RMIT University
Thermomechanical modelling and analysis of soft tissue

Capstone Project Final Report

Supervisor: Yongmin Zhong
Student: Fuhan Sun
Student number: s3653824

Executive Summary

Modern clinical treatment and drugs, such as cryosurgery, cryopreservation, cancer hyper-therapy and diagnosis of febrile diseases, need to understand the thermal life phenomenon and temperature behavior in life tissue. Therefore, the study of biological heat transfer in human body has been a hot topic, which is useful for the design of clinical heat treatment equipment, accurate skin temperature prediction and more effective establishment of thermal protection for various purposes. The main research direction of this project is to predict the skin temperature, thinking about how to link the bio-heat transfer equation with the three-dimensional mathematical model to accurately simulate the temperature changes of human skin.

This report introduces the statement of problem, background and literature review, methodology and engineering design, findings and discussions as well as conclusion.

The aim of this project is to model the temperature changes of human skin. And the fundamental concept of the project is a heat-transfer equation called The Pennes Bioheat Transfer Equation. The solution to this equation is to solve the problem discretely and build a mathematical model. The innovation of this project is that the mathematical model is a three-dimensional model, which can simulate the temperature distribution of human skin more intuitively and clearly.

The Pennes bioheat transfer equation is solved by the finite difference method, and through literature review, the variables can be defined and numerically obtained. Using the C++ programming language and Visual Studio programming software, a skin temperature distribution model has been gained and presented in this report. However, this project has not achieved the next step of researching skin stress deformation to predict thermal damage, that is, the study of the temperature limit that human skin can withstand in a safe range, which will be the focus of work in the future.

Key words: The Pennes Bioheat Transfer Equation, temperature distribution, three-dimension bioheat model.

Content

Statement of problem

Thermo-medicine plays an important role in the treatment of human cancer, like thermal ablation, which is a minimally invasive treatment that uses extreme temperatures to treat tumors. However, during the treatment, it is difficult for the surgeon to perform the operation with the naked eye. The surgeon cannot confirm whether the temperature applied to the tumor is sufficient and whether the temperature of the tissue surrounding the tumor is too high or not.

In this situation, modeling and analysis of the treatment process can overcome this difficulty. And establishing mathematical simulation models is conducive to surgeons for preoperative simulation, intraoperative guidance and adjustment of the appropriate treatment temperature which can not only ablate the tumor but also ensure that the healthy tissue is not affected by high temperature. In a word, simulation model can greatly improve the success rate of surgery and treatment accuracy. The following figure shows an example of hyperthermia for liver tumors.

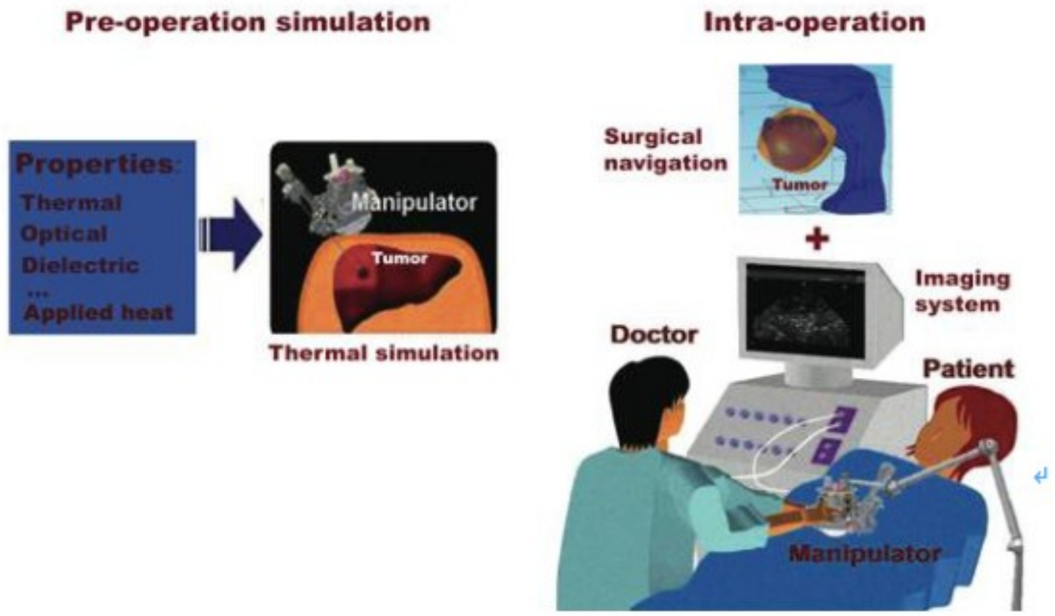


Figure1 Pre-operation simulation and intra-operation for thermal ablation [16]

As mentioned above, the focus of this project is to simulate the temperature distribution of human tissue to help the effective operation of thermal therapy. This project selected human skin which is applied external heat source as the research organization and established a model to simulate the temperature change process of the skin affected by the heat source.

In order to confirm the procedure of temperature changes, we introduce the Pennes Bioheat Transfer Equation to make the skin temperature distribution as a mathematic project. The equation is shown that:

$$\rho C \frac{\partial T}{\partial t} = k\frac{\partial^2 T}{\partial x^2} + k\frac{\partial^2 T}{\partial y^2} + k\frac{\partial^2 T}{\partial z^2} + w_b C_b \left(T_a - T\right) + Q_m + Q_r\left(x, y, z, t\right),$$

Figure 2 The Pennes Bioheat Transfer Equation Example [7]

The bioheat equation can be solved by finite difference method which is numerical methods for heat transfer problems. The values of parameters can be defined by literature review. Using C++ to code the equation with finite difference method can simulate the three-dimensional temperature distribution model, and the difference of temperature values can be expressed by different colors.

Background and literature review

In this project, we focus on the temperature distribution model of the soft tissue which can help surgeons gain the more realistic view about the change of temperature for the heated tissue. The fundamental concept is The Pennes Bioheat equation which the temperature field is obtained from. And Solving equations with mathematical modeling is a difficult part of this project.
The Pennes Bioheat Transfer Equation

The Pennes Bioheat Transfer Equation

There are many treatment processes in biomedicine that require temperature prediction, such as hyperthermia and thermal diagnosis. And the Pennes equation is the most widely used in temperature prediction. This equation can establish an effective mathematical model with the assumption that model is uniform and isotropic [1]. Although it is an approximation equation, it is still greatly used in mathematical modeling due to its simple calculation and high accuracy [2]. Deriving the one- dimensional bioheat equation can obtain instantaneous temperatures in multiple layers which each representing independent biological tissue [3]. Using the differential transformation method and the finite difference method to solve the bioheat equation, the temperature distribution field of the three-layer skin structure can be obtained [4]. Combined with mechanical analysis, the equation can effectively predict the thermal- induced mechanical behavior of soft tissue, which can improve the accuracy of heat transfer during thermal ablation [5]. In magnetic fluid hyperthermia (MFH), in order to ensure that high temperatures are confined to tumors rather than surrounding healthy tissue, it is necessary to predict the temperature profile that can be achieved during treatment, and a large part of the work of applying heat transfer in living tissue involves Pennes equation [1]. The purpose of high temperatures in cancer treatment is to raise the temperature of the cancerous tissue above the therapeutic value while maintaining the surrounding normal tissue at a sublethal temperature value [6]. Using the finite element method to solve the one-dimensional bioheat equation to accurately estimate the tissue temperature has always been a very important issue in the tumor hyperthermia treatment plan [6].

