

Georg-Simon-Ohm-University of Applied Sciences Nuremberg

Report of the application project at the Faculty of AMP

Simulation of a medical therapy method with finite elements

Martin Michel

Keßlerplatz 12

DE-90489 Nuremberg

Advisor: Prof. Dr. rer. nat. Tim Kröger

Advisor: Prof. Dr. rer. nat. habil. Jörg Steinbach

Advisor: Prof. Dr. rer. nat. Thomas Lauterbach

Nuremberg, 01. January 1900

Contents

1. Introduction to radio frequency ablation	4
2. Computer-aided simulation of radio frequency ablation	4
2.1. Discrete Numerical Simulation	4
2.2. About Errors in simulations and numerical approaches	5
2.3. The physics behind radio frequency ablation	6
3. Mathematical aspects of discrete simulation	8
3.1. Theory of finite elements	8
3.1.1. Elliptical problems	8
3.1.2. Assembling system of equations	9
3.1.3. Boundary conditions	9
3.1.4. Parabolic problems	10
3.1.5. FEM in Electrostatics	10
3.1.6. FEM in Temperature Fields / perhaps Fluid Dynamics	10
3.2. Axial symmetrie	10
4. Discretization of PDEs	11
4.1. Computational Domain	11
4.2. FEM in cylindric Coordinates	12
4.3. PDE for Electric potential	12
4.3.1. Discretization of the problem	12
4.3.2. Inner Domain	13
4.3.3. Electrodes	14
4.3.4. Outer boundary	14
4.3.5. Rotation axis	15
4.4. Calculation of electrical energy	15
4.5. PDE for temperature Distribution	15
4.5.1. Discretization of the problem	15
4.5.2. Needle	17
4.5.3. Outer boundary	17
5. Applied FEM technologies	18
5.1. Weak solutions	18
5.1.1. Electric potential	18
5.1.2. Temperature Distribution	18
5.2. Discretization / Triangulation	20
5.3. Assembling system of equation	20
5.4. ODE for parabolic problems	20
5.5. Solving the system of equations	21
5.6. Error estimations	21
5.6.1. Element Error on 2D geometry	21
5.6.2. Time step error on discrete intervalls	21

6. Other numerical aspects	22
6.1. Surface integral	22
6.2. Numerical gradient on discrete points	22
6.3. Grid refinement	22
6.4. Troublesome neighbours	22
7. Applied simulation	23
7.1. Generating TestData / Get reference data	23
7.2. Solving the PDEs	23
7.3. Combine everything to continous time dependent simulation	23
7.4. Interpretation of result numbers	23
8. Programming technologies	23
8.1. Performance Optimization	23
8.2. MatLab vs C++	23
8.3. Graphical output	24
9. Summary and Outlook	24
9.1. Project Summary	24
9.1.1. strengths and flaws	25
9.1.2. future modifications	25
9.2. State of the current Research	26
9.3. Other FEM projects and software	26
A. Test Source code Visual C++	29
B. Source code MatLab	29

1. Introduction to radio frequency ablation

- Medical Treatment of Tumor
- Minimally invasive surgery
- Radio frequency ablation
- RFA has been proven to be one of the best therapies for small tumors (smaller than 5 cm in size)
- Simulation of RFA is important
- If you have stable and converging simulation model, it is easy to try out different modifications without cumbersome experiments
- It is easy to configure
- Is also used on a larger scale for Virtual Reality simulation to train mediciners and surgeons
- For more information see TODO
-
- - Motivation / This project in General
- This is the documentation to study approach on numerical simulation
- Radio frequency ablation is the underlying model
- Study finite elements and behaviour of convergence
- FEM simulation for RFA in 2D space domain with discrete time solutions
- The modeled problem is a 3D problem
- By modeling one needle there, is a axis symetrie in the domain
- 3D axis symmetric cylindrical domain can be described with 2D torus elements and special FEM techniques

