NASA Technical Memorandum 103773
ICOMP-91-03

IN-64
13481
p28

The L_1 Finite Element Method for Pure Convection Problems

(NASA-TM-103773) THE L SUB 1 FINITE ELEMENT METHOD FOR PURE CONVECTION PROBLEMS (NASA) 28 p CSCL 12A

N91-24817

Unclas
G3/64 0013481

Bo-nan Jiang
Computational Physics System
Ann Arbor, Michigan

and Institute for Computational Mechanics in Propulsion
Lewis Research Center
Cleveland, Ohio

April 1991

NASA

THE L_1 FINITE ELEMENT METHOD FOR PURE CONVECTION PROBLEMS

Bo-nan Jiang

Computational Physics System, P.O.Box 4255
Ann Arbor, MI 48106

and

Institute for Computational Mechanics in Propulsion
Lewis Research Center
Cleveland, OH 44135

Summary

In this paper we first introduce the least-squares (L_2) finite element method for two-dimensional steady-state pure convection problems with smooth solutions. We prove that the L_2 method has the same stability estimate as the original equation, that is, the L_2 method has better control of the streamline derivative. Numerical convergence rates are given to show that the L_2 method is almost optimal. Then we use this L_2 method as a framework to develop an iteratively reweighted L_2 finite element method to obtain a least absolute residual (L_1) solution for problems with discontinuous solutions. This L_1 finite element method produces a non-oscillatory, non-diffusive and highly accurate numerical solution that has a sharp discontinuity in one element on both coarse and fine meshes. We also devise a robust reweigting strategy to obtain the L_1 solution in a few iterations. A number of examples solved by using triangle and bilinear elements are presented.

1. Introduction

In this paper we introduce and test numerically the L_1 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:

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

$$u = g \qquad on \quad \Gamma_-, \tag{1.b}$$

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 constant vector with $|\beta| = 1$, $u_\beta = \beta \cdot \nabla u$ 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. The characteristics of the problem (1) are the straight lines parallel to β. The analytic solution of problem (1) is very simple. The solution is a 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. However, the solution is discontinuous with a jump across a characteristic, if the boundary data g is discontinuous. This creates the greatest challenge to numerical solutions.

Commonly used numerical methods for hyperbolic problems are of the following types(see e.g., Fletcher[10], Hirsch[13], Johnson[20] and Pironneau[29]): 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 and finite element methods based on a mesh, which is not adapted to fit the characteristics of the particular problem. In such a case, if the exact solution has a jump 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 strategy, such as that extensively investigated by Oden and his colleagues[28]. However, the data structure and the programming of h-refinement are complicated, especially for three-dimensional problems.

Impressive sharp discontinuities may be obtained by using filter methods[5,22,9]. For example, one may use TVD-type finite difference schemes[11,12] to get a good approximation, then use filter methods to improve the resolution.

Another potential way is to use conventional finite difference or finite element methods to get approximate solutions, then apply imaging processing techniques to detect the locations of discontinuities[30,31].

The L_1 procedure for non-oscillatory solutions, first proposed by Lavery in [23,24], is a rather different approach. The L_1 idea can be explained as follows. In the usual L_2 curve fitting, the L_2 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 L_2 fitting leads to a very good approximation. However, if the data contain abrupt changes, the L_2 procedure will produce an oscillatory and diffusive curve around sharp changes. In such a case, the trouble comes from the fact that the L_2 fitting makes the use of individual *datum* equally important. The tendency of L_1 fitting is to give up the outliers in the data and to require the remaining data be satisfied exactly. Therefore, the L_1 fitting is the choice for discontinuous functions. The same thing happens in the L_2 and L_1 solutions of discretized hyperbolic equations. The L_1 capacity translates into a capacity to permit the equation in the "shocked" cell (in which the discretized scheme does not hold) not to be satisfied while requiring that the remaining equations be satisfied exactly. The L_1 solutions are non-oscillatory, highly accurate and right up to the edge of the discontinuity.

The key point of L_1 procedure is how to get overdetermined discretized systems. The standard finite difference and finite volume methods lead to determined linear systems. In order to get an overdetermined linear algebraic systems, one must rely on non-traditional tricks, such as, artificially adding the extra boundary conditions on the outflow boundary for two-dimensional pure convection problems[25], or gradually adding a smaller viscous term for one-dimensional Burgers' equation[23]. Another big difficulty associated with the usual L_1 procedure is that the linear programming algorithm of Barrodale and Roberts[1] is very expensive. It requires at least $O(n^4)$ and perhaps as many as $O(n^6)$ operations, here n is the number of grids in each axis. This excludes the possibility of practical use of the usual L_1 method.

In this paper we first introduce the least-squares (L_2) finite element method for two-dimensional steady-state pure convection problems with smooth solutions. We prove that the L_2 method has the same stability estimate as the original equation, that is, the L_2 method has better control of the streamline derivative. Numerical convergence rates are given to show that the L_2 method is almost optimal. Then we use this L_2 method as a framework to develop an L_1 finite element method for the solution of hyperbolic equations. The L_2 finite element method with numerical quadrature is equivalent to a weighted collocation least-squares method[4], in which at first the residual equations are collocated at the interior points in each element, then the algebraic system is approximately solved by the weighted least-squares method. The Gaussian points for calculating the element matrices in the L_2 finite element method correspond to the collocation points in collocation methods. If the order of Gaussian quadrature (or the number of quadrature points) is appropriately chosen, the L_2 finite element method amounts to solving an overdetermined system.

