# MATH1072

Practical 1: Introduction to MATLAB and Numerical Differentiation

**Motivation:** There are many concepts we take for granted in learning how to derive analytic solutions to mathematical problems. We are able to learn the rules of differentiation and integration and apply them to symbols to obtain a representation of the true, (possibly continuous) solution to a problem. Computers cannot process in infinitesimals and cannot think in continuity. (This is sort of a lie, see `symbolics`.)

One major aspect of Numerical Analysis (or Scientific Computing) is the task of transforming continuous problems into a discrete space so that a computer can be used to solve the problem. Of course, this discretization introduces error on the solution and estimating/reducing this error is another major aspect of this field.

## 1 1D Numerical Differentiation

Consider the 1D definition of the derivative given by,

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

One natural numerical approximation of this derivative is to fix $h$ small and consider,

$$f'(x) \approx \frac{f(x+h) - f(x)}{h},$$

this is known as the *forward-difference* approximation of the derivative of $f$ at the point $x$.

### MATLAB Practice

(a). Write a `for` loop to calculate the *forward-difference* approximation of the derivative of $f(x) = \cos(x)$ at points $x$ along the interval $[0, 2]$ for values of $h = 1, 0.5, 0.25, 0.1$.
(b). Plot the true solution before the `for` loop and type `hold on` to keep the figure

from refreshing. Plot the estimated solution you obtain for every $h$ in the loop on the same plot using `plot`.
A script outline is given below:

```
h = []; %define the vector h containing your initial
   condition
f = ; %define the function cos(x)
%plot the analytical solution (derivative of cos(x))
plot()
hold on

for i =  %run the for loop for all h values
    x =  %define the 1D grid of x values you will evaluate
        the
    %function on
    df = ; %perform the forward-difference approximation
       and save it to a matrix
    plot()
end

legend()
```

(c). What is happening to the solution as $h$ gets smaller (e.g. as $h$ gets closer to 0)?

# 2 2D Numerical Differentiation

The concept here is similar to the 1D case, only now we consider the *forward-difference* approximation of the partial derivatives given in lecture,

$$\frac{\partial f}{\partial x}(x,y) \approx \frac{f(x+h,y)-f(x,y)}{h}$$

$$\frac{\partial f}{\partial y}(x,y) \approx \frac{f(x,y+h)-f(x,y)}{h}$$

## MATLAB Practice

(a). Consider the function $f(x,y) = \cos(x)+\sin(y)$ over the interval $[-2,2]\times[-2,2]$. Take $h = 0.1$ and generate a 2D (uniform) grid of step size $h$ using the `meshgrid` function.

(b). Compute the *forward-difference* approximation of the partial derivatives for each grid point pair.
(c). On the same figure, plot the 2D function using the `contour` and the resulting gradient using the `quiver` function.
A script outline is given below:

```
[X, Y] = meshgrid(); %Use the meshgrid function to create
   the 2D uniform grid of step size h
f = %define the function f
dfx = %define the approximation of the partial derivative
   with respect to x
dfy = %define the approximation of the partial derivative
   with respect to y

contour()  %plot the 2D function
hold on
quiver() %plot the gradient
```