JOURNAL OF COMPUTATIONAL PHYSICS 73, 59–72 (1987)

Finite Element Method with Optimal Nodal Velocity

ICHIRO KAWAKAMI

Atomic Energy Research Institute,
College of Science and Technology, Nihon University,
1-8 Kanda-Surugadai, Chiyodaku, Tokyo 101, Japan

Received April 11, 1986; revised July 11, 1986

The finite element method FEMALE [15] is outlined. Accuracy is much improved when the velocity of a finite element is equal to that of a local nonlinear wave. In this paper we give a proof of the improvement of accuracy for the discretized equations. The need for node annihilation is discussed for a hyperbolic system of equations and a restructuring algorithm of finite elements is proposed. © 1987 Academic Press, Inc.

I. INTRODUCTION

FEMALE [15] is a numerical method for solving a system of evolutionary partial differential equations. It was formulated in the frame of the finite element method (FEM), since FEM is adaptable to a variety of fluid dynamical systems. A characteristic feature of the method is that finite elements can move in space with arbitrary velocity and are arbitrary–Lagrangian–Eulerian (ALE).

Numerical accuracy is much improved when the velocity of the finite elements equals that of a nonlinear wave such as a shock wave, contact discontinuity, or simple wave for the hyperbolic system of equations. Our previous formulation was restricted to the hyperbolic system of conservative form. We shall show here that the method can also be applied to a parabolic system of equations.

The price paid for the adaptive mesh motion is a distortion of finite elements which rapidly decreases the numerical accuracy. To remedy this distortion we have to reconstruct the finite element layout.

In Section II the formulation of our method is summarized. In Section III the optimal nodal velocity, in view of numerical accuracy, is given and discussed for the equations discretized in space. In the final section, annihilation, creation, and reconnection of nodes are discussed. Reconnecton usually requires extra storage and execution time, so we propose an efficient algorithm of reconnection.

II. MATHEMATICAL FORMULATION

Consider the following system of partial differential equations of conservation form

$$\partial U/\partial t + \operatorname{div} \mathbf{F} = 0, \tag{1}$$

0021-9991/87 $3.00

where U is an n-column vector of unknowns. $\mathbf{F}$ is a 3-vector function of U, and three components of $\mathbf{F}$ (F_x, F_y, and F_z in the cartesian coordinate system) are n-column vectors. An example of such a system of equatons would be three-dimensional equations of ideal magnetohydrodynamics [1]:

$$U = \begin{pmatrix} \rho \\ \rho \mathbf{v} \\ \varepsilon \\ \mathbf{B} \end{pmatrix}, \qquad \mathbf{F} = \begin{pmatrix} \rho \mathbf{v} \\ \mathbf{P} \\ \mathbf{g} \\ \mathbf{E}^* \end{pmatrix},$$

where $\mathbf{P}$ and $\mathbf{E}^*$ are dyadic and

$$\begin{aligned} \varepsilon &= \tfrac{1}{2}\rho v^2 + \tfrac{1}{2}B^2 + \tfrac{3}{2}p \\ \mathbf{P} &= (p + \tfrac{1}{2}B^2)\,\mathbf{I} + \rho \mathbf{v}\mathbf{v} - \mathbf{B}\mathbf{B} \\ \mathbf{g} &= [\tfrac{1}{2}\rho v^2 + \tfrac{5}{2}p]\,\mathbf{v} - (\mathbf{v} \times \mathbf{B}) \times \mathbf{B} \\ \mathbf{E}^* &= \mathbf{v}\mathbf{B} - \mathbf{B}\mathbf{v} \end{aligned}$$

and ρ, $\mathbf{v}$, p, and $\mathbf{B}$ are the fluid density, velocity, pressure, and magnetic field in the appropriate units. In this case, U is an eight-component vector ($n = 8$), and Eq. (1) expresses conservations of the mass, momentum, energy, and magnetic flux.

The physical space, Ω, is divided into Ne finite elements, R_e ($e = 1, 2, ..., Ne$).

$$\Omega = \bigcup_{e=1}^{Ne} R_e,$$

so that they are not overlapped,

$$R_e \cap R_f = 0 \qquad \text{(null) for } e \neq f.$$

For simplicity, we are considering the simplex model, so the finite elements are line segments (1D), triangles (2D), and tetrahedra (3D). At the vertices of the finite elements, we locate N nodes. The coordinates of the nodes are denoted by $\mathbf{x}^\mu$ ($\mu = 1, 2, ..., N$). Let the nodal values of $U(\mathbf{x}, t)$ and $F(\mathbf{x}, t)$ be denoted by $U^\mu = U(\mathbf{x}^\mu, t)$ and $\mathbf{F}^\mu = \mathbf{F}(\mathbf{x}^\mu, t)$.

