UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

Subscale Capturing in Numerical Analysis

Stanley Osher

July 1994
CAM Report 94-24

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

Subscale Capturing in Numerical Analysis

Stanley Osher
Mathematics Department
University of California
Los Angeles, CA, USA

and

Cognitech, Inc.
Santa Monica, CA, USA

1 Introduction

In this talk we shall describe numerical methods which were devised for the purpose of computing small scale behavior without either fully resolving the whole solution or explicitly tracking certain singular parts of it. Techniques developed for this purpose include: shock-capturing, front-capturing and multiscale analysis. Areas in which these methods have recently proven useful include: image processing, computer vision, differential geometry, as well as more traditional fields of physics and engineering.

Shock capturing methods were devised for the numerical solution of nonlinear conservation laws. At the 1990 meeting of the ICM, Ami Harten [16] gave an overview of recent developments in that area, culminating in the construction of essentially nonoscillatory (ENO) schemes [17], [18]. We shall describe some of the ideas and results relating to this subject in section III.

L.I. Rudin, in his Ph.D. thesis [36] noted that the ideas and techniques from the theory of hyperbolic conservation laws and their numerical solution are relevant to the field of image processing. Images have features such as edges, lines, and textures and shock capturing is therefore an appropriate tool. Later developments [38], [39], [26], [27], [35] indicate that subscale capturing contains a great number of relevant tools for both image and video processing, as well as computer vision. We shall discuss this in section IV.

In 1987, together with J.A. Sethian [31] we devised a new numerical procedure for capturing fronts and applied it to curves and surfaces whose speeds depend on local curvature. The method uses a fixed (Eulerian) grid and finds the front as particular level set (moving with time) of a scalar function. The method applies to a very general class of problems.

The technique handles topological merging and breaking, works in any number of space dimensions, does not require that the moving surface be written as a function, captures sharp gradients and cusps in the front, and is relatively easy to program. Theoretical justification, involving the concept of viscosity solutions, has been given in [13], [7].

Many applications and extensions have recently been found. We shall describe the method and some applications in section II. We also note that the motion of multiple junctions using related ideas has been studied in [28]. A particularly novel application and extension (done with E. Harabetian) is to the numerical study of unstable fronts – e.g. vortex sheets, in [15]. This will also be described in section II. The level set formulation allows for the capturing of the front with minimal regularization because the zero level set of a continuous function can become quite complicated, even though the function itself is easy to compute.

Our last example of subscale capturing involves wavelet based algorithms for linear initial value problems. Using ideas of Beylkin, Coifman, Rokhlin, [2], we have with B. Engquist, S. Zhong, and A. Jiang [11], [19] devised very fast algorithms for evaluating the solution of linear initial value problems with time independent coefficients. This will be described in section V.

2 The Level Set Method for Capturing Moving Fronts

In a variety of physical phenomena, one wishes to follow the motion of a front whose speed is a function of the local geometry and an underlying flow field. Generally the location of the interface or front affects the flow field. Typically there have been two types of numerical algorithms employed in the solution of such problems. The first parameterizes the moving front by some variable and discretizes this parameterization into a set of marker points. The positions of these marker points are updated according to approximations of the equations of motion. For large complex motion, several problems occur. First, marker particles come together in regions where the curvature builds, causing numerical instability unless regridding is used. The regridding mechanism often dominates the real effects. Moreover the numerical methods tend to become quite stiff in these regions – see e.g. [41]. Secondly, such methods suffer from topological problems: e.g. when two regions merge or a single region splits, ad-hoc technologies are required.

Other algorithms commonly employed fall under the category of "volume of fluid" techniques which track the motion of the interior region e.g. [29], [3]. These are somewhat more adaptable to topological changes than the tracking methods but still lack the ability to easily compute geometrical quantities such as curvature of the front.

Both methods are difficult to implement in three space dimensional problems. Our idea, as first developed with J.A. Sethian in [31] is as follows. Given a region Ω in R^2 or R^3 (which could be multiply connected), and whose boundary is moving with time, we construct an auxiliary function $\varphi(\bar{x}, t)$ which is Lipschitz continuous and has the property

$$\varphi(\bar{x}, t) > 0 \quad \Leftrightarrow \quad \bar{x} \,\varepsilon\, \Omega \ \text{ at time } t \tag{2.1}$$

$$\varphi(\bar{x}, t) < 0 \quad \Leftrightarrow \quad \bar{x} \,\varepsilon\, \Omega^c \ \text{ at time } t \tag{2.2}$$

$$\varphi(\bar{x}, t) = 0 \quad \Leftrightarrow \quad \bar{x} \,\varepsilon\, \partial\Omega \ \text{ at time } t \tag{2.3}$$

On any level set of φ we have

$$\varphi_t + \vec{u} \cdot \nabla\varphi = 0 \tag{2.4}$$

where $\vec{u} = (\dot{x}(t), \dot{y}(t))$, the motion of the front and the set $\varphi \equiv 0$ characterizes $\partial\Omega$ at time t.

