ETH Zürich

Fundamentals of CFD Methods Project

Fall Semester 2017

Packed Bed Thermal Energy Storage Design

Lorenzo Giacomel

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):

Packed Bed Thermal Energy Storage Design -- Fundamentals of CFD Methods Project

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

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

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)**
Zürich, 13.12.2017	

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

Contents

1 Introduction

The aim of this project is to study the design of a Thermal Energy Storage (TES). Such a device allows to store the heat energy generated by a power plant in order to make it available in case of high demand or when the plant is not able to produce energy itself. A TES can be implied, for example, in solar power plants to make electricity available also when the sun is not shining.

1.1 Structure of a TES with Packed Bed

A TES with packed bed is an adiabatic cylinder partially filled with a porous solid (such as rocks or some kind of salt) which has two inflow/outflow pipes plugged at its bases. During a charging period, hot fluid inflows the cylinder through the upper pipe and cold fluid outflows through the bottom one. Due to this flow, the solid phase raises its temperature, storing the heat transported by the hot fluid. On the other hand, during a discharge period, cold fluid inflows the storage cooling down the solid phase, while hot fluid outflows through the top pipe. Now the hot fluid can be used to generate electricity through heat exchangers and turbines. The typical temperature distribution of a TES during the charge phase is depicted in figure 1.

1.2 Modelling Equations

The temperature of the two phases can be modelled by means of the Schumann equations,

$$\varepsilon \rho_f C_{p,f} \frac{\partial T_f}{\partial t} + \varepsilon \rho_f C_{p,f} u_{f,i} \frac{\partial T_f}{\partial x} = k_f \frac{\partial^2 T_f}{\partial x^2} + h_v (T_s - T_f), \tag{1}$$

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

In practice, we need to solve an advection-diffusion equation (1) coupled with a diffusion equation (2).

2 Finite Volumes Discretization

In order to numerically solve the equations, we carry out a Finite Volumes discretization in space, while we use first order finite differences for time derivatives. First we concentrate on the decoupled equations and after we include the coupling terms into our method. The coupling comes from the heat exchange between the two phases.

2.1 Numerical Solution of the Decoupled Equations

Following, we derive a Forward-Time-Backward-Space discretization for the charging periods ($u_{f,i} > 0$), still neglecting the terms related to heat exchange. Analogously, it is possible to derive a Forward-Time-Forward-Space discretization for the discharging periods ($u_{f,i} < 0$) and a Forward-Time-Centered-Space for the idle periods.

2.1.1 Semi-Discretization in Space

Starting from the equations

$$\frac{\partial T_f}{\partial t} + u_{f,i} \frac{\partial T_f}{\partial x} = \alpha_f \frac{\partial^2 T_f}{\partial x^2}, \tag{3}$$

$$\frac{\partial T_s}{\partial t} = \alpha_s \frac{\partial^2 T_s}{\partial x^2}, \tag{4}$$

with

$$\alpha_f = \frac{k_f}{\varepsilon \rho_f C_{p,f}} \quad \text{and} \quad \alpha_s = \frac{k_s}{(1-\varepsilon)\rho_s C_s},$$

we carry out a semi-discretization in space by means of Finite Volumes method (FTBS). We first deal with equation (3). We split the domain into N cells of size Δx as indicated in figure 2 so that we can

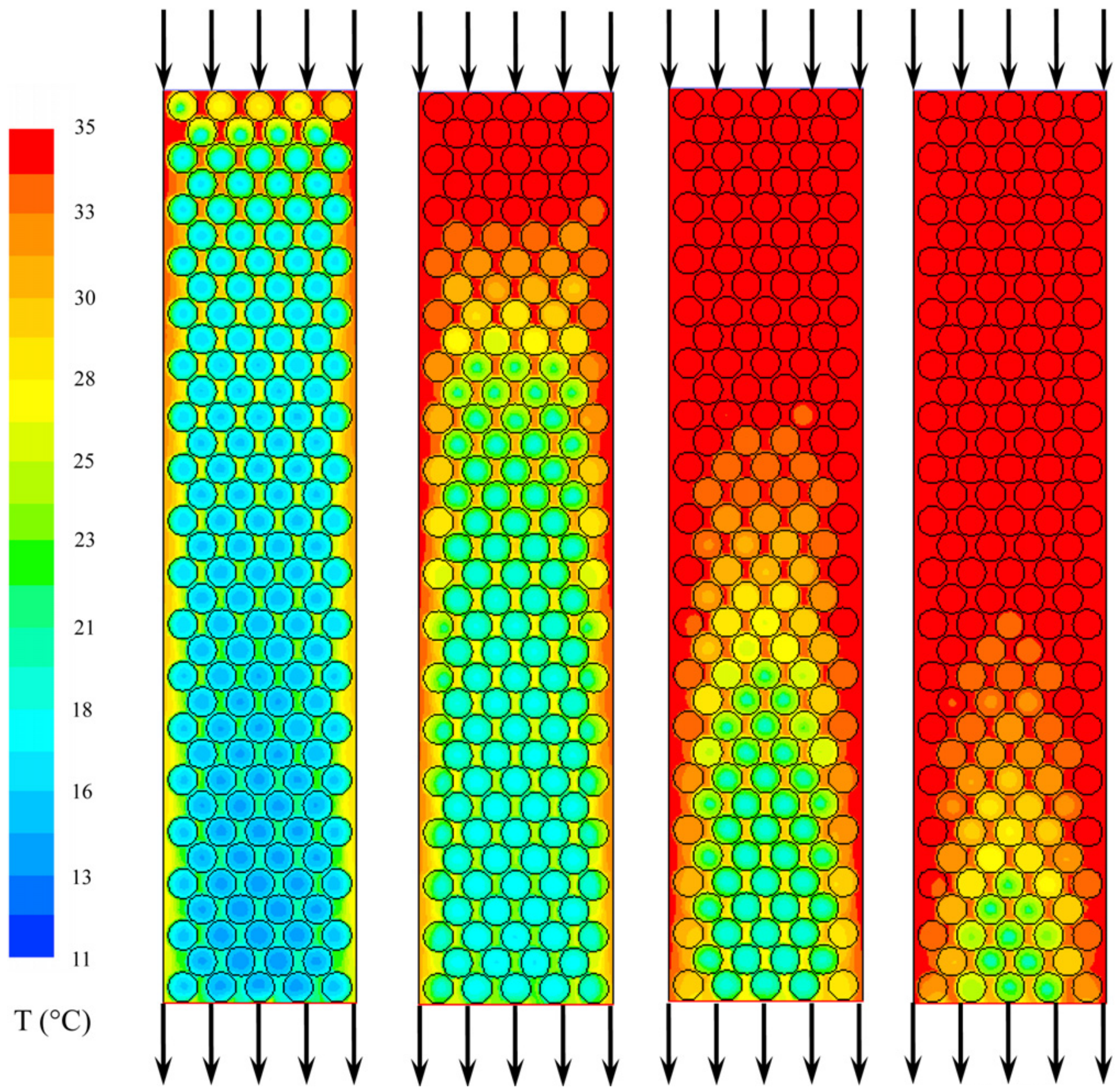


Fig. 1: Typical temperature distribution of a TES during the charge phase, the arrows denote the flow direction.
Image from: L. Xia, P. Zhang, R.Z. Wang, Numerical heat transfer analysis of the packed bed latent heat storage system based on an effective packed bed model, Energy 35 (2010), 2022-2032

