Chapter 8

Time Discrete Approximation of Deterministic Differential Equations

In this chapter we summarize the basic concepts and assertions of the numerical analysis of initial value problems for deterministic ordinary differential equations. The material is presented so as to facilitate generalizations to the stochastic setting and to highlight the differences between the deterministic and stochastic cases.

8.1 Introduction

In general it is not possible to find explicitly the solution $x = x(t; t_0, x_0)$ of an initial value problem (IVP)

$$\dot{x} = \frac{dx}{dt} = a(t, x), \qquad x(t_0) = x_0 \tag{1.1}$$

for the deterministic differential equations that occur in many scientific and technological models. Even when such a solution can be found, it may be only in implicit form or too complicated to visualize and evaluate numerically. Necessity has thus lead to the development of methods for calculating numerical approximations to the solutions of such initial value problems. The most widely applicable and commonly used of these are the *time discrete approximation* or *difference methods*, in which the continuous time differential equation is replaced by a discrete- time difference equation generating values $y_1, y_2, \ldots, y_n, \ldots$ to approximate $x(t_1; t_0, x_0)$, $x(t_2; t_0, x_0)$, $\ldots$, $x(t_n; t_0, x_0)$, $\ldots$ at given discretization times $t_0 < t_1 < t_2 < \cdots < t_n < \cdots$. These approximations should be quite accurate, one hopes, if the time increments $\Delta_n = t_{n+1} - t_n$ for $n = 0, 1, 2, \ldots$ are sufficiently small. As a background for the development of discretization methods for stochastic differential equations, in this chapter we shall review the basic difference methods used for ordinary differential equations and consider some related issues such as their convergence and stability.

The simplest difference method for the IVP (1.1) is the *Euler method*

$$y_{n+1} = y_n + a(t_n, y_n)\,\Delta_n \tag{1.2}$$

for a given time discretization $t_0 < t_1 < t_2 < \cdots < t_n < \cdots$ with increments $\Delta_n = t_{n+1} - t_n$ where $n = 0, 1, 2, \ldots$. Once the initial value y_0 has been specified, usually $y_0 = x_0$, the approximations $y_1, y_2, \ldots, y_n, \ldots$ can be calculated by recursively applying formula (1.2). We can derive (1.2) by freezing the right

hand side of the differential equation over the time interval $t_n \le t < t_{n+1}$ at the value $a(t_n, y_n)$ and then integrating to obtain the tangent to the solution $x(t; t_n, y_n)$ of the differential equation with the initial value $x(t_n) = y_n$. The difference

$$l_{n+1} = x(t_{n+1}; t_n, y_n) - y_{n+1}, \tag{1.3}$$

which is generally not zero, is called the *local discretization error* for the nth time step. This is usually not the same as the *global discretization error*

$$e_{n+1} = x(t_{n+1}; t_0, x_0) - y_{n+1} \tag{1.4}$$

for the same time step, which is the error with respect to the sought solution of the original IVP (1.1). Nevertheless, we can use the local discretization error to estimate the global discretization error. It must be emphasized that (1.3) and (1.4) assume that we can perform all arithmetic calculations exactly. In practice, both we and digital computers are restricted to a finite number of decimal places when doing calculations and roundoff all excess decimal places, thus introducing *roundoff error*. We shall denote it by r_{n+1} for the nth time step.

The key to estimating the size of discretization errors is the Taylor formula with remainder, which for a twice continuously differentiable function $x = x(t)$ is

$$x(t_{n+1}) = x(t_n) + \dot{x}(t_n)\,\Delta_n + \frac{1}{2!}\,\ddot{x}(\theta_n)\,\Delta_n^2 \tag{1.5}$$

for some θ_n satisfying $t_n < \theta_n < t_{n+1}$. For $x(t) \equiv x(t; t_n, y_n)$, the solution of the differential equation with $x(t_n) = y_n$, we thus have

$$x(t_{n+1}) = x(t_n) + a(t_n, x(t_n))\,\Delta_n + \frac{1}{2!}\,\ddot{x}(\theta_n)\,\Delta_n^2. \tag{1.6}$$

Since $x(t_n) = y_n$ here, on subtracting (1.5) from (1.6) we find that the local discretization error (1.3) has the form

$$l_{n+1} = \frac{1}{2!}\,\ddot{x}(\theta_n)\,\Delta_n^2.$$

If we knew that $|\ddot{x}(t)| < M$ for all t in some interval $[t_0, T]$ of interest, then we would have the estimate

$$|l_{n+1}| \le \frac{1}{2!}\,M\,\Delta_n^2 \tag{1.7}$$

for any discretization time subinterval $[t_n, t_{n+1}]$ with $t_0 \le t_n < t_{n+1} \le T$. We can obtain such a bound on $\ddot{x}$ using the fact that

$$\begin{aligned}
\ddot{x} = \frac{d}{dt}\,\dot{x} &= \frac{d}{dt}\,a(t, x(t)) \\
&= \frac{\partial a}{\partial t}(t, x(t)) + \frac{\partial a}{\partial x}(t, x(t))\frac{dx}{dt} \\
&= \frac{\partial a}{\partial t}(t, x(t)) + \frac{\partial a}{\partial x}(t, x(t))\,a(t, x(t)).
\end{aligned}$$

If a, $\frac{\partial a}{\partial t}$ and $\frac{\partial a}{\partial x}$ are continuous and if we knew that all solutions $x(t)$ under consideration remained in some closed and bounded set C for all $t_0 < t < T$, we could use

$$M = \max\left|\frac{\partial a}{\partial t}(t, x)\right| + \max\left|\frac{\partial a}{\partial x}(t, x)\,a(t, x)\right|,$$

where the maxima are taken over $(t, x) \in [t_0, T] \times C$, in the inequality (1.7). This particular value of M will usually give a gross overestimate, but from (1.7) we can see that the local discretization error for the Euler method (1.2) is of order Δ_n^2.

To estimate the global discretization error we shall assume, for simplicity, that $a = a(t, x)$ satisfies a uniform Lipschitz condition

$$|a(t, x) - a(t, y)| \le K\,|x - y|$$

and that the time discretization involves equidistant time instants $t_n = t_0 + n\Delta$ for $n = 0, 1, 2, \ldots$. Applying the Taylor formula (1.5) to the solution $x(t) \equiv x(t; t_0, x_0)$ we have (1.6) with $\Delta_n \equiv \Delta$, but now $x(t_n) \ne y_n$ in general. Subtracting (1.2) then gives

$$e_{n+1} = e_n + \{a(t_n, x(t_n)) - a(t_n, y_n)\}\,\Delta + \frac{1}{2}\,\ddot{x}(\theta_n)\,\Delta^2 \tag{1.8}$$

and, using the Lipschitz condition on a and a bound on $\ddot{x}$, thus

$$|e_{n+1}| \le |e_n| + K\,|e_n|\,\Delta + \frac{1}{2}\,M\,\Delta^2.$$

We can then show by induction that the difference inequality

$$|e_{n+1}| \le (1 + K\,\Delta)\,|e_n| + \frac{1}{2}\,M\,\Delta^2 \tag{1.9}$$

with $e_0 = x_0 - y_0 = 0$ implies that

$$|e_{n+1}| \le \frac{1}{2}\left(\frac{(1 + K\Delta)^n - 1}{(1 + K\Delta) - 1}\right) M\,\Delta^2 \le \frac{1}{2}\,(e^{nK\Delta} - 1)\,\frac{M}{K}\,\Delta,$$

since $(1 + K\Delta)^n \le e^{nK\Delta}$. Hence the global discretization error for the Euler method (1.2) satisfies

$$|e_{n+1}| \le \frac{1}{2}\left(e^{K(T - t_0)} - 1\right)\frac{M}{K}\,\Delta \tag{1.10}$$

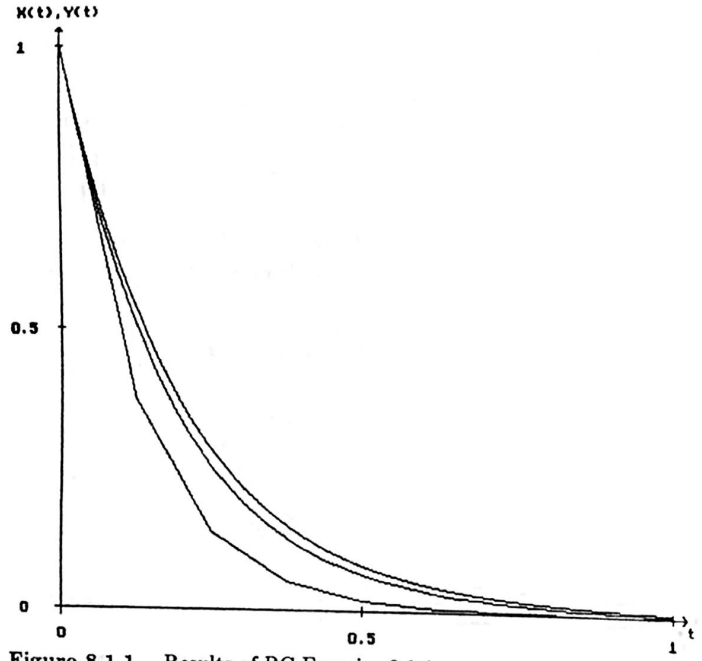

Figure 8.1.1 Results of PC-Exercise 8.1.1.

for discretization times $t_n = t_0 + n\Delta \leq T$. It is obviously one power of Δ less than the local discretization error.

