CHAPTER 13

Numerical Solution of Differential Equations

We have considered numerical solution procedures for two kinds of equations: In chapter 10 the unknown was a real number; in chapter 6 the unknown was a sequence of numbers. In a differential equation the unknown is a function, and the differential equation relates the function to its derivative(s).

In this chapter we start by considering how the simplest differential equations, the first order ones which only involve the unknown function and its first derivative, can be solved numerically by the simplest method, namely Euler's method. We analyse the error in Euler's method, and then introduce some more advanced methods with better accuracy. After this we show that the methods for handling one equation in one unknown generalise nicely to systems of several equations in several unknowns. What about equations that involve higher order derivatives? It turns out that even systems of higher order equations can be rewritten as a system of first order equations. At the end we discuss briefly the important concept of stability.

13.1 What are differential equations?

Differential equations is an essential tool in a wide range of applications. The reason for this is that many phenomena can be modelled by a relationship between a function and its derivatives. Let us consider a simple example.

13.1.1 An example from physics

Consider an object moving through space. At time $t = 0$ it is located at a point P and after a time t its distance to P corresponds to a number $f(t)$. In other words, the distance can be described by a function of time. The divided difference

$$\frac{f(t+\Delta t)-f(t)}{\Delta t} \tag{13.1}$$

then measures the average speed during the time interval from t to $t+\Delta t$. If we take the limit in (13.1) as Δt approaches zero, we obtain the speed $v(t)$ at time t,

$$v(t) = \lim_{\Delta t\to 0} \frac{f(t+\Delta t)-f(t)}{\Delta t}. \tag{13.2}$$

Similarly, the divided difference of the speed is given by $\bigl(v(t+\Delta t)-v(t)\bigr)/\Delta t$. This is the average acceleration from time t to time $t+\Delta t$, and if we take the limit as Δt tends to zero we get the acceleration $a(t)$ at time t,

$$a(t) = \lim_{\Delta t\to 0} \frac{v(t+\Delta t)-v(t)}{\Delta t}. \tag{13.3}$$

If we compare the above definitions of speed and acceleration with the definition of the derivative we notice straightaway that

$$v(t) = f'(t), \qquad a(t) = v'(t) = f''(t). \tag{13.4}$$

Newton's second law states that if an object is influenced by a force, its acceleration is proportional to the force. More precisely, if the total force is F, Newton's second law can be written

$$F = ma \tag{13.5}$$

where the proportionality factor m is the mass of the object.

As a simple example of how Newton's law is applied, we can consider an object with mass m falling freely towards the earth. It is then influenced by two opposite forces, gravity and friction. The gravitational force is $F_g = mg$, where g is acceleration due to gravitation alone. Friction is more complicated, but in many situations it is reasonable to say that it is proportional to the square of the speed of the object, or $F_f = cv^2$ where c is a suitable proportionality factor. The two forces pull in opposite directions so the total force acting on the object is $F = F_g - F_f$. From Newton's law $F = ma$ we then obtain the equation

$$mg - cv^2 = ma.$$

Gravity g is constant, but both v and a depend on time and are therefore functions of t. In addition we know from (13.4) that $a(t) = v'(t)$ so we have the equation

$$mg - cv(t)^2 = mv'(t)$$

which would usually be shortened and rearranged as

$$mv' = mg - cv^2. \tag{13.6}$$

The unknown here is the function $v(t)$, the speed, but the equation also involves the derivative (the acceleration) $v'(t)$, so this is a differential equation. This equation is just a mathematical formulation of Newton's second law, and the hope is that we can solve the equation and determine the speed $v(t)$.

13.1.2 General use of differential equations

The simple example above illustrates how differential equations are typically used in a variety of contexts:

Procedure 13.1 (Modelling with differential equations)**.**

1. *A quantity of interest is modelled by a function x.*
2. *From some known principle a relation between x and its derivatives is derived, in other words, a differential equation.*
3. *The differential equation is solved by a mathematical or numerical method.*
4. *The solution of the equation is interpreted in the context of the original problem.*

There are several reasons for the success of this procedure. The most basic reason is that many naturally occurring quantities can be represented as mathematical functions. This includes physical quantities like position, speed and temperature, which may vary in both space and time. It also includes quantities like 'money in the bank' and even vaguer, but quantifiable concepts like for instance customer satisfaction, both of which will typically vary with time.

Another reason for the popularity of modelling with differential equations is that such equations can usually be solved quite effectively. For some equations it is possible to find an explicit expression for the unknown function, but this is rare. For a large number of equations though, it is possible to compute good approximations to the solution via numerical algorithms, and this is the main topic in this chapter.

13.1.3 Different types of differential equations

Before we start discussing numerical methods for solving differential equations, it will be helpful to classify different types of differential equations. The simplest equations only involve the unknown function x and its first derivative x', as in (13.6); this is called a *first order differential equation.* If the equation involves higher derivatives up ot order p it is called a *pth order differential equation.* An important subclass are given by *linear differential equations.* A linear differential equation of order p is an equation on the form

$$x^{(p)}(t) = f(t) + g_0(t)x(t) + g_1(t)x'(t) + g_2(t)x''(t) + \cdots + g_{p-1}(t)x^{(p-1)}(t).$$

For all the equations we study here, the unknown function depends on only one variable which we usually label as t. Such equations are referred to as *ordinary differential equations.* This is in contrast to equations where the unknown function depends on two or more variables, like the three coordinates of a point in space, these are referred to as *partial differential equations.*

13.2 First order differential equations

A first order differential equation is an equation on the form

$$x' = f(t, x).$$

Here $x = x(t)$ is the unknown function, and t is the free variable. The function f tells us how x' depends on both t and x and is therefore a function of two variables. Some examples may be helpful.

Example 13.2. Some examples of first order differential equations are

$$x' = 3, \qquad x' = 2t, \qquad x' = x, \qquad x' = t^3 + \sqrt{x}, \qquad x' = \sin(tx).$$

The first three equations are very simple. In fact the first two can be solved by integration and have the solutions $x(t) = 3t + C$ and $x(t) = t^2 + C$ where C is an arbitrary constant in both cases. The third equation cannot be solved by integration, but it is easy to check that the function $x(t) = Ce^t$ is a solution for any value of the constant C. It is worth noticing that all the first three equations are linear.

For the first three equations there are simple procedures that lead to the solutions. On the other hand, the last two equations do not have solutions given by simple formulas. In spite of this, we shall see that there are simple numerical methods that allow us to compute good approximations to the solutions. ∎

The situation described in example 13.2 is similar to what we had for nonlinear equations and integrals: There are analytic solution procedures that work in some special situations, but in general the solutions can only be determined approximately by numerical methods.

In this chapter our main concern will be to derive numerical methods for solving differential equations on the form $x' = f(t,x)$ where f is a given function of two variables. The description may seem a bit vague since f is not known explicitly, but the advantage is that once the method has been deduced we may plug in almost any f.

When we solve differential equations numerically we need a bit more information than just the differential equation itself. If we look back on example 13.2, we notice that the solution in the first three cases involved a general constant C, just like when we determine indefinite integrals. This ambiguity is present in all differential equations, and cannot be handled very well by numerical solution methods. We therefore need to supply an extra condition that will specify the value of the constant. The standard way of doing this for first order equations is to specify one point on the solution of the equation. In other words, we demand that the solution should satisfy the equation $x(a) = x_0$ for some real numbers a and x_0.

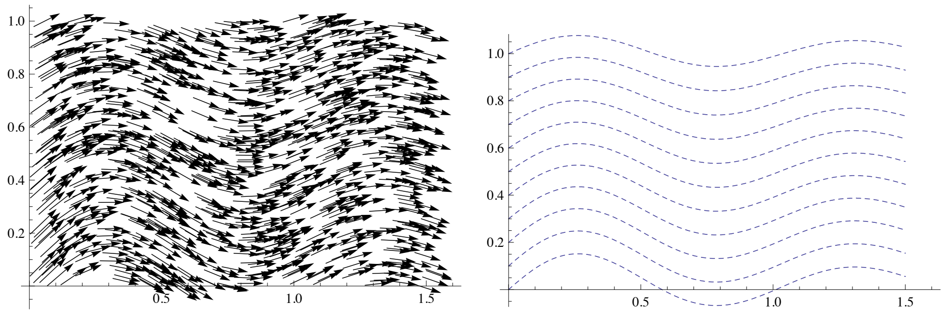
Example 13.3. Let us consider the differential equation $x' = 2x$. It is easy to check that $x(t) = Ce^{2t}$ is a solution for any value of the constant C. If we add the initial value $x(0) = 1$, we are led to the equation $1 = x(0) = Ce^0 = C$, so $C = 1$ and the solution becomes $x(t) = e^{2t}$.

If we instead impose the initial condition $x(1) = 2$, we obtain the equation $2 = x(1) = Ce^2$ which means that $C = 2e^{-2}$. In this case the solution is therefore $x(t) = 2e^{-2}\,e^t = 2e^{2(t-1)}$.

The general initial condition is $x(a) = x_0$. This leads to $x_0 = x(a) = Ce^{2a}$ or $C = x_0e^{-2a}$. The solution is therefore

$$x(t) = x_0e^{2(t-a)}. \quad \blacksquare$$

Adding an initial condition to a differential equation is not just a mathematical trick to pin down the exact solution; it usually has a concrete physical interpretation. Consider for example the differential equation (13.6) which describes the speed of an object with mass m falling towards earth. The speed at a certain time is clearly dependent on how the motion started — there is a difference between just dropping a ball and throwing it towards the ground, but note that there is nothing in equation (13.6) to reflect this difference. If we measure time such that $t = 0$ when the object starts falling, we would have $v(0) = 0$ in the situation where it is simply dropped, we would have $v(0) = v_0$ if it is thrown down-



(a) (b)

Figure 13.1. Figure (a) shows the tangents to the solutions of the differential equation $x' = \cos 6t/\left(1+t+x^2\right)$ at 1000 random points in the square $[0,1.5]\times[0,1]$. Figure (b] shows the 11 solutions corresponding to the initial values $x(0) = i/10$, for $i = 0, 1, \ldots, 10$.

wards with speed v_0, and we would have $v(0) = -v_0$ if it was thrown upwards with speed v_0. Let us sum this up in an observation.

Observation 13.4 (First order differential equation)**.** *A first order differential equation is an equation on the form $x' = f(t,x)$, where $f(t,x)$ is a function of two variables. In general, this kind of equation has many solutions, but a specific solution is obtained by adding an initial condition $x(a) = x_0$. A complete formulation of a first order differential equation is therefore*

$$x' = f(t,x), \qquad x(a) = x_0. \tag{13.7}$$

It is equations of this kind that we will be studying in most of the chapter, with special emphasis on deriving numerical solution algorithms.

13.2.1 A geometric interpretation of first order differential equations

The differential equation in (13.7) has a natural geometric interpretation: At any point (t,x), the equation $x' = f(t,x)$ prescribes the slope of the solution through this point. This is illustrated in figure 13.1a for the differential equation

$$x' = f(t,x) = \frac{\cos 6t}{1+t+x^2}. \tag{13.8}$$

A typical arrow starts at a point (t,x) and has slope given by $x' = f(t,x)$, and therefore shows the tangent to the solution that passes through the point. The image was obtained by picking 1000 points at random and drawing the corresponding tangent at each of the points.

Behind the many arrows in figure 13.1 we perceive a family of wave-like functions. This is shown much more clearly in figure 13.1b. The 11 functions in this figure represent solutions of the differential equation (13.8), each corresponding to one of the initial conditions $x(0) = i/10$ for $i = 0, \dots, 10$.

Observation 13.5 (Geomteric interpretation of differential equation)**.** *The differential equation $x' = f(t,x)$ describes a family of functions whose tangent at the point (t,x) has slope $f(t,x)$. By adding an initial condition $x(a) = x_0$, a particular solution, or solution curve, is selected from the family of solutions.*

13.2.2 Conditions that guarantee existence of one solution

The class of differential equations described by (13.7) is quite general since we have not placed any restrictions on the function f, and this may lead to some problems. Consider for example the equation

$$x' = \sqrt{1-x^2}. \tag{13.9}$$

Since we are only interested in solutions that are real functions, we have to be careful so we do not select initial conditions that lead to square roots of negative numbers. The initial condition $x(0) = 0$ would be fine, as would $x(1) = 1/2$, but $x(0) = 2$ would mean that $x'(0) = \sqrt{1-x(0)^2} = \sqrt{-3}$ which does not make sense.

For the general equation $x' = f(t,x)$ there are many potential pitfalls. As in the example, the function f may involve roots which require the expressions under the roots to be nonnegative, there may be logarithms which require the arguments to be positive, inverse sines or cosines which require the arguments to not exceed 1 in absolute value, fractions which do not make sense if the denominator becomes zero, and combinations of these and other restrictions. On the other hand, there are also many equations that do not require any restrictions on the values of t and x. This is the case when $f(t,x)$ is a polynomial in t and x, possibly combined with sines, cosines and exponential functions.

The above discussion suggests that the differential equation $x' = f(t,x)$ may not have a solution. Or it may have more than one solution if f has certain kinds of problematic behaviour. The most common problem that may occur is that there may be one or more points (t,x) for which $f(t,x)$ is not defined, as with equation (13.9) above. So-called *existence and uniqueness theorems* specify conditions on f which guarantee that a unique solutions can be found. Such theorems may appear rather abstract, and their proofs are often challenging. We are going to quote one such theorem, but the proof requires techniques which are beyond the scope of these notes.

Before we state the theorem, we need to introduce some notation. It turns out that how $f(t,x)$ depends on x influences the solution in an essential way. We therefore need to restrict the behaviour of the derivative of $f(t,x)$ when viewed as a function of x. We will denote this derivative by $\partial f/\partial x$, or sometimes just f_x to save space. If for instance $f(t,x) = t + x^2$, then $f_x(t,x) = 2x$, while if $f(t,x) = \sin(tx)$ then $f_x(t,x) = t\cos(tx)$.

The theorem talks about a rectangle. This is just a set in the plane on the form $\mathbb{A} = [\alpha,\beta] \times [\gamma,\delta]$ and a point (t,x) lies in $\mathbb{A}$ if $t \in [\alpha,\beta]$ and $x \in [\delta,\gamma]$. A point (t,x) is an interior point of $\mathbb{A}$ if it does not lie on the boundary, i.e., if $\alpha < t < \beta$ and $\gamma < x < \delta$.

Theorem 13.6. *Suppose that the functions f and f_x are continuous in the rectangle $\mathbb{A} = [\alpha,\beta] \times [\gamma,\delta]$. If the point (a,x_0) lies in the interior of $\mathbb{A}$ there exists a number $\tau > 0$ such that the differential equation*

$$x' = f(t,x), \quad x(a) = x_0 \tag{13.10}$$

has a unique solution on the interval $[a-\tau, a+\tau]$ which is contained in $[\alpha,\beta]$.

Theorem 13.6 is positive and tells us that if a differential equation is 'nice' near an initial condition, it will have a unique solution that extends both to the left and right of the initial condition. 'Nice' here means that both f and f_x are continuous in a rectangle $\mathbb{A}$ which contains the point (a,x_0) in its interior, i.e., they should have no jumps, should not blow up to infinity, and so on, in $\mathbb{A}$. This is sufficient to prove that the equation has a unique solution, but it is generally not enough to guarantee that a numerical method will converge to the solution. In fact, it is not even sufficient to guarantee that a numerical method will avoid areas where the function f is not defined. For this reason we will strengthen the conditions on f when we state and analyse the numerical methods below and assume that $f(t,x)$ and $f_x(t,x)$ (and sometimes more derivatives) are continuous and bounded for t in some interval $[\alpha,\beta]$ and any real number x.

