



**Politecnico
di Torino**

Master's Degree Course in Biomedical Engineering
December 2022

**Temperature Map Reconstruction in
Head and Neck Cancer Hyperthermia**

Advisor

Prof. Giuseppe Vecchi

Candidate

Diego Verdini
s279408

Co-Advisors

Dr. Rossella Gaffoglio
Dr. Giorgio Giordanengo

To my parents and family

Abstract

Hyperthermia treatment (HT) is a thermal treatment which uses non-ionizing electromagnetic radiation, like microwaves, to rise the temperature in the tumor region (where cells reach 42/44°C) creating tissue damage. Recent studies show that HT can sensitize cancer cells to both chemotherapy and radiotherapy treatments, increasing their efficacy without providing new dose exposition.

During HT, it is crucial to limit the heating process only in the tumor site while maintaining the temperature values in the surrounding healthy tissues around the physiological level. For internal cancers, phased antenna arrays are properly designed to focus the electromagnetic radiation on the tumor region, avoiding hotspots, i.e., dangerous overheated regions far from the tumor site, which could be harmful especially in the treatment of head and neck (H&N) cancers. In the clinical practice, a water-bolus (plastic bag filled with cold water) is introduced between the antenna applicator and the patient to avoid overheating of the skin, where temperature must remain tolerable, and improve the radiation coupling into the body.

Good clinical outcomes are achieved using a patient-specific treatment planning, where the antenna feedings are optimized by means of numerical simulations using patient-specific phantoms derived from MRI and CT scans of the patient. After optimizing the power deposition on the tumor target, thermal solvers are used to generate the temperature map distribution in the numerical phantom. Although the solvers produce high valuable results, they are often not reliable per se. The dielectric and thermal parameters assigned to the different tissues of the phantom are currently known with high uncertainty. These values, reported in the Literature and derived from ex-vivo measurements, can introduce inaccuracies in the simulations, leading to incorrect temperature maps of the patient. For this reason, during HT treatments it is essential to quantify the temperature in the heated tissues with invasive temperature probes (inserted into closed-tip catheters), which cause great discomfort to the patient and provide limited spatial information.

The goal of the present Master Thesis is to validate and improve the accuracy of the simulated 3D temperature map from few known temperature values during HT treatments in the H&N cancer scenario. A realistic numerical phantom is used for the thesis purpose; then, via the Sim4Life software and Python and MATLAB scripts, simulations are performed to maximize the specific absorption rate (SAR) on the tumor target and to generate the corresponding temperature map. Assuming to know few temperature values (along the direction of a catheter) of different target maps corresponding to random combinations of the tissue parameters, a numerical reconstruction method is applied to retrieve the target maps in the whole region of

interest. This method employs high-performance simulations of the temperature map (to be performed prior to treatment). An in-depth analysis is carried out to estimate the number of such simulations needed to ensure a sufficiently good reconstruction, for different scenarios, i.e., when a different number of parameters of a given number of tissues are supposed to be different from their baseline values. Finally, the quality of the reconstruction is also discussed for different directions of insertion of the catheter in the region of interest.

Acknowledgements

I would like to express my sincere gratitude to Prof. Giuseppe Vecchi for giving me the opportunity to work on this thesis project, as well as to my co-advisors Dr.ssa Rossella Gaffoglio and Dr. Giorgio Giordanengo for their assistance. Their support and endless patience throughout the work allowed me to complete this thesis, and I am grateful to them.

Finally, I would like to thank all those who have been close to me during this milestone: colleagues and friends with whom I have shared part of this journey and who, in their own way, have contributed to make it more enjoyable.

Table of Contents

List of Tables	VI
List of Figures	VII
Acronyms	X
Introduction	1
1 Head and Neck Cancer Therapies	3
1.1 Overview and General Attributes	3
1.2 HNC Treatments	4
1.2.1 Radiotherapy	4
1.2.2 Chemotherapy	4
1.3 Hyperthermia Therapy	4
1.3.1 State of the Art	5
1.3.2 Hyperthermia Treatment Planning	7
2 Numerical Simulation	9
2.1 Equations and Parameters Involved	9
2.1.1 Electromagnetic Modeling	9
2.1.2 Thermal Modeling	11
2.1.3 Tissue Parameters	13
2.2 Sim4Life Software	14
2.2.1 Anthropomorphic Phantom	15
2.2.2 Antenna Array	16
2.3 HTQ SAR-based Optimization	18
2.3.1 Phase and Phase-Amplitude Optimization	19
2.3.2 Optimized SAR Maps	22
2.3.3 Temperature Maps	24

3	Temperature Maps Reconstruction	27
3.1	Multigrid Creation	27
3.1.1	Multigrid A + Baseline	28
3.1.2	Multigrid B + Baseline	29
3.2	Svar Matrix Creation	30
3.2.1	Map Extraction via Python Scripts	30
3.2.2	Temperature Map Reconstruction	34
3.2.3	Variation of Catheter Insertion	37
4	Reconstruction Fidelity Evaluation	42
4.1	Multigrid Type Comparison	42
4.2	Svar matrix Dimension	53
4.2.1	Thermal Parameters Only	53
4.2.2	Dielectric Parameters	64
4.3	Discussion and Temperature Reconstruction	70
	Conclusions	76
	MATLAB Script	78
	Python Script	86
	Bibliography	95

List of Tables

2.1	Yoon-sun Phantom Characteristics	15
2.2	Tumor Model Dimensions	16
2.3	Patch Antenna Dimensions	17
2.4	Phase-Only Optimization	22
2.5	Phase-Amplitude Optimization	22
3.1	Parameters Literature Values [32]	28
3.2	MGrid Type A + Baseline Values, k and ω of <i>muscle</i> and <i>tumor</i> . .	34
4.1	Multigrids Values, k and ω of <i>muscle</i> and <i>tumor</i>	43
4.2	MGrid Type B + Baseline σ and ω of <i>muscle</i> and <i>tumor</i>	65
4.3	MGrid Type B + Baseline ϵ_r and ω of <i>muscle</i> and <i>tumor</i>	68

List of Figures

1.1	Head and Neck Cancer Regions	3
1.2	HYPERcollar applicator for HT in H&N region. [4]	7
1.3	H&N HTP workflow. [21]	8
2.1	Yoon-sun Phantom	15
2.2	Tumor	16
2.3	Tumor Location	16
2.4	Array of Antennas	16
2.5	Patch Antenna	17
2.6	S_{11} of the 8 th antenna	17
2.7	Domain Voxels	18
2.8	Sim4Life viewer, xy plane at 20W power	19
2.9	PSO Comparison Phase-Only and Phase-Amplitude Optimization	22
2.10	SAR distribution in different cases, plane xy	23
2.11	Temperature Maps at 20W	24
2.12	Temperature Maps at 40W	25
2.13	Temperature Map Phase-Amplitude Optimization at 60W	26
2.14	Temperature Map Phase-Amplitude Optimization at 80W	26
3.1	MGrid type A with N=3	29
3.2	MGrid type B with N=3	29
3.3	Tmap 5	32
3.4	Tmap 25	33
3.5	Case 1	38
3.6	Case 2	39
3.7	Case 3	40
3.8	Case 4	41
4.1	Case 1	44
4.2	Case 1	45
4.3	Case 2	46

4.4	Case 2	47
4.5	Case 3	48
4.6	Case 3	49
4.7	Case 4	50
4.8	Case 4	51
4.9	4 Cases Comparison	52
4.10	$N_{p,min}$ Statistics Comparison varying k and ω of <i>muscle</i> and <i>tumor</i>	54
4.11	MGrid Type B + Baseline. The limits on the y -axis are set to show more clearly the values reached.	55
4.12	$N_{p,min}$ Statistics Comparison varying k and ω of <i>muscle</i> and <i>tumor</i>	56
4.13	MGrid Type B Without Baseline. The limits on the y -axis are set to show more clearly the values reached.	57
4.14	T0 Map Visualization	59
4.15	$N_{p,min}$ Statistics Comparison varying k and ω of <i>muscle</i> , <i>tumor</i> and <i>fat + SAT</i>	60
4.16	MGrid Type B + Baseline. The limits on the y -axis are set to show more clearly the values reached.	61
4.17	$N_{p,min}$ Statistics Comparison varying k and ω of <i>muscle</i> , <i>tumor</i> , <i>fat + SAT</i> and <i>skin</i>	63
4.18	MGrid Type B + Baseline. The limits on the y -axis are set to show more clearly the values reached.	63
4.19	$N_{p,min}$ Statistics Comparison varying σ and ω of <i>muscle</i> and <i>tumor</i>	67
4.20	$N_{p,min}$ Statistics Comparison varying ϵ_r and ω of <i>muscle</i> and <i>tumor</i>	69
4.21	$N_{p,min}$ Statistics Comparison MGrid Type A + Baseline with Different Dimensions of the Parameters Space.	72
4.22	Comparison Between The Reconstructed Map with $\Delta T_{max}=89.51$ °C, Obtained Using the MGrid Type A and $N_{p,min}=24$, and The Corresponding Target Map	73
4.23	Comparison Between The Reconstructed Map with $\Delta T_{max}=0.09$ °C, Obtained Using the MGrid Type B and $N_{p,min}=24$, and The Corresponding Target Map	74
4.24	Comparison Between The Reconstructed Map with $\Delta T_{max}=0.48$ °C, Obtained Using the MGrid Type B and $N_{p,min}=16$, and The Corresponding Target Map	75

Acronyms

HNC	Head and Neck Cancers
H&N	Head and Neck
RT	Radiotherapy
IMRT	Intensity-Mediated Therapy
VMAT	Volumetric Modulated Therapy
RF	Radiofrequency
MW	Microwave
IR	Infrared
CEM	Cumulative Equivalent Minutes
HT	Hyperthermia Treatment
EM	Electromagnetic
SAR	Specific Absorption Rate
HTP	Hyperthermia Treatment Planning
FDTD	Finite Difference Time Domain
CT	Computed Tomography
MRI	Magnetic Resonance Imaging
ViP	Virtual Population
HTQ	Hotspot to Tumor Quotient
PSO	Particle Swarm Optimization
SAT	Subcutaneous Adipose Tissue

Introduction

During thermal cancer therapies, the temperature within the tumor is altered in order to cause cell death. The majority of cancer cases cannot be successfully treated with one cancer treatment alone due to the complexity of the disease. Thus, combining traditional treatments, like radiotherapy (RT) and chemotherapy, with thermal therapies could improve the effectiveness.

In combination with radiotherapy or chemotherapy, hyperthermia treatment (HT) has shown good clinical result [1, 2, 3]. In this context, HT acts as an enhancer mechanism of action ensuring the same clinical outcome with a lower dose of ionizing radiation in RT or drugs in chemotherapy. Hyperthermia treatment aims to increase the temperature in the tumor region at 42/44 °C to damage the cells, by exploiting the energy provided by an electromagnetic applicator. The applicator in head and neck cancers (HNC) usually consists of multiple antennas placed equidistantly around the patient's neck inside a structure having the shape of a collar [4]. As part of the latter, the waterbolus system formed by bags filled with water is also included, with the function to avoid overheating of the skin and favor the electromagnetic coupling into the body.

Electromagnetic energy interacts with the biological tissue and causes temperature rise due to its dissipation through the human body. Different frequencies allow distinct penetration depths inside the human body, so it is crucial to set the working frequency correctly. Head and neck cancers are usually treated with microwave frequencies due to the physiology of the region. It is essential to optimize the energy provided and focus it on the tumor without damaging sensitive surrounding tissues, thus avoiding hotspots, and creating customized treatments for each patient to improve the effectiveness of the therapy. The optimization process involves the antenna feedings which need to be properly modified, acting on the phases and amplitudes of each antenna. Hyperthermia treatment planning (HTP) is the process that allows to customize the therapy and make it patient-specific. HTP starts with the acquisition of patient CT scans by which the different tissues are segmented, creating a virtual model where the treatment can be simulated. Firstly, Maxwell's equations are solved, describing the interaction between tissues and the EM energy provided by the antenna applicator in the region of interest. From those

outcomes, the thermal solver is employed to solve the bio-heat equation generating the related temperature map distribution.

Solvers provide very valuable results but are often unreliable by themselves. The dielectric and thermal parameters associated to the different tissues of the phantom, which are known in Literature with high uncertainty, affect the simulation outcomes and so the heat map distribution may differ from the real patient situation. Hence, in HT treatments, invasive temperature probes (inserted into closed-tip catheters) are necessary for measuring the temperature inside the heated tissues. In this thesis, numerical algorithms aimed at reconstructing the real temperature map in the whole region of interest starting from few known temperature values (the ones along the catheter) are analyzed to improve their performance in terms of results and computational time.

Thesis Overview

This thesis contains an extensive analysis of all the mentioned topics, organized into four chapters and two appendices as follows.

Chapter 1 illustrates the common head and neck cancer therapies outlining their principal aspects. Hyperthermia therapy is emphasized, analyzing its treatment planning and related issues.

In Chapter 2, the equations underlying the HT mechanism are introduced, together with the chosen phantom and tumor model involved inside the simulation. The EM solver is run and the SAR-optimization process is performed modifying phases and amplitudes of each antenna to maximize the temperature rising inside the tumor region. After the optimization, the temperature maps are produced.

Chapter 3 presents the multigrids generation which are the structures used to create several combinations of parameter values varying within the Literature range. The Python script that assigns the different sets to the phantom tissues and generates the corresponding temperature maps is also presented. Finally, the reconstruction algorithm performed in MATLAB is introduced and applied for different directions of insertion of the catheter's points used in the reconstruction.

In Chapter 4, the quality of the reconstruction is evaluated for different choices of the multigrids and a different number of parameters and tissues involved.

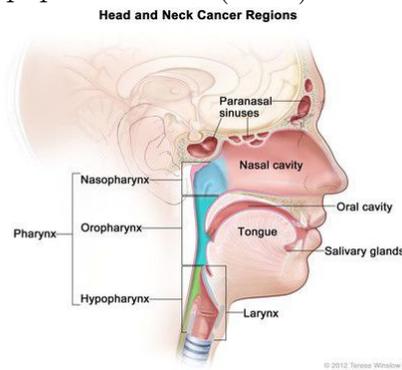
Chapter 1

Head and Neck Cancer Therapies

1.1 Overview and General Attributes

Cancer is a complex, multifunctional disease that begins with mutations at a single-cell level followed by the combination of several sequential mutations. The mutations can alter the normal mechanisms that regulate cell proliferation leading to loss of apoptotic control: cells grow uncontrollably and spread to other parts of the body.

Cancers that are known collectively as head and neck cancers (HNC) usually begin in the squamous cells (squamous cell carcinomas) that line the mucosal surfaces of the head and neck (i.e., oral cavity, pharynx, larynx, paranasal sinuses and nasal cavities, tongue, and salivary glands) [5, 6, 7]. HNC incidence is around 900,000 cases worldwide and almost half a million deaths yearly. Bad habits like smoking and alcohol consumption represent one of the major risk factors, together with viruses as the human papillomavirus (HPV) infection [8].



© 2012 Terese Winslow LLC,
U.S. Govt. has certain

Figure 1.1: Head and Neck Cancer Regions

1.2 HNC Treatments

In head and neck (H&N) regions, cancer treatments can involve surgery (physically removing tumor mass from the patient's body), immunotherapy (helping the patient's immune system fight cancer), radiotherapy (RT), chemotherapy, or a combination of them. The treatment plan for an individual patient depends on several factors including the tumor location, the stage of cancer, and the patient's overall health [7]. HNC are the specific disorder involved in the present thesis.

1.2.1 Radiotherapy

H. Coutard, almost 100 years ago, was the first to use X-rays for laryngeal cancer treatment [9], since then radiation therapy (RT) has been key for the treatment of these tumors. RT is a cancer treatment that uses high doses of radiation to shrink tumors, kill cancer cells or slow their growth by damaging their DNA. In order to reduce tissue toxicity next to cancer, new advanced radiation techniques are currently employed. Intensity-modulated radiation therapy (IMRT) and volumetric-modulated therapy (VMAT) are widely used for HNC [10].

1.2.2 Chemotherapy

Chemotherapy is a treatment that uses drugs to stop the growth of cancer cells, either by killing the cells or by stopping them from dividing [11]. This therapy, usually combined with other treatments, typically occurs in stages III and IV of HNC where the latest standard regimes [12], i.e., docetaxel, cisplatin, and fluorouracil (TPF), have been accepted. However, in some cases, those drugs may fail due to chemotherapy resistance; innovative therapeutic strategies should be studied [13].

1.3 Hyperthermia Therapy

Over the most known tumor therapies, like RT and chemotherapy, also thermal cancer therapies have progressed during the last decades thanks to new research. This improvement, even in clinical trials, is linked to the technological development in the radiation techniques on which these processes are based. The energy delivered comes from electromagnetic-based technologies which are divided, considering the frequency applied during the treatment, into radiofrequency (RF), microwaves (MW), and infrared (IR) [14]. The concept, behind these therapies for cancer treatment, is the temperature rising in the tumor area, above the physiological values, creating cell damage. The beneficial effects, however, depend on the thermal dose exposition (i.e., the temperature reached and accumulative time exposition) in

the region of interest which is given in $CEM_{43^{\circ}C}$ (cumulative equivalent minutes), where $43^{\circ}C$ is target value based on the Arrhenius relationship.

Hyperthermia treatment (HT) aims at inducing a temperature increase, through energy delivery, in a range of $42/44^{\circ}C$ for nearly one hour, triggering changes in the immune system, oxidation, and vasodilatation. Blood perfusion and vascular permeability play a significant role in heat dissipation inside the tissues. The tumor microenvironment is characterized by a heterogeneous structure, with a high number of proliferative cells, creating a random and chaotic distribution. This structure led to different nutrients concentration, forms hypoxic areas, and increases tumor interstitial fluid pressure. For all these reasons, heat dissipation in the tumor area is dissimilar to healthy tissues and aids treatment success. According to some research [15, 1, 2], due to these biological modifications in the tumor region, HT can improve the effectiveness of radiotherapy and chemotherapy at the same dosage, hence limiting the patient dose exposition.

1.3.1 State of the Art

Paramount settings are related to the EM-based mechanism, which enhances the temperature increase inside the body. The operating frequency is a critical parameter, it determines the interaction with the tissues hence the achievable penetration depth. During the MW energy application, all tissues of the body behave as lossy dielectrics (i.e., poor insulators and poor electrical conductors) prevalently with a frequency above 100 MHz [14]. The microwave frequency range is between 100 MHz and 10 GHz, and is characterized by non-ionizing radiation capable to prevent health risks. Moreover, HT MW is immediate, and uniform compared to the previous methods.

Generally, the higher frequencies in MW localize heating on superficial tissues like skin, while the lower ones are involved in larger and deeper areas. Power deposition decreases as microwaves penetrate through the tissue. HT can be divided into Superficial hyperthermia or Deep hyperthermia according to the distance between the applicator, made by antennas, and the tumor region and also considering the target volume size [14]:

- **Superficial HT:** heat limited to a volume of tissue, i.e., skin and outer tissue, which is facing the heating device. The applicator can be single or multi-antenna according to the dimensions of the treated area. Superficial HT is usually employed for the treatment of large regions.
- **Deep HT:** the heating concentration is optimized in the tumor region, placed internally to the human body, limiting the surrounding healthy tissues from irradiation. This is achievable with a phased-array approach, placing an array

of antennas around the anatomical region of interest, and optimizing their amplitudes and phases.

For both these two techniques, the thermal penetration it is around 2-4 cm obtained with a frequency in a range from 400 MHz to 1 GHz. Waterbolus, bags filled with deionized water at a controlled temperature [16], are placed between the sources and the patient's surface to provide and improve the applicator-body EM coupling and to prevent skin from possible burns. Acting on the temperature of the circulating water inside the waterbolus, it is possible to modify the depth penetration.

The mentioned cooling system, and the radiation area optimization, become crucial in anatomical sites like the head and neck (H&N) region where the structures are extremely sensitive to temperature rising. Thanks to their previous research, in antenna design for H&N regions, M.M. Paulides et al., have realized their own applicator called HYPERcollar [4]. The HYPERcollar consists of 12 probe-fed patch antennas divided into 2 circular arrays, spaced 6 cm, creating a cylinder positioned around the neck of the patient. The antennas work, with independent phase shifts and amplitude, at 434 MHz which is the frequency clinically validated for head and neck region [16]. The waterbolus system completes the applicator, cooling the skin and coupling the EM field into the body. In this system, the feedings of the antennas are properly optimized to maximize the Specific Absorption Rate (SAR) in the tumor region, minimizing the risk of overheatings (hotspots) in the surrounding healthy tissues. It has been demonstrated that is the most effective way to optimize properly the antenna feedings to target the tumor region. This is related to its short computational time, the possibility to easily control its performance, and the correlation between SAR indicators and clinical result [17]. However, the effect of thermal boundary conditions could affect the outcomes and form incongruities between SAR distributions and temperature. This can be associated to the external cooling (i.e., the waterbolus) and physiology processes like airflow able to modify the related temperature maps [18]. The other optimization mechanism is the Temperature-based which works directly on the temperature distribution. This approach, in contrast with the SAR-based, is characterized by high computational costs due to the solution of the thermal equation for every step. Additionally, the results could be not reliable due to the uncertainty characterizing tissue thermal parameters, which are known only with ex-vivo measurements.

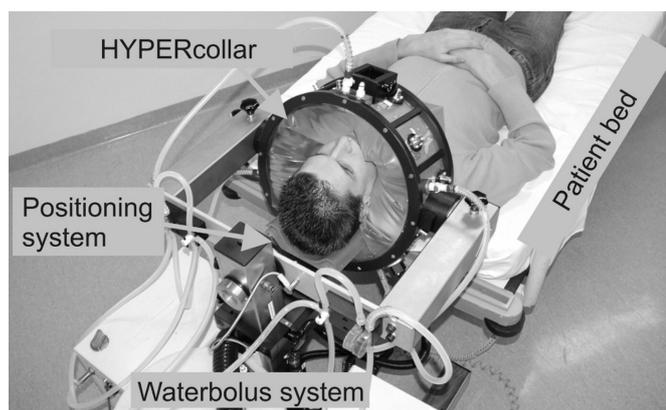


Figure 1.2: HYPERcollar applicator for HT in H&N region. [4]

1.3.2 Hyperthermia Treatment Planning

Starting from retrospective verification of clinical therapy capabilities, hyperthermia treatment planning (HTP) is used nowadays as the standard process to improve clinical outcomes [17] (i.e., Quality Assurance Guidelines for clinical application of locoregional hyperthermia [19]). The HTP allows the creation of a patient-specific treatment and the radiation focalization only on the tumor region avoiding hotspots, i.e., dangerous overheated regions far from the tumor site. The typical HTP workflow for the SAR-based techniques, on which this work is focused, can be divided into the following steps [20]:

1. Patient CT or/and MRI scan in the therapy position, whose images allow the reconstruction of the 3D patient model improving the specificity of the treatment.
2. From the previous slices tissue segmentation is performed, in a manual or semi-automatic way, generating the model. This step influences simulation results due to the association of electrical and thermal properties to every tissue. The model of the antenna applicator is included in the simulation and Maxwell's equations are solved using an EM software. Different techniques can be used like the finite difference time domain method (FDTD), the finite integration technique (FID), or others.
3. Phase-amplitude optimization is the critical stage to optimize tumor heating while limiting the healthy tissues. SAR-based techniques evaluate the SAR distribution and modify the applicator feedings to maximize it on the tumor region avoiding the other tissues.

4. Finally, the thermal solver provides 3D temperature map distribution, corresponding to the optimized SAR, in the patient model solving the bio-heat equation. This step considers the heat removal by the blood flow, however, large uncertainties are present.

H&N HTP: workflow

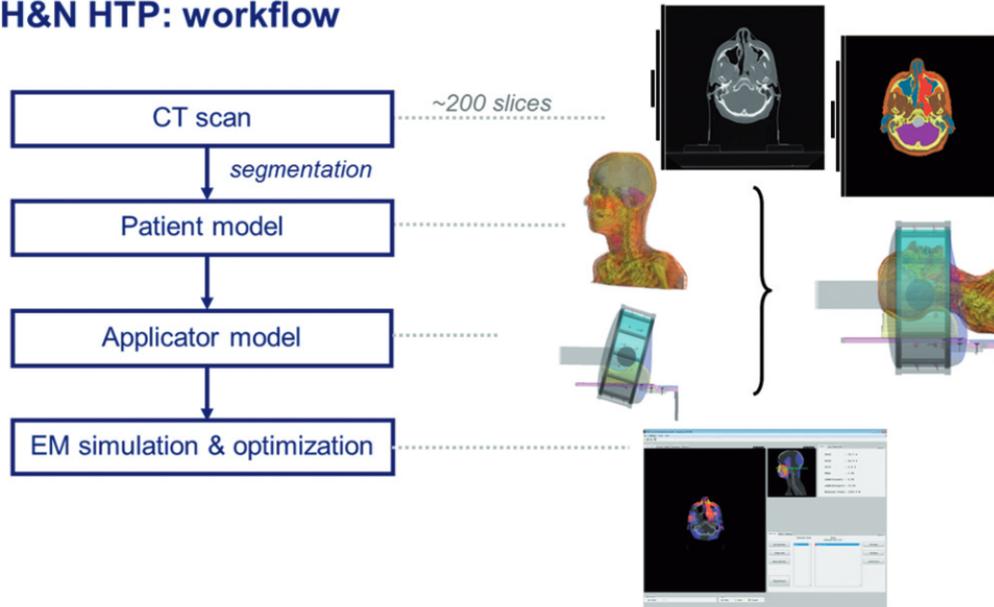


Figure 1.3: H&N HTP workflow. [21]

Chapter 2

Numerical Simulation

The importance of creating a patient-specific hyperthermia treatment is crucial to enhance the efficacy of the therapy. As mentioned before, the electromagnetic applicator transfers energy inside the body and increases the temperature thanks to its interaction with tissues. The numerical simulation allows to reproduce the interaction between the radiation, produced by the antennas, and the patient tissue. Thanks to the electromagnetic solver and thermal solver it is possible to evaluate the treatment and further improved it by optimizing the heating process only on the tumor region.

2.1 Equations and Parameters Involved

2.1.1 Electromagnetic Modeling

The electromagnetic solver generates results solving Maxwell's equations, which describe the interaction between the magnetic field and the electric field, but also how microwave energy propagates into the patient tissue [22].

$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t} \quad (2.1)$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad (2.2)$$

$$\nabla \cdot \mathbf{D} = \rho \quad (2.3)$$

$$\nabla \cdot \mathbf{B} = 0 \quad (2.4)$$

The (2.1) is the “Maxwell-Ampere Equation” and describes how the conduction, current density \mathbf{J} [A/m²] and the electric flux density \mathbf{D} [A s m⁻²], as well as

displacement current $\partial\mathbf{D}/\partial t$ [A/m²], influence the magnetic field \mathbf{H} [A/m].

The (2.2) is the ‘‘Maxwell-Faraday Equation’’, it states that changes in the magnetic field \mathbf{B} [V s m⁻²] affect the electric field strength \mathbf{E} [V/m]. The Gauss’ Law for the electric field (2.3) enounces that the electrical charge density is the source of the electric field, while the (2.4) is the Gauss’ Law for the magnetic field and defines that magnetic monopoles do not exist.

Dielectric losses convert microwave energy into heat within tissues, so modeling wave propagation is necessary [22]. The time-dependent energy balance for a volume (Ω) in which the electromagnetic field propagates is given by:

$$p_{\text{diss},\Omega}(t) + \frac{d}{dt}W_{\Omega}(t) = p_{\partial\Omega}(t) \quad (2.5)$$

where $p_{\text{diss},\Omega}$ [W] is the power dissipated, so transformed in heat, within volume. W_{Ω} is the energy stored in (Ω) and $p_{\partial\Omega}$ is the power exchanged through the volume with the outside.

The dissipation term $p_{\text{diss},\Omega}$ can be defined as:

$$\frac{p_{\text{diss}}(t)}{d\Omega} = \mathbf{E} \cdot \sigma\mathbf{E} = \sigma|\mathbf{E}(t)|^2 \quad (2.6)$$

Also the harmonic form can be introduced using the electric field’s magnitude (peak value) $|\mathbf{E}|$ [V/m]:

$$\frac{p_{\text{diss}}(t)}{d\Omega} = \frac{\sigma}{2}|\mathbf{E}|^2 \quad (2.7)$$

The SAR is obtained by normalizing by the tissue mass density (ρ):

$$\text{SAR} = \frac{\sigma}{2\rho}|\mathbf{E}|^2 \quad (2.8)$$

At a given location, the SAR is calculated as the ratio between the dissipated power and the mass densities. In EM dosimetry, the quantity that measures the energy absorbed by biological tissues is denoted as the Specific Absorption Rate (SAR) and provides the temperature rise estimation induced into the tissue. However, SAR is susceptible to the computational methods approximation so is more commonly described in an averaged form in two different ways, namely over the region mass:

$$\langle\text{SAR}\rangle_M = \frac{1}{M} \int_{R(M)\text{SAR}(\mathbf{r})} dm \quad (2.9)$$

or over the region volume:

$$\langle\text{SAR}\rangle_V = \frac{1}{V} \int_{R(V)\text{SAR}(\mathbf{r})} dv \quad (2.10)$$

Though the unit of thermal dose it's possible to evaluate the effectiveness of the hyperthermia treatment. It is expressed in cumulative equivalent minutes (CEM) of exposure to 43 °C [23] and is defined as follows:

$$CEM43 = \sum_{i=1}^n t_i \cdot R^{(43-T_i)} \quad (2.11)$$

$$R=0.25 \text{ per } T < 43 \text{ } ^\circ\text{C}$$

$$R=0.5 \text{ per } T > 43 \text{ } ^\circ\text{C}$$

where t_i is the i -th exposure time interval, R is related to the dependence among temperature and cell death rate, and T is the average temperature in the period of t_i . The resulting $CEM43^\circ\text{C}$ value represents the effect of the entire history of heat exposure on cell death [24, 25].

Modeling $CEM43$ does not require specific tissue parameters. Hotspots, exposure duration, the transient development of temperature and the different tissue temperature sensibilities are considered, establishing the tissue damage correlation between both temperature and time. R is the thermal isoeffect dose and specifies the exposure times necessary to obtain the same effect with a temperature increase of 1 °C.

Accuracy in simulating therapeutic heat treatment is essential for the planning of the same and for the dosimetry of single patients.

2.1.2 Thermal Modeling

On the other hand, thermal solver produces outcomes solving the Pennes' equation (or bio-heat equation):

$$\rho C \frac{\partial T}{\partial t} = k \nabla^2 T + q_s \quad (2.12)$$

where ρ is the tissue mass density [kg/m^3], C is the heat capacity [$\text{J kg}^{-1} \text{K}^{-1}$], k is the thermal conductivity [$\text{W m}^{-1} \text{K}^{-1}$]. The source term, q_s [W/m^3] can be divided into 3 components:

$$q_s = q_{hs} + q_m + q_p \quad (2.13)$$

- q_{hs} external heat source term
- q_m metabolic heat generation term (metabolic reactions)
- q_p heat loss due to blood perfusion

In this case, the heat source term is the EM heating (2.7):

$$q_{hs} = \frac{\sigma}{2} |\mathbf{E}_0|^2 \quad (2.14)$$

σ is the electrical conductivity in [S/m] and $|\mathbf{E}_0|^2$ is the electric field intensity [V/m]. To simplify, metabolic heat generation can be assumed negligible ($q_m \approx 0$).

If the heat source does not change with time, (2.12) can be rewritten as follows:

$$\rho C \frac{\partial T}{\partial t} = k \nabla^2 T + q_{hs} + q_p \quad (2.15)$$

After reaching equilibrium, i.e., in stationary conditions, the term relating to the temperature variation over time is canceled $\partial T_{ss}/\partial t = 0$ (where T_{ss} represents the steady-state temperature). That allows to rewrite the (2.15) by omitting the C and forming the Pennes' steady-state equation:

$$k \nabla^2 T_{ss} + q_{hs} + q_p = 0 \quad (2.16)$$

During EM heating in a stationary state (SS), the EM source can be approximately considered, in the region of interest, bigger than all the other thermal gradient:

$$q_{hs} \gg |k \nabla^2 T_{ss}| \quad (2.17)$$

which leads the (2.16) to:

$$q_{hs} \approx -q_p \quad (2.18)$$

According to the (2.14) and also to the (2.8), it is possible to write:

$$\rho \text{SAR} \approx -q_p \quad (2.19)$$

The (2.15) enables to describe the physiological phenomenon including however some constraints. It avoids the consideration of the veins heat, the arteries heat released is assumed to be absorbed locally by the tissues, and so it equals arteries temperature to the body temperature. All these factors produce a blood perfusion overestimation, that could be calculated [26]:

$$q_p = -\omega_b C_b \rho_b (T - T_a) \quad (2.20)$$

where ω_b is the blood perfusion rate [s^{-1}], C_b is the specific heat of blood [$J \text{ kg}^{-1} \text{ K}^{-1}$], ρ_b the mass density of blood and T_a is the arterial blood temperature [K]. The minus sign in (2.20) represents the blood compensation to temperature variations, it provides heat if the temperature decreases and removes heat if there is a rise. From the ω_b in (2.20) it is possible to calculate the perfusion rate relative to tissue mass ω as follows:

$$\omega = \frac{\omega_b}{\rho} \quad (2.21)$$

which is expressed in $[\text{ml kg}^{-1} \text{min}^{-1}]$ and it is one of the thermal parameters that will be considered and varied for the examined tissues.

