JOURNAL OF COMPUTATIONAL PHYSICS 105, 108–121 (1993)

Non-oscillatory and Non-diffusive Solution of Convection Problems by the Iteratively Reweighted Least-Squares Finite Element Method

BO-NAN JIANG

ICOMP, NASA Lewis Research Center, Cleveland, Ohio 44135

Received October 16, 1990; revised October 11, 1991

In this paper we first introduce the least-squares finite element method (LSFEM) for two-dimensional steady-state pure convection problems with smooth solutions and compare the LSFEM with other finite element methods. We prove that the LSFEM has the same stability estimate as the original equation; i.e., the LSFEM has better control of the streamline derivative than the streamline upwinding Petrov–Galerkin method. Numerical convergence rates are given to show that the LSFEM is almost optimal. Then we use this LSFEM as a framework to develop an iteratively reweighted least-squares finite element method (IRLSFEM) to obtain non-oscillatory and non-diffusive solutions for problems with contact discontinuities. This new method produces a highly accurate numerical solution that has a sharp discontinuity in one element. A number of examples solved by using triangular and bilinear elements are presented to show that the method can convect contact discontinuities without error.

1. INTRODUCTION

In this paper we introduce and test numerically the iteratively reweighted least-squares finite element method for the solution of two-dimensional steady-state pure convection problems. This new method is designed to obtain accurate non-oscillatory discontinuous solutions.

We shall consider the following steady-state boundary value problem,

$$u_\beta = 0 \qquad \text{in } \Omega, \tag{1a}$$

$$u = g \qquad \text{on } \Gamma_- \tag{1b}$$

where Ω is a bounded convex domain in $\Re^2$ with boundary Γ, $u = u(x, y)$ is the dependent variable (e.g., the concentration), $\beta = (\beta_1, \beta_2)$ is a specified convection vector which may be constant or spatially varying, $u_\beta = \beta \cdot \nabla u$ denotes the derivative in β direction and g is the given data on the inflow boundary Γ_- defined by

$$\Gamma_- = \{(x, y) \in \Gamma : \mathbf{n}(x, y) \cdot \beta < 0\},$$

in which $\mathbf{n}$ is the outward unit normal to Γ at point $(x, y) \in \Gamma$. The problem (1) is purely hyperbolic.

For convenience of discussion, at first let us assume that the convection vector β is a constant. In this case, the characteristics of the problem (1) are straight lines parallel to β, and the analytic solution of problem (1) is constant along a characteristic. The value of this constant is equal to the given value of g at the intersection of this characteristic and the inflow boundary. Thus the solution will be discontinuous with a jump across a characteristic, if the boundary data g is discontinuous. As a simple 2D testing problem, this situation still represents a great challenge to researchers of numerical shock-capturing techniques.

Commonly used numerical methods for hyperbolic problems are of the following types (see, e.g., Fletcher [10], Hirsch [13], Johnson [21], and Pironneau [31]): method of characteristics, finite difference, and finite element methods. In principle the method of characteristics is very good, but it is rather cumbersome in practice. Usually one uses finite difference or finite element methods based on a mesh which may not be adapted to fit the characteristics of the particular problem. In such a case, if the exact solution has a jump discontinuity (contact discontinuity) across a characteristic, all conventional finite difference and finite element methods will produce approximate solutions which either oscillate or smear out a sharp front. Finding accurate approximations of the discontinuous solutions of hyperbolic equations has been a persistent difficult task in modern numerical mathematics and computational physics.

One research direction towards better resolution around discontinuities is to use an adaptive h-refinement or remeshing strategy, such as that extensively investigated by Oden and Demkowicz [29] and Peraire *et al.* [30]. However, the data structure and the programming of h-refinement are complicated, especially for three-dimensional problems.

Impressive resolution may be obtained by using filter methods [4, 23, 9]. For example, one may use any good

0021-9991/93 $5.00

finite difference schemes, such as the schemes developed by Deconinck *et al.* [7], to obtain an initial approximation and then use filter methods to sharpen the discontinuities.

Another potential way is to use conventional finite difference or finite element methods to obtain approximate solutions and then apply imaging processing techniques to detect the locations of discontinuities [32, 33].

All of the above-mentioned methods can only improve the resolution around the contact discontinuities. The L_1 procedure for non-oscillatory solutions, first proposed by Lavery in [24, 25] for one-dimensional problems, is a rather different approach. The L_1 solutions are highly accurate and right up to the edge of the discontinuity. Lavery uses linear programming to minimize the L_1 norm of the residuals of the overdetermined algebraic equations. Standard finite difference leads to determined linear algebraic systems. In order to obtain an overdetermined system Lavery relies on non-traditional tricks, such as gradually adding a small viscous term to one-dimensional Burgers' equation [24]. However, it is very difficult to extend these tricks to two-dimensional problems. Another difficulty in Lavery's L_1 procedure is that the linear programming algorithm of Barrodale and Roberts [1] is very expensive. This excludes the possibility of practical use of the L_1 method.

Because of the difficulties associated with the L_1 procedure in both theory and calculation, we completely avoid the application of the L_1 concept in this study. Our procedure for obtaining non-oscillatory solutions is based on the least-squares method. The idea can be explained as follows. In the usual least-squares curve fitting, the least-squares procedure does its best in a sense of least-squares of the residual to make the curve pass through or by **all** of the data. If the data are smooth, the least-squares fitting yields a very good approximation. However, if the data contain abrupt changes, the least-squares procedure will produce an oscillatory and diffusive curve around sharp changes. In such a case, the trouble comes from the fact the least-squares fitting makes the use of individual datum equally important. If we give up the outliers in the data and require the remaining data be satisfied exactly, the curve will be more smooth. The same thing happens in the least-squares solutions of discretized hyperbolic equations. The least-squares method treats equally the equations in "shocked" elements (in which the discretized scheme does not hold) and "smooth" elements. If we can identify the "shocked" elements and permit the equations in the "shocked" elements not to be satisfied while requiring that the remaining equations be satisfied exactly, then this modified least-squares solution will not have oscillation.

In this paper we first introduce the least-squares finite element method for two-dimensional steady-state pure convection problems with smooth solutions and compare it with other finite element methods. We prove that the least-squares method has the same stability estimate as the original equation; that is, the least-squares method has better control of the streamline derivative than the streamline upwinding Petrov–Galerkin method. Numerical convergence rates are given to show that the least-squares method is almost optimal.

Since the LSFEM produces a very good initial approximation to the exact solution, we use this information to find "shocked" elements. Then we use the least-squares method again. But this time we put a small weight for "shocked" elements to eliminate their "pollution," and repeat this procedure a few times until a convergent discontinuous solution is reached. This is our iteratively reweighted least-squares finite element method.

The reweighting must be based on an overdetermined system. Fortunately, it is trivial to have an overdetermined system in the least-squares finite element method. This can be explained as follows. The LSFEM with numerical quadrature is equivalent to a weighted collocation least-squares method [3], in which at first the residual equations are collocated at the interior points in each element and then the algebraic system is approximately solved by the weighted least-squares method. The Gaussian points for calculating the element matrices in the LSFEM correspond to the collocation points in collocation methods. If the order of Gaussian quadrature (or the number of quadrature points) is appropriately chosen, the least-squares finite element method amounts to solving an overdetermined system.

The arrangement of this paper is as follows. The least-squares method, its comparison with other methods, and the convergence tests for smooth problems are presented in Section 2. In Section 3 we describe the iteratively reweighted least-squares finite element method. The numerical results in Section 4 contain non-oscillatory solutions for problems with constant or spatially varying convection vector fields on uniform or unstructured meshes. In Section 5 we discuss the limitation of the method in the present implementation and the possibility of further improvement. Conclusions are drawn in Section 6.

2. THE LEAST-SQUARES FINITE ELEMENT METHOD

2.1. *Preliminaries and Notations*

As we have already shown in our previous papers (see [20] and the references therein), the LSFEM is a universal method for numerical solution of partial differential equations. It does not matter whether the partial differential equations are elliptic, parabolic, or hyperbolic. As long as the partial differential equation has a unique solution, the LSFEM always gives a reasonably good approximate solution. The work done in this paper is a natural extension of our least-squares method.

The problem (1) can be taken as a time-dependent problem, if we consider one space coordinate as a time-like coordinate. Then we may use the implicit time-marching least-squares finite element method introduced in [2] to obtain an approximate solution. The time marching is necessary for the Euler equations in aerodynamics [18, 19], since the Euler equations have nonunique solutions. The time-marching least-squares finite element method implicitly introduces an artificial dissipation to exclude the solutions with expansion shocks. For the linear hyperbolic problem (1), the time-marching is not necessary, because it has a unique solution, continuous or discontinuous. For this reason, we rather treat the problem (1) as two-dimensional and directly apply the LSFEM to attack it. The general formulation of least-squares finite element methods for first-order partial differential equations can be found in [20].

