Available online at www.sciencedirect.com

SCIENCE DIRECT®

Comput. Methods Appl. Mech. Engrg. 193 (2004) 3667–3686

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

Algebraic multigrid for stabilized finite element discretizations of the Navier–Stokes equations

T. Okusanya, D.L. Darmofal *, J. Peraire

Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 77 Massachusetts Avenue Rm. 37-442, Cambridge, MA 021139, USA

Received 14 July 2003; received in revised form 22 January 2004; accepted 30 January 2004

Abstract

An algebraic multigrid method for the solution of 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 strategy employing a semi-coarsening scheme designed to reduce grid anisotropy. The viscous terms are discretized in a consistent manner on coarse grids using an algebraic Galerkin coarse grid approximation in which higher-order grid transfer operators are constructed from the underlying triangulation. However, the combination of higher-order transfer operators and Galerkin rediscretization diminishes the stability of stabilized inviscid operators on coarse grids and a modification is proposed to alleviate this problem. A generalized line implicit relaxation scheme is also described where the lines are constructed to follow the direction of strongest coupling. Applications are demonstrated for convection–diffusion, Euler, and laminar Navier–Stokes. The results show that the convergence rate is largely unaffected by mesh size over a wide range of Reynolds (Peclet) numbers.

Keywords: Algebraic multigrid; Stabilized finite element discretizations; Navier–Stokes equations

1. Introduction

Rapid advances in unstructured mesh methods for computational fluid dynamics (CFD) have been made in recent years for both inviscid and viscous flows. In particular, unstructured meshes offer a practical means for computation with complex domains and have the additional advantage of allowing easy adaptation of the mesh. However, accurate and efficient solutions of the compressible Navier–Stokes equations, especially in the turbulent high Reynolds number limit, remain a challenging problem due in part to the wide range of associated length scales required to properly resolve flow features. This is especially true in the boundary layer regions which are characterized by strong variations in the normal

* Corresponding author. Tel.: +1-617-258-0743; fax: +1-617-258-5143.
E-mail address: darmofal@mit.edu (D.L. Darmofal).

0045-7825/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.cma.2004.01.025

direction but relatively weak streamwise variations. In order to accurately resolve the boundary layer in a computationally efficient manner, grid anisotropy is employed. The grid anisotropy and the number of unknowns required for these high Reynolds number flows (as compared to purely inviscid flows) leads to a significant decrease in the effectiveness of many iterative solution methods.

The state of the art in solution methods for high Reynolds number flows on unstructured meshes is well represented by the work of Mavriplis [1–4]. Mavriplis discretizes the Reynolds-averaged Navier–Stokes (RANS) equations using a node-based finite volume method stabilized by a matrix-based artificial dissipation. The solution method is a semi-coarsening non-linear multigrid scheme with a hybrid point/line implicit smoother. For a variety of aerodynamic applications from airfoils to three-dimensional aircraft, convergence rates for this approach ranged from 0.78 to 0.97, respectively. Of particular note with respect to our work, Mavriplis [4] demonstrated that the limiting factor in the potential convergence rates for Navier–Stokes was the use of lower-order approximations for the coarse grid discretizations. He concludes that any improvement in the multigrid algorithm for these applications will have little effect on convergence rates unless the coarse grid approximations are improved.

The use of lower-order approximations in multigrid solution of traditional finite volume discretizations occurs because the stencils are *non-compact*. For a linear multigrid method (in which coarser grids are used to solve the linear system arising from a Newton–Raphson iteration), the computational cost of evaluating and/or storing the linearized operators is often impractical as a result of the extended stencils. For a non-linear multigrid method (in which coarser grids approximate the non-linear problem directly), while storage and/or calculation of the entire linearized operator is not required, a lower-order approximation of the non-linear equations is still often used to decrease the work on the coarse grid. A possible solution to these problems exists in the finite element method (FEM) since the resulting stencils are compact. Furthermore, FEM discretizations readily permit higher-order formulations on unstructured meshes. We believe the combination of compactness and higher-order accuracy are the critical ingredients for the development of an accurate and efficient approximation method for the Navier–Stokes equations.

Our long term goal is the development of an effective solution method for accurate simulations of the Navier–Stokes and Reynolds-averaged Navier–Stokes equations in complex geometries. The basic approach we are following employs higher-order FEM discretizations with multigrid solution techniques. In this paper, we take an initial step towards this goal by considering multigrid solution of piecewise-linear finite element discretizations for convection-dominated problems including laminar Navier–Stokes flows. We begin by summarizing our strategy for applying multigrid to convection-dominated problems. Then, the detailed algorithm is described and results are demonstrated for scalar convection–diffusion. Finally, we apply an extension of the algorithm to Euler and Navier–Stokes solutions. Subsequent papers will consider applications to higher-order discretizations and more complicated problems.

2. Overview of proposed multigrid strategy

Multigrid introduces coarser meshes to accelerate the convergence on the desired (fine) mesh. This naturally leads to a decomposition of the solution error into rough error components, which cannot be resolved on a coarser grid, and the complementary smooth error components which can be resolved on the coarser grid. The crux of multigrid methods is the reduction of the rough error modes by means of an inexpensive relaxation scheme on the current mesh, and correction of the smooth errors using the coarse grid. Hence, the effectiveness of the multigrid performance is dependent on the synergy of the relaxation scheme with the coarsening algorithms.

Until recently, multigrid solution of the Navier–Stokes equation for high Reynolds number flows had very poor convergence rates often around 0.99 or worse [5]. As demonstrated by Pierce and Giles [6], the use of high aspect ratio cells within the boundary layer collapses the convective eigenvalues onto the origin while decoupling the acoustic modes from the streamwise coordinate direction. Motivated by Fourier analysis, Pierce and Giles developed a multigrid methodology for structured meshes based on point block implicit smoothing and semi-coarsening. While this leaves some error modes lightly damped, the resulting algorithm was shown to be a significant improvement over previous multigrid performance on structured meshes. The work of Pierce and Giles led to the unstructured mesh multigrid algorithm of Mavriplis who employed line solvers (in addition to semi-coarsening) to further strengthen the multigrid algorithm. In both of these works, multigrid convergence rates where shown to be insensitive to cell aspect ratios in the boundary layers.

The multigrid algorithm we have developed is based on these recent advances for solving finite volume discretizations of high Reynolds number problems. In particular, we combine an implicit line relaxation algorithm with a semi-coarsening, agglomeration multigrid algorithm to solve stabilized FEM discretizations. The line relaxation is designed to solve implicitly along the direction of strongest coupling in the mesh and the agglomeration strategy is designed to reduce mesh anisotropy. Nodal agglomeration is a popular coarsening technique in which fine grid nodes are merged to produce a coarse mesh [7–10,17]. However, a known problem with existing nodal agglomeration methods is their inability to accurately represent higher-order differential operators due to low accuracy multigrid transfer operators [1]. An alternative agglomeration method is elemental (or volume) agglomeration that fuses neighboring elements into macroelements [11–16]. In particular, Chan et al. [14] have developed higher-accuracy transfer operators which are critical for efficient multigrid solutions of viscous problems. Our agglomeration multigrid algorithm is an extension of Chan's approach but produces semi-coarsened grids to reduce anisotropy. Furthermore, while the use of higher-order interpolants results in consistently discretized viscous operators, in this work we demonstrate an important duality with respect to inviscid operators. Specifically, the use of higher-order transfer operators diminishes the stability of stabilized inviscid operators on coarser meshes.

3. Application to convection–diffusion

The strong form of the two-dimensional, linear convection–diffusion equation is,

$$\nabla \cdot (\mathbf{V}\Phi) - \nabla \cdot [\mu(x,y)\nabla\Phi] = f \quad \text{in } \Omega, \tag{1}$$

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 $\Gamma_{\mathrm{N}} = \Gamma \setminus \Gamma_{\mathrm{D}}$. $\mathbf{V} = (u, v)$ 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

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

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

and also consider the following boundary subsets,

$$\Gamma^{\pm}_{\mathrm{D}} = \Gamma_{\mathrm{D}} \cap \Gamma^{\pm},$$

$$\Gamma^{\pm}_{\mathrm{N}} = \Gamma_{\mathrm{N}} \cap \Gamma^{\pm}.$$

We simplify the boundary conditions by assuming that $\Gamma^- \subset \Gamma_{\mathrm{D}}$ such that Eq. (1) is subject to the boundary conditions,

$$\Phi = g_{\mathrm{D}} \quad \text{on } \Gamma_{\mathrm{D}},$$

$$(-\Phi\mathbf{V} + \mu\nabla\Phi)\cdot\mathbf{n} = g_{\mathrm{N}} \quad \text{on } \Gamma^{+}_{\mathrm{N}}.$$

The behavior of Eq. (1) is governed by the Peclet number,

$$Pe = \frac{UL}{\mu},$$

which represents the relative importance of advection to diffusion where U is some reference speed and L is a characteristic length of the problem. The application of the classical Galerkin finite element method to the advection-dominated convection–diffusion equation is well known to lack stability, hence, we consider the streamline upwind/Petrov Galerkin (SUPG) method introduced by Brooks and Hughes [18]. A generalization of the SUPG method is the Galerkin/least squares (GLS) method [19] which is equivalent to the SUPG method for purely hyperbolic operators and/or for piecewise-linear elements.