Monitoring patient temperature during HT only via simulations is not safe, since the parameters, included in the simulations, are known with high uncertainty. This could lead to bias in the temperature maps and failure in heating the target region. Controlling the temperature in these areas is essential, so accurate thermometry tools are employed to obtain reliable data. Commonly, interstitial catheters with fiber optic sensors are involved, despite being an invasive and discomfort method for the patient. For this reason, new progressing studies aim to discover specific non-invasive methods for temperature monitoring [27].

2.1.3 Tissue Parameters

The electromagnetic energy absorption is related to electrical tissue parameters, whereas thermal parameters affect heat transfer within tissues [22].

- **Dielectric Parameters**

- *Electric permittivity*

$$\epsilon = \frac{D}{E} \quad (2.22)$$

ϵ [F/m] defined as the ratio between the electric displacement and electric field within a material. From ϵ , for one specific material, the *Relative Permittivity* ϵ_r can be calculated as:

$$\epsilon_r = \frac{\epsilon}{\epsilon_0} \quad (2.23)$$

it is the ratio between the electric permittivity of that material and the electric permittivity of a vacuum (ϵ_0), it is a dimensionless quantity.

- *Electric conductivity*

$$\sigma = \frac{J}{E} \quad (2.24)$$

σ [S/m] defines the ratio between the current density within a material and the electric field.

- **Thermal Parameters**

- *Thermal conductivity*

$$k = \frac{Q}{\nabla^2 T} \quad (2.25)$$

k [W m⁻¹ K⁻¹] describes heat conduction within tissue.

- *Perfusion*

represented by the letter ω [ml kg⁻¹ min⁻¹].

Temperature distribution is influenced by blood perfusion inside the vascular system, particularly in local regions. The distribution of tumor vasculature, due to multiple angiogenic growth factor release, is chaotic and generally reduces the blood flow. One potential explanation for this evident contradiction is that the mechanisms regulating growth and adaptation of vascular networks are impaired. According to a recent theory for structural adaptation of vascular networks, heterogeneous perfusion in tumors may result from perturbation of this system. Consequently, angiogenesis may increase perfusion heterogeneity [28] and leads to less efficient heat removal within the tumor region, which possibly results in higher temperatures during heating. Thermal and dielectric tissue parameters of normal and tumor tissues, assigned to the different tissues in the numerical model, are currently known with high uncertainty. These values, reported in the Literature, derived from ex-vivo measurements and can introduce errors in the simulations, leading to wrong temperature maps of the patient.

2.2 Sim4Life Software

Sim4Life, product by ZMT Zurich MedTech, is a revolutionary Multiphysics simulation platform for computational life scientists. It combines computable human phantoms with the most powerful physics solvers and the most advanced tissue models. The software directly analyzes biological real-world phenomena and complex technical devices in a validated biological and anatomical environment. The software's phantoms, subsequently provide a realistic biological environment for conducting fundamental studies for testing the effectiveness and safety of medical devices and treatments, and for supplementing clinical trials [29].

The Electromagnetics Full Wave Solvers (P-EM-FDTD) enable accelerated full-wave, large-scale EM modeling (billion voxels) with Yee discretization on geometrically adaptive, inhomogeneous, rectilinear meshes with conformal sub-cell correction and thin layer models. These solvers, which are the most frequently applied in near-field dosimetry, have been extensively validated and documented according to the IEEE/IEC 62704-1 standard as well as by comparisons with measured data [30].

The Thermodynamic Solvers (P-THERMAL) enable the modeling of heat transfer in living tissue using advanced perfusion and thermoregulation models. Exclusive thermal damage and effect quantification models, e.g., *CEM43*, are included [31].

One of the key features of Sim4Life is its data visualization capability, it can process and analyze various medical image data with the fully integrated toolbox Isaac, or the medical image segmentation tool set. Anatomical models from various types of imaging data can be generated efficiently, rapidly, and flexibly. The wide selection of available segmentation methods ensures that surface models with many different tissue types can be also generated with efficiency and flexibility.

2.2.1 Anthropomorphic Phantom

The human phantoms are based on the virtual population ViP 3.x/4.0 of the IT'IS Foundation at ETH Zurich. These models include integrated posing and morphing tools, characterized to predict the real-world physiological phenomena. Tissues are linked to the “physical properties database” which is constantly updated [32].

The phantom used for this work, taken from ViP, is the female young adult Yoon-sun shown in Figure 2.1.



Yoon-sun	
Sex	Female
Type	Young Adult
Age [Years]	26
Height [m]	1.52
Weight [kg]	54.60
BMI [kg/m ²]	23.60

Table 2.1: Yoon-sun Phantom Characteristics

Figure 2.1: Yoon-sun Phantom

The tumor model was created inside Yoon-sun phantom in the neck region (in green in Figure 2.3), in order to recreate a H&N cancer scenario and perform a hyperthermal treatment. The tumor, represented by an ellipsoid (see Figure 2.2), is modelled using the modelling tool in Sim4Life.

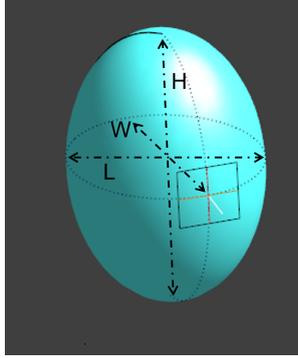


Figure 2.2: Tumor

Tumor Model	
Volume [mm ³]	4041.44
H [mm]	26.25
L [mm]	16.36
W [mm]	18.07

Table 2.2: Tumor Model Dimensions

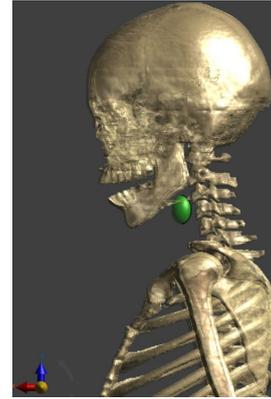


Figure 2.3: Tumor Location

2.2.2 Antenna Array

The applicator used for the thesis purpose is similar to M.M Paulides' model [33]. It is made by a circular array of 8 equidistant patch antennas, which create a ring shape applicator, positioned around the neck of the Yoon-sun phantom as presented in Figure 2.4.

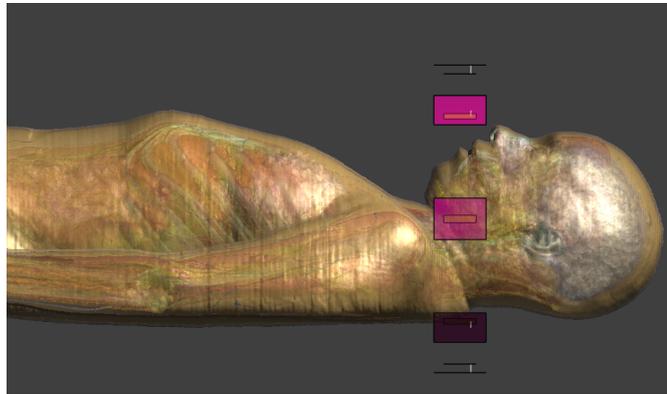
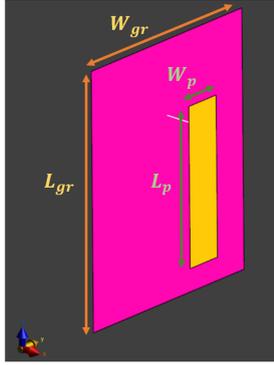


Figure 2.4: Array of Antennas

This set-up, according to Henke et al. [34, 35], in symmetrical geometry regions like the neck, guarantees the optimal configuration to maximize transversal waves interference [33]. Single patch antenna is designed as in Figure 2.5, then was set the operating frequency at 434 MHz, clinically approved for H&N cancers, preventing hotspots and improving treatment outcomes.



Patch Antenna	
Parameters	Dimensions [mm]
W_{gr}	40.00
L_{gr}	50.00
W_p	7.21
L_p	31.00
x_d	4.96
h_{sub}	8.40

Table 2.3: Patch Antenna Dimensions

Figure 2.5: Patch Antenna

Patch antennas, also known as microstrip antennas, are commonly used in medical applications due to their simple design and easy band optimization. Microstrip patch antenna consists of a dielectric substrate and of a metal patch, the ground, placed on the side. The substrate is usually thin and the metal patch on the front can have numerous shapes [36]. The patch antennas considered in the present thesis have the substrate made by the water of the waterbolus ($\epsilon_r = 76.7$ for distilled water and thickness h_{sub}) in which they are immersed. The geometry of the antenna influences the working frequency, so the antennas are dimensioned properly to provide 434 MHz.

The reflection coefficient S_{11} of each antenna, the ratio between the amplitude of the reflected signal to the incident signal, is validated to verify the frequency setup. Figure 2.6 shows the S_{11} of the source 8, acquired maintaining this as the only active source and keeping off the others, plotted both through Sim4Life. The resonance in bandwidth around the set frequency (434 MHz) testifies to the accurate set.

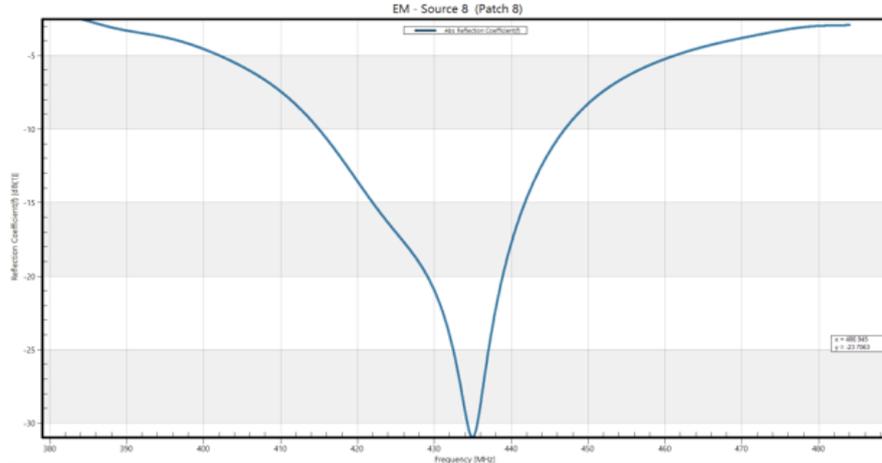


Figure 2.6: S_{11} of the 8th antenna

The modeling process is now completed: the highly realistic human phantom Yoon-sun is loaded from the ViP library, tumor model is allocated in the neck region and the antenna array, made by 8 microstrip patch antennas, is placed around the anatomical region of interest. The next step consists in solving the Maxwell's equation in the volume, which was divided into voxels, i.e., a finite number of cubic elements (see Figure 2.7), running the Sim4Life's EM FDTD Multiport simulation. It computes the fields generated by each antenna acting as standalone (i.e., with unitary feeding voltage, when all the other antennas are switched off and closed to a 50Ω load). Then, the computed electric fields are saved for all the computational domain points and exported to MATLAB for future optimization.

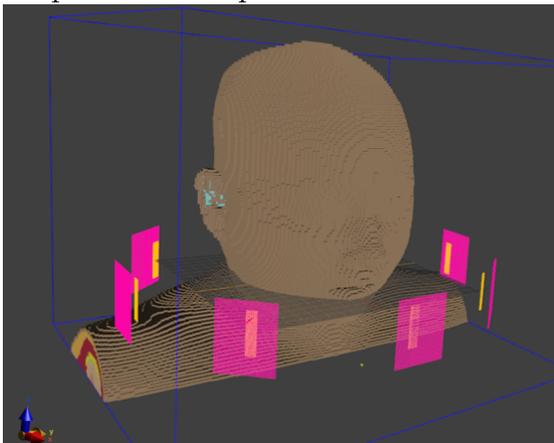


Figure 2.7: Domain Voxels

2.3 HTQ SAR-based Optimization

The SAR is related with the tumor surface area and depth, depending on the hyperthermia applicator characteristic [17]. The perturbations on the SAR distribution are provided by anatomy, antenna array and waterbolus biases. Accurate analysis can be done separately to these three cases, verifying which one has more influence on the outcome. In this work the validation process does not cover the waterbolus distortion, analyzing however the antenna array phase and amplitude optimization and the accuracy of the tissue parameters. By considering the two latter aspects, it is possible to validate the performances of the temperature map reconstruction algorithm which recreates the temperature distribution inside the human phantom from few known thermal points.

In the previous Section 2.2.2, the entire model is created in order to analyze the antenna array outcomes without any optimization. Figure 2.8 reports the $|\mathbf{E}|$, $|\mathbf{J}|$ and SAR, in the xy plane, relative to the whole array with a total input power of 20W, when the feedings of the different antennas are not optimized. As the figures show, no specific target region is present in the anatomical site, all the volume is

exposed to the microwave energy. In SAR distribution (Fig. 2.8c) however, it is evident that the highest values are in a specific spot, in the posterior neck of the phantom, far from the tumor region, which is the circular element easily detectable in the image.

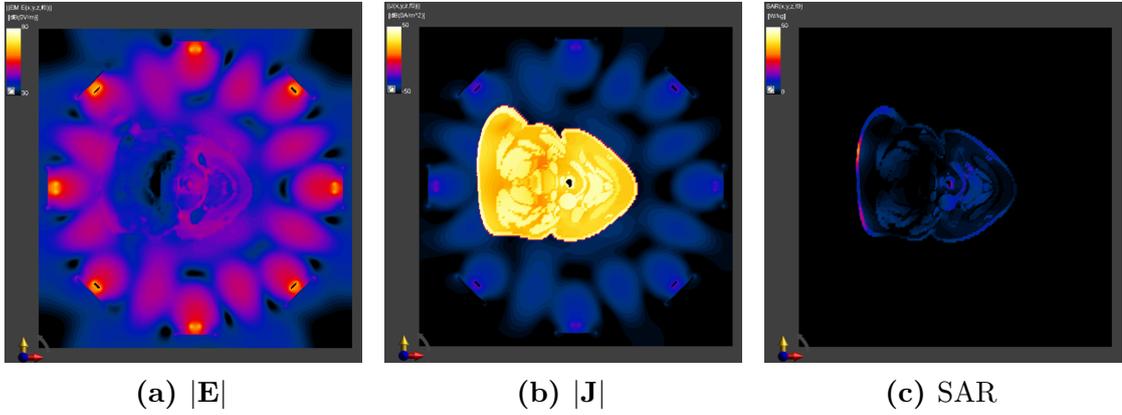


Figure 2.8: Sim4Life viewer, xy plane at 20W power

2.3.1 Phase and Phase-Amplitude Optimization

Firstly, the results are exported automatically, thanks to the Python code (see Code 11 full reported in Python Scripts 4.3), as MATLAB files; in MATLAB is then possible to run the optimization algorithms. Code lines 11, thanks to the “mask filters”, which allows to mask a 3D field assigning specific value in the region of interest and other outside it, divide the entire domain into two main structures: the healthy region H , made by the whole neck region with all the sensible organs without tumor, and the tumor region T .

```
def all_entities_within_group(entity_group):
    if isinstance(entity_group, model.EntityGroup):
        return list(itertools.chain.from_iterable(
            all_entities_within_group(e) for e in entity_group.Entities))
    else:
        return [entity_group]

vip_group = model.AllEntities()['Yoon-sun']
entities_H = all_entities_within_group(vip_group)
entities_T = model.AllEntities()['Tumor']
```

Code 2.1: Tissues Selection for Mask Filter

Mask filters are applied multiple times to export (see Code 16) all the 8 standalone electric fields E for each region, using an iterative loop, but also to export the induction current J and SAR related to each structure.

```

# Adding a new MatlabExporter (T)
inputs = [field_masking_filter.Outputs["EM E(x,y,z,f0)"]]
matlab_exporter = analysis.exporters.MatlabExporter(inputs=inputs)
matlab_exporter.UpdateAttributes()
document.AllAlgorithms.Add(matlab_exporter)

for i, output_port in enumerate(em_multi_port_simulation_extractor.Outputs):
    em_port_simulation_extractor.raw.SetInputConnection(0, output_port.raw)
    em_port_simulation_extractor.UpdateAttributes()
inputs = [field_masking_filter.Outputs["EM E(x,y,z,f0)"]]
matlab_exporter.FileName = (dir + "E{}_T.mat".format(i+1))

print(matlab_exporter.FileName)
matlab_exporter.UpdateAttributes()
matlab_exporter.Update()
    
```

Code 2.2: Mask Filter Application and MATLAB Exporter for the E^T

The latter are used to calculate, for both groups, the electrical conductivity σ and the mass density ρ :

$$\sigma = \frac{|\mathbf{J}|}{|\mathbf{E}|} \quad (2.26)$$

$$\rho = \frac{1}{2 \text{ SAR} \sigma |\mathbf{E}|^2} \quad (2.27)$$

Then, in MATLAB environment, the 8 electric fields of each region, E^H and E^T , are sorted in 3 matrices dividing them by coordinates.

$$E^H.X = \left(\begin{array}{c} \left[\begin{array}{c} E_{1,x}(r_1) \\ E_{1,x}(r_2) \\ \dots \\ E_{1,x}(r_{M_H}) \end{array} \right] \\ \left[\begin{array}{c} E_{2,x}(r_1) \\ E_{2,x}(r_2) \\ \dots \\ E_{2,x}(r_{M_H}) \end{array} \right] \\ \dots \\ \left[\begin{array}{c} E_{N,x}(r_1) \\ E_{N,x}(r_2) \\ \dots \\ E_{N,x}(r_{M_H}) \end{array} \right] \end{array} \right), E^H.Y, E^H.Z \quad (2.28)$$

M_H : total points in H . $N = 8$ number of antennas

$$E^T.X = \left(\begin{array}{c} \left[\begin{array}{c} E_{1,x}(r_1) \\ E_{1,x}(r_2) \\ \dots \\ E_{1,x}(r_{M_T}) \end{array} \right] \\ \left[\begin{array}{c} E_{2,x}(r_1) \\ E_{2,x}(r_2) \\ \dots \\ E_{2,x}(r_{M_T}) \end{array} \right] \\ \dots \\ \left[\begin{array}{c} E_{N,x}(r_1) \\ E_{N,x}(r_2) \\ \dots \\ E_{N,x}(r_{M_T}) \end{array} \right] \end{array} \right), E^T.Y, E^T.Z \quad (2.29)$$

M_T : total points in T . $N = 8$ number of antennas

From these two matrices, one filled with all fields related to tumor while the other containing all fields related to the healthy area, the Hotspot to Tumor Quotient

(HTQ) coefficient is minimized focusing the SAR on the tumor preventing the surrounding tissues.

$$HTQ = \frac{\langle \text{SAR}_{V1}^H \rangle}{\langle \text{SAR}_{target} \rangle} \quad (2.30)$$

where $\langle \text{SAR}_{V1}^H \rangle$ is the hotspot SAR, defined as the average SAR in V1, which is the 1% of the healthy volume with the highest SAR, whereas the SAR_{target} is the average SAR in the tumor [37, 38]. Minimizing the HTQ , which is our objective function, allow us to know the proper characteristics we need to set on our array, so to each antenna, leading to a SAR pattern focused on the tumor region.

The electric field \mathbf{E} in equation (2.8), as mentioned before, is exported in each point of the region of interest and allocated into two structures (healthy and tumor). The total electric field \mathbf{E} generated by the antenna array can be written as:

$$\mathbf{E} = \sum_{n=1}^N \tilde{v}_n \cdot \mathbf{E}_n \quad (2.31)$$

In (2.31) the \mathbf{E}_n is the electric filed generated by the N_{th} antenna of the array, acting as standalone (i.e., feeded with unitary excitation when all the other antennas of the array are off) and \tilde{v}_n are the coefficients that need to be optimized. For the phase optimization \tilde{v}_n are expressed as:

$$\tilde{v}_n = C \cdot \nu_0 \cdot e^{i\varphi_n} \quad (2.32)$$

where φ_n are the antennas phases included in the range $[0, 2\pi]$, $\nu_0 = \sqrt{2R_0P_0}$, with $R_0=50\Omega$ and P_0 as the total input power of the array, while C :

$$C = \frac{1}{\sqrt{N}} \quad (2.33)$$

is a coefficient for the power normalization where N represent the number of antennas.

For the optimization of phase and amplitude the \tilde{v}_n can be written as follow:

$$\tilde{v}_n = C \cdot \nu_0 \cdot \zeta_n \cdot e^{i\varphi_n} \quad (2.34)$$

where ζ_n are included in the range $[0, 1]$. The optimization of the amplitude of each antenna in the array is achieved varying them. In this case C is:

$$C = \frac{1}{\sqrt{\sum_n \zeta_n^2}} \quad (2.35)$$

To minimize the objective function HTQ , and so obtain the phases only and phase and amplitude values for each antenna, particle swarm optimization is involved. Particle swarm optimization (PSO) [39] allows function minimization,

receiving the limits and the objective function, in this case HTQ is a hybrid function, through an iterative process. The results of the two cases are reported below (Fig. 2.9).

Antenna	φ [deg]
1.00	28.29
2	29.54
3	0.00
4	337.37
5	294.87
6	296.53
7	301.06
8	$2.08e^{-04}$

Table 2.4: Phase-Only Optimization

Antenna	φ [deg]	ζ^2 [W]
1.00	27.05	0.86
2	32.53	0.75
3	36.00	0.54
4	0.00	0.04
5	307.29	0.35
6	293.56	0.41
7	311.66	0.99
8	$5.28e^{-06}$	0.75

Table 2.5: Phase-Amplitude Optimization

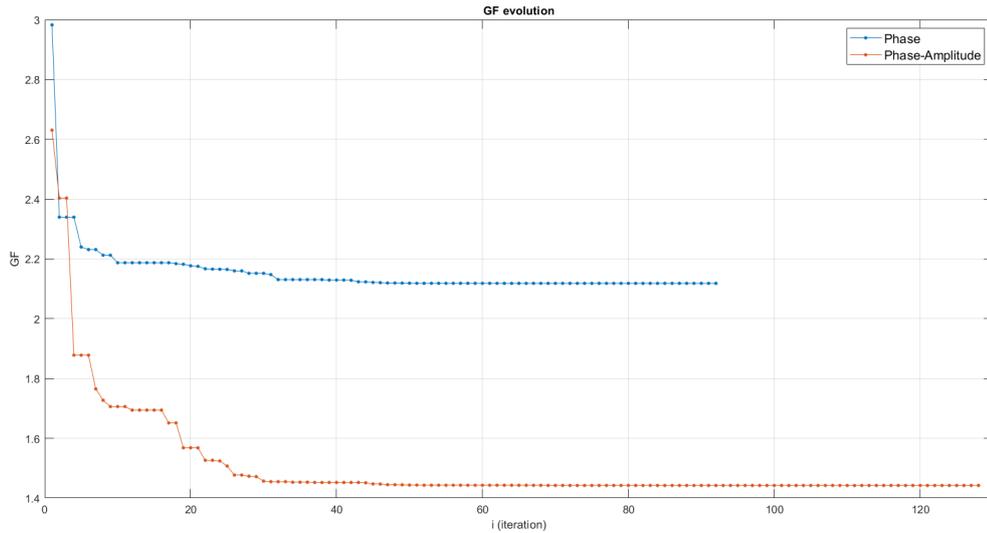
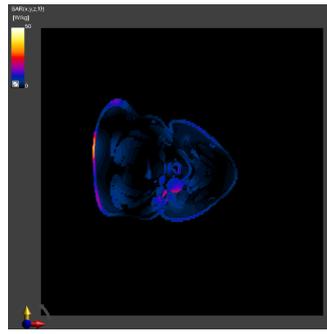


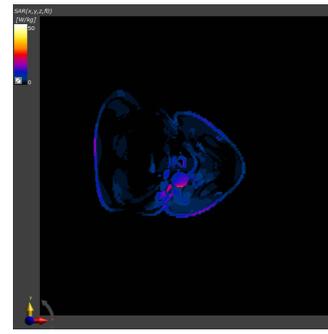
Figure 2.9: PSO Comparison Phase-Only and Phase-Amplitude Optimization

2.3.2 Optimized SAR Maps

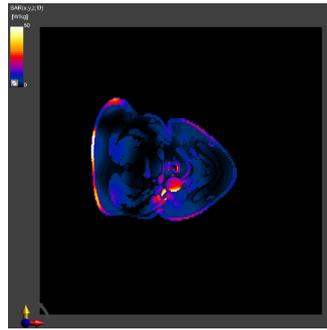
Thanks to the MATLAB algorithm, the SAR optimization is achieved working both only on the phases of the antennas in the array and also modifying their phases and amplitudes.



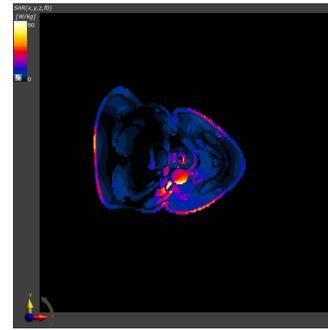
(a) Phase-Only $P_0=20\text{W}$



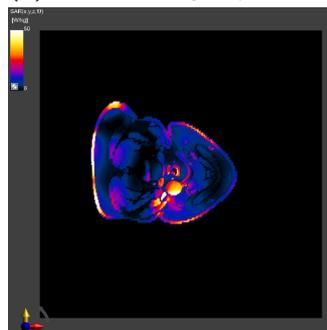
(b) Phase-Amplitude $P_0=20\text{W}$



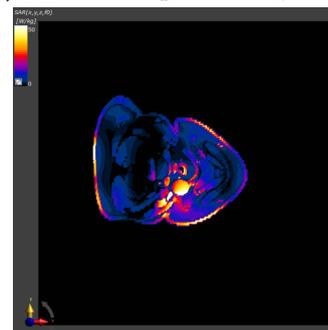
(c) Phase-Only $P_0=40\text{W}$



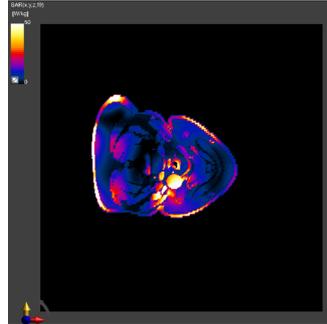
(d) Phase-Amplitude $P_0=40\text{W}$



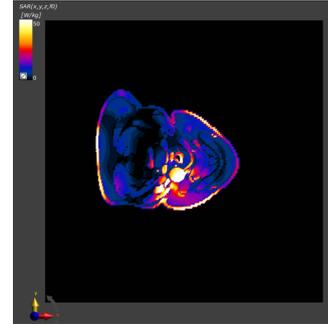
(e) Phase-Only $P_0=60\text{W}$



(f) Phase-Amplitude $P_0=60\text{W}$



(g) Phase-Only $P_0=80\text{W}$



(h) Phase-Amplitude $P_0=80\text{W}$

Figure 2.10: SAR distribution in different cases, plane xy

Figure 2.10 displays the comparison between the two different scenarios. It is evident that in second case the hotspots in the posterior neck healthy region on the phantom are reduced, and a better radiation focusing on the tumor region is achieved. In Figures 2.12c and 2.12d both array powers are normalized at 20W like done in the 2.8c. The next step consists in different normalization by increasing the power given to the array. As displayed in all the plots reported in Figure 2.10, more power added results in higher SAR values in the tumor region.

2.3.3 Temperature Maps

In hyperthermia treatments, the microwave array applicator provides energy to the tumor region to increase the local temperature damaging its cells. Hence, is crucial to see the temperature map distribution inside the model, corresponding to the optimized SAR maps (i.e., when the optimized SAR maps shown in Section 2.3.2 are used as input source for the bioheat equation), at different power normalizations. In the computational domain for the bioheat equation all neck tissues are considered except the trachea and the lungs.

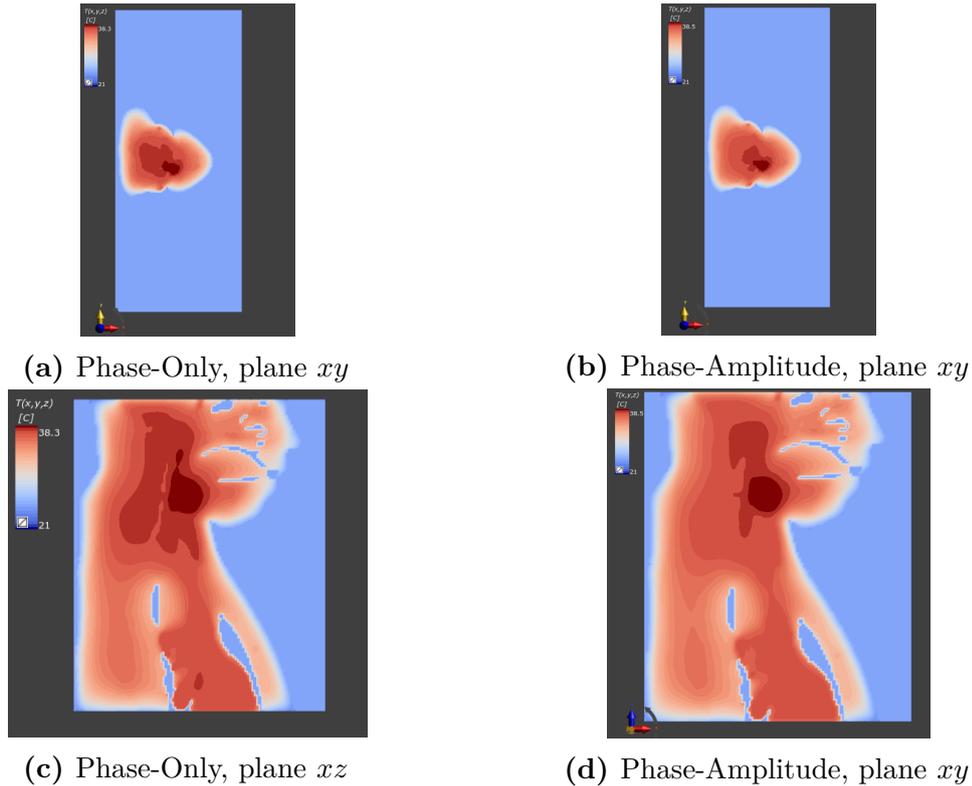


Figure 2.11: Temperature Maps at 20W

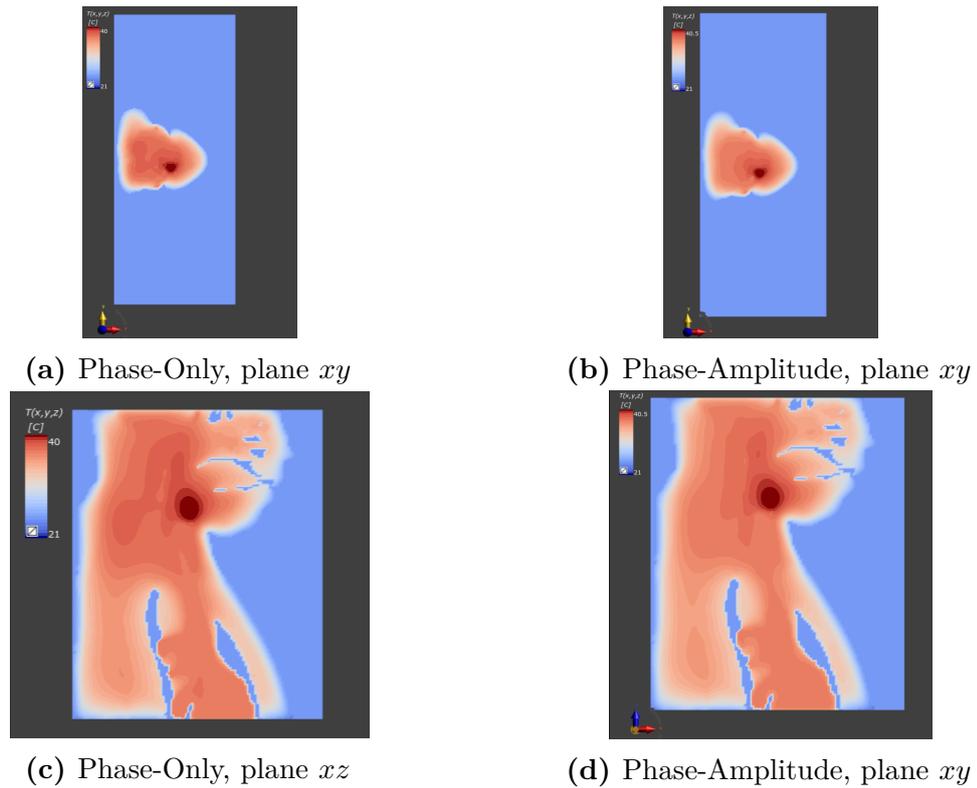


Figure 2.12: Temperature Maps at 40W

In Figures 2.11 and 2.12 the temperature maps show that 20W and 40W power are not sufficient to reach the treatment goal temperature in tumor area, the maximum values do not exceed 40.5 °C. Furthermore, it is clearer here how important it is to also optimize the amplitude of the antenna array due to its better temperature distribution, which leads to a better and safer hyperthermia outcome for the patient. Therefore, the examples reported in the following will only report the results corresponding to the phase and amplitude optimization.

