PERGAMON Computers and Mathematics with Applications 37 (1999) 57–70

An International Journal
computers & mathematics
with applications

Application of an Element-by-Element BiCGSTAB Iterative Solver to a Monotonic Finite Element Model

T. W. H. SHEU*, C. C. FANG AND S. F. TSAI
Department of Naval Architecture and Ocean Engineering
National Taiwan University
73 Chou-Shan Road, Taipei, Taiwan, R.O.C.
sheu@cindy.na.ntu.edu.tw

(Received April 1997; revised and accepted July 1998)

Abstract—This paper deals with the advection-diffusion equation in adaptive meshes. The main feature of the present finite element model is the use of Legendre-polynomials to span finite element spaces. The success that this model gives good resolutions to solutions in regions of boundary and interior layers lies in the use of M-matrix theory. In the monotonic range of Peclet numbers, the Petrov-Galerkin method performs well in the sense that oscillatory solutions are not present in the flow. With proper stabilization, finite element matrix equations can be iteratively solved by the Lanczos method, used concurrently with local minimization provided by GMRES(1). The resulting BiCGSTAB iterative solver, supplemented with the Jacobi preconditioner, is implemented in an element-by-element fashion. This gives solutions which are computationally feasible for large-scale flow simulations. The results of two computations are presented in support of the ability of the present finite element model to resolve sharp gradients in the solution. As is apparent from this study is that considerable savings in computer storage and execution time are achieved in adaptive meshes through use of the preconditioned BiCGSTAB iterative solver.

Keywords—Adaptive meshes, Legendre-polynomials, M-matrix theory, BiCGSTAB iterative solver, Jacobi preconditioner, Element-by-element, Sharp gradients.

1. INTRODUCTION

Study of the advection-diffusion equation is a subject of fundamental importance because this equation has been viewed as a linear, steady-state model for Navier-Stokes equations in the development of numerical methods for solving fluid dynamics and heat transfer problems. This model equation is also of considerable academic interest because of the accessible analytic solution, which provides a convenient test for benchmarking the discretization methods so far devised.

Solution accuracy, numerical stability, and scheme consistency are among issues addressed in the numerical modeling of an advection-diffusion transport equation. Besides these issues, computational efficiency and ease of programming also must be considered for a scheme to be

*Author to whom all correspondence should be addressed.
The authors would like to thank the Computer Center of National Taiwan University and the National Center for High-performance Computing (NCHC) for providing CRAY J916 and IBM RS/6000-590 computers, which made this study possible.

0898-1221/99/$ - see front matter Typeset by AMS-TEX
PII: S0898-1221(99)00046-2

termed robust. However, these properties are rarely achieved concurrently. A scheme endowed with higher solution accuracy may be offset by solution instability. Retaining solution stability without compromising computational efficiency and prediction accuracy constitutes the goal of the present study.

We begin by describing in Section 2 the advection-diffusion equation. This is followed by a description of the finite element model which accommodates the monotonicity-preserving property. The main attribute of the adopted finite element model for the advection-diffusion equation is that weighting functions are spanned by Legendre polynomials. This facilitates numerical integration. To broaden the application scope, we add grid adaptivity to the formulation as described in Section 3. In Section 4, we present a preconditioned BiCGSTAB iterative solver to solve the finite element matrix equation in an element-by-element fashion. In order to validate the proposed monotonic flux discretization scheme, we will present in Section 5 a closed-form solution for the scalar transport equation defined in a simple domain. Attention is directed towards assessing the effectiveness of the employed h-adaptivity used together with the preconditioned BiCGSTAB iterative solver.

2. THEORETICAL ANALYSIS

2.1. Model Equation and Discretization Method

The flow regime of interest is modeled by the following advection-diffusion for a passive scalar variable Φ

$$u\Phi_x + v\Phi_y = \mu(\Phi_{xx} + \Phi_{yy}). \tag{2.1}$$

The flow conditions under investigation are those with constant velocities of u and v and a fixed value of the diffusion coefficient μ. Due to the elliptic nature of the partial differential equation (2.1), we require that boundary conditions be prescribed along the entire boundary of D.

Finite element solutions, $\hat{\Phi}$, to the advection-dominated transport equation for Φ in (2.1) result from enforcing an orthogonality between the residuals of the equation, $R = u\hat{\Phi}_x + v\hat{\Phi}_y - \mu(\hat{\Phi}_{xx} + \hat{\Phi}_{yy})$, and the test functions. Solutions thus obtained are viewed as a search for weak solutions to equation (2.1). Standard finite element procedures are carried out with substitution of the presently employed 4-node isoparametric bilinear basis functions into the weighted residuals statement, thus forming a matrix equation for each element. This is followed by assembling of finite elements to construct a global coefficient matrix. To close the algebraic equations, test and basis finite element spaces must be selected. Construction of a best-suited pair of finite element spaces is of pivotal importance to the analysis and warrants further discussion given below.

2.2. Finite Element Model

While the Petrov-Galerkin model provides much more stability in the numerical modeling of equation (2.1), the upwind method is not free of shortcomings. In the presence of high gradient solutions, a flow-oriented flux discretization scheme no longer suffices for production of oscillation-free solutions. Bounding methods developed for suppressing over- or under-shoots in the solution are mostly formulated within the one-dimensional context.

Monotonic solutions can be obtained by, among other ways, adopting the Total Variational Diminishing (TVD) property [1] or by applying flux limiters [2]. Due to the lack of a sound theoretical framework, extension of these filtering techniques to multidimensional analyses is still beyond our current ability. The Flux Corrected Transport (FCT) algorithm of Boris and Book [3], generalized later by Zalesak [4], is regarded as the first multidimensional shock-capturing scheme so far developed. The other well-known multidimensional finite element formulation is due to Hughes and Mallet [5]. They introduced a discontinuity-capturing operator to their streamline formulation of advective-diffusive differential equations [6]. The added benefit is that

the discontinuity-capturing operator provides a mechanism to control over sharp gradients in boundary and interior layers. In this article, we have no intention to justify whether our scheme will outperform other bounding schemes.

The basis for the development of our bounded finite element model is construction of a stiffness matrix of the M-matrix type. Since the M-matrix theory serves as a guideline for constructing a monotonic scheme, it is instructive to present here some useful definitions and theorems [7–9].

