MULTIDIMENSIONAL ASTROPHYSICAL STRUCTURAL AND DYNAMICAL ANALYSIS.
I. DEVELOPMENT OF A NONLINEAR FINITE ELEMENT APPROACH

D. L. MEIER
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109
Received 1998 June 4; accepted 1998 November 24

ABSTRACT

A new field of numerical astrophysics is introduced that addresses the solution of large multidimensional structural or slowly evolving problems (rotating stars, interacting binaries, thick advective accretion disks, four-dimensional spacetimes, etc.), as well as rapidly evolving systems. The technique employed is the finite element method (FEM), which has been used to solve engineering structural problems for more than three decades. The approach developed here has the following key features:

1. The computational mesh can extend into the time dimension, as well as space—generally, only a few cells deep for most (flat-space) astrophysical problems, but throughout spacetime for solving Einstein’s field equations.

2. When time is treated as a mesh dimension, virtually all equations describing the astrophysics of continuous media, including the field equations, can be written in a compact form similar to that routinely solved by most engineering finite element codes (albeit for nonlinear equations in a four-dimensional spacetime instead of linear ones in two or three space dimensions): the divergence of a generalized stress tensor equals a generalized body-force vector, both of which are functions only of position, the state variables, and their gradients.

3. The transformations that occur naturally in the four-dimensional FEM possess both coordinate and boost features, such that: (A) although the computational mesh may have a complex, nonanalytic, curvilinear structure, and may be adapted to the geometry of the problem, the physical equations still can be written in a simple coordinate system that is independent of the mesh structure; and (B) if the mesh has a complex flow velocity with respect to coordinate space, the transformations will form the proper advective derivatives, automatically converting the equations to arbitrary Lagrangian-Eulerian.

4. Only relatively simple differential equations need to be encoded. The complex difference equations on the arbitrary curvilinear grid are generated automatically by the FEM integrals. A different integration method must be used for equations of odd and even orders.

This first paper concentrates on developing a robust and widely applicable set of techniques using the nonlinear FEM, and presents some examples. The second paper in this series will deal with making the method fast and efficient, so that large, astrophysically interesting computational meshes can be employed.

Subject headings: hydrodynamics — methods: numerical — MHD — relativity — stars: rotation

1. INTRODUCTION

The first problem to be solved with the techniques of numerical astrophysics was the structure and evolution of stars—an “implicit” problem that involves a static or slowly evolving structure (Chandrasekhar 1957; Aller & McLaughlin 1965). Its solution consists of determining the state variables of the fluid (density, temperature, pressure, flux of radiation, and composition) at each radius in the stellar interior, and is obtained by relaxing a large set of coupled nonlinear difference equations (derived from the differential equations of stellar structure) along with boundary conditions at the stellar center and surface. Radial stellar structure is only a one-dimensional problem, and, while once considered difficult and CPU-intensive, it is now solved easily on personal computers. Since then, many other fields of numerical astrophysics have been developed: “explicit” hydrodynamic simulations of explosive and jet phenomena (Norman 1997); N-body and smooth-particle hydrodynamics (SPH) of discrete or semidiscrete systems of particles (Dubinski & Hernquist 1997; Monaghan 1992); Monte Carlo simulations of radiation flow (Leahy 1997; Park & Hong 1998); etc. All have matured to the point where the solution of three-dimensional, time-dependent problems is not uncommon.

Ironically, however, only modest progress has been made in extending the original implicit problems into several dimensions: rapidly rotating stars, evolving and interacting binaries, detailed accretion disk structure and evolution, etc. There are several important reasons for this. First, the geometry of these systems is unknown until the problem is solved. For example, the shape of the outer surface of a rotating star or (possibly thick and advective) accretion disk will be part of the solution, and the shape of a rapidly rotating stellar core may have a different oblateness (or even prolateness) from that of the outer envelope. No numerical method capable of operating in nearly arbitrary geometries has been applied extensively in astrophysics. Instead, one either has assumed spherical symmetry and treated only slowly rotating perturbation problems (Kippenhahn & Thomas 1970), or has assumed that the isosurfaces of the state variables are coincident (which implies rotation on cylinders via von Zeipel’s theorem) and again solved an
essentially one-dimensional, or limited two-dimensional problem (Eriguchi & Müller 1991; Clement 1994). Recently, some progress for two- and three-dimensional stellar models has been made using a multidomain approach and treating the stellar surface as a discontinuity (Bonazzola, Gourgoulhon, & Marck 1998).

Second, even if a general geometrical method applicable to large numbers of two- and three-dimensional problems could be developed, the current stellar structure methods for solving the immense system of simultaneous nonlinear equations would take a prohibitively large amount of CPU time and memory. For example, a relatively modest equations would take a prohibitively large amount of CPU for solving the immense system of simultaneous nonlinear could be developed, the current stellar structure methods (Eriguchi & 1991; Clement 1994). Recently, Müller essentially one-dimensional, or limited two-dimensional problem (Eriguchi & 1991; Clement 1994). Recently, Müller essentially one-dimensional, or limited two-dimensional problem (Eriguchi & 1991; Clement 1994). Recently, Müller essentially one-dimensional, or limited two-dimensional problem (Eriguchi & 1991; Clement 1994). Recently, Müller essentially one-dimensional, or limited two-dimensional problem (Eriguchi & 1991; Clement 1994). Recently, Müller essentially one-dimensional, or limited two-dimensional problem (Eriguchi & 1991; Clement 1994). Recently, Müller essentially one-dimensional, or limited two-dimensional problem (Eriguchi & 1991; Clement 1994). Recently, Müller essentially one-dimensional, or limited two-dimensional problem (Eriguchi & 1991; Clement 1994). Recently, Müller essentially one-dimensional, or limited two-dimensional problem (Eriguchi & 1991; Clement 1994). Recently, Müller essentially one-dimensional, or limited two-dimensional proble

Fortunately, there exist techniques for solving both of these problems that are well developed and have been in use in the engineering field for many years (although it is still rare to see both used at the same time). The finite element method (FEM), introduced more than four decades ago and the preferred method for treating multidimensional structural engineering problems since the late 1960s (Zienkiewicz 1977), approximates objects as distorted lattices of small structural members called elements. For solving the large systems of coupled equations generated by such grid problems, the multigrid method was introduced in the 1970s (Brandt 1977) and is now beginning to be used in astrophysics as well (Truelove et al. 1998; Norman 1998). This approach dispenses with the large matrix, cleverly reaching a solution after a few sweeps of the mesh. Together, these techniques promise to make the solution of multidimensional astrophysical structural problems possible, and to bring the time required to solve them to within an order of magnitude or so of that for explicit problems. When coupled with the expected increased in the speed of computers over the next few decades (which has averaged about a factor of 2 every 2 yr for the past 20 yr), it is not inconceivable that three-dimensional structure problems will soon be solved routinely on future models of PCs and that four-dimensional problems will become commonplace on supercomputers.

This first paper deals with the development of a general geometrical method for solving multidimensional structure and evolution problems. At this stage, the speed and efficiency of execution of the method will not be a concern; the focus will be on producing a robust and widely applicable set of useful techniques. Our goal will be to determine the essential features of most astrophysical systems of equations and the geometrical demands they place on the numerical method. These properties will then be encoded at the outset, ensuring some measure of generality. Section 2 describes the set of equations that can be addressed with nonlinear astrophysical finite element analysis and develops a method for solution on fixed (nonmoving) grids. In § 3, this solution is then generalized to include situations in which the positions of the grid points are part of the solution and the grid can change with time. Finally, tests and examples are given, using the author’s code, including rotating polytropic star models.

2. THE BASIC FOUR-DIMENSIONAL NONLINEAR FINITE ELEMENT METHOD ON FIXED GRIDS

This section describes the techniques used in the author’s computer code, called GENRAL, for solving general astrophysical problems. It utilizes the techniques of finite element analysis (FEA), which has been in use in the field of engineering for some time, but generalizes them to nonlinear equations in four dimensions, instead of linear equations in two or three dimensions. For this initial development, it is assumed that the coordinates of the grid points (or "nodes") at which the variables are evaluated do not change while the solution is being computed.

2.1. General Form for Equations of Continuum Astrophysics

Appendix A shows that the differential equations of continuum astrophysics in curved spacetime can be cast into the generic form

\[ \mathbf{R}_q = \left[ T^\alpha_\beta(\nabla \mathbf{w}, \mathbf{w}, x) \sqrt{-g} \right]_{,\beta} - F_q(\nabla \mathbf{w}, \mathbf{w}, x) \sqrt{-g} = 0, \]  

where \( \mathbf{w}^r = \mathbf{w}^r \) is a generalized solution vector holding all of the \( r = 1, \ldots, V \) unknown state variables, \( \mathbf{R}_q \) is a generalized residual for each of the \( q = 1, \ldots, V \) equations (which will be forced to zero through numerical relaxation techniques), \( T^\alpha_\beta \) and \( F_q \) are, respectively, the generalized stress tensor and the force vector for these equations, and \( g \) is the determinant of the metric tensor \( g \). \( \left(-+++\right) \) signature.\(^2\) In a similar manner, the boundary conditions on these equations can be cast as

\[ \mathbf{R}_r = \left[ t^\alpha_\nu(\nabla \mathbf{w}, \mathbf{w}, x) \sqrt{-g} \right]_{,\nu} + \left[ h_{g,\nu}(\nabla \mathbf{w}, \mathbf{w}, x) \sqrt{-g} \right]_{,\nu} \mathbf{n}^\nu - f_q(\nabla \mathbf{w}, \mathbf{w}, x) \sqrt{-g} = 0 \]  

for each of the \( r = 1, \ldots, R \) boundary conditions. In general, \( R \neq V \), since, depending on the highest order derivative in \( \mathbf{R}_r \), there may be 0, 1, or 2 associated boundary conditions. The last term in equation (2) is adequate for handling Dirichlet, Neumann, and mixed boundary condi

\(^2\) Throughout this paper, the notation of Misner, Thorne, & Wheeler (1973) is used, with Greek letters indicating coordinate indices in four-dimensional spacetime (\( \alpha = 0, 1, 2, 3 \)) and Latin “integer” letters indicating three-space indices only (\( i = 1, 2, 3 \)). The comma denotes ordinary differentiation with respect to the coordinates, while a semicolon will denote covariant differentiation. Repeated indices indicate summation over the entire range of those indices (the Einstein summation convention), so that

\[ \mathbf{g}^{\alpha\beta} \mathbf{g}_{\alpha\beta} = \sum_{\alpha=0}^{3} \mathbf{g}^{\alpha\beta} \frac{\partial \mathbf{g}_{\alpha\beta}}{\partial x^\gamma}. \]

A raised index indicates contravariant properties of the tensor, and a lowered index indicates covariant properties. Note that \( \mathbf{w}^\alpha \) is written as a contravariant vector with a raised index, like the coordinates \( x^\alpha \); this is partly for convenience (to facilitate the summation convention) and partly to draw attention to the variables as generalized coordinates of the system.
tions, such as the radiative condition at a stellar or accretion disk surface. The first two terms are necessary for including constraints on field equations. \( F \) is the projection tensor along the boundary \( \partial \Omega \) and orthogonal to the boundary normal \( n \),

\[
F = n \otimes n + g ,
\]

(3)

where \( n \cdot n = -1, n \cdot F = 0 \), and \( F \cdot F = F \). “ \( \otimes \) ” is the outer (dyadic) product, and a bold dot is the inner (scalar) product. \( F \) causes the divergence in equation (2) to be performed on the boundary only, and the derivative normal to the boundary to be only first order in \( n^a \partial / \partial x^a \). An alternative form for equation (2) is

\[
\mathcal{R}_e = \left[ e^{ab} (\nabla w, w, x) \sqrt{-g} \right]_{,b} \frac{\partial n^b}{\partial x^a}
\]

\[
+ \left[ h^{ab} (S \cdot \nabla w, w, x) \sqrt{-g} \right]_{,a} - f^a_{,b} (\nabla w, w, x) \sqrt{-g} - g \]

\[
= 0 ,
\]

which also has no second derivatives in \( n^a \partial / \partial x^a \).

It is important to note that equation (1) includes not only structural and steady problems, but evolving ones as well. For these cases, in addition to having three spatial coordinates, the computational grid can extend into the fourth (time) dimension, possibly from the initial time step or hypersurface to the final one. While certainly increasing the computational and memory load on the computer, this approach will have distinct advantages over conventional approaches to initial-value problems.

The boundary conditions are not necessarily completely described by equation (2). While it is well known that, physically, the boundary of a boundary is zero (\( \partial \partial \Omega = 0 \)), computationally one often introduces subboundaries by truncating the mesh or imposing symmetries on the problem. These conditions produce right-angle kinks in the boundary, where \( n \) suddenly rotates by 90° and the boundary conditions abruptly change. Such boundary corners occur, e.g., where the \( t = 0 \) initial hypersurface intersects the world line of a stellar surface, or (in the case of axisymmetry or plane symmetry) where the stellar surface intersects the symmetry axis or plane. For example, when solving Maxwell’s evolutionary equations on the four-dimensional domain \( \Omega \) under such conditions, they will be bounded on the \( t = 0 \) portion of \( \partial \Omega \) by the initial value (solenoidal and Coulomb) constraints; these will be bounded further at an external stellar two-surface \( \partial \), and these may be bounded still further by the rotation axis or equatorial plane at edges \( \partial \partial \). Therefore, additional equations, similar to equation (2), with successive projection of the first two terms into the subboundaries of lower dimension, may be needed until one reaches the zero-dimensional \( \partial \partial \partial \partial \) (endpoints of line segments), where the conditions become simply

\[
\mathcal{R}_e = - f^a_{,b} (\nabla w, w, x) = 0 .
\]

(4)

In the examples in this paper, all boundary conditions have the simple form of equation (4), but in general astrophysical situations, ones like equation (2) will be needed.

