European Consortium for Mathematics in Industry

28th ECMI Modelling Week
Final Report

19.07.2015—26.07.2015
Lisboa, Portugal

TÉCNICO LISBOA

Departamento de Matemática
Universidade de Coimbra

Group 1

Modelling Drying in Paper Production

Björn Nilsson
Lund University, Faculty Of Engineering,
Lund, Sweden
Endtmayer Bernhard
Technisch-Naturwissenschaftliche Fakultät ,Johannes Kepler Univeristy,
Linz, Austria
João Reis
Instituto Superior Técnico (IST), University of Lisbon,
Lisbon, Portugal
Sonia Vivace
Universitá degli Studi di Milano, Facoltá di Matematica
Milan, Italy

Instructor: Gonçalo Pena
CMUC, University of Coimbra
Coimbra, Portugal

Abstract

In paper industry, paper sheets are dried through a chain of paper dryers. The sheets roll through spinning cans at a very high speed. The initial goal of this project was to understand the profile of the temperature on the surface of the paper. Due to the complexity of such a problem, we had to simplify it. Therefore, in this work, we aim at solve the heat equation for the temperature inside an industrial paper dryer. Since the temperature distribution depends on the velocity of the air, we first have to solve the Navier-Stokes for the velocity of the air. After having our models set up, we use the Finite Element Method to find an approximate solution for the temperature. Finally, we present a graph of the temperature profile on the bottom of a chain of 7 paper dryers as well as some further considerations on how the the model could be improved.

1.1 Introduction

Between July 19 and July 26 2015, the Department of Mathematics of the IST-UL and the Department of Mathematics of the University of Coimbra organised the ECMI-Modelling Week 2015. The aim of this week was to model a real world problem and simulate.

The present work is the final report of the project group entitled *Modelling drying in paper production.* This project was proposed by Professor Gonçalo Pena, CMUC, Universidade de Coimbra. In this project, we aimed at developing and simulating a partial differential equations model for the drying of paper soaked in water or resin inside a drying tunnel. In order to achieve this result, we divided this work in 4 main section: a model for the temperature inside each dryer, a description of the numerical method used to solve the given model, the numerical results obtained from the implementation of this method and finally the main conclusions that one can take from these results.

We start with section 1.2. Here, we derive a model that describes both the temperature of the air inside the chain of seven dryers, denoted by T, and the velocity of this same air, denoted by $\mathbf{u}$. The temperature T is modelled using the heat equation. However, the temperature depends on the velocity $\mathbf{u}$ thus, to solve the heat equation we first need to derive $\mathbf{u}$. This will be done using the *Navier-Stokes* equations. It is reasonable to assume that an industrial dryer is turned on along the day for 24/7, i.e., heated up continuously. In practical terms, this means that the Therefore, we assume the steady state. This is an important feature of our model. Moreover, we have also considered the density to be constant. These two assumptions simplify our model substantially, while keep it accurate. Finally, the derivation of the Navier-Stokes equations becomes easier to handle if derived for Newtonian fluids. Since we are interested in the velocity of the air, a Newtonian fluid, this is enough for our purposes. Lastly, a variational formulation of our problem will also be presented, as it is essential for the numerical method.

Since both the heat and Navier-Stokes equations cannot be solved explicitly, we have to find a numerical approximation for the solution. To do do we introduce the Finite Element Method (FEM) in section 1.3. Here we start by discretizing our domain, which is of course the dryer. Then, using the variational formulations of previous sections and interpolating T and $\mathbf{u}$ linearly, we end up with a system of linear equations for the vector β of the coefficients of the linear combination. Both the discretization, the interpolation and the computation of β are done by the program FreeFem++. The code used for this implementation can be found in the appendix.

In paper industry, a drying tunnel is composed of several paper dryers (or paper ovens) align in a chain and connected by small junctions. In this project we consider 7 dryers, that will be our domain Ω. The paper rolls over several dryer cans that spin, passing through the dryers at a very high speed. The temperature in each dryer is crucial in order to obtain a certain type of paper sheets. This temperature is set independently in each dryer by a heat source (named HS in the equation), positioned in the top center of each dryer and it is supposed to be higher in the middle ovens, decreasing as approaching the extremities of the chain. Each dryer is also built with isolated walls, so the inside air is independent of the outside. We also consider a fan placed on the top of each dryer in the same place as the heat source. Finally, we call $\Gamma \subseteq \Omega$ the two connections between the first and last junctions of the chain, and the outside air.

With the set up from above, in section 1.4 we find two figures. The first represents the temperature profile in the chain of dryers, while the second one the temperature profile on the bottom of this same chain.

