UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

High Order 2D Non Oscillatory Methods for Solving Hamilton-Jacobi Scalar Equations

F. Lafon
S. Osher

May 1992

CAM Report 92-20

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90024-1555

High Order 2D Non Oscillatory Methods for solving Hamilton-Jacobi scalar equations

F.Lafon*
C.E.A - CEL-V
Department of Mathematics, MCN-LH, BP 27
94195 Villeneuve Saint Georges, France
& S.Osher†
Department of Mathematics - University of California
Los Angeles, CA 90024-1555

April 27, 1992

Abstract

For the computation of nonlinear solutions of Hamilton-Jacobi scalar equations in two space dimensions, we develop high order accurate numerical schemes that can be applied on complicated geometries. Previously, the recently developed Essentially Non-Oscillatory (ENO) technology has been applied in simple domains like squares or rectangles by using dimension by dimension algorithms. On arbitrary two dimensional closed or multiply connected domains, first order monotone methods were used. In this paper, we propose two different techniques to construct high order accurate methods using the ENO philosophy. Namely, any arbitrary domain is triangulated by finite elements into which two dimensional ENO polynomials are constructed. These polynomials are then differentiated to compute a high order accurate numerical solution. These new techniques are shown to be very useful in the computation of numerical solutions of various applications without significantly increasing CPU running times as compared to dimension by dimension algorithms. Furthermore, these methods are stable and no spurious oscillations are detected near singular points or curves.

*Research supported in parts by ONR grant N00014-86-K-0691 and NSF grant DMS 88-11863 while the author was visiting the Department of Mathematics at University of California, Los Angeles.

†Research supported by ONR grant N00014-86-K-0691, DARPA grant in the ACMP program, NFS grant DMS 88-11863, and NASA Langley NAGI-270.

1 Introduction

During the past decades, there has been a considerable amount of work committed to constructing high order accurate numerical methods for solving nonlinear partial differential equations whose solutions (or their derivatives) develop steep gradients. This paper is devoted to the special case of the so-called nonlinear scalar Hamilton-Jacobi equations. The model problem is to find a numerical solution of :

$$\begin{aligned} U_t + H(U_x, U_y, x, y) &= 0 \\ U_(x, y, 0) &= U_0(x, y), \end{aligned} \tag{1}$$

given some combination of Dirichlet and Neumann boundary conditions, where $H(U_x, U_y, x, y)$ is a generally nonlinear Hamiltonian in two space dimensions. The domain Ω is an arbitrary simply or multiply connected domain in R^2 whose boundaries $\partial\Omega$ are assumed to be at least either piecewise C^0 (Dirichlet or periodic boundary conditions) or piecewise C^1 (Neumann boundary condition) so that an outward normal can be defined except at isolated points.

Up to now, most numerical experiments have used cartesian based algorithms and regular meshes restricting the shape of domains Ω to simple configurations like squares or rectangles. Some numerical methods using first order monotone numerical schemes can be found in [2]. Later, high order numerical methods for solving combustion problems in two dimensions were derived in [11] and extended in [12] to high order Essentially Non Oscillatory schemes (ENO) for rather general Hamiltonians. Other attempts for solving hyperbolic problems on unstructured grids can been found in [4, 8, 10, 13, 17].

The domain Ω is assumed to be arbitrary and may have complicated boundaries. For that reason, we introduce new numerical methods acting on arbitrary grids that fit the boundaries by triangular meshes. However, the interior of the domain is discretized by square meshes and more conventional two dimensional techniques are constructed in order to reduce significantly CPU time calculations. The regular square triangulation is denoted by TI_h and the boundary fitting triangulation by TB_h (refer to figure 9 for an example). These new techniques are closely related to the ENO idea. The ENO apparatus can be used as a preprocessing step in order to evaluate some non-oscillatory two dimensional polynomials which are then differentiated for computing accurate pointwise value of U_h at vertices A_j and some derivatives $\partial_x^{n_x}\partial_y^{n_y}$ of U_h at vertices C_k. The vertices A_j for $j = 1...NV$ are defined as the nodal values of the triangulation τ_h, and the $C_k's$, for $k = 1...NT$, are the center of gravity of each triangle K of τ_h.

The general algorithm to compute a numerical approximation of equation (1) will follow these three steps :

- 1.) On each triangle, we construct a two dimensional ENO polynomial of degree 2,3, and up.
- 2.) The numerical solution $U_h(A_j)$, $j = 1...NV$, is computed at each time step by using the ENO polynomials defined in step 1) and high order non-oscillatory Runge-Kutta discretization time schemes.
- 3.) We evaluate an accurate approximation of $\nabla U_h(C_k)$, for $k = 1...NT$, by solving local Riemann problems along the edges of K. To do so, we approximate a numerical solution of the non-strictly hyperbolic system obtained by differentiating (1) with respect to x and y :

$$\frac{\partial V}{\partial t} + \frac{H(V, W)}{\partial x} = 0, \tag{2}$$

$$\frac{\partial W}{\partial t} + \frac{H(V, W)}{\partial y} = 0, \tag{3}$$

where ($V = \frac{\partial U}{\partial x}, W = \frac{\partial U}{\partial y}$) is the gradient vector of U. Some boundary and initial conditions are added to (2) and (3) according to prescribed conditions on the model problem (1).

Our goal is to compute the best possible non oscillatory polynomial (step 1 of previous algorithm) on any K of τ_h without excessively increasing CPU time calculations as compared to dimension by dimension algorithms. In addition, we want to minimize (make as local as possible) the search for suitable ENO polynomials with the goal of constructing a sequence of stable approximate solutions. Finally, we want to compute a high order accurate sequence of numerical approximations U_h to the exact solution U of (1).

These results are achieved by using two dimensional numerical methods on the grids TB_h and TI_h. The simplest calculations will be involved in TI_h, thus reducing the computational time significantly compared to more expensive work needed in TB_h. To be more precise, we use Finite Element Techniques (FET) in TI_h. That framework simplifies the algorithms and the search for an acceptable non oscillatory polynomial is straightforward. Note however that the final constructed ENO polynomial is not the optimal one. Despite this fact, numerical results demonstrate that this approach is very accurate and stable even when the solution becomes nonsmooth. In addition, this technique requires evaluations at nodal values of U_h only. Therefore, step 3 of the previous algorithm can be omitted. In the triangulation TB_h, we solve a mixed problem obtained by coupling the Hamilton-Jacobi equation (1) along with the non strictly hyperbolic system (2,3). This technique reduces the size of the stencil and makes the construction of the polynomial more local. However, it requires that accurate values of the gradient of U_h are computed via step 3 of previous algorithm. The degrees of freedom on each triangle of both triangulations are sketched in figure 1.

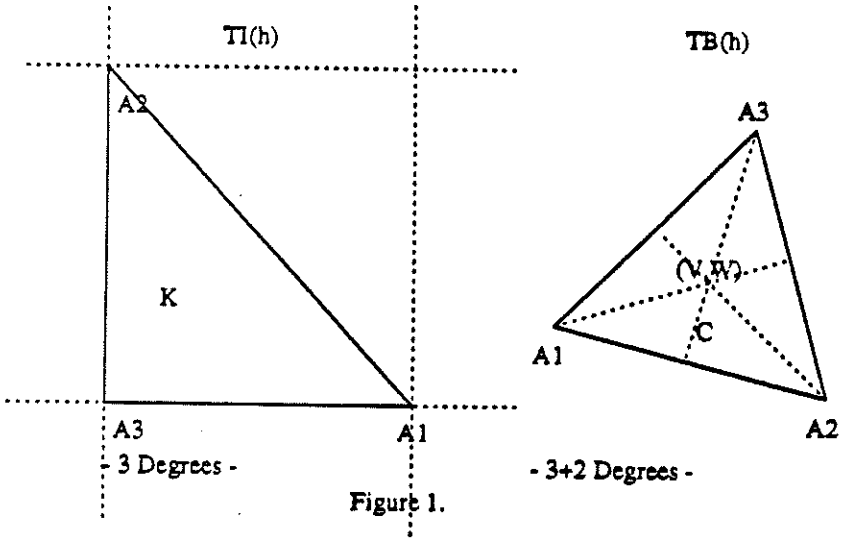


Figure 1.

In the next two sections, we present a detailed description of both techniques when Ω is either a regular square domain Ω or the unit disc. In section four, we investigate the interface problem when both triangulations are matched together. Section five presents some additional numerical examples using non convex Hamiltonians and rather general Hamiltonians of the form $H(U_x, U_y, x, y)$. The last section of this paper concludes this study and gives further comments on stability and improvements that can be done on the interior grid for which grid refinement techniques can be implemented without too much additional work.

2 Square grids and Finite Element Techniques (FET)

In this section, we consider the model problem (1) in which we assume periodic boundary conditions on the unit square $\Omega = [-1, 1]^2$. The vertices $A_{i,j} = (x_j = j * \Delta x - 1, y_i = i * \Delta y - 1)$ are defined uniformly inside the domain up to the boundary $\partial\Omega$, and the boundary set is defined by the set of points $i = 0$, $j = 0$, $i = N$, and $j = N$ where $N = \frac{2}{h}$. The stepsizes are kept uniformly constant $\Delta x = \Delta y = h$ in order to be able to implement finite element techniques requiring that the set of vertices defining the stencils of larger extended (P_2, P_3, ...) triangles, still belongs to the computational grid. Moreover, the initial triangulation is defined by cutting each square mesh into two P_1 triangles along the main diagonal $y - y_i = x - x_j$. Consequently, from the set of nodes $(A_{i,j}, A_{i,j+1}, A_{i+1,j}, A_{i+1,j+1})$, we define two triangles with nodal values $(A_{i,j}, A_{i,j+1}, A_{i+1,j+1})$ and $(A_{i,j}, A_{i+1,j}, A_{i+1,j+1})$. Therefore, from $(N+1)^2$ vertices, we define a triangulation TI_h of $2N^2$ triangles.

The algorithm introduced in section 1 can be modified as follows :

- 1.) Preprocessing step : For all triangles of TI_h, we define second, third and higher degree ENO polynomials by using FET.
- 2.) Time evolution scheme : We advance the solution in time by taking high order ENO multistage Runge-Kutta time schemes [12, 16].

2.1 Preprocessing step - Construction of two dimensional ENO polynomials

First of all, we review [6, 7, 12, 15, 16] some basic properties of ENO interpolating polynomials in one dimension. To define a $(m-1)^{th}$ accurate approximation of $(U_h)_x$ at some grid point, we interpolate some pointwise values of U_h by making use of Newton's divided differences. The final m^{th} degree polynomial satisfies :

$$Q_m(x) = \sum_{k=0}^{k=m} R_k(x) \text{ and } R_k = D_k(x - x_{i-r(k)})...(x - x_{i+s(k)+1}), \tag{4}$$

in which R_k is a polynomials of degree $\leq k$, $r(k) + s(k) = k - 1$, $r(k) = r(k-1)$ (Extension of the stencil to the right) or $r(k) = r(k-1) - 1$ (Extension of the stencil to the left), and D_k is the minimum k^{th} divided difference corresponding to the right and left extended stencils [15, 16]. The non oscillatory behavior of the final approximation comes from the fact that the polynomials R_k minimize successive k^{th} derivatives of Q_m. The initial "zero" stencil is upwinded on the direction of the characteristics and a Lax-Friedrichs approximation is enforced at sonic points. Finally, the m^{th} degree polynomial uses $m+1$ pointwise values $U_h(x_{i-r(m)}), ..., U_h(x_{i+s(m)+1})$, where $r(m) + s(m) = m - 1$; and $(U_x)_i^n$ is simply set to $\frac{\partial Q_m(x)}{\partial x}_{x=x_i}$. This construction easily extends to two dimensional problems by a dimension by dimension algorithm, and requires evaluations of ENO polynomials in each direction. Moreover, the full method is essentially non oscillatory because the ENO interpolating polynomials stay essentially monotone within meshes into which the approximate solution develops large step gradients.

Our method extends parts of the ENO basic properties for two dimensional polynomials. Somehow, derivatives or linear combinations of derivatives of two dimensional ENO polynomials should be minimized according to some preset criterion. Similarly, the m^{th} degree ENO polynomial must satisfy an equation similar to (4) but extended to two dimensions. In addition, first derivatives of the interpolating polynomials should be essentially monotone so that no significant new overshoots or undershoots are created in singular triangles.

In the case of the regular triangulated grid TI_h, we use finite element techniques for defining a m^{th} degree polynomial that interpolates $\sum_{k=1}^{m}(k+2)$ nodal values of U_h at some vertices $A_{l,n}$, for some $i-m \leq l \leq i+m$, and $j-m \leq n \leq j+m$. The construction follows basic finite element techniques. One ingredient is to use the linear transformation F (figure 2) that maps each triangle $K=(A_1,A_2,A_3)$ of TI_h onto the reference element $\hat{K}=(\hat{A}_3(\hat{x}=0,\hat{y}=0),\hat{A}_1(\hat{x}=1,\hat{y}=0)$, and $\hat{A}_2(\hat{x}=0,\hat{y}=1))$. Then, basis functions on appropriate finite element spaces $V_h^m=\{f \in C^k(K)|$ f is a m^{th} degree polynomial$\}$ are used to evaluate some m^{th} degree polynomials on $\hat{K}$.

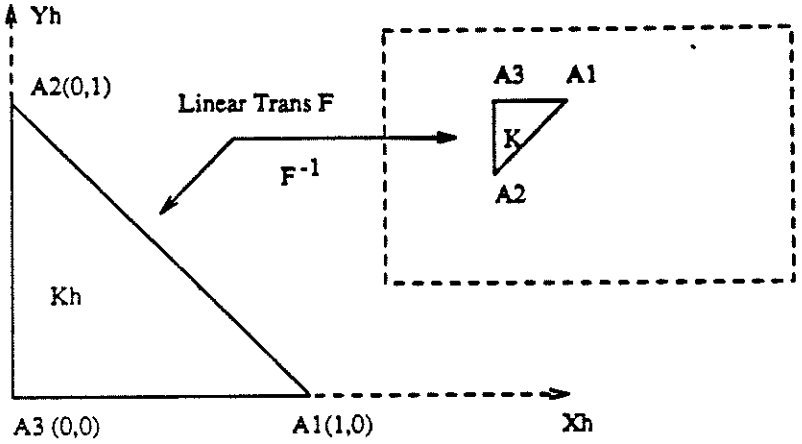


Figure 2.

In short, we proceed as follows : on the reference element, we start by constructing a piecewise linear two dimensional polynomial $\hat{P}_1(x,y)$ by interpolating the values of U_h at $(\hat{A}_1,\hat{A}_2,\hat{A}_3)$:

$$\hat{P}_1(\hat{x},\hat{y}) = U_h(\hat{A}_3) + (U_h(\hat{A}_1) - U_h(\hat{A}_3))\hat{x} + (U_h(\hat{A}_2) - U_h(\hat{A}_3))\hat{y}, \tag{5}$$

where $(\hat{x},\hat{y})$ are the coordinates on $\hat{K}$ of any point $A(x,y)$ located on K. They are classically obtained via the linear transformation F :

$$\begin{aligned} x &= (x_1-x_3)\hat{x} + (x_2-x_3)\hat{y} + x_3 = F_1(\hat{x},\hat{y}) \\ y &= (y_1-y_3)\hat{x} + (y_2-y_3)\hat{y} + y_3 = F_2(\hat{x},\hat{y}), \end{aligned}$$

where (x_j,y_j), for $j=1,2,3$, are the coordinates of A_j, $j=1,2,3$. Then, given this polynomial, we construct $\hat{P}_2(\hat{x},\hat{y})$ by extending the initial triangle K along two of its edges and by keeping fixed one of its three vertices. Figure 3 describes this extension.

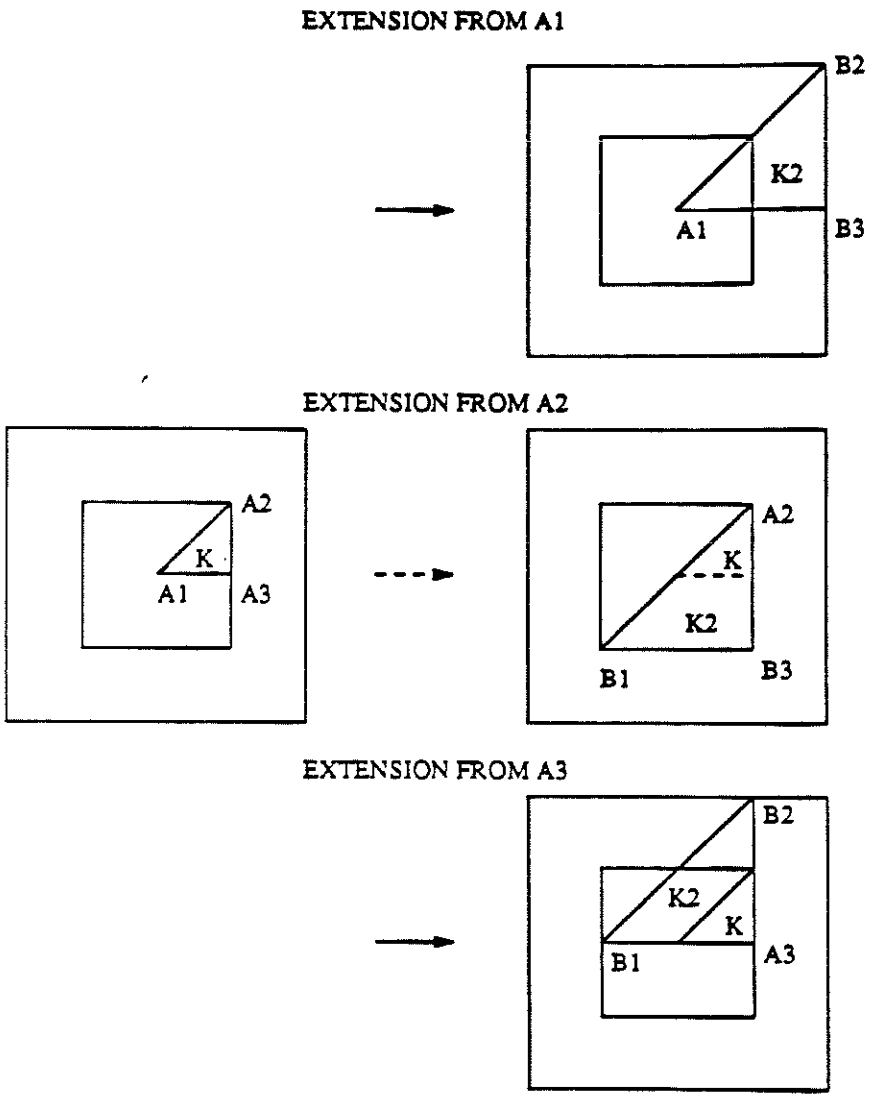


Figure 3.