2. Computer-aided simulation of radio frequency ablation

2.1. Discrete Numerical Simulation

- Physical relationships can be described by physical laws which are described by mathematical relationships
- Usually highly complex systems that are described by ODEs, PDEs and statistics
- Phxsic can be approximated by simplified models
- In real world physics these models are usually bounded by reality
- In most physic fields excluding quantum mechanics and modern theoretical physics, rough

models are determined, definite mathematical solutions can be found

- Since the physics is determined, simulations can be done instead of experiment and measurement
 - In practice, geometrical dimensions of points of interest are often usually vague
 - In many cases there is no reasonable analytical approach to solve these problems, especially in the field of engineering
 - Modern numerical approaches are very flexible in this regard
 - Instead of a smooth analytical solution, an arbitrarily precise approximation can often be calculated
 - Simulations done right can be easily modified and adapted to different models and boundaries
 - These models are usually described by PDEs
-
- Finite difference method
 - Finite element method
 - TODO: most frequently used methods
 - In this approach we will model radio frequency ablations with simple finite element techniques

2.2. About Errors in simulations and numerical approaches

- see TUM dissertation
- There are different sources for errors following the simulation from the line from the real problem down to the discrete solution
- Idealization error: discrepancy between reality and the idealized reality and the idealized constitutive laws and boundary conditions -> Systems are often way more complex in reality, every patient is different
- Modeling errors: discrepancy between mathematical formulation and physical model -> e.g. using dimensionally reduced approaches, like linear dependencies or even constant parameters
- Discretization errors: discrepancy between the continuous description and discrete description of the model
- Solution errors: using iterative approximation methods and rounding errors
- It's basically a butterfly effect
- Optimizing one error source often conflicts with another one -> e.g. handling nonlinearity can cause fatal numerical errors (at least that's what Kroeger said ...)

2.3. The physics behind radio frequency ablation

- Needles stuck in tissue have electrodes with generate a potential
 - Generating electrical energy by power of an extern generator
 - Energy does not get lost
 - Heats the tissue, temperature rises do to continuing energy input
 - Generated heat is distributed on the tissue,
 - Thus additionally there is a cooling effect by the perfusion of blood
 - Heated tissue can kill harmful cells
 - The whole process is regulated from outside
-
- Interesting for our simulation is:
 - Temperature distribution over time
 - What else? TODO
-
- For the simulation, we will use a RFA model as described following kroeger
 - For simplicity, material parameters are seen constant
 - In real world, material parameters are highly dependant on body conditions, temperature and vary heavily with different patients
 - For some research to this topic, see Stein
-
- Electrical energy depends on the electrical power
 - Power is defined by potential of the domain

$$-\nabla \cdot (\sigma(x, y, z, t) \nabla \phi(x, y, z, t)) = 0 \quad (1)$$

Power applied by the probe:

$$p(x, y, z, t) = \sigma(x, y, z, t) \cdot |\nabla \phi(x, y, z, t)|^2 \quad (2)$$

Due to tissue resistance, the effective power of the generater is obtained by a scaling. Thus can be modeled as follwing

$$Q_{rf} = p(x, y, z, t) \cdot \frac{p_{eff}(t)}{p_{total}(t)} \quad (3)$$

$$p_{total}(t) = \int_{\Omega} p(x, y, z, t) dx \quad (4)$$

$$p_{eff}(t) = \frac{4 \cdot p_{setup} \cdot R_{tis}(t) \cdot R_I}{(R_{tis}(t) + R_I)^2} \quad (5)$$

$$R_{tis}(t) = \frac{U^2}{p_{total}(t)} \quad (6)$$

σ : electric conductivity

$\varphi = \varphi(x, y, z, t)$: electric potential

$p = p(x, y, z, t)$: power

R_{tis} : tissue resistance

R_I : inner resistance generator

p_{setup} : setup power generator

U : potential difference

- No energy is lost
- Electrical Energy becomes heat energy by Tissue resistance
- Heat Energy is distributed by heat equation

$$\partial_t(\rho c T) - \nabla \cdot (\lambda \nabla T) = Q \quad (7)$$

The heat equation is a well known parabolic partial differential equation.

