COMPARISON OF SOME MOVING MESH METHODS IN HIGHER DIMENSIONS

SHAOHUA CHEN *, ROBERT D. RUSSELL †, AND WENTAO SUN ‡

Abstract. We discuss recent progress made in developing basic grid generation and moving mesh methods for solving PDEs and compare the advantages and disadvantages for some of these methods. In particular, we contrast the standard MMPDE (moving mesh partial differential equation) methods developed in [12] and [8] and some improvements developed by Huang, especially a new variational version of the grid formulation [11]. An alternative approach investigated by Ceniceros and Hou [9] is also discussed. Several numerical examples are presented by way of comparison.

1. Introduction. While the mesh formulation problem and mesh generation are generally well understood in 1D (one spatial dimension) and there are reliable moving mesh methods based upon such formulations for solving 1D time dependent PDEs, significant challenges remain for higher dimensions. Here, we investigate these topics for 2D problems, although most conclusions about the relative merits of the methods are valid for 3D as well. In the numerical simulations, we focus on blowup solutions of nonlinear heat equations and the nonlinear Schrödinger equation (NLS).

Numerical studies with analysis for blowup problems in higher dimensions have to date been very limited. Ren and Wang [21] use a method based on an iterative procedure for the Winslow map to statically redistribute an adaptive mesh for some blowup problems where the location of the singularity is fixed. Ceniceros and Hou [9] develop a very simple but efficient dynamically adaptive mesh generator for time-dependent problems. More recently, the standard variational approach for mesh adaptation [12] and [8] and a new variational form of Huang [11] have been discussed in the context of blowup problems [6]. A limited computational comparison of some aspects of these methods is done in our paper.

2. Implementation of MMPDEs.

2.1. The Basic Moving Mesh Methods. Let Ω_p be a physical domain in R^2 with corresponding computational domain Ω_c and suppose a mesh on Ω_c is given. Then, to construct a mesh on Ω_p, it suffices to define a one-to-one mapping $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi})$ from Ω_c onto Ω_p, or equivalently its inverse mapping $\boldsymbol{\xi} = \boldsymbol{\xi}(\boldsymbol{x})$ from Ω_p onto Ω_c.

A standard variational approach to determine the map involves minimizing a functional of the form

$$I[\boldsymbol{\xi}] = \frac{1}{2}\int_{\Omega_p} \left(\nabla\xi^T G_1^{-1}\nabla\xi + \nabla\eta^T G_2^{-1}\nabla\eta\right) d\boldsymbol{x}, \tag{2.1}$$

where $\boldsymbol{x} = (x, y)$, $\boldsymbol{\xi} = (\xi, \eta) = (\xi^1, \xi^2)$, $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})^T$ and G_1, G_2 are 2×2 symmetric positive definite matrices called *monitor functions* (see [1], [5], [7], [8], [10], [12] and [18]). The minimizer is determined by the Euler-Lagrange equations

$$\nabla(G_1^{-1}\nabla\xi) = 0,\ \nabla(G_2^{-1}\nabla\eta) = 0. \tag{2.2}$$

This method has proven to generally be quite effective for mesh generation. For example, if we generate a mesh adapting to the function

$$u(x, y) = 20\sin^2(2\pi x)\sin^2(\pi y) \tag{2.3}$$

*Department of Mathematics, University College of Cape Breton (*george_chen@uccb.ca*). This author was supported in part by NSERC (Canada) under Grant OGP-00255307.,

†Department of Mathematics, Simon Fraser University, Burnaby, BC V5A 1S6, Canada (*rdr@cs.sfu.ca*). This author was supported in part by NSERC (Canada) under grant OGP-0008781.,

‡Department of Mathematics, Simon Fraser University, Burnaby, BC V5A 1S6, Canada (*wtsun@pims.math.ca*). The author was supported in part by NSERC (Canada) through Grant OGP-0008781.

on the physical domain $[0,1] \times [0,1]$ by choosing the monitor function

$$G_1 = G_2 = \lambda I + \lambda \boldsymbol{v}_1 \boldsymbol{v}_1^T + \frac{1}{\lambda} \boldsymbol{v}_2 \boldsymbol{v}_2^T \tag{2.4}$$

with

$$\lambda = \sqrt{1 + c|\nabla u|^2 + u^4}, \quad \boldsymbol{v}_1 = \boldsymbol{v}_2^{\perp} = \nabla u / |\nabla u|,$$

then the mesh concentrates well around the two peaks (see Fig. 1). Unfortunately, for blowup problems

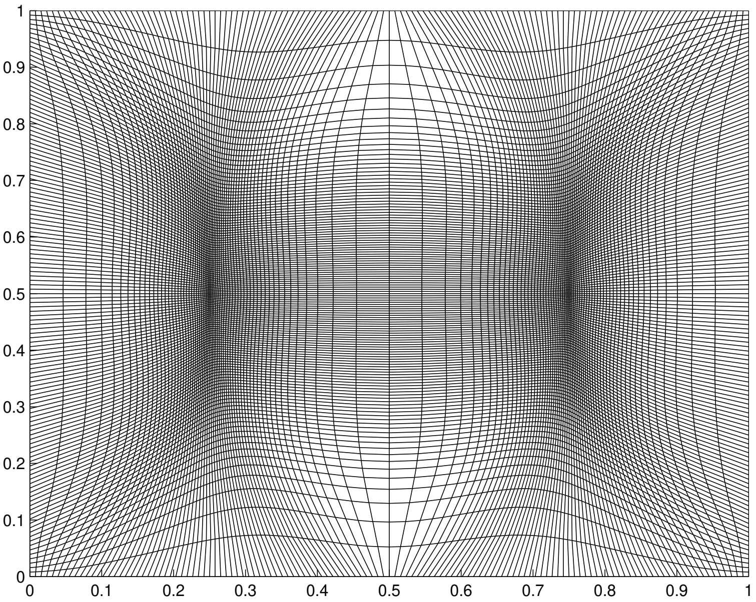


FIG. 2.1. *Initial mesh generated by the standard variational form.*

it is known that this method does not generally provide adequate mesh concentration as the solution values grow [6]. In our computational experience (with implementation details described below), when the amplitude of a solution is around 10^6, the mesh fails to concentrate around the peaks, and some mesh lines become irregular.