Since the L_2 finite element method produces a very good approximation to the exact solution, we may use the information provided by the L_2 solution to find "shocked" elements. Then we use the L_2 method again, but this time we put a small weight for "shocked" elements, and repeat this procedure a few times until the L_1 solution is reached.

The arrangement of this paper is as follows. The L_2 method and the convergence tests for smooth problems are presented in Section 2. In Section 3 we describe the L_1 procedure. The numerical results in Section 4 contains the L_1 solutions for the pure convection problems with discontinuities. Conclusions are drawn in Section 5.

2. The L_2 Finite Element Method

2.1 *Preliminaries and Notations.* As we have already shown in our previous papers (see [19] and the references therein), the L_2 finite element method is a universal method for the numerical solution of any type 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 L_2 finite element method always gives a reasonably good approximate solution. The work done in this paper is a natural extension of our L_2 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 L_2 finite element method introduced in [3] to get an approximate solution. The time marching is necessary for the Euler equations in aerodynamics[17,18], since the Euler equations have nonunique solutions. The time-marching L_2 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 L_2 finite element method to attack it. The general formulation of L_2 finite element methods for first-order partial differential equations can be found in [19].

In order to compare the L_2 finite element method with other finite element methods, we would like to discuss more details here. Let us consider the following more general linear hyperbolic equation:

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

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

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

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

$L_1(\Omega)$ denotes the space of functions defined on Ω, whose absolute values are integrable with the norm

$$\|u\|_{L_1} = \int_\Omega |u| d\Omega \qquad u \in L_1(\Omega).$$

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

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

$$< u,w >_+ = \int_{\Gamma_+} uw\mathbf{n}\cdot\beta ds,$$

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

where

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

We note that by Green's formula

$$(u_\beta, w) =< u,w > -(u,w_\beta).$$

Further we also define the function space

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

and the corresponding finite element subspace S_h, i.e. S_h is the space of continuous piecewise polynomial functions of degree k. Here the parameter h represents the maximal diameter of the elements. By the finite element interpolation theory[7,27] we have: Given a function $u \in H^{k+1}(\Omega)$, there exists an interpolant $\hat{u}^h \in S_h$ such that

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

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

2.2 *The Standard Galerkin Method.* Now let us look at the following standard Galerkin method for the problem (2) (see e.g., Johnson[20]): 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

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

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

$$\|u\| + \|u_\beta\| + |u|_\Gamma \le c\|f\|, \tag{6}$$

But in the standard Galerkin method the stability estimate is

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

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

2.3 *The SUPG Method.* In order to get better accuracy and stability, methods of upwinding type have appeared, see e.g., papers by Dendy[8], Wahlbin[32],Christie, Griffths, Mitchell and Zienkiewicz[6], Hughes and Brooks[15], Johnson, Nävert and Pitkäranta[21], Morton and Parrot[26]. Below we shall look at the streamline upwinding Petrov-Galerkin method (SUPG)(Hughes[14]) or the streamline diffusion method(Johnson[20]): Find $u^h \in S_h$ such that

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

Johnson and his colleagues have derived the error estimate:

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

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

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

which means that the streamline derivative is less controlled. 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 L_2 Method.* Now let us introduce the L_2 finite element method. 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 L_2 method is based on minimizing the residual function in a least-squares sense. We construct the least-squares functional:

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

The L_2 method reads: Find $u \in S$ such that

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

Taking variation of I with respect to v, and setting $\delta I = 0$ and $\delta v = w$, lead to the L_2 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 L_2 finite element method 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 get 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 $\theta = u^h - \hat{u}^h$ so that $e = \rho + \theta$. Then we have

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

or

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

Recalling (3) we obtain the error estimate:

$$\|e_\beta + e\| \leq c h^k \|u\|^{k+1}. \tag{15}$$

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

$$\|R^h\| \leq c h^k \|u\|^{k+1}, \tag{16}$$

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

$$(\|e\|^2 + \|e_\beta\|^2 + < e, e >_+)^{\frac{1}{2}} \leq ch^k \|u\|_{k+1}, \tag{17}$$

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 L_2 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:

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

This estimate is the same as the above estimate (6) for the continuous problem. It means that the L_2 method has better control of the streamline derivative. 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 L_2 method over other methods in practice.

2.5 *Numerical Experiments of the L_2 Method.* We chose the following model problem:

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

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

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

We have tested the bilinear element with uniform meshes. At first the one-point Gaussian quadrature was used for calculating the stiffness matrices. In the case of one-point quadrature, the L_2 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 L_2 solution and the direct collocation solution. The numerical result for convergence rate is shown in Figure 1. The optimal rate, i.e. $\|e\| \leq ch^2$, is observed.

Also we would like to mention that the L_2 method with one-point quadrature is equivalent to the central finite difference scheme. Therefore, the L_2-optimality may be derived by using the finite difference theory.

The numerical rate of convergence with the 2×2 Gauss rule is also included in Figure 1. In this case, the L_2 method 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 L_2 method with the 2×2 Gauss rule gives the optimal rate of convergence $\|e\| \leq ch^2$ (Figure 1).

3. The L_1 Finite Element Method

If the solution is non-smooth, the above L_2 method still performs quite well. Of course, the argument about the error estimates does not hold. As expected, the L_2 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 L_2 method does not recognize them, and just equally treats "shocked" and "smooth" elements. This is the reason that we would like to use the L_1 idea to suppress the interference of "shocked" elements.