We are assuming ρ and c are constant

ρ : density

c : specific heat capacity

λ : thermal conductivity

v : perfusion coefficient

$T = T(x, y, z, t)$: temperature

$Q = Q(x, y, z, t)$: heat energy

- Additionally to temperature distribution, there is the cooling effect of perfusion

$$Q_{total} = Q_{rf} + Q_{perf} \quad (8)$$

- Q_{rf} is described above
- Q_{perf} is blood perfusion

$$Q_{perf}(x, y, z, t) = v_{perf} \cdot \rho_{blood} c_{blood} \cdot (T_{body} - T(x, y, z, t)) \quad (9)$$

- Discretization of the equation in space and time domain
 - Time dependency can be modeled continuously or in discrete intervals
 - Discrete intervals are typically more practical in modeling but less efficient or exact
 - Discrete intervals can be refined if necessary
-
- As described above, the whole simulation will be done on a 2D half section representing the 3D axis symmetric geometry

3. Mathematical aspects of discrete simulation

3.1. Theory of finite elements

3.1.1. Elliptical problems

- Elliptical problems in general
 - build up system of PDE's to describe problem
-
- Ritz-Galerkin-Verfahren, for Details see ref book TODO
 - This documentation will be focused on the 3D -> 2D aspect

- Using the cylindric domain, different domains, will be described later

3.1.2. Assembling system of equations

- Systems of equations by finite number of elements calculated with shape function
- solution is approximation of real solution

$$Matrix \quad (10)$$

3.1.3. Boundary conditions

- Dirichlet with trick, define value to be solution of variable
- TODO

- Neumann boundary with curve integral

$$f = f_{vol} + f_{surf} \quad (11)$$

$$b = \int_{\Gamma_2} g \cdot \phi_i dS \quad (12)$$

- Note : In 2D dimension the surface integral is around the boundary line
- For natural boundary conditions $g(x) = 0$, the surface integral added is zero, which is a trivial solution

3.1.4. Parabolic problems

- Parabolic / time-dependent problems
- Usually involve an elliptic problem with additional time dependent factor
- Ignoring time dependency at first and calculate the resulting elliptical problem
- Build system of equations for elliptical problem
- Describe the parabolic PDE as an ODE with matrices from elliptical PDE
- Solving the system of ODE equations over discrete time intervalls
- Runge-Kutta
- For this simulation, backward euler / implicit euler formula does the job

3.1.5. FEM in Electrostatics

- TODO
- General info
- special domain
- boundary conditions

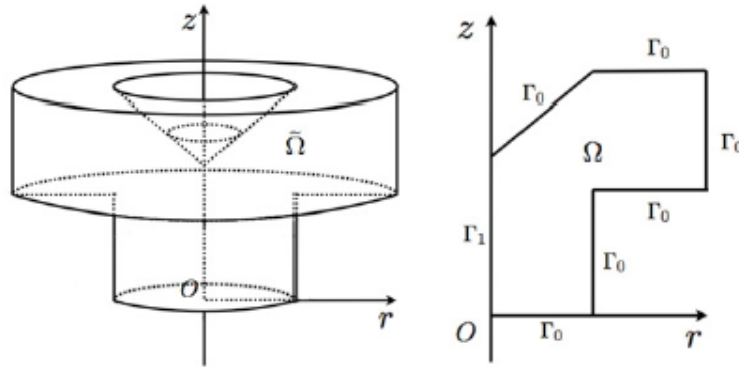
3.1.6. FEM in Temperature Fields / perhaps Fluid Dynamics

- General info
- boundary condition (heat source or sink)
- In this simulation, heat source is from electrical energy

3.2. Axial symmetrie

- Using axial symmetrie to simplify computations
- reducing one dimension 3D -> 2D
- problems are aquivalent
- significant savings calculations time and complexity

- approach: fourier decomposition in angular direction to reduce dependency on the angular φ
- using static models, only dependency on space
- maybe Torus elements



A three-dimensional axisymmetric domain $\tilde{\Omega}$ (left); the corresponding two-dimensional half section Ω (right).