PC-Exercise 8.1.1 *Apply the Euler method (1.2) to the IVP*

$$\frac{dx}{dt} = -5x, \qquad x(0) = 1,$$

with time steps of equal length $\Delta = 2^{-3}$ *and* 2^{-5} *over the time interval* $0 \leq t \leq 1$. *Plot the results and the exact solution* $x(t) = e^{-5t}$ *against* t.

In Figure 8.1.1 the upper curve corresponds to the exact solution and the lower and middle ones to the Euler method with step sizes $\Delta = 2^{-3}$ and 2^{-5}, respectively. We see that the global discretization error is smaller for the smaller step size.

In the next PC-Exercise we shall look more closely at the dependence of the global truncation error on the step size. For this we recall that a function $f(\Delta) = A\Delta^\gamma$ becomes linear in logarithmic coordinates, that is

$$\log_a f(\Delta) = \log_a A + \gamma \log_a \Delta$$

for logarithms to the base $a \neq 1$. In comparative studies we shall take time steps of the form $\Delta = a^{-n}$ for $n = 1, 2, \ldots$ and $a > 1$. We shall usually halve the time step successively, in which case logarithms to the base $a = 2$ will be

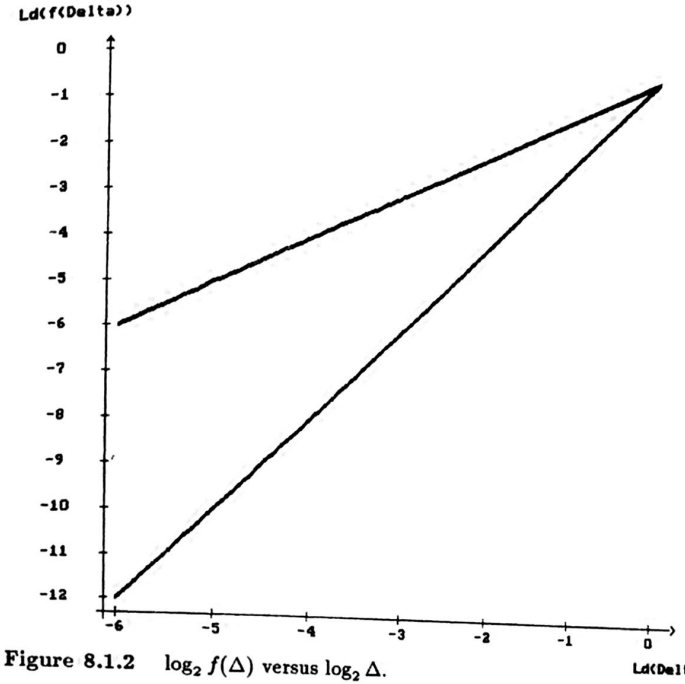

Figure 8.1.2 $\log_2 f(\Delta)$ versus $\log_2 \Delta$.

appropriate. In Figure 8.1.2 we plot the values of $\log_2 f(\Delta)$ against $\log_2 \Delta$ for the two functions $f(\Delta) = \Delta$ and $f(\Delta) = \Delta^2$ with $\Delta = 2^{-n}$ for $n = 0, 1, \ldots, 6$.

PC-Exercise 8.1.2 *For the IVP in PC-Exercise 8.1.1 calculate the global discretization error at time* $t = 1$ *for the Euler method with time steps of equal length* $\Delta = 1, 2^{-1}, 2^{-2}, \ldots, 2^{-13}$, *rounding off to 5 significant digits. Plot the logarithm to the base 2 of these errors against* $\log_2 \Delta$ *and determine the slope of the resulting curve.*

From Figure 8.1.3 we see that the calculated global discretization error $\tilde{e}_{n+1}$ for the Euler method is proportional to the step size Δ for $\Delta \leq 2^{-3}$, provided Δ is not too small. For $\Delta \leq 2^{-11}$ the error $\tilde{e}_{n+1}$ begins to increase here as Δ is further decreased. This does not contradict (1.10), but occurs because $\tilde{e}_{n+1}$ also includes the roundoff error. To estimate $\tilde{e}_{n+1}$ we must add the roundoff error r_n to the right hand side of (1.9). For $|r_n| \leq R$ for each n we then obtain the estimate

$$|\tilde{e}_{n+1}| \leq \frac{1}{2}\left(e^{K(T-t_0)} - 1\right)\left(\frac{2R}{K}\frac{1}{\Delta} + \frac{M}{K}\Delta\right)$$

instead of (1.10). For very small Δ the reciprocal term dominates the bound. While it represents the worst case scenario, this bound is still indicative of the cummulative effect of roundoff error, since for smaller Δ more calculations are

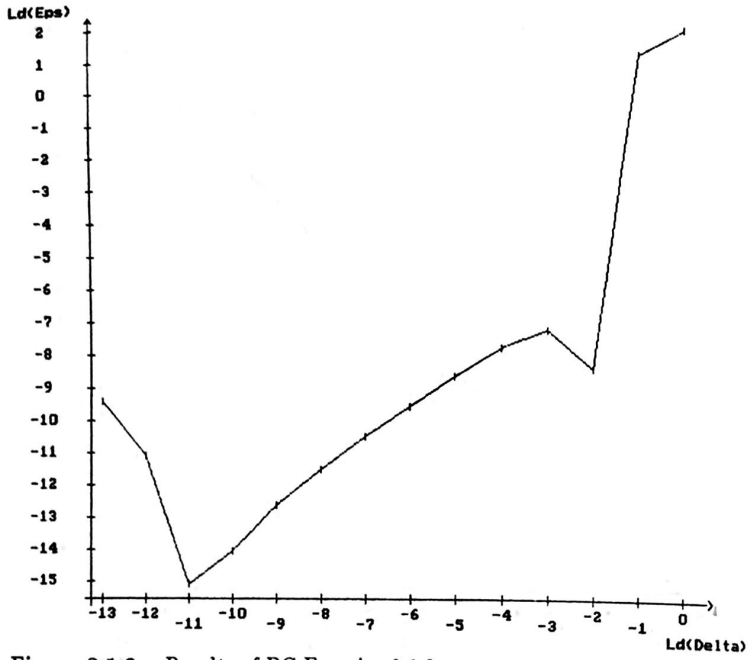

Figure 8.1.3 Results of PC-Exercise 8.1.2.

required to reach the given end time. We shall look more closely at randomly distributed roundoff errors in Section 4.

The presence of roundoff error means there is a minimum step size Δ_{min} for each initial value problem, below which we cannot improve the accuracy of the approximations calculated by means of the Euler method. To obtain a more accurate approximation we need to use another method with a higher order discretization error. The Taylor expansion provides a systematic framework for developing and investigating such schemes. For the rest of this section we shall, however, continue with a more heuristic approach.

For the Euler method we simply froze the right hand side of the differential equation at the value $a(t_n, y_n)$ at the beginning of each discretization subinterval $t_n < t < t_{n+1}$. We should obtain a more accurate approximation if we included more information from elsewhere in the subinterval. For instance, we could use the average of the values at both end points, in which case we have the *trapezoidal method*

$$y_{n+1} = y_n + \frac{1}{2}\{a(t_n, y_n) + a(t_{n+1}, y_{n+1})\}\,\Delta_n. \tag{1.11}$$

This is called an *implicit* scheme because the unknown quantity y_{n+1} appears in both sides of (1.11) and, in general, cannot be isolated algebraically. To circumvent this difficulty we could use the Euler method (1.2) to approximate

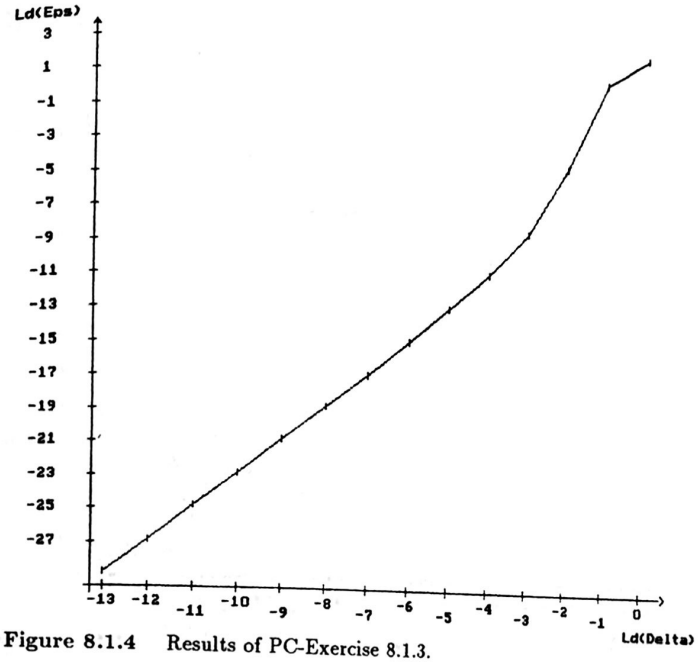

Figure 8.1.4 Results of PC-Exercise 8.1.3.

the y_{n+1} term on the right hand side of (1.11). Then we obtain the *modified trapezoidal method*

$$\begin{aligned} \bar{y}_{n+1} &= y_n + a(t_n, y_n)\,\Delta_n \\ y_{n+1} &= y_n + \frac{1}{2}\{a(t_n, y_n) + a(t_{n+1}, \bar{y}_{n+1})\}\,\Delta_n, \end{aligned}$$

or

$$y_{n+1} = y_n + \frac{1}{2}\{a(t_n, y_n) + a(t_{n+1}, y_n + a(t_n, y_n)\Delta_n)\}\,\Delta_n, \tag{1.12}$$

which is also known as the *improved Euler* or *Heun method*. It is a simple example of a *predictor-corrector method* with the predictor $\bar{y}_{n+1}$ inserted into the corrector equation to give the next iterate y_{n+1}.

