1.5 Euler's method

sec:Euler

For "generic" ODEs, there is no hope of writing down an explicit formula for a typical solution. Fortunately, we can use computers in order to obtain numerical approximations of solutions.

Suppose we have a differential equation of the form

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

We can compare the values of a solution y at times t and $t + \Delta t$ by integrating – the Fundamental Theorem of Calculus tells us that

$$y(t + \Delta t) = y(t) + \int_t^{t+\Delta t} f(\tau, y(\tau))\, d\tau. \tag{1.5.1}$$

euler:FTC

Suppose now that we know the value $y(t)$, but do not know the value of the function y for times larger than t. If Δt is not too large, then we can approximate the integral in (1.5.1) using the left endpoint approximation

$$\int_t^{t+\Delta t} f(\tau, y(\tau))\, d\tau \approx f(t, y(t))\, \Delta t; \tag{1.5.2}$$

euler:left-endpoint-approximation

see Figure 1.5.1.

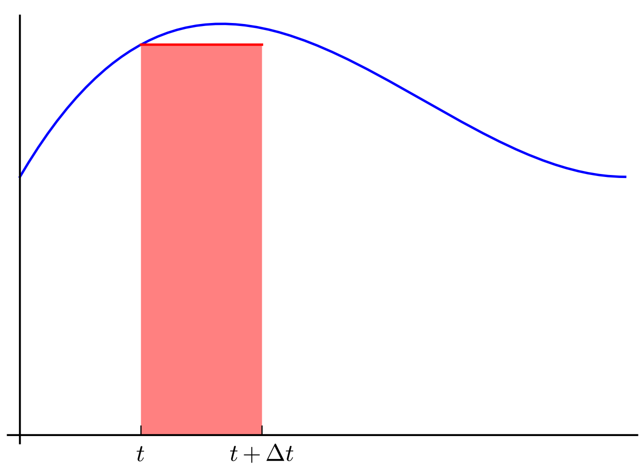

fig:left-endpoint-integration

Figure 1.5.1: If Δt is small then the approximation (1.5.2) is "reasonably good."

Using (1.5.2) and (1.5.1) we obtain the approximation

$$y(t + \Delta t) \approx y(t) + f(t, y(t))\, \Delta t, \tag{1.5.3}$$

euler:main-approximation

which is "reasonably good" if Δt is small.

We now show how to iteratively use the approximation (1.5.3) in order to construct a numerical approximation of the solution to the initial value problem

$$\frac{dy}{dt} = f(t, y) \qquad y(t_0) = y_0. \tag{1.5.4}$$

euler:ivp

We begin by choosing a fixed number Δt and constructing a list of timest_k, indexed by counter k, that each differ by Δt:

k	t_k
0	t_0
1	$t_1 = t_0 + \Delta t$
2	$t_2 = t_0 + 2(\Delta t)$
3	$t_3 = t_0 + 3(\Delta t)$
$\vdots$	$\vdots$
k	$t_k = t_0 + k(\Delta t)$

Notice that for each k we have $t_{k+1} = t_k + \Delta t$.

We now construct a list of values y_k that approximate the values of the solution $y(t)$ to (1.5.4) at the times t_k. The initial condition in (1.5.4) provides the value y_0. The approximation (1.5.3) states that

$$y(t_1) \approx y_0 + f(t_0, y_0)\Delta t.$$

Thus we define

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

Similarly, (1.5.3) implies that

$$y(t_2) \approx y(t_1) + f(t_1, y(t_1))\Delta t.$$

We don't know what $y(t_1)$ is, but we do know that $y(t_1) \approx y_1$. Using this approximation we have

$$y(t_2) \approx y_1 + f(t_1, y_1)\Delta t.$$

Thus we set

$$y_2 = y_1 + f(t_1, y_1)\Delta t.$$

Repeating this procedure, we see that the numbers y_k, defined by

$$y_k = y_{k-1} + f(t_{k-1}, y_{k-1})\Delta t, \tag{1.5.5}$$

euler:euler

approximate the value of $y(t_k)$. The approximation scheme (1.5.5) is called ***Euler's method***.