The three extended triangles are denoted by $(K_2)^{1,2,3}$ and their six vertices by $(B_1, B_2, B_3, B_{12}, B_{13}, B_{23})$. where B_{ij} are the middle points of the edges defined from B_i to B_j. Note that the vertices B_l, B_{ij} still belong to the grid TI_h. Next, the second degree interpolating polynomial directly follows from basis functions defined on P_2 finite element space, i.e :

$$\hat{P}_2(\hat{x}, \hat{y}) = \sum_{i=1}^{i=3} U_h(\hat{B}_i)\lambda_i(2\lambda_i - 1) + 4 \sum_{i,j=1;i<j}^{3} U_h(\hat{B}_{ij})\lambda_i\lambda_j \tag{6}$$

where λ_i, $i = 1, 2, 3$ are the barycentric coordinates on $(\hat{K}_2)^{1,2,3}$, i.e $\lambda_1 + \lambda_2 + \lambda_3 = 1$, $\lambda_1 = \hat{x}$, and $\lambda_2 = \hat{y}$.

Again, as ENO polynomial, we choose the one that minimizes some "elliptic" combination of second derivatives (the Laplacian for example). The corresponding extended triangle from K is then denoted by S_2. This triangle can be interpreted as the non oscillatory second order stencil and is the support of the P_2 ENO polynomial.

This procedure is used to define higher order ENO polynomials by extending S_2 to S_3 (ten degrees of freedom), S_3 to S_4 (fifteen degrees), and so on. The cost for constructing these polynomials is not too high

because only simple evaluations of basis functions on appropriate finite element spaces are required. These ENO polynomials satisfy non oscillatory properties since successive derivatives are minimized in some sense. Moreover, it is proved in [1] that jumps upon the highest derivatives of any approximate solutions determine the regions in which the solution is highly singular. Therefore, by testing the amplitude of the highest derivatives of a given function, it is possible to determine the smoother part of that function in a given region (in our case, the area defined by the interpolating stencil).

Finally, the gradient of U_h at the vertex $A_{i,j}$ is defined by :

$$\begin{aligned} V_{i,j} = \frac{\partial U_{i,j}}{\partial x} &= \frac{\partial P_m(x,y)}{\partial x}_{(x=x_j, y=y_i)} \\ W_{i,j} = \frac{\partial U_{i,j}}{\partial y} &= \frac{\partial P_m(x,y)}{\partial y}_{(x=x_j, y=y_i)}, \end{aligned}$$

for a $(m-1)^{th}$ accurate approximation of the gradient (V, W) at $A_{i,j}$.

2.2 Evolution operator

For the time evolution operator, we implement any of the high order accurate ENO Runge-Kutta time schemes derived in [16]. This is a multistage scheme that needs m steps to advance the solution from time T to $T + \Delta t$, where m is the maximum order of accuracy reached by the method. The scheme has the general form :

$$U_{ij}^{n+k/m} = L(U^n, ..., U^{n+(k-1)/m}) - \Delta t \hat{H}(U^n, ..., U^{n+(k-1)/m}),$$

for $k = 1, ...m$. Here, $\hat{H}$ is the numerical Hamiltonian and depends only on $U^{n+(k-1)/m}$ for $m \leq 3$, and L is a linear combination of $U_{ij}^{n+l/m}$, for $l = 0, ..., k-1$. More details can be found in [15] and in [16].

As numerical Hamiltonian, we choose the Roe-Fix approximation defined in [12]. We recall below its definition :

$$\hat{H}(V^{\pm}, W^{\pm}) = \begin{cases} H(V^*, W^*) & \text{if } H_V(V,W) \text{ and } H_W(V,W) \\ & \text{does not change signs} \\ & \text{in } V \in I(V^-, V^+), \\ & W \in I(W^-, W^+) ; \\ H(\frac{V^+ + V_-}{2}, W^*) & \\ -\frac{1}{2}\alpha_x(V^+, V^-)(V^+ - V^-); & \text{otherwise} \\ & \text{and if } H_W(V,W) \\ & \text{does not change sign in} \\ & A \leq V \leq B, \\ & W \in I(W^-, W^+) ; \\ H(V^*, \frac{W^+ + W^-}{2}) & \\ -\frac{1}{2}\alpha_y(W^+, W^-)(W^+ - W^-); & \text{otherwise} \\ & \text{and if } H_V(V,W) \\ & \text{does not change sign in} \\ & A \leq W \leq B, \\ & V \in I(V^-, V^+); \\ H^{LF}(V^-, V^+, W^-, W^+) & \text{otherwise,} \end{cases}$$

where $(V_{ij}^{\pm}, W_{ij}^{\pm})$ represents discontinuous gradient vectors of U at A_{ij} from $\pm$ half planes, α_x and α_y are local bounded coefficients and are defined for adding enough numerical viscosity at sonic points, A and

B are lower and upper bounds within a given cell, and V^* and W^* are defined by upwinding :

$$V^* = \begin{cases} V^+ & \text{if } H_V(V,W) \leq 0, \\ V^- & \text{if } H_V(V,W) \geq 0 \end{cases}$$

$$W^* = \begin{cases} W^+ & \text{if } H_W(V,W) \leq 0, \\ W^- & \text{if } H_W(V,W) \geq 0 \end{cases}$$

where $\pm$ indicates in which half plane is located the initial support of the ENO interpolating polynomial for the x and y derivatives, respectively ; and H^{LF} is the Lax-Friedrichs approximation of the Hamiltonian H. Note that in the case of triangulation TI_h, each interior vertex of the computational grid has six neighboring triangles (figure 4).

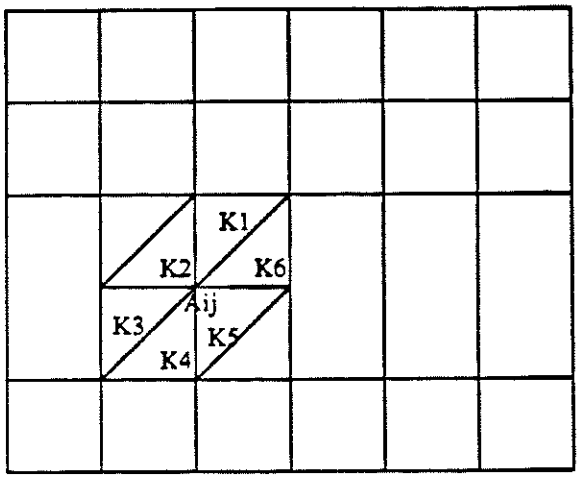


Figure 4.

Therefore, the four derivatives $(V^{\pm}, W^{\pm})$ defined in each of the four half spaces $x \leq x_{A_{ij}}, x \geq x_{A_{ij}}, y \leq y_{A_{ij}}, y \geq y_{A_{ij}}$ can be chosen according to two criteria : 1) The x and y derivatives are computed from the ENO polynomials whose initial supports (one among the triangles K_i, $i = 1, ..., 6$) are located in corresponding half planes ; 2) Since three different derivatives can be computed (three possible triangles in each half plane), we choose the one which is the least oscillatory of the three. For example, we take :

$$V_{ij}^+ = \min_{\{m^{th} \text{ derivatives}\}} \frac{\partial P_m^{K_1,K_5,K_6}(x_{ij}, y_{ij})}{\partial x^m} \tag{7}$$

where the triangle indexing is the one defined in figure 4, P_m is a m^{th} degree ENO polynomial, and the minimum is taken over the set of linear combinations of m^{th} derivatives as explained in previous section.

2.3 Model Problem - Numerical Experiment

We investigate the model problem (1) with Hamiltonian $H(V,W) = -\sqrt{1+V^2+W^2}$ and two sets of initial conditions :

$$U_0^+(x,y) = +\sin\frac{\pi}{2}(x^2+y^2)$$

$$U_0^-(x,y) = -\sin\frac{\pi}{2}(x^2+y^2).$$

Using U_0^-, the solution $U(x,y,t)$ is smooth for all time. Starting with U_0^+, the solution develops a singularity point at the origin with respect to its first derivatives after some finite time $t \geq T_0$ from a characteristic converging feature.

Numerically, U_h is compared to the exact solution :

$$U(X(r,t),t) = U_0^{\pm}(r) + t(H_r(DU_0^{\pm}(r))DU_0^{\pm}(r) - H(DU_0^{\pm}(r)),$$

where $r = \sqrt{X^2+Y^2}$, $U_0^{\pm}(r) = \pm \sin \frac{\pi}{2} r^2$, and $H(DU_0^{\pm}(r)) = H(\pm \pi r \cos \frac{\pi r^2}{2})$. Computed order of accuracy for second and third order accurate methods are given in table 1.

Scheme	L^1-norm	L^∞-norm
(2-2)ENO - FET	2.00	1.45
(3-3)ENO - FET	3.81	3.35

Table 1.

The kink does not influence the performance of the full scheme because it is precisely located at a vertex of the grid TI_h ($h = 1/10$). Comparison with first order methods have been done. Important improvements at the location of the singularity are obtained. The kink is sharper and its location very accurate even if it is not located on a grid point. Moreover, no significant improvement on the shape of the full solution is obtained from second to third order accurate approximations except at regions of smoothness in which the expected overall accuracy is attained. CPU times for these two methods were $1.05s$ and $3.65s$ for a $20 * 20$ grid per iteration for second and third order accurate methods on a sparc station, respectively. The CFL coefficient was set to 0.5 with a maximum theoretical value of $\frac{1}{\sqrt{2}}$.

3 General Grid - The Mixed Formulation

The domain Ω is now non uniformly triangulated by TB_h. We investigate the model problem given by the equations (1,2,3) with either Dirichlet or Neumann boundary conditions. For simplicity, we assume that Ω is the unit disc.

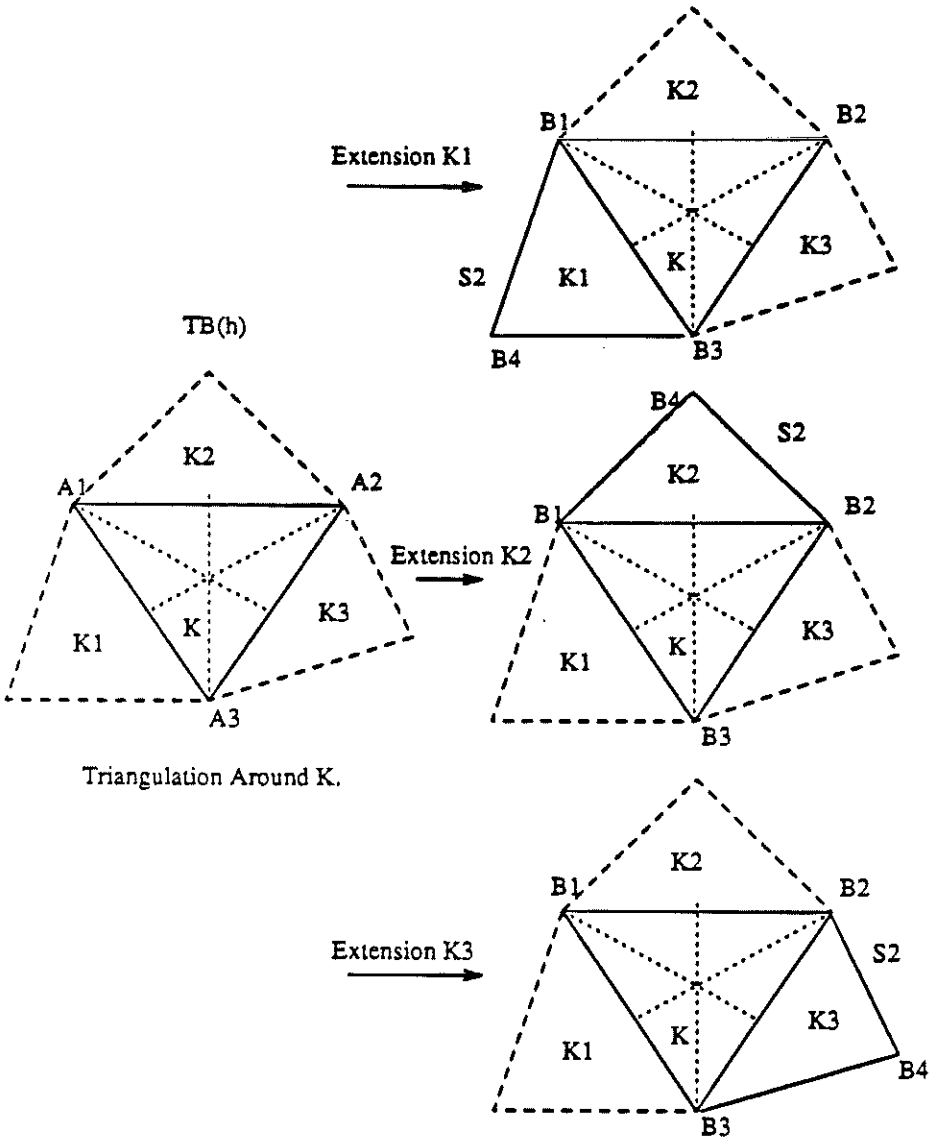
To compute an accurate numerical approximation of this problem, we follow the algorithm stated in the introduction. Each step is investigated in forthcoming subsections.

3.1 Preprocessing step - ENO Polynomial

This section is devoted to computing some ENO polynomials in each triangle K of TB_h. The basic construction is done again for simplicity in the reference element $\hat{K}$ for any K. For further simplifications, the vertices of K are denoted as before by (A_1, A_2, A_3), and the P_1 polynomial on $\hat{K}$ is simply the one defined by equation (5). Next, we construct the $\hat{P}_2$ ENO polynomial by adding second degree terms to $\hat{P}_1$. We obtain the representation :

$$\hat{P}_2(\hat{x},\hat{y}) = \hat{P}_1(\hat{x},\hat{y}) + a_1 \frac{\hat{x}(\hat{x}-1)}{2} + b_1 \hat{x}\hat{y} + c_1 \frac{\hat{y}(1-\hat{y})}{2}, \tag{8}$$

where (a_1, b_1, c_1) are free parameters. To find the unique second degree polynomial, we choose the known values of the gradient (V, W) at the center of K, and a pointwise value of U_h at a vertex B_l, $l = 4,5,6$ which is in the vicinity of either one of the three edges of K (figure 5).



Triangulation Around K.

Figure 5.

The unknown coefficients directly follows if we first write (a_1, c_1) as a function of b_1 :

$$\begin{aligned} a_1 &= 6(U_1 - U_3) + 2b_1 - 6V_K(x_1 - x_3) - 6W_K(y_1 - y_3) \\ c_1 &= 6(U_2 - U_3) + 2b_1 - 6V_K(x_2 - x_3) - 6W_K(y_2 - y_3), \end{aligned}$$

where (x_j, y_j) are the coordinates of A_j, $j = 1, 2, 3$; and finally solve for b_1. Consequently, three possible set of coefficients (a_1, b_1, c_1) can be obtained. Again, we choose as ENO polynomial the one that minimizes some combination of second derivatives [1]. The corresponding extended ENO stencil is denoted by S_2. Note that the stencil S_2 is no longer a triangle but a simply connected region with boundary points (B_1, B_2, B_3, B_4) (refer to figure 5 for the notations).

To construct a third degree polynomial, we again use $\hat{P}_2$ and its support S_2. We require :

$$\hat{P}_3(\hat{x}, \hat{y}) = \hat{P}_2(\hat{x}, \hat{y}) + a_2 \frac{\hat{x}(\hat{x} - 1)(\hat{x} - \hat{x}_4)}{3} + b_2 \frac{\hat{y}(\hat{y} - 1)(\hat{y} - \hat{y}_4)}{3} + c_2 \frac{\hat{x}(\hat{x} - \hat{x}_4)\hat{y}}{2} + d_2 \frac{\hat{x}\hat{y}(\hat{y} - \hat{y}_4)}{2}, \qquad (9)$$

where (a_2, b_2, c_2, d_2) are unknown coefficients. To define them, we use the values of the gradient in S_2 (figure 6), and write two of these coefficients with respect to the others. Finally, we finish the computations

by taking pointwise numerical values of U_h at two neighboring vertices to two edges of S_2 (figure 6). This leads to six possible set of coefficients. Again, as third degree ENO polynomial, we take the one that minimizes some elliptic combination of third derivatives [1]. The new extended stencil is denoted by S_3 and its boundaries are defined by six vertices and four triangles.