DEFINITION 1. *A real $n \times n$ matrix $\underline{\underline{\mathbf{A}}} = (a_{ij})$ is classified as being irreducible diagonally dominant if $|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}|$ for at least one i.*

THEOREM 1. *Consider a matrix $\underline{\underline{\mathbf{A}}} = (a_{ij})$ which is a real, irreducible diagonally dominant $n \times n$ matrix with the properties $a_{ij} \leq 0$ for $i \neq j$ and $a_{ii} > 0$ for $1 \leq i \leq n$; then $\underline{\underline{\mathbf{A}}}^{-1} > 0$ holds.*

DEFINITION 2. *A real $n \times n$ matrix $\underline{\underline{\mathbf{A}}} = (a_{ij})$ with $a_{ij} \leq 0$ for all $i \neq j$ is called an M-matrix if $\underline{\underline{\mathbf{A}}}$ is nonsingular and $\underline{\underline{\mathbf{A}}}^{-1} > 0$.*

DEFINITION 3. *A real $n \times n$ matrix $\underline{\underline{\mathbf{A}}}$ is defined as monotonic if $\underline{\underline{\mathbf{A}}}\,\underline{\phi} \geq 0$ holds for any vector $\underline{\phi}$; this implies $\underline{\phi} \geq 0$.*

THEOREM 2. *If the off-diagonal entries of $\underline{\underline{\mathbf{A}}}$ are nonpositive, then we are led to a monotonic matrix equation $\underline{\underline{\mathbf{A}}}$ if and only if $\underline{\underline{\mathbf{A}}}$ is an M-matrix.*

Along the line of our previous study [10], we construct test functions as below:

$$W_i = D_i \left[d_{\xi_0} P_0(\xi) + d_{\xi_1} P_1(\xi)\right] \left[d_{\eta_0} P_0(\eta) + d_{\eta_1} P_1(\eta)\right]. \tag{2.2}$$

It is noteworthy that the matrix equation taking an M-matrix form is a key to permitting bounded solutions. Specific to this Petrov-Galerkin finite element analysis is that the weighting functions favor the upwind side and are spanned by Legendre polynomials $P_0(t) = 1$ and $P_1(t) = t$. For further details on the justification for using the weighting functions given in equation (2.2), the reader is referred to [10]. The coefficients shown in equation (2.2) are summarized as follows for $i = 1 \sim 4$ and $n = 0, 1$:

$$D_i = \frac{1}{4} \exp\left(\frac{u h_\xi \xi_i}{2\mu}\right) \exp\left(\frac{v h_\eta \eta_i}{2\mu}\right), \tag{2.3}$$

$$d_{\xi_n} = \frac{2n+1}{2} \int_{-1}^{1} W_\xi(t) P_n(t)\, dt, \tag{2.4}$$

$$d_{\eta_n} = \frac{2n+1}{2} \int_{-1}^{1} W_\eta(t) P_n(t)\, dt, \tag{2.5}$$

$$e_{\xi_n} = \frac{2n+1}{2} \int_{-1}^{1} W'_\xi(t) P_n(t)\, dt, \tag{2.6}$$

$$e_{\eta_n} = \frac{2n+1}{2} \int_{-1}^{1} W'_\eta(t) P_n(t)\, dt, \tag{2.7}$$

where h_ξ and h_η denote grid sizes and

$$W_\xi(\xi) = (1 + \xi_i \xi) \exp\left(-\frac{u h_\xi \xi}{2\mu}\right), \tag{2.8}$$

$$W_\eta(\eta) = (1 + \eta_i \eta) \exp\left(-\frac{v h_\eta \eta}{2\mu}\right). \tag{2.9}$$

The weighting function given in (2.2) allows higher-order differentiation. The advantage of increased smoothness in the weighting function is, however, offset by a marked increase in the

Gaussian integration points required for exact integration. We thus utilize the orthogonal property inherent in the space of the Legendre polynomials to eliminate this drawback:

$$\int_{-1}^{+1} P_i(t)P_j(t)\,dt = \frac{2}{2i+1}\delta_{ij}, \qquad (i \text{ is dummy index}). \tag{2.10}$$

It is the above integral identity that dramatically reduces the number of Gaussian integration points needed for an analytic integration and, thus, causes the CPU time to decrease substantially. For this reason, we are prompted to rewrite the bilinear shape functions $N_i(\xi,\eta)$ in terms of Legendre polynomials:

$$N_i(\xi,\eta) = \frac{1}{4}\left[P_0(\xi) + \xi_i P_1(\xi)\right]\left[P_0(\eta) + \eta_i P_1(\eta)\right], \qquad (i = 1 \sim 4). \tag{2.11}$$

In the following, we will examine whether this upwind model can unconditionally yield monotonic solutions. In order to make use of the M-matrix theory, we rewrite the finite element equation in a form similar to that in the finite-difference setting. For a point at $j = 5$ in Figure 1, it is rather cumbersome to obtain the functional expressions of a_i in $\sum_1^9 a_i\,\Phi_i = 0$ from a pack of four bilinear elements. To show that the coefficients $a_1 \sim a_9$ fall into the M-matrix category, we have calculated a_i against Pe_x and Pe_y. For values (Pe_x, Pe_y) which fall into the shaded area shown in Figure 2, a matrix equation manifested by the coefficients $a_1 \sim a_9$ in each row is, by definition, an M-matrix equation.

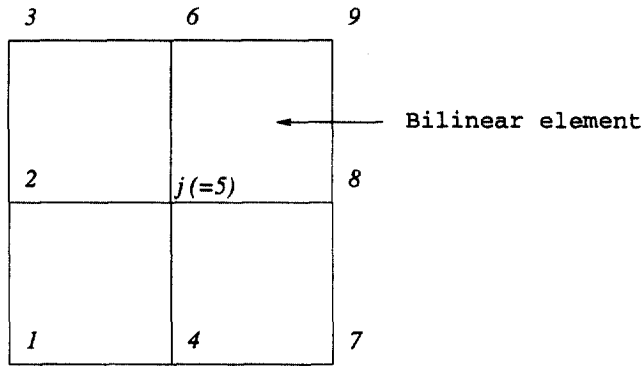


Figure 1. Illustration of nodal points in a pack of four bilinear elements.