Both the trapezoidal and the modified trapezoidal methods have local discretization errors of third order in Δ_n. This can be verified by comparing the Taylor formula with third order remainder of the solution $x(t; t_0, x_0)$ of the differential equation and (1.11) or (1.12) with $a(t_{n+1}, y_{n+1})$ or $a(t_{n+1}, \bar{y}_{n+1})$ expanded about (t_n, y_n). The global discretization error for both methods is of second order in $\Delta = \max_n \Delta_n$, which is again one order less than the local discretization error.

PC-Exercise 8.1.3 *Repeat PC-Exercise 8.1.2 with the usual arithmetic of the PC for the modified trapezoidal method (1.12). Compare the results with those for the Euler method.*

Figure 8.1.4 indicates that the global truncation error for the modified trapezoidal method is proportional to Δ^2 for $\Delta \leq 2^{-3}$, whereas that of the Euler method is proportional to Δ.

Exercise 8.1.4 *Show that the local discretization errors of the trapezoidal and modified trapezoidal methods are of third order in Δ_n.*

Even higher order difference methods can be derived by using more accurate approximations of the right hand side of the differential equation over each discretization subinterval $t_n < t < t_{n+1}$. These are called *one-step methods* if they involve only the values y_n and y_{n+1} in addition, of course, to t_n and Δ_n. Explicit one-step methods are usually written in the general form

$$y_{n+1} = y_n + \Psi(t_n, y_n, \Delta_n)\,\Delta_n, \tag{1.13}$$

for some function $\Psi = \Psi(t, x, \Delta)$, which is called the *increment function*. We say that it is a pth *order* method if its global discretization error is bounded by the pth power of $\Delta = \max_n \Delta_n$. If the functions a and Ψ are sufficiently smooth it can be shown that a $(p+1)$th order local discretization error implies a pth order global discretization error; see Theorem 8.3.2 in the next section. For example, the Euler method (1.2) is a 1st order one-step scheme with $\Psi(t, x, \Delta) = a(t, x)$ and the Heun method (1.12) is a 2nd order one-step scheme with

$$\Psi(t, x, \Delta) = \frac{1}{2}\left\{a(t, x) + a\left(t + \Delta, x + a(t, x)\,\Delta\right)\right\}.$$

The function Ψ cannot be chosen completely arbitrarily. For instance, it should be *consistent* with the differential equation, that is satisfy

$$\lim_{\Delta \downarrow 0} \Psi(t, x, \Delta) = a(t, x),$$

if the values calculated from (1.13) are to converge to the desired solution of the differential equation. The Euler method is obviously consistent and the Heun method is consistent when $a(t, x)$ is continuous in both variables.

Some difference methods achieve higher accuracy by using information from previous discretization subintervals when calculating y_{n+1} on $t_n < t < t_{n+1}$. In these *multi-step* methods y_{n+1} depends on the previous k values y_n, y_{n-1}, $\cdots$, y_{n-k} for some $k > 1$. An example involving equal time steps Δ is the *3-step Adams-Bashford method*

$$y_{n+1} = y_n + \frac{1}{12}\left\{23\,a(t_n, y_n) - 16\,a(t_{n-1}, y_{n-1}) + 5\,a(t_{n-2}, y_{n-2})\right\}\Delta, \tag{1.14}$$

which turns out to have third order global discretization error. It is derived by replacing the right hand side of the differential equation on the time interval $t_n < t < t_{n+1}$ by the unique cubic polynomial passing through the points $(t_n, a(t_n, y_n))$, $(t_{n-1}, a(t_{n-1}, y_{n-1}))$ and $(t_{n-2}, a(t_{n-2}, y_{n-2}))$. Notice that these are the only points where the function a has to be evaluated. Since the evaluations from the previous steps can be saved for use in the current step, this method essentially requires only the value of $a(t_n, y_n)$ to be calculated in

the nth step, once the procedure has been started. In order to start it we must specify the first 3 values y_0, y_1 and y_2, which is usually done be calculating y_1 and y_2 with a one-step method starting at y_0.

PC-Exercise 8.1.5 *Repeat PC-Exercise 8.1.3 using the 3-step Adams-Bashford method (1.14) with the Heun method (1.12) as its starting routine.*

Exercise 8.1.6 *Show that the 3-step Adams-Bashford method (1.14) has fourth order local discretization error.*

Finally, we note that we can sometimes obtain higher order accuracy from a one-step scheme by the method of extrapolation. For example, suppose we use the Euler scheme (1.2) with N equal time steps $\Delta = T/N$ on the interval $0 \leq t \leq T$. If $x(T)$ is the true value at time T and $y_N(\Delta)$ the corresponding value from the Euler scheme, then we have

$$y_N(\Delta) = x(T) + e(T)\,\Delta + O(\Delta^2), \tag{1.15}$$

where we have written the global truncation error as $e(T)\,\Delta + O(\Delta^2)$. If, instead, we use the Euler scheme with $2N$ time steps of equal length $\Delta/2$, then we have

$$y_{2N}\left(\frac{1}{2}\Delta\right) = x(T) + \frac{1}{2}e(T)\,\Delta + O(\Delta^2). \tag{1.16}$$

We can eliminate $e(T)$ from (1.15) and (1.16) to obtain

$$x(T) = 2y_{2N}\left(\frac{1}{2}\Delta\right) - y_N(\Delta) + O(\Delta^2).$$

Thus we have a second order approximation

$$Z_N(\Delta) = 2y_{2N}\left(\frac{1}{2}\Delta\right) - y_N(\Delta) \tag{1.17}$$

for $x(t)$ from the first order Euler scheme. Of course, this requires our repeating the Euler scheme calculations for half the original time step, but for complicated differential equations it may involve fewer and simpler calculations than a second order one-step scheme. This method is known as *Richardson* or *Romberg extrapolation*. It can also be applied to more general one-step schemes and to multi-step schemes.

PC-Exercise 8.1.7 *Compare the error of the Euler and Richardson extrapolation approximations of $x(1)$ for the solution of the initial value problem*

$$\frac{dx}{dt} = -x, \qquad x(0) = 1$$

for equal time steps $\Delta = 2^{-3}, 2^{-4}, \ldots, 2^{-10}$. Plot $\log_2$ of the errors against $\log_2 \Delta$.

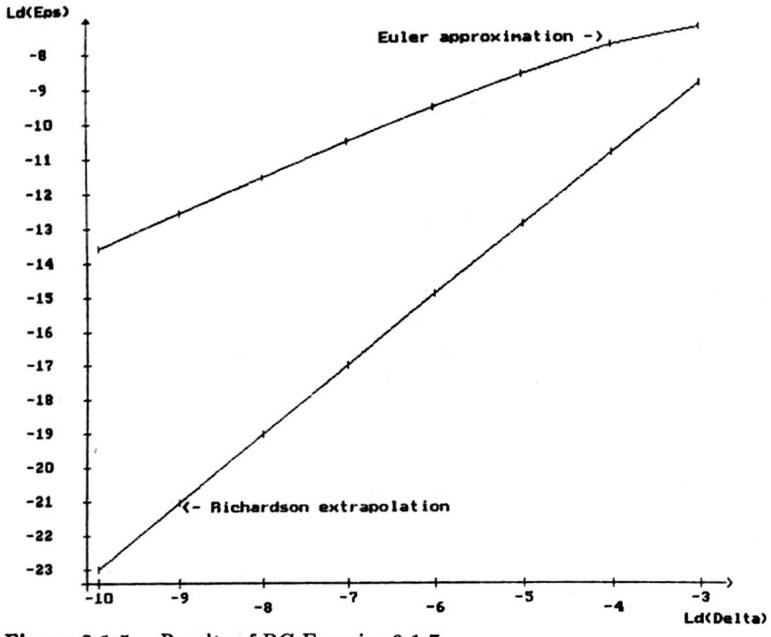

Figure 8.1.5 Results of PC-Exercise 8.1.7.

