SIAM J. Sci. Comput.
Vol. 19, No. 3, pp. 728–765, May 1998

DESIGN AND APPLICATION OF A GRADIENT-WEIGHTED MOVING FINITE ELEMENT CODE I: IN ONE DIMENSION*

NEIL N. CARLSON† AND KEITH MILLER‡

Abstract. This paper reports on the design of a robust and versatile gradient-weighted moving finite element (GWMFE) code in one dimension and its application to a variety of difficult PDEs and PDE systems. A companion paper, part II, will do the same for the two-dimensional (2D) case. These moving node methods are especially suited to problems which develop sharp moving fronts, especially problems where one needs to resolve the fine-scale structure of the fronts. Brief explanations are given of the variational interpretation of GWMFE, the geometrical-mechanical interpretation, simplified regularization terms, and the treatment of PDE systems. There are many possible pitfalls in the design of GWMFE codes; section 5 discusses special features of the implicit one-dimensional (1D) and 2D codes which contribute greatly to their robustness and efficiency. Section 6 uses a few simple examples to illustrate the workings of the method, some difficulties, and reasons for the standard choices of the internodal viscosity regularization coefficient. Section 7 reports numerical trials on several more difficult PDE systems. Section 8 discusses the failure of the method on certain steady-state convection problems. Section 9 describes a simple nonlinear "Krylov subspace" accelerator for Newton's method, a routine which greatly decreases the number of Jacobian evaluations required for our stiff ODE solver.

Key words. finite elements, moving nodes, moving finite elements, partial differential equations, deforming grids, adaptive grids, drift-diffusion equations, gas dynamics, nonlinear Krylov methods

AMS subject classifications. 65M60, 65M50, 65N30, 65N50

PII. S106482759426955X

1. Introduction.

We report on the design of a robust and versatile gradient-weighted moving finite element (GWMFE) code in one dimension and on its application to a variety of difficult PDEs and PDE systems. In a companion paper, part II, we will do the same for the two-dimensional (2D) case. These moving node methods are especially suited to those many highly nonlinear systems of PDEs which develop sharp moving fronts. For such problems (or rather for the important subclass of such problems where one needs to *resolve* the fine-scale structure of the fronts to compute the correct answer) the authors believe that GWMFE shows great promise as an efficient multiple-purpose solver for one-dimensional (1D), 2D, and (potentially) three-dimensional (3D) computations.

The second author and his students and collaborators have been involved in the development of MFE-type methods since their beginning (see [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16], listed in order of development), with emphasis since 1983 on the more robust and geometrically motivated gradient-weighted version, introduced in [8]. The authors have been collaborating on the development and improvement of 1D and 2D GWMFE codes since 1985, with emphasis, of course, on the 2D work. We reported on some early trials of GWMFE in one dimension in [9] and in two dimensions in [10].

In section 2 we briefly present the geometrical-mechanical interpretation of GWMFE (as opposed to its usual variational interpretation), simplified internodal viscosi-

*Received by the editors June 10, 1994; accepted for publication (in revised form) July 17, 1996. http://www.siam.org/journals/sisc/19-3/26955.html

†Department of Mathematics, Purdue University, West Lafayette, IN 47907 (carlson@math.purdue.edu).

‡Department of Mathematics, University of California, Berkeley, CA 94720 (kmiller@math.berkeley.edu).

ties, and simplified internodal tensions and pressures, all topics discussed at greater length in [12]. Section 3 discusses our treatment for systems of PDEs. Section 4 derives a catalog of formulae for the integral inner products which occur in GWMFE, for a great variety of different linear and nonlinear first- and second-order partial differential operators. Section 5 discusses some special robustness-enhancing features of our codes. Section 6 uses a few simple 1D examples to illustrate the working of the method and some of its difficulties and their solution, with special attention to the reasons for our standard choices of the internodal viscosity coefficient. In section 7 we present numerical results on several more difficult PDE systems. Section 8 discusses the failure of GWMFE for certain steady-state convection problems. Finally, in section 9 we describe our nonlinear "Krylov subspace" accelerator for our modified Newton's method.

There are many possible pitfalls in the development of MFE or GWMFE codes. Because of the exceedingly tiny cell widths which often occur (as they should) and because of certain possible geometrical degeneracies of the mass matrix which must be regularized by appropriate "internodal viscosities," the resulting ODEs of GWMFE are extremely stiff and require implicit stiffly stable ODE solvers. Several researchers have attempted their own versions of MFE or GWMFE using existing stiff ODE packages and have had rather disappointing results. In sections 4.11, 4.12, and especially 5, we discuss certain special features of our implicit 1D and 2D codes which contribute greatly to their robustness and efficiency.

One of the major failings of the original MFE method was the need for excessive tuning in the choice of the internodal regularization coefficients. It was in part for this reason that gradient-weighting was introduced, and we have found in 1D and 2D computations since 1983 that GWMFE is far less sensitive than MFE. This excessive sensitivity of MFE has been reported by other researchers, most recently by Furzeland, Verwer, and Zegeling [17] in a comparison of MFE with two other moving node methods in one dimension. Since then that study has been extended from MFE to GWMFE by Zegeling and Blom [18], who, adapting an earlier version of our 1D code to an existing stiff ODE package, reported greater robustness of GWMFE over MFE in the choice of regularization coefficients. Nevertheless, their choice of coefficients sometimes differs by factors of hundreds from the standard choices which we have long advised. We hope that the description and analysis in the present paper of the various forces acting on the GWMFE nodes will help to make the choice of the regularization coefficients more automatic and intuitive for researchers wanting to use the method and its generalizations.

The recent thesis of Zegeling [19], which includes [17] and [18] in its seven chapters, contains significant trials on moving node methods in one and two dimensions, including MFE and GWMFE, as well as an excellent bibliography, with special attention paid to the work of the Amsterdam school of researchers. We defer to that bibliography for the many references which we do not have space to mention here.

Glasser [20] developed a general-purpose 1D GWMFE code for controlled fusion work and applied it with great success to simulation of the quite physically complicated high density z-pinch experiment. Lie and Langseth [21] have independently rediscovered the gradient-weighting improvement for MFE in one dimension. They report a vastly improved condition number for the Newton's method Jacobian in solving the implicit equations of his stiff ODE solver.

Much significant work on MFE has been done by Baines and Wathen and their students and collaborators [22, 23, 24, 25, 26, 27, 28, 29, 30, 31]. In particular, we mention the recent book by Baines [32] and its excellent bibliography.

Since our previous joint paper [10] in 1988 we have continued our collaborative research and have incorporated significant innovations and improvements into our 1D and 2D codes. Most of the numerical examples reported in these two papers had been computed by 1989 and were presented (especially the 2D results) at the 1989 SIAM Annual Meeting in San Diego in the minisymposium on moving node finite element methods organized by Wathen and Miller, and at various other colloquia around the world. However, for the present two papers we repeated these computations in June 1992 in one consistent format using the latest versions of the 1D code GWMFE1DS and the 2D code GWMFE2DS.

There is one very important recent innovation for GWMFE which is not included in these two papers. Kuprat has developed [15] an idea which promises to make GWMFE much more robust and efficient. This is a "graph-massage" routine added to the 1D and 2D implicit codes, which every 5 or 10 time steps creates and annihilates nodes as needed, based on the edge lengths and the "local flatness" of the GWMFE solution's component graphs. His numerical results, with complicated shock formations and trouble-free repeated corner-reflections for the shallow water equations in a 2D square as his basic test case, are quite impressive (see [15]). Since then Kuprat has joined with Glasser, merging his graph-massage routine with Glasser's own user-friendly general-purpose version of the implicit 2D GWMFE systems code. Their results on the Kelvin–Helmholtz instability (in collaboration with the authors) and on a model "tokamak edge problem" are extremely promising and were presented at the 1993 APS tutorial on MFE methods.

The Fortran source code for GWMFE1DS, together with input files for most of the examples of this paper, can be obtained by e-mail from carlson@math.purdue.edu.

2. Variational and mechanical interpretations, regularization.

For greater detail on the topics of this section, see [12]. Consider a single time evolving PDE

$$u_t = L(u), \tag{2.1}$$

where L is a first- or second-order differential operator on the 1D spatial interval Ω. The basic idea of 1D piecewise linear GWMFE is extremely geometrical and simple. The above PDE is the equation for the *vertical motion* u_t of the graph. We wish instead to consider the *normal motion* $\dot{n}$ of the graph, where $\dot{n} \equiv u_t/(1+u_x^2)^{1/2}$. We convert the PDE (2.1) into the normal motion form

$$\dot{n} = K(u) \tag{2.2}$$

by dividing the equation by $(1+u_x^2)^{1/2}$. Here u is now allowed to be an evolving oriented 1D *manifold* or curve immersed in two dimensions (not necessarily the graph of a function). Implicit in this geometrical treatment is the assumption that a choice of the ratio between the horizontal and vertical scales has been made and fixed.

Consider any parameterization of the evolving solution manifold by $\mathbf{u}(\tau, t) \equiv (x(\tau,t), u(\tau,t))$, where τ is a 1D parameter. The *motion* $\dot{\mathbf{u}}$ of the parameterized points on the manifold and the *normal motion* $\dot{n}$ of the manifold are then given by

$$\dot{\mathbf{u}} = (\dot{x}, \dot{u}) \qquad \text{and} \qquad \dot{n} = \dot{\mathbf{u}} \cdot \mathbf{n}, \tag{2.3}$$

where $\mathbf{n}$ is the unit normal vector to the manifold. Equation (2.2) can then be written as

$$\dot{\mathbf{u}} \cdot \mathbf{n} = K(u), \tag{2.4}$$

which is really just a concise "change of parameterization" formula, equivalent to the original equation (2.1) so long as the manifold remains the graph of a function.

The GWMFE method is a discretization of the normal motion equation (2.4). Our approximant u is allowed to be an evolving piecewise linear 1D manifold (*the GWMFE manifold*) with its 2D nodal positions $\mathbf{u}_j = (x_j, u_j)$ all treated as unknowns, and with the natural linear parameter τ on the 1D cells. The position $\mathbf{u}(\tau, t)$ and motion $\dot{\mathbf{u}}(\tau, t)$ of the parameterized points are thus given by linear interpolation from the nodal positions $\mathbf{u}_j$ and nodal motions $\dot{\mathbf{u}}_j$; that is,

$$\mathbf{u}(\tau) = \sum_j \alpha^j(\tau)\mathbf{u}_j \qquad \text{and} \qquad \dot{\mathbf{u}}(\tau) = \sum_j \alpha^j(\tau)\dot{\mathbf{u}}_j, \tag{2.5}$$

where $\alpha^j(\tau)$ is the usual jth "hat function." Hence the normal motion is given by

$$\dot{n}(\tau) = \dot{\mathbf{u}}(\tau) \cdot \mathbf{n}(\tau) = \sum_j \dot{x}_j(\alpha^j n_1) + \dot{u}_j(\alpha^j n_2). \tag{2.6}$$

2.1. The variational interpretation. The usual interpretation of GWMFE has been in terms of its *variational* interpretation. That is, we obtain ODEs for the evolution of the nodes by requiring that the *nodal motions* $\dot{\mathbf{u}}_j$ be chosen at each instant so as to

$$\text{minimize} \quad \psi \equiv \textstyle\int (\dot{\mathbf{u}} \cdot \mathbf{n} - K(u))^2 \, ds, \tag{2.7}$$

where this L^2 integral of the normal motion residual is taken with respect to the arclength ds on the GWMFE manifold. The canonical equations for this linear least squares minimization are then that the residual be orthogonal to the two basis functions $\alpha^i n_1, \alpha^i n_2$ at each ith node,

$$\begin{aligned} \frac{1}{2}\frac{\partial \psi}{\partial \dot{x}_i} &= \int [\dot{\mathbf{u}} \cdot \mathbf{n} - K(u)]\alpha^i n_1 \, ds = 0, \\ \frac{1}{2}\frac{\partial \psi}{\partial \dot{u}_i} &= \int [\dot{\mathbf{u}} \cdot \mathbf{n} - K(u)]\alpha^i n_2 \, ds = 0. \end{aligned} \tag{2.8}$$

2.2. The geometrical-mechanical interpretation. Instead, we can write the equation (2.4) for the exact solution u in a 2D vector-valued form,

$$(\dot{\mathbf{u}} \cdot \mathbf{n})\mathbf{n} = K(u)\mathbf{n}, \tag{2.9}$$

at each point on the manifold. Both sides in (2.9) should be thought of as forces per unit arclength ds on the solution manifold (both in the normal direction). The right-hand side represents the *applied forces* on the manifold from the PDE, and the left-hand side represents viscous *drag forces* (with viscosity coefficient +1) opposing the normal motion of the manifold.

The GWMFE equations (2.8) are merely a discretized approximation to the force balance equation (2.9). We let u denote our piecewise linear GWMFE manifold approximating the true solution. Then we *concentrate* the distributed forces in (2.9) onto the nodes by the laws of leverage (i.e., the principle of virtual work in mechanics). Notice that an infinitesimal perturbation $d\mathbf{u}_i$ of the ith node of the piecewise linear manifold induces the infinitesimal perturbation $d\mathbf{u}(\tau) = \alpha^i(\tau) d\mathbf{u}_i$ of the point at parameter τ on the manifold. Thus we have the *leverage factor* α^i, and the concentrated forces on the ith node are

$$\mathbf{F}_i \equiv -\int (\dot{\mathbf{u}} \cdot \mathbf{n})\mathbf{n}\, \alpha^i \, ds + \int (K(u)\mathbf{n})\, \alpha^i \, ds = \mathbf{0}, \tag{2.10}$$

which is exactly (2.8) in vector form.

Hence, equation (2.10) states that *the GWMFE manifold moves its nodes in such a way that the distributed viscous drag forces* $(\dot{\mathbf{u}}\cdot\mathbf{n})\mathbf{n}$ *and applied forces* $K(u)\mathbf{n}$*, when concentrated onto the nodes by the laws of leverage, exactly balance at each node.*

In terms of the original u_t and $L(u)$ notation of (2.1), the force balance equations (2.10) have the form

$$\mathbf{F}_i \equiv \int [-u_t + L(u)]\alpha^i \mathbf{n}\, dx \; = \mathbf{0}. \tag{2.11}$$

Letting $y(t) = (x_1, u_1; x_2, u_2; \ldots)$ denote the vector of unknown nodal parameters grouped node-by-node, we see that the GWMFE equations (2.8), or equivalently (2.10), have the form

$$C(y)\dot{y} = g(y), \tag{2.12}$$

where the "mass matrix" $C(y)$ ("viscosity matrix" would really be a better term) is the Grammian matrix of inner products of the basis functions $\alpha^i n_1, \alpha^i n_2$ and $g(y)$ is the vector of inner products of $K(u)$ with those basis functions.

2.3. Examples. A few simple examples will illustrate the normal motion form (2.4) or (2.9) of a PDE.

Example 1. Consider the equation for "motion by mean curvature plus pressure,"

$$(\dot{\mathbf{u}}\cdot\mathbf{n})\mathbf{n} = \mathcal{C}(u)\mathbf{n} + h\mathbf{n}. \tag{2.13}$$

This is already in the form (2.9). Here u is an oriented 1D manifold or curve embedded in two dimensions, $\mathcal{C}(u)$ is its curvature with respect to the "outward" normal direction $\mathbf{n}$, and h is an "internal" pressure on the manifold (which may vary with position and time). Here $\mathbf{t} \equiv (\cos\theta, \sin\theta)$ is the unit tangent vector, where θ is the "turning angle" of the curve; $\mathbf{n} = (-\sin\theta, \cos\theta)$; and $\mathcal{C}(u) = d\theta/ds$.

Integrating the $\mathcal{C}(u)\mathbf{n}$ term over a portion of the curve between two of its points "$-$" and "$+$", we get the geometrical identity

$$\int_-^+ \mathcal{C}(u)\mathbf{n}\, ds \; = \int_-^+ \frac{d\theta}{ds}\begin{pmatrix} -\sin\theta \\ \cos\theta \end{pmatrix} ds = \mathbf{t}^+ - \mathbf{t}^-. \tag{2.14}$$

The right-hand side represents the force that would be imposed on the portion by a "surface tension" of $+1$ in the curve. Hence, this identity says that a surface tension of $+1$ in the curve or manifold is equivalent to having a force per unit area of $\mathcal{C}(u)\mathbf{n}$ imposed on the manifold. (The similar geometrical identity with the same physical interpretation holds in higher space dimensions, as explained in section 2 of [12].) Thus, (2.13) is the force balance equation for an oriented 1D "soap film" moving under a surface tension of $+1$ and an internal pressure of h, with a viscosity of $+1$ resisting the normal component of its motion.

Example 2. Consider the equation

$$u_t = -a(x,u)u_x + b(x,u), \tag{2.15}$$

which represents convection with speed a plus growth with a growth rate term b. In its normal motion form (2.4) this becomes

$$\dot{\mathbf{u}}\cdot\mathbf{n} = \begin{pmatrix} a \\ b \end{pmatrix}\cdot\mathbf{n} \equiv K(u). \tag{2.16}$$

Thus the applied forces $K(u)\mathbf{n}$ on the manifold are exactly those that would be given by a "solar wind" with momentum flux of (a, b) bouncing off the manifold and imparting the normal component of its momentum to the manifold. Note that a parameterized point on the exact solution manifold *could* choose to move with velocity $\dot{\mathbf{u}} = (a, b)$. (This would be the method of characteristics, which gives a *particular* parameterization to the evolving manifold.) Quite different parameterizations are possible, however, since the parameterized points in the exact solution manifold are free to move as they like in the tangential directions.