Generally, if the normal velocity $\vec{u} \cdot \vec{n}$ is a given function, f, of the geometry, the level set motion is governed by

$$\varphi_t + |\nabla\varphi| f = 0. \tag{2.5}$$

Typically (in 2 dimensions) f is a function of the curvature of the front, $f = f(\kappa) = f\left(\nabla \cdot \left(\frac{\nabla\varphi}{|\nabla\varphi|}\right)\right)$. In this case we can replace (2.4) by an equation involving φ only

$$\varphi_t + |\nabla\varphi| f\left(\nabla \cdot \left(\frac{\nabla\varphi}{|\nabla\varphi|}\right)\right) = 0. \tag{2.6}$$

Our algorithm is merely to extend (2.6) to be valid throughout space and just pick out the zero level set as the front at all later times. Equations of this type, for $f'(0) < 0$, have been analyzed in [13], [7] using the theory of viscosity solutions. In addition to well-posedness, it was shown that modulo a few exceptions, the level set method works. This means that the zero level set agrees with the classical motion for smooth, noninteracting curves. Moreover, the asymptotic behavior of certain fronts arising in reaction diffusion equations leads to this motion as the small parameter goes to zero [12].

In many applications involving multiphase flow in fluid dynamics the interface between any two regions can be represented by judiciously using delta functions as source terms in the equations of motion. This is true in particular for computing rising air bubbles in water, falling water drops in air, and in numerous other applications – see e.g. [46],[5]. In fact surface tension often plays a role and this quantity is just proportional to curvature, here easy to compute. Thus an Eulerian framework is easily set up, using the level set approach, allowing phenomena such as merging of water drops, resulting in surface tension driven oscillations, and drops hitting the base and deforming [46].

A key requirement here and elsewhere is that the level set function φ stay well behaved, i.e. $0 < c \leq |\nabla\varphi| \leq C$ for fixed constants (except for isolated points). In fact it would be desirable to set

$$|\nabla\varphi| = 1 \tag{2.7}$$

with the additional criteria (2.1, 2.2, 2.3). In other words, we wish to replace (at least near $\partial\Omega$) φ by d, signed distance to the boundary.

We can do this as described in [46], through reinitialization after every discrete update of the system, in a very fast way by obtaining the viscosity solution of

$$d_\tau + (|\nabla d| - 1)H(\varphi) = 0 \tag{2.8}$$

for $\tau > 0$, in fact as $\tau \uparrow \infty$, with $d(\bar{x}, 0) = \varphi(x, t)$. Here $H(\varphi)$ is any smooth monotone function of φ with $H(0) = 0$.

ENO schemes for Hamilton-Jacobi equations, as defined in [31],[32] may be used to solve this. By the method of characteristics it is clear that, near $\partial\Omega$, which is the zero level set of φ, the steady state is achieved very quickly. We thus have a fast method of computing signed distance to an arbitrary set of closed curves in R^2 or surfaces in R^3.

Another example of the use of this method in fluid dynamics involves area (or volume) preserving motion by mean curvature. This represents the simplified motion of foam and can be modelled simply by finding the zero level set of

$$\varphi_t = |\nabla\varphi| \left(\nabla \cdot \left(\frac{\nabla\varphi}{|\nabla\varphi|} \right) - \bar{\kappa} \right) \tag{2.9}$$

where $\bar{k}$ is the average curvature of the interface. This last can be easily computed

$$\bar{\kappa} = \frac{\int\int_\Omega \left(\nabla \cdot \left(\frac{\nabla\varphi}{|\nabla\varphi|} \right) \right) \delta(\varphi)|\nabla\varphi|}{\int\int_\Omega \delta(\varphi)|\nabla\varphi|}. \tag{2.10}$$

The distance reinitialization is used and the method easily yields merging and topological breaking, see [20]. More realistic models involving volume preserving acceleration by mean curvature are being developed and analyzed with the same group of people.

Another interesting example concerns Stefan problems. Earlier work was done using the level set formulation [42]. Our formulation seems to be quite simple and flexible. We solve for the temperature (in 2 or 3 dimensions)

$$T_t = \nabla \cdot k(\bar{x})\nabla T \tag{2.11}$$

$$k(\bar{x}) = k_1 \text{ if } \bar{x} \in \Omega \tag{2.12}$$

$$k(\bar{x}) = k_2 \text{ if } \bar{x} \in \Omega^c \tag{2.13}$$

$$T = 0 \text{ for } \bar{x} \in \partial\Omega \tag{2.14}$$

and the boundary of Ω moves with normal velocity

$$\vec{v} \cdot \vec{n} = \left[\frac{\partial T}{\partial n} \right] c_1 + c_2\kappa \tag{2.15}$$

where $\kappa =$ curvature of the front.

