Adaptive Mesh Techniques for Unsteady Multi-Gradient Problems in Computational Fluid Dynamics

Timothy James Bartel

January 1987

UWFDM-720

Ph.D. thesis.

FUSION TECHNOLOGY INSTITUTE

UNIVERSITY OF WISCONSIN

MADISON WISCONSIN

Adaptive Mesh Techniques for Unsteady Multi-Gradient Problems in Computational Fluid Dynamics

Timothy James Bartel

Fusion Technology Institute
University of Wisconsin
1500 Engineering Drive
Madison, WI 53706

http://fti.neep.wisc.edu

January 1987

UWFDM-720

Ph.D. thesis.

ADAPTIVE MESH TECHNIQUES FOR UNSTEADY MULTI-GRADIENT PROBLEMS IN COMPUTATIONAL FLUID DYNAMICS

by

TIMOTHY JAMES BARTEL

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

(Nuclear Engineering)

at the

UNIVERSITY OF WISCONSIN-MADISON

1987

A C K N O W L E D G E M E N T S

I would like to express my appreciation to Dr. Gregory A. Moses for his advice and supervision during this investigation. The availability of unlimited supercomputer time by him was an invaluable help to my work. The assistance of Dr. Robert Peterson with the mysteries of his equation-of-state package is also acknowledged. I greatly benefited from the numerous discussions we had.

I would also like to thank Messrs. James Barry and Faustino Gonzalez for the many stimulating discussions of both a technical nature and of life in general. The shock of returning to graduate school was eased by their friendship.

Financial support for this work was provided by Sandia National Laboratory and Lawrence Livermore National Laboratory.

T. J. B.

The University of Wisconsin-Madison

13 January 1987

A B S T R A C T

An adaptive grid algorithm has been developed for the purpose of simulating highly nonlinear and unsteady 1-D problems in computational fluid dynamics with high gradient regions in one or more physical quantities. The equidistribution principle was used to determine the updated mesh location. Smoothing of the distribution and minimum mesh size control was included in the algorithm. The method was tested against a simple propagating shock, a shock tube, an interacting blast wave, and a radiation hydrodynamic blast wave. The transformed governing equations for these problems were cast in a strong conservative form; a control volume formulation using first upwind differencing was used to develop the difference equations. This simple numerical method coupled with an adaptive mesh resolved high gradient regions very well but difficulties occurred with the simulation of both the rarefaction fan and a shock region.

TABLE OF CONTENTS

L I S T O F T A B L E S

L I S T O F F I G U R E S

L I S T O F S Y M B O L S

c1,c2,c3	smoothing constants
C	Constant wave speed, viscosity model coefficent
e	energy per unit volume
J	Jacobian
N	number of mesh points
q	flux-limited diffusion term
P	pressure
u	velocity
U	sample variable
R	percentage of grid points assigned to a derivative
S	Arc length
T	temperature
W	Weight function
x,t	physical coordinate system variables
ξ, τ	transformed coordinate system variables
α	weight for first derivative
β	weight for second derivative
ρ	density
ϵ	internal energy
Ω	net energy exchange between fluid and radiation field
μ	viscosity

L I S T O F S Y B M O L S (cont)

superscripts

n, n+1	time step levels

subscripts

x, t, ξ, τ	partial derivatives
a, b	weight function variable
c	cell centered quantity
e	cell edged quantity
i	mesh location

1.0 INTRODUCTION

The application of finite difference numerical methods to systems of partial differential equations has reached a relatively high state of development in the past twenty years. Significant advances have been made in computer technology, numerical methods, and grid generation during this time. Direct numerical simulation, that is obtaining information about physical phenomena which are difficult or impractical to measure experimentally, is a common place practice. However, the complexity of the problems to be solved has also grown. The computer having an infinite amount of memory with an infinitely fast cycle time has yet to be built. Therefore the investigator, with limited computer resources, is constantly faced with the practical problem of how many grid points to use and where to place them to achieve both accuracy and reasonable execution times. Thus, interest has developed in the past few years in developing techniques for constructing solution adaptive mesh systems; that is, mesh systems where the grid evolves as part of the solution to the system of differential equations. Since the mesh is adaptive, the grid point positions are determined by some measure of the quality of the numerical solution. The

computed results are in some sense "better" than those obtained on a uniform grid with the same number of points (Lagrangian mesh systems will be discussed latter).

Adaptive mesh generation essentially addresses two, sometimes conflicting, concerns: accuracy of the solution and minimization of the number of mesh points. The reason for the latter is obvious, as most matrix inversion techniques scale directly by some power of the number of points. The accuracy issue arises from the fact that the more challenging simulation problems are unsteady in time and have regions of high gradients of one or more variables. Typical numerical methods are second order accurate in the space variables. One would like to optimize the mesh point spacing at these high gradient regions in order to accurately resolve them. Adaptive mesh construction attempts to resolve this dilemma by clustering the grid points where needed to resolve certain features of the solution and also allow them to be displaced in the space domain as the solution develops in time.

1.1 Model Problem

Problems with high gradient regions benefit the most from adaptive mesh systems; many examples exist in numerous applications. The present investigation is concerned with the class of problems containing multiple

high gradient regions resulting from different quantities (i.e., density, temperature, pressure) which one would like to accurately resolve. Typical examples include spark plug ignition, high temperature blast waves[1], and laser or ion beam ablation of a surface[2]. The first situation has a pressure wave (shock) and a deflagration wave which need to be resolved. The second has a blast and thermal wave; the front needs to be finely meshed to obtain the correct propagation speed for this precursor shock problem[3]. The third has coupled density and temperature gradients.

The model problem for the present work is the laser/ion beam ablation of a surface. It is further narrowed to consider situations typical for inertial confinement fusion (ICF)[4,5]. In this application, a high intensity beam is used to ablatively implode a target to achieve extremely high densities and therefore obtain a temperature sufficiently high to have a thermonuclear burn. The incident beam is attenuated in the "corona" region (low density plasma) of the ablated debris. The beam penetrates the corona until the point where the plasma frequency equals the beam frequency--defined as the critical density. Then thermal conduction transports the deposited energy from this point to the target surface where ablation takes place. Thus this problem possesses

two high gradient regions of importantance: the critical density point (laser beam energy) and the target ablation front (density). This is an unsteady problem, as both fronts propagate away from the surface as the solution progresses. The efficiency of this process is determined by the ablation rate; both gradient regions need to be accurately resolved to determine it. Radiation hydrodynamic effects are also important in this problem. Thus the laser/ion beam ablation problem is a good one with which to use an adaptive grid scheme.

Lagrangian grid schemes, developed from radiation hydrodynamic simulations, have historically been used to model this problem. In these methods the mass of each computational cell is a constant and the cell propagates at the fluid velocity (no mass flux across the cell face). This technique adequately resolves the ablation gradient. However, since the density decreases rapidly from the surface, the meshes will be elongated in the region of the critical density (typically 50 times larger). This will tend to smear the temperature gradient in space and thus will reduce the peak temperature. This appears not to be a problem for long wavelength laser light, but becomes one for shorter wavelenghts as the critical surface is closer to the ablation surface[6]. Figure 1, from work done at KMS[7], illustrates this problem. Figure 1.a was obtained

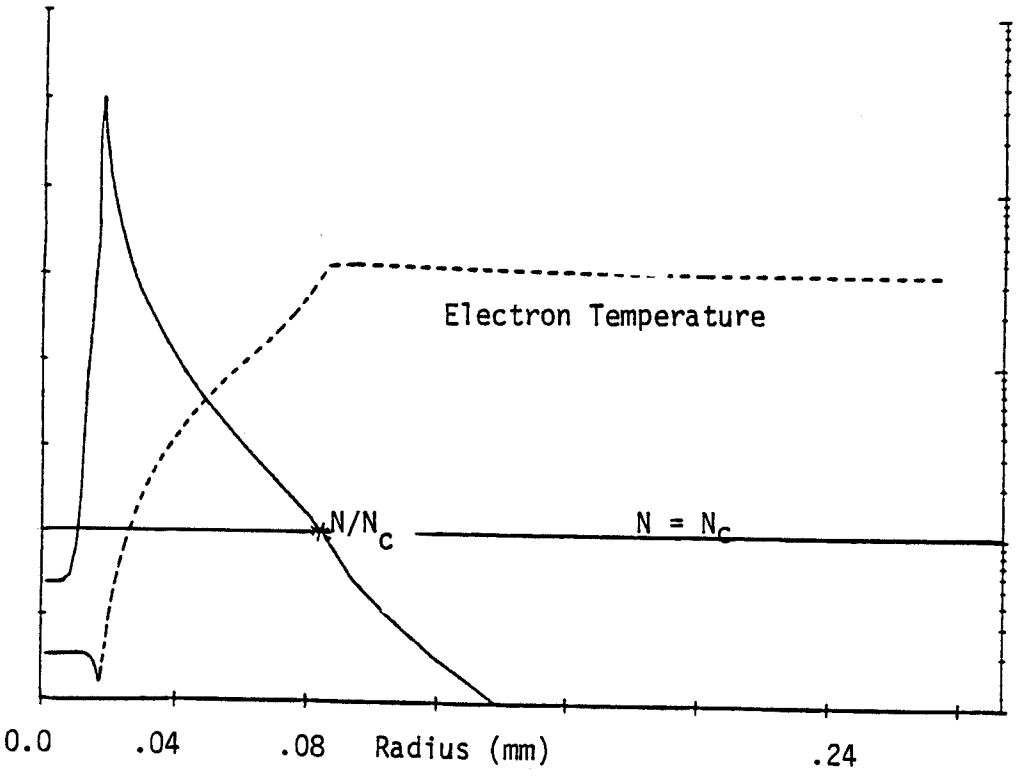

Figure 1.a Simulation with lagrangian mesh

using a standard lagrangian grid scheme while figure 1.b used fixed subgridding or mesh refinement near the critical density surface to resolve this gradient. One will note the reduced temperature and the effect of the ponderomotive force on the density profile for the case with better resolution. This local mesh refinement scheme required the user to have *a priori* knowledge about the form of the solution; it is not a very robust technique.

Two dimensional modelling of the model problem with a lagrangian mesh is very difficult. Emery[8] gives numerical simulations using an eulerian mesh; the flow field contains vorticies generated from Rayleigh-Taylor instabilities. A lagrangian mesh has severe difficulties in simulating rotational fluid flow.

1.2 Adaptive Grids

A numerical simulation of a problem requires three distinct parts: stating the governing equations, choosing a method to solve the discretized equations, and finally determining the placement of the observer points (mesh points). As was just illustrated, the grid can play a very important role in determining the accuracy (quality) of the solution even though the same governing equations and numerical methods were used. Thus a numerical method might have a high formal accuracy (with respect to a

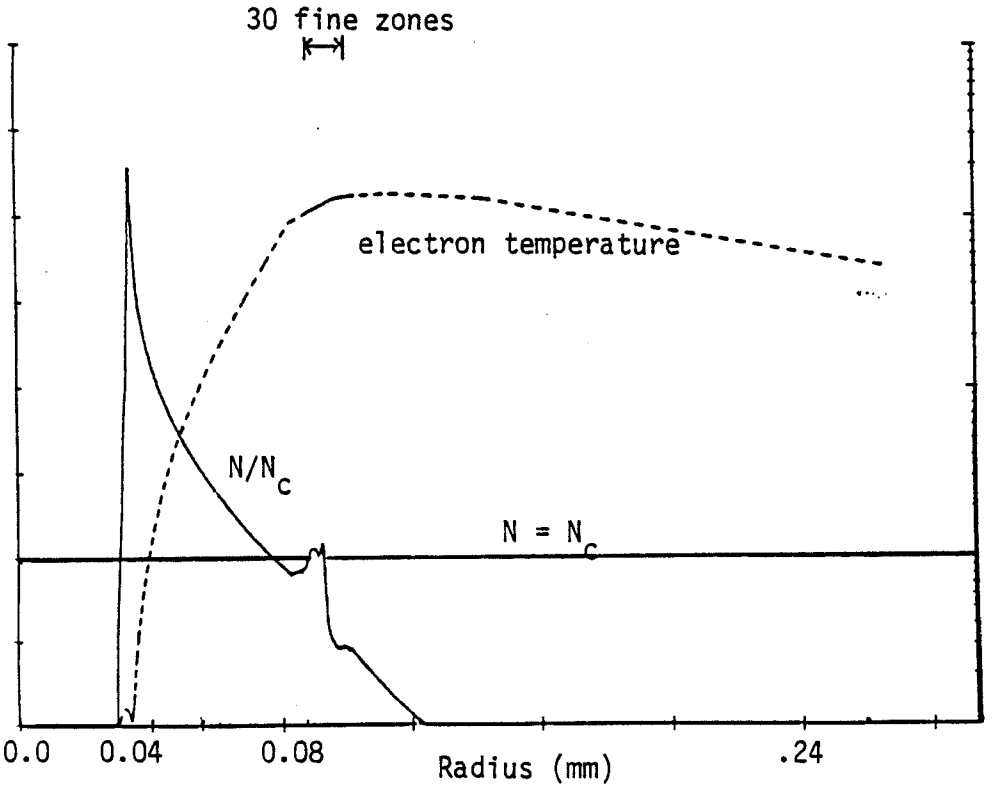

Figure 1.b Simulation with lagrangian mesh and local mesh refinement.

Taylor series expansion), but the actual solution might be meaningless. The computational grid is just as important as the other two parts of the problem. A good grid scheme, however, will not always overcome deficiencies in the numerical method. It is important that all three phases of the problem be coordinated so that the results of the numerical simulation are meaningful.

Finite difference methods for the solution of partial differential equations originally used either eulerian or lagrangian coordinate systems. An eulerian system (inertial or laboratory frame system) is one in which the grid points, observers of the solution, remain fixed in some space coordinate system. Fluid passes freely through the computational meshes. This is a natural grid system for matching experimental data since transducers usually remain stationary. Grid points could be placed at these probe points and the numerical results could be compared directly with the experimental data. In a lagrangian system, the grid points move with the local velocity of the fluid. Each computational cell can be thought of as having a given quantity of material which remains fixed throughout the simulation. Therefore, unlike the eulerian grid which remains fixed in time, the lagrangian grid undergoes constant deformation.

Each of these systems has inherent advantages and disadvantages, dependent upon the physical problem to be solved. Table 1 compares these for both systems. The main strength of the eulerian mesh system is that it is robust; it can model many complex problems given a sufficient number of mesh points. The lagrangian mesh is perfect for strong shocks and blast waves; the high pressure gradients pose no difficulty. Since the present model problem has multiple gradient regions, a mesh having attributes of both systems would be desired.

Eulerian and lagrangian mesh systems can be viewed as special cases of the general set of adaptive systems. This relationship can best be illustrated by the following problem. We consider the first-order wave equation of

$$U_t + C\,U_x = 0 \qquad (1.1)$$

where $U(t,x)$ is the unknown dependent variable, U_t and U_x are the partial derivatives with respect to time and space respectively, and C is a constant wave speed. We let the transformation relating the physical domain (x,t) and the computational domain (ξ,τ) be written as

$$\tau = t \qquad (1.2)$$

$$\xi = \xi(x,t) \ .$$

Table 1

Eulerian and Lagrangian Grid Systems

	strengths	weakness
EULERIAN	- fixed points in space for data comparison - complex geometries - easy programming - time asymptotic and transient analysis - rotational flows - instability growth - robust	- smear strong discontinuities (blast waves) - dissipation - multi-material transport
LAGRANGIAN	- multi-material transport - blast wave modelling - moving front problems (velocity driven) - transient analysis only	- multi-dimensional grid distortion - rezone schemes required - interpolation required for mesh skewing

Transforming the wave equation into computational coordinates yields

$$U_\tau + \left(\frac{C - x_\tau}{x_\xi}\right) U_\xi = 0 \qquad (1.3)$$

where x_ξ is the grid metric (Jacobian) and x_τ is the grid speed. The coordinate system transformations can be found in Appendix A. We now examine this equation to determine the eulerian and lagrangian systems subsets. First, if the grid speed is set to zero, the result is simply an eulerian mesh in a stretched coordinate system. It is interesting to note that the wave speed is now C/x_ξ and is thus mesh dependent. If the grid speed is set to the wave speed ($x_\tau = C$), the result is a lagrangian mesh system. And in the general adaptive case, the effective wave speed is both space (mesh) and time dependent in the computational domain.

In conclusion, it has been demonstrated that eulerian and lagrangian mesh systems are only special cases of the adaptive grid system. Local grid refinement and periodic rezoning are only *ad hoc* patches to either the eulerian or lagrangian systems to enable them to solve more complicated problems. Adaptive grid systems provide a systematic approach to determine the 'best' time dependent mesh for a problem. Thus, the direct numerical simulation

of complex problems can be attempted where the lack of information about the solution prevents *a priori* grid modeling.

1.3 Problem Statement

The present investigation is concerned with developing an efficient numerical technique to simulate one-dimensional unsteady problems with sharp gradients. The first goal of this work is to develop a simple procedure for generating an adaptive mesh and to then couple the translating grid points to the system of governing partial differential equations. The Navier-Stokes equations will be used instead of the simpler nonlinear Burger's equation based on the experience of Lax[9]: procedures which work for the simple equation might not work for the full set of equations. The equations will be cast in a strong conservative form[10]. Computational efficiency and robustness are prime attributes which are desired for the method; code vectorization will be achieved whenever possible.

The second goal is to evaluate the method with a set of simple problems which capture the salient features of the model problem; the present work is limited to developing and demonstrating the method. The intent of the present investigation is to develop a methodology for simulating a variety of problems rather than just one

particular application. Four test problems have been chosen for this task:

1--Inviscid shock wave,

2--Shock tube,

3--Interacting blast waves, and

4--Radiation hydrodynamic blast wave.

The first three problems model a single ideal gas while the last one models a high temperature absorbing and emitting real gas with the two temperature diffusion approximation.

2.0 LITERATURE REVIEW

The problem under investigation is one of numerical simulation. As was stated in the introduction, there are three facets to the problem: governing equations, mesh generation, and numerical methods. The present literature review will be presented along those general lines: model problem, adaptive mesh systems, and numerical methods for systems of partial differential equations.

2.1 Model Problem Simulation

As was mentioned earlier, lagrangian mesh systems have been historically used to simulate the present model problem. The LASNEX code from LLNL[11] and PHD-IV from Wisconsin[12] are just two typical examples. Both use standard lagrangian hydrodynamics to advance the flow solution in time. Neither employ any automatic mesh subgridding or rezoning based on the solution gradients. That is, the mesh point movement is determined solely by the fluid velocity irrespective of other structures in the solution. They also both have the same modelling difficulties as was experienced with the KMS code, discussed earlier[7], due to the lack of mesh resolution at the critical density plane.

NRL has developed an eulerian based radiation hydrodynamics code, CASTOR[13], using the flux corrected transport (FCT)[14] algorithm to overcome some of the difficulties of the eulerian mesh for blast wave problems(see Table 1). The FCT method will be discussed in a later section. They have since developed a new version with an adaptive mesh or 'sliding eulerian zones' as they refer to it (FAST2D[15]). They have divided the spatial domain into several regions based on the physical ablation processes, with each region having its own adaptive grid. A diffusion equation was used to control the mesh point movement. This is a highly problem dependent procedure and appears to not be very robust.

The NRL group has reported differences in modelling the ablation of thin targets using short wavelength light when compared with LASNEX[15]. It appears that LASNEX solutions approached the FAST2D results only when many lagrangian mesh points were used. A major penalty was paid in computational time due to both the reduced mesh size in the solid region and the increased number of mesh points. Recall that as the foil ablates, the mesh will naturally elongate due to the density reduction. This will smear the otherwise sharp critical density plane.

As mentioned earlier, KMS has developed a lagrangian code with a predetermined local subgridding[7]. This al-

lowed them to capture the detailed features of the critical density region. Unfortunately, the process requires *a priori* knowledge of where this region will be located. A true adaptive method uses the current time solution to determine the mesh point locations.

Combined eulerian and lagrangian schemes have also been used[16,17]. The strategy of these codes is to first establish a fixed mesh (eulerian) and then integrate the governing partial differential equations in a lagrangian sense. At the completion of each time step, the results are interpolated back to the original mesh locations. An intelligent choice for the original grid must be made to capture localized gradients.

2.2 Adaptive Mesh Systems

In a recent paper, Thompson[18] reviewed the current work in adaptive grid generation. Only the most important ideas of adaptive griding and their references will be summarized here.

The majority of the current work has been focused on obtaining a grid system for a steady-state problem. That is, the initial mesh point distribution has been perturbed as the solution achieves a steady-state nature; the local mesh velocity will tend to zero at this time.

First, we recall the one-dimensional first-order wave equation in computational coordinates developed earlier

$$U_\tau + \left(\frac{C - x_\tau}{x_\xi}\right)U_\xi = 0 \qquad (2.1)$$

The metric, x_ξ, represents the ratio of arc lengths in the physical and computational planes and the grid speed, x_τ, provides the dynamic coupling of the moving grid with the evolving solution of the differential equation. Any method for constructing an adaptive grid must provide a technique for estimating these terms since they explicitly appear in the transformed equations. Exceptions to this are time asymptotic (steady-state) problems; the grid speed need not be tightly coupled to the solution. In fact, as the solution converges, the grid speed will naturally go to zero. Since the problems under consideration are unsteady in time, this will not be a consideration.

Adaptive grid methods can be divided into two categories. In the first one, some set of rules relating the grid points in the physical and computational domain is used to establish new physical grid point locations at the end of each time step based on the physical variables. The grid speed is then simply estimated for the next integration by using a backward difference on the

new and old grid locations. Either implicitly coupling the determination of the metric and the solution of the PDEs, or using an explicit procedure, has been tried. The second class of schemes relies upon directly establishing the grid speed by some rule. Then the grid speed is integrated along with the differential equation and the new grid point positions are established; the metrics are computed by evaluating the ratio of arc lengths in the physical and computational domains. A lagrangian scheme would be of the second class as the fluid velocity is the mesh velocity; typically no stretching is done and the problem is solved in the physical domain.

There are advantages and disadvantages to both approaches. Methods which directly generate the new coordinates (class 1) are conceptually easy to apply and grid clustering is generally easily controlled. The use of a backward time difference to determine the grid speed is only first order accurate. A major limitation of these grid point location schemes is that they may be difficult to extend to multidimensional problems. Conversely, techniques which directly determine grid speed from some grid speed law (class 2) are easily applied in multidimensional applications because grid point location is determined by a simple integration. The major disadvantage with these techniques is in formulating the physical laws

which determine the grid speed. Also, mesh point control or clustering is a difficulty.

The philosophy of Anderson[19] is appropriate here: "Ideas used in construction of adaptive grid techniques are limited only by one's imagination, and any scheme that works in the sense of providing a better solution is a good one".

2.2.1 Equidistribution Schemes Thompson[20] states that numerous studies on ODEs have shown that the solution error can be reduced by distributing the grid points so that some positive weight function, $W(x)$, is equally distributed over the field, i.e.,

$$\Delta x_i \, W_i = \text{constant}. \tag{2.2}$$

This same strategy can be applied to the numerical solution of partial differential equations. Now recall that a grid transformation is being used which maps x to ξ. And since it is convenient to define the grid points in the ξ plane by successive integer values of ξ ($\Delta\xi = 1$ and $\xi_{max} = N$, the number of grid points), then $\Delta x = x_\xi \Delta\xi = x_\xi$. Thus, the equidistribution statement becomes

$$x_\xi \, W = \text{constant}. \tag{2.3}$$

Now this weight function W can be taken as either a function of ξ or x. If it is a function of ξ, then eq. 2.3 is the Euler equation[21] for the minimization of the integral

$$I = \int_0^N W(\xi) x_\xi^2 d\xi \tag{2.4}$$

And if W is taken as a function of x, the corresponding integral is

$$I = \int_0^N [W(x) x_\xi]^2 \, d\xi \tag{2.5}$$

Equation 2.4 can be interpreted as representing the energy of a system of springs, with spring constants $W(\xi)$, spanning each grid interval. The grid point distribution resulting from the equidistribution principle represents the equilibrium state of the spring system. The variational problem for equation 2.5 is the least-squares minimization over the grid of the cumulative grid point spacing weighted by the weight function. Thus this approach obtains a smooth grid point distribution.

The weight function taken as a function of x is somewhat easier to implement since it is easier to associate W with a physical property gradient. The first approach, with W as a function of ξ, is more useful if a uniform value of $W\Delta x$ is fixed; for example, if the weight function

is taken to be representative of truncation error and limits are imposed on this error.