3. REFINEMENTS ON LEGENDRE-POLYNOMIAL FINITE ELEMENT MODEL-SOLUTION ADAPTIVITY

Even though it can yield an M-matrix equation, the Legendre-polynomial finite element model can only be useful over a limited range of Peclet numbers. This rather restrictive constraint on the range of Peclet numbers limits application to practical calculations. Acknowledging this restriction, our efforts are directed towards refinement of this model. We can, of course, continuously refine the mesh until the Peclet numbers fall within the stable region shown in Figure 2. While this refinement will yield oscillation-free solutions, the computational cost will be prohibitively high, limiting extension of this theoretically appealing model to practical flow simulations. A plausible remedy for this difficulty is to adopt grid adaption as a way to locally decrease the Peclet number. Grid adaption terminates until the maximum Peclet number falls below the critical values 3.6.

Our refinement strategy is to add mesh points solely in regions where Peclet numbers exceed the critical value. As a result, the grid adaption is that of the dynamic grid adaption. Among

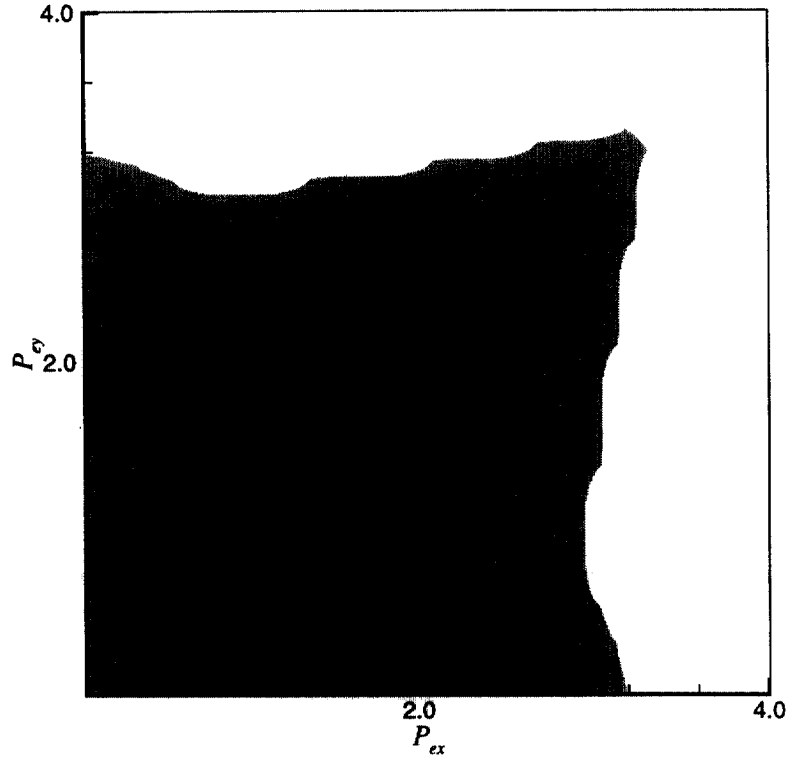


Figure 2. Stable (monotone) region plotted in terms of $Pe_x = u\Delta x/\mu$ and $Pe_y = v\Delta y/\mu$.

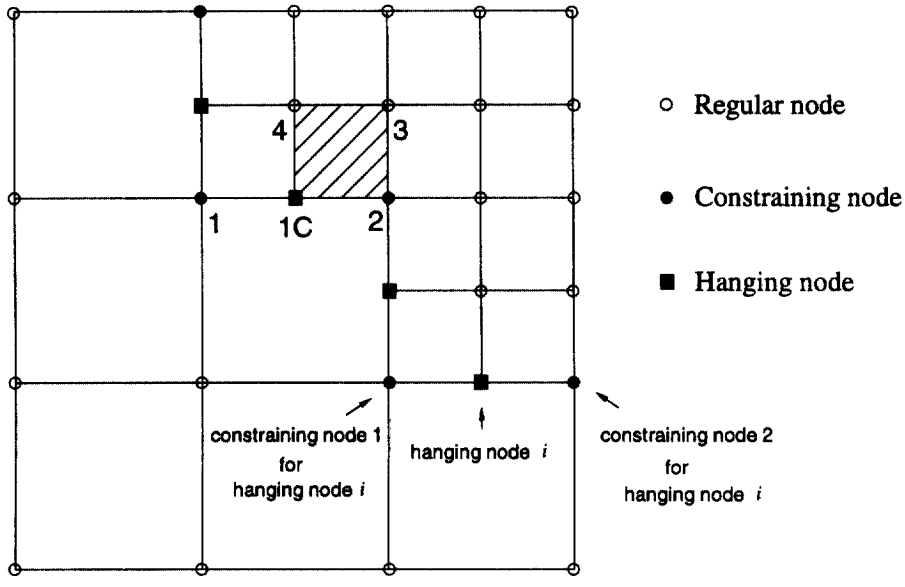


Figure 3. Illustration of the hanging and constraining nodes.

the mesh refinement strategies that have been proven, we adopt the strategy of bisecting the element. Use of this strategy as a means to refine the mesh results in hanging nodes. Since the numerical solution in an element is approximated by the bilinear shape function, solutions at hanging nodes, say 1C in Figure 3, are not considered as degrees of freedom but are rather

constrained by the following relation:

$$u_{1C} = \frac{1}{2}(u_1 + u_2). \tag{3.1}$$

The above constrained relation causes the coefficient matrix for an element, $\mathbf{M}^e$, and the assembled matrix, $\mathbf{M}$, to be modified. Following the approach of Demkowicz *et al.* [11], for entries at hanging nodes, the associated rows and columns of the global mass matrix are eliminated. This is followed by the following redistributions:

$$M_{k_r(i),\hat{j}} = M_{k_r(i),\hat{j}} + \frac{1}{2} M_{i,\hat{j}}, \tag{3.2}$$

$$M_{\hat{i},l_s(j)} = M_{\hat{i},l_s(j)} + \frac{1}{2} M_{\hat{i},j}, \tag{3.3}$$