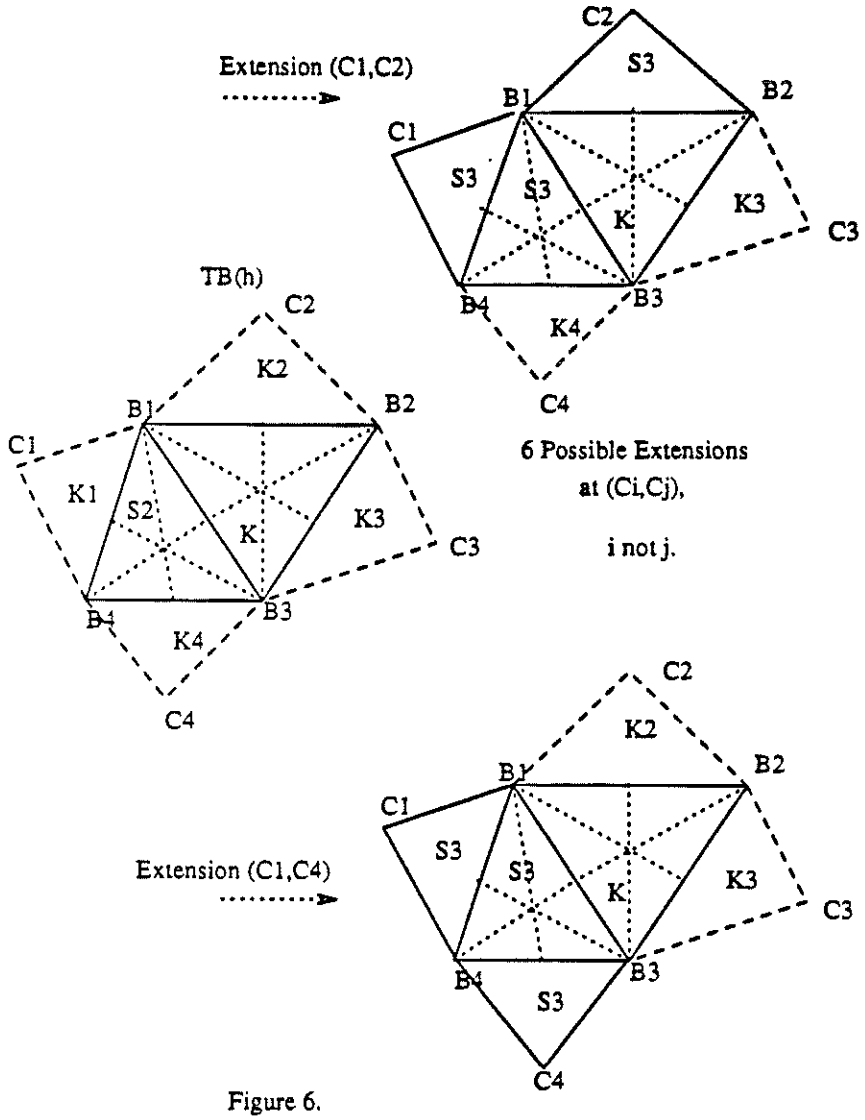
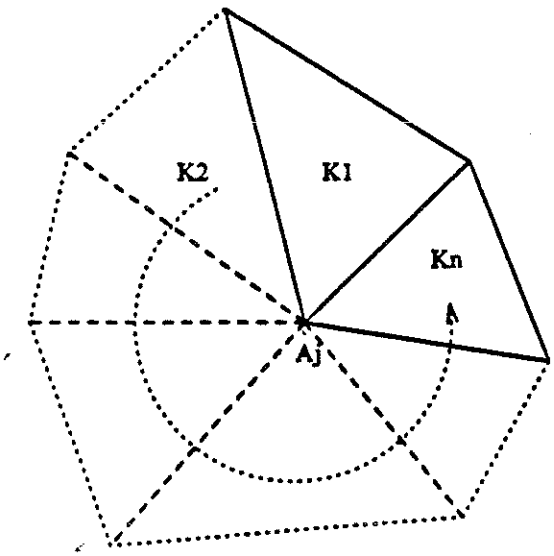


Figure 6.

This process can be repeated to construct polynomials of higher degree. Of course, the construction of such polynomials requires more and more work. However, note that only one unknown coefficient has to be determined to construct a fourth degree ENO polynomial.

3.2 Evolution operator

We implement the evolution operator defined in section 2.2. The main difference results in the triangle numbering. So, if we assume that A_j belongs to $n(j)$ triangles, then the values of $(V_j^{\pm}, W_j^{\pm})$ are computed among the $n(j)$ possible ENO polynomials whose original stencils (triangles $K_1, ..., K_{n(j)}$) are located in the appropriate half planes. This situation is briefly sketched in figure 7.



n Triangles about the vertex Aj.

Figure 7.

Hence, by defining the sets

$$K_x^{\pm} = \left\{K = (A_1, A_2, A_3) \in K_1, ..., K_{n(j)} \text{ so that } x_{A_1}, x_{A_2}, x_{A_3}(\geq) \quad (\leq) x_j\right\},$$
$$K_y^{\pm} = \left\{K = (A_1, A_2, A_3) \in K_1, ..., K_{n(j)} \text{ so that } y_{A_1}, y_{A_2}, y_{A_3}(\geq) \quad (\leq) y_j\right\},$$

we compute approximate values of the $\pm$ gradient vector according to the location of A_j as follows :

$$V_j^{\pm} = \min_{\{m^{th} \text{ derivatives}\}} \frac{\partial P_m^{K_x^{\pm}}(x_j, y_j)}{\partial x}$$
$$W_j^{\pm} = \min_{m^{th} \text{ derivatives}} \frac{\partial P_m^{K_y^{\pm}}(x_j, y_j}{\partial y}.$$

Note however that other choices for computing stable derivatives can be implemented. Again the main ENO property is satisfied since at each time step the least oscillatory polynomial among the predefined ENO polynomials on $K_{x,y}^{\pm}$ is chosen. Moreover, the ENO polynomials ensure quasi monotone variations within their ENO stencils as long as no jump in U_h appears along the calculations. Therefore, no spurious oscillations should be detected.

Finally, in order to advance the solution in time, we choose the high order Runge-Kutta time discretization schemes introduced in section 2.2.

3.3 Postprocessing Step: Evolution of the gradient vector (V_k, W_k)

The gradient vector (V_k, W_k) satisfies the system of non strictly hyperbolic equations (2,3) introduced earlier. In addition, Dirichlet boundary conditions are taken if the initial Hamilton-Jacobi equation assumes Neumann boundary conditions. Neumann boundary condition can be formulated as follows : if $H_V(\partial\Omega)N_x + H_W(\partial\Omega)N_y < 0$, then we must prescribe some boundary conditions. In the other case free outflow conditions can be implemented. Here (N_x, N_y) is the outward normal to $\partial\Omega$. Consequently,

upwinding schemes should be implemented with the appropriate boundary treatment. Remark that for consistency, we require that $\frac{\partial V}{\partial y} = \frac{\partial W}{\partial x}$ strongly if $U \in C^2(\Omega)$, or weakly in the distribution sense otherwise. Therefore, the numerical scheme used to solve the above system must define a sequence of solutions satisfying such a consistency relation up to some order. In next section will shall indicate that a positive result can be obtained if the solution is smooth.

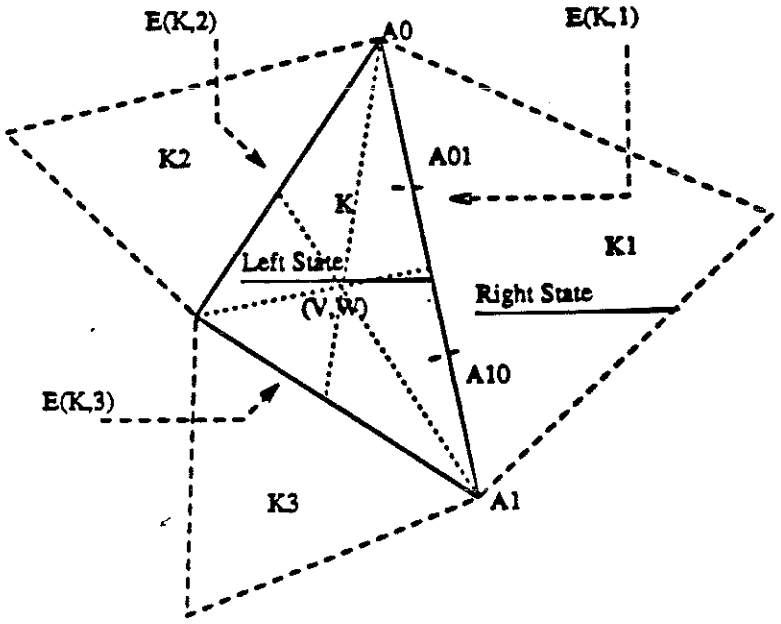
To construct a numerical approximation of (V_K, W_K), we integrate (2) and (3) over K and make use of volume averages. For example, integrating (2) over K and using the divergence theorem, we obtain :

$$\begin{aligned} \frac{\partial}{\partial t}\Big(\int_K V d\gamma\Big) &= -\int_{\partial K} H(V,W).N_x d\sigma \\ &= -\sum_{i=1}^{3} \int_{E(K,i)} H(V,W).N_x^i d\sigma, \end{aligned} \tag{10}$$

where $(E(K,1), E(K,2), E(K,3))$ are the three edges of K, and N_x^i is the x component of the outward unit normal to the edge $E(K,i)$. The right hand side integrals over $E(K,i)$ are evaluated by a quadrature formula to the same order of accuracy as the full ENO method. Therefore, for second order accurate approximation, we choose either the midpoint formula or the trapezoidal rule. For third order accurate method, either a Gauss node quadrature or the so called "three eight rule" described in (12) is implemented. The left hand side of (10) is the volume average of V on K and is denoted by V_K. Introducing this notation, V_k satisfies :

$$\begin{aligned} Surf(K)\frac{\partial V_K}{\partial t} &= -\sum_{i=1}^{3} |E(K,i)| H_q^{RF}(\partial_x P_m^r(\sigma_i), \partial_x P_m^l(\sigma_i) \\ &\quad \partial_y P_m^r(\sigma_i), \partial_y P_m^l(\sigma_i)) N_x^i, \end{aligned} \tag{11}$$

where $surf(K)$ is the surface of the triangle K ($surf(K) = \frac{1}{2}|\vec{A_1A_3} \wedge \vec{A_2A_3}|$), $|E(K,i)|$ is the length of the i^{th} edge of K, and $P_m^{r,l}(\sigma_i)$ are the ENO polynomials defined in section 3.1 taken at a quadrature point σ_i, and are constructed in regions containing either the right or left triangles from each side of the edge $E(K,i)$, i.e $E(K,i) = E(K_r,i) = E(K_l,i)$. The reader can refer to figure 8 for the notation.



Gradient Computation - Local Riemann Problem.

Figure 8.

The numerical flux $H_q^{RF}(V_r, V_l, W_r, W_l)$ is the one defined in [16]. Here "RF" stands for the "Roe-Fix" scheme, and "q" for the "quadrature" rule which is taken for approximating the integrals along the edges. The numerical flux H^{RF} uses a Local Lax-Friedrichs approximation at sonic points and a Roe upwinding scheme otherwise. More details can be found in [16]. As example of quadrature formula, we give the interesting fourth order three eight rule :

$$\begin{aligned} |E(K,i)|H_3^{RF}(V_r, V_l, W_r, W_l) &= \frac{3}{8}|E(K,i)|(H^{RF}(A_0^i) + 3H^{RF}(A_{01}^i) + \\ &\quad 3H^{RF}(A_{10}^i) + H^{RF}(A_1^i)), \end{aligned} \tag{12}$$

where $H^{RF}(A_{ln}^i)$, for $l, n = 0, 1$, $i = 1, 2, 3$, are the numerical fluxes evaluated at some vertices A_{ln}^i (figure 8).

The numerical time evolution scheme is the Runge-Kutta multistage scheme defined in section 2.2. The final time evolution operator is conservative

$$\sum_K surf(K) V_K^{n+k/m} = \sum_K surf(K) V_K^{n+l/m} \text{ for any } 0 \leq l \leq k-1, \forall n \geq 0, \tag{13}$$

and the numerical approximation of V_K^n is "weakly" accurate of order m whenever a m^{th} order quadrature formula and a m^{th} R-K multistage scheme are implemented. Finally, we notice that the derivation of the W_K approximate equation runs along the same lines by integrating (3) over K.

3.4 Consistency $\partial_{x,y} U = \partial_y V = \partial_x W$ for smooth solutions U

In this section, we prove that the consistency relation between the Hamilton-Jacobi equation (1) with the system (2) and (3) is satisfied under sufficient smoothness conditions for the exact solution U. Our

main result shows that $|\frac{\partial V_K(T)}{\partial y} - \frac{\partial W_K(T)}{\partial x}| \leq CTh^\beta$, for some $0 < \beta < 1$ and some $h < h_0$ small enough. In order to obtain such a result, we first state a few lemmas in which we assume $U \in C^2(\Omega)$. Moreover, first order space discretizations only are investigated, and a semi discrete time appoach is used for further simplicity.

Lemma 3.1 *If we let* $J_K(t) = \frac{\partial V_K(t)}{\partial y} - \frac{\partial W_K(t)}{\partial x}$, *then :*

$$\frac{\partial J_K(t)}{\partial t} = -\sum_{i=1}^{3} \frac{|E(K,i)|}{surf(K)}.H'(N^\star) < \partial^2_{x,y}P(n_x^2 - n_y^2) + (\partial^2_{y,y}P - \partial_{x,x}P)n_x n_y, \delta_{\sigma_i} >, \tag{14}$$

where $< .,. >$ *is the duality product,* P *is the interpolating polynomials defined either in* K_l^i *or in* K_r^i, (n_x, n_y) *is the unit outward normal to* $E(k,i)$ *at the point* σ_i, $N^\star = (V^\star, W^\star)^t.(n_x, n_y)^t$, *and the superscript* $\star$ *represents either the left or right state across* $E(K,i)$.

Proof : The proof can be done within a few lines provided some preliminary assumptions are made. First of all, we assume that the numerical Hamiltonian is simply defined by upwinding. The general case will be handled in one of the corollaries following theorem 3.1. On the other hand, we can suppose that $|V^l(\sigma_i) - V^r(\sigma_i)| = |\partial_x P^l(\sigma_i) - \partial_x P^r(\sigma_i)| < Ch^2$ from the smoothness property of U. Finally, we assume that $H = H(V,W)$, so that there is no direct dependency upon the space variables. Again, the general case is handled later.

Following these remarks, the proof merely consists in writing precisely $H_y.n_x - H_x.n_y$ for obtaining (14).

Lemma 3.2 *If we let* $I_K(t) = \frac{\partial V_K(t)}{\partial x} - \frac{\partial W_K(t)}{\partial y}$, *then the following equality holds :*

$$\frac{\partial I_K(t)}{\partial t} = -\sum_{i=1}^{3} \frac{|E(K,i)|}{surf(K)}.H'(N^\star) < \partial_{x,x}Pn_x^2 - \partial_{y,y}Pn_y^2, \delta_{\sigma_i} >. \tag{15}$$

Proof : The proof is similar to the one sketched for lemma 3.1 ; and thus is not duplicated here.

The next technical lemma indicates how to pass to the limit as the triangle K_h shrinks to its center σ_c (limit as $h \to 0$).

Lemma 3.3 $\forall \varphi \in C^\infty(B(\sigma_c, h))$, *there exists a regularizing sequence of functions* $\theta_\epsilon(x,y) = \frac{1}{\epsilon^2}\theta(\frac{x}{\epsilon}, \frac{y}{\epsilon})$ *so that* $\forall \varepsilon = h^{1+\alpha}$, $\frac{1}{2} > \alpha > 0$, *and* $\forall h < h_0$ *small enough, we have :*

$$|\sum_{i=1}^{3} < c_i(h)\varphi, \delta_{\sigma_i} > - < \tau_{\sigma_c}\varphi, \delta_{\sigma_c} * \theta_\varepsilon > | \leq Lh^2 \tag{16}$$

where τ_{σ_c} *is the translation operator of scale* $-\sigma_c$, *and* $*$ *is the usual convolution product.*

Proof : The proof can be decomposed into three steps : First of all, we define a sequence of cut-off functions $\theta_\varepsilon \in C^\infty(B(\sigma_c, 2h)$ which are equal to one if the distance from $A(x,y)$ to either one of the three diagonals D_i of K is less than ε, and is set to zero outside the ball $B(\sigma_c, h)$. On the other hand, we make a partition of unity of the diagonals D_i, $i = 1,2,3$, and denote the indicator functions of these sets by ϕ_i. Finally, we can write that $\theta = \sum_{i=1}^3 \theta c_i(h)\phi_i = \sum_{i=1}^3 \theta_i$ where θ_{i_ϵ} is equal to $c_i(h)$ if the distance from D_i does not exceeds ε. It is then easy to finish the proof by letting $\varepsilon = h^{1+\alpha}$; α is as defined in the lemma, and then by taking the limit as h tends to zero. Indeed, we obtain that for some $h < h_0$ (h_0 small) the left hand side of (16) is less that $Lh^2|\varphi'|_\infty$ where L is a fixed constant. This concludes the proof of lemma 3.3.

The next lemma applies the result obtained in lemma 3.3 to the relations (14) and (15).

Lemma 3.4 *$\forall h < h_0$, the following inequalities hold :*

$$|\frac{\partial J_K(t)}{\partial t}| \leq \frac{1}{surf(K)}|\sum_{i=1}^{3} < c_i(h) f_i, \delta_{\sigma_i} >| \leq L_1 h^{1-2\alpha}|\sum_{i=1}^{3} f_i(\sigma_c)| \tag{17}$$

$$|\frac{\partial I_K(t)}{\partial t}| \leq \frac{1}{surf(K)}|\sum_{i=1}^{3} < c_i(h) g_i, \delta_{\sigma_i} >| \leq L_2 h^{1-2\alpha}|\sum_{i=1}^{3} g_i(\sigma_c)|, \tag{18}$$

where $f_i = \partial_{x,y} P(n_x^2 - n_y^2) + (\partial_{y,y} P - \partial_{x,x} P) n_x . n_y$, and $g_i = \partial_{x,x} P n_x^2 - \partial_{y,y} P n_y^2$.