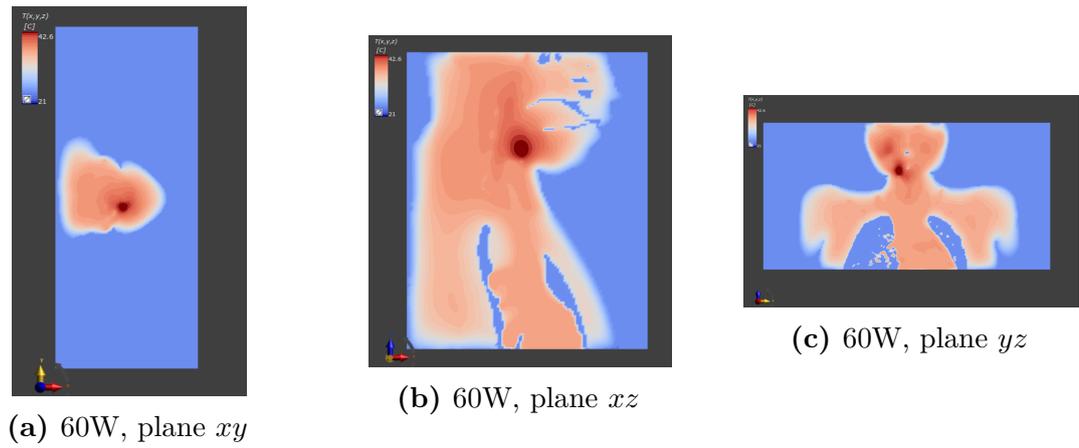


Figure 2.13: Temperature Map Phase-Amplitude Optimization at 60W

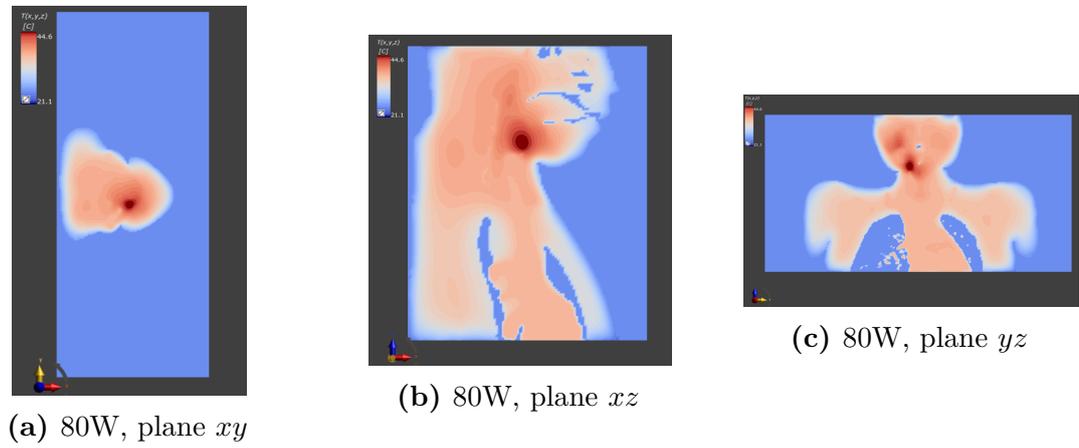


Figure 2.14: Temperature Map Phase-Amplitude Optimization at 80W

The 60W power case provides its highest quantity at 42.6 °C (Fig. 2.13), while at 80W the peak temperature value is 44.6 °C (Fig. 2.14).

For the subsequent works we must examine and select one of these two latter cases of antenna array power normalizations. Due to the uncertainty that affects the tissue parameters assigned to the solvers, we opted for a conservative choice that could preserve the patient from reaching too high temperature values, over the 44 °C which is considered the upper limit, in the tumor region. For this reason, the 80W case was discarded and all the follow simulations involved a 60W power normalization of the applicator.

Chapter 3

Temperature Maps Reconstruction

In this chapter, the tissue parameters of the phantom are modified several times in order to reach a most reliable 3D temperature map of the patient by means of a reconstruction algorithm. Sections 1 and 2, especially the equation (2.20), explain how blood flow plays a key role in heat modification and distribution within tissues during biological processes. As mentioned before, parameters values are reported in the Literature, but they cannot be considered 100% correct since they derived from anatomical measurements made on dead bodies. Moreover, these parameters may vary from one patient to another, and be different from the baseline values, making the treatment less patient-specific. Therefore, ex-vivo values could lead to misleading patient-simulated temperature maps making the therapy ineffective or even dangerous.

3.1 Multigrid Creation

Despite the uncertainty characterizing the values reported in Literature, they are consistent and provide an idea of the dimensions involved in the problem. Over the baseline, i.e., the average value, also the upper and lower limit are described giving a range (see Table 3.1) [32].

In this thesis, different combinations of parameter values are produced for the most relevant tissues in the H&N region: *muscle*, *tumor*, *skin*, and *fat*, also considering the *SAT* (subcutaneous adipose tissue), varying the parameters indicated in Section 2.

ϵ_r	Baseline	Min	Max
Muscle	56.90	54.00	59.70
Fat	11.60	11.00	12.20
Skin	49.40	46.90	51.90
Tumor	59.00	56.00	62.00

(a) Relative Permittivity Values

σ	Baseline	Min	Max
[S m ⁻¹]			
Muscle	0.81	0.77	0.85
Fat	0.08	0.08	0.09
Skin	0.68	0.65	0.72
Tumor	0.89	0.85	0.94

(b) Dielectric Conductivity Values

k	Baseline	Min	Max
[W m ⁻¹ K ⁻¹]			
Muscle	0.49	0.40	0.56
Fat	0.21	0.18	0.50
Skin	0.37	0.32	0.50
Tumor	0.51	0.48	1.50

(c) Thermal Conductivity Values

ω	Baseline	Min	Max
[ml kg ⁻¹ min ⁻¹]			
Muscle	39.10	39.10	442.00
Fat	33.00	33.00	255.00
Skin	106.00	106.00	175.00
Tumor	72.30	72.30	848.00

(d) Perfusion Values

Table 3.1: Parameters Literature Values [32]

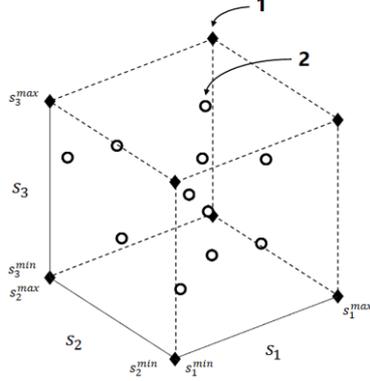
In order to create different combinations, pseudo-random multidimensional grids are employed. Multidimensional grids (multigrids) are grids of N_{tot} n -tuples, where N_{tot} is the number of combinations considered, and n is the number of parameters in each set. Thanks to the multigrids, performed in the MATLAB environment, several distinct sets are generated. Providing the baseline, maximum and minimum values of the parameters investigated, the MATLAB algorithm produces value combinations that vary in the range, stated in Literature, for each tissue in analysis. Thus, it is possible to evaluate separately the different nature of parameters, dielectric and thermal, or the various tissues type present in the region.

Two different types of multidimensional grids are created (see Figures 3.1 and 3.2), building in that way two different approaches by which the sequences of values are generated. Consequently, it is possible to have several sets and then produce their temperature maps, observing how the changed values alter them.

3.1.1 Multigrid A + Baseline

The first type of multidimensional grid is the ‘‘MGrid Type A’’ (see Figure 3.1). The dimension of the points forming the multigrid is indicated as N , i.e., as the product of the number of tissues n and parameters m considered. MGrid Type A is formed from the different combinations of the extreme bound values (1 in Fig.3.1) of the parameters involved plus some pseudo-random or Sobol [40] values N_S (2 in Fig.3.1). For this reason, there is a minimum number of combinations

created, which is $2^{n \cdot m} = 2^N$. The total combinations is reported in equation (3.1):



$$N_{tot} = 2^N + N_S \quad (3.1)$$

where $N = n \cdot m$

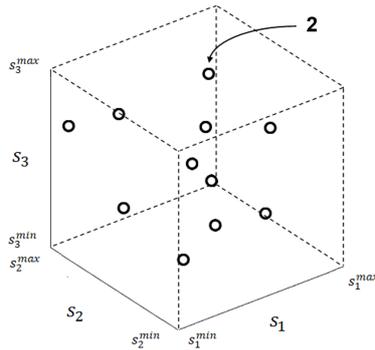
Figure 3.1: MGrid type A with $N=3$

For the next analysis, the “Multigrad Type A” is made considering 2 tissues, *muscle* and *tumor*, and their 2 thermal parameters, w and k , having dimension $N = 2 \cdot 2 = 4$ and forming $2^{2 \cdot 2} = 16$ minimum combinations. The number of Sobol values N_S inserted is defined during the creation process. Moreover, one line is added: the first set is made forcedly by the baseline values, creating the “MGrid Type A + Baseline”.

$$N_{tot} = 1 + 2^N + N_S \quad (3.2)$$

3.1.2 Multigrad B + Baseline

The other multidimensional grid type, called the “Multigrad Type B” (fig.3.2), is produced only from pseudo-random values N_S within the range provided. Its dimension is related to the number of tissues and parameters inserted without having a minimum number of combinations.



$$N_{tot} = N_S \quad (3.3)$$

where $N = n \cdot m$

Figure 3.2: MGrid type B with $N=3$

This case is also modified as before by adding the baseline values as the first set

for the next steps:

$$N_{tot} = 1 + N_S \quad (3.4)$$

3.2 Svar Matrix Creation

After the generation of several combination sets for tissue parameters, the allocation step occurs. One set at a time is assigned to the corresponding tissue parameters inside the thermal solver which runs the simulation and creates the corresponding temperature maps. These maps allow the creation of the S_{var} and S_{cath} matrices, used in the reconstruction algorithm.

3.2.1 Map Extraction via Python Scripts

Thanks to the Python interface present in Sim4Life, it is possible to create a Python code that allocates diverse parameters sets to the segmented tissues and runs the thermal solver iteratively. Firstly, MATLAB multigrid generation script (see Code 4.3 in Matlab Scripts) reads the Excel file “Parameters”, where all the Literature parameter values for each tissues are reported, and generates the required sets in a text file organized in columns and rows. Each column represents the values of a different parameter for a specific tissue, so every row corresponds to a different set of combinations. In Table 3.2 are reported the 27 combinations created with a MGrid Type A + Baseline, modifying both the thermal parameters, ω and k , in *tumor* and *muscle* and considering $N_S = 10$ (see equation (3.2)). These sets are then imported in Sim4Life via Python Code lines 3.1, which read the output text file, and by Code lines 3.2 that change the properties for the tissues selected.

```
for i in range (0,num_rows):
    km = coord[i,0]
    kt = coord[i,1]
    wm = coord[i,2]
    wt = coord[i,3]
```

Code 3.1: Set Reading

```
entity__tumor = model.AllEntities()["Muscle"]
muscle_new_w_value = wm, Unit("ml/min/kg")
muscle_new_k_value = km, Unit("W/m/K")
entity__tumor = model.AllEntities()["Tumor"];
tumor_new_w_value = wt, Unit("ml/min/kg")
tumor_new_k_value = kt, Unit("W/m/K")
# Change the Perfusion value and Thermal Conductivity of an existing material
sets = simulation.AllSettings
for idx,set in enumerate(sets):
    if set.Name == 'Muscle':
        set.HeatTransferRate.UsePerfusionUnits = True
        set.HeatTransferRate.PuConstantTerm = muscle_new_w_value
        set.ThermalConductivity = muscle_new_k_value
    if set.Name == 'Tumor':
```

```

set.HeatTransferRate.UsePerfusionUnits = True
set.HeatTransferRate.PuConstantTerm = tumor_new_w_value
set.ThermalConductivity = tumor_new_k_value

simulation.UpdateAllMaterials()

```

Code 3.2: New Values Association

Finally, the Code 3.3 run the simulation and generate the corresponding 27 temperature maps, using as a EM source the antenna array optimized in both phases and amplitude and normalized at the total input power of 60 W (see Section 2).

```

simulation = document.AllSimulations["Th 60W"]
simulation.ClearResults()
simulation.ResetVoxels()
# RUN
simulation.UpdateGrid()
simulation.CreateVoxels()
simulation.RunSimulation(wait=True)

```

Code 3.3: Simulation Run

In Figures 3.3 and 3.4 is noticeable how different temperature distributions relate to different combinations. These temperature maps are created in Sim4Life, then exported, iteratively by the Code 3.4, and visualized in MATLAB. The fact that some maps, e.g fig. 3.4, do not reach 42/43 °C on the tumor is a normal consequence. The perfusion ω and thermal conductivity k of two important tissues (*muscle* and *tumor*) have been varied so the conditions of the problem are strongly altered, also recalling that the optimization process was carried out from the baseline values.

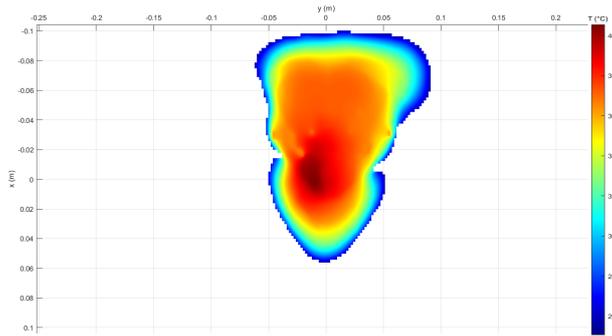
```

# EXPORT MATLAB
inputs = [overall_field_sensor.Outputs["T(x,y,z)"]]
field_masking_filter = analysis.core.FieldMaskingFilter(inputs=inputs)
field_masking_filter.SetAllMaterials(False)
field_masking_filter.SetEntities(entities__all)
field_masking_filter.UpdateAttributes()
document.AllAlgorithms.Add(field_masking_filter)

inputs = [field_masking_filter.Outputs["T(x,y,z)"]]
matlab_exporter = analysis.exporters.MatlabExporter(inputs=inputs)
matlab_exporter.Name = ("{}".format(i))
matlab_exporter.FileName = (dir + "T_Map{}.mat".format(i))
matlab_exporter.UpdateAttributes()
document.AllAlgorithms.Add(matlab_exporter)
matlab_exporter.Update(overwrite=True)
matlab_exporter.Update(overwrite=True)

```

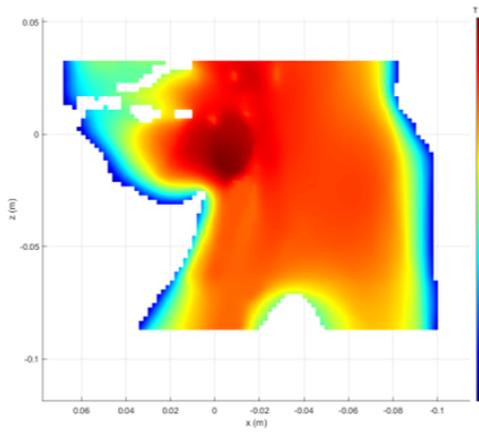
Code 3.4: MATLAB Exporter



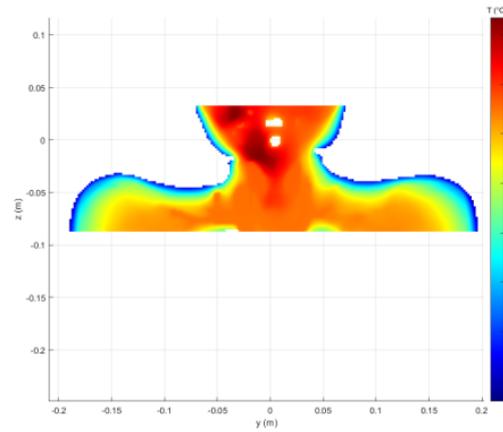
(a) Plane xy

tissue	k [W m ⁻¹ K ⁻¹]	ω [ml kg ⁻¹ min ⁻¹]
muscle	0.56	39.10
tumor	1.50	72.30

(b) Parameters Values

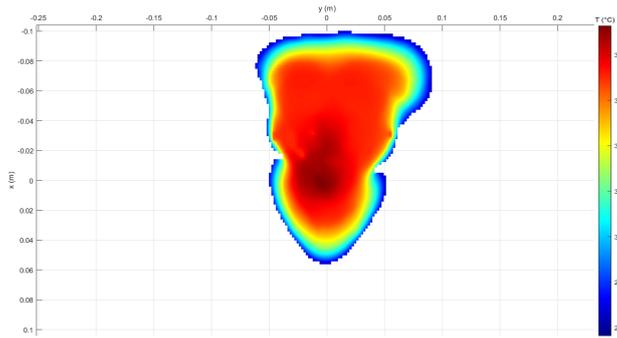


(c) Plane xz



(d) Plane yz

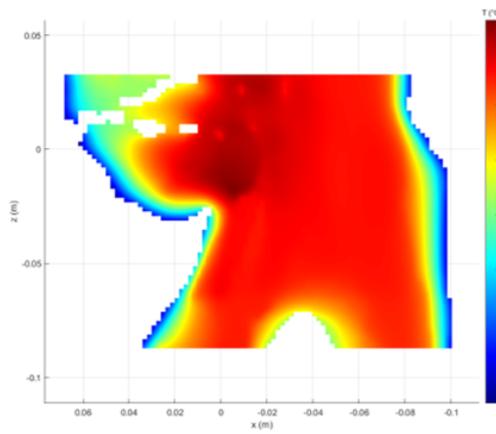
Figure 3.3: Tmap 5



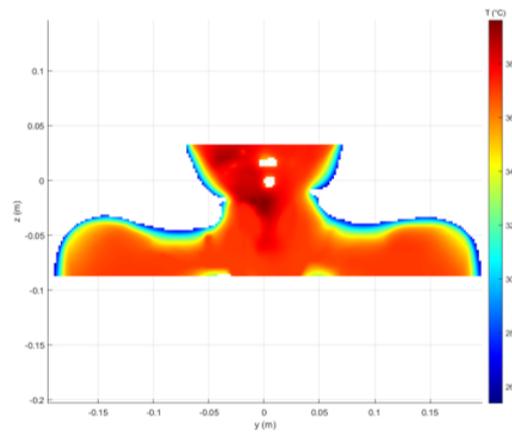
(a) Plane xy

tissue	k [W m ⁻¹ K ⁻¹]	ω [ml kg ⁻¹ min ⁻¹]
muscle	0.41	361.09
tumor	1.44	314.71

(b) Parameters Values



(c) Plane xz



(d) Plane yz

Figure 3.4: Tmap 25

k_m	k_t	ω_m	ω_t
[W m ⁻¹ K ⁻¹]	[W m ⁻¹ K ⁻¹]	[ml kg ⁻¹ min ⁻¹]	[ml kg ⁻¹ min ⁻¹]
0.49	0.51	39.10	72.30
0.40	0.48	39.10	72.30
0.56	0.48	39.10	72.30
0.40	1.50	39.10	72.30
0.56	1.50	39.10	72.30
0.40	0.48	442.00	72.30
0.56	0.48	442.00	72.30
0.40	1.50	442.00	72.30
0.56	1.50	442.00	72.30
0.40	0.48	39.10	848.00
0.56	0.48	39.10	848.00
0.40	1.50	39.10	848.00
0.56	1.50	39.10	848.00
0.40	0.48	442.00	848.00
0.56	0.48	442.00	848.00
0.40	1.50	442.00	848.00
0.56	1.50	442.00	848.00
0.48	0.99	240.55	460.15
0.44	1.25	139.83	654.08
0.52	0.74	341.28	266.23
0.42	1.12	391.64	751.04
0.50	0.61	190.19	363.19
0.46	0.86	290.91	169.26
0.54	1.37	89.46	557.11
0.41	1.44	316.09	314.71
0.49	0.93	114.64	702.56
0.45	0.67	416.85	508.63

Table 3.2: MGrid Type A + Baseline Values, k and ω of *muscle* and *tumor*

3.2.2 Temperature Map Reconstruction

After the maps exportation in MATLAB, the reconstruction algorithm is employed. It requires the creation of two matrices:

- S_{var} formed by different temperature maps N_{tot} obtained after the modification of the tissues values. Each map is an array containing the temperature values for each point in the whole domain. The S_{var} matrix is created by adding

every temperature map as a column.

$$S_{var} = \begin{pmatrix} \begin{bmatrix} T_1(r_1) \\ T_1(r_2) \\ \dots \\ T_1(r_M) \end{bmatrix} & \begin{bmatrix} T_2(r_1) \\ T_2(r_2) \\ \dots \\ T_2(r_M) \end{bmatrix} & \dots & \begin{bmatrix} T_N(r_1) \\ T_N(r_2) \\ \dots \\ T_N(r_M) \end{bmatrix} \end{pmatrix} \quad (3.5)$$

$(r_1, r_2 \dots r_M)$ spatial domain points

- S_{cath} is the matrix formed by temperature maps evaluated along the direction of insertion of a catheter, included as columns. This matrix is related to the known thermal points that, in real clinical application, can be measured through the minimally invasive catheter to obtain some measured temperature values inside the patient. The direction of insertion and the number of catheter capture points, hence the number of the investigated points of the whole domain, affect the outcomes of the reconstruction algorithm which tries to recreate the entire region from them. To generate the columns of the matrix S_{cath} , the coordinates in xy plane of the tumor are required, in that way it is possible to explore in the tumor direction.

```

for i = 1:Np
    T = load([filepath, 'T Map', num2str(i-1), '.mat']);
    Tmap = reshape(T.Snapshot0, [length(T.Axis0)-1, length(T.Axis1)-1, length(T.Axis2)-1, 1]);
    Tmap = Tmap(:, :, sz1:sz2);
    Tmap = reshape(Tmap, [numel(Tmap), 1]);
    Tmap(isnan(Tmap)) = [];
    Svar(1: numel(Tmap), i) = Tmap;
end
    
```

Code 3.5: S_{var} Matrix Input

```

type = 'y'           % type of catheter considered (along 'x' or 'y')
coord = -0.00333;    %[m] coordinate xc or yc of the catheter (e.g., xc_tum =
                    -0.00333 m, yc_tum = -0.01236 m)
Nc = 20;            % number of points along the catheter
dir_c = 'rev';      % catheter's direction (options: 'fw', 'rev')
step = 1;           % spacing among points along the catheter
    
```

Code 3.6: S_{cath} Matrix Input

The reconstruction algorithm considers one of the temperature maps of the S_{var} matrix as a target; it is supposed to be unknown. From the target map the algorithm creates the β vector, made by coefficients as follows:

$$[S_{var}] = \left([T_b], [T_1] \dots [T_{t=2^N}], [T_{sobol_{t=1}}] \dots [T_{sobol_{t=N_s}}] \right) \quad (3.6)$$

$$[\mathbf{T}_{\text{target}}] = \begin{bmatrix} T_{\text{sobol},t}(r_1) \\ \vdots \\ T_{\text{sobol},t}(r_M) \end{bmatrix} \quad (3.7)$$

$$\beta = \text{pinv}([S_{\text{var}}]_{1\dots N_{\text{cath}}}, 10^{-3}) \cdot [\mathbf{T}_{\text{target}}]_{1\dots N_{\text{cath}}} \quad (3.8)$$

$$[S_{\text{var}}]_{1\dots N_{\text{cath}}} = S_{\text{cath}} \quad (3.9)$$

Finally, the temperature map can be reconstructed:

$$[\mathbf{T}_{\text{rec}}] = [S_{\text{var}}][\beta] \quad (3.10)$$

Starting from the S_{var} matrix (3.6), a S_{cath} matrix is built using the first $N_{p,\text{min}}$ columns of the S_{var} matrix. The $(N_{p,\text{min}+1})_{\text{th}}$ column of the S_{var} matrix is assumed as a target field and reconstructed using S_{cath} . Then, the $(N_{p,\text{min}+1})_{\text{th}}$ column is added to the S_{cath} matrix and the $(N_{p,\text{min}+2})_{\text{th}}$ column is assumed as a new target field and reconstructed. Assuming $N_{p,\text{min}} = 4$, $27-4 = 23$ target fields are sequentially reconstructed using an S_{cath} matrix which has an additional columns at each step.

$$S_{\text{var}}^{\text{rec}} = \left(\begin{array}{c} \left[\begin{array}{c} T_1(r_1) \\ T_1(r_2) \\ \dots \\ T_1(r_M) \end{array} \right] \left[\begin{array}{c} T_2(r_1) \\ T_2(r_2) \\ \dots \\ T_2(r_M) \end{array} \right] \dots \left[\begin{array}{c} T_{N_{p,\text{min}}}(r_1) \\ T_{N_{p,\text{min}}}(r_2) \\ \dots \\ T_{N_{p,\text{min}}}(r_M) \end{array} \right] \left[\begin{array}{c} T_{N_{p,\text{min}+1}}(r_1) \\ T_{N_{p,\text{min}+1}}(r_2) \\ \dots \\ T_{N_{p,\text{min}+1}}(r_M) \end{array} \right] \left[\begin{array}{c} T_{N_{p,\text{min}+2}}(r_1) \\ T_{N_{p,\text{min}+2}}(r_2) \\ \dots \\ T_{N_{p,\text{min}+2}}(r_M) \end{array} \right] \dots \end{array} \right) \quad (3.11)$$

Minimum Number of Columns
First Target Field
t=1
Second Target Field
t=2

3.2.3 Variation of Catheter Insertion

Table 3.2 reports the 27 sets of combinations created with the MGrid type A + Baseline. From them, the linked temperature maps are generated in Sim4Life, two of these are plotted in Figures 3.3 and 3.4. From this group of values, the S_{var} matrix is created and the reconstruction algorithm is applied, varying multiple catheter direction and characteristics. Starting with a S_{cath} matrix with a minimum number of columns $N_{p,min} = 4$, the other 23 maps are considered one at a time as target fields and then added to the S_{cath} as new columns. In this way, the dimension of the S_{cath} matrix increases at each step, i.e., each target field is reconstructed using a reconstruction matrix with increasing dimension. The reconstructed fields are then compared to the starting target fields (i.e., the columns of the matrix S_{var}). The direction of the catheter insertion is evaluated in different positions on the xy plane where the tumor centroid is localized. However, one of the two coordinates on the plane is set to be the tumor one, in that way the insertion, so the investigation, is always towards the tumor. Different cases are created varying the number of points of the catheter, the spacing between them and the direction as mentioned previously. For each different case, the xy plane with the representation of the catheter on the baseline map is reported, together with a statistical analysis performed on the reconstructed fields. The three plots of the statistical analysis are organized as follows:

1. The first plot (upper row, right) reports the boxplot statistics of the difference $\Delta T = |T_{rec} - T_{target}|$ performed over the points of the whole region of interest, for each of the 23 target fields considered.
2. The second plot (lower row, left) reports on the same plot: the boxplot relative to $|T_{rec} - T_{baseline}|$ and the boxplot relative to $|T_{rec} - T_{target}|$ for the 5th target field considered.
3. The third plot (lower row, right), finally reports the median and the maximum value of ΔT for the different target fields considered.

Case 1

- Coordinates: $x = 0, y = -3.33$ [mm]
- Direction: along y
- Type: forward
- N.Points: 20
- Step: 1 [mm]

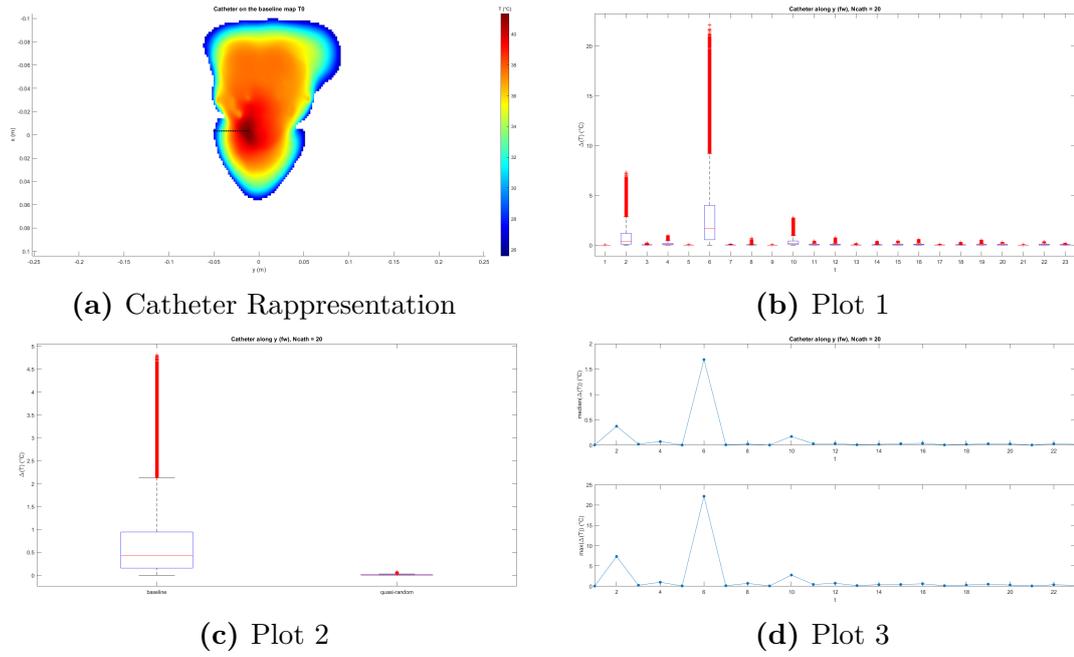


Figure 3.5: Case 1

Case 2

- Coordinates: $x = 0.00, y = -3.33$ [mm]
- Direction: along y
- Type: reverse
- N.Points: 40
- Step: 1 [mm]

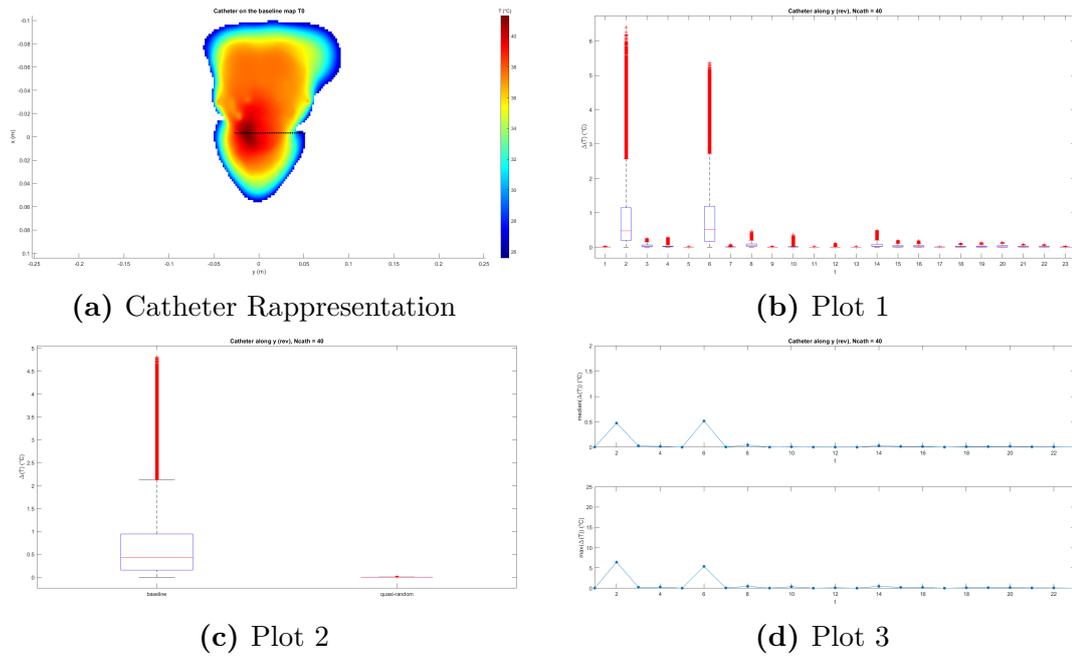


Figure 3.6: Case 2

Case 3

- Coordinates: $x = -12.36, y = 0.00$ [mm]
- Direction: along x
- Type: forward
- N.Points: 40
- Step: 1 [mm]

