JOURNAL OF COMPUTATIONAL PHYSICS 47, 387–411 (1982)

Unified Approach to Simulation on Deforming Elements with Application to Phase Change Problems

DANIEL R. LYNCH

Thayer School of Engineering, Dartmouth College, Hanover, New Hampshire 03755

Received October 28, 1981; revised March 26, 1982

Several different approaches to simulation on deforming finite elements are shown to generate the same weighted residuals formulation for the evolution of the dependent variables. Control of node motion by means of mesh stretching in two dimensions is illustrated in the context of phase change problems. Simulation of problems with large phase boundary excursions shows good agreement with analytic solutions.

INTRODUCTION

A variety of numerical methods have now been advanced which are based on finite elements which move and/or deform during the course of simulation. Development of these methods has been inspired by a broad range of physical problems, including applications in heat and mass transfer [8, 20], phase change [3, 12], hydraulics [11], porous medium flow [6, 19], free surface problems [27], and gas dynamics [7]. Generally speaking, all of these diverse applications share the common feature that simulation can be effectively achieved on a finite element grid whose topology—i.e., the number, type, and relative spatial configuration of elements—remains constant for large numbers of time steps; but for various reasons the node locations move as the simulation proceeds, giving rise to element deformation.

In some applications, grid deformation arises from the changing location of a shock, phase transition, or other essential discontinuity or singularity in the solution which is most effectively represented at nodes or element boundaries. In other applications, the solution may be continuous throughout the domain, but may exhibit steep gradients which move about, as in convection-dominated transport problems. Advantage in these problems can be gained by concentrating numerical detail at any point in time in localized portions of the domain. Similarly, problems with distributed phase transition zones may require specially formulated elements which would then follow the motion of the zone. In other applications, the motion of an external boundary may be prescribed exogenously or may be tied to the solution at the boundary, as in free surface or ablation problems. For problems specified on infinite domains, accuracy and economy can be achieved with a fixed number of elements by keeping the numerical domain as small as possible, and stretching or expanding it as

0021-9991/82/090387-25$02.00/0

Copyright © 1982 by Academic Press, Inc.
All rights of reproduction in any form reserved.

the solution evolves. Finally, discontinuities or steep gradients in coefficients and/or forcing functions which move about through time may require element deformation even when the solution itself is well behaved.

A basic concept in these applications is that node locations are adjusted continuously, i.e., during each time step, such that an appropriate numerical representation is maintained throughout the spatial and temporal domains. The finite element method has shown itself to be well suited for this type of service for many of the same reasons which recommend it for fixed-grid application: irregular mesh geometry arising from node motion is easily accommodated; representation of functional variation in coefficients is possible; and higher order elements with various degrees of continuity among elements are available.

Of course, grid deformation introduces additional complexities not normally encountered in fixed-grid simulations. Specifically, two general types of problems may be identified:

(a) The changing nature of the numerical interpolation on deforming elements must be accounted for in the discrete problem formulation. Specifically, expressions involving derivatives and/or integrals of the numerical solution must be adjusted to account for node motion.

(b) Node motion must be managed and controlled automatically, in a way which satisfies problem-specific criteria and which avoids undesirable side effects such as excessive element shearing, tangling, etc., and/or inappropriate local interpolation.

Successful simulation with deforming finite elements ultimately depends on the solution of these special problems in a way which can be automated cost-effectively for a given application.

It is shown below that several apparently different approaches to problem formulation on a deforming grid, i.e., problem (a) above, can in fact be reduced to a single *unified* formulation for the dependent variables which is both intuitively appealing and easily automated. A discussion follows on the control of node motion, problem (b) above. While some generalizations are possible here, it appears that this problem is most effectively addressed in the context of specific applications. The remainder of the article is devoted to the synthesis and testing of a numerical method for phase change problems, starting with the unified formulation and demonstrating the selection of node control strategy as well as the conventional numerical features: element type, numerical quadrature, time stepping procedure, etc. Two-dimensional model results are shown for problems with analytic solutions, in which elements are subjected to very large deformation.

FORMULATION ON DEFORMING ELEMENTS

At the outset, a basic identity may be established concerning the derivatives of the finite element basis functions ϕ_i with respect to the node coordinates $\mathbf{X}_j$. Since the node locations are assumed to be variable, the explicit dependence of ϕ_i on $\mathbf{X}_j$ as well as on $\mathbf{X}$ (the vector of independent space variables) must be recognized:

$$\phi_i = \phi_i(\mathbf{X}, \mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_N), \tag{1}$$

where N is the number of nodes. Following the customary practice, ϕ_i may also be expressed in the simpler form

$$\phi_i = \phi_i(\boldsymbol{\xi}), \tag{2}$$

where $\boldsymbol{\xi}$ is the vector of local coordinates related to $\mathbf{X}$ by the transformation

$$\mathbf{X} = \sum_j \mathbf{X}_j \psi_j(\boldsymbol{\xi}) \tag{3}$$

(The case $\psi_j = \phi_j$ is the isoparametric case.)

Application of the chain rule to (1) yields an expression for an incremental change in ϕ_i:

$$d\phi_i = \nabla\phi_i \cdot d\mathbf{X} + \sum_j \nabla_j \phi_i \cdot d\mathbf{X}_j, \tag{4}$$

where ∇_j is the gradient with respect to $\mathbf{X}_j$, holding $\mathbf{X}$ constant. (In conventional 3–D Cartesian coordinates, $\nabla_j = \{\partial/\partial x_j, \partial/\partial y_j, \partial/\partial z_j\}$.) Similarly, from (3), an incremental change in $\mathbf{X}$ is given by

$$d\mathbf{X} = \sum_j (d\mathbf{X}_j \psi_j + \mathbf{X}_j \, d\psi_j). \tag{5}$$

Thus, assembling (4) and (5),

$$d\phi_i = \sum_j (\psi_j \nabla\phi_i + \nabla_j \phi_i) \cdot d\mathbf{X}_j + \nabla\phi_i \cdot \sum_j \mathbf{X}_j \, d\psi_j. \tag{6}$$

Note that since both sets of functions ϕ_i and ψ_j depend upon $\boldsymbol{\xi}$ alone, $d\phi_i$ and $d\psi_j$ simultaneously vanish when $d\boldsymbol{\xi} = 0$. In this case,

$$\sum_j (\psi_j \nabla\phi_i + \nabla_j \phi_i) \cdot d\mathbf{X}_j = 0. \tag{7}$$

Since the condition $d\boldsymbol{\xi} = 0$ can be achieved with arbitrary values of the node displacements $d\mathbf{X}_j$, then each component of (7) must vanish individually

$$\nabla_j \phi_i = -\psi_j \nabla\phi_i. \tag{8}$$

In 3–D Cartesian coordinates, Eq. (8) may be written as

$$\begin{Bmatrix} \frac{\partial \phi_i}{\partial x_j} \\ \frac{\partial \phi_i}{\partial y_j} \\ \frac{\partial \phi_i}{\partial z_j} \end{Bmatrix} = -\psi_j \begin{Bmatrix} \frac{\partial \phi_i}{\partial x} \\ \frac{\partial \phi_i}{\partial y} \\ \frac{\partial \phi_i}{\partial z} \end{Bmatrix}. \tag{9}$$

The importance of the result (8) lies in the fact that the unconventional derivatives with respect to the node locations can be expressed in conventional terms which are readily computable. Thus, any problem in which these terms appear may be recast into friendlier form, avoiding a cumbersome set of calculations (particularly in the case of higher-order elements). Of direct interest herein is the implicit time dependence of the ϕ_i when the $\mathbf{X}_j$ are time dependent. Differentiation of (1) with respect to time, while holding $\mathbf{X}$ fixed, yields

$$\frac{\partial \phi_i}{\partial t} = \sum_j \nabla_j \phi_i \cdot \frac{d\mathbf{X}_j}{dt}. \tag{10}$$

Substitution from (8) yields

$$\frac{\partial \phi_i}{\partial t} = -\sum_j \psi_j \nabla \phi_i \cdot \frac{d\mathbf{X}_j}{dt} = -\mathbf{V}^e \cdot \nabla \phi_i, \tag{11}$$

where the velocity $\mathbf{V}^e$ is interpolated among the node velocities by the coordinate transformation functions

$$\mathbf{V}^e = \sum_j \frac{d\mathbf{X}_j}{dt} \psi_j. \tag{12}$$

Result (11), (12) was obtained by Lynch and Gray [11] by a different (although similar) line of reasoning.