Figure 1: Schematic projection on 2D half section

- TODO: write this more formal - Lax-Milgram-Lemma and Poincaré inequality on 3D-Domain determine a unique solution
- $u \in H_r^1(\Omega) \cap \{v|_{\Gamma_0}\}$ - Trace-operator
- rotation axis becomes an artificial boundary on 2D half section
- A more formal description can be found in :
- Use input from article TODO

4. Discretization of PDEs

4.1. Computational Domain

- Using one needle, whole geometry domain is axis symmetric around one needle
- Problem can be reduced to 2D problem using ring elements and cylindric coordinates
- Eliminate dependency on angular ϕ from the calculations
- For visualisation, symmetric results can be reconstructed to 3D
- Whole calculation will be in cylindric coordinates

4.2. FEM in cylindric Coordinates

- Rewrite the equations to cylindric coordinates
- Calculations are made on a cross-section with angular $\phi = 0$
- Define boundaries -> new artificial boundary around the rotation axis to be taken into consideration
- Explain how the new boundary can be treated

Laplace in cartesian coordinates:

$$\nabla^2 := \Delta := \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \quad (13)$$

Laplace in cylindric coordinates:

$$\Delta := \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} + \frac{1}{r^2} \frac{\partial^2}{\partial \phi^2} + \frac{\partial^2}{\partial z^2} \quad (14)$$

TODO

Laplace in polar coordinates:

$$\Delta := \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} + \frac{1}{r^2} \frac{\partial^2}{\partial \phi^2} \quad (15)$$

4.3. PDE for Electric potential

4.3.1. Discretization of the problem

4 areas can be distinguished from a mathematical point of view

- Inner domain
- Fixed Potential of electrodes
- Outer boundaries with no fixed potential -> Robin
- Rotation axis, artificial boundary -> Neumann

Constant material parameters ->

Coefficients are constants

This will simplify calculations and coupling

4.3.2. Inner Domain

- The electric potential of the inner domain is described as :

$$-\nabla \cdot (\sigma(x, y, z, t) \nabla \varphi(x, y, z, t)) = 0 \quad (16)$$

- Elliptical boundary problem
- Assuming constant material parameters: $\nabla \sigma = 0$
- Solution is independent from σ so we can cut it out - Equation becomes Laplace's equation, φ becomes time independent

$$-\Delta \varphi(x, y, z) = 0 \quad (17)$$

- Using a cylindric domain, it is preferred to change coordinate systems
- Axis symmetry is given cylindrical coordinates (see ref Laplace in cylinder)

$$-\Delta \varphi(r, \phi, z) = -\frac{1}{r} \frac{\partial \varphi}{\partial r} - \frac{\partial^2 \varphi}{\partial r^2} - \frac{1}{r^2} \frac{\partial^2 \varphi}{\partial \phi^2} - \frac{\partial^2 \varphi}{\partial z^2} = 0 \quad (18)$$

- Since the domain has axis symmetry, the solution for φ is independent from the angular ϕ
- So equation basically simplifies to

$$-\nabla(\sigma \cdot \nabla \varphi(r, \phi, z)) \stackrel{!}{=} -\Delta \varphi(r, z) = -\frac{1}{r} \frac{\partial \varphi}{\partial r} - \frac{\partial^2 \varphi}{\partial r^2} - \frac{\partial^2 \varphi}{\partial z^2} = 0 \quad (19)$$

To keep the following descriptions simple, we will (ignore) "verzichten" the explicit dependencies r and z , since this is trivial

4.3.3. Electrodes

- Potential difference on the electrodes is fixed by definition
- For calculations, potential will be defined as ± 1 chosen arbitrarily
- Bipolar probes have one fixed potential at each electrode

$$\varphi = \pm 1 \quad (20)$$

- Other values would be possible too, but are rather unusual at monopolar probes

4.3.4. Outer boundary

- An easy approach is a simplification with natural boundary conditions

$$n \cdot \nabla \varphi = 0 \quad (21)$$

- In cylindrical coordinates

$$TODO \quad (22)$$

- In actual RFA treatment, there is a mass to nullify the potential in the outer regions
- For the lower boundary, this behaviour can be easily modeled with dirichlet conditions

$$\varphi = 0 \quad (23)$$

- Neither the neumann nor the dirichlet conditions actually reflect physics properly,, but is rather accurate enough for this approach
- Since there are made other simplifications, especially with the unrealistic "Annahme" of constant material parameters, this simplification of boundary conditions is within the margin of acceptable errors