Let the spatial domain Ω_h, be discretized into non-overlapping elements T_e, such that $\Omega_h = \cup T_e$, and $T_e \cap T_{e'} = \emptyset$, $e \neq e'$. Within each element, the solution is approximated by a set of kth-order interpolation polynomials $\mathscr{P}_k$ which are $\mathscr{C}^0$ continuous across the elements,

$$\mathscr{P}^h = \{U^h | U^h \in \mathscr{C}^0(\Omega_h),\ U^h|_{\Omega^e} \in \mathscr{P}_k(\Omega^e)\ \forall \Omega^e \in \Omega_h\}.$$

We consider the trial finite element solution space $\mathscr{S}^h$,

$$\mathscr{S}^h = \{V^h | V^h \in \mathscr{P}^h,\ V^h = g_{\mathrm{D}} \text{ on } \Gamma_{\mathrm{D}}\},$$

and introduce the space $\mathscr{V}^h$ of test functions,

$$\mathscr{V}^h = \{W^h | W^h \in \mathscr{P}^h,\ W^h = 0 \text{ on } \Gamma_{\mathrm{D}}\}.$$

The discrete variational form of Eq. (1) reduces to finding $\Phi_h \in \mathscr{S}^h$ such that:

$$B(\Phi_h, w_h)_{\mathrm{gal}} + B(\Phi_h, w_h)_{\mathrm{gls}} + B(\Phi_h, w_h)_{\mathrm{bc}} = 0 \quad \forall w_h \in \mathscr{V}^h, \tag{2}$$

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

$$B(\Phi_h, w_h)_{\mathrm{gal}} = \int_{\Omega} (-\Phi_h \mathbf{V} \cdot \nabla w_h + \mu \nabla \Phi_h \cdot \nabla w_h - w_h f)\,\mathrm{d}\boldsymbol{\Omega},$$

$$B(\Phi_h, w_h)_{\mathrm{gls}} = \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} (\mathbf{V} \cdot \nabla w_h - \mu \nabla^2 w_h - f)\tau_e \cdot (\mathbf{V} \cdot \nabla \Phi_h - \mu \nabla^2 \Phi_h - f)\,\mathrm{d}\boldsymbol{\Omega},$$

$$B(\Phi_h, w_h)_{\mathrm{bc}} = \int_{\Gamma_{\mathrm{N}}} w_h g_{\mathrm{N}}\,\mathrm{d}\Gamma.$$

The coefficient τ_e is the GLS stabilization parameter which is positive and represents an intrinsic time scale. The dependence of τ_e on the local element length scale (h_e) and local element Peclet number ($Pe_h = \|\mathbf{V}\| h_e/\mu$) is important in order to provide accuracy and stability [20]. For the construction of τ_e, we consider a generic triangle T_e and introduce an affine mapping $\mathbf{x} = \mathbf{x}(\xi)$, between the master triangle $\widehat{T}_e$, in the parametric space $\boldsymbol{\xi} = [\xi, \eta]^{\mathrm{T}}$, and $T_e \in \Omega$ in the mapped space $\mathbf{x} = [x, y]^{\mathrm{T}}$. Let

$$\mathscr{B}_1 = (y_{,\eta}\bar{u} - x_{,\eta}\bar{v})/x_{,\xi}y_{,\eta} - y_{,\xi}x_{,\eta},$$

$$\mathscr{B}_2 = (x_{,\xi}\bar{v} - y_{,\xi}\bar{u})/x_{,\xi}y_{,\eta} - y_{,\xi}x_{,\eta},$$

$$\mathscr{B}_3 = -\mathscr{B}_1 - \mathscr{B}_2,$$

where $\overline{\mathbf{V}} = (\bar{u}, \bar{v})$ is the average of the velocity defined at the three element vertices. Then, the parameter τ_e is defined by

$$D(\mathscr{B}_1,\mathscr{B}_2,\mathscr{B}_3)=\frac{3}{(|\mathscr{B}_1|+|\mathscr{B}_2|+|\mathscr{B}_3|)},$$

$$Pe_h=(\overline{\mathbf{V}}\cdot\overline{\mathbf{V}})\frac{D}{\mu},$$

$$\tau_e=D\frac{Pe_h}{\sqrt{9+Pe_h^2}}.$$

For the choice of the nodal basis functions and linear $\mathscr{P}_1$ interpolation polynomials in $\mathscr{S}^h$ and $\mathscr{V}^h$, Eq. (2) represents a sparse linear system of equations which can also be written simply as

$$\mathbf{A}_h\Phi_h=b_h. \tag{3}$$

3.1. Multigrid algorithm

Given a linear system of equations $\mathbf{A}_h\Phi_h=b_h$, derived on a grid Ω_h, we consider a multigrid formulation for the solution. The application of an iterative scheme or smoother $\mathscr{S}(\Phi_h^0,b_h,\mathbf{A}_h,n)$ results in a better estimate Φ_h^n to the solution after a given number of iterations n, starting from an initial guess Φ_h^0. The multigrid algorithm considers the computation of corrections to the error $e_h^n=\Phi_h^n-\Phi_h$, on a coarser grid Ω_H via a set of coarse grid equations $\mathbf{A}_He_H=b_H$ and the transfer operators, $\mathbf{R}_h$ and $\mathbf{P}_h$ which are the restriction and prolongation operators respectively [21]. This two grid algorithm may be applied recursively to a hierarchy of grids $\{\Omega_k:(k=0,\dots,m)\}$ 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. In our implementation of multigrid, the coarsening procedure terminates when the coarse grid system of equations $\mathbf{A}_m\Phi_m=b_m$ is small enough to be solved exactly.

3.2. Implicit line smoother

Consider a splitting of the matrix $\mathbf{A}_k$,

$$\mathbf{A}_k=\mathbf{M}_k-\mathbf{N}_k,$$

where $\mathbf{M}_k$ is non-singular. A fixed-point iterative method for the system is,

$$\begin{aligned}\Phi_k^{i+1}&=\mathbf{S}_k\Phi_k^i+\omega\mathbf{M}_k^{-1}b_k,\\ \mathbf{S}_k&=\mathbf{I}-\omega\mathbf{M}_k^{-1}\mathbf{A}_k,\end{aligned} \tag{4}$$

where Φ_k^i represents the current solution estimate at iteration i and ω is a relaxation factor. The matrix $\mathbf{M}_k$ is the preconditioning matrix and $\mathbf{S}_k$ is called the iteration matrix. It may be shown that the basic iterative method Eq. (4) will converge for any initial guess Φ_k^0 iff $\rho(\mathbf{S}_k)<1$ where $\rho(\mathbf{S}_k)$ is the spectral radius of the matrix $\mathbf{S}_k$. Smoothers such as point Jacobi that possess good damping properties for elliptic operators have been shown to exhibit poor convergence rates when applied to strongly convective systems [21,22]. Furthermore, many simple iterative schemes such as point Jacobi or point Gauss–Seidel discretizations are known to be unstable for higher-order solutions of hyperbolic partial differential equations [21]. A possible solution is to increase the stability region of the smoother by the introduction of a multistaging scheme. Multistage methods were developed by Jameson [23] for the solution of the Euler equations and have been applied with great success to a variety of applications [23,24]. An N-stage scheme using the $\mathbf{M}_k$ to precondition the linear residual has the following generic form,

$$\begin{aligned} u_k^0 &= \Phi_k^i, \\ u_k^1 &= u_k^0 + \omega\alpha_1 \mathbf{M}_k^{-1}(b_k - \mathbf{A}_k u_k^0), \\ \cdots &= \cdots \\ u_k^n &= u_k^0 + \omega\alpha_n \mathbf{M}_k^{-1}(b_k - \mathbf{A}_k u_k^{n-1}), \\ \cdots &= \cdots \\ u_k^N &= u_k^0 + \omega\alpha_{\mathrm{N}} \mathbf{M}_k^{-1}(b_k - \mathbf{A}_k u_k^{N-1}), \\ \Phi_k^{i+1} &= u_k^N, \end{aligned}$$

where α_n are the multistage coefficients. Although optimized multistage coefficients may be determined for a specific spatial discretization [24–29], in this work the optimized multistage coefficients developed by Lynn [29] for finite volume discretizations are utilized. Table 1 shows the numerical values employed for the 3-stage and 5-stage schemes.