Due to the complexity of the main goal, and the short amount of time the team had to work on it, it was not possible to reach a solution for the main problem. This was mainly because the model that describes the moisture of the paper along the dryers is a formidable problem to understand and model in three days. Given that we hadn't had any such model by the third day, we decided it would be worth to simplify the problem in order to understand the underlying physics, even if the obtained model was oversimplified. This is discussed on a conclusion part, in section 1.5 Due to the high velocity of the paper, we assume that the impact of the paper on the air is negligible.

All the students involved in this project would like to congratulate the outstanding week the ECMI, the Department of Mathematics of the IST-UL and the Department of Mathematics of the University of Coimbra has provided to us. Also, we would like to acknowledge Professor Gonçalo Pena for his guidance and support throughout all the project.

1.2 Modelling

Heat Equation

In paper industry, a drying tunnel is composed of several paper dryers (or paper ovens) align in a chain and connected by small junctions. In this project we consider 7 dryers. The paper rolls over several dryer cans that spin, passing through the dryers at a very high speed. The temperature in each dryer is crucial in order to obtain a certain type of paper sheets. This temperature is set independently in each dryer by an heat source, but preferable, it is higher in the middle ovens, decreasing as approaching the extremities of the chain. Each dryer is also built with isolated walls, so the inside air is independent of the outsides. We also consider a fan placed on the top of each dryer in the same place as the heat source. Furthermore we assumed that the temperature of the connections to the outside of the ovens to have the temperature outside (approximately 300 K). We know that the diffusion coefficient depends on the temperature but to simplify the problem we set it to the diffusion coefficient of air under standard conditions.

We consider the seven ovens connected to each other as our domain Ω. The heat source (HS) and the fan are placed at the top center of each oven. Moreover, to set up initial conditions to solve our problems, we define $\Gamma \subset \Omega$ as the two connections of the ovens to the outside.

The law of the temperature in the ovens is given by the heat equation:

$$\frac{\partial T}{\partial t} - \alpha \Delta T = -\nabla \cdot (Tv) \tag{1.1}$$

where the boundary conditions are,

$$\begin{cases} T = T_{HS} & \text{on the heat source (HS)} \\ -(\alpha \nabla T + Tv).n = 0 & \text{on the wall of the dryers} \\ T = T_{outside} & \text{on the connections to the outside}(\Gamma) \end{cases}$$

$$\begin{aligned} &T: \quad \Omega \to \mathbb{R} \quad \text{temperature of the air, K} \\ &v: \quad \Omega \to \mathbb{R}^2 \quad \text{velocity of the air}, \mathrm{m/s} \\ &\alpha: \quad \text{diffusion coefficient of air}, 1.9 \times 10^{-5} \mathrm{m^2/s} \\ &t: \quad \text{time,s} \end{aligned}$$

There are two different types of flows in our model: the convective flow and the diffusion flow. This two different flows are represented by two different terms in 1.1

Recall that the sheets of paper roll through a chain of 7 dryers, joint by small junctions. Due to the fan, there is a convective flow of heat in our oven. The convective flow is the inflow and outflow of heated particles of an arbitrary domain. This is represented by $-\nabla \cdot (Tv)$. Furthermore the temperature distribution changes with time. This is the diffusion flow, represented by the term $\alpha \Delta T$. Of course the T evolves in time, and the variation of temperature in time will be equal to the sum of both flows. Hence, we have 1.1.

This equation is deduced by a balance made over the heat. The law conversion of energy tells us there is no way that either energy is produced or it gets lost in closed domain. So if there is a change of energy in some domain then there has to go some energy flow outside. In our case we neglected all energy flow except the heat flux. The heat flux represents the flux of heat from a place with higher temperature to one with a lower one and it also depends on the velocity of the particles which leaves or enter our domain.

The temperature dynamics is controlled by two phenomena :

- Heat has to be harmonized and heat is directed where the temperature is the lowest. This transport term is given by the vector $j_{diffusion} = -k\nabla T$,with k as the specific thermal conductivity of our fluid(air),it corresponds to the Fourier Law.

- Temperature is moving through the movement of the air particles. It is given by the vector : $j_{convective} = C_v T v$.

