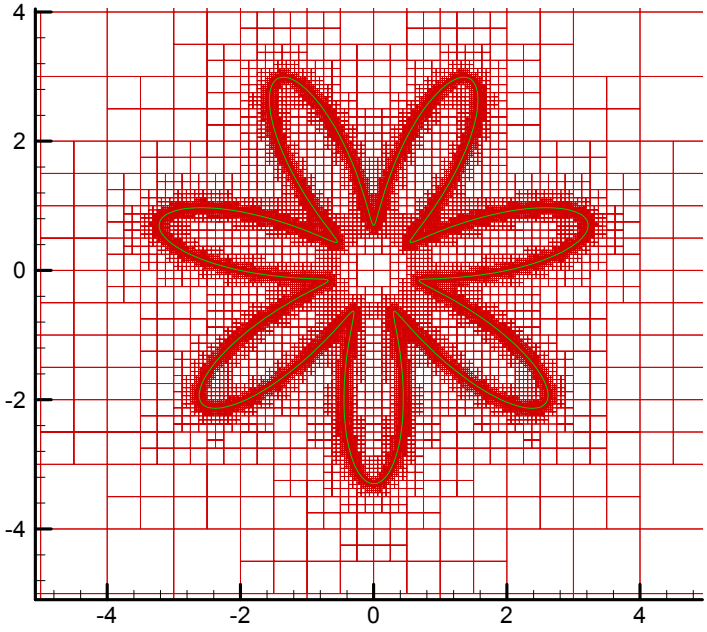
AIAA-2004-0082
The Level Set Method on Adaptive Cartesian Grid for Interface Capturing

Zhu Wang and Z.J. Wang
Michigan State University
East Lansing, Michigan



42nd Aerospace Sciences Meeting & Exhibit
4-8 January 2004
Reno, Nevada

THE LEVEL SET METHOD ON ADAPTIVE CARTESIAN GRID FOR INTERFACE CAPTURING

Zhu Wang[1] and Z.J. Wang[2]
Department of Mechanical Engineering
Michigan State University, East Lansing, Michigan 48824

ABSTRACT

The level set method, introduced by Osher and Sethian in 1988, is a powerful numerical approach for analyzing and computing interface motion. In this paper, we extend the level set method to two-dimensional unstructured adaptive Cartesian grid to further improve the solution accuracy and efficiency. A quadtree-based grid generator is developed for 2D adaptive Cartesian mesh generation. Following ideas from the finite volume method for hyperbolic conservation laws, we have developed a level-set numerical algorithm for the adaptive Cartesian grid. Some two-dimensional numerical examples are given to illustrate the capability of the developed algorithm. Second-order accuracy in smooth regions, good stability and convergence to viscosity solutions are demonstrated.

1. INTRODUCTION

In this paper, we consider the numerical solution of the level set equation introduced by Osher and Sethian[1]:

$$\frac{\partial \phi}{\partial t} + \vec{V} \cdot \nabla \phi = 0 \,. \qquad (1)$$

Here $\vec{V}$ is the velocity field and ϕ is the signed distance function to the interface under consideration. In 1988, Osher and Sethian devised the level set method as a simple and versatile method for computing and analyzing the motion of interfaces in two or three dimensions.

Numerical algorithms for the level set equation have been further developed over the past decade[2,3]. Successful examples include the finite difference method[4], high-order ENO schemes by Osher and Shu[5], F. Lafon and Osher[6], discontinuous Galerkin finite element method by Hu and Shu[7], high-order weighted essentially non-oscillatory (WENO) schemes by Zhang and Shu[8] et al. Meanwhile the level set methods have been applied to solve a wide variety of problems involving external physics, such as compressible flow[9,10], incompressible flow[11,12], Stefan problems[13], kinetic crystal growth, epitaxial growth of thin films, vortex dominated flows and multiphase flows.

Most of these simulations with the level set method were carried out on structured Cartesian meshes[5-7]. These algorithms have been applied successfully to a variety of applications. Since the mesh is uniform everywhere, it is not possible to have higher resolution for the interface than elsewhere. This may be very costly for some applications. There also have been a few attempts to use unstructured triangular meshes for the level set equation. Barth and Sethian[14] developed such a level set method in 1998. The unstructured grids are much more flexible in handling complex geometries. However, the numerical result may be influenced by the quality of the computational mesh. Zhang and Shu[8] designed a high-order WENO scheme on triangular meshes in 2001. High-order accuracy can be achieved with the WENO algorithm, while the algorithm is relatively complex to implement because of the large reconstruction stencils.

In this paper, the adaptive Cartesian mesh is advocated. The adaptive Cartesian grid method has been widely used in computational fluid dynamics (CFD). Its ability in handling complex geometries and in solution based grid adaptations is demonstrated in many applications[15,16]. One unique feature of the adaptive Cartesian grid is that diverse length scales can be handled very efficiently with local feature based grid adaptations. The adaptive Cartesian grid was also used by Strain to solve the level set equation using a Lagrangian algorithm[17]. In this paper, a finite-volume algorithm is chosen to solve the level-set equation. Ultimately we wish to solve multi-phase flow problems, and coupling between the level set equation and the flow governing equations may be easier if finite volume methods are used in solving both equations. With the automatic adaptive grid refinement, the Narrow Band method[18] can also be implemented easily. Higher resolution of interfaces will result in higher accuracy for the flow physics since the accuracy of the interface plays a critical role in the overall flow physics. The main advantages of the method are:

---

[1] Graduate Student (wangzhu@egr.msu.edu).
[2] Associate Professor (zjw@egr.msu.edu), AIAA Associate Fellow.

(1) Automatic and fast grid generation;
(2) Straightforward grid adaptation based on the location of the interface and other key flow features;
(3) Simple flux formulation and tree-based data structures suitable for object oriented programming;

The remainder of this paper is organized as follows. Following introduction, the numerical algorithm is presented and detailed formulations are given in Section 2. Then, the method to generate two-dimensional adaptive Cartesian mesh is described and discussed in Section 3. After that, numerical examples are given to verify the accuracy, stability and convergence properties. Finally concluding remarks based on the current study are given in Section 5.