In order to see the advantages of the LSFEM and compare it with other finite element methods, let us consider the following linear hyperbolic equation:

$$u_\beta + u = f \qquad \text{in } \Omega, \tag{2a}$$

$$u = g \qquad \text{on } \Gamma_-, \tag{2b}$$

where f is a given source function. Without loss of generality we assume that the boundary data g is zero.

Throughout this paper, we use the following notations: $L_2(\Omega)$ denotes the space of square-integrable functions defined on Ω with the inner product

$$(u, v) = \int_\Omega uv \, d\Omega \qquad u, v \in L_2(\Omega),$$

and the norm

$$\|u\|^2 = (u, u), \qquad u \in L_2(\Omega).$$

$H^r(\Omega)$ denotes the Sobolev space of functions with square-integrable derivatives of order up to r; $\|\cdot\|_r$ denotes the usual norm for $H^r(\Omega)$. We also use the following notations:

$$\langle u, w \rangle = \int_\Gamma uw\mathbf{n} \cdot \beta \, ds,$$

$$\langle u, w \rangle_+ = \int_{\Gamma_+} uw\mathbf{n} \cdot \beta \, ds,$$

$$|u|_\Gamma = \left(\int_\Gamma u^2 \, |\mathbf{n} \cdot \beta| \, ds \right)^{1/2},$$

where

$$\Gamma_+ = \{(x, y) \in \Gamma : \mathbf{n}(x, y) \cdot \beta \geqslant 0\}.$$

We note that by Green's formula

$$(u_\beta, w) = \langle u, w \rangle - (u, w_\beta).$$

Further, we also define the function space

$$S = \{u \in H^1(\Omega) : u = 0 \text{ on } \Gamma_-\},$$

and the corresponding finite element subspace S_h; i.e., S_h is the space of continuous piecewise polynomial functions of order k. Here the parameter h represents the maximal diameter of the elements. By the finite element interpolation theory [6, 28] we have: Given a function $u \in H^{k+1}(\Omega)$, there exists an interpolant $\hat{u}^h \in S_h$ such that

$$\|u - \hat{u}^h\| \leqslant ch^{k+1} \|u\|_{k+1}, \tag{3a}$$

$$\|\nabla u - \nabla \hat{u}^h\| \leqslant ch^k \|u\|_{k+1}; \tag{3b}$$

here and below c denotes a contant independent of the mesh parameter h, with possibly different values in each appearance.

2.2. *The Standard Galerkin Method*

Now let us look at the following standard Galerkin method for the problem (2) (see, e.g., Johnson [21]). Find $u^h \in S_h$ such that

$$(u^h_\beta + u^h, w^h) = (f, w^h) \qquad \forall w^h \in S_h. \tag{4}$$

We define the error $e = u - u^h$. Then the error estimate for the standard Galerkin method is

$$\|e\| + |e|_\Gamma \leqslant ch^k \|u\|_{k+1}, \tag{5}$$

which is one order lower than that for elliptic and parabolic problems. Furthermore, in the continuous problem (2), we have the following stability estimate:

$$\|u\| + \|u_\beta\| + |u|_\Gamma \leqslant c \|f\|. \tag{6}$$

But in the standard Galerkin method the stability estimate is

$$\|u^h\| + |u^h|_\Gamma \leqslant c \|f\|, \tag{7}$$

which has no control of $\|u^h_\beta\|$.

2.3. *The SUPG Method*

In order to obtain better accuracy and stability, methods of upwinding type have appeared, see, e.g., papers by Dendy [8], Wahlbin [34], Christie *et al.* [5], Hughes and Brooks [15], Johnson *et al.* [22], Morton and Parrot [27]. Below

we shall look at the streamline upwinding Petrov–Galerkin method (SUPG) (Hughes [14]) or the streamline diffusion method (Johnson [21]): Find $u^h \in S_h$ such that

$$(u^h_\beta + u^h, w^h + hw^h_\beta) = (f, w^h + hw^h_\beta) \qquad \forall w^h \in S_h. \tag{8}$$

Johnson and his colleagues have derived the error estimate,

$$\left(\|e\|^2 + h \|e_\beta\|^2 + \frac{(1+h)}{2} |e|^2_\Gamma \right)^{1/2} \leqslant ch^{k+1/2} \|u\|_{k+1}, \tag{9}$$

which is near optimal. However, in the SUPG the corresponding stability estimate is

$$\|u^h\| + \sqrt{h} \|u^h_\beta\| + |u^h|_\Gamma \leqslant c \|f\|, \tag{10}$$

which means that the streamline derivative is less controlled than in the original problem. Another disadvantage of the SUPG in practical calculation is that the stiffness matrix is non-symmetric, which makes the solution of large-scale problems very difficult.

2.4. *The Least-Squares Method*

Now let us introduce the LSFEM. We assume that $f \in L_2(\Omega)$. For an arbitrary trial function $v \in S$, we define the residual function $R = v_\beta + v - f$. The least-squares method is based on minimizing the residual function in a least-squares sense. We construct the following quadratic functional:

$$\begin{aligned} I(v) &= \|R\|^2 = \|v_\beta + v - f\|^2 \\ &= (v_\beta + v - f, v_\beta + v - f). \end{aligned} \tag{11}$$

The least-squares method reads: Find $u \in S$ such that

$$I(u) \leqslant I(v) \qquad \forall v \in S.$$

Taking variation of I with respect to v and setting $\delta I = 0$ and $\delta v = w$, lead to the least-squares weak statement: Find $u \in S$ such that

$$b(u, w) = l(w) \qquad \forall w \in S, \tag{12}$$

where $b(u, w) = (u_\beta + u, w_\beta + w)$ and $l(w) = (f, w_\beta + w)$. The corresponding LSFEM has the following form: Find $u^h \in S_h$ such that

$$b(u^h, w^h) = l(w^h) \qquad \forall w^h \in S_h. \tag{13}$$

Let us now turn to the error estimate. Since we can replace w in (12) by w^h, we have

$$b(u, w^h) = l(w^h) \qquad \forall w^h \in S_h. \tag{14}$$

By subtracting (13) from (14) we obtain the following orthogonality for the error e:

$$b(e, w^h) = 0.$$

Let $\hat{u} \in S_h$ be the interpolant of u satisfying (3) and write $\rho = u - \hat{u}^h$ and $\theta = \hat{u}^h - u^h$ so that $e = \rho + \theta$. Then we have

$$\begin{aligned} \|e_\beta + e\|^2 &= b(e, e) = b(e, \rho) + b(e, \theta) = b(e, \rho) \\ &\leqslant \|e_\beta + e\| \, \|\rho_\beta + \rho\| \end{aligned}$$

or

$$\|e_\beta + e\| \leqslant \|\rho_\beta + \rho\| \leqslant \|\rho_\beta\| + \|\rho\|.$$

Recalling (3) we obtain the error estimate:

$$\|e_\beta + e\| \leqslant ch^k \|u\|_{k+1}. \tag{15}$$

Since the residual of approximate solution $R^h = u^h_\beta + u^h - f = e_\beta + e$, (15) is also the residual estimate,

$$\|R^h\| \leqslant ch^k \|u\|_{k+1}, \tag{16}$$

which means that the residual estimate is optimal. Using Green's formula and the boundary condition, (16) can be rewritten as

$$(\|e\|^2 + \|e_\beta\|^2 + \langle e, e \rangle_+)^{1/2} \leqslant ch^k \|u\|_{k+1}, \tag{17}$$

which shows that the error estimate for e_β is optimal, but the error estimate for e is one order lower than optimal. Although in numerical tests (see below) we have observed that the accuracy of the least-squares method is higher than the kth order, it is still an open question for getting a better theoretical error estimate, in general.

By taking $w^h = u^h$ in (13), we can obtain the stability estimate:

$$\|u^h\| + \|u^h_\beta\| + |u^h|_\Gamma \leqslant c \|f\|. \tag{18}$$

This estimate is the same as the above estimate (6) for the original problem (2). It means that the least-squares method has better control of the streamline derivative than the SUPG. We also note that the bilinear form in (13) is symmetric. Therefore, the matrix of the resulting algebraic system is symmetric and positive definite. This is a very important advantage of the least-squares method over other methods in practice.

