

Development of a conjugate heat transfer solver in FEniCS

SZÉCHENYI ISTVÁN EGYETEM

FACULTY OF MECHANICAL ENGINEERING, INFORMATICS
AND ELECTRICAL ENGINEERING



Author:

Bernold Rodrigo Abarca Zúñiga

Computer Science MSc

Supervisor:

Dr. Lotfi Abdelhakim

Associate professor

Győr, 2024

Contents

1	Introduction	2
2	Governing Equations	5
2.1	Interface conditions	6
2.2	Initial conditions	6
3	Numerical Approximation	8
3.1	Weak formulation	8
3.2	Numerical discretization	9
3.3	Implicit time discretization	9
3.4	Spatial discretization	9
4	Algorithm implementation under Fenics	11
4.1	Mesh creation	11
4.2	Function spaces	12
4.3	Boundary conditions	13
4.4	Variational problems	14
4.5	Time-stepping loop	15
4.6	Output results	15
5	Applications	16
5.1	Convection Cooling of Circuit boards	16
5.1.1	Materials properties	18
5.1.2	Boundary and initial conditions	19
5.2	Coaxial Borehole Heat Exchanger	20
5.2.1	Material properties	22
5.2.2	Boundary and initial conditions	23
6	Numerical Results	24

6.1	Numerical results: Convection Cooling of Circuit boards	24
6.2	Numerical results: Coaxial Borehole Heat Exchanger	26
7	Conclusions	29
	Bibliography	29
	List of Figures	32
	List of Tables	33
A	Own Results	34
A.1	Scientific question	34
A.2	Research methods	34
A.3	Results	34
B	Use of Artificial Intelligence	35
C	Full code	36

Abstract

The main objective of this work is to develop a solver for conjugate heat transfer problems using the Finite Element Method and investigate its applications in thermal engineering problems. The next purpose is to demonstrate that open-source computing platforms, such as FeniCS and Gmsh, can be used to solve complicated problems in science and engineering.

The conjugate heat transfer describes the interaction of heat conduction in solid materials and convection in fluids, crucial for engineering applications like domestic heating and cooling systems, cooling electronic devices, and industrial heat exchangers. These processes are modeled using time-dependent mathematical models, which consist of mass and momentum conservation and energy conservation equations. In this work, the fluid is assumed to be incompressible, and the Boussinesq approximation is used to simulate the buoyancy effects.

The algorithm developed to solve the conjugate heat transfer problem is based on a partitioned coupling approach, meaning that the flow and heat transfer problems are modeled independently and solved interactively through a time-marching algorithm.

To assess the accuracy of the developed coupled solvers, two examples are presented: the cooling of a circuit board and a coaxial borehole heat exchanger. In the verification process, the results obtained were compared with those reported in the literature. Results show that the open-source framework Fenics provides an efficient and flexible computing platform to handle complex computational challenges.

Keywords

Conjugate heat transfer, Navier-Stokes, finite element method, FEniCS project, convection cooling of circuit boards, coaxial borehole heat exchanger

Notation

Symbol	Description
u	Fluid velocity vector
ρ	Material density
ρ_f	Fluid density
p	Pressure
μ	Dynamic viscosity
α	Thermal expansion coefficient
g	Gravitational acceleration
T	Temperature vector
T_0	Initial temperature vector
c_v	Specific heat capacity
t	Current time
γ	Thermal conductivity
f	Heat source term
f_f	Force term in fluid domain (bouyancy)
Ω	Domain of the problem
∇	Gradient operator
Δ	Laplacian operator
k	Thermal conductivity coefficient
v	Test function for velocity
q	Test function for pressure
w	Test function for temperature
n	Normal vector at the interface or boundary
T_s	Temperature in the solid domain
T_f	Temperature in the fluid domain
ν	Kinematic viscosity ($\frac{\mu}{\rho_f}$)
u_n	Previous time step velocity
T_n	Previous time step temperature
dx	Volume integration element

Table 1: Notation table for symbols used in the article

Chapter 1

Introduction

In recent years, both open-source and commercial Partial Differential Equation (PDE) solvers have advanced significantly, allowing researchers and engineers to model complex physical processes with precision. For tasks such as heat transfer, fluid dynamics, and structural analysis, commercial solvers like ANSYS Fluent [1] and COMSOL Multiphysics [2] are widely used due to their extensive support and pre-built functionalities. However, many independent researchers and small organizations cannot afford these solutions due to their licensing fees.

In the other hand, open-source alternatives, such as FEniCS, have emerged as powerful and accessible options that rival commercial software in both flexibility and performance. FEniCS, for instance, provides an efficient and highly customizable environment for solving PDEs, making it suitable for complex problems like conjugate heat transfer. Its key strengths include strong Python integration, active community support, versatility, and high-quality and precise results[3].

The conjugate heat transfer (CHT) is a physical phenomenon that describes the process of heat conduction in solid and fluid domains simultaneously, where the two domains of heat transfer are coupled at a solid-fluid interface [4]. It considers the mutual effects of thermal conduction in the solid and convection in the fluid, providing a more comprehensive analysis of the heat transfer processes [5]. This is crucial for thermal engineering problems that require accurate predictions of the temperature profiles and heat transfer rates in systems involving both solid and fluid domains. Domestic heating and cooling systems, industrial heat exchangers, and cooling electronics are examples of these types of engineering problems [6]. In this work, we focused on 2 important applications: the cooling of a circuit board [7] and a coaxial borehole heat exchanger [8].

Conjugate heat transfer problems can be modeled by partial differential equations (PDE).

However, due to their complex behavior, analytical solutions are generally difficult or impossible to find, requiring numerical methods to approximate the solution. The method used in this article is the Finite element method (FEM). It is one of the most popular methods for solving PDEs and is widely employed in engineering and physics [9]. The main advantages of this method are its flexibility in discretizing irregular domains, applying various boundary conditions, material properties, and its ability to perform localized mesh refinement [10].

The algorithm developed for solving the CHT problems via the FEM is based on a partitioned coupling approach [11], which allows us to solve the decomposed problem interactively through a time marching algorithm. It decomposes the CHT problem into two independent sub-problems, the Navier-Stokes and heat transfer problems, and solve them independently and simultaneously over N time steps. Finally, the coupling is ensured by exchanging results data between the two sub-problems at each time step.

This algorithm was developed using the open-source framework Dolfin, part of the FEniCS project. It offers an expressive and efficient environment for solving partial differential equations (PDEs) using the finite element method. We selected this framework not only for the strengths mentioned earlier but also for its high level of mathematical abstraction, extensibility, customization options, and the possibility to create high-performance programs [3].

Because of the importance of the conjugate heat transfer in industrial applications and the powerful capabilities of the Dolfin framework for solving complex PDEs, we decided to explore how to solve this type of problems using Dolfin. The aim is to solve CHT problems and show its applications in thermal engineering problem, while also demonstrating how Dolfin can be effectively used to solve computationally demanding problems, offering both practical insights and advanced computational solutions.

The article is organized as follows:

- Chapter 2: introduces the mathematical problem by the governing equations that model the CHT.
- Chapter 3: explains how the problem is solved by using the weak formulation of the equations and applying a spacial and time discretization.
- Chapter 4: describe the logic implemented in the solver algorithm.
- Chapter 5: provides a technical description of the specifics cases covered in the article and highlights their relevance in engineering.

- Chapter 6: shows and explains the obtained numerical results for each case and compare them with the literature results.
- Chapter 7: presents discussion about the results and future work.

Chapter 2

Governing Equations

The governing equations for the CHT between a solid and a fluid are presented in this sections. The fluid is assumed to be incompressible, the physical properties of the fluid and solid are independent of the temperature and the Boussinesq approximation is used to simulate the buoyancy effects. These governing equations are:

Continuity equation,

$$\nabla \cdot \mathbf{u} = 0 \quad (2.1)$$

Momentum equation,

$$\frac{\partial}{\partial t}(\rho_f \mathbf{u}) + \mathbf{u} \cdot (\rho_f \nabla \mathbf{u}) = -\nabla p + \mu \nabla^2 \mathbf{u} - \alpha \rho_f g (T - T_0) \quad (2.2)$$

Energy equation,

$$\rho c_v \left(\frac{\partial T}{\partial t} + (\mathbf{u} \cdot \nabla) T \right) = \nabla \cdot (\gamma \nabla T) + f \quad (2.3)$$

Where $\mathbf{u}$, $\mathbf{p}$, $\mathbf{T}$ are the velocity, pressure and temperature vectors, t is the time, ρ is the material density (ρ_f fluid density), μ is the dynamic viscosity, α is the thermal expansion coefficient, g is the gravitational acceleration, T_0 is the initial temperature, c_v is material heat capacity, γ is material thermal conductivity and f is the heat source.

