Numerical Approximations

As the problems at the end of Chapter One show, even though the solution may exist, carrying out the integration may be impossible. We need a way to approximate the solutions computationally.

Let's suppose we wish to approximate solutions to

$$\frac{dy}{dt} = F(t, y), \tag{2.1}$$
$$y(0) = y_0.$$

We suppose for simplicity that the initial data is given at $t = 0$ and that we wish to approximate the solution for positive t. We assume that (2.1) has a smooth solution on the interval $0 \le t \le T$ for some $T > 0$. We want to find a way to approximate solutions to (2.1) on the interval $0 \le t \le T$.

If we are to implement our approximations on a computer, we first have to discretize the time since computers do not understand continuous time. To do this we subdivide the interval $0 \le t \le T$ into N equal parts. Here N is a natural number. We set

$$\Delta t = \frac{T}{N},$$

and set $t_0 = 0,\ t_1 = \Delta t,\ t_2 = 2\Delta t,\ \ldots,\ t_N = T$. We need to find a way to fill in the table

Time	Approximation
$0 = t_0$	$y_0 = y(0)$
$\Delta t = t_1$	$y_1 \approx y(\Delta t)$
$2\Delta t = t_2$	$y_2 \approx y(2\Delta t)$
$3\Delta t = t_3$	$y_3 \approx y(3\Delta t)$
$4\Delta t = t_4$	$y_4 \approx y(4\Delta t)$
$\vdots$	$\vdots$
$N\Delta t = T$	$y_N \approx y(T)$

Once we find a way to compute y_n, the data can be used to construct plots to reveal qualitative features of the solutions to (2.1), or to provide precise estimates of the solution for engineering problems. We will study three numerical schemes in this chapter.

2.1. Forward Euler

A Forward Euler scheme, sometimes called an Explicit Euler, is perhaps the simplest scheme to implement. To construct the approximation, we must first deal with the time derivative in (2.1). Recall that the derivative is define by

$$\frac{dy}{dt}(t) = \lim_{h\to 0} \frac{y(t+h) - y(t)}{h}.$$

Once again the computer has no chance of understanding this quantity. Since we have discretized the time, no information about the solution on time scales less than Δt can be found. This suggest replacing the derivative with

$$\frac{dy}{dt}(t) \approx \frac{y(t+\Delta t) - y(t)}{\Delta t}.$$

The Forward Euler scheme is as follows. For $1 \le n \le N$

$$\begin{aligned} y(0) &= y_0, \\ \frac{dy}{dt} &= F(t,y), \end{aligned} \qquad \text{(Truth)}$$

$$\begin{aligned} y_0 &= y_0, \\ \frac{y_{n+1} - y_n}{\Delta t} &= F(t_n, y_n). \end{aligned} \qquad \text{(Approximation)}$$

To implement a Forward Euler method, set y_0 to the given initial data. Then y_1 is obtained from

$$y_1 = y_0 + \Delta t F(t_0, y_0).$$

Note all of the quantities on the right are known from the previous step. Once y_1 is known, y_2 can be computed in a similar fashion and so on.

Example 2.1. Approximate (Problem 1.2.1) the solution to

$$y' - y = 2te^{2t},$$

$$y(0) = 1$$

at $t = .2$ if $\Delta t = .1$ using a forward Euler scheme.

Solution. Here $F(t,y) = y + 2te^{2t}$ and $y_0 = 1$. Since the time step is $\Delta t = .1$, we need to iterate the solution twice. That is, we need to find y_2. The first iterate gives

$$\begin{aligned} y_1 &= y_0 + \Delta t F(t_0, y_0) \\ &= 1 + 0.1\Big(1 + 2\cdot 0 \cdot e^{2\cdot 0}\Big) \\ &= 1 \end{aligned}$$

and

$$\begin{aligned} y_2 &= y_1 + \Delta t F(t_1, y_1) \\ &= 1 + 0.1\Big(1 + 2 \cdot 0.1 \cdot e^{2 \cdot 0.1}\Big) \\ &\simeq 1.2344 \end{aligned}$$

This is to be compared to the exact answer which is $y(.2) = 3e^{0.2} + 2(0.2 - 1)e^{2 \cdot 0.2} \simeq 1.2723$.

Table 1 below provides a better comparison of the error. Notice that the error, $|y_n - y(n\Delta t)|$, is cut in half when the time step is halved.

Time	Error		
	$\Delta t = .1$	$\Delta t = .05$	$\Delta t = .025$
.05	.00399	.0021	.00104
.1	.00896	.0046	.00233
.15	.0151	.0077	.000393
.2	.0226	.0116	.000589
.25	.0317	.0163	.000828

Table 1. Error Vs. Time for the ODE in Example 2.1 using a forward Euler scheme.

