...ed States Department of Energy ...

TITLE NUMERICAL METHODOLOGIES FOR SOLVING PARTIAL DIFFERENTIAL EQUATIONS

LA-UR--89-3136

DE90 001842

AUTHOR(S) James M. Hyman

SUBMITTED TO NATO Advanced Research Workshop on the Numerical Modelling of Nonlinear Stellar Pulsations, Problems and Prospects

DISCLAIMER

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

...

Los Alamos National Laboratory
Los Alamos, New Mexico 87545

MASTER

NUMERICAL METHODOLOGIES FOR SOLVING PARTIAL DIFFERENTIAL EQUATIONS

James M. Hyman, T-7
Center for Nonlinear Studies
Theoretical Division, MS B284
Los Alamos National Laboratory
Los Alamos, NM 87545
USA

ABSTRACT. The numerical methods for solving systems of partial differential equations can be analyzed by decoupling the space and time discretizations and analyzing them independently. First a method is selected to discretize the differential equation in space and incorporate the boundary conditions. The spectrum of this discrete operator is then used as a guide to choose an appropriate method to integrate the equations through time. The dissipative effects of a numerical method are crucial to constructing reliable methods for conservation laws. This is particularly true when the solution is discontinuous as in a shock wave or contact discontinuity. Choosing an accurate method to accomplish each of these tasks, space and time discretization and incorporating artificial dissipation in the numerical solution, determines the success of the calculation. We will describe the methodologies used in each of these choices to construct reliable, accurate and efficient methods.

1. Introduction

Each year significant and powerful algorithms are discovered for the solution of nonlinear partial differential equations (PDEs). These algorithms can be highly complicated and problem-dependent; a method developed for a particular test problem may not work for similar problems. Methods that work well in one space dimension may not be easily extended to two or three dimensions. Linear analysis can rarely ensure accuracy with nonlinear methods or highly nonlinear equations. Most of these methods have a similar underlying structure. To better predict when a method, which is almost always developed for relatively simple test problems, will extend to more complicated situations, we must understand this underlying structure. First, we will simplify the structure of these methods so that common features among seemingly different methods emerge that were not evident when analyzing a specific method for a specific set of equations. By understanding the general patterns found in most methods, we may gain a better view of how and why the algorithms work as they do.

The prototype system of PDEs we will study can be written as

$$u_t = f(x, t, u), \quad u(x, 0) = u_0 \qquad , \qquad . \tag{1.1}$$

where the solution $u(x,t)$ lies in some function space, x is in some domain Ω, and f is a nonlinear differential operator. We use the notation u_t and u_x to represent partial differentiation with respect to time and space. On the boundary of Ω, the solution is constrained to satisfy the boundary condition

$$b(x, t, u(x,t)) = 0, \quad x \in \partial\Omega \ , \tag{1.2}$$

where b is a nonlinear spatial differential operator.

A discrete numerical method approximates u by an element U in some finite dimensional space whose components are the values of u at a discrete set of mesh points or the average

values $\bar{u}$ over a cell bounded by the mesh points. The differential operators f and b are replaced by discrete operations F and B operating on U.

The discretized approximation to Eqs. (1.1) and (1.2) is a constrained system of ordinary differential equations (ODEs),

$$U_t = F(X, t, U), \quad B(X, t, U) = 0 \quad , \tag{1.3}$$

which are then integrated numerically This report is organized so that the crucial choices for methods to discretized space, boundary conditions, time, and methods to solve the algebraic systems are analyzed independently.

2. Discretizations in Space

The numerical approximation of the spatial derivatives and the distribution of the mesh points determine how well the spatial operator f(u) and the solution u will be approximated. We describe some typical methods to approximate spatial derivatives and then we describe how the errors in a calculation are related to the order of accuracy of the method.

The guiding principle in choosing a numerical method to approximate the spatial operator is that *the resulting discrete model should retain as closely as possible all the crucial properties of the original differential operator*. For instance, for a hyperbolic PDE, the operator f is antisymmetric, so we try to approximate f by an antisymmetric discrete operator F. For a parabolic PDE when f is dissipative, we approximate f with a dissipative discrete operator. If f is in conservation form, we also choose a conservation form of F.

All spatial differentiation methods we describe follow the same algorithmic flow. At time t during a calculation, we are given the approximate solution vector U at a discrete set of mesh points X and must generate a numerical approximation F(U) of f(u) at these mesh points. When f(u) is a nonlinear spatial operator, it will have terms such as $g(u,x,t)_x$ or $[s(u,x)\ g(u,x,t)_x]_x$. First, pointwise values of the solution are defined at a set of grid points. If average values of the solution within each grid cell are being calculated, then they must be interpolated to pointwise values at the edges of the grid cell. Next all nonlinear functions are evaluated to generate, say, the vectors G and S. These vectors are then differenced to approximate G_x and $(SG_x)_x$ at the mesh points.

The spatial differentiation is totally divorced from the nonlinearities of the PDE. This modularity also reduces the redundancy of programming the same approximation to the spatial derivatives each time they appear in an equation. These differentiation routines are debugged and optimized for a particular machine only once--with no specific PDE in mind.

2.1 FINITE VOLUME METHODS

Many physically motivated systems of PDEs are derived from a limiting process applied to integral equations. For example, a quantity u is conserved under the flow of a conservation law if the amount of u contained in any fixed volume Ω is due entirely to the flux f(u) across the boundary $\partial\Omega$ of Ω. These conservation laws can be expressed in integral form as

$$\frac{d}{dt}\int_{\Omega} u = \int_{\partial\Omega} f(u)\cdot\vec{n} \tag{2.1}$$

Where $\vec{n}$ denotes the outward normal to the boundary.

Moving the time derivative under the integral sign and applying the divergence theorem Eq. (2.1) can be rewritten as

$$\int_{\Omega} [\partial_t u + \nabla \cdot f(u)] = 0 \tag{2.2}$$

By letting the volume shrink to a point we obtain the PDE

$$\partial_t u + \nabla \cdot f = 0 \tag{2.3}$$

at every point where u and f are differentiable.

When numerically solving Eq. (2.1) it is natural to stop the limiting process at the local mesh spacing and solve Eq. (2.1) where Ω is a control volumes defined by the grid cells. The divergence theorem a natural identity that can be used to define the rate of change in a cell. This approach is called the finite volume method (FVM) [3, 7,9,11,13]. In one space dimension we solve

$$\partial_t \bar{u} + \partial_x \bar{f}(x) = 0 \tag{2.4}$$

where the bar represents an integral average over the grid cell:

$$\bar{u}(x_{i+1/2}) = \frac{1}{\Delta x_{i+1/2}} \int_{x_i}^{x_{i+1}} u(s)ds \quad , \tag{2.5a}$$

$$\bar{f}(x_{i+1/2}) = \frac{1}{\Delta x_{i+1/2}} \int_{x_i}^{x_{i+1}} f[u(s)]ds \quad , \tag{2.5b}$$

$\Delta x_{i+1/2} = x_{i+1} - x_i$ and the notation $\bar{f}(x)$ refers to the averaged quantity $\overline{f(u(x))}$. The control volume (grid cell) centered at $x_{i+1/2} = (x_{i+1} + x_i)/2$ has the boundaries x_i and x_{i+1}. The discrete divergence operator can be computed exactly in the (i + 1/2)-th cell as

$$\partial_x \bar{f}_{i+1/2} = \frac{f(x_{i+1}) - f(x_i)}{\Delta x_{i+1/2}} \quad . \tag{2.6}$$

The point values of f in Eq. (2.6) can be obtained by evaluating f at the point values of u, which are obtained by differentiating the cumulative integral of u,

$$U(x) = \int_{x_0}^{x} u(s)ds \quad . \tag{2.7}$$

That is,

$$u(x) = U_x(x) \quad . \tag{2.8}$$

At the grid points U_i is known exactly;

$$U_i = \int_{x_0}^{x_i} u(s)ds = \sum_{j=1}^{i-1} \Delta x_{j+1/2}\, \bar{u}_{j+1/2} \quad . \tag{2.9a}$$

and

$$\bar{u}_{i+1/2} = (U_{i+1} - U_i) / \Delta x_{i+1/2} \quad . \tag{2.9b}$$

Between the grid points, U is approximated by an interpolant. For example, if U is approximated by a piecewise polynomial, then the reconstructed u is also a piecewise polynomial. Typically, u is approximated by a piecewise constant [6], linear [11], parabolic [3,11], or a higher order Hermite interpolant [5, 7, 9]. The derivative of this interpolant at x_i is an $0(\Delta x^k)$ approximation to $U_x(x_i)$. This can be used in Eq. (2.6) to define the pointwise values f_i and f_{i+1} to give

$$\partial_x \bar{f}_{i+1/2} = \frac{f(U_x(x_{i+1})) - f(U_x(x_i))}{\Delta x_{i+1/2}} + O(h^{k-1}) \tag{2.10}$$

This process of going from average values of u to evaluating the flux in and out of a control volume is illustrated in Fig. 2.1.