In this formulation, the flow problem and thermal problem are coupled by two terms. First, the buoyancy term is a source term that depends on temperature. The second term is the convective term, which is based on the velocity of the fluid in the convection-diffusion equation for the temperature variable.

To solve a CHT problem, two different approaches can be used. The first one is the monolithic approach, where the CHT problem is solved simultaneously in one global system of non-linear equations to solve for the velocity u , the pressure p and the temperature T simultaneously

[12]. This approach is not efficient in terms of the required computing time. By contrast, the second one is the staggered approach [11], where the CHT problem is decomposed into two independent sub-problems, which can be solved independently and simultaneously with algorithms developed for the Navier-Stokes and heat transfer problems respectively. In this case, the coupling is ensured by transferring results data between the two sub-problems. An efficient implementation of this method might reduce the amount of computation time and memory requirements.

In this work, the developed algorithm is using the second approach. It solves iteratively the 2 sub-problems, first, the Navier-stokes in the fluid domain and second, the heat transfer in all domains, and their results are used to couple the problem.

2.1 Interface conditions

Define the nature of the interface of the domains is an important aspect of the coupled model. For both sub-problems, it is necessary to describe the desired behavior in this region. For the CHT problem, the following rules are applied:

- No slip condition

$$u = 0, \quad (2.4)$$

- Continuity of Temperature

$$T_f = T_s, \quad (2.5)$$

- Continuity of Flux

$$-\gamma \nabla T_s \cdot n = -\gamma \nabla T_f \cdot n \quad (2.6)$$

where T_f and T_s are the temperature at the fluid and solid and n is the normal vector.

These conditions express that the velocity at the interface is 0, the temperature at the fluid is equal to the temperature at the solid and the final condition ensures that the rate of heat leaving one material is the same as the rate of heat entering the other material.

2.2 Initial conditions

The initial conditions define the state of the model at the beginning of the simulation ($t = 0$) and are crucial for capturing the behavior of the system. In both models, the initial conditions for velocity and pressure are set to zero, reflecting a stationary starting state. For the temperature, all domains start with the same initial value. but the actual value varies between the two cases.

These initial conditions, at $t = 0$ are expressed as:

$$\begin{aligned}\mathbf{u}(x,y) &= 0 \\ p(x,y) &= 0 \\ T(x,y) &= T_{\text{init}}\end{aligned}\tag{2.7}$$

T_{init} represents the specified initial temperature, and (x,y) denote the coordinates within the domain.

Chapter 3

Numerical Approximation

As mentioned, an analytical solution to a CHT problem is generally difficult or impossible to find. Therefore, numerical methods are used to approximate the solution. In this work, the method used is the Finite Element method which requires 2 key components: the weak formulation of the governing equations and their subsequent discretization in both time and space.

3.1 Weak formulation

The CHT problem is transformed into a weak formulation by introducing the so-called test function, which allows to approximate the solution. The weak form of the PDE is obtained by multiplying the Strong formulation of the PDEs (2.1)-(2.3) by a test function and integrating by parts separately the mass, momentum and energy conservation equations. This reduces the complexity of the problem but also provides a natural way to implement boundary conditions. The obtained weak formulations are the following.

For the Navier-Stokes equations:

$$\int_{\Omega} (\nabla \cdot \mathbf{u}) q d\Omega = 0, \quad (3.1)$$

$$\int_{\Omega} \rho_f \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) \cdot \mathbf{v} d\Omega + \int_{\Omega} \mu \nabla \mathbf{u} \cdot \nabla \mathbf{v} d\Omega - \int_{\Omega} (\nabla \cdot \mathbf{v}) p d\Omega = \int_{\Omega} \mathbf{f}_f \cdot \mathbf{v} d\Omega, \quad (3.2)$$

And for the heat transfer equation:

$$\int_{\Omega} \rho c_p \left(\frac{\partial T}{\partial t} + (\mathbf{u} \cdot \nabla) T \right) w d\Omega + \int_{\Omega} (\gamma \nabla T) \cdot \nabla w d\Omega = \int_{\Omega} f w d\Omega + \int_{\partial\Omega} (\gamma \nabla T \cdot \mathbf{n}) w ds \quad (3.3)$$

Here, $u(x, y)$, $p(x, y)$ and $T(x, y)$ are called trial functions and $v(x, y)$, $q(x, y)$ and $w(x, y)$ are called test functions for the velocity, pressure and temperature, respectively and f_f is the force

applied in the fluid (bouyancy).

3.2 Numerical discretization

The second requirement is the discretization in both time and space domains. We employ a combination of the Backward Differentiation Formula (BDF) for temporal discretization and the Finite Element Method (FEM) for spatial discretization.

3.3 Implicit time discretization

The time interval $[0, t_f]$ is partitioned into N sub-intervals of equal size $\Delta t = t_f/N$ and denote by $t_n = n\Delta t$, for $n = 0, \dots, N$ the discrete time instances. Moreover, we denote by u_n, p_n and T_n the approximations of u, p and T at time t_n , respectively. We approximate the time derivative of the velocity and the temperature as

$$\begin{aligned}\frac{\partial u}{\partial t} &\approx \frac{u_{n+1} - u_n}{\Delta t}, \\ \frac{\partial T}{\partial t} &\approx \frac{T_{n+1} - T_n}{\Delta t}\end{aligned}\tag{3.4}$$

The BDF approximation of the the conjugate heat transfer problem reads: given u_n, p_n, T_n , find $(u_{n+1}, p_{n+1}, T_{n+1})$ such that:

$$\int_{\Omega} \nabla \cdot \mathbf{u}_{n+1} q d\Omega = 0,\tag{3.5}$$

$$\begin{aligned}\int_{\Omega} \frac{\mathbf{u}_{n+1} - \mathbf{u}_n}{\Delta t} \mathbf{v} d\Omega + \int_{\Omega} (\mathbf{u}_{n+1} \cdot \nabla) \mathbf{u}_{n+1} \mathbf{v} d\Omega = & - \int_{\Omega} \frac{1}{\rho_f} \nabla \mathbf{p}_{n+1} \mathbf{v} d\Omega + \int_{\Omega} \nu \nabla \mathbf{u}_{n+1} \cdot \nabla \mathbf{v} d\Omega \\ & - \int_{\Omega} \alpha g (T_n - T_0) \mathbf{v} d\Omega,\end{aligned}\tag{3.6}$$

$$\int_{\Omega} \rho c_v \frac{T_{n+1} - T_n}{\Delta t} w d\Omega + \int_{\Omega} \rho c_v (\mathbf{u}_{n+1} \cdot \nabla) T_{n+1} w d\Omega = \int_{\Omega} (\gamma \nabla T_{n+1}) \cdot \nabla w d\Omega + \int_{\Omega} f w d\Omega\tag{3.7}$$

where $\nu = \frac{\mu}{\rho_f}$ represents the kinematic viscosity.

3.4 Spatial discretization

The standard Galerkin finite element method with Taylor-Hood finite elements for velocity u and pressure p and quadratic or linear finite elements for temperature T are used for the

discretization of the decoupled problems [13]. To linearize the nonlinear terms of the Galerkin formulation, a Newton iteration is used.

