INSTITUTE

FOR

AEROSPACE STUDIES

UNIVERSITY OF TORONTO

# REVIEW OF
# ADAPTIVE-GRID TECHNIQUES FOR SOLUTION OF
# PARTIAL DIFFERENTIAL EQUATIONS

by

D. F. Hawken

December 1985

UTIAS Review No. 46
CN ISSN 0082-5247

# REVIEW OF
# ADAPTIVE-GRID TECHNIQUES FOR SOLUTION OF
# PARTIAL DIFFERENTIAL EQUATIONS

by

D. F. Hawken

Submitted May 1985

December 1985

UTIAS Review No. 46
CN ISSN 0082-5247

# ACKNOWLEDGEMENTS

I wish to thank Professors J. J. Gottlieb and J. S. Hansen for their encouragement in the writing of this report. My gratitude also goes to my wife, Jill, for all her help in proofreading the various drafts.

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

## SUMMARY

This report 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 blast-wave or other nonstationary flow problems that contain moving regions of rapid change in the flow variables, surrounded by regions of relatively smooth variation. Shock waves, contact surfaces, phase-change interfaces, boundary layers, and other structures, can be modelled in detail 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.

## CONTENTS

## 1.0 INTRODUCTION

Numerical methods for solving partial differential equations (PDE's) which are based on finite-difference or finite-element formulations achieve significant economies of execution if nodes can be moved so that they remain concentrated in regions of rapid variation of the solution. This report focuses on such methods. 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 since such methods are of little 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 [1], Thompson [2], and Turkel [3]. Body-fitted coordinate systems are also discussed by these authors and are not covered in this report since emphasis will be placed on methods that move the nodes so as to adapt to solution characteristics rather than boundary characteristics.

Most of the methods discussed in this report require that the PDE's have continuous solutions. If, as is usually the case, the PDE's are nonlinear, 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.

As pointed out by Thompson [2] 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 their distribution remains fairly regular as the nodes 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 following sections.

## 2.0 OVERVIEW OF ADAPTIVE METHODS

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 differential equation

$$\varepsilon \frac{d^2P}{dX^2} + P \left[\frac{dP}{dX} - 1\right] = 0 \qquad (1)$$

on the domain $0 \leq X \leq 1$ where $\varepsilon$ is a small positive number and $P(0) = A$ and $P(1) = B$ are the boundary conditions. It is easily seen that if $dP/dX = 1$, then the slope will tend to remain constant at unity, since the second derivative is then zero. However, except for very special values of A and B, the slope must eventually depart from unity. For arbitrary values of A and B the solution will consist of one or two regions with dP/dX = 1, bordered by region(s) of rapid solution variation so as to match the boundary conditions. The region(s) of rapid variation will take the form of one or two boundary layers or a more centrally located stationary shock layer. The locations of these regions can be predicted only by the use of extensive analysis.

Figure 1 illustrates the results of an unsuccessful attempt to solve equation (1) with A = 4 and B = 2, using a nonadaptive finite-element method with N, fixed, equally spaced nodes. P was approximated using piecewise, linear basis functions, and the method of Galerkin was applied to equation (1). The value of $\varepsilon$ was initially set to 0.1, and the solution approximated by a linear ramp with $P^1 = A$ and $P^N = B$. The nodal values $P^n$ were adjusted using a Newton's method iteration so as to minimize the Galerkin residual. The value of $\varepsilon$ was reduced whenever the residual fell below a preset threshold. The region of rapid variation of the solution is of the order of $\varepsilon$. Hence, as the value of $\varepsilon$ was reduced, there were eventually not enough nodes in this region to properly represent the solution, and a large truncation error was obtained. Thus, even though 171 nodes were used, the solution with $\varepsilon = 0.0015$ shown in figure 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; otherwise one will obtain the numerical solution of an ODE with extra terms in it. If odd-order derivatives predominate in the truncation error (known as numerical dispersion), the extra terms cause the new ODE to have an oscillatory behaviour (Gibb's oscillations) that the proper ODE did not possess. If even-order derivatives predominate in the truncation error (known as numerical viscosity or numerical diffusion), excessively thick wave fronts will result. Many nonadaptive methods add an even-order artificial-viscosity term in regions of rapid solution variation, so as to dominate odd-order truncation-error terms, and hence avoid Gibb's oscillations, but this also results in a physically invalid broadening of wave 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-element technique, equation (1) was solved with the same boundary conditions as above, but employing only 41 nodes. Figures 2, 3, and 4 illustrate solutions for $\varepsilon = 0.0015$, $\varepsilon = 0.001$, and $\varepsilon = 0.00001$, respectively. Only the region of the boundary layer is plotted in the last two figures, in order to adequately resolve the behaviour of the solution. Note the complete absence of Gibb's oscillations for values of $\varepsilon$ 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 regions 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 the truncation error. An adaptive algorithm was devised in which the node separation was gradually increased in regions of near-unit slope, with the excess nodes being equidistributed in the other regions. 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.

Most adaptive methods transform the PDE's for the physical variables (denoted by vector $\underline{P}$) from physical coordinates (herein written $(X_i,t)$) for an $i^{th}$ dimensional physical space to $i^{th}$ dimensional computational coordinates (herein written $(\xi_i,t)$) in which the nodes are equally spaced. An excellent account of such transformations can be found in Thompson [4]. 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 defined by $dS = [1 + [d\underline{P}/dX]^2]^{1/2}dX$.

$\underline{P}(\xi_i,t)$ and $X_j(\xi_i,t)$ are the unknowns in the computational coordinate system. The transformed PDE's contain additional terms in $\partial X_i/\partial t$ at fixed computational coordinate. Each PDE for $\underline{P}$ is reduced to a set of ordinary differential equations in time (one ODE for each node n) by application of a discretization method such as the finite-element or finite-difference formulations. The ODE's can be written in the form

$$\frac{d\underline{P}^n}{dt} = \underline{F}^n(t) \ . \tag{2}$$

$\underline{P}^n$ is the value of the solution at node n, and the details of the computation of $\underline{F}^n$ are determined by the particular discretizing method used to reduce the PDE's to ODE's. Particular attention must be paid to the initial placement of the nodes, so that spacial truncation error will not cause inaccurate initial estimates of $d\underline{P}^n/dt$.

The ODE's can be solved employing a time-stepping finite-difference method. The method may be time explicit, in which case $\underline{F}^n(t)$ is evaluated at the current time level, or time implicit, in which case the value of $\underline{F}^n(t)$ 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 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 $\underline{P}^n$ at each node, one can compute the value of $\underline{P}^n$ at the next time step. Since $d\underline{P}^n/dt$ can vary markedly from node to node (consider shock propagation through a system of nodes) the system of ODE's is often 'stiff' [5]. A stiff system is characterized by the fact that most 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 drastic difference in ODE-solution time constant from node to node causes some of the numerical difficulty. Another cause for the stiffness is the fact that nodes approach one another very closely in regions of large solution variation, thus causing Courant limitation of time step if an explicit method is used. In order to obtain reasonably large time steps, the system of ODE's must usually be solved by implicit methods such as the backward-differentiation algorithm of Gear [6]. A version of the Gear method has been implemented by Hindmarsh and Gelinas [7], and this in modified form is available as an IMSL subroutine. Enright, Hull, and Lindberg [8] discuss and compare a number of stiff-ODE solution methods.

A measure of the spacial truncation error is computed at intervals in the time-stepping solution of the PDE's. The measure is usually some grid-derivative function of $X_i$ and/or $\underline{P}$ 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 to the creation of long-range internodal pseudoforces. In addition to taking error measures, some workers use measures of transformation smoothness, orthogonality, or other transformation property to control the quality of the node distribution.

A bewildering variety of error measures is used in the literature, including those of change in the solution or its derivatives of various orders from node to node; circular curvatures or torsions along lines of constant curvilinear coordinate; solution slope lengths; and complex combinations of all the above. In some cases the error measure is related to truncation-error expressions; however, in many cases the authors have chosen their error measures heuristically and provide little or no justification for them. The fact that so many different measures are used successfully means that the problem of adaptation is not critical. In one dimension a finite-difference method can be successful if it merely places many equally spaced and equidistributed 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. An analysis of truncation error induced by grid deformation in finite-difference formulations may be found in Mastin [9], Kalnay de Rivas [10], and Hindman [11]. Dupont [12] 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 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 $\underline{P}^n$ 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 ODE's are obtained, and the values of $\underline{P}^n$ 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 $X_i^n$ at each node. The nodes are not shifted between time steps; rather, a set of ODE's for $X_i^n$ is solved along with the ODE's for the physical solution on each time step, in order to simultaneously compute the new values of $\underline{P}^n$ and $X_i^n$ in the next time step. Interpolation of $\underline{P}$ 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 PDE's for $\underline{P}$ in computational coordinates are usually set to zero when the

PDE's 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 [16] and Anyiwo [29], so as to allow node movement to effect the solution at each node.

Dwyer, Smooke, and Kee [22] 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 due to mismatching of node distribution and solution, unless time steps are kept very small.