The result (8) is also useful in steady state problems wherein the node locations are unknown at the outset and thus change during the course of iteration (e.g., free broudary problems). Newton–Raphson solution of the weighted residual equations in such a case would ultimately require evaluation of the derivatives $\nabla_j \phi_i$, since the $\mathbf{X}_j$ are unknowns of the problem. Equation (8) may be used in these problems to simplify the calculations, particularly for higher-order elements.

Equations (8)–(12) are used below to demonstrate the inherent unity in several approaches to deforming-grid simulation.

A. *Weighted Residuals Formulation*

Consider the partial differential equation

$$\frac{\partial U}{\partial t} + LU = f, \tag{13}$$

which is to be solved on a deforming grid. The operator L contains the unknown function U and its spatial derivatives, and need not be linear. The conventional weighted residuals formulation is given by the set of equations

$$\left\langle \left[\frac{\partial \hat{U}}{\partial t} + L\hat{U} - f \right], W_i \right\rangle = 0, \tag{14}$$

where $\hat{U}$ is the approximate numerical solution, W_i are a suitable set of spatial weighting functions, and $\langle\ ,\ \rangle$ is the inner product notation implying integration over the spatial domain. Assuming a numerical solution of the form

$$\hat{U}(\mathbf{X}, t) = \sum_j U_j(t)\, \phi_j(\mathbf{X}, \mathbf{X}_1(t), \mathbf{X}_2(t), \ldots, \mathbf{X}_N(t)), \tag{15}$$

differentiation with respect to time yields

$$\frac{\partial \hat{U}}{\partial t} = \sum_j \left[\frac{dU_j}{dt} \phi_j + U_j \frac{\partial \phi_j}{\partial t} \right]. \tag{16}$$

Use of (11) yields

$$\frac{\partial \hat{U}}{\partial t} = \sum_j \left[\frac{dU_j}{dt} \phi_j - U_j \mathbf{V}^e \cdot \nabla \phi_j \right] = \sum_j \frac{dU_j}{dt} \phi_j - \mathbf{V}^e \cdot \nabla \hat{U}. \tag{17}$$

The weighted residuals formulation thus becomes

$$\left\langle \left[\sum_j \frac{dU_j}{dt} \phi_j - \mathbf{V}^e \cdot \nabla \hat{U} + L\hat{U} - f \right], W_i \right\rangle = 0 \tag{18}$$

Since the space derivatives of $\hat{U}$ are independent of any node motion, Eq. (18) differs from its fixed-grid counterpart only in the term $\mathbf{V}^e \cdot \nabla \hat{U}$, which corrects the time derivatives to account for element deformation. Once the node motion is specified (thereby specifying both $\mathbf{V}^e$ and the domain of integration for the inner product), (18) may be evaluated exactly as in a fixed-grid treatment of the associated problem

$$\frac{\partial U}{\partial t} + L'U = f, \tag{19}$$

where

$$L' = L - \mathbf{V}^e \cdot \nabla. \tag{20}$$

Numerical integration of (18) in time may be formally stated as

$$\int \left\langle \left[\sum_j \frac{dU_j}{dt} \phi_j + L'\hat{U} - f \right], W_i \right\rangle \Omega^k \, dt = 0, \tag{21}$$

where typically $\Omega^k(t)$ spans only a limited portion of the time domain, allowing a stepwise solution in time.

This type of formulation has been used by Lynch and Gray [11], O'Neill and Lynch [21], and Lynch and O'Neill [12, 13] for both hyperbolic and parabolic problems; and by Mori [17, 18] for Stefan problems.

B. *Coordinate Transformation*

Another line of development which has been used to advantage involves first restating the governing equation(s) in terms of a more convenient set of space coordinates $\boldsymbol{\chi}$ which are moving with respect to the original reference frame $\mathbf{X}$. If the coordinate transformation is stated as

$$\mathbf{X} = \mathbf{X}(\boldsymbol{\chi}, t), \tag{22}$$

then a dependent variable U may be expressed as

$$U(\mathbf{X}, t) = U(\mathbf{X}(\boldsymbol{\chi}, t), t), \tag{23}$$

and the various time derivatives are related by the chain rule

$$\left.\frac{\partial U}{\partial t}\right|_{\chi} = \left.\frac{\partial U}{\partial t}\right|_{\mathbf{X}} + \nabla U \cdot \left.\frac{\partial \mathbf{X}}{\partial t}\right|_{\chi}, \tag{24}$$

where, as above, ∇ refers to the $\mathbf{X}$ system, holding t constant. If for clarity the time derivative in the $\boldsymbol{\chi}$ system, $\partial/\partial t|_{\chi}$, is identified as D/Dt, reserving the notation $\partial/\partial t$ for the $\mathbf{X}$ system, then substitution of (24) into the governing equation (13) yields

$$DU/Dt - (D\mathbf{X}/Dt) \cdot \nabla U + LU = f. \tag{25}$$

Discussions of the use of coordinate transformations in conjunction with various numerical methods are found for example in Mollowney [16]; Jensen [8]; Jensen and Finlayson [9]; Thompson [24]; and O'Neill [20].

Following O'Neill [20], finite element solution of (25) may be conveniently implemented by identifying the transformed coordinates $\boldsymbol{\chi}$ with the local coordinate system $\boldsymbol{\xi}$ on which the spatial interpolation functions are defined, such that the mapping is given in the form of (3). Assuming a numerical solution $\hat{U}$ as in (15), the time derivatives in (25) become

$$\frac{D\hat{U}}{Dt} = \sum_j \frac{dU_j}{dt} \phi_j, \tag{26}$$

$$\frac{D\mathbf{X}}{Dt} = \sum_j \frac{d\mathbf{X}_j}{dt} \psi_j = \mathbf{V}^e; \tag{27}$$

and the weighted residual statement of (25) is given by

$$\left\langle \left[\sum_j \frac{dU_j}{dt} \phi_j - \mathbf{V}^e \cdot \nabla \hat{U} + L\hat{U} \right], W_i \right\rangle = 0, \tag{28}$$

which is identical to (18). Integration through time may thus be expressed as in (21).

Note that the space derivatives in (25) and (28) may be expressed and evaluated in either the **X** domain or, following suitable transformation, in the ξ domain, without ambiguity. While in computational practice the latter would normally be chosen for higher-order elements, herein these expressions are retained in their original, untransformed form for clarity of presentation.

A subtle feature of (28) is the element of differential volume implied by the inner product notation. As written, (28) signifies integration over the original **X** domain. However, integration over the transformed domain would also be consistent with this approach. While the limits of integration coincide, the integrands in each case would differ by the presence or absence of the Jacobian determinant.

C. *Space-Time Elements*

Space-time elements have natural appeal for moving boundary problems, stemming from the observation that the motion of boundary nodes (or any interior nodes, for that matter) necessitates an irregular $(\mathbf{X}, t)$ grid. Most generally, the numerical solution on these elements is expressed as

$$\hat{U} = \sum_j \alpha_j N_j(\mathbf{X}, t), \tag{29}$$

where α_j are constants and the N_j are the space-time basis functions. Of interest herein is the usual case where the N_j can be expressed as products of time and space basis functions:

$$\hat{U} = \sum_j \sum_l \alpha_j^l \omega^l(t)\, \phi_j(\mathbf{X}, \mathbf{X}_1, ..., \mathbf{X}_n), \tag{30}$$

where the ϕ_j are defined exactly as above, and the temporal interpolation ω is independent of **X**. The associated *deformed prism* elements (Fig. 1) have the operational advantage that equal numbers of nodes in the same relative positions appear at sequential points in time. Consequently, node numbering, interpolation, matrix construction, and grid generation are simplied.

