ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Point Implicit Solver for 1D Two-Phase Thermal Energy Storage

Semester Project

Abhimanyu Bhadauria

December 13, 2017

Submitted to: Prof. Dr. A. Haselbacher

Department of Mechanical & Process Engineering, ETH Zürich

Abstract

In this project we attempted to code a 1D coupled two-phase solver for sensible thermal energy storage (TES) system using the Finite Volume Method. To develop the code, we started with the uncoupled system before moving on to coupled system. The code developed was verified at each step with an Order Verification Study (OVS) carried out using the Method of Manufactured Solutions (MMS). The final code was tested once with the exact solution obtained from a simplified version of the governing equations. 5 simulations were carried out by varying the geometric dimensions of the TES system, and the *exergy flux* and *capacity factor* were reported for each case.

Contents

Chapter 1

Introduction

A sensible thermal energy storage system stores energy by raising the temperature of a dense, high heat capacity material. Often the energy required to do this is obtained from renewable sources such as wind or solar. The TES enables us to store excess energy and use it when it is not being generated. For example, in the case of a solar based TES, it can be charged during the day and discharged (used) at night.

1.1 Thermocline Heat Storage

For this project we wish to simulate a *thermocline*, a thermal storage medium in which hot and cold sources co-exist leading to a temperature gradient. The thermocline simulated in this project is heated with molten salt, which acts as the heat transfer fluid, and the thermal storage material that is being heated (or charged) is rocks. A thermocline cycle then consists of

1. charging state in which hot heat transfer fluid flows into the system from one side, thereby raising the temperature of the rocks.
2. an idle state after the charging state, where no more fluid flows into the system
3. a discharging state in which cold heat transfer fluid flows into the system from the other side, thereby cooling the storage medium
4. another idle state following the discharging state

1.2 Governing Equations

The system defined above is governed by two coupled partial differential equations that can be solved for the temperature of the solid and liquid phases. The equations are stated below.

$$\varepsilon\rho_f C_{p,f}\frac{\partial T_f}{\partial t} + \varepsilon\rho_f C_{p,f}u_{f,i}\frac{\partial T_f}{\partial x} = k_f\frac{\partial^2 T_f}{\partial x^2} + h_v(T_s - T_f)$$

$$(1-\varepsilon)\rho_s C_s\frac{\partial T_s}{\partial t} = k_s\frac{\partial^2 T_s}{\partial x^2} - h_v(T_s - T_f)$$

1.3 Finite Volume Discretisation

The coupled partial differential equations are discretised using a finite volume method on a uniform collocated grid. The discretisation is carried out using a Forward Time Backward Space (FTBS) scheme which is an explicit scheme in time. In the first step we decoupled the equations by removing the coupling therm h_v. The discretised equations hence obtained (for the interior cells) are as follows:

$$T_{f,i}^{n+1} = T_{f,i}^n - \frac{u_{f,i}\Delta t}{\Delta x}(T_{f,i}^n - T_{f,i-1}^n) + \frac{\alpha_f\Delta t}{\Delta x^2}(T_{f,i+1}^n - 2T_{f,i}^n + T_{f,i-1}^n)$$

$$T_{s,i}^{n+1} = T_{s,i}^n + \frac{\alpha_s\Delta t}{\Delta x^2}(T_{s,i+1}^n - 2T_{s,i}^n + T_{s,i-1}^n)$$

Here we drop the cell average notation as our nominal order of accuracy is 1.

To solve the full coupled equations, we use the *point-implicit method* where we treat just the coupling term implicitly. This allows us to use the FTBS discretisation scheme developed previously in the last step. The solutions of the explicitly discretised equations at time $n+1$ are treated as intermediate solutions $T_{f,i}^*$ & $T_{s,i}^*$. A linear system involving the coupling term is now solved to arrive at the *actual* solution at time $n+1$. The linear system hence solved is described below.

$$\Delta t[(1+h_{v,f})T_{f,i}^{n+1} - h_{v,f}T_{s,i}^{n+1}] = T_{f,i}^*$$

$$\Delta t[-h_{v,s}T_{f,i}^{n+1} + (1+h_{v,s})T_{s,i}^{n+1}] = T_{s,i}^*$$

1.4 Physical Parameters

For all the simulations carried out in this project, the following physical and thermodynamic parameters were used, unless otherwise mentioned.

V	$300m^3$
d_s	$0.03m$
ε	0.4
$\dot{m}$	$10kg/s$
rho_f	$1835.6kg/m^3$
$C_{p,f}$	$1511.8J/kg-K$
μ_f	$2.63kg/m-s$
k_f	$0.52W/m-K$
rho_s	$2600kg/m^3$
C_s	$900J/kg-K$
k_f	$2W/m-K$

Chapter 2

Code Verification

2.1 Order Verification Study (OVS)

In this section, a discussion of the OVS is presented to confirm that the code does in-fact adhere to the nominal order of accuracy we expect from it. For all the verification studies presented here, the method of manufactured solutions (MMS) employing trigonometric functions was used.

2.1.1 MMS Study 1: Uncoupled System

The uncoupled system was analysed by taking the manufactured solution $cos(\frac{2\pi nx}{L})$. The study was carried out for both the solid and fluid equations by varying the values of wave-number, n, and Peclet Number, Pe, over a large range.

Fluid Equation Figures 2.1, 2.2, & 2.3 show the study results for the low values of Pe. This is the case when the flow is dominated by diffusive effects rather than by advective transport. The nominal order of accuracy of the discrete equations here is 1. The OVS can be seen to agree well with the nominal order of accuracy here, leading us to believe no errors were made while coding the discrete equations.

In Figures 2.4 & 2.5, the results of the OVS for higher values of Pe are plotted. This is represented by cases where the transport phenomena is largely dominated by advective effects. At this point, we start to notice deviation from ideal conditions. The order of accuracy plot for $Pe = 213,280$ deviates significantly from 1. In fact, it is observed that as we further increase the n the order of accuracy starts to deviate from the ideal case for successively lower values of Pe.

