An Algebraic Two-Level Preconditioner for Stabilized Finite Element Formulations

Thomas E. Giddings
Metron Inc., Reston, VA 20190-5602

Jacob Fish
Rensselaer Polytechnic Institute, Troy N.Y. 12180-3590

Abstract

A two-level, linear algebraic solver for asymmetric, positive definite systems is developed, using matrices arising from stabilized finite element formulations to motivate the approach. Based on the analysis of a representative smoother, the parent space is divided into oscillatory and smooth subspaces according to the eigenvectors of the associated normal system. Using an aggregation technique, a restriction/prolongation operator is constructed, which relies only on information contained in the matrix. Various numerical examples, on both structured and unstructured meshes, are performed using the two-level cycle as the basis for a preconditioner. Results demonstrate the complementarity between the smoother and the coarse-level correction.

1 Introduction

Stabilized finite element formulations, commonly used in computational fluid dynamics (CFD), produce positive definite algebraic systems; but, these systems are generally asymmetric due to the presence of first-order derivatives in the model equations.

Practical finite element simulations of fluid flow problems often generate enormous algebraic systems which require very efficient solution strategies in order to render such applications feasible. So there is a clear need for robust, highly efficient solvers for asymmetric, positive definite systems. And multilevel techniques, which ideally achieve linear scaling between computational work and problem size, represent the elusive grail of linear algebraic solvers.

Since their introduction in the seminal paper by Fedorenko [18] in 1962, the literature on multilevel methods has exploded. Many early results and the basic theoretical and computational foundations of the multilevel approach can be found in texts by Hackbusch [21] and Wesseling [31], and articles by Brandt [6] and Stüben & Trottenberg [26]. An outline of a basic multilevel methodology is given in the next section, where the focus is maintained on the algebraic problem, and algebraic manipulations.

Multilevel methods are based on a two-level kernel, which is used recursively to achieve the multilevel structure. On the higher level, a so-called "smoother" is applied to reduce certain components of the error. Then a "restriction" operator is used to project the approximate solution down to the next lowest level—the "coarse level" in the two-level paradigm—where additional components of the error are reduced to complement the smoother. Following this correction on the coarse level, the updated approximation is injected using a "prolongation" operator back into the higher-level space where, again, the smoother is applied.

Smoothers are computationally inexpensive and generally pulled off the shelf, as it were. Gauss-Seidel, Kaczmarz and variants of SOR methods are commonly used as smoothers for symmetric, positive definite problems. Kettler [23] and Wesseling [30] demonstrated that incomplete LU (ILU) decompositions are effective and robust for a variety of problems, including those involving asymmetric, positive definite systems. But, what really distinguishes different multilevel approaches is the method used to generate the coarse-level matrix and the nature of restriction and prolongation operators, which combine to specify the transfer between different levels.

Multilevel methods are classified as either "geometric" or "algebraic" depending on how the coarse-level matrix is formed and how the components of the error are separated and classified. With the geometric approach, restriction and prolongation are most often accomplished by averaging and interpolating, respectively. The coarse-level problem is then formed by coarsening the underlying approximation space used to discretize the continuous problem, so that the error corrections are progressively smoother in appearance as one proceeds to lower levels; and, the coarse-level matrix results from a rediscretization using a coarser approximation space. In this sense, geometric multilevel methods are tied directly to the discretization of the problem, which presents many numerical difficulties and limitations (cf. reference [8]).

It is often preferable to deal directly with the algebraic problem once it has been formed at the highest level. Ideally, this leads to a purely algebraic approach that allows us to separate the algebraic solver from the rest of the problem. Because algebraic multilevel methods only require that the coarse-level variables satisfy fairly flexible algebraic criteria, the problem can be restricted based on any subspace; and, the coarse-level matrix is formed as a matrix product. Also, algebraic multilevel theory can be abstracted from the underlying differential operator and the details of its discretization, as with the mathematical framework described by McCormick [24].

The motivation and many of the basic principles behind Algebraic Multi-Grid (AMG) methods are described by Brandt *et al.* [8]. A theoretical study is given by Brandt [7] for the particular variant of AMG proposed in [8], which also establishes some basic requirements for multilevel success from the algebraic perspective. Algebraic Multigrid, as advanced in [8] and [7], demonstrates that the smoothing procedure is ineffective for positive definite systems when the residual is small relative to the error, and consequently restricts the algebraic problem by averaging in the directions of the strong couplings between unknowns, as measured by the relative magnitudes of the off-diagonal matrix entries.

The Black Box Multigrid of Dendy [15, 16, 17] exploits the structure of nested,

uniform rectangular grids, and provides a prolongation operator based on the coefficients of the finite differencing operator used to discretize the differential operator. The restriction operator is the transpose of the prolongation operator, and the coarse-level matrix is obtained by a matrix product involving the higher-level matrix and the restriction and prolongation matrices. However, the implicit reliance on grid structuring to formulate the prolongation operator severely limits the practical applications of this approach. Similar limitations apply to the multilevel proconditioner proposed by Axelsson & Vassilevski [1, 2] and the blackbox multigrid solver of De Zeeuw [14], among many others, which rely on nested, structured, and interelated meshes.

One of the important challenges of multilevel methods involves the use of unstructured meshes, or nonuniform p-refinement, which leads to an irregular structure of the matrix entries. Bulgakov [10, 11], Bulgakov & Kuhn [12], and Vaněk, *et al.* [27, 28] employ aggregation techniques whereby adjacent finite element domains, and the degrees of freedom attached to them, are clustered together. The unknowns associated with each cluster can be combined to form a smaller set of coarse-level variables, which define the restriction operator; and, its transpose provides the prolongation operator. Again, the coarse-level matrix is computed as a matrix product.

Fish & Belsky [19] use an aggregation technique as part of a two-level solver for symmetric, positive definite systems. It is noted that the smoother acts to reduce the components of the error in the directions of the eigenvectors of the system matrix associated with the largest eigenvalues, which was also observed by Brandt [7]. The restriction operator is based on the eigenspace of the submatrices associated with the aggregates of elements. In the present paper, we extend the method of [19] to problems involving asymmetric, positive definite systems, where the eigenvectors of the system matrix do not necessarily separate the subspaces of interest.

While the multilevel methodology may be used by itself to invert the algebraic system, it is often most effective when used as a preconditioner for an appropriate iterative accelerator. It is noted that several multilevel cycles effectively invert the

matrix, so that one or two cycles represent an approximate inverse to this same matrix, and can be used to precondition the original system. The preconditioned system can then be solved using a projection-type iterative technique, such as GMRES or TFQMR (cf. Saad [25] or Hackbusch [22]). This is demonstrated in the examples of section 6.

2 Overview of Multilevel Methodology

Consider the linear algebraic system of equations,

$$\mathbf{K}\mathbf{u} = \mathbf{f} \tag{2.1}$$

with the solution $\mathbf{u} \in \mathbb{R}^N$, and the right-hand-side vector $\mathbf{f} \in \mathbb{R}^N$. The system matrix $\mathbf{K} \in \mathbb{R}^N \times \mathbb{R}^N$ is positive definite, but generally asymmetric.

If we take the regular splitting

$$\mathbf{K} = \mathbf{M} - \mathbf{N}, \tag{2.2}$$

where $\mathbf{M}, \mathbf{N} \in \mathbb{R}^N \times \mathbb{R}^N$, we can construct a (stationary) relaxation scheme whereby the successive approximations are updated recursively in the following way:

$$\mathbf{u}^{i+1} = (\mathbf{I} - \mathbf{M}^{-1}\mathbf{K})\,\mathbf{u}^i + \mathbf{M}^{-1}\mathbf{f}, \tag{2.3}$$

the superscripts "i" and "$i+1$" indicating the iteration numbers, so that $\mathbf{u}^i$ is the approximation after the i^{th} iteration. The matrix $\mathbf{M}$ is chosen so that it is computationally inexpensive to invert. (cf. the standard references by Varga [29] and Young [32], and a nice summary in Ciarlet [13], §5.)

If we specify the exact solution as $\mathbf{u}$, and the error after iteration i as $\mathbf{e}^i = \mathbf{u}^i - \mathbf{u}$,

then the relaxation scheme (2.3) is equivalent to the following iterations on the error:

$$\mathbf{e}^{i+1} = (\mathbf{I} - \mathbf{M}^{-1}\mathbf{K})\,\mathbf{e}^i = \mathbf{S}\,\mathbf{e}^i \tag{2.4}$$

The matrix $\mathbf{M}^{-1}$ is called the *preconditioner*, since the relaxation scheme (2.3) is equivalent to fixed-point iterations on the preconditioned system $\mathbf{M}^{-1}\mathbf{K}\mathbf{u} = \mathbf{M}^{-1}\mathbf{f}$. The matrix $\mathbf{S} = (\mathbf{I} - \mathbf{M}^{-1}\mathbf{K})$ in equation (2.4) is called the *iteration matrix*, and it is the so-called "smoothing" property of this matrix operator that is important in the multilevel context.

The main drawback of simple relaxation schemes of the form given in equation (2.3) is that they typically fail to reduce all components of the error with equal effectiveness. A pertinent example is the homogeneous, one-dimensional advection problem

$$\frac{\partial u}{\partial x} - \kappa \frac{\partial^2 u}{\partial x^2} = 0 \quad \text{on} \quad \Omega = (0,1)$$
$$u(0) = g_1 \quad \text{and} \quad u(1) = g_2$$

as $\kappa \to 0$, which we discretize using the SUPG stabilized finite element formulation of Brooks & Hughes [9] and linear elements. Figure 1 shows: (a) the initial error; (b) the error after five iterations of the weighted Jacobi relaxation scheme (with $\omega = 2/3$); and, (c) the error after five more iterations.

This example, as Figure 1 shows, demonstrates how the "oscillatory" components of the error can be eliminated in only a few iterations, whereas the "smooth" components of the error are reduced very slowly. While simple relaxation schemes may not converge to the solution quickly, if at all, they can often be used as a *smoother* to eliminate the oscillatory components of the error very efficiently. multilevel methods seek to complement this smoothing behavior by addressing the smooth components of the error separately.

We note that the overall, parent space $\mathbb{R}^N$ can be divided into two subspaces $\mathbb{A}$

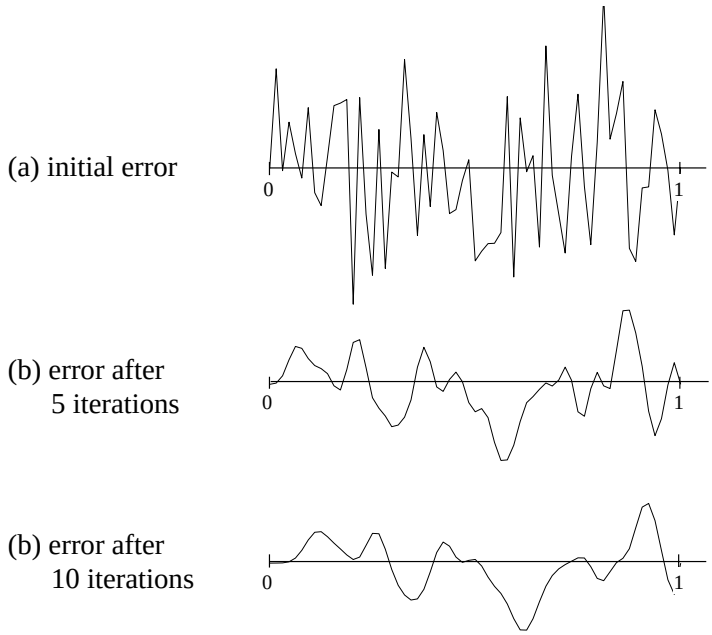


Figure 1: Weighted Jacobi iterations

and $\mathbb{B}$,

$$\mathbb{R}^N = \mathbb{A} \oplus \mathbb{B}, \tag{2.5}$$

where $\mathbb{A} \subset \mathbb{R}^N$ is of dimension $M < N$, and $\mathbb{B} \subset \mathbb{R}^N$ is of dimension $N - M$. We will call $\mathbb{A}$ the *smooth*, or *coarse*, subspace, and $\mathbb{B}$ is the *oscillatory* subspace. These two subspaces are not necessarily orthogonal.