In regions with highly stretched grids or strong convection, the convection–diffusion model problem is characterized by strong coupling in one direction. Hence, we opt to use an implicit line preconditioner where the implicit lines are constructed to follow directions of strong influence. For the highly stretched grids used boundary layer simulations, strong coupling often occurs normal to the boundary layer. In strongly convective regions of the flow, the strong coupling occurs in the streamwise directions. The line smoother developed here for unstructured grids is similar to the geometry-based lines formed by Mavriplis [2,3,30]. In Mavriplis' scheme, the formation of the lines is based purely on the local grid stretching and, in the isotropic regions of the grid, reverts to a point implicit smoother.

To measure nodal coupling for use in forming the implicit lines, we introduce a *coupling measure*. For a scalar convection–diffusion problem, we use the coefficients $a_{i,j}$ of $\mathbf{A}_k$ to form the coupling measure. Let the set of points, denoted by S_i, connected to node i be,

$$S_i \equiv \{j \neq i : a_{i,j} \neq 0\}.$$

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

$$\begin{aligned} &\mathrm{coup}(i,j) \equiv \max(\beta_i, \beta_j), \\ &\beta_i \equiv \frac{|a_{i,j}|}{\max|a_{i,k}|} \quad \{k : k \in S_i\}, \\ &\beta_j \equiv \frac{|a_{j,i}|}{\max|a_{j,k}|} \quad \{k : k \in S_j\}. \end{aligned} \tag{5}$$

In convection-dominated flows with upwind-type discretizations, strong coupling tends to be one-sided which necessitates the two way consideration of Eq. (5). We refer to [21,31] for other methods regarding the detection of strong coupling.

Table 1
Optimized damping multistage coefficients by Lynn [29]

	3 Stage	5 Stage
α_1	0.2239	0.08699
α_2	0.5653	0.1892
α_3	1	0.3263
α_4		0.5558
α_5		1

The algorithm for constructing the lines given the coupling measure is:

Procedure 1 (*Implicit line construction*).
Step 0: Create tag vector for the nodes and untag all nodes.
Step 1: Tag the current node and compute coupling measure for all of the untagged nodes connected to the current node.
Step 2: Choose the node which has the largest coupling measure greater than a threshold (typically 0.75). Repeat from Step 1.
Step 3: If no node exists meeting criteria, terminate line. Select next untagged node to initiate a new line. Repeat from Step 1 until all nodes are tagged.

Fig. 1 shows an example of the implicit line construction for a 2D GLS/FEM discretization of the linear convection–diffusion equation with an imposed velocity field of $\mathbf{V} = (-y, x)$. The use of a line relaxation scheme leads to a natural splitting of the matrix into tridiagonal sub-matrices which may be solved in $\mathrm{O}(N)$ time. Furthermore, after each solve of an individual line, the linear solution is updated resulting in a line Gauss–Seidel preconditioner.

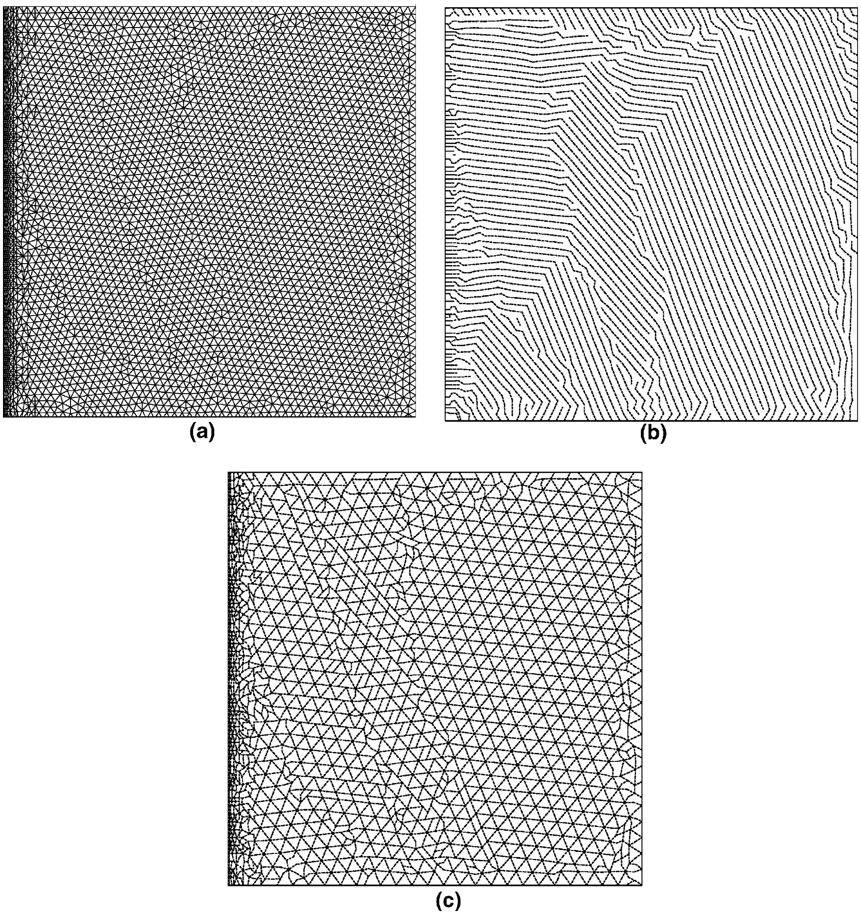

Fig. 1. Example of implicit line and coarse grid construction from a fine grid for scalar convection–diffusion (see Fig. 4 for a description of the problem): (a) grid, (b) implicit lines and (c) coarse grid.

3.3. Agglomeration and coarse grid operators

The coarse space matrix $\mathbf{A}_k$ may be constructed by rediscretization, however algebraic multigrid (AMG) considers an algebraic alternative to this [21] given by the recursive relation,

$$\mathbf{A}_{k+1} = \mathbf{R}_k \mathbf{A}_k \mathbf{P}_k.$$

Although the $\mathbf{R}_k$ and $\mathbf{P}_k$ operators are independent, the choice of $\mathbf{R}_k = \mathbf{P}_k^{\mathrm{T}}$ 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. This choice for the interpolation operators also has the added advantage that only one of the operators needs to be constructed. This method for coarse space operator construction is called the Galerkin coarse grid approximation (GCA) and is the approach we use.

To obtain mesh size independent convergence rates, the interpolation operators must have sufficient accuracy [21,32]. Specifically,

$$m_{\mathrm{P}} + m_{\mathrm{R}} > 2m, \tag{6}$$

where m_{P} and m_{R} are the order (degree plus one) of the polynomials that are interpolated exactly by the prolongation $\mathbf{P}$ and restriction $\mathbf{R}$ operators respectively, and $2m$ is the order of the governing partial differential equation.

As described previously, the agglomeration technique used to construct coarse grids can be node-based or element-based. However, the standard implementation of nodal agglomeration [7–10] uses simple injection (summation) operators for prolongation (restriction). As a result, m_{P} and m_{R} are unity, and for a second-order operator ($2m = 2$), the accuracy condition (Eq. (6)) is violated [21]. Thus, either the prolongation, restriction, or both must be exact for linear polynomials. To achieve this accuracy, we have chosen an elemental agglomeration strategy because the existence of the underlying triangulation allows the relatively straightforward development higher-order interpolation [14–16,33,34]. We note, however, that our use of the geometric information from the triangulation makes our approach a blend of geometric and algebraic multigrid.

A review of elemental agglomeration is given by Chan et al. [15] who also propose alternative elemental agglomeration algorithms. The goal of our elemental agglomeration method is to reduce mesh anisotropy while coarsening the grid. One important distinction between the proposed method and that described by Chan et al. [15] 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 [15]. One drawback of this formulation is that the support for the basis functions defined on these macroelements is larger than standard triangular elements.

The coarse grid topology is constructed by partitioning the elements into macroelement groups as shown in Fig. 2 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 (and would therefore be degenerate elements) are modified by the addition of extra supporting coarse nodes using fine grid nodes which lie on the macroedge connecting these two coarse nodes.

The decision to agglomerate two neighboring elements is determined by the *macroelement skew*. 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 0 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 and Mavriplis [35].