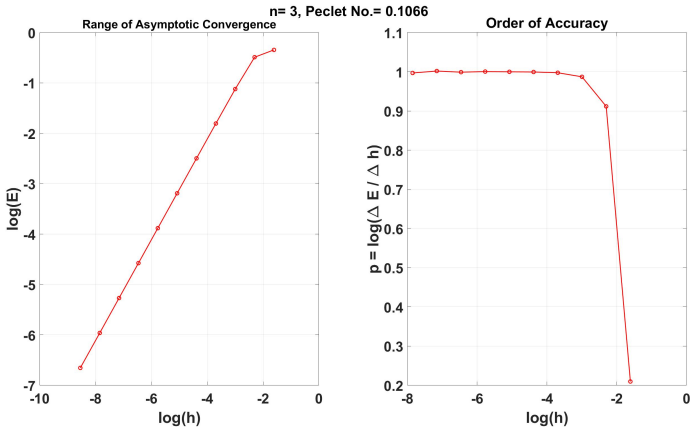

Figure 2.1: Pe = 0.1066, n=3

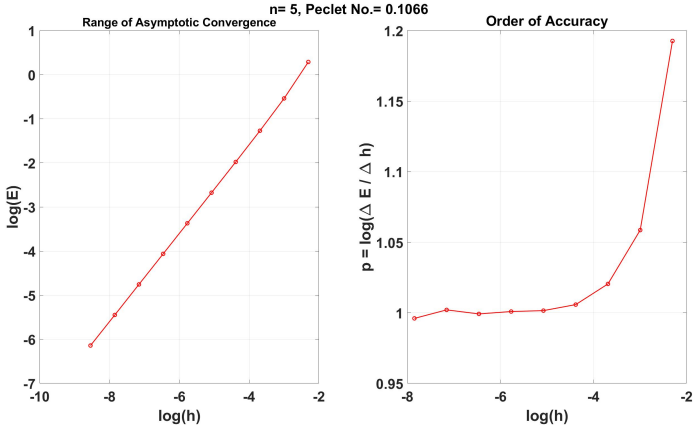

Figure 2.2: Pe = 0.1066, n=5

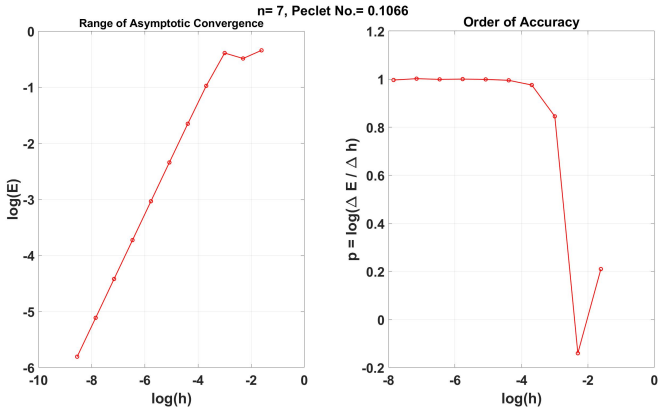

Figure 2.3: Pe = 0.1066, n=7

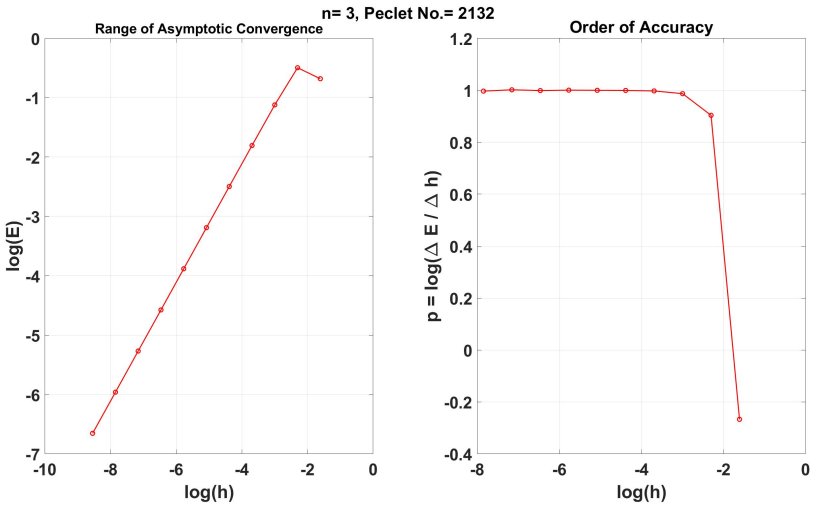

Figure 2.4: Pe = 2132, n=3

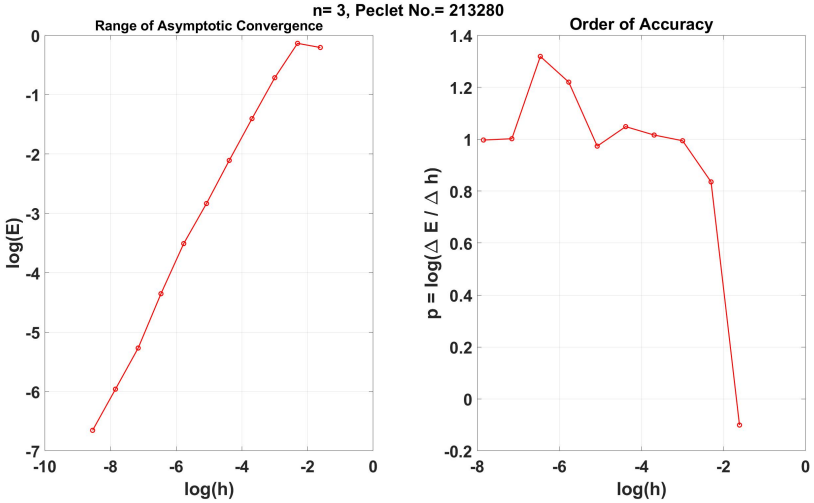

Figure 2.5: Pe = 213280, n=3

Solid Equation Figures 2.6 and 2.7 show the OVS plots for the solid equation obtained for the uncoupled case. However, for the solid equations, the nominal order of accuracy of 2 was not obtained. All the studies reported a steady order of accuracy of 1. One possible explanation for this could be that the temporal discretisation, which is forward step and first order accurate in time, may be contributing to the bulk of the discretisation errors, thereby limiting the order of accuracy to 1.

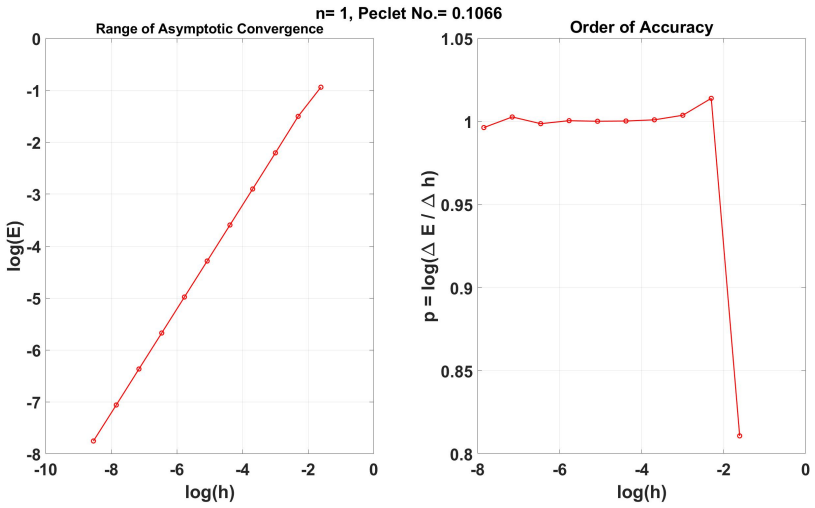

Figure 2.6: Pe = 0.1066, n=1

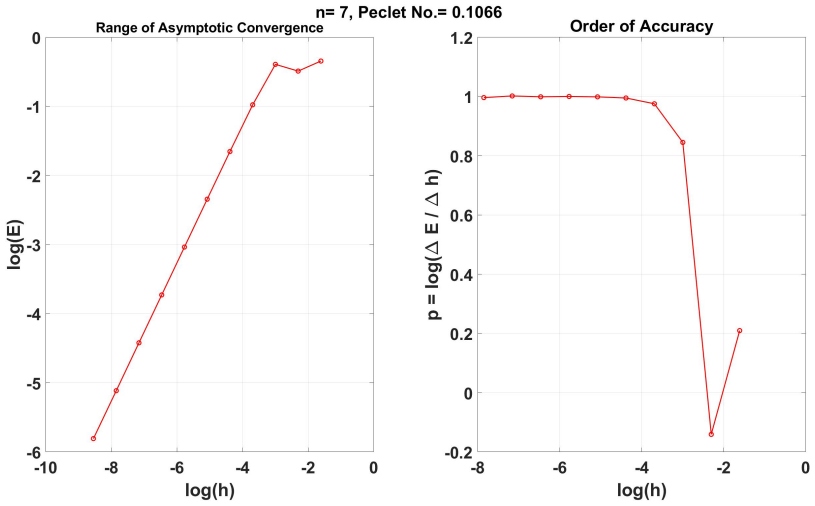

Figure 2.7: Pe = 0.1066, n=7

2.1.2 MMS Study 2: Coupled System

For the coupled system of equations, two different functions were used as manufactured solutions, one for each equation: $cos(\frac{2\pi nx}{L})$ was used for the fluid equation, while $sin(\frac{2\pi nx}{L})$ was used for the solid equation. The same approach as earlier was used during this study, *i.e* the tests were carried out for varying values of Pe and n.

Fluid Equation

The coupled system was developed on top of the uncoupled system that was tested in the previous section, and hence the nominal order of accuracy remains 1. The OVS confirms this and the results are plotted for a wide range of *Pe*. Figures 2.8 & 2.9 show the OVS results for small values of *Pe* at different *n* values.