Now eq. 2.3 can be integrated to obtain

$$\xi(x) = N \frac{\int_0^x W dx'}{\int_0^L W dx} \qquad (2.6)$$

where the physical domain is taken to be from 0 to L and the computational domain contains N meshes. Rearranging equation 2.6 slightly, we obtain

$$\int_0^x W dx' = \xi(x) \left(\frac{\int_0^L W dx}{N} \right) . \qquad (2.7)$$

An explicit adaptive grid generation technique can be developed from this equation. At the beginning of each time step, the weight function can be determined from the physical variables. The second quantity on the right hand side of the equation is simply the average $\Delta x W$ for each mesh: the constant in equation 2.3. The first term, $\xi(x)$, is just the set of integers since $\Delta\xi$ has been previously defined as unity. Thus, all that is required is to perform the integral on the left hand side for the new value of x at time n+1. The grid metric, x_ξ, is then just $(x^{n+1} - x^n)$ and the grid speed is determined from a simple backward difference on the x location. These were the two

(This page is intentionally blank.)

variables which were added to the set of problem unknowns when the PDEs were transformed to the computational space. Now the PDEs can be integrated for the next time step physical variables, the new weight functions determined, and the process repeated until the end time is achieved.

The strategy of explicit adaptive grid construction has now been reduced to determining the appropriate weight functions W(x). Various candidate functions will be given in the following section.

2.2.2 Weight Functions The majority of the published literature have considered a single dependent variable, U.

The effect of the weight function is to reduce the point spacing x where W is large and increase it when W is small; therefore, the weight function should be some measure of the solution variation. The simplest choice would be the solution gradient

$$W = U_x. \tag{2.8}$$

Equation 2.3 then reduces to

$$U_\xi = \text{constant}. \tag{2.9}$$

The grid point distribution would then be adjusted so that the same change in the solution occurs over each grid interval. This is illustrated in fig. 2.1 (taken from ref. 20). This choice for the weight function has the

disadvantage of making the spacing infinitely large where the solution is constant.

A similar function is

$$W = \sqrt{1 + U_x^2} \tag{2.10}$$

This is an arc length weighting function. Now eq. 2.3 reduces to

$$S_\xi = \text{constant} \tag{2.11}$$

with S defined as the arc length. The grid point distribution is now such that the same increment in arc length on the solution curve occurs over each grid interval. Figure 2.2 (taken from ref. 20) illustrates this feature. Unlike the previous choice, this weight function yields uniform spacing where the solution is constant. But, the point concentration in the high gradient region is now reduced. A possible modification of eq. 2.10 to increase this grid point clustering could be

$$W = \sqrt{1 + \alpha^2 U_x^2} \tag{2.12}$$

where α controls the relative weighting of the gradient.

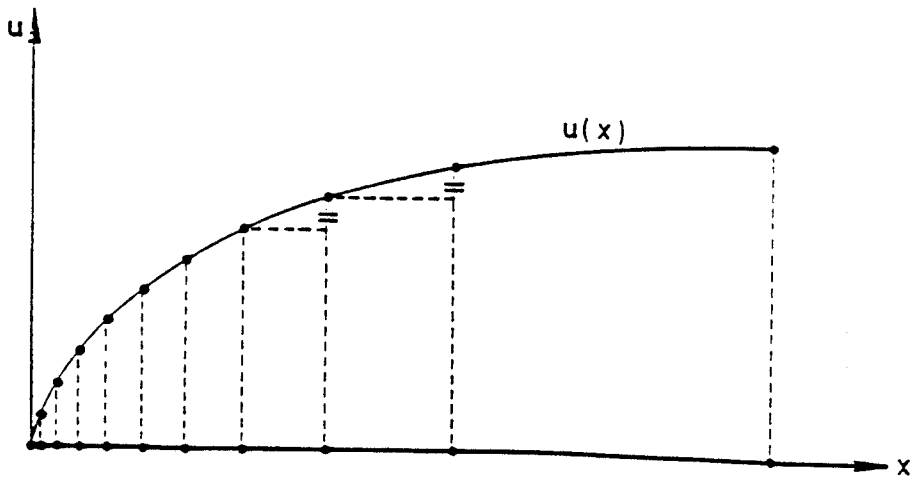

Figure 2.1 Solution gradient weighting.

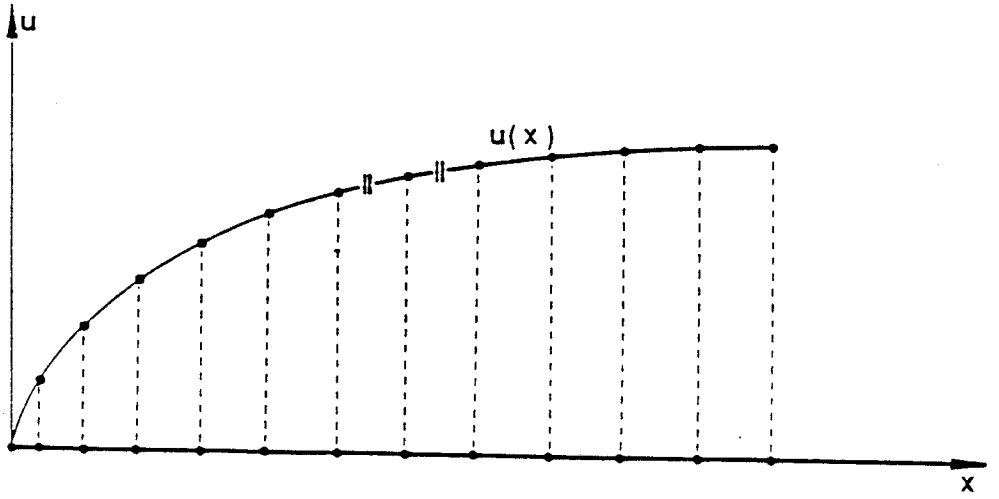

Figure 2.2 Arc length weighting

All of the weight functions mentioned have the disadvantage that regions near solution extrema, that is where $U_x = 0$, are treated similar to constant regions. Figure 2.3 (taken from ref. 20) illustrates this effect; there is insufficient point clustering at the solution extrema. Thus, the previously mentioned weight functions concentrate points only near gradients and not extrema.

Concentration of points near solution extrema could be achieved by incorporating some effect of the second derivative, U_{xx}, into the weight function. Thompson[20] suggested using the curvature of the solution curve:

$$K = \frac{U_{xx}}{(1 + U_x^2)^{3/2}} \tag{2.13}$$

He proposed a weight function of the form:

$$W = 1 + a^2|K| \ . \tag{2.14}$$

Eiseman[22,23] has proposed the following similar weight function:

$$W = (1 + \beta^2|K|)\sqrt{1 + \alpha^2 U_x^2} \tag{2.15}$$

Mesh point concentration near extrema is emphasized by large β and near high gradients by large values of α.

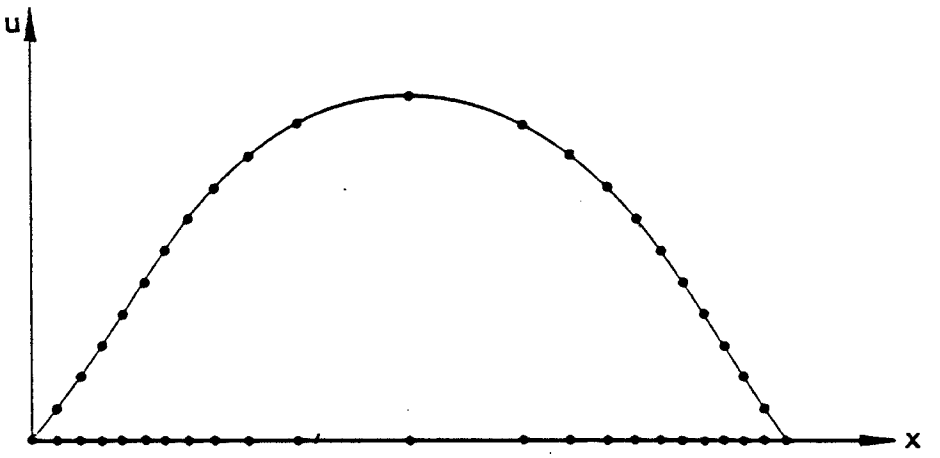

Figure 2.3 Solution extrema example

Dwyer[24-26] has had success with a weight function of the following simple form:

$$W = 1 + \alpha|U_x| + \beta|U_{xx}| \quad . \tag{2.16}$$

He has developed a strategy for determining α and β which essentially allows a specified fraction of points to be assigned to each function variation. If R_α is defined as the percentage of grid points to be assigned to the first derivative, then

$$R_\alpha = \frac{\alpha \int_0^{x_{max}} |U_x| \, dx}{\int_0^{x_{max}} \left(1 + \alpha|U_x| + \beta|U_{xx}|\right) dx} \tag{2.17}$$

A similar expression can be given for the second derivative control, R_β. The time dependent problem proceeds with R_α and R_β held constant and used to determine the solution varying α and β. Dwyer suggests performing this procedure with each time step to prevent point location oscillations due to rapid changes in the second derivative.

2.2.3 Grid Speed Schemes We recall from equation 2.1 that using adaptive grid schemes introduces two unknowns: the grid speed and grid metric. The previous section presented models for determining the grid metric; the grid speed would be determined from a simple backward

difference. Grid speed schemes do the opposite; the grid speed is modeled and the grid metric or points are determined by simple integration.

Anderson[27-29] has been the major contributor in developing grid speed models; his work has been primarily concerned with time asymptotic problems. An early model was based on obtaining a steady grid equation by differentiating the equidistribution equation 2.3 with respect to the computational space coordinate. The following grid speed equation was then postulated:

$$x_\tau = C \left[x_{\xi\xi} + \frac{x_\xi W_\xi}{W} \right] \qquad (2.18)$$

where C is a constant and W is some weight function. This choice forced the grid to have a zero speed when the steady grid equation was satisfied. In principle this approach was successful, but Anderson found that solving the parabolic equation to be computationally intensive in real applications. Also, this model was limited to time asymptotic problems.

Anderson has also proposed an 'attraction model'. The basis for this model is that decreased mesh spacing will reduce the solution error. The idea behind this model is similar to that of determining the net potential at a space location from a set of discrete point potentials.

This scheme also proved to be unsatisfactory for unsteady time problems. In his recent work, Anderson has shifted to either equidistribution methods or variational methods (discussed in section 2.5).

Lagrangian methods are of the class of grid speed schemes. The grid speed is simply taken as the local fluid velocity.

2.2.4 Other Methods Winkler, *et al.*[30,31] have developed an adaptive grid scheme for unsteady radiation hydrodynamic problems. They chose to proceed directly with the construction of differential operators to obtain their grid distribution, based on physical insight rather than on the formalism of the Euler-Lagrange equations and the equidistribution principle. Since they compute the mesh spacing directly, they had to consider mesh overlapping and point distribution smoothness problems. Also, it is not clear whether the PDEs were in conservative form and included the grid translation terms or whether interpolation has been used to obtain the physical variables at the new mesh locations. They employed a parabolic diffusion equation to transport the grid points at each new time step. The robust character of their method is questioned.

2.2.5 Multidimensional Methods Dwyer[24-26] has used his equidistribution scheme for two-dimensional simulations by adapting along only one axis. That is, only gradients along one grid coordinate are adaptively resolved. Dwyer believes that many 2-D problems can be modeled by this procedure if a judicious choice of the grid is made[32]. He has presented solutions for flow around cylinders where the solution is adapted along the radial axis. The angular axes distribution remains fixed in the problem. This resulted in a simple and efficient scheme which did not have the grid skewing problem which might have occurred if the adaption took place along both axes. To implement this method, the equations need only have been modified to consider the arc length, s, rather than the physical coordinate (x or y) for the derivatives and integration limits. For example, equation 2.7 would become

$$\int_0^s W ds' = \xi(s)\left(\frac{\int_0^L W ds}{N}\right) \qquad (2.19)$$

where it has been assumed that adaption would be along the ξ axis in the computational system of (ξ,η). The weight function W is now a function of s (e.g., U_s instead of U_x).

Dwyer has also modelled 2-D problems with two

gradients: a developing velocity front and a thermal burn wave. He used a separate mesh for each adapted variable and interpolated the solutions between the grids. He has privately admitted to mass conservation problems with the calculations[32].

Anderson[28] has attempted to use the equidistribution principle to adapt along both axes of a 2-D problem (using a method similar to Dwyer's). He reports severe grid distortion in unsteady problems.

Eiseman[23] has tried to use ADI-like techniques to iterate for the adaptive mesh locations on a 2-D grid. He first adapts along one coordinate axis and then the other, iterating back-and-forth until the mesh points have converged. He has only tried the scheme on simple test cases; he also reports problems with grid distortion and skewing.

Brackbill[33] has applied the full variational approach to 2-D transient problems with good success. The basis for his method is the Thompson[20] static grid generation method; smoothness is guaranteed by using a Laplace system. The grid orthogonality and local weights (an extension of equidistribution principle to 2-D) are also considered. We first consider a computational space of ξ and η. The global smoothness is measured by the integral

$$I_s = \int_D [\nabla\xi^2 + \nabla\eta^2] \, dv \tag{2.20}$$

the orthogonality by

$$I_o = \int_D [\nabla\xi \cdot \nabla\eta]^2 \, J^3 dv \tag{2.21}$$

and the local weights by

$$I_w = \int_D W \, J dv \tag{2.22}$$

where J is the Jacobian and W is the weight function $W(x,y)$. The result is that the integral I, given as

$$I = I_s + a\, I_o + b\, I_w \quad , \tag{2.23}$$

must be minimized to obtain an optimal grid. This is an extremely complicated and computationally intensive method[19]. It has often been more expensive to generate the time varying grid than it has been to solve the physical problem on the grid[34]. This method attempts to generate the 'perfect' grid.

2.3 Numerical Methods

No attempt has been made to review the entire body of literature on finite difference numerical methods used to solve systems of PDEs. Scores of methods exist: first, second, third, fourth, and higher formal orders of accuracy[19,35,36]. Dwyer[37] has suggested that mesh adap-

tion allows most numerical methods to work better (a statement about convergent finite-difference schemes) and that simpler ones work best. Dwyer and Anderson have used simple first-order upwind differencing and a single step of MacCormack's method[38-41] in their adaptive grid studies[24-29]. The large dissipative qualities of the upwind scheme are minimized with grid adaption; the grid speed near high gradient regions (shocks) will be close to the fluid velocity. The adaptive system is then similar to a lagrangian system in these regions.

Woodward and Colella[42] have compared several methods for solving strong shock (blast wave) problems. The best performer was their own method[43], which essentially is a lagrangian scheme limited to inviscid flow (they use the method of characteristics to determine local fluxes). Pure eulerian schemes (MacCormack method) achieved similar accuracy with many more grid points. Flux corrected transport (FCT) methods[14], with fewer mesh points, obtained comparable accuracy with the MacCormack methods but required almost twice the computer time. The FCT methods blend low-order and high-order fluxes in an attempt to resolve sharp discontinuities without oscillations. Adaptive grid schemes simply add more points to these regions.

The literature has shown that the MacCormack scheme is

robust, easily adapted and vectorizable[44]. MacCormack's method is a predictor-corrector version of the Lax-Wendroff method[45]. In the past years, MacCormack has come full circle in philosophy for his method. It was originally an explicit scheme for the full Navier-Stokes equations. Later, a method of increasing the step size was implemented but was limited to inviscid regions (through the use of characteristic equations). The code converged time asymptotic problems much faster (on a scalar computer) but its range of applicability was limited. This new method also lost its full vectorizable performance. MacCormack then switched to an alternating bi-diagonal method to obtain an implicit scheme but still retain the explicit method's degree of vectorization. In his latest perturbation[41], MacCormack has returned to a variant of his original explicit code because it handles various boundary conditions in a more consistent manner than did his implicit version. Implicit methods have no inherent advantage over explicit methods if time accurate solutions (unsteady time) are desired[46]; both must be effectively limited to the Courant time step for accuracy (implicit schemes) or stability (explicit schemes) requirements. The latest MacCormack method is rather general and robust; it can be easily modified to include flux splitting, implicit solvers, multigrid convergence

schemes, and Newton iteration for steady state solutions. The MacCormack schemes are very flexible schemes[46].

The artifical diffusion characteristics of first upwind differencing schemes are being addressed by second order or quadratic upwind schemes (QUD). Leonard[47] has developed such a method. Essentially, the upwind gradients are determined assuming a quadratic rather than a linear profile for the dependent variables. These methods have the same formal accuracy as does MacCormack's method.

3.0 METHODOLOGY

There are three different areas which must be coupled together before successful numerical simulations can be achieved: the physical equations (PDEs), the adaptive mesh, and the numerical method. Each of these will be discussed in the following sections.

3.1 Model Equations

The first three test problems involve a single ideal gas; the Navier-Stokes equations were used with a gamma law equation of state. The last test problem simulated the propogation of a radiation hydrodynamics blast wave. The fluid was modeled by the Navier-Stokes equations and the radiation field by a single temperature diffusion approximation.

3.1.1 Navier-Stokes Equations The Navier-Stokes equations have been derived in countless publications for the past century so the results will only be stated here (for example, see ref. 48 for a complete derivation).

The equations, written in one-dimensional conservation vector form (cartesian coordinates), are:

$$U_t + F_x = 0 \qquad (3.1)$$

transformation to an adaptive system remains in a conservative form.

3.1.2 Radiation Hydrodynamics A two temperature diffusion model was used to simulate the radiation hydrodynamic blast wave problem. In this model, the fluid was modeled by the Euler equations (inviscid Navier-Stokes equations) and the radiation field with a diffusion equation. This was the same technique which the author previously used to simulate two-dimensional problems on a fixed eulerian mesh[51]. The equations are quite similar to the Navier-Stokes equations presented above[52]. The continuity equation is identical and the momentum equation is only modified to include the radiation pressure (assumed to be one-third of the radiation field energy density). The major difference is found in the energy equation: there are now two. They are as follows:

$$[e]_t + [ue + uP - u\mu^* u_x - kT_x]_x + \Omega = 0 \quad (3.5)$$

and

$$[e_r]_t + [ue_r + uP_r - q]_x - \Omega = 0 \quad (3.6)$$

where e and P are the energy density and pressure of the fluid and e_r and P_r are the radiation energy density and pressure. Ω is the net energy exchange between the fluid

and the radiation field from absorption and emission (Note that this quantity is a source or sink term for the transfer of energy between the two equations). The q term is a flux-limited equivalent diffusion term used to model the radiation field as a diffusion equation[53]. Detailed descriptions of these terms can be found in reference 53.

3.2 Adaptive Mesh Generation

The present investigation used an explicit adaptive mesh generation technique to determine the mesh metric and velocity. Recall that equation 2.7 was the explicit equation for determining the mesh point distribution given the weight function. We restate it here for convenience,

$$\int_0^x W dx' = \xi(x) \left(\frac{\int_0^L W dx}{N} \right) . \qquad (3.7)$$

The procedure is explicit since all the quantities on the right hand side are known at time n and the weight function on the left side is also defined at time n. Therefore, the only unknown is the upper limit, x, on the left hand integral, which is the new mesh spacing at time n+1.

Numerical experience with discontinuities has indicated that a simple trapezoid integration scheme is preferred to an Euler method. The use of a linear in-

tegration function prevented discontinuous function values. The weight function was defined as a cell center quantity while the mesh location, x, was at the cell edge. $\xi(x)$ is simply a function whose value is equal to the mesh; that is, for the first mesh its value is 1, the second its value is 2 and so on until at the last mesh its value is N. The second term on the right hand side represents the average $W\Delta x$ per each mesh cell. The upper integral limit, x, is then numerically determined to satisfy this equation. This loop was the only one in the computer program which resisted efforts to vectorize. The adequacy of this explicit procedure was one of the goals of this investigation and was evaluated via unsteady flow numerical experiments.

After this loop had been performed, the mesh edge velocity could be calculated via a first order backward difference:

$$x_\tau = (x^{n+1} - x^n) / \Delta t \qquad (3.8)$$

where Δt is the computational time step. The mesh metric is simply the difference in the space locations during the time step. The PDEs can now be integrated in time to obtain the new values and the process repeated.

The key part of this explicit procedure is the determination of the weight function. The following section

will describe this portion of the method. Analytical functions will be used to illustrate specific features of the method. That is, the PDEs have been replaced with simple functions; therefore, the test is essentially a steady-state one.

3.2.1 Weight Function As was previously discussed, many forms of the weight function have been postulated. The present investigation has chosen to extend Dwyer's weight function (equation 2.16) to more than one control variable. The following form was used

$$W = 1 + \alpha_a |A_x| + \beta_a |A_{xx}| + \alpha_b |B_x| + \beta_b |B_{xx}| . \quad (3.9)$$

The functions A and B represent some normalized physical quantity such as velocity, pressure, or temperature. The α's and β's are determined from relationships similar to equation 2.17; thus, there is a corresponding R for each α and β. One could easily consider adding more terms as needed.

Simple centered differences on both the first and second derivative terms were used. Stability questions concerning the use of centered differences for a first derivative term are not an issue since the weight function only determines the mesh point movement, which is unre-

lated to the solution of the governing PDEs.

3.2.2 Smoothing the Mesh It is generally accepted that, irrespective of which form of the weight function is used, the mesh point distribution must be smoothed prior to time step advancement with the PDEs. In this regard, an explicit adaptive approach has a distinct advantage over an implicit approach. Dwyer[54] used a control function which limits the change in the mesh size from one mesh to another; interpolation must be used to obtain the new physical properties since mesh points may be added or deleted. Winkler[30] used a parabolic equation to control the mesh movement and to achieve some degree of smoothness.

The present work does not smooth the time n+1 mesh point distribution; rather, the weight function is smoothed. Since the weight function was chosen to be a function of x and not ξ, smoothing it will naturally result in a smooth mesh point distribution. Also, the logic for performing this step is very simple and robust; the code is completely vectorizable. A simple 5-point weighted average scheme has provided acceptable results in the simple steady-state and the four unsteady test

problems. It can be expressed as:

$$W_i^* = c_1 W_{i-2} + c_2 W_{i-1} + c_3 W_i + c_2 W_{i+1} + c_1 W_{i+2} \qquad (3.10)$$

where c_1, c_2, and c_3 are user defined constants and the i subscripts are the mesh point locations. W^* is the new smoothed weight function distribution; W is defined from equation 3.9. The following values of the c's were used for all of the calculations in the present investigation:

$$c_1 = 0.083 \qquad (3.11)$$
$$c_2 = 0.166$$
$$c_3 = 0.500.$$

They were chosen so that the center point had a weight of 1/2, the two nearest neighbor points had a combined weight of 1/3, and the next two a combined weight of 1/6.

A simple test problem demonstrates the effect of this smoothing algorithm. Consider the function shown in figure 3.1; it is a simple discontinuity (ideal shock). The adaptive mesh point distribution which would have been obtained with 20 points and no smoothing is shown; the distribution from utilizing the smoothing algorithm is also shown. We can note that the mesh spacing in the regions away from the discontinuity appear to be similar for both distributions while the smooth distribution has a more gradual change in the vicinity of the discontinuity.

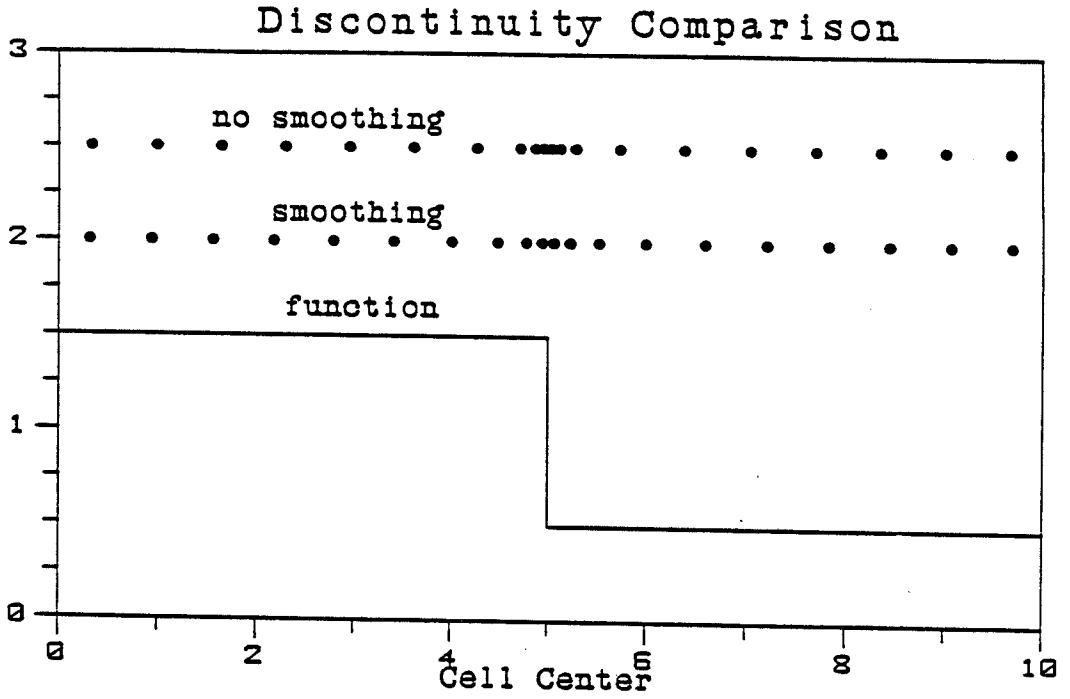

Figure 3.1 Mesh spacing for a simple discontinuity, with and without smoothing.

