



Scientific Computing assignment

Porous medium heated from below

Authors:

Volodymyr Yelnyk (0472102)
Ferran de Andrés (0472129)
Ferhat Sindy (0472118)

Submitted to:

Dr. Jürgen Fuhrmann

April 18, 2022

Abstract

We implement a finite volume method solver applied to a simple porous medium heated from below modelled through a system of Darcy's equation for flow in a porous medium saturated with water and a convective heat equation, including the effect of heat expansion leading to a decrease of the water density and the effect of convective heat transport.

The problem is approached using an own-code developed finite volume solver in Julia¹. The discretization strategies, both in terms of space and time, are discussed.

Finally, some comments are developed on how to improve the performance of the code, covering alternative timestepping methods and high-dimensional linear solvers.

Code available at [GitHub](#).

¹[The Julia programming language](#)

1 Introduction

One of the main academic interests in the study of convection in a porous medium is that the system is non-linear solely because of the advection of heat. Thus, the model is one of the simplest non-linear elliptic systems. Reviews of the firsts advances in this domain were presented by Cheng [1], Nield [6], Trevisan [10] and McKibbin [5]

In terms of more contemporary applications, the fluid flow and heat transfer in a porous media have widely attracted interest in many fields. Geophysical applications, heating and cooling in buildings, insulation materials and solar energy are examples of fluid flow in porous mediums, as introduced by Nojoomizadeh [7] and Meghdadi Isfahani [4].

In this report, the chosen approach is the so-called Boussinesq approximation, a widely used and powerful simplification in studies of density-dependent flows. The approximation in its strictest form includes the following assumptions (Gartling and Hickox [2]; Gray and Giorgini [3]):

- Density variations induced by variations in temperature and/or solute concentration are neglected except for the gravity term.
- All other material properties are assumed to be constant.
- Viscous dissipation is assumed negligible.
- The equation of state is linearized

The physically governing equations for this problem are the ones modelling the heat and mass transport in a porous medium: mass and energy balances and the Darcy's law. Additionally, an state equation is needed to model the dependence of the density on the temperature.

The mass balance equation is

$$n \frac{\partial}{\partial t} \rho_{\text{ref}} + \nabla \vec{q} = 0, \quad (1.1)$$

where n is the porosity factor, ρ_{ref} is the reference density and \vec{q} is the water mass flux. On the other hand, the energy balance is modelled as

$$nc_f \frac{\partial}{\partial t} T + (1 - n)\rho_s c_s \frac{\partial}{\partial t} T - \nabla \cdot (\lambda \nabla T - c_f \vec{q} T) = 0, \quad (1.2)$$

where c_f and c_s are heat capacities, T is the temperature and λ is the heat conductivity. It should be noted that we assume that the medium is completely porous, so $n = 1$ and thus the second term is neglected.

Finally, the Darcy's law describing the flow through the porous medium is defined as

$$\vec{q} + \rho_{\text{ref}} k (\nabla p - \rho \vec{g}) = 0, \quad (1.3)$$

where p denotes the fluid pressure, k the intrinsic permeability, and \vec{g} the acceleration of gravity.

It is relevant to point out that the handling of this system of equation during the development of the project is dimensionless.

[Figure 1.1](#) shows a representation of the proposed problem. As it can be seen, there are different boundary condition in each of the limits of the considered domain.

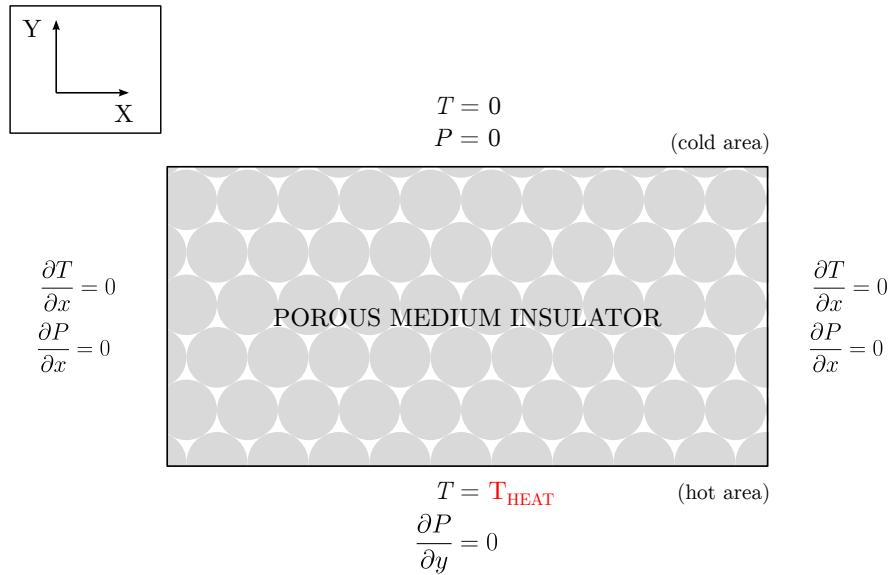


Figure 1.1: Porous medium insulator system representation

Due to the symmetry of the system in the X-direction, the developed code allows for a reduced 1D approach with the same results for this specific application and a much more efficient performance.

The system of equations modeling this approach is developed in [Section 2](#). The discretization methods both in terms of space and time are presented in [Section 3](#). Then, the implementation of the solver is explained in [Section 4](#). Finally, the simulations results and their validation are presented in [Section 5](#), while a brief insight on ways to improve performance is discussed in [Section 6](#).

2 Problem formulation

As developed in Section 1, the original system of equations is

$$\vec{q} = -\rho_{ref} k(\nabla p - \rho(T)\vec{g}), \quad (2.1)$$

$$\partial_t \rho_{ref} + \nabla \cdot \vec{q} = 0, \quad (2.2)$$

$$\partial_t(cT) - \nabla \cdot (\lambda \nabla T - c\vec{q}T) = 0, \quad (2.3)$$

where the unknowns and parameters are defined as follow:

p	-	pressure
T	-	temperature
\vec{q}	-	water mass flux
$\rho(T)$	$\rho_{ref} - \alpha(T - T_{ref})$	density
k	100	permeability
\vec{g}	(0, -1)	gravity vector
c	0.001	heat capacity
α	0.01	heat expansion coefficient
λ	0.01	heat conductivity
ρ_{ref}	1	reference density
T_{ref}	0	reference temperature
P_{ref}	0	reference pressure

The considered domain is defined as $\Omega = (0, 300) \times (0, 150)$. Let $\Gamma_{top} = (0, 300) \times 150$, $\Gamma_{bot} = (0, 300) \times 0$. Then, we set Dirichlet boundary conditions

$$\begin{aligned} P &= 0 && \text{on } \Gamma_{top}, \\ T &= 0 && \text{on } \Gamma_{top}, \\ T &= T_{HEAT} && \text{on } \Gamma_{bot}, \end{aligned}$$

and homogeneous Neumann boundary conditions for all other boundaries.

Several variables are considered constant, thus the system of equation is simplified. In [Equation 2.2](#), $\partial_t \rho_{ref} = 0$ because ρ_{ref} is constant. Therefore [Equation 2.2](#) simplifies as $\nabla \cdot \vec{q} = 0$. Also \vec{q} can be substituted in [Equation 2.3](#). In this way we only need to solve for P and T , this is

$$c \frac{\partial T}{\partial t} = \nabla \cdot (\lambda \nabla T + c \rho_{ref} k T (\nabla p - \rho(T) \vec{g})), \quad (2.4)$$

$$0 = \nabla \cdot (\nabla p + \alpha T \vec{g}). \quad (2.5)$$

Substituting ρ_{ref} ,

$$\frac{\partial T}{\partial t} = \nabla \cdot \left(\frac{\lambda}{c} \nabla T + \rho_{ref} k T \nabla p - \rho_{ref}^2 k T \vec{g} + \rho_{ref} k \alpha T^2 \vec{g} \right), \quad (2.6)$$

$$0 = \nabla \cdot (\nabla p + \alpha T \vec{g}). \quad (2.7)$$

Note that the only species of interest are temperature (T) and pressure (p) scalar fields. The water mass flux can be obtained from the later two with [Equation 2.1](#).

3 Discretizing the problem and solving the discretized problem

In this Section the discretization of the system of equations: [Equation 2.6](#), [Equation 2.7](#) and [Equation 2.3](#) is discussed by applying the method of lines. That means that the system is first discretized in space and thereafter in time. The space is discretized with the Finite Volume Method (FVM), while different approaches have been used and compared for the time discretization, using both classical algorithms (such as implicit and explicit Euler) and more sophisticated ones via `DifferentialEquations.jl` [9] package.

3.1 Space discretization

The spatial domain of the problem is given by $\Omega = (0, 300) \times (0, 150)$ and $\partial\Omega = \bigcup_{m \in \mathcal{G}} \Gamma_m$, where Γ_m are planar such that $\vec{n}|_{\Gamma_m} = \vec{n}_m$. The domain is subdivided into a finite number of control volumes $\bar{\Omega} = \bigcup_{k \in \mathcal{N}} \bar{\omega}_k$ such that

1. ω_k are open convex domains such that $\omega_k \cap \omega_l = \emptyset$ if $\omega_k \neq \omega_l$,
2. $\sigma_{kl} = \bar{\omega}_k \cap \bar{\omega}_l$ are either empty, points or straight lines. If $|\sigma_{kl}| > 0$ we say that ω_k, ω_l are neighbours,
3. $\vec{n}_{kl} \perp \sigma_{kl}$: normal of $\partial\omega_k$ at σ_{kl} ,
4. $\mathcal{N}_k = \{l \in \mathcal{N} : |\sigma_{kl}| > 0\}$: set of neighbours of ω_k ,
5. $\gamma_{km} = \partial\omega_k \cap \Gamma_m$: boundary part of $\partial\omega_k$,
6. $\mathcal{G}_k = \{m \in \mathcal{G} : |\gamma_{km}| > 0\}$: set of non-empty boundary parts of $\partial\omega_k$. This implies that $\partial\omega_k = (\cup_{l \in \mathcal{N}_k} \sigma_{kl}) \cup (\cup_{m \in \mathcal{G}_k} \gamma_{km})$.

To each control volume ω_k a collocation point $\vec{x}_k \in \bar{\omega}_k$ is assigned such that

1. the admissibility condition holds: if $l \in \mathcal{N}_k$ then the line $\vec{x}_k \vec{x}_l$ is orthogonal to σ_{kl} . For a given function $u : \Omega \rightarrow \mathbb{R}$ this will allow to associate its value $u_k = u(\vec{x}_k)$ as the value of an unknown at \vec{x}_k . For two neighboring control volumes ω_k, ω_l , this will allow to approximate $\vec{\nabla}u \cdot \vec{n}_{kl} \approx \frac{u_l - u_k}{h_{kl}}$.
2. the placement of boundary unknowns are at the boundary: if ω_k is situated at the boundary, i.e. for $|\partial\omega_k \cap \partial\Omega| > 0$, then $\vec{x}_k \in \partial\Omega$. This will allow to apply boundary conditions in a direct manner.

To obtain such a space discretization for a rectangular domain a regular grid is considered as shown in Figure 3.1 (and a labeled subdomain in Figure 3.2). The dots represent the collocation points, the thick lines are the boundaries of the rectangular control volumes and the dotted lines the connected graph of the collocation points. In this way the admissibility condition $\vec{x}_k \vec{x}_l \perp \sigma_{kl}$ and boundary placement of collocation points is fulfilled.

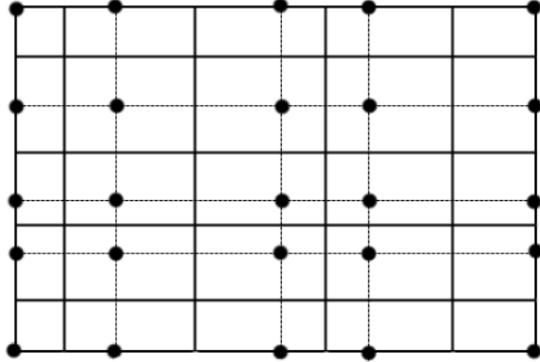


Figure 3.1: regular grid in 2D.

