{gal} &= \int_{\Omega_h} (\mu \nabla u_h \cdot \nabla w_h - w_h f) \, d\Omega \\ B(u_h, w_h)_{bc} &= \int_{\Gamma_N} \mu w_h (\nabla u_h \cdot \mathbf{n}) \, d\Gamma = \int_{\Gamma_N} w_h g_N \, d\Gamma \end{aligned}$$

The nodal basis functions provide local support such that a choice of $w_h = N_i$, $\{i = 1 \dots n_p\}$ results in Eq. 2.12 being written as a linear system

$$\mathbf{A}_h u_h = b_h, \quad \mathbf{A}_h \in \mathbb{R}^{n_p \times n_p}; \, u_h, b_h \in \mathbb{R}^{n_p} \quad (2.13)$$

where u_h is a vector of nodal unknowns and each row of $\mathbf{A}_h$ corresponds to a different weighting function. For the model problem, we make use of linear $\mathcal{P}_1$ interpolation polynomials which result in the basis function being the so-called hat function. Eq. 2.12 now represents a sparse linear system of equations with the following properties [73]:

1. Symmetric: $\mathbf{A}_h = \mathbf{A}_h^T$
2. Positive definite: $u^T \mathbf{A}_h u > 0 \quad \forall u \neq 0 \text{ and } \Gamma_D \neq \emptyset$
3. Diagonally dominant: $|a_{ii}| \geq \sum_{j \neq i} |a_{ij}|$, $\forall i$ and isotropic elements.

2.2 Multigrid: Operation and Components

Given the linear system of equations $\mathbf{A}_h u_h = b_h$, arising from the Finite Element discretization of Eq. 2.1, we consider a multigrid formulation for the solution. Discrete operators which are derived from Finite Difference, Finite Volume and Finite Element discretizations tend to be large, sparse matrix systems of equations which are well adapted to iterative schemes. The application of an iterative scheme or smoother $\mathcal{S}(u_h^0, b_h, \mathbf{A}_h, n)$ to the solution of Eq. 2.13 results in a better estimate u_h^n to the solution after a given number of iterations n , starting from an initial guess u_h^0 . The iteration error may be defined by:

$$e_h^n = u_h^n - u_h. \quad (2.14)$$

Since u_h^n represents an approximation to the solution, it does not satisfy Eq. 2.13 exactly and so we can define a residual r_h ;

$$r_h = b_h - \mathbf{A}_h u_h^n. \quad (2.15)$$

We may now take the difference between Eq. 2.13 and Eq. 2.15 and further use the linearity of the discrete operator to obtain:

$$r_h = \mathbf{A}_h u_h - \mathbf{A}_h u_h^n \quad (2.16)$$

$$= \mathbf{A}_h \{u_h - u_h^n\} \quad (2.17)$$

$$= \mathbf{A}_h \{-e_h^n\} \quad (2.18)$$

The basic idea behind multigrid is the computation of corrections to the error e_h^n on a coarser grid Ω_H given a set of equations which have been discretized on a fine grid Ω_h . Specifically, consider a restriction operator $\mathbf{R}_h : \mathbb{R}^{n_h} \mapsto \mathbb{R}^{n_H}$ where n_h and n_H are the dimensions of the finite dimensional spaces associated with grids Ω_h and Ω_H . We may now define the rough error component e_h^{rough} of e_h^n as error components for which

$$\mathbf{R}_h e_h^{\text{rough}} = 0, \quad (2.19)$$

such that the smooth error component e_h^{smooth} is simply the complement. This ensures that the transfer of the error to the coarse grid involves the smooth error component only.

Hence, the application of a smoother $\mathcal{S}(u_h^0, b_h, \mathbf{A}_h, n)$ to the current estimate of the solution implies that the selected smoother must be effective in the elimination of e_h^{rough} . A coarse grid representation of Eq. 2.18 which can be more efficiently solved is

$$\mathbf{A}_H e_H^n = -\mathbf{R}_h r_h \quad (2.20)$$

where $\mathbf{A}_H$ and e_H^n are the coarse grid representations of $\mathbf{A}_h$ and e_h^n respectively. The error correction e_H^n computed on the coarse grid is now interpolated back to the fine grid and used to update the solution by means of the prolongation operator $\mathbf{P}_h : \mathbb{R}^{n_H} \mapsto \mathbb{R}^{n_h}$,

$$u_h^n \leftarrow u_h^n - \mathbf{P}_h e_H^n. \quad (2.21)$$

2.2.1 Multilevel Algorithm

Given the basic description of the multigrid process, we may now consider a recursive multilevel formulation. Let $\{\Omega_k : (k = 0, \dots, m)\}$ represent a hierarchy of grids such that $\Omega_0 = \Omega_h$. Let n_k be the dimension of the finite dimensional vector space $\mathbb{R}^{n_k}$ associated with each grid Ω_k such that $n_k > n_{k+1}$ and let $\{\mathbf{A}_k : (k = 0, \dots, m), \mathbf{A}_k \in \mathbb{R}^{n_k \times n_k}\}$ be the representations of $\mathbf{A}_h$ on these coarse grids such that $\mathbf{A}_0 = \mathbf{A}_h$. Also, let $\{\mathbf{R}_k : \mathbb{R}^{n_k} \mapsto \mathbb{R}^{n_{k+1}}\}$ and $\{\mathbf{P}_k : \mathbb{R}^{n_k} \mapsto \mathbb{R}^{n_{k-1}}\}$ represent the restriction and prolongation operators defined on these spaces respectively. Reduction of rough error components on each grid level k requires a smoother and these smoothers may be different on each grid level depending on $\mathbf{R}_k$ but are typically chosen to be the same.

A recursive application of the 2-grid multigrid algorithm can now be constructed on a sequence of coarser grids where the error on each coarse grid level k may also be decomposed into smooth and rough components by means of a suitable definition for $\mathbf{R}_k$. This leads to the m -grid linear multigrid algorithm (Algorithm 1) where $\mathcal{S}(u, b, \mathbf{A}, \nu)$ is the smoothing procedure.

In our implementation of multigrid, the coarsening procedure terminates when the coarse grid system of equations $\mathbf{A}_m u_m = b_m$ is small enough to be solved exactly. Depending on the scheduling of operations between the coarse spaces, we end up with different multigrid cycles. Two of the most common cycles are the $V(\nu_1, \nu_2)$ -cycle ($\gamma_k = 1$) and the $W(\nu_1, \nu_2)$ -cycle ($\gamma_k = 2$) which are depicted in Fig. 2.2.

Algorithm 1 $\text{MG}(u_k, b_k, \nu_1, \nu_2, m)$

```

if  $k = m$  then
  Set  $u_m = (\mathbf{A}_m)^{-1} b_m$ 
else
  Perform  $\nu_1$  smoothing sweeps:  $\mathcal{S}(u_k, b_k, \mathbf{A}_k, \nu_1)$ 
  Compute residual:  $r_k = b_k - \mathbf{A}_k u_k$ 
  Restrict residual:  $b_{k+1} = \mathbf{R}_k r_k$ 
  Initialize correction:  $u_{k+1} = 0$ 
  for  $i = 0$  to  $\gamma_k$  do
    “Solve” on level  $k+1$ :  $\text{MG}(u_{k+1}, b_{k+1}, \nu_1, \nu_2, m)$ 
  end for
  Correct solution:  $u_k \leftarrow u_k + \mathbf{P}_k u_{k+1}$ 
  Perform  $\nu_2$  smoothing sweeps:  $\mathcal{S}(u_k, b_k, \mathbf{A}_k, \nu_1)$ 
end if

```

This figure was broken... original no longer exists.

Figure 2.2: V- and W-cycles

2.2.2 Multigrid Smoother

Let us consider a general coarse space matrix system given by

$$\mathbf{A}_k u_k = b_k. \quad (2.22)$$

Let us also consider a splitting of the matrix $\mathbf{A}_k$:

$$\mathbf{A}_k = \mathbf{M}_k - \mathbf{N}_k \quad (2.23)$$

where $\mathbf{M}_k$ is non-singular. The basic idea in defining a smoother $\mathcal{S}(u_k, b_k, \mathbf{A}_k, \mathbf{M}_k, \nu)$ is to obtain a matrix $\mathbf{M}_k$ such that the rough error modes are effectively reduced while the inversion of $\mathbf{M}_k$ is much less expensive than $\mathbf{A}_k$. A basic iterative method for the system is defined as the following linear fixed-point iteration:

$$u_k^{i+1} = u_k^i + \mathbf{M}_k^{-1}(b_k - \mathbf{A}_k u_k^i) \quad (2.24)$$

$$= \mathbf{S}_k u_k^i + \mathbf{M}_k^{-1} b_k$$

$$\mathbf{S}_k = \mathbf{I} - \mathbf{M}_k^{-1} \mathbf{A}_k \quad (2.25)$$

$$= \mathbf{M}_k^{-1} \mathbf{N}_k$$

where u_k^i represents the current solution estimate at iteration i . The matrix $\mathbf{M}_k$ is the preconditioning matrix and $\mathbf{S}_k$ is called the iteration matrix or smoother. It may be shown that the iteration error $e_k^i = u_k^i - u_k$ satisfies

$$e_k^{i+1} = \mathbf{S}_k e_k^i \quad (2.26)$$

$$= (\mathbf{S}_k)^i e_k^0 \quad (2.27)$$

and the residual $r_k^i = b_k - \mathbf{A}_k u_k^i$ satisfies

$$r_k^{i+1} = \mathbf{A}_k \mathbf{S}_k \mathbf{A}_k^{-1} r_k^i \quad (2.28)$$

Hence, for the basic iterative method to be convergent

$$\lim_{i \rightarrow \infty} \|\mathbf{S}_k^i\| = 0 \quad (2.29)$$

where $\|\mathbf{A}\|$ is the contraction number of the matrix $\mathbf{A}$ for any vector norm $\|\cdot\|$, and is defined as

$$\|\mathbf{A}\| = \sup_{x \neq 0} \frac{\|\mathbf{A}x\|}{\|x\|} \quad (2.30)$$

Theorem 2.1 *The basic iterative method Eq. 2.24 will converge for any initial guess u_k^0 iff*

$$\rho(\mathbf{S}_k) < 1 \quad (2.31)$$

where $\rho(\mathbf{S}_k)$ is the spectral radius of the matrix $\mathbf{S}_k$. Convergence conditions for many relaxation schemes may be found in [46]. Damping may also be introduced using a relaxation factor ω by defining:

$$\begin{aligned} u_k^{i+\frac{1}{2}} &= \mathbf{S}_k u_k^i + \mathbf{M}_k^{-1} b_k \\ u_k^{i+1} &= \omega u_k^{i+\frac{1}{2}} + (1 - \omega) u_k^i \\ &= \mathbf{S}_k^* u_k^i + \omega \mathbf{M}_k^{-1} b_k \end{aligned} \quad (2.32)$$

where

$$\mathbf{S}_k^* = \omega \mathbf{S}_k + (1 - \omega) \mathbf{I}$$

It may be shown that Eq. 2.32 corresponds to the splitting

$$\mathbf{M}_k^* = \mathbf{M}_k / \omega, \quad \mathbf{N}_k^* = \mathbf{A}_k - \mathbf{M}_k^* \quad (2.33)$$

Let us consider the general splitting $\mathbf{A}_k = \mathbf{D}_k - \mathbf{L}_k - \mathbf{U}_k$ where $\mathbf{D}_k$ is the matrix diagonal, and $-\mathbf{L}_k$ and $-\mathbf{U}_k$ are the strictly lower and upper triangular parts of $\mathbf{A}_k$, respectively. Two popular relaxation schemes we will use throughout this thesis are given by:

$$\mathbf{M}_k = \begin{cases} \mathbf{D}_k & \text{Jacobi} \\ \mathbf{D}_k - \mathbf{L}_k & \text{Gau\ss - Seidel} \end{cases} \quad (2.34)$$

The discrete system defined by Eq. 2.12 is characterized by a linear symmetric positive definite (SPD) operator for which many theoretical properties have been rigorously proven.

For SPD matrices, the Gauß-Seidel method always converges for any initial guess while the Jacobi method converges for any initial guess if the matrix is strictly diagonally dominant [74]. The Jacobi and Gauß-Seidel method are two very popular choices for their simplicity and ability to quickly damp out spatially rough error modes. However, the spectral radius for both the Jacobi and Gauß-Seidel methods is such that

$$\rho(\mathbf{M}_h^{-1} \mathbf{A}_h) = 1 - O(h^2). \quad (2.35)$$

Hence, without multigrid, the asymptotic convergence rate of these methods deteriorates rapidly with increasing grid size.

2.2.3 Interpolation and Coarse Grid Operators

The multigrid algorithm requires an approximation of the fine grid operator on the hierarchical coarse spaces. The coarse space matrix $\mathbf{A}_k$ may be constructed by rediscrretization, however Algebraic Multigrid (AMG) considers an algebraic alternative to this [46] given by the recursive relation:

$$\mathbf{A}_{k+1} = \mathbf{R}_k \mathbf{A}_k \mathbf{P}_k \quad (2.36)$$

Although the $\mathbf{R}_k$ and $\mathbf{P}_k$ operators are independent, the choice of $\mathbf{R}_k = \mathbf{P}_k^T$ results in the minimization of the error in the solution after coarse grid corrections when measured in the A-norm, $\|\cdot\|_{\mathbf{A}_k}$, for SPD operators. Given Eq. 2.36, let us assume the restriction and prolongation operators are independent. Let v_{k+1} represent the correction from the coarse grid and $\bar{u}_k$ the current solution estimate on the fine grid. The error in the solution after correction is thus

$$e_k = \bar{u}_k + \mathbf{P}_k v_{k+1} - \mathbf{A}_k^{-1} b_k$$

Let us measure the error in the A-norm, $\|\cdot\|_{\mathbf{A}_k}$ and minimize the error:

$$\begin{aligned} \min_{v_{k+1}} \|e_k\|_{\mathbf{A}_k} &= \min_{v_{k+1}} \|(\bar{u}_k + \mathbf{P}_k v_{k+1}) - \mathbf{A}_k^{-1} b_k\|_{\mathbf{A}_k} \\ &= \min_{v_{k+1}} (\bar{u}_k + \mathbf{P}_k v_{k+1} - \mathbf{A}_k^{-1} b_k)^T \mathbf{A}_k (\bar{u}_k + \mathbf{P}_k v_{k+1} - \mathbf{A}_k^{-1} b_k) \end{aligned} \quad (2.37)$$

Differentiation of the quadratic form with respect to v_{k+1} and application of the stationarity condition gives

$$\mathbf{P}_k^T(\mathbf{A}_k + \mathbf{A}_k^T)(\bar{u}_k + \mathbf{P}_k v_{k+1} - \mathbf{A}_k^{-1} b_k) = 0 \quad (2.38)$$

For a symmetric matrix $\mathbf{A}_k$, we may easily solve for v_{k+1} and obtain

$$v_{k+1} = (\mathbf{P}_k^T \mathbf{A}_k \mathbf{P}_k)^{-1} \mathbf{P}_k^T (b_k - \mathbf{A}_k \bar{u}_k) \quad (2.39)$$

The Hessian matrix for Eq. 2.37 is

$$\mathbf{H} = 2\mathbf{P}_k^T \mathbf{A}_k \mathbf{P}_k \quad (2.40)$$

which shows that Eq. 2.39 is a minimum when $\mathbf{A}_k$ is positive definite. The coarse grid correction from the linear multigrid algorithm is:

$$v_{k+1} = (\mathbf{A}_{k+1})^{-1} \mathbf{R}_k (b_k - \mathbf{A}_k \bar{u}_k) \quad (2.41)$$

Comparing Equations 2.36, 2.39 and 2.41, we find that

$$\mathbf{R}_k = \mathbf{P}_k^T \quad (2.42)$$

The same argument, however, cannot be made for a non-symmetric matrix. This choice for the interpolation operators also has the added advantage that only one of the operators needs to be constructed. This algebraic method for coarse space operator construction is called the Galerkin Coarse Grid Approximation (GCA).

Mesh size independent convergence rate for the multigrid GCA formulation makes use of a fundamental rule for the accuracy of the interpolation operators [17, 46]. This may be simply stated as:

Theorem 2.2 *Let m_P and m_R be the order (degree plus one) of the polynomials that are interpolated exactly by the prolongation $\mathbf{P}$ and restriction $\mathbf{R}$ operators respectively, and $2m$ be the order of the governing partial differential equation. A necessary condition for mesh independent convergence is*

$$m_P + m_R > 2m \quad (2.43)$$

Chan *et al* [18] carry out a convergence analysis for domain decomposition based subspace correction methods as applied to FEM discretizations of the Poisson problem using linear elements. In order to obtain grid independent convergence rates, the subspaces have to satisfy certain properties. This analysis is extended to the convergence rate of the multigrid method where the definition of the space associated with Ω_k is obtained by interpolation in Ω_{k-1} . According to the analysis in [18], these multigrid subspaces must satisfy stability and approximation properties to ensure grid independent convergence. These properties are:

$$|\mathcal{R}u|_{1,\Omega} \leq C|u|_{1,\Omega}, \quad (\text{stability}) \quad \forall u \in H^{1h}(\Omega) \quad (2.44)$$

$$\|\mathcal{R}u - u\|_{0,\Omega} \leq Ch|u|_{1,\Omega} \quad (\text{approximation}) \quad \forall u \in H^{1h}(\Omega) \quad (2.45)$$

where $\mathcal{R}$ is some continuous interpolation or projection operator unto the subspaces and is related to $\mathbf{R}_k$. Eq. 2.45 is closely related to the standard Finite Element theory of interpolating polynomials for linear elements [75] while Eq. 2.44 is a statement regarding the smoothness of the interpolation operator. If we consider the use of a zeroth degree interpolant (injection), Eq. 2.44 is violated since it does not lie in H^{1h} . Further consideration of Eq. 2.45 implies that the interpolating operator $\mathbf{R}_k$ has to be at least linear. If we make this minimal assumption about the order of $\mathbf{R}_k$, then Theorem 2.2 is automatically satisfied for any choice of interpolation order for $\mathbf{P}_k$. Hence, Theorem 2.2 is a more general statement of the approximation and stability properties. For a more detailed convergence theory, we refer to [19–21, 76–80].

A promising technique for the construction of these interpolation operators which satisfy the rules outlined above is based on the agglomeration technique [81, 82] which operates by fusing neighboring fine grid entities to form coarse grid macroentities. This provides a natural and automatic way for coarse space construction. The agglomeration technique defined on the Finite Element space can be vertex based (nodal) [48–51] or element based [47, 52]. Our choice of elemental agglomeration is motivated by the need to address higher order accuracy for the interpolation operators and has been the subject of recent research [18, 38, 47, 54, 83].

The standard use of nodal agglomeration [48–51] to construct the interpolation operators results in the definition of the restriction as an injection operator. This presents two fundamental problems for discretizations of higher order differential operators. First, this

operator violates the stability property. Secondly, m_P and m_R are unity and for the Laplacian operator ($2m = 2$), the accuracy condition (Eq. 2.43) is violated [46]. This results in suboptimal convergence rates due to mesh dependent scaling errors in the coarse grid corrections.

Coarse Grid	Coarse Grid Op.	Restriction	Prolongation	Convergence
(a) Independent	Rediscretization	Linear	Linear	0.100
(b) Triangulated Seed pts	Rediscretization	Linear	Linear	0.125
(c) Agglomerated	GCA	Injection	Injection	0.512
(d) Agglomerated	Scaled GCA	Injection	Injection	0.254

Table 2.1: Effect of Coarse Grid Operator for Agglomeration Multigrid [36]

Table 2.1 is a reproduction of the results by Mavriplis [36] for the two grid solution of the isotropic Laplace equation using a multigrid V-cycle with three pre- and post-smoothing Jacobi sweeps on the fine grid, and 20 sweeps on the coarse grid. A comparison for the multigrid convergence rates is made for (a) a two grid overset mesh approach using an independently generated coarse mesh, (b) a coarse triangular mesh generated by triangulating the seed points of the agglomerated coarse grid, (c) a nodal agglomeration implementation of the GCA formulation and (d) a scaled nodal agglomeration implementation of the GCA formulation where the coarse space matrix is scaled to take the interpolation order deficiency into account. As can be observed in Table 2.1, there is appreciable degradation in the convergence rate when the interpolation operators are of zeroth degree. Appendix B outlines a brief one dimensional proof of the inadequacy of nodal agglomeration when applied to the solution of a simple Poisson equation.

2.3 Coarse Space Agglomeration

As discussed, the standard application of nodal agglomeration techniques results in sub-optimal convergence rates when applied to higher order differential equations due to the violation of the interpolation order rule (Eq. 2.43). An alternative to this is the elemental agglomeration technique which enables us to easily construct higher order interpolants that satisfy the interpolation order rule.

The proposed algorithm is based on the fusion of elements into macroelements with a

subsequent definition of the coarse grid topology and basis functions. This method is applied recursively to generate the hierarchy of coarse spaces. A review of elemental agglomeration is given by Chan *et al* [38] who also propose alternative elemental agglomeration algorithms.

In our approach to elemental agglomeration, the driving force behind the agglomeration is the reduction of mesh anisotropy which becomes important later in the discretization of convection-diffusion type equations [6, 71]. This is however, less important for FEM discretizations of isotropic elliptic problems. One important distinction between this proposed method and that described by Chan *et al* [38] is that the coarse mesh elements are not converted into standard elements by a retriangulation but are generalized polygons formed by the agglomerated fine mesh elements. This is especially attractive in 3D because of the complicated rules which may be involved for the retriangulation [38]. One drawback of this formulation is that the support for the basis functions defined on these macroelements is larger than standard triangular elements. This algorithm also has the feature that the resulting coarse space is nested. This implies that the interpolants automatically satisfy the stability condition (Eq. 2.44) [18, 38, 52].

2.3.1 Coarse Space Topology

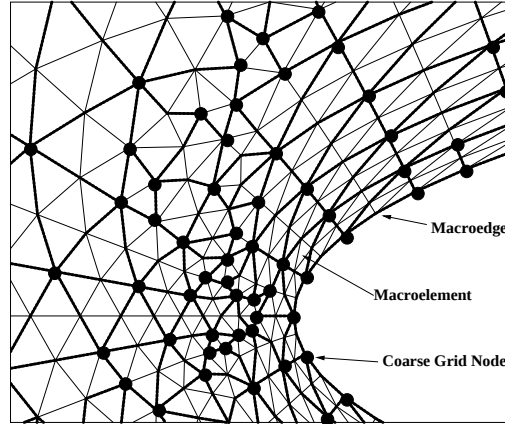


Figure 2.3: Coarse Space Topology

The coarse grid topology is constructed by partitioning the elements into macroelement

groups as shown in Fig. 2.3 for a 2D mesh. A macroedge is defined to be the ordered collection of fine grid edges which are shared by two neighboring macroelements. To complete the definition of the coarse grid graph, the coarse nodes are chosen to be the fine grid nodes where three or more macroedges meet. Macroelements with exactly two coarse nodes are modified by the addition of extra supporting coarse nodes using fine grid nodes which lie on the macroedge connecting these two coarse nodes as shown in Fig. 2.4

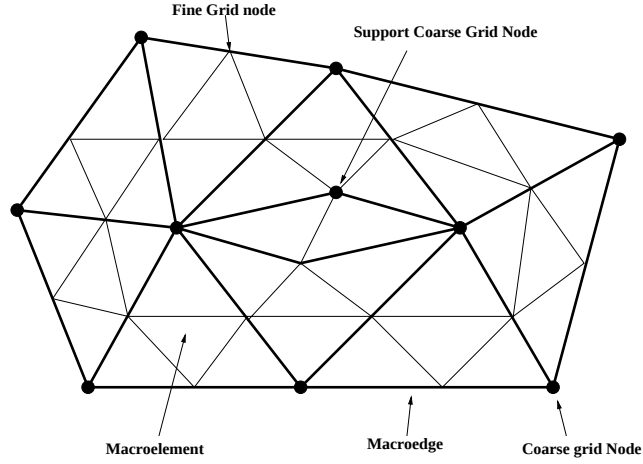


Figure 2.4: Coarse space topology with exceptional macroelement bearing two coarse nodes and extra support node

2.3.2 Elemental Agglomeration Algorithm

The proposed algorithm is based on the removal of grid anisotropy and makes use of edge lengths such that the geometry for the coarse spaces is defined entirely in terms of the fine grid, i.e. the macroedge lengths are simply the sum of the edge lengths of the constituting fine grid edges. If the fine grid geometry is not specified, then this technique becomes a purely topological one where the elements are assumed to be isotropic. The decision to agglomerate two neighboring elements is determined by a geometry based connectivity concept which we term *macroelement skew*.

Definition 1 For a macroelement defined by a general polygon, the macroelement skew is a measure of anisotropy and is defined as the area of the n -gon divided by the area of an

isotropic n -gon with the same perimeter.

In the extreme cases, this is zero for co-linear polygon vertices and unity for an isotropic n -gon. Macroelement skew can be extended to 3D through a suitable redefinition such as ratio of macroelement volume to macroelement circumsphere volume similar to the control volume skew described by Venkatakrishnan *et al* [84]. The macroelemental areas for the coarse spaces are also easy to compute as they are simply sums of the agglomerated element areas. In order to complete the operators required for this algorithm, we need to define an edge based connectivity concept which we term *edge skew*.

Definition 2 *For an element which borders a macroelement/element on a given edge, edge skew is defined as the macroelement skew of the macroelement which would be created if the element is merged with the macroelement/element across that edge.*

We now present the algorithm in detail:

Procedure 1 (*Macroelement Construction*)

Step 0: *Consider the graph of the mesh: $G = (V, E)$ and calculate the edge length for the edges E .*

Step 1: *Obtain seed element: If there is no seed element in the queue, choose any suitable element which does not belong to a macroelement group.*

Step 2: *Perform accretion around the seed element. Fuse unassigned neighboring elements with edge skew larger than some specified fraction (typically 0.75) of the average edge skew.*

Step 3: *Enqueue seed elements. New seed elements are placed in the queue to continue the algorithm. These are chosen to be elements which share a vertex but no edges with the last macroelement created. In 3D, this would extend to elements which share a vertex and/or an edge but no faces with the macroelement.*

Step 5: *Repeat Step 1 until either all elements belong to a macroelement or there are no more seed elements.*

After the algorithm terminates, post-processing is necessary to deal with “sliver” elements. These are fine mesh elements which were not originally selected by the algorithm to be merged into a macroelement. A determination of which macroelement to merge these elements with is made a-priori based on edge skew. In the case where the lengths and areas are equal, the algorithm degenerates to a 4:1 isotropic agglomeration in 2D and fully recovers the natural coarse structure for a regular grid. Unstructured mesh examples are shown in Fig. 2.5.

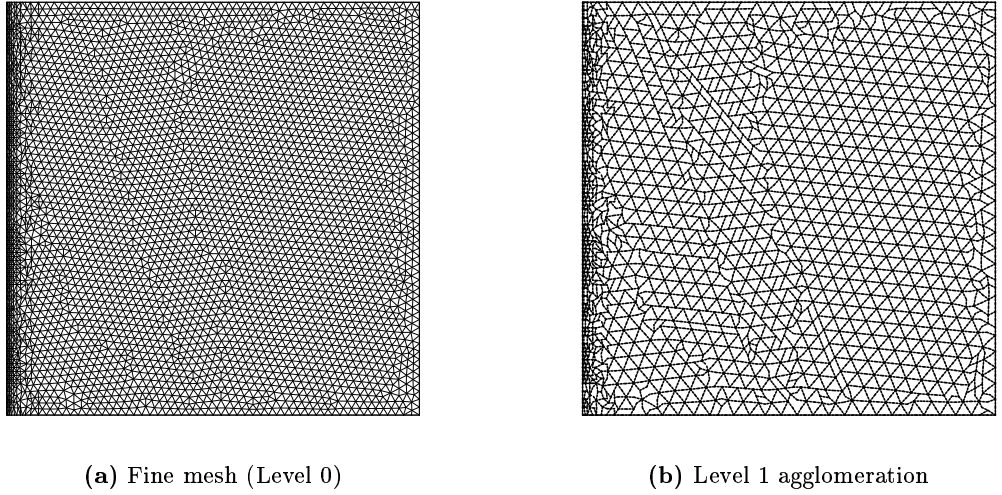


Figure 2.5: Multilevel Elemental Agglomeration Example

2.3.3 Coarse Space Basis Functions

The construction of the interpolation operators may be facilitated by the definition of nodal basis functions on the coarse space and serves as a natural extension of the Finite Element algorithm on these coarse spaces. As outlined by Chan [18], a number of desirable properties have to be satisfied by the coarse spaces to ensure a good convergence rate. These are summarized briefly as:

- *Smoothness*: To guarantee satisfaction of the stability property (2.44)
- *Approximation*: To guarantee satisfaction of the approximation property (2.45)

- We require the basis functions to at least satisfy the stability (Eq. 2.44) and approximation (Eq. 2.45) conditions, preserve the constant function, and behave like standard interpolants i.e

$$\Phi_i(\mathbf{x}_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \quad (2.46)$$

Figure 2.6: Coarse Space Basis Function Based on Graph Distance

Fig. 2.6 shows the basis function Φ defined over the agglomerated macroelement for the coarse grid node at $\mathbf{x}_0$. This basis function is constructed from a graph distance interpolation over the macroelement and weighted with edge length. It satisfies Eq. 2.46 such that it has a value of 1 at $\mathbf{x}_0$ and 0 at every other coarse grid node. Interpolation over the macroelement interior as well as the boundary is also present.