$$M_{k_r(i),l_s(j)} = M_{k_r(i),l_s(j)} + \frac{1}{4} M_{i,j}. \tag{3.4}$$

The subscripts i, j denote the indices of the rows and columns of the hanging node in the algebraic equation. In equations (3.2)–(3.4), $k_r(i)$, $l_s(j)$ are indices of the rows and columns of the corresponding constraining node. Here, r, s are equal to 1 or 2 for the two constraining nodes. We denote by $\hat{i}$, $\hat{j}$ indices for rows and columns other than those of the hanging node. With matrices thus modified, the vector $\mathbf{b}$ varies accordingly as follows:

$$b_{k_r(i)} = b_{k_r(i)} + \frac{1}{2} b_i. \tag{3.5}$$

4. PRECONDITIONED ELEMENT-BY-ELEMENT BICGSTAB ITERATIVE SOLVER

4.1. Literature Review

With definitions of test and basis spaces, finite element solutions to the working equation (2.1) can be solved either using a direct or iterative solver. The question may now be raised whether direct solvers will out-perform iterative solvers. It is the opinion of the authors that direct solvers are better suited for two-dimensional analyses, whereas iterative solvers are more practical for truly three-dimensional calculations. The reason is that storage demands can be prohibitive in fill-in processes using direct solvers with the increasing size of problems. Recognizing this, direct solvers are the preferred choice for the present two-dimensional analysis. However, for future extension of the adaptive Legendre-polynomial finite element model to practical applications in three-dimensions, it is a good idea to apply an iterative solver to simpler problems to gain some understanding of iterative solvers.

Typical of iterative solvers is that the iteration number needed to reach convergence tolerance criterion increases dramatically with the increase in the number of grid points. Iterative solvers constructed by the minimization idea turn out to be effective in the reduction of the iteration number. The conjugate gradient method of Hestenes and Stiefel [12] is a representative of this class of solvers. The idea of the conjugate gradient method is to find solutions $\underline{\mathbf{x}}$ from $\underline{\underline{\mathbf{A}}}\,\underline{\mathbf{x}} = \underline{\mathbf{b}}$ via the following sequence of calculations:

$$\underline{\mathbf{x}}_i \in \underline{\mathbf{x}}_0 + \text{span}\left(\underline{\mathbf{r}}_0, \underline{\underline{\mathbf{A}}}\,\underline{\mathbf{r}}_0, \ldots, \underline{\underline{\mathbf{A}}}^{i-1}\,\underline{\mathbf{r}}_0\right). \tag{4.1}$$

In the above, $\underline{\mathbf{x}}_0$ is the initial vector and

$$\underline{\mathbf{r}}_0 = \underline{\mathbf{b}} - \underline{\underline{\mathbf{A}}}\,\underline{\mathbf{x}}_0. \tag{4.2}$$

The span $(\underline{\mathbf{r}}_0, \underline{\underline{\mathbf{A}}}\,\underline{\mathbf{r}}_0, \ldots, \underline{\underline{\mathbf{A}}}^{i-1}\,\underline{\mathbf{r}}_0)$ involved in the iterative sequence (4.1) is known as the Krylov subspace. Worthy noting is that this optimization procedure works effectively only for a matrix

equation having clustered eigenvalues. Use of the conventional conjugate gradient method suffers from pivoting breakdown when the symmetry of the matrix is lost.

Research into designing of Krylov subspace methods with the ability to resolve problems of matrix asymmetry has grown during the last two decades. One possibility is to deal with normal equations, among which the Conjugate Gradient method on the Normal Residual (CGNR) and Normal Equations (CGNE) [13] are representative examples. Experience has shown that computational benefit is gained from this normalization when matrix equations are symmetrized since the condition number for an equivalent normal equation becomes much larger than that of the original stiffness matrix $\underline{\underline{\mathbf{A}}}$.

In the literature, two classes of nonstationary iterative methods have the ability to resolve matrix asymmetry and are frequently referred to. The Chebyshev method is classified as a nonstationary iterative method and is only applicable to positive definite equations. Also, use of this method requires knowledge of the spectrum *a priori*. To circumvent deficiencies inherent in the Bi-Conjugate Graduate (Bi-CG) method [14], namely the irregular convergence behavior and the indispensable transpose operation of the coefficient matrix, the Arnoldi or Lanczos algorithm was proposed. Like the Arnoldi algorithm, the generalized minimized residuals (GMRES) method [15] accommodates a self-orthogonal sequence. Due to the prohibitive storage demand, the residual can be minimized optimally through an incorporation of a restart capability. In the iteration, no more than n steps are needed for a n by n matrix to reach convergence.

In the Lanczos context, product methods such as Conjugate Gradient Squares (CGS) [16], Quasi-Minimal Residual (QMR) [17], and Bi-conjugate gradient stabilized (BiCGSTAB) [18], are preferable methods for unsymmetric equations. The exploration of a dual orthogonal vector set forms the building block of this class of methods. The QMR method of Freund and Nachtigal [17] was designed to avoid irregular convergence behavior. This method, unfortunately, suffers from a need to transpose the stiffness matrix. CGS, on the other hand, obviates the need for $\underline{\underline{\mathbf{A}}}^{\mathsf{T}}$ but inhibits irregular convergence behavior because this method accommodates the contraction polynomial of Bi-CG. Besides the transpose-free version of QMR [19], the BiCGSTAB method of Van der Vorst [18] is another rational alternative. BiCGSTAB was developed underlying the Lanczos method, in conjunction with the local minimization method GMRES(1). Through manipulation of equal-order contraction polynomials of different kinds, we can dispense with transpose matrix procedures and suppress the irregular convergence behavior. Nevertheless, much work still needs to be done to avoid the pivoting breakdown and Lanczos breakdown mainly because BiCGSTAB still inhibits some features of Bi-CG.

4.2. Element-by-Element Preconditioned BiCGSTAB Method

It has been well known that an effective use of an iterative method depends highly on the nonzero profile of the coefficient matrix. The strategies of ordering nodal points and allocating working variables warrant consideration because they have a direct effect on the band-width of the coefficient matrix and, thus, the sparsity of the matrix. Since we encounter a sparse matrix equation in the finite element analysis, a useful means of storing the matrix in the core memory is needed. Like the compressed matrix used in the finite difference setting, we can store a matrix at the element level in order to dispense with unnecessary storage of voids. This motivates us to conduct finite element analysis in an Element-by-Element (EBE) fashion. With this goal in mind, we incorporate the element-by-element capability into the presently employed BiCGSTAB iterative solver.