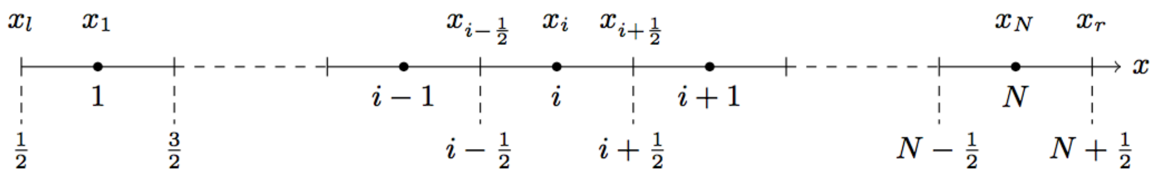


Fig. 2: Domain discretization into N cells.

derive a discrete equation for the interior cells ($1 < i < N$).
Then, we integrate equation (3) over the i-th cell obtaining

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial T_f}{\partial t} dx + u_{f,i} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial T_f}{\partial x} dx = \alpha_f \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial^2 T_f}{\partial x^2} dx.$$

Since the integration domain is fixed in time, it is possible to exchange the derivatives and the integrals which results in

$$\Delta x \frac{\partial \bar{T}_{f,i}}{\partial t} + u_{f,i} \left(T_{f,i+\frac{1}{2}} - T_{f,i-\frac{1}{2}} \right) = \alpha_f \left(\left.\frac{\partial T_f}{\partial x}\right|_{i+\frac{1}{2}} - \left.\frac{\partial T_f}{\partial x}\right|_{i-\frac{1}{2}} \right),$$

having defined the cell average

$$\bar{T}_{f,i} = \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} T_{f,i} dx.$$

Now we can express the temperature value at each the cell face as a function of the cell averages. This is possible through the first order approximation

$$T_{f,i+\frac{1}{2}} = \bar{T}_{f,i}.$$

We can also express the derivatives as function of the cell averages, but this time we use a centred scheme

$$\left.\frac{\partial T_f}{\partial x}\right|_{i+\frac{1}{2}} = \frac{1}{\Delta x} \left(\bar{T}_{f,i+1} - \bar{T}_{f,i} \right),$$

which has second order of accuracy. These approximations lead to the equation

$$\frac{\partial \bar{T}_{f,i}}{\partial t} = -\frac{u_{f,i}}{\Delta x} \left(\bar{T}_i - \bar{T}_{i-1} \right) + \frac{\alpha_f}{(\Delta x)^2} \left(\bar{T}_{f,i+1} - 2\bar{T}_{f,i} + \bar{T}_{f,i-1} \right).$$

Proceeding analogously with equation (2) we obtain

$$\frac{\partial \bar{T}_{s,i}}{\partial t} = \frac{\alpha_s}{(\Delta x)^2} \left(\bar{T}_{s,i+1} - 2\bar{T}_{s,i} + \bar{T}_{s,i-1} \right).$$

We can now take care of the boundary cells. As a consequence of the boundary conditions, at the leftmost cell ($i = 1$) we have:

- $T_{\frac{1}{2}} = T_{f,in}$, which is the given temperature at left boundary needed for advection term (Dirichlet boundary condition);
- $\left.\frac{\partial T_f}{\partial x}\right|_{\frac{1}{2}} = 0$, due to absence of conductive fluxes at the boundaries (Neumann boundary condition)

Taking into account these boundary conditions in the previous derivation, we obtain:

$$\frac{\partial \bar{T}_{f,1}}{\partial t} = -\frac{u_{f,i}}{\Delta x} \left(\bar{T}_1 - T_{left} \right) + \frac{\alpha_f}{(\Delta x)^2} \left(\bar{T}_{f,2} - \bar{T}_{f,1} \right),$$
$$\frac{\partial \bar{T}_{s,1}}{\partial t} = \frac{\alpha_s}{(\Delta x)^2} \left(\bar{T}_{s,2} - \bar{T}_{s,1} \right).$$

Furthermore, we have homogeneous Neumann boundary conditions at the rightmost cell $\left(\left.\frac{\partial T_f}{\partial x}\right|_{N+\frac{1}{2}} = 0 \text{ and } \left.\frac{\partial T_s}{\partial x}\right|_{N+\frac{1}{2}} = 0 \right)$, thus we can write the analogous equations for $i = N$:

$$\frac{\partial \bar{T}_{f,N}}{\partial t} = -\frac{u_{f,i}}{\Delta x} \left(\bar{T}_N - \bar{T}_{N-1} \right) + \frac{\alpha_f}{(\Delta x)^2} \left(-\bar{T}_{f,N} + \bar{T}_{f,N-1} \right),$$
$$\frac{\partial \bar{T}_{s,N}}{\partial t} = \frac{\alpha_s}{(\Delta x)^2} \left(-\bar{T}_{s,N} + \bar{T}_{s,N-1} \right).$$

2.1.2 Semi-Discretization in Time

We now need to solve the ODEs with respect to time. For this purpose, we employ a Backward Euler scheme with time step Δt, which for the ODE

$$\frac{\partial \phi}{\partial t} = f(x,t),$$

reads

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = f^n(x),$$

which is equivalent to

$$\phi^{n+1} = \phi^n + \Delta t f^n(x). \tag{5}$$

This is a first order approximation in time.
Applying the scheme (5) to the semi-discrete equations, we finally have:

