

Lab 6: Euler's Method (2.4) Solutions

Famous Number Investigation

The problems below describe the numbers e , $\ln 2$ and π , as specific values of certain initial value problem solutions.

In each case, apply Euler's method with $n=50, 100, 200, \dots$ subintervals (doubling each time). How many subintervals are needed to obtain- twice in succession- the correct value of the target number rounded to 3 decimal places?

- 1) The number $e = y(1)$ where $y(x)$ is the solution of the initial value problem $y' = y$, $y(0) = 1$.
- 2) The number $\ln 2 = y(2)$ where $y(x)$ is the solution of the initial value problem $y' = 1/x$, $y(1) = 0$.
- 3) The number $\pi = y(1)$ where $y(x)$ is the solution of the initial value problem $y' = 4/(1+x^2)$, $y(0) = 0$.

1)

```
> evalf(exp(1));
2.718281828
> f:=(x,y)->y;
f := (x, y) -> y
> x0:=0: y0:=1: xf:=1:
> n:=50: h:=evalf((xf-x0)/n);
h := 0.02000000000
> x[0]:=x0; y[0]:=y0;
x0 := 0
y0 := 1
> for i from 1 to n do k:=f(x[i-1],y[i-1]): y[i]:= y[i-1]+h*k:
x[i]:= x[i-1]+h: od:
> y[50];
2.691588026
> n:=100: h:=evalf((xf-x0)/n);
h := 0.01000000000
> for i from 1 to n do k:=f(x[i-1],y[i-1]): y[i]:= y[i-1]+h*k:
x[i]:= x[i-1]+h: od:
> y[100];
2.704813833
> n:=200: h:=evalf((xf-x0)/n): for i from 1 to n do
k:=f(x[i-1],y[i-1]): y[i]:= y[i-1]+h*k: x[i]:= x[i-1]+h: od:
```

```
y[n];
```

```
2.711517128
```

```
> n:=400: h:=evalf((xf-x0)/n): for i from 1 to n do  
k:=f(x[i-1],y[i-1]): y[i]:= y[i-1]+h*k: x[i]:= x[i-1]+h: od:  
y[n];
```

```
2.714891751
```

```
> n:=800: h:=evalf((xf-x0)/n): for i from 1 to n do  
k:=f(x[i-1],y[i-1]): y[i]:= y[i-1]+h*k: x[i]:= x[i-1]+h: od:  
y[n];
```

```
2.716584843
```

```
> n:=1600: h:=evalf((xf-x0)/n): for i from 1 to n do  
k:=f(x[i-1],y[i-1]): y[i]:= y[i-1]+h*k: x[i]:= x[i-1]+h: od:  
y[n];
```

```
2.717432865
```

```
> n:=3200: h:=evalf((xf-x0)/n): for i from 1 to n do  
k:=f(x[i-1],y[i-1]): y[i]:= y[i-1]+h*k: x[i]:= x[i-1]+h: od:  
y[n];
```

```
2.717857206
```

```
> n:=6400: h:=evalf((xf-x0)/n): for i from 1 to n do  
k:=f(x[i-1],y[i-1]): y[i]:= y[i-1]+h*k: x[i]:= x[i-1]+h: od:  
y[n];
```

```
2.718069503
```

```
> n:=12800: h:=evalf((xf-x0)/n): for i from 1 to n do  
k:=f(x[i-1],y[i-1]): y[i]:= y[i-1]+h*k: x[i]:= x[i-1]+h: od:  
y[n];
```

```
2.718175587
```

[Whew. To get it in 3 decimal places you need 12800 intervals.