2. NUMERICAL ALGORITHM

2.1 Level Set Equation

Let Γ be the interface bounding an open region Ω (possibly multiply connected) in R^n. Osher and Sethian[1] defined a smooth signed distance function $\phi(x,t)$ (at least Lipschitz continuous), which represents the interface as the set where $\phi(x,t)=0$. With this definition, the level set function ϕ has the following properties:

$$\phi(x,t)<0 \text{ for } x \in \Omega$$
$$\phi(x,t)>0 \text{ for } x \notin \Omega$$
$$\phi(x,t)=0 \text{ for } x \in \partial\Omega=\Gamma(t).$$

Thus, the interface is to be captured by calculating the set $\Gamma(t)$ for which $\phi(x,t)$ is equal to zero. The motion of interface $\Gamma(t)$ can then be analyzed by convecting the level set function $\phi(x,t)$ with the velocity field $\vec{V}$. The governing equation for $\phi(x,t)$ is then (1). Because topological changes such as breaking and merging are well defined, the level set equation is of significance for numerical computation.

Note that the velocity field $\vec{V}$ is the velocity on the interface and can be arbitrary elsewhere. And it can be simulated through external physics such as the basic laws of fluid mechanics. Consider that the normal component of the velocity is $V_N=\vec{V}\cdot\frac{\nabla\phi}{|\nabla\phi|}$, so the level set equation can be rewritten as the general level set equation,

$$\frac{\partial\phi}{\partial t}+V_N(\vec{x},\nabla\phi,\kappa)|\nabla\phi|=0, \qquad (2)$$

where $\nabla\phi$ is the gradient of the distance function ϕ, and κ denotes the mean curvature of the interface. The curvature κ can be calculated using $\kappa=\nabla\cdot\frac{\nabla\phi}{|\nabla\phi|}$. If we ignore the high-order derivative terms in (2), (2) can be viewed to be a first order Hamilton-Jacobi (H-J) equation

$$\frac{\partial\phi}{\partial t}+H(\vec{x},\nabla\phi)=0. \qquad (3)$$

A more general formulation than (3) can be written as

$$\frac{\partial\phi}{\partial t}+H(\vec{x},\nabla\phi,\kappa)=0,\quad (\vec{x},t)\in R^n\times R^+ \qquad (4)$$
$$\phi(\vec{x},t=0)=\phi_0(\vec{x})$$

If we take the gradient of (4), we obtain the following equations[6]

$$\begin{cases} u_t+H(x,y,u,v,\kappa)_x=0 \\ v_t+H(x,y,u,v,\kappa)_y=0 \end{cases}, \qquad (5)$$

where $(u,v)=(\phi_x,\phi_y)$, with the initial condition of $\begin{bmatrix} u(x,y,t=0) \\ v(x,y,t=0) \end{bmatrix}=\begin{bmatrix} u_0(x,y) \\ v_0(x,y) \end{bmatrix}$. Equation (5) can be viewed as a hyperbolic conservation law, for which many numerical methods have been developed in the last three decades, including the finite volume method. The curvature κ in 2D can be expressed as

$$\kappa=\nabla\cdot\frac{\nabla\phi}{|\nabla\phi|}=\frac{\phi_{xx}\phi_y^2-2\phi_y\phi_x\phi_{xy}+\phi_{yy}\phi_x^2}{(\phi_x^2+\phi_y^2)^{3/2}}. \qquad (6)$$

Because equations (5) are not a strictly hyperbolic system, it may not be equivalent to equation (4) if we treat u and v as independent variables. Thus ϕ is still treated as the solution variable, which is a piecewise polynomial over each cell. Then the derivatives of the polynomial are taken to obtain u and v.

2.2 Finite Volume Method

Next we focus on finding the numerical solution of the hyperbolic system (5). For a 2D arbitrary un-structure grid, the finite volume method is chosen due to its efficiency and simplicity. Assume Ω is divided into general polygons of size h. The cell is called a control volume (CV). The solution unknowns are the cell-averages of the level set function ϕ. Alternatively, the cell average can also be taken to be the point value of ϕ at the cell centroid since the numerical scheme is 2nd order accurate. If we integrate (5) in CV i, the following equations are obtained:

$$\begin{aligned} \int_i u_t\,dxdy+\int_i H(x,y,u,v,\kappa)_x\,dxdy=0 \\ \int_i v_t\,dxdy+\int_i H(x,y,u,v,\kappa)_y\,dxdy=0 \end{aligned}. \qquad (7)$$

By using the divergence theorem, the volume integral terms in equation (7) can be transformed into surface integrals

$$\text{vol}(i)\cdot\frac{d}{dt}\overline{u}_i+\oint_{\partial V_i}\left(H\cdot n_x\right)ds=0 \tag{8}$$

and

$$\text{vol}(i)\cdot\frac{d}{dt}\overline{v}_i+\oint_{\partial V}\left(H\cdot n_y\right)ds=0\,, \tag{9}$$

where the vector $\left(n_x,n_y\right)$ is the normal of each face of cell *i*, $\text{vol}(i)$ is the volume, and $\overline{u}_i$ and $\overline{v}_i$ are the cell averages of u and v respectively. These variables and their arrangement are illustrated in Fig. 1.

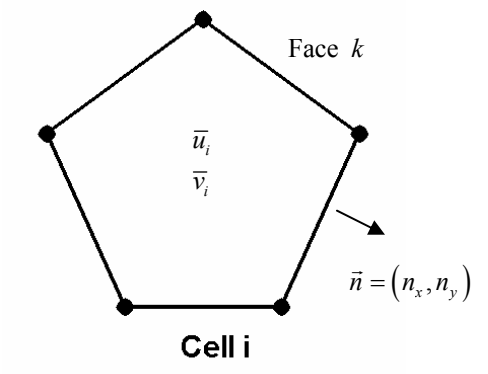


Fig.1 Schematic of a polygon control volume

Once the solution unknowns are given, they are used to build piecewise polynomials in each CV to achieve high order accuracy. No solution continuity is enforced at the cell interfaces. Therefore the fluxes through these interfaces are not uniquely defined since different solutions exist on both sides of the interfaces. Similar to the 2D Euler equation, we introduce Riemann numerical fluxes for (8) and (9). Therefore (8) and (9) can be rewritten as

$$\text{vol}(i)\cdot\frac{d}{dt}\overline{u}_i+\sum_{k=1}^{k=ne}\hat{H}_{1k}S_{ik}=0 \tag{10}$$

and

$$\text{vol}(i)\cdot\frac{d}{dt}\overline{v}_i+\sum_{k=1}^{k=ne}\hat{H}_{2k}S_{ik}=0\,. \tag{11}$$

For the i[th] cell, S_{ik} denotes length of the k[th] face, *ne* is the number of faces, and the numerical flux $\hat{H}_{1k}$ and $\hat{H}_{2k}$ is defined at the k^{th} face. The numerical fluxes $\hat{H}_{1k}$ and $\hat{H}_{2k}$ are Lipschitz continuous monotone fluxes which are consistent with function H in (4). Details of the Riemann flux can be found in the paper of Osher and Shu[5].