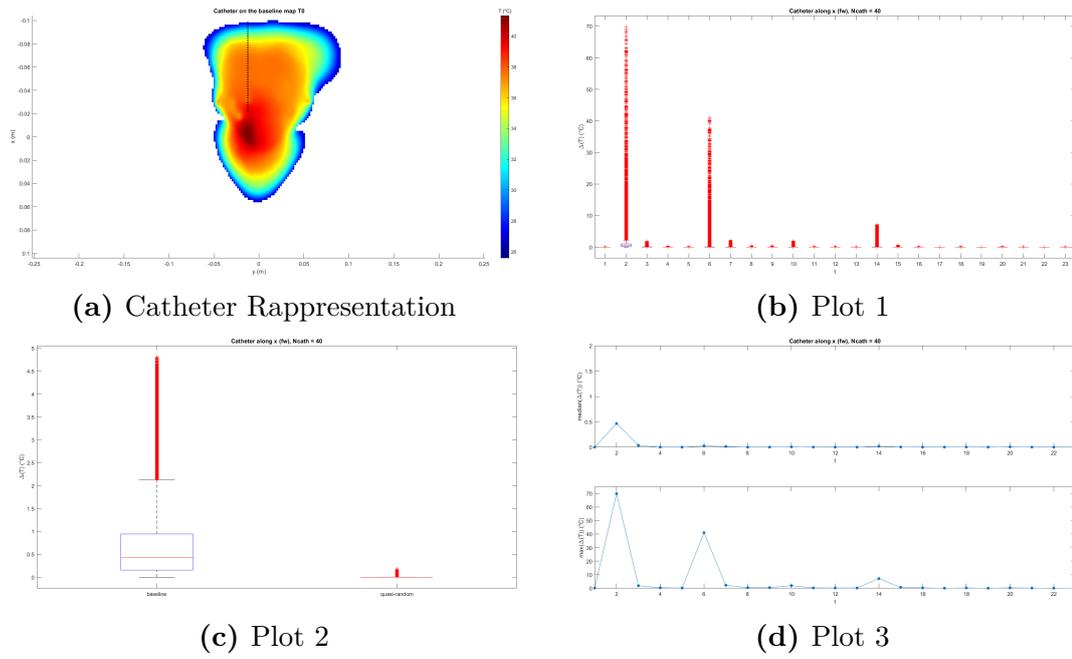


Figure 3.7: Case 3

Case 4

- Coordinates: $x = -12.36, y = 0.00$ [mm]
- Direction: along x
- Type: reverse
- N.Points: 20
- Step: 2 [mm]

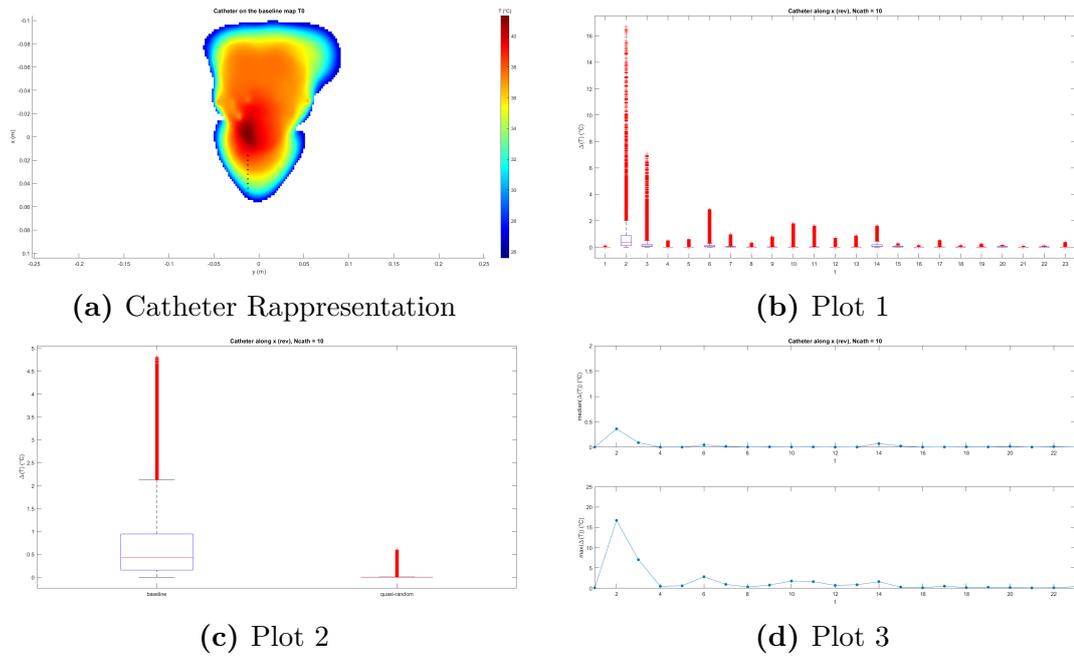


Figure 3.8: Case 4

Chapter 4

Reconstruction Fidelity Evaluation

Chapter 3 introduced some algorithms, both in MATLAB and Python, used to change the tissue values inside the thermal solver and produce the corresponding temperature maps. Firstly, the multigrid generation algorithm, in MATLAB environment, that creates several combinations of parameter values, varying them inside the range provided in the Literature. Then, thanks to the Python script, these are iteratively, each set at a time, attached to the human phantom in Sim4Life and the bio-heat equation is solved forming the associated maps. The reconstruction algorithm is performed as the last step, creating the S_{var} and S_{cath} matrices, including different catheter conditions and the related statistics.

Now the performance of the reconstruction algorithm is evaluated producing results from both the MGrid Type A and MGrid Type B, modifying more parameters, including more tissues, and varying the minimum number of temperature maps involved in the reconstruction.

4.1 Multigrid Type Comparison

In the section above, different 4 cases of catheter insertion are plotted. These derive from the reconstruction algorithm performed with the creation of a MGrid Type A + Baseline made by 27 combination sets, modifying thermal parameters of tumor and muscle.

Similar analysis is conducted here where the two different multigrid typologies are compared. For both multigrids, Type A + Baseline and Type B + Baseline, 25 combinations (four-tuples) of parameters are created using the MATLAB script, aiming to change k and ω of *tumor* and *muscle*. As derived from the equation (3.2), the Type A + Baseline is formed by the baseline values as a first set, 16

combinations created from the extreme bounds of the parameters and completed with 8 Sobol-based sets (Table 4.1a). On the other hand, the Type B + Baseline is made by all pseudo-random values generated from the literature ranges except the first row which is the baseline (see the equation (3.4)), reported in Table 4.1b. After that, the Python script is run twice to assign the parameters combinations of each multigrid to the segmented tissues in the Sim4Life simulation, producing the temperature maps. Then, the maps are exported in MATLAB where the reconstruction algorithm is executed for the two map types, creating 4 different cases. All these cases, reported below with the statistic plots described in Section 3.2.3, are formed by altering the insert position of the catheter which is made by 20 points spaced 1mm.

Map	k_m	k_t	ω_m	ω_t	Map	k_m	k_t	ω_m	ω_t
	[W m ⁻¹ K ⁻¹]	[W m ⁻¹ K ⁻¹]	[ml kg ⁻¹ min ⁻¹]	[ml kg ⁻¹ min ⁻¹]		[W m ⁻¹ K ⁻¹]	[W m ⁻¹ K ⁻¹]	[ml kg ⁻¹ min ⁻¹]	[ml kg ⁻¹ min ⁻¹]
T0	0.49	0.51	39.10	72.30	T0	0.49	0.51	39.10	72.30
T1	0.40	0.48	39.10	72.30	T1	0.48	0.99	240.55	460.15
T2	0.56	0.48	39.10	72.30	T2	0.44	1.25	139.83	654.08
T3	0.40	1.50	39.10	72.30	T3	0.52	0.74	341.28	266.23
T4	0.56	1.50	39.10	72.30	T4	0.42	1.12	391.64	751.04
T5	0.40	0.48	44.20	72.30	T5	0.50	0.61	190.19	363.19
T6	0.56	0.48	44.20	72.30	T6	0.46	0.86	290.91	169.26
T7	0.40	1.50	44.20	72.30	T7	0.54	1.37	89.46	557.11
T8	0.56	1.50	44.20	72.30	T8	0.41	1.44	316.09	314.71
T9	0.40	0.48	39.10	848.00	T9	0.49	0.93	114.64	702.56
T10	0.56	0.48	39.10	848.00	T10	0.45	0.67	416.82	508.63
T11	0.40	1.50	39.10	848.00	T11	0.53	1.18	215.37	120.78
T12	0.56	1.50	39.10	848.00	T12	0.43	0.79	165.01	605.59
T13	0.40	0.48	442.00	848.00	T13	0.51	1.31	366.46	217.74
T14	0.56	0.48	442.00	848.00	T14	0.47	1.05	64.28	411.67
T15	0.40	1.50	442.00	848.00	T15	0.55	0.54	265.73	799.52
T16	0.56	1.50	442.00	848.00	T16	0.41	1.02	202.78	241.98
T17	0.48	0.99	240.55	460.15	T17	0.49	0.51	404.23	629.83
T18	0.44	1.25	139.83	654.08	T18	0.45	0.77	102.05	823.76
T19	0.52	0.74	341.28	266.23	T19	0.53	1.28	303.50	435.91
T20	0.42	1.12	391.64	751.04	T20	0.43	0.64	253.14	726.79
T21	0.50	0.61	190.19	363.19	T21	0.51	1.15	51.69	338.95
T22	0.46	0.86	290.91	169.26	T22	0.47	1.40	353.87	145.02
T23	0.54	1.37	89.46	557.11	T23	0.55	0.89	152.42	532.88
T24	0.41	1.44	316.09	314.71	T24	0.42	0.96	379.05	387.43

(a) MGrid Type A + Baseline

(b) MGrid Type B + Baseline

Table 4.1: Multigrids Values, k and ω of *muscle* and *tumor*

Type A - Case 1

- Coordinates: $x = -12.36, y = 0.00$ [mm]
- Direction: along x
- Type: reverse
- N.Points: 20
- Step: 1 [mm]

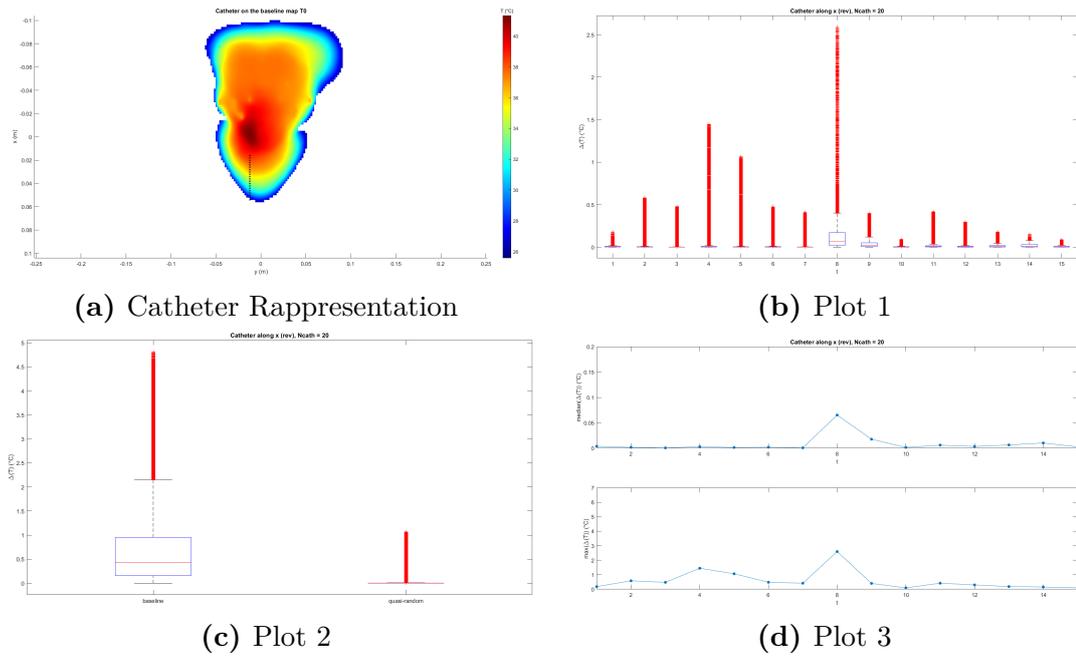


Figure 4.1: Case 1

Type B - Case 1

- Coordinates: $x = -12.36, y = 0.00$ [mm]
- Direction: along x
- Type: reverse
- N.Points: 20
- Step: 1 [mm]

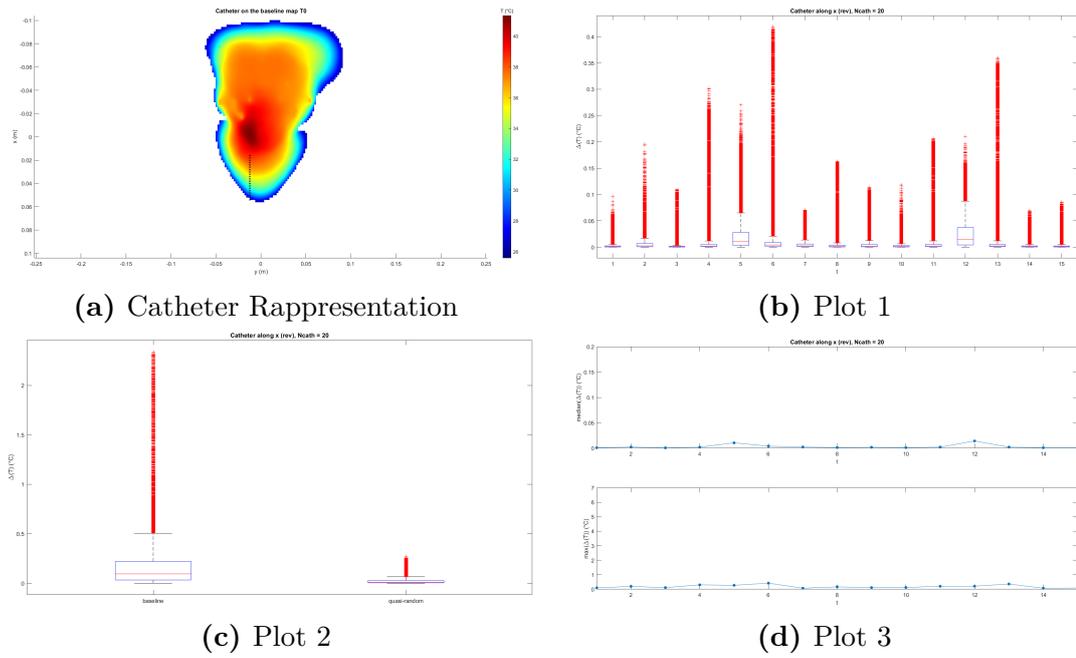


Figure 4.2: Case 1

Type A - Case 2

- Coordinates: $x = -12.36, y = 0.00$ [mm]
- Direction: along x
- Type: forward
- N.Points: 20
- Step: 1 [mm]

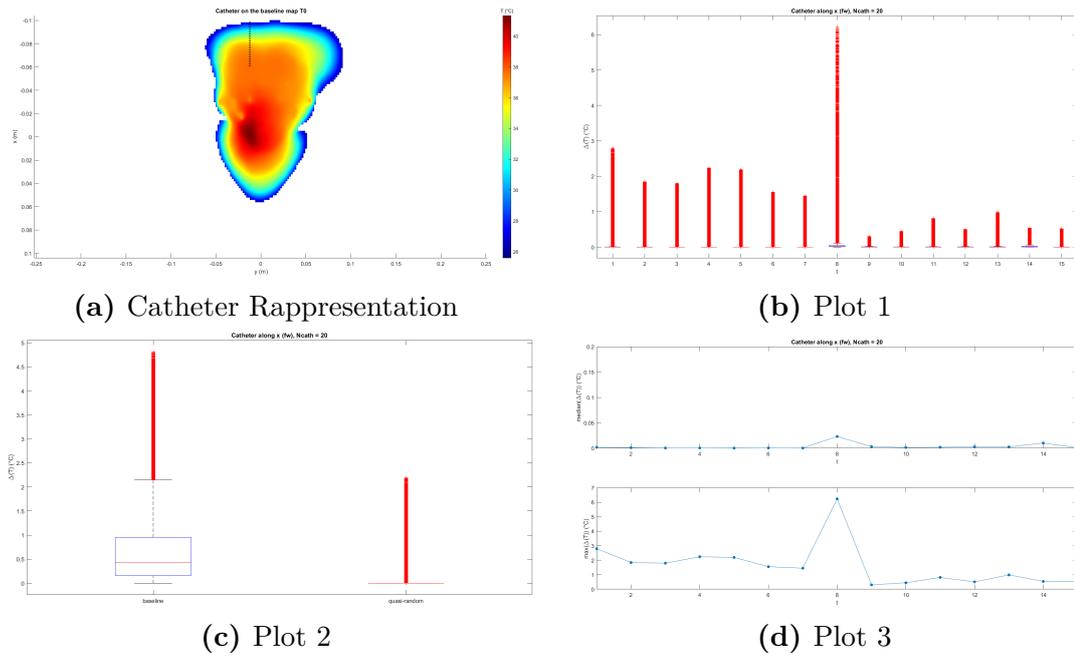


Figure 4.3: Case 2

Type B - Case 2

- Coordinates: $x = -12.36, y = 0.00$ [mm]
- Direction: along x
- Type: forward
- N.Points: 20
- Step: 1 [mm]

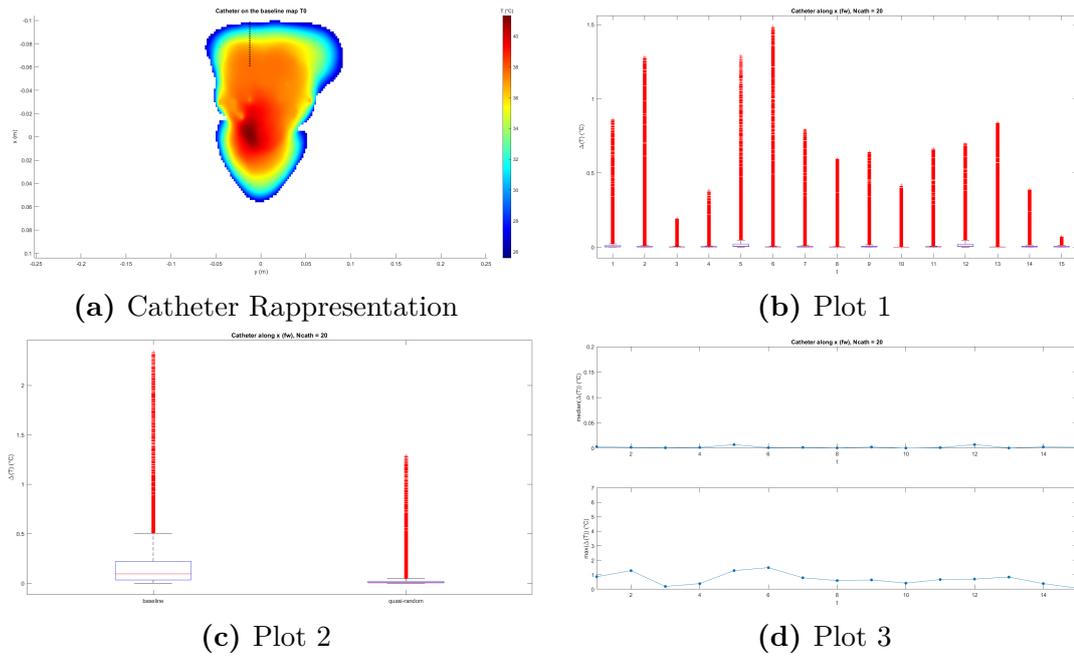


Figure 4.4: Case 2

Type A - Case 3

- Coordinates: $x = 0.00, y = -3.33$ [mm]
- Direction: along y
- Type: reverse
- N.Points: 20
- Step: 1 [mm]

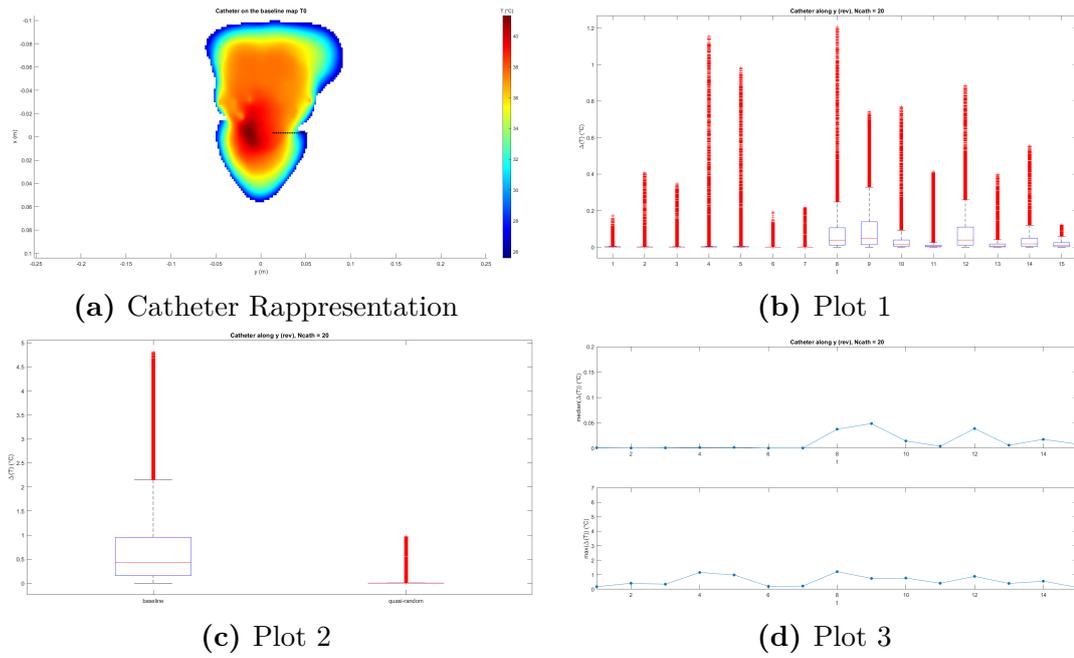


Figure 4.5: Case 3

Type B - Case 3

- Coordinates: $x = 0.00, y = -3.33$ [mm]
- Direction: along y
- Type: reverse
- N.Points: 20
- Step: 1 [mm]

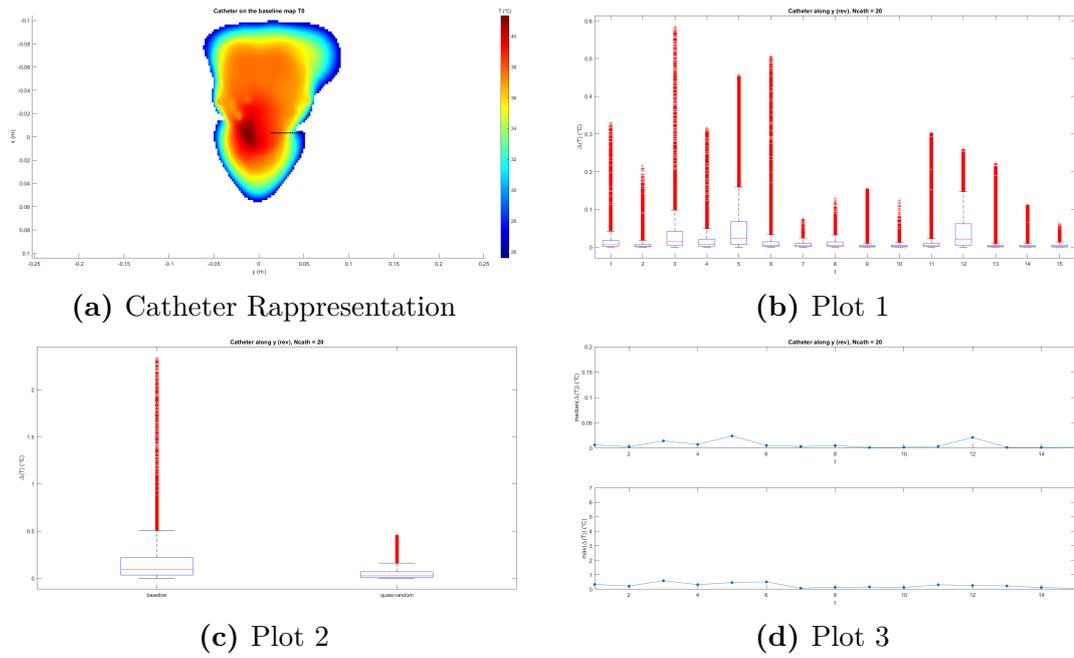


Figure 4.6: Case 3

Type A - Case 4

- Coordinates: $x = 0.00, y = -3.33$ [mm]
- Direction: along y
- Type: forward
- N.Points: 20
- Step: 1 [mm]

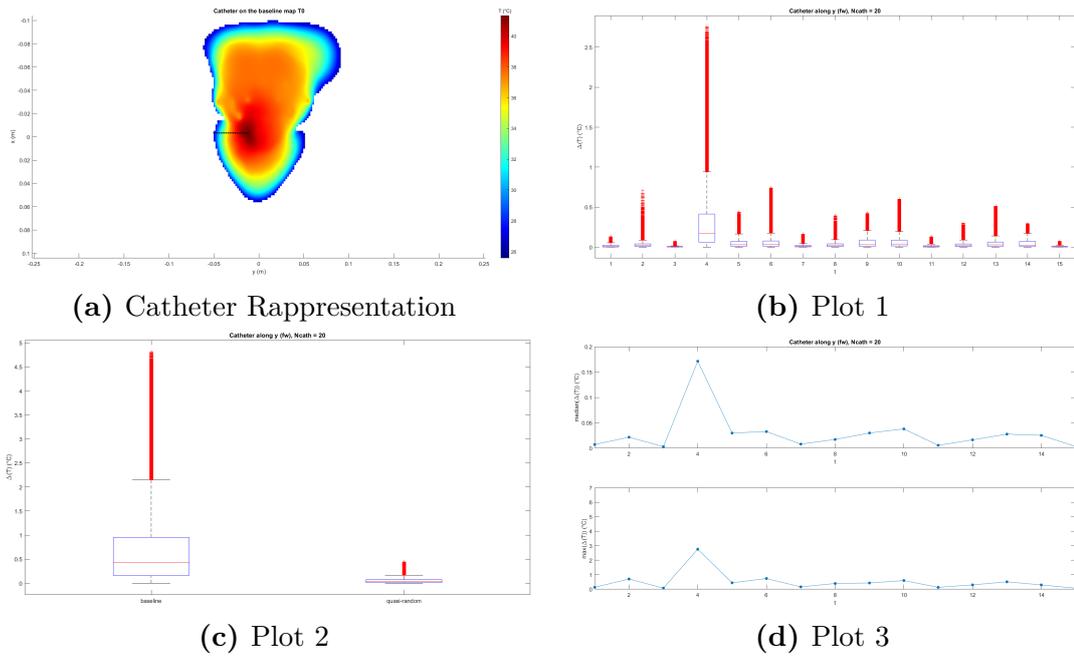


Figure 4.7: Case 4

Type B - Case 4

- Coordinates: $x = 0.00$, $y = -3.33$ [mm]
- Direction: along y
- Type: forward
- N.Points: 20
- Step: 1 [mm]

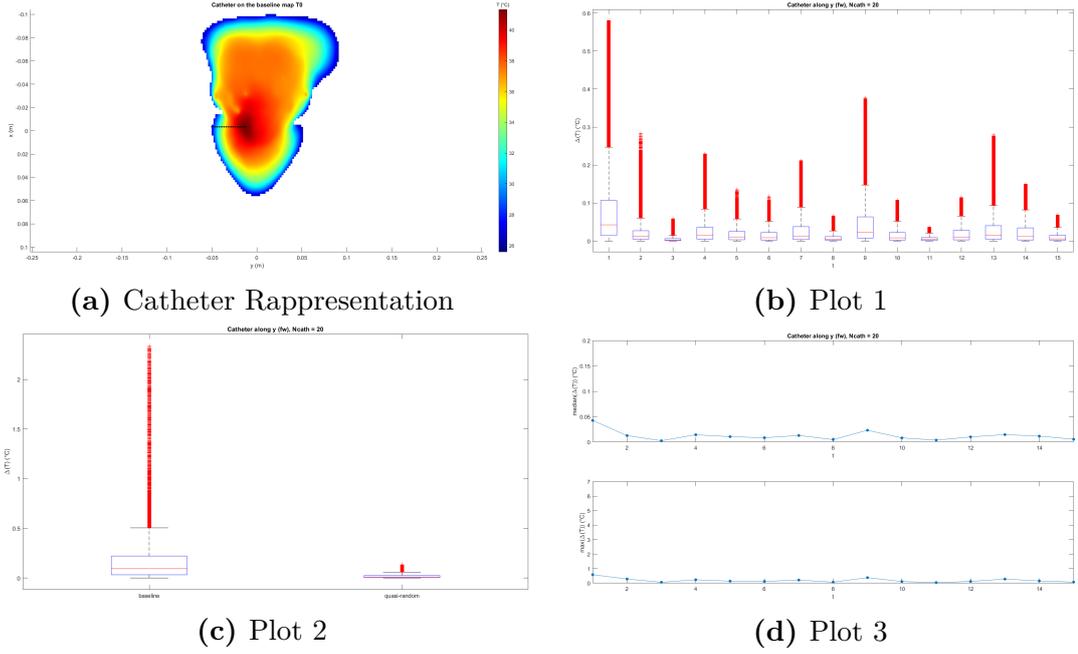
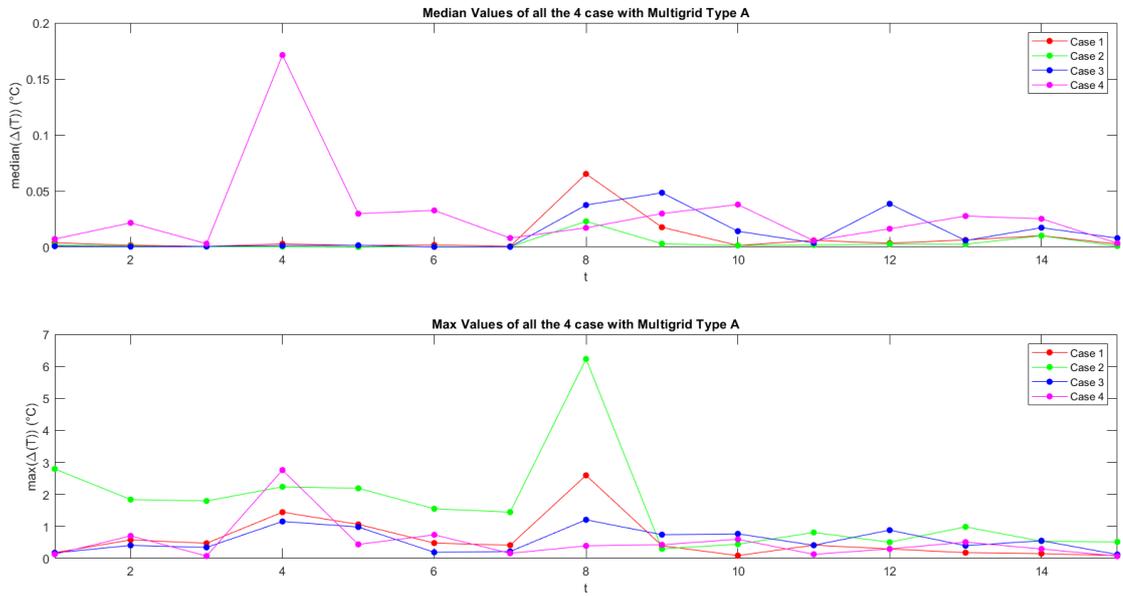
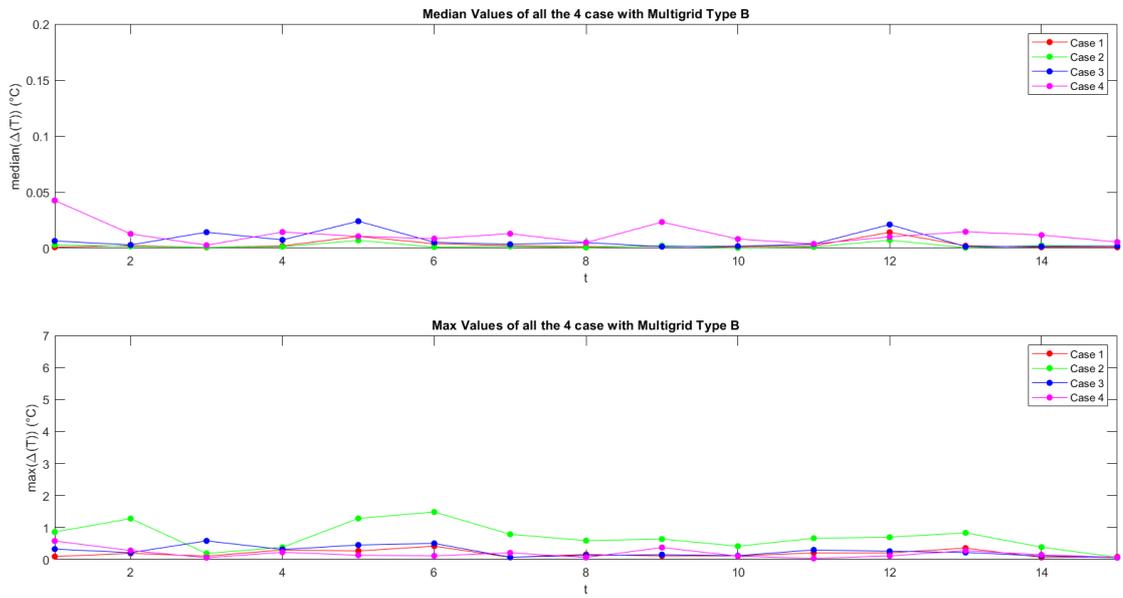


Figure 4.8: Case 4

One more specific statistical comparison is done. Figure 4.9, which considers all four cases for each multigrid type, plots the ΔT median value and maximum ΔT calculated in each case. It is possible to notice, looking at Type A graphs, that the highest maximum error occurs in case 2. This could be expected since in this case the catheter is inserted very far from the tumor region, therefore it does not provide information on the variation of the parameters in that region. For a similar reason, it is quite logical that the highest mean error occurs in case 4 where the catheter direction passes through the tumor, measuring in this scenario the fewest points in the muscle tissue. On the other hand, comparing the two multigrid types it is evident that temperature maps created by multigrid Type B have globally lower values than in the other case. As a result, for this first analysis MGrid Type B seems to be the most convenient.



