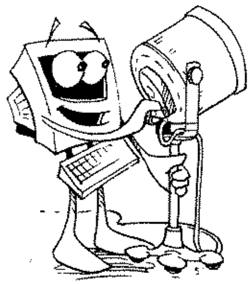
Spotlight on Approximate Numerical Solutions



Reference: Sections 2.3 and 2.4.

From the Existence and Uniqueness Theorem 2.3.1 we know that the IVP

$$y' = f(t, y), \qquad y(t_0) = y_0 \tag{1}$$

has a unique solution $y(t)$ on an interval containing t_0 if the rate function $f(t, y)$ and its derivative $\partial f/\partial y$ are continuous in some rectangle in the ty-plane that contains the initial point (t_0, y_0). How do we go about describing this solution? The collection of rate functions $f(t, y)$ for which we can find a solution formula for IVP (1) is remarkably small, so the solution formula approach is not often a realistic option. Even when we can find a solution formula, it is not always very informative. Lacking a solution formula, how do we describe the solution? We present some basic numerical procedures for finding approximate values for $y(t)$ at a discrete set of times near t_0.

At last we'll see how some numerical solvers do their work!

Euler's Method

The direction field approach used in Section 2.4 to characterize solution curves of a first-order ODE suggests techniques for finding approximate numerical solutions for IVP (1). Euler's Method is the simplest of these approximation methods.

Say we wish to approximate the value of the solution of IVP (1) at some future time T. First, we partition the interval $t_0 \leq t \leq T$ with N *steps* of equal *step size* h:

$$t_n = t_0 + nh, \qquad n = 0, 1, 2, \ldots, N$$
$$h = (T - t_0)/N$$

We know that (t_0, y_0) is on the solution curve. To find an approximation to $y(t_1)$, just follow the tangent line to the solution curve through (t_0, y_0) until $t = t_1$. Since the slope of the tangent line to the solution curve at (t_0, y_0) is $f(t_0, y_0)$, we see that

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

is a reasonable approximation to $y(t_1)$ if h is small. Using (t_1, y_1) as a base point and pretending that (t_1, y_1) is on the desired solution curve, we construct an approximation y_2 to $y(t_2)$ in the same way:

$$y_2 = y_1 + hf(t_1, y_1)$$

Since (t_1, y_1) is most likely not on the desired solution curve, the calculated value y_2 also acquires an error from this source. We can repeat this calculation N times to produce an approximation y_N to the value $y(T)$ of the true solution of IVP (1) at $t = T$.

Euler's Method For the IVP $y' = f(t, y)$, $y(t_0) = y_0$, the recursive scheme

$$y_n = y_{n-1} + hf(t_{n-1}, y_{n-1}), \qquad t_n = t_{n-1} + h, \quad 1 \leq n \leq N \tag{2}$$

is called *Euler's Method* with step size h.

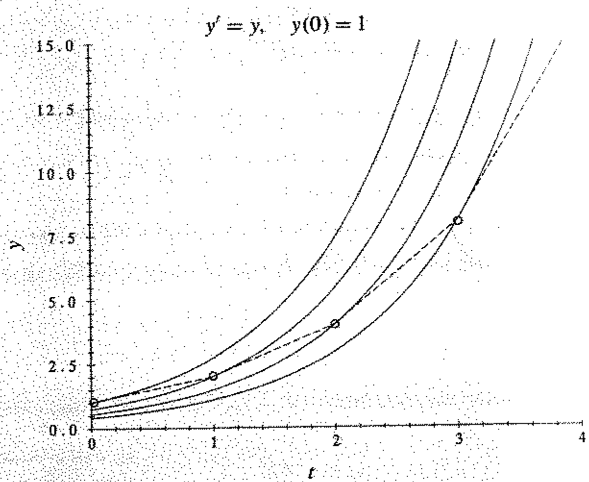


FIGURE 1 True solution curves (solid) through the Euler points (circles); broken-line Euler Solution (dashed) of IVP (3) with $h = 1$ (Example 1).

