Steel heat treating: industrial process, mathematical modelling, free software implementation and numerical simulation

Bruno Degli Esposti¹, Claudia García², José Miguel Herrera², José Ramón Obaya², Desy Paone³, Edoardo Prezioso³

¹Università Degli Studi Firenze
²Universidad Complutense de Madrid
³Università Degli Studi Di Napoli Federico II

Supervised by:
Francisco Ortegón Gallego
Departamento de Matemáticas
Universidad de Cádiz

June 2019
Contents

1 Introduction 3

2 Introduction to the Modeling Problem 5
  2.1 Introduction to Freefem++ ................................. 5
  2.2 Building a 2D-mesh ........................................... 5
  2.3 Elliptic Equation ............................................. 8
  2.4 Parabolic Equation ........................................... 9

3 3D Mesh creation 11
  3.1 Gear construction ........................................... 11
    3.1.1 The toothed surfaces .................................. 11
    3.1.2 The inner and the outer surfaces ....................... 13
  3.2 The 3D gear: straight and helicoidal ....................... 13
    3.2.1 An optional work: a conic gear ......................... 14
    3.2.2 Coding 3D mesh with FreeFem++ ....................... 15

4 Heating Problem 16
  4.1 Induction heating simulation .............................. 16
  4.2 Flame hardening simulation ............................... 19

5 Cooling problem 21

6 Conclusions 26

7 Future work 27
1 Introduction

Steel is an alloy of iron and carbon and it is used in industry to manufacture mechanical pieces that are going to be exposed to high stresses as they are in close contact in order to transmit the desired rotation/translation movement.

In order to avoid these stresses on the piece, a heat treating is made, making they lifetime longer. The steel heat treating produces a hard boundary layer to prevent abrasion and a soft inner part to reduce fatigue.

It is important to know some physical or mechanical properties of steel in order to understand why raw steel is a ductile material and why it can be hardened at critical parts changing its mechanical properties.

Steel can be found in nature in two different crystal lattices, face centered cubic (fcc) and body centered cubic (bcc).

As can be observed, the face centered cubic crystal lattice is denser than the body centered cubic and, therefore, it is more compact. There are different solid phases in steel:

- Austenite: is a solution of carbon in a face centered cubic iron. The only way to get austenite is having a carbon concentration lower than 2.11%; if so, only is possible at a high temperature range.
• Ferrite: nearly pure body centered cubic iron.
• Pearlite: lamellar structure of ferrite and cementite.
• Martensite: tetragonally body centered cubic iron crystal distorted by Carbon atoms. It can only stem from austenite.

The main difference between them is the cooling strategy. If the austenite is cooling down too fast, martensite is obtained. On the other hand, if the austenite is cooling down slow, perlite is obtained. It can be observed in the following diagram.
2 Introduction to the Modeling Problem

The mathematical description of the steel heating treating is given by a system of nonlinear PDEs in a 2D/3D setting subject to boundary/initial value problems governed by elliptic and parabolic partial differential equations, precisely. It recommended to have meticulous handling on Lax-Milgram’s theorem to have a uniqueness and existence of the solution for the elliptic PDE formulation; Sobolev spaces used to reduce the domain, where will be applied the boundary conditions; and transform the nonlinear problem into linear and resolve it with the finite element method with the Galerkin-Method. Moreover, implement the problem with numerical software and can to look at the numerical solution graphically in order to compare with the physical solution. We provide you with some free efficient packages software, namely Freefem++, MEdit and gmsh.

2.1 Introduction to Freefem++

Freefem++ is used for partial differential equation resolution with finite element method. Specifically, it is an integrated software solving PDE of elliptic type. By means some of semidiscretization in time scheme, Freefem++ may also deal with parabolic/elliptic evolution problems.

To solve the problem in question with Freefem++ it is possible follows that scheme: preprocessing (building a mesh in 2D/3D in order to realize the construction the domain to approximate the gear), find a variational formulation mathematically, and processing (in order to obtain and visualize the numerical solution). This software can be freely downloaded from the Internet (http://www.freefemm.org/ff++/index.htm) and it is fully documented.

