Chapter 2

Fundamentals of numerical analysis

2.1 Numerical approximation of partial differential equations

2.1.1 The Equivalence Theorem

Consider a linear partial differential equation

$$\mathcal{L}(\widetilde{u}) = 0, \tag{2.1}$$

and let L be the discrete version of the operator $\mathcal{L}$, and $\widehat{u_j}$ be the exact solution of the difference equation

$$L(\widehat{u}_j) = 0. \tag{2.2}$$

The actual numerical solution of (2.2) is $u_j \neq \widehat{u}_j$. Important issues to determine are:

1. under which conditions the exact solution of the difference equation $\widehat{u_j}$ is a good approximation of the exact solution of the PDE, $\widetilde{u}$;

2. under which conditions the actual solution of the difference equation u_j is a good approximation of $\widehat{u_j}$;

3. (most important) under which conditions u_j is a good approximation of $\widetilde{u}$.

If a numerical approximation represents the PDE accurately (*i.e.*, $\widehat{u_j} \to \widetilde{u}$ as $\Delta t,\ \Delta x \to 0$), the approximation is said to be *consistent* [condition (1) above]. If errors in the numerical solution remain bounded (*i.e.*, $||u_j - \widehat{u_j}|| < \infty$ as $\Delta t,\ \Delta x \to 0$), the approximation is said to be *stable* [condition (2) above]. Finally, if the actual numerical solution approaches the exact solution (*i.e.*, $u_j \to \widetilde{u}$ as $\Delta t,\ \Delta x \to 0$), the approximation is said to be *convergent*. An additional concern regards the *accuracy* of discretization itself, namely the rapidity with which the numerical solution approaches the exact one (for example, like Δx or Δx^2, etc.)

A very important theorem in numerical analysis is *Lax's equivalence theorem*. This theorem states that:

> If a numerical approximation of a well-posed linear initial value problem is both *consistent* and *stable*, its solution is also*convergent* to that of the initial value problem.

Thus, to prove convergence of a numerical scheme applied to the solution of a given initial value problem, it is only necessary to prove the consistency and stability of the method (which are considerably easier to show).

2.1.2 Model equations

To prove stability and consistency, and also to determine the accuracy of various numerical approximations the discussion will be limited to *model equations*, linear equations that contain the important physical features of the governing equations of fluid flows. The model equations used are:

1. The hyperbolic model equation
$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0 \tag{2.3}$$
with initial condition $u(x,0) = g(x)$ and boundary condition $u(0,t) = u_o(t)$, to represent wave-like phenomena.

2. The parabolic model equation
$$\frac{\partial u}{\partial t} = \nu\frac{\partial^2 u}{\partial x^2} \tag{2.4}$$
with initial condition $u(x,0) = g(x)$ and homogeneous boundary conditions $u(0,t) = u(L,t) = 0$, to represent dissipative phenomena.

3. The elliptic model equation
$$A\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \tag{2.5}$$
with either Dirichlet or Neumann boundary conditions specified on the boundary of the domain.

2.2 Accuracy of finite-difference schemes

2.2.1 Standard finite-difference schemes

Let ϕ_j be the value of a function $\phi(x)$ at the discrete points x_j; moreover, let the grid size $h = x_{j+1} - x_j$ be constant. Given any finite-difference approximation of a derivative, its order of accuracy can be calculated by using Taylor series. Consider, for example, the following approximation:
$$\phi' \simeq \frac{\delta_c \phi_j}{\delta x} = \frac{\phi_{j+1} - \phi_{j-1}}{2h}, \tag{2.6}$$
where a prime denotes differentiation with respect to x, and $\delta_c\phi_j/\delta x$ denotes the central finite-difference approximation to the first derivative. To evaluate the accuracy of (2.6), expand ϕ_{j+1} and ϕ_{j-1} in Taylor series about point x_j:
$$\left\{ \begin{array}{lllllll} \phi_{j+1} & = & \phi_j & +\phi_j' h & +\phi_j'' h^2/2! & +\phi_j''' h^3/3! & +o(h^4) \\ \phi_{j-1} & = & \phi_j & -\phi_j' h & +\phi_j'' h^2/2! & -\phi_j''' h^3/3! & +o(h^4), \end{array} \right. \tag{2.7}$$
in which $o(h^4)$ represents terms of order h^4 and higher. Subtracting the second of (2.7) from the first yields
$$\frac{\delta_c \phi_j}{\delta x} = \frac{\phi_{j+1} - \phi_{j-1}}{2h} = \phi_j' + \phi_j''' \frac{h^2}{6} + o(h^3). \tag{2.8}$$
The second and third terms on the right-hand-side of (2.8) are the errors committed by replacing the analytical first derivative with the finite-difference approximation (2.6) (*truncation error*). Since the leading term of the truncation error is proportional to h^2, the approximation is said to be *second order accurate.*

Conversely, one can use Taylor series to construct a finite-difference scheme of any order of accuracy. For example, one can devise a second-order accurate forwards scheme (that is, one that uses only ϕ_j, ϕ_{j+1}, ϕ_{j+2} *etc.*) by multiplying the Taylor series expansion of ϕ_j by a parameter α, that of ϕ_{j+1} by β and that of ϕ_{j+2} by γ:
$$\left\{ \begin{array}{llllllll} \alpha\phi_j & = & \alpha\,[& \phi_j & & & &] \\ \beta\phi_{j+1} & = & \beta\,[& \phi_j & +\phi_j' h & +\phi_j'' h^2/2! & +o(h^3) &] \\ \gamma\phi_{j+2} & = & \gamma\,[& \phi_j & +\phi_j' 2h & +\phi_j'' 4h^2/2! & +o(h^3) &]. \end{array} \right. \tag{2.9}$$
Adding up the three expressions above and dividing through by h yields
$$(\alpha\phi_j + \beta\phi_{j+1} + \gamma\phi_{j+2})/h = (\alpha+\beta+\gamma)\phi_j/h + (\beta + 2\gamma)\phi_j' + (\beta+4\gamma)\phi_j'' h/2 + 0(h^2); \tag{2.10}$$

to determine the unknown constants α, β and γ, the coefficient of ϕ'_j must be set equal to one, and the zeroth- and first-order error terms must be required to vanish:

$$\begin{array}{rcr} \beta + 2\gamma = 1 & \Rightarrow & \beta = 1 - 2\gamma = 2 \\ \alpha + \beta + \gamma = 0 & \Rightarrow & \alpha = -\beta - \gamma = -3/2 \\ \beta + 4\gamma = 0 & \Rightarrow & \gamma = -1/2, \end{array}$$

which gives

$$\frac{\delta_+ \phi_j}{\delta x} = \frac{1}{2h}(-3\phi_j + 4\phi_{j+1} - \phi_{j+2}) + o(h^2), \tag{2.11}$$

where $\delta_+\phi_j/\delta x$ denotes the forward finite-difference approximation to the first derivative. Commonly used finite-difference schemes are shown below.

$$\begin{aligned}
\phi'_j &= \frac{\phi_j - \phi_{j-1}}{h} + o(h) \qquad \text{(1st order backwards)} && (2.12)\\
&= \frac{\phi_{j+1} - \phi_j}{h} + o(h) \qquad \text{(1st order forwards)} && (2.13)\\
&= \frac{\phi_{j+1} - \phi_{j-1}}{2h} + o(h^2) \qquad \text{(2nd order central)} && (2.14)\\
&= \frac{3\phi_j - 4\phi_{j-1} + \phi_{j-2}}{2h} + o(h^2) \qquad \text{(2nd order backwards)} && (2.15)\\
&= \frac{-3\phi_j + 4\phi_{j+1} - \phi_{j+2}}{2h} + o(h^2) \qquad \text{(2nd order forwards)} && (2.16)\\
&= \frac{-\phi_{j+2} + 8\phi_{j+1} - 8\phi_{j-1} + \phi_{j-2}}{12h} + o(h^4) \qquad \text{(4th order central)} && (2.17)\\
\phi''_j &= \frac{\phi_{j+1} - 2\phi_j + \phi_{j-1}}{h^2} + o(h^2) \qquad \text{(2nd order central)} && (2.18)\\
&= \frac{\phi_j - 2\phi_{j-1} + \phi_{j-2}}{h^2} + o(h) \qquad \text{(1st order backwards)} && (2.19)\\
&= \frac{\phi_j - 2\phi_{j+1} + \phi_{j+2}}{h^2} + o(h) \qquad \text{(1st order forwards)} && (2.20)
\end{aligned}$$

The procedure described above shows that, if a stencil with n points is used to calculate a finite-difference approximation for a derivative, at most one can obtain an approximation of order h^{n-1}.