Equation (30) may be re-expressed as

$$\hat{U} = \sum_j \mathscr{U}_j(t)\, \phi_j, \tag{31}$$

where

$$\mathscr{U}_j = \sum_l \alpha_j^l \omega^l. \tag{32}$$

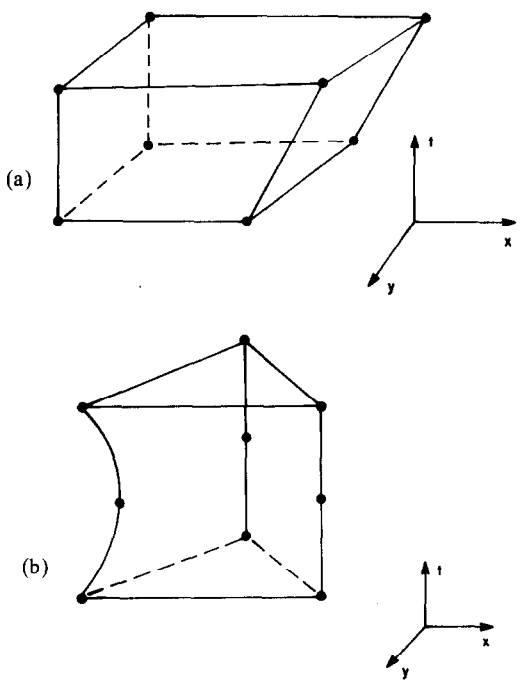

FIG. 1. (a) Bilinear element in (x, y); linear in t. (b) Linear element in (x, y); quadratic in t.

Since $\mathscr{U}_j(t)$ is the numerical solution at the point $\mathbf{X} = \mathbf{X}_j$, it is analogous to $U_j(t)$, the only distinction being that a particular functional form has been assumed for $\mathscr{U}_j$. Differentiation of (31) with respect to time yields a result analogous to (17)

$$\frac{\partial \hat{U}}{\partial t} = \sum_j \left[\frac{d\mathscr{U}_j}{dt} \phi_j + \mathscr{U}_j \frac{\partial \phi_j}{\partial t} \right] = \sum_j \frac{d\mathscr{U}_j}{dt} \phi_j - \mathbf{V}^e \cdot \nabla \hat{U}, \tag{33}$$

and the weighted residual statement of the governing equation (13) becomes

$$\int \left\langle \left[\sum_j \frac{d\mathscr{U}_j}{dt} \phi_j + L' \hat{U} - f \right], W_i \right\rangle \Omega^k \, dt = 0. \tag{34}$$

The space-time element approach thus leads to a problem statement which is formally the same as that developed in Section *A* above, Eq. (21), the only distinction being the appearance of the numerical analog $\mathscr{U}_j(t)$ in place of its more general form $U_j(t)$. To further illustrate the analogy between the two, consider the case of linear time dependence, i.e., the $\omega^l(t)$ are chapeau functions. In this simple but popular case, the expression

$$d\mathscr{U}_j/dt = \alpha_j^{l+1}(d\omega^{l+1}/dt) + \alpha_j^l(d\omega^l/dt) = (\alpha_j^{l+1} - \alpha_j^l)/(t^{l+1} - t^l) \tag{35}$$

holds everywhere within the elements $t^l < t < t^{l+1}$. Substitution of (35) into (34) thus yields an expression which could easily be obtained from (21) via conventional finite difference techniques.

Perhaps the best-known application of space-time elements to moving boundary problems is the work of Bonnerot and Jamet [2, 3], who solved one- and two-dimensional Stefan problems using linear elements. In the present terminology, they used the Galerkin method in space ($W_i = \phi_i$) and the subdomain method in time ($\Omega^k = 1$, $t^k \leqslant t \leqslant t^{k+1}$; $\Omega^k = 0$ elsewhere). Numerical integration was achieved with quadrature points at the nodes of each space-time element. Jamet and Bonnerot [7] used the same approach for one-dimensional compressible flow simulations, as did Varoğlu and Finn for convective-diffusive problems [26, 28] and Burgers' equation [29]. (The latter authors introduce node motion based on the characteristics of the associated hyperbolic equation.) Application to a two-dimensional transport problem is shown in Varoğlu [25].

D. *Error Minimization*

Miller and Miller [15] developed a one-dimensional deforming-grid approach based on the $\mathscr{L}^2$ norm of the residual $R = \partial\hat{U}/\partial t + L\hat{U} - f$

$$I\left(\{U_j\}, \left\{\frac{dU_j}{dt}\right\}\right) \equiv \langle R, R\rangle \tag{36}$$

Subsequent work by Alexander, *et al.* [1] extends the development to two space dimensions. Assuming a solution of the form (15), differentiation of I with respect to dU_i/dt, keeping the U_i fixed, yields the conditions

$$\left\langle \left[\sum_j \left(\frac{dU_j}{dt}\phi_j + \frac{d\mathbf{X}_j}{dt} \cdot \boldsymbol{\beta}_j\right) + L\hat{U} - f\right], \phi_i \right\rangle = 0, \tag{37}$$

where

$$\boldsymbol{\beta}_j = \nabla_j \hat{U} = \sum_i U_i \nabla_j \phi_i. \tag{38}$$

Use of (8) yields

$$\boldsymbol{\beta}_j = -\sum_i U_i \nabla\phi_i \psi_j = -\psi_j \nabla\hat{U}, \tag{39}$$

and thus (37) may be restated as

$$\left\langle \left[\sum_j \left(\frac{dU_j}{dt}\phi_j - \frac{d\mathbf{X}_j}{dt}\psi_j \cdot \nabla\hat{U}\right) + L\hat{U} - f\right], \phi_i \right\rangle = 0. \tag{40}$$

Noting the definition in Eq. (12) of $\mathbf{V}^e$, Eq. (40) can be seen to be identical to the Galerkin version of (18). (Note that the integration in (36) is over the spatial domain

only, while a classical least-squares approach would involve integration over time as well and subsequent application of the calculus of variations.)

Extension to systems of equations is demonstrated by Gelinas, *et al.* [5], wherein the authors consider the weighted sum of the squared $\mathscr{L}^2$ norms of the residuals R^l, where

$$R^l = \frac{\partial \hat{U}^l}{\partial t} + L^l \hat{U} - f^l. \tag{41}$$

Differentiation with respect to dU_i^l/dt while holding the U_i^l constant yields, regardless of the weights,

$$\left\langle \left[\sum_j \left(\frac{dU_j^l}{dt} \phi_j + \frac{d\mathbf{X}_j}{dt} \cdot \boldsymbol{\beta}_j^l \right) + L^l \hat{U} - f^l \right], \phi_i \right\rangle = 0, \tag{42}$$

where

$$\boldsymbol{\beta}_j^l = \nabla_j \hat{U}^l. \tag{43}$$

As above, it is easy to show that

$$\boldsymbol{\beta}_j^l = -\psi_j \nabla \hat{U}^l, \tag{44}$$

and thus (42) reduces to

$$\left\langle \left[\sum_j \frac{dU_j^l}{dt} \phi_j - \mathbf{V}^e \cdot \nabla \hat{U}^l + L^l \hat{U} - f \right], \phi_i \right\rangle = 0. \tag{45}$$

Thus, if all node motion is externally specified, then this approach to error minimization on a deforming grid produces the Galerkin formulation. (The results (39) and (44) can be seen to be implicit in the formulae for $\boldsymbol{\beta}_j^l$ noted in the above references, for the particular cases of one- and two-dimensional linear elements.)

The novel feature of the significant development by Miller, *et al.*, is minimization with respect to node motion as well as dU_i/dt. For nodes whose motion is unconstrained, differentiation with respect to $d\mathbf{X}_i/dt$ while holding the $\mathbf{X}_i$ constant yields the weighted residual expressions

$$\langle R, \boldsymbol{\beta}_i \rangle = 0 \tag{46}$$

for the scalar case. In light of (39), this can be stated as

$$\left\langle \left[\sum_j \frac{dU_j}{dt} \phi_j - \mathbf{V}^e \cdot \nabla \hat{U} + L\hat{U} - f \right], \psi_i \nabla \hat{U} \right\rangle = 0. \tag{47}$$

Thus, node motions which are not otherwise specified are determined by solution of (47) in conjunction with (40). This intrinsic feature is not shared by any of the other

approaches considered herein, in which separate, additional considerations must be invoked to specify the motion of unconstrained nodes.