In order to fully describe the algorithm, we give a description of an important component called the Breadth First Search (BFS) algorithm which is essentially a Greedy algorithm for graph traversal. The definition for the BFS algorithm [85] is

Definition 3 *The Breadth First Search (BFS) is a search algorithm which considers neighbors of a vertex, that is, outgoing edges of the vertex's predecessor in the search, before any outgoing edges of the vertex such that extremes are searched last.*

The BFS algorithm forms an integral part of the coarse space basis function construction and the detailed description for the BFS algorithm is given in Algorithm 2.3.3:

Algorithm 2 BFS

```

Unmark all vertices
Choose some starting vertex x
Mark x
Set list L = x
Set tree T = x
Set level set (LS) of x = 0
while L nonempty do
  Choose some vertex v from front of list
  Visit v
  for each unmarked neighbor w do
    Mark w
    Set LS(w) = LS(v) + 1
    Add it to end of list
    Add edge v-w to T
  end for
end while

```

The detailed algorithm for the construction of the coarse space interpolant as well as the coefficients for the prolongation operator $\mathbf{P}_k$ in 2D is given below. The restriction operator $\mathbf{R}_k$ is simply defined using the GCA formulation.

Procedure 2 (*Basis Function Construction*)

Step 1: *For each macroelement, create a local subgraph. In the process, create an ordering of the boundary edges such that the boundary can be traversed.*

Step 2: *Extract the list of interior vertices. Extract the ordered list of coarse grid vertices by traversing the boundary edges.*

Step 3: *For all fine grid edge vertices which lie between consecutive coarse grid nodes, construct length weighted interpolation data. The macroedge length is also computed simultaneously.*

Step 4: *Interior vertex interpolation. For each coarse grid node in the macroelement, a Breadth First Search (BFS) iteration on the local subgraph is done with the coarse grid node as a seed. Both the level set as well as the graph distance from the coarse node is recorded for all interior (fine) nodes in the subgraph during the process. The graph distance of each fine grid node from the coarse grid nodes is then computed. For each fine grid node, these distances are then weighted to sum to unity.*

The necessary matrix coefficients for the prolongation operator $\mathbf{P}_k$ may now be extracted from the basis functions as follows.

Procedure 3 (*Prolongation Operator Construction*)

Step 1: *For every fine grid node i in macroelement which corresponds to a coarse grid node j , set the prolongation operator coefficient*

$$\mathbf{P}_k(i, j) = 1 \quad (2.47)$$

Step 2: *For every other fine grid node i in macroelement which does not correspond to a coarse grid node, given the length weighted graph distance $\text{dist}(i, j)$ from every coarse node j , set the prolongation operator coefficient*

$$\mathbf{P}_k(i, j) = \frac{\frac{1}{\text{dist}(i, j)}}{\sum_j \frac{1}{\text{dist}(i, j)}} \quad (2.48)$$

2.4 Consistency Scaling Formulation

The success of the multigrid methods depends heavily on how good of an approximation the coarse space matrix $\mathbf{A}_{k+1}$ is to $\mathbf{A}_k$. Let us choose the restriction operator to be

$$\mathbf{R}_k = \sigma \mathbf{P}_k^* \quad (2.49)$$

where $\mathbf{P}_k^*$ is the formal adjoint of the prolongation and σ is a suitable scaling factor. The scaling of $\mathbf{R}_k$ is determined by the role of $\mathbf{R}_k$. If $\mathbf{R}_k$ is to be used to construct coarse grid representations of the fine grid unknown u_k , then

$$\sum_j \mathbf{R}_k(i, j) = 1$$

However, if $\mathbf{R}_k$ is to be used to transfer residuals to the coarse grid, then the correct value of the scaling depends on the scaling of the fine grid and coarse grid problems. This implies that the coarse grid discretization should be consistent with the governing PDE in the same way as the fine grid discretization. Let A_j represent a characteristic area (e.g control volume area) on the fine grid associated with fine node j and let $\bar{A}_i$ represent a corresponding characteristic area on the coarse grid associated with coarse node i . Finite Volume and Finite Element schemes in 2D lead to a scaling rule which states that

$$\sum_j \mathbf{R}_k(i, j) A_j = \bar{A}_i$$

This is derived by considering the integral terms for the interior fluxes. However, the boundary flux integral terms are line integrals which necessitates a modification of the restriction operator. Let L_j represent a characteristic length on the fine grid associated with boundary fine node j and let $\bar{L}_i$ represent a corresponding characteristic length on the coarse grid associated with boundary coarse node i . The corresponding scaling rule for the restriction operator as applied to the boundary integral terms is

$$\sum_j \mathbf{R}_k(i, j) L_j = \bar{L}_i$$

In order to deal with the dual scaling issues, we introduce the splitting:

$$\mathbf{A}_k = \mathbf{A}_k^{gal} + \mathbf{A}_k^{bc} \quad (2.50)$$

where $\mathbf{A}_k^{gal}$ consists of the Galerkin terms which scale with area and $\mathbf{A}_k^{bc}$ consists of the boundary condition terms which scale with length. We may now correspondingly split $\mathbf{R}_k$ into $\mathbf{R}_k^{gal}$ and $\mathbf{R}_k^{bc}$ such that GCA definition for the coarse space matrix becomes

$$\mathbf{A}_{k+1} = \mathbf{R}_k^{gal} \mathbf{A}_k^{gal} \mathbf{P}_k^{gal} + \mathbf{R}_k^{bc} \mathbf{A}_k^{bc} \mathbf{P}_k^{bc} \quad (2.51)$$

where

$$\mathbf{R}_k^{gal} = \boldsymbol{\sigma}^{gal} \mathbf{P}_k^{galT} \quad (2.52)$$

$$\mathbf{R}_k^{bc} = \boldsymbol{\sigma}^{bc} \mathbf{P}_k^{bcT} \quad (2.53)$$

The construction for $\mathbf{P}_k^{gal}$ exactly follows the algorithm described in Sec. 2.3.3. However, $\mathbf{P}_k^{bc}$ is constructed by deleting the row entries for all the interior fine grid nodes in $\mathbf{P}_k^{gal}$. This works because interpolation for the fine grid boundary nodes is based on the coarse grid boundary nodes. The construction of the scaling matrices $\boldsymbol{\sigma}^{gal}$ and $\boldsymbol{\sigma}^{bc}$ is done by looping through the elements/edges and sending element/edge contributions (area/length divided by the number of element/edge vertices) to the associated vertices. However, the actual operator used in the prolongation process is $\mathbf{P}_k^{gal}$.

Given the modification for the restriction operator, we consider the following splitting for the residual

$$r_k = r_k^{gal} + r_k^{bc} \quad (2.54)$$

such that a restriction for the residual can be written as

$$b_{k+1} = \mathbf{R}_k^{gal} r_k^{gal} + \mathbf{R}_k^{bc} r_k^{bc} \quad (2.55)$$

However, restriction at the boundary poses a special problem. In general, it is not possible to ensure that $b_{k+1} = 0$ if $r_k = 0$ unless the restriction operators are specially designed to satisfy this condition. The proposed solution to this issue is to lump r_k^{gal} with r_k^{bc} on the

boundary nodes according to:

$$b_{k+1}^{bc} = \mathbf{R}_k^{bc} (r_k^{gal} + r_k^{bc}) \quad (2.56)$$

This has been found to work well in practice.

2.5 Dirichlet Boundary Modification

This figure was broken... original no longer exists.

Figure 2.7: Dirichlet Boundary Modification for $\mathbf{P}_k$

The prolongation operator needs to be modified in the presence of Dirichlet boundary conditions where the correction to the fine grid solution is zero [46]. For a given fine grid node adjacent to a Dirichlet boundary, any matrix coefficient for this node in $\mathbf{P}_k$ which corresponds to a dependency on a coarse grid node that lies on the Dirichlet boundary, is set to zero. This may be easily exemplified by Fig. 2.7. Let us consider fine grid node A such that the standard prolongation operator defined for this node is:

$$v_k^A = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix} \begin{Bmatrix} u_{k+1}^a \\ u_{k+1}^b \\ u_{k+1}^d \\ u_{k+1}^e \end{Bmatrix} \quad (2.57)$$

The presence of the Dirichlet boundary requires a modification of the prolongation since there is no coarse grid correction for nodes on this boundary i.e $u_{k+1}^a = u_{k+1}^d = u_{k+1}^g = 0$.

Hence, we redefine the prolongation for the fine grid node A as:

$$v_k^A = \begin{bmatrix} 0 & \frac{1}{4} & 0 & \frac{1}{4} \end{bmatrix} \begin{Bmatrix} u_{k+1}^a \\ u_{k+1}^b \\ u_{k+1}^d \\ u_{k+1}^e \end{Bmatrix} \quad (2.58)$$

such that the modified prolongation is employed in the GCA construction of the coarse grid equations. This has been found to work well in practice and is required for block systems of equations where Dirichlet boundary conditions may be imposed on a subset of the unknown vector block such as no-slip boundary conditions in discretizations of the Navier-Stokes equations.

2.6 Anisotropic Problems

The convergence rate of the standard multigrid algorithm is severely degraded when anisotropic effects are present. The introduction of anisotropy may be done either through grid stretching, which is important in convection-diffusion problems where boundary layers are present, or through variable coefficients in the governing differential equation. The breakdown of the multigrid algorithm is even more severe for AMG where the coarse space operators are constructed through algebraic means. Improper formulations which may exist in the interpolation operators are manifested more strongly in these coarse space operators than Geometric Multigrid where the coarse space operators are rediscretized independently of the fine space.

The main problem associated with anisotropy is due to the decoupling of the error modes into preferential directions. Using Fourier analysis applied to the anisotropic diffusion equation, Wesseling [46] shows that many of the popular smoothers break down in the presence of anisotropy.

Anisotropy can be dealt with through modification of the smoother to perform implicit sweeps along the preferential directions [70,71,86] as well as the coarse space semi-coarsening algorithm [6,71,72,87] where the aim of the coarse space construction is the reduction of mesh anisotropy. The elemental agglomeration algorithm described above is designed to automatically include semi-coarsening when the fine grid geometry is supplied and an example

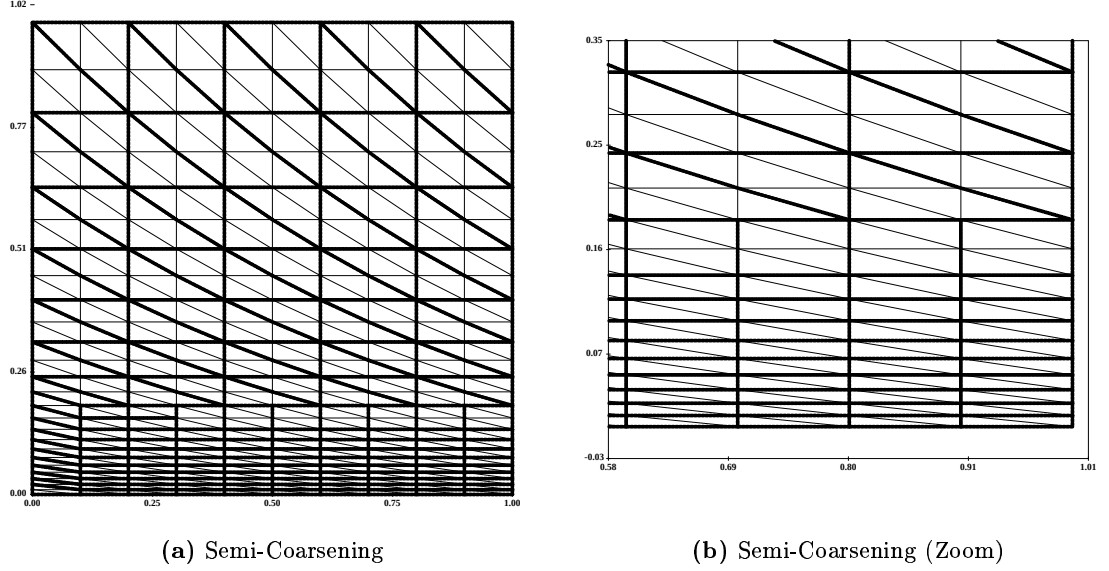


Figure 2.8: Semi-coarsening in anisotropic mesh region

of the coarsening technique is shown in Fig. 2.8. Morano *et al* [72] present experimental results for the Laplace equation for several anisotropic cases with high aspect ratio cells when semi-coarsening is applied.

2.7 Results

We consider the Poisson problem (Eq. 2.1) where

$$\nabla^2 u = -1 \quad \text{in } \Omega =]0, 1[^2, \quad (2.59)$$

$$u = 0 \quad \text{on } \Gamma_D, \quad (2.60)$$

The discretization is performed by triangulating the square grid ($\Omega =]0, 1[^2$) with a set of N ($= n \times n$) uniformly distributed points. We use three different fine grid sizes and compare the multigrid convergence rate for a two grid agglomeration problem. In all cases, we use a V(2,1) cycle and terminate the algorithm when the L_2 norm of the residual is less than 10^{-17} . Table 2.2 depicts the obtained convergence rates for two different smoothers.

The average multigrid convergence rate is defined as

$$\bar{\epsilon} = \left(\frac{\|\mathbf{r}^n\|}{\|\mathbf{r}^0\|} \right)^{\frac{1}{n}} \quad (2.61)$$

where n is the number of multigrid cycles while the asymptotic convergence rate ϵ is defined as the average convergence rate computed over the last 5 multigrid cycles.

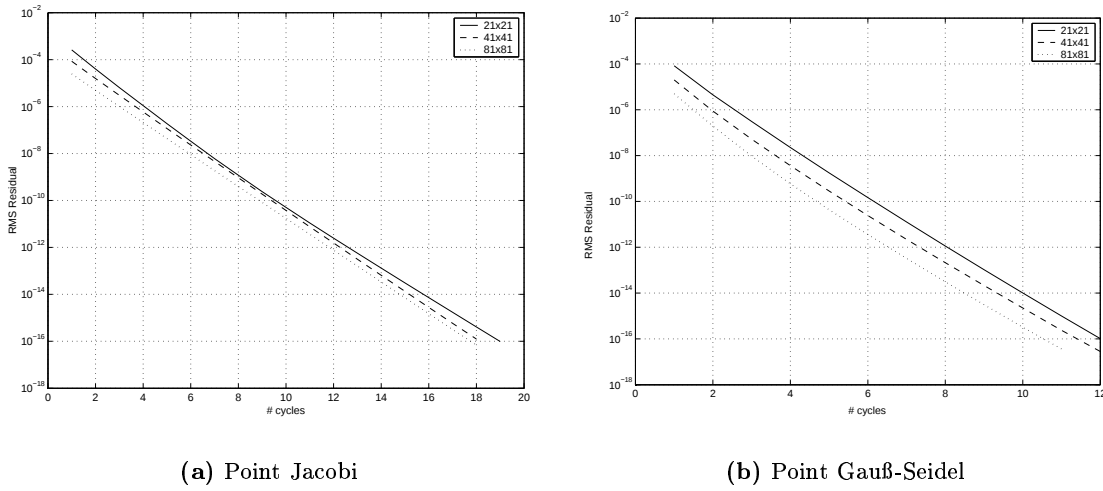


Figure 2.9: Agglomeration Multigrid Convergence History for Poisson Problem: Isotropic Mesh

	Smoother	ω	Average Convergence $\bar{\epsilon}$	Asymptotic Convergence ϵ
21x21	Jacobi	0.8	0.194	0.236
	Gauß-Seidel	1.0	0.073	0.096
41x41	Jacobi	0.8	0.197	0.210
	Gauß-Seidel	1.0	0.075	0.105
81x81	Jacobi	0.8	0.206	0.211
	Gauß-Seidel	1.0	0.071	0.100

Table 2.2: Agglomeration Multigrid Results for Poisson Problem: Isotropic Mesh

The ordering of the nodes for both smoothers as well as the sweep pattern for the Gauß-Seidel smoother is such that for a given node at $\{(i, j) : \{1..n_x\} \times \{1..n_y\}\}$, the lexicographical ordering is given by $\{\text{ord}(i, j) = j + i \times n_y\}$ i.e sweeps in the y-direction

with increasing x-direction. The numerical results in Fig. 2.9 and Table 2.2 show that the proposed AMG convergence rate is uniform with respect to h and also satisfies the Gauß-Seidel theoretical convergence rate of $\bar{\epsilon} \leq 0.1$ [16] for the Poisson problem discretized on isotropic meshes.

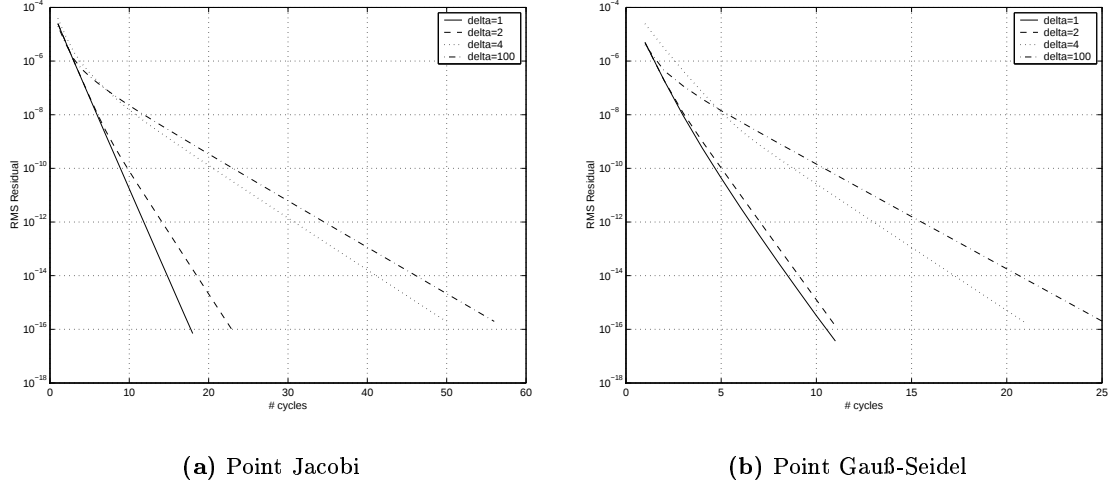


Figure 2.10: Agglomeration Multigrid Convergence History for Poisson Problem: Anisotropic Mesh

δ_s	Smoother	ω	ϵ w/o semi-coarsening	ϵ w/ semi-coarsening
1	Jacobi	0.8	0.211	0.211
	Gauß-Seidel	1.0	0.100	0.100
2	Jacobi	0.8	0.352	0.315
	Gauß-Seidel	1.0	0.105	0.100
4	Jacobi	0.8	0.647	0.642
	Gauß-Seidel	1.0	0.345	0.339
100	Jacobi	0.8	0.979	0.672
	Gauß-Seidel	1.0	0.949	0.408

Table 2.3: Agglomeration Multigrid Results for Poisson Problem: Anisotropic Mesh

Anisotropic results based on grid stretching are shown in Fig. 2.10 and Table 2.3 for the same domain $\Omega =]0, 1[^2$. Comparisons are made for the multigrid convergence rate with and without semi-coarsening. The stretched grid is created by uniformly generating

n_x points in the x-direction with spacing Δx . The points in the y-direction are generated using a geometric growth algorithm with an initial spacing of $\frac{\Delta x}{\delta_s}$, where δ_s is the initial aspect ratio, and a growth factor of 1.1 until $\Delta y \approx \Delta x$, after which uniform spacing is used as shown in Fig. 2.8. For the test cases, n_x was chosen to be 81 and a comparison of the convergence rate for $\delta_s = 1, 2, 4$ and 100 is made.

The results show an appreciable degradation in the multigrid convergence rates for these point implicit schemes even with the inclusion of semi-coarsening. The ideal solution to obtain better convergence rates would be use a line implicit smoother which will be introduced in the next chapter.

Chapter 3

AMG: Convection-Diffusion Operators

The direct extension of standard multigrid algorithms as well as the algorithms described in Chap. 2 to convection-diffusion operators has been shown to result in a degradation in the convergence performance [46, 76]. This is especially true for convection dominated operators which introduce features that violate certain assumptions these algorithms are based on.

The classic implementation of AMG by Ruge and Stüben [44] assumes that the underlying matrix belongs to the class of M-matrices that are characterized by diagonal dominance as well as symmetry which may no longer hold in the presence of strong convection. Chang *et al* [45] have extended the method of Ruge and Stüben to include generalized matrices. However, some of the results shown for Finite Difference approximations to the convection-diffusion equation and the anisotropic diffusion equation show that mesh independent convergence rates are not obtained (in some cases, the algorithm actually diverged).

Convergence analysis for smoothers [46, 88–90] shows that point iterative schemes typically have poor convergence properties when applied to Finite Difference approximations of strongly convective equations. The convergence rate may be improved by the addition of damping, however for some iterative schemes such as point Jacobi, no amount of damping will make it convergent

This chapter focuses on application of the multigrid algorithm to the linear convection-diffusion equation, especially in the high Peclet number limit. The linear convection-diffusion equation represents a simplification of the target objective of this thesis which is the multigrid solution of the high Reynolds number Navier-Stokes equations. In this chapter we consider the modification of the proposed AMG algorithm to a stabilized Fi-

nite Element discretization of the linear convection-diffusion equation. The stability of the linear iterative smoothers, when applied to strongly convective flows, is considered and is modified with the implementation of a multistage scheme. The poor convergence rates of the point implicit smoothers for strongly convective flows eventually results in the choice of a line implicit smoother. Finally, the application of multigrid to stabilized numerical schemes raises several issues regarding the representation of the stabilization and boundary condition terms on the coarse spaces. The modified multigrid algorithm is applied to stabilized Finite Element discretizations of convection-diffusion problems and the results are analyzed.

3.1 Model Problem

The model problem considered is the two dimensional stationary linear convection diffusion problem represented by the differential equation:

$$\nabla \cdot (\vec{V}\Phi) - \nabla \cdot (\mu(x, y)\nabla\Phi) = f \quad \text{in } \Omega, \quad (3.1)$$

where Ω is a bounded domain in $\mathbb{R}^2$ with boundary Γ which is made up of a Dirichlet boundary Γ_D and a Neumann boundary $\Gamma_N = \Gamma \setminus \Gamma_D$. $\vec{V} = (u, v)$ is a prescribed divergence free velocity field, the coefficient $\mu(x, y)$ is a strictly positive diffusivity (or viscosity) coefficient and f is a source function. Let us consider the partition of Γ into $\{\Gamma^-, \Gamma^+\}$ where

$$\Gamma^- = \{(x, y) \in \Gamma : \vec{V} \cdot \mathbf{n} < 0\} \text{ (inflow boundary)} \quad (3.2)$$

$$\Gamma^+ = \Gamma \setminus \Gamma^- \quad \text{(outflow boundary)} \quad (3.3)$$

and also consider the following boundary subsets:

$$\Gamma_D^\pm = \Gamma_D \cap \Gamma^\pm \quad (3.4)$$

$$\Gamma_N^\pm = \Gamma_N \cap \Gamma^\pm \quad (3.5)$$

We simplify the boundary conditions by assuming that $\Gamma^- \subset \Gamma_D$ such that Eq. 3.1 is subject to the boundary conditions:

$$\Phi = g_D \quad \text{on } \Gamma_D, \quad (3.6)$$

$$(-\Phi \vec{\mathbf{V}} + \mu \nabla \Phi) \cdot \mathbf{n} = g_N \quad \text{on } \Gamma_N^+, \quad (3.7)$$

The behavior of Eq. 3.1 is governed by the Peclet number defined by:

$$\text{Pe} = \frac{UL}{\mu} \quad (3.8)$$

which represents the relative importance of advection to diffusion where U is some reference speed and L is a characteristic length of the problem. In the advective limit $\text{Pe} \rightarrow \infty$, Eq. 3.1 is a hyperbolic equation which is characterized by the transport of information along characteristic lines. In this case, the solution may be discontinuous with jumps across the characteristic lines. Also, boundary conditions may only be specified on the inflow boundary such that the problem reduces to:

$$\nabla \cdot (\vec{\mathbf{V}} \Phi) = f \quad \text{in } \Omega, \quad (3.9)$$

$$\Phi = g_D \quad \text{on } \Gamma_D^-, \quad (3.10)$$

Hence, discontinuous jumps in the solution may occur if the boundary data g_D is discontinuous. The introduction of diffusion results in the solution being continuous over the entire domain such that any jumps will spread out around the characteristic line over a region of width $O(1/\sqrt{\text{Pe}})$. It is well known that the application of the classical Galerkin Finite Element method to the advection-dominated convection-diffusion equation lacks stability [91]. This results in solutions which are polluted with spurious oscillations due to unresolved internal and boundary layers. Oscillation-free solutions may be obtained by the application of upwind differencing to the convective terms, however upwind differences are first order and produce overly diffusive solutions. Hence, we consider the Streamline Upwind/Petrov Galerkin (SUPG) method introduced by Brooks and Hughes [92] which is a stabilized Finite Element formulation that is a consistent weighted residual method. This method attempts to introduce upwinding with no crosswind diffusion while maintaining a higher order discretization [93, 94]. A generalization of the SUPG method is the Galerkin/Least Squares

(GLS) method [95] which is equivalent to the SUPG method for purely hyperbolic operators and/or for piecewise linear elements. Though quite similar to SUPG, the analysis of GLS is simpler.

Introducing the variational form of Eq. 3.1, the discrete problem reduces to finding $\Phi_h \in \mathcal{S}^h$ such that:

$$B(\Phi_h, w_h)_{gal} + B(\Phi_h, w_h)_{gl s} + B(\Phi_h, w_h)_{bc} = 0, \quad \forall w_h \in \mathcal{V}^h \quad (3.11)$$

where the forms $B(\cdot, \cdot)_{gal}$, $B(\cdot, \cdot)_{gl s}$ and $B(\cdot, \cdot)_{bc}$ account for the Galerkin, GLS stabilization, and boundary condition terms respectively. These are defined as

$$\begin{aligned} B(\Phi_h, w_h)_{gal} &= \int_{\Omega} \left(-\Phi_h \vec{\mathbf{V}} \cdot \nabla w_h + \mu \nabla \Phi_h \cdot \nabla w_h - w_h f \right) d\Omega \\ B(\Phi_h, w_h)_{gl s} &= \sum_{e=1}^{n_{el}} \int_{\Omega^e} \left(\vec{\mathbf{V}} \cdot \nabla w_h - \mu \nabla^2 w_h - f \right) \tau_e \cdot \left(\vec{\mathbf{V}} \cdot \nabla \Phi_h - \mu \nabla^2 \Phi_h - f \right) d\Omega \\ B(\Phi_h, w_h)_{bc} &= \int_{\Gamma_N} w_h (\Phi_h \vec{\mathbf{V}}) \cdot \mathbf{n} d\Gamma - \int_{\Gamma_N} w_h (\mu \nabla \Phi_h) \cdot \mathbf{n} d\Gamma = \int_{\Gamma_N} w_h g_N d\Gamma \end{aligned}$$

where the coefficient τ_e is the GLS stabilization parameter which is positive and represents an intrinsic time scale. Local element Peclet number (Pe_h) and length scale (h_e) dependency for τ_e [96] is important in order to provide accuracy and proper stability especially in problems where severe cell stretching is required for computational efficiency in the resolution of internal/boundary layers. The local element Peclet number Pe_h is given by

$$Pe_h = \frac{\|\mathbf{V}\| h_e}{\mu} \quad (3.12)$$

Carette [97] gives a review of several designs for the definition of the elemental length scale h_e for linear triangular elements. In general, these definitions fall into two broad classes which are geometric and projection based methods. The geometric methods consider the geometry of the element and try to determine some suitable length scale for h_e while projection methods consider the projected length of the triangle onto a line parallel to the local velocity field. We make use of the definition by Mizukami [98] which corresponds to the maximum of the intersection between the triangle and a line parallel to the local velocity field as shown in Fig. 3.1.

This figure was broken... original no longer exists.

Figure 3.1: Illustration of Linear Triangle Length Scale h_e

The stability parameter needs to be selected to satisfy the following conditions [97]:

$$\tau_e = \mathcal{O}\left(\frac{h_e}{\|\mathbf{V}\|}\right) \quad \text{Pe}_h \gg 1 \quad (3.13)$$

$$\tau_e = \mathcal{O}\left(\frac{h_e^2}{\mu}\right) \quad \text{Pe}_h \ll 1 \quad (3.14)$$

For the scalar convection-diffusion equation, the definition for τ_e has usually been done in a heuristic fashion such that τ_e takes the general form:

$$\tau_e = \tau_e^{\text{conv}} \bar{\zeta}(\text{Pe}_h) \quad (3.15)$$

where τ_e^{conv} is the convective limit for τ_e given by Eq. 3.13 and $\bar{\zeta}$ is a function of the local Peclet number Pe_h . In order for τ_e to satisfy the conditions laid out by Eq. 3.13 and Eq. 3.14, the asymptotic behavior for τ_e is governed by:

$$\bar{\zeta} \rightarrow 1 \quad \text{as } \text{Pe}_h \rightarrow \infty \quad (3.16)$$

$$\bar{\zeta} \rightarrow C \text{Pe}_h \quad \text{as } \text{Pe}_h \rightarrow 0 \quad (3.17)$$

where C is a constant. In the absence of convection, $\text{Pe}_h = 0$ which shows that stabilization is not required for purely diffusive flows. Analysis of the GLS discretization of the one dimensional linear scalar convection-diffusion model problem with constant flow, diffusivity, absence of source and uniform mesh using piecewise linear elements [99], shows that nodally

exact solutions may be obtained with an optimal definition for τ_e :

$$\tau_{opt} = \frac{h_e}{2U} \bar{\zeta}_{opt}(\alpha) \quad (3.18)$$

$$\bar{\zeta}_{opt}(\alpha) = \coth(\alpha) - \frac{1}{\alpha} \quad (3.19)$$

$$\alpha = \frac{h_e U}{2\mu} \quad (3.20)$$

where α is the local element Peclet number. Following this example, Shakib [96] proposes some simpler forms for τ_e . We make use of one such form such that:

$$\bar{\zeta}(\text{Pe}_h) = \frac{\text{Pe}_h}{\sqrt{9 + (\text{Pe}_h)^2}} \quad (3.21)$$

The construction of τ_e now follows. For a triangle T_e as shown in Fig. 3.2, we introduce the affine mapping $\mathbf{x} = \mathbf{x}(\xi)$, between the master triangle $\hat{T}_e$, in the parametric space $\boldsymbol{\xi} = [\xi, \eta]^T$, and $T_e \in \Omega$ in the mapped space $\mathbf{x} = [x, y]^T$. The explicit form for the affine mapping is given by:

$$\begin{Bmatrix} \xi \\ \eta \end{Bmatrix} = \frac{1}{2A_e} \begin{Bmatrix} (x_2 y_3 - x_3 y_2) \\ (x_3 y_1 - x_1 y_3) \end{Bmatrix} + \frac{1}{2A_e} \begin{bmatrix} (y_2 - y_3) & (x_3 - x_2) \\ (y_3 - y_1) & (x_1 - x_2) \end{bmatrix} \begin{Bmatrix} x \\ y \end{Bmatrix} \quad (3.22)$$

where A_e is the element area given by

$$A_e = \frac{1}{2} \begin{vmatrix} 0 & 0 & 1 \\ x_2 - x_1 & y_2 - y_1 & 0 \\ x_3 - x_1 & y_3 - y_1 & 0 \end{vmatrix} \quad (3.23)$$

Let

$$\bar{u} = \frac{u_1 + u_2 + u_3}{3} \quad \bar{v} = \frac{v_1 + v_2 + v_3}{3}$$

This figure was broken... original no longer exists.