Note that there is nothing in the normal motion form (2.16) which requires the manifold to remain the graph of a function; in fact, if our solar wind (a, b) flows harder at one value of u than at another (as in Burgers's equation, $u_t = -uu_x$, for example) the manifold will eventually fold over and $u(x)$ will become multiple-valued. This foldover happens smoothly and our GWMFE method doesn't even decrease the time step Δt as it occurs, as is illustrated in section 6.6.

Example 3. We add a second-order diffusion term νu_{xx} to the previous equation, with a tiny diffusion coefficient ν,

$$u_t = -au_x + b + \nu u_{xx}. \tag{2.17}$$

Since $u_{xx} = (1 + u_x^2)^{3/2} C(u)$ in one dimension, this equation in its normal motion form is

$$\dot{\mathbf{u}} \cdot \mathbf{n} = \begin{pmatrix} a \\ b \end{pmatrix} \cdot \mathbf{n} + \nu(1 + u_x^2)\mathcal{C}(u). \tag{2.18}$$

2.4. Mollification for second-order terms. Second-order terms, such as the curvature forces $\mathcal{C}(u)\mathbf{n}$ in Example 1 and the $\nu(1+u_x^2)\mathcal{C}(u)\mathbf{n}$ diffusion forces in Example 3, need to be interpreted in GWMFE in the sense of *mollification* or *smoothing*. That is, we imagine the corners of our piecewise linear GWMFE manifold to be ever so slightly mollified or smoothed-off. One then forms the force balance equations (2.10) or (2.11) and takes for our GWMFE equations the limiting equations as the "mollification radius δ" tends to zero.

Under mollification the curvature and diffusion forces are concentrated in tiny $O(\delta)$ neighborhoods about the nodes. However, because of the force identity (2.14) for Example 1, and the fact that $\mathbf{n} = (-u_x(1 + u_x^2)^{-1/2}), (1 + u_x^2)^{-1/2})$ for Example 3, the required integrals in (2.10) or (2.11) have consistent limits as δ tends to zero, completely independently of the particular "smoothing-off" process which is used. See section 4 for these and similar derivations.

There seems to be some confusion in the literature concerning supposedly different treatments for second-order terms in MFE. Some researchers seem to think, for example, that there is a difference between the "smoothing" or "mollification" interpretation always given in MFE papers by Miller and coworkers [1, 2, 3, 4, 5, 6, 7, 8, 9] and the "integration by parts" interpretation given by Mueller and Carey [33]. But of course the smoothing interpretation and the properly handled integration by parts interpretation (where it applies) are equivalent, because integration by parts for generalized derivatives is *based* on the idea of smoothing. But integration by parts is inadequate in some cases; see, for example, sections 4.6 and 4.8.

2.5. The role of diffusion in causing shocks. The added diffusion term νu_{xx} in Example 3 drastically changes the character of the solution. The diffusion (no matter how tiny the coefficient ν) prevents infinite gradients, abruptly blocks foldover, keeps $u(x)$ single-valued, and transmits flux throughout the near-shock. One usually thinks of diffusion as something that *prevents* shocks by spreading them into thin near-shocks. However, within the setting of *manifold solutions* of (2.16) and (2.18) it is clear that the role of the tiny diffusion coefficient is to *cause* these near-shocks, for without it the manifold solution would just smoothly fold over; see section 6.6 and compare Figures 6.3 and 6.10.

In the GWMFE method this diffusion can be seen to create a "horizontal pressure" of magnitude $\approx \nu|u_x|$ in the steep cells of a near-shock. See formula (4.9) later and see [12] for details. In a shock the cells are being squeezed together under external compressions from the nonlinear "solar wind" and it is these diffusive horizontal pressures which keep the cells from collapsing to zero width, transmit horizontal forces throughout the shock, and keep the cells there locked into motion together.

We emphasize that the nonlinear PDE (2.15) is *incomplete* if one wants single-valued solutions; even within the setting of "weak solutions," *additional* information involving "entropy" or the "zero-diffusion limit" must be invoked to avoid nonuniqueness. We also emphasize that the GWMFE method has *no* artificial diffusion in itself; it would instead smoothly follow the multiple-valued manifold solution of (2.16). Hence, if one wants single-valued solutions and thin near-shocks from GWMFE then one must *put* a tiny ν in the PDE. However, ν must not be chosen too tiny, for the resulting exceedingly thin cells and exceedingly stiff ODEs would be difficult to handle numerically.

2.6. Simplified internodal viscosities. See [12] for details on this section. Equation (2.9) is a *degenerate* system of two PDEs for the two unknown functions x and u of $\mathbf{u}(\tau, t)$; that is, there are no forces in this equation telling these parameterized points how to move or stabilize in the tangential direction. Since GWMFE is a discretization of (2.9), it is not surprising that the GWMFE equations (2.10) sometimes become indeterminate in certain tangential directions.

Notice that in the nodal force balance equation (2.10) each cell adjacent to the ith node provides viscous resistance only to the normal components of the nodal motion $\dot{\mathbf{u}}_i$ and absolutely no resistance to the tangential components of this motion. Fortunately, because of the coarseness of the discretization, the normals $\mathbf{n}$ of the two adjacent cells are usually nonparallel, and hence there is viscous resistance in the mass matrix to nodal motions $\dot{\mathbf{u}}_i$ in all directions. However, whenever the graph becomes nearly "straight" at an ith node, i.e., the two adjacent normals $\mathbf{n}$ are nearly *parallel*, there are tangential nodal motions $\dot{\mathbf{u}}_i$ which produce almost no normal motion $\dot{n}$ of the manifold and thus meet almost no viscous resistance. Hence the quadratic form of the mass matrix $C(y)$ is nearly degenerate in those directions.

In fact, we can quantify the extent of that near-degeneracy quite simply and geometrically. Consider an ith node which is nearly degenerate. Since the mass matrix forces of GWMFE are invariant under rotations of coordinates, let's simplify notation by assuming that the coordinates are rotated such that the chord joining the $i-1$ and $i+1$ node is parallel to the x-axis. Let $c = \Delta u_i = -\Delta u_{i+1}$ denote the very tiny "break" between the graph and its chord. Consider a perturbation $\delta\dot{\mathbf{u}}_i = (\delta\dot{x}_i, \delta\dot{u}_i)^T$ of the motion of the ith node alone. This perturbation then produces the perturbation $\delta\mathbf{F}_i = D_i\delta\dot{\mathbf{u}}_i$ to the viscous resistance forces acting on that node in the left-hand side of (2.10), where D_i is the 2×2 diagonal block of the mass matrix C. Computing D_i (see section 3.1 of [12]), we see

$$D_i \approx \frac{1}{3}\begin{pmatrix} c^2/\ell_i + c^2/\ell_{i+1} & 0 \\ 0 & \ell_i + \ell_{i+1} \end{pmatrix}, \tag{2.19}$$

where ℓ_i and ℓ_{i+1} are the lengths of these adjacent cells. Hence, we see that the chordal direction becomes the degenerate eigendirection for D_i as $c \to 0$ and that nodal motions of magnitude $\delta\dot{x}_i$ in that direction are met by a viscous resistance of only $(c^2/\ell)\delta\dot{x}_i$ from each cell.

Note, however, that the "break" c becomes numerically indeterminate whenever it gets down toward the size of the local predictor error tolerance η_1 of our ODE solver. We therefore choose to add in "internodal viscosity" forces from each cell to its adjacent nodes, in the tangential direction and of magnitude $(A^2/\ell)\dot{\ell}$, where A is chosen several times larger than η_1. Thus the c^2/ℓ_i on the diagonal of D_i is replaced by $(A^2+c^2)/\ell_i$; this ensures that the regularization takes over to keep the mass matrix positive definite shortly before it would become numerically degenerate (through η_1-sized indeterminacies of the nodal positions). This regularization is always present, but its effect is inconsequential when the break c is considerably larger than A.

The forces $\mathbf{F}_i$ on the ith node in the force balance equation (2.10) therefore have added in the extra "internodal viscosity" regularization forces

$$\mathbf{F}_i^{reg} \equiv \frac{A^2}{\ell_i}[\dot{\mathbf{u}}_{i-1} - \dot{\mathbf{u}}_i]_T + \frac{A^2}{\ell_{i+1}}[\dot{\mathbf{u}}_{i+1} - \dot{\mathbf{u}}_i]_T, \tag{2.20}$$

where the $[\,\cdot\,]_T$ notation denotes the tangential projection of that vector on each cell. Each cell has thereby been turned into a "dashpot" with viscosity A^2/ℓ.

In 1989 we realized that these internodal viscosity forces can be greatly simplified. The 1D version of our simplified viscosities merely ignores the tangential projection in (2.20). We use instead

$$\mathbf{F}_i^{reg} \equiv \frac{A^2}{\ell_i}[\dot{\mathbf{u}}_{i-1} - \dot{\mathbf{u}}_i] + \frac{A^2}{\ell_{i+1}}[\dot{\mathbf{u}}_{i+1} - \dot{\mathbf{u}}_i]. \tag{2.21}$$

Our *standard choices* for η_1 and A^2 are discussed more fully in section 6.1.

2.7. Internodal tensions and pressures. For problems which go to long times and near-equilibrium the *dynamic* regularization terms above (whose effects are proportional to $1/\Delta t$) are sometimes not sufficient to stabilize against numerical long-term nodal drift in nearly straight portions of the GWMFE manifold. The PDE forces from the nondynamic $K(u)\mathbf{n}$ terms in the force balance equation (2.10) are all in the normal directions; thus, in a nearly degenerate portion of the graph there is no balance of opposing forces in the tangential directions to tell the nodes how to space themselves tangentially over the long term. In those cases we may choose to supplement the regularization by adding tiny tangential "internodal spring forces" in the cells.

In years past (see [8]) we have advocated certain standard inverse-quadratic compressive spring forces (i.e., outward *pressures* of magnitude C^2/ℓ_i^2 in each ith cell). In recent years, however, we have realized that for many problems internodal *tensions* are better than compressions and we recommend simple *linear tensions* instead (i.e., inward *tensions* of magnitude $B^2\ell_i$ in each ith cell). The size of these pressure or tension coefficients C^2 or B^2 will depend on the problem at hand.

We emphasize, however, that *for most problems these tensions and pressures will not be used at all.* It is best not to add tunable coefficients needlessly to a method. In fact, B^2 and C^2 are set to zero for *all* the examples of this paper except those in section 8.

3. GWMFE for systems of PDEs. Consider, for example, the *system* of PDEs

$$u_t = L_1(u,v) \quad \text{and} \quad v_t = L_2(u,v), \tag{3.1}$$

for the two unknown functions $u(x,t)$ and $v(x,t)$ on the 1D interval Ω. Here, as in (2.1), the L_1 and L_2 are first- or second-order nonlinear differential operators.

Our GWMFE approximants u, v will be piecewise linear functions with commonly *shared* moving nodes. There are many reasons for preferring not to use a different grid for each of the two functions u and v, as is discussed in detail in section 4 of [12].

With a shared moving grid we therefore have three unknowns (x_i, u_i, v_i) at each ith node. For the present code we treat the u and v graphs as two *independent* 1D manifolds $(x, u(x), 0)$ and $(x, 0, v(x))$ embedded in three dimensions. We therefore gradient-weight the two equations in (3.1) independently, dividing the first by $(1+u_x^2)^{1/2}$ and the second by $(1+v_x^2)^{1/2}$ to get the two equations in their normal motion forms, as in (2.4). The functional for minimization is then a weighted sum of the above functionals of form (2.7) over the two component graphs, i.e.,

$$\psi = w_1\psi_1 + w_2\psi_2. \tag{3.2}$$

We minimize this with respect to the choice of the nodal motions $(\dot{x}_i, \dot{u}_i, \dot{v}_i)$ by setting the derivatives $\partial\psi/\partial\dot{x}_i$, etc., equal to zero, as in (2.8). This gives us the x, u, v components of the force balance equations on the ith node,

$$\mathbf{F}_i = w_1\mathbf{F}_{1,i} + w_2\mathbf{F}_{2,i} = w_1\begin{pmatrix} \cdot \\ \cdot \\ 0 \end{pmatrix} + w_2\begin{pmatrix} \cdot \\ 0 \\ \cdot \end{pmatrix} = \mathbf{0}, \tag{3.3}$$

where the $\mathbf{F}_{1,i}$ and the $\mathbf{F}_{2,i}$ are the concentrated 3D forces, à la (2.10), on the ith nodes of the u and v graphs, respectively. Included in the $\mathbf{F}_{1,i}$ and $\mathbf{F}_{2,i}$ forces are the *regularization* forces on the u and v graphs discussed in section 2.

Our choice of the constant *PDE weights* is often $w_1 = w_2 = 1$. In many situations, however, we must assign different relative weights because of the different natures and scalings of the two equations. Thus, these weights often must be chosen with a bit of judgment and experimentation for the situation at hand.

An alternative *second GWMFE formulation for systems*, which is *not* used in the present code, would treat the unknowns as a single 1D manifold $(x, u(x), v(x))$ embedded in three dimensions, as discussed in [12].

4. A catalog of inner products for GWMFE.

We need to be able to compute the concentrated force integrals

$$\int K(u)\alpha^i\mathbf{n}\,ds \quad \text{or} \quad \int L(u)\alpha^i\mathbf{n}\,dx \tag{4.1}$$

which appear in (2.10) or (2.11). These are the inner products of $L(u)$ with the basis functions $\alpha^i n_1$ and $\alpha^i n_2$. We now develop a list of the inner products for certain standard forms of the differential operator $L(u)$.

These inner products will be computed and accumulated cell-by-cell (with a few exceptions). On the ith cell (between the nodes $i-1$ and i) we have $\Delta x_i \equiv x_i - x_{i-1}$, $\Delta u_i \equiv u_i - u_{i-1}$, the length $\ell_i \equiv \Delta s_i \equiv ((\Delta x_i)^2 + (\Delta u_i)^2)^{1/2}$, the constant slope $m_i \equiv \Delta u_i/\Delta x_i$, the constant unit normal vector

$$\mathbf{n}_i = \begin{pmatrix} n_1 \\ n_2 \end{pmatrix}_i = \begin{pmatrix} -u_x/(1+u_x^2)^{1/2} \\ 1/(1+u_x^2)^{1/2} \end{pmatrix} = \begin{pmatrix} -\Delta u_i/(\Delta x_i^2 + \Delta u_i^2)^{1/2} \\ \Delta x_i/(\Delta x_i^2 + \Delta u_i^2)^{1/2} \end{pmatrix}, \tag{4.2}$$

and the constant unit tangent vector $\mathbf{t}_i = (n_2, -n_1)$.

In the case of systems we call the unknown functions $u, v, \ldots$. It suffices to describe the inner products for the first equation, which is of the form $u_t = L_1(u, v, \ldots)$. The other equations $v_t = L_2(u, v, \ldots)$, etc., will be similar.

The integrals in (4.1) will be over both cell_i and cell_{i+1}. If f is a function of $(x, u(x), v(x))$ we denote by f_i its value at the ith node and by $[f]_i$ its *mean value* over the cell_i. That is,

$$f_i \equiv f(x_i, u_i, v_i) \qquad \text{and} \qquad [f]_i \equiv \frac{1}{\Delta x_i} \int_{\text{cell}_i} f(x, u(x), v(x))\, dx. \tag{4.3}$$

We usually approximate this integral by Simpson's rule. When higher accuracy is desired we instead use Boole's rule or a Gaussian quadrature.

4.1. Convection-growth equation. Consider the equation $u_t = -au_x + b$, where a, b are functions of x, u, v. This leads to the inner product

$$\int_{\text{cell}_i} (-au_x + b)\alpha^i \mathbf{n}\, dx = (-[a\alpha^i]_i \Delta u_i + [b\alpha^i]_i \Delta x_i)\mathbf{n}_i. \tag{4.4}$$

4.2. Conservation laws. Consider the equation $u_t = -f_x$ where the flux function f is a function of (x, u, v). Integrating by parts,

$$\begin{aligned} \int_{\text{cell}_i} -f_x \alpha^i \mathbf{n}\, dx &= \left(\int_{\text{cell}_i} -f_x \alpha^i\, dx \right) \mathbf{n}_i = \left(\int_{\text{cell}_i} f \alpha^i_x\, dx - f_i \right) \mathbf{n}_i \\ &= ([f]_i - f_i)\, \mathbf{n}_i. \end{aligned} \tag{4.5}$$

Here we have used the fact that α^i_x is constant, with integral 1; hence the integral of f times α^i_x above picks out the mean value of f on cell_i.

4.3. Conservation laws in differentiated form. Consider

$$u_t = -(f(u, v))_x = -f_u u_x - f_v v_x \equiv -au_x - bv_x. \tag{4.6}$$

Sometimes it is best to differentiate these terms out like this, rather than leaving them in divergence form. This leads to the integral

$$\int_{\text{cell}_i} (-au_x - bv_x)\alpha^i \mathbf{n}\, dx = (-[a\alpha^i]_i \Delta u_i - [b\alpha^i]_i \Delta v_i)\mathbf{n}_i. \tag{4.7}$$

Notice that the final formula in (4.5) corresponds to a type of numerical differencing and could lead to roundoff error due to cancellation; formula (4.7), however, does not.

We now come to second-order terms such as u_{xx} and v_{xx}. Under mollification the u_{xx} has its support in very tiny neighborhoods of the nodes. Since α^i is ≈ 1 in nbd_i, the neighborhood of the ith node, and ≈ 0 in all the others, we need worry only about the integral in nbd_i. In the following let m_i, m_{i+1} and $\widetilde{m}_i, \widetilde{m}_{i+1}$ denote the constant values of u_x and of v_x in cell_i and cell_{i+1}. Let $m(x) = u_x(x)$ and $\widetilde{m}(x) \equiv v_x(x)$ denote the variable values of these slopes as they make their rapid transition within nbd_i.