Current state-of-the-art iterative solvers have not advanced to the point where the potential gain in speed can be consistently realized in matrix equations which are typically encountered in high-Peclet number flows. Their performance deteriorates dramatically when the properties of matrix equations with diagonally dominant, positive-definite, and symmetric coefficient matrices are lost. A way of obtaining a better convergence is thus needed in the use of iteration solvers.

Preconditioning of a matrix is considered as a good way to change the original system of linear equations into an equivalent one that has a coefficient matrix with properties that are more amenable to iterative solutions. Therefore, the choice of a good preconditioner is absolutely crucial to the successful application of an iterative solver. However, there lacks a convincing guideline to design an appropriate preconditioner for matrix equations encountered in the present analysis. The preconditioner chosen for the present analysis is that of the Jacobi preconditioner. We address the inclusion of this preconditioner into the present finite element formulation which helps optimize the performance of the iterative solver BiCGSTAB.

The resulting EBE-BiCGSTAB procedures can be briefly described as follows:

Compute $\underline{\mathbf{r}}_0 = \underline{\mathbf{b}} - \underline{\underline{\mathbf{A}}}\,\underline{\mathbf{x}}_0$ from the initial guess $\underline{\mathbf{x}}_0$
Restrict $\underline{\mathbf{r}}_0$
Choose $\overline{\underline{\mathbf{r}}}$, such that $(\overline{\underline{\mathbf{r}}}, \underline{\mathbf{r}}_0) \neq 0$
For $i = 1, 2, \ldots$
  $\rho_{i-1} = (\overline{\underline{\mathbf{r}}}, \underline{\mathbf{r}}_{i-1})$
  if $\rho_{i-1} < \epsilon_1$ **[near break down]**
  if $i = 1$
    $\underline{\mathbf{p}}_i = \underline{\mathbf{r}}_{i-1}$
  else
    $\beta_{i-1} = (\rho_{i-1}/\rho_{i-2})(\alpha_{i-1}/\omega_{i-1})$
    $\underline{\mathbf{p}}_i = \underline{\mathbf{r}}_{i-1} + \beta_{i-1}(\underline{\mathbf{p}}_{i-1} - \omega_{i-1}\underline{\mathbf{v}}_{i-1})$
  endif
  solve $\underline{\underline{\mathbf{K}}}\,\overline{\underline{\mathbf{p}}} = \underline{\mathbf{p}}_i$ **[preconditioning]**
  $\underline{\mathbf{v}}_i = \sum_{\text{elem}}(\underline{\underline{\mathbf{A}}}_{\text{elem}}\overline{\underline{\mathbf{p}}})$ ⟵ element-by-element procedure
  Restrict $\underline{\mathbf{v}}$
  $\alpha_i = \rho_{i-1}/(\overline{\underline{\mathbf{r}}}, \underline{\mathbf{v}}_i)$
  if $(\overline{\underline{\mathbf{r}}}, \underline{\mathbf{v}}_i) < \epsilon_2$ **[near break down]**
  $\underline{\mathbf{s}} = \underline{\mathbf{r}}_{i-1} - \alpha_i\underline{\mathbf{v}}_i$
  solve $\underline{\underline{\mathbf{K}}}\,\overline{\underline{\mathbf{s}}} = \underline{\mathbf{s}}$ **[preconditioning]**
  $\underline{\mathbf{t}} = \sum_{\text{elem}}(\underline{\underline{\mathbf{A}}}_{\text{elem}}\overline{\underline{\mathbf{s}}})$ ⟵ element-by-element procedure
  Restrict $\underline{\mathbf{t}}$
  if $\|\underline{\mathbf{s}}\|_2 < \epsilon$
    $\omega_i = 0$
  else
    $\omega_i = (\underline{\mathbf{t}}, \underline{\mathbf{s}})/(\underline{\mathbf{t}}, \underline{\mathbf{t}})$
  endif
  $\underline{\mathbf{x}}_i = \underline{\mathbf{x}}_{i-1} + \alpha_i\overline{\underline{\mathbf{p}}}_i + \omega_i\overline{\underline{\mathbf{s}}}$
  Distribute $\underline{\mathbf{x}}_i$
  $\underline{\mathbf{r}}_i = \underline{\mathbf{b}} - \underline{\underline{\mathbf{A}}}\,\underline{\mathbf{x}}_i$
  Restrict $\underline{\mathbf{r}}_i$
  check convergence; continue if necessary ($\omega_i \neq 0$)
End

The $\underline{\underline{\mathbf{K}}}$ is denoted as the preconditioning matrix. By definition, the **Restrict** $\underline{\mathbf{r}}$ procedure is given by

$$r_{k_r} = r_{k_r} + \frac{1}{2}r_i,$$

which takes a form similar to that shown in equation (3.5). In the above procedures, **Distribute** $\underline{\mathbf{x}}$ is defined as

$$x_i = \frac{1}{2}(x_{k_1} + x_{k_2}),$$

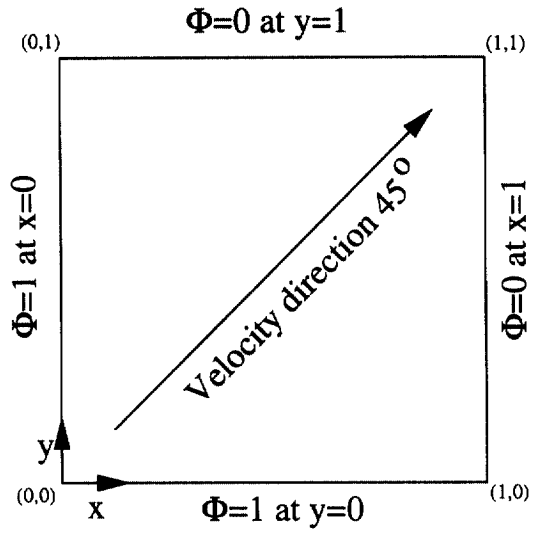


Figure 4. Configuration and boundary conditions of Φ for the problem defined in Section 5.1.

