

# Temperature Imaging of Thermal Therapy using Finite Element Method

Sijia Huang

September 17, 2014

## Contents

<b>1</b>	<b>Background</b>	<b>2</b>
1.1	Thermal Therapy . . . . .	2
1.2	Temperature Imaging using Ultrasound . . . . .	2
1.3	COMSOL Tools . . . . .	3
<b>2</b>	<b>COMSOL Model</b>	<b>3</b>
2.1	Arguments Setting . . . . .	3
2.1.1	Heat Pattern in Homogeneous Tissue . . . . .	3
2.1.2	Heat Pattern in Mixed Tissue . . . . .	4
2.1.3	Heat Pattern in Mixed Tissue with Perfusion Added . . . . .	6
2.2	Output COMSOL Generates . . . . .	7
<b>3</b>	<b>Matlab Control of COMSOL</b>	<b>7</b>
3.1	Automatic COMSOL Running using Matlab Sever Command . . . . .	8
3.2	Processing of COMSOL Data using Matlab . . . . .	9
3.2.1	Read COMSOL Data . . . . .	10
3.2.2	Generate Thermal Image . . . . .	10
3.2.3	Plot Slice . . . . .	10
<b>4</b>	<b>Results</b>	<b>11</b>
4.1	Simulation Results for Homogeneous Tissue . . . . .	11
4.2	Simulation Results for Mixed Tissue . . . . .	12
4.3	Simulation Results for Mixed Tissue with Perfusion Added . . . . .	13
4.4	Animation . . . . .	14
<b>5</b>	<b>Conclusion</b>	<b>14</b>
<b>6</b>	<b>Future Work</b>	<b>15</b>
6.1	COMSOL Limitation . . . . .	15
6.1.1	Tissue Pattern . . . . .	15
6.1.2	Object Oriented Selection . . . . .	15
6.2	Vessel Pattern . . . . .	15
6.3	Simulation in 3D . . . . .	16
<b>7</b>	<b>Appendix A - Matlab Code</b>	<b>18</b>

# 1 Background

## 1.1 Thermal Therapy

Nowadays, cancer becomes a leading cause of mortality worldwide. Therefore, diagnosing and treating cancer have always remained urgent tasks in medical and clinical fields. Normally, we treat cancer with Surgery and radiation therapy, which has high-risk and may trigger serious side effect. Therefore, Thermal Therapy, which is low-risk in comparison, becomes more and more widely used in the treatment for pathological tissues (e.g. tumor tissue).

The principle of Thermal Therapy is killing the tissues with overheat or freezing. In general, Thermal therapy consists of two methods called Hyperthermia and Ablation. Hyperthermia is the therapy which treats the target tissue with continuous and constant heat, which is normally generated by modern microwave or ultrasound array and the temperature is about 6°C higher compared to normal temperature of human body. Hyperthermia is suitable for treatment on large tissue region, breast cancer for example.

Ablation, on the other hand, is focused on specific points instead of large region. Therefore, it supports a wider temperature range than hyperthermia, usually from -20°C to 100°C. Nowadays, two types of Ablation are widely used in thermal treatment. The first one is called Cryo ablation, which is a process using extreme cold to destroy or damage cancer tissues. The other one is called Radio frequency (RF), using the heat generated from the high frequency alternating current to treat medical disorder [15].

However, the major limitation of thermal therapies is that healthy tissues near the target tissues are easily damaged during the heating process. A method to help us get the detailed thermal information during the therapy is needed. In other words, we are seeking methods that can monitor and guide the heating process in order to reduce the effect of overheat on normal tissue.

There are several ways of measuring temperature non-invasively, such as MRI [7] [?]. MRI meets accuracy and spatial resolution requirement, but it is expensive, and requires a fixed installation. On the other hand, Ultrasound is a non-ionizing, convenient, and inexpensive modality with relatively simple signal processing requirements. For those therapies using Ultrasound array for heat generation, it seems an ideal method if we can use the device for thermal monitor at the same time. These attributes make Ultrasound an attractive method to use for temperature estimation if a temperature- dependent ultrasonic parameter can be measured [2].

## 1.2 Temperature Imaging using Ultrasound

Since Ultrasound is an inexpensive and non-invasive modality for thermal imaging, studies on the thermal dependence Ultrasound parameters like speed of sound, echo shift, and backscattered energy [2] have been launched. The exploiting directions for using ultrasound as a non-invasive thermometer fall into three categories: 1. Echo-shifts due to changes in tissue thermal expansion and speed of sound (SOS), 2. Change in acoustic attenuation coefficient and 3. Change in backscattered energy (CBE).

Our group are interested in monitoring thermal changing using change in backscattered energy [14], [4], [3]. Straube and Arthur had predicted a monotonic change in backscattered energy with the rise of temperature [13]. And Debomita had added the discrete scatter model developed by Jason Trobaugh [14] onto the COMSOL generated temperature maps in order to verify and validate temperature imaging using CBE [5]. However, Debomita only tested non-uniform heating on homogenous tissue. The simulation showed blow is an extension for Debomita's study. The model imported to COMSOL has extended to mixed tissue model which is derived from the result of Gomori Stained turkey meat and the model with perfusion added [10].

### 1.3 COMSOL Tools

COMSOL Multiphysics is a finite element analysis, solver and Simulation software package for various physics and engineering applications, especially coupled phenomena, or multiphysics. COMSOL Multiphysics also offers an extensive interface to Matlab and its toolboxes for a large variety of programming, pre-processing and post-processing possibilities [16]. Our simulation uses the advantages of interfacing with Matlab and builds COMSOL models using Matlab command. COMSOL generates Delaunay mesh on the provided model and calculate the temperature at each Delaunay nodes using Penns bio-heat equation. A set of output temperature maps at different time is returned and stored in the desired folder as *.txt* files.

## 2 COMSOL Model

### 2.1 Arguments Setting

#### 2.1.1 Heat Pattern in Homogeneous Tissue

We start by studying the non-uniform heating pattern under 3 basic tissue types in the body: muscle, collagen and lipid. These basic scenarios are simulated under uniform circle region. The diameter of the circle is standard 80mm which corresponds to the size of ultrasound transducer. A water tube with diameter 10mm is placed at the center of the region, which has been regarded as the heat source in the system.

The heating pattern will become quite predictable due to the homogeneous property. However, what makes them distinguish from each other is the time different model used to achieve steady state based on tissue types. Since the thermal parameters related to three mediums varies, the pattern of heat flow expanded over time will be different. By observing the heat flow simulated using COMSOL, we can then progress in predicting heat flow under different cases in real life.

The heat is produced by a 65°C water tube heat source with a 37°C surrounding temperature, then the heat flow for different materials are calculated using Pennes Bioheat Equation:

$$\rho C_p \frac{\partial T}{\partial t} = \nabla (k \nabla T) + Q \quad (1)$$

where  $\rho$  and  $C_p$  are the density and specific heat of the specimen.  $k$  is heat conductivity of the specimen.  $T$  is the surrounding temperature which is 37°C here and  $Q$  is the heat deposited to the specimen. Table 1 lists the thermal parameters for different tissues [5].

Tissue Type	Conductivity W/(K*m)	Specific Heat J/(g*K)	Density $g/cm^3$
Water	0.6	4.185	1
Collagen	0.47	3.500	1.097
Muscle	0.5	3.639	1.05
Lipid	0.5	2.500	0.916

Table 1: Thermal parameters of tissues

The time in the study ranges from 0s (initial condition) to 600s with 100s step. COMSOL generate a Delaunay Triangulation mesh (Figure 1) to discretize the spatial domain. Then, the temperature at each Delaunay node is calculated with equation (1).

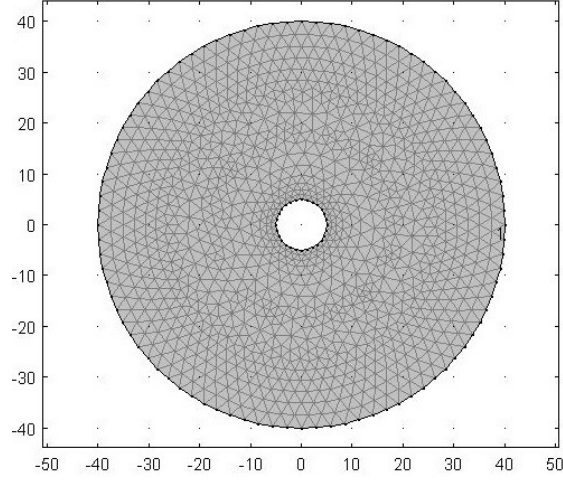


Figure 1: COMSOL Generated Delaunay Triangulation Mesh for Homogeneous Tissue Type

### 2.1.2 Heat Pattern in Mixed Tissue

Biological tissue, however, is inhomogeneous. Therefore, our next step was simulating the heat flow with mixed tissue types. In order to make the simulation meaningful and has some practical significance, we scanned a piece of turkey meat and did quantized analysis by Matlab.

One of the limitations of COMSOL is that it cannot create a new type of tissue with random combination of the existing materials (e.g. 30% lipid and 70% collagen). Therefore, clustering within the desired region was done and the material property of the cluster cell was assigned with the tissue which dominates the cluster cell. For example, if the cluster contains 70% muscle tissues, 12% collagen tissues and 18% lipid tissues, then the cell would be defined as cluster of muscle. Then we imported the quantized turkey meat data into COMSOL and calculated the results.

The basic arguments are the same as in homogeneous model. In order to distinguish between different tissue components, we applied Gomori Trichrome Stain method [6] since it can emphasize the morphology of the specimen. The stain resulted in good contrast among various kinds of tissue. The turkey meat had been stained for 30min and scanned using a particular scanning set. The image was send to Matlab for further quantized analysis.

Based on [6], the stain left muscle dark red and collagen medium blue, while lipid largely unstained and remained its natural color. Therefore, we did RGB channel analysis and regarded pixels with R value in range (150, 160) as muscle, pixels with B value in range (45, 65) as collagen and the others as lipid. Then we took a 1cm by 1cm region, divided it into 3 by 3 cluster, counted the number of pixels for each tissue type within the cluster and assigned the whole cluster with the kind of tissue which dominants the region. The original 1cm by 1cm region and the clustering results are showed in Figure 2 and Figure 3 below.