2.2. The New Variational Approach. W. Huang [11] has developed a more robust and error-oriented mesh adaptation procedure. The variational form involves two key terms, the isotropy (or conformity) and uniformity (or equidistribution) controls, each of which can be analyzed separately. This new functional in R^2 is

$$I[\boldsymbol{\xi}] = \theta \int_{\Omega_p} \sqrt{g} \left(\sum_{i=1}^{2} (\nabla \xi^i)^T G^{-1} \nabla \xi^i \right)^q d\boldsymbol{x} + (1-2\theta) 2^q \int_{\Omega_p} \frac{\sqrt{g}}{(J\sqrt{g})^q} d\boldsymbol{x}, \tag{2.5}$$

where J and g are the determinants of the Jacobian matrix $(\partial \boldsymbol{x})/(\partial \boldsymbol{\xi})$ and the monitor function G, respectively. Denoting $\beta = \sum_i (\nabla \xi^i)^T \bar{G}^{-1} \nabla \xi^i$ with $\bar{G} = G/g^{1/(2q)}$, the minimizer satisfies the Euler-Lagrange equation

$$\nabla \cdot \left(\theta q \beta^{q-1} \bar{G}^{-1} \nabla \xi^i + (1-2\theta) q 2^{q-1} \sqrt{g} \frac{1}{(J\sqrt{g})^q} \frac{\partial \boldsymbol{x}}{\partial \xi^i} \right) = 0, \quad i = 1, 2. \tag{2.6}$$

Taking $\boldsymbol{\xi}$ as the independent variables, i.e., using $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi})$ instead of $\boldsymbol{\xi} = \boldsymbol{\xi}(\boldsymbol{x})$, we have

$$\sum_{j=1}^{2} \frac{\partial}{\partial \xi^j} J (\boldsymbol{a}^i)^T \left[\theta q \beta^{q-1} \bar{G}^{-1} \nabla \xi^i + (1-2\theta) q 2^{q-1} \sqrt{g} \frac{1}{(J\sqrt{g})^q} \frac{\partial \boldsymbol{x}}{\partial \xi^i} \right] = 0, \quad i = 1, 2. \tag{2.7}$$

where

$$\boldsymbol{a}^1 = \nabla \xi^1 = \frac{1}{J} \begin{pmatrix} y_\eta \\ -x_\eta \end{pmatrix}, \quad \boldsymbol{a}^2 = \nabla \xi^2 = \frac{1}{J} \begin{pmatrix} -y_\xi \\ x_\xi \end{pmatrix}.$$

For a nonconservative form of (2.7), see eqn. (22) of [11].

2.3. Some Implementation Details. To construct a so-called moving mesh PDE (MMPDE) corresponding to (2.7), we replace it by a parabolic system (with 0 replaced by $\frac{\partial \boldsymbol{x}}{\partial t}$ in (2.7)) – as in eqns. (24) - (27) of [10] with $\tau = 1$ and Δt fixed, except with the coefficients in eqn. (27) replaced by the appropriate ones for this new variational form – see eqn. (22) of [11] [1].

We use standard central finite differences for the spatial discretization and the backward Euler scheme for the time integration for the parabolic MMPDE and provide the flexibility to choose a combination of the harmonic and Winslow monitor functions:

$$G = c_1 \lambda I + c_2 (\lambda \boldsymbol{v}_1 \boldsymbol{v}_1^T + \frac{1}{\lambda} \boldsymbol{v}_2 \boldsymbol{v}_2^T), \tag{2.8}$$

with

$$\lambda = \sqrt{1 + c|\nabla u|^2 + f'^2(u)}, \;\; \boldsymbol{v}_1 = \boldsymbol{v}_2^\perp = \nabla u / |\nabla u|,$$

where u is the physical solution at time t and $f(u)$ is the nonlinear term in the physical PDE. In order to move meshes smoothly in time, we smooth the monitor functions in both the spatial and temporal variables. As in [13], we apply the following spatial smoothing three times at each time level:

$$G^n_{i,j} \leftarrow \frac{4}{16} G^n_{i,j} + \frac{2}{16} \left(G^n_{i+1,j} + G^n_{i-1,j} + G^n_{i,j+1} + G^n_{i,j-1} \right) \tag{2.9}$$

$$+ \frac{1}{16} \left(G^n_{i+1,j+1} + G^n_{i-1,j+1} + G^n_{i+1,j-1} + G^n_{i-1,j-1} \right). \tag{2.10}$$

In addition, we apply the following temporal smoothing

$$G^{n+1}_{i,j} = \alpha G^{n+1}_{i,j} + (1-\alpha) G^n_{i,j}. \tag{2.11}$$

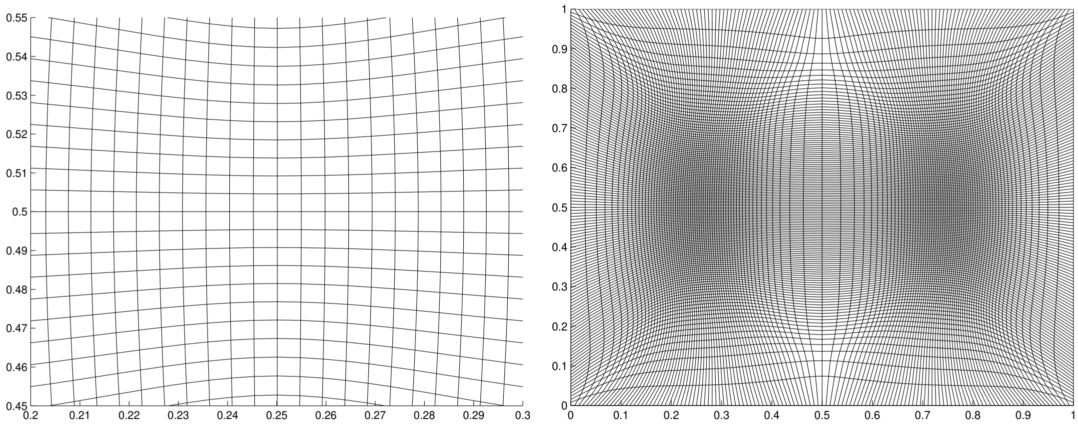
once with $\alpha = 0.7$.