We still consider the problem (2). We want to minimize the L_1 norm of the residual:

$$\|R\|_{L_1} = \int_\Omega |u_\beta + u - f| d\Omega. \tag{20}$$

Obviously, we should work on the discretized version of (20) by using the finite element method: i.e. Find the minimizer $\bar{u}^h$ of

$$\|R^h\|_{L_1} = \sum_{j=1}^{nelem} \sum_{l=1}^{ngaus} w_l |R_l| |J(\xi_l, \eta_l)|, \tag{21}$$

in which

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

where R_l stands for the residual at each Gaussian point, *ngaus* denotes the number of Gaussian points, w_l is the Gaussian weighting, $|J|$ is the determinant of the Jacobian matrix, and (ξ_l, η_l) is the local coordinates of Gaussian points. As usual, u^h can be expressed by the shape functions and the nodal values:

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

where "*nnode*" is the number of nodes in an element, Ψ_m are the shape functions, and U_m are the nodal values. In order to make the problem (21) meaningful, we must have an overdetermined algebraic system. It can be realized simply by appropriately choosing the number of Gaussian points.

Now the task becomes finding the solution which has the least absolute residual. It is well known that any L_1 problem can be transferred into a linear programming problem[2]. Then we may use a linear programming algorithm to find the L_1 solution. But right now this type of algorithm is too time-consuming. We are still waiting for a fast algorithm which can at least take the advantage of sparse finite element matrix.

Fortunately, we can use another approach, e.g., an iteratively reweighte least-squares method($\mathrm{IR}L_2$)[2], which is based on repeatedly solving a weighted least-squares problem: Find the minimizer $\bar{u}^h$ of

$$I_1(u^h) = \sum_{j=1}^{nelem} \sum_{l=1}^{ngaus} W_l w_l |R_l|^2 |J(\xi_l, \eta_l)|, \tag{23}$$

in which

$$W_l = \frac{1}{|R_l|_{previous}}, \tag{24}$$

where W_l denotes the weight set, which in turn depends on the information of the previous step. The $\mathrm{IR}L_2$ would begin with the initial weight set $W_l = 1$. This first step is nothing but the L_2 method introduced in Section 2. The result of this L_2 method determines a new set of weights by (24). 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 $\|\bar{u}^h_{current} - \bar{u}^h_{previous}\|$ is small.

Our numerical experiments reveal that the above method converges extremely slowly. The problem is that the difference between the residuals of "shocked" and their neighboring elements in the first L_2 solution is not significant enough. This difficulty can be overcome simply by using the following weights:

$$W_l = \frac{1}{|R_l|^6_{previous}}. \tag{25}$$

It corresponds to additionally increasing the importance of "smooth" elements and reducing the inference of "shocked" elements. This is reasonable, since the theory of L_1 fitting[2] tells us that the L_1 procedure eliminates completely the equations, which will have nonzero residuals, from the system. This trick is usable, also because our non-weighted L_2 method is good enough to locate the "shocked" elements. That is, in the results of our L_2 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

$$V = \sum_{m=1}^{nnode} |U_m - U_{m-1}|, \qquad U_0 = U_{nnode}. \tag{26}$$

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 only 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 L_1 method is really straightforward. If an L_2 finite element code is already available, it needs only a few additional lines of FORTRAN statements.

4. Numerical Results of the L_1 Method

We consider the following problem:

$$\frac{\partial u}{\partial x} + tan(35^o)\frac{\partial u}{\partial y} = 0 \qquad in \quad \Omega, \tag{27.a}$$

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

$$u = 2 \quad on \quad \Gamma_1 = \{(x,y) \in \Gamma : x = 0\}, \tag{27.b}$$

$$u = 1 \quad on \quad \Gamma_2 = \{(x,y) \in \Gamma : x \geq h \ and \ y = 0\}, \tag{27.c}$$

in which h is a positive constant less than 1 (h will be a mesh length). Equations (27) represent uniform flow along straight lines inclined at an angle of 35^o with respect to the x-axis. In this case, any straight line, which is between and parallel to $y = xtan(35^o)$ and $y = (x-h)tan(35^o)$, could be considered as the location of discontinuity. For example, we may write the solution of (27) as

$u = 2$ on and above the line $y = (x - \frac{h}{2})tan(35^o)$,

$u = 1$ below the line $y = (x - \frac{h}{2})tan(35^o)$.

The jump discontinuity occurs along the line $y = (x - \frac{h}{2})tan(35^o)$.

The boundary conditions (27.b) and (27.c) can be transferred into the source term in the equation (27.a). Therefore, the formulation of the L_2 method described in Section 2 can be directly applied to the problem (27).

The computational results presented in this paper were obtained in double precision on our 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 cut by using the preconditioned conjugate gradient method[16], since the L_2 solution is already close to L_1 solution and the final iteration is often just for correcting one or two nodal values which have not yet reached 15 digit accuracy.

4.1 *Linear Triangle Element.* At first numerical experiments were carried out for the problem (27) using linear triangle elements on uniform meshes with $n = 5, 15$. Here n is the number of grids in each coordinate. For triangle elements, we use the one-point Gaussian quadrature. Since there are $2n^2$ elements, it corresponds to having $2n^2$ 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 L_2 method amounts to solving an overdetermined system. It does not make sense to take more quadrature points, because in a linear triangle 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 L_2 results for $n = 5$ (50 triangle elements) are listed in Table 1. The numbers in Table 1 are the nodal values. Because the mesh is very coarse, the jump discontinuity is smeared severely. Starting from this bad L_2 solution, after 4 iterations of IRL_2, we obtained the perfect L_1 solution listed in Table 2. This solution has absolutely accurate 15 digits. Here we should note that a double precision (8 bytes or 64 bits) real number in a computer can only represent a decimal number with 15 digits. This solution has completely no oscillation and no diffusion. The transition over the discontinuity is accurately located in the vicinity of the line $y = (x - \frac{h}{2})tan(35^o)$, which has a width of $hsin(35^o)$ in a sense discussed above, and is accomplished in just one element.