This difference is shown in figure 3.2, the mesh spacing for the two distributions. Here it can be easily seen that the smoothed case has a more gradual transition zone from the constant function value to the discontinuity.

3.2.3 Minimum Mesh Control A problem exists if the weighting scheme just describe is used to simulate a blast wave in an ideal gas. The large discontinuity implies a large gradient which results in a large local weight and, from the equidistribution principle, a very small mesh spacing. The stable time step would therefore be quite small and the calculation would require large amounts of computer resources. The obvious solution is to limit the mesh spacing to a user input value. Winkler[30] incorporates some mesh spacing control into his method; however, it is done in an *ad hoc* manner.

The present investigation has chosen to modify the local weights, W, as did the smoothing algorithm to limit the mesh spacing. By changing the weights before equation 3.7 is solved, a smooth mesh spacing distribution is obtained. The procedure was not designed to limit the mesh spacing to an exact value, but rather the minimum spacing will be approximately the limit value. The method is simply that after the weight values have been calculated from equation 3.9, the average $W\Delta x$ is determined

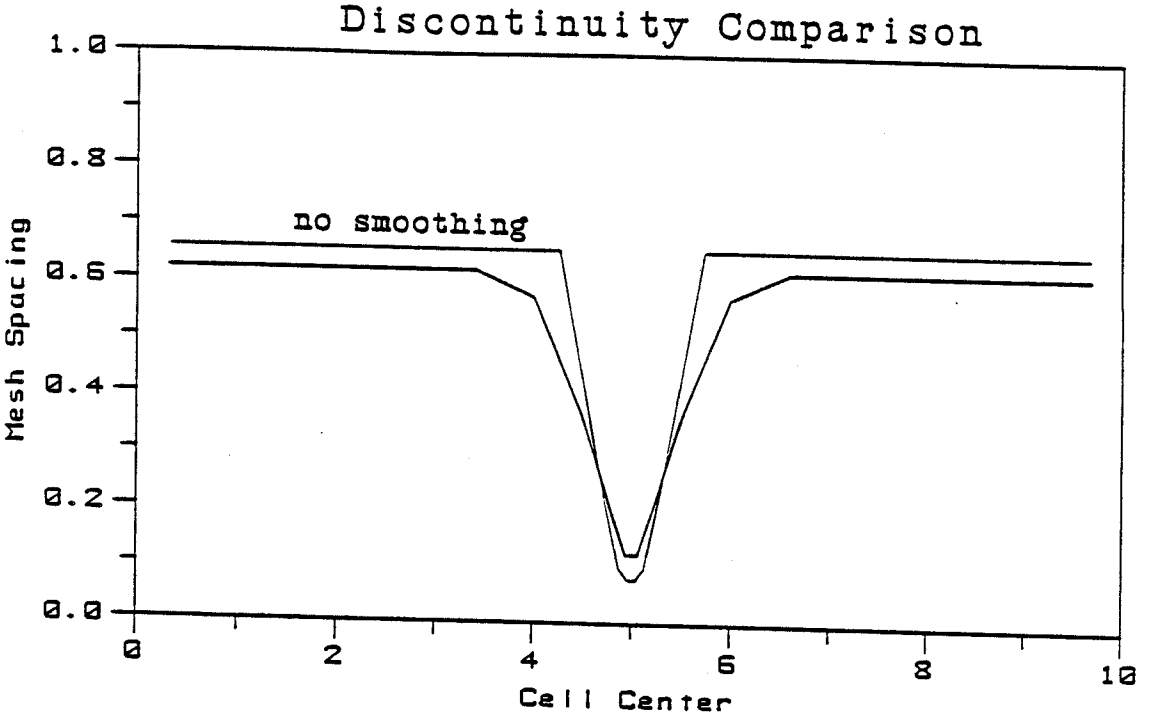

Figure 3.2 Mesh spacing for smoothed and nonsmoothed simulation

(the second term on the right hand side of equation 3.7). Since a minimum Δx is specified, the maximum W can then be obtained. Finally, a simple loop through the weight values using a vectorized compare statement results in an upper bound on the weight values. The smoothing algorithm, previously described, is then performed. All of the calculations presented, including the preceding analytical discontinuity, used this minimum mesh control logic.

3.2.4 Steady-State Simulation A simple problem was devised to test all of the preceding features of the mesh spacing distribution algorithm. This problem consisted of two functions and is shown in figure 3.3. The general characteristics of the functions are similar to those of the model problem: laser ablation of a surface where F1 represents the density and F2 the temperature. A 500 point uniform mesh was used to obtain the figure.

As a comparison calculation, a 20 point uniform mesh was calculated. Figure 3.4a shows the point distribution as well as the function values superimposed on the 500 point curve. The points indicate the locations of the mesh centers. Figure 3.4b shows the first derivative values for the 20 and 500 point uniform meshes. It is clear that although the function values do not differ

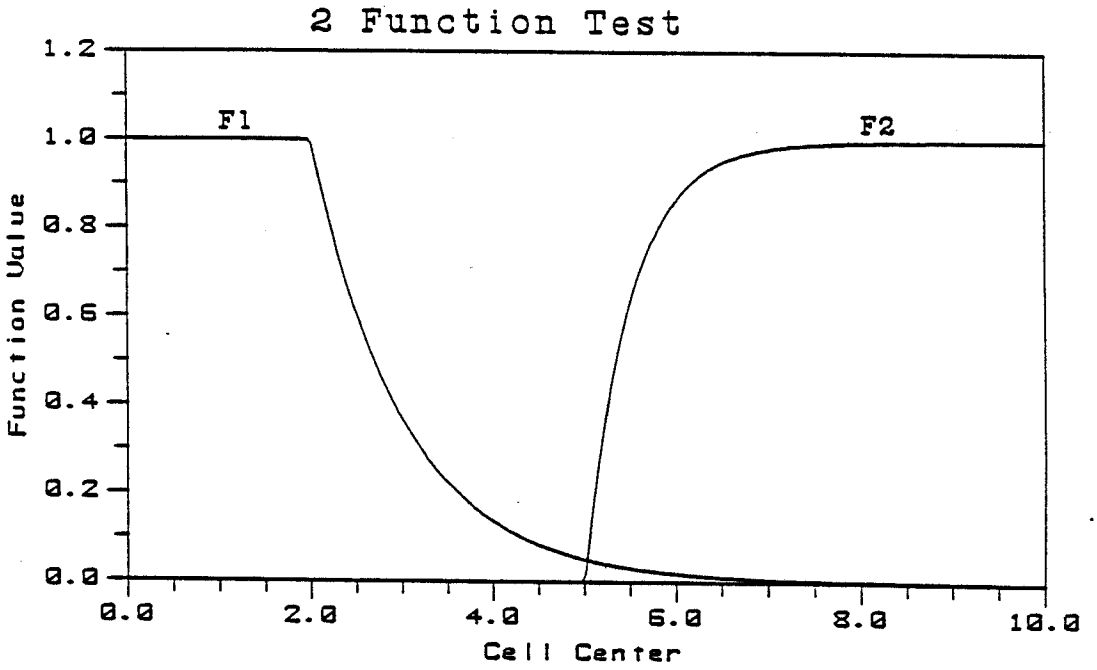

Figure 3.3 Function values for steady-state 2 function test.

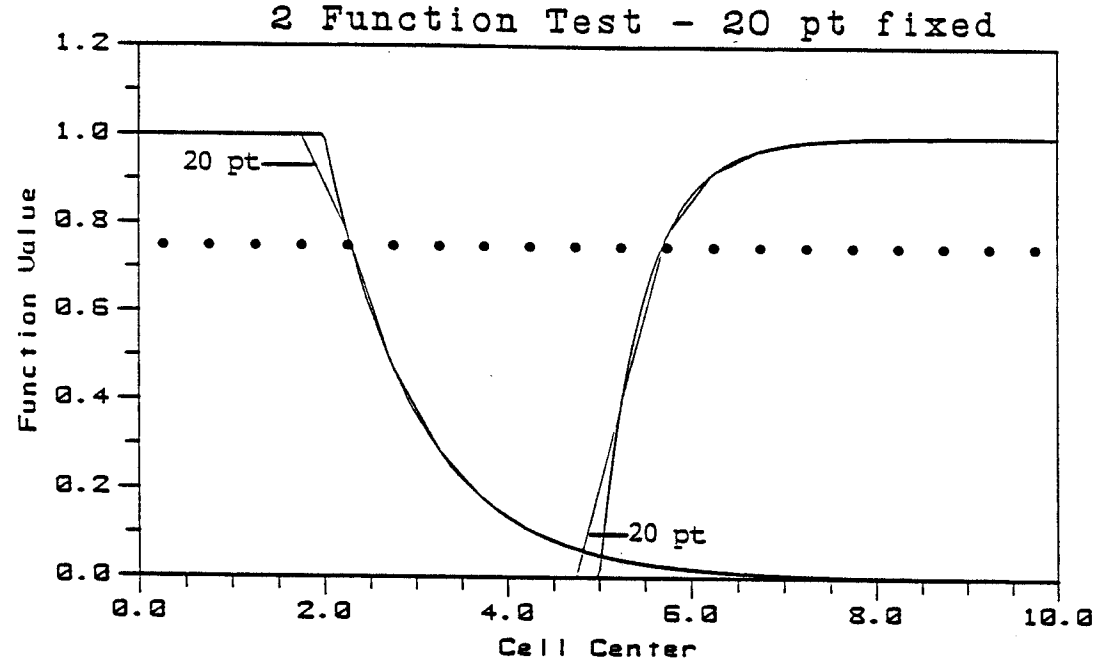

Figure 3.4a Function values for 20 pt fixed mesh

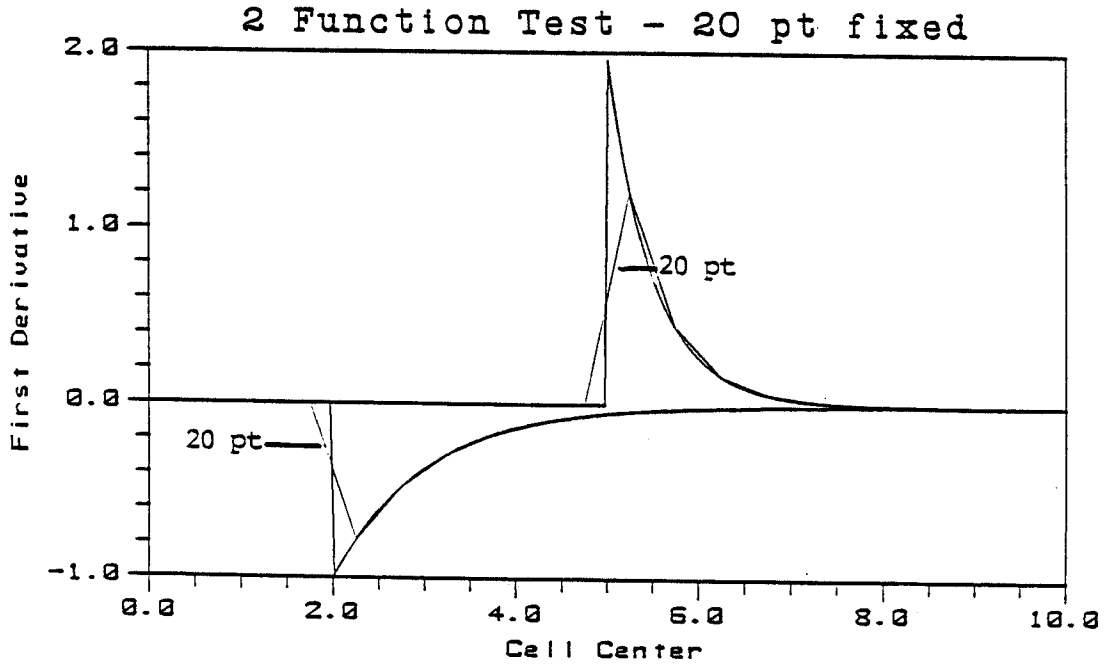

Figure 3.4b Function derivatives for 20 pt fixed mesh

significantly in figure 3.4a, the first derivative values differ near regions of high gradients. Since the governing PDEs consist mainly of gradients across a mesh, it is important to be able to accurately determine this quantity.

Figure 3.5a shows the point distribution and the function values for a 20 point adaptive mesh calculation. The R values used were $R_{\alpha a} = 0.3$, $R_{\beta a} = 0.1$, $R_{\alpha b} = 0.3$, and $R_{\beta b} = 0.1$, where $R_{\alpha a}$ is essentially the percentage of grid points to be assigned to the first derivative of the first function (see equation 2.17). These values were typical of the values used for the four unsteady test problems. The clustering of the mesh points near the high gradient regions can be clearly seen. Figure 3.5b shows the first derivative values for this mesh distribution. Comparing the adaptive and non-adaptive results (figure 3.4b), we can see that the adaptive mesh is a better predictor of the second gradient region.

Figure 3.6a shows the point distribution and function values for a 50 point adaptive mesh calculation. The R values were the same as for the 20 point calculation; the minimum mesh spacing was reduced from 0.1 to 0.05. Here the mesh clustering at the gradient regions is clearly shown. Figure 3.6b shows that the first derivative is predicted much better than the 20 point adaptive

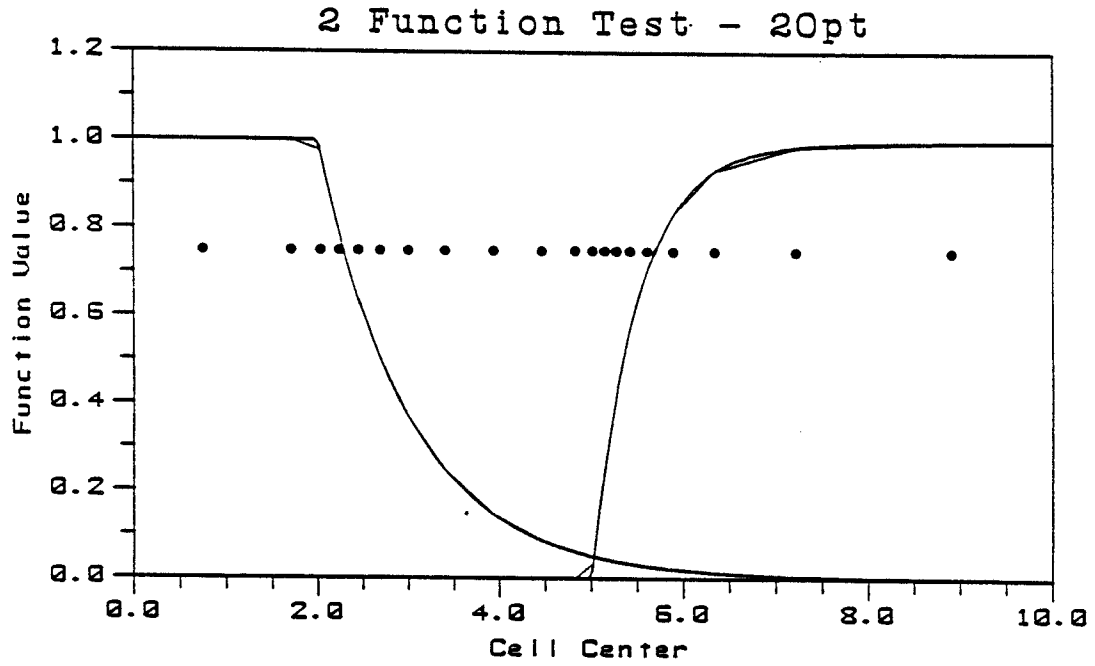

Figure 3.5a Function values for 20 pt adaptive mesh

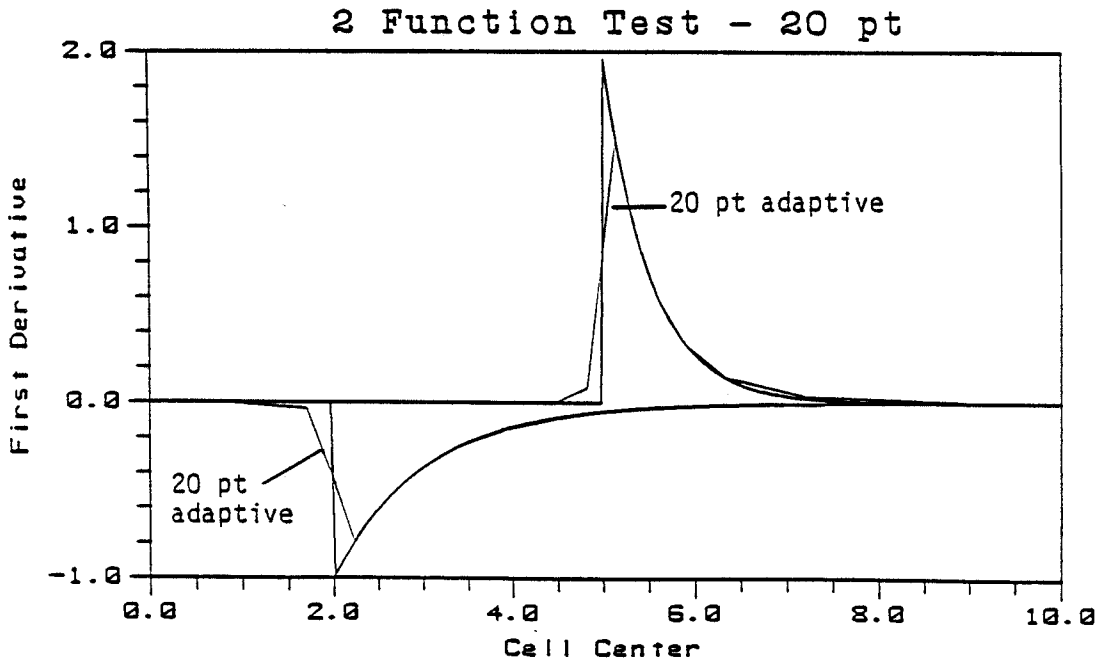

Figure 3.5b Function derivatives for 20 pt adaptive mesh

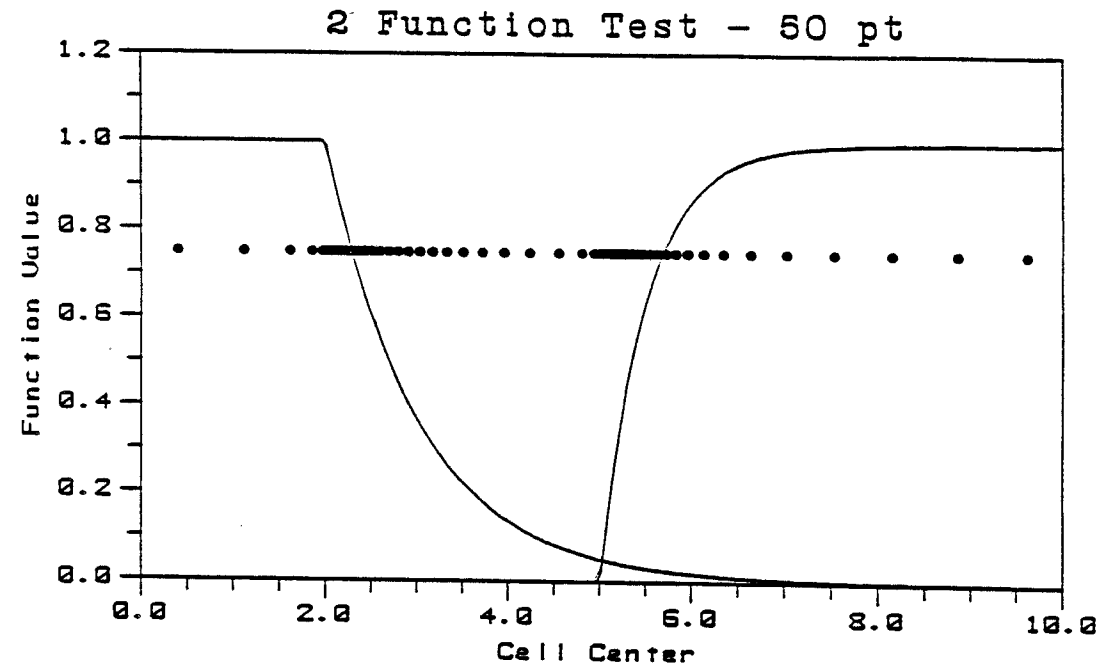

Figure 3.6a Function values for 50 pt adaptive mesh

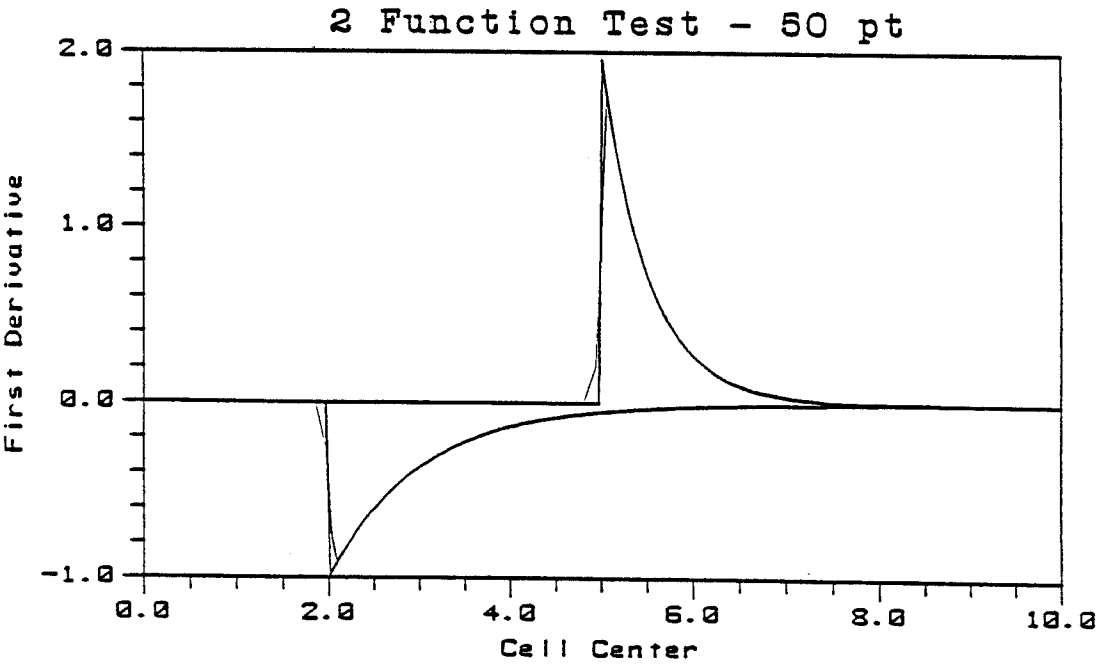

Figure 3.6b Function derivatives for 50 pt adaptive mesh

calculation. The differences are not quantitatively given, for example by a least-squares error estimate, since these calculations only represent simple test problems. In an actual numerical simulation, centered differencing of the first derivative would not be used.

Figure 3.7 shows the mesh spacing for the 20 and 50 point adaptive calculations. Recall that the minimum value of the 50 point calculation was reduced from 0.1 to 0.05. One can note that the mesh clustering at the two high gradient regions is relatively smooth. Also, the 50 point calculation not only clustered more points at the high gradient regions, but the maximum mesh spacing was also reduced. This is a useful feature for problems which have both high and mild gradient regions (e.g. the shock tube test problem).

The weight function values, with the smoothing and minimum Δx algorithms, for the 50 point calculation are shown in figure 3.8. The smoothing algorithm has rounded the high gradient regions. The locations of the maximum and minimum values of this plot and the corresponding mesh spacing shown in figure 3.7 are reversed since $W\Delta x$ is a constant for each mesh.

Since this adaptive procedure explicitly solves equation 3.6 for the mesh spacing, several iterations are needed to achieve a converged solution. Figure 3.9 shows

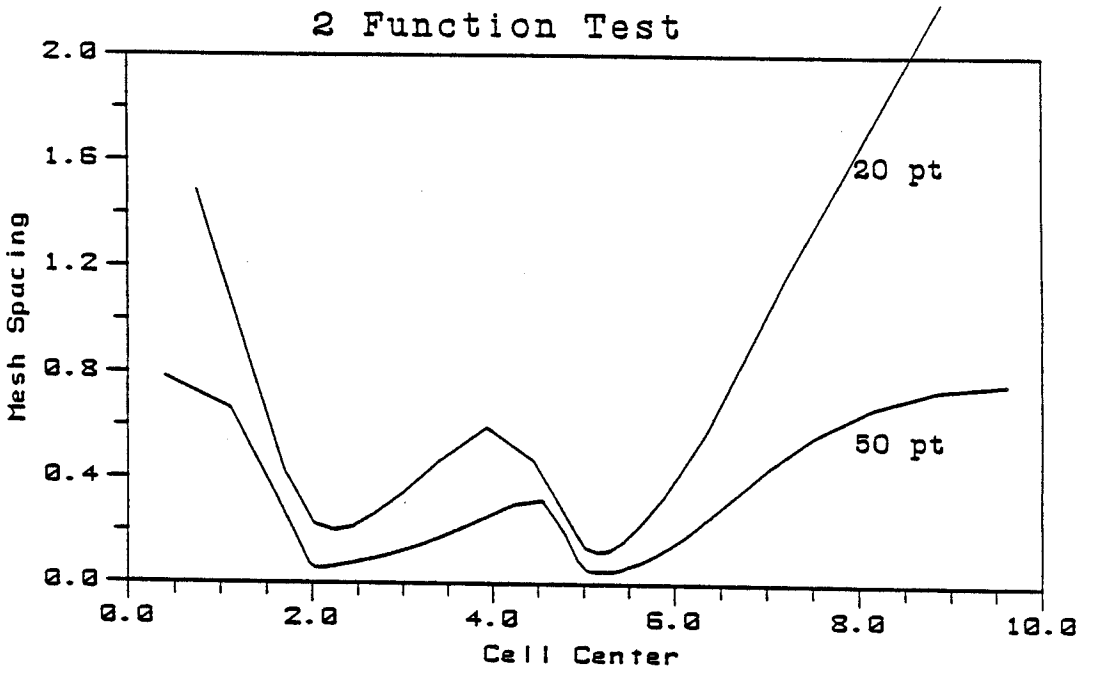

Figure 3.7 Mesh spacing for 20 and 50 pt adaptive mesh cases

(This page was intentionally left blank.)