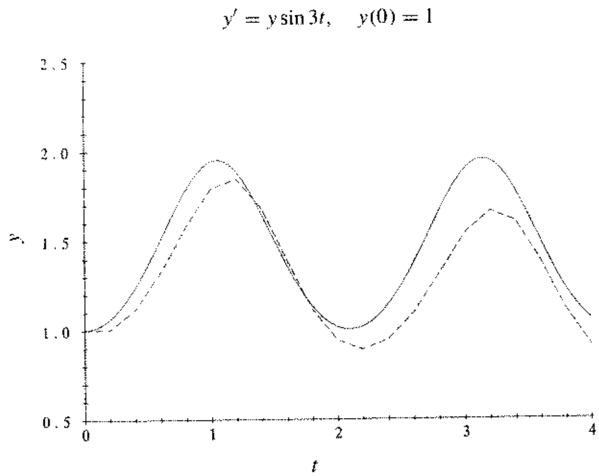


FIGURE 2 The true solution curve (solid) and an Euler Solution (dashed) with $h = 0.2$ (Example 2).

Euler's Method is named for the Swiss mathematician Leonhard Euler.[9] Connecting the *Euler points* $(t_0, y_0), (t_1, y_1), \ldots, (t_N, y_N)$ by line segments produces the *Euler Solution* approximation to the true solution curve of IVP (1).

EXAMPLE 1

A Simple IVP and an Euler Solution

Figure 1 illustrates the geometry of Euler's Method with $h = 1$ for the IVP

$$y' = y, \qquad y(0) = 1 \tag{3}$$

In this case $f(t, y) = y$ so we have

$$y_n = y_{n-1} + hf(t_{n-1}, y_{n-1}) = y_{n-1} + y_{n-1} = 2y_{n-1}, \qquad n = 1, \ldots, N \tag{4}$$

The values y_n produced by Euler's Method (4) differ from the values of the true solution, $y = e^t$. In fact, $y_1 = 2y_0 = 2,\ y_2 = 2y_1 = 4,\ \ldots,\ y_n = 2y_{n-1} = 2^n$, but the true value is $y(n) = e^n \approx (2.71828)^n$; the error $e^n - 2^n$ grows as n increases.

Each solid curve in Figure 1 is the solution curve of the corresponding IVP

$$y' = y, \qquad y(t_n) = y_n, \quad n = 1, 2, 3$$

where y_n is the Euler estimate for e^{t_n}, $n = 1, 2, 3$, and it is run forward and backward from t_n.

Leonhard Euler

[9]Born in Switzerland, Leonhard Euler (1707–1783) was one of the greatest mathematicians of all time. He was also one of the most prolific writers in any field, producing a flood of papers in every area of pure and applied mathematics. His mathematical abilities were immense, leading one physicist to remark that "he calculated without apparent effort, as men breathe, or as eagles sustain themselves in the wind." He was blind the last 17 years of his life, but dictated his seemingly endless flow of new mathematical results until the day he died. The Swiss government is nearing the end of a monumental project to publish all of Euler's work—100 massive volumes so far. Incidentally, his name is pronounced "oiler," not "youler."

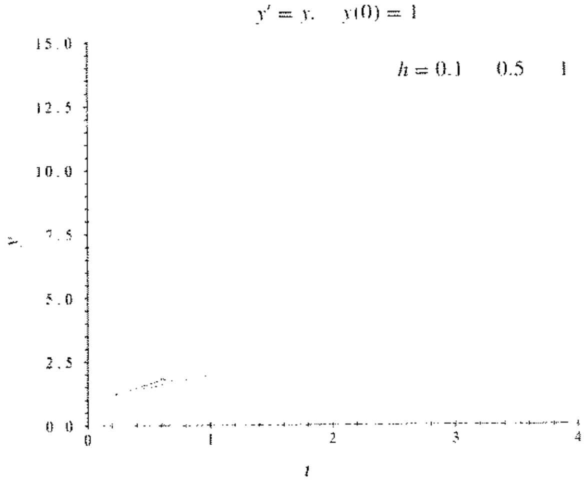


FIGURE 3 True solution (solid) and Euler Solutions (dashed) (Example 3): lower h, better approximation.

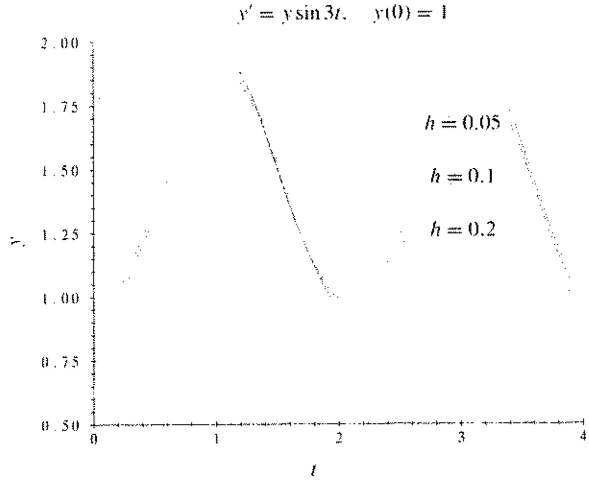


FIGURE 4 True solution (solid) and Euler Solutions (dashed) (Example 3): lower h, better approximation.