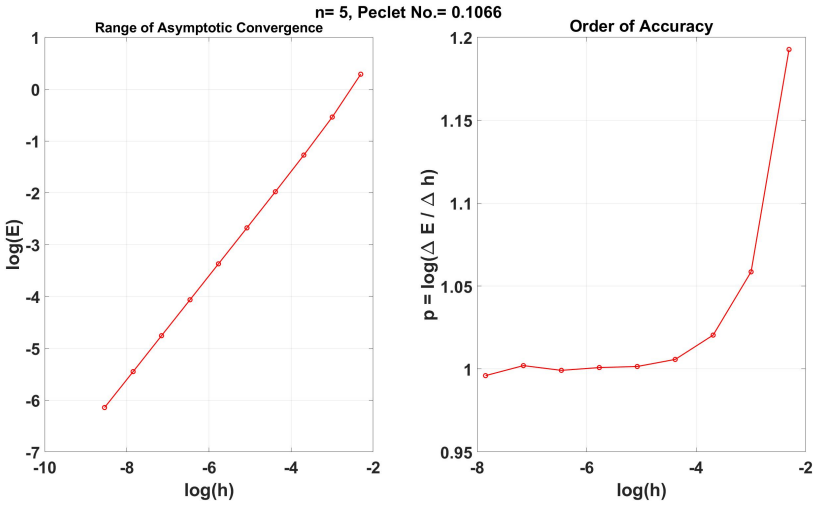

Figure 2.8: Pe=0.1066, n=5

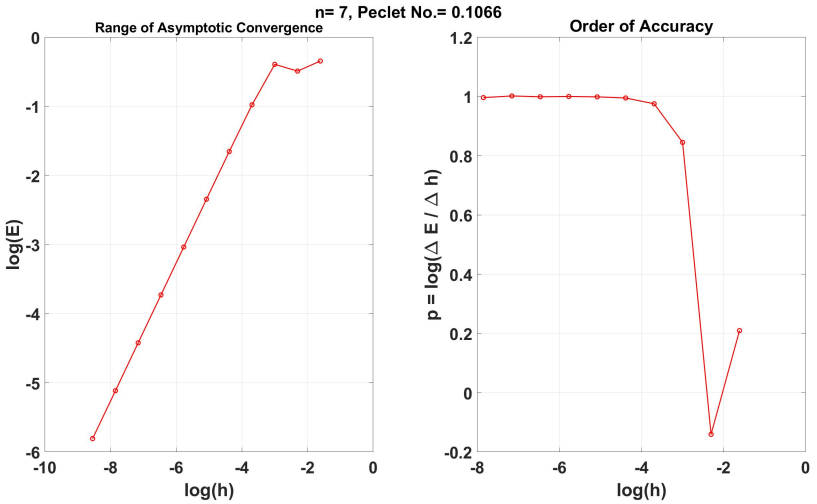

Figure 2.9: Pe=0.1066, n=7

The OVS results for higher values of *Pe* can be seen in Figures 2.10 & 2.11. Once again we can observe the same phenomena where the order of accuracy deviates from the nominal as we increase *n* and *Pe*. The highly oscilla-

tory nature of the input signal, means our first order accurate approximation is unable to keep up with the signal.

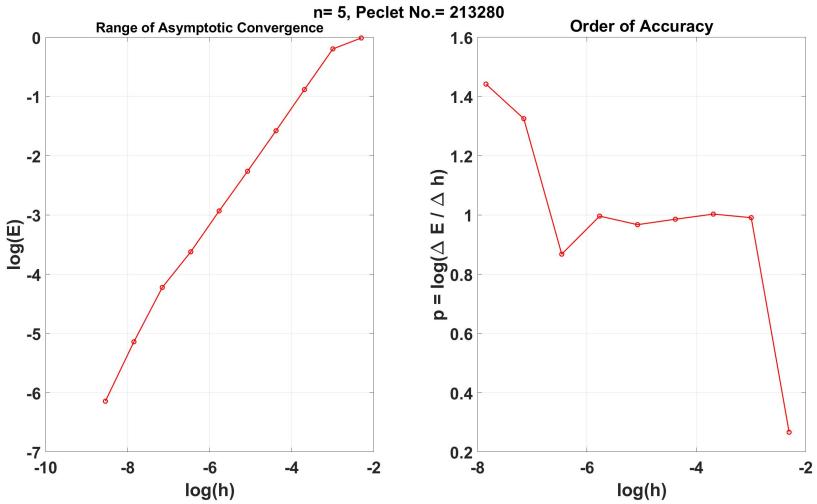

Figure 2.10: Pe=213280, n=5

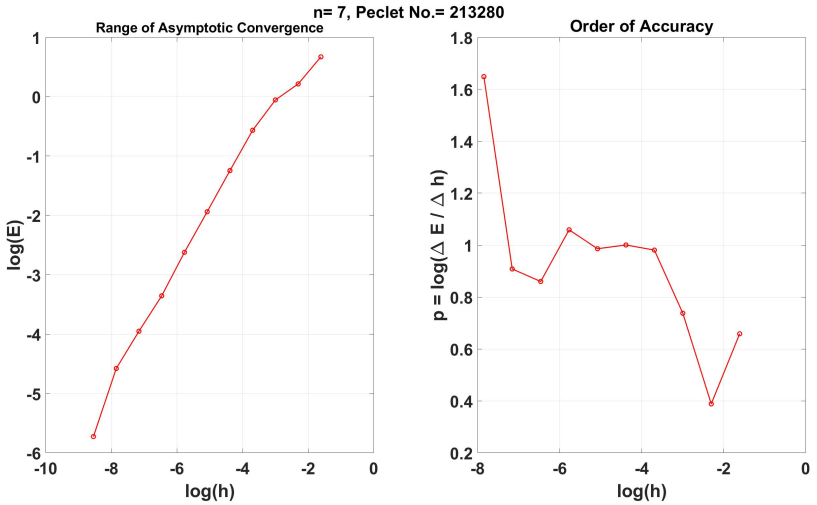

Figure 2.11: Pe=213280, n=7

Solid Equation

For the coupled solid equation, we now expect a nominal order of accuracy of 1 due to coupling with the liquid equation, which is first order accurate. This is indeed confirmed by the OVS plots,as can be seen in figures 2.12 & 2.13.

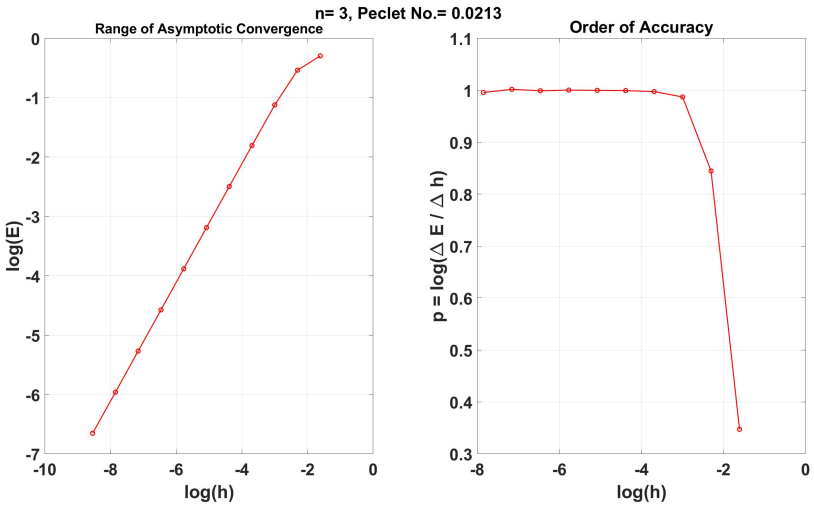

Figure 2.12: Pe=0.0213, n=3

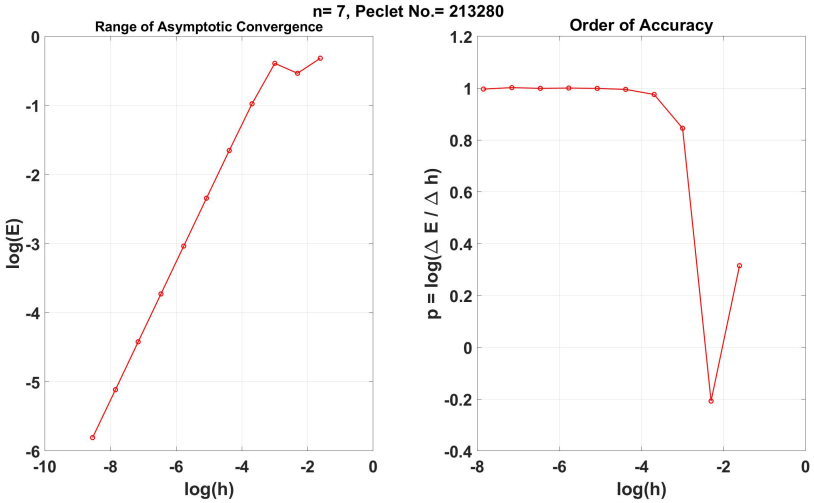

Figure 2.13: Pe=213280, n=7

2.2 Code Validation

The exact solution of our coupled system of equations, for the charging state, was obtained at 5000 seconds. To check the correctness of our code, it was run for the same time with identical physical parameters and the result compared to ensure that the solution of the discretised equations actually satisfy the actual physical system and are not spurious.