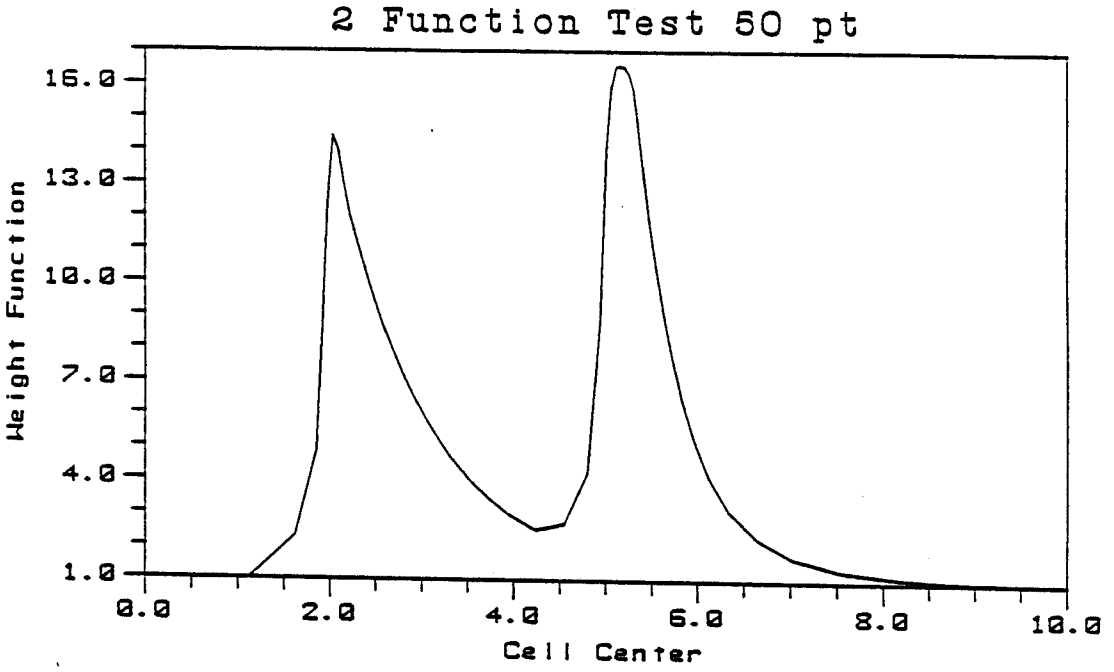

Figure 3.8 Weight function distribution for 50 pt adaptive mesh

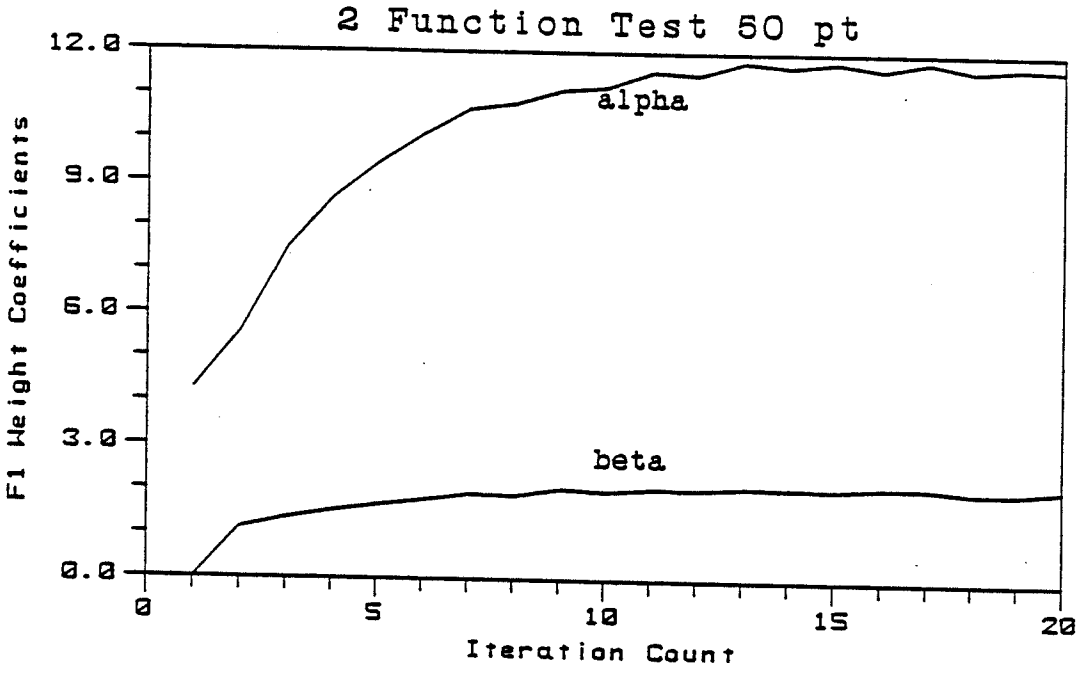

Figure 3.9 Convergance plot for the weight coefficients (50 pt adaptive case)

α_a and β_a as a function on the number of iterations. Two points can be made from this plot. The first is that it is important that the adaptive procedure be iterated at the start of a simulation to achieve a converged mesh distribution. Otherwise, the mesh movement will not be tracking the time varying solution. No iteration was performed in the four test problems after time 0+. The second point to be observed is the slight wiggle in the α curve. This oscillation was smoothed out in the unsteady test problems by linearly time averaging the α's and β's.

3.3 Numerical Method

The previous sections described the governing partial differential equations and a procedure for generating an adaptive mesh. The present section will describe the coupling of these two parts of the numerical simulation problem. First, the transformation of the PDEs from a fixed grid to an adaptive grid will be described and then various numerical procedures for solving these equations will be examined.

3.3.1 Adaptive Equations Only the transformation of the momentum equation from a fixed coordinate system to an adaptive system will be shown; the continuity and energy

equations are only minor perturbations from it.

The momentum equation can be written in a strong conservative form[10] from the vector equations 3.1 as

$$[\rho u]_t + [\rho u^2 + P - \mu^* u_x]_x = 0 \ . \qquad (3.12)$$

We now use the same transformations relating the physical domain (x,t) and the computational domain (ξ,τ) as was used earlier,

$$\tau = t \qquad (3.13)$$
$$\xi = \xi(x,t) \ .$$

Transforming the momentum equation into computational coordinates (see Appendix A for details) and retaining the strong conservation form yields:

$$[\rho u x_\xi]_\tau + [\rho u u + P - \rho u x_\tau - \mu^* u_\xi / x_\xi]_\xi = 0 \ . \qquad (3.14)$$

It is noted that the time conserved quantity has been changed from ρu in equation 3.11 to $\rho u x_\xi$ in equation 3.14. Also, an additional convective term, $\rho u x_\tau$, has appeared in the spatial derivative. This represents the convective momentum flux due to the motion of the mesh. The ρu^2 term in equation 3.12 was intentionally rewritten as $\rho u u$ to illustrate that it represents the convective momentum flux due to the motion of the fluid through a fixed mesh. We now can slightly rewrite equation 3.14 as

$$[\rho u x_\xi]_\tau + [\rho u(u - x_\tau) + P - \mu^* u_\xi / x_\xi]_\xi = 0 \quad .(3.15)$$

Now the velocity of the convective momentum flux is explicitly shown as the relative velocity between the fluid and the mesh. Recall that in a pure eulerian coordinate system $x_\tau = 0$; in a lagrangian system $x_\tau = u$ and there is no convective momentum flux.

There are two paths which one can take to formulate the numerical approximations of these equations: the differential approach or the control volume approach. Typically the differential approach has been used[19]. In this methodology, equation 3.15 is considered to be a PDE and not as a conservation equation. The development of the difference equations follows a pure mathematical tack; the physical significance of the variables has been lost. In the control volume approach, the conservation nature of the equations is retained. For a typical eulerian mesh system, both approaches usually arrive at the same set of discrete equations. However, for the adaptive mesh, the control volume approach is the only viable approach if conservation of the physical quantities is to be achieved. This is because the mesh dilates and translates each time step; the relative fluid and mesh velocities at the left mesh edge differ from the right mesh edge velocities and the cell center velocity. The distinction

between a cell edge and a cell centered velocity is very important.

The different velocities can be more easily illustrated if we first write a reduced momentum equation, neglecting the pressure and viscous terms:

$$[(\rho u)_c x_\xi]_\tau + [(\rho u)_e (u_e - x_\tau)]_\xi = 0. \qquad (3.16)$$

The subscript c indicates a cell centered quantity while the subscript e indicates a cell edge quantity. The form of this equation leads quite naturally to a control volume formulation of the difference equations.

We see from equation 3.16 that quantities need to be defined both at the mesh edge as well as the mesh center. Most control volume computer codes define the locations of the density, pressure and energy variables to be at the center of the mesh. The location of the velocity can be at the mesh edge (staggered mesh), shown in figure 3.10b, or at the cell center, shown in figure 3.10a. The present investigation defined the velocity at the cell center. Therefore the velocity at the cell edge, u_e, must be determined from the cell center values. Investigators typically choose a weighted average of u_c based on the mesh spacing; that is, a simple average when the mesh spacings are equal. This was also done in the present work. The location of the grid velocity, x_τ, is defined

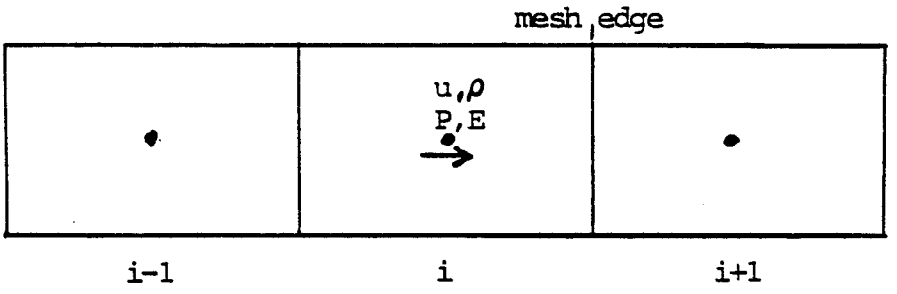

Figure 3.10a Cell centered mesh orientation

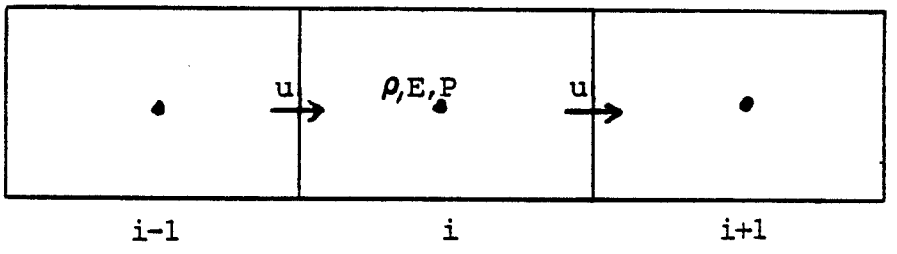

Figure 3.10b Cell edged mesh orientation for velocity

to be at the cell edge; this is also its location in equation 3.16. Therefore the mesh edge momentum flux, $(\rho u)_e$, is the last quantity which needs to be defined. The numerical methods which are discussed in the following sections calculate this quantity differently.

Essentially all of the methods use a first order forward time differencing scheme to approximate the time derivative. The time difference equation for the momentum equation would be:

$$[(\rho u)_c x_\xi]_\tau \simeq \frac{[(\rho u)_c^{n+1} x_\xi^{n+1/2}] - [(\rho u)_c^n x_\xi^n]}{\Delta t} \quad (3.17)$$

The mesh metric is shown at the n+1/2 time rather than the n+1 time to illustrate the point that since an explicit adaptive procedure was used, the metric was not solved at the same time level as the conserved quantity. If the adaption procedure would have been implicit, the metric would have been at the n+1 time level and the solution of governing equations would have been directly coupled to the adaptive mesh equation. Thus, the present investigation used an explicit numerical method and an explicit adaptive grid scheme to advance the numerical solution.

The energy transfer term, Ω, in equation 3.5 poses a problem in the development of a conservative method; equation 3.5 is not in a strong conservative form. This

term is essentially an energy source or sink term and has units of energy density per unit time, or a volumetric heat flux term. The technique by which the present investigation considered this problem can best be illustrated by the following simple differential energy balance:

$$V(t)de = V(t)\Omega\, dt \qquad (3.18)$$

where V(t) is some time varying volume, de is the differential energy density of that volume and dt is the differential time step. Since the governing PDEs were to be solved by an explicit method, Ω is a constant over the differential time. If we integrate both sides from time n to n+1, we have

$$[e^{n+1}V(n+1)] - [e^{n}V(n)] = \Omega\int_{n}^{n+1} V(t)dt. \qquad (3.19)$$

The left side of the equation just represents the net change in energy in the volume from time n to n+1; only the values at the end states are needed to determine it. The right side of the equation represents the time varying source or sink energy term in the volume; the path that this expression takes from time n to n+1 is important. Although Ω has been assumed a constant, the volume element cannot be constant since the adaptive mesh technique

changes the mesh volume for each time step. However, the adaptive procedure only identifies the mesh volume at the time end points; the temporal change is not a known quantity. The present investigation has simply assumed a linear profile for V(t); that is,

$$\int_{n}^{n+1} V(t)dt \simeq 0.5\ [V(n+1) + V(n)]\Delta t \quad . \qquad (3.20)$$

3.3.2 First Upwind Differencing First upwind differencing or 'donor cell' methods have been extensively used in numerical simulation for a number of years. For a detailed description, see Roache[35]. Recall from the previous section that quantities such as the cell edge momentum density are required (see equation 3.16). This method simply equates the edge value with one of the neighboring cell center values dependent upon the sign of the edge velocity, u_e. That is, if u_e is positive, the left center value is used; the right center value is used if this velocity is negative. Thus the edge flux is an 'upwinded' term. For the adaptive mesh problem, the relative cell edge velocity, $(u_e - x_\tau)$, was used as the control velocity.

The major problem with using this method is its inherent numerical dissipation; that is, since it is a first

order accurate method, the Taylor series truncation error is a viscous-like term. Roache points out that the transient diffusion-like term is multiplied by the quantity (1 - CFL). Recall that the CFL number is the ratio between the fluid velocity and the characteristic mesh velocity. Thus if the time step is chosen to be such that the CFL is unity, this term vanishes. If the CFL number is much smaller than 1, the numerical diffusion is then similar to the von Neumann-Richtmyer 'artificial diffusion' model. A uniform mesh spacing would be required to obtain a CFL of unity as a shock traversed the mesh; many mesh points would be needed to accurately simulate the problem.

The explicit 'donor cell' method is easily vectorizable and has proven to be quite robust.

3.3.3 Quadratic Upwind Differencing In an effort to overcome the inherent dissipation property of first upwind differencing, quadratic upwind differencing methods are currently being developed. This method is similar to the 'donor cell' method except that now a quadratic profile is constructed on the 'upwind' side of the cell to predict the cell edge value. Leonard's method[47] was evaluated as the numerical method for the present investigation. However, severe oscillations developed in the cases of

strong shocks. This will be illustrated in chapter 4.0 for the shock tube problem. It is postulated that this is due to the quadratic profile; the edge pressure was less than the physical minimum value. Pollard[55] and Patel[56] report problems with QUD methods with high cell Peclet numbers. For problems with discontinuities or shocks, a hybrid method using 'donor cell' differencing at the discontinuity and QUD elsewhere could be an appropriate compromise method.

3.3.4 MacCormack's Method As discussed earlier, MacCormack's explicit method is a two-step, predictor-corrector version of the Lax-Wendroff method. This is a very popular method because it is vectorizable, robust, and formally second order accurate. Essentially, the results from both an upwind and a downwind differencing step are averaged to obtain the new cell centered quantities. However, this method could not be used since it is a two-step procedure; the method calculates intermediate quantities at some time between n+1 and n. Recall that in equation 3.15 the mesh metric appears in the time differencing. There is no physical meaning to an intermediate quantity since the mesh has changed from its time n location to its n+1 location. For this reason, ADI and ADE methods also do not appear to be useful for adaptive

mesh systems.

3.4 Computer Codes

Two computer codes have been developed for this investigation: a one-dimensional compressible flow Navier-Stokes solver and a one-dimensional radiation hydrodynamics code. Each code was written to take full advantage of the vector capabilities of a Cray supercomputer. All output was postprocessed on an IBM-PC workstation.

Time step control was controlled by setting the CFL number to a constant value. Since an explicit numerical procedure was used, this number has a maximum of unity. The compressible flow CFL number is typically defined[35] for a fixed mesh system as:

$$CFL = (|u| + a)\Delta t/\Delta x \qquad (3.21)$$

where a is the local sound speed. The present investigation defined an adaptive CFL as follows:

$$CFL = (|u_{cell}| + a)\Delta t/\Delta x \qquad (3.22)$$

where $u_{cell} = 0.5(u_r + u_l)$. u_r and u_l are defined as the right and left edge relative velocities respectively. For example $u_r = u_e - x_\tau$, where u_e is the edge fluid velocity computed from a geometric average of the cell center

velocities and x_τ is the mesh edge velocity.

A summary of the major computational steps which were used to advance the solution from time n to n+1 are:

1. Explicitly calculate α's and β's from equations similar to 2.17. Integral of the weight function is from old time step (n-1).
2. Time average α's and β's.
3. Calculate new weight distribution from equation 3.9 using updated variables (time n).
4. Perform limit on maximum weight value for minimum mesh control.
5. Smooth weight distribution using equation 3.10.
6. Compute new mesh point distribution by integrating equation 3.7.
7. Calculate mesh velocity using equation 3.8.
8. Time x_ξ and x_τ are now defined at time n+1.
9. Solve the governing PDEs for time n+1 variables.
10. Repeat steps.

The difference equations for the solution of the PDEs can be found in Appendix B and for the weight function in Appendix C.

3.4.1 Navier-Stokes This computer code used the explicit 'donor cell' differencing method with an explicit adaptive mesh generation technique. As previously mentioned, an iteration step was provided to converge the initial mesh

distribution. No iteration was performed during the time advancement.

This code consists of approximately 1500 lines of standard Fortran and is written in a modular form.

3.4.2 Radiation Hydrodynamics This computer code started with the compressible code mentioned above and added an explicit two temperature diffusion approximation for the radiation field and an EOS package in tabular form. Essentially the same techniques from a prior two-dimensional radiation hydrodynamics computer code (eulerian mesh)[51] were used in the present code. However, there were some special modifications which are unique to the grid adaption algorithm.

The first has been previously discussed; the dilatation of the mesh during a time step affects the effective volumetric source or sink term, Ω (see equations 3.5 and 3.6). The current investigation has used a simple trapezoid integration technique to approximate this term, see equation 3.20. Since Ω is a source term in one energy equation and a sink term in another, this approximation does not impact the conservation of energy of the method.

The other difference from the prior code to the present one is in the solution strategy for the two energy equations (equations 3.5 and 3.6). Recall that one energy

equation is for the radiation field and the other is for the fluid. The time constant for the propagation of energy via the radiation field is much shorter than by either thermal conduction or convection. The previous code took advantage of this fact by solving the equations using two time steps: the first step calculated only the radiation energy equation and a 'frozen flow' fluid energy equation and the second time step advanced the complete mass, momentum, and energy equations. Typically, the first time step was 1,000 to 10,000 times smaller than the second or hydrodynamic time step. This philosophy was used to simply reduce the code execution time.

However, the mesh dilatation and adaption prevents the use of this simple multiple time step philosophy. The fluid temperature is an important variable and one which would be used as the adaption function in equation 3.9. There are three ways which one could proceed. The first would be to hold the mesh constant for the inner or small time step loop and then the adaption on the fluid temperature would be accomplished in the outer or hydrodynamic loop. The second would be to adapt the mesh in the inner loop. The final technique would be to have only a single time step and simply perform the mesh adaption at the end of the step.

If the first method is used, inadequate tracking of

the thermal wave would occur since significant energy exchange occurs during the inner loop. The mesh movement would not be tightly coupled with the fluid temperature. The second choice, adaption in the inner loop, would result in mass and momentum conservation problems. Since the flow is assumed to be 'frozen', moving the mesh in the inner loop would not change the fluid density or velocity; however, the mesh volume would be changed. This is clearly not a viable choice. The final method would be to simply adapt at the end of a single time step. Since the present investigation is concerned with demonstrating the benefits of grid adaption, computer efficiency is not an overriding factor. Therefore the third method, the single time step approach, was used for the present work.

This code consists of approximately 2600 lines of standard Fortran.

4.0 NUMERICAL SIMULATIONS

Four test problems were chosen to evaluate various features of the adaptive grid technique which has been described in chapter 2.0: a shock wave, a shock tube, interacting blast waves, and a radiation hydrodynamic blast wave. The ideal shock wave problem was chosen as a test problem since it has only a single gradient region and has an analytical solution. The second problem, the shock tube, was chosen since it has two discontinuties and a mild gradient region, the rarefaction fan. The interacting blast wave problem tested the present method to resolve severe gradients and also the adaption procedure as the two waves merged. The final problem was a test with a coupled temperature problem. The intent of these test problems was to evaluate different aspects of the grid method: the adaption variable, the sensitivity to the user defined R's, the minimum mesh control logic, and the coupling between the grid adaption control and the numerical solution method.

The following discussions will not try to develop the optimum set of parameters which would obtain the 'best' solution with the least computer time, but to simply demonstrate the role the mesh has on the quality of the

solution. And more specifically, the role that grid adaption can play in numerical simulation. Also, prototypic values of various parameters will be given.

4.1 Inviscid Shock Wave

The first test problem was a simple shock wave propagating into a polytropic gas at rest; the shock moved from the left boundary to the right. The initial pressure of the gas was 1.0×10^6 (dynes/cm^2) with a temperature of 300°K. The upstream conditions for a Mach 2 shock were analytically defined at the left boundary (for example see ref. 57 for the shock jump relations). This problem served not only to test the grid adaption technique, but also to verify the Navier-Stokes solver.

The initial sharp gradient of this problem prompted the development of the smoothing algorithm for the mesh point distribution. This problem started with the fully developed velocity field, unlike the other three test problems in which the initial and boundary velocities were zero. That is, the velocities had instantaneously high values while the system inertia limited the response time of the other test problems. The grid point clustering provided by the smoothing technique allowed a larger CFL number to be used which reduced the numerical viscosity.

The three calculations to be presented had several parameters in common. First, the momentum density, ρu, was used as the single adaption function. That is, A in equation 3.9 was ρu and B was not used. The CFL number was 0.8 and finally, the artificial viscosity model coefficient was 2.0 (C = 2.0 in equation 3.4).