Chapter 4

Algorithm implementation under Fenics

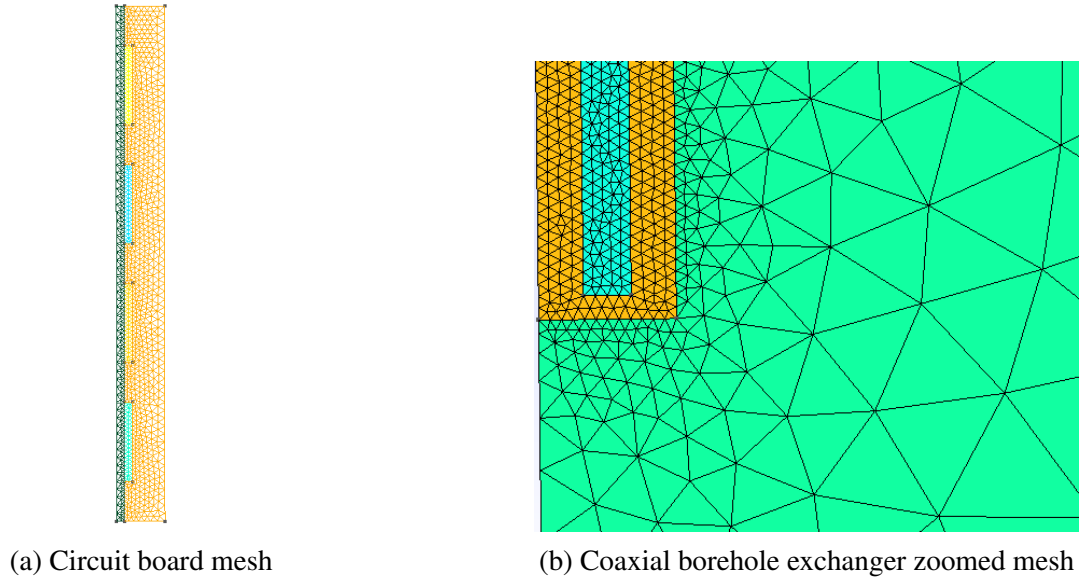
In this section is presented, step by step, the developed algorithm to solve the 2 CHT problems. The structure and logic of the solver is general for both CHT problems, however for each case, some specific modification were made to satisfy each problem domains, material properties and boundary conditions. The full code can be found in the github repository from appendix ??.

4.1 Mesh creation

The open source framework Gmsh [14] was used for constructing the domains and generate the mesh. The Gmsh features we used the most in this work are the ability to "mark" physical elements, useful to define all subdomains and boundaries for the heat transfer part of the problems and the liberty to apply finer or coarser meshes in specific regions of the mesh, used to apply a finer mesh in fluid domain and any critical region in order to achieve better approximations.

The resulted meshes were:

Figure 4.1: Meshes created in Gmsh



For the simulations we used a finer mesh in the fluid and in the interface and a coarser mesh in the solid part.

4.2 Function spaces

Function spaces for temperature, velocity, and pressure are defined. To satisfy the well-known inf-sup or Ladyzenskaja-Babuška-Brezzi (LBB) condition [15], a Taylor-Hood finite element spaces were selected, where the velocity has quadratic elements and the pressure has linear elements, this method is called mixed finite-element method. In the other hand, for the temperature a simple linear element space was used. Here is how it was declared in the code:

```
1 # Navier Stokes
2 V = VectorElement("CG", triangle, 2)
3 Q = FiniteElement("CG", triangle, 1)
4 W = FunctionSpace(air, V * Q)
5 # Heat transfer
6 Q1 = FunctionSpace(mesh, 'P', 1)
```

4.3 Boundary conditions

FEniCS, through the Dolfin library, supports various boundary conditions, including Dirichlet, Neumann, Robin, and periodic conditions. In this work, boundary conditions were applied to control the behavior of the fluid and temperature fields, utilizing Dirichlet and periodic conditions. Specifically, Dirichlet conditions were used to set the inflow velocity and temperature at the fluid inlet, as well as the pressure at the outlet. Dirichlet conditions were also used to enforce the no-slip condition on the fluid's walls, setting the velocity to zero. Periodic conditions were applied only in the circuit board problem to simulate the symmetry of temperature along the model's vertical walls.

How boundary conditions were applied depends on whether the boundary regions were predefined in Gmsh. If a boundary region was defined in Gmsh, creating the boundary condition in Dolfin was straightforward. It just needs to specify the boundary type and reference the Gmsh marked region, as in the next example:

```
1 DirichletBC(Q1, inlet_temperature, boundary_markers, 201)
```

For regions not defined in Gmsh, in our case for the Navier-Stokes problem, we created custom boundary classes in FEniCS. These classes specify the exact boundary areas, which are then marked accordingly. An example of this approach is shown below, where we define the fluid inflow boundary:

```
1 class inflow_fluid(SubDomain):
2     def inside(self, x, on_boundary):
3         return near(x[1], 0.) and on_boundary
4
5 inflow = inflow_fluid()
6 inflow.mark(boundary_air, 2)
7 DirichletBC(W.sub(0), Constant(inflow_profile), boundary_air, 2)
```

The boundaries are created by specifying the x and y coordinates of the boundary and the FEniCS function `near` is also used such that a tolerance of $1\text{E-}14$ is applied.

4.4 Variational problems

One of FEniCS biggest advantages is its intuitive handling of variational forms, allowing us to define the weak form of equations in a way that closely resembles the mathematical expressions of the problem. This feature is particularly useful in our setup, where we decompose the conjugate heat transfer (CHT) problem into two sub-problems: the Navier-Stokes equations for fluid flow and the heat transfer equations.

For the Navier-Stokes problem, the incompressibility condition (3.5) is implemented using the Lagrange Multiplier Method [16]. In this method, the weak form of the momentum (3.6) is augmented by the constraint (3.5), with the aid of Lagrange multiplier, leading to an augmented form of the momentum equation. The resulting equation is:

$$0 = \left(\frac{\langle \mathbf{u}_{n+1} - \mathbf{u}_n, \mathbf{v} \rangle}{\Delta t} + \nu \langle \nabla \mathbf{u}_{n+1}, \nabla \mathbf{v} \rangle + \langle \nabla \cdot \mathbf{u}_{n+1}, q \rangle \right. \\ \left. + \langle \frac{-1}{\rho_f} \nabla p_{n+1} + \nabla u_{n+1} \cdot u_{n+1}, \mathbf{v} \rangle + \langle \alpha \cdot \mathbf{g} \cdot (T - T_0), \mathbf{v} \rangle \right) dx \quad (4.1)$$

In code, this is expressed as follows:

```
1 Boussinesq = (alpha) * (T_old - T_init) * g
2 F_momentum = (inner(u - u_n, v)/k + nu*inner(grad(u), grad(v)) +
  ↪ inner((-1./rho_fluid)*grad(p) + grad(u)*u, v) + div(u)*q +
  ↪ Boussinesq * v[1])*dx_fluid
```

For the heat transfer problem, the weak form remains structurally similar to its initial form (3.7). Just small adjustments were made to account for the different materials. Here, the terms are grouped for solid and fluid regions, but depending in the problem it may have more or less regions:

$$(\rho c_v \frac{T_{n+1} - T_n}{\Delta t} + \Delta t f) w dx = (\gamma \langle \nabla T_{n+1}, \nabla w \rangle \Delta t + \rho c_v \langle \mathbf{u}_n, \nabla T_{n+1} \rangle w) dx \quad (4.2)$$

This form is implemented as separate bilinear and linear forms::

```
1 L = (rho_solid * c_p_solid * (T - T_old) * ww * dx_solid +
2     rho_fluid * c_p_fluid * (T - T_old) * ww * dx_fluid +
3     dt * heat_source * ww * dx_solid)
4 A = (dt * k_solid * dot(grad(T), grad(ww)) * dx_solid +
```

```
5 dt * k_fluid * dot(grad(T), grad(w)) * dx_fluid +  
6 dt * rho_fluid * c_p_fluid * inner(dot(u_n, grad(T)) , w) *  
  ↪ dx_fluid)
```

Finally the solvers used are nonlinear Newton solver for the Navier-Stokes equation and the default linear solver for the heat transfer problem.

4.5 Time-stepping loop

The main computational load of the algorithm is this section, where it solves iteratively the equations from the initial time t_{init} to the final time t_{final} . To measure the execution time of the simulation, the starting and ending times are recorded. For each time step, the Navier-Stokes equation is solved first and the heat transfer equation second. An early stopping condition is included, which terminates the loop if the changes in temperature, velocity, and pressure between successive iterations fall below a specified tolerance value. If this condition is not satisfied, the loop continues until t_{final} is reached. After each step, the current variable values are assigned as the previous values in preparation for the next iteration.

4.6 Output results

Upon completing the time-stepping loop, we obtain the final temperature, velocity, and pressure vector fields at the last time stamp. These vectors represent our output results and provide detailed information on the system's state at the end of the simulation. From these results, we created contour plots for temperature and velocity. Additionally, temperature and velocity were measured at specific critical areas within the model, offering insights into localized thermal and flow behaviors. These measurements are particularly valuable for evaluating the model's performance and offering practical feedback relevant to the physical problem being addressed.