We solve this using φ, the level set function, with reinitialization, by using

$$\varphi_t + \vec{u} \cdot \nabla\varphi = 0 \tag{2.16}$$

for u defined semi-numerically as

$$\begin{aligned} \vec{u} &= c_1 \left[\Delta x \Delta_+^x \Delta_-^x T,\ \Delta y \Delta_+^y \Delta_-^y T\right] \\ &+ c_2 \left[\nabla \cdot \left(\frac{\nabla \varphi}{|\nabla \varphi|}\right)\right] \frac{\nabla \varphi}{|\nabla \varphi|} \end{aligned} \tag{2.17}$$

for Δ_+, Δ_- the usual undivided difference operators. The first term on the right is $O(\Delta x, \Delta y)$ except at the front.

We solve (2.11) by using the piecewise constant values k_1 or k_2 except when the discrete operators above cross the level set $\varphi = 0$. At such points we merely interpolate using the distance function to find the x and/or y value at which $T = 0$. We thus can get a one sided arbitrary high order approximation to $\Delta x T_{xx}$ and/or $\Delta y T_{yy}$ there. This is also used in (2.17). The results appear to be state of the art for this simple method. This is joint work with S. Chen, B. Merriman, and P. Smereka [6].

Next, with E. Harabetian [15] we consider an extension of the level set method where the normal velocity need not be intrinsic (solely geometry or position based) and for which the problem written in Lagrangian (moving) coordinate is Hadamard ill-posed. The main observation is that our approach provides an automatic regularization. There appear to be at least two reasons for this. The first is topological: a level set of a function cannot change its winding number – certain topological shapes based on the curve crossing itself are impossible. The second is analytical: the linearized problem is well-posed in the direction of propagation normal to the level set in this formulation; however it is ill-posed overall.

We shall describe the method in R^2. The three dimensional extension is relatively straightforward. Our two paradigms will be: (1) the initial value problem for the Cauchy-Riemann equations and (2) the motion of a vortex sheet in two dimensional, incompressible, inviscid fluid flow.

Our general problem is to move a curve $\Gamma_0 : (x_0(s), y_0(s))$, where s need not be arclength, through a system of partial differential equations

$$\begin{pmatrix} x_t \\ y_t \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \vec{v}(x, y, x_s, y_s) \tag{2.18}$$

with initial conditions

$$\begin{pmatrix} x(s,0) \\ y(s,0) \end{pmatrix} = \begin{pmatrix} x_0(s) \\ y_0(s) \end{pmatrix}, \quad 0 \le s \le L \tag{2.19}$$

and periodic boundary conditions

$$\begin{aligned} x(0,t) &= x(L,t) \\ y(0,t) &= y(L,t). \end{aligned} \tag{2.20}$$

Here $\Gamma_0(s)$ (which might be multiply connected) divides R^2 into an inside Ω and outside Ω^c. Also, v could depend on higher order derivatives (which it does in the curvature dependent case) or it could be nonlocal (as in the vortex sheet case).

In addition to the level set function φ, we define a conjugate function $\psi(x,y,t)$ with

$$\psi(x(0,s),y(0,s),0) \equiv s \tag{2.21}$$

and

$$\nabla\varphi\cdot(\nabla\psi)^* = \varphi_x\psi_y - \varphi_y\psi_x \neq 0 \quad \text{at } t=0 \text{ on } \Gamma_0. \tag{2.22}$$

We require an additional important condition on the conjugate function ψ

$$\psi(x(s,t),y(s,t),t) \equiv s \quad \text{for } t>0. \tag{2.23}$$

Differentiating both equation (2.3) and (2.23) leads us to two equations on $\Gamma(s,t)$

$$\varphi_t \quad + \quad \vec{v}\cdot\nabla\varphi = 0 \tag{2.24}$$
$$\psi_t \quad + \quad \vec{v}\cdot\nabla\psi = 0. \tag{2.25}$$

It remains to define x_s, y_s in terms of $\nabla\varphi$ and $\nabla\psi$ within the arguments of v in (2.24), (2.25). We do this by differentiating (2.23) and (2.3) with respect to s, which leads us to

$$\begin{pmatrix} x_s \\ y_s \end{pmatrix} = [(\nabla\varphi)\cdot(\nabla\psi)^*]^{-1}\begin{bmatrix} -\varphi_y \\ \varphi_x \end{bmatrix}. \tag{2.26}$$

We replace (x_s, y_s) by this expression in the arguments of v in (2.24),(2.25), extend this to all space, and arrive at our formulation

$$\varphi_t \quad + \quad \vec{v}\left(x, y, \frac{-\varphi_y}{(\nabla\varphi\cdot(\nabla\psi)^*)}, \ \frac{\varphi_x}{(\nabla\varphi\cdot(\nabla\psi)^*)}\right)\cdot\nabla\varphi = 0 \tag{2.27}$$
$$\psi_t \quad + \quad \vec{v}\left(x, y, \frac{-\varphi_y}{(\nabla\varphi\cdot(\nabla\psi)^*)}, \ \frac{\varphi_x}{(\nabla\varphi\cdot(\nabla\psi)^*)}\right)\cdot\nabla\psi = 0. \tag{2.28}$$

At every time step φ is reinitialized to be signed distance. We also reinitialize ψ as follows. As described in [46], we can construct ψ initially so that $\nabla\varphi\cdot\nabla\psi = 0$ on and near Γ, i.e. we generate an orthonormal coordinate system.