2.2. Continuous Solution: The Element Mesh

Formally, in the finite element method (FEM), the computational domain \( \Omega \) is subdivided not into nodes, but into subdomains (\( \partial \Omega \)), called “elements”—similar to “zones” or “cells” in the finite-difference method (FDM). (Nodes come later, and then only to facilitate the element process.) The elements are constructed in such a way that each function \( w^a \) is continuous over the entire domain, but its derivatives are only piecewise continuous; i.e., \( w^a \) is continuously differentiable only within each element. The \( x^a \) are treated in the same manner; they also are continuous across element boundaries, with no spatial “gaps” between elements.

Although a variety of generic element shapes can be used, the most common are triangular and quadrangular. Of course, these assume higher order shapes in three and four dimensions (\( \text{e.g.} \), equilateral triangles, tetrahedra, simplices [hypertetrahedra]; squares, cubes, and hypercubes), but they still will be referred to here as the triangular and quadrangular classes. In GENRAL, elements of the quadrangular type are used exclusively because of their convenience. The computational domain is filled with a topologically rectangular set of

\[
\delta = \prod_{x=0}^{D-1} \mathcal{N}_x ,
\]

(5)

of these building blocks, where \( D(\leq 4) \) is the dimensionality of the problem, and \( \mathcal{N}_x \) is the number of elements along each mesh dimension \( x \). Each element that borders the domain has one surface lying on the boundary that itself is an element of dimension \( D - 1 \). The total number of such boundary elements enclosing this rectangular mesh is a sum over the rectangular faces,

\[
\mathcal{B} = 2 \sum_{x=0}^{D-1} \prod_{y \neq x} \mathcal{N}_y .
\]

(6)

The element mesh can be distorted by stretching, compressing, bending, or even twisting it to conform to the geometry of the domain (as, for example, in a curvilinear coordinate system). In the engineering FEM, this coordinate transformation is called the “isoparametric” transformation, because coordinate values \( x^a \) and the variables \( w^a \) are specified at the same nodal points. In general relativity, this transformation is the generalized Lorentz transform

\[
\mathcal{L}_x = \frac{\partial \mathcal{X}^x}{\partial \xi^x} ,
\]

(7)

with inverse

\[
\mathcal{L}_x^{-1} = \frac{\partial \xi^x}{\partial \mathcal{X}^x} ,
\]

(8)

where \( \xi^x \) is the coordinate in mesh space, with a range of \( 0 \leq \xi^x \leq 1 \) in each dimension \( x \). Basis vectors along the mesh coordinate direction \( x \), and corresponding one-forms, are

\[
[e_x]^a = \mathcal{L}_x^a , \quad [\omega^a]_x = \mathcal{L}_x^a .
\]

Appendix B discusses conditions that may need to be satisfied by this transformation. However, unless one wishes to use the mesh as an actual Lorentz frame of reference, or wants to follow the evolution of all wave phenomena, only the Jacobi condition is necessary for numerical stability:

\[
\mathcal{L} \equiv \det \| \mathcal{L} \| \neq 0 .
\]

(9)

2.3. The Choice of a Basic Coordinate System

In the past, when developing a finite-difference numerical simulation code, for example, it has been customary (and
considered necessary) to write the differential equations in the same coordinate system described by the computational mesh. That is, if the mesh is spherical-polar, then the equations are written in spherical-polar coordinates, and so on. However, in the numerical method developed in this paper, this degeneracy is neither necessary nor desirable, since the mesh coordinate system is unknown until the problem is solved.

To allow for an arbitrary, unknown mesh, the differential equations will be written in a “basic” or “real-space” ($\mathbf{x}^a$) system that does not change as the calculation proceeds. The derivatives still will be computed in the mesh ($\zeta^a$) system, but in order to use them in the differential equations, they will then be transformed to the basic system,

$$ \[ \nabla w \]_a^e \equiv w_a^e = \mathcal{L}_a^{\mathbf{x}} w_a^e , \quad \text{(10)} $$

using the isoparametric/Lorentz transformation.

The choice of coordinate system for the mesh is determined by how one lays out the elements in real space. That is,

$$ g_a^b = \mathcal{L}_a^{\mathbf{x}} \mathcal{L}_b^{\mathbf{x}} g_{ab} \quad \text{(11)} $$
gives the metric coefficients in mesh space. One still needs to choose a system in which to write the differential equations; however, since the computer will be doing all the curvilinear work for us, one can select a very simple basic system, keeping the coordinates as Cartesian (or as Minkowskian) as possible. For example, in axisymmetric problems, cylindrical coordinates will be used, not spherical-polar. For three-dimensional problems, Cartesian coordinates will be used, no matter how spherical the star or flattened the accretion disk. Any curvilinear properties of the metric orthogonal to the computational domain will be embodied in the volume element $(-g)^{1/2}$, and curvilinear behavior within the domain will be handled by the isoparametric transformation.

### 2.4. Discrete Solution: The Nodal Mesh and Interpolation Scheme

As with all continuum numerical methods, the solution is expressed as a finite set of discrete values. In the FDM, these are values of the solution at specified points (nodes) in space; in spectral methods, these are coefficients of basis or interpolation functions. In the FEM, these discrete values are both nodal values and basis function coefficients. That is, the FEM has properties of both finite-difference and spectral methods.

The finite element nodes are distributed within each element in such a way that the $w^a$ can be interpolated across the element in each dimension with at least linear accuracy or better. For quadrangular elements, the simplest approach is to fill each element box with a (possibly hyper-) cubical mesh of $(\Omega_x^a + 1)$ nodes per dimension $a'$, where $\Omega_x^a$ is the order of interpolation in that dimension, and nodes are shared by adjacent elements at all the interfaces (corners, edges, faces, and hyperfaces). For a problem of total number of dimensions $D$, the total number of nodes describing each element is, then,

$$ I = \prod_{a=0}^{D-1} (\Omega_x^a + 1) . \quad \text{(12)} $$

That is, for four-dimensional elements, $I = 16$ for first-order (linear) interpolation, $I = 81$ for second-order (quadratic) interpolation, and $I = 256$ for third-order (cubic) interpolation—just within each element. The total number of nodes in the entire mesh is then

$$ \mathcal{F} = \prod_{a=0}^{D-1} (\Omega_x^a + 1) ; \quad \text{(13)} $$

(no sum on $\mathbf{x}'$). The number of nodes on each element’s boundary is the total minus those in the interior,

$$ K = I - \prod_{a=0}^{D-1} (\Omega_x^a - 1) , \quad \text{(14)} $$

and for the entire mesh,

$$ \mathcal{N} = \mathcal{F} - \prod_{a=0}^{D-1} (\Omega_x^a - 1) . \quad \text{(15)} $$

No matter what the value of $\Omega_x^a$, in these simple cases the basis functions in mesh space for a node $i$ within a given element $e$ are products of Lagrange interpolation polynomials $\mathcal{L}_e^{\mathbf{x}}$ in each dimension $a$:

$$ \mathcal{N}_i^e(\xi) = \Omega_{ie}(\xi) $$

$$ \equiv \prod_{a=0}^{D-1} \mathcal{L}_e^{\mathbf{x}}(\xi^a) $$

$$ = \prod_{a=0}^{D-1} I \left( \xi^a - \xi_{ie}^a \right) , $$

$$ \text{(16)} $$

where $\xi_{ie}^a$ is the mesh coordinate value at node $i$ in element $e$, and $\mathcal{N}_i^e$ is the contribution from that element to the basis or “shape” function for node $i$, as measured in the mesh (primed) system. The range of the nodal indices in the entire mesh is $i, j = 1, \ldots, I$, but the product in equation (16) only runs over the nodes within the element. The total basis function for node $i$ is then a sum over the element contributions,

$$ \mathcal{N}_i(\xi) = \sum_e \mathcal{N}_i^e(\xi) ; \quad \text{(17)} $$

(which actually involves only the elements containing that node). Note that each shape function attains unit value at its own node and zero at all other nodes in its associated elements (and in the mesh as well),

$$ \mathcal{N}_i(\xi_e) = \delta_{ie} \quad \text{(18)} $$

Often, the body-centered nodes, and sometimes even the face-centered nodes, are removed from the standard Lagrangian elements to form the so-called “serendipitous” elements (Zienkiewicz 1977). In that case, if node $\tilde{i}$ is removed, then the basis functions are given by the normal Lagrange shape function with the Lagrange shape function for the missing node subtracted from

$$ \mathcal{N}_i(\xi_e) = \Omega_i(\xi) - \mathcal{L}_{\tilde{i}}(\xi) , \quad \text{(19)} $$

with $\tilde{i} \neq i$.

Most of the properties of Lagrangian elements can be illustrated in one dimension. Figure 1 shows a simple one-dimensional five node mesh and its discretization in linear and quadratic elements. Note that interior shape functions have continuous derivatives at their respective nodes, while shape functions on element boundaries have discontinuous derivatives. (The latter also involve more nodes, since they are composed of shape function pieces from adjacent elements.) The FEM, therefore, can be considered to be a multidomain spectral method, with each of the thousands to millions of elements being a separate domain.
2.5. Formation of Derivatives and the Differential Equations

We now have a numerical procedure for computing the derivatives of the \( w^\ast \) with respect to \( x^\ast \). First the coordinate transformation matrix is formed,

\[
\mathcal{L}^\ast_a = N^\ast_{1,a} x_{1}^\ast ,
\]

and then it is inverted to obtain \( \mathcal{L}^{\ast,a} \). Then the derivatives of the variables are computed in mesh space,

\[
w^{\ast}_a = N_{1,a} w_1^\ast ,
\]

and, finally, transformed to real space by the chain rule

\[
w^a_1 = \mathcal{L}^{\ast,a} N^\ast_{1,a} w^\ast_1 .
\]

Although not usually used in practice in the actual computer code, it is sometimes useful for analytic purposes to express the shape functions in real-space coordinates and then use them to interpolate the \( w^\ast \) and compute their derivatives,

\[
w^a(x) = N^\ast_{i}(x) w^\ast_i ,
\]

\[
w^a_1(x) = N^\ast_{1,a}(x) w^\ast_1 .
\]

The real-space \( N^\ast_i \) also have the normalized property at their respective nodes that

\[
N^\ast_i(x_i) = \delta_i .
\]

By comparing equations (26) and (27) with equations (20) and (25), one concludes that the real-space basis functions and their derivatives are

\[
N^\ast_{i}(x) = N^\ast_{i}[\xi(x)] ,
\]

\[
N^\ast_{1,a}(x) = \mathcal{L}^{\ast,a} N^\ast_{1,a}[\xi(x)] .
\]

With expressions for the \( w^\ast \) and \( w^\ast_1 \) (in either mesh or real space), we can now calculate the residuals (eq. [4]) at any point in the domain \( \Omega \), and \( \mathbb{R}_q \) (eq. [4]) at any point on the boundary \( \partial \Omega \), not just at the nodes.

2.6. Generation of the Nodal (“Difference”) Equations: The Weighted Residual Method

2.6.1. General Formulation

The next step in the development of the astrophysical FEM is to construct a set of \( V,F \) equations for the \( V,F \) shape function coefficients \( (w^\ast_i) \) that fully describe \( w^\ast(x) \). This is accomplished by integrating the physical differential equation (1) and/or the boundary conditions given in equation (4) over a function \( w^\ast_i(x) \) that peaks near (but not necessarily at) node \( i \) and falls to zero far from that node. This produces \( V,F \) discrete nodal equations

\[
\mathcal{J}_{q}(w^\ast_i) = \int_{\Omega} w^\ast_i(x) \mathbb{R}_{q}(w^\ast_i) d\Omega = 0 .
\]

Relaxation schemes in the code then attempt to force the nonlinear \( \mathcal{J}_{q} \) to zero. In principle, each \( \mathcal{J}_{q} \) is a function of all the \( w^\ast_i \). However, in practice, because \( w^\ast_i \) is peaked near node \( i \), \( \mathcal{J}_{q} \) involves only nodes local to \( i \)—in fact, only nodes in those elements containing node \( i \). The \( \mathcal{J}_{q} \), therefore, are more similar to difference equations than to spectral equations, and when linearized produce a banded rather than a filled matrix.

Because \( \mathbb{R}_q \) can contain second derivatives, the integral in equation (30) cannot be performed uniquely for every node.
using only the interpolation within a single element. The standard solution to this problem, and the key step in the finite element process, is to integrate the weighted residual by parts to arrive at the so-called “weak” form,

\[
\mathcal{S}_q(i, \mathbf{w}) = \int_{\Omega} \mathbf{w}^q(x) \mathbf{T}^q_\beta \, d(\partial \Omega)_\beta \left[ \mathbf{w}^q(x), \mathbf{F}_q \right] \, d\Omega = 0 ,
\]

expected in finite-difference schemes (although in general geometry), but scalars and source terms are weighted averages of nodes surrounding \(i\) rather than evaluated exclusively at \(i\).

An important property of the shape functions hints at a more fundamental interpretation of the weighted residual method that is not usually discussed in the engineering literature. As the element volume \(\partial \Omega\) approaches zero, the shape functions become good approximations to the Dirac delta function

\[
limit_{\partial \Omega \to 0} \mathcal{N}_q(i, x) = k_i \delta(\mathbf{x} - \mathbf{x}_i)
\]

(no sum on \(i\), where \(k_i\) is a scaling constant of order unity, but generally different for each node \(i\). Therefore, with the Galerkin method, the weights \(\mathbf{w}^q\) in equation (30) are generalized approximations to \(\delta(x - x_i)\), which, when integrated over a differential equation, generate or “pick out” the corresponding difference equation near node \(i\). As the number of finite elements approaches infinity, the \(\mathcal{S}_q\) more closely approximate the complete set of \(\mathbf{q}_q\) defined at all points in \(\Omega\). Therefore, while originally derived for linear equations, the weighted-residual method is valid for nonlinear problems as well.

2.6.3. First-Order Equations: Petrov-Galerkin Schemes and “Staggered Grids”

The lack of a single, universally applicable weighting function \(\mathbf{w}^q\) is the main impediment to developing a truly general simulation code. One must always know the order of differential equation being integrated. For example, while the Galerkin scheme works well for second-order equations, it has the same pitfalls for first-order equations and fluid flow as centered-differencing schemes have in the FDM: leapfrogging, in which important terms in the equations do not depend on variables at the node at which the integral in equation (31) is evaluated \([\vec{\delta} \mathbf{w}/\partial \mathbf{x}_i \approx (w_{i+1} - w_{i-1})/2 \Delta \mathbf{x}]\), and two-point oscillations near shocks.