2.5. *Numerical Experiments of the Least-Squares Method*

We chose the model problem,

$$\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = \sin(x + y) \qquad \text{in } \Omega, \tag{19a}$$

$$u = 0 \qquad \text{on } \Gamma_-, \tag{19b}$$

where $\Omega = \{(x, y) \in \Re^2 : 0 < x < 1,\ 0 < y < 1\}$ is the unit square, and $\Gamma_- = \{(x, y) \in \Gamma : x = 0 \text{ or } y = 0\}$, in which Γ is the boundary. This problem has a smooth exact solution $u = \sin(x)\sin(y)$.

We have tested the bilinear element on uniform meshes. At first the one-point Gaussian quadrature was used for calculating stiffness matrices. In this case, the least-squares method is equivalent to the collocation least-squares method with one collocation point at the center of each element. It is easy to check that in such a collocation method, the number of discretized algebraic equations "*Nequ*" is equal to the number of unknowns "$Nelem^2$," here "*Nelem*" is the number of elements. In other words, in such a case we solve a determined system. Therefore, there is no difference between the least-squares solution and the direct collocation solution. The numerical result for convergence rate is shown in Fig. 1. The optimal rate, i.e., $\|e\| \leqslant ch^2$, is observed.

Also we would like to mention that the least-squares method with one-point quadrature is equivalent to the central finite difference scheme. Therefore, the optimality of the LSFEM may be derived by using the finite difference theory.

The numerical rate of convergence with the 2×2 Gauss rule is also included in Fig. 1. In this case, the LSFEM solves an overdetermined system. The convergence rate is around $O(h^{1.75})$, which is near optimal. Here, more theoretical study is needed.

We also did the numerical tests with specified extra boundary conditions on the outflow boundary $\Gamma_+ = \{(x, y) \in \Gamma : x = 1 \text{ or } y = 1\}$. In this case, the least-squares method with the 2×2 Gauss rule gives the optimal rate of convergence $\|e\| \leqslant ch^2$ (Fig. 1).

From numerical experiments we may conclude that the LSFEM for the hyperbolic equation has an optimal or near optimal rate of convergence depending on the number of Gaussian points in calculation of element matrices. More Gaussian points yield slightly less optimal results, because the least-squares method is not able to make the residual of each equation in the overdetermined system equal to zero.

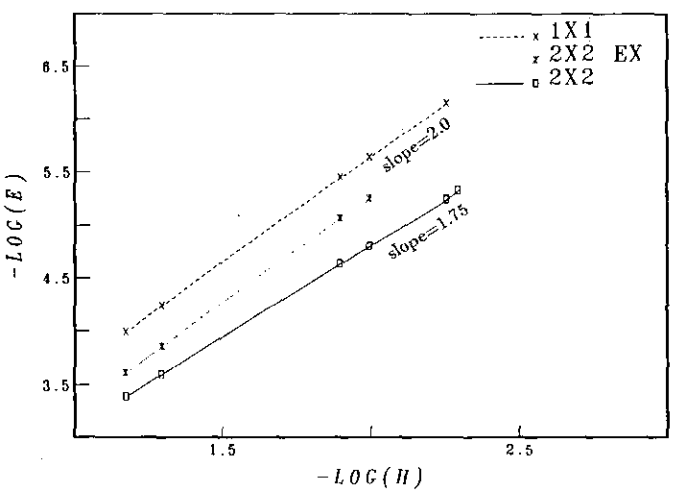

FIG. 1. Computed convergence rate for the pure convection problem.

3. THE ITERATIVE REWEIGHTED LEAST-SQUARES FINITE ELEMENT METHOD

If the exact solution is discontinuous, the above least-squares method still performs quite well. Of course, the argument about the error estimates does not hold. As expected, the least-squares method smears out the jump across a characteristic. As we pointed out in Section 1, the trouble comes from "shocked" elements, where the direction derivative across the jump approaches infinity and the discretized equation is not valid. But the usual least-squares method does not recognize them and just treats equally "shocked" and "smooth" elements. This is the reason that we would like to use the reweighting to suppress the interference of "shocked" elements.

Let us consider the problem (1). Our iteratively reweighted least-squares method is based on repeatedly solving a weighted least-squares problem: Find the minimizer $\bar{u}^h$ of

$$I(u^h) = \sum_{j=1}^{Nelem} \left(\sum_{l=1}^{Ngaus} W_l w_l (R_l)^2 \, |J(\xi_l, \eta_l)| \right)_j, \tag{20}$$

in which

$$W_l = \frac{1}{|R_l|^6_{\text{previous}}} \tag{21}$$

and

$$R_l = u^h_\beta(\xi_l, \eta_l) - f(\xi_l, \eta_l), \tag{22}$$

where W_l denotes the assigned weight, and R_l stands for the residual at each Gaussian point, *Ngaus* denotes the number of Gaussian points, w_l is the weight of Gaussian quadrature, $|J|$ is the determinant of the Jacobian matrix, and (ξ_l, η_l) are the local coordinates of Gaussian points. As usual, u^h can be expressed as

$$u^h(\xi, \eta) = \sum_{m=1}^{Nnode} \Psi_m(\xi, \eta)\, U_m, \tag{23}$$

where "*Nnode*" is the number of nodes in an element, Ψ_m denotes the shape function, and U_m is the nodal value. In order to make the problem (20) meaningful, we must have an overdetermined algebraic system. It can be realized simply by appropriately choosing the number of Gaussian points.

The IRLSFEM would begin with the initial weight

$W_l = 1$. This first step is nothing but the least-squares method introduced in Section 2. The result then determines a new set of weights by (21). In the second iteration, the residual $|R_l|$ is larger in "shocked" elements. Thus the weight W_l for "shocked" elements is smaller, and their inference becomes less important. This procedure is repeated until $\|\bar{u}^h_{\text{current}} - \bar{u}^h_{\text{previous}}\|$ is small. Our numerical experiments reveal that the difference between the residuals of "shocked" elements and those of their neighboring elements in the first least-squares solution is not significant enough. We put the sixth power in (21) in order to additionally increase the importance of "smooth" elements and reduce the contamination of "shocked" elements, and also accelerate the convergence. This is reasonable, since we want to completely eliminate the equations, which will have nonzero residuals, from the sytem. This trick is applicable, also because our non-weighted least-squares method is good enough to locate the "shocked" elements. That is, in the results of our least-squares method, the absolute value of the residuals in "shocked" elements is always greater than that in other elements.

We may further simplify the procedure by using another simple and reliable "shock" indicator—the variation of nodal values in each element—instead of the residual. The variation is defined as

$$V = \sum_{m=1}^{N_{node}} |U_m - U_{m-1}|, \qquad U_0 = U_{N_{node}}. \tag{24}$$

Therefore, the following weight is suggested:

$$W_l = \begin{cases} 10^{15} & \text{if } |V|^6_{\text{previous}} < 10^{-7}, \\ \dfrac{1}{|V|^6_{\text{previous}}} & \text{otherwise.} \end{cases} \tag{25}$$

Here some measures have been taken to prevent overflow. The advantage of using the variation as a "shock" indicator is as follows: Once the jump in the boundary data is given, we may know the exact values of the variation in "shocked" elements in advance. There are only a few possible values which depend on the type of finite element and are independent of the shape and size of the particular element and have no relation with the location of quadrature points.

The implementation of this reweighted least-squares method is really straightforward. If a least-squares finite element code is already available, it needs only a few additional line of FORTRAN statements.

4. NUMERICAL RESULTS

4.1. *Constant Convection Field*

As the first example, we consider the following problem with a constant convection vector,

$$\frac{\partial u}{\partial x} + (\tan 35°)\frac{\partial u}{\partial y} = 0 \qquad \text{in } \Omega, \tag{26a}$$

where $\Omega = \{(x, y) \in \Re^2 : 0 < x < 1,\ 0 < y < 1\}$ is the unit square with the boundary Γ. The inflow boundary conditions are

$$u = 2 \qquad \text{on} \quad \Gamma_1 = \{(x, y) \in \Gamma : x = 0\}, \tag{26b}$$

$$u = 1 \qquad \text{on} \quad \Gamma_2 = \{(x, y) \in \Gamma : x > 0 \text{ and } y = 0\}, \tag{26c}$$

Equations (26) represent uniform flow along straight lines inclined at an angle of 35° with respect to the x-axis. The jump discontinuity occurs along the line $y = x \tan 35°$. In this case, the analytic solution is

$u = 2$ on and above the line $y = x \tan 35°$,

$u = 1$ below the line $y = x \tan 35°$.

The boundary conditions (26b) and (26c) can be transferred into the source term in Eq. (26a). Therefore, the formulation of the least-squares method described in Section 2 can be applied to the problem (26).