$$\begin{aligned}
\bar{T}^{n+1}_{f,i} &= \bar{T}^n_{f,i} - \frac{u_{f,i}\Delta t}{\Delta x}\left(\bar{T}^n_i - \bar{T}^n_{i-1}\right) + \frac{\alpha_f \Delta t}{(\Delta x)^2}\left(\bar{T}^n_{f,i+1} - 2\bar{T}^n_{f,i} + \bar{T}^n_{f,i-1}\right) && i = 2,\dots,N-1,\\
\bar{T}^{n+1}_{f,1} &= \bar{T}^n_{f,1} - \frac{u_{f,i}\Delta t}{\Delta x}\left(\bar{T}^n_1 - T^n_{left}\right) + \frac{\alpha_f \Delta t}{(\Delta x)^2}\left(\bar{T}^n_{f,2} - \bar{T}^n_{f,1}\right),\\
\bar{T}^{n+1}_{f,N} &= \bar{T}^n_{f,N} - \frac{u_{f,i}\Delta t}{\Delta x}\left(\bar{T}^n_N - \bar{T}^n_{N-1}\right) + \frac{\alpha_f \Delta t}{(\Delta x)^2}\left(-\bar{T}^n_{f,N} + \bar{T}^n_{f,N-1}\right),\\
\bar{T}^{n+1}_{s,i} &= \bar{T}^n_{s,i} + \frac{\alpha_s \Delta t}{(\Delta x)^2}\left(\bar{T}^n_{s,i+1} - 2\bar{T}^n_{s,i} + \bar{T}^n_{s,i-1}\right) && i = 2,\dots,N-1,\\
\bar{T}^{n+1}_{s,1} &= \bar{T}^n_{s,1} + \frac{\alpha_s \Delta t}{(\Delta x)^2}\left(\bar{T}^n_{s,2} - \bar{T}^n_{s,1}\right),\\
\bar{T}^{n+1}_{s,N} &= \bar{T}^n_{s,N} + \frac{\alpha_s \Delta t}{(\Delta x)^2}\left(-\bar{T}^n_{s,N} + \bar{T}^n_{s,N-1}\right),
\end{aligned}$$

which are the fully discrete equations that we use to obtain the approximated temperatures of the liquid and solid phases.
The scheme for T_f has first order of accuracy in both space and time, while the scheme for Ts has second order of accuracy in space and first in time.

2.2 Numerical Solution of the Coupled Equations

In this section we show the numerical method used to solve equations (1) and (2) including the coupling terms.
Let us rewrite the equations as,

$$\frac{\partial T_f}{\partial t} + u_{f,i}\frac{\partial T_f}{\partial x} = \alpha_f \frac{\partial^2 T_f}{\partial x^2} - h_{v,f}(T_f - T_s), \tag{6}$$

$$\frac{\partial T_s}{\partial t} = \alpha_s \frac{\partial^2 T_s}{\partial x^2} - h_{v,s}(T_s - T_f), \tag{7}$$

with

$$\alpha_f = \frac{k_f}{\varepsilon \rho_f C_{p,f}}, \qquad h_{v,f} = \frac{h_v}{\varepsilon \rho_f C_{p,f}},$$
$$\alpha_s = \frac{k_s}{(1-\varepsilon)\rho_s C_s}, \qquad h_{v,s} = \frac{h_v}{(1-\varepsilon)\rho_s C_s}.$$

We could easily include the coupling term in the previous derivations with the backward Euler method but this would introduce further stability restrictions. Using forward Euler, instead, would be computationally very expensive, as it would require the inversion of a $2N$x$2N$ system at each time step. In order to keep the stability region untouched without introducing expensive computations, we use the *point-implicit method*, in which the derivatives are evaluated at time n while the coupling term is evaluated at time $n+1$:

$$\begin{aligned}
\bar{T}_{f,i}^{n+1} &= \bar{T}_{f,i}^{n} - \frac{u_{f,i}\Delta t}{\Delta x}\left(\bar{T}_{i}^{n} - \bar{T}_{i-1}^{n}\right) + \frac{\alpha_f \Delta t}{(\Delta x)^2}\left(\bar{T}_{f,i+1}^{n} - 2\bar{T}_{f,i}^{n} + \bar{T}_{f,i-1}^{n}\right) - \Delta t h_{v,f}(\bar{T}_{f,i}^{n+1} - \bar{T}_{s,i}^{n+1}) && i = 2,\dots,N-1,\\
\bar{T}_{f,1}^{n+1} &= \bar{T}_{f,1}^{n} - \frac{u_{f,i}\Delta t}{\Delta x}\left(\bar{T}_{1}^{n} - T_{left}^{n}\right) + \frac{\alpha_f \Delta t}{(\Delta x)^2}\left(\bar{T}_{f,2}^{n} - \bar{T}_{f,1}^{n}\right) - \Delta t h_{v,f}(\bar{T}_{f,1}^{n+1} - \bar{T}_{s,1}^{n+1}),\\
\bar{T}_{f,N}^{n+1} &= \bar{T}_{f,N}^{n} - \frac{u_{f,i}\Delta t}{\Delta x}\left(\bar{T}_{N}^{n+1} - \bar{T}_{N-1}^{n+1}\right) + \frac{\alpha_f \Delta t}{(\Delta x)^2}\left(-\bar{T}_{f,N}^{n} + \bar{T}_{f,N-1}^{n}\right) - \Delta t h_{v,f}(\bar{T}_{f,N}^{n+1} - \bar{T}_{s,N}^{n+1}),\\
\bar{T}_{s,i}^{n+1} &= \bar{T}_{s,i}^{n} + \frac{\alpha_s \Delta t}{(\Delta x)^2}\left(\bar{T}_{s,i+1}^{n} - 2\bar{T}_{s,i}^{n} + \bar{T}_{s,i-1}^{n}\right) - \Delta t h_{v,s}(\bar{T}_{s,i}^{n+1} - \bar{T}_{f,i}^{n+1}) && i = 2,\dots,N-1,\\
\bar{T}_{s,1}^{n+1} &= \bar{T}_{s,1}^{n} + \frac{\alpha_s \Delta t}{(\Delta x)^2}\left(\bar{T}_{s,2}^{n} - \bar{T}_{s,1}^{n}\right) - \Delta t h_{v,s}(\bar{T}_{s,1}^{n+1} - \bar{T}_{f,1}^{n+1}),\\
\bar{T}_{s,N}^{n+1} &= \bar{T}_{s,N}^{n} + \frac{\alpha_s \Delta t}{(\Delta x)^2}\left(-\bar{T}_{s,N}^{n} + \bar{T}_{s,N-1}^{n}\right) - \Delta t h_{v,s}(\bar{T}_{s,N}^{n+1} - \bar{T}_{f,N}^{n+1}).
\end{aligned}$$

which can be converted into the following two steps method:

$$\begin{aligned}
\bar{T}_{f,i}^{*} &= \bar{T}_{f,i}^{n} - \frac{u_{f,i}\Delta t}{\Delta x}\left(\bar{T}_{i}^{n} - \bar{T}_{i-1}^{n}\right) + \frac{\alpha_f \Delta t}{(\Delta x)^2}\left(\bar{T}_{f,i+1}^{n} - 2\bar{T}_{f,i}^{n} + \bar{T}_{f,i-1}^{n}\right) && i = 2,\dots,N-1,\\
\bar{T}_{f,1}^{*} &= \bar{T}_{f,1}^{n} - \frac{u_{f,i}\Delta t}{\Delta x}\left(\bar{T}_{1}^{n} - T_{left}^{n}\right) + \frac{\alpha_f \Delta t}{(\Delta x)^2}\left(\bar{T}_{f,2}^{n} - \bar{T}_{f,1}^{n}\right)\\
\bar{T}_{f,N}^{*} &= \bar{T}_{f,N}^{n} - \frac{u_{f,i}\Delta t}{\Delta x}\left(\bar{T}_{N}^{n} - \bar{T}_{N-1}^{n}\right) + \frac{\alpha_f \Delta t}{(\Delta x)^2}\left(-\bar{T}_{f,N}^{n} + \bar{T}_{f,N-1}^{n}\right),\\
\bar{T}_{s,i}^{*} &= \bar{T}_{s,i}^{n} + \frac{\alpha_s \Delta t}{(\Delta x)^2}\left(\bar{T}_{s,i+1}^{n} - 2\bar{T}_{s,i}^{n} + \bar{T}_{s,i-1}^{n}\right) && i = 2,\dots,N-1\\
\bar{T}_{s,1}^{*} &= \bar{T}_{s,1}^{n} + \frac{\alpha_s \Delta t}{(\Delta x)^2}\left(\bar{T}_{s,2}^{n} - \bar{T}_{s,1}^{n}\right),\\
\bar{T}_{s,N}^{*} &= \bar{T}_{s,N}^{n} + \frac{\alpha_s \Delta t}{(\Delta x)^2}\left(-\bar{T}_{s,N}^{n} + \bar{T}_{s,N-1}^{n}\right),
\end{aligned}$$

$$\begin{bmatrix} 1 + h_{v,f}\Delta t & -h_{v,f} \\ -h_{v,s} & 1 + h_{v,s}\Delta t \end{bmatrix} \begin{bmatrix} \bar{T}_{f,i}^{n+1} \\ \bar{T}_{s,i}^{n+1} \end{bmatrix} = \begin{bmatrix} \bar{T}_{f,i}^{*} \\ \bar{T}_{s,i}^{*} \end{bmatrix} \qquad i = 1,\dots,N.$$

Notice that the 2x2 systems can be solved analytically and overall in linear time, therefore no more error or complexity are introduced.

2.3 The Reconstruction

Starting from the cell averages we reconstruct the face values through the following second order scheme:

$$\begin{aligned}
T_{f,i+\frac{1}{2}} &= \frac{1}{2}\left(T_{f,i-1} + T_{f,i}\right) && i = 1,\dots,N-1,\\
T_{s,i+\frac{1}{2}} &= \frac{1}{2}\left(T_{s,i-1} + T_{s,i}\right) && i = 1,\dots,N-1,\\
T_{f,\frac{1}{2}} &= T_{f,in},\\
T_{s,\frac{1}{2}} &= T_{s,1},\\
T_{f,N+\frac{1}{2}} &= T_{f,N},\\
T_{s,N+\frac{1}{2}} &= T_{s,N},
\end{aligned}$$

3 C++ Implementation

We implemented the algorithm in a c++, which consists of:

- the file main.cpp, which contains the main for the simulations;
- the file mainOVS.cpp, which contains the main for the Order Verification Study (OVS);
- the file init.hpp, which contains the routine which initializes the various parameters needed for the simulations;
- the file create_states.hpp, containing the code to create the state vector, which contains the state of the TES at every time step (charging/discharging/idle);
- the file step.hpp, which contains all the functions needed to perform one step of the previously derived algorithm;
- the file reconstruct.hpp, containing the routine used to carry out the reconstruction of the solution;
- the file errors.hpp, which contains the functions to compute the normalized errors in L^∞ and L^2 norms and to compute the L^∞ norm of the difference between two vectors;
- the file write.hpp, which embeds two auxiliary functions to write the results into an external file;
- the file postprocessing.hpp, which provides the functions used to compute the exergy efficiency and the capacity factor of the storage;
- the MATLAB script anim_thermoclines.m allows to visualize the evolution in time of the thermoclines of fluid and solid phases;
- the MATLAB script anim_TES.m displays a nice animation of the time evolution of T_f and T_s in the storage;

3.1 The Simulations Main

In main.cpp, at first, we determine the state for each time step through the function create_states, then we check if the chosen parameters ensure numerical stability and, after that, we start looping over the time steps. For each of them we compute the appropriate advection speed for the given state and we perform a step of the previously derived algorithm by means of the function coupled_solver. Moreover, we periodically carry out the reconstruction of the solutions and we write them into an external file. These files can be used for visualization purposes (see section 3.4).
In the end, we use the functions implemented in the postprocessing.hpp file in order to compute the exergy efficiency, the capacity factor and the temperature increase at the outflow at the end of a charging period. The results are shown as output in the terminal together with the parameters needed to set up the MATLAB simulations.

3.2 The OVS Main

We used mainOVS.cpp to carry out the Order Verification Study. We focused on the charging periods but the code can easily be adapted to verify the method used for the discharging periods. Since we use the Method of Manufactured Solutions (MMS) with a steady state solution we need to keep running the method until the time-independent solution is reached. For this purpose, at every time step we compute the solution change with respect to the previous one and we stop when the change is very small. These computations are performed in a while loop. After the loop we compute the errors with respect to the manufactured solution in L^∞ and in L^2 norm and we display them as outputs in the terminal.

3.3 How to Compile, Execute and Choose the Parameters

The code can be easily compiled and run using the terminal. After having opened a terminal window it is necessary to move to the Code folder and type `c++ main.cpp -o main`. An executable called main will be created, it can be executed typing ./main followed either by the flag terminal or file, which determines whether the parameters should be asked in the terminal or if they should be read from the file params.dat.
The file params.dat must contain one parameter in each line in the following order:

- Volume of the storage, V, m^3;

- Diameter of the storage D, m;
- Temperature of inflowing fluid during charge periods T_c, K;
- Temperature of inflowing fluid during discharge periods T_d, K;
- Duration of the charge periods in seconds
- Duration of the idle periods between charging and discharging in seconds;
- Duration of the discharge periods in seconds
- Duration of the idle periods between discharging and charging in seconds;
- Number of charging-discharging cycles
- Number of time steps per cycle
- Conductivity of the fluid phase k_f, W/m K;
- Conductivity of the solid phase k_s, W/m K;
- Porosity ε;
- Filler bed particle diameter d_s, m;
- Density of the solid phase ρ_f, kg/m^3;
- Density of the fluid phase ρ_s, kg/m^3;
- Specific heat of the solid phase C_s, J/kg K;
- Specific heat at constant pressure of the fluid phase $C_{p,f}$, J/kg K;
- Viscosity of the fluid phase μ_f, kg/m s;
- Mass flow rate $\dot{m}_f$, kg/s;

3.4 MATLAB Animations

We provide two MATLAB scripts which can be used to visualize the simulations of TES: anim_thermoclines.m and anim_TES.m.
The former shows the evolution of the thermoclines while the TES is in action. The latter, instead, gives a simple 2D visualization of the temperature distribution of the two phases in the storage, assuming that the temperature is uniform along the x axis. Obviously, we do not claim to have a 2D solution but we simply try to give a visualization of the solution which intuitively shows the temperature evolution inside the storage during each period.
The temperatures folder already embeds the results from a simulation which can be simply visualized running the scripts.