We can construct a basis $\{\mathbf{q}_i\}$ for the smooth subspace $\mathbb{A}$ where $\mathbf{q}_i \in \mathbb{R}^N$, so that:

$$\mathbb{A} = \text{span}\{\mathbf{q}_i\} \qquad \text{for } i = 1, 2, \ldots, M \tag{2.6}$$

If we define the matrix $\mathbf{Q}$ to be the $N \times M$ matrix whose columns are the vectors $\mathbf{q}_i$, we can uniquely specify an arbitrary vector $\mathbf{v} \in \mathbb{A}$ as

$$\mathbf{v} = \sum_{i=1}^{M} a_i \mathbf{q}_i = \mathbf{Q}\,\mathbf{a} \tag{2.7}$$

where $\mathbf{Q} = [\mathbf{q}_1\ \mathbf{q}_2\ \cdots\ \mathbf{q}_M] \in \mathbb{R}^N \times \mathbb{R}^M$, $\mathbf{a} = (a_1, a_2, \cdots, a_M)^t \in \mathbb{R}^M$, and the superscript "$t$" indicates the transpose. The matrix $\mathbf{Q} : \mathbb{R}^M \to \mathbb{R}^N$ is the *prolongation* operator; and its transpose, $\mathbf{Q}^t : \mathbb{R}^N \to \mathbb{R}^M$, is the *restriction* operator. These two operators, $\mathbf{Q}$ and $\mathbf{Q}^t$, allow us to transfer between the parent space $\mathbb{R}^N$ and the smooth, or coarse, subspace $\mathbb{A}$.

Given an approximate solution $\mathbf{u}^{\nu_1}$ after $i = \nu_1$ smoothing iterations of the form (2.3), we can obtain a corrected approximation $\mathbf{u}^c = \mathbf{u}^{\nu_1} + \mathbf{Q}\mathbf{a}$ which removes certain components of the error so that the corrected residual $\mathbf{R}^c = \mathbf{K}\mathbf{u}^c - \mathbf{f}$ is orthogonal to the smooth subspace $\mathbb{A}$:

$$\langle \mathbf{R}^c, \mathbf{Q} \rangle = \langle \mathbf{K}\mathbf{u}^c - \mathbf{f}, \mathbf{Q} \rangle = 0 \tag{2.8}$$

where $\langle \cdot, \cdot \rangle$ is the standard inner product on $\mathbb{R}^N$. This gives the following *coarse-level problem*:

$$\mathbf{K}_0\, \mathbf{a} = -\mathbf{Q}^t \mathbf{R}^{\nu_1} \tag{2.9}$$

where $\mathbf{R}^{\nu_1} = \mathbf{K}\mathbf{u}^{\nu_1} - \mathbf{f}$ is the residual after ν_1 smoothing iterations, and $\mathbf{K}_0 = \mathbf{Q}^t\mathbf{K}\mathbf{Q}$ is the coarse-level matrix.

The solution to the coarse-level problem (2.9) is then:

$$\mathbf{a} = -\mathbf{K}_0^{-1}\, \mathbf{Q}^t \mathbf{R}^{\nu_1} \tag{2.10}$$

so that the corrected approximation is given by:

$$\mathbf{u}^c = \left(\mathbf{I} - \mathbf{Q}\mathbf{K}_0^{-1}\mathbf{Q}^t\mathbf{K}\right)\, \mathbf{u}^{\nu_1} + \mathbf{Q}\mathbf{K}_0^{-1}\mathbf{Q}^t\mathbf{f}. \tag{2.11}$$

The *coarse-level correction* of equation (2.11) is equivalent to the following operation

on the error:

$$\mathbf{e}^c = \left(\mathbf{I} - \mathbf{Q}\mathbf{K}_0^{-1}\mathbf{Q}^t\mathbf{K}\right)\, \mathbf{e}^{\nu_1} = \mathbf{T}\,\mathbf{e}^{\nu_1} \tag{2.12}$$

where the operator

$$\mathbf{T} = \mathbf{I} - \mathbf{Q}\mathbf{K}_0^{-1}\mathbf{Q}^t\mathbf{K}, \tag{2.13}$$

and $\mathbf{e}^{\nu_1} = \mathbf{u}^{\nu_1} - \mathbf{u}$, $\mathbf{e}^c = \mathbf{u}^c - \mathbf{u}$.

The smoothing iterations (2.3) and the coarse-level correction (2.11) can be combined to form a two-level cycle, which is diagrammed in Figure 2. Starting with the same initial error as in the previous example, Figure 2a, we perform only two *presmoothing* iterations using equation (2.3) and a weighted Jacobi smoother. The smoothing effect on the error is given by equation (2.4) and shown in Figure 2b. The *coarse-level correction* is accomplished using equation (2.11), which yields the attenuated but highly oscillatory error shown in Figure 2c. Finally, two *postsmoothing* iterations are performed, using the same smoother, resulting in the error shown in Figure 2d. We note that a significant reduction in all components of the error has been accomplished at the end of the this cycle.

The two-level cycle is equivalent to the following operation on the error:

$$\mathbf{e}^{cycle} = \mathbf{S}^{\nu_2}\,\mathbf{T}\,\mathbf{S}^{\nu_1}\,\mathbf{e}^{cycle-1} \tag{2.14}$$

where ν_1 and ν_2 are the number of presmoothing and postsmoothing iterations, respectively. This two-level methodology can be extended to multiple levels by applying the two-level scheme recursively to the coarse-level problems (cf. Hackbusch [21]). Practical implementations must allow for multiple levels in order to maintain a relatively small problem at the coarsest level, which is solved directly. But important questions can be answered by the two-level scheme, which is the focus of the current paper.

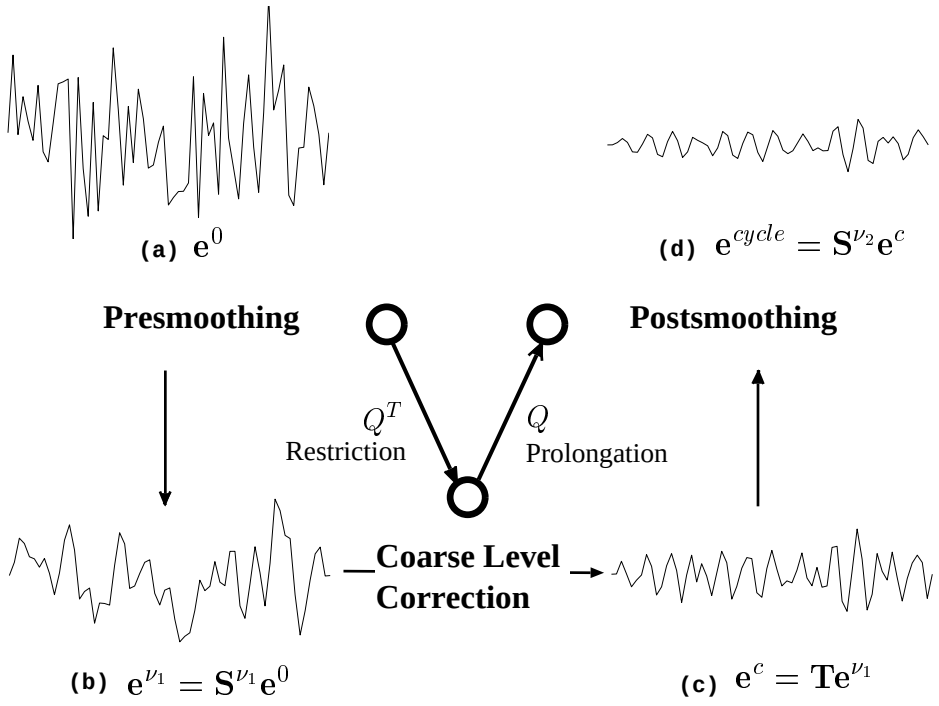


Figure 2: Diagram of Two-level Cycle

An effective multilevel scheme will achieve a similar level of error reduction with each cycle, so that every component of the error is reduced significantly (unlike what happened with the simple relaxation scheme illustrated in Figure 1). There is one important proviso, however: the coarse-level correction of equation (2.11) must complement the smoothing of equation (2.3). That is to say, the coarse-level correction must address those components of the error that are effectively ignored by the smoother. This complementary relationship is precisely what we wish to demonstrate with the two-level scheme. So the first step in designing a multilevel scheme is to characterize the subspace on which the smoother is most effective at reducing the error.

3 Analysis of Smoothing in One Dimension

Once an effective smoother has been chosen, a complementary prolongation operator $\mathbf{Q}$ must be designed; that is, an appropriate basis $\{\mathbf{q}_i\}$ for the coarse subspace $\mathbb{A}$

must be found. The oscillatory subspace $\mathbb{B}$ is, by definition, the subspace on which the smoothing iterations of equation (2.3) are most effective at reducing the error. The coarse subspace can then be viewed as the orthogonal complement of the oscillatory subspace in the parent space $\mathbb{R}^N$, or $\mathbb{A} = \mathbb{R}^N \setminus \mathbb{B}$. So, to design the prolongation operator we must first characterize the oscillatory subspace $\mathbb{B}$ by studying a representative smoother.

To find the directions in which the smoother is most effective at reducing the error, we study the iteration matrix $\mathbf{S}$ of equation (2.4) for a relatively simple model problem on the one-dimensional domain $\Omega = [0, L]$, as depicted in Figure 3. The domain is divided into $N + 1$ segments of equal length $h = L/(N + 1)$, so that $x_j = jh$ where $j = 0, 1, \ldots, N + 1$. We consider the scalar advection-diffusion model equation,

$$c\,\frac{du}{dx} - \kappa\frac{d^2u}{dx^2} = f, \tag{3.1}$$

where c is the advection coefficient (or speed) and κ is the coefficient of diffusion.

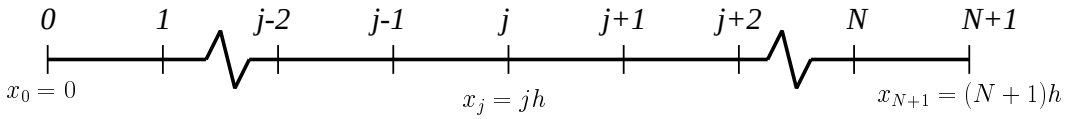


Figure 3: Domain of 1D Model Problem

The *element Peclet number* for a particular discretization is given by $\alpha = \frac{|c|h}{2\kappa}$, and it measures the relative strength of the advective and diffusive mechanisms; whereby, $\alpha \ll 1$ indicates that the second-order diffusion term dominates the problem on the length scale of the elements, and $\alpha \gg 1$ indicates that the first-order advection term dominates the discrete problem. The element Peclet number is the free parameter which delineates the various cases of interest.

Assuming Dirichlet boundary conditions, the SUPG stabilized finite element for-

mulation [9] leads to a linear algebraic problem,

$$\mathbf{K}\mathbf{u} = \mathbf{f} \tag{3.2}$$

where $\mathbf{K} \in \mathbb{R}^N \times \mathbb{R}^N$, and $\mathbf{u}, \mathbf{f} \in \mathbb{R}^N$. The matrix $\mathbf{K}$ is given by:

$$\mathbf{K} = \begin{bmatrix} a+b & -b & & & & \\ -a & a+b & -b & & & \\ & & \ddots & & & \\ & & -a & a+b & -b & \\ & & & -a & a+b \end{bmatrix} \tag{3.3}$$

where

$$a = \frac{\kappa + c^2\tau}{h} + \frac{c}{2}, \qquad b = \frac{\kappa + c^2\tau}{h} - \frac{c}{2}, \qquad \text{and} \qquad \tau = \frac{h}{2|c|}\left(\coth\alpha - \frac{1}{\alpha}\right).$$

The matrix $\mathbf{K}$ is positive definite, generally asymmetric, and diagonally semidominant for $0 \leq \alpha < \infty$.

We consider the weighted Jacobi smoother, whose iteration matrix is given by:

$$\mathbf{S}_\omega \;\equiv\; \mathbf{I} - \omega\mathbf{D}^{-1}\mathbf{K} \;=\; \mathbf{I} - \frac{\omega}{a+b}\,\mathbf{K} \;=\; \mathbf{I} - \omega\,\bar{\mathbf{K}} \tag{3.4}$$

where $\mathbf{D}$ is the diagonal of $\mathbf{K}$, $\mathbf{I}$ is the identity matrix, ω is the weighting factor, and $\bar{\mathbf{K}} = \frac{1}{a+b}\,\mathbf{K}$. Referring to equation (2.3), the weighted Jacobi method results in the following relaxation scheme:

$$\mathbf{u}^{i+1} = \left(\mathbf{I} - \frac{\omega}{a+b}\mathbf{K}\right)\,\mathbf{u}^i + \frac{\omega}{a+b}\mathbf{f} \tag{3.5}$$

so that $\mathbf{M}^{-1} = \frac{\omega}{a+b}\mathbf{I}$. This converges for $0 < \omega \leq 1$ given the above system (cf. Varga

[29] or Young [32]). (We will use $\omega = 2/3$ throughout, but we note that this is only considered optimal, for smoothing purposes, in the symmetric case where $\alpha = 0$.)