The new variational form has many advantages as well as some disadvantages. Finding a suitable choice of $\theta \in (0, 0.5]$ can be critical in solving time dependent PDEs. If a solution eventually develops an isotropy property, we can choose $\theta \in [0.1, 0.3]$; otherwise, we have found $\theta \in (0.3, 0.45]$ to work well. We may also choose variable $\theta = \theta(t)$. For θ close to 0.5, the mesh can fail to adequately move towards a solution peak, behaving like the standard variational form. If θ is close to 0, even if the mesh is close to the peak, the mesh curves can be concave in the opposite direction (for the initial function (2.3), see the resulting mesh in Fig. 2.2(a)).

The index $q \geq 1$ in (2.7) can also play an important role in mesh generation when the blowup time t is approached. For $q \in [3, 4]$, the mesh keeps its symmetry longer than for $q \in [1, 2]$. As a result, the solution can be computed closer to blowup since the two peaks have essentially the same amplitudes (so neither dominates).

In conclusion, the new variational method has at least three advantages over the standard one: (i) more easily controlling the mesh movements; (ii) obtaining larger amplitude solutions before breakdown of the numerics, especially for problems with isotropy; and (iii) keeping the mesh symmetric even as the amplitude of the physical solution becomes large. There are two main disadvantages: (i) it can be more costly (due partly to the highly nonlinear form of the modified MMPDEs); and (ii) for some special initial functions, such as the one defined in (2.3), the outer mesh lines can move in the wrong directions (see Fig. 2.2(b)) for any θ and q. Apparently, the additional term $\sqrt{g}$ in the first term of the new variational form (2.5) can be responsible for much of this difficulty, although more comprehensive numerical investigation is necessary.

[1] We are most grateful to Weizhang Huang for his detailed suggestions which were incorporated in our implementation described herein and to Weiming Cao for aid during much of the actual coding.



(a) $\theta = 0.1$ and $q = 3$. The vertical curves near one of the two peaks are concave outward.

(b) $\theta = 0.4$ and $q = 3$. The vertical curves near the vertical boundaries are concave outward.

FIG. 2.2. *Initial mesh generated by new variational form.*

3. An Alternative Approach. Here, we discuss the alternative approach for mesh adaptation used by Ceniceros and Hou [9].

3.1. Basic Mesh Generation Method. In [9], the method is only considered for problems where the computational domain coincides with a convex physical domain. A dynamically adaptive mesh generator for time dependent 2D problems is developed for which the MMPDEs have certain desirable properties: Their formulation is done directly in terms of $\boldsymbol{x}(\boldsymbol{\xi})$, with the variational form for the x and y coordinates

$$I[\boldsymbol{\xi}] = \frac{1}{2} \int_{\Omega_c} \nabla \boldsymbol{x}^T G \nabla \boldsymbol{x} d\boldsymbol{\xi}. \tag{3.1}$$

The resulting MMPDEs are based on the simple elliptic generator

$$\nabla_{\boldsymbol{\xi}} \cdot (G \nabla_{\boldsymbol{\xi}} \boldsymbol{x}) = 0, \tag{3.2}$$

called the Length generator [17]. The resulting MMPDE is

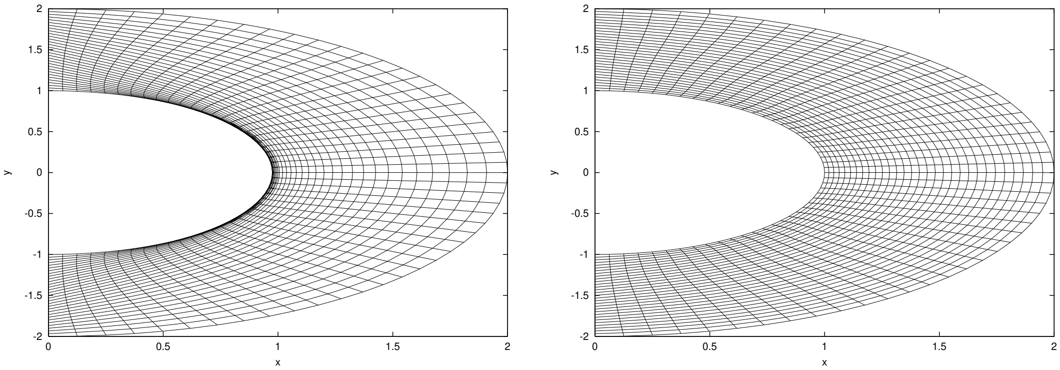
$$\frac{\partial \boldsymbol{x}}{\partial t} = \nabla_{\boldsymbol{\xi}} (G \nabla_{\boldsymbol{\xi}} \boldsymbol{x}), \tag{3.3}$$

which we discretize by

$$\frac{\boldsymbol{x}^{n+1} - \boldsymbol{x}^n}{\Delta t} = \nabla'_h (G^n \nabla'_h \boldsymbol{x}^n) + \alpha (\Delta'_h \boldsymbol{x}^{n+1} - \Delta'_h \boldsymbol{x}^n), \tag{3.4}$$

where $\alpha = \max G^n$, h and Δt are time and spatial steps, respectively, and Δ'_h is the numerical derivative in the computational domain. The solution of (3.4) can be easily computed using a fast method for a linear Laplace operator, viz., an FFT for a rectangular mesh. The final term in (3.4), found in [9] to improve stability when added to the system, can in fact be proven to guarantee stability of the scheme under general conditions.

3.2. Comparison with Other Moving Mesh Methods. Unfortunately, a formulation made directly in terms of maps $\boldsymbol{x}(\boldsymbol{\xi})$ is known to often have difficulty with folding for nonconvex domains, and here we investigate its potential for more complex nonconvex domains. For the case where Ω_p is a



(a) Mesh generated by MMPDE (3.2) (b) Mesh generated by MMPDE (3.5)

FIG. 3.1. *Meshes generated by MMPDEs (3.2) and (3.5)*

horseshoe-like (nonconvex) region, Ω_c is a unit square, and $G = 1$, numerical results are are shown in Fig. 3.1(a).