We reinitialize ψ to have this property by solving to steady state near Γ

$$\tilde{\psi}_t + H(\varphi)\frac{\nabla\varphi\cdot\nabla\tilde{\psi}}{|\nabla\varphi|} = 0 \tag{2.29}$$

where H is defined as in (2.8).

An interesting example is the Cauchy-Riemann system

$$\begin{aligned} x_t &= y_s \\ y_t &= -x_s. \end{aligned} \tag{2.30}$$

The level set formulation is to find the set $\varphi \equiv 0$ where

$$\varphi_t \quad + \quad \frac{|\nabla\varphi|^2}{(\nabla\varphi)\cdot(\nabla\psi)^*} = 0 \tag{2.31}$$
$$\psi_t \quad + \quad \frac{(\nabla\varphi)\cdot(\nabla\psi)}{(\nabla\varphi)\cdot(\nabla\psi)^*} = 0 \tag{2.32}$$

with the reinitialization described above. This formulation appears to stabilize the problem. Justification is given in [15].

In special cases when the velocity v is purely normal to Γ we have an alternative formulation to (2.27),(2.28). The system (2.16) can be rewritten

$$x_t = g y_s \tag{2.33}$$
$$y_t = -g x_s \tag{2.34}$$

for $g = g(x, y, x_s, y_s)$. If we set $f = \sqrt{x_s^2 + y_s^2}$, (the arclength), then Γ is moving normal to itself with velocity fg. Differentiating f with respect to t gives us a system of two equations φ and f (rather than φ and ψ) in which curvature of level sets appears in a transparent way:

$$\varphi_t + gf \cdot |\nabla \varphi| = 0 \tag{2.35}$$
$$f_t + \frac{gf}{|\nabla \varphi|} \nabla \varphi \cdot \nabla f = g \kappa f^2 \tag{2.36}$$

(for the Cauchy Riemann equations, $g \equiv 1$).

The second equation is almost a Riccati equation for the arclength f. Ill-posedness is reflected in the blow up of f or of $f \to 0$, depending on the sign of the curvature κ.

An ill-posed problem of great physical interest is the motion of a vortex sheet in the incompressible Euler equations. We have a velocity vector field $\vec{v}$ which is incompressible

$$\nabla \cdot \vec{v} = 0 \tag{2.37}$$

and which satisfies

$$\nabla \times \vec{v} = \omega \tag{2.38}$$

where the vorticity $\omega(x, y, 0)$ is a singular distribution which can be written, using the level set function φ

$$\omega(x, y, 0) = R(s) = R(x(s, 0), y(s, 0)) \tag{2.39}$$

is the strength along the initial vortex sheet $(x(s, 0), y(s, 0))$.

The vorticity moves according to the

$$\omega_t + \vec{v} \cdot \nabla \omega = 0. \tag{2.40}$$

Rather than evolving the vortex sheet by the well-known Birkhoff-Rott equation (see e.g. [23]), we shall use our new formulation

$$R(s, t) = \frac{\nabla \varphi \cdot (\nabla \psi)^*}{|\nabla \varphi|} \tag{2.41}$$

and the full problem is thus (2.24),(2.25) with

$$\vec{v} = \begin{pmatrix} -\partial_y \\ \partial_x \end{pmatrix} (-\Delta^{-1})(\nabla \varphi) \cdot (\nabla \psi)^* \cdot \delta(\varphi) \tag{2.42}$$

using reinitialization (which is crucial here).

We were able to compute the roll-up of a vortex sheet past the time of singularity as computed by Krasny in [23]. We do not do any explicit filtering in the Fourier frequencies, nor do we use blobs to smooth out the flow as in [23]. We note that the tangential velocity is discontinuous, so the level curves of ψ tend to break at the vortex sheet. This is kept manageable by the constant reinitialization of ψ.

Finally we mention that complicated motion of multiple junctions can be rather simply implemented by using as many level set functions as there are regions – see [28]. Also, in the special case of mean curvature motion, the simple heat equation together with a projection may be used [28].

3 Shock Capturing Methods

There is a vast literature on this subjection, also see [16] for a recent review article at the 1990 ICM. The fundamental problem is that the solution to the initial value problem for a system of hyperbolic conservation laws generally develops discontinuities (shocks) in finite time, no matter how smooth the initial data is. Weak solutions must be computed. The goal is to develop numerical methods which "capture" shocks automatically. Reasonable design principles are:

(1) Conservation form (defines shock capturing – see [14], [25]).

