Section 2.4: Euler's Method

Prior Knowledge

To approximate an ordinary differential equation of the form

$$\frac{dy}{dx} = f(x, y), \quad y(x_0) = y_0$$

we need to recall the limit definition of the derivative, which is

$$\frac{dy}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Motivation

It is the exception rather than the rule when a differential equation of the general form

$$\frac{dy}{dx} = f(x, y)$$

can be solved exactly and explicitly, by traditional methods (e.g., separation of variables, first-order linear equations, etc.). For example, if we had

$$\frac{dy}{dx} = e^{-x^2}$$

A solution of this differential equation is simply an antiderivative of e^{-x^2} but what could it be? Well it is difficult to say with the techniques we know (e.g., integration by parts, integration by substitution, etc.).

With the limit definition we can approximate the derivative of any function with finite differences. So, by the limit definition of the derivative

$$\left.\frac{dy}{dx}\right|_{x=0} \approx \frac{y(h) - y(0)}{h} \approx \frac{y_1 - y_0}{h}$$

But remember what dy/dx equals, it equals $f(x, y)$ as stated above in the prior knowledge. So

$$f(0, y_0) = f(0, y(0)) = \left.\frac{dy}{dx}\right|_{x=0} \approx \frac{y_1 - y_0}{h}$$

Let 0 be x_0. To get the equation

$$f(x_0, y_0) = \frac{y_1 - y_0}{h}$$

Now let's solve for y_1.

$$\begin{aligned} f(x_0, y_0) &= \frac{y_1 - y_0}{h} \\ h \cdot f(x_0, y_0) &= y_1 - y_0 \\ h \cdot f(x_0, y_0) + y_0 &= y_1 \end{aligned}$$

Note that (x_0, y_0) and (x_1, y_1) are arbitrary points which means I can write for a general case. In other words,

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

This equation we found is the iterative formula for Euler's Method.

Algorithm: The Euler Method (Edwards, Penney, Calvis, 2023)

Given the initial value problem

$$\frac{dy}{dx} = f(x, y), \quad y(x_0) = y_0$$

Euler's method with step size h consists of applying the iterative formula

$$y_{n+1} = h \cdot f(x_n, y_n) + y_n \quad (n \geq 0)$$

to calculate successive approximations $y_1, y_2, y_3, \cdot$ to the [true] values $y(x_1), y(x_2), y(x_3), \cdot$ of the [exact] solution $y = y(x)$ at the points $x_1, x_2, x_3, \ldots$, respectively.

We could do these computations by hand or with an old-fashioned computer plotter like ink and pen to draw curves mechanically, but we live in a world with personal CPU machines (i.e., laptops with coding programs like Python. So let's move on the coding phase of this lecture.

(The following code was developed and/or tweaked from Dr Jon Shiach's video https://www.youtube.com/watch?v=N7Oh0mk4YGc&t=378s)

Problem 1:

Apply Euler's method twice to approximate the solution to the initial value problem on the interval $\left[0, \frac{1}{2}\right]$, first with step size $h = 0.25$, then with step size $h = 0.1$. Compare the three-decimal-place values of the two approximations at $x = \frac{1}{2}$ with the value of $y(1/2)$ of the actual solution.

$$y' = y, \quad y(0) = 3, \quad y(x) = 3e^x$$

In [3]:

```
import numpy as np
import math

#Solver function
def solveIVP(f, tspan, y0, h, solver):
    t = np.arange(tspan[0], tspan[1] + h, h)
    y = np.zeros(len(t))
    y[0] = y0

    for n in range(len(t) - 1):
        y[n+1] = solver(f, t[n], y[n], h)

    return t, y

# Euler method
def euler(f, tn, yn, h):
    return yn + h * f(tn,yn)

# Define IVP
def f(t,y):
    return y # becuase y'=f(t,y)=y

# Problem Specific
tspan = [0, 0.5]
x0 = 0
y0 = 3
x = 0.5
h = 0.25
```

```python
# Solve IVP
t, y = solveIVP(f, tspan, y0, h, euler)

# Exact solution
def exact(x):
    return 3 * math.exp(x)

e = exact(x)

# Collect results
results = []

t, y = solveIVP(f, tspan, y0, h, euler)
results.append((h, y[-1]))

# Print table
print(" Steps (h) | Euler y(0.5) |  Exact y(0.5) |  Error ")
print("---------------------------------------------------")
for n in range(len(t)):
    print(f" {t[n]:6.2f}    |    {y[n]:9.3f} |     {exact(x):9.3f} | {abs(exact(x)-y[n]):6.3f}")
```

```
 Steps (h) | Euler y(0.5) |  Exact y(0.5) |  Error
---------------------------------------------------
   0.00    |        3.000 |         4.946 |  1.946
   0.25    |        3.750 |         4.946 |  1.196
   0.50    |        4.688 |         4.946 |  0.259
```

In [4]:

```python
h = 0.1

t, y = solveIVP(f, tspan, y0, h, euler)

# Print table of solutions
print(" Steps (h) | Euler y(0.5) |  Exact y(0.5) |  Error ")
print("---------------------------------------------------")
for n in range(len(t)):
    print(f" {t[n]:6.2f}    |    {y[n]:9.3f} |     {exact(x):9.3f} | {abs(exact(x)-y[n]):6.3f}")
```

```
 Steps (h) | Euler y(0.5) |  Exact y(0.5) |  Error
---------------------------------------------------
   0.00    |        3.000 |         4.946 |  1.946
   0.10    |        3.300 |         4.946 |  1.646
   0.20    |        3.630 |         4.946 |  1.316
   0.30    |        3.993 |         4.946 |  0.953
   0.40    |        4.392 |         4.946 |  0.554
   0.50    |        4.832 |         4.946 |  0.115
```

Application

This method can be used to solve the following real-world example

Problem 2:

A baseball is thrown straight downward from a helicopter hovering at an altitude of 3000 ft. Let $v(t)$ be the downward velocity (ft/s). The motion is modeled by the initial value problem

$$\frac{dv}{dt} = 32 - 0.16v, \qquad v(0) = 0.$$

1. Use Euler's method with step size $h = 1$ second to approximate the velocity of the baseball at $t = 5$ seconds.
2. Write out the iterative formula you are using.
3. Compute and tabulate the approximate values $v(1), v(2), \ldots, v(5)$.

4. Compare your final approximation $v(5)$ with the exact solution

$$v(t) = 200\big(1 - e^{-0.16t}\big).$$

Sources:

1. Differential Equations and Boundary Value Problems (2023) by Edwards, Penney, Calvis
2. Dr. Jon Shiach's Euler Method (Python) video found at https://www.youtube.com/watch?v=N7Oh0mk4YGc&t=378s

In [ ]:

```
```