Figure 3.2: Master Element and Mapping Function

$$\begin{aligned}
 \mathcal{J} &= x_{,\xi} y_{,\eta} - y_{,\xi} x_{,\eta} \\
 \mathcal{B}_1 &= (y_{,\eta} \bar{u} - x_{,\eta} \bar{v}) / \mathcal{J}, \\
 \mathcal{B}_2 &= (x_{,\xi} \bar{v} - y_{,\xi} \bar{u}) / \mathcal{J}, \\
 \mathcal{B}_3 &= -\mathcal{B}_1 - \mathcal{B}_2
 \end{aligned}$$

where $(\bar{u}, \bar{v})$ is the average of the velocity defined at the three element vertices and $\mathcal{J}$ is the Jacobian of the reverse mapping. The parameter τ_e is now defined by

$$\begin{aligned}
 D(\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3) &= \frac{3}{(|\mathcal{B}_1| + |\mathcal{B}_2| + |\mathcal{B}_3|)} \\
 \bar{\mathbf{U}} &= \frac{\mathbf{U}_1 + \mathbf{U}_2 + \mathbf{U}_3}{3} \\
 \text{Pe}_h &= (\bar{\mathbf{U}} \cdot \bar{\mathbf{U}}) \frac{D}{\mu} \tag{3.24}
 \end{aligned}$$

$$\tau_e = D \frac{\text{Pe}_h}{\sqrt{9 + \text{Pe}_h^2}} \tag{3.25}$$

For the choice of the nodal basis functions and linear $\mathcal{P}_1$ interpolation polynomials in $\mathcal{S}^h$ and $\mathcal{V}^h$, Eq. 3.11 represents a sparse linear system of equations which can also be written simply as

$$\mathbf{A}_h \Phi_h = b_h$$

3.2 Multistaging

The application of linear iterative schemes such as point Jacobi to discretizations of hyperbolic or almost-hyperbolic partial differential equations has been known to fail due to the lack of stability of the smoothing algorithm. Consider the basic iterative scheme (as outlined in Chap. 2) defined by Eq. 3.26:

$$\Phi_h^{n+1} = \Phi_h^n + \mathbf{M}_h^{-1} (b_h - \mathbf{A}_h \Phi_h^n) \quad (3.26)$$

where Φ_h^n is the current solution estimate at iteration n . Application of Theorem 2.1 to the scheme implies that the eigenvalues of the iteration matrix $-\mathbf{M}_h^{-1} \mathbf{A}_h$ have to lie in a region in the complex plane defined by

$$|z + 1| < 1 \quad (3.27)$$

Wesseling [46] has shown that the application of the basic iterative scheme given by Eq. 3.26, to an upwind Finite Difference approximation of the pure convection problem results in a non-robust smoother for the point Jacobi and point Gauß-Seidel schemes. This is a well known problem especially regarding the application of these point relaxation schemes to higher order discretizations which causes eigenvalues of the iteration matrix to fall outside the stability region of the basic iterative scheme. A possible solution for this is to increase the stability region of the smoother by the introduction of a multistaging scheme. Multistage methods were developed by Jameson [24] for the solution of the Euler equations which is a hyperbolic system of equations and have been applied with great success to a variety of applications [24, 100, 101]. An N-stage multistage formulation of the basic iterative scheme may be implemented as follows. Let:

$$\begin{aligned} \Phi_h^{(0)} &= \Phi_h^n \\ \Phi_h^{(1)} &= \Phi_h^{(0)} + \alpha_1 \mathbf{M}_h^{-1} (b_h - \mathbf{A}_h \Phi_h^{(0)}) \\ &\vdots \\ \Phi_h^{(i+1)} &= \Phi_h^{(0)} + \alpha_{i+1} \mathbf{M}_h^{-1} (b_h - \mathbf{A}_h \Phi_h^{(i)}) \\ &\vdots \\ \Phi_h^{n+1} &= \Phi_h^{(N)} \end{aligned} \quad (3.28)$$

where α_i are the multistage coefficients. Standard Runge-Kutta definitions give

$$\alpha_i = \frac{1}{N - i} \quad (3.29)$$

although optimized coefficients may be found [25, 101–104]. For the rest of this thesis, the optimized multistage coefficients employed are those by Lynn [104] for Roe’s $\kappa = 0$ scheme. These coefficients have been optimized for the damping of the high frequency modes for locally preconditioned discretizations of the full Euler or Navier-Stokes spatial operator. Table 3.1 shows the numerical values employed for the 3-stage and 5-stage schemes for both full and semi-coarsening.

	3 stage (full)	3 stage (semi)	5 stage (full)	5 stage (semi)
α_1	0.2075	0.2239	0.09621	0.08699
α_2	0.5915	0.5653	0.2073	0.1892
α_3	1	1	0.3549	0.3263
α_4			0.6223	0.5558
α_5			1	1

Table 3.1: Optimized multistage coefficients for full and semi-coarsening by Lynn [104]

Given one step with the full scheme described by Eq. 3.28, the multistage error amplification factor ψ_N which is equivalent to Eq. 2.26 may be defined by:

$$e_h^{n+1} = \psi_N(\mathbf{z}) e_h^n \quad (3.30)$$

where e_h^n is the solution error defined by Eq. 2.14, $\mathbf{z} = -\mathbf{M}_h^{-1} \mathbf{A}_h$ and ψ_N is the polynomial matrix defined by:

$$\psi_N(\mathbf{z}) = \mathbf{I} + \sum_{j=1}^N c_j \mathbf{z}^j \quad (3.31)$$

$$c_j = \prod_{s=N+1-j}^N \alpha_s \quad (3.32)$$

Hence, for the multistage formulation to be convergent:

$$\lim_{n \rightarrow \infty} \|\psi_N^n(\mathbf{z})\| = 0 \quad (3.33)$$

Theorem 3.1 *The multistage method Eq. 3.28 will converge for any initial guess Φ_h^0 iff*

$$\rho(\psi_N(\mathbf{z})) < 1 \quad (3.34)$$

Theorem 3.1 implies that the spectral radius of the matrix $\mathbf{z}$ must lie within the stability region defined by $|\psi_N(z)| \leq 1$. Fig. 3.3 shows the stability regions for the optimized multistage schemes as well as standard Runge-Kutta multistage schemes for comparison.

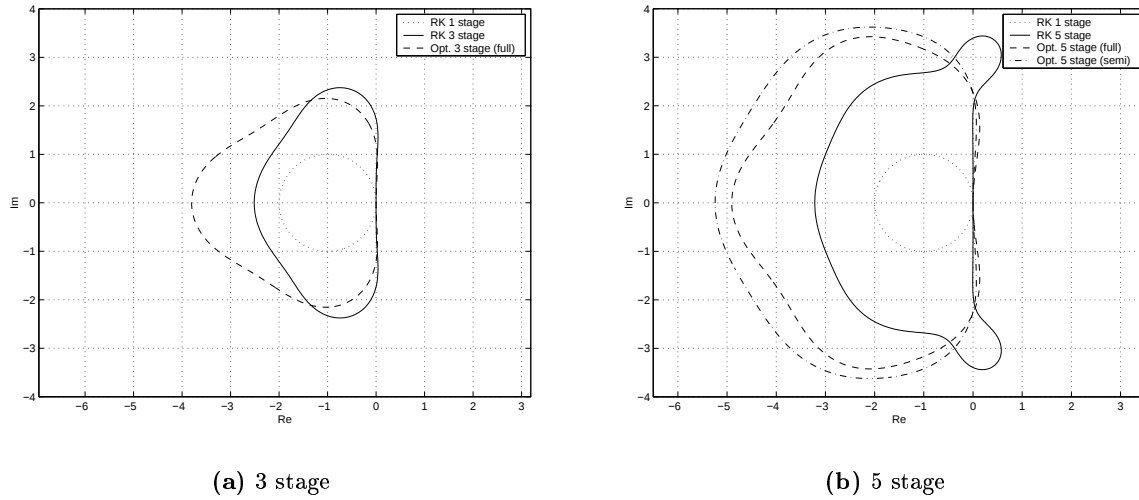


Figure 3.3: Stability Contours

3.3 Multigrid Smoother for Convection-Diffusion Operators

The efficiency of the multigrid algorithm lies in the synergy of the various multigrid components, which can be encapsulated in the smoother and the coarse spaces. The decomposition of the error is made with the idea that the smoother on each space is responsible for the elimination of a portion of the error. Convergence analysis of smoothers such as point Ja-

cobi which possess excellent damping properties for elliptic operators has been shown to exhibit poor convergence rates when applied to strongly convective systems [46, 88]. These systems are characterized by the transport of information along characteristic directions which causes the error modes to decouple into preferential directions. Point relaxation schemes result in preferential error smoothing along these directions [49] with a subsequent deterioration in the multigrid convergence rate.

We propose a solution to the outlined issues by a modification to the algorithms given in Chap. 2. In highly stretched grid regions and strongly convective regions, the convection-diffusion model problem is characterized by strong alignment. Hence, we opt to use an implicit line relaxation scheme as the smoother where the implicit lines are constructed to follow directions of strong influence. In strongly convective problems, the directions of strong influence align with the characteristic directions. This can lead to exact solvers under the right conditions due to the propagation of advected information along these characteristic directions.

The line smoother developed here for unstructured grids is similar to the geometry based line implicit scheme described by Mavriplis [6, 71, 105]. In Mavriplis' scheme, the smoother is a line implicit smoother in regions of highly stretched elements and in the isotropic regions of the grid, the smoother reverts to a point implicit smoother. While this represents an improvement, the use of a point implicit solver in the isotropic grid region which is convection dominated does not fully address the multigrid convergence rate issues. The proposed line implicit smoother is designed to take into account, the error directional decoupling issues induced by anisotropic meshes as well as the hyperbolic characteristics of the governing equation.

Wesseling [46] gives a Fourier analysis for some line implicit smoothers applied to Finite Difference approximations of the linear convection-diffusion equation and shows that these are superior smoothers to point implicit schemes. Fig. 3.4 shows an example of the implicit line construction for a 2D GLS discretization of the linear convection-diffusion equation with an imposed velocity field of $\vec{V} = (-y, x)$.

The use of a line relaxation scheme leads to a natural splitting of the matrix into tridiagonal sub-matrices which may be solved in $O(N)$ time. The elemental agglomeration procedure as well as the construction of the interpolation operators and the coarse space operators remain as previously described.

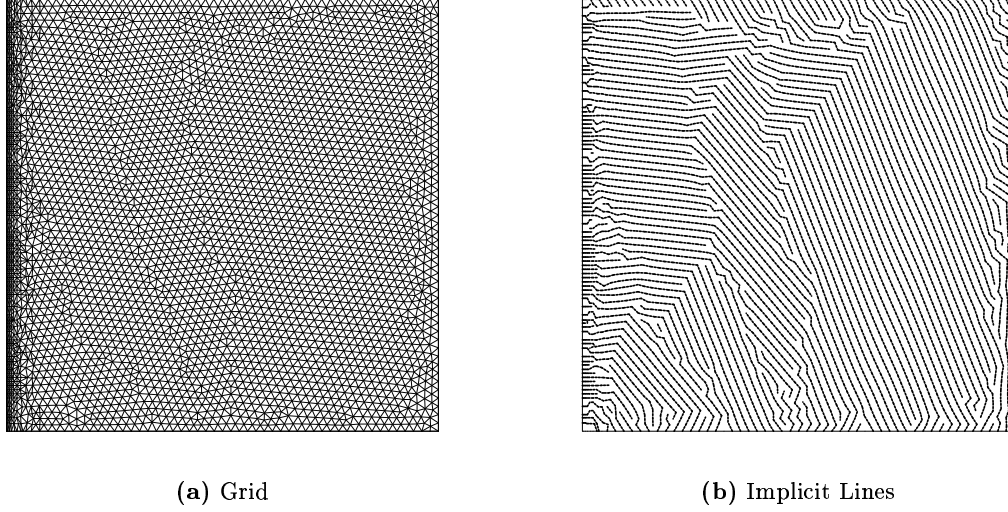


Figure 3.4: Implicit Line System

3.4 Implicit Line Construction

The implicit line construction process is based on the idea of linking strongly coupled nodes. To measure nodal coupling, we use a matrix based concept which we term *coupling measure*.

Definition 4 *The coupling measure between two connected vertices is a local quantification of the connectivity/influence between these vertices.*

Typically, this is based on the matrix stencil but other measures such as a projection of the flow velocity on the mesh graph edges or streamlines may be used. This becomes even more complicated in the case of block systems of equations which arise in discretizations of the Navier-Stokes equations. In any case, given some local quantification of the local coupling between any two nodes, we may construct a coupling matrix. For the model problem, we simply choose the discrete matrix in Eq. 3.11 as the coupling matrix. Based on the scalar coupling matrix $(c_{i,j})$ chosen, let the set of points, denoted by S_i , connected to a node be

$$S_i \equiv \{j \neq i : c_{i,j} \neq 0\} \quad (3.35)$$

We define the coupling measure between any two connected vertices (i, j) by

$$\begin{aligned}\beta_i &= \frac{|c_{i,j}|}{\max_k |c_{i,k}|} & \{k : k \in S_i\} \\ \beta_j &= \frac{|c_{j,i}|}{\max_k |c_{j,k}|} & \{k : k \in S_j\} \\ \text{coup}(i, j) &= \max(\beta_i, \beta_j)\end{aligned}\tag{3.36}$$

In convection dominated flows or upwind-type methods, strong coupling tends to be one-sided which necessitates the two way consideration of Eq. 3.36. We refer to [46, 106] for other methods regarding the detection of strong coupling.

Line construction is done in a two pass process which involves the construction of individual lines and then a merging of lines to reduce the line count. The construction of a line begins by choosing a seed node. For a given line, it should ideally be such that any member node of the line can be chosen as a seed, such that tracing out the line from the seed node in both directions creates the line. This leads to the concept of forward and backward mode line construction. In order to describe the algorithm properly, we require one more definition which we term *line tridiagonality*.

Definition 5 *Line tridiagonality is a matrix based connectivity property. A line which has no nodes that are coupled to more than two other nodes in the line is said to satisfy this property. This ensures that the sub-matrix associated with the line is a true tridiagonal matrix.*

In the construction of the implicit lines, overlap of the lines is permitted. We now present the algorithm in detail:

Procedure 4 (*Forward Mode Line Construction*)

Step 0: *Set overlap counter to zero, create tag vector for the nodes and untag all nodes.*

Step 1: *Tag the current node and compute coupling measure for all the nodes connected to the current node.*

Step 2: *Compute a coupling measure threshold value which is defined as a fraction (typically 0.75) of the maximum coupling measure of all coupled nodes.*

Step 3: *Scan the coupled nodes and choose the one which*

- (a) Has the largest coupling measure greater than the threshold **and**
- (b) Is untagged **and**
- (c) Is not already in the current line **and**
- (d) Preserves tridiagonality.

Step 4: *If the chosen node already exists in another line and is not an extremity of that line, increment overlap counter. If counter reaches predetermined limit, terminate line.*

Step 5: *If the chosen node already exists in another line and is an extremity of that line, terminate the algorithm and merge the current line with the other line only if tridiagonality is preserved.*

Step 6: *Repeat Step 1 with the chosen node as the current node until no nodes satisfy the criteria in Step 3 or Step 4 or 5 is triggered.*

The forward mode line construction acts as an integral part of the backward mode line construction. The algorithm for the backward mode line construction is now presented:

Procedure 5 (*Backward Mode Line Construction*)

Step 0: *Create tag vector for the nodes and untag all nodes.*

Step 1: *Tag the current node and for each untagged node connected to the current node, perform one iteration of the forward mode algorithm.*

Step 2: *From all the coupled nodes which would have chosen the current node as the next node in Step 1, pick the one with the strongest coupling. Repeat Step 1 with the chosen node as the current node until no nodes satisfy the criterion in Step 2.*

Using these two components of line construction, the full algorithm for the line construction is now given by:

Procedure 6 (*Implicit Line Construction*)

Step 1: *Obtain seed node which does not belong to any line.*

Step 2: *Using seed node, perform partial line construction using the forward and backward mode procedures and merge the two pieces into one contiguous line.*

Step 3: Repeat **Step 1** until there are no more seed nodes.

Step 4: Perform line merging. Consider the node extremities of each line and scan for all coupled nodes to these extreme nodes. If any of these coupled nodes is also an extremity of another line, compute the coupling measure between these two nodes. If the coupling measure is greater than a threshold value, merge the two lines provided that tridiagonality is preserved.

The overlap between the lines is kept to a minimum for two reasons. The first is to reduce the amount of work by the relaxation. The second is to prevent possible amplification of local modes due to solving a vertex equation multiple times. In order to alleviate such amplifications, we can parametrically control the amount of overlap in the line construction. However for the rest of the thesis, all results presented will be based on no overlap.

3.5 Fourier Analysis of Implicit Line Smoother

A simplified Fourier stability analysis of the implicit line smoother as applied to a convective-diffusive model problem is now presented. We consider the linear scalar convection-diffusion model (Eq. 3.37) with no body force, constant diffusivity and an imposed velocity field of $\vec{V} = (V \cos \Lambda, V \sin \Lambda)$. The discretization is done using the Galerkin Least Squares (GLS) formulation on the grid shown in Fig. 3.5.

$$\vec{V} \cdot \nabla u = \mu \nabla^2 u \quad (3.37)$$

We now consider periodic boundary conditions

$$\begin{aligned} u(L_x, y) &= u(0, y) \\ u(x, L_y) &= u(x, 0) \end{aligned}$$

which enables us to perform a Fourier decomposition of the exact solution. The GLS

This figure was broken... original no longer exists.

Figure 3.5: Scalar convection-diffusion model domain

variational form is given by:

$$\begin{aligned} \sum_{e=1}^{n_{el}} \int_{\Omega^e} \left(V \left[\cos \Lambda w_x^h + \sin \Lambda w_y^h \right] - \mu \left[w_{xx}^h + w_{yy}^h \right] \right) \tau \left(V \left[\cos \Lambda u_x^h + \sin \Lambda u_y^h \right] - \mu \left[u_{xx}^h + u_{yy}^h \right] \right) d\Omega \\ + \int_{\Omega} \left(w^h V \left[\cos \Lambda u_x^h + \sin \Lambda u_y^h \right] + \mu \left[w_x^h u_x^h + w_y^h u_y^h \right] \right) d\Omega = 0 \quad (3.38) \end{aligned}$$

where

$$u^h(x, y) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} N_{ij}(x, y) u_{ij}^h \quad (3.39)$$

$$w^h(x, y) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} N_{ij}(x, y) w_{ij}^h \quad (3.40)$$

We consider the shape functions $N_{ij}(x, y)$ to be piecewise bilinear interpolants in x and y such that

$$N_{ij}(x, y) = F(x)G(y)$$

For the stencil shown in Fig. 3.6,

$$F(x) = \begin{cases} \frac{x-x_{i-1}}{\Delta x} & \text{Regions 1 \& 3} \\ \frac{x_{i+1}-x}{\Delta x} & \text{Regions 2 \& 4} \end{cases} \quad (3.41)$$

$$G(y) = \begin{cases} \frac{y-y_{j-1}}{\Delta y} & \text{Regions 1 \& 2} \\ \frac{y_{j+1}-y}{\Delta y} & \text{Regions 3 \& 4} \end{cases} \quad (3.42)$$

This figure was broken... original no longer exists.

Figure 3.6: Computational 9 point stencil for Fourier Analysis of Implicit Line Smoother

Substitution into Eq. 3.38 yields

$$\begin{aligned} 0 &= \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} w_{ij}^h \left\{ \int_{\Omega} \sum_{m=1}^{n_x} \sum_{n=1}^{n_y} \left(V N_{ij} \left[\cos \Lambda \frac{\partial N_{mn}}{\partial x} + \sin \Lambda \frac{\partial N_{mn}}{\partial y} \right] \right) u_{mn}^h d\Omega \right\} \\ &+ \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} w_{ij}^h \left\{ \int_{\Omega} \sum_{m=1}^{n_x} \sum_{n=1}^{n_y} \left([\mu + \tau V^2 \cos^2 \Lambda] \frac{\partial N_{ij}}{\partial x} \frac{\partial N_{mn}}{\partial x} \right) u_{mn}^h d\Omega \right\} \\ &+ \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} w_{ij}^h \left\{ \int_{\Omega} \sum_{m=1}^{n_x} \sum_{n=1}^{n_y} \left([\mu + \tau V^2 \sin^2 \Lambda] \frac{\partial N_{ij}}{\partial y} \frac{\partial N_{mn}}{\partial y} \right) u_{mn}^h d\Omega \right\} \\ &+ \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} w_{ij}^h \left\{ \int_{\Omega} \sum_{m=1}^{n_x} \sum_{n=1}^{n_y} \left([\tau V^2 \sin \Lambda \cos \Lambda] \left(\frac{\partial N_{ij}}{\partial x} \frac{\partial N_{mn}}{\partial y} + \frac{\partial N_{ij}}{\partial y} \frac{\partial N_{mn}}{\partial x} \right) \right) u_{mn}^h d\Omega \right\} \end{aligned} \quad (3.43)$$

This is of the form

$$\sum_{i=1}^{n_x} \sum_{j=1}^{n_y} w_{ij}^h G_{ij}(\tilde{\mathbf{u}}^h) = 0 \quad \tilde{\mathbf{u}}^h = \left\{ u_{ij}^h : i = 1, \dots, n_x; j = 1, \dots, n_y \right\}$$

$$\Rightarrow \quad G_{ij}(\tilde{\mathbf{u}}^h) = 0 \quad \forall w_{ij}^h \in \mathbb{R} \quad \{i = 1, \dots, n_x - 1; j = 1, \dots, n_y - 1\}$$

Evaluation of the integrals lead to the 9-point stencil matrix

$$[\mathbf{A}] = q_1 [\mathbf{Q}] + r_1 [\mathbf{R}] + q_2 [\mathbf{Q}]^T + r_2 [\mathbf{R}]^T + z_1 [\mathbf{Z}] + z_2 [\mathbf{Z}]^T \quad (3.44)$$

where

$$q_1 = \frac{V \Delta y \cos \Lambda}{12} \quad (3.45)$$

$$q_2 = \frac{V \Delta x \sin \Lambda}{12} \quad (3.46)$$

$$r_1 = \frac{\Delta y}{6 \Delta x} (\mu + \tau V^2 \cos^2 \Lambda) \quad (3.47)$$

$$r_2 = \frac{\Delta x}{6 \Delta y} (\mu + \tau V^2 \sin^2 \Lambda) \quad (3.48)$$

$$z_1 = \frac{\tau V^2 \sin \Lambda \cos \Lambda}{4} \quad (3.49)$$

$$z_2 = \frac{\tau V^2 \sin \Lambda \cos \Lambda}{4} \quad (3.50)$$

$$[\mathbf{Q}] = \begin{bmatrix} -1 & 0 & 1 \\ -4 & 0 & 4 \\ -1 & 0 & 1 \end{bmatrix} \quad (3.51)$$

$$[\mathbf{R}] = \begin{bmatrix} -1 & 2 & -1 \\ -4 & 8 & -4 \\ -1 & 2 & -1 \end{bmatrix} \quad (3.52)$$

$$[\mathbf{Z}] = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \quad (3.53)$$

We now extract three non-dimensional groups from Equations 3.45 - 3.48

$$\delta = \frac{\Delta y}{\Delta x} \quad (3.54)$$

$$\aleph = \frac{V\tau}{\Delta x} \quad (3.55)$$

$$\text{Pe}_h = \frac{V\Delta x}{\mu} \quad (3.56)$$

where Eq. 3.54 represents the mesh aspect ratio, Eq. 3.55 gives a measure of the convection/diffusion dominance i.e $\aleph \rightarrow 0$ in the pure diffusion limit and $\aleph \rightarrow 1$ in the convective limit (for $\Lambda = 0^\circ$), while Eq. 3.56 gives the local mesh Peclet number (also for $\Lambda = 0^\circ$). The non-dimensional group $\aleph$ may be further analyzed by considering τ such that:

$$\tau = \frac{h_e}{V} \frac{\text{Pe}_h}{\sqrt{9 + (\text{Pe}_h)^2}} \quad (3.57)$$

For the determination of h_e , let us consider Fig. 3.7. Misalignment of the flow angle Λ

This figure was broken... original no longer exists.

Figure 3.7: Determination of characteristic length h_e

with the mesh as well as the mesh aspect ratio (δ) results in a dependence of h_e with both Λ and δ . We now take into account, the two following cases of flow misalignment :

Case 1: $\Lambda \leq \tan^{-1} \frac{\Delta y}{\Delta x}$ ($= \tan^{-1}(\delta)$)

$$h_e = \frac{\Delta x}{\cos \Lambda} \quad (3.58)$$

$$\Rightarrow \tau = \frac{\Delta x}{V \cos \Lambda} \frac{\text{Pe}_h}{\sqrt{9 + \text{Pe}_h^2}} \quad (3.59)$$

$$\Rightarrow \aleph = \frac{\text{Pe}_h}{\cos \Lambda \sqrt{9 + \text{Pe}_h^2}} \quad (3.60)$$

Case 2: $\Lambda > \tan^{-1} \frac{\Delta y}{\Delta x}$ ($= \tan^{-1}(\delta)$)

$$h_e = \frac{\Delta y}{\sin \Lambda} \quad (3.61)$$

$$\Rightarrow \tau = \frac{\Delta y}{V \sin \Lambda} \frac{\text{Pe}_h}{\sqrt{9 + \text{Pe}_h^2}} \quad (3.62)$$

$$\Rightarrow \aleph = \frac{\delta}{\sin \Lambda} \frac{\text{Pe}_h}{\sqrt{9 + \text{Pe}_h^2}} \quad (3.63)$$

Since the non-dimensional group $\aleph = \aleph(\delta, \text{Pe}_h, \Lambda)$, we are left with the parameter set $(\delta, \text{Pe}_h, \Lambda)$ for numerical studies. We now consider the splitting of the stencil matrix as applied to the line implicit smoother such that $[\mathbf{A}] = [\mathbf{M}] - [\mathbf{N}]$. Depending on the stencil coefficients, the line creation algorithm will result in one of the two splitting stencils for the damped line Gauß-Seidel smoother:

$$[\mathbf{M}]_1 = \frac{1}{\omega} \begin{bmatrix} 0 & 0 & 0 \\ X & X & X \\ X & X & X \end{bmatrix} \quad (3.64)$$

$$[\mathbf{M}]_2 = \frac{1}{\omega} \begin{bmatrix} X & X & 0 \\ X & X & 0 \\ X & X & 0 \end{bmatrix} \quad (3.65)$$

and

$$[\mathbf{M}]_1 = \frac{1}{\omega} \begin{bmatrix} 0 & 0 & 0 \\ X & X & X \\ 0 & 0 & 0 \end{bmatrix} \quad (3.66)$$

$$[\mathbf{M}]_2 = \frac{1}{\omega} \begin{bmatrix} 0 & X & 0 \\ 0 & X & 0 \\ 0 & X & 0 \end{bmatrix} \quad (3.67)$$

for the damped Jacobi smoother.

