

Lab 2: Numerical Methods of Euler

A *numerical method* for *approximating* the solution of the initial-value problem

$$(*) \quad \begin{cases} y' &= f(x, y) \\ y(a) &= y_0 \end{cases}$$

involves replacing the continuous variable x by a set of discrete values $x_0 = a$, $x_1 = x_0 + h$, $x_2 = x_1 + h = x_0 + 2h$, \dots with uniform *step size* h . The goal is to produce suitable *approximations* y_1, y_2, \dots to the *true values* $y(x_1), y(x_2), \dots$ of the solution $y(x)$ evaluated at the points x_1, x_2, \dots . Usually, the approximations y_1, y_2, \dots are defined using an *iteration scheme*: assuming we have found y_1, y_2, \dots, y_n , what is the recipe for finding y_{n+1} ? Since we are given y_0 in the initial condition, we can find y_1 , and then using y_0 and y_1 we find y_2 , etc. The different numerical methods discussed below all follow this pattern, and only differ in the particular iteration scheme. (Any one method may be made more accurate by decreasing the step size h .)

I. EULER'S METHOD. The idea of this method is to approximate the curve $y(x)$ on $[x_n, x_{n+1}]$ by a straight line with slope $s = \Delta y / \Delta x = (y_{n+1} - y_n) / h = f(x_n, y_n)$. Solving for y_{n+1} we obtain the simple iteration scheme

$$(E) \quad y_{n+1} = y_n + hf(x_n, y_n).$$

Computing the values y_1, y_2, \dots can be time-consuming by hand, but is a simple task for a computer. One can write a simple computer program (say in PASCAL or BASIC) to compute the sequence y_1, y_2, \dots , but we shall carry out the iteration on a spreadsheet program such as EXCEL, which is available on all the machines in the Math Computer Lab in 553 LA. Spreadsheets have the advantage of giving a clear visual display of the procedure, as well as a graphical representation of the outcome.

Let us discuss a simple example.

Example 1. Use Euler's method to approximate the solution of $y' = x + y$, $y(0) = 1$ with step size $h = 0.1$ on the interval $0 \leq x \leq 1$.

1. Introduce a vertical indexing column $n = 0, 1, \dots, 10$ and then a vertical column of the corresponding values of x_n .

(In EXCEL, this can easily be achieved by the following: Type "n" in cell A1 to label the column, type "0" in cell A2, type "=A2+1" in cell A3, drag down to select cells A3-A12, and select "Fill Down" from the Edit menu. Type "xn" in cell B1, type "0" in cell B2, type "=B2+.1" in cell B3, drag down to select cells B3-B12, and select "Fill Down" from the Edit menu.)

2. In the next vertical column, place the values of y_n .

(This is achieved by labelling column C as “ yn ”, typing “1” in cell C2, typing “=C2+.1*D2” in C3, and filling down C3 into C4-C12. Do not worry about the values that appear in column C at this point.)

3. In the next vertical column, place the values of the slopes $s_n = f(x_n, y_n)$.
 (This is achieved by labelling column D as “ $sn = f(xn, yn)$ ”, typing “=B2+C2” in cell D2, and filling down D2 into D3-D12. Notice that this affects the C column as well, which now displays the desired values y_n .)

To investigate the accuracy of Euler’s method, let us compare the approximations y_1, y_2, \dots with the actual solution values $y(x_1), y(x_2), \dots$. Of course, if we can actually solve (*), then we do not even need a numerical method. But for experimental purposes, let us consider a linear equation $y' + P(x)y = Q(x)$ for which the solution is given in the text (Blanchard, Devaney, & Hall, Section 1.9). In particular, for the initial value problem of Example 1, $y' = x + y$, $y(0) = 1$, we find the solution is $y(x) = 2e^x - x - 1$. (Be sure that you know how to find $y(x)$.)

Example 1 (cont’d). Compare the approximations y_1, \dots, y_{10} with the true values $y(x_1), \dots, y(x_{10})$.

5. Create column E for the values $y(x_n)$.
 (Use the formula “= 2 * exp(B2) - B2 - 1” in cell E2, and then fill down into E3-E12.)
6. Create column F for the difference $y(x_n) - y_n$.
 (Use the formula “= E2 - C2” in cell F2, and then fill down into F3-F12.)