8.2 Taylor Approximations and Higher Order Methods

A one-step difference method with global discretization error of order p is readily suggested locally by the *Taylor formula with* $(p+1)$th *order remainder*

$$(2.1) \qquad x(t_{n+1}) = x(t_n) + \frac{dx}{dt}(t_n)\,\Delta_n + \cdots + \frac{1}{p!}\frac{d^p x}{dt^p}(t_n)\,\Delta_n^p + \frac{1}{(p+1)!}\frac{d^{p+1}x}{dt^{p+1}}(\theta_n)\,\Delta_n^{p+1},$$

where $t_n < \theta_n < t_{n+1}$ and $\Delta_n = t_{n+1} - t_n$, for any $p = 1, 2, 3, \ldots$. We can apply this formula to a solution $x(t)$ of the differential equation

$$(2.2) \qquad \frac{dx}{dt} = a(t,x)$$

if the function $a = a(t,x)$ and its partial derivatives of orders up to and including p are continuous as this assures that $x(t)$ is $p+1$ times continuously differentiable. Indeed, from (2.2) and the chain rule, by repeatedly differentiating $a(t,x(t))$ we have

$$\frac{dx}{dt} = a, \qquad \frac{d^2x}{dt^2} = a_t + a_x\,a,$$

$$\frac{d^3x}{dt^3} = a_{tt} + 2a_{tx}\,a + a_{xx}\,a^2 + a_t\,a_x + a_x^2\,a,$$

and so on, where we have used subscripts to indicate partial derivatives. Evaluating these expressions at (t_n, y_n) for the solution $x(t) = x(t; t_n, y_n)$ of (2.2) and omitting the remainder term in (2.1) we obtain a one-step method for y_{n+1}, which we shall call the pth *order truncated Taylor method*. This method obviously has local discretization error of order $p+1$ and can be shown to have global discretization error of order p. The 1st order truncated Taylor method is just the Euler method (1.2), the 2nd *order truncated Taylor method* is

$$(2.3) \qquad y_{n+1} = y_n + a(t_n,y_n)\,\Delta_n + \frac{1}{2!}\left\{a_t(t_n,y_n) + a_x(t_n,y_n)a(t_n,y_n)\right\}\Delta_n^2$$

and the 3rd *order truncated Taylor method*

$$(2.4) \qquad y_{n+1} = y_n + a\,\Delta_n + \frac{1}{2!}\left\{a_t + a_x\,a\right\}\Delta_n^2 + \frac{1}{3!}\left\{a_{tt} + 2a_{tx}\,a + a_{xx}\,a^2 + a_t\,a_x + a_x^2\,a\right\}\Delta_n^3,$$

where a and its partial derivatives are evaluated at (t_n, y_n).

PC-Exercise 8.2.1 *Use the 2nd order truncated Taylor method (2.3) with equal length time steps* $\Delta = 2^{-3}, \ldots, 2^{-10}$ *to calculate approximations to the solution* $x(t) = 2/(1+e^{-t^2})$ *of the initial value problem*

$$\frac{dx}{dt} = t\,x\,(2-x), \qquad x(0) = 1$$

over the interval $0 \le t \le 0.5$. *Repeat the calculations using the 3rd order truncated Taylor method (2.4). Plot* $\log_2$ *of the global discretization errors at time* $t = 0.5$ *against* $\log_2 \Delta$.

In Figure 8.2.1 the upper curve with slope 2 corresponds to the 2nd order truncated Taylor method and the lower one with slope 3 to the 3rd order truncated Taylor method.

The coefficients in higher order truncated Taylor methods soon become unwieldy and error prone to determine for all but the simplest differential equations. Moreover, considerable computational effort is needed for the evaluation of these coefficients, so these methods are not particularly efficient. They are almost never used in practice, except to provide a reference point for the development and analysis of other, more efficient higher order difference schemes.

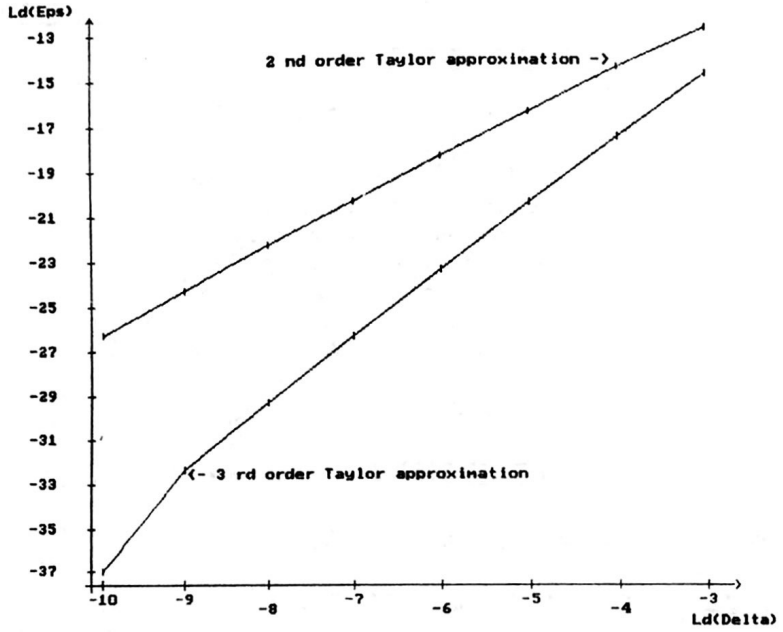

Figure 8.2.1 Results of PC-Exercise 8.2.1.

One way of simplifying the coefficients in a truncated Taylor method is to replace the partial derivatives by their forward difference quotients, for example replacing $a_t(t_n, y_n)$ and $a_x(t_n, y_n)$ by

$$\frac{a(t_{n+1}, y_n) - a(t_n, y_n)}{\Delta_n} \quad \text{and} \quad \frac{a(t_n, y_{n+1}) - a(t_n, y_n)}{y_{n+1} - y_n},$$

respectively. This will lead to an implicit scheme because y_{n+1} appears on both sides of the recursion formula and, generally, cannot be solved for algebraically. As in the trapezoidal method (1.11) we could use the Euler method (1.2), say, to predict a value of y_{n+1} to use in the terms on the right hand side of the formula, thus obtaining an explicit method. For the 2nd order truncated Taylor method this results first in the trapezoidal method (1.11) and then in the Heun method (1.12). The higher order coefficients will usually be considerably more complicated, but are at least derivative free.

The standard procedure with most one-step methods

$$y_{n+1} = y_n + \Psi(t_n, y_n, \Delta_n)\,\Delta_n \tag{2.5}$$

is first to derive the function $\Psi = \Psi(t, x, \Delta)$ by an heuristic argument and then to compare the method with a truncated Taylor method or expansion to determine the order of its discretization error. The *Runge-Kutta methods* are typical of this approach. For what will turn out to be the 2nd order methods of this type, Ψ is chosen with the form

$$\Psi(t, x, \Delta) = \alpha\, a(t, x) + \beta\, a\,(t + \gamma\,\Delta, x + \gamma\, a(t, x)\,\Delta), \tag{2.6}$$

for certain constants α, β and γ, which represents an weighted averaging of the right hand side of the differential equation (2.2) over two points. Expanding the second term about (t, x), we obtain

$$\begin{aligned}\Psi \;=\; & (\alpha + \beta)\, a + \gamma\beta\,(a_t + a_x\, a)\,\Delta \\ & + \frac{1}{2}\,\gamma^2\beta\,(a_{tt} + 2a_{tx}\, a + a_{xx}\, a^2)\,\Delta^2 \\ & + \text{higher order terms},\end{aligned}$$

where a and its partial derivatives are all evaluated at (t, x). Hence, subtracting (2.5) with this expansion for Ψ evaluated at (t_n, y_n, Δ_n) from the 3rd order truncated Taylor method (2.4) we get

$$\begin{aligned}(1 - \alpha - \beta)\, a\,\Delta_n \;+\; & \left(\frac{1}{2!} - \gamma\beta\right)(a_t + a_x\, a)\,\Delta_n^2 \\ & + \frac{1}{2}\left(\frac{1}{3} - \gamma^2\beta\right)(a_{tt} + 2a_{tx}\, a + a_{xx}\, a^2)\,\Delta_n^3 \\ & + \frac{1}{6}\,(a_t\, a_x + a_x^2\, a)\,\Delta_n^3 \;+\; \text{higher order terms},\end{aligned}$$

where everything is now evaluated at (t_n, y_n). The first two terms here drop out if we choose the weighting parameters α, β and γ so that

$$\alpha + \beta = 1, \qquad \gamma\beta = \frac{1}{2}. \tag{2.7}$$

In general it will not be possible to eliminate both of the Δ^3 terms by a judicious choice of parameters β and γ because the second of these terms need not vanish identically. The parameter constraints (2.7) assure that a difference method with Ψ given by (2.6) will have local discretization error of order 3 and hence global discretization error of order 2. Since one of the parameters in (2.7) can be chosen arbitrarily, this gives an infinite number of 2nd order difference schemes. Note that the first constraint in (2.7) assures that all of these methods are consistent, as defined in Section 1. The choice $\alpha = \beta = 1/2$, $\gamma = 1$ gives the Heun method (1.12), which is also called *the* 2nd order Runge-Kutta method.