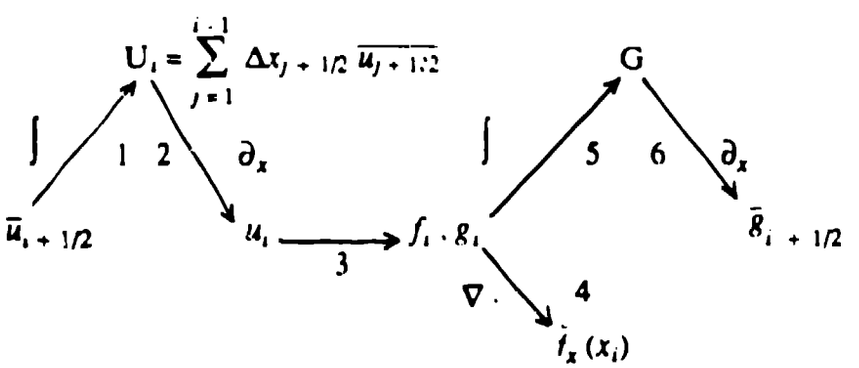


Figure 2.1 Schematic of the FMV for $\partial_t \bar{u} + \partial_x \overline{f(u)} = \overline{g(u)}$: 1. The cumulative integral U_i is defined at the FV boundaries. 2. The interpolant of U is differentiated to define u_i. 3. The functions f_i, g_i are evaluated. 4. The divergence term is evaluated $\bar{f}_x = (f_{i+1} - f_i)/\Delta x_{i+1/2}$. 5. The lower order term g is interpolated and integrated to form G(x). 6. The average values of $\overline{g(u)}$ are defined from G..

If a local quadratic interpolant is used, then a second-order linear approximation to u at the midpoints can be found with the centered formula

$$u_i = (\Delta x_{i-1/2}\,\bar{u}_{i+1/2} + \Delta x_{i+1/2}\,\bar{u}_{i-1/2}) / (\Delta x_{i-1/2} + \Delta x_{i+1/2}) + O(h^2) \qquad (2.11a)$$

or the one-sided formula

$$u_i = [(2\Delta x_{i+1/2} + \Delta x_{i+3/2})\,\bar{u}_{i+1/2} - \Delta x_{i+1/2}\,\bar{u}_{i+3/2}] / (\Delta x_{i+1/2} + \Delta x_{i+3/2}) + O(\Delta x^2) \,. \qquad (2.11b)$$

On uniform grids the fourth- and sixth-order formulas are:

$$u_i = (-\bar{u}_{i+3/2} + 7\bar{u}_{i+1/2} + 7\bar{u}_{i-1/2} - \bar{u}_{i+3/2}) / 12 + O(\Delta x^4) \,, \qquad (2.11c)$$

and

$$u_i = (\bar{u}_{i+5/2} - 8\bar{u}_{i+3/2} + 37\bar{u}_{i+1/2} + 37\bar{u}_{i-1/2} - 8\bar{u}_{i-3/2} + \bar{u}_{i-5/2}) / 60 + O(\Delta x^6) \,. \qquad (2.11d)$$

Higher order formulas can be generated by using higher degree interpolants [4, 7, 8, 13]. In many applications u represents density, energy, pressure, concentration or some positive quantity. When u is positive then U must be monotone, and, therefore, applying a monotonicity constraint [5] on the numerical derivative approximations of $\partial_x U$ in step 2 of Fig. 2.1 is appropriate. For example, if a cubic spline interpolant is used to approximate the derivatives, then constraining U_x to satisfy

$$0 \le u_i = \partial_x U_i \le 3 \min (S_{i-1/2}, S_{i+1/2}) \,, \qquad (2.12)$$

where

$$S_{i+1/2} = (U_{i+1} - U_i) / \Delta x_i \,.$$

will guarantee the resulting interpolant preserves the monotonicity properties of the data points.

In some applications u is known to be monotone, consequently U is convex and a convexity constraint [5] such as

$$\min (S_{i+1/2}, S_{i-1/2}) \le \partial_x U_i \le \max (S_{i+1/2}, S_{i-1/2}) \qquad (2.13)$$

is more appropriate.

Next $f(u_i)$ is evaluated and $\partial \bar{f}$ is evaluated using (2.6). Note that applying these nonlinear constraints on the derivatives of U does not affect the final divergence conservation form of the derivative approximations of $\bar{f}$. This would not be the case if the constraints (2.13) or (2.13) were applied directly to $\partial_x \bar{f}$. Also, even though the high-order interpolates and derivative approximations are rarely symmetric on nonuniform grids, the resulting FVM is a conservative approximation in divergence form.

The FVM can also be applied to PDEs with lower order nonlinear terms such as occur in chemically reacting flows,

$$\partial_t \bar{u} + \partial_x \overline{f(u)} = \overline{g(u)} \,. \qquad (2.14)$$

Here $\overline{g(u)}$ may not be well approximated by $g(\bar{u})$ and this term must also be carefully treated as described in Fig. 2.1

2.2 PHASE AND DAMPING ERRORS

The errors in approximating the derivatives by finite volume approximations can be divided into two classes; phase or dispersion errors and damping or dissipation errors. The second-, fourth- and sixth-order finite volume approximations Eq. (2.11) of $\bar{u}_x$ can be written as [1]

$$\bar{u}_x(x_i) = (\bar{u}_{i+1} - \bar{u}_{i-1}) / (2\Delta x) + 0(\Delta x^2) , \tag{2.15a}$$

$$\bar{u}_x(x_i) = (- \bar{u}_{i+2} + 8\bar{u}_{i+1} - 8\bar{u}_{i-1} + \bar{u}_{i-2}) / (12\Delta x) + 0(\Delta x^4) , \tag{2.15b}$$

and

$$\bar{u}_x(x_i) = (\bar{u}_{i+3} - 9\bar{u}_{i+2} + 45\bar{u}_{i+1} - 45\bar{u}_{i-1} + 9\bar{u}_{i-2} - \bar{u}_{i-3}) / (60\Delta x) + 0(\Delta x^6) . \tag{2.15c}$$

The errors in a finite volume approximation can be computed exactly for numerical approximations of traveling wave solutions to

$$\bar{u} + v\bar{u}_x = 0 , \tag{2.16}$$

with periodic boundary conditions on the unit interval and constant velocity v. The solution is a traveling wave with the solution u (x,t) = u (x-vt, 0).

When the initial conditions consist of a single frequency, u(x,0) = a sin(kx) + b cos(kx), then the phase error introduced by the finite volume approximation will be the same using second-, fourth- or sixth-order differences if the number of mesh points in the calculations satisfy [8]

$$M_2 \equiv 0.36\, M_4^2 \equiv 0.12\, M_6^3 . \tag{2.17}$$

Here M_j is the number of mesh points when using j-th order finite volume method.

Table 2.1 compares the number of points per wave length necessary to obtain a given phase error e in the solution to (3.2) at time t using second-, fourth- and sixth-order centered differences.