Your spreadsheet should now look like the following:

n	xn	yn	$sn = f(xn, yn)$	$y(xn)$	$y(xn) - yn$
0	0	1	1	1	0
1	0.1	1.1	1.2	1.110342	0.010342
2	0.2	1.22	1.42	1.242806	0.022806
3	0.3	1.362	1.662	1.399718	0.037718
4	0.4	1.5282	1.9282	1.583649	0.055449
5	0.5	1.72102	2.22102	1.797443	0.076423
6	0.6	1.943122	2.543122	2.044238	0.101116
7	0.7	2.197434	2.897434	2.327505	0.130071
8	0.8	2.487178	3.287178	2.651082	0.163904
9	0.9	2.815895	3.715895	3.019206	0.203311
10	1	3.187485	4.187485	3.436564	0.249079

Notice that the differences $y(x_n) - y_n$ get larger as n increases; in other words, the accuracy of the approximations gets worse as we get further away from the initial value (a, y_0) .

Exercise 1. Consider the initial-value problem $y' - 2y = 3e^{2x}$, $y(0) = 0$.

- (a) Use Euler's method with step size $h = .1$ on the interval $0 \leq x \leq 2$ to find the approximations y_1, \dots, y_{20} . What is the approximation for $y(2)$ that you have obtained?
- (b) Find the actual solution $y(x)$ using Section 1.9 in the text. What is the *actual* value $y(2)$?
- (c) Use Euler's method with step size $h = .05$ on the interval $0 \leq x \leq 2$ to find approximations y_1, \dots, y_{40} . What is the approximation for $y(2)$ that you have obtained?
- (d) Which of the approximations (a) or (c) is closer to the actual value (b)? How could you improve the approximation even more?

→**Hand In:** A print-out of your EXCEL worksheet and your answers to all five of the questions asked in (a-d). (Be very complete in your answer to the last question.)

II. IMPROVED EULER'S METHOD. As we have seen, inaccuracies in Euler's method develop quickly with successive steps. This is partly due to the asymmetry in using the slope at x_n to approximate y_{n+1} without taking into account the slope at x_{n+1} . Consider, instead, the following iteration scheme:

$$(E^*) \quad \begin{aligned} y_{n+1} &= y_n + h(k_1 + k_2)/2 \\ k_1 &= f(x_n, y_n) \\ k_2 &= f(x_{n+1}, y_n + hk_1). \end{aligned}$$

Notice that k_1 , which we formerly called s_n , is the slope at the starting point (x_n, y_n) , and k_2 is the slope at the point $(x_{n+1}, y_n + hk_1)$ which the Euler method produced. The *improved Euler method* (E^*) uses the *average* of these two slopes to produce the new value y_{n+1} .

If we use EXCEL to perform the improved Euler method on the problem of Example 1, we obtain the following display:

n	x_n	y_n	k_1	k_2	$y(x_n)$	$y(x_n) - y_n$
0	0	1	1	1.2	1	0
1	0.1	1.11	1.21	1.431	1.110342	0.000342
2	0.2	1.24205	1.44205	1.686255	1.242806	0.000756
3	0.3	1.398465	1.698465	1.968312	1.399718	0.001252
4	0.4	1.581804	1.981804	2.279985	1.583649	0.001845
5	0.5	1.794894	2.294894	2.624383	1.797443	0.002549
6	0.6	2.040857	2.640857	3.004943	2.044238	0.003380
7	0.7	2.323147	3.023147	3.425462	2.327505	0.004358
8	0.8	2.645578	3.445578	3.890136	2.651082	0.005504
9	0.9	3.012364	3.912364	4.40360	3.019206	0.006843
10	1	3.428162	4.428162	3.870978	3.436564	0.008402

where, as before, y_n are the approximations, and $y(x_n)$ are the exact values. (Notice that the value of k_2 in the last line is suspicious, since it *decreased* from the previous line. This is because the formula for the n th line of k_2 requires the value x_{n+1} which does not exist for $n = 10$; so EXCEL assigns the default value 0, making this value for k_2 incorrect! However, this value for k_2 is irrelevant for the other values in the table (do you see why?).

The last column shows the accuracy of the method. Notice that the improved Euler method is indeed much more accurate than the ordinary Euler method; however, even here the method becomes less accurate with successive steps.

Exercise 2. Use the improved Euler method with step size $h = .1$ on the interval $0 \leq x \leq 2$ to find numerical approximations for the same problem as in Exercise 1. Graph the values y_n versus x_n , and compute the accuracy $y(x_n) - y_n$. Are these results more accurate than those of Exercise 1?

→**Hand In:** A print-out of your EXCEL worksheet showing the values of x_n , y_n , k_1 , k_2 , $y(x_n)$, and $y(x_n) - y_n$, and the graph of y_n versus x_n ; also a complete-sentence answer to the question.