4 Order Verification Study

In this section, we present the results of the Order Verification Study (OVS) with the Method of Manufactured Solutions (MMS) for the spatial discretizations, showing that the actual order accuracy in space of the code is equal to the theoretical one. First, we analyse the decoupled equations, afterwards we include the coupling term into the analysis. Moreover, we compare the theoretical solution (ignoring the conduction) with the numerical one. In theory the method used for the diffusion term has second order of accuracy while the method used for the advection term has first order.
The Manufactured solutions that we used are $T_f = \cos(\frac{3\pi x}{L})$ and $T_s = \cos(\frac{2\pi x}{L})$.
For the error computation we have used both the L^2 and the L^∞ norms.

4.1 OVS for the Decoupled Equations

For the MMS we need to add the following source terms to the discrete equations:

$$\begin{aligned} s_{f,i} &= \alpha_f \omega_f \left[sin(\omega_f x)\right]_{i-\frac{1}{2}}^{i+\frac{1}{2}} + u_{f,i} \left[cos(\omega_f x)\right]_{i-\frac{1}{2}}^{i+\frac{1}{2}} , \\ s_{s,i} &= \alpha_s \omega_s \left[sin(\omega_s x)\right]_{i-\frac{1}{2}}^{i+\frac{1}{2}} , \end{aligned}$$

with $\omega_f = \frac{3\pi}{L}$ and $\omega_s = \frac{2\pi}{L}$.

Fluid phase
We carried out the OVS for different Peclet numbers ($\text{Pe} = \frac{u_{f,i} L}{\alpha_f}$). Figure 3 shows that the method used has first order of accuracy when the problem is advection dominant (i.e. $\text{Pe} \gg 1$), while the plots in figure figure 4 show second order of accuracy when the problem is diffusion dominant (i.e. $\text{Pe} \gg 1$). We also notice that, when $\text{Pe} = 1$, no exact order of accuracy is observed (figure 5).
Solid phase
As expected the results in figure 6 show that the discretization carried out for the diffusion term has second order of accuracy.

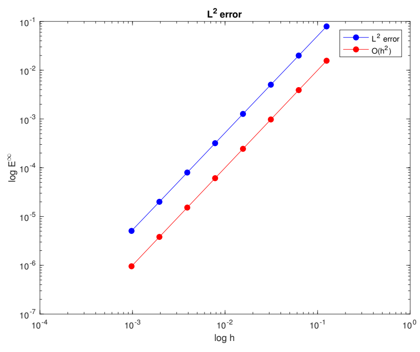
4.2 OVS for the Coupled Equations

At this point, we include also the coupling term into the equations. We need to modify the source terms to be:

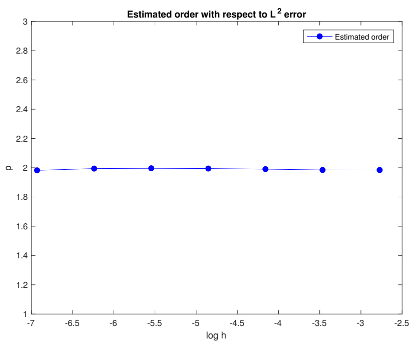
$$\begin{aligned} s_{f,i} &= \left(\alpha_f \omega_f + \frac{h_{v,f}}{\omega_f}\right) \left[sin(\omega_f x)\right]_{i-\frac{1}{2}}^{i+\frac{1}{2}} + u_{f,i} \left[cos(\omega_f x)\right]_{i-\frac{1}{2}}^{i+\frac{1}{2}} - \frac{h_{v,s}}{\omega_s} \left[sin(\omega_s x)\right]_{i-\frac{1}{2}}^{i+\frac{1}{2}} , \\ s_{s,i} &= \left(\alpha_s \omega_s + \frac{h_{v,s}}{\omega_s}\right) \left[sin(\omega_s x)\right]_{i-\frac{1}{2}}^{i+\frac{1}{2}} - \frac{h_{v,f}}{\omega_f} \left[sin(\omega_f x)\right]_{i-\frac{1}{2}}^{i+\frac{1}{2}} , \end{aligned}$$

with $\omega_f = \frac{3\pi}{L}$ and $\omega_s = \frac{2\pi}{L}$.

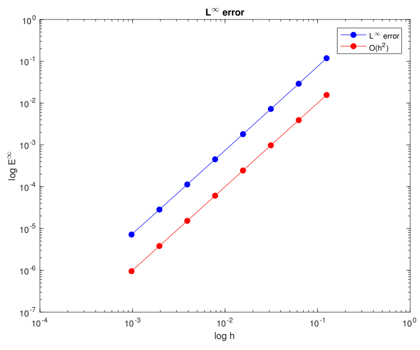
We now analyse the accuracy of T_f and T_s on the same plots because, given the coupling, the order of accuracy is the same for both of them. In figure 7 we observe that for both of the phases we have order one. The reason is that the errors sum up and the leading term becomes of first order also for the solid phase.