2.2 Building a 2D-mesh

This section is devoted to the triangulation of domain. The domain considered Ω is a circumference with an inner ellipse Γ, where ∂Ω = Ω_D ∪ Ω_N with Ω_D and Ω_N Dirichlet and Neumann boundary condition respectively, from the PDEs equation that will be considered in that section. Freefem++ use the finite element method to build a mesh. We discretized the space in order to have finite dimension. The purpose is to divided the domain in triangles as a graph with nodes and edges and the nodes represent the discretized points of space.

For simplicity, we assume that Ω ⊂ \mathbb{R}^d is a polygonal, so that we are able to tile the domain with a set of triangles Δ_k, k=1,.....,K defining a triangulation T_h. This means that the vertices of neighboring triangles coincide and that:

- \bigcup_k \Delta_k = \overline{\Omega}
- \Delta_l \cap \Delta_m = \emptyset \text{ for } l \neq m
Surrounding any node is a patch of triangles that each have that node as a vertex (see Figure 2). If we label the nodes $j = 1, ..., n$, then for each $j$, we define a basis function $\phi_j$ that is nonzero only on that patch. $\phi_j$ is a linear function on each triangle, which takes the value one at the node point $j$ and zero at all other node points on the mesh.

Moreover, although $\phi_j$ has discontinuities in slope across element boundaries, it is smooth enough that $\phi_j \in H^1(\Omega)$. To make that triangulation with Freefem++, we need to use the functions: "border" (to build up the boundary) and "buildmesh" (to build a mesh with the contours used in "border").

Shown in the following figure:

```cpp
// First arc of the circumference
border Gamma1(t=pi-pi/4,pi+pi/4){x=cos(t);y=sin(t);label=1;}
// Remaining arc of the circumference
border Gamma2(t=pi+pi/4,3*pi-pi/4){x=cos(t);y=sin(t);label=2;}
// Inner ellipse
border Gamma3(t=0,2*pi){x=0.3*cos(t);y=0.2*sin(t);}
```

adding(mesh Th=buildmesh(Gamma1(10)+Gamma2(20)+Gamma3(40))) in the code, it is possible to build the triangles, where Gamma1, Gamma2 are the contours of the circumference with Dirichlet and Neumann boundary condition respectively, and Gamma3 for the boundary of the ellipse. Numbers (10), (20), (40) represent the number of the points that we have considered to discretize the domain.
Moreover, using the function "plot" we can obtain the numerical solution with Freefem++ (see Figure 3). Indeed it returns a graphical solution shown below. It appears a more dense discretization inside the ellipse than outside because in the code shown before we have considered a different numbers of points to take for Gamma1, Gamma2, Gamma3.

Figure 3: Approximation Numerical Solution with Freefem++ software.
2.3 Elliptic Equation

The Poisson equation

\[ -\Delta u = f \]  

is the simplest and the most famous elliptic partial differential equation. The source function \( f \) is given on some two- or three-dimensional domain denoted by \( \Omega \subset \mathbb{R}^d \) or \( \mathbb{R}^3 \). A solution \( u \) satisfying (1) will also satisfy boundary conditions on the boundary \( \partial\Omega \) of \( \Omega \); for example:

\[ u = g \quad \text{on} \quad \partial\Omega_D \]
\[ \frac{\partial u}{\partial n} = h \quad \text{on} \quad \partial\Omega_N \]

where \( \frac{\partial u}{\partial n} \) denotes the directional derivative in the direction normal to the boundary \( \partial\Omega \).

In order to write the elliptic equation in Freefem++ must be manipulated that equation applying Gauss theorem and integration by part to get the order reduced from second-order to first-order. The variational formulation will have this shape,

\[ a(u, v) = f(v) \quad \forall v \in V \]

This can be written in matrix form as the linear system of equations:

\[ Au = f \]

where \( A \) is a symmetric matrix and the equation is called Galerkin system.

