Euler's Method

James Rohal

First load these packages.

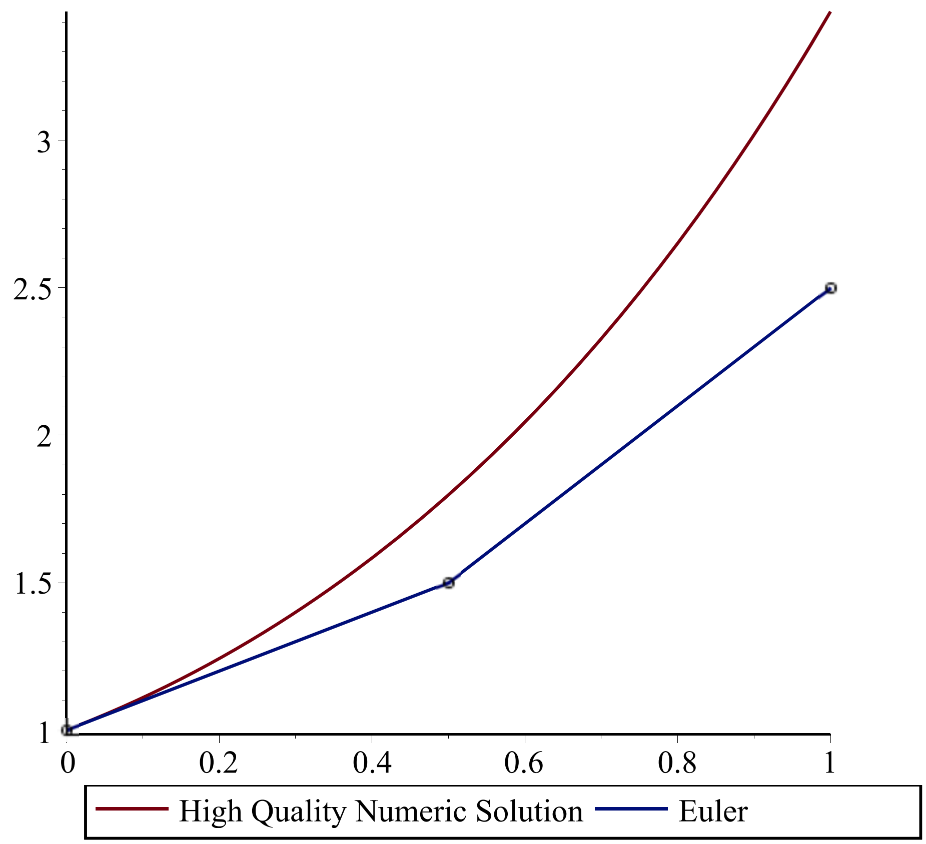
```
> with(Student[NumericalAnalysis]):
  # this just allows us to show bigger tables
  interface(rtablesize=14):
  with(plots):
```

Euler's Method

We wish to solve the differential equation $y'(x) = F(x, y), y(x_0) = y_0$. We illustrate with the example $y' = x + y, y(0) = 1$. Here $F(x, y) = x + y, x_0 = 0, y_0 = 1$.

```
> num_of_steps := 2:

  Euler(diff(y(x),x)=x + y(x), y(0)=1, x=1, numsteps=
  num_of_steps, output=plot);
  Euler(diff(y(x),x)=x + y(x), y(0)=1, x=1, numsteps=
  num_of_steps, output=information);
```



x	[*Maple's numeric solution*]	[*Euler*]	[*Error*]
0.	1.	1.	0.
0.5000	1.797	1.500	0.2970
1.	3.437	2.500	0.9370