Most of the computational results presented in this paper were obtained in double precision from a PC-386. A direct solver with variable band-width was used to obtain the solution of linear algebraic equations. The computing time will be significantly shortened by using the preconditioned conjugate gradient method [17], since the least-squares solution is already close to the accurate solution and final iterations are often just for correcting one or two nodal values which have not yet reached 15-digit accuracy.

4.1.1. *Linear triangular element.* Numerical experiments were carried out for the problem (26) using linear triangular elements on uniform meshes with $n = 5, 15$. Here n is the number of grids in each coordinate. For triangular elements, we use the one-point Gaussian quadrature. There are $2n^2$ elements, so we have $2n^2$ equations. Since there are $(n+1)^2$ nodal values and $(2n+1)$ boundary conditions, the number of unknowns is $(n+1)^2 - (2n+1) = n^2$. That is, the number of equations is double the number of unknowns. Therefore, the least-squares method amounts to solving an overdetermined system. It does not make sense to take more quadrature points, because in a linear triangular element $\partial u^h/\partial x$ and $\partial u^h/\partial y$ are constants, and thus the residuals at different points are the same.

The least-squares results for $n = 5$ (50 triangles) are listed in Table I. The numbers in Table I are the nodal values. Because the mesh is very coarse, the jump discontinuity is smeared severely. Starting from this bad least-squares solution, after four iterations of the IRLSFEM, we obtained the perfect discontinuous solution listed in Table II. This solution is correct to 15 digits. Here we should note that a

TABLE I

Nodal Values of LSFEM Solution for Constant Convection Problem (50 Triangles)

2.00000000000000	2.01915334035031	2.00878377705754	1.95795161549498	1.87078986840619	1.76004129143337
2.00000000000000	1.99603039945673	1.94399655185347	1.84950746061654	1.72946724328320	1.60187592334918
2.00000000000000	1.94806983617172	1.83312457760137	1.68739850887601	1.53694793617237	1.39560269306355
2.00000000000000	1.84015296868239	1.64119153265938	1.45765775017840	1.30434056518568	1.17833947348198
2.00000000000000	1.58961965405281	1.34285281744499	1.19086213790645	1.09256035611348	1.02302063798659
2.00000000000000	1.00000000000000	1.00000000000000	1.00000000000000	1.00000000000000	1.00000000000000

TABLE II

Nodal Values of IRLSFEM Solution for Constant Convection Problem (50 Triangles)

2.00000000000000	2.00000000000000	2.00000000000000	2.00000000000000	2.00000000000000	2.00000000000000
2.00000000000000	2.00000000000000	2.00000000000000	2.00000000000000	2.00000000000000	2.00000000000000
2.00000000000000	2.00000000000000	2.00000000000000	2.00000000000000	2.00000000000000	2.00000000000000
2.00000000000000	2.00000000000000	2.00000000000000	2.00000000000000	1.00000000000000	1.00000000000000
2.00000000000000	2.00000000000000	1.00000000000000	1.00000000000000	1.00000000000000	1.00000000000000
2.00000000000000	1.00000000000000	1.00000000000000	1.00000000000000	1.00000000000000	1.00000000000000

TABLE III

Nodal Values of LSFEM Solution for Constant Convection Problem (450 Triangles)

2.0000000	2.0000000	1.9999991	1.9999978	2.0000043	2.0000324	2.0000792	2.0000841	1.9999031	1.9993477	1.9983422	1.9972057	1.9969902	1.9997302	2.0084309	2.0266882
2.0000000	1.9999994	1.9999999	2.0000102	2.0000396	2.0000696	2.0000175	1.9997292	1.9990557	1.9980523	1.9972818	1.9981173	2.0028601	2.0144271	2.0352553	2.0577675
2.0000000	2.0000012	2.0000123	2.0000370	2.0000430	1.9999327	1.9995575	1.9988315	1.9979632	1.9977589	1.9998528	2.0066385	2.0206392	2.0430727	2.0619832	2.0526709
2.0000000	2.0000088	2.0000254	2.0000073	1.9998496	1.9994238	1.9987233	1.9981178	1.9986321	2.0020686	2.0107267	2.0264960	2.0492754	2.0613019	2.0320698	1.9367367
2.0000000	2.0000099	1.9999761	1.9997933	1.9993623	1.9987634	1.9985193	1.9998269	2.0045468	2.0147111	2.0314111	2.0532226	2.0550311	2.0019622	1.8760488	1.6882363
2.0000000	1.9999672	1.9997887	1.9993972	1.9989567	1.9991208	2.0011985	2.0069996	2.0181516	2.0348735	2.0545522	2.0426499	1.9620305	1.8058091	1.5997689	1.3859685
2.0000000	1.9998552	1.9995343	1.9992720	1.9998212	2.0025413	2.0090988	2.0206351	2.0365176	2.0531928	2.0236216	1.9117724	1.7266195	1.5086493	1.3035851	1.1447709
2.0000000	1.9997529	1.9996358	2.0004676	2.0036055	2.0105131	2.0218246	2.0361625	2.0493453	1.9971664	1.8504066	1.6391658	1.4167745	1.2273869	1.0940975	1.0191373
2.0000000	1.9999310	2.0008637	2.0041219	2.0109463	2.0215017	2.0338277	2.0434385	1.9619584	1.7768645	1.5444663	1.3264704	1.1592926	1.0532532	1.0007009	0.9846115
2.0000000	2.0007854	2.0038302	2.0101735	2.0195932	2.0297290	2.0360694	1.9157434	1.6898964	1.4442300	1.2406204	1.1012316	1.0226002	0.9895001	0.9835187	0.9887509
2.0000000	2.0025083	2.0080708	2.0161825	2.0242529	2.0279403	1.8548782	1.5884666	1.3413083	1.1627018	1.0549356	1.0020142	0.9845132	0.9852021	0.9916119	0.9970319
2.0000000	2.0046362	2.0115086	2.0179190	2.0197922	1.7736683	1.4724987	1.2401616	1.0965466	1.0215749	0.9907151	0.9843607	0.9885294	0.9945579	0.9985752	1.0001280
2.0000000	2.0059503	2.0113358	2.0123445	1.6634838	1.3441659	1.1471893	1.0458292	1.0013261	0.9871864	0.9874108	0.9924190	0.9970772	0.9996177	1.0003480	1.0002567
2.0000000	2.0051508	2.0062402	1.5115863	1.2102058	1.0706330	1.0131927	0.9930033	0.9892513	0.9919701	0.9959726	0.9988566	1.0001329	1.0003334	1.0001647	1.0000323
2.0000000	2.0020006	1.2993847	1.0856744	1.0194373	0.9989902	0.9938830	0.9943473	0.9965288	0.9985854	0.9997993	1.0002071	1.0001815	1.0000639	1.0000041	0.9999972
2.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000

TABLE IV

Nodal Values of IRLSFEM Solution for Constant Convection Problem (450 Triangles)

2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000
2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000
2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000
2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000
2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000
2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	1.0000000
2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	1.0000000	1.0000000	1.0000000
2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	1.0000000	1.0000000	1.0000000	1.0000000
2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
2.0000000	2.0000000	2.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
2.0000000	2.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
2.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000

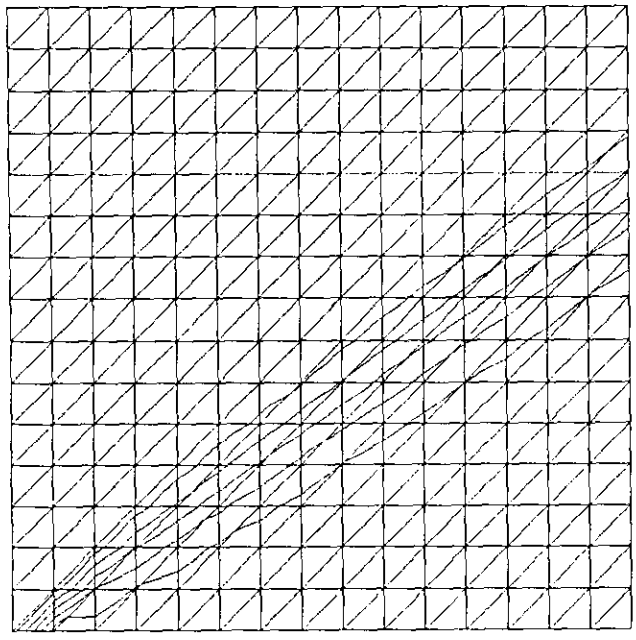

FIG. 2. Contours of LSFEM solution for constant convection problem (450 triangles).

double precision (8 bytes or 64 bits) real number in a computer can only represent a decimal number with 15 digits. This solution has no oscillation nor diffusion. The transition over the discontinuity is accurately located in the vicinity of the line $y = x \tan 35°$ and is accomplished in just one element.