Numerical form for Freefem++ is shown in (Figure 4), where the chosen functions are:

\( f = 3x + y, \quad g = x^2, \quad h = y^3; \)

```
problem a1(uh1,vh1) = int2d(Th)(alpha*uh1*vh1) +
int2d(Th)(beta*(dx(uh1)*dx(vh1) + dy(uh1)*dy(vh1)))-
int2d(Th)(f*vh1) - int1d(Th,Gamma2)(H*vh1)+on(uh1=g);

problem a2(uh2,vh2) = int2d(Th)(alpha*uh2*vh2) +
int2d(Th)(beta*(dx(uh2)*dx(vh2)+dy(uh2)*dy(vh2)))-
int2d(Th)(f*vh2)-int1d(Th,Gamma2)(H*vh2)+on(uh2=g);
```

Figure 4: Freefem++ Code.
2.4 Parabolic Equation

A possible solution of this problem with the Freefem++ software is represented in (Figure 5). Freefem++ use colors to underline the different level lines during the time.

![Figure 5: Numerical Simulation result of elliptic equation.](image)

2.4 Parabolic Equation

Parabolic Equation or Heat Equation

\[
\begin{align*}
\frac{\partial u}{\partial t} - k \Delta u + a \nabla u &= f \quad \text{in } \Omega \\
\quad u &= g \quad \text{on } \partial \Omega_D \\
\frac{\partial u}{\partial n} &= h \quad \text{on } \partial \Omega_N
\end{align*}
\]

\(u\) represent the temperature field in \(\Omega\) subject to the external heat source \(f\). Other important physical models include gravitation, electromagnetism, elasticity and inviscid fluid mechanics. Necessary is the time discretization in order to write the parabolic equation in Freefem++ language and to find a variational formulation. This time discretization is possible for the Crank-Nicolson Method.
A code in Freefem++ is written (shown below):

```cpp
problem heat(wh2,vh2) = int2d(Th)(wh2*vh2) +
int2d(Th)(deltat2*kh*(dx(wh2)*dx(vh2)+dy(wh2)*dy(vh2)))
+int2d(Th)(deltat2*(a1*dx(wh2)+a2*dy(wh2))*vh2)
-int2d(Th)(F*vh2) - int1d(Th,Gamma2)(H*vh2) + on(Gamma1,wh2=G);
```

In the numerical simulation obtained from Freefem++ (Figure 6) is clearly visible the heat diffusion with time values $t$.

On right-side is represented Neumann Condition and on the left-side Dirichlet. The Heat is concentrated more on the second one where there is the red color.

![Figure 6: Numerical simulation result of heat equation.](image)
3 3D Mesh creation

As it has been shown before, the illustrated method for the 2D mesh construction with FreeFem++ has been straightforward. It will be seen that the method for the 3D mesh construction results straightforward as well, allowing the construction of complicated figures, as long as the mathematical formulas for the generation of the boundary are provided. For a convenience in the exposure, a brief recap of the 3D mesh characteristics is shown.

A 3D mesh is composed by 3D polygons, but most works use in particular tetrahedra, which are the most simple geometric objects which can be built in 3D. For the creation of the 3D mesh, the mesh of the skin of the object is required. It is important to note that the skin is made by composing multiple surfaces, such that their sum matches the boundary of a compact and connected 3D set, i.e. it should not present ”holes”, nor jump discontinuities. The construction of the 3D mesh of the surface requires the creation of the 2D mesh of a suitable planar geometry, then a mathematical transformation to give shape and orientation in the 3D world is used. This is discussed for the mesh creation of the helicoidal gear.

3.1 Gear construction

In this project, the most simple model of the gear is constructed by creating four pieces: $S_u$, $S_d$, $S_i$ and $S_o$. Both $S_u$ and $S_d$ are the toothed annulus and are placed above and below the gear, respectively, while $S_i$ is the inner cylinder characterizing the ’hole’ and $S_o$ is the toothed outside surface.

