Euler’s Method

Author **Aaron Tresham**

Date **2017-03-06T20:08:03**

Project **9189c752-e334-4311-afa9-605b6159620a**

Location Notes for Videos/Euler's Method Notes.sagews

Original file Euler's Method Notes.sagews

Euler’s Method

Euler’s Method is an algorithm used to construct approximate solutions to a differential equation of the form $\frac{dy}{dx} = f(x, y)$ starting at an initial point (x_0, y_0).

Since the differential equation $\frac{dy}{dx} = f(x, y)$ tells us the slope of the tangent line at any point on the xy-plane, we can find the slope at (x_0, y_0) and move along the tangent line some distance to a point (x_1, y_1). Since the solution curve is close to its tangent line (as long as we’re not too far from the point of tangency), the point (x_1, y_1) is almost on the solution curve.

Now we repeat the process. Find the tangent line at (x_1, y_1) using the differential equation, follow it for a short distance, and find a new point (x_2, y_2). This point is also close to the solution curve.

Repeat the process as many times as you like.

1

The process is the same each time, so we can develop an iterated formula and automate the process.

Let’s determine how to get from (x_n, y_n) to (x_{n+1}, y_{n+1}).

First, we need to find the tangent line at (x_n, y_n).

In general, the tangent line to a function $y(x)$ at the point $x = a$ has equation $TL(x) = y(a) + y'(a)(x - a)$.

In this case, the derivative is given by the differential equation, $a = x_n$, and $y(a) = y_n$, so we have $TL(x) = y_n + f(x_n, y_n)(x - x_n)$.

Therefore,

$$y_{n+1} = TL(x_{n+1}) = y_n + f(x_n, y_n)(x_{n+1} - x_n)$$

2

We will move along the tangent line the same horizontal distance each step of the process. In other words, $x_{n+1} - x_n$ is a constant, called the “step size.” We will call this h.

In summary, we have the following:

Euler’s Formula

$$x_{n+1} = x_n + h$$

$$y_{n+1} = y_n + f(x_n, y_n) \cdot h$$

Example 1

Use Euler’s Method to approximate the solution curve to the differential equation $\frac{dy}{dx} = x \cdot y$ that passes through the point $(0, 1)$. Plot the approximation for $0 \le x \le 2$.

We’ll start with a small example by hand, and then we’ll let the computer do the work.

We will use just 5 steps. That means the step size is $h = \frac{2-0}{5} = \frac{2}{5}$.

We’ll start with $x_0 = 0$ and $y_0 = 1$, and then we will calculate new x- and y-coordinates with the formulas

$$x_{n+1} = x_n + h$$

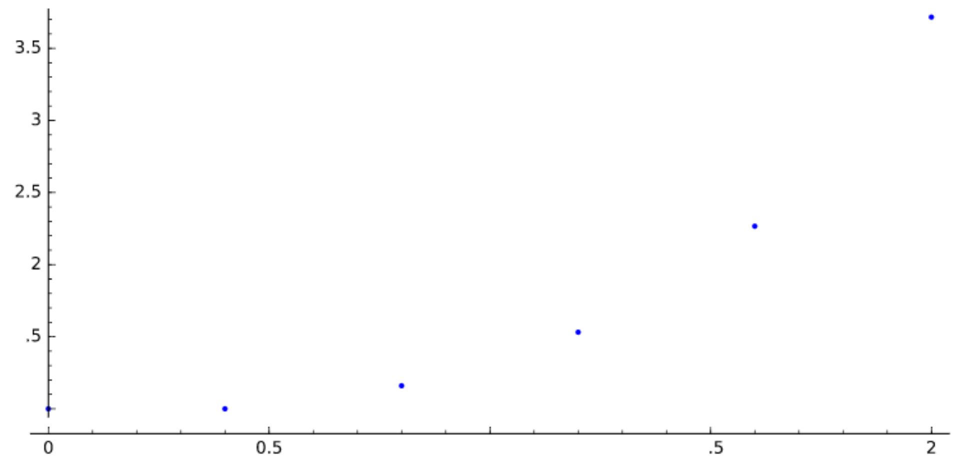
$$y_{n+1} = y_n + x_n \cdot y_n \cdot h$$

