

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36.

TITLE MOVING FINITE ELEMENTS: A CONTINUOUSLY ADAPTIVE METHOD FOR COMPUTATIONAL FLUID DYNAMICS

AUTHOR(S) Alan H. Glasser

SUBMITTED TO Proceedings of Energy Research Power Supercomputer Users Symposium, Gaithersburg, Maryland, May 21-22, 1991

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes.

The Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy.

MASTER

Los Alamos Los Alamos National Laboratory Los Alamos, New Mexico 87545

# **Moving Finite Elements: A Continuously Adaptive Method for Computational Fluid Dynamics**

Alan H. Glasser, Los Alamos National Laboratory  
Keith Miller and Neil Carlson, University of California at Berkeley

## **Introduction**

Moving Finite Elements (MFE),<sup>1-5</sup> a recently developed method for computational fluid dynamics, promises major advances in the ability of computers to model the complex behavior of liquids, gases, and plasmas. Applications of computational fluid dynamics occur in a wide range of scientifically and technologically important fields. Examples include meteorology, oceanography, global climate modeling, magnetic and inertial fusion energy research, semiconductor fabrication, biophysics, automobile and aircraft design, industrial fluid processing, chemical engineering, and combustion research. The improvements made possible by the new method could thus have substantial economic impact.

Numerical methods for treating such problems have been studied for more than forty years, starting primarily with the work of John von Neumann and coinciding with his development of the programmable digital computer in the early 1950s. In spite of these efforts, there are important features of fluid flow which remain difficult or impossible for most numerical methods. MFE provides a new approach for handling these challenges.

Modeling a fluid on a digital computer requires approximating the smooth variation of fluid quantities, such as density, velocity, temperature, and electromagnetic fields, by a large number of discrete values. The most common approach is to represent these quantities by their values on a regular mesh or grid covering the spatial domain of interest. The behavior of the fluid at successive moments of time is calculated by converting the known fluid equations to a large system of algebraic equations which can be solved on the computer. The finer the mesh and the shorter the time steps, the more accurate the solution becomes, at the expense of greater demands on computer time and storage capacity. In fact, the fastest and largest supercomputers have been developed mainly to allow accurate solution of such complex fluid problems.

Some of the most challenging fluid problems are those characterized by small regions of rapid variation which must be accurately resolved to get the right behavior. Examples are shock waves, burn fronts, and boundary layers. For a method which uses a fixed, uniform grid, it may require such fine grid spacing and short time steps to resolve the shock that solution is impractical even on the most powerful supercomputers.

Moving Finite Elements is a moving node adaptive grid method which has a tendency to pack the grid finely in regions where it is most needed at each time and to leave it coarse elsewhere. It does so in a manner which is simple and automatic, and does not require a large amount of human ingenuity to apply it to each particular problem. At the same time, it often allows the time step to be large enough to advance a moving shock by many shock thicknesses in a single time step, moving the grid smoothly with the solution and minimizing the number of time steps required for the whole problem. For 2D problems (two spatial variables) the grid is composed of irregularly shaped and irregularly connected triangles which are very flexible in their ability to adapt to the evolving solution. While other adaptive grid methods have been developed which share some of these desirable properties, this is the only method which combines them all. In many cases, the method can save orders of magnitude of computing time, equivalent to several generations of advancing computer hardware.

While the MFE method has been tested on a variety of 2D problems and shown to behave as desired, several essential improvements are still required before it is ready for routine application. The code currently uses a simple direct method for solving large, sparse linear systems of algebraic equations, which would be inefficient and impractical for large problems. Work is in progress to replace this technique with a more efficient iterative solution procedure. This should also make the code adaptable to efficient use on massively parallel computers, such as the Connection Machine and the Intel Hypercube. It is desirable in certain cases to allow the grid to break and reconnect infrequently in order to permit a better fit to the evolving solution. Such a procedure has been tested on a few simple, special-purpose applications, but has yet to be incorporated into the general-purpose code. Once these improvements have been made, it should be straightforward to extend the code to 3D problems, using a grid of tetrahedra instead of triangles.

The range of applications of the MFE method is both somewhat broader and somewhat narrower than computational fluid dynamics. In addition to fluid problems, the method can also be applied to studying the behavior of evolving manifolds and films, such as soap films. This is made possible by the fact that the independent spatial variables and the dependent variables are treated on an equal basis, allowing the method to follow the behavior of quantities which are not single-valued functions of the independent variables.