Notice also that we seek a solution on an interval $[a,b]$. The left end is where the initial condition is and the right end limits the area in which we seek a solution. It should be noted that it is not essential that the initial condition is at the left end; the numerical methods can be easily adapted to work in a situation where the initial condition is at the right end, see exercise 1.

Assumption 13.7. *In the numerical methods for solving the equation*

$$x' = f(t,x), \quad x(a) = x_0,$$

to be introduced below, it is assumed that the function $f(t,x)$ and its derivative $f_x(t,x)$ with respect to x are well-defined, continuous, and bounded in a set $[\alpha,\beta]\times\mathbb{R}$, i.e., for all (t,x) such that $\alpha \le t \le \beta$ and $x \in \mathbb{R}$. It is also assumed that a solution $x(t)$ is sought for t in an interval $[a,b]$ that is strictly contained in $[\alpha,\beta]$, i.e., that $\alpha < a < b < \beta$.

The conditions in assumption 13.7 are quite restrictive and leave out many differential equations of practical interest. However, our focus is on introducing the ideas behind the most common numerical methods and analysing their error, and not on establishing exactly when they will work. It is especially the error analysis that depends on the functions f and f_x (and possibly other derivatives of f) being bounded for all values of t and x that may occur during the computations. In practice, the methods will work for many equations that do not satisfy assumption 13.7.

13.2.3 What is a numerical solution of a differential equation?

In earlier chapters we have derived numerical methods for solving nonlinear equations, for differentiating functions, and for computing integrals. A common feature of all these methods is that the answer is a single number. However, the solution of a differential equation is a function, and we cannot expect to find a single number that can approximate general functions well.

All the methods we derive compute the same kind of approximation: They start at the initial condition $x(a) = x_0$ and then compute successive approximations to the solution at a sequence of points t_1, t_2, t_3, ..., t_n where $a = t_0 < t_1 < t_2 < t_3 < \cdots < t_n = b$.

Fact 13.8 (General strategy for numerical solution of differential equations)**.** *Suppose the differential equation and initial condition*

$$x' = f(t,x), \quad x(a) = x_0$$

are given together with an interval $[a,b]$ where a solution is sought. Suppose also that an increasing sequence of t-values $(t_k)_{k=0}^n$ are given, with $a = t_0$ and $b = t_n$, which in the following will be equally spaced with step length h, i.e.,

$$t_k = a + kh, \quad \text{for } k = 0, \dots, n.$$

A numerical method for solving the equation is a recipe for computing a sequence of numbers x_0, x_1, ..., x_n such that x_k is an approximation to the true *solution $x(t_k)$ at t_k. For $k > 0$, the approximation x_k is computed from one or more of the previous approximations x_{k-1}, x_{k-2}, ..., x_0. A continuous approximation is obtained by connecting neighbouring points by straight lines.*

13.3 Euler's method

Most methods for finding analytical solutions of differential equations appear rather tricky and unintuitive. In contrast, many numerical methods are based on simple, often geometric ideas. The simplest of these methods is *Euler's method* which is based directly on the geometric interpretation in observation 13.5.

13.3.1 Basic idea

We assume that the differential equation is

$$x' = f(t,x), \quad x(a) = x_0,$$

and our aim is to compute a sequence of approximations $(t_k, x_k)_{k=0}^n$ where $t_k = a + kh$. The initial condition provides us with one point on the true solution, so our first point is (t_0, x_0). We compute the slope of the tangent at (t_0, x_0) as $x_0' = f(t_0, x_0)$ which gives us the tangent $T(t) = x_0 + (t - t_0)x_0'$. As the approximation x_1 at t_1 we use the value of the tangent which is given by

$$x_1 = T(t_1) = x_0 + hx_0' = x_0 + hf(t_0, x_0).$$

But now we have a new approximate solution point (t_1, x_1), and from this we can compute the slope $x_1' = f(t_1, x_1)$. This allows us to compute an approximation $x_2 = x_1 + hx_1' = x_1 + hf(t_1, x_1)$ to the solution at t_2. If we continue this we can compute an approximation x_3 to the solution at t_3, then an approximation x_4 at t_4, and so on.

From this description we see that the basic idea is how to advance the approximate solution from a point (t_k, x_k) to a point (t_{k+1}, x_{k+1}).

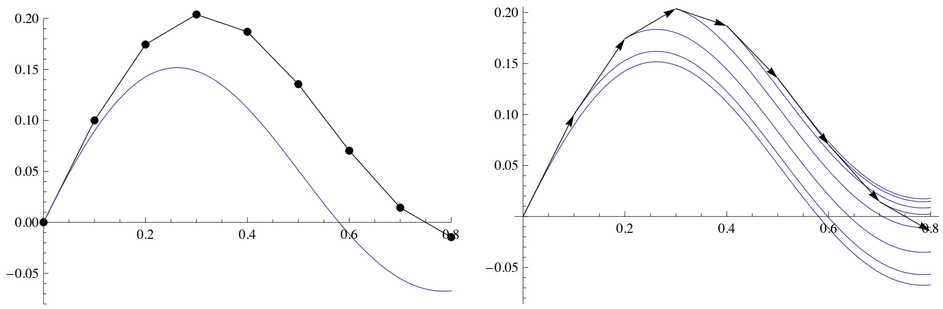
Idea 13.9. *In Euler's method, an approximate solution (t_k, x_k) is advanced to (t_{k+1}, x_{k+1}) by following the tangent*

$$T(t) = x_k + (t - t_k)x_k' = x_k + (t - t_k)f(t_k, x_k)$$

to $t_{k+1} = t_k + h$. This results in the approximation

$$x_{k+1} = x_k + hf(t_k, x_k) \tag{13.11}$$

to $x(t_{k+1})$.



(a)

(b)

Figure 13.2. The plot in (a) shows the approximation produced by Euler's method to the solution of the differential equation $x' = \cos 6t/(1+t+x^2)$ with initial condition $x(0) = 0$ (smooth graph). The plot in (b) shows the same solution augmented with the solution curves that pass through the points produced by Euler's method.

Idea 13.9 shows how we can get from one point on the approximation to the next, while the initial condition $x(a) = x_0$ provides us with a starting point. We therefore have all we need to compute a sequence of approximate points on the solution of the differential equation.

Algorithm 13.10 (Euler's method)**.** *Let the differential equation $x' = f(t,x)$ be given together with the initial condition $x(a) = x_0$, the solution interval $[a,b]$, and the number of steps n. If the following algorithm is performed*

```
h=(b-a)/n;
for $k = 0, 1, \dots, n-1$
    $x_{k+1} = x_k + hf(t_k, x_k)$;
    $t_{k+1} = a + (k+1)h$;
```

the value x_k will be an approximation to the solution $x(t_k)$ of the differential equation, for each $k = 0, 1, \dots, n$.

Figure 13.2 illustrates the behaviour of Euler's method for the differential equation

$$x' = \frac{\cos 6t}{1+t+x^2}, \quad x(0) = 0.$$

This is just a piecewise linear approximation to the solution, see the figure in (a), but the figure in (b) illustrates better how the approximation is obtained. We start off by following the tangent at the initial condition $(0,0)$. This takes us to a point that is slightly above the graph of the true solution. At this point we compute a new tangent and follow this to the next point. However, there is a solution curve that passes through this second point, and the line from the second to the third point is tangent to the solution curve which has the second point as initial condition. We therefore see that as we compute new approximate points on the solution, we jump between different solution curves of the differential equation $x' = f(t,x)$.

Note that Euler's method can also be obtained via a completely different argument. A common approximation to the derivative of x is given by

$$x'(t) \approx \frac{x(t+h) - x(t)}{h}.$$

If we rewrite this and make use of the fact that $x'(t) = f\big(t, x(t)\big)$, we find that

$$x(t+h) \approx x(t) + hf\big(t, x(t)\big)$$

which is the basis for Euler's method.

13.3.2 Error analysis

We know that Euler's method in most cases just produces an approximation to the true solution of the differential equation, but how accurate is the approximation? To answer this question we need to think more carefully about the various approximations involved.

The basic idea in Euler's method is to advance the solution from (t_k, x_k) to (t_{k+1}, x_{k+1}) with the relation

$$x_{k+1} = x_k + hf(t_k, x_k) \tag{13.12}$$

which stems from the approximation $x(t_{k+1}) \approx x(t_k) + hx'(t_k)$. If we include the error term in this simple Taylor polynomial, we obtain the identity

$$x(t_{k+1}) = x(t_k) + hx'(t_k) + \frac{h^2}{2}x''(\xi_k) = x(t_k) + hf\big(t_k, x(t_k)\big) + \frac{h^2}{2}x''(\xi_k), \tag{13.13}$$

where ξ_k is a number in the interval (t_k, t_{k+1}). We subtract (13.12) and end up with

$$x(t_{k+1}) - x_{k+1} = x(t_k) - x_k + h\Big(f\big(t_k, x(t_k)\big) - f(t_k, x_k)\Big) + \frac{h^2}{2}x''(\xi_k). \tag{13.14}$$

The number $\epsilon_{k+1} = x(t_{k+1}) - x_{k+1}$ is the global error accumulated by Euler's method at t_{k+1}. This error has two sources:

1. The *global error* $\epsilon_k = x(t_k) - x_k$ accumulated up to the previous step. This also leads to an error in computing $x'(t_k)$ since we use the value $f(t_k, x_k)$ instead of the correct value $f\big(t_k, x(t_k)\big)$.

2. The *local error* we commit when advancing from (t_k, x_k) to $\bigl(t_{k+1}, x_{k+1}\bigr)$ and ignore the remainder in Taylor's formula,
$$\frac{h^2}{2}x''(\xi_k).$$

The right-hand side of (13.14) can be simplified a little bit by noting that
$$f\bigl(t_k, x(t_k)\bigr) - f(t_k, x_k) = f_x(t_k, \theta_k)\bigl(x(t_k) - x_k\bigr) = f_x(t_k, \theta_k)\epsilon_k,$$
where θ_k is a number in the interval $\bigl(x_k, x(t_k)\bigr)$. The result is summarised in the following lemma.

Lemma 13.11. *If the two first derivatives of f exist, the error in using Euler's method for solving $x' = f(t,x)$ develops according to the relation*
$$\epsilon_{k+1} = \bigl(1 + hf_x(t_k, \theta_k)\bigr)\epsilon_k + \frac{h^2}{2}x''(\xi_k). \tag{13.15}$$
where ξ_k is a number in the interval (t_k, t_{k+1}) and θ_k is a number in the interval $\bigl(x_k, x(t_k)\bigr)$. In other words, the global error *at step $k+1$ has two sources:*

1. *The advancement of the global error at step k to the next step*
$$\bigl(1 + hf_x(t_k, \theta_k)\bigr)\epsilon_k.$$

2. *The* local truncation error *committed by only including two terms in the Taylor polynomial,*
$$h^2 x''(\xi_k)/2.$$

The lemma tells us how the error develops from one stage to the next, but we would really like to know explicitly what the global error at step k is. For this we need to simplify (13.15) a bit. The main complication is the presence of the two numbers θ_k and ξ_k which we know very little about. We use a standard trick: We take absolute values in (13.15) and replace the two terms $|f_x(t_k, \theta_k)|$ and $|x''(\xi_k)|$ by their maximum values,
$$\begin{aligned}
|\epsilon_{k+1}| &= \Bigl|\bigl(1 + hf_x(t_k, \theta_k)\bigr)\epsilon_k + \frac{h^2}{2}x''(\xi_k)\Bigr| \\
&\le \Bigl|1 + hf_x(t_k, \theta_k)\Bigr||\epsilon_k| + \frac{h^2}{2}|x''(\xi_k)| \\
&\le (1 + hC)|\epsilon_k| + \frac{h^2}{2}D.
\end{aligned}$$

This is where the restrictions on f and f_x that we mentioned in assumption 13.7 are needed: We need the two maximum values used to define the constants $D = \max_{t\in[a,b]}|x''(t)|$ and $C = \max_{t\in[a,b]}|f_x(t,x(t))|$ to exist. To simplify notation we write $\tilde{C} = 1 + hC$ and $\tilde{D} = Dh^2/2$, so the final inequality is

$$|\epsilon_{k+1}| \leq \tilde{C}|\epsilon_k| + \tilde{D}$$

which is valid for $k = 0, 1, \dots, n-1$. This is a 'difference inequality'which can be solved quite easily. We do this by unwrapping the error terms,

$$\begin{aligned}
|\epsilon_{k+1}| &\leq \tilde{C}|\epsilon_k| + \tilde{D} \\
&\leq \tilde{C}\big(\tilde{C}|\epsilon_{k-1}| + \tilde{D}\big) + \tilde{D} = \tilde{C}^2|\epsilon_{k-1}| + \big(1+\tilde{C}\big)\tilde{D} \\
&\leq \tilde{C}^2\big(\tilde{C}|\epsilon_{k-2}| + \tilde{D}\big) + \big(1+\tilde{C}\big)\tilde{D} \\
&\leq \tilde{C}^3|\epsilon_{k-2}| + \big(1+\tilde{C}+\tilde{C}^2\big)\tilde{D} \\
&\;\vdots \\
&\leq \tilde{C}^{k+1}|\epsilon_0| + \big(1+\tilde{C}+\tilde{C}^2+\cdots+\tilde{C}^k\big)\tilde{D}.
\end{aligned} \tag{13.16}$$

We note that $\epsilon_0 = x(a) - x_0 = 0$ because of the initial condition, and the sum we recognise as a geometric series. This means that

$$|\epsilon_{k+1}| \leq \tilde{D}\sum_{i=0}^{k}\tilde{C}^i = \tilde{D}\frac{\tilde{C}^{k+1}-1}{\tilde{C}-1}.$$

We insert the values for $\tilde{C}$ and $\tilde{D}$ and obtain

$$|\epsilon_{k+1}| \leq hD\frac{(1+hC)^{k+1}-1}{2C}. \tag{13.17}$$

Let us sum up our findings and add some further refinements.

Theorem 13.12 (Error in Euler's method)**.** *Suppose that f, f_t and f_x are continuous and bounded functions on the rectangle $\mathbb{A} = [\alpha,\beta]\times\mathbb{R}$ and that the interval $[a,b]$ satisfies $\alpha < a < b < \beta$. Let $\epsilon_k = x(t_k) - x_k$ denote the error at step k in applying Euler's method with n steps of length h to the differential equation $x' = f(t,x)$ on the interval $[a,b]$, with initial condition $x(a) = x_0$. Then*

$$|\epsilon_k| \leq h\frac{D}{2C}\Big(e^{(t_k-a)C}-1\Big) \leq h\frac{D}{2C}\Big(e^{(b-a)C}-1\Big) \tag{13.18}$$

for $k = 0, 1, \dots, n$ where the constants C and D are given by

$$C = \max_{(t,x)\in\mathbb{A}} |f_x(t,x)|,$$
$$D = \max_{t\in[a,b]} |x''(t)|.$$

Proof. From Taylor's formula with remainder we know that $e^t = 1+t+t^2e^\eta/2$ for any positive, real number t, with η some real number in the interval $(0,t)$ (the interval $(t,0)$ if $t<0$). We therefore have $1+t \le e^t$ and therefore $(1+t)^k \le e^{kt}$. If we apply this to (13.17), with $k+1$ replaced by k, we obtain

$$|\epsilon_k| \le \frac{hD}{2C} e^{khC},$$

and from this the first inequality in (13.18) follows since $kh = t_k - a$. The last inequality is then immediate since $t_k - a \le b - a$.