Proof : From lemma 3.3, we construct a C^∞ function φ on the ball $B(\sigma_c, 2h)$ which is equal to f_i on K and is extended to zero outside the ball $B(\sigma_c, h)$. In order to finish the proof, we just remark that $surf(K) = \frac{1}{2}|\vec{A_1 A_3} \wedge \vec{A_2 A_3}| = Ch^2$ which is combined to the inverse power of $\varepsilon^2 = h^{2-2\alpha}$ in order to pass to the limit over h.

We are now ready to state and prove our main result.

Theorem 3.1 *Under the assumptions of previous lemmas, and $\forall h < h_0$, we have :*

$$|J_K(t)| \leq Cth^{1-2\alpha}, \tag{19}$$

where C is a constant, and α is the parameter defined in lemma 3.3.

Proof : From lemma 3.4, it is easy to check that these two inequalities

$$|\frac{\partial I_K(t)}{\partial t}| \leq C_1 h^{1-2\alpha}|I_K(t)|, \tag{20}$$

$$|\frac{\partial J_K(t)}{\partial t}| \leq h^{1-2\alpha}(C_2|I_K(t)| + C_3|J_K(t)|), \tag{21}$$

are satisfied $\forall h < h_0$ and for the set of positive constants C_1, C_2, and C_3. The remainder of the proof consists in using a Gronwall lemma twice (first in (20), and then in (21)) to show that :

$$|I_K(t)| \leq \exp(C_1 h^{1-2\alpha})|I_K(0)|$$

$$|J_K(t)| \leq \exp(C_3 h^{1-2\alpha})|J_K(0)| + C_2 t h^{1-2\alpha} \exp(C_1 h^{1-2\alpha})|I_K(0)|.$$

Indeed, if we assume that the initial condition is consistent, i.e if $|J_K(0)| \leq C_4 h$ and if $|I_K(0)| \leq C_5$, for some additional nonnegative constants C_4 and C_5, we prove that (19) is verified.

We now extend the result of theorem 3.1 to the Roe-Fix approximation of the numerical Hamiltonian H. In this case, we consider the Lax-Friedrichs approximation of the Hamiltonian: $H^{LF}(N^r, N^l) = H(\frac{N^r+N^l}{2}) - \sup_{N \in I(N^r,N^l)} |H'(N)|\frac{N^r-N^l}{2}$, where $I(N^r, N^l)$ is the closed curved passing through N^r and N^l (convex case).

Corollary 3.1 *if H^{RF} is taken instead of H^G, then the consistency result of theorem 3.1 still holds.*

Proof : The proof is obvious as long as we notice that $|N^r(\sigma_i) - N^l(\sigma_i)| \leq Ch^2$, for $i = 1, 2, 3$. Consequently, terms of order h must be added to the previous proofs.

The next corollary generalizes that framework to rather general Hamiltonians of the form $H(V, W, x, y) = H_1(V, W) + I(x, y)H_2(V, W)$, where $I \in C^k(\Omega)$, for $k \geq 1$.

Corollary 3.2 *if the numerical Hamiltonian is defined by $H_1^{RJ}(V_i, W_i) + I(\sigma_i)H_2^{RF}(V_i, W_i)$, where (V_i, W_i) are constructed from some interpolating polynomials at points σ_i, $i = 1, 2, 3$; then the conclusion of theorem 3.1 still holds.*

Proof : The proof of theorem 3.1 can be successively applied to $\frac{\partial H_1}{\partial y} n_x - \frac{\partial H_1}{\partial x} n_y$, $|I|_\infty (\frac{\partial H_2}{\partial y} n_x - \frac{\partial H_2}{\partial x} n_y)$, and $H_2 |\frac{\partial I}{\partial y} n_x - \frac{\partial I}{\partial x} n_y|_\infty$. From the smoothness properties of U and I, three similar final results than the one of theorem 3.1 can be respectively obtained.

Theorem 3.1 indicates that the mixed formulation is consistent provided the solution of (1) is sufficiently smooth. However, if U does not have enough smoothness properties, then all of our numerical experiments have shown that the constructed solutions were accurate and stable. The next two sections investigate several examples for which consistent and stable high order numerical solutions are obtained.

3.5 Numerical Application

We consider the model problem defined by (1,2,3) in which $H = -\sqrt{1 + V^2 + W^2}$ and Ω is the unit disc. The domain is triangulated by NT triangles and NV vertices in an arbitrary way. An example of such a triangulation is displayed in figure 3a. The boundary conditions are of Neumann type and for further simplicity, we assume that $(V_{\partial\Omega}, W_{\partial\Omega}) = (0, 0)$. As initial conditions, we use $U_0^{\pm}$ (section 2.3) and run both second and third order ENO schemes. The results are plotted in the set of figures 3b (Second Order ENO), and 3c (Third order ENO). The location of the kink is well located and no visible improvements can be seen from the second to the third order accurate methods. Moreover, table 2 shows that the order of accuracy in smooth transition areas is what we have expected when $\Omega = [-1, 1]^2$ is triangulated using a uniform cartesian grid.

Scheme	L^1-norm	L^∞-norm
(2-2)ENO - MF	2.02	1.62
(3-3)ENO - MF	2.73	2.89

Table 2.

CPU times for second and third order accurate methods were 1.49s, and 4.95s, respectively. These running times can be compared to those established in section 2.3. A speed up of about 1.5 to 2 is gained by using our FET method. This result certainly agrees with the complexity of the mixed formulation. Therefore, the mixed formulation should be used only if necessary. In the next section, we investigate the interface problem by implementing both techniques whenever the two types of grids are mixed together.

4 Interface Problem : $\Omega = TI_h \cup TB_h$:

As test problem, we choose to let TI_h triangulate the interior part of Ω and TB_h triangulate $\Omega - TI_h$ by matching the interfaces of $\partial\Omega$ and ∂TI_h. The main algorithm runs along the lines :

- 1.) Construct a square or a polygonal (vertical and horizontal polygonal line only) grid TI_h that fills in partially the domain Ω,
- 2.) Define the grid TB_h whose vertices are located in between the boundaries of Ω and TI_h,
- 3.) Run the code with additional care at the intersection curves of TI_h and TB_h.

Some numerical experiments have shown that the solution was stable provided that we used the mixed formulation for all triangles having at least one vertex at the intersection of both grids. If not, some instabilities occured at corners of the square grid if a kink developed nearby. This phenomenon can be easily explained : at a corner of an interface, one or no extension from first to second or from second to third order ENO stencils can be obtained in TI_h since vertices of TB_h do not necessarily belong to the

preset uniform square grid. Figure 9 briefly sketches the interface problem and localizes the first "slice" of interface triangles of TI_h.

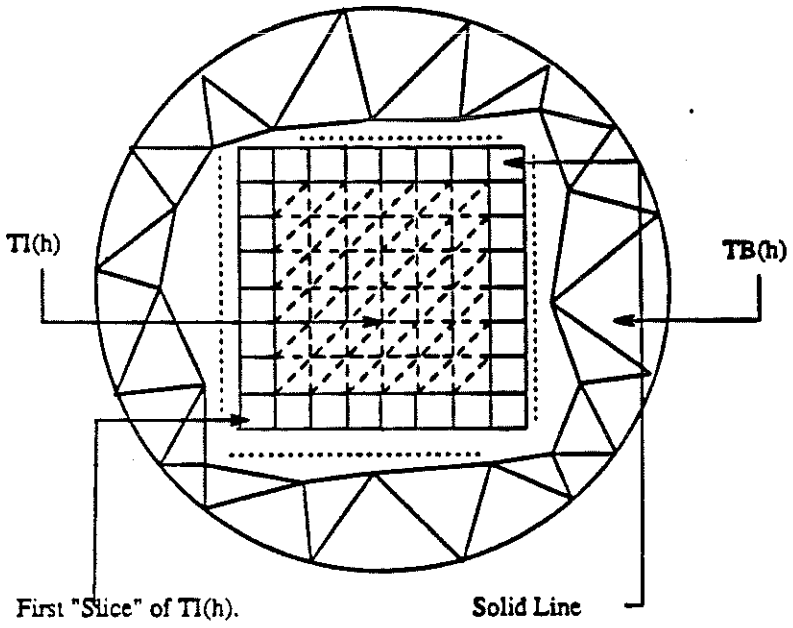


Figure 9.

The numerical experiment of section 3.5 has been implemented. Ω is still the unit disc and TI_h is the square grid of 10 by 10 vertices in $[-1/2, 1/2]^2$. Numerical results are plotted in the set of figures 4b and 4c for the second order accurate ENO method. Note that no numerical instabilities can be detected at the square interface. The total number of vertices (NV) and triangles (NT) of TB_h were 438 and 748, respectively. The CFL coefficient was set to $\frac{1}{2}$ in this experiment.

5 Numerical Experiments

Now, we investigate various applications that have been studied in [11, 12]. We will particularly focus on an optimal control-cost determination and a combustion problem with curvature dependency. In addition, we study a shape reconstruction from shading problem introduced by P.L. Lions during his stay at the University of California, Los Angeles in April, 1990. See [14] for the rigorous analysis of this problem.

5.1 Optimal Control Problem

We solve the optimal control problem of example 4 in [12]. This problem is related to control-optimal cost determination. It can be modeled by the Hamilton-Jacobi equation :

$$\begin{aligned} U_t + \sin y U_x + (\sin x + sign(U_y))U_y - \frac{1}{2}\sin^2 y - (1 - \cos x) &= 0 \\ U(x, y, 0) &= 0, \end{aligned}$$

to be solved in $[-\pi/2, +\pi/2]^2$. We assume periodic boundary condition and stop the iterative process as soon as the steady state solution is reached. Numerical results are displayed in figure 5a at time $t = 6.2$

for a CFL coefficient of 0.4. In the lower half of figure 5a, we plot the optimal solution $\omega = sign(U_y)$. This numerical test has been implemented for our second order method using FET.

5.2 Combustion problem with curvature dependency

We solve the regularized Hamilton-Jacobi equation : [11]

$$\begin{aligned} U_t + F(K)H(U_x, U_y) &= 0 \\ U(x, y, 0) &= \cos 2\pi x - \cos 2\pi y, \end{aligned} \tag{22}$$

in which $F(K) = 1 - \varepsilon K$, K is the mean curvature, and $H(U_x, U_y)$ is the Hamiltonian defined in section 3.1. As computational domain, we consider the square $[0, 1]^2$, and use our FET method on TI_h. We run two examples with $\varepsilon = 0$ and $\varepsilon = 0.1$. Numerically, we follow the framework introduced in [11] by splitting the space differencing into two parts. Namely, by writing $F(K) = F_1 + \varepsilon F_2(K)$, we compute the convective part $F_1 H(U_x, U_y)$ as before, while the second order term $\varepsilon F_2(K)H(U_x, U_y)$ is approximated by centered differences.

The results for $\varepsilon = 0$ are displayed in figure 5b using our second order ENO method. The solution has its first derivative discontinuous along the line $x = \frac{1}{2}$. Numerical experimentation gives the exact location of the discontinuity after 100 iterations for a CFL coefficient of 0.3. Furthermore, the discontinuity is sharply resolved, and no diffusion can be observed. No visible improvements in the shapes were obtained by using our third order accurate ENO scheme. However, the overall accuracy is improved in smooth regions. In figure 5c, we plot the results after 100 iterations for $\varepsilon = 0.1$. The singularity is smoothed out under the effect of curvature. We noticed that mixed second derivatives of the interpolating polynomials were always zero in smooth transition areas. This result is in agreement with the initial condition for which $\frac{\partial^2 U(x,y,0)}{\partial x \partial y} = 0$. Therefore, dimension by dimension algorithms and two dimensional methods are equivalent in this case.

5.3 Computer Vision Problems

In this section, we study the nonlinear Hamilton-Jacobi equation :

$$\begin{aligned} U_t + I(x, y)\sqrt{1 + U_x^2 + U_y^2} - (\alpha U_x + \beta U_y + \gamma) &= 0 \\ U_{\partial\Omega} &= 0 \\ U(x, y, 0) &= 0. \end{aligned} \tag{23}$$

The problem is to find the shape of a region $U(x, y, \infty)$ lighted by a source located at infinity with direction $\vec{D} = (\alpha, \beta, -\gamma)$. The normal to the surface is defined by $\vec{N} = (U_x, U_y, -1)$ and the intensity function $I(x, y)$ is defined by $I(x, y) = \cos(\vec{N}, \vec{D})$. Next, if we assume that $\vec{D}$ is a unit vector, then the intensity function can be rewritten as follows :

$$I(x, y) = \frac{\alpha U_x(x, y, \infty) + \beta U_y(x, y, \infty) + \gamma}{\sqrt{1 + U_x(x, y, \infty)^2 + U_y(x, y, \infty)^2}}.$$

In the numerical experiments 1 and 2, we consider horizontal light, e.g $\alpha = \beta = 0, \gamma = 1$. In experiment three, the source of light has a direction $(\pi/10, \pi/10)$ as measured from the vertical axis. Therefore, the corresponding direction angles are defined by $\alpha = cos(\pi/2 - \pi/10) = \beta$, and $\gamma = \sqrt{1 - \alpha^2 - \beta^2}$.

- 1.) Reconstruction of a smooth function $V(x,y) = U(x,y,\infty) = (1-x^2)(1-y^2)$: defining the intensity function $I(x,y) = \frac{1}{\sqrt{1+4x^2(1-y^2)^2+4y^2(1-x^2)^2}}$, we discovered that the solution of (23) was not unique. This fact can be easily explained by writing the steady state solution of (23) at the set of points for which the intensity function is set to one. We obtain that $|\nabla U(x,y,\infty)| = 0$; thus, any constant is solution. Of course the final desired solution must be the viscosity solution. The interesting reader can get a detailed presentation of viscosity solutions of Hamilton-Jacobi equations in [2], [3], and [9]. The analysis of viscosity solutions for shape from shading problems appears in [14]. Numerically, two "tricks" were investigated as suggested by Lions and by the work of Rouy and Tourin [14] : either we prescribed the value of $U(x,y,\infty)$ at the set of points for which the intensity is one ; or we multiply this intensity function by $1-\varepsilon$. In the second case, $0 \leq (1-\varepsilon)I(x,y) \leq 1$, and the solution $U_\varepsilon(x,y,\infty)$ is expected to converge in some sense to $U(x,y,\infty)$. Both approaches have been investigated and numerical results are presented in the set of figures 5d and 5e ($\varepsilon = h/5$) for our second order ENO method. We used the grid TI_h for $h = 2/10$, and the CFL coefficient was set to 0.7 (maximum value $\frac{1}{\sqrt{2}}$). The steady state solution was always reached before 800 (this number increased drastically as ε tends to zero) iterations for a CPU time of 0.6 second per iteration.

- 2.) Reconstruction of the piecewise linear function $U(x,y,\infty) = (1-|x|)(1-|y|)$: We use the same numerical method as in 1.) without particular attention to the set of points for which the intensity function is one, i.e. $I(x,y) = \frac{1}{\sqrt{1+(1-|x|)^2+(1-|y|)^2}} = 1 <=> (x,y) = (0,0)$. The solution is well reconstructed and the steady state solution develops sharp discontinuous first derivatives along the lines $x = 0$ and $y = 0$. Figures 5f and 5g display the solution after 100 and 30 iterations for the second order accurate method using our FET method and the mixed formulation with a CFL condition of 0.7. Next, we modify the intensity function $I(x,y)$ by simulating noise $N(x,y) = (1-\theta W(x,y))$, where $W(x,y)$ is the highly oscillatory function $5\cos\frac{2\pi x}{h}\cos\frac{2\pi y}{h}$. The second order ENO method using FET was implemented and the solution after 60 iterations with $\theta = 0.05$ is plotted in the figure 5h. The steady state solution is well reconstructed with the noise.

- 3.) We need reconstruct the function of example 2.) for non vertical light. Numerical results are plotted after 200 iterations using our FET method in figure 5i for $\varepsilon = 0.1$ and $\varepsilon = 0.01$. Again, the solution is well reconstructed with very sharp discontinuous first derivatives along the lines $x = 0$ and $y = 0$. Note that we were unable to compute the solution using the mixed formulation. This failure can be easily explained since the numerical approximated fluxes of the non strictly hyperbolic system have discontinuous coefficients, e.g $I(x,y) = \frac{-\alpha*sign(x)(1-|y|)-\beta*sign(y)(1-|x|)+\gamma}{\sqrt{1+(1-|x|)^2+(1-|y|)^2}}$. Therefore, a special treatment must be implemented in order to efficiently solve the Riemann problems across the lines $x = 0$ and $y = 0$.

6 Remarks and Conclusions

We have defined two new stable and computationally efficient techniques for solving non linear Hamilton-Jacobi equations on general closed domains preserving high order accurate approximate solutions. These techniques can be applied to problems arising in various fields of classical applied mathematics, optimal control, computer vision, and more.

The first advantage of our techniques relies on the global two dimensional approach which extends the supports of the ENO polynomials to a local simplex. Hence, the information taken into account to construct the final approximation does not depend only on two directions. This leads us to more sensitive algorithms. This result has been obtained without too much additional computational time compared to

cartesian based algorithms. Moreover, these techniques can be easily vectorized. However, the resulting code is of higher complexity than the "line" ENO algorithm. In addition, some additional improvements for the resolution of large transition areas can be added to the original code. For example, it is possible to apply grid refinement techniques since our data structure is based on finite elements. Also, a subcell resolution algorithm can be implemented in the case of discontinuous coefficients.

To conclude, we mention that our methods are very sensitive to the "minimizing criterion" selected during the construction of the ENO polynomials. However, this criterion can be set freely in all major applications and may depend on some constraints imposed by the physics of the problem. In all numerical experiments, either the Laplacian tests or a combination of discrete l^2 norm of derivatives were implemented. Moreover, whatever criterion we chose, no spurious oscillations were detected. This demonstrates that our methods are very robust even in the case of arbitrary grids showing that there is no a-priori dependency on the triangulation.