Thus, the heat flux is given by: $j_Q = j_{diffusion} + j_{convective} = -k\nabla T + C_v T v$. Where C_v is the calorific capacity of the air,which is assumed to be constant. The heat flow is a continuous variable, and it's continuous at the interface between the ovens.
By using the conservation law inside the field for the heat Q, we get :

$$\frac{\partial Q}{\partial t} + \nabla \cdot (j_Q) = 0.$$

Where $Q = C_v T + Q_0$

By rewriting this expression in terms of the temperature, we get the expression (1.1:

$$0 = \frac{\partial (C_v T + Q_0)}{\partial t} + \nabla \cdot (-k\nabla T + C_v T v) = C_v \frac{\partial T}{\partial t} - k\Delta T + C_v \nabla \cdot (Tv)$$

Moreover, dividing by C_v and taking $\alpha = \frac{k}{C_v}$

$$0 = \frac{\partial T}{\partial t} - \alpha \Delta T + div(Tv)$$

Recall that in order to solve the heat equation for the velocity, one must first solve Navier-Stokes equations. To do so, we need to set the boundary conditions for the Navier-Stokes equation. Firstly, no particle can pass the walls of the dryers. Recall that, if a particle is creating friction on a steady object, than it won't move while is touching it; this is known as rubbing. Since the particles of the air are rubbing the walls of the dryer, the velocity **u** on the walls is set to zero. Moreover, we also assume the fan to be placed at the same point as the heat source, blowing the air inside with a constant velocity. Therefore the velocity on the heat source is set to be constant. Lastly, consider the very first and last oven of our chain and assume no change of velocity in the junctions to the outside.

In order to solve this system using the Finite Element Method, we need the weak formulation of our model is:

Find $T \in V$ which solves:

$$\begin{cases} \int_\Omega (-k\nabla T + T\mathbf{u}) \cdot \nabla w - \int_{HS\cup\Gamma} w(-k\nabla T + T\mathbf{u}) \cdot n = 0 \quad \forall w \in V_1 \\ \text{Dirichlet condition on HS} \\ \text{Dirichlet condition on } \Gamma \end{cases}$$

Where $V_1 = \{f \in V \mid f = T_s \text{ on HS and } f = T_\Gamma \text{ on } \Gamma\}$ i.e. the set of functions of V that satisfy the Dirichlet boundary conditions.

Navier-Stokes Equations

The Navier-Stokes' equations describes the behavior of a fluid. The air that is being simulated here has a very low viscosity, a physical property of the fluid which represent the internal resistance against flow. This might be a problem though even the slightest speed of a particle in the dryer will make the temperature distribution totally homogeneous.

For a Newtonian fluid, such as the air, the Navier-Stoke's Equations are

$$\begin{cases} \rho\frac{\partial \mathbf{u}}{\partial t} - \mu\Delta\mathbf{u} + \rho(\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p = \rho g \\ \nabla\cdot\mathbf{u} = 0 \end{cases}$$

where,

$$\begin{aligned} u: &\quad \Omega \to \mathbb{R}^2 \quad \text{velocity of the air, m/s} && (1.2)\\ p: &\quad \Omega \to \mathbb{R} \quad \text{pressure of the air,Pa} && (1.3)\\ \rho: &\quad \text{density of the air}, 1.225\text{kg/m}^3 && (1.4)\\ \mu: &\quad \text{kinematic viscosity}, 1.7 \cdot 10^{-5} Pa && (1.5)\\ t: &\quad \text{time,s} && (1.6)\\ g: &\quad \text{body force (gravity), m/s}^2 && (1.7)\\ & && (1.8) \end{aligned}$$

Using Newtons s law of momentum and the law of conservation of momentum, we obtain that the change of momentum $I(t)$ over time on an arbitrary domain $\omega(t)$ is equal to the external forces which means:

$$\frac{d}{dt} I(t) = F(\omega(t)) \,, \text{with } I(t) = \int_{\omega(t)} \rho(x,t) u(x,t) dx \tag{1.9}$$

where $I(t)$ is the momentum and $F(\omega(t))$ is the force in $\omega(t)$.

It is well know that forces are the sum of the distinct forces, the body forces, such as gravity, and surface forces, like rubbing, for instance. Indeed, taking σ as the Cauchy stress tensor,

$$F(\omega(t)) = F_b(\omega(t)) + F_s(\omega(t)) \tag{1.10}$$

where

$$F_b(\omega(t)) := \int_{\omega(t)} \rho(x,t) g(x,t) dx$$
$$F_s(\omega(t)) := \int_{\partial\omega(t)} \sigma(x,t).n ds_x$$

Integrating F_s by parts we obtain

$$\int_{\partial\omega(t)} \sigma(x,t).n ds_x = \int_{\omega(t)} \nabla \cdot \sigma(x,t) dx \tag{1.11}$$

By combining 1.9,1.10 and 1.11 we obtain

$$\frac{d}{dt} \int_{\omega(t)} \rho(x,t) u(x,t) dx = \int_{\omega(t)} \rho(x,t) g(x,t) + \nabla \cdot \sigma(x,t) dx$$