We will see in lemma 13.18 that $x'' = f_t + f_x f$. By assuming that f, f_t and f_x are continuous and bounded we are therefore assured that x'' is also continuous, and therefore that the constant D exists. ∎

The error estimate (13.18) depends on the quantities h, D, C, a and b. Of these, all except h are given by the differential equation itself, and therefore beyond our control. The step length h, however, can be varied as we wish, and the most interesting feature of the error estimate is therefore how the error depends on h. This is often expressed as

$$|\epsilon_k| \le O(h)$$

which simply means that $|\epsilon_k|$ is bounded by a constant times the step length h, just like in (13.18), without any specification of what the constant is. The error in numerical methods for solving differential equations typically behave like this.

Definition 13.13 (Accuracy of a numerical method)**.** *A numerical method for solving differential equations with step length h is said to be of order p if the error ϵ_k at step k satisfies*

$$|\epsilon_k| \le O(h^p).$$

The significance of the concept of order is that it tells us how quickly the error goes to zero with h. If we first try to run the numerical method with step length h and then reduce the step length to $h/2$ we see that the error will roughly be reduced by a factor $1/2^p$. So the larger the value of p, the better the method, at least from the point of view of accuracy.

The accuracy of Euler's method can now be summed up quite concisely.

Corollary 13.14. *Euler's method is of order 1.*

In other words, if we halve the step length we can expect the error in Euler's method to also be halved. This may be a bit surprising in view of the fact that the local error in Euler's method is $O(h^2)$, see lemma 13.11. The explanation is that although the error committed in replacing $x(t_{k+1})$ by $x_k + hf(t_k, x_k)$ is bounded by Kh^2 for a suitable constant K, the error accumulates so that the global order becomes 1 even though the local approximation order is 2.

13.4 Differentiating the differential equation

Our next aim is to develop a whole family of numerical methods that can attain any order of accuracy, namely the Taylor methods. For these methods however, we need to know how to determine higher order derivatives of the solution of a differential equation at a point, and this is the topic of the current section.

We consider the standard equation

$$x' = f(t,x), \quad x(a) = x_0. \tag{13.19}$$

The initial condition explicitly determines a point on the solution, namely the point given by $x(a) = x_0$, and we want to compute the derivatives $x'(a)$, $x''(a)$, $x'''(a)$ and so on. It is easy to determine the derivative of the solution at $x = a$ since

$$x'(a) = f\big(a, x(a)\big) = f(a, x_0).$$

To determine higher derivatives, we simply differentiate the differential equation. This is best illustrated by an example.

Example 13.15. Suppose the equation is $x' = t + x^2$, or more explicitly,

$$x'(t) = t + x(t)^2, \quad x(a) = x_0. \tag{13.20}$$

At $x = a$ we know that $x(a) = x_0$, while the derivative is given by the differential equation

$$x'(a) = a + x_0^2.$$

If we differentiate the differential equation, the chain rule yields

$$x''(t) = 1 + 2x(t)x'(t) = 1 + 2x(t)\big(t + x(t)^2\big) \tag{13.21}$$

where we have inserted the expression for $x'(t)$ given by the differential equation (13.20). This means that at any point t where $x(t)$ (the solution) and $x'(t)$ (the derivative of the solution) is known, we can also determine the second derivative of the solution. In particular, at $x = a$, we have

$$x''(a) = 1 + 2x(a)x'(a) = 1 + 2x_0(a + x_0^2).$$

Note that the relation (13.21) is valid for any value of t, but since the right-hand side involves $x(t)$ and $x'(t)$ these quantities must be known. The derivative in turn only involves $x(t)$, so at a point where $x(t)$ is known, we can determine both $x'(t)$ and $x''(t)$.

What about higher derivatives? If we differentiate (13.21) once more, we find

$$x'''(t) = 2x'(t)x'(t) + 2x(t)x''(t) = 2\bigl(x'(t)^2 + x(t)x''(t)\bigr). \tag{13.22}$$

The previous formulas express $x'(t)$ and $x''(t)$ in terms of $x(t)$ and if we insert this at $x = a$ we obtain

$$x'''(a) = 2\bigl(x'(a)^2 + x(a)x''(a)\bigr) = 2\Bigl(\bigl(a + x_0^2\bigr)^2 + x_0\bigl(1 + 2x_0(a + x_0^2)\bigr)\Bigr).$$

In other words, at any point t where the solution $x(t)$ is known, we can also determine $x'(t)$, $x''(t)$ and $x'''(t)$. And by differentiating (13.22) the required number of times, we see that we can in fact determine any derivative $x^{(n)}(t)$ at a point where $x(t)$ is known. ■

It is important to realise the significance of example 13.15. Even though we do not have a general formula for the solution $x(t)$ of the differential equation, we can easily find explicit formulas for the derivatives of x at a single point where the solution is known. One particular such point is the point where the initial condition is given. One obvious restriction is that the derivatives must exist.

Lemma 13.16 (Determining derivatives)**.** *Let $x' = f(t,x)$ be a differential equation with initial condition $x(a) = x_0$, and suppose that the derivatives of $f(t,x)$ of order $p-1$ exist at the point (a, x_0). Then the pth derivative of the solution $x(t)$ at $x = a$ can be expressed in terms of a and x_0, i.e.,*

$$x^{(p)}(a) = F_p(a, x_0), \tag{13.23}$$

where F_p is a function defined by f and its derivatives of order less than p.

Proof. The proof is essentially the same as in example 13.15, but since f is not known explicitly, the argument becomes a bit more abstract. We use induction on n, the order of the derivative. For $p=1$, equation (13.23) just says that $x'(a) = F_1(a, x_0)$. In this situation we therefore have $F_1 = f$ and the result is immediate.

Suppose now that the result has been shown to be true for $p=k$, i.e., that

$$x^{(k)}(a) = F_k(a, x_0), \tag{13.24}$$

where F_k depends on f and its derivatives of order less than k; we must show that it is also true for $p = k+1$. To do this we differentiate both sides of (13.24) and obtain

$$x^{(k+1)}(a) = \frac{\partial F_k}{\partial t}(a, x_0) + \frac{\partial F_k}{\partial x}(a, x_0)x'(a). \tag{13.25}$$

The right-hand side of (13.25) defines F_{k+1},

$$F_{k+1}(a, x_0) = \frac{\partial F_k}{\partial t}(a, x_0) + \frac{\partial F_k}{\partial x}(a, x_0) f(a, x_0),$$

where we have also used the fact that $x'(a) = f(a, x_0)$. Since F_k involves partial derivatives of f up to order $k-1$, it follows that F_{k+1} involves partial derivatives of f up to order k. This proves all the claims in the theorem. ∎

Example 13.17. The function F_p that appears in Lemma 13.16 may seem a bit mysterious, but if we go back to example 13.15, we see that it is in fact quite straightforward. In this specific case we have

$$x' = F_1(t, x) = f(t, x) = t + x^2, \tag{13.26}$$

$$x'' = F_2(t, x) = 1 + 2xx' = 1 + 2tx + 2x^3, \tag{13.27}$$

$$x''' = F_3(t, x) = 2(x'^2 + xx'') = 2\big((t + x^2)^2 + x(1 + 2tx + 2x^3)\big). \tag{13.28}$$

This shows the explicit expressions for F_1, F_2 and F_3. The expressions can usually be simplified by expressing x'' in terms of t, x and x', and by expressing x''' in terms of t, x, x' and x'', as shown in the intermediate formulas in (13.26)–(13.28). ∎

For later reference we record the general formulas for the first three derivatives of x in terms of f.

Lemma 13.18. *Let $x(t)$ be a solution of the differential equation $x' = f(t, x)$. Then*

$$x' = f, \quad x'' = f_t + f_x f, \quad x''' = f_{tt} + 2f_{tx} f + f_{xx} f^2 + f_t f_x + f_x^2 f,$$

at any point where the derivatives exist.

Lemma 13.16 tells us that at some point t where we know the solution $x(t)$, we can also determine all derivatives of the solution, just as we did in example 13.15. The obvious place where this can be exploited is at the initial condition. But this property also means that if in some way we have determined an approximation $\hat{x}$ to $x(t)$ we can compute approximations to all derivatives at t as well. Consider again example 13.15 and let us imagine that we have an approximation $\hat{x}$ to the solution $x(t)$ at t. We can then successively compute the approximations

$$\begin{aligned}
x'(t) &\approx \hat{x}' = F_1(t,\hat{x}) = f(t,\hat{x}) = x + \hat{x}^2,\\
x''(t) &\approx \hat{x}'' = F_2(t,\hat{x}) = 1 + 2\hat{x}\hat{x}',\\
x'''(t) &\approx \hat{x}''' = F_3(t,\hat{x}) = 2(\hat{x}'^2 + \hat{x}\hat{x}'').
\end{aligned}$$

This corresponds to finding the exact derivatives of the solution curve that has the value $\hat{x}'$ at t. The same is of course true for a general equation.

The fact that all derivatives of the solution of a differential equation at a point can be computed as in lemma 13.16 is the foundation of a whole family of numerical methods for solving differential equations.

13.5 Taylor methods

In this section we are going to derive the family of numerical methods that are usually referred to as Taylor methods. An important ingredient in these methods is the computation of derivatives of the solution at a single point which we discussed in section 13.4. We first introduce the idea behind the methods and the resulting algorithms, and then discuss the error. We focus on the quadratic case as this is the simplest, but also illustrates the general principle.

13.5.1 Derivation of the Taylor methods

The idea behind Taylor methods is to approximate the solution by a Taylor polynomial of a suitable degree. In Euler's method, which is the simplest Taylor method, we used the approximation

$$x(t+h) \approx x(t) + hx'(t).$$

The quadratic Taylor method is based on the more accurate approximation

$$x(t+h) \approx x(t) + hx'(t) + \frac{h^2}{2}\,x''(t). \tag{13.29}$$

To describe the algorithm, we need to specify how the numerical solution can be advanced from a point (t_k, x_k) to a new point (t_{k+1}, x_{k+1}) with $t_{k+1} = t_k + h$.

The idea is to use (13.29) and compute x_{k+1} as

$$x_{k+1} = x_k + hx'_k + \frac{h^2}{2}x''_k. \tag{13.30}$$

The numbers x_k, x'_k and x''_k are approximations to the function value and derivatives of the solution at t. These are obtained via the recipe in lemma 13.16. An example should make this clear.

Example 13.19. Let us consider the differential equation

$$x' = f(t,x) = F_1(t,x) = t - \frac{1}{1+x}, \quad x(0) = 1, \tag{13.31}$$

which we want to solve on the interval $[0,1]$. To illustrate the method, we choose a large step length $h = 0.5$ and attempt to find an approximate numerical solution at $x = 0.5$ and $x = 1$ using a quadratic Taylor method.

From (13.31) we obtain

$$x''(t) = F_2(t,x) = 1 + \frac{x'(t)}{\bigl(1+x(t)\bigr)^2}. \tag{13.32}$$

To compute an approximation to $x(h)$ we use the quadratic Taylor polynomial

$$x(h) \approx x_1 = x(0) + hx'(0) + \frac{h^2}{2}x''(0).$$

The differential equation (13.31) and (13.32) yield

$$\begin{aligned} x(0) &= x_0 = 1, \\ x'(0) &= x'_0 = 0 - 1/2 = -1/2, \\ x''(0) &= x''_0 = 1 - 1/8 = 7/8, \end{aligned}$$

which leads to the approximation

$$x(h) \approx x_1 = x_0 + hx'_0 + \frac{h^2}{2}x''_0 = 1 - \frac{h}{2} + \frac{7h^2}{16} = 0.859375.$$

To prepare for the next step we need to determine approximations to $x'(h)$ and $x''(h)$ as well. From the differential equation (13.31) and (13.32) we find

$$\begin{aligned} x'(h) &\approx x'_1 = F_1(t_1, x_1) = t_1 - 1/(1+x_1) = -0.037815126, \\ x''(h) &\approx x''_1 = F_2(t_1, x_1) = 1 + x'_1/(1+x_1)^2 = 0.98906216, \end{aligned}$$

rounded to eight digits. From this we can compute the approximation

$$x(1) = x(2h) \approx x_2 = x_1 + hx'_1 + \frac{h^2}{2}x''_1 = 0.96410021.$$

The result is shown in figure 13.3a. ∎

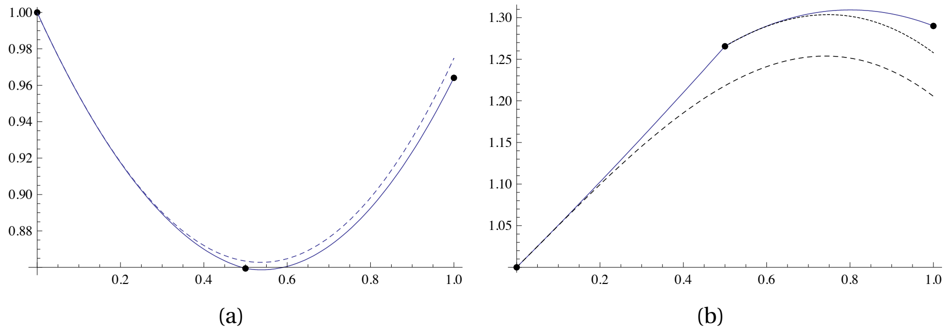


Figure 13.3. The plots show the result of solving a differential equation numerically with the quadratic Taylor method. The plot in (a) show the first two steps for the equation $x' = t - 1/(1+x)$ with $x(0) = 1$ and $h = 0.5$, while the plot in (b) show the first two steps for the equation $x' = \cos(3t/2) - 1/(1+x)$ with $x(0) = 1$ and $h = 0.5$. The dots show the computed approximations, while the solid curves show the parabolas that are used to compute the approximations. The exact solution is shown by the dashed curve in both cases.

Figure 13.3 illustrates the first two steps of the quadratic Talor method for two equations. The solid curve shows the two parabolas used to compute the approximate solution points in both cases. In figure (a) it seems like the two parabolas join together smoothly, but this is just a feature of the underlying differential equation. The behaviour in (b), where the two parabolas meet at a slight corner is more representative, although in this case, the first parabola is almost a straight line. In practice the solution between two approximate solution points will usually be approximated by a straight line, not a parabola.

Let us record the idea behind the quadratic Taylor method.

Idea 13.20 (Quadratic Taylor method)**.** *The quadratic Taylor method advances the solution from a point (t_k, x_k) to a point (t_{k+1}, x_{k+1}) by evaluating the approximate Taylor polynomial*

$$x(t) \approx x_k + (t - t_k)x'_k + \frac{(t-t_k)^2}{2}x''_k$$

at $x = t_{k+1}$. In other words, the new value x_{k+1} is given by

$$x_{k+1} = x_k + hx'_k + \frac{h^2}{2}x''_k$$

where the values x_k, x'_k and x''_k are obtained as described in lemma 13.16 and $h = t_{k+1} - t_k$.

This idea is easily translated into a simple algorithm. At the beginning of a new step, we know the previous approximation x_k, but need to compute the approximations to x'_k and x''_k. Once these are known we can compute x'_{k+1} and t_{k+1} before we proceed with the next step. Note that in addition to the function $f(t,x)$ which defines the differential equation we also need the function F_2 which defines the second derivative, as in lemma 13.16. This is usually determined by manual differentiation as in the examples above.

