Guide for Treatment Planning

Table of Contents

[**Overview** 2](#_Toc488409859)

[**Programs** 2](#_Toc488409860)

[Optimization 2](#_Toc488409861)

[Temperature 2](#_Toc488409862)

[Windows 2](#_Toc488409863)

[Mac 3](#_Toc488409864)

[Linux 3](#_Toc488409865)

[**Folder System** 3](#_Toc488409866)

[General\_data 4](#_Toc488409867)

[Information 4](#_Toc488409868)

[Treatment\_plan 4](#_Toc488409869)

[**E-fields and Octrees** 5](#_Toc488409870)

[Octrees 5](#_Toc488409871)

[**Optimization** 6](#_Toc488409872)

[EF\_optmization 6](#_Toc488409873)

[Goal Functions 6](#_Toc488409874)

[M1-M1 6](#_Toc488409875)

[M1-HTQ 6](#_Toc488409876)

[M2 6](#_Toc488409877)

[Particle Swarm 6](#_Toc488409878)

[Particle Swarm Size 6](#_Toc488409879)

[Max Iterations 6](#_Toc488409880)

[Max Stall Iterations 6](#_Toc488409881)

[**Temperature Conversion** 6](#_Toc488409882)

[Windows (using Ubuntu) 7](#_Toc488409883)

[Other Systems 7](#_Toc488409884)

# Overview

After a model has been created in the program CST and E-fields have been generated and exported to .mat format using matlab (one field per antenna), a treatment plan is created in four steps:

1. Optimization
   1. Input e-fields
   2. Creates a PLD matrix and settings with unscaled amplitudes, and if more than one frequency is used also time settings
2. Prepping files that are used by the temperature conversion
   1. Input boundaries, PLD and max amplitude
   2. Creates many different files that need to be moved to FEniCS
3. Temperature conversion – in FEniCS
   1. Input files from prepping
   2. Creates a temperature.h5 file and scales amplitudes according to max amplitude
4. Evaluate temperature
   1. Input temperature.h5 file and scaledAmplitudes.txt from FEniCS
   2. Creates a temperature matrix and scales the old settings according to the calculated scale

Main is a matlab-file that performs all operations. Use this for an overview of the optimization process. Most inputs find themselves in the folder system, the rest are to be input in the *Input data* window that enters the screen when the program is run. To use the results for a real treatment, the scaled settings and time settings are needed. The PLD and temperature matrices are needed for evaluation of how good the recommended treatment is.

# Programs

The following programs/add-ons are needed to perform the optimization/temperature transformation.

## Optimization

* Matlab (R2017a)
* Matlab add-on: MinGW64 (GGC version 4.9.2 by MathWorks Supported Compilers Team)
* Matlab add-on: Myslicer (version 1.1 by Anders Brun)
* Iso2mesh (Iso2mesh 2017 (v1.8), codenamed Deviled Egg)

<http://iso2mesh.sourceforge.net/cgi-bin/index.cgi/?Download>

## Temperature

### Windows

* Virtualbox (VirtualBox 5.1.20 platform packages, windows hosts)  
  <https://www.virtualbox.org/wiki/Downloads>
* Virtualbox: Ubuntu (give at least 15 MB space)  
  <https://www.youtube.com/watch?v=QkJmahizwO4>
* Ubuntu: FEniCS

Type following commands in Ubuntu terminal to install

sudo add-apt-repository ppa:fenics-packages/fenics  
 sudo apt-get update  
 sudo apt-get install --no-install-recommends fenics  
 sudo apt-get dist-upgrade

### Mac

* Anaconda (Anaconda 1.6.2)
  + Dolfin (Dolfin 2017.1.0) should come with Anaconda
* Python (Python 2.7.13)
* Anaconda Navigator: FEniCS  
  To activate FEniCS type following commands in normal mac terminal  
   conda create -n fenicsproject -c conda-forge fenics  
   source activate fenicsproject
* Install h5py  
  Type following command in terminal found in Anaconda Navigator > Environments > fenicsproject > Open Terminal:  
   pip install h5py