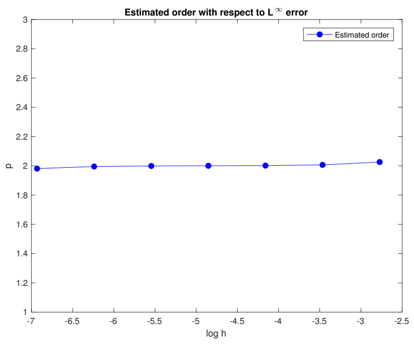
(a) L^2 error



(b) Order of accuracy with respect to L^2 error

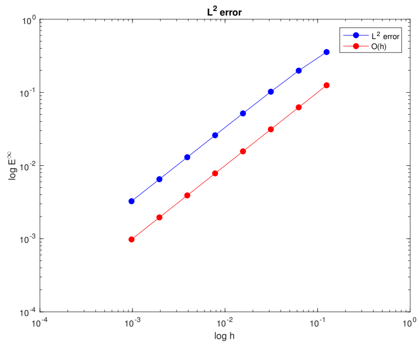


(c) L^∞ error

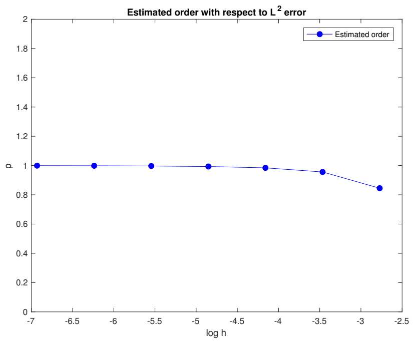


(d) Order of accuracy with respect to L^∞ error

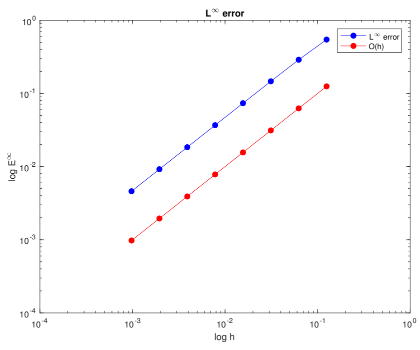
Fig. 3: Order of accuracy for fluid phase with $\mathrm{Pe} = 10^6$



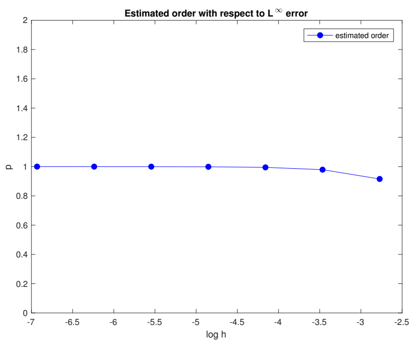
(a) L^2 error



(b) Order of accuracy with respect to L^2 error

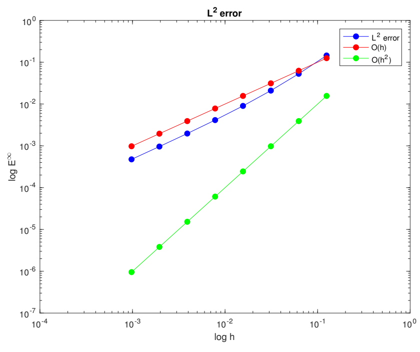


(c) L^∞ error

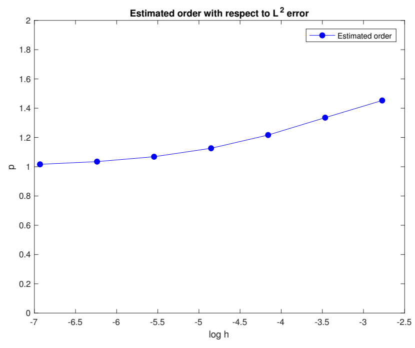


(d) Order of accuracy with respect to L^∞ error

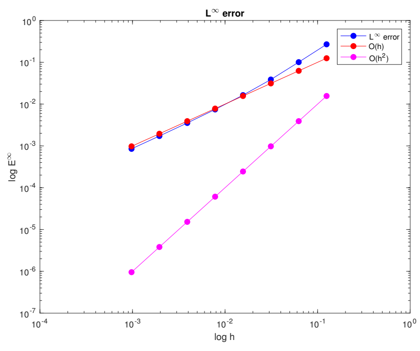
Fig. 4: Order of accuracy for fluid phase with $\mathrm{Pe} = 10^{-6}$



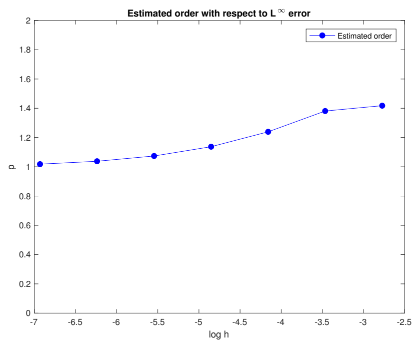
(a) L^2 error



(b) Order of accuracy with respect to L^2 error

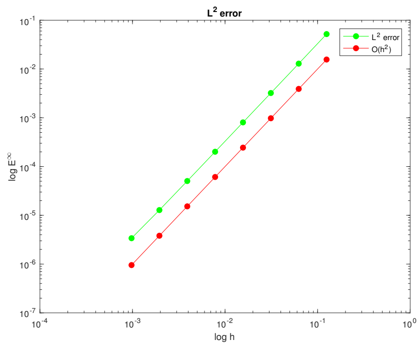


(c) L^∞ error

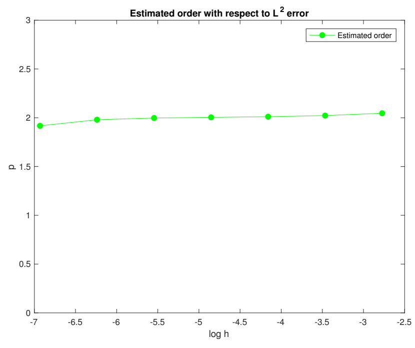


(d) Order of accuracy with respect to L^∞ error

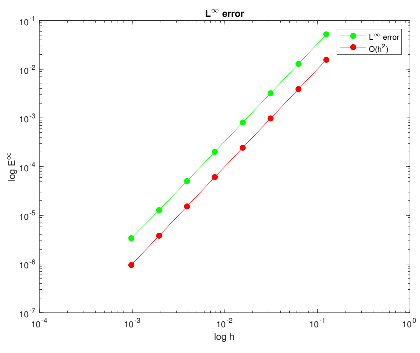
Fig. 5: Order of accuracy for fluid phase with Pe $= 1$



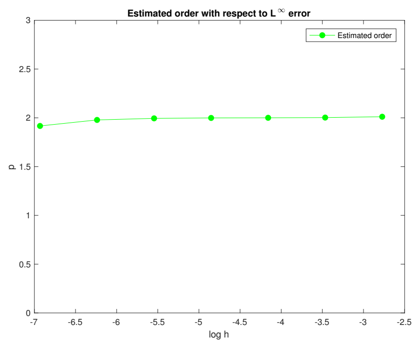
(a) L^2 error



(b) Order of accuracy with respect to L^2 error

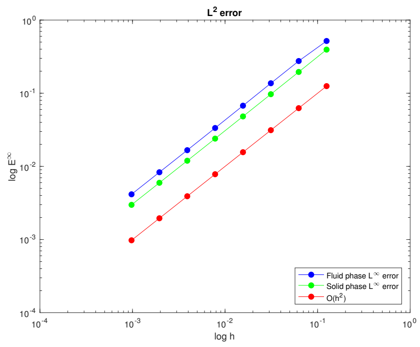


(c) L^∞ error

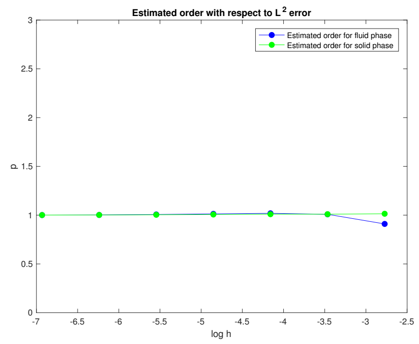


(d) Order of accuracy with respect to L^∞ error

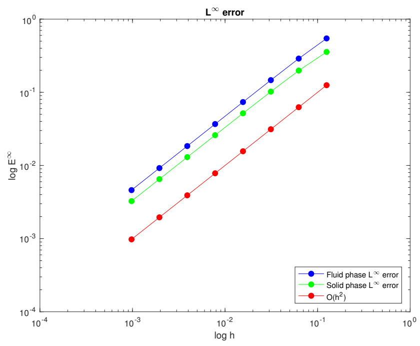
Fig. 6: Order of accuracy for solid phase