2.2.2 Padè approximants

More accurate (for a given stencil) approximations can be obtained by expanding both the function and its derivatives in Taylor series:

$$\left\{\begin{array}{lclllllllll}
\alpha_1\phi_{j+1} &=& \alpha_1\,[& \phi_j & +\phi'_j h & +\phi''_j h^2/2! & +\phi'''_j h^3/3! & +\phi^{iv}_j h^4/4! & +\phi^{v}_j h^5/5! & +o(h^6) &] \\
\alpha_2\phi_j &=& \alpha_2\,[& \phi_j & & & & & & &] \\
\alpha_3\phi_{j-1} &=& \alpha_3\,[& \phi_j & -\phi'_j h & +\phi''_j h^2/2! & -\phi'''_j h^3/3! & +\phi^{iv}_j h^4/4! & -\phi^{v}_j h^5/5! & +o(h^6) &] \\
\beta_1 h\phi'_{j+1} &=& \beta_1 h\,[& & \phi'_j & +\phi''_j h & +\phi'''_j h^2/2! & +\phi^{iv}_j h^3/3! & +\phi^{v}_j h^4/4! & +o(h^5) &] \\
\beta_3 h\phi'_{j-1} &=& \beta_3 h\,[& & \phi'_j & -\phi''_j h & +\phi'''_j h^2/2! & -\phi^{iv}_j h^3/3! & +\phi^{v}_j h^4/4! & +o(h^5) &].
\end{array}\right. \tag{2.21}$$

Adding up the expressions above yields

$$\begin{aligned}
\alpha_1\phi_{j+1} &+ \alpha_2\phi_j + \alpha_3\phi_{j-1} + \beta_1 h\phi'_{j+1} + \beta_3 h\phi'_{j-1} = (\alpha_1 + \alpha_2 + \alpha_3)\,\phi_j \\
&+ (\alpha_1 - \alpha_3 + \beta_1 + \beta_3)\,\phi'_j h + (\alpha_1 + \alpha_3 + 2\beta_1 - 2\beta_3)\,\phi''_j h^2/2! \\
&+ (\alpha_1 - \alpha_3 + 3\beta_1 + 3\beta_3)\,\phi'''_j h^3/3! + (\alpha_1 + \alpha_3 + 4\beta_1 - 4\beta_3)\,\phi^{iv}_j h^4/4! \\
&+ (\alpha_1 - \alpha_3 + 5\beta_1 + 5\beta_3)\,\phi^{v}_j h^5/5! + o(h^6).
\end{aligned} \tag{2.22}$$

The coefficient of ϕ'_j can now be set equal to 1, to yield an equation relating the unknown constants α and β. Four more equations can be obtained by setting the coefficients of the zeroth- through fourth-order terms to zero. This gives

$$\alpha_1 = -\alpha_3 = 3/4; \qquad \alpha_2 = 0; \qquad \beta_1 = \beta_3 = -1/4. \tag{2.23}$$

Substituting (2.23) into (2.22) yields

$$\phi'_{j-1} + 4\phi'_j + \phi'_{j+1} = \frac{3}{h}\left(\phi_{j+1} - \phi_{j-1}\right) + o(h^4). \tag{2.24}$$

Equation (2.24) (known as *Padè's formula*) represents a system of linear algebraic equations for the unknown first derivative array $\boldsymbol{\phi}'$. Solving this system by a standard tridiagonal matrix inversion procedure yields a fourth-order accurate approximation of the first derivative. Even higher orders of accuracy can be obtained using the values of ϕ and ϕ' at additional points. One can write a generalized form of (2.24) in the following form:

$$\alpha\phi'_{j-1} + \phi'_j + \alpha\phi'_{j+1} = \frac{b}{4h}\left(\phi_{j+2} - \phi_{j-2}\right) + \frac{a}{2h}\left(\phi_{j+1} - \phi_{j-1}\right), \tag{2.25}$$

where

$$a = 2(\alpha + 2)/3; \qquad b = (4\alpha - 1)/3. \tag{2.26}$$

Padè's formula is recovered for $\alpha = 1/4$, while for $\alpha = 1/3$, the approximation given in (2.25) is sixth-order accurate [5].

Padè schemes can be obtained for the second derivatives as well. They can be written in the form

$$\alpha\phi''_{j-1} + \phi''_j + \alpha\phi''_{j+1} = \frac{b}{4h^2}\left(\phi_{j+2} - 2\phi_j + \phi_{j-2}\right) + \frac{a}{h^2}\left(\phi_{j+1} - 2\phi_j + \phi_{j-1}\right), \tag{2.27}$$

with

$$a = 4(1 - \alpha)/3; \qquad b = (-1 + 10\alpha)/3. \tag{2.28}$$

For $\alpha = 1/10$ the classical Padè scheme is recovered, while for $\alpha = 2/11$ a sixth-order accurate tridiagonal scheme is obtained.

While Padè schemes can significantly increase the accuracy of spatial differentiation, they also increase the cost of a computation, since they require additional operations due to the matrix inversions. Presently, they are often employed in explicit, time-accurate codes for the study of transitional or turbulent flows, in which the need to resolve accurately the small scales of motion makes high-order schemes attractive regardless of their cost.

2.2.3 The modified wavenumber

The analytical derivative of a complex exponential $f(x) = e^{ikx}$ is $f'(x) = ike^{ikx}$; if f is differentiated numerically, however,the result is

$$\frac{\delta f}{\delta x} = ik'e^{ikx}, \tag{2.29}$$

where k' is the *modified wavenumber.* A modified wavenumber corresponds to each differencing scheme; the real part of k' represents the attenuation of the computed derivative compared to the actual one, whereas a non-zero imaginary part of k' indicates that phase errors are introduced by the numerical differentiation. For a second-order centered scheme, for instance, $k' = k\sin k\Delta x/k\Delta x$. For small wavenumbers k the numerical derivative is quite accurate; high wavenumber fluctuations, however, are resolved increasingly poorly. No phase errors are introduced.

In Figure 2.1, the modified wavenumber for several finite difference schemes are compared. Centered schemes do not introduce phase errors, but can damp high-wavenumber fluctuations significantly, whereas one-sided schemes also introduce phase errors. The modified wavenumbers of the Padè schemes, shown in Figure 2.2, agree much more closely with the exact one for a wider range of wavenumbers. Phase errors, as in all centered schemes, are identically zero.

The effect of amplitude and phase errors are also shown in Fig. 2.3, which illustrates the phase error (shifting) introduced by one-sided schemes even when at small wavenumbers. While the derivative amplitude is computed accurately for small wavenumbers ($\lambda = L/2 \Rightarrow k/k_{max} = 1/4$). The error becomes, however, significant for even moderate wavenumbers ($\lambda = L/4 \Rightarrow k/k_{max} = 1/2$) with first- and second-order schemes; at wavenumbers of the order of 3/4 of the maximum allowable, the numerical derivatives are inaccurate.

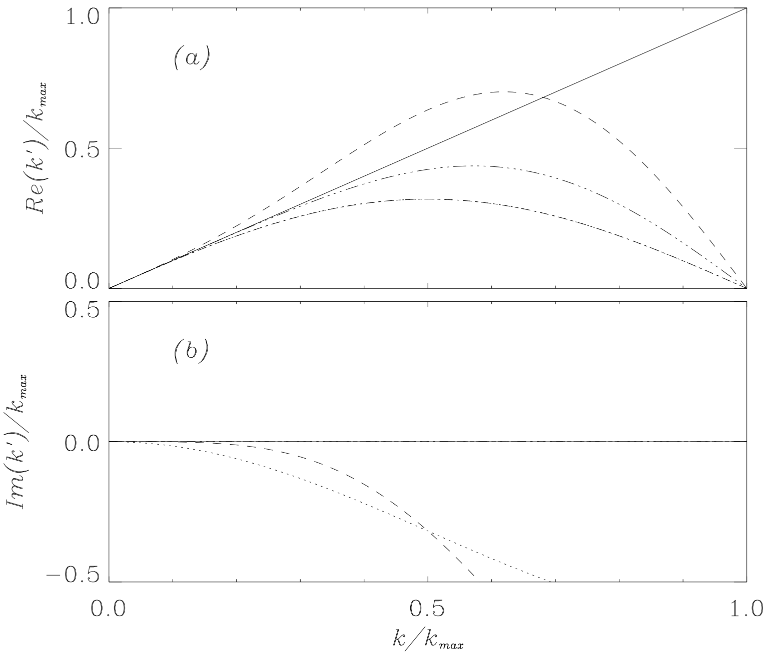

Figure 2.1: Modified wavenumber analysis for the approximation to the first derivative. —— Exact; ········ first-order backwards (2.12); – – – second-order backwards (2.15); —·— second-order central (2.14); —···— fourth-order central (2.17). (a) Real part; (b) imaginary part.

2.3 Consistency of finite difference schemes

Consider the generic partial differential equation

$$\mathcal{L}(\tilde{u}) = 0, \tag{2.30}$$

and its finite-difference approximation

$$L(u) = 0. \tag{2.31}$$

One can write, in general, that

$$L(u) = \mathcal{L}(\tilde{u}) + \text{Truncation error}. \tag{2.32}$$

A finite difference scheme is said to be *consistent* if the truncation error approaches zero as Δt and Δx approach zero.

