JOURNAL OF COMPUTATIONAL PHYSICS 95, 254–302 (1991)

Review Article

Review of Some Adaptive Node-Movement Techniques in Finite-Element and Finite-Difference Solutions of Partial Differential Equations

D. F. HAWKEN, J. J. GOTTLIEB, AND J. S. HANSEN

Institute for Aerospace Studies, University of Toronto, 4925 Dufferin Street, Downsview, Ontario, Canada, M3H 5T6

Received November 28, 1988; revised May 16, 1990

This article summarizes the results of a literature search for adaptive numerical methods of solving partial differential equations; the methods discussed involve the adaptive movement of nodes, so as to obtain a low level of solution truncation error while minimizing the number of nodes used in the calculation. Such methods are applicable to the solution of nonstationary flow problems that contain moving regions of rapid change in the flow variables, surrounded by regions of relatively smooth variation. Flows with shock waves, contact surfaces, slip streams, phase-change interfaces, and boundary layers can be modelled with great precision by these methods. It will be shown that significant economies of execution can be attained if nodes are moved so that they remain concentrated in regions of rapid variation of the flow variables. © 1991 Academic Press, Inc.

1. ADVANTAGES OF USING MOVING-NODE ADAPTIVE METHODS

Two of the most widely used methods of discretizing partial differential equations (PDEs) are finite elements and finite differences. Finite-element and finite-difference methods that use uniformly spaced nodes often waste computational effort because, in order to obtain acceptable truncation errors in regions of large solution variation, much smaller node separations than are necessary in regions of negligible solution variation must be used. In the case of flows containing moving shocks, contact surfaces, and slip streams, only a very small portion of the domain requires small node separations; thus significant economies can be obtained by moving the nodes so that they remain concentrated about areas of large solution variation.

In order to properly resolve a shock wave the node separation in the vicinity must be several times smaller than the shock thickness. The shock thickness is related to the value of the coefficients of the second-order spatial-derivative (or diffusion) terms in the PDEs, often being of the same order of magnitude as these coefficients. Hence, as the coefficients of the second-order terms are reduced, the shock thickness is reduced also. Nonzero coefficients are required to prevent

0021-9991/91 $3.00

infinitely steep, and thus aphysical, shock fronts and allow a nonzero node separation within a properly modelled shock. For commonly modelled physical scales and gas viscosities, physically accurate coefficients are usually extremely small and are in practice replaced by larger coefficients to allow larger node separations to be used. The use of adaptive methods allows much smaller node separations, and hence much smaller (more physically realistic) coefficients and much thinner shock waves, than can in general be accommodated by nonadaptive methods. The ability to use smaller coefficients in the second-order terms also results in a more realistic growth of the thickness of contact surfaces with time. It should be noted that much larger time-steps can be made when the nodes move with a shock wave than can be obtained with a fixed non-uniform grid; time derivatives evaluated in the moving frame tend to be constant or zero.

The solution of the gasdynamics equations in inviscid form will always result in large truncation errors in the vicinity of a shock. Thus, for all true finite-element or finite-difference algorithms, either the algorithm is applied to the viscid form of the gasdynamics equations or some form of artificial or numerical viscosity is used as part of the algorithm. There are a few algorithms, such as the random-choice method invented by Glimm [1], that solve a Riemann problem as part of the solution process; these algorithms can be used to solve the gasdynamics equations in inviscid form without the use of artificial or numerical viscosity. Even though the domain is broken up with nodes, as in a true finite-difference method, the Riemann problem solution technique is in essence an analytical procedure. Such methods allow quite large node separations without any oscillations near a shock; however, if modelling of the interior of the shock transition is desired, then the viscid gasdynamics equations must be solved and much smaller node separations employed. Again, adaptive methods can be employed to advantage.

2. Review of Adaptive Methods

2.1. *Introduction*

This review focuses on numerical PDE solution methods that move nodes so that they remain concentrated in regions of rapid variation of the solution. A contrasting group of methods, which adapt to the solution by adding new fixed nodes in regions of rapid change in the solution, will not be discussed because such methods are of limited utility in solving problems where the locations of the regions of rapid variation move with time; discussions of such methods, along with overviews of the entire field of grid generation, can be found in Thompson, Warsi, and Mastin [2, 3], Thompson [4], and Turkel [5]. Body-fitted coordinate systems are also discussed by these authors and are not covered in this review since emphasis will be placed on methods that move the nodes so as to adapt to solution properties rather than boundary properties. More specialized reviews of adaptive methods are given in Anderson [6], Thompson [7], and Eiseman [8].

Most of the methods discussed in this review require that the PDEs have continuous solutions. If the PDEs are nonlinear, as is usually the case, they will tend to develop discontinuous solutions unless they contain viscosity or analogous terms. Hence, most of the methods discussed will run into numerical problems unless viscous terms are included within the PDEs.

As pointed out by Thompson [4], an adaptive-node method must have several ingredients:

an orderly method of numbering (or mapping) nodes distributed over the physical region of interest;

a means of "communicating" between nodes so that the distribution of nodes remains fairly regular as they are shifted;

a means of representing the continuous solutions discretely and a means of evaluating the discrete values with sufficient accuracy;

a measure of the error in the discrete values that bears some relation to the truncation error;

a means of redistributing the nodes as indicated by the measure, so as to reduce solution error.

Each of these ingredients is discussed in more detail in the text below.

The numerical solution of a simple, nonlinear, ordinary differential equation (ODE) will be discussed in order to illustrate some of the concepts involved in the literature. Consider the equation

$$\mu \frac{d^2A}{dX^2} + A\left[\frac{dA}{dX} - 1\right] = 0 \tag{2.1}$$

on the domain $0 \leqslant X \leqslant 1$, where μ is a small positive number and $A(0) = \alpha$ and $A(1) = \beta$ are the boundary conditions. It is easily seen that if $dA/dX = 1$, then the slope will tend to remain constant at unity, since the second derivative is then zero. However, except for very special values of α and β, the slope must eventually depart from unity. For arbitrary values of α and β, the solution will consist of one or two regions with $dA/dX = 1$ and a region in which the solution varies rapidly so as to match the boundary conditions. The region of rapid variation will take the form of a boundary layer or a more centrally located stationary shock layer. The location of the layer can be predicted only by the use of extensive analysis. As μ is reduced in size, the thickness of the layer is reduced proportionately and the values of the first and second derivatives within the layer become correspondingly large.

Figure 2.1 illustrates the results of an unsuccessful attempt to solve Eq. (2.1) with $\alpha = 4$ and $\beta = 2$, using a nonadaptive finite-element method with $N = 171$ fixed, equally spaced nodes. "A" was approximated using piecewise-linear basis functions, and the method of Galerkin was applied to Eq. (2.1). The value of μ was initially set to 0.1, and a linear ramp with $A^1 = \alpha$ and $A^N = \beta$ was used as an initial guess for the solution. The nodal values A^k were adjusted so as to minimize the Galerkin residual by using an iteration employing Newton's method. The value of μ was

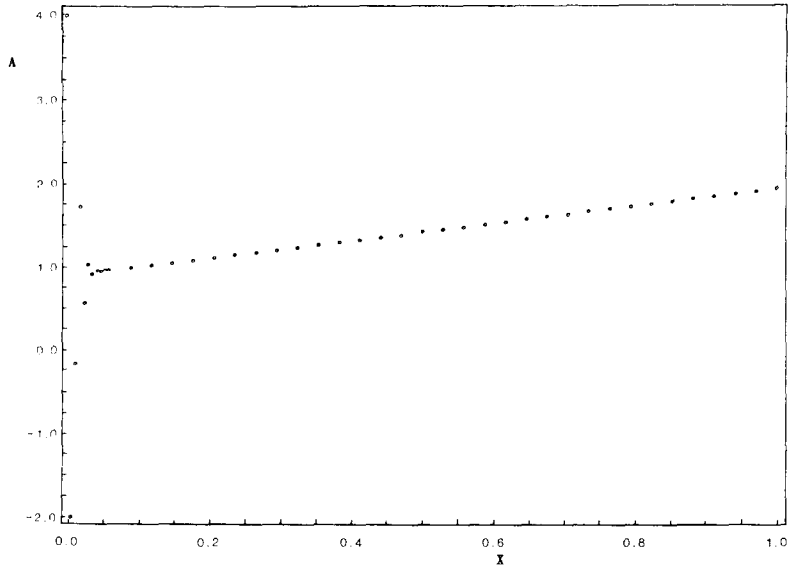

FIG. 2.1. Nonadaptive solution with 171 nodes and $\mu = 0.0015$; showing Gibbs' oscillations. Only every fifth node is illustrated in the straight-line portion of the graph.

reduced whenever the residual fell below a preset threshold. The region of rapid variation of the solution is of the order of μ; as the value of μ was reduced, there were eventually not enough nodes in this region to properly represent the solution, and hence a large truncation error was produced. Thus, even though 171 nodes were used, the solution with $\mu = 0.0015$ shown in Fig. 2.1 exhibits Gibb's oscillations.

One may apply Taylor's theorem to the discrete representation of an ODE in order to recover the original ODE. The higher-order terms in the Taylor series, which were not in the original ODE, represent the truncation error in the numerical solution. The nodes should be spaced close enough that these extra terms are negligible compared to the terms in the original ODE; otherwise one will obtain the numerical solution of an ODE with extra terms in it. Such considerations are especially important in the solution of the differential equations that occur in gas-dynamics. These equations possess second-order spatial derivatives with very small coefficients; the second-order terms can easily be swamped by the truncation error in first-order spatial derivatives contained in the differential equations. The proper solution to such differential equations often contains very rapid transitions in the form of shock waves or contact surfaces. If odd-order derivatives predominate in the truncation error (known as numerical dispersion), the extra terms can cause the new ODE to have an oscillatory behaviour (Gibbs' oscillations) that the proper ODE did not possess, in the vicinity of the transition. If even-order derivatives predominate in the truncation error (known as numerical viscosity or numerical diffusion), excessively thick transitions can result. Many nonadaptive methods add

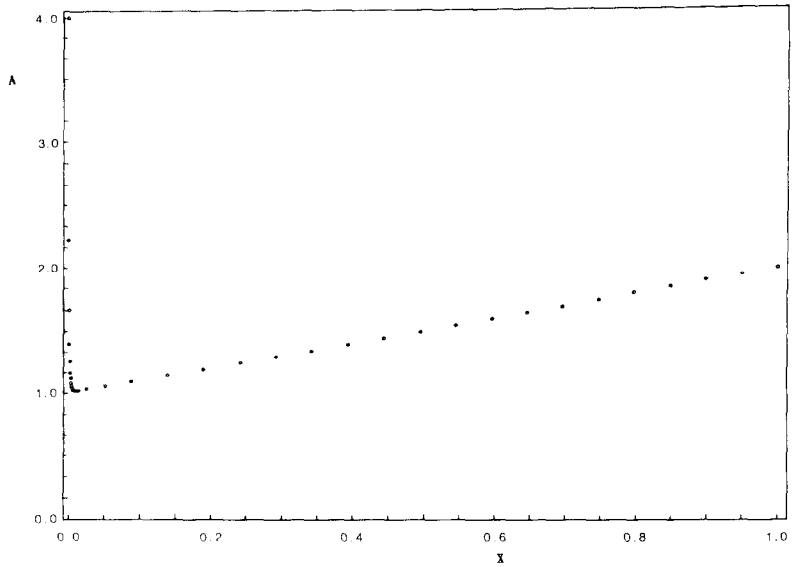

FIG. 2.2. Adaptive solution with 41 nodes and $\mu = 0.0015$.

an even-order artificial-viscosity term in regions of rapid solution variation so as to dominate odd-order truncation-error terms, and hence avoid Gibbs' oscillations, but this also results in a physically invalid broadening of transition thicknesses.

If an adaptive method is used, then nodes can be moved into regions of rapid solution variation so as to minimize the truncation error. Using an adaptive finite-

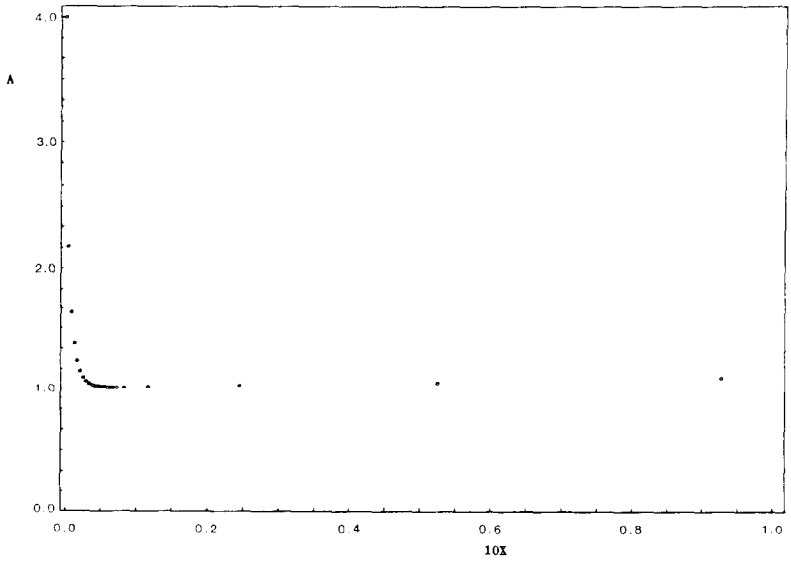

FIG. 2.3. Adaptive solution with 41 nodes and $\mu = 0.001$; high-resolution plot of boundary-layer region.

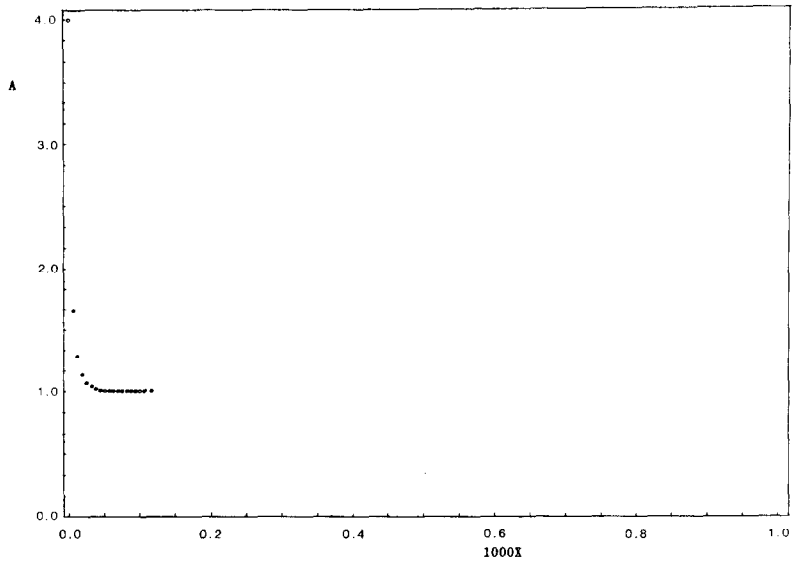

FIG. 2.4. Adaptive solution with 41 nodes and $\mu = 0.00001$; high-resolution plot of boundary-layer region.

element technique, Eq. (2.1) was solved with the same boundary conditions as above, but employing only 41 nodes. Figures 2.2, 2.3, and 2.4 illustrate solutions for $\mu = 0.0015$, $\mu = 10^{-3}$, and $\mu = 10^{-5}$, respectively. In order to adequately resolve the behaviour of the solution, only the region of the boundary layer is plotted in the last two figures. Note the complete absence of Gibbs' oscillations for values of μ more than 100 times smaller than the value that caused problems in the nonadaptive solution. Analysis of the Newton's method matrix used in the nonadaptive solution indicated that the matrix was always diagonally dominant in regions where the slope was approximately one, but less and less diagonally dominant as the slope varied from one, unless node separation was decreased in the region of rapid solution variation. Nonsingularity of the matrices employed in a method is required to obtain a unique solution. Reducing the node separation in regions of large solution variation also minimizes the truncation error. An adaptive algorithm was devised in which the node separation was gradually increased in regions of near-unit slope, with the remaining nodes being equidistributed in the boundary or shock layer. The methods described in the following text all try to obtain this redistribution of nodes, with a minimal computational cost, so as to maximize solution accuracy.

2.2. *Overview of Adaptive Methods*

Many adaptive methods transform the PDEs for the physical variables (denoted by vector **A**) from physical coordinates (herein written (X_i, t)) for an i-dimensional physical space to i-dimensional computational coordinates (herein written (ξ_i, t)) in which the nodes are equally spaced. An excellent account of such transformations

can be found in Thompson [9]. Integral values of the computational coordinates often correspond to the grid numbering; in such a case these coordinates are denoted by the phrase "numerical coordinates." In some one-dimensional methods, there is an auxiliary transformation to slope coordinates that can be defined by $dS = [1 + \|d\mathbf{A}/dX\|^2]^{1/2}\, dX$.