We defined the error for iteration number i to be $\mathbf{e}^i = \mathbf{u}^i - \mathbf{u}$ where $\mathbf{e}^i \in \mathbb{R}^N$, so that the weighted Jacobi iterations are equivalent to the following iterations on the error:

$$\mathbf{e}^{i+1} = \mathbf{S}_\omega \, \mathbf{e}^i \tag{3.6}$$

For this particular problem we know that the iterations (3.5) converge; what we want to know is which components of the error are reduced most effectively by the iteration matrix $\mathbf{S}_\omega$ of equation (3.4), as per equation (3.6).

Before proceeding we define the *amplification factor* in the algebraic context, $\mathcal{A}(\mathbf{A}, \mathbf{x})$, for a given matrix $\mathbf{A}$ operating on a vector:

$$\mathcal{A}(\mathbf{A}, \mathbf{x}) \equiv \frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|} = \left(\frac{\mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \right)^{1/2} \tag{3.7}$$

where the norm $\|y\| \equiv (y^T y)^{1/2}$ measures the "length" of the vector y. The amplification factor quantifies the relative change in length of a given vector in the direction of $\mathbf{x}$ under the action of $\mathbf{A}$. When $\mathcal{A}(\mathbf{A}, \mathbf{x}) > 1$ the vector is amplified; and, when $\mathcal{A}(\mathbf{A}, \mathbf{x}) < 1$ the vector is attenuated. The maximum of $\mathcal{A}(\mathbf{A}, \mathbf{x})$ over all $\mathbf{x}$ is the spectral norm of $\mathbf{A}$.

It is clear from equation (3.7) that the amplification factor is closely related to the *normal matrix* $\mathbf{A}^T\mathbf{A}$. In fact, if the eigenvalues of the normal matrix are arranged in ascending order, the associated eigenvectors point in the directions of increasing amplification with respect to the matrix $\mathbf{A}$. Furthermore, since the normal matrix is symmetric it possesses a complete set of orthogonal eigenvectors which form a basis for the space $\mathbb{R}^N$. As we shall see, this basis naturally separates the space into smooth and oscillatory subspaces.

For the symmetric case the eigenvectors of the system matrix $\mathbf{K}$ and the normal

matrix $\mathbf{K}^T\mathbf{K}$ coincide; but, in general the differential operator in equation (3.1) is not self-adjoint, and the resulting stiffness matrix $\mathbf{K}$ is asymmetric. To facilitate the analysis, we note that the matrix $\bar{\mathbf{K}}$ is equivalent to the linear translation, or differencing, operator:

$$\bar{\mathsf{L}} = \bar{\alpha} h D - \frac{h^2}{2} D^2 \tag{3.8}$$

where the central-differencing operators D and D^2 are given by:

$$Dv_j = \frac{v_{j+1} - v_{j-1}}{2h} \qquad \text{and} \qquad D^2 v_j = \frac{v_{j+1} - 2v_j + v_{j-1}}{h^2}. \tag{3.9}$$

In equation (3.8), the parameter $\bar{\alpha} = \tanh(\alpha)$ is an effective Peclet number for the stabilized finite element formulation, where $\bar{\alpha} = 0$ for the purely diffusive case ($c = 0, k \neq 0, \alpha = 0$), and $\bar{\alpha} = 1$ for the purely advective case ($k = 0, c \neq 0, \alpha \to \infty$). Stabilization is responsible for limiting the parameter $\bar{\alpha}$ between 0 and 1 in accordance with the problem's physical parameters and the element length (cf. Brooks & Hughes [9]).

We can express the nodal error in terms of a discrete Fourier vectors,

$$\mathbf{e}(j) = \Re e \left(\frac{1}{\sqrt{N}} \sum_{n=1}^{N} \beta_n \exp\left(-\imath \frac{2n\pi j}{N} \right) \right) = \Re e \left(\sum_{n=1}^{N} \beta_n \boldsymbol{\varphi}_n(j) \right) \tag{3.10}$$

where $\Re e$ indicates the real part of a complex vector, $\mathbf{e} = (\mathbf{e}(1), \mathbf{e}(2), \ldots, \mathbf{e}(N)) \in \mathbb{R}^N$ is the discrete error vector, the coefficients $\beta_n \in \mathbb{C}$ are complex constants, $\imath = \sqrt{-1}$, and the vectors

$$\boldsymbol{\varphi}_n(j) = \frac{1}{\sqrt{N}} \exp\left(\imath \frac{2n\pi j}{N} \right) \qquad \text{for} \quad j = 1, 2, 3, ..., N \tag{3.11}$$

are the discrete Fourier modes indexed by n.

We specify an inner product on the space $\mathbb{C}^N$,

$$\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{j=1}^{N} v^*(j)\, u(j) \qquad\qquad \mathbf{u}, \mathbf{v} \in \mathbb{C}^N \tag{3.12}$$

where the superscript "$*$" indicates the complex conjugate. This inner product induces the norm,

$$\|\mathbf{u}\| = \left(\langle \mathbf{u}, \mathbf{u} \rangle\right)^{1/2}. \tag{3.13}$$

We observe that the basis vectors $\boldsymbol{\varphi}_n(j)$ form an orthonormal set in that $\langle \boldsymbol{\varphi}_n, \boldsymbol{\varphi}_m \rangle = 0$ when $n \neq m$, and $\langle \boldsymbol{\varphi}_n, \boldsymbol{\varphi}_n \rangle = 1$.

The objective of the following analysis is to characterize the amplifying effect of $\mathbf{S}_\omega$ on arbitrary directional vectors. Since the differencing operator is linear we may consider its effect on each individual Fourier vector separately, and then combine these results by superposition to characterize the effect on arbitrary vectors. Applying the operator $\bar{\mathsf{L}}$ to the n^{th} Fourier mode we obtain:

$$\bar{\mathbf{K}}\boldsymbol{\varphi}_n = \bar{\mathsf{L}}(\boldsymbol{\varphi}_n) = \left[\left(1 - \cos\left(\frac{n\pi}{N}\right)\right) + \imath\bar{\alpha}\sin\left(\frac{n\pi}{N}\right)\right]\boldsymbol{\varphi}_n = z_n\boldsymbol{\varphi}_n \tag{3.14}$$

The amplification factor of $\bar{\mathbf{K}}$ operating on the n^{th} mode, $\boldsymbol{\varphi}_n$, is then given by the argument of the complex multiplier $z_n \in \mathbb{C}$:

$$\mathcal{A}(\bar{\mathbf{K}}, \boldsymbol{\varphi}_n) = \left(\left(1 - \cos\left(\frac{n\pi}{N}\right)\right)^2 + \bar{\alpha}^2 \sin^2\left(\frac{n\pi}{N}\right)\right)^{1/2} \tag{3.15}$$

And, the modulus of z_n is denoted by θ_n, and is given by

$$\cos(\theta_n) \;=\; \frac{\langle \boldsymbol{\varphi}_n, \bar{\mathbf{K}}\boldsymbol{\varphi}_n \rangle}{\|\bar{\mathbf{K}}\boldsymbol{\varphi}_n\|} \;=\; \frac{1 - \cos\left(\frac{n\pi}{N}\right)}{\mathcal{A}(\bar{\mathbf{K}}, \boldsymbol{\varphi}_n)}. \tag{3.16}$$

So, finally, we have $\bar{\mathbf{K}}\boldsymbol{\varphi}_n = \bar{\mathsf{L}}(\boldsymbol{\varphi}_n) = z_n\boldsymbol{\varphi}_n = \mathcal{A}_n \exp(\imath\theta_n)\boldsymbol{\varphi}_n$, where the complex con-

stant $z_n = \mathcal{A}_n \exp(\imath\theta_n)$ accounts for the rotation and amplification, $\mathcal{A}_n = \mathcal{A}(\bar{\mathbf{K}}, \boldsymbol{\varphi}_n)$, for a given Fourier mode. Here, $\mathcal{A}_n$ is the amplification in the Fourier context.

We consider the two limiting cases where $\alpha = 0$ and $\alpha \to \infty$. The purely diffusive case, where $\alpha = 0$, is straightforward and well documented:

$$\mathcal{A}(\bar{\mathbf{K}}, \boldsymbol{\varphi}_n)_{\alpha=0} = 1 - \cos\left(\frac{n\pi}{N}\right) = 2\sin\left(\frac{n\pi}{2N}\right) \tag{3.17}$$

$$\cos(\theta_n)_{\alpha=0} = 1 \quad \Longrightarrow \quad \theta_n = 0 \tag{3.18}$$

where the modulus is always and, therefore, the real parts of the discrete Fourier vectors correspond to the eigenvectors of $\bar{\mathbf{K}}$.

For the purely advective case, $\alpha \to \infty$, we have:

$$\mathcal{A}(\bar{\mathbf{K}}, \boldsymbol{\varphi}_n)_{\alpha\to\infty} = \sqrt{2}\left(1 - \cos\left(\frac{n\pi}{N}\right)\right)^{1/2} \tag{3.19}$$

$$\cos(\theta_n)_{\alpha\to\infty} = \frac{1}{\sqrt{2}}\left(1 - \cos\left(\frac{n\pi}{N}\right)\right)^{1/2} = \frac{1}{2}\mathcal{A}(\bar{\mathbf{K}}, \boldsymbol{\varphi}_n)_{\alpha\to\infty} \tag{3.20}$$

For this case it remains to be shown how the modulus is related to the amplification factor for an arbitrary vector, which will allow us to relate $\|\mathbf{S}_\omega \boldsymbol{\psi}\|$ directly to $\|\mathbf{K}\boldsymbol{\psi}\|$ for any $\boldsymbol{\psi}$, and in particular for the eigenvectors $\boldsymbol{\psi}_k$ of the normal matrix $\mathbf{K}^T\mathbf{K}$.

In the algebraic context, the amplification factor of $\bar{\mathbf{K}}$ for an arbitrary unit vector $\boldsymbol{\psi}$, where $\|\boldsymbol{\psi}\| = 1$, is defined by equation (3.7) to be:

$$\mathcal{A}(\bar{\mathbf{K}}, \boldsymbol{\psi}) = \frac{\|\bar{\mathbf{K}}\boldsymbol{\psi}\|}{\|\boldsymbol{\psi}\|} = \|\bar{\mathbf{K}}\boldsymbol{\psi}\| \tag{3.21}$$

The real unit vector $\boldsymbol{\psi}$ can be represented as a linear combination of the Fourier

vectors:

$$\boldsymbol{\psi} = \sum_{n=1}^{N} \Re e\left(\mathsf{p}_n \boldsymbol{\varphi_n}\right) \tag{3.22}$$

where $\mathsf{p}_n \in \mathbb{C}$ are the complex coefficients. We can use the following relation for a real matrix $\bar{\mathbf{K}}$ and the real unit vector $\boldsymbol{\psi}$:

$$\boldsymbol{\psi}^t \bar{\mathbf{K}} \boldsymbol{\psi} = \langle \bar{\mathbf{K}} \boldsymbol{\psi}, \boldsymbol{\psi} \rangle = \|\bar{\mathbf{K}} \boldsymbol{\psi}\| \cos(\theta). \tag{3.23}$$

where θ is the acute angle formed by the vectors $\boldsymbol{\psi}$ and $\bar{\mathbf{K}}\boldsymbol{\psi}$. Rearranging we obtain:

$$\cos(\theta) = \frac{\boldsymbol{\psi}^t \bar{\mathbf{K}} \boldsymbol{\psi}}{\|\bar{\mathbf{K}} \boldsymbol{\psi}\|}. \tag{3.24}$$

Now an expression for $\cos(\theta)$ exclusively in terms of $\|\bar{\mathbf{K}}\boldsymbol{\psi}\|$ is needed.