The three-dimension Pennes equation is in Figure3 is given by [7]:

$$\rho C\frac{\partial T}{\partial t} = k\frac{\partial^2 T}{\partial x^2} + k\frac{\partial^2 T}{\partial y^2} + k\frac{\partial^2 T}{\partial z^2} + w_b C_b\left(T_a - T\right) + Q_m + Q_r\left(x, y, z, t\right),$$

Figure 3 The Pennes Bioheat Transfer Equation Example [7]

From Figure 3, it can be seen that the Pennes bioheat equation has variable parameters. Refer to Shen, Zhang and Yang, the definitions of parameters in bioheat equation have shown below [7].

Parameter	Definition	Parameter	Definition
T	Temperature °C	C_b	The blood specific heat $J/(kg°C)$
ρ	The tissue density kg/m^3	T_a	The arterial temperature °C
C	The tissue specific heat $J/(kg°C)$	Q_m	The metabolic heat generation rate W/m^3
k	The tissue thermal conductivity $w/(m°C)$	Q_r	The regional heat sources W/m^3
w_b	The blood perfusion rate $kg/(m^3s)$		

Figure 4 The definitions of parameters in bioheat equation [7]

Method to solve Pennes equation-Finite difference method

Finite difference method is a discretization method which is useful for numerical solutions of partial differential equations [8]. Usually the domain of the equation would be divided into a uniform grid, which means that finite difference method typically produces a set of discrete numerical approximations of the derivatives in a "time step" manner [8]. For one-dimension Pennes bioheat equation, finite difference method has a wide range of use. The standard seven-point central difference scheme can discretize the bioheat equation [7]. Furthermore, the finite difference scheme of the one-dimensional biothermal transfer equation is constructed by using the second- order central difference scheme in space and the Crank-Nicholson format in time [9]. Generalized finite difference scheme which is a vertex-centered variant of the finite volume method, is used to simulate the irregular model of wave front propagation in cardiac tissue [10]. The interval finite difference method is used to construct the stage, and the interval numbers are used to numerically analyze the heat transfer process in the non-uniform biological tissue domain [18].

Model for solving Pennes equation-bioheat transfer model

A biothermal model is a model that exhibits a distribution of temperature over time and space. This allows surgeon to predict the temperature before treatment and visually observe the distribution of temperature in human tissue. For the study of biothermal heat transfer, bioheat transfer models are widely used.

Gowrishankar, Stewart and Martin et al. used a two-dimensional model built from the image to illustrate the spatial temperature distribution [11]. The model is derived from a section of skin tissue that is altered by the thermal insulation of the epidermis to change the temperature of the skin tissue [11].

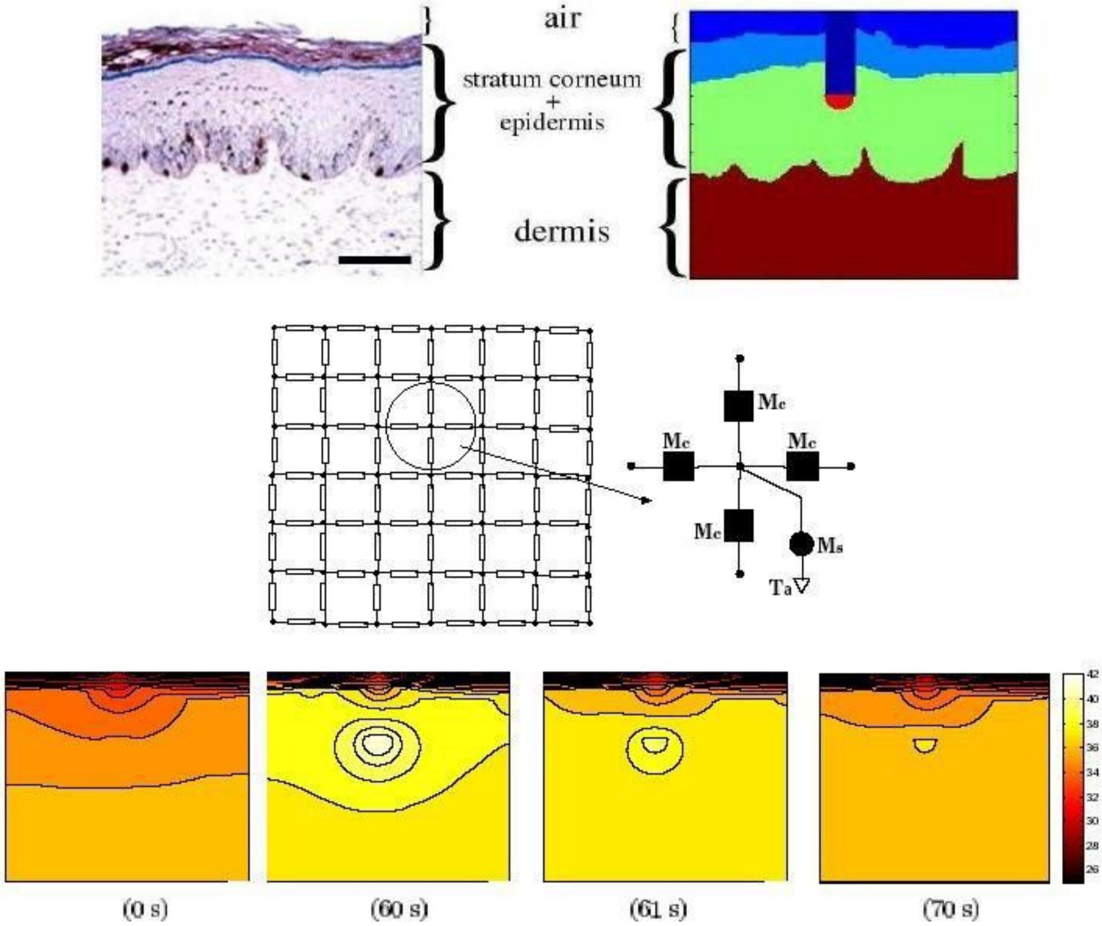


Figure 5 The temperature distribution model [11]

Brain nerves are very sensitive to temperature changes, so temperature control is extremely important when using chronic brain stimulation (DBS) to treat mental illness [12]. Elwassif, Kong and Vazquez et al. used the finite element method to establish a two-dimensional heat conduction model to study the temperature changes caused by DBS [12].