References

[1] Abgrall R., ICASE Report 91-84, (1991).

[2] Crandall M., P.L. Lions, Trans. Amer. Math. Soc., 277 ,1 (1983).

[3] Crandall M., P.L. Lions, Math. Comput., 43,1 (1984)

[4] Durlofsky L., B. Engquist, S. Osher, J. Comput. Phys., 98, 64 (1992).

[5] Harten A., S. Osher, S. Chakravarthy, Private communication.

[6] Harten A., B. Engquist, S. Osher, S. Chakravarthy, J. Comput. Phys., 71, 231 (1987).

[7] Harten A., S. Osher, Siam J. Numer. Anal., 24, 279 (1987).

[8] Hughes T.J.R., T. Tezduyar, Comput. Methods Appl. Mech. Engrg., 45, 217 (1984).

[9] Lions P.L, "Generalized solutions of Hamilton-Jacobi equations", Pitman (1982).

[10] Lohner R., K. Morgan, M. Vahdati, J.P. Boris, D.L. Book, Comm. In Applied Numerical Methods, 4, 717 (1988).

[11] Osher S., J. Sethian , J. Comput. Phys., 79, 12 (1988).

[12] Osher S., C-W. Shu , "High order essentially non-oscillatory schemes for Hamilton-Jacobi equations", UCLA CAM Report, # 89-32, (1989).

[13] Rostand P., B. Stoufflet, IMA Conference Series, 9, 681-700, A. Iserles and M.J.D. Powell, Eds, Clarendon Press, (1987).

[14] Rouy E., A. Tourin, "A viscosity solution approach to Shape from Shading", To appear in Siam J. Numer. Anal., 1992.

[15] Shu C-W, S. Osher, J. Comput. Phys., 77, 439 (1988).

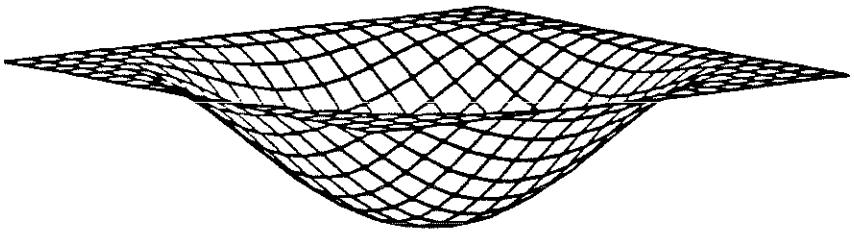
[16] Shu C-W, S. Osher, J. Comput. Phys., 83, 32 (1989).

[17] Sanders R., Math. Comp., 9, 535 (1988).

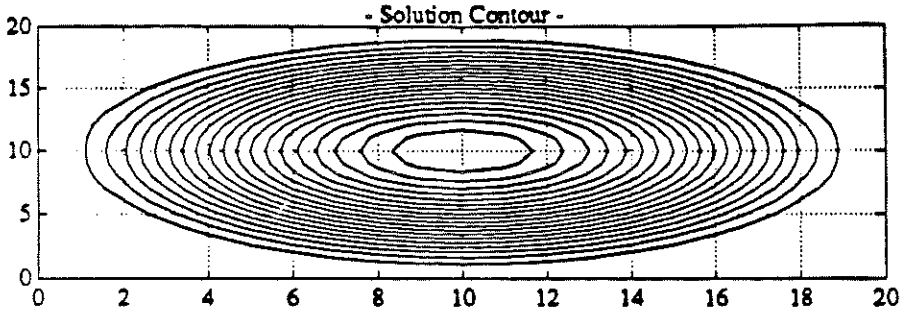
Figure Captions

- 2a) (2,2) ENO-RF (Roe Fix): Combustion Problem, Initial Condition U_0^+, $H(U_x, U_y) = -\sqrt{1 + U_x^2 + U_y^2}$, Iterations 1 and 60.
- 2b) (3,3) ENO-RF (Roe Fix): Same as example 2a) with Initial Condition U_0^+.
- 2c) (3,3) ENO-RF (Roe Fix): Same as example 2b) with Initial Condition U_0^-.
- 3a) Computational grid TB_h for the unit disc,
- 3b) (2,2) ENO-RF (Roe Fix): Combustion Problem - Mixed Formulation, Iterations 1 and 120 on grid 3a).
- 3c) (3,3) ENO-RF (Roe Fix): Combustion Problem - Mixed Formulation, Iterations 1 and 120 on grid 3a).
- 4a) Computational grid TB_h and TI_h for the unit disc - Interface problem.
- 4b) (2,2) ENO-RF (Roe Fix): Combustion Problem, Solution and contour plots at iteration 1 on grid 4a).
- 4c) (2,2) ENO-RF (Roe Fix): Combustion Problem, same as Example 4b) - Iteration 90 on grid 4a).
- 5a) (2,2) ENO-RF (Roe Fix): Control-Cost Problem - t=6.2, Optimal cost function $\omega = sign(U_y)$,
- 5b) (2,2) ENO-RF (Roe Fix): Combustion Problem with Curvature dependency, $\varepsilon = 0$, solutions at 1 and 100 iterations.
- 5c) (2,2) ENO-RF (Roe Fix): Combustion Problem with Curvature dependency, $\varepsilon = 0.1$, solutions at 1 and 100 iterations.
- 5d) (2,2) ENO-RF (Roe Fix): Vision Problem - Reconstruction of $U(x, y) = (1-x^2)(1-y^2)$, Inside Boundary Condition,
- 5e) (2,2) ENO-RF (Roe Fix): Vision Problem - Reconstruction of $U(x, y) = (1 - x^2)(1 - y^2)$, Intensity function multiply by $(1 - \varepsilon)$, $h = 0.1, \varepsilon = 0.02$,
- 5f) (2,2) ENO-RF (Roe Fix): Vision Problem - Reconstruction of $U(x, y) = (1 - |x|)(1 - |y|)$, 60 iterations,
- 5g) (2,2) ENO-RF (Roe Fix): Vision Problem - Reconstruction of $U(x, y) = (1 - |x|)(1 - |y|)$, 60 iterations - mixed formulation.
- 5h) (2,2) ENO-RF (Roe Fix): Vision Problem - Reconstruction of $U(x, y) = (1 - |x|)(1 - |y|)$ plus white noise of amplitude $\theta = 0.05$, 150 iterations.
- 5i) (2,2) ENO-RF (Roe Fix): Vision Problem - Reconstruction of $U(x, y) = (1 - |x|)(1 - |y|)$, $\alpha = \cos \pi/2 - \pi/10, \beta = \cos \pi/2 - \pi/10$, solution after 200 iterations for $\varepsilon = 0.1, 0.01$.

