ECMI Modelling Week, July 17–24, 2016, Sofia, Bulgaria

Group 2

Computation of free surface in electromagnetic shaping of liquid metals

Juan Ignacio Bedoya Cabanelas

University of Santiago de Compostela
juan.bedoya.cabanelas@gmail.com

Arnaud Bonnet

University of Strasbourg
arnaud.bonnet@etu.unistra.fr

Andreas Schafelner

Johannes Kepler University Linz
a.schafelner@drei.at

Andreas Söderlund

Lund University
andreas.j.soderlund@gmail.com

Instructor: **Jean Rodolphe Roche**

ESSTIN, Institut Elie Cartan de Lorraine, University of Lorraine
jean-rodolphe.roche@univ-lorraine.fr

Abstract. When producing ingots the use of a mold can lead to several difficulties. By using an electromagnetic field created by a set of inductors we can influence the shape of the liquid metal without the use of a mold. This report investigates the theory of a liquid metal in 2D under the influence of an electromagnetic field and creates an algorithm for iteratively finding the boundary shape of the liquid metal. The algorithm was implemented in Matlab and some test cases are shown.

2.1 Introduction

The design of inductors in electromagnetic shaping of molten metals consists in looking for the position and the shape of a set of electric wires such that the induced electromagnetic field makes a given mass of liquid metal acquire a predefined shape.
This kind of technique is used in the production of aluminium ingots (see [10]), this process can be seen in Figure 2.1.

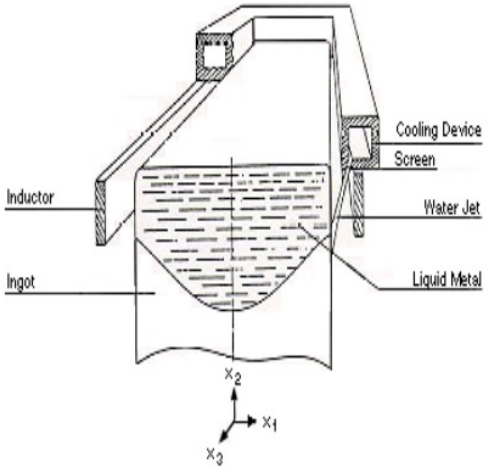

Figure 2.1: Sketch of the production of aluminium ingots.

The process consists of an induction coil surrounding the molten metal. The current flowing through the coil produces an electromagnetic field which applies a force to the surface of the molten metal. This force keeps the molten metal in an artificial mold which gives the form that we wanted. The coil gets heated so, in general, there is water flowing through the inductors just to avoid overheating and to prolong their lifespan.
The overall process is highly complex and involves different physical phenomena: electromagnetism, thermodynamics and fluid dynamics, and also we have to take into account the optimization problem to calculate the shape. In our case, making some assumptions we can consider an electromagnetic problem coupled with the optimization one.

From the mathematical point of view, the problem is reduced to a shape optimization one.
In this paper we formulate a direct optimization problem, where the position and the form of the inductors are given and the objective is to calculate the shape of a molten metal column such that we get an equilibrium energy state. The state equation of the problem is a linear system of partial differential equations obtained by modelling the electromagnetic problem (for more information about this technique see [1], [3] and [8]).
The paper has been structured as following: We begin by investigating the theory in Section 2.2. We use this to implement the theory into an algorithm in Section 2.3 which we use to produce some results in Section 2.4. Finally, we recap the report and discusses some possible future work in Section 2.5.

2.2 Theory

The general problem consists of a liquid material falling in a magnetic field which is generated by a surrounding current density. Such a setup is pictured in Figure 2.2. Let ω be the domain occupied by the fluid, $\Omega = \mathbb{R}^3 \setminus \omega$ the exterior domain and $\Gamma = \partial\omega$ the substance's boundary.

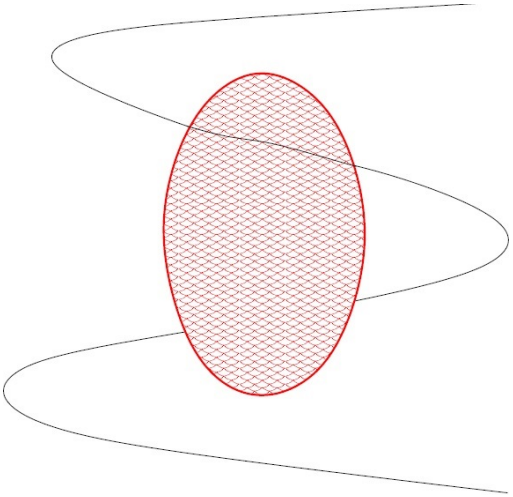

Figure 2.2: The metallic fluid (red shape) is floating due to the surrounding magnetic field generated by the black wire.