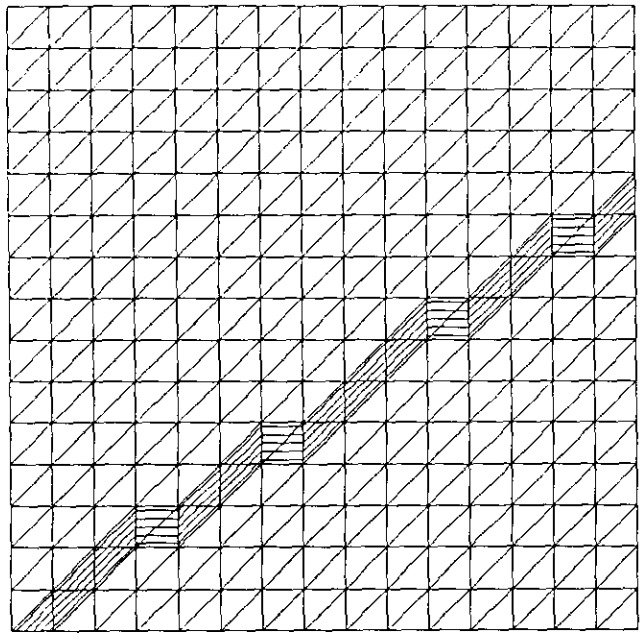

FIG. 3. Contours of IRLSFEM solution for constant convection problem (450 triangles).

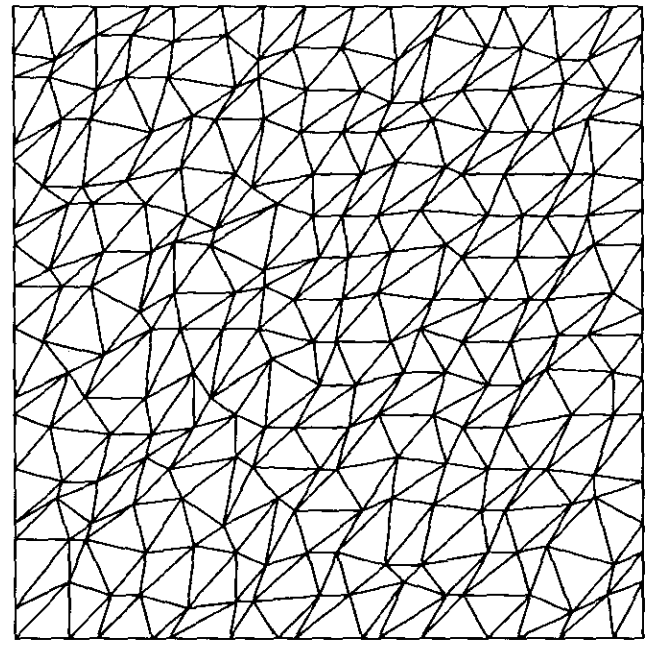

FIG. 4. Unstructured mesh with 450 triangles.

The least-squares solution for $n = 15$ (450 triangles) is given in Table III and depicted in Fig. 2. This solution is diffused and slightly oscillatory around the jump. Starting from this least-squares solution, the accurate discontinuous solution is obtained after four iterations (see Table IV and Fig. 3). This solution again has 15-digit accuracy. Because of the limitation of page size, we give the nodal values with only 8 digits in Table IV.

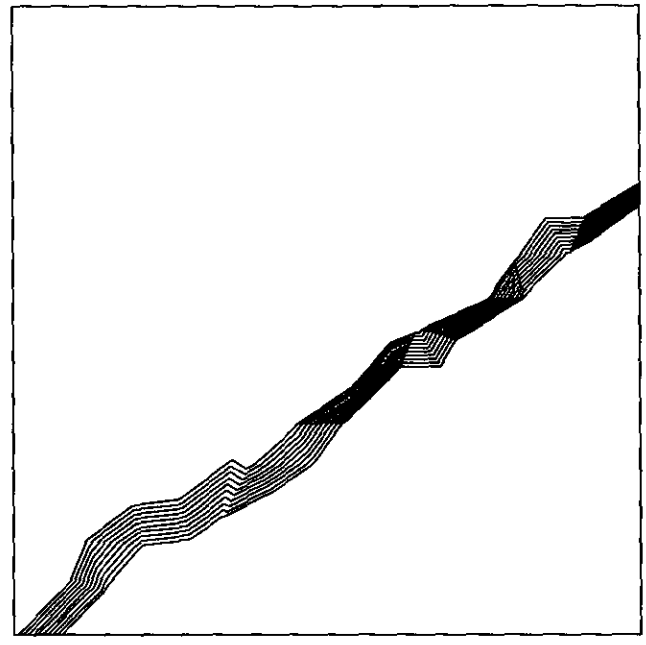

FIG. 5. Contours of IRLSFEM solution for constant convection problem on unstructured mesh.

TABLE V

Nodal Values of LSFEM Solution for Constant Convection Problem (5×5 Bilinear Elements)

2.00000000000000	2.00118821860769	2.01641923943302	2.03803285463046	2.02836549017373	1.94221664119897
2.00000000000000	2.01241750321268	2.02821882122379	2.00791191195100	1.90483206371286	1.72670330578520
2.00000000000000	2.01537346600061	1.99258728453429	1.86179903261828	1.62746979769939	1.37690702085225
2.00000000000000	1.98845751860405	1.82899110753578	1.51246025807770	1.25623579014445	1.07987201470759
2.00000000000000	1.83715624421649	1.34523292809047	1.12702336347438	1.02575100553365	0.98654845422136
2.00000000000000	1.00000000000000	1.00000000000000	1.00000000000000	1.00000000000000	1.00000000000000

TABLE VI

Nodal Values of IRLSFEM Solution for Constant Convection Problem (5×5 Bilinear Elements)

2.00000000000000	2.00000000000000	2.00000000000000	2.00000000000000	2.00000000000000	2.00000000000000
2.00000000000000	2.00000000000000	2.00000000000000	2.00000000000000	2.00000000000000	2.00000000000000
2.00000000000000	2.00000000000000	2.00000000000000	2.00000000000000	2.00000000000000	2.00000000000000
2.00000000000000	2.00000000000000	2.00000000000000	2.00000000000000	1.00000000000000	1.00000000000000
2.00000000000000	2.00000000000000	1.00000000000000	1.00000000000000	1.00000000000000	1.00000000000000
2.00000000000000	1.00000000000000	1.00000000000000	1.00000000000000	1.00000000000000	1.00000000000000

TABLE VII

Element Residuals of IRLSFEM Solution for Constant Convection Problem (5×5 Bilinear Elements)

0.18702E-14	0.17055E-15	0.49194E-14	0.48678E-14	0.22330E-14
0.92255E-15	0.49022E-14	0.35653E-14	0.11563E-15	0.18920E-14
0.18702E-14	0.18828E-14	0.59523E-16	0.20000E+01	0.20000E+01
0.85381E-15	0.20000E+01	0.20000E+01	0.20000E+01	0.14050E-14
0.20000E+01	0.20000E+01	0.49960E-15	0.64103E-16	0.49960E-15

TABLE VIII

Element Variations of IRLSFEM Solution for Constant Convection Problem (5×5 Bilinear Elements)

0.71054E-14	0.10658E-13	0.19540E-13	0.19540E-13	0.88818E-14
0.10658E-13	0.19540E-13	0.15987E-13	0.53291E-14	0.88818E-14
0.71054E-14	0.88818E-14	0.35527E-14	0.80000E+01	0.80000E+01
0.71054E-14	0.80000E+01	0.80000E+01	0.80000E+01	0.53291E-14
0.80000E+01	0.80000E+01	0.17764E-14	0.17764E-14	0.17764E-14

TABLE IX

Nodal Values of LSFEM Solution for Constant Convection Problem (15 × 15 Bilinear Elements)