In an attempt to overcome folding, we have investigated adding terms to the variational form as suggested in [11] and [16] to improve the mesh qualities. In order to keep the advantage of using a fast FFT algorithm, the terms added to the MMPDE usually need to be source terms so that the simple structure of the linearity of the MMPDE is maintained, and this can potentially affect the stability of the scheme. In Fig. 3.1(b), we give an example mesh generated by the variational form

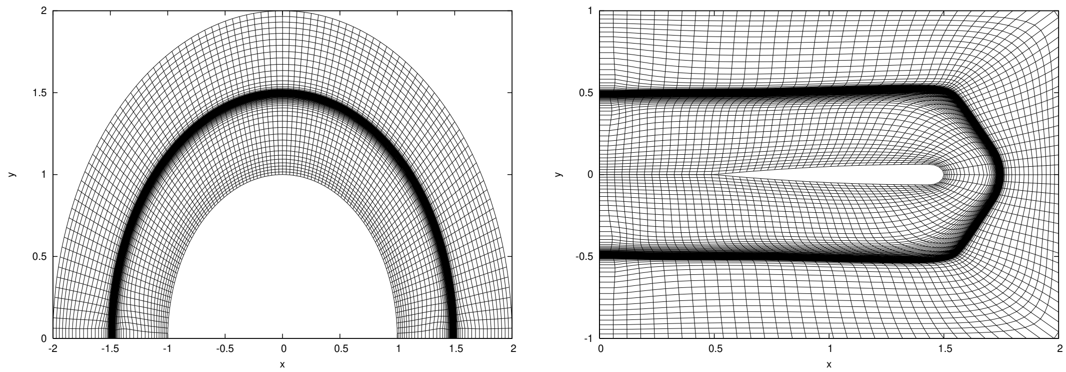
$$I[\boldsymbol{\xi}] = \alpha \int_{\Omega_c} \nabla \boldsymbol{x}^T \nabla \boldsymbol{x} d\xi d\eta + \beta \int_{\Omega_c} \frac{1}{J^q} d\xi d\eta + \gamma \int_{\Omega_p} \|\mathcal{J}^T \mathcal{J}\|_F^p dx dy \tag{3.5}$$

where $\mathcal{J}$ is Jacobian matrix, J is its determinant, and $\|\cdot\|_F$ is the Frobenius norm. This is a direct analogue of what is done to modify the variational form for the standard methods (written in terms of the physical variables instead of the computational variables in [11]), where normally $q > 2$ and $p \geq 1$. Note that the Euler-Lagrange equation for the first term of the variational equation (3.5) gives the MMPDE in [9] if $G = 1$. Adding the other two terms in the variational equation as source terms, we maintain the needed structure for fast computation. For mesh adaptation, the variational functional is combined with a proper monitor function depending upon the physical solution, e.g. as discussed in [11] and [16].

In spite of the fact that folding can still occur, we believe that this generator can nevertheless be useful for some problems where an algebraic generator alone fails to produce a smooth grid. Specifically, evolving the mesh with the MMPDE of [9] before it folds, the integration can be stopped and the computed grid used as an initial value for a more sophisticated grid generator.

For nonconvex domains, the variational forms (2.1) and (2.5) have been used extensively. Their potential, however, relies upon one's being able to numerically solve the mesh equations efficiently. Interestingly, it is claimed in [17] that the Euler-Lagrange equations for some important variational grid generation equations are generally solved in a nonconservative form since the fully conservative approach based upon some numerical schemes do not converge. Unfortunately, a non-conservative numerical scheme may produce artificial sources and sinks, changing the balance of some physical variables, especially on relatively coarse grids. Despite these potential limitations, these nonconservative MMPDEs have proven to be useful for many problems (see, e.g., [8]) where efficient and reliable solutions can be obtained using a fast linear algebra solver such as GMRES combined with a good preconditioner. To illustrate this, in Fig. 4 we give some adaptive meshes computed on some relatively complex domains, using the simple variational form (2.1) for the function $u(\xi, \eta) = 10^6 \sinh(\frac{\eta - 0.5}{.06})$ and (2.4).

While a more careful comparison is warranted, for problems with nonconvex Ω_p, due to the robustness of the methods from (2.1) and (2.5) our preference is for them over using (3.1) directly.



(a) Adaptive mesh on horseshoe domain (b) Adaptive mesh on airfoil domain

FIG. 3.2. *Some adaptive meshes on some concave domain generated by Winslow equidistribution principle*

4. Numerical Examples for Time-Dependent PDEs. In this section, we consider blowup problems of two kinds: the nonlinear heat equation and the nonlinear Schrödinger equation (NLS). Such blowup solutions are of great importance when considering nonlinear evolution processes. Blowup can represent an explosive instability of the process due to the effect of the nonlinear terms, when the magnitude of some quantity (e.g. temperature, pressure, density etc.) tends to infinity in a finite time. Such phenomena have important applications in many areas, such as laser beam theory, plasma physics, chemical reactor theory and combustion theory (see [22], [23], [20] and [4]).

Since blowup problems can have very subtle blowup structure and it is impossible to obtain exact solutions, numerical computations become very important. To compute such singular behavior accurately, it is essential to use a numerical method which adapts the spatial mesh as the singularity develops [2]. However, due to complicated solution structure and mesh generation complexities in higher dimensions, it can prove very challenging to develop a robust and computationally stable adaptive mesh strategy.

4.1. Implementation of Moving Mesh Algorithm. Following [13], we express the physical PDE in terms of the computational coordinates (ξ, η) and then alternately integrate in time the MMPDE and the transformed PDE. For simplicity, Ω_p and Ω_c are the same in the examples below.