Recall Reynold's transport theorem. Using this result in the above equation yields

$$\int_{\omega(t)} \frac{\partial(\rho u)}{\partial t}(x,t) + \nabla \cdot (\rho u.u^T)(x,t) dx = \int_{\omega(t)} \rho(x,t) g(x,t) + \nabla \cdot \sigma(x,t) dx$$

Because this is true for any domain $\omega(t)$, thus

$$\frac{\partial(\rho u)}{\partial t} + \nabla \cdot (\rho u.u^T) = \rho g + \nabla \cdot \sigma \tag{1.12}$$

Now we look for a tensor σ that represents how a body is transformed when force is applied on it. For newtonian fluids holds that:

$$\sigma = -p.I + \tau\,, \text{ with } \tau = \lambda(\nabla \cdot u).I + 2\mu\epsilon \tag{1.13}$$

Here, τ is the shear stress tensor. In our case, this force is the pressure on the boundary. Notice also that we consider no additional forces. Also, air is a newtonian fluid, so this assumption is reasonable for your problem.

Moreover, take $\lambda = -\frac{2}{3}\mu$ and $\epsilon = \frac{1}{2}(\nabla u + \nabla u^T)$. Applying the divergence to 1.13 we obtain

$$\nabla \cdot \sigma = -\nabla p + \delta u \nabla \cdot \tau$$

Thereby,

$$\nabla \cdot \sigma = -\nabla p + \mu \Delta u + \frac{1}{3}\nabla(\nabla \cdot u)$$

Finally, plugging this equation in 1.12 we get

$$\frac{\partial(\rho u)}{\partial t} + \nabla \cdot (\rho u.u^T) = \rho g + +\mu \Delta u + \frac{1}{3}\nabla(\nabla \cdot u) - \nabla p \tag{1.14}$$

Now we use the law of conversion of mass

$$\nabla \cdot (\rho u) + \frac{\partial(\rho)}{\partial t} = 0 \tag{1.15}$$

Since the density is constant, from 1.15 we get

$$\nabla \cdot u = 0$$

Notice that $\rho \cdot \nabla \cdot (u \cdot u^T) = \rho(u \cdot \nabla)u$. Using this and 1.14 we derive the uncompromisable Navier-Stokes' Equations.

$$\rho \frac{\partial(u)}{\partial t} + \rho(u \cdot \nabla)u = \rho g + \mu \Delta u - \nabla p$$

and

$$\nabla \cdot u = 0$$

Similarly to the heat equation, in order to solve the this system numerically one need the vibrational form of the problem.

Find ($\boldsymbol{u}$, p)$\in V \times Q$ which solves:

$$\begin{cases} \dfrac{\mu}{\rho} \displaystyle\int_\Omega \nabla u : \nabla v \ + \int_\Omega (u \cdot \nabla u) \cdot v - \int_\Omega p \ \nabla \cdot v - \int_{\partial\Omega \setminus \Gamma} (\nabla u \ v) \cdot n = 0 \ \ \forall v \in V \\ -\int_\Omega q \ \nabla \cdot u = 0 \ \ \forall q \in Q \\ \text{Dirichlet condition on the Heat Sources and the wall} \\ \text{Neumann condition on the connections with the outside (first and last oven)} \end{cases}$$

1.3 Numerical Methods

Now we have to solve the problem of finding a function $f \in V = \mathbb{H}^1(\Omega)$ which solves the heat equation and satisfies the initial conditions. In general, we won't find an exact solution, so we have to approximate it. We chose to use the **Finite Element Method**.
The idea of this method is to split the domain Ω using a partition made up of polygons (in general triangles), to choose a "simple" space P of functions defined on T and to approximate the solution of the problem with a function whose restriction to every element of the partition is an element of P.
A Finite Element is a tern (T, P, Σ) where:

- T is the polygon chosen for the partition T_h. In general the partition is chosen so that every element has a maximum diameter h;
- P is the n-dimensional function space defined on T with values in $\mathbb{R}$. In general, the space of polynomials of degree k is chosen;
- Σ is the set of degree of freedom, i.e. the set of linear functions $\Sigma_i : P \to \mathbb{R}, \ \ \forall i = 1, ..., n$, that satisfy:

  $$\forall p \in P, \ \forall i \in \{1, 2, ..., n\} \ \ \exists! c_i \ \ s.t. \ \ \Sigma_i(p) = c_i$$

  and are useful to identify unequivocally a function in T.