4.3.5. Rotation axis

- this is an artificial boundary, as discribed above
- To keep the axis symmetry, "leicht einsehbar" that here must natural boundary conditions
- Other conditions would break the trace operator
- A more mathematical description for this "Annahme" will described in the following chapter
-

$$n \cdot \nabla \varphi = 0 \quad (24)$$

4.4. Calculation of electrical energy

- φ can be calculated on every discrete point
 - Calculate power for every point with equations above (TODO add ref here)
 - Phi is evaluated on discrete mesh points
 - In a nutshell, Tissue Resistance and nonlinear behaviour of the generator generate an effective power which is as well the total power of the domain a scalar multiplication factor for every discrete value for phi
- Thus leading to the calculation of the electric energy from electric power above

4.5. PDE for temperature Distribution

4.5.1. Discretization of the problem

From physics above, the temperature distribution is modeled by the heat equation:

$$\partial_t(\rho c T) - \nabla \cdot (\lambda \nabla T) = Q \quad (25)$$

The heat equation is a well known parabolic partial differential equation.

$T = T(r,z,t)$ = temperature

$Q = Q(r,z,t)$ = heat energy

The material parameters are taken as constant, as described above

ρ = density

c = specific heat capacity,

λ = thermal conductivity

- In temperature range of radio frequency ablation, c and λ are especially nonlinear depending on the temperature, for which the phase change of water is reason to
Evaporated water has a totally different heat capacity and thermal conductivity, also the transition is smooth and not abrupt

- However, Modeling the phase change of water and corresponding heat capacity does in practise lead to big numerical problems

- It is reasonable to at least model the temperature dependency of λ , which leads to the following description

Cylindrical coordinates: see 'Transient Heat Transfer in a Partially Cooled Cylindrical Rod' from Lawrence Agbezuge

$$\rho c \frac{\partial T}{\partial t} - \frac{d\lambda}{dT} \left[\left(\frac{\partial T}{\partial r} \right)^2 + \left(\frac{\partial T}{\partial z} \right)^2 \right] - \lambda \left(\frac{\partial^2 T}{\partial r^2} + \frac{1}{r} \frac{\partial T}{\partial r} + \frac{\partial^2 T}{\partial z^2} \right) = Q \quad (26)$$

- For sake of simplicity in this academic approach, we will ignore the phase change of water willingly as a whole

- So it is "good" to assume that λ would also be constant too, which greatly reduces the complexity of the whole PDE

$$\rho c \frac{\partial T}{\partial t} - \lambda \left(\frac{\partial^2 T}{\partial r^2} + \frac{1}{r} \frac{\partial T}{\partial r} + \frac{\partial^2 T}{\partial z^2} \right) = Q \quad (27)$$

- The right hand side is described in the physics, see ref

- It is depending on the generated electrical energy and the cooling effect of perfusion

$$Q_{total} = Q_{rf} + Q_{perf} \quad (28)$$

- Since the material parameters are constant, Q_{perf} can be easily evaluated linear depending on T for every discrete time step

4.5.2. Needle

- If we assume that the probe is cooled from outside, it is as simple and also accurate assumption to take the body temperature as constant temperature for the probe over the whole intervall

$$T = T_{body} \quad (29)$$

- Usually body temperature is around $\cong 37$ degree Celsius or 310.15 degree Kelvin
- If the probe is not cooled, another simple approach is to model the behaviour of needle likewise as the outer boundary

4.5.3. Outer boundary

- Assuming there is no "Disturbing" from outside, natural boundary conditions conditions are the correct assumption for the heat equation

-

$$n \cdot \nabla \varphi = 0 \quad (30)$$

- These can also be used to model non cooled probes
- In the following, we will discribe how the PDE models for φ and T can be transformed in the context of finite elements and numerically evaluated at discrete points

5. Applied FEM technologies

5.1. Weak solutions

5.1.1. Electric potential

Electric potential / Laplace's equation in cylindrical domain:

$$a_w(u, v) := \int_{\Omega} (\partial_r u \partial_r v + \partial_z u \partial_z v) r dr dz = \int_{\Omega} f v r dr dz \quad (31)$$

$$u \in H_r^1(\Omega) \cap \{v|_{\Gamma_0} = 0\} \quad (32)$$

$$v \in H_r^1(\Omega) \cap \{v|_{\Gamma_0} = 0\} \quad (33)$$

Specific PDE for electric potential, inner domain:

$$a_w(u, v) := \int_{\Omega} (\partial_r u \partial_r v + \partial_z u \partial_z v) r dr dz = 0 \quad (34)$$

