Computational Physics: PHAS0030, 2018-2019
Problem Sheet 2

Hand in your answers by **Monday 18 February**, *on Moodle through the link provided. Marks per section are shown in square brackets. Your answers should be either a Jupyter Notebook or Python code. Ensure that you comment your code, and give every function a docstring.*

1. We will be examining the electrostatic potential and electric field due to various charges in two dimensions. For this question we will work in atomic units, so that $q_e = 1$ and $4\pi\epsilon_0 = 1$, and $\phi(r) = q/r$.

   (a) Create a 2D grid with x and y both extending between -5 and +5 and a spacing of 0.1 using `np.meshgrid`. (You will need to pass the optional parameter `indexing='xy'` so that the plotting works as expected below.) Calculate the potential for an electric dipole formed of a charge $+1$ at $(-1.05, 0)$ and a charge -1 at $(1.05, 0)$. [Hint: you can calculate r for all grid points relative to each charge with a single line of code - do NOT use a loop!] [4]

   (b) Plot the potential using `plt.imshow` and `plt.contour`. You can pass a normalisation parameter to `plt.imshow` using `norm=`; here, I found that `norm=colors.SymLogNorm(0.1)` worked very well (it makes the colouring logarithmic, and accounts for the negative numbers). You will need `import matplotlib.colors as colors` to use this. I also found it useful to pass an array of contour values to `plt.contour`; experiment with different settings until you are happy. [2]

   (c) Now create the electric field, $\mathbf{E} = -\nabla\phi$, using a finite difference approach (`np.roll` may be helpful). You should make the components E_x and E_y separately and plot them using `plt.streamplot(x,y,Ex,Ey)` where `x` and `y` are the arrays returned from `np.meshgrid`. You might like to explore the effect of using the optional parameter `color=` and passing the magnitude of the electric field. [If you choose to use `np.roll` for the difference, with the meshgrid arrays as specified, you will need to pass `axis=1` for the x-derivative and `axis=0` for the y-derivative. You should be able to calculate the electric field for this system analytically; you might choose to plot this to check your finite difference.] [4]

2. We will model the Hartree energy of an atom using a simplified model. The *radial* charge density will be given by $n(r) = Ar\exp(-2r)$ where A is a normalisation constant.

   (a) Using an initial spacing of `dr = 0.2`, plot $n(r)$ against r and decide on a sensible maximum range to use for the density. Explain your choice. [2]

   (b) Use `trapz` from the `integrate` module in `scipy` to integrate the density over r, and find the value of A that gives $2\pi\int drrn(r) = 1$. Be careful with the spacing of the grid: you must converge this integra. [3]

   (c) The Hartree energy will be calculated in 2D as $\int dxdyn(r)V(r)$, where $r = \sqrt{x^2 + y^2}$. Create a 2D grid of appropriate size and spacing (based on the extent of the density you chose above) and calculate $n(r)$ on all grid points for an atom placed at the origin. [2]

   (d) Now assume that $V(r) = -1/r$. Calculate the product $n(r)V(r)$ at all points on the grid and integrate by summing over grid points and scaling by the area per grid point. [Hint: if you get python errors, think carefully about what the product is and how you calculate it.] [3]

   (e) Explore the convergence of the total energy as the grid spacing is decreased. Comment on the balance between accuracy and time. [2]

3. The equation for heat flow, Q, in one dimension along a bar of varying cross-sectional area $A(x)$ and thermal conductivity κ is:

$$Q = -\kappa A(x)\frac{d\theta}{dx}$$

   where θ is the temperature. We will model a bar with $A(x) = 10^{-4}(2 - x)^2$, extending from $x = 0$m to $x = 1$m, with the temperature at $x = 0$ held: $\theta(x = 0) = 500$K. We will combine Q/κ into a single parameter.

   (a) For $Q/\kappa = 0.05$, write a function to evaluate the right-hand side of the appropriate differential equation. Now write a simple Euler solver for the temperature of the bar. Plot your solution and ensure that the spacing you use for the array of x values is reliable. [4]

   (b) Now using your function, solve for the temperature using `solve_ivp` from `scipy.integrate`. Plot your solution again. [Remember that the parameter `y0` passed to `solve_ivp` must be an array, so you may need to use `np.array([value])` to achieve this.] [2]

   (c) Now we will solve a different problem: the bar is now held so that $\theta = 500K$ at $x = 0$m and $\theta = 300K$ at $x = 1$m. Write a new function for the right-hand side of the equation that takes Q/κ as an argument. Use the secant method and `integrate.odeint` to find the value of Q/κ that solves this boundary value problem, and plot the final temperature distribution. [4]