

## A NEW ADAPTIVE GRID METHOD BASED ON ITERATIVE GRID REDISTRIBUTION

WEIQING REN AND XIAO-PING WANG\*

**Abstract.** We introduce an iterative grid redistribution method based on the variational approach which enables us to gain more precise control of the grid distribution near the regions of large solution variations. The method is particularly effective for solving PDEs with singular solutions (e.g. blow up solutions).

**1. Introduction .** In this paper, we are mainly interested in adaptive mesh redistribution methods based on grid generations. That is, one seeks a change of independent variables (or a mapping from the physical domain to the computational domain), such that, in the new variables, the variation of the concerned quantity (defined by the monitor function) is reduced so that the solution can be resolved by a uniform grid. In the discrete level, this means a uniform grid in the computational domain is mapped into the physical domain so that more grid points are concentrated in the regions of large variations. The question that we are concerned with in this paper is whether it is possible (and how) to determine an optimal coordinate mapping in certain sense, so that we can achieve the best possible behavior in the computation variables.

In one dimension, grid generation is usually done by equidistribution rule. In two (or higher) spatial dimensions, the commonly used mesh generation techniques are based on a variational approach, many of which are derivations of a technique first proposed by Winslow. The functional is chosen so that the minimum is suitably influenced by the solution of the PDE. In these grid generation rules, one hopes to control the resulting mesh behavior from the monitor function. Practically, however, this is very difficult to achieve as we will show in Section 4.

In this paper, we introduce an iterative grid redistribution method. The fundamental idea is that one realizes that a successive application of the mapping (the Winslow mapping) (or the generalization of them) improves the adaptivity of the grids. In fact, our results indicate that the iteration converges to an optimal coordinate mapping in the sense that the monitor function converges to a constant function almost everywhere. With the iteration procedure, we gain control of the mesh distribution near the singular point, i.e., we may achieve desirable grid adaption by controlling the number of mesh iterations. In Section 5, we show how our iterative grid redistribution procedure can be easily implemented in a static adaptive grid redistribution procedure for PDEs.

In Section 6, we show a numerical examples. We apply our method to the non-linear Schrödinger equation. More detailed numerical tests are given in [14] where we also compare our results obtained from the dynamic rescaling method [11][12]. In those numerical examples, we will demonstrate the capability of our method for tracking multiple singularities. as well as being able to resolve solution away from the singularities.

### 2. Various grid generation rules.

---

\*Department of Mathematics, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong.

**2.1. Grid distribution based on the equidistribution principle in one dimension.** The equidistribution principle was introduced by de Boor [7] for solving boundary value problems for ordinary differential equations. It involves selecting mesh points such that some measure of the solution error is equalized over each subinterval. Based on this principle, many moving mesh methods have been developed.

Let  $x$  and  $\xi$  denote the physical and computational coordinates, respectively, on the unit interval  $[0, 1]$ . A one-to-one coordinate transformation between these domains is denoted by

$$(2.1) \quad \begin{cases} x = x(\xi), & \xi \in [0, 1] \\ x(0) = 0, & x(1) = 1 \end{cases}$$

Suppose that a uniform mesh is given on the computational domain by

$$\xi_i = \frac{i}{n}, \quad i = 0, 1, \dots, n$$

where  $n$  is a certain positive integer, and denote the corresponding mesh in  $x$  by  $\{x_0, x_1, \dots, x_n\}$ . For a chosen monitor function  $w(x)$  ( $> 0$ ), which provides some measure of the computational error in the solution  $u(x)$  of the underlying physical PDE, the (one-dimensional) equidistribution principle can be expressed in its integral form as

$$(2.2) \quad \int_0^x w(s) ds = \xi C$$

where

$$(2.3) \quad C = \int_0^1 w(s) ds$$

or equivalently, in the discrete form

$$\int_{x_i}^{x_{i+1}} w(x) dx = \int_{x_{i-1}}^{x_i} w(x) dx \quad \text{for } i = 1, 2, \dots, n-1$$

Differentiating (2.2) once, we obtain a differential form,

$$(2.4) \quad w(x(\xi)) \frac{\partial}{\partial \xi} x(\xi) = C$$

When  $w(x) = \sqrt{1 + u_x^2}$ , the above method is known as the ‘‘arclength method’’.

**2.2. Grid distribution based on the variational principle.** The above equidistribution principles cannot be generalized directly to two or higher dimensions. In fact, equidistribution can only be achieved locally in a certain way [9]. In two (or higher) spatial dimensions, mesh adaption is commonly done using the variational approach, specifically by minimizing a functional of the coordinate mapping between the physical domain and the computational domain. The functional is chosen so that the minimum is suitably influenced by the desired properties of the solution of the PDE itself.