5.1.2. Temperature Distribution

This is basically the problem above but as a hyperbolic problem

Using semidiscrete solution and iterate solution over time

We are applying method of the discontinuous galerkein fem

For reference see Jung, Langer : Methode der finiten Elemente für Ingenieure, chapter 7.1

Weak formulation for the problem:

We are looking for $u(r, z, t) \in V_{g1}$ with $\dot{u} \in L_2(\Omega)$ for almost every $t \in (0, T)$, so

$$(\dot{u}, v)_0 + a(t; u, v) = \langle F(t), v \rangle \text{ for all } v \in V_0 \quad (35)$$

and for almost every $t \in (0, T)$ is the "Anfangsbedingung -> suche englische Formulierung"

$$(u(r, z, 0), v)_0 = (u_0, v)_0 \text{ for all } v \in V_0 \quad (36)$$

The formal model above is given by

$$\begin{aligned}
(\dot{u}, v)_0 &= \int_{\Omega} \dot{u}(r, z, t) v(r, z) dr dz = \int_{\Omega} \frac{\partial u(r, z, t)}{\partial t} v(r, z) dr dz, \\
a(t; u, v) &= \int_{\Omega} \left[\lambda_1(r, z, t) \frac{\partial u}{\partial r} \frac{\partial v}{\partial r} + \lambda_2(r, z, t) \frac{\partial u}{\partial z} \frac{\partial v}{\partial z} \right] \cdot r \cdot dr dz + \int_{\Gamma_3} \alpha(r, z, t) u(r, z, t) v(r, z) ds, \\
\langle F(t), v \rangle &= \int_{\Omega} f(r, z, t) v(r, z) dr dz + \int_{\Gamma_2} g_2(r, z, t) v(r, z) ds + \int_{\Gamma_3} \alpha(r, z, t) u_A(r, z, t) v(r, z) ds, \\
V_{g1} &= \text{TODO}, \\
V_0 &= \text{TODO}
\end{aligned}$$

Adapted for the temperature distribution, assuming λ and all material parameters are constant:

$$a_w(t; u, v) := \int_{\Omega} \rho c (\partial_t u \cdot v) dr dz + \int_{\Omega} \lambda (\partial_r u \partial_r v + \partial_z u \partial_z v) r dr dz = \int_{\Omega} f v r dr dz \quad (37)$$

5.2. Discretization / Triangulation

- Grid generation and refinement
- 2D domain
- initial coarse grid triangulation can be done by hand
- coarse grid can be refined by algorithm
- refinement of triangle into 4 new pieces
- [insert picture here](#)

5.3. Assembling system of equation

- assemble elementwise
- Approximate elements with linear regression functions
- Calculation on reference triangles to speed up performance and accuracy

Linear regression functions for reference triangles:

$$\phi_1(\xi, \eta) = 1 - \xi - \eta \quad (38)$$

$$\phi_2(\xi, \eta) = \xi \quad (39)$$

$$\phi_3(\xi, \eta) = \eta \quad (40)$$

- use symmetrie of elements if coefficients are constant or allow it
- Add boundary conditions afterwards
- grouping of similar calculations allow more vectorized operations

5.4. ODE for parabolic problems

[TODO](#)

5.5. Solving the system of equations

- direct solver would be preferred in general
- if numbers of element grows, this computations takes up lots of memory
- up from a certain point, high amount memory can't be efficiently handled with RAM anymore
- using additional memory is extremely costly in terms of calculation speed
- since matrix is sparse, iterative solver can be very fast
- iterative solver is preferred if elements grow in numbers
- TODO : when should you use iterative solver

5.6. Error estimations

5.6.1. Element Error on 2D geometry

- H1-Norm
- L2-Norm

- The analytical solution of the problem is unknown, but we can estimate it by calculating different element sizes on different sized fine grids

Idea: Define geometric some geometric coordinates on triangle and look for best matching or choose some points on different geometrical entities

Refine triangles around that point and see whether the error is smaller on the finer grid

5.6.2. Time step error on discrete intervalls

See Skript Numerik 2 Kroeger, after some take make a smaller or larger time step and compare the difference of results, use the smaller steps for calculation