x	y

$x_0 = 0$	$y_0 = 1$
$x_1 = 0 + \frac{2}{5} = \frac{2}{5}$	$y_1 = 1 + 0 \cdot 1 \cdot \frac{2}{5} = 1$
$x_2 = \frac{2}{5} + \frac{2}{5} = \frac{4}{5}$	$y_2 = 1 + \frac{2}{5} \cdot 1 \cdot \frac{2}{5} = \frac{29}{25}$
$x_3 = \frac{4}{5} + \frac{2}{5} = \frac{6}{5}$	$y_3 = \frac{29}{25} + \frac{4}{5} \cdot \frac{29}{25} \cdot \frac{2}{5} = \frac{957}{625}$
$x_4 = \frac{6}{5} + \frac{2}{5} = \frac{8}{5}$	$y_4 = \frac{957}{625} + \frac{6}{5} \cdot \frac{957}{625} \cdot \frac{2}{5} = \frac{35409}{15625}$
$x_5 = \frac{8}{5} + \frac{2}{5} = 2$	$y_5 = \frac{35409}{15625} + \frac{8}{5} \cdot \frac{35409}{15625} \cdot \frac{2}{5} = \frac{1451769}{390625}$

Now let’s plot these six points.

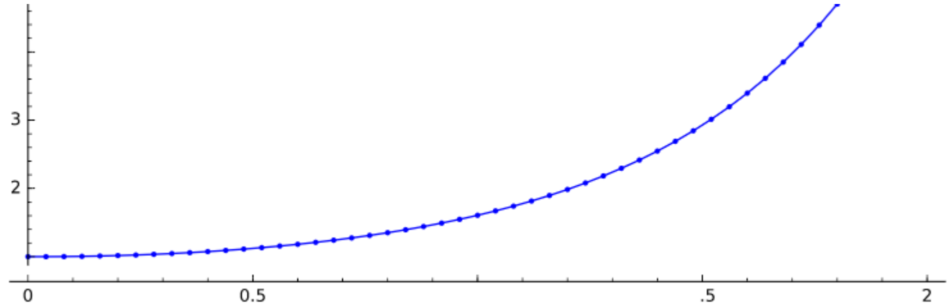
```
point([(0,1),(2/5,1),(4/5,29/25),(6/5,957/625),(8/5,35409/15625),(2,1451769/390625)])
```



The six points above are approximately on the solution curve. If we connect the points with straight lines, we will have an approximate solution curve.

Of course, just 5 steps is not enough to get a good approximation, so we’ll use the computer with many more steps.

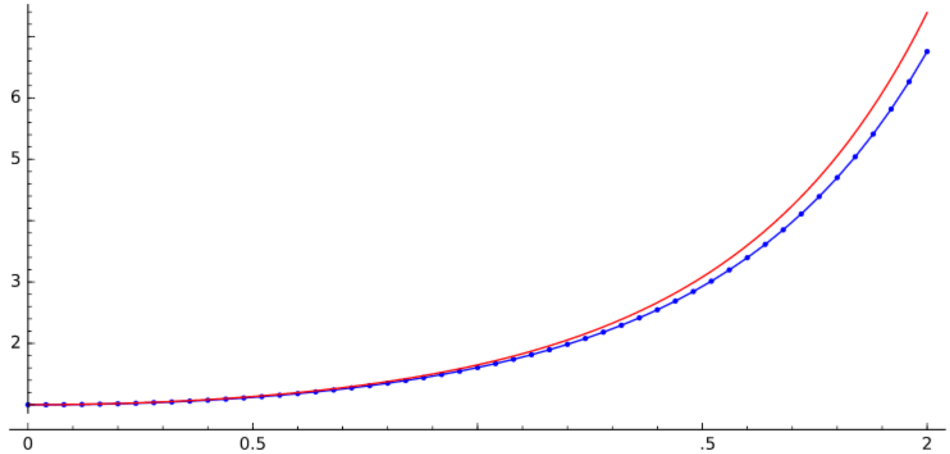
```
%var y
f(x,y)=x*y            #this is the function given by the differential equation
x0=0                  #initial value of x given in the problem
y0=1                  #initial value of y given in the problem
x_end=2               #the x-value you want to stop at
n=50                  #number of points to calculate
h=(x_end-x0)/n        #this calculates the step size for you
xlist=[x0];ylist=[y0] #we will use lists to keep track of all the x's and y's
p=point((x0,y0))      #we'll start keeping track of the points to graph
for i in range(n):    #here we apply Euler's Formula
    xlist+=[xlist[i]+h]
    ylist+=[ylist[i]+RR(f(xlist[i],ylist[i])*h)]  #Note: RR converts to a floating-point number to avoid Sage taking too much time with exact values.
    p=p+point((xlist[i+1],ylist[i+1]))+line([(xlist[i],ylist[i]),(xlist[i+1],ylist[i+1])])
show(p)
```