Again, let  $\mathbf{x}$  and  $\xi$  denote the physical and computational coordinates, respectively, on a domain  $\Omega \in \mathbf{R}^d$ . A one-to-one coordinate transformation on  $\Omega$  is denoted by

$$(2.5) \quad \mathbf{x} = \mathbf{x}(\xi), \quad \xi \in \Omega$$

The functionals used in existing variational approaches for mesh generation and adaptation can usually be expressed in the form

$$(2.6) \quad E(\xi) = \int_{\Omega} \sum_{i,j,\alpha,\beta} g^{i,j} \frac{\partial \xi^\alpha}{\partial x^i} \frac{\partial \xi^\beta}{\partial x^j} dx$$

where  $G = (g_{i,j})$ ,  $G^{-1} = (g^{i,j})$  are symmetric positive definite matrices that are monitor functions in a matrix form. The coordinate transformation and the mesh are determined from the Euler-Lagrange equation

$$(2.7) \quad \nabla \cdot (G^{-1} \nabla \xi) = 0.$$

(2.6) and (2.7) are related to the theory of the harmonic map, where the Hamilton-Schoen-Yau theorem guarantees the existence and uniqueness of the mapping with non-vanishing Jacobian.

We note that more terms can be added to the functional (2.6) to control other properties of the mesh, such as orthogonality of the mesh and the alignment of the mesh lines with a prescribed vector field [4].

**2.3. Winslow’s variable diffusion equation.** A special case of (2.6) is Winslow’s variable diffusion method. Winslow [17] suggested a functional of the form

$$(2.8) \quad E(\xi) = \int_{\Omega} \sum_j \frac{1}{w} |\nabla \xi^j|^2 dx$$

where  $w > 0$  is a weight function depending on the physical solution to be adapted, and this corresponds to (2.6) with the monitor function

$$G = wI.$$

The Euler-Lagrange equations whose solution minimizes  $E$  are

$$(2.9) \quad \nabla \cdot \frac{1}{w} \nabla \xi^j = 0,$$

which are diffusion equations.

**3. Properties of the resulting grid distribution in the extreme case .** From equidistribution property in one dimension and the properties of the solution of the Winslow equations, we expect more grid points will be concentrated at the regions of rapid variation of the solution. However, we will show that when the solution becomes singular, such grid adaptivity is actually lost with the above methods.

**3.1. Numerical tests.** We first consider a one dimensional example. Let

$$u(x) = \frac{1}{\varepsilon} e^{-\left(\frac{x-0.5}{\varepsilon}\right)^2}$$

and the monitor function be  $w = \sqrt{1 + u}$ . We show the grid distribution for various  $\varepsilon$  in Fig. 3.1. It is clear from the graph that as one decreases  $\varepsilon$ , the grid distribution is getting worse and worse.

A two dimensional example for function

$$u(x, y) = ce^{-c^2(x^2+y^2)}$$

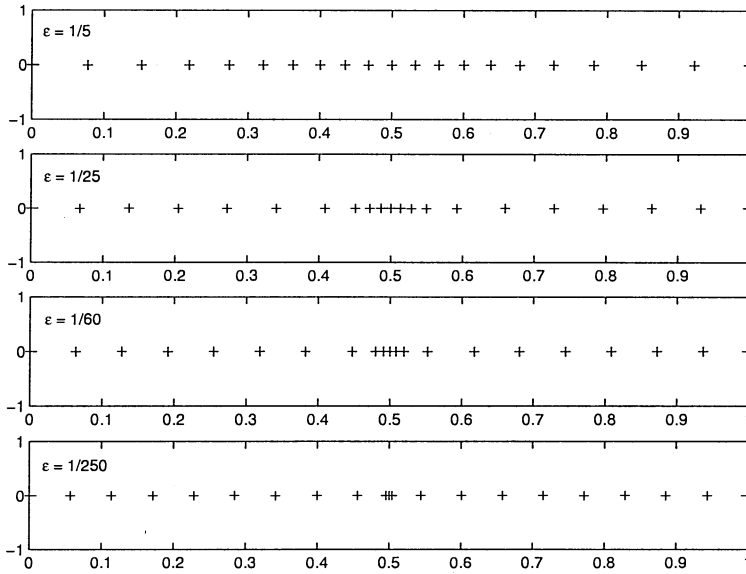


FIG. 3.1. Grid behavior as  $\epsilon$  decreases

shows similar grid behavior. Fig. 3.2 shows the grid distributions obtained from (2.9) for two different monitor functions

$$(a) \quad w(x, y) = \sqrt{1 + |u|^2}, \quad (b) \quad w(x, y) = \sqrt{1 + |\nabla u|^2}.$$

In both cases, we see that, just like in 1-d, grid adaption is achieved for smaller  $c$  and then lost when  $c$  is large.

Other examples with various choices of monitor functions exhibit similar phenomena. Such behavior seems to be generic when the monitor function becomes singular.