Consider the heat equation

$$u_t = \Delta u + f(u) \quad \text{in } \Omega_p = [a, b] \times [a, b], \tag{4.1}$$

$$u|_{\partial\Omega_p} = 0 \tag{4.2}$$

$$u(x, y, 0) = \phi(x, y). \tag{4.3}$$

Denoting $\bar{u}(\xi, \eta) = u[x(\xi, \eta), y(\xi, \eta)]$, in the computational domain we have

$$\bar{u}_t|_{\text{fixed } \boldsymbol{\xi}} = u_t|_{\text{fixed } \boldsymbol{x}} + \frac{1}{J}\begin{pmatrix} \frac{\partial \bar{u}}{\partial \xi} & \frac{\partial \bar{u}}{\partial \eta} \end{pmatrix} \begin{pmatrix} y_\eta & -y_\xi \\ -x_\eta & x_\xi \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix}$$

and

$$\Delta u = \frac{1}{J}\begin{pmatrix} y_\eta & -y_\xi \\ -x_\eta & x_\xi \end{pmatrix} \begin{pmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \end{pmatrix} \frac{1}{J}\begin{pmatrix} y_\eta & -y_\xi \\ -x_\eta & x_\xi \end{pmatrix} \begin{pmatrix} \frac{\partial \bar{u}}{\partial \xi} \\ \frac{\partial \bar{u}}{\partial \eta} \end{pmatrix} \tag{4.4}$$

$$= A\bar{u}_{\xi\xi} + B\bar{u}_{\xi\eta} + C\bar{u}_{\eta\eta} + D\bar{u}_\xi + E\bar{u}_\eta, \tag{4.5}$$

where $J = x_\xi y_\eta - y_\xi x_\eta$ is the Jacobian of the coordinate transformation and

$$A = \frac{1}{J^2}\left(y_\eta^2 + x_\eta^2\right), \quad B = -\frac{2}{J^2}\left(y_\eta y_\xi + x_\eta x_\xi\right), \quad C = \frac{1}{J^2}\left(y_\xi^2 + x_\xi^2\right),$$

$$D = \frac{1}{J^2}\left(x_\eta x_{\xi\eta} - x_\xi x_{\eta\eta} + y_\eta y_{\xi\eta} - y_\xi y_{\eta\eta}\right) + \frac{1}{J^3}\left[\frac{\partial J}{\partial \eta}\left(y_\xi y_\eta + x_\xi x_\eta\right) - \frac{\partial J}{\partial \xi}\left(y_\eta^2 + x_\eta^2\right)\right]$$

$$E = \frac{1}{J^2}\left(-x_\eta x_{\xi\xi} + x_\xi x_{\xi\eta} - y_\eta y_{\xi\xi} + y_\xi y_{\xi\eta}\right) + \frac{1}{J^3}\left[\frac{\partial J}{\partial \xi}\left(y_\eta y_\xi + x_\eta x_\xi\right) - \frac{\partial J}{\partial \eta}\left(y_\xi^2 + x_\xi^2\right)\right].$$

Our alternating solution procedure is similar to that in [13] and [9]. Having obtained the approximate physical solution u^n and adaptive mesh $\boldsymbol{x}^n = (x^n, y^n)$ at time $t^n = t^{n-1} + \Delta t_n$, we first compute the monitor function $G^n = G^n(x^n, y^n, u^n)$ and then the new mesh $\boldsymbol{x}^{n+1}$ by integrating the MMPDE for a fixed time step between 0.005 and 0.01 with an implicit Euler method. This time step for solving the MMPDE is different from that in solving the physical PDE, which is described below. With some moving mesh strategies, the MMPDEs are solved to steady state (as t approaches infinity) for each time step Δt_n, where in turn Δt_n approaches 0 as t approaches the blowup time.

At a given time step, to obtain the approximate physical solution u^{n+1}, the Crank-Nicholson method is used, i.e.,

$$\left[I - \Delta t_{n+1}\frac{\Delta + f'(u(t_n))}{2}\right]u(t_{n+1}) = u(t_n) + \Delta t_{n+1}\left[f(u(t_n)) + \frac{\Delta - f'(u(t_n))}{2}u(t_n)\right],$$

where Δ is the Laplace operator in physical domain Ω_p. In solving the linear system, we choose a 20 stage GMRES method. To increase the accuracy of the numerical solutions, we also use a second order, L-stable, singly diagonal implicit Runge-Kutta (SDIRK) method (see [15]).

A central difference scheme is used for the second order derivatives and an upwind scheme for the first order derivatives. We discretize $\dot{\boldsymbol{x}}$ by $(\boldsymbol{x}^{n+1} - \boldsymbol{x}^n)/\Delta t_n$.

In solving blowup problems, we adapt the time steps by choosing $\Delta t_n = \Delta t_0/||f'(u)||_\infty$ as in [9]. We have found it is difficult to use error estimates to select the time step because the amplitude of the solution u is very large as t approaches the blowup time and $||e||_\infty$ becomes larger and larger, which as a consequence causes Δt_{n+1} to approach 0 faster than desired.

4.2. Heat Equation. We first let $f(u) = u^3$ and $\phi(x, y) = 10\sin(\pi x)\sin(\pi y)$ on $[0, 1] \times [0, 1]$ in (4.1) and use the new variational form with $\theta = 0.3$ and $q = 3$. The monitor function is defined in (2.4) with $c = 0.1$. See Fig. 4.1 - 4.3 for different views of the solution near blowup.

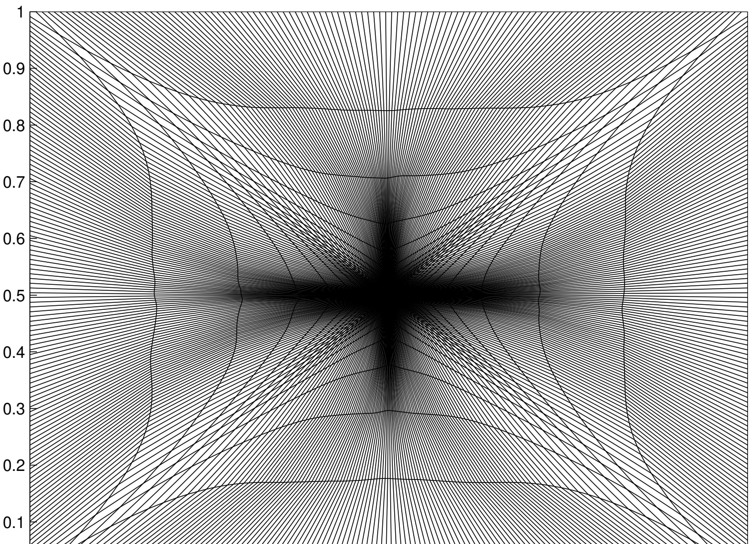
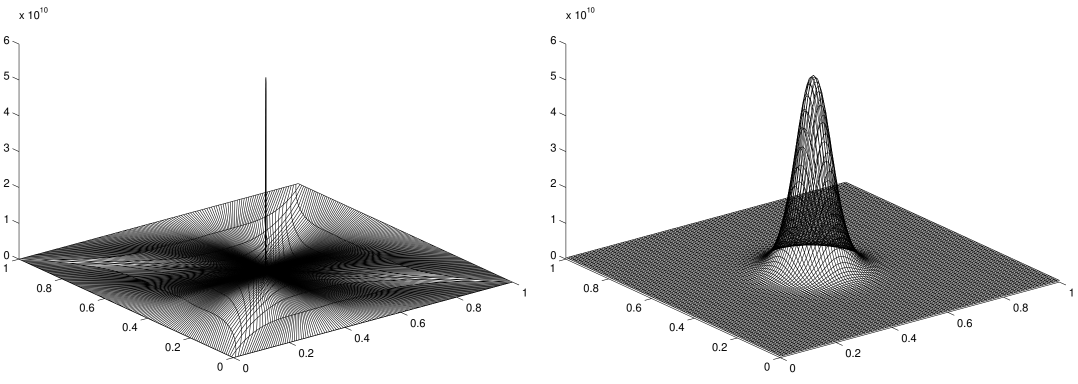