Chapter 5

Applications

With all the mathematical background covered, in this sections, the specifics engineering problems used for this work are explained in detail. The two problems are the convection cooling of circuit boards [7] and the coaxial borehole heat exchanger [8]

5.1 Convection Cooling of Circuit boards

There is an increasing power density in modern circuits due to the development of more microminiaturized components. These compact designs mean that components are generating more heat in more limited spaces, risking overheating if not managed properly, which can lead to performance degradation, failure or reduced the components lifespan [17]. To maintain optimal operating temperatures, advanced cooling methods are being used, usually employing natural or forced convection cooling methods, utilizing either air or more thermally efficient liquids to remove heat [18]. In this work, a convection cooling method is used, in which the buoyancy forces caused by the temperature differences in the air, makes the warmer air to rise and cooler air to enter and cool the system.

The problem mentioned in [7] describe a steady state of a 3D natural convection cooling of a circuit boards as presented in figure 5.1. The boards are placed vertically to facilitate effective natural convection cooling and as the integrated circuits (ICs) generate heat, the surrounding air becomes warmer and rises. This upward movement of hot air allows cooler air to flow in, creating a continuous cycle that helps cool the boards and prevent overheating.

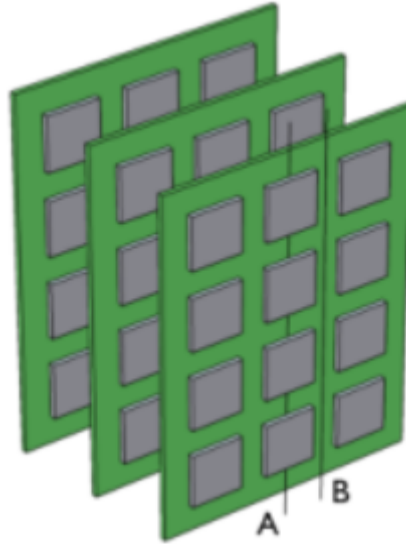


Figure 5.1: Circuit board 3D

In this work, the problem will be simplified into a 2D vertical view of one single board as show in figure 5.2. The dimentions and materials used are the following:

Component	Dimension (mm)	Material
Circuit Board	2 x 130	FR4
Integrated Circuit (IC)	2 x 20	Silicon
Ventilation Duct	10 x 130	Air

Table 5.1: Dimensions and materials of the circuit board assembly.

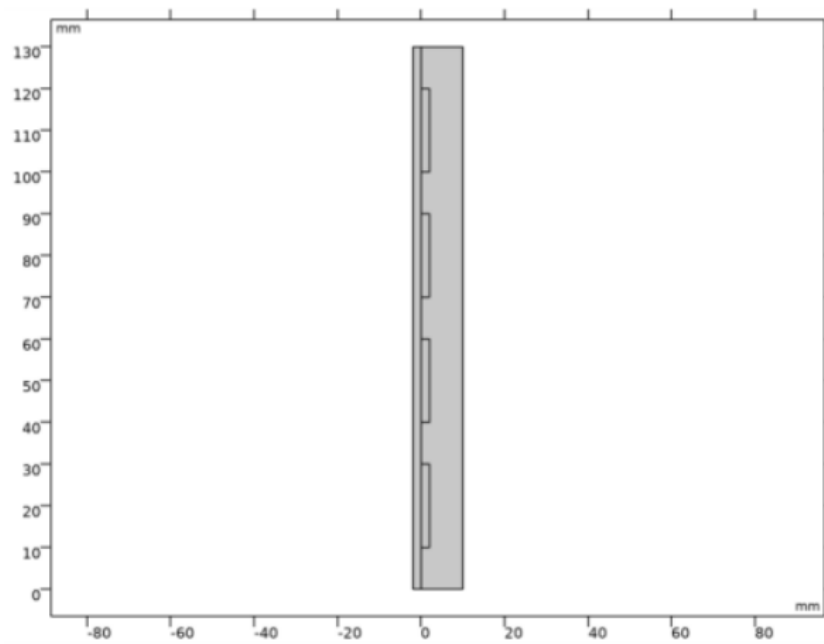


Figure 5.2: Circuit board 2D

The full domain is a rectangle of dimensions $10\text{mm} \cdot 130\text{mm}$, and it is divided in 3 subdomains, board, ICs and the ventilation duct as shown in 5.3.

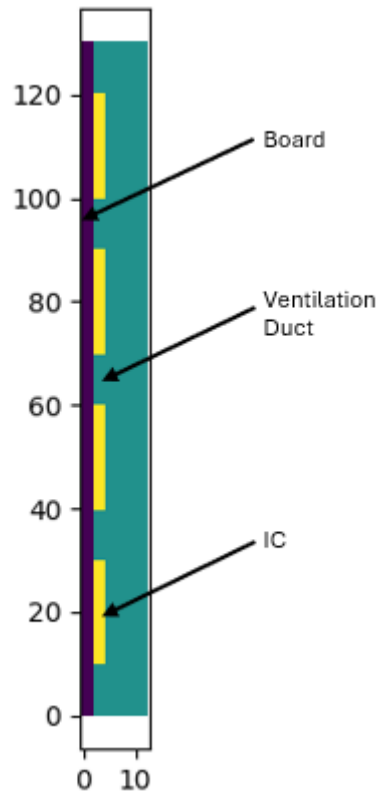


Figure 5.3: Circuit board subdomains

5.1.1 Materials properties

The material along with the external forces are the constant values utilized in the model. The properties of the materials are as follows:

Board Properties		
Property	Value	Unit
Density	1250	kg/m^3
Heat Capacity	1300	$\text{J/kg} \cdot \text{K}$
Thermal Conductivity	0.35	$\text{W/m} \cdot \text{K}$
IC Properties		
Density	2300	kg/m^3
Heat Capacity	710	$\text{J/kg} \cdot \text{K}$
Thermal Conductivity	150	$\text{W/m} \cdot \text{K}$
Air Properties		
Density	1.16	kg/m^3
Heat Capacity	1.0082	$\text{J/kg} \cdot \text{K}$
Thermal Conductivity	0.0282	$\text{W/m} \cdot \text{K}$
Dynamic Viscosity	$1.78 \cdot 10^{-5}$	$\text{kg/m} \cdot \text{s}$
Thermal expansion coefficient	0.0034	K

Table 5.2: Materials properties

And the external forces applied to the model are:

Property	Value	Unit
Gravitational acceleration	9.81	m/s^2
Heat source	$8.3333 \cdot 10^5$	W/m^3
Atmospheric pressure	$1.0133 \cdot 10^5$	Pa

Table 5.3: External forces

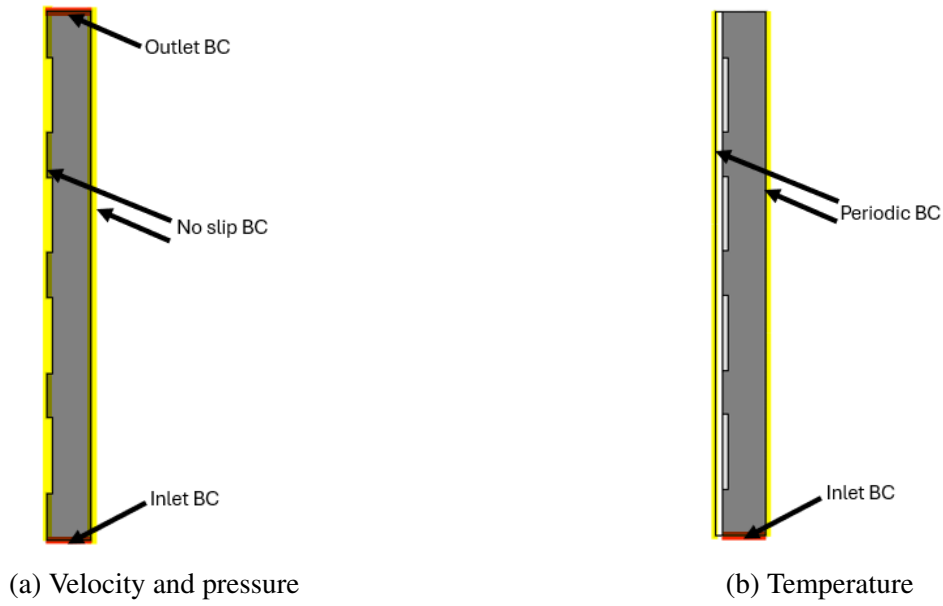
5.1.2 Boundary and initial conditions

Because of the nature of a conjugate problem, the boundary and initial conditions are divided into the boundary/initial conditions for the heat and for velocity and pressure.