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 which follows.

## 3.0 SUMMARY OF VARIOUS ADAPTIVE METHODS

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 included in the discussion below.

### 3.1 Methods Based on Integral Minimization or Equidistribution of Error Measure

Gough, Spiegel, and Toomre [13] 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 $\xi$ 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 $m^{th}$ derivative of $P_k$ and X with respect to $\xi$ was computed using

$$E_m = \sum_{k=1}^{K} \frac{1}{R_k}\left|\frac{d^m P_k}{d\xi^m}\right|^2 + \frac{\lambda}{X_f - X_o}\left|\frac{d^m X}{d\xi^m}\right|^2 , \qquad (3)$$

where $P_k$ is the $k^{th}$ component of the solution, and $R_k$ and $X_f - X_o$ are the oscillations in the $\xi$ domain of $P_k$ and X, respectively. The weighting constant $\lambda$ was usually set to K. In practice only $E_1$ and $E_2$ have been used as measures. The transformation from X to $\xi$ was computed by solving an ODE that was obtained by minimizing the integral of $E_m$ over the $\xi$ domain.

Node movement was alternated with the solution of the ODE for $P_k$ so as to obtain convergence to a final, highly adaptive, and accurate solution. This alternating-node-movement procedure was claimed to be faster and more stable than solving ODE's for $P_k$ 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 [14] 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 ODE's 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 coefficients 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 [15] 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 differentiated with respect to the nodal coordinate in order to obtain a finite-difference formula for the nodal position, which 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. [13].

Klopfer and McRae [16] 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 ∂X/∂ξ was a linear function of the smoothed error measure E such that

$$\frac{\partial X}{\partial \xi} \propto 1 - \frac{E}{E_{max}} \quad . \tag{4}$$

The node-spacing distribution was integrated to obtain an 'optimal' node distribution. The node-movement process was repeated after each time step. The time derivatives of the nodal coordinates $X^n$ were computed from the changes in nodal position and included in the PDE's for the solution $\underline{P}$. Artificial viscosity was used in the solution algorithm to stabilize the calculation. Since the explicit predictor-corrector scheme of MacCormack [17] 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 [18] solved a two-point, vector-valued boundary-value problem. The ODE for $\underline{P}$ 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

$$\xi(X) \quad = \int_{X_o}^{X} \frac{E(x)dx}{\theta} \quad , \tag{5}$$

where

$$\theta = \int_{X_o}^{X_f} E(x)dx . \tag{6}$$

Thus, the nodes were equidistributed in the $\xi$ domain. The integral was manipulated to obtain ODE's for X and $\theta$ in the $\xi$ frame. The ODE's for $\underline{P}$, X, and $\theta$ in the $\xi$ 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 $dS = [1 + (d\underline{P}/dX)^2]dX$, local truncation error of the ODE, and weighted node-to-node change in the solution.

White [19] extended the method for use in solving time-dependent problems. The ODE's for X and $\theta$ in the $(\xi,t)$ frame were written as

$$\left|\frac{\partial X}{\partial \xi}\right|^2 + \left|\frac{\partial \underline{P}}{\partial \xi}\right|^2 - \theta^2 = 0 \tag{7}$$

and

$$\frac{\partial \theta}{\partial \xi} = 0 \tag{8}$$

and then solved along with transformed PDE's for $\underline{P}$ time step by time step. Variables $\underline{P}$, X, and $\theta$ were written as an average of their values at the present and next time steps, and the derivatives with respect to $\xi$ 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 [18] was taken by Ablow and Shechter [20], who used the error measure

$$E = 1 + \lambda|\Omega| , \tag{9}$$

where $\lambda$ is a weighting constant and $\Omega$ is the circular curvature of the solution given by

$$\Omega = \frac{\dfrac{d^2P}{dX^2}}{\left[1 + \left[\dfrac{dP}{dX}\right]^2\right]^{3/2}} . \tag{10}$$

The authors point out that, as a general rule, P 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 $\xi$ having equidistributed error measure and nodes. A second ODE was obtained by differentiation of the equidistribution formula

$$E^2[dX^2 + dP^2] = d\xi^2 \tag{11}$$

with respect to $\xi$. The two ODE's were solved simultaneously in a manner similar to that of White [18].

In order to solve a time-dependent PDE, Dwyer, Kee, and Sanders [21], Dwyer, Smooke, and Kee [22], Dwyer, Kee, Barr, and Sanders [23], Dwyer, Sanders, and Raiszadek [24], and Dwyer [25,26] used a transformation from the spacial coordinate system $(X_1,X_2,t)$ to a coordinate system $(\xi_1,\xi_2,t)$ 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(S) is used involving derivatives with respect to arc length S along curves of constant $\xi_1$ or $\xi_2$ in physical space ($dS^2 = [dX_1]^2 + [dX_2]^2$):

$$E(S) = 1 + \lambda_1 \left|\frac{\partial P}{\partial S}\right| + \lambda_2 \left|\frac{\partial^2 P}{\partial S^2}\right| \quad . \qquad (12)$$

Weighting factors $\lambda_1$ or $\lambda_2$ can be made large in order to equidistribute nodal differences in P or nodal differences in $\partial P/\partial S$, respectively.

The PDE was transformed from the physical coordinates $(X_1,X_2,t)$ to computational coordinates $(\xi_1,\xi_2,t)$ and solved for P using an alternating-direct-implicit finite-difference method (an implicit Newton 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 $\xi_2$ in physical space was

$$\xi_1(X_1,X_2,t) = \frac{\int_o^S E(s)ds}{\int_o^{S_{max}} E(s)ds} \quad . \qquad (13)$$

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 (13) was then solved for equally spaced values of $\xi_1$ 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 $\xi_2$ but Dwyer et al. [21-26] 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. [21], solution oscillations occurred as portions of the grid developed increased skewness in physical space. The authors [21] have suggested that the procedure of Potter and Tuttle [27] might be utilized to reduce the computed grid skewness. Potter and Tuttle move nodes along lines of constant $\xi_1$ in order to define lines of constant $\xi_2$ on which the solution of Laplace's equation is constant. The curvilinear coordinate system produced is orthogonal. Dwyer et al. [21] also encountered problems when adaptation was permitted along boundaries. Coordinate-system collapse occurred in a heat-transfer problem when Neumann boundary conditions were used.

Kansa, Morgan, and Morris [28] developed a method in which the nodes are moved so that the PDE's, when transformed to a numerical frame, have a minimal dependence on spacial gradients. The transformed K-component system of PDE's was written in the form

$$\frac{\partial P_k}{\partial t} = G_k - \frac{\partial F_k}{\partial X} + \frac{\partial X}{\partial t}\frac{\partial P_k}{\partial X} , \tag{14}$$

where $P_k$ is one of the components of the solution; $G_k$ is a function of the solution components and the spacial coordinate X; and $F_k$ is a function of X, the solution components, and gradients of the solution components. The last term in equation (14) results from the transformation from spacial coordinates to numerical coordinates where the nodes are stationary and equidistributed. One can define a characteristic velocity $V_k$ for each component, which is given below:

$$V_k = \frac{\dfrac{\partial F_k}{\partial X}}{\dfrac{\partial P_k}{\partial X}} . \tag{15}$$

If the nodes are moved so that $\partial X/\partial t = V_k$, then the PDE for component k is reduced to an ODE in time of the form

$$\frac{\partial P_k}{\partial t} = G_k . \tag{16}$$

At a shock wave, equation (15) is a restatement of the Rankine-Hugoniot jump conditions, with $V_k$ being the shock velocity. In a rarefaction fan, $V_k$ is the velocity of the characteristics. In a multicomponent problem, $V_k$ will not, in general, be the same for all components. Kansa et al. [28] chose the node velocities $\partial X/\partial t$ to minimize

$$E = \sum_{k=1}^{K} \left[ \frac{\partial X}{\partial t} - V_k \right]^2 \tag{17}$$

approximated by using a three-point-collocation polynomial to compute gradients. The result was a tridiagonal-matrix equation for the node velocities. An implicit finite-difference method that incorporated artificial viscosity terms was used time step by time step, alternately with the computation of node speeds. 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 equidistribute 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 [61]. Kansa et al. [28] used their method, with 21 nodes, to solve a gas diffusion problem. It was claimed 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 is also discussed.