Once we have chosen the finite element, we can construct a finite dimensional space of continuous functions defined on Ω with real values, whose restrictions to a polygon of the partition are functions of P.

$$S_h^{k,0} = \{f \in \mathcal{C}^0 \ \ s.t. \ \ f|_T \in P \ \ \forall T \in T_h\}$$

Finally, we define $V_h := \{f \in S_h^{k,0}\text{satisfying the initial conditions}\}$
Clearly $V_h \in V$, and has finite dimension, so we can look for an approximated solution $u_h \in V_h$ of the problem.

For our project we chose P as the space of polynomials of degree 1 for the temperature in the heat equation and for the pressure in the Navier-Stokes equation, and the polynomials of degree 2 for the velocity in the Navier-Stokes equation.
Regarding the variational formulation of our problems, we can rewrite them for the discrete cases, in particular:

For the heat equation, we look for a solution $\tilde{T} \in V_h$ of the heat problem which satisfies:

$$\begin{cases} \displaystyle\int_\Omega (-k\nabla T + T\mathbf{u}) \cdot \nabla w - \int_{HS\cup\Gamma} w(-k\nabla T + T\mathbf{u}) \cdot n = 0 \ \ \forall w \in V_h \\ \text{Dirichlet condition on HS} \\ \text{Dirichlet condition on } \Gamma \end{cases}$$

Thanks to the finite dimension of V_h, we can take a basis of the space $\{\phi_1, \phi_2, ..., \phi_n\}$, write T as a linear combination of ϕ_i and check if the first equation is satisfied $\forall \phi_j,\ j = \{1, 2, ..., n\}$. Using the linearity of the integral and the gradient, the first equation becomes:

$$\sum_{i=1}^{n} \beta_i \int_\Omega (-k\nabla\phi_i + \phi_i \mathbf{u}) \cdot \nabla\phi_j = 0 \ \ \forall \phi_j,\ j = \{1, 2, ..., n\}$$

Finally, defining a matrix $\Phi = [\Phi_{ij}], \ \ \Phi_{ij} = \int_\Omega (-k\nabla\phi_i + \phi_i \mathbf{u}) \cdot \nabla\phi_j$ it turns out that the problem can be seen as a problem of linear algebra:

$$\Phi \cdot \beta = 0,$$

and the solution we now look for is the vector β of the coefficients of the linear combination of ϕ_i which identifies out T.

For the Navier Stokes equation, we look for a solution $(\mathbf{u}, p) \in V_h \times Q_h$ which satisfies:

$$\begin{cases} \dfrac{\mu}{\rho} \displaystyle\int_\Omega \nabla u : \nabla v \ + \int_\Omega (u \cdot \nabla u) \cdot v - \int_\Omega p \ \nabla \cdot v - \int_{\partial\Omega \setminus \Gamma} (\nabla u \ v) \cdot n = 0 \ \ \forall v \in V_h \\ - \int_\Omega q \ \nabla \cdot u = 0 \ \ \forall q \in Q_h \\ \text{Dirichlet condition on the Heat Sources and the wall} \\ \text{Neumann condition on the connections with the outside (first and last oven)} \end{cases}$$

Since this equation is not linear, some iteratives methods should be used to implement it, and we're not going to examine them in depth in this work.

1.4 Numerical Results

The plot below illustrates the steady case temperature distribution of seven dryers connected with different temperature in each one of them. Notation: the white lines that are present in the picture are the trajectory path of the particles. It is also clear that there are two thermal bridges at the ends of the connected dryer and that the air is flowing in that direction. There is also some circulation of air at the top corners of each individual dryer.

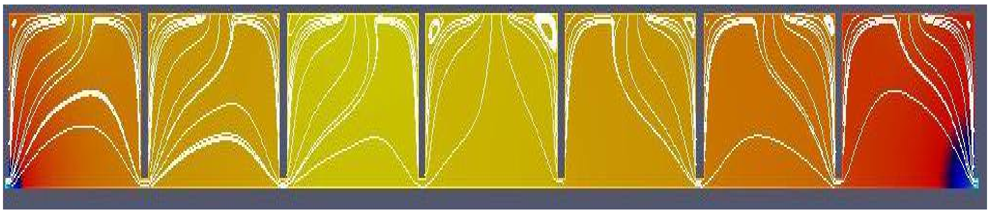

Figure 1.1: Temperature profile inside the chain of dryers. The wait lines represent the movement of a particle of air.

This second picture illustrates how the temperature profile at the bottom of the connected dryer looks like. Here the two thermal bridges mentioned above are also present, the two dips at the ends of the dryer represent this. The blue line represents the result of having a small fan incorporated in the dryer, to make the temperature distribution more homogeneous. The difference of the temperature distribution between the blue and the red line is very clear. Even though there are no walls to illustrate the different dryers, it is easily seen where the different dryers begin and ends.