$\mathbf{A}(\xi_i, t)$ and $X_j(\xi_i, t)$ are the unknowns in the computational coordinate system. The transformed PDEs contain additional terms of the form $(\partial X_i/\partial t)(\partial \mathbf{A}/\partial X_i)$, with the derivatives taken at a fixed computational coordinate. Each PDE for $\mathbf{A}$ is reduced to a set of ordinary differential equations in time (one ODE for each node) by application of a discretization method such as the finite-element or finite-difference formulations. The ODEs can be written in the form

$$\frac{d\mathbf{A}^k}{dt} = \mathbf{F}^k(t). \tag{2.2}$$

Here, $\mathbf{A}^k$ is the value of the solution at node k, $d\mathbf{A}^k/dt$ is its time derivative in the computational coordinate frame, and the details of the computation of $\mathbf{F}^k$ are determined by the particular discretizing method used to reduce the PDEs to ODEs. $\mathbf{F}^k$ also contains, in discretized form, the additional terms in $(\partial X_i/\partial t)(\partial \mathbf{A}/\partial X_i)$ that appear in the computational coordinate frame. Particular attention must be paid to the original placement of the nodes, so that spatial truncation error will not cause inaccurate initial estimates of $d\mathbf{A}^k/dt$.

The ODEs can be solved employing a time-stepping finite-difference method. The method may be time-explicit, in which case $\mathbf{F}^k(t)$ is evaluated at the current time level, or may be time-implicit, in which case the value of $\mathbf{F}^k(t)$ at an advanced time level is predicted as part of the solution algorithm. In general, implicit methods permit much larger time-steps than explicit methods without causing instabilities, but they require the time-consuming solution of matrix equations. As nodes are allowed to approach one another more closely, an implicit method eventually becomes essential.

Given the time derivative of $\mathbf{A}^k$ at each node, one can compute the value of $\mathbf{A}^k$ at the next time-step. Because $d\mathbf{A}^k/dt$ can vary markedly from node to node, causing growth and decay of the solution in different sections of the domain, the system of ODEs is often "stiff," as discussed by Gelinas [10]. A system is stiff if explicit methods of numerical solution must use very small values of time-step in order to maintain solution stability, even though accuracy limits may allow much larger time-steps; the root cause of the stiffness is that the numerical solution of the system contains decaying components that experience error growth if the time-step size is not limited severely. The drastic difference in ODE-solution time-constant from node to node causes some of the stiffness. A closely related cause for the stiffness is the fact that nodes approach one another very closely in regions of large solution variation, thus causing CFL limitation (first explained by Courant, Friedrichs, and Lewy [11]) of time-step if an explicit method is used. In order to obtain reasonably large time-steps, the system of ODEs must usually be solved by

implicit methods such as the backward-differentiation algorithm of Gear [12]. A version of the Gear method has been implemented by Hindmarsh [13], and this in modified form is available as an International Mathematical and Statistical Library (IMSL) subroutine. Enright, Hull, and Lindberg [14] discuss and compare a number of stiff-ODE solution methods. Some implicit methods applicable to PDEs that can be written in specific forms are discussed in Anderson, Tannehill, and Pletcher [15]. Kee, Petzold, Smooke, and Grcar [16] present an excellent discussion on the use of implicit methods in the context of adaptive solution of PDEs.

A measure of the spatial truncation error is computed at intervals in the time-stepping solution of the PDEs. The measure is usually some grid-derivative function of X_i and/or $\mathbf{A}$ in the computational-coordinate system. The derivatives are usually approximated by finite differences, and hence higher-order derivative estimates are highly susceptible to grid irregularities. Often some measure smoothing is attempted to reduce the effect of grid irregularities. Types of smoothing range from simple node-to-node derivative averaging [45–48, 105–113] to the creation of long-range internodal pseudoforces [90–92, 99–104]. Temporal smoothing has been applied in some methods [102–104]. In addition to taking error measures, some workers use measures of transformation smoothness [44–48, 50, 51, 55, 59–63], orthogonality [45–48, 50, 51, 59–63, 110–113], or other transformation property [28, 42, 43, 52–54, 64–87, 110–113] to control the quality of the node distribution.

A bewildering variety of error measures is used in the literature, including change in the solution, or its derivatives of various orders, from node to node [23, 24, 28, 29, 31–39, 55, 87, 90–96, 99–109]; circular curvatures or torsions along lines of constant curvilinear coordinate [30, 44, 96, 110–113]; solution slope lengths [28, 29, 31, 33–39, 41, 44, 55, 104]; solution-property weighted cell volumes [59, 110–113]; and complex combinations of the above [44, 96, 99–103, 110–113]. In some cases the error measure is related to truncation-error expressions [25, 26, 28, 29, 57, 64–87, 96–98, 114–116]; however, in many cases the authors have chosen their error measures heuristically and provide little or no justification for them. Many different measures have been used with some degree of success. In one-dimensional problems, a finite-difference method can be successful if it merely places many equally spaced nodes in regions of large error measure and smoothly increases the node spacing in regions of smaller error measure. A finite-element method does not require as smooth a change of node spacing and hence may require fewer nodes than finite-difference methods, which require continuity of all orders of derivative of the transformation $\partial X/\partial \xi$ at each node. Similarly, in two- or three-dimensional problems a finite-difference method requires a higher degree of grid smoothness and orthogonality than a finite-element method. Analyses of truncation error induced by grid deformation in finite-difference formulations are given in Mastin [17], Thompson and Mastin [18], Kalnay de Rivas [19], Hoffman [20], and Hindman [21]. Dupont [22] has performed a partial error analysis of adaptive finite-element techniques.

Most methods move the nodes to equidistribute a function of the measure at each node. Three basic styles of node movement have been used.

In periodic node movement, the nodes are held stationary in physical space for several time-steps, after which the measure is computed; the nodes are then shifted abruptly to their new positions. Data on $\mathbf{A}^k$ is usually moved from the old grid to the new grid by a method such as interpolation so as to prevent solution distortion or instability.

In alternating node movement, the measure is computed after every time-step, and the nodes are shifted in response to the measure. New ODEs are obtained, and the values of $\mathbf{A}^k$ computed for the next time-step. Interpolation is often used to transfer data from the old to the new grid.

In simultaneous node movement, the measures are taken and used to compute an ODE in time for X_i^k at each node. The nodes are not shifted between time-steps; rather, a set of ODEs for X_i^k is solved along with the ODEs for the physical solution on each time-step, in order to simultaneously compute the new values of $\mathbf{A}^k$ and X_i^k in the next time-step. Interpolation of $\mathbf{A}$ data from the old grid to the new grid is not necessary in this case.

In the periodic and alternating methods, terms in $\partial X_i/\partial t$ contained in the PDEs for $\mathbf{A}$ in computational coordinates are usually set to zero when the PDEs are solved on each time-step. However, it is better to obtain some nonzero value for $\partial X_i/\partial t$, as done by Klopfer and McRae [26] and Anyiwo [44], so as to allow node movement to effect the solution at each node. A variant form of alternating node movement is well suited to this approach; temporary values of $\mathbf{A}^k$ for the next time-step are computed on the old grid with fixed nodes. The temporary $\mathbf{A}^k$ determine new node positions and hence velocities. The node velocities are used in computing corrected $\mathbf{A}^k$ values.

Dwyer, Smooke, and Kee [35] have pointed out that simultaneous computation of node position and solution value converts a linear problem into a nonlinear one and usually makes a nonlinear problem harder. Thus, simultaneous node movement might appear to be less appropriate than the other two methods. In the solution of stiff problems, however, alternating or periodic node movement may result in instabilities owing the mismatching of node distribution and solution, unless time-steps are kept very small.

2.3. *Summary of Various Adaptive Methods*

Adaptive methods can be conveniently grouped in terms of the manner of redistribution of the nodes: redistribution to minimize or equidistribute the integral of the error measure, or redistribution by use of pseudoforces derived from the error measure. Specific details can be found in the summary that follows. In many ways, the classification is artificial and, as shown by Anderson [6], the formulas that finally evolve in either case have much in common. As much as possible, conceptually similar methods are treated sequentially. Some methods have been discussed in great detail as a basis for shorter accounts of related work. In general, critique of and commentary on the adaptive methods are reserved for Section 2.4.

Since variants of these adaptive methods have been used in time-independent problems, such as two-point boundary-value problems, some relevant adaptive ODE solution methods are reviewed.

2.3.1. *Methods Based on Minimization or Equidistribution of the Integral of the Error Measure*

Gough, Spiegel, and Toomre [23] used an adaptive algorithm to solve two-point boundary-value problems. The one-dimensional ODE for a K-component dependent variable was transformed from the physical frame X to a frame ξ with uniform node separation and was written in central-difference form at each node. The system of equations obtained was solved using a Newton–Raphson iteration. An error measure based on the mth derivative of A_k and X with respect to ξ was computed using

$$E_m = \sum_{k=1}^{K} \frac{1}{R_k} \left| \frac{d^m A_k}{d\xi^m} \right|^2 + \frac{\lambda}{X_f - X_0} \left| \frac{d^m X}{d\xi^m} \right|^2, \tag{2.3}$$

where A_k is the kth component of the solution, and R_k and $X_f - X_0$ are the maximum changes over the ξ domain of A_k and X, respectively. The weighting constant λ was usually set to K. In practice, only E_1 and E_2 have been used as measures. The transformation from X to ξ was computed by solving an ODE obtained by minimizing the integral of E_m over the ξ domain.

Node movement was alternated with the solution of the ODE for A_k so as to obtain convergence to a final, highly adapted, and accurate solution. This alternating-node-movement procedure was claimed to be faster and more stable than solving ODEs for A_k and X simultaneously. The ODE based on E_2 yielded a better truncation-error reduction than the ODE based on E_1, but the first ODE required a better initial physical node distribution for its solution to converge.

Pierson and Kutler [24] solved a one-dimensional problem in which a PDE was transformed from physical coordinates (X, t) to computational coordinates (ξ, t) with equidistributed nodes. The error measure was the square of the third derivative of the solution with respect to ξ. The PDE was reduced to an ODE at each node by use of central differences. The ODEs were solved time-step by time-step using an implicit finite-difference method.

The nodes were moved every few time-steps so as to minimize the integral of the error measure, as approximated by the trapezoidal rule. The transformation between X and ξ was obtained by writing X as a finite series of Chebyshev polynomials in ξ. Equations for the coefficient of the Chebyshev polynomials were obtained by requiring the above minimization, subject to constraints on maximum and minimum node separations, and solved by utilizing a simplex method. Truncation-error reductions achieved were equivalent to those obtainable using twice the number of fixed equidistributed nodes.

Denny and Landis [25] adaptively solved a two-point boundary-value problem using, as an error measure, the leading truncation-error terms in a three-point finite-difference approximation to the ODE. The truncation-error terms were dif-

ferentiated with respect to the nodal coordinate in order to obtain a finite-difference formula for the nodal position that minimized the truncation error. The systems of equations for node-position and node-solution value were solved alternately in a manner similar to that of Gough *et al.* [23]. As pointed out by Thompson [7], this procedure concentrates the nodes where the solution is smooth, rather than in high-gradient regions as would be desirable.

Klopfer and McRae [26] adaptively solved a one-dimensional shock-tube problem using finite differences. The error measure was the leading term of the truncation error of the PDE transformed to computational coordinates (ξ, t). The node spacing $\partial X/\partial \xi$ was a linear function of the smoothed error measure E such that

$$\frac{\partial X}{\partial \xi} \propto 1 - \frac{E}{E_{\max}}. \tag{2.4}$$

The node-movement process was repeated after each time-step. The time derivatives of the nodal coordinates, X^k, were computed from the changes in nodal position and included in the PDEs for the solution **A**. Artificial viscosity was used in the solution algorithm to stabilize the calculation. Since the explicit predictor–corrector scheme of MacCormack [27] was used, the minimum node spacing was limited to more than one-tenth of the maximum to avoid the development of excessive stiffness. Despite this limitation, Klopfer and McRae were able to use one-fifth of the nodes necessary for a nonadaptive calculation.

White [28] solved a two-point, vector-valued boundary-value problem. The ODE for **A** was transformed from physical coordinate X to a computational frame ξ, where the integral of an error measure E was equidistributed. The transformation can be written in the form

$$\xi(X) = \int_{X_0}^{X} \frac{E(x)\,dx}{\theta}, \tag{2.5}$$

where

$$\theta = \int_{X_0}^{X_l} E(x)\,dx. \tag{2.6}$$

Thus, the nodes were equidistributed in the ξ domain. The integral was manipulated to obtain ODEs for X and θ in the ξ frame. The ODEs for **A**, X, and θ in the ξ frame were expressed in central-difference form and solved simultaneously, using a Newton–Raphson iteration.

Various versions of the measure were used by White; these included solution arc length $dS = [1 + \|d\mathbf{A}/dX\|^2]^{1/2}\,dX$, local truncation error of the ODE, and weighted node-to-node change in the solution.

White [29] extended the method for use in solving time-dependent problems. The equations for X and θ in the (ξ, t) frame were written as

$$\left|\frac{\partial X}{\partial \xi}\right|^2 + \left\|\frac{\partial \mathbf{A}}{\partial \xi}\right\|^2 - \theta^2 = 0 \tag{2.7}$$

and

$$\frac{\partial \theta}{\partial \xi} = 0 \tag{2.8}$$

and then solved along with transformed PDEs for **A** time-step by time-step. Note that Eq. (2.7) puts a bound on the values of $|\partial X/\partial \xi|$ and $\|\partial \mathbf{A}/\partial \xi\|$. Variables **A**, X, and θ were written as an average of their values at the present and next time-steps, and the derivatives with respect to ξ were approximated by central differences. A Newton–Raphson iteration was used to solve the resultant system of equations.

An approach that is very similar to that of White [28] was taken by Ablow and Schechter [30], who used the error measure

$$E = 1 + \lambda\, |\Omega|, \tag{2.9}$$

where λ is a weighting constant and Ω is the circular curvature of the solution given by

$$\Omega = \frac{d^2A/dX^2}{[1 + [dA/dX]^2]^{3/2}}. \tag{2.10}$$

The authors point out that, as a general rule, A and X should be normalized so as to be of approximately the same magnitude. The ODE for a fixed-point boundary-value problem was transformed to computational coordinates ξ having equidistributed error measure and nodes. A second ODE was obtained by differentiation of the equidistribution formula

$$E^2\left[\left[\frac{dX}{d\xi}\right]^2 + \left[\frac{dA}{d\xi}\right]^2\right] = 1 \tag{2.11}$$

with respect to ξ. The two ODEs were solved simultaneously in a manner similar to that of White [28].

Sanz-Serna and Christie [31] have used the method of White [29] as a starting point for their adaptive method. S, the solution arc-length, was approximated as the sum over all nodes of $\Delta S = [[\Delta X]^2 + [\Delta A]^2]^{1/2}$, where ΔX and ΔA are the change in spatial coordinate and in solution amplitude, respectively, between adjacent nodes. New node positions were chosen so that S changed by an equal amount from node to node. A simple projection of the new value of S onto X was employed at each node. The solution was transferred from the old grid to the new grid by means of interpolation. A limit on the maximum ratio of ΔX at adjacent nodes was indirectly enforced by specifying a value β such that for all intervals with $[\Delta A]^2$ less than β, the change in arc length was computed using $\Delta S = [[\Delta X]^2 + \beta]^{1/2}$.

Nonlinear, viscid, one-component model equations were solved using a variable time-step implicit finite-difference method. Node adaptation was alternated with time-stepping. One adaptive solution using 100 nodes was comparable in accuracy to a nonadaptive solution using 400 nodes.

Revilla [32] has used the same procedure as Sanz-Serna and Christie but with $[\Delta A]^2$ replaced by the absolute value of the change in ΔA between adjacent intervals. He reported an additional halving of the number of nodes required to match the results of nonadaptive solutions.

In order to solve a time-dependent PDE, Dwyer, Kee, and Sanders [33], Dwyer, Raiszadeh, and Otey [34], Dwyer, Smooke, and Kee [35], Dwyer, Kee, Barr, and Sanders [36], Dwyer, Sanders, and Raiszadeh [37], and Dwyer [38, 39] used a transformation from the spatial coordinate system (X_1, X_2, t) to the coordinate system (ξ_1, ξ_2, t) having an equidistributed integral of an error measure. Much of their work was based on the literature of adaptive solution of two-point boundary-value problems. An error measure $E(Z)$ was used involving derivatives with respect to arc length Z along curves of constant ξ_1 or ξ_2 in physical space ($dZ^2 = [dX_1]^2 + [dX_2]^2$):

$$E(Z) = 1 + \lambda_1 \left| \frac{\partial A}{\partial Z} \right| + \lambda_2 \left| \frac{\partial^2 A}{\partial Z^2} \right|. \tag{2.12}$$

Weighting factors λ_1 or λ_2 can be made large in order to equidistribute nodal differences in A or nodal differences in $\partial A/\partial Z$, respectively.

The PDE was transformed from the physical coordinates (X_1, X_2, t) to computational coordinates (ξ_1, ξ_2, t) and was solved for A using an alternating-direct-implicit finite-difference method (an implicit iteration employing Newton's method is used in one-dimensional problems). The error measure was taken and used in a coordinate transformation to move the nodes adaptively. The coordinate transformation along each curve of constant ξ_2 in physical space was

$$\xi_1(X_1, X_2, t) = \frac{\int_0^Z E(z)\, dz}{\int_0^{Z_{\max}} E(z)\, dz}. \tag{2.13}$$

Because movement after each time-step led to oscillations, the nodes were held stationary for several time-steps before an error measure was taken. Equation (2.13) was then solved for equally spaced values of ξ_1 by means of numerical quadrature so as to reposition the nodes. A bound on the ratio of node separation from element to element was used to maintain transformation smoothness. The solution was transferred from the old grid to the new grid by means of interpolation. A similar transformation could be used along a curve of constant ξ_2, but Dwyer *et al.* [33–39] have adapted the node positions along only one set of numerical coordinates.

