Downloaded by [China Science & Technology University] at 02:41 26 February 2014

ON A MONOTONIC CONVECTION-DIFFUSION SCHEME IN ADAPTIVE MESHES

C. C. Fang, Tony W. H. Sheu, and S. F. Tsai
Department of Naval Architecture and Ocean Engineering,
National Taiwan University, Taipei, Taiwan, Republic of China

In this article we apply our recently proposed upwind model to solve the two-dimensional steady convection-diffusion equation in adaptive meshes. In an attempt to resolve high-gradient solutions in the flow, we construct finite-element spaces through use of Legendre polynomials. According to the fundamental analysis conducted in this article, we confirm that this finite-element model accommodates the monotonicity property. According to M-matrix theory, we know within what range of Peclet numbers the Petrov-Galerkin method can perform well in a sense that oscillatory solutions are not present in the flow. This monotonic region is fairly restricted, however, and limits the finite-element practioner's choices of a fairly small grid size. This limitation forbids application to practical flow simulations because monotonic solutions are prohibitively expensive to compute. Circumvention of this shortcoming is accomplished by remeshing the domain in an adaptive way. To alleviate the excessive memory requirement, our implementation incorporates a reverse Cuthill-McKee (RCM) renumbering technique. Numerical results are presented in support of the ability of the finite-element model developed herein to resolve sharp gradients in the solution. Also shown from these numerical exercises is that considerable savings in computer storage and execution time are achieved in adaptive meshes through use of the RCM element-reordering technique.

INTRODUCTION

The advective-diffusive equation is examined as a linear, steady-state model for Navier-Stokes equations. Research into this model equation has been the subject of fundamental importance in areas of fluid mechanics and heat transfer. This prototype equation is also of academic significance because it is amendable to analytic solution and thus provides a convenient test for benchmarking the advection-diffusion discretization methods so far devised. Practical as well as academic importance explain why much attention has been devoted in the past few decades to numerical investigation of this model problem, and why it is the focus of the present study.

Numerical modeling of a convection-diffusion transport equation involves prediction accuracy, numerical stability, and scheme consistency. Computational

Received 9 April 1997; accepted 10 June 1997.

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.

Address correspondence to Professor Tony W. H. Sheu, Institute of Naval Architecture, National Taiwan University, 73 Chow-Shan Road, Taipei, Taiwan, Republic of China. E-mail: sheu@indy.na.ntu.edu.tw

Numerical Heat Transfer, Part B, 32:351–368, 1997
Copyright © 1997 Taylor & Francis
1040–7790/97 $12.00 + .00

NOMENCLATURE

a_{ij}	components of the matrix $\underline{\underline{A}}$	$P_i(t)$	Legendre polynomials
$\underline{\underline{A}}$	matrix	r	element number
e, f	element numbering	R	residual
E	a finite set of elements	u	x-component velocity
$\hat{i}, \hat{j}$	indices for rows and columns other than those at hanging nodes	u_{1C}, u_1, u_2	unknown variables at different nodes
$l(r)$	eccentricity of an element r	v	y-component velocity
$\mathscr{L}(r)$	level structure rooted at r	W_i (i = 1–4)	weighting functions
$\mathbf{M}$	matrix	μ	kinematic viscosity
n	dimension of the matrix $\underline{\underline{A}}$	Φ	working variable
$N_i(\xi, \eta)$	shape functions of a master element in coordinates ξ, η	$\hat{\Phi}$	computed solution of ϕ
		$\underline{\Phi}$	a vector that is packed with a column of working function ϕ in nodes
$\mathbf{P}$	permutation matrix		
$\mathrm{Pe}_x, \mathrm{Pe}_y$	Peclet number in the x and y directions, respectively	Φ_i	nodal values of ϕ at node i

efficiency and ease of programming also warrant consideration for a scheme to be termed robust. Achieving all these properties is, of course, very difficult. What fits for solution accuracy may not for solution stability. Retaining solution stability, without at the expense of compromising computational efficiency and prediction accuracy, constitutes the core of the present study.

We begin by describing in Section 2 the convection-diffusion equation. This is followed by a description of the finite-element employed. The M-matrix theory, which provides a basis for capturing sharp profiles in the flow, is briefly reviewed. The main attribute of the finite-element model adopted for the convection-diffusion equation is that use of Legendre polynomials to span finite-element spaces facilitates numerical integration. To broaden the application scope, we add grid adaptivity to the formulation in Section 3. For the sake of completeness, solution accuracy, scheme stability, and computational efficiency all must be considered together. To this end, we carry out a theoretical analysis and discuss the consequences in greater detail. In order to validate the proposed monotonic flux discretization scheme, we present a closed-form solution for the scalar transport equation defined in a simple domain. Attention is directed toward assessing the effectiveness of the h-adaptivity employed together with the reverse Cuthill-McKee (RCM) element-reordering technique.

THEORETICAL ANALYSIS

Model Equation

In this article we consider the transport equation for a passive scalar Φ given below:

$$u\Phi_x + v\Phi_y = \mu(\Phi_{xx} + \Phi_{yy}) \tag{1}$$

To simplify the analysis, a simple flow that involves constant velocities u and v is considered. In what follows, analysis is restricted to a constant-property flow with a prescribed value of diffusion coefficient μ. The elliptic nature of the partial differential equation (1) requires that boundary conditions be prescribed along the entire boundary of D.