We introduce a set of shape functions (or interpolation functions),

$$\{\phi_1(\mathbf{x}, t), \phi_2(\mathbf{x}, t), ..., \phi_N(\mathbf{x}, t)\}$$

such that ϕ_μ's are linearly independent, and they span the function space which is subspace approximate to the space spanned by $U(\mathbf{x}, t)$. ϕ_μ is localized in the sense that $\phi_\mu(\mathbf{x}^\nu, t) = \delta_{\mu\nu}$ and $\phi_\mu(\mathbf{x}, t) = 0$ if $\mathbf{x}$ is located outside the elements which include $\mathbf{x}^\mu$. Since we are considering the simplex model, $\phi_\mu(\mathbf{x}, t)$ is a linear function of $\mathbf{x}$ in the finite elements which include the μth node.

The crux of our finite element method is that ϕ_μ is not only the function of space, $\mathbf{x}$, but also that of time, t, such that ϕ_μ satisfies the following hyperbolic equation:

$$\partial\phi_\mu(\mathbf{x}, t)/\partial t + \mathbf{v}_g \cdot \text{grad}\ \phi_\mu(\mathbf{x}, t) = 0. \tag{2}$$

Equation (2) shows that $\phi_\mu(\mathbf{x}, t)$ is constant along the characteristics

$$d\mathbf{x}/dt = \mathbf{v}_g(\mathbf{x}, t) \tag{3}$$

or the finite elements move with the local velocity $\mathrm{v}_g(\mathbf{x}, t)$. The nodal velocity is denoted by $\mathbf{v}_g{}^\mu \equiv \mathbf{v}_g(\mathbf{x}^\mu, t)$. Heretofore, $\mathbf{v}_g(\mathbf{x}, t)$ has not been specified. When $\mathbf{v}_g = 0$, the finite element is Eulerian, and when $\mathbf{v}_g = \mathbf{v}$ (the fluid velocity) it is Lagrangian. As we can specify $\mathbf{v}_g$ arbitrarily, it is arbitrary–Lagrangian–Eulerian (ALE [21]).

We approximate $U(\mathbf{x}, t)$ by mapping

$$U(\mathbf{x}, t) \simeq U^*(\mathbf{x}, t) = \sum_{\mu=1}^{N} U^\mu(t)\, \phi_\mu(\mathbf{x}, t), \tag{4}$$

where $U^*(\mathbf{x}, t)$ is approximate to $U(\mathbf{x}, t)$. Note that $U^\mu(t)$ is the nodal value of $U(\mathbf{x}, t)$ and $U^\mu(t) = U^*(\mathbf{x}^\mu, t) = U(\mathbf{x}^\mu, t)$. According to Galerkin's method [3], we require that the residual of Eq. (1) is orthogonal to all $\phi_\mu(\mathbf{x}, t)$ $(\mu = 1, 2, 3, ..., N)$:

$$\int_\Omega d\Omega\ \phi_\mu(\partial U^*/\partial t + \text{div}\ \mathbf{F}(U^*)) = 0. \tag{5}$$

Substituting Eq. (4) into Eq. (5), we obtain

$$\begin{aligned} \sum_{\nu=1}^{N} \alpha_{\mu\nu} dU^\nu/dt &= \int_\Omega d\Omega\ \phi_\mu(\mathbf{v}_g \cdot \text{grad}\ U^* - \text{div}\ \mathbf{F}^*) \\ &= \sum_{\nu=1}^{N} \mathbf{R}_{\mu\nu} \cdot \mathbf{v}_g{}^\nu - S_\mu, \end{aligned} \tag{6}$$

where

$$\alpha_{\mu\nu} = \int_\Omega d\Omega\ \phi_\mu \phi_\nu \tag{7a}$$

$$\begin{aligned} \mathbf{R}_{\mu\nu} &= \int_\Omega d\Omega\ \phi_\mu \phi_\nu\ \text{grad}\ U^* \\ &= \sum_\rho^N \boldsymbol{\beta}_{\mu\nu\rho} U^\rho \end{aligned} \tag{7b}$$

$$\begin{aligned} S_\mu &= \int_\Omega d\Omega\ \phi_\mu\ \text{div}\ \mathbf{F}^* \\ &= \sum_{i=1}^{3} \sum_{\nu=1}^{N} \boldsymbol{\beta}_{i\mu\nu} F_i^\nu \end{aligned} \tag{7c}$$

$$\boldsymbol{\beta}_{\mu\nu\rho} = \int_\Omega d\Omega\, \phi_\mu \phi_\nu \operatorname{grad} \phi_\rho \tag{7d}$$

$$\boldsymbol{\beta}_{i\mu\nu} = \int_\Omega d\Omega\, \phi_\mu \operatorname{div}(\phi_\nu \mathbf{e}_i) \tag{7e}$$

and $\mathbf{e}_i$ is the unit vector along the x_i-axis. Note that $\mathbf{R}_{\mu\nu}$ is a 3-vector with three n-column vector components. We approximated $\mathbf{F}(\mathbf{x}, t)$ and $\mathbf{v}_g(\mathbf{x}, t)$ as

$$\mathbf{F}(\mathbf{x}, t) \simeq \mathbf{F}^*(\mathbf{x}, t) = \sum_{\mu=1}^{N} \mathbf{F}^\mu(t)\, \phi_\mu(\mathbf{x}, t) \tag{8}$$

$$\mathbf{v}_g(\mathbf{x}, t) \simeq \mathbf{v}_g{}^*(\mathbf{x}, t) = \sum_{\mu=1}^{N} \mathbf{v}_g{}^\mu(t)\, \phi_\mu(\mathbf{x}, t); \tag{8}$$

