Mathematics 255—January 18, 1996

More about numerical approximations

The error in Euler's method

We have seen so far what is known as **Euler's method** for approximating the solution of a differential equation plus initial value condition

$$y' = F(x, y), \quad y(x_0) = y_0$$

in the range $[x_0, x_f]$. It runs like this:

(1) We choose an integer N, the number of steps we are going to take.

(2) We set an x-interval to be

$$\Delta x = \frac{x_f - x_0}{N}$$

(3) We find successive approximate values for $y(x_0)$, $y(x_1)$, ... at values x_0, $x_1 = x_0 + \Delta x$, etc. according to the following recipe:

$$\begin{aligned} x_{n+1} &= x_n + \Delta x \\ y_{n+1} &= y_n + \Delta x\, F(x_n, y_n) \end{aligned}$$

for $n = 0, 1, \ldots, N - 1$.

For example with the equation

$$y' = y, \quad y(0) = 1$$

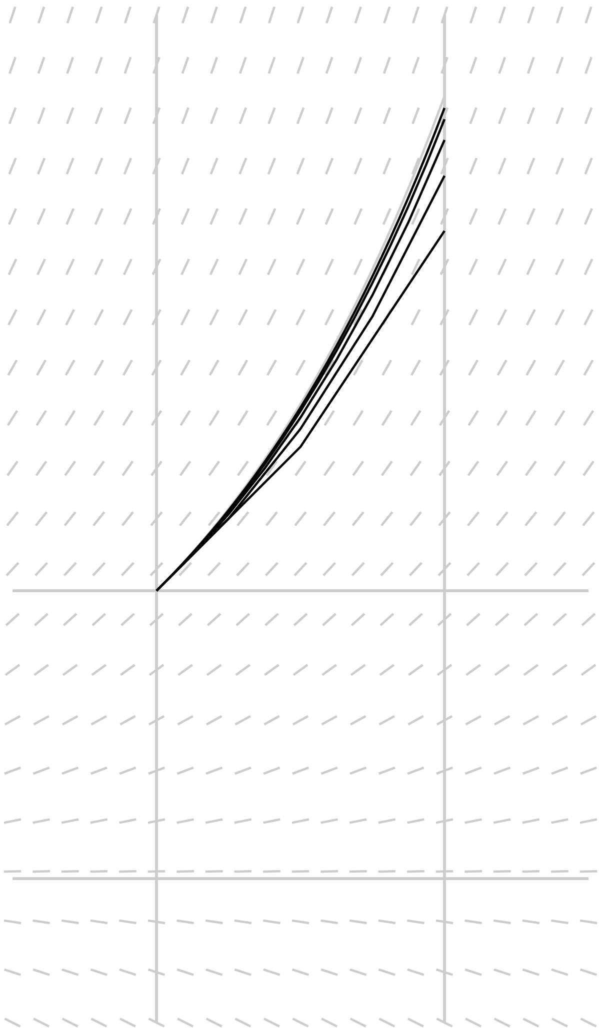
and $N = 4$ we get a sequence

x_n	y_n
0.00000000	1.00000000
0.25000000	1.25000000
0.50000000	1.56250000
0.75000000	1.95312500
1.00000000	2.44140625

Here are the approximations you get for $y(1)$ with $N = 4, 8, 16, 32, 64$. In the column next to these approximations I have put the true value of $y(1) = e$, and in the last column the error:

N	$y(1)$		
4	2.44140625	2.71828183	0.27687558
8	2.56578451	2.71828183	0.15249731
16	2.63792850	2.71828183	0.08035333
32	2.67699013	2.71828183	0.04129170
64	2.69734495	2.71828183	0.02093688

What happens shows up in these graphs of the approximate solutions:



The universal rule for estimating the error in Euler's method is that *when you double the number of steps, the error gets cut essentially in half.* In other words, the error in Euler's method is essentially proportional to the step size Δx.

We can use this to estimate the error in Euler's method even when we don't know the exact answer! For example, suppose we look at the equation

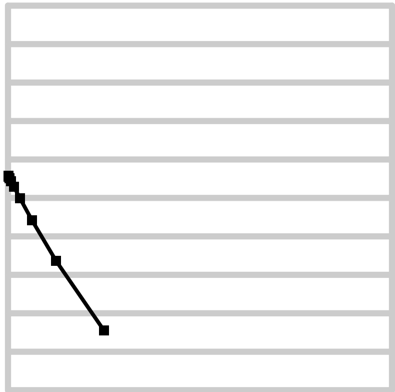
$$y' = xy + 1, \quad y(0) = 1$$

This is a linear equation, but we have seen that the general formula comes up with an integral we cannot find explicitly. Here is the estimate we get for $y(1)$ with various values of N:

4	2.65515137
8	2.83657060
16	2.94189329
32	2.99898700
64	3.02876204
128	3.04397338

256	3.05166223
512	3.05552774
1024	3.05746579

And here is a graph of the pairs $(\Delta x, y(1))$. The top and bottom horizontal lines are at $y = 2.5$, $y = 3.5$.