The inaccuracy of the Euler Solution in Example 1 is largely due to the choice of the step size h. Let's look at another example.

EXAMPLE 2

An IVP and Euler Solutions

We seek an approximate solution curve for the IVP

$$y' = y \sin 3t, \qquad y(0) = 1, \quad 0 \le t \le 4 \tag{5}$$

Let's take $h = 0.2$ and $N = 20$, so $t_n = (0.2)n$, $n = 0, 1, \ldots, 20$. Then Euler's Method becomes

$$y_n = y_{n-1} + 0.2 y_{n-1} \sin 3t_{n-1}, \qquad n = 1, 2, \ldots, 20, \quad \text{with } y_0 = 1 \tag{6}$$

The linear IVP (5) has the unique solution $y = e^{(1-\cos 3t)/3}$. Figure 2 displays the true solution (solid) and the Euler Solution (dashed). We see again that Euler's Method may provide only a rough approximation to the true solution if h is not small enough.

Now let's see what happens if we take smaller step sizes in Examples 1 and 2.

EXAMPLE 3

Take Smaller Steps and Improve Accuracy

Let's use Euler's Method to approximate solutions of the IVPs in Examples 1 and 2. Figures 3 and 4 show that the smaller the step size, the better the approximation.

"Precise arithmetic" means no rounding or chopping of decimal strings.

In spite of the roughness of the Euler approximation, it can be shown that if f and $\partial f/\partial y$ are continuous (as in these examples) and if precise arithmetic is used, then we can make the error as small as desired by taking the step size sufficiently small. On the other hand, chaos can result if the step size is too large.

One-Step Methods

Euler's Method for producing approximate solutions of IVP (1) is an example of a *one-step method*. Such methods produce an approximate value for the solution of IVP (1) at a selected point T in the following way. Suppose that $T > t_0$. Select an increasing sequence $t_1, t_2, \ldots, t_N$ with $t_N = T$ and $t_1 > t_0$ and define the *step size* $h_n = t_n - t_{n-1}$ at step n for $n = 1, 2, \ldots, N$. For a given y_0 and a given function $A(t, y, h)$, a one-step method computes an approximation y_n to $y(t_n)$ using the *discretization scheme*

$$y_n = y_{n-1} + h_n A(t_{n-1}, y_{n-1}, h_n), \qquad n = 1, 2, \ldots, N \tag{7}$$

To compute y_n, only the value of y_{n-1} is required (hence, the name "one-step method"). Method (7) uses the value y_0 to generate y_1, y_1 to generate y_2, and so on, until the process terminates with the calculation of y_N, which is an approximation of $y(T)$.

The function A is called an *approximate slope function* for $y(t)$ at t_{n-1} because

$$\frac{y_n - y_{n-1}}{t_n - t_{n-1}} = \frac{y_n - y_{n-1}}{h_n} = A(t_{n-1}, y_{n-1}, h_n)$$

where the last equality comes from (7). As we will see, the slope function $f(t_{n-1}, y_{n-1})$ used in Euler's Method is not always the best choice for $A(t_{n-1}, y_{n-1}, h_n)$.

The approximation to the solution of IVP (1) by the discrete one-step method (7) has a simple interpretation. For each $n = 1, \ldots, N$, join the point (t_{n-1}, y_{n-1}) to (t_n, y_n) by a line segment to form a broken-line path from (t_0, y_0) to (t_N, y_N). This path approximates the graph of the solution $y(t)$ (see Figure 5 for a schematic).

Errors

The approximations $y_1, y_2, \ldots, y_N$ a one-step method generates deviate from the exact values $y(t_1), y(t_2), \ldots, y(t_N)$. If precise arithmetic is used, the deviation

$$E_n = |y(t_n) - y_n|, \qquad n = 1, \ldots, N$$

is the *global discretization error* at the nth step.