Table 1. Comparison of CPU seconds for solving the problem in Section 5.1 using different solution solvers. Elements: 2044, Nodes: 2365. Convergent tolerance is 1.0×10^{-4}.

Frontal	BiCGSTAB	Preconditioning BiCGSTAB
521.31	170.12	112.65

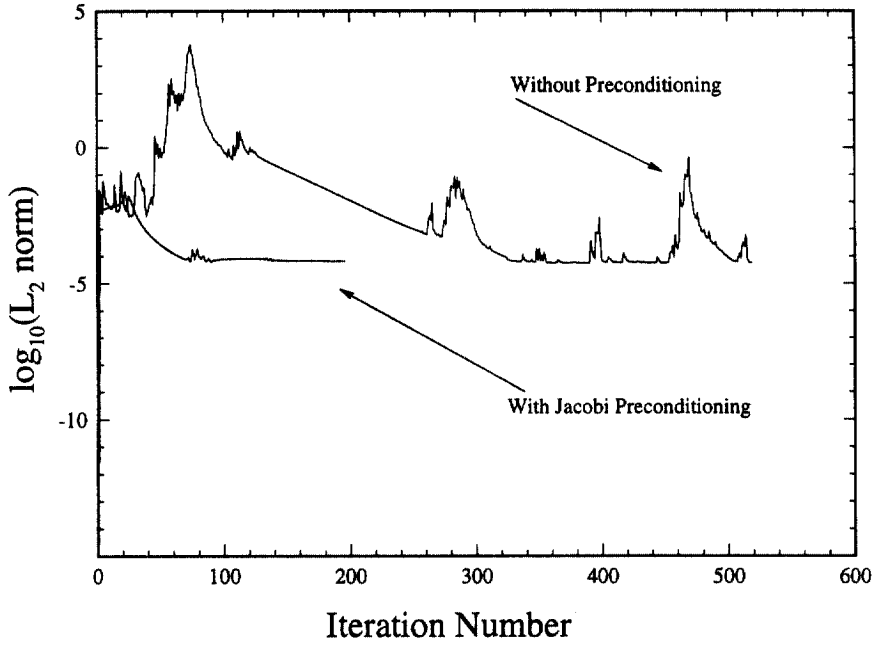


Figure 5. Residual reductions against iteration numbers using the BiCGSTAB iterative solver with/without preconditioner.

which takes the similar form as that in equation (3.1). The way we deal with hanging nodes in the element-by-element procedures is identical to Huang [20].

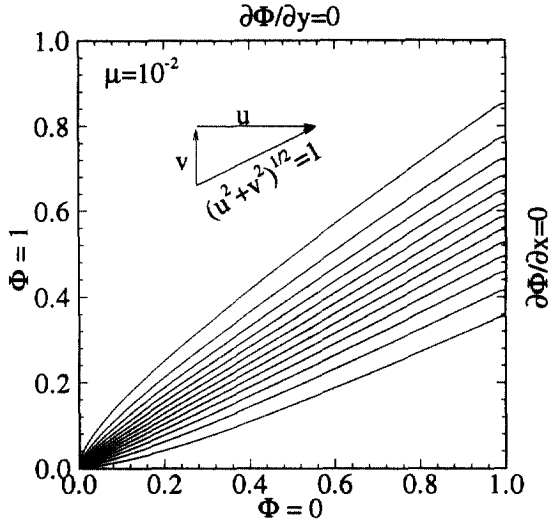


Figure 6. Configuration for the problem defined in Section 5.2.

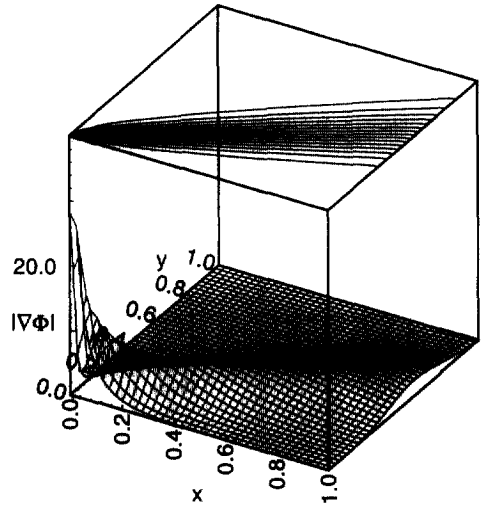


Figure 7. Computed solution Φ and $|\nabla\Phi|$ for the case with a grid resolution of 160 x 160 defined in Section 5.2.

5. NUMERICAL RESULTS

5.1. Analytic Validation

The first problem under consideration is configured in Figure 4. Due to the simplicity of its geometry, this problem is, thus, taken as a test case for demonstrating the potential of the h-adaptive Legendre-polynomial finite element model for modeling a boundary layer profile in the flow. Subject to the prescribed boundary data of Φ, the advection-diffusion equation is analytically amenable to the following boundary-layer type solution [21]:

$$\Phi(x,y) = \frac{[1-\exp((x-1)(u/\mu))]}{[1-\exp(-u/\mu)]}\frac{[1-\exp((y-1)(v/\mu))]}{[1-\exp(-v/\mu)]}. \tag{5.1}$$

Finite element solutions are sought on dynamically adoptive grids (3952 elements, 4593 mesh points, 584 slave nodes). We cast prediction errors for the case considered ($\mu = 2\times10^{-3}$) in their L_2-norm form. The error computed in the adaptive mesh is 3.415×10^{-3} while in the uniform

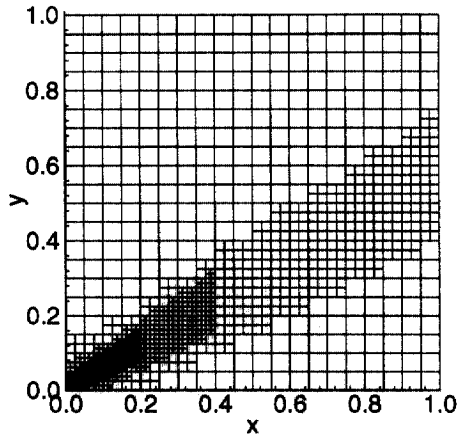


Figure 8. The mesh for the problem defined in Section 5.2.

mesh with a total number 4593 nodes is 0.1197. Revealed from this study is that the solution accuracy improves with the introduction of grid adaptivity. For comparison purposes, calculations were performed for cases involving use of Frontal solver, BiCGSTAB iterative solver, and the preconditioned BiCGSTAB solver. According to the CPU seconds shown in Table 1, we conclude that use of iterative solvers outperforms their direct solver counterparts. Also revealed from Figure 5, which plots the residuals against iteration numbers for iterative solvers with/without use of preconditioner, is the explanation for the saving of CPU seconds, as shown in Table 1, using the Jacobi preconditioner.