**3.2. Asymptotic analysis.** To understand the above phenomena, let's assume that we have a monitor function that is close to a self-similar singularity characterized by a parameter  $\epsilon$

$$w_\epsilon(x) = \sqrt{1 + \frac{1}{\epsilon^k} g\left(\frac{x}{\epsilon}\right)}$$

where  $g(y)$  is positive, has maximum at  $x = 0$  and decays rapidly (say, fast than  $\frac{1}{y^n}$  for a large enough  $n$ ) as  $y \rightarrow \infty$ . We want to study the behavior of the grid as  $\epsilon \rightarrow 0$ . From (2.4), we have

$$(3.1) \quad x_\xi = \frac{C(\epsilon)}{w} = \frac{C(\epsilon)}{\sqrt{1 + \frac{1}{\epsilon^k} g\left(\frac{x}{\epsilon}\right)}}$$

where

$$C(\epsilon) = \int_0^1 \sqrt{1 + \frac{1}{\epsilon^k} g\left(\frac{x}{\epsilon}\right)} dx$$

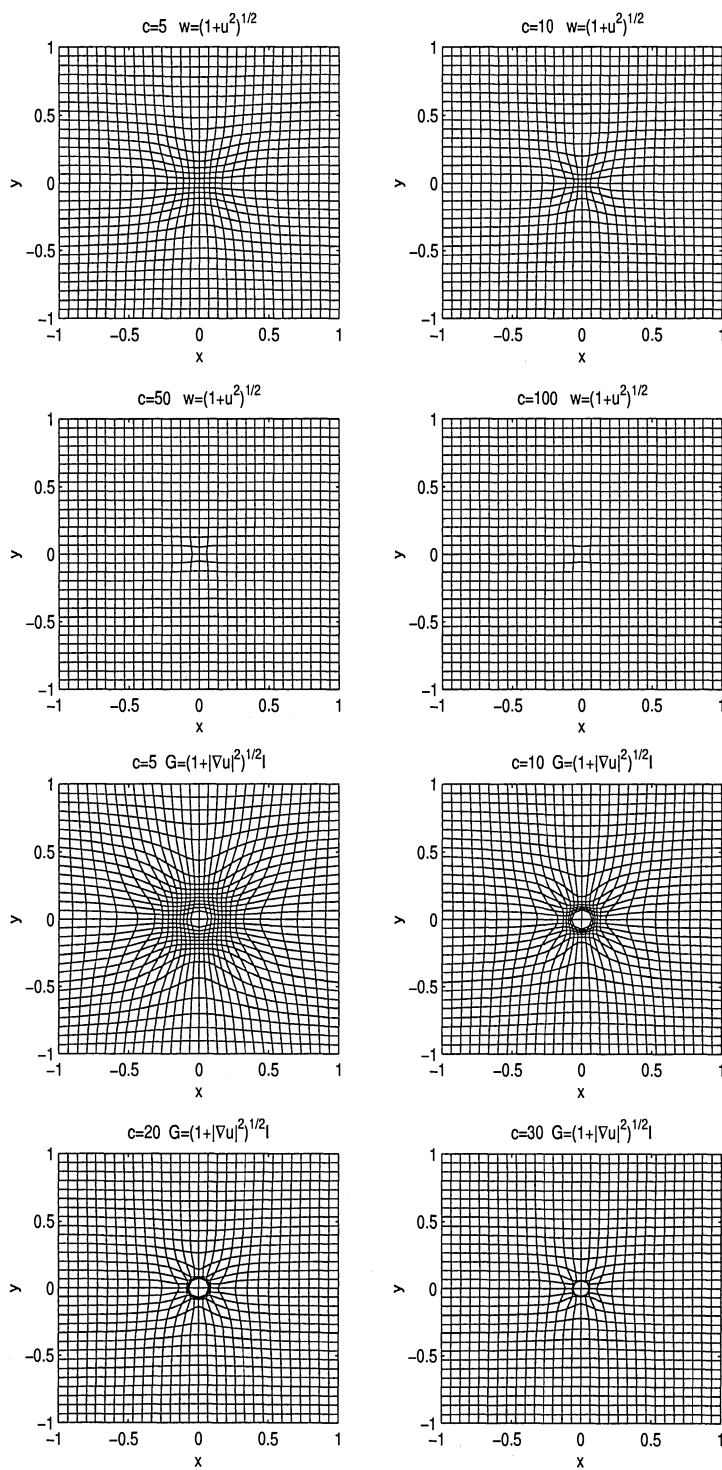


FIG. 3.2. Grid behavior for  $u = ce^{-c^2(x^2+y^2)}$  in 2-d as  $c$  increases

Integration by parts, we have

$$\begin{aligned}
 C(\varepsilon) &= x\sqrt{1 + \frac{1}{\varepsilon^k}g\left(\frac{x}{\varepsilon}\right)} \Big|_0^1 - \int_0^1 \frac{x\frac{1}{\varepsilon^k}g'(x)}{2\sqrt{1 + \frac{1}{\varepsilon^k}g\left(\frac{x}{\varepsilon}\right)}} dx \\
 &= \sqrt{1 + \frac{1}{\varepsilon^k}g\left(\frac{x}{\varepsilon}\right)} - \varepsilon^{1-\frac{k}{2}} \int_0^{\frac{1}{\varepsilon}} \frac{yg'(y)}{2\sqrt{\varepsilon^k + g(y)}} dy
 \end{aligned}$$