Such problems can be addressed by using weighting functions other than \(\mathcal{N}_q\). These are called Petrov-Galerkin schemes (Hughes 1987). For odd-order equations, functions that shift the peak of the weight away from node \(i\) reduce or eliminate many of these problems. This is the case for weights that are shape functions of twice the element interpolation order,

\[
\mathbf{w}^q_{1PG1}(x) = \mathcal{N}_{1PG1}(x) ,
\]

and for those that are distorted by the shape function derivative,

\[
\mathbf{w}^q_{1PG2}(x) = \mathcal{N}_{1PG2}(x) .
\]

In the first case, \(i + \frac{1}{2}\) signifies a position in the mesh centered between nodes. The functions \(\mathbf{w}^q_{1PG1}\) peak in between nodes and have almost the same effect as using a staggered grid in the FDM. Both \(\mathbf{w}^q_{1PG1}\) and \(\mathbf{w}^q_{1PG2}\) generate non-leapfrogging differences \([\vec{\delta} \mathbf{w}/\partial \mathbf{x}_i \approx (w_{i+1} - w_{i-1})/\Delta \mathbf{x}]\), and are still generally second-order accurate (or higher), since the scalar source terms are evaluated at the same place as the derivatives.

An upwinding scheme can be generated by setting \(\vec{u}^+ \propto \vec{u}^\pm\) and integrating only the advective terms with this weight. However, this scheme has low-order accuracy and is rather diffusive (Hughes 1987). Better methods for handling shocks
are the van Leer scheme (van Leer 1979), in which one enforces monotonicity in the gradient of the flux of a conserved quantity, and higher order Godonov schemes (Colella & Woodward 1984; Colella 1990), in which one applies the shock jump conditions within the element itself. Upwinding schemes in the FEM are a subfield in themselves, and are largely beyond the scope of this paper.

2.7. Application of Boundary Conditions

Not all the nodal equations generated by equation (31) are useful. For example, for each second-order differential equation (one in which a given \( \mathcal{F}_q \) is a function of the gradient of at least one \( w \)), exactly \( \mathcal{X} \) of the nodal equations (those integrated over shape functions peaking at boundary nodes on \( \partial \Omega \)) are meaningless or incomplete. This is a consequence of the absence of elements beyond the boundary needed to complete the integrals. For first-order equations (ones where \( \partial \mathcal{F}_q \) is, at most, a function of \( w \) and \( x \) only), those on only one portion of the boundary (e.g., on one side or surface) must be discarded. These ignored nodal equations must be replaced by exactly the same number of boundary or initial conditions. These are generated in a weighted-residual manner similar to that used in equation (30), except that the integral and weighting functions now are evaluated on the boundary

\[
\mathfrak{B}_d(w_j) = -\int_{\partial \Omega} \mathcal{W}_k(x_j) f_j(w_j) d(\partial \Omega), \tag{38}
\]

where \( d(\partial \Omega) \) is the magnitude of the surface element

\[
d(\partial \Omega) = n^b d(\partial \Omega)_b, \tag{39}
\]

and the nodes \( k \) lie on \( \partial \Omega \). The number of equations, therefore, will remain equal to the number of unknowns, as is necessary for a well-posed problem.

2.8. Numerical Aspects of the Method

2.8.1. Integration of the Weighted Residuals

In the FEM, the integrals in equations (31) and (38) often are performed numerically using Gaussian integration (Abramowitz & Segun 1965), with all sampled points \( x_j \) interior to element boundaries. In practice, the integrals are calculated piecwise, element by element, with each element’s contribution to the various integrals summed accordingly. To accomplish this, one defines a local mesh coordinate system, referenced to the element center and parallel to the global \( \xi^e \) system:

\[
s^e = \delta^e \frac{2(\xi^e - \xi^e_0)}{\Delta \xi^e}, \tag{40}
\]

\[
\frac{\partial}{\partial s^e} = \delta^e \frac{\Delta \xi^e}{2} \frac{\partial}{\partial \xi^e}, \tag{41}
\]

\[-1 \leq s^e \leq 1, \tag{42}
\]

where \( \xi^e_0 \) is the position in mesh space of element \( e \)'s center, \( \Delta \xi^e \) is the element width in direction \( \xi^e \), and double primes refer to the local element system. The Lagrangian shape functions within each element take on a very simple form in this local system. For linear interpolation in one dimension (with nodes lying at \( s_{1e} = -1 \) and \( s_{2e} = +1 \)),

\[
\phi''_e(s_e) = 0.5(1 + s_e s_e), \tag{43}
\]

For quadratic interpolation (with element nodes lying at \( s_{1e} = -1, 0, +1 \)),

\[
\phi''_e(s_e) = \begin{cases} 0.5 s_e (1 + s_e s_e) & \text{for } s_e = \pm 1, \\ 1 - s_e^2 & \text{for } s_e = 0, \end{cases} \tag{44}
\]

and so on for higher order interpolation. (Eqs. [43] and [44] are the functions depicted in Fig. 1.) Higher dimensional element shape functions are products of these in a manner similar to equations (16) for Lagrangian elements and equation (19) for serendipitous elements.

Dropping the first term, as discussed earlier, and summing over elements and Gaussian integration points, equation (31) then becomes

\[
\lambda_{q1} \approx -\sum_{e} \omega_e \mathcal{W}_1(x_{1e}) T^{b}_{q} + \mathcal{W}_1(x_{2e}) F_{q} \delta(\partial \Omega)_e, \tag{45}
\]

where

\[
\delta(\partial \Omega)_e = \mathcal{L} \sqrt{-g} \prod_{a=0}^{D-1} \Delta \xi^e_a \tag{46}
\]

is the element volume in the real-space coordinate system, \( e = 1, \ldots, \mathcal{E} \) is the element number in the mesh, \( g = 1, \ldots, \mathcal{G} \) is the number of the Gaussian integration point within element \( e \), and \( \omega_e \) is the Gaussian weight at that point. Similarly, equation (38) becomes

\[
\lambda_{kb} \approx -\sum_{e} \omega_e \mathcal{W}_k(x_{1e}) f_e n^e \delta(\partial \Omega)_e, \tag{47}
\]

where \( b = 1, \ldots, \mathcal{B} \) is the boundary element number,

\[
\delta(\partial \Omega)_e = \mathcal{L} \sqrt{-g} \prod_{\mathcal{F} = 0}^{D-1} \Delta \xi^e_\mathcal{F} \tag{48}
\]

is the surface one-form on the \( \mathcal{A} \) boundaries, and \( n^e \) is one of \( \delta^e_\mathcal{F}/(-g_{00})^{1/2}, \delta^e_\mathcal{F}/(-g_{11})^{1/2}, \) etc., depending on the boundary.

Because the residual weights \( \mathcal{W}_1(x) = \mathcal{W}_1''(s) \), their derivatives, and the Gaussian weights \( \omega_e \) are the same for all elements, they can be precomputed and stored prior to beginning the relaxation of the solution. The only quantities we need to compute during the relaxation are \( T^{b}_{q} \) and \( F_{q} \) at each \( x_{1e}, x_{2e} \) interior integration point and the \( f_e \) at each \( x_{1e}, x_{2e} \) boundary integration point.

Historically, the number of integration points \( G \) used in each element is a function of the expected nonlinearity of the product \( \mathcal{W} \mathcal{R} \) with respect to position \( x \). In engineering, this is usually of low order (linear or quadratic), but in astrophysics this product can vary with exponential order or higher. Nevertheless, in practice, even with highly nonlinear functions the author has had quite satisfactory results using the same number of integration points as nodes in each dimension (i.e., \( G = \mathcal{I} \)). Fewer than this (“underintegration”) reduces the order of accuracy or can even produce a singular matrix. More than this (“overintegration”) does little to improve accuracy (of order unity improvements only), at great computational expense.

2.8.2. Solution of the Simultaneous Nonlinear Equations: The Multidimensional Henyey Method

For solving the \( V \mathcal{F} \) nodal equations and boundary conditions, the author currently uses a standard multivariate Newton-Raphson technique, sometimes called the Henyey method in astrophysics (Clayton 1968; Aller & McLaughlin
1965). The $\mathcal{S}_{q_i}$ are linearized by expanding in a Taylor series about the solution $w_{j}^{*}$, resulting in a matrix inversion problem,

$$\left(\frac{\partial \mathcal{S}_{q_i}}{\partial w_{j}^{*}}\right)^{[n]} \delta w_{j}^{[n]} = - \mathcal{S}_{q_i}^{[n]} \quad (49)$$

(no sum on the iteration number $[n]$), to solve for the corrections $\delta w_{j}^{[n]}$ that need to be applied to the current guess to obtain a new $[n + 1]$st approximation to the solution,

$$w_{j}^{[n+1]} \approx w_{j}^{[n]} + \delta w_{j}^{[n]} . \quad (50)$$

In the engineering FEM, $\partial \mathcal{S}_{q_i} / \partial w_{j}^{*}$ is called the “tangent stiffness” matrix. It has a length $V, \mathcal{S}$ on a side and a bandwidth $\sim V, \mathcal{S}^{1-1}/b$. At present, GENRAL uses direct methods (Gaussian elimination with lower-upper decomposition) to solve equation (49), repeatedly applying the corrections until the norm over all $v$ and $j$,

$$\Delta_{w} \equiv \| \delta w_{j}^{[n]} / w_{j}^{[n+1]} \| ,$$

falls below a certain tolerance.

The tangent stiffness matrix need not be extremely accurate. Indeed, when $\Delta_{w} \ll 1$, the matrix need not be recomputed at all, with little impact on the rate of convergence and no impact on the accuracy of the solution. Furthermore, the elements of the stiffness matrix in equation (49) can be calculated using numerical differentiation, rather than writing out explicitly the partial derivatives of each equation with respect to each variable. This eliminates the need to know the geometry of the mesh beforehand—a feature important for multidimensional astrophysical structures. Numerical differentiation of $\partial \mathcal{S}_{q_i} / \partial w_{j}^{*}$ is not necessarily more expensive than algebraic differentiation, especially if the residual result integrals $\mathcal{S}_{q_i}^{[n]}$ are calculated only once for each matrix, and those partial derivatives known to be identically zero are not computed.

### 2.8.3. Logarithmic Variables

It is quite common for an astrophysical state variable (e.g., the density, $\rho$) to vary by many orders of magnitude over $\Omega$. Therefore, in order to maintain the same relative accuracy over the domain, it may be necessary to solve for a much more slowly varying function, e.g., $\bar{v} \equiv \log_{10} \rho$. In addition, for variables that can be positive or negative (such as velocity), one may need a more complex function and its inverse, e.g.,

$$\bar{v} = \log_{10} v \equiv S \log_{10} (1 + S v/v_{\text{scale}}) , \quad (51)$$

$$\text{sdx} \bar{v} = \text{sdx} v_{\text{scale}} (10^{\text{sdx}} - 1) , \quad (52)$$

where $S \equiv \text{sgn } v = \text{sgn } \bar{v}$, and $v_{\text{scale}}$ is a fixed scaling value for $v$. These “scaled logarithmic variables” are linear for $|v| \ll v_{\text{scale}}$ and logarithmic for $|v| \gg v_{\text{scale}}$, and can be negative or positive.

Unfortunately, there is some loss of convenience and intuition in rewriting the equations in terms of these new, modified variables, especially if there are many of this nature. Therefore, the following procedure has been devised in order that the differential equations can still be coded in their basic form (i.e., using $\rho$ and $v$), while maintaining the accuracy of solving for $\log_{10} \rho$ and $\log_{10} v$:

1. Each variable is flagged as being linear, logarithmic, or scaled logarithmic, and then stored as

$$\bar{w}^{e} = \ell (w^{e}) \equiv \begin{cases} w^{e} & \text{if linear,} \\ \log_{10} w^{e} & \text{if logarithmic,} \\ \text{slog}_{10} w^{e} & \text{if scaled logarithmic.} \end{cases}$$

2. Partial derivatives of the integrals in equation (49) are calculated with respect to $\bar{w}$. However, when computing the functions $T_{q_i}, F_{q_i}$, and $f_{j}$, the stored variables and their gradients are reexponentiated to their unmodified forms, with

$$w^{e} = \ell (\bar{w}) \equiv \begin{cases} w^{e} & \text{if linear,} \\ 10^{w^{e}} & \text{if logarithmic,} \\ \text{sdx} w^{e} & \text{if scaled logarithmic,} \end{cases}$$

and

$$w_{j}^{e} = \frac{\bar{w}_{j}^{e}}{\partial \ell (w^{e}) / \partial w^{e}} .$$

With this scheme, one can choose a variable to be logarithmic or not at run time, or even switch its character during execution, without modifying the code.

#### 2.8.4. Pivoting

Lower-upper decomposition techniques work well only when the matrix elements on the diagonal are decidedly nonzero. That is, one must identify which equation $\mathcal{S}_{q_i}$ is “for” which variable $w_{j}^{*}$. The term pivoting refers to the exchange of rows and/or columns in the matrix to ensure that all elements on the diagonal are indeed large—i.e., that the trace of the stiffness matrix is a maximum.

**Local**, or partial, pivoting ensures that, at a given node, the correct physical equation is paired with the correct variable. For example, in MHD computations, when the electrical conductivity is infinite, the current $\mathcal{S}$ is determined by Maxwell’s equations, not Ohm’s law. Or, in a hydrostatic star, the momentum equation determines the pressure structure, while velocity is determined by energy or particle conservation. In the standard partial pivoting algorithm, one searches a matrix column for the largest element and switches the row of that element with the one presently on that column’s diagonal (Press et al. 1989). When applied at a single node, this algorithm is very successful in automatically pairing equations and variables. That is, the solution will be able to evolve from a dynamic state to a hydrostatic one without recasting the equations or writing a new simulation code.