Algorithm 13.21 (Quadratic Taylor method)**.** *Let the differential equation $x' = f(t,x)$ be given together with the initial condition $x(a) = x_0$, the solution interval $[a,b]$ and the number of steps n, and let the function F_2 be such that $x''(t) = F_2\big(t, x(t)\big)$. The quadratic Taylor method is given by the algorithm*

```
$h = (b-a)/n;$
$t_0 = a;$
for $k = 0, 1, \dots, n-1$
    $x'_k = f(t_k, x_k);$
    $x''_k = F_2(t_k, x_k);$
    $x_{k+1} = x_k + hx'_k + h^2 x''_k/2;$
    $t_{k+1} = a + (k+1)h;$
```

After these steps the value x_k will be an approximation to the solution $x(t_k)$ of the differential equation, for each $k = 0, 1, \dots, n$.

The quadratic Taylor method is easily generalised to higher degrees by including more terms in the Taylor polynomial. The *Taylor method of degree p* uses the formula

$$x_{k+1} = x_k + hx'_k + \frac{h^2}{2}x''_k + \cdots + \frac{h^{p-1}}{(p-1)!}x_k^{(p-1)} + \frac{h^p}{p!}x_k^{(p)} \tag{13.33}$$

to advance the solution from the point (t_k, x_k) to (t_{k+1}, x_{k+1}). Just like for the quadratic method, the main challenge is the determination of the derivatives, whose complexity may increase quickly with the degree. It is possible to make use of software for symbolic computation to produce the derivatives, but it is much more common to use a numerical method that mimics the behaviour of the Taylor methods by evaluating $f(t,x)$ at intermediate steps instead of computing higher order derivatives, like the *Runge-Kutta methods* in section 13.6.3.

13.5.2 Error analysis for Taylor methods

In section 13.3.2 we discussed the error in Euler's method. In this section we use the same technique to estimate the error in the Taylor methods.

The Taylor method of degree p advances the numerical solution from the point (t_k, x_k) to (t_{k+1}, x_{k+1}) with the formula (13.33). This is based on the exact relation

$$x(t_{k+1}) = x(t_k) + hx'(t_k) + \cdots + \frac{h^p}{p!}x^{(p)}(t_k) + \frac{h^{p+1}}{(p+1)!}x^{(p+1)}(\xi_k), \tag{13.34}$$

where ξ_k is a number in the interval (t_k, t_{k+1}). When we omit the last term on the right and use (13.33) instead we have a local truncation error given by

$$\frac{h^{p+1}}{(p+1)!}x^{(p+1)}(\xi_k),$$

i.e., of order $O(h^{p+1})$. As for Euler's method, the challenge is to see how this local error at each step filters through and leads to the global error.

We will follow a procedure very similar to the one that we used to estimate the error in Euler's method. We start by rewriting (13.33) as

$$x_{k+1} = x_k + h\Phi(t_k, x_k, h) \tag{13.35}$$

with the function Φ given by

$$\begin{aligned}\Phi(t_k, x_k, h) &= x'_k + \frac{h}{2}x''_k + \cdots + \frac{h^{p-1}}{(p-1)!}x_k^{(p-1)}\\ &= F_1(t_k, x_k) + \frac{h^2}{2}F_2(t_k, x_k) + \cdots + \frac{h^{p-1}}{(p-1)!}F_{p-1}(t_k, x_k),\end{aligned} \tag{13.36}$$

where F_1, F_2, …, F_{p-1} are the functions given in lemma 13.16. With the same notation we can write (13.34) as

$$x(t_{k+1}) = x(t_k) + h\Phi\big(t_k, x(t_k), h\big) + \frac{h^{p+1}}{(p+1)!}x^{(p+1)}(\xi_k). \tag{13.37}$$

The first step is to derive the relation which corresponds to (13.14) by subtracting (13.35) from (13.37),

$$x\big(t_{k+1}\big) - x_{k+1} = x(t_k) - x_k + h\Big(\Phi\big(t_k, x(t_k), h\big) - \Phi(t_k, x_k, h)\Big) + \frac{h^{p+1}}{(p+1)!}x^{(p+1)}(\xi_k).$$

We introduce the error $\epsilon_k = x(t_k) - x_k$ and rewrite this as

$$\epsilon_{k+1} = \epsilon_k + h\Big(\Phi\big(t_k, x(t_k), h\big) - \Phi(t_k, x_k, h)\Big) + \frac{h^{p+1}}{(p+1)!}x^{(p+1)}(\xi_k). \tag{13.38}$$

In the following we assume that the derivatives of the functions $\{F_i\}_{i=1}^{p-1}$ with respect to x exist. This means that the derivative of $\Phi(t,x)$ with respect to x also exist, so the mean value theorem means that (13.38) can be written as

$$\begin{aligned}\epsilon_{k+1} &= \epsilon_k + h\frac{\partial\Phi}{\partial x}(t_k,\theta_k,h)\epsilon_k + \frac{h^{p+1}}{(p+1)!}x^{(p+1)}(\xi_k)\\ &= \Big(1+h\frac{\partial\Phi}{\partial x}(t_k,\theta_k,h)\Big)\epsilon_k + \frac{h^{p+1}}{(p+1)!}x^{(p+1)}(\xi_k),\end{aligned} \tag{13.39}$$

where θ_k is a number in the interval $\big(x_k, x(t_k)\big)$. This relation is similar to equation (13.15), and the rest of the analysis can be performed as in section 13.3.2. We take absolute values, use the triangle inequality, and introduce the constants C and D. This gives us the inequality

$$|\epsilon_{k+1}| \le (1+hC)|\epsilon_k| + \frac{h^{p+1}}{(p+1)!}D. \tag{13.40}$$

Proceedings just as in section 13.3.2 we end up with an analogous result.

Theorem 13.22 (Error in Taylor method of degree p). *Let $\epsilon_k = x(t_k) - x_k$ denote the error at step k in applying the Taylor method of degree p with n steps of length h to the differential equation $x' = f(t,x)$ on the interval $[a,b]$, with initial condition $x(a) = x_0$. Suppose that the derivatives of f of order p exist and are continuous in a set $[\alpha,\beta]\times\mathbb{R}$ with $\alpha < a < b < \beta$. Then*

$$|\epsilon_k| \le h^p\frac{D}{C(p+1)!}\Big(e^{(t_k-a)C}-1\Big) \le h^p\frac{D}{C(p+1)!}\Big(e^{(b-a)C}-1\Big) \tag{13.41}$$

for $k = 0, 1, \dots, n$ where

$$\begin{aligned}C &= \max_{(t,x)\in\mathbb{A}}\left|\frac{\partial\Phi}{\partial x}\big(t,x(t)\big)\right|,\\ D &= \max_{t\in[a,b]}|x^{(p+1)}(t)|.\end{aligned}$$

Note that even though the local truncation error in (13.34) is $O(h^{p+1})$, the global approximation order is p. In other words, the local errors accumulate so that we lose one approximation order in passing from local error to global error, just like for Euler's method. In fact the error analysis we have used here both for Euler's method and for the Taylor methods (which include Euler's method as a special case), work in quite general situations, and below we will use it to analyse other methods as well.

We end the section with a shorter version of theorem 13.22.

Corollary 13.23. *The Taylor method of degree p has global approximation order p.*

13.6 Other methods

The big advantage of the Taylor methods is that they can attain any approximation order. Their disadvantage is that they require symbolic differentiation of the differential equation (except for Euler's method). In this section we are going to develop some methods of higher order than Euler's method that do not require differentiation of the differential equation. Instead they advance from (t_k, x_k) to (t_{k+1}, x_{k+1}) by evaluating $f(t,x)$ at intermediate points in the interval $[t_k, t_{k+1}]$.

13.6.1 Euler's midpoint method

The first method we consider is a simple extension of Euler's method. If we look at the plots in figure 13.2, we notice how the tangent is a good approximation to a solution curve at the initial condition, but the quality of the approximation deteriorates as we move to the right. One way to improve on Euler's method is therefore to estimate the slope of each line segment better. In *Euler's midpoint method* this is done via a two-step procedure which aims to estimate the slope at the midpoint between the two solution points. In proceeding from (t_k, x_k) to (t_{k+1}, x_{k+1}) we would like to use the tangent to the solution curve at the midpoint $t_k + h/2$. But since we do not know the value of the solution curve at this point, we first compute an approximation $x_{k+1/2}$ to the solution at $t_k + h/2$ using the traditional Euler's method. Once we have this approximation, we can determine the slope of the solution curve that passes through the point and use this as the slope for a straight line that we follow from t_k to t_{k+1} to determine the new approximation x_{k+1}. This idea is illustrated in figure 13.4.

Idea 13.24 (Euler's midpoint method)**.** *In Euler's midpoint method the solution is advanced from (t_k, x_k) to $(t_k + h, x_{k+1})$ in two steps: First an approximation to the solution is computed at the midpoint $t_k + h/2$ by using Euler's method with step length $h/2$,*

$$x_{k+1/2} = x_k + \frac{h}{2} f(t_k, x_k).$$

Then the solution is advanced to t_{k+1} by following the straight line from (t_k, x_k) with slope given by $f(t_k + h/2, x_{k+1/2})$,

$$x_{k+1} = x_k + h f(t_k + h/2, x_{k+1/2}).$$

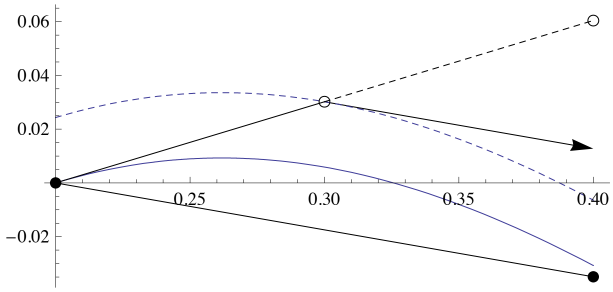


Figure 13.4. The figure illustrates the first step of the midpoint Euler method, starting at $x = 0.2$ and with step length $h = 0.2$. We start by following the tangent at the starting point ($x = 0.2$) to the midpoint ($x = 0.3$). Here we determine the slope of the solution curve that passes through this point and use this as the slope for a line through the starting point. We then follow this line to the next t-value ($x = 0.4$) to determine the first approximate solution point. The solid curve is the correct solution and the open circle shows the approximation produced by Euler's method.

Once the basic idea is clear it is straightforward to translate this into a complete algorithm for computing an approximate solution to the differential equation.

Algorithm 13.25 (Euler's midpoint method)**.** *Let the differential equation $x' = f(t,x)$ be given together with the initial condition $x(a) = x_0$, the solution interval $[a,b]$ and the number of steps n. Euler's midpoint method is given by*

```
$h = (b-a)/n$;
for $k = 0, 1, \dots, n-1$
    $x_{k+1/2} = x_k + hf(t_k, x_k)/2$;
    $x_{k+1} = x_k + hf(t_k + h/2, x_{k+1/2})$;
    $t_{k+1} = a + (k+1)h$;
```

After these steps the value x_k will be an approximation to the solution $x(t_k)$ of the differential equation, for each $k = 0, 1, \dots, n$.

As an alternative viewpoint, let us recall the two approximations for numerical differentiation given by

$$\begin{aligned} x'(t) &\approx \frac{x(t+h)-x(t)}{h}, \\ x'(t+h/2) &\approx \frac{x(t+h)-x(t)}{h}. \end{aligned}$$

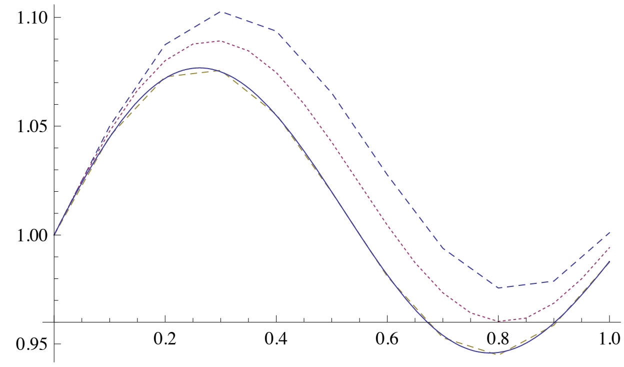


Figure 13.5. Comparison of Euler's method and Euler's midpoint method for the differential equation $x' = \cos(6t)/(1+t+x^2)$ with initial condition $x(0) = 1$ with step length $h = 0.1$. The solid curve is the exact solution and the two approximate solutions are dashed. The dotted curve in the middle is the approximation produced by Euler's method with step length $h = 0.05$. The approximation produced by Euler's midpoint method appears almost identical to the exact solution.

As we saw above, the first one is the basis for Euler's method, but we know from our study of numerical differentiation that the second one is more accurate. If we solve for $x(t+h)$ we find

$$x(t+h) \approx x(t) + hx'(t+h/2)$$

and this relation is the basis for Euler's midpoint method.

In general Euler's midpoint method is more accurate than Euler's method since it is based on a better approximation of the first derivative, see Figure 13.5 for an example. However, this extra accuracy comes at a cost: the midpoint method requires two evaluations of $f(t,x)$ per iteration instead of just one for the regular method. In many cases this is insignificant, although there may be situations where f is extremely complicated and expensive to evaluate, or the added evaluation may just not be feasible. But even then it is generally better to use Euler's midpoint method with a double step length, see figure 13.5.

13.6.2 Error analysis for Euler's midpoint method

In this section we are going to analyse the error in Euler's midpoint method with the same technique as was used to analyse Euler's method and the Taylor methods. From idea 13.24 we recall that the approximation is advanced from (t_k, x_k) to (t_{k+1}, x_{k+1}) with the formula

$$x_{k+1} = x_k + hf\big(t_k + h/2, x_k + hf(t_k, x_k)/2\big). \tag{13.42}$$

The idea behind the analysis is to apply Taylor's formula and replace the outer evaluation of f to an evaluation of f and its derivatives at (t_k, x_k), and then subtract a Taylor expansion of f, in analogy to the analysis of Euler's method. We first do a Taylor expansion with respect to the first variable in (13.42),

$$f(t_k+h/2,x) = f(t_k,x) + \frac{h}{2} f_t(t_k,x) + \frac{h^2}{8} f_{tt}(t_k,x) + O(h^3), \tag{13.43}$$

where $x = x_k + hf(t_k,x_k)/2$, and the error is indicated by the $O(h^3)$ term. We then replace each of the function terms by Taylor expansions with respect to the second variable about x_k,

$$\begin{aligned}
f(t_k,x) &= f + \frac{hf}{2} f_x + \frac{h^2f^2}{8} f_{xx} + O(h^3),\\
f_t(t_k,x) &= f_t + \frac{hf}{2} f_{tx} + O(h^2),\\
f_{tt}(t_k,x) &= f_{tt} + O(h),
\end{aligned}$$

where the functions with no arguments are evaluated at (t_k,x_k). If we insert this in (13.43) we obtain

$$\begin{aligned}
f(t_k+h/2,x) &= f + \frac{h}{2} f_x f + \frac{h^2}{8} f_{xx} f^2 + \frac{h}{2} f_t + \frac{h^2}{4} f_{tx} f + \frac{h^2}{8} f_{tt} + O(h^3)\\
&= f + \frac{h}{2}(f_t + f_x f) + \frac{h^2}{8}(f_{tt} + 2f_{tx} f + f_{xx} f^2) + O(h^3).
\end{aligned}$$

This means that (13.42) can be written

$$x_{k+1} = x_k + hf + \frac{h^2}{2}(f_t + f_x f) + \frac{h^3}{8}(f_{tt} + 2f_{tx} f + f_{xx} f^2) + O(h^4). \tag{13.44}$$

On the other hand, a standard Taylor expansion of $x(t_{k+1})$ about t_k with remainder yields

$$\begin{aligned}
x(t_{k+1}) &= x(t_k) + hx'(t_k) + \frac{h^2}{2} x''(t_k) + \frac{h^3}{6} x'''(\xi_k)\\
&= x(t_k) + hf\big(t_k,x(t_k)\big) + \frac{h^2}{2}\Big(\big(f_t(t_k,x(t_k)\big) + f_x\big(t_k,x(t_k)\big)\Big) f\big(t_k,x(t_k)\big)\\
&\quad + \frac{h^3}{6} x'''(\xi_k).
\end{aligned} \tag{13.45}$$

If we compare this with (13.44) we notice that the first three terms are similar. We follow the same recipe as for the Taylor methods and introduce the function

$$\Phi(t,x,h) = f(t,x) + \frac{h}{2}\big(f_t(t,x) + f_x(t,x) f(t,x)\big). \tag{13.46}$$

The equations (13.44) and (13.45) can then be written as

$$x_{k+1} = x_k + h\Phi(t_k, x_k, h) + \frac{h^3}{8}(f_{tt} + 2f_{tx}f + f_{xx}f^2) + O(h^4), \tag{13.47}$$

$$x(t_{k+1}) = x(t_k) + h\Phi\big(t_k, x(t_k), h\big) + \frac{h^3}{6}x'''(\xi_k). \tag{13.48}$$

We subtract (13.47) from (13.48) and obtain

$$x(t_{k+1}) - x_{k+1} = x(t_k) - x_k + h\Big(\Phi\big(t_k, x(t_k), h\big) - \Phi(t_k, x_k, h)\Big) + O(h^3), \tag{13.49}$$

where all the terms of degree higher than 2 have been collected together in the $O(h^3)$ term.