2.3 Flux Computation

Because the polynomial for each cell is constructed locally within a stencil, the left solution $\left(u^-,v^-\right)$ and the right solution $\left(u^+,v^+\right)$ are different. A simple local Lax-Friedrichs (LLF) flux is used for the numerical tests presented in this paper. This flux is computed given the left and right solutions, and has smaller numerical dissipation than the global Lax-Friedrichs flux. The LLF flux is defined as:

$$\hat{H}_{1k}\left(u^+,u^-,v^+,v^-\right)=\frac{1}{2}\left[H\left(u^+,v^+\right)+H\left(u^-,v^-\right)\right]\cdot n_x-\frac{1}{2}\alpha\cdot\left(u^+-u^-\right) \tag{12}$$

and

$$\hat{H}_{2k}\left(u^+,u^-,v^+,v^-\right)=\frac{1}{2}\left[H\left(u^+,v^+\right)+H\left(u^-,v^-\right)\right]\cdot n_y-\frac{1}{2}\beta\cdot\left(v^+-v^-\right) \tag{13}$$

where

$$\alpha=\max_{u,v}\left|\frac{\partial H(u,v)}{\partial u}\right|,\ \beta=\max_{u,v}\left|\frac{\partial H(u,v)}{\partial v}\right|. \tag{14}$$

2.4 Data Reconstruction

In order to evaluate the flux through a face, we need to know the solutions u and v at both sides of face. This can be done through solution reconstruction. Two reconstruction approaches are considered in the present study.

In the first method, a linear reconstruction is performed for the distance function ϕ to find the gradient ϕ_x and ϕ_y. Then given ϕ_x and ϕ_y, another linear reconstruction is carried out to obtain the second-derivatives ϕ_{xx}, ϕ_{xy} and ϕ_{yy}. The reconstruction is performed with a least-squares approach, and can be implemented efficiently using a face-based data structure. For a linear reconstruction, only face neighbor cells are included in the reconstruction stencil.

In the second method a cell-wise quadratic reconstruction[19] is performed. A larger reconstruction stencil is necessary to build a quadratic polynomial. In addition to face neighbors used in linear reconstructions, cells sharing a node with the current cell (node neighbors) are also included in the quadratic stencil. Fig. 2 shows the stencil used to perform quadratic reconstruction in an adaptive grid.

Let us consider the Taylor expansion of the level set function $\phi\left(x_j,y_j\right)$ about the cell center $\left(x_i,y_i\right)$ by omitting third and higher order terms:

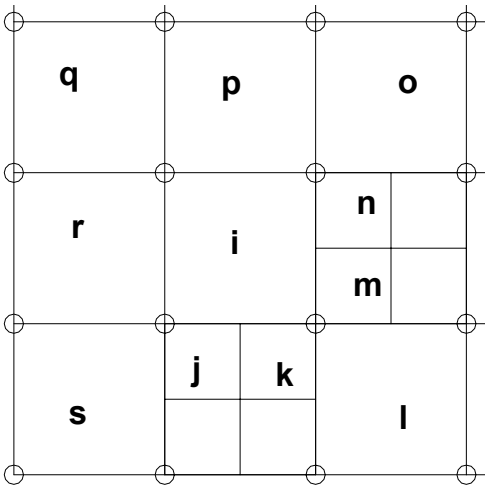


Figure 2. Quadratic reconstruction stencil of cell i for the Adaptive Cartesian Grid

$$\phi_j = \phi\left(x_j, y_j\right) = \phi_i + \phi_x \cdot \Delta x + \phi_y \cdot \Delta y + \frac{1}{2}\left[\phi_{xx} \cdot \left(\Delta x\right)^2 + 2\phi_{xy}\Delta x \cdot \Delta y + \phi_{yy} \cdot \left(\Delta y\right)^2\right], \quad (15)$$

where $\Delta x = x_j - x_i$, $\Delta y = y_j - y_i$. Assume N_j is the number of cells involved in the stencil supporting the reconstruction. Substituting the coordinates of the centroid of cell j in (15), we obtain the following system

$$A\begin{bmatrix}\phi_x\\ \phi_y\\ \phi_{xx}\\ \phi_{yy}\\ \phi_{xy}\end{bmatrix} = \Delta\phi \quad (16)$$

where:

$$\begin{aligned} A_{j1} &= x_j - x_i \\ A_{j2} &= y_j - y_i \\ A_{j3} &= \left(x_j - x_i\right)\left(x_j - x_i\right)/2 \\ A_{j4} &= \frac{1}{2}\left(y_j - y_i\right)\left(y_j - y_i\right)/2 \\ A_{j5} &= \left(x_j - x_i\right)\left(y_j - y_i\right) \end{aligned} \quad (17)$$

and

$$\Delta\phi_j = \phi_j - \phi_i . \quad (18)$$

Because (16) is an over-determined equation, it is solved by a least-squares method. A singular value decomposition (SVD) method is employed to compute the pseudo inverse of matrix A, which is denoted by A^+. Then the first and second order derivatives of ϕ can be solved directly using $A^+ \cdot \Delta\phi$.

If the solution is smooth, data limiting is not necessary. However, for non-smooth solutions with discontinuities, a limiter similar to that developed by Barth[20] is used to avoid numerical oscillations.

2.5 Time Integration

Eqns. (10) and (11) can be rewritten in the form of

$$\frac{d}{dt}\bar{Q}_i = \mathrm{res}(\bar{Q}_i) \quad (19)$$

where

$$\mathrm{res}\left(\bar{Q}_i\right) = -\frac{\sum_{j=1}^{j=J}\hat{H}_{ij}S_{ij}}{\mathrm{vol}(i)\cdot} \quad (20)$$

and

$$\bar{Q}_i = \begin{bmatrix}\bar{u}_i\\ \bar{v}_i\end{bmatrix}. \quad (21)$$

For time discretization, a two-stage Runge-Kutta time integration is used to solve (19), i.e.,

$$\begin{aligned} \bar{Q}_i^{(1)} &= \bar{Q}_i^n + \Delta t \cdot res\left(\bar{Q}_i^n\right) \\ \bar{Q}_i^{(2)} &= \frac{1}{2}\bar{Q}_i^n + \frac{1}{2}\bar{Q}_i^{(1)} + \frac{1}{2}\Delta t \cdot res\left(\bar{Q}_i^{(1)}\right). \\ \bar{Q}_i^{n+1} &= \bar{Q}_i^{(2)} \end{aligned} \quad (22)$$

The two-stage Runge-Kutta time integration scheme can achieve second order time accuracy. Since (22) is only a scheme for the gradient $\nabla\phi$, we need to update the value of ϕ using (4). This procedure can be coupled with (22).