Fourier Footprint

According to Eq. 2.26, the application of the basic iterative scheme results in a solution error at iteration $n + 1$ that satisfies

$$e^{n+1} = \mathbf{S}e^n$$

where $\mathbf{S} = \mathbf{M}^{-1}\mathbf{N}$. It is now assumed that the operator $\mathbf{S}$ has a complete set of eigenfunctions or *local modes* denoted by $\Psi(\theta_x, \theta_y), \{\theta_x, \theta_y\} \in \Theta$, with Θ some discrete index set such that

$$\mathbf{S}(\theta_x, \theta_y)\Psi(\theta_x, \theta_y) = \lambda(\theta_x, \theta_y)\Psi(\theta_x, \theta_y) \quad (3.68)$$

and $\lambda(\theta_x, \theta_y)$ is the eigenvalue belonging to $\Psi(\theta_x, \theta_y)$ where $\{\theta_x, \theta_y\}$ are the Fourier angles. Let us now consider a discrete Fourier representation of the exact solution. Let $I_x = \{0, 1, \dots, n_x - 1\}$ and $I_y = \{0, 1, \dots, n_y - 1\}$. Then, every $u^h : \{I_x, I_y\} \rightarrow \mathbb{R}$ can be written as

$$u^h(j_x, j_y) = \sum_{k_x=-m_x}^{m_x+p_x} \sum_{k_y=-m_y}^{m_y+p_y} c_{k_x k_y} \Psi(\theta_x, \theta_y) \quad (3.69)$$

where the Fourier eigenmodes, $\Psi(\theta_x, \theta_y)$, are defined as

$$\Psi(\theta_x, \theta_y) = e^{i(j_x \theta_x + j_y \theta_y)} \quad (3.70)$$

and

$$p_i = 1, m_i = \frac{n_i}{2} - 1 \text{ for } n_i \text{ even} \quad i \in \{x, y\} \quad (3.71)$$

$$p_i = 0, m_i = \frac{n_i - 1}{2} \text{ for } n_i \text{ odd.} \quad i \in \{x, y\} \quad (3.72)$$

For simplicity, it is assumed that n_x and n_y are odd. The Fourier angles as defined in Eq. 3.69 are

$$\begin{aligned} \theta_x &= \frac{2\pi k_x}{n_x} & k_x \in I_x \\ \theta_y &= \frac{2\pi k_y}{n_y} & k_y \in I_y. \end{aligned}$$

The smooth and rough Fourier modes may now be defined by considering a partitioning of the grid wave numbers (Θ) into smooth (Θ_s) and rough (Θ_r) components. In 2D, we may define

$$\Theta = \{\theta : \theta = (\theta_x, \theta_y), \theta_\alpha = \frac{2\pi k_\alpha}{n_\alpha}, k_\alpha = -\frac{n_\alpha - 1}{2}, \dots, \frac{n_\alpha - 1}{2} \quad \alpha = \{x, y\}\} \quad (3.73)$$

such that

$$\Theta = \Theta_s \cup \Theta_r, \quad \Theta_s \cap \Theta_r = \emptyset \quad (3.74)$$

Depending on the coarsening algorithm, we have the following partitions:

Full Coarsening: Standard coarsening in both directions as shown in Fig. 3.8(a) gives

$$\Theta_s = \Theta \cap \prod_{\alpha=1}^2 \left(-\frac{\pi}{2}, \frac{\pi}{2}\right) \quad (3.75)$$

$$\Theta_r = \Theta \setminus \Theta_s \quad (3.76)$$

Semi Coarsening: Semi-coarsening in the y-direction as shown in Fig. 3.8(b) gives

$$\Theta_s = \Theta \cap \left\{[-\pi, \pi] \times \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)\right\} \quad (3.77)$$

$$\Theta_r = \Theta \setminus \Theta_s \quad (3.78)$$

Broken figure

Broken figure

(a) Semi-coarsening

(b) Full coarsening

Figure 3.8: Smooth (Θ_s) and rough (Θ_r , greyed) wavenumber sets in two dimension

The local mode smoothing factor or amplification factor $\bar{\rho}$ is now defined as

$$\bar{\rho} = \sup\{|\lambda(\theta)| : \theta \in \Theta_r\} \quad (3.79)$$

In order to compute $\bar{\rho}$, the eigenvalue problem

$$\begin{aligned} \mathbf{S}\Psi &= \lambda\Psi \\ \Rightarrow \mathbf{N}\Psi &= \lambda\mathbf{M}\Psi \end{aligned}$$

has to be solved. Based on the stencil matrix, substitution of the eigenmodes Ψ gives

$$\lambda(\theta_x, \theta_y) = \frac{\sum_{j,k} \mathbf{N}(j, k) e^{i(j\theta_x + k\theta_y)}}{\sum_{j,k} \mathbf{M}(j, k) e^{i(j\theta_x + k\theta_y)}}, \quad \theta_x, \theta_y \in \{-\pi, \dots, \pi\} \quad (3.80)$$

$$= \frac{\{e^{i(k+1)\theta_y}, e^{ik\theta_y}, e^{i(k-1)\theta_y}\} \mathbf{N} \{e^{i(k+1)\theta_x}, e^{ik\theta_x}, e^{i(k-1)\theta_x}\}^T}{\{e^{i(k+1)\theta_y}, e^{ik\theta_y}, e^{i(k-1)\theta_y}\} \mathbf{M} \{e^{i(k+1)\theta_x}, e^{ik\theta_x}, e^{i(k-1)\theta_x}\}^T} \quad (3.81)$$

$$= \frac{\{e^{i\theta_y}, 1, e^{-i\theta_y}\} \mathbf{N} \{e^{i\theta_x}, 1, e^{-i\theta_x}\}^T}{\{e^{i\theta_y}, 1, e^{-i\theta_y}\} \mathbf{M} \{e^{i\theta_x}, 1, e^{-i\theta_x}\}^T} \quad (3.82)$$

Hence, for the scheme to be a smoother, $\bar{\rho} < 1$. In the context of a multistage scheme, Theorem 3.1 shows that the smoothing factor is determined by the amplification polynomial $\psi_N(z(\theta_x, \theta_y))$ where $z(\theta_x, \theta_y)$ is the eigenvalue of the matrix $-\mathbf{M}^{-1}\mathbf{A}$. The operator $\mathbf{S}$ and $-\mathbf{M}^{-1}\mathbf{A}$ share the same set of eigenfunctions such that

$$z(\theta_x, \theta_y) = - \frac{\{e^{i(k+1)\theta_y}, e^{ik\theta_y}, e^{i(k-1)\theta_y}\} \mathbf{A} \{e^{i(k+1)\theta_x}, e^{ik\theta_x}, e^{i(k-1)\theta_x}\}^T}{\{e^{i(k+1)\theta_y}, e^{ik\theta_y}, e^{i(k-1)\theta_y}\} \mathbf{M} \{e^{i(k+1)\theta_x}, e^{ik\theta_x}, e^{i(k-1)\theta_x}\}^T} \quad (3.83)$$

$$= - \frac{\{e^{i\theta_y}, 1, e^{-i\theta_y}\} \mathbf{A} \{e^{i\theta_x}, 1, e^{-i\theta_x}\}^T}{\{e^{i\theta_y}, 1, e^{-i\theta_y}\} \mathbf{M} \{e^{i\theta_x}, 1, e^{-i\theta_x}\}^T} \quad (3.84)$$

Hence, the multistage smoothing factor is defined by

$$\bar{\rho}(\delta, \text{Pe}_h, \Lambda, \omega) = \sup\{|\psi_N(z(\theta_x, \theta_y))| : \{\theta_x, \theta_y\} \in \Theta_r\} \quad (3.85)$$

δ	Pe_h	Λ	$\aleph$	$\bar{\rho}_G (\omega = 1.0)$	Matrix Splitting
1	10^0	0°	0.316	0.50	Horizontal
1	10^0	10°	0.321	0.49	Horizontal
1	10^0	20°	0.336	0.48	Horizontal
1	10^2	0°	0.999	0.56	Horizontal
1	10^2	10°	1.015	0.49	Horizontal
1	10^2	20°	1.064	0.49	Horizontal
1	10^6	0°	1.000	0.56	Horizontal
1	10^6	10°	1.015	0.49	Horizontal
1	10^6	20°	1.064	0.49	Horizontal

Table 3.2: Fourier Smoothing Factor: Full coarsening and 3-stage scheme with optimized multistage coefficients

δ	Pe_h	Λ	$\aleph$	$\bar{\rho}_G (\omega = 1.0)$	Matrix Splitting
10^{-2}	10^0	0°	0.316	0.55	Vertical
10^{-2}	10^0	10°	0.018	0.55	Vertical
10^{-2}	10^0	20°	0.009	0.55	Vertical
10^{-2}	10^2	0°	0.999	0.55	Vertical
10^{-2}	10^2	10°	0.058	0.54	Vertical
10^{-2}	10^2	20°	0.029	0.54	Vertical
10^{-2}	10^6	0°	1.000	0.55	Horizontal
10^{-2}	10^6	10°	0.058	0.52	Vertical
10^{-2}	10^6	20°	0.029	0.53	Vertical

Table 3.3: Fourier Smoothing Factor: Semi-coarsening and 3-stage scheme with optimized multistage coefficients

We consider the 3-stage line Gauß-Seidel smoother for different system parameters as shown in Table 3.2 and Table 3.3 where the smoothing factor for the line Gauß-Seidel smoother is denoted by $\bar{\rho}_G$. Table 3.2 represents a mesh aspect ratio of 1 so that full coarsening is applied. Table 3.3, however represents a mesh aspect ratio of 100 which

introduces strong mesh-induced anisotropic effects and as such, semi-coarsening is applied.

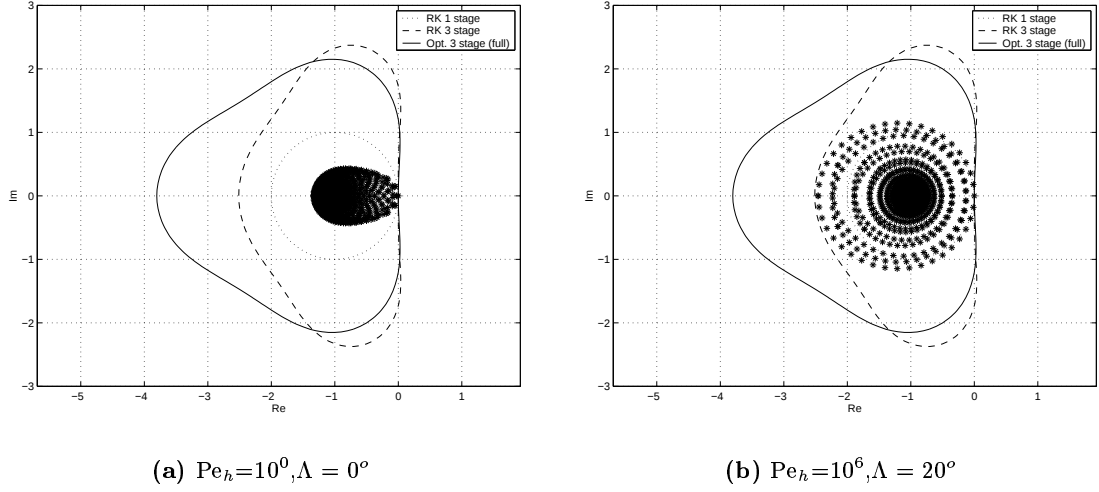


Figure 3.9: Fourier footprints for line Gauß-Seidel smoother using full coarsening ($\delta=1$)

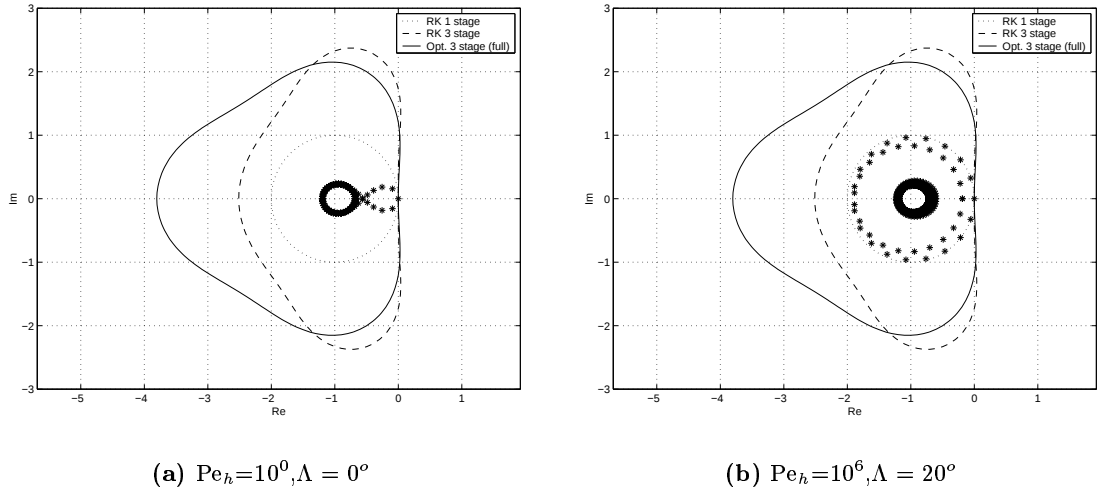


Figure 3.10: Fourier footprints for line Gauß-Seidel smoother using semi-coarsening ($\delta=10^{-2}$)

The last column in Table 3.2 and Table 3.3 denotes the type of line that is created by the line creation algorithm based on the magnitude of the coefficients in the $[\mathbf{A}]$ matrix

stencil. The resulting plots of the Fourier footprint over the Fourier space for some of the different problems are shown in Fig. 3.9 and Fig. 3.10.

The multistage amplification factors for the implicit line schemes show good damping properties for these schemes in general, even in the presence of flow misalignment. In all cases, the implicit Gauß-Seidel scheme is stable and in most cases, is stable for a single stage formulation with light or no damping. The anisotropic mesh results depicted in Table 3.3 show a fairly consistent trend. The strong anisotropy induced by the mesh results in all the lines being created vertically. The exception to this is the case where $\text{Pe}_h = 10^6, \Lambda = 0^\circ$ which resulted in horizontal lines. Introduction of flow misalignment as well as higher local mesh Peclet number have little effect on the amplification factors.

The conclusion of the Fourier analysis performed is that the implicit line Gauß-Seidel smoother in conjunction with a multistaging scheme is a viable option especially in the anisotropic mesh regions.

3.6 Consistency Scaling Issues for Stabilized Methods

Elemental agglomeration brings up special issues when AMG is applied to convection-diffusion operators which are discretized using stabilized methods such as SUPG or GLS. Let us consider the matrix equation (Eq. 2.22) where $\mathbf{A}_k$ is some matrix resulting from a stabilized method such that

$$\mathbf{A}_k = \mathbf{A}_k^{\text{base}} + \mathbf{A}_k^\tau \quad (3.86)$$

where $\mathbf{A}_k^\tau$ is the component containing the stabilization parameter and $\mathbf{A}_k^{\text{base}}$ is the base stiffness matrix (Galerkin + boundary terms for FEM). Artificial dissipation schemes [107] and stabilization schemes for convection-diffusion operators [93–95] operate by the addition of a stabilization term of the form $\tau \nabla^2 u$, where $\tau \sim \mathcal{O}(h)$. As discussed earlier, nodal agglomeration techniques which use injection based interpolation operators result in consistent coarse space convective terms but fail for the constant-coefficient diffusive terms. However, elemental agglomeration results in both consistent convective and diffusive terms on the coarse spaces. Following the analysis of Appendix B, consider a Laplacian operator with an $\mathcal{O}(h)$ coefficient:

$$h \frac{d^2 u}{dx^2} \quad (3.87)$$

Discretization of this term on the 1D grid as shown in Appendix B yields

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{h} \quad (3.88)$$

such that a discretization on a coarse grid with spacing $H = 2h$ would result in the coarse grid term

$$\frac{\bar{u}_{I+1} - 2\bar{u}_I + \bar{u}_{I-1}}{2h} \quad (3.89)$$

If both the restriction and prolongation operators are based on injection, as is the case in nodal agglomeration, we may multiply the left hand side of Eq. B.5 by h to obtain the discrete coarse grid term:

$$\frac{\bar{u}_{I+1} - 2\bar{u}_I + \bar{u}_{I-1}}{2h} \approx H \left. \frac{d^2 \bar{u}}{dx^2} \right|_{\Omega_H} \quad (3.90)$$

This is consistent with a discretization on the coarse grid. We now perform a similar analysis with a restriction operator based on injection and a prolongation based on linear interpolation. We multiply the left hand side of Eq. B.6 by h to obtain the discrete coarse grid term:

$$\frac{\bar{u}_{I+1} - 2\bar{u}_I + \bar{u}_{I-1}}{4h} \approx \frac{H}{2} \left. \frac{d^2 \bar{u}}{dx^2} \right|_{\Omega_H} \quad (3.91)$$

which is *inconsistent* with a discretization of the stabilization on the coarse grid. This implies that there is an improper scaling of the stabilization parameter on the coarse grid due to the h -dependence of the coefficient. This provides an interesting duality between nodal agglomeration which fails to discretize the constant coefficient diffusive terms but, as a result of this inaccuracy, properly scales stabilization contributions, whereas for elemental agglomeration the opposite is true. In order to address this stabilization issue, we introduce a length scaling matrix σ . The matrix is split according to Eq. 3.86 and the coarse grid matrices are computed using the GCA formulation

$$\mathbf{A}_{k+1} = \mathbf{R}_k \mathbf{A}^{\text{base}} \mathbf{P}_k + \sigma \mathbf{R}_k \mathbf{A}^\tau \mathbf{P}_k \quad (3.92)$$

$$\equiv \mathbf{A}_{k+1}^{\text{base}} + \mathbf{A}_{k+1}^\tau \quad (3.93)$$

where σ is a diagonal scaling matrix which accounts for h -dependency in $\mathbf{A}^\tau$. This matrix is currently implemented by computing a characteristic area for each node based on the nodal

control volume and taking the square root of this area. The currently implemented scaling procedure is only fully appropriate for a full coarsening scheme with isotropic elements. A more appropriate version for semi-coarsening with anisotropic elements has not been developed yet.

The effect of τ stabilization scaling is tested by plotting the eigenspectrum of the system matrix $\mathbf{A}_0$ and the coarse space matrix $\mathbf{A}_1$ with and without scaling. The test case considered is the linear convection diffusion equation discretized over a square domain $\Omega =]0, 1[^2$ (Fig. 3.4(a)) with a prescribed velocity field $\mathbf{U} = (-y, x)$ and Peclet number of 10^6 . An eigenspectrum decomposition of the fine grid matrix and the first coarse grid matrix is shown in Fig. 3.11, with and without τ scaling. The convection equation is a hyperbolic equation with imaginary eigenvalues. However, the numerical discretization introduces dissipation which is manifested in the real part of the eigenvalues of the discrete system. The eigenvalues λ for a Finite Difference approximation of the stabilization term are such that $\lambda \sim \frac{1}{h}$ from which we may expect the eigenvalues $\bar{\lambda}$ for the FEM discretization to follow:

$$\bar{\lambda} \sim \int_{\Omega} \left(\frac{1}{h} \right) d\Omega \quad (3.94)$$

$$\sim \frac{1}{h} \quad (3.95)$$

For an approximate 4:1 full coarsening ratio of the coarse grid such that $\frac{H}{h} \approx 2$, we expect the real portion of the coarse grid matrix eigenvalues to scale similarly. This behavior is observed in Fig. 3.11(c) where stabilization scaling is performed. Fig. 3.11(b), which represents the coarse grid matrix obtained without stabilization scaling, does not preserve this property. In fact, it may be observed that it appears to scale the imaginary parts of the eigenvalues with h without scaling the real parts.

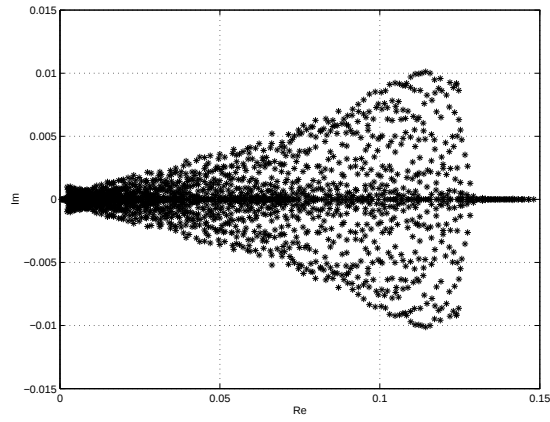
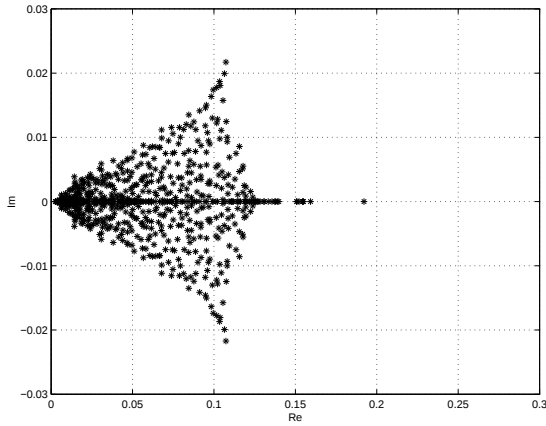
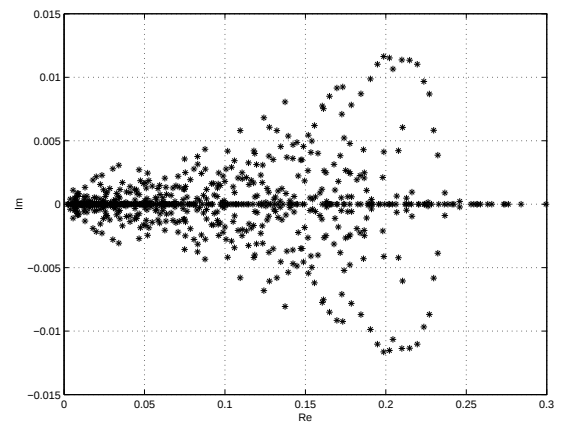
(a) Fine grid matrix $\mathbf{A}_0$ (b) Coarse grid matrix $\mathbf{A}_1$ w/o rescaling(c) Coarse grid matrix $\mathbf{A}_1$ w/ rescaling

Figure 3.11: Comparison of scalar convection-diffusion equation eigenspectrum w/ and w/o τ scaling for $Pe = 1e6$

3.7 Results: Elliptic Operator (revisited)

As observed in Chap. 2, the convergence of the proposed multigrid algorithm degrades appreciably when applied to the Poisson problem discretized on a stretched grid. This degradation becomes worse with increasing grid anisotropy and can be partly ameliorated through semi-coarsening. Unfortunately, the directional decoupling of the error modes due to the grid stretching is such that semi-coarsening may not be enough as illustrated in Table 2.3. An effective solution to this is a combination of semi-coarsening and directional smoothing [6, 71].

We revisit the anisotropic grid results of Chap. 2 and apply the modified multigrid algorithm using the developed line Gauß-Seidel implicit smoother to the anisotropic grid discretization of the Poisson problem (Eq. 2.59), on the same domain $\Omega =]0, 1[^2$ and for the same choice of initial aspect ratio δ_s . The asymptotic multigrid convergence rates ϵ are shown in Table 3.4 for the point implicit and line implicit smoothers using semi-coarsening. As can be observed, grid independent convergence rates are achieved.

δ_s	ω	ϵ (Point Implicit)	ϵ (Line Implicit)
1	1.0	0.100	0.054
2	1.0	0.100	0.067
4	1.0	0.339	0.072
100	1.0	0.408	0.059

Table 3.4: Multigrid results for Poisson problem on anisotropic mesh using semi-coarsening: Comparison of convergence rates for point and line Gauß-Seidel smoothers

3.8 Results: Convection-Diffusion Operator

We consider the linear convection diffusion equation (Eq. 3.1) over a square domain defined by $\Omega =]0, 1[^2$ (Fig. 3.12) and prescribed velocity field $\vec{U} = (-y, x)$. The forcing function f is set to zero, the Neumann outflow boundary on the upper boundary is $g_N = 0$ and the

Dirichlet boundary on the other boundaries is

$$g_D = \begin{cases} 5.0(x - 0.2), & \text{for } 0.2 < x \leq 0.4, & y = 0 \\ 1, & \text{for } 0.4 < x \leq 0.6, & y = 0 \\ 1 - 5(x - 0.6), & \text{for } 0.6 < x \leq 0.8, & y = 0 \\ 0, & \text{otherwise.} \end{cases}$$

This particular set of conditions is chosen to simulate a boundary layer flow with the nominal Peclet number

$$\text{Pe} = \frac{U_h \cdot L_h}{\mu} = \frac{1}{\mu} \quad (3.96)$$

This figure was broken... original no longer exists.

Figure 3.12: Computational domain for scalar convection diffusion boundary layer problem

The discretized domain is adapted on the $x = 0$ boundary to capture the boundary layer as shown in Fig. 3.4(a). All presented results are based on a V(1,1) multigrid cycle using the line Gauß-Seidel smoother and the solver is terminated when the RMS absolute error in the residual is less than 10^{-13} . The preconditioning matrix for the proposed line smoother was stable without any multistaging, hence, all the results presented are based on a single stage formulation. The relaxation factor ω chosen for all the test cases was 0.95.

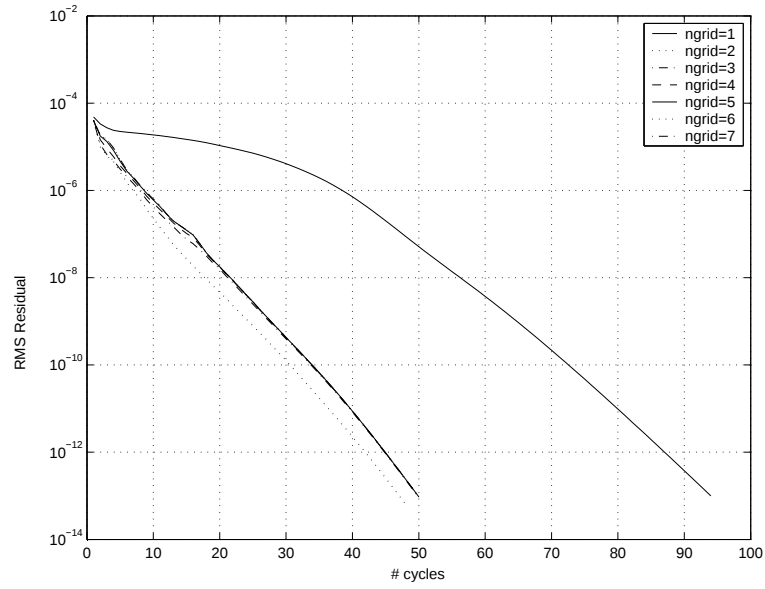


Figure 3.13: Boundary Layer Grid Level Dependency: 60399 points; $Pe = 1e6$

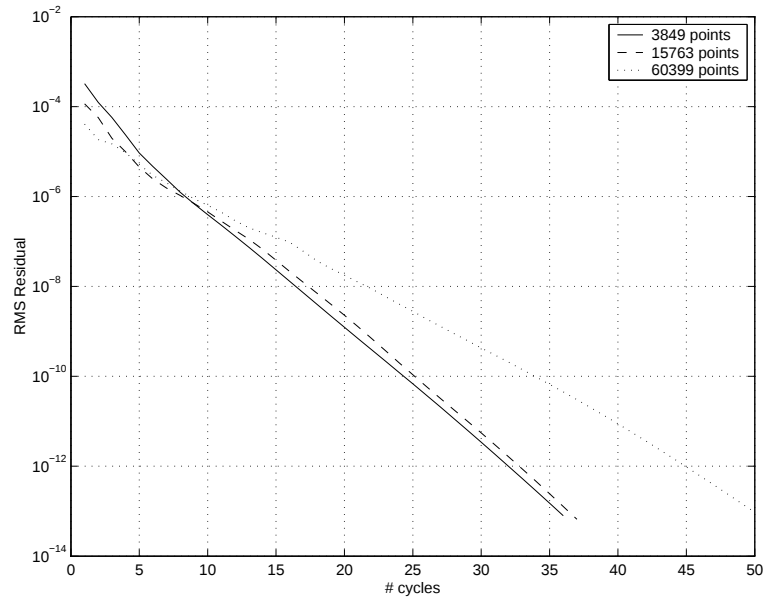
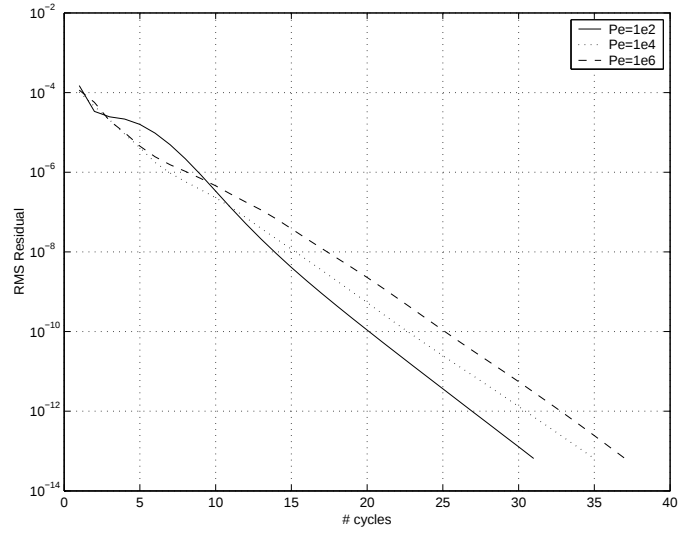
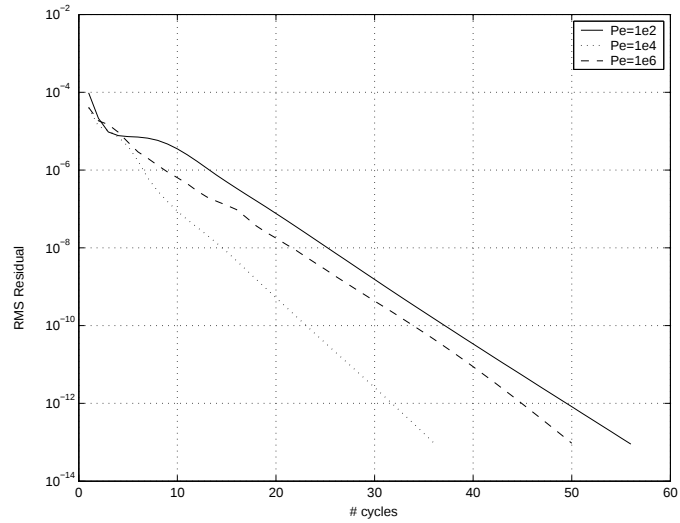


Figure 3.14: Boundary Layer Mesh Size Dependency: $Pe = 1e6$



(a) 15763 points



(b) 60399 points

Figure 3.15: Multigrid Results for Scalar Convection-Diffusion: Peclet number dependency

Multigrid Level Dependency:

The dependence of the convergence rate on the number of coarse spaces is shown in Fig. 3.13. The fine mesh has 60,399 vertices and 119,714 elements and a total of 6 coarse grids were constructed (7 including the fine grid). In the asymptotic limit, the convergence rate is the same for all the curves and beyond the two-grid case, the curves fall onto the same line.