There are fluid problems for which MFE is not suitable. The clearest cases are systems with widespread strong turbulence. The strong point of the MFE method is its ability to resolve sharp, smoothly-moving fronts by automatically concentrating the grid nodes in the front and moving them with the front. In turbulent systems, however, the motion is quite chaotic and nonsmooth, and a fine grid is often required essentially everywhere. Such problems are better handled by methods which require less computation per grid point and per time step. Problems in which turbulence is modeled by empirical enhanced transport coefficients might nevertheless be fair game for the method.

Visualization of the complex results of our computations is treated with the help of the state-of-the-art graphics hardware and software provided by the Silicon Graphics Iris Workstations. Computer-generated movies with hidden surfaces, real-time animation, nearly continuous variation of color, and flexible user interaction have been developed to provide maximum comprehension.

The remainder of this paper is devoted to a mathematical formulation of the method and an examples of results.

### Mathematical Formulation

We treat a general system of fluid equations of the form

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F} = \mathbf{S}, \quad (1)$$

where  $t$  is the time,  $\mathbf{x}$  is a vector of independent spatial variables,  $\mathbf{u}$  is an  $n$ -vector of unknowns. For each component of  $\mathbf{u}$  there is a flux  $\mathbf{F}$  and a source  $\mathbf{S}$  with the general form

$$\mathbf{F} = \mathbf{C}(t, \mathbf{x}, \mathbf{u}) - \mathbf{D}(t, \mathbf{x}, \mathbf{u}) \cdot \nabla \mathbf{u}, \quad \mathbf{S} = \mathbf{S}(t, \mathbf{x}, \mathbf{u}, \nabla \mathbf{u}), \quad (2)$$

where  $\mathbf{C}$  represents convective flux and  $\mathbf{D}$  is a diffusion tensor, and all functions may depend in an arbitrary nonlinear manner on their arguments. A variational for Eq. (1) is obtained by noting that

$$L = \frac{1}{2} \int \left[ \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F} - \mathbf{S} \right]^2 w(t, \mathbf{x}, \mathbf{u}, \nabla \mathbf{u}) d\mathbf{x}, \quad (3)$$

with  $w$  an arbitrary weight function, is variational in the sense that

$$\frac{\delta L}{\delta(\partial \mathbf{u}/\partial t)} = 0 \quad (4)$$

recovers Eq. (1).

The general class of conventional Galerkin methods is obtained from Eqs. (3) and (4) by expanding  $\mathbf{u}$  in a set of basis functions,

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{u}_i(t)\alpha_i(\mathbf{x}), \quad (5)$$

where the  $\mathbf{u}_i(t)$  are time-dependent amplitudes and the  $\alpha_i(\mathbf{x})$  are the spatially-varying basis functions, which may, for example, be Fourier series, orthogonal polynomials, or conventional fixed finite elements. Since the time derivative of Eq. (5) contains only the amplitude variations,

$$\dot{\mathbf{u}}(\mathbf{x}, t) = \dot{\mathbf{u}}_i(t)\alpha_i(\mathbf{x}), \quad (6)$$

the discretized equations of the method are obtained from

$$\frac{\delta L}{\delta \dot{\mathbf{u}}_i} = 0, \quad (7)$$

which yields a coupled system of ordinary differential equations,

$$(\alpha_i, \alpha_j)\dot{\mathbf{u}}_j = (\alpha_i, \mathbf{g}), \quad (8)$$

where  $\mathbf{g} = \mathbf{S} - \nabla \cdot \mathbf{F}$ .

The key issue for any adaptive grid method is how to move the grid. In Moving Finite Elements, this issue is resolved by treating the grid positions in exactly the same manner as the amplitudes, *i.e.* as variational parameters. For basis functions we choose linear finite elements on a grid of irregularly shaped and connected triangles in 2D, tetrahedra in 3D, and simplices in general. Because the grid is allowed to move, we replace Eqs. (5) and (6) by