In such a scenario, the following forces affect the fluid (see [3]):

- the gravitational force
- the magnetic force
- the internal interactions of the particles

- the effect of the surface tension of the fluid

We simplify the problem by assuming the whole scene is invariant by translation along the vertical axis. A way to justify this is to say that the height of the shape is way superior to the other dimensions, and that the wires are vertically disposed. This reduces our problem to the space spanned by the vectors $\{(1,0,0),(0,1,0)\}$ which is 2-dimensional. This also removes the influence of gravity, as the third coordinate is no longer relevant (see [1]).
The wires are also assumed to have a cross-section of area equal to zero and a very high current frequency, which prevents the magnetic field from getting inside the material due to the skin effect.
Moreover, we will assume the internal velocity of the particles are negligible. Finally, we are assuming the material's density is constant over time, in other words the overall surface s_0 doesn't change. This can be rewritten with the following integral :

$$\int_\omega \mathrm{d}\mathbf{x} = s_0.$$

For the current density, we have

$$\int_\Omega j = 0,$$

where j is compactly supported.
Figure 2.3 provides an illustration of the new problem with these simplifications.

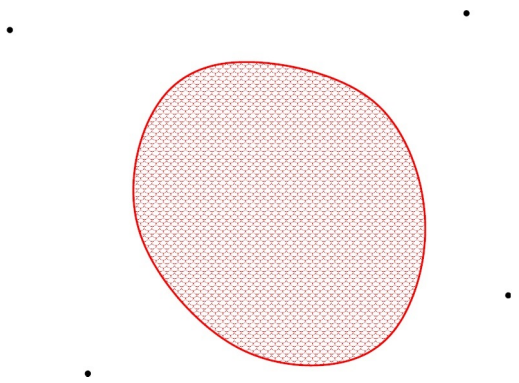

Figure 2.3: Under the specific assumptions, the shape is now a simply connected 2D domain surrounded by dot charges

The magnetic field $\mathbf{B}$ dictates the shape of the liquid metal and only depends on the position of the wires and the boundary of the liquid. Thus we have the current density $\overrightarrow{j_0} = (0,0,j_0)$ and the magnetic field $\mathbf{B}(x,y) = (B_x(x,y), B_y(x,y), 0)$. The Maxwell equations provide the expression of the global magnetic field, when taking into account the boundary conditions (see [5]):

$$\begin{aligned} \nabla \times \mathbf{B} &= \mu_0 \overrightarrow{j_0} \quad \text{in } \Omega, \\ \nabla \cdot \mathbf{B} &= 0 \quad \text{in } \Omega, \end{aligned}$$

$$\begin{aligned} \mathbf{B}\cdot\nu &= 0 \quad \text{on } \partial\omega = \Gamma, \\ ||\mathbf{B}|| &= O(x^{-2}) \text{ at } \infty, \end{aligned}$$

where ν denotes the exterior normal to the shape's boundary and μ_0 is the magnetic permeability.
We will later use the fact that $\mathbf{B}$ can be written as the sum of two magnetic fields, one generated by the current density from the wires and the other one by the induced current on the surface of the shape. The magnetic pressure at the surface in a high frequency configuration with a skin thickness equal to zero, as in our case, is equal to $\|\mathbf{B}\|^2/2\mu$. With the other remaining variable of the system, the surface tension σ, we have at equilibrium the following equation that is satisfied for each boundary point

$$\|\mathbf{B}\|^2/2\mu + \sigma\mathcal{H} = 0, \tag{2.1}$$

where $\mathcal{H}$ is the boundary curvature. Since $\mathbf{B}$ is divergence-free, there exists a scalar function φ, such that $\mathbf{B}(x,y) = (\partial_y\varphi(x,y), -\partial_x\varphi(x,y), 0)$. This function solves the scalar potential formulation of the Maxwell equations (see [6])

$$\begin{aligned} -\Delta\varphi &= \mu_0 j_0 \text{ in } \Omega, \qquad (2.2)\\ \varphi &= 0 \quad \text{in } \Gamma, \\ |\nabla\varphi| &= \mathcal{O}(1) \quad \text{as } ||x|| \to \infty, \\ \frac{1}{2\mu_0}|\nabla\varphi|^2 + \sigma\mathcal{H} &= 0 \quad \text{on } \Gamma. \qquad (2.3) \end{aligned}$$

So, the electromagnetic problem can be written by the following form

$$\begin{aligned} -\Delta\varphi &= \mu_0 j_0 \quad \text{in } \Omega, \\ \varphi &= 0 \quad \text{on } \Gamma, \\ \varphi &= \mathcal{O}(1) \quad \text{as } ||x|| \to \infty \end{aligned}$$