Since  $g(y)$  decays rapidly as  $y \rightarrow \infty$ , we have, as  $\varepsilon \rightarrow 0$ ,

$$(3.2) \quad C(\varepsilon) \sim 1 + A\varepsilon^{1-\frac{k}{2}}$$

where

$$A = - \int_0^\infty \frac{yg'(y)}{2\sqrt{g(y)}} dy = \int_0^\infty \sqrt{g(y)} dy.$$

We thus have for  $x \neq 0$  (i.e., away from the singularity) and  $\varepsilon \rightarrow 0$

$$\begin{aligned}
 x_\xi &\rightarrow 1 \text{ when } k < 2 \\
 x_\xi &\rightarrow 1 + A \text{ when } k = 2 \\
 x_\xi &\rightarrow \infty \text{ when } k > 2
 \end{aligned}
 \tag{3.3}$$

This implies that, in the case  $k < 2$ , we have  $\Delta X = \Delta \xi$  away from the singularity. Therefore, as  $\varepsilon \rightarrow 0$ , only grid points very close to 0 are allowed to move, i.e., the grid adaption is very limited near the place that function  $g$  is large. This also shows that one has to know the solution behavior in some detail in order to design the monitor function which can generate the desired grid distribution near the singularity.

**4. An iterative remeshing procedure .** To improve the mesh adaption, we introduce an iterative remeshing procedure. Let us first define the *Winslow mapping*:

$$\mathbf{T} : (\mathbf{x}, u(\mathbf{x})) \rightarrow (\xi, v(\xi))$$

Here

$$\begin{aligned}
 \mathbf{x} &= \mathbf{x}(\xi) \text{ is determined from (2.9) where the monitor function } w(x) \text{ depends on} \\
 &u(x); \\
 v(\xi) &= u(\mathbf{x}(\xi))
 \end{aligned}$$

If the monitor function  $w$  is chosen properly, the resulting mesh should concentrate more grid points in the regions with large variations. This also means that  $v(\xi)$  should be better behaved than the original function  $u(x)$  in the sense that the variation of the monitor function in the new variables is reduced. However, the examples in the previous section show that in some cases, such improvement is very limited. A natural idea to improve further is to repeat the same procedure for  $v(\xi)$ . In fact, this process can be repeated until a satisfactory  $v(\xi)$  is achieved. Based on this intuition, an iterative remeshing procedure is introduced by applying the Winslow mapping  $\mathbf{T}$  iteratively:

- Let  $u^k(\mathbf{x})$  be the function after  $k$  iterations.
- Determine the mapping  $\mathbf{x}^{k+1}(\xi)$  from  $u^k(\mathbf{x})$  according to (2.9) where monitor function  $w^k$  is defined using  $u^k(x)$ .
- Define  $u^{k+1}(\xi) := u^k(\mathbf{x}(\xi))$ .

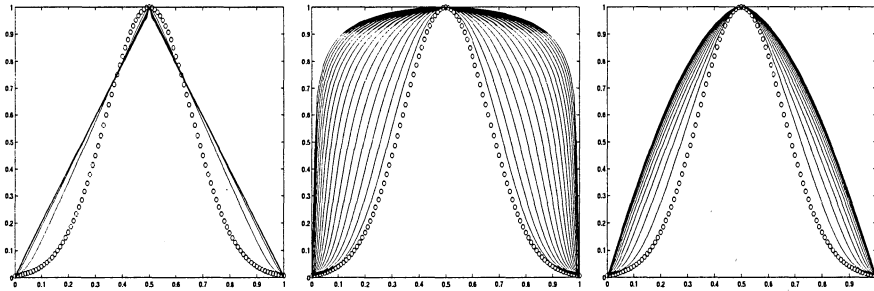


FIG. 4.1. Iterative remeshing for three different monitor functions. Circle represents  $u(x)$  in physical variable.

The results of the iteration is to flatten out the monitor function gradually. In fact, if  $u^k(x)$  and  $x^k(\xi)$  converge, then we must have  $x^k \rightarrow x^*(\xi) = \xi$  and  $u^k \rightarrow u^*(x)$ .

Claim:

$w^k$  converges to a constant function almost everywhere.

This shows that we achieve the maximum adaption for the monitor function. The above claim is verified by many of our numerical examples below, although so far, rigorous proof can only be obtained in some special cases.

