# Computational Exercise One

for PHYS*2330 Electricity and Magnetism I

**This exercise uses python to visualize electrostatic fields: the electric potential, $V$, and electric field, $\vec{E}$. While these are continuous fields in space, computationally they will be handled *discretely* – that is, by dividing space into an array of uniformly separated points.**

**Learning outcomes:**

- **Gain competency in the use of numpy arrays as a form of data structure in python**
- **Become familiar with the general process of discretizing physical quantities and phenomena as needed for computational physics**
- **Explore different ways in python to visualize scalar and vector fields in two dimensions**

## Part (Done in Tutorial):

- We will begin by looking at how to plot 2D spatial data in python, using a ***meshgrid***. This is especially useful when plotting vector and scalar fields in E&M. We will write a python script, "***fieldPlot.py***" that computes and plots the electric potential for a distribution of point charges:
- We will create a function, ***potential***, that computes the electric potential at a point in space $(x, y)$ due to a point charge $q$ located at position $\vec{r}_q = (x_q, y_q)$.
- We will create a 2D array of points in the neighbourhood, $-5\ \mathrm{m} \leq x, y \leq 5\ \mathrm{m}$. Position coordinates for a regular array of points in space can be created using **numpy.meshgrid**. an array of 100 x 100 points should be sufficient for this exercise.
- Using the function, ***potential***, we will calculate the electric potential at all $(x, y)$ points above, due to an *electric dipole*, with charges $|q| = 1.0\ \mathrm{C}$, and length $d = 2\ \mathrm{m}$, located at the origin, oriented parallel to the x-axis.
- We will create a 2D plot showing lines of equipotential using **matplotlib.pyplot.contour**. Some attention should be given to how best to select the values for the equipotential lines. Be sure to format your plot with, title, labels, etc.

## Part B (Graded):

Next, add a function to the python script that computes the electric field for the same array of points in 2D space.

- Create a function, ***Efield***, that computes the electric field at a point in space $(x, y)$ due to a point charge $q_o$ located at position $\vec{r}_q = (x_q, y_q)$. Have your function output **x- and y-components** of the field (not magnitude & direction).
    - When creating the function ***Efield***, you will need to return two outputs. The final line of your function should list outputs separated by a comma:

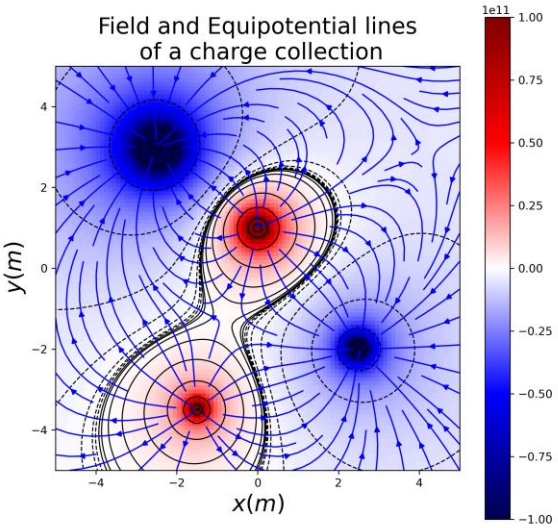
```
return result1, result2
```

    - When calling the function ***Efield***, use commas when assigning the outputs to two variables:

```
out1, out2 = Efield(q, rq, x, y)
```

- Create a configuration of four charges:
  - the 4 charges have values $-3, -1, +2$ and $+6$.
  - Position the charges on the vertices of a square of side length $d = 3\ m$
    - Center the square at position $x = -1, y = +0.5$.
  - There are many ways to arrange the charges on the vertices; **choose the arrangement that results in the maximum electric field pointing down-and-to-the-right**.
- Plot the electric field and the electric potential for your charge configuration.
  - You may choose to use some of the plotting styles shown: imshow, contour, streamplot, quiver.
  - You can plot the electric field and potential together on a single axis or use the subplot option to plot them on separate axes within the same figure.
  - Be sure to format your plot(s) with, colour, title, labels, etc.
  - With all the above choices, keep in mind the goal is to clearly communicate your results.
- Save your plot(s) as a single image file, "***fieldPlot.png***".

- **QUESTION B1:** The picture to the right shows a charge configuration. Use your code, with some trial/error guesswork to determine the location and the charge of the four charges.




## Part C (Give any of these a try! Not for marks):

The **matplotlib** library offers numerous ways to plot data. This can be very useful to help visualize and emphasize characteristics of your data. Explore the image gallery (https://matplotlib.org/gallery/index.html) and create a 3D representation of one of your calculated fields above. Save and submit your code and image as "***my3DPlot.py***"and "***my3DPlot.png***".

**REFLECTION QUESTION**: Did your computational work help your understanding of the physics content explored in this exercise?

**Assignment Submission:**

Your submission will consist of **two parts within Courselink**:

1) Question & Reflection 2330_Ex1: Under the "Quizzes" tab; you will enter your answers to questions and the reflection question within.
   - Also, if you learned any new python features or made efforts in some area (e.g. plot formatting, algorithms/logic, etc.), tell us about it in the Reflection space!
2) Once (1) has been completed, you will be able to Submit your completed work into the Courselink Dropbox folder titled "2330_Ex1: field plots". For this assignment, submit the following two files:
   - Python scripts: fieldPlot.py
   - Plots: fieldPlot.png

**This exercise will be graded out of 40 marks; 4 marks for your answer to Question B1; 36 marks will be awarded using the rubric posted in Courselink.**