One-dimensional versions of the method worked very well, giving truncation errors which were only obtainable by use of ten times the number of nodes in a nonadaptive calculation. However, in one two-dimensional problem solved by Dwyer *et al.* [33], solution oscillations occurred as portions of the grid developed increased skewness in physical space. The authors [33] suggested that the procedure of Potter and Tuttle [40] might be utilized to reduce the computed grid skewness. Potter and Tuttle move nodes along lines of constant ξ_1 in order to define

lines of constant ξ_2 on which a specified function of ξ_1 satisfies Laplace's equation. The curvilinear coordinate system produced is orthogonal. Dwyer *et al.* [33] also encountered problems in two dimensions when adaptation was permitted along boundaries; coordinate-system collapse occurred in a heat-transfer problem when Neumann boundary conditions were used.

Dwyer and Onyejekwe [41] have incorporated the orthogonalization method of Potter and Tuttle in the adaptive solution of axisymmetric flow past a parachute-like body. Nodes were moved along one set of coordinate lines in response to the error measure and moved along the other set of coordinate lines to impose orthogonality. As the flow Reynolds number was merely 200, only a small amount of adaptation was required, but the number of nodes required to obtain an acceptable solution was reduced significantly and oscillations owing to grid skewness were eliminated.

The grid adaptation and orthogonalization were performed in response to error measures resulting from an initial estimate of the solution computed using an implicit upwind-difference scheme on a stationary grid. The time-step was then repeated using an implicit central-difference scheme and, although left unstated, utilizing node velocities computed using the difference in node locations in the old grid and in the new grid so as to avoid errors induced by interpolation.

Kansa, Morgan, and Morris [42] developed a method in which the nodes are moved so that the PDEs, when transformed to a numerical frame, have a minimal dependence on spatial derivatives and may therefore be easier to solve. The transformed K-component system of PDEs was written in the form

$$\left.\frac{\partial A_k}{\partial t}\right|_{\xi} = G_k - \frac{\partial F_k}{\partial X} + \frac{dX}{dt}\frac{\partial A_k}{\partial X}. \tag{2.14}$$

In this equation, A_k is one of the components of the solution; G_k is a function of the solution components and the spatial coordinate X; and F_k is a function of X, the solution components, and gradients of the solution components. The last term in Eq. (2.14) results from the transformation from spatial coordinates to numerical coordinates where the nodes are stationary and equidistributed. One can define a velocity V_k for each component, as given below:

$$V_k = \frac{\partial F_k/\partial X}{\partial A_k/\partial X}. \tag{2.15}$$

If the nodes are moved so that $dX/dt = V_k$, then the PDE for component k is reduced to

$$\left.\frac{\partial A_k}{\partial t}\right|_{\xi} = G_k. \tag{2.16}$$

At a shock wave, Eq. (2.15) is a restatement of the Rankine–Hugoniot jump condi-

tions, with V_k being the shock velocity. Within a rarefaction fan, V_k is the velocity of the characteristics. At a contact surface, V_k is the velocity of the gas. Owing to numerical inaccuracies, in a multicomponent problem, V_k will not, in general, be the same for all components. Kansa *et al.* [42] chose the node velocities dX/dt to minimize

$$E = \sum_{k=1}^{K} \left[\frac{dX}{dt} - V_k \right]^2, \tag{2.17}$$

approximated by using a three-point-collocation polynomial to compute gradients (and using values of X at the unknown time level to make the expression implicit). The result was a tridiagonal-matrix equation for the node velocities. For each time-step, solution of Eq. (2.16) using an implicit finite-difference method that incorporated artificial viscosity terms was alternated with the computation of node velocities. The nodes were remapped by means of interpolation after each time-step, so as to enforce a minimum node separation and prevent node crossing. Interpolation was also used to prevent excessive clustering of nodes about shocks: the nodes were moved in order to equistribute the magnitude of the third derivative of the solution times the cube of node separation. Use was made of the ideas of Dwyer and coworkers and also of those of Davis and Flaherty [114].

Kansa *et al.* [42] used their method, with 21 nodes, to solve a gas diffusion problem. It was stated that, if a nonadaptive method had been used to solve the same problem with equivalent accuracy, 670 nodes would have been required. Extension of the method to higher dimensions was also discussed.

Ghia, Ghia, and Shin [43] have developed a method, similar to that of Kansa *et al.* [42], in which the nodes are moved so that a PDE, when transformed to a numerical frame, has a minimal dependence on first-order spatial derivatives. The model PDE for a one-dimensional problem can be written in the spatial frame as

$$\left. \frac{\partial A}{\partial t} \right|_X = \mu \frac{\partial^2 A}{\partial X^2} - F[X, A] \frac{\partial A}{\partial X} - G[X, A] A - g[X], \tag{2.18}$$

where $F[X, A]$, $G[X, A]$, and $g[X]$ are functions of the indicated variables and μ is a very small number analogous to the inverse of the Reynolds number. When transformed to the numerical frame the PDE can be written

$$\left[\left. \frac{\partial A}{\partial t} \right|_\xi + GA + g \right] \left[\frac{\partial X}{\partial \xi} \right]^2 - \mu \frac{\partial^2 A}{\partial \xi^2} = R \frac{\partial A}{\partial \xi} \left[\frac{\partial X}{\partial \xi} \right]^{-1}, \tag{2.19}$$

where

$$R = \left[\left. \frac{\partial X}{\partial t} \right|_\xi - F \right] \left[\frac{\partial X}{\partial \xi} \right]^2 - \mu \frac{\partial^2 X}{\partial \xi^2}. \tag{2.20}$$

In Eq. (2.19) the first-order spatial derivatives of the solution are isolated in the

right-hand side. If μ is very small or, equivalently, if the Reynolds number is very large, it becomes very difficult, using a central-difference discretization of $\partial A/\partial \xi$, to obtain a solution of Eq. (2.19) free of Gibbs' oscillations unless the coefficient of $\partial A/\partial \xi$ is as small as or smaller than μ in magnitude. Otherwise the truncation error in $\partial A/\partial \xi$ swamps the second-order term. The adaptive strategy consists of moving the nodes so that

$$\left[\left.\frac{\partial X}{\partial t}\right|_{\xi} - F\right]\left[\frac{\partial X}{\partial \xi}\right]^2 - \mu \frac{\partial^2 X}{\partial \xi^2} = 0. \tag{2.21}$$

Equation (2.21) was discretized using upwind differences for $\partial X/\partial \xi$ and solved for the velocity at each node using an implicit technique that required inversion of a tridiagonal matrix. The resultant node velocities were used to compute R (the residue of Eq. (2.20)), which was substituted into Eq. (2.19). Central differences were used to discretize both Eq. (2.19) and R (which would be zero only if central differences were used in computing the node velocities) and the same implicit technique was used to obtain a value for the time derivatives of the solution, A, at each node. The use of upwind differences in the first part of the algorithm enhanced stability. Stability was also enhanced by varying (probably increasing) the value of μ used in Eq. (2.21) from that used in Eq. (2.20) so that not all nodes would enter the steepest portion of the solution but some would be forced out and made to round out the corners in the solution. If μ were set to zero in Eq. (2.21), then the nodes would move to follow the characteristics of an inviscid model problem and would thus tend to cross each other inside shock waves as in the method of Kansa *et al.* [42]. As μ is increased, this tendency to cross is lessened.

The method has been generalized to solve problems in two spatial dimensions. Both one-dimensional (using 51 nodes with values of μ as small as 10^{-4}) and two-dimensional (using 41×41 nodes with a Reynolds number as large as 667) problems have been solved. Time-stepping proceeded until the time-asymptotic solution emerged. Extremely smooth solutions were maintained throughout the time integration. Very few iterations were required if only the steady-state solutions were desired. The method has not yet been generalized to solve multicomponent problems.

Anyiwo [44] has used an adaptive method with a two-stage coordinate transformation. The PDE is transformed from physical coordinates (X_i, t) to orthogonal, curvilinear coordinates (S_i, t), and thence to a computational space (ξ_i, t) having equidistributed nodes. Anyiwo used an error measure E defined by

$$E = \exp(\gamma) = \exp\left[\sum_{i=1}^{I} \gamma_i\right]. \tag{2.22}$$

That is, E is a product of factors given by

$$E_i = \exp(\gamma_i). \tag{2.23}$$

γ_i, which is a measure of grid deformation in each coordinate direction, with weighted contributions from each coordinate system, is given by

$$\gamma_i = \lambda^1 \alpha_i^1 \Omega_i + \lambda^2 |\alpha_i^2 + \beta_i^2| \Omega_i + \lambda^3 \sigma^3 \Omega_i. \tag{2.24}$$

λ^1, λ^2, and λ^3 are weighting constants. $\alpha_i^1 = 1/N_i$ and $\Omega_i = 1/ln(N_i - 1)$, where N_i is the number of nodes in direction i in (ξ_i, t) coordinates. α_i^2 and β_i^2 are the circular curvature and torsion of curves of constant S_i in physical coordinates, and

$$\sigma^3 = |\delta_i^3| + |\alpha_i^3 + \beta_i^3|. \tag{2.25}$$

δ_i^3, α_i^3, β_i^3 are the slope, circular curvature, and torsion of the solution variable A in (S_i, t) coordinates. σ^3 is replaced by a weighted sum of terms if A is a multicomponent solution variable.

The transformation between the (S_i, t) coordinates and the (ξ_i, t) coordinates is obtained by requiring that, along each S_i direction,

$$\frac{\partial S_i}{\partial \xi_i} = \frac{\int_C E_i(s_i)\, ds_i}{N_i E_i(S_i)}, \tag{2.26}$$

which in essence equidistributes E over the nodes. The quantity $\partial S_i/\partial \xi_i$ is proportional to the separation between curves of constant S_i in physical space.

The transformation between the (X_i, t) coordinates and the (S_i, t) coordinates is given by

$$dS_i = \cos(\theta_{ij})\, dX_j, \tag{2.27}$$

where θ_{ij}, the angle between coordinate direction S_i and coordinate direction X_j, is constrained—$\cos(\theta_{jk}) \cdot \cos(\theta_{ik}) = 0$ if $i \neq j$—such that the (S_i, t) coordinates are orthogonal. The variation of θ_{ij} is chosen so that the boundaries of the physical domain correspond to curves of constant curvilinear coordinate. A solution-weighted interpolation formula is used to compute θ_{ij} in the interior of the physical domain. θ_{ij} is calculated in response to the variation of $\partial S_i/\partial \xi_i$ along the ξ_j direction at each interior node in such a way as to blend in smoothly with the angular behaviour of the boundaries of the physical domain. Anyiwo described a simple interpolation procedure for use on two-dimensional grids.

A typical time-step solution proceeds as follows. The error measure E is used to compute the metric transformation derivatives $\partial S_i/\partial \xi_i$ at each node and hence the (S_i, t) to (ξ_i, t) coordinate transformation. The values of $\partial S_i/\partial \xi_i$ are in turn used to determine the new values of θ_{ij} and hence of the (X_i, t) to (S_i, t) coordinate transformation. The PDEs for the physical solution are transformed to the (ξ_i, t) frame and solved for the next time-step values by use of a finite-difference scheme. The new solution and transformations are used to compute a new error measure. The cycle continues time-step by time-step. The time-stepping is omitted for several cycles at the start of a computation in order to obtain a grid adapted to the initial conditions.

An additional means of control of adaptation was obtained by inserting a nonzero value for $\partial X_i/\partial t$ in the transformed PDEs for the physical variable. This value was obtained by requiring that the spatial coordinates satisfied a conservation equation

$$\left.\frac{\partial X_i}{\partial t}\right|_{\xi} = \nabla[-cUX_i + \mu \nabla X_i] \tag{2.28}$$

in the (ξ_i, t) frame, where c is a positive constant less than or equal to one. μ is the viscosity coefficient, and U the advective velocity, for the physical solution PDEs. It is reported that Eq. (2.28) adds a constraint on node movement that maintains transformation smoothness.

Brackbill and Saltzman [45, 46], Brackbill [47], and Saltzman and Brackbill [48], extending a boundary-adaptive automatic-grid generation method used by Winslow [49], transformed a two-dimensional PDE for the physical variable A from physical coordinates (X_1, X_2, t) to numerical coordinates (ξ_1, ξ_2, t). A weighted sum of three error measures was used to determine the coordinate transformation. That is,

$$E = \lambda_s E_s + \lambda_0 E_0 + \lambda_v E_v, \tag{2.29}$$

where

$$E_s = [\nabla \xi_1]^2 + [\nabla \xi_2]^2 \tag{2.30}$$

is a measure of transformation smoothness,

$$E_0 = [\nabla \xi_1 \cdot \nabla \xi_2]^2 \tag{2.31}$$

is a measure of grid orthogonality in physical space, or

$$E_0 = [\nabla \xi_1 \cdot \nabla \xi_2]^2 J^3 \tag{2.32}$$

is an alternate measure of grid orthogonality with increased weighting in regions of large Jacobian of the transformation

$$J = \frac{\partial X_1}{\partial \xi_1}\frac{\partial X_2}{\partial \xi_2} - \frac{\partial X_1}{\partial \xi_2}\frac{\partial X_2}{\partial \xi_1}. \tag{2.33}$$

Regions of large Jacobian correspond to regions of large node separation in physical coordinates. The third measure, E_v, is defined by

$$E_v = WJ, \tag{2.34}$$

where W is a measure of the truncation error in the solution and has been given as

$$W = \left|\frac{\nabla A}{A}\right|^{\varrho}, \tag{2.35}$$

with Q being two or four in practice. W was averaged over several nodes and scaled between maximum and minimum values, so as to decrease the effect of grid irregularities on finite-difference approximations of ∇A. The Jacobian of the transformation, J, will tend to be small in regions of large W if E_v is equidistributed, and hence the grid will be refined in regions of large error measure. In this method the weighting constants λ_s, λ_0, and λ_v were generally of the order of unity.

The authors obtained a system of ODEs for the spatial coordinates by using the calculus of variations to minimize

$$\int E\,dV = \iint EJ\,d\xi_1\,d\xi_2, \tag{2.36}$$

the volume integral of E transformed to numerical coordinates. Brackbill [47] also used a weighted-measure procedure to move nodes adaptively along boundary lines. One-dimensional analogues of E_s and E_v were minimized over a boundary curve in order to obtain a system of ODEs for the positions of the nodes along the boundary.

The nodes were held stationary for several time-step solutions of the transformed PDEs for A. An explicit finite-difference method was used, with stability maintained by specifying a CFL limitation on time-step. The system of ODEs for the spatial coordinates were then solved using a Jacobi iteration. Only a few iterations were needed for convergence. Information on the solution was transferred from the old grid to the new one either by means of an interpolation function or by solution of transport equations in conservation-law form.

Excellent control of grid characteristics, so as to reduce solution error, was obtained by the authors. In one case, a nonadaptive, uniformly spaced grid solution required nine times the number of nodes used in an adaptive solution to obtain comparable accuracy. Saltzman and Brackbill [48] have applied the method to computation of nonsteady two-dimensional supersonic internal flow over a step.

Anderson [50] discusses a number of closely related adaptive techniques. Kreis, Thames, and Hassan [51] report that performance of the method of Brackbill and Saltzman can be improved in certain problems with large spatial variation in J if $E_v = W/J$, rather than $E_v = WJ$, is used for the solution-error measure.

Yanenko, Kovenya, Liseikin, Fomin, and Vorozhtsov [52], Yanenko, Kroshko, Liseikin, Fomin, Shapeev, and Shitov [53], and Yanenko, Liseikin, and Kovenya [54] also used a transformation from physical coordinates (X_1, X_2, t) to computational coordinates (ξ_1, ξ_2, t). The error measure was of the form

$$E = \lambda_c E_c + \lambda_a E_a + \lambda_v E_v, \tag{2.37}$$

where

$$E_c = \left|\frac{\partial \xi_1}{\partial X_1} - \frac{\partial \xi_2}{\partial X_2}\right|^2 + J^R \left|\frac{\partial \xi_1}{\partial X_2} + \frac{\partial \xi_2}{\partial X_1}\right|^2 \tag{2.38}$$

is a measure of the transformation's departure from conformality, with R as a small positive integral power, and where

$$E_a = \left| U_1 - \frac{\partial X_1}{\partial t} \right|^2 + \left| U_2 - \frac{\partial X_2}{\partial t} \right|^2 \tag{2.39}$$

is a measure of the degree to which the nodes move with the medium, in other words, the "Lagrangianness" of the transformation. (U_1, U_2) is the medium velocity, and $(\partial X_1/\partial t, \partial X_2/\partial t)$ is the grid velocity in physical space. E_v is a measure of the solution variation:

$$E_v = WJ^Q. \tag{2.40}$$

Here, W is a weighted function of gradients of solution components, and Q is a small positive integral power. The Jacobian of the transformation, J, will tend to be small in regions of large W if E_v is equidistributed, and hence the grid will be refined in regions of large error measure.