In the case of a system of governing equations, the minimization produces a weighted sum of individual components as in (46)

$$\sum_I w_I \langle R^I, \beta_i^I \rangle = 0, \tag{48}$$

where w_I are the externally specified weights and β_i^I is defined above.

The potential, noted by Miller and Miller [15], for Eqs. (40) and (46) to become indeterminate can be seen herein to apply to a broad class of multidimensional elements. The indeterminacy arises when the functions ϕ_i and β_i are not linearly independent. In view of (39), this will be true when at least one component of $\nabla \hat{U}$ is constant over the support space for any of the ψ_j. (This is most clearly demonstrated in the isoparametric case $\phi_j = \psi_j$, and is readily extended to the super- and subparametric cases, where one of the sets ϕ_i or ψ_i is a linear combination of the other.) In the case of systems of equations, indeterminacy will result when the above is true simultaneously for all $\nabla \hat{U}^I$.

Miller [14] introduced additional control over mesh motion by adding to the $\mathscr{L}^2$ norm a penalty function containing terms analogous to internodal viscoelastic forces, and minimizing the sum. This additional degree of mesh control has no effect on the Galerkin equations (40) and (42); additional terms are added to the equations for mesh motion (46)–(48) which eliminates the potential indeterminacy. No general approach seems to be available for parameterization of the penalty function, and thus the nature of the quantity being minimized becomes less clear in this case. Nevertheless, the results obtained by Miller [14] and Gelinas *et al.* [5] provide a compelling demonstration of the capabilities of simulation on deforming elements.

E. *Synthesis*

It is shown above that four different approaches to deforming grid simulation can be formally stated in the single unified formulation (18) or (21) for a broad class of multidimensional elements. The intuitive appeal of this formulation lies in the fact that the original PDE structure is evident in weighted residual form, with the addition of an extra *Lagrangian* term to account for the intrinsic effects of node motion. The analyst may choose to interpret this correction term in the context of any of the four approaches—the point here is that (18) may be stated at the outset, thus proceeding immediately and without error to a relatively high level of problem formulation. The algebraic equations actually solved then depend strictly on conventional choices of element type, weighting functions, and quadrature, given a suitable specification of node motion.

Operationally, intuitive appeal brings benefits of programming simplicity. The unified approach makes published pages of algebraic forms corresponding to specific operators, elements, quadrature, etc., unnecessary, and indeed makes simulation on complex, multidimensional deforming elements feasible. Programming may proceed

directly from Eq. (18) using conventional finite element apparatus, since the Lagrangian correction is stated in familiar, easily computable gradient form. Adaptation of existing fixed-grid programs is similarly facilitated.

Specification of Node Motion

In addition to the dependent variables of the original, continuous problem, deforming grid simulation implies several numerical degrees of freedom associated with the trajectories of moving nodes. While the formulation (18) incorporates the effects of this motion on the dependent variable equations, it does not generate additional equations to specify the motion. Automatic control of node motion is the second significant departure from conventional fixed-grid methods, and significant methodological diversity exists in this area.

For nodes which lie on a moving boundary or internal shock, a boundary or *jump* condition will normally be available which will specify at least the normal component of these node motions. Nodes which lie on fixed boundaries either have their positions fixed or are constrained to move parallel to a fixed boundary. Additionally, some node placement may be dictated by the need for accurate representation of coefficients or source terms exhibiting steep gradients, discontinuities, or singularities which are fixed in space or which move along known trajectories. Nevertheless, once these constraints on node motion are satisfied, one can expect to have additional unspecified degrees of freedom.

Several possibilities have been explored for specifying node motion; most are problem-dependent and rely heavily on one's intuition regarding the form of the solution. For example, Lynch and Gray [10, 11] allow node motion only at moving boundaries—all interior nodes remain fixed in space. The moving boundary condition prescribes the value of node velocity normal to the boundary, and in order to reduce element shearing, the tangential boundary node velocity is set equal to zero. It is apparent that this approach is useful only when boundary movement is relatively small.

O'Neill and Lynch [21] and Lynch and O'Neill [12] solved several one-dimensional Stefan problems wherein large displacements occur and all nodes not attached to the fixed boundary are in motion. The phase boundary node location $S(t)$ is calculated from the boundary condition; all interior nodes are then moved proportionally

$$X_i(t) = S(t)(X_i(0)/S(0)). \tag{49}$$

Bonnerot and Jamet [3] employ a similar criterion for node movement in two dimensions. Since the primary boundary motion is in one coordinate direction, no node motion is allowed in the other direction throughout the grid. Thus, the degrees of freedom associated with node positions are halved. The initial grid is composed of "strings" of nodes which are stretched individually in a way similar to Eq. (49) as the

simulation progresses. Neumann and Witherspoon [19] employ a similar procedure in two dimensions, although the directional orientations of the individual strings of nodes are allowed to change through time. Santos [23] retriangulates the domain contained by a moving boundary after each time step, subject to the constraint that mesh topology remain constant.

Varoğlu and Finn [26, 28, 29] move the nodes along characteristic lines of hyperbolic equations; this procedure requires that nodes be created and/or removed from time to time. Jensen [8] and Jensen and Finlayson [9] translate the undeformed mesh to follow steep fronts in the computed solution, requiring similar adjustments at fixed boundaries. O'Neill [20] concentrates nodes in the vicinity of steep one-dimensional fronts based on consideration of the numerical Peclet number in the deforming coordinate system; element stretching is allowed in regions where solution gradients are small and near fixed boundaries.

Most of these methods rely on an intuitive, a priori knowledge of the solution in order to simplify the problem of node location and movement. Thus, for complex problems, a poor solution could result due to awkwardly or excessively constrained nodes. A more general approach is needed when foreknowledge of the solution is lacking or when intuitive rules for node motion are too complex to implement manually. Such a general approach should exhibit at least three qualities: (1) It should be capable of generating node locations given the minimum amount of information, i.e., the location of fixed nodes and the total or normal displacements of shocks or moving boundary nodes. (2) Bunching of nodes, element shearing, and general tangling of the grid should be avoided. (3) The computations should be capable of automation within the usual finite element time stepping procedure.

The minimization technique of Miller [14] generally exhibits these qualities, although the use of the penalty function to avoid degeneracy appears to be somewhat arbitrary, relying upon intuitive notions of interelement forces. An alternative which will be pursued herein is a generalization of the grid-stretching approach (Eq. 49): the grid is constrained to behave as a linearly elastic solid. Node displacements from their initial positions thus correspond to states of strain which are uniquely specified given total or normal displacements of all boundary nodes. This approach will in a sense minimize mesh irregularities, and the computations of node displacements are readily automated using standard finite element procedures. In some applications, the elasticity equations may be solved on the initial, undeformed grid, in which case a single assembly and upper-triangularization of the stiffness matrix will suffice for a (potentially) large number of time steps. While this approach is most applicable to moving boundary problems, internal discontinuities or steep gradients can be dealt with by partitioning the domain into separate elastic subdomains, and by judicious variation of elastic properties.

Formulation for Phase Change Problems

To illustrate the synthesis of these concepts into a computational algorithm, consider the one-phase Stefan problem in two space dimensions. On the interior of the domain, the temperature field T is governed by the heat equation:

$$C\frac{\partial T}{\partial t} - \nabla \cdot K\nabla T = 0, \tag{50}$$

where C and K are thermal properties. Initial conditions and boundary conditions on fixed boundaries are those normally associated with this equation. The distinguishing feature of the problem is the presence of a moving phase boundary, where conditions are imposed on both T and ∇T

$$T = 0, \tag{51}$$

$$K(\nabla T \cdot \mathbf{n}) = L(\mathbf{V} \cdot \mathbf{n}), \tag{52}$$

where $\mathbf{V}$ is the motion of the boundary, $\mathbf{n}$ is the unit vector normal to the boundary, and L is a thermal property.