Example 1:

Consider the following finite difference approximation of the hyperbolic model equation (2.5)

$$L(u) = \frac{u_j^{n+1} - u_j^n}{\Delta t} + c\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} = 0. \tag{2.33}$$

The truncation error can be found by using a Taylor series expansion about u_j^n:

$$\left\{ \begin{array}{lllllll} u_j^{n+1} & = & u_j^n & +\Delta t \frac{\partial u_j^n}{\partial t} & +\frac{\Delta t^2}{2!}\frac{\partial^2 u_j^n}{\partial t^2} & +\frac{\Delta t^3}{3!}\frac{\partial^3 u_j^n}{\partial t^3} & +o(\Delta t^4) \\ u_{j+1}^n & = & u_j^n & +\Delta x \frac{\partial u_j^n}{\partial x} & +\frac{\Delta x^2}{2!}\frac{\partial^2 u_j^n}{\partial x^2} & +\frac{\Delta x^3}{3!}\frac{\partial^3 u_j^n}{\partial x^3} & +o(\Delta x^4) \\ u_{j-1}^n & = & u_j^n & -\Delta x \frac{\partial u_j^n}{\partial x} & +\frac{\Delta x^2}{2!}\frac{\partial^2 u_j^n}{\partial x^2} & -\frac{\Delta x^3}{3!}\frac{\partial^3 u_j^n}{\partial x^3} & +o(\Delta x^4). \end{array} \right. \tag{2.34}$$

Substituting (2.34) into (2.33) yields

$$L(u_j^n) = \frac{\partial u_j^n}{\partial t} + c\frac{\partial u_j^n}{\partial x} + \frac{\Delta t}{2!}\frac{\partial^2 u_j^n}{\partial t^2} + c\frac{\Delta x^2}{3}\frac{\partial^3 u_j^n}{\partial x^3} + o(\Delta t^2, \Delta x^4), \tag{2.35}$$

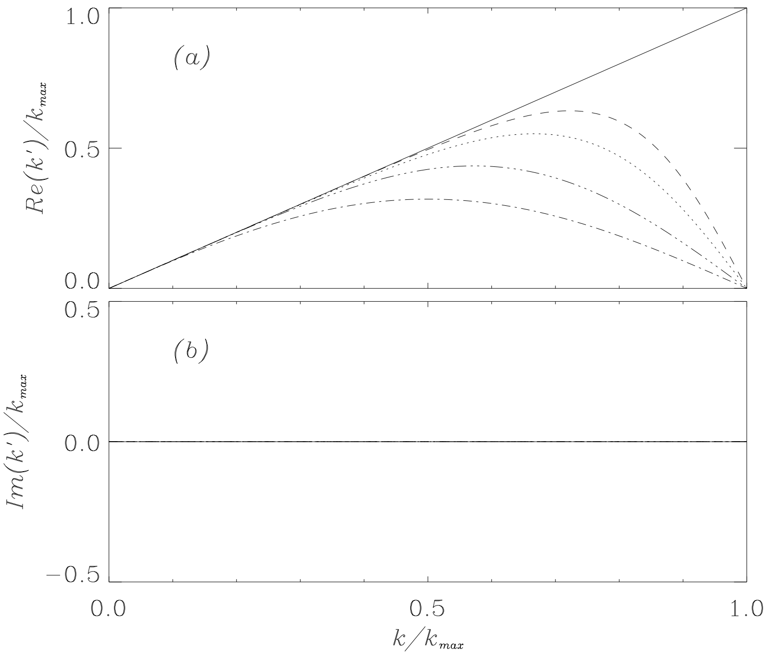

Figure 2.2: Modified wavenumber analysis for the approximation to the first derivative. —— Exact; ⋯⋯ fourth-order Padè; – – – sixth-order Padè; —·— second-order central (2.14); —···— fourth-order central (2.17). (a) Real part; (b) imaginary part.

and, using (2.5),

$$L(u_j^n) = \mathcal{L}(\tilde{u}_j^n) + \frac{\Delta t}{2!}\frac{\partial^2 u_j^n}{\partial t^2} + c\frac{\Delta x^2}{3}\frac{\partial^3 u_j^n}{\partial x^3} + o(\Delta t^2, \Delta x^4). \tag{2.36}$$

The truncation error is, therefore, given by

$$TE = \frac{\Delta t}{2!}\frac{\partial^2 u_j^n}{\partial t^2} + c\frac{\Delta x^2}{3}\frac{\partial^3 u_j^n}{\partial x^3} + o(\Delta t^2, \Delta x^4), \tag{2.37}$$

and approaches zero as Δt and Δx approach zero. The approximation (2.33) is, therefore, consistent with the hyperbolic model equation.

Example 2:

Consider the Dufort-Frankel scheme applied to the parabolic model equation (2.4). The Dufort-Frankel scheme can be obtained by writing $\partial^2 u/\partial x^2 = (u_{j+1}^n - 2u_j^n + u_{j-1}^n)/\Delta x^2$ and then applying the approximation $u_j^n = (u_j^{n+1} + u_j^{n-1})/2$:

$$L(u_j^n) = \frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} - \nu\frac{u_{j+1}^n - u_j^{n+1} - u_j^{n-1} + u_{j-1}^n}{\Delta x^2}; \tag{2.38}$$

Taylor series expansions give

$$u_{j+1}^n + u_{j-1}^n = 2u_j^n + 2\frac{\Delta x^2}{2!}\frac{\partial^2 u_j^n}{\partial x^2} + 2\frac{\Delta x^4}{4!}\frac{\partial^4 u_j^n}{\partial x^4} + o(\Delta x^6), \tag{2.39}$$

$$u_j^{n+1} + u_j^{n-1} = 2u_j^n + 2\frac{\Delta t^2}{2!}\frac{\partial^2 u_j^n}{\partial t^2} + 2\frac{\Delta t^4}{4!}\frac{\partial^4 u_j^n}{\partial t^4} + o(\Delta t^6), \tag{2.40}$$

$$u_j^{n+1} - u_j^{n-1} = 2\Delta t\frac{\partial u_j^n}{\partial t} + \frac{\Delta t^3}{3!}\frac{\partial^3 u_j^n}{\partial t^3} + o(\Delta t^5). \tag{2.41}$$

Substituting (2.39–2.41) into (2.38) yields

$$L(u_j^n) = \mathcal{L}(\tilde{u}_j^n) + \frac{\Delta t^2}{6}\frac{\partial^3 u_j^n}{\partial t^3} - \nu\frac{\Delta t^2}{\Delta x^2}\frac{\partial^2 u_j^n}{\partial t^2} + o(\Delta t^4/\Delta x^2, \Delta x^2, \ldots). \tag{2.42}$$

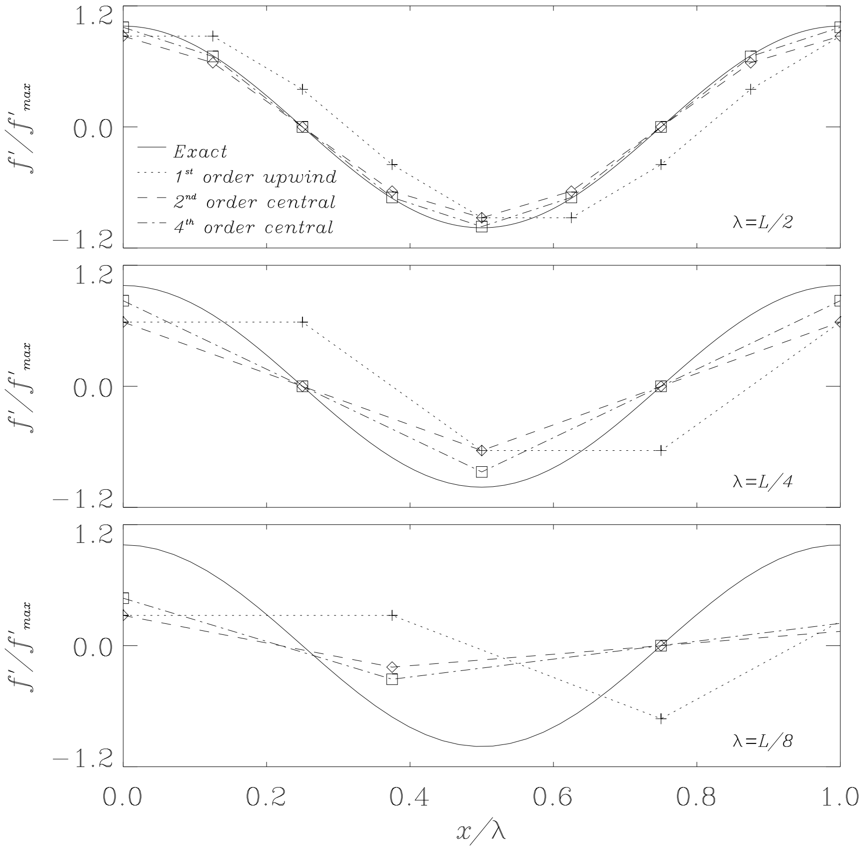

Figure 2.3: Numerical evaluation of the derivative of $\sin 2\pi x/\lambda$.

It can be seen from (2.42) that, if Δt and Δx approach zero in such a way that their ratio remains constant, the truncation error does not vanish. The Dufort-Frankel scheme, therefore, is inconsistent with the parabolic model equation.

2.4 Accuracy and stability of time-advancement schemes

2.4.1 Equations in wave space

Consider the inhomogeneous hyperbolic model equation

$$\frac{\partial \tilde{u}}{\partial t} + c\frac{\partial \tilde{u}}{\partial x} = f, \tag{2.43}$$

where $\tilde{u} = \tilde{u}(x,t)$, with homogeneous boundary conditions $\tilde{u}(0,t) = \tilde{u}(L,t) = 0$. Define now a new dependent variable, $u_j(t)$, continuous in time but discrete in space [*i.e.*, $u_j(t) = \tilde{u}(x_j,t) = \tilde{u}(j\Delta x,t)$, and $\Delta x = L/(M+1)$, so that $j = 1,\dots,M$ corresponds to the inner points only]. Replace then the spatial derivative with a finite-difference approximation to obtain

$$\frac{du_j}{dt} = -\frac{c}{2\Delta x}\left(u_{j+1} - u_{j-1}\right) + f_j \qquad \text{for} \quad j = 1,2,\dots,M. \tag{2.44}$$

Inhomogeneous boundary conditions fo the form $u(0,t) = a$, $u(L,t) = b$ can easily be cast in this form by modifying the right-hand side vector $\mathbf{f}$.