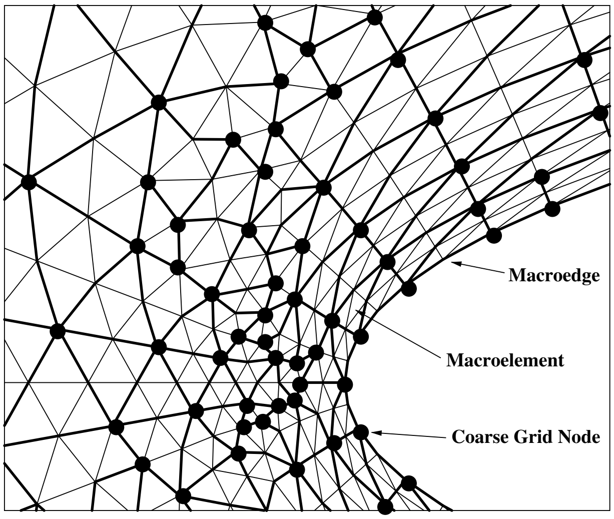

Fig. 2. Coarse space topology.

We also define the related concept of *edge skew*. Specifically, 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.

The macroelement construction algorithm is:

Procedure 2 (*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: Agglomerate unassigned neighboring elements with edge skew larger than some specified fraction (typically 0.75) of the average edge skew.
Step 3: Place elements which share a vertex but no edges with the last macroelement created into the queue.
Step 4: Repeat Step 1 until either all elements belong to a macroelement or there are no more seed elements.

In the case where the lengths and areas of the fine grid elements are equal, the algorithm degenerates to a 4:1 isotropic agglomeration in 2D and fully recovers the natural coarse structure for a regular grid. After the algorithm terminates, some 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 sliver elements with is based on the minimum edge skew. An example of an agglomerated coarse mesh is shown in Fig. 1(c).

To construct the multigrid interpolation operators, nodal basis functions are derived for the coarse space. The basis functions use geometry-weighted graph distance interpolation on both the boundary and interior. This approach is an extension of an interpolation proposed by Chan et al. [14] which makes use of graph distance interpolation on the boundary and constant interpolation over the interior. For every fine grid node i in a macroelement which corresponds to a coarse grid node j, the prolongation operator $\mathbf{P}_k(i,j) = 1$. For every other fine grid node i in the macroelement which does *not* correspond to a coarse grid node, the prolongation operator is,

$$\mathbf{P}_k(i,j) = \frac{\frac{1}{\mathrm{dist}(i,j)}}{\sum_l \frac{1}{\mathrm{dist}(i,l)}},$$

where $\mathrm{dist}(i,j)$ is the length along the graph from node i to coarse grid node j. If node i is on a macroelement edge, then the summation over l only includes the coarse grid nodes on that macroelement edge. Otherwise, if node i is in the interior of the macroelement, the summation occurs over all coarse grid nodes for the macroelement. We note that this prolongation operator is not linearly exact for general triangulations in the interior of the macroelements. Finally, with the prolongation operator defined as above, the restriction operator is determined using the Galerkin approach, $\mathbf{R}_k = \mathbf{P}_k^{\mathrm{T}}$.

3.4. Coarse grid stabilization scaling

Coarse grid algebraic discretizations can result in a lack of stability when applied to convection-dominated problems. The problem stems from the use of stabilized methods such as SUPG or GLS in which the stabilization enters through a term which is proportional to the local mesh size. As the grid is coarsened, the stabilization should be proportional to the coarse grid mesh size, however, straightforward application of the GCA algebraic discretization will result in the stabilization being based on the fine mesh spacing. Let us consider the matrix equation (Eq. (3)) where $\mathbf{A}_k$ is some matrix resulting from a stabilized method such that,

$$\mathbf{A}_k = \mathbf{A}_k^{\mathrm{base}} + \mathbf{A}_k^{\tau}, \tag{7}$$

where $\mathbf{A}_k^{\tau}$ is the component containing the stabilization parameter and $\mathbf{A}_k^{\mathrm{base}}$ is the base stiffness matrix (Galerkin + boundary terms for FEM). Stabilization schemes for convection–diffusion operators such as those used here add a dissipative term (roughly) of the form $\tau_h \nabla^2 \Phi_h$, where $\tau_h = \mathcal{O}(h)$. The higher-order interpolant produced with our approach results in a consistent approximation of second-order operators. Thus, on the coarse grid, $\tau_H \nabla^2 \Phi_H = \mathcal{O}(h)$ since $\tau_H = \mathcal{O}(h)$ and $\nabla^2 \Phi_H = \mathcal{O}(1)$. Clearly, the stabilization term is too small for the coarse grid by a factor of H/h since τ_H is based on h instead of H. For nodal agglomeration using the standard injection-based interpolation, second-order operators are not consistently approximated and are scaled by a factor of roughly H/h. Thus, the overall stabilization tends to be $\mathcal{O}(H)$ as desired when using this approach. This demonstrates an interesting and important duality, specifically: using higher-order interpolation results in consistently discretized viscous operators but diminishes the stability of (stabilized) inviscid operators, whereas using lower-order interpolation results in inconsistently discretized viscous operators but preserves the stability of the (stabilized) inviscid operators.

This loss of stabilization when using higher-order transfer operators can be demonstrated in the simple setting of scalar convection in one-dimension. For pure convection with constant velocity u, the stabilization parameter on a uniform fine mesh is, $\tau_e = h/2|u|$. The resulting stabilized discretization for linear elements is the standard first-order upwind scheme given by,

$$\tfrac{1}{2}u(\Phi_{h_{i+1}} - \Phi_{h_{i-1}}) - \tfrac{1}{2}|u|(\Phi_{h_{i+1}} - 2\Phi_{h_i} + \Phi_{h_{i-1}}) = 0.$$

The first term is the central difference operator which arises from the Galerkin term in the SUPG/GLS formulation. The second term is the usual upwind dissipation which arises from the stabilization term in the SUPG/GLS formulation. On a uniform mesh, the prolongation operator based on linear interpolation is,

$$\mathbf{P} = \begin{pmatrix} 1 & 0 \\ \frac{1}{2} & \frac{1}{2} \\ 0 & 1 \end{pmatrix}.$$

Using a Galerkin coarse grid discretization in which $\mathbf{R} = \mathbf{P}^{\mathrm{T}}$, the coarse grid discrete equation is,

$$\tfrac{1}{2}u(\Phi_{H_{i+1}} - \Phi_{H_{i-1}}) - \tfrac{1}{4}|u|(\Phi_{H_{i+1}} - 2\Phi_{H_i} + \Phi_{H_{i-1}}) = 0,$$

where $H = 2h$. Clearly, the stabilization term has been reduced by a factor of 1/2 relative to the (Galerkin) central difference term.

In order to address the loss of stabilization, we introduce a length scaling matrix $\boldsymbol{\sigma}$ to increase the coarse grid stabilization. We note that Mavriplis has used a similar scaling approach to improve the accuracy of coarse grid discretizations when using nodal agglomeration with low-order transfer operators [1]. The fine grid matrix is split according to Eq. (7) and the coarse grid matrices are computed using the GCA formulation,

$$\mathbf{A}_{k+1} = \mathbf{R}_k\mathbf{A}^{\mathrm{base}}\mathbf{P}_k + \boldsymbol{\sigma}\mathbf{R}_k\mathbf{A}^{\tau}\mathbf{P}_k \equiv \mathbf{A}_{k+1}^{\mathrm{base}} + \mathbf{A}_{k+1}^{\tau},$$

where $\boldsymbol{\sigma}$ is a diagonal scaling matrix which accounts for h-dependency in $\mathbf{A}^{\tau}$. For each coarse grid node, the scaling is the square root of the coarse-to-fine mesh areas associated with the node. Note, this scaling procedure is only fully appropriate for coarse grid nodal control volumes with shapes that are similar to the fine grid nodal control volume. A more appropriate version for semi-coarsening has not been developed yet.

The effect of $\boldsymbol{\tau}$ stabilization scaling is demonstrated by plotting the eigenspectrum of the system matrix $\mathbf{A}_{k=0}$ and the coarse space matrix $\mathbf{A}_{k=1}$ with and without scaling. The test case considered is the linear convection–diffusion equation discretized over a square domain $\Omega =]0,1[^2$ (Fig. 1(a)) with a prescribed velocity field $\mathbf{V} = (-y, x)$ 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, with and without $\boldsymbol{\tau}$ scaling. If the coarse grid is discretized with the proper level of stabilization for the mesh, the real portion of the eigenvalues would be expected to scale with H/h relative to the fine grid. Since numerical stabilization is designed to only introduce dissipation, the imaginary portion of the eigenvalues is set by the model problem and should remain largely unchanged by the rediscretization on the coarse mesh. This behavior is clearly observed in the eigenspectrum of the coarse grid with $\boldsymbol{\tau}$ scaling shown in Fig. 3. Furthermore, without $\boldsymbol{\tau}$ scaling, the imaginary portion of the eigenvalues has grown on the coarse grid.

