

Chapter Six

Numerical Methods

In previous chapters we have discussed methods of constructing the solutions for single ordinary differential equations and for systems of first order equations. In particular, we have considered initial value problems of the form

$$\begin{aligned}y'(t) &= F(t, y(t)), & 0 < t < T, & \quad (1) \\y(0) &= y_0.\end{aligned}$$

as well as the more difficult problem for dynamical systems

$$\begin{aligned}x'(t) &= F(t, x(t), y(t)), & 0 < t < T, \\y'(t) &= G(t, x(t), y(t)), & \quad (2) \\x(0) &= x_0, & y(0) = y_0.\end{aligned}$$

We saw that although it may be possible to determine that a solution exists, at least locally, it may not be possible to construct this solution in terms of elementary functions. In such cases we may be forced to resort to approximating the solution to the problem by means of a numerical method.

Here we are going to present only the simplest numerical scheme (Euler's method) just to illustrate how numerical methods are developed. With the advent of computer languages like Maple, Matlab and Mathematica, numerical methods for solving differential equations are available as packaged software so there is no need to write your own programs to implement numerical schemes. On the other hand, it is useful to have some idea of what the packaged programs are doing.

1. Terminology

A function, $y(t)$, which satisfies the conditions of the initial value problem (1) is what we have previously referred to as the (possibly) unique particular solution of the initial value problem. Here it will be referred to as an exact solution for (1) in order to distinguish between the analytic solution and the numerical approximation to the analytic solution. In order to construct an approximation for $y(t)$ it will be convenient to define a partition of the interval $[0, T]$ by letting $t_k = k(\frac{T}{N})$ for $k = 0, 1, \dots, N$. Then $\{y_0, \dots, y_N\}$ is called an approximate solution for (1) on the partition, $P_N = \{t_0, \dots, t_N\}$, if, for each k , $|y(t_k) - y_k|$ tends to zero as N tends to infinity.

We define the local truncation error for our approximate solution as the difference $E_k = |y(t_{k+1}) - y_{k+1}|$ assuming that $y(t_k) = y_k$. That is, E_k is the error made at the step where we compute y_{k+1} , assuming that the solution is correct up to the point y_k . On the other hand, the accumulated error is defined as $e_k = |y(t_k) - y_k|$ assuming only that $y(0) = y_0$. That is, e_k is the error that is accumulated up to computing y_k , starting with the correct initial value but allowing errors at each subsequent step up to the k -th. Similar terminology applies to the dynamical system (2). We now describe some algorithms by means of which an approximate solution can be constructed.

2. Numerical Methods for Constructing Solutions

Taylor's theorem asserts that for $y(t)$ sufficiently smooth, we have

$$y(t_{k+1}) = y(t_k) + y'(t_k)h + \frac{1}{2}y''(t_k)h^2 + \frac{1}{3!}y^{(3)}(t_k)h^3 + \dots$$

where $h = \frac{T}{N}$. Then we can approximate as follows,

$$y(t_{k+1}) \approx y(t_k) + y'(t_k)h.$$

Since it follows that the solution to (1) satisfies, $y'(t_k) = F(t_k, y(t_k))$, we have the following algorithm for constructing approximate values y_k , starting from the initial value y_0 ,

$$y_{k+1} \approx y_k + hF(t_k, y_k), \quad k = 0, 1, \dots, N-1. \quad (3)$$

This algorithm is known as Euler's method for approximating the solution to (1).

Applying the same approach to the dynamical system (2), we have

$$x(t_{k+1}) \approx x(t_k) + x'(t_k)h$$

$$y(t_{k+1}) \approx y(t_k) + y'(t_k)h$$

Then it follows from (2) that approximate solution values can be computed from

$$x_{k+1} \approx x_k + F(t_k, x_k, y_k)h$$

$$y_{k+1} \approx y_k + G(t_k, x_k, y_k)h \quad (4)$$

Here we begin with the initial values x_0, y_0 and k runs from zero to N .