EXAMPLE 1: Let  $u(x) = \exp(-20(x - 0.5)^2)$  on  $[0, 1]$ . We apply the above iteration using three different monitor functions:

$$(a) \quad w = \sqrt{1 + u_x^2} \quad (b) \quad w = \sqrt{1 + u^2} \quad (c) \quad w = \sqrt{1 + 0.1u_x^2 + u^2}$$

In Fig. 4.1, we show  $u^k(x)$  for different  $k$ . Figure 4.1a shows that  $u^k(x)$  converges to two straight lines, i.e.,  $|u_x^k(x)|$  converges to constants a.e. Figure 4.1b shows that  $|u^k(x)|$  converges to 1. Fig.4.1c shows that a combination of (a) and (b) gives a limiting function with much better behavior. In all cases, we have  $w \rightarrow Const.$  a.e. as  $k \rightarrow \infty$ .

EXAMPLE 2: Next, we show the capability of improving the mesh adaption by the iterative remeshing in Fig. 4.2. Again for function  $u(x, y) = ce^{-c^2(x^2+y^2)}$  with  $c = 50$ , we see significant improvement (compared to Fig. 3.2) in mesh concentration at the origin as the number of iterations increases.

The above examples also suggest that to achieve better limiting behavior in the computational domain, the monitor function should include both  $|u|$  and  $|\nabla u|$ . Theoretically, the more higher order derivatives are added (although difficult to implement in practice), the better limiting behavior of the function in the computational domain is expected. To see this, let the monitor function be  $w = \sqrt{1 + |u| + |Du| + |D^2u|}$ . Since in the iteration limit, the monitor function  $w$  tends to a constant, which is the integral average of  $w$ , we have that, in the computational variables,  $\max |Du|$ ,  $\max |D^2u|$  are all bounded by the same constant.

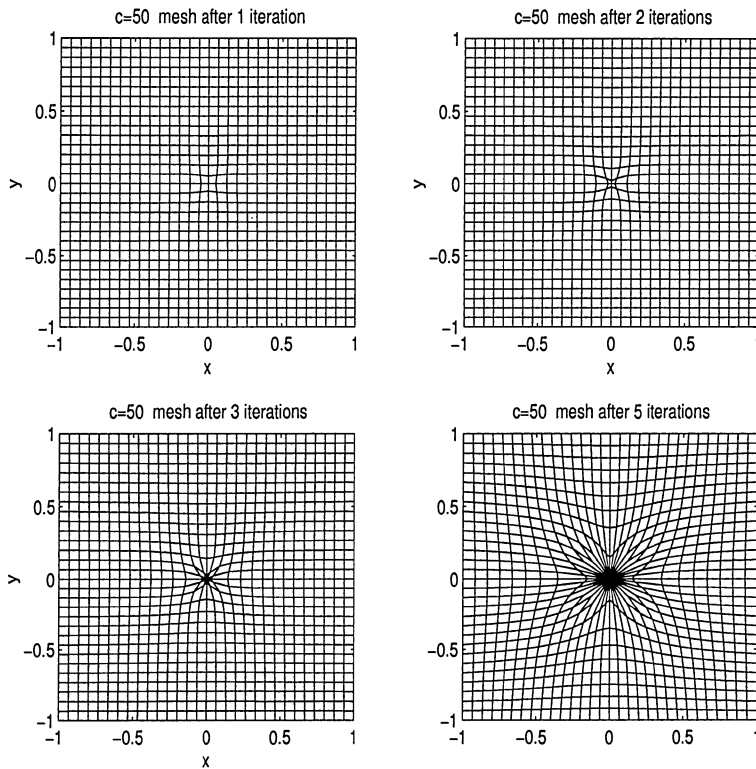


FIG. 4.2. Improved mesh distribution (compare to Fig. 3.2) for  $u = ce^{-c^2(x^2+y^2)}$  ( $c = 50$ ) after several iterations

**5. Adaptive procedure for solving PDEs.** In this section, we design a numerical scheme that incorporates the iterative remeshing into a static adaptive method for solutions of PDEs. For simplicity, we will only formulate the scheme in two dimensions. Generalization to three dimensions is straightforward. Our scheme is static, i.e., we use a fixed grid to solve the PDE in the computational variables until a certain criterion is violated that indicates that the adaption is needed. Then iterative remeshing is used to generate a new grid.

Let's assume that we solve a PDE or system of PDEs of the form

$$(5.1) \quad u_t = F(\mathbf{x}, u, Du, D^2u), \quad \mathbf{x} \in \Omega_p$$

supplemented with initial and boundary conditions.  $D$  is the first order differential operator. The one level grid generation is based on (2.9)

$$(5.2) \quad \begin{cases} \nabla \cdot (\frac{1}{w} \nabla \xi) = 0 \\ \nabla \cdot (\frac{1}{w} \nabla \eta) = 0 \end{cases}$$

which determines the transformation  $(x(\xi, \eta), y(\xi, \eta))$  from the computational domain  $\Omega_c$  to the physical domain  $\Omega_p$ . The choice of the monitor function  $w$  is problem dependent. In most cases,  $w = \sqrt{1 + \alpha|\nabla \bar{u}|^2 + \beta|\bar{u}|^2}$  is good enough for certain constants  $\alpha > 0, \beta > 0$ .