Figure 4.1 shows pressure profiles for 0.4 ms and 1.0 ms into the simulation. Fifty mesh points were used; the R values were $R_\alpha = 0.35$ and $R_\beta = 0.10$. The minimum mesh spacing was set to 0.1 cm. The results show that an essentially square wave pressure pulse has propagated down the mesh; the propagation speed agreed with the analytical value. These results deviate from the ideal by only the pressure overshoot at the shock front and the slight curvature of the pressure foot. Figure 4.2 shows pressure profiles for the second calculation at the same two times. The input parameters were identical except the minimum mesh spacing was reduced to 0.05 cm. The results now show an ideal shock front propagation. The problems with the previous calculation were due to inadequacies of the simple donor cell numerical method to resolve the gradient region. Note that the total number of mesh points remained constant, only their placement in time was altered. The validity of the Navier-Stokes solver was assured since there were no abrupt changes in the vari-

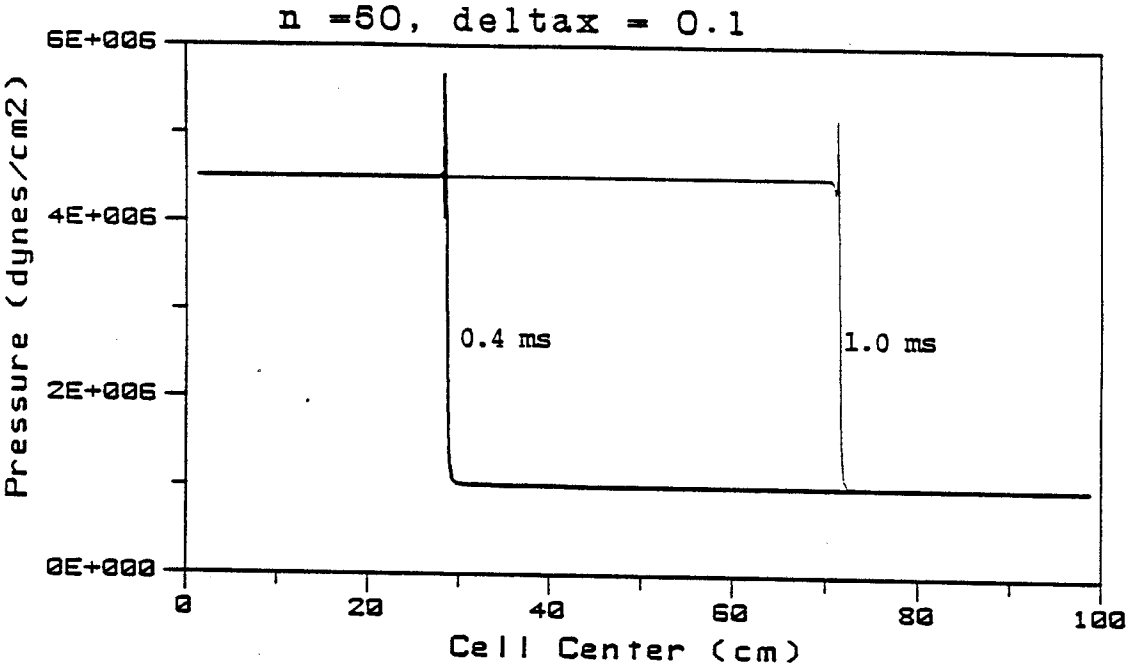

Figure 4.1 Pressure distribution for 50 pt adaptive mesh

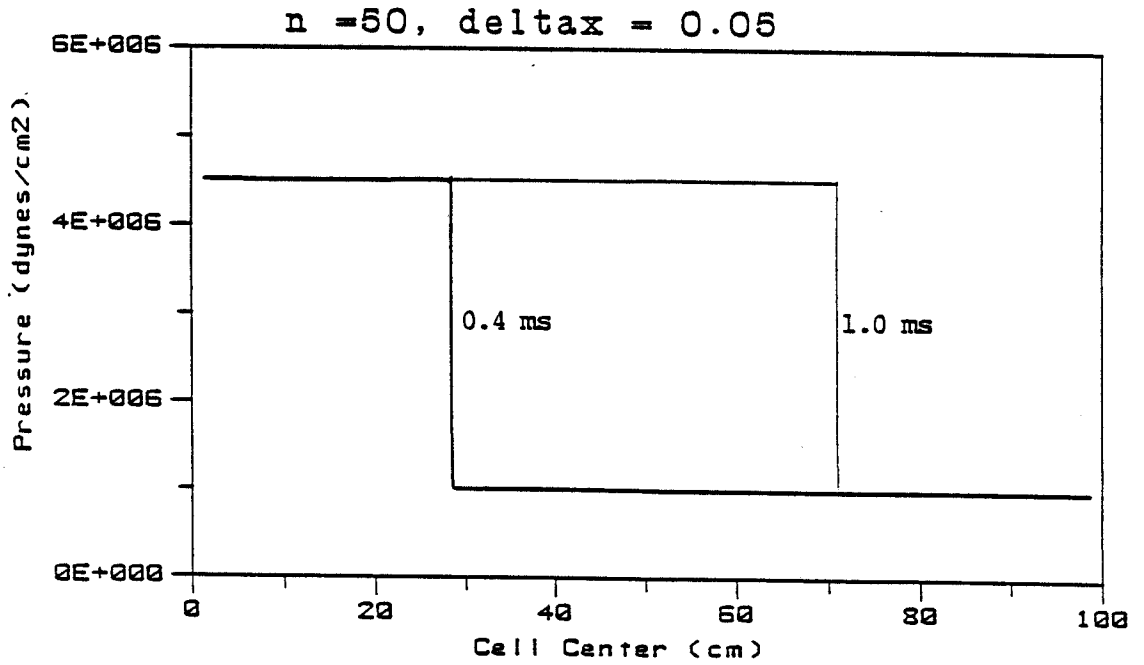

Figure 4.2 Pressure distribution for 50 pt mesh

ables at the left hand, boundary, mesh cells. Since the analytical solution was used to provide the boundary variables, any difference in the code would appear as a change in these variables in the interior of the mesh.

Figure 4.3 again shows pressure profiles for the third calculation at the same two simulation times. However, the number of mesh points has been reduced from 50 to 20, the R values changed to $R_\alpha = 0.50$ and $R_\beta = 0.10$, and the minimum mesh spacing reduced to 0.01 cm. This calculation also shows pressure profiles typical of an ideal shock wave. A pure eulerian mesh would have required 10,000 mesh points to achieve the same results as this adaptive mesh did with 20 points.

We recall that the adaptive momentum differential equation, equation 3.15, had a relative mesh edge velocity, $(u - x_\xi)$, instead of the fluid velocity, u, as in an eulerian mesh. Figure 4.4 shows both of these velocities for the last calculation (20 mesh point) at a time of 1.0 ms. The fluid velocity is typical of a simple shock; the relative mesh velocity is much higher than the fluid, except in the region near the shock. This is due to the movement of the mesh points toward the shock region. Although the relative mesh velocity is from 3 to 7 times larger than the fluid velocity, this does not limit the computational time step (from equation 3.22).

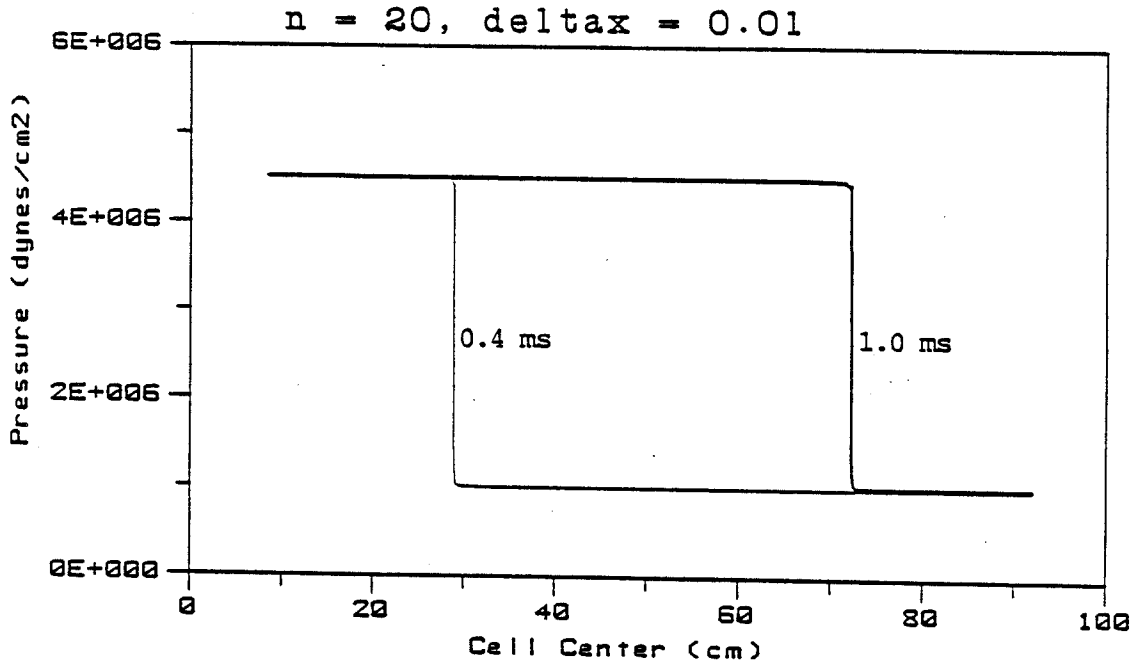

Figure 4.3 Pressure distribution for 20 pt adaptive mesh

This is a non-physical velocity term with respect to the simulation of the fluid problem; its value is transparent to the physical results. Figure 4.5 shows the local CFL number for the same computational time. We see that the limit is at the shock front; the maximum CFL number was 0.8. The reason for this is that the ratio of mesh spacing between the shock and non-shock regions is much smaller than the ratio of the velocities. Recall that the numerical dissipation for the donor cell method has a factor of (1 - CFL); therefore, little numerical smoothing would be expected at the shock front. Of course, the von Neumann-Richtmyer artificial viscosity model is not affected by the CFL number.

Finally, figure 4.6 shows the mesh spacing for three times: 0.2 ms, 0.6 ms, and 1.0 ms. The plot symbols represent the locations of the centers of the computational meshes. Although this simulation used 20 mesh points, only 7 are clearly visible at each time distribution. Thus, 13 points are clustered near the shock front. We note that as the shock propagates from left to right, the mesh spacing outside of the shock zone remains approximately constant. This is unlike a lagrangian mesh, where the mesh points move at the fluid velocity. In this system, the mesh points ahead of the shock would begin to 'bunch', and additional mesh points

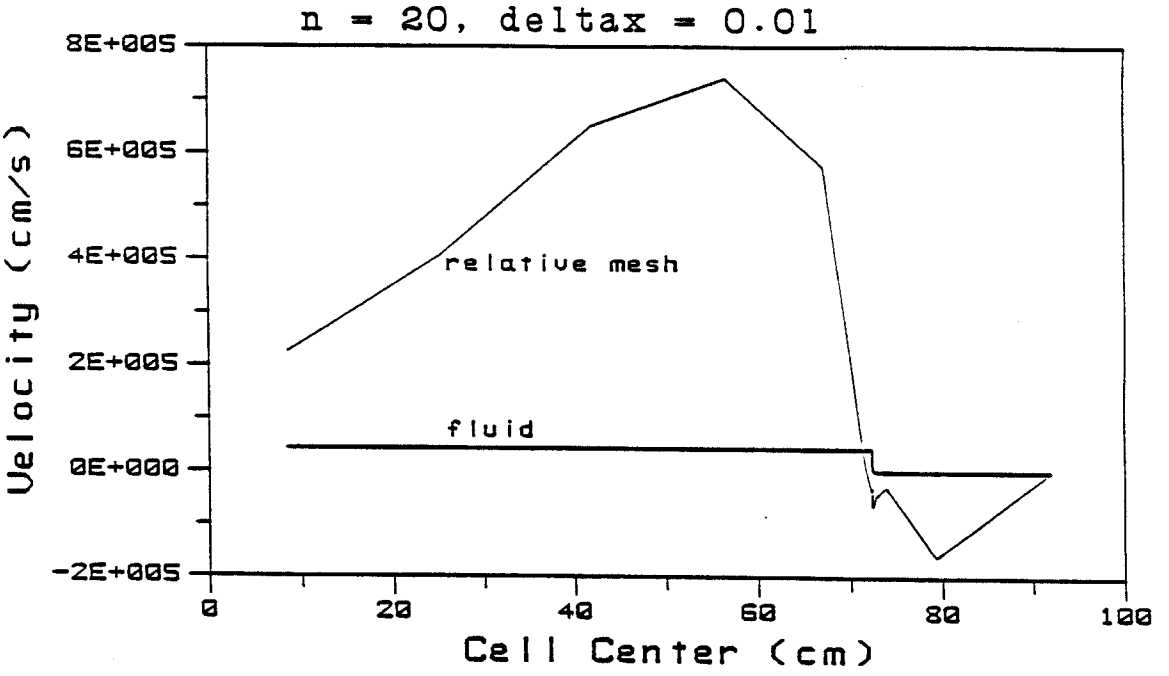

Figure 4.4 Fluid and Relative edge velocity distribution for 20 pt adaptive mesh

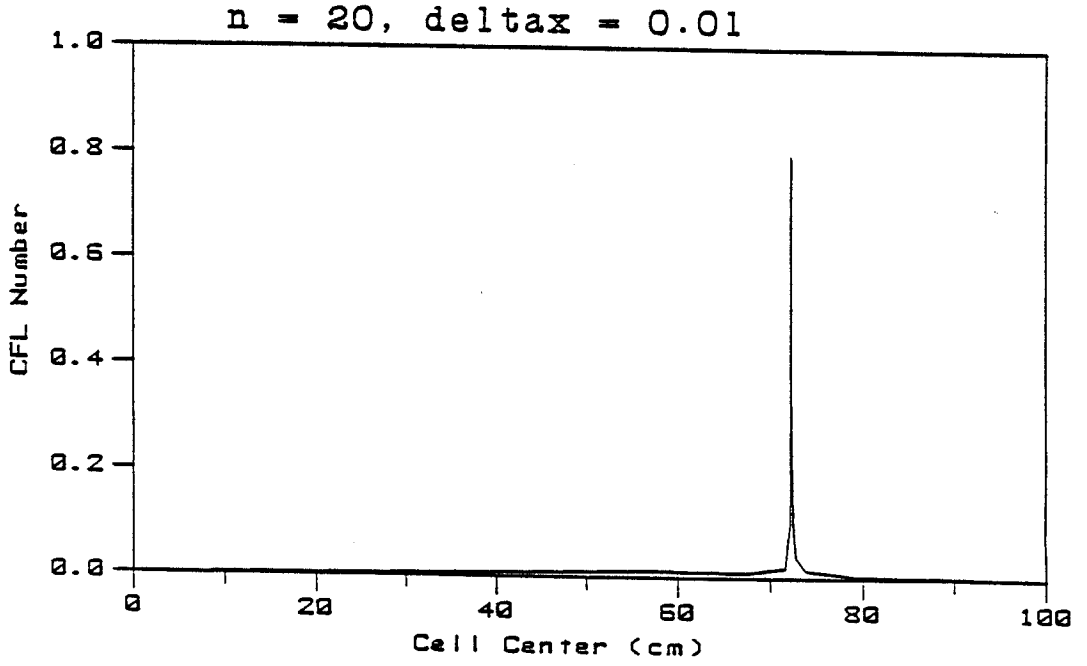

Figure 4.5 Local CFL number for 20 pt adaptive mesh

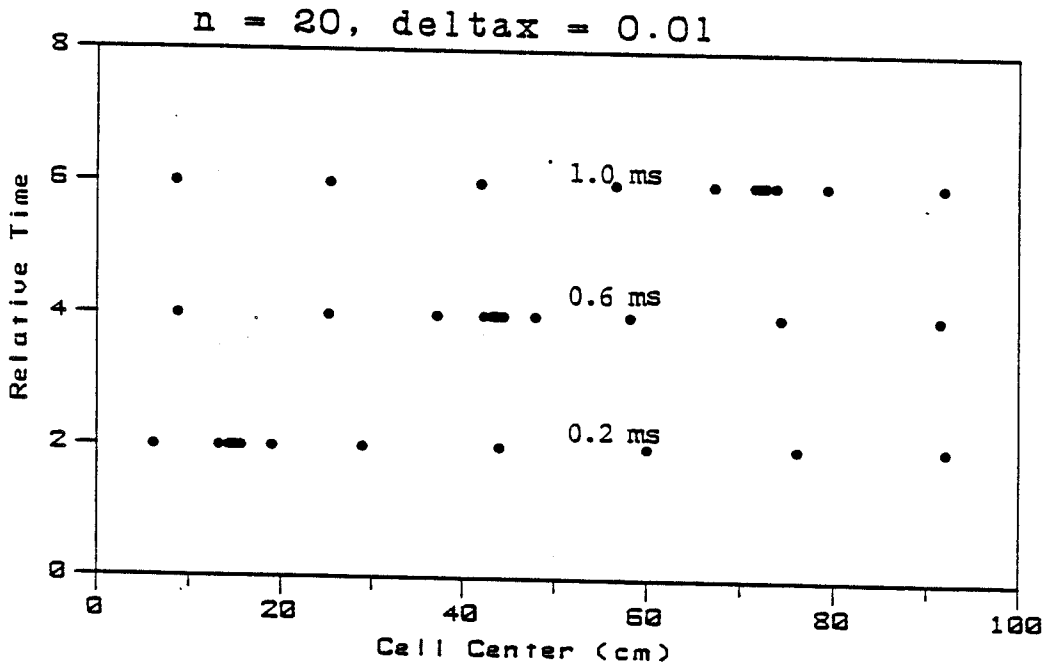

Figure 4.6 Mesh locations for 20 pt adaptive grid

would have to be introduced at the left boundary to maintain the constant zone mass; the mesh would exhibit an accordion-like behavior.

4.2 Shock Tube

The second test problem was a one-dimensional shock tube simulation. In this problem, a membrane at the centerline separates a polytropic gas into two regions of unequal pressure and density. The gas was in thermal equilibrium. At time 0+, the membrane is instantaneously broken and a shock is formed which propagates from left to right; the left region was the high pressure region. A rarefaction wave propagates from right to left as the pressure is being relieved. Between these two waves, a contact discontinuity propagates from left to right. The velocity of this discontinuity is lower than the shock speed. There are therefore three distinct regions of interest which must be captured: the shock wave, the contact discontinuity, and the rarefaction wave. This will be a severe test of the adaption technique since the gradients of the first two regions are much higher than the rarefaction fan. The wave speeds and jump relations for this problem can be found in ref. 57.

Figure 4.7 shows the initial pressure distribution for this problem; a pressure ratio of 3.0 was used for this problem. Since gas had an initial isothermal profile, the density profile would be similar to the pressure profile.

Recall from chapter 3.0 that three numerical methods were initially evaluated for the present investigation: MacCormack's method, Leonard's QUD method, and donor cell differencing. Figure 4.8 shows results for these methods; a uniform eulerian mesh was used with no grid adaption. The profiles were all obtained at a simulation time of 1.0 ms.

Figure 4.8a shows the pressure and density profiles for the MacCormack method; 990 mesh points, an artificial viscosity model coefficient of 1.75, and a fixed time step of 1.0 μs were used for this simulation. The rarefaction region, contact discontinuity, and shock region are clearly shown in this figure. We note the existence of the characteristic Gibbs phenomenon[58], the dampened oscillations, at discontinuites in the solution which is typical of second-order numerical methods. If an adaptive grid scheme was used with this method, artificial grid point clustering would occur since regions of numerically generated gradients would influence the weight function distribution. Thus points would be needlessly clustered in these regions and would reduce the mesh spacing else-

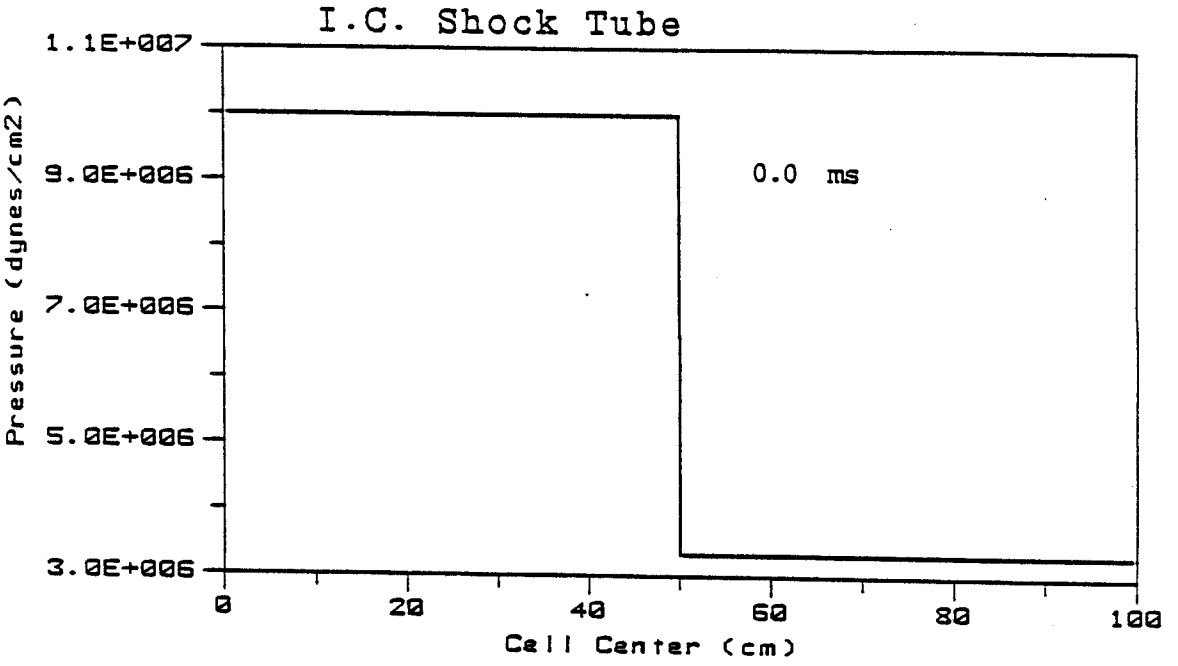

Figure 4.7 Initial pressure distribution for shock tube test problem

where in the grid. More mesh points would then be needed to adequately simulate the problem.

Figure 4.8b shows the density distribution for Leonard's QUD scheme; 500 mesh points, an artificial viscosity model coefficient of 1.0, and a CFL number of 0.1 were used for this simulation. We note that the amplitude of the oscillations at the shock location are much larger than for MacCormack's method. The same concern of artificial grid point clustering at these oscillations which has been previously discussed is also a problem with this method.

Figure 4.8c shows the density distribution for a donor cell differencing with a uniform grid of 500 mesh points; the artificial viscosity model coefficient was 1.75 and a CFL number of 0.95 was used. Note that the results are totally meaningless. One could have achieved a more realistic profile by reducing the CFL number; in essence, additional numerical artificial viscosity would be introduced and the ideal sharp profiles would now be diffuse. However, the goal of numerical simulation is to obtain results which model the physical problem which can exhibit these discontinuities. One could also have simply increased the number of mesh points to reduce the gradient across a mesh; the adaptive mesh strategy attempts to do just that, but at specified regions of the grid.