Evaluating the numerator in the previous result, using (3.20), we have:

$$\boldsymbol{\psi}^t \bar{\mathbf{K}} \boldsymbol{\psi} = \frac{1}{2} \sum_{n=1}^{N} p_n^2 \mathcal{A}_n^2. \tag{3.25}$$

where $\mathcal{A}_n = \mathcal{A}(\bar{\mathbf{K}}, \boldsymbol{\varphi_n})$. We also find

$$\|\bar{\mathbf{K}} \boldsymbol{\psi}\| = \left(\boldsymbol{\psi}^t \bar{\mathbf{K}}^t \bar{\mathbf{K}} \boldsymbol{\psi}\right)^{1/2} = \left(\sum_{n=1}^{N} p_n^2 \mathcal{A}_n^2 \right)^{1/2}. \tag{3.26}$$

Finally, substituting equations (3.25) and (3.26) into (3.24), the general result for the case where $\alpha \to \infty$ is obtained:

$$\cos(\theta)_{\alpha \to \infty} = \frac{1}{2} \mathcal{A}(\bar{\mathbf{K}}, \boldsymbol{\psi}) = \frac{1}{2} \|\bar{\mathbf{K}} \boldsymbol{\psi}\| \tag{3.27}$$

This gives us an expression for $\cos(\theta)$, in equation (3.20), in terms of $\|\bar{\mathbf{K}}\boldsymbol{\psi}\|$ only, so that the the amplification factor of the matrix $\mathbf{S}_\omega$ may be related directly to $\|\mathbf{K}\boldsymbol{\psi}\|$

for any unit vector when $\alpha \to \infty$.

The amplification factor of $\bar{\mathbf{K}}$ has stationary points in the direction of the eigenvectors of the normal matrix, $\bar{\mathbf{K}}^t\bar{\mathbf{K}}$, which we denote by $\boldsymbol{\psi}_k$. The amplification factors in these directions are the eigenvalues σ_k of $\bar{\mathbf{K}}^t\bar{\mathbf{K}}$. The normal matrix is symmetric and we are, therefore, guaranteed a full set of orthogonal eigenvectors, so that the normalized system can always be diagonalized,

$$\bar{\mathbf{K}}^t\bar{\mathbf{K}} = \Psi\Sigma\Psi^t \tag{3.28}$$

where the eigenvector matrix is $\Psi = (\boldsymbol{\psi}_1\ \boldsymbol{\psi}_2 \cdots \boldsymbol{\psi}_N)$, when the eigenvalue matrix $\Sigma = diag(\sigma_1\ \sigma_2 \cdots \sigma_N)$ is arranged so that $\sigma_1 \leq \sigma_2 \leq \cdots \leq \sigma_N$.

Using equation (3.4), again assuming normalized eigenvectors $\|\boldsymbol{\psi}_k\| = 1$, we can calculate the amplification factors of $\mathbf{S}_\omega$ relative to those of $\bar{\mathbf{K}}$ for the eigenvectors $\boldsymbol{\psi}_k$ for $\alpha \to \infty$:

$$\begin{aligned} \|\mathbf{S}_\omega\boldsymbol{\psi}_k\|^2 &= 1 + \omega^2\|\bar{\mathbf{K}}\boldsymbol{\psi}_k\|^2 - \omega(\boldsymbol{\psi}_k^T(\bar{\mathbf{K}} + \bar{\mathbf{K}}^T)\boldsymbol{\psi}_k) \\ &= 1 + \omega^2\|\bar{\mathbf{K}}\boldsymbol{\psi}_k\|^2 - 2\omega\|\bar{\mathbf{K}}\boldsymbol{\psi}_k\|\cos(\theta) \end{aligned} \tag{3.29}$$

We can now summarize the results for the two limiting cases. For $\alpha = 0$ we have the relation:

$$\|\mathbf{S}_\omega\boldsymbol{\psi}_k\|^2 = 1 + \left(\frac{\omega}{a+b}\right)^2\|\mathbf{K}\boldsymbol{\psi}_k\|^2 - 2\left(\frac{\omega}{a+b}\right)\|\mathbf{K}\boldsymbol{\psi}_k\| \tag{3.30}$$

And, for $\alpha \to \infty$, using equation (3.27), we have:

$$\|\mathbf{S}_\omega\boldsymbol{\psi}_k\|^2 = 1 + \left(\frac{\omega}{a+b}\right)^2\|\mathbf{K}\boldsymbol{\psi}_k\|^2 - \left(\frac{\omega}{a+b}\right)\|\mathbf{K}\boldsymbol{\psi}_k\|^2 \tag{3.31}$$

These provide the lower and upper bounds, respectively, for the amplification factor of $\mathbf{S}_\omega$ for any given $0 \leq \alpha < \infty$ (or, $0 \leq \bar{\alpha} < 1$). Figure 4 relates these bounds to both the amplification factor of $\bar{\mathbf{K}}$, and the indices k of the eigenvectors $\boldsymbol{\psi}_k$ of $\bar{\mathbf{K}}^T\bar{\mathbf{K}}$, such

that the eigenvalues σ_k are arranged in increasing order ($\omega = 2/3$).

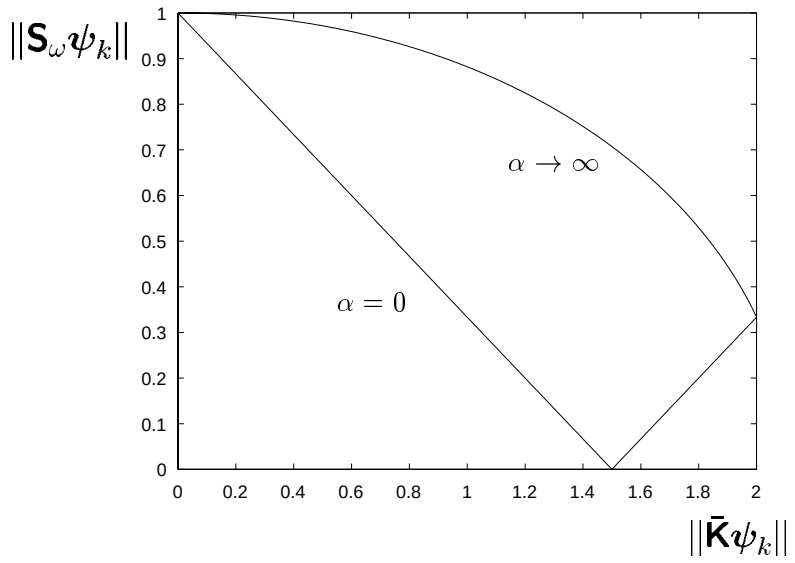
Since the eigenvectors $\boldsymbol{\psi}_k$ span the space $\mathbb{R}^N$, we can specify the error vector as a linear combination of these eigenvectors,

$$\mathbf{e} = \sum_{k=1}^{N} \gamma_k \boldsymbol{\psi}_k, \qquad \text{where} \quad \gamma_k \in \mathbb{R}, \tag{3.32}$$

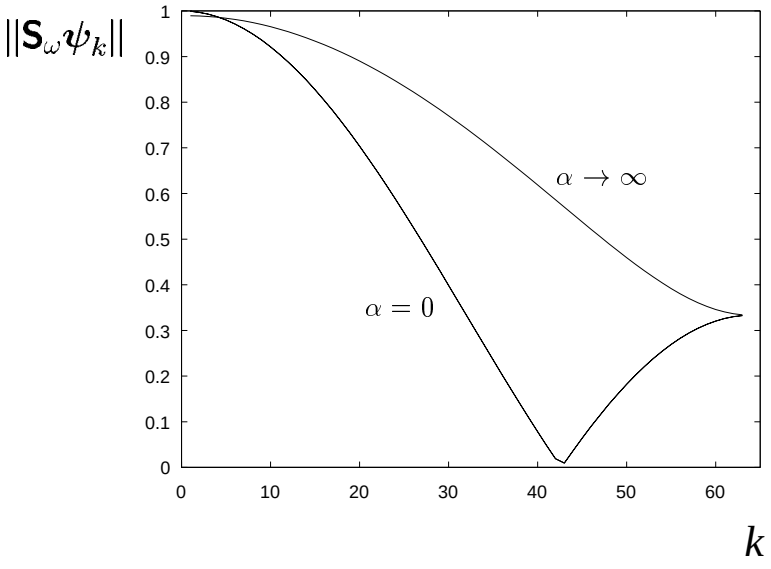
and we refer to the different terms in the sum as the "components" for the given basis. Figure 5 shows how the amplification factor of $\mathbf{S}_\omega$ varies with element Peclet number. The curves corresponding to $\alpha = 0$ ($\bar{\alpha} = 0$) and $\alpha \to \infty$ ($\bar{\alpha} = 1$) were calculated according to the equations (3.30) and (3.31), while the intermediate curves were calculated numerically. The smoother reduces most effectively those components of the error in the directions of the eigenvectors of the normal matrix associated with its largest eigenvalues. By definition, the span of these eigenvectors becomes the oscillatory space: $\mathbb{B} = \text{span}\{\boldsymbol{\psi}_k\}$, for $k = M \ldots N$.

We can see this a little more clearly if we consider a couple of examples where the initial error is the sum of the eigenvectors for the associated normal system, $\mathbf{e}^0 = \sum_{k=1}^{N} \psi_k$. First, we consider the symmetric case where $\alpha = 0$. Figure 6a depicts the number of iterations required to reduce the various components of the error by a factor of 100. The components are in the directions of the eigenvectors ψ_k of $\mathbf{K}^t\mathbf{K}$, where the associated eigenvalues have been arranged in increasing order. We observe that the components associated with the largest eigenvalues require far fewer iterations to be reduced than the components in the balance of the space. Further, if we look at the projection of the error after 10 smoothing iterations onto the eigenvectors of the normal matrix, in Figure 6b, we see that the components of the error associated with the largest eigenvalues have been virtually eliminated while those associated with the smaller eigenvalues persist.

We also consider the asymmetric case where $\alpha \to \infty$. The graphs in Figure 7 mirror those in Figure 6, and again we see in Figure 7a that the smoother requires



(a)



(b)

Figure 4: Amplification factors of $S_{\omega=2/3}$ for the limiting cases vs.: (a) amplification factor of $\bar{K}$; and, (b) the indices k of the eigenvalues of $K^T K$ arranged in increasing order. (N=64)

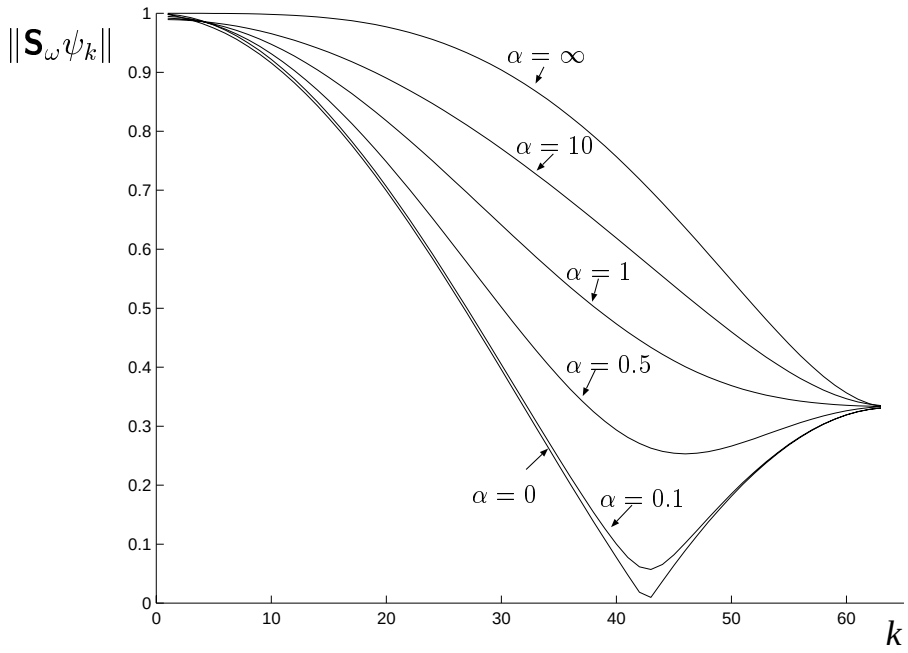
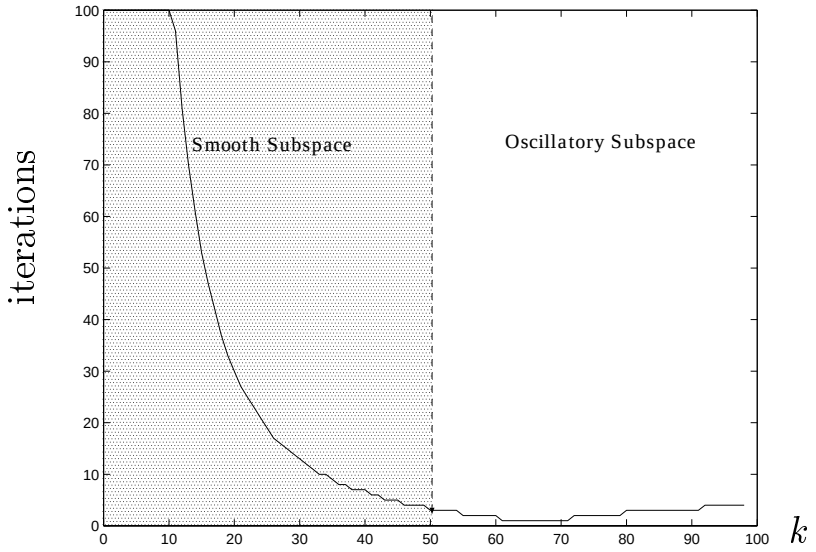