**Global** pivoting ensures that an equation $\mathcal{S}_{q_i}$ (integrated near node $i$) is applied at the correct node $j$. This is a more difficult task than local pivoting and is not easily automated in our case. The correct identification is not always $j = i$, especially for first-order equations, for which the answer is determined by where the boundary conditions are applied. Fortunately, the global pivot of the matrix is a property that usually does not evolve with the simulation; it needs to be determined only once. Therefore, an elaborate pre-pivoting scheme has been devised for GENRAL, in which shape function coefficients are identified with nodal equations, useless nodal equations are discarded, and boundary conditions are inserted based only on the equation order and the location of boundary conditions. Local pivoting can still shift emphasis for a given variable from one differ-
ential equation to another, but the overall global identification of integrals given in equation (31) with nodes remains fixed.

3. THE FINITE ELEMENT METHOD WITH NODAL COORDINATES AS PART OF THE SOLUTION

In most multidimensional astrophysics problems, the grid chosen initially will not be a good match for the final structure, because of a poor fit to the object boundary or poor resolution in areas of rapid gradients. Therefore, the coordinates and/or quantity of nodes should be changed as the solution converges in order to get a better fit (“adaptation”). In addition, one needs to allow for the mesh at one time step to be different from that at the previous time step (grid motion).

3.1. Adaptive Gridding

Adaptive gridding, as used here, means modifying the mesh spacing in order to achieve greater accuracy or stability without changing the total number of nodes or the topology of the mesh. One-dimensional stellar structure models use a form of adaptive gridding, since they utilize a mass coordinate rather than radius, allowing the radius of each mass zone, including the outer stellar radius, to expand or contract depending on the current state of the star. Our general multidimensional adaptive gridding scheme takes the same form as equation (1), except that it is written in mesh space:

\[ \mathcal{A}^{a} \hat{\beta} = 0 , \tag{53} \]

where \( \mathcal{A}^{a} \hat{\beta} \) is the adaptive gridding tensor. Currently, the author is using a diagonal form,

\[ \mathcal{A}^{a}_{\alpha} = f_{\alpha} \left( C_{1} e_{\alpha} \cdot \nabla w - \frac{\langle \Delta w^{\alpha} \rangle}{\Delta x^{\alpha}} \right) + \frac{C_{2}}{\Delta x^{\alpha}} , \tag{54} \]

which ties the mesh spacing to the local gradient of the state variables. Note the sum over \( V \) variables; \( f_{\alpha} \) is a vector of 1’s and 0’s that, at run time, selects those state variables to which the grid should be adapted, \( e_{\alpha} \) is a unit vector in spacetime along the local mesh direction \( \xi^{\alpha} \),

\[ [ e_{\alpha} ]^{a} = \frac{\mathcal{A}^{a}_{\alpha}}{g_{\mu\nu} \mathcal{A}^{\mu}_{\alpha} \mathcal{A}^{\nu}_{\alpha} |^{1/2} } \]

(no sum on \( \alpha \)), \( \langle \Delta w^{\alpha} \rangle \) is the average change in \( w^{\alpha} \) along the mesh direction \( \xi^{\alpha} \) (determined from the current solution for \( w^{\alpha} \)), \( \Delta x^{\alpha} \) is a measure of the local linear mesh spacing along an element edge,

\[ \Delta x^{\alpha} = \frac{\Delta \xi^{\alpha}}{e_{\alpha} \cdot \mathcal{A}^{\alpha}_{\alpha}} = | g_{\alpha\beta} |^{1/2} \Delta x^{\alpha} \]  \( \tag{55} \)

(again, no sum on \( \alpha \)), and \( C_{1} \approx 0.2 \) is a constant that regulates the strength of the gradient term. Equations (53) provide four constraints on the nodal coordinate values, allowing the mesh spacing along \( \xi^{\alpha} \) to decrease in regions of high gradients in the variables, but to be uniform otherwise. Note that the adaptive gridding equations are not the coordinate conditions needed to complete the description of the metric (see Appendix A). They simply move nodal positions around in an already determined metric. \( C_{2} \) is a very small constant \( (\approx 10^{-10}) \), such that in the absence of adaptive gridding (i.e., all \( f_{\alpha} = 0 \)) and with the proper boundary conditions, the mesh will assume an appropriate curvilinear character, with uniform spacing \( \Delta x^{\alpha} \) in each \( \xi^{\alpha} \).

3.2. Moving Grids and the Advection Derivative

Adapting the grid to local conditions changes the spatial and time coordinates of the nodes. For an observer traveling along \( \xi^{0} \), this produces what appears to be motion of the grid through space. Now, computational fluid dynamics on stationary, uniformly spaced meshes is a fairly complex field in itself. On arbitrarily spaced and moving grids, the resulting mixed Lagrangian-Eulerian hydrodynamic equations might seem intractable. However, with the FEM, the exact opposite is true. If the elements have an extent in time as well as in space, then the inverse isoparametric transformation will automatically take into account the complicated effects of differencing the fluid equations with respect to a moving grid. It is unnecessary, and indeed incorrect, to attempt to include the grid velocity in the differential equations.

As an example, consider the nonrelativistic total advective derivative in flat spacetime,

\[ \frac{D}{Dt} = \frac{\partial}{\partial t} + v^{i} \frac{\partial}{\partial x^{i}} \]

\[ = \frac{\partial}{\partial t} \frac{\partial}{\partial \xi^{0}} + v^{i} \frac{\partial}{\partial \xi^{0}} \frac{\partial}{\partial \xi^{i}} , \tag{56} \]

where \( v^{i} \) is the three-velocity of fluid flow. Now, let \( v^{i} \) be a grid three-velocity such that \( e_{ni} \) remains parallel to \( e_{nx} \), although \( e_{ny} \) can make an angle \( (\tan \theta = |v_{y}| |\Delta t| / |\Delta x|) \) with \( e_{x} \). (These are the conditions LeBlanc & Wilson [1970; 1971, private communication] placed on their moving grid in their Lagrangian-Eulerian MHD calculations.) The isoparametric transform and its inverse are then

\[ \frac{\partial x^{\beta}}{\partial \xi^{\alpha}} = \begin{cases} \frac{\Delta x^{\beta}}{\Delta x^{\alpha}} & \text{if } \beta = \alpha' \\ \frac{v_{g} \Delta t}{0} & \text{if } \beta = i; \alpha' = 0 \\ 0 & \text{otherwise}, \end{cases} \tag{57} \]

\[ \frac{\partial \xi^{\alpha}}{\partial x^{\beta}} = \begin{cases} \frac{\Delta \xi^{\alpha}}{\Delta x^{\beta}} & \text{if } \alpha' = \beta \\ - \frac{v_{g} \Delta \xi^{\alpha}}{\Delta x^{\beta}} & \text{if } \alpha' = 0; \beta = i' \tag{58} \\ 0 & \text{otherwise} \end{cases} \]

(no sum on \( \beta \) or \( i \)). Substituting equation (58) into equation (56), one obtains

\[ \frac{D}{Dt} = \frac{\Delta \xi^{0}}{\Delta t} \frac{\partial}{\partial \xi^{0}} + (v^{i} - v_{g}^{i}) \frac{\Delta \xi^{i}}{\Delta x^{i}} \frac{\partial}{\partial \xi^{i}} . \tag{59} \]

The first term in equation (59) is the apparent time derivative (along \( \xi^{0} \)) at a given node in the grid frame, while
the second is simply the moving advective derivative, \((v - v_n) \cdot V\). Thus, the isoparametric transform reproduces the Lagrangian-Eulerian equations of LeBlanc & Wilson under the same conditions.

3.3. Adaptive Mesh Refinement

The purpose of adaptive mesh refinement (AMR) is similar to that of adaptive gridding: to increase resolution in a local region of the mesh. However, in the case of AMR, the numbers of elements and nodes usually increase as the calculation proceeds. When the element mesh extends into the time dimension, AMR makes it possible to treat efficiently the propagation of particularly important wave phenomena. Rather than subdividing the entire domain finely, one does so only for the region of spacetime near the null geodesics along which the phenomenon propagates.

The Courant condition (Appendix B) prescribes one form of AMR, since it places an upper limit on the temporal mesh spacing, and therefore a lower limit on the number of elements necessary to solve a problem. If the Courant condition is not satisfied in a local region, the number of elements in \(\varepsilon_0\) will have to be increased.

In the FEM, AMR can be achieved by subdividing some of the elements, usually by a power of 2 in a given dimension, and then rebuilding the interpolation grid of nodes within the newly gridded subdomain. There are two matters of concern in this case: (1) the subdivision must be done by elements, not nodes, regardless of the final order of the interpolation grid used within each element, and (2) larger elements bounding the more finely subdivided region must have their internal interpolation scheme, and hence their nodal shape functions, modified by the addition of new boundary nodes. That is, these modified elements must be serendipitous elements. This ensures that the functions \(w^\prime(\xi)\) and \(x^\prime(\xi)\) are continuous across element boundaries and that there are no spatial gaps or "hanging nodes." This second requirement can significantly increase the complexity of the mesh and the number of element types for which quantities need to be precomputed and stored.

Recently, however, the astrophysical community has been embracing an alternative, multilevel approach to AMR (Truelove et al. 1998; Norman 1998). A region of space is refined not by subdividing the grid cells themselves, but by applying separate, and successively more refined, grids at the same location and with some nodes in common between each level. This hierarchical approach eliminates the hanging node problem without resorting to defining many new serendipitous element types: each grid level is subdivided with standard linear or quadratic elements. This technique also fits naturally into a multigrid iterative scheme for solving the coupled nodal equations.

4. TESTS AND EXAMPLES

Below are presented some tests of the code GENRAL on problems with known solutions. At present, the code uses the multidimensional Henyey technique to solve the difference equations it generates. As discussed earlier, this approach has severe computer time and memory limitations. Therefore, all the examples involve a much smaller number of elements than the millions that one would use in a typical astrophysical simulation. (Generally, the full code running on a desktop workstation is limited to 4096 elements or 6561 nodes total: \(64^2, 16^3,\) or \(8^4\) linear elements or \(32^2, 8^3,\) or \(4^4\) quadratic elements). Nevertheless, the tests serve to demonstrate the unique features of the FEM, including its ability to solve nonlinear astrophysics-like problems in multidimensional, arbitrary curvilinear coordinate systems and to achieve high accuracy in the solution by employing higher order interpolation, adaptive gridding, and logarithmic variables.

4.1. Simple Tests

4.1.1. Tests of the Interpolation Scheme

The first tests of the method involve no differential equations. Instead, the integrands in equations (45) and (47) are set to unity, so that only domain volume and surface area are computed. The results are then compared with known solutions for various shapes (rectangular hypersolids, hyperspheres, and ellipsoids), different element interpolation schemes (linear \([\theta = 1]\) and quadratic \([\theta = 2]\)), different numbers of elements \(N\) in each dimension \((1, 2, 4, 8,\) and 16), and different numbers of integration points \(G\) \([\theta_D]\) and \((\theta + 1)^D\)). This procedure is more than simply a verification that the numerical integrations work. It also exercises the coordinate transformations and the ability to form proper volume and surface elements in fairly arbitrary curvilinear situations.

Table 1 summarizes the results, normalizing to grids with nine nodes in each dimension.\(^4\) In this case (but, unfortunately, not for the differential equation tests below), both choices for \(G\) produce the same results: for linear elements, the error \(E_g = (\Omega_k - \Omega)\Omega\) varies as \(\Delta x^2 \propto (N_\theta - 1)^{-2}\), while quadratic elements produce fourth-order errors, \(E_g \propto \Delta x^4 \propto (N_\theta - 1)^{-4}\). Results for surface-area integration are similar, although the actual fractional error is about a factor of 2 smaller. Results for other ellipsoidal objects are also similar.

4.1.2. Fixed Cartesian Grid Tests: Poisson’s Equation in One to Four Dimensions

The second set of tests involves solving a differential equation in up to four dimensions, but on a regular Carte-

### Table 1

**Error Results for Volume Integration of D-Spheres\(^a\)**

| \(D\) | Linear\(^b\) | Quadratic\(^b\) |
|------|-------------|----------------|
| 2 ... | \(-6.4 \times 10^{-3}\) | \(-4.9 \times 10^{-3}\) |
| 3 ... | \(-2.0 \times 10^{-2}\) | \(-9.0 \times 10^{-5}\) |
| 4 ... | \(-3.7 \times 10^{-2}\) | \(-2.2 \times 10^{-4}\) |

*Note.*—Meshes with nine nodes in each dimension.

\(^a\) Surface area results have a similar scaling, but with a factor of 2 better accuracy.

\(^b\) Accuracy for linear elements varies with \(\Delta x^2\), and for quadratic elements as \(\Delta x^4\).

\(^4\) Given that the method is element based, one could compare equal numbers of linear and quadratic elements in each dimension. In that case, the errors in the right-hand column of Table 1 would be a factor of 16 smaller. However, performing the comparison in the latter manner would make it difficult to tell which part of the improvement for quadratic elements is due to the interpolation scheme alone and which is due to the increase in the number of nodes.
sian (not Minkowskian) grid of dimension $D$. The equation used is Poisson’s equation

$$w_{ss} - \rho = 0,$$

(60)

with a known solution

$$w_0 \equiv r^n,$$

(61)

where $r \equiv (x^2 + y^2 + z^2 + t^2)^{1/2}$ is the radial distance from the origin, and $0 < x^2 < 1$. The components of the generalized equation (1) are then

$$T^\theta = w_\theta,$$

(62)

$$F = \nabla^2 w_0 = n(n + D - 2)y^{(n-2)},$$

(63)

with Dirichlet conditions on the entire boundary

$$f = w - \left[r(\partial \Omega)\right]^n,$$

(64)

where $r(\partial \Omega)$ is the expression for $r$ evaluated on the boundary. This test exercises the code’s ability to solve equations (45) and (47), but does not test the coordinate transformations.