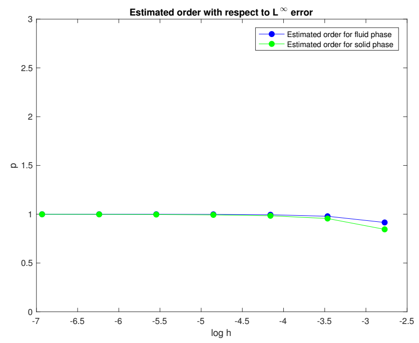
(a) L^2 error



(b) Order of accuracy with respect to L^2 error



(c) L^∞ error



(d) Order of accuracy with respect to L^∞ error

Fig. 7: Order of accuracy for coupled phases

4.3 Comparison with the Exact Solution

By neglecting the diffusion terms, it is possible to write in closed form the exact solution for our equations. If we define the non-dimensional coordinates

$$\xi(x) = \frac{h_v x}{\varepsilon \rho_f C_{p,f} u_{f,i}},$$

$$\eta(x,t) = \frac{h_v}{(1-\varepsilon)\rho_s C_s}\left(t - \frac{x}{u_{f,i}}\right)$$

and the non-dimensional temperature

$$\bar{T} = \frac{T - T_{s,0}}{T_{f,in} - T_{S,0}},$$

we obtain

$$\frac{\partial \bar{T}_f}{\xi} = \bar{T}_s - \bar{T}_f,$$

$$\frac{\partial \bar{T}_s}{\eta} = \bar{T}_f - \bar{T}_s.$$

The solution of these equations is

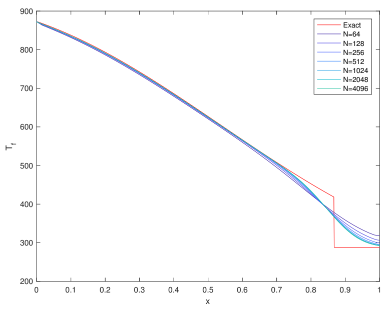
$$\bar{T}_f(\xi,\eta) = 1 - e^{-\eta}\int_0^\xi e{-}\xi' J_0(2i\sqrt{\xi'\eta})d\xi',$$

$$\bar{T}_f(\xi,\eta) = e^{-\xi}\int_0^\eta e{-}\eta' J_0(2i\sqrt{\xi\eta'})d\eta'.$$

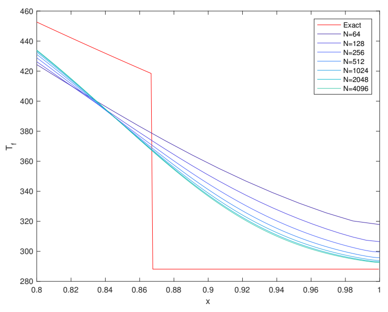
Where J_0 is the zeroth-order Bessel function of the first kind.
We compare our numerical solution with the exact one using the following parameters:

- $D = 1$ m;
- $H = 1$ m (the height of the storage);
- $\varepsilon = 0.4$;
- $d_s = 0.03$ m;
- $\rho_s = 2600$ kg/m^3;
- $\rho_f = 1835.6$ kg/m^3;
- $C_s = 900.0$ J/kg K;
- $C_{p,f} = 1511.8$ J/kg K;
- $k_s = 2.0$ W/m K;
- $k_f = 0.52$ W/m K;
- $\mu_f = 2.63$ kg/m s;
- $\dot{m}_f = 0.1$ kg/s;
- $T_{s,0} = 288.15$ K;
- $T_{f,0} = 873$ K;
- $t = 5000$ s;

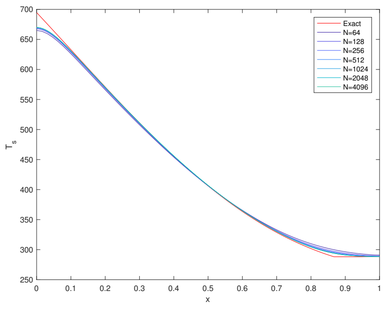
The plots in figure 8 show that the exact solution has a discontinuity of the first kind. The numerical solution doesn't approximate really well the analytical one in the region close to the step. More specifically, in the numerical solution, instead of a step, we observe a smooth temperature gradient. There are two reasons for this behaviour: the first is that in our numerical solution we took $k_f \neq 0$ and $k_s \neq 0$, the second is that we used the upwind scheme for the first derivative, which introduces some numerical dissipation. We also tried to run the same simulations taking $k_f = 0$ and $k_s = 0$, we report the results in figure 9. We can observe that the smoothing effect has diminished but it is still present, due to the numerical dissipation.



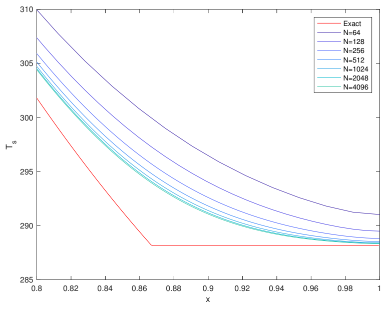
(a) T_f at $t = 5000s$ with diffusion term