4.4. Simple constant coefficient diffusion. Consider the equation $u_t = u_{xx}$, which gives the integral

$$\int u_{xx}\alpha^i \mathbf{n} dx \to \int_{\mathrm{nbd}_i} u_{xx} \begin{pmatrix} -u_x w \\ 1w \end{pmatrix} dx = \int_{\mathrm{nbd}_i} \begin{pmatrix} -m_x m(1+m^2)^{-1/2} \\ m_x(1+m^2)^{-1/2} \end{pmatrix} dx$$

$$= \int_{m_i}^{m_{i+1}} \begin{pmatrix} -m(1+m^2)^{-1/2} \\ (1+m^2)^{-1/2} \end{pmatrix} dm \tag{4.8}$$

$$= \begin{pmatrix} (1+m_i^2)^{1/2} - 1 \\ -\log((1+m_i^2)^{1/2} + m_i) \end{pmatrix} - \begin{pmatrix} (1+m_{i+1}^2)^{1/2} - 1 \\ -\log((1+m_{i+1}^2)^{1/2} + m_{i+1}) \end{pmatrix}.$$

Thus each of the two adjacent cells pushes the ith node with an outward force $\mathbf{F}(u_x)$, which for *large* $|u_x|$ is

$$\mathbf{F} \approx \begin{pmatrix} |u_x| \\ -\mathrm{sgn}(u_x)\log(2|u_x|) \end{pmatrix}. \tag{4.9}$$

This force represents an outward pressure in its horizontal component and an inward tension in its vertical component. For *small* $|u_x|$ this force is

$$\mathbf{F} \approx \begin{pmatrix} -(1-\frac{1}{2}u_x^2) \\ -u_x \end{pmatrix} \approx -\mathbf{t}, \tag{4.10}$$

where $\mathbf{t}$ is the unit tangent vector, which represents a magnitude-1 *surface tension* by each cell on the node, as we would expect since then $u_{xx} \approx \mathcal{C}(u)$.

The formulae of (4.8) are exceedingly susceptible to roundoff error cancellation if either m_i is small or m_i is large and negative. In sections 4.11 and 4.12 we give versions of these formulae with roundoff error control.

4.5. Laplacian with a variable coefficient. Consider the equation $u_t = au_{xx}$, where $a = a(x, u(x), v(x))$. Since a is essentially the constant a_i near the ith node where u_{xx} has its tiny support, we have the previous case,

$$\int (au_{xx})\alpha^i \mathbf{n} dx \to a_i \int_{\mathrm{nbd}_i} u_{xx}\mathbf{n}\, dx = a_i \times [\text{formula (4.8)}]. \tag{4.11}$$

4.6. Laplacian of another variable. Consider the equation $u_t = av_{xx}$. Now, for the first time, we have to make an *additional assumption* on our smoothing process. We assume that the *same* smoothing process is used on the v graph as on the u graph. That is, we assume that, in the exceedingly tiny neighborhood of x_i where u_x, v_x make their transition from the constant values m_i, $\widetilde{m}_i$ to the constant values m_{i+1}, $\widetilde{m}_{i+1}$, we have $u_{xx} = \rho_1(x)(m_{i+1} - m_i)$ and $v_{xx} = \rho_2(x)(\widetilde{m}_{i+1} - \widetilde{m}_i)$ where the smoothing functions ρ_1, ρ_2 have support in this tiny neighborhood with total integrals of 1, and where ρ_1 and ρ_2 are the *same* function. Thus, in this neighborhood we have

$$v_{xx} = \left(\frac{\widetilde{m}_{i+1} - \widetilde{m}_i}{m_{i+1} - m_i}\right) u_{xx}, \tag{4.12}$$

which reduces the integral to a multiple of the previous formula (4.8)

$$\begin{aligned}\int (av_{xx})\alpha^i \mathbf{n}\,dx &\to a_i \left(\frac{\widetilde{m}_{i+1}-\widetilde{m}_i}{m_{i+1}-m_i}\right)\int_{\mathrm{nbd}_i} u_{xx}\mathbf{n}\,dx \\ &= a_i \left(\frac{\widetilde{m}_{i+1}-\widetilde{m}_i}{m_{i+1}-m_i}\right) \times [\text{formula (4.8)}].\end{aligned} \tag{4.13}$$

This formula makes it obvious that we want to rewrite formula (4.8) in a form where the term $m_{i+1}-m_i$ can be factored out; this is done in section 4.12.

Note that for the first time, in (4.13), we have an integral which cannot be computed cell-by-cell. We must know *simultaneously* the slopes u_x, v_x from both cell_i and cell_{i+1}. Such terms, which require information from two cells simultaneously in order to compute the terms at the node, we call *two-cell quantities*.

4.7. Semilinear diffusion in conservation form. Consider $u_t = (au_x)_x$, where $a = a(x,u,v)$. Now we divide the region of integration into cell_i plus nbd_i plus cell_{i+1}, with the understanding that the "cells" here do not include the tiny neighborhoods of the nodes where $u_x, v_x, \ldots$, make their transitions. Thus

$$\begin{aligned}\int (au_x)_x\alpha^i\mathbf{n}\,dx &= \int_{\mathrm{cell}_i} + \int_{\mathrm{nbd}_i} + \int_{\mathrm{cell}_{i+1}} \\ &= m_i\int_{\mathrm{cell}_i} a_x u_x \alpha^i \mathbf{n}\,dx + \int_{\mathrm{nbd}_i} a u_{xx}\alpha^i\mathbf{n}\,dx \\ &\quad + \int_{\mathrm{nbd}_i} a_x u_x \alpha^i\mathbf{n}\,dx + m_{i+1}\int_{\mathrm{cell}_{i+1}} a_x u_x\alpha^i\mathbf{n}\,dx \\ &\to m_i[\text{formula (4.5)}] + [\text{formula (4.11)}] + 0 + m_{i+1}[\text{formula (4.5)}] \\ &= m_i(a_i - [a]_i)\mathbf{n}_i + a_i[\text{formula (4.8)}] + m_{i+1}([a]_{i+1} - a_i)\mathbf{n}_{i+1}.\end{aligned} \tag{4.14}$$

Here we have used the fact that u_x is constant on each of the cells and that $a_x u_x$ is bounded on the tiny nbd_i.

4.8. Semilinear diffusion of another variable. Consider $u_t = (bv_x)_x$, where $b = b(x,u,v)$. Thus we have the integral

$$\begin{aligned}\int (bv_x)_x\alpha^i\mathbf{n}\,dx &= \int_{\mathrm{cell}_i} + \int_{\mathrm{nbd}_i} + \int_{\mathrm{cell}_{i+1}} \\ &= \widetilde{m}_i\int_{\mathrm{cell}_i} b_x v_x \alpha^i \mathbf{n}\,dx + \int_{\mathrm{nbd}_i} b v_{xx}\alpha^i\mathbf{n}\,dx \\ &\quad + \int_{\mathrm{nbd}_i} b_x v_x \alpha^i\mathbf{n}\,dx + \widetilde{m}_{i+1}\int_{\mathrm{cell}_{i+1}} b_x v_x\alpha^i\mathbf{n}\,dx \\ &\to \widetilde{m}_i[\text{formula (4.5)}] + [\text{formula (4.13)}] + 0 + \widetilde{m}_{i+1}[\text{formula (4.5)}] \\ &= \widetilde{m}_i(b_i - [b]_i)\mathbf{n}_i + b_i\left(\frac{\widetilde{m}_{i+1}-\widetilde{m}_i}{m_{i+1}-m_i}\right)[\text{formula (4.8)}] + \widetilde{m}_{i+1}([b]_{i+1} - b_i)\mathbf{n}_{i+1}.\end{aligned} \tag{4.15}$$

4.9. Other forms of diffusion. Consider the forms

$$u_t = (\phi(u))_{xx} = (\phi'(u)u_x)_x, \tag{4.16}$$

which is again of the form $(au_x)_x$, and

$$u_t = (\phi(u,v))_{xx} = (\phi_u u_x)_x + (\phi_v v_x)_x, \tag{4.17}$$

which is of the form $(au_x)_x + (bv_x)_x$. These were handled in sections 4.7 and 4.8 above.

Alternatively, the semilinear diffusion of section 4.7, where $a = a(u,v)$ and the partials $b \equiv a_u$ and $c \equiv a_v$ are available, can be written in the form

$$u_t = (au_x)_x = au_{xx} + bu_x u_x + cv_x u_x. \tag{4.18}$$

The first term is handled in section 4.4, and the $u_x u_x$ and $v_x u_x$ in the second and third terms are bounded and are constant on each cell, and hence give integrals like that in (4.4).

Similarly, if $b = b(u,v)$ and the partials $c \equiv b_u$ and $d \equiv b_v$ are available, the diffusion form in section 4.8 could instead be written as

$$u_t = (bv_x)_x = bv_{xx} + cu_x v_x + dv_x v_x. \tag{4.19}$$

This brings us back to the forms handled in sections 4.6 and 4.1.

4.10. Curvature, surface tension, and pressure. We return to the "1D soap film" problem of Example 1 in section 2,

$$(\dot{\mathbf{u}} \cdot \mathbf{n})\mathbf{n} = \mathcal{C}(u)\mathbf{n} + h\mathbf{n}, \tag{4.20}$$

where $\mathcal{C}(u) \equiv d\theta/ds$ denotes the curvature of the 1D manifold u and where the "pressure" h is a function of the position (x,u). Because of the identity (2.14) we have for GWMFE the integral

$$\int \mathcal{C}(u)\alpha^i \mathbf{n}\, ds = \int_{\mathrm{nbd}_i} \mathcal{C}(u)\mathbf{n}\, ds = \mathbf{t}_{i+1} - \mathbf{t}_i, \tag{4.21}$$

where these are the unit tangent vectors. The resultant of these curvature normal forces on the ith node, therefore, is in the bisector direction and is equivalent to that which would be imposed by a unit surface tension in each of the adjacent cells.

4.11. Roundoff control for (4.8), "cell-by-cell" version. We pointed out that the two formulae for cell$_i$ in (4.8) are extremely susceptible to roundoff error cancellation. Instead we code the first formula as

$$(1 + m_i^2)^{1/2} - 1 = m_i^2/(1 + (1 + m_i^2)^{1/2}), \tag{4.22}$$

which takes care of the severe problems for small m_i. The second formula we code as a truncated Taylor series

$$\log((1 + m_i^2)^{1/2} + m_i) = \frac{1}{2}\log\left(\frac{1+\eta}{1-\eta}\right) \approx \eta + \frac{1}{3}\eta^3 + \frac{1}{5}\eta^5 + \frac{1}{7}\eta^7, \tag{4.23}$$

provided $\eta \equiv m_i/(1 + m_i^2)^{1/2}$ is less than 10^{-2} in magnitude (giving about 10^{-17} relative accuracy), and otherwise as

$$\log((1 + m_i^2)^{1/2} + m_i) = \operatorname{sgn}(m_i)\log((1 + m_i^2)^{1/2} + |m_i|), \tag{4.24}$$

which takes care of the problems for large negative m_i.

4.12. Roundoff control for (4.8) and (4.13), "two-cell" version. Suppose that we instead compute the integrals of (4.8) as "two-cell quantities." Then we have further opportunity for roundoff control. Moreover, we can then factor out a term of $(m_{i+1} - m_i)$ when $m_{i+1} \approx m_i$, as is seen to be needed in formula (4.13). We write the formulae in terms of the parameter

$$e \equiv (m_{i+1} - m_i)/((1 + m_i^2)^{1/2} + (1 + m_{i+1}^2)^{1/2}). \tag{4.25}$$

Then (4.8) becomes

$$\int u_{xx} \alpha^i \mathbf{n}\, dx = \begin{pmatrix} e(m_{i+1} + m_{i-1}) \\ \log((1+e)/(1-e)) \end{pmatrix}. \tag{4.26}$$

The log term here has the possibility of significant loss of relative accuracy when e is small. Thus, when $|e| < 10^{-2}$ we use the Taylor expansion of (4.23), from which we can factor a term of $(m_{i+1} - m_i)$ as needed in (4.13) and (4.15).

We emphasize that experience shows that this roundoff control of sections 4.11 and 4.12 is absolutely essential.

5. Special features of our code.

There are some special features of our code which allow us to handle the extremely tiny Δx's and exceedingly stiff ODE systems which result from the difficult nonlinear PDEs to which we apply GWMFE.

5.1. Use of an implicit stiff ODE solver. We are presently using our own implementation of a variable time step BDF2 method (backward differentiation formula of second order). Consider the variable time steps $\Delta t \equiv h \equiv t_{n+1} - t_n$, $h_1 \equiv t_n - t_{n-1}$. Then we have the BDF2 formula, for all sufficiently smooth functions $y(t)$,

$$y_{n+1} = a y_n + b y_{n-1} + c h y'_{n+1} + d h^3 y'''(\xi), \tag{5.1}$$

where the coefficients a, b, ch, dh^3 (all functions of h, h_1) can be determined by the method of undetermined coefficients. (In the case of equal time steps $h = h_1$ they are $a = 4/3$, $b = -1/3$, $c = 2/3$, and $d = -2/9$.) Thus (for this BDF2 method and for most other successful stiff ODE solvers) we need to do a succession of extrapolations and basic BCE (backward Cauchy–Euler) steps to solve our ODE system (2.12). That is, we first do a linear extrapolation to get

$$\tilde{y} \equiv a y_n + b y_{n-1}, \tag{5.2}$$

then solve the implicit BCE equations

$$R(y) \equiv C(y) \left(\frac{y - \tilde{y}}{\Delta \tau} \right) - g(y) = 0 \tag{5.3}$$

for the unknown $y \equiv y_{n+1}$, with time step $\Delta \tau = ch$. We solve this with a (greatly modified) Newton-type method.

The initial guess for our Newton's method is given by second-order extrapolation from the previous values y_{n-2}, y_{n-1}, y_n, to give the predicted value $y^\star_{n+1}$. We use the computed *predictor error* $(y_{n+1} - y^\star_{n+1})$ as our measure of local error, which we keep smaller than a prescribed tolerance in our norm of choice. This is justified for two reasons: first, both the BDF2 formula (5.1) and the second-order extrapolation formula have local error terms which are proportional to $y'''(\xi) h^3$, and second, this strategy ensures that the initial guess $y^\star_{n+1}$ for our Newton's method is always extremely close to the converged value $y \equiv y_{n+1}$ (otherwise we would decrease the Δt).

5.2. Cell-by-cell accumulation of the residual and Jacobian. Except for some rarely encountered exceptions involving v_{xx} terms in the u_t equation, we saw that all the integrals in section 4 split into "cell quantities," i.e., separate contributions from cell_i and from cell_{i+1} to the ith node residuals. Thus we run through our computations for the *residual function* $R(y)$ (and for its numerical Jacobian matrix $R'(y)$ when needed) cell-by-cell.

We have

$$R(y) \equiv C(y)z - g(y) \equiv F(y, z) \text{ with } z \equiv (y - \tilde{y})/\Delta\tau. \tag{5.4}$$

Hence,

$$R'(y) = C(y)\frac{1}{\Delta\tau} + \frac{\partial F}{\partial y}. \tag{5.5}$$

We compute the $C(y)$ directly (to minimize roundoff) and the $\partial F/\partial y$ by numerical differentiation, using forward differences in each variable with a fixed tiny increment.

The modified Newton's method would involve solving the linearized equations

$$R(\bar{y} + \delta y) \approx R(\bar{y}) + R'(\bar{\bar{y}})\delta y = 0, \tag{5.6}$$

where $\bar{y}$ is our latest Newton iterate, δy is the correction which we desire to make, and $\bar{\bar{y}}$ is the value where the Jacobian was last evaluated and saved.

5.3. Diagonal preconditioning of our BCE equations. The residual equations (5.3) may be extremely ill conditioned and also extremely ill suited for the linear approximation assumption (5.6) upon which Newton's method is based. Recall the discussion surrounding (2.19) in section 2. Near a geometrical degeneracy (where the "break" c between the graph and its chord becomes very tiny) we saw that, first, $R'(y)$ has a nearly singular mass matrix $C(y)$ (the $\partial F/\partial y$ will also be nearly singular in the tangential directions) and, second, tiny changes in the nodal position in the normal direction will change the $O(c^2)$ eigenvalue of the diagonal block D_i drastically in a relative sense, and thus change $(R'(y))^{-1}$ drastically. Hence $R(y)$ is wildly nonlinear near a geometrical degeneracy and Newton's method cannot be expected to perform well there.

Wathen's results [24], as extended by Miller [11], show, however, that $D^{-1}(y)C(y)$ (where D is the block-diagonal of C) is a far better behaved and more nearly constant matrix than $C(y)$ itself; in fact, in n dimensions it has its eigenvalues in $[\frac{1}{2}, 1 + \frac{n}{2}]$.

We therefore choose to precondition our BCE equations (5.3) by $D^{-1}(y)$ and solve instead the equations with the *preconditioned residual function*

$$\mathcal{R}(y) \equiv D^{-1}(y)R(y) = 0. \tag{5.7}$$

Here (and *not* in Wathen's results discussed above) our C and its diagonal D of course include the internodal regularization terms which force D to be nonsingular. The preconditioned equations (5.7) are much more well conditioned and much more nearly linear than the original equations (5.3).