We can summarize this in the following table:

k	t_k	y_k
0	t_0	y_0
1	$t_1 = t_0 + \Delta t$	$y_1 = y_0 + f(t_0, y_0)\Delta t$
2	$t_2 = t_0 + 2(\Delta t)$	$y_2 = y_1 + f(t_1, y_1)\Delta t$
3	$t_3 = t_0 + 3(\Delta t)$	$y_3 = y_2 + f(t_2, y_2)\Delta t$
$\vdots$	$\vdots$	
k	$t_k = t_0 + k(\Delta t)$	$y_k = y_{k-1} + f(t_{k-1}, y_{k-1})\Delta t$

Let's now look at how this approximation works for an equation for which we know the exact solution.

ex:first-euler

Example 1.5.1. *Consider the initial value problem*

$$\frac{dy}{dt} = y^2 \qquad y(0) = 1.$$

We know that the solution to the initial value problem is

$$y(t) = \frac{1}{1-t}. \tag{1.5.6}$$

Let's now construct an approximation using Euler's method.

We choose $\Delta t = 0.1$ *and decide to approximation the solution* $y(t)$ *on the interval* $0 \leq t \leq 0.5$. *Since* $t_0 = 0$, *we are interested in the times*

k	t_k
0	$t_0 = 0$
1	$t_1 = 0.1$
2	$t_2 = 0.2$
3	$t_3 = 0.3$
4	$t_4 = 0.4$
5	$t_5 = 0.5$

We now use (1.5.5) *in order to construct the list of approximate values. Since* $f(t, y) = y^2$ *and* $y_0 = 1$, *we compute*

$$y_1 = y_0 + f(t_0, y_0)\Delta t = 1 + (0.1)^2(0.1) = 1.001.$$

Proceeding iteratively, we obtain the following table:

k	t_k	y_k
0	$t_0 = 0$	$y_0 = 1$
1	$t_1 = 0.1$	$y_1 = 1.100$
2	$t_2 = 0.2$	$y_2 = 1.221$
3	$t_3 = 0.3$	$y_3 = 1.370$
4	$t_4 = 0.4$	$y_4 = 1.558$
5	$t_5 = 0.5$	$y_5 = 1.800$

A plot of the points (t_k, y_k), *along with a plot of the solution, appears in Figure 1.5.2.*

As evidenced by the plot in Figure 1.5.2, the approximation scheme with $\Delta t = 0.1$ *is not bad, but is not great. We can obtain a better approximation by choosing* $\Delta t = 0.01$, *in which case we must plot* 50 *points. The comparison between this better approximation and the actual solution appears in Figure 1.5.3.*

We can easily make Sage do the computations in Euler's method. Here I explain how to build the code used in Example 1.5.1. First we define variable y and also the function $f(y) = y^2$ that determines the right hand side:

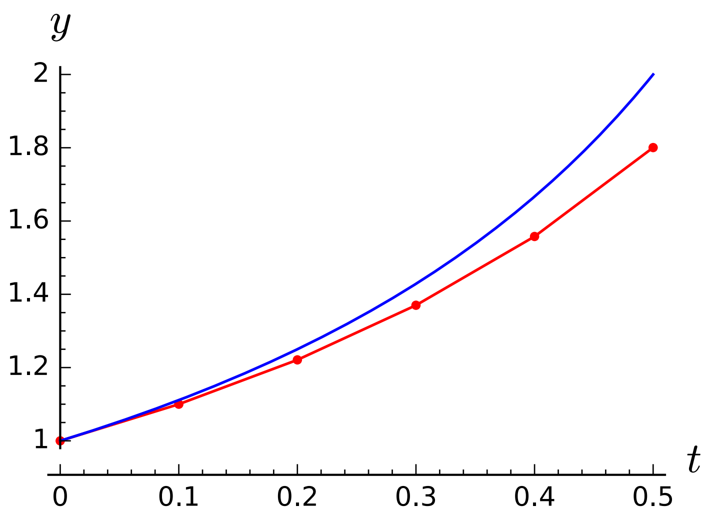

Figure 1.5.2: A plot of the approximating points (t_k, y_k), as computed with $\Delta t = 0.1$, in red and a plot of the actual solution $y(t)$ in blue. Notice that the error accumulates as time progresses.