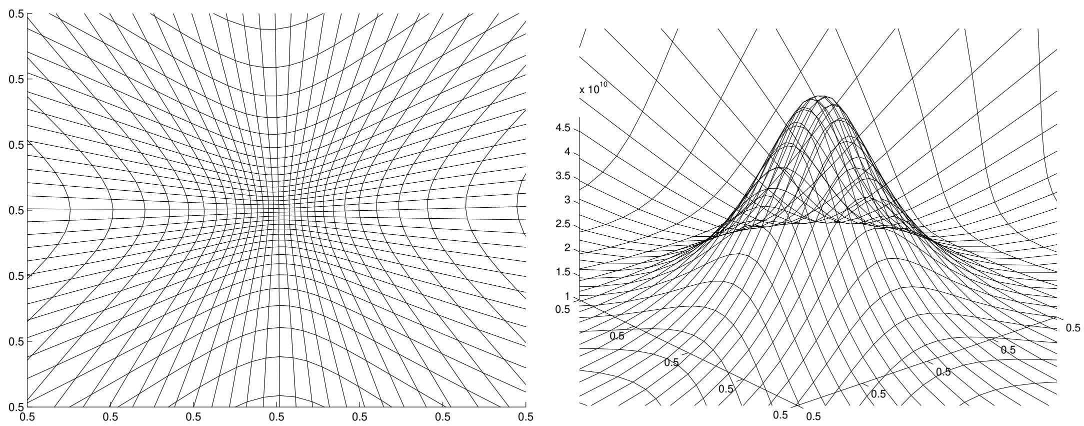
FIG. 4.1. *Mesh distributions at t=0.00629.*



(a) In physical space.

(b) In computational space.

FIG. 4.2. *Global views of the solution.*



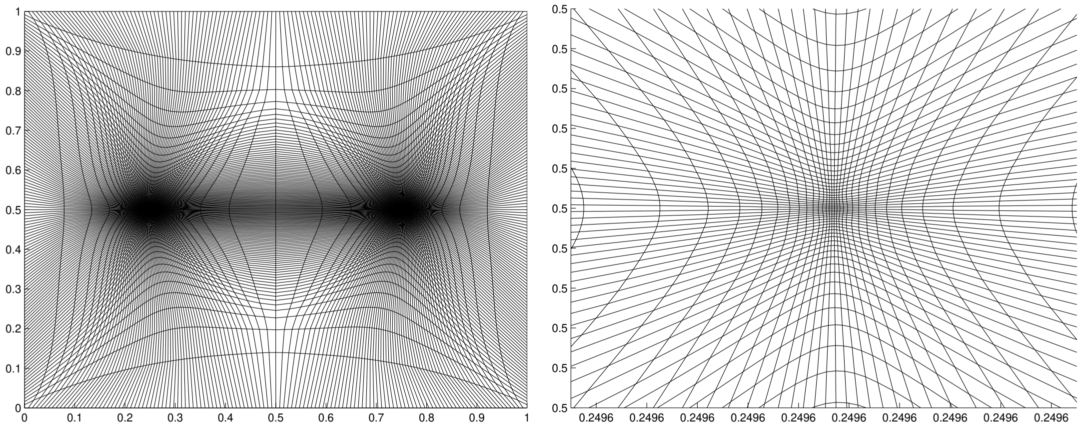
(a) The mesh

(b) The solution

FIG. 4.3. *Close-up views of the solution around the peak.*

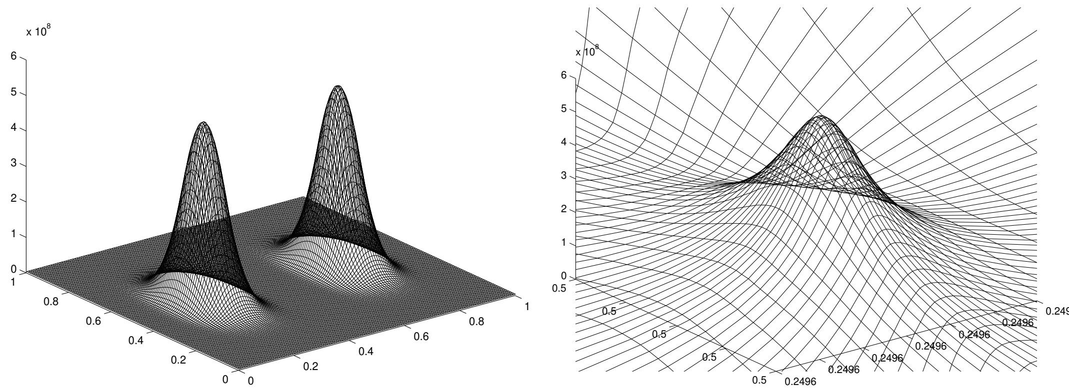
Fig. 4.2(a) shows the numerical solution at $t = 0.00629$ where $||u||_\infty = 5.3\times 10^{10}$. Despite the singular behavior, with only 120×120 points the adaptive mesh still resolves the solution near blowup, where it is still smooth as Fig. 4.2(b) and 4.3(b) show. A close-up of the mesh near the peak is given in Fig. 4.3(a), where the figure range is [0.4999993 0.5000007] $\times$ [0.499993 0.5000007] and the smallest distance between two adjacent mesh curves is 5.8×10^{-9}. Note that the mesh curves next to the boundaries in Fig. 4.1 show a little oscillation due to insufficient smoothness of the monitor function defined in (2.3) and (2.11). Increasing the smoothness of the monitor functions to remove the oscillation weakens the level of mesh concentration near the peak.