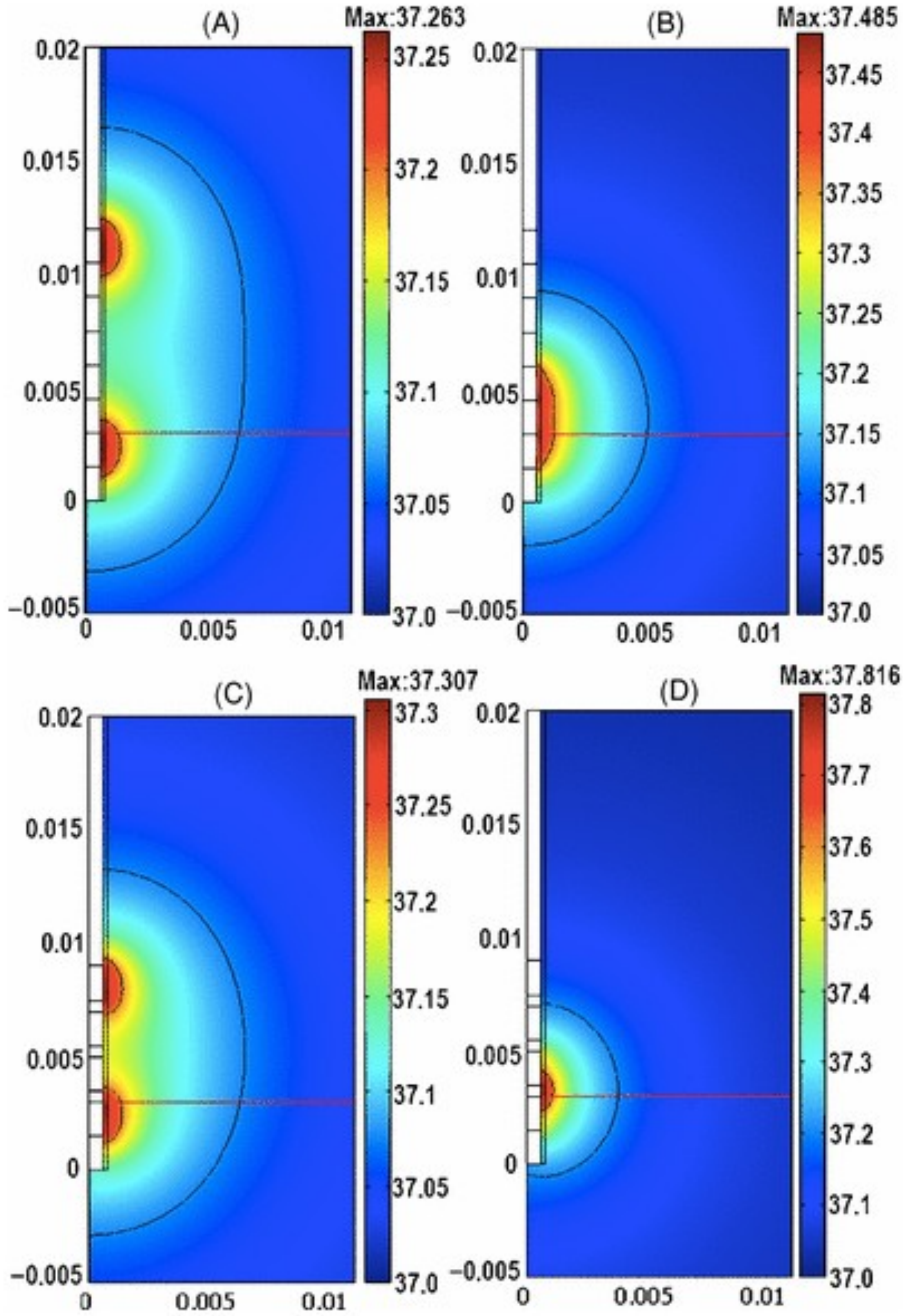


Figure 6 The temperature changes caused by DBS [12]

Ng and Ooi focus on RFEMLAB to convert the finite element 2D model into a 3D model, in order to obtain a more accurate actual human eye temperature [13].

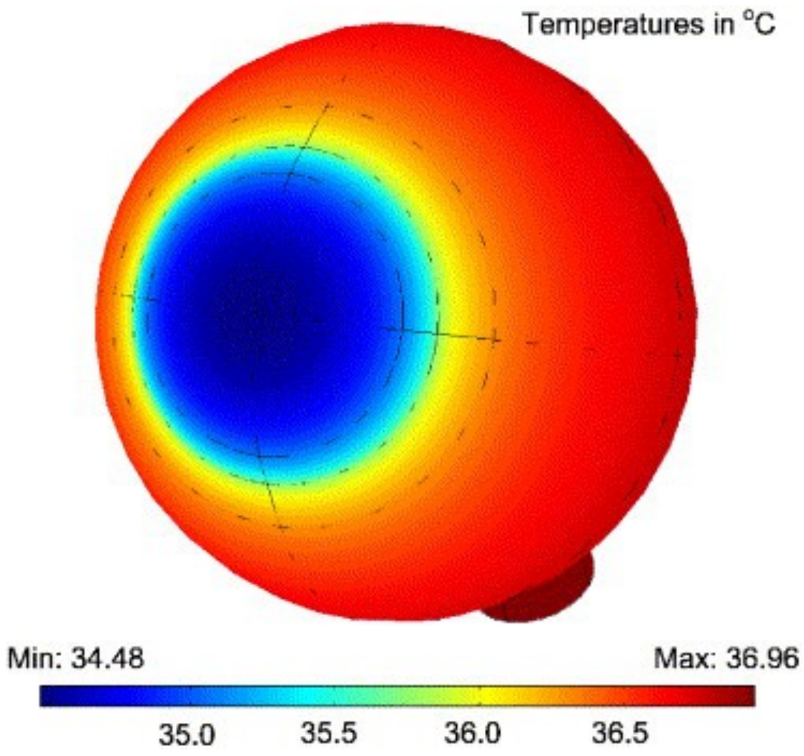


Figure 7 The temperature distribution in eye for the normal condition [13]

Deng and Liu obtained a three-dimensional model of skin surface temperature changes by Monte Carlo method (14). The following figure shows the instantaneous temperature distribution when the skin surface is turbulent.

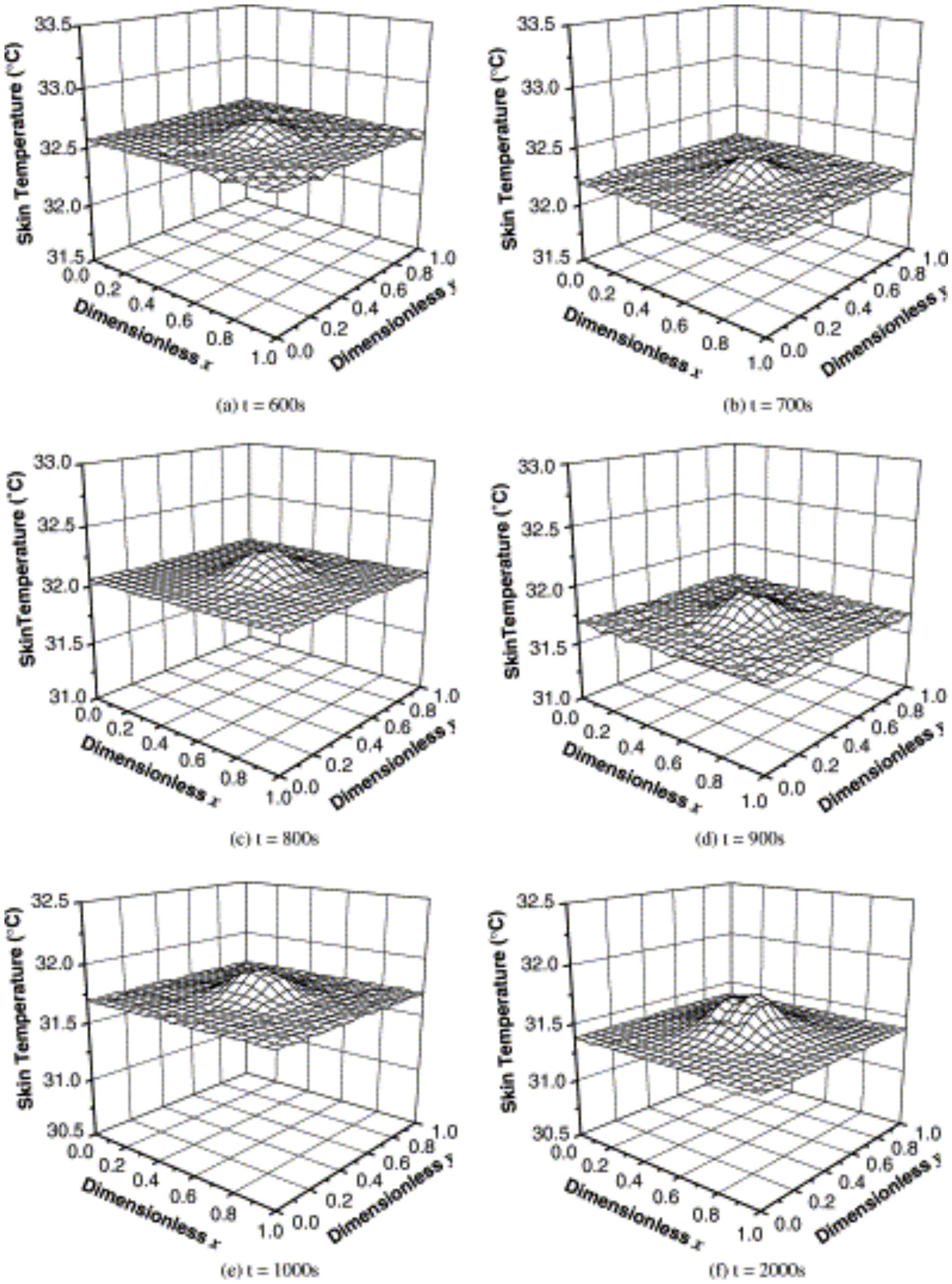


Figure 8 The instantaneous temperature distribution [14]

Programming for the Pennes equation

Programming is the most difficult part during the project. C language and C++ language are used commonly, and the common computing platforms are Visual Studio, MATLAB and CUDA.