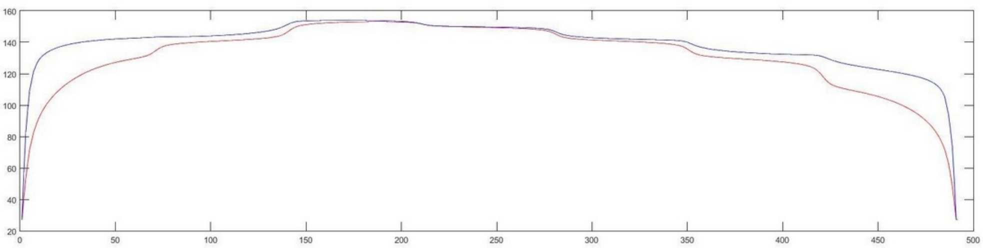

Figure 1.2: Temperature on the bottom of the chain of dryers.

1.5 Conclusions

Given the temperature of the dryers, the model simulates the profile of the velocity and temperature of the air. This is done in the steady case, meaning that there is no time dependency, and the temperature at the bottom of the dryers is extracted.

The model has no time dependency because it is interesting to investigate the distribution of the air in the steady case. There will not occur any paper drying in the oven when it is cold or during it is being heated up. That is why the model is in steady case or preheated mode as it can be expressed in a bit less formal way.

As seen in the model, the density and the viscosity is set to constant, also the gravity was neglected because of the Buoyancy. This is done because of simplicity. To get a more accurate model, the density should depend on the temperature of the air. The viscosity of the air is dependent on the temperature, though it is between: $(1.12 - 15.1) \cdot 10^{-4}\text{ft}^2/\text{s}$. So it does not affect that much and was therefore set as a constant for simplicity.

The Navier-Stokes equations were never used in the final results. Though there was a lot of time spent on this equation, it had to be a part of the paper. As explained above, both the density and the viscosity were put as constants. This meant that the compressible Navier-Stokes equations could not be used, which is a very good model to simulate fluid, but our problem gets much more complicated. So the incompressible Navier-Stokes equations were used instead. Even though the compressible Navier-Stokes were used instead of the incompressible Navier-Stokes, an approximate solution could not be calculated. A reason for this could be that there is no proof there exist a stationary solution, but even the time dependent problem was numerically unstable. Refining the mesh helped, but the numerical error was still dominant.

A very important part of the boundary conditions, on the Stokes equation, of a dryer is the air extraction of humid air which was neglected. The problem with the boundary condition was that it has to be a closed system. So the moisture had to go somewhere, and because it's a closed system, it could not just vanish.

This model's purpose was to model the temperature distribution in a dryer. Although this was completed, the intend of it, was to use it for a real problem. An attempt to model the drying process of a paper that went through the oven was made. The problem was that there were a lot of physics behind the actual drying process that was not fully understood. And that is why only the temperature profile of the dryers bottom and the temperature distribution of the air was simulated and plotted.

Overall the model simulates the temperature distribution and the temperature profile quite good and the result seems plausible with the approximations in mind.