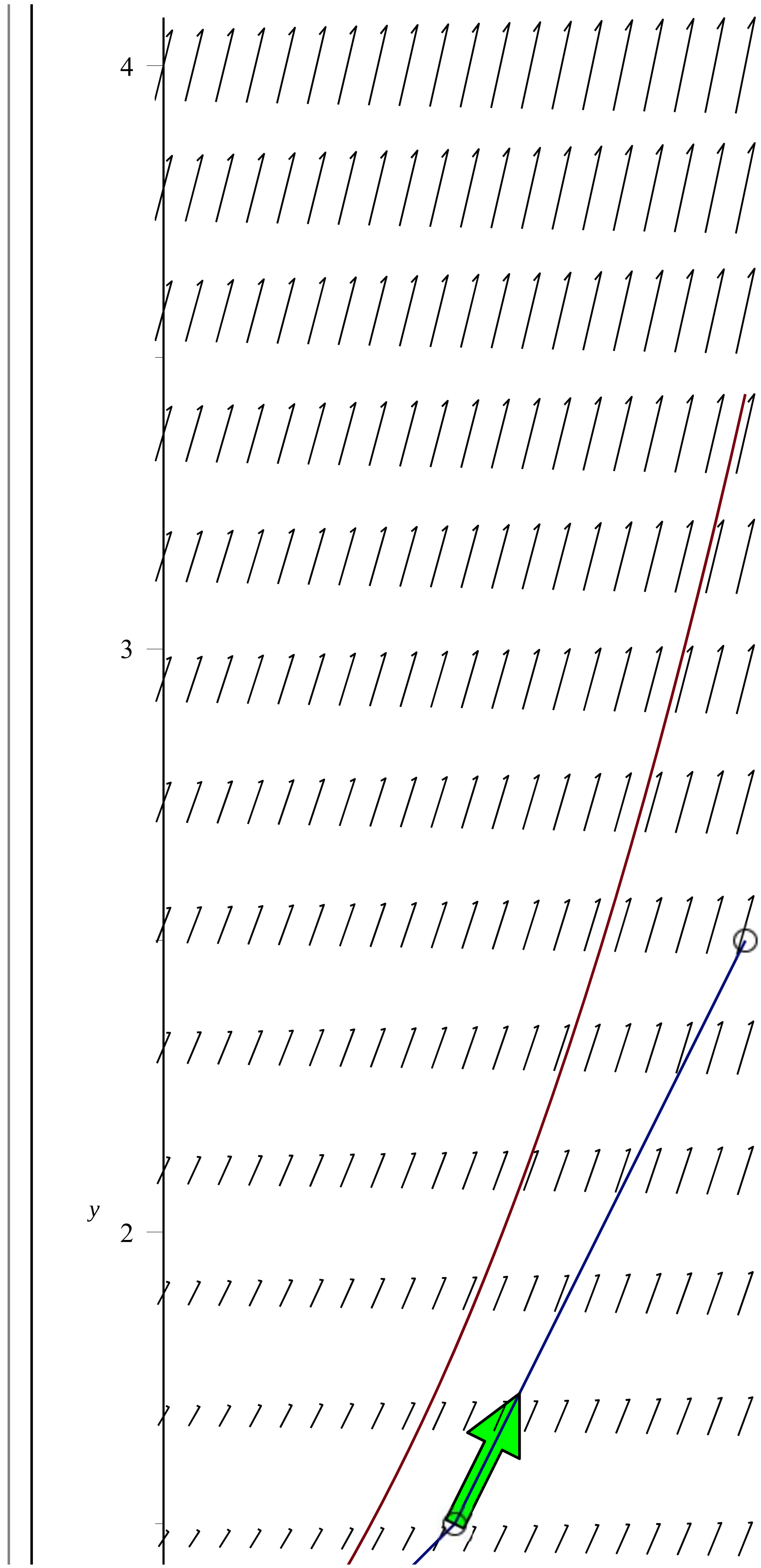
(1.1)

```
> field           := fieldplot([1, x+y], x=0..1, y=0..4):
  euler_soln_plot := Euler(diff(y(x),x)=x + y(x), y(0)=1, x=1,
  numsteps=num_of_steps, output=plot):
  euler_soln      := Euler(diff(y(x),x)=x + y(x), y(0)=1, x=1,
  numsteps=num_of_steps, output=information):

  dir := (x,y) -> x + y:
  step_size    := (1-0)/num_of_steps:

  arrow_list := seq(  arrow([euler_soln[i,1], euler_soln[i,3]],
  [step_size, step_size*dir(euler_soln[i,1], euler_soln[i,3])],
  color=green, width=1/30, shape=double_arrow, head_width=1/10,
  head_length=1/10, length=1/4),  i=2..num_of_steps + 1):

  display(field, euler_soln_plot, arrow_list, scaling=
  constrained);
```



4
3
y
2

Homework

Write a series of commands to solve the differential equation $y'(x) = F(x, y), y(x_0) = y_0$ using Euler's method. Use this as a template to solve the differential equation $y' = x + y, y(0) = 1$. Here $F(x, y) = x + y, x_0 = 0, y_0 = 1$.

Your solution should output n, x_n, y_n at each step. This is done using **print**. The output you should get is:

0, 0, 1
1, 0.1, 1.1
2, 0.2, 1.22
3, 0.3, 1.362
4, 0.4, 1.5282
5, 0.5, 1.72102
6, 0.6, 1.943122
7, 0.7, 2.1974342
8, 0.8, 2.48717762
9, 0.9, 2.815895382
10, 1.0, 3.187484920 **(2.1)**

Below you need to fill in spots designated by ______.

```
> ################
  # INITIAL DATA #
  ################
  F             := x + y:
      # the RHS of the differential equation in terms of x and y
  num_of_steps := 10:
      # number of steps to do
  x_0           := 0:
      # initial condition:
  y_0           := 1:
      #   y(x_0) = y_0
  h             := 0.1:
      # choose a step size


  ##################
  # SETUP FOR LOOP #
  ##################
  print(0,x_0,y_0);
      # print initial data

  x_n_minus_1 := _____:

  y_n_minus_1 := _____:
```

```
########
# LOOP #
########
for n from 1 to num_of_steps do
    # performs num_of_steps-1 steps of all the code between
"do" and "end do"
  x_n := _____;
  y_n := _____;

  print(n, x_n, y_n);

  x_n_minus_1 := _____;
  y_n_minus_1 := _____;
end do:
```

0, 0, 1
1, _, _
2, _, _
3, _, _
4, _, _
5, _, _
6, _, _
7, _, _
8, _, _
9, _, _
10, _, _ **(2.2)**