Finite-element solutions, $\hat{\Phi}$, to the convection-dominant transport equation for Φ in (1) are obtained by demanding that $R = u\hat{\Phi}_x + v\hat{\Phi}_y - \mu(\hat{\Phi}_{xx} + \hat{\Phi}_{yy})$ be orthogonal to the weighting function. Solutions thus obtained are viewed as a search for weak solutions to Eq. (1). Depending on the circumstances, one can choose between centered and upwind schemes. Provided that the values of $\mathrm{Pe}_x = u\,\Delta x/\mu$ and $\mathrm{Pe}_y = v\,\Delta y/\mu$ exceed 2, the weighted-residuals formulation suffers stability problems. Upwinding techniques have proven to be the methods of choice to alleviate this problem. Here, Δx and Δy denote the mesh sizes along the x and y directions, respectively. To introduce the upwind effect into the formulation, we demand that the test space, W_i, be different from the trial space N_i in an ad hoc manner. The stabilization prevents the numerical oscillations. Substitution of the presently employed four-node isoparametric bilinear basis functions, $\hat{\Phi} = \Sigma_{i=1}^{4} N_i(\xi, \nu)\Phi_i$, where ξ and ν are called natural coordinates, into the weighted-residuals statement yields matrix equations for each element. This is followed by an assemblage of finite elements, thus forming the global coefficient matrix. To close the algebraic equations, there remains selection of test functions. Construction of a best-suited weighting function warrants further discussion.

Finite-Element Model

In the presence of steep gradients, flow-oriented flux discretization schemes no longer suffice for production of oscillation-free solutions. Attempts to suppress over- or under-shoots in the solution have led to the development of bounding schemes. In a domain of single dimension, monotonic solutions can be obtained by, among other ways, accommodating the total variational diminishing (TVD) property [1] or by applying flux limiters [2]. Extension of these filtering techniques to multidimensional analyses, however, lacks a sound theoretical foundation. The flux corrected transport (FCT) algorithm of Boris and Book [3], which was later generalized by Zalesak [4], is regarded as the first truly multidimensional shock-capturing scheme so far developed. In this article, we do not intend to justify whether there exists one scheme that outperforms the others. Instead, our aim is to extend the application scope so that the monotonic finite-element model [5] is applicable to adaptive meshes.

Since the **M**-matrix theory serves as the guideline for constructing our monotonic scheme, it is instructive to present here some useful definitions and theorems [6–8].

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

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

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

Definition 3. *A real n × n matrix* $\underline{\underline{A}}$ *is defined as monotone if* $\underline{\underline{A}}\underline{\phi} \geq \underline{0}$ *holds for any vector* $\underline{\phi}$ *under the circumstances* $\underline{\Phi} \geq 0$.

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

Recognizing that matrix equation taking an M-matrix form is a key to permitting bounded solutions, we are guided to construct weighting functions in favor of the upwind side [5]:

$$W_i = D_i\left[d_{\xi 0}P_0(\xi) + d_{\xi 1}P_1(\xi)\right]\left[d_{\eta 0}P_0(\eta) + d_{\eta 1}P_1(\eta)\right] \tag{2}$$

Specific to our Petrov-Galerkin finite-element analysis is the introduction of Legendre polynomials $P_0(t) = 1$ and $P_1(t) = t$ to the analysis. The coefficients shown in Eq. (2) are summarized as follows [5]:

$$D_i = \frac{1}{4}\exp\left(\frac{uh_\xi \xi_i}{2_\mu}\right)\exp\left(\frac{vh_\eta \eta_i}{2_\mu}\right)$$

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

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

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

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

where

$$W_\xi(\xi) = (1 + \xi_i\xi)\exp\left(-\frac{uh_\xi \xi}{2\mu}\right)$$

$$W_\eta(\eta) = (1 + \eta_i\eta)\exp\left(-\frac{vh_\eta \eta}{2\mu}\right)$$

In Eq. (2), h_ξ and h_η denote grid sizes.

The weighting functions given in Eq. (2) allow higher-order differentiation. The increased smoothness in the weighting function, however, leads to a marked demand on the number of Gaussian integration points required to yield an analytic integration. To alleviate this problem, we make use of the following orthogonal

Downloaded by [China Science & Technology University] at 02:41 26 February 2014

property:

$$\int_{-1}^{+1} P_i(t) P_j(t)\, dt = \frac{2}{2i+1}\delta_{ij} \qquad (i \text{ is dummy index}) \tag{3}$$

Thanks to the above integral identity, the number of Gaussian integration points needed for an analytic integration is dramatically reduced, which causes the CPU time to decrease substantially. For this reason, we are prompted to rewrite the bilinear shape functions $N_i(\xi,\eta)$ as functions of Legendre polynomials:

$$N_i(\xi,\eta) = \tfrac{1}{4}[P_0(\xi) + \xi_i P_1(\xi)][P_0(\eta) + \eta_i P_1(\eta)] \tag{4}$$

Fundamental Studies on the Finite-Element Model

Question arises as to whether the finite-element model formulated within the Legendre polynomial finite-element spaces can yield monotonic solutions unconditionally. To answer this question, we derive the discrete finite-element equation, $\Sigma_1^9 a_i \Phi_i = 0$, for a point at $j = 5$ in Figure 1, from four bilinear elements. Although obtaining functional expressions of a_i is quite involved, it is necessary to conduct this derivation to ascertain whether the stencil falls into the M-matrix category. In so doing, we have calculated a_i ($i = 1$–9) against Pe_x and Pe_y. The shaded area in Figure 2 is regarded as a stable domain since, by definition, matrix equations obtained under the circumstances are classified as irreducible diagonally dominant.