The heat boundary conditions are: 300°K at the bottom part of the ventilation duct and every other boundary follows the rules of the Continuity of Temperature and Flux (2.5)-(2.6). The initial condition is: 300°K in all domains.

On the other hand, the velocity and pressure conditions are applied only within the ventilation duct domain. The boundary conditions are: no slip conditions (2.4) at the vertical borders of the duct (left and right borders), at the top border the pressure is 0.10133MPa (Atmospheric pressure from table 5.3) and the bottom border is treated as a free boundary to ensure natural convection. The initial conditions for both velocity and pressure throughout the entire ventilation duct domain are set to 0.

Figure 5.4: Boundary Conditions



5.2 Coaxial Borehole Heat Exchanger

The second problem, the Coaxial Borehole Heat Exchanger (CBHE), is a highly relevant technology for addressing high heating and cooling needs due to its efficient use of geothermal energy. As an example, in the European Union, nearly half of the energy consumed is used for heating and cooling, with a significant majority still dependent on fossil fuels. As governments aim to transition to greener energy sources, geothermal energy offers a stable, location-independent solution that minimizes emissions and meets both heating and cooling demands without requiring large surface areas, as it relies on the Earth's subsurface temperatures [8].

The CBHE is a system used to transfer heat between the ground and a circulating fluid, often water or a special mixture. It consists of two pipes, where the fluid flows down through one pipe and back up through the space between the two pipes, as shown in figure 5.5. As the fluid moves, it absorbs or releases heat to the ground, depending on whether the system is used for heating or cooling [8]. In this case, the problem that is going to be solved is a cooling system taken from Emilio Giménez Polo's work "Coaxial Borehole Heat Exchanger: Modeling, thermal and hydronic analysis".

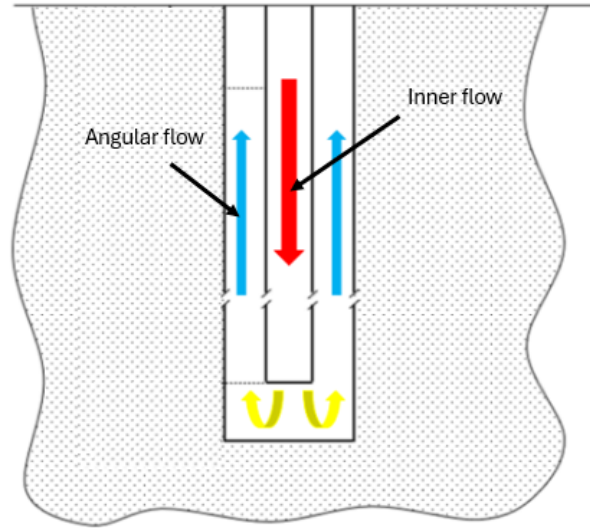


Figure 5.5: Coaxial borehole flow diagram

A CBHE can be modeled in two primary configurations: first with fluid flowing inward through the annular (outer) section and exiting through the central (inner) flow path, or second with fluid entering through the inner flow path and exiting via the annular flow. In our model, we adopt the second approach, where fluid flows into the system through the inner section and exits through the annular section, designed to cool the fluid as it circulates. To simplify the simulation, we applied vertical symmetry, simplifying the domain by cutting it in half along the inner flow path. The resulting domain is:

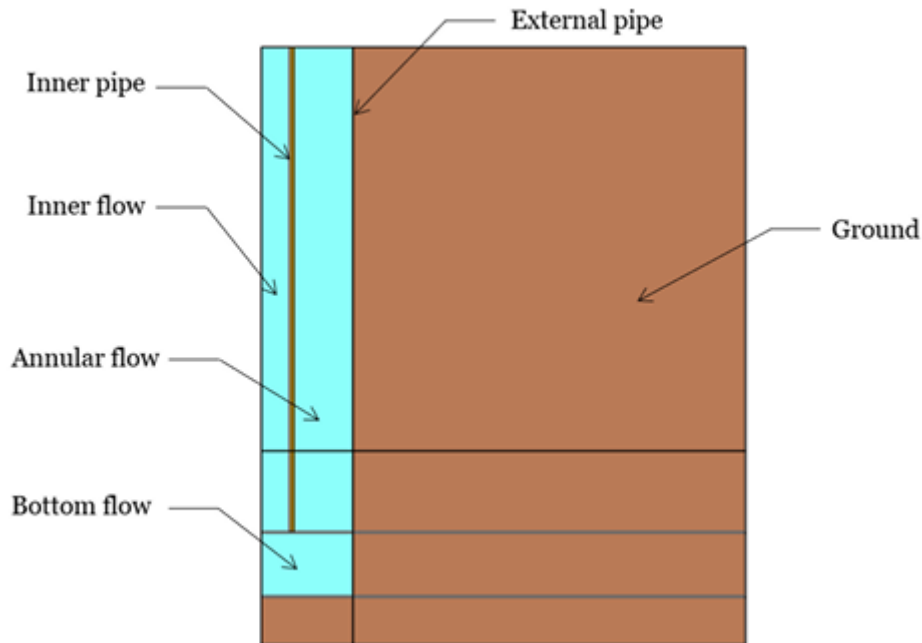


Figure 5.6: Coaxial Borehole Domain

The subdomains are the fluid, inner pipe and the ground. And the specific dimensions used are:

Component	Dimension
Inner pipe length	165m
Inner pipe diameter	150mm
Inner pipe thickness	40mm
Outer pipe length	165.04m
Outer pipe diameter	230mm
Ground length	170m
Ground width	5m

Table 5.4: Dimensions of the coaxial borehole assembly.

5.2.1 Material properties

Granite (Ground) properties		
Property	Value	Unit
Density	2915	Kg/m^3
Heat Capacity	$2240 \cdot 10^3$	$J/\text{Kg} \cdot K$
Thermal Conductivity	3.15	$W/m \cdot K$
MDPE (inner pipe) properties		
Density	938.5	Kg/m^3
Heat Capacity	2300	$J/\text{Kg} \cdot K$
Thermal Conductivity	0.40	$W/m \cdot K$
Water properties		
Density	999	Kg/m^3
Heat Capacity	$4.19 \cdot 10^6$	$J/m^3 \cdot K$
Thermal Conductivity	0.590	$W/m \cdot K$
Dynamic Viscosity	$1.138 \cdot 10^{-3}$	Pa
Thermal expansion coefficient	$207 \cdot 10^{-6}$	1/K

Table 5.5: Materials properties

For this work, the external pipe properties are ignored. We assume that the thickness of it is so small that is irrelevant for the simulation.

And the external forces applied to the model are:

Property	Value	Unit
Gravitational acceleration	9.810	m/s^2
Atmospheric pressure	$1.0133 \cdot 10^3$	Pa

Table 5.6: External forces

5.2.2 Boundary and initial conditions

The boundary conditions are define in the next image:

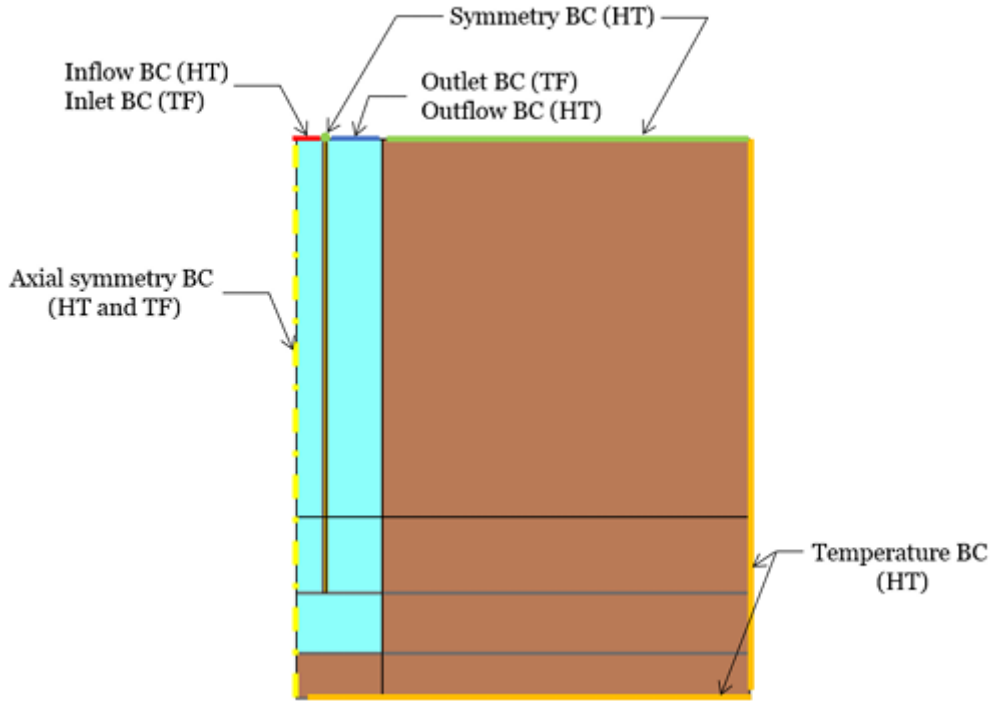


Figure 5.7: Coaxial Borehole Boundary Conditions

Where the values for the heat problem are: 16°C for the inflow temperature and 8.4°C for the Ground Temperature, all other boundaries are open boundaries. For the velocity and pressure the boundary conditions are: no slip conditions (2.4) at walls of the internal and external walls, besides the left wall of the model, which has a symmetry condition of $velocity = 0$ at the X axis only, the velocity at the inflow is 0.596L/s and at the outflow the pressure is 1.0133Pa (Atmospheric pressure from table 5.6). The initial conditions are 0 velocity and pressure in the whole domain, and the temperature is 8.4°C at the ground and inner pipe and 16°C at the fluid.

Chapter 6

Numerical Results

In this section, we present the numerical results, which include a comparison of plots and values from the literature with the results obtained from our simulations for the two selected examples. The solver ran in a google Collab notebook with a Python 3 environment without hardware acceleration, is important to notice these settings because metrics like execution time may change depending of the environment. This analysis will demonstrate the ability of the solver to be able to create accurate digital twins of CHT models and prove that FEniCS-based solver are able to replicate established results.

6.1 Numerical results: Convection Cooling of Circuit boards

For the natural convection cooling of circuit boards, the solver ran from $t_0 = 0$ to $t_{final} = 10$ seconds in 400 intervals, $dt = 0.025$ seconds, with an early stopping condition if the difference between the temperature, velocity and pressure in current and next step is lower than a tolerance of 10^{-5} . The mesh had 17218 triangular cells and 8762 vertices.

After running the solver with the material properties, initial and boundary conditions described in 5.1 and the initial and final time mentioned, the obtained results for the velocity and temperature at the final time are:

Figure 6.1: Surface temperature contours in K°

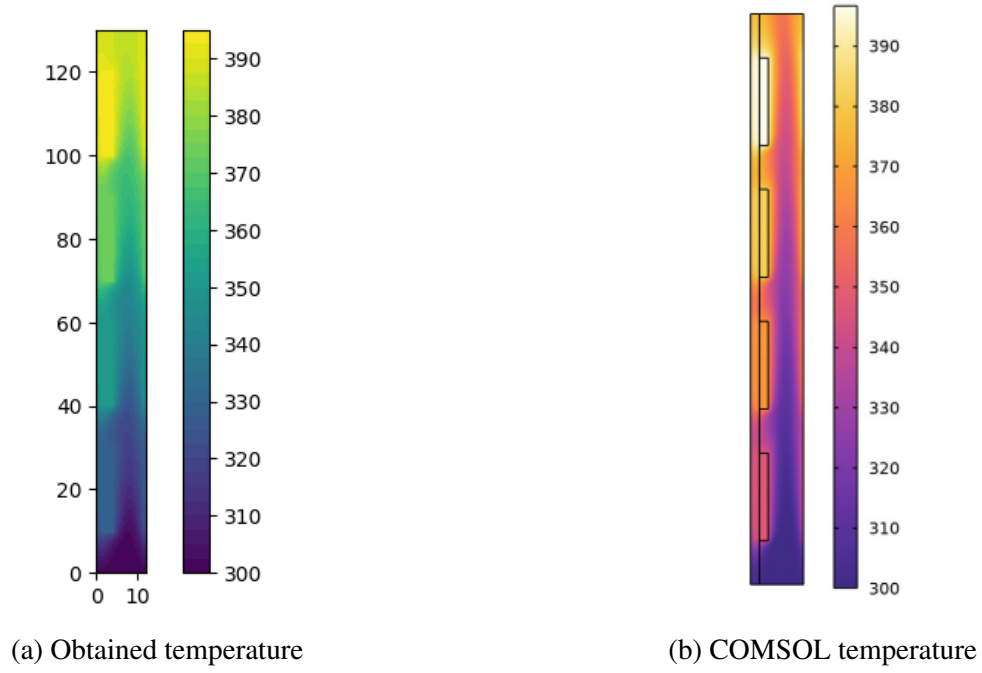
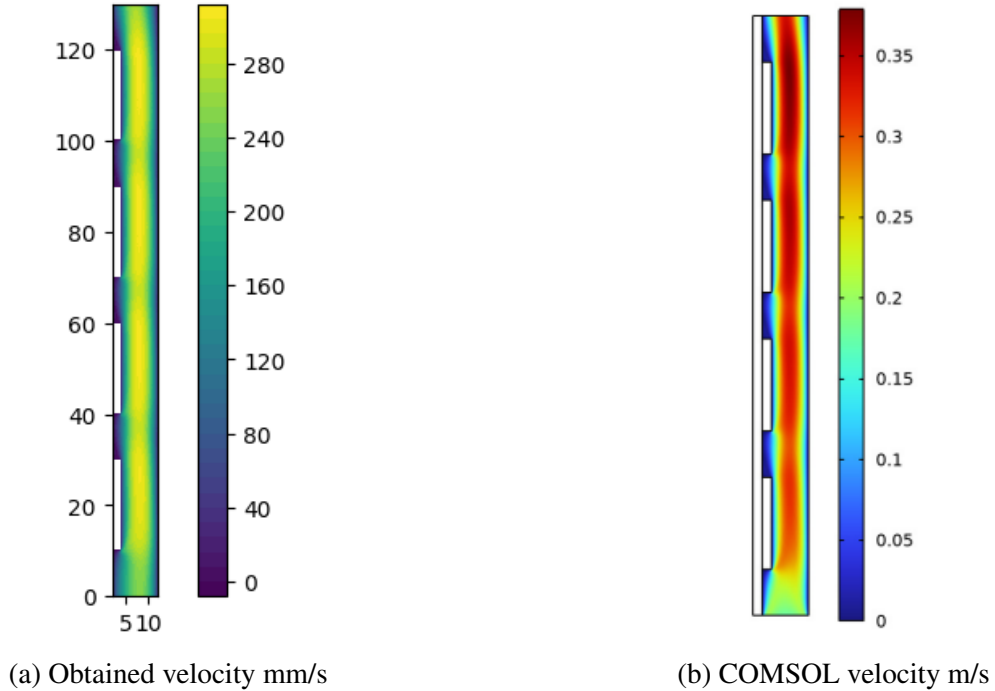