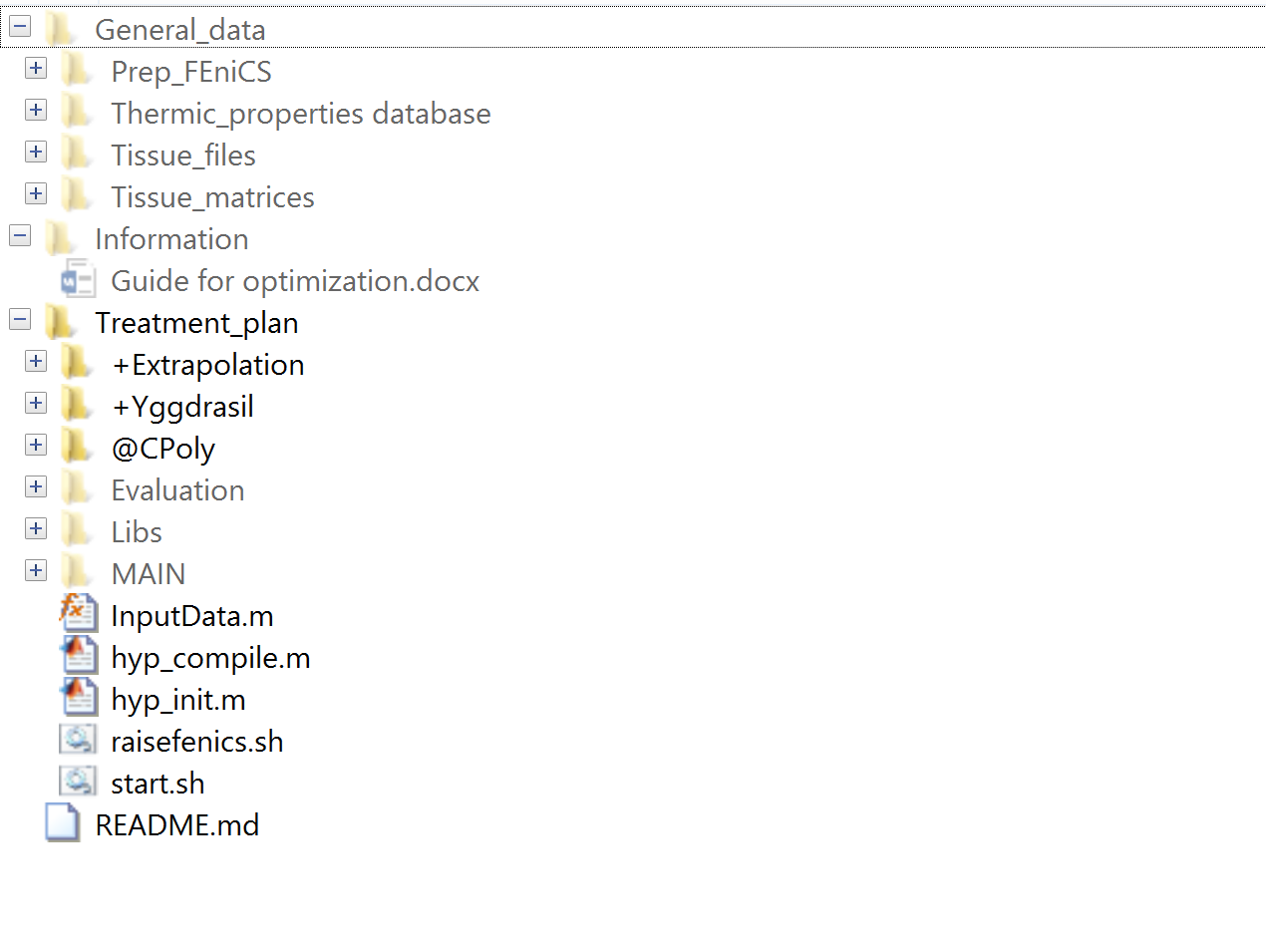
### Linux

We have not used Linux, but doing so should be easier since FEniCS is more compatible with the system.

# Folder System

The folder system used for the hyperthermia treatment planning process is explained in this section.

The first folder encountered is the project folder *Hyperthermia\_Treatment\_Plan*. This folder contains the subfolders *General\_data*, *Information* and *Treatment\_plan, see image below*.



## General\_data

General data is the library used to store the different data for all models. The data is stored in descriptive folders. The data needed to perform a treatment plan is collected here and moved to the data folder used by Main, see *Treatment\_plan* below.

This folder currently contains files used for temperature conversion in FEniCS, a database (excel-file) for generating new thermic properties in preparation for temperature conversions, tissue files for different models, as well as tissue matrices.

## Information

This folder contains information about how to use the Treatment planning system, the different programs that are needed to do so and some results produced by the planning system.

In this folder there is also information about the correct file names of the data and how to quickly change them if they are wrong.