What we can see from this graph is that the points $(\Delta x, y(1))$ are to a good approximation on a straight line. This is in complete agreement with the principle I stated above, that the error is proportional to Δx. But we can also see that to some extent we can predict what the end of the line is. This is because we can estimate from each successive pair what the slope of the purported straight line is—it ought to be about

$$\frac{y_n - y_{n-1}}{\Delta_n - \Delta_{n-1}}$$

for each n. Of course this is only an approximation, so we don't expect any of these estimates to be exact. Here is what we get in the runs above:

N	$\Delta = 1/N$	$y_n(1)$	$y_n(1) - y_{n-1}(1)$	estimated slope
4	0.25000000	2.65515137	*	*
8	0.12500000	2.83657060	0.18141923	−1.45135383
16	0.06250000	2.94189329	0.10532269	−1.68516309
32	0.03125000	2.99898700	0.05709371	−1.82699868
64	0.01562500	3.02876204	0.02977504	−1.90560247
128	0.00781250	3.04397338	0.01521134	−1.94705210
256	0.00390625	3.05166223	0.00768885	−1.96834562
512	0.00195312	3.05552774	0.00386551	−1.97913877
1024	0.00097656	3.05746579	0.00193806	−1.98457249

The rough proportionality certainly shows up here, and the last column tells us roughly what the constant of proportion is.

A simple improvement

Euler's method is quite inefficient. Because the error is essentially proportional to the step size, if we want to halve our error we must double the number of steps taken across the same interval. So if 10 steps give us one significant figure, then 100 give us 2 figures, 1000 give us 3, 1000000 give us 6, and 10^n give us n figures. Not too good, because even a fast computer will think a bit about doing a billion operations. The method can be improved dramatically by a simple trick. Look again at what goes on in one step of Euler's method. Say we are at step n. We know x_n and y_n and want to calculate x_{n+1}, y_{n+1}. Essentially we are going to assume that the slope across the interval $[x_n, x_n + \Delta x] = [x_n, x_{n+1}]$ is what it is at x_n. Of course it is not true, but it is very roughly OK. We use this assumption to get an approximate value

$$y_* = y_n + \Delta x\, F(x_n, y_n)$$

for y_{n+1}. But now we can already see whether our assumption is good, because we can calculate the true slope at (x_{n+1}, y_*). It will be

$$F(x_{n+1}, y_*)$$

The improvement we are going to make is to *assume in the new method that the slope across the interval is the average of these two slopes at either end.* In other words we calculate the sequence

$$\begin{aligned} s_0 &= F(x_n, y_n) \\ y_* &= y_n + \Delta x\,(s_0) \\ x_{n+1} &= x_n + \Delta x \\ s_1 &= F(x_{n+1}, y_*) \\ y_{n+1} &= y_n + \Delta x \left(\frac{s_0 + s_1}{2}\right) \end{aligned}$$

This is in contrast with the simpler method:

$$\begin{aligned} s &= F(x_n, y_n) \\ y_{n+1} &= y_n + \Delta x\,(s) \\ x_{n+1} &= x_n + \Delta x \end{aligned}$$

We are doing essentially twice as much work in one step, but it should be a somewhat more accurate. It is, and dramatically so.

Let's see, for example, how the new method does with the simple problem

$$y' = y, \quad y(0) = 1$$

with $\Delta x = 0.25$.

x	y	s_0	y_*	s_1
0.00000000	1.00000000	1.00000000	1.25000000	1.25000000
0.25000000	1.28125000	1.28125000	1.60156250	1.60156250
0.50000000	1.64160156	1.64160156	2.05200195	2.05200195
0.75000000	2.10330200	2.10330200	2.62912750	2.62912750
1.00000000	2.69485569			

This is certainly much more accurate than the equivalent run for Euler's method:

x	y	s	s
0.00000000	1.00000000	1.00000000	
0.25000000	1.25000000	1.25000000	
0.50000000	1.56250000	1.56250000	
0.75000000	1.95312500	1.95312500	
1.00000000	2.44140625		

I recall that the final value of y should be $e = 2.71828183$.

Here is a list of values of approximations of $y(1)$ for different N:

N	Δx	$y(1)$	e	error
2	0.50000000	2.64062500	2.71828183	0.07765683
4	0.25000000	2.69485569	2.71828183	0.02342614
8	0.12500000	2.71184124	2.71828183	0.00644059
16	0.06250000	2.71659352	2.71828183	0.00168831
32	0.03125000	2.71784967	2.71828183	0.00043215
64	0.01562500	2.71817251	2.71828183	0.00010932
128	0.00781250	2.71825434	2.71828183	0.00002749
256	0.00390625	2.71827494	2.71828183	0.00000689
512	0.00195312	2.71828010	2.71828183	0.00000173
1024	0.00097656	2.71828140	2.71828183	0.00000043

Here the error turns out to be proportional to (Δx^2).

N	error$/(\Delta x)^2$
4	0.37481821
8	0.41219772
16	0.43220621
32	0.44252572
64	0.44776012
128	0.45039090
256	0.45169110
512	0.45226337
1024	0.45223391

This is always true: *the error of the improved Euler's method is proportional to the square of the step size.*

This is much more efficient. If it takes 10 intervals to get 2 decimal accuracy, it will take 100 to get 4 decimals, 1000 to get 6 (as opposed to say 1000000 in Euler's method).

Relations with numerical calculation of integrals

We can understand better how the two methods compare if we apply to them to the special type of differential equation

$$y' = F(x)$$

where $F(x, y)$ does not depend on y. This equation can be solved explicitly by integration:

$$y = \int^x F(t)\, dt$$

and finding a numerical approximation to its solution amounts to numerical calculation of integrals. Euler's method gives us

$$y_{n+1} = y_n + \Delta x\, F(x_n)$$

This is the same calculation as in the **rectangle rule**. The second method gives us

$$\begin{aligned} s_0 &= F(x_n) \\ x_{n+1} &= x_n + \Delta x \\ s_1 &= F(x_{n+1}) \\ y_{n+1} &= y_n + \Delta x \left(\frac{s_0 + s_1}{2}\right) \end{aligned}$$

which is the **trapezoid rule**. These pictures explain in this case why the errors are proportioinal to (Δx) and $(\Delta x)^2$ respectively.