III. ERROR ESTIMATES. As we have seen with both the Euler method (E) and the improved Euler method (E^*), errors creep in at each step and accumulate with successive steps to limit the accuracy of the numerical values. The error at each step is called the *local error*, and the accumulated error at the last step is called the *global or cumulative error*. For a given step size h , we would like to estimate these errors.

The local error, $e(h)$, is easy to estimate using Taylor's formula with remainder:

$$y(x) = y(a) + y'(a)(x - a) + \frac{1}{2}y''(c)(x - a)^2 \quad \text{where } c \text{ lies between } x \text{ and } a.$$

Assuming $y(x_n) = y_n$ is accurate, let $a = x_n$, $x = x_{n+1}$, $h = x_{n+1} - x_n$, and $y'(a) = y'(x_n) = f(x_n, y_n)$ to obtain

$$y(x_{n+1}) = y(x_n) + f(x_n, y_n)h + \frac{1}{2}y''(c)h^2.$$

Comparing this with (E), we see that the local error $e(h) = |y(x_{n+1}) - y_{n+1}|$ for Euler's method is given by

$$e(h) \leq Mh^2, \quad \text{where } M = \frac{1}{2} \max\{|y''(c)| : x_n \leq c \leq x_{n+1}\}.$$

Of course, we have no idea how big M is, since it depends on $y''(x)$ which we do not know in practice. But, as an estimate of how the accuracy depends on the step size h , this means that the local error goes to zero like h^2 , as $h \rightarrow 0$. This is sometimes written as

$$e(h) = O(h^2) \quad \text{as } h \rightarrow 0.$$

The global error, $E(h)$, is of course more important than the local error since it measures the accuracy of our final numerical answer, say at $x = b$ where $b > a$. Of

course, it is more complicated to estimate, because the local error at each step can be different. One way around this is to assume the constant M above works for every n :

$$M = \frac{1}{2} \max\{|y''(x)| : a \leq x \leq b\}.$$

The number of steps N is related to the step size h by $N = Lh^{-1}$, where $L = b - a$ is the total length of the x -values. (In Example 1, $L = 1$ and $N = 10$.) Since the local errors accumulate N times, we find that

$$E(h) \leq Ne(h) = Lh^{-1}e(h) \leq Lh^{-1}Mh^2 = LMh;$$

without trying to compute M , we can at least conclude that

$$E(h) = O(h) \quad \text{as } h \rightarrow 0.$$

This is expressed by saying that Euler's method is a *first-order method*.

Let us illustrate this error analysis with our example.

Example 1 (cont'd). Compare the local error and global error for various step sizes in the Euler approximations for the solution of $y' = x + y$, $y(0) = 1$, taking $N = 10$, $N = 20$, $N = 30$, $N = 40$, and $N = 50$. Based on these, what errors would you expect for $N = 100$?

1. We have already taken $N = 10$ ($h = .1$), so let us take $N = 20$ ($h = .05$).

(Increase the length of the "n" column to end with $n = 20$. Then revise the "xn" column to reflect $x_{n+1} = x_n + .05$, and the "yn" column to reflect $y_{n+1} = y_n + .05f(x_n, y_n)$; in both cases, you may change the first entry, and then "fill down". We shall also need to "fill down" the three remaining columns to fill the table.)

An abbreviated version of your table is the following:

n	xn	yn	$sn = f(xn, yn)$	$y(xn)$	$y(xn) - yn$
0	0	1	1	1	0
1	0.05	1.05	1.1	1.052542	0.002542
19	0.95	3.103900	4.053900	3.221419	0.117519
20	1	3.306595	4.306595	3.436564	0.129968

The important information for our purposes is the error $y(xn) - yn$ for $n = 1$, which represents the local error in the first step, and $y(xn) - yn$ for $n = 20$ (which represents the global error). In other words, from this and our previous Euler approximation with $N = 10$, we find the following values for $e(h)$ and $E(h)$: $e(0.1) = 0.010342$, $e(0.05) = 0.002542$, $E(0.1) = 0.249079$, and $E(0.05) = 0.1299682$.

2. Now let us repeat the same procedure with $N = 30$, $N = 40$, and $N = 50$; for $N = 30$, we are careful to use $h = 1/30$ and not the approximation $h = 0.033$. Let

us record the results in the following table:

N	h	$e(h)$	$E(h)$
10	0.1	0.010342	0.249079
20	0.05	0.002542	0.129968
30	0.033	0.001157	0.091426
40	0.025	0.000630	0.066436
50	0.02	0.000403	0.053388

3. What sort of error might we expect for $N = 100$? Let us suppose $E(h) \approx Kh$ for some constant K . Then we expect $E(\frac{h}{2}) \approx K\frac{h}{2} \approx \frac{1}{2}E(h)$, i.e. halving the step size h should roughly halve the global error. Since increasing N from 50 to 100 involves halving $h = .02$ to $h = .01$, we expect for $N = 100$ to have $E(0.01) \approx \frac{1}{2}E(0.02) \approx 0.026694$.