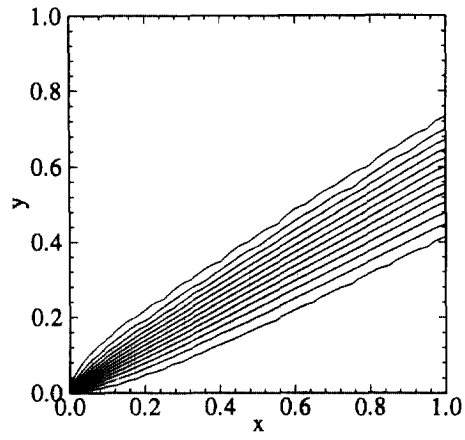
5.2. Skew Advection-Diffusion Problem

We next proceed to analysis of an even more difficult problem, namely, the skewed flow transport problem. This problem is featured as having an interior layer and is regarded as a worst case scenario for upwinding methods so far developed [22]. As Figure 6 shows, there is a tilted line, with an angle of $\theta = \tan^{-1} v/u$, which divides the cavity into two subdomains. In the square of unit length, there is a uniform flow which is parallel to the dividing line. The magnitude of the velocity remains unchanged with a value of 1.

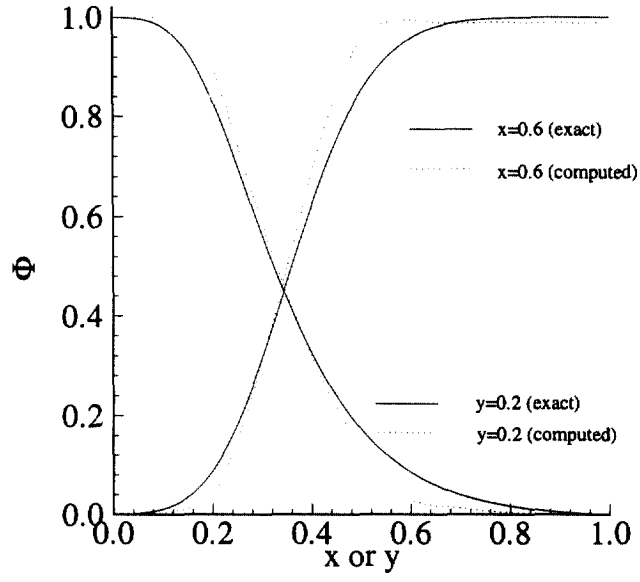
Since the skew-advection problem is not amenable to analytic solution, the solution computed in the uniform grids with a resolution of 161×161 is taken as the exact solution for this problem. According to Figure 7 which plots the contours of Φ and its gradient $|\nabla \Phi|$, as computed from 161×161 finite element solutions, we are led to know that high gradient solutions are present in the immediate vicinity of the dividing line, especially in the corner region near (0,0). In the adapitve finite element calculation, grids are accordingly refined in high gradient regions, leading to an unstructured grid shown in Figure 8. As Figure 9 shows, monotonic solutions are also adaptively computable in the flow which has fewer elements. This clearly demonstrates the algorithmic superiority of the adaptive monotonic scheme adopted here as a way of resolving interior sharp layers. Performance comparison is made on the use of BiCGSTAB and preconditioned BiCGSTAB iterative solvers. Clearly seen from Figure 10 is the advantage of incorporating the Jacobi preconditioner into the iterative solutions. The reduction of iteration numbers to reach the convergence tolerance is evidenced in the CPU times reported in Table 2. In conclusion, we benefit greatly by using grid adaption and the preconditioner used together with the BiCGSTAB iterative solver.

6. CONCLUDING REMARKS

For this study, Legendre polynomials have been chosen to span finite element spaces. By virtue of the orthogonal property in the Legendre polynomials, we can analytically calculate integral



(a) The contour plots of Φ for the problem defined in Section 5.2 using the preconditioned BiCGSTAB iterative solver.



(b) Solution profiles of $\Phi(0.6, y)$ and $\Phi(x, 0.2)$.

Figure 9.

terms using many fewer Gaussian integration points. Notable in the development of the finite element model is that use of the M-matrix theory helps us to clarify whether or not the discretization scheme accommodates the monotone property. For Peclet numbers smaller than the critical value, the discrete system can be classified as an M-matrix; thus, monotonic solutions are computable. This limitation explains the motivation behind our seeking an adaptive technique. Guided by the discrete maximum principle, we are able to determine which grid warrants further refinement so as to reduce the local Peclet number. In the adaptive mesh refinement, remeshing procedure is invoked only in regions of Peclet numbers whose values are above critical values of 3.6. The inclusion of h-adaptive capability provides the ability to resolve high-gradient profiles in the flow. In an attempt to further improve the computational performance, we have incorporated the element-by-element capability into the finite element analysis using the BiCGSTAB as the iterative solver for the unsymmetric and indefinite matrices. This matrix solver is used in conjunction with the Jacobi preconditioner in the hope of yielding an equivalent matrix with