6. Other numerical aspects

6.1. Surface integral

- Sum of integral of every single triangle
- Single triangle on discrete gauss quadrature points
- Can use more nodes, which is usually worth the calculation effort
- Implemented on 3 or 7 points
- TODO insert picture

6.2. Numerical gradient on discrete points

- Trial functions approach the solution of the variation problem
- in our case, gradient is constant on every triangle
- actually discrete gradient on vertices is interesting
- vertices as average value of surrounding the surrounding triangles

6.3. Grid refinement

- Refine triangle uniformly into 4 new ones
- Can be done as efficient vector operation

6.4. Troublesome neighbours

- Collision on corner nodes and boundaries of electrodes
- Corner node is defined well for dirichlet as well as for neumann conditions
- Handling dirichlet nodes is done by setting row and column vectors to zero and rhs to dirichlet value, which is a problem for the neumann boundary
- Dirichlet value has higher priority for corners
- This is a problem for the node on the neumann boundary next to the corner
- So this exactly neumann node has some unforeseen behaviour
- To handle this, best way has shown to be to ignore the result for this node and interpolate the

result by weighing the neighboring nodes

- This particular solution might have its obvious flaws, but gets pretty good results in practise

7. Applied simulation

7.1. Generating TestData / Get reference data

- Material parameters see Stein TODO
- Using specification data from electrical generator TODO

7.2. Solving the PDEs

7.3. Combine everything to continuous time dependent simulation

7.4. Interpretation of result numbers

- Interpret numbers
- Compare with data from experiment or other simulations
- TODO compare with other simulations

8. Programming technologies

8.1. Performance Optimization

8.2. MatLab vs C++

- Could have done the whole simulation using only MATLAB
- MATLAB is a scripting language that calls Fortran Subroutines, which are highly efficient in calculating problems of linear algebra
- However, MATLAB has to call these subroutines in an efficient way to take these performance

advantages

- It is extremely easy to write bad and inperformant code in MATLAB, if it is used in the wrong way
- Efficient implementation required hardcoding routines and is very stiff
- To me it was important to write flexible code, that can be easily adapted and extended to try out different modification
- This is way easier when using loops and subroutines instead of hard coded implementations, also the code because way more easier to read and fix
- So I was going for a combined implementation of MatLab and C++
- When using flexible code design, C++ allows performance advantages in using loops etc over MATLAB
- However, MATLAB allows easy function hadnling, what makes the algorithms more accessible for the reader
- In the end I combined the advantages of both languages
- MatLab serves as frame for the pre- and postprocessing, like grid generation and graphical output of the numerical results.
- Also the scripts serve as a mathematical documentation of the whole simulation for the reader
- Computation intense subroutines are done in C++

8.3. Graphical output

- Data can be easily recreated
- Solution is axis symmetric, so values apply for every angle ϕ
- This gets 3D data in cylindrical coordinates
- Can be reevaluated into cartesian coordinates to be plotted
- Discrete points are linear interpolated

9. Summary and Outlook

9.1. Project Summary

- One could argue that writing a numerical simulation from scratch is a waste of time
- There are many highly useful numerical software solutions for numerical simulation and numer-

ical problems

- Usually there is no need to write an own detailed implementation
- However, creating own scripts and implementations helps to understand numerical problems and error sources
- This approach helps enormously to increase the ability to use these software products effectively and to generate better simulations and is mandatory to improve
- There can also be no software developer without understanding how a computer works numbers
- Implementing the algorithms on own trains understanding of numerical problems
- Should be programmed by everyone working with finite elements at least once

9.1.1. strengths and flaws

- good: numerical results do match the general expectation
 - "zuverlässige" convergency
 - numerical solution matches "qualitativ" the expectations
 - Implementation can be easily adopted to more complex approaches
-
- bad: model is way to simplified to represent real world conditions of RFA
 - Even further modifications would not change the results heavily

9.1.2. future modifications

- Material parameters are dependent on Temperature and potential ->
- Using variable instead of fixed material parameters
- Take the evaporation of water into account
- Different types of perfusion
- Defining more realistic and complex boundary conditions
- Perhaps a second needle in a 3D simulation
- Basic algorithm can be adopted to 3D problems