We can use an analogous derivation for the 4th order Runge-Kutta method, starting with a weighted average over four points to approximate the right hand side of the differential equation. Now the comparison is made with the 5th order truncated Taylor method. The classical 4th *order Runge-Kutta method* is an explicit method given by

$$y_{n+1} = y_n + \frac{1}{6}\left\{k_n^{(1)} + 2\,k_n^{(2)} + 2\,k_n^{(3)} + k_n^{(4)}\right\}\Delta_n \tag{2.8}$$

where

$$
\begin{aligned}
k_n^{(1)} &= a(t_n, y_n), \\
k_n^{(2)} &= a\left(t_n + \frac{1}{2}\Delta_n, y_n + \frac{1}{2}k_n^{(1)}\Delta_n\right), \\
k_n^{(3)} &= a\left(t_n + \frac{1}{2}\Delta_n, y_n + \frac{1}{2}k_n^{(2)}\Delta_n\right), \\
k_n^{(4)} &= a\left(t_{n+1}, y_n + k_n^{(3)}\Delta_n\right).
\end{aligned}
$$

When $a = a(t)$, a function of t only, the increment $y_{n+1} - y_n$ in (2.8) is just a Simpson rule approximation of the definite integral

$$\int_{t_n}^{t_{n+1}} a(t)\,dt.$$

PC-Exercise 8.2.2 *Repeat PC-Exercise 8.2.1 using the 4th order Runge-Kutta method (2.8) with equal length time steps* $\Delta = 2^{-2}, \ldots, 2^{-7}$

Even higher order Runge-Kutta schemes have been derived. It turns out that the number of evaluations of the function a needed for a pth order Runge-Kutta method is p for $2 \le p \le 4$, $p+1$ for $5 \le p \le 7$ and $p+2$ for $p \ge 8$. The 4th order Runge-Kutta methods are the most commonly used, representing a good compromise between accuracy and computational effort.

Often multi-step methods do not require as many evaluations of the function a per time step as one-step methods of the same order. An example is the 3-step Adams-Bashford method (1.14), which, essentially, requires the function a to be evaluated only at a single point, namely (t_n, y_n), for each iteration once the recursion procedure has been got going. This contrasts with the two evaluations per iteration needed by the Heun method (1.12), which is also a second order method. These considerations were of some importance before digital computers came into widespread usage and the calculations had to be done manually. They must still be borne in mind today, particularly for lengthy calculations, both for efficiency and to reduce roundoff error. Most explicit multi-step methods express y_{n+1} as a linear combination of the values y_i and $a(t_i, y_i)$ at the previous k discretization times, where k is fixed and denotes the number of steps of the method. In implicit methods the term $a(t_{n+1}, y_{n+1})$ also appears. For time steps of equal length Δ we write such multi-step methods in the general form

$$y_{n+1} = \sum_{j=1}^{k} \alpha_j\, y_{n+1-j} + \sum_{j=0}^{k} \beta_j\, a\left(t_{n+1-j}, y_{n+1-j}\right)\Delta, \tag{2.9}$$

where the α_j and β_j are given constants, with $\beta_0 = 0$ for an explicit scheme and $\beta_0 \neq 0$ for an implicit scheme. Most of these methods are derived by replacing the right hand side of the differential equation over $t_n \le t \le t_{n+1}$ by a polynomial passing through the points $(t_j, a(t_j, y_j))$ under consideration. Their local discretization error can be determined by comparison with truncated Taylor methods. The global discretization error also depends on the order of the starting routine, which should be, preferably, at least the same as that of the multi-step method itself.

Examples of multi-step methods are the *midpoint method*

$$y_{n+1} = y_{n-1} + 2\,a(t_n, y_n)\,\Delta, \tag{2.10}$$

the *Milne method*

$$y_{n+1} = y_{n-3} + \frac{4}{3}\left\{2a(t_n, y_n) - a(t_{n-1}, y_{n-1}) + 2a(t_{n-2}, y_{n-2})\right\}\Delta \tag{2.11}$$

and the *Adams-Moulton method*

$$y_{n+1} = y_n + \frac{1}{12}\left\{5a(t_{n+1}, y_{n+1}) + 8a(t_n, y_n) - a(t_{n-1}, y_{n-1})\right\}\Delta. \tag{2.12}$$

The first two of these are explicit methods and the third is implicit. They have local discretization errors of orders 3, 5 and 4, respectively.

Note that an arbitrary choice of coefficients α_j and β_j in (2.9) may result in an inconsistent method. Also, even if the coefficients are determined by an interpolating polynomial, a multi-step method may have some undesirable properties, such as being susceptible to numerical instabilities.

PC-Exercise 8.2.3 *Calculate the discretization errors in using the Euler method (1.2) and the midpoint method (2.10) started with the Euler method to approximate the solution* $x(t) = \frac{2}{3}e^{-3t} + \frac{1}{3}$ *of the initial value problem*

$$\frac{dx}{dt} = -3x + 1, \qquad x(0) = 1$$

over the interval $0 \le t \le 1$. *Use time steps of equal length* $\Delta = 0.1$ *and plot on* x *versus* t *axes.*

The trapezoidal method (1.11) and the Adams-Moulton method (2.12) are examples of implicit difference methods. These are often more stable than their explicit counterparts. A root finding method such as the Newton method could be used at each step to calculate an approximation of the unknown value y_{n+1}. Another approach is to use an explicit method to predict an approximation y_{n+1}^0 to y_{n+1}, which is then inserted into the right hand side of the implicit method to calculate another approximation y_{n+1}^1. This correction procedure can then be repeated $l \ge 0$ times to produce a final approximation y_{n+1}^{l+1}. The resulting method is called a *predictor-corrector method*. A very simple example with $l = 0$ is the Heun or modified trapezoidal method (1.12), which uses the Euler method as its predictor. Besides providing the desired approximate value at each time step, a predictor- corrector method also gives an easy indication of the local dicretization error. This could be useful in choosing an appropriate, possibly varying, step size.

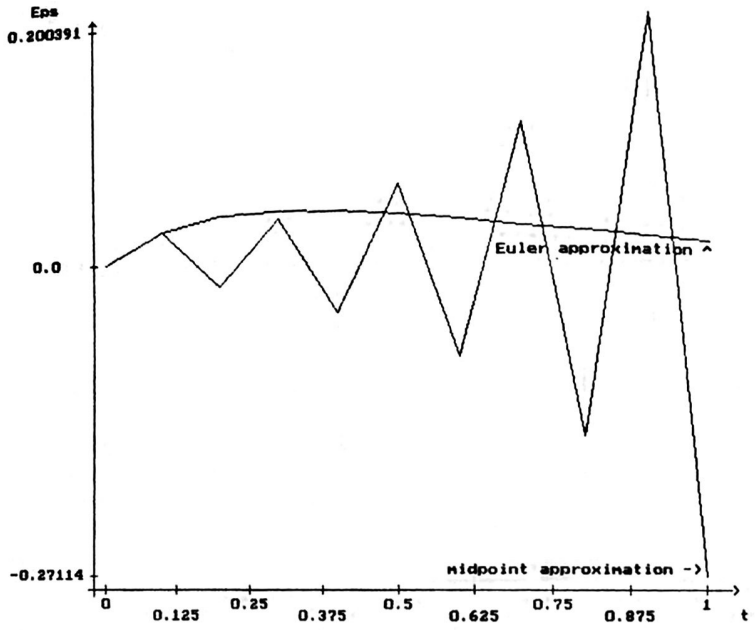

Figure 8.2.2 Results of PC-Exercise 8.2.3.

8.3 Consistency, Convergence and Stability

We usually not know the exact solution of an initial value problem that we are trying to approximate by a finite difference method. Then, to assure that an approximation will be reasonably accurate, we have to be able to keep the unknown discretization and roundoff errors under control and sufficiently small. We can use certain a priori information about the difference method, that is information obtainable without explicit knowledge of the exact solution, to tell us whether this is possible. In particular, we can check if the method is *consistent* with the differential equation, if the estimates of the global discretization error *converge* to zero with the maximum time step, and if the method is *stable*, that is if propagated errors remain bounded.

We shall assume for any differential equation

(3.1) $$\frac{dx}{dt} = a(t,x)$$

under consideration that the function $a = a(t,x)$ and its partial derivatives of sufficiently high order are continuous everywhere. For most of the common one-step methods

(3.2) $$y_{n+1} = y_n + \Psi(t_n, y_n, \Delta_n)\,\Delta_n$$

the increment function $\Psi = \Psi(t,x,\Delta)$ will then be continuous in all three variables and satisfy a local Lipschitz condition in x. Such methods are generally also *consistent* with the differential equation (3.1), that is they satisfy

(3.3) $$\Psi(t,x,0) = a(t,x)$$

everywhere. Comparing (3.2) with a truncated Taylor method, we can then establish that the local discretization error (1.3) has order $p+1$ for some $p \geq 1$; when (3.3) is violated we can only get a local discretization error of order 1.