There is a local version of the error due to discretization. By the time we reach the point (t_{n-1}, y_{n-1}) on the broken-line path of approximation, method (7) has "forgotten" previously computed results. In the next step all we can hope for is to estimate the difference between y_n as given by (7) and $\tilde{y}(t_n)$, which is the value at t_n of the true solution of the IVP

$$\tilde{y}' = f(t, \tilde{y}), \qquad \text{with initial data} \quad \tilde{y}(t_{n-1}) = y_{n-1}$$

The *local discretization error* at step n is the magnitude

$$e_n = |\tilde{y}(t_n) - y_n|, \qquad n = 1, \ldots, N$$

☞ Software designers use the local version of errors to control global errors because it's easier to estimate local errors than global ones.

The global discretization error E_n is affected by the local errors $e_1, \ldots, e_{n-1}$ that produce the inexact values $y_1, \ldots, y_{n-1}$. See Figure 5.

We can classify one-step methods according to the order of magnitude of the global discretization errors incurred when we apply the method.

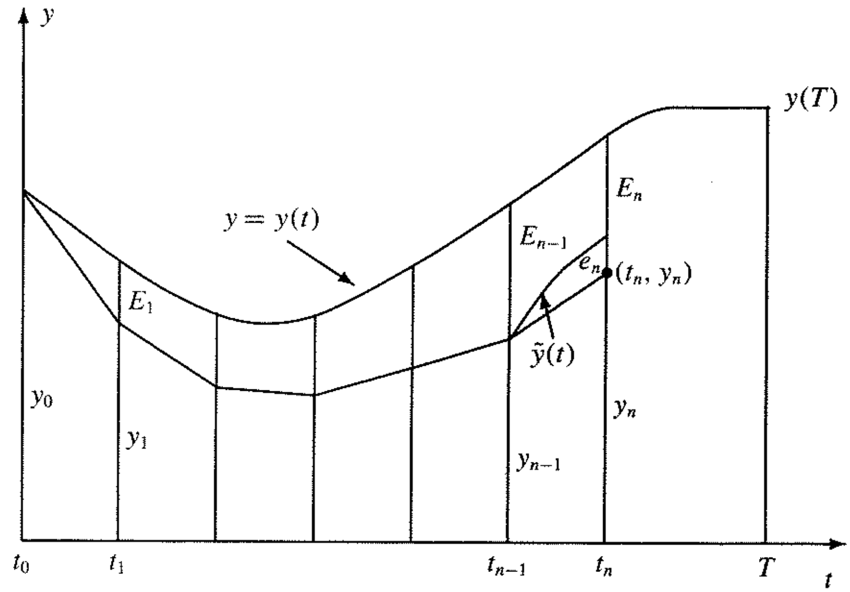


FIGURE 5 Broken-line approximation to true solution: discretization errors.

❖ **Order of a One-Step Method**. Suppose that we approximate the solution of the IVP

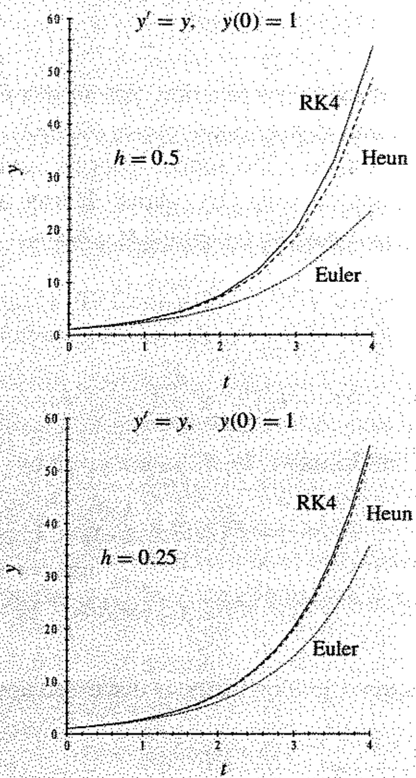
$$y' = f(t, y), \qquad y(t_0) = y_0, \quad t_0 \le t \le T$$

by method (7) with fixed step $h = (T - t_0)/N$. If there are positive constants M and p such that for every N,

$$E_N \le Mh^p$$

then method (7) is of *order* p.