Even though the test functions chosen here can yield an M-matrix equation, this finite-element analysis is applicable only to a rather limited range of Peclet numbers. Application of this model to larger grid sizes is thus prohibited. Acknowledging this restriction, effort is dedicated to improving the structured-type Legendre polynomial model. We can, of course, continuously refine the mesh until the Peclet numbers fall within the stable region shown in Figure 2. Such refinement, however, is accompanied by a prohibitively high computational cost, which limits the extension of this theoretically appealing model to practical flow simulations. A plausible remedy for this difficulty is to adopt solution adaptivity in an attempt to reduce the local Peclet number.

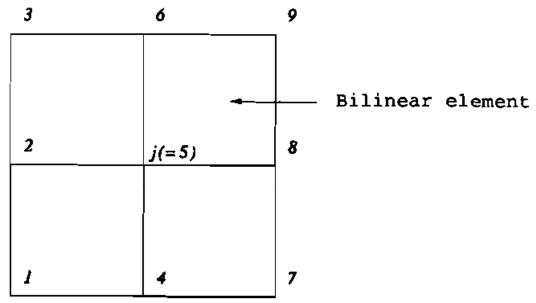


Figure 1. Illustration of nodal points for a pack of four bilinear elements.

Downloaded by [China Science & Technology University] at 02:44 26 February 2014

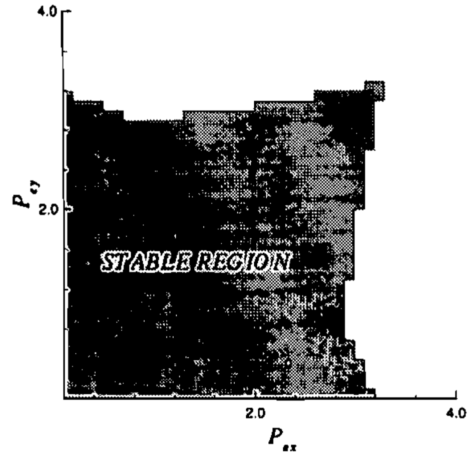


Figure 2. Stable (monotonic) region plotted in terms of the Peclet numbers, $\mathrm{Pe}_x = u\,\Delta x/\mu$ and $\mathrm{Pe}_y = v\,\Delta y/\mu$.

EXTENSION TO THE UNSTRUCTURED FINITE-ELEMENT MODEL

Solution Adaptivity

As noted earlier, monotonic solutions are quite expensive to compute at structured grids using the Legendre polynomial finite-element model [5] given in Section 2.2. Improvement needs to be made by reducing the computational cost and also the disk storage demand. Our strategy is to apply the Legendre polynomial finite-element model in a sufficiently refined domain in a sense that the resulting Peclet number falls into the monotonic region. We also refine the mesh in regions where solution gradients are considerable. This necessitates the introduction of grid adaptivity to the analysis so as to monotonically capture the local high-gradient profile in the solution. According to the M-matrix theory, the decision about which cells need to be divided and which undivided depends mainly on the cell Reynolds number (or Peclet number) as well as on the gradient of the solution. As a result, our grid adaption is a dynamic grid adaption.

Referring to Figure 3, nodal points are regarded as being regular if they are located at the corner vertices of each neighboring element [9]. Nodes other than regular ones in the mesh are classified as irregular (or hanging nodes). Numerical solutions at irregular nodes are not computed but are rather constrained by the solutions at regular nodes.

Depending on the mesh refinement strategy, the constrained relations vary. In this article, we adopt bisecting of the element. Use of this strategy to refine the mesh results in a hanging node located at the midpoint of a side of the coarser mesh. Since the numerical solution in an element is approximated by the bilinear shape function, the solution at the hanging node, say as 1C in the shaded element of Figure 3, is approximated by the following constraint relation:

$$u_{1C} = \tfrac{1}{2}(u_1 + u_2) \tag{5}$$

Downloaded by [China Science & Technology University] at 07:11 26 February 2014

Downloaded by [China Science & Technology University] at 02:41 26 February 2014

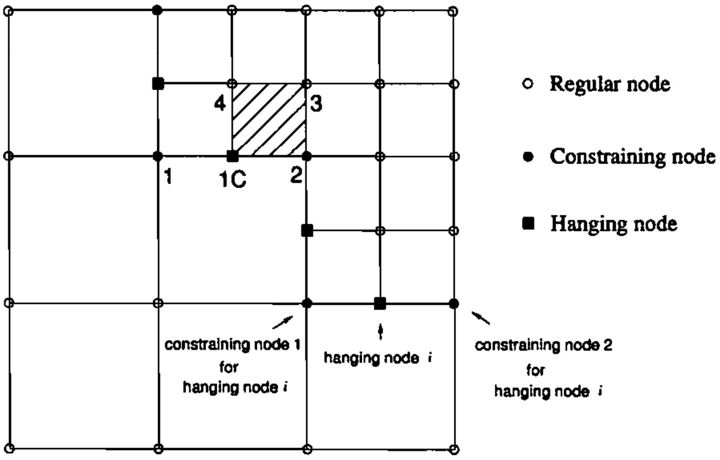

4
3
1
1C
2
○ Regular node
● Constraining node
■ Hanging node
constraining node 1 for hanging node *i*
hanging node *i*
constraining node 2 for hanging node *i*

Figure 3. Illustration of the hanging and constraining nodes.

Use of the above relation eliminates the degree of freedom of the nodal point 1C. The coefficient matrix for an element, $\mathbf{M}^e$, and the assembled matrix, $\mathbf{M}$, thus require modification.