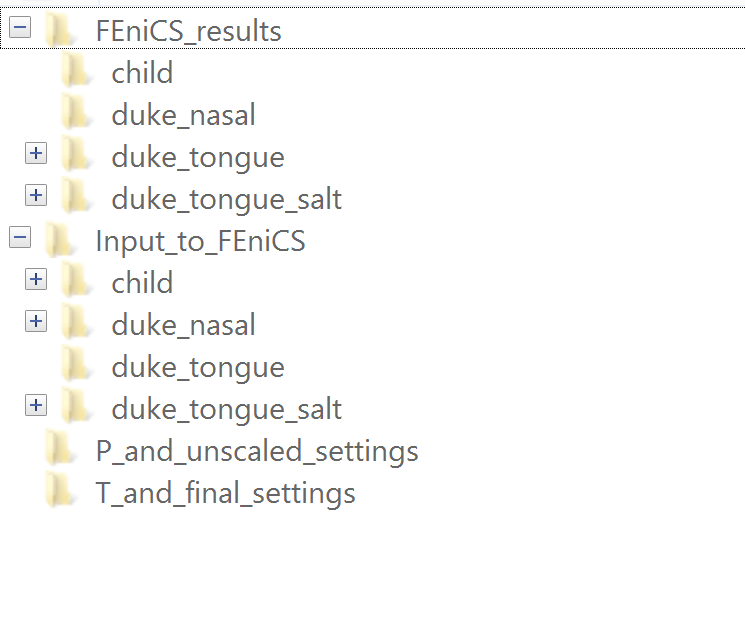
## Treatment\_plan

This is the folder used to perform the treatment planning. The subfolders *+Extrapolation, +Yggdrasil, @CPoly* contain different functions that are needed for creating octrees, octree operations, polynomial calculations and functions for loading files. *Evaluation* contains scripts used to evaluate the treatment plan after and during optimization and to compare different results. It has the subfolders *quality\_indicators* and *visualize* that contain different scripts that calculate HTQ, TC, or visualize different values in histograms/graphs. *Libs* contains *iso2mesh* and *myslicer.* Iso2mesh is vital for creating a mesh that is used in the temperature conversion and myslicer is used to plot the results in matlab.

The folder *MAIN* is the folder that will be actively used during the optimization and temperature conversion. This is also where the *Main* script can be found. The Main script is the script that calls on all functions necessary to perform the optimization, preparation for FEniCS and matrix conversion after temperature calculations. Run this script to input your variables! *MAIN* has the subfolders *Data, Scripts* and *Results.* Data is the folder where the data needed to run the current calculations is added. The data needed for a complete run is:

* E-fields or octrees for frequency(ies) of interest
* Tissue-matrix
* Tissue-file
* Boundary condition
* Temperature
* Thermal compilation or thermal\_db\_index\_to\_mat\_index

More specific information about these files can be found in a README in the data folder. *Scripts* contains scripts that are used for different parts of the planning, split into descriptive folders. *Results* is the folder where all generated results are saved, see image below, and contains *FEniCS\_results, Input\_to\_FEniCS, P\_and\_unscaled\_settings* and *T\_and\_final\_settings*.



*P\_and\_unscaled\_settings* is where P-matrices (PLDs) and settings with unscaled amplitudes are saved. When only one frequency is used, optimized E-fields are also saved. These can be used in the *combine\_single* script, see section *Optimization* below. Do not change the format of the settings file, since this format is compatible with *Labview* (used to apply the settings in the lab). *Input\_to\_FEniCS* (with subfolders named after the models)is where files that need to be input to FEniCS are saved and FEniCS\_results is where the user inputs the results after temperature conversions in FEniCS. The files *temperature.h5* and *scaledAmplitudes.txt* should be here, placed in a subfolder named after the model. *T\_and\_final\_settings* contains the temperature matrix created from the *temperature.h5* and the settings with scaled amplitudes. These amplitudes are the ones that should be used for the treatment. Do not change the format of the settings file, since this format is compatible with *Labview* (used to apply the settings in the lab).

*Treatment\_plan* also contains *hyp\_compile, hyp\_init, InputData, raisefenics* and *start.* The first three scripts compiles the c-code, adds paths and opens the input-data dialogue when running *Main*. *raisefenics* and *start* can be used if the user uses eg. Docker to run FEniCS, and in that case help to open and run FEniCS automatically. These files have not been used in combination with Ubuntu but are saved for future use.

# E-fields and Octrees

The program CST microwave studios (2014) is used to simulate the E-fields, that are then exported to matlab using the function *ExportEfields4D* that can be found on NewHopa. These E-fields are then converted to Octrees (to save space) in the optimization. When they have been converted to Octrees, the original E-fields are no longer needed for the optimization.

## Octrees

Creating octree structures from normal matrices is a way to save memory and time. The matrices are then compressed to a given limit and the octrees contain less elements than the original matrices. This means that they take up a lot less memory. The octrees for each frequency are weighted with their corresponding sigma.