A PDE in time-coordinates and space-coordinates for the transformation was obtained by minimizing the integral of E over X_1, X_2, and t. Alternately, in Yanenko, Liseikin, and Kovenya [54], a PDE in space-coordinates was obtained by minimizing the integral of E over X_1 and X_2. This PDE was solved simultaneously with the PDE for **A** using a time-stepping finite-difference method. Yanenko *et al.* [52, 54] also discussed adaptation along a single coordinate direction.

Hindman and Spencer [55], inspired by the boundary-adaptive automatic-grid generation method of Winslow [49], have developed a one-dimensional adaptive method in which the transformation from the spatial coordinates (X, t) to computational coordinates (ξ, t) (where $\xi^k = k/N$ and k ranges from 0 to N) satisfies a one-dimensional Poisson equation

$$\frac{\partial^2 P}{\partial X^2} = P. \tag{2.41}$$

In this equation, $P = P(\xi)$ is a forcing function that is large in regions where it is desirable to concentrate nodes. The formulation leads to a basic method of estimating node velocities. The problem, transformed to the numerical frame, requires the solution of

$$\frac{\partial^2 X}{\partial \xi^2} + \left[\frac{\partial X}{\partial \xi} \right]^3 P = 0, \tag{2.42}$$

subject to the boundary conditions $X(\xi^0) = X^0$ and $X(\xi^N) = X^N$.

Hindman and Spencer point out that an equation in the form of (2.42) can be obtained by taking the second derivative with respect to ξ of the expression

$$\xi(X, t) = \text{const} \int_{X^0}^{X} E\, dx. \tag{2.43}$$

This mapping of ξ onto X was inspired by the adaptive method of Dwyer, Kee, and Sanders [33]. The integrand is an error measure of the form

$$E = 1 + \lambda W, \tag{2.44}$$

where $\sqrt{W} = \partial A/\partial \xi$ is the slope, in the numerical frame, of the solution of a single-component PDE used to test the adaptive method. Comparison of Eq. (2.42) to the result of differentiating equation (2.43) twice implies that

$$P = \frac{1}{E}\frac{\partial E}{\partial \xi}\left[\frac{\partial X}{\partial \xi}\right]^{-2} = \frac{\lambda}{1+\lambda W}\frac{\partial W}{\partial \xi}\left[\frac{\partial X}{\partial \xi}\right]^{-2}. \tag{2.45}$$

Hindman and Spencer deduced an alternate form of the forcing function by requiring that the integral

$$\int_{X^0}^{X^n}\left[\left[\frac{\partial \xi}{\partial X}\right]^2 + \lambda W \frac{\partial X}{\partial \xi}\right] dx \tag{2.46}$$

be minimized. This procedure was inspired by the work of Brackbill [47]. The first term in the integrand promotes smoothness of the transformation, while the second term promotes concentration of nodes in regions with a large value of $W = [\partial A/\partial \xi]^2$. The calculus of variations is applied to the integral to yield an PDE in ξ and $\partial \xi/\partial X$ for the transformation. The solution,

$$-2\left[\frac{\partial X}{\partial \xi}\right]^{-1} + \lambda W\left[\frac{\partial X}{\partial \xi}\right]^2 = \text{const}, \tag{2.47}$$

of the PDE is differentiated with respect to ξ, to obtain an equation that can be put in the same form as Eq. (2.42) if

$$P = \lambda \frac{\partial W}{\partial \xi}\frac{\partial X}{\partial \xi}\Big/\left(2 + 2\lambda W\left[\frac{\partial X}{\partial \xi}\right]^3\right). \tag{2.48}$$

Only the first form of the forcing function has been used for an actual calculation. It should be pointed out here that an alternate definition of $W = [\partial A/\partial X]^2$ would be more consistent with the usage of Dwyer, Kee, and Sanders [33] and Brackbill [47].

Following the boundary adaptive method of Hindman, Kutler, and Anderson

[56], rather than solving Eq. (2.42) directly, Hindman and Spencer differentiate the equation with respect to time to obtain

$$\frac{\partial^2 \dot{X}}{\partial \xi^2} + 3 \frac{\partial \dot{X}}{\partial \xi} \left[\frac{\partial X}{\partial \xi} \right]^2 P + \left[\frac{\partial X}{\partial \xi} \right]^3 \dot{P} = 0, \tag{2.49}$$

where $\dot{X}$ is the node velocity and $\dot{P}$ is the derivative of P with respect to time at fixed computational coordinate. The equation above is approximated by central differences at each node. The central-difference expression for $\dot{P}$ is expanded in terms of $\dot{A}$ and $\dot{X}$ and the PDE for A used to eliminate the value of $\dot{A}$ at each node from the expression. The resulting equations at each node are manipulated into a matrix equation for node velocities. The matrix is tridiagonal since P^k, the value of the forcing function at node k, depends only on the values of X and A at nodes $k-1$, k, and $k+1$.

The original PDE for A was transformed to the computational coordinate system. A time-stepping algorithm was applied in an alternating manner to the transformed PDE and to the matrix equation for the node velocities. The time-stepping algorithm used was either an implicit first- or second-order accurate method or the explicit second-order accurate predictor–corrector method of MacCormack [27]. There was a tendency for the nodes to drift out of optimal adjustment despite the use of the matrix equation for the node velocities. To compensate for this, the node positions were periodically adjusted using a tridiagonal matrix equation based on the approximation of Eq. (2.42) by central differences. The initial placement of the nodes also satisfied Eq. (2.42).

A number of single-component test problems were solved using the adaptive method, and the results were compared to those obtained employing a grid with fixed nodes. In all cases the adaptive grid provided better results, but the solutions exhibited numerical oscillations on both the adaptive and nonadaptive grid because not enough nodes were used to resolve the solutions properly. Hindman and Spencer discuss a theoretical extension of their method to solve two-dimensional problems but have not yet performed any two-dimensional calculations.

Petzold [57] uses a more direct approach to computing adaptive node velocities and has applied her method to the solution of one-dimensional multicomponent PDEs. The method developed by Petzold generalizes and extends earlier work by Hyman [58]. Petzold obtains an equation for the node velocities by minimizing the measure:

$$I = \sum_k \omega [\dot{X}^k]^2 + \| \dot{\mathbf{A}}^k \|^2, \tag{2.50}$$

consisting of the sum over all nodes of the square of the node velocity and the square of the vector norm of the time derivative of the solution at fixed node number. ω is a weighting constant. The second term is expressed as a function

of node velocity by transferring the model PDE to the frame where the nodes are stationary and approximating the PDE by spatial central differences to obtain

$$\dot{\mathbf{A}}^k = \dot{X}^k \frac{\partial \mathbf{A}^k}{\partial X} + \mathbf{f}^k \left[\mathbf{A}^k, \frac{\partial \mathbf{A}^k}{\partial X}, \frac{\partial^2 \mathbf{A}^k}{\partial X^2} \right]. \tag{2.51}$$

The equation for the velocity of node k is obtained by differentiating the measure with respect to the velocity of node k to obtain

$$\omega \dot{X}^k + \dot{\mathbf{A}}^k \cdot \frac{\partial \mathbf{A}^k}{\partial X} = 0. \tag{2.52}$$

The intent is to minimize the time-rate of change both of the solution and of the node coordinate in the moving frame so that large time-steps can be taken. Petzold added an extra term to the equation above to discourage node crossing. The additional term is the result of the minimization of an auxiliary "penalty" function that becomes large when node separation approaches zero. The "penalty" function may be interpreted as diffusing the mesh velocity. The transformed model PDE and the equations for node movement form a system of ODEs, which is solved using an implicit ODE solver. Petzold discusses the effect of variant forms of the system of ODEs on the stability of the solution.

The penalty functions are unable to completely prevent node crossing but increase the size of time-step that can be taken without the nodes crossing. After every time-step, a reference node distribution is computed by requiring that the local node spacing be adjusted to make the quantity

$$\Delta X \left\| \frac{\partial \mathbf{A}}{\partial X} \right\| + [\Delta X]^2 \left\| \frac{\partial^2 \mathbf{A}}{\partial X^2} \right\| \tag{2.53}$$

less than a specified tolerance and that the number of nodes be the smallest able to fulfill the tolerance condition. The new grid is based on a comparison between the old grid and the reference grid in such a way that most nodes are not moved and nodes are seldom added or deleted. Thus, interpolation to transfer the solution from the old grid to the new grid need only be applied to a few nodes at any time-step.

One nonlinear viscid model two-component equation was solved using 57 nodes with an accuracy comparable to that obtained by a nonadaptive solution using 600 nodes. If Eq. (2.53) was used without computing node velocities, the same problem took 17 times as many time-steps to solve.

Carcaillet, Dulikravich, and Kennon [59] have devised a method that constructs the sum, over all nodes, of measures of grid smoothness, grid orthogonality, and solution smoothness. A simplex procedure is applied to the sum to adapt a rectangular grid. The method is an extension of the work of Hayes, Kennon, and Dulikravich [60] and Kennon and Dulikravich [61], which used the first two

measures only and which was inspired by the method of Brackbill and Saltzman [46]. The value of the three measures are calculated at a given node as follows.

The area of the four cells adjacent to the node is computed. Grid smoothness at the node is monitored by computing the sum of the squares of the differences in area of cells with common borders. The sum will be zero if all adjacent cells are of equal area.

The relative-position vectors of adjacent nodes connected by grid lines are also computed. The four relative-position vectors with origin at the given node are used to form four dot products, which are squared and summed to monitor orthogonality at the node. The dot products are chosen so that their sum will be zero if the grid is locally orthogonal.

Solution smoothness is monitored at the node by approximating the following error measure; the area of the four cells connected to the node multiplied by a function of the gradient of a solution component. The function is of the form of Eq. (2.35) with Q given the value of two. To reduce computational effort, only an approximation to the above is used. It is computed as follows. The square of the magnitude of each of the four relative-position vectors originating at the node is multiplied by the average value of the function in the two cells to each side of the relative-position vector. The four results are summed. The function values are manipulated to limit variation of cell areas.

The sums at all the nodes are totalled to form a global measure of grid quality, which is analogous to the volume integral of Brackbill and Saltzman [45 48]. A nonlinear simplex procedure is used to vary the node positions until the measure is minimized. The method is quite robust, as is demonstrated by its ability to orthogonalize a grid deliberately tangled with overlapping grid lines in order to produce regions of negative Jacobian of the transformation. Orthogonalization methods using a differential equation to compute node positions, such as that of Brackbill and Saltzman, would be unable to deal with an initial grid containing regions of negative Jacobian.

The method of Carcaillet *et al.* [59] has been used to adapt a grid to the time-asymptotic solution of a single component PDE. It is able to reduce error to at least a sixth of that obtained with a nonadaptive grid with the same number of nodes. Carcaillet *et al.* plan to apply their method to the iterative time-asymptotic solution of transonic flow problems with shocks. Procedures for transferring information from the old grid to the new grid are under study.

Kennon and Dulikravich [61] and Carcaillet, Kennon, and Dulikravich [62] extended the grid-property measures to three-dimensional grids. Carcaillet, Kennon, and Dulikravich [62] and Kennon and Dulikravich [63] used a modified grid-smoothness measure to achieve clustering of nodes about specified lines, surfaces, or three-dimensional regions.

Miller and Miller [64] and Miller [65] devised a finite-element-based PDE solution technique, which they call the moving finite element (MFE) method. The method has been further developed in Djomehri and Miller [66], Gelinas, Doss, and Miller [67], Gelinas, Doss, Vajk, Djomehri, and Miller [68], Gelinas and

Doss [69–71], Gelinas, Doss, and Carlson [72], Djomehri [73], and Miller [74, 75]. In this method the error measure may be interpreted as being the square of the residual of the PDE written in finite-element form. ODEs for the nodal value of the physical variable and the nodal coordinate are obtained by minimizing the integral of error measure over spatial coordinates. Only the simplest one-dimensional version will be described (and reformulated) herein, but the method has also been applied in two dimensions.

The method is most simply described in numerical coordinates where the transformed one-dimensional PDE for a single component physical variable A can be written as

$$\left.\frac{\partial A}{\partial t}\right|_{\xi} - \left.\frac{\partial A}{\partial X}\frac{\partial X}{\partial t}\right|_{\xi} = f\left[t, X(\xi, t), A(\xi, t), \frac{\partial A}{\partial \xi}, \frac{\partial^2 A}{\partial \xi^2}, \frac{\partial X}{\partial \xi}, \frac{\partial^2 X}{\partial \xi^2}\right]. \tag{2.54}$$

The continuous solution, A, and physical coordinate, X, are approximated by the piecewise linear functions A^h and X^h given by

$$A^h = \sum_{k=1}^{N} \alpha^k(\xi)\, A^k(t) \tag{2.55}$$

and

$$X^h = \sum_{k=1}^{N} \alpha^k(\xi)\, X^k(t). \tag{2.56}$$

The coefficients A^k and X^k are the values at each node k of the solution and of the spatial coordinate, respectively. α^k is the basis function at node k. Note that, in what follows, $\xi^k - \xi^{k-1} = 1$. In this simplest formulation of the MFE method, α^k is a linear "hat" function defined by

$$\alpha^k = \begin{cases} 0 & \text{if } \xi < \xi^{k-1} \\ \xi - \xi^{k-1} & \text{if } \xi^{k-1} \leqslant \xi < \xi^k \\ \xi^{k+1} - \xi & \text{if } \xi^k \leqslant \xi < \xi^{k+1} \\ 0 & \text{if } \xi^{k+1} \leqslant \xi. \end{cases} \tag{2.57}$$

Since the nodes are allowed to move, it can be shown that

$$\begin{aligned} \left.\frac{\partial A^h}{\partial t}\right|_X &= \left.\frac{\partial A^h}{\partial t}\right|_{\xi} - \frac{\partial A^h}{\partial \xi}\left[\frac{\partial X^h}{\partial \xi}\right]^{-1}\left.\frac{\partial X^h}{\partial t}\right|_{\xi} \\ &= \sum_{k=1}^{N} \alpha^k \dot{A}^k + \beta^k \dot{X}^k, \end{aligned} \tag{2.58}$$

where β^k is a second basis function defined by

$$\beta^k = -\frac{\partial A^h}{\partial X}\alpha^k = \begin{cases} 0 & \text{if } \xi < \xi^{k-1} \\ -\alpha^k M^k & \text{if } \xi^{k-1} \leqslant \xi < \xi^k \\ -\alpha^k M^{k+1} & \text{if } \xi^k \leqslant \xi < \xi^{k+1} \\ 0 & \text{if } \xi^{k+1} \leqslant \xi \end{cases} \tag{2.59}$$

with

$$M^k = \frac{A^k - A^{k-1}}{X^k - X^{k-1}}. \tag{2.60}$$

A set of ODEs for A^k and X^k at each node was obtained by minimizing the integral of the square of the residual of the PDE, with X and A replaced by their discrete representations, X^h and A^h. The minimization was obtained by requiring that

$$\frac{\partial I}{\partial \dot{A}^k} = \frac{\partial I}{\partial \dot{X}^k} = 0 \tag{2.61}$$

for all nodes k, where

$$I = \int \left[\left.\frac{\partial A^h}{\partial t}\right|_X - F\left[t, X^h(\xi, t), A^h(\xi, t), \frac{\partial A^h}{\partial \xi}, \frac{\partial X^h}{\partial \xi} \right] \right]^2 \frac{\partial X^h}{\partial \xi}\, d\xi. \tag{2.62}$$

It should be noted that PDE terms containing the second derivative with respect to ξ can be reduced to integral terms containing the first derivative with respect to ξ by using suitable manipulation and integration by parts. The resultant integral equations are

$$\int \alpha^k \left[\left.\frac{\partial A^h}{\partial t}\right|_X - F\left[t, X^h(\xi, t), A^h(\xi, t), \frac{\partial A^h}{\partial \xi}, \frac{\partial X^h}{\partial \xi} \right] \right] \frac{\partial X^h}{\partial \xi}\, d\xi = 0 \tag{2.63}$$

and

$$\int \beta^k \left[\left.\frac{\partial A^h}{\partial t}\right|_X - F\left[t, X^h(\xi, t), A^h(\xi, t), \frac{\partial A^h}{\partial \xi}, \frac{\partial X^h}{\partial \xi} \right] \right] \frac{\partial X^h}{\partial \xi}\, d\xi = 0. \tag{2.64}$$

The basis functions α^k and β^k act as test functions over the domain of integration.

The integrals can be manipulated to yield a system of ODEs. The ODEs can be written in the matrix form

$$[D]\dot{\mathbf{U}} = \mathbf{E}. \tag{2.65}$$

$[D]$ is a block tridiagonal matrix and $\dot{\mathbf{U}}$ a column vector containing the values of $\dot{A}$ and $\dot{X}$ at each node. $\mathbf{E}$, also, is a column vector. Additional terms are added to $[D]$ and $\mathbf{E}$ to prevent excessive node speed or the crossing of node positions and

to prevent singularity of $[D]$ in regions of constant slope where the β^k and α^k are linearly dependent. The additional terms are the result of the minimization of auxiliary "penalty" functions that become large when node separation becomes small. Miller [65, 74] discusses the many different forms that have been used for these penalty functions and also discusses other strategies for node-movement regularization. The numerical values of $[D]$ and $\mathbf{E}$ are obtained at each time-step by evaluating the integrals of Eq. (2.63) and (2.64) either analytically or by means of numerical quadrature.