Euler's Method is a first-order method (i.e., $p = 1$). Note that whatever the order p of a method, the bound on E_n decreases with the step size h. It is often easy to adjust the step size when implementing a one-step method on a computer. For a fourth-order method (i.e., $p = 4$), cutting the step size in half results in a 16-fold drop in the upper bound on E_N, since $(h/2)^4 = h^4/16$. But halving the step size of a first-order method ($p = 1$) gives only a two-fold decrease in the upper bound. This suggests that the higher the order of a method, the more accurately it will approximate the solution of IVP (1). However, in specific instances this may not hold true since the constant M may be larger for a higher-order method than for a lower-order algorithm. Also, higher-order methods usually involve more calculations and function evaluations, so the accompanying round-off errors may undo the advantages of the higher order. Still, the order of a method is a good indication of its accuracy. Here are some higher-order methods of approximation that are widely used.



Time	Euler	Heun	RK4	exp(t)
0.0	1.0000	1.0000	1.0000	1.0000
0.5	1.5000	1.6250	1.6484	1.6487
1.0	2.2500	2.6406	2.7173	2.7183
1.5	3.3750	4.2910	4.4794	4.4817
2.0	5.0625	6.9729	7.3840	7.3891
2.5	7.5938	11.3310	12.1720	12.1825
3.0	11.3906	18.4123	20.0643	20.0855
3.5	17.0859	29.9208	33.0756	33.1155
4.0	25.6289	48.6213	54.5230	54.5982

Time	Euler	Heun	RK4	exp(t)
0.00	1.0000	1.0000	1.0000	1.0000
0.25	1.2500	1.2813	1.2840	1.2840
0.50	1.5625	1.6416	1.6487	1.6487
0.75	1.9531	2.1033	2.1170	2.1170
1.00	2.4414	2.6949	2.7182	2.7183
1.25	3.0518	3.4528	3.4902	3.4903
1.50	3.8147	4.4239	4.4815	4.4817
1.75	4.7684	5.6681	5.7543	5.7546
2.00	5.9605	7.2623	7.3887	7.3891
2.25	7.4506	9.3048	9.4872	9.4877
2.50	9.3132	11.9217	12.1817	12.1825
2.75	11.6415	15.2747	15.6415	15.6426
3.00	14.5519	19.5707	20.0839	20.0855
3.25	18.1899	25.0749	25.7881	25.7903
3.50	22.7374	32.1273	33.1124	33.1155
3.75	28.4217	41.1631	42.5169	42.5211
4.00	35.5271	52.7402	54.5924	54.5982

FIGURE 6 Approximate solutions of $y' = y$, $y(0) = 1$ (Example 4).

Heun's Method

We can convert Euler's Method to a second-order method if we compute the approximate slope function by averaging the slopes at (t_{n-1}, y_{n-1}) and at (t_n, y_n). We use Euler's Method to find a first approximation to y_n so that we can calculate a slope at (t_n, y_n). This leads to the following method:

☞ Also called Improved Euler's Method.

❖ **Heun's Method.** *Heun's Method* [10] is a second-order, one-step method with constant step size h for the IVP $y' = f(t, y),\ y(t_0) = y_0$. The formula is

$$y_n = y_{n-1} + \frac{h}{2}[f(t_{n-1}, y_{n-1}) + f(t_n, y_{n-1} + hf(t_{n-1}, y_{n-1}))], \quad n = 1, \ldots, N \tag{8}$$

[10] Karl Heun (1859–1929) worked in classical mechanics and applied mathematics.

Runge–Kutta Methods

One-step algorithms that use averages of the slope function $f(t, y)$ at two or more points over the interval $[t_{n-1}, t_n]$ to calculate y_n are *Runge–Kutta Methods*.[11] (Heun's method is a second-order Runge–Kutta Method.) The fourth-order method given below is the most widely used of any of the one-step algorithms. It uses a weighted average of slopes at the midpoint $t_{n-1} + h/2$ and the endpoints t_{n-1} and t_n.

❖ **Fourth-Order Runge–Kutta Method (RK4).** For the IVP $y' = f(t, y)$, $y(t_0) = y_0$, the *Fourth-Order Runge-Kutta Method* is the one-step method

$$y_n = y_{n-1} + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \quad n = 1, \ldots, N \tag{9}$$

where h is fixed, $t_n = t_{n-1} + h$, and the slopes k_1, k_2, k_3, k_4 are given by

$$k_1 = f(t_{n-1}, y_{n-1}), \qquad k_2 = f\left(t_{n-1} + \frac{h}{2}, y_{n-1} + \frac{h}{2}k_1\right)$$

$$k_3 = f\left(t_{n-1} + \frac{h}{2}, y_{n-1} + \frac{h}{2}k_2\right), \qquad k_4 = f(t_n, y_{n-1} + hk_3)$$

The next example compares the three approximation methods discussed above.