Modification at the element level can be made through the operator form

$$\tilde{\mathbf{M}}^e = \mathbf{P}\mathbf{M}^e\mathbf{P}^T \tag{6}$$

by virtue of the introduction of the permutation matrix $\mathbf{P}$:

$$\mathbf{P} = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

This modification complicates the coding and may cause problems in vectorizing the code; therefore, it is not adopted in our code. Instead, we modify the mass matrix $\mathbf{M}$ directly, with the aim of improving the overall computation efficiency. Following the approach of Demkowicz et at. [10], for entries at hanging nodes, the rows and columns of the global mass matrix corresponding to the hanging node are eliminated and distributed to the rows and columns of the two corresponding constraining nodes by using

$$M_{k_r(i),\hat{j}} = M_{k_r(i),\hat{j}} + \tfrac{1}{2}M_{i,\hat{j}} \tag{8}$$

$$M_{\hat{i},l_s(j)} = M_{\hat{i},l_s(j)} + \tfrac{1}{2}M_{\hat{i},j} \tag{9}$$

$$M_{k_r(i),l_s(j)} = M_{k_r(i),l_s(j)} + \tfrac{1}{4}M_{i,j} \tag{10}$$

The subscripts i, j denote the indices of the rows and columns of the hanging node in the algebraic equation. As for $k_r(i), l_s(j)$ in Eqs. (8)–(10), they 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, and $\hat{i}, \hat{j}$ are indices for rows and columns other than those at the hanging node. With matrices thus modified, the vector **b** is changed accordingly as follows:

$$b_{k_r(i)} = b_{k_r(i)} + \tfrac{1}{2}b_i \tag{11}$$

In this way, row i and column j are eliminated. Removal of these hanging nodes leads us to obtain the u_i from Eq. (5) instead of obtaining them directly from the algebraic equation.

Element-Reordering Algorithm

For this study, we choose the frontal method [11] for solving the resulting sparse matrix equations. While the direct solution solvers are much more insensitive to the particular property of the matrix equation than their iterative counterparts, they are not immune to failure. For example, the application of the frontal method to sparse matrix equations involves large bandwidth. For larger problems, memory limitations make reducing the profile or wavefront of the matrix equations a necessity. In fact, this is a topic of fundamental importance in the finite-element community. Since the frontal scheme works in an element-by-element manner, reduction of the frontwidth has closer relevance to the element-ordering strategy than to the node-ordering strategy.

Several element-ordering algorithms in the literature can be chosen to reduce the envelop (or profile) of an assembled sparse matrix. Among others, the reverse Cuthill-McKee (RCM)[12], Gibbs-Poole-Stockmeyer (GPS) [13], and Gibbs-King (GK) [14] algorithms have gained wide popularity. These algorithms, falling into the profile-reduction ordering category, are closely related to one another. For this study, the ordering algorithm chosen is a variant of the original ordering algorithm of Cuthill and McKee [12], which is the most widely used profile-reduction ordering algorithm for the sparse symmetric matrix. Reduction of the bandwidth is accomplished through local minimization of the row width of the ith row and, thus, the profile of the matrix. George and Liu [15] found that it was advantageous to reverse the algorithmic ordering of the Cuthill-McKee. Reduction of the amount of envelope storage and the envelope operation count results. This implies that the RCM algorithm involves a local search of a pseudo-peripheral vertex to generate a long-rooted level structure. For the sake of brevity, the reverse Cuthill-McKee element-ordering algorithm will be described in the following [15].

Step 1. Find a starting element.

Step a. Choose an arbitrary element r in a finite set of elements E.

Downloaded by [China Science & Technology University] at 02:41 20 February 2014

Step b. Construct the level structure rooted at r:

$$\mathscr{L}(r) = \{L_0(r), L_1(r), \ldots, L_{l(r)}(r)\}$$

where $L_0(r) = \{r\}$, $L_1(r) = \text{Adj}(L_0(r))$, and $L_i(r) = \text{Adj}[L_{i-1}(r) - L_{i-2}(r)]$, $(i = 2, 3, \ldots)$. We denote here Adj$(\cdot)$ as an adjacent operator of "$\cdot$".

Step c. Choose an element e such that $L_{l(r)}(r)$ is of minimum degree.

Step d. (i) Construct the level structure rooted at e:

$$\mathscr{L}(e) = \{L_0(e); L_1(e), \ldots, L_{l(e)}(e)\}$$

(ii) If $l(e) > l(r)$, set r as e, and go to step c. Here, $l(r)$ is called the eccentricity of an element r.

Step e. Take element e as the root.

We then assign e_1 as r and proceed with the following two steps.

Step 2. Main loop: do Cuthill-McKee renumbering. For elements $i = 1, 2, \ldots, N$, find all the unnumbered neighbors of an element e_i, and number them in increasing order. Here, N is the element number.

Step 3. Reverse ordering: The resulting ordering if $f_1, f_2, \ldots, f_N$, where $f_i = e_{(N-i+1)}, i = 1, 2, \ldots, N$.

At this point, it is instructive to show the effectiveness gained using the RCM reordering technique. To this end, we consider in Figure 4 the unstructured-type mesh and plot in Figure 5 the sparse graph for the original element ordering for the unstructured element considered. Sparseness is a feature of this graph. The black dots ij plotted in Figure 5 indicate the finite element i that is adjacent to element j. The apparent sparseness creates large bandwidth in the matrix equations. This in turn causes the amount of computation to become formidable using conventional computers. Parallel to the above original element ordering, we plot in Figure 6 the structure of the reverse Cuthill-McKee reordering of the finite-element graph. The results clearly show that application of the RCM reordering yields a matrix representing the same algebraic system, but with a dramatically different frontal profile. As seen later in Section 4.2, the stored entries take up only 2.5% of the full matrix. With this marked reduction of bandwidth and in turn the savings in memory demand, we confirm the appropriate use of the RCM element-ordering technique as a model of simulating fluid flows using the finite-element method.