By *convergence* of a one-step method (3.2) we mean that the global discretization error (1.4) converges to zero with the maximum time step $\Delta = \max_n \Delta_n$, that is

(3.4) $$\lim_{\Delta \downarrow 0} |e_{n+1}| = \lim_{\Delta \downarrow 0} |x(t_{n+1}; t_0, x_0) - y_{n+1}| = 0,$$

where $y_0 = x_0$, on any finite time interval $[t_0, T]$. Of more practical significance than convergence itself is the rate of convergence, which is provided by the order of the global discretization error. Since the local discretization error effects the global discretization error, we cannot expect to have convergence if (3.2) is not consistent. The following two theorems indicate the precise link between consistency and convergence, and between the orders of the local and global discretization errors. To simplify the proofs we shall assume that the increment function Ψ of (3.2) satisfies a global Lipschitz condition

(3.5) $$|\Psi(t',x',\Delta') - \Psi(t,x,\Delta)| \leq K\,(|t'-t| + |x'-x| + |\Delta'-\Delta|)$$

in (t,x,Δ) and a global bound of the form

(3.6) $$|\Psi(t,x,0)| \leq L$$

for all (t,x), although it is possible to weaken these assumptions.

Theorem 8.3.1 *A one-step method (3.2) with increment function Ψ satisfying conditions (3.5) and (3.6) is convergent if and only if it is consistent.*

Proof It follows from the Lipschitz condition (3.5) that the differential equation

(3.7) $$\frac{dz}{dt} = \Psi(t,z,0)$$

has a unique continuously differentiable solution $z(t) = z(t;t_0,x_0)$ with the initial value $z(t_0) = x_0$. Hence by the Mean Value Theorem there exists a θ_n with $0 < \theta_n < 1$ such that

(3.8) $$z(t_{n+1}) - z(t_n) = \Psi(t_n + \theta_n\Delta_n, z(t_n + \theta_n\Delta_n), 0)\,\Delta_n.$$

Writing $\bar{e}_n = y_n - z(t_n)$, where y_n satisfies (3.2) with $y_0 = x_0$, we have

$$\begin{aligned}\bar{e}_{n+1} &= \bar{e}_n + \{\Psi(t_n, y_n, \Delta_n) - \Psi(t_n + \theta_n\Delta_n, z(t_n + \theta_n\Delta_n), 0)\}\ \Delta_n \\ &= \bar{e}_n + \{\Psi(t_n, y_n, \Delta_n) - \Psi(t_n, z(t_n), 0)\}\ \Delta_n \\ &\quad + \{\Psi(t_n, z(t_n), 0) - \Psi(t_n + \theta_n\Delta_n, z(t_n + \theta_n\Delta_n), 0)\}\ \Delta_n,\end{aligned}$$

from which we obtain

$$\begin{aligned}|\bar{e}_{n+1}| &\le |\bar{e}_n| + K\ (|\bar{e}_n| + \Delta_n)\ \Delta_n \\ &\quad + K\ (\theta_n\Delta_n + |z(t_n) - z(t_n + \theta_n\Delta_n)|)\ \Delta_n\end{aligned} \tag{3.9}$$

by means of the Lipschitz condition (3.5). Using the Mean Value Theorem again and the bound (3.6) we have

$$|z(t_n) - z(t_n + \theta_n\Delta_n)| = \left|\Psi(t_n + \bar{\theta}_n\theta_n\Delta_n, z(t_n + \bar{\theta}_n\theta_n\Delta_n), 0)\right|\ \theta_n\Delta_n \le L\,\theta_n\Delta_n$$

for some $0 < \bar{\theta}_n < 1$. Inserting this into (3.9), we then get

$$|\bar{e}_{n+1}| \le (1 + K\Delta)\,|\bar{e}_n| + K\,(L + 2)\ \Delta^2, \tag{3.10}$$

where $\Delta = \max_n \Delta_n$. We can use induction to show that

$$|\bar{e}_n| \le (L + 2)\left(e^{K(T - t_0)} - 1\right)\ \Delta$$

on an interval $[t_0, T]$, from which we conclude that the approximations y_n generated by the one-step method (3.2) converge to the solution $z(t) = z(t; t_0, x_0)$ of (3.7) on $[t_0, T]$.

Assuming consistency, the differential equations (3.1) and (3.7) are the same, so by the uniqueness of solutions of initial value problems $z(t) \equiv x(t)$ for $t_0 \le t \le T$. From the above considerations we have thus established convergence of the one-step method (3.2).

Assuming convergence, we have $z(t) \equiv x(t)$ for $t_0 \le t \le T$. If there were a point (t_0, x_0) where $a(t_0, x_0) \ne \Psi((t_0, x_0, 0)$, we would have

$$\frac{dx}{dt}(t_0) = a(t_0, x_0) \ne \Psi((t_0, x_0, 0) = \frac{dz}{dt}(t_0),$$

which contradicts the fact that $z(t) \equiv x(t)$. Hence the consistency condition (3.3) must hold.

This completes the proof of Theorem 8.3.1. □

Theorem 8.3.2 *A one-step method (3.2) with increment function Ψ satisfying the global Lipschitz condition (3.5) and with local discretization error of order $p + 1$ has global discretization error of order p.*

Proof Let $x(t) = x(t; t_0, x_0)$ be the solution of the initial value problem (3.1) and let y_n be generated by (3.2) with $y_0 = x_0$. Then the global discretization error (1.4) satisfies

$$\begin{aligned}e_{n+1} &= y_{n+1} - x(t_{n+1}) \\ &= e_n + \Psi(t_n, y_n, \Delta_n)\,\Delta_n + x(t_n) - x(t_{n+1}) \\ &= e_n + \{\Psi(t_n, y_n, \Delta_n) - \Psi(t_n, x(t_n), \Delta_n)\}\ \Delta_n \\ &\quad + \{\Psi(t_n, x(t_n), \Delta_n)\,\Delta_n + x(t_n) - x(t_{n+1})\},\end{aligned}$$

where the very last term is the local discretization error. From the global Lipschitz condition (3.5) and the assumption that the local discretization error is of order $p + 1$ we obtain

$$\begin{aligned}|e_{n+1}| &\le |e_n| + K\,|e_n|\ \Delta_n + D\,\Delta_n^{p+1} \\ &\le (1 + K\Delta)\,|e_n| + D\,\Delta^{p+1},\end{aligned}$$

where $\Delta = \max_n \Delta_n$ and D is some positive constant, from which it follows that

$$|\bar{e}_n| \le \frac{D}{K}\left(e^{K(T - t_0)} - 1\right)\ \Delta^p$$

on the interval $[t_0, T]$. The global discretization error is thus of order p. □

Exercise 8.3.3 *Show that the increment function $\Psi(t, x, \Delta)$ of the Heun method (1.12) satisfies a global Lipschitz condition (3.5) in (t, x, Δ) when $a(t, x)$ satisfies a global Lipschitz condition in (t, x). Also, show that the Heun method is consistent and hence convergent with global discretization error of order 2.*

We may still encounter difficulties when trying to implement a difference method which is known to be convergent. For example, the differential equation

$$\frac{dx}{dt} = -16\,x \tag{3.11}$$

has exact solutions $x(t) = x_0 e^{-16t}$, which all converge very rapidly to zero. For this differential equation the Euler method with constant time step Δ,

$$y_{n+1} = (1 - 16\Delta)\,y_n,$$

has exact iterates $y_n = (1 - 16\Delta)^n\,y_0$. If we choose $\Delta > 2^{-3}$ these iterates oscillate with increasing amplitude instead of converging to zero like the exact solutions of (3.11). This is a simple example of a *numerical instability*, which, in this particular case, we can overcome simply by taking the time step $\Delta < 2^{-3}$. For some other methods, such as the midpoint method (2.10) investigated in PC-Exercise 8.2.3, the numerical instabilities persist no matter how small we take Δ. The structure of these methods can make them intrinsically unstable, causing small errors such as roundoff errors to grow rapidly and ultimately rendering the calculations useless.

The idea of numerical stability of a one-step method is that errors will remain bounded with respect to an initial error for any differential equation (3.1) with right hand side $a(t, x)$ satisfying a Lipschitz condition. To be specific,

we say that a one-step method (3.2) is *numerically stable* if for each interval $[t_0, T]$ and differential equation (3.1) with $a(t,x)$ satisfying a Lipschitz condition there exist positive constants Δ_0 and M such that

$$|y_n - \tilde{y}_n| \leq M\ |y_0 - \tilde{y}_0| \tag{3.12}$$

for $n = 0, 1, \ldots, n_T$ and any two solutions y_n, $\tilde{y}_n$ of (3.2) corresponding to any time discretizations with $\max_n \Delta_n < \Delta_0$. The constants Δ_0 and M here may also depend on the particular time interval $t_0 \leq t \leq T$ in addition to the differential equation under consideration. (3.12) is analogous to the continuity in initial conditions, uniformly on finite time intervals, of the solutions of the differential equation (3.1). The following result thus comes as no surprise.

