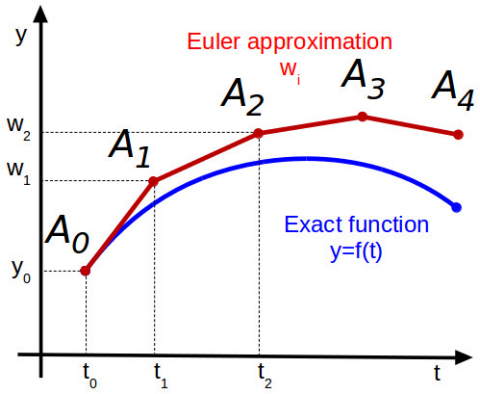
Euler Code:
A first ODE solver

MATH1090: Numerical Methods in Scientific Computing II

http://people.sc.fsu.edu/~jburkardt/classes/math1090_2020/euler_code/euler_code.pdf

---



The Euler method steps off in the right direction, and gradually drifts away.

Euler Code

Writing a simple, general Euler code to solve a differential equation will give you an idea of the kind of mathematical problems we are interested in, and the computational methods we use to solve them.

1 Introduction

This is an introduction for all of us. You would like to know what we are working on. We would like to know how well prepared you are for the computational efforts in this directed study class.

Without knowing much about your backgrounds, we are asking you to try the following sequence of tasks, writing up the necessary MATLAB code. You are welcome to ask us for help at any point. When you think you are done, send us the files you have written and the plot you have created. We will meet on Tuesday, January 21, at 2:00pm in Thackeray 624 to go over this task, and to talk about plans for future work.

2 A differential equation

Consider the following differential equation:

$$\frac{dy}{dt} = \frac{-2 * (t - 0.3)}{((t - 0.3)^2 + 0.01)^2} - \frac{2 * (t - 0.9)}{((t - 0.9)^2 + 0.04)^2}$$

with initial condition $y(0) = 5.1765$. From this information, we can estimate the value of $y(t)$ for future times. We are interested in creating a plot of our estimate over the interval $0 \leq t \leq 2$.

3 Write a derivative function

In MATLAB, a derivative function has a typical structure. If we follow this structure, then we can use the same derivative function with many different ODE solvers that MATLAB provides. Even though we aren't ready to do that yet, we will follow the pattern, and this will pay off in the future.

Write a MATLAB function named *hump_deriv.m*, with the following format, filling in the details:

```
function dydt = hump_deriv ( t, y )
  dydt = (the formula above)
  return
end
```

Listing 1: Outline of a derivative code.

4 Write a (forward) Euler ODE solver

Write a MATLAB function named *euler.m* which uses Euler's method to solve an ODE. It should have the form:

```
function [ t, y ] = euler ( dydt, tspan, y0, n )
  (initialization)
  for i = 1 : n
    compute next t and y
  end
  return
end
```

Listing 2: Outline of euler code.

Here the input is:

- `dydt` is the name of a derivative function.
- `tspan` is a vector of length 2. `t(1)` is the start, and `t(2)` the final time.
- `y0` contains the value of y at the initial time.
- `n` is the number of equal steps to take from `t(1)` to `t(2)`.

and the output is:

- `t` a vector of length `n+1`, containing the initial time, and the `n` later times.
- `y` is a vector of length `n+1`, containing the initial y0, and the `n` later estimates.

5 Use the solver and the derivative functions to make an estimate

Make an estimate of the solution of the differential equation by using the following statement:

```
[ t, y ] = euler ( @hump_deriv, [0.0,2.0], 5.1765, 100 );
```

Listing 3: How to request a solution.

or, if you like to define your input variables:

```
  dydt = @hump_deriv;
  tspan = [ 0.0, 2.0 ];
  y0 = 5.1765;
  n = 100;
  [ t, y ] = euler ( dydt, tspan, y0, n );
```

Listing 4: How to request a solution.

6 Plot your solution

The output from your computation should be two vectors, `t` and `y`, of length `n+1`. You can make a plot of these with the command:

```
plot ( t, y );
```

Listing 5: Basic plot command.

A fancier plot can be made, with thicker lines, a grid, and labels:

```
plot ( t, y, 'linewidth', 3 );
grid ( 'on' );
xlabel ( '<-- T -->' );
ylabel ( '<-- Y(T) -->' );
title ( 'Euler solution to an ODE' );
```

Listing 6: Customized plot command.

In order to send the plot, you have to save a copy of it. By default, MATLAB will save your plot as a ".fig" file. However, to view this file later, you have to run MATLAB again. A better option is to save it as a PNG file, which you can do by a command like this:

```
print ( '-dpng', 'my_plot.png' );
```

Listing 7: Saving your plot as a PNG file.

7 Submit your data

Send your versions of *hump_deriv.m*, *euler.m*, and your plot to me at `jvb25@pitt.edu`.