This coupled set of ordinary differential equations (ODE's) can be expressed, in matrix form, as

$$\frac{d\mathbf{u}}{dt} = A\mathbf{u} + \mathbf{f}, \tag{2.45}$$

where

$$\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \\ \vdots \\ u_M \end{pmatrix}; \qquad A = \frac{c}{2\Delta x}\begin{pmatrix} 0 & -1 & & & & \\ 1 & 0 & -1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & 1 & 0 & -1 & \\ & & & \ddots & \ddots & \ddots \\ & & & & 1 & 0 \end{pmatrix}; \qquad \mathbf{f} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \\ \vdots \\ f_M \end{pmatrix}. \tag{2.46}$$

Such a procedure can be applied to any model equation, set of boundary conditions, differencing scheme: the system (2.45) is completely general. Assuming that A is non-singular with a set of distinct eigenvalues and linearly independent eigenvectors, let $\mathbf{x_m}$ be the eigenvector associated with the m-th eigenvalue of the matrix A, λ_m:

$$A\mathbf{x_m} = \lambda_m \mathbf{x_m}, \tag{2.47}$$

where the eigenvalues λ are the roots of the characteristic polynomial

$$\det\left[A - \lambda\mathcal{I}\right] = 0, \tag{2.48}$$

in which $\mathcal{I}$ is the identity matrix; the system (2.45) is said to be *stable* if $\Re(\lambda_m) \le 0$ for all m. Let X be the right-handed eigenvector matrix:

$$X = \begin{pmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{x_1} & \mathbf{x_2} & \dots & \mathbf{x_m} & \dots & \mathbf{x_M} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}. \tag{2.49}$$

By definition,

$$X^{-1}AX = \Lambda, \tag{2.50}$$

where Λ is a matrix whose diagonal elements are the eigenvalues and the off-diagonal elements are zero. Multiplying(2.45) by X we obtain

$$X^{-1}\frac{d\mathbf{u}}{dt} = X^{-1}A\mathbf{u} + X^{-1}\mathbf{f}, \tag{2.51}$$

$$\frac{d}{dt}(X^{-1}\mathbf{u}) = X^{-1}AX(X^{-1}\mathbf{u}) + X^{-1}\mathbf{f}, \tag{2.52}$$

$$\frac{d\mathbf{w}}{dt} = \Lambda\mathbf{w} + \mathbf{g}, \tag{2.53}$$

with $\mathbf{w} = X^{-1}\mathbf{u}$ and $\mathbf{g} = X^{-1}\mathbf{f}$. The two systems (2.45) and (2.53) are equivalent, but the latter is completely uncoupled: it can be written as a set of ODE's

$$\left\{\begin{array}{lcl} dw_1/dt & = & \lambda_1 w_1 + g_1 \\ dw_2/dt & = & \lambda_2 w_2 + g_2 \\ \vdots & & \vdots \\ dw_m/dt & = & \lambda_m w_m + g_3 \\ \vdots & & \vdots \\ dw_M/dt & = & \lambda_M w_M + g_M, \end{array}\right. \tag{2.54}$$

each of which can be solved separately to yield

$$w_j = C_j e^{\lambda_j t} + PS_j, \tag{2.55}$$

where PS is a particular solution that depends on $\mathbf{g}$. The solution for u_i is

$$u_i = \sum_{j=1}^{M} x_{ij} w_j = \sum_{j=1}^{M} x_{ij}\left(C_j e^{\lambda_j t} + PS_j\right). \tag{2.56}$$

If the ODE's are stable, the first term corresponds to a transient that vanishes with time, while the particular solution represents the steady-state condition of the system. The system of equations (2.53) is referred to as the *system of ODE's in wave space*, while (2.45) is the *system of ODE's in real space.*

2.4.2 The Isolation Theorem

The *Isolation Theorem* states that:

> Applying any standard numerical scheme to each equation in a coupled set of ODE's with constant coefficients is mathematically equivalent to:
>
> 1. Uncoupling the set (including the forcing term).
> 2. Integrating each equation in the uncoupled set.
> 3. Recoupling the result to form the final solution.

As a consequence of this theorem, studying a time advancement method for any model equation can be reduced to studying the same time advancement method for a single ODE. In the following, therefore, the equation

$$\frac{du}{dt} = \lambda u + ae^{\mu t}, \tag{2.57}$$

(where λ and μ are complex scalars) will be examined with the understanding that it is representative of system (2.53), and that the original system (2.45) can be recovered multiplying $\mathbf{w}$ by X.

Equation (2.57) has an exact solution of the form

$$u(t) = C_1 e^{\lambda t} + \frac{ae^{\mu t}}{\mu - \lambda}. \tag{2.58}$$

Since

$$\begin{aligned} e^{\lambda t} &= e^{\lambda n \Delta t} = \left(e^{\lambda \Delta t}\right)^n \\ &= \left[1 + \lambda\Delta t + \frac{1}{2}(\lambda\Delta t)^2 + \frac{1}{6}(\lambda\Delta t)^3 + \ldots\right]^n, \end{aligned} \tag{2.59}$$

equation (2.58) can be recast in the form

$$u(t) = C_1 \left[1 + \lambda\Delta t + \frac{1}{2}(\lambda\Delta t)^2 + \frac{1}{6}(\lambda\Delta t)^3 + \ldots\right]^n + \frac{a e^{\mu t}}{\mu - \lambda}. \tag{2.60}$$

2.4.3 The σ roots. Accuracy of time advancement schemes.

Consider a homogeneous difference equation:

$$\sum_{k=P}^{Q} A_{n+k} u_{n+k} = A_{n+P} u_{n+P} + \ldots + A_{n+Q} u_{n+Q} = 0, \tag{2.61}$$

where $u_n = u(n\Delta t)$. Introduce the operator $E = \exp(\Delta t\ d/dt)$ such that $E^k u_n = u_{n+k}$. Equation (2.61) can then be rewritten as

$$\sum_{k=P}^{Q} A_{n+k} E^k u_n = \left[A_{n+P} E^P + \ldots + A_{n+Q} E^Q\right] u_n = 0, \tag{2.62}$$

which has the exact solution

$$u_n = \sum_{k=1}^{Q-P} C_k \left(\sigma_k\right)^n, \tag{2.63}$$

where σ_k are the roots of the characteristic polynomial

$$P(\sigma) = \sum_{k=0}^{Q-P} A_{n+P+k}\, \sigma^k = 0. \tag{2.64}$$

A general form of the difference equation (2.61) is

$$P(E) u_n = Q(E) a b^n, \tag{2.65}$$

where $P(E)$ and $Q(E)$ are polynomials in E. The exact solution of (2.65) is

$$u_n = C_1 \left(\sigma_1\right)^n + C_2 \left(\sigma_2\right)^n + C_3 \left(\sigma_3\right)^n + \ldots + PS, \tag{2.66}$$

where the particular solution is

$$PS = a b^n \frac{Q(b)}{P(b)}. \tag{2.67}$$

If a time advancement scheme is applied to the differential equation (2.57), a difference equation of the form (2.65) is obtained, and there is *at least* one root of the characteristic equation $P(\sigma) = 0$. The roots of the characteristic equation are called the *σ-roots*, and the σ-root that forms the approximation

$$\sigma \simeq e^{\lambda\Delta t} = 1 + \lambda\Delta t + (\lambda\Delta t)^2/2! + (\lambda\Delta t)^3/3! + \ldots + (\lambda\Delta t)^k/k! + \Delta t\ o\left(\Delta t^k\right) \tag{2.68}$$

is called the *principal* root; the others are the *spurious* roots. If the representative equation (2.57) represents a coupled system of the form (2.45), for every eigenvalue λ there will be *at least* one σ-root.

The σ-roots can be used to evaluate the accuracy of the time-advancement scheme. The accuracy of the time-advancement scheme is affected by errors in the evaluation of the homogeneous part of the solution of the difference equation (2.57) or by errors in the evaluation of the particular solution. The error in the evaluation of the homogeneous, or transient, solution will be defined here as

$$\epsilon_\lambda = e^{\lambda \Delta t} - \sigma_1, \tag{2.69}$$

while the error connected with the particular solution is

$$\epsilon_\mu = \Delta t(\mu - \lambda)\left[\frac{PS_{num}}{PS_{ex}} - 1\right], \tag{2.70}$$

where PS_{num} is the particular solution evaluated numerically, and given by (2.67), and PS_{ex} is the exact one given by the last term in (2.58). The order of accuracy of a given time-advancement method is equal to the smallest order of the two errors defined above.