$\alpha_{\mu\nu}$, $\boldsymbol{\beta}_{\mu\nu\rho}$, and $\beta_{i\mu\nu}$ are geometrical factors, independent of the unknown $U^\mu(t)$ and analytically evaluated by a standard finite element method [4]. Thus, the system of partial diffeential equations (1) is (approximately) equivalent to Eq. (6) and

$$dx^\mu/dt = \mathbf{v}_g{}^\mu(t) \tag{10}$$

which determines the motion of the nodes $\mathbf{x}^\mu$, with prescribed initial conditions. Some nodes are located on the boundary of the system. For these nodes, Eqs. (6) and (10) are replaced by the boundary conditions. We are now in a position to solve Eqs. (6) and (10) if the nodal velocities $\mathbf{v}_g{}^\mu$s are specified.

II. Optimal Nodal Velocity

The nodal velocity $\mathbf{v}_g{}^\mu$ is arbitrary. This arbitrariness introduces freedom into the numerical method. We would like to make use of this freedom to improve numerical accuracy. If the right-hand side of Eq. (6),

$$C_\mu \equiv \int_\Omega d\Omega\, \phi_\mu(\mathbf{v}_g \cdot \operatorname{grad} U - \operatorname{div} \mathbf{F}) \tag{11a}$$

$$= \sum_\nu \mathbf{R}_{\mu\nu} \cdot \mathbf{v}_g{}^\nu - S_\mu \tag{11b}$$

can be minimized by choosing appropriate $\mathbf{v}_g{}^\mu$, then the rate of change of U^μ

$$dU^\mu/dt = \sum_{\nu=1}^{N} \alpha^{\mu\nu} C_\nu, \tag{12}$$

where $\alpha^{\mu\nu}$ is an element of the inverse of the matrix $(\alpha_{\mu\nu})$, would also be minimized and the numerical solution of Eq. (6) is easy and efficient. So a reasonable requirement for determining the nodal velocity $\mathbf{v}_g^\mu$ is such that the Euclidian norm

$$\|C\|^2 \equiv \sum_{\mu=1}^{N} \|C_\mu\|^2$$

is minimized, where $C=(C_1^{\mathrm{T}}, C_2^{\mathrm{T}}, ..., C_N^{\mathrm{T}})^{\mathrm{T}}$ and T denotes transposition of a vector. Note that $C_1, C_2, ..., C_N$ are n-column vectors.

In some cases, $\mathbf{v}_g^\mu$ obtained in this way makes $C_\mu=0$. In such cases, Eq. (6) reduces to

$$dU^\mu/dt=0 \tag{13}$$

so U^μ is constant along

$$dx^\mu/dt=v_g^\mu. \tag{14}$$

Hence the solution U^μ is exact and free from any numerical errors such as numerical diffusion or numerical oscillation. The time step size Δt will be restricted by truncation error for the solution of Eq. (14). The square pulse propagation problem [5, 15], the solid body rotation problem [6], as well as the Riemann problem [7, 15] belong to this kind of problems. Note that these problems are used as test problems for numerical schemes and are not trivial since the numerical solution of these problems suffers strong numerical oscillations and diffusions for most conventional algorithms.

In order to see how we can make $C_\mu=0$, we note that $C_\mu=0$ is the weak form of the Rankine–Hugoniot relation [8].

When the discontinuity passes through the node $\mathbf{x}^\mu$, C_μ is expressed as

$$C_\mu=\mathbf{n}\cdot\mathbf{v}_g^\mu[U]-[\mathbf{n}\cdot\mathbf{F}^\mu], \tag{15}$$

where $\mathbf{n}$ is the unit normal vector to the surface of discontinuity, and [ ] denotes the jump across the discontinuity. Hence if $\mathbf{n}\cdot\mathbf{v}_g^\mu$ is the normal speed with which the discontinuity moves, then $C_\mu=0$. For smooth U, the integrand is

$$\phi_\mu(\mathbf{v}_g\cdot\operatorname{grad} U-\operatorname{div}\mathbf{F})=\phi_\mu(\mathbf{v}_g-\mathbf{J})\cdot\operatorname{grad} U,$$

where

$$\mathbf{J}=(J_x, J_y, J_z)\equiv(\partial F_x/\partial U, \partial F_y/\partial U. \partial F_z/\partial U) \tag{16}$$

is the Jacobian. The Jacobian has n real and distinct eigenvalues for a hyperbolic system of equations [9]. If grad U is an eigenstate with eigenvalue λ, then the integrand vanishes when $\mathbf{v}_g=\lambda$. For the constant state, where grad $U=\operatorname{div}\mathbf{F}=0$, the integrand is zero. This completes the proof that $C_\mu=0$ for the above test problems. In these problems we have only to find the nodal position $\mathbf{x}^\mu$ by solving Eq. (14) with $\mathbf{v}_g^\mu$ equal to the local velocity of the nonlinear wave at $\mathbf{x}^\mu$.