2nd order M_2	4th order M_4	6th order M_6	Accuracy e/ (vkt)
4	4	3	2.6
8	5	4	0.65
16	7	5	0.16
32	10	7	0.04
64	14	8	0.01
128	19	10	0.0025
256	27	13	0.0006

Table 2.1. Points per wavelength for second-, fourth- and sixth-order differences to have the same accuracy.

In a calculation where the solution contains many different frequencies, the high modes (2-5 points per wavelength) are approximated equally poorly with all the methods. The middle modes (6-16) points per wavelength) are computed much more accurately with the fourth and sixth-order differences than with the second order methods. The sixth-order differences are more accurate for the lower modes than either second- or fourth-order differences.

The relationship of the accuracies of the different methods compared to the number of points per wavelength is even more impressive in higher dimensions. In two space dimensions the numbers in Table 2.1 should be squared; in three dimensions cubed. An example of the gain in accuracy by using a third-order method are illustrated in Fig. 2.1. In these calculations, done on an irregular grid with volume ratios of over 100 to 1, the higher order method was obtained by fitting a quartic polynomial through the cummulative integral of u and its four neighboring points and differentiating it using the DERMOD package described in [8].

Also, in calculations with shock waves or other sharp fronts, the post shock oscillations are reduced by the high order differences. Furthermore, these methods are able to resolve the discontinuities better and require less artificial dissipation to eliminate post shock oscillations.

This error analysis has been for periodic boundary conditions. For nonperiodic boundary conditions the results are also valid if the boundary conditions have been approximated as accurately as the solution in the interior. Often, important properties of the solution behavior originate at the boundary and the numerical differentiation procedure must take the boundary conditions into account.

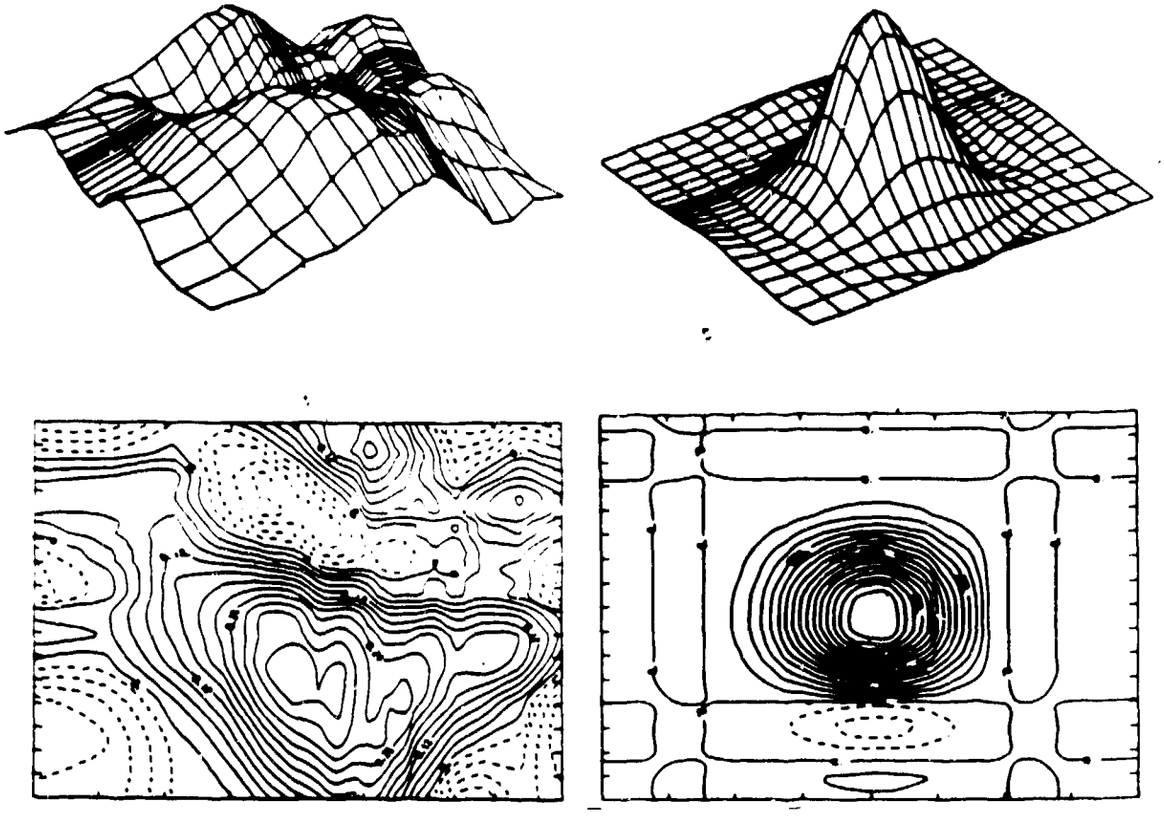


Fig. 2.1a first-order approximation

Fig. 2.1b Second-order approximaton

Figure 2.1 The numerical solution of the wave equation is shown after a Gaussian has moved around the periodic box once in the vertical and horizontal direction using a first-order (2.1a) and third-order FVM. The grid is exponentially refined in both x and y with cell volume ratios of over 100 to 1.

3. Boundary Conditions

Before calculating the solution to any differential equation one should determine if the boundary conditions are consistent with a well posed problem. A numerical method cannot be expected to generate reasonable results for a problem which does not have a well defined reasonable solution. The importance of proper boundary conditions cannot be overstressed, the boundary conditions exert one of the strongest influence on the behavior of the solution. Also, the errors introduced into the calculation from improper boundary conditions persist even as the mush spacing tends to zero.

A common error in prescribing boundary conditions for conservation laws is to over or under specify the number of boundary conditions. Overspecification usually results in nonsmooth solutions with mesh oscillations near the boundary. Underspecification does not insure the solution is unique and the numerical solution may tend to wander around in steady state calculations. In either case the results of the calculation are not accurate and one should be skeptical of even the qualitative behavior of the solution.

Once it has been determined that the differential equations and boundary conditions are well posed, special care must be taken to preserve this in the difference approximation. The way in which boundary conditions are specified for the difference equations can change a well-posed PDE into an ill-posed (unstable) discrete problem. Two of the most reliable methods to incorporate boundary conditions into the discrete equations are the extrapolation and the uncentered difference methods. Both of these methods work best by enforcing constituent relationships on the difference equations such that *the discrete equations are consistent with as many relationships that can be derived from the boundary conditions and differential equation as possible*.

3.1 FICTITIOUS POINTS

One of the most effective methods to incorporate the boundary conditions into the discrete approximation of the PDE is to extrapolate the solution to fictitious points outside the region of integration using both the differential equations and the boundary conditions. To define the nonphysical solution at the fictitious points, the boundary conditions are differentiated with respect to time and the time derivatives are replaced with spatial derivatives using the PDE to obtain differential constraints for the extrapolation formulas.

We will demonstrate this technique for reflecting boundary conditions to the Euler equations of fluid dynamics:

$$\overline{W}_t + \overline{F(W)}_x = 0 \tag{3.1}$$

$$W = \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix} \qquad F = uW + \begin{pmatrix} 0 \\ p \\ pu \end{pmatrix}$$

where

ρ = mass density
u = velocity
ρu = momentum
$E = \rho(I + 1/2\, u^2)$ = total energy per unit volume
I = internal energy
p = pressure

The numerical solution of Eq. (3.1) is a highly complicated and problem dependent process. The solution usually contains dynamic interactions between shock waves, rarefaction waves and contact discontinuities. A method developed for a particular test problem may or may not work for another with stronger (or weaker) shocks and contact discontinuities.