Results of the fixed-grid tests are given in Table 2, which shows how the accuracy of the solution in different dimensions, determined by the normalized $L_2$ error norm

$$E_{w^2} \equiv \frac{\int (w - w_0)^2 d\Omega}{\int w_0^2 d\Omega}^{1/2}$$

varies with number of nodes and elements used. Two differences from the volume and surface-area integration tests are worth noting. First, the solution errors for quadratic elements are third-order accurate [$E_{w^2} \propto \Delta x^3 \propto (N^{\beta} - 1)^{-3}$], not fourth-order. Second, underintegration ($G = \beta D$) does not work. $G$ must always be equal to or greater than $(N^{\beta} + 1)^D$ (i.e., $\geq 2^D$ for linear elements and $\geq 3^D$ for quadratic elements) in order to generate the proper second-order differences in the Laplacian. At best, underintegration reduces the accuracy of the solution by one order. At worst, it destroys nearest-neighbor differencing, producing leapfrogged differences, and can lead to no solution at all. “Isointegration” ($G = I$) is probably sufficient for any equation of the form of equation (1), but it should be checked in each circumstance to be certain.

4.2. Adaptive and Curvilinear Grid Tests

The third set of tests exercises nearly all the features of the code in order to obtain a solution to a rather pathological Poisson problem, a Fermi-Dirac-like function

$$w_0 = \frac{1}{e^{(x/0.5)} + 1}$$

(65)

with a cold temperature of $1/f = 0.02$. Such sudden exponential drops in the solution are common at stellar core-halo boundaries or stellar surfaces and are difficult to resolve accurately without a large number of nodes or a variable change (to optical depth, for example). For this demonstration, the conservative form for the stress and force terms (in up to four Cartesian dimensions) has been chosen:

$$T^\theta = w_\theta + f w \frac{a_\theta x^\theta}{r[1 + e^{-f(\theta - 0.5)}]},$$

(66)

$$F = 0,$$

(67)

(no sum on $\beta$), where $a_\theta = (1, a, b, c)$ are constants, and

$$r \equiv \sqrt{x^2 + a^2 y^2 + b^2 z^2 + c^2 t^2}.$$  

(For example, in one dimension, $a = b = c = 0$; in two dimensions, $b = c = 0$; and so on.) Since we are still interested at this stage in testing the FEM machinery and not the astrophysical viability of the code, simple Dirichlet boundary conditions are once again employed, so equation (4) becomes

$$f = w - \frac{1}{\exp[f(r(\partial \Omega) - 0.5)] + 1},$$

(68)

rather than, for example, a multipole expansion of the interior solution. The above conservative form (eqs. [66] and [67]) was chosen in favor of other forms (such as eqs. [62] and [63]) because its solutions are particularly accurate for a small number of nodes and are suitable for demonstrating adaptive gridding and the use of logarithmic variables.

Figure 2 shows the solution of this Poisson problem in one dimension as one applies successively more features of the code. The left and right top panels of Figure 2 show standard fixed, equally spaced grids of 8 and 16 linear elements, respectively. Some improvement in accuracy can be obtained by doubling the resolution, but this incurs additional storage and computational expense. Turning on the adaptive gridding equations (53), however (middle panels), significantly improves the accuracy for the same number of elements. This also demonstrates one aspect of the iso-parametric transformation: variable node spacing. A closer

| TABLE 2 |
|---------|
| **ERROR RESULTS FOR CARTESIAN GRID TESTS** |
|---------|
| **NORMALIZED $L_2$ ERROR FOR ELEMENT TYPE** |
| $n$ | $D$ | Linear$^a$ | Quadratic$^a$ |
|-------|-----|-----------|-------------|
| 1. . . . | 1 | $b$ | $b$ |
| 1 . . . . | 2 | $2.5 \times 10^{-3}$ | $1.2 \times 10^{-3}$ |
| 1 . . . . | 3 | $1.4 \times 10^{-3}$ | $4.0 \times 10^{-4}$ |
| 1 . . . . | 4 | $9.9 \times 10^{-4}$ | $1.8 \times 10^{-4}$ |
| 2 . . . . . | 1 | $5.8 \times 10^{-3}$ | $b$ |
| 2 . . . . . | 2 | $3.3 \times 10^{-3}$ | $b$ |
| 2 . . . . . | 3 | $2.3 \times 10^{-3}$ | $b$ |
| 2 . . . . . | 4 | $1.8 \times 10^{-3}$ | $b$ |
| 3 . . . . . | 1 | $1.1 \times 10^{-2}$ | $1.2 \times 10^{-3}$ |
| 3 . . . . . | 2 | $6.1 \times 10^{-3}$ | $6.2 \times 10^{-4}$ |
| 3 . . . . . | 3 | $4.0 \times 10^{-3}$ | $4.0 \times 10^{-4}$ |
| 3 . . . . . | 4 | $3.0 \times 10^{-3}$ | $2.9 \times 10^{-4}$ |
| 4 . . . . . | 1 | $2.1 \times 10^{-2}$ | $3.1 \times 10^{-3}$ |
| 4 . . . . . | 2 | $1.0 \times 10^{-2}$ | $1.6 \times 10^{-3}$ |
| 4 . . . . . | 3 | $6.6 \times 10^{-3}$ | $1.0 \times 10^{-3}$ |
| 4 . . . . . | 4 | $4.6 \times 10^{-3}$ | $7.2 \times 10^{-4}$ |

Note. — Meshes with nine nodes in each dimension.

$^a$ Accuracy for linear elements varies with $\Delta x^3$, and for quadratic elements as $\Delta x^5$.

$^b$ Exact solution obtained for one-dimensional first-order problems and for all second-order problems with quadratic elements.
Fig. 2.—One-dimensional results for adaptive grid tests. Top left, solution of the Fermi-Dirac function test with \( f = 50 \), using eight linear, equally spaced elements, and yielding a very large error (\( E^w_{\text{ref}} = 0.27 \)); top right, using 16 equally spaced elements (\( E^w_{\text{ref}} = 0.027 \)); middle left, eight adaptive elements (\( E^w_{\text{ref}} = 0.031 \)); middle right, 16 adaptive elements (\( E^w_{\text{ref}} = 0.0068 \)); bottom left, same as middle right, but plotted against \( \log w \); bottom right, results when solving for \( \tilde{w} = \log_{10} w \). The dotted line shows the exact solution.
examination of this more accurate solution, however (Fig. 2, bottom left), shows very large relative errors in the log when $w < 1$. Nevertheless, these can be overcome easily without rewriting and recoding the equations, as the bottom right panel of Figure 2 shows. When $w$ is identified as a logarithmic variable rather than a linear one, the solution remains accurate over 11 orders of magnitude. (Note the different node spacing in the bottom panels, with the grid adapting to $w$ on the left and $\tilde{w} = \log_{10} w$ on the right.)

Figure 3 shows a similar development for a two-dimensional Poisson problem in which the Fermi-Dirac surface has an elliptical ratio of 2:1 ($a = 2$). The first three panels demonstrate the errors possible in locating the surface if the proper coordinate geometry is not used. Additional improvements in accuracy can be obtained by using adaptive gridding (Fig. 3, middle right). However, this solution for $r > 0.5$ suffers the same oscillatory errors seen in the one-dimensional case (Fig. 3, bottom left), which again can be eliminated by identifying $w$ as logarithmic (bottom right).

It is important to note that in all of the solutions displayed in Figures 2 and 3, no explicit curvilinear coordinate system is used. The coordinates of the grid points (whether

---

**Fig. 3.** Two-dimensional results for adaptive grid tests, showing contour plots of the solution $w$. All use meshes of $16 \times 16$ linear elements with $f = 50$, $a = 2$, and $b = c = 0$ in the Fermi-Dirac function. In all cases, coordinates and differential equations are expressed in $x$ and $y$ only, and although derivatives are calculated on the curvilinear grid, they are immediately transformed to the $(x, y)$ system and used as such in the equations. Top left, uniformly spaced Cartesian grid ($E_{w}^2 = 0.081$); top right, circular-polar grid ($E_{w}^2 = 0.067$); middle left, elliptical grid with the same axial ratio as the solution ($E_{w}^2 = 0.037$); middle right, adaptive elliptical grid allowing finer resolution of the Fermi surface ($E_{w}^2 = 0.010$); bottom left, same as middle right, but with logarithmic contours (note oscillations similar to those in the bottom left panel of Fig. 2); bottom right, resulting solution when solving for $\tilde{w} = \log_{10} w$. 
fixed or part of an adaptive gridding solution) are stored only as \( x_i \) and \( y_i \), not as \( r_i \) and \( \theta_i \) (for example), and yet they are still fully arbitrary (subject to the Jacobi condition). The Poisson equation is written only in terms of \( x \) and \( y \) as well. Of course, the derivatives are still calculated using the coordinate grid shown, but then they are immediately transformed to \((x, y)\) space using the transformations given in equations (7) and (8). Thus, with these new techniques, the grid can be moved around to obtain a more accurate solution, while the physical equations remain coded in the same very simple form.

The Fermi-Dirac Poisson tests were used to determine a good value for the adaptive gridding constant in equation (54). Several dozen models like those in Figures 2 and 3 were computed for different values this parameter. It was found that the accuracy improved by factors of 3–10 as \( C_1 \) was increased from 0 to 0.2, but beyond this point the accuracy did not improve much. In fact, for values much greater than 0.2, the models became unstable, often not converging. Therefore, \( C_1 = 0.2 \) was chosen as a semiu universal value in the adaptive gridding equation. It has proven to be useful in both the Fermi-Dirac tests in Figures 2 and 3 and the stellar structure models below.

One important point about adaptive gridding should be mentioned. As currently implemented, the technique is rather volatile and unstable. Unless great care is taken, iterations with adaptive grids often diverge, violating the Jacobi condition in the process. In the case where a single solution to a steady state problem is sought, sometimes less CPU time cost will be incurred by subdividing the mesh more finely or using quadratic elements, rather than using adaptive gridding techniques. On the other hand, when many thousands of successive models are to be computed, as is the case for evolutionary problems, each newly converged model will be a good initial approximation to the next evolutionary state, yielding convergence for each time step in only a few iterations. In this case, the amount of time spent converging the first adaptively gridded model will be a small cost compared to the CPU time adaptive gridding saves over the course of the evolution by using a smaller number of elements and nodes to obtain the same high level of accuracy.

4.3. Stellar Structure Tests: Polytropic Stars

The fourth series of tests adds the ability to solve a coupled set of both first- and second-order partial differential equations. It also demonstrates the use of \((-g)^{1/2}\) to solve a problem in which the basic coordinate system (not just the grid) is curvilinear, as a consequence of assuming symmetry conditions involving a coordinate direction orthogonal to the computational domain.

4.3.1. Spherical Polytropes in One Dimension

In one dimension, the equations for polytropic stellar structure are hydrostatic equilibrium (Euler’s equation with zero velocity), Poisson’s equation for gravity, and the polytropic equation of state,

\[
\frac{dp}{dr} + \rho(n + 1) \frac{dw}{dr} = 0 , \tag{69}
\]

\[
\frac{1}{r^2} \frac{d}{dr} \left( r^2 \frac{dw}{dr} \right) - \rho = 0 , \tag{70}
\]

\[
p = \rho^{\frac{n+1}{n}} . \tag{71}
\]

The pressure, \( p \), and density, \( \rho \), are unity at the stellar center and zero at its surface, and \( r \) is the spherical radius coordinate. The polytropic index \( n \) is a measure of the hardness of the equation of state; the factor \((n + 1)\) in the hydrostatic equilibrium (HSE) equation is a normalization constant for the gravitational potential \( w \).

In semianalytic treatments, the hydrostatic equilibrium equation is multiplied by \( r^2/\rho \), differentiated with respect to the radius \( r \), and combined with Poisson’s equation to give a single second-order equation. Following this procedure here, however, tests only our ability to solve that simple equation and not much else. A slightly stronger test of the method would be to leave the system as a set of coupled equations, and identify

\[
T_r = p_r + \rho(n + 1)w_r ,
\]

\[
F_p = 0
\]

\[
T_w = w_r , \tag{72}
\]

\[
F_w = \rho
\]

\[
\sqrt{-g} = r^2 , \tag{74}
\]

with boundary conditions at \( r = 0 \) of \( p = 1 \) and \( p_r = 0 \). Note the need for a non–unit volume element of \( r^2 \) in order to form the proper divergence. A basic curvilinear coordinate system must be used because of the spherical symmetry assumed in directions orthogonal to \( \epsilon_r \).

Unfortunately, the above approach is still unsatisfactory, because it cleverly casts HSE as a second-order equation, avoiding the first-order equation problems discussed earlier. In addition, as shown below, it does not lend itself to generalization to two or more dimensions. To address these issues, the following alternate identification of the stress and force terms for the pressure equation has been chosen:

\[
T_r = 0 \tag{75}
\]

\[
F_p = p_r + \rho(n + 1)w_r , \tag{76}
\]

which casts HSE as a first-order equation (with only the boundary condition \( p = 1 \) at \( r = 0 \)). Boundary conditions on the potential \( w \) are \( w_r = 0 \) at the stellar center and \( w = -r_{\text{surface}}/r \) at the stellar surface. In addition, the condition \( p = p_s = 10^{-4} \) is applied at the surface on the adaptive gridding equation to determine the stellar radius.

Figure 4 shows an \( n = 1 \) polytrope in one dimension, which has the analytic solution

\[
\rho = \frac{\sin r}{r} , \quad w = -\rho - 1 , \quad p = \rho^2 .
\]

The left panel uses only the Galerkin method, while the right two panels show the solution using the Petrov-Galerkin scheme in equation (37) to integrate the HSE equation. The need to treat first-order equations differently from second-order ones is clearly evident. The leapfrogging first-order differences produced by the Galerkin method not only display point-to-point oscillations, but also miss an implicit boundary condition \( p_r = 0 \) at \( r = 0 \), implied by \( w_r = 0 \) and eq. [76]). The Petrov-Galerkin scheme improves the accuracy considerably, and with adaptive gridding remains roughly second-order accurate for linear elements and third-order accurate for quadratic.
At present, no robust, automated method for determining the order of the differential equation has yet been developed. The code must be told explicitly not only the order of each equation but also the location(s) of the boundary conditions. While this is the greatest obstacle to producing a truly general continuum simulation code to solve all types of equations, it is a relatively modest amount of effort compared to writing a new code for each problem.