Even when C_μ does not vanish for any $\mathbf{v}_g{}^\mu$, the nodal velocity which minimizes $\|C\|$ would be optimal in view of the numerical integration of Eq. (6). If the optimal nodal velocity can be found analytically, it should be substituted into C_μ and Eq. (6) integrated numerically. When the optimal velocity cannot be found analytically, we have to find it numerically. This is the least squares problem, to find $\mathbf{v}_g{}^\mu$ for given U^μ and $\mathbf{F}^\mu$.

We would like to see the relation between the optimal nodal velocity obtained by the least squares problem and the eigenvalue of the Jacobian matrix (16). We restrict ourselves to the hyperbolic equations. The theory of a generalized inverse of matrix [10] gives the least squares solution of equations $C_\mu = 0$ ($\mu = 1, 2, ..., N$) as

$$\mathbf{v}_g{}^\mu = \sum_\nu \mathbf{R}^{+\mu\nu} S_\nu$$
$$+ \sum_{\nu,\rho} (\mathbf{I}\delta_{\mu\nu}\delta_{\nu\rho} - \mathbf{R}^{+\mu\nu}\mathbf{R}_{\nu\rho}) \cdot \mathbf{v}_{g0}{}^\rho, \tag{17}$$

where S_ν and $\mathbf{R}_{\nu\rho}$ are defined in Eq. (7), $\mathbf{R}^{+\mu\nu}$ is the Moore–Penrose generalized inverse of $\mathbf{R}_{\mu\nu}$, and $\mathbf{v}_{g0}{}^\rho$ is an arbitrary velocity. $\mathbf{R}^{+\mu\nu}$ is uniquely determined by the following four conditions:

$$\sum_{\rho,\lambda} \mathbf{R}_{\mu\rho} \cdot \mathbf{R}^{+\rho\lambda}\mathbf{R}_{\lambda\nu} = \mathbf{R}_{\mu\nu} \tag{18a}$$

$$\sum_{\rho,\lambda} \mathbf{R}^{+\mu\rho}\mathbf{R}_{\rho\lambda} \cdot \mathbf{R}^{+\lambda\nu} = \mathbf{R}^{+\mu\nu} \tag{18b}$$

$$\sum_\rho (\mathbf{R}^{+\mu\rho}\mathbf{R}_{\rho\nu})^{\mathrm{T}} = \sum_\rho \mathbf{R}^{+\nu\rho}\mathbf{R}_{\rho\mu} \tag{18c}$$

$$\sum_\rho (\mathbf{R}_{\mu\rho} \cdot \mathbf{R}^{+\rho\nu})^{\mathrm{T}} = \sum_\rho \mathbf{R}_{\nu\rho} \cdot \mathbf{R}^{+\rho\mu}. \tag{18d}$$

The second term of Eq. (17) includes the arbitrary vector $\mathbf{v}_{g0}{}^\rho$, is orthogonal to the first terms, and does not contribute to C_μ since

$$\sum_{\nu,\rho} \mathbf{R}_{\mu\nu} \cdot (\mathbf{I}\delta_{\nu\rho}\delta_{\rho\lambda} - \mathbf{R}^{+\nu\rho}\mathbf{R}_{\rho\lambda}) = 0.$$

The first term of Eq. (17) is the least squares solution of the minimal norm $\sum_\mu (\mathbf{v}_g{}^\mu)^2$, and the second term shows arbitrariness of the solution for the least squares problem. For this $\mathbf{v}_g{}^\mu$,

$$C_\mu = \sum_{\nu,\rho} (\mathbf{R}_{\mu\rho} \cdot \mathbf{R}^{+\rho\nu} - \mathbf{I}\delta_{\mu\rho}\delta_{\rho\nu})\, S_\nu. \tag{19}$$

Let the Jacobian $\mathbf{J}$ at $\mathbf{x}^\mu$ be $\mathbf{J}^\mu$ and assume that $\mathbf{J}^\mu$ has n real and distinct eigenvalues $\lambda_k{}^\mu$ ($k = 1, 2, ..., n$). Let the normalized (right) eigenvector corresponding to $\lambda_k{}^\mu$ be $\mathbf{r}^k{}_\mu$:

$$\mathbf{J}^{\mu}\cdot\mathbf{r}^{k}{}_{\mu}=\boldsymbol{\lambda}_{k}{}^{\mu}\cdot\mathbf{r}^{k}{}_{\mu} \tag{20a}$$

and

$$(\mathbf{r}^{k}{}_{\mu})^{\mathrm{T}}\cdot\mathbf{r}^{k}{}_{\mu}=1, \tag{20b}$$

where the $\mathbf{r}^{k}{}_{\mu}$'s form a normalized complete set in the n-vector-valued 3-vector space. In addition to hyperbolicity, if we assume that $\mathbf{J}^{\mu}$ is symmetric, then

$$(\mathbf{r}^{i}{}_{\mu})^{T}\,\mathbf{r}^{j}{}_{\mu}=\delta_{ij}. \tag{20c}$$

$\mathbf{R}_{\mu\nu}$ can be expressed as

$$\mathbf{R}_{\mu\nu}=\sum_{k=1}^{n}a_{\mu\nu k}\mathbf{r}^{k}{}_{\nu}. \tag{21}$$

