Portfolio Assignment 2: Using python to solve boundary value problems representing plasma profiles

David Dickinson d.dickinson@york.ac.uk

Semester 1

1 Guidance

You should submit all of your code, figures and any working or written answers (this can be a scan of hand written work and does not need to be typed up, but should be legible).

It should be possible for the marker to run your code by typing `python <your_file>` (where `<your_file>` is the submitted code file name) and this should produce any relevant plots.

Some sample example results are shown at the end in order to allow you to check your implementation. These are intentionally not presented perfectly and you should make sure that any figures you produce are appropriately labelled and clear.

The code you submit should try to follow best practice including, but not limited to, clear structure, sensible variable names, avoiding duplication, clear structure and appropriate code comments.

Should you make a mistake in, or be unable to write, code that is used in a later part of the assignment the error will be carried forward as appropriate so that you are not penalised multiple times for the same mistake.

Should your results not agree with the sample results then you may use the sample results to inform your response to any interpretation or discussion questions if you wish. You should make it clear which set of results you are referring to.

2 Overview

In this assignment you are going to use Python and the SciPy library to explore toy models for steady state plasma pressure profiles in a tokamak.

We can model the evolution of a quantity, such as the pressure P, using an advection-diffusion equation such as

$$\frac{\partial P}{\partial t} = D\frac{\partial^2 P}{\partial x^2} + v\frac{\partial P}{\partial x} + S$$

movement through random dispersion

where the first term on the right hand side represents diffusion with D the diffusion coefficient, the second describes advection by a flow v and the third represents the net source or sink, S. Any other process aside from movement that changes the evolution of the quantity (how much of it there is)

movement through bulk movement/flow

In steady-state the pressure stops evolving in time (i.e. the left hand side is zero) and we can seek solutions subject to boundary conditions enforced at the plasma centre ($x = 0$) and the plasma edge ($x = L_x$).

3 Discretisation and implementation

D d^2P/dx^2 + x dP/dx + S = 0

Task [7 marks]: Write a python function which returns a sparse matrix representation of the discretised steady state transport equation given the x grid, the diffusion coefficient, D and the advection speed, v. You may assume that D and v are constant and that the x grid is uniformly spaced. You should impose a second order accurate Neumann boundary condition at the left boundary (where $x = 0$) and a Dirichlet boundary condition at the right boundary (where $x = L_x$).

Hint: Second order accurate one-sided differences are discussed in Lecture 5.

Task [2 marks]: Write a python function which takes the sparse matrix representation from the previous task, the net source, S, and the left boundary gradient and right boundary value and returns the steady-state pressure profile, P.

4 Application

Task [2 marks]: Suppose the net source is described by

$$S(x) = 100\left[1 - \tanh\left(-5(x - 0.75)\right)\right]$$

which represents an edge localised source. Use the functions you wrote earlier to find the steady state pressure profiling assuming $L_x = 1$ (i.e. x spans the range from 0 to 1), $D = 0.2$, $v = 0.2$. The left boundary condition should be a zero gradient whilst the right boundary condition should be a zero value. Plot the resulting profile.

Task [1 mark]: If v becomes $v = -0.2$ how does the core pressure change and why?

5 Extension

We can add a reaction term to the transport equation to give us some more control of the solution. Doing so, the equation becomes

$$\frac{\partial P}{\partial t} = D\frac{\partial^2 P}{\partial x^2} + v\frac{\partial P}{\partial x} + S + RP$$

where R represents a *reaction* coefficient.

Task [2 marks]: Extend your existing code to include this additional term. You may assume that R is a constant.

Task [1 marks]: Use your code to solve for the steady state pressure profile under the same assumptions and boundary conditions as previously, for the edge localised source, $D = 0.2$, $v = 0.2$ and for R taking each of the values 0.0, -0.1 and -0.2. Plot the three solutions on a single figure.