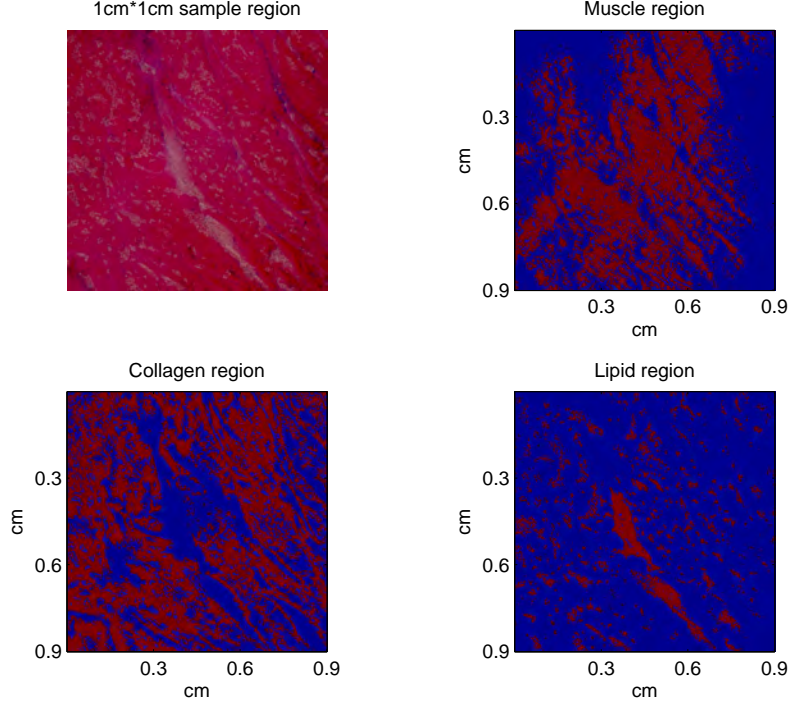


Figure 2: Results of component analysis using Gomori Trichrome Stain

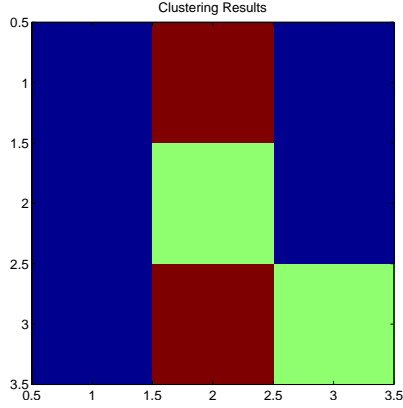


Figure 3: Results of clustering after quantized analysis. And the color legend is: Blue-Muscle, Green-Collagen and Red-Lipid

Then we built a 3 by 3 square in COMSOL and scaled the region size to standard 80mm. The region property for each cluster cell was assigned based on Matlab results.

The time in the study ranges from 0s (initial condition) to 3600s (an hour) with 600s step. COMSOL generate a Delaunay Triangulation mesh to discretize the spatial domain (Figure 4). Then, the temperature at each Delaunay node is calculated with equation (1).

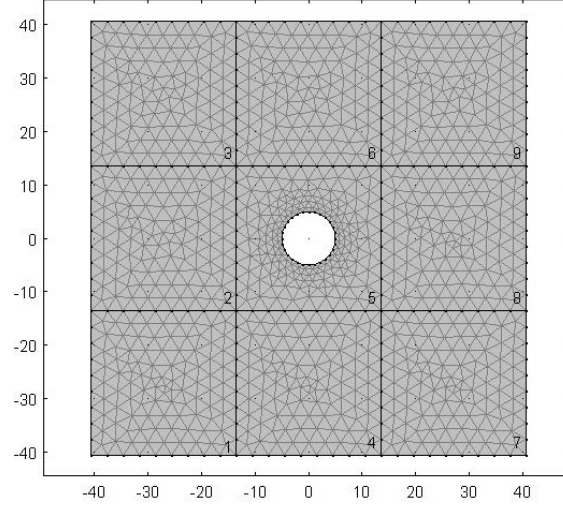


Figure 4: COMSOL Generated Delaunay Triangulation Mesh for Mixed Tissue Type. Clustering into 3 by 3 region after quantized analysis using Matlab

### 2.1.3 Heat Pattern in Mixed Tissue with Perfusion Added

Then we started considering the effect of perfusion of vessel structure. The blood circulates in the vessel can carry away part of the heat, resulting in a local minimum temperature around the vessel. For convenience, we call the region around vessel a "cool spot". The size of the cool spot may be determined by many factors: the flow rate of the blood, the size of the vessel or the forms and position of the vessel. To observe the perfusion effect, we imported vessel structure to COMSOL model based on Figure 5 from [10] which is an image of human liver. Zone 1 encircles a series of vessels where the oxygenated blood from hepatic arteries enters. Zone 3 encircles central veins, where oxygenation is poor. In our simulation, we elided the factors like vessel forms or the size of vessel and regarded the temperature within any vessel area a steady  $37^{\circ}\text{C}$ .

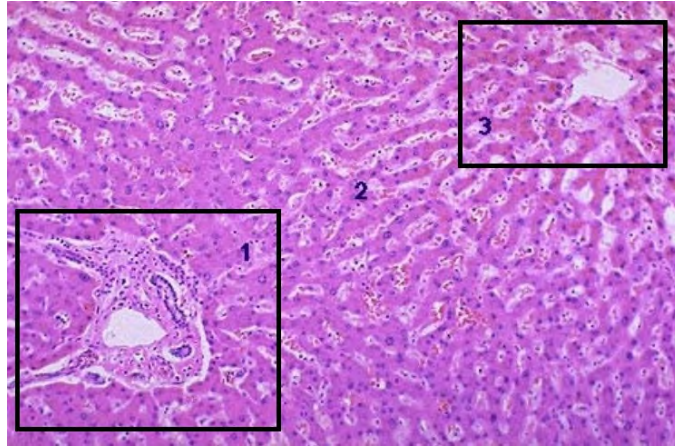


Figure 5: Vessel Structure Derived from Liver Image

The model is calculated under Pennes Bioheat Equation with perfusion term being added:

$$\rho C_p \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) + Q + \omega_b C_b (T_a - T) \quad (2)$$

where  $\omega_b$  and  $C_b$  are the rate of blood perfusion and specific heat of blood while  $T_a$  is the arterial blood temperature.

The basic arguments such as material prosperities were the same as in homogeneous model. And tissue clustering was based on the turkey analysis. In this simulation, we added perfusion using vessel structure based on [10].

As you can see in Figure 6, a group of vessels had been added at the bottom left region based on the vessel structure circulated in Zone 1 showed in Figure 5. The centered artery has a diameter of 6mm, surrounded by two small round vessels (with diameter 2mm and 1.4mm) and a vessel transaction in 2mm by 8mm ellipse shape. A vein with diameter 6mm has been added at the upper right region based on Zone 3 in Figure 5.

The time in the study ranges from 0s to 3600s (an hour) with 600s step. COMSOL generate the corresponding Delaunay Triangulation mesh to discretize the spatial domain (Figure 6). Then, the temperature at each Delaunay node is calculated with equation (1).

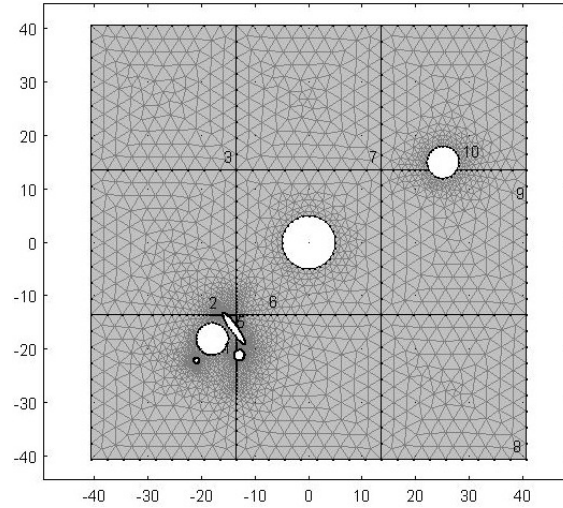


Figure 6: COMSOL Generated Delaunay Triangulation Mesh for Mixed Tissue Type with Vessel Added. Vessel structure added refer to Figure 5

## 2.2 Output COMSOL Generates

The output of COMSOL was a series of *.txt* files. Each of them stored the temperature information under a specific time. The data in *.txt* files were classified into three groups: 1.Coordinates, which recorded the coordinate of each Delaunay node; 2.Elements(Triangles), which recorded the corresponding triangle number of each node; 3.Data, which recorded the corresponding temperature at each node. The output *.txt* files were stored under the named folder and would be read by Matlab for further processing.

## 3 Matlab Control of COMSOL

The Matlab routine is stored in the package named *comsol\_sijia*. Two main scripts *COMSOL\_control* and *MatlabImage\_Processing* are the user controlled files which enable user to set up their desired



arguments. The other six function files are all installed function in charge of the backend control and calculation.

### 3.1 Automatic COMSOL Running using Matlab Sever Command

In this part we run COMSOL in the background using Matlab sever commands. To start sever, be sure to open "COMSOL 4.1 with Mablab.exe in your program bar. The Matlab interface and command window are as usual, the only difference is that it is opened with linking to the COMSOL end thus it can process COMSOL commands.

After open Matlab, go to package *comsol\_sijia* and open COMSOL\_CONTROL.m. The script is consisted of two parts: 1. **User defined arguments for COMSOL**, which includes all the variables which can be set based on user's experiment. 2. **COMSOL Run**, which takes the user input and runs the corresponding function to generate the final results.

Now Let's take a detailed view into those user defined arguments described in the script showed in Figure 7.

```
%% ***** User defined arg for COMSOL *****

MODEL = 3; % Choose the model you like and then set the arg based on your choose
           % Homo = 1; Mixed = 2; Mixed_Perfusion = 3

path = 'C:\comsol\comsol_sijia\'; % the path for storing your generated data
%path = 'C:\comsol\'; % the path for storing your generated data

if (MODEL == 1)
    % For MODEL == Homo
    range = 80; % the diameter of your circle(in mm)
    radiusOfTube = 5;
    MatType = 1; % 1.collagen 2.muscle 3.fat (Default type is collagen)
    StartTime = 0;
    EndTime = 1000;
    TimeStep = 100;

elseif (MODEL == 2)
    % For MODEL == Mixed
    range = 81; % in mm (the number should be divisible by 3)
    radiusOfTube = 5;
    StartTime = 0;
    EndTime = 3600;
    TimeStep = 600;

elseif (MODEL == 3)
    % For MODEL == Mixed_Perfusion
    range = 81; % in mm (the number should be divisible by 3)
    radiusOfTube = 9;
    StartTime = 0;
    EndTime = 4800;
    TimeStep = 600;

else
    disp('Wrong input. Please enter MODEL type: Homo = 1; Mixed = 2; Mixed_Perfusion = 3');
    break;
end
```

Figure 7: User defined arguments in COMSOL\_control script

- Model: number 1 stands for homogeneous model, number 2 stands for 3 by 3 mixed tissue model, number 3 stands for mixed tissue with vessel added. If you enter any number except the indicated one, you will be required to re-enter your model number.
- Path: The result of COMSOL will be a series of temperature data. The information of temperature images at different time are stored in separate *.txt* files. Therefore, path argument indicates the path for storing your generated COMSOL text data.
- For different models, you have to set:
  1. range: the size of your region



2. radiusOfTube: the radius of the water tube placed in the middle
3. MatType (only if you choose model as homogeneous one): number 1 as collagen; number 2 as muscle; number 3 as fat
4. StartTime (EndTime): start (end) time of your study
5. TimeStep: time step you choose between each plot

After this step, the COMSOL generated temperature data will be stored in the folder under user specified path and all the relevant user defined arguments will be saved in a *.mat* file called *UserArg.mat* under current path.