Here is a plot of our approximation (blue) along with the actual solution (red).

18

```
p+plot(e^(1/2*x^2),xmin=x0,xmax=x_end,color='red')
```



We can make the approximation better by increasing n (this decreases the step size).

If we want to plot the approximation past $x = 2$, then we can change x_end. Of course, the approximation is going to get worse when we are farther away from our starting point.

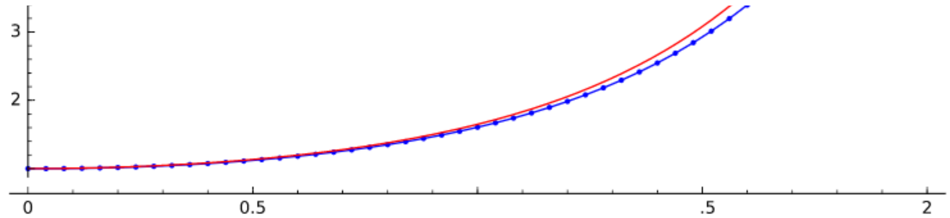
The interactive box below allows us to change n and x_end. Experiment with different values.

x_end

Submit

n

Submit



Example 2

Consider the initial value problem $\frac{dy}{dx} = y + x, \quad y(0) = 0$.

Use Euler's Method to approximate $y(2)$.

I will copy and paste the formulas from above, skipping the plot:

```
%var y
f(x,y)=y+x  #dy/dx = y+x
x0=0        #initial x-value = 0
y0=0        #initial y-value = 0
x_end=2     #we want to stop at x = 2
n=50        #we'll try 50 for now
h=(x_end-x0)/n
xlist=[x0];ylist=[y0]
for i in range(n):
    xlist+=[xlist[i]+h]
    ylist+=[ylist[i]+RR(f(xlist[i],ylist[i])*h)]
N(ylist[n]) #notice that ylist[n] is the last point calculated, in this case y(2)
```

```
4.10668334627831
```

We have found that $y(2) \approx 4.1067$.

Let's try a higher value of n and see what happens.

```
%var y
f(x,y)=y+x
x0=0
y0=0
x_end=2
n=100        #we'll try n=100 this time
h=(x_end-x0)/n
xlist=[x0];ylist=[y0]
for i in range(n):
    xlist+=[xlist[i]+h]
    ylist+=[ylist[i]+RR(f(xlist[i],ylist[i])*h)]
N(ylist[n])
```

```
4.24464611825234
```

Now we have $y(2) \approx 4.2446$.

Let's find the actual value. First, solve the differential equation.

```
y=function('y',x)
desolve(derivative(y,x)==y+x,y,[0,0])
```

```
-x + e^x - 1
```

Now plug in $x = 2$.

```
F(x)=-x + e^x - 1 #I'll call the solution F(x)
F(2)
N(F(2))
```

```
e^2 - 3
4.38905609893065
```

So $y(2) = 4.38905609893065$.

Notice that increasing n has gotten us closer to the actual answer. Let's increase n one more time and see if we can get at least the first decimal place correct.

```
%var y
f(x,y)=y+x
x0=0
y0=0
x_end=2
n=500        #we'll try n=500 this time
h=(x_end-x0)/n
xlist=[x0];ylist=[y0]
for i in range(n):
    xlist+=[xlist[i]+h]
    ylist+=[ylist[i]+RR(f(xlist[i],ylist[i])*h)]
N(ylist[n])
```

```
4.35963717586897
```

Here is a summary of our results:

n	Approximation
50	4.10668334627831
100	4.24464611825234
500	4.35963717586897

The actual value is $e^2 - 3 \approx 4.38905609893065$.