Anyiwo [29] 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 $(\xi_i,t)$ having equidistributed nodes. Anyiwo used an error measure E defined by

$$E = \exp(\gamma) = \exp\left[ \sum_{i=1}^{I} \gamma_i \right] \qquad (18)$$

or

$$E_i = \exp(\gamma_i) \quad . \qquad (19)$$

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

$$\gamma_i = \lambda^1 \alpha^1_i \Omega_i + \lambda^2 \left| \alpha^2_i + \beta^2_i \right| \Omega_i + \lambda^3 \sigma^3 \Omega_i \quad . \qquad (20)$$

$\lambda^1$, $\lambda^2$, and $\lambda^3$ are weighting constants. $\alpha^1_i = 1/N_i$ and $\Omega_i = 1/\text{logn}(N_i-1)$, where $N_i$ is the number of nodes in direction i in $(\xi_i,t)$ coordinates. $\alpha^2_i$ and $\beta^2_i$ are the circular curvature and torsion of curves of constant $S_i$ in physical coordinates and

$$\sigma^3 = \left| \delta^3_i \right| + \left| \alpha^3_i + \beta^3_i \right| \quad . \qquad (21)$$

$\delta^3_i$, $\alpha^3_i$, and $\beta^3_i$ are the slope, circular curvature, and torsion of the physical variable P in $(S_i,t)$ coordinates. $\sigma^3$ is replaced by a weighted sum of terms if P is a multicomponent physical variable.

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

$$\frac{\partial S_i}{\partial \xi_i} = \frac{\int_c E_i(s_i)ds_i}{N_i\, E_i(S_i)} \qquad (22)$$

which in essence equidistributes E over the nodes. $\partial S_i/\partial \xi_i$ 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

$$dS_i = \cos(\theta_{ij})dX_j \qquad (23)$$

where $\theta_{ij}$, 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 $\theta_{ij}$ 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 $\theta_{ij}$ in the interior of the physical domain. $\theta_{ij}$ is computed in response to the variation of $\partial S_i/\partial \xi_i$ 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 $(\xi_i,t)$ coordinate transformation. The values of $dS_i/\partial \xi_i$ are in turn used to determine the new values of $\theta_{ij}$ and hence of the $(X_i,t)$ to $(S_i,t)$ coordinate transformation. The PDE's for the physical solution are transformed to the $(\xi_i,t)$ 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. Anyiwo gave no description of the method used to transfer solution information from an old grid to a new grid.

An additional means of control of adaptation was obtained by inserting a nonzero value for $\partial X_i/\partial t$ in the the transformed PDE's for the physical variable. This value was obtained by requiring that the spacial coordinates satisfied a conservation equation

$$\frac{\partial X_i}{\partial t} = \nabla[ - c\, U\, X_i + \mu\, \nabla X_i] \qquad (24)$$

in the $(\xi_i,t)$ frame, where c is a positive constant less than or equal to one. $\mu$ is the viscosity coefficient, and U the advective velocity for the physical solution PDE's. It is claimed that equation (24) adds a constraint on node movement that maintains transformation smoothness.

Brackbill and Saltzman [30,31], Brackbill [32], and Saltzman and Brackbill [33], extending a boundary-adaptive automatic-grid generation method used by Winslow [34], transformed a two-dimensional PDE for P from physical coordinates $(X_1,X_2,t)$ to numerical coordinates $(\xi_1,\xi_2,t)$. A weighted sum of three error measures was used to determine the coordinate transformation. That is,

$$E = \lambda_s E_s + \lambda_o E_o + \lambda_v E_v \qquad (25)$$

where

$$E_s = [\nabla \xi_1]^2 + [\nabla \xi_2]^2 \qquad (26)$$

is a measure of transformation smoothness,

$$E_o = [\nabla \xi_1 \cdot \nabla \xi_1]^2 \qquad (27)$$

is a measure of grid orthogonality in physical space, or

$$E_o = [\nabla \xi_1 \cdot \nabla \xi_2]^2\, J^3 \qquad (28)$$

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

$$J = \frac{\partial X_1}{\partial \xi_1}\frac{\partial X_2}{\partial \xi_2} - \frac{\partial X_1}{\partial \xi_2}\frac{\partial X_2}{\partial \xi_1} \quad . \tag{29}$$

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

$$E_v = W J \quad , \tag{30}$$

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

$$W = \left|\frac{\nabla P}{P}\right|^Q \tag{31}$$

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 $\nabla P$. 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 $\lambda_s$, $\lambda_o$, and $\lambda_v$ were generally of the order of unity.

The authors obtained an Euler equation for the spacial coordinates by minimizing the spacial integral of E transformed to numerical coordinates. Brackbill [32] 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 an Euler equation for the positions of the nodes along the boundary.

The nodes are held stationary for several time-step solutions of the transformed PDE's for P. An explicit finite-difference method was used with stability maintained by specifying a Courant limitation on time step. The Euler equations for the spacial coordinates are then solved using a Jacobi iteration. Only a few iterations are needed for convergence. Information on the solution is transferred from the old grid to the new one either by means of an interpolation function or by solution of transport equations in conservative 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.

Yanenko, Kovenya, Lisejkin, Fomin, and Vorozhtsov [35], Yanenko, Kroshko, Liseikin, Fomin, Shapeev, and Shitov [36], and Yanenko, Lisseikin, and Kovenia [37] also used a transformation from physical coordinates $(X_1, X_2, t)$ to computational coordinates $(\xi_1, \xi_2, t)$. The error measure used was a weighted sum of error measures. That is,

$$E = \lambda_c E_c + \lambda_a E_a + \lambda_v E_v \tag{32}$$

where

$$E_c = \left| \frac{\partial \xi_1}{\partial X_1} - \frac{\partial \xi_2}{\partial X_2} \right|^2 + J^R \left| \frac{\partial \xi_1}{\partial X_2} + \frac{\partial \xi_2^2}{\partial X_1} \right| \tag{33}$$

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

$$E_a = \left| U_1 - \frac{\partial X_1}{\partial t} \right|^2 + \left| U_2 - \frac{\partial X_2}{\partial t} \right|^2 \tag{34}$$

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:

$$E_v = W J^Q \ . \tag{35}$$

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 time- and space-dependent Euler equation for the transformation was obtained by minimizing the integral of E over $X_1$, $X_2$, and t. Alternately [37], a space-dependent Euler equation was obtained by minimizing the solution of E over $X_1$ and $X_2$ only. The Euler equation was solved simultaneously with the PDE for $\underline{P}$ using a time-stepping finite-difference method. Yanenko et al. [35,36] also discussed adaptation along a single coordinate direction.

Miller and Miller [38] and Miller [39] 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 [40], Gelinas, Doss, and Miller [41], Gelinas, Doss, Vajk, Djomehri, and Miller [42,43], Gelinas and Doss [44,45], Gelinas, Doss, and Carlson [46], Djomehri [47], and Miller [48,49]. In this method the error measure may be interpreted as being the square of the residual of the PDE written in finite-element form. ODE's for the nodal value of the physical variable and the nodal coordinate are obtained by minimizing the integral of error measure over spacial coordinates. Only the simplest one-dimensional version will be described herein, but the method has been applied very successfully in two dimensions.

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

$$\frac{dP}{dt} = \frac{\partial P}{\partial t} - \frac{\partial P}{\partial X}\frac{\partial X}{\partial t} = f\left[ t,\ X(\xi,t),\ P(\xi,t),\ \frac{\partial P}{\partial \xi},\ \frac{\partial^2 P}{\partial \xi^2},\ \frac{\partial X}{\partial \xi},\ \frac{\partial^2 X}{\partial \xi^2} \right] . \tag{36}$$

The continuous solution P and physical variable X are approximated by piecewise continuous functions $P^h$ and $X^h$.

$$P^h = \sum_{n=1}^{N} \alpha^n(\xi)\, P^n(t) \tag{37}$$

$$X^h = \sum_{n=1}^{N} \alpha^n(\xi)\, X^n(t) \tag{38}$$

The coefficients $P^n$ and $X^n$ are, respectively, the values of the solution and the spacial coordinate at each node n. $\alpha^n$ is the basis function at node n. In this simplest formulation of the MFE method, $\alpha^n$ is a linear 'hat' basis function defined by

$$\alpha^n = \begin{cases} 0 & \text{if} \quad \xi < \xi^{n-1} \\ \xi - \xi^{n-1} & \text{if} \quad \xi^{n-1} \leq \xi < \xi^n \\ \xi^{n+1} - \xi & \text{if} \quad \xi^n \leq \xi < \xi^{n+1} \\ 0 & \text{if} \quad \xi^{n+1} \leq \xi \end{cases} . \tag{39}$$