Figure 6.2: Surface velocity contours



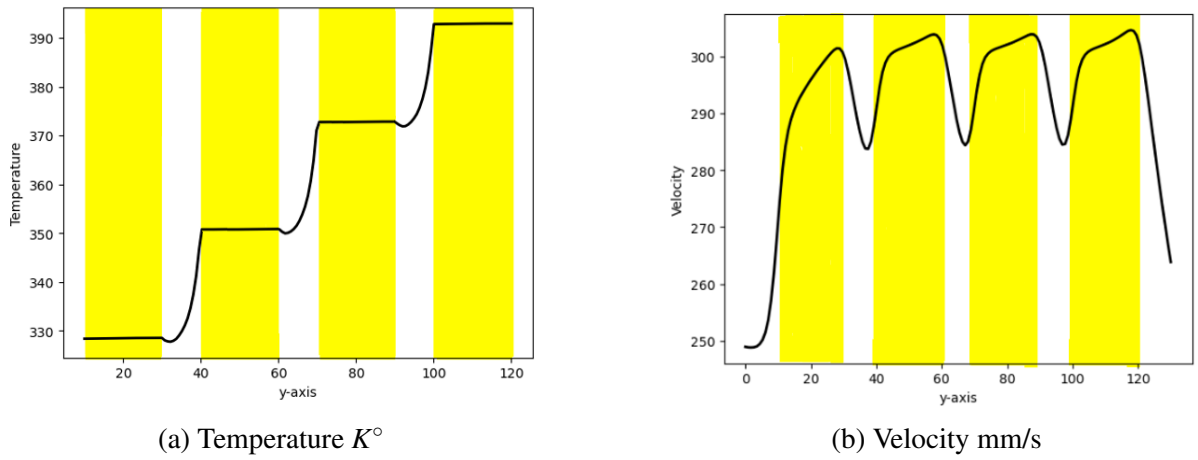
The early stopping condition did not get triggered because the velocity did not fit the criteria (Temperature and pressure did), which means that the results are at $t = 10$ seconds for a total execution of 17 minutes and 43 seconds. The obtained results show that for the temperature we

got a minimum temperature of 300°K at the bottom and around 395 °K. In the other hand, for the velocity we got a maximum speed of 280mm/s at the middle of the ventilation duct, close to the ICs.

Compared to the consol results, we can appreciate the similarity between the results. Specifically for the temperature. For the velocity even though there is a discrepancy with the maximum speed, the velocity contour plot shows a similar behavior.

The next figure gives a better perspective of the temperature specifically at the ICs and the velocity at the middle of the ventilation duct.

Figure 6.3: Temperature and velocity distributions on the y-axis



In these figures, the yellow areas represent the ICs. It is notorious that the velocity in each IC is higher due to the buoyancy effect and the higher the IC is placed, the higher the temperature.

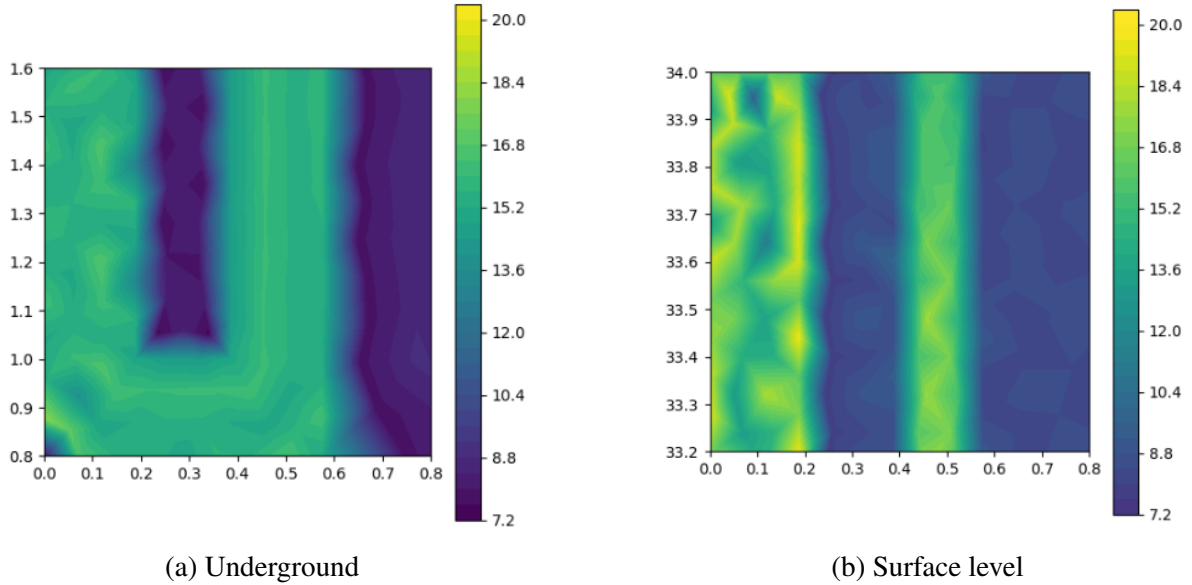
These results show that the hottest integrated circuit (IC) reaches a peak temperature of 395K, for a theoretical operational temperature for silicone-based electronic circuits is between 273.15K and 398.15K, it means that the temperature remains within the operational range. However, it is very close to the upper limit for the last IC. Consequently, any unexpected heat increase could easily push the temperature beyond safe operating limits, potentially compromising circuit reliability.

6.2 Numerical results: Coaxial Borehole Heat Exchanger

For the Coaxial Borehole Heat Exchanger, the solver ran 7000 iterations with a $dt = 0.1$ for a total execution time of 2 hours and 14 minutes. The mesh had a total of 25300 triangular cells and 12987 vertices.

After running the solver with the material properties, initial and boundary conditions described in 5.2 and the initial and final time mentioned, the obtained results for the temperature are the following:

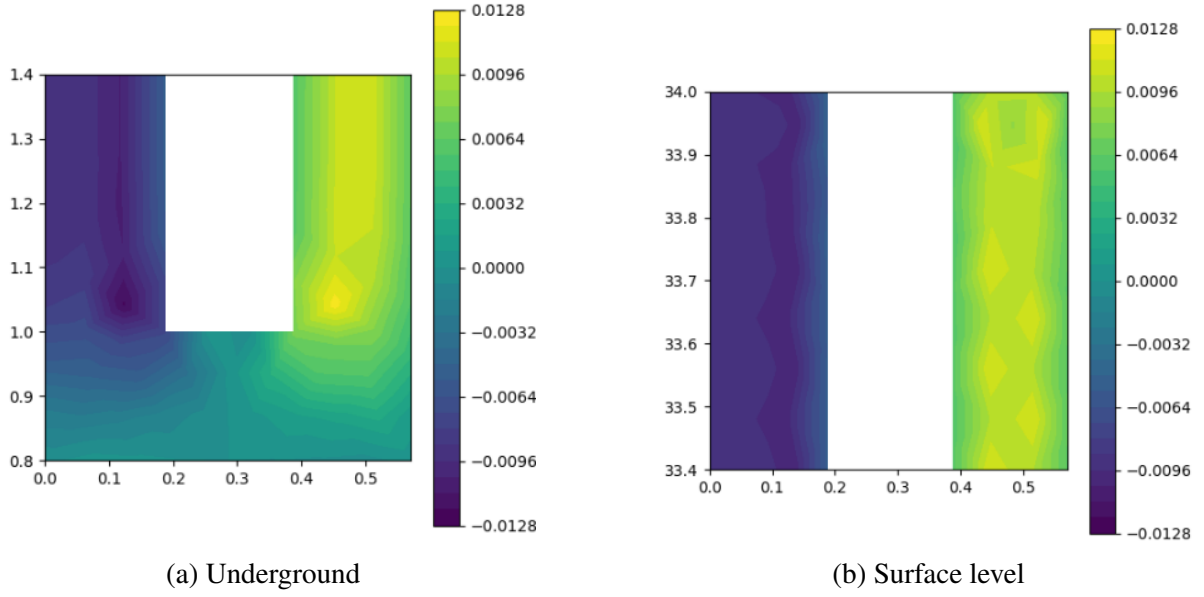
Figure 6.4: Temperature contours



Due to the large dimensions of the model, the plots displays the most important subsection of the domain, the underground area at the bottom of the pipe which is the area with the lowest temperature at approximately 13.6°C , and the surface where the inlet and outlet are located with a calculated temperature of 14.12°C at the outlet and 15.87°C at the inlet. In this model, the inner flow (left) carries the hot water entering the model, while the annular flow (right) carries the cooler water exiting the model. In the plot is visible this difference in the temperature between the two flows, proving the cooling process in the exchanger.

Additionally, the temperature and velocity profiles at the bottom of the pipes provide valuable insights, as illustrated in the following figure:

Figure 6.5: Velocity (m/s) contours along the Y axis



This area is crucial in the coaxial borehole model, as the flow transitions from the inner flow to the angular flow. The high pressure at the corners of the inner pipe can lead to minor temperature fluctuations; therefore, a fine mesh was implemented in this region to mitigate these fluctuations.