.1 FreeFem++ Implementation Code

```
load "iovtk"

int[int] fforder =[1,1];
int[int] fforder2=[1,1,1];

int Omega=1;
int Gamma=2;
int HS1=3;
int HS2=4;
int HS3=5;
int HS4=6;
int HS5=7;
int HS6=8;
int HS7=9;

int Time=5;
func NN=[N.x, N.y];
real $k=1.9*10^(-5.)$;
real dt=0.2;
real high=2;
real length=3.25;
real lengthheatsource=0.5;
real xcon=0.1;
real ycon=0.1;
int meshmultip=3;
real s=1.3;
real Miu=0.1861*10^(-4.);
real gamma= 1;
int M=32;
//func v=[0,00.*y/2*(x-length+xcon-s)*(x-length+xcon
-s-lengthheatsource)]*(x<length+xcon-s)*
(x>length+xcon-s-lengthheatsource)*(y<high);
real deltaA=(length+2*xcon)/M;
border Omega0(t=0,7*length+14*xcon){x=t ;y=0; label=Omega;};
border Omega00(t=0,ycon){x=7*length+14*xcon ;y=t; label=Gamma;};
border Omega71(t=0,xcon){x=7*length+14*xcon-t ;y=ycon; label=Omega;};
border Omega72(t=ycon,high){x=7*length+13*xcon ;y=t; label=Omega;};
border Omega73(t=0,s){x=7*length+13*xcon-t ;y=high; label=Omega;};
border Omega74(t=0,lengthheatsource)
{x=7*length+13*xcon-s-t ;y=high; label=HS7;};
border Omega75(t=0,length-s-lengthheatsource)
{x=7*length+13*xcon-s-lengthheatsource-t ;y=high; label=Omega;};
```

```
border Omega76(t=0,high-ycon){x=6*length+13*xcon ;
y=high-t; label=Omega;};

border Omega61(t=0,2*xcon){x=6*length+13*xcon-t ;y=ycon; label=Omega;};
border Omega62(t=ycon,high){x=6*length+11*xcon ;y=t; label=Omega;};
border Omega63(t=0,s){x=6*length+11*xcon-t ;y=high; label=Omega;};
border Omega64(t=0,lengthheatsource)
{x=6*length+11*xcon-s-t ;y=high; label=HS6;};
border Omega65(t=0,length-s-lengthheatsource)
{x=6*length+11*xcon-s-lengthheatsource-t ;
y=high; label=Omega;};
border Omega66(t=0,high-ycon)
{x=5*length+11*xcon ;y=high-t; label=Omega;};

border Omega51(t=0,2*xcon){x=5*length+11*xcon-t ;y=ycon; label=Omega;};
border Omega52(t=ycon,high){x=5*length+9*xcon ;y=t; label=Omega;};
border Omega53(t=0,s){x=5*length+9*xcon-t ;y=high; label=Omega;};
border Omega54(t=0,lengthheatsource)
{x=5*length+9*xcon-s-t ;y=high; label=HS5;};
border Omega55(t=0,length-s-lengthheatsource)
{x=5*length+9*xcon-s-lengthheatsource-t ;
y=high; label=Omega;};
border Omega56(t=0,high-ycon){x=4*length+9*xcon ;y=high-t; label=Omega;};

border Omega41(t=0,2*xcon){x=4*length+9*xcon-t ;y=ycon; label=Omega;};
border Omega42(t=ycon,high){x=4*length+7*xcon ;y=t; label=Omega;};
border Omega43(t=0,s){x=4*length+7*xcon-t ;y=high; label=Omega;};
border Omega44(t=0,lengthheatsource){x=4*length+7*xcon-s-t ;y=high; label=HS4;};
border Omega45(t=0,length-s-lengthheatsource)
{x=4*length+7*xcon-s-lengthheatsource-t ;
y=high; label=Omega;};
border Omega46(t=0,high-ycon){x=3*length+7*xcon ;y=high-t; label=Omega;};

border Omega31(t=0,2*xcon){x=3*length+7*xcon-t ;y=ycon; label=Omega;};
border Omega32(t=ycon,high){x=3*length+5*xcon ;y=t; label=Omega;};
border Omega33(t=0,s){x=3*length+5*xcon-t ;y=high; label=Omega;};
border Omega34(t=0,lengthheatsource){x=3*length+5*xcon-s-t ;y=high; label=HS3;};
border Omega35(t=0,length-s-lengthheatsource)
{x=3*length+5*xcon-s-lengthheatsource-t ;
y=high; label=Omega;};
border Omega36(t=0,high-ycon){x=2*length+5*xcon ;y=high-t; label=Omega;};

border Omega21(t=0,2*xcon){x=2*length+5*xcon-t ;y=ycon; label=Omega;};
border Omega22(t=ycon,high){x=2*length+3*xcon ;y=t; label=Omega;};
```

```
border Omega23(t=0,s){x=2*length+3*xcon-t ;y=high; label=Omega;};
border Omega24(t=0,lengthheatsource){x=2*length+3*xcon-s-t ;