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

$$\frac{dP^h}{dt} = \sum_{n=1}^{N} \alpha^n \dot{P}^n + \beta^n \dot{X}^n , \tag{40}$$

where $\beta^n$ is a second basis function defined by

$$\beta^n = \frac{\partial P^h}{\partial X} = \begin{cases} 0 & \xi < \xi^{n-1} \\ \alpha^n M^n & \text{if} \quad \xi^{n-1} \leq \xi < \xi^n \\ \alpha^n M^{n+1} & \xi^n \leq \xi < \xi^{n+1} \\ 0 & \xi^{n+1} \leq \xi \end{cases} \tag{41}$$

with

$$M^n = \frac{P^n - P^{n-1}}{X^n - X^{n-1}} . \tag{42}$$

$\dot{P}^n$ and $\dot{X}^n$ are the time derivatives of $P^n$ and $X^n$, respectively. A set of ODE's for $X^n$ and $P^n$ at each node was obtained by minimizing the integral of the square of the residual of the PDE, with X and P replaced by their discrete representations, $X^h$ and $P^h$. The minimization was obtained by requiring that

$$\frac{\partial I}{\partial \dot{P}^n} = \frac{\partial I}{\partial \dot{X}^n} = 0 \tag{43}$$

for all nodes n where

$$I = \int \left[ \frac{dP^h}{dt} - F\left[ t, X^h(\xi,t), P^h(\xi,t), \frac{\partial P^h}{\partial \xi}, \frac{\partial X^h}{\partial \xi} \right] \right]^2 \frac{\partial X^h}{\partial \xi} \, d\xi \qquad (44)$$

over the numerical domain. It should be noted that PDE terms containing the second derivative with respect to $\xi$ can be reduced to integral terms containing the first derivative with respect to $\xi$ by suitable manipulation and by integration by parts. The resultant integral equations are

$$\int \alpha^n \left[ \frac{dP^h}{dt} - F\left[ t, X^h(\xi,t), P^h(\xi,t), \frac{\partial P^h}{\partial \xi}, \frac{\partial X^h}{\partial \xi} \right] \right] \frac{\partial X^h}{\partial \xi} \, d\xi = 0 \qquad (45)$$

and

$$\int \beta^n \left[ \frac{dP^h}{dt} - F\left[ t, X^h(\xi,t), P^h(\xi,t), \frac{\partial P^h}{\partial \xi}, \frac{\partial X^h}{\partial \xi} \right] \right] \frac{\partial X^h}{\partial \xi} \, d\xi = 0 \quad . \qquad (46)$$

The basis functions $\alpha^n$ and $\beta^n$ act as test functions over the domain of integration.

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

$$[A] \, \underline{S} = \underline{R} \quad . \qquad (47)$$

[A] is a block diagonal matrix and $\underline{S}$ a column vector of the time derivatives of P and X at each node. $\underline{R}$, also, is a column vector. Additional terms are added to [A] and $\underline{R}$ to prevent excessive node speed or the crossing of nodes and to prevent singularities in [A] in regions of constant slope where the $\beta^n$ are linearly dependent. Miller [39,48] discusses the many different forms that have been used for these additional terms and also discusses other strategies for node-movement regularization. The numerical values of [A] and $\underline{R}$ are obtained on each time step by evaluating the integrals of equations (45) and (46) either analytically or numerically. A Newtons method iteration was used as part of the solution of equation (47). Since the matrices involved can be very large, the computer time taken to solve the system of equations can constitute a large portion of the total solution time.

The implicit multistep backward-difference method of Gear [6], with many modifications, has been used to solve equation (47) for P and X at each node. Alternately, an implicit Runge-Kutta method devised by one of the authors (K. Miller) has been used. Adaptive-node computations typically required one-tenth the number of nodes used by nonadaptive computations with results of comparable accuracy.

In problems with multicomponent solutions $\underline{P}$, the authors defined multicomponent functions $\underline{\beta}^n$ and obtained weighted ODE's for $X^n$. The use of higher-order basis functions (quadratic or cubic) is also discussed. Dupont [12] has completed a partial analysis of the MFE method.

Gelinas, Doss, and Miller [41] have implemented the method in a commercial package for the solution of PDE's. The package allows the user to describe simple PDE's by executing a series of subroutine calls which automatically assemble the required finite-element terms into equation (47).

Herbst, Schoombie, and Mitchell [50,51] extended the work of Miller and Miller [38,39] by using piecewise Hermite cubic polynomials, in place of the test functions $\alpha^n$ and $\beta^n$ used in equations (45) and (46), while retaining $\alpha^n$ and $\beta^n$ in equations (37), (38), and (40). Herbst, Schoombie, Griffiths, and Mitchell [52] have analyzed their method and that of Miller and Miller and conclude that both methods tend to equidistribute the spacial 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.

Lee and Ramos [53] solved a flame-propagation problem using an adaptive finite-element technique. The PDE's were transformed from physical coordinates (X,t) to a normalized Langrangian coordinate system $(\xi,t)$ where $\xi$ varied from 0 to 1. The continuous solution $\underline{P}$ at time $t_i$ was approximated by

$$\underline{P}^h(t_i) = \sum_{n=1}^{N} \alpha^n(\xi)\, \underline{P}^n(t_i) \quad , \tag{48}$$

where $\underline{P}^n$ is the value of the solution at node n for the current node distribution, and the $\alpha^n$ are linear basis 'hat' functions defined by

$$\alpha^n = \begin{cases} 0 & \text{if} \quad \xi < \xi^{n-1} \\ \dfrac{\xi - \xi^{n-1}}{\xi^n - \xi^{n-1}} & \text{if} \quad \xi^{n-1} \leq \xi < \xi^n \\ \dfrac{\xi^{n+1} - \xi}{\xi^{n+1} - \xi^n} & \text{if} \quad \xi^n \leq \xi < \xi^{n+1} \\ 0 & \text{if} \quad \xi^{n+1} \leq \xi \end{cases} \tag{49}$$

The method of Galerkin was employed to obtain ODE's for the solution at each node, and the ODE's were solved using finite differences in time. The values of $\xi^n$ were changed every 10 time steps, so as to concentrate the nodes in regions of steepest solution gradient. The exact method of repositioning the the nodes was not specified, but reference was made to the work of Dwyer and Sanders. The new basis functions thus obtained were used to define a new discretized solution approximation

$$\underline{P}^h_*(t_i) = \sum_{n=1}^{N} \alpha^n_*(\xi)\, \underline{P}^n_*(t_i) \quad , \tag{50}$$

where the new nodal $\underline{P}^n_*$ values were computed by requiring that

$$\int_0^1 \left[ \frac{1}{2}\, \underline{P}^h_*\, \underline{P}^h_* - \underline{P}^h\, \underline{P}^h_* \right] d\xi \tag{51}$$

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

## 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 [54,55] and Anderson and Rai [56] have developed one such method for one-dimensional and two-dimensional problems. They used an error measure $E_i$ in numerical coordinates $(\xi i,t)$. Various forms of $E_i$ were used, including $|\partial P/\partial \xi_i|$, $|(\partial P/\partial \xi_i)/(\partial X_i/\partial \xi_i)|$, $|\partial^2 P/\partial {\xi_i}^2|$, and $|\partial X_i/\xi_i| + \lambda|\partial P/\xi_i|$. 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 $\xi_i$ 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 time dependence of $\xi_1$ 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:

$$\frac{\partial \xi_1^{kq}}{\partial t} = \lambda_1 \sum_{m=1}^{M} \left[ \sum_{n=k+1}^{N} \frac{[\,E_1^{nm} - E_1^{avm}\,]}{r^Q} - \sum_{n=1}^{k-1} \frac{[\,E_1^{nm} - E_1^{avm}\,]}{r^Q} \right] . \qquad (52)$$

$E_1^{nm}$ is error measure $E_1$ at node n,m; $E_1^{avm}$ is an average of error measure $E_1$ along a line of constant $\xi_1$ 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 $\partial \xi_2^{kq}/\partial t$:

$$\frac{\partial \xi_2^{kq}}{\partial t} = \lambda_2 \sum_{n=1}^{N} \left[ \sum_{m=q+1}^{M} \frac{[\,E_2^{nm} - E_2^{avn}\,]}{r^Q} - \sum_{m=1}^{q-1} \frac{[\,E_2^{nm} - E_2^{avn}\,]}{r^Q} \right] . \qquad (53)$$

The current transformation between frame $(X_i,t)$ and frame $(\xi_i,t)$ was used to compute the corresponding node velocities $(\partial X_1^{kq}/\partial t,\ \partial X_2^{kq}/\partial t)$ from the $\xi_1$ and $\xi_2$ time derivatives. Constants $\lambda_1$ and $\lambda_2$ were adjusted on each time step, so that no node velocity could exceed a prespecified value.

The velocities constitute ODE's for $(X_1,X_2)$ at each node, which were solved, time step by time step, along with the ODE's for P at each node, using the explicit finite-difference method of MacCormack [17].

$\lambda_1$ and $\lambda_2$ were held below a prespecified value so that grid speeds were damped as the error measures became equidistributed. All grid speeds 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 nodes at boundaries improved results. Given a real node, its image (with 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 (52) and (53) 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 the authors estimated these cross derivatives, resulting in an inappropriate node distribution. The authors suggested the use of such measures as $|\partial^3P/\partial X_1\partial X_1\partial X_2|$ or $|\partial^3P/\partial X_1\partial X_2\partial X_2|$ 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; 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 [57] 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. In the same paper Anderson has reviewed a number of alternate methods for computing node speeds and introduces some new ideas, as yet untested.