### 3.2 Processing of COMSOL Data using Matlab

This part is the Matlab processing of the COMSOL passed data. Open script *Matlab\_ImageProcessing.m* and you will see the code is consisted of two parts: 1. **User defined arguments for Matlab**, which includes all the variables which can be set by user. 2. **Matlab Processing**, which loads *UserArg.mat* from pervious result, takes the user input, and runs the corresponding data processing functions to generate the final Matlab images Figure 8.

```
%% ***** User defined arg for MATLAB *****
MODEL = input('Please input the type of your model(Homo = 1; Mixed = 2; Mixed_Perfusion = 3): ');

if (MODEL == 1)
    % For MODEL == Homo
    s1 = 1; s2 = 4; s3 = 7; s4 = 11; % Select the slice number you want to see
    title = ' Homo tissue'; % the title showed in Matlab image
    nameSave = 'Homo';
elseif (MODEL == 2)
    % For MODEL == Mixed
    s1 = 1; s2 = 2; s3 = 3; s4 = 7; % Select the slice number you want to see
    title = ' Mixed tissue'; % the title showed in Matlab image
    nameSave = 'Mixed';
elseif (MODEL == 3)
    % For MODEL == Mixed_Perfusion
    s1 = 1; s2 = 6; s3 = 7; s4 = 9; % Select the slice number you want to see
    title = ' Mixed tissue with Perfusion'; % the title showed in Matlab image
    nameSave = 'Mixed_Vessel';
else
    disp('Wrong input. Please enter MODEL type: Homo = 1; Mixed = 2; Mixed_Perfusion = 3');
    break;
end
```

Figure 8: User defined arguments in *Matlab\_ImageProcessing* script

You have to set:

- s1-s4: 4 slices you want to see. E.g. if the time range is 0:600:3600, you have 7 slices in total, then slice=1 will be 0s, slice=5 will be 2400s, etc.
- name: the title showed in Matlab image
- nameSave: the name of the saved *.eps* image

The program will let you re-enter the model you selected in previous section. Then load the *User-Arg.mat* file based on your model type.

The following sections are some details of functions in Matlab processing part: *readComsolData()*, *GenTempImage()*, and *plotSlice()*.

### 3.2.1 Read COMSOL Data

The *.txt* data file passed by COMSOL is made up of three parts: 1.Coordinates, which records the coordinate of each Delaunay node 2. Elements(Triangles), which records the corresponding triangle number of each node 3. Data, which records the corresponding temperature at each node. `readComsolData()` function reads string data into mat file and save it in the structure named *pd*. It divides the whole file into 3 pieces by comparing strings among the headers of each part, and then covert the data into matrix by part and store them in *pd* structure. *pd.p* is coordinates information, *pd.t* is triangle information, and *pd.d* is temperature information. *pd* will be returned and passed to the next function.

### 3.2.2 Generate Thermal Image

`GenTempImage()` takes results from `GenTempImage()` as input, interpolates points into the Delaunay mesh grid, then scale temperature data onto that grid. If you are under Homogenous model, the function will set the temperature in the tube 65°C, and surrounding temperature 37°C.

### 3.2.3 Plot Slice

`plotSlice()` plots the selected slices and save the images in *.eps* format. You can launch the function along once you have the *.mat* file provided by `GenTempImage()`, which contains all the information needed for this function. Therefore, you only need to enter 4 desired slice numbers and the other inputs will passed by the *.mat* file.

## 4 Results

### 4.1 Simulation Results for Homogeneous Tissue

We simulated the model described in section 2.1.1 under homogeneous collagen, muscle and lipid tissue type. The following images in Figure 9 are the simulation of the heat flow in the three scenarios after 100s, 200s, and 600s.

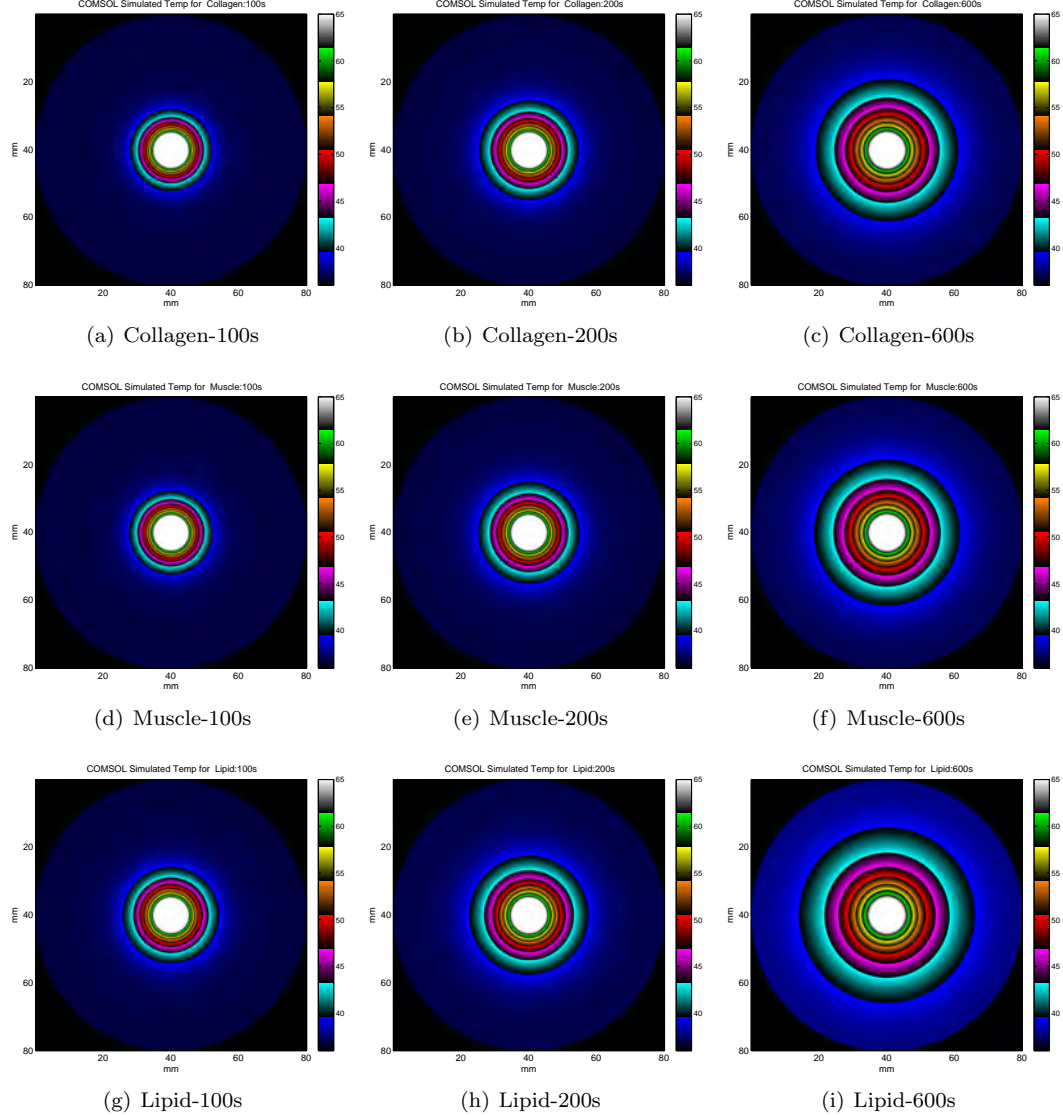


Figure 9: Heat distribution in homogeneous tissue type after (a)100s, (b)200s, (c)600s

The three consecutive sub-images in horizontal direction stand for the simulation under the same tissue type: read from top to bottom are collagen, muscle and lipid. The three consecutive sub-images in vertical direction stand for the simulation of heat flow after same duration of time. All images are generated under same temperature range: 36°C to 65°C thus we can compare one with the others.

By comparing in horizontal direction, we can observe how heat flow expands over time. The

heating patterns are symmetric since we are monitoring under homogeneous region. By comparing in vertical direction, we can see the difference in heat expanding speed under different tissue types. From the images, Lipid owns the fastest heat diffusion speed among the three types and this result fits the Pennes Bioheat Equation. For Lipid, the ratio of heat conductivity  $k$  with the product of density  $\rho$  and specific heat  $C_\rho$  is smallest among those three. Therefore, to achieve the same spatial heating pattern, which means  $\nabla T$  in the equations are same,  $\frac{\partial T}{\partial t}$  will be biggest for lipid specimen, followed by collagen and muscle.

## 4.2 Simulation Results for Mixed Tissue

We simulated the model described in section 2.1.2 under the generated mixed tissue type. The initial condition for the turkey meat was set to  $37^\circ\text{C}$  and the boundary condition of the heat source is  $65^\circ\text{C}$ . Figure 10 is the simulation of the temperature distributions based on the scenarios after 0s, 600s, 1200s, and 3600s.

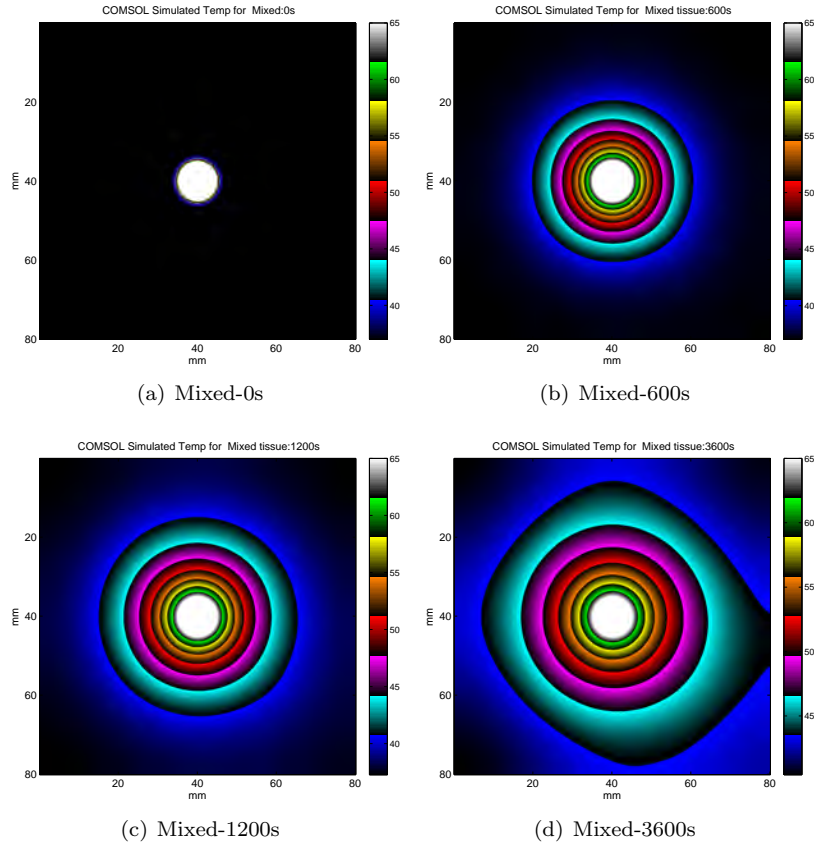


Figure 10: Heat distribution with mixed tissue type after (a)0s, (b)600s, (c)1200s, (d)3600s

The temperature map is uneven due to inhomogeneous distribution of the tissue. Two observations are made based on Figure 3 and Figure 10: First, the heat flows fast towards the lower right region which is dominated by lipid tissue. The result make sense using the conclusion derived from the simulation under homogeneous environment (2.1.2). Under same time duration, lipid tissue owns the fastest thermal conductivity. Therefore, under the scenario with mixed tissues, the heat tends to flow to the direction where lipid dominates. Second, the heat tends to flow to the upper right

region which is the boundary of lipid and collagen rather than the lower right with is the boundary of lipid and muscle. The phenomenon can also be explained using conclusion in 2.1.2 since collagen conduct heat faster than muscle based on their thermal parameter.