The values used to assess the accuracy and validity of our model are based on the results from Emilio Giménez Polo's thesis [8]. In his thesis, Giménez Polo conducted a series of simulations in COMSOL and MATLAB to evaluate heat exchange in various materials. For our comparison, we specifically use his simulations that employed the same materials and mesh size as detailed in Section 5.2.

The next table presents a comparison between the obtained numerical results and Giménez Polo's results.

Simulation	Inlet Temperature	Outlet Temperature
Obtained results by FEniCS	16	14.1
Matlab results from 5.2	16	14
Comsol results from 5.2	16	13.5

Table 6.1: Comparison between obtained results and Giménez Polo's results

These findings demonstrate the consistency of our results with previously established outcomes, confirming that the solver effectively manages critical regions with high-pressure effects without compromising the final results.

Chapter 7

Conclusions

This work addressed the development of a solver for the conjugate heat transfer (CHT) problem using the Finite Element Method (FEM) and explored its applications in thermal engineering. We investigated the interaction between heat conduction in solids and convection in fluids, modeling essential engineering applications like circuit board cooling and coaxial borehole heat exchanger. Using FEniCS and Gmsh, we developed an algorithm based on a partitioned coupling approach that solves independently and interactively the flow and heat transfer subproblems over time.

Our results align closely with established literature, validating the model's accuracy and reliability. This similarity suggests that these findings could provide practical insights for real-world physical setups, informing design adjustments for improved thermal management.

Demonstrating the potential of FEniCS as a tool for creating digital twins of real-world engineering problems that also enables rapid adaptation to different engineering challenges, due to its flexibility by allowing easy modification of model parameters and formulation.

While execution time was relatively low, we worked only with two-dimensional simulations. For three-dimensional simulations computational cost can become very high. In this case, several optimization techniques can be used to reduce the calculation time. One approach, to accelerate the simulation, is to change the code to run in parallel. A more advanced solution could involve using the model to generate a large dataset of results across various initial and boundary conditions, which will be used to train an AI model capable of predicting outcomes without running full simulations.

Bibliography

- [1] ANSYS. *ANSYS Fluent - CFD Software* | ANSYS. 2016. URL: <http://www.ansys.com/products/fluids/ansys-fluent>.
- [2] COMSOL Multiphysics. “Introduction to COMSOL multiphysics®”. In: *COMSOL Multiphysics, Burlington, MA* 9 (1998), p. 2018.
- [3] Igor A. Baratta et al. *DOLFINx: the next generation FEniCS problem solving environment*. preprint. 2023. DOI: 10.5281/zenodo.10447666.
- [4] Aimad Koulali et al. “Numerical investigation of conjugate heat transfer in the presence of two fluid separated by a rigid wall”. In: *Heat Transfer* 49.8 (2020), pp. 4981–5001. URL: onlinelibrary.wiley.com/doi/10.1002/htj.21863.
- [5] Giacomo Pareschi et al. “Conjugate heat transfer with the entropic lattice Boltzmann method”. In: *Physical Review E* 94.1 (2016), p. 013305. URL: <https://journals.aps.org/pre/abstract/10.1103/PhysRevE.94.013305>.
- [6] Amir Nadem Leyli, Mojtaba Moatamedi, and Basem Alzahabi. “Multiphysics Study of Forced Convection Conjugate Heat Transfer (CHT) Problem”. In: (2019). URL: <https://munin.uit.no/handle/10037/16467>.
- [7] COMSOL. *Modeling a Heat Transfer Problem: Natural Convection in a 2D Circuit Board*. 2023. URL: https://doc.comsol.com/6.0/doc/com.comsol.help.models.heat.circuit_board_nat_2d/models.heat.circuit_board_nat_2d.pdf.
- [8] Emilio Giménez Polo. “Coaxial Borehole Heat Exchanger: Modeling, Thermal and Hydronic Analysis”. PhD thesis. Industriales, 2020. URL: <https://oa.upm.es/62825/>.
- [9] Jae Yong Lee, Seungchan Ko, and Youngjoon Hong. “Finite element operator network for solving parametric pdes”. In: *arXiv* (2023). URL: <https://arxiv.org/abs/2308.04690>.

- [10] Gouri Dhatt, Emmanuel Lefrançois, and Gilbert Touzot. *Finite element method*. John Wiley & Sons, 2012.
- [11] Charbel Farhat, Kristoffer G Van der Zee, and Philippe Geuzaine. “Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity”. In: *Computer methods in applied mechanics and engineering* 195.17-18 (2006), pp. 1973–2001. URL: <https://www.sciencedirect.com/science/article/pii/S0045782505001945>.
- [12] Oluwadamilare Imam-Lawal, Tom Verstraete, and Jens-Dominik Müller. “Discrete adjoint for coupled conjugate heat transfer”. In: *Structural and Multidisciplinary Optimization* 66.5 (2023), p. 107.
- [13] Kelly Piaras. *Chapter 2 The Finite Element Method: The (Galerkin) Finite Element Method 2.1 Approximate Solution and Nodal Values*. URL: https://pkel015.connect.amazon.auckland.ac.nz/SolidMechanicsBooks/FEM/One_Dimensional/02_FE_Method.pdf.
- [14] Geuzaine, Christophe and Remacle, Jean-Francois. *Gmsh*. Version 4.6.0. June 22, 2020. URL: <http://http://gmsh.info/>.
- [15] Abner J Salgado, Johnny Guzman, and Francisco-Javier Sayas. “A Note on the Ladyzenskaja-Babuska-Brezzi Condition”. In: *arXiv preprint arXiv:1203.1870* (2012).
- [16] Ted Belytschko et al. *Nonlinear finite elements for continua and structures*. John Wiley & Sons, 2014.
- [17] Abdullah Alrashidi. “Numerical study of conjugate heat transfer for cooling the circuit board”. In: *Journal of Electronics Cooling and Thermal Control* (2016). URL: <https://www.scirp.org/journal/paperinformation?paperid=70464>.
- [18] Sassan Abedi Arani. *Experimental and computational investigation of forced convection cooling of rectangular blocks in a duct*. University of Bath (United Kingdom), 1992.

List of Figures

4.1	Meshes created in Gmsh	12
5.1	Circuit board 3D	17
5.2	Circuit board 2D	17
5.3	Circuit board subdomains	18
5.4	Boundary Conditions	20
5.5	Coaxial borehole flow diagram	21
5.6	Coaxial Borehole Domain	21
5.7	Coaxial Borehole Boundary Conditions	23
6.1	Surface temperature contours in K°	25
6.2	Surface velocity contours	25
6.3	Temperature and velocity distributions on the y-axis	26
6.4	Temperature contours	27
6.5	Velocity (m/s) contours along the Y axis	28

List of Tables

1	Notation table for symbols used in the article	1
5.1	Dimensions and materials of the circuit board assembly.	17
5.2	Materials properties	19
5.3	External forces	19
5.4	Dimensions of the coaxial borehole assembly.	22
5.5	Materials properties	22
5.6	External forces	22
6.1	Comparison between obtained results and Giménez Polo's results	28

Appendix A

Own Results

A.1 Scientific question

- What is the impact of the conjugate heat transfer on the modeling and analysis of thermal engineering problems?
- How can the conjugate heat transfer be solved in fEniCS?
- How effective are open-source frameworks in addressing complex scientific and engineering challenges?

A.2 Research methods

- Comparative analysis of the obtained results against established results from previous studies.
- The results were obtained by iteratively solving the model in FEniCS until the difference between consecutive iterations approached zero.

A.3 Results

- Results obtained for the Convection Cooling of Circuit Boards can be observed at 6.1-6.2-6.3.
- Results obtained for the Coaxial Borehole Heat Exchanger can be observed at 6.4-6.5.

Appendix B

Use of Artificial Intelligence

In the preparation of this thesis, artificial intelligence was employed exclusively to paraphrase some sentences, to ensure that the text was clear and easy to read. This assistance was limited to language refinement, and all outputs were critically reviewed to maintain the academic integrity and originality of the thesis. ¹

¹OpenAI's ChatGPT version 4 was used to provide suggestions for paraphrasing text to enhance readability.

Appendix C

Full code

All the code developed for this article, including the mesh generation and the dolfin solvers, can be found in the GitHub repository: <https://github.com/BernoldAZ/CHT-solvers>