Chapter 4

Code Description

Code Structure

The solver was coded in C to take advantage of the faster execution times due to the lower level nature of C. The source code consists of 7 files and a header. A brief description of each source file and the functions contained within is presented here.

- **`main.c`** : Calls functions to take data input, displays it and passes the input to function that calculates solution
- **`get_param.c`** : Contains functions to either read data from file or input it at runtime. Also contains function that checks for stability depending on the input variables.
- **`solver.c`** : Contains function definitions for the discretised equations for each state as well as a routine to call them at appropriate times and write the solutions to file. Also includes a function to dynamically allocate memory to store store solution variables based on the size of the grid.
- **`liner_solver.c`** : Solves a linear system of equations using LU Decomposition algorithm.
- **`write_data.c`** : Contains functions that write data to file in Tecplot formatted data or in simplified form for MATLAB import.
- **`ovs.c`** : Contains all the necessary functions to carry out the Order Verification Study via the Method of Manufactured Solutions.
- **`mkdir_p.c`** : Tool to create another directory in which to save results, to avoid filling up the case directory with result files.

All function declarations are contained in the header **`"cfd_hs2017.h"`**.

Compilation A **`makefile`** is included with the source code to make compilation as easy as typing *`make`* from inside the case directory. This generates an executable *`run_sol`* which can be run to start the solver.

User Input & Execution The code requires the user to input the various physical parameters to execute. This can be done in two ways:

- Enter parameters at run time
- The code can also read parameters from the file `parameters.txt` which is a specifically formatted text file that the code can read. The file should be edited making sure to not alter the order of the parameters in anyway.

Which of these two methods to use can be chosen at runtime.

Other parameters that can be chosen during runtime are:

- Number of grid cells
- Timestep size (in seconds), Δt
- Whether to save a condensed data file or individual files for each time step
- How often to save results to file
- Total number of cycles for which to run the simulation

Output The code outputs **`.dat`** files which can be easily exported to MATLAB, or visualised using Tecplot. The headers required for the files to be read in Tecplot are hard-coded. Using the Tecplot format allows for easier visualisation as the different time steps are recognized as different zones by Tecplot and these can be easily switched between using the GUI.
The output file contain three variables: *position*, *temperature solid*, and *temperature liquid* and are stored in the following tab separated format:

Position (in m)	Solid Temperature	Fluid Temperature
...	...	...
...	...	...
...	...	...
...	...	...