The procedure is described in the following:



- (0) Given an initial condition  $\bar{u}(x, y, 0)$ , the initial grid transforms  $x(\xi, \eta)$ ,  $y(\xi, \eta)$  are determined from the iterative remeshing, which in turn gives an initial condition in the computational domain  $\bar{u}(x(\xi, \eta), y(\xi, \eta), 0)$ .
- (1) Solve the PDE in the computational variables  $\xi, \eta$  with the grid transformation  $x(\xi, \eta)$ ,  $y(\xi, \eta)$  being fixed, until some time  $t^*$  when the solution  $\bar{u}(\xi, \eta, t^*)$  cannot meet a certain criterion.
- (2) Generate a new mesh by the iterative remeshing, starting with  $\bar{u}(\xi, \eta, t^*)$ . The remeshing iteration stops if the criterion in (1) is satisfied. The interpolation is used to move the solution on the new grids.
- (3) Go to (1) to continue the integration.

REMARK 1. The stopping criterion in step (1) depends on the specific problem. In our numerical examples below, the criterion is set so that the maximum amplitude of the gradient is smaller than a given value  $TOL$ . The choice of  $TOL$  is flexible. However, it has to be larger than the integral average of the gradient of the solution over the domain. It is easy to see that a smaller  $TOL$  requires more remeshing iterations. In actual applications, one should include quantities of interests near the singularity in both the monitor function and in the stopping criterion. For example, in the fluid problem, vorticity might be the desired quantity to be included in the monitor function.

REMARK 2. In most of the cases, only one remeshing iteration is needed when we start the iteration with  $\bar{u}(\xi, \eta, t^*)$  in Step 2. Since we always start the iteration from the most recent solution in the computational domain, when the cycle (1)-(3) is repeated  $k$  times, effectively we have at least  $k$  remeshing iterations at  $t^*$  from the solution in the original physical variables.

REMARK 3. Our grid movement method is static, i.e. the grids are held stationary during the evolution of PDEs until the stopping criterion is violated and are shifted to their new positions by our iterative procedure. The solution values are moved from old grid to the new grid by interpolation. The interpolation is carried out on the uniform mesh and using cubic polynomial interpolations. As pointed out in Remark 2, our iterative procedure was carried out gradually as the solution evolves toward the singularity and solution behavior in the computational domain is always controlled by the stopping criterion (e.g.  $\max |\nabla \mathbf{u}| \leq TOL$ ). Therefore interpolation errors are also controlled.

**6. Numerical examples.** In this section, we show one example which demonstrates capability of our method for handling multiple singularities. More numerical tests can be found in [14].

We solve the nonlinear Schrödinger equation(NLS)

$$(6.1) \quad \begin{cases} i\psi_t + \Delta\psi + |\psi|^2\psi = 0, & (x, y) \in \Omega_p, t > 0, \\ \psi(x, y, t)|_{\partial\Omega_p} = 0 \end{cases}$$

in two dimensions.

Let  $(x(\xi, \eta), y(\xi, \eta))$  be the spatial coordinate transformations. Both  $\Omega_p$  (the physical domain) and  $\Omega_c$  (the computational domain) are chosen to be  $[-1, 1] \times [-1, 1]$ . As in the dynamic rescaling, we also rescale the time from  $t$  to  $\tau$  as

$$(6.2) \quad \frac{d\tau}{dt} = \frac{1}{\lambda^2(t)}$$

Let

$$\psi = \frac{1}{L(t)}\phi,$$

where  $L(t)$  is a scaling factor chosen to be

$$(6.3) \quad L(t) = \frac{1}{\max_{(x,y) \in \Omega_p} |\psi(x,y,t)|}.$$

To balance the coefficients in the transformed equation, we choose  $\lambda = L^\sigma$ . In the coordinate system  $(\xi, \eta, \tau)$ , the NLS becomes

$$(6.4) \quad \phi_\tau - \frac{L_\tau}{L}\phi - i(\lambda^2 \Delta_B \phi + |\phi|^2 \phi) = 0$$

together with

$$(6.5) \quad L_\tau = L^{2\sigma+1} \text{Im}(\phi^* \Delta_B \phi)|_{(\xi_0, \eta_0)}$$

where  $(\xi_0(t), \eta_0(t))$  is the maximum point of  $|\phi(\xi, \eta, t)|$ , and

$$\Delta_B \phi = \frac{1}{J} \left\{ \frac{\partial}{\partial \xi} \left( \frac{b_{22}\phi_\xi - b_{12}\phi_\eta}{J} \right) + \frac{\partial}{\partial \eta} \left( \frac{b_{11}\phi_\eta - b_{12}\phi_\xi}{J} \right) \right\},$$

$$b_{11} = x_\xi^2 + y_\xi^2, \quad b_{12} = x_\xi x_\eta + y_\xi y_\eta, \quad b_{22} = x_\eta^2 + y_\eta^2.$$

$J$  is the Jacobian of the coordinate transformation.