4.3.2. Rotating Polytropes in Two Dimensions

Treatment of a uniformly rotating polytropic star in two dimensions poses several problems. First, it appears to be an overdetermined system,

$$\mathbf{V}p + \rho(n + 1)\mathbf{V}w - \rho R E_{\text{rot}} e_R = 0 \ ,$$

$$\frac{1}{R} \frac{\partial}{\partial R} \left( R \frac{\partial w}{\partial R} \right) + \frac{\partial^2 w}{\partial Z^2} - \rho = 0 \ ,$$

$$p = \rho^{(1 + 1/n)} \tag{79}$$

(where $E_{\text{rot}} \equiv \omega^2/4\pi G \rho$ is the normalized rotational energy per unit mass, $\omega$ is the uniform angular velocity of stellar rotation, and $R$ and $Z$ are the cylindrical radius and axis coordinates), with four equations but only three unknowns. In normal astrophysical situations this dilemma will not arise, since the fluid equations plus the conservation of energy are a well-posed problem. However, it occurs here because two unknowns ($v_R$ and $v_Z$) and only one equation (conservation of mass) have been removed from the full set. The trick is to convert the two redundant equations for HSE (eq. [77]) into only one.

One possible solution is to form a single second-order equation by taking the divergence of the HSE equation, similar to the standard semianalytic approach. However, while it works fine in one dimension, this approach is unstable to two-dimensional perturbations when both the Dirichlet and Neumann boundary conditions are applied on the same ($r = 0$) surface, leaving the stellar surface free. Setting $p = p_s$ at the surface does not help either; in this case the mesh must become adaptive, and this constraint must be used as a boundary condition on the adaptive gridding equations, not on HSE. Moving the Neumann condition to the stellar surface is a better approach, but difficult to apply for more complex problems (e.g., rotating polytropes).

An approach that does work is to project the HSE equation along a direction in which $p$ and $w$ have significant gradients. The projection direction need not be along $e_p \equiv -\mathbf{V}p / |\mathbf{V}p|$; $e_p$ appears sufficient, even when the polytrope is rotating rapidly. But it must not be orthogonal to $e_p$, along which direction the gradients are zero. The components of the general stress-force equation for the rotating polytrope are, therefore, similar to equations (75)–(76) and equations (72)–(74):

$$T_p^i = 0 \tag{80}$$

$$F_p = e_p^i [p_{,i} + \rho(n + 1)w_{,i}] - \rho R E_{\text{rot}} e_R^i \tag{81}$$

$$T_w^i = w_{,i} \tag{82}$$

$$F_w = \rho \tag{83}$$

$$\sqrt{-g} = R \ , \tag{84}$$

with a sum on $i$ over $R$ and $Z$, boundary conditions $p = 1$ and $e_p^i w_{,i} = 0$ at the stellar center, and at the stellar surface

$$f_p = p - p_s \ , \tag{85}$$

$$f_w = w - w_s \ , \tag{86}$$

where $p_s \ll 1$ is a small fraction of the central pressure and $w_s$ is the specified surface potential.

Two different methods were tested for calculating $w_s$. The first was an exterior multipole expansion,

$$w_s^M = \sum_{\ell=0}^{L} M_{\ell} r^{-\ell-1} P_{\ell} (Z/r) \ , \tag{87}$$

where $P_{\ell}$ are Legendre polynomials and $M_{\ell}$ are the moments of the mass distribution in the star,

$$M_{\ell} = \int_{\Omega} \rho(R, Z) r^{\ell} R dR dZ \ , \tag{88}$$

which, because of additional equatorial plane symmetry, are nonzero only for even $\ell$. Generally, orders up to $L = 12$ were used. The second method used a Green’s function inte-
F. Fig. 5.—Two-dimensional $n = 1$ spherical polytropic stars with $9^2$ nodes, using the multipole boundary condition ($w^m$). Top: mesh (left) and pressure (right) contours for (bi-) linear elements; bottom, mesh (left) and pressure (right) contours for (bi-) quadratic elements. Contours follow the precise interpolation within the respective elements. Note the adaptive gridding near the stellar surface, similar to that in Fig. 4.

The integral in equation (89) is over both northern and southern hemispheres of the star. If, as is often the case, reflection symmetry is assumed at the equator and $\Omega$ refers only to the northern hemisphere, then the integral must take into account the southern contribution as well by summing one term as above plus one term, with $\frac{1}{2} I_e$ replaced by $\sum_{\Omega} I_e$. The single-parameter complete elliptic integral

$$I_e(\alpha) \equiv \int_0^1 \frac{ds}{(1 - \tanh \alpha \cos 2\pi s)^{1/2}}$$

represents the summed relative contributions to the surface potential at $(R, Z)$ from different angular elements of a ring of matter at $(R, Z)$. $I_e$ diverges with $\alpha$ (a measure of how close the ring is to $x_s$), but it can easily be evaluated numerically and tabulated to a part in $10^{10}$ accuracy for the useful range $0 \leq \alpha \leq 10$ (i.e., for rings approaching within only a fraction $2 \times 10^{-10} = 9.1 \times 10^{-5}$ of $|x_s|$), over which the integral lies in the range $1.0 \leq I_e \leq 5.1$. In all models presented here, even with the largest meshes ($33^2$) and adaptive gridding ($R_s/\delta R_s \geq 200$), $\alpha$ remains well below 8.0 and $I_e$ below 4.2.

As implemented in the author’s code, the speed of the integral outer boundary condition technique was significantly slower than the exterior multipole expansion, increasing the time required to form the stiffness matrix.

\[ \]
(although not affecting the time to invert it) by factors of several. While the author made no attempt to optimize the implementation, even after such efforts it nevertheless should remain somewhat expensive, since it requires (for each nonzero stiffness matrix element and each right-hand vector element) a complete integration of the potential (eq. [89]) at 4–6 surface points, each integration being the equivalent of a multipole moment computation. However, while the multipole expansion began to break down for modest rotation speeds, the integral technique converged with no problem for all rotation speeds up to breakup, making the extra computational effort worthwhile and necessary. In addition, since the integrals are done only for surface points, this hybrid technique (differential equation with integral boundary conditions) will still be much cheaper than computing the global potential by performing such an integral for every point in the domain.

Figure 5 shows two-dimensional $n = 1$ nonrotating polytropes for the two element classes, each with $9^2$ nodes (the analog of the middle and right panels of Fig. 4), using the multipole boundary condition. Note especially the variable grid spacing near the stellar surface and the difference in smoothness between linear and quadratic interpolation. Figure 6 shows the same models for the $33^2$ node cases. Note especially the departure from sphericity in the pressure contours in the models employing linear elements, which is absent in the quadratic element cases; the $9^2$ quadratic model is more accurate than the $33^2$ linear model. However, errors in the two cases scale only as

\[ E_{\text{linear}}^{L^2} \propto (N^p)^{-1.0}, \]
\[ E_{\text{quadratic}}^{L^2} \propto (N^p)^{-2.0}, \]

one full order less accurate than expected. This may be a result of the reflective boundary conditions along the axis and equator, which are only first- and second-order accurate, respectively, in the linear and quadratic cases. The nature of this boundary condition cannot be improved at this time, but it will be once iterative/multigrid methods for solving the coupled equations are implemented. This will
FIG. 7.—As in Fig. 6, but using the integral boundary condition \( \omega') \) and solving for a two-dimensional \( n = 0 \) rotating polytropic star (Maclaurin spheroid) with \( \omega^2/2 \pi G \rho = 0.224 \)—very close to the theoretical limit of 0.2246656.

allow boundary “ghost” elements to be handled easily, allowing the application of boundary conditions as accurate as the mesh interior.

The Maclaurin spheroid sequence (a series of \( n = 0 \) uniform density polytropes) provides a full two-dimensional test of the method. (Note that the method takes no advantage of the uniform rotation, uniform density, or polytropicity, so the ability of the code to solve the Maclaurin problem is a good indication of how it will do on general multidimensional stellar structure and other more complex problems.) The appearance of Maclaurin spheroids is similar to that of the \( n = 1 \) polytropes in Figures 5 and 6, but the logarithm of the pressure varies little with radius except near the surface, where the radial grid spacing decreases dramatically because of the sudden pressure drop. For this reason, it is sufficient to use a larger pressure boundary value \( (\rho_p = 10^{-2} \text{ rather than } 10^{-4}) \) in order to determine the locus of the Maclaurin spheroid surface. Figure 7 shows an analogy to Figure 6 for a Maclaurin spheroid with \( \omega^2/2 \pi G \rho = 0.224 \), very near the theoretical limit of 0.2246656 (Tassoul 1978). The integral boundary condition given by equation (89) is used to compute the surface potential.

The complete Maclaurin series from \( \omega^2/2 \pi G \rho = 0 \) to 0.224 was computed in four different ways, using \( 9^2 \) and \( 17^2 \) nodes for the two classes of elements (linear and quadratic), respectively. Values of \( \tau \equiv E_{rot}/|E_w| \), the rotational flattening ratio of the semiminor and semimajor axes, \( f = 1 - a_y/a_R \), and total angular momentum \( J = \int_{R=0}^{R} \rho(R, Z) R^2 d\Omega \) have been computed from these models and are compared in Figure 8 with analytic curves from Tassoul (1978) and Chandrasekhar (1969). The fractional errors for the four series are shown in Figure 9. The models are quite accurate for such a small number of elements, with errors in the \( 10^{-3} \) to \( 10^{-4} \) range in the \( 17^2 \) quadratic case. They show the expected result that third-order interpolation is significantly more accurate than second-order, but the decrease in the errors with increasing numbers of elements is not as steep as expected. Most of this behavior is probably due to the less accurate reflective boundary conditions mentioned earlier.

5. DISCUSSION

5.1. Summary

This paper has developed a general method for solving multidimensional structural and dynamical problems of astrophysics. Virtually all situations involving continuous media are potentially addressable, whether in normal flat Cartesian space or in curved spacetime. Problems in this
Fig. 8.—Comparison of rotating Maclaurin spheroid model stars, computed with different finite element meshes, with the analytic solution. Top, linear elements with 9² (left) and 17² (right) nodes, respectively; bottom, quadratic elements with 9² (left) and 17² (right) nodes. Although ω² is the primary model parameter, the flattening ratio, total angular momentum, and ω² itself, with the normalizations in Tassoul (1978), are plotted as a function of the derived parameter τ, for comparison with the analytic theory.
Fig. 9.—Absolute value of the fractional errors for the Maclaurin spheroids as shown in Fig. 8 as a function of $\omega^2$. Solid lines show models with linear elements, dashed lines show quadratic elements; curves without symbols show 92 node models, and curves with symbols show 172 models.
area include, but are not limited to, the full structure and secular evolution of viscous, rotating (and even magnetized) stars and accretion disks in two and three dimensions, interacting binaries, asymmetric stellar envelopes and winds, nonradial pulsating stars, nonlinear development of secular and thermal accretion disk instabilities, and stationary or evolving spacetimes.

While this method is most useful for structures that evolve on timescales that are long compared to a dynamical time, there is no formal restriction on how short the evolution time must be. Therefore, the approach to dynamical instabilities from a stable configuration, and even initial dynamical development, can also be studied, although the author still recommends the use of an explicit code for full dynamical evolution.

The equations of continuum astrophysics have been condensed into a general compact covariant form, and that form has been encoded into the author’s FEM program GENRAL. A user can solve a particular astrophysics problem by supplying a single subroutine that takes as input one given coordinate position plus the value of the variables and their gradients there, and returns as output the differential equations (i.e., the generalized stress tensor, body-force vector, and possible boundary conditions appropriate for that problem). The program then generates the nodal or “difference” equations on a user-specified general curvilinear grid using the finite element weighted residual integrals, and solves the large set of coupled equations to produce the solution to the equations. While described by discrete nodal values, as in finite difference methods, the finite element solution is continuous, as in spectral methods, because of the finite element interpolation functions. Either second-order (linear interpolation) or third-order (quadratic interpolation) accurate solutions are possible in the code. In addition, the positions of the nodes themselves can be part of the solution (a “rubber” mesh), allowing grids to be fitted to unknown boundary shapes and regions of high gradients in order to be more finely resolved with the same number of mesh points.

While the method is cast in a full covariant form, it is anticipated that the initial applications will be mainly in the area of nonrelativistic stars or accretion disks in static gravitational fields. The covariant form, however, is important even for nonrelativistic problems. When the mesh extends into the time domain, even for only one or two elements, the coordinate transformations that are a natural component of the finite element method automatically generate any arbitrary Lagrangian-Eulerian (ALE) advective derivatives needed to take possible grid motion into account.

The method has been demonstrated on astrophysically interesting problems (spherical or rotating polytropic stars) in one and two dimensions, with full adaptive gridding, and on simpler problems in three and four dimensions.

5.2. A Note on the Solution of Elliptic Potential Problems

A great deal has been written on the numerical solution of astrophysical potential problems, such as equation (78). The technique used here has elements of past approaches plus some new features, and is well suited to the FEM. Like many past authors (Clement 1978; Bonazzola et al. 1998), the approach here casts the problem as a differential equation with boundary conditions specified on the exterior of the domain. However, rather than being a simple $1/r$ or low-order multipole potential at a large radius in the vacuum region, the author’s preferred boundary condition is an integral solution of the differential equation specified at the stellar surface. This integral is physically equivalent (on that surface) to the “full integral” technique that computes the potential throughout the computational domain using a Green’s function integral rather than solving the differential equation itself (Tassoul (1978); Eriguchi & Müller 1985; Komatsu, Eriguchi, & Hachisu 1989). However, the author’s method of evaluating this integral is somewhat different, as it requires no expansion in terms of Legendre polynomials, relying instead on a single, slowly varying, numerically tabulated function $I_1(\alpha)$ to handle the axisymmetry of the potential field. When calculated in this manner, using the integration techniques already available in the FEM, the numerical integral is a solution of the discrete finite element equations themselves to within the truncation error. The boundary condition and interior differential equation therefore match well, leading to good convergence of the models.