2.0000000	2.0000212	2.0000235	1.9999043	1.9995902	1.9991208	1.9987157	1.9988191	2.0000744	2.0032143	2.0087875	2.0167032	2.0257342	2.0329932	2.0339627	2.0231248
2.0000000	1.9999904	1.9998523	1.9995395	1.9991398	1.9989339	1.9994092	2.0012133	2.0049874	2.0110995	2.0191630	2.0275467	2.0331377	2.0315441	2.0183976	1.9915600
2.0000000	1.9998790	1.9995999	1.9992896	1.9992720	2.0000576	2.0022737	2.0064850	2.0128797	2.0208777	2.0285851	2.0325802	2.0282442	2.0107099	1.9760764	1.9204989
2.0000000	1.9997621	1.9995272	1.9996492	2.0006521	2.0031274	2.0075757	2.0140821	2.0218686	2.0288300	2.0311260	2.0235360	2.0003933	1.9569290	1.8901454	1.7993285
2.0000000	1.9997900	1.9999678	2.0010622	2.0036448	2.0081435	2.0145886	2.0220904	2.0282379	2.0286770	2.0172917	1.9876125	1.9344352	1.8555985	1.7527483	1.6304775
2.0000000	2.0001224	2.0011670	2.0036893	2.0080903	2.0143273	2.0214609	2.0267985	2.0252756	2.0097012	1.9725557	1.9088083	1.8174699	1.7025820	1.5719411	1.4346342
2.0000000	2.0008544	2.0031610	2.0073092	2.0132509	2.0199634	2.0245242	2.0210091	2.0008813	1.9554102	1.8797042	1.7752395	1.6487816	1.5109898	1.3733297	1.2456297
2.0000000	2.0019629	2.0057279	2.0112725	2.0175946	2.0215171	2.0161906	1.9911834	1.9364385	1.8474582	1.7286379	1.5912454	1.4483964	1.3127210	1.1939395	1.0981475
2.0000000	2.0033085	2.0083867	2.0143503	2.0179346	2.0112377	1.9813466	1.9159006	1.8118530	1.6776949	1.5296183	1.3842675	1.2537882	1.1460899	1.0648495	1.0103898
2.0000000	2.0046162	2.0102761	2.0139136	2.0065678	1.9719395	1.8940468	1.7715333	1.6205645	1.4630452	1.3178925	1.1963829	1.1025525	1.0366001	0.9961249	0.9770358
2.0000000	2.0054601	2.0095997	2.0028918	1.9640549	1.8711947	1.7256691	1.5553954	1.3907720	1.2504091	1.1417529	1.0644335	1.0143365	0.9866239	0.9760920	0.9773165
2.0000000	2.0050206	2.0007867	1.9598332	1.8479140	1.6716377	1.4801908	1.3122381	1.1831069	1.0921987	1.0331987	0.9986778	0.9820493	0.9779457	0.9815875	0.9886704
2.0000000	2.0002628	1.9620044	1.8260667	1.6028546	1.3897235	1.2272944	1.1178980	1.0498305	1.0103198	0.9899060	0.9820555	0.9821432	0.9866074	0.9924748	0.9974802
2.0000000	1.9748483	1.8122040	1.5070281	1.2783532	1.1386935	1.0602254	1.0183334	0.9972563	0.9880863	0.9859612	0.9879052	0.9917679	0.9958585	0.9989563	1.0004876
2.0000000	1.8325134	1.3476253	1.1428679	1.0556664	1.0179156	1.0013554	0.9943825	0.9921568	0.9924889	0.9941953	0.9964107	0.9984429	0.9998264	1.0003956	1.0003061
2.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000

The method was further tested on a unstructured mesh with 450 non-uniform triangles shown in Fig. 4. After four reweighted iterations the discontinuity was correctly captured again in one element (see Fig. 5).

4.1.2. *Bilinear Element.* Numerical experiments were also carried out for the problem (26) using bilinear elements on uniform meshes with $n = 5$, 15, 40, and 100. For bilinear elements we use the 2×2 quadrature. In each element we may write the finite element approximation of u as a bilinear function:

$$u^h(x, y) = a + bx + cy + dxy.$$

Thus the residual is

$$R^h = \frac{\partial u^h}{\partial x} + \tan(35°)\frac{\partial u^h}{\partial y}$$
$$= b + \tan(35°)c + d(y + x\tan(35°)),$$

which means that the four discretized equations at four Gaussian points are independent in this case. (If the flow inclines at an angle of 45° or 135° with respect to the x-axis, we have three independent equations, because the location of Gaussian points is symmetric.) All together we have $4n^2$ equations and n^2 unknown nodal values. Therefore, we deal with an overdetermined system.

The least-squares solution on a coarse mesh with 5×5 bilinear elements is listed in Table V. Starting from this rough solution, after four iterations we obtained the discontinuous solution listed in Table VI. We again observe a crisp computational jump in one element. This solution is perfectly non-oscillatory and non-diffusive. We also list the summation of four absolute residuals and variations in each element in Table VII and Table VIII. The large numbers (two for residual and eight for variation) indicate the "shocked" elements.

The contours of the least-squares solution on a mesh with

TABLE X

Nodal Values of IRLSFEM Solution for Constant Convection Problem (15 × 15 Bilinear Elements)

2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000
2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000
2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000
2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000
2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000
2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000
2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	1.0000000	1.0000000
2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	1.0000000	1.0000000	1.0000000	1.0000000
2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
2.0000000	2.0000000	2.0000000	2.0000000	2.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
2.0000000	2.0000000	2.0000000	2.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
2.0000000	2.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
2.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000

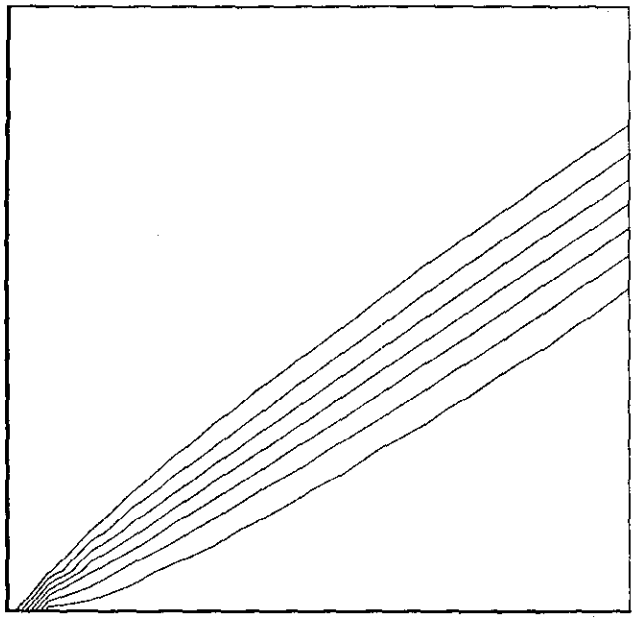

FIG. 6. Contours of LSFEM solution for constant convection problem (15×15 bilinear elements).

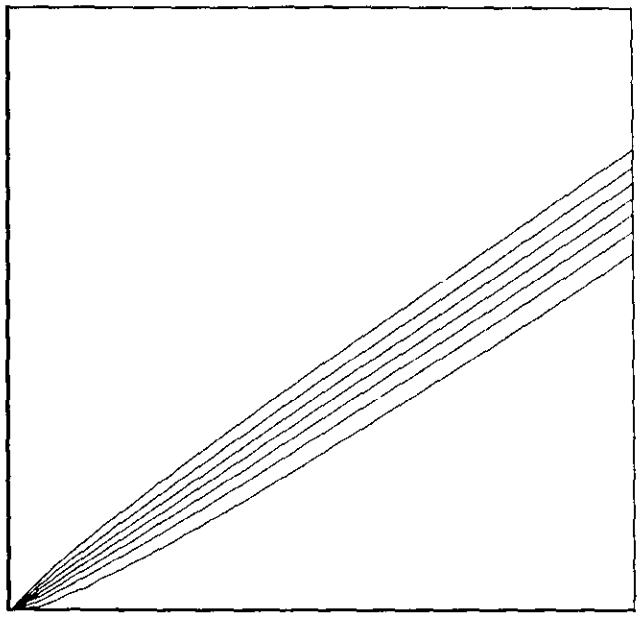

FIG. 8. Contours of LSFEM solution for constant convection problem (40×40 bilinear elements).

15×15 bilinear elements is presented in Fig. 6. This approximate solution is reasonably good, although the discontinuity is smeared out and slight oscillations occur. From this figure, we can hardly tell where the jump is located. However, after five iterations, a clean non-diffusive solution is reached (see Fig. 7). This solution again is 15-digit correct.

The least-squares solution on a mesh with 40×40 bilinear elements is illustrated in Fig. 8. Taking this initial least-squares solution, and after eight steps of processing, we obtained the highly accurate solution illustrated in Fig. 9.

We also did numerical tests for meshes with up to 100×100 elements, combined with various inflow angles and different boundary conditions. All of our IRLSFEM results are perfect. Because of the page limitation, we do not present these results here.