5.4. A nonlinear "Krylov subspace" accelerator for our modified Newton's method. This is a very worthwhile addition to our 1D and 2D codes. We delay its description until section 9. The principal advantages of this accelerator are that (i) it gives somewhat faster and more robust convergence in general, (ii) it lets us use an old Jacobian $J = [D^{-1}(\bar{\bar{y}})R'(\bar{\bar{y}})]$ (or rather, its saved LU decomposition) much longer (this is of lesser importance in one dimension but gives great savings in two dimensions), (iii) it lets us use an old Jacobian even when $\Delta\tau$ has changed by a considerable factor.

5.5. Other features. We mention a few further features which contribute to the robustness of our code.

(i) Careful attention to roundoff error control in computing residuals.

(ii) A test for negative cell widths Δx after the predictor step, before every residual evaluation, and before accepting a converged solution to the implicit BCE equations (5.3). At times it will be required to cut the Δt to avoid negative Δx's. This test is, of course, disabled for geometrical problems (such as that of section 4.10) where our 1D manifold u is not required to be the graph of a function.

(iii) A *relative* error tolerance on the cell widths Δx. This is absolutely necessary for problems with extremely thin shocks, where Δx may become exceedingly small (say, $\approx 10^{-4}$ or 10^{-5} in normalized units). We usually impose a relative tolerance of 10% at the predictor error level and of 1% at the Newton's convergence test level.

(iv) *Different* absolute predictor error tolerances $\eta_1, \eta_2, \ldots, \eta$ for the *max norm* errors in each of the $u_i, v_i \ldots, x_i$ components. The error tolerances for Newton convergence are set 10 times smaller than these.

6. Some illustrative numerical examples. In this section we use a few simple 1D examples to illustrate the working of the GWMFE method, some of its difficulties and their solutions, and some of its difficulties for which we can only suggest possible solutions for future development.

The computations for this paper were done on a Sun Sparc ELC workstation using double precision arithmetic. For each case we report the following computational statistics: NSTEP, the number of steps of our BDF2 ODE solver; CPU, the number of cpu seconds; NRES, the number of nonlinear residual $\mathcal{R}(y)$ evaluations required for the Krylov-accelerated Newton steps; and NJAC, the number of Jacobian evaluations and their LU factorizations required for the modified Newton's method.

We have often been asked about the accuracy of GWMFE. Is it $O(\Delta x^2)$, $O(\Delta x^3)$, $\ldots$? Such order of accuracy statements have little meaning when the nonuniform Δx's vary over several orders of magnitude. We must admit that our principal analysis of error (on the highly nonlinear problems with sharp moving features to which we apply the method) is *visual*. We frequently compare solutions with different numbers and distributions of nodes by overlaying them graphically on the screen, often under high magnification in regions of interest. Many such overlaid graphical comparisons are shown in the present 1D and 2D papers. Our experience is that once the method succeeds in getting and keeping a few nodes in the features of interest, the convergence as we add nodes thereafter is extremely rapid.

Several authors have reported difficulties in choosing the internodal viscosity coefficient A^2 of (2.20) (now revised to the simplified form (2.21)). With our first example we hope to make more graphic and intuitive the effects of this coefficient and the reasons for our standard choices of it.

6.1. A boundary layer problem. The interaction of convection with a tiny diffusion often produces boundary layers or shocks. The following simple linear problem serves as our model of the "lip" of a shock and helps to explain the behavior of GWMFE (or MFE) in much more complicated settings. Consider the convection-diffusion problem

$$\begin{aligned} &u_t = -u_x + \nu u_{xx} \quad \text{on } 0 \le x \le 1, \quad t \ge 0, \\ &u(0,t) = 0, \quad u(1,t) = 1, \quad u(x,0) = x^{16}, \end{aligned} \tag{6.1}$$

TABLE 6.1
Computational statistics to $t = 500$ for the boundary layer problem (6.1).

ν	A^2	NSTEP	CPU	NRES	NJAC	Max undershoot	Lift-up
10^{-2}	10^{-4}	265	5.2	821	37	.0055	.0060
”	10^{-6}	219	7.1	732	40	.0010	”
10^{-3}	10^{-3}	279	9.1	971	69	.0110	.0015
”	10^{-4}	330	10.2	1088	67	.0044	”
”	10^{-5}	397	12.4	1312	90	.0028	”
”	10^{-6}	446	14.1	1504	104	.00115	”
”	10^{-7}	452	14.1	1502	110	.00046	”
”	10^{-8}	428	14.2	1484	114	.00020	”
”	10^{-9}	433	13.9	1468	103	.000085	”
10^{-4}	10^{-4}	363	12.4	1322	110	.00190	.00033
”	10^{-6}	280	16.6	1732	157	.00085	”

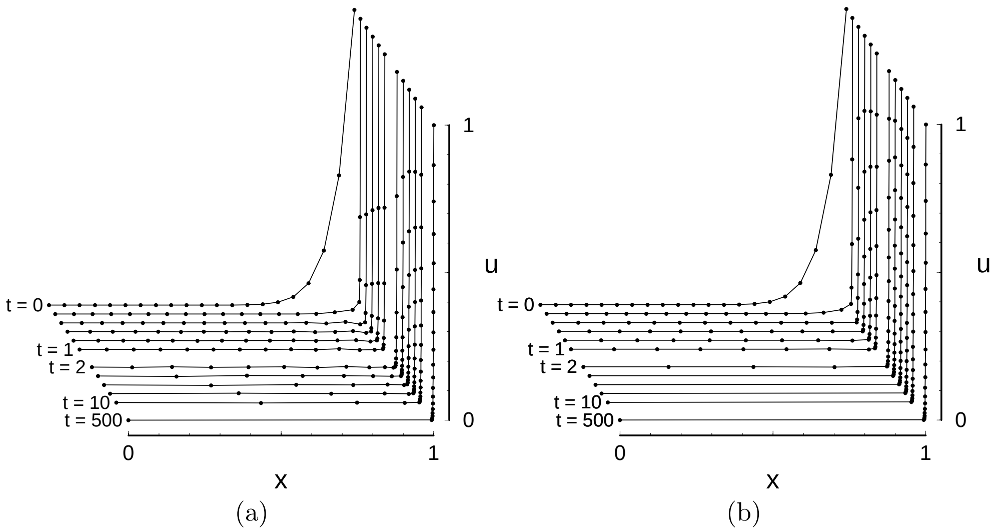

FIG. 6.1. (a) *The boundary layer problem* (6.1) *with an overlarge internodal viscosity coefficient* $A^2 = 10^{-4}$. *Note undershoot at lip of the boundary layer at times* .4–.8. (b) *Same except with a smaller* $A^2 = 10^{-6}$. *Less undershoot. Nodes flow more freely into the boundary layer*

with a small diffusion coefficient ν. Each point on the initial u profile is convected to the right with speed 1 until it collides with the boundary where it merges with a boundary layer of $O(\nu)$ thickness. Immediately after $t = 1$ the entire wave has reached the right-hand boundary and forms a steady-state boundary layer solution of the form $u \sim \exp((x-1)/\nu)$.

Shown in Table 6.1 are the computational statistics to $t = 500$ for several 20-node GWMFE solutions using different values of diffusion coefficient ν and internodal viscosity coefficient A^2. In each case the nodes are initially equally spaced and the predictor error tolerances on the u_i and x_i are $\eta_1 = \eta = 10^{-3}$. Shown in Fig. 6.1a is the solution at a sequence of times with $\nu = 10^{-3}$ and with $A^2 = 10^{-4}$, which is larger than our recommended choice of A^2 (to be stated shortly). Each successive graph is offset by $(.02, -.02)$ in (x, u) to ensure that the nodal positions are clearly visible. Figure 6.2 is a 100× enlargement of the bottom of Fig. 6.1a. Observe especially the undershoots near the “bottom lip” of the boundary layer at $(x, u) \approx (1, 0)$. With

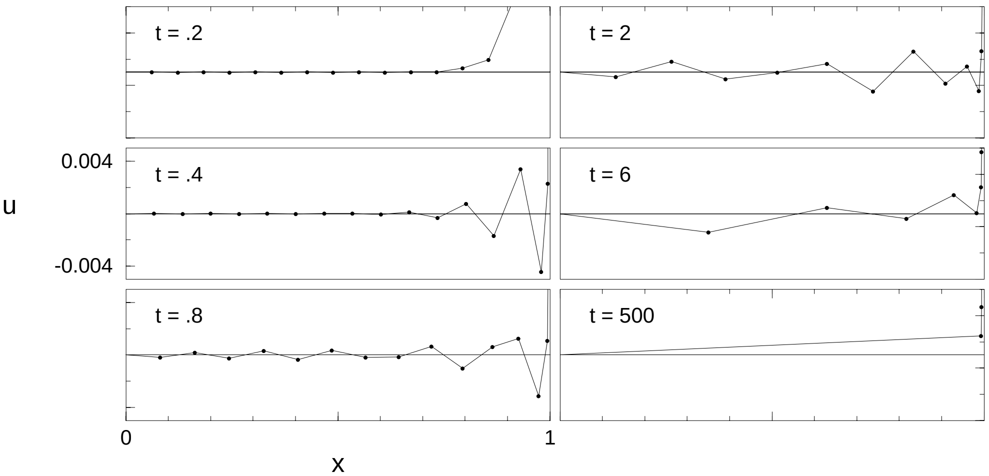

FIG. 6.2. *A* 100× *vertical enlargement of the bottom of Fig.* 6.1a. *Note the oscillations, with a maximum undershoot of* .0044 *at* $t = .4$. *Note the "lift-up" of* .0015 *at* $t = 500$.

$A^2 = 0$ there is no undershoot; each node moves to the right with speed 1 until it meets the boundary layer where it is then lifted up into the layer by diffusion. However, with $A^2 > 0$, the ith node is retarded by a horizontal internodal viscosity force of

$$\approx (A^2/\Delta x_{i+1})(-\Delta \dot{x}_{i+1}) \approx (A^2/\Delta x_{i+1})(\dot{x}_i - 0) \tag{6.2}$$

from the $(i+1)$st cell as it collides with the bottom lip of the boundary layer. This is added to the natural "viscous resistance" force of $\approx (c^2/\Delta x_{i+1})\dot{x}_i$ built into the mass matrix itself, where c is the "break" or "roughness" of the graph at the ith node, as discussed in section 2.6. Thus we have a viscosity of $\approx (A^2+c^2)/\Delta x_{i+1}$ in the $(i+1)$st cell. When $A^2 + c^2 \approx c^2$, this is exactly right to allow the ith node to move with the correct convection speed of 1. But when $A^2 + c^2 \gg c^2$, the node is considerably retarded from its correct speed. The method then behaves more like a fixed-node method, building up spatial oscillations. In fact, in regions where $\Delta x_{i+1} \gg \nu$ there are nearly equal oscillations (see [8]). The result is that the method should develop a "roughness" of $c^2 \approx A^2$ so that the nodes can convect with approximately their correct +1 speeds, especially in that final cell just before the boundary layer, where the approximation $-\Delta \dot{x}_{i+1} \approx \dot{x}_i$ of (6.2) is most appropriate.

This behavior appears to be evident in actual computations. Notice in Fig. 6.2, where $A^2 = 10^{-4}$, that the undershoot at the lip of the boundary layer continues in time as each new node approaches the lip, with nearly equal spatial oscillations across the whole domain. Table 6.1 shows that the maximum undershoot decreases with decreasing A^2. The solution using a much smaller $A^2 = 10^{-6}$ is shown in Fig. 6.1b. Not only are the undershoot oscillations much smaller, but the nodes are much freer to move toward the boundary layer with approximately characteristic velocity.

On the other hand, the "break" c becomes numerically indeterminate when it approaches the size of the predictor error tolerance η_1. Although not very apparent for this simple problem, this generally leads to the method becoming less efficient when $A^2 \lesssim \eta_1^2$. Consider, for example, the problems of Tables 6.2 and 6.3, where the increasing computational effort is reflected by the statistics CPU, NRES, and NJAC.

In view of the preceding, we can now explain our recommended *standard choices* of the parameters η_1, η, and A^2. We usually normalize the horizontal and vertical dimensions to $O(1)$ variations, *then choose the predictor error tolerances* $\eta_1 \approx \eta \approx 10^{-3}$ *in these normalized units.* This is about as much accuracy as our graphics will show, and this (with the additional 10% *relative* accuracy on the Δx's) seems to be enough predictor accuracy to ensure reliable Newton convergence. *We then choose* $A^2 = ((.5 \text{ to } 5) \times \eta_1)^2$. Greater A^2 often leads to visible roughness or oscillations on the lips of shocks. A much smaller A^2 often leads to numerical indeterminacy at geometrical degeneracies and a resultant degradation of computational efficiency. This recommended choice of A^2 includes considerably smaller A^2 than were previously recommended in [9] and [10]. This is due to the introduction of the diagonal preconditioning of section 5.3 and the Krylov subspace accelerator of section 5.4. These two innovations have made the code far more robust, and we can now compute efficiently with considerably smaller A^2.

Also evident in Figs. 6.1 and 6.2 is the phenomenon of *"lift-up"* on the lip of a boundary layer or shock, which was extensively analyzed in [8, p. 168] for MFE and for which GWMFE gives a satisfactory cure. Notice that at steady state all nodes have entered the boundary layer with the final node perched on the lip a bit inside the layer. Hence the single cell that remains outside the layer lifts up across the whole solution domain, here by an amount $\approx .0015$. It was shown in [8] that the steady-state behavior of the original MFE method for this problem (without tunable internodal pressure forces) is to move all nodes well within the boundary layer with exactly equal *vertical* spacing. Thus, with the present 20 cells for this problem, the MFE "lift-up" would be an intolerable 1/20! Gradient-weighting de-emphasizes the steep portions of the graph, thereby moving the final node farther out on the lip of the boundary layer and decreasing the "lift-up." Decreasing the diffusion coefficient ν steepens the boundary layer, and the "lift-up" decreases correspondingly, as shown in Table 6.1.

6.2. Burgers's equation with "double-sine" initial values. Consider the standard test problem (introduced in [2])

$$\begin{aligned} &u_t = -uu_x + \nu u_{xx} \text{ on } 0 \le x \le 1,\ t \ge 0, \\ &u(0,t) = u(1,t) = 0, \quad u(x,0) = \sin 2\pi x + \frac{1}{2}\sin \pi x, \end{aligned} \tag{6.3}$$

with a quite small diffusion coefficient of $\nu = 10^{-4}$. This is a problem for which Zegeling and Blom [18] reported great difficulty in finding a decent internodal viscosity coefficient A^2 for GWMFE. We now explain why, and in the following section present an easy remedy through *vertical rescaling.*

We solve (6.3) with 40 GWMFE nodes, initially equally spaced, with various values of A^2, and with the standard choice $\eta_1 = \eta = 10^{-3}$. Shown in Fig. 6.3 is the solution with $A^2 = 10^{-7}$ chosen several times smaller than the smallest standard choice of A^2. This solution appears to be quite accurate since it agrees with more accurate GWMFE computations to within the width of the lines of our graphics (even under considerable magnification). By $t = 2$ the solution has formed a shock that collides with the boundary at $t \approx 1.44$. Shown in Fig. 6.4a are highly magnified details of the bottom lip of the shock as the collision occurs.

However, for this problem, the GWMFE solution crashes as the shock collides with the boundary if A^2 is chosen larger than a critical value $(A^2)^\star$. As is evident in Table 6.2, $(A^2)^\star \approx .5 \times 10^{-6}$. Shown in Fig. 6.4b are the magnified details of this collision with $A^2 = 25 \times 10^{-6}$ (too large). Notice that the solution first "buckles"

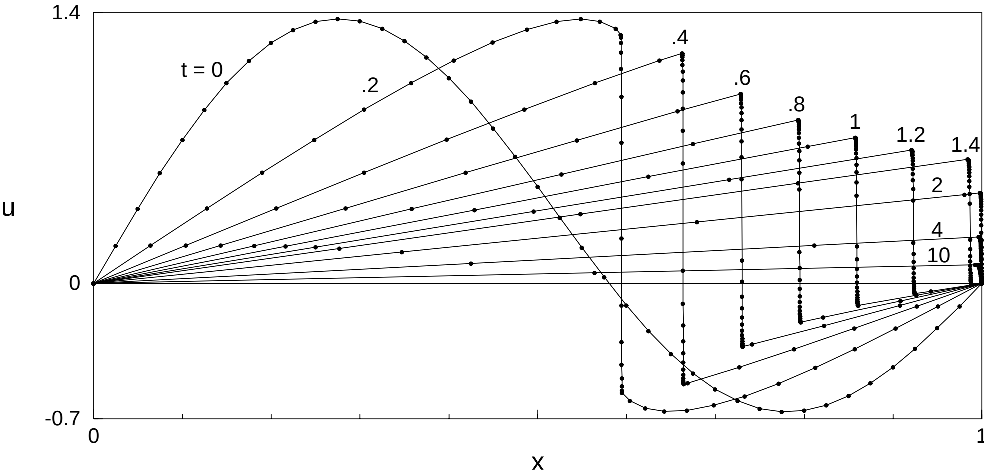

FIG. 6.3. *Burgers's equation problem* (6.3) *with diffusion coefficient* $\nu = 10^{-4}$ *and a small internodal viscosity coefficient* $A^2 = 10^{-7}$.