Example 3:
Apply the explicit Euler scheme to solve the representative ODE

$$u' = \frac{du}{dt} = \lambda u + a e^{\mu t}. \tag{2.71}$$

The explicit Euler scheme approximates the time derivative by a forward difference: $u'_n = (u_{n+1} - u_n)/\Delta t$. Since $\exp(\mu t) = \exp(\mu n \Delta t)$, we can rewrite (2.71) as

$$u_{n+1} = u_n + \lambda \Delta t u_n + \Delta t\ a\left(e^{\mu \Delta t}\right)^n, \tag{2.72}$$

or, introducing the displacement operator E,

$$\left(E - 1 - \lambda \Delta t\right) u_n = \Delta t\ a\left(e^{\mu \Delta t}\right)^n; \tag{2.73}$$

for the explicit Euler time-advancement scheme, therefore, we have

$$P(E) = E - 1 - \lambda \Delta t, \qquad Q(E) = \Delta t \qquad \text{and} \qquad b = e^{\mu \Delta t}, \tag{2.74}$$

and there is only one (principal) root $\sigma = 1 + \lambda \Delta t$ that can be obtained by setting $P(E) = 0$. Considering that one can write

$$e^{\lambda \Delta t} = 1 + \lambda \Delta t + \Delta t\ o(\Delta t), \tag{2.75}$$

the error in the transient solution, ϵ_λ, is

$$\epsilon_\lambda = e^{\lambda \Delta t} - \sigma = \Delta t\ o(\Delta t). \tag{2.76}$$

Since the particular solution is also first-order accurate, the explicit Euler scheme is a first-order, one-root method.

Example 4:
Consider a two-step scheme, MacCormack's predictor-corrector scheme:

$$\left\{\begin{array}{lcl} u^*_{n+1} & = & u_n + \Delta t u'_n \\ u_{n+1} & = & \frac{1}{2}\left[u_n + u^*_{n+1} + \Delta t\ (u^*_{n+1})'\right]. \end{array}\right. \tag{2.77}$$

Introduce the representative equation (2.57) into (2.77) to yield:

$$\left\{\begin{array}{lcl} u^*_{n+1} & = & u_n + \lambda \Delta t\ u_n + \Delta t\ a e^{\mu n \Delta t} \\ u_{n+1} & = & \frac{1}{2}\left[u_n + u^*_{n+1} + \lambda \Delta t\ u^*_{n+1} + \Delta t\ a e^{\mu (n+1) \Delta t}\right]; \end{array}\right. \tag{2.78}$$

use now the displacement operator E and collect terms

$$\left\{\begin{array}{cclcl} E u^*_n & & -(1+\lambda \Delta t) u_n & = & \Delta t\ a e^{\mu n \Delta t} \\ -\frac{1}{2}(1+\lambda \Delta t) E u^*_n & & +\left(E - \frac{1}{2}\right) u_n & = & \frac{1}{2} E \Delta t\ a e^{\mu n \Delta t}. \end{array}\right. \tag{2.79}$$

The characteristic polynomial is

$$\begin{aligned} P(E) &= \det\left[\begin{array}{cc} E & -(1+\lambda \Delta t) \\ -(1+\lambda \Delta t)E/2 & E - 1/2 \end{array}\right] \\ &= E\left[E - 1 - \lambda \Delta t - (\lambda \Delta t)^2/2\right], \end{aligned} \tag{2.80}$$

while $Q(E)$ is given by

$$\begin{aligned} Q(E) &= \det \begin{bmatrix} E & \Delta t \\ -(1+\lambda\Delta t)E/2 & \frac{1}{2}\Delta t E \end{bmatrix} \\ &= E\left[E+1+\lambda\Delta t\right]\Delta t/2. \end{aligned} \tag{2.81}$$

The σ-roots are:

$$\sigma_1 = 1+\lambda\Delta t+(\lambda\Delta t)^2/2 \quad ; \qquad \sigma_2 = 0. \tag{2.82}$$

The first root is the principal one, while the second one is trivial. The particular solution is

$$PS = ae^{\mu n\Delta t}\frac{\Delta t\left(e^{\mu\Delta t}+1+\lambda\Delta t\right)/2}{e^{\mu\Delta t}-1-\lambda\Delta t-(\lambda\Delta t)^2/2} \tag{2.83}$$

The errors are respectively

$$\epsilon_\lambda = (\lambda\Delta t)^3/6 = \Delta t\ o(\Delta t^2); \qquad \epsilon_\mu = (\mu-\lambda)\mu^2\Delta t^3 = \Delta t\ o(\Delta t^2). \tag{2.84}$$

The scheme is a one-root method of order Δt^2.

Example 5:
Consider the leapfrog scheme

$$u_{n+1} = u_{n-1} + 2\Delta t\ u'_n. \tag{2.85}$$

The polynomials $P(E)$ and $Q(E)$ can be found by introducing the displacement operator and the representative equation into (2.85)

$$\left(E-\frac{1}{E}-2\lambda\Delta t\right)u_n = 2\Delta t a\left(e^{\mu\Delta t}\right)^n, \tag{2.86}$$

and are given by

$$P(E) = E^2-2\lambda\Delta t E-1; \qquad Q(E) = 2\Delta t E; \tag{2.87}$$

the roots of the characteristic polynomial are

$$\sigma = \lambda\Delta t \pm \sqrt{1+(\lambda\Delta t)^2}. \tag{2.88}$$

The principal and spurious roots can be found by expanding $[1+(\lambda\Delta t)^2]^{1/2}$ in Taylor series:

$$\sigma = \lambda\Delta t \pm \left[1+(\lambda\Delta t)^2/2-(\lambda\Delta t)^4/8+\ldots\right]; \tag{2.89}$$

the σ-roots of the leapfrog scheme are given by

$$\sigma_1 = 1+\lambda\Delta t+(\lambda\Delta t)^2/2-(\lambda\Delta t)^4/8+\ldots, \tag{2.90}$$

$$\sigma_2 = -1+\lambda\Delta t-(\lambda\Delta t)^2/2+(\lambda\Delta t)^4/8+\ldots. \tag{2.91}$$

The leapfrog scheme is, therefore, a two-root method. The principal root, σ_1, determines the order of accuracy of the scheme (second order); the spurious root does not affect the accuracy of the method, but only its stability. Schemes with multiple roots are not self-starting, and a different scheme is required to advance the solution for the first m steps, where m is the number of spurious roots.

2.4.4 Stability of time advancement schemes

For tridiagonal matrices of order M of the form

$$B = \begin{pmatrix} a_2 & a_3 & & & & & \\ a_1 & a_2 & a_3 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & a_1 & a_2 & a_3 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & a_1 & a_2 & \end{pmatrix} = B(a_1,a_2,a_3), \tag{2.92}$$

the eigenvalues are given by

$$\lambda_m = a_2 + 2\sqrt{a_1 a_3}\cos\left[\frac{m\pi}{M+1}\right] \qquad \text{for } m = 1, 2, \ldots, M \quad . \tag{2.93}$$

If central differences are used to approximate the spatial differences, and homogeneous boundary conditions are applied, the parabolic and hyperbolic model equations (2.4) and (2.5) can be written as:

$$\frac{d\mathbf{u}}{dt} = \frac{\nu}{\Delta x^2} B(1, -2, 1)\mathbf{u} \tag{2.94}$$

$$\frac{d\mathbf{u}}{dt} = \pm\frac{c}{2\Delta x} B(-1, 0, 1)\mathbf{u}; \tag{2.95}$$

their eigenvalues, therefore, are

$$\lambda_m = -\frac{4\nu}{\Delta x^2}\sin^2\left[\frac{m\pi}{2(M+1)}\right], \tag{2.96}$$

$$\lambda_m = \pm i\frac{c}{\Delta x}\cos\left[\frac{m\pi}{M+1}\right]. \tag{2.97}$$

Similar expressions can be obtained if periodic boundary conditions are used. In most cases, however, the eigenvalues must be computed numerically, and depend on the space discretization *and* boundary conditions. The eigenvalues of the hyperbolic equation are imaginary, since they represent a wave-like phenomenon. Those of the parabolic equation, on the other hand, are real and negative, and represent a solution that is decaying in time.

An ODE is said to be stable if $\Re(\lambda_m) \leq 0$ for all m. Both the parabolic and hyperbolic model equations are, therefore, stable. A numerical scheme, on the other hand, is said to be stable if the solution of the difference equations remains finite with time (and, therefore, if the error norm $||u(t) - u_n|| < \infty$). Since the solution of the difference equation is given by (2.66), this requirement is satisfied if $|\sigma| \leq 1$ for all σ.

2.4.5 Stability of time advancement schemes for the parabolic model equation

Stability bounds can be derived for any time-advancement scheme in terms of λ; they are independent of the spatial discretization and boundary conditions. The eigenvalue spectrum, however, may have to be evaluated numerically when no exact solution is available to yield actual limitations (if any) on the timestep Δt. If central differences and Dirichlet boundary conditions are used, the eigenvalues of the parabolic model equation are given by (2.96), the one with largest magnitude is $\lambda_M \simeq -4\nu/\Delta x^2$, and explicit limits for the time step Δt can be given.

Explicit Euler scheme

The σ-root for the explicit Euler scheme (2.71) is $\sigma = 1 + \lambda\Delta t$ (see above). The requirement that $|\sigma| < 1$ is satisfied if $|1 + \lambda\Delta t| \leq 1$, or

$$-1 \leq 1 + \lambda_m \Delta t \leq 1 \qquad \text{for } m = 1, 2, \ldots, M. \tag{2.98}$$

The inequality on the right is satisfied trivially; the one on the left is satisfied if $|\lambda\Delta t| \leq 2$. When the model equation can be written as (2.94), this yields

$$\Delta t < \frac{1}{2}\frac{\Delta x^2}{\nu}. \tag{2.99}$$

Leapfrog scheme

The σ-roots for the leapfrog scheme (2.85) are given by (2.90) and (2.91). The scheme is second-order accurate, and is always unstable, since $\sigma_2 < -1$ for λ real and negative.