3.1.1 The toothed surfaces

The construction of $S_u$ (or equivalently $S_d$) is made by an annulus with the external circle transformed to a truncated sinusoidal line. The equation describing its inner circle is the following:

\[
\begin{align*}
  x(t) &= \gamma R_e \cos(t) \\
  y(t) &= \gamma R_e \sin(t)
\end{align*}
\quad t \in [0, 2\pi]
\]

(1)

, with $R_e$ the radius of the outer circle, $\gamma$ the radius portion for the inner circle (which is, the inner radius is $\gamma R_e$ ). The outer curve is obtained by the composition of the following functions:

\[
F(t) = \cos N_T t \quad t \in [0, 2\pi]
\]

(2)

, which describes a parametric sinusoidal function with parameter $N_T$ (the number of the dents of the gear),

\[
R_C(t) = \begin{cases} 
F(t), & |F(t)| \leq C \\
-C, & F(t) \leq -C \\
C, & F(t) \geq C
\end{cases}
\]

(3)
3.1 Gear construction

which applies a truncation to a function $F$ in the interval between $-C$ and $C$, with $C$ the truncation ratio (it is effective for the sinusoidal function when $C \leq 1$). The final parametric curve equations, which combine the circle and the truncated sinusoidal function $F \circ R_C$ are as follows:

$$
\begin{align*}
  x(t) &= (R_e + H_TR_C(t)) \cos t \\
  y(t) &= (R_e + H_TR_C(t)) \sin t
\end{align*}
$$

where $H_T$ is half the height of the dent if the truncation is ignored.

The process of composition is illustrated in Fig. 7. These were generated by FreeFem++ with the following mesh parameters: npteeth is 700 and npinnerring is 60.

After the construction of the meshes for both $S_u$ and $S_d$, they are moved to the 3D world by placing $S_u$ and $S_d$ planar to $xy$ and placed at $z = T$ and $z = -T$, respectively.

![Figure 7: The generation of the toothed surface. The end result was generated with the parameters $\gamma = 0.7, R_e = 0.2, N_T = 20, H_T = \frac{R_e}{\pi}, C = 0.8$.](image)

![Figure 8: The four skin pieces of the gear. This was generated with $T = 0.04$.](image)
3.2 The 3D gear: straight and helicoidal

3.1.2 The inner and the outer surfaces

Both $S_i$ and $S_o$ are built from a rectangle $D = [0, 2\pi] \times [0, 2T]$. In the 3D world, $S_i$ is defined by the following transformation:

$$
\mathcal{T}_{S_i} : \begin{bmatrix} x \\ y \end{bmatrix} \in D \longrightarrow \begin{bmatrix} \gamma R_e \cos(x) \\ \gamma R_e \sin(x) \\ y \end{bmatrix}
$$

which defines a cylinder. For surface $S_o$, this is the transformation:

$$
\mathcal{T}_{S_o} : \begin{bmatrix} x \\ y \end{bmatrix} \in D \longrightarrow \begin{bmatrix} (R_e + H_T R_C(t)) \cos x \\ (R_e + H_T R_C(t)) \sin x \\ y \end{bmatrix}
$$

3.2 The 3D gear: straight and helicoidal

The four skin pieces of the gear are represented by Fig. 8.

These images were created with the tool Gmsh from the single meshes of the surfaces. The 3D mesh can be created from just the skin of the object by using the software library TetGen [6]. The result of the final merge and the 3D mesh construction is displayed in Fig. 9.

Figure 9: The straight gear.

The original problem was to solve the heat equation for a helicoidal gear. This is obtained by transforming a straight gear through this mathematical function:

$$
\mathcal{T}_{G_{hel}} : \begin{bmatrix} x \\ y \\ z \end{bmatrix} \in G \longrightarrow \begin{bmatrix} \cos \alpha z & \sin \alpha z & 0 \\ -\sin \alpha z & \cos \alpha z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x \cos \alpha z + y \sin \alpha z \\ -y \sin \alpha z + x \cos \alpha z \\ z \end{bmatrix}
$$

where $\alpha$ is the stretching factor. The result is shown with Fig. 10.

Figure 10: The helicoidal gear.