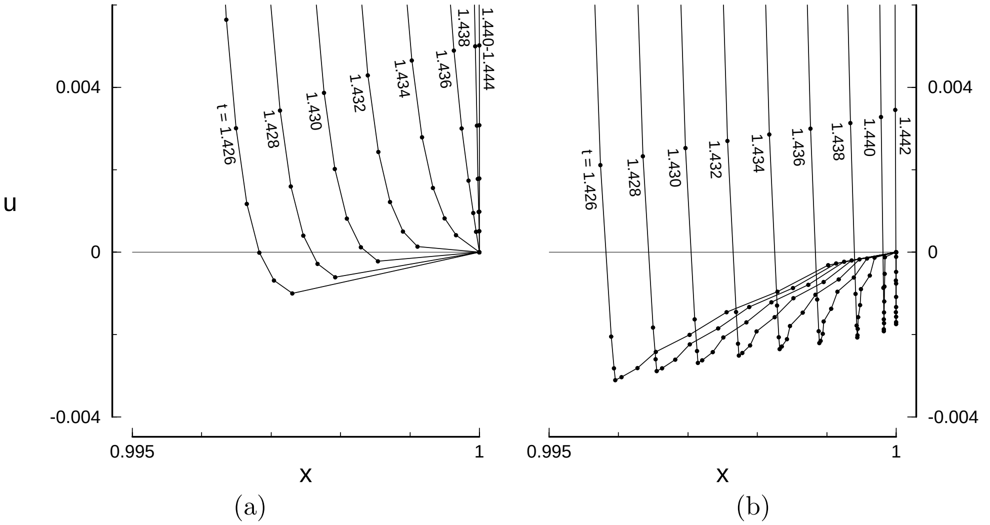

FIG. 6.4. (a) *Burgers's equation with* $\nu = 10^{-4}$ *and* $A^2 = 10^{-7}$, *as in Fig.* 6.3. *Details of bottom lip of shock as shock collides with boundary at* $t \approx 1.44$ *(enlarged* 200× *horizontally and* 100× *vertically).* (b) *Same but with an overly large value of* $A^2 = 25 \times 10^{-6}$. *Solution "buckles" in cells trapped between bottom lip and boundary, due to overlarge "dynamic compressions" in these cells. Solution by* $t = 1.438$ *has formed a "kink" from which it can never recover.*

in the thin nonsteep cells trapped between the boundary and the bottom lip of the shock as the shock comes crashing in to the boundary. In time this buckling develops into an ever sharper and thinner "kink" that causes the time step to drop to zero (or at least to remain exceedingly small thereafter).

The reason for this buckling with $A^2 \geq (A^2)^\star$ is quite easily explained. In these trapped nonsteep cells there is a "dynamic compression" of magnitude $A^2(-\dot{\ell}_i/\ell_i)$ due to the internodal viscosity regularization (see (2.20)–(2.21)). On the other hand, the diffusion term νu_{xx} generates a "surface tension" of magnitude $\nu(1+u_x^2)$ in the ith cell (see (2.18) and (4.10)). Thus in these nonsteep cells the total tension is

Table 6.2
Statistics to $t = 10$ for the Burgers equation problem (6.3).

A^2	NSTEP	CPU	NRES	NJAC	Comments
25×10^{-6}					Aborts with kink
10×10^{-6}					Aborts with kink
1×10^{-6}					Aborts, 2 tiny cells
$.5 \times 10^{-6}$					Aborts, 1 tiny cell
10^{-7}	534	30.7	2309	218	Good solution
10^{-8}	690	40.9	3101	292	Good solution
10^{-9}	1551	83.8	6786	510	Good solution

$\approx \nu - A^2(-\dot{\ell}_i/\ell_i)$. The buckling occurs because these cells go into a net compression rather than tension due to the large values of $-\dot{\ell}_i/\ell_i$ occurring there. Dimensional analysis shows that the critical value $(A^2)^\star$, above which the tension turns negative and hence buckling and a kink appear, should be proportional to ν^2. This in fact is what we observe computationally; the observed critical value is $(A^2)^\star \approx .5 \times 10^{-4}$ for $\nu = 10^{-3}$ and $(A^2)^\star \approx .5 \times 10^{-6}$ for $\nu = 10^{-4}$.

We are therefore stuck with the embarrassing conclusion that with this small diffusion coefficient of $\nu = 10^{-4}$ the solution will crash due to buckling for all A^2 above a critical value which happens to lie in the middle of our recommended standard choices of $A^2 = ((.5 \text{ to } 5) \times \eta_1)^2 = (.25 \text{ to } 25) \times 10^{-6}$. (The situation with a larger diffusion coefficient $\nu = 10^{-3}$ is far more comfortable; we get good and decently efficient solutions for all A^2 in the range 10^{-6} to 5×10^{-5}.)

6.3. Vertical rescaling. A way out of our dilemma, which leads to a far wider and more robust choice of A^2, is through a *vertical rescaling* of the problem by a large factor M. Recall that implicit in our geometrical-mechanical discussions of GWMFE in section 2 was the assumption that a choice of the ratio between the horizontal and vertical scales had been made and fixed. We have usually normalized variables such that the horizontal and vertical variations are $O(1)$. This is not necessary, however; we are free to choose the vertical scale.

In the present situation, where an exceedingly steep shock is colliding violently with the boundary, it seems intuitive that it may be better to flatten the vertical dimensions by a large factor M so that the colliding shock (and any incipient vertical oscillations in the trapped cells) have smaller apparent slopes. This is easily accomplished by replacing "u" by "Mu" everywhere in (6.3). The variation in u has now been reduced from $O(1)$ to $O(1/M)$. We should therefore reduce our predictor error tolerance η_1 on the u_i accordingly, from the standard choice of $\eta_1 = 10^{-3}$ to $\eta_1 = 10^{-3}/M$. This allows us to take our standard choice of $A^2 = ((.5 \text{ to } 5) \times \eta_1)^2$ a significant M^2 times smaller than before, without degrading the computational efficiency due to too-small A^2.

We try a "$10u$" vertical rescaling of problem (6.3), with $\eta_1 = 10^{-4}$, $\eta = 10^{-3}$. Shown in Table 6.3 are the computational statistics for trials using various values of A^2. The critical value $(A^2)^\star$ above which buckling occurs is only slightly higher than before. The values of A^2 for which the computation remains decently efficient, however, have been shifted downward by a factor of $M^2 = 100$. Hence, the choice of good values for A^2 has been made considerably wider and more robust than before. In fact, Table 6.3 shows that we get good and decently efficient solutions for all values of A^2 between 10^{-8} and 10^{-5}. For example, Fig. 6.5 shows the results with the standard choice $A^2 = (5 \times \eta_1)^2 = 25 \times 10^{-8}$.

Table 6.3

Statistics to $t = 10$ for the Burgers equation problem (6.3) *with "10u" vertical rescaling.*

A^2	NSTEP	CPU	NRES	NJAC	Comments
4000×10^{-8}					Aborts with kink
2000×10^{-8}					Aborts with kink
1000×10^{-8}	261	12.5	886	061	Good solution
100×10^{-8}	315	16.7	1193	095	Good solution
25×10^{-8}	396	20.8	1579	129	Good solution
10×10^{-8}	468	26.0	1901	161	Good solution
10^{-8}	721	41.6	3115	260	Good solution
10^{-9}	1854	128.0	>10,000	843	Good solution

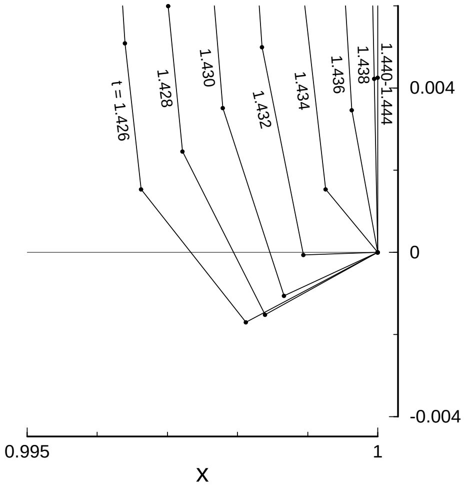

Fig. 6.5. *Foot of the shock for Burgers's equation. Same as Fig.* 6.4 *except with "10u" vertical rescaling and with the standard choice $A^2 = (5 \times \eta_1)^2 = 25 \times 10^{-8}$. No buckling. Good and efficient solution.*

A "$100u$" rescaling gives an even wider latitude in the choice of good values for A^2. However, for $M \geq 100$ with $\nu = 10^{-4}$ the rescaled steepness of the shocks begins to be reduced to such an extent that the gradient-weighting by the factor $w = (1 + u_x^2)^{-1/2}$ no longer forces the nodes well out onto the lip of the shocks. Hence we then begin to see the phenomenon of excessive MFE "lift-up" on the lip of the shocks discussed in section 6.1. The "small-slope limit of GWMFE," that is, its limit as $M \to \infty$, is of course the original MFE method, as is discussed in [12].

6.4. Burgers's equation with a strong nonlinear source. Let us consider traveling wave near-shock solutions of the PDE

$$u_t = -(f(u))_x + \nu u_{xx} + \frac{1}{\nu} g(u) \tag{6.4}$$

with a tiny diffusion coefficient ν and an extremely large source term $g(u)/\nu$, where $g(u)$ is a function which is zero at $u = 0$ and 1 and positive for $0 < u < 1$. We suppose this thin traveling wave near-shock to have speed c (thus $u_t = -cu_x$) and values $u_- \approx 1$ and $u_+ \approx 0$ at points $x_- < x_+$ slightly behind and ahead of the front.

Thus, integrating (6.4) across the front we have the shock speed c given by

$$c(u_- - u_+) = (f_- - f_+) + \nu(u_x^+ - u_x^-) \quad + \quad \int_{u_+}^{u_-} g(u)(-\nu u_x)^{-1}\, du. \tag{6.5}$$

When $g(u) = 0$ we have a pure conservation law and, since ν is tiny and u_x is decently small outside the shock, the shock speed c is given by the usual Rankine–Hugoniot condition $c = (f_- - f_+)/(u_- - u_+)$. Thus, there is absolutely no need for a numerical method to resolve the fine-scale structure of the shock profile in order to obtain the proper shock speed c. All that is required is that the method itself be conservative and that it avoid large oscillations at the lips of the shocks. There are many highly specialized methods designed to handle such pure conservation laws; typically they function by automatically adding just enough *numerical diffusion* within and on the lips of shocks to avoid oscillations and to spread the shock monotonically over several of the uniformly spaced grid points.

The situation changes drastically when $g(u) \neq 0$, however. Now, since u_x can be expected to be proportional to ν^{-1}, the final integral in (6.5) may contribute significantly to the shock speed c. This is a problem where one must accurately *resolve* the fine-scale profile of the shock to obtain the correct shock speed (in fact, one must resolve $g(u)\ u_x^{-1}$ here). Those methods that depend on enough numerical diffusion to spread the shock over several grid points (thus increasing u_x^{-1} in (6.5)) are doomed to disaster here. Equation (6.4) is a simple scalar prototype for such problems; other problems in this class include flame fronts with realistic chemistry, flow plus reaction in chemical reactors, surfactant or CO_2 or steam flooding in tertiary oil recovery, and arsenic ion implantation in semiconductor manufacturing. *It is for this class of problems demanding fine-scale resolution that* GWMFE *is principally intended.* For problems in pure conservation form we expect that other more highly specialized methods (which do not achieve or attempt fine-scale resolution) will be more efficient.

In particular, we consider (6.4) with $f(u) = u^2/2$ and with the source function

$$g(u) = C_1 \cdot 2^8 u^2 (1-u)^6 \tag{6.6}$$

on the interval $-1 \leq x \leq +1$, with fixed Dirichlet boundary conditions and the piecewise linear initial values $u(x, 0) = 1$ on $[-1, -5\nu]$, $= 0$ on $[5\nu, 1]$, and linear on $[-5\nu, 5\nu]$. By varying the "combustion coefficient" C_1 we can make the source term $g(u)$ either more or less significant. The mean value integrals $[g\alpha^i]_i$ occurring in the GWMFE inner products are computed exactly. Thus we are assured that any strange behavior of the method is not due to quadrature error.

We choose $\nu = 10^{-3}$ and solve on the interval $-1 \leq x \leq 1$, but show graphics only on $-.2 \leq x \leq .8$ in order to focus on the important features. The GWMFE initial values have $2, 5, 20, 10, 10$ equal cells on each of the x intervals $(-1, -100\nu)$, $(-100\nu, -5\nu)$, $(-5\nu, 5\nu)$, $(5\nu, 100\nu)$, $(100\nu, +1)$, for a total of 47 cells. The predictor tolerances on u and x are $\eta_1, \eta = 10^{-3}, 10^{-3}$; the internodal viscosity is the standard $A^2 = (5 \times \eta_1)^2 = 25 \times 10^{-6}$.

Shown in Figs. 6.6a, b and 6.7a are the GWMFE solutions with values of $C_1 = 0$, .02, and .04. The statistics for NSTEP, CPU, NRES, NJAC on these three runs are 61, 4.4 sec, 172, 14; 77, 6.0 sec, 240, 20; 132, 9.4 sec, 429, 27.

The run with $C_1 = .0$ is just the Burgers equation; the steady-state traveling wave forms quickly and moves to the right with speed $c = \frac{1}{2}(u_- - u_+) = .5$. The run with

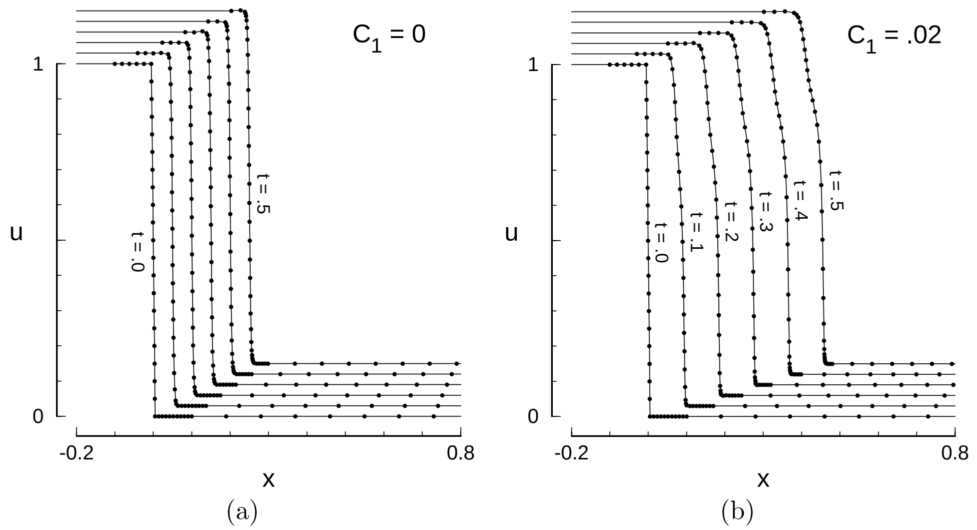

FIG. 6.6. (a) *Burgers's equation plus a zero strength source term,* $C_1 = .0$. *Traveling wave with speed* .5. $A^2 = 25 \times 10^{-6}$. (b) *Burgers's equation plus a moderately strong source term,* $C_1 = .02$. *Traveling wave with speed* $\approx .93$. $A^2 = 25 \times 10^{-6}$. *Successive graphs are offset vertically.*

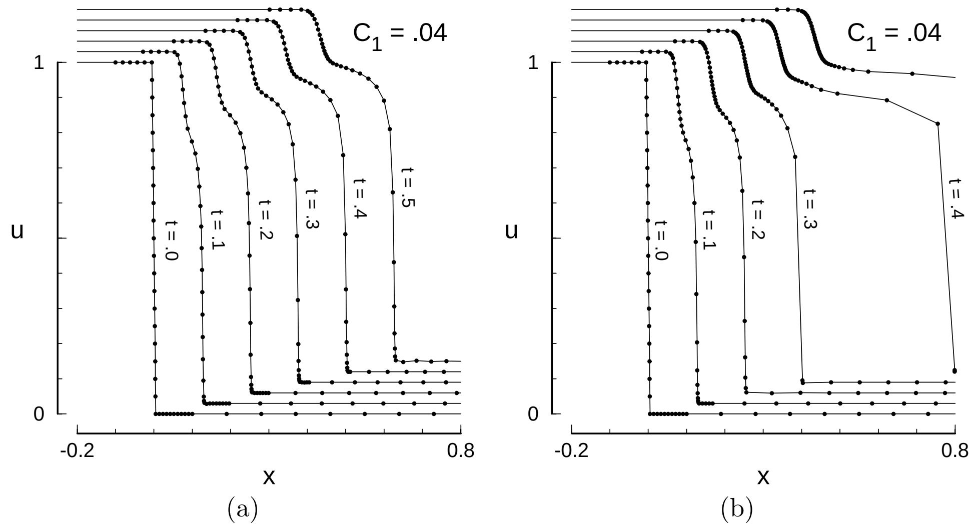

FIG. 6.7. (a) *Burgers's equation plus a strong source term,* $C_1 = .04$. *Leading front (for* $0 \le u \lesssim .8$*) with speed* ≈ 1.22*, fueled by the source. Trailing pure Burgers's equation front (for* $.8 \lesssim u \le 1.$*) with speed* $\approx .9$. $A^2 = 25 \times 10^{-6}$. (b) *Same except with a smaller* $A^2 = 4 \times 10^{-6}$. *Solution by* $t = .3$ *has exhausted its nodes at foot of the front. Disastrous solution thereafter.*

the medium value of $C_1 = .02$, by time $t \approx .2$, forms a steady-state traveling wave with significantly higher speed, $c \approx .93$. In Fig. 6.7a are the solution profiles with the larger combustion coefficient of $C_1 = 0.04$. In this case a single simple traveling wave solution does not develop. Instead, the $\nu^{-1}g(u)$ source term has become so large that this source seems to be creating its own traveling wave solution (for $0 \le u \lesssim .8$) with constant speed of $c \approx 1.22$. Trailing behind it is an essentially pure Burgers equation front (for $.8 \lesssim u \le 1.$) with constant speed $c \approx .9 = \frac{1}{2}(1. + .8)$.