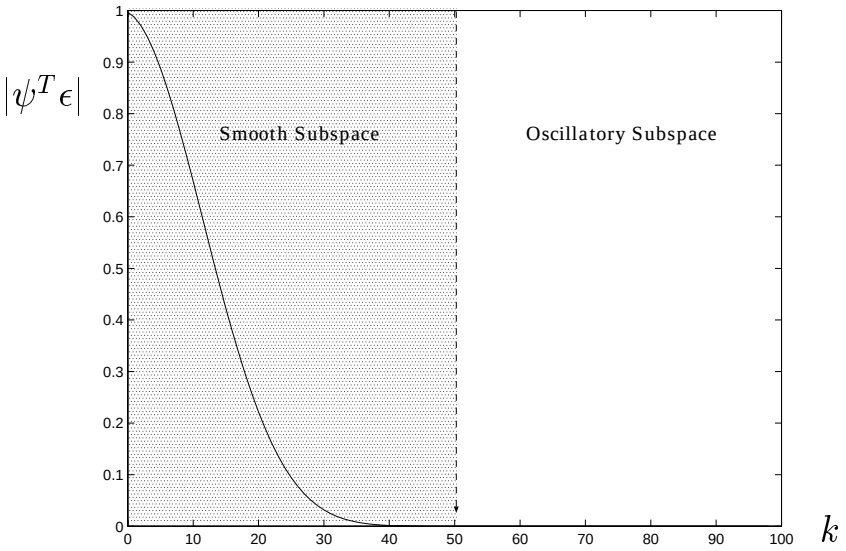
Figure 5: Amplification Factors of $\mathbf{S}_{\omega=2/3}$ as a Function of the Indices k of the Eigenvalues of $\mathbf{K}^t\mathbf{K}$ Arranged in Increasing Order. (N=64)

relatively few iterations to reduce the error in the directions of the eigenvectors of the normal matrix associated with its largest eigenvalues. Referring to Figure 7b, we find, once again, that after 10 smoothing iterations the error has been drastically reduced in the upper part of the spectrum, while in the lower part of the spectrum the error is still quite large. (Note that the precise division between the smooth and oscillatory subspaces is somewhat arbitrary.)

For positive definite systems the connection with the normal matrix generalizes, and the upshot of all this is that the smoother naturally divides the space in terms of the eigenvectors of the normal matrix. The smoother eliminates the components in the directions of the eigenvectors associated with the largest eigenvalues. This suggests that the ideal basis for the smooth, or coarse, subspace would be the eigenvectors of the normal matrix associated with its smallest eigenvalues. If one could successfully and inexpensively approximate these eigenvectors—say: $\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_{N/3}$—one could then construct a suitable prolongation operator to complement the smoother.

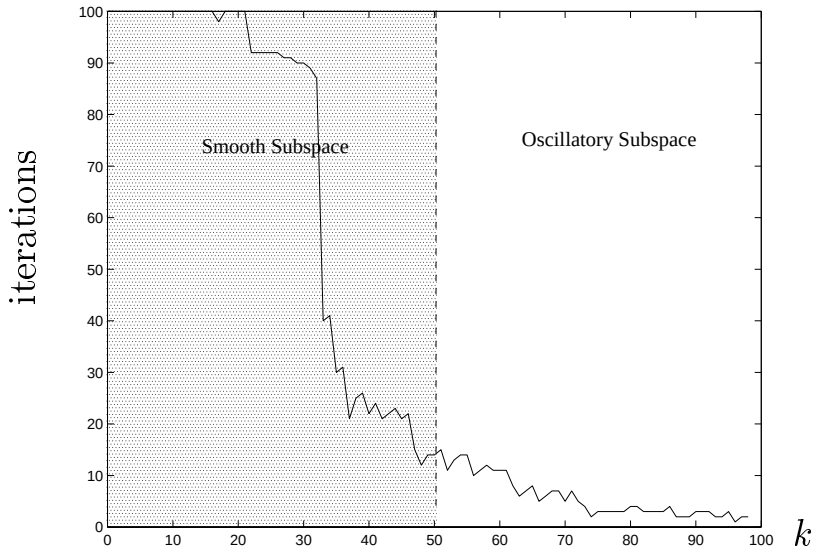


(a) number of iterations required by the weighted Jacobi smoother to reduce various components of the error by a factor of 100.

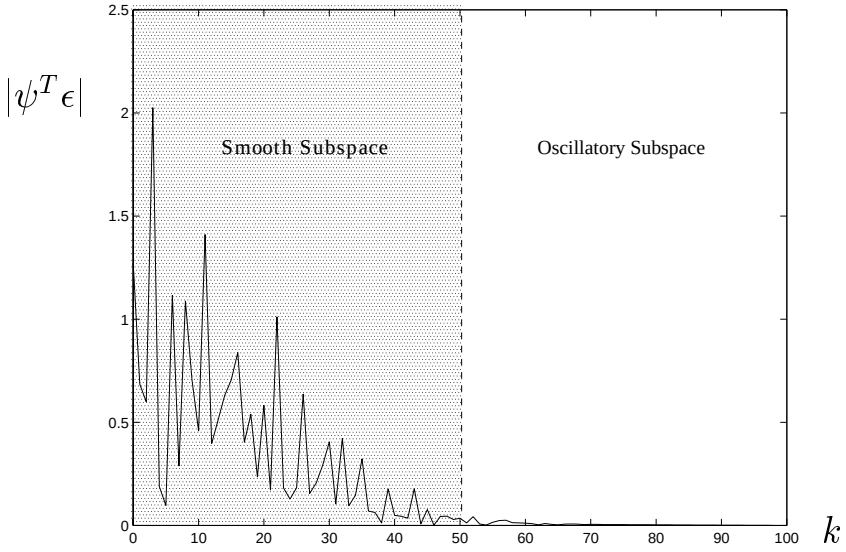


(b) projection of the error onto the eigenvectors of the normal matrix after 10 iterations of the weighted Jacobi smoother.

Figure 6: Symmetric Case ($\alpha = 0$, N=100)



(a) number of iterations required by the weighted Jacobi smoother to reduce various components of the error by a factor of 100.



(b) projection of the error onto the eigenvectors of the normal matrix after 10 iterations of the weighted Jacobi smoother.

Figure 7: Asymmetric Case ($\alpha \to \infty$, N=100)

4 Restriction/Prolongation Operator

According to the preceding analysis, the ideal subspace on which to perform the coarse-level correction of equation (2.11) is the space spanned by the eigenvectors of the normal matrix associated with its smallest eigenvalues. For the two-level scheme these eigenvectors can be approximated effectively, and relatively inexpensively, using an aggregation technique, the basic idea of which has been used in the multi-grid framework by Bulgakov [11, 10], Bulgakov & Kuhn [12], Vaněk, *et al.* [27, 28], and Fish $ Belsky [19].

An aggregate is simply a collection of adjacent, interior elements as depicted in Figure 8. The aggregates are separated by buffer elements so that for linear interpolating elements, each interior nodal degree of freedom is contained in only one aggregate. The aggregation procedure is very similar to that outlined in [19], so we refer to this paper for the details. The main difference here is that the degrees of freedom attached to the boundaries are isolated so that the essential boundary conditions are satisfied exactly on all levels; and, the restriction/prolongation operator is based on the eigenspace of the normal system.

Element contributions from the buffer elements are assembled into the global matrix in the traditional way. For the aggregates, however, the element contributions, $\mathbf{k_{ae}}$, are first assembled into an *aggregate submatrix*, $\mathbf{K}_a$,

$$\mathbf{K}_a = \mathop{\mathbf{A}}_{ae=1}^{N_{ae}} \mathbf{k_{ae}}. \tag{4.1}$$

From this we can construct the *normal aggregate submatrix*, $\mathbf{K}_a^T\mathbf{K}_a$, which is symmetric and, therefore, can be diagonalized,

$$\mathbf{K}_a^T\mathbf{K}_a = \Psi_a \Sigma_a \Psi_a^T \tag{4.2}$$

where the eigenvector matrix is $\Psi_a = (\boldsymbol{\psi}_{a_1}\ \boldsymbol{\psi}_{a_2} \cdots \boldsymbol{\psi}_{a_{N-1}})$, and the eigenvalue matrix

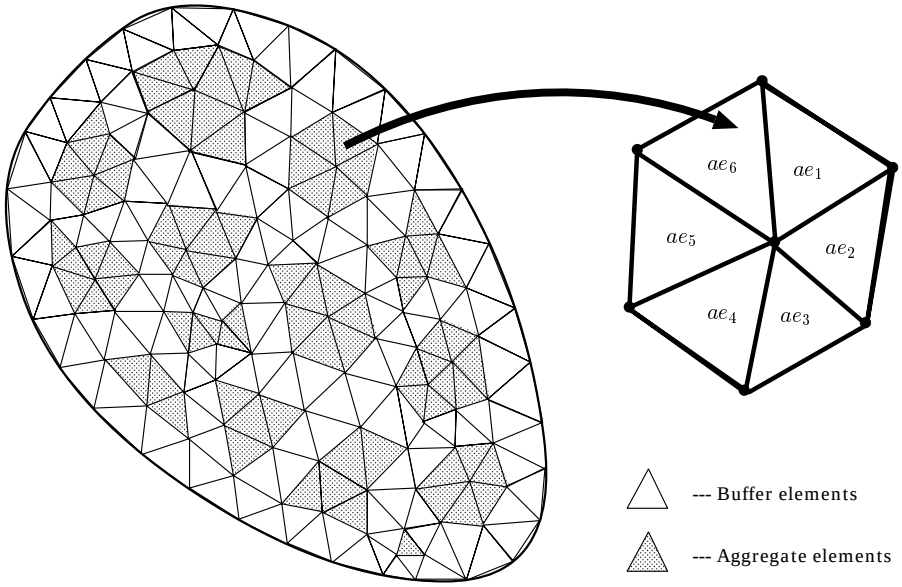


Figure 8: Aggregate

$\Sigma = diag(\sigma_{a_1}\ \sigma_{a_2} \cdots \sigma_{a_{N-1}})$ is arranged so that the eigenvalues are in ascending order $\sigma_{a_1} \leq \sigma_{a_2} \leq \cdots \leq \sigma_{a_{N-1}}$. We use the eigenvectors of the normal aggregate submatrix to approximate the eigenvectors of the normal matrix. For each aggregate, we assemble only those eigenvectors of $\mathbf{K}_a^T\mathbf{K}_a$ associated with its smallest eigenvalues into the columns of the prolongation operator $\mathbf{Q}$,

$$\mathbf{Q} = \mathop{\mathbf{A}}_{ae=1}^{m<N_{ae}} \boldsymbol{\psi}_{ae}. \tag{4.3}$$

In this way we constrain the degrees of freedom on each aggregate so that they vary only as a linear combination of this subset of vectors $\{\boldsymbol{\psi}_{a_1}, \cdots, \boldsymbol{\psi}_{a_m}\}$.

The scheme is adaptive in the sense that the dimension of the smooth, or coarse, subspace can be adjusted to better complement the smoother by taking more or fewer eigenvectors. The importance of adjustability lies in the observation that while the smoother deals with the oscillatory end of the spectrum and the coarse-level correction deals with the smooth end of the spectrum, these subspaces must meet, or overlap, somewhere in the middle. This can be accomplished by enhancing the smoother

(e.g., adding more fill levels to an ILU(n) preconditioner) or by enlarging the coarse subspace. We have allowed both possibilities.

It should be noted that using the connectivities of the mesh vertices is not suited to higher-order basis functions, since it assumes that the mesh edges constitute a graph for the unknowns; and, this only holds when the discrete unknowns are associated with nodes attached to the mesh vertices. Nor is this technique directly suited to aggregation beyond two levels, which is required for a true multilevel scheme. To consider these cases we can use the non-zero pattern of the sparse matrices to construct a graph of the unknown degrees of freedom and collect these into aggregates based on this graph.