Equation (3.1) is hyperbolic if pressure is an increasing function of density at constant entropy. This is the case if we assume the equation of state to be that of a polytropic gas, i.e. $p = (\gamma-1)I\rho$. The parameter γ is a constant greater than one and equal to the ratio of the specific heats of the gas. For this equation of state we have $c^2 = \gamma p/\rho$ at constant entropy. The quantity c is called the local sound speed of the gas and is related to the characteristic velocities u, $u + c$ and $u - c$ of Eq. (3.1)

The reflecting boundary conditions for a thermally insulated wall for Eq. (3.1) at $x = x_0$ are

$$u(x_0, t) = 0, \quad I_x(x_0, t) = 0. \tag{3.2}$$

The thermally insulating boundary condition, $I_x = 0$, is obtained from the limit of the viscous dissipative equations as the viscosity and heat dissipation tend to zero. This condition is necessary to prevent a boundary layer in the difference approximation of inviscid calculations due to the presence of artificial dissipation.

To incorporate these boundary conditions into our numerical solution when using fourth-order centered differences we will introduce two fictitious points at $x_{-1} = x_0 - \Delta x$ and $x_{-2} = x_0 - 2\Delta x$ outside the region of integration. At these points we need an approximation to ρ, ρu, and E to preferably fourth-order.

Combining Eqs. (3.1), and (3.2) at $x = x_0$ we have

$$0 = -(\rho u)_t = (\rho u^2 + p)_x = p_x = (\gamma-1)I\rho_x \quad \text{and} \quad \rho_x = 0 \quad ,$$

$$E_x = [\rho(I + \tfrac{1}{2}u^2)]_x = 0 \, ,$$

and

$$-(\rho u)_{xx} = (\rho_t)_x = -(\rho_x)_t = 0 \quad .$$

Since these equations are valid for all time and $\gamma \neq 1$ we have

$$\rho_x = (\rho u) = (\rho u)_{xx} = E_x = 0 \tag{3.3a}$$

as auxiliary boundary conditions at $x = x_0$ consistent with the original problem. The nonphysical solution at the fictitious points outside the region of integration are defined so the finite difference approximation of Eqs. (3.3) are satisfied at the boundary.

When we replace the derivatives in these auxiliary boundary conditions by the standard centered finite differences we see that Eqs. (3.3) are satisfied if

$$\rho_{-i} = \rho_i, \quad (\rho u)_i = -(\rho u)_{-i}, \quad E_{-i} = E_i, \quad \text{for } i = 1 \text{ or } 2. \tag{3.3b}$$

3.2 NO FICTITIOUS POINTS

There is not always a simple extrapolation formula such as Eq. (3.3) to extend the solution to the fictitious points. For these problems it is often better to use uncentered differences near the boundary. The goal of this approach is to extend the number of boundary conditions so that all components of the solution are defined at the boundary. Again, these additional boundary conditions must be consistent with the original problem and as many relationships as can be derived from it. An uncentered difference approximation is then used to approximate the spatial derivatives at the mesh points.

This method will be described for the linear hyperbolic system of M equations

$$W_t = H(x)W_x \tag{3.4}$$

with the boundary conditions

$$SW_0 = b(t), \quad x = x_0 \, . \tag{3.5}$$

Difficulties arise in defining the solution at the boundary whe $0 < \text{Rank}(S) < \text{Rank}(H) = M$ and there does not exist a unique solution W_0 of (3.5). If $\text{Rank}(S) = 0$ then all the characteristics are outgoing and using either uncentered differences at the points near the boundary or straight forward extrapolation to the fictitious points gives accurate results. When $\text{Rank}(S) = M$ then all the characteristics are entering the boundary and all the components of the solution can be solved for on the boundary. Uncentered spatial

differences can then be used at the points near the boundary and will result in an accurate approximation of the boundary conditions.

When Rank(S) is greater than zero but less than M then by differentiating Eq. (3.5) with respect to time and replacing W_t from Eq. (3.4) we have

$$SH(x)W_x = b'(t), \quad x = x_0 . \tag{3.6}$$

Approximating W_x by second-order one-sided differences results in

$$SH_0W_0 = [SH_0(4W_1 - W_2) - 2\Delta x b'(t)] / 3 + 0(\Delta x^3) \tag{3.7}$$

where $H_0 = H(x_0)$. Equation (3.7) gives us additional information about the boundary conditions that is consistent with both the original boundary conditions (3.5) and the differential Eq. (3.4). If we still do not have enough boundary conditions to solve for W_0 uniquely then we can continue by differentiating (3.6) with respect to time and using Eq. (3.4) again.

It is often the case that H_0 is nonlinear and the above procedure must be iterated. Usually one or two iterations are sufficient for a stable accurate boundary approximation.

Once W_0 has been found we can use uncentered finite differences to approximate the spatial derivatives at the mesh point nearest the boundary or we can extrapolate the solution to fictitious points outside the region of integration. This extrapolation can be done by replacing the derivatives in Eqs. (3.6) with second-order centered differences and solving for W_{-1}.

3.3 NONPHYSICAL BOUNDARIES

There are many initial boundary value problems where it is essential to introduce artificial boundaries to reduce the computing time and storage of a calculation. These problems are usually posed in a domain much larger than the subregion where the solution is of interest. The subregion is blocked off and imbedded in the original problem by creating artificial boundaries. The boundary conditions at the artificial boundary are chosen such that the solution on the full domain would automatically satisfy these internal boundary conditions if the full problem were solved. The goal, of course, is to approximate the original problem as closely as possible on the reduced domains.

Consider mapping the initial boundary value problem for the Euler equations on the half line $[0,\infty)$ into $[0,b)$ with a map such as

$$y = \begin{cases} x & 0 < x < 1 \\ b+(1-b)/x & 1 < x < \infty . \end{cases} \tag{3.8}$$

In this new coordinate system Eq. (3.1) transforms to

$$W_t + s(y)F_y = 0 , \quad y \in [0,b)$$

where

$$s(y) = \begin{cases} 1 & 0 < y < 1 \\ (b-y) / (b-1)^2 & 1 < y < b \end{cases} \tag{3.9}$$

The solution to (3.9) is identical to the solution of our original problem. Therefore, the transformed system has the correct number of signals entering and leaving through the artificial break point at x = 1.

In this transformed system a wave slows down in the region (1,b) and approaches zero speed as x nears b. This causes a wave train to squeeze up, with the lower frequencies being pushed into higher ones. These high frequencies cannot be computed accurately and it is best to add some dissipation to damp them out as they approach the transformed infinity boundary b. This damping should be chosen such that the signals propagating into the region of interest [0,1] depend in some sense on an average of the solution outside this region, i.e. (1,b). A possible form for the dissipation is

$$W_t + s(y)F_y = (\Delta y d(y) W_y)_y \tag{3.10}$$

where

$$d(y) = \begin{cases} 0 & 0 < y < 1 \\ \delta[(y-1)/(b-1)]^2 & 1 < y < b \end{cases}$$

Notice that the equation is unchanged in the interval [0,1] and becomes parabolic in the interval (1,b]. In fact at y = b the equation reduces to a simple diffusion equation. Boundary conditions must be given for all the variables at y = b for the problem to be well-posed. The boundary condition for steady flow at infinity ($W_y = 0$) gave the best results in a series of test problems.

By imbedding the equation in the subregion into a well-posed problem in a slightly larger domain the difficulty of maintaining the correct number of boundary conditions at the artificial boundary was solved automatically. Furthermore, the information entering the region at this boundary depends on some global average properties of the solution outside the subregion.

3.4 SPECIAL PDE FORMS

3.4.1 Characteristic Form. In problems where the solution is sensitive to the approximation of the boundary conditions it may be more stable to transform the boundary conditions or the equation into characteristic form at the boundary. The extrapolation formulas are then derived to extrapolate the outgoing characteristic variables to the fictitious points.

Characteristic variables are also important when no amount of algebra seems to yield enough relationships to uniquely define all the solution variables at the fictitious points. When this happens one is forced to extrapolate on some of the variables without any boundary relationships to guide the extrapolation. It is usually best to extrapolate on outgoing characteristic variables and use their values at the fictitious points to provide the extra needed information.