(a) MGrid Type A + Baseline



(b) MGrid Type B + Baseline

Figure 4.9: 4 Cases Comparison

4.2 Svar matrix Dimension

This section aims at estimating the correlation between the reconstruction fidelity and the number of columns used for the reconstruction algorithm. Thus, some reconstructions are made multiple times, with the same target, varying the number of the minimum maps $N_{p,min}$ taken from the S_{var} matrix and included in the algorithm for the reconstruction. Two parallel investigations are done for both the multigrid types.

In this analysis, target fields are set the as the last 5 temperature maps taken from the S_{cath} matrix, created from MGrid Type B using the same catheter condition present in case 4 (see Figure 4.8), while the minimum number of temperature maps $N_{p,min}$ required for the reconstruction is increased iteratively. Each target field is reconstructed several times and, at each step, one more temperature map is added to the minimum number of maps required for the reconstruction. For each value of $N_{p,min}$, the corresponding mean and maximum error values in the whole region are calculated. Case 4 is taken as the catheter condition being the only case that reaches several points in the tumor region. Besides, the S_{cath} is created from the multigrid Type B because is random generated so more adequate for the assessment.

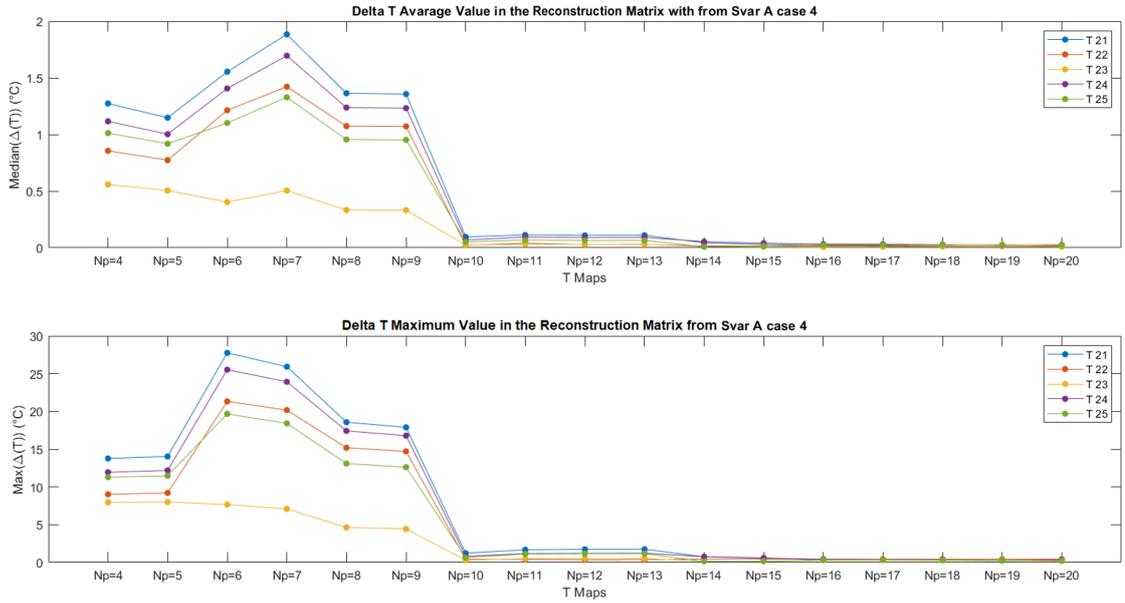
4.2.1 Thermal Parameters Only

The tissue parameters characterized by the higher uncertainty are the thermal parameters k and ω , so the analysis starts with them.

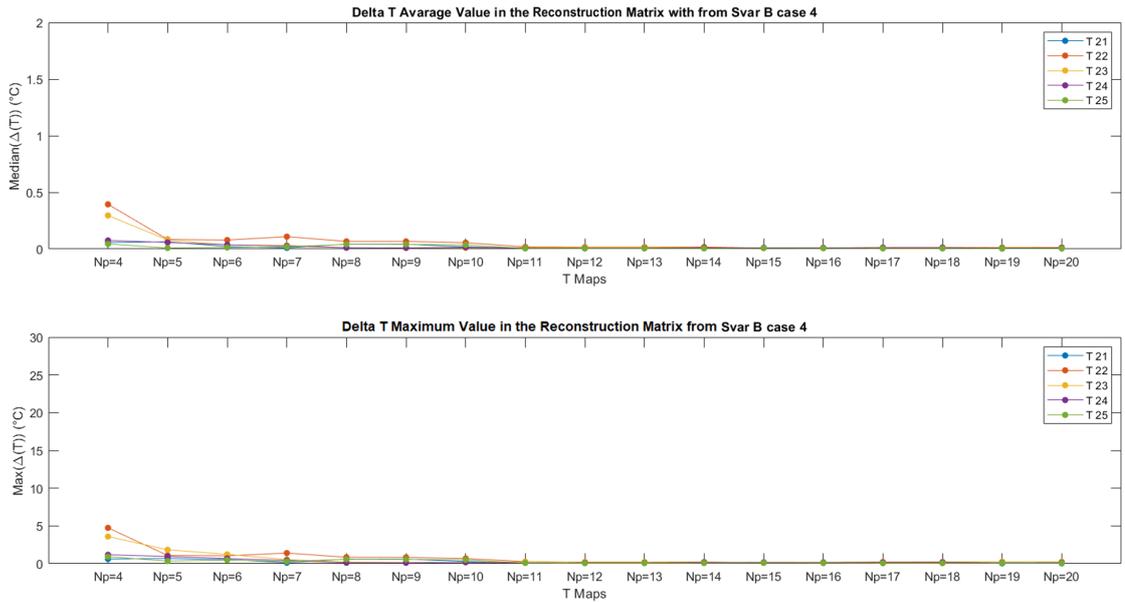
4.2.1.1 Multigrid with and without Baseline

With the aim of changing thermal conductivity k and perfusion ω in *tumor* and *muscle*, the combination set types reported in Table 4.1 are used. Because of the previous section, the related S_{var} matrix and S_{cath} matrices, with 25 maps each, are already created for the two multigrids including the baseline. From the latter, the Code 4.3 is run to start the analysis. To improve the comprehension, it plots both the trend of maximum and median of the ΔT values related to the different number of maps $N_{p,min}$ used to reconstruct each target fields, which are the last five maps of the S_{cath} MGrid Type B as mentioned above. Figure 4.10, which compares the results for both different multigrids, shows that for all the ΔT error types there is a decreasing trend: the higher the number of maps $N_{p,min}$ considered in the reconstruction matrix, the lower the errors achieved. The significative drop for both ΔT errors in Figure 4.10a, referred to MGrid type A + Baseline, occurs after 10 maps involved ($N_{p,min}=10$). The higher ΔT , both in maximum and median values, are related to the algorithm that used the S_{var} created with the MGrid

Type A + Baseline.



(a) MGrid Type A + Baseline



(b) MGrid Type B + Baseline

Figure 4.10: $N_{p,min}$ Statistics Comparison varying k and ω of *muscle* and *tumor*

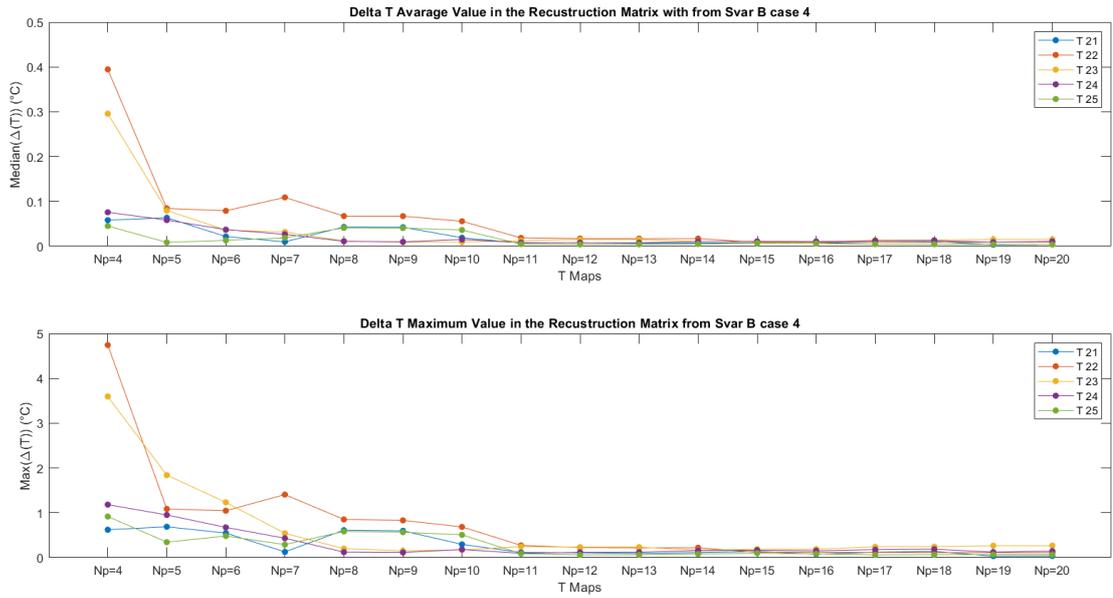
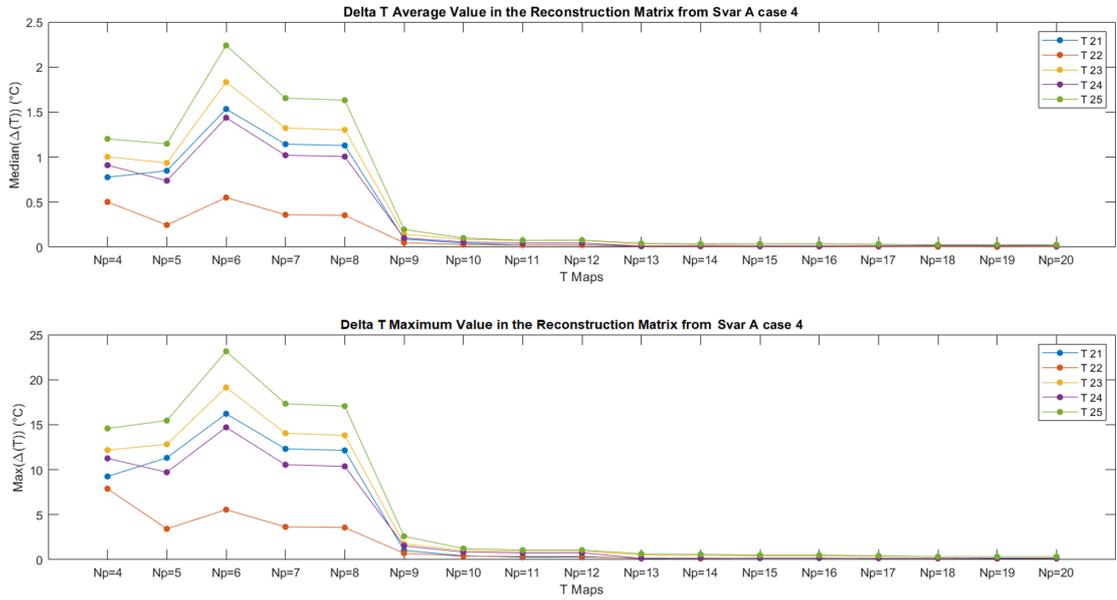


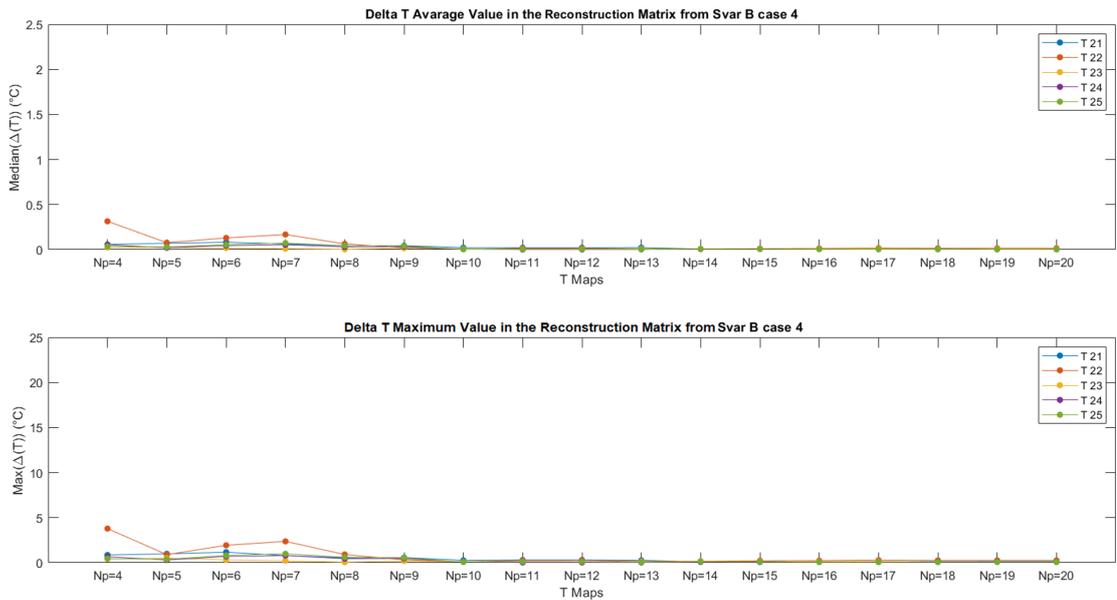
Figure 4.11: MGrid Type B + Baseline.

The limits on the y -axis are set to show more clearly the values reached.

The same analysis is done using MGrid Type A and MGrid Type B without adding the baseline values. These two are created from MATLAB script as before, modifying the number of Sobol points in order to obtain also here 25 maps, and all the passages explained in Section 3 are repeated. After the exportation in MATLAB, the reconstruction algorithm is run, creating new S_{var} and S_{cath} matrices and providing the results plotted in Figure 4.12.



(a) MGrid Type A Without Baseline



(b) MGrid Type B Without Baseline

Figure 4.12: $N_{p,min}$ Statistics Comparison varying k and ω of muscle and tumor

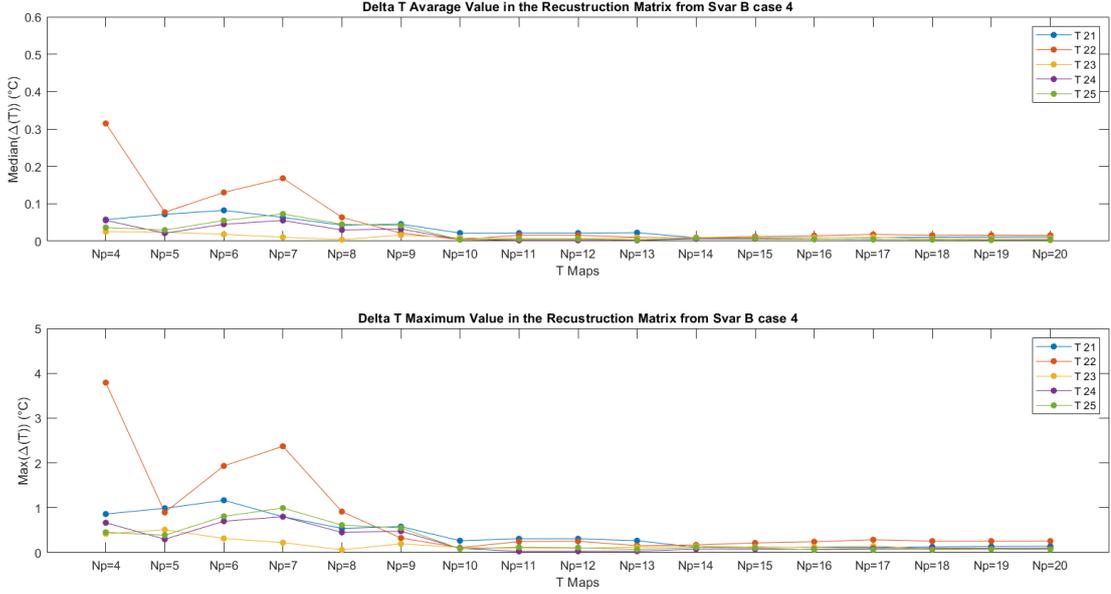


Figure 4.13: MGrid Type B Without Baseline.

The limits on the y -axis are set to show more clearly the values reached.

Figure 4.12a displays two ΔT trendlines similar to the to the previous case reported in Fig. 4.10, with lower values in the MGrid Type B condition. However, in this image the drops in the S_{var} created from MGrid Type A occurs at $N_{p,min}=9$ instead of 10. The only difference between this case and the previous one is the baseline row which in the present case is absent, so in this S_{var} it is realistic that everything is shifted one row behind. Values below 1 °C in the max ΔT in MGrid Type B plot (Fig.4.13) are immediately obtained for almost all the target fields except for T_{22} .

4.2.1.2 Adding Fat Tissue

In order to enlarge the analysis, the *fat* and also the *SAT*, as mentioned in Section 3.1, tissues are added. Both the multigrids are created by adding the baseline values like before but, including one more tissue, their dimension increase and consequently the number of sets. From equation (3.1), the MGrid Type A has a minimum number of combinations 2^N which is related to dimension. Particularly, by increasing the number of tissues $n = 3$, the minimum number of combination is calculated, according to equation (3.2), as:

$$\begin{aligned}
 N_{tot} &= 1 + 2^N + N_S \\
 &= 1 + 2^{3-2} + 5 \\
 &= 1 + 64 + 5 = 70
 \end{aligned} \tag{4.1}$$

by adding the baseline values and 5 Sobol points the total sets number results $N_{tot} = 70$.

The same number of combinations for the MGrid Type B + Baseline is reached by using 69 Sobol points plus the baseline values included in the first row, according to the equation (3.4).

The steps listed in Section 3 are then replicated, exporting 70 maps for each multigrid. S_{var} and S_{cath} matrices, with the same catheter conditions used in Section 3.2.3, are created from the two maps group in a domain reduced along the z -axis reduced z domain to narrow the analysis closer to the tumor region (see Figure 4.14); from them the reconstruction algorithm is performed. The last five maps of the matrix S_{cath} , formed by the MGrid Type B + Baseline, are taken as target values like mentioned in Section 4.2. The plots, created by the algorithm for this analysis, are shown in Figure 4.15.

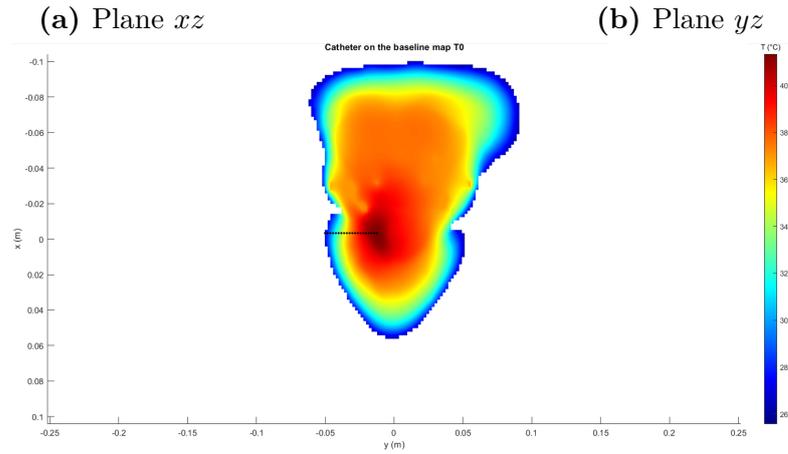
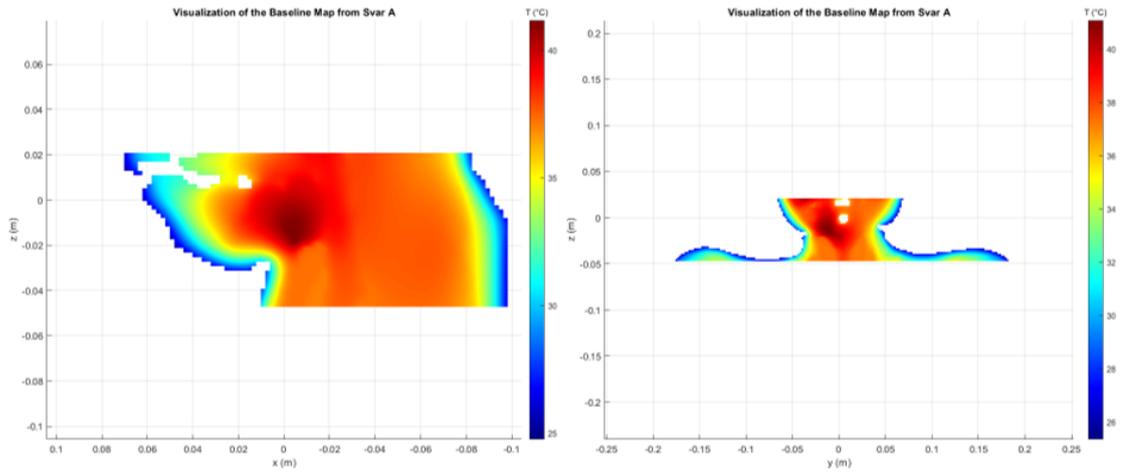
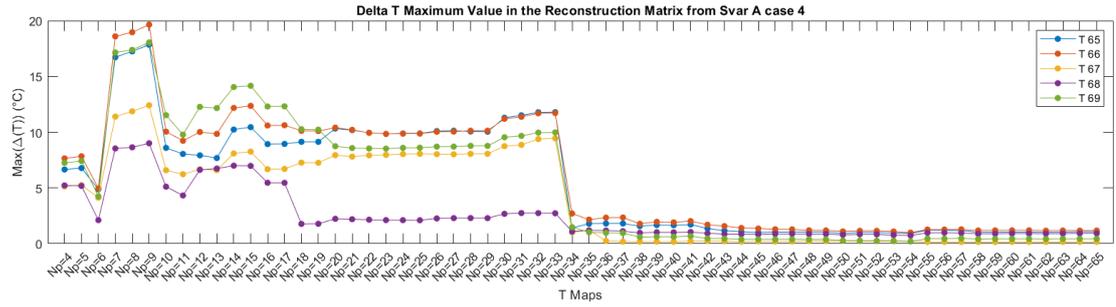
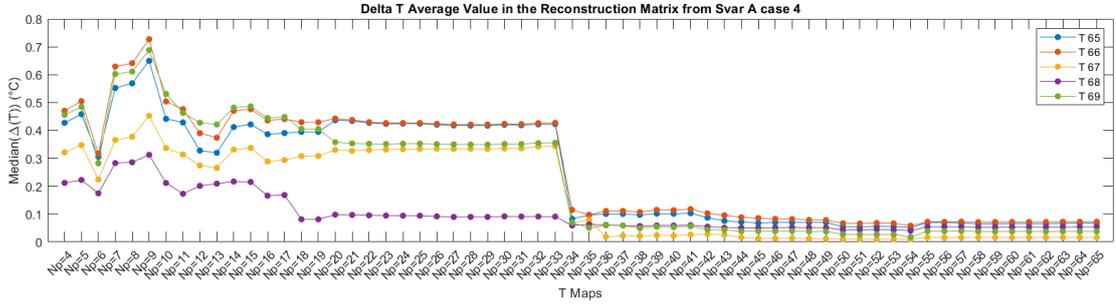
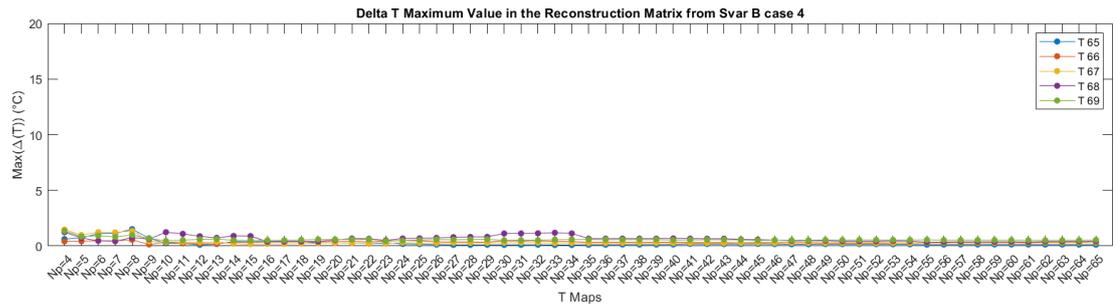
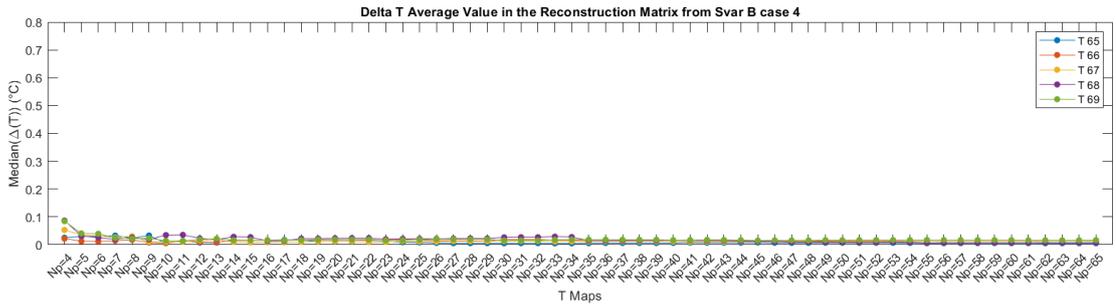


Figure 4.14: T0 Map Visualization



(a) MGrid Type A + Baseline



(b) MGrid Type B + Baseline

Figure 4.15: $N_{p,min}$ Statistics Comparison varying k and ω of *muscle*, *tumor* and *fat + SAT*

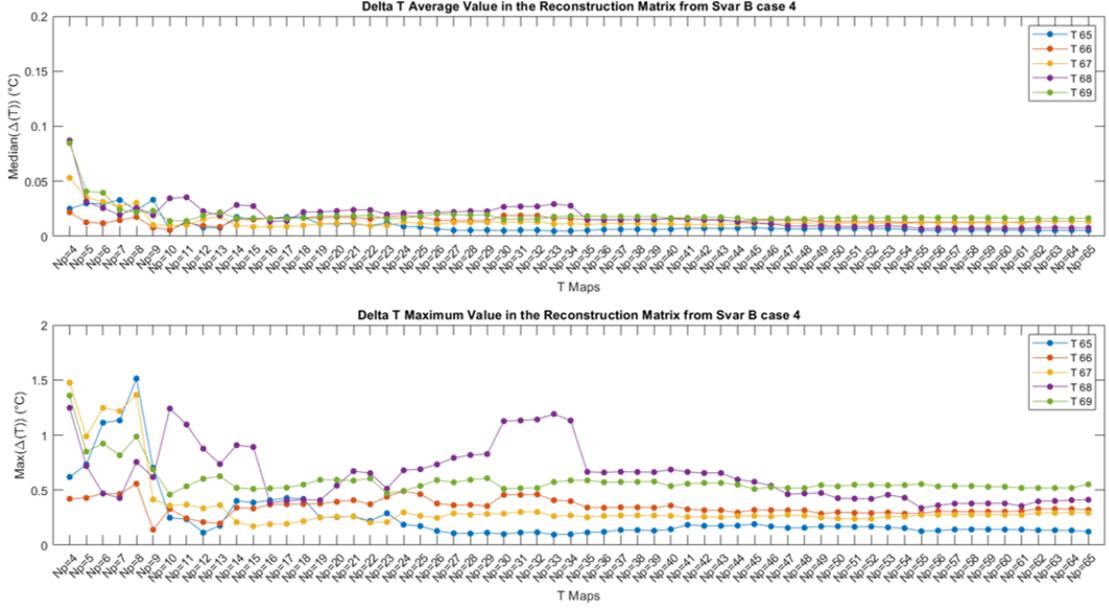


Figure 4.16: MGrid Type B + Baseline.

The limits on the y -axis are set to show more clearly the values reached.

Even if the number of total maps is greater than the case above, also here we have a tendentially decreasing trend in both multigrids. In MGrid Type A there are critical drops for both the errors like before, here around $N_{p,min}=34$ number of maps involved in the reconstruction, so almost in the half of the total maps number. MGrid maximum error, on the other hand, remains low for all the maps, reaching acceptable values, so ΔT max around 1 °C, after just few maps added.

4.2.1.3 Adding Skin Tissue

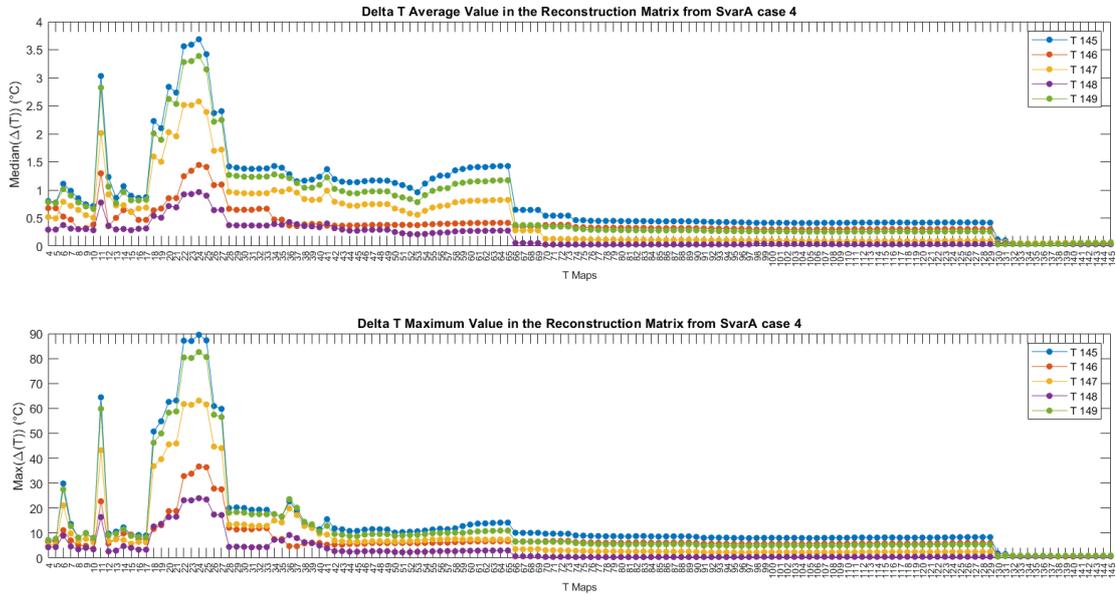
To complete the investigation in all the most relevant tissues in head and neck region, *skin* is included. Creating as usual the two multigrids and repeating the previous steps, it is possible to change the thermal parameter values for all the four tissues. For MGrid Type A + Baseline, the number of combinations is:

$$\begin{aligned}
 N_{tot} &= 1 + 2^N \\
 &= 1 + 2^{4 \cdot 2} \\
 &= 1 + 256 = 257
 \end{aligned}
 \tag{4.2}$$

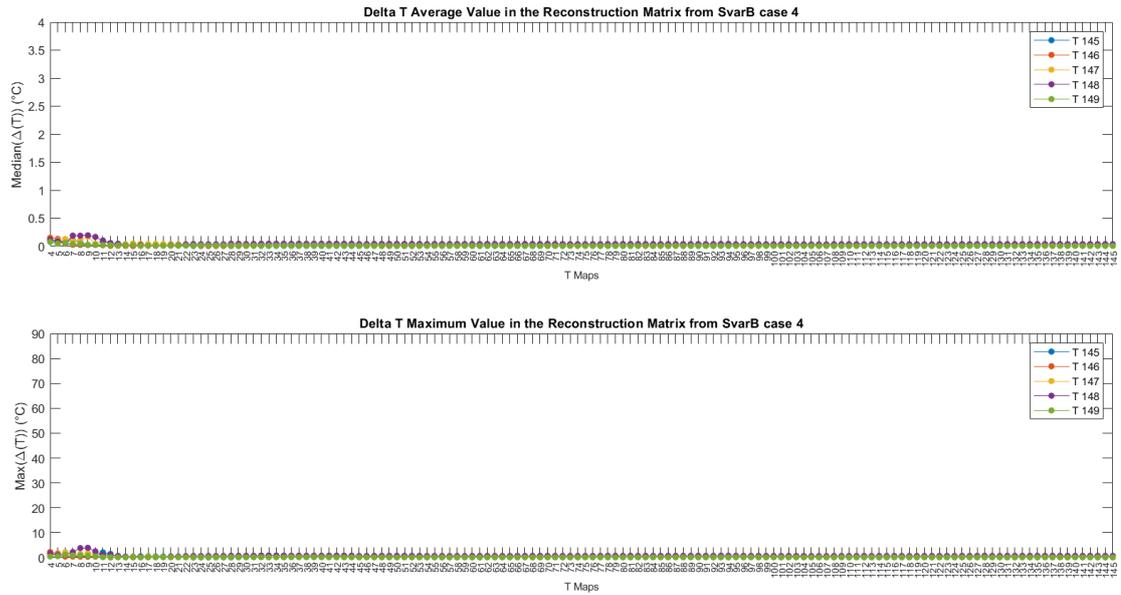
In this specific case, due to the high number of maps that will be created, the first 150 combinations are selected with the aim of reducing the computational costs and verifying the goodness of the reconstruction considering a reasonable

number of maps. For MGrid Type B + Baseline, 149 Sobol points are added to the baseline reaching the same number of combinations of the other multigrad.