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{u}_i(t)\alpha_i[\mathbf{x}, \mathbf{s}_j(t)], \quad (9)$$

$$\dot{\mathbf{u}}(\mathbf{x}, t) = \dot{\mathbf{u}}_i(t)\alpha_i(\mathbf{x}, t) + \dot{\mathbf{s}}_i(t)\beta_i(\mathbf{x}, t), \quad (10)$$

where  $\beta_i$  is defined as the coefficient of  $\dot{\mathbf{s}}_i$  obtained by differentiating Eq. (9). Then Eqs. (7) and (8) are generalized to

$$\frac{\delta L}{\delta \dot{\mathbf{u}}_i} = \frac{\delta L}{\delta \dot{\mathbf{s}}_i} = 0, \quad (11)$$

$$(\alpha_i, \alpha_j)\dot{\mathbf{u}}_j + (\alpha_i, \beta_j)\dot{\mathbf{s}}_j = (\alpha_i, \mathbf{g}), \quad (12)$$

$$(\beta_i, \alpha_j)\dot{\mathbf{u}}_j + (\beta_i, \beta_j)\dot{\mathbf{s}}_j = (\beta_i, \mathbf{g}). \quad (13)$$

Just as Eq. (8) determines the amplitudes by minimizing the positive-definite variational over the space of amplitudes, Eqs. (12) and (13) determine the amplitudes and node positions by minimizing

over this larger space, and thus obtain a better minimum. This causes the nodes to move where they are needed to resolve the solution.

While the basic idea is simple, there are computational details which are essential to make the method work correctly. In regions where the solution is flat, the prescription for moving the nodes becomes indeterminate because a range of different node motions give equally "best" fits. This is manifested mathematically in the vanishing of the determinant of the mass matrix in Eqs. (11) and (12), and is resolved by adding regularization terms to  $L$  which can be interpreted as internodal viscosity and grid tension. The weight function in Eq. (3) is chosen to be  $w = [1 + (\nabla u)^2]^{-1/2}$ , which converts the integral over the domain into an integral over the area of the solution manifold, resulting in much better node motion and in placing the independent spatial variables  $\mathbf{x}$  and the dependent variables  $\mathbf{u}$  on more of an equal basis. The time step must be implicit in order to exceed the Courant condition and use efficiently large step sizes. This requires the solution of large, sparse linear systems, which is currently done with a direct band solution but will shortly be replaced with a more efficient iterative method using buffered relaxation as a preconditioner for a nonlinear Krlov subspace method.

### Illustration: Shock Formation

We conclude with an illustration of the ability of the MFE method to efficiently treat the development of a very thin moving shock front. We study the 2D Burgers' equation,

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = D \nabla^2 \mathbf{u}, \quad (14)$$

Here  $\mathbf{u}$  represents a velocity which is nonlinearly convected in its own direction. If the initial value of  $\mathbf{u}$  is positive on the left and negative on the right, the solution convects toward the middle and forms a shock. The final thickness of the shock is determined by the diffusion coefficient  $D$ . In the example shown below, the diffusion coefficient is  $10^{-5}$ , and so the shock width is about 1/1000th the width of the domain. For a method using a fixed grid, accurate resolution of this shock front would require a grid size of several thousand on each side and a similarly small time step. The solution shown below uses a grid of  $13 \times 15$  and required about 1.5 minutes of cpu time on an IBM RS/6000 Model 320 workstation. The pictures represent four equally spaced times during the run. The long straight lines denote  $x$ ,  $y$ , and  $z$  axes. These are examples of the Silicon Graphics 3D animated graphics developed for this code, with colors replaced by patterns.

### References

1. K. Miller and R. N. Miller, "Moving Finite Elements I," *SIAM J. Num. Anal.* **18**, 6, 1019 (1981).
2. K. Miller, "Moving Finite Elements II," *SIAM J. Num. Anal.* **18**, 6, 1033 (1981).
3. K. Miller, "Alternative Modes to Control the Nodes in the Moving Finite Element Method," in: "Adaptive Computational Methods for PDE's," ed. Babuska, Chandra, and Flaherty, Workshop held in College Park, MD, February 14-16, 1983, SIAM, 1983.
4. Neil Carlson and Keith Miller, "Gradient Weighted Moving Finite Elements in Two Dimensions," in *Proceedings of the ICASE/NASA Workshop on Theory and Application of Finite Elements*, July 28-30, 1986, Hampton, Virginia, published by Springer-Verlag.
5. Alan H. Glasser, "A Moving Finite Element Model of the High Density Z Pinch," *J. Comp. Phys.* **85**, 159 (1989).

### Illustration of Moving Finite Elements

Four equally spaced time steps illustrating the formation of a shock in the solution of the 2D Burgers' equation. Triangular cells of the moving grid is shown in these perspective views, with straight lines denoting x, y, and z axes.