Figure 2.14 shows the result of this comparison. Both the exact solution and our solution of the discrete equations are plotted together, for both fluid

and solid medium. It can be concluded that our code is able to effectively capture the physical behaviour of the system, as can be seen from the nearly overlapping plots.

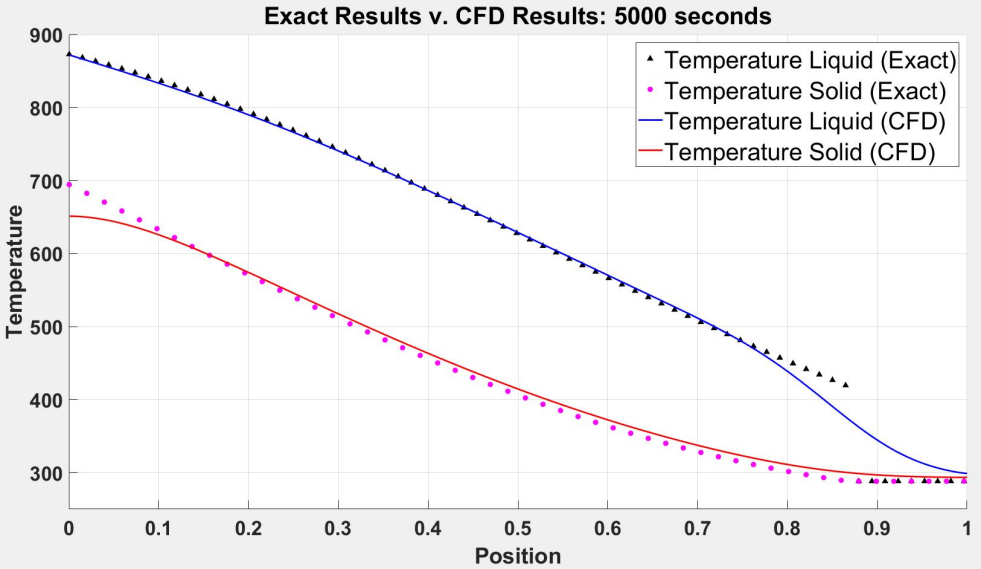

Figure 2.14: Exact Solution vs. Discretised Solution : Charging after 5000s

The sudden jump in the exact solution that is not captured by our code, is in-fact due the exact solution being derived from a simplified set of governing equations which neglect the second derivative. This leads to the exact solution having no diffusion effect at all. Our code on the other hand calculates the diffusion term and is also affected by numerical diffusion due to a biased approximation for the advection term. Therefore, our numerical solution is smoothed out where the exact solution exhibits a sharp drop in liquid temperature.

Chapter 3

Results

3.1 Overview

Five cases were simulated by varying the physical geometry of the cylindrical storage medium while keeping the volume fixed at $300m^3$. The various parameters that change for each case are as follows:

	Case 1	Case 2	Case 3	Case 4	Case 5
D	4.000000	5.000000	6.000000	7.000000	8.000000
L	23.873241	15.278875	10.610330	7.795344	5.968310
u_f	0.001084	0.000694	0.000482	0.000354	0.000271
h_v	1120.818462	835.392540	656.591698	535.421410	448.587788

Apart from the temperature evolution of the heat transfer fluid and solid over time, two main parameters were calculated:

1. Cycle Exergy Efficiency, η

2. Capacity Factor, Q

3. Temperature Rise at Outflow at the end of charging state, ΔT

After an initial simulation run was found to converge after 15-20 cycles, all the cases were to run for 51 cycles to ensure the invariance of exergy efficiency with the number of cycles simulated. The results of the study are presented here, along with the temperature evolution of the fluid. The plots for solid temperature evolution can be found in the Appendix.