To solve this problem in the exterior domain Ω we decompose (see [3])

$$\varphi(x) = \varphi_1(x) + v(x), \tag{2.4}$$

where φ_1 is the fundamental solution of

$$-\nabla\varphi_1 = \mu_0 j \quad \text{in } \mathbb{R}^2,$$

and is given by

$$\varphi_1 = -\frac{\mu_0}{2\pi}\int_{\mathbb{R}^2} \ln|x-y| j(y)\, dy + \frac{\mu_0}{2\pi}\ln|x| \int_{\mathbb{R}^2} j(y)\, dy. \tag{2.5}$$

We consider a distribution of the electric current density j of the form:

$$j = I\sum_{p=1}^{m} \alpha_p \delta_{y_p},$$

where I is a given intensity of current, p, with $1 \leq p \leq m$, are subsets of $\mathbb{R}^2$, δ_p are their characteristic functions or Dirac masses functions, and α_p are dimensionless coefficients which represents the punctual charges at each point. If we take in account this, we can write the expression (2.5) as

$$\varphi_1 = -\frac{\mu_0}{2\pi}\int_{\mathbb{R}^2} \ln|x-y| I\Big(\sum_{p=1}^{m}\alpha_p\delta_{y_p}\Big)\, dy + \frac{\mu_0}{2\pi}\ln|x| \int_{\mathbb{R}^2} I\Big(\sum_{p=1}^{m}\alpha_p\delta_{y_p}\Big)\, dy.$$

By the definition of the Dirac masses, the last expression is equivalent to

$$\varphi_1 = -\frac{\mu_0}{2\pi}\sum_{p=1}^{m}\ln|x-y_p| I\alpha_p + \frac{\mu_0}{2\pi}\ln|x|\sum_{p=1}^{m} I\alpha_p,$$

and due to $\sum_{p=1}^{m}\alpha_p = 0$, the final expression of φ_1 is given by

$$\varphi_1 = -\frac{\mu_0}{2\pi}\sum_{p=1}^{m}\ln|x-y_p| I\alpha_p.$$

Once we introduce φ_1, the function φ can be computed as the expression (2.4), where $v(x)$ is the solution of the exterior problem (see [7])

$$-\Delta v(x) = 0 \quad \text{in } \Omega, \tag{2.6}$$

$$v(x) = -\varphi_1 \quad \text{on } \Gamma, \tag{2.7}$$

$$v(x) = \mathcal{O}(1) \quad \text{as } ||x|| \to \infty \tag{2.8}$$

The integral representation of $v(x)$ restricted to Γ is equal to

$$v(x) = -\frac{1}{2\pi}\int_{\Gamma} q(y) ln|x-y|\ dl(y) + C, \tag{2.9}$$

where the C is a constant and the function $q(y) \in H^{-1/2}(\Gamma)$ satisfies

$$\int_{\Gamma} q(y)\ dl(y) = 0. \tag{2.10}$$

Determining Γ given a current density j_0 is a free boundary problem. One way to solve it (the one we use) is by defining a functional E, called the energy of the system, as

$$E(\omega) = -\frac{1}{2\mu_0}\int_{\Omega} |\nabla u|^2 \mathrm{d}x + \sigma\int_{\Gamma}\mathrm{d}S \tag{2.11}$$

where u is solution to (2.2). Then the solution shape ω is a stationary point of this functional. Such a way to find the equilibrium shape is called shape optimization, as the argument of the functional is the shape ω (see [11]).

2.3 Implementation

2.3.1 Discretisation

To solve an equation in Ω we would have to discretize the whole space $\mathbb{R}^2$. As this is not possible for any computer, we will instead use the boundary element method (BEM) (see for instance [2] and [9]). For this method, we need to discretise the boundary $\Gamma = \partial\omega$. Therefore we introduce points x_i on Γ and a parametrisation of the boundary. We decided to use piecewise linear functions, so the parametrisation is not defined on the whole boundary $\partial\Omega$ but only on each segment. This means each point x on an edge $[x_i, x_{i+1}]$ can be described by a unique value $t \in [0, 1]$, i.e. $x = \gamma_i(t)$, where

$$\gamma_i(t) := (1-t)x_i + tx_{i+1}, \quad \text{for } t \in [0,1].$$

Furthermore, we restrict ourselves to closed boundary curves and simply connected domains.