MacCormack predictor-corrector scheme

The σ-root for the MacCormack predictor-corrector scheme (2.77) is $\sigma = 1 + \lambda\Delta t + (\lambda\Delta t)^2/2$; the scheme is second order accurate, and stable for $|\lambda\Delta t| \leq 2$.

Adams-Bashforth scheme

The Adams-Bashforth scheme,

$$u_{n+1} = u_n + \frac{\Delta t}{2}\left(3u'_n - u'_{n-1}\right), \tag{2.100}$$

is a two root scheme, whose roots are

$$\sigma_{1,2} = \frac{1}{2}\left[1 + \frac{3}{2}\lambda\Delta t \pm \sqrt{1 + \lambda\Delta t + \frac{9}{4}(\lambda\Delta t)^2}\right], \tag{2.101}$$

and is stable for $|\lambda\Delta t| \leq 1$.

Runge-Kutta schemes

The Runge-Kutta (RK) schemes are a class of multistep methods; they are one-root schemes whose σ-root is given by a truncation of Taylor's expansion of $\exp(\lambda\Delta t)$ to the order of the scheme. The explicit Euler scheme is the first-order Runge-Kutta scheme (RK1) and $\sigma = 1 + \lambda\Delta t$. MacCormack's predictor-corrector is RK2 and $\sigma = 1 + \lambda\Delta t + (\lambda\Delta t)^2/2$. The most popular Runge-Kutta scheme is fourth-order accurate; for an ODE of the form

$$\frac{du}{dt} = F(u,t), \tag{2.102}$$

it can be written as

$$\left\{\begin{array}{rcl} u^*_{n+1/2} & = & u_n + \frac{\Delta t}{2}F(u_n,t_n) \\ u^{**}_{n+1/2} & = & u_n + \frac{\Delta t}{2}F(u^*_{n+1/2},t_n+\Delta t/2) \\ u^{***}_{n+1} & = & u_n + \Delta t F(u^{**}_{n+1/2},t_n+\Delta t) \\ u_{n+1} & = & u_n + \frac{\Delta t}{6}\Big[F(u^{***}_{n+1},t_n+\Delta t) + 2F(u^{**}_{n+1/2},t_n+\Delta t) \\ & & \qquad +2F(u^*_{n+1/2},t_n+\Delta t/2) + F(u_n,t_n)\Big], \end{array}\right. \tag{2.103}$$

An equivalent formulation, that requires storing only three, rather than four, words is

$$\left\{\begin{array}{lll} U = u_n; & G = U; & P = F(U,t_n) \\ U = U + \frac{\Delta t}{2}P; & G = P; & P = F(U,t_n+\Delta t/2) \\ U = U + \frac{\Delta t}{2}(P-G); & G = G/6; & P = F(U,t_n+\Delta t/2) \\ U = U + \Delta t P; & G = G - P; & P = F(U,t_n+\Delta t) + 2P \\ u_{n+1} = U + \Delta t(G + P/6). & & \end{array}\right. \tag{2.104}$$

The σ-root is $\sigma = 1 + \lambda\Delta t + (\lambda\Delta t)^2/2 + (\lambda\Delta t)^3/6 + (\lambda\Delta t)^4/24$ and the scheme is stable for $|\lambda\Delta t| \leq 2.79$.

There are also several formulations of third-order Rung-Kutta schemes. One that is often used because of its low storage requirement is:

$$\left\{\begin{array}{ll} U = u_n; & G = F(U,t_n) \\ U = U + \frac{\Delta t}{3}G; & G = -\frac{5}{9}G + F(U,t_n+\Delta t/3) \\ U = U + \frac{15\Delta t}{16}G; & G = -\frac{153}{128}G + F(U,t_n+3\Delta t/4) \\ u_{n+1} = U + \frac{8\Delta t}{15}G. & \end{array}\right. \tag{2.105}$$

The σ-root of RK3 schemes is $\sigma = 1+\lambda\Delta t+(\lambda\Delta t)^2/2+(\lambda\Delta t)^3/6$; the scheme above is stable for $|\lambda\Delta t| \leq 2.51$.

Implicit Euler scheme

The implicit Euler scheme,

$$u_{n+1} = u_n + \Delta t u'_{n+1}, \tag{2.106}$$

has a single root $\sigma = 1/(1 - \lambda\Delta t)$. The scheme is unconditionally stable, and first-order accurate.

Table 2.1: Stability bounds for Runge-Kutta schemes applied to the parabolic model equation.

Scheme	Order	Stability
RK1	$o(\Delta t)$	$\lvert\lambda\Delta t\rvert \leq 2$
RK2	$o(\Delta t^2)$	$\lvert\lambda\Delta t\rvert \leq 2$
RK3	$o(\Delta t^3)$	$\lvert\lambda\Delta t\rvert \leq 2.51$
RK4	$o(\Delta t^4)$	$\lvert\lambda\Delta t\rvert \leq 2.79$

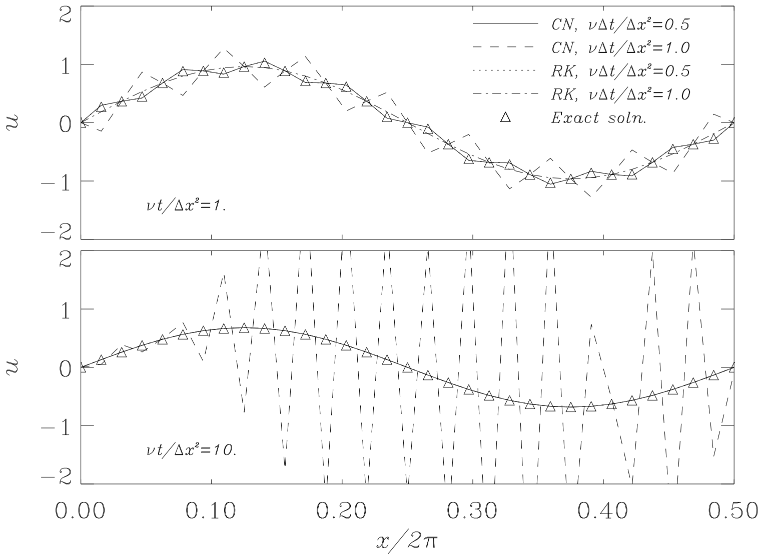

Figure 2.4: Comparison of the 3rd-order Runge-Kutta and Crank-Nicolson schemes for the parabolic model equation.

Crank-Nicolson scheme (trapezoidal rule)

The Crank-Nicolson scheme,

$$u_{n+1} = u_n + \frac{\Delta t}{2}\left(u'_{n+1} + u'_n\right), \tag{2.107}$$

is second-order accurate and its root is

$$\sigma = \frac{1 + \lambda\Delta t/2}{1 - \lambda\Delta t/2}. \tag{2.108}$$

It is also unconditionally stable.

Example 6:

Consider the diffusion equation

$$\frac{\partial u}{\partial t} - \nu\frac{\partial^2 u}{\partial x^2} = 0, \tag{2.109}$$

with $\nu = 1$, and $0 \leq x \leq 2\pi$, and $u(0,t) = u(2\pi,t) = 0$, and $u(x,0) = \sin 2x + \sin 16x$. This equation was integrated using the third-order Runge-Kutta scheme and the Crank-Nicolson scheme. A second-order central approximation was used for the spatial derivative, 65 equi-spaced grid points were used, and the time step was given by $\nu\Delta t/\Delta x^2 = 1$ and $\nu\Delta t/\Delta x^2 = 10$. The results are shown in Fig. 2.4. The two simulations with the small timestep give the same results, and both results are in good agreement with the exact solution at both times examined . When the larger timestep is used, however, the RK3 method gives an instability that is already evident at the early time, and becomes very significant at the later time. The Crank-Nicolson scheme, on the other hand, gives stable results. At the early time, however, the accuracy of the scheme is decreased.

2.4.6 Stability of time advancement schemes for the hyperbolic model equation

The eigenvalues for the hyperbolic model equation, given by (2.97), are imaginary; let, therefore $\lambda = i\omega$. If the discretizzation (2.95) is used, the largest magnitude of the eigenvalues is $|\lambda|_{max} \simeq |c|/\Delta x$.

Explicit Euler scheme

The σ-root for the explicit Euler scheme is complex: $\sigma = 1 + \lambda\Delta t = 1 + i\omega\Delta t$, and

$$|\sigma| = \left[1 + (\omega\Delta t)^2\right]^{1/2} > 1. \tag{2.110}$$

The scheme is, therefore, absolutely unstable.

Leapfrog scheme

The σ-roots for the leapfrog scheme are

$$\sigma = \lambda\Delta t \pm \sqrt{1 + (\lambda\Delta t)^2} = i\omega\Delta t \pm \sqrt{1 - (\omega\Delta t)^2}; \tag{2.111}$$

if $|\omega\Delta t| \leq 1$, their magnitudes are

$$|\sigma_1|^2 = |\sigma_2|^2 = 1. \tag{2.112}$$

If, on the other hand, $|\omega\Delta t| > 1$, the two roots are

$$\sigma_{1,2} = i\left[\omega\Delta t \pm \sqrt{(\omega\Delta t)^2 - 1}\right], \tag{2.113}$$

and their magnitudes are

$$|\sigma_1| = \left|\omega\Delta t + \sqrt{(\omega\Delta t)^2 - 1}\right| = |\omega\Delta t|\left|1 + \sqrt{1 - 1/(\omega\Delta t)^2}\right| \tag{2.114}$$

$$|\sigma_2| = \left|\omega\Delta t - \sqrt{(\omega\Delta t)^2 - 1}\right| = |\omega\Delta t|\left|1 - \sqrt{1 - 1/(\omega\Delta t)^2}\right| \tag{2.115}$$

Since $|\sigma_1| > 1$, the leapfrog scheme is only stable for $|\omega\Delta t| \leq 1$.