Theorem 13.26. *The global error $\epsilon_k = x(t_k) - x_k$ in Euler's midpoint method is advanced from step k to step $k+1$ by the relation*

$$\epsilon_{k+1} = \epsilon_k + h\Big(\Phi\big(t_k, x(t_k), h\big) - \Phi(t_k, x_k, h)\Big) + O(h^3), \tag{13.50}$$

where Φ is defined in equation (13.46)*. This means that $|\epsilon_k| = O(h^2)$, i.e., the global error in Euler's midpoint method is of second order, provided f, f_t, f_x and f_{xx} are all continuous and bounded on a set $[\alpha, \beta] \times \mathbb{R}$ such that $\alpha < a < b < \delta$.*

Proof. The relation (13.50) is completely analogous to relation (13.38) for the Taylor methods. We can therefore proceed in the same way and end up with an inequality like (13.40),

$$|\epsilon_{k+1}| \leq (1 + hC)|\epsilon_k| + Dh^3.$$

As for Euler's method and the Taylor methods we lose one order of approximation when we account for error accumulation which means that $|\epsilon_k| = O(h^2)$. ∎

Theorem 13.26 shows that Euler's midpoint method is of second order, just like the second order Taylor method, but without the need for explicit formulas for the derivatives of f. Instead the midpoint method uses an extra evaluation of f halfway between t_k and t_{k+1}. The derivation of the error formula (13.49) illustrates why the method works; the formula (13.42), which is equivalent to (13.44), reproduces the first three terms of the Taylor expansion (13.45). We therefore see that the accuracy of the Taylor methods may be mimicked by performing extra evaluations of f between t_k and t_{k+1}. The Runge-Kutta methods achieve the same accuracy as higher order Taylor methods in this way.

In our error analysis we did not compare the $O(h^3)$ terms in (13.44) and (13.45), so one may wonder if perhaps these match as well? Lemma 13.18 gives an expression for x''' in terms of f and its derivatives and we see straightaway that (13.44) only matches some of these terms.

13.6.3 Runge-Kutta methods

Runge-Kutta methods is a family of methods that generalise the midpoint Euler method. The methods use several evaluations of f between each step in a clever way which leads to higher accuracy.

In the simplest Runge-Kutta methods, the new value x_{k+1} is computed from x_k with the formula

$$x_{k+1} = x_k + h\big(\lambda_1 f(t_k, x_k) + \lambda_2 f(t_k + r_1 h, x_k + r_2 h f(t_k, x_k)\big), \tag{13.51}$$

where λ_1, λ_2, r_1, and r_2 are constants to be determined. The idea is to choose the constants in such a way that (13.51) mimics a Taylor method of the highest possible order. This can be done by following the recipe that was used in the analysis of Euler's midpoint method: Replace the outer function evaluation in (13.51) by a Taylor polynomial and choose the constants such that this matches as many terms as possible in the Taylor polynomial of $x(t)$ about $x = t_k$, see (13.44) and (13.45). It turns out that the first three terms in the Taylor expansion can be matched. This leaves one parameter free (we choose this to be $\lambda = \lambda_2$), and determines the other three in terms of λ,

$$\lambda_1 = 1 - \lambda, \quad \lambda_2 = \lambda, \quad r_1 = r_2 = \frac{1}{2\lambda}.$$

This determines a whole family of second order accurate methods.

Theorem 13.27 (Second order Runge-Kutta methods)**.** *Let the differential equation $x' = f(t,x)$ with initial condition $x(a) = x_0$ be given. Then the numerical method which advances from (t_k, x_k) to $(t_{k+1}, x_{k+1}$ according to the formula*

$$x_{k+1} = x_k + h\left((1-\lambda) f(t_k, x_k) + \lambda f\left(t_k + \frac{h}{2\lambda}, x_k + \frac{h f(t_k, x_k)}{2\lambda}\right)\right), \tag{13.52}$$

is second order accurate for any nonzero value of the parameter λ, provided f, f_t, f_x and f_{xx} are continuous and bounded in a set $[\alpha, \beta] \times \mathbb{R}$ with $\alpha < a < b < \beta$.

The proof is completely analogous to the argument used to establish the convergence rate of Euler's midpoint method. In fact, Euler's midpoint method corresponds to the particular second order Runge-Kutta method with $\lambda = 1$. Another commonly used special case is $\lambda = 1/2$. This results in the iteration formula

$$x_{k+1} = x_k + \frac{h}{2}\Big(f(t_k, x_k) + f\big((t_k, x_k + h(t_k, x_k)\big)\Big),$$

which is often referred to as *Heun's method* or the improved Euler's method. Note also that the original Euler's may be considered as the special case $\lambda = 0$, but then the accuracy drops to first order.

It is possible to devise methods that reproduce higher degree polynomials at the cost of more intermediate evaluations of f. The derivation is analogous to the procedure used for the second order Runge-Kutta method, but more involved because the degree of the Taylor polynomials are higher. One member of the family of fourth order methods is particularly popular.

Theorem 13.28 (Fourth order Runge-Kutta method)**.** *Suppose the differential equation $x' = f(t,x)$ with initial condition $x(a) = x_0$ is given. The numerical method given by the formulas*

$$\left.\begin{aligned} k_0 &= f(t_k, x_k), \\ k_1 &= f(t_k + h/2, x_k + hk_0/2), \\ k_2 &= f(t_k + h/2, x_k + hk_1/2), \\ k_3 &= f(t_k + h, x_k + hk_2), \\ x_{k+1} &= x_k + \frac{h}{6}(k_0 + 2k_1 + 2k_2 + k_3), \end{aligned}\right\} \quad k = 0, 1, \dots, n$$

is fourth order accurate provided the derivatives of f up to order four are continuous and bounded in the set $[\alpha, \beta] \times \mathbb{R}$ with $a < \alpha < \beta < b$.

It can be shown that Runge-Kutta methods which use p evaluations pr. step are pth order accurate for $p = 1, 2, 3$, and 4. However, it turns out that 6 evaluations pr. step are necessary to get a method of order 5. This is one of the reasons for the popularity of the fourth order Runge-Kutta methods—they give the most orders of accuracy pr. evaluation.

13.6.4 Multi-step methods

The methods we have discussed so far are all called *one-step methods* since they advance the solution by just using information about one previous step. This is in contrast to an *order m multi-step method* which computes x_{k+1} based on x_k,

$x_{k-1}, \ldots, x_{k+1-m}$. The methods are based on integrating the differential equation,

$$x(t_{k+1}) = x(t_k) + \int_{t_k}^{t_{k+1}} x'(t)\,dt = x(t_k) + \int_{t_k}^{t_{k+1}} f\bigl(t, x(t)\bigr)\,dt.$$

The idea is to replace $x'(t)$ with a polynomial that interpolates $x'(t) = f(t,x)$ at previously computed points. More specifically, suppose that the m approximate values $x_{k-(m-1)}, \ldots, x_k$ have been computed. We then determine the polynomial $p_m(t)$ of degree $m-1$ such that

$$p_m(t_{k-i}) = f_{k-i} = f(t_{k-i}, x_{k-i}), \quad i = 0, 1, \ldots, m-1 \tag{13.53}$$

and compute x_{k+1} by integrating p_m instead of $f\bigl(t, x(t)\bigr)$,

$$x_{k+1} = x_k + \int_{t_k}^{t_{k+1}} p_m(t)\,dt.$$

Recall from chapter 9 that the interpolating polynomial may be written as

$$p_m(t) = \sum_{i=0}^{m-1} f_{k-i}\ell_{k-i}(t) \tag{13.54}$$

where ℓ_{k-i} is the polynomial of degree $m-1$ that has the value 1 at t_{k-i} and is 0 at all the other interpolation points $t_{k-m+1}, \ldots, t_{k-i-1}, t_{k-i+1}, \ldots, t_k$. We integrate (13.54) and obtain

$$\int_{t_k}^{t_{k+1}} p_m(t)\,dt = \sum_{i=0}^{m-1} f_{k-i} \int_{t_k}^{t_{k+1}} \ell_{k-i}(t)\,dt = h\sum_{i=0}^{m-1} c_{k-i}\, f_{k-i}$$

where

$$c_{k-i} = \frac{1}{h}\int_{t_k}^{t_{k+1}} \ell_{k-i}(x)\,dt.$$

The division by h has the effect that the coefficients are independent of h. The final formula for solving the differential equation becomes

$$x_{k+1} = x_k + h\sum_{i=0}^{m-1} c_{k-i} f_{k-i}. \tag{13.55}$$

The advantage of multi-step methods is that they achieve high accuracy but just require one new evaluation of f each time the solution is advanced one step. However, multi-step methods must be supplemented with alternative methods with the same accuracy during the first iterations as there are then not sufficiently many previously computed values.

13.6.5 Implicit methods

All the methods we have considered so far advance the solution via a formula which is based on computing an approximation at a new time step from approximations computed at previous time steps. This is not strictly necessary though. The simplest example is the *backward Euler method* given by

$$x_{k+1} = x_k + hf(t_{k+1}, x_{k+1}), \quad k = 1, \ldots, n. \tag{13.56}$$

Note that the value x_{k+1} to be computed appears on both sides of the equation which means that in general we are left with a nonlinear equation or *implicit equation* for x_{k+1}. To determine x_{k+1} this equation must be solved by some nonlinear equation solver like Newton's method.

Example 13.29. Suppose that $f(t,x) = t + \sin x$. In this case the implicit equation (13.56) becomes

$$x_{k+1} = x_k + h(t_{k+1} + \sin x_{k+1})$$

which can only be solved by a numerical method.

Another example is $x' = 1/(t+x)$. Then (13.56) becomes

$$x_{k+1} = x_k + \frac{h}{t_{k+1} + x_{k+1}}.$$

In this case we obtain a quadratic equation for x_{k+1},

$$x_{k+1}^2 - (x_k - t_{k+1})x_{k+1} - t_{k+1}x_k - h = 0.$$

This can be solved with some nonlinear equation solver or the standard formula for quadratic equations. ∎

The idea of including x_{k+1} in the estimate of itself can be used for a variety of methods. An alternative midpoint method is given by

$$x_{k+1} = x_k + \frac{h}{2}\big(f(t_k, x_k) + f(t_{k+1}, x_{k+1})\big),$$

and more generally x_{k+1} can be included on the right-hand side to yield implicit Runge-Kutta like methods. A very important class of methods is implicit multi-step methods where the degree of the interpolating polynomial in (13.53) is increased by one and the next point $\big(t_{k+1}, f(t_{k+1}, x_{k+1})\big)$ is included as an interpolation point. More specifically, if the interpolating polynomial is q_m, the interpolation conditions are taken to be

$$q_m(t_{k+1-i}) = f_{k+1-i} = f(t_{k+1-i}, x_{k+1-i}), \quad i = 0, 1, \ldots, m.$$

At t_{k+1} , the value x_{k+1} is unknown so the equation (13.55) is replaced by

$$x_{k+1} = x_k + hc_{k+1} f(t_{k+1}, x_{k+1}) + h \sum_{i=0}^{m-1} c_{k-i} f_{k-i} \tag{13.57}$$

where $(c_{k+1-i})_{i=0}^{m}$ are coefficients that can be computed in the same way as indicated in section 13.6.4 (they will be different though, since the degree is different). This shows clearly the implicit nature of the method. The same idea can be used to adjust most other explicit methods as well.

It turns out that an implicit method has quite superior convergence properties and can achieve a certain accuracy with considerably larger time steps than a comparable explicit method. The obvious disadvantage of implicit methods is the need to solve a nonlinear equation at each time step. However, this disadvantage is not as bad as it may seem. Consider for example equation (13.57). If in some way we can guess a first approximation x_{k+1}^0 for x_{k+1}, we can insert this on the right and compute a hopefully better approximation x_{k+1}^1 as

$$x_{k+1}^1 = x_k + hc_{k+1} f(t_{k+1}, x_{k+1}^0) + h \sum_{i=0}^{m-1} c_{k-i} f_{k-i}.$$

But now we can compute a new approximation to x_{k+1}^2 by inserting x_{k+1}^1 on the right, and in this way we obtain a tailor-made numerical method for solving (13.57).

In practice the first approximation x_{k+1}^0 is obtained via some explicit numerical method like a suitable multi-step method. This is often referred to as a *predictor* and the formula (13.57) as the *corrector*; the combination is referred to as a *predictor-corrector method.* In many situations it turns out that it is sufficient to just use the corrector formula once.

13.7 Stability

An important input parameter for a differential equation, and therefore for any numerical method for finding an approximate solution, is the value x_0 that enters into the initial condition $x(a) = x_0$. In general, this is a real number that cannot be be represented exactly by floating point numbers, so an initial condition like $x_\epsilon(a) = x_0 + \epsilon$ will be used instead, where ϵ is some small number. This will obviously influence the computations in some way; the question is by how much?

13.7.1 Stability of a differential equation

Initially, we just focus on the differential equation and ignore effects introduced by particular numerical methods. A simple example will illustrate what can happen.

Example 13.30. Consider the differential equation

$$x' = \lambda\bigl(x - \sqrt{2}\bigr), \quad x(0) = \sqrt{2}$$

for t in some interval $[0, b]$ where $b > 0$. It is quite easy to see that the exact solution of this equation is

$$x(t) = \sqrt{2}. \tag{13.58}$$

On the other hand, if the initial condition is changed to $x_\epsilon(0) = \sqrt{2} + \epsilon$, where ϵ is a small number, the solution becomes

$$x_\epsilon(t) = \sqrt{2} + \epsilon e^{\lambda t}. \tag{13.59}$$

If we try to solve the equation numerically using floating point numbers, and commit no errors other than replacing $\sqrt{2}$ by the nearest floating point number, we end up with the solution given by (13.59) rather than the correct solution (13.58).