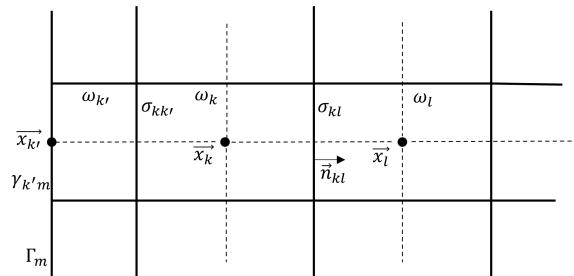


Figure 3.2: Sub domain with labels.

In addition Figure 3.3 shows the control volumes of the regular grid for the domain of the problem. Note that a local refinement is applied to the bottom boundary to capture steep gradient that arise due to a presence of a boundary layer.

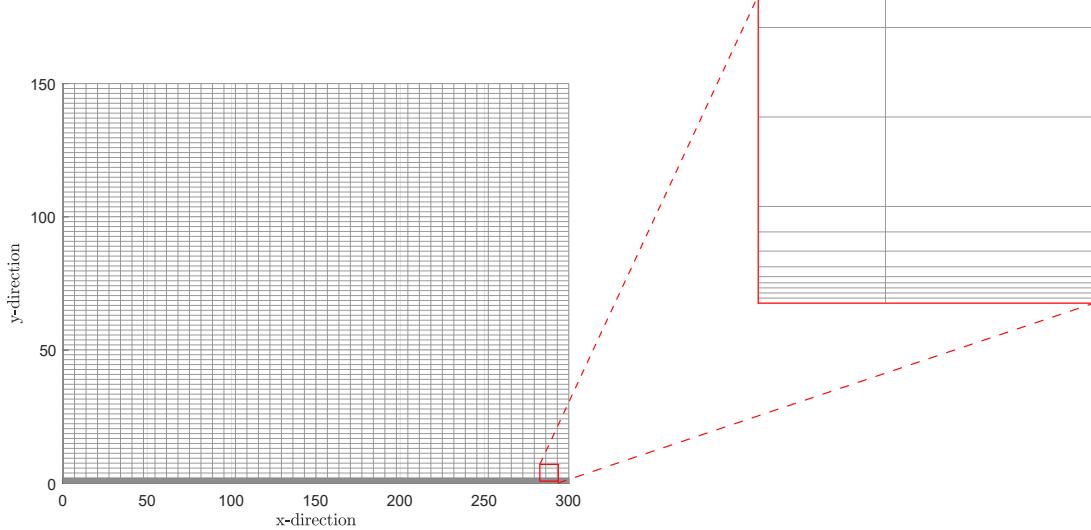


Figure 3.3: Regular mesh of the domain Ω with a close up of local refinement at the bottom boundary.

Having split the domain in rectangular control volumes, Equation 2.6 is discretized in space. Given control volume ω_k , integrate the equation over the control volume ω_k :

$$\int_{w_k} \partial_t(T) - \nabla \cdot \left(\frac{\lambda}{c} \nabla T + \rho_{ref} k T \nabla p - \rho_{ref}^2 k T \vec{g} + \rho_{ref} k \alpha T^2 \vec{g} \right) d w_k = 0 \quad (3.1)$$

$$|w_k| \partial_t T_k - \int_{\partial w_k} \left(\frac{\lambda}{c} \nabla T + \rho_{ref} k T \nabla p - \rho_{ref}^2 k T \vec{g} + \rho_{ref} k \alpha T^2 \vec{g} \right) \cdot \vec{n}_{w_k} ds = 0 \quad (3.2)$$

$$\begin{aligned} |w_k| \partial_t T_k - \sum_{l \in N_k} \int_{\partial \sigma_{kl}} \left(\frac{\lambda}{c} \nabla T + \rho_{ref} k T \nabla p - \rho_{ref}^2 k T \vec{g} + \rho_{ref} k \alpha T^2 \vec{g} \right) \cdot \vec{n}_{kl} ds \\ = 0 \\ - \sum_{m \in G_k} \int_{\partial \gamma_{km}} \left(\frac{\lambda}{c} \nabla T + \rho_{ref} k T \nabla p - \rho_{ref}^2 k T \vec{g} + \rho_{ref} k \alpha T^2 \vec{g} \right) \cdot \vec{n}_m ds \end{aligned} \quad (3.3)$$

$$\begin{aligned} |w_k| \partial_t T_k - \sum_{l \in N_k} |\sigma_{kl}| \left(\frac{\lambda}{c} \frac{T_k - T_l}{h_{kl}} + \rho_{ref} k \frac{T_k p_k - T_l p_l}{2 h_{kl}} - \rho_{ref}^2 k \frac{T_k + T_l}{2} \vec{g} \cdot \vec{n}_{kl} + \rho_{ref} k \alpha \frac{T_k^2 + T_l^2}{2} \vec{g} \cdot \vec{n}_{kl} \right) \\ = 0 \\ - \sum_{m \in G_k} \int_{\partial \gamma_{km}} \left(\frac{\lambda}{c} \nabla T + \rho_{ref} k T \nabla p - \rho_{ref}^2 k T \vec{g} + \rho_{ref} k \alpha T^2 \vec{g} \right) \cdot \vec{n}_m ds \end{aligned} \quad (3.4)$$

Some terms over the integral are approximated, those are

- the term $\nabla T \cdot \vec{n}_{kl}$ in the integral over σ_{kl} , which is approximated as $\frac{T_k - T_l}{h_{kl}}$, where $h_{kl} = |\vec{x}_k - \vec{x}_l|$,
- $T \nabla p \cdot \vec{n}_{kl}$ as $\frac{T_k p_k - T_l p_l}{2 h_{kl}}$,
- T as $\frac{T_k + T_l}{2}$, and
- T^2 as $\frac{T_k^2 + T_l^2}{2}$.

The last term in [Equation 3.4](#) represents the flux in and out the domain Ω . The value of this term depends on the location of the control volume. If the control volume borders the right or left boundary of the domain a homogeneous Neumann boundary is prescribed for the pressure and temperature, consequently the terms $\nabla T \cdot \vec{n}_m$ and $\nabla p \cdot \vec{n}_m$ drop out. If the control volume borders the top or bottom boundary a Dirichlet boundary condition is prescribed. To approach such a boundary the penalty method is applied. The existing Dirichlet boundary conditions are replaced by a Robin boundary condition with a small penalty parameter ϵ multiplying the Neumann part of the robin boundary:

$$\begin{aligned} T = 0 \text{ on } \Gamma_{\text{top}} &\implies \nabla T \cdot \vec{n}_{\text{top}} + \frac{1}{\epsilon} T = 0 \\ T = T_{\text{heat}} \text{ on } \Gamma_{\text{bot}} &\implies \nabla T \cdot \vec{n}_{\text{bot}} + \frac{1}{\epsilon} T = \frac{1}{\epsilon} T_{\text{heat}} \\ p = 0 \text{ on } \Gamma_{\text{top}} &\implies \nabla p \cdot \vec{n}_{\text{top}} + \frac{1}{\epsilon} p = 0 \end{aligned}$$

Applying the penalty method gives:

$$|w_k| \partial_t T_k - \sum_{l \in N_k} |\sigma_{kl}| \left(\frac{\lambda}{c} \frac{T_k - T_l}{h_{kl}} + \rho_{ref} k \frac{T_k + T_l}{2} \frac{p_k - p_l}{h_{kl}} - \rho_{ref}^2 k \frac{T_k + T_l}{2} \vec{g} \cdot \vec{n}_{kl} + \rho_{ref} k \alpha \frac{T_k^2 + T_l^2}{2} \vec{g} \cdot \vec{n}_{kl} \right) - \sum_{m \in G_k} |\gamma_{km}| \begin{cases} \frac{\lambda}{c} \left(\frac{1}{\epsilon} T_{heat} - \frac{1}{\epsilon} T_k \right) - \rho_{ref} k T_{heat} \vec{g} \cdot \vec{n}_m + \rho_{ref} k \alpha T_{heat}^2 \vec{g} \cdot \vec{n}_m & \text{if control volume } k \text{ edges } \Gamma_{bottom} = 0 \\ - \frac{\lambda}{c} \frac{1}{\epsilon} T_k - \rho_{ref} k \frac{1}{\epsilon} p_k & \text{if control volume } k \text{ edges } \Gamma_{top} \\ (-\rho_{ref}^2 k T_k \vec{g} + \rho_{ref} k \alpha T_k^2 \vec{g}) \cdot \vec{n}_m & \text{otherwise} \end{cases} \quad (3.5)$$

Next [Equation 2.6](#) is discretized in a similar fashion. Given control volume ω_k , integrate [Equation 2.6](#) over control volume ω_k :

$$\int_{\omega_k} \nabla \cdot (\nabla p + \alpha T \vec{g}) d\omega_k = 0 \quad (3.6)$$

$$\int_{\partial \omega_k} \nabla p \cdot \vec{n}_w + \alpha T \vec{g} \cdot \vec{n}_w ds = 0 \quad (3.7)$$

$$\sum_{l \in N_k} \int_{\sigma_{kl}} \nabla p \cdot \vec{n}_{kl} + \alpha T \vec{g} \cdot \vec{n}_{kl} ds + \sum_{m \in G_k} \int_{\gamma_{km}} \nabla p \cdot \vec{n}_m + \alpha T \vec{g} \cdot \vec{n}_m ds = 0 \quad (3.8)$$

$$\sum_{l \in N_k} |\sigma_{kl}| \left(-\frac{p_k - p_l}{h_{kl}} + \frac{\alpha}{2} (T_k + T_l) \vec{g} \cdot \vec{n}_{kl} \right) + \sum_{m \in G_k} \int_{\gamma_{km}} \nabla p \cdot \vec{n}_m + \alpha T \vec{g} \cdot \vec{n}_m ds = 0 \quad (3.9)$$

The term $\nabla p \cdot \vec{n}_{kl}$ in the integral over σ_{kl} is approximated as $\frac{p_k - p_l}{h_{kl}}$, where $h = |\vec{x}_k - \vec{x}_l|$. The term p is approximated as the average of the pressure evaluated at the collocation point of the two neighbouring control volumes, i.e $p \approx \frac{1}{2}(p_k - p_l)$. The flux in and out of the domain Ω is approached in a similar fashion as the discretization of [Equation 2.7](#).

$$\sum_{l \in N_k} |\sigma_{kl}| \left(-\frac{p_k - p_l}{h_{kl}} + \frac{\alpha}{2} (T_k + T_l) \vec{g} \cdot \vec{n}_{kl} \right) + \sum_{m \in G_k} |\gamma_{km}| \begin{cases} \alpha T_{heat} \vec{g} \cdot \vec{n}_m & \text{if control volume } k \text{ edges } \Gamma_{bottom} \\ -\frac{1}{\epsilon} p_k & \text{if control volume } k \text{ edges } \Gamma_{top} \\ \alpha T_k \vec{g} \cdot \vec{n}_m & \text{otherwise} \end{cases} = 0 \quad (3.10)$$

At last \vec{q} is discretized. Given control volume w_k , integrate [Equation 2.1](#) over control volume w_k :

$$\int_{w_k} \vec{q} d\omega_k = \int_{w_k} -\rho_{ref} k (\nabla p - \rho_{ref} \vec{g} + \alpha (T - T_{ref}) \vec{g}) d\omega_k \quad (3.11)$$

$$|w_k| \vec{q}_k = \int_{w_k} -\rho_{ref} k (\nabla p - \rho_{ref} \vec{g} + \alpha (T - T_{ref}) \vec{g}) d\omega_k \quad (3.12)$$

$$|w_k| \vec{q}_k = -|w_k| \rho_{ref} k \left(\frac{1}{|w_k|} \int_{w_k} \nabla p d\omega_k - \rho_{ref} \vec{g} + \alpha \left(\frac{1}{|w_k|} \int_{w_k} T d\omega_k - T_{ref} \right) \vec{g} \right) \quad (3.13)$$

$$\vec{q}_k = -\rho_{ref} k \left(\frac{1}{|w_k|} \int_{w_k} \nabla p d\omega_k - \rho_{ref} \vec{g} + \alpha (T_k - T_{ref}) \vec{g} \right) \quad (3.14)$$

Note that p_k and T_k are obtained from the discretization of the equation [Equation 2.6](#) and [Equation 2.7](#).