Then

$$S_{\mu}=\sum_{\nu}\mathbf{J}^{\nu}\cdot\mathbf{R}_{\mu\nu}=\sum_{\nu,k}a_{\mu\nu k}\boldsymbol{\lambda}_{k}{}^{\nu}\cdot\mathbf{r}^{k}{}_{\nu} \tag{22}$$

and so

$$\begin{aligned}\mathbf{v}_{g}{}^{\mu}=&\sum_{\nu,k}W^{\mu}{}_{\nu k}\cdot\boldsymbol{\lambda}_{k}{}^{\nu}\\&+\sum_{\nu,\rho}(\mathbf{I}\delta_{\mu\nu}\delta_{\nu\rho}-\mathbf{R}^{+\mu\nu}\mathbf{R}_{\nu\rho})\cdot\mathbf{v}_{g0}{}^{\rho}\end{aligned} \tag{23}$$

$$C_{\mu}=\sum_{\nu,\lambda,k}a_{\lambda\nu k}\left(\sum_{\rho}\mathbf{R}_{\mu\rho}\cdot\mathbf{R}^{+\rho\lambda}-\mathbf{I}\delta_{\mu\lambda}\right)\mathbf{r}^{k}{}_{\nu}\cdot\boldsymbol{\lambda}_{k}{}^{\nu}. \tag{24}$$

where

$$W^{\mu}{}_{\nu k}\equiv\sum_{\rho}a_{\rho\nu k}\mathbf{R}^{+\mu\rho}\mathbf{r}^{k}{}_{\nu} \tag{25}$$

is a three-by-three weight matrix. If $\mathbf{R}_{\mu\nu}$ is an eigenstate, that is, $a_{\lambda\nu k}\mathbf{r}_{\nu}{}^{k}=\mathbf{R}_{\lambda\nu}$ for some k, then $C_{\mu}=0$ as expected, and $W^{\mu}{}_{\nu k}=\sum_{\rho}\mathbf{R}^{+\mu\nu}\mathbf{R}_{\rho\nu}$ is a symmetric projection operator, $\sum_{\rho}W^{\mu}{}_{\rho k}W^{\rho}{}_{\nu k}=W^{\mu}{}_{\nu k}$, with the eigenvalues zero and unity. These properties come from the definition of $\mathbf{R}^{+\mu\nu}$ in Eqs. (18). For a one-dimensional problem with a symmetric J^{μ},

$$R_{\mu\nu}=\sum_{k}a_{\mu\nu k}r^{k}{}_{\nu} \tag{26}$$

$$\begin{aligned}R^{+\mu\nu}&=(R_{\mu\nu}{}^{\mathrm{T}}R_{\mu\nu})^{+}\,R_{\mu\nu}{}^{\mathrm{T}}\\&=\sum_{k}a_{\mu\nu k}(r^{k}{}_{\nu})^{\mathrm{T}}\Big/\sum_{1}a_{\mu\nu 1}{}^{2}\end{aligned} \tag{27}$$

and

$$W^{\mu}{}_{\nu k}=a_{\mu\nu k}{}^{2}\Big/\sum_{1}a_{\mu\nu 1}{}^{2}. \tag{28}$$

In this case, $v_g{}^\mu$ is a weighted sum of nonlinear wave velocities at the nodes around x^μ, where each weight is proportional to the wave energy of the respective wave.

We have to be careful in finding the least squares solution of the nodal velocity $\mathbf{v}_g{}^\mu$ numerically for given $\mathbf{R}_{\mu\nu}$ and S_μ since round-off error or truncation error would produce erroneous singular values which must be replaced by exactly zero in the course of the computation. For this purpose, we implemented the LSQR method [11] in the codes.

However, numerical solution of the least squares problem to find $\mathbf{v}_g{}^\mu$ requires large computer resources, especially in three-dimensional problems. We save computer storage and execution time by lumping symmetric matrices $\alpha_{\mu\nu}$ and $\mathbf{R}_{\mu\nu}$, that is, by replacing $\alpha_{\mu\nu}$ and $\mathbf{R}_{\mu\nu}$ by

$$\alpha_\mu = \sum_{\nu=1}^{N} \alpha_{\mu\nu} = \int_\Omega d\Omega\, \phi_\mu \tag{29a}$$

$$\mathbf{R}_\mu = \sum_{\nu=1}^{N} \mathbf{R}_{\mu\nu} = \int_\Omega d\Omega\, \phi_\mu \operatorname{grad} U^*, \tag{29b}$$

respectively, so that Eq. (6) is simplified as

$$\alpha_\mu dU^\mu/dt = C_\mu, \tag{29c}$$

where

$$C_\mu = \mathbf{R}_\mu \cdot \mathbf{v}_g{}^\mu - S_\mu. \tag{29d}$$

Note that $\sum_\mu \phi_\mu(\mathbf{x}, t) = 1$ for the simplex model. As for the effects of lumping, see Ref. [4].