Theorem 8.3.4 *A one-step method (3.2) is numerically stable if the increment function Ψ satisfies a global Lipschitz condition (3.5).*

The commonly used one-step methods are numerically stable. However, the constant M in (3.12) may be quite large. For example, if we replace the minus sign by a plus sign in the differential equation (3.11), we obtain

$$y_n - \tilde{y}_n = (1 + 16\Delta)^n\ (y_0 - \tilde{y}_0)$$

for the Euler method. The numerical stability condition (3.12) requires a bound like $e^{16(T-t_0)}$ for M, in contrast with $M \leq 1$, provided $\Delta_0 < 2^{-3}$, for the original differential equation. The difference is due to the fact that the solutions of the modified differential equation are diverging exponentially fast, whereas those of the original are converging exponentially fast. In both cases the Euler method keeps the error under control, but in the former case the initial error must be considerably smaller if it is to remain small.

Exercise 8.3.5 *Prove Theorem 8.3.4.*

To ensure that the errors in the Euler method for (3.11) do not grow, that is the bound $M \leq 1$ in (3.12), we need to take step sizes less than 2^{-3}. This may seem inordinately small given that the differential equation itself is very stable. The situation does not improve if we use the higher order Heun method (1.12). However, the implicit trapezoidal method (1.11) offers a substantial improvement. In this case it is

$$y_{n+1} = y_n + \frac{1}{2}\{-16\, y_n - 16\, y_{n+1}\}\ \Delta, \tag{3.13}$$

which we can solve explicitly to get

$$y_{n+1} = \left(\frac{1 - 8\Delta}{1 + 8\Delta}\right)\ y_n.$$

Here

$$\left|\frac{1 - 8\Delta}{1 + 8\Delta}\right| < 1 \tag{3.14}$$

for any $\Delta > 0$. For a nonlinear differential equation we usually cannot solve an implicit method algebraically for y_{n+1} as in (3.13). Nevertheless this example highlights a significant advantage of implicit methods, which sometimes makes the additional work needed to solve numerically for y_{n+1} worthwhile.

In the preceding discussion we tried to ensure that the error would not grow over an infinite time horizon. This leads to the idea of asymptotic numerical stability. We shall say that a one-step method (3.1) is *asymptotically numerically stable* for a given differential equation if there exist positive constants Δ_a and M such that

$$\lim_{n\to\infty} |y_n - \bar{y}_n| \leq M\, |y_0 - \bar{y}_0| \tag{3.15}$$

for any two solutions y and $\bar{y}$ of (3.2) corresponding to any time discretization with $\max_n \Delta_n < \Delta_a$.

It is easy to see that the Euler method is asymptotically numerically stable for the differential equation (3.11) with $\Delta_a \leq 2^{-3}$, whereas the implicit trapezoidal method (3.13) is asymptotically numerically stable for this differential equation without any restriction on Δ_a. On the other hand the Euler method is not asymptotically numerically stable for the differential equation $\dot{x} = 16x$ for any $\Delta_a > 0$.

Knowing just that a one-step method is numerically stable does not tell us how to pick an appropriate step size Δ. In fact, the answer will depend very much on the particular differential equation under consideration. To obtain an indication of suitable values of Δ we consider a class of test equations. These are the complex-valued linear differential equations

$$\frac{dx}{dt} = \lambda\, x, \tag{3.16}$$

with $\lambda = \lambda_r + \imath\lambda_i$, which have oscillating solutions when $\lambda_i \neq 0$. We can obviously write (3.16) equivalently as a 2-dimensional differential equation

$$\frac{d}{dt}\begin{pmatrix} x^1 \\ x^2 \end{pmatrix} = \begin{bmatrix} \lambda_r & -\lambda_i \\ \lambda_i & \lambda_r \end{bmatrix} \begin{pmatrix} x^1 \\ x^2 \end{pmatrix}$$

where $x = x^1 + \imath x^2$. The suitable values of the step size $\Delta > 0$ are expressed in terms of the *region of absolute stability* for the method, consisting of the complex numbers $\lambda\Delta$ for which an error in y_0 at t_0 will not grow in subsequent iterations of the method applied to the differential equation (3.16). Essentially, these are the values of λ and Δ producing a bound $M \leq 1$ in (3.12). For the Euler method we thus require

$$|1 + \lambda\,\Delta| \leq 1,$$

so its region of absolute stability is the unit disc in the complex plane centered on $z = -1 + 0\imath$.

Exercise 8.3.6 *Determine and sketch the region of absolute stability for the trapezoidal method (1.11).*

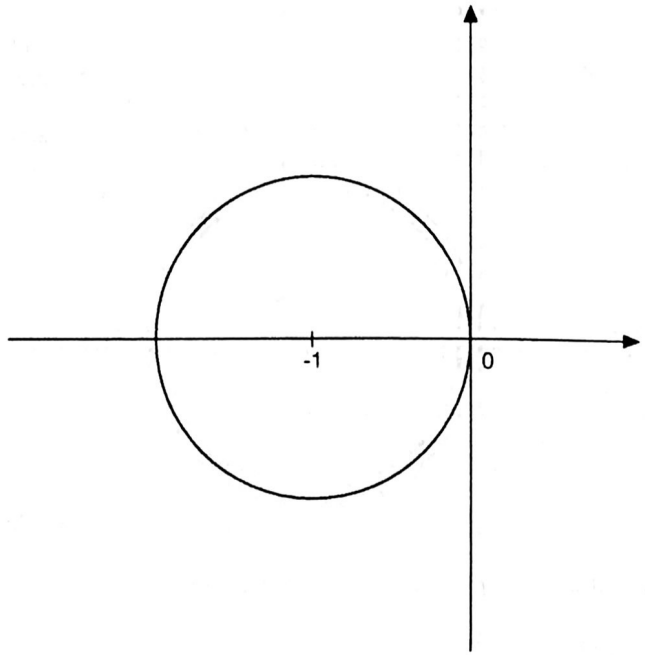

Figure 8.3.1 Stability region for the Euler method.

We shall now consider a 2-dimensional linear differential equation

$$\frac{d}{dt}\begin{pmatrix} x^1 \\ x^2 \end{pmatrix} = \begin{bmatrix} -\alpha_1 & 0 \\ 0 & -\alpha_2 \end{bmatrix} \begin{pmatrix} x^1 \\ x^2 \end{pmatrix} \tag{3.17}$$

with initial value $(x_0^1, x_0^2) = (1, 1)$, where the two eigenvalues of the coefficient matrix are negative and very different, that is with

$$0 \leq \alpha_2 \ll \alpha_1.$$

The components of (3.17) are uncoupled, so they can be solved separately to give

$$x^1(t) = e^{-\alpha_1 t}, \qquad x^2(t) = e^{-\alpha_2 t}. \tag{3.18}$$

Since α_1 is much larger than α_2 the first component shows a very fast exponential decay in comparison with the second, that is the relaxation time of the first component is very much smaller than that of the second. In other words the two components have widely differing time scales. In the literature such a system of equations is often called a *stiff system*. In the general d-dimensional case we shall say that a linear system is *stiff* if the real parts of the eigenvalues $\lambda_1, \ldots, \lambda_d$ of the coefficient matrix satisfy

$$\max_{k=1,\ldots,d} \mathrm{Re}(\lambda_k) \gg \min_{k=1,\ldots,d} \mathrm{Re}(\lambda_k).$$

Now, if we apply the Euler method (1.2) to (3.17), for the first component to remain within the region of absolute stability we need a step size

$$\Delta \leq \frac{2}{\alpha_1}.$$

We saw in Figure 8.1.3 of Section 1 that there is a lower bound on the step size for which the influence of the roundoff error to remain acceptable. But the upper bound $2/\alpha_1$ might already be too small to allow for the control over roundoff errors in the second component. Thus, the Euler scheme may not be applicable for a stiff system.

A much more stable result is shown when we apply the *implicit Euler scheme*

$$y_{n+1} = y_n + a(t_{n+1}, y_{n+1})\,\Delta \tag{3.19}$$

to the test equation (3.16). Using similar notation as above we obtain

$$y_n = (1 - \lambda\Delta)^{-n} y_0$$

and, hence, for λ with $\mathrm{Re}(\lambda) = \lambda_r < 0$ and all $\Delta > 0$ we have

$$|y_n - \bar{y}_n| \leq |y_0 - \bar{y}_0|$$

for all $n = 0, 1, \ldots$ and any two solutions y_n, $\bar{y}_n$ of (3.19). Thus, the implicit Euler method (3.19) applied to the stiff system (3.17) would still behave stably in its first component when $\Delta > 2/\alpha_1$.

We shall say that a numerical method is *A-stable* if its region of absolute stability contains all of the left half of the complex plane, that is all $\lambda\Delta$ with $\mathrm{Re}(\lambda) < 0$ and $\Delta > 0$. Hence the implicit Euler method (3.19) is A-stable, whereas the Euler method (1.2) is not.