4. To check this error estimate we can actually compute $E(0.01)$, using EXCEL as before. We find $E(0.01) = 0.026936$. Not too bad!

Exercise 3. In Exercise 1, you used the Euler method for the initial-value problem $y' - 2y = 3e^{2x}$, $y(0) = 0$, with $h = 0.1$ ($N = 20$) and $h = 0.05$ ($N = 40$). Now take $h = 0.033$ ($N = 60$) and $h = 0.025$ ($N = 80$), and fill in the following table.

N	h	$e(h)$	$E(h)$
20	0.1		
40	0.05		
60	0.033		
80	0.025		

On the basis of this and the fact that Euler's method is a first-order method, approximate the global error $E(0.01)$ ($N=200$) *without* calculating the approximate value of $y(2)$ for $h = 0.01$.

→**Hand In:** A copy of the filled-in table, and your estimate for $E(0.01)$ **with** an explanation of why and how you obtained it.

Of course, you will now want to know what sort of error estimates pertain for the improved Euler method. To distinguish from the Euler method, let us denote the local and global errors for the improved Euler method by $e^*(h)$ and $E^*(h)$ respectively (this is not standardized notation!) Let us proceed first numerically, by generating the local error and global error for our example.

Example 1 (cont'd). Compare the local error and global error for various step sizes in the improved Euler approximations of the solution of $y' = x + y$, $y(0) = 1$, taking $N = 10$, $N = 20$, $N = 30$, $N = 40$, and $N = 50$. Based on these, what errors would you expect for $N = 100$?

1. From our previous work, we know $e^*(0.1) = 0.000342$ and $E^*(0.1) = 0.008402$ when $N = 10$. If we repeat the procedure for $N = 20$, $N = 30$, $N = 40$, and $N = 50$, we

generate the following table.

N	h	$e^*(h)$	$E^*(h)$
10	0.1	0.000342	0.008402
20	0.05	0.000042	0.002182
30	0.033	0.000012	0.000982
40	0.025	0.000005	0.000556
50	0.02	0.0000026	0.000357

2. How does the global error $E^*(h)$ change with h ? Between $N = 10$ and $N = 20$, we have divided the step size from $h = 0.1$ to 0.05 , but the global error has been roughly divided by 4! Similarly, from $N = 20$ to $N = 40$, the step size has been divided by 2, but the global error is roughly divided by 4! The pattern seems to be $E^*(\frac{h}{2}) \approx \frac{1}{4}E^*(h)$, which suggests $E^*(h) = O(h^2)$. Based on this, we would expect for $N = 100$ to have $E^*(0.01) \approx \frac{1}{4}E^*(0.02) = 0.000089$.
3. Of course, we know how to check this error estimate, by computing for $N = 100$ that $E^*(0.01) = 0.000090$. Again, pretty darn good!

Our experience with computing the error for the improved Euler method for Example 1 leads us to expect that the global error behaves like h^2 :

$$E^*(h) = O(h^2) \quad \text{as } h \rightarrow 0.$$

This in turn suggests that the local error behaves like h^3 :

$$e^*(h) = O(h^3) \quad \text{as } h \rightarrow 0.$$

Indeed, it is not difficult to use Taylor's formula with remainder to prove this behavior for the local and global errors. The property $E^*(h) = O(h^2)$ is expressed by saying that the improved Euler method is a *second-order method*.

Exercise 4. In Exercise 2, you used the improved Euler method for the initial-value problem $y' - 2y = 3e^{2x}$, $y(0) = 0$, with $h = 0.1$ ($N = 20$). Now take $h = 0.05$ ($N = 40$), $h = 0.033$ ($N = 60$), and $h = 0.025$ ($N = 80$), and fill in the following table.

N	h	$e^*(h)$	$E^*(h)$
20	0.1		
40	0.05		
60	0.033		
80	0.025		

On the basis of this and the fact that the improved Euler's method is a second-order method, approximate the global error $E^*(0.01)$ ($N = 200$) *without* calculating the approximate value of $y(2)$ for $h = 0.01$.

→**Hand In:** A copy of the filled-in table, and your estimate for $E^*(0.01)$ **with** an explanation of why and how you obtained it.