### 4.3 Simulation Results for Mixed Tissue with Perfusion Added

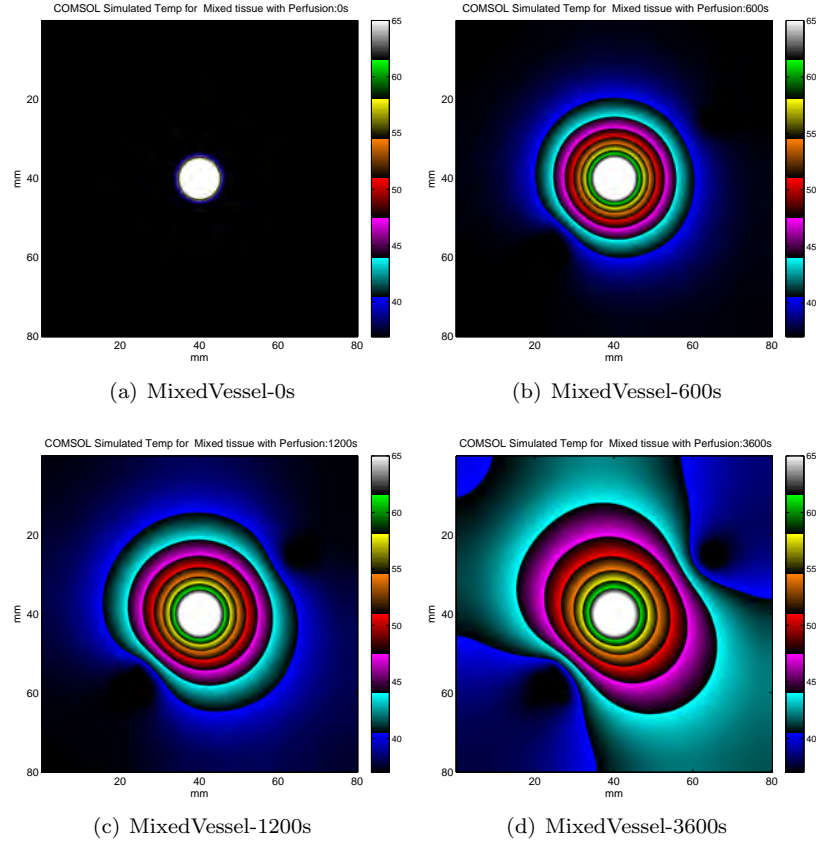


Figure 11: Heat distribution under mixed tissue type with perfusion added after (a)0s, (b)600s, (c)1200s, (d)3600s

As you can see in Figure 11, two cold spots are resulting in two vessel groups as they are carrying heat away. The vessels kind of "block" the heat from expanding any further to its region. And the size of the cold spot is closely depending on the size of the vessel group. The next step is to find a converge/steady state with the perfusion effect by setting the time to extremely large.

## 4.4 Animation

Loading stHomo.Col.avi

Loading stMixed.avi

Loading stMixed.Vessel.avi

## 5 Conclusion

In our study, we have simulated the temperature distribution under homogenous tissue type, mixed tissue type and with perfusion added. The simulation was calculated by COMSOL with control using Matlab. The thermal parameters for different tissue types were passed to COMSOL and Pennes Bioheat Equation is used to calculate the heat flow. Then the temperature was evaluated at every node on the Delaunay mesh COMSOL generated.

Our simulation achieved semi-auto control using Matlab. Users only need to load the given routine package, input those required user defined arguments, and then the routine will in charge of everything including: 1. run COMSOL in background, 2. pass the COMSOL data back to Matlab, and 3. generate processed temperature data in Matlab.

From the set of images in the result section, we observed the time difference for distinct tissue type to achieve same temperature pattern. We also got the idea of how blood flow affected the heating process and what were the determination factors in perfusion effects. The conclusion will further help us evaluating the thermal dose needed in the treatment in real life.

However, due to some of the COMSOL limitations, our simulation cannot be set under completely similar conditions to real life model.

1. The tissue model is clustered in 3 by 3 blocks as COMSOL cannot go any further to 4 by 4 due to Jacobian equation limitation.
2. Each node can only be assigned with one particular tissue type other than a percentage combination of different tissue types in real life.
3. We measured the temperature within blood vessel a steady 37 °C. However, based on DSK Lu and SS Raman [8], vessel with diameter less than 3 mm will occluded and show vessel wall injury due to continuous heating.

The research studies under these limitations will be the goal for our further experiment. And we are recommending to explore the new released COMSOL 4.4 version and see if the bio-heat package provided by the new version capable in solving the problems listed above.

## 6 Future Work

### 6.1 COMSOL Limitation

#### 6.1.1 Tissue Pattern

In reality, the component of the turkey slice presented in experiment is complex. A particular unit may be made up of 30% lipid and 70% collagen. However, COMSOL 4.1 cannot create a new type of tissue with random combination of the existing materials. We use clustering method to reduce the complexity to clusters' properties. This limitation creates gap between the simulation results and the observation results in real experiment. In the future, we will explore the upgrade version of COMSOL, see if there is any new features with capabilities of combining multiple tissue types. If so, we could then load the analyzed tissue components from Matlab to COMSOL and generate thermal parameters for different complex tissue.

#### 6.1.2 Object Oriented Selection

In our simulation, we have successfully achieved COMSOL control using Matlab and made COMSOL run in background to get the result. The program is almost automatic except the selection creation part. After setting geometry, we have to make selections e.g. boundary selection of tube/vessel, region selection based on the clustering results. However, since COMSOL 4.1 is not object oriented designed software, we could not make the selection using name of the object we assigned to them. COMSOL will set numbers to regions and boundaries based on its own rules which is clearly, not based on the order you create them. Therefore, when we try to set selections, we have to go into COMSOL interface, record the desired region and boundary numbers, and then set selection (Figure 12) in Matlab using the numbers you recorded. We will explore the upgrade version of COMSOL (e.g. COMSOL 4.3) as further study and optimize our code if the object oriented characteristic is enabled in COMSOL.

```
% *****Set selection*****
% NOTE: The number do not following your self-def. COMSOL has its own rules.
model.selection.create('sel1');
model.selection('sel1').set([1 2 3 4 9 10]);
model.selection('sel1').name('collagen');
model.selection.create('sel2');
model.selection('sel2').set([5 7]);
model.selection('sel2').name('muscle');
model.selection.create('sel3');
model.selection('sel3').set([6 8]);
model.selection('sel3').name('fat');
model.selection.create('sel4');
model.selection('sel4').geom(1);
model.selection('sel4').set([31 52 53 54]);
model.selection('sel4').name('water tube');
model.selection.create('sel5');
model.selection('sel5').geom(1);
model.selection('sel5').set([29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 48 49 50 55 56 57 58 59 60]);
model.selection('sel5').name('vessel');
```

Figure 12: User defined selections in Matlab. The numbers with the red box are the boundary and region number read from COMSOL interface

### 6.2 Vessel Pattern

Our simulation have imported the model in [10]. In the future, we will take more vessel patterns found in thermal therapy and learn the detailed effects producing by different type of vessels during heating process. Evidence showed in [8] experiment result that vessels less than 3mm get damaged and even occluded by continuous heating. The greater the vessel diameter is, the less likely the vessel wall get injury. Therefore, we will test and validate the conclusion in [8] by COMSOL model, and if it is true, we will consider vessels greater than 3mm as steady 37°C and minimize the impact



of vessels less than 3mm on the heat pattern as time grows in our future models.

The other concern is finding a method importing vessel patterns. In our simulation, vessel geometry is manually set by the model designer. In the presentation of research by Gal Shafirstein [9], Chris Diederich [1], and John Pearce [11] at 2014 Society for Thermal Medicine Annual Meeting, we have already seen the evidence of directly importing the setting geometry into COMSOL 4.3. If we can explore this function provided by COMSOL, it will be much easier for us to learn existing vessel patterns.

### 6.3 Simulation in 3D

In [12], a multilayer structure within a malignant tumor is imported and the temperature is calculated using 3D bio-heat equation and Maxwell's equations. As we successfully simulated heat distribution in 2D, the next step is moving to 3D structure and importing the model to COMSOL layer by layer. We can start by reproducing the model in [12] which is show in Figure13.

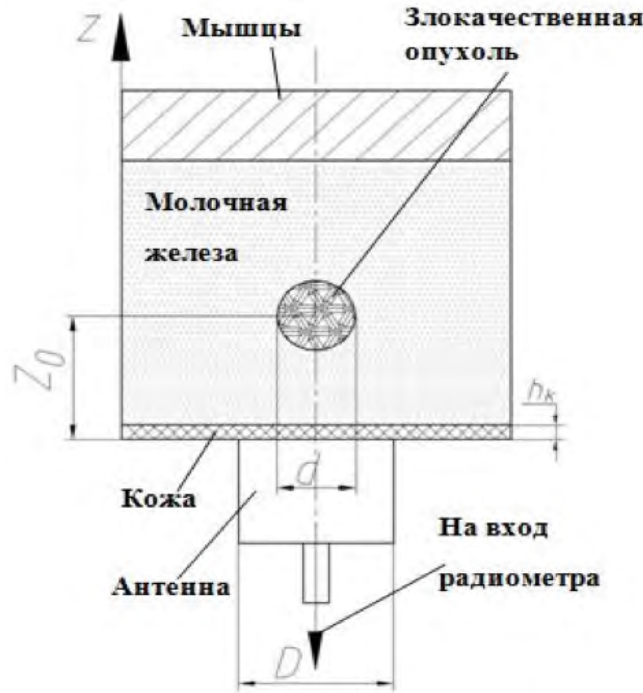


Figure 13: Model of Multilayer Structure within a Malignant Tumor, from Vesnin SG and Sedankin MK [12]

## References

- [1] Matthew Adams, Serena Scott, Vasant Salgaonkar, Graham Sommer, and Chris Diederich. Theoretical design and evaluation of endoluminal ultrasound applicators for thermal therapy of pancreatic cancer under image guidance. In *Society for Thermal Medicine Annual Meeting*, May 2014.
- [2] R.M Arthur. Temperature imaging using ultrasound. In EG Moros, editor, *Physics of Thermal Therapy: Fundamentals and Clinical Applications*, Imaging In Medical Diagnosis And Therapy, chapter 13, pages 1–15. Taylor & Francis, 2012.
- [3] R.M Arthur, Debomita Basu, Yuzheng Guo, Jason W. Trobaugh, and Eduardo G. Moros. 3D in vitro estimation of temperature using the change in backscattered ultrasonic energy. *IEEE Transactions on Ultrasonics*, 2010.
- [4] R.M Arthur, Jason W. Trobaugh, William L. Straube, and Eduardo G. Moros. Temperature dependence of ultrasonic backscattered energy in motion-compensated images. *IEEE TRANSACTIONS ON UFFC*, February 2006.
- [5] Debomita Basu. *3D Temperature Imaging Using Ultrasonic Backscatter Energy During Non-Uniform Tissue Heating*. PhD thesis, Department of Electrical and Systems Engineering, Washington University in St. Louis, St. Louis, MO, May 2010.
- [6] Laura M. Hinkelman, Leon A. Metlay, Charles J. Churukian, and Robert C. Wang. Modified Gomori Trichrome stain for macroscopic tissue slices. *The Journal of Histotechnology*, Vol.19(No.4), 1996.
- [7] K Hynynen, A Chung, M Buchanan, and T Fjeld. Feasibility of using ultrasound phased arrays for MRI monitored noninvasive surgery. *IEEE Trans on UFFC*, 43:1043–1053, 1996.
- [8] DSK Lu, SS Raman, D Vodopich, M Wang, J Sayre, and C Lassman. Effect of vessel size on creation of hepatic radiofrequency lesions in pigs: Assessment of the “heat sink” effect. *Am J Roentgen*, 178(1):47–51, 2002.
- [9] Emily Oakley, Brian Warzen, Hassan Arshad, and Gal Shafirstein. A new algorithm for light dosimetry during interstitial laser therapy of locally advanced tumors. In *Society for Thermal Medicine Annual Meeting*, May 2014.
- [10] University of Utah. <http://library.med.utah.edu/webpath/livehtml/liver003.html>, 2014.
- [11] John Pearce, Alicia Petryk, and Jack Hoopes. FEM numerical models of magnetic nanoparticle heating facilitate treatment planning and analysis. In *Society for Thermal Medicine Annual Meeting*, May 2014.
- [12] Vesnin SG and Sedankin MK. Miniature antennas for medical microwave radiometers. *Biomedical Technologies and Radio Electronics*, 10(4):51–56, 2011.
- [13] W.L. Straube and R.M Arthur. Theoretical estimation of the temperature dependence of backscattered ultrasonic power for noninvasive thermometry. *UTRASOUND IN MEDICINE AND BIOLOGY*, 20:915–922, 1994.
- [14] JW Trobaugh and RM Arthur. A physically based, probabilistic model for ultrasonic images incorporating shape, microstructure, and system characteristics. *IEEE Trans. Ultrasonics, Ferroelectrics, and Frequency Control*, 48(6):1594–1605, 2001.
- [15] Wikipedia. <http://en.wikipedia.org/wiki/ablation>, 2014.
- [16] Wikipedia. <http://en.wikipedia.org/wiki/comsol>, 2014.