Another way to update ϕ is to use the following equation starting from a particular cell

$$\phi(\vec{r}_2, t) = \phi(\vec{r}_1, t) + \int_{\vec{r}_1}^{\vec{r}_2}\left(\phi_x dx + \phi_y dy\right). \quad (23)$$

The integral path should be taken to avoid crossing a derivative discontinuity if possible.

3. ADAPTIVE CARTESIAN GRID GENERATION

To make the level set method more efficient and accurate, adaptive mesh refinement strategy is adopted. Generally grid refinement is desired in regions where high curvature occurs and the speed function changes rapidly. To accurately simulate the motion of interfaces, the meshes are adaptively refined around the zero level interface. In this paper, the adaptive Cartesian grid is employed to implement this strategy. Quadtree-based adaptive Cartesian grids have been employed for CFD problems[15] and for interface capturing[16]. In the present study, a quadtree-based grid generator and a grid

adaptor have been developed for the lever set solver. The Cartesian grid is generated from a root cell, which represents the complete computational domain. Through recursive quadtree subdivisions, one can automatically generate fine grids near particular flow features such as a material interface. Fig. 3 shows an example grid after interface based grid refinement.

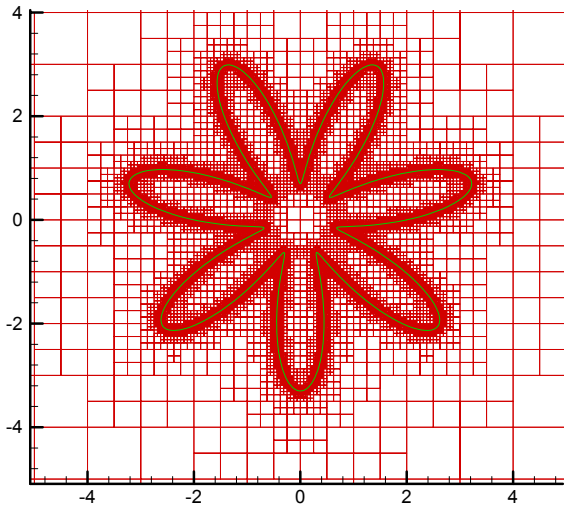


Fig.3 Adaptive Cartesian grid after interface-based grid refinement

If one is interested in only the interface, the Narrow Band method[18] is prefered because of its efficiency. With the Cartesian adaptive mesh, the minimum grid resolution is achieved within our narrow band thus high accuracy is guaranteed.

4. NUMERICAL EXPERIMENTS

In this section, the FV numerical method described in the previous sections is tested and applied to several two dimensional problems. Firstly, an accuracy study is carried out for the linear wave equation. Then 2D Burgers equation is solved to test the stability of the algorithm for non-smooth problems with discontinuities. After that, the motions of an initially circular interface are computed. A narrow band procedure is tested on the adaptive Cartesian grids. Finally, more sophisticated problems with high-order term (curvature κ) are solved to demonstrate the developed numerical method.

4.1 Accuracy Study with 2D linear wave equation

As the first test case, the 2D linear wave equation is chosen to study the numerical order of accuracy of the finite volume method. The 2D linear wave equation takes the following form

$$\begin{aligned} &\phi_t + \phi_x + \phi_y = 0 \qquad 0 \le x \le 2, 0 \le y \le 2 \\ &\phi(x,y,0) = \sin(\pi(x+y)) \end{aligned} \tag{24}$$

with periodic boundary conditions. The exact solution is:

$$\phi(x,y,t) = \sin\left(\frac{\pi}{2}(x+y-2t)\right). \tag{25}$$

For this accuracy study, the computations were carried out on uniform Cartesian meshes. Both the first order and second order schemes were tested. The results of the grid refinement study are presented in Table. 1. Note that the designed order of accuracy is obtained in both cases.

4.2 2D non-linear Burgers equation

To demonstrate the ability of the present method in solving non-smooth problems, the non-linear two-dimensional Burgers equation was employed. We choose the same initial condition as that in Hu and Shu[6]

$$\begin{aligned} &\phi_t + \frac{(\phi_x + \phi_y + 1)^2}{2} = 0; -2 \le x \le 2, -2 \le y \le 2 \\ &\phi(x,y,0) = -\cos\left(\frac{\pi}{2}(x+y)\right) \end{aligned} \tag{26}$$

with again periodic boundary conditions. In this test case, the second-order scheme is used and a limiter is applied in the data reconstruction to avoid numerical oscillations. Smooth solution at $t = 1/\pi^2$ and non-smooth solution at $t = 2/\pi^2$ with discontinuous derivatives are shown in Fig. 4. Note both solutions are non-oscillatory at least to the naked eye.

4.3 Expansion of An Initially Circular Interface

In this test case, (2) is solved with unit normal component of velocity V_n . The level set equation is then

$$\frac{\partial \phi}{\partial t} + \sqrt{\phi_x^2 + \phi_y^2} = 0. \tag{27}$$

For this case, the initial interface is a circle with a radius of 2. To study the accuracy, the initialization of the distance function ϕ is exact to avoid introducing extra numerical errors. The exact boundary conditions are imposed and a second-order scheme was employed. Fig. 5 displays the position of the interface from $t = 0$ to $t = 4$. Table 2 presents the numerical order of accuracy of the solution for $-1 \le \phi \le 1$ using uniform Cartesian meshes. The computations with adaptive refinement and a narrow band procedure were also carried out, and are shown in Figs. 6-8. The smallest

and largest cells in the computation differ by a factor of 16 or 4 levels. Note that high resolution of the interface was achieved through interface based grid adaptations.