2.4.2 Differential Form. Whatever extrapolation formula is used there may be some inherent truncation error in the extrapolated solution at the fictitious points. Some of these truncation errors can be eliminated by changing the differential form of the equation at the boundary. For example, the reflecting boundary conditions (3.2) can be incorporated in the Euler equations at the boundary to give

$$W = \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}_t + \begin{pmatrix} \rho u \\ 0 \\ pu \end{pmatrix}_x = 0 \qquad (3.11)$$

at the boundary. By differencing and integrating these equations, rather then Eq. (3.1), at the boundary we have prevented some of the possible truncation errors inherent in the extrapolation formula, from creeping into our calculation. Notice that the modified Eq. (3.11) has been kept in divergence form. This is particularly important to maintain conservation when shocks are reflected at the wall.

Using the modified differential form of the equations is especially important when there is a removable singularity at the boundary. This often occurs at the orgin in PDEs formulated in cylinderical or spherical symmetry At the singularity these terms should be replaced by their equivalent nonsingular form obtained using L'Hôpital's rule.

4. Artificial Dissipation

The purpose of the artificial dissipation or artificial viscosity is to remove many of the numerical difficulties of integrating hyperbolic PDEs with shock waves or other discontinuities in the solution.by dissipating or damping out energy in the high frequencies of the solution. This approach does in some sense mock up the effects of the viscous and dissipative terms discarded in the derivation of the Euler equations in that it primarily dissipates the high wave numbers, but it has little to do with true heat dissipation or viscosity. Artificial dissipation is a special form of truncation error either inherent to a finite difference approximation or resulting from explicitly adding an additional term to the equation. This dissipation is the leading truncation error in the numerical approximation and is chosen on the basis of the expected form of the solution.

There are six primary reasons for including artificial dissipation in the numerical approximation. They are:

1. To achieve proper entropy production across shock fronts.
2. To smooth out nonphysical discontinuities in the flow.
3. To solve the problem of the energy cascade when computing only a finite number of modes.
4. To compensate for spatial interpolation errors, such as the Gibb's phenomenon, near discontinuities in the solution.
5. To counteract the dispersion error in the numerical scheme.
6. To stabilize certain time differencing methods.

The form of a good artificial dissipation term tailored for a particular problem will depend on which of these points are most important. It is therefore essential to designing a numerical method to have a basic understanding of each of them. In this section we will review each reason for adding artificial dissipation and suggest a form which works for a large class of problems.

4.1.1. Entropy Production. The most common reason given for adding artificial dissipation is so that one can calculate shock waves. Entropy increases across a shock front, but Eq. (3.1) has no mechanism for the increase. We must add a term to the equation which will allow entropy to increase by the proper amount. The term should be in conservation form to maintain the Rankine-Hugoniot jump conditions and therefore give the correct shock speed.

4.1.2. Nonphysical Discontinuities. Another desired effect of the artificial dissipation is to smooth out nonphysical discontinuities in the flow. That is, it would be advantageous if the artificial dissipation were formulated in such a way that physical shocks are stable and

nonphysical sudden compression shocks are unstable. These nonphysical discontinuities often occur in the initial conditions and can be smoothed out by using more artificial dissipation in the first few time steps than later in the calculation.

4.1.3. Energy Cascade. Typically, in Eq. (3.1) energy enters the system at low wave numbers and cascades upward to higher wave numbers where it is eventually dissipated by molecular viscosity and enters the system as heat (Kolmogoroff hypothesis). In numerical calculations the energy spectrum is limited by the number of mesh points. When there is no artificial dissipation in the system the energy cascade backs up at the higher frequencies and shows up in the calculation as high frequency noise or trash. Some of this energy is aliased or reflected back into the lower wave numbers. This closed loop energy cascade can destroy the accuracy in all wave numbers during even moderately short computations.

4.1.4. Gibb Phenomenon. Artificial dissipation can help compensate for some of the errors introduced by approximating U and U_x with an interpolant whose values agree with U only at a discrete set of points. The errors in the interpolant are most severe near discontinuities in the function being approximated. At these points the continuity conditions used to derive the interpolant breakdown and oscillations appear in the calculation. These oscillations can destroy the accuracy of the calculation by creating nonlinear instabilities or introducing nonphysical features in the flow such as negative mass or pressure. The oscillations may generate new artifacts into the calculation such that the numerical calculation is stable but converges to the wrong solution In reacting flows, overshoots in temperature can prematurely trigger a chemical reaction or combustion front and lead to meaningless results. Adding artificial dissipation to the numerical approximation damps the high frequencies and helps reduce superfluous oscillations in the solution.

4.1.5. Dispersion error. Dispersion errors come from the inexactness in both the time and space differencing methods. The dispersion errors due to the different modes of the solution traveling at different and incorrect velocities can accumulate and destroy the accuracy of the computation. This is particularly true for the higher modes even in calculations of flows which should have only smooth solutions. Increasing the accuracy in both the time and space differencing methods will reduce the dispersion the low and middle frequencies, but not the high modes. It may be best to damp these out by some form of artificial dissipation.

4.1.6. Stabilization of Time Integration Methods. The ability of artificial dissipation to stabilize, what may otherwise be an unstable time differencing method for hyperbolic PDEs, results from the fact that it shifts the spectrum of the spatial operator to the left half plane so that the solution to the modified equation is mathematically and numerically more stable. This may be necessary for such standard integration methods, such as the forward Euler method, that are unstable for ODEs with imaginary eigenvalues. This need can be overcome by using an integration method that is stable for ODEs with imaginary eigenvalues and hence approiate for hyperbolic PDEs. An example is the leap-frog predictor-corrector method described in Sec. 5

4.2 DIFFERENTIAL FORM

For many problems the artificial dissipation inherent to the time integration method is sufficient to compensate for the energy cascade problem and also the entropy production in weak shocks. For strong shocks it is necessary to add significantly more dissipation. The extra dissipation can be added by explicitly adding a dissipative truncation error to Eq.

(3.1). A simple scaling analysis can show that adding a second order artificial dissipation term of the form

$$\overline{W}_t + \overline{F}_x = \overline{(\Delta x\, \delta \lambda_{max} W_x)_x} \qquad (4.1)$$

will fix the shock width to be a fixed number of grid points and will be independent of linear transformations of the equation. Here λ_{max} is the magnitude of the largest characteristic velocity of the system [$\lambda_{max} = \delta(|u|+c)$ for Eq. (3.1)] and δ is a dimensionless scaling parameter. Numerical experiments have verified that choosing the parameter δ = 0.25 results in monotone solution profiles when second-order finite volume methods are used and the equations are integrated accurately in time. The higher order finite volume methods remain monotone with slightly lower values of δ. To retain accuracy away from shock waves some of the more sophisticated methods include a switch that detects the presence of a shock and scales δ to be small where the solution is smooth.

5. Time Discretization

The numerical solution of (1.3) is advanced in time in discrete steps that vary depending on the local behavior of the solution; that is, the length of the time steps depends on whether the solution is evolving on a slow or fast time scale. The major difference between time and space differentiation is that time has a direction. This time flow allows savings in computer storage, but introduces questions about the time stability of the difference equations relative to the stability of the differential equations. For example, in choosing the an appropriate numerical method to integrate the Euler equations through time one has to consider the accuracy, stability, storage requirements, computational complexity and the relative cost of the different methods. These factors are dependent on each other and tradeoffs must be made as to which criteria are more important for a particular problem.

5.1 SPECTRAL ANALYSIS

Both the phase and damping errors depend on the spectrum of the differential equation and the time step size. The time step can be varied during the calculation to reduce the numerical integration errors, but the spectrum of the differential equations is determined by the spatial difference operator. A good integration method depends on how accurately it can integrate a particular set of equations. For this reason *the spectrum of the spatial difference operator is the most important guide in selecting an efficient numerical method to integrate through time*. The spectrum can be determined by analyzing the linearized continuous time - discrete space approximation of the PDE.