(b) T_f at $t = 5000s$ with diffusion term restricted to $x \in (0.8, 1)$

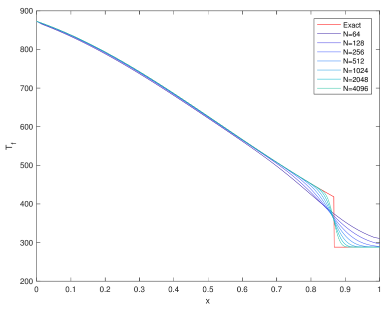


(c) T_s at $t = 5000s$ with diffusion term

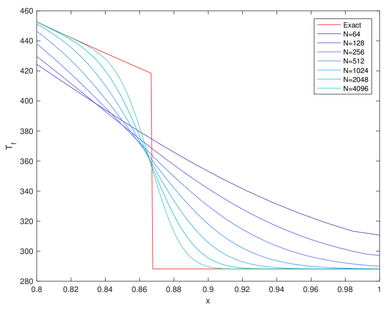


(d) T_s at $t = 5000s$ with diffusion term restricted to $x \in (0.8, 1)$

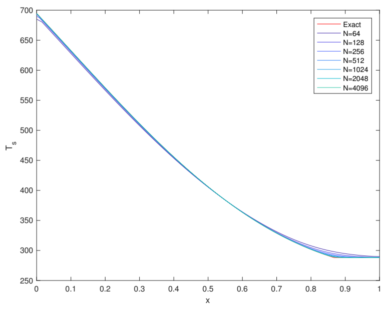
Fig. 8: Comparison between exact and numerical solution with diffusion term



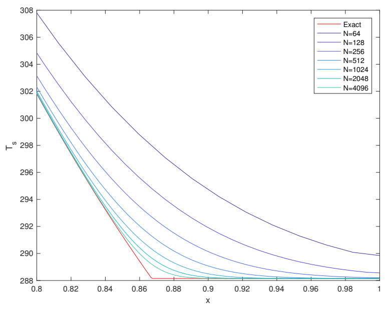
(a) T_f at $t = 5000s$ without diffusion term



(b) T_f at $t = 5000s$ with diffusion term restricted to $x \in (0.8, 1)$



(c) T_s at $t = 5000s$ without diffusion term



(d) T_s at $t = 5000s$ with diffusion term restricted to $x \in (0.8, 1)$

Fig. 9: Comparison between exact and numerical solution without diffusion term

5 TES Design

In this section we investigate which are the best dimensions for a TES. More specifically, we fix the volume to be 400 m^3 and we run our simulations with diameter values $\{4, 5, 6, 7, 8\}$ m. We analyse the change of three quantities which give a an idea of how well the TES has performed. Those three quantities are:

- The temperature increase at the outflow at the end of a charging period;
- The exergy efficiency, defined as:
$$\eta = \frac{\dot{\Xi}_{d,out} - \dot{\Xi}_{d,in}}{\dot{\Xi}_{c,in} - \dot{\Xi}_{c,out}},$$
with
$$\Xi = \int \dot{m}_f Cp, f \left[T_f - T_0 - T - 0 \ln(\frac{T_f}{T_0}) \right] dt$$
and $T_0 = 288.15$;
- The capacity factor, which is defined as the ratio of the thermal energy actually stored to the maximum thermal energy that can be stored, having
$$Q(t) = \frac{\pi}{4} D^2 \left[\varepsilon \rho_f C_{p,f} \int (T_f(z,t) - T_d) dz + (1-\varepsilon) \rho_s C_s \int (T_s(z,t) - T_d) dz \right]$$
and
$$Q_{max} = [\varepsilon \rho_f C_{p,f} + (1-\varepsilon)\rho_s C_s] \frac{\pi}{4} D^2 H (T_c - T_d).$$

The values of these quantities are computed after 80 charge-idle-discharge-idle cycles, so that the TES is in steady state regime. Each period of the cycle lasts 6 hours, thus in total each cycle lasts 24 hours. For our simulations we used the following parameters:

- $V = 300$ m^3;
- $D = 4, 5, 6, 7, 8$ m
- $\varepsilon = 0.4$;
- $d_s = 0.03$ m;
- $\rho_s = 2600$ kg/m^3;
- $\rho_f = 1835.6$ kg/m^3;
- $C_s = 900.0$ J/kg K;
- $C_{p,f} = 1511.8$ J/kg K;
- $k_s = 2.0$ W/m K;
- $k_f = 0.52$ W/m K;
- $\mu_f = 2.63$ kg/m s;
- $\dot{m}_f = 10$ kg/s;
- $T_c = 293$ K;
- $T_d = 873$ K;
- $dt = 4$ s;
- $dx = \frac{H}{2000}$;

The obtained results are summarized in the following table.

Diameter	Outflow temperature increase	Exergy efficiency	Capacity factor
4 m	93.5 K	0.998	0.485
5 m	105.5 K	0.997	0.489
6 m	116.5 K	0.993	0.493
7 m	126.5 K	0.989	0.497
8 m	135.9 K	0.983	0.501

When increasing the storage diameter (and consequently lowering the height), we notice the following trends: the temperature increase at the outflow raises, the capacity factor decreases and the exergy efficiency increases. At first glance, it seems reasonable to choose 6 m as the best diameters because it is a good trade off between capacity and exergy efficiency. Nevertheless, we need to be aware of the fluid temperature increase at the outflow. This quantity needs to be kept controlled because, if the fluid outflowing the TES has an excessive temperature, it may put into danger other parts of the plant. For this reason it may be necessary to give up some performance for the sake of security of the whole plant.

6 Conclusion

Overall we obtained a good simulation tool and an intuitive graphic visualization of the TES. However, we would like to conclude with some remarks about our work.

6.1 About the Model and the Methods

Although, thanks to its simplicity, it has been simple to deal with our mathematical model, we have to underline that some advanced physical issues haven't been taken into account. One for example is that, depending on the packed bed disposition, the fluid phase advection speed may vary in different regions of the storage. If we took this effect into account, we would not have an uniform temperature distribution along the y axis (the one parallel to the basis of the storage). The fact that the advection speed is not uniform is even more evident if we consider that the inflow and outflow of the fluid will happen through pipes whose cross sections are much smaller than the basis of the storage. If we were to take into account the non-uniform advection speed we would then need a two-dimensional model which would require a lot more computational effort to be solved. On the other hand, it is also possible to point out that, if we use an average advection speed, our one-dimensional model is still valid.

Our simulations took long time as we needed to let the TES work for a sufficient number of cycles so that enters the steady state regime. For this reason, the computations are so long that we could not afford to shrink too much dx and dt (also due to stability restrictions). Nevertheless, a very good feature of our time-explicit method is that it can be easily parallelized so that it can be run on multi-core or multi-node architectures (using for example OpenMP and/or MPI). One easy extension, would be to parallelize the big for loops in the stepTfCharge, stepTfDischarge and stepTs functions. A faster code would allow to keep the computational time reasonable even on finer grids.

6.2 About the Project

There's a big lesson that we learnt while developing the code. At first we wrote the a main file containing also all the needed functions and we used it both for testing and simulating. This structure has turned out to be very inconvenient when we needed to test our functions. A better practice both for code testing and reuse is to write the functions in external header files so that they can be more easily invoked from external codes. If we were to start a similar project from scratch we would probably group the functions into header files from the very beginning, which would make our life much easier.