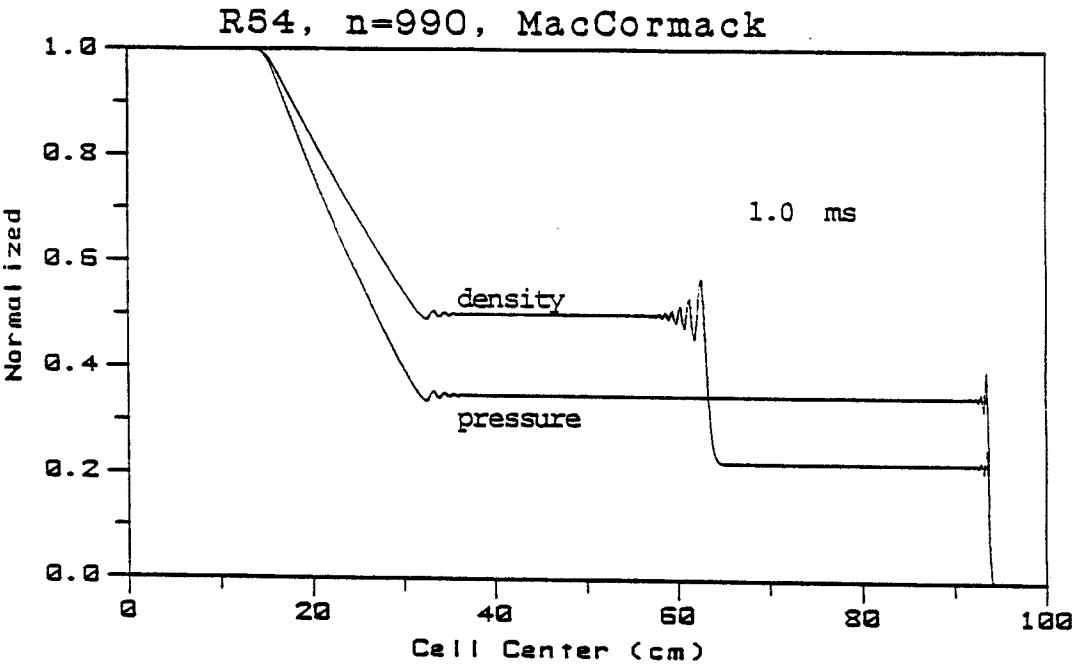

Figure 4.8a Density and pressure distributions for 990 pt MacCormack method

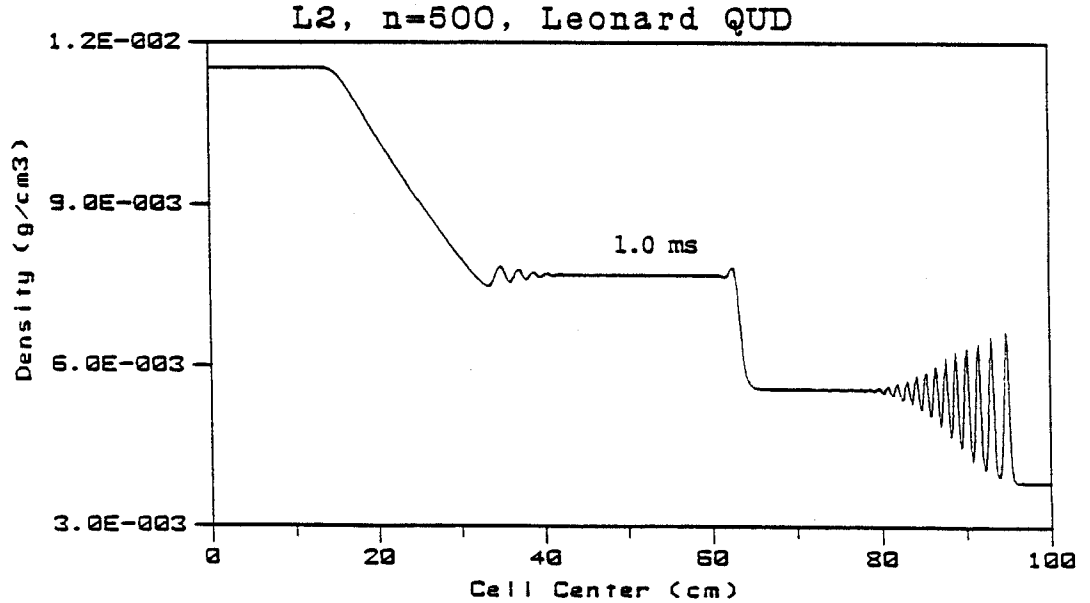

Figure 4.8b Density distribution for 500 pt Leonard method

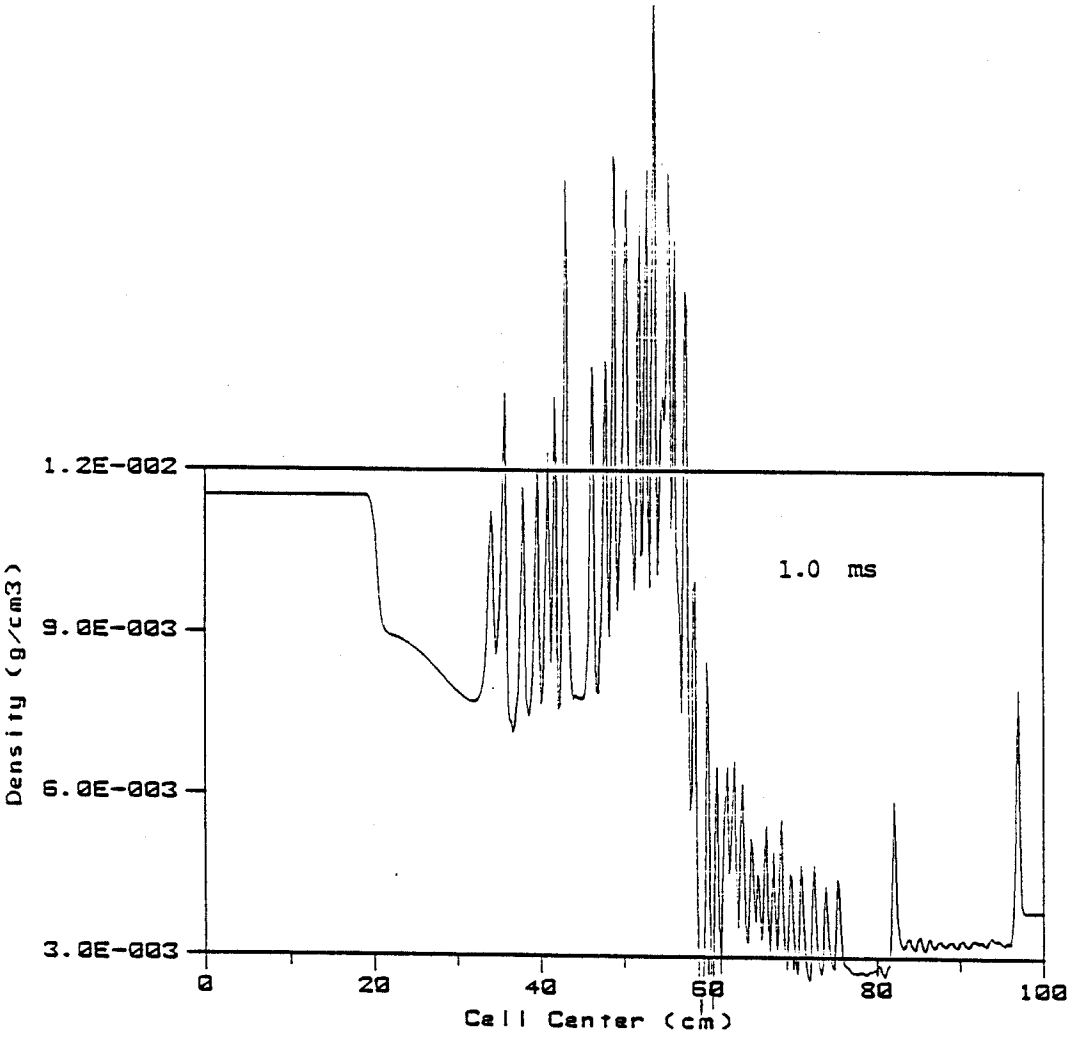

Figure 4.8c Density distribution for First Upwind Differencing with 500 pts. -- no adaption

Figures 4.9 - 4.11 show densities profiles at 1.0 ms for simulations using the adaptive mesh technique and the donor cell differencing; 50 mesh points, a CFL number of 0.95, and a minimum mesh spacing of 0.10 cm. were used for these simulations. The density and pressure were chosen as the adaption functions A and B in equation 3.9 respectively. The relative weight coefficients, R's, can be found in Table 4.1 for these calculations. The density distribution in figure 4.9 shows the result of inadequate zoning near the contact discontinuity region. Also the rarefaction fan has a distorted shape. Figure 4.10 shows the results with an increased weighting on the density distribution. We can see that the contact discontinuity region was now simulated. Finally, figure 4.11 shows the density profile from an increased weighting on the density distribution. The contact discontinuity has now been more sharply resolved; however, a computational artifact has now appeared in the rarefaction fan. This artifact has the characteristics of a zeroth-order numerical error: the error persists when the mesh size is reduced[60]. This behavior was observed when the mesh was 'over-adapted'; that is, when the grid point clustering was tightly coupled to the solution. Since this occurs only in regions of smooth curvature, it is speculated that the first-order donor cell method on an unequal mesh grid

Table 4.1

Relative Weight Coefficients for the Shock Tube

Run	$R_{\alpha a}$	$R_{\beta a}$	$R_{\alpha b}$	$R_{\beta b}$
R45	0.2	0.1	0.2	0.1
R46	0.4	0.0	0.2	0.0
R49	0.5	0.1	0.3	0.1
R204	0.5	0.15		
R207	0.35	0.15		

Table 4.2

Relative Weight Coefficients for the Blast Wave

Run	$R_{\alpha a}$	$R_{\beta a}$	$R_{\alpha b}$	$R_{\beta b}$
Rx20	0.20	0.1	0.3	0.1
Rx22	0.15	0.1	0.5	0.1
Rx23	0.15	0.1	0.4	0.1

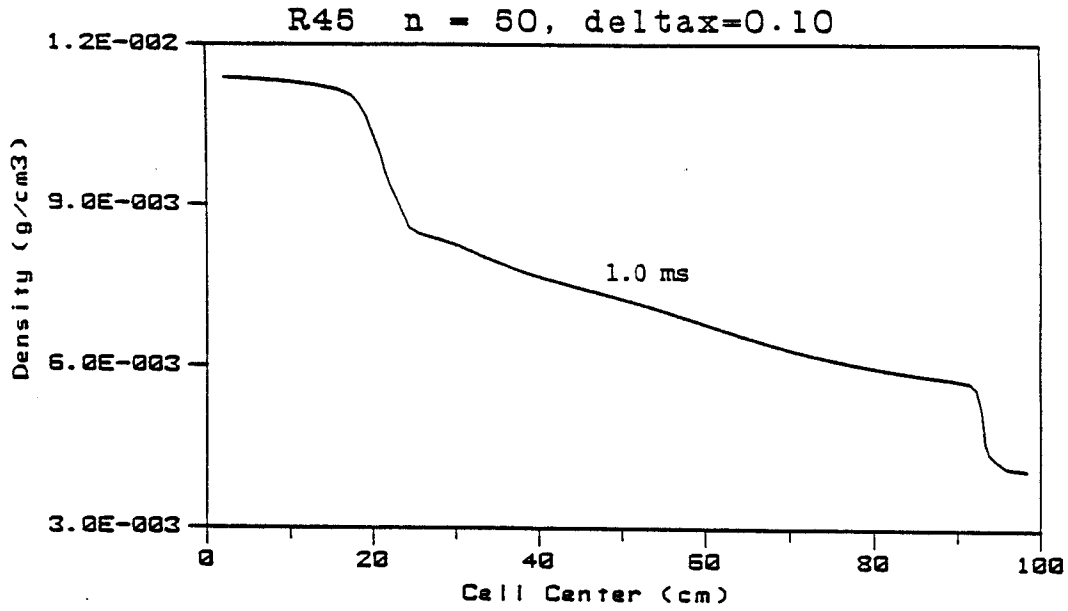

Figure 4.9 Density distribution for 50 pt adaptive mesh

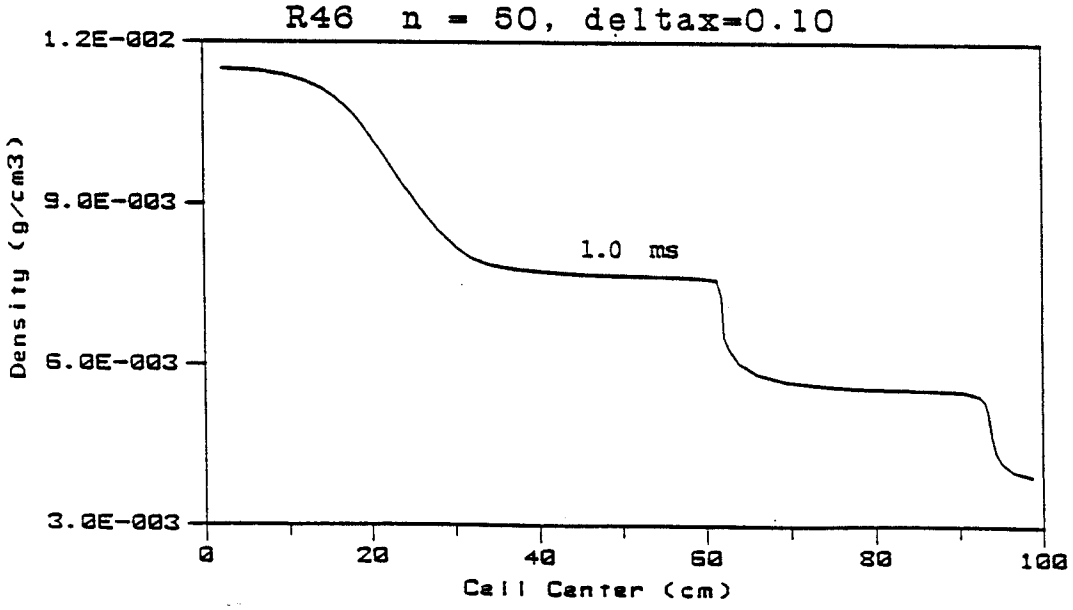

Figure 4.10 Density distribution for 50 pt adaptive mesh

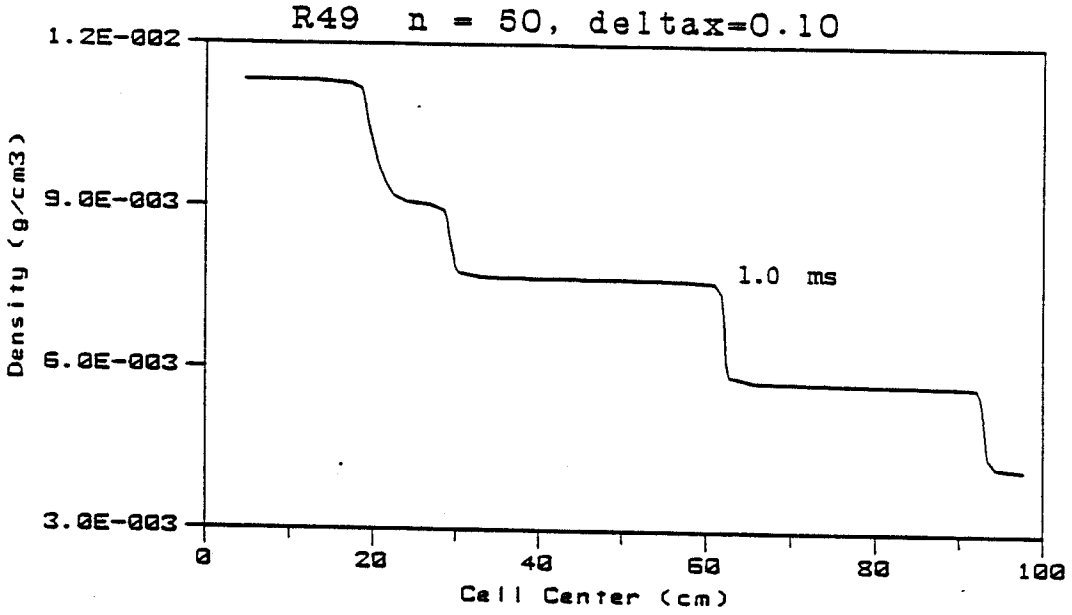

Figure 4.11 Density distribution for 50 pt adaptive mesh

contributed to this error.

The remaining figures, 4.12 - 4.15, were obtained using only the momentum density as the single weight variable. This function was chosen since its derivatives capture not only the shock front and rarefaction fan, but also the contact discontinuity. It appears to be a natural weight variable for flow driven problems. The relative weight coefficients for these calculations can be found in Table 4.1.

Figures 4.12a and 4.12b show the pressure and density distributions, respectively, for a simulation using 50 mesh points, a minimum mesh spacing of 0.1 cm, a CFL number of 1.0, and an artificial viscosity coefficient of 1.0. Computational times of 0.4 and 1.0 ms are shown. We note that the artifact seen in the previous simulation does not appear. The pressure and density profiles at the shock front compare favorably with the MacCormack simulation, figure 4.8a. However, the rarefaction fan is more diffuse in the donor cell calculations. Also the contact discontinuity profile has diffused from its sharp profile in time.

Figures 4.13a and 4.13b show the same profiles as the previous simulation, but using 100 mesh points, slightly different relative weights, and a minimum mesh spacing of 0.05 cm. We see that the density profiles are now quite

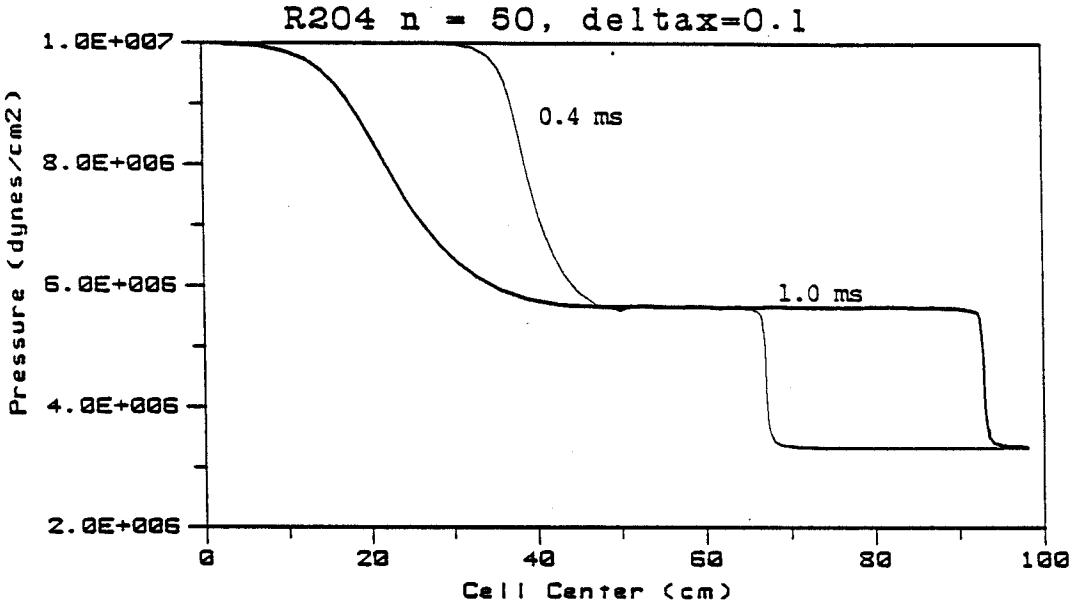

Figure 4.12a Pressure distributions for 50 pt adaptive mesh

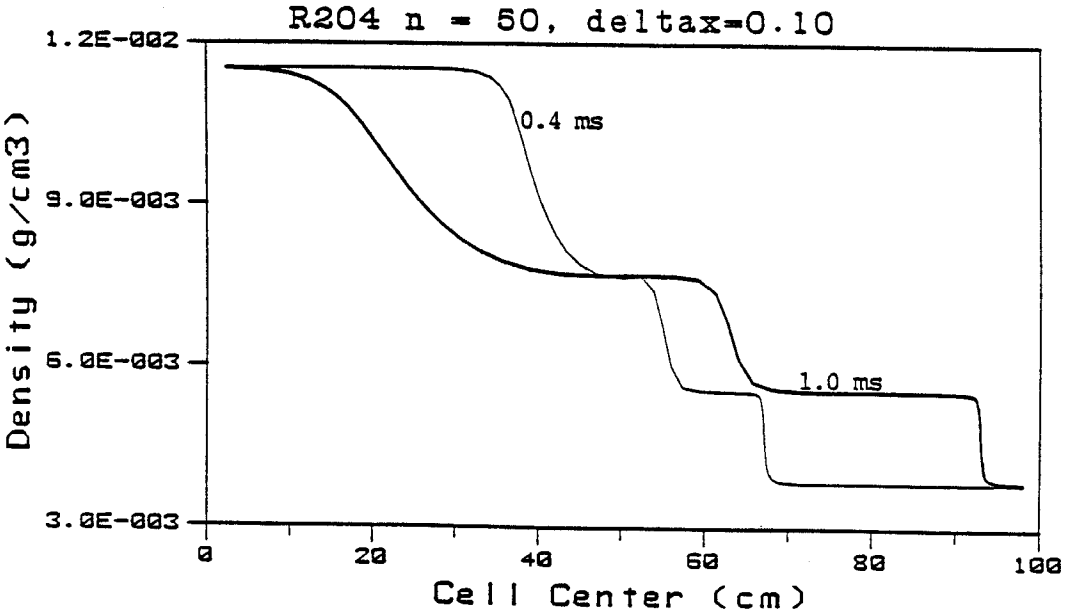

Figure 4.12b Density distributions for 50 pt adaptive mesh

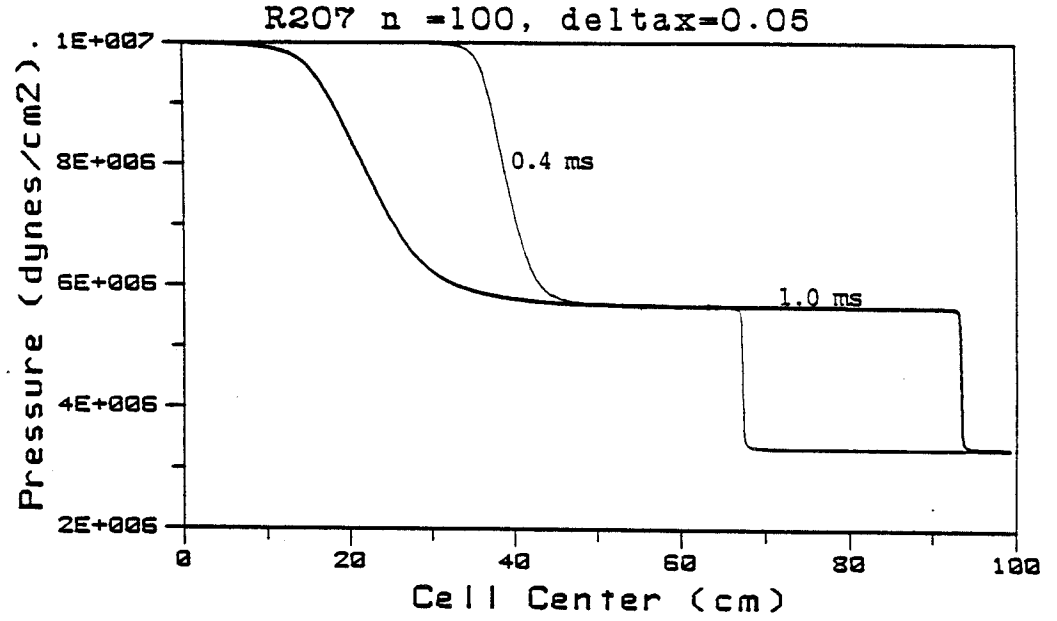

Figure 4.13a Pressure distributions for 100 pt adaptive mesh

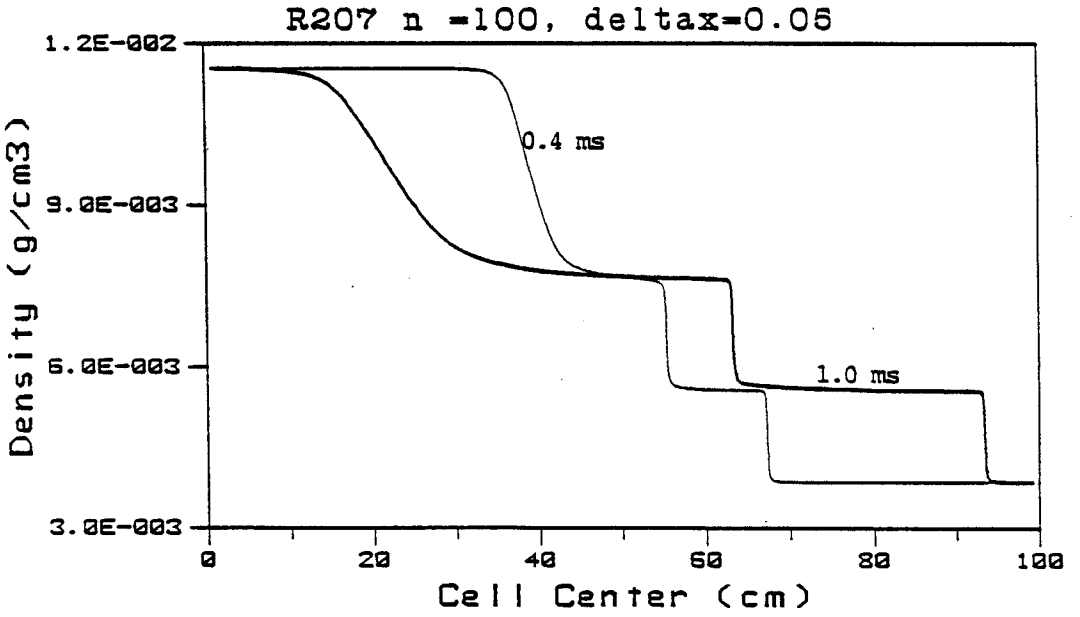

Figure 4.13b Density distributions for 100 pt adaptive mesh

sharp for both the shock and the contact discontinuity. The rarefaction fan still exhibits a more diffuse profile than the MacCormack simulation; this is probably due to the difference in a first and second order method. Figure 4.13c shows the velocity profiles for this simulation. We can see that the shock front is steeply defined, with only a very small precursor diffusion zone.

Figure 4.14 shows the CFL number distribution for the simulation generating the results shown in figure 4.13; the only time shown is for 1.0 ms. We see that the CFL number is unity at the shock front. Thus we would expect little numerical diffusion in this region. However, it is only about half of that at the contact discontinuity; therefore, numerical diffusion would tend to smear this discontinuity. The CFL number in the rarefaction fan region is only about 0.1. We would therefore expect that the increased numerical diffusion in this region would contribute greatly to the smeared results. It is postulated that a local mesh anti-diffusion technique[59] might alleviate this problem.

Finally figure 4.15 shows three mesh point distributions for the last simulation. The three regions can be easily seen from the increased point clustering. We note that the grid has an essentially uniform distribution with three finer zoned regions superimposed on it.