6 Example plots

To help you check your implementation the following figure shows an example result. It should be noted that whilst the figures shows "correct" data the presentation can be improved for full marks.

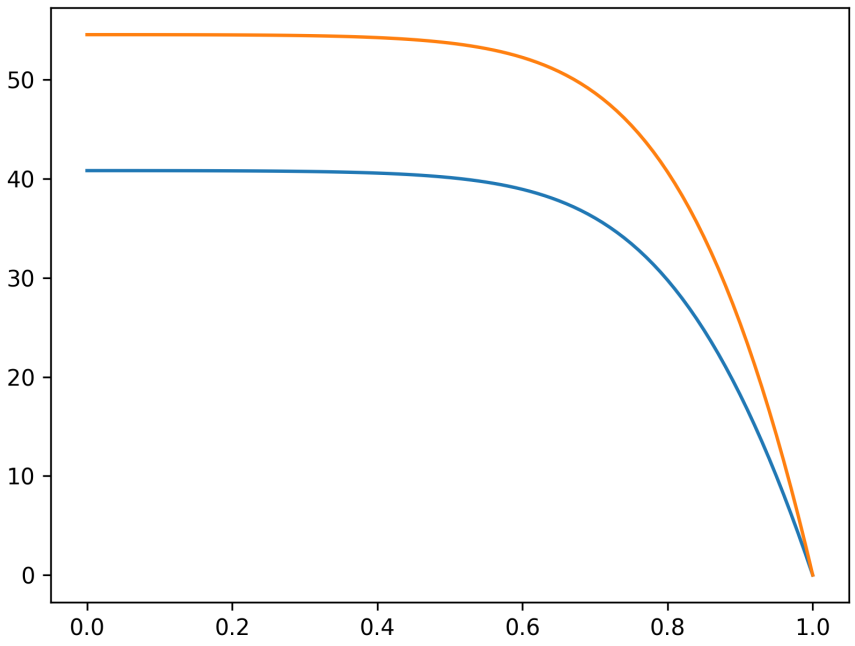

Figure 1: Plot of solution for different values of v.

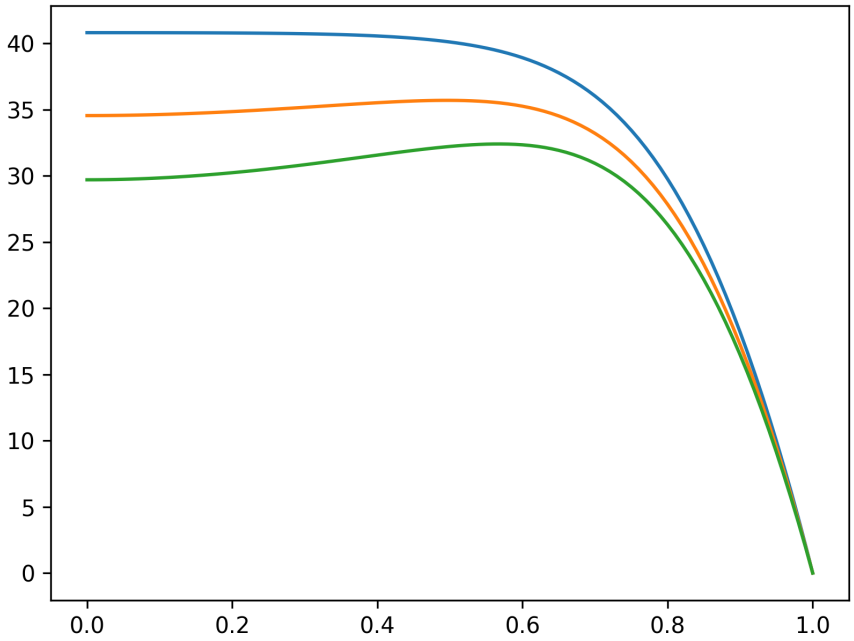

Figure 2: Plot of solution including a reaction term for different values of R.