(2) No spurious overshoots, wiggles near discontinuities, yet sharp discrete shock profiles.

(3) High accuracy in smooth regions of the flow.

(4) Correct physical solution, i.e. satisfaction of the entropy conditions in the convergent limit [24].

Conventional methods had trouble with combining (1) and (3). It should be noted that wiggles can pollute the solution causing e.g. negative densities and pressures and other instabilities.

We have developed with Harten, Engquist and Chakravarthy [18], [17] and later simplified with Shu [44], [45] a class of shock capturing algorithms designed to satisfy (1-4).

These methods are called essentially nonoscillatory (ENO) schemes. They resemble their predecessors – total variation diminishing (TVD) schemes in that the stencil is adaptive, however the total variation of the solution of the approximation to a one space dimensional scalar model might increase, but only at a rate $O((\text{grid size})^p)$, for p the order of the method, up to discontinuities, and the order of accuracy can be made arbitrary in regions of smoothness. TVD schemes traditionally degenerate to first order at isolated extrema (see [40] for extensions up to second order).

The basic idea is to extend Godunov's [14] ingenious idea past first order accuracy. This was first done up to second order accuracy by van Leer [47]. A key step, and the only one we have time to describe here, is the construction of a piecewise polynomial of degree m, which interpolates discrete data w given at grid x_j. In each cell $d_j = \{(x)x_j \leq x \leq x_{j+1}\}$ we construct a polynomial of degree m which interpolates $w(x)$ at $m+1$ successive points $\{x_i\}$ including x_j and x_{j+1}.

The idea is to avoid creating oscillations by choosing the points using the "smoothest" values of w. (This is a highly nonlinear choice, as it must be). One way of doing this is to use the Newton interpolating polynomials and the associated coefficients. We start with a linear interpolant in each cell

$$q_{1,j+\frac{1}{2}} = w[x_j] + (x - x_j)w[x_j, x_{j+1}] \tag{3.1}$$

using the Newton coefficients

$$w[x_i] = w(x_i) \tag{3.2}$$
$$w[x_i, \ldots, x_{i+k}] = (x_{i+k} - x_i)^{-1}(w[x_{i+1}, \ldots, x_k] - w[x_i, \ldots, x_{k-1}]). \tag{3.3}$$

We get two candidates for $q_{2,j+\frac{1}{2}}$; which interpolate w at x_j, x_{j+1} and either x_{j-1}, or x_{j+2}

$$q_{2,j+\frac{1}{2}} = q_{1,j+\frac{1}{2}} + (x - x_j)(x - x_{j+1})[w[x_{j-1}, x_j, x_{j+1}] \text{ or } w[x_j, x_{j+1}, x_{j+2}]]. \tag{3.4}$$

Since we are trying to minimize oscillations by taking information from regions of smoothness, we pick the coefficient which is *smaller* in magnitude. We store this choice and proceed inductively up to degree m. The result is a method which is exact for piecewise polynomials of degree $\leq m$ and which is nonoscillatory (i.e. essentially monotone) across jumps. See [17] for further discussions.

Other choices are possible, in fact it seems advantageous to minimize truncation error by biasing the choice of stencil towards the center – see [43], [33].

4 Image Processing

In his 1987 Ph.D. thesis, L. Rudin [36] made the connection between various tasks in image processing and the numerical solution of nonlinear partial differential equations whose solutions develop steep gradients. Images are characterized by edges and other singularities, thus the techniques used in shock capturing are relevant here. There are now many examples of this connection. We shall discuss only a few here.

We extend the notion of "shock filter" described first in [36] to enhance images which were first blurred by a mild smoothing process. Consider the (apparently ill-posed) initial value problem

$$u_t = -|\nabla u| F[(D^2 u \cdot \nabla u, \ \nabla u)] \tag{4.1}$$
$$u(x, y, 0) = u_0(x, y) \tag{4.2}$$

where $F(A)$ is an increasing function with $F(0) = 0$.

Here, $u_0(x, y)$ is the blurry image to be processed. Intuitively if, for example, $F(A)$ is the sign function, then the process involves propagating data towards blurred out edges, (zeros of the edge detector $((D^2u)\nabla u, \nabla u))$). The apparent ill-posedness is take care of by the choice of finite difference approximation, which has the effect of turning off the motion at isolated extrema. See [30] for a further discussion of this. We note here that the resulting motion satisfies a local maximum principle and, in one space dimension, preserves the total variation of the original image.

An important extension of these ideas come in the development of a total variation based restoration algorithm [38],[39]. We are given a blurry noisy image

$$u_0(x, y) = (Au)(x, y) + \bar{n}(x, y) \tag{4.3}$$

where A is a linear integral operator and $\bar{n}$ is additive noise. Also u_0 is the observed intensity function while u is the image to be restored. The method is quite general – A needs only to be a compact operator.