This technique will work in any situation where the full integral technique can be used: extension of the Newtonian case into three dimensions will be trivial, and it will be straightforward for the general relativistic case as well. In three dimensions there is no axisymmetry, so $I_1(\alpha)$ will not be needed, and $|x - x_0|$ will take the simple Pythagorean form. For axisymmetric relativistic stars, the four metric potentials are given by three Green’s function integrals plus a first-order equation (Komatsu et al. 1989). Therefore, the three elliptic equations can be solved in the same manner as Poisson’s equation is solved here (although probably using different tabulated $I_1$ functions), preserving the differential equations in the stellar interior but using the integral solution on the surface. The fourth equation would be handled with the Petrov-Galerkin scheme demonstrated in § 4. The advantage of this approach compared to the full integral technique is speed (the number of volume integrals is proportional to the domain surface area, not the volume). The advantage over the multidomain technique is convenience (one does not have to deal with the vacuum region and the matching of stellar and vacuum solutions).

5.3. Unresolved Issues

While the code and method are mature enough to begin solving two-dimensional structural problems routinely, there are several unresolved issues mentioned in the text that must be addressed more completely before the full potential of the astrophysical finite element method can be realized.

First and foremost are the execution speed and memory issues. While the reader may consider the generation of the transformations and finite element integrals rather time consuming, by far the greatest use of computer resources is the technique currently used to solve the coupled equations, the Henyey technique. For large meshes in three or more dimensions, it becomes prohibitively expensive, requiring thousands to many millions of years of CPU time ($\propto [N^{2P}][10^{-2}]$) and equally absurd amounts of memory ($\propto [N^{2P}][10^{-1}]$) to invert once. However, multigrid methods (Brandt 1977) need only about twice the grid size in storage ($\propto [N^{2P}]$) and only require a few sweeps of the mesh to converge ($\propto [N^{2P}][\log [N^{2P}]]$). The author and P. Godon have been experimenting with modern parallel multigrid algorithms in finite difference codes with considerable
success. The CPU time and memory scalings, as well as linear speedup on parallel supercomputers, have all been realized. Efforts are currently underway to make GENRAL a parallel, multigrid FEM code.

Implementation of iterative schemes such as the multigrid for solving the equations will make straightforward the application of accurate reflective and periodic boundary conditions. While this process is possible with the Henyey technique, it is much more difficult since it involves columns of matrix elements far from the diagonal and special techniques for inversion. With iterative methods, as with explicit codes, one can enclose the computational domain in a layer of “ghost” elements whose properties are determined at each iteration by the interior solution. The ghost-element approach will have the same order of accuracy as the interpolation scheme, unlike the current approach for the reflective boundary condition, which uses essentially a backward difference.

Another possibly important issue is time evolution. All examples in this paper, even the four-dimensional ones, are time-independent and use a Cartesian metric. The inclusion of time dependence may be as simple as employing a Minkowskian metric and time derivatives of the variables, and letting the finite element machinery solve the problem. Often, however, the addition of a new feature generates new numerical problems that require modification of that machinery. Until more experience is obtained with time-dependent problems, it is not clear whether the techniques discussed here are complete or whether they will need additional major development to handle evolutionary situations.

Finally, many issues remain concerning the use of the finite element method for dynamical evolution problems. These are currently important topics in the engineering field, but because explicit finite difference codes do well for astrophysical problems in this area, development of these issues here will have lower priority. These issues include adaptive mesh refinement (for dynamical collapse situations), implementation of the general boundary conditions given in equation (2) (for magnetohydrodynamics and solving Maxwell’s or Einstein’s equations), and proper upwinding schemes with behavior comparable to the higher order Godunov schemes (for problems that develop shocks).

5.4. A Note on Numerical Relativity with Finite Element Analysis

In numerical relativity, it is customary to perform a “3 + 1 split” of the metric, such that

$$ds^2 = (-\alpha^2 + \beta_i \beta^i)dt^2 + 2\beta_i dx^i dt + \gamma_{ij} dx^i dx^j,$$  \hspace{1cm} (91)

where $\gamma_{ij}$ is the three-metric that raises or lowers indices on the shift three-vector $\beta_i$, and $\alpha$ is the “lapse function,” of which are functions of position in spacetime (Arnowitt, Deser, & Misner 1962; York 1979). The three-metric is specified on the initial hypersurface by solving the field constraints (initial value data), the lapse and shift are computed from four coordinate (or “gauge”) conditions, and the Einstein field equations are used to evolve $\gamma_{ij}$ to the next hypersurface. The goal is to choose a gauge in which the hypersurfaces do not intersect a singularity before a significant amount of evolution occurs in some part of the mesh.

The current method for avoiding singularities is to eliminate pathological parts of spacetime from the mesh (“excise the black hole”); Cook et al. 1998).

While such an approach is also possible with the FEM (Arnold, Mukherjee, & Pouly 1998), advancing time in a step-by-step fashion, the full covariant nature of the method, and the lifting of the degeneracy between basic and mesh coordinate systems allow additional approaches. In particular, it becomes possible to extend the mesh fully in the time dimension, from initial to final hypersurface, choosing a relatively simple gauge for $\alpha$ and $\beta_i$. Then, adaptive gridding in all four dimensions can be used to keep the grid boundaries away from singularities and to further adjust the separation in time between spacelike hypersurfaces. Because of the isoparametric transformation, the foliation no longer needs to be along surfaces of constant time $x^0$. The separation between adjacent surfaces can be nonuniform, the time coordinate can vary considerably over the hypersurfaces, and the final hypersurface can even end at different times. In effect, the adaptive gridding completes, in mesh coordinates, the job of slicing and singularity-avoiding that a poor gauge choice may fail to do. One advantage of this approach is that some or all of the field constraints can be applied on the final, instead of initial, hypersurface, turning an explicit hyperbolic problem into an implicit boundary value problem (such as stellar structure), and possibly stabilizing the growth of errors.

However, while such techniques probably can succeed in keeping physical singularities at bay, it is doubtful that they can avoid coordinate singularities in general situations. (These arise in the most benign of curved surfaces—on the surface of the earth, for example.) Apart from knowing the geometry beforehand and choosing the proper basic coordinate system, there are only a few ways to avoid these problems entirely. One is to embed the spacetime in a higher dimensional, flat Minkowskian space. In principle, as many dimensions as independent spacetime metric coefficients (ten) would be needed for the embedding, although it may be possible with fewer. From the ten hyperspace coordinates, and how they vary in the four-dimensional mesh, one then could derive the local metric of the spacetime and use it in the physical equations. These ten equations for the metric in terms of the hypercoordinates, plus the six Einstein equations and the four adaptive gridding equations, would be sufficient to determine the twenty independent $g_{ab}$ and hypercoordinates at each node in the finite element mesh. While it would be a fairly immense job for present-day computers, this prescription has the advantage of being singularity-free in general situations.

Another method for avoiding coordinate singularities using the FEM is to dispense with global coordinates entirely, using only line-segment lengths and deficit angles to describe the geometry and the Regge calculus to describe the physics (Regge 1961; Holst 1998). At present, this approach has been developed only for simplex-type elements and not hypercubes, so it is not a straightforward application of the code discussed here. However, it may be useful to recast the Regge calculus for other element types.

Finally, all methods that involve a full four-dimensional finite element spacetime are probably well beyond the capabilities of present computer technology, even with the use of parallel multigrid techniques. Nevertheless, they appear to have such attractive features and elegance that it is important to begin to develop them.
The author is grateful to J. Fanselow for support during the early development of this work, to L. Caroff and M. Bicay at NASA for allowing a small portion of a theoretical astrophysics grant to be used for this purpose, and to the JPL Director’s Research and Development Fund for support to complete this work. Discussions with numerous people were very helpful, including G. Lyzenga and A. Raefsky on the finite element method, M. Norman on adaptive gridding, P. Godon on spectral methods and multigrid methods, and W. Cannon, S. Finn, M. Holst, K. Thorne, and J. York on the use of these techniques for general relativity. This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under contract to the National Aeronautics and Space Administration.

APPENDIX A
CASTING OF THE DIFFERENTIAL EQUATIONS OF CONTINUUM ASTROPHYSICS INTO GENERAL
FINITE ELEMENT FORM

This appendix shows that virtually all the equations of astrophysics of continuous media can be cast into the flux-conservative, finite element form (eq. [1]) and their boundary conditions into equation (2). That is, while possibly second-order in spatial and time derivatives, they can be written as the four-divergence of a generalized stress tensor plus a generalized body force vector, each of which are functions of no more than the first spacetime derivative (four-gradient) of the variables. Of course, it is always possible to define additional variables (e.g., the 24 connection coefficients) and turn the field equations and conservation laws into first-order equations involving only the $F_{4}$ term in equation (1). The challenge, however, is to use only the original metric and field components as variables (avoiding additional computational expense) and still maintain the flux-conservative form. Below is one solution to this problem.

A1. THE EQUATIONS IN GEOMETRIC FORM

The discussion here is concerned only with differential equations. Local physics, such as the equations of state, opacity, emissivity, viscosity, etc., is not treated in detail. While having position and time dependence, these processes can be described (where $G$ is the covariant gradient operator, the antisymmetric Maxwell tensor $F$ is the dual of the antisymmetric Faraday tensor $F$, and $J$ is the four-current. With the symmetry, equations (A1) are 10 in number, and equations (A2) and (A3) constitute 8, for a total of 18. However, because of identities satisfied by the Einstein and Faraday tensors, there are actually only 12 independent equations (6 metric and 6 electromagnetic), but 16 unknowns at each point in space: the 10 independent components of the metric $g$ and the 6 independent components (the electric and magnetic fields $E$ and $B$) of the antisymmetric Faraday tensor. The remaining four-metric unknowns are determined by the choice of a coordinate system or gauge.

The standard method for generating a set of 12 evolutionary equations is to project equations (A1)-(A3) into the hypersurface normal to a timelike vector (or world line) $n$ with the projection tensor (eq. [3]). For example, if $n_{a} = g_{0a}(-g_{00})^{1/2}$, then only the spatial part $\mathcal{P}_{i}$ will be nonzero, with $i, j = 1, 2, 3$. In general, however, $n$ can be any timelike vector, so the equations will be left in general form. If the 12 spacelike components of equations (A1)-(A3) are satisfied throughout the four-dimensional spacetime domain $\Omega$ (with one factor of $\mathcal{P}$ for each tensor order),

$$\mathcal{P} \cdot \mathcal{P} \cdot \mathcal{P} = 8\pi \mathcal{P} \cdot \mathcal{P} \cdot \mathcal{P} \; ;$$

$$\mathcal{P} \cdot (\nabla \cdot \mathcal{F}) = 4\pi \mathcal{P} \cdot \mathcal{F} \; ;$$

$$\mathcal{P} \cdot (\nabla \cdot \mathcal{M}) = 0 \; ,$$

then all that is necessary to satisfy the timelike components

$$n \cdot \mathcal{P} = 8\pi n \cdot \mathcal{P} \; ;$$

$$n \cdot \nabla \cdot \mathcal{F} = 4\pi n \cdot \mathcal{F} \; ;$$

$$n \cdot \nabla \cdot \mathcal{M} = 0 \; ,$$

throughout all spacetime is to satisfy the latter equations on one hypersurface only. Equations (A4)-(A6), therefore, are the 12 independent differential equations to be solved for the six metric and six electromagnetic field components, while equations (A7)-(A9) are the constraints that need to be satisfied in order for a solution to exist at all. (For reference, eq. [A5] is Ampere’s
law, eq. [A6] is Faraday’s law, eq. [A7] contains the Hamiltonian and momentum constraints, by further contraction with \( n \) or \( \mathcal{J} \), respectively, eq. [A8] is Coulomb’s law, and eq. [A9] is the solenoidal condition on the magnetic field.

Equation (A4), with equation (A7) as initial conditions, constitutes the Cauchy problem of general relativity. Equations (A6) and (A9) are the covariant form of the Evans-Hawley “constrained transport” method for enforcing the solenoidal constraint in nonrelativistic magnetohydrodynamics (Evans & Hawley 1988). Equations (A5) and (A8) represent constrained transport in the presence of sources. When equations (A4)–(A6) are solved as Cauchy problems, equations (A7)–(A9) are applied on the initial hypersurface. However, since our approach here is to relax the system for a four-dimensional spacetime, instead of evolving a three-dimensional surface forward in time, they can be applied on any spacelike hypersurface.

In addition to the field equations, there are conservation laws that follow from identities satisfied by the fields. The Einstein curvature tensor is constructed in such a way that \( \nabla \cdot \mathcal{E} = 0 \), so the conservation of energy and momentum

\[
\nabla \cdot \mathcal{F} = 0
\]

must also hold from equation (A1). Similarly, since \( \mathcal{F} \) satisfies \( \nabla \cdot (\nabla \cdot \mathcal{F}) = 0 \), then the four-current must also be conserved,

\[
\nabla \cdot \mathcal{J} = 0 \, .
\]

The field equations are then “closed” by expressing the SEM tensor and four-current in terms of the state variables, and solving the conservation laws of energy and momentum for those variables. For most conceivable astrophysical situations, including those with multifluid dynamics, electromagnetic fields and currents, radiation, viscosity, and nuclear reactions, expressions for \( \mathcal{F} \) and \( \mathcal{J} \) involve terms with, at most, first-order derivatives of the state variables with respect to space or time. This is true even in situations near black hole horizons, where particle interaction and fluid flow time scales are comparable, and equations like Ohm’s law, for example, are no longer valid.

A final group of differential equations comes from forming the zeroth, first, and second moments of the Boltzmann-Vlasov equation for each particle species (\( j \) (photons, nuclei, etc.). These determine each species’ individual number density, velocity (including the peculiar drift velocity \( q^{(j)} \)), and internal energy per particle. The kinetic equations give rise to familiar processes such as nuclear burning, radiative transport, viscosity, and electrical conductivity. Nevertheless, they all have the same “conservative” form, with the divergence of a term that involves (at most) first-order spatial derivatives of the state variables.

For example, the zeroth moment of these kinetic equations yields

\[
\nabla \cdot N^{(j)} \left[ \mathbf{u} + q^{(j)} \right] = c^{(j)},
\]

where \( N^{(j)} \) is the particle number density and \( c^{(j)} \) is the net particle creation rate due to nuclear reactions. For many astrophysical situations, particle conservation will be the only one of these equations needed, the drift velocity and particle energies being determined by the diffusion or other approximations.