In conclusion the spatial discretization over a control volume w_k is given by [Equation 3.10](#) and [Equation 3.5](#). For each control volume $k \in \{1, \dots, \mathcal{N}\}$ we have two equations, one linear and one nonlinear, and each control volume has 2 unknowns. All these equations together give rise to a nonlinear system of equation which can be written as:

$$M\partial_t u + A(u) = 0$$

where u is the vector containing the unknowns T_k and p_k (i.e. $u = [T_1 \dots T_{|\mathcal{N}|} p_1 \dots p_{|\mathcal{N}|}]^T := [T \ p]^T$) with size $2|\mathcal{N}|$, M a matrix of size $2|\mathcal{N}| \times 2|\mathcal{N}|$ and $A(u)$ can be seen as a nonlinear operator $A : D \rightarrow \mathbb{R}^n$ where $D \subset \mathbb{R}^n$ is its domain of definition.

Furthermore

$$M = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix},$$

where $I \in \mathbb{R}^{|\mathcal{N}| \times |\mathcal{N}|}$. Exploiting this block structure and defining $f(T, p) = MA(T, p)$ and $g(T, p) = (I - M)(A(T, p))$ the system is reduced to equations [3.16](#).

Note that the discretized water mass flux q_k is determined using [Equation 3.14](#) after p_k and T_k are obtained. In conclusion the spatial part is discretized with the finite volume method. As a result a huge nonlinear DAE system is obtained.

3.2 Time discretization

After discretizing spacial derivatives with finite volume method, [Equation 2.6](#) and [Equation 2.7](#) turn into a system of differential algebraic equations (DAE) of the form

$$\frac{dT}{dt} = f(T, P), \quad (3.15)$$

$$0 = g(T, P), \quad (3.16)$$

where f and g are nonlinear functions, f represents differential part and g is algebraic constraint

DAE [3.16](#) has a special form, called semi-explicit DAE. Very important characteristics of DAEs is so called strangeness index. For [Equation 3.16](#) this index is 0. Strangeness free DAEs behave nicely when solved by numerical methods, which will come in handy in the next part. The numerical solution of the DAE system is discussed in [Section 4](#).

4 Solving the nonlinear system

4.1 Steady case

To obtain a steady solution of [Equation 2.6](#) and [Equation 2.7](#), all species are considered time independent, so the time derivative of temperature, $\frac{\partial T}{\partial t}$, is set to zero. The resulting equations are

$$\begin{aligned} 0 &= \nabla \cdot \left(\frac{\lambda}{c} \nabla T + \rho_{ref} k T \nabla p - \rho_{ref}^2 k T \vec{g} + \rho_{ref} k \alpha T^2 \vec{g} \right), \\ 0 &= \nabla \cdot (\nabla p + \alpha T \vec{g}). \end{aligned} \quad (4.1)$$

The resulting system of equations can be denoted as $F(p, T) = F(u) = 0$. If the domain is discretized with n control volumes, then $F : R^{2n} \rightarrow R^{2n}$. Finding temperature and pressure that satisfy [Equation 4.1](#) is equivalent to finding a root of F .

To solve such a nonlinear system the Newton's method is considered. Given $F(u) = 0$ the Newton's method iteratively solves a nonlinear system with the following iteration scheme

$$u_{i+1} = u_i - J(u_i)^{-1}(F(u_i)),$$

where J is the Jacobi matrix. The iteration is terminated when the residual reaches a specified value. For large systems, the most expensive part of the algorithm is the construction of the Jacobi matrix. It is possible to speed up this step by using sparse matrices for J . The sparsity pattern looks as shown in [Figures 4.1](#) and [4.2](#).

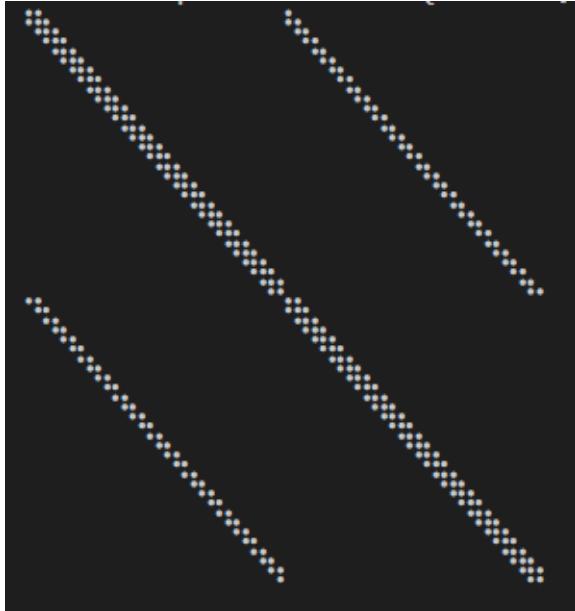


Figure 4.1: 1D Jacobian sparsity pattern



Figure 4.2: 2D Jacobian sparsity pattern

Newton's method converges quadratically near the solution, but can diverge if the initial guess is far from the solution, therefore it is important to find good starting values. By the problem formulation, pressure and temperature at the top and bottom of the domain are known, so these values are included in the initial guess u_0 .

4.2 Transient solution

Transient solution is obtained by solving 3.16 numerically. It can be rewritten in the form

$$E \frac{du}{dt} = \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & & 0 \end{bmatrix} \frac{du}{dt} = \begin{bmatrix} f_1(u) \\ \vdots \\ f_n(u) \\ g_1(u) \\ \vdots \\ g_n(u) \end{bmatrix}, \quad (4.2)$$

where $\mathbf{u} = (T_1, T_2..T_n, P_1, P_2...P_n)$.

It can be seen that explicit methods cannot be applied to this problem because it would require an inverse of matrix E, which is singular. Before solving the DAE, one has to prescribe an initial condition. Not every initial condition is consistent with the constraint Equation 2.7. To find suitable initial condition, first initial temperature distribution is assigned complying with the Dirichlet boundary conditions, then temperature is substituted in 2.7 and solved for pressure using Newton's method.

As was mentioned in Section 3, [Equation 3.16](#) is strangeness free, and according to the literature [8] it can be solved by any implicit stiffly-accute Runge-Kutta method or any implicit BDF method. Both approaches were tried.

For Runge-Kutta, Implicit Euler, Radau2A and Rodas5 was tried using DifferentialEquations.jl package.

For BDF method, IDA method from Sundials.jl was used. All methods produced same results, with Rodas5 being faster for the same step size. Also DifferentialEquations.jl allowed to specify Jacobian sparsity, therefore Rodas5 was used for final simulations with fine mesh due to less memory usage and linear time complexity(instead of quadratic as for dense matrices). Transient solution for 2D case was not obtained due to large number of control volumes, resulting in too long computing times. Code optimization is required to achieve this.

5 Simulation results

The key insight of the present study is to solve the non-linear system of differential equations in the spatial and temporal domain using a finite volume approach.

Hitherto we discussed the discretization and solving strategies. This section shows the results of the performed simulations.

Unless otherwise specified, the presented results correspond to calculations performed with the mesh of Figure 3.3 and using Rodas5 as time-solver, a 5th order A-stable stiffly stable Rosenbrock method recommended for medium tolerance applications. Additionally, due to the x-direction symmetry of the problem these results are obtained with the 1D solver introduced in Section 3 and expanded to the whole domain, allowing for a much more efficient computation.

First, it is interesting to study temperature and pressure distributions with completely homogeneous boundary conditions (both Dirichlet and Neumann). Under these conditions, and taking into account that the reference temperature and pressure are both 0, constant and null steady-state distributions are to be expected. As can be seen in Figures 5.1 and 5.2, this condition is fulfilled.

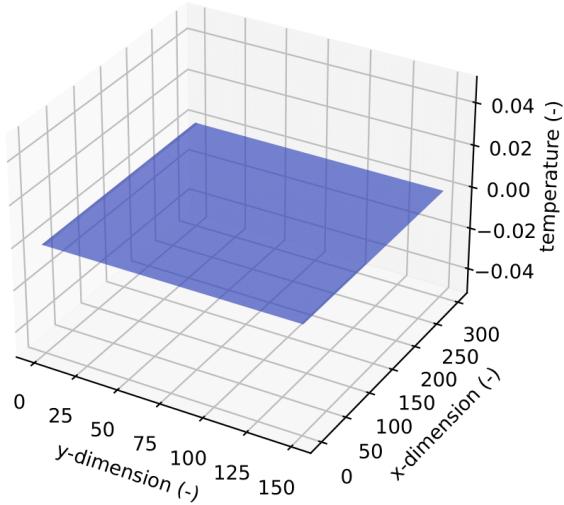


Figure 5.1: Temperature profile in steady state for $T_{heat} = 0$

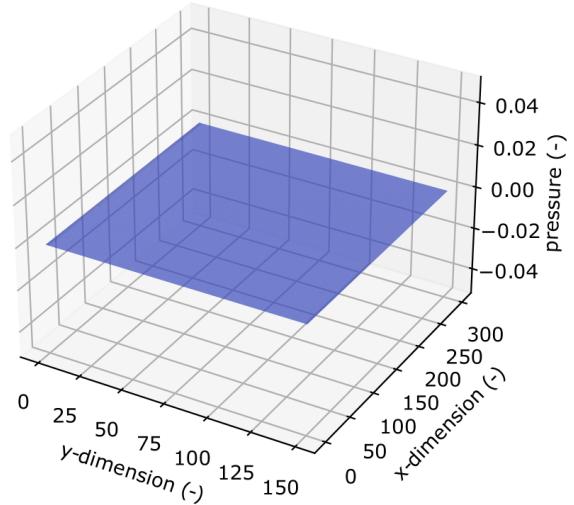


Figure 5.2: Pressure profile in steady state for $T_{heat} = 0$

Following the physical basis of the problem, the next calculation to be performed is to study the evolution of the steady case as the temperature boundary condition at the base (heated from below) is increased. Figure 5.3 and 5.4 show the temperature and pressure distribution, respectively, with $T_{heat} = 0.5$.

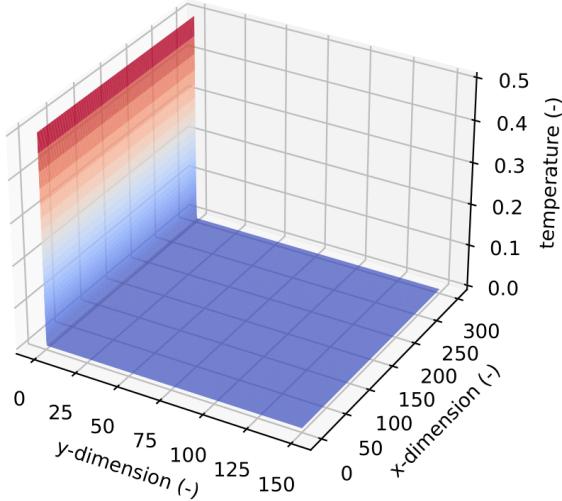


Figure 5.3: Temperature profile in steady state with $T_{heat} = 0.5$

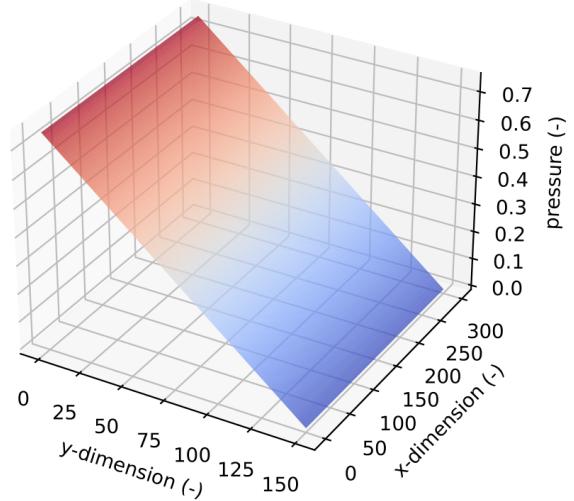


Figure 5.4: Pressure profile in steady state with $T_{heat} = 0.5$

It can be seen that the pressure distribution is quasi-linear, while the temperature distribution is practically constant except for a boundary layer zone in the hot zone of the domain.