J. B. Greenberg [58] has solved a one-component PDE transformed to computational coordinates $(\xi,t)$ 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

$$\Delta X^n = X^n - X^{n-1} \quad . \tag{54}$$

The ODE

$$\frac{d(\Delta X^n)}{dt} = \sum_{m=1}^{N} K^{mn}\,\Delta X^m - \sum_{m=1}^{N} K^{nm}\,\Delta X^n \tag{55}$$

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

$$\sum_{n=1}^{N} X^n - X^{n-1} \tag{56}$$

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 preset limits on the maximum and minimum allowed node separation. The $K^{nm}$ reduce in magnitude as the error measure is equidistributed. Equation (55) was linearized and solved analytically to determine the node distribution at each time step. Preliminary results have been 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.

Gnoffo [59,60], making use of the work of Dwyer et al. [21] and that of Rai and Anderson, used an error measure to define 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

$$F = K \Delta S \quad , \qquad (57)$$

where $\Delta S$ is the arc length between the two nodes and K is the local spring constant. Typically, K was of the form

$$K = 1 + E \quad , \qquad (58)$$

where E is an error measure. E was usually a weighted sum of the magnitudes of derivatives of each component of the solution $\underline{P}$. 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

$$\Delta S^n = \frac{S_{tot}}{K^n \sum\limits_m \frac{1}{K^m}} \qquad (59)$$

so as to equidistribute the spring forces $K^m \Delta S^m$. The nodes on a curve have been numbered consecutively for explanatory convenience. $\Delta S^n$ 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 solution was transferred from the old grid to the new grid using an interpolation function, and new values of $K^n$ were computed. Use of equation (59) 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.

The solution PDE's were solved using a finite-volume method. A finite-volume method is a geometrically conservative finite-difference approximation to the integral form of the PDE's. The domain is broken up into control volumes and finite-difference equations obtained from the conservation of intervolume 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 method of MacCormack [17]. Hindman [11] 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. 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 large Reynold's number were simulated.

The one-dimensional method of Davis and Flaherty [61] may be interpreted as using pseudoforces to move nodes. The PDE's were solved with fixed time step using a finite-element formulation in computational coordinates ($\xi$,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 $m^{th}$ derivative of the solution with respect to $\xi$. 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

$$\frac{X^{n+1} - X^n}{X^n - X^{n-1}} = \frac{\left|\dfrac{\partial^m P^n}{\partial X^m}\right|^{1/m}}{\left|\dfrac{\partial^m P^{n+1}}{\partial X^m}\right|^{1/m}} . \qquad (60)$$

The $m^{th}$ derivatives, approximated by finite differences, were given lower bounds to ensure stability of node movement. An under-relaxation iteration was used to solve equation (60), subject to constraints on first and last node coordinates and deformation of the trapezoidal space-time elements. The resultant and previous node positions were then used to extrapolate 'optimal' node positions for the next time step, thus determining the shape of the space-time elements to be used for computing $P^n$ at the next time step.

Only preliminary results have been published but they seem promising. Davis and Flaherty expect to introduce the use of variable time steps and will attempt the solution of problems with free boundaries or with more than one spacial dimension.

## 4.0 DISCUSSION OF PDE SOLVERS

The methods vary considerably from author to author in the ways of solving the ODE's obtained from the discretization of continuous PDE problems. 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 easily to use stiff ODE solvers. As a penalty for obtaining larger time steps, however, one generally must solve large matrix equations.

For gas flows with 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 ODE's for the physical variables $P^n$ can be altered so that 'optimal' node speeds can be derived from the difference between old and 'optimal' node positions. The node speeds constitute a set of ODE's for node position, which can be solved simultaneously with the ODE's for the $P^n$. Even though a larger set of ODE's must be solved simultaneously, the increase in computational cost is compensated in part because interpolation is not needed to transfer information from an old to a new grid.

The most promising two-dimensional methods are those of Yanenko et al. [35-37] or Brackbill and Saltzman [30-33], which control grid orthogonality and smoothness, and that of Miller and Miller [38,39], 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 very well with stiff problems. One technique of conversion, which White [19] has used, involves expressing P 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 [57], the time-independent Euler equation for node coordinates developed by Brackbill and Saltzman can be converted into a PDE for node speeds by use of differentiation with respect to time. The 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, the solution of the matrix equation $[A]\ \underline{S} = \underline{R}$ can consume a very large proportion of total computational time. The method of Herbst et al. [50-52] results in a smaller truncation error than that of Miller and Miller but suffers from the same problems of large computational overhead. It should be possible to obtain a much 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. Davis and Flaherty [61] have begun work on such a method but much must yet be done before their method can be used on stiff problems. The work of Lee and Ramos [53] 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 both Dwyer et al. [21-26] and Rai and Anderson [54-57] 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 [27] or of Anyiwo [29] might also be of value. The method of Gnoffo [59,60] 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.

Anyiwo [29] describes his adaptive method sketchily, but his paper contains some useful ideas applicable to the methods of other authors. These 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.

Causing the nodes to follow the solution characteristics, as done by Kansa et al. [28], 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 one-dimensional method of Pierson and Kutler [14] 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 other methods.

Methods, such as that of Klopfer and McRae [16], that depend on explicit calculation of PDE truncation error are probably awkward to apply for general PDE solvers.

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 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.

## 5.0 REFERENCES

1. J. F. Thompson, Z. U. A. Warsi, C. W. Mastin, 'Boundary-Fitted Coordinate Systems for Numerical Solution of Partial Differential Equations — A Review,' Journal of Computational Physics, Vol. 47, pp. 1-108, 1982.

2. J. F. Thompson, 'Grid Generation Techniques in Computational Fluid Dynamics,' AIAA Journal, Vol. 22, No. 11, pp. 1505-1523, 1984.

3. E. Turkel, 'Progress in Computational Physics,' Computers and Fluids, Vol. 11, No. 2, pp. 121-144, 1983.

4. J. F. Thompson, 'General Curvilinear Coordinate Systems,' in Numerical Grid Generation, edited by J. F. Thompson, North-Holland Publishing, Amsterdam, pp. 1-30, 1982.

5. R. J. Gelinas, 'Stiff Systems of Kinetic Equations — A Practitioner's View,' Journal of Computational Physics, Vol. 9, p. 222-236, 1972.

6. C. W. Gear, Numerical Initial Value Problems in Ordinary Differential Equations, Prentice Hall, New Jersey, 1971.

7. A. C. Hindmarsh, R. J. Gelinas, 'GEAR: Ordinary Differential Equation Solver,' Report UCID-30001, Lawrence Radiation Laboratory, Livermore, April 1, 1971.

8. W. H. Enright, T. E. Hull, B. Lindberg, 'Comparing Numerical Methods for Stiff Systems of O.D.E.'s,' International BIT, Vol. 15, pp. 10-48, 1975.

9. C. W. Mastin, 'Error Induced by Coordinate Systems,' in Numerical Grid Generation, edited by J. F. Thompson, North-Holland Publishing, Amsterdam, pp. 31-40, 1982.

10. E. Kalnay de Rivas, 'On the Use of Nonuniform Grids in Finite-Difference Equations,' Journal of Computational Physics, Vol. 10, pp. 202-210, 1972.

11. R. G. Hindman, 'Generalized Coordinate Forms of Governing Fluid Equations and Associated Geometrically Induced Errors,' AIAA Journal, Vol. 20, No. 10, pp. 1359-1367, 1982.

12. T. Dupont, 'Mesh Modification for Evolution Equations,' Mathematics of Computation, Vol. 39, No. 159, pp. 85-107, 1982.

13. D. O. Gough, E. A. Spiegel, J. Toomre, 'Highly Stretched Meshes as Functionals of Solutions,' Proceedings of the 4th International Conference on Numerical Methods in Fluid Dynamics, pp. 191-196, 1974.

14. B. L. Pierson, P. Kutler, 'Optimal Nodal Point Distribution for Improved Accuracy in Computational Fluid Dynamics,' AIAA Journal, Vol. 18, No. 1, pp. 49-54, 1980.

15. V. E. Denny, R. B. Landis, 'A New Method for Solving Two-Point Boundary Value Problems Using Optimal Node Distribution,' Journal of Computational Physics, Vol. 9, pp. 120-137, 1972.

16. G. H. Klopfer, D. S. McRae, 'The Nonlinear Modified Equation Approach to Analyzing Finite Difference Schemes,' AIAA Paper 81-1029, pp. 317-322, 1981.

17. R. W. MacCormack, 'The Effect of Viscosity in Hypervelocity Impact Cratering,' AIAA Paper 69-354, pp. 1-7, 1969.

18. A. B. White, Jr., 'On Selection of Equidistributing Meshes for Two-Point Boundary-Value Problems,' SIAM Journal on Numerical Analysis, Vol. 16, No. 3, pp. 472-502, 1979.

19. A. B. White, Jr., 'On the Numerical Solution of Initial/Boundary-Value Problems in One Space Dimension,' SIAM Journal on Numerical Analysis, Vol. 19, No. 4, pp. 683-697, 1982.

20. C. M. Ablow, S. Schechter, 'Campylotropic Coordinates,' Journal of Computational Physics, Vol. 27, pp. 351-362, 1978.