The implicit multistep backward-difference method of Gear [12], with many modifications, has been used to solve Eq. (2.65) for A and X at each node. Alternately, an implicit Runge–Kutta method devised by K. Miller has been used. A Newton iteration was used as part of the solution of Eq. (2.65). Adaptive computations typically required one-tenth the number of nodes used by nonadaptive computations with results of comparable accuracy.

In problems with multicomponent solutions $\mathbf{A}$, the authors defined multicomponent functions $\boldsymbol{\beta}^k$ and obtained weighted ODEs for X^k. The use of higher-order basis functions (quadratic or cubic) was also discussed. The method has been successfully extended to the solution of problems with two spatial dimensions in Djomehri and Miller [66], Gelinas *et al.* [68, 72], and Miller [75]. However, as indicated by Miller [75], this method has not achieved the impressive time-step values observed in one-dimensional problems. Miller [74, 75] expounds the advantages of an alternative form of the MFE method in which the MFE terms are normalized by a function of the gradient of the solution. Dupont [22] has completed a partial analysis of the MFE method.

Gelinas, Doss, and Miller [67] have implemented the method in a commercial package for the solution of PDEs on one-dimensional domains. The package allows the user to describe simple PDEs by executing a series of subroutine calls that automatically assemble the required terms into Eq. (2.65).

Dukowicz [76] has simplified the moving finite element method of Miller and Miller [64, 65] by using finite differences to reduce the size of the matrix equation that must be solved. Equation (2.54) is approximated by three point differences and used to eliminate $\dot{A}^k$ in Eq. (2.58). The result is substituted into Eq. (2.62). Differentiation of Eq. (2.62) with respect to $\dot{X}^k$ yields a tridiagonal system of equations which is solved for node velocities. The finite difference form of Eq. (2.54) is then used to obtain $\dot{A}^k$.

The MFE method both in unmodified form and in simplified form was used to solve a one-component viscid nonlinear model equation whose solution consisted of a moving shock wave of constant height. The solution given by the simplified method was slightly more accurate, but special differencing had to be used at the top and bottom of the shock, where the node separation changed abruptly, in order to avoid instability in the solution.

The chief advantage of the simplification is that a much smaller system of equations may be solved when an explicit time-stepping algorithm is in use. Unfortunately, many problems of interest are stiff and require the use of implicit

time-stepping algorithms. The size of the matrix equation associated with the use of an implicit time-stepping algorithm is not reduced by the simplification.

Mueller and Carey [77, 78] extended the work of Miller and Miller [64, 65] by using piecewise polynomial basis functions and a different form of penalty function. Their method was applied to a number of scalar one- and two-dimensional problems that required modest adaptivity.

Herbst, Schoombie, and Mitchell [79, 80] and Herbst and Schoombie [81] extended the work of Miller and Miller [64, 65] by using piecewise Hermite cubic polynomials, in place of the test functions α^k and β^k used in Eqs. (2.63) and (2.64), while retaining the basis functions α^k and β^k in Eqs. (2.55), (2.56), and (2.58). Herbst, Schoombie, Griffiths, and Mitchell [82] have analyzed their method and that of Miller and Miller and conclude that both methods tend to equidistribute the spatial second derivative of the solution over the grid. The error resulting from use of piecewise cubic test functions was significantly reduced in comparison to that resulting from use of piecewise linear test functions. Mitchell and Herbst [83] discuss the applicability of moving finite element methods to various types of one-dimensional problems and point out difficulties associated with the solution of two- or three-dimensional problems.

Wathen and Baines [84] have reformulated the method of Miller and Miller [64, 65] so as to avoid the use of penalty functions. They point out that, for many hyperbolic PDEs, time-step is not limited by stability considerations if the nodes move with the characteristics of the solution. The solution at a node remains constant as the node moves at a constant velocity. Hence, large time-steps can be obtained even if the ODE solution method is explicit. The use of penalty functions in the MFE method to ensure nonsingularity of the matrix $[D]$ stops the MFE method from moving the nodes along characteristics; hence a stiff ODE solver is required.

A single-component one-dimensional version of the method of Wathen and Baines is most easily described. The MFE method matrix $[D]$ is expresed in the form $[M]^{\mathrm{T}}[C][M]$, where $[M]$ and $[C]$ are matrices and $[M]^{\mathrm{T}}$ is the transpose of $[M]$. $[C]$ is singular only if nodes are coincident, and $[M]$ is singular only if regions of constant slope, where the nodes are collinear, exist.

If two nodes approach one another closely and are about to cross, matrix $[C]$ is modified to introduce a moving boundary containing a shock transition between two coincident nodes. Any node that subsequently enters the shock is eliminated from the calculation.

If three adjacent nodes are collinear, then the matrix equation is modified to make the velocity of the middle node zero when the matrix equation is solved. The zero velocity is then adjusted so that the middle node will be equispaced from the two adjacent nodes at the end of the time-step. Multiple collinear nodes are treated in a similar fashion.

In both cases the treatment removes the singularity in $[D]$. It is shown that the modified matrix $[D]$ possesses very desirable properties with regard to inversion by sparse matrix techniques. Owing to the necessity to manipulate $[D]$ at each time-

step, an explicit ODE solution method was used. The method has been extended to solve two-dimensional problems.

Baines and Wathen [85] have extended the method to solve a multicomponent one-dimensional problem. It is shown that there are definite advantages in using a separate node distribution for each component. Several one- and two-dimensional model PDEs have been solved. The number of nodes used and the size of the time-steps compare very favourably to those resulting from use of the unmodified MFE method. Wathen [86] has extended the analysis of the method. Issues, such as criteria for adding nodes and treatment of reflection of shocks off solid boundaries, must be resolved before the method becomes generally useful.

Benner, Davis, and Scriven [87] have also sought to avoid use of the penalty functions of Miller and Miller [64, 65]. They retain Eq. (2.63) and the piecewise linear basis functions α^k, but they replace Eq. (2.64) by

$$E - E^k = 0, \tag{2.66}$$

where E^k is an error measure given as

$$E^k = \Delta X^k [|M^{k+1} - M^k| + |M^k - M^{k-1}|]. \tag{2.67}$$

The factor $|M^{k+1} - M^k| + |M^k - M^{k-1}|$, obtained by approximating the integral from X^{k-1} to X^k of the absolute value of the second derivative of the piecewise linear function A^h, will be large when the slope M of A^h has a large jump at X^k or X^{k-1}. E is the average value of E^k over all elements. The aim is to equidistribute E^k so that $\Delta X^k = X^k - X^{k-1}$ will be smallest where $|M^{k+1} - M^k| + |M^k - M^{k-1}|$ is largest.

Equations (2.66) and (2.63) for all k constitute a system of equations whose residual can be minimized by means of a modified Newton iteration to obtain values for A^k, X^k, and E at each time-step. However, Benner, Davis, and Scriven prefer to separate the computation of the grid from the solution of the PDE in order to give faster convergence of a Newton iteration on Eq. (2.63) and to reduce the size of the system of equations that must be solved. Equation (2.63) is solved for a stationary grid, and the nodes are then placed so that, subject to limitations on maximum node separation and on the value of $\Delta X^k / \Delta X^{k-1}$, the linear interpolant of the square root of E^k is distributed evenly over the elements. Use of the square root provides a form of measure smoothing.

The solution of a scalar one-dimensional model PDE using 40 nodes was comparable to the solution obtained by the MFE method of Miller and Miller. However, the comparison suffered a bit since the MFE method used only 21 nodes and since the method of Benner, Davis, and Scriven encountered convergence difficulties at an outflow boundary. Extension of the method to multicomponent or two-dimensional problems was discussed.

Lee and Ramos [88] solved a flame-propagation problem using an adaptive finite-element technique. The PDEs were transformed from physical coordinates (X, t) to a normalized Lagrangian coordinate system (ξ, τ), where ξ varied from 0

to 1. The transformation was defined by $\tau = t$, $\partial\xi/\partial X = \rho$, and $\partial\xi/\partial t = -\rho V$, where V was the velocity and ρ the normalized density of the gas. The continuous solution $\mathbf{A}$ at time τ_n was approximated by

$$\mathbf{A}^h(\tau_n) = \sum_{k=1}^{N} \alpha^k(\xi)\, \mathbf{A}^k(\tau_n), \tag{2.68}$$

where $\mathbf{A}^k$ is the value of the solution at node k for the current node distribution, and where the α^k are linear basis "hat" functions defined by

$$\alpha^k = \begin{cases} 0 & \text{if } \xi < \xi^{k-1} \\ \dfrac{\xi - \xi^{k-1}}{\xi^k - \xi^{k-1}} & \text{if } \xi^{k-1} \leqslant \xi < \xi^k \\ \dfrac{\xi^{k+1} - \xi}{\xi^{k+1} - \xi^k} & \text{if } \xi^k \leqslant \xi < \xi^{k+1} \\ 0 & \text{if } \xi^{k+1} \leqslant \xi. \end{cases} \tag{2.69}$$

The method of Galerkin was employed to obtain ODEs for the solution at each node, and the ODEs were solved using finite differences in time.

The values of ξ^k where changed every 10 time-steps, in order to concentrate the nodes in regions of steepest temperature gradient. The element with the largest temperature gradient was assigned the minimum node separation $h_m = \Delta\xi_m$, dictated by physical considerations. The node spacing h for the other elements was adjusted to minimize the integral

$$I = \int_0^1 \frac{1}{h}\left[\frac{\partial h}{\partial \xi}\right]^2 d\xi, \tag{2.70}$$

subject to the constraint

$$\int_0^1 \frac{1}{h}\, d\xi = N - 1. \tag{2.71}$$

The minimization yielded a differential equation for h constrained so the total number of elements was $N-1$. The differential equation was solved analytically to obtain the new value of $\Delta\xi$ for each element.

The new basis functions thus obtained were used to define a new approximation to the solution,

$$\mathbf{A}_*^h(\tau_n) = \sum_{k=1}^{N} \alpha_*^k(\xi)\, \mathbf{A}_*^k(\tau_n), \tag{2.72}$$

where the new nodal $\mathbf{A}_*^k$ values were computed by requiring that

$$\int_0^1 \left[\frac{1}{2} \mathbf{A}_*^h \mathbf{A}_*^h - \mathbf{A}^h \mathbf{A}_*^h \right] d\xi \tag{2.73}$$

be minimized. An adaptive computation with 171 nodes gave results comparable to nonadaptive finite-element and finite-difference solutions using 901 nodes.

In a later paper, Ramos [89] reports that a similar adaptive computation using 37 nodes gives results comparable to a nonadaptive finite-difference method using 272 nodes. Ramos points out that the algorithm used to compute the node spacing $\Delta\xi$ cannot, in its present form, deal with problems characterized by the presence of multiple moving fronts.

2.3.2. *Methods Based on Attraction and Repulsion Pseudoforces between Nodes*

In a number of methods, a node attracts others when a measure of the truncation error at the node is larger than average. If the measure is smaller than average, the other nodes are repelled.

Rai and Anderson [90, 91] and Anderson and Rai [92] have developed one such method for one- and two-dimensional problems. They used an error measure E_i in numerical coordinates (ξ_i, t). Various forms of E_i were used, including $|\partial A/\partial \xi_i|$, $|(\partial A/\partial \xi_i)/(\partial X_i/\partial \xi_i)|$, $|\partial^2 A/\partial \xi_i^2|$, and $|\partial X_i/\partial \xi_i| + \lambda\, |\partial A/\partial \xi_i|$. The inclusion of functions of $\partial X_i/\partial \xi_i$ in the measures enhanced the smoothness of the transformation. Measure-averaging was used in the case of higher-order derivatives to avoid instabilities.

The time dependence of X_i was computed at each node using a sum of pseudoforces between nodes as follows. Consider a two-dimensional problem solved on a rectangular grid. Given a numerical space of N by M nodes, the partial time-derivative of ξ_1 at a fixed point in physical space was defined for each node k, q at its current location in physical space, by using the following formulation:

$$\left.\frac{\partial \xi_1^{kq}}{\partial t}\right|_X = -\lambda_1 \sum_{m=1}^{M} \left[\sum_{n=k+1}^{N} \frac{E_1^{nm} - E_1^{avm}}{r^Q} - \sum_{n=1}^{k-1} \frac{E_1^{nm} - E_1^{avm}}{r^Q} \right]. \tag{2.74}$$

E_1^{nm} is the error measure E_1 at node n, m; E_1^{avm} is an average of error measure E_1 along a line of constant ξ_1 in physical space. r is the distance in the numerical frame between nodes k, q and n, m. Q is a positive power which, if it is small, will allow distant nodes to affect one another's motion, resulting in a form of measure smoothing. An equivalent sum was used to compute $\partial \xi^{kq}/\partial t$:

$$\left.\frac{\partial \xi_2^{kq}}{\partial t}\right|_X = -\lambda_2 \sum_{n=1}^{N} \left[\sum_{m=q+1}^{M} \frac{E_2^{nm} - E_2^{avn}}{r^Q} - \sum_{m=1}^{q-1} \frac{E_2^{nm} - E_2^{avn}}{r^Q} \right]. \tag{2.75}$$

The current transformation between frame (X_i, t) and frame (ξ_i, t) was used to compute the corresponding node velocities $(\partial X_1^{kq}/\partial t, \partial X_2^{kq}/\partial t)$ from the ξ_1 and ξ_2

time derivatives. Constants λ_1 and λ_2 were adjusted on each time-step, so that no node velocity could exceed a prespecified value. The velocities constitute ODEs for (X_1, X_2) at each node, which were solved, time-step by time-step, along with the ODEs for A at each node, using the explicit finite-difference method of MacCormack [27].

λ_1 and λ_2 were held below a prespecified value so that grid velocities were damped as the error measures became equidistributed. All grid velocities were exponentially damped if the Jacobian of the transformation at any node changed from the Jacobian at the start of the calculation by more than a prespecified ratio. Unfortunately, limitation of Jacobian ratio tends to reduce the adaptability of the method. Grid distortion occurred when boundary nodes were forced to move tangentially to the domain boundaries while the forces on internal nodes were left unmodified, so that the internal-node velocities had significant nontangential components. "Reflection" of node properties about boundaries improved results. Given a real node, its image (with an identical value of error measure) was placed at an opposite and equal distance from the boundary. The image nodes were included in the force sums (2.74) and (2.75) so that the boundary nodes experienced tangential forces only, and near-boundary nodes experienced reduced nontangential forces.

A one-dimensional version of this method worked well; however, in some two-dimensional problems failure to reduce truncation error occurred because large cross derivatives of the solution were present in sparsely noded portions of the grid. None of the measures used by Rai and Anderson estimated these cross derivatives, resulting in an inappropriate node distribution. The authors suggested the use of such measures as $|\partial^3 A/\partial X_1\, \partial X_1\, \partial X_2|$ or $|\partial^3 A/\partial X_1\, \partial X_2\, \partial X_2|$ to improve performance but have not published any results to date.

The truncation-error reduction achieved was equivalent to the reduction obtainable by a nonadaptive method that uses four times the number of nodes. The authors state that the computer time required to solve a simple problem using an adaptive grid was usually higher than that required to solve it using a fixed grid. The allowable time-step sizes were reduced where fine mesh clustering occurred because of the use of an explicit ODE solution method; moreover, adaptive mesh movement added significant computational overhead. In more complex problems, execution times for adaptive solutions have been smaller than those for nonadaptive solutions.

In a later paper, Anderson [93] has indicated that this adaptive method is best suited for computation of time-asymptotic solutions and was originally developed for that purpose. As Anderson points out, a steady-state solver would take less computational effort to obtain the same results. Anderson [93, 94] has reviewed a number of alternate methods for computing node velocities and introduces some new ideas, as yet untested.

Greenberg [95] has solved a one-component PDE transformed to computational coordinates (ξ, t) in which the nodes are equidistributed. The transformed PDE was solved time-step by time-step using an explicit central finite-difference method. After each time-step, central differences were used to compute the local gradient of

the solution as an error measure, E^n. The error measure was employed to compute spring constants, K^{nm}, which were used to define an ODE for

$$\Delta X^n = X^n - X^{n-1}. \tag{2.76}$$

The ODE

$$\frac{d(\Delta X^n)}{dt} = \sum_{m=1}^{N} K^{mn}\,\Delta X^m - \sum_{m=1}^{N} K^{nm}\,\Delta X^n, \tag{2.77}$$

inspired by chemical-rate-constant equations, automatically ensures that the sum

$$\sum_{n=1}^{N} \Delta X^n \tag{2.78}$$

remains constant. The form of the K^{nm} was chosen to ensure that node separation would decrease in areas of large error measure at the expense of increase in node separation in areas of smaller error measure, subject to present limits on the maximum and minimum allowed node separation. The K^{nm} decrease in magnitude as the error measure is equidistributed. Equation (2.77) was linearized and solved analytically to determine the node distribution at each time-step. Preliminary results were obtained on some simple problems. Unfortunately, these problems did not test the method very thoroughly since they required only modest adaptation. Extension of the method to general-shaped multidimensional domains is under development.