## 7 Appendix A - Matlab Code

### COMSOL\_control.m

```
% Author: Sijia Huang
% Date: 07/09/2014

% COMSOL_CONTROL : This is the main file of COMSOL modeling control.

clear; clc; close all;

%% User defined arg

MODEL = 3; % Choose the model you like and then set the arg based on your choose
          % Homo = 1; Mixed = 2; Mixed.Perfusion = 3

%path = 'C:\comsol\comsol.sijia\'; % the path for storing your generated data
path = pwd; % the path for storing your generated data

%% ***** User defined arg for COMSOL *****
if (MODEL == 1)
    % For MODEL == Homo
    range = 80; % the diameter of your circle(in mm)
    radiusOfTube = 5;
    MatType = 1; % 1.collagen 2.muscle 3.fat (Default type is collagen)
    StartTime = 0;
    EndTime = 1000;
    TimeStep = 100;
elseif (MODEL == 2)
    % For MODEL == Mixed
    range = 81; % in mm (the number should be divisible by 3)
    radiusOfTube = 5;
    StartTime = 0;
    EndTime = 3600;
    TimeStep = 600;
elseif (MODEL == 3)
    % For MODEL == Mixed.Perfusion
    range = 81; % in mm (the number should be divisible by 3)
    radiusOfTube = 9;
    StartTime = 0;
    EndTime = 4800;
    TimeStep = 600;
else
    disp('Wrong input. Please enter MODEL type: Homo = 1; Mixed = 2; Mixed.Perfusion = 3');
    break;
end

%% COMSOL Run
if (MODEL == 1)
    [foldername, filename, ] = HomoModel.st80(path, range, radiusOfTube,...
        MatType, StartTime, EndTime, TimeStep);
    save Homo_UserArg.mat radiusOfTube range MatType path foldername...
        filename StartTime EndTime TimeStep;
elseif (MODEL == 2)
    [foldername, filename] = MixedModel.st80(path, range, radiusOfTube,...
        StartTime, EndTime, TimeStep);
    save Mixed_UserArg.mat radiusOfTube range path foldername filename...
        StartTime EndTime TimeStep;
```

```

elseif (MODEL == 3)
    [foldername, filename] = MixedModel.VesselAdded(path, range, radiusOfTube,...
        StartTime, EndTime, TimeStep);
    save MixedVessel.UserArg.mat radiusOfTube range path foldername filename...
        StartTime EndTime TimeStep;
else
    disp('Wrong input. Please enter MODEL type: Homo = 1; Mixed = 2; Mixed.Perfusion = 3');
    break;
end

```

## HomoModel\_st80.m

```

% COMSOL Simulation for Homogeneous Tissue
% Author: Sijia Huang
% Date: 03/18/2014

% This function generates the COMSOL data for a Homogeneous model.
% It is divided into parts:
% 1.Set Geometry: the default is a doughnut shape region with diameter of
%                 outside circle 80mm, and inside heat source 10mm.
% 2.Set selection: Select the region and boundary
% 3.Set Material: collagen, muscle and fat
% 4.Temperature Setting: the default is 65°C heat source with a 37°C surrounding temperature
% 5.Time Range: the default is 0:100:1000
% 6.Save data and model

function [foldername, filename] = HomoModel_st80(MyPath, range, radiusOfTube,...
    MatType, StartTime, EndTime, TimeStep)

% % *****user defined arg*****
% range = 80; % in mm
% radiusOfTube = 5;
% MatType = 1; % 1.collagen 2.muscle 3.fat (Default type is collagen)
%
% StartTime = 0;
% EndTime = 1000;
% TimeStep = 100;

% ***** START *****

import com.comsol.model.*
import com.comsol.model.util.*

model = ModelUtil.create('Model');

model.modelPath('C:\Users\Administrator\Desktop');

model.modelNode.create('mod1');

model.geom.create('geom1', 2);

model.mesh.create('mesh1', 'geom1');

model.physics.create('ht', 'BioHeat', 'geom1',...
    {'I1' 'I2' 'I3' 'I4' 'I5' 'I6' 'I7' 'I8' 'I9' 'I10' 'I11' 'I12'});

model.study.create('std1');
model.study('std1').feature.create('time', 'Transient');

```

```

% *****Set geometry*****

model.geom('geom1').run;
model.geom('geom1').lengthUnit('mm');
model.geom('geom1').feature.create('c1', 'Circle');
model.geom('geom1').feature('c1').set('pos', [0 0]);
model.geom('geom1').feature('c1').set('r', num2str(range/2));
model.geom('geom1').feature.create('c2', 'Circle');
model.geom('geom1').feature('c2').set('pos', [0 0]);
model.geom('geom1').feature('c2').set('r', num2str(radiusOfTube));
model.geom('geom1').runAll;

% Difference operator
model.geom('geom1').feature.create('dif1', 'Difference');
model.geom('geom1').feature('dif1').selection('input').set({'c1'});
model.geom('geom1').feature('dif1').selection('input2').set({'c2'});
model.geom('geom1').run;

% Finer mesh
model.mesh('mesh1').autoMeshSize(3);
model.mesh('mesh1').run;

% *****Set selection*****
% NOTE: The number do not following your self-def. COMSOL has its own rules.
model.selection.create('sel1');
model.selection('sel1').geom(1); % choose boundary
model.selection('sel1').set([3 4 6 7]);
model.selection('sel1').name('water tube');

% *****Set Material*****
model.material.create('mat1');
model.material.create('mat2');
model.material.create('mat3');
model.material('mat1').name('collegen');
model.material('mat2').name('muscle');
model.material('mat3').name('fat');

model.material('mat1').materialModel('def').set('thermalconductivity', '0.47[W/(m*K)]');
model.material('mat1').materialModel('def').set('density', '1.097[g/cm^3]');
model.material('mat1').materialModel('def').set('heatcapacity', '3.5[J/(g*K)]');

model.material('mat2').materialModel('def').set('thermalconductivity', '0.5[W/(m*K)]');
model.material('mat2').materialModel('def').set('density', '1.05[g/cm^3]');
model.material('mat2').materialModel('def').set('heatcapacity', '3.639[J/(g*K)]');

model.material('mat3').materialModel('def').set('thermalconductivity', '0.5[W/(m*K)]');
model.material('mat3').materialModel('def').set('density', '0.916[g/cm^3]');
model.material('mat3').materialModel('def').set('heatcapacity', '2.5[J/(g*K)]');

% 1.collagen 2.muscle 3.fat
if (MatType == 1)
    model.material('mat2').active(false);
    model.material('mat3').active(false);
    model.material('mat1').selection.all;
elseif (MatType == 2)
    model.material('mat1').active(false);
    model.material('mat3').active(false);
    model.material('mat2').selection.all;
elseif (MatType == 3)
    model.material('mat1').active(false);
    model.material('mat2').active(false);
    model.material('mat3').selection.all;
else

```

```

        model.material('mat2').active(false);
        model.material('mat3').active(false);
        model.material('mat1').selection.all;
    end

% *****Temperature Setting*****
model.physics('ht').feature('init1').set('T', 1, '310.15[K]');
model.physics('ht').feature.create('temp1', 'TemperatureBoundary', 1);
model.physics('ht').feature('temp1').name('water tube');
model.physics('ht').feature('temp1').selection.named('sel1');
model.physics('ht').feature('temp1').set('T0', 1, '338.15[K]');

% *****Time Range*****
model.study('std1').feature('time').set('tlist', strcat('range(', num2str(StartTime), ...
    ', ', num2str(TimeStep), ', ', num2str(EndTime), ')'));

% *****Create Study*****
model.sol.create('sol1');
model.sol('sol1').study('std1');
model.sol('sol1').feature.create('st1', 'StudyStep');
model.sol('sol1').feature('st1').set('study', 'std1');
model.sol('sol1').feature('st1').set('studystep', 'time');
model.sol('sol1').feature.create('v1', 'Variables');
model.sol('sol1').feature.create('t1', 'Time');
model.sol('sol1').feature('t1').set('tlist', strcat('range(', num2str(StartTime), ...
    ', ', num2str(TimeStep), ', ', num2str(EndTime), ')'));
model.sol('sol1').feature('t1').set('plot', 'off');
model.sol('sol1').feature('t1').set('plotfreq', 'tout');
model.sol('sol1').feature('t1').set('probesel', 'all');
model.sol('sol1').feature('t1').set('probes', {});
model.sol('sol1').feature('t1').set('probefreq', 'tsteps');
model.sol('sol1').feature('t1').set('estrat', 'exclude');
model.sol('sol1').feature('t1').set('maxorder', 2);
model.sol('sol1').feature('t1').set('control', 'time');
model.sol('sol1').feature('t1').feature.create('fc1', 'FullyCoupled');
model.sol('sol1').feature('t1').feature('fc1').set('jtech', 'once');
model.sol('sol1').feature('t1').feature('fc1').set('maxiter', 5);
model.sol('sol1').feature('t1').feature.create('d1', 'Direct');
model.sol('sol1').feature('t1').feature('d1').set('linsolver', 'pardiso');
model.sol('sol1').feature('t1').feature('fc1').set('linsolver', 'd1');
model.sol('sol1').feature('t1').feature('fc1').set('jtech', 'once');
model.sol('sol1').feature('t1').feature('fc1').set('maxiter', 5);
model.sol('sol1').feature('t1').feature.remove('fcDef');
model.sol('sol1').attach('std1');

model.result.create('pg1', 2);
model.result('pg1').set('data', 'dset1');
model.result('pg1').feature.create('surfl', 'Surface');
model.result('pg1').feature('surfl').set('expr', 'T');
model.result('pg1').feature('surfl').set('descr', 'Temperature');

model.sol('sol1').runAll;

model.result('pg1').set('windowtitle', 'Graphics');
model.result('pg1').run;
model.result.export.create('plot1', 'Plot');
model.result.export('plot1').set('struct', 'sectionwise');

% *****Save Data*****
% 1.collagen 2.muscle 3.fat
if(MatType == 1)
    foldername = 'stHomoCollagen';