Firstly, the Python script and then the reconstruction algorithm are applied with the same settings enounced in the previous case, creating the graphs reported in Figure 4.17.



(a) MGrid Type A + Baseline



(b) MGrid Type B + Baseline

Figure 4.17: $N_{p,min}$ Statistics Comparison varying k and ω of *muscle*, *tumor*, *fat + SAT* and *skin*

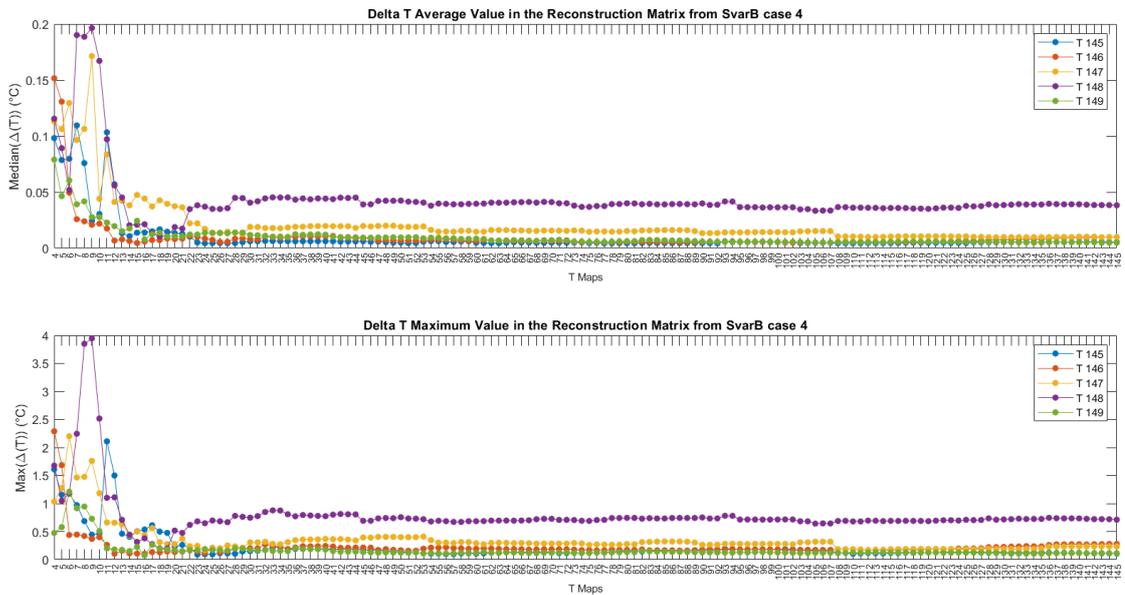


Figure 4.18: MGrid Type B + Baseline.

The limits on the y-axis are set to show more clearly the values reached.

The two multigrids present the decreasing trend and MGrid Type B results to have lower errors like above. One difference here is that in MGrid Type A there are two different drops: one at almost 60 temperature maps involved and the other around the 120. It is evident in MGrid Type A that acceptable ΔT max values are achieved with a large number of maps, over the 80% of the total maps. Otherwise, in MGrid Type B the maximum errors drop below 1 °C after considering only 20 maps.

4.2.2 Dielectric Parameters

In the previous Subsection 4.2.1, the thermal parameters are examined for four tissues in the H&N region due to their direct connection with the temperature maps. Nevertheless, as equation (2.14) explains, the deposition of EM energy in biological tissues depends on the dielectric properties of the tissues involved.

The first electric parameter that is chosen and altered is the *electric conductivity* σ , since it affects both the EM and thermal solvers (see equations (2.26) and (2.27)). The following analyses involved only the two most important tissues for the HTP which are the *muscle*, the most present, and *tumor* which is the target tissue. The different set combinations are created only with the MGrid Type B, because is the one that comprehensively provides the lowest error. To create the same number of maps generated in Section 4.2.1.1, where only two tissues are involved, MGrid Type B + Baseline is made by adding 24 Sobol points. In addition to the *electric conductivity* σ , the *perfusion* ω is also changed, forming an heterogeneous parameter analysis. Thanks to the MATLAB Script 4.3, appropriately modified to include the selected tissue properties, the sets in Table 4.2 are generated:

The electric conductivity has to be changed inside the EM solver, because is a dielectric tissue property, so a new Python script (see Codes 4.1 and 4.2) is written to:

1. Change the σ of the two tissues in the electromagnetic simulation.
2. Run the EM solver, where the antennas are normalized at $P0=60W$ and optimized in phases and amplitudes using the baseline values, and save the outcome
3. Change tissues perfusion inside the thermal solver.
4. Solve the bio-heat equation using the previous EM solver results as a source.
5. Export the related temperature maps in MATLAB.

Map	σ_m [S m ⁻¹]	σ_t [S m ⁻¹]	ω_m [ml kg ⁻¹ min ⁻¹]	ω_t [ml kg ⁻¹ min ⁻¹]
T0	0.81	0.89	39.10	72.30
T1	0.81	0.89	240.55	460.15
T2	0.79	0.91	139.83	654.08
T3	0.83	0.87	341.28	266.23
T4	0.78	0.90	391.64	751.04
T5	0.82	0.86	190.19	363.19
T6	0.80	0.88	290.91	169.26
T7	0.84	0.92	89.46	557.11
T8	0.77	0.93	316.09	314.71
T9	0.81	0.88	114.64	702.56
T10	0.79	0.86	416.82	508.63
T11	0.83	0.91	215.37	120.78
T12	0.78	0.87	165.01	605.59
T13	0.82	0.92	366.46	217.74
T14	0.80	0.90	64.28	411.67
T15	0.84	0.85	265.73	799.52
T16	0.77	0.89	202.78	241.98
T17	0.81	0.85	404.23	629.83
T18	0.79	0.87	102.05	823.76
T19	0.83	0.92	303.50	435.91
T20	0.78	0.86	253.14	726.80
T21	0.82	0.90	51.69	338.95
T22	0.80	0.93	353.87	145.02
T23	0.84	0.88	152.42	532.87
T24	0.77	0.89	379.05	387.43

Table 4.2: MGrid Type B + Baseline σ and ω of *muscle* and *tumor*

```

for i in range (0, num_rows):

    sm = coord[i,0]
    st = coord[i,1]
    wm = coord[i,2]
    wt = coord[i,3]

    print (sm, st, wm, wt)
    #Define the version to use for default values
    ReleaseVersion.set_active(ReleaseVersion.version7_0)
    simulation = document.AllSimulations["EM 1 - Copy"]
    simulation.ClearResults()
    simulation.ResetVoxels()

    entity__muscle = model.AllEntities()["Muscle"]
    muscle_new_s_value = sm
    muscle_new_w_value = wm, Unit("ml/min/kg")
    entity__tumor = model.AllEntities()["Tumor"]
    tumor_new_s_value = st
    tumor_new_w_value = wt, Unit("ml/min/kg")

    #Change the Mass Density value of an existing material
    sets = simulation.AllSettings

```

```

for idx, set in enumerate(sets):
    if set.Name == 'Muscle':
        set.ElectricConductivity=muscle_new_s_value
        #set.ElectricProps.Conductivity=muscle_new_s_value
    if set.Name == 'Tumor':
        set.ElectricConductivity=tumor_new_s_value
        #set.ElectricProps.Conductivity=tumor_new_s_value

#Update the materials with the new frequency parameters
simulation.UpdateAllMaterials()
#RUN
simulation.UpdateGrid()
simulation.CreateVoxels()
simulation.RunSimulation(wait=True)

simulation = document.AllSimulations["EM 1 - Copy"]
simulation_extractor = simulation.Results()

```

Code 4.1: Dielectric Parameter Variation

```

##Thermal Simulation
ReleaseVersion.set_active(ReleaseVersion.version7_0)
simulation1 = document.AllSimulations["Th 60 W - Dielectric"]
simulation1.ClearResults()
simulation1.ResetVoxels()

# Change the Mass Density value of an existing material
sets = simulation1.AllSettings
for idx, set in enumerate(sets):
    if set.Name == 'Muscle':
        set.HeatTransferRate.UsePerfusionUnits = True
        set.HeatTransferRate.PuConstantTerm = muscle_new_w_value
    if set.Name == 'Tumor':
        set.HeatTransferRate.UsePerfusionUnits = True
        set.HeatTransferRate.PuConstantTerm = tumor_new_w_value

simulation1.UpdateAllMaterials()
# Adding a new StationaryUserDefinedHeatSource

if i > 0:
    simulation1.Remove(stationary_user_defined_heat_source, components)

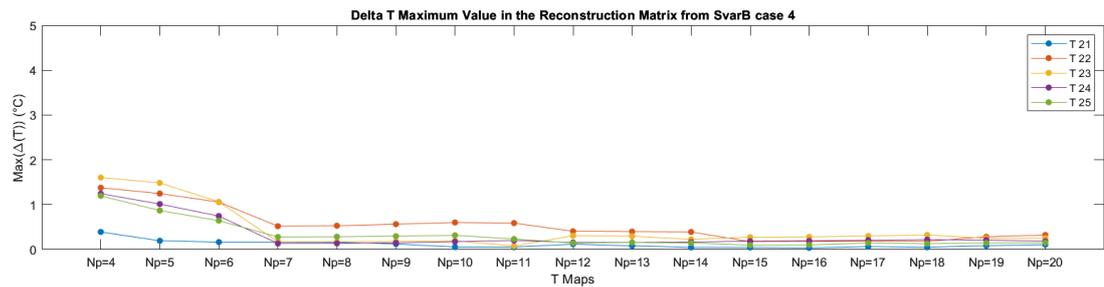
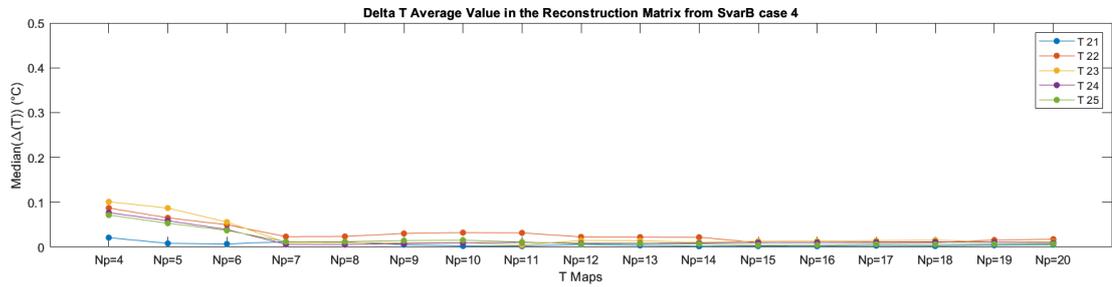
stationary_user_defined_heat_source = thermal.StationaryUserDefinedHeatSource
()
components = []
stationary_user_defined_heat_source.UserDefinedFileName = (dir + "Source{}.
cache".format(i+1))
simulation1.Add(stationary_user_defined_heat_source, components)

```

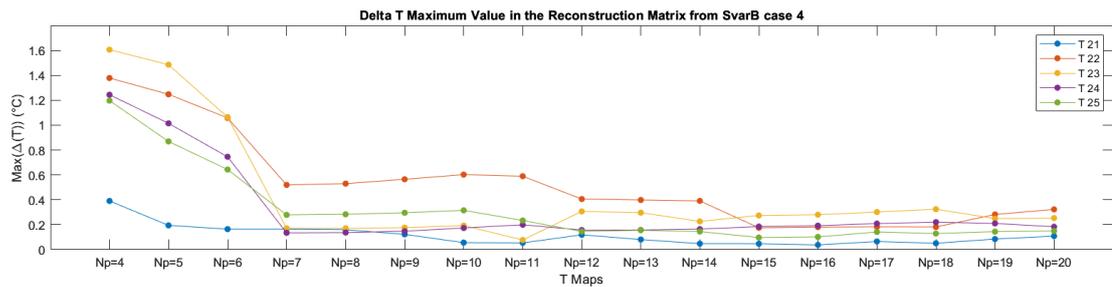
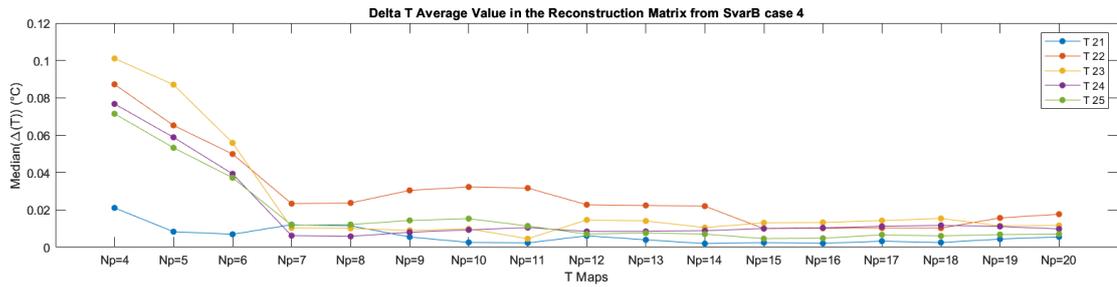
Code 4.2: Thermal Solver With the New EM Source

The reconstruction algorithm is then applied using as target fields the last 5 maps of the S_{cath} matrix. The results are displayed in Figure 4.19.

As shown in Figure 4.19, errors have a decreasing trend but not so marked due to the are very small global values. Despite that, a little drop is shown at $N_{p,min}=7$ for all the target values.



(a) $N_{p,min}$ Statistics with the Limits of the Section 4.2.1.1



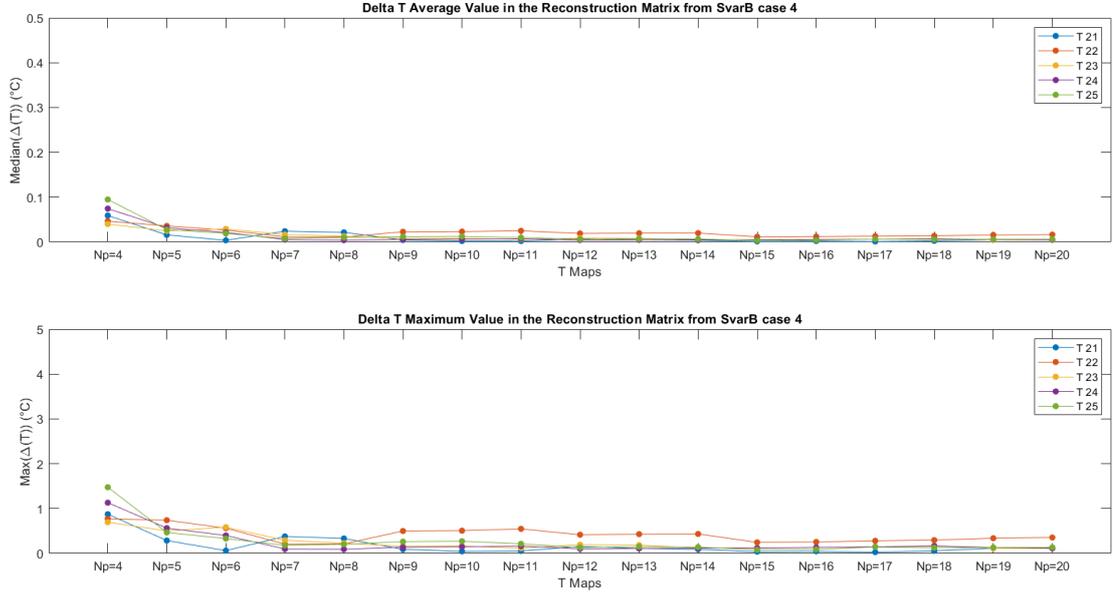
(b) $N_{p,min}$ Statistics Proper Limits

Figure 4.19: $N_{p,min}$ Statistics Comparison varying σ and ω of *muscle* and *tumor*

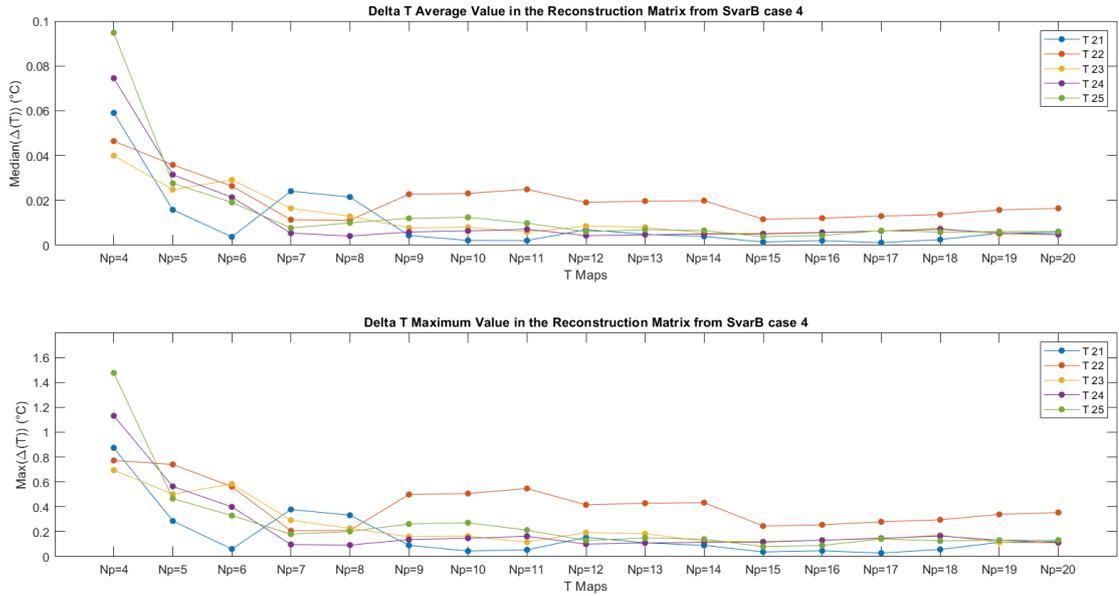
To complete the dielectric analysis, the *relative permittivity* ϵ_r is changed for the same tissues, also considering the variation of *perfusion* ω . Same steps are performed for this case, creating 25 sets through the MGrid Type B + Baseline (see Table 4.3), running the Python script with appropriate modifications for modifying the tissues properties, exporting the temperature maps and running the reconstruction algorithm. The visual outcome, reported in Figure 4.20, illustrates plots that are really similar to the previous one (Fig. 4.19) in terms of errors values, while the drop visualized before here is almost absent.

Map	ϵ_{rm}	ϵ_{rt}	ω_m [ml kg ⁻¹ min ⁻¹]	ω_t [ml kg ⁻¹ min ⁻¹]
T0	56.90	59.00	39.10	72.30
T1	56.85	59.00	240.55	460.15
T2	55.43	60.50	139.83	654.08
T3	58.28	57.50	341.28	266.23
T4	54.72	59.75	391.64	751.04
T5	57.56	56.75	190.19	363.19
T6	56.14	58.25	290.91	169.26
T7	58.99	61.25	89.46	557.11
T8	54.36	61.63	316.09	314.71
T9	57.21	58.63	114.64	702.56
T10	55.78	57.13	416.82	508.63
T11	58.63	60.13	215.37	120.78
T12	55.07	57.88	165.01	605.59
T13	57.92	60.88	366.46	217.74
T14	56.49	59.38	64.28	411.67
T15	59.34	56.38	265.73	799.52
T16	54.18	59.19	202.78	241.98
T17	57.03	56.19	404.23	629.83
T18	55.60	57.69	102.05	823.76
T19	58.45	60.69	303.50	435.91
T20	54.89	56.94	253.14	726.80
T21	57.74	59.94	51.69	338.95
T22	56.32	61.44	353.87	145.02
T23	59.17	58.44	152.42	532.87
T24	54.53	58.81	379.05	387.43

Table 4.3: MGrid Type B + Baseline ϵ_r and ω of *muscle* and *tumor*



(a) $N_{p,min}$ Statistics with the Limits of the Section 4.2.1.1



(b) $N_{p,min}$ Statistics Proper Limits

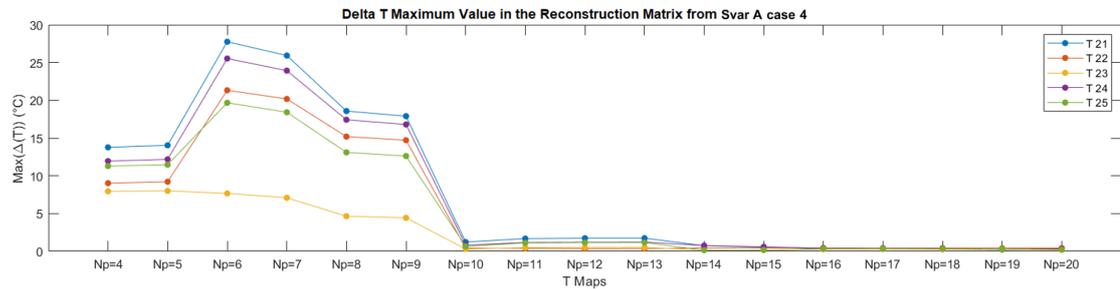
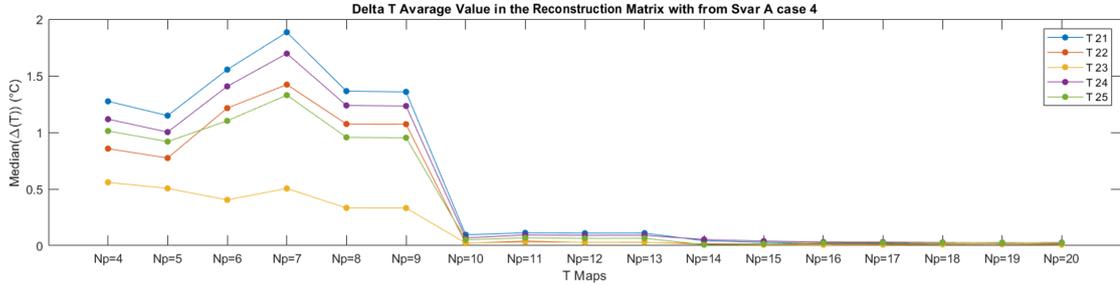
Figure 4.20: $N_{p,min}$ Statistics Comparison varying ϵ_r and ω of *muscle* and *tumor*

4.3 Discussion and Temperature Reconstruction

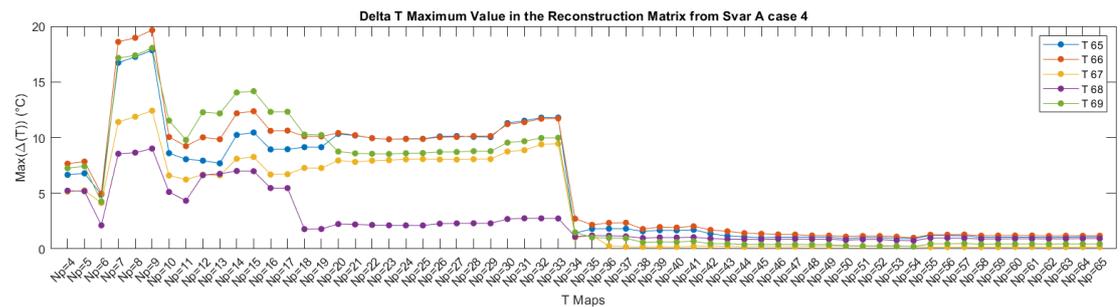
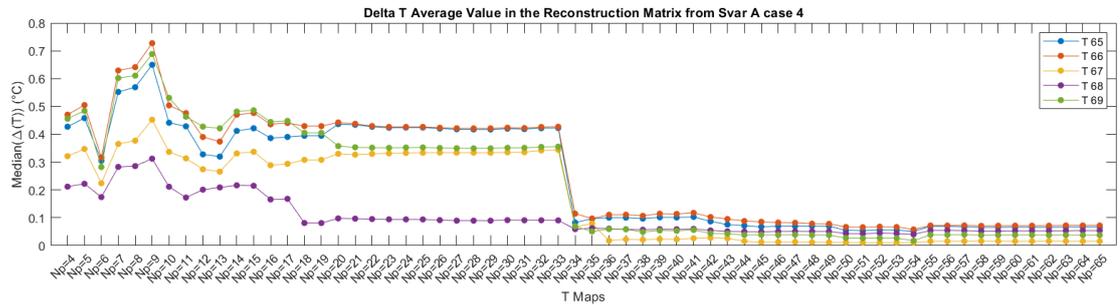
General observation can be done in all the cases presented, for thermal and dielectric parameter values variation, validating the algorithm employed.

Globally MGrid Type B has lower ΔT values compared to the MGrid Type A, the difference between the two is relevant especially when few maps are involved. As mentioned, the target maps are taken from Type B multigrid, which is formed only by pseudo-random values varying within the Literature ranges. The different nature of the MGrid Type A, which is composed mainly by the different combination of the extreme bounds of the ranges, results a weak strategy to recreate those random values due to the gap between the domain limits and its average content (see Figure 3.2 for a visual representation). On the other hand, even if MGrid Type B values inside the S_{var} are random-generated and so they may differ largely to the target ones, they remain closer to the targets rather than to the domain boundaries.

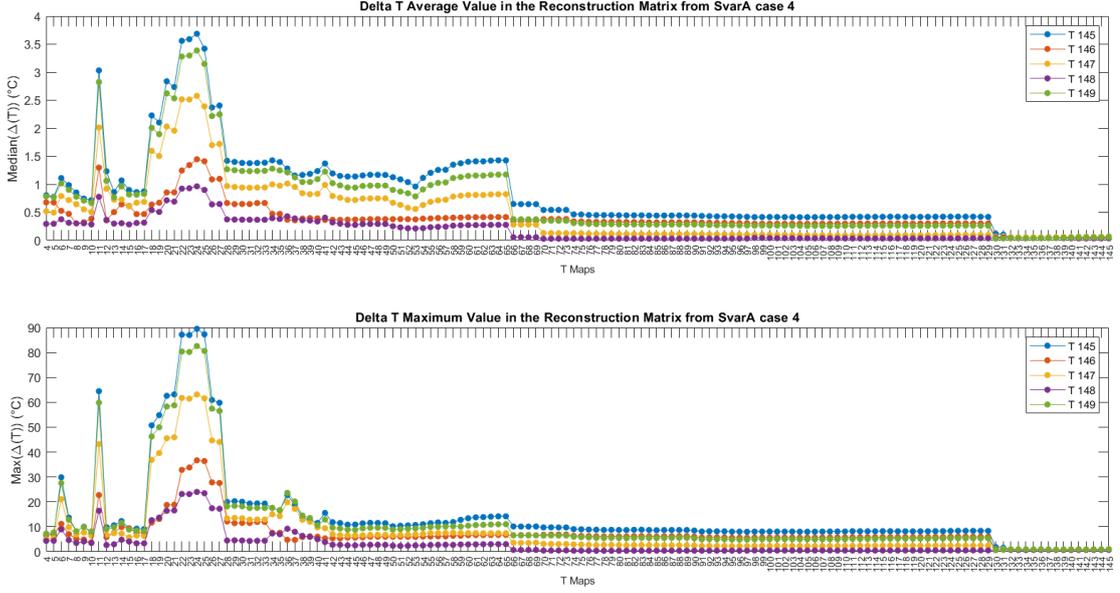
Another consideration can involve the S_{var} size and the minimum number of maps involved in the reconstruction. Setting at 0.5 °C the threshold for acceptable maximum ΔT errors in the whole region of interest, it is possible to estimate the minimum number of temperature maps needed to generate appropriate quality temperature map reconstruction. Taking into consideration MGrid Type A, which is the one that have the largest fluctuations of values, it is evident that a higher number of parameters and tissues involved makes impossible to reach small ΔT values with few temperature maps in the reconstruction matrix. To corroborate this argument, it is possible to compare 3 different cases where the dimension of the problem is gradually bigger (see Figures 4.10a, 4.15a and 4.17a). The reported plots are assembled in the Figure 4.21 reported below. It is evident that the critical drop of the ΔT max errors arise sequentially forward as the dimension is extended. In conclusion, as expected, as the size of the domain increases, the number of minimum maps needed to produce a sufficiently good reconstruction reconstruction also increases.



(a) MGrid Type A + Baseline varying k and ω of *muscle* and *tumor*



(b) MGrid Type A + Baseline varying k and ω of *muscle*, *tumor* and *fat + SAT*



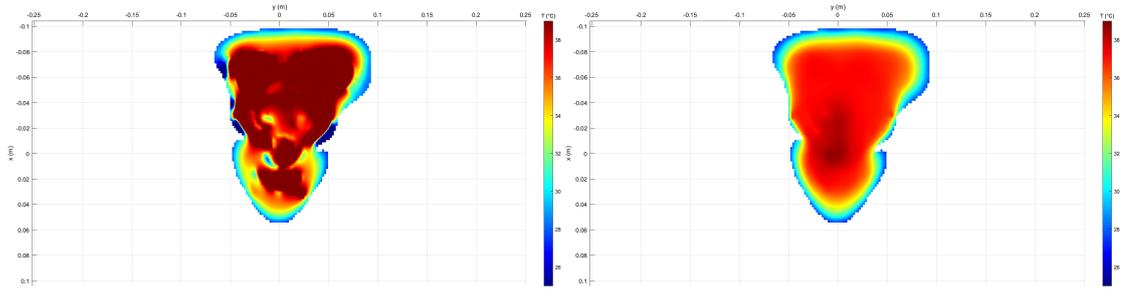
(c) MGrid Type A + Baseline varying k and ω of *muscle*, *tumor*, *fat* + *SAT* and *skin*

Figure 4.21: $N_{p,min}$ Statistics Comparison MGrid Type A + Baseline with Different Dimensions of the Parameters Space.

MGrid Type B represents the proper type of multigrid to choose between the two. From the analyses involving 25 combinations sets, i.e., when 2 tissue parameters are changed, it is possible to assess that the number of minimum maps necessary to obtain a reconstruction map of good quality reconstruction map is around 9 maps involved.

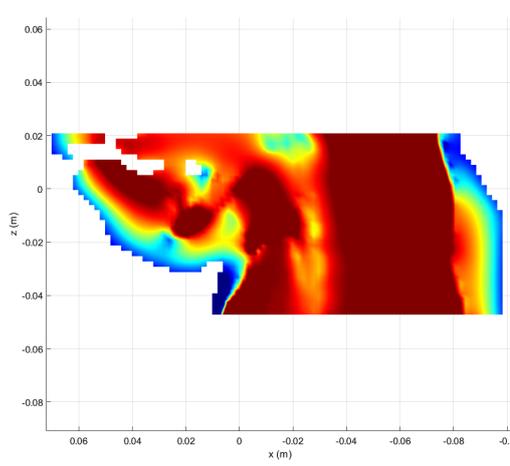
Finally, two different reconstruction maps are reported from the case where the thermal parameters were changed for all the four tissues. Figure 4.22, shows the reconstructed temperature map with the highest ΔT max error, i.e., $\Delta T=89.51$ °C (see Figure 4.17a), next to the target map that the algorithm tried to recreate. With the same settings, i.e., same number of minimum maps involved $N_{p,min}=24$ and same target map, the reconstruction process applied using MGrid Type B (see Figure 4.18) provides $\Delta T=0.09$ °C as a max error and for this reason there is no perceptible difference between the reconstructed map and the target map (see Figure 4.23), reconfirming which of the two enhances the quality of outcomes. This borderline case demonstrates how important it is to set properly the algorithm to prevent completely wrong reconstructed maps.