- Hamilton Jacobi 2D Equation Ut+H(Ux,Uy)=0 IC with IC: COS(X+Y) -

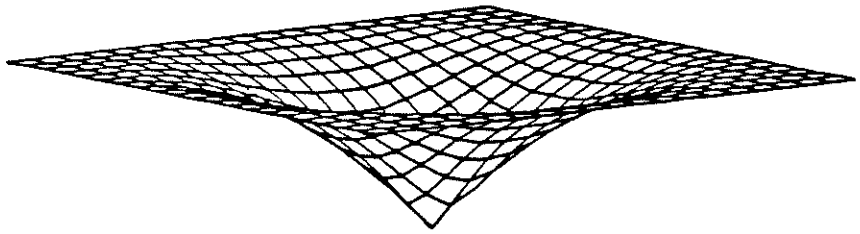


- Iteration # 1 , cfl=.5 , m=n=20 -

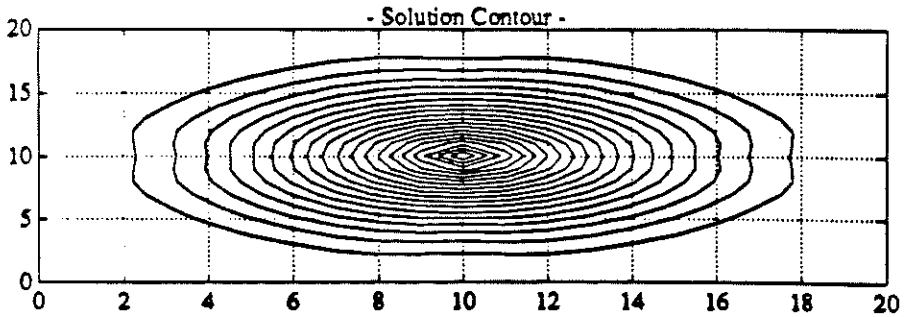


- (2,2) RF - Zmin=0.02, Zmax=1.02 -

- Hamilton Jacobi 2D Equation Ut+H(Ux,Uy)=0 IC with IC: COS(X+Y) -



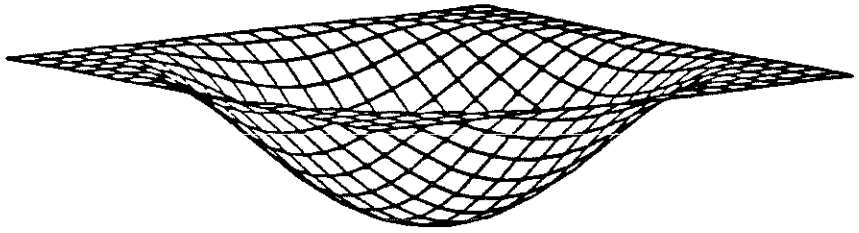
- Iteration # 60 , cfl=.5 , m=n=20 -



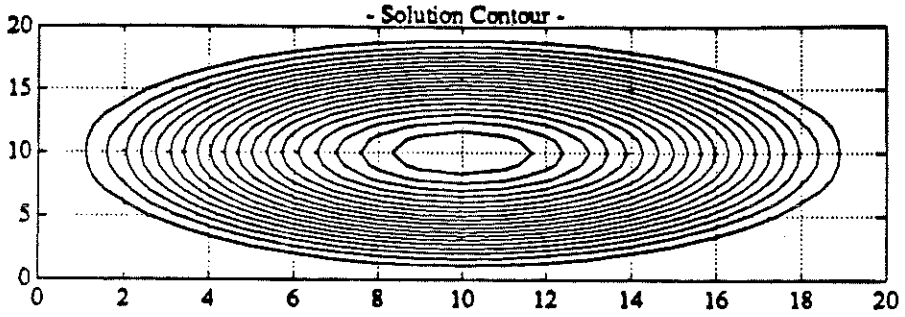
- (2,2) RF - Zmin=1.771, Zmax=2.20 -

- Figure 2a -

- Hamilton Jacobi 2D Equation Ut+H(Ux,Uy)=0 IC with IC: COS(X+Y) -

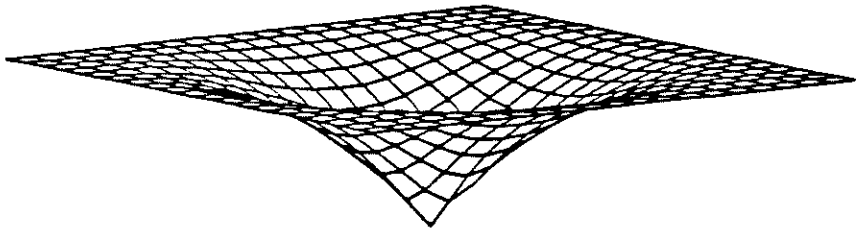


- Iteration # 1 , cfl=.5 , m=n=20 -

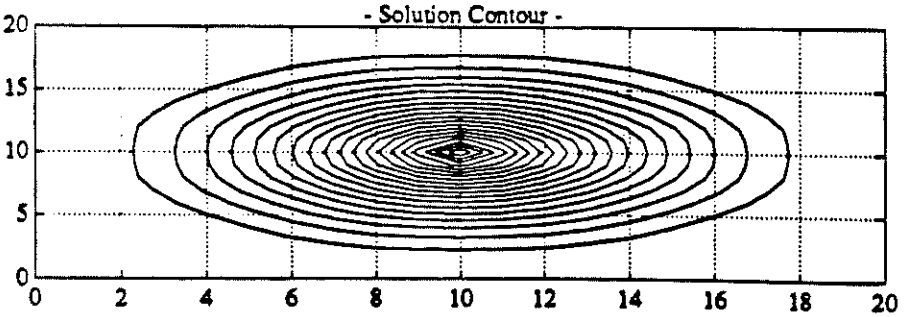


- (2,2) RF - Zmin=0.02, Zmax=1.02 -

- Hamilton Jacobi 2D Equation Ut+H(Ux,Uy)=0 IC with IC: -COS(X+Y) -



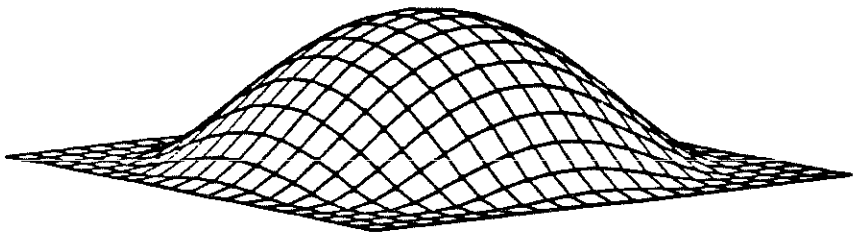
- Iteration # 60 , cfl=.5 , m=n=20 -



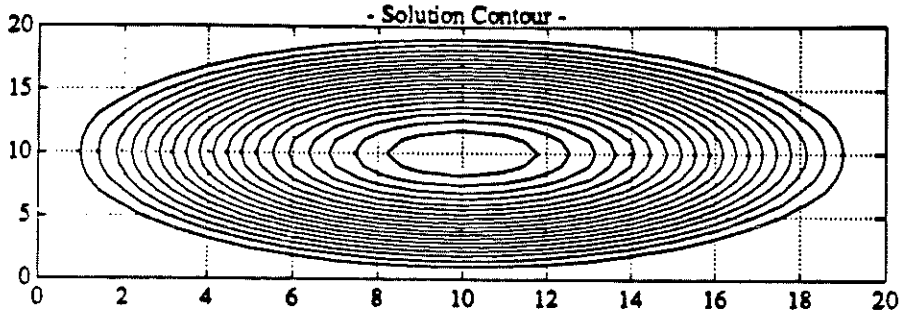
- (3,3) RF - Zmin=1.77, Zmax=2.20 -

- Figure 2b. -

- Hamilton Jacobi 2D Equation Ut+H(Ux,Uy)=0 IC with IC: -COS(X+Y) -

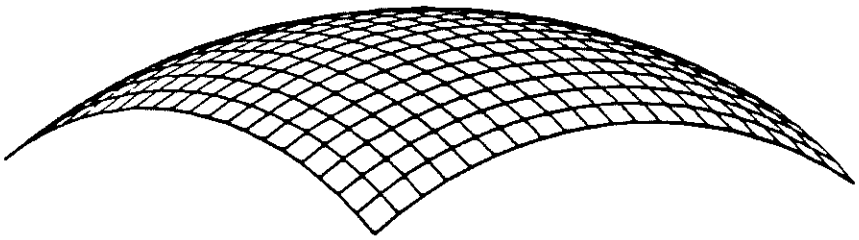


- Iteration # 1 , cfl=.5 , m=n=20 -

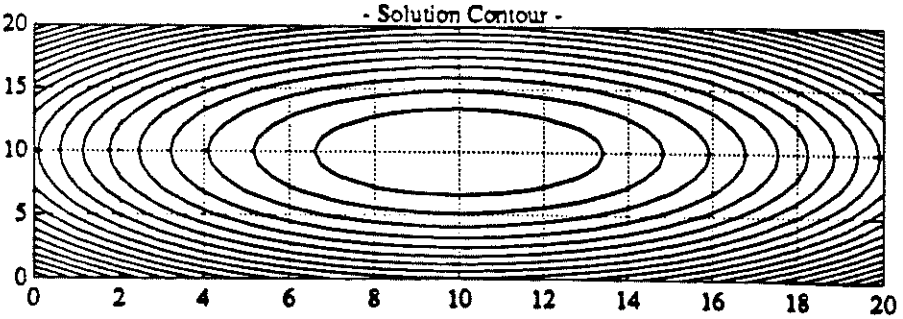


- (3,3) RF - Zmin=-0.98, Zmax=0.02 -

- Hamilton Jacobi 2D Equation Ut+H(Ux,Uy)=0 IC with IC: -COS(X+Y) -



- Iteration # 60 , cfl=.5 , m=n=20 -



- (3,3) RF - Zmin=0.42012, Zmax=1.20 -

- Figure 2c. -

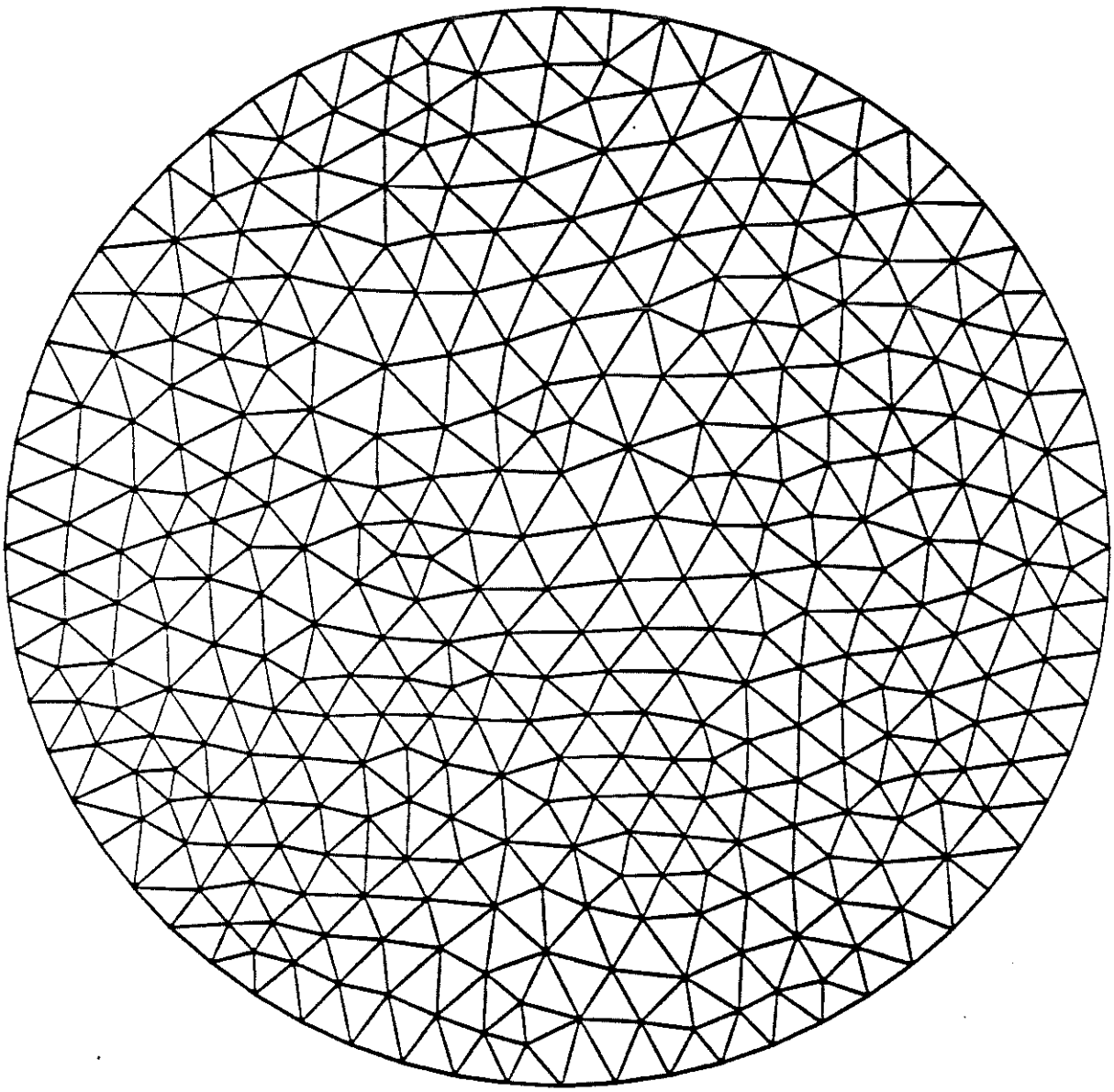
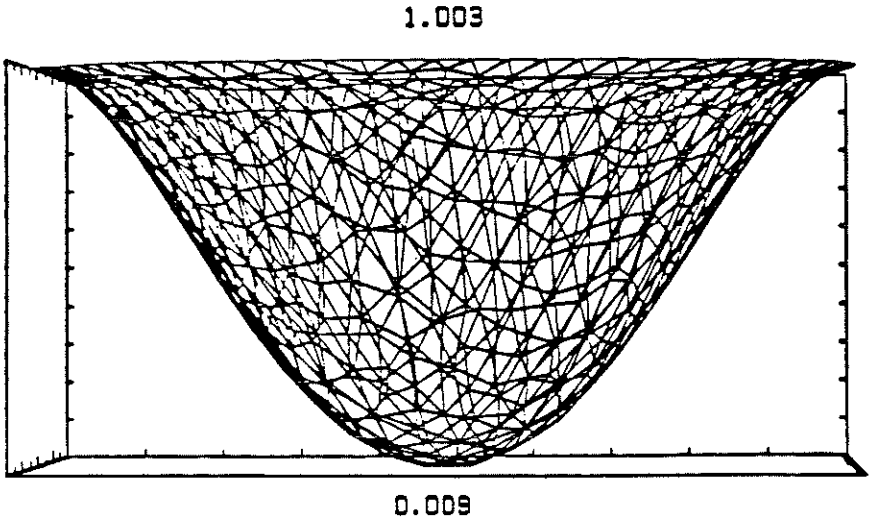


FIGURE 3.4

Two dimensional H-J using finite elements

Second Order ENO formula.

ITERATIØN NUMBER 1



Two dimensional H-J using finite elements

Second Order ENO formula.

ITERATIØN NUMBER 12

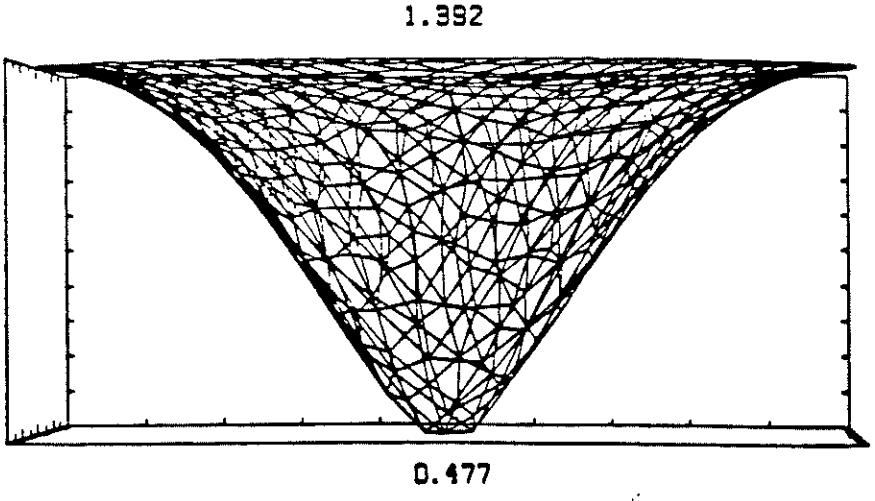
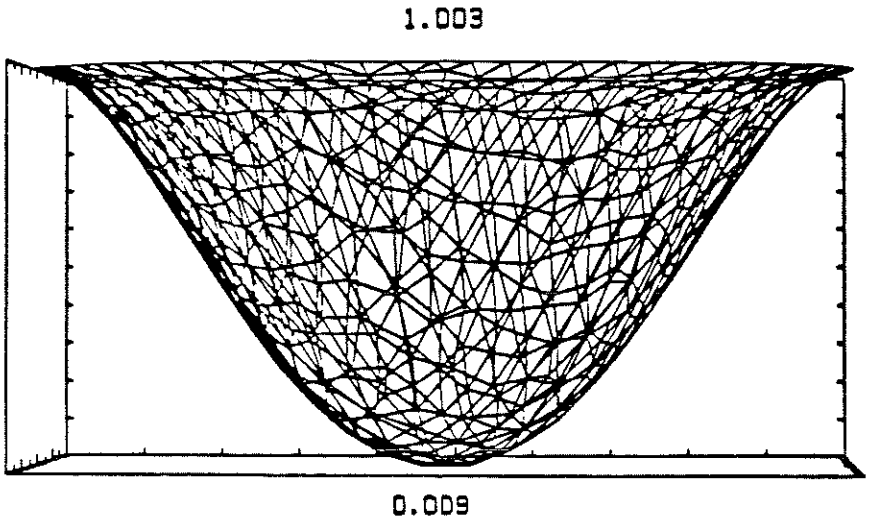


FIGURE 3.8

Two dimensional H-J using finite elements

Third Order ENO formula.

ITERATIØN NUMBER 1



Two dimensional H-J using finite elements

Third Order ENO formula.

ITERATIØN NUMBER 12

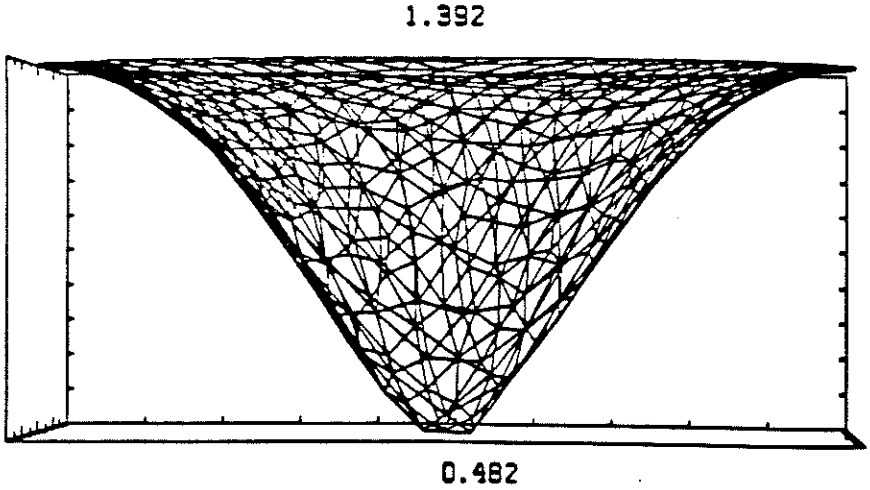


FIGURE 3.C

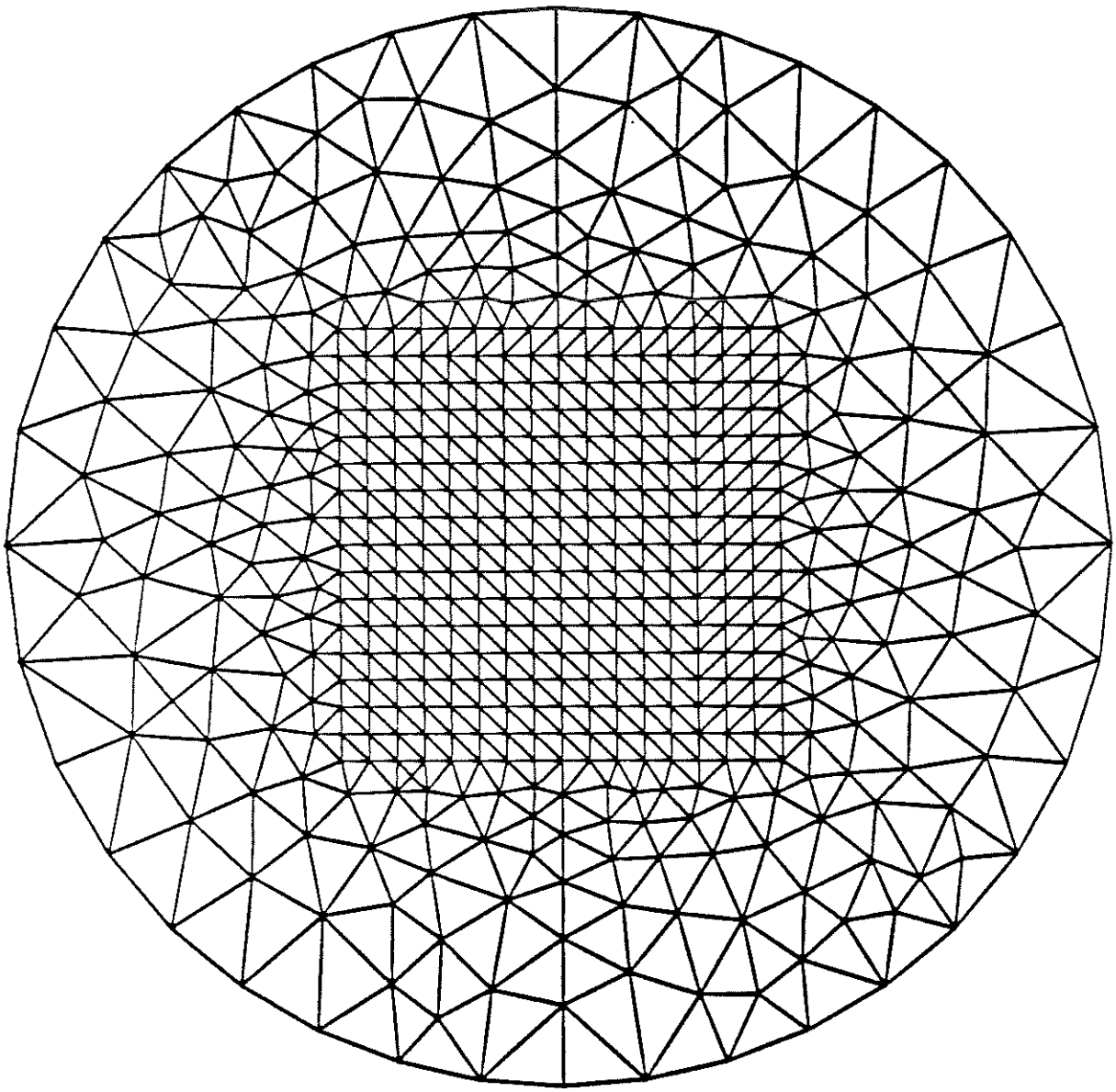
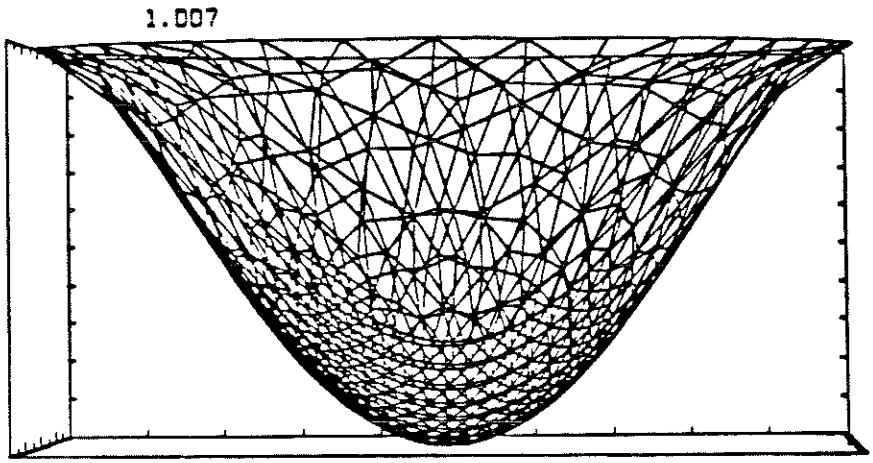


FIGURE 4.A

END (2,2) RF , RFEG-SFEG.

ITERATIØN NUMBER 1



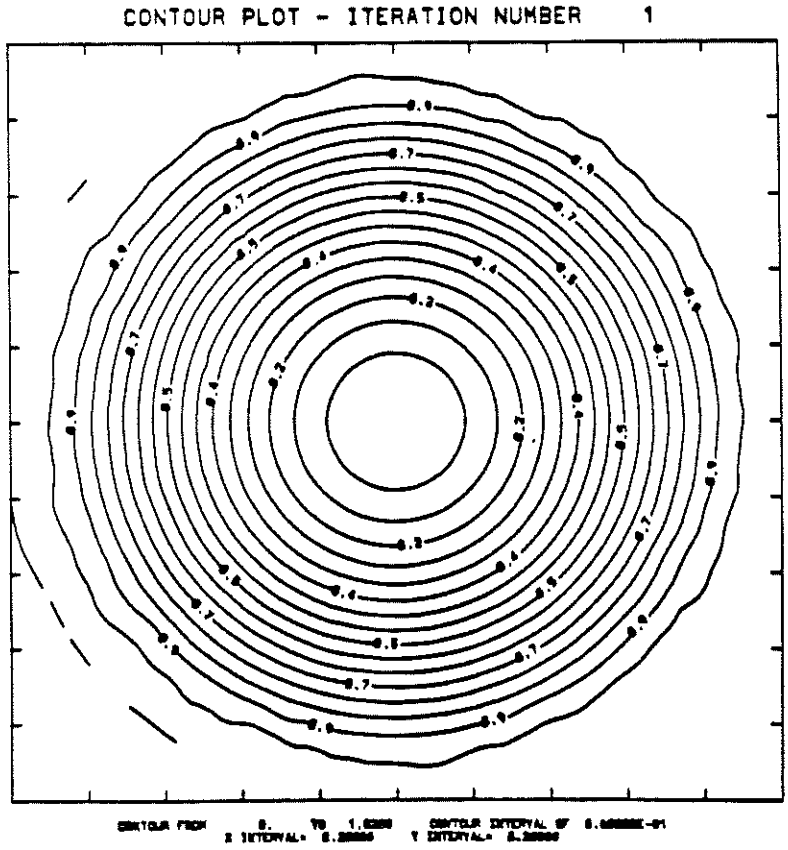
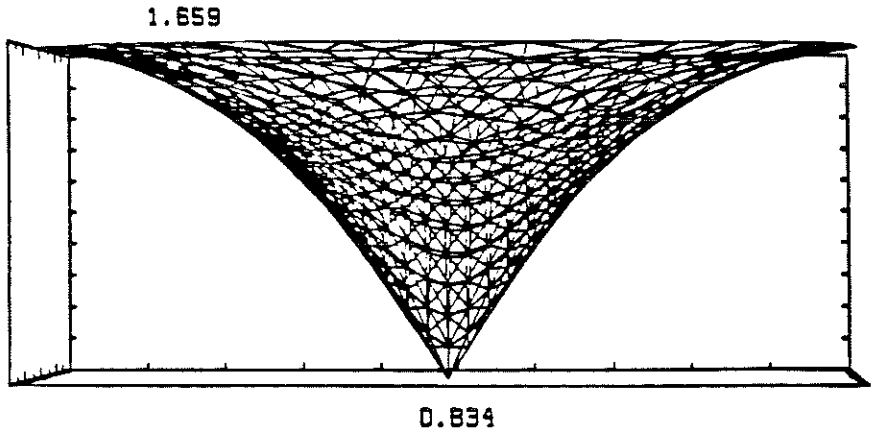


FIGURE 4.8

END (2,2) RF . RFEG-SFEG.