```

```

        filename = 'HomoCollagen';
elseif(MatType == 2)
    foldername = 'stHomoMuscle';
    filename = 'HomoMuscle';
elseif(MatType == 3)
    foldername = 'stHomoFat';
    filename = 'HomoFat';
else
    foldername = 'stHomoCollagen';
    filename = 'HomoCollagen';
end

mkdir(MyPath, foldername); % generate a new folder

numberOfSlice = (EndTime - StartTime)/TimeStep + 1 ;

for i = 1:numberofSlice
    model.result('p1').run;
    model.result('p1').set('solnum', num2str(i));
    path = strcat(MyPath, '\', foldername, '\', filename, num2str((i-1)*TimeStep), '.txt');
    model.result.export('plot1').set('filename', path);
    model.result.export('plot1').run;
end

% *****Save Model*****
% 1.collagen 2.muscle 3.fat
if(MatType == 1)
    model.save('stHomoCollagen');
elseif(MatType == 2)
    model.save('stHomoMuscle');
elseif(MatType == 3)
    model.save('stHomoFat');
else
    model.save('stHomoCollagen');
end
end

```

## MixedModel\_st80.m

```

% COMSOL Simulation for Mixed Tissue
% Author: Sijia Huang
% Date: 03/18/2014

% This function generates the COMSOL data for a Mixed model.
% It is divided into parts:
% 1.Set Geometry: the default is a 3 by 3 square with region size 81mm,
%                 inside tube as heat source with diameter 10mm.
% 2.Set selection: Select the region and boundary
% 3.Set Material: collagen, muscle and fat
% 4.Temperature Setting: the default is 65°C heat source with a 37°C surrounding temperature
% 5.Time Range: the default is 0:600:3600
% 6.Save data and model

function [foldername, filename] = MixedModel_st80(MyPath, range, radiusOfTube,...
    StartTime, EndTime, TimeStep)

% % *****user defined arg*****
% range = 81; % in mm (the number should be divisible by 3)
% radiusOfTube = 5;
%
% StartTime = 0;
% EndTime = 3600;
% TimeStep = 600;

```



```

% ***** START *****

import com.comsol.model.*
import com.comsol.model.util.*

model = ModelUtil.create('Model');

model.modelPath('C:\Users\Administrator\Desktop');

model.modelNode.create('mod1');

model.geom.create('geom1', 2);

model.mesh.create('mesh1', 'geom1');

model.physics.create('ht', 'BioHeat', 'geom1',...
    {'I1' 'I2' 'I3' 'I4' 'I5' 'I6' 'I7' 'I8' 'I9' 'I10' 'I11' 'I12'});

model.study.create('std1');
model.study('std1').feature.create('time', 'Transient');

model.geom('geom1').run;
model.geom('geom1').lengthUnit('mm');

% *****Set geometry*****
unitSize = range/3;
count = 1;
for i = -1:1
    for j = -1:1
        model.geom('geom1').feature.create(strcat('sq',num2str(count)), 'Square');
        model.geom('geom1').feature(strcat('sq',num2str(count))).set('base', 'center');
        model.geom('geom1').feature(strcat('sq',num2str(count))).set('size', num2str(unitSize));
        model.geom('geom1').feature(strcat('sq',num2str(count))).set('pos', [i*27 j*27]);
        count = count +1 ;
    end
end

model.geom('geom1').feature.create('c1', 'Circle');
model.geom('geom1').feature('c1').set('pos', [0 0]);
model.geom('geom1').feature('c1').set('r', num2str(radiusOfTube));
model.geom('geom1').runAll;

% Difference operator
model.geom('geom1').feature.create('dif1', 'Difference');
model.geom('geom1').feature('dif1').selection('input').set...
    ({ 'sq1' 'sq2' 'sq3' 'sq4' 'sq5' 'sq6' 'sq7' 'sq8' 'sq9'});
model.geom('geom1').feature('dif1').selection('input2').set({'c1'});
model.geom('geom1').run;

% Finer mesh
model.mesh('mesh1').autoMeshSize(3);
model.mesh('mesh1').run;

% *****Set selection*****
% NOTE: The number do not following your self-def. COMSOL has its own rules.
model.selection.create('sel1');
model.selection('sel1').set([1 2 3 4]);
model.selection('sel1').name('collegen');
model.selection.create('sel2');
model.selection('sel2').set([6 8 9]);
model.selection('sel2').name('muscle');

```

```

model.selection.create('sel3');
model.selection('sel3').set([5 7]);
model.selection('sel3').name('fat');
model.selection.create('sel4');
model.selection('sel4').geom(1); % choose boundary
model.selection('sel4').set([25 26 27 28]);
model.selection('sel4').name('water tube');

% *****Set Material*****
model.material.create('mat1');
model.material.create('mat2');
model.material.create('mat3');
model.material('mat1').selection.named('sel1'); % sel1 - collagen
model.material('mat2').selection.named('sel2'); % sel2 - muscle
model.material('mat3').selection.named('sel3'); % sel3 - fat

model.material('mat1').materialModel('def').set('thermalconductivity', {'0.47[W/(m*K)]'});
model.material('mat1').materialModel('def').set('density', {'1.097[g/cm^3]'});
model.material('mat1').materialModel('def').set('heatcapacity', {'3.5[J/(g*K)]'});

model.material('mat2').materialModel('def').set('thermalconductivity', {'0.5[W/(m*K)]'});
model.material('mat2').materialModel('def').set('density', {'1.05[g/cm^3]'});
model.material('mat2').materialModel('def').set('heatcapacity', {'3.639[J/(g*K)]'});

model.material('mat3').materialModel('def').set('thermalconductivity', {'0.5[W/(m*K)]'});
model.material('mat3').materialModel('def').set('density', {'0.916[g/cm^3]'});
model.material('mat3').materialModel('def').set('heatcapacity', {'2.5[J/(g*K)]'});

% *****Tempreture Setting*****
model.physics('ht').feature('init1').set('T', 1, '310.15[K]'); % 37 o^C
model.physics('ht').feature.create('temp1', 'TemperatureBoundary', 1);
model.physics('ht').feature('temp1').name('water tube');
model.physics('ht').feature('temp1').selection.named('sel4');
model.physics('ht').feature('temp1').set('T0', 1, '338.15[K]'); % 65 o^C

% *****Time Range*****
model.study('std1').feature('time').set('tlist', strcat('range(', ...
    num2str(StartTime), ',', num2str(TimeStep), ',', num2str(EndTime), ')'));

% *****Create Study*****
model.sol.create('sol1');
model.sol('sol1').study('std1');
model.sol('sol1').feature.create('st1', 'StudyStep');
model.sol('sol1').feature('st1').set('study', 'std1');
model.sol('sol1').feature('st1').set('studystep', 'time');
model.sol('sol1').feature.create('v1', 'Variables');
model.sol('sol1').feature.create('t1', 'Time');
model.sol('sol1').feature('t1').set('tlist', strcat('range(', num2str(StartTime), ...
    ',', num2str(TimeStep), ',', num2str(EndTime), ')'));
model.sol('sol1').feature('t1').set('plot', 'off');
model.sol('sol1').feature('t1').set('plotfreq', 'tout');
model.sol('sol1').feature('t1').set('probesel', 'all');
model.sol('sol1').feature('t1').set('probes', {});
model.sol('sol1').feature('t1').set('probefreq', 'tsteps');
model.sol('sol1').feature('t1').set('estrat', 'exclude');
model.sol('sol1').feature('t1').set('maxorder', 2);
model.sol('sol1').feature('t1').set('control', 'time');
model.sol('sol1').feature('t1').feature.create('fc1', 'FullyCoupled');
model.sol('sol1').feature('t1').feature('fc1').set('jtech', 'once');
model.sol('sol1').feature('t1').feature('fc1').set('maxiter', 5);
model.sol('sol1').feature('t1').feature.create('d1', 'Direct');
model.sol('sol1').feature('t1').feature('d1').set('linsolver', 'pardiso');
model.sol('sol1').feature('t1').feature('fc1').set('linsolver', 'd1');

```

```

model.sol('sol1').feature('t1').feature('fc1').set('jtech', 'once');
model.sol('sol1').feature('t1').feature('fc1').set('maxiter', 5);
model.sol('sol1').feature('t1').feature.remove('fcDef');
model.sol('sol1').attach('std1');

model.result.create('pg1', 2);
model.result('pg1').set('data', 'dset1');
model.result('pg1').feature.create('surf1', 'Surface');
model.result('pg1').feature('surf1').set('expr', 'T');
model.result('pg1').feature('surf1').set('descr', 'Temperature');

model.sol('sol1').runAll;
model.result('pg1').set('windowtitle', 'Graphics');
model.result('pg1').run;
model.result.export.create('plot1', 'Plot');
model.result.export('plot1').set('struct', 'sectionwise');

% *****Save Data*****
foldername = 'Mixed_st80';
filename = 'Mixed';

mkdir(MyPath, foldername); % generate a new folder

numberOfSlice = (EndTime - StartTime)/TimeStep + 1 ;

for i = 1:numberofSlice
    model.result('pg1').run;
    model.result('pg1').set('solnum', num2str(i));
    path = strcat(MyPath, '\', foldername, '\', filename, num2str((i-1)*TimeStep), '.txt');
    model.result.export('plot1').set('filename', path);
    model.result.export('plot1').run;
end

% *****Save Model*****
model.save('MixedTissue');

```

## MixedModel\_VesselAdded.m

```

% COMSOL Simulation for Mixed Tissue with Perfusion Added
% Author: Sijia Huang
% Date: 04/20/2014

% This function generates the COMSOL data for a Mixed model.
% It is divided into parts:
% 1.Set Geometry: the default is a 3 by 3 square with region size 81mm,
%                 inside tube as heat source with diameter 10mm.
%                 Random vessel with different size have been added based
%                 on related documents
% 2.Set selection: Select the region and boundary
% 3.Set Material: collagen, muscle and fat
% 4.Temperature Setting: the default is 65°C heat source with a 37°C surrounding temperature
%                       the vessel temp is also 37°C
% 5.Time Range: the default is 0:600:3600
% 6.Save data and model

function [foldername, filename] = MixedModel_VesselAdded(MyPath, range,...
    radiusOfTube, StartTime, EndTime, TimeStep)

% % *****user defined arg*****
% range = 81; % in mm (the number should be divisible by 3)
% radiusOfTube = 5;