Equation (3.1) is solved after adding artificial dissipation and therefore we must analyze a system of the form (4.1). Most of the essential properties of this system are also found in the simple prototype equation

$$\rho_t + \rho_x = (\Delta x \delta \rho_x)_x \ . \qquad (5.1)$$

A semi-discrete approximation of (5.1) results when the spatial derivatives are approximated by finite differences on a mesh of N points. This system can be written in the form of ordinary differential equations (ODEs)

$$y' = Ay + \delta \Delta x B y = Cy = f(y) \ . \qquad (5.2)$$

The vector y is an array of the approximate solution at the mesh points and the prime denotes the derivative of y with respect to time.

When second-order centered differences are used the eigenvalues of A are imaginary and the eigenvalues of B negative real. The eigenvalues of $\Delta x C$ are complex and lie on the ellipses in the complex plane shown in Fig. 5.1 .

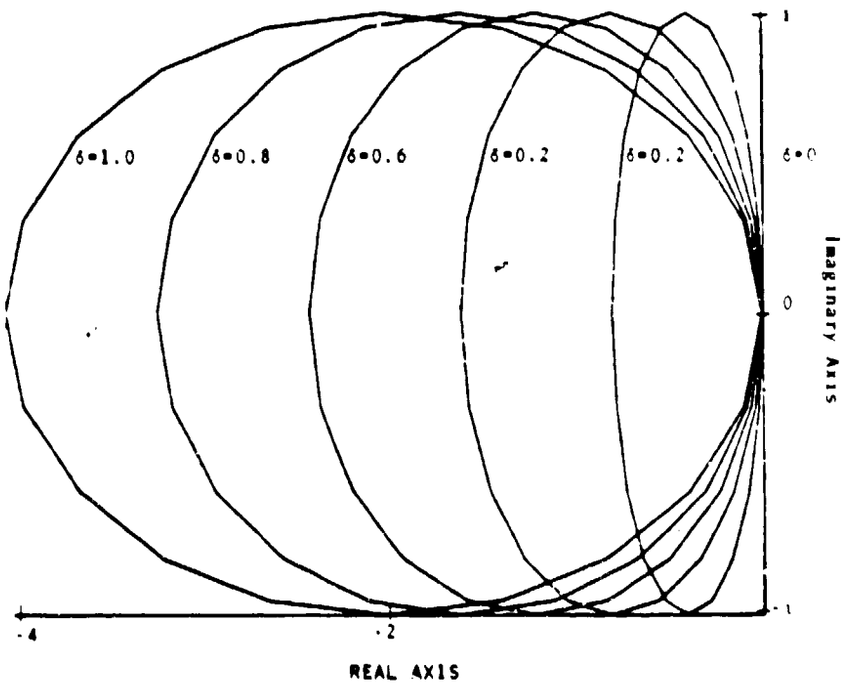


Figure 5.1 The eigenvalues of $\Delta x C$ in Eq. (5.2) are complex and lie on ellipses for values of $\delta = 0.0, 0.2, ...,1.0$.

We shall first analyze Eq. (5.2) when there is no artificial dissipation (i.e. $\delta = 0$) and include the effects of the dissipation as a perturbation on this equation. When $\delta = 0$ Eq. (5.2) is dispersive since the eigenvalues of A lie on the imaginary axis. These eigenvalues, λ, are equal to $i\alpha u$, $i\alpha(u + c)$ and $i\alpha(u - c)$, where α depends upon the spatial order of approximation. When second- or fourth-order centered differences in Eq. (3.1) are used and the boundary conditions are periodic on the unit interval the corresponding α's are

$$\alpha_2 = (\sin(2\pi j\Delta x)) / \Delta x , \tag{5.3}$$

and

$$\alpha_4 = (8 \sin(2\pi j\Delta x) - \sin(4\pi j\Delta x)) / 6\Delta x , \tag{5.4}$$

for $j = N/2, -N/2 + 1,...,N/2$ and $\Delta x = 1/N$.

To facilitate studying the properties of different time integration methods we use the isolation theorem. That is, the stability and accuracy of a numerical integration method for Eq. (5.2) is determined entirely by how it approximates the decoupled diagonalized system

$$y_i' = \lambda_i y_i , \tag{5.5}$$

with the solution $y_i(t) = y_i(0)e^{\lambda_i t}$ where the λ_i are the eigenvalues of C.

5.2 EXPLICIT METHODS

5.2.1 Numerical Algorithms. The simplest integration method to integrate from t_n to t_{n+1} is called the forward Euler method

$$y_{n+1}^{(1)} = y_n + \Delta t\, f_n + 0(\Delta t^2) \quad . \tag{5.6a}$$

This method is linearly stable if Δt is chosen so that $\lambda\Delta t$ lies within the stability region 1 shown in Fig. 5.2. These regions are symmetric about the real axis and are shown in Fig. 5.2. The method is stable if Δt is chosen small enough that $\lambda\Delta t$ lies within its stability region for all the eigenvalues λ of C. Here, λ is any of the eigenvalues of the linearized Jacobian matrix of F in Eq. (1.3). This approximate solution can be improved by iterating the corrector of order k + 1, in this case the improved Euler corrector

$$y_{n+1}^{(2)} = y_n + 1/2\, \Delta t\, (f_{n+1}^{(1)} + f_n) + 0(\Delta t^3) \quad . \tag{5.6b}$$

Here the superscript is the iteration index, $f_{n+1}^{(1)} = f(y_{n+1}^{(1)})$. After the corrector cycle, additional iterations are based on the simple recurrence relation

$$y_{n+1}^{(i)} = y_{n+1}^{(i-1)} + c_i \Delta t(f_{n+1}^{(i-1)} - f_{n+1}^{(i-2)}) + 0(\Delta t^{i+1}) \tag{5.7}$$

for i = 3,4,... . The order of accuracy increases by one for every iteration for linear autonomous systems of equations. The constants c_i depend on the iteration count and the predictor-corrector method used to start the process. The c_i are chosen to increase the order of accuracy of the method for linear autonomous systems and each iteration. When Eqs. (5.6) are used to start the iteration the constants c_i have the simple explicit formula $c_i = 1/i$, i = 3,4,... . This method is called the iterated Runge-Kutta method since the stability region after the i-th iteration is equivalent to the stability region of an i-th order Runge-Kutta method. The stability regions, shown in Fig. 5.2, increase on each iteration and the approximations will converge to the exact solution when solving linear autonomous systems such as Eq. (5.2).

Another iterated method which has excellent stability and accuracy properties for the ODEs with eigenvalues near the imaginary axis, such as the Euler equations, is the iterated leap-frog method. The second-order leap-frog predictor is given by

$$y_{n+1}^{(1)} = (1-r^2)y_n + r^2 y_{n-1} + \Delta t(1+r)f_n + 0(\Delta t^3) \quad , \tag{5.8a}$$

where $r = (t_{n+1}-t_n)/(t_n-t_{n-1})$ and the third-order leap-frog corrector is

$$y_{n+1}^{(2)} = [(2-r)(1+r)^2 y_n + r^3 y_{n-1} + \Delta t(1+r)^2 f_n + \Delta t(1+r) f_{n+1}^{(1)}] / (2+3r) + 0(\Delta t^4) \tag{5.8b}$$

The coefficients for this method are c_i = 3/10, 7/30, 4/21, 451/2800, 314/2255, 1153/9420, and 126/1153 for i=3,4...9 when r=1. The c_i are functions of r and are not known for general r at this time. The stability regions are shown in Fig. 5.3

The leap-frog predictor is unstable for systems of equations with eigenvalues having a nonzero real part. Therefore, when artificial dissipation is added or the boundary conditions shift the spectrum of the discretized equation the leap-frog method cannot be used without the corrector cycle. The first corrector application extends the bound on the maximum time step by 50%, increases the method to third-order and is stable in smooth regions of the solution with or without any spatial artificial dissipation. Another difficulty with using the leap-frog predictor is a unique type of error due to time and space mesh decoupling. The odd and even points of a mesh are only weakly coupled when integrating conservation laws and errors with frequency = $2\Delta x$ can degrade the accuracy of the solution with high frequency noise. The corrector cycle couples the mesh points among the three time levels and prevents this weak instability.