NUMERICAL STUDIES

Two test cases are chosen for validation and assessment purposes. These problems are known as the boundary-layer-type test problem, and the skew advection-diffusion problem.

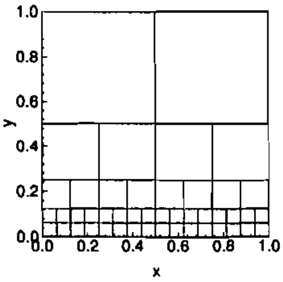


Figure 4. Illustration of h-adaptive meshes.

Analytic Study of a Boundary-Layer-Type Problem

The first problem is configured in Figure 7. Subject to the prescribed boundary data of Φ, the advection-diffusion equation is analytically amendable to the following boundary-layer-type solution:

$$\Phi(x, y) + \frac{\{1 - \exp[(x - 1)(u/\mu)]\}\{1 - \exp[(y - 1)(v/\mu)]\}}{[1 - \exp(-u/\mu)][1 - \exp(-v/\mu)]} \tag{12}$$

This problem is chosen to demonstrate the potential of the h-adaptive Legendre polynomial finite-element model to model problems involving a boundary-layer-like profile. Finite-element solutions are sought at the adaptive mesh shown in Figure 8 for the case of $\mu = 2 \times 10^{-3}$. We plot in Figure 9 the computed profiles at several selected planes. For comparison purposes, we also plot in this figure other solutions computed from uniform meshes and analytic solutions.

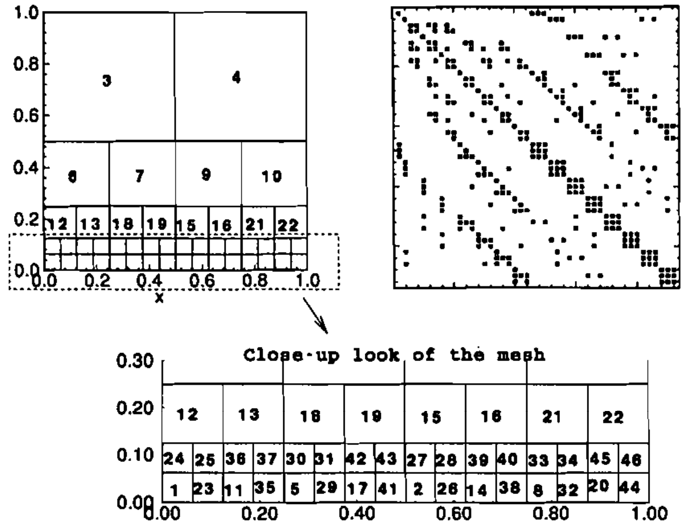


Figure 5. Illustration of original adaptive mesh used and its associated graph.

Downloaded by [China Science & Technology University] at 02:41 26 February 2014

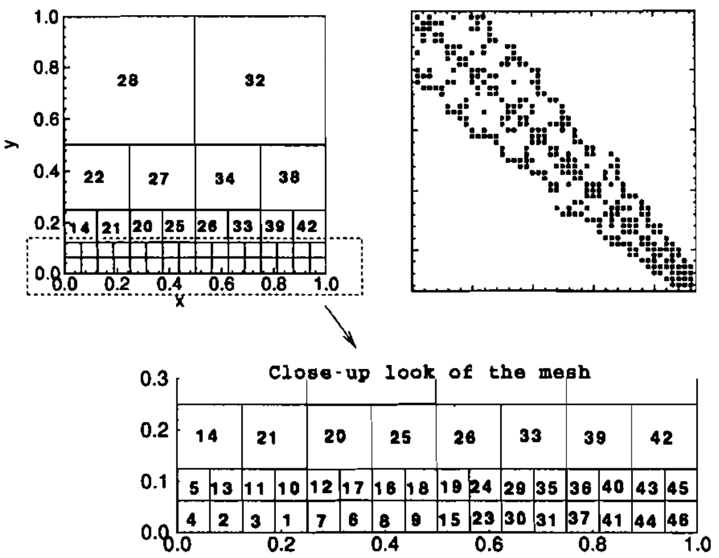


Figure 6. Illustration of the reordered adaptive mesh used and its associated graph.

Clearly revealed from Figure 9 is the usefulness of adding grid adaptivity to the formulation. Aside from the gain in monotonicity, the solution accuracy also improves with introduction of grid adaptivity. For completeness, calculations were performed for cases involving roughly the same number of regular mesh points. According to Table 1, we conclude that use of the present finite-element formulation gives us the capability to capture a rapid change of solution near the boundary of the flow.

Skew Advection-Diffusion Problem

We now consider an even more difficult problem, configured in Figure 10. The problem considered is that of the skewed flow transport problem, which is featured as having an internal layer and is regarded as a worst-case scenario for any upwinding method [16]. In a square cavity of unit length, through (0,0) there is

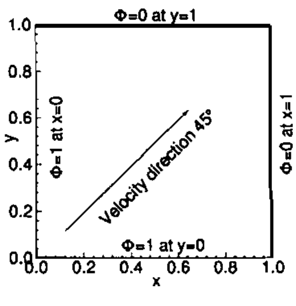


Figure 7. Configuration and boundary conditions of Φ for the problem defined in Section 4.1.

