Introduction

We present an introductory course on using `FiPy`, a PDE solver in python. It is hoped that the exercises in the module will expose the user to both the syntax needed to solve a problem of interest and also certain mathematical and numerical features that provide insight on general issues related to numerically solving PDEs.

There are a variety of PDE solvers native to python out there:

- FiPy
- Fenics
- Dedalus ...

In this repository you will find the following folders:

- `Setup`: This folder contains documentation relating to the installation of Python on your laptop which will be required to run the various exercises. If you have not installed it already, it is recommended you go through this folder first.
- `EX1`: The first exercise in this repository will re-introduce the concept of PDEs through the familiar Laplace equation and the syntax used within `FiPy` to solve such an equation numerically. Details on how to visualise the results are also provided.
- `EX2`: Moving on to the transient-diffusion problem (and its variants), this exercise will look at how transient problems are treated differently from steady-state ones. An introduction to the concept of numerical stability is also given.
- `EX3`: Now looking at the more complex convection-diffusion-reaction problem, time is spent examining the formulation of such a problem as well as discussing the complexities associated with modelling convection. An example problem is given, solved and visualised within the exercise.
- `EX4`: As you would now be familiar with how PDEs are solved within `FiPy`, a more complicated but very relevant problem areas of research, the Cahn-Hilliard equation will be introduced and solved in this exercise. Optional information as to how one can visualise solutions of such problems in Paraview is also given.
- `Figures`: A directory that contains files needed to generate the various plots and figures found in the notes. It is a bit messy, but you can ignore it.
- `Sample_exercises`: Modified versions of the exercises given above are provided with additional complexities.

The course is aimed at people who have a basic understanding of PDEs and vector calculus such as 2nd year students and above. Extensive notes are provided for each exercise. Basic notes on how to set-up a python environment on Windows and how to use Paraview for post-processing are also available.