4.4 Moving Circle

In this test case, two kinds of motions are considered. The first one is the advection with a constant velocity $\vec{V} = (1,1)$. Thus the solution is just a circle translating from the bottom-left to the top-right and the shape of circle should remain the same. The exact boundary conditions are imposed at $x = 0$ and $y = 0$. Simple extrapolations are applied at the other two boundaries. The second-order scheme is used in the simulation. The numerical solutions with uniform and adaptive Cartesian meshes are shown in Fig. 9 and Fig. 10 respectively. Fig.11 shows the adaptive Cartesian mesh at $t = 4$. Note that the solution on the adaptive mesh is as good as that on the uniform Cartesian mesh, and was obtained with much fewer cells.

In the second case the velocity field is set to $(y, -x)$. This velocity field is time-dependent and leads to a solid body rotation of a circle of radius 0.5 centered at (-1,0). Both first order and second order schemes are tested in this case. The computation was carried out on both uniform and adaptive Cartesian meshes. The numerical solutions are shown in Figs.12-14.

With the adaptive Cartesian grid, the numerical scheme can obtain a high accuracy and efficiency for capturing moving interfaces.

4.5 Propagating Surface

The motion of a propagating surface is computed in this section. The governing equation is:

$$\frac{\partial \phi}{\partial t} - (1 - \varepsilon\kappa)\sqrt{1 + \phi_x^2 + \phi_y^2} = 0, \quad x \in [0,1], y \in [0,1] \quad (28)$$

with a smooth initial condition

$$\phi(x, y, 0) = 1.0 - \frac{1}{4}(\cos(2\pi x) - 1)(\cos(2\pi y) - 1), \quad (29)$$

where ε is a small constant and κ is the mean curvature defined by

$$\kappa == \frac{\phi_{xx}(1 + \phi_y^2) - 2\phi_y\phi_x\phi_{xy} + \phi_{yy}(1 + \phi_x^2)}{(1 + \phi_x^2 + \phi_y^2)^{3/2}}. \quad (30)$$

Uniform Cartesian mesh was used and periodical boundary conditions were applied. In the case of ε=0, discontinuous derivatives appeared and the convergence to a viscosity solution is shown in Fig.15. The solution for ε =0.1 is shown in Fig. 16. This demonstrates the ability of the numerical method to solve curvature-dependent problems.

4.6 Star-Shaped Interface

Finally, we consider the motion of an initially seven-point star-shaped interface defined by

$$\gamma(s) = 20 * [\ 0.1 + (0.065 \sin(7 \cdot 2\pi s)(\cos(2\pi s), \sin(2\pi s)))\ ]. \quad (31)$$

This case was considered by S. Osher and J.A. Sethian[1]. First, it is assumed that the normal velocity component is $V_N(\vec{x}, \nabla\phi, \kappa) = 1$. The non-smooth solution of the interface, which is converged to a viscosity solution, is shown in Fig. 17. Then the interface velocity is set to $V_N(\vec{x}, \nabla\phi, \kappa) = -\kappa$. The curvature-dependent motion of the interface is simulated and shown in Fig.18. Both solutions agree well with other simulations presented in the literature.

5. CONCLUSIONS

A finite volume method for the level set equation has been developed for arbitrary 2D polygon grids including adaptive Cartesian grids. Two different least squares reconstruction approaches were tested. One is linear and the other is quadratic. Both approaches gave satisfactory numerical results Accuracy studies have been performed with the linear wave equation, and the present finite volume method was capable of achieving the designed numerical order of accuracy. Interface based grid adaptations have been demonstrated to capture the interfaces with high resolution. A variety of test cases have been used to test the present method, and results have been satisfactory. We plan to couple the level set solver with a flow solver to tackle multi-phase flow problems in the future.

6. ACKNOWLEDGEMENTS

The authors gratefully acknowledge support from the National Science Foundation under contracts DCC-0305504 and DSC-0325760.

7. REFERENCES

1. S. Osher and J.A. Sethian, “Fronts Propagating with Curvature Depentdent Speed: Algorithms Based on Hamilton-Jacobi Formulations”, *J.Comput. Phys.* 79, pp.12-49, 1988.
2. S. Osher and R.P. Fedkiw, “Level Set Methods: An Overview and Some Recent Results”, *J.Comput. Phys.* 169, pp.463-502, 2001.
3. J. A. Sethian and and P. Smereka, “Level Set Methods For Fluid Interface”, *Annu. Rev. Fluid Mech.*, 2003. 35:341–72.