Mesh Size Dependency:

The dependence of the convergence rate on the mesh size is shown in Fig. 3.14. Three meshes with increasing mesh sizes of 3,849, 15,763 and 60,399 points were used. Good convergence properties are observed with some departure for the largest mesh. This can probably be attributed to the fact that the meshes were not generated by refinement as well as the simplification for the stabilization parameter τ which was employed. This may result in too little/much stabilization in the boundary layer region.

Peclet Number Dependency:

The dependence of the convergence rate on the Peclet number is shown in Fig. 3.15(a) and Fig. 3.15(b) for a range of Reynolds numbers from 10^2 to 10^6 . Figures 3.15(a) and 3.15(b) were generated on a set of fine meshes with 15763 and 60399 vertices respectively which represents an approximate halving of the mesh spacing. In both cases, we find a similar asymptotic convergence rate. Even more important is the fact that the algorithm works well for such a wide range of Peclet numbers while maintaining a fairly constant bound on the number of iterations required for convergence.

The average convergence rates ($\bar{\epsilon}$) and asymptotic convergence rates (ϵ) for the different test cases are summarized in Table 3.5.

Grid Size	Pe	Average Convergence $\bar{\epsilon}$	Asymptotic Convergence ϵ
3849	10^2	0.31	0.29
15763	10^2	0.47	0.50
60399	10^2	0.67	0.69
3849	10^4	0.53	0.57
15763	10^4	0.52	0.55
60399	10^4	0.55	0.58
3849	10^6	0.52	0.54
15763	10^6	0.54	0.54
60399	10^6	0.65	0.63

Table 3.5: Agglomeration Multigrid Results for scalar convection-diffusion problem

Chapter 4

Euler Applications

The Euler equations describe the flow of an inviscid fluid and may be considered to be the limit of the Navier-Stokes equations as the Reynolds number goes to infinity. The Navier-Stokes equations contains the full set of physical processes which occur in fluid flows but for many problems, viscosity may be neglected. In attached flows where viscosity is important, this importance is emphasized in a thin region near solid boundaries such that the remaining flow is convection-dominated. Hence, the Euler equations provide a good testbed for multigrid algorithms developed for the Navier-Stokes equations.

Standard multigrid methods have been applied to the Euler equations with varying success [23, 24, 35, 56] due to inherent properties of the Euler equations. The Euler equations are a non-linear, non-elliptic system of equations which do not satisfy any of the underlying assumptions of these standard multigrid algorithms that were designed based on elliptic operators. Brandt has summarized the current progress and outlined the barriers to achieving ideal multigrid convergence rates for the equations of fluid dynamics [30]. Using local mode analysis, Brandt [64] shows that for a p -th order Finite Difference approximation of the constant-coefficient advection-diffusion equation, the convergence rate of standard multigrid algorithms asymptotically approaches $(1 - 2^{-p})$ in the limit of vanishing diffusivity coefficient. This is based on the argument that the coarse grid only provides a fraction $(1 - 2^{-p})$ of the required correction for the smooth error components which limits the convergence rate of the multigrid process. Brandt proposed that in order to obtain ideal multigrid convergence rates for subsonic, inviscid flows, the discretization must be able to effectively distinguish between the elliptic and hyperbolic factors of the governing differential opera-

tor. By splitting the system into advective and elliptic components, the convergence rate of the full system should be limited by the convergence rate of the slower of the two subsystems. This leads to Brandt introducing the distributive relaxation scheme. Typically, multigrid techniques developed using this principle are based on space marching of the advective terms while the elliptic terms are treated with multigrid. Using this approach, Brandt and Yavneh [65] have shown ideal multigrid convergence rates for the incompressible Navier-Stokes equations for a simple geometry and a Cartesian grid, using a staggered-grid discretization of the equations. A closely related approach was presented by Ta'asan [108] for the compressible Euler equations using a set of canonical variables which partitions the Euler equations into elliptic and hyperbolic components. Ideal multigrid convergence rates are demonstrated for two dimensional subsonic flows using body fitted grids. An extension of the distributive relaxation scheme by Brandt for the incompressible Navier-Stokes equations applied to high Reynolds number wakes and boundary layers was done by Thomas *et al* [29]. This was subsequently extended to the compressible Navier-Stokes equations [66].

Roberts *et al* [28] present an alternative to distributive relaxation and to Ta'asan's canonical variable decomposition for the steady, incompressible Euler equations. This is based on a generalization of the approach of Sidilkover and Ascher [109] which applies a projection operator to the system of equations such that a Poisson equation for the pressure may be constructed. Ideal multigrid convergence rates were demonstrated for internal flows for both structured grids using Finite Difference and unstructured grids using a Finite Volume discretization. However, application of this algorithm to airfoil flows led to difficulties near stagnation points which were not present in internal flows. An extension of the algorithm was made in [110] for airfoil flows by the introduction of an artificial dissipation term which stabilizes the momentum equations in stagnation regions.

The pressure Poisson method may be extended to compressible flows, however it is not conservative and is not suitable for flows with shocks. Also, the extension to viscous flows is limited to the incompressible Navier-Stokes equations. Sidilkover obtained a discretization of the compressible flow equations that overcomes these limitations based on a multidimensional upwind scheme [111, 112] which is shown to be factorizable into advective and elliptic components. Roberts *et al* demonstrate ideal multigrid convergence rates based on a generalization of Sidilkover's factorizable scheme in internal flows for a range of Mach numbers from low subsonic to supercritical [113]. This is done using a Finite Differencing scheme on curvilinear, body-fitted grids.

In our approach to multigrid for the Euler equations, we make use of the principles developed in Chap. 3 for the scalar convection-diffusion equations. The fundamental idea is the construction of a line implicit smoother which effectively removes the hyperbolic error components, while still smoothing the elliptic errors. In this sense, the line implicit smoother is equivalent to the space marching used by Brandt *et al* for the hyperbolic components while the multigrid coarse spaces are used to treat the elliptic components. This chapter focuses on the extension and application of the proposed multigrid algorithm to the Euler equations. In this chapter we consider the extension of the proposed AMG algorithm as applied to a stabilized Finite Element discretization of the Euler equations. As analyzed in Chap. 3, the application of multigrid to stabilized schemes raises several issues regarding the representation of the stabilization terms on the coarse spaces. This is further analyzed for the Euler equations and will be shown to have important consequences for the implicit line smoother.

4.1 FEM Discretization

The discretization used for the Euler equations is based on the Finite Element code provided by Wong [114] which we shall now proceed to describe. Let us consider the time dependent 2D compressible Euler equations in conservative form

$$\mathbf{U}_{,t} + \mathbf{F}_{1,x_1} + \mathbf{F}_{2,x_2} = 0, \quad (4.1)$$

where

$$\mathbf{U} = \begin{Bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho E \end{Bmatrix}, \mathbf{F}_1 = \begin{Bmatrix} \rho u_1 \\ \rho u_1^2 + p \\ \rho u_1 u_2 \\ u_1(\rho E + p) \end{Bmatrix}, \mathbf{F}_2 = \begin{Bmatrix} \rho u_2 \\ \rho u_1 u_2 \\ \rho u_2^2 + p \\ u_2(\rho E + p) \end{Bmatrix}. \quad (4.2)$$

such that ρ is the density; $\mathbf{u} = \{u_1, u_2\}^T$ is the velocity vector; E is the specific total energy and p is the pressure. The system of equations is closed through the equation of state, $p = (\gamma - 1)\rho e$, where $e = E - |\mathbf{u}|^2/2$ is the internal energy and γ is the ratio of specific heats which is assumed to be constant. The conservative form of the above equation allows shock waves to be captured as weak solutions of the governing equations which avoids

difficulties in the use of shock fitting techniques for arbitrary geometries. We may now non-dimensionalize the above variables using reference values for density (ρ^*), velocity (u^*) and length (L) via

$$\bar{\rho} = \frac{\rho}{\rho^*}, \quad \bar{u}_i = \frac{u_i}{u^*}, \quad i = 1, 2, \quad \bar{p} = \frac{p}{\rho^* u^{*2}}, \quad \bar{E} = \frac{E}{u^{*2}}, \quad \bar{x}_i = \frac{x_i}{L}, \quad i = 1, 2, \quad \text{and} \quad \bar{t} = \frac{u^* t}{L}.$$

From this point, we will drop the overbars which denote the non-dimensionalized variables. Given the reference speed of sound c^* , we may now define the reference Mach number:

$$M = \frac{u^*}{c^*} \tag{4.3}$$

Equation (4.1) can be written in quasi-linear form as

$$\mathbf{U}_{,t} + \mathbf{A}_1 \mathbf{U}_{,x_1} + \mathbf{A}_2 \mathbf{U}_{,x_2} = 0 \tag{4.4}$$

where the inviscid Jacobian matrices $\mathbf{A}_i = \mathbf{F}_{i,\mathbf{U}}$, $i = 1, 2$, are non-symmetric but have real eigenvalues and a complete set of eigenvectors. Equation (4.4) is symmetrized by the introduction of entropy variables $\mathbf{V}$ [115–117], such that the change $\mathbf{U} = \mathbf{U}(\mathbf{V})$ applied to (4.1) gives the transformed system

$$\mathbf{U}(\mathbf{V})_{,t} + \mathbf{F}_1(\mathbf{V})_{,x_1} + \mathbf{F}_2(\mathbf{V})_{,x_2} = 0 \tag{4.5}$$

or equivalently in symmetric quasi-linear form as

$$\mathbf{A}_0 \mathbf{V}_{,t} + \tilde{\mathbf{A}}_1 \mathbf{V}_{,x_1} + \tilde{\mathbf{A}}_2 \mathbf{V}_{,x_2} = 0 \tag{4.6}$$

where $\mathbf{A}_0 = \mathbf{U}_{,\mathbf{V}}$ is symmetric positive definite, and $\tilde{\mathbf{A}}_i = \mathbf{A}_i \mathbf{A}_0 = \mathbf{F}_{i,\mathbf{V}}$, $i = 1, 2$, are symmetric.

For a hyperbolic system defined by

$$\mathbf{u}_{,t} + \mathbf{f}^i(\mathbf{u})_{x_i} = 0 \tag{4.7}$$

Barth [116] outlines two fundamental theorems regarding the symmetrization of systems via entropy functions:

Theorem 4.1 (Godunov [118]) *If a hyperbolic system (Eq. 4.7) is symmetrized via change of variables, then there exists a generalized entropy pair $\{U(\mathbf{u}), F^i(\mathbf{u})\} : \mathbb{R}^d \mapsto \mathbb{R}$ such that*

$$U_{,t} + F_{,x_i}^i \leq 0 \quad (4.8)$$

Theorem 4.2 (Mock [119]) *If a hyperbolic system (Eq. 4.7) is equipped with a generalized entropy pair $\{U(\mathbf{u}), F^i(\mathbf{u})\} : \mathbb{R}^d \mapsto \mathbb{R}$, then the system is symmetrized under the change of variables*

$$\mathbf{v}^T = U_{,\mathbf{u}} \quad (4.9)$$

An explicit formulation of the flux Jacobians is given by

$$\mathbf{v}^T \mathbf{f}_{,\mathbf{u}}^i = F_{,\mathbf{u}}^i \quad (4.10)$$

such that an inner product of Eq. 4.7 and the entropy variable yields

$$\mathbf{v}^T (\mathbf{u}_{,t} + \mathbf{f}_{,x_i}^i) = U_{,t} + F_{,x_i}^i = 0 \quad (4.11)$$

for smooth solutions.

Following Harten [117], we introduce a scalar entropy function $U(\mathbf{U}) = -\rho g(s)$, where s is the non-dimensional entropy $s = \ln(p/\rho^\gamma)$. The required change of variables is obtained by taking

$$\mathbf{V} = H_{,\mathbf{U}}^T = \frac{g'}{e} \begin{bmatrix} e(\gamma - g/g') - |\mathbf{u}|^2/2 \\ u_1 \\ u_2 \\ -1 \end{bmatrix}. \quad (4.12)$$

The conditions $g' > 0$ and $g''/g' < \gamma^{-1}$, ensure that $U(\mathbf{U})$ is a convex function and therefore $\mathbf{A}_0^{-1} = \mathbf{V}_{,\mathbf{U}} = H_{,\mathbf{U}\mathbf{U}}$, and $\mathbf{A}_0$, are symmetric positive definite. For the Euler equations, if we chose $g(s) = e^{\frac{s}{2\gamma}}$ then $\mathbf{V}$ takes the form:

$$\mathbf{V} = \sqrt{\rho p^{\frac{1-2\gamma}{\gamma}}} \begin{bmatrix} E + \frac{p}{\rho} \\ -u_1 \\ -u_2 \\ 1 \end{bmatrix} \quad (4.13)$$

The fluxes $\mathbf{F}_1$ and $\mathbf{F}_2$ expressed as functions of $\mathbf{V}$, are now homogeneous functions of degree $q = -\frac{2\gamma}{\gamma-1}$. Let us now consider the variational formulation for the steady state problem. The problem is defined in a domain Ω with boundary Γ by

$$\mathbf{F}_1(\mathbf{V})_{,x_1} + \mathbf{F}_2(\mathbf{V})_{,x_2} = 0 \quad \text{in } \Omega, \quad (4.14)$$

$$\tilde{\mathbf{A}}_n^- \mathbf{V} = \tilde{\mathbf{A}}_n^- \mathbf{g} \quad \text{on } \Gamma \setminus \Gamma_w, \quad (4.15)$$

$$\mathbf{u} \cdot \mathbf{n} = 0 \quad \text{on } \Gamma_w. \quad (4.16)$$

where $\mathbf{g}$ is boundary defined data and the domain boundary is made up of an impermeable solid wall Γ_w , and a computational far field boundary $\Gamma_{ff} = \Gamma \setminus \Gamma_w$. In (4.15, 4.16), $\mathbf{n} = [n_1, n_2]^T$ is the outward unit normal vector to Γ , and $\tilde{\mathbf{A}}_n = \mathbf{A}_n \mathbf{A}_0$, $\mathbf{A}_n = \mathbf{A}_1 n_1 + \mathbf{A}_2 n_2$. Finally, $\tilde{\mathbf{A}}_n^- = \mathbf{A}_n^- \mathbf{A}_0$ where $\mathbf{A}_n^-$ denotes the negative definite part of $\mathbf{A}_n$. Let the spatial domain Ω , be discretized into non-overlapping elements T_e , such that $\Omega = \bigcup T_e$, and $T_e \cap T_{e'} = \emptyset$, $e \neq e'$. We consider the space of functions $\mathcal{V}_h$, defined over the discretization and consisting of the continuous functions which are piecewise linear over each element

$$\mathcal{V}_h = \{\mathbf{W} \mid \mathbf{W} \in (C^0(\Omega))^4, \mathbf{W}|_{T_e} \in (\mathcal{P}_1(T_e))^4, \forall T_e \in \Omega\}.$$

The discrete GLS formulation can then be written as:

Find $\mathbf{V}_h \in \mathcal{V}_h$ such that:

$$\mathbf{r}(\mathbf{V}_h) = B(\mathbf{V}_h, \mathbf{W})_{gal} + B(\mathbf{V}_h, \mathbf{W})_{gls} + B(\mathbf{V}_h, \mathbf{W})_{bc} = 0, \forall \mathbf{W} \in \mathcal{V}_h \quad (4.17)$$

where the forms $B(\cdot, \cdot)_{gal}$, $B(\cdot, \cdot)_{gls}$ and $B(\cdot, \cdot)_{bc}$ account for the Galerkin, GLS stabilization, and boundary condition terms respectively, and are defined as

$$B(\mathbf{V}_h, \mathbf{W})_{gal} = \int_{\Omega} (-\mathbf{W}_{,x_1} \cdot \mathbf{F}_1(\mathbf{V}_h) - \mathbf{W}_{,x_2} \cdot \mathbf{F}_2(\mathbf{V}_h)) d\Omega, \quad (4.18)$$

$$B(\mathbf{V}_h, \mathbf{W})_{gls} = \int_{\Omega} \{\mathbf{F}_1(\mathbf{W}) + \mathbf{F}_2(\mathbf{W})\} \cdot \boldsymbol{\tau} \{\mathbf{F}_1(\mathbf{V}_h) + \mathbf{F}_2(\mathbf{V}_h)\} d\Omega, \quad (4.19)$$

$$B(\mathbf{V}_h, \mathbf{W})_{bc} = \int_{\Gamma \setminus \Gamma_a} \mathbf{W} \cdot \mathbf{F}_{ff}(\mathbf{V}_h, \mathbf{g}; \mathbf{n}) ds. + \int_{\Gamma_a} \mathbf{W} \cdot \mathbf{F}^w(\mathbf{V}_h, \mathbf{n}) ds. \quad (4.20)$$

where $\boldsymbol{\tau}$ is the stabilization matrix which must be symmetric, positive definite, have dimensions of time and scale linearly with the element size [94]. Standard definitions for $\boldsymbol{\tau}$

have been derived which work well in general practice [91, 116] . However, these choices have inappropriate low Mach number behavior [114] such that there is a degradation in the solution with decreasing accuracy as the Mach number is reduced. The low Mach number τ stabilization matrix as described by Wong *et al* [114] is employed to complete the GLS algorithm.

For the Euler equations, the numerical flux function $\mathbf{F}_w$ on the impermeable wall boundary, is simply $[0, pn_1, pn_2, 0]^T$. The numerical flux function on the far field boundary $\mathbf{F}_{ff}$, is defined by

$$\mathbf{F}_{ff}(\mathbf{V}_h^-, \mathbf{V}_h^+; \mathbf{n}) = \frac{1}{2}(\mathbf{F}_n(\mathbf{V}_h^-) + \mathbf{F}_n(\mathbf{V}_h^+)) - \frac{1}{2}|\mathbf{A}_n(\mathbf{V}_h^{Roe}(\mathbf{V}_h^-, \mathbf{V}_h^+))|(\mathbf{U}(\mathbf{V}_h^+) - \mathbf{U}(\mathbf{V}_h^-)).$$

Here, $|\mathbf{A}_n(\mathbf{V}_h)| = \mathbf{A}_n^+(\mathbf{V}_h) - \mathbf{A}_n^-(\mathbf{V}_h)$ is the absolute value of $\mathbf{A}_n$ evaluated at $\mathbf{V}_h$, and $\mathbf{V}_h^{Roe}(\mathbf{V}_h^+, \mathbf{V}_h^-)$, is the Roe average [120], between the states $\mathbf{V}_h^+$ and $\mathbf{V}_h^-$.

4.2 AMG Extension to the Euler Equations

Eq. 4.17 is a non-linear system of equations to which the linear multigrid formulation no longer applies. For the solution of non-linear equations, either generalized non-linear multigrid formulations can be defined for the non-linear problem [24, 76] or the linear multigrid algorithms can be applied to a linearization of the problem in a Newton solution context [50, 51]. A popular non-linear multigrid scheme is the Full Approximation Storage (FAS) scheme developed by Brandt [76, 121]. Non-linear multigrid methods require a re-evaluation of the full non-linear residual at each iteration on all grid levels but do not require construction and storage of a Jacobian matrix as in Newton methods. This provides an advantage in terms of memory savings but may be costly on a cpu-time efficiency basis when the non-linear residual is expensive to evaluate. Also, non-linear methods may fail due to non-existence of a solution to the physical problem when discretized on the coarse mesh.

Newton solution methods for non-linear problems which implement linear multigrid solvers can also fail when the initial guess does not lie in the domain of convergence for the non-linear problem. This however, may be overcome by the use of a quasi-Newton method which implements pseudo-time stepping. It has been shown by Mavriplis [9] that in the asymptotic convergence region, where solution updates are small and the effects of non-linearities decrease, both the non-linear multigrid algorithm as well as the Newton scheme

converge at the same rate per multigrid cycle provided that equivalent iteration strategies are employed for both. This however is only valid for an exact Newton linearization for the Jacobian, which is usually violated in practice where a first order discretization for the Jacobian is employed and a higher-order discretization for the residual is implemented. In the current context of this thesis however, we implement the Newton solution strategy with an exact linearization for the flux Jacobian matrix. This is possible due to the compactness of the GLS Finite Element discretization which results in a nearest neighbor stencil for the non-linear residuals.

4.2.1 Newton Scheme

For the extension of the previously described linear multigrid scheme to the Euler equations, we consider the Newton scheme applied to the time dependent form of Eq. 4.17:

$$B(\mathbf{V}_h, \mathbf{W})_t + \mathbf{r}(\mathbf{V}_h) = 0, \quad \forall \mathbf{W} \in \mathcal{V}^h \quad (4.21)$$

where $B(\mathbf{V}_h, \mathbf{W})_t$ accounts for the time dependent term and is defined by:

$$B(\mathbf{V}_h, \mathbf{W})_t = \int_{\Omega} \mathbf{A}_0 \mathbf{W} \frac{\partial \mathbf{V}_h}{\partial t} d\Omega \quad (4.22)$$

If we consider a simple first order discretization for the time derivative term, then the Newton scheme for solution advancement may be written as:

$$[\mathbf{A}_t + \mathbf{J}] \Delta \mathbf{V}_h^n = -\mathbf{r} \quad (4.23)$$

$$\mathbf{V}_h^{n+1} = \Delta \mathbf{V}_h^n + \mathbf{V}_h^n \quad (4.24)$$

where $\mathbf{J}$ is the exact Jacobian of the non-linear spatial residual and $\mathbf{A}_t$ is the mass matrix:

$$\mathbf{A}_t = \frac{1}{\Delta t^n} \int_{\Omega} \mathbf{A}_0 \mathbf{W} d\Omega \quad (4.25)$$

$$\mathbf{J} = \frac{\partial \mathbf{r}(\mathbf{V}_h)}{\partial \mathbf{V}_h} \quad (4.26)$$

The time step Δt^n is chosen to be the maximum time step allowed by the Courant-Friedrichs-Lewy (CFL) condition for an explicit first order scheme [122] modified by a function of the

iteration number n :

$$\Delta t^n = \frac{\Delta t_{\text{CFL}}}{\bar{f}(n)} \quad (4.27)$$

where $\bar{f}(1) = 1$ and $\lim_{n \rightarrow \infty} \bar{f}(n) = 0$ which ensures that the pure Newton scheme is recovered. The mass matrix $\mathbf{A}_t$ may be retained for a number of iterations to ensure that the system is out of the initial non-linear startup phase after which it is turned off. In practice however, it is found that this procedure is usually not required for smooth solutions of the Euler equations or for small enough grid sizes.

Algorithm 3 Non-linear solution procedure for linear m-level multigrid

```

repeat
  Form residual  $\mathbf{r}$ , and  $\mathbf{A}^{\text{gal}}$ ,  $\mathbf{A}^{\text{bc}}$ ,  $\mathbf{A}^{\text{gls}}$  matrices.
  Set grid level  $\Omega_{k=0}$ .
  while sizeof(grid level  $k$ ) > specified size do
    Create implicit lines based on  $\mathbf{A}_k^{\text{gal}}$ 
    Create agglomerated coarse grid  $\Omega_{k+1}$ 
    Create interpolation operators  $\mathbf{P}_k^{\text{gal}}$ ,  $\mathbf{R}_k^{\text{gal}}$ ,  $\mathbf{R}_k^{\text{bc}}$ 
    Create coarse grid matrices  $\mathbf{A}_{k+1}^{\text{gal}}$ ,  $\mathbf{A}_{k+1}^{\text{bc}}$ ,  $\mathbf{A}_{k+1}^{\text{gls}}$ 
  end while
  Create  $\mathbf{A}_m = \mathbf{A}_m^{\text{gls}} + \mathbf{A}_m^{\text{gls}} + \mathbf{A}_m^{\text{gls}}$  and compute LU factorization of  $\mathbf{A}_m$ 
  if  $|\mathbf{r}| > \text{machine tolerance}$  then
    Solve linear system using multigrid to specified tolerance
    Update solution
  end if
until  $|\mathbf{r}| < \text{machine tolerance}$ 

```

Eq. 4.23 now represents a large sparse block matrix system to which the linear multigrid algorithm can be applied. A modification of the Newton algorithm is required to prevent updates ($\Delta \mathbf{V}_h^n$) which would produce infeasible solutions such as negative density. This is done by implementing a damping procedure that limits individual components of the update vector. The solution algorithm for the non-linear system of equations may now be formally described by Algorithm 3

4.2.2 Implicit Line Creation Extension

The line creation algorithm for the line implicit smoother as described in Chap. 2 requires a scalar coupling matrix which represents the relative coupling between the nodal equations.

For a block matrix system where each matrix coefficient is a local block matrix, a suitable scalar for the corresponding coupling matrix coefficient is unclear. For the Euler equations, we have used two formulations for this coupling matrix.

Algebraic Reconstruction:

The first formulation is an attempt to reconstruct an entropy-based stationary linear convection equation by exploiting the symmetric formulation of the Euler equations. As discussed, the given choice of the scalar entropy function $U(\mathbf{U})$ results in the fluxes $\mathbf{F}_1(\mathbf{V})$ and $\mathbf{F}_2(\mathbf{V})$ being homogeneous functions of degree $q = -\frac{2\gamma}{\gamma-1}$. Using linear algebra, it can be shown that

$$\frac{d\mathbf{F}_i}{d\mathbf{V}} \mathbf{V} = q\mathbf{F}_i \quad i = \{1, 2\} \quad (4.28)$$

such that

$$\frac{d\mathbf{F}_i}{dx_i} = \frac{1}{q} \frac{d}{dx_i} \left[\frac{d\mathbf{F}_i}{d\mathbf{V}} \mathbf{V} \right] \quad (4.29)$$

$$= \frac{1}{q} \left[\frac{d\mathbf{F}_i}{d\mathbf{V}} \frac{d\mathbf{V}}{dx_i} + \frac{d}{d\mathbf{V}} \left(\frac{d\mathbf{F}_i}{dx_i} \right) \mathbf{V} \right] \quad (4.30)$$

$$= \frac{1}{q} \frac{d\mathbf{F}_i}{dx_i} + \frac{1}{q} \frac{d}{d\mathbf{V}} \left(\frac{d\mathbf{F}_i}{dx_i} \right) \mathbf{V} \quad (4.31)$$

$$\Rightarrow \frac{d\mathbf{F}_i}{dx_i} = \left(\frac{1}{q-1} \right) \frac{d}{d\mathbf{V}} \left(\frac{d\mathbf{F}_i}{dx_i} \right) \mathbf{V} \quad (4.32)$$

We may now construct the stationary convection equation for the scalar entropy function using Eq. 4.11 such that:

$$\begin{aligned} \frac{dF^i}{dx_i} &= \mathbf{V}^T \frac{d\mathbf{F}_i}{dx_i} \\ &= \left(\frac{1}{q-1} \right) \mathbf{V}^T \frac{d}{d\mathbf{V}} \left(\frac{d\mathbf{F}_i}{dx_i} \right) \mathbf{V} \end{aligned} \quad (4.33)$$

This represents a scaled inner product with respect to the Jacobian of the governing PDE. Using this idea, we attempt to reconstruct a discrete FEM formulation of Eq. 4.33 based on the discrete variational weak form of the Euler equations. The scalar matrix derived from this reconstruction process is chosen to be the weighting matrix for the line construction

algorithm and is defined by:

$$c_{i,j} = \{\mathbf{V}_i^T\} \left[\mathbf{J}_{i,j}^{gal} \right] \{\mathbf{V}_j\} \quad (4.34)$$

where $\mathbf{J}^{gal}$ is the Jacobian of the Galerkin term $B(\cdot, \cdot)_{gal}$. Any homogeneous differential equation consisting of derivatives only is satisfied by the constant function. This implies that a linearization $c_{i,j}$ of any discretization of the problem must satisfy the relation

$$\sum_j c_{i,j} = 0 \quad (4.35)$$

It is observed that in general, Eq. 4.35 is only satisfied at convergence.

Rediscretization:

The second formulation for the weighting matrix is based on a GLS rediscretization of the stationary linear convection equation (Eq. 3.9) using the velocity field $\vec{\mathbf{V}} = (u_1, u_2)$ from the current estimate of the velocity field computed during the Newton solution scheme. The line creation coupling matrix is defined based on the Galerkin terms from the resulting discretization.

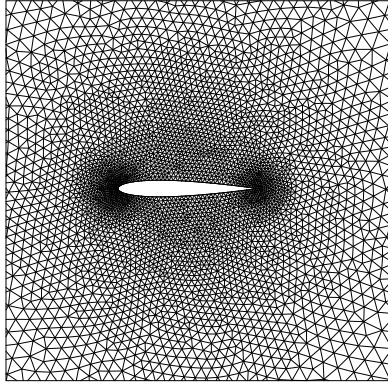
The coarse space definition of the coupling matrix is based on the GCA formulation. Given the weighting matrix $\mathbf{c}_k$, the coarse space definition for $\mathbf{c}_{k+1}$ is given by

$$\mathbf{c}_{k+1} = \mathbf{R}_k \mathbf{c}_k \mathbf{P}_k \quad (4.36)$$

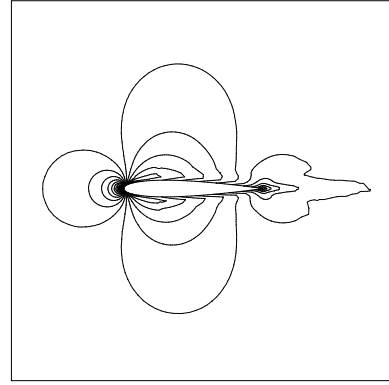
The quality of the implicit lines created by the two algorithms may be compared in Fig. 4.1 for the converged solution of the Euler equations. The problem is the flow around a NACA 0012 airfoil at a zero angle of attack and a freestream Mach number of 0.1. Fig. 4.1(c) shows the implicit lines created using the algebraic reconstruction while Fig. 4.1(d) shows the implicit lines created using a rediscretization at convergence. In both cases, no noticeable difference in the convergence rate was observed.