The L_2 solution for n = 15 (450 triangle elements) is given in Table 3. This solution is diffused, and slightly oscillatory around the jump. Starting from this L_2 solution, the accurate L_1 solution is obtained after 4 iterations of IRL_2 (see Table 4). The L_1 solution again has 15 digit accuracy. Because of the limitation of page size, we give the nodal values with only 8 digits in Table 4.

4.2 *Bilinear Element.* Numerical experiments were also carried out for the problem (27) using bilinear elements on uniform meshes with n = 5,15,40, and 80. 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

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

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^o or 135^o 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 L_2 solution of a coarse mesh with 5×5 bilinear elements is listed in Table 5. Starting from this rough solution, after 4 iterations we obtained the L_1 solution listed in Table 6. We again observed 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 7 and Table 8. The large numbers (2 for residual and 8 for variation) indicate the "shocked" elements.

The L_2 solution of a mesh with 15×15 bilinear elements is presented in Table 9, and the corresponding contours are given in Figure 2. This approximate solution is reasonably good, although the discontinuity is smeared out, and slight oscillations occur. From this table and this figure, we can hardly tell where the jump is located. However, after 5 iterations, a clean L_1 solution is reached (Table 10 and Figure 3). This solution again has the correct 15 digits. The element residuals and variations are given in Table 11 and Table 12 to contrast "shocked" elements with "smooth" elements.

The L_2 solution of a mesh with 40×40 bilinear elements is illustrated in Figure 4. Taking this L_2 solution as an initial solution, after 8 steps of processing, we obtained the L_1 solution illustrated in Figure 5. No other currently available methods can produce such a sharp discontinuity as this.

We also did numerical tests for meshes with up to 80×80 elements, combined with various inflow angles and different boundary conditions. All of our L_1 results are perfectly accurate. Because of the page limitation, we do not present these results here.

5. Conclusions

A new L_1 procedure based on the iteration of L_2 finite element method for the solution of pure convection problems is developed. The overdetermined algebraic system is inherently obtained by choosing an appropriate number of Gaussian points in the formation of element matrices. The time-consuming linear programming for solving overdetermined systems turns out to be not necessary.

This L_1 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 and no oscillation, and has superior accuracy. The method is simple and robust, and can be easily extended to three-dimensional pure convection 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, and image enhancement. We have already extended this method to two-dimensional compressible flows with shocks.

Acknowledgement

The author would like to express his appreciation of the discussion with Dr. John Lavery and Professor S.H.Chang in the preparation of this paper.

References

1. I.Barrodale and F.D.K. Roberts, *SIAM J.Numer.Anal.* **10**, 839 (1973).
2. P.Bloomfield and W.L.Steiger, *Least Absolute Deviations* (Birkhauser, Boston, 1983).
3. G.F.Carey and B.N.Jiang, *Inter.J.Numer.Meth.Engng.* **26**, 81 (1988).
4. G.F.Carey and J.T.Oden, *Finite Elements: A Second Course, Vol.II*, (Prentice-Hall, Englewood Cliffs, NJ, 1986), p.199.
5. M.Chapman,*J.Comput.Phys.* **44**, 84 (1981).
6. I.Christie,D.F.Griffiths, A.R.Michell and O.C.Ziekiewicz, *Inter.J.Numer.Meth.Engng.* **10**, 1389 (1976).
7. P.G.Ciarlet, *The Finite Element Method for Elliptic Problems* (North-Holland, Amsterdam, 1978).
8. J.E.Dendy, *SIAM J.Numer.Anal.* **11**, 637 (1974).
9. B.Engquist, P.Lötstedt and B.Sjögreen, *Mathematics of Computation*, **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* (John Wiley, Chichester, 1988).
14. T.J.R.Hughes, *Inter.J.Numer.Meth.Fluids* **7**, 1261 (1987).
15. T.J.R.Hughes and A.Brooks, "A multidimensional upwind scheme with no crosswind

diffusion", in *Finite Element Methods for Convection Dominated Flows*, Edited by T.J.Hughes, AMD Vol.34, (ASME, NY, 1979).

16. 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, U.K., 1984).
17. B.N.Jiang and G.F.Carey, *Inter.J.Numer.Meth.Fluids* **8**, 933 (1988).
18. B.N.Jiang and G.F.Carey, *Inter.J.Numer.Meth.Fluids* **10**, 557 (1990).
19. B.N.Jiang and L.A.Povinelli, *Comput.Meth.Appl.Mech.Engng.* **81**, 13 (1990).
20. C.Johnson, *Numerical Solution of Partial differential Equations by the Finite Element Method* (Cambridge University Press, Cambridge,1987).
21. C.Johnson, U.Nävert and J.Pitkäranta, *Comput.Meth.Appl.Mech.Engng.* **45**, 285 (1984).
22. S.Koshizuka, C.B.Carrico, S.W.Lomperski, Y.Oka, and Y.Togo, *Computational Mechanics* **6**, 65(1990).
23. J.E.Lavery, *J.Comput.Phys.* **79**, 436 (1988).
24. J.E.Lavery, *SIAM J.Numer.Anal.* **26**, 1081 (1989).
25. J.E.Lavery, "Capturing contact discontinuities in steady-state conservation laws", *J.Comput.Phys.* submitted.
26. K.W.Morton and A.K.Parrot, *J.Comput.Phys.* **36**, 249 (1980).
27. J.T.Oden and G.F.Carey, *Finite Elements: Mathematical Aspects, Vol.IV* (Prentice-Hall, Englewood Cliffs, NJ, 1983).
28. 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, NY, 1988).
29. O.Pironneau, *Finite Element Method for Fluids* (John Wiley, London, UK, 1989).
30. E.V.Vorozhtsov, *Comput. fluids* **15**, 13 (1987).
31. E.V.Vorozhtsov, *Comput. fluids* **18**, 35 (1990).
32. L.B.Wahlbin, *R.A.I.R.O.* **8**, 109 (1974).

