

Euler's Method (Part I)

Accompanies Section 1.4 in ODEP

Dave Rosoff

Department of Mathematics and Physical Sciences

The College of Idaho

Spring 2018

1 Problems for this activity

Refer to the following initial value problems while completing the activities in this worksheet.

$$y' = 3 + t - y \qquad y(0) = 1 \qquad (1.1)$$

$$y' = 2y - 1 \qquad y(0) = 1 \qquad (1.2)$$

$$y' = 0.5 - t + 2y \qquad y(0) = 1 \qquad (1.3)$$

$$y' = 3 \cos(t) - 2y \qquad y(0) = 0 \qquad (1.4)$$

2 Euler's method

For each of the initial value problems given in [Section 1](#), complete the following:

- Find approximate values of the solution of the IVP at $t = 0.1, 0.2, 0.3, 0.4$ using the Euler method with $h = 0.1$.
- Repeat, but with $h = 0.05$. Graph and tabulate to compare with your previous results.
- Repeat, but with $h = 0.01$. Graph and tabulate to compare with your previous results.
- Use the techniques we previously developed to find the unique solution $y(t)$ and evaluate it at $t = 0.1, 0.2, 0.3, 0.4$. Summarize all your results in a table and with a plot.

3 Worked example

Let's work through some of the details for equation 1. One way to do this is to store all the results in two lists.

```
t, y = var('t,y')
ts = []
ys = [1]
yp = 3 + t - y
h = 0.1
```

```
ts.append(ts[-1] + h)
ys.append(ys[-1] + h*yp.subs(t = ts[-1], y=ys[-1]))
```

To see the contents of the `ts` and `ys` lists in a familiar form, run the next cell. Keep running the previous cell to add more values to the lists. Use the next cell to preview the new results. Add values until you have enough data to answer the questions.

```
our_table = [ts, ys]
table(columns=our_table, header_row=['t', 'y'])
```

Here's a good way to graph the points you have stored in a table like this.

```
flip_table = map(list, zip(*our_table)) # take the transpose
of the table
p = list_plot(flip_table, color="blue", size=72)
p += line(flip_table, color="blue", thickness=3)
show(p)
```

The activity asks you to summarize your findings for different values of h . You may find it helpful to know how to extract every k th element from a list. Here is a good Pythonic way to do it.

```
my_list = ["differential", "equations", "are",
           "equations", "relating", "functions",
           "and", "their", "derivatives"]
my_list[0::3] # start at 0th list entry and use step size 3,
              until list is done
```