## Chapter 2

## Numerical methods for solving ODEs

We will study two methods for finding approximate solutions of ODEs. Such methods may be used for (at least) two reasons

- the ODE does not have an exact solution or, the solution is difficult to find
- even if we know or could find the exact solution we are only interested in obtaining a approximate solution (for example to estimate the solutions at a given time or to draw a plot of the solution.)

The methods to be considered are both easy to use but are not as fast or as reliably accurate as the methods most often used on a computer to solve real problems. Nevertheless they illustrate the important aspects of the techniques involved in numerical integration of ODEs.

## 2.1 Euler's method

Consider the initial value problem

$$\frac{dy}{dt} = f(t,y), \quad y(a) = A.$$
(2.1)

We will assume that this has a unique solution y(t). The value of this solution at t = a + h may be approximated using Taylor's theorem to obtain

$$y(a+h) \approx y(a) + h \frac{dy}{dt}(a),$$

and then using the ODE (2.1) to express the derivative of y in terms of y and t,

$$y(a+h) \approx y(a) + hf(a, y(a)).$$

If we define  $t_0 := a$ ,  $y_0 := y(a)$ ,  $t_1 := a + h$ , we can express the above result in the following way;

$$y_1 := y_0 + hf(t_0, y_0)$$

gives an approximation to  $y(t_1)$ . This method of approximating the solution to an initial value problem is usually called *Euler's method*. The *absolute error* (or sometimes just *error*)  $\varepsilon$  and the *relative error*  $\varepsilon_r$  in this approximation are

$$\varepsilon = |y(t_1) - y_1|, \quad \varepsilon_r = \frac{|y(t_1) - y_1|}{y(t_1)}$$

Figure 2.1 gives a geometrical interpretation of Euler's method in which the solution between  $t = t_0$ and  $t = t_1$  is approximated by the tangent to the solution curve at  $(t_0, y_0)$ .

We wish to repeat this process through a number of steps in time and to describe this it is useful to extend the above notation;

$$\begin{aligned} t_2 &:= t_1 + h, & y_2 &:= y_1 + hf(t_1, y_1) \\ t_3 &:= t_2 + h, & y_3 &:= y_2 + hf(t_2, y_2) \\ &\vdots & \vdots & \vdots \end{aligned}$$



Figure 2.1: Geometrical interpretation of Euler's method

that is,

$$t_0 := a, \quad t_{i+1} := t_i + h \quad \text{for } i = 0, 1, 2, \dots$$
 (2.2)

$$y_0 := y(t_0), \quad y_{i+1} := y_i + hf(t_i, y_i) \quad \text{for } i = 0, 1, 2, \dots$$
 (2.3)

Here  $y_i$  gives an approximation to  $y(t_i)$  and the lines joining  $(t_0, y_0)$  to  $(t_1, y_1)$  to  $\cdots$  to  $(t_n, y_n)$  gives an approximation to the solution curve y as a function of t on the time interval  $[t_0, t_n]$ , that is [a, a + nh], as shown in Figure 2.2.



Figure 2.2: Approximate solution found by Euler's method

**Euler's method algorithm** To approximate the solution to the initial value problem (2.1) on the interval [a, b] Euler's method expressed in algorithmic form as follows

Pseudo code	Maple code
Set t=a and y=A	t:=a; y:=A;
Store t and y	ans:=[t,y];
Repeat while t < b	while t <b< td=""></b<>
{	do
Set F=f(t,y), t=t+h and y=y+hF	F:=f(t,y); t:=t+h; y:=y+h*F;
Store t and y	<pre>ans:=ans,[t,y];</pre>
}	od;

This algorithm is illustrated in the following example.

**Example 2.1** Use Euler's method with h = 1.0, h = 0.5 and h = 0.25 to approximate y(1) where y(t) is the solution of the initial value problem

$$\frac{dy}{dt} = y + t, \quad y(0) = 1.$$

Show that the exact solution of this IVP is  $y(t) = 2e^t - t - 1$  and find the relative error in y(1) for each step length.

**Solution** We use the algorithm described above with F = f(t, y) = y + t and work to six significant figures throughout.

h = 1.0

F	t	y
-	0.0	1.0
1.0	1.0	2.0

Therefore h = 1 give the approximation  $y_1 = 2.0$  to y(1).

h=0.5

F	t	y
-	0.0	1.0
1.0	0.5	1.5
2.0	1.0	2.5

Therefore h = 0.5 give the approximation  $y_2 = 2.5$ .

h = 0.25

F	t	y
-	0.0	1.0
1.0	0.25	1.25
1.5	0.5	1.625
2.125	0.75	2.15625
2.90625	1.0	2.88281

Therefore h = 0.25 give the approximation  $y_4 = 2.88281$ . For  $y(t) = 2e^t - t - 1$ ,

$$\frac{dy}{dt} = 2e^t - 1 = y + t, \quad y(0) = 2e^0 - 0 - 1 = 1,$$

therefore  $y(t) = 2e^t - t - 1$  is the solution of the IVP. The exact answer is therefore y(1) = 2e - 2 = 3.43656. The relative errors for the different values of h are

h	$\varepsilon_r$
1.0	0.418023
0.5	0.272528
0.25	0.161135

Observe in the above table that as the step size h is halved, the relative error is also halved. For smaller h this phenomenon persists

h	$\varepsilon_r$
0.1	0.0724795
0.01	0.00783632
0.001	0.000788579

and we see that the relative error is of the same order as the step size. In fact this relationship between error size and step length is true in general and we write

$$\varepsilon_r = O(h^1),$$

and say that Euler's method is an order 1 method.