In Figure 1 we compare the numerical solutions to the exact solution. Notice that as time increases the error gets worse. This is because we start by replacing the true derivative with a discrete derivative; this introduces an error at the first iterate. The next iterate makes the same error only we use *bad* information from the first iterate. The errors accumlate.

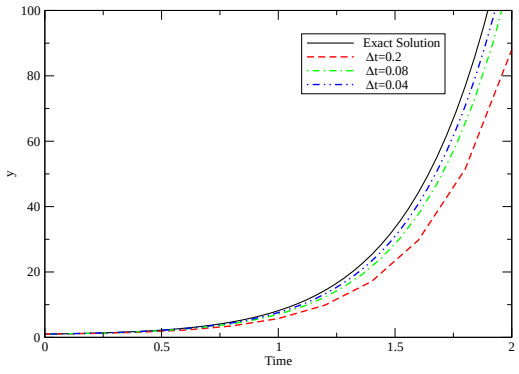

Figure 1. Solutions to $y' - y = 2te^{2t}$ and a forward Euler approximation

We should address the accuracy and choice of Δt in a forward Euler scheme. As the table and figure above indicate, the smaller Δt the more accurate the approximation. The theorem below will corborate this observation. In practice, making

the time step too small will allow computer round-off error to dominte the error and the scheme will cease to improve in accuracy below some Δt.

Theorem 2.2. *(Forward Euler) Suppose $y' = F(t, y)$, $y(0) = y_0$ has a smooth solution on $0 \leq t \leq T$. Let N be a natural number, and set $\Delta t = T/N$. Then there exists a constant, $C(T)$ such that*

$$\max_{1 \leq n \leq N} |y_n - y(n\Delta t)| \leq C(T)\Delta t.$$

The theorem indicates a forward Euler scheme is a first-order scheme - the convergence is proportional to Δt to the power one. If the time step is cut in half, then the error will be reduced by a half. This is a low convergence rate. In general the constant $C(T)$ grows with increasing T, reflecting the accumulation of errors as time increases.

Before studying more accurate scheme, we show what will happens if Δt is chosen too large.

Example 2.3. Consider the ODE

$$y' = -y,$$
$$y(0) = 1.$$

Of course the exact solution is $y(t) = e^{-t}$. Applying a forward Euler scheme, we find

$$\begin{aligned} y_0 &= 1 \\ y_1 &= y_0 - \Delta t y_0 = (1 - \Delta t) \\ y_2 &= y_1 - \Delta t y_1 = (1 - \Delta t)^2 \\ &\vdots \quad\quad \vdots \\ y_n &= (1 - \Delta t)^n. \end{aligned}$$

Notice what happens if $\Delta t = 3$. We find $y_n = (-2)^n$ which oscillates wildly. This oscillatory behavior is typical of numerical schemes when the time step is chosen too large. Figure 2 below depicts the situation.

We will examine the behavior of numerical approximation more closely when we go to the lab.

2.2. Backward Euler

A Backward Euler, sometime called an implicit Euler, is given by

$$\begin{aligned} y(0) &= y_0, \\ \frac{dy}{dt} &= F(t, y), \end{aligned} \qquad \text{(Truth)}$$

$$\begin{aligned} y_0 &= y_0, \\ \frac{y_{n+1} - y_n}{\Delta t} &= F(t_{n+1}, y_{n+1}). \end{aligned} \qquad \text{(Approximation)}$$

It is not easy to implement!

Example 2.4. Apply a backward Euler method to the ODE

$$y' = \sin y,$$

$$y(0) = 1.$$

Here $F(t, y) = \sin y$, and the first iteration in the approximation is

$$\begin{aligned} y_0 &= 1 \\ y_1 &= y_0 + \sin y_1. \end{aligned}$$

To continue the iterations we must solve $y_1 = 1 + \sin y_1$. The solution to this nonlinear equation is not readily found. Its solution would require a study, which we will not do, of nonlinear solvers such as Newton's method. Given the extra work required to carry out a backward Euler, it must be a more accurate scheme - right? It is not as the next theorem shows.

Theorem 2.5. *(Backward Euler) Suppose $y' = F(t, y)$, $y(0) = y_0$ has a smooth solution on $0 \leq t \leq T$. Let N be a natural number, and set $\Delta t = T/N$. Then there exists a constant, $C(T)$ such that*

$$\max_{1 \leq n \leq N} |y_n - y(n\Delta t)| \leq C(T)\Delta t.$$

The convergence rate is exactly the same as for a forward Euler! Why use the scheme? To see why, we return to Example 2.3.