Regarding the boundary layer, its presence is an effect of the porous system acting as an insulator, and mathematically induced by the non-linearity of the energy conservation equation. In fact, as can be seen in Figure 5.5, as the thermal conductivity increases this boundary layer becomes much less pronounced, thus temperature inside the boundary layer increases with the thermal conductivity. This is true because with an increase in thermal conductivity (decrease in the Prandtl number) the heat transfer rate increases, consequently thickening the thermal boundary layer. It can be seen that the temperature field tends to a linear distribution, as the non-linear terms of the modelling equations become neglectable (see Equation 2.5).

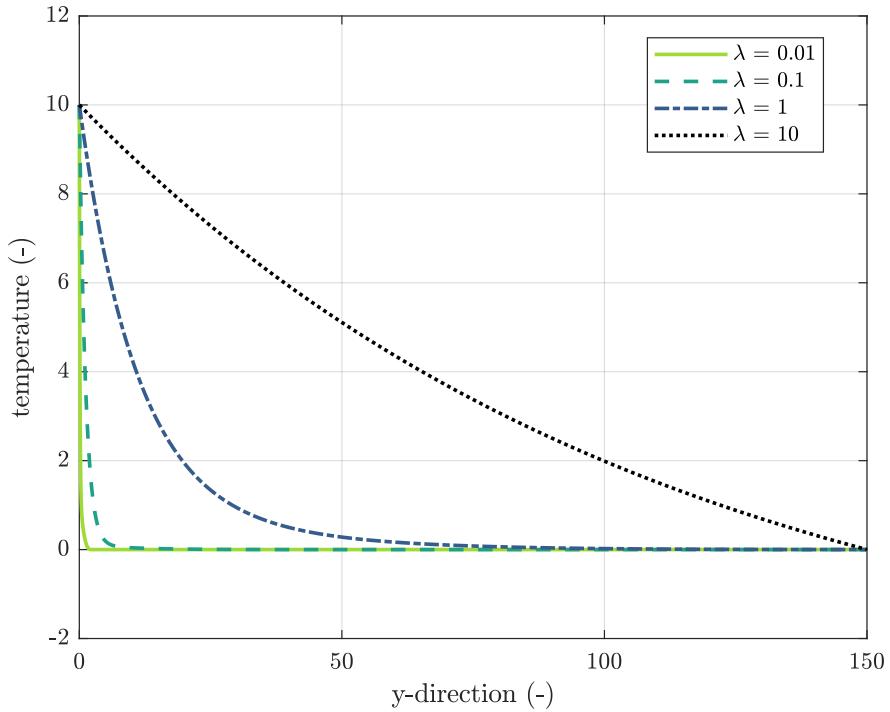


Figure 5.5: Variation of the temperature profile at the boundary layer with the heat conductivity λ

Having characterized the steady state, we proceed to analyze the transient temperature and pressure for different boundary conditions. For more details on the time discretization of these simulations, please refer to Section 3.2.

The initial conditions for temperature are defined as a linear distribution between the temperature at the boundaries in order to comply with the boundary conditions. Note that the change in shape observed at $t = 0$ s is just an arbitrary choice of initial conditions. Any initial temperature distribution could be studied, but we settled on piece-wise linear. The method for finding initial condition for pressure is described in 4.2