When integrating nonlinear equations, the iterated methods (5.7) reduce to the order of the predictor-corrector of Runge-Kutta starting method. The stability regions still expand with extra iterations but the order of accuracy remains the same.

5.2.2 Stability Properties For most numerical methods it is the largest eigenvalue λ_{max} of the linearized equations that determines the stability condition. When this occurs simpler stability restrictions on Δt can be derived using Figs. 5.2 and 5.3. When second-order centered differences are used in space and the leap-frog predictor is used in time, then if $\delta = 0$ the stability condition requires $\Delta t|\lambda_{max}| < 1$ or (using Eq. (6.3)).

$$\Delta t \max(|u| + c)\left|\frac{\sin(2\pi j\Delta x)}{\Delta x}\right| = \frac{\Delta t}{\Delta x}\max(|u| + c) < 1 \qquad (5.9)$$

This is the usual Courant-Friedrichs-Lewy stability condition for explicit methods when solving the Euler equations. If fourth-order centered differences are used in space and the leap-frog predictor-corrector method in time, the corresponding stability condition is

$$\Delta t \frac{8}{6\Delta x}\max(|u| + c) < 1.5 \qquad (5.10).$$

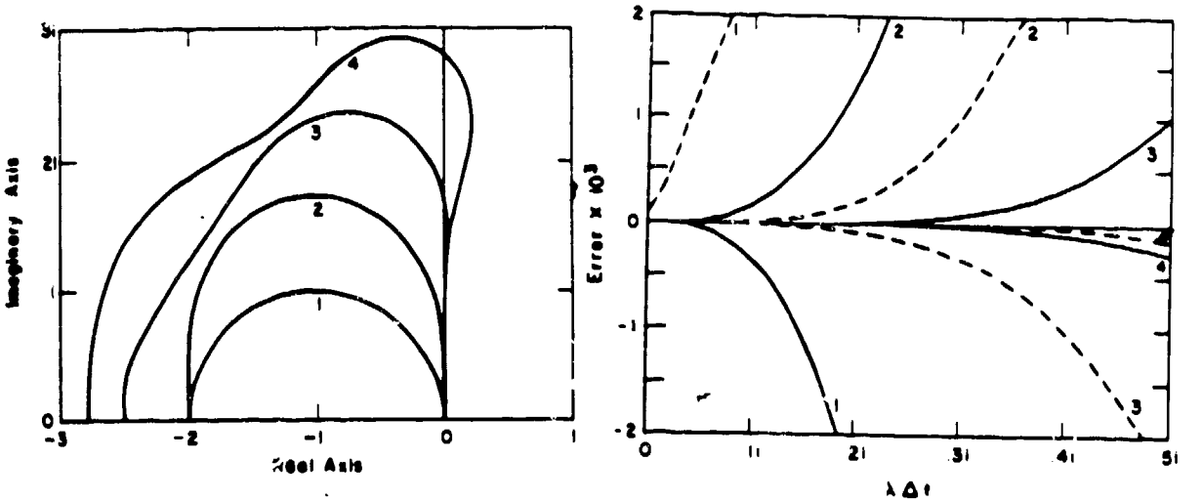
Notice in Fig. 4 that some integration schemes such as forward Euler are unconditionally unstable for all $\Delta t > 0$ when the spectrum of the discretized system lies on the imaginary axis. It is well known that forward Euler is the heart of many standard methods to solve Eq. (5.1) and in fact is not always unconditionally unstable. This is because of the addition of artificial dissipation shifts the eigenvalues of the linearized system to the left so they have a negative real part as seen in Fig. 5.1.

We caution the reader that this stability analysis is linear and is not necessarily valid for highly nonlinear phenomena such as shock waves. In practice to prevent nonlinear instabilities, it is necessary to restrict the time step slightly below the upper bound given by the linear analysis.

5.3 IMPLICIT METHODS

5.3.1 Numerical Algorithms Many problems occur when the solution changes on a slow time scale but the stability criteria limit the time step far below that needed to retain accuracy. In these cases, it is often best to use a more stable implicit method. One of the best methods for PDEs is the second-order backward difference formula

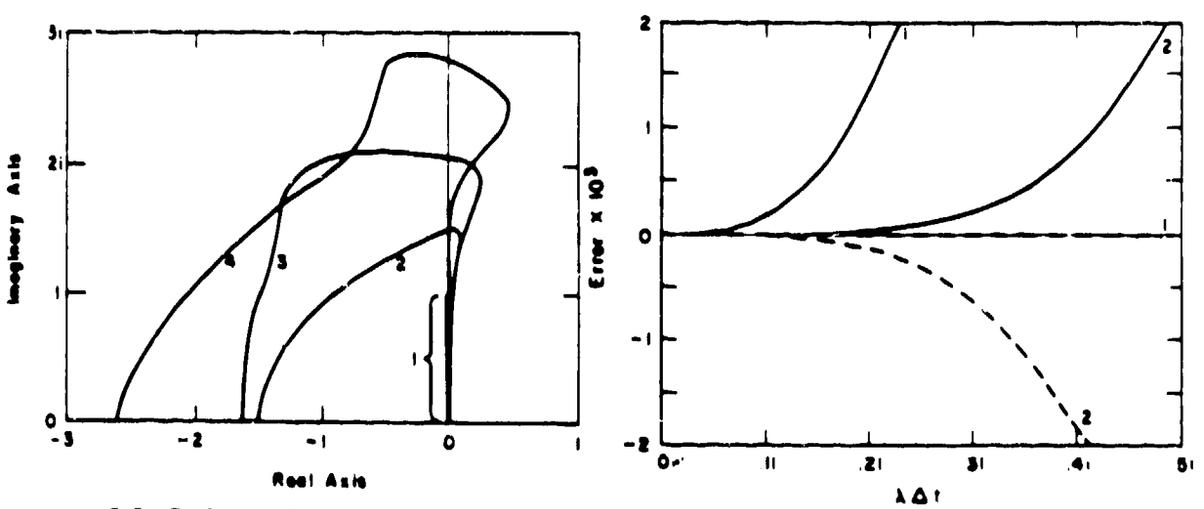
$$y_{n+1}^{(1)} = [(1 + r)^2 y_n - r^2 y_{n-1} + \Delta t(1+r)f_{n+1}]/(1 + 2r) + 0(\Delta t^3) \qquad (5.11)$$



5.2a Stability Region

5.2b Integration Error

Figure 5.2 The stability regions for the iterated Runge-Kutta method are shown in Fig. 5.2a. The i-th iteration is stable if $\lambda\Delta t$ is within the i-th curve for all eigenvalues λ of the Jacobian of F in Eq. (1.3). The phase (solid line) and damping (dashed line) errors in the different modes $\lambda\Delta t$ of the numerical solution of (5.1), δ=0, due to time truncation errors are shown in Fig. 5.2b.



5.3a Stability Region

5.3b Integration Error

Figure 5.2 The stability regions for the iterated Leap-Frog method are shown in Fig. 5.3a. The i-th iteration is stable if $\lambda\Delta t$ is within the i-th curve for all eigenvalues λ of the Jacobian of F in Eq. (1.3). The phase (solid line) and damping (dashed line) errors in the eigenvalues $\lambda\Delta t$ of numerical solution of (5.1), $\delta = 0$, due to time truncation errors are shown in Fig. 5.3b. Note that there is no damping error in the leap-frog predictor and that the corrector slightly damps the higher modes and greatly reduces the phase error in all the modes.

These method is stable when $Re(\lambda) < 0$ for all Δt, as can be seen in Fig. 5.4,. retains the positivity of the solution to (5.5), and has the proper limit for large $\Delta t\lambda$.

