Exercises for Practical #1

In this practical you will learn to construct matrices, solve linear systems, and and write basic Python code in a Jupyter Notebook.

1. Create a Juypter Notebook called 'problems1.ipynb' and add a suitable markdown section at the top. Use the notebook to write down your solutions to the following problems.

2. Check the help for `np.diag` and use it to construct the 16×16 matrix, with $h = 2/16$:

$$D2 = \frac{1}{h^2} \begin{pmatrix} -2 & 1 & 0 & 0 & \dots & 0 & 1 \\ 1 & -2 & 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & \dots & 0 & 1 & -2 & 1 \\ 1 & 0 & 0 & \dots & 0 & 1 & -2 \end{pmatrix} \tag{1}$$

   We will be using this matrix to solve ODEs on the interval $[-1, 1]$. In general, for a matrix of size $N \times N$, we have $h = 2/N$.

3. Close juypter lab, install scipy in your conda environment, and reopen juypter lab. Read about `scipy.linalg.toeplitz`. Use it to construct `D2` again for `N` = `16`.

4. Adapt the code from questions 2 and 3 to allow for an `N` × `N` matrix (replacing also h by `2/N`), where `N` is a variable which can be easily adjusted.

5. For $N = 16$, create a vector x containing N evenly spaced points in the interval $[-1, 1)$ (including -1 but not 1). Define a function $f(x) = \sin \pi x$. Compute `D2@f` and plot it against `x`.

6. What does the above graph show? (Hint: google "finite difference".) Amend your markdown at the start of the notebook accordingly. Plot (on a `subplot`) the error in the `D2@f` approximation.

7. Let $f(x) = e^{\sin \pi x}$. Using the same `x` as in problem 5 with $N = 256$, compute `D2@f` and plot it on top of a plot of $f(x)$. Now compute the second derivative of f by hand! Define a lambda function for the second derivative, `fpp = lambda x:  ...`. Plot `fpp(x)` on top of the existing plots. Now plot `D2*f - fpp(x)` in a new figure.

8. With `D2`, `x` and `f` **as in problem 5**, define

```
A = (D2 - eye(N))/(1+pi^2)
```

   solve $Au = f$ and plot `u` against `x`. (Congratulations!! You have just solved your first differential equation in Python! — do you know what it was?)

9. (Advanced) Repeat all of the above using *sparse* matrices. (Hint: check out the `scipy.sparse` sublibrary.)

10. (Advanced) What is the closed-form expression for the eigenvalues of the matrix `D2`? Compare with a numerical approach in Python.