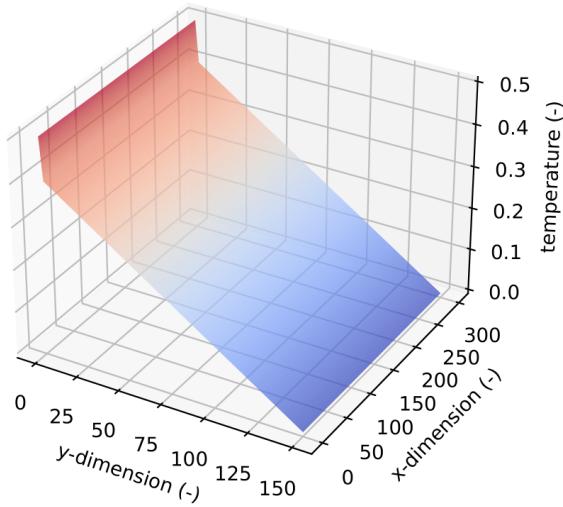


Figure 5.6: Temperature profile in $t = 0$ s for $T_{heat} = 0.5$

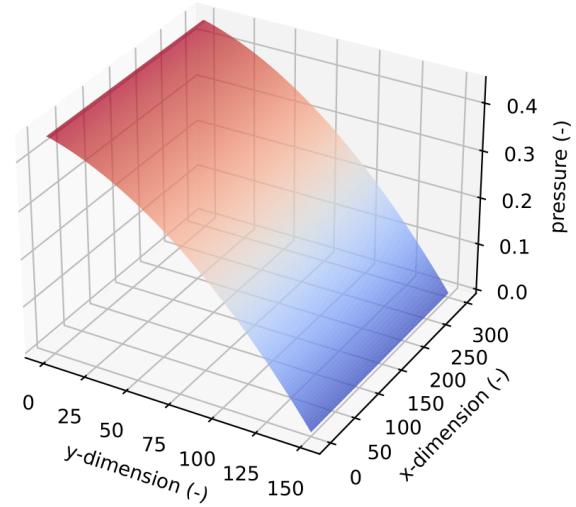


Figure 5.7: Pressure profile in $t = 0$ s for $T_{heat} = 0.5$

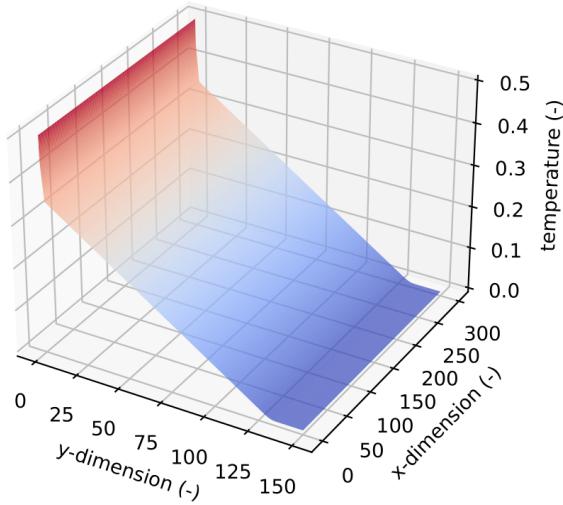


Figure 5.8: Temperature profile in $t = 1$ s for $T_{heat} = 0.5$

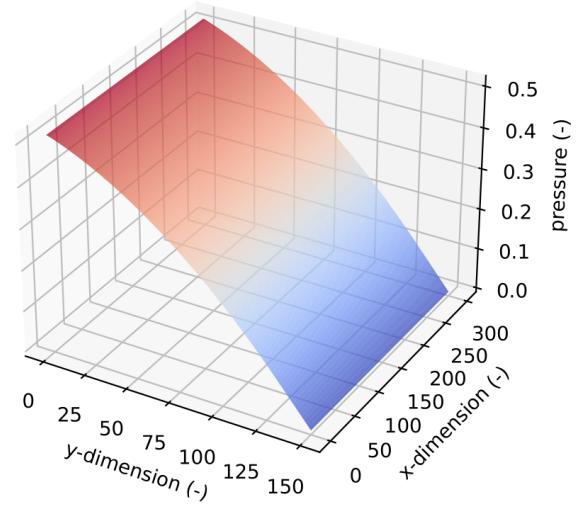


Figure 5.9: Pressure profile in $t = 1$ s for $T_{heat} = 0.5$

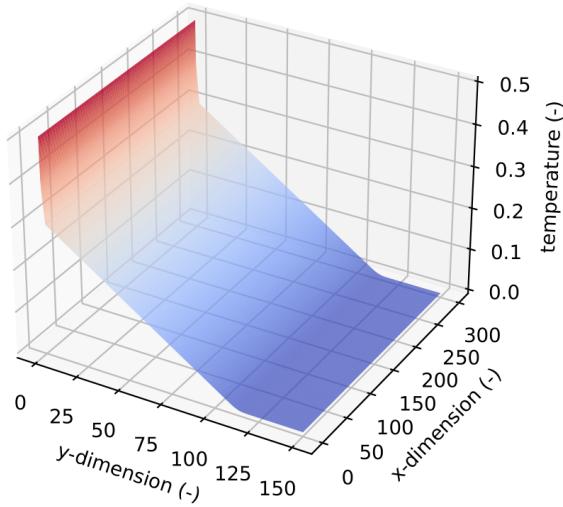


Figure 5.10: Temperature profile in $t = 2$ s for $T_{heat} = 0.5$

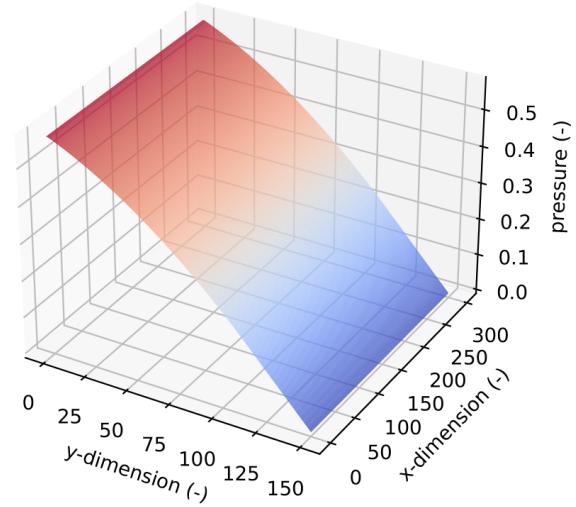


Figure 5.11: Pressure profile in $t = 2$ s for $T_{heat} = 0.5$

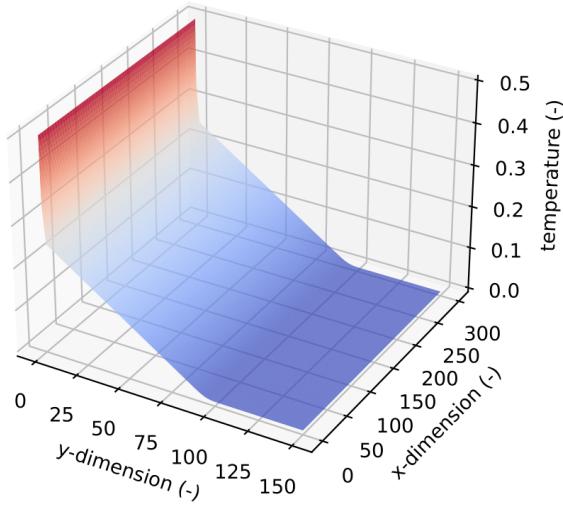


Figure 5.12: Temperature profile in $t = 3$ s for $T_{heat} = 0.5$

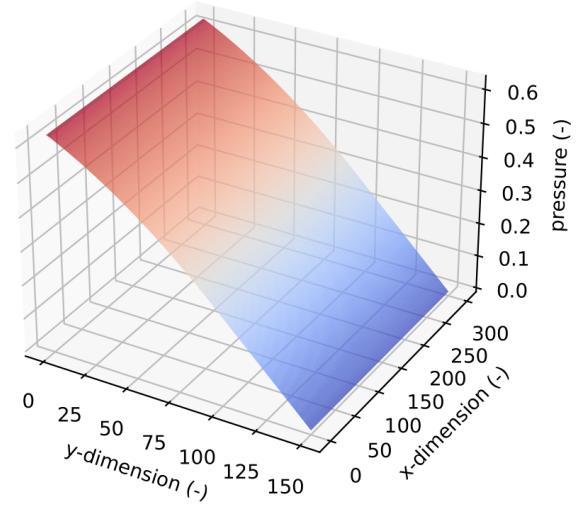


Figure 5.13: Pressure profile in $t = 3$ s for $T_{heat} = 0.5$

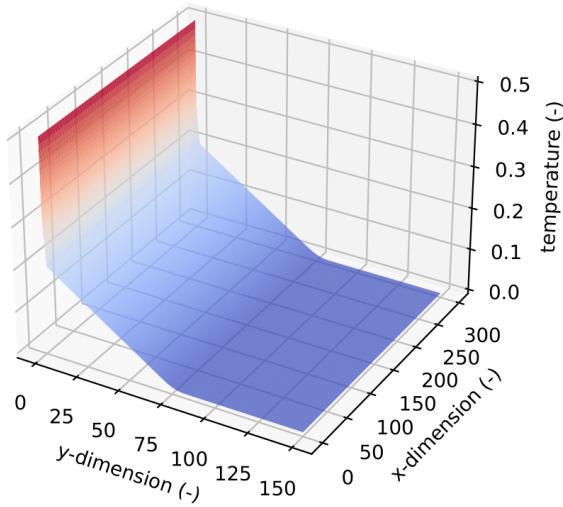


Figure 5.14: Temperature profile in $t = 4$ s for $T_{heat} = 0.5$

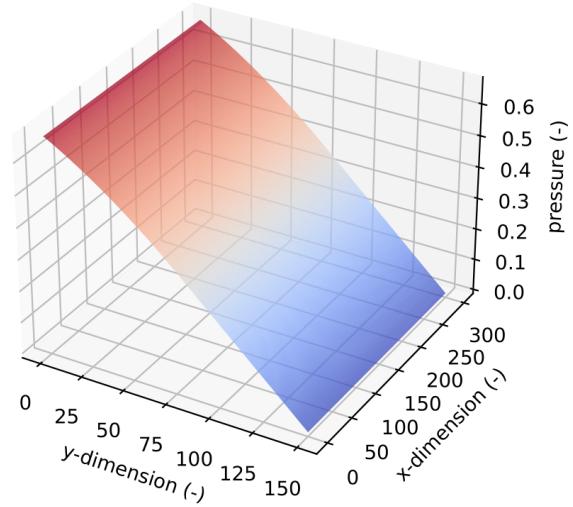


Figure 5.15: Pressure profile in $t = 4$ s for $T_{heat} = 0.5$

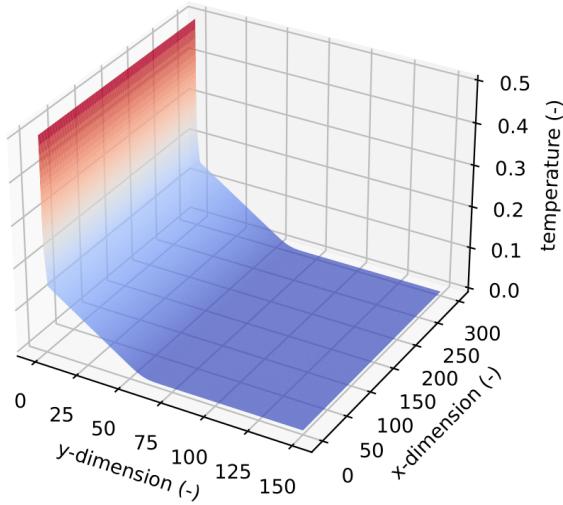


Figure 5.16: Temperature profile in $t = 5$ s for $T_{heat} = 0.5$

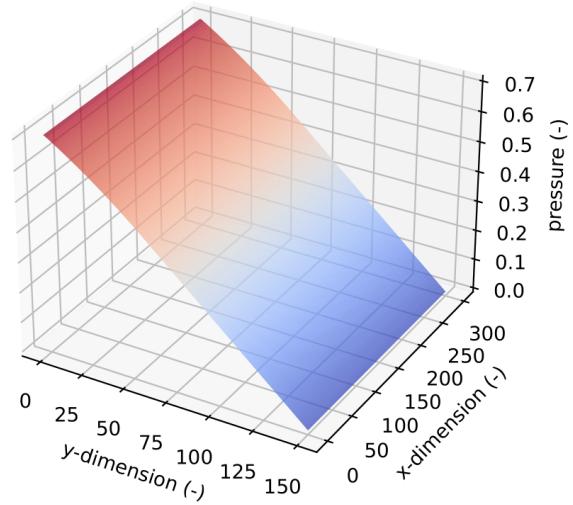


Figure 5.17: Pressure profile in $t = 5$ s for $T_{heat} = 0.5$

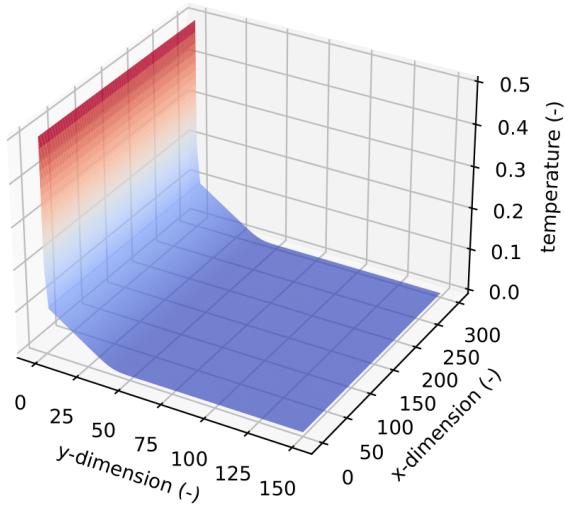


Figure 5.18: Temperature profile in $t = 6$ s for $T_{heat} = 0.5$

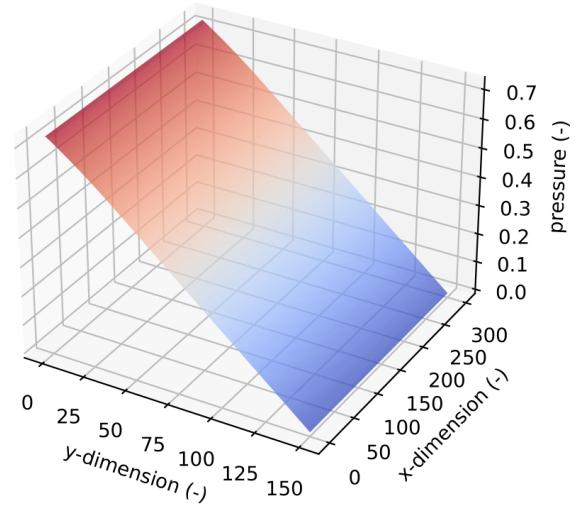


Figure 5.19: Pressure profile in $t = 6$ s for $T_{heat} = 0.5$

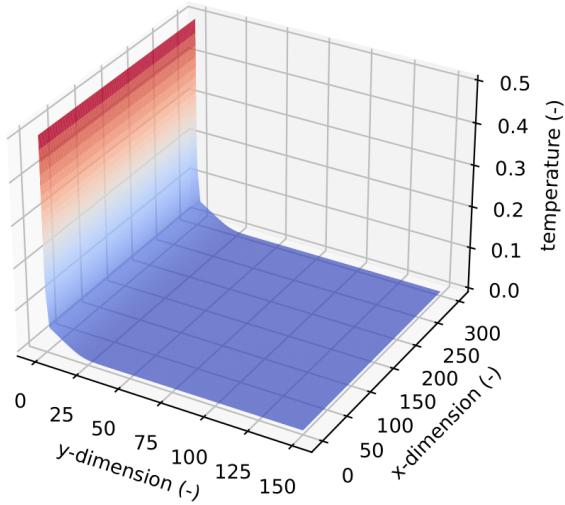


Figure 5.20: Temperature profile in $t = 7$ s for $T_{heat} = 0.5$

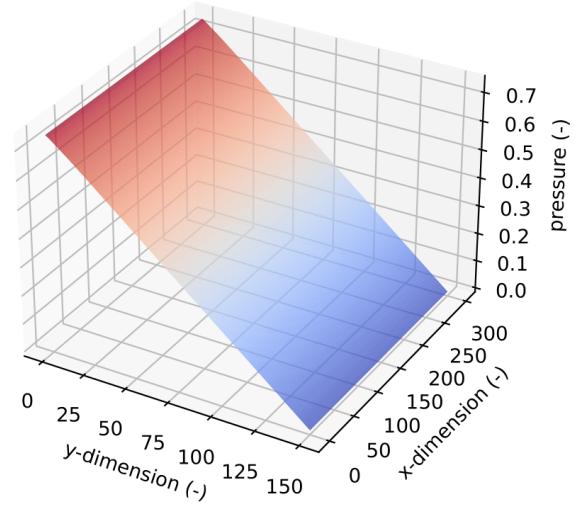


Figure 5.21: Pressure profile in $t = 7$ s for $T_{heat} = 0.5$

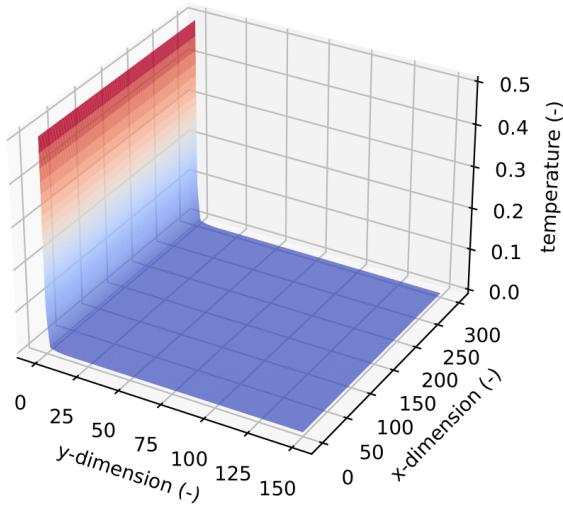


Figure 5.22: Temperature profile in $t = 8$ s for $T_{heat} = 0.5$

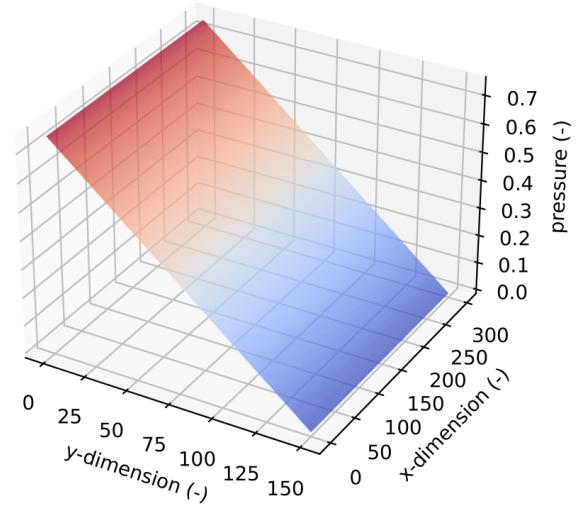


Figure 5.23: Pressure profile in $t = 8$ s for $T_{heat} = 0.5$

For the sake of simplicity in the report, although the bottom boundary temperature is a simple parameter to modify in the model and the solver, the number of figures necessary for the analysis of the transistor is very high, so we will directly consider a case in which the temperature is increased by one order of magnitude, setting $T_{heat} = 10$.

As can be seen comparing Figures 5.6 to 5.23 with Figures 5.24 to 5.50, the difference is merely quantitative, but no qualitative differences are observed in the time evolution of the transient or in the time in which the solution reaches the steady state.