3.2 Case 1: Diameter 4m

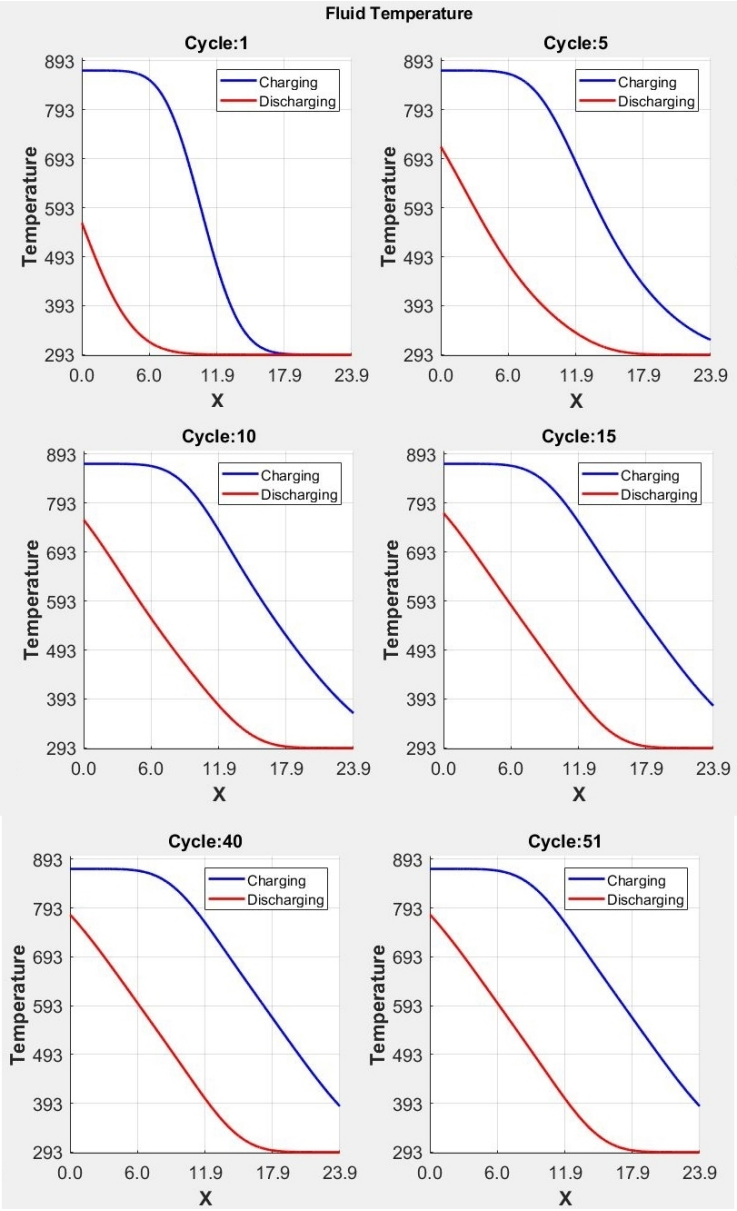

Figure 3.1: Temperature Evolution

- $\eta = 0.9576$
- $Q = 0.4168$
- $\Delta T = 93.976K$

3.3 Case 2: Diameter 5m

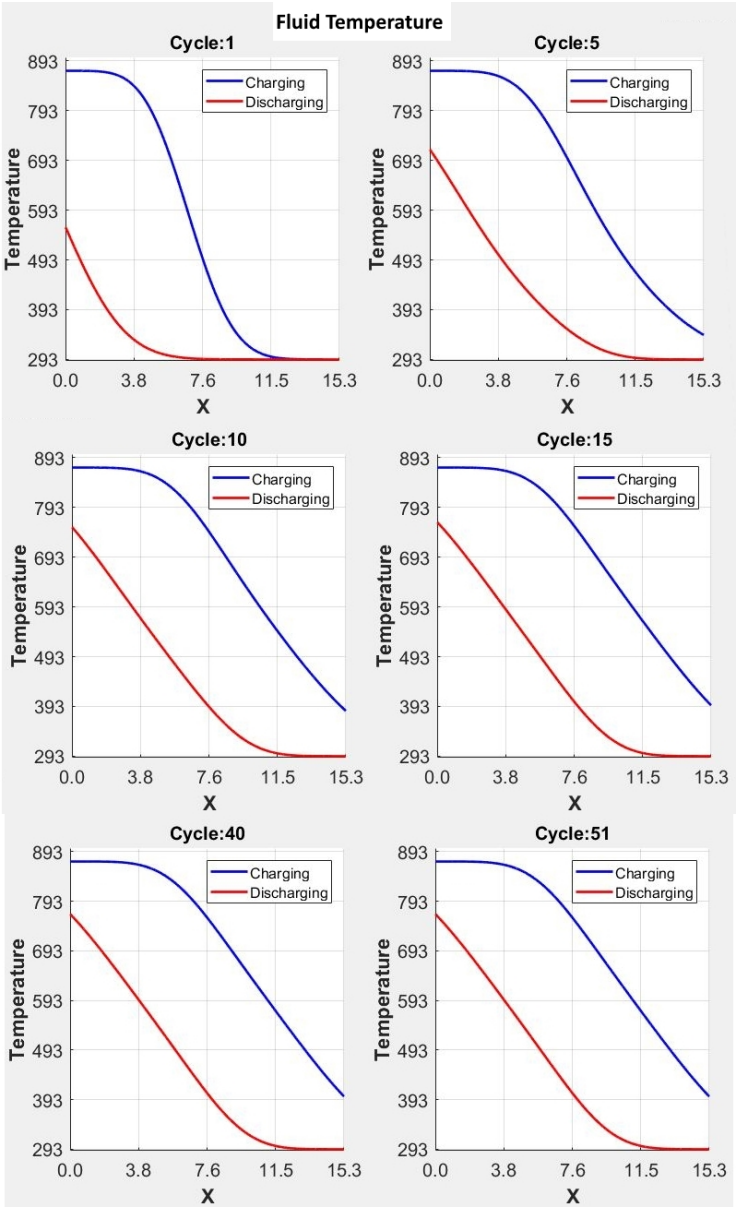

Figure 3.2: Temperature Evolution

- $\eta = 0.9460$
- $Q = 0.4121$
- $\Delta T = 106.456K$

3.4 Case 3: Diameter 6m

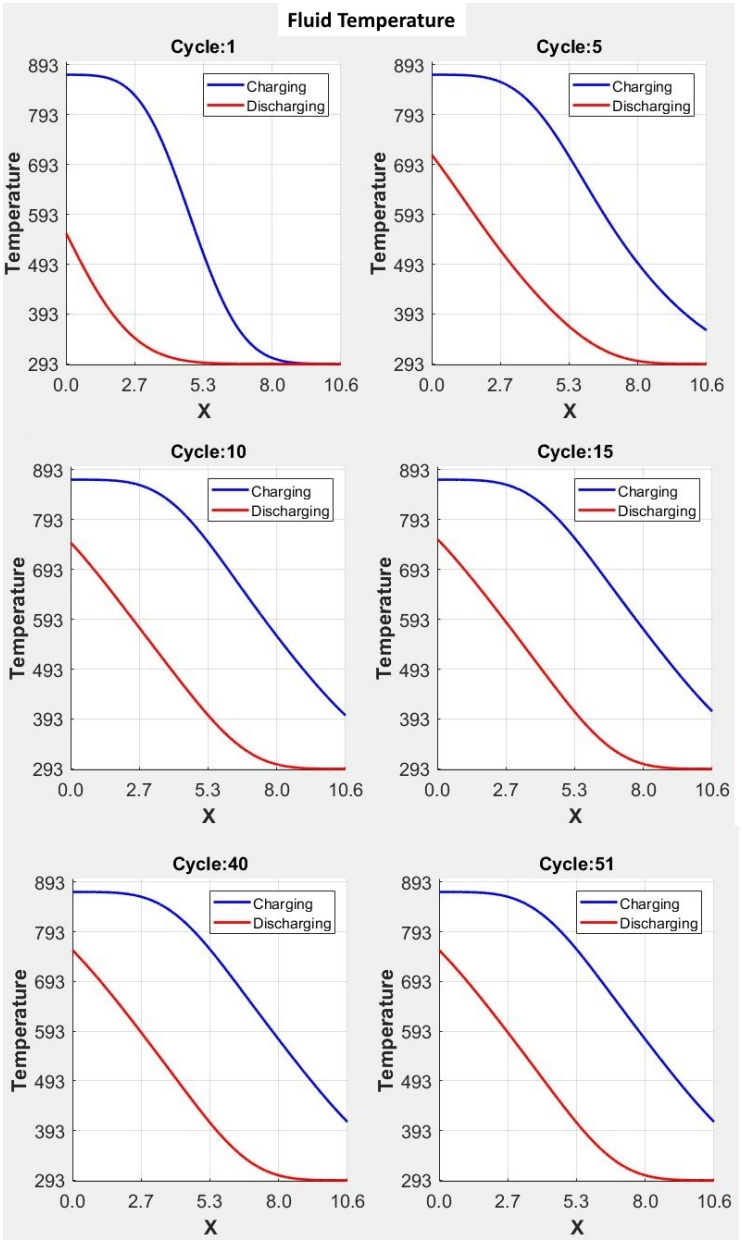

Figure 3.3: Temperature Evolution

- $\eta = 0.9343$
- $Q = 0.4073$
- $\Delta T = 117.732K$

3.5 Case 4: Diameter 7m

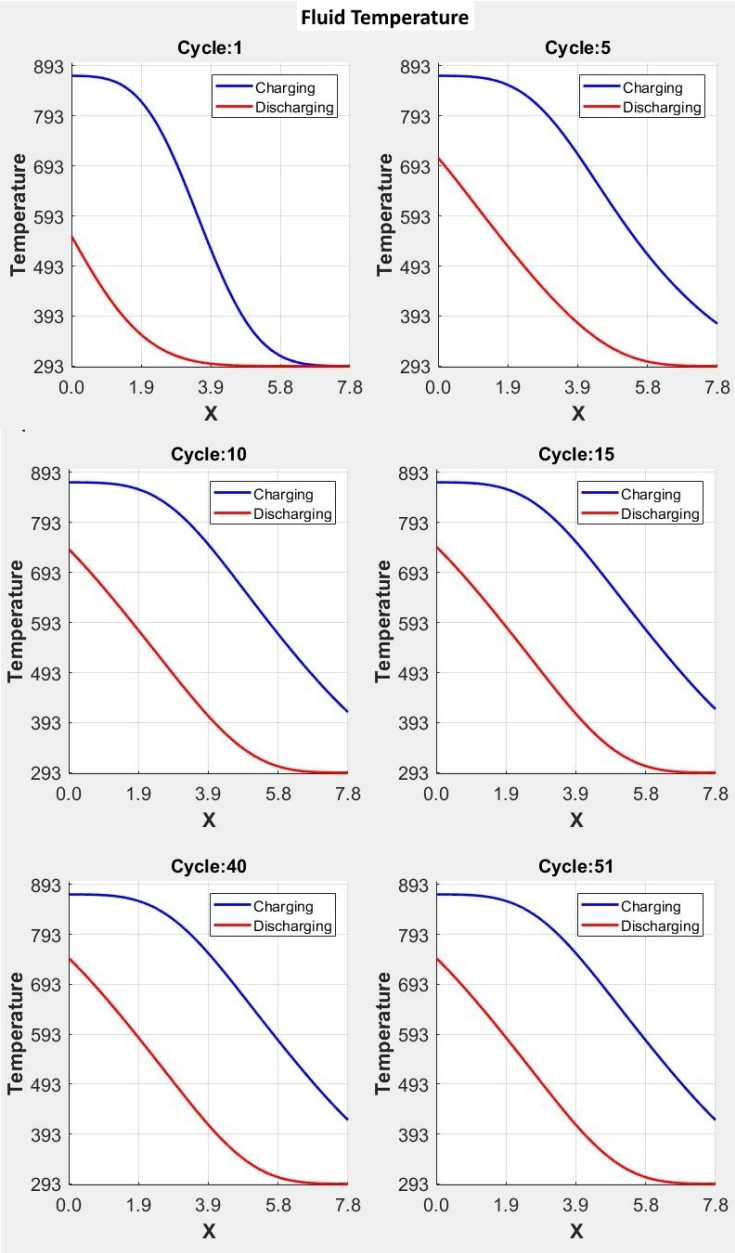

Figure 3.4: Temperature Evolution

- $\eta = 0.9224$
- $Q = 0.4021$
- $\Delta T = 128.248$

3.6 Case 5: Diameter 8m

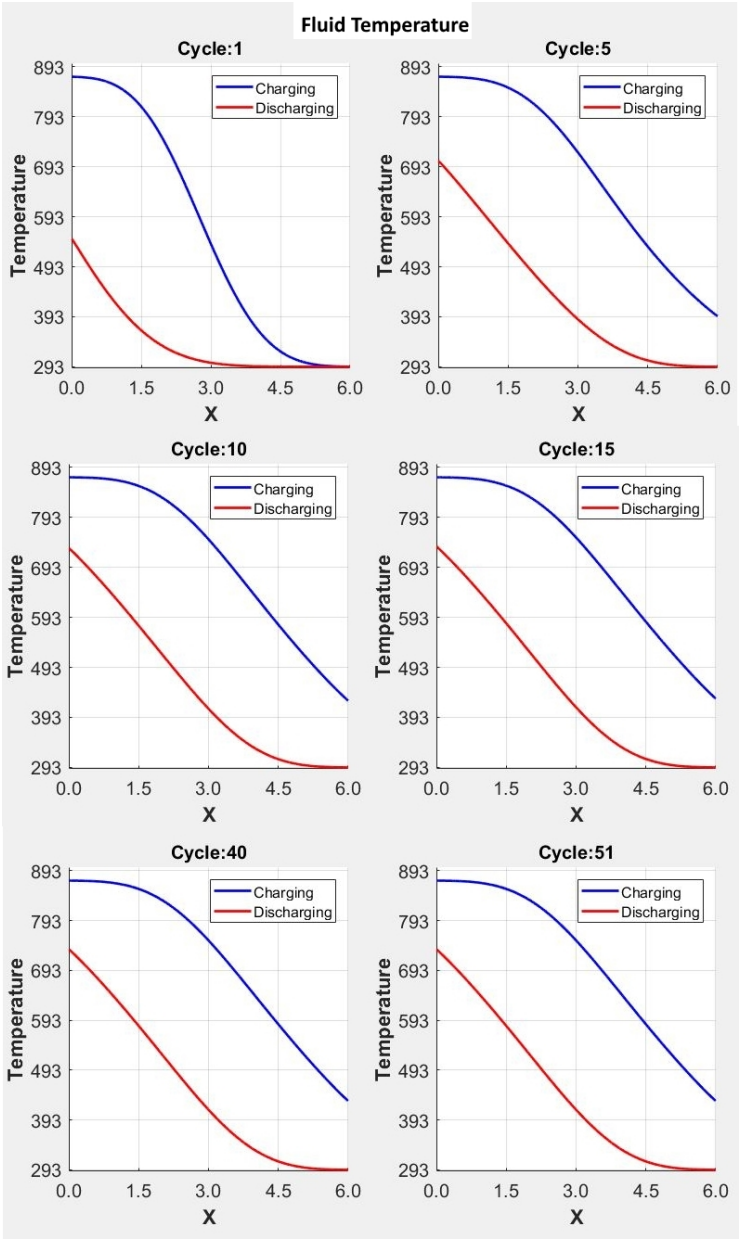

Figure 3.5: Temperature Evolution

- $\eta = 0.9103$
- $Q = 0.3969$
- $\Delta T = 138.2060$

3.7 Discussion

A comparison of the various parameters obtained from the design study are was done and the results plotted.

