Tutorial 9

Exercise 1

Write your 2D Burger's equation solution in a form that:

1. Modular, i.e. each component of the code is separated from the others. You can switch in and out the different components, and each component can be reused (without copying and pasting!).
2. Writes out the outputs so that evaluation of the outputs can be done separately.
3. (optional) Try to use the git integration in Visual Studio Code to commit and push your solution.
4. (optional) Try to make use of classes to gain familiarity with it.

Debugger

We will use the `pdb` debugger in Visual Studio Code:

1. Open Visual Studio Code and go to the `Run and Debug` tab (ctrl+shift+D).
2. Click the dropdown menu and select `Add configuration...`.
3. This will open a `.json` file. Under `"program"`, enter the path your run file. For example, mine is

```
"program": "${workspaceFolder}/w9/heat_eqn/main.py",
```

4. Add a new line that tells the what arguments the run file needs. For the heat equation example,

```
"args" : ["-ic", "gauss"],
```

5. Now, pressing `F5` or clicking the *run* symbol in the `Run and Debug` menu will start the debugger.

Using the debugger:

1. Breakpoints: For to a line in the code, e.g. line 10 of `expl_mdpt.py`. Click on the left of the line number. You will see a red dot appear. This is a breakpoint, and it tells the debugger to stop at this of the code after you run the debugger. Stopping here will allow you to inspect your code.
2. `Run and Debug` tab: Demonstration in class.
3. Debug console: At any point of the code where you have a breakpoint, you can also use the debug console to, for example, plot a variable.

Profiler

Since we have setup our code such that it can be run from the terminal, using a profile is as simple as running the following line of code:

```
python3 -m cProfile main.py -ic gauss
```

This will print the output of the profiler to the console. A more user-friendly approach would be to save the output, then visualise it separately. To do this, first run

```
python3 -m cProfile -o profile.dat main.py -ic gauss
```

This saves the output to a file called `profile.dat`. A few weeks ago, we installed `SnakeViz`. This is a useful tool to visualise the output of the Python profiler:

```
snakeviz profile.dat
```