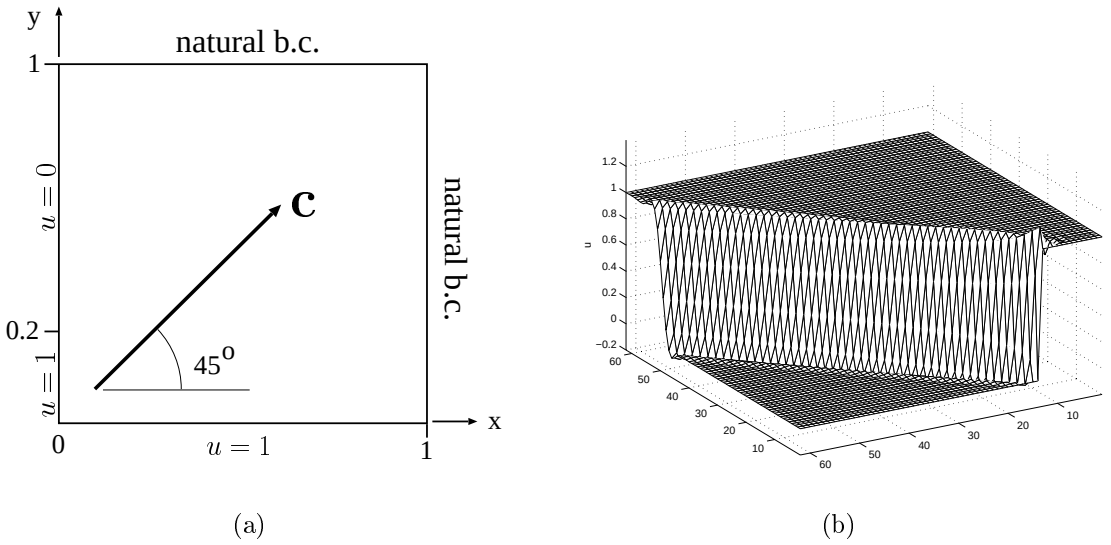
5 Numerical Examples

Ultimately, we want to use these multilevel cycles as a preconditioner, and solve the preconditioned system with a projection-type iterative solver, or accelerator, such as GMRES or TFQMR. To simplify notation we will denote the two-level preconditioner in the following way:

$$GAM\{\nu_1, \nu_2, n, Sm\} \tag{5.1}$$

where ν_1 is the number of presmoothing iterations, ν_2 is the number of postsmoothing iterations, n is the number of cycles ($n = 1$ is the V-cycle, and $n = 2$ the W-cycle), and Sm is the smoother used (e.g., $ILU(0)$). GAM stands for *Generalized Aggregation Multilevel* solver [19], and the approach described in the current paper extends this method from symmetric to asymmetric, positive definite systems. The solvers were implemented using the PETSc software developed at Argonne National Laboratories, which is described in references [3], [4], and [5].

The objective behind the multilevel method is to scale the computational work linearly with the number of unknowns in the algebraic system, N. If the number of



(a) (b)

Figure 9: (a) 2D Problem Specification and (b) Exact Finite Element Solution for $\alpha \to \infty$

iterations required for convergence is independent of the problem size, and if each iteration requires $O(N)$ work, then the computational work required to solve the system scales linearly with the problem size, N. With the two-level scheme we can test the first requirement; but, unfortunately, we cannot test the second because the work required by the exact solve at the coarse level grows faster than N, and will eventually dominate the problem. The cure for this is to allow for multiple levels so that the coarsest-level problem can be made relatively small and inexpensive.

Since overall timings are not really relevant for two-level methods (the coarse-level problem being too expensive) we will not present comparisons for examples 1 and 3. We will, however, address the issue of scalability in example 2. Here we examine the key components of the two-level solver to verify that the CPU time scales linearly with problem size, noting that the exact solve at the coarse level is inordinately expensive and should be dealt with recursively using a multilevel scheme.

5.1 Example 1—2D Advection-Diffusion

We first consider the two-dimensional, scalar advection-diffusion problem:

$$(\mathbf{c} \cdot \nabla)u - \kappa \Delta u = 0 \tag{5.2}$$

where $\mathbf{c}$ is the advective velocity, κ is the coefficient of diffusion, and Δ is the Laplacian operator. A schematic of the problem is given in Figure 9. For this example, we have used a GMRES accelerator, but results for the TFQMR accelerator were similar.

In Figure 10 we compare the performance of the $ILU(1)$ preconditioner with the $GAM\{1, 1, 1, ILU(1)\}$ preconditioner in terms of iteration count for the case as $\alpha \to \infty$. The comparison is made on the basis of comparable reductions in the absolute residual (as opposed to the preconditioned residual). In terms of iteration count, the two-level preconditioner significantly outperforms the ILU(1) preconditioner, but what is most notable is that the iteration count for the two-level method is essentially constant over a wide range of problem sizes.

We also consider the effects of varying the element Peclet number α. Here, a $GAM\{2, 2, 1, ILU(0)\}$ preconditioner was used, and the reduction of the normalized (absolute) residual is shown in Figure 11 for various Peclet numbers. We find that for all cases, convergence requires very few iterations, with the diffusion dominated cases converging in fewer iterations than the advection dominated cases. The natural boundary conditions along the outflow boundaries made the problem more sensitive, but did not upset the convergence when using the two-level preconditioner.

5.2 Example 2—2D Supercritical Airfoil

The second example is of a supercritical transonic airfoil. We solved the nonlinear Transonic Small-Disturbance (TSD) system of equations, which is approximate for thin bodies in inviscid fluids. The stabilized finite element formulation is presented in [20] and the pressure field is shown in Figure 12. We can see from the figure

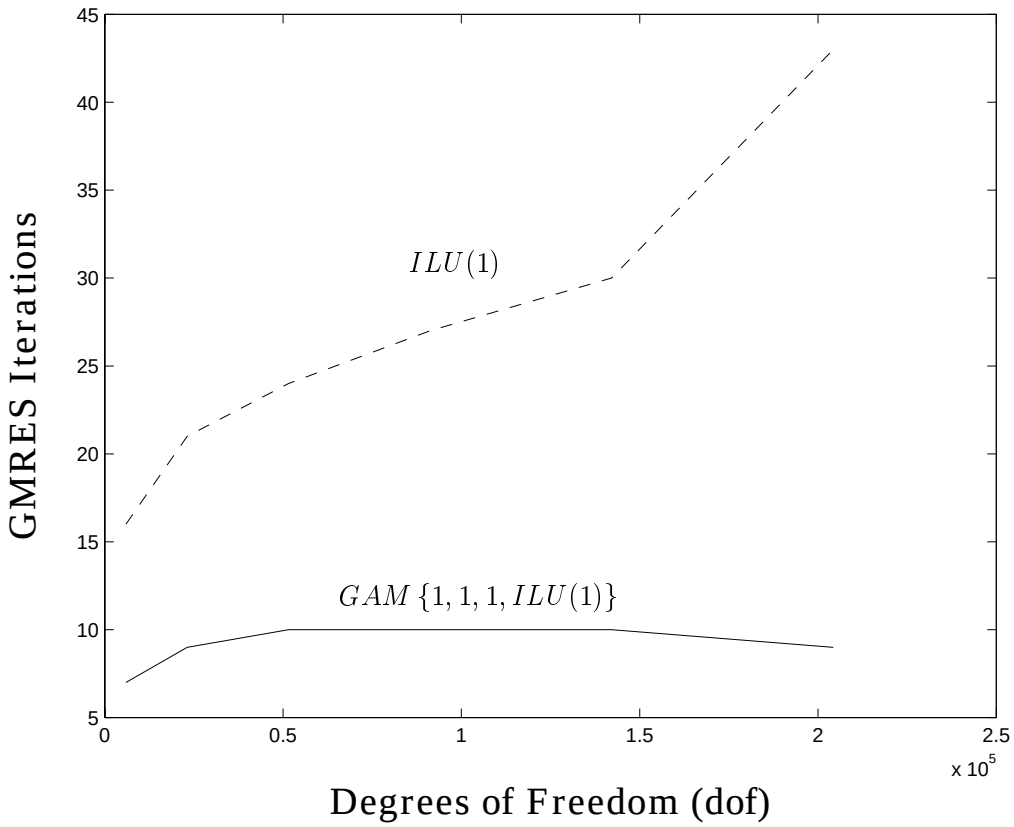


Figure 10: Iteration Count vs. Problem Size for $ILU(1)$ and $GAM\{1,1,1,ILU(1)\}$ Preconditioners with GMRES Accelerator for $\alpha \to \infty$.

that the flow accelerates to supersonic speeds over the airfoil section, then transitions through a shock wave attached to the airfoil surface, and leaves the trailing edge at subsonic speeds. The equations are nonlinear, so the iteration counts represent the average number of iterations per Newton step. (The standard deviation from the mean was very small for the two-level preconditioner, but relatively large for the ILU preconditioner.)

The convergence rates for the $ILU(0)$ and $GAM\{2,2,1,ILU(0)\}$ were compared for several irregular, triangular meshes with uniform element size. Here, the convergence was measured with respect to relative, preconditioned residuals. The convergence criteria was a reduction by six orders of magnitude. Again, referring to Figure 13, the iteration count for the two-level $GAM\{2,2,1,ILU(0)\}$ preconditioned

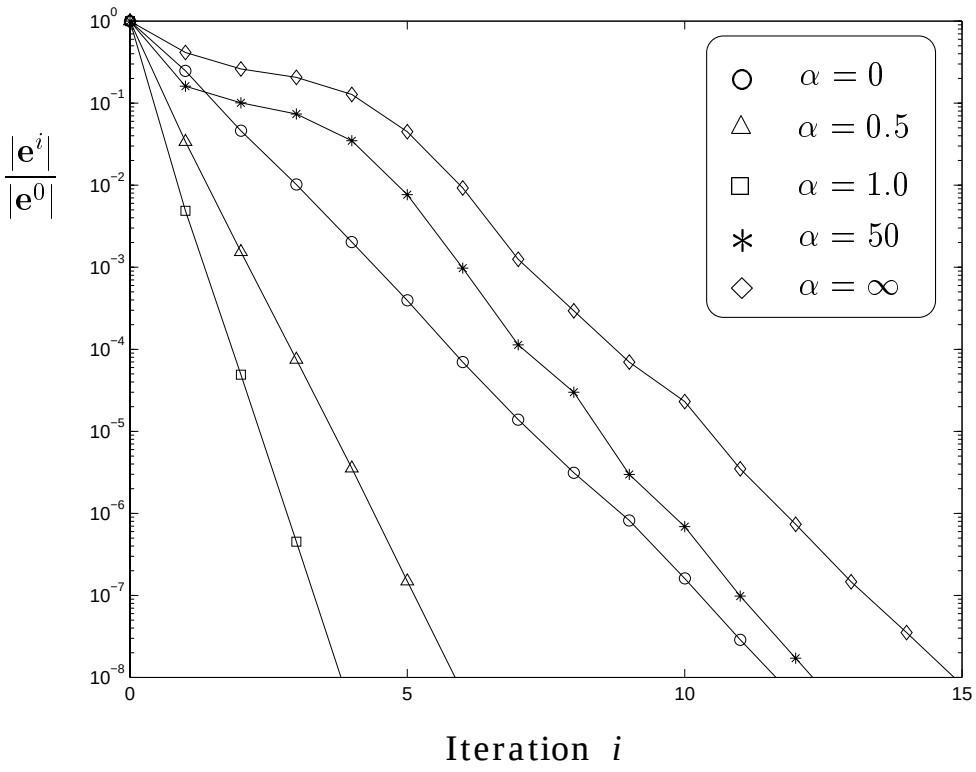


Figure 11: Effect of Peclet Number α on Convergence Rate. Relative Error vs. Iteration Count

GMRES solver remains almost constant over a fairly large range of problem sizes, whereas the number of iterations required by the $ILU(0)$ preconditioned GMRES solver grew at a faster rate. Results were similar for cases involving local mesh refinement near the airfoil section.

