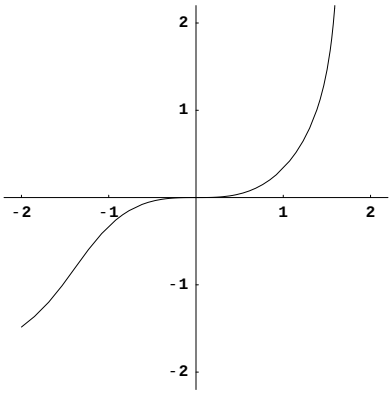
Euler's method

On September 8 we discussed the initial-value problem

$$\frac{dy}{dt} = y^3 + t^2, \quad y(0) = 0.$$

Here's the graph of its solution.



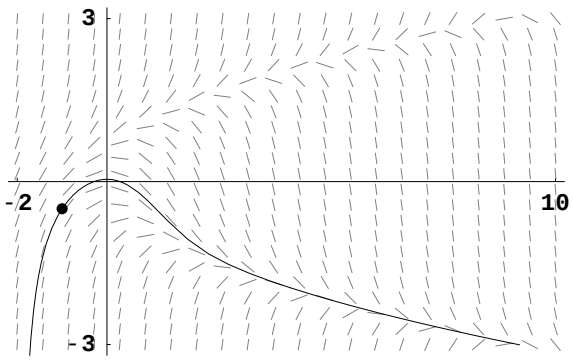
There isn't a nice formula for this function, so how do we obtain its graph? The answer is that we use a numerical algorithm to obtain an approximate solution.

Today we study the numerical algorithm known as Euler's method. It is the most basic of all of the numerical algorithms that are used to approximate solutions to differential equations. Let's start with an example to get an idea of how the method works.

Example. Consider the initial-value problem

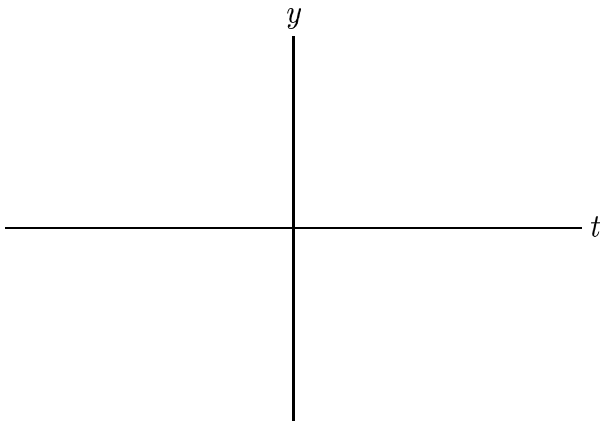
$$\frac{dy}{dt} = y^2 - t, \quad y(-1) = -\tfrac{1}{2}.$$

First, let's see what `HPGSolver` produces:

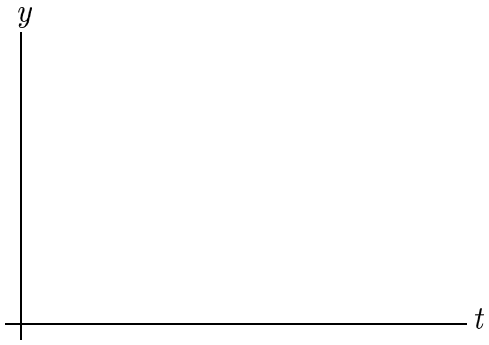


Now let's see what happens when we use Euler's method to approximate the solution with a step size of $\Delta t = 0.5$. We'll use the `EulersMethod` tool from `DETools`.

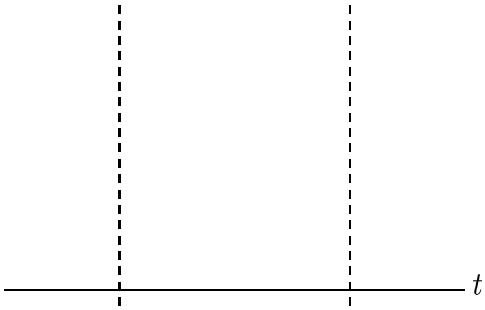
k	t_k	y_k	m_k
0	-1	-0.50	
1			
2			
3			
4			
5			
6			

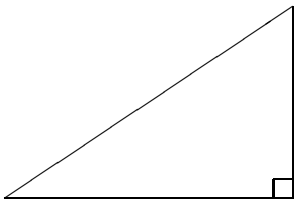


Here is a general picture of the algorithm and the associated notation:



Let's look more closely at the kth step and the key triangle:





These observations yield Euler's method:

Euler's method is easy to program—even with just a spreadsheet.