The Maple code to carry out Euler's method in the case h = 0.25 is shown below;

```
h:=0.25;
f:=(t,y)->t+y;
t:=0; y:=1;
ans:=[t,y];
while t<1
do
F:=f(t,y); t:=t+h; y:=y+h*F;
ans:=ans,[t,y];
od:
```

od:

After this, **ans** stores the values of t and y as the sequence

[0,1],[0.25,1.25],[0.5,1.6250],[0.75,2.156250],[1,2.88281250].

To plot the corresponding approximate solution curve, the following Maple code may be used

```
with(plots);
pointplot([ans],style=line);
```

and

```
with(plots);
display({plot(2*exp(t)-t-1,t=0..1),pointplot([ans],style=line),pointplot([ans])});
```

which plots the exact solution, this curve and the points  $(t_i, y_i)$  on the same axes. The resulting graph is shown in Figure 2.3.

## 2.2 The Euler predictor-corrector method

This method is also sometimes called the *improved Euler's method*. The aim of this method is the same as Euler's method, to approximate  $y(t_1) = y(a+h)$  for the solution of the IVP (2.1). First, as in Euler's method the tangent to the solution through  $(t_0, y_0)$ , of gradient  $f(t_0, y_0)$ , is used *predict* an approximation  $z_1$  to  $y(t_1)$ , that is

 $t_1 := t_0 + h, \quad z_1 := y_0 + hf(t_0, y_0).$ 

The gradient of the solution curve through  $(t_1, z_1)$  is then used to *correct* the first approximation by using the line through  $(t_0, y_0)$  whose gradient is the *average* of  $f(t_0, y_0)$  and  $f(t_1, z_1)$ 

$$y_1 := y_0 + \frac{h}{2}(f(t_0, y_0) + f(t_1, z_1))$$

This is illustrated in Figure 2.4

Euler predictor-corrector method algorithm To approximate the solution to the initial value problem (2.1) on the interval [a, b] the Euler predictor corrector method expressed in algorithmic form as follows

```
Pseudo code
                                        Maple code
Set t=a and y=A
                                         t:=a; y:=A;
                                         ans:=[t,y];
Store t and y
Repeat while t < b
                                         while t<b
{
                                         do
  Set F=f(t,y), t=t+h and z=y+hF
                                           F:=f(t,y); t:=t+h; z:=y+h*F;
  Set Z=f(t,z), y=y+h(F+Z)/2
                                           Z:=f(t,z); y:=y+h*(F+Z)/2;
  Store t and y
                                           ans:=ans,[t,y];
}
                                         od;
```





Figure 2.3: Euler's method and exact solution in Maple

**Example 2.2** Repeat the numerical parts of Example 2.1 using the Euler predictor-corrector method rather than Euler's method. Compare the relative errors for the two methods for the different step lengths.

**Solution** Here F = t + y and Z = t + z. In evaluating F and Z we always use the most recent values of the arguments; for F the values of t and y used are on the previous row but for Z they are on the same row.

h = 1.0

F	t	z	Z	y
-	0.0	-	-	1.0
1.0	1.0	2.0	3.0	3.0



Figure 2.4: The Euler predictor-corrector method

F	t	z	Z	y
-	0.0	-	-	1.0
1.0	0.5	1.5	2.0	1.75
2.25	1.0	2.875	3.875	3.28125

h = 0.25

F	t	z	Z	y
-	0.0	-	-	1.0
1.0	0.25	1.25	1.50	1.3125
1.5625	0.50	1.70312	2.20312	1.78320
2.28320	0.75	2.35400	3.10400	2.45660
3.20660	1.00	3.25826	4.25826	3.38971

Here we tabulate the relative errors for the two methods and different step lengths

h	Euler $\varepsilon_r$	Euler p-c $\varepsilon_r$
1.0	0.418023	0.127034
0.5	0.272528	0.0451934
0.25	0.161135	0.0136328

Using Maple we can also calculate approximations for smaller step lengths, together with the corresponding relative errors

	h	Euler $\varepsilon_r$	Euler p-c $\varepsilon_r$
0.	1	0.072479	0.00244430
0.	01	0.007836	0.00002616
0.	001	0.000789	0.00000026

From the above tables it is clear that the Euler predictor-corrector relative error is proportional to the *square* of the step length, in the example at least. In fact, it may be shown that this is true generally for this method and so

 $\varepsilon_r = O(h^2),$ 

i.e. Euler predictor-corrector is an order 2 method.