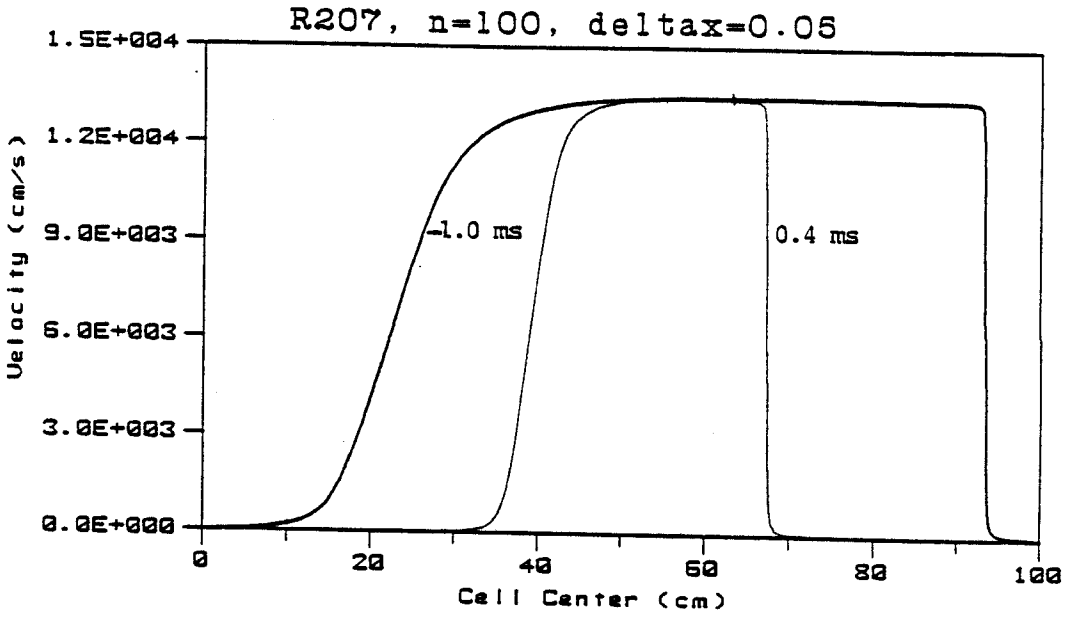

Figure 4.13c Velocity distributions for 100 pt adaptive mesh

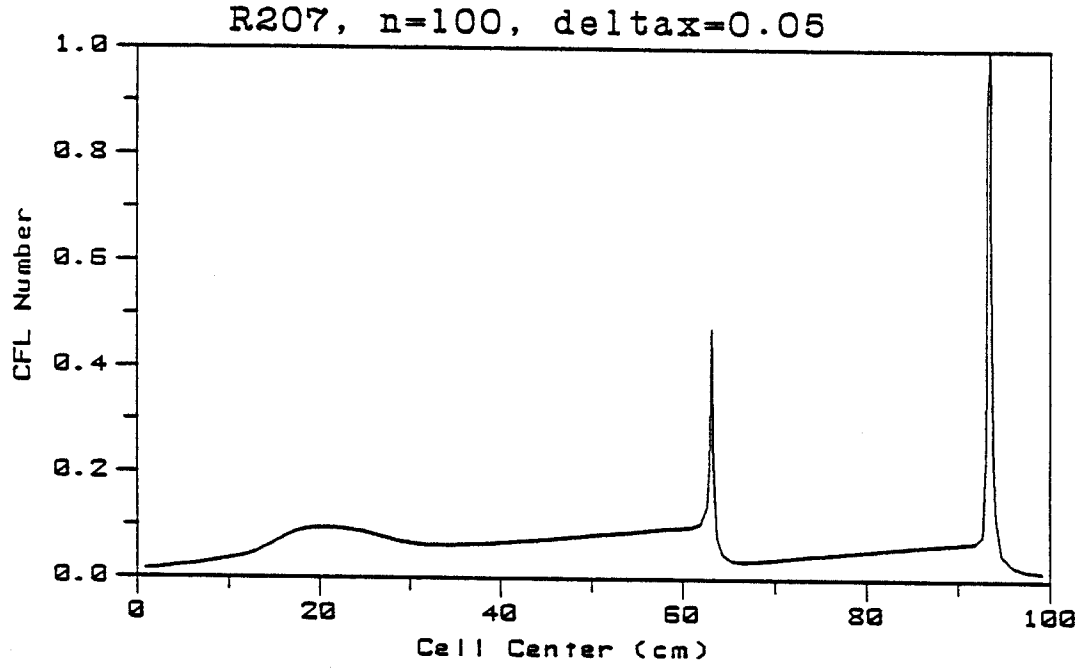

Figure 4.14 Local CFL number for 100 pt adaptive mesh

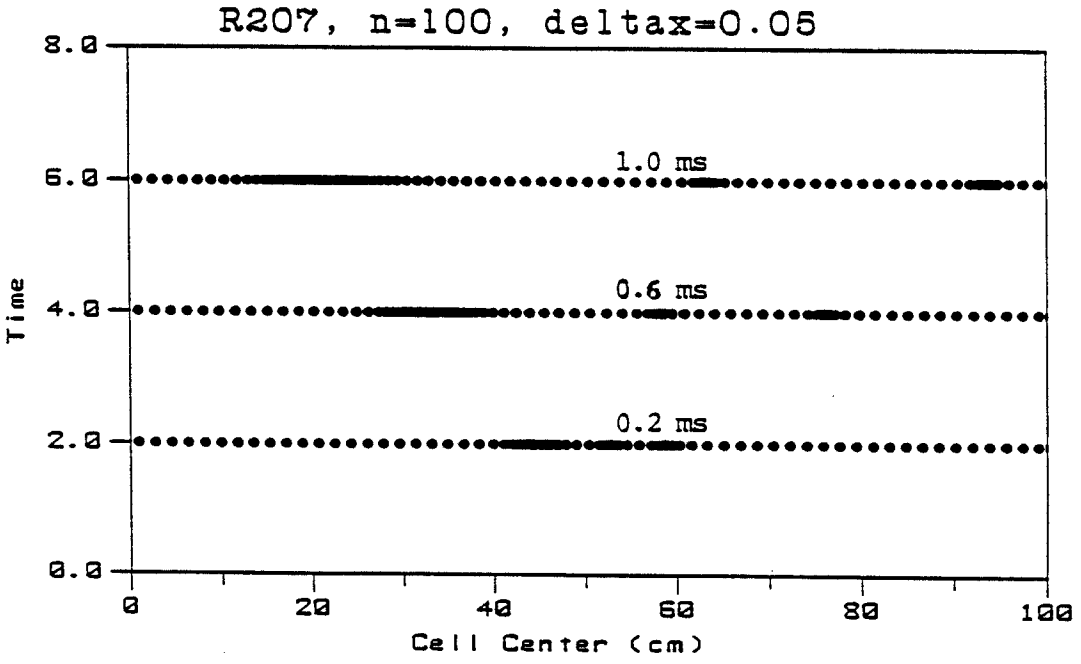

Figure 4.15 Mesh point locations for 100 pt adaptive mesh

4.3 Interacting Blast Waves

The third test problem presents a very severe test of an eulerian fluid dynamics computer code. Figure 4.16 shows the initial pressure distribution for this problem; the steep pressure ratio of 4 and 5 orders of magnitude is extremely high. Eulerian methods have difficulties with pressure ratios of approximately 3 or 4 times, not orders of magnitude. The initial density is uniform throughout the mesh; therefore, the temperature distribution exhibits a similar behavior as the pressure profile, since a polytropic gas was used. No-flow boundaries were used. The physical interpretation of this problem is a very intense energy deposition into two regions of a gas with a uniform density. There are no analytical solutions to this problem[42]. This high pressure gradient can cause negative internal energies to be obtained using an explicit numerical procedure if the CFL number is too high: a CFL number of between 0.2 and 0.3 was used in the three simulations which are presented here. The stability of this problem is determined by the pressure gradient; the CFL number is a stability requirement for convectively dominated problems. Comparison simulations without the grid adaption algorithm could not be made since computationally unstable results were obtained with 1000 mesh points and a CFL number of 0.1.

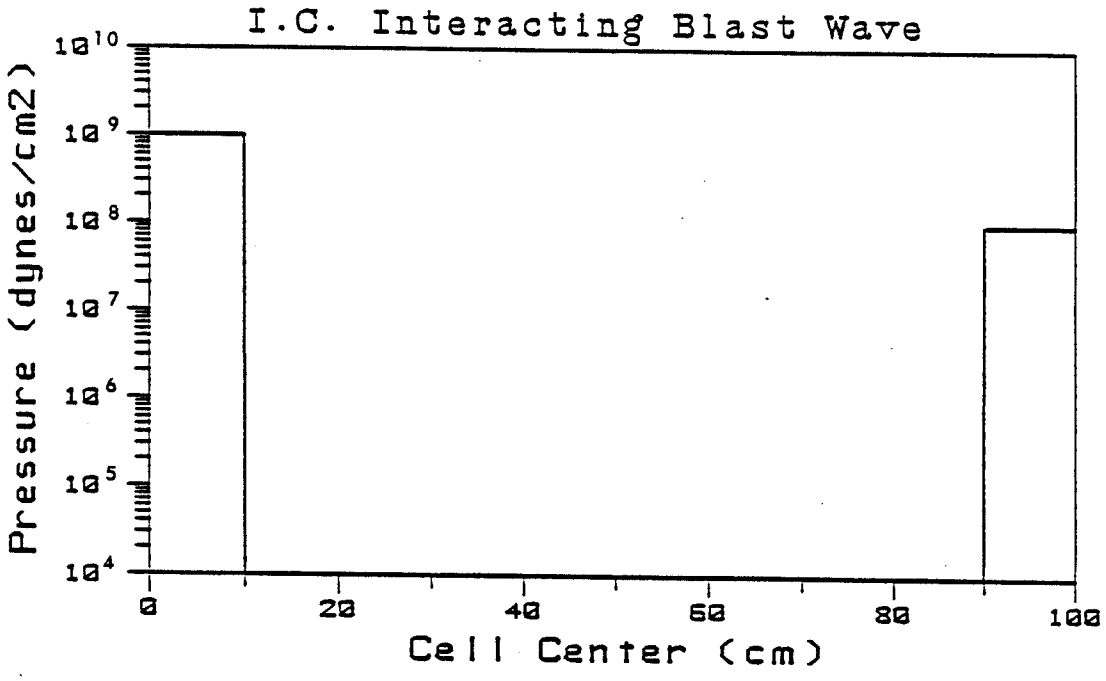

Figure 4.16 Initial pressure distribution for the interacting blast wave test problem

Figure 4.17 shows the pressure distributions for three simulation times: 2.5, 3.0, and 4.0 ms. These correspond to a time just prior to the interaction of both waves, the merging of the two waves, and the post-interaction expansion region. This simulation used 200 mesh points, a CFL number of 0.3, and a minimum mesh spacing of 0.05. The momentum density and pressure were used as the two adaption functions; the relative weight coefficients can be found in Table 4.2. Since this problem was driven by the pressure distribution, the pressure weighting was larger than the momentum density. Figure 4.18 shows a similar simulation but where the weighting of the pressure profile was increased. The same computational artifact which was seen in the shock tube problem is evident here in the pressure distribution at time 4.0 ms. It is postulated that the same reasons also apply for this problem.

Figure 4.19 shows the same pressure distributions for a simulation which now had 500 mesh points and used a CFL number of 0.2; all of the discontinuities are finely resolved. One can notice that the pressure profile at time 4.0 ms differed from that in figure 4.17; this difference is probably due to the use of a first-order method for this problem. Figures 4.20a and 4.20b show the pressure and velocity profiles at time 2.0 ms for this last simulation. We note that the pressure fronts have been

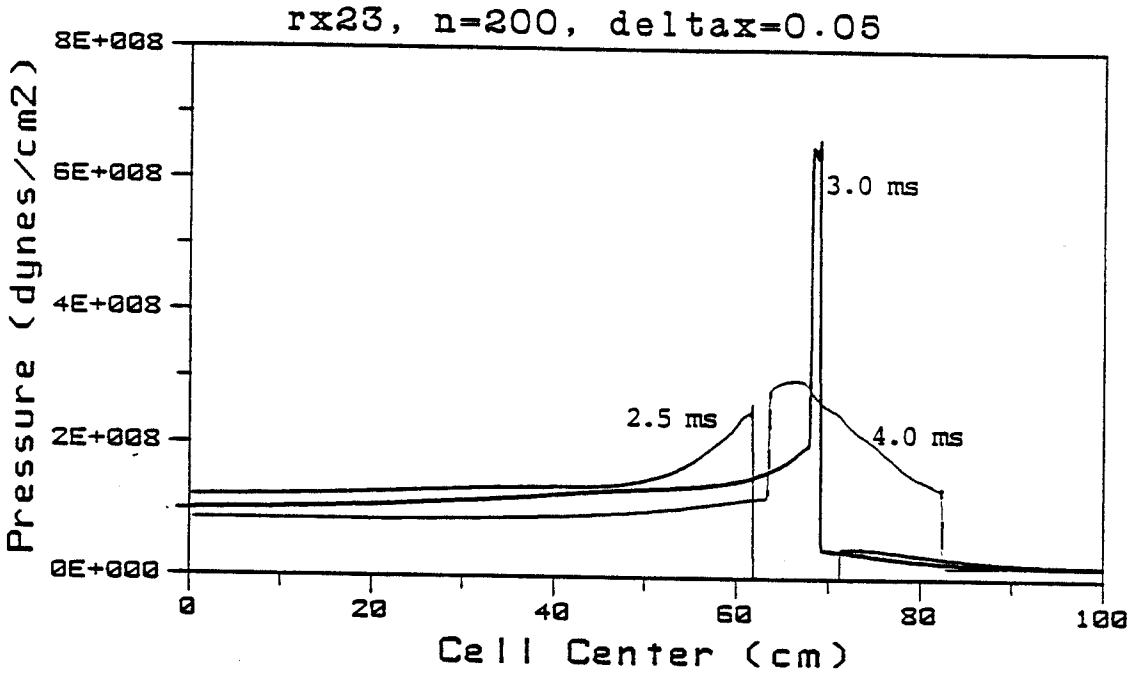

Figure 4.17 Pressure distribution for 200 pt adaptive mesh

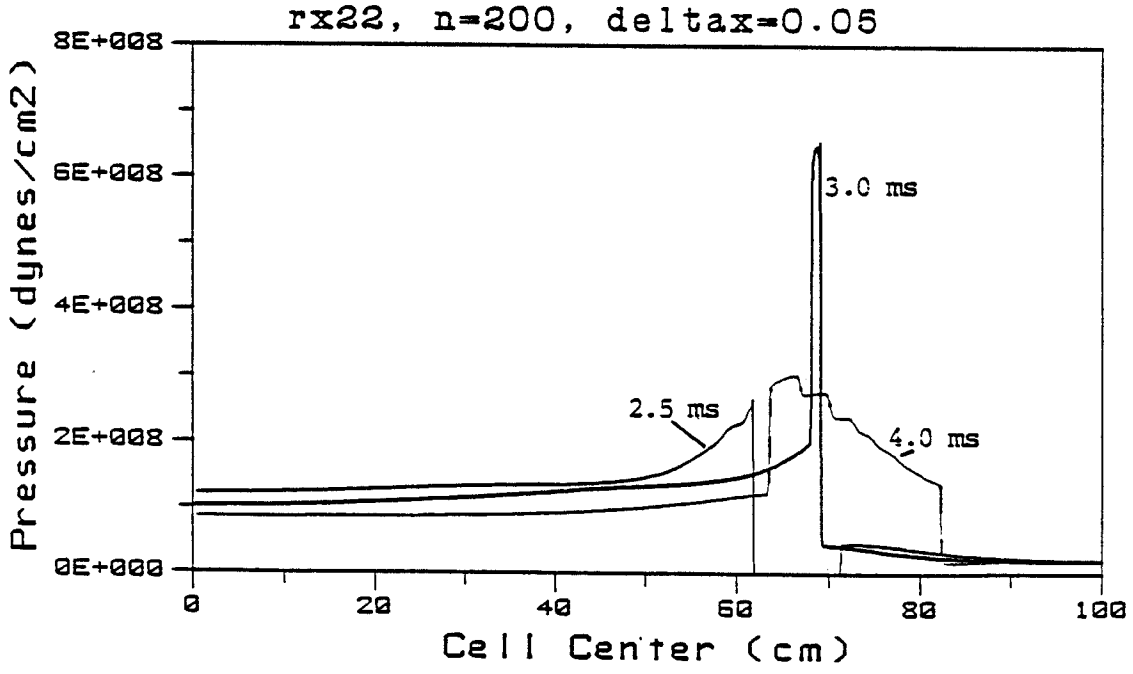

Figure 4.18 Pressure distributions for 200 pt adaptive mesh

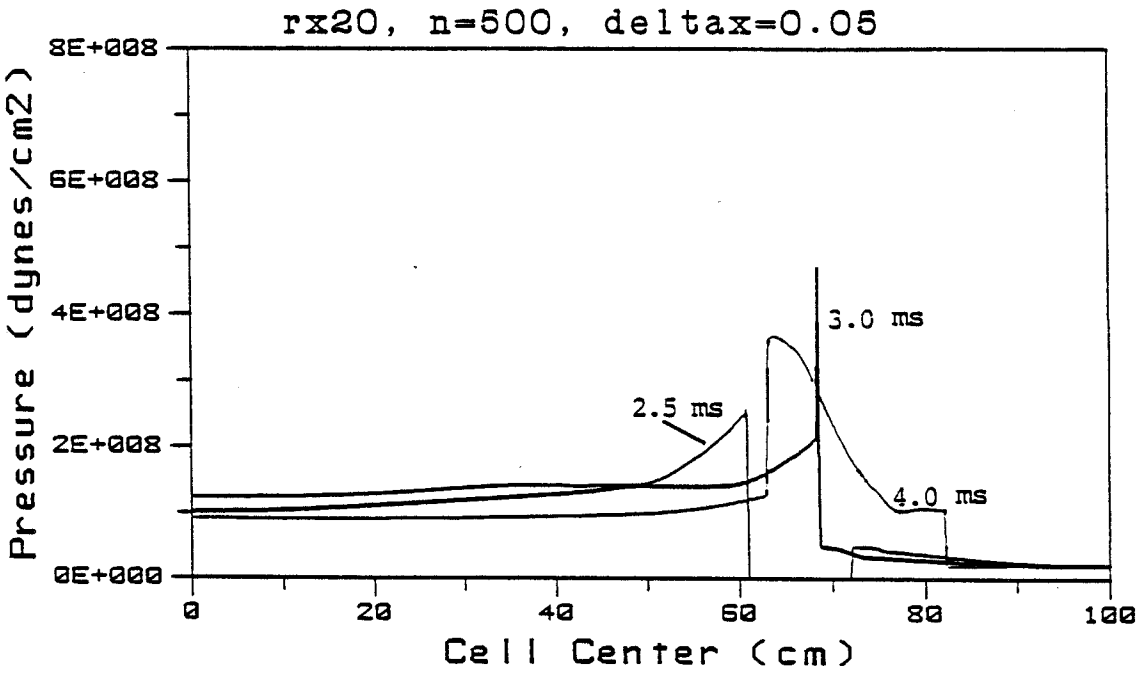

Figure 4.19 Pressure distributions for 500 pt adaptive mesh

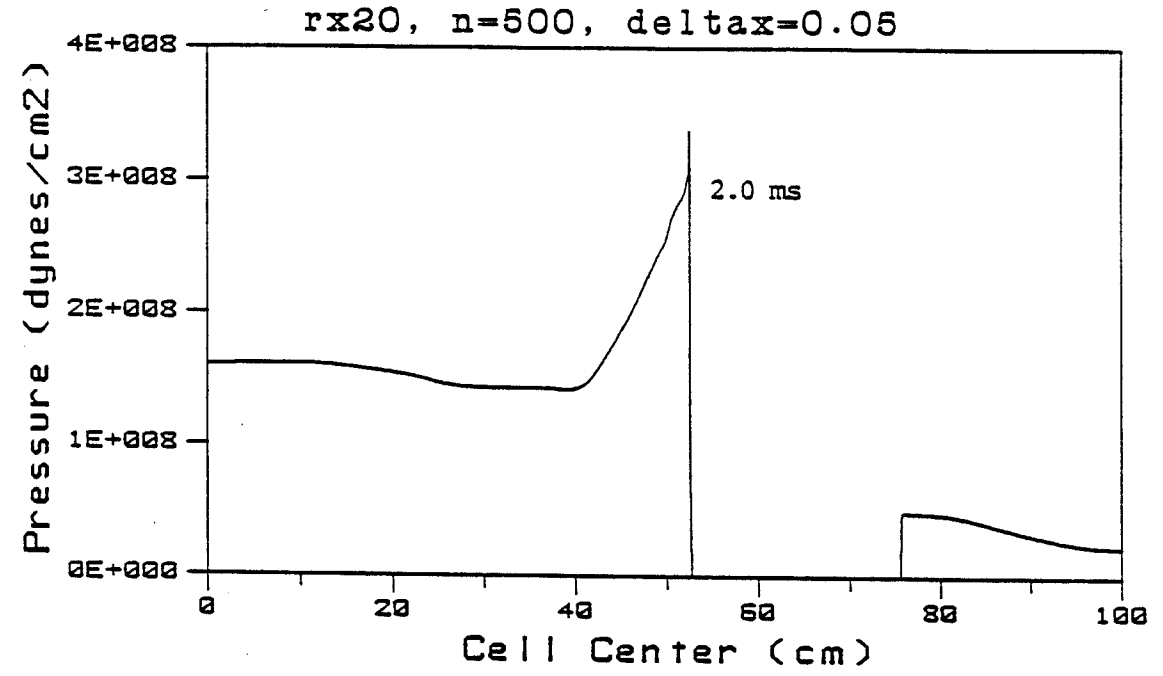

Figure 4.20a Pressure distribution for 500 pt adaptive mesh

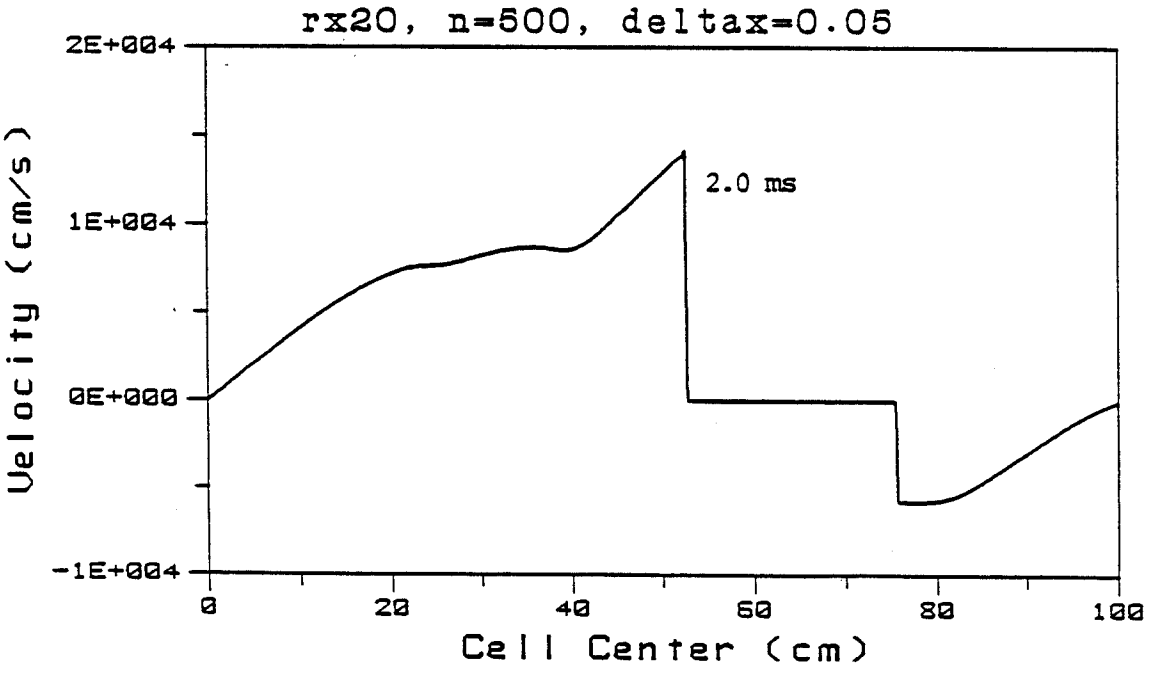

Figure 4.20b Velocity distribution for 500 pt adaptive mesh

sharply resolved, even with the extremely large pressure gradients. The corresponding velocity profile also shows this sharp front. The low frequency wave behavior in the left portion of the velocity profile is attributed to the use of a first-order method.

4.4 Radiation Hydrodynamic Blast Wave

The final test problem was a radiation hydrodynamics problem. A simple problem was run only to demonstrate the features of the adaptive mesh. The two temperature diffusion approximation used in a previous computer code[51] was incorporated into the adaptive mesh hydro code. The multifrequency modelling of the radiation field[53] would be recommended for actual numerical simulation problems.

Figure 4.21 shows results at an early time in the simulation of a plane blast wave; a cartesian grid was used. This problem used 50 mesh points. The fluid temperature and pressure were used for the two weighting functions A and B; R_{aa} and R_{ab} were both 0.25 and both R_{β}'s were 0.0. Only the first derivative was adapted. Figure 4.21a shows the fluid temperature profile and figure 4.21b the corresponding grid point locations. Since these profiles were at an early time, the velocity gradients would not have developed and a lagrangian mesh would not have the mesh point clustering.