	A	B	C	D	E	F	G
0	-1	-0.5		0.5			
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							

There are two spreadsheets corresponding to this example posted on the course web site—one that uses a lot of the defaults in Excel and one that has been customized for the particular example we are discussing. The second spreadsheet has a slider for Δt that illustrates what happens when different values of Δt are used.

The Error in Euler's method

A precise analysis of the error involved in Euler's method is somewhat complicated, but we can learn a lot about the method by studying an initial-value problem for which we know the exact solution

Example. Consider the initial-value problem

$$\frac{dy}{dt} = y + 1, \quad y(0) = 0.$$

By separating variables we can derive the solution

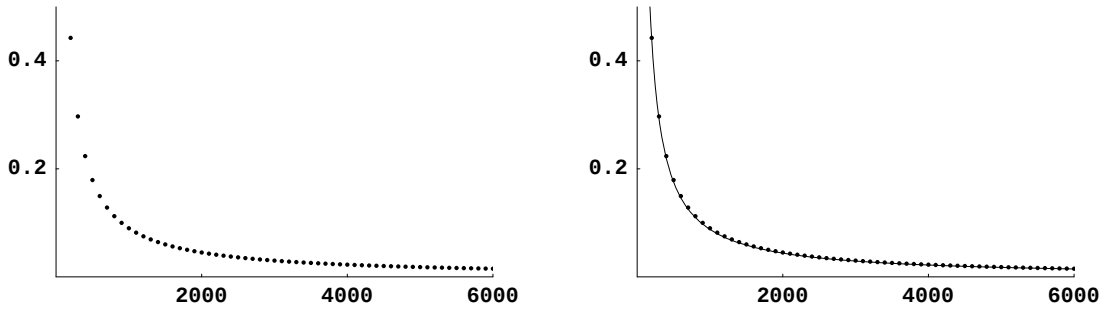
$$y(t) = e^t - 1.$$

Let's see how the error involved in the method behaves as we decrease the step size. We use the interval $0 \le t \le 3$. Note that the exact value of the solution at $t = 3$ is $y(3) = e^3 - 1 \approx 19.0855$. In the table below, n is the number of subdivisions, and consequently $\Delta t = 3/n$. The number y_n is the result of Euler's method for $t = 3$, and the error e_n is the difference between the actual value $y(3)$ and the approximation y_n.

n	y_n	error e_n
1000	18.9955	0.0900023
2000	19.0404	0.0450966
3000	19.0555	0.0300857
4000	19.0630	0.0225722
5000	19.0675	0.0180616
6000	19.0705	0.0150535

We repeat this calculation using $n = 100, 200, \ldots, 6000$, and we plot the error as a function of the number n of subdivisions on the left below. On the right we plot the same points along with the graph of

$$\frac{87.6474}{n}.$$



To see why this result is typical, let's carry out an analysis of the error in general. To fix notation, we consider the initial-value problem

$$\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0.$$

The number n is the number of subdivisions, and therefore

$$\Delta t = \frac{t_n - t_0}{n}.$$

The number y_k is the result of Euler's method after k steps (at $t = t_k$).

The error in the first step:

The error in the second step:

Note that the analysis of the error in this step is complicated by the fact that the point (t_1, y_1) is not necessarily on the graph of the solution $y(t)$.

The error in the kth step ($k \geq 2$):

In general we obtain a recursive formula for the error e_k in terms of the error e_{k-1}. We have

$$e_k \leq (1 + M_2 \Delta t) e_{k-1} + M_1 \frac{(\Delta t)^2}{2}.$$

In Exercise 11 of Section 7.1, we show that this recursive formula yields the following theorem.

Theorem. Given the bounds M_1 and M_2 as above, then the error

$$e_n \leq (C)(\Delta t),$$

where C is a constant that is determined by M_1, M_2, and the length of the interval over which the solution is approximated.

Note that the constant C does not depend on the number of steps used. Another way to write this result is as

$$e_n \leq (C)(\Delta t) = (C)\left(\frac{t_n - t_0}{n}\right) = \frac{K}{n}.$$

Euler's method is the most basic "fixed-step-size" algorithm for numerically approximating solutions. `HPGSolver` also uses a fixed-step-size algorithm called the Runge-Kutta method. The Runge-Kutta method is usually more efficient and more accurate than Euler's method (see Section 7.3 of our text). Unfortunately, there are differential equations that are not amenable to fixed-step-size algorithms.

Example. Consider the initial-value problem

$$\frac{dy}{dt} = e^t \sin y, \quad y(0) = 5.$$

Let's see what happens when we use Euler's method to approximate the solution with various step sizes $0.01 \leq \Delta t \leq 0.1$.

The spreadsheet for this example is also posted on the course web site.