In the grid redistribution, the monitor function is taken to be  $w(\xi, \eta) = (1 + 2|\phi(\xi, \eta)|^2 + |\nabla\phi(\xi, \eta)|^2)$ . The criterion is set so that the maximum amplitude of the gradient is smaller than a given value of  $TOL$ . Two values of  $TOL$ , 5 and 7, are used for our computations.

we use the initial condition

$$\psi(x, y, 0) = 20(e^{-20((x+0.5)^2+y^2)} + e^{-20((x-0.5)^2+y^2)})$$

with two maximum points. our results show that the solution blows up at two points. Such a calculation cannot be done by the dynamic rescaling method because one can only rescale around one point. We have not seen other successful calculations on such problems. We solve the equation with  $160 \times 160$  grid points in the computational domain and  $TOL = 7$ . Fig. 6.1 and 6.2 show solutions in computational and physical variables respectively at  $\tau = 76.40$  ( $t = 0.005651$ ) when the maximum of the solution reaches  $1.488 \times 10^5$ . The grid distribution at the same time is shown in Fig. 6.3 and 6.4. The blow up structure of each singularity is the same as that of the solution with single blow up point.

**7. Conclusions.** We have introduced an iterative grid redistribution method for computing singularities in multiple dimensions. The major improvement of our method over the conventional grid redistribution method is that we gain control of the mesh distribution around the singularity, which the usual mesh generation procedure based on equidistribution or variational approach is unable to achieve. Our method is also rather general: it is capable of handling multiple singularities and requires little

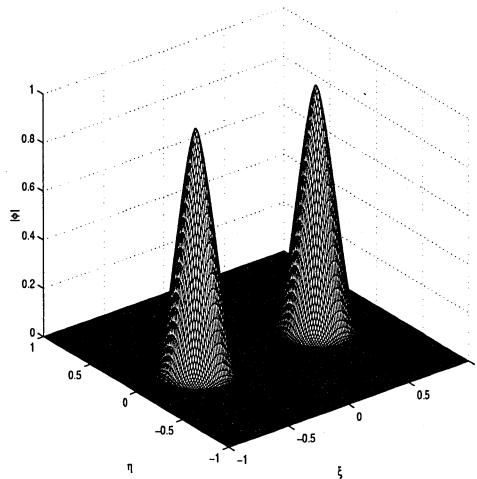


FIG. 6.1.  $\phi(\xi, \eta, \tau)$  in  $\Omega_c$  at  $\tau = 76.40(t = 0.005651)$ ,  $1/\lambda = 1.4880e + 05$  (NLS with initial value (ii))

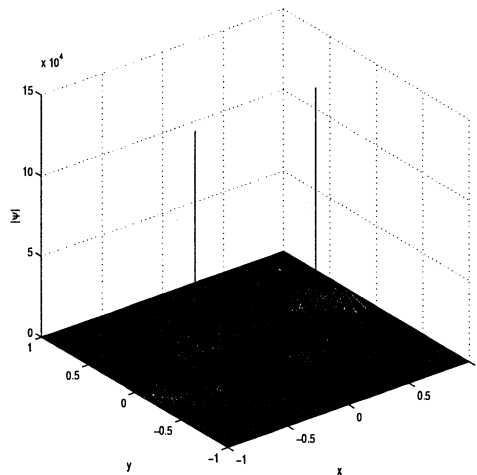


FIG. 6.2.  $\psi(x, y, t)$  in  $\Omega_p$  at  $t=0.005651$  corresponding to Fig. 6.1 (NLS with initial value (ii))

information in advance about the locations and structure of the singularities. Moreover, it is relatively easy to implement compared to many grid refinement methods. Although our examples in this paper are in two dimensions, it is straightforward to generalize the method to the three dimensions. Many three dimensional problems exhibit richer behavior in singularities such as line singularities and transitions from line singularities to point singularities. Our method would be more desirable in such cases. Research on the three dimensional problems is currently being conducted.

**Acknowledgments:** XPW wishes to thank George Papanicolaou for his constant encouragement and for some useful discussions. We thank Qiang Du, Weinan E, Gađi Fibich and Zhouping Xin for some helpful suggestions and discussions. This work is supported in part through the Research Grant Council of Hong Kong by grant HKUST 6165/97P.