Madsen [96] has developed an adaptive method that is reminiscent of that of Greenberg [95]. The method has been applied to the solution of a two-component nonlinear set of model PDEs on a one-dimensional domain. The ODE for the node velocities, which bears a striking resemblance to the equation for node placement developed by Klopfer and McRae [26], can be expressed in the form

$$\frac{d(\Delta X^n)}{dt} = \langle E \rangle - E^n, \tag{2.79}$$

where E^n is the value of E, a weighted multicomponent error measure, for the interval between node n and node $n-1$. Each component of E^n consists of either the absolute value of an estimate of the spatial-truncation error in the approximation to the right-hand side of a PDE or the absolute value of one of the following quantities: the change of the value, the slope, or the curvature of each component of the solution from node $n-1$ to node n. In order to smooth the node distribution, $|\Delta X^n|$ is also used as a component of E^n. $\langle E \rangle$ is the average value of E^n over all the intervals. The weighting constants are adjusted as the values of E^n change with time so that $\langle E \rangle$ remains constant.

If the node separation in a given interval is less than a specified minimum, the corresponding value of E^n is reduced (i.e., the weight is reduced) to discourage or

prevent the nodes from approaching any closer. In order to ensure that $\langle E \rangle$ remains constant, the values of E^n for each of the other intervals are increased in proportion to the amount by which the node separation in each interval exceeds the specified minimum separation.

The system of PDEs, approximated using spatial central differences, and the grid velocity equation were solved simultaneously by using a modified form of the implicit ODE-solver of Gear [12]. The use of the adaptive method resulted in a significant improvement in the solution of a problem involving the collision of two waves.

Adjerid and Flaherty [97] have employed, as the basis of an adaptive finite-element method, an equation identical to Eq. (2.79) except for the inclusion of a proportionality factor, λ, on the right-hand side. The error measure E^n was obtained by computing the integral of a function of discretization error from X^{n-1} to X^n. The term $\langle E \rangle$ was eliminated by subtracting the basic equation on two adjacent elements to obtain a tridiagonal system of equations for node velocities, with rows given by

$$\dot{X}^{n+1} - 2\dot{X}^n + \dot{X}^{n-1} = -\lambda[E^{n+1} - E^n]. \tag{2.80}$$

After each time-step, the value of $\langle E \rangle$ is compared to a tolerance. If $\langle E \rangle$ exceeds the tolerance, then every element is split in half. If $\langle E \rangle$ falls below one-tenth of the tolerance, then every other node is removed. Removal of every second node typically only doubles $\langle E \rangle$ on the next time-step. Hence, the above strategy ensures that the number of nodes remains constant for long periods of a calculation.

A number of viscid model PDEs have been solved. The above set of ODEs, a set of ODEs obtained by discretizing the model PDEs using a Galerkin procedure with linear basis functions, and a set of ODEs that estimates the discretization error were solved simultaneously with a modified form of the stiff ODE solver of Gear [12]. A large value of λ accelerated equidistribution of discretization error but also increased stiffness and, for reasons not completely understood by Adjerid and Flaherty, caused inappropriate movement of the nodes in some cases. Initial results for problems requiring modest adaptivity were good.

Adjerid and Flaherty [98] improved the method in a number of ways. These included an heuristic procedure for automatic selection of appropriate values for λ and provision for local, rather than global, refinement of the grid. Problems requiring substantial adaptivity were solved. Future plans include extension to solution of problems in higher dimensions and the use of higher-order basis functions.

Winkler, Norman, and Newman [99], extending the work of Tscharnuter and Winkler [100] and Winkler and Newman [101], have applied an adaptive method to the determination of gas flows and electromagnetic radiation fields during star formation. The equation for node placement can be expressed in the form

$$\eta^n - \eta^{n-1} = 0, \tag{2.81}$$

where

$$\eta^n = \omega_x \Delta\alpha^n + \omega_{xz} \frac{[\Delta\alpha^n]^2}{\Delta\alpha^{n+1}\,\Delta\alpha^{n-1}} + \Omega_x \Delta\beta^n + \Omega_{xz} \frac{[\Delta\beta^n]^2}{\Delta\beta^{n+1}\,\Delta\beta^{n-1}}$$
$$+ \omega_M \frac{M^n - M^{n-1}}{M_{\text{scale}}} + \Omega_M \frac{M^n - M^{n-1}}{M^n + M^{n-1}} + \sum_k \Omega_k E_k^n. \tag{2.82}$$

When Eq. (2.81) is satisfied, each term in Eq. (2.82) is more or less equidistributed over the grid in rough proportion to the relative size of the corresponding weighting constant ω_x, ω_{xz}, Ω_x, Ω_{xz}, ω_M, Ω_M, or Ω_k.

$\Delta\alpha^n$ and $\Delta\beta^n$ are normalized node separations defined by

$$\Delta\alpha^n = \frac{X^n - X^{n-1}}{X_{\text{scale}}} \tag{2.83}$$

and

$$\Delta\beta^n = \frac{X^n - X^{n-1}}{X^n + X^{n-1}}. \tag{2.84}$$

If only the weighting constant ω_x was nonzero, the node coordinates X^n would be equally spaced apart; however, if only Ω_x was nonzero, the change in the logarithm of X^n would be equidistributed over the grid. The latter node distribution is used in preference to the former for the problems solved. The change in M^n, the Lagrangian mass variable, would be equidistributed if only ω_M was nonzero. If only Ω_M was nonzero, the change in the logarithm of M^n would be equidistributed over the grid.

The summation contains solution-error measures usually defined as

$$E_k^n = \frac{|G_k^n - G_k^{n-1}|}{G^n + G^{n-1}}, \tag{2.85}$$

where G_k^n is the value at node n of various properties of the gas or electromagnetic radiation field. The normalization results in equidistribution of the change in the logarithm of G_k^n. In some cases, the denominator contains a "floor" value to prevent division by zero. Some error measures contain no denominator; in those cases, the change in G_k^n is equidistributed across the grid.

The second or fourth term in Eq. (2.82) is used to counteract a tendency for nodes to cross in some problems. These terms have a function similar to that of the penalty-function terms of Miller and Miller [64, 65].

The equations describing the radiation field and gas were discretized using finite differences and were solved along with the equation for node placement using an implicit ODE solver with second-order time accuracy. Because of the form of Eq. (2.81) the matrix to be solved was block pentadiagonal. One-dimensional spherically symmetric Euler equations with artificial viscosity-tensor terms were

used to describe the time evolution of the gas. In one calculation using 292 nodes, node separation varied over five orders of magnitude to resolve the solution; time-step size exceeded the explicit CFL limit by a factor of 5×10^5. In another problem using 80 nodes, node separation varied by a factor of 10^4 and the CFL limit was exceeded by nine orders of magnitude. Very thin shock waves were well resolved without evidence of Gibbs' oscillations.

Winkler, Mihalas, and Norman [102] introduce the additional terms

$$+\omega_{\max}\left[\frac{\Delta\alpha^n}{\Delta\alpha_{\max}}\right]^Q+\omega_{\min}\left[\frac{\Delta\alpha_{\min}}{\Delta\alpha^n}\right]^Q+\Omega_{\max}\left[\frac{\Delta\beta^n}{\Delta\beta_{\max}}\right]^Q+\Omega_{\min}\left[\frac{\Delta\beta_{\min}}{\Delta\beta^n}\right]^Q \tag{2.86}$$

into Eq. (2.82). The terms encourage node separation (if $\omega_{\max} \neq 0$ and $\omega_{\min} \neq 0$), or the change in the logarithm of X^n (if $\Omega_{\max} \neq 0$ and $\Omega_{\min} \neq 0$), to remain between the specified minimum and maximum. The integer power Q is typically set to four. The authors also take the square of the error measures containing absolute values in Eq. (2.82), so as to eliminate the need for the use of absolute values.

Winkler, Mihalas, and Norman have altered Eq. (2.81) into the form

$$\eta^n - \eta^{n-1} - \omega_t \frac{dX^n}{dt} = 0, \tag{2.87}$$

so as to provide temporal smoothing. The temporal weight, ω_t, is nonzero only when the solution-error measures are decreasing with time and is proportional to the square of the sum of the fractional change in the error measures, as compared to previous times. The net effect is to prevent the large reduction of time-step size that results from the brief loss of node concentration when the solution is temporarily smooth, as occurs during the reflection of a shock wave off a wall. Winkler, Norman, and Mihalas [103] provide a comprehensive discussion of all aspects of the physics and numerical solution procedures used in their work.

Dorfi and Drury [104], inspired by the work of Winkler *et al.* [99–103], have produced a similar adaptive method, which has also been applied to the solution of one-dimensional Euler equations containing artificial viscosity terms. They define the K-component error measure

$$E^n = \left[1 + \sum_{k=1}^{K}\left[\frac{\beta^n}{\alpha_k^n}\frac{\Delta A_k^n}{\Delta X^n}\right]^2\right]^{1/2}, \tag{2.88}$$

where $\Delta A_k^n = A_k^n - A_k^{n-1}$ is the change in the kth component of the solution in the interval between node n and node $n-1$. β^n and α_k^n are scale factors associated respectively with the spatial coordinate and the solution; in the simplest case these scale factors have the value one, so that E^n is proportional to the arc length from node n to node $n-1$ of the solution. Sensitivity of the error measure was improved in spherically symmetric problems by using the arithmetic mean of X^n and X^{n-1} for β^n and the harmonic mean of A_k^n and A_k^{n-1} for α_k^n. A form of E^n containing second-order spatial derivatives has been used in some problems.

The nodes are moved to make $\beta^n/\Delta X^n$ proportional to the time- and space-averaged value of E^n by requiring that

$$\frac{\bar{\eta}^n}{E^n} = \frac{\bar{\eta}^{n-1}}{E^{n-1}}, \tag{2.89}$$

where $\bar{\eta}^n$ is a time-smoothed quantity defined by

$$\bar{\eta}^n = \eta^n + \frac{\tau}{\Delta t}[\eta^n - \eta^n_{\text{old}}], \tag{2.90}$$

which is, in essence, a time-differenced equation for the spatially smoothed quantity η^n defined by

$$\eta_n = \frac{\beta^n}{\Delta X^n} - \gamma[\gamma+1]\left[\frac{\beta^{n+1}}{\Delta X^{n+1}} - 2\frac{\beta^n}{\Delta X^n} + \frac{\beta^{n-1}}{\Delta X^{n-1}}\right]. \tag{2.91}$$

The last expression is a difference equation for the variation of $\beta^n/\Delta X^n$ from node to node and is shown to be equivalent to requiring that the ratio $\beta^n/\Delta X^n/[\beta^{n-1}/\Delta X^{n-1}]$ be between $\gamma/[\gamma+1]$ and $[\gamma+1]/\gamma$. As γ is increased, the response of $\beta^n/\Delta X^n$ to the value of the error measure at distant nodes is increased. The value of γ is selected so that ΔX^n will not vary by more than 20 or 30% from node to node.

η^n_{old} is the value of η^n at the previous time-step. As $\tau/\Delta t$ is increased, the response of $\beta^n/\Delta X^n$ to the value of the error measure at previous time-steps is increased or, equivalently, the response to the values of the error measure at the current time-step is diminished. τ is, in effect, a damping time-constant, and the damping becomes significant when τ is a large fraction of the time-step size Δt. Small values of τ allow very fast response time but result in a loss of node concentration if two waves pass through one another. Larger values of τ reduce the loss of concentration but may also diminish the ability of the nodes to follow rapidly moving features.

The difference between the current and old values of η^n divided by Δt can be expressed as a function of node velocities. Hence, Eq. (2.89) could be expanded into a matrix equation for the node velocities. The expressions are reminiscent of some of the penalty terms used in Miller [74] but result in matrices of larger bandwidth. However, Dorfi and Drury instead make use of a technique similar to those of White [29] and of Winkler, Norman, and Mihalas [103]. The system of PDEs describing the flow is differenced and solved simultaneously with Eq. (2.87) by means of a Newton–Raphson iteration, which requires the computation of a Jacobian matrix. Because of the form of η^n, the Jacobian is block pentadiagonal. The iteration is also applied to a uniform grid with fixed initial-condition data to obtain the initial grid for a problem.

A number of problems have been solved by Dorfi and Drury with results comparable to those of Winkler *et al.* [99–103]. Note that the calculations of the former authors required a much smaller number of weighting constants than the

calculations of the latter authors. Dorfi and Drury suggest that their method can be extended to multidimensional problems by defining E^n and η^n as tensors.

Gnoffo [105, 106], making use of the work of Dwyer *et al.* [33] and that of Rai and Anderson, used an error measure to define tension–spring forces between nodes on a two-dimensional grid. The nodes are moved so as to equidistribute the spring force along lines of constant computational coordinate.

The force between adjacent nodes along a line of constant computational coordinate was defined as

$$F = K \Delta S, \tag{2.92}$$

where ΔS is the arc length between the two nodes and K is the local spring constant. Typically, K was of the form

$$K = 1 + E, \tag{2.93}$$

where E is an error measure. E was usually a weighted sum of the magnitudes of spatial derivatives of each component of the solution **A**. The adaptive procedure was applied along one set of computational coordinate lines only. The nodes were moved periodically every few time-steps using the following iterative procedure.

One curve of constant computational coordinate at a time was selected, and the nodes were moved along this curve using the formula

$$\Delta S^n = \frac{S_{\text{tot}}}{K^n \sum_m (1/K^m)}, \tag{2.94}$$

so as to equidistribute the spring forces $K^m \Delta S^m$. The nodes on a curve have been numbered consecutively for explanatory convenience. ΔS^n is the arc length between nodes n and $n-1$; K^n is the spring constant between the two nodes; S_{tot} is the total arc length. Information on the soution was transferred from the old grid to the new grid using an interpolation function, and new values of K^n were computed. Use of Eq. (2.94) and the interpolation process was repeated until the node locations converged. An averaging formula was applied to the spring constants to smooth out their node-to-node variation, and node movement was damped by taking a weighted average of the old and new node locations from iteration step to iteration step. The iterative procedure was repeated for each curve of constant computational coordinate. A similar formulation was used in some cases to concentrate nodes at predetermined locations. In that case, K^n varied exponentially with n, thus inducing a systematic increase or decrease in node separation.

The solution PDEs were discretized using a finite-volume method. A finite-volume method is a geometrically conservative finite-difference approximation to the integral form of the PDEs. The domain is broken up into control volumes and finite-difference equations obtained from the conservation of inter-volume fluxes. The system of equations maintains conservation of mass, energy, and momentum from element to element. In Cartesian coordinates, the finite-volume equations

reduce to those of the explicit method of MacCormack [27]. Hindman [21] discusses the advantages of, and the pitfalls involved in, various conservative and nonconservative methods of expressing differential equations.

The procedure has been applied to solve the Navier–Stokes equations for complete (forebody and afterbody) flowfields around blunt bodies. Time-stepping was applied until a steady-state solution emerged. Excellent results were obtained for moderate values of the weighting constants in the expression for K, but solution oscillations were encountered if the weighting constants used to compute E were too large or if flows with Reynolds number exceeding 5×10^5 were simulated.

Nakahashi and Deiwert [107] have extended the method of Gnoffo [105, 106] to include torsion springs in addition to tension springs. They adjust the node distribution in a two-dimensional domain so as to enforce grid orthogonality as well as adaptivity.

Adaptation is applied in one direction of computational coordinate at a time. For instance, given nodes with indices i, j, the index i is fixed, and the corresponding nodes with varying index j are caused to move along a curve whose shape is fixed by the original positions of the nodes. The node at location i, j is moved along the line of constant i in response to tension–spring forces from node $i, j+1$ and node $i, j-1$ and in response to torsion–spring forces from node $i-1, j$. In the latter case, if grid orthogonality is desirable, a straight reference line is defined that is perpendicular to the line of constant i and that intersects it at node i, j. The torsion–spring force is proportional to the angle between the reference line and a straight line joining node i, j to node $i-1, j$. In practice, the angle is expressed in terms of the projection of the line joining node i, j to node $i-1, j$ onto the line of constant i. Nakahashi and Deiwert have defined alternative reference lines that follow stream lines or are smooth extensions of the line of constant j that passes through node i, j. In practice, all three types of reference lines are used with different weights.

The node location formula is written as a force balance at each node in order to form a tridiagonal system of equations that may be solved for the optimal position of each node along the line of constant i. The force balance at node i, j is

$$K_{i,j}[S_{i,j+1} - S_{i,j}] - K_{i,j-1}[S_{i,j} - S_{i,j-1}] - T_{i-1,j}[S_{i,j} - \tilde{S}_{i,j}] = 0. \quad (2.95)$$

$S_{i,j}$ is the arc length between node i, 1 and node i, j along the line of constant i. $K_{i,j}$ is a tension–spring constant between node i, j and node $i, j+1$ that is usually proportional to the gradient of the solution between said nodes. In some cases, a stretching function is added to $K_{i,j}$. The stretching function varies in such a way as to impose a systematic increase or decrease in node separation along the line of constant i. For some problems it is desirable to smooth the tension–spring constants or solution components contained in the tension–spring constants by averaging them over three adjacent nodes. Adjustment is also made near boundaries between regions with differing adaptive coefficients in order to prevent excessive changes in node separation.