Example 2.6. Again consider the ODE

$$y' = -y,$$

$$y(0) = 1.$$

The exact solution is $y(t) = e^{-t}$. Applying a backward Euler scheme, we find

$$\begin{aligned} y_0 &= 1 \\ y_1 &= y_0 - \Delta t y_1. \end{aligned}$$

Solving for y_1, we find

$$y_1 = \frac{1}{1 + \Delta t}.$$

Repeating the process gives

$$y_n = \frac{1}{(1 + \Delta t)^n}.$$

Recall that the exact solution is $y(t) = e^{-t}$. Note for **any** Δt the scheme does not oscillate! It is stable for all Δt. This is a nice feature for a numerical schemes to have, and in some instances, well worth the extra effort to code.

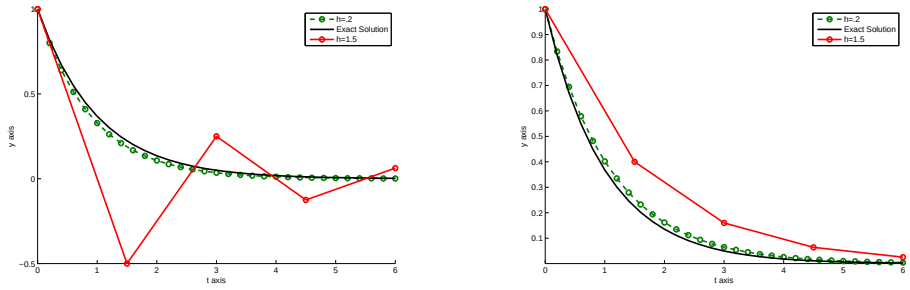

Figure 2. Exact and numerical solutions to $y' = -y$, $y(0) = 1$ using a forward (left) and backwards (right) Euler approximation ($h = \Delta t$).

2.3. Improved Euler

We seek a more accurate scheme. Here is an idea. A forward Euler scheme looses accuracy in part because of a lack of symmetry. Recall we replace the exact derivative with the discrete derivative

$$\frac{dy}{dt} \to \frac{y_{n+1} - y_n}{\Delta t}.$$

The discrete derivative, $\frac{y_{n+1}-y_n}{\Delta t}$ is supposed to approximate y' at $n\Delta t$, but all of the information in the discrete derivative is to the right of $n\Delta t$. If we thought of the discrete time derivative as approximating y' at $(n+1/2)\Delta t$, then the derivative would be symmetric - information in the slope is equally distributed around the point of interest. A symmetric scheme is more accurate because some error terms cancel due to the symmetry. Thus we might expect the schemes

$$\frac{y_{n+1} - y_{n-1}}{2\Delta t} = F(t_n, y_n)$$

or

$$\frac{y_{n+1} - y_n}{\Delta t} = \frac{1}{2}(F(t_n, y_n) + F(t_{n+1}, y_{n+1}))$$

to be more accurate. The first of these (called a Leap frog scheme) turns out to be unstable. The second scheme contains implicit terms, $F(t_{n+1}, y_{n+1}))$. We will not pursue the first scheme any further (but it is second-order accurate). To make the second scheme explicit, we approximate y_{n+1} inside $F(t, y)$ using a forward Euler. The new scheme is called an Improved Euler scheme. It is given by

$$y_0 = y(0),$$

$$\frac{y_{n+1} - y_n}{\Delta t} = \frac{1}{2}\Big(F(t_n, y_n) + F(t_{n+1}, y_n + \Delta t F(t_n, y_n))\Big)$$

Notice to implement the scheme the nonlinear term must be computed twice, adding to the expense of the scheme.

Theorem 2.7. *(Improved Euler) Suppose $y' = F(t, y)$, $y(0) = y_0$ has a smooth solution on $0 \leq t \leq T$. Let N be a natural number, and set $\Delta t = T/N$. Then there exists a constant, $C(T)$ such that*

$$\max_{1 \leq n \leq N} |y_n - y(n\Delta t)| \leq C(T)\Delta t^2.$$

Notice the scheme is second order - half the time step Δt and the error is one fourth.

Homework 2 (Numerical Methods)

Consider the ODE

$$y' = 2y - 3t,$$
$$y(0) = 1.$$

Suppose $\Delta t = .1$.

1. Use a forward Euler method to approximate $y(.2)$.
2. Use a Backward Euler method to approximate $y(.2)$.
3. Use an improved Euler method to approximate $y(.2)$.