21. H. A. Dwyer, R. J. Kee, B. R. Sanders, 'Adaptive Grid Method for Problems in Fluid Mechanics and Heat Transfer,' AIAA Journal, Vol. 18, No. 10, pp. 1205-1212, 1980.

22. H. A. Dwyer, M. D. Smooke, 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 Publishing, Amsterdam, pp. 339-356, 1982.

23. H. A. Dwyer, R. J. Kee, P. K. Barr, B. R. Sanders, 'Transient Droplet Heating at High Peclet Number,' Journal of Fluids Engineering, Vol. 105, No. 1, pp. 83-88, 1983.

24. H. A. Dwyer, B. R. Sanders, F. Raiszadek, 'Ignition and Flame Propagation Studies with Adaptive Numerical Grids,' Combustion and Flame, Vol. 52, pp. 11-23, 1983.

25. H. A. Dwyer, 'Grid Adaptation for Problems in Fluid Dynamics,' AIAA Journal, Vol. 22, No. 12, pp. 1705-1712, 1984.

26. H. A. Dwyer, 'A Discussion of Some Criteria for the Use of Adaptive Gridding,' AIAA Paper 83-1932, pp. 319-322, 1983.

27. D. E. Potter, G. H. Tuttle, 'The Construction of Discrete Orthogonal Coordinates,' Journal of Computational Physics, Vol. 13, pp. 483-501, 1973.

28. E. J. Kansa, D. L. Morgan, L. K. Morris, 'A Simplified Moving Finite Difference Scheme: Application to Dense Gas Dispersion,' SIAM Journal on Scientific and Statistical Computing, Vol. 5, No. 3, pp. 667-683, 1984.

29. J. C. Anyiwo, 'Idealized Dynamic Grid Computation of Physical Systems,' in Numerical Grid Generation, edited by J. F. Thompson, North-Holland Publishing, Amsterdam, pp. 837-858, 1982.

30. J. U. Brackbill, J. Saltzman, 'An Adaptive Computation Mesh for the Solution of Singular Perturbation Problems,' in Numerical Grid Generation Techniques, edited by R. E. Smith, Proceedings of the Numerical Grid Generation Workshop at Langley Research Center, NASA CP-2166, pp. 193-196, October 1980.

31. J. U. Brackbill, J. S. Saltzman, 'Adaptive Zoning for Singular Problems in Two Dimensions,' Journal of Computational Physics, Vol. 46, pp. 342-368, 1982.

32. J. U. Brackbill, 'Coordinate System Control: Adaptive Meshes,' in Numerical Grid Generation, edited by J. F. Thompson, North-Holland Publishing, Amsterdam, pp. 277-294, 1982.

33. J. Saltzman, J. Brackbill, 'Applications and Generalizations of Variational Methods for Generating Adaptive Meshes,' in Numerical Grid Generation, edited by J. F. Thompson, North-Holland Publishing, Amsterdam, pp. 865-884, 1982.

34. A. M. Winslow, 'Numerical Solution of the Quasilinear Poisson Equation in a Nonuniform Triangle Mesh,' Journal of Computational Physics, Vol. 2, pp. 149-172, 1967.

35. N. N. Yanenko, V. M. Kovenya, V. D. Lisejkin, V. M. Fomin, E. V. Vorozhtsov, 'On Some Methods for the Numerical Simulation of Flows with Complex Structure,' Computer Methods in Applied Mechanics and Engineering, Vol. 17/18, pp. 659-671, 1979.

36. N. N. Yanenko, E. A. Kroshko, V. V. Liseikin, V. M. Fomin, V. P. Shapeev, Yu. A. Shitov, 'Methods for the Construction of Moving Grids for Problems of Fluid Dynamics with Big Deformations,' Lecture Notes in Physics, Vol. 59, pp. 454-459, 1976.

37 N. N. Yanenko, V. D. Lisseikin, V. M. Kovenia, 'The Method of the Solution of Gaz Dynamical Problems in Moving Meshes,' Lecture Notes in Physics, Vol. 91, pp. 48-61, 1979.

38. K. Miller, R. N. Miller, 'Moving Finite Elements. I,' SIAM Journal on Numerical Analysis, Vol. 18, No. 6, pp. 1019-1032, 1981.

39. K. Miller, 'Moving Finite Elements. II,' SIAM Journal on Numerical Analysis, Vol. 18, No. 6, pp. 1032-1057, 1981.

40. J. Djomehri, 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, October 12, 1981.

41. R. J. Gelinas, S. K. Doss, K. Miller, 'The Moving Finite Element Method: Applications to General Partial Differential Equations with Multiple Large Gradients,' Journal of Computational Physics, Vol. 40, pp. 202-249, 1981.

42. R. J. Gelinas, S. K. Doss, J. P. Vajk, J. Djomehri, K. Miller, 'Moving Finite Elements in 2-D,' Preprint for 10th IMACS Congress, Montreal, August 8-13, 1982.

43. R. J. Gelinas, S. K. Doss, J. P. Vajk, J. Djomehri, K. Miller, 'Moving Finite Elements in 2-D: Fluid Dynamics Examples,' Preprint for 10th IMACS Congress, Montreal, August 8-13, 1982.

44. R. J. Gelinas, 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.

45. R. J. Gelinas, S. K. Doss, 'The Moving Finite Element Method: One-Dimensional Transient Flow Applications,' Preprint for 10th IMACS Congress, Montreal, August 8-13, 1982.

46. R. J. Gelinas, S. K. Doss, N. N. Carlson, 'Moving Finite Element Research, Physical Versus Non-Physical Dissipation Effects,' Presented at Semi-Annual DNA Airblast Review Meeting held at R and D Associates, Marina del Rey, California, December 8-9, 1982.

47. 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, January 1983.

48. K. Miller, 'Alternate Modes to Control the Nodes in the Moving Finite Element Method,' Technical Report PAM-159, Center For Pure and Applied Mathematics, University of California, Berkeley, June 1983.

49. K. Miller, 'Recent Results on Finite Element Methods with Moving Nodes,' Technical Report PAM-255, Center For Pure and Applied Mathematics, University of California, Berkeley, November 1984.

50. B. M. Herbst, S. W. Schoombie, A. R. Mitchell, 'A Moving Petrov-Galerkin Method for Transport Equations,' International Journal for Numerical Methods in Engineering, Vol. 18, pp. 1321-1336, 1982.

51. B. M. Herbst, S. W. Schoombie, A. R. Mitchell, 'Equidistributing Principles in Moving Finite Element Methods,' Journal of Computational and Applied Mathematics, Vol. 9, pp. 377-389, 1983.

52. B. M. Herbst, S. W. Schoombie, D. F. Griffiths, A. R. Mitchell, 'Generalized Petrov-Galerkin Methods for the Numerical Solution of Burgers' Equation,' International Journal for Numerical Methods in Engineering, Vol. 20, pp. 1273-1289, 1984.

53. D. N. Lee, J. I. Ramos, 'Application of the Finite-Element Method to One-Dimensional Flame Propagation Problems,' AIAA Journal, Vol. 21, No. 2, pp. 262-269, 1983.

54. M. M. Rai, D. A. Anderson, 'Grid Evolution in Time Asymptotic Problems,' in Numerical Grid Generation Techniques, edited by R. E. Smith, Proceedings of the Numerical Grid Generation Workshop at Langley Research Center, NASA CP-2166, pp. 409-430, 1980.

55. M. M. Rai, D. A. Anderson, 'Application of Adaptive Grids to Fluid-Flow Problems with Asymptotic Solutions,' AIAA Journal, Vol. 20, No. 4, pp. 496-502, 1982.

56. D. A. Anderson, 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 Publishing, Amsterdam, pp. 317-338, 1982.

57. D. A. Anderson, 'Adaptive Mesh Schemes Based on Grid Speeds,' AIAA Paper 83-1931, pp. 311-318, 1983.

58. J. B. Greenberg, 'A New Self-Adaptive Grid Method,' AIAA Journal, Vol. 23, No. 2, pp. 317-320, 1985.

59. 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 Publishing, Amsterdam, pp. 819-835, 1982.

60. P. A. Gnoffo, 'A Finite-Volume, Adaptive Grid Algorithm Applied to Planetary Entry Flowfields,' AIAA Journal, Vol. 21, No. 9, pp. 1249-1254, 1983.

61. S. F. Davis, J. E. Flaherty, 'An Adaptive Finite Element Method for Initial-Boundary Value Problems for Partial Differential Equations,' SIAM Journal on Scientific and Statistical Computing, Vol. 3, No. 1, pp. 6-27, 1982.