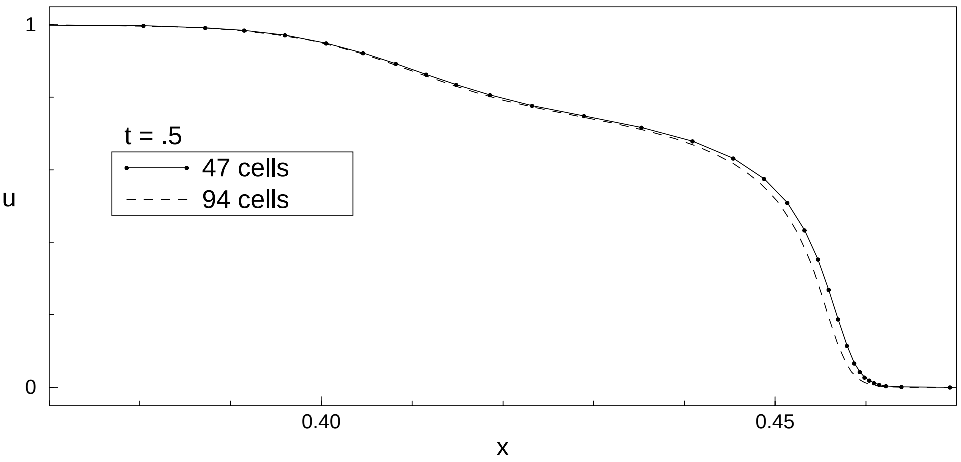

FIG. 6.8. *Comparison at $t = .5$ of the* 47*-cell solution of Fig.* 6.6b *with a more accurate 94-cell solution (enlarged* 10× *horizontally).*

The case $C_1 = .04$ is more difficult to compute than the cases with smaller C_1, and illustrates some of the difficulties in using GWMFE. This problem is of the form (2.15)–(2.18), with a "solar wind" (a, b) incident upon the GWMFE solution manifold whose horizontal velocity $a(u) = u$ is of order 1 and whose vertical velocity $b(u) = \frac{1}{\nu} g(u)$ reaches values of order 100. This strong vertical wind makes it difficult for GWMFE to achieve a near steady state in the leading portion of the front. Nodes which are out of balance in this vertical portion drift upward, then are dragged backward into the less vertical transition zone between the two fronts and tend to pull neighboring nodes along with them because of the large $A^2 = 25 \times 10^{-6}$ internodal viscosity forces. This process produces a gradual migration of all the nodes upward through the front which then exhausts all the nodes at the foot of the front. Increasing A^2 helps to slow the nodes in the vertical portion; but on the other hand it contributes to dragging the nodes backward into the transition zone.

For the above reasons we have had to resort to tuning in two different ways: by choice of the A^2 and by judicious placement of the nodes in the initial values. In fact, we show in Fig. 6.7b the results for $C_1 = .04$ of using a smaller $A^2 = 4 \times 10^{-6}$, all other parameters being the same. This solution agrees perfectly with that of Fig. 6.7a through $t = .2$, but by $t = .3$ the solution has exhausted all its nodes in the leading front and for $t > .3$ the solution is disastrous there due to the exhaustion of nodes.

This example also illustrates the need for Kuprat's "graph-massage" routine [15]. That routine would undoubtedly create new nodes in the foot and vertical portions of the leading front and remove some of the excess nodes that are accumulating at the base of the following front.

Nevertheless, so long as an adequate number of nodes are retained in the front to resolve its detailed profile, GWMFE does appear to yield highly accurate solutions. Shown in Fig. 6.8 is a magnified comparison at the final time between the 47-cell solution of Fig. 6.6b and a much more accurate 94-cell solution using 10× tighter predictor error tolerances. The front position appears to be in error by only ≈ .2%.

6.5. Nodes crashing into shocks. There is one feature of GWMFE which makes it not inaccurate at all but perhaps inefficient when compared with some specialized methods for pure hyperbolic conservation laws.

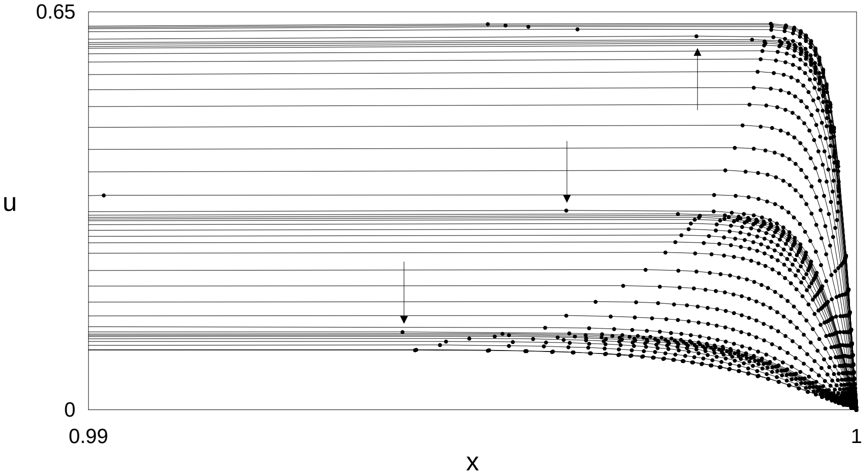

FIG. 6.9. *Burgers's equation problem* (6.3) *for* $t \geq 1.46$ *with* $\nu = 10^{-4}$*, "*$10u$*" vertical rescaling, and standard choice of* $A^2 = 25 \times 10^{-8}$*.* 100× *horizontal enlargement of right boundary with solution every two time steps showing nodes (marked) crashing in from the left into the lip of the shock.*

We repeat the computation of problem (6.3) for Burgers's equation with $\nu = 10^{-4}$, with "$10u$" vertical rescaling, with $\eta_1, \eta = 10^{-4}, 10^{-3}$, and with the standard $A^2 = (5 \times \eta_1)^2 = 25 \times 10^{-8}$. This gives a very typical and trouble-free solution. However, one finds that the time step Δt of the method is repeatedly increasing then decreasing drastically throughout the course of the computations. A closer magnified inspection of the graphs reveals the reason for this repeated fluctuation in Δt. In every case (except the collision of the shock with the boundary at $t \approx 1.44$) a large decrease in Δt occurs because some one node, moving at characteristic velocity (*as it should* to give the true solution outside the shocks), comes crashing into the lip of a shock. This is shown happening 3 times in Fig. 6.9 which is a 100× horizontal enlargement of the right edge of the graphs for all times after the boundary shock has fully formed at about $t = 1.46$. The output frequency here is every 2 BDF2 steps. In each instance we see a node (marked) come hurtling in from the left with the characteristic velocity. Only as the node gets to within the $O(\nu)$ characteristic width of the lip of the shock do the repulsive horizontal forces of diffusion take hold to slow it down. As this happens the Δt decreases sharply, the first node on the lip of the shock is nudged aside for the incoming node to take its place, and the other nodes in the shock shift over accordingly. The Δt then begins a steady climb back upward.

Each collision of a new node into the shock requires not only a sharp reduction in Δt but also several expensive Jacobian re-evaluations. The basic structure of the Jacobian of course changes drastically for the incoming node and for those few nodes on the lip of the shock which get nudged over; elsewhere the Jacobian probably remains largely unchanged.

The behavior described above probably decreases the computational efficiency of these moving node methods considerably. This would be even more true in multidimensions where there are many more nodes. We do not at the moment have a simple solution to suggest for this problem. This problem strongly suggests the need for work on spatially variable time steps for our stiff ODE solver.

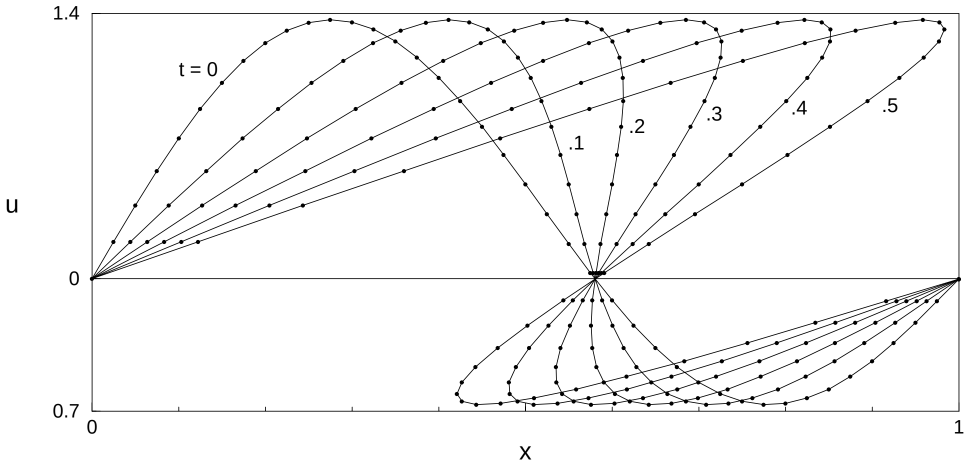

FIG. 6.10. *Burgers's equation problem* (6.3), *but with* no diffusion.

6.6. Multiple-valued manifold solutions. Consider the Burgers equation problem (6.3), except with the diffusion term νu_{xx} *deleted* from the PDE. Shown in Fig. 6.10 is the GWMFE solution using the same number of nodes and internal code parameters as those for Fig. 6.3.

This example illustrates certain remarks made in sections 2.3 and 2.5. The GWMFE method converts the PDE (2.15) into the normal motion form (2.16) and then proceeds to approximately solve for a manifold solution of (2.16). The two forms are equivalent so long as the manifold solution remains the graph of a function. However, (2.15) ceases to have a (classical) solution when the graph becomes vertical and begins to go multiple-valued, whereas the manifold solution of (2.16) folds over and continues to exist. This foldover happens smoothly with no decrease in the time step. The computational statistics to $t = 0.5$ for Fig. 6.3 are NSTEP, NRES, NJAC = 274, 862, 76, while those for Fig. 6.10 are 8, 11, 7.

In fact, the GWMFE (or MFE) method (without regularization) is *exact* on this problem, or on any problem of the form (2.16) where a and b are linear functions of x and u (see [32]), because any exact solution with piecewise linear initial values *remains* piecewise linear. Hence, because of the variational formulation (2.7), the GWMFE (or MFE) method recovers that exact piecewise linear solution (for which $\psi \equiv 0$).

Some users of MFE have applied it to problems of the form (2.15) with *no* diffusion and have complained of "node overtaking"; that is, the solution becomes multiple-valued. But of course it does, because that is the behavior of the true manifold solution of (2.16) which GWMFE (and its small-slope limit MFE) is approximating. If, on the other hand, these users *have* added a νu_{xx} term to the equation with a very tiny ν, then their problems of "node overtaking" may be because their stiff ODE solver is not equipped with special features (such as those in section 5.5) to handle the tiny Δx's in the shock or to handle nodes crashing into shocks as in section 6.5.

Comparing Figs. 6.3 and 6.10, the reader should understand our statement in section 2.5 that "within the setting of *manifold solutions* of (2.16) and (2.18) it is clear that the role of the tiny diffusion coefficient is to *cause* these near-shocks, for without it the manifold solution would just smoothly fold over."

7. Some nontrivial applications.

7.1. Drift-diffusion equations for semiconductors. We consider a greatly simplified 1D model problem for semiconductors (see [34]). Let $p(x,t)$ denote the density of mobile positively charged "holes" in a semiconductor and let $v(x,t)$ denote the voltage or electrostatic potential there. Let $d(x)$ be the given dopant density of immovable negative charges doped in the semiconductor and assume for this simplified problem that $n(x,t)$, the density of mobile electrons, is zero. The total excess density of negative charges is thus $d-p$; hence v satisfies Poisson's equation

$$v_{xx} = d - p. \tag{7.1}$$

The holes drift with a velocity proportional to the electric field $-v_x$; we choose normalized units such that this drift coefficient is 1. The holes also diffuse, with a tiny diffusion coefficient of λ. Hence, the flux of holes is $-v_x p - \lambda p_x$. Assuming no generation or annihilation of holes, the conservation law for holes is thus

$$\begin{aligned} p_t &= (v_x p + \lambda p_x)_x \\ &= v_{xx} p + v_x p_x + \lambda p_{xx}. \end{aligned} \tag{7.2}$$

We then use (7.1) to replace the v_{xx} term on the right-hand side of (7.2). This is an important point; it lets us replace a second-order term (which would involve delta functions in v_{xx} for the piecewise linear v of GWMFE) by a zeroth-order term. Without this substitution GWMFE does not give very good results. Thus (7.2) becomes

$$p_t = (d-p)p + v_x p_x + \lambda p_{xx}. \tag{7.3}$$

Mainly for the convenience of having our PDEs in our standard transient form (2.1), and thereby avoiding the need to write a special code for differential-algebraic equations, we also add an ϵv_t term to the voltage equation (7.1), with a small constant ϵ, yielding the slightly parabolic voltage equation

$$v_t = \frac{1}{\epsilon}(v_{xx} - (d-p)). \tag{7.4}$$

In normalized units we let $d \equiv 1$ and we solve (7.3) and (7.4) on the unit interval $0 \le x \le 1$. We prescribe the fixed Dirichlet boundary conditions $p = 1$, $v = 0$ at the left boundary and $p = 10^{-8}$, $v = .125$ at the right boundary. We also prescribe certain initial values $p = p_0(x)$ and $v = v_0(x)$ at $t = 0$.

Thus we have the problem, for $0 \le x \le 1$, $t \ge 0$,

$$p_t = (1-p)p + v_x p_x + \lambda p_{xx}, \tag{7.5a}$$

$$v_t = \frac{1}{\epsilon}(v_{xx} - (1-p)), \tag{7.5b}$$

$$p(0,t) = 1, \quad p(1,t) = 10^{-8}, \quad p(x,0) = p_0(x),$$

$$v(0,t) = 0, \quad v(1,t) = .125, \quad v(x,0) = v_0(x).$$

We make one final adjustment to problem (7.5) before applying GWMFE. Here we want to know the detailed structure of p also for the very small values of p. We therefore solve not for $p(x,t)$ but for its logarithm $u(x,t)$, where $p = e^{ku}$ and where the normalizing constant k is chosen such that $p = 10^{-8}$ corresponds to $u = -1$.

Making this substitution, (7.5a) is replaced by

$$u_t = \frac{1}{k}(1 - e^{ku}) + (v_x + \lambda k u_x)u_x + \lambda u_{xx}. \tag{7.6}$$

Note that in the transition from (7.5a) to (7.6) we have cancelled a factor of $e^{ku} = p$. Since p ranges from 10^{-8} to 1, we are now emphasizing the very low regions of the solution by a weighting of 10^8 times more than previously.

Notice that the electric field $-v_x$ will have a tendency to sweep the holes out of any region in which this field is nonzero. Thus, these drift-diffusion equations tend to produce at steady state a "depleted region" (where $p \approx 0$, $v_x \neq 0$, and v satisfies $v_{xx} \approx d$) and a "charge-neutrality" region (where $p \approx d$, $v_{xx} \approx 0$, and $v_x \approx 0$). Joining these two regions is a thin transition front where the effects of the tiny diffusion coefficient λ counterbalance the convective effects of the small residual electric field near the edge of the depleted region. We have chosen the "gate voltage" $v = .125$ such that this transition front should stabilize at the midpoint, $x \approx .5$.

We choose a small diffusion coefficient, $\lambda = 10^{-4}$. Shown in Fig. 7.1 are our GWMFE solutions for "holes" u and voltage v at times $t = 0, 1, 5$, and 40 (steady state). The successive graphs for u are shifted by .04, .04 in x and u.

The initial function $u_0(x)$ was chosen as a ramp which is $\equiv 0$ for $0 \leq x \leq .7$, linear for $.7 \leq x \leq .8$, and $\equiv -1$ (corresponding to $p(x) \equiv 10^{-8}$) for $.8 \leq x \leq 1$. There are $4, 6, 10, 6, 4$ equal width cells in the intervals $(0, .6), (.6, .7), (.7, .8), (.8, .9)$, and $(.9, 1)$.

The voltage initial function $v_0(x)$ was chosen to be simply linear on $[0, 1]$. The voltage $v(x, t)$ then quickly relaxes toward the steady-state solution of its equation (7.4) for the given $u_0(x)$ initial values, shoving some of the GWMFE nodes around with it as it does. A better initial value function $v_0(x)$ would be obtained by solving for the steady-state FE solution of (7.1) at $t = 0$ with the nodes fixed at their initial positions. This is what we in fact do with our 2D code, but for one dimension we don't bother and it makes little difference in the results.

Our parabolic coefficient ϵ in (7.4) was chosen to be $\epsilon = 10^{-1}$. Since the fundamental eigenmode of the heat equation (7.4) on $[0, 1]$ with Dirichlet boundary conditions has a decay time constant of $T = \epsilon/\pi^2$ seconds, which is $\approx 10^{-2}$ here, and since the $p(x, t)$ in (7.3) changes at a much slower rate than that, our substitution of the parabolic (7.4) for the elliptic (7.1) has very little effect on the results.

This 30-cell GWMFE solution shown in Fig. 7.1 appears to be extremely accurate. In fact, comparison of it with the solution with 60 cells and tighter tolerances shows that it appears to be accurate at all the times shown to within the width of the lines; therefore we don't bother to show that comparison.