ITERATION NUMBER 90



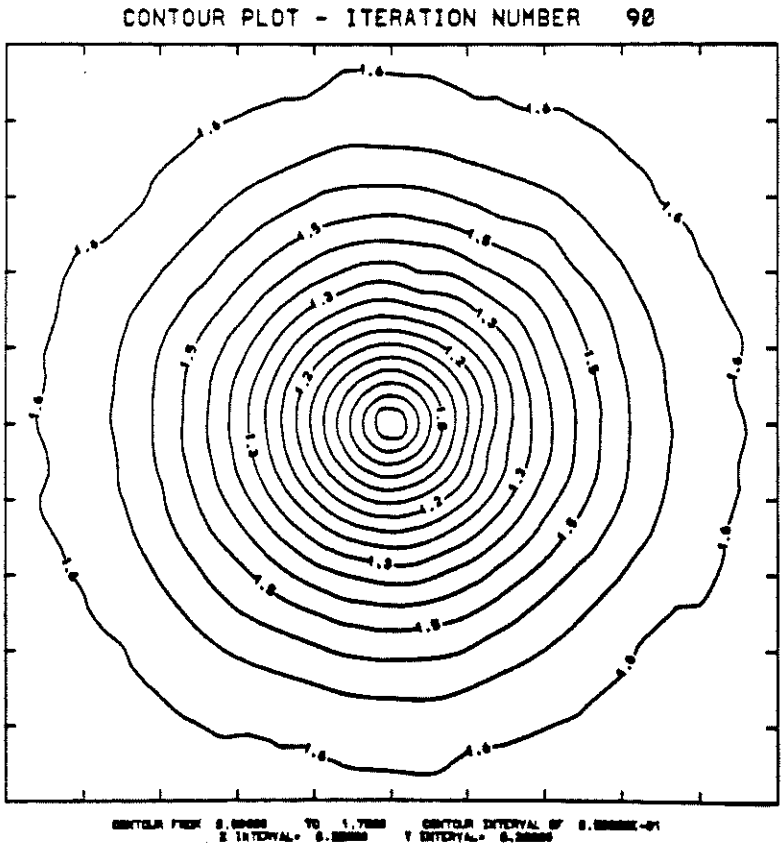
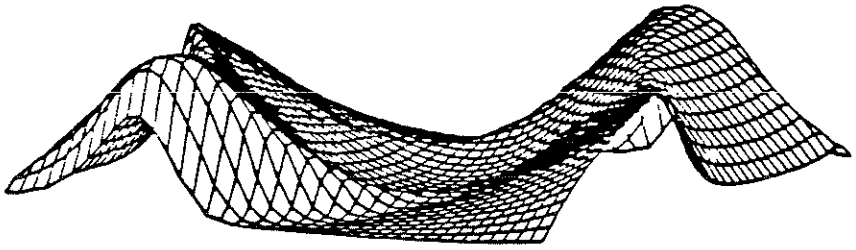
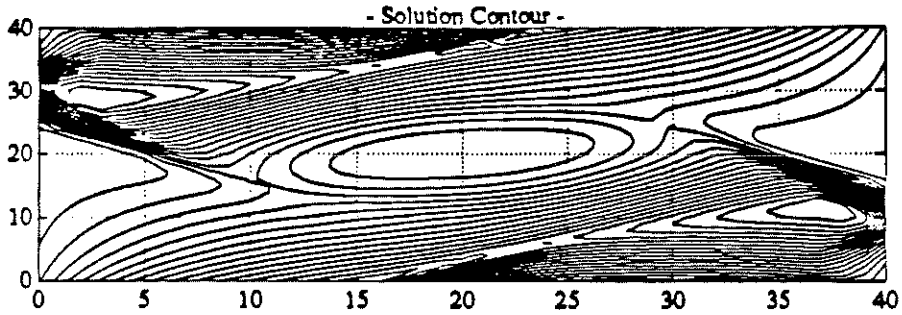


FIGURE 4.C

- Optimal Control Cost Problem - Ut+H(Ux,Uy,x,y)=0 -

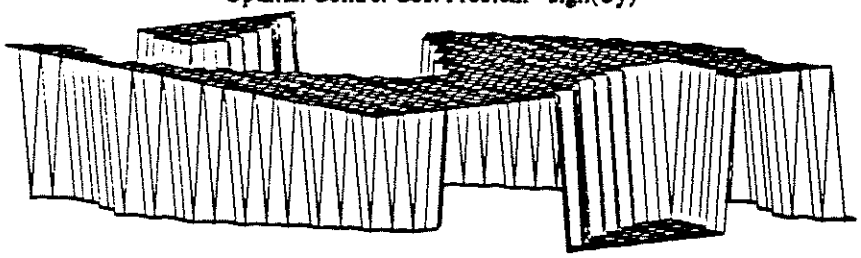


- t=6.2 , CFL=0.4 , m=n=40 -

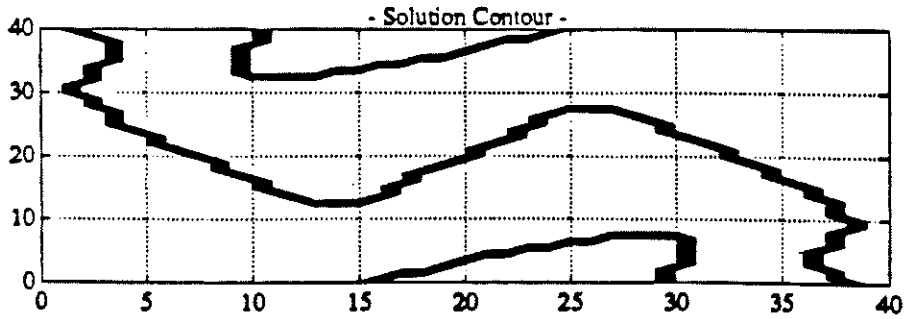


ENO (2,2) RF - Zmin= -0.0503181 , Zmax= 3.91945 -

- Optimal Control Cost Problem - sign(Uy) -



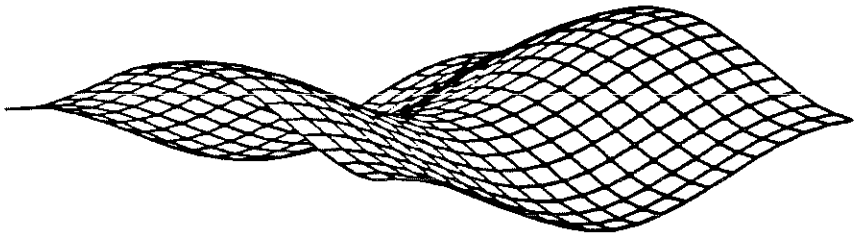
- t=6.2 , CFL=0.4 , m=n=40 -



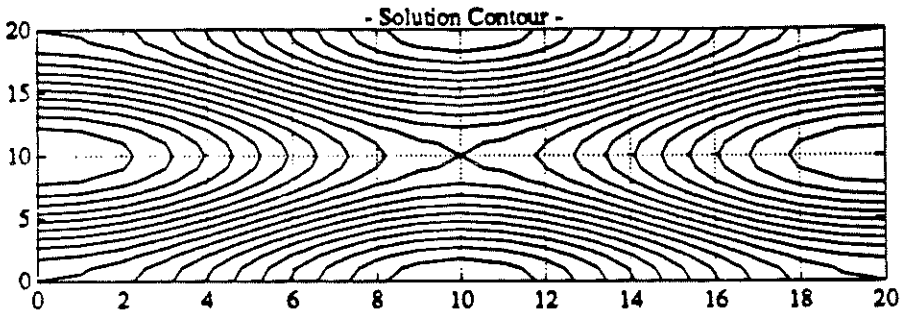
ENO (2,2) RF - Zmin= -1 , Zmax= 1 -

- Figure 5a -

- Combustion Pb with curvature dependancy- Ut+F(K)*H(Ux,Uy)=0 -

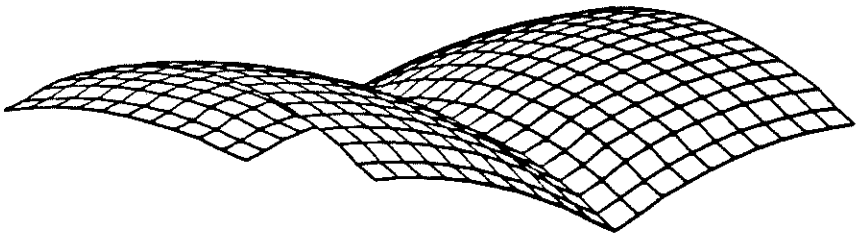


- Iteration # 1 , CFL=0.3 , eps=0.0 , m=n=20 -

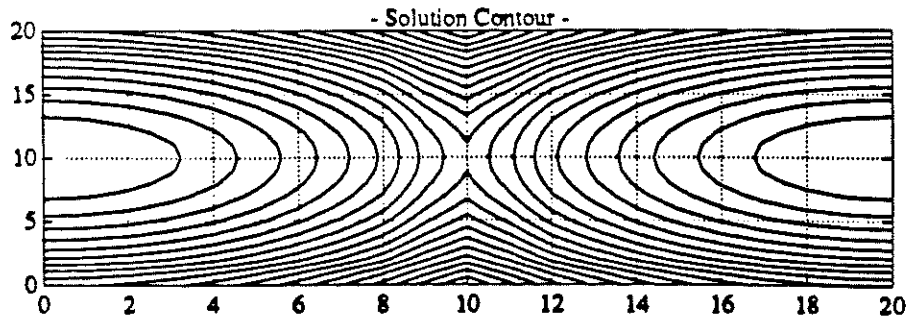


ENO (2,2) RF - Zmin= -1.98317 , Zmax= 2.01528 -

- Combustion Pb with curvature dependancy- Ut+F(K)*H(Ux,Uy)=0 -



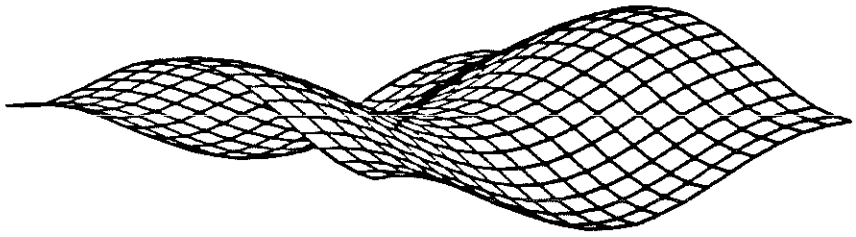
- Iteration # 100 , CFL=0.3 , eps=0.0 , m=n=20 -



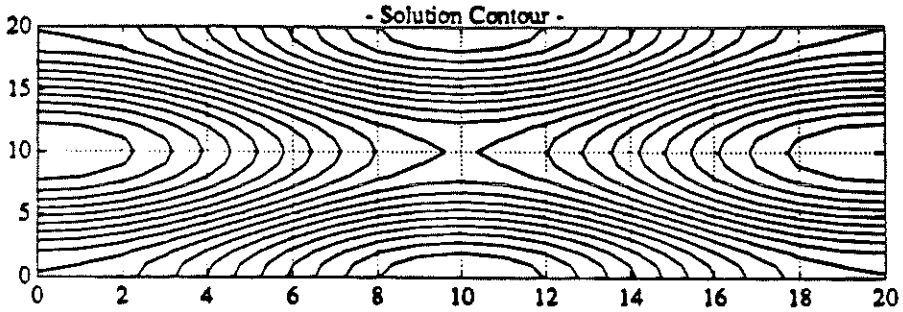
ENO (2,2) RF - Zmin= 3.32582 , Zmax= 3.50246 -

- Figure 5b. -

- Combustion Pb with curvature dependancy- Ut+F(K)*H(Ux,Uy)=0 -

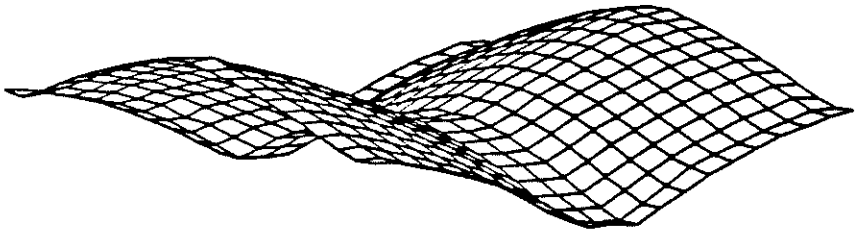


- Iteration # 1 , CFL=0.3 , eps=0.1 , m=n=20 -

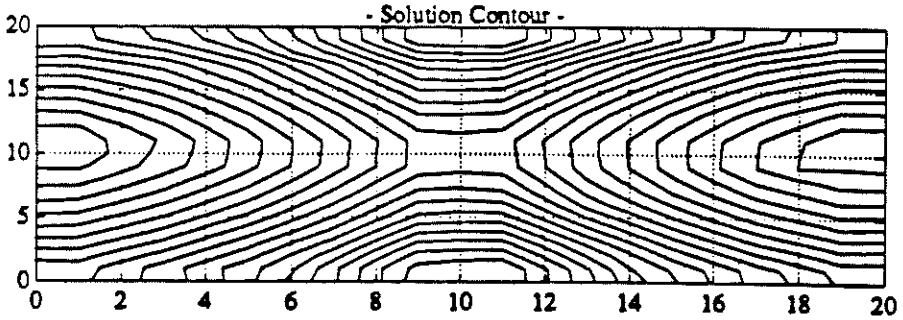


ENO (2,2) RF - Zmin= -1.88703 , Zmax= 1.94964 -

- Combustion Pb with curvature dependancy- Ut+F(K)*H(Ux,Uy)=0 -



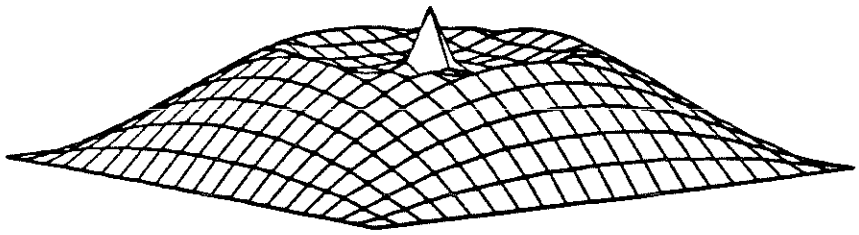
- Iteration # 100 , CFL=0.3 , eps=0.1 , m=n=20 -



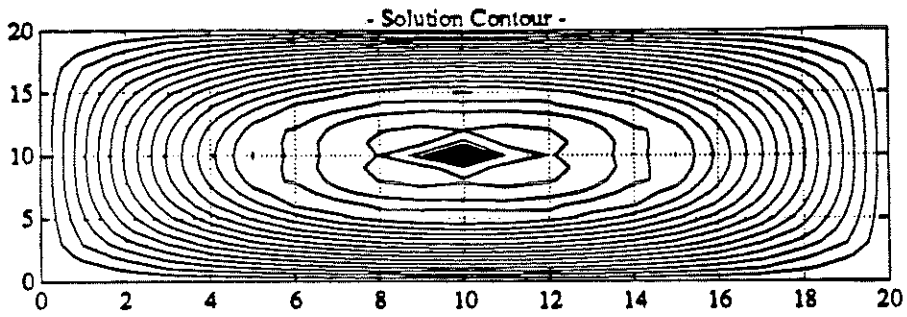
ENO (2,2) RF - Zmin= 2.81654 , Zmax= 2.96072 -

- Figure 5c. -

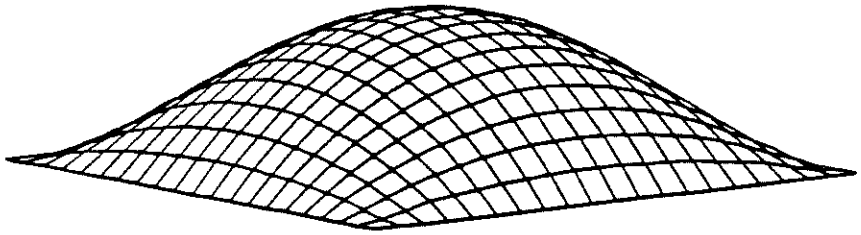
- Hamilton Jacobi 2D Equation Vision problem. -