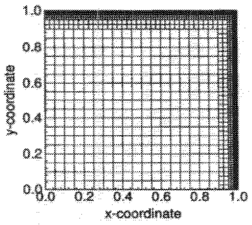
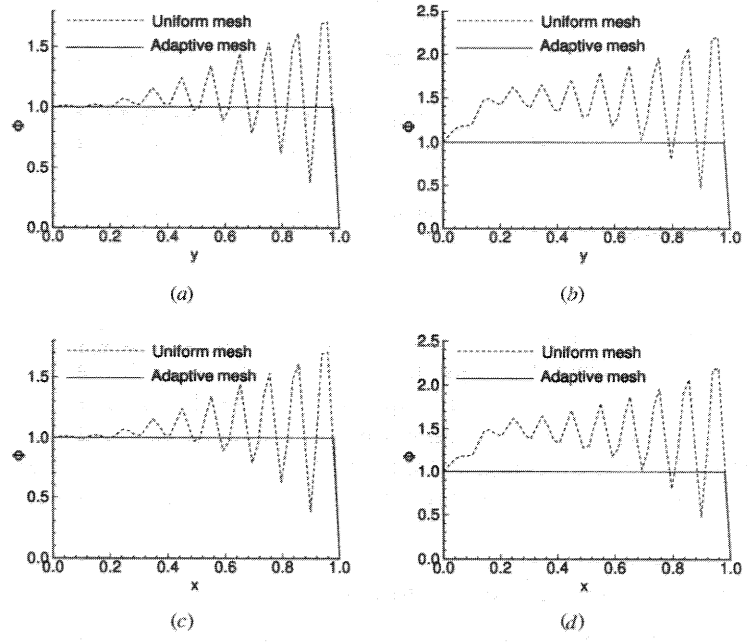


Figure 8. Mesh used to compute the problem defined in Section 4.1 (elements, 3,952; nodes, 4,593; slave nodes, 584).



(*a*) (*b*)

(*c*) (*d*)

Figure 9. A three-dimensional plot of solutions obtained in a mesh plotted in Figure 8 for the problem defined in Section 4.1: (*a*) $\Phi(0.25, y)$; (*b*) $\Phi(0.75, y)$; (*c*) $\Phi(x, 0.25)$; (*d*) $\Phi(x, 0.75)$.

Table 1. Computed L_2 error norms for the problem defined in Section 4.1

Grid type	Node number	L_2 error norm
Uniform grid	4,624	0.1197
h-Adaptive grid	4,593	3.415×10^{-2}

Downloaded by [China Science & Technology University] at 02:41 26 February 2014

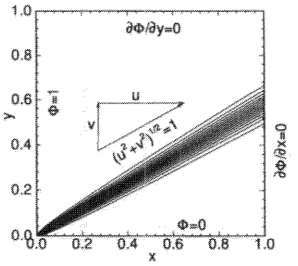


Figure 10. Configuration, together with the computed solution on 160 × 160 mesh, for the problem defined in Section 4.2.

a tilted line with an angle of $m = \tan^{-1} v/u$. This divides the square into two subdomains. In the whole domain, there is a uniform flow that is parallel to the dividing line. The magnitude of the velocity remains unchanged, with a magnitude of 1.

According to the diffusivity considered, $\mu = 2.0 \times 10^{-2}$, and the grid size chosen, $\Delta x = \Delta y = 0.05$, the Peclet numbers ($\mathrm{Pe}_x = 2.5, \mathrm{Pe}_y = 2.5$) fall into the monotonic region shown in Figure 2. As a result, oscillation-free finite-element solutions are observed in Figure 11 either in regions close to or in regions distant from the dividing line. Upon increasing the Peclet numbers or decreasing the diffusivity to $\mu = 10^{-3}$, oscillations become clearly seen in Figure 12. This example clearly demonstrates the drawback of applying the Legendre polynomial finite-element model to solve problems involving steep gradients. There may be two ways to suppress such wiggles. One may continuously reduce the mesh size until the maximum Peclet number falls into the monotonic region shown in Figure 2. This requires considerable computing cost and therefore makes practical computations nonfeasible. For this reason, we are motivated to refine the solution algorithm by adding an h-refinement technique to the finite-element analysis.

Since the skew-advection problem is not amendable to analytic solution, we take the solution computed at a 160 × 160 uniform mesh as a reference (Figure 13). For the case of $\mu = 10^{-3}$, we conduct the analysis in the adaptive mesh shown in Figure 14. As Figure 14 shows, mesh points are adaptively added in

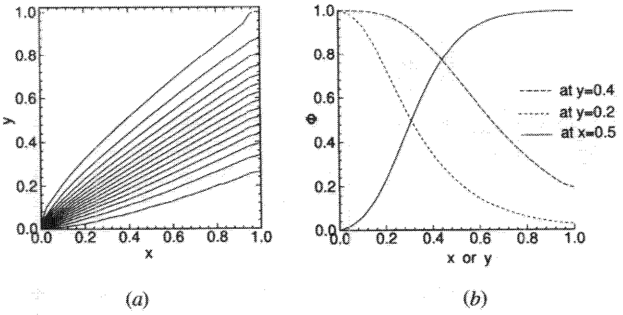
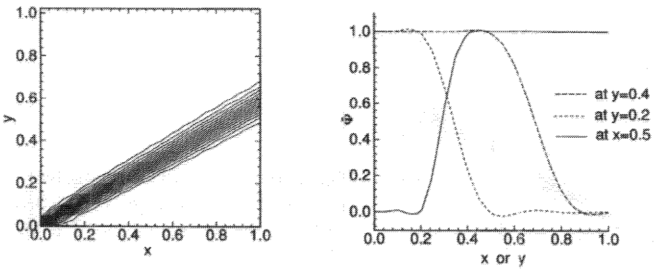


Figure 11. (a) Computed contours of Φ for the problem defined in Section 4.2 for the case $\mu = 2.0 \times 10^{-2}$. (b) Computed profiles of $\Phi(x, 0.4)$, $\Phi(x, 0.2)$, and $\Phi(0.5, y)$.