3.5. Results

We consider the linear convection–diffusion equation (Eq. (1)) over a square domain defined by $\Omega =]0,1[^2$ (Fig. 4) and prescribed velocity field $\mathbf{V} = (-y, x)$. The forcing function f is set to 0, the Neumann outflow condition on the upper boundary is $g_{\mathrm{N}} = 0$ and the Dirichlet conditions on the other boundaries are,

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

The discretized domain is adapted on the $x = 0$ boundary to capture the boundary layer as shown in Fig. 1(a). All presented results are based on a $V(1,1)$ multigrid cycle using the line Gauss–Seidel preconditioner. The line preconditioner was found to be stable for this problem even without multistaging. Thus, all results presented for this example are based on a single-stage formulation with $\omega = 0.95$.

The dependence of the convergence rate on the mesh size is shown in Fig. 5(a). Three meshes with increasing mesh sizes of 3849, 15,763 and 60,399 points were used. Good convergence properties are observed with some departure for the largest mesh. The dependence of the convergence rate on the Peclet number is shown in Fig. 5(b) for a range of Reynolds numbers from 10^2 to 10^6 on the 60,399 node mesh. Clearly, the algorithm works well for a wide range of Peclet numbers maintaining a nearly constant number

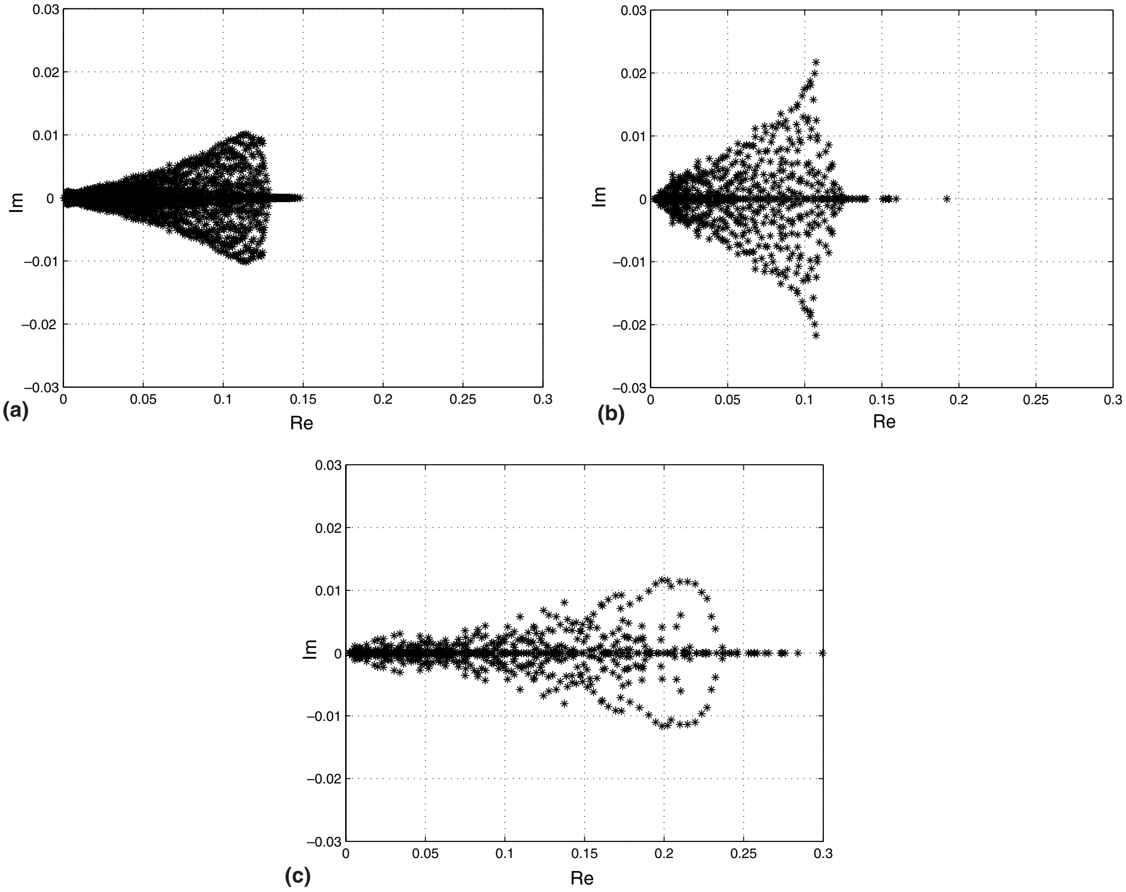

Fig. 3. Comparison of scalar convection–diffusion equation eigenspectrum with and without τ scaling for $Pe = 1e6$ (see Fig. 4 for a description of the problem): (a) fine grid matrix $\mathbf{A}_0$, (b) coarse grid matrix $\mathbf{A}_1$ without τ scaling and (c) coarse grid matrix $\mathbf{A}_1$ with τ scaling.

of iterations for convergence. The average convergence rates ($\bar{\epsilon}$) and asymptotic convergence rates (ϵ) for the different test cases are summarized in Table 2. The average multigrid convergence rate is defined as

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

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 five multigrid cycles.

4. Application to compressible flow

We now consider the application of the proposed AMG algorithm to a stabilized finite element discretization of the Navier–Stokes equations. The discretization used for the compressible flow equations was provided by Wong et al. [36]. The steady 2D compressible flow equations in conservative form can be written,

$$\mathbf{F}_1(\mathbf{V})_{,1} + \mathbf{F}_2(\mathbf{V})_{,2} = \mathbf{F}^v_1(\mathbf{V})_{,1} + \mathbf{F}^v_2(\mathbf{V})_{,2},$$

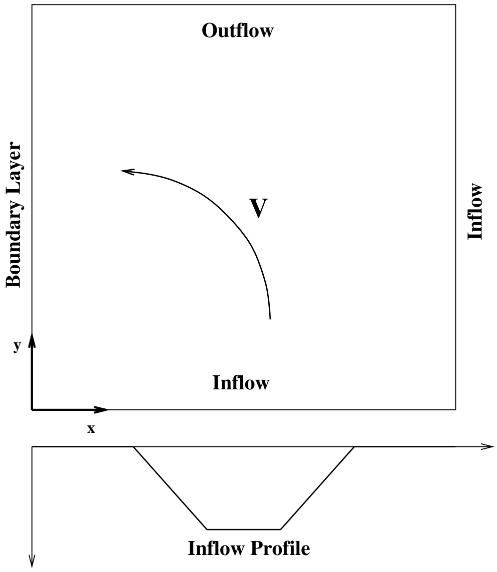

Fig. 4. Computational domain for scalar convection–diffusion boundary layer problem.

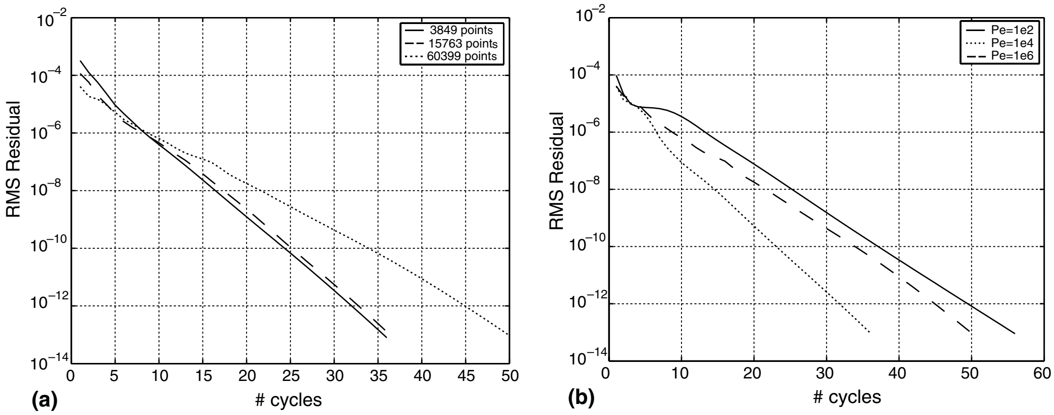

Fig. 5. Dependence of convergence on mesh size and Peclet number for convection–diffusion problem (see Fig. 4 for a description of the problem): (a) mesh size dependency for $Pe = 10^6$ and (b) Peclet dependency for 60,399 node mesh.