EXAMPLE 4

Comparison of Numerical Methods

The initial value problem $y' = y$, $y(0) = 1$, has the unique solution $y = e^t$. Approximations that use the Euler, Heun, and RK4 methods are plotted in Figure 6 on the interval $0 \le t \le 4$. As expected, the higher-order methods give more accuracy than those of lower order. Accuracy improves as the step size is reduced from 0.5 to 0.25.

Computing the values listed in Figure 6 by hand from the Euler, Heun and RK4 algorithms can be a very laborious process. However, with the ready availability of computer resources there is no need to go to all that trouble. Most solver packages contain several algorithms for producing approximating solutions of IVPs which the user can select at will. For example, ODE Architect has four solvers with Euler's Method among them. To produce the Euler solutions in Figure 6 proceed as follows: enter the IVP and fix the scale in the upper graphics screen to $0 \le t \le 4$, $0 \le y \le 30$. Then in the initial condition panel set the step size h by inserting the correct number in **# Points** (8, for $h = 0.5$, and 16 for $h = 0.25$) and the solve-time 4. Depress the **Solver** tab beneath the initial condition panel and select **Euler** and then hit the **IC** tab. Depressing the **Solve** button produces the Euler solution. The **Data** tab under the top graphics screen shows the values in the table. A spread sheet package would have worked just as well.

C. D. T. Runge

[11] The German applied mathematician C. D. T. Runge (1856–1927) did notable work in numerical analysis ar diophantine equations. M. W. Kutta (1867–1944) was a German applied mathematician who contributed to ti early theory of airfoils.

See the Student Resource Manual for the extension of Euler's Method and RK4 to planar systems.

Looking Back

There are many methods for finding approximate numerical solutions of IVPs.[12] Euler's Method has the virtue of being simple to visualize and implement, but it is not the most practical choice for numerical solvers. RK4 provides a combination of accuracy and efficiency that makes it an excellent choice for a one-step method.

Many commercial solvers use multistep methods, where the approximate slope function at each step is computed from the slopes at several of the previously computed solution points. Some solvers are adaptive, adjusting the step size and the numerical method automatically to meet the immediate computational needs. Some solvers (ODE Architect, for example) allow the user to select a particular method from a list of methods. The numerical solver we use is based on LSODA, an adaptive multistep method whose origins date back to C. W. Gear's DIFFSUB and ODEPACK developed by Alan Hindmarsh at Lawrence Livermore National Laboratories.

☞ So when we label a curve as the "true" solution curve, we mean that it was created by LSODA.

See the SPOTLIGHT ON COMPUTER IMPLEMENTATION for tips on how to make the most of your solver, no matter what numerical algorithm it uses.

PROBLEMS

Comparison of Numerical Methods.

1. Use the indicated method to estimate $y(1)$ if $y' = -y$, $y(0) = 1$, $h = 0.1$. Plot the broken-line approximation.

   (a) Euler's Method **(b)** RK4

2. Use the indicated method to estimate $y(1)$ if $y' = -y$, $y(0) = 1$, $h = 0.01, 0.001, 0.0001$, and plot the approximating polygons. Zoom in on part of plot if necessary to separate the polygons.

   (a) Euler's Method **(b)** RK4

3. Use the indicated method to estimate $y(1)$ for the IVP $y' = -y^3 + t^2$, $y(0) = 0$, using the step sizes $h = 0.1, 0.01, 0.001$. Plot the broken-line approximation.

   (a) Euler's Method **(b)** RK4

4. *Convergence of Euler Approximations* For each of the following IVPs, show that the Euler approximation $y_N(T) \to y(T)$ as $N \to \infty$ ($T > 0$ is fixed).

   (a) $y' = 2y$, $y(0) = 1$ **(b)** $y' = -y$, $y(0) = 1$

5. *RK4 and Simpson's Formula* Simpson's formula for approximating an integral is

$$\int_a^b f(t)\,dt \approx \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]$$

   Show that RK4 for the IVP $y' = f(t)$, $y(t_0) = y_0$, gives Simpson's formula at each step.

[12]For a good source on approximation methods for IVPs, see the book by the American numerical analyst L. F. Shampine, *Numerical Solution of Ordinary Differential Equations* (New York: Chapman & Hall, 1994).