Downloaded by [China Science & Technology University] at 23:11 26 February 2014

Downloaded by [China Science & Technology University] at 02:41 20 February 2014



(*a*) (*b*)

Figure 12. (*a*) Computed contours of Φ for the problem defined in Section 4.2 for the case $\mu = 10^{-3}$. (*b*) Computed profiles of $\Phi(x, 0.4)$, $\Phi(x, 0.2)$, and $\Phi(0.5, y)$.

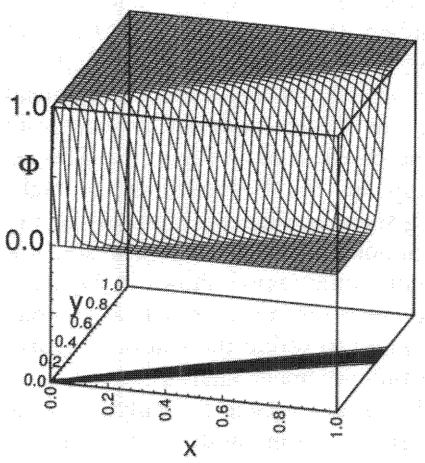


Figure 13. Computed solutions of Φ for the case, with a grid resolution of 160 × 160, defined in Section 4.2.

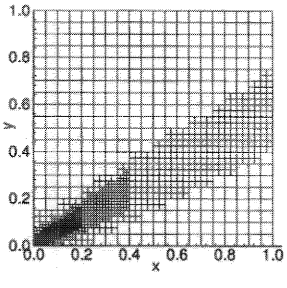


Figure 14. Dynamic h-adaptive grids employed (with 1,288 elements, 1,422 nodes, and 154 slave nodes) for the problem defined in Section 4.2.

regions where solution gradients of Φ are larger. Grids being adaptively meshed match well with the computed gradients $\nabla\Phi$ shown in Figure 15, which plots the contour values of $\nabla\Phi$ from the 160 × 160 solutions shown in Figure 13. Revealed by Figure 16 is that monotonic solutions are adaptively computable for the case of $\mu = 10^{-3}$, but with much fewer elements. This clearly demonstrates the algorith-

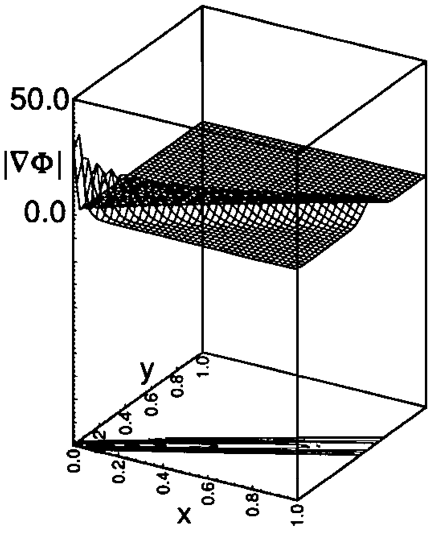
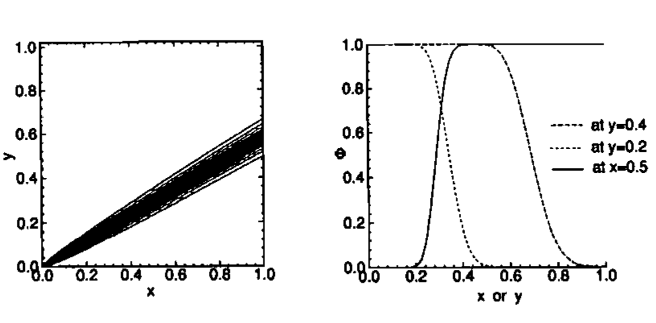


Figure 15. Computed gradients of Φ, |∇Φ|, from a grid resolution of 160 × 160 for the case defined in Section 4.2.

mic superiority of using the adaptive monotonic scheme as a way of resolving internal sharp layers.

By virtue of this test problem we will show the ability of the RCM element-reordering technique to improve the computational performance in the reduction of frontal width and CPU time (user as well as system time). In the first place, it is instructive to give readers an idea of the advantages gained in reducing the matrix bandwidth through use of the RCM reordering technique. To this end, we compare the structure of the reordered graph with the sparse graph structure of the original finite-element graph in Figure 17. This figure clearly shows that the RCM ordering algorithm reduces the envelope size and thus the bandwidth of the matrix. We summarize the aforementioned comparison items in Table 2 for an analysis using/without using the RCM reordering procedure. According to Table 2, which shows that both the CPU time and frontal width have been considerably reduced, we are benefiting greatly from the grid adaptivity and element renumbering.



(*a*) (*b*)

Figure 16. (*a*) Finite-element solutions of Φ for the problem defined in Section 4.2. Contours for solutions computed in the adaptive mesh. (*b*) Computed profiles of $\Phi(x, 0.4)$, $\Phi(x, 0.2)$, and $\Phi(0.5, y)$.