For the scope of this project, the factor $\alpha$ was set to 2.
3.2.1 An optional work: a conic gear

Since these transformations were straightforward, there was an attempt to transform the helicoidal gear into a conic helicoidal gear (like a bevel gear). An initial idea was by using this transformation:

$$\mathcal{T}_{G_{\text{conic}}} : \begin{bmatrix} x \\ y \\ z \end{bmatrix} \in G \rightarrow \begin{bmatrix} \varphi(z) & 0 & 0 \\ 0 & \varphi(z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x\varphi(z) \\ y\varphi(z) \\ z \end{bmatrix}$$  \hspace{1cm} (8)

where:

$$\varphi(z) = \frac{1}{2} \left( \frac{(1 - \lambda)z}{T} + \lambda + 1 \right);$$  \hspace{1cm} (9)

where \(\lambda\) is the ratio between the largest conic radius and the lowest conic radius. The result of this idea can be seen from Fig. 11, but one drawback was that the hole of the gear became conic, too, which is not desired for the modelling of the bevel gears.

Figure 11: The conic helicoidal gear. This was generated with \(\lambda = 0.5\) and, differently from Fig. 10, \(T = 0.15\).
3.2.2 Coding 3D mesh with FreeFem++

The implementation of all these steps with FreeFem++ are straightforward, as it can be seen from the following code snippets, containing the main steps of the 3D mesh construction.

```cpp
mesh3 Thbase = movemesh23(Th, transfo=[x,y,-thickness], label=labels);

Figure 12: Placement of the 2D mesh of the tooth into the 3D world at the base

mesh3 GearSkin = Thbase + Thtop + Thlaterali + Thlateralo;

Figure 13: Generation of the skin of the gear from the four surfaces

mesh3 ThGear3 = tetg(GearSkin, switch="paAAQY",
nbofregions=1, regionlist=domain);

Figure 14: 3D mesh construction from the skin of the gear
4 Heating Problem

In the last sections we have been explaining the basics of our problem, mainly all about the theoretical basics and the motivation and, afterwards, the construction process of the mesh which allows us to simulate now our problem.

Once we have built the mesh of the gear we can start solving the heating and cooling problems in order to obtain the approximated solution (by numerical simulation) of the transformation of Austenite into Martensite in the desired places of the workpiece, as we noted in the previous sections.

With this treatment of temperature variation we are able to increase the hardness of the steel on every tooth of the gear and, at the same time, maintain the rest of the workpiece soft and ductile in order to reduce fatigue. Prior to heat treating, steel is just soft and ductile material in the whole workpiece.

There are many industrial hardening procedures for gear depending on the size or the shape of the workpiece. We have two ways to heat the workpiece at a high temperature in order to obtain Austenite: magnetic induction and flame hardening.

Induction hardening can be applied in different manners. In our case we will simulate total induction around the workpiece by using a coil surrounding the gear, as we can see in the model of the mesh described in the previous chapter. During a time interval, a high frequency current passes through the coil generating a high alternating magnetic field, which induces eddy currents in the workpiece, which is placed close to the coil. The eddy current dissipate energy in the workpiece producing the necessary heating.

Moreover we can also obtain Austenite by heating the workpiece with direct flame. However, induction techniques have been successfully used more in industry since last century.

In this report we will not taking into account mechanical effects for the heating-cooling industrial procedure applied to a helical gear. In our numerical simulations we have considered very simple models for both explained methods above, without describing the deformation of the workpiece and having kept only dynamics of the Austenite and Martensite phase fractions.

4.1 Induction heating simulation

During the first stage, heat is produced by electromagnetics as we explain previously. Here, the main variables are the electric potential, the magnetic potential, the temperature and phase fractions corresponding to the austenite transformation.

First of all we are going to show the heating treatment model by induction. This problem can be simulated by the following PDE system:
As we can see above, the first term of the main equation corresponds to the variation of temperature \( u \) with time, where \( \rho \) and \( c_e \) are the density and the heat capacity of the steel. Then we have the diffusion term which depends on the steel diffusion coefficient \( k \). As source term we have now to introduce the Joule effect \( J \) produced by the magnetic field induced by the current which runs through the coil. Finally we need to introduce the latent heat \( L_1 \) caused by the absorption and emission of heat in the production of austenite. The austenite concentration \( a_t = f_{1a}(u, a) \) depends on a determined function provided by our instructor. And at the end, the Neumann boundary condition describe there is no heat diffusion with the environment.

Once we have develop the PDE model, now we can perform some numerical simulations of this model according to the industrial heating technique. To do so, we have used some efficient noncommercial software packages, namely FreeFem++, MEdit and gmsh.

FreeFem++ has been used for preprocessing (building the tetrahedralization of the helical gear as we saw in last section) and now for processing (to obtain the numerical approximation as we will see in this section). With the aim to visualize the graphic representations of the solutions in 3D we have used MEdit and gmsh package.

In order to figure out this PDE problem we have made a variational formulation using Crank-Nicholson scheme:

\[
\begin{align*}
\rho c_e u_t - \nabla \cdot [k \nabla u] &= \mathcal{J} - \rho L_1 a_t & \text{in } \Omega \times (0, T_c) \\
u|_{t=0} &= u_0 & \text{in } \Omega \\
a_t &= f_{1a}(u, a) & \text{in } \partial \Omega \times (0, T_c) \\
a(x, 0) &= 0 & \text{in } \Omega \\
k \frac{\partial a}{\partial n} &= 0 & \text{in } \partial \Omega \times (0, T_c)
\end{align*}
\]

This piece of code is an example of the reproduction of the variational formulation implemented in FreeFem++ software:

```plaintext
problem heatingup(w,v) = int3d(Th3corona) (rho0*c0*w*v + dtn*a0*(dx(w)*dx(v) + dy(w)*dy(v) +dz(w)*dz(v))) - int3d(Th3corona) (Source*v);
```

Afterwards, we only have to do a loop in order to obtain the temperature \( u \) for each step of time. In this way we can know the temperature distribution in all time range.
Therefore, now we can visualize the result that comes from solving these equations.

First, we can observe in the following graphics, after applying heat on the teeth of the gear, the difference of temperature between interior and exterior is considerably high. Our main goal was to produce austenite which is reached at determine high temperature, as we mentioned in the Introduction. We can see above the relation between zones with high temperature with zones with the higher austenite concentration, as we expected. Obviously we have more quantity in the exterior which is the place where we are more interested to become the workpiece harder.
4.2 Flame hardening simulation

As we explained before, another industrial technique used in the heat treatment of gears is flame hardening. This procedure can be used for both small and large gears.

The PDE model used in this simulations have been the following:

\[
\begin{align*}
\rho c_e \frac{\partial u}{\partial t} - \nabla \cdot [k \nabla u] &= - \rho L \frac{\partial a}{\partial t} \quad \text{in } \Omega \times (0, T_c) \\
\left. u \right|_{t=0} &= u_0 \quad \text{in } \Omega \\
u &= u_{\text{flame}} \quad \text{in } \partial \Omega \times (0, T_c) \\
k \frac{\partial u}{\partial n} &= 0 \quad \text{in } \partial \Omega \times (0, T_c) \\
\frac{\partial a}{\partial t} &= f_{1a}(a, u) \quad \text{in } \Omega \times (0, T_c)
\end{align*}
\]

This model is quite similar as the other one with induction, but now we should omit the Joule effect in the source since we do not have any magnetic field produced by any current. Remember that now we do not have any coil surrounding the gear, so we inject flame directly to the workpiece for heating it. On the other hand, in this case we have to add a new Dirichlet boundary condition which is given by the temperature of the external flame that heats the side wall of the teeth of the piece. The rest of the flame heating PDE model is the same.

So now, the variational formulation is pretty similar to the induction treatment. Now we only have to change the source term in the formulation since the main part of the PDE (left side) is the same:

\[
\int_{\Omega} w v + \frac{\Delta t}{2 \rho c_e} k \int_{\Omega} \nabla w \nabla v = -\rho L \frac{\Delta t}{2 \rho c_e} \int_{\Omega} (f_{1a} + u) v
\]

where

\[
w^{n+1} = \frac{u^{n+1} + u^n}{2}
\]

As we did in the last subsection, we have to implement this formulation in the FreeFem++ software we have been using auntill now in order to obtain the solution of the temperatures distribution in the workpiece and, related to that, the austenite concentration, which we are more interested at. The procedure is the same than before and here we show a piece of the code used in this example:

```plaintext
problem heat(wh,vh,solver=CG,eps=le-20) =
  int3d(Th3corona) (wh*vh) +
  int3d(Th3corona) (0.5*dtn*rhocel*aconductermica*
  (dx(wh)*dx(vh)+dy(wh)*dy(vh)+dz(wh)*dz(vh)))
  -int3d(Th3corona) (Source*vh)
  + on(78,wh=flamet);
```
In the same way, we can now also see the graphic representations of the 3D solutions obtained specifically with this heat treatment, provided by MEdit and gmsh packages:

![Figure 17: Austenite concentration at $t = 10\text{s}$](image1)

![Figure 18: Temperature at $t = 10\text{s}$](image2)

These figures show some interesting numerical results corresponding to a flame hardening simulation. A more uniform contour can be observed in this case for both the austenite and temperature profiles, which is a good agreement with experiment results.

Now, the distribution of the austenite is only in the profile of the teeth instead of being in the whole corona of the gear. Previously, with induction heating, austenite was obtained inside the teeth as well as in the profile. This is the main difference between both studied techniques and one the most important features of this study.

In the following section we will explain the production of Martensite by applying a rapid cooling procedure, such as aqua quenching, based on the knowledge of this first heat stage.
5 Cooling problem

After having studied the problem of heating up the gear and the results obtained, it must be cooled in order to obtain martensite in the teeth of the workpiece and also to make it stronger.

To carry out the cooling down process, two ways can be used mainly: diving into water or showering the workpiece. Each of these options causes the cooling down of the object at different speeds and therefore also generate a different amount of martensite.

In this work we will apply the process of diving into water to carry out the cooling of the workpiece. This process can be modeled from the following PDE problem.

\[
\begin{aligned}
\rho_c u_t - \nabla \cdot [k \nabla u] &= -\rho(L_1 a_t + L_2 m_t) & \text{in } \Omega \times (0, T_c) \\
\frac{\partial u}{\partial n} &= \alpha(u_{\text{ext}} - u) & \text{in } \partial \Omega \times (0, T_c) \\
u|_{t=T_c} &= u(T_c) & \text{in } \Omega \\
a|_{t=T_c} &= a(T_c) & \text{in } \Omega \\
m|_{t=T_c} &= m(T_c) & \text{in } \Omega \\
m|_{t=T_c} &= f_m(u, a, m) & \text{in } \Omega \times (O, T_c) \\
\end{aligned}
\]

This problem as can be verified is very similar to the problem used for the heating up of the gear. As main differences, we can highlight that new terms appear on the right side of the main equation, such as the latent heat linked to the formation of martensite as well as its generating function. On the other hand the generation function of the austenite appears modified because now also depends on the martensite that is been generated at each moment.

New boundary conditions appear, such as the Robin’s condition that describes the energy transfer between the gear and the environment. It also appear the martensite and austenite formation equations.

Below are the results of applying the cooling down process to a workpiece heated up by using a flame and after then the results of pre-heating the gear by induction.

As you will see the results are very similar although they present some slight differences.

In the case of heating up with flame, we can see in the following figures that the amount of austenite has disappeared almost completely although it remains in the interior holes of the workpiece but in a so small proportion. On the other hand, the temperature has been reduced significantly, although the piece remains quite hot inside, this is due to the contact between water and the gear, because it emits energy to the water in the form of heat. That situation makes the piece even hotter in the previous moments.
Finally, if we check the following images, corresponding to the formation of martensite, it can be verified that the initial austenite was transformed into martensite but only on the most superficial part of the gear’s tooth.
Now we go to see the results obtained during the cooling down of a gear heated up by induction. It can be seen that the results are quite similar to those of the previous case. The workpiece still having austenite inside it and continues remaining hot due to the process that has been discussed previously.

Figure 21: Martensite concentration at $t = 22s$

Figure 22: Austenite concentration at $t = 26s$
As a difference between this process and the previous one, we can say that we can see easily that martensite is not only present in the teeth’s border of the gear, it covers them completely.