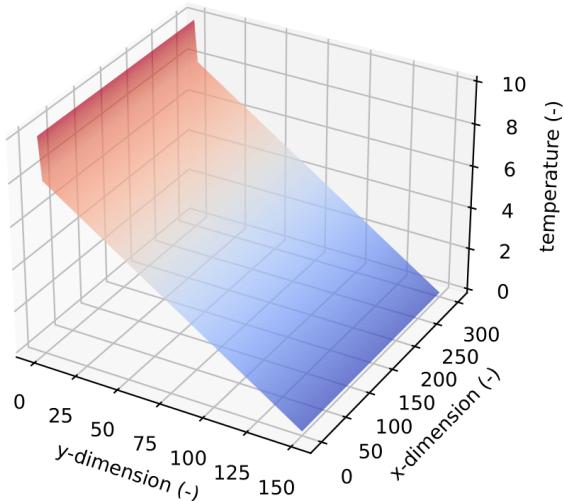


Figure 5.24: Temperature profile in $t = 0$ s for $T_{heat} = 10$

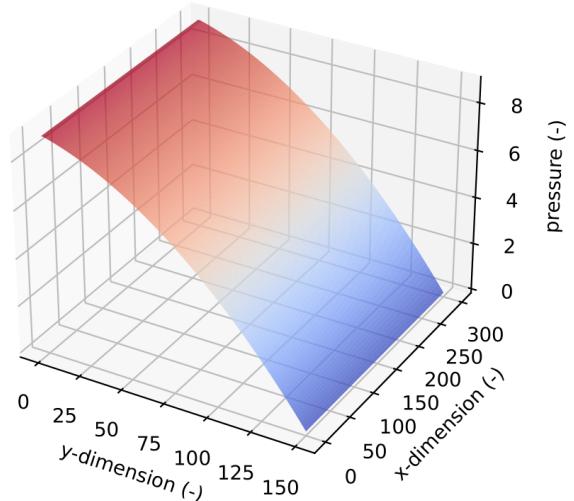


Figure 5.25: Pressure profile in $t = 0$ s for $T_{heat} = 10$

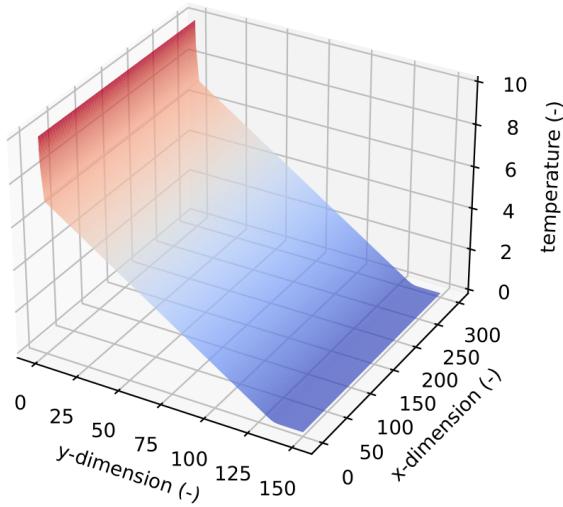


Figure 5.26: Temperature profile in $t = 1$ s for $T_{heat} = 10$

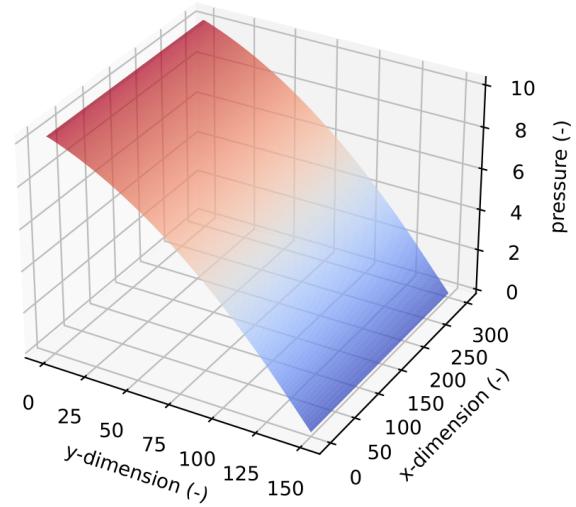


Figure 5.27: Pressure profile in $t = 1$ s for $T_{heat} = 10$

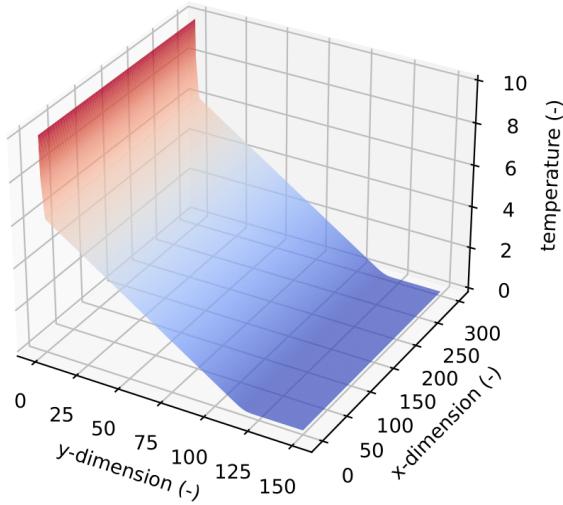


Figure 5.28: Temperature profile in $t = 2$ s for $T_{heat} = 10$

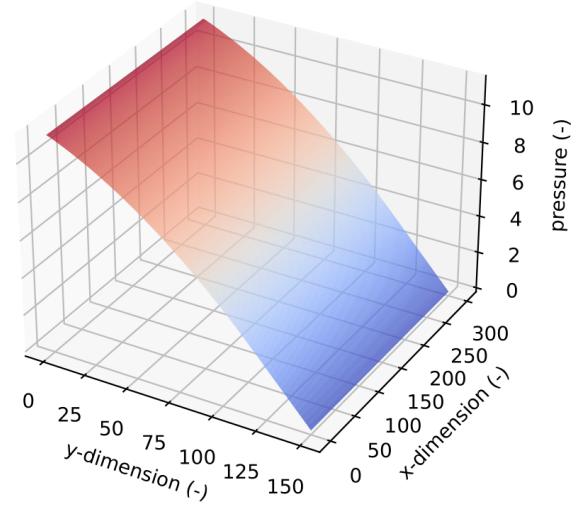


Figure 5.29: Pressure profile in $t = 2$ s for $T_{heat} = 10$

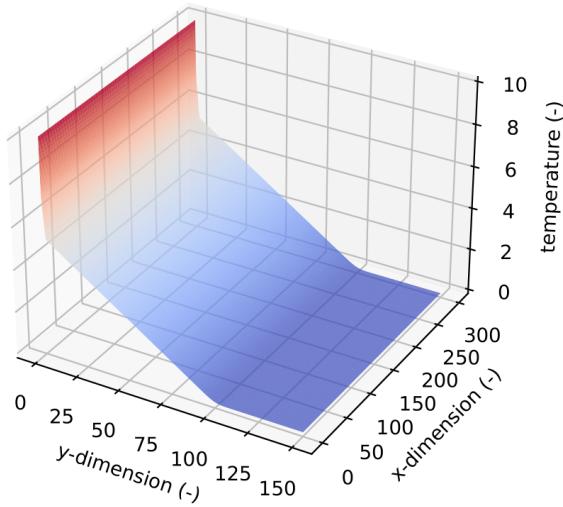


Figure 5.30: Temperature profile in $t = 3$ s for $T_{heat} = 10$

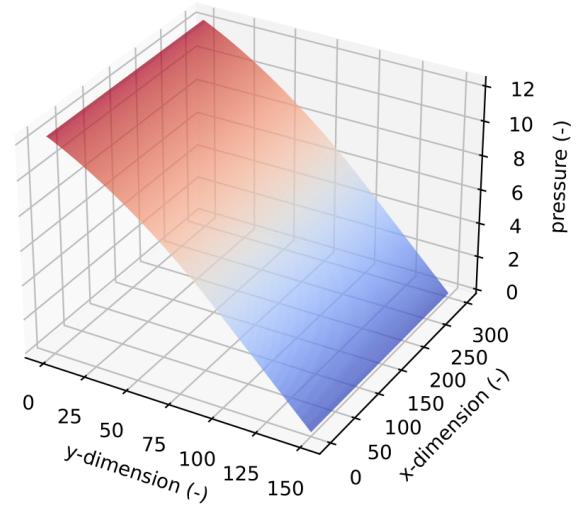


Figure 5.31: Pressure profile in $t = 3$ s for T_{heat} = 10

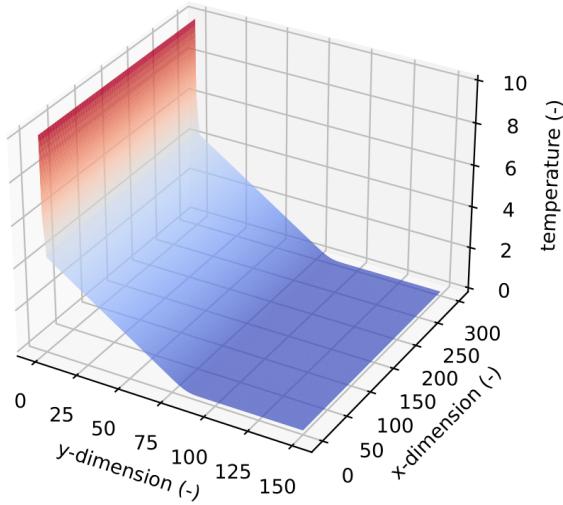


Figure 5.32: Temperature profile in $t = 4$ s for $T_{heat} = 10$

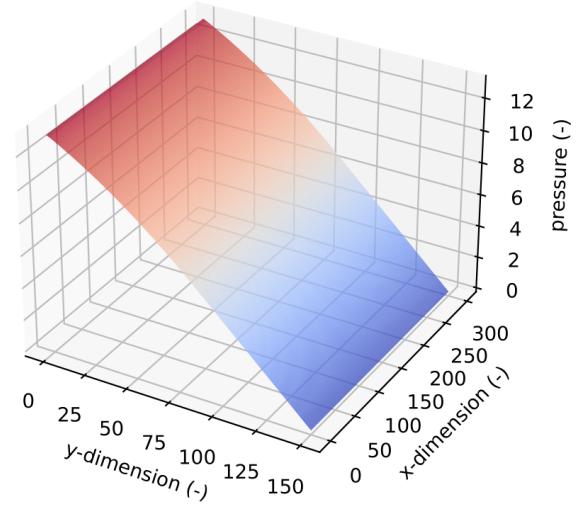


Figure 5.33: Pressure profile in $t = 4$ s for T_{heat} = 10

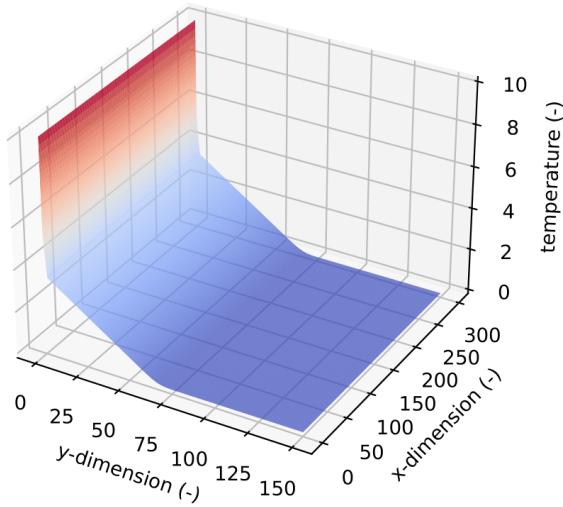


Figure 5.34: Temperature profile in $t = 5$ s for $T_{heat} = 10$

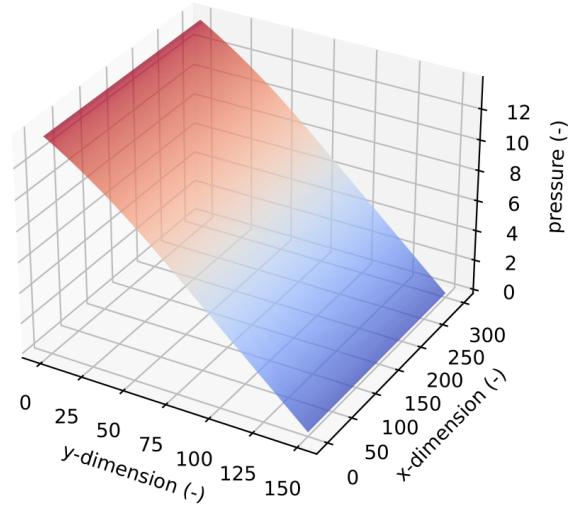


Figure 5.35: Pressure profile in $t = 5$ s for $T_{heat} = 10$

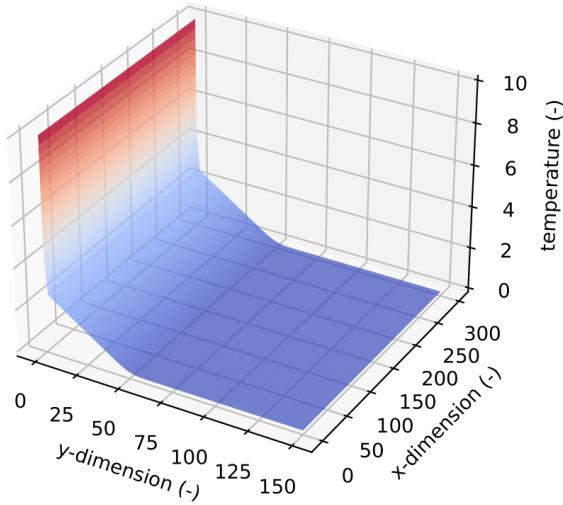


Figure 5.36: Temperature profile in $t = 6$ s for $T_{heat} = 10$

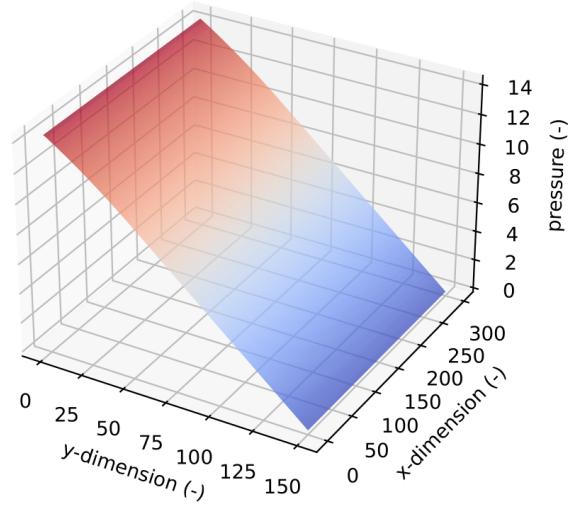


Figure 5.37: Pressure profile in $t = 6$ s for $T_{heat} = 10$

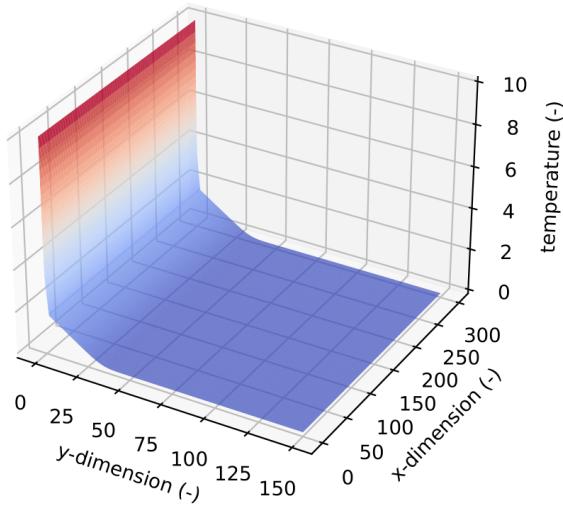


Figure 5.38: Temperature profile in $t = 7$ s for $T_{heat} = 10$

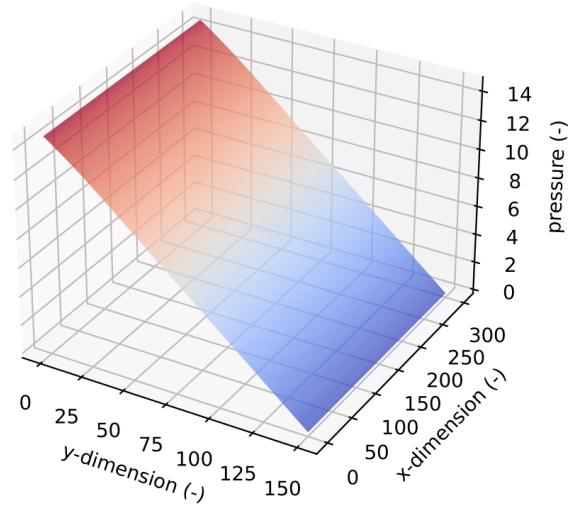


Figure 5.39: Pressure profile in $t = 7$ s for $T_{heat} = 10$

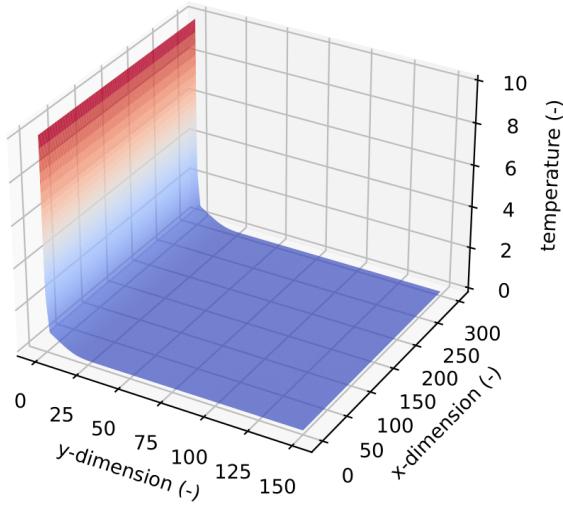


Figure 5.40: Temperature profile in $t = 8$ s for $T_{heat} = 10$

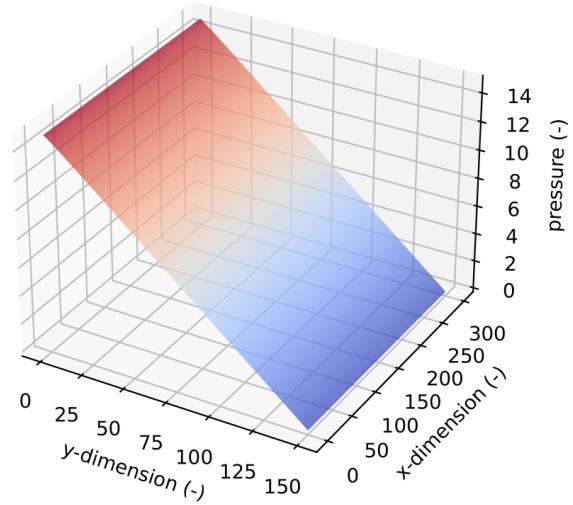


Figure 5.41: Pressure profile in $t = 8$ s for $T_{heat} = 10$