MacCormack predictor-corrector scheme

The σ-root of the MacCormack predictor-corrector scheme (2.77) is $\sigma = 1 + i\omega\Delta t - (\omega\Delta t)^2/2$; its amplitude is $|\sigma|^2 = 1 + (\omega\Delta t)^4 > 1$. The scheme is unstable.

Adams-Bashforth scheme

For the Adams-Bashforth scheme

$$\sigma_{1,2} = \frac{1}{2}\left[1 + \frac{3}{2}i\omega\Delta t \pm \sqrt{1 + i\omega\Delta t - \frac{9}{4}(\omega\Delta t)^2}\right]; \tag{2.116}$$

the scheme is weakly unstable.

Runge-Kutta schemes

The RK3 scheme is stable for $|\omega\Delta t| \leq 1.7$; the fourth-order RK4 scheme is stable for $|\omega\Delta t| \leq 2.8$.

Implicit Euler scheme

The implicit Euler scheme has a single root $\sigma = 1/(1 - i\omega\Delta t)$, whose magnitude is $|\sigma| = 1/\sqrt{1 + (\omega\Delta t)^2}$. The scheme is unconditionally stable, and first-order accurate.

Table 2.2: Stability bounds for Runge-Kutta schemes applied to the hyperbolic model equation.

Scheme	Order	Stability
RK1	$o(\Delta t)$	Unstable
RK2	$o(\Delta t^2)$	Unstable
RK3	$o(\Delta t^3)$	$\lvert\omega\Delta t\rvert \leq 1.7$
RK4	$o(\Delta t^4)$	$\lvert\omega\Delta t\rvert \leq 2.8$

Table 2.3: Stability of numerical schemes for the model equations. Central differences and Dirichlet boundary conditions.

Scheme	Parabolic Equation	Hyperbolic Equation
Explicit Euler	$\nu\Delta t/\Delta x^2 < 1/2$	Unstable
Leapfrog	Unstable	$\lvert c\Delta t/\Delta x\rvert < 1$
MacCormack	$\nu\Delta t/\Delta x^2 < 1/2$	Unstable
Adams-Bashforth	$\nu\Delta t/\Delta x^2 < 1/4$	Weakly unstable
RK3	$\nu\Delta t/\Delta x^2 < 0.62$	$\lvert c\Delta t/\Delta x\rvert < 1.7$
RK4	$\nu\Delta t/\Delta x^2 < 0.69$	$\lvert c\Delta t/\Delta x\rvert < 2.8$
Implicit Euler	Stable	Stable
Crank-Nicolson	Stable	Stable

Crank-Nicolson scheme (trapezoidal rule)

The σ-root for the Crank-Nicolson scheme is

$$\sigma = \frac{1 + i\omega\Delta t/2}{1 - i\omega\Delta t/2}. \tag{2.117}$$

Its magnitude is $|\sigma| = 1$. The scheme is unconditionally stable.

2.5 Stiffness

Consider the solution of a system of coupled ODE's (2.45), which is of the form

$$\mathbf{u}(t) = \sum_{j=1}^{M} \mathbf{x}_j C_j e^{\lambda_j t} + PS; \tag{2.118}$$

each j component of the transient decays at a different rate, depending on the magnitude of λ_j (which is a negative quantity). If the ratio

$$C_r = \frac{\max(|\lambda_j|)}{\min(|\lambda_j|)} \tag{2.119}$$

(known as the *condition number*) is $C_r > 100$, the system is said to be *stiff*. A generic equation in wave space

$$\frac{dw_m}{dt} = \lambda_m w_m + g_m, \tag{2.120}$$

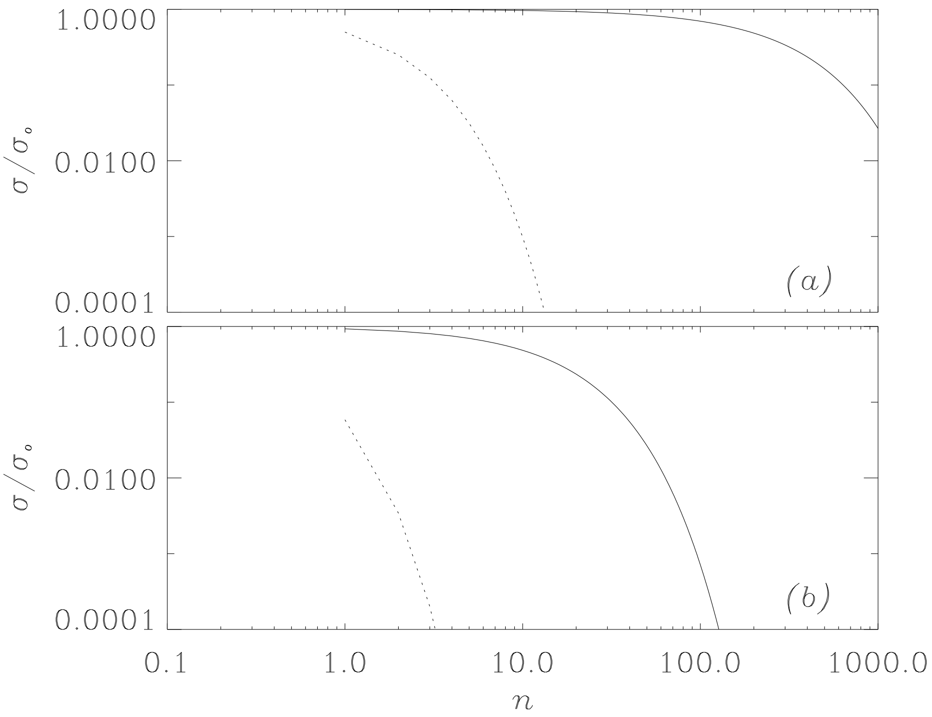

Figure 2.5: Decay of transient components for the parabolic model equation. $\nu = 1$, $M = 32$. ——— σ_{max}; ········ σ_{min}. (a) MacCormack scheme, $\Delta t = 0.0385531$; (b) Crank-Nicolson scheme, $\Delta t = 0.385531$.

has the solution

$$w_m^n = \sum_{k=1}^{K} \gamma_k \left(\sigma_k\right)^n + PS \tag{2.121}$$

where σ_k are the roots of the characteristic polynomial $P(\sigma) = 0$. If the ratio $|\sigma|_{min}/|\sigma|_{max}$ is large, tke various components of the solution will go to zero at very different rates. For the parabolic model equation, for instance, the eigenvalues are given by (2.96), and the condition number is $C_r = 4/\Delta x^2$; the system is stiff for $M > 15$.

If one uses an explicit MacCormack scheme, for instance (with $\nu = 1$, $M = 32$ and $\Delta t = 0.0385531$) then $\sigma_{max} = 0.996384$ and $\sigma_{min} = 0.500065$ (notice that σ_{min} does not correspond to either λ_{min} or λ_{max}). Under those conditions, it takes almost 2000 timesteps for the slowest component of the transient to drop to 0.0001 of its initial value (Figure 2.5a). With an implicit scheme, which allows a much larger timestep, the transient part of the solution decays much faster. Of course, if a time accurate solution is desired, initially a small timestep must be used with an implicit scheme as well, to resolve the transient. Then, as the rapid modes decay, the timestep can be progressively increased.

Example 7:

Consider again the parabolic model equation in the previous example, with $u(x, 0) = \exp(-16x^2)$ and $u(0, t) = 1$, $u(2\pi, t) = 0$. In Fig. 2.6 the CPU time required to reach the steady-state solution with Crank-Nicolson and 3rd-order Runge-Kutta schemes is compared. The stability limitation of the explicit scheme results in small timesteps; after 900 timesteps, the steady state had not yet been achieved. With the implicit Crank-Nicolson scheme, on the other hand, a larger timestep could be used, resulting in convergence to the steady state after fewer steps; even if the CPU time per step is less for an explicit scheme than for an implicit one, the possibility to use a larger timestep results in considerably faster achievement of the steady state.