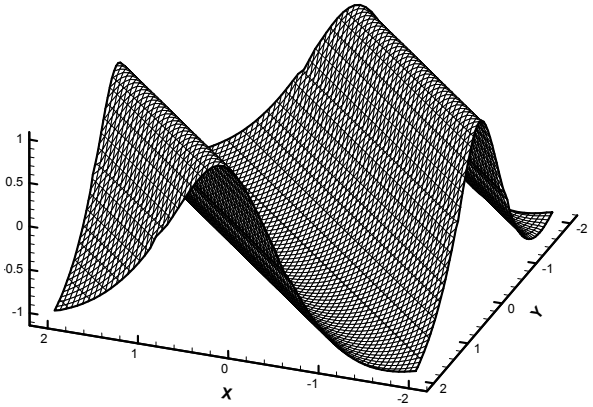
4. M. Crandall and P.L. Lions, "Two Approximations of Solutions of Hamilton-Jacobi Equations", *Math. Comp.*, vol.43, pp.1-19, 1984.
5. S. Osher and C.W. Shu, "High-Order Essentially Non-oscillatory Schemes For hamilton-Jacobi Equations", *SIAM J. NUMER. ANAL.*, Vol.28, No.4, pp.907-922, 1991.
6. F. Lafon and S. Osher, "High Order Two Dimensional Nonoscillatory Methods for Solving Hamilton-Jacobi Scalar Equations", *J. of Comp. Phys.* 123, pp.235-253, 1996.
7. C. Hu and C.W. Shu, "A Discontinuous Galerkin finite element method for HamiltonJacobi Equations", *ICASE Report No.98-2*, NASA/CR-1998-206903, 1998.
8. Y.Zhang and C.W.Shu, "High Order WENO Schemes for Hamilton-Jacobi Equations on Triangular Meshes", *ICASE Report No.2001-38*, NASA/CR-2001-211256, 2001.
9. W. Mulder, S.Osher and J.A. Sethian, "Computing Interface Motion in Compressible Gas Dynamics, *J. Comput. Phys.*100, pp.209-228, 1992.
10. R. Fedkiw, T. Aslam, B. Merriman and S. Osher, "A non- oscillatory Eulerian Approach to Interface in Multimaterial Flow (The Ghost Fluid Method), *J. Comput. Phys.* 152(2), pp.457-492, 1999.
11. M. Sussman, P. Smereka and S. Osher, "A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow", *J. Comput. Phys.* 114, pp.146-159, 1994.
12. M. Sussman, E. Fatemi, P. Smereka, and S. Osher, "An Improved Level Set Method for Incompressible Two-Phase Flows", *Computers and Fluids,* vol. 27, Nos 5-6, pp. 663-680,1998.
13. S. Chen, B. Merriman, S. Osher and P. Smereka, "A Simple Level Set Method For Solving Stefan Problems", *J. Comput. Phys.* 135, pp.8-29, 1997.
14. T.J. Barth and J.A. Sethian, "Numerical Schemes For The Hamilton-Jacobi And Level Set Equations On Triangulated Domains", *J. of Comput. Phys.*145, pp. 1-40, 1998.
15. Z.J. Wang, "A Quadtree Based Adaptive Cartesian Quad Grid Flow Solver For Navier Stokes Equations", *Computers & Fluids*, Vol.27, No.4, 1998.
16. J. Strain, "Tree Methods for Moving Interfaces", *Journal of Comput. Phys.*151, pp.616-648, 1999.
17. J. Strain, "A fast semi-Lagrangian contouring method for moving interfaces", *Journal of Comput. Phys.*169, pp.1-22, 2000.
18. J. A.Sethian, "Level Set Methods and Fast Marching Methods", *Cambridge University Press*, chap. 7, 1999.
19. M. Delanaye and Y. Liu, "Quadratic Reconstruction Finite Volume Schemes on 3D Arbitrary Unstructured Polyhedral Grids", *14 AIAA CFD Conference paper,* AIAA 99-3259, 1999.
20. T. Barth and D. Jesperson, "The Design and Application of Upwind Schemes on Unstructured Meshes", *AIAA paper 89-0366*, 1989.

Table 1. Accuracy Study for the 2D linear wave equation, errors at $t = 1.0$.

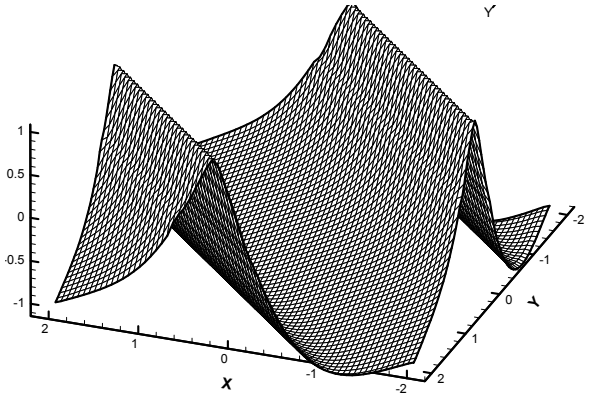
Order of accuracy	Grid	L_∞ Error	L_∞ Order	L_1 Error	L_1 Order
1st order	16×16	7.08e-01	---	4.56e-01	---
	32×32	4.60e-01	0.62	2.94e-01	0.63
	64×64	2.65e-01	0.80	1.69e-01	0.80
	128×128	1.43e-01	0.89	9.10e-02	0.89
	256×256	7.42e-02	0.95	4.72e-02	0.95
2nd order	16×16	8.19e-02	---	5.34e-02	---
	32×32	2.03e-02	2.01	1.30e-02	2.04
	64×64	5.06e-03	2.00	3.23e-03	2.01
	128×128	1.26e-03	2.01	8.04e-04	2.01
	256×256	3.16e-04	2.00	2.01e-04	2.00

Table 2. Accuracy study for the expansion of a circular interface, errors at $t = 4.0$.

Order of accuracy	Grid	L_∞ Error	L_∞ Order	L_1 Error	L_1 Order
2nd order	16×16	5.41e-02	---	1.39e-02	---
	32×32	8.65e-03	2.64	2.56e-03	2.44
	64×64	2.81e-03	1.62	5.81e-04	2.14
	128×128	9.11e-04	1.63	1.35e-04	2.11
	256×256	2.90e-04	1.65	3.18e-05	2.09



(a) $t = 1/\pi^2$



(b) $t = 2/\pi^2$

Fig.4 Numerical solution of 2D non-linear Burgers equation, 64×64 Cartesian meshes

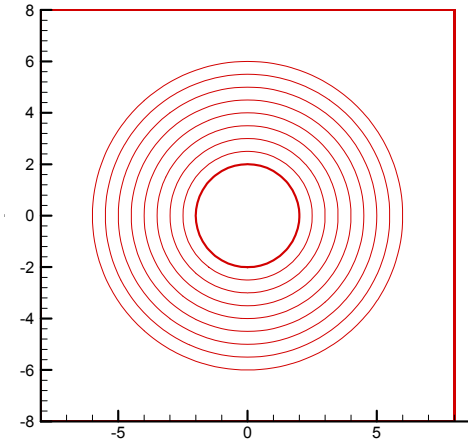


Fig. 5 Expansion of circular interface, $V_n = 1$ $T = 0, 4\,(0.5)$, Uniform 64×64 Cartesian Mesh

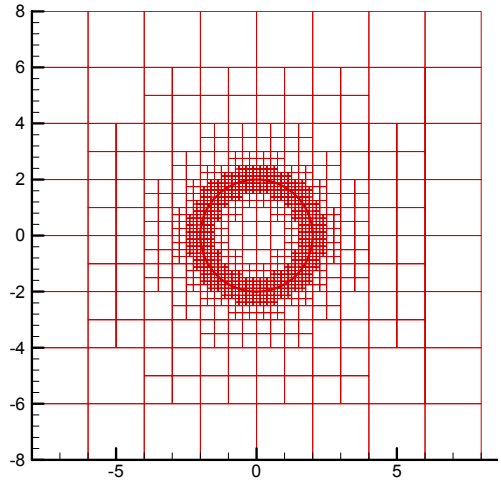


Fig. 6 Expansion of circlular interface, $V_n = 1$, $T = 0$, Adaptive Cartesian Mesh