The arc length $\tilde{S}_{i,j}$ is the projection of the line between node i, j and node $i-1, j$ on the line of constant i. $\tilde{S}_{i,j}$ is adjusted so as to increase monotonically along the curve of constant i in order to prevent node crossing when the system of equations is solved. The torsion-spring constant $T_{i-1,j}$ is chosen to be a small fraction of the proportionality factor used in the tension-spring constants so that orthogonality or other grid properties will be most stringently enforced in regions of small solution gradient.

Only node $i-1, j$ is allowed to influence the position of node i, j by means of the torsion term, so that adaptation along lines of constant i can be applied in a marching manner in the direction of increasing i. Interpolation is used to transfer information from the old grid to the new grid after all lines of constant i have been adapted. If required, the adaptation is also applied along lines of constant j.

The method was applied to the solution of the thin-layer Navier–Stokes equations for subsonic and supersonic flow around aircraft afterbodies having sonic and supersonic underexpanded jets. The initial grid was generated by an algebraic method. Adaptation was applied to selected parts of the grid two or three times during the course of implicit calculation of the steady-state solution. The results showed significant improvement in definition of flow discontinuities such as shock waves and slip surfaces. Application of the method to similar problems is discussed in great detail in Deiwert, Andrews, and Nakahashi [108].

The method has been extended to three-dimensional geometries by Nakahashi and Deiwert [109]. They suggest that the method can be applied to unsteady flow problems as well, by evaluating grid velocities after every adaptation.

Eiseman [110] has applied an adaptive method on two-dimensional grids with rectangular cells. An optimal position is computed for each node in turn. Since the grid is rectangular, each node is connected by a grid line to four neighbours. A relative-position vector is computed for each neighbour. An error measure is computed for each of the relative-position vectors. The error measure is a sum of cell areas and of cell-area weighted quantities in the two cells to each side of the relative-position vector. These quantities include the magnitude of solution gradient, of grid-line curvature, of departure from grid-line orthogonality, and of departure from grid conformality.

The node is moved in response to the error measures as follows. The four relative-position vectors form two pairs of opposing vectors. Consider a pair of opposing relative-position vectors $\mathbf{R}^+$ and $\mathbf{R}^-$ with associated error measures E^+ and E^-. If $E^+ \|\mathbf{R}^+\|$ exceeds $E^- \|\mathbf{R}^-\|$, then a displacement vector

$$\frac{E^+ \|\mathbf{R}^+\| - E^- \|\mathbf{R}^-\|}{3[E^+ + E^-] \|\mathbf{R}^+\|} \mathbf{R}^+ \tag{2.96}$$

is computed. Otherwise the displacement vector

$$\frac{E^- \|\mathbf{R}^-\| - E^+ \|\mathbf{R}^+\|}{3[E^+ + E^-] \|\mathbf{R}^-\|} \mathbf{R}^- \tag{2.97}$$

is used. In effect, the displacement is along a line of constant computational coordinate. The other pair of opposing relative-position vectors is used to compute a second displacement vector. The new position of the node relative to the old position of the node is the sum of the two displacement vectors. The factor of three in the denominator of the displacement vectors is a safety factor to prevent excessive deformation of cells because of displacement of the node. Additional limits on displacement are employed in some cases. Eiseman discussed variants of the procedure. After the entire grid has been adapted, the node distribution is smoothed using a procedure described in Eiseman [112] to remove high-frequency oscillations in node position.

The method has been applied only to adapt grids to simple test functions rather than to solutions generated by PDEs. Extension of the method to higher dimensions is discussed. Eiseman [111] has applied similar procedures on two-dimensional grids with triangular cells. He also discusses strategies for addition and removal of nodes.

Eiseman [112, 113] applies one-dimensional adaptation along lines of constant curvilinear coordinate on two- and three-dimensional grids using a closely related procedure. As in Nakahashi and Deiwert [107], adaptation is applied in a marching manner in alternating directions. Nodes were displaced along each line of constant computational coordinate so as to equidistribute error measures computed between pairs of nodes. A projection technique similar to that of Sanz-Serna and Christie [31] was employed. The procedure has only been applied to adapt grids to simple test functions.

The one-dimensional method of Davis and Flaherty [114] may be interpreted as using pseudoforces to move nodes. The PDEs were solved with fixed time-step using a finite-element formulation in computational coordinates (ξ, t). Rectangular elements, transformed from trapezoidal space-time elements in physical coordinates (X, t), were used. As a truncation-error measure, the authors used the product of grid-spacing raised to the power "m" times the magnitude of the mth derivative of the solution with respect to X. This expression is related to the truncation error expected for the order of basis function used. Both piecewise linear $(m=2)$ and piecewise cubic Hermite $(m=4)$ finite-element basis functions were used.

The nodes were moved after every time-step so as to equidistribute the error measure by requiring that the node coordinates satisfied

$$\frac{X^{n+1}-X^{n}}{X^{n}-X^{n-1}}=\left|\frac{\partial^{m}A^{n}}{\partial X^{m}}\right|^{1/m}\Bigg/\left|\frac{\partial^{m}A^{n+1}}{\partial X^{m}}\right|^{1/m}. \tag{2.98}$$

The mth derivatives, approximated by finite differences, were given lower bounds to ensure stability of node movement. An under-relaxation iteration was used to solve Eq. (2.98), subject to constraints on first and last node coordinates and deformation of the trapezoidal space-time elements. The resultant node positions were then used for the next time-step, thus determining the shape of the space-time

elements used in the finite-element calculation of A^n. An attempt to use higher-order extrapolation of node positions led to wild oscillations in the grid.

A number of problems requiring modest adaptation were solved. Flaherty, Coyle, Ludwig, and Davis [115] reported difficulty in their attempts to extend the method to compute node velocities. The subsequent analysis of Coyle, Flaherty, and Ludwig [116] led to the method of Adjerid and Flaherty [97, 98].

2.4. *Closing Discussion of the Adaptive Methods Surveyed*

The methods for solving the ODEs obtained from the discretization of PDE problems vary considerably from author to author. Some authors were able to use explicit time-stepping formulae because the particular problems solved did not exhibit stiffness. In nonstationary gasdynamics applications, the methods can be adapted to use stiff-ODE solvers. As a penalty for obtaining larger time-steps, however, one generally must solve large matrix systems.

For gas flows having moving regions of rapidly changing solution, it is probably most desirable to move the nodes and compute the physical solution simultaneously in order to keep node distribution suited to solution variation. Fortunately, methods that move the nodes periodically or alternately with a time-step solution of the ODEs for the physical variables A^n can be altered so that "optimal" node velocities can be derived from the difference between old and "optimal" node positions. The node velocities constitute a set of ODEs for node position, which can be solved simultaneously with the ODEs for the A^n. Even though a larger set of ODEs must be solved simultaneously, the increase in computational cost is compensated in part by the fact that interpolation is not needed to transfer information from a old to a new grid.

The most promising two-dimensional methods are those of Yanenko *et al.* [52–54], Brackbill and Saltzman [45–48], and Nakahashi and Deiwert [107, 109], which control grid orthogonality and smoothness, and that of Miller and Miller [64, 65], which is not as sensitive to lack of smoothness or orthogonality.

The PDE used by Yanenko *et al.* for computing node movement derives its time dependence from the use of the Lagrangian measure E_a. In many problems, this might not be an appropriate measure. An alternate measure that will result in a time-dependent PDE for node movement is not immediately apparent.

Brackbill and Saltzman's method is easily converted to allow simultaneous computation of node position and the solution at each node; when adapted for use with a stiff-ODE solution algorithm, their method should work well with stiff problems. One technique of conversion, which White [29] has used, involves expressing A and X as averages of their values in the current and next time-steps and solving the resulting equations using a Newton–Raphson iteration. Alternatively, as discussed by Anderson [93] and Hindman and Spencer [55], the time-independent equation for node coordinates developed by Brackbill and Saltzman can be converted into a PDE for node velocities by use of differentiation with respect to time. As in the work of Hindman and Spencer, it may be necessary to periodically apply the

equation of Brackbill and Saltzman to combat a tendency for node concentration to relax.

Nakahashi and Deiwert have obtained results equal to or better than those of Brackbill and Saltzman, and their method has been applied to more elaborate flow problems. Implicit time-asymptotic solutions of two- and three-dimensional flows about aircraft afterbodies have been obtained. The method may also be applied to time-accurate solution of unsteady flows; in this case it may be useful, though perhaps difficult, to modify the method to allow simultaneous computation of the solution and node positions.

The MFE method of Miller and Miller uses a very natural and elegant formulation to control node movement. The node position and node solution are both obtained by the equidistribution of one error measure: the residual of the PDE written in finite-element form. Unfortunately, inversion of the large matrices associated with the method, can consume a large proportion of total computational time, although this may be offset by the large time-steps an implicit method can take. The modifications of the MFE method by Mueller and Carey [77, 78] and Herbst *et al.* [79–82] result in a smaller truncation error than that of Miller and Miller but suffer from the same problems of large computational overhead. If explicit time-stepping is acceptable, the simplification of the MFE method due to Dukowicz [76] reduces the size of the matrix system that must be solved. The various types of penalty functions used in the MFE variants above to prevent node crossing might well be of value in other adaptive methods. Elimination of the need for penalty functions or implicit time-stepping by use of the techniques of Wathen and Baines [84–86] also bears further study; however, one must add and remove nodes during the course of the calculation.

If one must add or delete nodes, the method of node adjustment used by Petzold [57] minimizes the impact on accuracy and stability and therefore on allowable time-step size. The approach to node addition and removal favoured by Adjerid and Flaherty [97] is simpler but not as flexible.

The methods of both Dwyer *et al.* [33 39] and Rai and Anderson [90 92] probably consume less computer time than those above. Their behaviour for two-dimensional problems might be improved significantly by application of a measure of grid orthogonality. The coordinate orthogonalization techniques of Potter and Tuttle [40] or of Anyiwo [44] are also of value. The method of Gnoffo [105, 106] also fails to control grid orthogonality in two-dimensional problems and has run into difficulties similar to those experienced by Dwyer *et al.* and Rai and Anderson; the torsion spring interactions introduced by Nakahashi and Deiwert [107,109] seem to have eliminated the difficulties.

The error measures used by Sanz-Serna and Christie [31] and Revilla [32] bear a striking resemblance to those of Dwyer *et al.* [33–39]. The numerical-quadrature-based transformation used by Dwyer *et al.* may be more expensive than the projection technique of the former authors. The adaptive methods of Sanz-Serna and Christie and of Revilla can be extended to two-dimensional problems in a manner similar to that used by Dwyer *et al.*

It is possible to obtain a simpler set of equations in a finite-element formulation by using a separate error measure to control node movement in a manner similar to that used in finite-difference methods. Benner, Davis, and Scriven [87] and Adjerid and Flaherty [97, 98] have begun work in this area but their methods are as yet unperfected.

The work of Lee and Ramos [88, 89] provides a natural alternative to interpolation when the solution is being transferred from the old grid to the new grid in finite-element methods.

The methods of Rai and Anderson [90–92], Greenberg [95], Madsen [96], and Adjerid and Flaherty [97] compute node velocities in response to deviation of an error measure from some average value. This approach inherently results in a proportionality factor for which the optimum value will vary greatly from problem to problem or during the course of a solution. Too small a proportionality factor means that nodes will be unable to follow moving waves, while too large a proportionality factor will cause nodes to outrun the waves or oscillate back and forth from time-step to time-step in the region of the waves. It may not be possible to choose an appropriate proportionality factor in problems containing multiple regions with disparate wave velocities. The methods are probably best suited to computation of the time-asymptotic solutions of fairly simple problems where the deviation of an error measure from some average decreases with time. Note, however, that Adjerid and Flaherty [98] have introduced dynamic specification of the proportionality factor and have obtained improved results.

Causing the nodes to follow the solution characteristics, as done by Kansa *et al.* [42], may create problems when the characteristics intersect, such as during the formation of a shock wave. Auxiliary node-redistribution schemes must be employed to correct the tendency of the nodes to cross. The initial work of Kansa *et al.* included use of penalty functions, introduced by Miller and Miller [64, 65], as a means of preventing node crossing. This procedure was rejected because of a tendency to increase the size of the residue R, which led to greatly reduced time-step sizes. The method of Ghia *et al.* [43] also causes the nodes to follow the solution characteristics, but the tendency of the nodes to cross is reduced by increasing the coefficient of the second-order derivative in the equation defining node movement. A similar "over emphasis of diffusion" strategy contributes to stability in the later work of Miller [74] and an associate, Djomehri [73]. The method of Petzold [57] also causes nodes to follow the characteristics, but the tendency of the nodes to cross is reduced by the use of penalty functions. The penalty functions used in the MFE method of Miller and Miller [64, 65] and its variants would probably be more effective than those of Petzold in preventing node crossing.

Adjerid and Flaherty [97, 98] eliminate an error-measure average from their basic equation for node velocities by using the difference of the equations at adjacent elements. Dorfi and Drury [104] and Winkler *et al.* [99–103] eliminate a constant of proportionality by using equivalent manipulations. The net effect in the three methods is to extend the influence of the value of error measure at a particular node to node movement at adjacent nodes. In the methods of Dorfi and

Drury and of Winkler *et al.* an especially large range of influence of local error measure has resulted in excellent solutions of one-dimensional test problems. In addition, the work of both authors can be adapted to develop alternative penalty functions for methods that need them.

Anyiwo [44] describes his adaptive method sketchily, but his paper contains some useful ideas applicable to the methods of other authors. These ideas include the use of nonzero $\partial X_i/\partial t$ in an alternating-node-movement method and of an explicitly orthogonal transformation between physical and computational coordinates.

The one-dimensional method of Pierson and Kutler [24] is rather expensive since it uses Chebyshev polynomials in the computation of optimal node positions. This method has not been extended to multidimensional problems and exhibits a negligible truncation-error reduction, as compared to the truncation-error reduction obtainable using other methods. Pierson and Kutler, as well as Carcaillet, Dulikravich, and Kennon [59], use a simplex method to position nodes so that a global error measure is minimized. A simplex-method approach to placing nodes has a greater ability to deal with grids having difficult properties, such as a negative Jacobian of the transformation, than does a differential-equation approach. The simplex-method approach may also consume more computer time.

Methods that depend on explicit calculation of PDE truncation error, such as that of Klopfer and McRae [26], are probably awkward to apply for general PDE solvers. However, the truncation error can be estimated indirectly by computing the right-hand side of a PDE twice with different node spacings and taking the difference, as was done by Madsen [96].

All the methods require the specification of weighting constants. Hence, the solution of a problem often involves a cycle of choosing weighting constants, submitting the problem, and observing the behaviour of the solution to determine if the weighting constants should be changed and the problem resubmitted. The reduction of the number of weighting constants or their automatic selection is an area that in the future ought to receive more attention in adaptive techniques.

Much work has yet to be done on developing improved adaptive methods, but the field certainly holds great promise in the reduction of computational costs in many areas of numerical PDE solution.

ACKNOWLEDGMENT

Financial assistance from the Natural Sciences and Engineering Research Council of Canada was very helpful and is gratefully acknowledged.

REFERENCES

1. J. GLIMM, *Commun. Pure Appl. Math.* **18**, 697 (1965).
2. J. F. THOMPSON, Z. U. A. WARSI, AND C. W. MASTIN, *J. Comput. Phys.* **47**, 1 (1982).