For small values of t, the solution given by (13.59) will be a good approximation to the correct solution. However, if $\lambda > 0$, the function $e^{\lambda t}$ grows very quickly with t even for moderate values of λ, so the second term in (13.59) will dominate. If for example $\lambda = 2$, then $e^{\lambda t} \approx 5 \times 10^{21}$ already for $t = 25$. If we use 64 bit floating point numbers we will have $\epsilon \approx 10^{-17}$ and therefore

$$x_\epsilon(25) \approx \sqrt{2} + 10^{-17} \ \times 5 \times 10^{21} \approx 5 \times 10^{4}$$

which is way off the correct value.

This kind of error is unavoidable whichever numerical method we choose to use. In practice we will also commit other errors which complicates matters further. ∎

Example 13.30 shows that it is possible for a simple differential equation to be highly sensitive to perturbations of the initial value. As we know, different initial values pick different solutions from the total family of solution curves. Therefore, if the solution curves separate more and more when t increases, the equation will be sensitive to perturbations of the initial values, whereas if the solution curves come closer together, the equation will not be sensitive to perturbations. This phenomenon is called *stability* of the equation, and it can be shown that stability can be measured by the size of the derivative of $f(t, x)$ with respect to x.

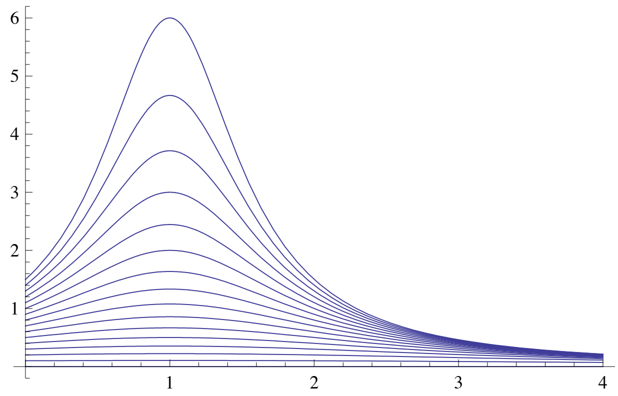


Figure 13.6. Solutions of the equation $x' = (1-t)x$ with initial values $x(0) = i/10$ for $i = 0, \ldots, 15$.

Definition 13.31 (Stability of differential equation)**.** *The differential equation $x' = f(t,x)$ is said to be stable in an area where the derivative $f_x(t,x)$ is negative (the solution curves approach each other), while it is said to be unstable (the solution curves separate) if $f_x(t,x) > 0$. Here $f_x(t,x)$ denotes the derivative of f with respect to x.*

If we return to example 13.30, we see that the equation considered there is stable when $\lambda < 0$ and unstable when $\lambda > 0$. For us the main reason for studying stability is to understand why the computed solution of some equations may blow up and become completely wrong, like the one in (13.59). However, even if $\lambda > 0$, the instability will not be visible for small t. This is true in general: It takes some time for instability to develop, and for many unstable equations, the instability may not be very strong. Likewise, there may be equations where f_x is negative in certain areas, but very close to 0, which means that the effect of dampening the errors is not very pronounced. For this reason it often makes more sense to talk about *weakly unstable* or *weakly stable* equations.

Example 13.32. Consider the differential equation

$$y' = f(t,x) = (1-t)x^2.$$

The solutions of this equation with initial values $x(0) = i/10$ for $i = 0, 1, \ldots, 15$ are shown in figure 13.6. If we differentiate f with respect to x we obtain

$$f_x(t,x) = (1-t)x.$$

This equation is therefore unstable for $t < 1$ and stable for $t > 1$ which corresponds well with the plots in figure 13.6. However, the stability effects visible in the plot are not extreme and it makes more sense to call this weak stability and instability. ∎

13.7.2 Stability of Euler's method

Stability is concerned with a differential equation's sensitivity to perturbations of the initial condition. Once we use a numerical method to solve the differential equation, there are additional factors that can lead to similar behaviour. The perturbation of the initial condition is usually insignificant compared to the error we commit when we step from one solution point to another with some approximate formula; and these errors accumulate as we step over the total solution interval $[a,b]$. The effect of all these errors is that we keep jumping from one solution curve to another, so we must expect the stability of the differential equation to be amplified further by a numerical method. It also seems inevitable that different numerical methods will behave differently with respect to stability since they use different approximations. However, no method can avoid instabilities in the differential equation itself.

We will only consider stability for Euler's method. The crucial relation is (13.15) which relates the global error $\epsilon_k = x(t_k) - x_k$ to the corresponding error at time t_{k+1}. The important question is whether the error is magnified or not as it is filtered through many time steps.

We rewrite (13.15) as

$$\epsilon_{k+1} = \bigl(1 + hL_k\bigr)\epsilon_k + G_k,$$

with $L_k = f_x(t_k,\theta_k)$ with θ_k some real number between x_k and $x(t_k)$, and $G_k = h^2 x''(\xi_k)/2$ with ξ_k some number between t_k and t_{k+1}. The decisive part of this relation is the factor that multiplies ϵ_k. We see this quite clearly if we unwrap the error as in (13.16),

$$\begin{aligned}
\epsilon_{k+1} &= (1 + hL_k)\epsilon_k + G_k\\
&= (1 + hL_k)((1 + hL_{k-1})\epsilon_{k-1} + G_{k-1}) + G_k\\
&= \prod_{j=k-1}^{k} (1 + hL_j)\epsilon_{k-1} + (1 + hL_k)G_{k-1} + G_k\\
&= \prod_{j=k-1}^{k} (1 + hL_j)((1 + hL_{k-2})\epsilon_{k-2} + G_{k-2}) + (1 + hL_k)G_{k-1} + G_k\\
&= \prod_{j=k-2}^{k} (1 + hL_j)\epsilon_{k-2} + \prod_{j=k-1}^{k} (1 + hL_j)G_{k-2} + (1 + hL_k)G_{k-1} + G_k
\end{aligned}$$

$$= \prod_{j=k-2}^{k} (1 + hL_j)\epsilon_{k-2} + \sum_{i=k-2}^{k} \prod_{j=i+1}^{k} (1 + hL_j) G_i$$

$$\vdots$$

$$= \prod_{j=0}^{k} (1 + hL_j)\epsilon_0 + \sum_{i=0}^{k} \prod_{j=i+1}^{k} (1 + hL_j) G_i.$$

This is a bit more complicated than what we did in (13.16) since we have taken neither absolute nor maximum values, as this would hide some of the information we are looking for. On the other hand, we have an identity which deserves to be recorded in a lemma.

Lemma 13.33. *The error in Euler's method for the equation $x' = f(t,x)$ at time t_{k+1} is given by*

$$\epsilon_{k+1} = \prod_{j=0}^{k} (1 + hL_j)\epsilon_0 + \sum_{i=0}^{k} \prod_{j=i+1}^{k} (1 + hL_j) G_i, \tag{13.60}$$

for $k = 0, \ldots, n-1$. Here $L_j = f_x(t_j, \theta_j)$ where θ_j is a number between x_j and $x(t_j)$, while $G_j = h^2 x''(\xi_j)/2$ with ξ_j some number between t_k and t_{k+1}. The number ϵ_0 is the error committed in implementing the initial condition $x(a) = x_0$.

We can now more easily judge what can go wrong with Euler's method. We see that the error in the initial condition, ϵ_0, is magnified by the factor $\prod_{j=0}^{k}(1 + hL_j)$. If each of the terms in this product has absolute value larger than 1, this magnification factor can become very large, similarly to what we saw in example 13.30. Equation 13.60 also shows that similar factors multiply the truncation errors (the remainders in the Taylor expansions) which are usually much larger than ϵ_0, so these may also be magnified in the same way.

What does this say about stability in Euler's method? If $|1 + hL_j| > 1$, i.e., if either $hL_j > 0$ or $hL_j < -2$ for all j, then the errors will be amplified and Euler's method will be unstable. On the other hand, if $|1 + hL_j| < 1$, i.e., if $-2 < hL_j < 0$, then the factors will dampen the error sources and Eulers's method will be stable.

This motivates a definition of stability for Euler's method.

Definition 13.34. *Euler's method for the equation*

$$x' = f(t,x), \quad x(a) = x_0,$$

is said to be stable in an area where $|1 + hf_x(t,x)| > 1$ *and unstable in an area where* $|1 + hf_x(t,x)| < 1$.

We observe that Euler's method has no chance of being stable if the differential equation is unstable, i.e., if $f_x > 0$. On the other hand, it is not necessarily stable even if the differential equation is stable, that is if $f_x < 0$; we must then avoid $1 + hf_x$ becoming smaller than -1. This means that we must choose h so small that $-1 - hf_x(t,x) > -1$ or

$$h < \frac{2}{|f_x(t,x)|}$$

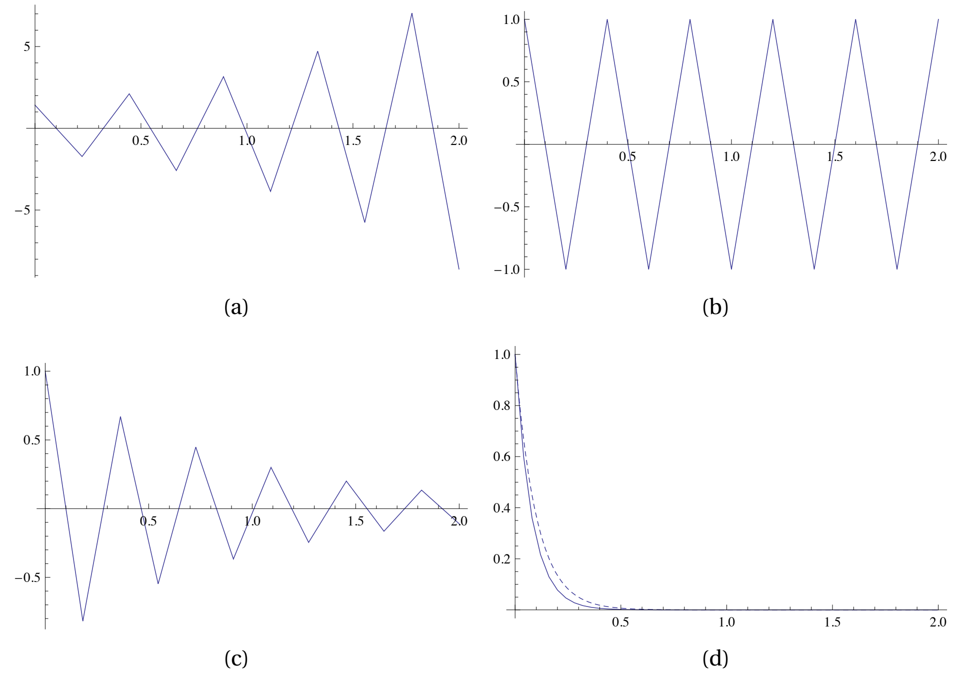
for all t and x. Otherwise, Euler's method will be unstable, although in many cases it is more correct to talk about *weak stability* or *weak instability*, just like for differential equations.

Example 13.35. The prototype of a stable differential equation is

$$x' = -\lambda x, \quad x(0) = 1$$

with the exact solution $x(t) = e^{-\lambda t}$ which approaches 0 quickly with increasing t. We will try and solve this with Euler's method when $\lambda = -10$, on the interval $[0,2]$. In this case $f_x(t,x) = -10$ so the stability estimate demands that $|1 - 10h| < 1$ or $h < 1/5 = 0.2$. To avoid instability in Euler's method we therefore need to use at least 11 time steps on the interval $[0,2]$. Figure 13.7 illustrates how Euler's method behaves. In figure (a) we have used a step length of $2/9$ which is just above the requirement for stability and we notice how the size of the computed solution grows with t, a clear sign of instability. In figure (b), the step length is chosen so that $|1 + hf_x(t,x)| = 1$ and we see that the computed solution does not converge to 0 as it should, but neither is the error amplified. In figure (c), the step length is just below the limit and the solution decreases, but rather slowly. Finally, in figure (d), the step size is well below the limit and the numerical solution behaves as it should. ∎

Stability of other numerical methods can be studied in much the same way as for Euler's method. It is essentially the derivative of the function Φ (see (13.36) and (13.46)) with respect to x that decides the stability of a given method. This is a vast area and the reader is referred to advanced books on numerical analysis to learn more.



(a) (b) (c) (d)

Figure 13.7. The plots illustrate the result of using Euler's method for solving the equation $x' = -10x$ with initial condition $x(0) = 1$. In (a) a step length of 2/9 was used, in (b) the step length was 2/10, and in (c) it was 2/11. The dashed curve in (d) shows the exact solution $x(t) = e^{-10t}$ while the solid curve shows the result produced by Euler's method with a step length of 1/25.

13.8 Systems of differential equations

So far we have focused on how to solve a single first order differential equation. In practice two or more such equations, coupled together, are necessary to model a problem, and perhaps even equations of higher order. In this section we are going to see how the methods we have developed above can easily be adapted to deal with both systems of equations and equations of higher order.

13.8.1 Vector notation and existence of solution

Many practical problems involve not one, but two or more differential equations. For example many processes evolve in three dimensional space, with separate differential equations in each space dimension.

Example 13.36. At the beginning of this chapter we saw that a vertically falling object subject to gravitation and friction can be modelled by the differential equation

$$v' = g - \frac{c}{m} v^2, \tag{13.61}$$

where $v = v(t)$ is the speed at time t. How can an object that also has a horizontal speed be modelled? A classical example is that of throwing a ball. In the vertical direction, equation (13.61) is still valid, but since the y-axis points upwards, we change signs on the right-hand side and label the speed by a subscript 2 to indicate that this is movement along the y- (the second) axis,

$$v_2' = \frac{c}{m} v_2^2 - g.$$

In the x-direction a similar relation holds, except there is no gravity. If we assume that the positive x-axis is in the direction of the movement we therefore have

$$v_1' = -\frac{c}{m} v_1^2.$$

In total we have

$$v_1' = -\frac{c}{m} v_1^2, \qquad v_1(0) = v_{0_x}, \tag{13.62}$$

$$v_2' = \frac{c}{m} v_2^2 - g, \quad v_2(0) = v_{0_y}, \tag{13.63}$$

where v_{0_x} is the initial speed of the object in the x-direction and v_{0_y} is the initial speed of the object in the y-direction. If we introduce the vectors $\boldsymbol{v} = (v_1, v_2)$ and $\boldsymbol{f} = (f_1, f_2)$ where

$$\begin{aligned} f_1(t, \boldsymbol{v}) &= f_1(t, v_1, v_2) = -\frac{c}{m} v_1^2, \\ f_2(t, \boldsymbol{v}) &= f_2(t, v_1, v_2) = \frac{c}{m} v_2^2 - g, \end{aligned}$$

and the initial vector $\boldsymbol{v}_0 = (v_{0_x}, v_{0_y})$, the equations (13.62)–(13.63) may be rewritten more compactly as

$$\boldsymbol{v}' = \boldsymbol{f}(t, \boldsymbol{v}), \quad \boldsymbol{v}(0) = \boldsymbol{v}_0.$$

Apart from the vector symbols, this is exactly the same equation as we have studied throughout this chapter. ∎

The equations in example 13.36 are quite specialised in that the time variable does not appear on the right, and the two equations are independent of each other. The next example is more general.