The exergy efficiency is observed to reduce linearly with a corresponding increase in the diameter of the storage medium. This hints at the fact that for the same volume, a longer/deeper storage device may be preferred over one that has more girth but is shallow. Figure 3.6 shows this observed drop in η with an increase in diameter.

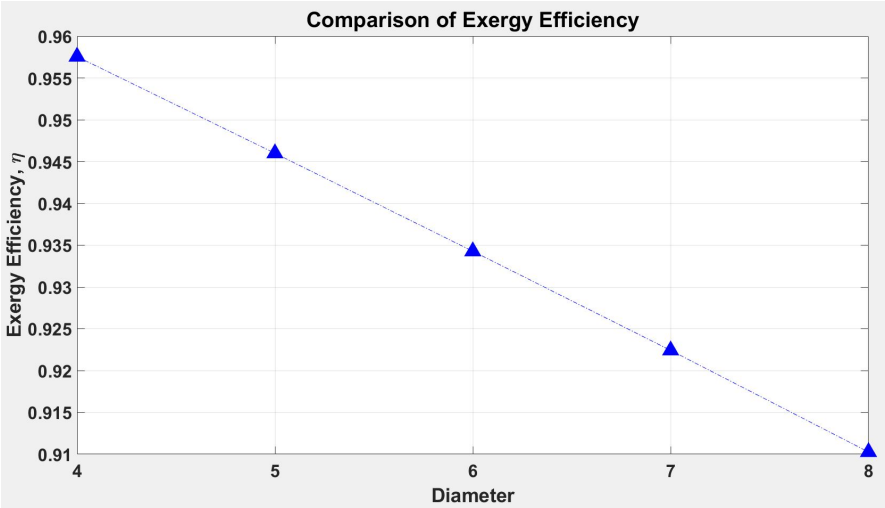

Figure 3.6: Exergy Efficiency

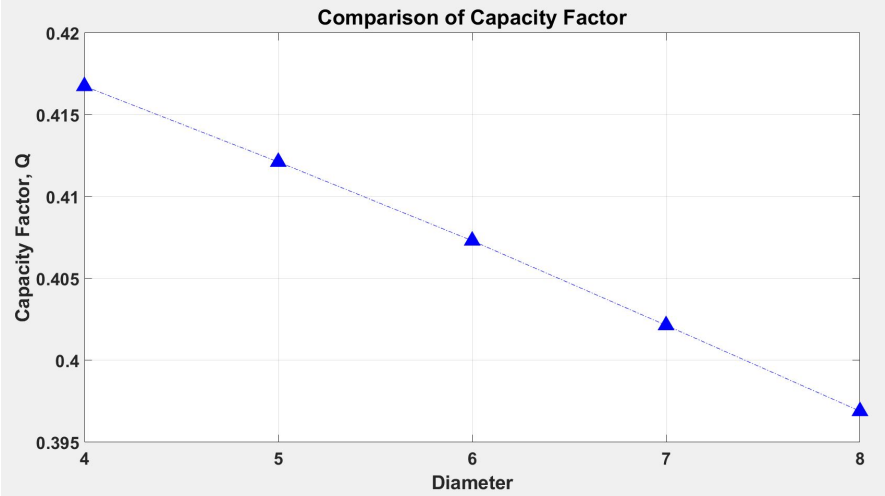

Figure 3.7: Capacity Factor

Figure 3.8 shows that the capacity factor exhibits a similar trend, also decreasing linearly with increase in diameter. However, the temperature increase at outflow at end of the charging state (Figure 3.7) exhibits diametrically opposite behaviour to both exergy efficiency and capacity factor, increasing linearly with rising diameter. One is tempted to attribute this to the fact that a larger diameter leads to a shorter/shallower device, however it is important to remember that the velocity for each case is scaled to ensure a fixed mass flow. Therefore, this must be attributed to the exergy efficiency rise and one can say that a higher exergy efficiency leads to a smaller rise in temperature or vice-versa.