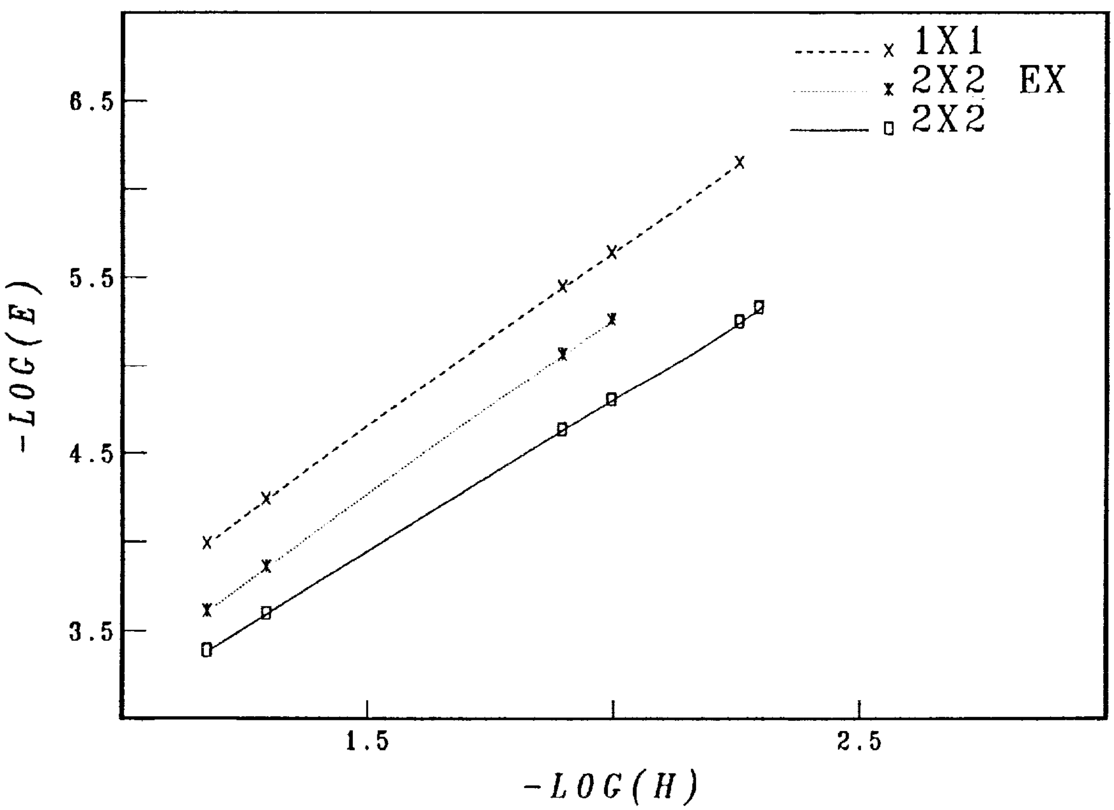

Figure 1. Computed convergence rate for the pure convection problem.

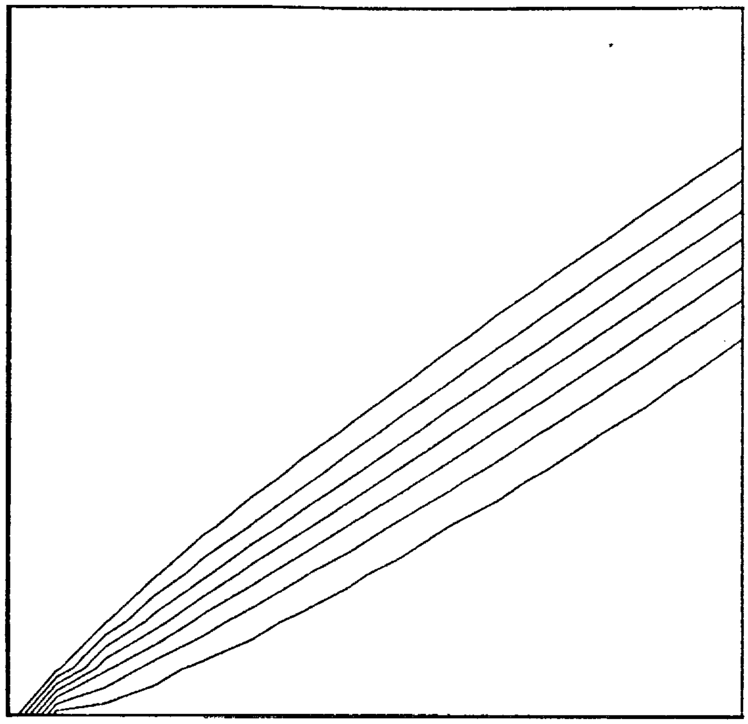

Figure 2. Contours of the L_2 solution for the pure convection problem (15×15 bilinear elements)