4.2.3 Interpolation Operator Extension

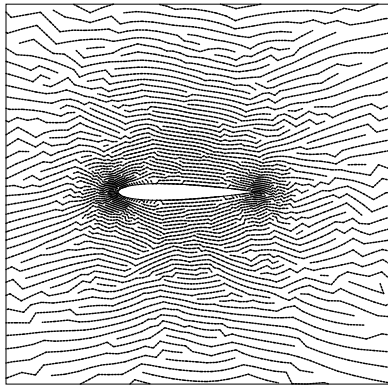
The extension of AMG to systems of equations where more than one function is being approximated does not present a clear choice for the definition of the interpolation operators. Cleary *et al* [123] and Ruge *et al* [44] discuss some of the associated issues. In [123],



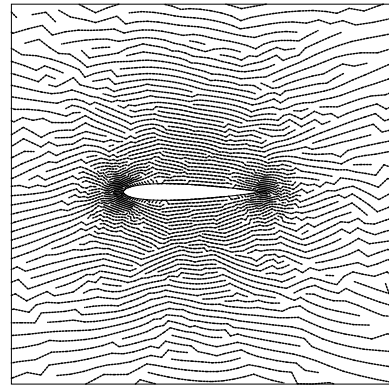
(a) Grid



(b) Solution



(c) Lines: Reconstruction



(d) Lines: Rediscretization

Figure 4.1: Implicit lines construction using the reconstruction and rediscretization schemes for inviscid flow over a NACA 0012 at Mach 0.1

the authors make use of a *function approach* where separate interpolation operators are defined for the functions. This corresponds to the *unknown approach* described by Ruge *et al* [44]. In our approach, we make use of the *point approach* described in [44] where the multigrid algorithm is applied in a block manner such that all variables corresponding to the same point are interpolated together. This implies the assumption that the partitioning of the error components associated with each variable into rough and smooth components is identical (Eq. 2.19).

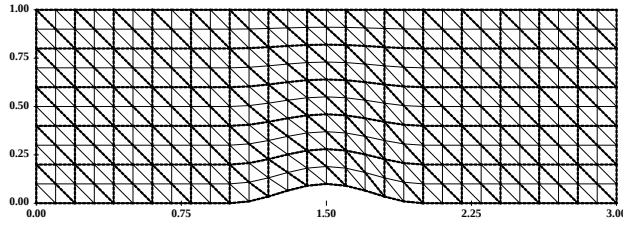
4.3 Consistency Scaling Issues for the Stabilized Euler Equations

As discussed earlier in Chap. 3, the application of AMG to stabilized FEM formulations implies that a rescaling of the coarse space representation of the discrete operator must be performed to ensure the consistency of the coarse space operator. Fig. 4.2 show the effect of τ scaling on a channel flow problem for the compressible Euler equations. The bump is a sine-squared bump of thickness 0.1, the inlet Mach number is 0.5 and the results show the eigenspectrum of the iteration matrix $-\mathbf{M}^{-1}\mathbf{A}$ where $\mathbf{A}$ is the residual Jacobian matrix after 3 Newton steps and $\mathbf{M}$ is the Jacobi line implicit preconditioning matrix.

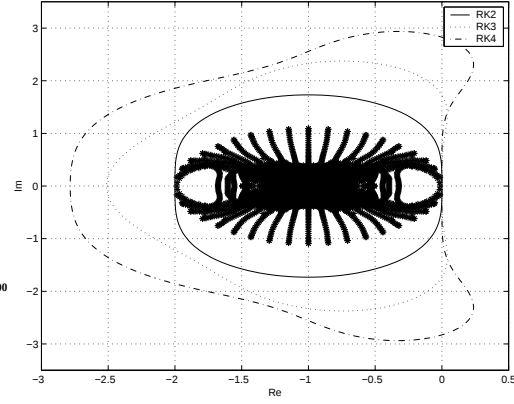
The effect of the stabilization scaling for a sequence of coarse grids is shown in Fig. 4.2. The stabilization scaling is clearly necessary to bound the eigenvalues on the coarser grids so that the line relaxation scheme behaves in the same manner on all the grid levels. Without the scaling, the deterioration of the spectral radius of the iteration matrix becomes rapidly worse with increasing coarse grid levels and the only choice left is to choose small damping factors in the multi-stage scheme resulting in poor convergence for the smoother.

4.4 Results

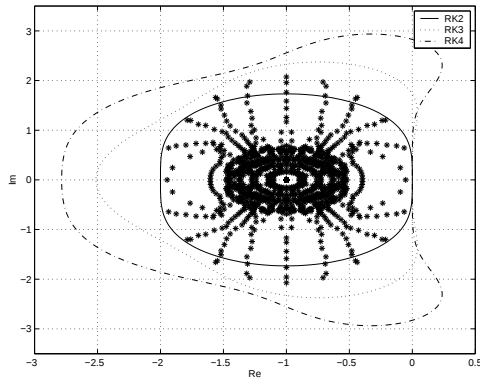
In order to demonstrate the effectiveness of the proposed multigrid scheme for the compressible Euler equations, we will consider two different test cases and conduct a grid dependency and Mach number dependency study.



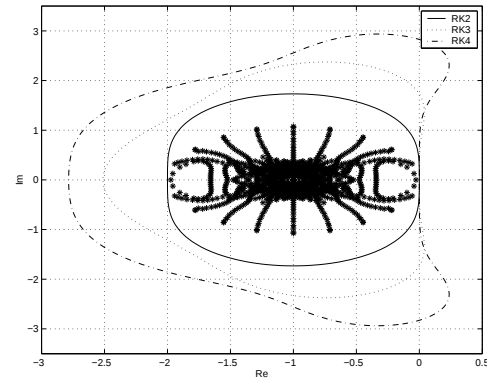
(a) Fine Grid and Agglomerated Coarse Grid



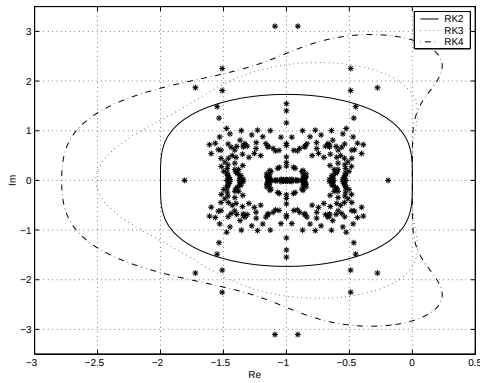
(b) Fine Grid



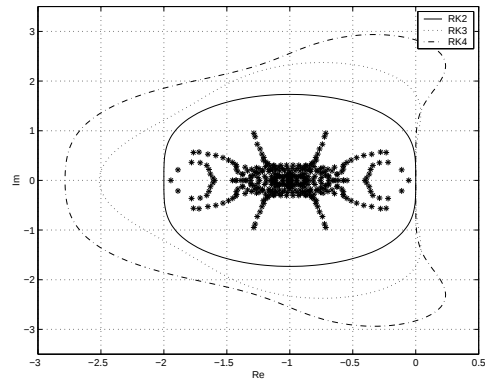
(c) Coarse Grid 1: w/o Scaling



(d) Coarse Grid 1: w/ Scaling



(e) Coarse Grid 2: w/o Scaling



(f) Coarse Grid 2: w/ Scaling

Figure 4.2: τ scaling effect on line Jacobi iteration matrix eigenspectrum for channel problem at freestream Mach number $M = 0.1$

This figure was broken... original no longer exists.

Figure 4.3: Channel Flow Geometry for Euler Equations

4.4.1 Channel Flow

The first case consists of a channel flow as shown in Fig. 4.3 where the shape of the bump on the lower wall is a sine-squared bump with a thickness ratio of 0.05. The domain is discretized into a structured grid by using a uniform grid spacing in the x -direction and evenly dividing the channel height at any given x -location. The resulting grid is then triangulated. The implicit line smoother considered is a damped 3-stage symmetric line Gauß-Seidel smoother with a relaxation factor of $\omega = 0.7$. The multistage coefficients chosen are the optimized 3-stage coefficients in Table 3.1.

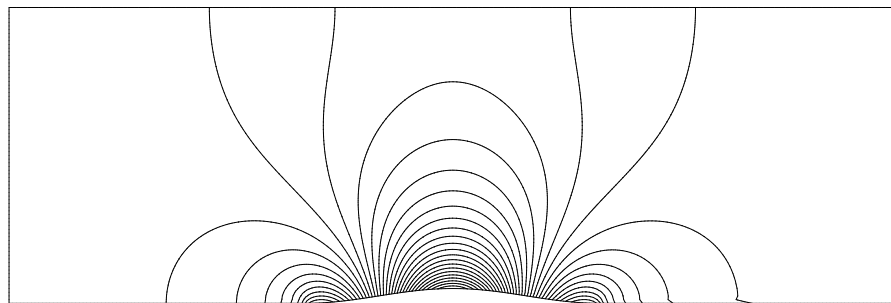


Figure 4.4: Mach Number Contours for $M = 0.1$ Compressible Euler 241×81 Bump Problem

On each grid, a $V(2,1)$ multigrid cycle is implemented and due to the smoothness of the solution, no initial time damping (Eq. 4.23) is required. Table 4.1 shows the asymptotic

convergence rate ϵ of the line solver for a number of fine grid sizes and Mach numbers. The agglomeration algorithm results in full coarsening such that the choice for the number of coarse grids results in the same number of elements and vertices on the coarsest grid. As can be observed from Table 4.1, very good convergence rates are achieved.

Fine Grid Size	# of coarse grids	ϵ : M = 0.1	ϵ : M = 0.5
31×11	1	0.04	0.06
61×21	2	0.04	0.06
121×41	3	0.04	0.07
241×81	4	0.06	0.08

Table 4.1: Compressible Euler Bump Results

4.4.2 Airfoil Flow

The second test case consists of external flow around a NACA 0012 airfoil at angle of attack of $\alpha = 0^\circ$ and $\alpha = 3^\circ$ as shown in Fig. 4.5. The discretization of the fine grid mesh is completely unstructured. The implicit line smoother considered is also a damped n-stage symmetric line Gauß-Seidel smoother with a relaxation factor of $\omega = 0.9$ and the same multistage coefficients as chosen in the channel flow problem. Both a 3-stage and a 5-stage scheme for this test case was chosen in order to compare the convergence rates for both schemes and as well as to make a fair comparison later on with the Navier-Stokes test cases which utilize a 5-stage scheme. The multistage coefficients chosen are the optimized 3-stage and 5-stage coefficients in Table 3.1. On each fine grid problem, a V(2,1) multigrid cycle is implemented and no initial time damping (Eq. 4.23) is also required.

The termination condition for the linear multigrid solver is chosen to be when the linear RMS residual is the square of the non-linear residual while the termination condition for the non-linear Newton method is when the RMS value of the non-linear residual is less than 10^{-14} . This ensures that Newton quadratic convergence is achieved without always requiring a linear multigrid solution to machine zero.

Table 4.2 shows the asymptotic convergence rate ϵ for a number of fine grid sizes and Mach numbers as well as the total element complexity (TEC) and total vertex complexity (TVC) over all the grids. Given the number of vertices n_k^v and elements n_k^e per grid level,

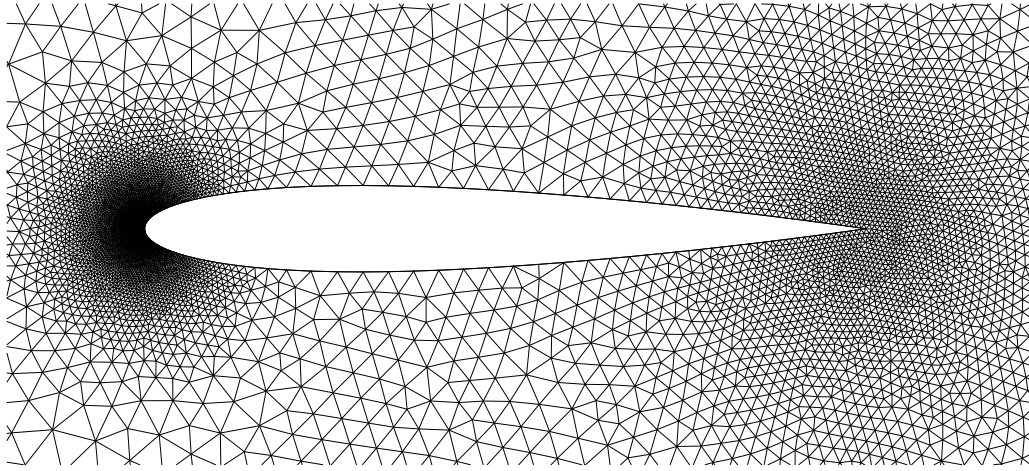


Figure 4.5: Unstructured Mesh for NACA 0012 Foil

these complexities for an m-grid (total of fine + coarse) problem are calculated as

$$\text{TVC} = \sum_{k=0}^m \frac{n_k^v}{n_0^v} \quad (4.37)$$

$$\text{TEC} = \sum_{k=0}^m \frac{n_k^e}{n_0^e} \quad (4.38)$$

Fine Grid Size	Coarse Grids	TVC	TEC	$\epsilon_{M=0.1}$ ($\alpha = 0^\circ$)		$\epsilon_{M=0.5}$ ($\alpha = 0^\circ$)		$\epsilon_{M=0.5}$ ($\alpha = 3^\circ$)	
				3 Stage	5 stage	3 Stage	5 stage	3 Stage	5 stage
2607	2	1.58	1.34	0.26	0.19	0.24	0.18	0.28	0.18
5258	3	1.64	1.36	0.31	0.22	0.34	0.24	0.25	0.17
10273	4	1.67	1.36	0.28	0.17	0.25	0.24	0.24	0.17
20621	5	1.64	1.36	0.33	0.26	0.34	0.21	0.23	0.21

Table 4.2: Asymptotic linear multigrid convergence rates for compressible Euler flow over a NACA 0012 airfoil

The number of coarse grids chosen was such that the number of vertices on the coarsest grid is less than 500. As can be observed from Table 4.2, excellent and relatively mesh independent convergence rates are achieved. Fig. 4.7 shows the non-linear Newton and

linear multigrid solver residual history for the compressible Euler computation on the 20,621 node fine grid for a freestream Mach number of 0.5 and a 5-stage multistage scheme.

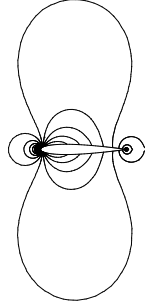


Figure 4.6: Mach Number contours for freestream $M = 0.1$ compressible Euler flow over a NACA 0012 airfoil at zero angle of attack

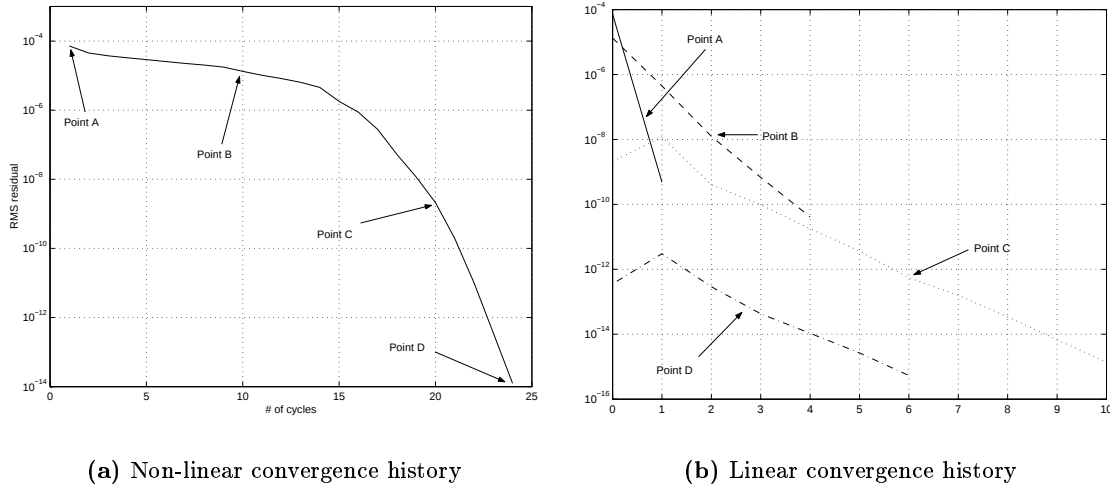


Figure 4.7: Non-linear Newton outer loop and linear multigrid convergence histories for compressible Euler flow over a NACA 0012 airfoil with 20,621 fine grid nodes at Mach 0.5 and zero angle of attack using a 5-stage scheme

As a point of comparison, Fig. 4.8 shows the convergence history for the inviscid computation of a compressible flow by Pierce [1] over a NACA 0012 airfoil at freestream Mach number $M = 0.5$ and angle of attack $\alpha = 3^\circ$. The discretization scheme is a conservative cell-

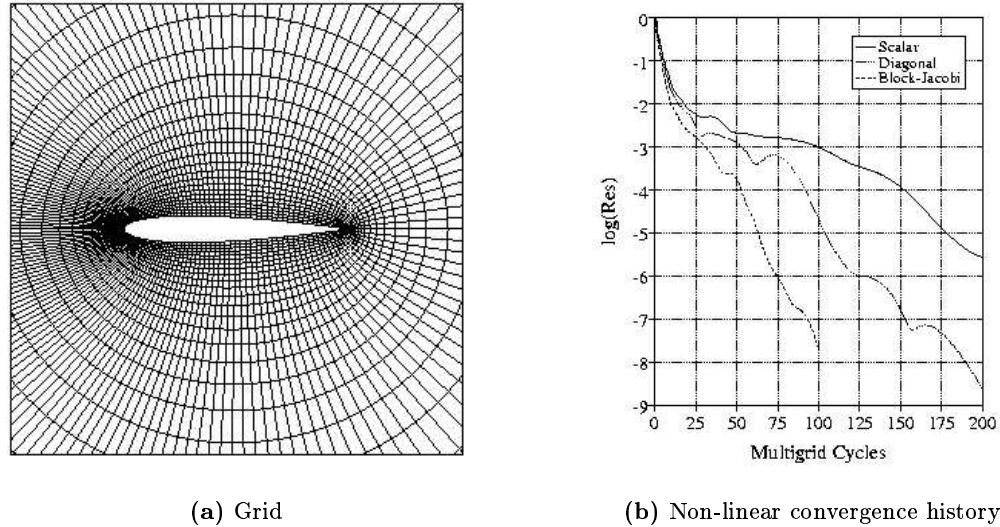


Figure 4.8: Multigrid results by Pierce *et al* [1] for inviscid flow over NACA 0012 airfoil at freestream Mach number $M = 0.5$, $\alpha = 3^\circ$ on a 160×32 O-mesh using scalar, diagonal and block-Jacobi preconditioning

centered semi-discrete Finite Volume scheme for structured grids which uses a characteristic based matrix dissipation. A full-coarsening FAS multigrid solution scheme is implemented using a $W(1,0)$ cycle and 5-stage multistage formulation. Fig. 4.8(a) shows the grid used for the inviscid calculation and Fig. 4.8(b) shows the multigrid convergence history using point implicit scalar, diagonal and block-Jacobi preconditioning. The best achievable convergence rate for this problem is around 0.8 while Mavriplis reports rates of around 0.75 [124]. For the same conditions, the rate achieved by the current code using a $W(1,0)$ cycle, a 5-stage multistage scheme, a relaxation factor of 1.0 and regular sweeps i.e no symmetric sweeps as were reported in Table 4.2 is 0.55.

Chapter 5

Navier Stokes Applications

The severity of the problems associated with discrete stiffness as well as directional decoupling for inviscid computations is not as pronounced as in viscous calculations. The use of standard multigrid algorithms with full coarsening and point implicit smoothers results in a significant deterioration in the multigrid convergence rates which worsens with increasing anisotropy [39]. Much of the recent research into the construction of robust multigrid algorithms for the Navier-Stokes equations has been done within the context of Preconditioned Multigrid Methods [13, 25, 26, 70, 125] using semi-coarsening. Pierce shows that the use of a Jacobi-preconditioning smoother for structured discretizations of the turbulent compressible Navier-Stokes equations with J-coarsening, where the grid is only coarsened in the J (*viscous*) direction, leads to an improvement in the convergence rate over standard multigrid algorithms employing scalar preconditioning [39]. Allmaras [25] and Venkatakrishnan [26] also compare the convergence rates, using several preconditioners, for the turbulent Navier-Stokes equations. The results demonstrate that line preconditioning is a viable scheme. In the context of unstructured meshes, Mavriplis [6, 71, 105] has implemented a semi-coarsening scheme based on a nodal agglomeration technique and an implicit line relaxation scheme, and has demonstrated multigrid convergence rates similar to those obtained by Pierce [39] in a structured mesh context.

In this chapter, we consider the application of the proposed AMG algorithm to a stabilized Finite Element discretization of the Navier-Stokes equations and numerical studies are performed to determine the behavior of the algorithm for airfoil flow problems.

5.1 FEM Discretization

The discretization for the Navier-Stokes equation is also based on the Finite Element code provided by Wong *et al* and is an extension of the inviscid FEM code as described in Sec. 4.1. Let us consider the time dependent 2D compressible Navier-Stokes equations in conservative form

$$\mathbf{U}_{,t} + \mathbf{F}_{1,1} + \mathbf{F}_{2,2} = \mathbf{F}_{1,1}^v + \mathbf{F}_{2,2}^v, \quad (5.1)$$

where

$$\mathbf{U} = \begin{Bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho E \end{Bmatrix}, \quad (5.2)$$

$$\mathbf{F}_1 = \begin{Bmatrix} \rho u_1 \\ \rho u_1^2 + p \\ \rho u_1 u_2 \\ u_1(\rho E + p) \end{Bmatrix}, \quad \mathbf{F}_2 = \begin{Bmatrix} \rho u_2 \\ \rho u_1 u_2 \\ \rho u_2^2 + p \\ u_2(\rho E + p) \end{Bmatrix} \quad (5.3)$$

and

$$\mathbf{F}_1^v = \begin{Bmatrix} 0 \\ \tau_{11} \\ \tau_{12} \\ u_1 \tau_{11} + u_2 \tau_{12} + q_1 \end{Bmatrix}, \quad \mathbf{F}_2^v = \begin{Bmatrix} 0 \\ \tau_{21} \\ \tau_{22} \\ u_1 \tau_{21} + u_2 \tau_{22} + q_2 \end{Bmatrix}.$$

such that $\mathbf{F}_i$ are the inviscid fluxes; $\mathbf{F}_i^v$ are the viscous fluxes; ρ is the density; $\mathbf{u} = \{u_1, u_2\}^T$ is the velocity vector; E is the specific total energy and p is the pressure. The system of equations is closed through the equation of state, $p = (\gamma - 1)\rho e$, where $e = E - |\mathbf{u}|^2/2$ is the internal energy. Here γ is the ratio of specific heats and μ is the absolute viscosity, both of which are assumed to be constant. We may now non-dimensionalize the above variables

using reference values for density (ρ^*), velocity (u^*), length (L) and viscosity (μ^*) via

$$\begin{aligned}\bar{\rho} &= \frac{\rho}{\rho^*}, \quad \bar{u}_i = \frac{u_i}{u^*}, \quad i = 1, 2, \quad \bar{p} = \frac{p}{\rho^* u^{*2}}, \quad \bar{E} = \frac{E}{u^{*2}}, \\ \bar{\mu} &= \frac{\mu}{\mu^*}, \quad \bar{x}_i = \frac{x_i}{L}, \quad i = 1, 2, \quad \text{and} \quad \bar{t} = \frac{u^* t}{L}.\end{aligned}$$

The auxiliary equations may now be written as

$$\tau_{11} = 2 \frac{\mu}{\text{Re}} \frac{\partial u_1}{\partial x_1} - \frac{2}{3} \frac{\mu}{\text{Re}} \left(\frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} \right)$$

$$\tau_{12} = \tau_{21} = \frac{\mu}{\text{Re}} \left(\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1} \right)$$

$$\tau_{22} = 2 \frac{\mu}{\text{Re}} \frac{\partial u_2}{\partial x_2} - \frac{2}{3} \frac{\mu}{\text{Re}} \left(\frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} \right)$$

$$q_1 = -\frac{1}{\text{Re}} \frac{1}{\text{Pr}} \mu \frac{\partial T}{\partial x_1}, \quad q_2 = -\frac{1}{\text{Re}} \frac{1}{\text{Pr}} \mu \frac{\partial T}{\partial x_2}.$$

where the overbars denoting the non-dimensional variables have been dropped. There are two resulting non-dimensional parameters which are the Reynolds number Re , and the Prandtl number Pr , defined as:

$$\begin{aligned}\text{Re} &= \frac{\rho^* u^* L}{\mu^*} \\ \text{Pr} &= \frac{\mu^* C_p}{\kappa}\end{aligned}$$

where C_p is the specific heat at constant pressure and κ is the heat conductivity coefficient, both assumed constant. Equation (5.1) can be written in linearized form as

$$\mathbf{U}_{,t} + \mathbf{A}_1 \mathbf{U}_{,1} + \mathbf{A}_2 \mathbf{U}_{,2} = (\mathbf{K}_{11} \mathbf{U}_{,1})_{,1} + (\mathbf{K}_{12} \mathbf{U}_{,2})_{,1} + (\mathbf{K}_{21} \mathbf{U}_{,1})_{,2} + (\mathbf{K}_{22} \mathbf{U}_{,2})_{,2}, \quad (5.4)$$

where the inviscid Jacobian matrices $\mathbf{A}_i = \mathbf{F}_{i,\mathbf{U}}$, $i = 1, 2$, are unsymmetric but have real eigenvalues and a complete set of eigenvectors and $\mathbf{K}_{ij} = \mathbf{F}_{i,\mathbf{U},j}^v$ are the viscous flux Jacobians. Equation (5.4) is symmetrized by the introduction of entropy variables $\mathbf{V}$ [115–117],

such that the change $\mathbf{U} = \mathbf{U}(\mathbf{V})$ applied to (5.1) gives the transformed system

$$\mathbf{U}(\mathbf{V})_{,t} + \mathbf{F}_1(\mathbf{V})_{,1} + \mathbf{F}_2(\mathbf{V})_{,2} = \mathbf{F}_1^v(\mathbf{V})_{,1} + \mathbf{F}_2^v(\mathbf{V})_{,2}, \quad (5.5)$$

or equivalently in symmetric quasi-linear form as

$$\mathbf{A}_0 \mathbf{V}_{,t} + \tilde{\mathbf{A}}_1 \mathbf{V}_{,1} + \tilde{\mathbf{A}}_2 \mathbf{V}_{,2} = (\tilde{\mathbf{K}}_{11} \mathbf{V}_{,1})_{,1} + (\tilde{\mathbf{K}}_{12} \mathbf{V}_{,2})_{,1} + (\tilde{\mathbf{K}}_{21} \mathbf{V}_{,1})_{,2} + (\tilde{\mathbf{K}}_{22} \mathbf{V}_{,2})_{,2}, \quad (5.6)$$

where $\mathbf{A}_0 = \mathbf{U}_{,\mathbf{V}}$ is symmetric positive definite, and $\tilde{\mathbf{A}}_i = \mathbf{A}_i \mathbf{A}_0 = \mathbf{F}_{i,\mathbf{V}}$, $i = 1, 2$, are symmetric. Following the discussion on the Euler equations discretization in Sec. 4.1, we introduce a scalar entropy function $H(\mathbf{U}) = -\rho g(s)$, where s is the non-dimensional entropy $s = \ln(p/\rho^\gamma)$. For the Navier-Stokes equations, if we chose $g(s) = \frac{s}{\gamma-1}$ then $\mathbf{V}$ takes the form:

$$\mathbf{V} = \begin{bmatrix} \frac{\gamma+1-s}{\gamma-1} - \frac{\rho E}{p} \\ \frac{\rho u_1}{p} \\ \frac{\rho u_2}{p} \\ -\frac{\rho}{p} \end{bmatrix} \quad (5.7)$$

such that the matrix $\tilde{\mathbf{K}}$

$$\tilde{\mathbf{K}} = \begin{bmatrix} \tilde{\mathbf{K}}_{11} & \tilde{\mathbf{K}}_{12} \\ \tilde{\mathbf{K}}_{21} & \tilde{\mathbf{K}}_{22} \end{bmatrix}$$

is symmetric as well as positive semi-definite.