Instead, Figure 4.24, reports the first reconstructed temperature map, using the MGrid Type B when the thermal parameters were varied for the 4 tissues, that has the maximum ΔT values under the threshold, i.e., $\Delta T=0.48$ °C. To achieved this result, the algorithm needs at least $N_{p,min}=16$ (see 4.18) and, as the maps reported in Figure 4.24 show, maximum error values below the threshold allow a high-quality reconstruction.

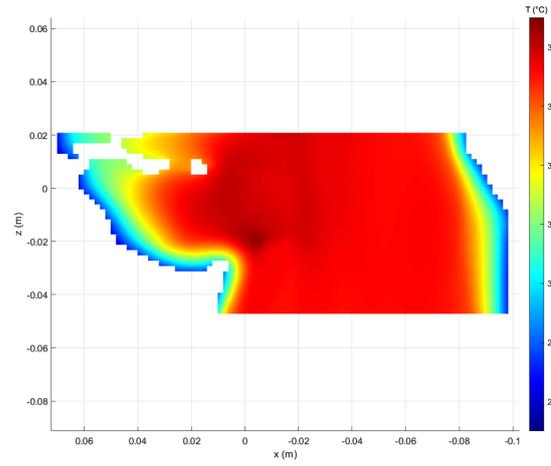


(a) Reconstructed Map, plane xy

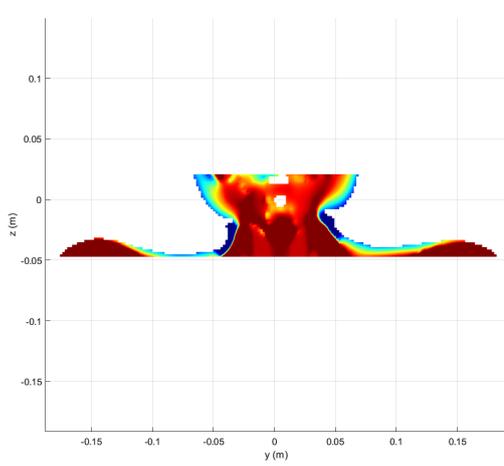
(b) Target Map, plane xy



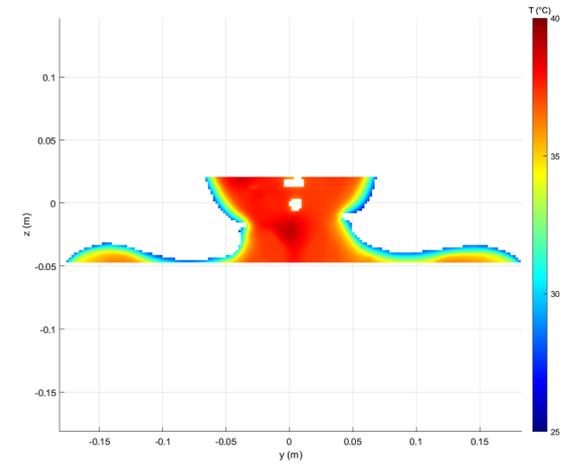
(c) Reconstructed Map, plane xz



(d) Target Map, plane xz

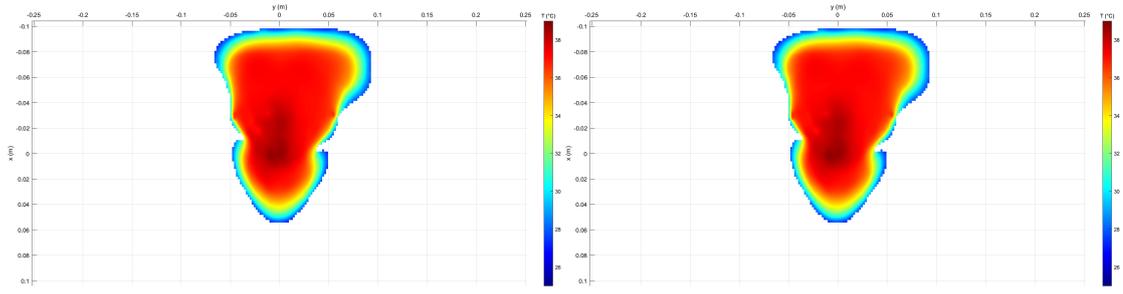


(e) Reconstructed Map, plane yz



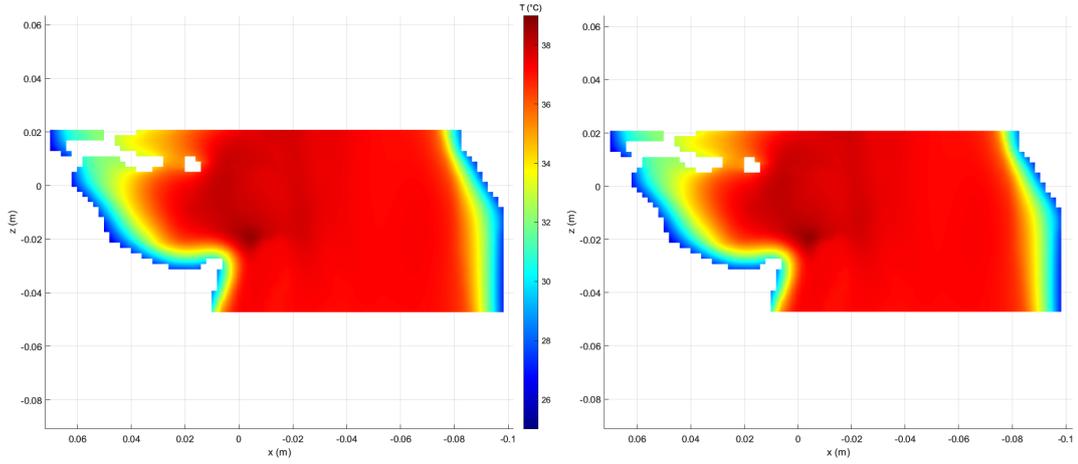
(f) Target Map, plane yz

Figure 4.22: Comparison Between The Reconstructed Map with $\Delta T_{max}=89.51$ °C, Obtained Using the MGrid Type A and $N_{p,min}=24$, and The Corresponding Target Map



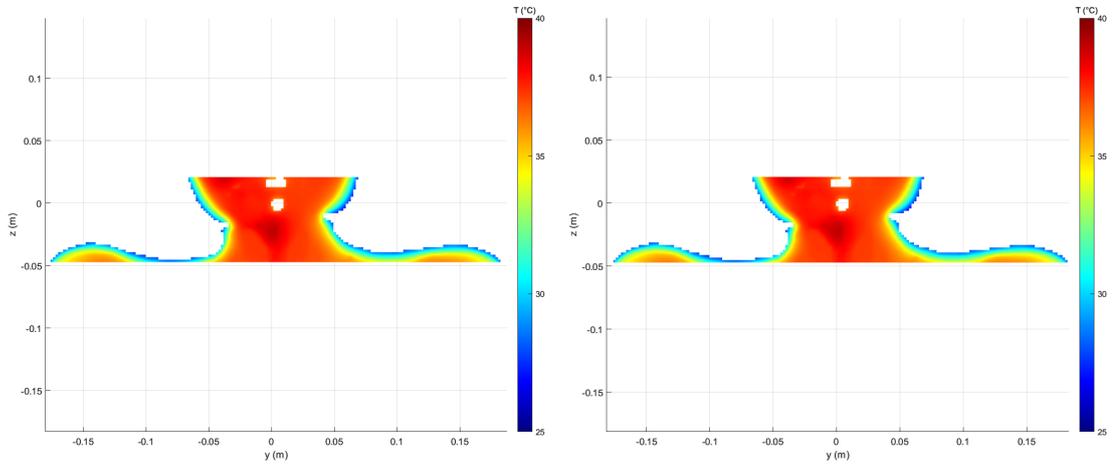
(a) Reconstructed Map, plane xy

(b) Target Map, plane xy



(c) Reconstructed Map, plane xz

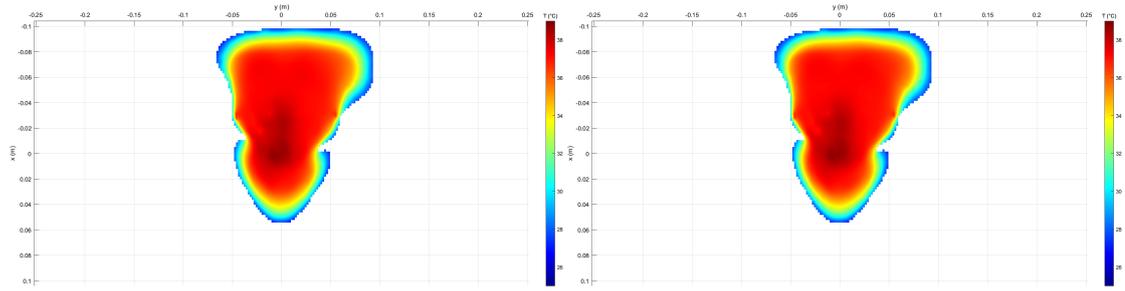
(d) Target Map, plane xz



(e) Reconstructed Map, plane yz

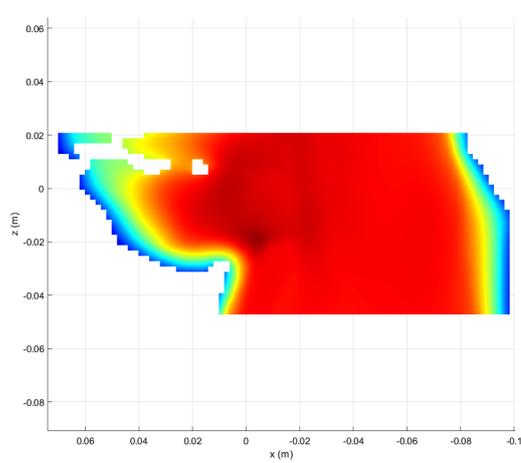
(f) Target Map, plane yz

Figure 4.23: Comparison Between The Reconstructed Map with $\Delta T_{max}=0.09$ °C, Obtained Using the MGrid Type B and $N_{p,min}=24$, and The Corresponding Target Map

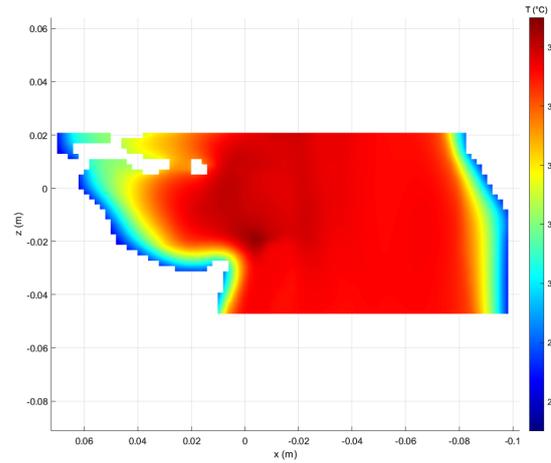


(a) Reconstructed Map, plane xy

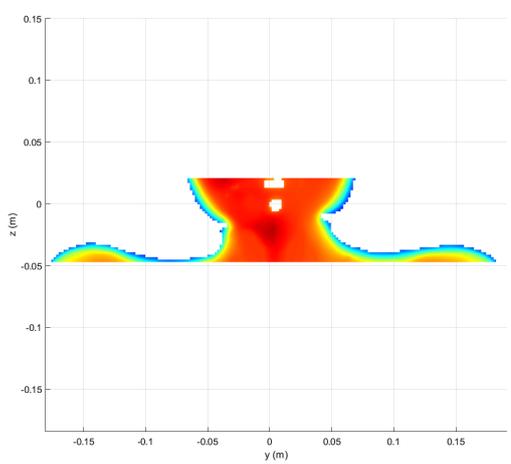
(b) Target Map, plane xy



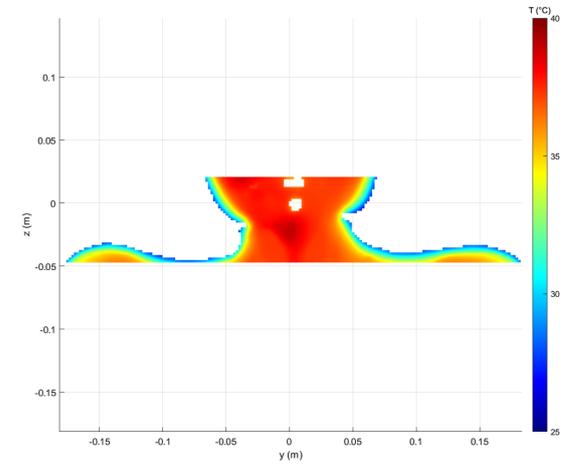
(c) Reconstructed Map, plane xz



(d) Target Map, plane xz



(e) Reconstructed Map, plane yz



(f) Target Map, plane yz

Figure 4.24: Comparison Between The Reconstructed Map with $\Delta T_{max}=0.48$ °C, Obtained Using the MGrid Type B and $N_{p,min}=16$, and The Corresponding Target Map

Conclusions

In this thesis was validate the algorithm, that allows to retrieve the patient temperature map of the whole region of interest from few known thermal points which are supposed to be known, and correspond to the temperature measurement points obtained in the clinical practice using invasive catheters. High realistic human phantom was taken from the ViP population, and an ellipsoidal tumor was inserted in its H&N region, obtaining a reliable patient model on which simulate the HTP. Then, after the proper optimization of the electromagnetic applicator, several combinations of parameter values were changed inside the phantom creating different temperature maps. Finally, the reconstruction algorithm, with different settings, was applied to recreate the entire map, assuming to know few temperature values. The latter outcomes were then compared trying to figure out which settings allowed a better reconstruction and how it can be further improved.

The electromagnetic simulation, so interaction between the microwaves and the phantom tissues, was simulated via the EM solver in Sim4Life and then exported to optimize the feedings of the antennas by maximizing the SAR in the tumor region. After the optimization and the normalization of the antennas, the bio-heat equation was solved through the Sim4Life thermal solver providing temperature maps. Different multigrids were employed to create distinct combinations of the parameters inside the human phantom and generate the related temperature maps. The modification affected two dielectric parameters, *electric conductivity* σ and *relative permittivity* ϵ_r , and two thermal parameters, *perfusion* ω and *thermal conductivity* k , for the most relevant tissue in the H&N region: *muscle*, *tumor*, *fat + SAT* and *skin*. The thermal parameters were changed for *tumor* and *muscle*, then the *fat* was added and finally these parameters were varied simultaneously for all the four tissues; on the other hand the dielectric parameters were altered one at the time together with the perfusion for only *muscle* and *tumor*.

The generation of the multigrids allow the construction of different bases of temperature maps corresponding to different combinations of the tissue parameters, which are varied within ranges fixed according to the Literature. These bases form the reconstruction matrix, which allows to retrieve a more reliable temperature map in the whole region of interest, starting from few known temperature values

of different target fields which are assumed to be unknown (except for the points along the catheter). The main goal of this thesis was to analyze the best way to generate these bases (using different multigrids), when different parameters and tissues are considered. Since, in principle, the best reconstruction should be achieved by varying all the parameters of all the tissues involved, another important point was to find the minimum number of elements of the reconstruction matrix (i.e., the minimum number of columns) necessary to achieve a sufficiently good reconstruction. This point is important to contain the computational cost necessary to produce the basis of temperature maps used to perform the reconstruction.

The implemented analyses, performed by considering a gradual number of tissues and parameters, show how a good way to create a reconstruction matrix is by using a grid where the points are obtained by randomly extracting the parameters' combinations according to a Sobol sequence. This choice limits the number of elements that is necessary to consider to achieve a good reconstruction. It should be noted that this number also depends on the considered catheter configuration used for the extraction of the known points and should be determined according to the specific clinical scenario.

Further analysis will involve the variation of all the thermal and dielectric parameters of all the tissues involved in the region of interest. This analysis can be performed in Sim4Life using the Python scripts developed during this thesis. Moreover, adaptive algorithms aimed at gradually populating the basis matrix to minimize the achieved errors by choosing the most proper points in the space of the parameters' combinations can be implemented.

MATLAB Scripts

Multigrad Generation Script

```
% Multi-dimensional pseudo-random Sobol grid.
% The parameters are organized in an external Excel file.

clear all
clc

filename = 'Parameters.xlsx';
filepath = [pwd,'\',filename];

param= [3,4]; % Type of parameters: [epsr,sigma,k,w]
tissue= [1,4]; % Type of Tissues: [Muscle,Fat,Skin,Tumor]

S = 10; % [-] number of Sobol quasi-random points
MGrid = MultiGrid_Sobol_A_baseline(filepath,param,tissue,S);

save([pwd,'\MGrid_A_baseline.txt'],'MGrid','-ascii','-tabs');
```

MGrid Type A + Baseline Function

```
function MGrid = MultiGrid_Sobol_A_baseline(filepath,param,tissue,S)

%% Grid of fixed bounds points (red points) %%

table = cell(1,numel(param));

for i = 1:numel(param)
    table{i} = xlsread(filepath,param(i),'A1:D5');
    table{i} = table{i}(tissue,:);
end

ranges = cell(numel(tissue),numel(param));
avg = cell(numel(tissue),numel(param));

for i = 1:numel(tissue)
    for j = 1:numel(param)
        ranges{i,j} = [table{j}(i,2),table{j}(i,3)];
        avg{i,j}= table{j}(i,1);
    end
end

output = cell(1,numel(param)*numel(tissue));
[output{:}] = ndgrid(ranges{:});

avg_output = cell(1,numel(param)*numel(tissue));
```

```

[avg_output{:}] = ndgrid(avg{:});

results = [];
avg_res = [];

for i = 1:length(output)
    results = [results output{1,i}(:)];
    avg_res = [avg_res avg_output{1,i}(:)];
end

%%% Quasi random Sobol sequence (green points)

q = grandstream('sobol',numel(param)*numel(tissue),'Skip',1);
X = rand(q,S,numel(param)*numel(tissue));

RF = reshape([ranges{:}],2,numel(param)*numel(tissue)).';

for i = 1:length(RF)
    X(:,i) = RF(i,1)+range(RF(i,:)).*X(:,i);
end

%%% Final grid %%%

MGrid = [avg_res; results; X];
end

```

MGrid Type B + Baseline Function

```

function MGrid = MultiGrid_Sobol_B_baseline(filepath,param,tissue,S)

table = cell(1,numel(param));

for i = 1:numel(param)
    table{i} = xlsread(filepath,param(i),'A1:D5');
    table{i} = table{i}(tissue,:);
end

ranges = cell(numel(tissue),numel(param));
avg = cell(numel(tissue),numel(param));

for i = 1:numel(tissue)
    for j = 1:numel(param)
        ranges{i,j} = [table{j}(i,2),table{j}(i,3)];
        avg{i,j}= table{j}(i,1);
    end
end

output = cell(1,numel(param)*numel(tissue));
[output{:}] = ndgrid(ranges{:});

avg_output = cell(1,numel(param)*numel(tissue));
[avg_output{:}] = ndgrid(avg{:});

avg_res = [];

for i = 1:length(output)
    avg_res = [avg_res avg_output{1,i}(:)];
end

```

```

%% Quasi random Sobol sequence (green points)

q = grandstream('sobol', numel(param)*numel(tissue), 'Skip', 1);
X = rand(q, S, numel(param)*numel(tissue));

RF = reshape([ranges{:}], 2, numel(param)*numel(tissue)).';

for i = 1:length(RF)
    X(:, i) = RF(i, 1) + range(RF(i, :)).*X(:, i);
end

MGrid = [avg_res; X];
end

```

4 Statistical Analysis

```

filepath = 'C:\Users\Diego\Desktop\Tesi\Script Matlab\Svar Matrix\T Map\';
d = dir([filepath, '*.*mat']);

Np = 27; % [-] number of target fields considered
Np_min = 10; % [-] minimum number of columns for the Svar matrix

sz1 = 50; % [-] lower cut (slice) along z for the neck region
sz2 = 111; % [-] upper cut (slice) along z for the neck region
szc = 88; % [-] slice along z passing through the z-coordinate
of the tumor's centroid

type = 'y'; % [-] type of catheter considered (along 'x' or 'y')
coord = -0.00333; % [m] coordinate xc or yc of the catheter (e.g.,
yc_tum = -0.00333 m, xc_tum = -0.01236 m)
Nc = 20; % [-] number of points along the catheter
dir_c = 'rev'; % [-] catheter's direction (options: 'fw', 'rev')
step = 1; % [-] spacing among points along the catheter Nt = Np
-Np_min;
Nt = Np - Np_min; % [-] number of target fields considered (i.e.,
columns of the Svar_cath matrix)

Svar_matrix(filepath, Np, sz1, sz2);

Svar_cath_matrix(filepath, Np, step, Nc, szc, type, dir_c, coord)

load('Svar_new.mat');
load('Svar_cath_new.mat');
delta_T = zeros(size(Svar, 1), Nt);
stat = zeros(Nt, 3);

for k = 1:Nt

    T_target_cath = Svar_cath(:, Np_min+k); % andrà da 5 a 27
    -> 23 colonne target
    beta = pinv(Svar_cath(:, 1:(Np_min+k-1)), 1e-3)*T_target_cath;
    T_rec = Svar(:, 1:(Np_min+k-1))*beta;

    delta_T(:, k) = abs(T_rec - Svar(:, Np_min+k));
    stat(k, 1) = median(delta_T(:, k));
    stat(k, 2) = max(delta_T(:, k));
    stat(k, 3) = min(delta_T(:, k));

end

```

```

figure()
boxplot(delta_T)
set(gcf,'color','w')
xlabel('t')
ylabel('\Delta(T) ( C )')
title(['Catheter along ',type,' (' ,dir_c,'), Ncath = ',num2str(round(Nc/step))])

target_ind = 5;
delta_T_base = cat(2,abs(Svar(:,1)-Svar(:,Np_min+target_ind)),delta_T(:,target_ind
));

figure()
boxplot(delta_T_base,'Labels',{'baseline','quasi-random'})
set(gcf,'color','w')
ylabel('\Delta(T) ( C )')
title(['Catheter along ',type,' (' ,dir_c,'), Ncath = ',num2str(round(Nc/step))])

figure()
subplot(2,1,1)
plot(1:Nt,stat(:,1),'.-','MarkerSize',18)
set(gcf,'color','w')
xlabel('t')
ylabel('median(\Delta(T)) ( C )')
title(['Catheter along ',type,' (' ,dir_c,'), Ncath = ',num2str(round(Nc/step))])
subplot(2,1,2)
plot(1:Nt,stat(:,2),'.-','MarkerSize',18)
set(gcf,'color','w')
xlabel('t')
ylabel('max(\Delta(T)) ( C )')

```

S_{var} Function

```

function Svar_matrix(filepath,Np,sz1,sz2)

Svar = size(4e6,Np);

for i = 1:Np

    T = load([filepath,'T',num2str(i-1),'.mat']);
    Tmap = reshape(T.Snapshot0,[length(T.Axis0)-1,length(T.Axis1)-1,length(T.Axis2
)-1,1]);
    Tmap = Tmap(:,:,sz1:sz2);
    Tmap = reshape(Tmap,[numel(Tmap),1]);
    Tmap(isnan(Tmap)) = [];
    Svar(1:numel(Tmap),i) = Tmap;

end

Svar = Svar(1:numel(Tmap),:);
save('Svar_new.mat','Svar');
end

```

S_{cath} Function

```

function Svar_cath_matrix(filepath,Np,step,Nc,szc,type,dir_c,coord)

```

```

T = load([filepath, 'T0.mat']);

T_map = reshape(T.Snapshot0, [length(T.Axis0)-1, length(T.Axis1)-1, length(T.Axis2)
    -1, 1]);

x = (T.Axis0(1:(length(T.Axis0)-1))+T.Axis0(2:length(T.Axis0)))./2;
y = (T.Axis1(1:(length(T.Axis1)-1))+T.Axis1(2:length(T.Axis1)))./2;
z = (T.Axis2(1:(length(T.Axis2)-1))+T.Axis2(2:length(T.Axis2)))./2;

[X,Y,-] = ndgrid(x,y,z);

c1 = squeeze(X(:,:,szc));
c2 = squeeze(Y(:,:,szc));
T_cut = squeeze(T_map(:,:,szc));

if strcmp(type, 'y')
    ind = find(abs(c1(:,1) .* -coord) < 1e-3);
    ind_c = ~isnan(T_cut(ind,:));

    c1_cath = c1(ind, ind_c);
    c2_cath = c2(ind, ind_c);
else
    ind = find(abs(c2(1,:) .* -coord) < 1e-3);
    ind_c = ~isnan(T_cut(:, ind));

    c1_cath = c1(ind_c, ind);
    c2_cath = c2(ind_c, ind);
end

figure()
s = surf(c1, c2, T_cut);
set(s, 'edgecolor', 'none');
h = colorbar;
title(h, 'T ( C )')
shading interp
colormap jet
axis([min(x) max(x) min(y) max(y)])
set(gcf, 'color', 'w')
xlabel('x (m)')
ylabel('y (m)')
grid off
view(90, 90)
title('Catheter on the baseline map T0');
hold on
if strcmp(dir_c, 'fw')
    plot3(c1_cath(1:step:Nc), c2_cath(1:step:Nc), 1e3.*ones(Nc,1), 'k.', 'MarkerSize',
        8)
    ind_c = find(ind_c==1);
    ind_c = ind_c(1:step:Nc);
else
    plot3(c1_cath(end-Nc+1:step:end), c2_cath(end-Nc+1:step:end), 1e3.*ones(numel(
        c1_cath(end-Nc+1:step:end)),1), 'k.', 'MarkerSize', 8)
    ind_c = find(ind_c==1);
    ind_c = ind_c(end-Nc+1:step:end);
end

Svar_cath = size(4e6, Np);

for i = 1:Np

```

```

T = load([filepath, 'T', num2str(i-1), '.mat']);
T_map = reshape(T.Snapshot0, [length(T.Axis0)-1, length(T.Axis1)-1, length(T.
Axis2)-1, 1]);
T_cut = squeeze(T_map(:, :, szc));

if strcmp(type, 'y')
    T_cut = T_cut(ind, ind_c);
else
    T_cut = T_cut(ind_c, ind);
end

Svar_cath(1: numel(T_cut), i) = T_cut;

end

Svar_cath = Svar_cath(1: numel(T_cut), :);
save('Svar_cath_new.mat', 'Svar_cath');
end

```

$N_{p,min}$ Statistic Comparison

```

grid_type = 'A';

filepath = [pwd, '\Mappe_MGrid_', grid_type, '\'];

filepath1 = [pwd, '\Mappe_MGrid_B\'];

d = dir([filepath, '*.*mat']);

Np = length(d(not([d.isdir]))); % [-] total number of columns in the Svar matrix
Nt = 5; % [-] number of target fields

Nric = Np - Nt - 3

sz1 = 70; % [-] lower cut (slice) along z for the neck
region
sz2 = 105; % [-] upper cut (slice) along z for the neck
region
szc = 88; % [-] slice along z passing through the z-
coordinate of the tumor's centroid

type = 'y'; % [-] type of catheter considered (along 'x' or '
y')
coord = -0.00333; % [m] coordinate xc or yc of the catheter (e.g
., xc_tum = 0.158 m, yc_tum = 0.281 m)
Nc = 20; % [-] number of points along the catheter
dir_c = 'fw'; % [-] catheter's direction (options: 'fw', 'rev')
step = 1; % [-] spacing among points along the catheter (
default value = 1)

if grid_type == 'A';
    if isfile(fullfile(cd, 'Svar_A.mat')) == 0
        Svar_matrix_ros(filepath, 'A', Np, sz1, sz2);
    end
    Svar_cath_matrix_ros(filepath, 'A', Np, step, Nc, szc, type, dir_c, coord, 'on');
end
if isfile(fullfile(cd, 'Svar_B.mat')) == 0
    Svar_matrix_ros(filepath1, 'B', Np, sz1, sz2);
end
end

```

```

% Svar_cath_matrix_ros(filepath,'A',Np,step,Nc,szc,type,dir_c,coord,'on')
Svar_cath_matrix_ros(filepath1,'B',Np,step,Nc,szc,type,dir_c,coord,'off');

Svar = importdata(['Svar_',grid_type,'_pt6.mat'],'Svar');
Svar_B = importdata('Svar_B_pt6.mat','Svar');
Svar_cath = importdata(['Svar_cath_',grid_type,'_pt6.mat'],'Svar_cath');
Svar_cath_B = importdata('Svar_cath_B_pt6.mat','Svar_cath');

stat = zeros(Nt,Np-Nt-3,3);

for k = 1:Nt
    kind = (Np-Nt)+k;

    for n = 1:(Np-Nt-3)
        Np_min = 3+n;

        T_target_cath = Svar_cath_B(:,kind);
        beta = pinv(Svar_cath(:,1:Np_min),1e-3)*T_target_cath;
        T_rec = Svar(:,1:Np_min)*beta;

        delta_T = abs(T_rec-Svar_B(:,kind));
        stat(k,n,1) = median(delta_T);
        stat(k,n,2) = max(delta_T);
        stat(k,n,3) = min(delta_T);
    end
end

array=1:Nric;

figure()
subplot(2,1,1)
plot(stat(1,:,1),'.-','MarkerSize',18);
hold on
plot(stat(2,:,1),'.-','MarkerSize',18);
hold on
plot(stat(3,:,1),'.-','MarkerSize',18);
hold on
plot(stat(4,:,1),'.-','MarkerSize',18);
hold on
plot(stat(5,:,1),'.-','MarkerSize',18);
hold off

ylim([0 4]);
xticks([array]);
xlim([0 Nric]);
title(['Delta T Average Value in the Reconstruction Matrix from Svar', grid_type,'
case 4']);
xlabel('T Maps');
ylabel('Median(\Delta(T)) ( C )');
set(gcf,'color','w')
legend('T 145','T 146','T 147','T 148','T 149')

subplot(2,1,2)
plot(stat(1,:,2),'.-','MarkerSize',18);
hold on
plot(stat(2,:,2),'.-','MarkerSize',18);
hold on
plot(stat(3,:,2),'.-','MarkerSize',18);
hold on
plot(stat(4,:,2),'.-','MarkerSize',18);

```

```
hold on
plot(stat(5, :, 2), '.-', 'MarkerSize', 18);
hold off

ylim([0 90]);
xticks([array]);
xlim([0 Nric]);
title(['Delta T Maximum Value in the Reconstruction Matrix from Svar', grid_type, '
      case 4']);
xlabel('T Maps');
ylabel('Max(\Delta(T)) ( C )');
set(gcf, 'color', 'w');
legend('T 145', 'T 146', 'T 147', 'T 148', 'T 149')
```