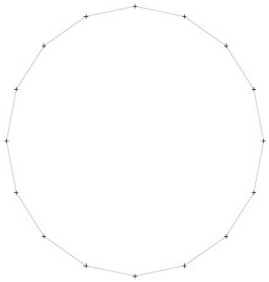

Figure 2.4: simple discretization of a circle

2.3.2 Calculations

Before we begin with the calculations, we need some notations. First, we denote each line segment of the boundary by l_i with endpoints x_i and x_{i+1} for $i = 1, ..., n$ and $x_{n+1} = x_1$. On each line segment l_i we denote the unit normal by n_i. As the normal in x_i is not uniquely defined, we decided that the normal approximation, denoted as $\hat{\boldsymbol{Z}}_i$, is defined as the mean of the normals of the two adjacent edges l_{i-1} and l_i, i.e.,

$$\hat{\boldsymbol{Z}}_i = \frac{n_{i-1} + n_i}{2}. \tag{2.12}$$

The boundary will be displaced in these directions in each point. Furthermore we define we define piecewise linear functions $\boldsymbol{Z}_i$ as

$$\boldsymbol{Z}_i(x_j) = \delta_{ij}\hat{\boldsymbol{Z}}_i. \tag{2.13}$$

The functions $\boldsymbol{Z}_i$ have compact support and define the vector field $\boldsymbol{Z}$, i.e.,

$$\boldsymbol{Z}(x) = \sum_{i=1}^{n} u_i \boldsymbol{Z}_i(x), \quad \text{for all } x \in \Gamma, \tag{2.14}$$

with $u_i \in \mathbb{R}$. We write the u_i into a vector $\bar{u} = (u_1, ..., u_n)^T$ and this unknown vector determines the evolution of the boundary in each iteration step. Now we have to calculate this vector u. We know that the total energy plus a penalty term is given by

$$E_r(\Omega, \varphi_\Omega) = -\frac{1}{2\mu_0}\int_\Omega ||\nabla\varphi_\Omega||^2 dx + \sigma\int_\Gamma d\gamma(x) + \frac{r}{2}(m(\omega) - s_0)^2,$$

where

$$\begin{aligned} -\Delta\varphi_\Omega &= \mu_0 j && \text{in } \Omega, \\ \varphi_\Omega &= 0 && \text{on } \Gamma, \\ \|\varphi_\Omega(x)\| &= \mathcal{O}(1) && \text{at } \infty. \end{aligned}$$

The derivative with respect to the shape is then defined as before, i.e.,

$$\begin{aligned} \frac{\partial}{\partial u_i}E_r(\bar{u})|_{u_i=0} = &-\frac{1}{2\mu_0}\int_\Gamma (\nabla\varphi_\Omega.n)^2(Z_i.n)d\gamma \\ &+ \sigma(\frac{x_i - x_{i-1}}{\|x_i - x_{i-1}\|} - \frac{x_{i+1} - x_i}{\|x_{i+1} - x_i\|}).\hat{Z}_i \\ &+ r\,(m(\omega) - s_0)\int_\Gamma (Z_i.n)d\gamma \end{aligned}$$

As the $\boldsymbol{Z}_i$ have compact support, the integrals can be simplified to

$$\begin{aligned} \frac{\partial}{\partial u_i}E_r(\bar{u})|_{u_i=0} = &-\frac{1}{2\mu_0}\left(\int_{l_{i-1}} (\nabla\varphi_\Omega.n)^2(Z_i.n)d\gamma \right. \\ &\left. + \int_{l_i} (\nabla\varphi_\Omega.n)^2(Z_i.n)d\gamma\right) \\ &+ \sigma(\frac{x_i - x_{i-1}}{\|x_i - x_{i-1}\|} - \frac{x_{i+1} - x_i}{\|x_{i+1} - x_i\|}).\hat{Z}_i \\ &+ r\,(m(\omega) - s_0)\left(\int_{l_{i-1}} (Z_i.n)d\gamma + \int_{l_i} (Z_i.n)d\gamma\right) \end{aligned}$$

Then at each iteration of the optimization algorithm, we can update $\bar{u}$ with

$$\bar{u} = -\rho\, H\nabla E_r(\bar{u})|_{\bar{u}=0}\,, \tag{2.15}$$

where H is a Broyden-Fletcher-Goldfarb-Shanno (BFGS) approximation of the inverse Hessian of E_r. So what is left is the computation of the shape gradient ∇E_r. For this, we need to compute the normal derivative of φ_Ω on a single edge l_i. We know we can decompose φ_Ω as $\varphi_\Omega = \varphi_1 + v$, which holds also for the normal derivative. For φ_1 we have the formula

$$\varphi_1(x) = -\frac{\mu_0}{2\pi}\int_{\mathbb{R}^2} \ln\|x - y\|\, j(y)dy + \frac{\mu_0}{2\pi}\ln\|x\|\int_{\mathbb{R}^2} j(y)\, dy.$$