Now we discuss the choice of the internal parameters of GWMFE. The only sensitive choice here was the choice of the PDE weights $w_1 = 1$ and $w_2 = .01$ in (3.2) for the relative weighting of the PDEs (7.6) for u and (7.5b) for v. Because of the large $\frac{1}{\epsilon}v_{xx}$ term in (7.5b) we had to unweight this equation quite a bit. Values of w_2 10 times smaller or larger also gave decent results. However, with a very small $w_2 = .0001$ the convective terms in the holes equation dominated and tended to sweep most of the nodes from the depleted region, thus leaving only a fairly coarse grid to resolve the $v_{xx} \approx 1$ voltage equation in that region. On the other hand, with a very large $w_2 = 1$ the voltage equation $v_{xx} \approx 1$ in the depleted region dominated in the positioning of the nodes, leaving too few nodes to accurately resolve the transition front. The

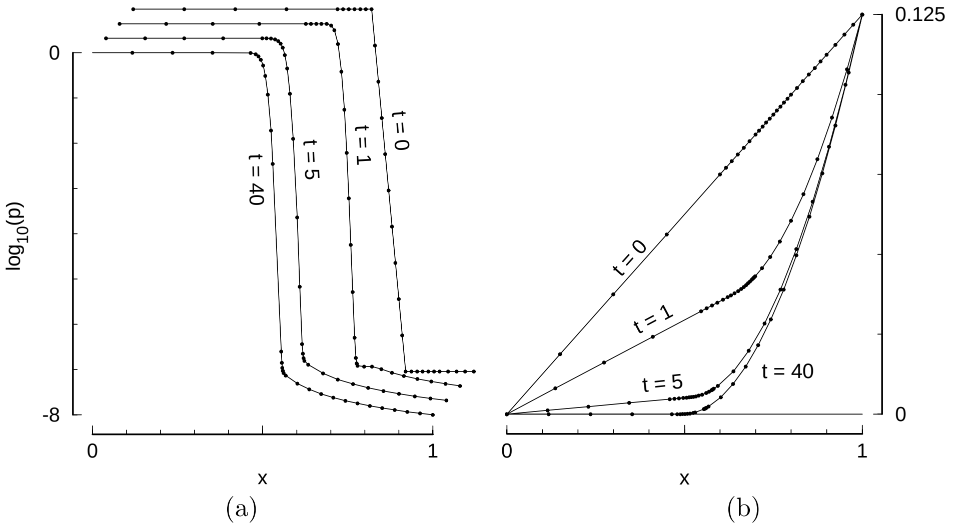

FIG. 7.1. *Drift-diffusion equations with diffusion coefficient* $\lambda = 10^{-4}$ *at times* 0, 1, 5, *and* 40 *(steady state);* (a) *the log of the hole density p;* (b) *the voltage.*

other parameters were chosen in standard fashion, with $\eta_1, \eta_2, \eta = 10^{-3}, 10^{-3}, 10^{-3}$ (the predictor error tolerances on the u_i, v_i, x_i) and $A^2 = 4 \times 10^{-6}$, 4×10^{-4} (the internodal viscosities on the u and v graphs).

The statistics on this run to steady state at $t = 40$ are NSTEP, CPU, NRES, NJAC = 163, 11.8 sec, 512, 27.

7.1.1. A family of steady-state solutions. One of the advantages of our stiff ODE solver for GWMFE is that it allows us to easily compute a smooth one-parameter family of steady-state solutions by varying some problem parameter extremely slowly in time. Consider, for example, a time-dependent diffusion coefficient $\lambda(t)$.

We start with the same 30-cell initial values at $t = 0$ as in Fig. 7.1, keeping $\lambda(t) \equiv 10^{-3}$ for $t \leq 40$, in order to obtain the steady-state solution for that value of λ. Then for $t \geq 40$, we let $\lambda(t)$ decrease exponentially so that it decreases by a factor of 10 for every time interval of 100. All other parameters of GWMFE remain exactly as before.

Shown in Fig. 7.2 is the solution (magnified 10× horizontally) at times corresponding to $\lambda = 10^{-3}, \ldots, 10^{-8}$. If the decay time interval is reduced from 100 to 50, the solution graphs are exactly the same (to within the width of the lines); thus we can be sure that we have varied $\lambda(t)$ sufficiently slowly that we really do have a succession of steady states. The computational statistics are NSTEP, CPU, NRES, NJAC = 487, 39.2 sec, 1672, 91. Approximately 1/4 of the work is required to reach the initial steady state at $t = 40$ with $\lambda = 10^{-3}$, and the remainder is to traverse the succession of steady states thereafter. Steady state in Fig. 7.1a corresponds to $\lambda = 10^{-4}$ here.

This 30-cell computation appears to be quite accurate, except that for $\lambda = 10^{-8}$, 10^{-7}, 10^{-6} there is clearly an inadequate number of nodes at the foot of the front. (Kuprat's "graph massage" would probably help greatly here.) Also shown in Fig. 7.2 is an overlaid comparison with a more accurate 60-cell computation. Except for the slight difficulty at the foot of the front, the two computations are in very good agreement.

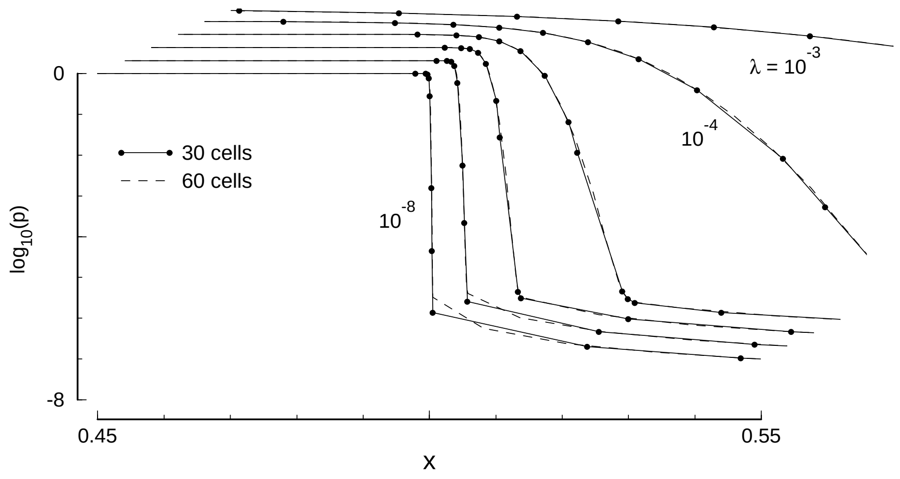

FIG. 7.2. *A family of steady states for the drift-diffusion equations by slowly varying* $\lambda(t)$*. Also, comparison of the* 30*-cell solution with a more accurate* 60*-cell solution.* 10× *horizontal enlargement. Successive graphs are offset.*

Here we have kept the boundary voltage constant and have varied λ. Clearly we could instead keep λ constant and vary the boundary voltage.

We point out the clear advantages of computing a smooth succession of steep-front steady states with moving nodes, as opposed to fixed-node adaptive methods. Here, as the steep fronts move and steepen smoothly; the nodes can merely move over smoothly without having to change their amplitudes greatly. With fixed-node adaptive methods the grid must be frequently re-adapted to follow the front movement and steepening, and with each new adaptive grid the amplitudes change drastically.

7.2. Conservation laws for gas dynamics. We consider the standard test problem of Sod [35] for the equations of gas dynamics in conservation law form,

$$u_t = -(v)_x + \nu u_{xx}, \tag{7.7a}$$

$$v_t = -(v^2/u + p)_x + \nu v_{xx}, \tag{7.7b}$$

$$w_t = -(v(w+p)/u)_x + \nu w_{xx}. \tag{7.7c}$$

Here u, v, and w are the densities of mass, momentum, and total energy of the gas as functions of x and t in a 1D shock tube, and p is the pressure. For an ideal gas $p = (\gamma - 1)(w - \frac{1}{2}v^2/u)$, where $\gamma = 5/3$ for a diatomic gas. Reflection boundary conditions, $u_x = 0$, $v = 0$, $w_x = 0$ are imposed at the ends of the tube, $0 \le x \le 1$. Initially the gas is at rest, and at a high density and pressure to the left of a diaphragm at $x = .5$. At $t = 0$ this diaphragm is burst, giving the piecewise constant initial conditions

$$(u, v, w) = \begin{cases} (1., 0, 2.5) & \text{on } 0 \le x < .5, \\ (.125, 0, .25) & \text{on } .5 < x \le 1. \end{cases} \tag{7.8}$$

Shown in Fig. 7.3 is the mass density for a 72-cell GWMFE solution at a sequence of times, using $\nu = 10^{-4}$, which is about as small a diffusion coefficient as we can handle for this problem. Successive graphs are offset by $(.04, .04)$ in (x, u). The solution immediately forms a rarefaction wave traveling to the left and a shock followed

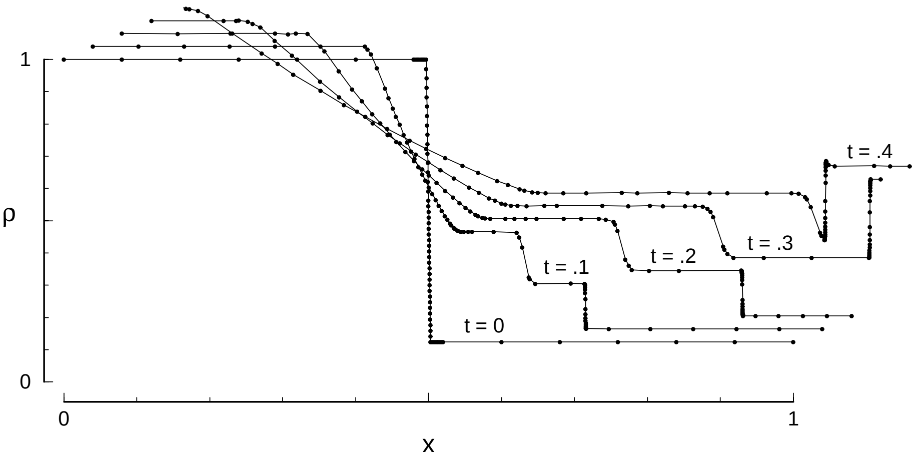

Fig. 7.3. *Gas dynamics problem with diffusion coefficient* $\nu = 10^{-4}$*, 72 cells, "*$10u$*" vertical rescaling,* $A^2 = (.5 \times \eta_1)^2 = .25 \times 10^{-8}$*. Shown is mass density with rarefaction wave, contact discontinuity, and shock. Shock hits and reflects from boundary at* $t \approx .288$.

by a contact discontinuity traveling to the right. The shock hits the boundary at $t \approx .288$ and by the time $t = .3$ shown it has already reflected from the boundary.

This is a problem which immediately develops structure out of nothingness at $t = 0$. Therefore we must replace the discontinuity of the idealized initial values by a thin ramp, and we must include enough nodes within and on the lips of this ramp to resolve the structure which immediately develops. Some who have tried MFE or GWMFE on this problem have made the mistake of placing none of the nodes within the ramp, thinking that GWMFE would somehow magically shift nodes to where they are needed. In fact, GWMFE is seeking a *manifold* solution to this problem and the vertical part of this manifold is where all the structure arises. Moreover, the initially zero momentum must immediately develop a sharp spike. Once the sharp corners of this structure have developed, the method tends to move the nodes with these corners, and the method is extremely disinclined to shift nodes around these corners from one part of the graph to another. Therefore, enough nodes must be in place *initially*, in approximately the right places, in order to resolve the structure which immediately develops; see Fig. 7.3. ("Graph massage" would probably help greatly here.)

Because of the small diffusion ($\nu = 10^{-4}$) we have used a "$10u$" vertical rescaling of this problem (as in section 6.3). The other parameters were chosen in standard fashion, with $\eta_1, \eta_2, \eta_3, \eta = 10^{-4}, 10^{-4}, 10^{-4}, 10^{-3}$ (the predictor error tolerances on the u_i, v_i, w_i, x_i), and $A^2 = (.5 \times 10^{-4})^2$, $(.5 \times 10^{-4})^2$, $(.5 \times 10^{-4})^2$ (the internodal viscosities on the u, v, w graphs). All PDE weights w_i were set equal to 1. The computational statistics to $t = .4$ are NSTEP, CPU, NRES, NJAC = 722, 202 sec, 2692, 151.

This solution appears to be extremely accurate. Shown, for example, in Fig. 7.4 is a comparison at $t = .4$ with a much more accurate solution using 144 cells and 10 times tighter predictor error tolerances. This is an enlargement of the region where the reflected shock meets the oncoming contact discontinuity. This comparison suggests that GWMFE computes very accurately not only the locations of the shocks, rarefactions, etc., but also the detailed viscous profile of the solution.

It should be admitted that most people attempting this problem don't *want* the detailed viscous profile for a finite ν. Instead they want the so-called "inviscid limit"

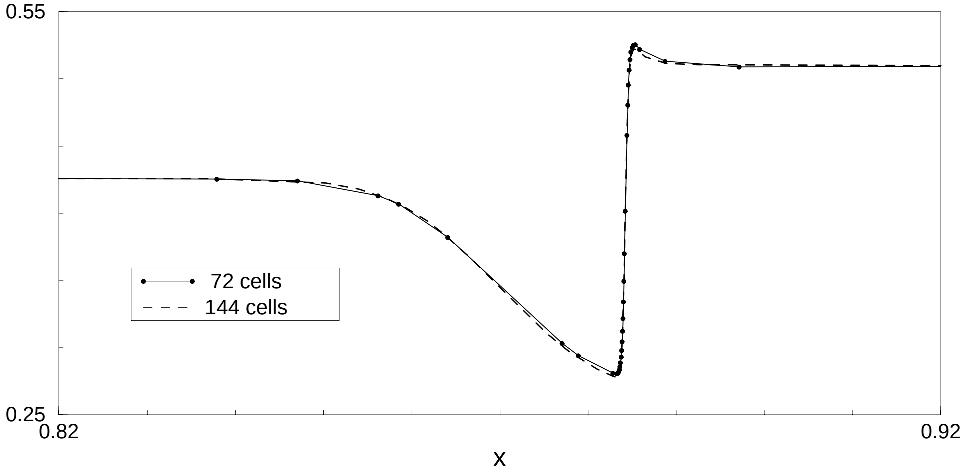

FIG. 7.4. *Enlargement (*$10\times$ *horizontal and* $3\times$ *vertical) of the reflected shock meeting the oncoming contact discontinuity at* $t = .4$ *in Fig.* 7.3*. Comparison of the* 72*-cell solution with a more accurate* 144*-cell solution.*

solution, the limiting solution as $\nu \to 0$, for which both the shocks and the contact discontinuities will be infinitely thin. We also point out that the very simple diffusion model in (7.7) is no way to attempt to compute very thin contact discontinuities. This is because (for the true PDE solution) the contact discontinuity width spreads at an $O(\sqrt{\nu t})$ rate, while the shock width is $O(\nu)$. Thus, in order to obtain a 10 times thinner contact discontinuity, ν must be taken 100 times smaller, resulting in a shock which is 100 times thinner. Since our shock with $\nu = 10^{-4}$ is already exceedingly thin, this approach to obtaining thinner contact discontinuities is a losing battle. Since many standard fixed-node finite difference methods have numerical diffusion which mimics the simple diffusion model in (7.7), this also explains why those methods have such great difficulty in computing both contact discontinuities and shocks. A more physically realistic second-order diffusion model for (7.7) would involve viscosity, heat conduction, and mass diffusion.

8. Failure of GWMFE for certain steady-state convection problems.

For certain problems the "internodal tensions and pressures," discussed at the end of section 2, are not sufficient to prevent a disastrous drift of nodes away from where they are needed. Consider, for example, the following problem:

$$\begin{aligned} &u_t = -u_x + b(x) + \nu u_{xx} \text{ on } -4 \le x \le 4, \\ &u(-4,t) = 0, \quad u_x(4,t) = 0, \quad u(x,0) = u_0(x), \end{aligned} \tag{8.1}$$

where we take $b(x) = .1x\exp(-x^8)$ and a tiny $\nu = 10^{-3}$ diffusion coefficient. This PDE is of the form discussed in (2.15)–(2.18). The applied forces $K(u)\mathbf{n}$ on the 1D solution manifold are, except for the tiny diffusion term, those that would be given by a "solar wind" with momentum flux of $(1, b)$. We have purposely chosen a fairly nonsmooth $b(x)$ which cannot be adequately resolved by a coarse (40-cell) uniform grid; our $b(x)$ is ≈ 0 for $|x| > 1$, $\approx .1x$ for $|x| < 1$, with sharp transitions at $x = \pm 1$.

The exact solution evolves to a steady state but has initial transients which grow and are then convected downstream and are swept out at the right-hand boundary. It is the reliable computation, through the transient phase and out to steady state,

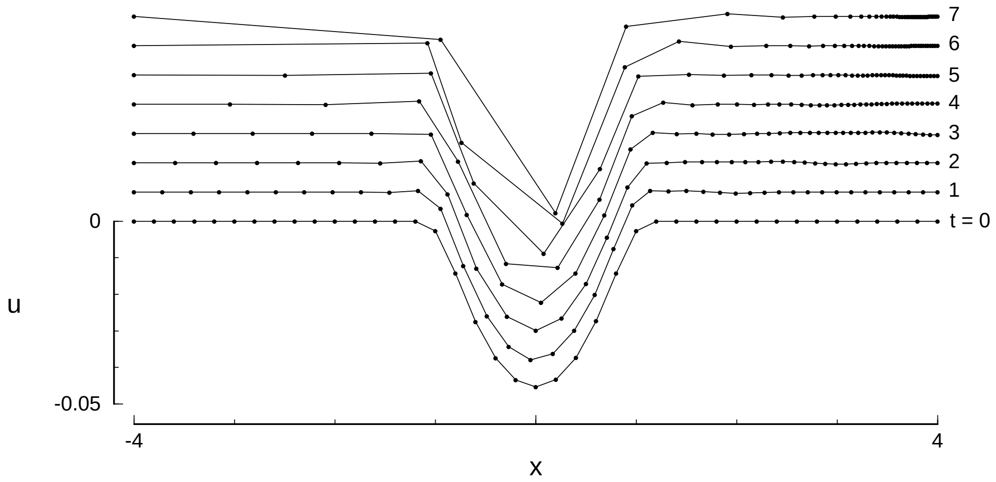

FIG. 8.1. *GWMFE solution at times $t = 0, 1, 2, \ldots, 7$ of a simple convection equation, starting from the essentially exact steady-state solution of the PDE. Successive graphs are offset vertically. GWMFE nodes are swept downstream and GWMFE does not achieve a steady state.*

of such nonsmooth problems which interests us here. Notice that a steady-state solution is one whose graph remains everywhere tangent to the solar wind vector field $(1, b)$. Unfortunately, with only 40 cells both GWMFE and uniform-grid fixed-node FE perform extremely poorly on this nonsmooth problem.