Exercise 8.3.7 *Check whether or not the trapezoidal method (1.11) is A-stable.*

We shall conclude this section with a few remarks on consistency, convergence and stability of multi-step methods, for which matters are somewhat more complicated than for one-step methods. The definition of convergence for a multi-step method assumes that the starting values are exact, although in practice these will be calculated approximately by means of a one-step method. The idea of stability is similar to that for one-step methods, but now some new phenomena can occur. Suppose we have a k-step method

$$\sum_{j=0}^{k} (\alpha_{n+1-j} y_{n+1-j} + \beta_{n+1-j}\, a(t_{n+1-j}, y_{n+1-j})\Delta) = 0. \tag{3.20}$$

For the linear differential equation (3.15) this is a linear recursion

$$\sum_{j=0}^{k} (\alpha_{n+1-j} + \beta_{n+1-j}\,\lambda\,\Delta)\; y_{n+1-j} = 0, \tag{3.21}$$

which is also satisfied by iterated errors $e_n = y_n - \tilde{y_n}$. To solve (3.21) we try solutions of the form $y_n = \xi^n$ and find that ξ must be a root, possibly complex valued, of the polynomial equation

$$\sum_{i=0}^{k} (\alpha_i + \beta_i\,\lambda\,\Delta)\;\xi^i = 0. \tag{3.22}$$

Any errors introduced will thus die out if all of the roots lie within the unit circle in the complex plane, that is have modulus $|\xi| < 1$. One of the roots will be approximately equal to $e^{\lambda\Delta}$ and the corresponding iterates $y_n = x_0\,\xi^n$ correspond to the differential equation solution values $x(t_n) = x_0 e^{\lambda t_n}$. The problem now is that (3.22) may have other roots lying outside of the unit circle and these may lead to iterates of the multi-step method increasing in magnitude when the differential equation has no such solutions. We say that the multi-step method (3.20) is *stable* when all of the roots of (3.22) lie within the unit complex circle for Δ sufficiently small. A necessary and sufficient condition for this is that the roots of the polynomial

$$\sum_{i=0}^{k} \alpha_i\,\xi^i = 0$$

lie within the unit complex circle, or possibly also on the unit circle if a root is simple. The term *strong stability* is used if all roots except $\xi = 1$ lie inside the unit circle and *weak stability* if other roots also lie on the circle. For example, the Adams-Bashford method (1.14) is strongly stable, whereas the midpoint (2.10) and the Milne (2.11) methods are weakly stable. The presence of the extra roots on the unit circle means that (3.22) may have roots lying outside the unit circle no matter how small Δ is taken, which can lead to numerical instabilities. For example, the midpoint method (2.10) has roots

$$\xi = \frac{1}{2}\lambda\Delta \pm \sqrt{1 + \left(\frac{1}{2}\lambda\Delta\right)^2},$$

one of which has modulus greater than 1. Finally, as a partial analogue of Theorem 8.3.1, we remark that it can be shown that a multi-step method is convergent if it is consistent and stable.

Exercise 8.3.8 *Determine the polynomials (3.22) for the Adams-Bashford method (1.14) and the Adams-Moulton method (2.12).*

8.4 Roundoff Error

Roundoff errors occur because, in practice, arithmetic operations can only be carried out to a finite number of significant decimal places. In principle we could determine the roundoff error of each calculation, and hence the accumulated roundoff error, exactly, though this is infeasible in all but the simplest situations and we have to use estimates instead. Assuming constant roundoff error r at each step, in Section 1 we derived a theoretical upper bound proportional to r/Δ, where Δ is the maximum time step, for the Euler method. This assumption is certainly not true, but the implication from it that there is a minimum time step Δ_{min} below which the total error will begin to increase is consistent with what actually happens in numerical calculations, as we saw in PC-Exercise 8.1.2.

More realistic estimates of the accumulated roundoff error can be determined from a statistical analysis, assuming that the local roundoff errors are independent, identically distributed random variables. It is commonly assumed that they are uniformly distributed over the interval

$$\left[-5 \times 10^{-(s+1)},\; 5 \times 10^{-(s+1)}\right], \tag{4.1}$$

where s is the number of significant decimal places used. To check the appropriateness of this distribution we could repeat the calculations using double precision arithmetic and use the difference of single and double precision results to represent the roundoff error. If double precision arithmetic is not available we can simulate the same effect by using arithmetic to s decimal places, say $s = 4$, instead of single precision and the computer's prescribed precision instead of double precision.

PC-Exercise 8.4.1 *Calculate 300 iterates of*

$$y_{n+1} = \frac{\pi}{3}\,y_n$$

with initial value $y_0 = 0.1$ using the prescribed arithmetic of the PC, at each step rounding the value of y_{n+1} obtained to four significant figures. Plot the relative frequencies of the roundoff errors in a histogram on the interval

$$[-5 \times 10^{-5}, 5 \times 10^{-5}]$$

using 40 equal subintervals.

If the local roundoff errors r_n take values in the interval (4.1), then after N calculations the accumulated roundoff error

$$R_N = \sum_{n=1}^{N} r_n$$

would lie in the interval

$$\left[-5N \times 10^{-(s+1)}, 5N \times 10^{-(s+1)}\right].$$

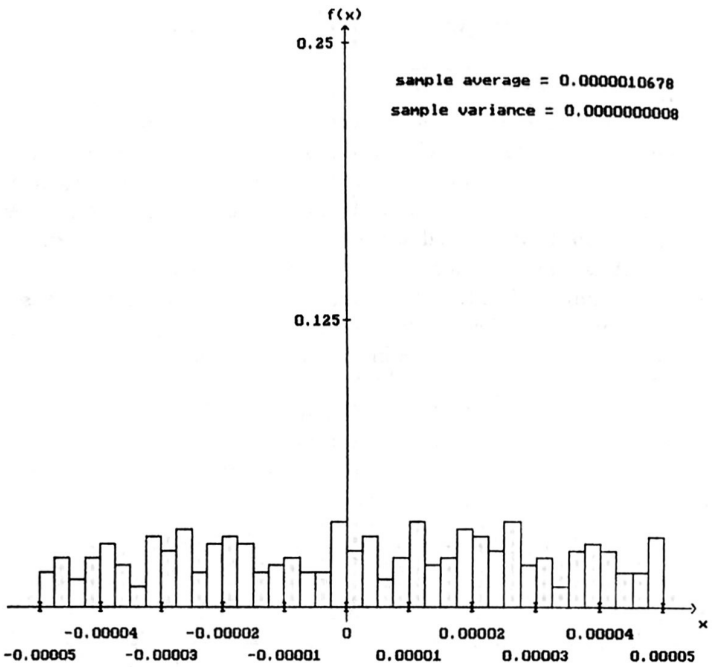

Figure 8.4.1 Histogram of the roundoff error in PC-Exercise 8.4.1.

After $N = 10^s$ calculations this is $[-0.5, +0.5]$, so all decimal places of accuracy may be lost. However, this worst case scenario is highly unlikely to occur. If the r_n are uniformly distributed over the interval (4.1) they have mean and variance

$$\mu = E(r_n) = 0, \qquad \sigma^2 = \mathrm{Var}(r_n) = \frac{1}{12}\,10^{-2s}.$$

Thus, if they are also independent, the accumulated roundoff error has mean and variance

$$E(R_N) = 0, \qquad \mathrm{Var}(R_N) = N\,\sigma^2.$$

By the Central Limit Theorem (see (1.5.9)) the normalized random variables $Z_N = R_N/\sigma\sqrt{N}$ are approximately standard Gaussian for large N. From this, as in Section 9 of Chapter 1, we can conclude that the values of R_N lie with probability 0.95 in the interval

$$\left[-1.96 \times 10^{-s}\sqrt{N/12},\ 1.96 \times 10^{-s}\sqrt{N/12}\right]$$

when N is large. The ratio of $1.96 \times 10^{-s}\sqrt{N/12}$ to $5N \times 10^{-(s+1)}$ is approximately $1/\sqrt{N}$, so for large N the accuracy is in fact considerably better than predicted by the worst case scenario above. Of course, it may be much worse in some instances, but these occur with small probabilities.

PC-Exercise 8.4.2 *Use the Euler method with equal time steps $\Delta = 2^{-2}$ for the differential equation*

$$\frac{dx}{dt} = x$$

over the interval $0 \le t \le 1$ with $N = 10^3$ different initial values $x(0)$ between 0.4 and 0.6. Use both four significant figure arithmetic and the prescribed arithmetic of the PC and determine the final accumulative roundoff error $R_{1/\Delta}$ in each case, plotting them in a histogram on the interval $[-5 \times 10^{-4}, 5 \times 10^{-4}]$ with 40 equal subintervals. In addition, calculate the sample mean and sample variance of the $R_{1/\Delta}$ values.

As a final comment we remark that the roundoff error may be considered as being independent of the discretization error.

PC-Exercise 8.4.3 *Repeat PC-Exercise 8.4.2 with $N = 200$ and with time steps $\Delta = 2^{-2}$, 2^{-3}, 2^{-4} and 2^{-5}, determining $R_{1/\Delta}$ in each case. Plot the 90% confidence intervals for the mean value of the error against Δ.*