9.2. State of the current Research

- Research in the simulation of medical therapy methods
- TODO

9.3. Other FEM projects and software

- There is a lot of good commercial FEM software out there
- Matlab PDE toolbox, libraries with efficient implementation of almost every aspect on solving PDEs
- COMSOL Multiphysics, a High-Power simulation program with model builder and GUI
- ANSYS, like COMSOL but a little weaker but more widely used

- Also, good open source software
- FENICS
- FREEFEM

References

- [1] Tim Kröger et. al. *Numerical Simulation of Radio Frequency Ablation with State Dependent Material Parameters in Three Space Dimensions*. Springer, 2006.
- [2] Hiroki Watanabe et. al. *Temperature Dependence of Thermal Conductivity of Liver Based on Various Experiments and a Numerical Simulation for RF Ablation*. IEEE EMBS 2010 : 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2010.
- [3] Thomas Stein. *Untersuchungen zur Dosimetrie der hochfrequenzstrominduzierten interstitiellen Thermotheapie in bipolarer Technik*. Institut für Med./Techn. Physik und Lasermedizin der Freien Universität Berlin, 1999.
- [4] Klaus Knothe u. Heribert Wessels. *Finite Elemente, Eine Einführung für Ingenieure*, 5. Auflage. Springer Vieweg, 2017.
- [5] Michael Jung u. Ulrich Langer. *Methode der finiten Elemente für Ingenieure, Eine Einführung in die numerischen Grundlagen und Computersimulation*, 2. Auflage. Springer Vieweg, 2013.
- [6] Aklilu T. G. Giorges. *Finite Element and Finite Difference Methods for Elliptic and Parabolic Differential Equations*. InTech, 2016.
- [7] John Loustau. *Numerical Differential Equations, Theory and Technique, ODE Methods, Finite Differences, Finite Elements and Collocation*. World Scientific, 2016.
- [8] Arndt Bode et. al. *Partielle Differentialgleichungen, Numerik und Anwendungen, Konferenzen Band 18*. Forschungszentrum Jülich GmbH, 1996.
- [9] Claus-Dieter Munz. *Numerische Behandlung gewöhnlicher und partieller Differenzialgleichungen, Ein anwendungsorientiertes Lehrbuch für Ingenieure*, 4. Auflage. Springer Vieweg, 2019.
- [10] Hengguang Li. *Finite element analysis for the axisymmetric Laplace operator on polygonal domains*, *Journal of Computational and Applied Mathematics*. Elsevier, 2011.
- [11] *Transient Heat Transfer in a Partially Cooled Cylindrical Rod*, *Transactions of the ASME Vol. 131*. ASME Digital Collection, 2009.
- [12] Christian G. Sorger. *Generierung von Netzen für Finite Elemente hoher Ordnung in zwei und drei Raumdimensionen*. Technische Universität München, Lehrstuhl für Computation in Engineering, 2012.
- [13] Wolf Dieter Pietruszka. *MATLAB und Simulink in der Ingenieurpraxis, Modellbildung, Berechnung und Simulation*, 4. Auflage. Springer Vieweg, 2014.

- [14] Jichun Li. *Computational Partial Differential Equations Using MATLAB*. CRC Press, 2009.
- [15] P.I. Kattan. *MATLAB Guide to Finite Elements, An Interactive Approach*. Springer, 2003.
- [16] Norbert Heiderich. *Technische Probleme lösen mit C/C++, Von der Analysis bis zur Dokumentation, 4. Auflage*. Hanser, 2020.
- [17] Michael McLaughlin. *C++ Succinctly*. Syncfusion Inc., 2012.
- [18] Stefan Kuhlins. *Die C++ Standardbibliothek, Einführung und Nachschlagewerk, 4. Auflage*. Springer, 2005.
- [19] Physicists like to think that all you have to do is say 'These are the conditions now what happens next?'. *Richard Feynman*. The Character of Physical Law, 1965.
- [20] Not only is the Universe stranger than we think it is stranger than we can think. *Werner Heisenberg*. Across the Frontiers, 1972.

A. Test Source code Visual C++

Listing 1: For loop to print numbers from 1 to 10

```
1 // Print numbers from 1 to 10
2 #include <stdio.h>
3 int main() {
4     int i;
5     for (i = 1; i < 11; ++i)
6     {
7         printf("%d_", i);
8     }
9     return 0;
10 }
```

B. Source code MatLab

TODO