Shen, Zhang and Yang established a model of biothermal machinery based on the bioheat transfer equation and the Duhamel-Neuman equation [7]. Finite difference schemes are discretized in MATLAB by C language [7].

CUDA is used to perform a finite element three-dimensional biothermal model [15]. The following figure shows the implementation process of CUDA on the finite element method.

Algorithm 1: Parallel Finite Difference Method Implementation

```
begin
    new = 1;
    old = 0;
    definition of block and grid dimensions sizes ;
    copy initial data and parameters from host to device;
    for t ← t_n to t_f do
        for r ← 0 to r_f do
            invoke the CUDA kernel
            cudaThreadSynchronize()
            new = !new
            old = !old
        end for
    end for
    copy back the results from device to host;
end
```

Figure 9 [15]

According to literature review, both two-dimension and three-dimension model have been established for the Pennes bioheat equation. Compared with 3D model, 2D condition is not accurate enough to show the temperature distribution in soft tissue. While most of 3D models is realized by finite element method which is difficult and needed long time to operator. In addition, the improvement of the project is to implement the three-dimension bioheat model by finite difference method which is relatively simple.

Methodology and Engineering Design

Methodology

In this project, the methodology is to solve the Pennes bioheat transfer equation which is the fundamental concept by using a discrete three-dimensional temperature distribution model. The bioheat equation is combined with the temperature represented by each grid point in the three-dimensional model. The mathematical method connecting the bioheat equation and the temperature model is the finite difference method, which discretizes the bioheat equation, and obtained numerical solution is applied to the three-dimensional model. Thus, a three-dimensional temperature model using color changes to stand for the magnitude of the numerical solution is obtained.

Engineering design

According to the object of this project which is to represent the temperature distribution of soft tissue by the Pennes equation using C++ programming, the concept of engineering design part should be the three-dimension bioheat model establishment and the code script of the solving process of the Pennes equation.

Pennes Bioheat transfer equation

As the aim is to obtain a three-dimensional mathematical model and make it show the change of human skin temperature under a specific external heat source, it is necessary to use the bioheat formula to gain the temperature in the range of the model.

$$\rho C \frac{\partial T}{\partial t} = k\frac{\partial^2 T}{\partial x^2} + k\frac{\partial^2 T}{\partial y^2} + k\frac{\partial^2 T}{\partial z^2} + w_b C_b \left(T_a - T\right) + Q_m + Q_r\left(x, y, z, t\right),$$

Figure 10 The three-dimension Pennes bioheat equation [7]

In this equation, $\rho, C\ and\ k$ are respectively the skin density, specific heat and thermal conductivity; W_b is the blood perfusion rate and C_b is the specific heat of the blood; T_a is used to represented the arterial temperature; Q_m is the metabolic heat generation rate and Q_r is the external heat source.

The three-dimension bioheat model

Figure 11 shows a cubic volume mesh which use finite difference method to solve bioheat equation. It can be seen clearly that the equation at point (i,j,k) would be described by its six neighboring points.

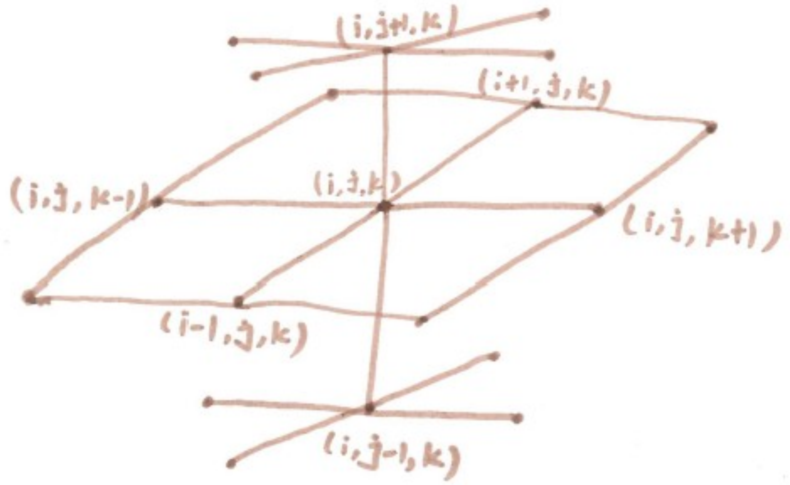


Figure 11

Without pixel values from bioheat equation's solutions, the empty, pixel-free model by Visual Studio cannot be shown. So, in this stage we use the sketch to explain. For the three-dimension model, the sketch has shown below. I choose a regular geometry as the model which stands for the soft tissue. L in the sketch is depth of the tissue in x-direction and the tissue surface is defined at x=L which means that the origin of the coordinates is inside the soft tissue. Besides, s1 and s2 are length of the tissue in y-direction and z-direction.

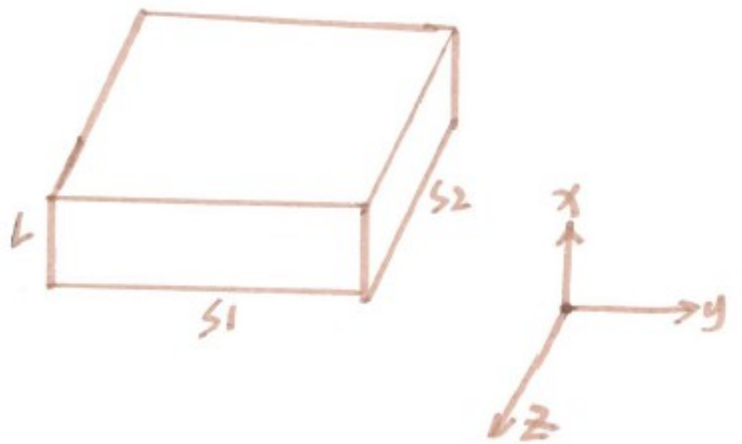


Figure 12

Code

The code scripts for Pennes bioheat equation and establishing three-dimensional model are the important part of engineering design. The realization of the finite difference method and the conversion of the temperature value to the appropriate pixel and the integration into the model need to be completed by code which would be shown in Appendix.

Findings and Discussion

The process of solving Pennes bioheat equation and simulating three-dimensional heat transfer model would be shown in this part. And the whole process can be realized by programming.