y=high; label=HS2;};
border Omega25(t=0,length-s-lengthheatsource)
{x=2*length+3*xcon-s-lengthheatsource-t ;y=high; label=Omega;};
border Omega26(t=0,high-ycon){x=1*length+3*xcon ;y=high-t; label=Omega;};

border Omega11(t=0,2*xcon){x=1*length+3*xcon-t ;y=ycon; label=Omega;};
border Omega12(t=ycon,high){x=1*length+1*xcon ;y=t; label=Omega;};
border Omega13(t=0,s){x=1*length+1*xcon-t ;y=high; label=Omega;};
border Omega14(t=0,lengthheatsource)
{x=1*length+1*xcon-s-t ;y=high; label=HS1;};
border Omega15(t=0,length-s-lengthheatsource)
{x=1*length+1*xcon-s-lengthheatsource-t ;y=high; label=Omega;};
border Omega16(t=0,high-ycon){x=1*xcon ;y=high-t; label=Omega;};

border Omega01(t=0,xcon){x=xcon-t ;y=ycon; label=Omega;};
border Omega000(t=0,ycon){x=0 ;y=ycon-t; label=Gamma;};
//border HSource(t=0,0.5){x=1-t;y=high; label=Gamma1;};
mesh Th = buildmesh(Omega0(M*meshmultip)+Omega00(2*meshmultip)+
Omega71(2*meshmultip)+Omega72(15*meshmultip)+
Omega73(6*meshmultip)+Omega74(5*meshmultip)+Omega75(5*meshmultip)+
Omega76(8*meshmultip)+
Omega61(2*meshmultip)+Omega62(15*meshmultip)+
Omega63(6*meshmultip)+Omega64(5*meshmultip)+Omega65(5*meshmultip)+
Omega66(8*meshmultip)+
Omega51(2*meshmultip)+Omega52(15*meshmultip)+
Omega53(6*meshmultip)+Omega54(5*meshmultip)+Omega55(5*meshmultip)+
Omega56(8*meshmultip)+
Omega41(2*meshmultip)+Omega42(15*meshmultip)+
Omega43(6*meshmultip)+Omega44(5*meshmultip)+Omega45(5*meshmultip)+
Omega46(8*meshmultip)+
Omega31(2*meshmultip)+Omega32(15*meshmultip)+
Omega33(6*meshmultip)+Omega34(5*meshmultip)+Omega35(5*meshmultip)+
Omega36(8*meshmultip)+
Omega21(2*meshmultip)+Omega22(15*meshmultip)+
Omega23(6*meshmultip)+Omega24(5*meshmultip)+Omega25(5*meshmultip)+
Omega26(8*meshmultip)+
Omega11(2*meshmultip)+Omega12(15*meshmultip)+
Omega13(6*meshmultip)+Omega14(5*meshmultip)+Omega15(5*meshmultip)+
Omega16(8*meshmultip)+
Omega01(2*meshmultip)+Omega000(2*meshmultip));
fespace Vh(Th,P1);
fespace Vh2(Th,P2);
```

```
Vh T, phi ,p,phi3;
Vh2 v1,v2,phi21,phi22;
func v=[v1,v2];
func phi2=[phi21,phi22];
macro Grad(u) [dx(u),dy(u)] //
macro div(u,v) (dx(u)+dy(v))//
macro gradgrad(u1,u2,v1,v2) (dx(u1)*dx(v1)+dy(u1)*dy(v1)+
dx(u2)*dx(v2)+dy(u2)*dy(v2))//

solve Stokes(v1,v2,p,phi21,phi22,phi3,solver=UMFPACK)=
int2d(Th)(Miu*gradgrad(v1,v2,phi21,phi22)-p*div(phi21,phi22)-
                       phi3*div(v1,v2))+
//int1d(Th,Omega1,Omega3,Omega4,Omega51,Omega52,Omega53,Omega6,Omega7)
                       (Miu*(Grad(v1)*phi21+Grad(v2)*phi22)'*NN-
                       p*phi21'*N.x-p*phi22'*N.y)+
on(HS1,v2=-0.000001,v1=0.)+
on(HS2,v2=-0.000001,v1=0.)+
on(HS3,v2=-0.000001,v1=0.)+
on(HS4,v2=-0.000001,v1=0.)+
on(HS5,v2=-0.000001,v1=0.)+
on(HS6,v2=-0.000001,v1=0.)+
on(HS7,v2=-0.000001,v1=0.)+
on(Omega,v1=0.,v2=0.);
solve thermic1(T,phi,solver=UMFPACK)=
 int2d(Th)((k*Grad(T)'*Grad(phi))-T*v'*Grad(phi))
 -int1d(Th, Gamma)(k*phi*Grad(T)'*NN)
 +int1d(Th, Gamma)(phi*T*v'*NN)
 + intalledges(Th)( gamma*(lenEdge^2)*abs(v'*NN)*
                        ( jump(dx(T))*jump(dx(phi))+
                        jump(dy(T))*jump(dy(phi))))
 +on(HS1, T=418.15)
 +on(HS2, T=414.15)
 +on(HS3, T=428.15)
   +on(HS4, T=423.15)
 +on(HS5, T=414.15)
 +on(HS6, T=403.15)
 +on(HS7, T=393.15)
 +on(Gamma, T=300.15);
savevtk("temperature.vtk",Th,T,dataname="Temperature",order=fforder);

plot(T,fill=120,value=1,nbiso=80);
savevtk("speedx.vtk", Th, v1, v2, order=fforder2, dataname="vx vy", bin=true );
```