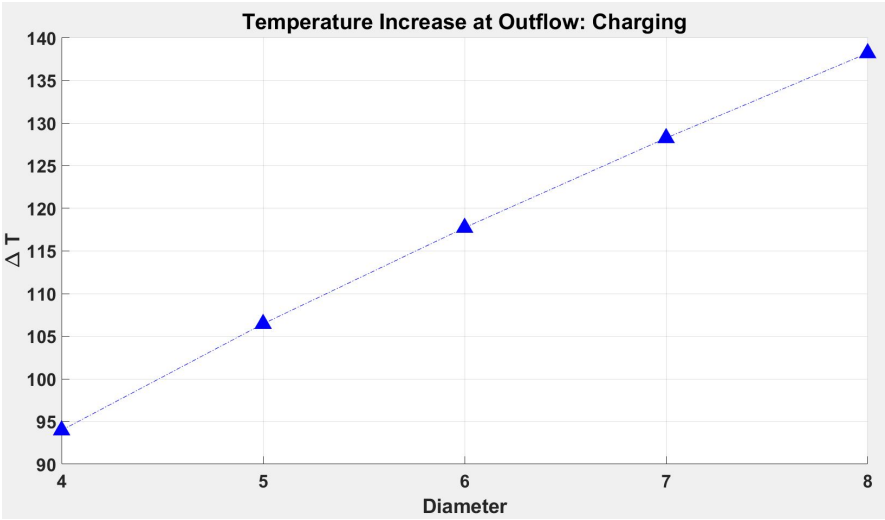

Figure 3.8: Temperature Rise during Charging

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
...	...	...
...	...	...
...	...	...
...	...	...

Chapter 5

Conclusions

5.1 Future Improvements

Several steps could be taken to improve the accuracy of the approximate solution we have obtained.

- An obvious solution is to use a second order accurate approximation for the advection term. A centered approximation could be used, with the added benefit that we get rid of the (dissipative) biased approximation.
- The fact that time approximation is first order may also be a bottleneck to the accuracy of our solution. A higher order time integration scheme might be a good idea here. The trapezoidal scheme or the RK4 time integration scheme could be alternatives to the forward Euler method used here, both of which include the left half of the complex plane in their stability region. However these improvements in accuracy will likely come at the cost of increased computational cost.
- In order to obtain a a better agreement with actual physical results, it would be a good idea to develop a 2D code, since for a cylindrical geometry, it appears that the temperature profile inside the storage device would be 2D in nature.

5.2 Take Away from the Project

As someone who is at ETH just to study computational fluid dynamics, this project was an eye opener, having previously worked only with commercial CFD codes during the course of my bachelors. The project (and the course in general) also showed me how vast the field is, and how much effort goes into things that I previously took for granted while working on commercial codes. However, writing my first CFD code has reinforced my wish to stick

to CFD and delve deeper into the subject. Actually writing the code helped me relate to many things from theory that I had earlier just heard of.

There's not a lot I would change if I redo the project, but a few organisational points to come up:

- Create a Git repository to keep track of all the changes that I was making to the code. Implementing the OVS can get very ugly since new loops pop up everywhere and it can get hard to keep track of the changes you made to the code.
- Write more succinct functions. In its present form there are a lot of functions that are too long and could be split leading to a code that can be read easily.