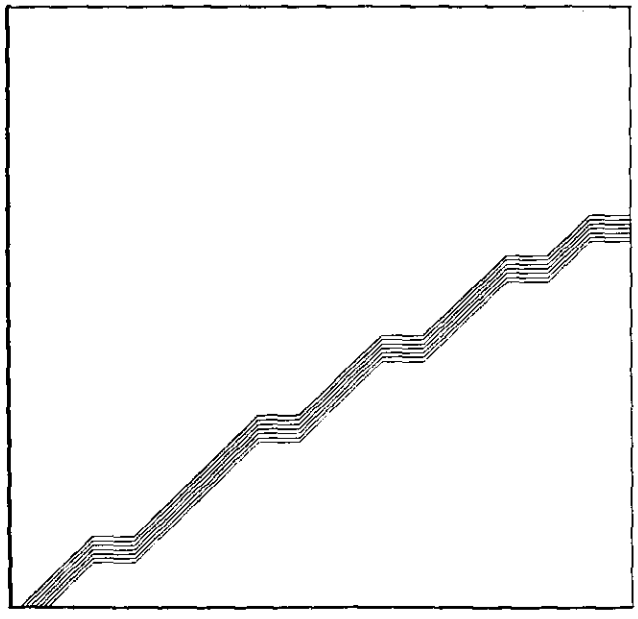

FIG. 7. Contours of IRLSFEM solution for constant convection problem (15×15 bilinear elements).

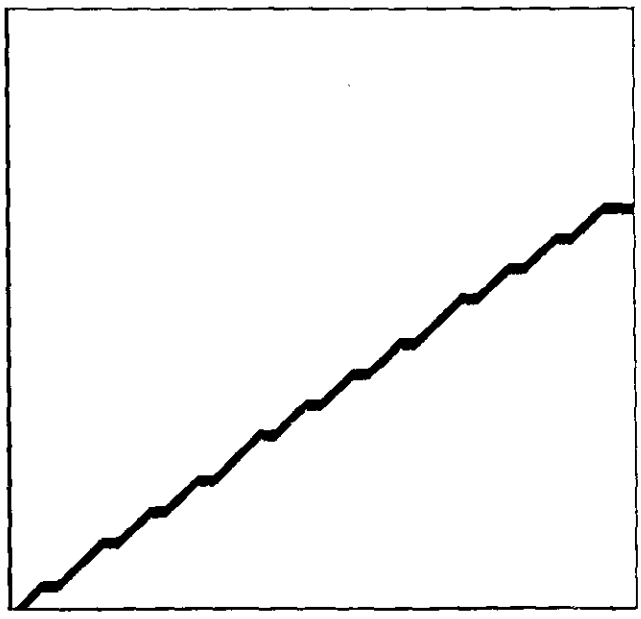

FIG. 9. Contours of IRLSFEM solution for constant convection problem (40×40 bilinear elements).

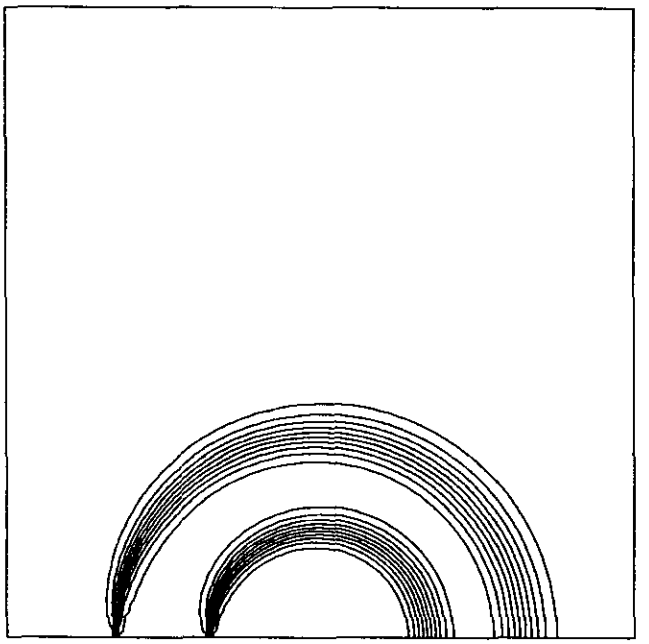

FIG. 10. Contours of LSFEM solution for circular convection problem (100 × 100 bilinear elements, 2 × 2 quadrature).

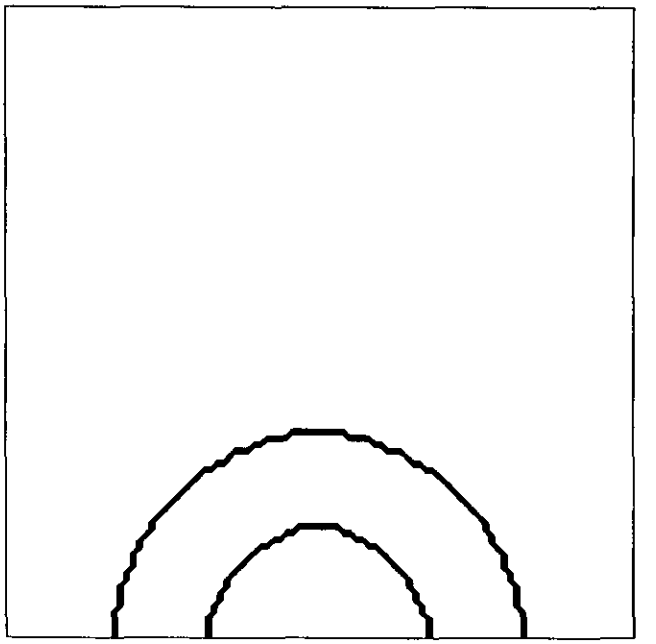

FIG. 12. Contours of IRLSFEM solution for circular convection problem (100 × 100 bilinear elements, 2 × 2 quadrature).

4.2. *Spatially Varying Convection Field*

Following Deconinck *et al.* [7], we considered the problem (1a) in the unit square with a circular convection field,

$$\beta_1 = y, \tag{27a}$$

$$\beta_2 = 0.5 - x, \tag{27b}$$

and boundary conditions,

$$u(0, y) = 0, \tag{27c}$$

$$u(x, 1) = 0, \quad x \geqslant 0.5 \tag{27d}$$

$$u(x, 0) = \begin{cases} 0 & x \leqslant 0.17; \\ 1 & 0.17 < x < 0.33; \\ 0 & 0.33 \geqslant x < 0.5. \end{cases} \tag{27e}$$

The mesh with 100×100 uniform bilinear elements was employed. The size of the elements in this mesh is 0.01

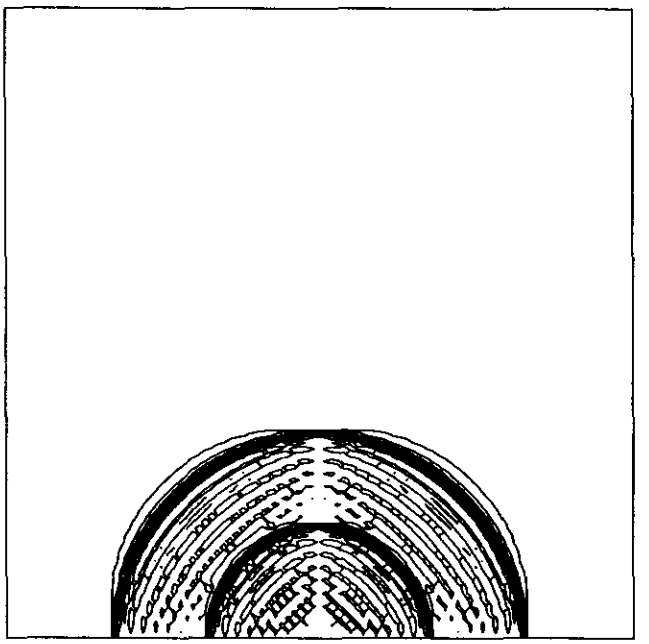

FIG. 11. Contours of LSFEM solution for circular convection problem (one-point quadrature).

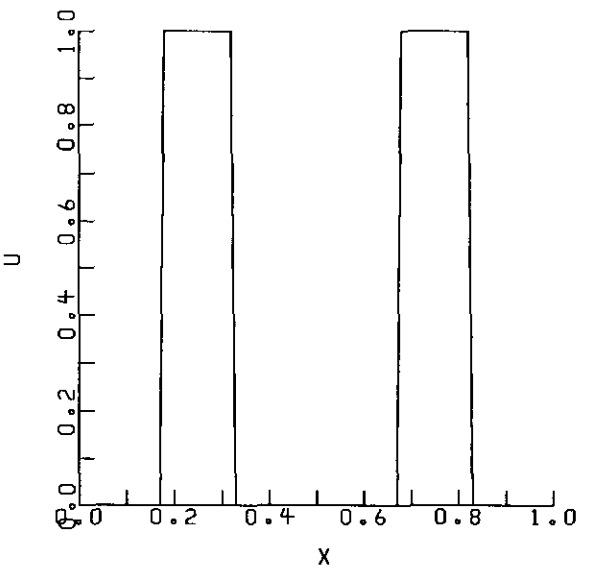

FIG. 13. Boundary distribution of u for circular convection problem.