Let us now consider the variational formulation for the steady state problem. The problem is defined in a domain Ω with boundary Γ by

$$\mathbf{F}_1(\mathbf{V})_{,1} + \mathbf{F}_2(\mathbf{V})_{,2} = \mathbf{F}_1^v(\mathbf{V})_{,1} + \mathbf{F}_2^v(\mathbf{V})_{,2} \quad \text{in } \Omega, \quad (5.8)$$

$$\tilde{\mathbf{A}}_n^- \mathbf{V} = \tilde{\mathbf{A}}_n^- \mathbf{g} \quad \text{on } \Gamma \setminus \Gamma_w, \quad (5.9)$$

$$\mathbf{F}^v \cdot \mathbf{n} = \mathbf{f} \quad \text{on } \Gamma_w \quad (5.10)$$

where the domain boundary is made up of an impermeable solid wall Γ_w , and a computational far field boundary $\Gamma \setminus \Gamma_w$. In (5.9, 5.10), $\mathbf{n} = [n_1, n_2]^T$ is the outward unit normal vector to Γ , and $\tilde{\mathbf{A}}_n = \mathbf{A}_n \mathbf{A}_0$, $\mathbf{A}_n = \mathbf{A}_1 n_1 + \mathbf{A}_2 n_2$. Finally, $\tilde{\mathbf{A}}_n^- = \mathbf{A}_n^- \mathbf{A}_0$, and $\mathbf{A}_n^-$ denotes the negative definite part of $\mathbf{A}_n$. Let the spatial domain Ω , be discretized into non-overlapping elements T_e , such that $\Omega = \bigcup T_e$, and $T_e \cap T_{e'} = \emptyset$, $e \neq e'$. We consider

the space of functions $\mathcal{V}_h$, defined over the discretization and consisting of the continuous functions which are piecewise linear over each element

$$\mathcal{V}_h = \{\mathbf{W} \mid \mathbf{W} \in (C^0(\Omega))^4, \mathbf{W}|_{T_e} \in (\mathcal{P}_1(T_e))^4, \forall T_e \in \Omega\}.$$

The GLS algorithm can then be written as:

Find $\mathbf{V}_h \in \mathcal{V}_h$ such that for all $\mathbf{W} \in \mathcal{V}_h$,

$$\mathbf{r}(\mathbf{V}_h, \mathbf{W}) = B(\mathbf{V}_h, \mathbf{W})_{gal} + B(\mathbf{V}_h, \mathbf{W})_{gl s} + B(\mathbf{V}_h, \mathbf{W})_{bc} = 0, \quad (5.11)$$

where the forms $B(\cdot, \cdot)_{gal}$, $B(\cdot, \cdot)_{gl s}$ and $B(\cdot, \cdot)_{bc}$ account for the Galerkin, GLS stabilization, and boundary condition terms respectively, and are defined as

$$B(\mathbf{V}_h, \mathbf{W})_{gal} = \int_{\Omega} (-\mathbf{W}_{,1} \cdot (\mathbf{F} - \mathbf{F}^v)_1(\mathbf{V}_h) - \mathbf{W}_{,2} \cdot (\mathbf{F} - \mathbf{F}^v)_2(\mathbf{V}_h)) d\Omega, \quad (5.12)$$

$$B(\mathbf{V}_h, \mathbf{W})_{gl s} = \int_{\Omega} \{(\mathbf{F} - \mathbf{F}^v)_1(\mathbf{W}) + (\mathbf{F} - \mathbf{F}^v)_2(\mathbf{W})\} \cdot \boldsymbol{\tau} \{(\mathbf{F} - \mathbf{F}^v)_1(\mathbf{V}_h) + (\mathbf{F} - \mathbf{F}^v)_2(\mathbf{V}_h)\} d\Omega, \quad (5.13)$$

$$B(\mathbf{V}_h, \mathbf{W})_{bc} = \int_{\Gamma \setminus \Gamma_a} \mathbf{W} \cdot (\mathbf{F}_{ff} + \mathbf{F}^v)(\mathbf{V}_h, \mathbf{g}; \mathbf{n}) ds + \int_{\Gamma_a} \mathbf{W} \cdot \mathbf{F}^v(\mathbf{V}_h, \mathbf{f}; \mathbf{n}) ds. \quad (5.14)$$

where $\boldsymbol{\tau}$ is the stabilization matrix which must be symmetric, positive definite, have dimensions of time and scale linearly with the element size. The current implementation in the code for $\boldsymbol{\tau}$ is based on the following modification for viscous simulations:

$$\boldsymbol{\tau}^{-1} = \boldsymbol{\tau}_i^{-1} + \boldsymbol{\tau}_v^{-1} \quad (5.15)$$

where $\boldsymbol{\tau}_i$ is the inviscid stabilization matrix defined in [114] and $\boldsymbol{\tau}_v$ is a viscous modification defined as

$$\boldsymbol{\tau}_v^{-1} = 2(\tilde{\mathbf{K}}_{11} + \tilde{\mathbf{K}}_{22})/h_e^2. \quad (5.16)$$

where h_e is chosen to be the length of the shortest edge of the element.

For the Navier-Stokes equations, no-slip Dirichlet boundary conditions replace the boundary integral over the solid wall. The numerical flux function on the far field boundary $\mathbf{F}_{ff}$, is defined by

$$\mathbf{F}_{ff}(\mathbf{V}_h^-, \mathbf{V}_h^+; \mathbf{n}) = \frac{1}{2}(\mathbf{F}_n(\mathbf{V}_h^-) + \mathbf{F}_n(\mathbf{V}_h^+)) - \frac{1}{2}|\mathbf{A}_n(\mathbf{V}_h^{Roe}(\mathbf{V}_h^-, \mathbf{V}_h^+))|(\mathbf{U}(\mathbf{V}_h^+) - \mathbf{U}(\mathbf{V}_h^-)).$$

Here, $|\mathbf{A}_n(\mathbf{V}_h)| = \mathbf{A}_n^+(\mathbf{V}_h) - \mathbf{A}_n^-(\mathbf{V}_h)$ is the absolute value of $\mathbf{A}_n$ evaluated at $\mathbf{V}_h$, and $\mathbf{V}_h^{Roe}(\mathbf{V}_h^+, \mathbf{V}_h^-)$, is the Roe average [120], between the states $\mathbf{V}_h^+$ and $\mathbf{V}_h^-$.

5.2 Limitations of Current FEM Implementation

Some limitations regarding the application of the multigrid algorithm for Navier-Stokes simulations exist based on the current FEM formulation. First of all, Eq. 5.15 is an ad-hoc viscous modification by Wong *et al* of a previously described inviscid stabilization matrix [114] and it appears to have the wrong limiting behavior with the Reynolds number. The effect of this is that an incorrect amount of stabilization is applied especially in the boundary layer region. An eigenspectrum decomposition of the iteration matrix $-\mathbf{M}^{-1}\mathbf{A}$ as defined in Sec. 3.2 will show that there is a rapid growth in the eigenvalues which can not be easily bounded in the stability region of the iterative scheme. This behavior becomes increasingly pronounced for larger grid sizes and cell stretching ratios. However, an implementation of a viscous stabilization matrix which scales properly with local Reynolds number has been done by Shakib *et al* [91, 96, 126].

Currently, turbulence modeling has not been implemented in the FEM formulation. Research into the implementation of Reynolds Averaged Navier-Stokes (RANS) within a Finite Element context has been rather limited to date. However, Hauke *et al* [127] report a RANS formulation augmented with a turbulent transport equation for stabilized FEM formulations. Also, implementations of Large Eddy Simulation (LES) turbulence modeling within a GLS/FEM context has been done by Jansen [128] for unstructured grids and parallelized [129–131].

In addition, a shock capturing scheme for discontinuities has not been incorporated into the current GLS/FEM formulation of the discrete equations. However, Barth [116] describes a discontinuity capturing operator for stabilized GLS/FEM by Galeão *et al* [132]. Hughes *et al* [93, 133] also describe a discontinuity capturing operator for stabilized GLS/FEM.

For these reasons, the Reynolds and Mach number at which any test case may be run is limited to the subsonic, laminar regime.

5.3 AMG Extension to the Navier-Stokes Equations

The extension of the proposed AMG algorithm to the Navier-Stokes equations closely follows the Euler equations extension as described in Sec. 4.2. The solution strategy employed for the non-linear system of equations is the damped Newton scheme described in Sec. 4.2.1 where an exact linearization for the Jacobian matrix is performed. The only difference between the application of the proposed AMG algorithm to the Navier-Stokes and Euler equations is in the definition of the implicit line creation coupling matrix and the implementation of the no-slip Dirichlet boundary condition.

5.3.1 Implicit Line Creation Extension

The definition of the implicit line creation coupling matrix for the Navier-Stokes equations can no longer be made using the algebraic construction technique described in Sec. 4.2.2 for the Euler equations, based on the choice of the scalar entropy function $H(\mathbf{U})$. The fluxes are no longer homogeneous in $\mathbf{V}$ such that the simplified form of the coupling matrix coefficients, Eq. 4.34, is not valid. Application of the algebraic reconstruction technique to the viscous equations results in implicit lines which follow streamlines in the boundary layer regions during the initial convergence phase. This is followed by the eventual formation of lines which are normal to the solid wall boundary at convergence. This region is typically characterized by highly anisotropic cells which means that there exist error modes that will be strongly aligned normal to the solid wall boundary. The use of implicit lines which are aligned with the streamlines results in a significant degradation of the convergence rate of the smoother.

Due to this phenomenon, we choose to rediscretize the stationary linear convection-diffusion equation Eq. 3.1 using the velocity field $\vec{\mathbf{V}} = (u_1, u_2)$ from the current solution estimate. The value for the diffusion coefficient is chosen a-priori such that the implicit lines in the boundary layer regions which are normal to the solid boundaries cover the full extent of the boundary layer.

5.3.2 Dirichlet Boundary Condition Extension

The imposition of a no-slip velocity condition at a solid wall boundary means that the interpolation operators as well as the momentum equations need to be modified on the

Dirichlet boundaries as outlined in Sec. 2.5. On the solid walls, the two momentum equations are replaced by the Dirichlet conditions

$$u_1 = 0 \quad (5.17)$$

$$u_2 = 0 \quad (5.18)$$

The modification of the prolongation operator is done by expanding the scalar operator so that each entry in the matrix is replaced by a local 4×1 vector where each entry is set to the original scalar value. The second and third rows are then simply zeroed out.

5.4 Results

In this chapter, the proposed multigrid algorithm is applied to laminar airfoil flow. The test case chosen is flow around a NACA 0005 airfoil at angle of attack of $\alpha = 0^\circ$ and $\alpha = 3^\circ$ as shown in Fig. 5.1 where the largest cell aspect ratio is of the order of 500.

In order to expand the stability region, the relaxation scheme considered is a damped 5-stage symmetric line Gauß-Seidel smoother with a relaxation factor of $\omega = 0.5$. The multistage coefficients chosen are the optimized 5-stage coefficients by Lynn [104] for Roe's $\kappa = 0$ scheme and semi-coarsening as listed in Table 3.1. On each grid, a V(2,1) multigrid cycle is implemented and in all the test cases, time damping was required to advance the solution out of the initial non-linear stage due to the unstable growth of the iteration matrix eigenvalues. Hence, the presented results are based on the algorithm behavior in the Newton quadratic convergence region.

As with the Euler test cases, the termination condition for the linear multigrid solver is chosen to be when the linear RMS residual is the square of the non-linear residual while the termination condition for the non-linear Newton method is when the RMS value of the non-linear residual is less than 10^{-14} . This ensures that Newton quadratic convergence is achieved without always requiring a linear multigrid solution to machine zero. The number of coarse grids chosen for these test cases was such that stability of the iterative scheme is maintained on the coarsest grids. The stabilization rescaling described in Sec. 3.6 assumes that the elements are reasonably isotropic such that a well defined length scale may be obtained. The high mesh anisotropy in the boundary layer region results in an inexact stabilization scaling which is manifested in the instability of the iterative scheme on the

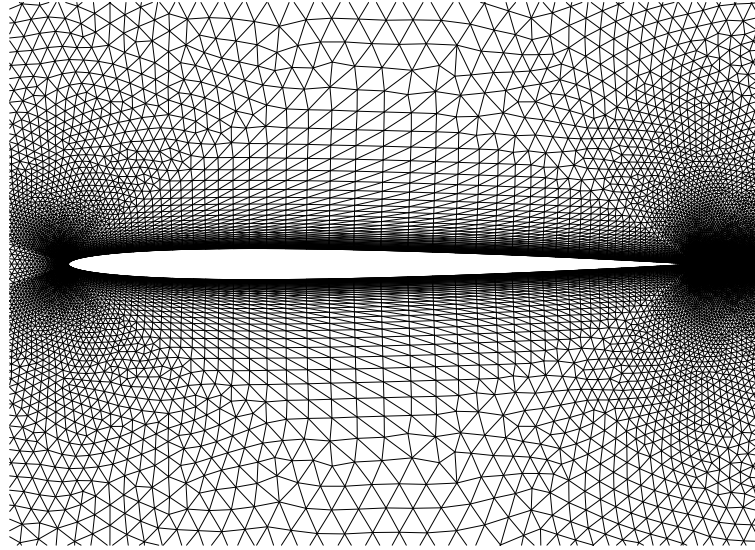


Figure 5.1: Unstructured Mesh for NACA 0005 Foil

coarsest meshes as the number of coarse grids is increased. Table 5.1 shows the asymptotic convergence rate ϵ as well as the total element complexity (TEC) and total vertex complexity (TVC) as described in Sec. 4.4.2 over all the grids for a sequence of independently generated fine grid sizes and Mach numbers at a Reynolds number of 5000. As can be observed from Table 5.1, excellent convergence rates are achieved.

Fine Grid Size	Coarse Grids	TVC	TEC	$\epsilon_{M=0.1}$ ($\alpha = 0^\circ$)	$\epsilon_{M=0.5}$ ($\alpha = 0^\circ$)	$\epsilon_{M=0.5}$ ($\alpha = 3^\circ$)
8872	2	1.64	1.37	0.16	0.37	0.21
18416	3	1.88	1.44	0.25	0.37	0.57
36388	4	1.67	1.37	0.35	0.35	0.45

Table 5.1: Asymptotic linear multigrid convergence rates for compressible Navier-Stokes flow over a NACA 0005 airfoil at $Re=5000$

Fig. 5.2 shows the non-linear Newton and linear multigrid solver residual history for the compressible Navier-Stokes computation on the 36,388 node fine grid for a freestream Mach number of 0.5, zero angle of attack and a 5-stage multistage scheme. For this particular case, the pseudo-time stepping scheme described in Sec. 4.2.1 is implemented with an exponentially decreasing CFL number as described by Eq. 4.27. After 35 non-linear Newton iterations, the time damping term is removed with a corresponding jump in the residual as shown in Fig. 5.2 followed by Newton convergence.

The convergence rates for the lifting ($\alpha = 3^\circ$) airfoil case shows some degradation and this may be explained by the interaction of the flow misalignment and the stabilization scaling for semi-coarsening with anisotropic elements. The inexact stabilization scaling results in an inappropriate amount of stabilization which is manifested in a degradation of the two-grid convergence rate on the two coarsest grids, with subsequent pollution of the total multigrid convergence rate.

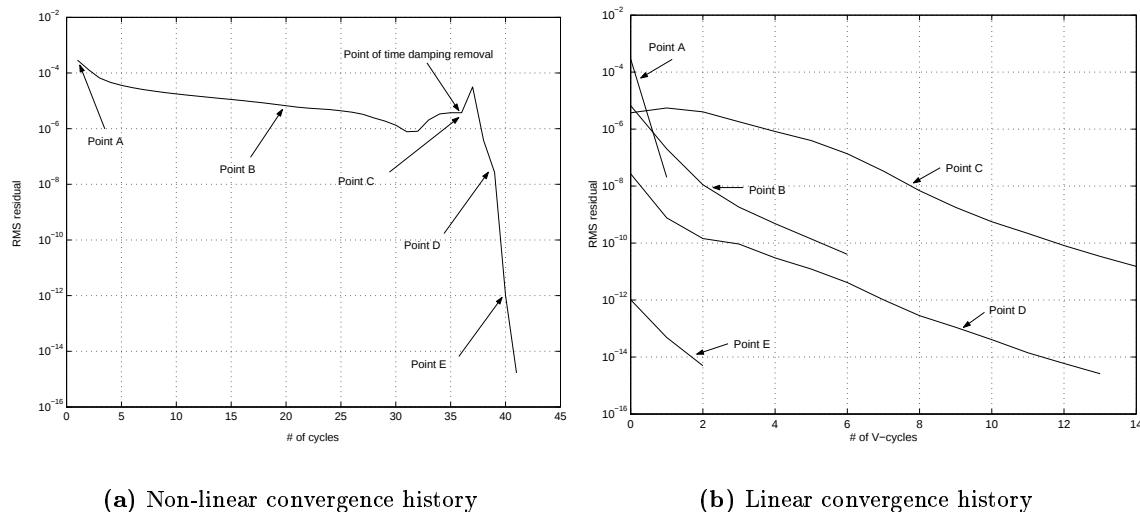
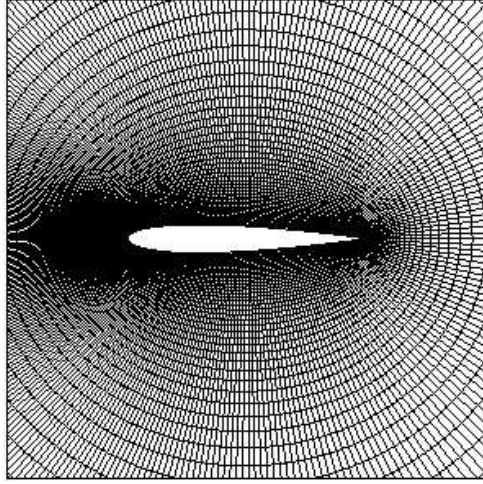
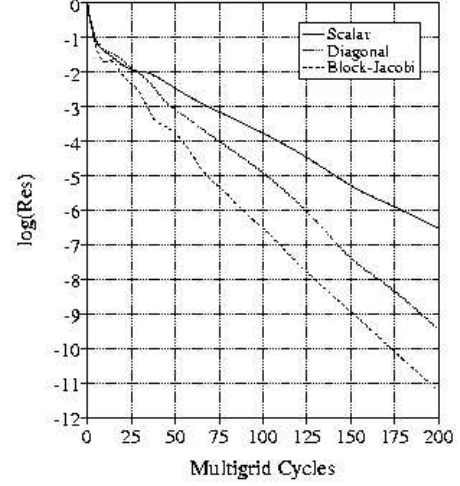


Figure 5.2: Non-linear Newton outer loop and linear multigrid convergence histories for compressible Navier-Stokes flow over a NACA 0005 airfoil at Mach 0.5 and zero angle of attack with 36,388 fine grid nodes and Reynolds number of 5000 using a 5-stage scheme

As a point of comparison, Fig. 5.3 shows the convergence history for the viscous computation of a compressible flow by Pierce [1] over a NACA 0012 airfoil at freestream Mach number $M = 0.5$, angle of attack $\alpha = 0^\circ$ and Reynolds number of 5000. The discretization



(a) Grid



(b) Non-linear convergence history

Figure 5.3: Multigrid results by Pierce *et al* [1] for viscous flow over NACA 0012 airfoil at freestream Mach number $M = 0.5$, $\alpha = 0^\circ$ and Reynolds number of 5000 on a 320×64 O-mesh using scalar, diagonal and block-Jacobi pre-conditioning

scheme is a conservative cell-centered semi-discrete Finite Volume scheme for structured grids which uses a characteristic based matrix dissipation. A J-coarsening FAS multigrid solution scheme is implemented using a $W(1,0)$ cycle and 5-stage multistage formulation. Fig. 5.3(a) shows the grid used for the viscous calculation and Fig. 5.3(b) shows the multigrid convergence history using point implicit scalar, diagonal and block-Jacobi preconditioning. The best achievable convergence rate for this problem is around 0.91. For the same conditions, the rate achieved by the current code using a $W(1,0)$ cycle, a 5-stage multistage scheme, a relaxation factor of 0.7 and regular sweeps i.e no symmetric sweeps as were reported in Table 5.1 is 0.81.

Chapter 6

Conclusions

6.1 Summary

In this thesis, a multigrid formulation has been proposed, analyzed and implemented for both Euler and Navier-Stokes applications for stabilized Finite Element discretizations. The success of the formulation is based on a proper choice of the multigrid components with respect to the characteristics of the underlying operator. As an alternative to recent research into hyperbolic/elliptic characteristic splitting of the discretization, the proposed formulation uses the relaxation scheme to reduce the hyperbolic error components and the multigrid coarse space in dealing with the elliptic components.

The contributions of this thesis are summarized as below:

- **Fast solution of the Euler and Navier-Stokes equations.** A method for the fast solution of the Euler and Navier-Stokes equations on unstructured meshes with the promise of handling more realistic flows has been developed. A key point in the development of this scheme is the utilization of a Finite Element discretization which results in a compact stencil. This permits the exact derivative for the flux Jacobians resulting in efficient and grid independent multigrid convergence using a proper choice of simple multigrid components.
- **Application of AMG to stabilized FEM formulations of the Euler and Navier-Stokes equations.** The coarse space operators are constructed in an alge-

braic fashion using the Galerkin Coarse Grid Approximation which allows the coarse grid equations to be constructed quickly without rediscrctization. A consistent weighting rescaling of the coarse grid equations is performed with particular attention paid to the boundary conditions. Analysis of the application of Algebraic Multigrid to stabilized discretizations of the two-dimensional flow equations showed that a length weighted modification of stabilization terms on the coarse spaces is required to ensure stability of the multigrid algorithm. Numerical studies of the proposed multigrid formulation for scalar elliptic operators, scalar convection-diffusion operators as well as the Euler and Navier-Stokes equations have exhibited significant improvements in the multigrid convergence rates when compared to other similar multigrid formulations. The convergence rates exhibited by the proposed multigrid algorithm for Euler and Navier-Stokes are relatively mesh independent and show asymptotic convergence as the number of coarse grids is increased.

- **Implementation of a semi-coarsening elemental agglomeration scheme to hyperbolic/parabolic systems on unstructured meshes.** The construction of the multigrid coarse spaces is based on an elemental agglomeration algorithm which allows for fast, automated coarse space construction as well as higher order multigrid interpolation operators. The coarse space construction is based on a semi-coarsening scheme which is mesh dependent and has been shown to be superior to full coarsening schemes for even the simplest elliptic problems. The agglomerated coarse space elements are generalized polygons which nullifies the need for a possibly expensive and/or complicated retriangulation.
- **Development of an improved coarse space interpolant.** The efficiency of the coarse space error corrections depends on the accuracy of the coarse space interpolants for the discrete problem. The multigrid interpolation operators are constructed using an extension of Chan *et al* [38] such that linear interpolation based on edge length weighting produces a superior interpolant.
- **Development of a generalized line implicit smoother for unstructured meshes.** The proposed multigrid relaxation scheme is a line implicit relaxation scheme where the lines are constructed to follow the direction of strongest influence which, under the right conditions, can lead to exact solvers. The developed smoother is a significant

improvement over similar smoothers [6, 71, 105] especially for viscous computations. A Fourier analysis of the smoother shows good predicted convergence rates.

6.2 Future Work

The current results in this thesis represent a proof of concept for the central idea of assigning the tasks of the multigrid components in dealing with the characteristic subsystems of the flow equations. Suggestions for further research include:

1. **Further development of FEM formulation for more realistic applications.** Further development of the current Finite Element formulation is required to handle more realistic flows. Currently, applications with discontinuities such as shocks cannot be handled properly due to the lack of a good shock capturing scheme, especially with higher order elements. Also, the current viscous formulation for the stabilization matrix in the GLS/FEM discretization does not appear to have the proper limiting behavior with Reynolds number. A reformulation which has the proper scaling properties should allow more practical applications at higher Reynolds numbers. Finally, practical applications at higher Reynolds numbers are precluded without the inclusion of a turbulence model. Of interest would be the behavior of the proposed multigrid algorithm in the presence of a turbulence model. Possible issues which may arise are related to the additional introduced stiffness as well as the degree of coupling with the flow equations. It is also not clear if an energy stable formulation for a turbulence model in entropy variables exists.
2. **Implementation of the FAS/FMG multigrid formulation.** Many of the issues relating to the consistency of the coarse grid equations arise from the fact that in the context of the linear multigrid formulation, the coarse grid equations are constructed algebraically such that special attention via rescaling becomes necessary. A reconstruction of the non-linear equations within an FAS formulation on the coarse spaces would remove the need for rescaling as well as the need for the initial time damping scheme as described in Sec. 4.2.1. This however is not without associated problems such as how to perform a rediscritization when the coarse space elements are not standard finite elements. Also present is the issue regarding implicit line construction since the line creation algorithm is based on a linearization of the non-linear equations.

This could potentially lead to extra overhead with regards to storage and compute time. In the current linear multigrid context, the implementation of a Full Multigrid (FMG) formulation [46] would help in dealing with the initial non-linear behavior of the system.

3. **Higher order finite elements.** The use of quadratic or higher order finite elements for more accurate solutions would be desirable. However, it is not clear how this might be implemented with the context of the current multigrid formulation. In such a case, chaining elements within a line as opposed to vertices may offer a better approach. One idea is to switch the discretization to a Discontinuous Galerkin/Least Squares (DGLS) formulation which leads to a natural description of the elements as separable entities that can be linked up in a line. The strong influence basis for the line creation could then be based on some formulation using the flux continuity between the elements.
4. **Extension to three dimensions.** The basic concepts in the proposed algorithm are extendible to three dimensions and any issues that arise should largely be implementation ones.
5. **Parallelization of the multigrid algorithm.** To address large scale applications, a parallel implementation of the multigrid scheme would be required. This need is most critical for 3-D applications. It should however be noted that parallel implementations of line implicit solvers as well as the Gauß-Seidel scheme are not trivial, especially in an unstructured grid context.

Appendix A

Multigrid Preconditioning Matrix

The linear algorithm described in Algorithm 1 may be viewed as a preconditioner for which an iteration matrix may be constructed. This can be shown fairly easily for the V-cycle multigrid cycle. We consider the general $V(\nu_1, \nu_2)$ cycle for the two-level method but simplify it by assuming that we have only one pre-smoothing and one post-smoothing i.e a $V(1,1)$ cycle. Let $\mathbf{A}_0$ represent the fine grid matrix and $\mathbf{A}_1$ represent the coarse grid matrix. For an initial guess $u^{(0)} = 0$:

1. Symmetric pre-smoothing: $u^{(1)} = \mathbf{S}_0^T b_0$

2. Coarse grid correction:

(a) Restrict residual:

$$q^{(0)} = \mathbf{R}_0(\mathbf{I} - \mathbf{A}_0 \mathbf{S}_0^T) b_0$$

(b) Coarse grid solve:

$$q^{(1)} = \mathbf{A}_1^{-1} \mathbf{R}_0(\mathbf{I} - \mathbf{A}_0 \mathbf{S}_0^T) b_0$$

(c) Fine grid correction:

$$\begin{aligned} u^{(2)} &= u^{(1)} + \mathbf{P}_0 q^{(1)} \\ &= [\mathbf{S}_0^T + \mathbf{P}_0 \mathbf{A}_1^{-1} \mathbf{R}_0(\mathbf{I} - \mathbf{A}_0 \mathbf{S}_0^T)] b_0 \end{aligned}$$

3. Post-smoothing:

$$\begin{aligned} u^{(3)} = \mathbf{M}_{mg}^{-1}b &= u^{(2)} + \mathbf{S}_0(b_0 - \mathbf{A}_0 u^{(2)}) \\ &= [\mathbf{S}_0 + \mathbf{S}_0^T - \mathbf{S}_0 \mathbf{A}_0 \mathbf{S}_0^T + (\mathbf{I} - \mathbf{S}_0 \mathbf{A}_0) \mathbf{P}_0 \mathbf{A}_1^{-1} \mathbf{R}_0 (\mathbf{I} - \mathbf{A}_0 \mathbf{S}_0^T)] b_0 \end{aligned}$$

The multigrid iteration matrix $\mathbf{S}_{mg}$ now takes the form:

$$\mathbf{S}_{mg} = (\mathbf{I} - \mathbf{S}_0 \mathbf{A}_0)(\mathbf{I} - \mathbf{P}_0 \mathbf{A}_1^{-1} \mathbf{R}_0 \mathbf{A}_0)(\mathbf{I} - \mathbf{S}_0^T \mathbf{A}_0) \quad (\text{A.1})$$

For the extension to multiple levels and variable number of pre- and post-smoothing sweeps, we refer to [18].

Appendix B

Nodal Agglomeration

It may be easily shown for a 1D Laplacian operator, uniform coarsening using a nodal agglomeration results in a scaling error.

This figure was broken... original no longer exists.

Figure B.1: 1D two-grid multigrid example

Following the analysis of Mavriplis [36], let us consider a discretization of the Poisson problem

$$\frac{d^2 u}{dx^2} = f \tag{B.1}$$

on the 1D grid as shown in Fig. B.1. This yields

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = f \tag{B.2}$$

If the coarse grid is created by agglomeration of neighboring pairs of cells as shown in Fig. B.1, the restriction operator based on injection corresponds to a simple summation of the $i - 1$ and i residuals. The prolongation operator based on injection gives

$$\begin{aligned} u_{i-2} &= u_{i-1} = \bar{u}_{I-1} \\ u_i &= u_{i+1} = \bar{u}_I \\ u_{i+2} &= u_{i+3} = \bar{u}_{I+1} \end{aligned} \tag{B.3}$$

where the overbar indicates the coarse grid values. The discrete coarse grid operator at I is obtained by the Galerkin Coarse Grid Approximation (GCA):

$$\mathbf{R} \mathbf{A} \mathbf{P} \bar{\mathbf{u}} = \mathbf{R} \mathbf{f} \tag{B.4}$$

which yields

$$\frac{\bar{u}_{I-1} - 2\bar{u}_I + \bar{u}_{I+1}}{2h^2} = f \tag{B.5}$$

This is inconsistent with a rediscrretization on the coarse grid which would yield

$$\frac{\bar{u}_{I-1} - 2\bar{u}_I + \bar{u}_{I+1}}{4h^2} = f \tag{B.6}$$

If we now perform a similar analysis with a restriction operator based on injection and a prolongation based on linear interpolation, the prolongation operator gives

$$\begin{aligned} u_{i-1} &= \frac{3}{4}\bar{u}_{I-1} + \frac{1}{4}\bar{u}_I \\ u_i &= \frac{1}{4}\bar{u}_{I-1} + \frac{3}{4}\bar{u}_I \\ u_{i+2} &= \frac{3}{4}\bar{u}_I + \frac{1}{4}\bar{u}_{I+1} \end{aligned} \tag{B.7}$$

Construction of the discrete coarse grid operator at point I now yields Eq. B.6. Hence, the choice of low-order interpolants results in a scaling factor of 2 for the Laplacian operator.