For this example we will take up the question of timings. While the exact solve at the coarse level for a two-level method is expected to be too computationally expensive to render such a scheme practical. The other components of the scheme should remain relatively inexpensive, however, since these do not change as we proceed to multiple levels. Figure 14 shows the growth rate of the critical components of the two-level method. The growth rate appears to be linear for all of these constituent parts. The figure also shows that the aggregation-based eigenvector approximation is relatively inexpensive; and, that the aggregation process and the formation of the coarse-level

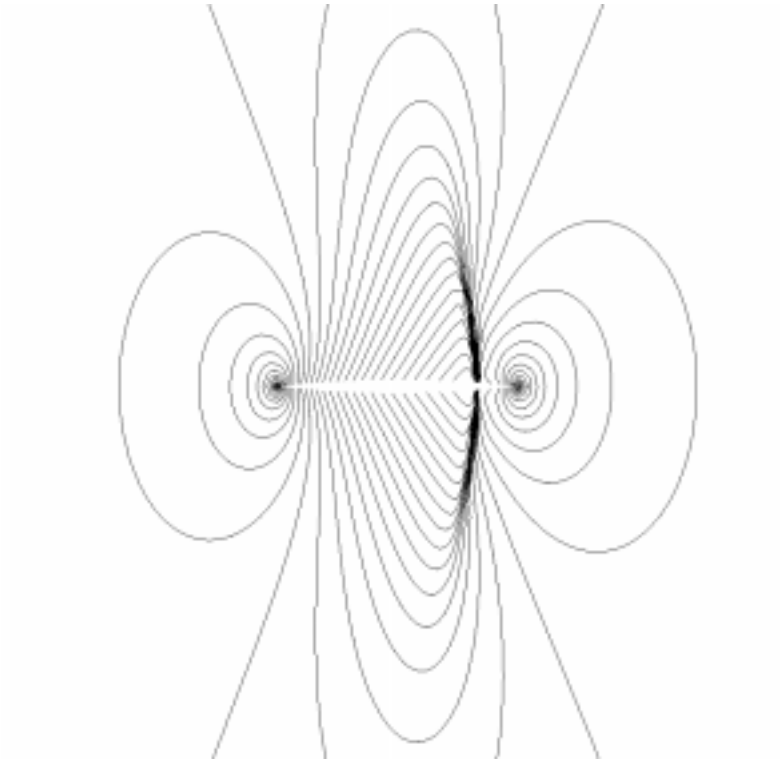


Figure 12: Pressure Contours for Subsonic, Supercritical Flow over a Circular-arc Airfoil

matrix $\mathbf{K}_0$ are the dominant components in terms of cost, although they too are fairly inexpensive. The sparse matrix multiplications ("amub" routine from SPARSPAK) used to calculate $\mathbf{K}_0$ are also shown.

5.3 Example 3—3D Transonic Nozzle

Finally, we consider the solution for a 3D, transonic (slightly subsonic) flow through a converging-diverging nozzle. Linear, tetrahedral elements were used and a representative solution is given in Figure 15. The same convergence criteria as with the preceding problem was used.

Figure 16 shows the comparison between the iteration counts for the $ILU(1)$ preconditioner and the $GAM\{2, 2, 2, ILU(1)\}$ preconditioner. Once again, we find that the iteration count for the two-level GAM preconditioner is nearly constant,

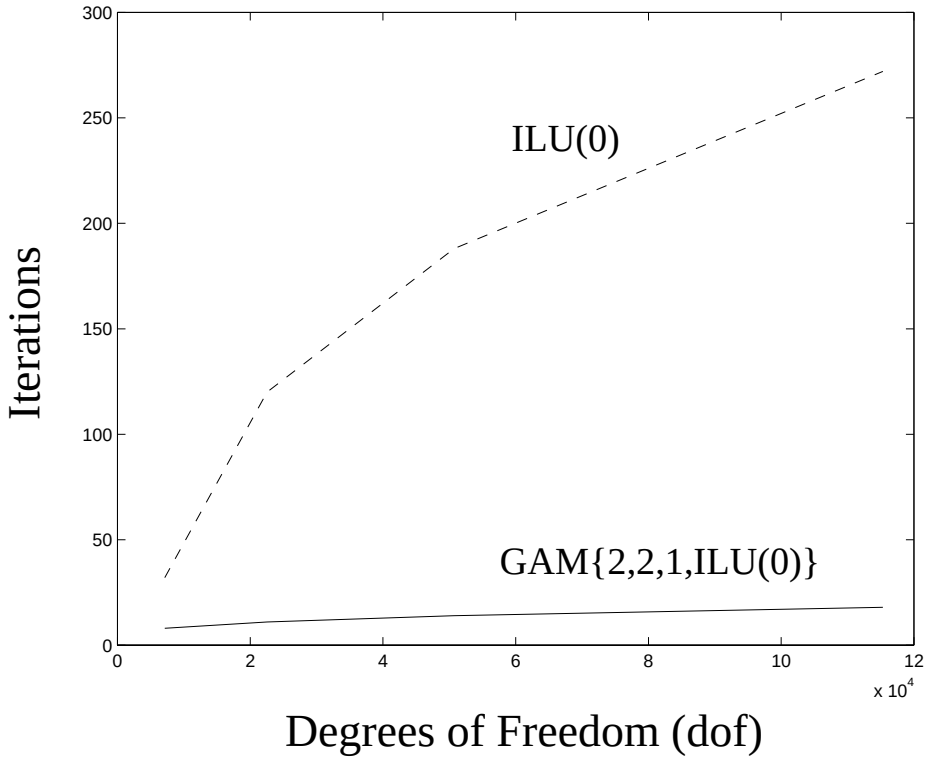


Figure 13: Iteration Count vs. Problem Size for $ILU(0)$ and $GAM\{2,2,1,ILU(0)\}$ Preconditioners with GMRES Accelerator.

while that of the $ILU(1)$ preconditioner grows quite rapidly. This demonstrates that the smooth and oscillatory subspaces complement each other well, but it should be noted that this complementarity requires some adjustment; specifically, the number of smoothing iterations, the type of smoother, and the number of two-level cycles were all varied to achieve good convergence behavior. The coarse-level reduction factor, which also could have been easily adjusted, was left fixed at around 2.5-3 for this problem.

6 Conclusions

Using a one-dimensional model problem, we have shown analytically that a representative smoother reduces most effectively those components of the error in the

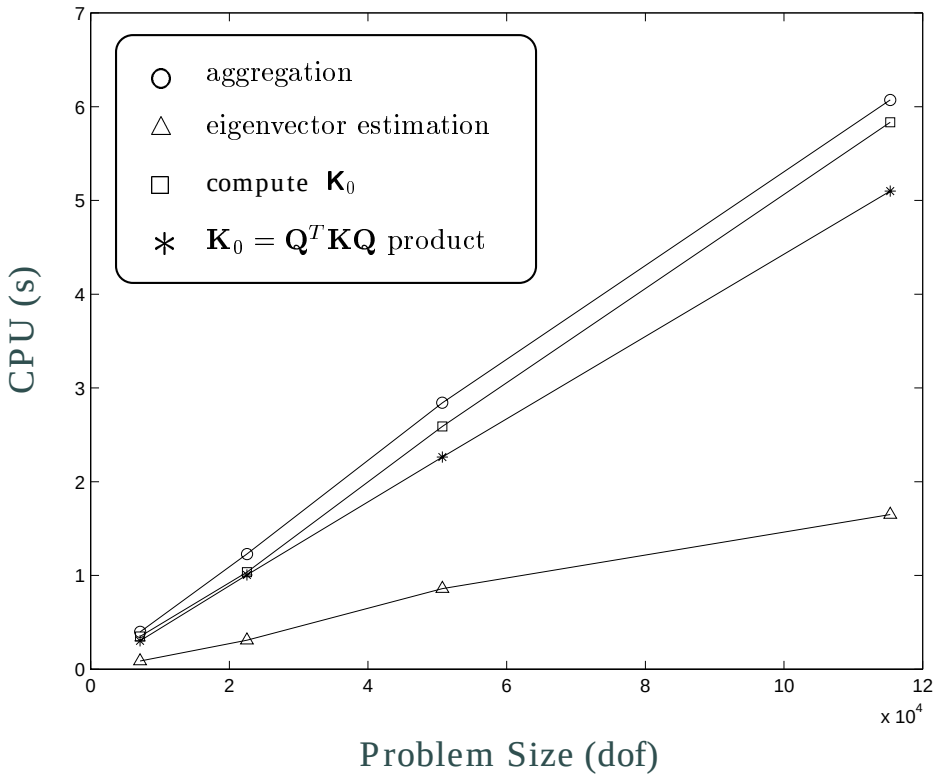


Figure 14: CPU Time vs. Problem Size (degrees of freedom) for the Critical Components of the Two-level Method.

directions of the eigenvectors of the normal matrix associated with its largest eigenvalues. This behavior suggests that the overall space can be divided, for multilevel purposes, according to the eigenvectors of the normal system. Since the normal system is symmetric, its eigenvectors constitute a complete, orthogonal basis.

An aggregation technique was then used to form the prolongation/restriction operator, but with some modifications this basic approach can be made purely algebraic. Since the approach is algebraically based, results generalize naturally to problems with higher dimensions and/or unstructured meshes. The results may hold for general asymmetric, positive definite systems, but tests were conducted only on systems which emanated from stabilized finite element formulations.

Several example problems demonstrate that the basic two-level scheme, constructed based on the analysis, serves well as a preconditioner when combined with

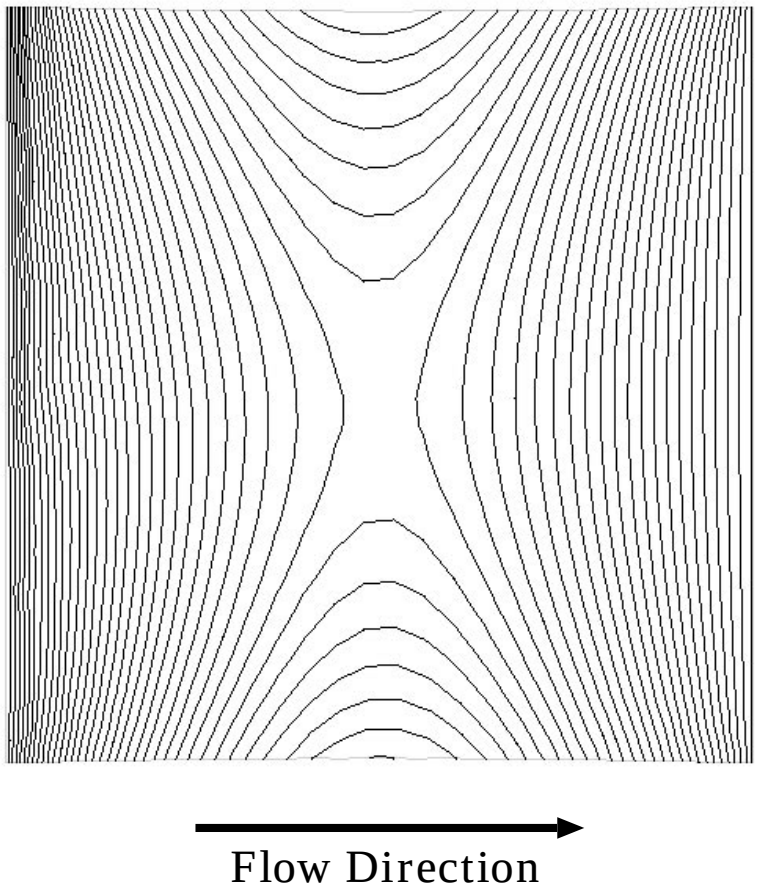


Figure 15: Pressure Contours for Transonic Nozzle Flow

either a GMRES of TFQMR accelerator. The iteration counts remain nearly constant over a fairly wide range of problem sizes and the solvers appear to be very robust, although, the stabilized finite element procedure generally does not yield pathologically ill-conditioned systems.

The next step is to make the scheme recursive so that we obtain a true multilevel method. Section 2 provides the basic framework, and based on the timings for the constituent parts of the solver (i.e. the solution of the aggregate eigenvalue problems, the formation of the coarse-level matrices, etc.), all components are relatively inexpensive to compute and possess linear growth rates with respect to the problem size.