Pennes bioheat transfer equation

For this project, the fundamental concept is the bioheat equation. The three-dimension from of the equation and the definition of parameters have been shown before, which are:

$$\rho C \frac{\partial T}{\partial t} = k\frac{\partial^2 T}{\partial x^2} + k\frac{\partial^2 T}{\partial y^2} + k\frac{\partial^2 T}{\partial z^2} + w_b C_b (T_a - T) + Q_m + Q_r (x, y, z, t),$$

Figure 13 The three-dimension Pennes bioheat equation [7]

Parameter	Definition	Parameter	Definition
T	Temperature ℃	C_b	The blood specific heat $J/(kg℃)$
ρ	The tissue density kg/m^3	T_a	The arterial temperature ℃
C	The tissue specific heat $J/(kg℃)$	Q_m	The metabolic heat generation rate W/m^3
k	The tissue thermal conductivity $w/(m℃)$	Q_r	The regional heat sources W/m^3
w_b	The blood perfusion rate $kg/(m^3 s)$		

Figure 14 The definitions of parameters in bioheat equation [7]

What the important thing is the decision of boundary conditions which is decided by the condition of practical hyperthermia. In this project, the boundary conditions that would be particularly used are from an ablation heater. It gives the deep body tissue a spatial heating while protects the surface healthy tissue from injury by the surface cooling water [17]. The illustration of the hyperthermia heater is shown below.

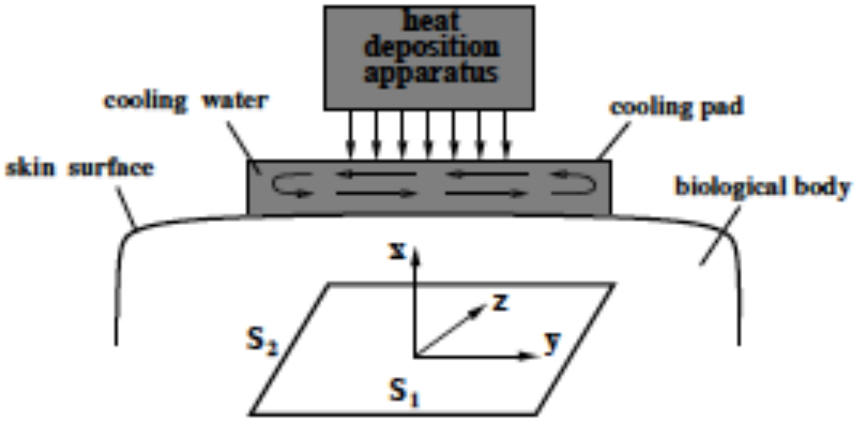


Figure 15 The illustration of the hyperthermia heater [17]

And the boundary conditions are

$$
\text{B.C.} \quad \begin{cases} T = T_c \text{ at } x = 0; \quad K\dfrac{\partial T}{\partial x} = h_f(T - T_f) \text{ at } x = L, \\ K\dfrac{\partial T}{\partial y} = 0 \text{ at } y = 0; \quad K\dfrac{\partial T}{\partial y} = 0 \text{ at } y = s_1, \\ K\dfrac{\partial T}{\partial z} = 0 \text{ at } z = 0; \quad K\dfrac{\partial T}{\partial z} = 0 \text{ at } z = s_2, \end{cases}
$$

Figure 16 The boundary conditions [17]

Where $T_c = 37°\text{C}$ which is the body core temperature;

$T_f = 25°\text{C}$ which is the cooling fluid temperature;

$h_f = 10W/m3$ which is the convection coefficient;

The definition of L, s1, s2 (all 3cm) are the length of the three-dimension model that stands for soft tissue:

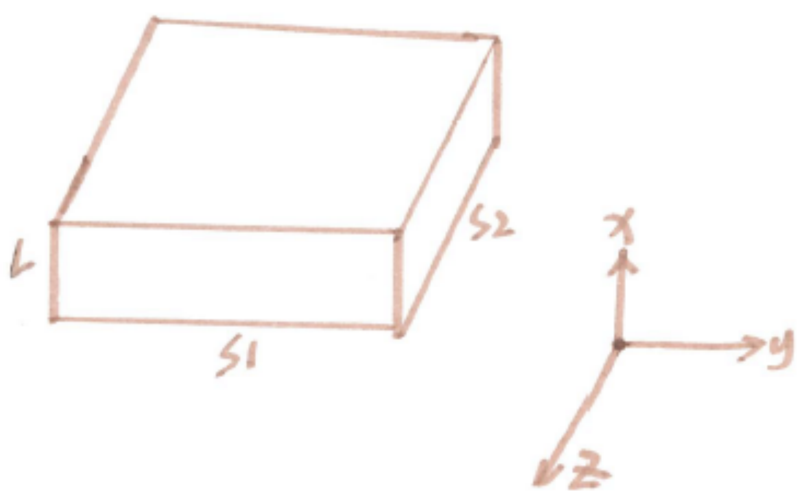


Figure 17 The three-dimension model

And the values of parameters are listed in [17]:

Parameter	Value	Parameter	Value
ρ	1000 kg/m^3	C_b	4000 $J/(kg°C)$
C	4000 $J/(kg°C)$	T_a	37 °C
K	0.5 $W/(m°C)$	Q_m	420W/m^3
W_b	0.5 $kg/(m^3s)$	Q_r	$\eta P_0(t)\exp[-\eta(L-x)]$, $P_0(t)$ is the heating power on skin surface and η is the scattering coefficient that both are vary from the heater

Figure 18

From the boundary conditions, we know that the temperature should keep on the body core temperature ($T_c = 37°C$) at x=0 which is the bottom of the three-dimensional model. At x=L which is the top of the three-dimensional model, the temperature is the cooling fluid temperature ($T_f = 25°C$) and then changes with heating time.

For the external heating source (Q_r), according to the heater we chose which takes the surface heating, the formula of external heat should be $Q_r = \eta P_0(t)\exp[-\eta(L-x)]$.

Since the solution method we choose is the finite difference method, we use 7-point implicit difference method to discrete the bioheat equation:

$$\frac{dT}{dt_{t+1}} = \frac{T_{i,j,k,t+1} - T_{i,j,k}}{\Delta t}$$

$$\frac{d^2T}{dx^2_{t+1}} = \frac{T_{i+1,j,k,t+1} - 2T_{i,j,k,t+1} + T_{i-1,j,k,t+1}}{\Delta x_i^2}$$

$$\frac{d^2T}{dy^2_{t+1}} = \frac{T_{i,j+1,k,t+1} + T_{i,j-1,k,t+1} - 2T_{i,j,k,t+1}}{\Delta y_i^2}$$

$$\frac{d^2T}{dz^2_{t+1}} = \frac{T_{i,j,k+1,t+1} + T_{i,j,k-1,t+1} - 2T_{i,j,k,t+1}}{\Delta \cancel{y} z_i^2}$$

$$\frac{\rho c}{k} \cdot \frac{T_{i,j,k,t+1} - T_{i,j,k,t}}{\Delta t} - \frac{Q}{k} = \frac{T_{i+1,j,k,t+1} + T_{i-1,j,k,t+1} - 2T_{i,j,k,t+1}}{\Delta x_i^2}$$
$$+ \frac{T_{i,j+1,k,t+1} + T_{i,j-1,k,t+1} - 2T_{i,j,k,t+1}}{\Delta y_i^2}$$
$$+ \frac{T_{i,j,k+1,t+1} + T_{i,j,k-1,t+1} - 2T_{i,j,k,t+1}}{\Delta z_i^2}$$

Figure 19

The findings above show that the parameters values and boundary conditions of the Pennes bioheat equation and the finite difference method which is used to discrete the equation, so we can solve the equation.