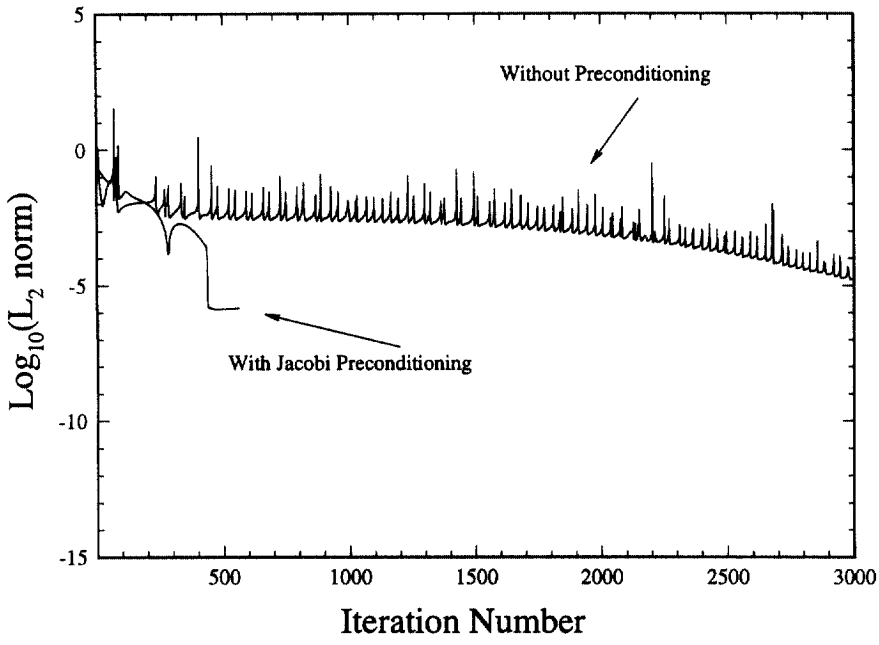


Figure 10. Comparison of the reduction of L_2-error norms against iteration number for the problem defined in Section 5.2 using the BiCGSTAB iterative solver with/without preconditioner.

Table 2. Comparison of CPU seconds for solving the problem in Section 5.2 using different solution solvers. Elements: 3616, Nodes: 3772. Convergent tolerance is 1.0×10^{-5}.

Frontal	BiCGSTAB	Preconditioning BiCGSTAB
806.21	936.94	295.51

properties that are more amenable to iterative solutions. The integrity of the finite element model developed here and the capabilities incorporated have been demonstrated by way of examples.

REFERENCES

1. A. Harten, High resolution schemes for hyperbolic conservation law, *J. Comput. Phys.* **49**, 357–393, (1983).
2. P.K. Sweby, High resolution schemes using flux limiters for hyperbolic conservation laws, *SIAM J. for Numerical Analysis* **21**, 995–1011, (1984).
3. J.P. Boris and D.L. Book, Flux corrected transport I SHASTA, A fluid transport algorithm that works, *J. Comput. Phys.* **11**, 38–69, (1973).
4. S.T. Zalesak, Fully multidimensional flux-corrected transport algorithm for fluids, *J. Comput. Phys.* **31**, 335–362, (1979).
5. T.J.R. Hughes and M. Mallet, A new finite element formulation for computational fluid dynamics: IV. A discontinuity-capturing operator for multidimensional advective-diffusive systems, *Comput. Meths. Appl. Mech. Engrg.* **58**, 329–336, (1986).
6. T.J.R. Hughes and M. Mallet, A new finite element formulation for computational fluid dynamics: III. The generalized for streamline operator for multidimensional advective-diffusive systems, *Comput. Meths. Appl. Mech. Engrg.* **58**, 305–328, (1986).
7. T. Meis and U. Marcowitz, Numerical solution of partial differential equations, In *Applied Mathematical Science*, Volume 32, Springer-Verlag, (1981).
8. T. Ikeda, Maximal principle in finite element models for convection-diffusion phenomena, In *Lecture Notes in Numerical and Applied Analysis*, Volume 4, North-Holland, Kinokuniya, (1983).
9. M. Ahués and M. Telias, Petrov-Galerkin scheme for the steady state convection diffusion equation, *Finite Elements in Water Resources* (2/3), (1982).
10. T.W.H. Sheu, S.F. Tsai and M.M.T. Wang, A monotone finite element method with test space of Legendre polynomials, *Comput. Meths. Appl. Mech. Engrg.* **143**, 349–372, (1997).

11. L. Demkowicz, J.T. Oden, W. Rachowicz and O. Hardy, Toward a universal *h-p* adaptive finite element strategy, Part 1, Constrained approximation and data structure, *Comput. Methods Appl. Mech. Engrg.* **77**, 79–112, (1989).
12. M. Hestenes and E. Stiefel, Methods of conjugate gradients for solving linear systems, *J. Research NBS* **49**, 409–436, (1952).
13. N. Nachtigal, S. Reddy and L. Trefethen, How fast are nonsymmetric matrix iterations?, *SIAM J. Matrix Anal. Appl.* **13**, 778–795, (1992).
14. R. Fletcher, Conjugate gradient methods for indefinite systems, *Lecture Notes in Mathematics* **506**, 73–89, (1976).
15. Y. Saad and M.H. Schultz, GMRES: A generalized minimum residual algorithm for solving nonsymmetric linear system, *SIAM J. Sci. Statist. Comput.* **7**, 856–569, (1986).
16. P. Sonneveld, CGS, a fast Lanczos-type solver for nonsymmetric systems, *SIAM J. Sci. Statist. Comput.* **10**, 36–52, (1989).
17. R. Freund and N. Nachtigal, QMR: A quasi-minimal residual method for non-Hermitian linear systems, *Numer. Math.* **60**, 315–339, (1991).
18. H.A. Van der Vorst, BI-CGSTAB: A fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems, *SIAM J. Sci. Statist. Comput.* **13** (2), 631–644, (1992).
19. R.W. Freund, A transpose-free quasi-minimum residual algorithm for non-Hermitian linear systems, *SIAM J. Sci. Comput.* **14**, 470–482, (1993).
20. C.Y. Huang, Adaptive computational methods for fluid structure interaction in internal flows, Quarterly Report #7 to NASA Lewis Research Center, Contract No. NAS3-25196, The Computational Mechanics Company, TR-89-01, (February 1989).
21. H.C. Elman and G.H. Golub, Line iterative methods for cyclically reduced descrete convection-diffusion problems, *SIAM J. Sci. Stat. Comput.* **13** (1), 339–363, (1992).
22. D.F. Griffiths and A.R. Mitchell, On generating upwind finite element methods, (Edited by T.J.R. Hughes), *ASME Monograph* **AMD-34**, 91–104, (1979).