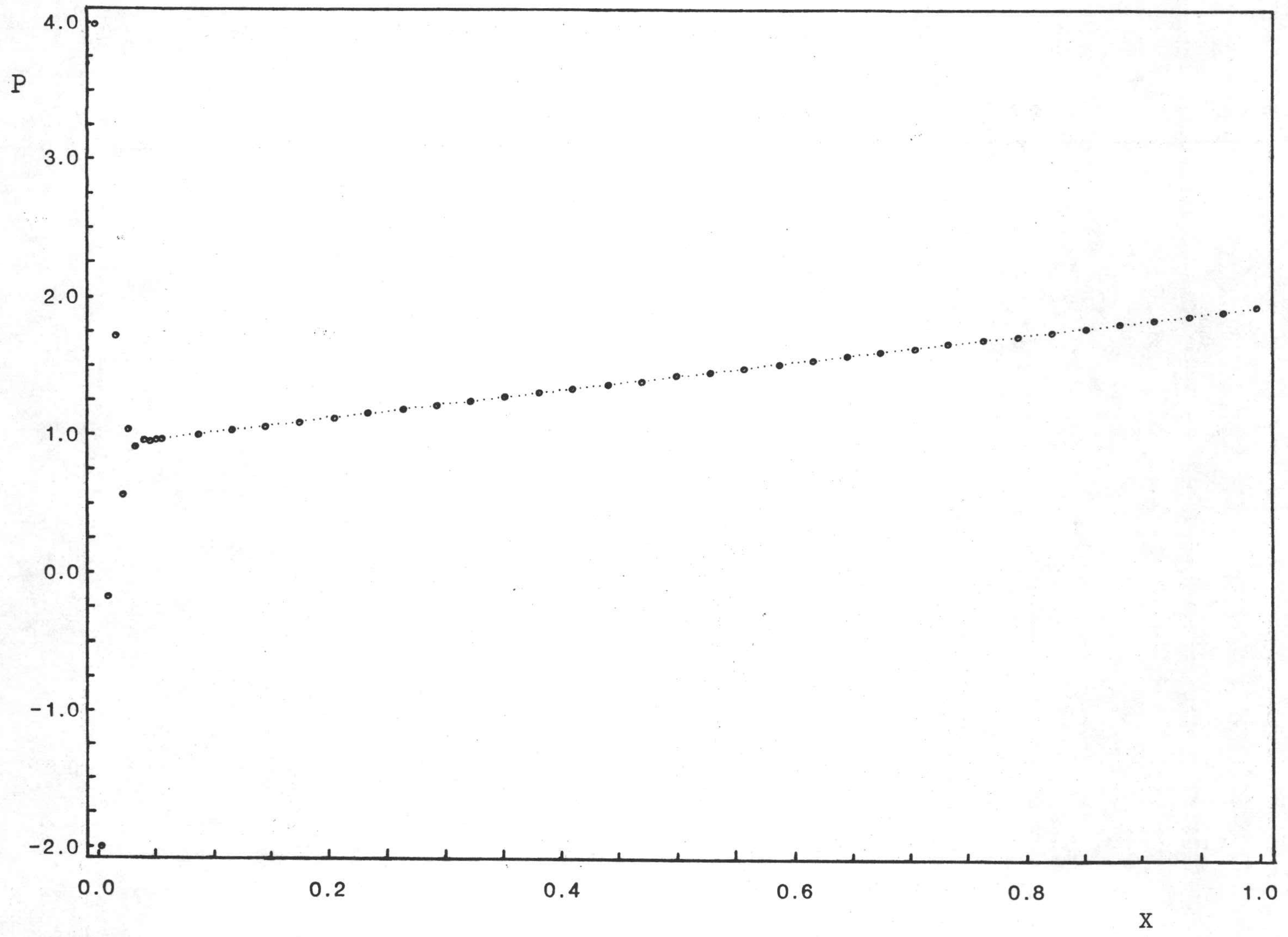

Fig. 1. Nonadaptive solution with 171 nodes and $\varepsilon = 0.0015$; showing Gibb's oscillations.

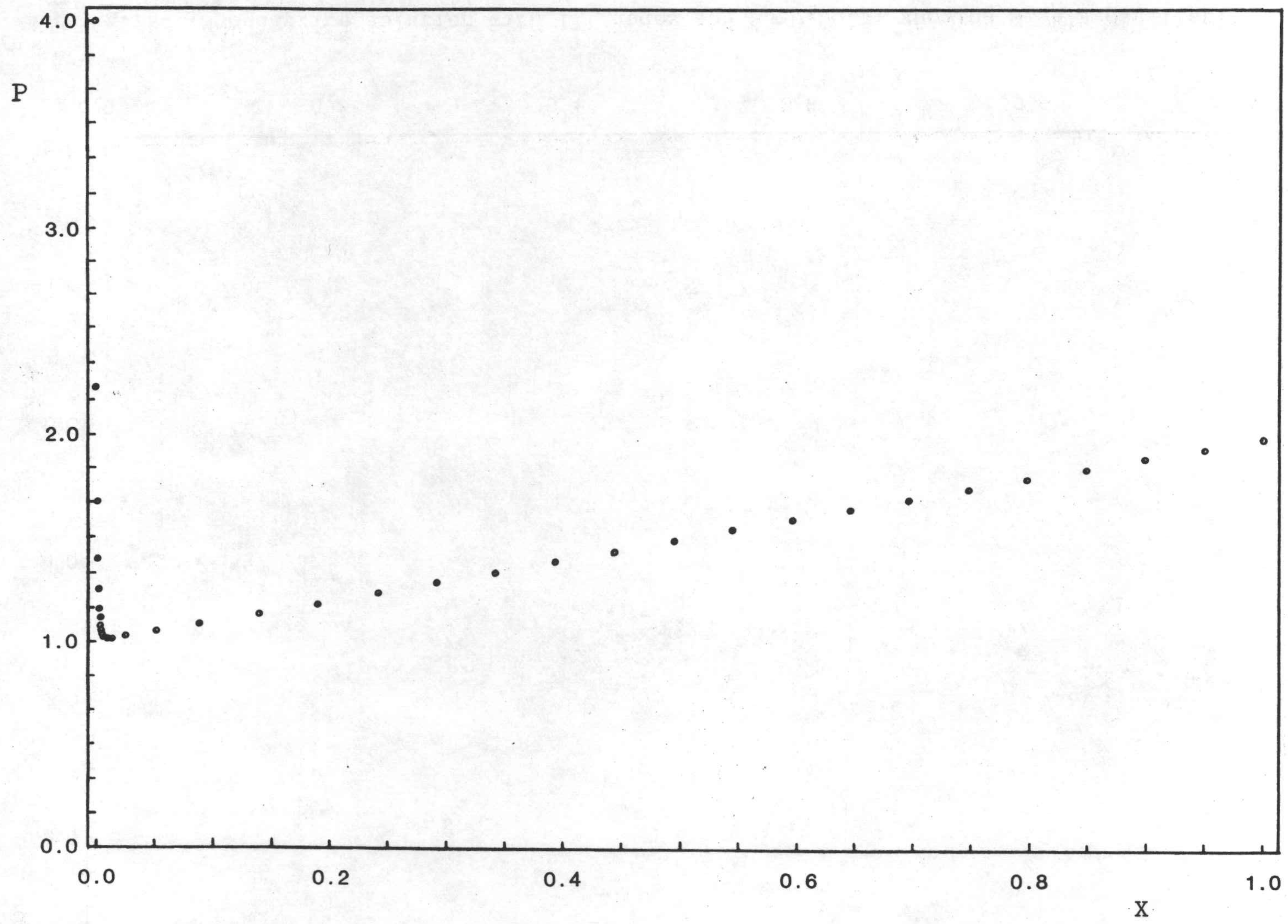

Fig. 2. Adaptive solution with 41 nodes and $\varepsilon = 0.0015$.