Appendix A

Solid Temperature Plots

Case 1: Diameter 4m

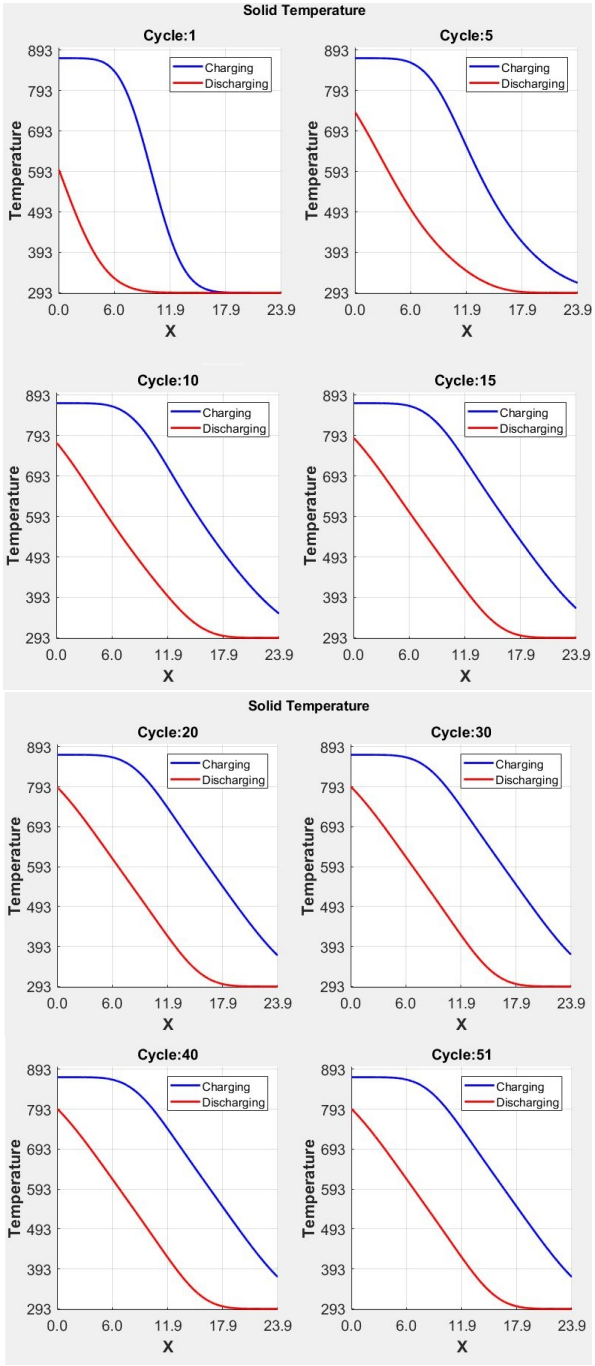

Figure A.1: Temperature Evolution

Case 2: Diameter 5m

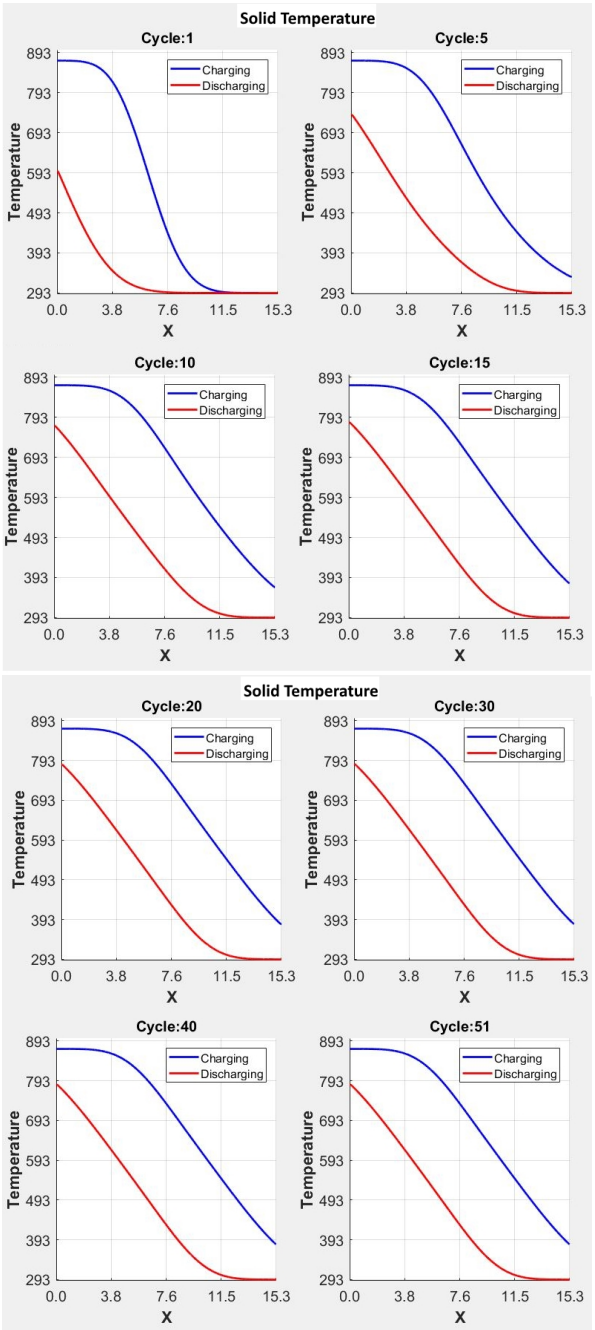

Figure A.2: Temperature Evolution

Case 3: Diameter 6m

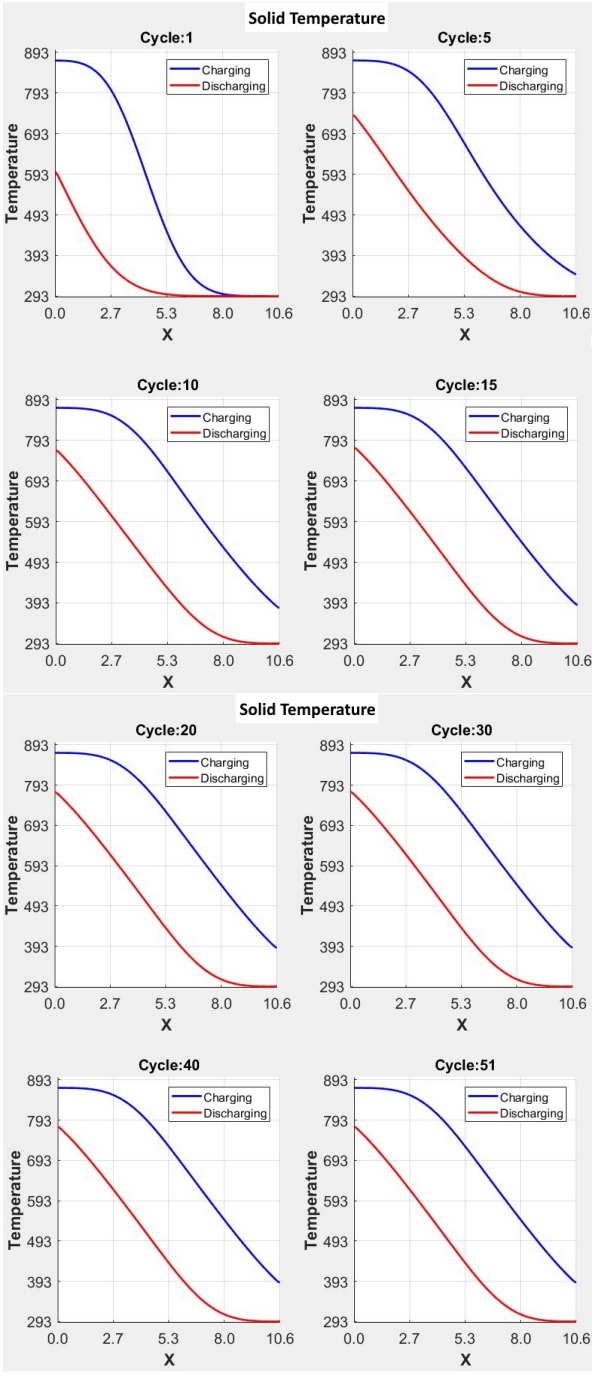

Figure A.3: Temperature Evolution

Case 4: Diameter 7m

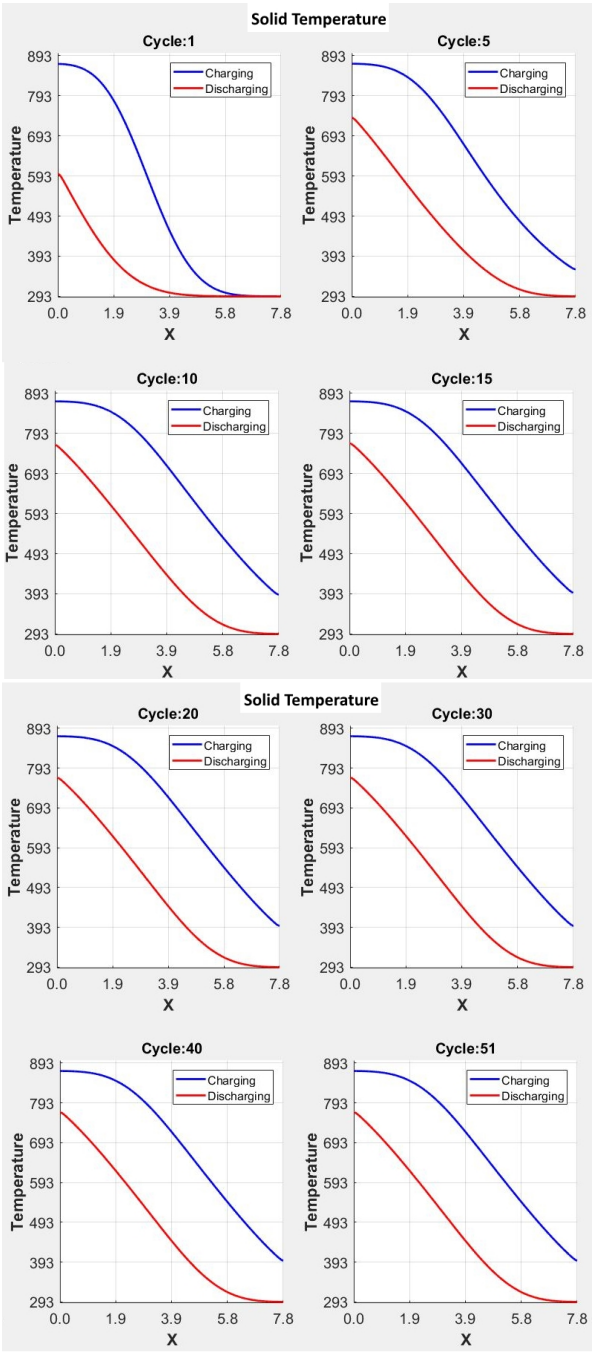

Figure A.4: Temperature Evolution

Case 5: Diameter 8m

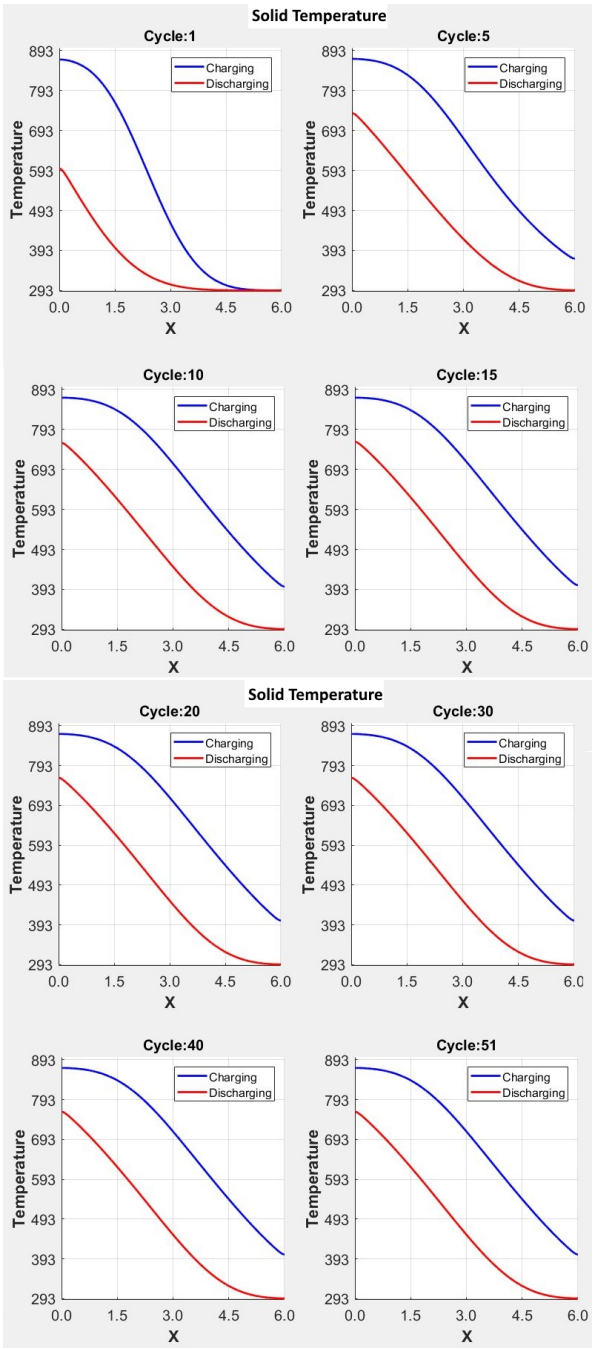

Figure A.5: Temperature Evolution

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.