As we saw at the last section, the current density j can be represented as a sum of Dirác masses, i.e.,

$$j = I \sum_{p=1}^{m} \alpha_p \delta_{x_p},$$

where x_p is the position of the conductors and α_p is the respective charge. We also know that

$$\sum_{p=1}^{m} \alpha_p = 0.$$

Using these two facts, the computation of φ_1 reduces to the calculation of a sum, i.e.,

$$\varphi_1(x) = -I \frac{\mu_0}{2\pi} \sum_{p=1}^{m} \alpha_p \ln \|x - x_p\|, \quad \text{for all } x \in \Gamma.$$

Therefore, the normal derivative is given by

$$\begin{aligned} \partial_n \varphi_1(x) &= -I \frac{\mu_0}{2\pi} \sum_{p=1}^{m} \alpha_p \partial_n \ln \|x - x_p\| \\ &= -I \frac{\mu_0}{2\pi} \sum_{p=1}^{m} \alpha_p \frac{(x - x_p)}{\|x - x_p\|^2}.n(x) \end{aligned}$$

for all $x \in \Gamma$. For the computation of $\partial_n v$ let us recall the integral single layer representation of v, i.e.,

$$v(x) = -\frac{1}{2\pi} \int_{\Gamma} q(y) ln\|x - y\| \, d\gamma(y) + c,$$

where q is a unknown function and c a unknown constant.
To approximate the solution of this problem we use boundary finite elements. We consider a piecewise approximation of $q(y)$, which is named $q_h(y)$

$$q_h(y) = \sum_{i=1}^{n} q_i e_i(y),$$

where q_i is the approximation of the value of $q(y)$ at the nodes of the partition of the boundary, and $e_i(y) = 1$ if $y \in l_i$ and zero elsewhere.
We do the weak formulation of the boundary value problem given by

Find $q(y) \in H^{-1/2}(\Gamma)$, satisfying (2.10) and verifying

$$\begin{aligned} a(q, f) &= -\frac{1}{2\pi} \int_{\Gamma} f(x) \int_{\Gamma} q(y) \ln|x - y| \; dl(y) \; dl(x) + C \int_{\Gamma} f(x) \; dl(x) \quad (2.16) \\ &= -\int_{\Gamma} \varphi_1 f(x) \; dl(x) \quad \forall f(x) \in H^{-1/2}(\Gamma). \quad (2.17) \end{aligned}$$

As we said, we are going to use boundary finite elements, so replacing the function f by e_i in the weak formulation and approximating $q(y)$ by q_h, we get

$$\begin{aligned} a(q_h, e_i) &= -\frac{1}{2\pi}\int_{l_i} e_i \int_{l_j} q_h \ln|x-y| \; dl \; dl + C\int_{l_i} e_i \; dl \quad \forall i \in \{1,\dots,n\} \\ &= -\int_{l_i} \varphi_1 e_i \; dl \quad \forall i \in \{1,\dots,n\}, \end{aligned}$$

and the second equation

$$\int_{l_j} q_h e_j \; dl = 0 \quad \forall j \in \{1,\dots,n\}.$$

Using the definition of q_h and the basis functions e_i, we get the next expression

$$\frac{1}{2\pi}\sum_{j=1}^{n} q_j \int_{l_i}\int_{l_j} \ln|x-y| \; dl \; dl + C|l_i| \; dl = -\int_{l_i} \varphi_1 \; dl \quad \forall i \in \{1,\dots,n\},$$

and the second equation

$$\sum_{j=1}^{n} q_j |l_j| = 0.$$

We can write the last two expressions as a system of the form $\mathbf{Aq} = \mathbf{b}$, where $\mathbf{A} \in \mathcal{M}_{n+1,n+1}(\mathbb{R})$, $\mathbf{q} = (q_1,\dots,q_n,C)^t \in \mathbb{R}^{n+1}$ and $\mathbf{b} = (b_1,\dots,b_{n+1})^t \in \mathbb{R}^{n+1}$. The matrix $\mathbf{A}$ is also symmetric and the coefficients are equal to

$$a_{ij} = \frac{1}{2\pi}\int_{l_i}\int_{l_j} ln|x-y| \; dl(y) dl(x), \quad i,j \in \{1,\dots,n\},$$

$$a_{n+1,j} = |l_j| \quad j \in \{1,\dots,n\},$$

$$a_{n+1,j} = a_{j,n+1} \quad j \in \{1,\dots,n\},$$

$$a_{n+1,n+1} = 0.$$

Also the coefficients of the vector $\mathbf{b}$ are equal to

$$b_i = -\int_{l_i} \varphi_1 \; dl(x) \quad i \in \{1,\dots,n\},$$

and

$$b_{n+1} = 0.$$

The final representation for v is then

$$v(x) = -\frac{1}{2\pi}\sum_{i=1}^{n} q_i \int_{\Gamma} \ln\|x-y\| \; d\gamma(y),$$

so the normal derivative is just

$$\frac{\partial}{\partial n} v(x) = -\frac{1}{2\pi}\sum_{i=1}^{n} q_i \frac{\partial}{\partial n}\int_{\Gamma} \ln\|x-y\| \; d\gamma(y) + \frac{1}{2} q(x), \quad \text{if } x \in \Gamma.$$