After adding the parameter values, the equation should be:

$$4 * 10^6 \frac{\partial T}{\partial t} = 0.5 * \frac{\partial^2 T}{\partial x^2} + 0.5 * \frac{\partial^2 T}{\partial y^2} + 0.5 * \frac{\partial^2 T}{\partial z^2} + 0.5 * 4000(273.15 + 37 - \mathrm{T}) + 420 + \eta \mathrm{P} * \exp(-\eta(\mathrm{L} - \mathrm{x}))$$

Which is:

$$8 * 10^6 \frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} + 2\eta \mathrm{P} * \exp(-\eta(\mathrm{L} - \mathrm{x})) - 4000\mathrm{T} + 1241440$$

The difference formula we use are:

$$\mathrm{t} = \mathrm{n}\Delta \mathrm{t}, \mathrm{x} = \mathrm{i}\Delta \mathrm{x}, \mathrm{y} = \mathrm{j}\Delta \mathrm{y}, \mathrm{z} = \mathrm{k}\Delta \mathrm{z}$$

$$\frac{\partial T}{\partial t} = \frac{T_{i,j,k}^{n+1} - T_{i,j,k}^{n}}{\Delta t}$$

$$\frac{\partial^2 T}{\partial x^2} = \frac{T_{i+1,j,k}^n - 2T_{i,j,k}^n + T_{i-1,j,k}^n}{(\Delta x)^2}$$

$$\frac{\partial^2 T}{\partial y^2} = \frac{T_{i,j+1,k}^n - 2T_{i,j,k}^n + T_{i,j-1,k}^n}{(\Delta y)^2}$$

$$\frac{\partial^2 T}{\partial z^2} = \frac{T_{i,j,k+1}^n - 2T_{i,j,k}^n + T_{i,j,k-1}^n}{(\Delta z)^2}$$

We have known that in this boundary condition, the bottom of the model should keep the core temperature, so what we need to solve are the top and side temperature of the model.

Firstly, for the top of the model (X=L):

$$8*10^6\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} - 3600\mathrm{T} + 2\eta\mathrm{P} + 1241440$$

$$T(0) = 25℃ = 298.15K$$

We discrete the equation:

$$8*10^6\frac{T_{i,j,k}^{n+1} - T_{i,j,k}^n}{\Delta t} = \frac{T_{i,j+1,k}^n - 2T_{i,j,k}^n + T_{i,j-1,k}^n}{(\Delta y)^2} + \frac{T_{i,j,k+1}^n - 2T_{i,j,k}^n + T_{i,j,k-1}^n}{(\Delta z)^2} - 3600\mathrm{T} + 2\eta\mathrm{P} + 1241440$$

$$\left(T_{i,j,k}^{n+1} - T_{i,j,k}^n\right) = \frac{\Delta t}{8*10^6}\left(\frac{T_{i,j+1,k}^n - 2T_{i,j,k}^n + T_{i,j-1,k}^n}{(\Delta y)^2} + \frac{T_{i,j,k+1}^n - 2T_{i,j,k}^n + T_{i,j,k-1}^n}{(\Delta z)^2} - 3600\mathrm{T} + 2\eta\mathrm{P} + 1241440\right)$$

$$T_{i,j,k}^{n+1} = \frac{\Delta t}{8*10^6}\left(\frac{T_{i,j+1,k}^n - 2T_{i,j,k}^n + T_{i,j-1,k}^n}{(\Delta y)^2} + \frac{T_{i,j,k+1}^n - 2T_{i,j,k}^n + T_{i,j,k-1}^n}{(\Delta z)^2} - 3600T_{i\Delta x,j\Delta y,k\Delta z}^{\mathrm{n}\Delta\mathrm{t}} + 2\eta\mathrm{P} + 1241440\right) + T_{i,j,k}^n$$

$$T_{j,k}^{n+1} = \frac{\Delta t}{8*10^6}\left(\frac{T_{j+1,k}^n - 2T_{j,k}^n + T_{j-1,k}^n}{(\Delta y)^2} + \frac{T_{j,k+1}^n - 2T_{j,k}^n + T_{j,k-1}^n}{(\Delta z)^2} - 3600T_{j\Delta y,k\Delta z}^{\mathrm{n}\Delta\mathrm{t}} + 2\eta\mathrm{P} + 1241440\right) + T_{j,k}^n$$

$\Delta\mathrm{y} = \Delta\mathrm{z}$：

$$T_{j,k}^{n+1} = \frac{\Delta t}{8*10^6}\left(\frac{T_{j+1,k}^n - 4T_{j,k}^n + T_{j-1,k}^n + T_{j,k+1}^n + T_{j,k-1}^n}{(\Delta y)^2} - 3600T_{j\Delta y,k\Delta y}^{\mathrm{n}\Delta\mathrm{t}} + 2\eta\mathrm{P} + 1241440\right) + T_{j,k}^n$$

We assume that $\Delta\mathrm{y} = 1\mathrm{mm}, \Delta\mathrm{t} = 1,\ \eta\mathrm{P} = 100*40$:

$$T_{j,k}^{n+1} = \frac{1}{8}\ \left(T_{j+1,k}^{n} - 4T_{j,k}^{n} + T_{j-1,k}^{n} + T_{j,k+1}^{n} + T_{j,k-1}^{n}\right)\ + \frac{1}{8*10^6}\left(-3600T_{j,k}^{\mathrm{n}} + 1249440\right) + T_{j,k}^{n}$$

Then we solve the side of the model.

When Y= s1,

$$8*10^6\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} + 2\eta\mathrm{P}*\exp(-\eta(\mathrm{L}-\mathrm{x})) - 4000\mathrm{T} + 1241440$$

$$8*10^6\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial z^2} + 2\eta\mathrm{P}*\exp(-\eta(\mathrm{L}-\mathrm{x})) - 4000\mathrm{T} + 1241440$$

Taking the difference fomula:

$$\begin{aligned} 8*10^6\frac{T_{i,j,k}^{n+1} - T_{i,j,k}^{n}}{\Delta t} &= 10^6\frac{T_{i+1,j,k}^{n} - 2T_{i,j,k}^{n} + T_{i-1,j,k}^{n}}{(\Delta x)^2} + 10^6\frac{T_{i,j,k+1}^{n} - 2T_{i,j,k}^{n} + T_{i,j,k-1}^{n}}{(\Delta z)^2} + 2\eta\mathrm{P} \\ &\quad *\exp(-\eta(\mathrm{L}-\mathrm{i}\Delta x)) - 4000T_{j\Delta y,k\Delta y}^{\mathrm{n\Delta t}} + 1241440 \end{aligned}$$

Assume that dt=1, dx=dz=1, dy=0, y=s:

$$\begin{aligned} T_{i,k}^{n+1} &= \frac{1}{8}\left(T_{i+1,k}^{n} - 4T_{i,k}^{n} + T_{i-1,k}^{n} + T_{i,k+1}^{n} + T_{i,k-1}^{n}\right) \\ &\quad + \frac{1}{8*10^6}\left(2\eta\mathrm{P}*\exp\left(-\eta(\mathrm{L}-\mathrm{i})\right) - 4000T_{i,k}^{\mathrm{n}} + 1241440\right) + T_{i,k}^{n} \end{aligned}$$

Next steps would be similar with the top of the model:

$$T_{i}^{n+1} = \frac{\eta p}{4}\exp\left(-\eta(L-i)\right) + \frac{1}{4*10^6}(620720 - 2000T_{i}^{n}) + T_{i}^{n}$$

Temperature distribution model

The mathematical model is realized by programming. Firstly, we establish an empty model with no relation to temperature. After discretizing the equation, the temperature can be obtained. In this situation, different temperature would be represented by different color which we set the color band within a range here. Finally, we put the color into the model with the position we calculated from the bioheat equation. In Visual Studio, we can gain the animation video that shows the temperature change of the model varies with the time under the external surface heating condition. In this report, we change the video to figure with different heating time.
Time (t)= 0s:

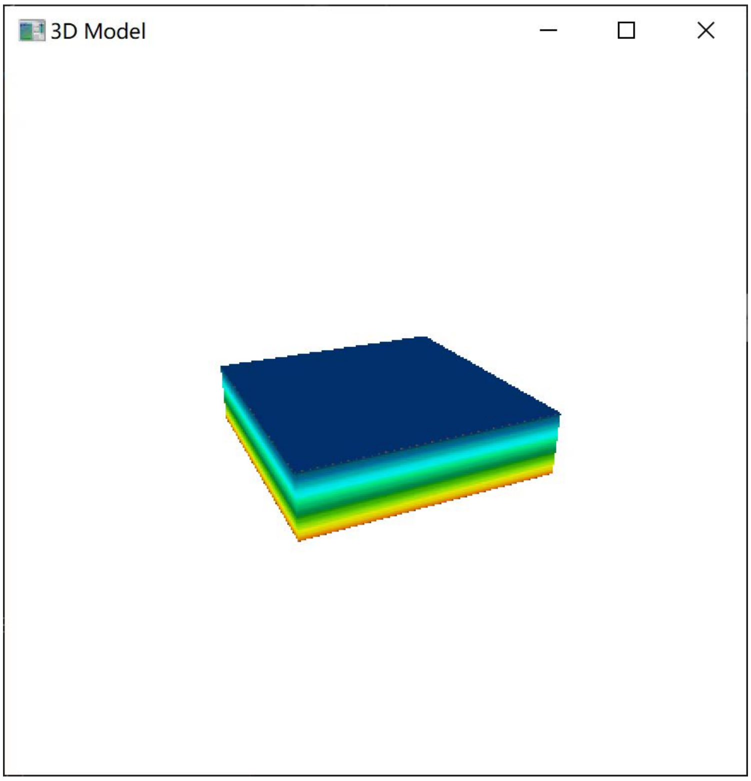


Figure 20 The temperature distribution initially

Time (t)= 500s:

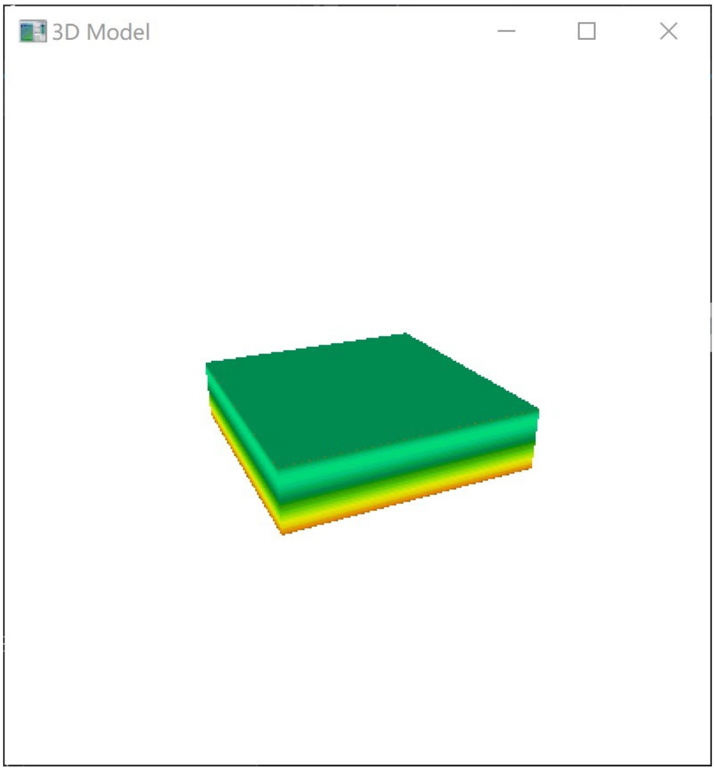


Figure 21

Time (t)= 1000s:

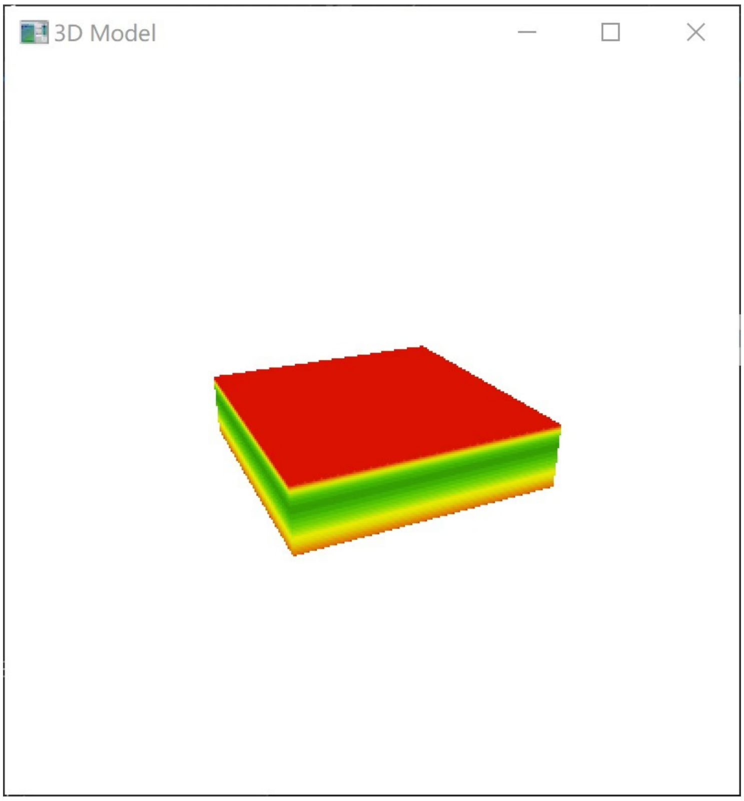


Figure 22

Time (t)= 3000s:

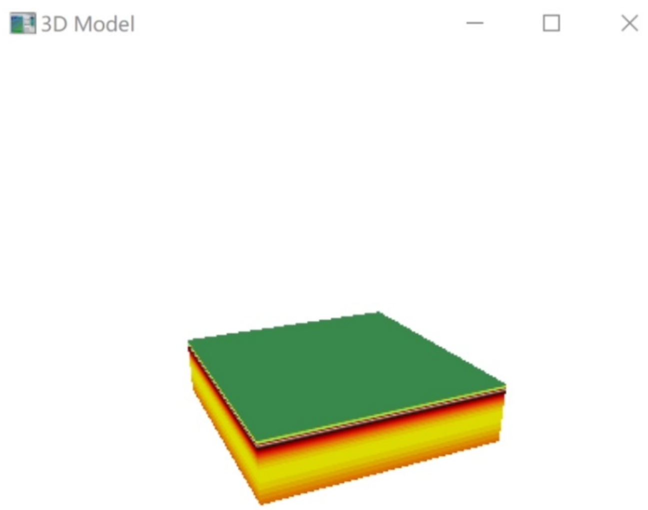
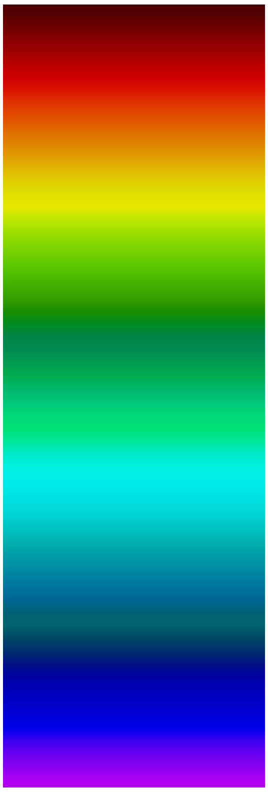


Figure 23

Here is the color band. The rang is from $22°C\ to\ 40°C$ and $22°C$ is represented by purple.