where

$$\mathbf{F}_1 = \left\{ \begin{array}{c} \rho u_1 \\ \rho u_1^2 + p \\ \rho u_1 u_2 \\ u_1(\rho E + p) \end{array} \right\}, \quad \mathbf{F}_2 = \left\{ \begin{array}{c} \rho u_2 \\ \rho u_1 u_2 \\ \rho u_2^2 + p \\ u_2(\rho E + p) \end{array} \right\},$$

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

Table 2
Convergence results for scalar convection–diffusion problem using a $V(1,1)$ multigrid cycle and a single-stage line Gauss–Seidel relaxation (see Fig. 4 for a description of the problem)

Grid size	*Pe*	Average convergence $\bar{\epsilon}$	Asymptotic convergence ϵ
3849	10^2	0.31	0.29
15,763	10^2	0.47	0.50
60,399	10^2	0.67	0.69
3849	10^4	0.53	0.57
15,763	10^4	0.52	0.55
60,399	10^4	0.55	0.58
3849	10^6	0.52	0.54
15,763	10^6	0.54	0.54
60,399	10^6	0.65	0.63

such that $\mathbf{F}_i$ are the inviscid fluxes, $\mathbf{F}_i^v$ are the viscous fluxes, ρ is the density, $\mathbf{u} = \{u_1, u_2\}^{\mathrm{T}}$ 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 where $\gamma = 1.4$ is the ratio of specific heats. The viscous stresses and heat transfer are,

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k}\right),$$

$$q_i = -k\frac{\partial T}{\partial x_i},$$

where μ is the dynamic viscosity and k the thermal conductivity, both of which are assumed to be constant. The entropy variables $\mathbf{V}$ are used for the dependent variables [37–39], specifically,

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

Barth [38] has shown that the use of entropy variables leads to a non-linear stability condition for these stabilized discretizations. The GLS algorithm can then be written as: Find $\mathbf{V}_h \in \mathscr{V}^h$ such that for all $\mathbf{W} \in \mathscr{V}^h$,

$$B(\mathbf{V}_h, \mathbf{W})_{\mathrm{gal}} + B(\mathbf{V}_h, \mathbf{W})_{\mathrm{gls}} + B(\mathbf{V}_h, \mathbf{W})_{\mathrm{bc}} = 0,$$

where

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

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

$$B(\mathbf{V}_h, \mathbf{W})_{\text{gls}} = \int_{\Omega} \{(\mathbf{F} - \mathbf{F}^v)_1(\mathbf{W}) + (\mathbf{F} - \mathbf{F}^v)_2(\mathbf{W})\} \cdot \boldsymbol{\tau}\{(\mathbf{F} - \mathbf{F}^v)_1(\mathbf{V}_h) + (\mathbf{F} - \mathbf{F}^v)_2(\mathbf{V}_h)\} \, \mathrm{d}\Omega,$$

$$B(\mathbf{V}_h, \mathbf{W})_{\text{bc}} = \int_{\Gamma \setminus \Gamma_{\text{w}}} \mathbf{W} \cdot (\mathbf{F}_{\text{ff}})(\mathbf{V}_h, \mathbf{g}; \mathbf{n}) \, \mathrm{d}s.$$

The domain boundary is separated into an impermeable solid wall Γ_{w}, and a computational far field boundary $\Gamma \setminus \Gamma_{\text{w}}$. For the Navier–Stokes equations, no-slip Dirichlet boundary conditions replace the momentum equations over the solid wall, and the wall is assumed adiabatic. On the far field boundary, a characteristic-based boundary condition is imposed from a known farfield state, $\mathbf{g}$ through an upwind flux-function $\mathbf{F}_{\text{ff}}$.

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

$$\boldsymbol{\tau}^{-1} = \boldsymbol{\tau}_{\text{i}}^{-1} + \boldsymbol{\tau}_{\text{v}}^{-1},$$

where $\boldsymbol{\tau}_{\text{i}}$ is the inviscid stabilization matrix by Wong et al. [36] and $\boldsymbol{\tau}_{\text{v}}$ is a viscous modification as developed by Shakib et al. [40].

4.1. AMG extension to the Navier–Stokes equations

The solution strategy employed for the non-linear system of equations is a damped Newton scheme in which multigrid is used to solve the linear system at each Newton iteration. The agglomeration and coarse grid discretizations are performed as described previously for the convection–diffusion application. The line preconditioner is extended to the system case by solving the local blocks for each node in a coupled manner. The coupling matrix used to form the lines is chosen as the stiffness matrix resulting from a GLS rediscretization of the stationary linear convection equation using the velocity field $\mathbf{V} = (u_1, u_2)$ from the current Newton iteration. The implicit lines for the inviscid flow around a NACA 0012 airfoil at a zero angle of attack and a freestream Mach number of 0.1 are shown in Fig. 6. Also, we have found that symmetric line

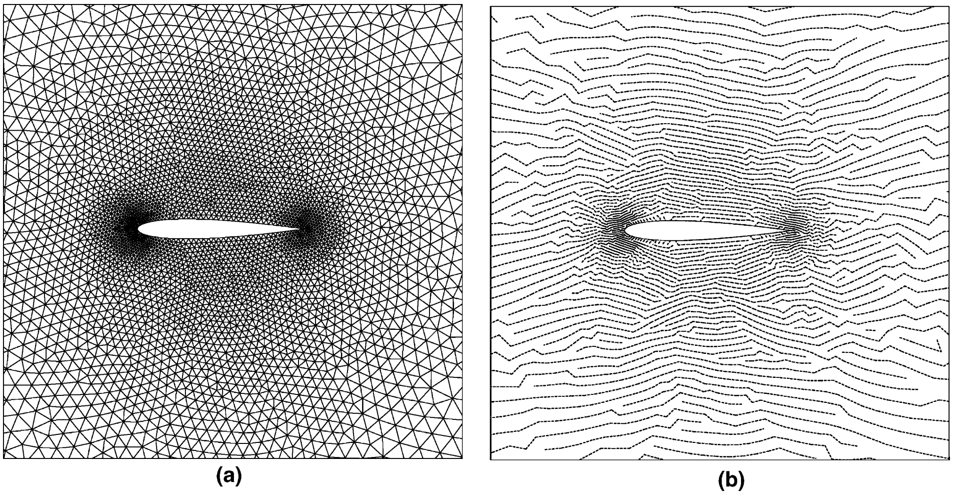

Fig. 6. Grid and implicit lines for inviscid flow simulation around a NACA 0012 airfoil: (a) grid and (b) lines for inviscid flow at Mach 0.1.

Gauss–Seidel (i.e. sweeping over the lines symmetrically from first-to-last and then from last-to-first) provides advantages in the robustness and convergence rate of the multigrid algorithm to justify the doubled cost relative to the uni-directional Gauss–Seidel sweep for Euler and Navier–Stokes solutions.

For both the Euler and Navier–Stokes solutions, the termination condition for the linear multigrid solver is chosen to be when the linear residual is the square of the non-linear residual. This ensures that Newton quadratic convergence is achieved without requiring the linear multigrid solution to converge to machine precision.

4.2. Euler 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 study of dependency on the grid size and Mach number.

The first case consists of a channel flow with a sine-squared bump of 5% height on the lower wall as shown in Fig. 7. The domain is discretized into a structured grid of quadrilaterals which are then triangulated. A $V(2,1)$ multigrid cycle is used with a 3-stage multistage scheme and the symmetric line Gauss–Seidel preconditioner. Table 3 shows the asymptotic convergence rate ϵ for a number of fine grid sizes and Mach numbers. As can be observed from Table 3, excellent convergence rates (below 0.1) are achieved for all cases.

The second test case consists of external flow around a NACA 0012 airfoil at angle of attack of $\alpha = 0°$ and 3°. A typical mesh for this test case is shown in Fig. 6(a). Both the 3-stage and 5-stage schemes were run for this test case, and a $V(2,1)$ multigrid cycle was used.

Table 4 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 elements n_k^e and vertices n_k^v per grid level, these complexities for an m-grid (total of fine + coarse) problem are defined as,

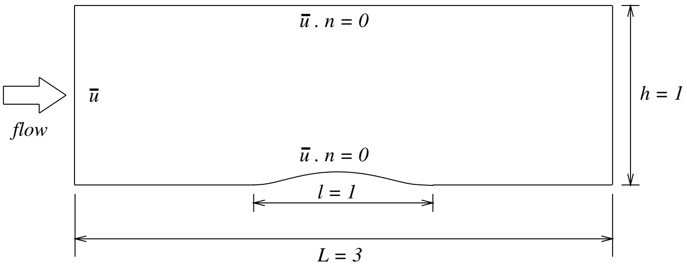

Fig. 7. Channel flow geometry for compressible, inviscid flow example.