Heretofore, we have been concerned with the hyperbolic system of equations, where the flux $\mathbf{F}(\mathbf{x}, t)$ is a function of $U(\mathbf{x}, t)$. When $\mathbf{F}$ is a function of the derivatives of U, the system of equations are not hyperbolic and the foregoing discussion cannot be applied. For a parabolic system, our proposal is that the derivatives in $\mathbf{F}$ are deleted first and $\mathbf{v}_g{}^\mu$ is determined according to the discussion above; then C_μ is computed in terms of $\mathbf{F}$ including the derivatives. Unfortunately we are not convinced this algorithm is always best. Sometimes better results may be obtained when $\mathbf{v}_g{}^\mu$ is determined with $\mathbf{F}$ including derivatives. We show numerical solutions of parabolic equations in the following section.

IV. Restructuring of Finite Elements

Moving grid algorithms, such as ICED ALE [2], the MFE [12], the self-adjusting grid method [13], the free-Langrange method [14], and, FEMALE [15], often have troubles caused by an unfavorable mesh deformation. Some of these algorithms use rezoning [2, 16] to restructure the grid layout by changing the nodal positions and assign transported values of variables to the vertices or central points of the cells, yielding numerical diffusion. Another method of restructuring keeps the nodal positions and the nodal values of variables unchanged, but the connections between nearest neighboring nodes are changed. In this reconnection method no numerical diffusions occur, so we recommend reconnection rather than rezoning.

A. *Node Deletion and Addition*

Before discussing reconnection we consider the discontinuity found in the solution of a hyperbolic system of partial differential equations. The simplest, but nontrivial, example is the solution of the following equation:

$$\partial u/\partial t + u\partial u/\partial x = 0. \tag{30}$$

Here $U = u$, $F = u^2/2$, and $J = u$, so the eigenvalue of Jacobian J is $\lambda = u$. Along the characteristics

$$dx/dt = \lambda, \tag{31}$$

u is constant. If initially $u\ (x, t = 0) = f\ (x)$, where $df/dx < 0$ at some x, say $x = 0$, then u will be multivalued at $x = 0$ for $t > -(df/dx)^{-1}$. Mathematical theory, to avoid this physically unacceptable multivaluedness, selects a unique solution among weak solutions [8, 9], regarding Eq. (30) as a limiting equation of Burger's equation:

$$\partial u/\partial t + u\partial u/\partial x = \sigma\partial^2 u/\partial x^2 \tag{32}$$

as $\sigma \to 0$.

This argument suggests that we should delete colliding nodes just before they overtake one another. Figure 1 shows a numerical solution of Eq. (30) under the initial condition $f(x)$ being discontinuous. $v_g{}^\mu$ is determined numerically by the LSQR method and approximately $v_g{}^\mu = (u^{\mu+1} + u^{\mu-1})/2$. We see the right moving shock eats up the right nodes. Figure 2 shows a numerical result of the same equation with $f(x) = -a \sin kx$. In this case, the nodes approach the point $x = 0$ and are annihilated there. In either case, we evaluate $|\partial^2 u/\partial x^2|$ at the overtaking nodes and delete one of these nodes which has the larger $|\partial^2 u/\partial x^2|$.

Next, we show a numerical solution of Burger's equation (32) with finite $\sigma = 10^{-2}$ in Fig. 3. $v_g{}^\mu$ is same as in Fig. 1 so we are computing a diffusion equation without a convection term in the moving frame. For comparison we take $v_g{}^\mu = 0$ in Fig. 4. Oscillation appears around the discontinuity and propagates to the left. This shows that our choice of the nodal velocity is better than that of the Eulerian scheme.

Addition or creation of nodes would be required on the element of large volume. The nodal value U^μ could be obtained by interpolation. The node addition will not be troublesome.

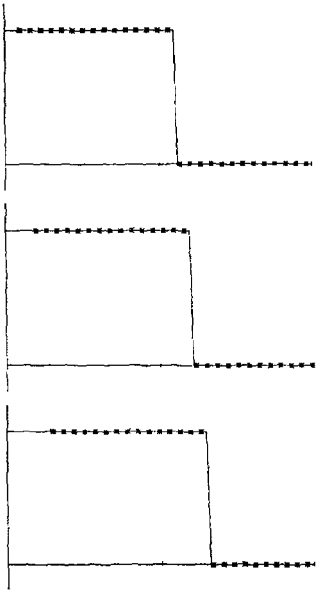

FIG. 1. Numerical solution of the hyperbolic equation (30) with the nodal velocity under a discontinuous initial condition.

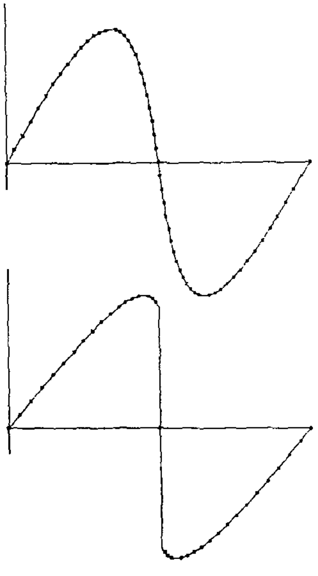

FIG. 2. Numerical solution of the hyperbolic equation (30) with the nodal velocity under a continuous initial condition.