The Galerkin formulation for (50) follows directly from (18). Integration of the heat flux term by parts yields

$$\sum_j \langle C\phi_j, \phi_i\rangle (dT_j/dt) - \langle C\mathbf{V}^e \cdot \nabla\phi_j, \phi_i\rangle T_j + \langle K\nabla\phi_j, \nabla\phi_i\rangle T_j$$

$$= \oint K\nabla T \cdot \mathbf{n}\phi_i \, dS \qquad i = 1,\ldots, N. \tag{53}$$

Boundary conditions on fixed boundaries are handled in the usual way. If node i lies on a Dirichlet boundary, then equation i is sacrified in favor of direct specification of $T_i(t)$. Neumann boundary conditions are recognized in the surface integral. The moving phase boundary is treated as a Dirichlet boundary. A conventional two-level implicit finite difference procedure is used to integrate (53) through time. The inner products are evaluated at time $t + \varepsilon\Delta t$ $(0 \leqslant \varepsilon \leqslant 1)$, as are all spatial gradients. The time derivatives of T_j and $\mathbf{X}_j$ (embodied in $\mathbf{V}^e$) are approximated by simple differences across the interval Δt.

Phase boundary motion must be determined in accordance with (52). Since this equation cannot be satisfied everywhere on the discretized boundary, a weaker integral form is imposed:

$$L\int \sum \mathbf{V}_i\phi_i \cdot \mathbf{n} \, dS = \int K\nabla T \cdot \mathbf{n} \, dS = \int \sum \phi_i K\nabla T \cdot \mathbf{n} \, dS, \tag{54}$$

where the integration is over the phase boundary and the summation is over phase

boundary nodes only. Satisfying (54) term-by-term gives a *local* allocation of the computed heat flux among the boundary nodes

$$L\mathbf{V}_i \cdot \int \phi_i \mathbf{n}\, dS = \int \phi_i K \nabla T \cdot \mathbf{n}\, dS \tag{55}$$

or

$$L\mathbf{V}_i \cdot \mathbf{n}_i = f_i, \tag{56}$$

where $\mathbf{n}_i$ and f_i are the weighted average of the unit normal and the heat flux

$$\mathbf{n}_i = \int \phi_i \mathbf{n}\, dS \Big/ \int \phi_i\, dS, \tag{57}$$

$$f_i = \int \phi_i K \nabla T \cdot \mathbf{n}\, dS \Big/ \int \phi_i\, dS. \tag{58}$$

The normal components of the phase boundary velocities are thus obtained from the computed heat flux; the tangential components may be arbitrarily specified to suit the particular application. In some cases it will be advantageous to specify no node motion tangent to the phase boundary; in other cases the tangential motion may be left unconstrained.

On fixed boundaries, the natural conditions are either no motion or no motion normal to the boundary.

Interior node motion is governed by the equations for elastic displacement in plane stress with constant material properties

$$\frac{\partial^2 \dot{U}}{\partial x^2} + \frac{v\partial^2 \dot{V}}{\partial x \partial y} + \left(\frac{1-v}{2}\right)\left[\frac{\partial^2 \dot{U}}{\partial y^2} + \frac{\partial^2 \dot{V}}{\partial x \partial y}\right] = 0, \tag{59}$$

$$\frac{\partial^2 \dot{V}}{\partial y^2} + \frac{v\partial^2 \dot{U}}{\partial x \partial y} + \left(\frac{1-v}{2}\right)\left[\frac{\partial^2 \dot{V}}{\partial x^2} + \frac{\partial^2 \dot{U}}{\partial x \partial y}\right] = 0, \tag{60}$$

where $(\dot{U}, \dot{V})$ is the displacement rate of a point in the mesh. Galerkin treatment of (59) and (60) subject to the above boundary conditions yields equations for the interior nodal velocities $\mathbf{V}_i$. The Poisson ratio v is arbitrarily set at zero, corresponding to a maximum shear modulus. On boundaries where elastic tangential motion is allowed, the tangential force is assumed to vanish.

In special cases, the stiffness matrix associated with (59) and (60) may be kept stationary by continually solving on an initial, undeformed mesh. A necessary condition for this economy is that the orientation of boundaries where tangential motion is allowed be constant. (This will not generally be the case when such motion is allowed on a moving phase boundary.) Of course, large displacements from an initial, undeformed mesh will ultimately require that the stiffness matrix be updated.

Integration of the nodal velocities is achieved within a time step by

$$d\mathbf{X}_i/dt = (\mathbf{X}_i^{t+\Delta t} - \mathbf{X}_i^t)/\Delta t = \theta \mathbf{V}_i^{t+\Delta t} + (1-\theta)\,\mathbf{V}_i^t. \tag{61}$$

The time weighting parameter θ is intentionally different from ε, to provide additional numerical flexibility.

When $\theta = 0$, a typical time step may be achieved as follows: At the known time level t, the phase boundary velocities are calculated from (56), and the elasticity equations are solved for all remaining node velocities. An explicit projection is then possible for the node locations at time $t + \Delta t$, using (61), and the term $\mathbf{V}^e$ is determined over the interval Δt. Solution of (53) may then be achieved and a new time step begun.

For $\theta \neq 0$, iteration is inevitable since evaluation of (61) requires unknown temperature information at $t + \Delta t$. Herein a simple predictor-corrector procedure is used. Initially, θ is taken as zero (or equivalently, $\mathbf{V}_i^{t+\Delta t}$ is assumed equal to $\mathbf{V}_i^t$) and the above procedure is executed. The resulting solution at $t + \Delta t$ is then used to recompute (61) with $\theta \neq 0$ and the steps are repeated. Further iteration is not pursued.

Results

Preliminary experiments with this model are reported by Lynch and O'Neill [13]. Herein two problems are solved which have analytic solutions for comparison and which involve large boundary displacement. In both cases, linear triangular elements are used with exact integration of the inner products, and the predictor-corrector scheme is used with $\theta = \varepsilon$.

A. *Freezing Around a Pipe*

The first case involves radially symmetric freezing around a pipe of finite radius, surrounded by an infinite unfrozen medium. An analytic solution is available [4, p. 295] for the case of a constant point sink of heat at $r = 0$, $t > 0$; initial conditions require uniform initial temperature with no frozen phase. The finite-radius pipe test case is obtained by prescribing as a boundary condition the temperature at the pipe boundary r_0 (Fig. 2) as calculated from the analytic solution. (Note that this implies a time-dependent boundary condition.) Simulation is initiated at that point in time when the phase boundary (analytically) is at $r = r_0 + \tau$, with an assumed linear variation of T with r (a reasonable but not precise initial temperature distribution). This problem was solved on the quarter-space illustrated in Fig. 2, with no heat flux assumed on the symmetry boundaries. The physical parameters used are as folows: $C = 0.5083$ cal/cm^3; $K = 0.0072$ cal/cm sec °C; $L = 33.012$ cal/cm^3; and $Q = 0.10$ (heat flux at $r = 0$ for the analytic solution); $r_0 = 3.0$ cm; and $\tau = 0.1$ cm (initial frozen thickness).

The finite element grid is illistrated in Fig. 3. Grid motion was managed as follows:

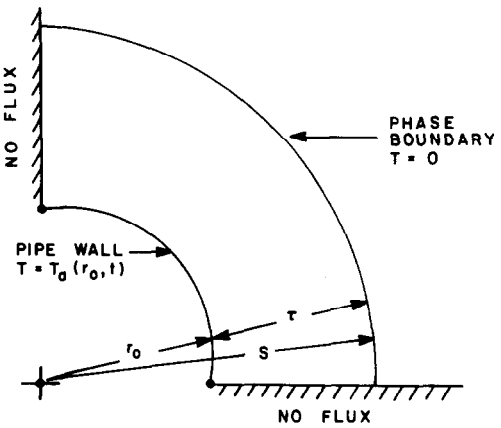

FIG. 2. Geometry for test case A: freezing around a pipe.

no motion at $r = r_0$,

no tangential motion on the phase boundary,

no normal motion along the symmetry boundaries.