Examples

1. Consider the initial value problem

$$y'(t) = 3t^2y^2 \quad y(0) = 1.$$

This equation is separable and the exact solution to the initial value problem is found to be

$$y(t) = \frac{1}{1-t^3}.$$

We shall attempt to approximate the exact solution on the interval $[0, 1]$. Note that the exact solution becomes undefined at $t = 1$ although there is nothing in the initial value problem to allow us to anticipate this if the exact solution is not known. We have $h = 1/N$ and $\{t_0, \dots, t_N\} = \{0, h, 2h, \dots, 1\}$ for various values of N . Using Euler's method (3), we have

$$y_0 = 1$$

$$y_1 = y_0 + 3[t_0^2 y_0^2]h$$

$$y_2 = y_1 + 3[t_1^2 y_1^2]h$$

$$\vdots$$

$$y_N = y_{N-1} + 3[t_{N-1}^2 y_{N-1}^2]h$$

For $N = 4$, this would give $P_4 = \{0, .25, .50, .75, 1\}$ and

$$y_0 = 1$$

$$y_1 = 1 + 0$$

$$y_2 = 1 + 3(.25^2)(1^2)(.25) = 1.05$$

$$y_3 = 1.05 + 3(.5^2)(1.05^2)(.25) = 1.26$$

$$y_4 = 1.26 + 3(.75^2)(1.25^2)(.25) = 1.92$$

For $N = 8$, we have the partition $P_8 = \{0, .125, .25, .375, .50, .625, .75, .875, 1\}$ and

$$\begin{aligned}
y_0 &= 1 \\
y_1 &= 1 + 0 \\
y_2 &= 1 + 3(.125^2)(1^2)(.125) = 1.006 \\
y_3 &= 1.006 + 3(.25^2)(1.006^2)(.125) = 1.03 \\
y_4 &= 1.03 + 3(.375^2)(1.03^2)(.125) = 1.086 \\
y_5 &= 1.086 + 3(.5^2)(1.086^2)(.125) = 1.197 \\
y_6 &= 1.197 + 3(.625^2)(1.197^2)(.125) = 1.407 \\
y_7 &= 1.407 + 3(.75^2)(1.407^2)(.125) = 1.825 \\
y_8 &= 1.825 + 3(.875^2)(1.825^2)(.125) = 2.781
\end{aligned}$$

The values for the exact solution on P_8 are as follows:

$$\begin{aligned}
y_0 &= 1 \\
y(.125) &= \frac{1}{1 - .125^2} = 1.0159 \\
y(.25) &= \frac{1}{1 - .25^2} = 1.0667 \\
y(.375) &= \frac{1}{1 - .375^2} = 1.1636 \\
y(.5) &= \frac{1}{1 - .5^2} = 1.3333 \\
y(.625) &= \frac{1}{1 - .625^2} = 1.641 \\
y(.75) &= \frac{1}{1 - .75^2} = 2.2857 \\
y(.875) &= \frac{1}{1 - .875^2} = 4.2667 \\
y(1) &= \infty
\end{aligned}$$

The local truncation errors E_k appear to be somewhat smaller when $N = 8$ than when $N = 4$, although in neither case are the E_k 's particularly small. We can show that E_k is proportional to the step size h , so as h tends to zero, the errors decrease.

2. Consider the dynamical system

$$\begin{aligned}
x'(t) &= y(t), & x(0) &= 0, \\
y'(t) &= -4x(t), & y(0) &= 4.
\end{aligned}$$

The exact solution for this initial value problem for the dynamical system is easily found to be $4x^2 + y^2 = 4$, so the orbit is an ellipse. The numerical solution is generated by the Euler's method algorithm for systems,

$$\begin{aligned}
x_0 &= 0 & y_0 &= 4 \\
x_{n+1} &= x_n + y_n h & y_{n+1} &= y_n - 4x_n h
\end{aligned}$$

The first few steps of this computation proceed as follows

$$\begin{array}{ll}
 x_0 = 0 & y_0 = 4 \\
 x_1 = 0 + 4h & y_1 = 4 - 0 \cdot h \\
 x_2 = 4h + 4h & y_2 = 4 - 4(4h)h \\
 x_3 = 8h + 4h & y_3 = 4 - 4(4h)h - 4(8h)h \\
 \vdots & \vdots
 \end{array}$$

Clearly, this computation is very tedious and can most efficiently be carried out using a computer. It is worth noting that although the exact solution trajectory is an ellipse, a closed curve, the numerical solution will be seen to be not a closed curve, but rather a spiral if h is larger than about .001. For smaller step sizes, h , the trajectory appears to be a closed curve as it should.

Exercises