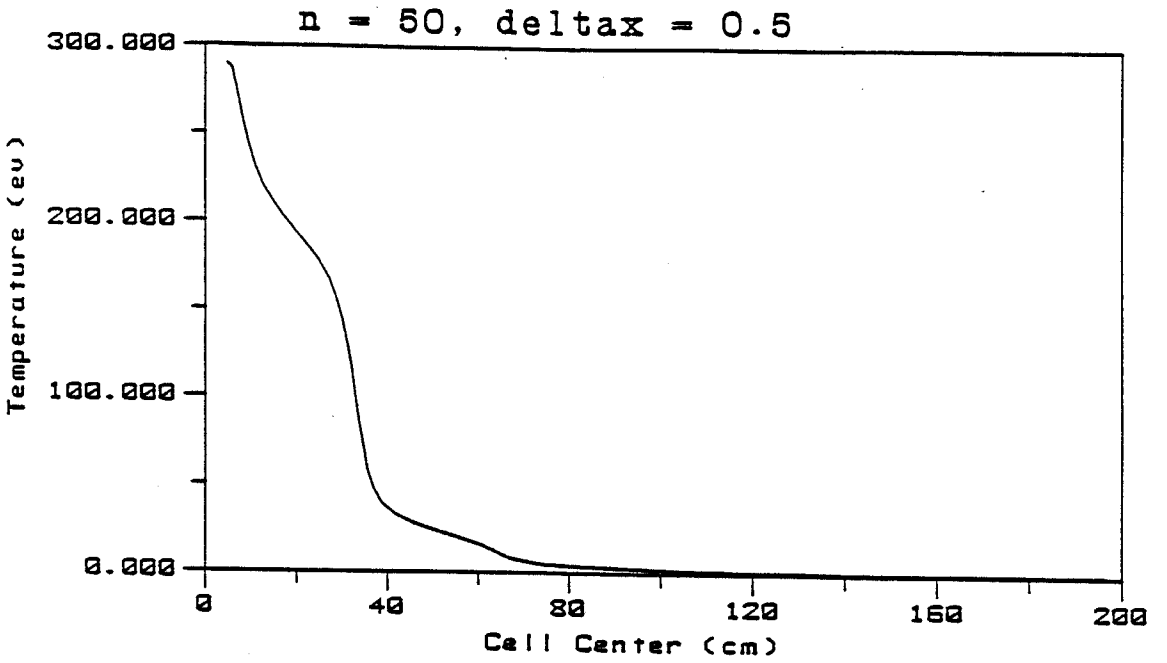

Figure 4.21a Fluid temperature for 50 pt adaptive mesh

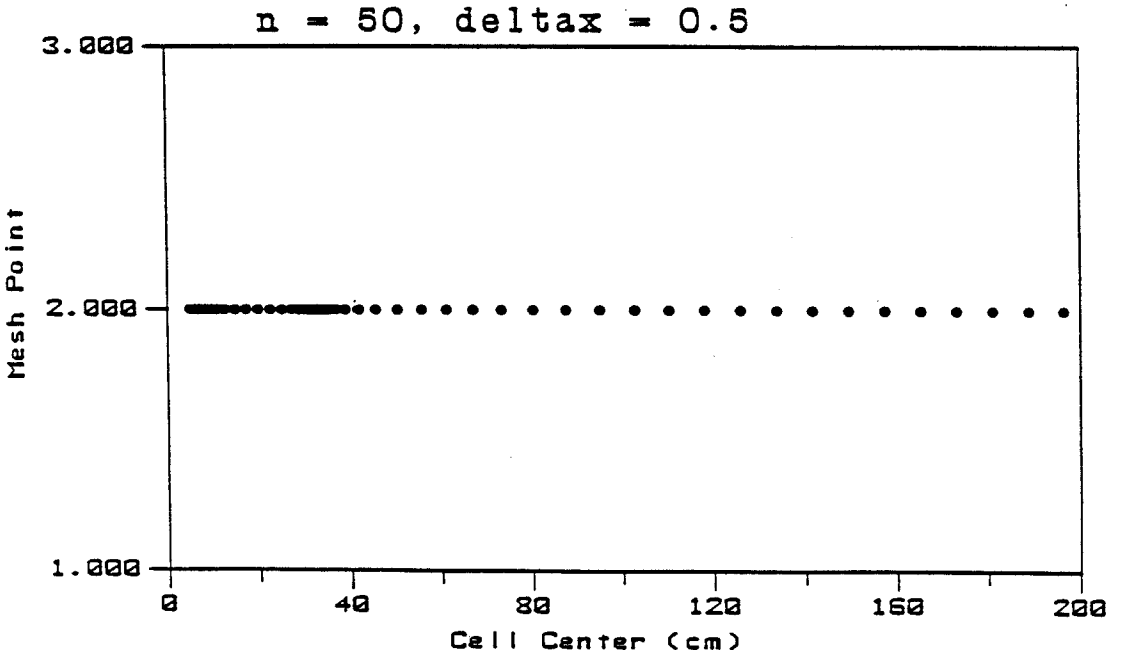

Figure 4.21b Mesh point locations for 50 pt adaptive mesh

The mesh clustering at the temperature gradients serves two purposes. The first is that the fluid opacities vary considerably in these temperature regimes; the energy coupling term in equations 3.5 and 3.6 is very dependent upon the opacities. This nonlinear term can require extremely small time steps in an explicit numerical procedure if the temperature gradient between adjacent meshes is large. The adaptive mesh grid distribution reduces the temperature gradient by moving many mesh points to the high gradient region. Therefore, the dominance of this nonlinear term upon the time step is reduced. The second benefit of the adaptive mesh is to cluster points at the edge of the blast wave. Margolin has indicated the importance of resolving the precursor temperature distribution on the wave front velocity[3]. By adaption on the temperature, this can be easily done. It is impossible to cluster mesh points in this region with a lagrangian mesh system.

5.0 CONCLUSIONS AND RECOMMENDATIONS

The following conclusions and recommendations can be made concerning this numerical investigation.

1. The numerical simulation of several different and difficult unsteady problems with an adaptive mesh generation technique was successful. These problems would have required many more mesh points and smaller time steps if simulated with an eulerian mesh. The adaptive mesh increases the accuracy of problems which have many sharp gradient regions.

2. The use of an adaptive mesh has removed the difficulty of defining an initial grid point distribution. This point distribution is very problem dependent and its implementation is almost an art form rather than a science. However, with the adaptive logic, the investigator need only decide on the number of grid points to use, the variables to adapt upon, and their relative weights. These are much easier than determining the mesh point distribution.

3. The explicit adaption method proved to be very robust. No code 'dials' or 'fudge factors' were required for the test problems presented. This procedure was also completely vectorizable so that its impact on execution time would be minimal.

4. Modifying the weight distribution to limit the mesh spacing and to smooth the mesh rather than altering the mesh directly was shown to be not only simple but to provide good results. The approximate control of the grid spacing is offset by the simple and robust nature of the method.

5. The ability to easily change the weight function variables or combinations of variables was very useful. For flow driven problems such as the shock wave and shock tube, the use of the momentum density allowed a variety of gradients regions to be resolved: a shock wave, a contact discontinuity, and a rarefaction fan. Simulations using any single primitive variable (i.e. pressure or density) as the weight function did not adequately capture all of these gradient regions. For pressure or temperature driven problems, the ability to use a simple combination of variables proved to be

very useful. The interacting blast wave problem is dominated by the pressure gradients; the use of pressure as a weight function is obvious. The addition of the momentum density in the blast wave weight function helped to better resolve the rarefaction regions. For the radiation hydrodynamics problem, the fluid temperature and momentum density were used. One could easily postulate problems where other variables would be important; the weight function could easily be extended to any number of variables.

6. The difference equations for the governing partial differential equations in the transformed plane were successfully cast in a strong conservative form. However, the highly nonlinear energy transfer term, Ω, in the radiation hydrodynamics equations could not be put in conservative form.

7. The use of a control volume approach rather than the more prevalent differential approach to develop the difference equations is the only feasible method since special attention has to be given to the different velocities: the fluid cell center velocity, the fluid mesh edge velocity, and the

physical mesh edge velocity.

8. The simple donor cell differencing method was very successful in capturing strong gradient regions, i.e., shocks and contact discontinuities, but was only acceptable in mild gradient regions, i.e., the rarefaction fan. Conditions developed when the mesh adaption was strongly coupled to the solution where zeroth-order errors developed in mild gradient regions. It is speculated that this is due to the use of a first-order method with a highly unequally spaced mesh in a mild gradient region. However, reducing the coupling between the grid adaption and the solution reduced this error.

9. The adaptive grid CFL number used in this investigation (equation 3.22) was successful in predicting the maximum stable explicit time step for flow or velocity driven problems; the blast wave was a pressure driven problem.

10. The mild gradient regions in the shock tube and blast wave problems illustrate the need for a second-order numerical method. However, the popular and robust MacCormack method is not

amenable to an adaptive mesh since it is a two-step method. Leonard's QUD method was not capable of simulating problems with high grid Peclet numbers or shock regions. Further work in this area is warranted.

A P P E N D I C E S

A P P E N D I X A

Coordinate Transformations

Coordinate Transformations

First we consider simple chain rule applications for the transformation of a function in (x,t) to (ξ,τ):

$$f_x = f_\xi \, \xi_x = \frac{f_\xi}{x_\xi}$$

$$f_{xx} = \frac{1}{x_\xi}\left[\frac{1}{x_\xi} f_\xi\right]_\xi = \frac{1}{x_\xi^2} f_{\xi\xi} - \frac{x_{\xi\xi}}{x_\xi^3} f_\xi$$

$$f_\tau(\xi,\tau) = f_x(x,t)\, x_\tau + f_t(x,t)\, t_\tau \, .$$

Now we note that $t_\tau = 1$ (from equation 1.2); thus:

$$f_t(x,t) = f_\tau(\xi,\tau) - \frac{f_\xi(\xi,\tau)}{x_\xi(\xi,\tau)} x_\tau \, .$$

Next, consider the first-order wave equation (equation 1.1):

$$U_t + C\, U_x = 0.$$

Using the above chain rule relations we can obtain equation 1.3, in a non-conservative form, as:

$$U_\tau + \left(\frac{C - x_\tau}{x_\xi}\right) U_\xi = 0 \, .$$

We now start with the wave equation in conservative form:

$$U_t + [C\ U]_x = 0.$$

C is a constant but has been included inside of the braces to illustrate the conservative form of the equation. Transforming this equation to the (ξ,τ) system, we obtain:

$$(x_\xi U)_\tau + \left[\left(\frac{C - x_\tau}{x_\xi}\right) x_\xi U\right]_\xi = 0.$$

One can note that in the conservative form, the conserved variable has been effectively transformed from U to $(x_\xi U)$; the convective term is written with x_ξ in both the numerator and denominator to simply make this point. The conservation form of the equations were used in the present investigation.

Finally, we consider the addition of a diffusion term to the wave equation, in written in conservative form:

$$U_t + (C\ U)_x = (D\ U_x)_x$$

with D taken as a constant. Using the same transformations as before, we obtain:

$$(x_\xi U)_\tau + \left[\left(\frac{C - x_\tau}{x_\xi}\right) x_\xi U\right]_\xi = \left[\frac{D\ U_\xi}{x_\xi}\right]_\xi$$

Again, we can note that the conserved variable is simply $(x_\xi U)$. The diffusion term remains in a conservative form.

Since the present investigation has defined $\Delta\xi$ to be unity, the x_ξ terms are simply the mesh spacing, Δx.

A P P E N D I X B

Difference Equations

Difference Equations

Only the difference equations for the momentum equation will be given here; the form of the continuity differencing equation is contained in it and the form of the energy equation is very similar.

Recall that an explicit upwind or donor cell differencing on the spacial derivatives and forward time differencing on the temporal derivative were the numerical methods chosen to integrate the PDEs from the time n level to n+1. The equation was in the strong conservative form. The adaptive mesh momentum equation, equation 3.15, is rewritten here:

$$[\rho u x_\xi]_\tau + [\rho u(u - x_\tau) + P - \mu^* u_\xi / x_\xi]_\xi = 0 \quad .(B.1)$$

An explicit adaptive scheme was used to obtain x_ξ prior to the time integration; therefore, the time level of this term should be n+1/2 and not n+1. The difference equation, in cartesian coordinates, can be written as:

$$[(\rho u)^{n+1}_{c,i}\ x^{n+1/2}_{\xi}] - [(\rho u)^{n}_{c,i}\ x^{n}_{\xi}] =$$

$$-\Delta t \Big([(\rho u)^{n}_{r}\ u^{n}_{re} - (\rho u)^{n}_{l}\ u^{n}_{le}]$$

$$+ [\ (p^{n}_{i+1} - p^{n}_{i-1})(\Delta x^{n}_{i}/dx^{n}_{i})]$$

$$- [\ (\mu^{*n}_{r}/dx^{n}_{r})(u^{n}_{i+1} - u^{n}_{i}) - (\mu^{*n}_{l}/dx^{n}_{l})(u^{n}_{i} - u^{n}_{i-1})] \Big)$$

where:

$$u^{n}_{re} = [u^{n}_{r} - x^{n}_{ri}]$$

$$u^{n}_{r} = [u^{n}_{i+1}\Delta x^{n}_{i+1} + u^{n}_{i}\Delta x^{n}_{i}]/(\Delta x^{n}_{i+1} + \Delta x^{n}_{i})$$

$$(\rho u)^{n}_{r} = \text{cvmgp}\ [\ (\rho u)^{n}_{i},\ (\rho u)^{n}_{i+1},\ u^{n}_{r}\ ]$$

$$dx^{n}_{r} = 0.5\ [\ \Delta x^{n}_{i+1} + \Delta x^{n}_{i}\ ]$$

$$dx^{n}_{l} = 0.5\ [\ \Delta x^{n}_{i} + \Delta x^{n}_{i-1}\ ]$$

$$dx^{n}_{i} = dx^{n}_{r} + dx^{n}_{l}$$

$$\mu^{*n}_{r} = 0.5\ [\ \mu_{i} + \mu_{i+1}\ ] + a$$

$$a = \rho^{n}_{i}\ (C\ \Delta x^{n}_{i})^{2}\ |dudx|$$

$$dudx = [\ u^{n}_{r} - u^{n}_{l}\ ]\ /\ \Delta x^{n}_{i}\ .$$

Note that $x^{n}_{r,i}$ is defined at the right mesh edge velocity and the subscripts r and l are the right and left mesh edge values respectively.

A P P E N D I X C

Weight Function Derivatives

Weight Function Derivatives

Recall that the weight function for the present investigation is of the form:

$$W = 1 + \alpha_a |A_x| + \beta_a |A_{xx}| + \alpha_b |B_x| + \beta_b |B_{xx}| .$$

Simple centered differences were used to determine the derivatives A_x and A_{xx} and the corresponding Bs. Since these derivatives are only used to determine the local cell weights and not to advance the governing PDEs, stability considerations are not a problem. Each derivative was scaled by the maximum absolute value of the particular function. This scaling was required to put the derivative of the two functions A and B on an equivalent domain. The function values are cell centered.

The difference equations for A_x and A_{xx} are:

$$A_{x,i} = 0.5 \frac{1}{A_{max}} \left(\frac{A_i - A_{i-1}}{x_i - x_{i-1}} + \frac{A_{i+1} - A_i}{x_{i+1} - x_i} \right)$$

$$A_{xx,i} = \frac{1}{A_{max}} \left\{ \left(\frac{\dfrac{A_{i+1} - A_i}{x_{i+1} - x_i} - \dfrac{A_i - A_{i-1}}{x_i - x_{i-1}}}{\Delta x_i} \right) \right\}$$

where A_{max} is the maximum absolute value of the As,
x_i is the cell center location for mesh i, and
Δx_i is the ith mesh spacing.

R E F E R E N C E S

R E F E R E N C E S

1. Zel'dovich, Ya. B. and Yu. P. Raizer, Physics of Shock Waves and High-Temperature Hydrodynamic Phenomena, Academic Press 1966.

2. Decoste, R., S. E. Bodner, et.al., "Ablative Acceleration of Laser-Irradiated Thin-Foil Targets", Phys. Rev. Letters, vol.42, no.25, 1979.

3. Margolin. L. G., Private Communication, Livermore National Lab, 1985.

4. Duderstadt, J. J. and G. A. Moses, Inertial Confinement Fusion, John Wiley and Sons, 1982.

5. Ripin, B. H., S. E. Bodner, et.al., "Laser-Ablative Acceleration of Targets to Near Inertial Fusion Conditions", NRL MR 4916, 1982.

6. Moses, G. A., Private Communication, University of Wisconsin-Madison, 1985.

7. Powers, L. V., D.J. Tanner and R. A. Grandey, "Resolution of Critical-Surface Density Profiles by Automatic Fine Zoning", APS meeting no. 21, 1979.

8. Emery, M. H., J. H. Gardner, et.al., "Vortex Shedding and Laser Ablation", NRL MR-5089, 1983.

9. Lax, P. D., "Nonlinear Partial Differential Equations and Computing", SIAM Review, Vol.11, No.1, Jan. 1969, 1-19.

10. Vinokur, M., "Conservation Equations of Gasdynamics in Curvilinear Coordinate Systems", J. of Comp. Phys, vol.14, 105-125, 1974.

11. LASNEX, Laser Program Annual Reports, Lawrence Livermore National Lab, UCRL-500021 (1979-1985).

12. Moses, G. A., G. Magelsen, R. Israel, T. Spindler, and B. Goel, "PHD-IV A Plasma Hydro-Thermonuclear Burn-Radiative Transfer Computer Code", University of Wisconsin Nuclear Engineering Dept. UWFDM-194, revised Jan 1986.

13. Christiansen, J. P. and N. K. Winsor, "CASTOR 2: A Two-Dimensional Laser Target Code", Comp. Phy. Comm. 17, 1979, 397-412.

14. Boris, J. P., "Flux-Corrected Transport Modules for Solving Generalized Continuity Equations",NRL MR-3237, March 1976.

15. Gardner, J. H. and S. E. Bodner, "High Efficiency Targets for High Gain Inertial Confinement Fusion", NRL MR 5814, Sept. 1986.

16. Thompson, S. L., "CSQII-An Eulerian Finite Difference Program for Two-Dimensional Material Response", SAND77-1339, 1979.

17. Brackbill, J. U. and L. G. Margolin, "An Algorithm for the Computation of Nonlinear Electron Thermal Conduction on an Arbitrarily Shaped, Two-Dimensional Domain", LA-6964-MS, 1977.

18. Thompson, J. F. "A Survey of Dynamically-Adaptive Grids in the Numerical Solution of Partial Differential Equations", AIAA-84-1606, 1984.

19. Anderson, D. A., J. C. Tannehill, and R. H. Pletcher, Computational Fluid Mechanics and Heat Transfer, McGraw-Hill, 1984.

20. Thompson, J. F., Z. U. A. Warsi and C. W. Mastin, Numerical Grid Generation, North-Holland, 1985.

21. Hildebrand, F. B.,Advanced Calculus for Applications, Prentice-Hall, 1976.

22. Eiseman, P. R., "High Level Continuity for Coordinate Generation with Precise Controls", J. of Comp. Phys. 47, 352, 1982.

23. Eiseman, P. R., "Alternating Direction Adaptive Grid Generation", AIAA J., vol.23, no.4, 551-560, 1985.

24. Dwyer, H. A., "Grid Adaption for Problems in Fluid Dynamics", AIAA J. vol. 22, no. 12, 1705-1712, 1984.

25. Dwyer, H. A., "A Discussion of Some Criteria for the Use of Adaptive Gridding", in Adaptive Computational Methods For Partial Differential Equations, I. Babuska ed., SIAM, 1983.

26. Dwyer, H. A., "Grid Adaption for Problems with Separation, Cell Reynolds Number, Shock-Boundary Layer Interaction, and Accuracy", AIAA 83-0449, 1983.

27. Anderson, D. A., "Adaptive Mesh Schemes Based on Grid Speeds", AIAA 83-1931, 1983.

28. Anderson, D. A., "Application of Adaptive Grids to Transient Problems", in Adaptive Computational Methods for Partial Differential Equations, I. Babuska ed., SIAM, 1983.

29. Anderson, D. A. and M. M. Rai, "The Use of Solution Adaptive Grids in Solving Partial Differential Systems", in Numerical Grid Generation, J. F. Thompson ed., Elsevier Science, 1982.

30. Winkler, K. H. A., D. Mihalas and M. L. Norman, "Adaptive-Grid Methods with Asymmetric Time-Filtering", Computer Physics Comm. 36(1985), 121-140.

31. Winkler, K. H. A., M. L. Norman and M. J. Newman, "Adaptive Mesh Techniques for Fronts in Star Formation", Physica 12D(1984), 408-425.

32. Dwyer, H. A., Private Communication, University of California-Davis, 1985.

33. Brackbill, J. U., "Coordinate System Control: Adaptive Meshes", in Numerical Grid Generation, J. F. Thompson ed., Elsevier Science, 1982.

34. Oberkampf, W. L., Private Communication, Sandia National Labs-Albuquerque, 1985.

35. Roache, P. J., "Computational Fluid Dynamics", Hermosa, 1972.

36. Peyret, R. and H. Viviand, "Computation of Viscous Compressible Flows Based on the Navier-Stokes Equations", AGARD-AG-212, 1975.

37. Dwyer, H. A., Private Communication, University of California-Davis, 1985.

38. MacCormack, R. W., "The Effect of Viscosity in Hypervelocity Impact Cratering", AIAA-69-354, 1969.

39. MacCormack, R. W., "An Efficient Numerical Method for Solving the Time-Dependent Compressible Navier-Stokes Equation at High Reynolds", NASA TM X-73,129, 1976.

40. MacCormack, R. W., "A Numerical Method for Solving the Equations fo Compressible Viscous Flow", AIAA J., vol. 20, no. 9, 1275-1281, 1982.

41. MacCormack, R. W., "Current Status of Numerical Solutions of the Navier-Stokes Equations", AIAA-85-0032, 1985.

42. Woodward, P. and P. Colella, "The Numerical Simulation of Two-Dimensional Flow with Strong Shocks", J. of Comp. Phys., 54, 115-173, 1984.

43. Colella, P. and P. Woodward, "The Piecewise Parabolic Method (PPM) for Gas-Dynamical Simulations", J. of Comp. Phys., 54, 174-201, 1984.

44. Shang, J. S., P. G. Buning, et.al., "Performance of a Vectorized Three-Dimensional Navier-Stokes Code on the Cray-1 Computer", AIAA J. vol. 18, no. 9, 1073-1079, 1980.

45. Lax, P. D. and B. Wendroff, "Systems of Conservation Laws", Comm. Pure Appl. Math., vol.13, 217-237, 1960.

46. Blottner, F. G., Private Communication, Sandia National Labs-Albuquerque, 1985.

47. Leonard, B. P., "A Stable and Accurate Convective Modelling Procedure Based on Quadratic Upstream Interpolation", Comp. Meth. in Applied Mech. and Engr., vol.19, 69-98, 1979.

48. Schlichting, H., Boundary Layer Theory, Mcgraw Hill, 1968.

49. Pulliam, T. H., "Artificial Dissipation Models for the Euler Equations", AIAA Journal vol.24, no.12, Dec. 1986, 1931-1940.

50. Richtmyer, R. D. and K. W. Morton, Difference Methods for Initial-Value Problems, Interscience, 1967.

51. Bartel, T. J., R. R. Peterson, and G. A. Moses, "Numerical Simulations of a Stratified Gas ICF Cavity", UWFDM-679, April 1986.

52. Pomraning, G. C., "Radiation Hydrodynamics", LA-UR-82-2625.

53. Moses, G. A., T. J. McCarville, and R. R. Peterson, "MF-FIRE, A Multifrequency Radiative Transfer Hydrodynamics Code", Comp. Phys. Comm. 36, 249 (1985).

54. Dwyer, H. A., M. D. Smooke and R. J. Kee,"Adaptive Gridding for Finite Differnece Solutions to Heat and Mass Transfer Problems", in Numerical Grid Generation, J. F. Thompson ed., Elsevier Science, 1982.

55. Pollard, A. and A. L. Siu, "The Calculation of Some Laminar Flows Using Various Discretisation Schemes", Computer Methods in Applied Mechanics and Engineering 35 (1982), 293-313.

56. Patel. M. K. and N. C. Markatos, "An Evaluation of Eight Discretization Schemes for Two-Dimensional Convection-Diffusion Equations", Int. J. for Numerical Methods in Fluids, vol. 6, 1986, 129-154.

57. Harlow, F. H. and A. A. Amsden, "Fluid Dynamics", LA-4700, June 1971.

58. Churchill, R. V., Fourier Series and Boundary Value Problems, McGraw-Hill, 1941.

59. Rivard, W. C., O. A. Farmer, T. D. Butler and P. J. O'Rourke, "A Method for Increased Accuracy in Eulerian Fluid Dynamics Calculations", LA-5426-MS, October 1973.

60. Margolin, L. G., H. M. Ruppel, and R. B. Demuth, "Gradient Scaling for Nonuniform Meshes", 4th Int. Conf. on Numerical Methods in Laminar and Turbulent Flow July 9-12, 1985.