Table 3
Asymptotic linear multigrid convergence rates for compressible channel flow using a $V(2,1)$ multigrid cycle and a 3-stage scheme with symmetric line Gauss–Seidel preconditioning

Fine grid size	# of coarse grids	$\epsilon : M = 0.1$	$\epsilon : M = 0.5$
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
Asymptotic linear multigrid convergence rates for compressible Euler flow over a NACA 0012 airfoil using a $V(2,1)$ multigrid cycle and symmetric line Gauss–Seidel preconditioning

Fine grid size	Coarse grids	TVC	TEC	$\epsilon_{M=0.1}$ $(\alpha = 0°)$		$\epsilon_{M=0.5}$ $(\alpha = 0°)$		$\epsilon_{M=0.5}$ $(\alpha = 3°)$	
				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
10,273	4	1.67	1.36	0.28	0.17	0.25	0.24	0.24	0.17
20,621	5	1.64	1.36	0.33	0.26	0.34	0.21	0.23	0.21

$$\mathrm{TEC} = \sum_{k=0}^{m} \frac{n_k^e}{n_0^e},$$

$$\mathrm{TVC} = \sum_{k=0}^{m} \frac{n_k^v}{n_0^v}.$$

These complexities can be used as a measure of the overall work and storage required for the multigrid algorithm in comparison to the fine grid calculation. 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, excellent and relatively mesh independent convergence rates are achieved. The vertex complexities show that the total storage for the multigrid algorithm is about 65% greater than the fine mesh computation. Fig. 8 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.

4.3. Navier–Stokes results

The final test case is the laminar flow over a NACA 0005 airfoil at a Reynolds number of 5000. The grid, shown in Fig. 9, has stretching in the boundary layer where the largest cell aspect ratio is of the order of 500. The $V(2,1)$ multigrid cycle is used again with a 5-stage scheme and symmetric line Gauss–Seidel preconditioner.

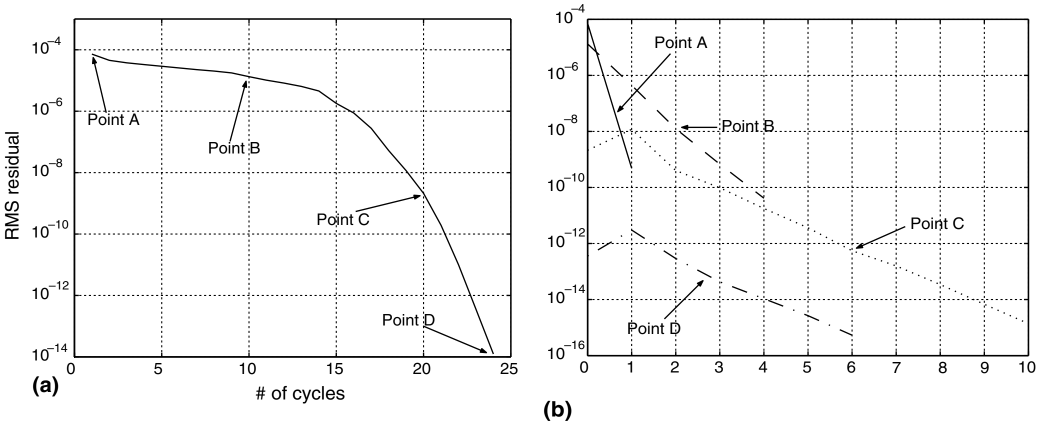

Fig. 8. Non-linear Newton outer loop and linear multigrid convergence histories for compressible Euler flow over a NACA 0012 airfoil with 20,621 fine grid nodes at Mach 0.5 and zero angle of attack using a $V(2,1)$ multigrid cycle and a 5-stage scheme with symmetric line Gauss–Seidel preconditioning: (a) non-linear convergence history and (b) linear convergence history.

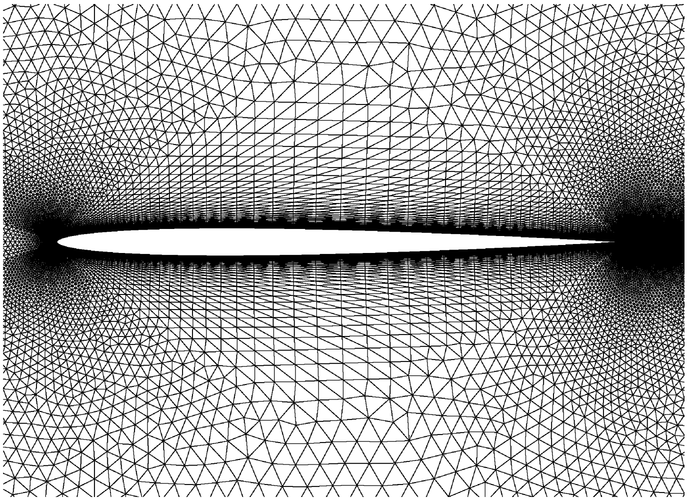

Fig. 9. Unstructured mesh for viscous simulation on a NACA 0005 airfoil.

Table 5
Asymptotic linear multigrid convergence rates for compressible Navier–Stokes flow over a NACA 0005 airfoil at Reynolds number 5000 using a $V(2,1)$ multigrid cycle and a 5-stage scheme with symmetric line Gauss–Seidel preconditioning

Fine grid size	Coarse grids	TVC	TEC	$\epsilon_{M=0.1}$ ($\alpha = 0°$)	$\epsilon_{M=0.5}$ ($\alpha = 0°$)	$\epsilon_{M=0.5}$ ($\alpha = 3°$)
8872	2	1.64	1.37	0.16	0.37	0.21
18,416	3	1.88	1.44	0.25	0.37	0.57
36,388	4	1.67	1.37	0.35	0.35	0.45

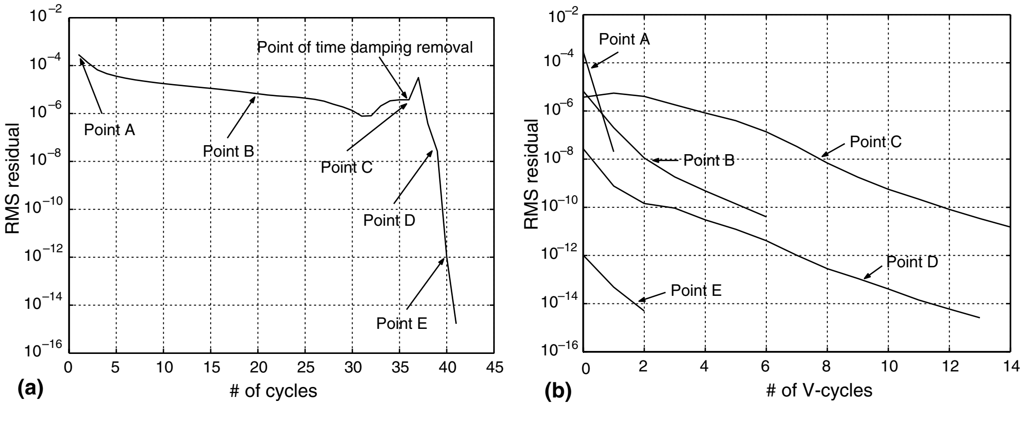

Fig. 10. Non-linear Newton outer loop and linear multigrid convergence histories for compressible Navier–Stokes flow over a NACA 0005 airfoil at Mach 0.5, Reynolds number 5000, and zero angle of attack with 36,388 fine grid nodes using a $V(2,1)$ multigrid cycle and a 5-stage scheme with symmetric line Gauss–Seidel preconditioning: (a) non-linear convergence history and (b) linear convergence history.

During the initial stages of the Newton iteration, it was found that large transient growths occurred in the solution preventing convergence. As a result, timestepping was introduced for the initial iterations. In the later iterations, the timestepping was eliminated and the Newton method was recovered. The asymptotic convergence rates for the linear multigrid solver are reported only for the final Newton iterations once the timestepping is eliminated. Table 5 shows the asymptotic convergence rate ϵ as well as the total element and vertex complexities for a sequence of independently generated fine grid sizes. As can be observed from Table 5, excellent convergence rates are achieved. Also, the vertex complexity is still reasonably low, despite the semi-coarsening, increasing the storage cost of the multigrid algorithm by at most 88% relative to the fine mesh. Fig. 10 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 and zero angle of attack. After 35 non-linear iterations, the timestepping was removed with a corresponding jump in the residual followed by Newton convergence as shown in Fig. 10.