A2. THE EQUATIONS IN COMPONENT FORM

With no loss of generality, one can choose to write the differential equations in a coordinate frame. In this case, the connection coefficients are given by

\[
\Gamma^a_{\beta\gamma} = \frac{1}{2} g^{\alpha\delta} \left( g_{\alpha\beta,\gamma} + g_{\alpha\gamma,\beta} - g_{\beta\gamma,\alpha} \right) ,
\]

and their trace reduces to

\[
\Gamma_{\beta\mu} = \left( \ln \sqrt{-g} \right)_{,\beta} ,
\]

where \( g \) is the determinant of the metric

\[
g \equiv \det |g| .
\]

This yields simple expressions for the gradients and divergences in curved space, and the field equations (eqs. [A4], [A5], and [A6]) and conservation laws (eqs. [A10], [A11], and [A12]) become

\[
\begin{align*}
[ \mathcal{F} \left( \Gamma^a_{\alpha \beta} \mathcal{F}^a - \mathcal{F}^a \Gamma^a_{\alpha \beta} \right) \sqrt{-g} \right]_{,\mu} - \left[ \mathcal{F}^a \mathcal{F}^b \left( \Gamma^a_{\alpha \mu} \Gamma^b_{\nu \beta} - \Gamma^b_{\alpha \mu} \Gamma^a_{\nu \beta} \right) + \left( \mathcal{F}^a \mathcal{F}_a \right)_{,\mu} - \left( \mathcal{F}_a \mathcal{F}^a \right)_{,\mu} \right] \Gamma^a_{\alpha \beta} \\
+ 8 \pi \left( \mathcal{F}^a \mathcal{F}_a + \frac{1}{2} \mathcal{F}_a \mathcal{F}^a \right) \sqrt{-g} = 0 ,
\end{align*}
\]

\[
\begin{align*}
\left( \mathcal{F}^a \sqrt{-g} \right)_{,\beta} - \left( \mathcal{F}^a \mathcal{F}^b \sqrt{-g} \right)_{,\beta} = 0 ,
\end{align*}
\]

\[
\begin{align*}
\left( \mathcal{F}^a \sqrt{-g} \right)_{,\beta} - \left( \mathcal{F}^a \mathcal{F}^b \sqrt{-g} \right)_{,\beta} = 0 ,
\end{align*}
\]

\[
\begin{align*}
\left( \mathcal{F}^a \sqrt{-g} \right)_{,\beta} + \Gamma^a_{\alpha \beta} \sqrt{-g} = 0 ,
\end{align*}
\]

\[
\begin{align*}
\left( \mathcal{F}^{(j)} \sqrt{-g} \right)_{,\beta} = 0 ,
\end{align*}
\]

\[
\begin{align*}
\left[ N^{(j)} \left( \mathbf{u} + q^{(j)} \right) \sqrt{-g} \right]_{,\beta} - c^{(j)} \sqrt{-g} = 0 ,
\end{align*}
\]

which are all in the conservative finite element form of equation (1). Use has been made of the expression for the Einstein tensor,

\[
\mathcal{E}_{ab} = R_{ab} - \frac{1}{2} g_{ab} R ,
\]
in terms of the Ricci tensor,
\[ R_{ab} = \frac{1}{\sqrt{-g}} (\Gamma^c_{ab} \sqrt{-g})_{;c} - (\ln \sqrt{-g})_{,ab} - \Gamma^c_{ca} \Gamma^a_{cb}, \]
where the Ricci scalar is the contraction \( R \equiv R_{\mu\nu} \) and the SEM scalar is the contraction \( \mathcal{S} \equiv \mathcal{S}_{\mu}^{\mu} \). Note that since equations (A16) are symmetric in \( i \) and \( j \), they are six in number, and equations (A17) and (A18) are three equations each. The term inside the divergence in equation (A16) is the trace-free curvature—the “flux” of the metric; the “force” terms involve derivatives of the projection tensor and a matter source generating the curvature.

Boundary conditions on the conservation laws are generally of the simple form of equation (4). Furthermore, it is well known that the Einstein field constraints do not depend on second derivatives with respect to time, and the Maxwell constraints do not depend on time derivatives at all. Therefore, it should be possible to write these in the form of equation (2), with at most a divergence on the boundary and a gradient normal to it. One of the more compact versions of this is
\[ (\mathcal{S}^{\mu\nu} \Gamma_{\mu\nu} \sqrt{-g} )_{,\nu} - (\Gamma_{\mu\nu} \sqrt{-g} g^{\mu\nu} - [\mathcal{S}^{\mu\nu}(\Gamma_{\mu\nu} \Gamma_{\mu\nu} - \Gamma_{\mu\nu} \Gamma_{\mu\nu}) + \mathcal{S}^{\mu\nu} \Gamma_{\mu\nu} + 8\pi(n^\mu \mathcal{S}_{\mu\nu} - \frac{1}{2} \mathcal{S})] \sqrt{-g} = 0, \]  
\[ (n^\mu \mathcal{S}_{\mu\nu} \sqrt{-g} g^{\mu\nu} - (\mathcal{S}^{\mu\nu} n^\mu \mathcal{S}_{\mu\nu} - 4\pi \mathcal{S}^\nu \mathcal{S}^\nu) \sqrt{-g} = 0, \]
where \( n^\mu \) is now identified as the boundary normal. Equations (A22) and (A23) are, respectively, the Hamiltonian and momentum constraints.

In addition to the Einstein field equations (A16), four more coordinate conditions and their possible boundary conditions, are needed to completely specify the metric. Since these are rather arbitrary, they are left unspecified for now. (Of course, if they are to be differential equations, they also must be in the form of eq. [1].)

APPENDIX B

CONDITIONS ON THE ISOPARAMETRIC TRANSFORMATION

Not all \( \mathcal{L}_s \) specified by equation (7) are valid transformations in all situations. Sometimes they must satisfy certain conditions, which are discussed below. Two of these are conditions on the new (element mesh) metric formed by the transformation, and the third is a condition on the element spacing itself. However, only the first condition always needs to be satisfied for a stable mesh. The other two are necessary only under certain circumstances.

B1. THE JACOBI CONDITION

The most important of these conditions is that the Jacobian determinant of the transformation be nonzero,
\[ \mathcal{L} \equiv \det |\mathcal{L}| \neq 0. \]
However, since \( \mathcal{L}\mathcal{L} \mathcal{L} \mathcal{L} = g' \), and since \( g < 0 \), this also means that the determinant of the new metric must be negative,
\[ g' = \det |g'| < 0. \]  
(B1)
The Jacobi condition (eq. [B1]) ensures that the unit vectors in the element (primed) system, while not necessarily orthogonal, nevertheless form an independent coordinate system that has an arrow of time. It is absolutely necessary that this condition be satisfied in order for the mesh to be well behaved.

B2. THE LOCAL-LORENTZ CONDITION

A second possible condition is that the element coordinate system may have a locally Lorentz character everywhere—i.e., that \( \xi^0 \) is the mesh time coordinate. This ensures that if the spatial portion of the element mesh moves with time, the mesh velocity will always be less than the speed of light. However, this is important only if the mesh is used as a frame of reference for measuring physical quantities.

There are various ways of ensuring the Lorentz nature of the transformation \( \mathcal{L}_s \). The simplest and safest way is to ensure that each unit vector in the new space satisfies the proper timelike or spacelike constraint. With the constraints that \( \xi^0 \) denotes the time dimension and that \( e_x \cdot e_v = g_{x\beta} \), these local Lorentz conditions become
\[ g_{00} > 0, \quad g_{00} > 0, \quad g_{\xi\xi} < 0, \quad g_{\xi\xi} < 0. \]  
(B2)
Equation (B3) is equivalent to demanding that each of the element sides in the \( \xi^0 \) direction be spacelike.

A less stringent, but still sufficient, condition on \( g_{\xi\xi} \) could be derived by choosing a specific timelike vector, such as \( n^\mu = \delta^\mu_0 / (\sqrt{-g_{00}})^{1/2} \) (which still requires \( g_{00} < 0 \)), and then constructing from the corresponding projection tensor a set of
three independent vectors orthogonal to \( n^\mu \),
\[
\delta^\mu_i = n_i n^\mu + \delta^\mu_i .
\]
The condition that these vectors be spacelike \((s_i \cdot s_i > 0)\) leads to a modified form for equation (B3),
\[
g_t > \frac{-\left(g_{00}/\sqrt{-g_{00}}\right)^2}{2} .
\]
In this less restrictive case, the \( e_i \) unit vectors can be a bit timelike, but no more so than that given by the above inequality.

**B3. THE COURANT CONDITION**

In standard nonrelativistic computational fluid dynamics, in order for the (explicit) forward integration in time to be stable, the distance traversed in a single time step by sound waves or by the fluid itself (whichever is faster) must be substantially less than the mesh spacing. The ratio of these distances, called the Courant number \( C \), is chosen to be \( \sim 0.1 - 0.4 \) or so depending on the stability of the numerical integration scheme. This nonrelativistic Courant condition (which can be written as \(-\Delta t^2/C^2v_{\text{max}} + \Delta x^2 > 0\) for one-dimensional flow) easily generalizes in the four-dimensional general relativistic case to
\[
g_{a\beta} \Delta x^a \Delta x^\beta > 0 ,
\]
where the vector \( \Delta x = (\Delta t/Cv_{\text{max}} , \Delta x , \Delta y , \Delta z) \). It further generalizes in the case of a general curvilinear element mesh to
\[
g_{a\beta} \Delta x^a \Delta x^\beta > 0 \tag{B4}
\]
in each element, with \( \Delta x = (\Delta x^0/Cv_{\text{max}} , \Delta x^1 , \Delta x^2 , \Delta x^3) \); \( \Delta x^0 \) is the width of the element in each mesh dimension, and \( v_{\text{max}} \) is a three-velocity magnitude equal to the maximum disturbance speed within the element. Because stable implicit techniques are used in time as well as space, a Courant number very close to unity probably can be tolerated. Therefore, in the case of relativistic flow, where \( v_{\text{max}} = 1 \), the Courant condition reduces to the requirement that the geodesics connecting opposing corners in each element must be spacelike \((g_{a\beta} \Delta x^a \Delta x^\beta = \Delta s^2 > 0)\).

Generally, the Courant condition given in equation (B4) is much too restrictive and is routinely violated in slowly evolving or steady state problems, where time steps are very long or even infinite. In implicit codes, the condition needs to be satisfied only if one wishes to follow every short-timescale transient phenomenon or wave.

**REFERENCES**

Abrahimowitz, M., & Segun, I. A. 1965, Handbook of Mathematical Functions (New York: Dover)
Aller, L. H., & McLaughlin, D. B. 1965, Stellar Structure (Chicago: Univ. Chicago Press)
Arnold, D. N., Mukherjee, A., & Pouly, L. 1998, in Numerical Analysis 1997, ed. D. F. Griffiths & G. A. Watson (Reading: Addison Wesley)
Arnovitt, R., Deser, S., & Misner, C. W. 1962, in Relativity, Groups and Topology (New York: Wiley), 227
Bonazzola, S., Gourgoulhon, E., & Marck, J.-A. 1998, Phys. Rev. D, in press
Brandt, A. 1977, Math. Comp., 31, 333
Chandrasekhar, S. 1957, An Introduction to the Study of Stellar Structure (New York: Dover)
1969, Ellipsoidal Figures of Equilibrium (New Haven: Yale Univ. Press)
Clayton, D. D. 1968, Principles of Stellar Evolution and Nucleosynthesis (New York: McGraw-Hill)
Clement, M. F. 1978, ApJ, 222, 967
1994, ApJ, 420, 797
Colella, P. 1990, J. Comput. Phys., 87, 171
Colella, P., & Woodward, P. R. 1984, J. Comput. Phys., 54, 174
Cook, G. B., et al. 1998, Phys. Rev. Lett., 80, 2512
Dubinski, D. R., & Hernquist, L. 1997, New Astron., 2, 277
Eriguchi, Y., & Muller, E. 1985, A&A, 146, 260
1991, A&A, 248, 435
Evans, C. R., & Hawley, J. F. 1988, ApJ, 332, 659
Holst, M. 1998, in preparation
Hughes, T. J. R. 1987, The Finite Element Method (Englewood Cliffs: Prentice-Hall)
Kippenhahn, R., & Thomas, H. C. 1970, in Stellar Rotation, ed. D. Slettebak (Dordrecht: Reidel)
Komatsu, H., Eriguchi, Y., & Hachisu, I. 1989, MNRAS, 237, 355
Leahy, D. A. 1997, in The 12th Kingston Meeting: Computational Astrophysics, ed. D. A. Clarke & M. J. West (San Francisco: ASP), 109
LeBlanc, J., & Wilson, J. 1998, ApJ, 161, 541
Misner, C. W., Thorne, K. S., & Wheeler, J. A. 1973, Gravitation (San Francisco: Freeman)
Monaghan, J. J. 1992, ARA&A, 30, 543
Norman, M. L. 1997, in The 12th Kingston Meeting: Computational Astrophysics, ed. D. A. Clarke & M. J. West (San Francisco: ASP), 3
1998, in Numerical Astrophysics 1998, forthcoming
Park, Y.-S., & Hong, S. S. 1998, ApJ, 494, 605
Press, W. H., Flannery, B. P., Teukolsky, S. A., & Vetterling, W. T. 1989, Numerical Recipes (Cambridge: Cambridge Univ. Press)
Regge, T. 1961, Nuovo Cimento, 19, 558
Tassoul, J.-L. 1978, Theory of Rotating Stars (Princeton: Princeton Univ. Press)
Truelove, J. K., Klein, R. I., McKee, C. F., Holliman, J. H., Howell, L. H., Grenough, J. A., & Woods, D. T. 1998, ApJ, 495, 821
van Leer, B. 1979, J. Comput. Phys., 89, 125
York, J. W. 1979, in Sources of Gravitational Radiation, ed. L. L. Smarr (Cambridge: Cambridge Univ. Press), 83
Zienkiewicz, O. C. 1977, The Finite Element Method (London: McGraw-Hill)