We minimize the total variation

$$\text{minimize} \int_\Omega \sqrt{u_x^2 + u_y^2}\,dxdy \tag{4.4}$$

subject to constraints on u involving the mean and variance of the noise

$$\int_\Omega u\,dxdy = \int_\Omega u_0\,dxdy \tag{4.5}$$

$$\int (Au - u_0)^2 dxdy = \sigma^2. \tag{4.6}$$

We use the gradient projection method of Rosen [34], which in this case becomes the interesting "constrained" time dependent partial differential equation

$$u_t = \nabla \cdot \left(\frac{\nabla u}{|\nabla u|}\right) - \lambda A^*(Au - u_0) \tag{4.7}$$

for $t > 0$, $(x, y) \in \Omega$ with boundary conditions

$$\frac{\partial u}{\partial n} = 0 \quad \text{on } \partial\Omega \tag{4.8}$$

and $u(x, y, 0)$ given so that (4.5), (4.6) are satisfied.

The Lagrange multiplier is chosen so as to preserve (4.6), (the constraint (4.5) is automatic).

The method generalizes to: multiplicative and other types of noise, and to localized constraints (suggested and implemented by L. Rudin). Theoretical justification and results on multiplicative noise are presented in [26]. The important observation is that noisy edges can be recovered to be crisp (reminiscent of shock capturing) without smearing or oscillations. See [38],[39],[26] for successful restoration of images using this approach and [9] for applications to different inverse problems. From a geometric point of view, (4.7) represents the motion of each level set of u normal to itself with normal velocity equal to its curvature divided by the magnitude of the gradient of u. The constraint term just acts to project the motion back so that (4.6) is satisfied. We note here that Alvarez-Guichard-Lions-Morel, in a very important paper [1] demonstrate that the axioms of multiscale analysis lead inexorably to motion by mean curvature and variants, as in [31]. This sort of motion is also important in computer vision and shape recognition – see e.g. [4],[21].

The notions of subscale resolution also appear in segmentation, [22] decluttering [37], reconstruction of shapes-from-shading, [35], [27], etc.

5 Fast Wavelet Based Algorithms for Linear Initial Value Problems

This is joint work with B. Engquist and S. Zhong [11] based on results in [2]. We are interested in the fast numerical solution of a system of evolution equations

$$u_t + L(x, \partial_x)u = f(x), \quad x \in \Omega \subset R^d, \quad t > 0 \qquad (5.1)$$
$$u(x,0) = u_0(x)$$

+ boundary conditions.

Here $L(x, \frac{\partial}{\partial x})$ is a linear differential operator.

We shall take an explicit discretization

$$u_j^n = u(x_j, t_n), \quad t_n = n\Delta t \qquad (5.2)$$
$$x_j = (j_1\Delta x_1, \ldots, j_d\Delta x_d)$$
$$u^{n+1} = Au^n + F \qquad (5.3)$$
$$u^0 = u_0$$
$$u_0, F \in R^{N^d}, \quad \Delta t = \text{const } (\Delta x)^r.$$

The u^n vector contains all u_j^n at time level t_n.

The matrix A is $(N^d \times N^d)$ with the number of nonzero elements in each row or column bounded by a constant.

Each time step requires $O(N^d)$ arithmetic operations. The overall complexity for $t = 0(1)$ is $O(N^{d+r}) =$ (number of unknowns).

We proposed a general approach to speed up this calculation which works extraordinarily well for parabolic equations and is quite promising for hyperbolic equations.

We solve the discretization:

$$u^n = A^n u_0 + \sum_{\nu=0}^{n-1} A^\nu F. \qquad (5.4)$$

We compute the solution for $F = 0$ in $\log_2 n$ steps ($n = 2^m$, $m =$ integer). Repeatedly square A

$$A^2, A^4, \ldots, A^{2^m}.$$

(This is why the equation needs to have time independent coefficients).

Unfortunately, the later squarings involve almost dense matrices so the overall complexity is $O(N^{3d}\log N)$, which is worse than the straightforward approach.

Observation based on [2]: For the representation of A in a wavelet basis, all of the powers of A^r may be approximated by uniformly sparse matrices, and the algorithm using repeated squaring is advantageous.

Algorithm:

$$\begin{aligned} B &= SAS^{-1} \\ C &= I \\ C &= TRUNC(C + BC, \epsilon) \\ B &= TRUNC(BB, \epsilon) \\ u^n &= S^{-1}(BSu^0 + CSF) \\ S &= \text{fast wavelet transform} \end{aligned}$$

$$TRUNC(A, \epsilon)\begin{cases} \tilde{a}_{ij} = a_{ij} \text{ if } |a_{ij}| > \epsilon \\ \tilde{a}_{ij} = 0, \text{ if } |a_{ij}| < \epsilon. \end{cases} \tag{5.5}$$

If $\epsilon = 0$ we get the usual operator (up to similarity).