5. Conclusion

A multigrid method has been developed for solving convection-dominated flows using stabilized finite element discretizations. The algorithm combines a line preconditioner with semi-coarsening using an algebraic multigrid approach in which elements are agglomerated. We demonstrate that numerical stabilization on the coarse grid scales with the fine grid spacing when accurate prolongation and restriction operators are used. A modification to the algebraic coarse grid discretization is proposed allowing the stabilization to scale appropriately with the coarse grid size. The multigrid convergence rates were shown to be very insensitive to both mesh size and Reynolds number. While the current algorithm is demonstrated only for linear elements, on-going work is investigating the extension of the basic approach to higher-order elements.

Acknowledgements

The first author wishes to acknowledge the financial support provided by the NASA Dryden Flight Research Center under research contract NAG4-157, a grant by NSF CAREER and the Singapore-MIT Alliance. Much appreciation is also given to Dr. Dimitri Mavriplis for his constructive evaluation and input.

References

[1] D.J. Mavriplis. Multigrid techniques for unstructured meshes. Technical Report VKI-LS 1995-02, Von Karman Institute, March 1995.

[2] D.J. Mavriplis, Directional agglomeration multigrid techniques for high Reynolds number viscous flows, AIAA J. 37 (1999) 1222–1230.

[3] D.J. Mavriplis, S. Pirzadeh, Large-scale parallel unstructured mesh computations for 3D high-lift analysis, AIAA J. Aircraft 36 (1999) 987–998.

[4] D.J. Mavriplis, An assessment of linear versus nonlinear multigrid methods for unstructured mesh solvers, J. Comput. Phys. 175 (2001) 302–325.

[5] N. Pierce, Preconditioned Multigrid Methods for Compressible Flow Calculation on Stretched Meshes, Ph.D. thesis, University of Oxford, 1997.

[6] N.A. Pierce, M.B. Giles, Preconditioned multigrid methods for compressible flow calculations on stretched meshes, J. Comput. Phys. 136 (1997) 425–445.

[7] S.R. Elias, G.D. Stubley, G.D. Raithby, An adaptive agglomeration method for additive correction multigrid, Int. J. Numer. Methods Engrg. 40 (1997) 887–903.

[8] B.R. Hutchinson, G.D. Raithby, A multigrid method based on the additive correction strategy, Numer. Heat Transfer 9 (1986) 511–537.

[9] M. Raw, Robustness of coupled algebraic multigrid for the Navier–Stokes equations, AIAA J. 1 (1996) 1–16.

[10] R. Webster, An algebraic multigrid solver for Navier–Stokes problems, Int. J. Numer. Methods Fluids 18 (1994) 761–780.

[11] M.H. Lallemand, H. Steve, A. Dervieux, Unstructured multigridding by volume agglomeration: current status, Comput. Fluids 21 (1992) 397–433.

[12] B. Koobus, M.H. Lallemand, A. Dervieux, Unstructured volume agglomeration multigrid: solution of the Poisson equation, Int. J. Numer. Methods Fluids 18 (1994) 27–42.

[13] J. Jones, P.S. Vassilevski, AMGe based on element agglomeration, SIAM J. Sci. Comput. 23 (2001) 109–133.

[14] T.F. Chan, S. Go, L. Zikatanov, Multilevel methods for elliptic problems on unstructured grids, Technical Report VKI-LS 1997-02, Von Karman Institute, March 1997.

[15] T.F. Chan, J. Xu, L. Zikatanov, An agglomeration multigrid method for unstructured grids, Technical Report 98-8, Computational and Applied Mathematics (CAM)/UCLA, February 1998.

[16] M. Brezina, A.J. Cleary, R.D. Falgout, V.E. Henson, J.E. Jones, T.A. Manteuffel, S.F. McCormick, J.W. Ruge, Algebraic multigrid based on element interpolation (AMGe), SIAM J. Sci. Comput. 22 (2000) 1570–1592.

[17] J.W. Ruge, K. Stüben, Algebraic multigrid (AMG), in: S.F. McCormick (Ed.), Multigrid Methods, Frontiers in Applied Mathematics, vol. 3, SIAM, Philadelphia, PA, 1987, pp. 73–130.

[18] A.N. Brooks, T.J.R. Hughes, Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 32 (1982) 199–259.

[19] T.J.R. Hughes, L.P. Franca, G.M. Hulbert, A new finite element formulation for computational fluid dynamics: VIII. The Galerkin least-squares method for advective diffusive equations, Comput. Methods Appl. Mech. Engrg. 73 (1989) 173–189.

[20] F. Shakib, Finite Element Analysis of the Compressible Euler and Navier–Stokes Equations, Ph.D. thesis, Dept. of Mech. Engineering, Stanford, November 1988.

[21] P. Wesseling, An Introduction to Multigrid Methods, John Wiley & Sons, Chichester, 1992.

[22] J. Bey, A. Reusken, On the convergence of basic iterative methods for convection–diffusion equations, Numer. Lin. Algebra Appl. 6 (1999) 329–352.

[23] A. Jameson, Solution of the Euler equations by a multigrid method, Appl. Math. Comput. 13 (1983) 327–356.

[24] A. Jameson, W. Schmidt, E. Turkel, Numerical solution of the Euler equations by finite volume methods using Runge–Kutta time stepping schemes, in: AIAA 14th Fluid and Plasma Dynamic Conference, AIAA Paper Number 81-1259, 1981.

[25] A. Jameson, Numerical solution of the Euler equations for the compressible inviscid fluids, in: F. An-grand, A. Dervieux, J.A. Desideri, R. Glowinski (Eds.), Numerical Methods for the Euler Equations of Fluid Dynamics, Proceedings in Applied Mathematics, vol. 21, SIAM, Philadelphia, 1985, pp. 199–245.

[26] S.R. Allmaras, Multigrid for the 2-D compressible Navier–Stokes equations, in: AIAA 14th Computational Fluid Dynamics Conference, AIAA Paper Number 99-3336-CP, 1999.

[27] B. van Leer, W.T. Lee, P.L. Roe, K.G. Powell, C.H. Tai, Design of optimally smoothing multistage schemes for the Euler equations, Commun. Appl. Numer. Methods 8 (1992) 761–769.

[28] C.H. Tai, Acceleration Techniques for Explicit Euler Codes, Ph.D. thesis, Dept. of Aero. Engineering, University of Michigan, 1990.

[29] J. Lynn, Multigrid Solution of the Euler Equations with Local Preconditioning, Ph.D. thesis, Dept. of Aero. Engineering, University of Michigan, 1995.

[30] D.J. Mavriplis, Multigrid strategies for viscous flow solvers on anisotropic unstructured meshes, J. Comput. Phys. 145 (1998) 141–165.

[31] T.F. Chan, P. Vanek, Detection of strong coupling in algebraic multigrid solvers, Technical Report 00-10, Computational and Applied Mathematics (CAM)/UCLA, March 2000.

[32] A. Brandt, Multi-level adaptive techniques (MLAT) for partial differential equations: ideas and software, in: J.R. Rice (Ed.), Mathematical Software III, Academic Press, New York, 1977, pp. 277–318.

[33] V.E. Henson, P.S. Vassilevski, Element-free AMGe: general algorithms for computing interpolation weights in AMG, SIAM J. Sci. Comput. 23 (2001) 629–650.

[34] T.F. Chan, P. Vanek, Multilevel algebraic elliptic solvers, Technical Report 99-9, Computational and Applied Mathematics (CAM)/UCLA, February 1999.

[35] V. Venkatakrishnan, D.J. Mavriplis, Agglomeration multigrid for the three-dimensional Euler equations, AIAA J. 33 (1995) 633–640.

[36] J.S. Wong, D.L. Darmofal, J. Peraire, The solution of the compressible Euler equations at low Mach numbers using a stabilized finite element algorithm, Comput. Methods Appl. Mech. Engrg. 190 (2001) 5719–5737.

[37] T.J.R. Hughes, L.P. Franca, 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, Comput. Methods Appl. Mech. Engrg. 54 (1986) 223–234.

[38] T. Barth, Numerical methods for gasdynamic systems on unstructured meshes, in: An Introduction to Recent Developments in Theory and Numerics Conservation Laws, Lecture Notes in Computational Science and Engineering, 1998, pp. 195–284.

[39] A. Harten, On the symmetric form of systems of conservation laws with entropy, J. Comput. Phys. 49 (1983) 151–164.

[40] F. Shakib, T.J.R. Hughes, Z. Johan, A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 89 (1991) 141–219.