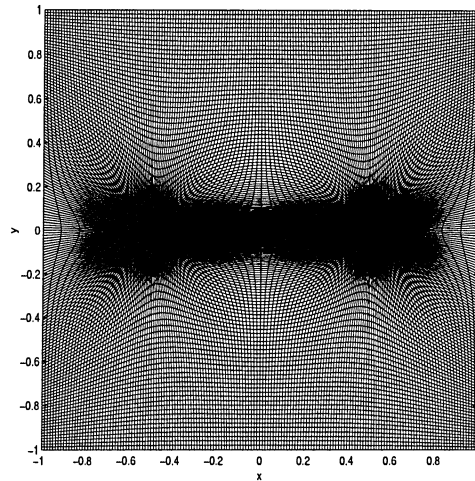


FIG. 6.3. Mesh in  $\Omega_p$  at  $t=0.005651$  ( $\tau = 76.40$ ) corresponding to Fig. 6.2 (NLS with initial value (ii))

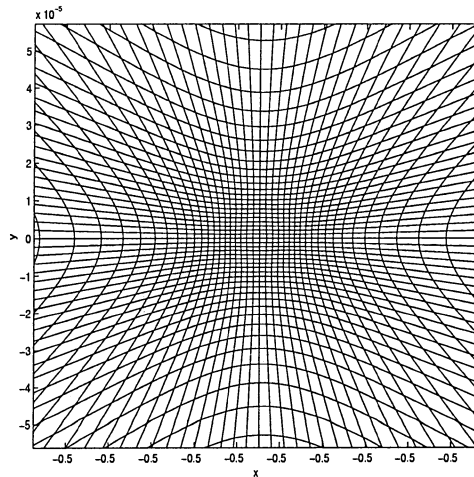


FIG. 6.4. Enlarged view of the above mesh in  $\Omega_p$  around one blow-up point at  $t=0.005651$  ( $\tau = 76.40$ ) corresponding to Fig. 6.2 (NLS with initial value (ii))

#### REFERENCES

- [1] D.A. ANDERSON, *Grid cell volume control with an adaptive generator*, Appl. Math. and Comp., 35 (1990), pp. 209–217.
- [2] M. BERGER AND P. COLELLA, *Local adaptive mesh refinement for shock hydrodynamics* J. Comput. Phys., 82 (1989), pp. 64–84.
- [3] J. BRACKBILL AND J. SALTZMAN, *Adaptive zoning for singular problems in two dimensions*, J. Comput. Phys., 46 (1982), pp. 342–368.
- [4] J. BRACKBILL, *A adaptive grid with directional control*, J. Comput. Phys., 108 (1993), pp. 38–50.
- [5] M.P. BRENNER, L.S. LEVITOV AND E.O. BUDRENE, *Physical mechanisms for chemotactic pattern formation by bacteria*, Biophysical Journal, 74 (1998), pp. 1677–1693.
- [6] M.P. BRENNER, P. CONSTANTIN, L.P. KADANOFF, A. SCHENKEL AND S.C. VENKATARAMANI, *Diffusion, attraction and collapse*, Preprint.
- [7] C. DE BOOR, *Good approximation by splines with variable knots II*, in Springer Lecture Notes

Series 363, Springer-Verlag, Berlin, 1973.

- [8] P.K. MOORE AND J.E. FLAHERTY, *J. Comput. Phys.*, 98 (1992), pp. 54.
- [9] W.Z. HUANG AND D.M. SLOAN, *A simple adaptive grid method in two dimensions*, *SIAM J. Sci. Comput.*, Vol. 15, No. 4 (1994), pp. 776–797.
- [10] W. HUANG AND R. D. RUSSELL, *Moving Mesh Strategy Based upon Gradient Flow Equation for two Dimensional Problems*, *SIAM J. Sci. Comput.*, to appear.
- [11] M. LANDMAN, G.C. PAPANICOLAOU, P.L. SULEM, C. SULEM AND X.P. WANG, *Stability of Isotropic Singularities for the Nonlinear Schrödinger Equation*, *Physica, D* 47 (1991), pp. 393–415.
- [12] D.W. MCLAUGHLIN, G. PAPANICOLAOU, C. SULEM, P.L. SULEM, *Phys. Rev.*, A 34 (1986), pp. 1200.
- [13] K. MILLER AND R.N. MILLER, *Moving finite elements I*, *SIAM J. Numer. Anal.*, 18 (1981), pp. 1019–1032.
- [14] W. REN AND X.P. WANG, *An iterative grid redistribution method for singular problem in multiple dimensions*, to appear in *J. Comput. Phys.*
- [15] M.M. RAI AND D.A. ANDERSON, *Grid Evolution in Time Asymptotic Problems*, *J. Comput. Phys.*, 43 (1981), pp. 327–344.
- [16] V. SHVETS, N.E. KOSMATOV AND B.J. LEMESURIER, *On collapsing solutions of the nonlinear Schrödinger equation in the supercritical case*, in *Singularities in Fluids, Plasmas and Optics*, Edited by R. Caflisch and G. Papanicolaou, NATO ASI Series Vol. 404, Kluwer Academic Publishers.
- [17] A. WINSLOW, *Numerical solution of the quasi-linear Poisson equation*, *J. Comput. Phys.*, 1 (1966), pp. 149–172.
- [18] H. ZHANG AND M.K. MOALLEMI, *MAGG - A Multizone Adaptive Grid Generation Technique for Simulation of Moving and Free Boundary Problems*, *Numerical Heat Transfer*, B27 (1995), pp. 255–276.