We now consider the problem discussed in [9]: $f(u) = 4\sqrt{1+u^5}$ in $[0,1] \times [0,1]$ and $\phi(x,y) = 20\sin^2(2\pi x)\sin^2(\pi y)$. Good results are obtained for the new variational form with $\theta = 0.4$ and $q = 3$ and with the monitor function defined in (2.8) with $c_1 = 0.1, c_2 = 1$ and $c = 0.1$. A 128×128 mesh is used for the computations shown in Fig. 4.4 and 4.5. In Fig. 4.4(b), the range of values is [0.249575 0.249595] $\times$ [0.49999 0.50001] and the smallest distance between two vertical curves is 2.3×10^{-7}. The blowup time is $t = 0.00249$, slightly smaller than the time $t = 0.00258$ reported in [9], and the left peak is shifted slightly from where it should be (see Fig. 5 of [9]). This shift may be caused by roundoff errors near blowup, and we intend to further investigate the relative sensitivities of the methods in a later paper.



(a) Global view. (b) Close-up view around one peak.

FIG. 4.4. *Mesh distributions.*



(a) Solutions in computational domain. (b) Close-up view around one peak.

FIG. 4.5. *Three dimensional views of the solution.*

4.3. Nonlinear Schrödinger Equation. We next solve the 2D NLS

$$i\frac{\partial u}{\partial t} + \Delta u + |u|^2 u = 0, \quad t > 0, \quad u(x,y,0) = u_0(x,y). \tag{4.6}$$

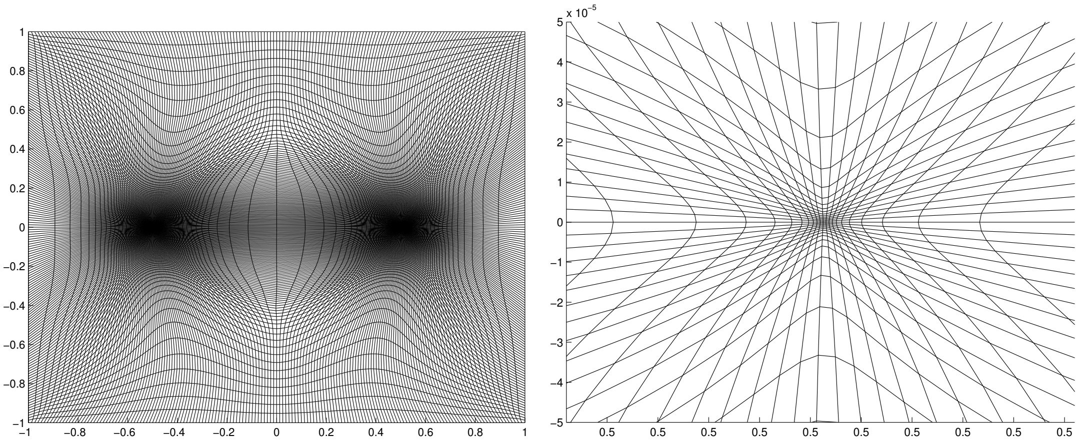
The initial condition is

$$u_0(x,y) = 20[\exp(-20((x+0.5)^2 + y^2)) + \exp(-20((x-0.5)^2 + y^2))],$$

as used by Ren and Wang [21]. Their method is not dynamically adaptive because no Lagrangian term $\dot{x}u_x + \dot{y}u_y$ is incorporated into the physical equation. Instead, iteration relies on using an interpolation process to redistribute the adaptive mesh statically (at each time step). In the radially symmetric case, the singular solutions with one blowup point are straightforward to compute using the dynamic rescaling method in [3] and [19]. For the results shown in Fig. 4.6 and 4.7 we use the new variational form with $\theta = 0.3$ and $q = 3$ and a monitor function of Winslow type, viz.,

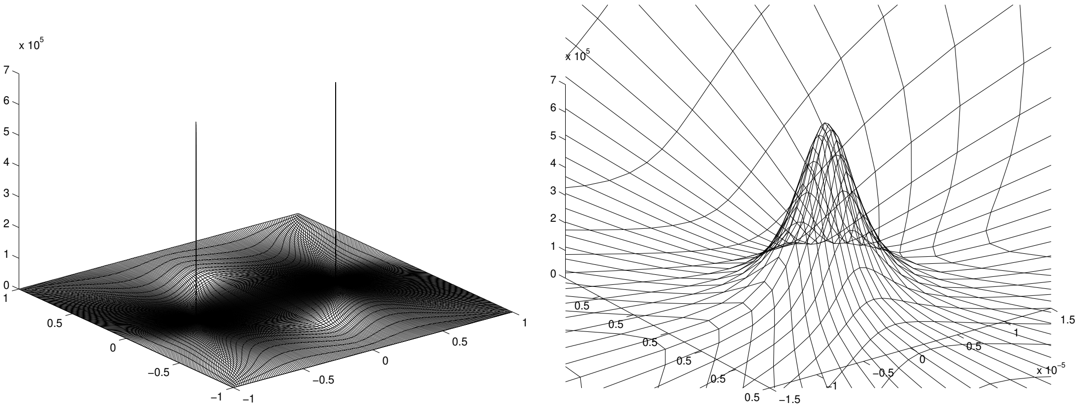
$$G = I\sqrt{1 + 0.04|\nabla|\boldsymbol{u}||^2 + |\boldsymbol{u}|^4}.$$

With a 160×160 mesh, the blowup time is $t = 0.005665$, slightly greater than $t = 0.005651$ reported in [21].



(a) Global view. (b) Close-up view around one peak.