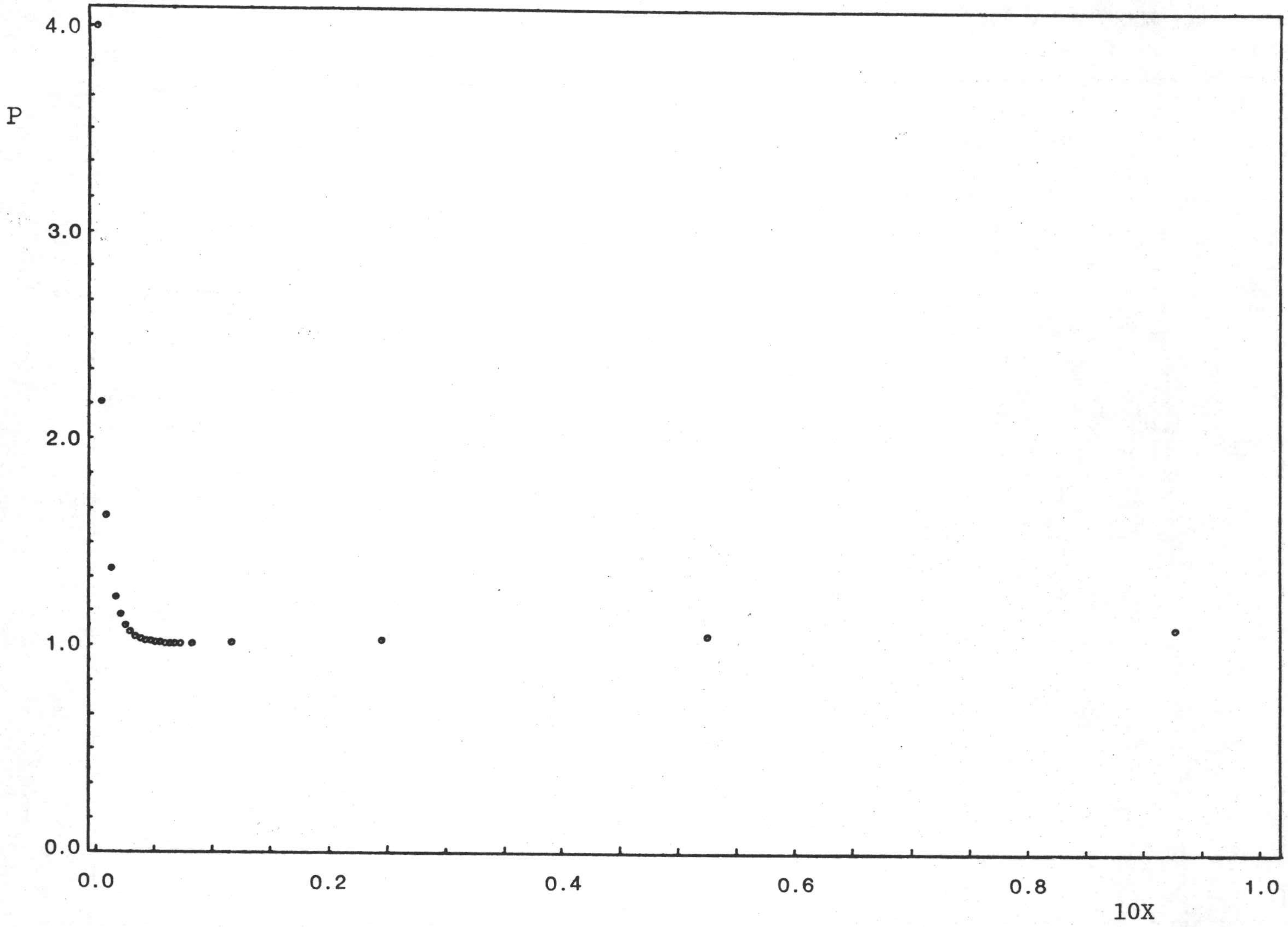

Fig. 3. Adaptive solution with 41 nodes and $\varepsilon = 0.001$; high-resolution plot of boundary-layer region.

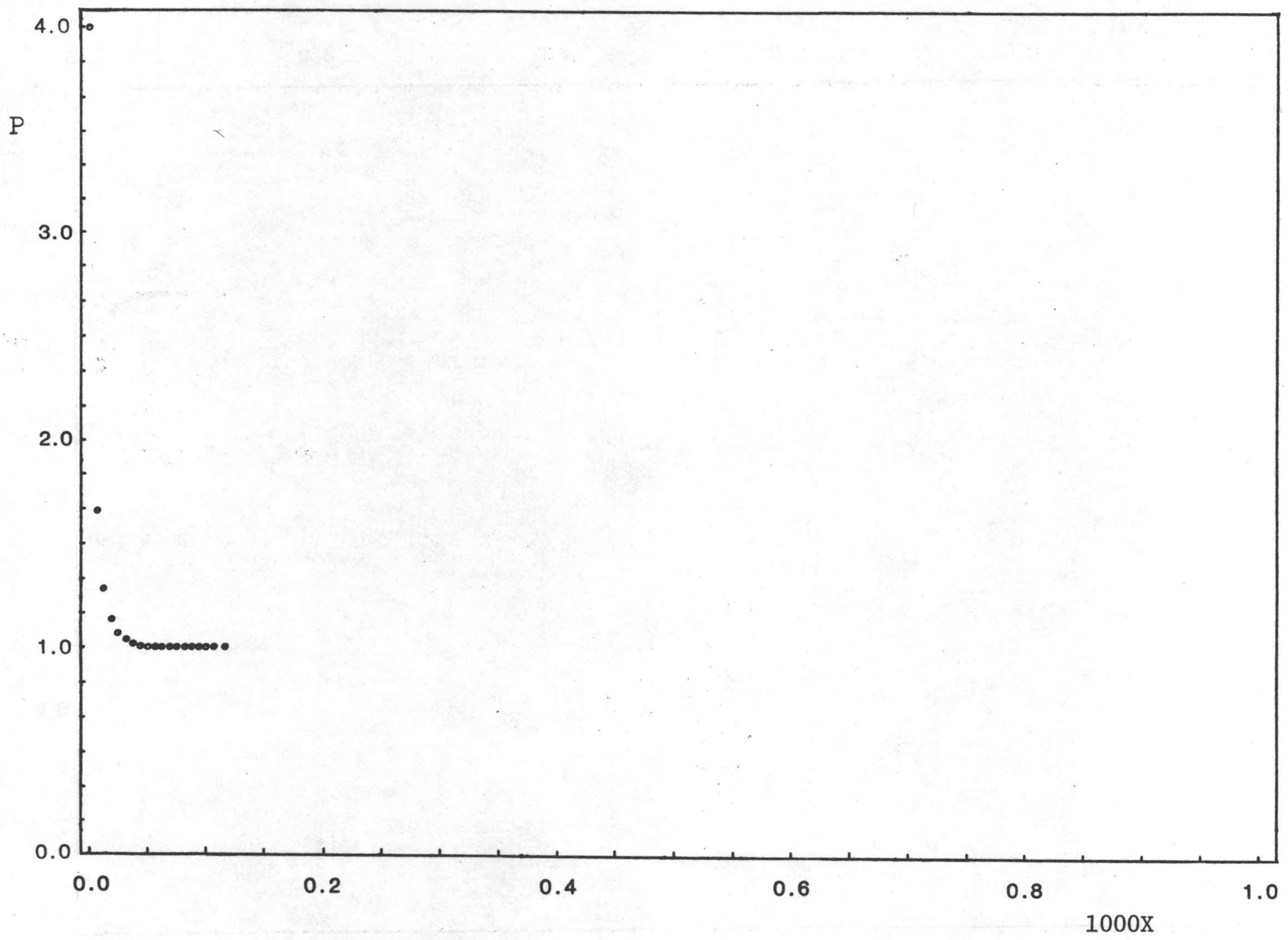

Fig. 4. Adaptive solution with 41 nodes and $\varepsilon = 0.00001$; high-resolution plot of boundary-layer region.

UTIAS Review No. 46

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

REVIEW OF ADAPTIVE-GRID TECHNIQUES FOR SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS

Hawken, D. F.

1. Computational grids 2. Computational mesh 3. Adaptive 4. Review
5. Grid optimization 6. Grid refinement

I. UTIAS Review No. 46 II. Hawken, D. F.

This report 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 blast-wave or other nonstationary flow problems that contain moving regions of rapid change in the flow variables, surrounded by regions of relatively smooth variation. Shock waves, contact surfaces, phase-change interfaces, boundary layers, and other structures, can be modelled in detail 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.

Available copies of this report are limited. Return this card to UTIAS, if you require a copy.

UTIAS Review No. 46

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

REVIEW OF ADAPTIVE-GRID TECHNIQUES FOR SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS

Hawken, D. F.

1. Computational grids 2. Computational mesh 3. Adaptive 4. Review
5. Grid optimization 6. Grid refinement

I. UTIAS Review No. 46 II. Hawken, D. F.

This report 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 blast-wave or other nonstationary flow problems that contain moving regions of rapid change in the flow variables, surrounded by regions of relatively smooth variation. Shock waves, contact surfaces, phase-change interfaces, boundary layers, and other structures, can be modelled in detail 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.

Available copies of this report are limited. Return this card to UTIAS, if you require a copy.

UTIAS Review No. 46

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

REVIEW OF ADAPTIVE-GRID TECHNIQUES FOR SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS

Hawken, D. F.

1. Computational grids 2. Computational mesh 3. Adaptive 4. Review
5. Grid optimization 6. Grid refinement

I. UTIAS Review No. 46 II. Hawken, D. F.

This report 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 blast-wave or other nonstationary flow problems that contain moving regions of rapid change in the flow variables, surrounded by regions of relatively smooth variation. Shock waves, contact surfaces, phase-change interfaces, boundary layers, and other structures, can be modelled in detail 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.

Available copies of this report are limited. Return this card to UTIAS, if you require a copy.

UTIAS Review No. 46

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

REVIEW OF ADAPTIVE-GRID TECHNIQUES FOR SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS

Hawken, D. F.

1. Computational grids 2. Computational mesh 3. Adaptive 4. Review
5. Grid optimization 6. Grid refinement

I. UTIAS Review No. 46 II. Hawken, D. F.

This report 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 blast-wave or other nonstationary flow problems that contain moving regions of rapid change in the flow variables, surrounded by regions of relatively smooth variation. Shock waves, contact surfaces, phase-change interfaces, boundary layers, and other structures, can be modelled in detail 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.

Available copies of this report are limited. Return this card to UTIAS, if you require a copy.