Example 13.37. Consider the three equations with initial conditions

$$x' = xy + \cos z, \quad x(0) = x_0, \tag{13.64}$$

$$y' = 2 - t^2 + z^2 y, \quad x(0) = y_0, \tag{13.65}$$

$$z' = \sin t - x + y, \quad z(0) = z_0. \tag{13.66}$$

If we introduce the vectors $\boldsymbol{x} = (x, y, z)$, $\boldsymbol{x}_0 = (x_0, y_0, z_0)$, and the vector of functions $\boldsymbol{f}(t, \boldsymbol{x}) = \big(f_1(t, \boldsymbol{x}), f_2(t, \boldsymbol{x}), f_3(t, \boldsymbol{x})\big)$ defined by

$$\begin{aligned}
x' &= f_1(t, \boldsymbol{x}) = f_1(t, x, y, z) = xy + \cos z, \\
y' &= f_2(t, \boldsymbol{x}) = f_2(t, x, y, z) = 2 - t^2 + z^2 y, \\
z' &= f_3(t, \boldsymbol{x}) = f_3(t, x, y, z) = \sin t - x + y,
\end{aligned}$$

we can write (13.64)–(13.66) simply as

$$\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x}), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0. \quad \blacksquare$$

Examples 13.36–13.37 illustrate how vector notation may camouflage a system of differential equations as a single equation. This is helpful since it makes it quite obvious how the theory for scalar equations can be applied to systems of equations. Let us first be precise about what we mean with a system of differential equations.

Definition 13.38. *A system of M first order differential equations in M unknowns with corresponding initial conditions is given by a vector relation on the form*

$$\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x}), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0. \tag{13.67}$$

Here $\boldsymbol{x} = \boldsymbol{x}(t) = \big(x_1(t), \ldots, \boldsymbol{x}_M(t)\big)$ is a vector of M unknown scalar functions and $\boldsymbol{f}(t, \boldsymbol{x}) : \mathbb{R}^{M+1} \to \mathbb{R}^M$ is a vector function of the $M+1$ variables t and $\boldsymbol{x} = (x_1, \ldots, x_M)$, i.e.,

$$\boldsymbol{f}(t, \boldsymbol{x}) = \big(f_1(t, \boldsymbol{x}), \ldots, f_M(t, \boldsymbol{x})\big).$$

The notation $\boldsymbol{x}'$ denotes the vector of derivatives of the components of $\boldsymbol{x}$ with respect to t,

$$\boldsymbol{x}' = \boldsymbol{x}'(t) = \big(x_1'(t), \ldots, x_M'(t)\big).$$

It may be helpful to write out the vector equation (13.67) in detail,

$$\begin{aligned}
x_1' &= f_1(t, \boldsymbol{x}) = f_1(t, x_1, \ldots, x_M), \qquad x_1(0) = x_{1,0} \\
&\vdots \\
x_M' &= f_M(t, \boldsymbol{x}) = f_M(t, x_1, \ldots, x_M), \quad x_M(0) = x_{M,0}.
\end{aligned}$$

We see that both the examples above fit into this setting, with $M=2$ for example 13.36 and $M=3$ for example 13.37.

Before we start considering numerical solutions of systems of differential equations, we need to know that solutions exist. This is settled by the following theorem which is completely analogous to theorem 13.6.

Theorem 13.39. *Suppose that the function $\boldsymbol{f}(t,\boldsymbol{x})$ and its first derivatives with respect to the components of $\boldsymbol{x}$ are continuous on the set*

$$\mathbb{A} = [\alpha,\beta] \times [\gamma_1,\delta_1] \times [\gamma_2,\delta_2] \times \cdots \times [\gamma_M,\delta_M]. \tag{13.68}$$

If the point $(a,\boldsymbol{x}_0)$ lies in the interior of $\mathbb{A}$ there exists an $\tau>0$ such that the differential equation

$$\boldsymbol{x}' = \boldsymbol{f}(t,\boldsymbol{x}), \quad \boldsymbol{x}(a) = \boldsymbol{x}_0 \tag{13.69}$$

has a unique solution on the interval $[a-\tau,a+\tau]$ which is contained in $[\alpha,\beta]$.

The set $\mathbb{A}$ defined in (13.68) may seem mysterious. It is just the collection of all points $(t,x_1,\ldots,x_M)$ such that $t \in [\alpha,\beta]$ and $x_i \in [\gamma_i,\delta_i]$ for $i=1,\ldots,M$. The precise derivatives of the components of $\boldsymbol{f}$ that are required to be continuous are

$$\begin{array}{cccc}
\dfrac{\partial f_1}{\partial x_1}, & \dfrac{\partial f_1}{\partial x_2}, & \cdots & \dfrac{\partial f_1}{\partial x_M}, \\
\dfrac{\partial f_2}{\partial x_1}, & \dfrac{\partial f_2}{\partial x_2}, & \cdots & \dfrac{\partial f_2}{\partial x_M}, \\
\vdots & \vdots & \ddots & \vdots \\
\dfrac{\partial f_M}{\partial x_1}, & \dfrac{\partial f_M}{\partial x_2}, & \cdots & \dfrac{\partial f_M}{\partial x_M}.
\end{array}$$

This may all seem complicated, but the conclusion is rather simple: the system of equations (13.69) has a solution near the initial value $(t,\boldsymbol{x}_0)$ provided all the component functions are reasonably well behaved near the point.

13.8.2 Numerical methods for systems of first order equations

There are very few analytic methods for solving systems of differential equations, so numerical methods are essential. It turns out that most of the methods for a single equation generalise to systems. A simple example illustrates the general principle

Example 13.40 (Euler's method for a system)**.** We consider the equations in example 13.37,

$$\boldsymbol{x}' = \boldsymbol{f}(t,\boldsymbol{x}), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0,$$

where

$$\begin{aligned} \boldsymbol{f}(t,\boldsymbol{x}) &= \big(f_1(t,x_1,x_2,x_3), f_2(t,x_1,x_2,x_3), f_3(t,x_1,x_2,x_3)\big) \\ &= (x_1x_2+\cos x_3, 2-t^2+x_3^2x_2, \sin t - x_1 + x_2). \end{aligned}$$

Euler's method is easily generalised to vector equations as

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + h\boldsymbol{f}(t_k,\boldsymbol{x}_k), \quad k=0,\,1,\,\ldots,\,n-1. \tag{13.70}$$

If we write out the three components explicitly, this becomes

$$\left.\begin{aligned} x_1^{k+1} &= x_1^k + hf_1(t_k,x_1^k,x_2^k,x_3^k) = x_1^k + h\big(x_1^kx_2^k + \cos x_3^k\big), \\ x_2^{k+1} &= x_2^k + hf_2(t_k,x_1^k,x_2^k,x_3^k) = x_2^k + h\big(2-t_k^2+(x_3^k)^2x_2^k\big), \\ x_3^{k+1} &= x_3^k + hf_3(t_k,x_1^k,x_2^k,x_3^k) = x_3^k + h\big(\sin t_k - x_1^k + x_2^k\big), \end{aligned}\right\} \tag{13.71}$$

for $k = 0,\,1,\,\ldots,\,n-1$, with the starting values (x_1^0, x_2^0, x_3^0) given by the initial condition. Although they look rather complicated, these formulas can be programmed quite easily. The trick is to make use of the vector notation in (13.70), since it nicely hides the details in (13.71). ∎

Example 13.40 illustrates Euler's method for a system of equations, and for most of the numerical methods we have encountered earlier in the chapter it is equally straightforward to generalise to systems of equations.

Observation 13.41 (Generalisation to systems)**.** *Euler's method, Euler's midpoint method, and the Runge-Kutta methods all generalise naturally to systems of differential equations.*

For example the formula for advancing one time step with Euler's midpoint method becomes

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + h\boldsymbol{f}\big(t_k + h/2, \boldsymbol{x}_k + h\boldsymbol{f}(t_k,\boldsymbol{x}_k)/2\big),$$

while the fourth order Runge-Kutta method becomes

$$\begin{aligned} \boldsymbol{k}_0 &= \boldsymbol{f}(t_k,\boldsymbol{x}_k), \\ \boldsymbol{k}_1 &= \boldsymbol{f}(t_k+h/2, \boldsymbol{x}_k + h\boldsymbol{k}_0/2), \\ \boldsymbol{k}_2 &= \boldsymbol{f}(t_k+h/2, \boldsymbol{x}_k + h\boldsymbol{k}_1/2), \\ \boldsymbol{k}_3 &= \boldsymbol{f}(t_k+h, \boldsymbol{x}_k + h\boldsymbol{k}_2), \\ \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \frac{h}{6}(\boldsymbol{k}_0 + \boldsymbol{k}_1 + \boldsymbol{k}_2 + \boldsymbol{k}_3). \end{aligned}$$

Systems of differential equations is an example where the general mathematical formulation is simpler than most concrete examples. In fact, if each component of these formulas are written out in detail, the details quickly become overwhelming, so it is important to stick with the vector notation. This also applies to implementation in a program: It is wise to use the vector formalism and mimic the mathematical formulation as closely as possible.

In principle the Taylor methods also generalise to systems of equations, but because of the need for manual differentiation of each component equation, the details swell up even more than for the other methods. Multi-step methods generalise nicely to systems as well. Implicit methods are not so easy to generalise though, since nonlinear solutions methods are considerably more complicated for systems than for scalar equations. However, a predictor-corrector approach works quite well for systems.

13.8.3 Higher order equations as systems of first order equations

Many practical modelling problems lead to systems of differential equations, and sometimes higher order equations are necessary. It turns out that these can be reduced to systems of first order equations as well.

Example 13.42. Consider the second order equation

$$x'' = t^2 + \sin(x + x'), \quad x(0) = 1, \quad x'(0) = 0. \tag{13.72}$$

This equation is nonlinear and cannot be solved with any of the standard analytical methods. If we introduce the new function $x_2 = x'$, we notice that $x_2' = x''$, so the differential equation can be written

$$x_2' = t^2 + \sin(x + x_2), \quad x(0) = 1, \quad x_2(0) = 0.$$

If we also rename x as $x_1 = x$ we see that the second order equation in (13.72) can be written as the system

$$\begin{aligned} x_1' &= x_2, & x_1(0) &= 1, \qquad (13.73)\\ x_2' &= t^2 + \sin(x_1 + x_2), & x_2(0) &= 0. \qquad (13.74)\end{aligned}$$

In other words, equation (13.72) can be written as the system (13.73)–(13.74). We also see that this system can be expressed as the single equation in (13.72), so the two equations (13.73)–(13.74) and the single equation (13.72) are in fact equivalent in the sense that a solution of one automatically gives a solution of the other. ∎

The technique used in example 13.42 works in general—a pth order equation can be rewritten as a system of p first order equations.

Theorem 13.43. *The pth order differential equation*

$$x^{(p)} = g\big(t, x, x', \dots, x^{(p-1)}\big) \tag{13.75}$$

with initial conditions

$$x(a) = d_0,\ x'(a) = d_1,\ \dots,\ x^{(p-2)}(0) = d_{p-2},\ x^{(p-1)}(0) = d_{p-1} \tag{13.76}$$

is equivalent to the system of p equations in the p unknown functions x_1, x_2, $\dots$, x_p,

$$\begin{aligned} x_1' &= x_2, & x_1(a) &= d_0,\\ x_2' &= x_3, & x_2(a) &= d_1,\\ &\ \vdots \\ x_{p-1}' &= x_p, & x_{p-1}(a) &= d_{p-2},\\ x_p' &= g(t, x_1, x_2, \dots, x_{p-1}), & x_p(a) &= d_{p-1}, \end{aligned} \tag{13.77}$$

in the sense that the component solution $x_1(t)$ of (13.77) *agrees with the solution $x(t)$ of*(13.75)–(13.76).

Proof. The idea of the proof is just like in example 13.42. From the first $p-1$ relations in (13.77) we see that

$$x_2 = x_1', \quad x_3 = x_2' = x_1'', \quad \dots, \quad x_p = x_{p-1}' = x_{p-2}'' = \dots = x_1^{(p-1)}.$$

If we insert this in the last equation in (13.77) we obtain a pth order equation for x_1 that is identical to (13.75). In addition, the initial values in (13.77) translate into initial values for x_1 that are identical to (13.76) so x_1 must solve (13.75)–(13.76). Conversely, if x is a solution of (13.75)–(13.76) it is easy to see that the functions

$$x_1 = x, \quad x_2 = x', \quad x_3 = x'', \quad \dots, \quad x_{p-1} = x^{(p-2)}, \quad x_p = x^{(p-1)}$$

solve the system (13.77). ∎

Theorem 13.43 shows that if we can solve systems of differential equations we can also solve single equations of order higher than one. We can also handle systems of higher order equations in this way.

Example 13.44 (System of higher order equations)**.** Consider the system of differential equations given by

$$\begin{aligned} x'' &= t + x' + y', & x(0) &= 1, & x'(0) &= 2,\\ y''' &= x'y'' + x, & y(0) &= -1, & y'(0) &= 1, \quad y''(0) = 2. \end{aligned}$$

We introduce the new functions $x_1 = x$, $x_2 = x'$, $y_1 = y$, $y_2 = y'$, and $y_3 = y''$. Then the above system can be written as

$$\begin{aligned}
x_1' &= x_2, & x_1(0) &= 1,\\
x_2' &= t + x_2 + y_2, & x_2(0) &= 2,\\
y_1' &= y_2, & y_1(0) &= -1,\\
y_2' &= y_3, & y_2(0) &= 1,\\
y_3' &= x_2 y_3 + x_1, & y_3(0) &= 2. \quad \blacksquare
\end{aligned}$$

Example 13.44 illustrates how a system of higher order equations may be expressed as a system of first order equations. Perhaps not surprisingly, a general system of higher order equations can be converted to a system of first order equations. The main complication is in fact notation. We assume that we have r equations involving r unknown functions $x_1, \ldots, x_r$. Equation no. i expresses some derivative of x_i on the left in terms of derivatives of itself and the other functions on the right,

$$x_i^{(p_i)} = g_i\Big(t, x_1, x_1', \ldots, x_1^{(p_1-1)}, \ldots, x_r, x_r', \ldots, x_r^{(p_r-1)}\Big), \quad i = 1, \ldots, r. \tag{13.78}$$

In other words, the integer p_i denotes the derivative of x_i on the left in equation no. i, and it is assumed that in the other equations the highest derivative of x_i is $p_i - 1$ (this is not an essential restriction, see exercise 20).

To write the system (13.78) as a system of first order equations, we just follow the same strategy as in example 13.44: For each variable x_i, we introduce the p_i variables

$$x_{i,1} = x_i, \quad x_{i,2} = x_i', \quad x_{i,3} = x_i'', \quad \ldots, \quad x_{i,p_i} = x_i^{(p_i-1)}.$$

Equation no. i in (13.78) can then be replaced by the p_i first order equations

$$\begin{aligned}
x_{i,1}' &= x_{i,2},\\
x_{i,2}' &= x_{i,3},\\
&\vdots\\
x_{i,p_i-1}' &= x_{i,p_i},\\
x_{i,p_i}' &= g_i\big(t, x_{1,1}, \ldots, x_{1,p_1}, \ldots, x_{r,1}, \ldots, x_{r,p_r}\big)
\end{aligned}$$

for $i = 1, \ldots, r$. We emphasise that the general procedure is exactly the same as the one used in example 13.44, it is just that the notation becomes rather heavy in the general case.