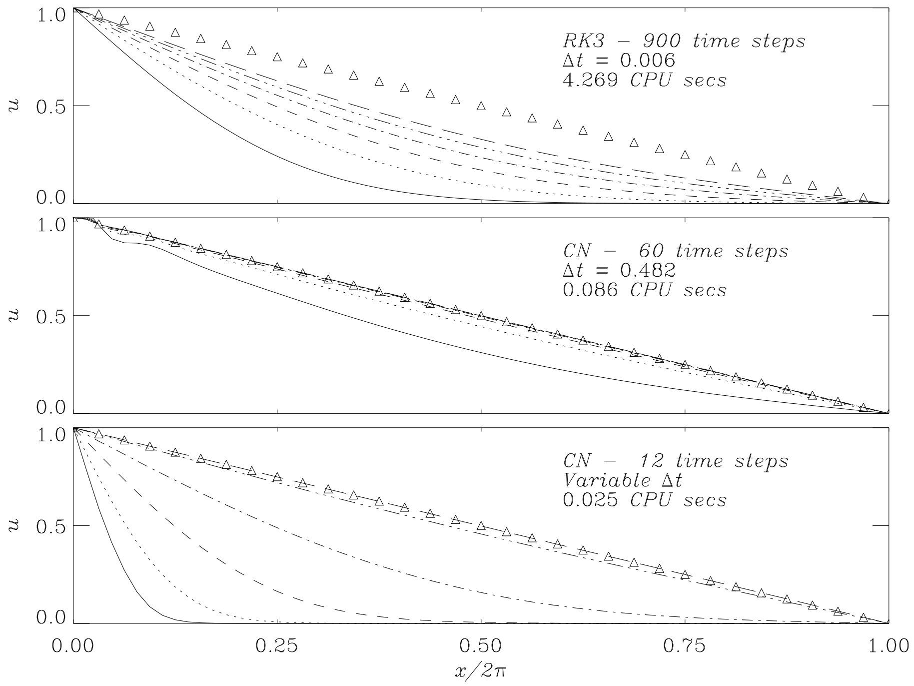

Figure 2.6: Comparison of steady-state solutions obtained with Crank-Nicolson and 3rd-order Runge-Kutta schemes. The lines are plotted every 6th time-step, and the symbols represent the steady-state solution. Calculations carried out on a $200MHz$ Pentium.

2.6 General considerations

For linear problems, implicit time-advancement schemes are considerably faster than explicit ones, and should be preferred, unless the physical time-scale of the problem under consideration is so short that using a small timestep carries no penalty. If only a steady-state is desired, then the largest possible time-step should be used (it is, however, desirable to increase the time-step slowly to damp evenly all the solution eigenvectors).

When choosing a time-advancement scheme, attention should be paid to the spatial discretization as well, which affects the overall error. Consider the diffusion equation

$$\frac{\partial u}{\partial t} - \nu \frac{\partial^2 u}{\partial x^2} = 0, \tag{2.122}$$

with $\nu = 1$, and $0 \leq x \leq 2\pi$. Let $u(x,0) = u_o(x)$, with $u_o(x) = \exp(-16x^2) + 2\sin 8x$ and $u(0,t) = u_o(0)$, $u(2\pi,t) = u_o(2\pi)$. This equation is solved using the Crank-Nicolson scheme using a fixed $\Delta t = 0.001$, and a variable number of grid points in x. The error at $t = 0.05$ is shown in Fig. 2.7. The expected Δx^2 behavior is observed. The fact that on the coarsest grid the error is less than on the next one is due to the fact that the grid points occur at nodes of the sine wave.

Compare now the solution at $t = 0.1$ for different timesteps Δt and a fixed number of grid points $N = 257$ (Fig. 2.8). The solution decreases like Δt^2 for $0.003 < \Delta t < 0.02$, but eventually flattens out. The flattening out is due to the fact that the error due to the spatial discretization becomes larger than that due to the spatial one. This is illustrated in Fig. 2.9, in which the calculation is repeated for several different grids: as the grid is coarsened, the accuracy limit is reached for a larger value of Δt.

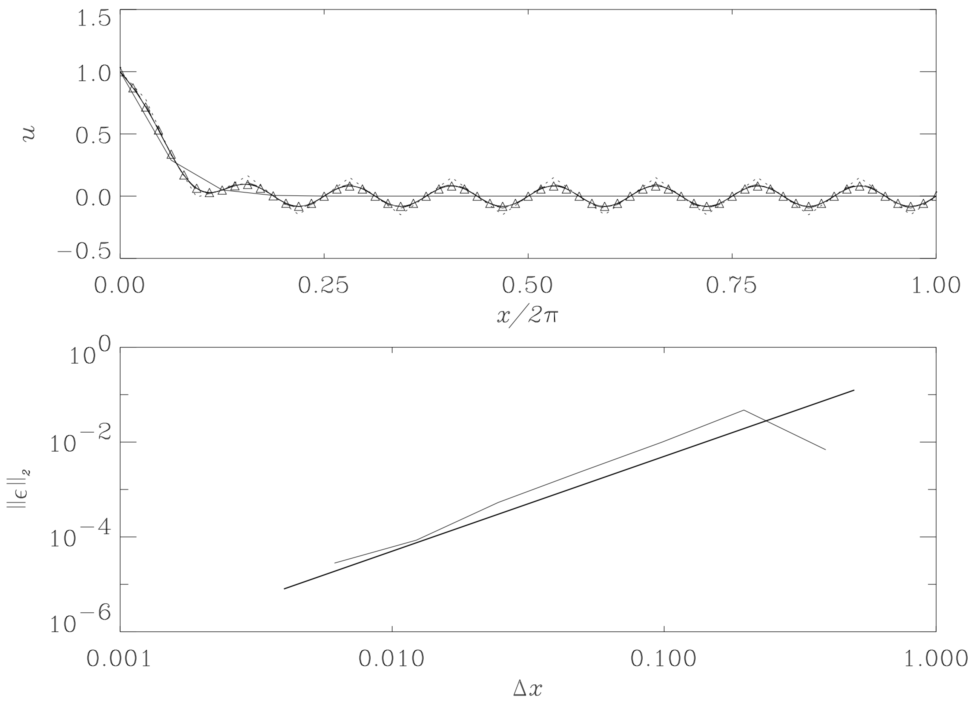

Figure 2.7: Effect of grid resolution on the solution accuracy.

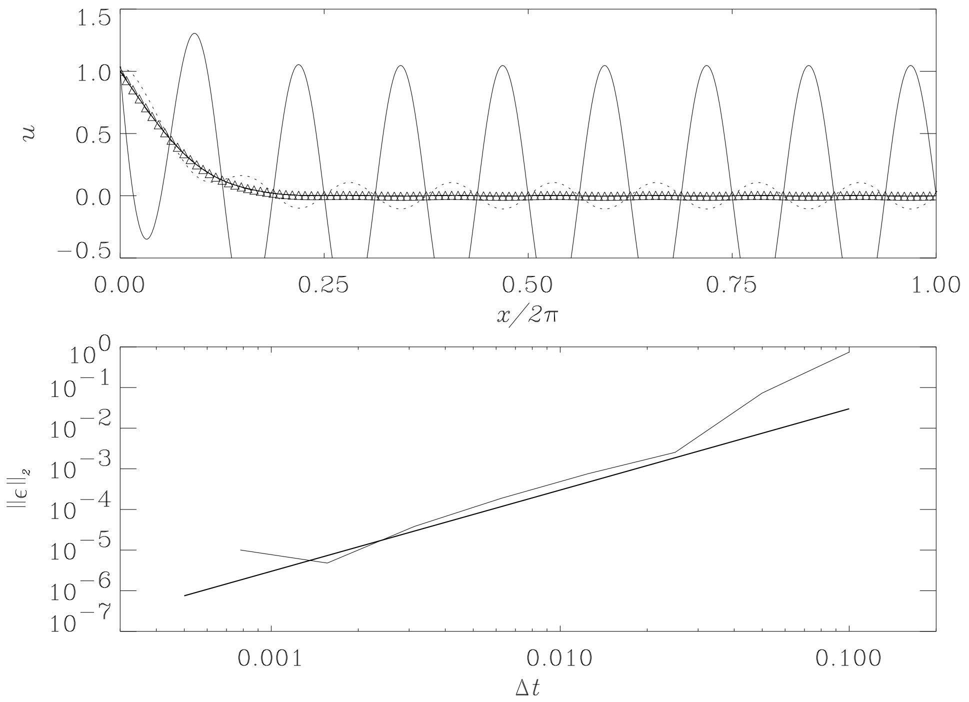

Figure 2.8: Effect of timestep on the solution accuracy.

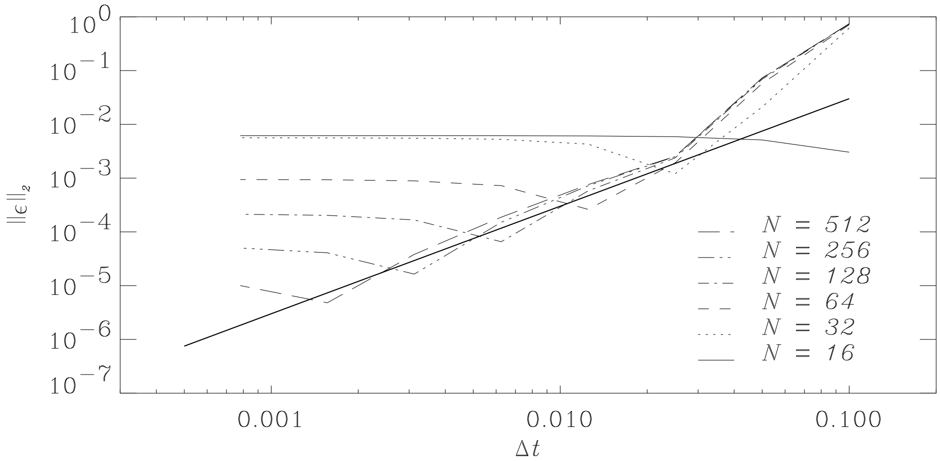

Figure 2.9: Effect of timestep and spatial resolution on the solution accuracy.