It can be shown that we are allowed to switch integration and differentiation, so for all $x \in l_j$ with $j = 1, ..., n$ the normal derivative of v is

$$\begin{aligned}\frac{\partial}{\partial n} v(x) &= -\frac{1}{2\pi} \sum_{i=1}^{n} q_i \frac{\partial}{\partial n} \int_\Gamma \ln \|x - y\| d\gamma(y) \\ &= -\frac{1}{2\pi} \sum_{\substack{i=1 \\ i \neq j}}^{n} q_i \int_\Gamma \frac{\partial}{\partial n} \ln \|x - y\| \, d\gamma(y) + \frac{1}{2} q_j.\end{aligned}$$

Having this representation of $\partial_n v$, we can easily compute the point evaluation with a quadrature rule. We can now calculate the shape gradient ∇E_r and moreover the update for $\bar{u}$, if we have the approximation of the inverse Hessian H. To calculate this approximation H, we use an iterative BFGS formula, i.e.,

$$H^{k+1} = H^k + \left(1 + \frac{v^T H^k v}{u^T v}\right) \frac{u u^T}{u^T v} - \frac{u v^T H^k + H^k v u^T}{u^T v}$$

with

$$\begin{aligned} u &= \bar{u}, \\ v &= \nabla E_r^k - \nabla E_r^{k-1} \end{aligned}$$

and $H^0 = I$.

2.3.3 The Algorithm

Combining all the calculations above, we derive an iterative scheme to optimize the shape.

Algorithm 1 Shape optimization

Input: number of discretisation points n, position of conductors x_p, charge of the conductors α_p, surface tension σ, current density I, penalty parameter r^0, damping parameter ρ^0, $H^0 = Id$

Output: position of the displaced points x_i^k with $i = 1, ..., n$

1: **repeat**
2: calculate the normals $\hat{\boldsymbol{Z}}_i^k$
2: assemble the linear system $A\boldsymbol{q} = \boldsymbol{b}$ and solve for $\boldsymbol{q}$
3: calculate the shape gradient ∇E_r^k
4: update $u^k = -\rho^k H^k \nabla E_r^k$
5: update H^k with the BFGS-formula
6: update ρ^k and r^k
7: displace the boundary $\Gamma^{k+1} = \left\{X = x + \sum_{i=1}^n u_i \boldsymbol{Z}_i^k(x); u_i \in \mathbb{R}, x \in \Gamma^k\right\}$
8: $k = k + 1$
9: **until** stopping criterion is fulfilled

2.4 Results

In this project we have managed to produce working code written in Matlab which correctly simulates the shape of the 2D liquid metal under the influence of an electromagnetic field. Using the results we have generated several plots illustrating the behaviour of the liquid metal. Several iterations of the algorithm

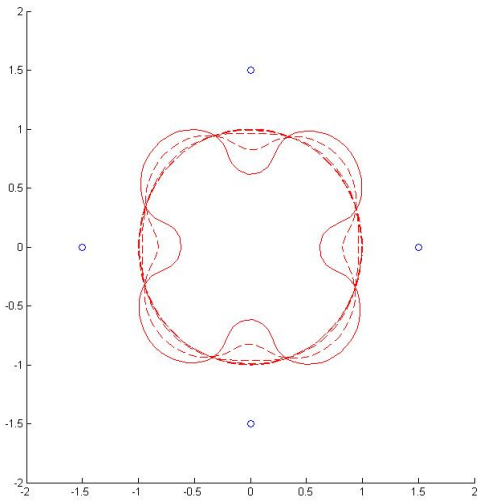

Figure 2.5: Plot showing several iteration of the updating scheme.