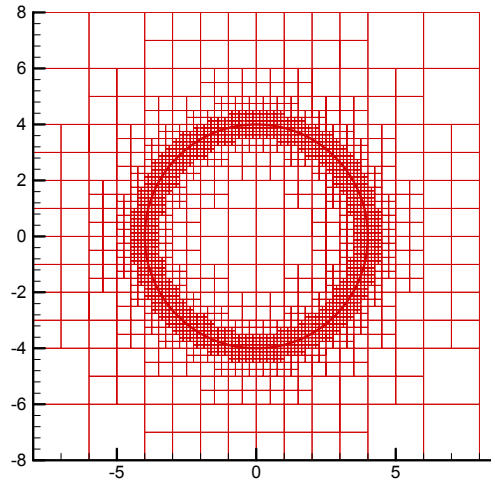


Fig. 7 Expansion of circular interface, $V_n = 1$, $T = 2$, Adaptive Cartesian Mesh

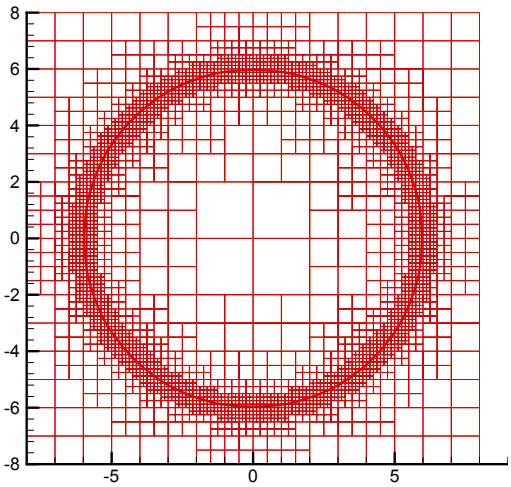


Fig. 8 Expansion of circular interface, $V_n = 1$, $T = 4$, Adaptive Cartesian Mesh

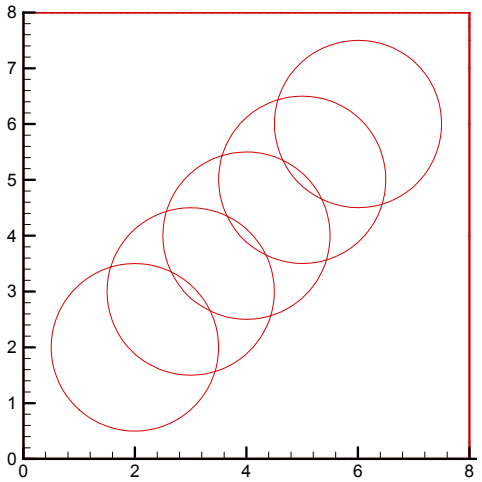


Fig. 9 Translation of a circular interface, 2nd order $T = 0, 4\,(1.0)$, uniform 64×64 Cartesian mesh

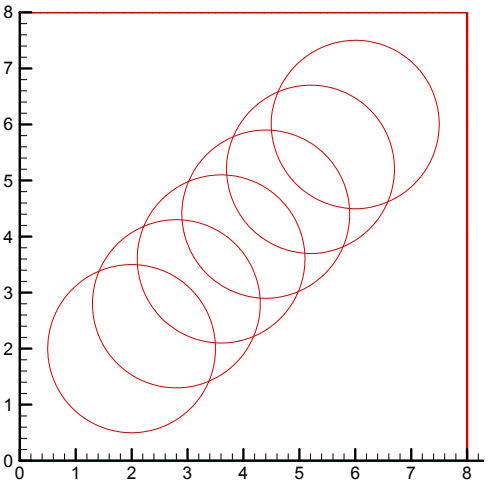


Fig. 10 Translation of a circular interface $T = 0, 4\,(0.8)$, adaptive Cartesian mesh

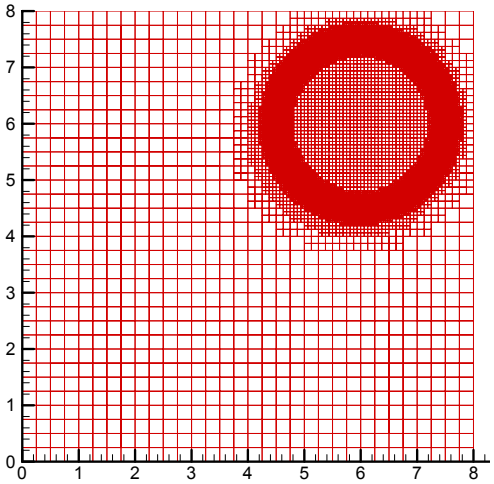


Fig. 11 The adaptive Cartesian mesh at $T = 4$

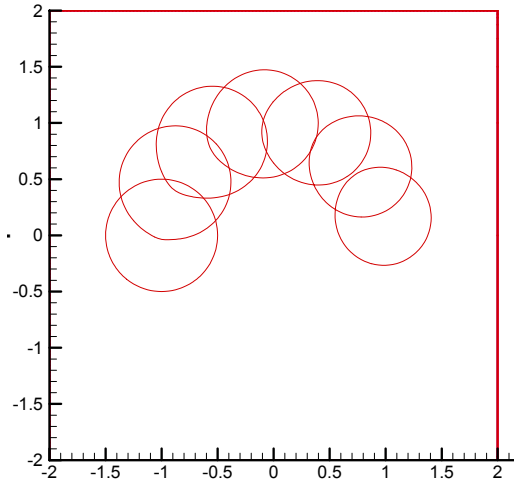


Fig. 12 Rotation of a circular interface, 1st order $T = 0, \pi\,(\pi/6)$, 128×128 Cartesian Mesh

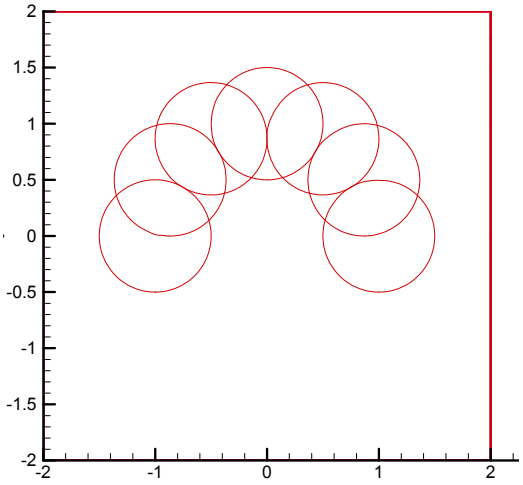


Fig. 13 Rotation of a circular interface, 2nd order $T = 0, \pi\,(\pi/6)$, 128×128 Cartesian Mesh