We record the conclusion in a non-technical theorem.

Theorem 13.45. *A system of differential equations can always be written as a system of first order equations.*

13.9 Final comments

Our emphasis in this chapter has been to derive some of the best-known methods for numerical solution of first order ordinary differential equations, including a basic error analysis, and treatment of systems of equations. There are a number of additional issues we have not touched upon.

There are numerous other numerical methods in addition to the ones we have discussed here. The universal method that is optimal for all kinds of applications does not exist; you should choose the method that works best for your particular kind of application.

We have assumed that the step size h remains fixed during the solution process. This is convenient for introducing the methods, but usually too simple for solving realistic problems. A good method will use a small step size in areas where the solution changes quickly and longer step sizes in areas where the solution varies more slowly. A major challenge is therefore to detect, during the computations, how quickly the solution varies, or equivalently, how large the error is locally. If the error is large in an area, it means that the local step size needs to be reduced; it may even mean that another numerical method should be used in the area in question. This kind of monitoring of the error, coupled with local control of the step size and choice of method, is an important and challenging characteristic of modern software for solving differential equations. Methods like these are called *adaptive methods*.

We have provided a basic error analysis of the simplest methods, and this kind of analysis can be extended to a number of other methods without much change. The analysis accounts for the error committed by making use of certain mathematical approximations. In most cases this kind of error analysis is adequate, but in certain situations it may also be necessary to pay attention to the round-off error.

Exercises

13.1 Suppose we have the differential equation

$$x' = f(t,x), \quad x(b) = x_0,$$

and we seek a solution on the interval $[a,b]$ where $a < b$. Adjust Euler's method so that it works in this alternative setting where the initial value is at the right end of the interval.

13.2 Compute numerical solutions to $x(1)$ for the equations below using two steps with Euler's method, the quadratic Taylor method and the quartic Taylor method. For comparison the correct solution to 14 decimal digits is given in each case.

a) $x' = t^5 + 4, \qquad x(0) = 1,$
$x(1) = 31/6 \approx 5.166666666667.$

b) $x' = x + t, \quad x(0) = 1,$
$x(1) \approx 3.4365636569181.$

c) $x' = x + t^3 - 3(t^2 + 1) - \sin t + \cos t, \quad x(0) = 7,$
$x(1) \approx 13.714598298644.$

13.3 We are given the differential equation

$$x' = e^{-t^2}, \quad x(0) = 0.$$

Compute an estimate of $x(0.5)$ by taking one step with each of the methods below, and find an upper bound on the absolute error in each case.

a) Euler's method.

b) The quadratic Taylor method.

c) The cubic Taylor method.

13.4 Suppose we perform one step of Euler's method for the differential equation

$$x' = \sin x, \quad x(0) = 1.$$

Find an upper bound for absolute the error.

13.5 Estimate the error in exercise 2 (a) (using Euler's method). Hint: Use equation 13.15.

13.6 This exercise is based on example 13.36 in which we modelled the movement of a ball thrown through air with the equations

$$\begin{aligned} v_1' &= -\frac{c}{m} v_1^2, & v_1(0) &= v_{0_x}, \\ v_2' &= \frac{c}{m} v_2^2 - g, & v_2(0) &= v_{0_y}, \end{aligned}$$

We now consider the launch of a rocket. In this case, the constants g and c will become complicated functions of the height y, and possibly also of x. We make the (rather unrealistic) assumption that

$$\frac{c}{m} = c_0 - ay$$

where c_0 is the air resistance constant at the surface of the earth and y is the height above the earth given in kilometers. We will also use the fact that gravity varies with the height according to the formula

$$g = \frac{g_0}{(y+r)^2},$$

where g_0 is the gravitational constant times the mass of the earth, and r is the radius of the earth. Finally, we use the facts that $x' = v_1$ and $y' = v_2$.

a) Find the second order differential equation for the vertical motion (make sure that the positive direction is upwards).

b) Rewrite the differential equation for the horisontal motion as a second order differential equation that depends on x, x', y and y'.

c) Rewrite the coupled second order equations from (a) and (b) as a system of four first order differential equations.

d) Optional: Use a numerical method to find a solution at $t = 1$ hour for the initial conditions $x(0) = y(0) = 0$, $x'(0) = 200$ km/h and $y'(0) = 300$ km/h. Use $a = 1.9 * 10^{-4} \frac{\text{Nh}^2}{\text{km}^3\text{kg}}$, $g_0 = 3.98 * 10^8 \frac{(\text{km})^2\text{m}}{\text{s}^2}$ and $c_0 = 0.19 \frac{\text{Nh}^2}{\text{km}^2\text{kg}}$. These units are not so important, but mean that distances can be measured in km and speeds in km/h.

13.7 Consider the first order differential equation

$$x' = x^2, \quad x(0) = 1.$$

a) Estimate $x(1)$ by using one step with Euler's method.

b) Estimate $x(1)$ by using one step with the quadratic Taylor method.

c) Estimate $x(1)$ by using one step with Euler's midpoint method.

d) Estimate $x(1)$ by using one step with the Runge Kutta fourth order method.

e) Estimate $x(1)$ by using two steps with the Runge Kutta fourth order method.

f) Optional: Write a computer program that implements one of the above mentioned methods and use it to estimate the value of $y(1)$ with 10, 100, 1000 and 10000 steps?

g) Do the estimates seem to converge?

h) Solve the equation analytically and explain your numerical results.

13.8 Solve the differential equation

$$x' + x \sin t = \sin t$$

and plot the solution on the interval $t \in [-2\pi, 2\pi]$ for the following initial values:

a) $x(0) = 1 - e$.

b) $x(4) = 1$.

c) $x(\pi/2) = 2$.

d) $x(-\pi/2) = 3$.

13.9 Rn-222 is a common radioactive isotope. It decays to 218-Po through α-decay with a half-life of 3.82 days. The average concentration is about 150 atoms per mL of air. Radon emanates naturally from the ground, and so is typically more abundant in cellars than in a sixth floor apartment. Certain rocks like granite emanates much more radon than other substances.

In this exercise we assume that we have collected air samples from different places, and these samples have been placed in special containers so that no new Rn-222 (or any other element) may enter the sample after the sampling has been completed. We now want to measure the Rn-222 abundance as a function of time, $f(t)$.

a) The abundance $x(t)$ of Rn-222 is governed the differential equation $x' = \lambda x$. Solve the differential equation analytically and determine λ from the half-life given above.

b) Make a plot of the solution for the first 10 days for the initial conditions $x(0) = 100$, 150, 200 and 300 atoms per mL.

c) The different initial conditions give rise to a family of functions. Do any of the functions cross each other? Can you find a reason why they do/do not?

d) The four initial conditions correspond to four different air samples. Two of them were taken from two different cellars, one was taken from an upstairs bedroom, and the fourth is an average control sample. Which is which?

13.10 In this problem we are going to use Euler's method to solve the differential equation you found in exercise 9 with the inital condition $x(0) = 300$ atoms per mL sample over a time period from 0 to 6 days.

a) Use 3 time steps and make a plot where the points (t_i, x_i) for each time step are marked. What is the relative error at each point? (Compare with the exact solution.)

b) For each point computed by Euler's method, there is an exact solution curve that passes through the point. Determine these solutions and draw them in the plot you made in (a).

c) Use Euler's midpoint method with 3 time steps to find the concentration of Rn-222 in the 300 atoms per mL sample after 6 days. Compare with the exact result, and your result from exercise 10. What are the relative errors at the computed points?

d) Repeat (a), but use the quadratic Taylor method instead.

13.11 In this exercise we are going to derive the quartic (degree four) Taylor method and use it to solve the equation for radioactive decay in exercise 9.

a) Derive the quartic Taylor method.

b) Use the quartic Taylor method to find the concentration of RN-222 in the 300 atoms per mL sample after 6 days using 3 time steps and compare your results with those produced by the quadratic Taylor method in exercise 10. How much has the solution improved (in terms of absolute and relative errors)?

c) How many time steps would you have to use in the two Taylor methods to achive a relative error smaller than 10^{-5}?

d) What order would the Taylor order have to be to make sure that the relative error is smaller than 10^{-5} with only 3 steps?

13.12 Write a program that implements Euler's method for first order differential equations on the form

$$x' = f(t, x), \quad x(a) = x_0,$$

on the interval $[a, b]$, with n time steps. You may assume that the function f and the numbers a, b, x_0, and n are given. Test the program with the equation $x' = x$ and $x(0) = 1$ on the interval $[0, 1]$. Plot the exact solution $y(x) = e^x$ alongside the approximation and experiment with different values of n.

13.13 In this problem we are going to solve the equation

$$x' = f(t, x) = -x \sin t + \sin t, \quad x(0) = 2 + e,$$

numerically on the interval $[0, 2\pi]$.

a) Use Euler's method with 1, 2, 5, and 10 steps and plot the results. How does the solution evolve with the number of steps?

b) Use Euler's mid-point method with 1 and 5 steps and plot the results.

c) Compare the results from Euler's mid-point method with those form Euler's method including the number of evaluations of f in each case. Which method seems to be best?

13.14 In this exercise we are going to solve the differential equation

$$x' = f(t,x) = t^2 + x^3 - x, \quad x(0) = 1 \tag{13.79}$$

numerically with the quadratic Taylor method.

a) Find a formula for $x''(t)$ by differentiating equation 13.79.

b) Use the quadratic Taylor method and your result from *a*) to find an approximation to $x(1)$ using 1, 2 and, 5 steps. .

c) Write a computer program that implements the quadratic Taylor method and uses it to find an approximation of $x(1)$ with 10, 100 and 1000 steps.

13.15 In this exercise we are going to derive the cubic Taylor method and use it for solving equation (13.79) in exercise 14.

a) Derive a general algorithm for the cubic Taylor method.

b) Find a formula for $x'''(t)$ by differentiating equation 13.79, and find an approximation to $x(1)$ using 1 time step with the cubic Taylor method. Repeat using 2 time steps.

c) How do the results from the cubic Taylor method compare with the results from the quadratic Taylor method obtained in exercise 14?

d) Implement the cubic Taylor method in a program and compute an approximation to $x(2)$ with 10, 100 and 1000 steps.

13.16 When investigating the stability of a numerical method it is common to apply the method to the model equation

$$x' = -\lambda x, \quad x(0) = 1$$

and check for which values of the step length h the solution blows up.

a) Apply Euler's method to the model equation and determine the range of h-values that for which the solution remains bounded.

b) Repeat (a) for Euler's midpoint method.

c) Repeat (a) for the second order Taylor method.

d) Repeat (a) for the fourth order Runge-Kutte method.

13.17 Radon-222 is actually an intermediate decay product of a decay chain from Uranium-238. In this chain there are 16 subsequent decays which takes 238-U into a stable lead isotope (206-Pb). In one part of this chain 214-Pb decays through β-decay to 214-Bi which then decays through another β-decay to 214-Po. The two decays have the respective halflifes of 26.8 minutes and 19.7 minutes.

Suppose that we start with a certain amount of 214-Pb atoms and 214-Bi atoms, we want to determine the amounts of 214-Pb and 214-Bi as functions of time.

a) Phrase the problem as a system of two coupled differential equations.

b) Solve the equations from (a) analytically.

c) Suppose that the inital amounts of lead and bismuth are 600 atoms and 10 atoms respectively. Find the solutions for these initial conditions and plot the two functions for the first 1.5 hours.

d) When is the amount of bismuth at its maximum?

e) Compute the number of lead and bismuth atoms after 1 hour with Euler's method. Choose the number of steps to use yourself.

f) Repeat (e), but use the fourth order Runge-Kutta method instead and the same number of steps as in (e).

13.18 Write the following differential equations as systems of first order equations. The unknowns x, y, and z are assumed to be functions of t.

a) $x'' + t^2 x' + 3x = 0.$

b) $mx'' = -k_s x - k_d x'.$

c) $y''(t) = 2(e^{2t} - y^2)^{1/2}.$

d) $2x'' - 5x' + x = 0$ with initial conditions $x(3) = 6$, $x'(3) = -1$.

13.19 Write the following systems of differential equations as systems of first order equations. The unknowns x, y, and z are assumed to be functions of t.

a)

$$y'' = y^2 - x + e^t,$$
$$x'' = y - x^2 - e^t.$$

b)

$$x'' = 2y - 4t^2 x,$$
$$y'' = -2x - 2tx'.$$

c)

$$x'' = y''x + (y')^2 x,$$
$$y'' = -y.$$

d)

$$x''' = y''x^2 - 3(y')^2 x,$$
$$y'' = t + x'.$$

13.20 Write the system

$$x'' = t + x + y',$$
$$y''' = x''' + y'',$$

as a system of 5 first order equations. Note that this system is not on the form (13.78) since x''' appears on the right in the second equation. Hint: You may need to differentiate one of the equations.

13.21 Solve the system

$$x'' = 2y - 4t^2 x, \quad x(0) = 1,$$
$$y'' = -2x - 2tx', \quad y(0) = 0,$$

numerically on the interval $[0,2]$. Try both Euler's method and Euler's mid-point method with two time steps and plot the results.

13.22 A block of mass m is attached to a horizontal spring. As long as the displacement x (measured in centimeters) from the equilibrium position of the spring is small, we can model the force as a constant times this displacement, i.e. $F = -kx$, where $k = 0.114$ N/cm is the spring constant. (This is Hooke's law). We assume the motion of the spring to be along the x-axis and the position of the centre of mass of the block at time t to be $x(t)$. We then know that the acceleration is given by $a(t) = x''(t)$. Newton's second law applied to the spring now yields

$$mx''(t) = -kx(t). \tag{13.80}$$

Suppose that the block has mass $m = 0.25\,\text{kg}$ and that the spring starts from rest in a position 5.0 cm from its equilibrium so $x(0) = 5.0$ cm and $x'(0) = 0.0$ cm/s.

a) Rewrite this second order differential equation (13.80) as a system of two coupled differential equations and solve the system analytically.

b) Use the second order Runge-Kutta method to solve the set of differential equations in the domain $t \in [0, 1.5]$ seconds with 3 time steps, and plot the analytical and approximate numerical solutions together.

c) Did your numerical method and the number of steps suffice to give a good approximation?

13.23 This is a continuation of exercise 22, and all the constants given in that problem will be reused here. We now consider the case of a vertical spring and denote the position of the block at time t by $y(t)$. This means that in addition to the spring force, gravity will also influence the problem. If we take the positive y-direction to be up, the force of gravity will be given by

$$F_g = -mg. \tag{13.81}$$

Applying Newton's second law we now obtain the differential equation

$$my''(t) = -ky(t) - mg. \tag{13.82}$$

The equilibrium position of the spring will now be slightly altered, but we assume that $y = 0$ corresponds to the horizontal spring equilibrium position.

a) What is the new equilibrium position y_0?

b) We let the spring start from rest 5.0 cm above the new equilibrium, which means that we have $x(0) = 5.0\text{cm} + y_0$, $x'(0) = 0.0$ cm/s. Rewrite the second order differential equation as a system of two first order ones and solve the new set of equations analytically.

c) Choose a numerical method for solving the equations in the interval $t \in [0, 1.5]$ seconds. Choose a method and the number of time steps that you think should make the results good enough.

d) Plot your new analytical and numerical solutions and compare with the graph from exercise 22. What are the differences? Did your choice of numerical method work better than the second order Runge-Kutta method in exercise 22?