can be seen in Figure 2.5 where the final shape is seen as the four leaf clover in the image. The code is divided into several modular parts written as class functions broadly following the steps in the algorithm. The structure can thus be described as follows

```
@FreeSurface
  FreeSurface.m
  solve.m
  updateZ.m
  updateHk.m
  ...
```

where `solve.m` contains the steps of the algorithm.

```
function obj = solve( obj, maxIter)

tol = 1e-5;
k = 0;

oldres = 1e100;
while (k <= maxIter)
    % calculate the normals
```

```
    obj = obj.updateZ();
    % calculate q
    obj.q = obj.calcQ2();
    % calculate the shape gradient
    obj = obj.calculateGradE();
    % update of uk
    obj.uK = (-obj.ro0) * obj.Hk * obj.gradE;
    % updating the matrix for the update of uk
    obj = obj.updateHk(k);
    % stopping criterion
    res = norm(obj.gradE,2);
    sprintf('iteration %d:\n res = %f\n', k,res)
    if (oldres < res)
        break
    end
    % displace the boundary
    obj = obj.updateBoundary(obj.uK);
    % prevent extreme length segments
    %obj = obj.subdividexK(maxLength);
    % storing the shape gradient for the next iteration
    obj.oldGradE = obj.gradE;
    oldres = res;
    k = k + 1;
    obj.ro0 = obj.ro0 *2;
    obj.r0 = obj.r0 / sqrt(1.1);
end
```

We initiate the object in the following function

```
function obj = FreeSurface(xP, alphaP, I, r0, ro0, mu0, sigma, radius, numNodes)
    obj.xP = xP;
    obj.alphaP = alphaP;
    obj.I = I;
    obj.r0 = r0;
    obj.ro0 = ro0;
    obj.mu0 = mu0;
    obj.sigma = sigma;
    obj.radius = radius;
    obj.numNodes = numNodes;

    % calculations for rest

    % initialize the nodes
    t = linspace(0, 2*pi, obj.numNodes + 1);
    % we don't want a double point at t=0 and t=2pi
    t = t(1:end-1);
    obj.xK = obj.radius*[cos(t); sin(t)]';
```

```
    % initialize the vector field
    obj = obj.updateZ();

    % calculate the inital volume s0
    obj.s0 = polyarea([ obj.xK(:,1); obj.xK(1,1)], [ obj.xK(:,2); obj.xK(1,2)]);

    % initialise the approx. of the Hessian with the identity
    obj.Hk = eye(obj.numNodes);
    obj.gradE = zeros(obj.numNodes, 1);
end
```

In the code we can also move the positions of the conductors to affect the shape of the liquid metal as seen in the initialization function as `xP`. By keeping the number of subdivisions constant and moving the conductors closer and closer together we can force the solution to be unstable.

Because we keep the number of subdivisions constant, the solution becomes unstable in the last graph of Figure 2.6. This could be avoided by introducing more subdivisions as we move the conductors. As we move the conductors even closer to each other we get the graph seen in Figure 2.7. Here, the conductors are so close to the boundary of the liquid metal at the start of the iteration that the algorithm has difficulty seeing them as being outside the boundary. This makes it close to an interior problem, something the code is not designed to solve.

2.5 Conclusion

In this project we have modeled liquid metal in 2D under the influence of an electromagnetic field. We have implemented this in Matlab and the code is working and is modular. The code is somewhat robust but we have pointed out and exemplified an occasion where the code creates an unstable solution.
For future work it would be interesting to generalize the theory to 3D (see [4]) and create working code for that too. The code would also need to be speeded up. This would lead to the possibility of adding more subdivisions without having slower code and the precision would increase.