Another way of inspecting the results of the simulation is by plotting the temperature of some relevant points inside the mesh as a function of time. By choosing four equally spaced points aligned inside a gear tooth, we obtain the diagram in figure 25. In this plot, point 1 (purple curve) is the innermost point, whereas point 4 (yellow curve) is the outermost point.

![Figure 25: Evolution of temperature at four points inside the gear](image)

This diagram suggests that an interesting physical phenomenon is taking place inside the gear: at the beginning of the cooling stage \( (t = 10 \text{ s}) \), the temperature of points 1 and 3 is counter-intuitively increasing, because austenite is releasing energy that had been stored in its crystal structure by the heating process.
6 Conclusions

In this work, we have used the finite element method to numerically solve simplified mathematical models describing the industrial process of flame and induction hardening of steel. The results of our numerical simulations are in line with existing research [4], match our physical intuition and agree with experimental data. Moreover, they are stable with respect to the choice of time step and mesh resolution: a smaller time step and a higher resolution do not alter the qualitative aspect of the solution.

In this work, we’ve also described a complete workflow for the simulation of industrial processes using only free and open source software, namely FreeFem++ [3] and Tetgen [6] for FEA and mesh generation, Gmsh [2], Medit [1], and Gnuplot for visualization. This workflow is summarized in figure 26, and can be adapted to different problems, not just steel heat treating. We think that this kind of approach has the following merits:

- It’s flexible, because FreeFem++, besides doing finite element analysis, is also a general-purpose programming language (a C++ idiom). Therefore, all aspects of a simulation can be customized to suit specific needs by writing appropriate code.

- Parts of the workflow can be scripted, so that a program can automatically run multiple instances of a simulation with different parameters or different geometry, then compare the results and keep the best one. This is a simple and effective, albeit crude, approach to the optimization of an industrial process.

- The users are in full control: they can choose the finite elements type, the linear algebra routines and the accuracy targets when the defaults are inadequate for the problem that has to be solved. Moreover, they can inspect the internal data structures used by FreeFem++ during the simulation and gain a deeper understanding of the underlying numerical methods, as we have experienced ourselves during this work.
7 Future work

The heat source term due to eddy currents in our model of induction heating uses precomputed coefficients for performance reasons. A more accurate simulation would describe the eddy currents using the vector potential formulation of Maxwell’s equations in the harmonic regime [5], and would therefore introduce the necessary geometry for the inductor and the surrounding air, as can be seen in figure 27. Such a complex simulation would require some form of parallel computation in order to be feasible. In this regard, FreeFem++ provides access to effective domain decomposition techniques through the ffddm framework.

<table>
<thead>
<tr>
<th>Description of the industrial process that we want to simulate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mathematical models of the physical phenomena involved (heat, electromagnetism, steel phase fraction evolution)</td>
</tr>
<tr>
<td>Definition of the domain of interest</td>
</tr>
<tr>
<td>Parametrization of the boundary describing the subdomains of this setting</td>
</tr>
<tr>
<td>Variational formulation using boundary conditions</td>
</tr>
<tr>
<td>Discretization and mesh generation</td>
</tr>
<tr>
<td>Computationally precise definition of the problem with FreeFem++</td>
</tr>
<tr>
<td>We discretize in time (e.g. Crank-Nicolson method) and solve a sparse linear system at each time step (e.g. with a conjugate gradients solver)</td>
</tr>
<tr>
<td>The simulation results (3D scalar fields) are saved in .mesh files</td>
</tr>
<tr>
<td>Inspection and validation of the results with Gmsh</td>
</tr>
</tbody>
</table>

Figure 26: Workflow
Another area for improvement is the way time steps and mesh resolutions are handled. In this work, we had to specify those parameters by hand, a time-consuming and error-prone process. A dynamic and adaptive choice would be ideal: the program could do only as much work as needed in order to meet some accuracy target, and the user would also be provided with numerical estimates of the discretization errors that occur during the simulation.

References