The elasticity equations were formulated once on the initial, undeformed grid ($r_0 = 3.0$; $\tau = 0.1$) and the resulting stiffness matrix was left unchanged throughout the simulations. The time step was continually adjusted such that during each step the frozen thickness would increase by 20%, based upon the phase boundary velocity calculated in the previous time step.

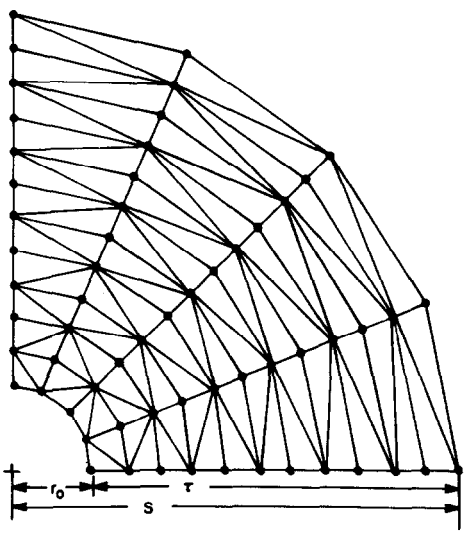

FIG. 3. Finite element grid for test case A. Radial coordinates are distorted for clarity of presentation; actual grid corresponds to initial conditions, with $r_0 = 3$ cm and $\tau = 0.1$ cm.

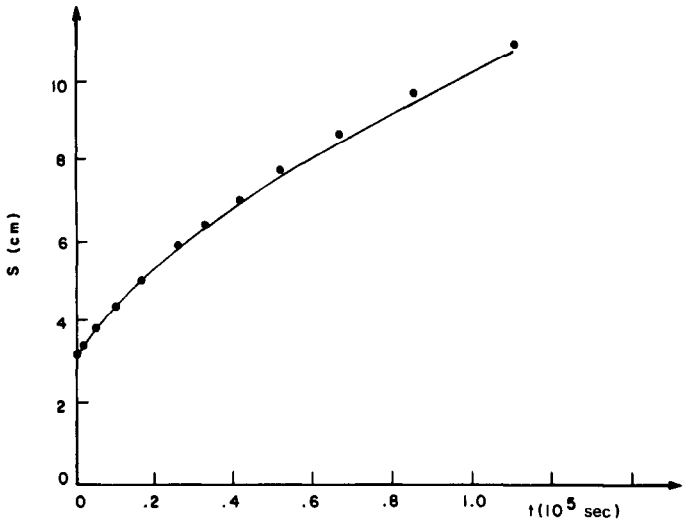

FIG. 4. Computed and analytic (—) values of the phase boundary location through 25 time steps, test case A, $\varepsilon = \theta = 0.5$.

Computed and analytic results are shown in Figs. 4–7 for the case $\varepsilon = \theta = 0.5$. At larger times, a persistent over-prediction of the phase boundary location develops, with a secondary effect on the temperature distribution. For $\theta = \varepsilon = 1.0$ (Figs. 8–11), very good accuracy is apparent.

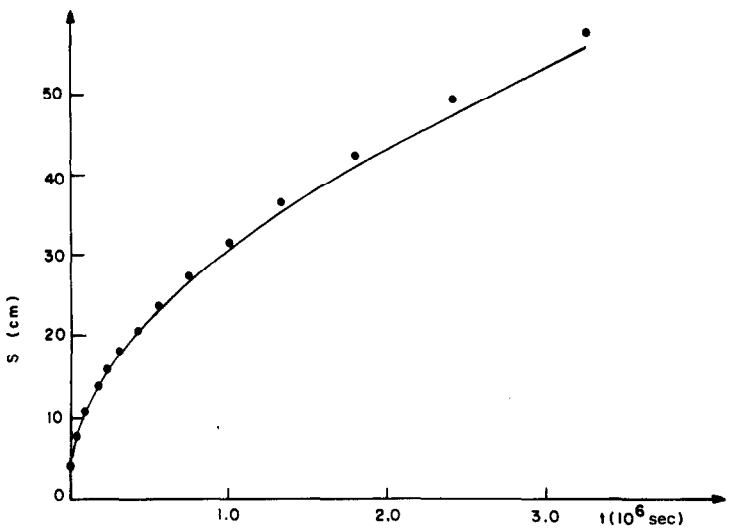

FIG. 5. Computed and analytic (—) values of the phase boundary location through 37 time steps, test case A, $\varepsilon = \theta = 0.5$.

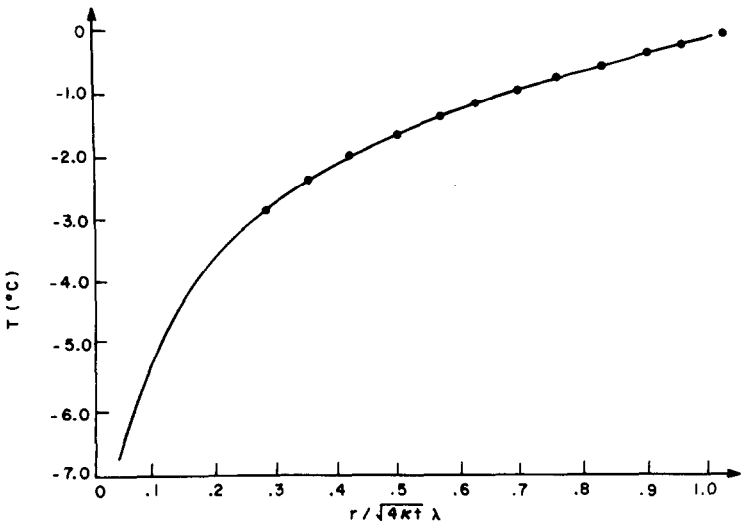

FIG. 6. Computed and analytic (—) temperature distribution after 25 time steps, test case A, $\varepsilon = \theta = 0.5$, $\kappa = K/C$; $\lambda = 0.12935$.

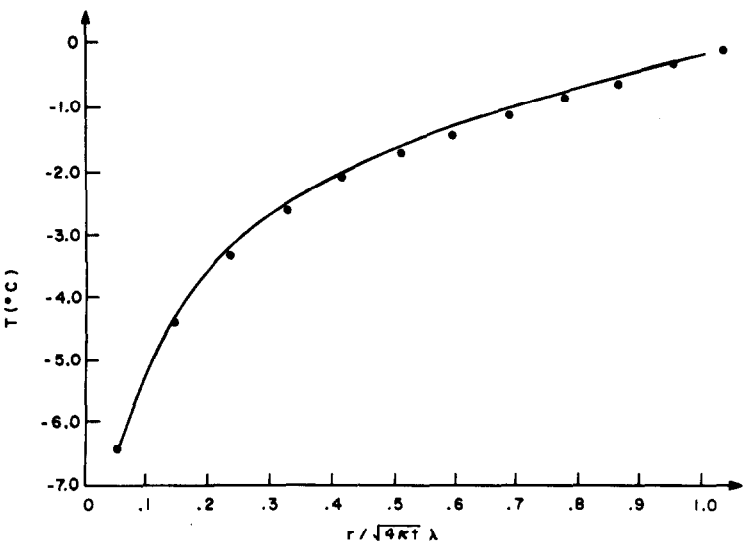

FIG. 7. Computed and analytic (—) temperature distribution after 37 time steps, test case A, $\varepsilon = \theta = 0.5$.