Let the coarse grid matrix constructed by nodal and elemental agglomeration be $\bar{\mathbf{A}}_V$ and $\bar{\mathbf{A}}_E$ respectively. From the analysis above,

$$\bar{\mathbf{A}}_V = 2\bar{\mathbf{A}}_E$$

If we now consider the multigrid iteration matrix (Eq. A.1) for these two methods, we obtain

$$\mathbf{S}_V = (\mathbf{I} - \mathbf{S}\mathbf{A})(\mathbf{I} - \mathbf{P}\bar{\mathbf{A}}_V^{-1}\mathbf{R}\mathbf{A})(\mathbf{I} - \mathbf{S}^T\mathbf{A}) \quad (\text{B.8})$$

$$\mathbf{S}_E = (\mathbf{I} - \mathbf{S}\mathbf{A})(\mathbf{I} - \mathbf{P}\bar{\mathbf{A}}_E^{-1}\mathbf{R}\mathbf{A})(\mathbf{I} - \mathbf{S}^T\mathbf{A}) \quad (\text{B.9})$$

The instantaneous convergence rate for the multigrid cycle at cycle $n+1$ is $\epsilon = \frac{\|\mathbf{r}^{n+1}\|}{\|\mathbf{r}^n\|}$. We may now compare the relative convergence rates for these two methods:

$$\begin{aligned} \frac{\epsilon_V}{\epsilon_E} &= \frac{\|\mathbf{A}\| \|\mathbf{S}_V\| \|\mathbf{A}^{-1}\|}{\|\mathbf{A}\| \|\mathbf{S}_E\| \|\mathbf{A}^{-1}\|} \\ &= \frac{\|\mathbf{S}_V\|}{\|\mathbf{S}_E\|} \\ &= \frac{\|(\mathbf{I} - \mathbf{S}\mathbf{A})\| \|(\mathbf{I} - \mathbf{P}\bar{\mathbf{A}}_V^{-1}\mathbf{R}\mathbf{A})\| \|(\mathbf{I} - \mathbf{S}^T\mathbf{A})\|}{\|(\mathbf{I} - \mathbf{S}\mathbf{A})\| \|(\mathbf{I} - \mathbf{P}\bar{\mathbf{A}}_E^{-1}\mathbf{R}\mathbf{A})\| \|(\mathbf{I} - \mathbf{S}^T\mathbf{A})\|} \\ &= \frac{\|(\mathbf{I} - \mathbf{P}\bar{\mathbf{A}}_V^{-1}\mathbf{R}\mathbf{A})\|}{\|(\mathbf{I} - \mathbf{P}\bar{\mathbf{A}}_E^{-1}\mathbf{R}\mathbf{A})\|} \\ &= \frac{1 - \|\mathbf{P}\bar{\mathbf{A}}_V^{-1}\mathbf{R}\| \|\mathbf{A}\|}{1 - \|\mathbf{P}\bar{\mathbf{A}}_E^{-1}\mathbf{R}\| \|\mathbf{A}\|} \\ &= \frac{1 - \frac{1}{2} \|\mathbf{P}\bar{\mathbf{A}}_E^{-1}\mathbf{R}\| \|\mathbf{A}\|}{1 - \|\mathbf{P}\bar{\mathbf{A}}_E^{-1}\mathbf{R}\| \|\mathbf{A}\|} \end{aligned}$$

Let $C_1 = \|\mathbf{P}\bar{\mathbf{A}}_E^{-1}\mathbf{R}\|$ and $C_2 = \|\mathbf{A}\|$. Since the multigrid methods are assumed to be convergent, the condition $C_1 C_2 < 1$ is satisfied. We then have:

$$\begin{aligned} \frac{\epsilon_V}{\epsilon_E} &= \frac{1 - \frac{1}{2}C_1 C_2}{1 - C_1 C_2} \\ &= 1 + \frac{1}{2} \left(\frac{C_1 C_2}{1 - C_1 C_2} \right) \\ &> 1 \end{aligned}$$

References

- [1] N.A. Pierce and M.B. Giles. Preconditioning on stretched meshes. Technical Report NA-95/10, Oxford University Computing Laboratory, 1995.
- [2] V. Venkatakrishnan. A Perspective on Unstructured Grid Flow Solvers. *ICASE Report 95-3*, 1995.
- [3] A. Jameson. The Present Status, Challenges, and Future Developments in Computational Fluid Dynamics. *Proceedings of the 7th AGARD Fluid Dynamics Panel Symposium, Seville*, October 1995.
- [4] A. Jameson and J.C. Vassberg. Computational Fluid Dynamics for Aerodynamic Design: Its Current and Future Impact. *39th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada*, 2001–0538, 2001.
- [5] E. Bertolazzi and G. Manzini. A Triangle-Based Unstructured Finite-Volume Method for Chemically Reactive Hypersonic Flows. *Journal of Computational Physics*, 166:84–115, 2001.
- [6] D.J. Mavriplis. Multigrid Strategies for Viscous Flow Solvers on Anisotropic Unstructured Meshes. *Journal of Computational Physics*, 145:141–165, 1998.
- [7] W. K. Anderson, R. D. Rausch, and D. L. Bonhaus. Implicit/multigrid algorithms for incompressible turbulent flows on unstructured grids. *J. Comput. Phys.*, 128:391–408, 1996.
- [8] N.A. Pierce, M.B. Giles, A. Jameson, and L. Martinelli. Accelerating Three-Dimensional Navier-Stokes Calculations. *AIAA*, 97–1953, 1997.
- [9] D.J. Mavriplis. An Assessment of Linear Versus Non-linear Multigrid Methods for Unstructured Mesh Solvers. *Journal of Computational Physics*, 175:302–325, 2001.
- [10] D. A. Knoll and W. J. Rider. A multigrid preconditioned Newton-Krylov method. *SIAM J. Sci. Comput.*, 21:692–710, 1999.

- [11] J.M.C. Pereira, M.H. Kobayashi, and J.C.F. Pereira. A Fourth-order-Accurate Finite Volume Compact Method for the Incompressible Navier-Stokes Solutions. *Journal of Computational Physics*, 167:217–243, 2001.
- [12] D.A. Knoll and V.A. Mousseau. On Newton-Krylov Multigrid Methods for the Incompressible Navier-Stokes Equations. *Journal of Computational Physics*, 163:262–267, 2001.
- [13] N.A. Pierce and M.B. Giles. Preconditioned Multigrid Methods for Compressible Flow Calculations on Stretched Meshes. *Journal of Computational Physics*, 136:425–445, 1997.
- [14] T. Barth, T.F. Chan, and Wei-Pai Tang. A Parallel Non-Overlapping Domain Decomposition Algorithm for Compressible Fluid Flow Problems on Triangulated Domains. *Contemporary Mathematics*, 218, 1998.
- [15] C. Vuik, P. Wesseling, and S. Zeng. Krylov subspace and multigrid methods applied to the incompressible Navier-Stokes equations. In N. D. Melson, T. A. Manteuffel, S. F. McCormick, and C. C. Douglas, editors, *Seventh Copper Mountain Conference on Multigrid Methods*, volume CP 3339, pages 737–753, Hampton, VA, 1996. NASA.
- [16] A. Brandt. Multi-level adaptive technique (MLAT) for fast numerical solution to boundary value problems. In H. Cabannes and R. Teman, editors, *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics*, volume 18 of *Lecture Notes in Physics*, pages 82–89, Berlin, 1973. Springer-Verlag.
- [17] A. Brandt. Multi-level adaptive techniques (MLAT) for partial differential equations: ideas and software. In J. R. Rice, editor, *Mathematical Software III*, pages 277–318. Academic Press, New York, 1977.
- [18] T.F. Chan, S. Go, and L. Zikatanov. Lecture Notes on Multilevel Methods for Elliptic Problems on Unstructured Grids. *VKI 28th Computational Fluid Dynamics*, pages 1–76, March 1997.
- [19] J. H. Bramble, J. E. Pasciak, J. Wang, and J. Xu. Convergence estimates for multigrid algorithms without regularity assumptions. *Math. Comp.*, 57:23–45, 1991.
- [20] J. H. Bramble and J. E. Pasciak. Uniform convergence estimates for multigrid V-cycle algorithms with less than full elliptic regularity. In *Domain Decomposition Methods in Science and Engineering: The Sixth International Conference on Domain Decomposition*, volume 157 of *Contemporary Mathematics*, pages 17–26, Providence, Rhode Island, 1994. American Mathematical Society.

- [21] J. H. Bramble and J. E. Pasciak. New estimates for multigrid algorithms including the V-cycle. *Math. Comp.*, 60:447–471, 1993.
- [22] J. Wang. Convergence Analysis Without Regularity Assumptions for Multigrid Algorithms Based on SOR Smoothing. *SIAM Journal on Numerical Analysis*, 29(4):987–1001, 1992.
- [23] R. Ni. A Multiple-Grid Scheme for Solving the Euler Equations. *AIAA*, 81–1025R, 1982.
- [24] A. Jameson. Solution of the Euler Equations by a Multigrid Method. *Applied Mathematics and Computation*, 13:327–356, 1983.
- [25] S.R. Allmaras. Multigrid for the 2D Compressible Navier-Stokes Equations. *Proceedings of AIAA 14th Computational Fluid Dynamics Conference, Norfolk, VA*, 99–3336-CP, 1999.
- [26] V. Venkatakrishnan. Improved Convergence of Compressible Navier-Stokes Solvers. *Proceedings of AIAA 29th Fluid Dynamics Conference, Albuquerque, NM*, 98–2967, 1999.
- [27] A. Reusken. Convergence Analysis of a Multigrid Method for Convection-Diffusion Equations. Technical Report 169, Institut fuer Geometrie und Praktische Mathematik, RWTH Aachen, 1998.
- [28] T.W. Roberts, D. Sidilkover, and R.C. Swanson. Textbook Multigrid Efficiency for the Steady Euler Equations. *13th AIAA Computational Fluid Dynamics Conference*, 97–1949, 1997.
- [29] J.L. Thomas, B. Diskin, and A. Brandt. Textbook Multigrid Efficiency for the Incompressible Navier-Stokes Equations: High Reynolds Number Wakes and Boundary Layers. *ICASE Report 99-51*, 1999.
- [30] A. Brandt. Barriers to Achieving Textbook Multigrid Efficiency (TME) in CFD. *ICASE Interim report No. 32*, 1998.
- [31] E. Nielsen, W.K. Anderson, R. Walters, and D. Keyes. Application of Newton-Krylov Methodology to A Three Dimensional Unstructured Euler Code. *AIAA*, 95–1733, 1995.
- [32] A. Pueyo and D.W. Zingg. An efficient newton-gmres solver for aerodynamic computations. *AIAA J.*, 36, no. 11:1991–1997, 1998.

- [33] M. Nemec and D.W. Zingg. Towards Efficient Aerodynamic Shape Optimization Based on the Navier-Stokes Equations. *15th AIAA Computational Fluid Dynamics Conference*, 2001–2532, 2001.
- [34] P.N. Brown and Y. Saad. Hybrid Krylov Methods for Non-Linear Systems of Equations. *SIAM J. Sci. Stat. Comput.*, 11:450–481, 1990.
- [35] J. Peraire, J. Peiro, and K. Morgan. Multigrid Solutions of the 3D Compressible Euler Equations on Unstructured Tetrahedral Grids. *International Journal for Numerical Methods in Engineering*, 36:1029–1044, 1993.
- [36] D.J. Mavriplis. Multigrid techniques for unstructured meshes. Technical Report 95-27, ICASE Report 95-27, 1995.
- [37] R. Bank and J. Xu. An Algorithm for Coarsening Unstructured Meshes. *Numerical Mathematics*, 73:1–36, 1996.
- [38] T.F. Chan, J. Xu, and L. Zikatanov. An Agglomeration Multigrid Method for Unstructured Grids. Technical Report 98-8, Computational and Applied Mathematics (CAM)/UCLA, February 1998.
- [39] N. Pierce. *Preconditioned Multigrid Methods for Compressible Flow Calculation on Stretched Meshes*. PhD thesis, University of Oxford, 1997.
- [40] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Math. Comp.*, 31:333–390, 1977.
- [41] A. Schmilovic and D.A. Caughey. Application of the Multigrid method to Calculations of Transonic Potential Flow about Wing-Fuselage Combinations. *Journal of Computational Physics*, 48:462–484, 1982.
- [42] H. Deconinck and C. Hirsch. A multigrid method for the transonic full potential equation discretized with finite elements on an arbitrary body fitted mesh. *J. Comput. Phys.*, 48:344–365, 1982.
- [43] D. C. Jespersen. A multigrid method for the Euler equations. *AIAA*, 83–0124, 1983.
- [44] J. W. Ruge and K. Stüben. Algebraic multigrid (AMG). In S. F. McCormick, editor, *Multigrid Methods*, volume 3 of *Frontiers in Applied Mathematics*, pages 73–130. SIAM, Philadelphia, PA, 1987.
- [45] Q. Chang, Y.S. Wong, and H. Fu. On the Algebraic Multigrid Method. *Journal of Computational Physics*, 125:279–292, 1996.

- [46] P. Wesseling. *An Introduction to Multigrid Methods*. John Wiley & Sons, Chichester, 1992.
- [47] M. Brezina, A.J. Cleary, R.D. Falgout, V.E. Henson, J.E. Jones, T.A. Manteuffel, S.F. McCormick, and J.W. Ruge. Algebraic Multigrid Based on Element Interpolation (AMGe). *SIAM*, 22:1570–1592, 2000.
- [48] S.R. Elias, G.D. Stubley, and G.D. Raithby. An Adaptive Agglomeration method for Additive Correction Multigrid. *International Journal for Numerical Methods in Engineering*, 40:887–903, 1997.
- [49] B.R. Hutchinson and G.D. Raithby. A Multigrid Method Based on the Additive Correction Strategy. *Numerical Heat Transfer*, 9:511–537, 1986.
- [50] M. Raw. Robustness of Coupled Algebraic Multigrid for the Navier-Stokes Equations. *AIAA*, pages 1–16, 1996.
- [51] R. Webster. An Algebraic Multigrid Solver for Navier-Stokes Problems. *International Journal for Numerical Methods in Fluids*, 8:761–780, 1994.
- [52] J. Jones and P.S. Vassilevski. AMGe Based on Element Agglomeration. Technical Report UCRL-JC-135441, CASC/Lawrence Livermore National Laboratory, 1999.
- [53] J. Fuhrman. Outlines of a modular algebraic multilevel method. In *AMLI'96: Proceedings of the Conference on Algebraic Multilevel Iteration Methods with Applications*, volume 1, pages 141–152, Nijmegen, The Netherlands, 1996. University of Nijmegen.
- [54] V.E. Henson and P.S. Vassilevski. Element Free AMGe: General Algorithms for Computing Interpolation Weights in AMG. Technical Report UCRL-JC-139098, CASC/Lawrence Livermore National Laboratory, 2000.
- [55] M.M. Gupta, J. Kouatchou, and J. Zhang. An Accurate and Stable Multigrid Method for Convection-Diffusion Equations. Technical report, Dept. of Mathematics, The George Washington University, October 1995.
- [56] D.J. Mavriplis. Three Dimensional Unstructured Multigrid for the Euler Equations. *AIAA*, 91–1549, 1991.
- [57] M. P. Leclercq and B. Stoufflet. Characteristic multigrid method application to solve the Euler equations with unstructured and unnested grids. *J. Comput. Phys.*, 104:329–346, 1993.
- [58] W.L. Wan, T.F. Chan, and B. Smith. An Energy-Minimizing Interpolation for Robust Multigrid Methods. *SIAM*, 21(4):1632–1649, 2000.

- [59] P. Vaněk, J. Mandel, and M. Brezina. Algebraic multigrid based on smoothed aggregation for second and fourth order problems. *Computing*, 56:179–196, 1996.
- [60] P. Vaněk, J. Mandel, and M. Brezina. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. In N. D. Melson, T. A. Manteuffel, S. F. McCormick, and C. C. Douglas, editors, *Seventh Copper Mountain Conference on Multigrid Methods*, volume CP 3339, pages 721–735, Hampton, VA, 1996. NASA.
- [61] T.F. Chan and W.L. Wan. Robust Multigrid methods for Elliptic Linear Systems. Technical Report 99-30, Computational and Applied Mathematics (CAM)/UCLA, September 1999.
- [62] D. Braess. Towards algebraic multigrid for elliptic problems of second order. *Computing*, 55:379–393, 1995.
- [63] D.J. Mavriplis. Multigrid Approaches to Non-linear Diffusion Problems on Unstructured Meshes. *ICASE Report 2001-3*, 2001.
- [64] A. Brandt. *Multigrid techniques: 1984 guide with applications to fluid dynamics*. GMD-Studien Nr. 85. Gesellschaft für Mathematik und Datenverarbeitung, St. Augustin, 1984.
- [65] A. Brandt and I. Yavneh. Accelerated multigrid convergence and high-reynolds recirculating flows. *SIAM J. Sci. Stat. Comput.*, 14:607–626, 1993.
- [66] J.L. Thomas, B. Diskin, and A. Brandt. Distributed Relaxation Multigrid and Defect Correction Applied to the Compressible Navier-Stokes Equations. *AIAA*, 99–3334, 1999.
- [67] H. Nishikawa and B. Leer. Optimal Multigrid Convergence by Elliptic/Hyperbolic Splitting. *AIAA*, 2002–2951, 2002.
- [68] R. Becker, C. Johnson, and R. Rannacher. Adaptive error control for multigrid finite element methods. *Computing*, 55:271–288, 1995.
- [69] J. H. Bramble and J. E. Pasciak. The analysis of smoothers for multigrid algorithms. *Math. Comp.*, 58:467–488, 1992.
- [70] S.R. Allmaras. Analysis of Semi-Implicit preconditioners for Multigrid Solution of the 2-D Compressible Navier-Stokes Equations. *Proceedings of AIAA 12th Computational Fluid Dynamics Conference, San Diego*, 95–1651-CP, 1995.
- [71] D.J. Mavriplis. Directional Agglomeration Multigrid Techniques for High Reynolds Number Viscous Flows. *ICASE Report 98-7*, pages 1–20, 1998.

- [72] E. Morano, D. J. Mavriplis, and V. Venkatakrishnan. Coarsening strategies for unstructured multigrid techniques with application to anisotropic problems. In N. D. Melson, T. A. Manteuffel, S. F. McCormick, and C. C. Douglas, editors, *Seventh Copper Mountain Conference on Multigrid Methods*, volume CP 3339, pages 591–606, Hampton, VA, 1996. NASA.
- [73] G. Strang and G. J. Fix. *An Analysis of the Finite Element Method*. Prentice-Hall, Englewood Cliffs, NJ, 1973.
- [74] G. H. Golub and C. F. Van Loan. *Matrix Computations*. John Hopkins University Press, Baltimore, MD, 2nd ed. edition, 1989.
- [75] Cl  s Johnson. *Numerical Solutions of Partial Differential Equations by the Finite Element Method*. Cambridge University Press, New York, NY, 1987.
- [76] W. Hackbusch. *Multigrid Methods and Applications*, volume 4 of *Computational Mathematics*. Springer-Verlag, Berlin, 1985.
- [77] J. H. Bramble and J. E. Pasciak. New convergence estimates for multigrid algorithms. *Math. Comp.*, 49:311–329, 1987.
- [78] A. Brandt. Rigorous quantitative analysis of multigrid, I: Constant coefficients two-level cycle with L_2 -norm. *SIAM J. Numer. Anal.*, 31:1695–1730, 1994.
- [79] S. Zhang. Optimal order nonnested multigrid methods for solving finite element equations. III. On degenerate meshes. *Math. Comput.*, 64:23–49, 1995.
- [80] S. Zhang. Optimal order nonnested multigrid methods for solving finite element equations. I. On quasi-uniform meshes. *Math. Comput.*, 55:23–36, 1990.
- [81] M. H. Lallemand, H. Steve, and A. Dervieux. Unstructured multigriding by volume agglomeration: current status. *Comput. Fluids*, 21:397–433, 1992.
- [82] B. Koobus, M. H. Lallemand, and A. Dervieux. Unstructured volume agglomeration MG: solution of the Poisson equation. *Int. J. Numer. Methods Fluids*, 18:27–42, 1994.
- [83] T.F. Chan and P. Vanek. Multilevel Algebraic Elliptic Solvers. Technical Report 99-9, Computational and Applied Mathematics (CAM)/UCLA, February 1999.
- [84] V. Venkatakrishnan and D. Mavriplis. Agglomeration Multigrid for the 3D Euler Equations. *AIAA*, 94-0069, 1994.
- [85] National institute of standards and definitions. <http://hiss.nist.gov/dads/terms.html>.

- [86] I.M. Llorente and N.D. Melson. Robust Multigrid Smoothers for Three Dimensional Elliptic Equations with Strong Anisotropies. *ICASE Report NASA/CR-1998-208700*, 1998.
- [87] B. Diskin. Solving Upwind-biased Discretizations II: Multigrid Solver Using Semi-coarsening. *ICASE Report 99-25*, 1999.
- [88] J. Bey and Arnold Reusken. On the Convergence of Basic Iterative Methods for Convection-Diffusion Equations. *Numer. Lin. Alg. Appl.*, 6:329–352, 1999.
- [89] S. Zeng and P. Wesseling. Numerical Study of a Multigrid Method with four smoothing methods for the Incompressible Navier–Stokes equations in general coordinates. In N. D. Melson, T. A. Manteuffel, and S. F. McCormick, editors, *Sixth Copper Mountain Conference on Multigrid Methods*, volume CP 3224, pages 691–708, Hampton, VA, 1993. NASA.
- [90] J. H. Bramble and J. E. Pasciak. The analysis of smoothers for multigrid algorithms. In T. A. Manteuffel and S. F. McCormick, editors, *Preliminary Proceedings of the Fifth Copper Mountain Conference on Multigrid Methods*, volume 1, pages 153–175, Denver, 1991. University of Colorado.
- [91] F. Shakib, T.J.R. Hughes, and Z. Johan. A New Finite Element Formulation for Computational Fluid Dynamics : X. The compressible Euler and Navier-Stokes equations. *Comp. Meth. in Appl. Mech. and Engrn.*, 89:141–219, 1991.
- [92] A.N. Brooks and T.J.R. Hughes. Streamline Upwind/Petrov Galerkin formulations for Convection Dominated Flows With Particular Emphasis on the Incompressible Navier-Stokes Equations. *Computer Methods in Applied Mechanics and Engineering*, 32:199–259, 1982.
- [93] T.J.R. Hughes, M. Mallet, and A. Mizukami. A New Finite Element Formulation for Computational Fluid Dynamics: II. Beyond SUPG. *Computer Methods in Applied Mechanics and Engineering*, 54:341–355, 1986.
- [94] T.J.R. Hughes, M. Mallet, and A. Mizukami. A New Finite Element Formulation for Computational Fluid Dynamics: III. The Generalized Streamline Operator for Multidimensional Advective-Diffusive Systems. *Computer Methods in Applied Mechanics and Engineering*, 58:305–328, 1986.
- [95] T.J.R. Hughes, L.P. Franca, and G.M. Hulbert. A New Finite Element Formulation for Computational Fluid Dynamics: VIII. The Galerkin Least-Squares method for advective diffusive equations. *Computer Methods in Applied Mechanics and Engineering*, 73:173–189, 1989.

- [96] F. Shakib. *Finite Element Analysis of the compressible Euler and Navier-Stokes equations*. PhD thesis, Dept. of Mech. Engineering, Stanford, November 1988.
- [97] J.C. Carette. *Adaptive Unstructured Mesh Algorithms and SUPG Finite Element Method for Compressible High Reynolds Number Flows*. PhD thesis, von Karman Institute, 1997.
- [98] A. Mizukami. An Implementation of the Streamline-Upwind/Petrov-Galerkin Method for Triangular Elements. *Computer Methods in Applied Mechanics and Engineering*, pages 357–364, 1985.
- [99] I. Christie, D.F. Griffiths, A.R. Mitchell, and O.C. Zienkiewicz. Finite element methods for second order differential equations with significant first derivatives. *Int. J. Num. Meth. Engr.*, 10:1389–1396, 1976.
- [100] B. van Leer, W. T. Lee, P. L. Roe, K. G. Powell, and C. H. Tai. Design of optimally smoothing multistage schemes for the Euler equations. *Comm. Appl. Num. Methods*, 8:761–769, 1992.
- [101] A. Jameson, W. Schmidt, and E. Turkel. Numerical Solution of the Euler Equations by Finite Volume Methods using Runge-Kutta Time Stepping Schemes. *AIAA*, 81–1259, 1981.
- [102] A. Jameson. Numerical solution of the Euler equations for the compressible inviscid fluids. In F. Angrand, A. Dervieux, J. A. Desideri, and R. Glowinski, editors, *Numerical Methods for the Euler Equations of Fluid Dynamics*, volume 21 of *Proceedings in Applied Mathematics*, pages 199–245. SIAM, Philadelphia, 1985.
- [103] C.H Tai. *Acceleration Techniques for Explicit Euler Codes*. PhD thesis, Dept. of Aero. Engineering, University of Michigan, 1990.
- [104] J. Lynn. *Multigrid Solution of the Euler Equations with Local Preconditioning*. PhD thesis, Dept. of Aero. Engineering, University of Michigan, 1995.
- [105] D.J. Mavriplis. Large-scale Parallel Viscous Flow Computations Using an Unstructured Multigrid Algorithm. *ICASE Report 99-44*, 1999.
- [106] T.F. Chan and P. Vanek. Detection of strong coupling in algebraic multigrid solvers. Technical Report 00-10, Computational and Applied Mathematics (CAM)/UCLA, March 2000.
- [107] A. Jameson and T.J. Baker. Solution of the Euler Equations for Complex Geometries. *Proceedings of AIAA 6th Computational Fluid Dynamics Conference, New York*, pages 293–302, 1983.

- [108] S. Ta'asan. Canonical-variables Multigrid Method for Steady-State Euler Equations. *ICASE Report 94-14*, 1994.
- [109] D. Sidilkover and U. M. Ascher. A multigrid solver for the steady state Navier Stokes equations using the pressure Poisson formulation. *Comput. Appl. Math.*, 14:21–35, 1995.
- [110] T.W. Roberts and R.C. Swanson. Extending ideally Converging Multigrid methods to Airfoil Flows. *14th AIAA Computational Fluid Dynamics Conference*, 99–3337, 1999.
- [111] D. Sidilkover. A Genuinely Multidimensional Upwind Scheme and Efficient Multigrid Solver for the Compressible Euler Equations. *ICASE Report 94-84*, 1994.
- [112] D. Sidilkover. Some Approaches Towards Constructing Optimally Efficient Multigrid Solvers for the Inviscid Flow Equations. *Computers and Fluids*, 28:551–557, 1999.
- [113] T.W. Roberts, D. Sidilkover, and J.L. Thomas. Multigrid Relaxation of a Factorizable, Conservative Discretization of the Compressible Flow Equations. *AIAA Fluids 2000 Conference*, 00–2285, 2000.
- [114] J.S. Wong, D.L. Darmofal, and J. Peraire. The Solution of the Compressible Euler Equations at Low Mach Numbers using a Stabilized Finite Element Algorithm. *Comp. Meth. in Appl. Mech.*, 190:5719–5737, 2001.
- [115] T.J.R. Hughes, L.P. Franca, and M. Mallet. A New Finite Element Formulation for Computational Fluid Dynamics: I. Symmetric Forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics. *Computer Methods in Applied Mechanics and Engineering*, 54:223–234, 1986.
- [116] T. Barth. Numerical Methods for Gasdynamic Systems on Unstructured Meshes. *An Introduction to Recent Developments in Theory and Numerics Conservation Laws Lecture Notes in Computational Science and Engineering*, pages 195–284, 1998.
- [117] A. Harten. On the Symmetric Form of Systems of Conservation Laws with Entropy. *Journal of Computational Physics*, 49:151–164, 1983.
- [118] S.K. Godunov. An Interesting Class of Quasilinear Systems. *Dokl. Akad. Nauk. SSSR*, 139:521–523, 1961.
- [119] M.S. Mock. Systems of Conservation Laws of Mixed Type. *J. Diff. Eqns*, 37:70–88, 1980.
- [120] P.L. Roe. Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes. *Journal of Computational Physics*, 43:357–372, 1981.

- [121] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Math. Comp.*, 31:333–390, 1977.
- [122] M.Giles. Energy Stability Analysis of Multistep Methods on Unstructured Meshes. Technical Report CFDL-TR-87-1, Computational Fluid Dynamics Laboratory/M.I.T, 1987.
- [123] A.J. Cleary, R.D. Falgout, V.E. Henson, J.E. Jones, J.E. Jones, T.A. Manteuffel, S.F. McCormick, G.N. Miranda, and J.W. Ruge. Robustness and Scalability of Algebraic Multigrid. *SIAM*, 21:1886–1908, 2000.
- [124] D.J. Mavriplis. Personal communication, 2002.
- [125] R.S. Montero and I.M. Llorente. Robust Multigrid Algorithms for the Incompressible Navier-Stokes Equations. *ICASE Report 2000-27*, 2000.
- [126] F. Shakib and T.J.R. Hughes. A New Finite Element Formulation for Computational Fluid Dynamics : IX. Fourier Analysis of Space-Time Galerkin/Least Squares Algorithms. *Comp. Meth. in Appl. Mech. and Engrn.*, 87:35–58, 1991.
- [127] K. Jansen T.J.R. Hughes, G. Hauke and Z. Johan. Current reflections on stabilized finite element methods for computational fluid mechanics. In *Finite Elements in Fluids; New Trends and Applications*, volume 1, pages 44–68. SEMNI, Barcelona, 1993.
- [128] K.E. Jansen. Unstructured grid large eddy simulation of flow over an airfoil. *CTR Annual Research Briefs 1994*, pages 161–173, 1994.
- [129] K.E. Jansen. Large-eddy simulation using unstructured grids. *Invited paper in Advances in DNS/LES*, Greyden Press, pages 117–128, 1997.
- [130] K.E. Jansen. Computation of turbulence with stabilized methods. *Invited paper in 4th Japan-US Symposium on Finite Element Methods in Large-Scale Computational Fluid Dynamics*, 1998.
- [131] K.E. Jansen. A stabilized finite element method for computing turbulence. *Computer Methods in Applied Mechanics and Engineering*, 174:299–317, 1999.
- [132] A.C. Galeão and E.G. Dutra do Carmo. A consistent approximate upwind Petrov-Galerkin method for convection-dominated problems. *Comp. Meth. Appl. Appl. Mech. Engrg.*, 68, 1989.
- [133] T.J.R. Hughes and M. Mallet. A New Finite Element Formulation for Computational Fluid Dynamics: IV. A Discontinuity Capturing Operator for Multidimensional

Advective-Diffusive Systems. *Computer Methods in Applied Mechanics and Engineering*, 58:329–336, 1986.