fig:euler-example1

```
var('y')
f(y) = y^2
```

Then we define Δt and set it equal to 0.1. We also define y_0, which we set equal to 1:

```
var('y')
f(y) = y^2

y0 = 1
deltaT = 0.1
```

Next we construct a data structure that will hold the various values of t_k and y_k. To do this we make a variable called `steps` that tells us how many time steps we will take. In this example, we set `steps` equal to 5. We call that data structure `eulerData`. For now, we define all the t_k values to be $t_k = k(\Delta t)$ and all the y_k values to be the same as y_0.

```
var('y')
f(y) = y^2

y0 = 1
deltaT = 0.1
steps=5

eulerData = [[k*deltaT,y0] for k in range(0,steps+1)]
```

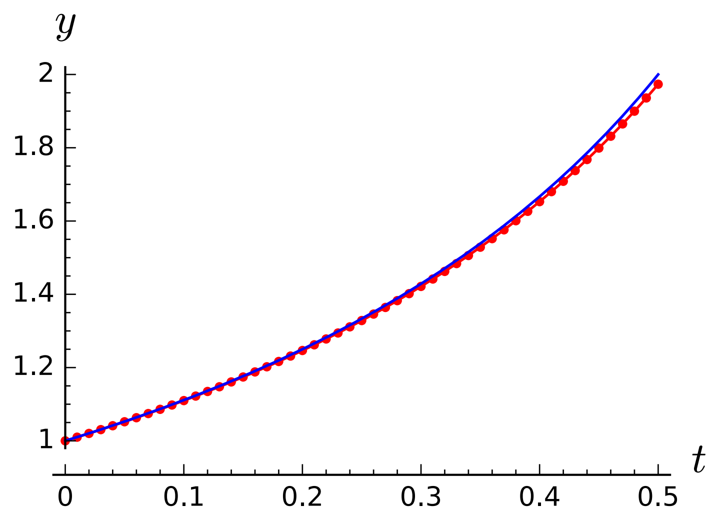

Figure 1.5.3: A plot of the approximating points (t_k, y_k) in red, now with $\Delta t = 0.01$, together with a plot of the actual solution $y(t)$ in blue. Notice that the approximation here is much better than the approximation in Figure 1.5.2.

fig:euler-example2

```
eulerData
```

The variable `eulerData` is organized as follows: the quantity `eulerData[2]` tells us the entries in the row corresponding to $k = 2$.

```
var('y')
f(y) = y^2

y0 = 1
deltaT = 0.1
steps=5

eulerData = [[k*deltaT,y0] for k in range(0,steps+1)]

eulerData[2]
```

If we want only to see the time t_2, then we need to ask Sage to show us `eulerData[2][0]`, while if we want Sage to show us y_2, we need to ask to be shown `eulerData[2][1]`. Try it:

```
var('y')
f(y) = y^2

y0 = 1
deltaT = 0.1
steps=5

eulerData = [[k*deltaT,y0] for k in range(0,steps+1)]
```

```
eulerData[2][0]
```

What we want to do now is systematically go through and update all the y_k values, starting at $k = 1$ and ending at $k =$ `steps`. We can do this with a loop:

```
var('y')
f(y) = y^2

y0 = 1
deltaT = 0.1
steps=5

eulerData = [[k*deltaT,y0] for k in range(0,steps+1)]

for k in [1..steps]:
    eulerData[k][1]=eulerData[k-1][1]+deltaT*f(eulerData[k
        -1][1])

eulerData
```

Finally, we can plot the resulting approximate solution:

```
var('y')
f(y) = y^2

y0 = 1
deltaT = 0.1
steps=5

eulerData = [[k*deltaT,y0] for k in range(0,steps+1)]

for k in [1..steps]:
    eulerData[k][1]= eulerData[k-1][1] + deltaT*f(eulerData[k
        -1][1])

eulerPlot = list_plot(eulerData, color="red", plotjoined=true,
    marker='.',axes_labels=['$t$','$y$'])
eulerPlot.show()
```

Activity 1.5.1. *Use Euler's method to construct an approximate solution to the initial value problem*

$$\frac{dy}{dt} = 1 - y, \qquad y(0) = 3$$

where $\Delta t = 0.5$. *Do this by hand, so that you get a sense of how the method works.*

In principle, you can code up Euler's method by hand (as is done above) any time you need a numerical approximation for a differential equation. However, there are much better algorithms available for doing this. The "industry standard" algorithm is a procedure called *Runge-Kutta*. Sage has a built-in Runge-Kutta function. The inputs to the Runge-Kutta function are

- the right side of the differential equation ($f(y)$),
- the unknown variable (y),
- the initial conditions (t_0, y_0),
- the independent variable (usually t)
- the endpoint of the time interval you are interested in,
- the number of time steps you want.