As mentioned in the problem statement, since some of the parameters included in the mathematical modeling of the system are constant and taking into account that according to Darcy's law the water mass flux is proportional to the pressure gradient and the gravitational force, said flux can be calculated as a post-process of the pressure and temperature maps.

Figure 5.42 shows this flux for $T = 10$ at a point close to stationary ($t = 8$ s).

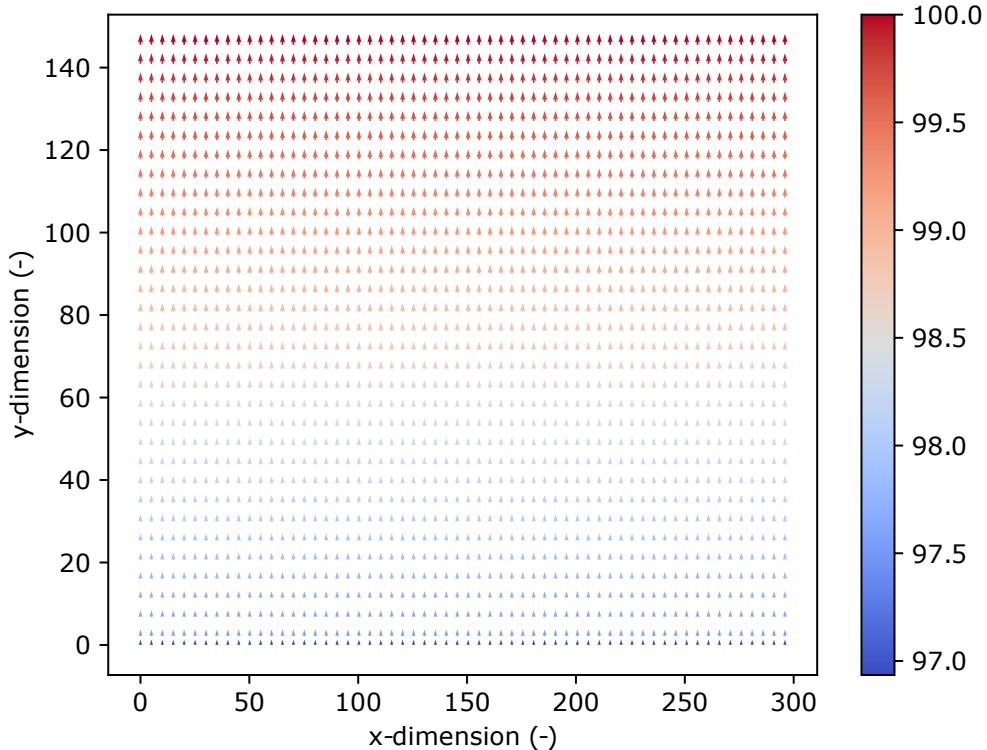


Figure 5.42: Water mass flux in $t = 8.0$ s

Finally, it is considered relevant to validate the simplification according to which the solver used is 1D instead of 2D due to the symmetry of the boundary value problem. Especially when solving the transient, the 1D code presents computation times several orders of magnitude lower than those of 2D, making it an economically more efficient solution.

In addition, because the boundary conditions induce a flow only in the y-direction, the result obtained in 2D is equivalent, as can be seen in Figures 5.43 to 5.46. As a consequence, the 1D version of the code is validated, so it is advisable to use it in these conditions for the sake of efficiency.

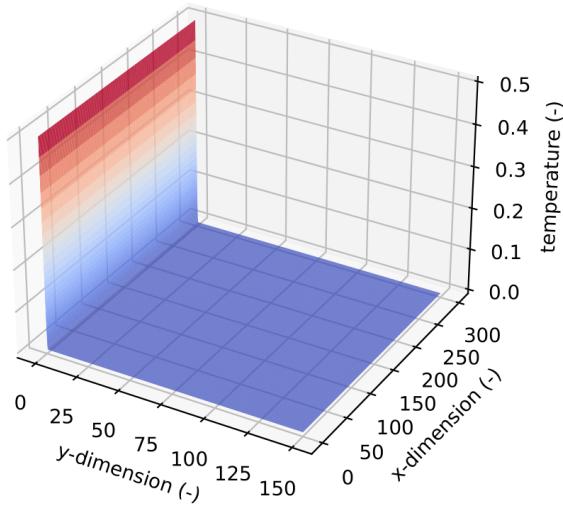


Figure 5.43: Temperature profile in steady state with $T_{heat} = 0.5$ using the 1D FVM solver

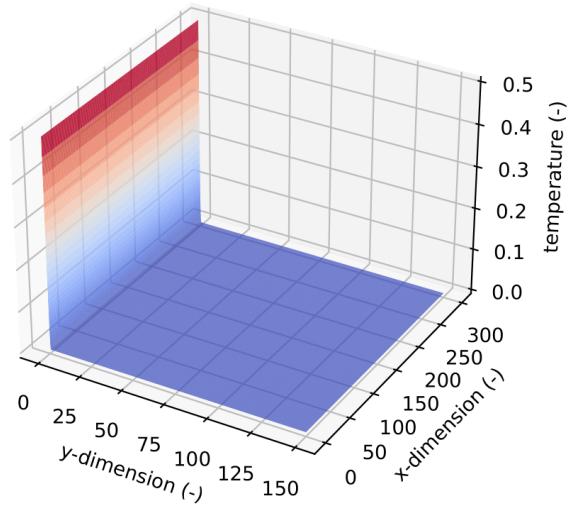


Figure 5.44: Temperature profile in steady state with $T_{heat} = 0.5$ using the 2D FVM solver

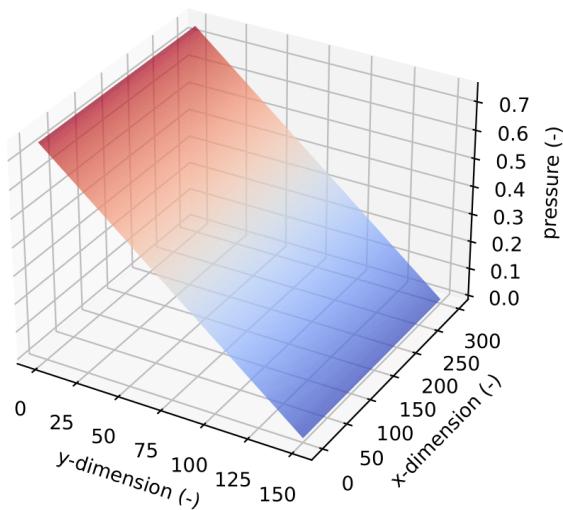


Figure 5.45: Pressure profile in steady state with $T_{heat} = 0.5$ using the 1D FVM solver

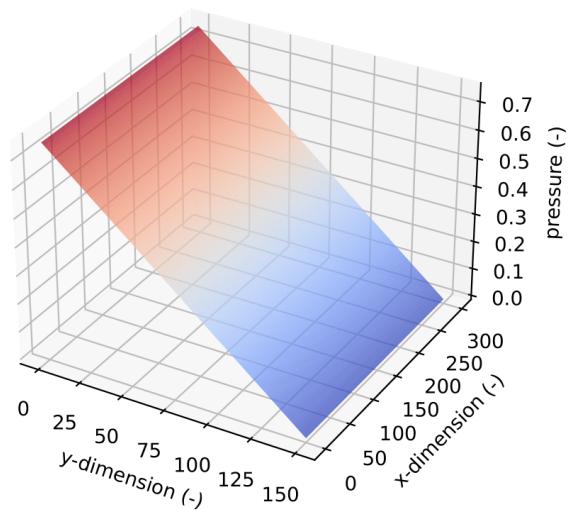


Figure 5.46: Pressure profile in steady state with $T_{heat} = 0.5$ using the 2D FVM solver

Apart from the benefits of the efficiency of the code, the 1D solver in transient case has been used due to time constraints and because an own-code developed 2D solver has been considered beyond the scope of the project. Thus, the developed code only solves the stationary case when the complete problem is considered in 2D. Even so, it is still useful for solving BVP in which the boundary conditions are not complex or induce composite flows in the 2 spatial dimensions.