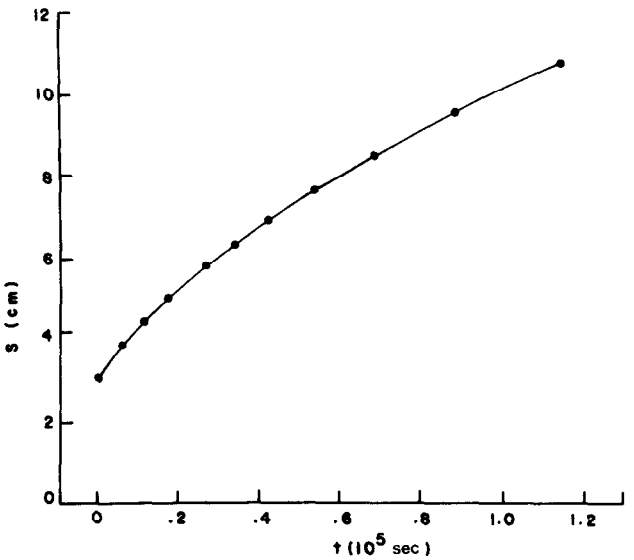

FIG. 8. Computed and analytic (—) values of the phase boundary location through 25 time steps, test case A, $\varepsilon = \theta = 1.0$.

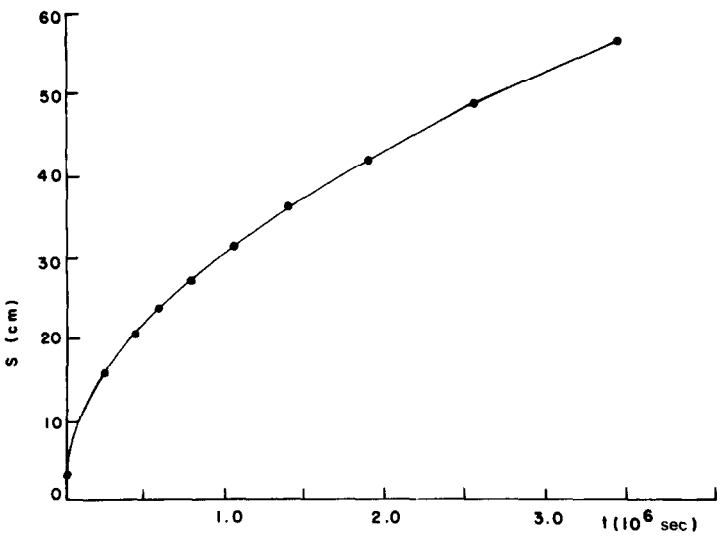

FIG. 9. Computed and analytic (—) values of the phase boundary location through 37 time steps, test case A, $\varepsilon = \theta = 1.0$.

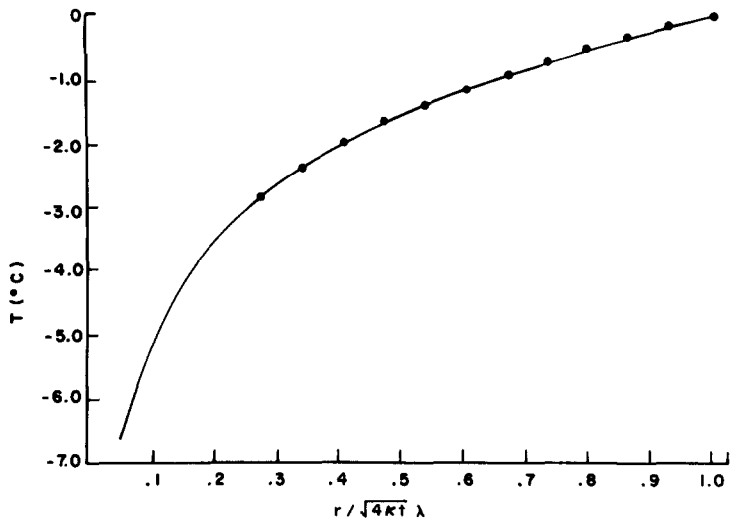

FIG. 10., Computed and analytic (—) temperature distribution after 25 time steps, test case A, $\varepsilon = \theta = 1.0$.

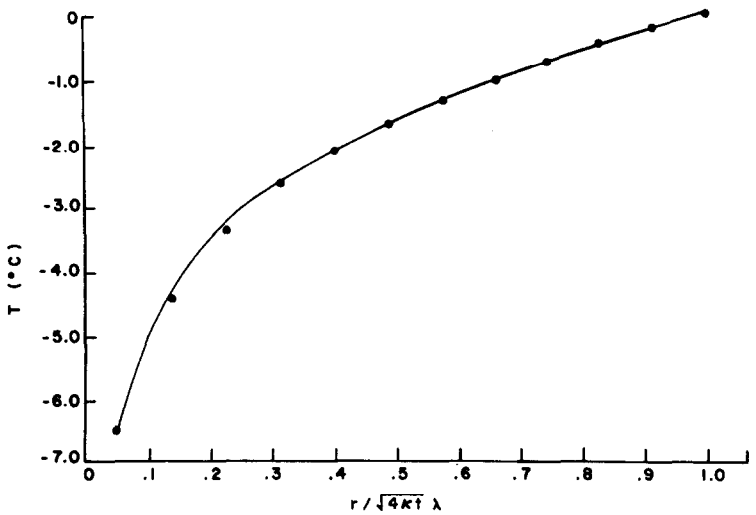

FIG. 11. Computed and analytic (—) temperature distribution after 37 time steps, test case A, $\varepsilon = \theta = 1.0$.

Note that in this case, the frozen thickness containing all of the finite elements increases from 0.1 cm to 5.4 cm over only 37 time steps, with no disturbing consequences. Nodal displacements and computed temperature gradients are essentially zero in the circumferential direction. Radial mesh deformation is practically uniform, yielding a satisfying deployment of interior nodes throughout the simulation (note the node point locations on Figs. 10 and 11).

B. *Freezing in a Corner*

The analytic solution for this test problem is given by Rathjen and Jiji [22] for the case of uniform wall temperature along $x = 0$ and $y = 0$, with boundaries at infinity. Initial conditions are constant temperature everywhere with no solid phase. This problem was solved using material properties as above, with wall temperature $= -10\,°C$.

Numerical initial conditions assume a small frozen zone varying in thickness from 0.2 cm at the semi-infinite end to 0.24 cm near the corner (Fig. 12), which appproximates the analytic values at $t = 10.0$ sec. Initial temperature variation was linear in y, with suitable adjustment in the vicinity of the corner. Grid motion was controlled as follows:

along the fixed boundaries at $y = 0$ and $y = x$, no normal motion and free tangential motion;

along the phase boundary, free tangential motion;

along the semi-infinite boundary, free tangential motion.

In order to keep the semi-infinite boundary sufficiently ahead of the advancing freezing front, the normal motion of this boundary was arbitrarily specified such that x^2/t remained constant. The stiffness matrix was reformulated during each time step. As above, Δt was continually adjusted such that the frozen thickness at the semi-infinite end increased by 10% during each step.

Computed node locations after 14 and 40 time steps are displayed in Figs. 13 and 14, for $\varepsilon = \theta = 0.6$. After 40 steps, the frozen zone had increased in thickness (y dimension) by about a factor of 35. The phase boundary values of $y/\sqrt{t}$ at the semi-infinite end and at the corner line $y = x$ reached asymptotes of 0.0645 and 0.0727 respectively, which agree well with the analytic solution.

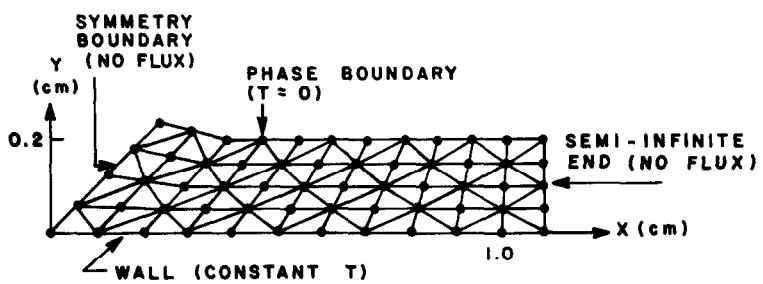

FIG. 12. Geometry and initial grid for test problem B: freezing in a corner.