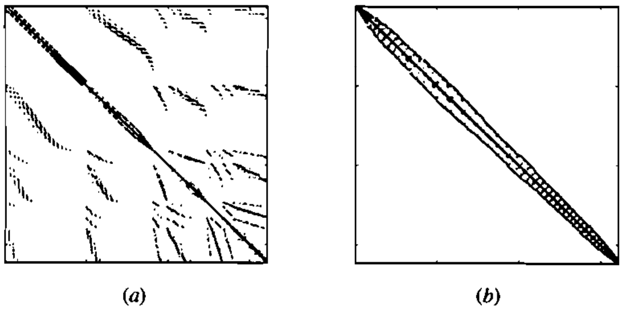


Figure 17. Graphs of the matrix equations: (*a*) graph of the original adaptive mesh. (*b*) graph of the reordered adaptive mesh.

It is now fair to conclude that we have extended our recently developed monotonic finite-element method, spanned by Legendre polynomials, with success to unstructured finite-element grids. It is believed that this adaptive finite-element model is more applicable to large-scale flow simulations.

CONCLUDING REMARKS

For this study, Legendre polynomials have been used to span finite-element basis and test spaces. There are several reasons for using Legendre polynomials, foremost of which is that this finite-element model can conditionally yield a monotone stiffness matrix. Second, with the inherent orthogonal property, much fewer Gaussian integration points are needed to render the exact integration. For a better understanding of this discretization scheme, a fundamental study of the discretization scheme has been conducted that provides insight into the estimation of monotonicity in the finite-element solution. We have used the discrete maximum principle as our guide in deciding whether or not the discretization scheme accommodates the monotone property. Through this study, we have learned that, if Peclet numbers exceed the critical Peclet number, then the discrete system can by

Table 2. Comparison of CPU time and required frontal width for a problem involving 5,370 adaptive nodes

	With element renumbering	Without element renumbering
Frontal width	125	5,046
User CPU time	1,308.98	2,426.49
System CPU time	10.62	25.55

no means be classified as an M-matrix; thus, monotonic solutions cannot be rendered. The motivation behind seeking an adaptive technique is that while monotonic solutions can be obtained in a continuously refined domain, the computational expense will be prohibitively high due to a severe restriction imposed on the grid size. Guided by the underlying discrete-maximum principle, we are able to determine which grid warrants further refinement so as to reduce the local Peclet number. Introduction of a h-adaptive capability to our upwind finite-element code makes analyses of convection-dominated and locally graded problems possible without loss of monotonicity near steep gradients. In an attempt to further improve the computational performance, we have incorporated the reversed Cuthill-McKee element-reordering technique into the finite-element analysis. This causes the frontal bandwidth to decrease dramatically. As a result, memory demand as well as computer time are also greatly reduced. Computational exercises reveal that the Legendre polynomial weighted-residuals formulation, supplemented with a h-adaptivity capability and RCM element-renumbering technique, gives solutions that are more accurate than solutions computed at uniform meshes.

REFERENCES

1. A. Harten, High Resolution Schemes for Hyperbolic Conversation Law, *J. Comput. Phys.*, vol. 49, pp. 357–393, 1983.
2. P. K. Sweby, High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws, *SIAM J. Numer. Anal.*, vol. 21, pp. 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.*, vol. 11, pp. 38–69, 1973.
4. S. T. Zalesak, Fully Multidimensional Flux-Corrected Transport Algorithm for Fluids, *J. Comput. Phys.*, vol. 31, pp. 335–362, 1979.
5. T. W. H. Sheu, S. F. Tsai, and M. M. T. Wang, A Monotone Finite Element Method with Test Space of Legendre Polynomials, *Comput. Meth. Appl. Mech. Engrg.*, vol. 143, pp. 349–372, 1997.
6. T. Meis and U. Marcowitz, Numerical Solution of Partial Differential Equations, in *Applied Mathematical Science*, Vol. 32, Springer-Verlag, New York, 1981.
7. T. Ikeda, Maximal Principle in Finite Element Models for Convection-Diffusion Phenomena, in *Lecture Notes in Numerical and Applied Analysis, Vol. 4*, North-Holland, Kinokuniya, Amsterdam, Tokyo, 1983.
8. M. Ahués and M. Telias, Petrov-Galerkin Scheme for the Steady State Convection Diffusion Equation, *Finite Elements in Water Resources*, no. 2/3, 1982.
9. L. Demkowicz and J. T. Oden, A Review of Local Refinement Techniques and the Corresponding Data Structures in h-Type Adaptive Finite Element Methods, TICOM Rep. 88-02, The Texas Institute for Computational Mechanics, The University of Texas at Austin.
10. 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. Meth. Appl. Mech. Eng.*, vol. 77, pp. 79–112, 1989.
11. B. M. Irons, A Frontal Solution Program, *Int. J. Numer. Meth. Eng.*, vol. 2, pp. 5–32, 1970.
12. E. H. Cuthill and J. McKee, Reducing the Bandwidth of Sparse Symmetric Matrices, *Proc. 24th Natl. Conf. Assoc. Comp. Mach.*, ACM Publications, pp. 157–172, 1969.

13. N. E. Gibbs, Jr., W. G. Poole, and P. K. Stockmeyer, An Algorithm for Reducing the Bandwidth and Profile of a Sparse Matrix, *SIAM, J. Numer. Anal.*, vol 13, no. 2, pp. 236–250, 1976.
14. N. E. Gibbs, A Hybrid Profile Reduction Algorithm, *ACM Trans. Math. Softw.*, vol.2, no. 4, pp. 378–387, 1976.
15. J. A. George and J. W. H. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981.
16. D. F. Griffiths and A. R. Mitchell, Finite Element for Convection Dominated Flows, in T. J. R. Hughes (ed.), *AMD* 34, pp. 91–104, ASME, New York, 1979.