```

```

%
% StartTime = 0;
% EndTime = 3600;
% TimeStep = 600;

% ***** START *****
import com.comsol.model.*
import com.comsol.model.util.*

model = ModelUtil.create('Model');

model.modelPath('C:\Users\Administrator\Desktop');

model.modelNode.create('mod1');

model.geom.create('geom1', 2);

model.mesh.create('mesh1', 'geom1');

model.physics.create('ht', 'BioHeat', 'geom1',...
    {'I1' 'I2' 'I3' 'I4' 'I5' 'I6' 'I7' 'I8' 'I9' 'I10' 'I11' 'I12'});

model.study.create('std1');
model.study('std1').feature.create('time', 'Transient');

model.geom('geom1').run;
model.geom('geom1').lengthUnit('mm');

% *****Set geometry*****

unitSize = range/3;
count = 1;
for i = -1:1
    for j = -1:1
        model.geom('geom1').feature.create(strcat('sq',num2str(count)), 'Square');
        model.geom('geom1').feature(strcat('sq',num2str(count))).set('base', 'center');
        model.geom('geom1').feature(strcat('sq',num2str(count))).set('size', num2str(unitSize));
        model.geom('geom1').feature(strcat('sq',num2str(count))).set('pos', [i*unitSize j*unitSize]);
        count = count + 1 ;
    end
end

% water tube
model.geom('geom1').feature.create('c1', 'Circle');
model.geom('geom1').feature('c1').set('pos', [0 0]);
model.geom('geom1').feature('c1').set('r', num2str(radiusOfTube));

% Set the Vessel
model.geom('geom1').feature.create('c2', 'Circle');
model.geom('geom1').feature('c2').set('r', '3');
model.geom('geom1').feature('c2').setIndex('pos', '-18', 0);
model.geom('geom1').feature('c2').setIndex('pos', '-18', 1);

model.geom('geom1').feature.create('el', 'Ellipse');
model.geom('geom1').feature('el').setIndex('pos', '-14', 0);
model.geom('geom1').feature('el').setIndex('pos', '-16', 1);
model.geom('geom1').feature('el').setIndex('semiaxes', '0.7', 0);
model.geom('geom1').feature('el').setIndex('semiaxes', '3.5', 1);
model.geom('geom1').feature('el').set('rot', '35');

model.geom('geom1').feature.create('c3', 'Circle');
model.geom('geom1').feature('c3').setIndex('pos', '-13', 0);
model.geom('geom1').feature('c3').setIndex('pos', '-21', 1);

```

```

model.geom('geom1').feature('c3').set('r', '1');

model.geom('geom1').feature.create('c4', 'Circle');
model.geom('geom1').feature('c4').setIndex('pos', '-21', 0);
model.geom('geom1').feature('c4').setIndex('pos', '-22', 1);
model.geom('geom1').feature('c4').set('r', '0.5');

model.geom('geom1').feature.create('c5', 'Circle');
model.geom('geom1').feature('c5').set('r', '3');
model.geom('geom1').feature('c5').setIndex('pos', '25', 0);
model.geom('geom1').feature('c5').setIndex('pos', '15', 1);

model.geom('geom1').feature.create('c6', 'Circle');
model.geom('geom1').feature('c6').set('r', '1.5');
model.geom('geom1').feature('c6').setIndex('pos', '-28', 0);
model.geom('geom1').feature('c6').setIndex('pos', '25', 1);

% Difference operator
model.geom('geom1').feature.create('dif1', 'Difference');
model.geom('geom1').feature('dif1').selection('input').set...
    ({'sq1' 'sq2' 'sq3' 'sq4' 'sq5' 'sq6' 'sq7' 'sq8' 'sq9'});
model.geom('geom1').feature('dif1').selection('input2').set...
%    ({'c1' 'c2' 'c3' 'c4' 'c5' 'e1'});
model.geom('geom1').feature('dif1').selection('input2').set...
    ({'c1' 'c2' 'c3' 'c4' 'c5' 'c6' 'e1'});

model.geom('geom1').runAll;
model.geom('geom1').run;

% Finer mesh
model.mesh('mesh1').autoMeshSize(3);
model.mesh('mesh1').run;

% *****Set selection*****
% NOTE: The number do not following your self-def. COMSOL has its own rules.
model.selection.create('sel1');
model.selection('sel1').set([1 2 3 4 9 10]);
model.selection('sel1').name('collegen');
model.selection.create('sel2');
model.selection('sel2').set([5 7]);
model.selection('sel2').name('muscle');
model.selection.create('sel3');
model.selection('sel3').set([6 8]);
model.selection('sel3').name('fat');
% model.selection.create('sel4');
% model.selection('sel4').geom(1);
% model.selection('sel4').set([51 52 53 54]);
% model.selection('sel4').name('water tube');
% model.selection.create('sel5');
% model.selection('sel5').geom(1);
% model.selection('sel5').set...
%    ([29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 48 49 50 55 56 57 58 59 60]);
% model.selection('sel5').name('vessel');
model.selection.create('sel4');
model.selection('sel4').geom(1);
model.selection('sel4').set([55 56 57 58]);
model.selection('sel4').name('water tube');
model.selection.create('sel5');
model.selection('sel5').geom(1);
model.selection('sel5').set...
    ([29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 52 53 54 59 60 61 62 63 64]);
model.selection('sel5').name('vessel');

% *****Set Material*****

```

```

model.material.create('mat1');
model.material.create('mat2');
model.material.create('mat3');
model.material('mat1').selection.named('sel1'); % sel1 - collagen
model.material('mat2').selection.named('sel2'); % sel2 - muscle
model.material('mat3').selection.named('sel3'); % sel3 - fat

model.material('mat1').materialModel('def').set('thermalconductivity', '0.47[W/(m*K)]');
model.material('mat1').materialModel('def').set('density', '1.097[g/cm^3]');
model.material('mat1').materialModel('def').set('heatcapacity', '3.5[J/(g*K)]');

model.material('mat2').materialModel('def').set('thermalconductivity', '0.5[W/(m*K)]');
model.material('mat2').materialModel('def').set('density', '1.05[g/cm^3]');
model.material('mat2').materialModel('def').set('heatcapacity', '3.639[J/(g*K)]');

model.material('mat3').materialModel('def').set('thermalconductivity', '0.5[W/(m*K)]');
model.material('mat3').materialModel('def').set('density', '0.916[g/cm^3]');
model.material('mat3').materialModel('def').set('heatcapacity', '2.5[J/(g*K)]');

% *****Tempreture Setting*****
model.physics('ht').feature('init1').set('T', 1, '310.15[K]'); % 37 o°C

model.physics('ht').feature.create('temp1', 'TemperatureBoundary', 1);
model.physics('ht').feature('temp1').name('water tube');
model.physics('ht').feature('temp1').selection.named('sel4');
model.physics('ht').feature('temp1').set('T0', 1, '338.15[K]'); % 65 o°C

model.physics('ht').feature.create('temp2', 'TemperatureBoundary', 1);
model.physics('ht').feature('temp2').name('vessel boundary');
model.physics('ht').feature('temp2').selection.named('sel5');
model.physics('ht').feature('temp2').set('T0', 1, '310.15[K]');

% *****Time Range*****
model.study('std1').feature('time').set('tlist', strcat('range(', ...
    num2str(StartTime), ',', num2str(TimeStep), ',', num2str(EndTime), ')'));

% *****Create Study*****
model.sol.create('sol1');
model.sol('sol1').study('std1');
model.sol('sol1').feature.create('st1', 'StudyStep');
model.sol('sol1').feature('st1').set('study', 'std1');
model.sol('sol1').feature('st1').set('studystep', 'time');
model.sol('sol1').feature.create('v1', 'Variables');
model.sol('sol1').feature.create('t1', 'Time');
model.sol('sol1').feature('t1').set('tlist', strcat('range(', num2str(StartTime), ...
    ',', num2str(TimeStep), ',', num2str(EndTime), ')'));
model.sol('sol1').feature('t1').set('plot', 'off');
model.sol('sol1').feature('t1').set('plotfreq', 'tout');
model.sol('sol1').feature('t1').set('probesel', 'all');
model.sol('sol1').feature('t1').set('probes', {});
model.sol('sol1').feature('t1').set('probefreq', 'tsteps');
model.sol('sol1').feature('t1').set('estrat', 'exclude');
model.sol('sol1').feature('t1').set('maxorder', 2);
model.sol('sol1').feature('t1').set('control', 'time');
model.sol('sol1').feature('t1').feature.create('fc1', 'FullyCoupled');
model.sol('sol1').feature('t1').feature('fc1').set('jtech', 'once');
model.sol('sol1').feature('t1').feature('fc1').set('maxiter', 5);
model.sol('sol1').feature('t1').feature.create('d1', 'Direct');
model.sol('sol1').feature('t1').feature('d1').set('linsolver', 'pardiso');
model.sol('sol1').feature('t1').feature('fc1').set('linsolver', 'd1');
model.sol('sol1').feature('t1').feature('fc1').set('jtech', 'once');
model.sol('sol1').feature('t1').feature('fc1').set('maxiter', 5);
model.sol('sol1').feature('t1').feature.remove('fcDef');
model.sol('sol1').attach('std1');

```

```

model.result.create('pg1', 2);
model.result('pg1').set('data', 'dset1');
model.result('pg1').feature.create('surfl', 'Surface');
model.result('pg1').feature('surfl').set('expr', 'T');
model.result('pg1').feature('surfl').set('descr', 'Temperature');

model.sol('sol1').runAll;

model.result('pg1').set('windowtitle', 'Graphics');
model.result('pg1').run;
model.result.export.create('plot1', 'Plot');
model.result.export('plot1').set('struct', 'sectionwise');

% *****Save Data*****

foldername = 'Mixed-st80-Vessel';
filename = 'MixedV';

mkdir(MyPath, foldername); % generate a new folder

numberOfSlice = (EndTime - StartTime)/TimeStep + 1 ;

for i = 1:numberofSlice
    model.result('pg1').run;
    model.result('pg1').set('solnum', num2str(i));
    path = strcat(MyPath, '\', foldername, '\', filename, num2str((i-1)*TimeStep), '.txt');
    model.result.export('plot1').set('filename', path);
    model.result.export('plot1').run;
end

% *****Save Model*****
model.save('MixedTissue-Vessel');

```

## Matlab\_ImageProcessing.m

```

% Author: Sijia Huang
% Date: 07/09/2014

% MATLAB_IMAGEPROCESSING : This is the main file of post-processing using
% Matlab. The data file generated by COMSOL will be read by Matlab and
% transfer them into Matlab images.

clear; clc; close all;

%% ***** User defined arg for MATLAB *****
MODEL = input('Please input the type of your model(Homo = 1; Mixed = 2; Mixed.Perfusion = 3):
');

if (MODEL == 1)
    % For MODEL == Homo
    s1 = 1; s2 = 4; s3 = 7; s4 = 11; % Select the slice number you want to see
    title = ' Homo tissue'; % the title showed in Matlab image
    nameSave = 'Homo'; % the name for .eps files
else if (MODEL == 2)
    % For MODEL == Mixed
    s1 = 1; s2 = 2; s3 = 3; s4 = 7; % Select the slice number you want to see
    title = ' Mixed tissue'; % the title showed in Matlab image
    nameSave = 'Mixed';