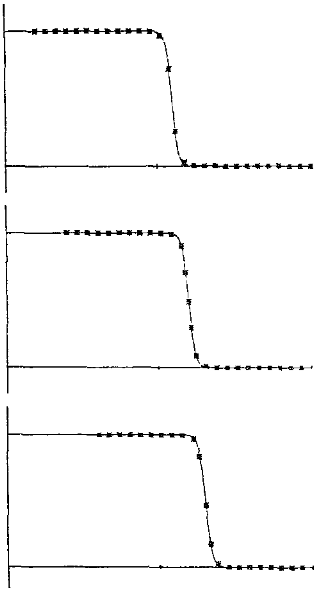

FIG. 3. Numerical solution of Burger's equation (31) with the nodal velocity under a discontinuous initial condition.

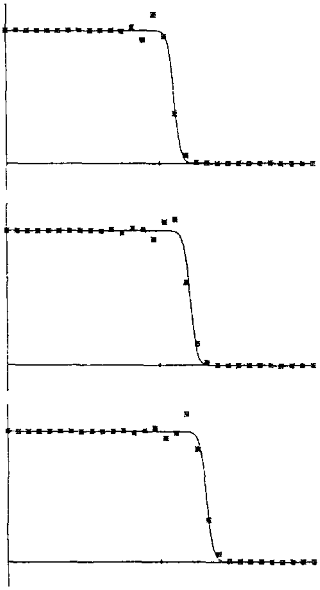

FIG. 4. Numerical solution of Burger's equation (31) with vanishing nodal velocity (Eulerian) under a discontinuous initial condition.

B. *Reconnection Algorithm*

Reconnection is to link one node with the nearest neighboring nodes. In a one-dimensional problem, the nearest neighboring nodes are obvious and reconnection is unnecessary, In two- and three-dimensional problems of N nodes, the "worst" algorithm needs about $N^2/2$ operations (comparisons) while the "best" algorithm needs about $N \log_2 N$ operations. The ratio $N/(2 \log_2 N)$ of these two operations exceeds 50 even for $N = 1024$. We will present a reconnection algorithm which needs about $N \log_2 N$ operations.

In order to visualize reconnection, consider a two-dimensional problem and attempt to fill the square region with triangular elements. N nodes are randomly distributed in the square region except N_b nodes on the boundary (Fig. 5a). The counting problem [17] tells that the number of triangular elements is $N_e = 2N - N_b - 2$. First we sort all N nodes in ascending order of the y-coordinates. This sort algorithm should be as fast as quick sort [18]. The ordered N nodes fall

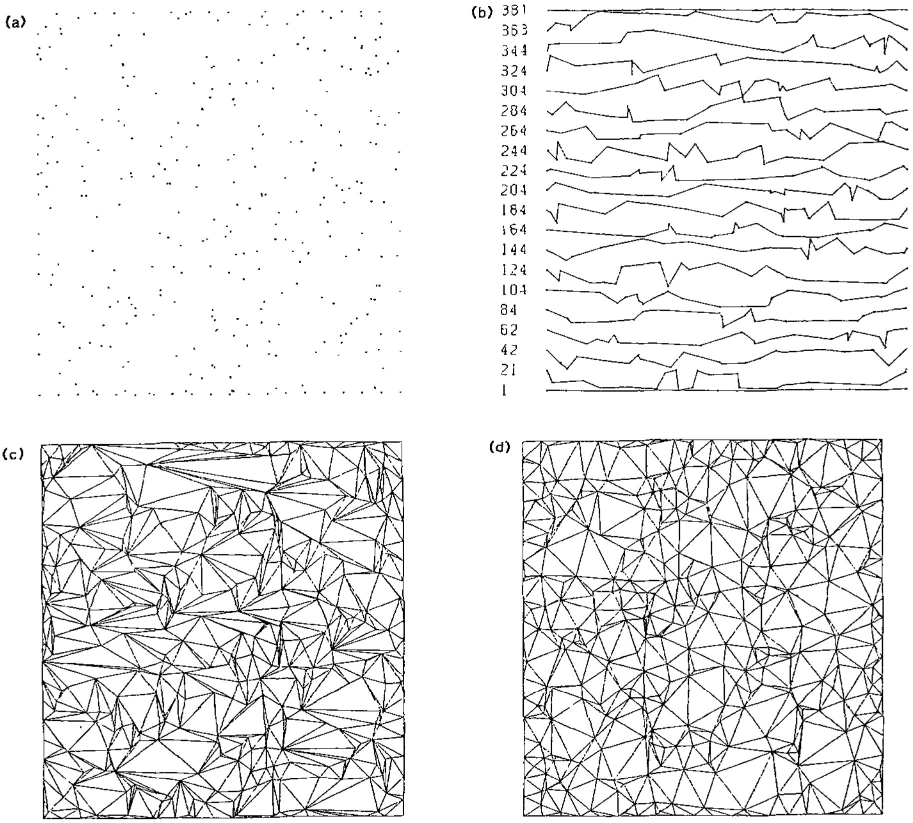

FIG. 5. Reconnection algorithm for $N = 400$: (a) distributed nodes; (b) grouping; (c) reconnected triangular elements; and (d) optimized elements by diagonal flip algorithm.