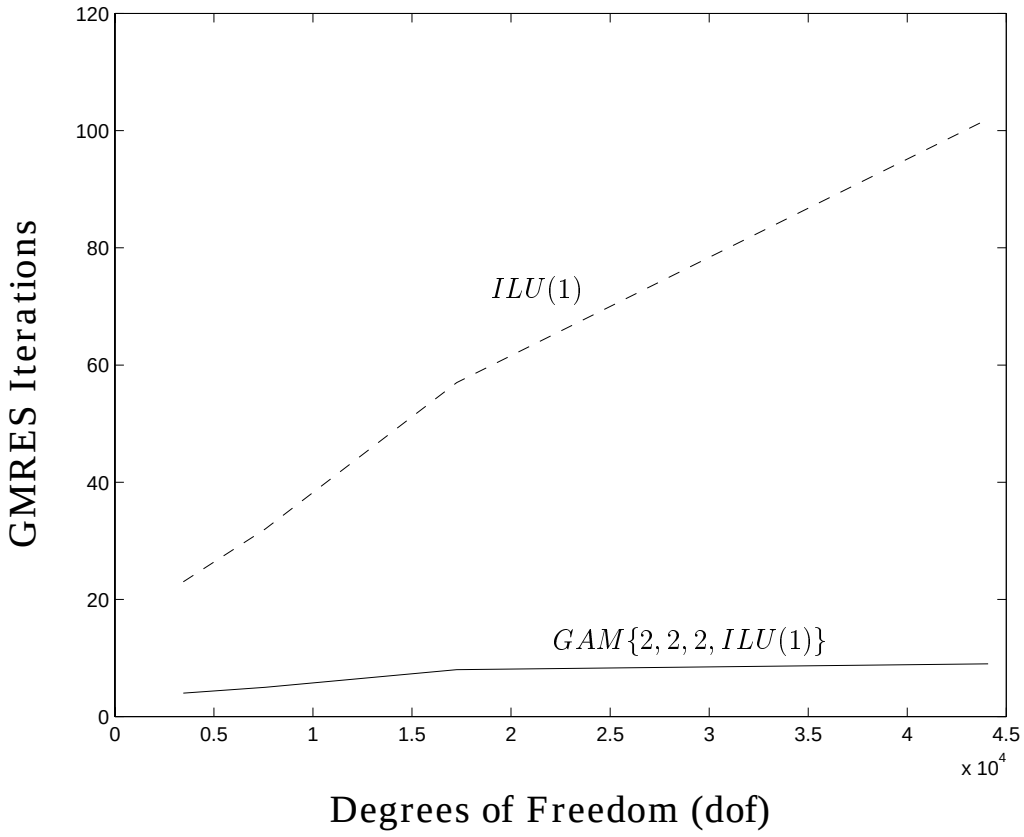


Figure 16: Iteration Count vs. Problem Size for $ILU(1)$ and $GAM\{2, 2, 2, ILU(1)\}$ Preconditioners with GMRES Accelerator.

7 Acknowledgment

This work was supported by the Office of Naval Research under grant number N00014-97-1-0687.

Appendix

We include some of the raw data and statistics for the various example problems.

For **Example 1, Figure 10**:

Prec.	Problem Size (dof)					
	5929 (1401)	23,104 (5301)	51,529 (11,701)	91,204 (20,601)	142,129 (32,001)	204,304 (45,901)
$ILU(1)$	16	21	24	27	30	43
GAM	7	9	10	10	10	9

Table 1: Iteration Count vs. Problem Size for $ILU(1)$ and $GAM\{1,1,1,ILU(1)\}$ Preconditioners with GMRES Accelerator (Ex. 1).

where the problem size in parentheses is for the coarse-level problem.

Example 1, Figure 11:

Iterations	Element Peclet Number				
	$\alpha = 0.0$	$\alpha = 0.5$	$\alpha = 1.0$	$\alpha = 50$	$\alpha = \infty$
0	1	1	1	1	1
1	2.48×10^{-1}	3.39×10^{-2}	4.87×10^{-3}	1.61×10^{-1}	4.15×10^{-1}
2	4.61×10^{-2}	1.54×10^{-3}	4.91×10^{-5}	1.01×10^{-1}	2.61×10^{-1}
3	1.02×10^{-2}	7.53×10^{-5}	4.52×10^{-7}	7.37×10^{-2}	2.06×10^{-1}
4	2.03×10^{-3}	3.56×10^{-6}	3.99×10^{-9}	3.48×10^{-2}	1.28×10^{-1}
5	3.94×10^{-4}	1.49×10^{-7}	—	7.67×10^{-3}	4.49×10^{-2}
6	6.99×10^{-5}	6.52×10^{-9}	—	9.77×10^{-4}	9.25×10^{-3}
7	1.39×10^{-5}	2.53×10^{-10}	—	1.13×10^{-4}	1.25×10^{-3}
8	3.13×10^{-6}	—	—	3.00×10^{-5}	2.94×10^{-4}
9	8.22×10^{-7}	—	—	2.99×10^{-6}	6.98×10^{-5}
10	1.61×10^{-7}	—	—	6.90×10^{-7}	2.31×10^{-5}
11	2.89×10^{-8}	—	—	9.83×10^{-8}	3.49×10^{-6}
12	5.60×10^{-9}	—	—	1.72×10^{-8}	7.37×10^{-7}
13	8.67×10^{-10}	—	—	2.97×10^{-9}	1.47×10^{-7}
14	—	—	—	5.50×10^{-10}	3.53×10^{-8}
15	—	—	—	—	8.12×10^{-9}

Table 2: Effect of Peclet Number α on Convergence Rate. Relative Error vs. Iteration Count

where the dashes "—" indicate a relative error of less than 1×10^{-10}.

Example 2, Figure 13:

Prec.	Problem Size (dof)			
	7122 (1444)	22,540 (4385)	50,724 (9354)	115,328 (21,178)
$ILU(0)$	32	120	188	272
GAM	8	11	14	18

Table 3: Iteration Count vs. Problem Size for $ILU(0)$ and $GAM\{2,2,1,ILU(0)\}$ Preconditioners with GMRES Accelerator (Ex. 2)

Example 2, Figure 14:

Process	Problem Size (dof)			
	7122	22,540	50,724	115,328
eigenvector estimation	0.088	0.310	0.860	1.650
aggregation	0.350	1.036	2.590	5.834
compute $\mathbf{K}_0$	0.304	1.006	2.264	5.100
$\mathbf{K}_0 = \mathbf{Q}^T\mathbf{K}\mathbf{Q}$ product	0.398	1.230	2.842	6.072

Table 4: CPU Time (s) vs. Problem Size for the Critical Components of the Two-level Method

Example 3, Figure 16:

Prec.	Problem Size (dof)			
	3450 (1471)	7584 (3025)	17,259 (6775)	44,082 (16,372)
$ILU(1)$	23	32	57	102
GAM	4	5	8	9

Table 5: Iteration Count vs. Problem Size for $ILU(1)$ and $GAM\{2,2,2,ILU(1)\}$ Preconditioners with GMRES Accelerator (Ex. 3).

References

[1] Axelsson, O., and Vassilevski, P. S. Algebraic Multilevel Preconditioning Methods, I. *Numerische Mathematik 56* (1989), pp. 157–177.

[2] Axelsson, O., and Vassilevski, P. S. Algebraic Multilevel Preconditioning Methods, II. *SIAM Journal of Numerical Analysis 27*, 6 (1990), pp. 1569–1590.

[3] Balay, S., Gropp, W. D., McInnes, L. C., and Smith, B. F. Efficient Management of Parallelism in Object Oriented Numerical Software Libraries. In *Modern Software Tools in Scientific Computing* (1997), E. Arge, A. M. Brauset, and H. P. Langtangen, Eds., Birkhauser Press, pp. 163–202.

[4] Balay, S., Gropp, W. D., McInnes, L. C., and Smith, B. F. PETSc 2.0 Users Manual. Tech. Rep. ANL-95/11-Recision 2.0.22, Argonne National Laboratory, 1998.

[5] Balay, S., Gropp, W. D., McInnes, L. C., and Smith, B. F. PETSc home page. http://www.mcs.anl.gov/petsc, 1998.

[6] Brandt, A. Guide to Multigrid Development. In *Multigrid Methods. Proceedings, Köln-Porz 1981. Lecture Notes in Mathematics 960*, W. Hackbusch and U. Trottenberg, Eds. Springer-Verlag, Berlin, Heidelberg, 1982, pp. 220–312.

[7] Brandt, A. Algebraic Multigrid Theory: The Symmetric Case. *Applied Mathematics and Computation 19* (1986), pp. 23–56.

[8] Brandt, A., McCormick, S. F., and Ruge, J. W. Algebraic Multigrid (AMG) for Sparse Matrix Equations. In *Sparsity and its Applications*, D. J. Evans, Ed. Cambridge University Press, 1984, pp. 257–284.

[9] Brooks, A. N., and Hughes, T. J. R. Streamline Upwind/Petrov-Galerkin Formulations for Convection Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equations. *Computer Methods in Applied Mechanics and Engineering 32* (1982), pp. 199–259.

[10] Bulgakov, V. E. Multi-level Iterative Technique and Aggregation Concept with Semi-analytical Preconditioning for Solving Boundary-value Problems. *Communications in Numerical Methods in Engineering 9* (1993), pp. 649–657.

[11] Bulgakov, V. E. Iterative Aggregation Technique for Large-scale Finite Element Analysis of Mechanical Systems. *Computer & Structures 52*, 4 (1994), pp. 829–840.

[12] Bulgakov, V. E., and Kuhn, G. High-performance Multilevel Iterative Aggregation Solver for Large Finite-element Structural Analysis Problems. *International Journal for Numerical Methods in Engineering 38* (1995), pp. 3529–3544.

[13] Ciarlet, P. G. *Introduction to Numerical Linear Algebra and Optimization.* Cambridge University Press, Cambridge, New York, Melbourne, 1989.

[14] De Zeeuw, P. M. Matrix-dependent Prolongations and Restrictions in a Black-box Multigrid Solver. *Journal of Computational and Applied Mathematics 33* (1990), pp. 1–27.

[15] Dendy, J. E. Black Box Multigrid. *Journal of Computational Physics 48* (1982), pp. 366–386.

[16] Dendy, J. E. Black Box Multigrid for Nonsymmetric Problems. *Applied Mathematics and Computation 13* (1983), pp. 261–283.

[17] Dendy, J. E. Black Box Multigrid for Systems. *Applied Mathematics and Computation 19* (1986), pp. 57–74.

[18] Fedorenko, R. P. A Relaxation Method for Solving Elliptic Difference Equations. *USSR Computational Mathematics and Mathematical Physics 1*, 5 (1962), pp. 1092–1096.

[19] Fish, J., and Belsky, V. Generalized Aggregation Multilevel Solver. *Intenational Journal for Numerical Methods in Engineering 40* (1997), pp. 4341–4361.

[20] Giddings, T. E., Fish, J., and Rusak, Z. A Stabilized Finite Element Formulation for the Transonic Small-Disturbance System of Equations. *to be submitted.*

[21] Hackbusch, W. *Multi-Grid Methods and Applications.* Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1985.

[22] Hackbusch, W. *Iterative Solution of Large Sparse Systems of Equations.* Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1994.

[23] Kettler, R. Analysis and Comparison of Relaxation Schemes in Robust Multigrid and Preconditioned Conjugate Gradient Methods. In *Multigrid Methods. Proceedings, Köln-Porz 1981. (Lecture Notes in Mathematics 960)*, W. Hackbusch and U. Trottenberg, Eds. Springer-Verlag, Berlin, Heidelberg, 1982, pp. 502–534.

[24] McCormick, S. F. An Algebraic Interpretation of Multigrid Methods. *SIAM Journal of Numerical Analysis 19*, 3 (1982), pp. 548–560.

[25] Saad, Y. *Iterative Methods for Sparse Linear Systems.* Pws Pub. Co., 1996.

[26] Stüben, K., and Trottenberg, U. Multigrid Methods: Fundamental Algorithms, Model Problem Analysis and Applications. In *Multigrid Methods. Proceedings, Köln-Porz 1981. (Lecture Notes in Mathematics 960)*, W. Hackbusch and U. Trottenberg, Eds. Springer-Verlag, Berlin, Heidelberg, 1982, pp. 1–176.

[27] Vaněk, P. Acceleration of Convergence of a Two-level Algorithm by Smoothing Transfer Operator. *Applications of Mathematics 37* (1992), pp. 265–274.

[28] Vaněk, P., Mandel, J., and Brezina, M. Algebraic Multigrid on Unstructured Meshes. Tech. Rep. UCD/CCM 34, Center for Computational Mathematics, University of Colorado at Denver, 1994.
http://www-math.cudenver.edu/ccmreports/rep34.ps.gz.

[29] Varga, R. S. *Matrix Iterative Analysis.* Prentice-Hall, Inc., New Jersey, 1962.

[30] Wesseling, P. A Robust and Efficient Multigrid Method. In *Multigrid Methods. Proceedings, Köln-Porz 1981. (Lecture Notes in Mathematics 960)*, W. Hackbusch and U. Trottenberg, Eds. Springer-Verlag, Berlin, Heidelberg, 1982, pp. 614–630.

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

[32] Young, D. M. *Iterative Solutions of Large Linear Systems.* Academic Press, Inc., New York, San Francisco, London, 1971.