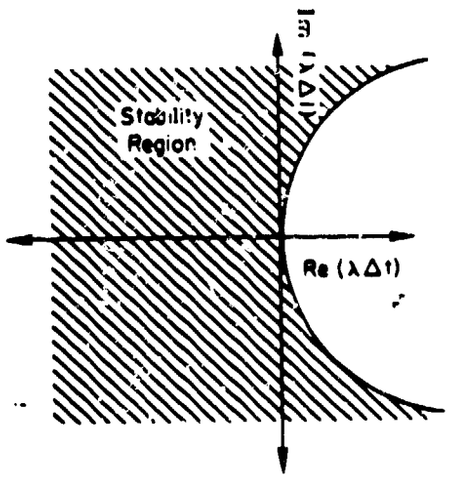


Figure 5.4 Stability region for the Second-order BDF formula (5.11) when r = 1.

On each time step of a one-cycle implicit method we must solve a nonlinear algebraic system of the form

$$y_{n+1} + \beta\Delta t f_{n+1} = \text{known quantities} \quad , \tag{5.12}$$

Several iterative methods, discussed in the next section, show how Eq. (5.12) might be solved. A good first guess can often be made by using polynomial extrapolation.

6. Algebraic Systems

6.1 ITERATIVE METHODS

When implicit methods are used then on each time step we must solve large sparse algebraic systems of equations. These equations can be written as

$$A(v) - b = 0 \quad , \tag{6.1}$$

where A is a nonlinear discrete operator, b is a known vector, and the discrete solution vector is v. Often the solution of Eq. 6.1 is difficult to obtain directly, but the residual error

$$r = A(w) - b \tag{6.2}$$

for an approximate solution w is easy to evaluate. If there is a related system

$$P(w) - b = 0 \tag{6.3}$$

that approximates Eq. (6.1) and is easier to solve, the defect correction algorithm may be appropriate.

Given a guess v_n near a root v_{n+1} of Eq. (6.1), we can expand Eq. (6.1) using Taylor series to get

$$
\begin{aligned}
0 &= A(v_{n+1}) - b \\
&= A(v_{n+1}) - b + P(v_{n+1}) - P(v_{n+1}) \\
&= A(v_n) - b + P(v_{n+1}) - P(v_n) - (J_P - J_A)(v_{n+1} - v_n) + 0(\varepsilon^2) \quad ,
\end{aligned} \tag{6.4}
$$

where $\varepsilon = v_{n+1} - v_n$, and J_P and J_A are the Jacobians of P and A. The defect correction iteration is any $0(\varepsilon)$ approximation to Eq. (6.1). The simplest such iteration is

$$
P(v_{n+1}) = P(v_n) - A(v_n) + b \quad . \tag{6.5}
$$

This iteration will converge if v_n and J_p the Jacobian of P, are near enough to v_{n+1} and J_A, respectively. Table 6.1 lists some of the more common applications of defect corrections.

The iteration (6.2) can often be speeded up by using an acceleration parameter ω to give

$$
P(v_{n+1}) = P(v_n) - \omega_n[A(v_n) - b] \quad . \tag{6.6}
$$

These methods include successive overrelaxation, dynamic alternating direction implicit methods, and damped Newton. Often a two-step acceleration method

$$
P(v_{n+1}) = b + \omega_n[P(v_n) - \alpha_n A(v_n)] + (1 - \omega_n)P(v_{n-1}) \tag{6.7}
$$

can speed up the convergence even more. These methods include the Tchebyshev [10] and conjugate-gradient [4] methods.

$P(v_{n+1}) =$	Method
$A(v_n) + J_A(v_n)(v_{n+1} - v_n)$	Newton [12]
Diagonal of J_A	Jacobi [12]
Lower triangular part of J_A	Gauss-Siedel [12]
Lower triangular part of J_A + first upper off-diagonal	Line Gauss-Siedel [12]
Coarse grid operator + relax using one of the above	Multigrid [2]
Symmetric part of J_A	Concus-Golub-O'Leary [4]
If $A = (I + \Delta t\, L_x + \Delta t\, L_y)$, then $P = (I + \Delta t\, L_x)(I + \Delta t\, L_y)$, where L_x = linearized lower order approximation to L.	ADI [12]
LU where L = lower triangular matrix U = upper triangular matrix	Incomplete LU method [10]

Table 6.1. Common examples of the defect correction iteration.

Whenever a numerical iteration is being used to solve an implicitly defined system of equations, it is extremely useful to unravel the iteration and determine exactly what equation

(6.1) the converged solution satisfies and what the preconditioning operator P is. Once these have been determined, often the iteration can be speeded up by improving the preconditioning or using an acceleration method.

7. Summary

We have used a modular approach to develop accurate and robust methods for the numerical solution of PDEs. The methods to discretize the spatial operator, the boundary conditions, and the time variable, and solve any algebraic system that may arise are combined when writing a code to solve the PDE system. Special care always must be taken when solving a nonlinear equation or when using a nonlinear method. This means that the code must be field tested. The field test is to check the reliability of the method on a particular nonlinear system of PDEs. The numerical results should be insensitive to reformulations of the equations, small changes in the initial conditions, the mesh orientation and refinement, and the choice of a stable accurate discretization method.

Another excellent analysis tool is verification that any known solutions are well approximated and that any auxiliary relationships (such as conservation laws) hold for the numerically generated solution. These checks should be made -- even if one is absolutely, positively sure that the numerical solution and coding are correct.

ACKNOWLEDGEMENT

The author thanks Bob Knapp and Clint Scovel for their insight in the many discussions we have had on the use of finite volume methods to solve PDEs. This work was performed under the auspices of the U.S. Department of Energy under contract W-7405-ENG-36 and the Office of Basic Energy Sciences, Applied Mathematical Sciences contract KC-07-01-01.

REFERENCES

1. M. Abramowitz and I. A. Stegun, **Handbook of Mathematical Functions**, National Bureau of Standards Appl. Math. Series 55, Washington D.C., USA, (1970) 877-899.

2. A. Brandt, "Multi-level Adaptive Solutions to Boundary Value Problems," Math. Comp. 31 (1977) 333-390.

3. P. Collela and P. R. Woodward, "The Piecewise-parabolic Method (PPM) for Gasdynamical Simulations," J. Comp. Phys., 54 (1984) 174-201.

4. P. Concus, G. H. Golub, and D. P. O'Leary, "Numerical Solution of Nonlinear Elliptic Partial Differential Equations by the Generalized Conjugate Gradient Method," Computing 19 (1978) 321-339.

5. R. L. Dougherty, A. S. Edelman, and J. M. Hyman, "Nonnegativity-Monotonicity-, or Convexity-Preserving Cubic and Quintic Hermite Interpolation," Math Comp. 52, No. 186, (1989), 471-494.

6. S. K. Godunov, "A Finite Difference Method for the Numerical Computation of Discontinuous Solutions of the Equations to Fluid Dynamics," Mat. Sb., 47 (1959), 271-290.

7. A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy, "Uniformly High Order Accurate Essentially Non-oscillatory Schemes, III, J. Comp. Phys., 71, 231-303 (1987)

8. J. M. Hyman and B. Larrouturou, "The Numerical Differentiation of Discrete Functions Using Polynomial Interpolation Methods," **Numerical Grid Generation,** J. F. Thompson, Ed., Elsevier North-Holland, New York (1982), 487-506.

9. J. M. Hyman, R. J. Knapp and J. C. Scovel, "Finite Volume Approximations to Differential Operators," in preparation.

10. T. A. Manteuffel, "The Tchebychev Iteration for Nonsymmetric Linear Systems," Numer. Math 28 (1977) 307-327.

11. B. van Leer, "Towards the Ultimate Conservative Difference Schemes V. A Second-order Sequal to Godunov's Method," J. Comp. Phys., 32, (1979) 101-136.

12. D. M. Young, "Iterative Solution of Large Linear Systems "(Academic Press, New York, NY, 1971).

13. S. T. Zalesak, "Very High-order and Pseudo-spectral Flux-corrected Tgransport (FCT) Algorithms for Conservation Laws," in Advances in Computer Methods for Partial Differential Equations, 4 (R. Vichnevetsky and R. S. Stapleman, Eds.), IMACS, Rutgers University (1981).