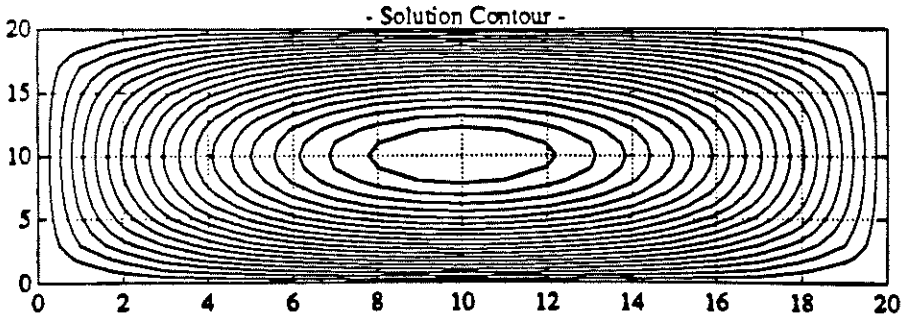
- Iteration # 200 , CFL=.4 , m=n=20 -



- Hamilton Jacobi 2D Equation Vision problem. -

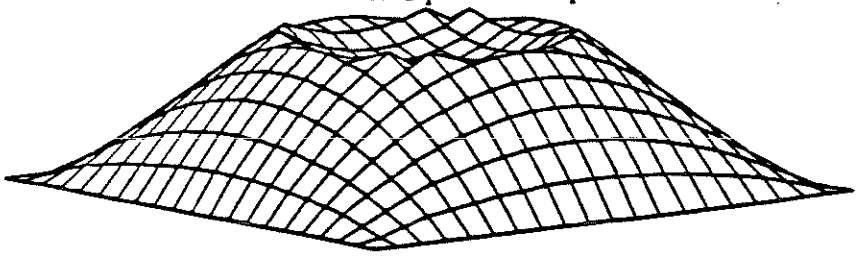


- Iteration # 400 , CFL=.4 , m=n=20 -

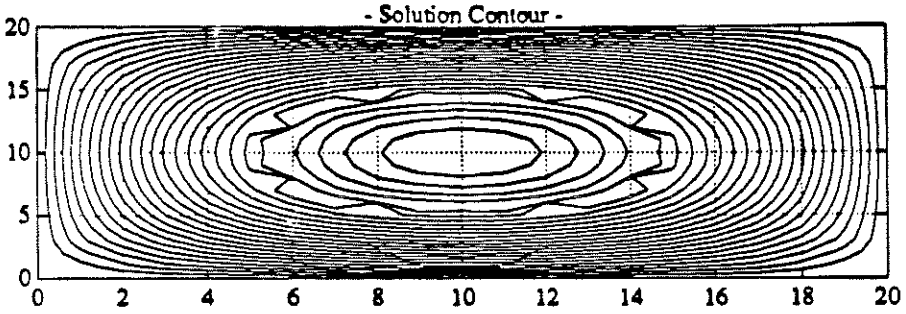


- Figure 5d. -

- Hamilton Jacobi 2D Equation Vision problem. -

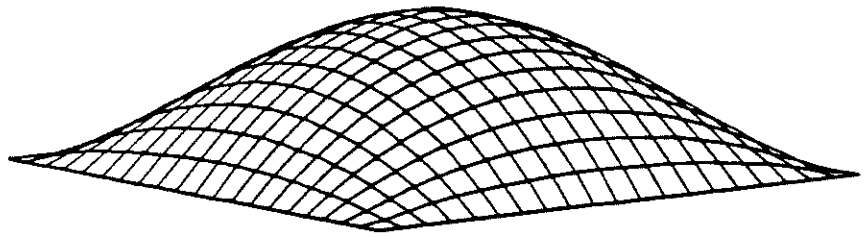


- Iteration # 400 , CFL=.7 , m=n=20 -

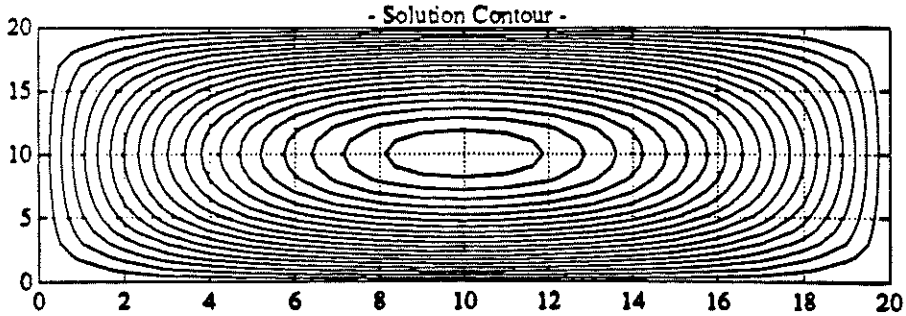


- Zmin=0., Zmax=0.7982, eps=0.02 -

- Hamilton Jacobi 2D Equation Vision problem. -



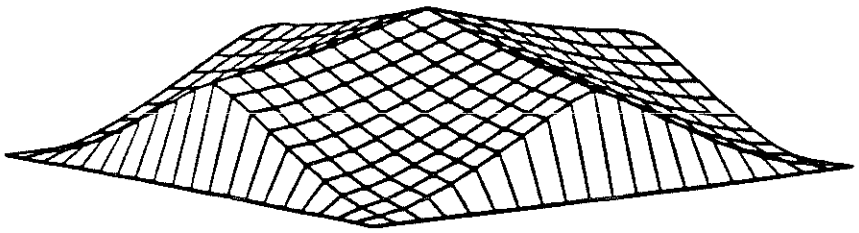
- Iteration # 800 , CFL=.7 , m=n=20 -



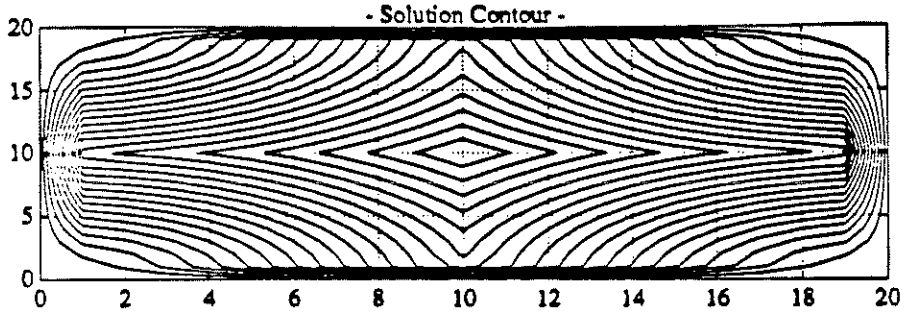
- Zmin=0., Zmax=1.0538, eps=0.02 -

- Figure 5c. -

- Hamilton Jacobi 2D Equation Vision problem. -

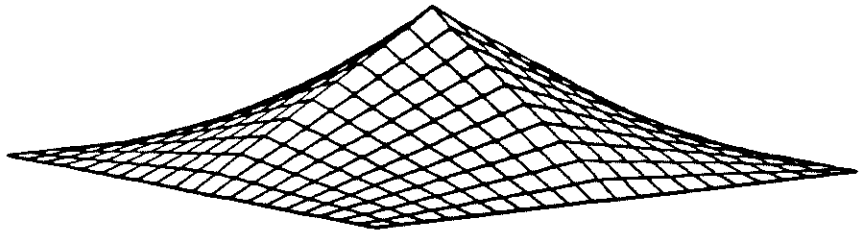


- Iteration # 1 , CFL=.7 , m=n=20 -

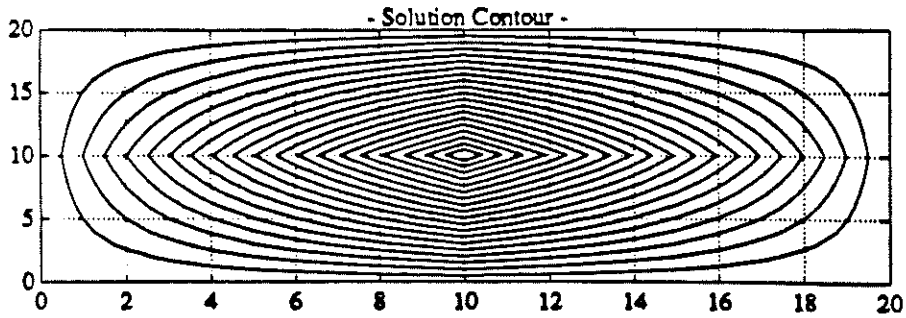


- Zmin=0., Zmax=0.0296, eps=0. -

- Hamilton Jacobi 2D Equation Vision problem. -



- Iteration # 100 , CFL=.7 , m=n=20 -

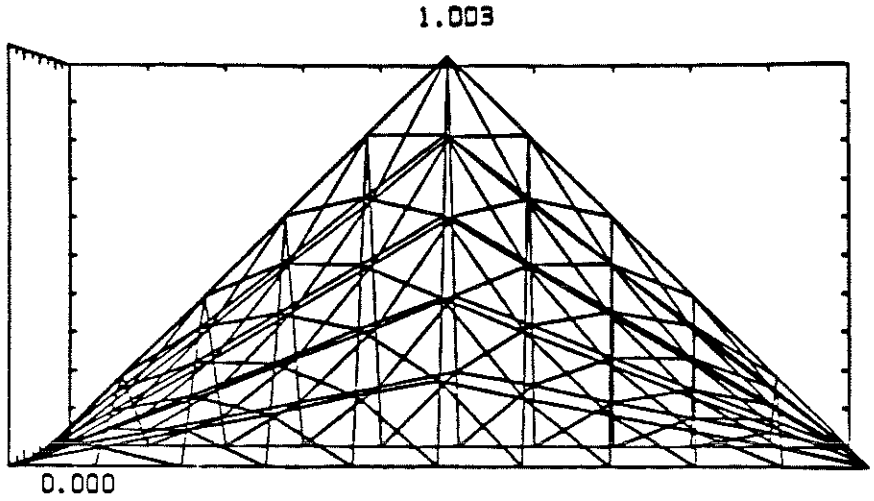


- Zmin=0., Zmax=1.0104, eps=0. -

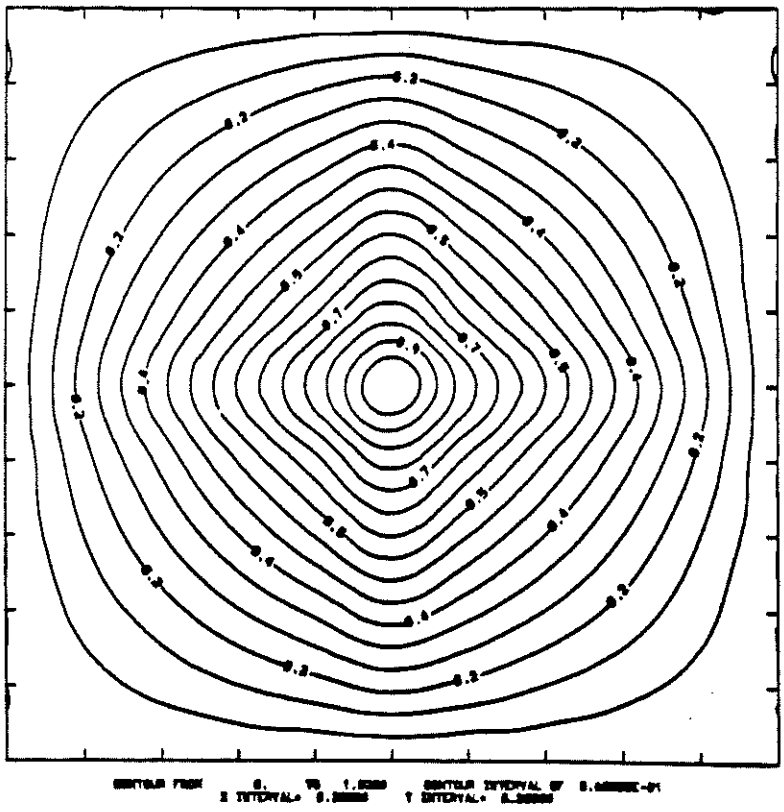
- Figure 5f. -

ENO (2,2) RF Mixed-formul

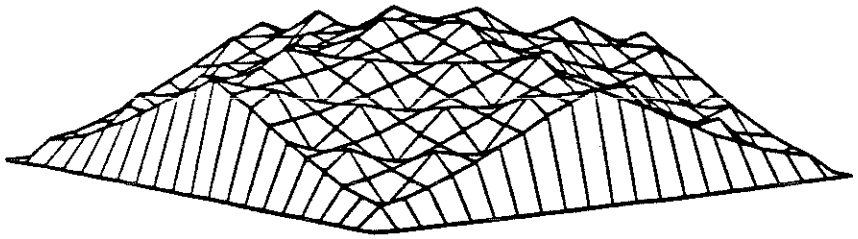
ITERATION NUMBER 30



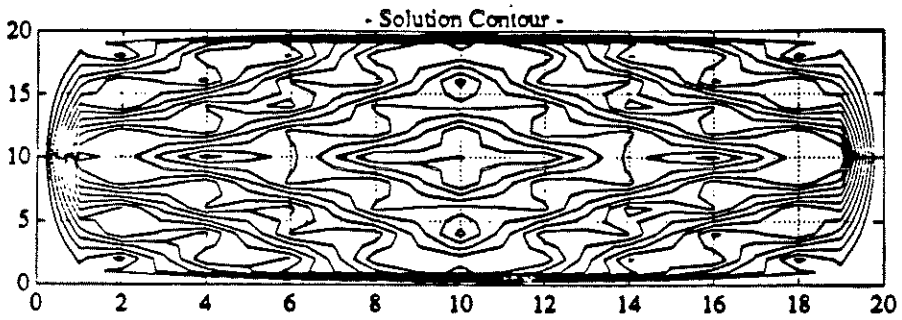
CONTOUR PLOT - ITERATION NUMBER 30



- Hamilton Jacobi 2D Equation Vision problem. -

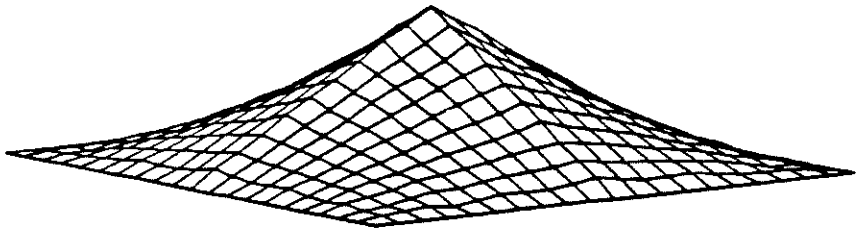


- Iteration # 1 , CFL=.5 , m=n=20 -

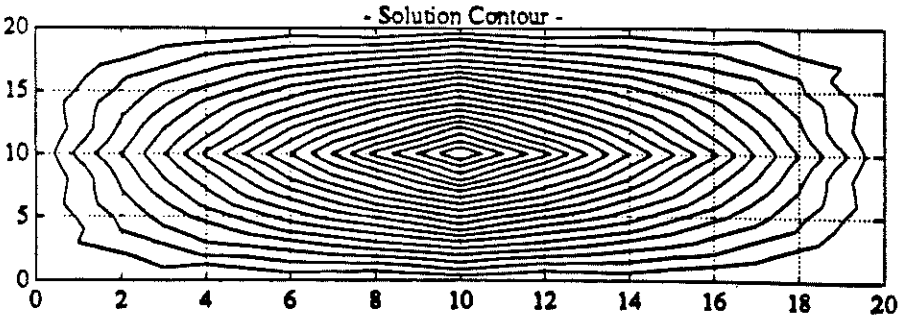


- Zmin=0., Zmax=0.0223, teta=0.05 -

- Hamilton Jacobi 2D Equation Vision problem. -



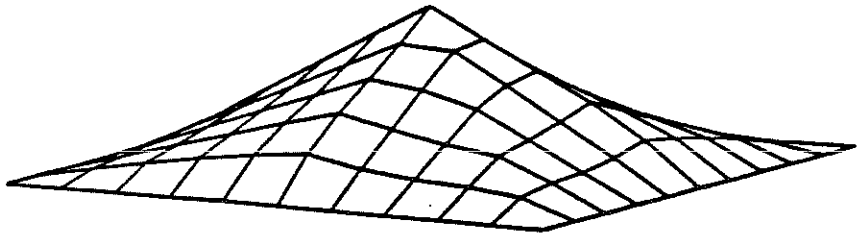
- Iteration # 150 , CFL=.5 , m=n=20 -



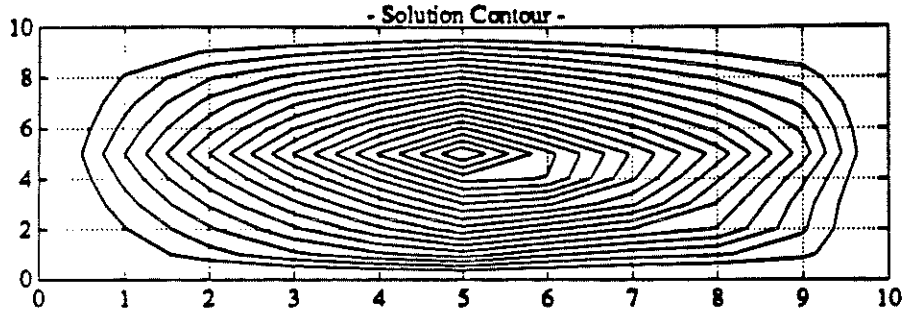
- Zmin=0., Zmax=1.1003, teta=0.05 -

- Figure 5h. -

- Hamilton Jacobi 2D Equation Vision problem. -

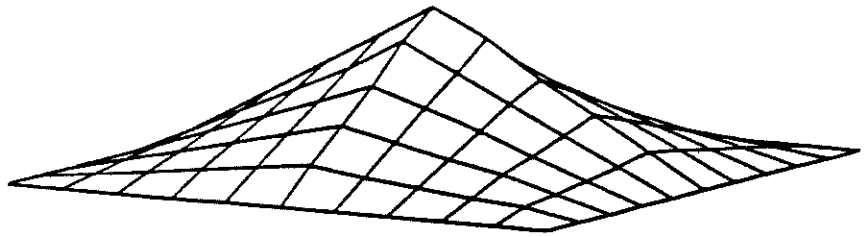


- Iteration # 200 , CFL=.7 , m=n=20 -

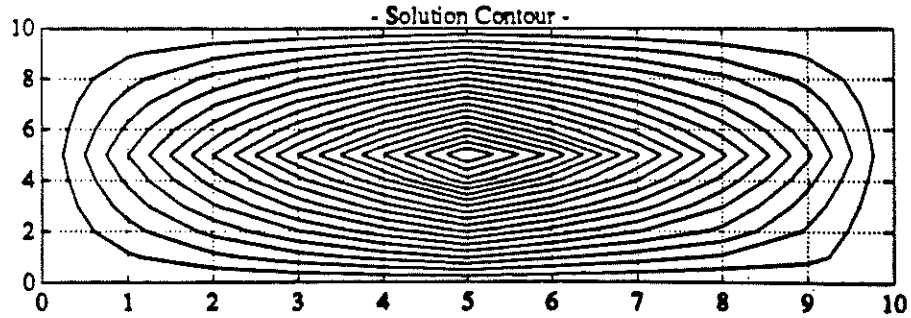


- Zmin=0., Zmax=1.18, eps=0.1, (x,z)=pi/10, (y,z)=pi/10 -

- Hamilton Jacobi 2D Equation Vision problem. -



- Iteration # 200 , CFL=.7 , m=n=20 -



- Zmin=0., Zmax=1.01, eps=0.01, (x,z)=pi/10, (y,z)=pi/10 -

- Figure 5i. -