When the octrees are opened in the optimization, they are opened as class SF-Efield. These contain the complex settings and their corresponding antennas. From the optimized E-fields (e\_tot\_opt), these are found by entering:

complex\_settings = e\_tot\_opt.C.values  
corresponding\_antenna = e\_tot\_opt.C.keys

# Optimization

## EF\_optmization

*EF\_optimization* exists in two versions – single and double. Single is for one frequency and double is for two. The appropriate function is called on automatically when using the Main script.

The function is responsible for loading E-fields, converting them to Octrees, creating a sigma matrix for each frequency, optimizing the E-fields, creating the P-matrix and unscaled settings. If more than one frequency is used, EF\_optimization also finds the best combination and time settings for the E-fields. The Octrees and sigma matrix can be found in the data folder since they are not needed directly by the user, and the P-matrix and unscaled settings-file can be found in Results>P\_and\_unscaled\_settings.

## Goal Functions

There are currently three different options for the goal function of the optimization. The user can choose between M1-M1, M1-HTQ and M2.

### M1-M1

The goal function M1 is used and the result is evaluated with M1.

### M1-HTQ

The goal function M1 is used and the result is evaluated with HTQ.

### M2

The goal function M2 is used and the result is evaluated with M2.

## Particle Swarm

Particle swarm is used as the optimization method. A swarm of particles move across the surface and communicate with each other to try to find the optimal solution. Different options can be chosen for the method and are input in the data window that appears when Main is run. With the settings recommended below, an optimization with single takes approximately 1.5 hours.

### Particle Swarm Size

This is the number of particles that are used by particle swarm during the optimization. Recommended size is 20 particles, but this could be further studied.

### Max Iterations

This is the number of iterations that particle swarm perform during the optimization. Recommended amount is 20 iterations, but this could be further studied.

### Max Stall Iterations

This is the number of iterations that particle swarm perform during the optimization. It should be set to at least 10.

## Combine single

A quick way to combine two or more frequencies is to use *combine\_single*, a script where optimized E-fields from EF\_optimization\_single are input and the best time share to combine them is calculated. The E-fields of each frequency are not optimized based on the others, but it is a very fast way to determine whether it could give good results.

# Temperature Conversion

Important: In pennes.py, which is the script that is run in FEniCS, most things are automatic. There is one thing that always need to be checked though: Tmax! Depending on if the used model is head and neck or brain, Tmax varies. If you use a head and neck model, it should be set to 8 and if you use a brain model it should be set to 5.

This is performed in FEniCS. This can be done in different ways depending on what system you are using. The files in Scripts > FEniCS and Results > Input\_to\_FEniCS need to be accessible for the program. Everything in this document should be downloaded and available on NewHopa. To open ubuntu on NewHopa; open virtualbox, start Ubuntu and log in using the password “hej!1234”. Open gnome control center to log in to your drive-account (for guide on this, see link “howtogeek” below).

## Windows (using Ubuntu)

Ubuntu and FEniCS need to be installed. It is also very helpful to install google drive on Ubuntu since it facilitates the copying of data between Ubuntu and Windows. For instructions on how to do this, see <https://www.howtogeek.com/196635/an-official-google-drive-for-linux-is-here-sort-of-maybe-this-is-all-well-ever-get/>.

We used the folders: Input\_to\_FEniCS, FEniCS and FEniCS\_results in Downloads on Ubuntu. The first time a PLD (P-matrix) is to be converted to temperature, we did:

1. Widows: copy the three folders (FEniCS\_results is empty) from Windows to a shared folder in google drive
2. Ubuntu: copy the folders from google drive on Ubuntu to Downloads
3. Ubuntu: rename P-matrix to only P.mat
4. Ubuntu: open the terminal och enter ”ls” to view current directory, then step into Downloads>Input\_to\_FEniCS using “cd”
5. Ubuntu: type ”python pennes.py” in the terminal to start the conversion
6. Ubuntu: copy the created files in Downloads>FEniCS\_results to google drive

Move the files to the results folder Results > FEniCS\_results in windows and keep running Main to plot the result using myslicer and to convert the .h5 file to a .mat. It is also possible to use paraview to view the temperature, but then the code that creates the .pvd and .vtu files needs to be not-commented in pennes.py.

After the process has been performed once and a model and a different frequency is to be used, only the P-matrix, amplitudes.txt (and ampLimit.txt if this has changed) need to be switched for new versions since the remaining files will be the same.

## Other Systems

Open the terminal used to run FEniCS and initiate the FEniCS project. Find the correct folder and make sure all the files you need to run pennes.py are there. The PLD-matrix should only be named P.mat. To run the script, enter “ python pennes.py “.