[2.

```
> evalf(ln(2));
```

```
0.6931471806
```

```
> f:=(x,y)->1/x;
```

$$f := (x, y) \rightarrow \frac{1}{x}$$

```
> x0:=1: y0:=0: xf:=2:
```

```
> x[0]:=x0: y[0]:=y0;
```

$$x_0 := 1$$

$$y_0 := 0$$

```
> n:=50: h:=evalf((xf-x0)/n): for i from 1 to n do  
k:=f(x[i-1],y[i-1]): y[i]:= y[i-1]+h*k: x[i]:= x[i-1]+h: od:  
y[n];
```

```
0.6981721793
```

```
> n:=100: h:=evalf((xf-x0)/n): for i from 1 to n do
k:=f(x[i-1],y[i-1]): y[i]:= y[i-1]+h*k: x[i]:= x[i-1]+h: od:
y[n];
```

0.6956534304

```
> n:=200: h:=evalf((xf-x0)/n): for i from 1 to n do
k:=f(x[i-1],y[i-1]): y[i]:= y[i-1]+h*k: x[i]:= x[i-1]+h: od:
y[n];
```

0.6943987432

```
> n:=400: h:=evalf((xf-x0)/n): for i from 1 to n do
k:=f(x[i-1],y[i-1]): y[i]:= y[i-1]+h*k: x[i]:= x[i-1]+h: od:
y[n];
```

0.6937725710

```
>
```

```
> n:=800: h:=evalf((xf-x0)/n): for i from 1 to n do
k:=f(x[i-1],y[i-1]): y[i]:= y[i-1]+h*k: x[i]:= x[i-1]+h: od:
y[n];
```

0.6934597777

This converged much faster, only 800 subintervals are needed.

3.

```
> evalf(Pi);
```

3.141592654

```
> f:=(x,y)->4/(1+x^2);
```

$$f := (x, y) \rightarrow \frac{4}{1+x^2}$$

```
> x0:=0: y0:=0: xf:=1:
```

```
> x[0]:=x0: y[0]:=y0;
```

$$x_0 := 0$$

$$y_0 := 0$$

```
> n:=50: h:=evalf((xf-x0)/n): for i from 1 to n do
k:=f(x[i-1],y[i-1]): y[i]:= y[i-1]+h*k: x[i]:= x[i-1]+h: od:
y[n];
```

3.161525983

```
> n:=100: h:=evalf((xf-x0)/n): for i from 1 to n do
k:=f(x[i-1],y[i-1]): y[i]:= y[i-1]+h*k: x[i]:= x[i-1]+h: od:
y[n];
```

3.151575985

```
> n:=200: h:=evalf((xf-x0)/n): for i from 1 to n do
k:=f(x[i-1],y[i-1]): y[i]:= y[i-1]+h*k: x[i]:= x[i-1]+h: od:
y[n];
```

3.146588490

```

[ > n:=400: h:=evalf((xf-x0)/n): for i from 1 to n do
  k:=f(x[i-1],y[i-1]): y[i]:= y[i-1]+h*k: x[i]:= x[i-1]+h: od:
  y[n];
                                     3.144091616
[ > n:=800: h:=evalf((xf-x0)/n): for i from 1 to n do
  k:=f(x[i-1],y[i-1]): y[i]:= y[i-1]+h*k: x[i]:= x[i-1]+h: od:
  y[n];
                                     3.142842396
[ > n:=1600: h:=evalf((xf-x0)/n): for i from 1 to n do
  k:=f(x[i-1],y[i-1]): y[i]:= y[i-1]+h*k: x[i]:= x[i-1]+h: od:
  y[n];
                                     3.142217596
[ > n:=3200: h:=evalf((xf-x0)/n): for i from 1 to n do
  k:=f(x[i-1],y[i-1]): y[i]:= y[i-1]+h*k: x[i]:= x[i-1]+h: od:
  y[n];
                                     3.141905147
[ > n:=6400: h:=evalf((xf-x0)/n): for i from 1 to n do
  k:=f(x[i-1],y[i-1]): y[i]:= y[i-1]+h*k: x[i]:= x[i-1]+h: od:
  y[n];
                                     3.141748899
[ >

```

[For this example 6400 subdivision were required.