(compare with 0.005 in the mesh used in [7]). Therefore, we modified the data a little bit in (27e) in order to make them consistent with the lower resolution of our mesh. In the least-squares finite element solution with 2×2 quadrature, as seen in Fig. 10, the input distribution (the left half of the lower boundary) is smeared quite a bit by the time it reaches the outflow (the right half of the lower boundary). Starting from this diffused solution, we still obtained a completely non-oscillatory and non-diffusive solution with correct jumps in just one element. However, the location of computed contact discontinuities on the outflow boundary has deviation of one or two grids from the exact solution. As indicated in Section 2, we may use the reduced integration (one point quadrature) to obtain a bette initial solution, which is shown in Fig. 11. This solution looks terrible but contains the correct information about the location of discontinuities. Starting from this oscillatory solution, after eight reweighted iterations (with 2×2 quadrature) we obtained a perfect 14-digit correct discontinuous solution shown in Fig. 12. The distribution of u on the bottom boundary of the domain is shown in Fig. 13. No other currently available methods can produce such a sharp and highly accurate resolution.

5. DISCUSSION

In the early version of this paper [16], the author mistakenly considered that the correct discontinuous solution is the minimizer of the L_1 norm of the residual, and thus called the IRLSFEM an L_1 method. Immediately after the publication of [16], Lowrie and Roe [26] reproduced the results and found that the accurate discontinuous solution does not minimize the L_1 norm of the residual. They further proved that for problem (26) the L_1 minimizer and the accurate discontinuous solution cannot be the same. This is an important discovery. It also implies that our method is totally new from concept to implementation.

Obviously, the present version of the iteratively reweighted least-squares finite element method cannot be used for problems with smooth solutions such as problem (19). This method will take the elements with larger gradients as "shocked" elements and completely ignore them and will then yield a wrong discontinuous solution.

If the given data g in problem (1) are varying along the boundary Γ_- with large gradient and a small jump, we will have the same difficulty. This type of difficulty is related to the numerical measure of discontinuity which is different from the mathematical definition of discontinuity. That is, we must distinguish the discontinuity from large gradients. This type of difficulty can be overcome by using fine meshes and specifying an appropriate level of jump for a particular problem. Only if the variation indicator of an element defined in (24) exceeds this level, should the contribution of this element be eliminated.

The choice of the sixth power in (21) is based on our limited experience. In fact, the method is not sensitive to the number of the power. The third, fourth, fifth, seventh, ... powers, all of them, work well. The only difference is that the number of iterations to convergence is different. Here the essential issue is how to completely eliminate the equations in "shocked" elements, and it does not matter what means are employed. Of course, a theoretic investigation to find an optimal way is very welcome.

6. CONCLUSIONS

A new procedure, based on the iteration of least-squares finite element method for the solution of pure convection problems with contact discontinuities, is developed. The overdetermined algebraic system is inherently obtained by choosing an appropriate number of Gaussian points in the formation of element matrices. Through reweighting the contamination of "shocked" elements is eliminated.

This finite element method captures two-dimensional discontinuity in bands of elements that are only one element wide on both coarse and fine meshes. The solution of this method has no smearing nor oscillation and has superior accuracy. The method is simple and robust. The concept can also be applied to three-dimensional problems.

We believe that the methodology developed in this paper can be transferred into many other areas which deal with sharp fronts such as oil reservoir simulation, weather forecast, pollution control, and image enhancement. We have already tested this method for two-dimensional compressible flows with shocks. The preliminary results are encouraging.

ACKNOWLEDGMENTS

The author appreciates the discussion with Professor S. H. Chang, Dr. J. E. Lavery, Dr. T. L. Lin, and Dr. W. G. Xue in the preparation of this paper. The author also acknowledges the comments of Professor P. L. Roe and Mr. R. B. Lowrie.

REFERENCES

1. I. Barrodale and F. D. K. Roberts, *SIAM J. Numer. Anal.* **10**, 839 (1973).
2. G. F. Carey and B. N. Jiang, *Int. J. Numer. Methods Eng.* **26**, 81 (1988).
3. G. F. Carey and J. T. Oden, *Finite Elements: A Second Course, Vol. II* (Prentice-Hall, Englewood Cliffs, NJ, 1986), p. 199.
4. M. Chapman, *J. Comput. Phys.* **44**, 84 (1981).
5. I. Christie, D. F. Griffiths, A. R. Michell, and O. C. Ziekiewicz, *Int. J. Numer. Methods Eng.* **10**, 1389 (1976).
6. P. G. Ciarlet, *The Finite Element Method for Elliptic Problems* (North-Holland, Amsterdam, 1978).
7. H. Deconinck, K. G. Powell, P. L. Roe, and R. Struijs, AIAA-91-1532-CP.
8. J. E. Dendy, *SIAM J. Numer. Anal.* **11**, 637 (1974).

9. B. Engquist, P. Lötstedt, and B. Sjögreen, *Math. Comput.* **52**, 509 (1989).
10. C. A. J. Fletcher, *Computational Techniques for Fluid Dynamics 1, 2* (Springer-Verlag, Berlin, 1988).
11. A. Harten, *J. Comput. Phys.* **49**, 357 (1983).
12. A. Harten, *SIAM J. Numer. Anal.* **21**, 1 (1984).
13. C. Hirsch, *Numerical Computation of Internal and External Flows I* (Wiley, Chichester, 1988).
14. T. J. R. Hughes, *Int. J. Numer. Methods Fluids* 7, 1261 (1987).
15. T. J. R. Hughes and A. Brooks, "A Multidimensionl Upwind Scheme with No Crosswind Diffusion," in *Finite Element Methods for Convection Dominated Flows*, edited by T. J. Hughes, AMD, Vol. 34, (ASME, New York, 1979).
16. B. N. Jiang, NASA TM 103773, ICOMP-91-03.
17. B. N. Jiang and G. F. Carey, "Element-by-Element Preconditioned Conjugate Gradient Algorithm for Compressible Flow," in *Innovative Methods for Nonlinear Problems*, edited by W. K. Liu, T. Belytschko, and K. C. Park (Pineridge Press, Swansea, UK, 1984).
18. B. N. Jiang and G. F. Carey, *Int. J. Numer. Methods Fluids* **8**, 933 (1988).
19. B. N. Jiang and G. F. Carey, *Int. J. Numer. Methods Fluids* **10**, 557 (1990).
20. B. N. Jiang and L. A. Povinelli, *Comput. Methods Appl. Mech. Eng.* **81**, 13 (1990).
21. C. Johnson, *Numerical Solution of Partial Differential Equations by the Finite Element Method* (Cambridge Univ. Press, Cambridge, UK, 1987).
22. C. Johnson, U. Nävert, and J. Pitkäranta, *Comput. Methods Appl. Mech. Eng.* **45**, 285 (1984).
23. S. Koshizuka, C. B. Carrico, S. W. Lomperski, Y. Oka, and Y. Togo, *Comput. Mech.* **6**, 65 (1990).
24. J. E. Lavery, *J. Comput. Phys.* **79**, 436 (1988).
25. J. E. Lavery, *SIAM J. Numer. Anal.* **26**, 1081 (1989).
26. R. B. Lowrie and P. L. Roe, "On the Numerical Solution of Conservation Laws by Minimizing Residuals," Conservation Laws and Shock Capturing, University of South Carolina, Sep. 13, 1991.
27. K. W. Morton and A. K. Parrot, *J. Comput. Phys.* **36**, 249 (1980).
28. J. T. Oden and G. F. Carey, *Finite Elements: Mathematical Aspects, Vol. IV* (Prentice–Hall, Englewood Cliffs, NJ, 1983).
29. J. T. Oden and L. Demkowicz, "Advances in adaptive improvements: A survey of adaptive methods in computational fluid mechanics," in *State of the Art Surveys in Computational Mechanics*, edited by A. K. Noor and J. T. Oden (ASME, New York, 1988).
30. J. Peraire, M. Vahdati, K. Morgan and O. C. Zienkiewicz, *J. Comput. Phys.* **72**, 449 (1987).
31. O. Pironneau, *Finite Element Method for Fluids* (Wiley, London, 1989).
32. E. V. Vorozhtsov, *Comput. Fluids* **15**, 13 (1987).
33. E. V. Vorozhtsov, *Comput. Fluids* **18**, 35 (1990).
34. L. B. Wahlbin, *RAIRO* **8**, 109 (1974).