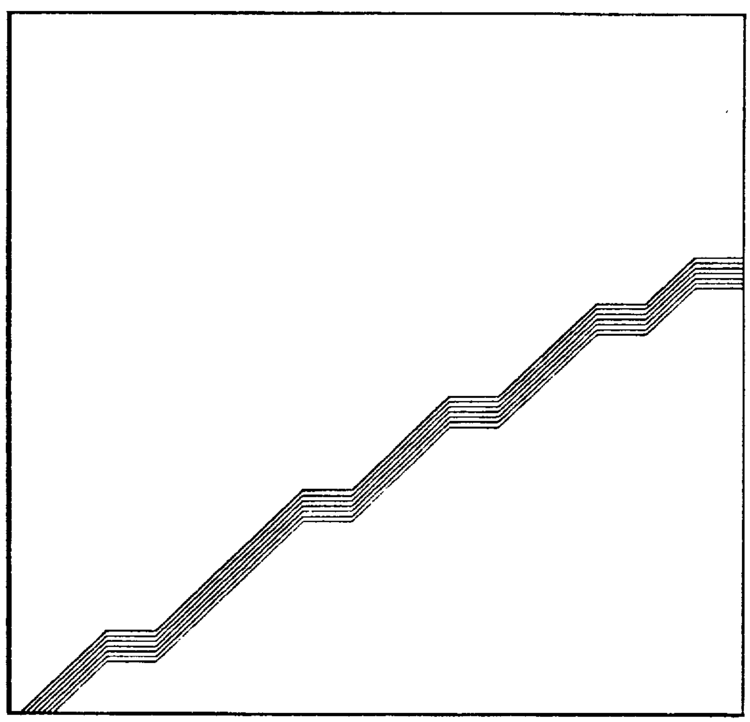

Figure 3. Contours of the L_1 solution for the pure convection problem (15×15 bilinear elements)

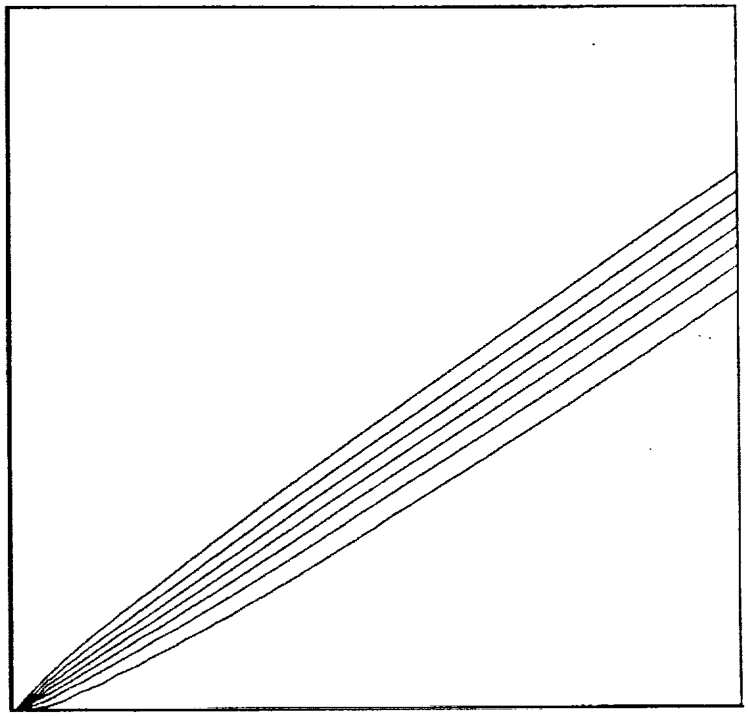

Figure 4. Contours of the L_2 solution for the pure convection problem (40 × 40 bilinear elements)

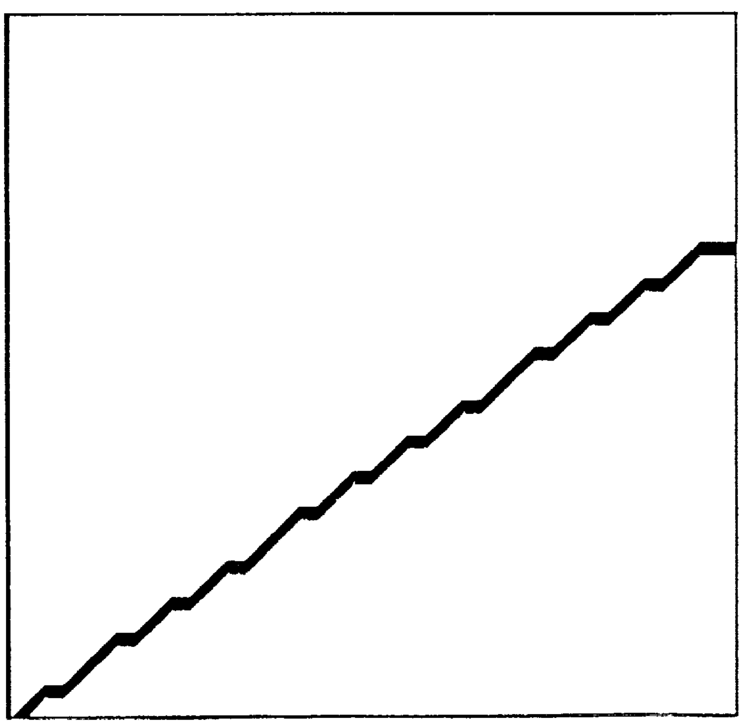

Figure 5. Contours of the L_1 solution for the pure convection problem (40 × 40 bilinear elements)