Due to the fact that the cases requested by the problem statement only induce cases that can be solved with the 1-dimensional simplification, as a bonus some stationary distributions have been calculated with several cases in which the boundary conditions are clearly 2D. As can be seen in Figures 5.47 to 5.52, the correct operation of the code has been validated.

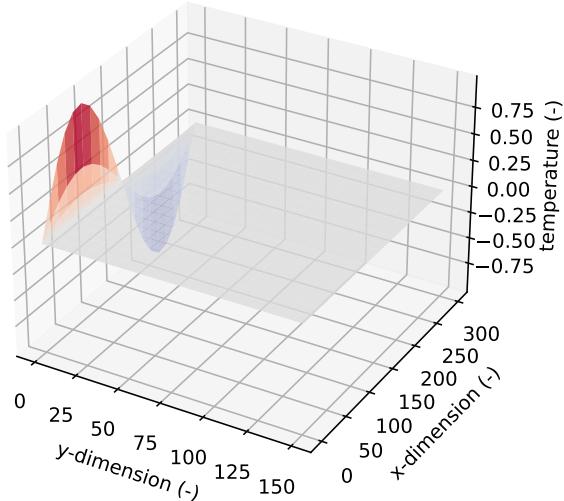


Figure 5.47: Temperature profile in steady state with $T_{heat} = \sin\left(\frac{2\pi}{300}x\right)$

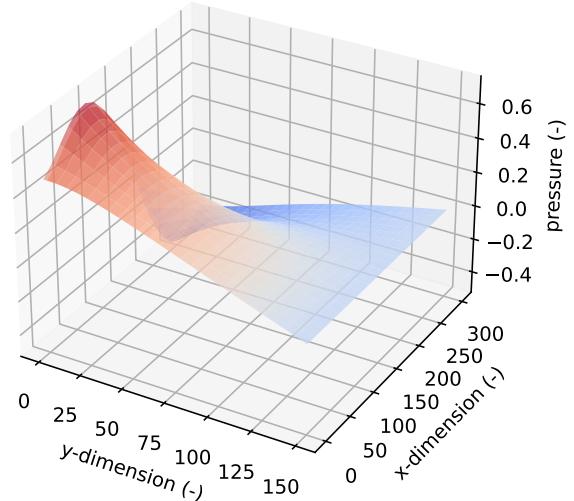


Figure 5.48: Pressure profile in steady state with $T_{heat} = \sin\left(\frac{2\pi}{300}x\right)$

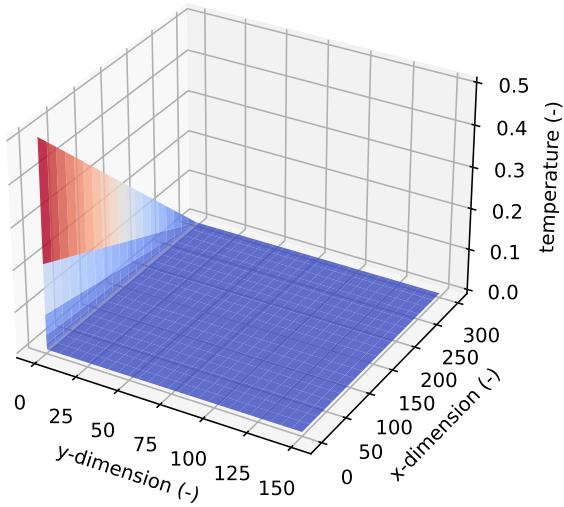


Figure 5.49: Temperature profile in steady state with $T_{heat} = 0.5 - x/600$

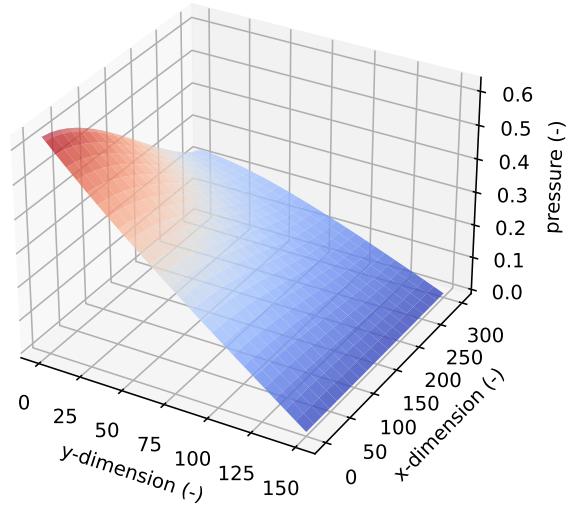


Figure 5.50: Pressure profile in steady state with $T_{heat} = 0.5 - x/600$

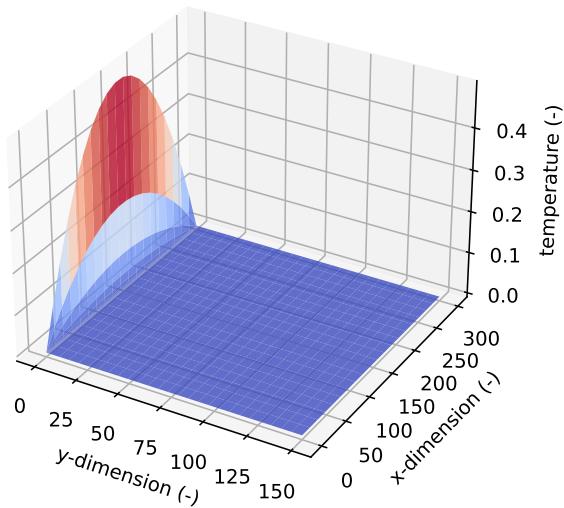


Figure 5.51: Temperature profile in steady state with $T_{heat} = 0.5 - 0.5(1 - x/150)^2$

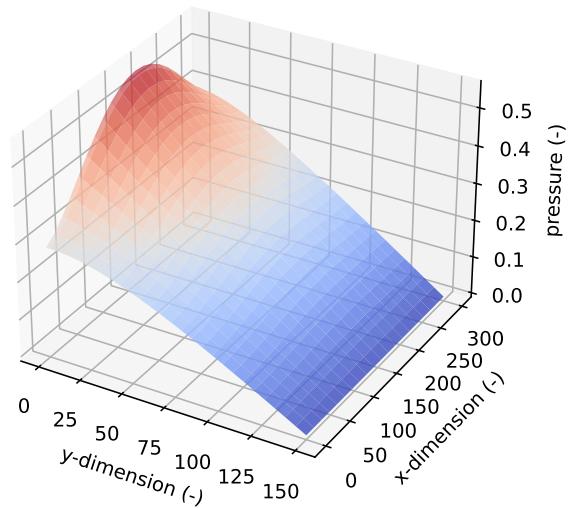


Figure 5.52: Pressure profile in steady state with $T_{heat} = 0.5 - 0.5(1 - x/150)^2$

6 Ways to improve performance

Throughout the whole project, and as it has been introduced in the different sections, decisions have been made in order to improve the performance of the custom code developed.

First of all, the use of a non-uniform mesh (Figure 3.3) has allowed to reduce the number of finite volumes needed in the model by several orders of magnitude. The large temperature gradient found near the thermal boundary layer zone requires a high amount of elements in that area to achieve stability in the results, but this requirement is not necessary in most of the domain. Although this process has been developed by the authors, future improvements can introduce adaptive mesh refinement (AMR) strategies, so that the mesh refinement process is dynamic and performed during calculation based on error and solution stability estimators. Additionally, current implementation of the solver uses admissible rectangular mesh, which means that the mesh cannot be refined in a small region of interest in both x and y dimension without also creating extra cells in other regions.

After the discretization of the system with the finite volume method, we need to solve the resulting system using a Newton-based solver, in which the system is linearized at each step. The given linear system is generally large and sparse and has to be solved via a linear solver method. This part being one of the most demanding one in terms of computation time. Although direct methods are robust and accurate, using them is unsuitable in large system, and iterative methods are preferred.

Then, one important way to improve performance is to remove type instabilities, which prevent efficient compilation of julia code to assembler. Also 2D solver has to be written in a way to work with dual numbers. This will allow to use automatic differentiation for Jacobian computations. Parallelization would yield to a big performance improvement, but it requires a significant programming effort.

7 Conclusions

In this report Darcy's equations are solved using finite volume method. To do this, equations are simplified and written in the conservative form.

A custom FVM solver utilizing rectangular control volumes is written. For simple boundary conditions the solution only varies along one axis, therefore 1D simulation is sufficient. It is found that pressure and temperature fields have a sharp gradient (boundary layer) near the heated surface, while the rest of the domain remains similar to the non-heated surface in terms of temperature. A non-uniform mesh (adaptive refinement) is used with smaller cells where needed.

In addition the solution behaviour with more exotic boundary conditions is performed with a 2D simulation. As with 1D case, sharp gradients of pressure and temperature near heated surfaces are also present.

The transient solution is found by solving a resulting DAE with Implicit Euler, Radau2A and BDF methods. Simulation is done for 1D case. For a 2D grid, the system size is too large for current implementation of the solver to simulate in a reasonable time. As a future improvement, the code for 2D simulation has to be rewritten to fully take advantage of sparse Jacobians, additionally type instabilities have to be removed.

Bibliography

- [1] Ping Cheng. "Heat transfer in geothermal systems". In: *Advances in heat transfer*. Vol. 14. Elsevier, 1979, pp. 1–105.
- [2] DK Gartling and CE Hickox. "A numerical study of the applicability of the Boussinesq approximation for a fluid-saturated porous medium". In: *International Journal for Numerical Methods in Fluids* 5.11 (1985), pp. 995–1013.
- [3] Donald D Gray and Aldo Giorgini. "The validity of the Boussinesq approximation for liquids and gases". In: *International Journal of Heat and Mass Transfer* 19.5 (1976), pp. 545–551.
- [4] AH Meghdadi Isfahani and Masoud Afrand. "Experiment and Lattice Boltzmann numerical study on nanofluids flow in a micromodel as porous medium". In: *Physica E: Low-dimensional Systems and Nanostructures* 94 (2017), pp. 15–21.
- [5] R McKibbin. "Thermal convection in layered and anisotropic porous media: a review". In: *Convective flows in porous media* (1985), pp. 113–127.
- [6] DA Nield. "Recent research on convection in a porous medium". In: *Proceedings of CSIRO/DSIR Seminar on Convective Flows in Porous Media. DSIR, Wellington, New Zealand*. 1985.
- [7] Mehdi Nojoomizadeh et al. "Investigation of permeability effect on slip velocity and temperature jump boundary conditions for FMWNT/Water nanofluid flow and heat transfer inside a microchannel filled by a porous media". In: *Physica E: Low-dimensional Systems and Nanostructures* 97 (2018), pp. 226–238.
- [8] Volker Mehrmann Peter Kunkel. *Differential algebraic equations*. 2006.
- [9] Christopher Rackauckas and Qing Nie. "DifferentialEquations.jl—a performant and feature-rich ecosystem for solving differential equations in julia". In: *Journal of Open Research Software* 5.1 (2017), p. 15.
- [10] Osvair V Trevisan and Adrian Bejan. "Mass and heat transfer by high Rayleigh number convection in a porous medium heated from below". In: *International Journal of Heat and Mass Transfer* 30.11 (1987), pp. 2341–2356.