For a fixed accuracy predetermined, the computational complexity to compute a one dimensional hyperbolic equation can be reduced from $O(N^2)$ to $O(N(\log N)^3)$ with small constant.

For parabolic d-dimensional an explicit calculation with standard complexity $O(N^{d+2})$ can be reduced to $O(N^d(\log N)^3)$.

Extensions to periodic in time sources $f(x,t)$ are easy.

Together with A. Jiang [19] we have shown the following: if we wish to evaluate the solution only in the neighborhood of one point $x = x^*$ at $t = t^n$, the complexity decreases tremendously, e.g. for one dimensional parabolic equation it becomes $O(\log^4 N)$ as opposed to $O(N(\log^3 N))$.

For a general multidimensional parabolic equation, the complexity is again only $O(\log^4 N))$.

For a d dimensional hyperbolic system the complexity is $O(N^{(2d-2)}\log^3 N)$. This is advantageous for dimension $d = 1, 2$. We expect to do better using more localized basis functions of Coifman and Meyer – see [8], and using a nonlinear partial differential equation based replacement for ray tracing – see [10].

References

[1] L. Alvarez, F. Guichard, P.L. Lions, and J.M. Morel, *Axioms and Fundamental Equations of Image Processing,* Arch. Rat. Mech. and Analys., **123**, (1993), 199-258.

[2] G. Beylkin, R. Coifman, and V. Rokhlin, *Fast Wavelet Transforms and Numerical Algorithms, I,* Comm. Pure Appl. Math., **64**, (1991), 141-184.

[3] J. Brackbill, D. Kothe, and C. Zemach, *A Continuum Method for Modeling Surface Tension*, J. Comput. Phys., **100**, (1992), 335-353.

[4] V. Caselles, F. Catte, T. Coll, F. Dibos, *A Geometric Model for Active Contours in Image Processing*, Numer. Mat., **66**, (1993), 1-31.

[5] Y.C. Chang, T.Y. Hou, B. Merriman, and S. Osher, *A Level Set Formulation of Eulerian Interface Capturing Methods for Incompressible Fluid Flows*, to appear, J. Comput. Phys., (1994).

[6] S. Chen, B. Merriman, P. Smereka, and S. Osher, *A Fast Level Set Based Algorithm for Stefan Problems*, preprint, (1994).

[7] Y.G. Chen, Y. Giga, and S. Goto, *Uniqueness and Existence of Viscosity Solutions of Generalized Mean Curvature Flow Equations*, J. Diff. Geom., **23**, (1986), 749-785.

[8] R. Coifman and Y. Meyer, *Remarques sur l'analyse de Fourier á fenetre, série I*, C.R. Acad. Sci. Paris, **312**, (1991), 259-261.

[9] D. Dobson and F. Santosa, *An Image Enhancement for Electrical Impedance Tomography, Inverse Problems*, to appear, (1994).

[10] B. Engquist, E. Fatemi, and S. Osher, *Numerical Solution of the High Frequency Asymptotic Expansion for Hyperbolic Equations*, in Proceedings of the 10^{th} Annual Review of Processers in Applied Computational Eletromagnetics, Monterey, CA, (1994), A. Terzuoli, ed., Vol. 1, 32-44.

[11] B. Engquist, S. Osher, and S. Zhong, *Fast Wavelet Algorithms for Linear Evolution Equations*, SIAM J. Scientific Stat. Computing, **15**, (1994), 755-775.

[12] L.C. Evans, M. Soner, and P. Souganidis, *Phase Transitions and Generalized Motion by Mean Curvature*, Comm. Pure Appl. Math., **45**, (1992), 1097-1123.

[13] L.C. Evans and J. Spruck, *Motion of Level Sets by Mean Curvature, I*, J. Diff. Geom., **23**, (1986), 69-96.

[14] S. Godunov, *A Difference Scheme for Computation of Discontinuous Solutions of Equations of Fluid Dynamics*, Math. Sbornik, **47**, (1959), 271-306.

[15] E. Harabetian and S. Osher, *Stabilizing Ill-Posed Problems Via the Level Set Approach*, preprint, (1994).

[16] A. Harten, *Recent Developments in Shock Capturing Schemes*, Proceedings of the ICM, Kyoto 1990, (1990), 1549-1573.

[17] A. Harten, B. Engquist, S. Osher, and S.R. Chakravarthy, *Uniformly High Order Accurate Essentially Nonoscillatory Schemes, II*, J. Comp. Phys., **71**, (1987), 231-303.

[18] A. Harten and S. Osher, *Uniformly High-Order Accurate Nonoscillatory Schemes, I*, SINUM, **24**, (1987), 279-304.

[19] A. Jiang, *Fast Wavelet Algorithms for Solving Linear Equations*, Ph.D. Prospectus, UCLA Math., (1993).