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 1. Nodal Values of L2 Solution for n = 5 (50 triangle 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 2. Nodal Values of L1 Solution for n = 5 (50 triangle Elements)

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 3. Nodal Value of L2 Solution for n = 15 (450 Triangle 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	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

Table 4. Nodal Values of L1 solution for n = 15 (450 triangle 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 5. Nodal Values of L2 Solution for n = 5 (25 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 6. Nodal Values of L1 Solution for n = 5 (25 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.7 Element Residuals of L1 Solution for n = 5 (25 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.8 Element Variations of L1 Solution for n = 5 (25 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

Table 9. Nodal Values of L2 Solution for n = 15 (225 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

Table 10. Nodal Values of L1 Solution for n = 15 (225 Bilinear Elements)

0.17E-14	0.98E-15	0.74E-15	0.26E-14	0.36E-14	0.98E-15	0.88E-15	0.49E-14	0.71E-14	0.11E-13	0.14E-13	0.12E-13	0.83E-14	0.41E-14	0.18E-14
0.34E-16	0.19E-14	0.37E-14	0.44E-14	0.12E-14	0.26E-14	0.62E-14	0.11E-13	0.13E-13	0.13E-13	0.97E-14	0.48E-14	0.98E-15	0.26E-14	0.62E-14
0.00E+00	0.13E-14	0.25E-14	0.95E-15	0.44E-14	0.10E-13	0.15E-13	0.14E-13	0.88E-14	0.53E-14	0.19E-14	0.17E-14	0.36E-14	0.73E-14	0.18E-14
0.89E-15	0.19E-14	0.50E-15	0.55E-14	0.11E-13	0.13E-13	0.89E-14	0.17E-14	0.82E-15	0.19E-14	0.35E-14	0.44E-14	0.56E-14	0.10E-15	0.12E-13
0.13E-15	0.52E-15	0.52E-14	0.99E-14	0.11E-13	0.69E-14	0.23E-15	0.35E-14	0.35E-14	0.38E-14	0.27E-14	0.21E-14	0.46E-14	0.16E-13	0.18E-13
0.00E+00	0.44E-14	0.11E-13	0.88E-14	0.54E-14	0.24E-16	0.44E-14	0.26E-14	0.36E-16	0.72E-15	0.18E-14	0.70E-14	0.11E-13	0.20E+01	0.20E+01
0.44E-14	0.80E-14	0.53E-14	0.98E-15	0.27E-14	0.51E-14	0.18E-14	0.93E-16	0.96E-15	0.72E-15	0.26E-14	0.20E+01	0.20E+01	0.20E+01	0.36E-14
0.36E-14	0.84E-15	0.29E-14	0.46E-14	0.46E-14	0.18E-14	0.11E-15	0.12E-17	0.85E-15	0.83E-15	0.20E+01	0.20E+01	0.26E-16	0.49E-15	0.17E-14
0.35E-14	0.80E-17	0.19E-14	0.80E-17	0.21E-14	0.15E-14	0.18E-14	0.19E-14	0.20E+01	0.20E+01	0.20E+01	0.26E-14	0.47E-15	0.17E-14	0.31E-14
0.82E-15	0.18E-14	0.42E-16	0.90E-15	0.25E-15	0.21E-14	0.78E-16	0.20E+01	0.20E+01	0.18E-14	0.22E-14	0.89E-15	0.10E-14	0.35E-14	0.22E-14
0.35E-15	0.74E-15	0.69E-16	0.19E-14	0.65E-14	0.20E+01	0.20E+01	0.20E+01	0.34E-14	0.22E-14	0.86E-15	0.39E-15	0.18E-14	0.84E-15	0.88E-15
0.18E-14	0.22E-14	0.34E-16	0.70E-15	0.20E+01	0.20E+01	0.23E-14	0.23E-14	0.17E-14	0.87E-15	0.31E-15	0.91E-15	0.13E-15	0.22E-16	0.14E-14
0.34E-16	0.92E-15	0.24E-14	0.20E+01	0.20E+01	0.40E-14	0.18E-14	0.60E-15	0.97E-15	0.19E-14	0.14E-14	0.98E-16	0.97E-15	0.26E-14	0.44E-14
0.00E+00	0.20E+01	0.20E+01	0.20E+01	0.14E-14	0.84E-15	0.17E-14	0.22E-14	0.30E-14	0.13E-14	0.52E-16	0.18E-14	0.35E-14	0.38E-14	0.22E-14
0.20E+01	0.20E+01	0.22E-14	0.13E-14	0.18E-14	0.18E-14	0.14E-14	0.14E-14	0.84E-15	0.47E-15	0.14E-14	0.22E-14	0.89E-15	0.62E-16	0.11E-14

Table 11. Element Residuals of L1 Solution for n = 15 (225 Bilinear Elements)