Here is an example that applies the built-in Runge-Kutta function to the problem

$$\frac{dy}{dt} = y^2, \qquad y(0) = 1$$

with $\Delta t = 0.25$ on the time interval $0 \leq t \leq 1$. Notice that the output is a list of times and values, just as we got from the Euler's method.

```
var('y','t')

f(y) = y^2   # rhs of ODE

t0, y0 = 0, 1 # initial conditions

numsoln = desolve_rk4( f(y), y, ics=[t0,y0], ivar=t,
     end_points=1, step=0.25)

numsoln
```

Here is an application of the Runge-Kutta function that generates a plot.

```
var('y','t')

f(y) = y^2   # rhs of ODE

t0, y0 = 0, 1 # initial conditions

numsoln = desolve_rk4( f(y), y, ics=[t0,y0], ivar=t,
     end_points=1, step=0.25)

numplot = list_plot(numsoln, marker=".",plotjoined=true)
numplot.show(figsize=[4,3])
```

For the homework below, I encourage you to use the Euler's method that we've coded up by hand. But after this assignment is done, it is better to just use the built-in Runge-Kutta algorithm.

TableForTwo

Exercise 1.5.1. Create an approximate table of values for the solution of the IVP

$$\frac{dy}{dt} = \cos(y), \quad y(0) = 0$$

over the interval $0 \leq t \leq 1$. Please use the step-size of $\Delta t = 0.25$.

WNM-MidEnd

Exercise 1.5.2. As you know, the function $y(t) = e^t$ is *the unique* solution of the IVP

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

In this problem you get to investigate the power and the limitation of the Euler method.

1. Make a table which shows *the actual values* of $y(0)$, $y(0.25)$, $y(0.5)$, $y(0.75)$ and $y(1)$.
2. What does the Euler method for step-size of $\Delta t = 0.25$ predict for the values of $y(0)$, $y(0.25)$, $y(0.5)$, $y(0.75)$ and $y(1)$?
3. What does the Euler method for step-size of $\Delta t = 0.05$ predict for the values of $y(0)$, $y(0.25)$, $y(0.5)$, $y(0.75)$ and $y(1)$?
4. Make a table that lists the errors that the two approximation methods make. Then write a sentence or so interpreting what you see.
5. Represent your findings graphically in the t–y plane. Then write a sentence or so interpreting what you see.

WNM-MidBegin

Exercise 1.5.3. Consider the differential equation

$$\frac{dy}{dt} = y(y-3)(y-6).$$

1. Perform qualitative analysis of this equation.
2. In addition to the differential equation above consider the initial condition $y(0) = 1$. Perform Euler method with the step size $\Delta t = 0.5$ in order to understand $y(t)$ for $0 \leq t \leq 2$. Make sure your answer includes the table of values and the corresponding piece-wise linear graph.
3. In light of part (a) please discuss the reliability of your answer in part (b).

HowAreYouTodayMyFriend

Exercise 1.5.4. Consider the initial value problem

$$\frac{dy}{dt} = y^3, \qquad y(0) = 1.$$

1. Using the Euler's Method with $\Delta t = 0.5$, graph an approximate solution over the interval $0 \leq t \leq 1$.
2. What happens if you make Δt considerably smaller than above (e.g. $\Delta t = 0.05$)?
3. Verify that the function
$$y(t) = \frac{1}{\sqrt{1-2t}}$$
is a solution to the IVP in this problem. Use this knowledge to interpret the findings in parts a) and b) of this problem.

WNM-Extra-Only

Exercise 1.5.5. Construct a numerical solution to the initial value problem

$$\frac{dy}{dt} = -y^2 + t, \qquad y(2) = 2.$$

Use your solution to estimate $y(10)$.