FIG. 4.6. *Mesh distributions.*



(a) Solutions in physical domain. (b) Close-up view around one peak.

FIG. 4.7. *Three dimensional views of the solution.*

It is considerably more challenging to obtain numerical solutions of NLS with two peaks because each component of the solutions oscillates slowly at each peak while the norm of the solutions $|\boldsymbol{u}|$ increases monotonously as t approaches the blowup time. Also, it is difficult to find suitable monitor functions because $\boldsymbol{u}$ has two components, each of which oscillates from maximum negative to a maximum positive value. One of the choices is to use $|\boldsymbol{u}|$; the computation of its derivatives is complicated.

5. Conclusions. It is difficult if not meaningless to say that a particular MMPDE method employs the best adaptive strategy since adaptation issues are very problem dependent and because there is no common standard to check the qualities of a mesh adaptive strategy. Generally, though, we say that an MMPDE strategy is attractive if the mesh is computed rapidly and is aligned with a desired vector field corresponding to the solution we want to compute.

For convex physical domains we believe that the MMPDEs from the approach in [9] deserve further investigation, where adding terms like those in (3.5) may be necessary to improve the mesh qualities. While they can be chosen in such a way as to keep the properties of fast algorithms, in our experience such terms can worsen a scheme's stability. For this reason, and because of the problem of mesh folding, in our experience they have not proven to be as robust as the variational forms suggested in [14], [11], and [17] for general domains, although they may provide good preconditioners for more sophisticated

MMPDEs. For convex domains, and especially when $\Omega_p = \Omega_c$, the situation is much more interesting. As the numerical examples herein and elsewhere demonstrate, comparable numerical results are obtained with a variety of adaptive strategies. Finding a balance between fast compuation and a proper rate of mesh evolution in the blowup region depends upon an assortment of factors, including (i) how the physical PDE and the MMPDE are written (in physical or directly in computational coordinates), (ii) which fast algorithms and/or iterative algorithms with preconditioners are used, and (iii) how the mesh evolution comensurate with the level of blowup is established. These are all issues demanding further study.

REFERENCES

[1] J.U. Brackbill, *An adaptive grid with direction control*, J. Comput. Phys., 108(1993), pp. 38 - 50.

[2] C.J. Budd, J. Chen, W. Huang and R.D. Russell, *Moving mesh methods with applications to blow-up problems for PDEs*, Numerical Analysis 1995: Proc. of 1995 Biennial Conference on Numerical Analysis, Ed. by D. F. Griffiths and G. A. Watson, Pitman Research Notes in Mathematics, Longman Scientific and Technical (1996), pp. 1–17.

[3] C.J. Budd, S. Chen and R.D. Russell, *New self-similar solutions of the nonlinear Schrodinger equation with moving mesh computations*, J. Comp. Phys., 152(1999), pp. 756-789.

[4] C.J. Budd, W. Huang and R.D. Russell, *Moving mesh methods for problems with blow-up*, SIAM J. Sci. Comput., 17(1996), pp. 305–327.

[5] C. J. Budd, H. Huang and R. D. Russell, *Mesh Selection for a Nearly Singular Boundary Value Problem*, J. of Sci. Comput., 16(2001), No. 4, pp. 525–552.

[6] W. Cao, R. Carretero-Gonzalez, H. Huang and R. D. Russell, *Variational mesh adaptation methods for axisymmetrical domains*, SIAM J. Numer. Anal., to appear.

[7] W. Cao, W. Huang and R. D. Russell, *A study of monitor functions for two dimensional adaptive mesh generation*, SIAM J. Sci. Comput. 20(1999), pp. 1978-1994.

[8] W. Cao, W. Huang and R. D. Russell, *An Error Indicator Monitor Function for an r-Adaptive Finite Element Method*, J. Comp. Phys. 170(2001), pp. 871-892.

[9] H. Ceniceros and T. Hou, *An Efficient Dynamically Adaptive Mesh for Potentially Singular Solutions*, to appear.

[10] W. Huang, *Practical aspects of formulation and solution of moving mesh partial differential equations*, J. Comp. Phys., 171(2001), pp. 753-775.

[11] W. Huang, *Variational Mesh Adaptation: Isotropy and Equidistribution*, J. Comp. Phys., 174(2001), pp. 903-924.

[12] W. Huang and R.D. Russell, *Adaptive Mesh Movement – the MMPDE Approach and its Applications*, J. Comp. Appl. Math., 28 (2001), pp. 383-398.

[13] W. Huang and R.D. Russell, *Moving mesh strategy based on a gradient flow equation for two-dimensional problems*, SIAM J. Sci. Comput., 20(1999), No. 3, pp. 998-1015.

[14] W. Z. Huang and R. D. Russell, *A high dimensional moving mesh strategy*, Appl. Numer. Math., 26(1998) pp. 63-76.

[15] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations*, Vol. I and II, Springer-Verlag, Berlin, New York, 1987.

[16] Patrick M. Knupp and Nicolas Robidoux, *A framework for variational grid generation: conditioning the matrix with matrix norms*, SIAM J. Sci. Comput. 21(2000), pp. 2029-2047.

[17] Patrick Knupp and Stanly Steinberg *Fundamentals of Grid Generation*, CRC Press, 1994.

[18] R. Li, T. Tang and P. Zhang, *Moving Mesh methods in Multiple Dimensions Based on Harmonic Maps*, J. Comp. Phys., 170(2001), pp. 562-588.

[19] D.W. McLaughlin, G.C. Papanicolaou, C. Sulem and P.L. Sulem, *Focusing singularity of the cubic Schrödinger equation*, Phys. Rev. A, 34(1986), pp. 1200-1210.

[20] C.V. Pao, *Nonlinear parabolic and elliptic equations*, Plenum Press, New York, 1992.

[21] W. Ren and X. Wang, *An Iterative Grid Redistribution Method for Singular Problems in Multiple Dimensions*, J. Comp. Appl., 159(2001), pp. 246-273.

[22] C. Sulem and P.L. Sulem, *The nonlinear Schrodinger equation*, Applied Mathematical Sciences 139, Springer, 1999.

[23] V.E. Zakharov, *Handbook of Plasma Physics*, Vol. 2, Elsevier, New York, 1984.