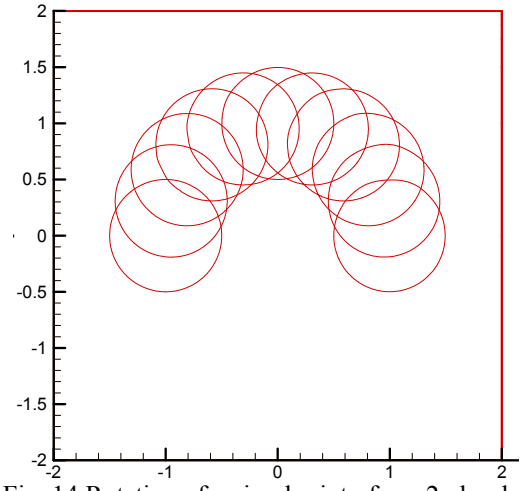


Fig. 14 Rotation of a circular interface, 2nd order $T = 0, \pi\,(\pi/10)$, Adaptive Cartesian Mesh

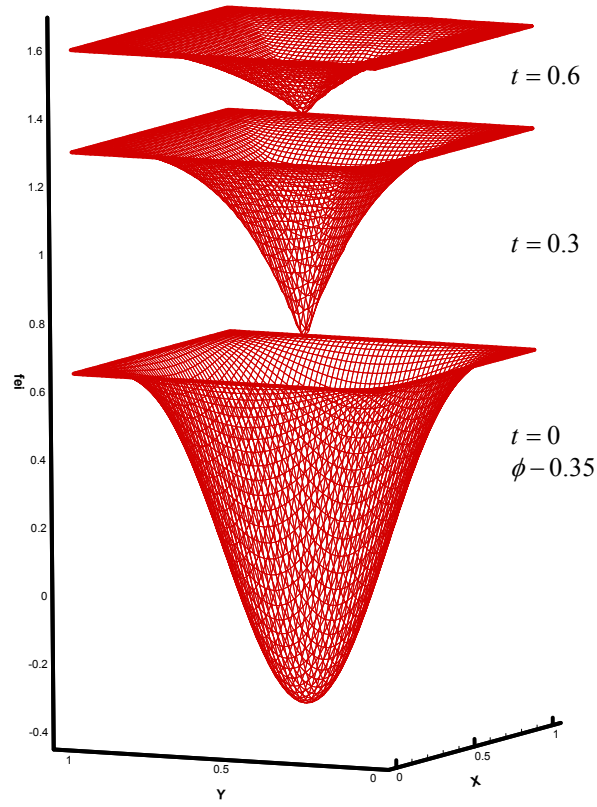


Fig. 15 Propagating surface, $\varepsilon = 0$

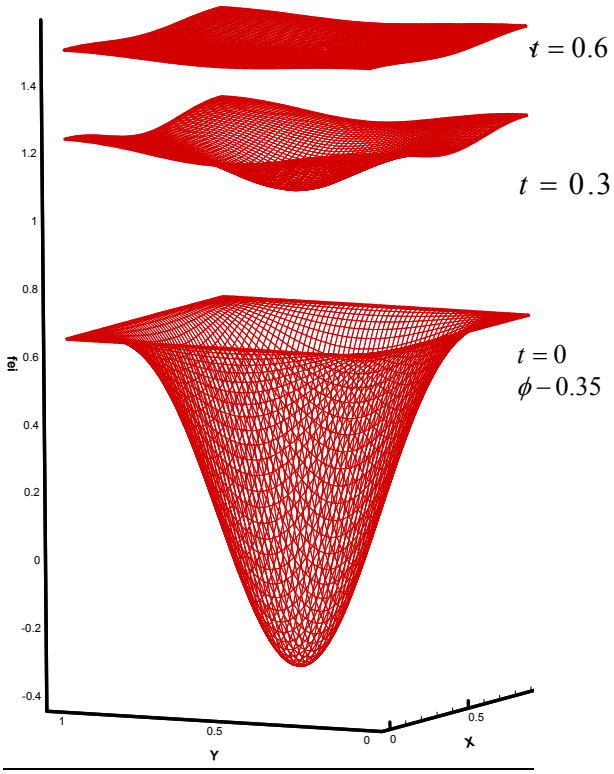


Fig. 16 Propagating surface, $\varepsilon = 0.1$

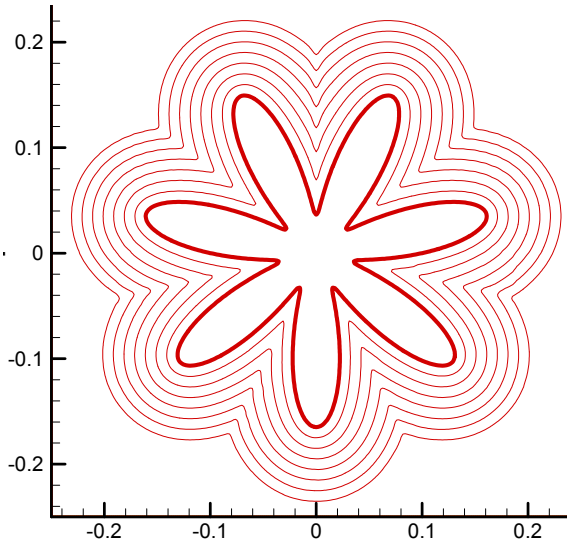


Fig. 17 Star-shaped interface, $V_N = 1$, $T = 0, 0.07(0.01)$

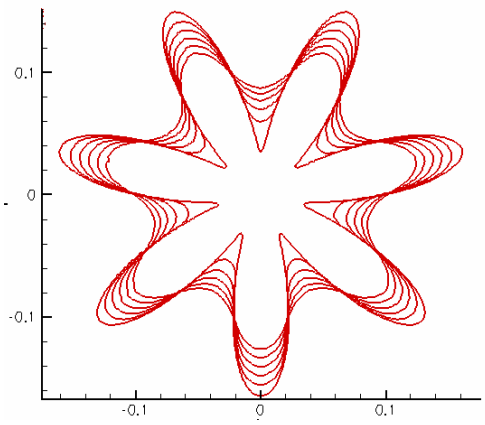


Fig. 18 Star-shaped interface, $V_N = -\kappa$, $T = 0, 0.005(0.001)$