3. J. F. THOMPSON, Z. U. A. WARSI, AND C. W. MASTIN, *Numerical Grid Generation* (North-Holland. New York, 1985).
4. J. F. THOMPSON, *AIAA J.* **22**, 1505 (1984).
5. E. TURKEL, *Comput. Fluids* **11**, 121 (1983).
6. D. A. ANDERSON, "Adaptive Grid Methods for Partial Differential Equations," in *Advances in Grid Generation*, edited by K. N. Ghia and U. Ghia (ASME, New York, 1983), p. 1.
7. J. F. THOMPSON, *Appl. Numer. Math.* **1**, 3 (1985).
8. P. R. EISEMAN, *Comput. Meth. Appl. Mech. Eng.* **64**, 321 (1987).
9. J. F. THOMPSON, "General Curvilinear Coordinate Systems," in *Numerical Grid Generation*, edited by J. F. Thompson (North-Holland, Amsterdam, 1982), p. 1.
10. R. J. GELINAS, *J. Comput. Phys.* **9**, 222 (1972).
11. R. COURANT, K. FRIEDRICHS, AND H. LEWY, *IBM J. Res. Devel.* **11**, 215 (1967) (transl. of *Math. Ann.* **100**, 32 (1928)).
12. C. W. GEAR, *Numerical Initial Value Problems in Ordinary Differential Equations* (Prentice–Hall, Englewood Cliffs, NJ, 1971).
13. A. C. HINDMARSH, *GEAR: Ordinary Differential Equation System Solver*, Lawrence Livermore Laboratory Report UCID-30001, Revision 3 (1974).
14. W. H. ENRIGHT, T. E. HULL, AND B. LINDBERG, *BIT Int.* **15**, 10 (1975).
15. D. A. ANDERSON, J. C. TANNEHILL, AND R. H. PLETCHER, *Computational Fluid Mechanics and Heat Transfer* (McGraw–Hill, New York, 1984).
16. R. J. KEE, L. R. PETZOLD, M. D. SMOOKE, AND J. F. GRCAR, "Implicit Methods in Combustion and Chemical Kinetics Modeling," in *Multiple Time Scales*, edited by J. U. Brackbill and B. I. Cohen (Academic Press, London, 1985), p. 113.
17. C. W. MASTIN, "Error Induced by Coordinate Systems," in *Numerical Grid Generation*, edited by J. F. Thompson (North-Holland, Amsterdam, 1982), p. 31.
18. J. F. THOMPSON AND C. W. MASTIN, "Order of Difference Expressions in Curvilinear Coordinate Systems," in *Advances in Grid Generation*, edited by K. N. Ghia and U. Ghia (ASME, New York, 1983), p. 17.
19. E. KALNAY DE RIVAS, *J. Comput. Phys.* **10**, 202 (1972).
20. J. D. HOFFMAN, *J. Comput. Phys.* **46**, 469 (1982).
21. R. G. HINDMAN, *AIAA J.* **20**, 1359 (1982).
22. T. DUPONT, *Math. Comput.* **39**, 85 (1982).
23. D. O. GOUGH, E. A. SPIEGEL, AND J. TOOMRE, *Lecture Notes in Physics, Vol.* 35 (Springer-Verlag, New York/Berlin, 1975), p. 191.
24. B. L. PIERSON AND P. KUTLER, *AIAA J.* **18**, 49 (1980).
25. V. E. DENNY AND R. B. LANDIS, *J. Comput. Phys.* **9**, 120 (1972).
26. G. H. KLOPFER AND D. S. MCRAE, "The Nonlinear Modified Equation Approach to Analyzing Finite Difference Schemes," AIAA Paper 81-1029 (1981), p. 317.
27. R. W. MACCORMACK, "The Effect of Viscosity in Hypervelocity Impact Cratering," AIAA Paper 69-354 (1969), p. 1.
28. A. B. WHITE, JR., *SIAM J. Numer. Anal.* **16**, 472 (1979).
29. A. B. WHITE, JR., *SIAM J. Numer. Anal.* **19**, 683 (1982).
30. C. M. ABLOW AND S. SCHECHTER, *J. Comput. Phys.* **27**, 351 (1978).
31. J. M. SANZ-SERNA AND I. CHRISTIE, *J. Comput. Phys.* **67**, 348 (1986).
32. M. A. REVILLA, *Int. J. Numer. Meth. Eng.* **23**, 2263 (1986).
33. H. A. DWYER, R. J. KEE, AND B. R. SANDERS, *AIAA J.* **18**, 1205 (1980).
34. H. A. DWYER, F. RAISZADEH, AND G. OTEY, *Lecture Notes in Physics, Vol.* 141 (Springer-Verlag, New York/Berlin, 1981), p. 170.
35. H. A. DWYER, M. D. SMOOKE, AND R. J. KEE, "Adaptive Gridding for Finite Difference Solutions to Heat and Mass Transfer Problems," in *Numerical Grid Generation*, edited by J. F. Thompson (North-Holland, Amsterdam, 1982), p. 339.
36. H. A. DWYER, R. J. KEE, P. K. BARR, AND B. R. SANDERS, *J. Fluids Eng.* **105**, 83 (1983).

37. H. A. DWYER, B. R. SANDERS, AND F. RAISZADEH, *Combust. Flame* **52**, 11 (1983).
38. H. A. DWYER, *AIAA J.* **22**, 1705 (1984).
39. H. A. DWYER, "A Discussion of Some Criteria for the Use of Adaptive Gridding," in *Adaptive Computational Methods for Partial Differential Equations*, edited by I. Babuska, J. Chandra, and J. E. Flaherty (SIAM, Philadelphia, 1983), p. 111.
40. D. E. POTTER AND G. H. TUTTLE, *J. Comput. Phys.* **13**, 483 (1973).
41. H. A. DWYER AND O. O. ONYEJEKWE, *Lecture Notes in Physics, Vol.* 218 (Springer-Verlag, New York/Berlin, 1985), p. 203.
42. E. J. KANSA, D. L. MORGAN, JR., AND L. K. MORRIS, *SIAM J. Sci. Stat. Comput.* **5**, 667 (1984).
43. K. N. GHIA, U. GHIA, AND C. T. SHIN, "Adaptive Grid Generation for Flows with Local High Gradient Regions," in *Advances in Grid Generation*, edited by K. N. Ghia and U. Ghia (ASME, New York, 1983), p. 35.
44. J. C. ANYIWO, "Idealized Dynamic Grid Computation of Physical Systems," in *Numerical Grid Generation*, edited by J. F. Thompson (North-Holland, Amsterdam, 1982), p. 837.
45. J. U. BRACKBILL AND J. SALTZMAN, "An Adaptive Computation Mesh for the Solution of Singular Perturbation Problems," in *Numerical Grid Generation Techniques, Proceedings of the Numerical Grid Generation Workshop at Langley Research Center*, edited by R. E. Smith (NASA CP-2166, Washington, DC, 1980), p. 193.
46. J. U. BRACKBILL AND J. S. SALTZMAN, *J. Comput. Phys.* **46**, 342 (1982).
47. J. U. BRACKBILL, "Coordinate System Control: Adaptive Meshes," in *Numerical Grid Generation*, edited by J. F. Thompson (North-Holland, Amsterdam, 1982), p. 277.
48. J. SALTZMAN AND J. BRACKBILL, "Applications and Generalizations of Variational Methods for Generating Adaptive Meshes," in *Numerical Grid Generation*, edited by J. F. Thompson (North-Holland, Amsterdam, 1982), p. 865.
49. A. M. WINSLOW, *J. Comput. Phys.* **2**, 149 (1967).
50. D. A. ANDERSON, "Constructing Adaptive Grids with Poisson Grid Generators," in *Numerical Grid Generation in Computational Fluid Dynamics*, edited by J. Hauser and C. Taylor (Pineridge Press, Swansea, UK, 1986), p. 125.
51. R. I. KREIS, F. C. THAMES, AND H. A. HASSAN, *AIAA J.* **24**, 404 (1986).
52. N. N. YANENKO, V. M. KOVENYA, V. V. LISEIKIN, V. M. FOMIN, AND E. V. VOROZHTSOV, *Comput. Meth. Appl. Mech. Eng.* **17/18**, 659 (1979).
53. N. N. YANENKO, E. A. KROSHKO, V. V. LISEIKIN, V. M. FOMIN, V. P. SHAPEEV, AND YU. A. SHITOV, *Lecture Notes in Physics, Vol.* 59 (Springer-Verlag, New York/Berlin, 1976), p. 454.
54. N. N. YANENKO, V. V. LISEIKIN, AND V. M. KOVENYA, *Lecture Notes in Physics, Vol.* 91 (Springer-Verlag, New York/Berlin, 1979), p. 48.
55. R. G. HINDMAN AND J. SPENCER, "A New Approach to Truly Adaptive Grid Generation," AIAA Paper 83-0450 (1983), p. 1.
56. R. G. HINDMAN, P. KUTLER, AND D. ANDERSON, *AIAA J.* **19**, 424 (1981).
57. L. R. PETZOLD, *Appl. Numer. Math.* **3**, 347 (1987).
58. J. M. HYMAN, "Adaptive Moving Mesh Methods for Partial Differential Equations," Los Alamos National Laboratory Report LA-UR-82-3690 (Los Alamos, New Mexico, 1982) (unpublished).
59. R. CARCAILLET, G. S. DULIKRAVICH, AND S. R. KENNON, *Comput. Meth. Appl. Mech. Eng.* **57**, 279 (1986).
60. L. J. HAYES, S. R. KENNON, AND G. S. DULIKRAVICH, *Comput. Meth. Appl. Mech. Eng.* **59**, 141 (1986).
61. S. R. KENNON AND G. S. DULIKRAVICH, *AIAA J.* **24**, 1069 (1986).
62. R. CARCAILLET, S. R. KENNON, AND G. S. DULIKRAVICH, *J. Aircraft* **23**, 415 (1986).
63. S. R. KENNON AND G. S. DULIKRAVICH, "Composite Computational Grid Generation Using Optimization," in *Numerical Grid Generation in Computational Fluid Dynamics*, edited by J. Hauser and C. Taylor (Pineridge Press, Swansea, UK, 1986), p. 217.
64. K. MILLER AND R. N. MILLER, *SIAM J. Numer. Anal.* **18**, 1019 (1981).
65. K. MILLER, *SIAM J. Numer. Anal.* **18**, 1033 (1981).

66. J. DJOMEHRI AND K. MILLER, "A Moving Finite Element Code for General Systems of PDE's in 2-D," Technical Report PAM-57, Center for Pure and Applied Mathematics, University of California, Berkeley, 1981 (unpublished).
67. R. J. GELINAS, S. K. DOSS, AND K. MILLER, *J. Comput. Phys.* **40**, 202 (1981).
68. R. J. GELINAS, S. K. DOSS, J. P. VAJK, J. DJOMEHRI, AND K. MILLER, in *10th IMACS World Congress on System Simulation and Scientific Computation, Montreal*, 1982, p. 58.
69. R. J. GELINAS AND S. K. DOSS, "The Moving Finite Element Method: A Semi-Automatic Solver for Diverse PDE Applications," preprint for Fourth IMACS International Symposium on Computer Methods for Partial Differential Equations, Lehigh University, Bethlehem, Pennsylvania, June 30–July 2, 1981 (unpublished).
70. R. J. GELINAS AND S. K. DOSS, in *10th IMACS World Congress on System Simulation and Scientific Computation, Montreal*, 1982, p. 156.
71. R. J. GELINAS AND S. K. DOSS, *Lecture Notes in Engineering, Vol.* 3 (Springer-Verlag, New York/Berlin, 1983), p. 192.
72. R. J. GELINAS, S. K. DOSS, AND N. N. CARLSON, "Moving Finite Element Research, Physical versus Non-Physical Dissipation Effects," in Semi-Annual DNA Airblast Review Meeting held at R and D Associates, Marina del Rey, California, 1982 (unpublished).
73. J. DJOMEHRI, "Moving Finite Element Solution of Systems of Partial Differential Equations in 1-Dimension," Technical Report PAM-125, Center for Pure and Applied Mathematics, University of California, Berkeley, 1983 (unpublished).
74. K. MILLER, "Alternate Modes to Control the Nodes in the Moving Finite Element Method," in *Adaptive Computational Methods for Partial Differential Equations*, edited by I. Babuska, J. Chandra, and J. E. Flaherty (SIAM, Philadelphia, 1983), p. 165.
75. K. MILLER, "Recent Results on Finite Element Methods with Moving Nodes," in *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, edited by I. Babuska, O. C. Zienkiewicz, J. Gago, and E. R. de A. Oliveira (Wiley, Chichester, 1986), p. 325.
76. J. K. DUKOWICZ, *J. Comput. Phys.* **56**, 324 (1984).
77. A. C. MUELLER AND G. F. CAREY, in *10th IMACS World Congress on System Simulation and Scientific Computation, Montreal*, 1982, p. 37.
78. A. C. MUELLER AND G. F. CAREY, "Continuously Deforming Finite Elements for Convection Dominated Flows," in *Advances in Grid Generation*, edited by K. N. Ghia and U. Ghia (ASME, New York, 1983), p. 135.
79. B. M. HERBST, S. W. SCHOOMBIE, AND A. R. MITCHELL, *Int. J. Numer. Meth. Eng.* **18**, 1321 (1982).
80. B. M. HERBST, S. W. SCHOOMBIE, AND A. R. MITCHELL, *J. Comput. Appl. Math.* **9**, 377 (1983).
81. B. M. HERBST AND S. W. SCHOOMBIE, "Parabolic Equations with Dominating Convection Terms," in *Numerical Solution of PDE's: Theory, Tools, and Case Studies, 1st International Series of Numerical Mathematics*, edited by D. P. Laurie (Birkhauser Verlag, Basel, 1983), Vol. 66, p. 185.
82. B. M. HERBST, S. W. SCHOOMBIE, D. F. GRIFFITHS, AND A. R. MITCHELL, *Int. J. Numer. Meth. Eng.* **20**, 1273 (1984).
83. A. R. MITCHELL AND B. M. HERBST, "Adaptive Grids in Petrov–Galerkin Computations," in *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, edited by I. Babuska, O. C. Zienkiewicz, J. Gago, and E. R. de A. Oliveira (Wiley, Chichester, 1986), p. 315.
84. A. J. WATHEN AND M. J. BAINES, *IMA J. Numer. Anal.* **5**, 161 (1985).
85. M. J. BAINES AND A. J. WATHEN, *Appl. Numer. Math.* **2**, 495 (1986).
86. A. J. Wathen, *SIAM J. Numer. Anal.* **23**, 797 (1986).
87. R. E. BENNER, JR., H. T. DAVIS, AND L. E. SCRIVEN, *SIAM J. Sci. Stat. Comput.* **8**, 529 (1987).
88. D. N. LEE AND J. I. RAMOS, *AIAA J.* **21**, 262 (1983).
89. J. I. RAMOS, *Int. J. Numer. Meth. Fluids* **5**, 13 (1985).
90. M. M. RAI AND D. A. ANDERSON, "Grid Evolution in Time Asymptotic Problems," in *Numerical Grid Generation Techniques, Proceedings of the Numerical Grid Generation Workshop at Langley Research Center*, edited by R. E. Smith (NASA, CP-2166, Washington, DC, 1980), p. 409.
91. M. M. RAI AND D. A. ANDERSON, *AIAA J.* **20**, 496 (1982).

92. D. A. Anderson and M. M. Rai, "The Use of Solution Adaptive Grids in Solving Partial Differential Equations," in *Numerical Grid Generation*, edited by J. F. Thompson (North-Holland, Amsterdam, 1982), p. 317.
93. D. A. Anderson, "Adaptive Mesh Schemes Based on Grid Speeds," AIAA Paper 83-1931 (1983), p. 311 (unpublished).
94. D. A. Anderson, "Application of Adaptive Grids to Transient Problems," in *Adaptive Computational Methods for Partial Differential Equations*, edited by I. Babuska, J. Chandra, and J. E. Flaherty (SIAM, Philadelphia, 1983), p. 208.
95. J. B. Greenberg, *AIAA J.* **23**, 317 (1985).
96. N. K. Madsen, "Molag: A Method of Lines Adaptive Grid Interface for Nonlinear Partial Differential Equations," in *PDE Software: Modules, Interfaces and Systems*, edited by B. Engquist and T. Smedsaas (North-Holland, Amsterdam, 1984), p. 207.
97. S. Adjerid and J. E. Flaherty, *SIAM J. Numer. Anal.* **23**, 778 (1986).
98. S. Adjerid and J. E. Flaherty, *Comput. Meth. Appl. Mech. Eng.* **55**, 3 (1986).
99. K.-H. A. Winkler, M. L. Norman, and M. J. Newman, *Phys. D* **12**, 408 (1984).
100. W. M. Tscharnuter and K.-H. A. Winkler, *Comput. Phys. Comm.* **18**, 171 (1979).
101. K.-H. A. Winkler and M. J. Newman, *Astrophys. J.* **238**, 311 (1980).
102. K.-H. A. Winkler, D. Mihalas, and M. L. Norman, *Comput. Phys. Comm.* **36**, 121 (1985).
103. K.-H. A. Winkler, M. L. Norman, and D. Mihalas, "Implicit Adaptive-Grid Radiation Hydrodynamics," in *Multiple Time Scales*, edited by J. U. Brackbill and B. I. Cohen (Academic Press, London, 1985), p. 145.
104. E. A. Dorfi and L. O'C. Drury, *J. Comput. Phys.* **69**, 175 (1987).
105. P. A. Gnoffo, "A Vectorized, Finite-Volume, Adaptive-Grid Algorithm for Navier–Stokes Calculations," in *Numerical Grid Generation*, edited by J. F. Thompson (North-Holland, Amsterdam, 1982), p. 819.
106. P. A. Gnoffo, *AIAA J.* **21**, 1249 (1983).
107. K. Nakahashi and G. S. Deiwert, *Lecture Notes in Physics, Vol.* 218 (Springer-Verlag, New York/Berlin, 1985), p. 422.
108. G. S. Deiwert, A. E. Andrews, and K. Nakahashi, *J. Spacecraft* **24**, 496 (1987).
109. K. Nakahashi and G. S. Deiwert, *AIAA J.* **24**, 948 (1986).
110. P. R. Eiseman, *J. Fluids Eng.* **107**, 477 (1985).
111. P. R. Eiseman, *Lecture Notes in Physics, Vol.* 238 (Springer-Verlag, New York/Berlin, 1985), p. 205.
112. P. R. Eiseman, *AIAA J.* **23**, 551 (1985).
113. P. R. Eiseman, *Lecture Notes in Physics, Vol.* 264 (Springer-Verlag, New York/Berlin, 1986), p. 258.
114. S. F. Davis and J. E. Flaherty, *SIAM J. Sci. Stat. Comput.* **3**, 6 (1982).
115. J. E. Flaherty, J. M. Coyle, R. Ludwig, and S. F. Davis, "Adaptive Finite Element Methods for Parabolic Partial Differential Equations," in *Adaptive Computational Methods for Partial Differential Equations*, edited by I. Babuska, J. Chandra, and J. E. Flaherty (SIAM, Philadelphia, 1983), p. 144.
116. J. M. Coyle, J. E. Flaherty, and R. Ludwig, *J. Comput. Phys.* **62**, 26 (1986).