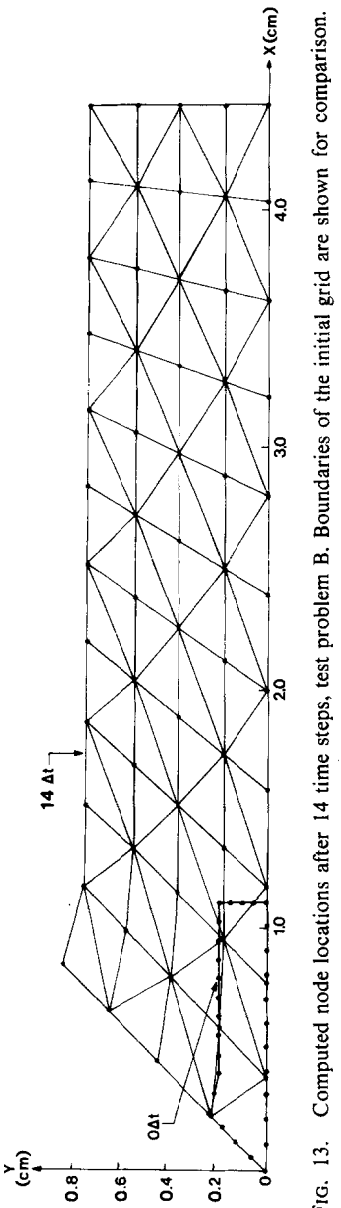

FIG. 13. Computed node locations after 14 time steps, test problem B. Boundaries of the initial grid are shown for comparison.

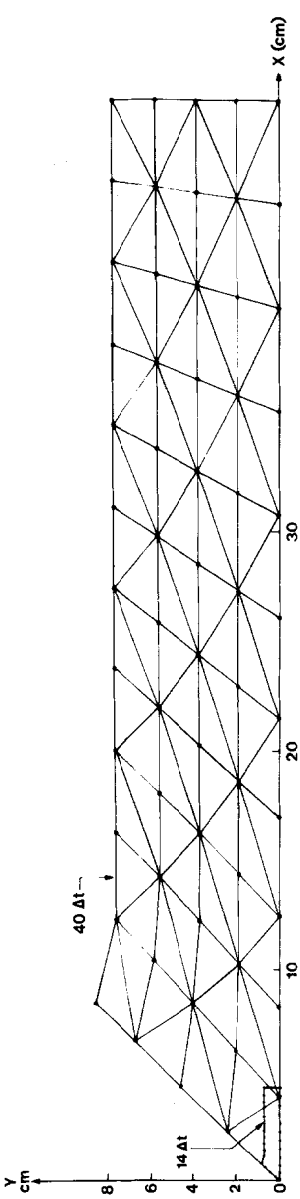

FIG. 14. Computed node locations after 40 time steps, test problem B. Boundaries of the grid at 14 time steps are shown for comparison.

CONCLUSIONS

It is shown above that several good, proven approaches to problem formulation on deforming elements can be expressed in a single, unified weighted residuals statement of the equations governing dependent variable evolution. This unified formulation accounts for mesh motion in an intuitively appealing way and is easily implemented. Significantly, existing multidimensional programs which treat only fixed-grid problems may be converted to more general service without major reorganization. Among the approaches considered, differences remaining in specific applications involve the important problem of node control strategy, and the choice of conventional numerical parameters—elements, quadrature, time stepping, etc. The significant methodological diversity concerning node control is a subject worthy of continuing attention. In particular, emerging methods based on error minimization appear quite promising in terms of their generality.

The entire discussion is restricted to simulations in which the finite element topology remains constant. Of course, this strategy is not perfectly general. There will always be situations in which an unacceptably distorted grid evolves, and rezoning is called for. The results presented herein demonstrate that, at least in the context of phase change problems, large element deformations and deformation rates are possible in two dimensions without adverse consequences. The elastic grid approach provides, in this limited context, a good and workable procedure for node motion control.

ACKNOWLEDGMENTS

The numerical results presented herein were facilitated by collaboration with Dr. Kevin O'Neill of the U.S. Army Cold Regions Research and Engineering Laboratory, Hanover, N.H. His contributions to the present work as well as in the earlier, formative stages of this research have been essential. This work has been supported in part by the National Science Foundation, Grant No. ENG-7908076.

REFERENCES

1. R. ALEXANDER, P. MANSELLI, K. MILLER, AND S. G. SANSONE, *Rend. Sci. fis. mat. e nat.*, LXVII (1979), 57.
2. R. BONNEROT AND P. JAMET, *Int. J. Numer. Methods Eng.* **8** (1974), 811.
3. R. BONNEROT AND P. JAMET, *J. Comput. Phys.* **25** (1977), 163.
4. H. S. CARSLAW AND J. C. JAEGER, "Conduction of Heat in Solids," 2nd ed., pp. 295–296, Clarendon, Oxford, 1959.
5. R. J GELINAS, S. K. DOSS, AND K. MILLER, *J. Comput. Phys.* **40** (1981), 202.
6. V. GUVANASEN AND R. E. VOLKER, *Int. J. Numer. Methods Eng.* **15** (1980), 1643.
7. P. JAMET AND R. BONNEROT, *J. Comput. Phys.* **18** (1975), 21.
8. O. K. JENSEN, "Numerical modeling with a moving coordinate system: application to flow through porous media," Ph.D. thesis, Univ. of Washington, 1980.
9. O. K. JENSEN AND B. A. FINLAYSON, *in* "Finite Elements in Water Resources III" (Wang *et al.*, Eds.), pp. 5.169–5.180, Univ. of Mississippi–Oxford, 1980.

10. D. R. LYNCH AND W. G. GRAY, *in* "Finite Elements in Water Resources II" (Brebbia *et al.*, Eds.), pp. 2.23–2.42, Pentech, London, 1978.
11. D. R. LYNCH AND W. G. GRAY, *J. Comput. Phys.* **36** (1980), 135.
12. D. R. LYNCH AND K. O'NEILL, *Int. J. Numer. Methods Eng.* **17** (1981), 81.
13. D. R. LYNCH AND K. O'NEILL, *in* "Finite Elements in Water Resources III" (Wang *et al.*, Eds.), pp. 7.111–7.120, Univ. of Mississippi–Oxford, 1980.
14. K. MILLER, *SIAM J. Numer. Anal.* **18** (1981), 1033.
15. K. MILLER AND R. N. MILLER, *SIAM J. Numer. Anal.* **18** (1981), 1019.
16. B. M. MOLLOWNEY, *in* "Mathematical Models for Environmental Problems" (Brebbia, Ed.), pp. 313–327, Pentech, London, 1976.
17. M. MORI, *in* "Proceedings, IFIP Conf. on Modeling of Environmental Systems, Tokyo," pp. 167–171, 1976.
18. M. MORI, *Publ. Res. Inst. Math. Sci., Kyoto Univ.* **13** (3) (1977), 723.
19. S. P. NEUMANN AND P. A. WITHERSPOON, *Water Resour. Res.* **7** (1971), 611.
20. K. O'NEILL, *Water Resour. Res.* **17** (1981), 1665.
21. K. O'NEILL AND D. R. LYNCH, *in* "Numerical Methods in Heat Transfer" (R. W. Lewis, K. Morgan, and O. C. Zienkiewicz, Eds.), Wiley, New York, 1981.
22. K. A. RATHJEN AND L. M. JIJI, *J. Heat Transfer*, (1971), 101.
23. V. R. B. SANTOS, *Comput. Methods Appl. Mech. Eng.* **25** (1981), 51.
24. J. F. THOMPSON, *in* "Computational Fluid Mechanics" (W. Kollman, Ed.), pp. 1–98, Hemisphere, Washington, 1980.
25. E. VAROĞLU, *in* "Finite Elements in Water Resources III" (Wang *et al.*, Eds.), pp. 3.3–3.19, Univ. of Mississippi–Oxford, 1980.
26. E. VAROĞLU AND W. D. L. FINN, *Advances in Water Resources* **1** (1978), 337.
27. E. VAROĞLU AND W. D. L. FINN, *Comput. Fluids* **6** (1978), 103.
28. E. VAROĞLU AND W. D. L. FINN, *J. Comput. Phys.* **37** (1980), 371.
29. E. VAROĞLU AND W. D. L. FINN, *Int. J. Numer. Methods Eng.* **16** (1980), 171.