0.71E-14	0.71E-14	0.71E-14	0.11E-13	0.14E-13	0.71E-14	0.12E-13	0.20E-13	0.28E-13	0.46E-13	0.57E-13	0.50E-13	0.32E-13	0.18E-13	0.14E-13
0.71E-14	0.11E-13	0.14E-13	0.18E-13	0.89E-14	0.11E-13	0.25E-13	0.43E-13	0.53E-13	0.50E-13	0.39E-13	0.18E-13	0.36E-14	0.14E-13	0.25E-13
0.00E+00	0.53E-14	0.11E-13	0.11E-13	0.18E-13	0.41E-13	0.60E-13	0.57E-13	0.36E-13	0.21E-13	0.14E-13	0.11E-13	0.14E-13	0.28E-13	0.21E-13
0.36E-14	0.71E-14	0.89E-14	0.21E-13	0.43E-13	0.53E-13	0.36E-13	0.14E-13	0.11E-13	0.14E-13	0.14E-13	0.18E-13	0.21E-13	0.32E-13	0.46E-13
0.36E-14	0.71E-14	0.21E-13	0.39E-13	0.43E-13	0.28E-13	0.11E-13	0.14E-13	0.18E-13	0.18E-13	0.14E-13	0.11E-13	0.25E-13	0.64E-13	0.75E-13
0.00E+00	0.23E-13	0.44E-13	0.36E-13	0.21E-13	0.14E-13	0.18E-13	0.14E-13	0.36E-14	0.36E-14	0.71E-14	0.28E-13	0.43E-13	0.80E+01	0.80E+01
0.18E-13	0.32E-13	0.21E-13	0.11E-13	0.18E-13	0.21E-13	0.18E-13	0.36E-14	0.71E-14	0.36E-14	0.11E-13	0.80E+01	0.80E+01	0.80E+01	0.14E-13
0.18E-13	0.18E-13	0.18E-13	0.21E-13	0.21E-13	0.14E-13	0.14E-13	0.11E-13	0.71E-14	0.71E-14	0.80E+01	0.80E+01	0.00E+00	0.53E-14	0.71E-14
0.14E-13	0.71E-14	0.71E-14	0.71E-14	0.71E-14	0.71E-14	0.71E-14	0.11E-13	0.80E+01	0.80E+01	0.80E+01	0.14E-13	0.53E-14	0.71E-14	0.12E-13
0.14E-13	0.11E-13	0.71E-14	0.71E-14	0.11E-13	0.14E-13	0.11E-13	0.80E+01	0.80E+01	0.71E-14	0.11E-13	0.53E-14	0.71E-14	0.14E-13	0.89E-14
0.53E-14	0.89E-14	0.71E-14	0.11E-13	0.30E-13	0.80E+01	0.80E+01	0.80E+01	0.13E-13	0.89E-14	0.36E-14	0.71E-14	0.71E-14	0.71E-14	0.36E-14
0.89E-14	0.89E-14	0.00E+00	0.36E-14	0.80E+01	0.80E+01	0.89E-14	0.89E-14	0.71E-14	0.36E-14	0.36E-14	0.36E-14	0.18E-14	0.00E+00	0.53E-14
0.71E-14	0.71E-14	0.89E-14	0.80E+01	0.80E+01	0.16E-13	0.71E-14	0.18E-14	0.53E-14	0.71E-14	0.71E-14	0.36E-14	0.53E-14	0.11E-13	0.18E-13
0.00E+00	0.80E+01	0.80E+01	0.80E+01	0.53E-14	0.71E-14	0.71E-14	0.11E-13	0.12E-13	0.89E-14	0.53E-14	0.11E-13	0.14E-13	0.15E-13	0.98E-14
0.80E+01	0.80E+01	0.89E-14	0.71E-14	0.71E-14	0.71E-14	0.53E-14	0.53E-14	0.36E-14	0.53E-14	0.89E-14	0.89E-14	0.36E-14	0.44E-14	0.44E-14

Table 12. Element Variations of L1 Solution for n = 15 (225 Bilinear Elements)

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No. NASA TM - 103773 ICOMP-91-03	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle The L_1 Finite Element Method for Pure Convection Problems		5. Report Date April 1991
		6. Performing Organization Code
7. Author(s) Bo-nan Jiang		8. Performing Organization Report No. E-6039
		10. Work Unit No. 505-62-21
9. Performing Organization Name and Address National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546-0001		14. Sponsoring Agency Code

15. Supplementary Notes

Bo-nan Jiang, Computational Physics System, P.O. Box 4255, Ann Arbor, Michigan 48106 and Institute for Computational Mechanics in Propulsion, Lewis Research Center (work funded by Space Act Agreement C-99066-G). Space Act Monitor: Louis A. Povinelli, (216) 433-5818.

16. Abstract

In this paper we first introduce the least-squares (L_2) finite element method for two-dimensional steady-state pure convection problems with smooth solutions. We prove that the L_2 method has the same stability estimate as the original equation, that is, the L_2 method has better control of the streamline derivative. Numerical convergence rates are given to show that the L_2 method is almost optimal. Then we use this L_2 method as a framework to develop an iteratively reweighted L_2 finite element method to obtain a least absolute residual (L_1) solution for problems with discontinuous solutions. This L_1 finite element method produces a non-oscillatory, non-diffusive and highly accurate numerical solution that has a sharp discontinuity in one element on both coarse and fine meshes. We also devise a robust reweighting strategy to obtain the L_1 solution in a few iterations. A number of examples solved by using triangle and bilinear elements are presented.

17. Key Words (Suggested by Author(s)) Convection; Hyperbolic; Discontinuity; Finite element; Least squares; Least absolute; Residual	18. Distribution Statement Unclassified - Unlimited Subject Category 64

19. Security Classif. (of the report)	20. Security Classif. (of this page)	21. No. of pages	22. Price*
Unclassified	Unclassified	28	A03

NASA FORM 1626 OCT 86 *For sale by the National Technical Information Service, Springfield, Virginia 22161

PAGE 27 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED