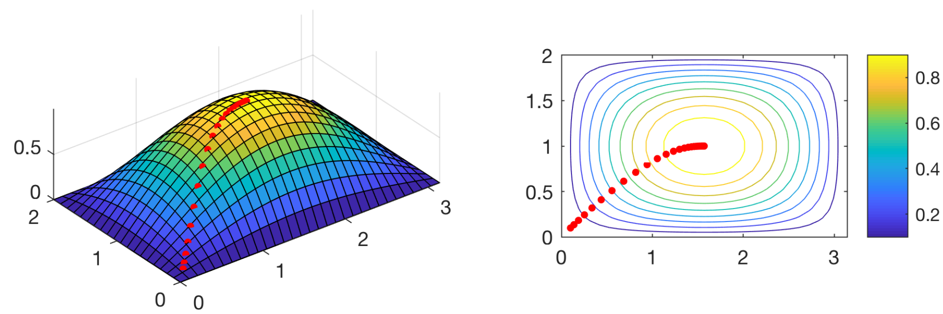
UBC MECH 222: MATLAB Computer Lab 4

Gradient Ascent Algorithm




```
>> f = @(x,y) (2*y - y.^2) .* sin(x);
>> [x,y,z,t,n] = gradient_ascent(f,[0,pi,0,2],[0.1,0.1],0.01,0.2,0.001,1000)
Local maximum found in 0.195 seconds and 40 iterations.
x =
    1.5700
y =
    1.0000
z =
    1.0000
t =
    0.1952
n =
    40
```

Instructions

Write a function called `gradient_ascent` which takes 7 input parameters:

- ☐ `f` is a function handle of a vectorized function of two variables $f(x, y)$
- ☐ `R` is a vector $[a, b, c, d]$ of length 4 defining a rectangle $R = [a, b] \times [c, d]$ in the xy-plane
- ☐ `start` is a vector $[x_0, y_0]$ defining the starting point of the ascent
- ☐ `h` is a number used in the central difference formula
- ☐ `step_size` is the step size used in the ascent calculation
- ☐ `tol` is a number which gives the stopping condition: ascent stops when $||\nabla f|| <$ `tol`
- ☐ `max_iter` is the maximum number of iterations before the algorithm is forced to stop

The function returns 5 values:

- ☐ `xmax`, `ymax`, `zmax` are the coordinates of the local maximum found by the gradient ascent algorithm
- ☐ `total_time` is the total amount of time spent by the algorithm to find the local maximum
- ☐ `n_iter` is the number of iterations performed by the algorithm

The function performs the following tasks:

- ☐ Use `subplot` to create a figure with two subplots. In the first subplot position, plot the surface $z = f(x, y)$ over the region R. In the second subplot position, plot the contour plot of $z = f(x, y)$ using `contour`.
- ☐ Implement the gradient ascent algorithm:
  1. Plot the starting point (x_0, y_0) in both plots
  2. Write a loop to perform the following calculation given the current point (x_i, y_i)
     (a) Approximate the gradient $\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)$ using the central difference formula

$$\left.\frac{\partial f}{\partial x}\right|_{(x_i,y_i)} = \frac{f(x_i + h, y_i) - f(x_i - h, y_i)}{2h}$$

$$\left.\frac{\partial f}{\partial y}\right|_{(x_i,y_i)} = \frac{f(x_i, y_i + h) - f(x_i, y_i - h)}{2h}$$

     (b) Compute coordinates of the new point (x_{i+1}, y_{i+1}) by adding a `step_size` multiple of the gradient ∇f to the current position:

$$(x_{i+1}, y_{i+1}) = (x_i, y_i) + \texttt{step\_size} \cdot \nabla f$$

     (c) Plot the point (x_{i+1}, y_{i+1}) in both plots
     (d) Stop the loop when either $||\nabla f|| <$ `tol` or the number of iterations equals `max_iter`

3. If the algorithm stops by the condition $||\nabla f|| <$ `tol`, then the program should display a message indicating a local maximum was found
4. If the algorithm stops by the condition that the number of iterations equals `max_iter`, then the program should display a message indicating a local maximum was *not* found and the output values `xmax`, `ymax` and `zmax` should be the starting values defined by `start`

Write comments at the beginning of your function to describe its purpose, inputs, outputs and **include your name and student number**. When you have completed each item above and are satisfied with your function, submit your M-file (called `gradient_ascent.m`) to Connect.