Python Scripts

Electric Fields Export

```
# Antenna electric fields extrapolation for the SAR-based focusing
import numpy
import s4l_v1.analysis as analysis
import s4l_v1.document as document
import s4l_v1.model as model
import s4l_v1.units as units
from s4l_v1 import ReleaseVersion
from s4l_v1 import Unit
import itertools

def all_entities_within_group(entity_group):
    if isinstance(entity_group, model.EntityGroup):
        return list(itertools.chain.from_iterable(
            all_entities_within_group(e) for e in entity_group.Entities))
    else:
        return [entity_group]

vip_group = model.AllEntities()['Yoon-sun']
entities_H = all_entities_within_group(vip_group)
entities_T = model.AllEntities()['Tumor']

# Name of the directory where the files will be saved
dir = "C:\\Users\\Diego\\Desktop\\S4L_Fields\\"

# HEALTHY REGION standalone electric fields Ei_H
try:
    # Define the version to use for default values
    ReleaseVersion.set_active(ReleaseVersion.version7_0)

    # Creating the analysis pipeline
    # Adding a new EmMultiPortSimulationExtractor
    simulation = document.AllSimulations["EM 1"]
    em_multi_port_simulation_extractor = simulation.Results()
    #Create the postprocessing pipeline once
    output_port = em_multi_port_simulation_extractor.Outputs[0]
    # Adding a new EmPortSimulationExtractor
    em_port_simulation_extractor = analysis.extractors.EmPortSimulationExtractor(
        inputs=[output_port])
    em_port_simulation_extractor.UpdateAttributes()
    document.AllAlgorithms.Add(em_port_simulation_extractor)
    # Adding a new EmSensorExtractor (H)
    em_sensor_extractor = em_port_simulation_extractor["Overall Field"]
    em_sensor_extractor.FrequencySettings.ExtractedFrequency = u"All"
    em_sensor_extractor.SurfaceCurrent.SurfaceResolution = 0.001, units.Meters
    document.AllAlgorithms.Add(em_sensor_extractor)
    # Adding a new FieldMaskingFilter (H)
    inputs = [em_sensor_extractor.Outputs["EM E(x,y,z,f0)"]]
```

```

field_masking_filter = analysis.core.FieldMaskingFilter(inputs=inputs)
field_masking_filter.SetAllMaterials(False)
field_masking_filter.SetEntities(entities_H)
field_masking_filter.UpdateAttributes()
document.AllAlgorithms.Add(field_masking_filter)
# Adding a new MatlabExporter (H)
inputs = [field_masking_filter.Outputs["EM E(x,y,z,f0)"]]
matlab_exporter = analysis.exporters.MatlabExporter(inputs=inputs)
matlab_exporter.UpdateAttributes()
document.AllAlgorithms.Add(matlab_exporter)

for i, output_port in enumerate(em_multi_port_simulation_extractor.Outputs):
    em_port_simulation_extractor.raw.SetInputConnection(0, output_port.raw) #
    this is the main "trick"
    em_port_simulation_extractor.UpdateAttributes()

    inputs = [field_masking_filter.Outputs["EM E(x,y,z,f0)"]]
    matlab_exporter.FileName = (dir + "E{}_H.mat".format(i+1))

    print(matlab_exporter.FileName)
    matlab_exporter.UpdateAttributes()
    matlab_exporter.Update()

except Exception as exc:
    import traceback
    traceback.print_exc(exc)
    # Reset active version to default
    ReleaseVersion.reset()
    raise(exc)

# TUMOR REGION standalone electric fields Ei_T
try:
    # Define the version to use for default values
    ReleaseVersion.set_active(ReleaseVersion.version6_2)

    # Creating the analysis pipeline
    # Adding a new EmMultiPortSimulationExtractor
    simulation = document.AllSimulations["EM 1"]
    em_multi_port_simulation_extractor = simulation.Results()

    #Create the postprocessing pipeline once
    output_port = em_multi_port_simulation_extractor.Outputs[0]
    # Adding a new EmPortSimulationExtractor
    em_port_simulation_extractor = analysis.extractors.EmPortSimulationExtractor(
        inputs=[output_port])
    em_port_simulation_extractor.UpdateAttributes()
    document.AllAlgorithms.Add(em_port_simulation_extractor)
    # Adding a new EmSensorExtractor (T)
    em_sensor_extractor = em_port_simulation_extractor["Overall Field"]
    em_sensor_extractor.FrequencySettings.ExtractedFrequency = u"All"
    em_sensor_extractor.SurfaceCurrent.SurfaceResolution = 0.001, units.Meters
    document.AllAlgorithms.Add(em_sensor_extractor)
    # Adding a new FieldMaskingFilter (T)
    inputs = [em_sensor_extractor.Outputs["EM E(x,y,z,f0)"]]
    field_masking_filter = analysis.core.FieldMaskingFilter(inputs=inputs)
    field_masking_filter.SetAllMaterials(False)
    field_masking_filter.SetEntities([entities_T])
    field_masking_filter.UpdateAttributes()
    document.AllAlgorithms.Add(field_masking_filter)
    # Adding a new MatlabExporter (T)
    inputs = [field_masking_filter.Outputs["EM E(x,y,z,f0)"]]

```

```

matlab_exporter = analysis.exporters.MatlabExporter(inputs=inputs)
matlab_exporter.UpdateAttributes()
document.AllAlgorithms.Add(matlab_exporter)

for i, output_port in enumerate(em_multi_port_simulation_extractor.Outputs):
    em_port_simulation_extractor.raw.SetInputConnection(0, output_port.raw) #
    this is the main "trick"
    em_port_simulation_extractor.UpdateAttributes()

    inputs = [field_masking_filter.Outputs["EM E(x,y,z,f0)"]]
    matlab_exporter.FileName = (dir + "E{}_T.mat".format(i+1))

    print(matlab_exporter.FileName)
    matlab_exporter.UpdateAttributes()
    matlab_exporter.Update()

except Exception as exc:
    import traceback
    traceback.print_exc(exc)
    # Reset active version to default
    ReleaseVersion.reset()
    raise(exc)

# HEALTHY and TUMOR REGION SAR and J fields for the antenna n
try:
    # Define the version to use for default values
    ReleaseVersion.set_active(ReleaseVersion.version6_2)

    # Creating the analysis pipeline
    # Adding a new EmMultiPortSimulationExtractor
    simulation = document.AllSimulations["EM 1"] # check the name of the
    Simulation node (EM, EM 1, etc)
    em_multi_port_simulation_extractor = simulation.Results()

    #Create the postprocessing pipeline once
    output_port = em_multi_port_simulation_extractor.Outputs[0] # use [n-1] for
    Source n

    # Adding a new EmPortSimulationExtractor
    em_port_simulation_extractor = analysis.extractors.EmPortSimulationExtractor(
    inputs=[output_port])
    em_port_simulation_extractor.UpdateAttributes()
    document.AllAlgorithms.Add(em_port_simulation_extractor)

    # Adding a new EmSensorExtractor
    em_sensor_extractor = em_port_simulation_extractor["Overall Field"]
    em_sensor_extractor.FrequencySettings.ExtractedFrequency = u"All"
    em_sensor_extractor.SurfaceCurrent.SurfaceResolution = 0.001, units.Meters
    document.AllAlgorithms.Add(em_sensor_extractor)

    # Adding a new FieldMaskingFilter for SAR (H)
    inputs = [em_sensor_extractor.Outputs["SAR(x,y,z,f0)"]]
    field_masking_filter = analysis.core.FieldMaskingFilter(inputs=inputs)
    field_masking_filter.SetAllMaterials(False)
    field_masking_filter.SetEntities(entities_H)
    field_masking_filter.UpdateAttributes()
    document.AllAlgorithms.Add(field_masking_filter)

    # Adding a new FieldMaskingFilter for SAR (T)
    inputs = [em_sensor_extractor.Outputs["SAR(x,y,z,f0)"]]
    field_masking_filter_2 = analysis.core.FieldMaskingFilter(inputs=inputs)

```

```

field_masking_filter_2.SetAllMaterials(False)
field_masking_filter_2.SetEntities([entities_T])
field_masking_filter_2.UpdateAttributes()
document.AllAlgorithms.Add(field_masking_filter_2)

# Adding a new FieldMaskingFilter for J (H)
inputs = [em_sensor_extractor.Outputs["J(x,y,z,f0)"]]
field_masking_filter_3 = analysis.core.FieldMaskingFilter(inputs=inputs)
field_masking_filter_3.SetAllMaterials(False)
field_masking_filter_3.SetEntities(entities_H)
field_masking_filter_3.UpdateAttributes()
document.AllAlgorithms.Add(field_masking_filter_3)

# Adding a new FieldMaskingFilter for J (T)
inputs = [em_sensor_extractor.Outputs["J(x,y,z,f0)"]]
field_masking_filter_4 = analysis.core.FieldMaskingFilter(inputs=inputs)
field_masking_filter_4.SetAllMaterials(False)
field_masking_filter_4.SetEntities([entities_T])
field_masking_filter_4.UpdateAttributes()
document.AllAlgorithms.Add(field_masking_filter_4)

# Adding a new MatlabExporter for SAR (H)
inputs = [field_masking_filter.Outputs["SAR(x,y,z,f0)"]]
matlab_exporter = analysis.exporters.MatlabExporter(inputs=inputs)
matlab_exporter.FileName = (dir + "SAR1_H.mat")
matlab_exporter.UpdateAttributes()
document.AllAlgorithms.Add(matlab_exporter)
matlab_exporter.Update()

# Adding a new MatlabExporter for SAR (T)
inputs = [field_masking_filter_2.Outputs["SAR(x,y,z,f0)"]]
matlab_exporter_2 = analysis.exporters.MatlabExporter(inputs=inputs)
matlab_exporter_2.FileName = (dir + "SAR1_T.mat")
matlab_exporter_2.UpdateAttributes()
document.AllAlgorithms.Add(matlab_exporter_2)
matlab_exporter_2.Update()

# Adding a new MatlabExporter for J (H)
inputs = [field_masking_filter_3.Outputs["J(x,y,z,f0)"]]
matlab_exporter_3 = analysis.exporters.MatlabExporter(inputs=inputs)
matlab_exporter_3.FileName = (dir + "J1_H.mat")
matlab_exporter_3.UpdateAttributes()
document.AllAlgorithms.Add(matlab_exporter_3)
matlab_exporter_3.Update()

# Adding a new MatlabExporter for J (T)
inputs = [field_masking_filter_4.Outputs["J(x,y,z,f0)"]]
matlab_exporter_4 = analysis.exporters.MatlabExporter(inputs=inputs)
matlab_exporter_4.FileName = (dir + "J1_T.mat")
matlab_exporter_4.UpdateAttributes()
document.AllAlgorithms.Add(matlab_exporter_4)
matlab_exporter_4.Update()

except Exception as exc:
    import traceback
    traceback.print_exc(exc)
    # Reset active version to default
    ReleaseVersion.reset()
    raise(exc)

```

Thermal Parameter Variation and Map Extraction

```

import numpy
import math
import s4l_v1.document as document
import s4l_v1.materials.database as database
import s4l_v1.model as model
import s4l_v1.simulation.thermal as thermal
import s4l_v1.analysis as analysis
import s4l_v1.units as units
from s4l_v1 import ReleaseVersion
from s4l_v1 import Unit
import itertools

def all_entities_within_group(entity_group):
    '''return a list of all model entities within a given group, including all
    subdirectories'''
    if isinstance(entity_group, model.EntityGroup):
        return list(itertools.chain.from_iterable(
            all_entities_within_group(e) for e in entity_group.Entities))
    else:
        return [entity_group]

vip_group = model.AllEntities()['Yoon-sun']
entities__all = all_entities_within_group(vip_group)
entities__air_internal = model.AllEntities()['Air_internal']
entities__tumor = model.AllEntities()['Tumor']
entities__all.remove(entities__air_internal)
entities__all.append(entities__tumor)

dir = u"C:\\Users\\Diego\\Desktop\\Tesi\\6_Adding Skin Tissue\\Mappe_MGrid_A\\"

coord = numpy.loadtxt("C:/Users/Diego/Desktop/Tesi/6_Adding Skin Tissue/
    MGrid_A_base_pt_6.txt")

print (coord)
num_rows, num_columns = coord.shape

for i in range (0,1):
    km = coord[i,0]
    kf = coord[i,1]
    ks = coord[i,2]
    kt = coord[i,3]
    wm = coord[i,4]
    wf = coord[i,5]
    ws = coord[i,6]
    wt = coord[i,7]

    print(km, kf, ks, kt, wm, wf, ws, wt)

    # Define the version to use for default values
    ReleaseVersion.set_active(ReleaseVersion.version7_0)
    #simulation = document.AllSimulations["60 W T#"]
    simulation = document.AllSimulations["Th 60 W - Dielectric"]
    simulation.ClearResults()
    simulation.ResetVoxels()

    entity__muscle = model.AllEntities()["Muscle"]
    muscle_new_k_value = km, Unit("W/m/K")
    muscle_new_w_value = wm, Unit("ml/min/kg")

```

```

entity__fat = model.AllEntities()["Fat"]
fat_new_k_value = kf, Unit("W/m/K")
fat_new_w_value = wf, Unit("ml/min/kg")

entity__fat = model.AllEntities()["SAT"]
SAT_new_k_value = kf, Unit("W/m/K")
SAT_new_w_value = wf, Unit("ml/min/kg")

entity__fat = model.AllEntities()["Skin"]
skin_new_k_value = ks, Unit("W/m/K")
skin_new_w_value = ws, Unit("ml/min/kg")

entity__tumor = model.AllEntities()["Tumor"]
tumor_new_k_value = kt, Unit("W/m/K")
tumor_new_w_value = wt, Unit("ml/min/kg")

# Change the Perfusion value of an existing material
sets = simulation.AllSettings
for idx, set in enumerate(sets):
    if set.Name == 'Muscle':
        set.ThermalConductivity = muscle_new_k_value
        set.HeatTransferRate.UsePerfusionUnits = True
        set.HeatTransferRate.PuConstantTerm = muscle_new_w_value
    if set.Name == 'Fat':
        set.ThermalConductivity = fat_new_k_value
        set.HeatTransferRate.UsePerfusionUnits = True
        set.HeatTransferRate.PuConstantTerm = fat_new_w_value
    if set.Name == 'SAT':
        set.ThermalConductivity = SAT_new_k_value
        set.HeatTransferRate.UsePerfusionUnits = True
        set.HeatTransferRate.PuConstantTerm = SAT_new_w_value
    if set.Name == 'Skin':
        set.ThermalConductivity = skin_new_k_value
        set.HeatTransferRate.UsePerfusionUnits = True
        set.HeatTransferRate.PuConstantTerm = skin_new_w_value
    if set.Name == 'Tumor':
        set.ThermalConductivity = tumor_new_k_value
        set.HeatTransferRate.UsePerfusionUnits = True
        set.HeatTransferRate.PuConstantTerm = tumor_new_w_value

# Update the materials with the new frequency parameters
simulation.UpdateAllMaterials()

# RUN
simulation.UpdateGrid()
simulation.CreateVoxels()
simulation.RunSimulation(wait=True)

# Create extractor for a given simulation output file
results = simulation.Results()

# overall field sensor
overall_field_sensor = results[ 'Overall Field' ]

# EXPORT MATLAB
inputs = [overall_field_sensor.Outputs["T(x,y,z)"]]
field_masking_filter = analysis.core.FieldMaskingFilter(inputs=inputs)
field_masking_filter.SetAllMaterials(False)
field_masking_filter.SetEntities(entities__all)
field_masking_filter.UpdateAttributes()
document.AllAlgorithms.Add(field_masking_filter)

```

```

inputs = [field_masking_filter.Outputs["T(x,y,z)"]]
matlab_exporter = analysis.exporters.MatlabExporter(inputs=inputs)
matlab_exporter.Name = ("{}".format(i))
matlab_exporter.FileName = (dir + "T_PT6_A_{}.mat".format(i))
matlab_exporter.UpdateAttributes()
document.AllAlgorithms.Add(matlab_exporter)
matlab_exporter.Update(overwrite=True)
matlab_exporter.Update(overwrite=True)

```

Dielectric Parameter Variation and Map Extraction

```

import numpy
import math
import s4l_v1.document as document
import s4l_v1.materials.database as database
import s4l_v1.model as model
import s4l_v1.simulation.thermal as thermal
import s4l_v1.analysis as analysis
import s4l_v1.units as units
from s4l_v1 import ReleaseVersion
from s4l_v1 import Unit
import itertools

def all_entities_within_group(entity_group):
    '''return a list of all model entities within a given group, including all
    subdirectories'''
    if isinstance(entity_group, model.EntityGroup):
        return list(itertools.chain.from_iterable(
            all_entities_within_group(e) for e in entity_group.Entities))
    else:
        return [entity_group]

vip_group = model.AllEntities()['Yoon-sun']
entities__all = all_entities_within_group(vip_group)
entities__air_internal=model.AllEntities()['Air_internal']
entities__tumor=model.AllEntities()['Tumor']
entities__all.remove(entities__air_internal)
entities__all.append(entities__tumor)

dir = u"C:\\Users\\Diego\\Desktop\\Tesi\\7_Dielectric\\Mappe_MGrid_B\\"
#dir2 = u"C:\\Users\\Diego\\Desktop\\Tesi Vecchi\\Focusing_RealisticModel - Copia.
smash_Results\\"

coord = numpy.loadtxt("C:/Users/Diego/Desktop/Tesi/7_Dielectric/MGrid_B_base_pt_7.
txt")

print (coord)
size=coord.size
num_rows, num_columns = coord.shape

for i in range (0, num_rows):

    sm = coord[i,0]
    st = coord[i,1]
    wm = coord[i,2]
    wt = coord[i,3]

    print (sm, st, wm, wt)

```

```

#Define the version to use for default values
ReleaseVersion.set_active(ReleaseVersion.version7_0)
simulation = document.AllSimulations["EM 1 - Copy"]
simulation.ClearResults()
simulation.ResetVoxels()

entity__muscle = model.AllEntities()["Muscle"]
muscle_new_s_value = sm
muscle_new_w_value = wm, Unit("ml/min/kg")
entity__tumor = model.AllEntities()["Tumor"]
tumor_new_s_value = st
tumor_new_w_value = wt, Unit("ml/min/kg")

#Change the Mass Density value of an existing material
sets = simulation.AllSettings
for idx,set in enumerate(sets):
    if set.Name == 'Muscle':
        set.ElectricConductivity=muscle_new_s_value
        #set.ElectricProps.Conductivity=muscle_new_s_value
    if set.Name == 'Tumor':
        set.ElectricConductivity=tumor_new_s_value
        #set.ElectricProps.Conductivity=tumor_new_s_value

#Update the materials with the new frequency parameters
simulation.UpdateAllMaterials()

#RUN
simulation.UpdateGrid()
simulation.CreateVoxels()
simulation.RunSimulation(wait=True)

simulation = document.AllSimulations["EM 1 - Copy"]
simulation_extractor = simulation.Results()

#Adding a new EmSensorExtractor
em_sensor_extractor = simulation_extractor["Overall Field"]
em_sensor_extractor.FrequencySettings.ExtractedFrequency = u"All"
em_sensor_extractor.Normalization.Normalize = True
em_sensor_extractor.Normalization.AvailableReferences = u"EM Input Power(f)"
em_sensor_extractor.Normalization.NewReferenceValue = 60.0, units.Watts
em_sensor_extractor.SurfaceCurrent.SurfaceResolution = 0.001, units.Meters
document.AllAlgorithms.Add(em_sensor_extractor)

#Adding a new FieldSnapshotFilter
inputs = [em_sensor_extractor.Outputs["El. Loss Density(x,y,z,f0)"]]
field_snapshot_filter = analysis.field.FieldSnapshotFilter(inputs=inputs)
#field_snapshot_filter.Snapshots.ExistingValues = u"not matching"
#field_snapshot_filter.Snapshots.TargetValue = 0.0, units.Hz
field_snapshot_filter.UpdateAttributes()
document.AllAlgorithms.Add(field_snapshot_filter)

#Adding a new DataCacheExporter
inputs = [field_snapshot_filter.Outputs["El. Loss Density(x,y,z,f0)"]]
data_cache_exporter = analysis.exporters.DataCacheExporter(inputs=inputs)
data_cache_exporter.Name = "Data Cache Exporter - User Defined Source"
data_cache_exporter.FileName = (dir + "Source{}.cache".format(i+1))
data_cache_exporter.UpdateAttributes()
document.AllAlgorithms.Add(data_cache_exporter)
data_cache_exporter.Update(overwrite=True)
data_cache_exporter.Update(overwrite=True)

```

```

##Thermal Simulation
ReleaseVersion.set_active(ReleaseVersion.version7_0)
simulation1 = document.AllSimulations["Th 60 W - Dielectric"]
simulation1.ClearResults()
simulation1.ResetVoxels()

# Change the Mass Density value of an existing material
sets = simulation1.AllSettings
for idx, set in enumerate(sets):
    if set.Name == 'Muscle':
        set.HeatTransferRate.UsePerfusionUnits = True
        set.HeatTransferRate.PuConstantTerm = muscle_new_w_value
    if set.Name == 'Tumor':
        set.HeatTransferRate.UsePerfusionUnits = True
        set.HeatTransferRate.PuConstantTerm = tumor_new_w_value

simulation1.UpdateAllMaterials()
# Adding a new StationaryUserDefinedHeatSource

if i > 0:
    simulation1.Remove(stationary_user_defined_heat_source, components)

stationary_user_defined_heat_source = thermal.StationaryUserDefinedHeatSource
()
components = []
stationary_user_defined_heat_source.UserDefinedFileName = (dir + "Source{}.
cache".format(i+1))
simulation1.Add(stationary_user_defined_heat_source, components)

# RUN
simulation1.UpdateGrid()
simulation1.CreateVoxels()
simulation1.RunSimulation(wait=True)

# Create extractor for a given simulation output file
results1 = simulation1.Results()

# overall field sensor
overall_field_sensor1 = results1[ 'Overall Field' ]

# EXPORT MATLAB
inputs = [overall_field_sensor1.Outputs["T(x,y,z)"]]
field_masking_filter = analysis.core.FieldMaskingFilter(inputs=inputs)
field_masking_filter.SetAllMaterials(False)
field_masking_filter.SetEntities(entities__all)
field_masking_filter.UpdateAttributes()
document.AllAlgorithms.Add(field_masking_filter)

inputs = [field_masking_filter.Outputs["T(x,y,z)"]]
matlab_exporter = analysis.exporters.MatlabExporter(inputs=inputs)
matlab_exporter.Name=("Matlab {}".format(i))
matlab_exporter.FileName = (dir + "T_PT7_B{}.mat".format(i))
matlab_exporter.UpdateAttributes()
document.AllAlgorithms.Add(matlab_exporter)
matlab_exporter.Update(overwrite=True)
matlab_exporter.Update(overwrite=True)

```

Bibliography

- [1] Nikola Cihoric et al. «Hyperthermia-related clinical trials on cancer treatment within the ClinicalTrials.gov registry». en. In: *Int. J. Hyperthermia* 31.6 (May 2015), pp. 609–614 (cit. on pp. 1, 5).
- [2] N R Datta, S Gómez Ordóñez, U S Gaipl, M M Paulides, H Crezee, J Gellermann, D Marder, E Puric, and S Bodis. «Local hyperthermia combined with radiotherapy and-/or chemotherapy: recent advances and promises for the future». en. In: *Cancer Treat. Rev.* 41.9 (Nov. 2015), pp. 742–753 (cit. on pp. 1, 5).
- [3] Niloy R Datta, Susanne Rogers, Silvia Gómez Ordóñez, Emsad Puric, and Stephan Bodis. «Hyperthermia and radiotherapy in the management of head and neck cancers: A systematic review and meta-analysis». en. In: *Int. J. Hyperthermia* 32.1 (2016), pp. 31–40 (cit. on p. 1).
- [4] M M Paulides, J F Bakker, E Neufeld, J van der Zee, P P Jansen, P C Levendag, and G C van Rhoon. «Winner of the “New Investigator Award” at the European Society of Hyperthermia Oncology Meeting 2007. The HYPERcollar: a novel applicator for hyperthermia in the head and neck». en. In: *Int. J. Hyperthermia* 23.7 (Nov. 2007), pp. 567–576 (cit. on pp. 1, 6, 7).
- [5] Laura Q M Chow. «60-72 10.1056/NEJMra1715715 eng Journal Article Review». en. In: *N. Engl. J. Med.* 382.1 (Jan. 2020), pp. 60–72 (cit. on p. 3).
- [6] Eugene Son, Aru Panwar, Charles H Mosher, and Daniel Lydiatt. «Cancers of the major salivary gland». en. In: *J. Oncol. Pract.* 14.2 (Feb. 2018), pp. 99–108 (cit. on p. 3).
- [7] National Cancer Institute. *Head and Neck Cancers was originally published by the National Cancer Institute.* 2021. URL: <https://www.cancer.gov/types/head-and-neck/head-neck-fact-sheet> (visited on 11/02/2022) (cit. on pp. 3, 4).

- [8] Anna Krentowska, Elzbieta Sierko-Nobis, Alicja Strzalka, W. Pietruszewska, D. Hempel, and Ewa Sierko. «Awareness of head and neck cancer - a multi-centre survey among young respondents in Poland». en. In: *Int. Dent. J.* 68.6 (Dec. 2018), pp. 441–449 (cit. on p. 3).
- [9] Henri Coutard. «Principles of x ray therapy of malignant diseases». en. In: *Lancet* 224.5784 (July 1934), pp. 1–8 (cit. on p. 4).
- [10] National Cancer Institute. *Radiation Therapy to Treat Cancer was originally published by the National Cancer Institute*. 2019. URL: <https://www.cancer.gov/about-cancer/treatment/types/radiation-therapy> (visited on 11/02/2022) (cit. on p. 4).
- [11] National Cancer Institute. *Chemotherapy to Treat Cancer was originally published by the National Cancer Institute*. 2022. URL: <https://www.cancer.gov/about-cancer/treatment/types/chemotherapy> (visited on 11/02/2022) (cit. on p. 4).
- [12] Jan B Vermorken et al. «Cisplatin, fluorouracil, and docetaxel in unresectable head and neck cancer». en. In: *N. Engl. J. Med.* 357.17 (Oct. 2007), pp. 1695–1704 (cit. on p. 4).
- [13] Yuzuka Kanno, Chang-Yu Chen, Hsin-Lun Lee, Jeng-Fong Chiou, and Yin-Ju Chen. «Molecular mechanisms of chemotherapy resistance in head and neck cancers». en. In: *Front. Oncol.* 11 (May 2021), p. 640392 (cit. on p. 4).
- [14] M M Paulides, H Dobsicek Trefna, S Curto, and D B Rodrigues. «Recent technological advancements in radiofrequency- and microwave-mediated hyperthermia for enhancing drug delivery». en. In: *Adv. Drug Deliv. Rev.* 163-164 (Mar. 2020), pp. 3–18 (cit. on pp. 4, 5).
- [15] Candida S. Punla, Rosemarie C. Farro, and Bataan Peninsula State University Dinalupihan, Bataan, Philippines. «Are we there yet?: An analysis of the competencies of BEED graduates of BPSU-DC». In: *International Multidisciplinary Research Journal* 4.3 (Sept. 2022), pp. 50–59 (cit. on p. 5).
- [16] Paul R Stauffer, Dario B Rodrigues, Randolph Sinahon, Lyndsey Sbarro, Valeria Beckhoff, and Mark D Hurwitz. «Using a conformal water bolus to adjust heating patterns of microwave waveguide applicators». In: *Energy-based Treatment of Tissue and Assessment IX*. Ed. by Thomas P Ryan. San Francisco, California, United States: SPIE, Feb. 2017 (cit. on p. 6).
- [17] H K Lee et al. «Superficial hyperthermia and irradiation for recurrent breast carcinoma of the chest wall: prognostic factors in 196 tumors». en. In: *Int. J. Radiat. Oncol. Biol. Phys.* 40.2 (Jan. 1998), pp. 365–375 (cit. on pp. 6, 7, 18).

- [18] Rossella Gaffoglio, Marco Righero, Giorgio Giordanengo, Marcello Zucchi, and Giuseppe Vecchi. «Fast optimization of temperature focusing in hyperthermia treatment of sub-superficial tumors». In: *IEEE J. Electromagn. RF Microw. Med. Biol.* 5.3 (Sept. 2021), pp. 286–293 (cit. on p. 6).
- [19] G Bruggmoser et al. «Guideline for the clinical application, documentation and analysis of clinical studies for regional deep hyperthermia: quality management in regional deep hyperthermia». en. In: *Strahlenther. Onkol.* 188 Suppl 2.S2 (Sept. 2012), pp. 198–211 (cit. on p. 7).
- [20] H P Kok, P Wust, P R Stauffer, F Bardati, G C van Rhoon, and J Crezee. «Current state of the art of regional hyperthermia treatment planning: a review». en. In: *Radiat. Oncol.* 10.1 (Sept. 2015), p. 196 (cit. on p. 7).
- [21] Zef Rijnen, Jurriaan F Bakker, Richard A M Canters, Paolo Togni, Gerda M Verduijn, Peter C Levendag, Gerard C Van Rhoon, and Margarethus M Paulides. «Clinical integration of software tool VEDO for adaptive and quantitative application of phased array hyperthermia in the head and neck». en. In: *Int. J. Hyperthermia* 29.3 (May 2013), pp. 181–193 (cit. on p. 8).
- [22] Christian Rossmanna and Dieter Haemmerich. «Review of temperature dependence of thermal properties, dielectric properties, and perfusion of biological tissues at hyperthermic and ablation temperatures». en. In: *Crit. Rev. Biomed. Eng.* 42.6 (2014), pp. 467–492 (cit. on pp. 9, 10, 13).
- [23] M W Dewhurst, B L Viglianti, M Lora-Michiels, M Hanson, and P J Hoopes. «Basic principles of thermal dosimetry and thermal thresholds for tissue damage from hyperthermia». en. In: *Int. J. Hyperthermia* 19.3 (Jan. 2003), pp. 267–294 (cit. on p. 11).
- [24] Gerard C van Rhoon, Theodoros Samaras, Pavel S Yarmolenko, Mark W Dewhurst, Esra Neufeld, and Niels Kuster. «CEM43°C thermal dose thresholds: a potential guide for magnetic resonance radiofrequency exposure levels?» en. In: *Eur. Radiol.* 23.8 (Aug. 2013), pp. 2215–2227 (cit. on p. 11).
- [25] E J Hall and L Roizin-Towle. «Biological effects of heat». en. In: *Cancer Res.* 44.10 Suppl (Oct. 1984), 4708s–4713s (cit. on p. 11).
- [26] Eugene H Wissler. «Pennes’ 1948 paper revisited». en. In: *J. Appl. Physiol.* 85.1 (July 1998), pp. 35–41 (cit. on p. 12).
- [27] Esra Neufeld, Maximilian Fuetterer, Manuel Murbach, and Niels Kuster. «Rapid method for thermal dose-based safety supervision during MR scans». en. In: *Bioelectromagnetics* 36.5 (July 2015), pp. 398–407 (cit. on p. 13).
- [28] R J Gillies, P A Schornack, T W Secomb, and N Raghunand. «Causes and effects of heterogeneous perfusion in tumors». en. In: *Neoplasia* 1.3 (Aug. 1999), pp. 197–207 (cit. on p. 14).

- [29] ZMT Zurich MedTech AG. *Zurich Med Tech Sim4Life, version 7.0.2*. 2022. URL: <https://zmt.swiss/sim4life/> (visited on 11/07/2022) (cit. on p. 14).
- [30] ZMT Zurich MedTech AG. *Zurich Med Tech Sim4Life, version 7.0.2, Physics Models: P-EM-FDTD*. 2022. URL: <https://zmt.swiss/sim4life/physics-models/p-em-fdtd/> (visited on 11/07/2022) (cit. on p. 14).
- [31] ZMT Zurich MedTech AG. *Zurich Med Tech Sim4Life, version 7.0.2, Physics Models: P-THERMAL*. 2022. URL: <https://zmt.swiss/sim4life/physics-models/p-thermal/> (visited on 11/07/2022) (cit. on p. 15).
- [32] IT'IS Foundation. *Tissue properties database V4.1*. 2021 (cit. on pp. 15, 27, 28).
- [33] Margarethus M Paulides, Jurriaan F Bakker, Nicolas Chavannes, and Gerard C Van Rhoon. «A patch antenna design for application in a phased-array head and neck hyperthermia applicator». en. In: *IEEE Trans. Biomed. Eng.* 54.11 (Nov. 2007), pp. 2057–2063 (cit. on p. 16).
- [34] Maarten Paulides, S.H.J.A. Vossen, Adrianus Zwamborn, and Gerard van Rhoon. «Theoretical Investigation of the Feasibility to Deposit RF Energy Centrally in the Head-and-Neck Region». In: *International Journal of Radiation: Oncology - Biology - Physics* (Feb. 2005). URL: <http://hdl.handle.net/1765/8080> (cit. on p. 16).
- [35] F M Henke, W T Joines, and T V Samulski. «Variations of focal regions versus numbers and positions of sources in two-dimensional media». en. In: *Int. J. Hyperthermia* 17.5 (Sept. 2001), pp. 382–400 (cit. on p. 16).
- [36] «Antenna Elements and Arrays». In: *The Electrical Engineering Handbook*. Elsevier, 2005, pp. 569–583 (cit. on p. 17).
- [37] R A M Canters, M Franckena, M M Paulides, and G C Van Rhoon. «Patient positioning in deep hyperthermia: influences of inaccuracies, signal correction possibilities and optimization potential». en. In: *Phys. Med. Biol.* 54.12 (June 2009), pp. 3923–3936 (cit. on p. 21).
- [38] R A M Canters, P Wust, J F Bakker, and G C Van Rhoon. «A literature survey on indicators for characterisation and optimisation of SAR distributions in deep hyperthermia, a plea for standardisation». en. In: *Int. J. Hyperthermia* 25.7 (Nov. 2009), pp. 593–608 (cit. on p. 21).
- [39] J Kennedy and R Eberhart. «Particle swarm optimization». In: *Proceedings of ICNN'95 - International Conference on Neural Networks*. Perth, WA, Australia: IEEE, 2002 (cit. on p. 21).
- [40] I M Sobol'. «On the distribution of points in a cube and the approximate evaluation of integrals». en. In: *U.S.S.R. comput. math. math. phys.* 7.4 (Jan. 1967), pp. 86–112 (cit. on p. 28).