[20] M. Kang, P. Smereka, B. Merriman, and S. Osher, *On Moving Interfaces by Volume Preserving Velocities or Accelerations*, preprint, (1994).

[21] R. Kimmel, N. Kiryati, and A. Bruckstein, *Sub-Pixel Distance Maps and Weighted Distance Transforms*, JMIV, to appear, (1994).

[22] G. Koepfler, C. Lopez, and J.M. Morel, *A Multiscale Algorithm for Image Segmentation by Variational Method*, SINUM, **31**, (1994), 282-299.

[23] R. Krasny, *Computing Vortex Sheet Motion*, Proceedings of the ICM, Kyoto 1990, (1990), 1573-1583

[24] P.D. Lax, *Weak Solutions of Nonlinear Hyperbolic Equations and their Numerical Computation*, Comm. Pure Appl. Math., **7**, (1954), 159-193.

[25] P.D. Lax and B. Wendroff, *Systems of Conservation Laws*, Comm. Pure Appl. Math., **13**, (1960), 217-237.

[26] P.L. Lions, S. Osher, and L. Rudin, *Denoising and Deblurring Images with Constrained Nonlinear Partial Differential Equations*, submitted to SINUM.

[27] P.L. Lions, E. Rouy, and A. Tourin, *Shape from Shading, Viscosity Solutions, and Edges*, Numer. Math., **64**, (1993), 323-354.

[28] B. Merriman, J. Bence, and S. Osher, *Motion of Multiple Junctions: A Level Set Approach*, J. Comput. Phys., **112**, (1994), 334-363.

[29] W. Noh and P. Woodward, *A Simple Line Interface Calculation*, Proceeding, Fifth Int'l. Conf. on Fluid Dynamics, A.I. van de Vooran and D. J. Zandberger, eds., Springer-Verlag, (1970).

[30] S. Osher and L.I. Rudin, *Feature-Oriented Image Enhancement Using Shock Filters*, SINUM, **27**, (1990), 919-940.

[31] S. Osher and J.A. Sethian, *Fronts Propagating with Curvature Dependent Speed, Algorithms Based on a Hamilton-Jacobi Formulation*, J. Comp. Phys., Vol 79, (1988), 12-49.

[32] S. Osher and C.-W. Shu, *High-Order Essentially Nonoscillatory Schemes for Hamilton-Jacobi Equations*, SINUM, **28**, (1991), 907-922.

[33] A. Rogerson, E. Meiburg, *A Numerical Study of Convergence Properties of ENO Schemes*, J. Scientific Computing, **5**, (1990), 151-167.

[34] J.G. Rosen, *The Gradient Projection Method for Nonlinear Programming, II, Nonlinear Constraints*, J. SIAM, **9**, (1961), 514-532.

[35] E. Rouy and A. Tourin, *A Viscosity Solution Approach to Shape from Shading*, SINUM, **27**, (1992), 867-884.

[36] L. Rudin, *Images, Numerical Analysis of Singularities, and Shock Filters*, Caltech Comp. Sc. Dept. Report # TR 5250:87, (1987).

[37] L. Rudin, G. Koepfler, F. Nordby, and J.M. Morel, *Fast Variational Algorithm for Clutter Removal Through Pyramidal Domain Decomposition*, Proceedings SPIE Conference, San Diego, CA, July, 1993.

[38] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear Total Variation Based Noise Removal Algorithms*, Physica D, **60**, (1992), 259-268.

[39] L. Rudin, S. Osher, and C. Fu, *Total Variation Based Restoration of Noisy, Blurred Images*, SINUM, to appear.

[40] R. Sanders, *A Third Order Accurate Variation Nonexpansive Difference Scheme for a Single Nonlinear Conservation*, Math. Comp., Vol. 51, (1988), 535-558.

[41] J.A. Sethian, *Curvature and the Evolution of Fronts*, Comm. Math. Phys., **101**, (1985), 487-499.

[42] J. Sethian and J. Strain, *Crystal Growth and Dendrite Solidification*, J. Comp. Phys., **98**, (1992), 231-253.

[43] C.-W. Shu, *Numerical Experiments on the Accuracy of ENO and Modified ENO Schemes*, J. Scientific Computing, **5**, (1990), 127-150.

[44] C.-W. Shu and S. Osher, *Efficient Implementation of Essentially Nonoscillatory Schemes I*, J. Comput. Phys., **77**, (1988), 439-471.

[45] C.-W. Shu and S. Osher, *Efficient Implementation of Essentially Nonoscillatory Schemes II*, J. Comput. Phys., **83**, (1989), 32-78.

[46] M. Sussman, P. Smereka, and S. Osher, *A Level Set Approach for Computing Solutions to Incompressible Two Phase Flow*, to appear, J. Comput. Phys., (1994).

[47] B. Van Leer, *Towards the Ultimate Conservative Difference Scheme V. A Second Order Sequel to Godunov's Method*, J. Comp. Phys., **32**, (1979), 101-136.