```



```

elseif (MODEL == 3)
    % For MODEL == Mixed.Perfusion
    s1 = 4; s2 = 6; s3 = 7; s4 = 9; % Select the slice number you want to see
    title = ' Mixed tissue with Perfusion'; % the title showed in Matlab image
    nameSave = 'Mixed.Vessel';

else
    disp('Wrong input. Please enter MODEL type: Homo = 1; Mixed = 2; Mixed.Perfusion = 3');
    break;
end

%% Matlab Processsing
isCircle = 0;

if (MODEL == 1)
    load Homo.UserArg.mat;
    isCircle = 1;
    numberOfSlice = (EndTime - StartTime)/TimeStep + 1 ;
    path = pwd;
    pd = readComsolData (strcat(path,'/',foldername,'/'), filename, numberOfSlice, TimeStep);
    GenTempImage (pd, numberOfSlice, isCircle, range, radiusOfTube, MODEL, TimeStep, MatType);
    plotSlice (s1, s2, s3, s4, title, nameSave, numberOfSlice, TimeStep, MODEL, MatType);
elseif (MODEL == 2)
    load Mixed.UserArg.mat;
    numberOfSlice = (EndTime - StartTime)/TimeStep + 1 ;
    path = pwd;
    pd = readComsolData (strcat(path,'/',foldername,'/'), filename, numberOfSlice, TimeStep);
    GenTempImage (pd, numberOfSlice, isCircle, range, radiusOfTube, MODEL, TimeStep);
    plotSlice (s1, s2, s3, s4, title, nameSave, numberOfSlice, TimeStep, MODEL);
elseif (MODEL == 3)
    load MixedVessel.UserArg.mat;
    numberOfSlice = (EndTime - StartTime)/TimeStep + 1 ;
    path = pwd;
    pd = readComsolData (strcat(path,'/',foldername,'/'), filename, numberOfSlice, TimeStep);
    GenTempImage (pd, numberOfSlice, isCircle, range, radiusOfTube, MODEL, TimeStep);
    plotSlice (s1, s2, s3, s4, title, nameSave, numberOfSlice, TimeStep, MODEL);
else
    disp('Wrong input. Please enter MODEL type: Homo = 1; Mixed = 2; Mixed.Perfusion = 3');
    break;
end

```

## readComsolData.m

```

% Author: Sijia Huang
% Date: 3/25/2014

% NOTE: The function is just an improved(atomated read) version for "udont2.m"
% in "comsol.db".

% This function is used for read the corresponding data from the extracted
% COMSOL data. Since the output files' format from COMSOL are the same, the
% desired start line for every data cluster are fixed.
% (pd.p = Coordinates, pd.t = Elements(triangles) and pd.d = Data)
% The user need to open one of the file and define the required arguments,
% then the program will run and generate pd structure for future use.

function pd = readComsolData (path, file, numberOfSlice,TimeStep)

clear pd;

%% User defined parameters
% path = '/Users/sijiahuang/Documents/MATLAB/st80mm.Mixed/';

```

```

% % path = '/Users/sijiahuang/Documents/MATLAB/st80mm.Homo/stHomoMuscle/';
% file = 'Mixed';

% Create string matrix hdr containing text file lines
% Search for s1, s2 & s3
s1='Nodes'; s2='Elements'; s3='Coord'; ln=0;
filename = strcat(file, '0.txt');
fid = fopen(strcat(path, filename), 'r');
while ln<19;
    %while feof(fid)==0
    hdr=fgetl(fid); ln=ln+1;
    tf=strcmp(hdr(3:7),s1); if tf==1; nn=str2num(hdr(end-4:end)); end;
    tf=strcmp(hdr(3:10),s2); if tf==1; ne=str2num(hdr(end-4:end)); end;
    tf=strcmp(hdr(3:7),s3); if tf==1; lnc=ln; end;
end;
fclose(fid);
CoordStartLine = lnc;
TriStartLine = lnc+nn+1;
dataStartLine = lnc+nn+ne+2;

%% Strat processing the data
pd.p = [];
filename = strcat(file, '0.txt');
fid = fopen(strcat(path, filename), 'rt');
CoordCell= textscan(fid, '', 'headerlines', CoordStartLine);
pd.p = [CoordCell{1} CoordCell{2}]';
fclose(fid);

pd.t = [];
filename = strcat(file, '0.txt');
fid = fopen(strcat(path, filename), 'rt');
TriCell = textscan(fid, '', 'headerlines', TriStartLine);
pd.t = [TriCell{1} TriCell{2} TriCell{3}]';
fclose(fid);

pd.d = [];
for i = 0:numberofSlice-1
    filename = strcat(file, num2str(i*TimeStep), '.txt');
    fid = fopen( strcat(path, filename), 'rt');
    dataCell = textscan(fid, '', 'headerlines', dataStartLine);
    data = dataCell{1};
    pd.d =[pd.d data];
    fclose(fid);
end
pd.d = pd.d';

end

```

## GenTempImage.m

```

% Author(Modified by): Sijia Huang
% Date: 3/31/2014

% GenTempImage : Using the COMSOL returned temperature data to generate 2D
% temperature maps which can be used to simulate B-mode images. COMSOL
% returns temperature values at distributed points. Using griddata function
% the temperature values at the known points can be extended to a grid of
% points.

function GenTempImage (pd, numberOfSlice, isCircle, range, radiusOfTube, MODEL, TimeStep, MatType)

```

```

pts = pd.p;
x = pts(1,:); % x-coordinates
y = pts(2,:); % y-coordinates
ut = pd.d;

% set scale
step1 = 0.1;
step2 = 0.1;
t1 = -40:step1:40;
t2 = -40:step2:40;
[xi,yi] = meshgrid(t2,t1);

for k=1:numberofSlice
    z = ut(k,:);
    % Transform to o^C form. K = o^C + 273.15
    UT(k, :, :) = griddata(x,y,z,xi,yi,'linear')- 273.15;
end

if (isCircle == 1)
    center = [size(UT,2)/2 size(UT,3)/2];
    for k=1:size(UT,2)
        for l=1:size(UT,3)
            r = sqrt(((k-1-center(1)) * step1)^2 + ((l-1-center(2)) * step2)^2);
            % outer environment
            if(r > range/2)
                UT(:,k,l)=36;
            end
            % water tube
            if( r < radiusOfTube)
                UT(:,k,l)=65;
            end
        end
    end
end
end

if (MODEL == 1)
    if (MatType == 1)
        save st80_HomoCol.mat UT xi yi numberOfSlice TimeStep MODEL MatType;
    elseif (MatType == 2)
        save st80_HomoMuscle.mat UT xi yi numberOfSlice TimeStep MODEL MatType;
    elseif (MatType == 3)
        save st80_HomoFat.mat UT xi yi numberOfSlice TimeStep MODEL MatType;
    end
elseif (MODEL == 2)
    save st80_Mixed.mat UT xi yi numberOfSlice TimeStep MODEL;
elseif (MODEL == 3)
    save st80_Mixed.Vessel.mat UT xi yi numberOfSlice TimeStep MODEL;
end

```

## plotSlice.m

```

% Author(Modified by): Sijia Huang
% Date: 3/31/2014

% plotSlice : Plot the selected slices based on user's choice.
% The silce number is based on the time range set by user.
% e.g. if the time range is 0:600:3600, you have 7 silces in total, silce=1
% will be 0s, silce=5 will be 2400s, etc.

```

```

function plotSlice (s1, s2, s3, s4, name, nameSave, numberOfSlice, TimeStep, MODEL, MatType)

close all;

% Loading the stored .mat data
if (MODEL == 1)
    if (MatType == 1)
        load st80_HomoCol;
    elseif (MatType == 2)
        load st80_HomoMuscle;
    elseif (MatType == 3)
        load st80_HomoFat;
    end
elseif (MODEL == 2)
    load st80_Mixed;
elseif (MODEL == 3)
    load st80_Mixed.Vessel;
end

% User defined arguments
% s1 = 1; s2 = 2; s3 = 3; s4 = 7;
% name = 'Mixed tissue with Perfusion';
% nameSave = 'Mixed.Vessel';
% TimeStep = 600;

% Extract specific slices
slice1 = UT(s1, :, :);
slice2 = UT(s2, :, :);
slice3 = UT(s3, :, :);
slice4 = UT(s4, :, :);

[a,b,c] = size(UT);
slice1 = reshape(slice1,b,c);
slice2 = reshape(slice2,b,c);
slice3 = reshape(slice3,b,c);
slice4 = reshape(slice4,b,c);

% Since the xy-coordinate in COMSOL is different from MATLAB, we need to
% transform the image.
slice1 = slice1(end:-1:1,:);
slice2 = slice2(end:-1:1,:);
slice3 = slice3(end:-1:1,:);
slice4 = slice4(end:-1:1,:);

% load colormap file
concolor;

% plot
figure(1);
set(gca,'FontSize',12);
imagesc(slice1); colormap(conclr); colorbar('EastOutside');
axis image
set(gca,'XTick',0:200:800);
set(gca,'XTickLabel',{'0','20','40','60','80'});
xlabel('mm');
set(gca,'YTick',0:200:800);
set(gca,'YTickLabel',{'0','20','40','60','80'});
ylabel('mm');
title(strcat('COMSOL Simulated Temp for',name, ': ',num2str((s1-1)* TimeStep), 's'));
saveas(gcf, strcat(nameSave, num2str((s1-1)* TimeStep), '.eps'), 'psc2');

figure(2);
set(gca,'FontSize',12);
imagesc(slice2); colormap(conclr); colorbar('EastOutside');

```

```

axis image
set(gca,'XTick',0:200:800);
set(gca,'XTickLabel',{'0','20','40','60','80'});
xlabel('mm');
set(gca,'YTick',0:200:800);
set(gca,'YTickLabel',{'0','20','40','60','80'});
ylabel('mm');
title(strcat('COMSOL Simulated Temp for',name, ': ', num2str((s2-1)* TimeStep), 's'));
saveas(gcf, strcat(nameSave, num2str((s2-1)* TimeStep), '.eps'), 'psc2');

figure(3);
set(gca,'FontSize',12);
imagesc(slice3); colormap(conclr); colorbar('EastOutside');
axis image
set(gca,'XTick',0:200:800);
set(gca,'XTickLabel',{'0','20','40','60','80'});
xlabel('mm');
set(gca,'YTick',0:200:800);
set(gca,'YTickLabel',{'0','20','40','60','80'});
ylabel('mm');
title(strcat('COMSOL Simulated Temp for', name, ': ', num2str((s3-1)* TimeStep), 's'));
saveas(gcf, strcat(nameSave, num2str((s3-1)* TimeStep), '.eps'), 'psc2');

figure(4);
set(gca,'FontSize',12);
imagesc(slice4); colormap(conclr); colorbar('EastOutside');
axis image
set(gca,'XTick',0:200:800);
set(gca,'XTickLabel',{'0','20','40','60','80'});
xlabel('mm');
set(gca,'YTick',0:200:800);
set(gca,'YTickLabel',{'0','20','40','60','80'});
ylabel('mm');
title(strcat('COMSOL Simulated Temp for', name, ': ', num2str((s4-1)* TimeStep), 's'));
saveas(gcf, strcat(nameSave, num2str((s4-1)* TimeStep), '.eps'), 'psc2');

clear F;
F(7) = struct('cdata',[],'colormap',conclr);
set(gca,'NextPlot','replaceChildren');

for i = 1:numberofSlice
    slice = UT(i, :, :);
    slice = reshape(slice,b,c);
    slice = rot90(slice);
    imagesc(slice); colormap(conclr); colorbar('EastOutside');
    F(i) = getframe;
end
movie(F,-1,0.5);

if (MODEL == 1)
    if (MatType == 1)
        movie2avi(F, 'stHomo.Col.avi');
    elseif (MatType == 2)
        movie2avi(F, 'stHomo.Muscle.avi');
    elseif (MatType == 3)
        movie2avi(F, 'stHomo.Fat.avi');
    end
elseif (MODEL == 2)
    movie2avi(F, 'stMixed.avi');
elseif (MODEL == 3)
    movie2avi(F, 'stMixed.Vessel.avi');
end

```