Consider now the transient GWMFE computation for such a problem, but starting instead with the easier situation of initial values $u_0(x)$ that are close to the exact steady-state solution of the PDE. We would *hope* that GWMFE would evolve to its own steady state, with its now stationary nodes well concentrated to resolve the corners of the exact solution. Unfortunately, this is not the case; GWMFE (without adequate regularizing "spring forces" to stabilize the tangential movement of its nodes) does *not* evolve to a steady state on this problem for the following reason. Because the GWMFE graph is piecewise linear it can never be exactly parallel everywhere to the solar wind; as a result there will always be slight residual forces (proportional to $\sin^2\theta$, where θ is the angle between the wind and the graph) on the nodes which will tend to sweep them downstream.

The internodal viscosity forces of (2.21) can slow this flow of nodes downstream, but they cannot stop it. In fact, a very large coefficient A^2 slows the nodes, leading to larger oscillations (as discussed in the boundary-layer problem of section 6.1), and these larger oscillatory angles then lead to larger forces tending to sweep the nodes downstream.

Shown in Fig. 8.1 is the 40-cell GWMFE solution for this problem. We use the u_i, x_i predictor error tolerances of $\eta_1, \eta = 10^{-4}, 10^{-3}$; a rather large viscosity coefficient of $A^2 = (10 \times \eta_1)^2 = 10^{-6}$; and zero internodal tensions. The initial value function $u_0(x)$ is obtained from the essentially exact steady state of a 120-cell fixed-node FE computation. We see that all the nodes are gradually swept downstream, leading to total disaster by the time $t = 4$. Other computations with a wide range of A^2 values show that this downstream flight of nodes is somewhat impeded by large values of A^2 (such as our $A^2 = 10^{-6}$ here); however, a large A^2 leads to oscillations on the two corners, such as is seen here at $t = 2, 3$, and 4.

Introduction of sufficiently strong internodal tensions does lead to a steady-state GWMFE solution in this problem. However, the final positioning of the nodes exhibits no decent adaptive tendency near the corners and the resulting solutions are quite unsatisfactory. We have tried linear, quadratic, and cubic internodal tensions.

Somewhat better stabilizing forces are obtained by *tethering* of the nodes. This was introduced in [8]. Each node is tied by an elastic (linear or cubic) tether to a fixed base point $b(j)$. For the present problem, however, tethering gives only limited success. It does stop the downstream drift of the nodes and gives a steady state, but it does *not* concentrate nodes into the corners of the solution.

Application of Kuprat's graph-massage routines to this problem would probably produce good solutions, but at considerable cost. New nodes would be created to bisect the overly large cells to the left of $x = -1$, and in the sharp corners as needed for good resolution, and would be removed as they accumulate in the flat region to the right of $x = +1$. However, the nodes would still drift downstream from the corners and new nodes would have to be continually created. Thus, although we might compute a very accurate "steady state," it would be a "dynamic steady state" rather than a true steady state with stationary adaptive nodes.

In conclusion, we admit failure on this class of problems for the GWMFE method as it presently stands. However, the second author has some ideas which seem promising in 1D trials.

9. Our nonlinear "Krylov subspace" accelerator. We now explain the accelerator for our modified Newton's method. In sections 5.1–5.3 we explained the nature of the diagonally preconditioned implicit BCE equations (5.7) which must be solved at each time step. These are of the form

$$\mathcal{R}(y) = 0. \tag{9.1}$$

As pointed out in section 5.1, our predictor-error control strategy ensures that the initial guess $y_0 \equiv y^*_{n+1}$ for our Newton's method will always be extremely close to the desired converged value $y \equiv y_{n+1}$ (otherwise we would decrease the time step). Moreover, the diagonal preconditioning of section 5.3 helps to ensure that the preconditioned equations (5.7) are better conditioned and more nearly linear than the original equations (5.3).

As in the usual "modified Newton" method we first precondition equation (9.1) by premultiplying by $-J^{-1}$, where $J \equiv \mathcal{R}'(\bar{\bar{y}})$ is the Jacobian evaluated at some *old* value $\bar{\bar{y}}$ (and where the *LU* decomposition of J is computed and saved since we are using *direct* methods to solve linear equations in the present 1D and 2D codes). This yields the equations

$$f(y) \equiv -J^{-1}\mathcal{R}(y) = 0, \tag{9.2}$$

where $f'(y)$ would be approximately $-I$ near the root *if* J had been freshly updated.

For the purpose of designing a nonlinear iteration, we *pretend* that the following two assumptions hold:

$$f(x + z) = f(x) - Az \tag{9.3}$$

for all x near the desired root y and all tiny corrections z, where A is an unknown constant nonsingular matrix; and, moreover,

$$Az \approx Iz \tag{9.4}$$

for any small correction vectors z for which we have no better information.

Thus we begin with the good initial guess y_0, we have the initial residual $r_0 \equiv f(y_0)$, and our desired correction v should solve the *linear* residual correction equation

$$f(y_0 + v) \equiv r_0 - Av = 0. \tag{9.5}$$

The standard "modified Newton" method makes constant use of the assumption (9.4) that $A \approx I$. This yields the standard fixed-point iteration

$$y_{k+1} \equiv y_k + f(y_k) \approx r_0 + (I - A)y_k, \tag{9.6}$$

which converges only if the spectral radius of $(I - A)$ is less than 1.

Our method instead proceeds with successive iterates

$$y_1 \equiv y_0 + v_1, \ \ldots, \ y_k \equiv y_{k-1} + v_k, \tag{9.7}$$

and after each step we are able to evaluate $Av_k = f(y_{k-1}) - f(y_k)$ by differencing. Thus, after the kth step we will have accumulated the correction vectors $\{v_1, \ldots, v_k\}$, their A values $\{Av_1, \ldots, Av_k\}$, and the kth residual $r_k \equiv f(y_k)$.

Our desired correction v at this point should solve the residual correction equation

$$f(y_k + v) = r_k - Av = 0. \tag{9.8}$$

We make an initial pass at approximately solving (9.8) by letting w_{k+1} be that element of the subspace $V_k \equiv \text{span}\{v_1, \ldots, v_k\}$ such that Aw_{k+1} best fits r_k in the l_2 norm. (This is an easy linear least squares problem involving the $k \times k$ Grammian matrix of the (Av_i, Av_j) inner products and also the (r_k, Av_j) inner products.)

Then, at the point $y^*_{k+1} \equiv y_k + w_{k+1}$ we would have the residual $q_{k+1} \equiv r_k - Aw_{k+1}$ (which is orthogonal to $A(V_k)$) and we would need to make the further correction z satisfying

$$f(y^*_{k+1} + z) \equiv q_{k+1} - Az = 0. \tag{9.9}$$

Now, having used up all available information about exact values of A on certain subspaces, we have to invoke the dubious assumption (9.4), yielding the approximate solution $z = q_{k+1}$ for (9.9). Our combined correction v for (9.8) is thus

$$v_{k+1} \equiv w_{k+1} + q_{k+1}. \tag{9.10}$$

Of course, at the first step ($k = 0$) the subspace V_0 is empty and hence $w_1 = 0$ and the first correction vector is just $v_1 = r_0$, as in the "modified Newton" method.

Under the linearity hypothesis (9.3), plus an additional hypothesis such as that A is positive definite, one can easily show that the subspace $V_k \equiv \text{span}\{v_1, \ldots, v_k\}$ coincides with the Krylov subspace $K_k \equiv \text{span}\{r_0, Ar_0, \ldots, A^{k-1}r_0\}$.

This accelerator is essentially a nonlinear version of the GMRES (generalized minimum residual) method. It is somewhat similar to the nonlinear hybrid Krylov methods of Gear and Saad [36] and of Brown and Saad [37], and in fact we wish to thank Saad for a fruitful conversation and for making one of his codes available to us. That code, however, had many nonlinear fail-safe features and was somewhat profligate in its evaluations of the very expensive f. The present Krylov accelerator, designed by Miller and implemented by Carlson, instead requires only one evaluation of f at each iteration. Other nonlinear Krylov methods for stiff ODEs can be found in Brown and Hindmarsh [38].

This accelerator was incorporated into our codes in 1990. After a bit of testing on various problems we fixed upon one standard set of parameters for the method which we have used on *all* 1D and 2D computations since. See our report [13] for further details on this accelerator and its parameters.

Note that if the Jacobian J has been freshly evaluated, then the assumption $A \approx I$ of (9.4) is quite valid. However, that assumption is not valid if J is old and if $\Delta\tau$ has changed by a considerable factor since J was last evaluated; see (5.5) for the dependence of J on $\Delta\tau$. Also, the example of section 6.5, nodes crashing into shocks, gives an example where the basic structure of the Jacobian probably changes drastically (over several time steps) only at a few colliding nodes; elsewhere it probably remains largely unchanged. Thus, this is an example where the A probably differs drastically from I only on a subspace of small dimension. Krylov subspace methods have a tendency to handle such small-rank corrections very efficiently.

This accelerator has been extremely robust and reliable in all our 1D and 2D computations since 1990. Its principal advantages were listed in section 5.4. In particular, our experience is that it cuts by a factor of 2 or 3 the number of evaluations of J (and its LU decomposition) which need to be done for a particular problem (which gives a great savings in two dimensions). It therefore ameliorates one of the principle problems that Zegeling and Blom [18] found with the GWMFE method, which is that it appeared to require an excessive number of Jacobian evaluations.

REFERENCES

[1] K. Miller and R. Miller, *Moving finite elements, Part* I, SIAM J. Numer. Anal., 18 (1981), pp. 1019–1032.

[2] K. Miller, *Moving finite elements, Part* II, SIAM J. Numer. Anal., 18 (1981), pp. 1033–1057.

[3] R. Alexander, P. Manselli, and K. Miller, *Moving finite elements for the Stefan problem in two dimensions*, Rend. Accad. Naz. Lincei (Rome), LXVII (1979), pp. 57–61.

[4] R. Gelinas, S. Doss, and K. Miller, *The moving finite element method: Applications to general equations with multiple large gradients*, J. Comput. Phys., 40 (1981), pp. 202–249.

[5] M. J. Djomehri and K. Miller, *A Moving Finite Element Code for General Systems of PDE's in* 2*-D*, Center for Pure Appl. Math. Report PAM-57, University of California, Berkeley, CA, 1981.

[6] M. J. Djomehri, *Moving Finite Element Solution of Systems of Partial Differential Equations in One Dimension*, Ph.D. thesis, Department of Mathematics, University of California, Berkeley, CA, 1982; also Center for Pure Appl. Math. Report PAM-125, University of California, Berkeley, CA, 1983.

[7] M. J. Djomehri, S. Doss, R. Gelinas, and K. Miller, *Applications of the Moving Finite Element Method for Systems in* 2*-D*, Center for Pure Appl. Math. Report PAM-427, University of California, Berkeley, CA, 1988.

[8] K. Miller, *Alternate Modes to Control the Nodes in the Moving Finite Element Method*, in Adaptive Computational Methods for Partial Differential Equations, SIAM, Philadelphia, PA, 1983, pp. 165–182.

[9] K. Miller, *Recent results on finite element methods with moving nodes*, in Accuracy Estimates and Adaptive Refinement in Finite Element Calculations, I. Babuška, O. C. Zienkiewicz, J. Gago, and E. R. de A. Oliveira, eds., Wiley, New York, 1986, pp. 325–338.

[10] N. N. Carlson and K. Miller, *Gradient weighted moving finite elements in two dimensions*, in Finite Elements Theory and Application, D. L. Dwoyer, M. Y. Hussaini, and R. G. Voight, eds., Springer-Verlag, New York, 1988, pp. 151–164.

[11] K. Miller, *On the mass matrix spectrum bounds of Wathen and the local moving finite elements of Baines*, SIAM J. Numer. Anal., 29 (1992), pp. 89–106.

[12] K. Miller, *A geometrical-mechanical interpretation of gradient-weighted moving finite elements*, SIAM J. Numer. Anal., 34 (1997), pp. 67–90.

[13] N. N. Carlson and K. Miller, *Design and Application of a Gradient-Weighted Moving Finite Code, Part* I, *in* 1*-D*, Technical Report 236, Center for Applied Mathematics, Purdue University, West Lafayette, IN, 1994 (this report is an earlier unabridged version of the present paper).

[14] N. N. Carlson and K. Miller, *Design and application of a gradient-weighted moving finite code* II: *in two dimensions*, SIAM J. Sci. Comput., 19 (1998), pp. 766–798.

[15] A. Kuprat, *Creation and Annihilation of Nodes for the Moving Finite Element Method*, Ph.D. thesis, Department of Mathematics, University of California, Berkeley, CA, 1992.

[16] S. Nazari, *Rotational Surfaces in Euclidean and Hyperbolic Spaces, Mean Curvature Motion, and the Moving Finite Element Method*, Ph.D. thesis, Department of Mathematics, University of California, Berkeley, CA, 1993.

[17] R. M. Furzeland, J. G. Verwer, and P. A. Zegeling, *A numerical study of three moving grid methods for one-dimensional partial differential equations which are based on the method of lines*, J. Comput. Phys., 89 (1990), pp. 349–388.

[18] P. A. Zegeling and J. G. Blom, *An evaluation of the gradient-weighted moving-finite-element method in one space dimension*, J. Comput. Phys., 103 (1992), pp. 422–441.

[19] P. A. Zegeling, *Moving-Grid Methods for Time-Dependent Partial Differential Equations*, Ph.D. thesis, University of Amsterdam and Centre for Mathematics and Computer Science (CWI), Amsterdam, 1992. Several of the chapters appear separately as [17] and [18].

[20] A. Glasser, *A moving finite element model of the high density Z-pinch*, J. Comput. Phys., 85 (1989), pp. 159–209.

[21] I. Lie and J. O. Langseth, *A Moving Finite Element Method with Solution-Dependent Norm, and the ODE-Aspects of the MFE Equations*, preprint, NRDE, Div. for Electronics, Kjellev, Norway, 1988.

[22] A. J. Wathen, *Moving Finite Elements and Oil Reservoir Modeling*, Ph.D. thesis, Department of Mathematics, University of Reading, UK, 1984.

[23] A. J. Wathen and M. J. Baines, *On the structure of the moving finite element equations*, IMA J. Numer. Anal., 5 (1985), pp. 161–182.

[24] A. J. Wathen, *Mesh-independent spectra in the moving finite element equations*, SIAM J. Numer. Anal., 23 (1986), pp. 797–814.

[25] A. J. Wathen, *Realistic eigenvalue bounds for the Galerkin mass matrix*, IMA J. Numer. Anal., 7 (1987), pp. 449–457.

[26] M. J. Baines, *Locally adaptive moving finite elements*, in Numerical Methods for Fluid Dynamics II, K. W. Morton and M. J. Baines, eds., Oxford University Press, UK, 1986, pp. 1–14.

[27] M. J. Baines and A. J. Wathen, *Moving finite element modelling of compressible flow*, Appl. Numer. Math., 2 (1986), pp. 495–514.

[28] M. J. Baines and A. J. Wathen, *Moving finite element methods for evolution problems,* I, *theory*, J. Comput. Phys., 79 (1988), pp. 245–269.

[29] I. W. Johnson, A. Wathen, and M. J. Baines, *Moving finite element methods for evolutionary problems,* II, *applications*, J. Comput. Phys., 79 (1988), pp. 270–297.

[30] M. J. Baines, *An analysis of the moving finite element procedure*, SIAM J. Numer. Anal., 28 (1991), pp. 1323–1349.

[31] P. K. Jimack, *On steady and large time solutions of the semi-discrete moving finite element equations for* 1*-D diffusion problems*, IMA J. Numer. Anal., 12 (1992), pp. 545–564.

[32] M. J. Baines, *Moving Finite Elements*, Oxford University Press, UK, 1994.

[33] A. C. Mueller and G. F. Carey, *Continuously deforming finite elements*, Internat. J. Numer. Methods Engrg., 21 (1985), pp. 2099–2126.

[34] P. A. Markowich, *Semiconductor Equations*, Springer-Verlag, New York, 1990.

[35] G. A. Sod, *A survey of several finite difference methods for systems of hyperbolic conservation laws*, J. Comput. Phys., 27 (1978), pp. 1–31.

[36] W. C. Gear and Y. Saad, *Iterative solution of linear equations in ODE codes*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 583–601.

[37] P. N. Brown and Y. Saad, *Hybrid Krylov methods for nonlinear systems of equations*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 450–481.

[38] P. N. Brown and A. C. Hindmarsh, *Matrix-free methods for stiff systems of ODEs*, SIAM J. Numer. Anal., 24 (1987), pp. 610–638.