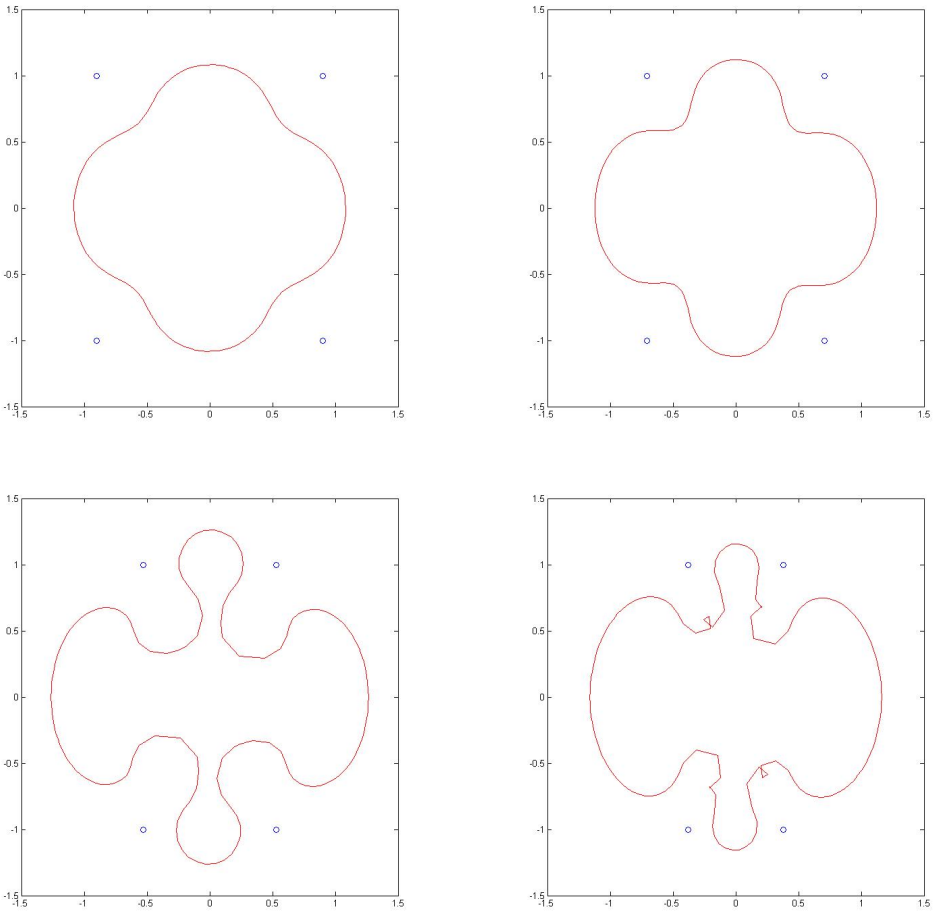

Figure 2.6: Looking from top right to bottom left we see the conductors moving closer together horisontally.

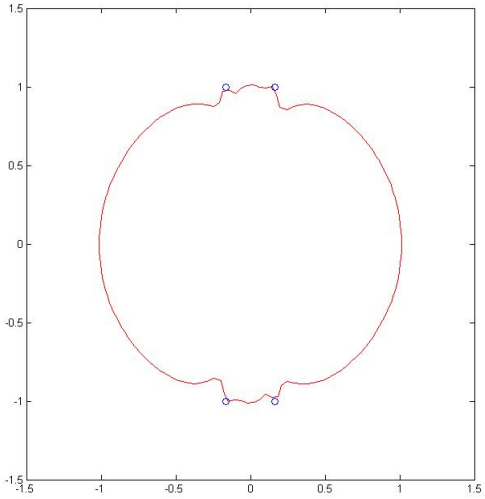

Figure 2.7: Plot showing several iteration of the updating scheme.

Bibliography

[1] A. Canelas, J.R.Roche and J. Herskovits, Inductor shape optimization for electromagnetic casting, Structural and Multidisciplinary Optimization,39 : 589-606,DOI: 10.1007/s00158-009-0386-0, (2009)

[2] G. Hsiao, W. L. Wendland, Boundary Integral Equations, Vol 164, Springer-Verlag, Berlin, 2008

[3] Jean R. Roche, Classical Shape Optimization The Electromagnetic Casting Problem, 2016, Presentation, Unpublished

[4] M. Pierre, Elisabeth Rouy. A Tridimensional Inverse Shaping Problem. [Research Report] RR-2655, INRIA. 1995, pp.34.

[5] M. Pierre and J.R. Roche, Computation of free surfaces in the electromagnetic shaping of liquid metals by optimization algorithms, European Journal of Mechanics. B Fluids, Vol. 10, n 5, pp 489-500, 1991

[6] A. Canelas, J.R.Roche and J. Herskovits, The inverse electromagnetic shaping problem. Structural and Multidisciplinary Optimization, 38(4), 389-403, 2009, DOI : 10.1007/s00158-008-0285-9

[7] Jean R. Roche. Adaptative Newton-like Method for Shape Optimization. Control and Cybernetics, Polish Academy of Sciences, 2005, 34, pp.363-377.

[8] O. Coulaud and O. Henrot, Numerical approximation of free boundary problem arising inelectromagnetic shaping. [Research Report] RR-1610, INRIA. 1992, pp.28.

[9] R. Kress, Linear integral equations, Applied Mathematical Sciences, Vol. 82, Springer-Verlag, 1999.

[10] O. Besson, J. Bourgeois, P.-A. Chevalier, J. Rappaz, and R. Touzani. 1991. Numerical modelling of electromagnetic casting processes. J. Comput. Phys. 92, 2 (January 1991), 482-507. DOI=10.1016/0021-9991(91)90219-B http://dx.doi.org/10.1016/0021-9991(91)90219-B

[11] M.C. Delfour and J.-P. Zolésio, Shapes and Geometries: Metrics, Analysis, Differential Calculus, and Optimization, Second Edition, SIAM series on Advances in Design and Control, 2011.