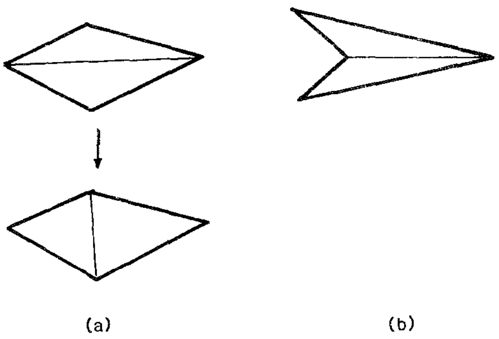

FIG. 6. (a) Diagonal flip and (b) boomerang quadrilateral.

into $\sqrt{N}$ groups of about $\sqrt{N}$ nodes. Each group must have two nodes on the right and the left boundaries, respectively. The members of each group are sorted in ascending order of the x-coordinates and are connected with lines (Fig. 5b). Finally, fill the area between the neighboring two groups with triangles (Fig. 5c). Be careful not to make the triangles overlapped or inverted. This algorithm is very fast since the construction of triangles can be done locally.

It is known that the regular triangle is favorable for the finite element method. Unfortunately some of the triangles resulting from the above algorithm are strongly obtuse. For regularization, the diagonal flip algorithm [17] is applied (Fig. 5d). The diagonal flip algorithm is illustrated in Fig. 6a. It consists of a procedure of eliminating the longer diagonal and inserting the shorter diagonal of the quadrilateral to create triangles. In this algorithm we must not flip diagonals for boomerang quadrilateral (Fig. 6b).

We can extend the above algorithms to three-dimensional problems. We assume that the three-dimensional region is cubic. First we sort N nodes in the ascending order of z-coordinates. Then the ordered N nodes fall into $\sqrt[3]{N}$ groups of about $N^{2/3}$ nodes. These groups correspond to a two-dimensional square although it is not planar. After application of the two-dimensional algorithm to make triangles, we fill the volumes between neighboring two groups with tetrahedra, with three nodes on one group and one node on the other.

ACKNOWLEDGMENTS

The author would like to express his sincere thanks to Dr. M. Aizawa and Mr. K. Suzuki for their help in application and assessment of the algorithm presented here. This work is supported by a Grant-in-Aid for Fusion Research of the Ministry of Education, Science and Culture in Japan.

REFERENCES

1, K. V. ROBERTS AND D. E. POTTER, "Magnetohydrodynamic Calculations," in *Methods in Computational Physics*, Vol. 9 (Academic Press, New York/London, 1970), p. 340.

2. C. W. HIRST, A.A. AMSDEN, AND J. L. COOK, *J. Comput. Phys.* **14**, 227 (1974).

3. J. T. ODEN, *Finite Elements of Nonlinear Continua* (McGraw Hill, New York/London, 1972).

4. O. C. ZIENKIEWICZ, *The Finite Element Method*, 3rd ed. (McGraw–Hill, New York/London, 1977).

5. J. P. BORIS AND D. L. BOOK, *J. Comput. Phys.* **11**, 38 (1973).

6. S. T. ZALESK, *J. Comput. Phys.* **31**, 335 (1979).

7. G. A. SOD, *J. Comput. Phys.* **27**, 1 (1978).

8. A. JEFFERY AND T. TANIUTI, *Non-Linear Wave Propagation with its Application to Physics and Magnetohydrodynamics*, (Academic Press, New York/London/Tokyo, 1964).

9. P. D. LAX, *Commun. Pure Appl. Math.* **Z**, 159 (1954); **10**, 537 (1957).

10. C. R. RAO AND S. K. MITRA, *Generalized Inverse of Matrices and its Applications* (Wiley, London, 1971).

11. C. C. PAIGE AND M. A. SAUDERS, *ACM Trans. Math. Software* **8**, 43 (1982).

12. R. J. GELINAS AND S. K. DOSS, *J. Comput. Phys.* **40**, 202 (1981).

13. A. HARTEN AND J. M. HYMAN, *J. Comput. Phys.* **50**, 235 (1983).

14. M. J. FRITTS, W. P. CROWLEY, AND H. TREASE (Eds.), *Free-Lagrange Method*, Lecture Notes in Physics Vol. 238 (Springer-Verlag, Berlin/Heidelberg/New York/Tokyo, 1985).

15. I. KAWAKAMI, M. AIZAWA, K. HARADA, AND H. SAITO, *J. Phys. Soc. Japan* **54**, 544 (1985).

16. J. U. BRACKBILL AND W. E. PRACHT, *J. Comput. Phys.* **13**, 455 (1973).

17. M. J. FRITTS AND J. P. BORIS, *J. Comput. Phys.* **31**, 173 (1979).

18. B. W. KERNIGHAN AND P. J. PLAUGER, *The Elements of Programming Style* (Addision–Wesley, Reading, MA, 1976).