Conclusion

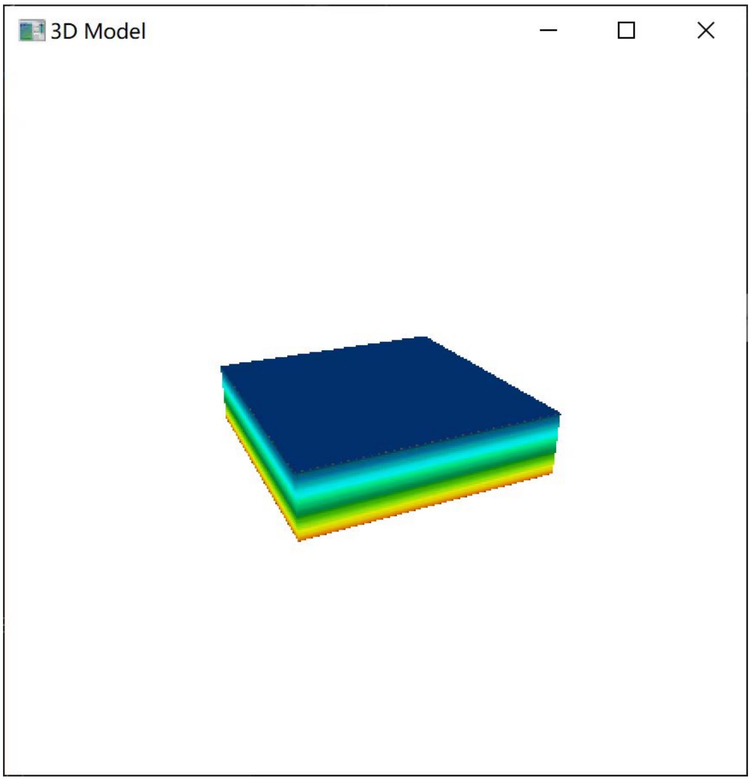
In this project, a partial differential heat transfer equation, Pennes bioheat transfer equation, is used as the fundamental concept to analyze and simulate the temperature distribution of human tissue. In order to solve the equation, the finite difference method is used to discretize the equation, and a three-dimensional temperature model is established to express the relationship between the equation and the temperature. The definitions and values of the variables of the equation are obtained by literature review. For the process, C++ programming language and Visual Studio programing software are used to realize the establishment of three-dimensional model and the simulation of temperature change process.

The Pennes bioheat equations with discretization for the top and side of the model are as follow:

$$T_{j,k}^{n+1} = \frac{1}{8}\left(T_{j+1,k}^{n} - 4T_{j,k}^{n} + T_{j-1,k}^{n} + T_{j,k+1}^{n} + T_{j,k-1}^{n}\right) + \frac{1}{8 * 10^6}\left(-3600T_{j,k}^{n} + 1249440\right) + T_{j,k}^{n}$$

$$T_{i}^{n+1} = \frac{\eta p}{4}\exp\left(-\eta(L-i)\right) + \frac{1}{4 * 10^6}\left(620720 - 2000T_{i}^{n}\right) + T_{i}^{n}$$

With a loop of the equation realized by code, we can obtain the whole temperature matrix in the range of model. Then we represent the temperature matrix in color band of the model. Here is a figure shows the temperature distribution initially (t=0s):



The aim of the project has been achieved basically. However, what we use is the simple boundary condition and external heating. In the future work, we can try more complex and accurate boundary condition heating style, such as points heating.

Reflection and Future work

The aim of this project is to simulate the temperature distribution of human tissues and the process of temperature change under external heat sources. In the implementation, we use a simple heating method, i.e., surface heating, which makes the temperature change of the model in the plane almost the same because of the small volume of the model. In the following work, we can try different heating methods and consult more literature for more complex and precise boundary condition.

The skin damage prediction would be the next step for this project which is the most important part of the future work. Compared with the achievement part, the skin damage prediction should be simulation by a three-dimensional model with more fundamental concepts such as Duhamel-Neuman equation for thermal stress and strain calculation and Henriques's suggestion of Arrhenius equation for the skin damage prediction [19]. For this work, finite difference method is still the mathematical method to discrete the fundamental equations. And the establishment of three-dimensional model is also realized by code.

Reference

[1] M. A. Giordano, G. Gutierrez, and C. Rinaldi, "Fundamental solutions to the bioheat equation and their application to magnetic fluid hyperthermia," *International Journal of Hyperthermia,* vol. 26, no. 5, pp. 475-484, 2010/08/01 2010.

[2] H.-W. Huang and T.-L. Horng, "Chapter 1 - Bioheat Transfer and Thermal Heating for Tumor Treatment," in *Heat Transfer and Fluid Flow in Biological Processes*, S. M. Becker and A. V. Kuznetsov, Eds. Boston: Academic Press, 2015, pp. 1-42.

[3] D. B. Rodrigues, P. J. S. Pereira, P. Limão-Vieira, P. R. Stauffer, and P. F. Maccarini, "Study of the one dimensional and transient bioheat transfer equation: Multi-layer solution development and applications," *International Journal of Heat and Mass Transfer,* vol. 62, pp. 153-162, 2013/07/01/ 2013.

[4] S. Lin and T. Chou, "Numerical Analysis of the Pennes Bioheat Transfer Equation on Skin Surface," in *2015 Third International Conference on Robot, Vision and Signal Processing (RVSP)*, 2015, pp. 71-74.

[5] X. Li, Y. Zhong, R. Jazar, and A. Subic, "Thermal-mechanical deformation modelling of soft tissues for thermal ablation," *Bio-medical materials and engineering,* vol. 24, no. 6, pp. 2299-2310, 2014.

[6] Murat Tunç, Ünal Çamdali, Cem Parmaksizoğlu and Sermet Çikrikçi, "The bio- heat transfer equation and its applications in hyperthermia treatments", *Engineering Computations*, vol. 23, no. 4, pp. 451-463, 2 0 1 6 .

[7] W. Shen, J. Zhang, and F. Yang, "Modeling and numerical simulation of bioheat transfer and biomechanics in soft tissue," *Mathematical and Computer Modelling,* vol. 41, no. 11-12, pp. 1251-1265, 2005.

[8] *Fpl.fs.fed.us*, 2019. [Online]. Available: https://www.fpl.fs.fed.us/documnts/pdf2001/green01d.pdf. [Accessed: 29-May- 2019].

[9] J. J. Zhao, J. Zhang, N. Kang, and F. Yang, "A two level finite difference scheme for one dimensional Pennes' bioheat equation," *Applied Mathematics and Computation,* vol. 171, no. 1, pp. 320-331, 2005/12/01/ 2005.

[10] S. Zozor *et al.*, "A numerical scheme for modeling wavefront propagation on a monolayer of arbitrary geometry," *IEEE transactions on biomedical engineering,* vol. 50, no. 4, pp. 412-420, 2003.

[11] T. Gowrishankar, D. A. Stewart, G. T. Martin, and J. C. Weaver, "Transport lattice models of heat transport in skin with spatially heterogeneous, temperature-dependent perfusion," *BioMedical Engineering OnLine,* vol. 3, no. 1, p. 42, 2004.

[12] M. M. Elwassif, Q. Kong, M. Vazquez, and M. Bikson, "Bio-heat transfer model of deep brain stimulation-induced temperature changes," *Journal of Neural Engineering,* vol. 3, no. 4, pp. 306-315, 2006/11/06 2006.

[13] E. Y. K. Ng and E. H. Ooi, "Ocular surface temperature: A 3D FEM prediction using bioheat equation," *Computers in Biology and Medicine,* vol. 37, no. 6, pp. 829-835, 2007/06/01/ 2007.

[14] Z.-S. Deng and J. Liu, "Mathematical modeling of temperature mapping over skin surface and its implementation in thermal disease diagnostics," *Computers in Biology and Medicine,* vol. 34, no. 6, pp. 495-521, 2004/09/01/ 2004.

[15] R. F. Reis, F. d. S. Loureiro, and M. Lobosco, "3D numerical simulations on GPUs of hyperthermia with nanoparticles by a nonlinear bioheat model," *Journal of Computational and Applied Mathematics,* vol. 295, pp. 35-47, 2016/03/15/ 2016.

[16] X. Li, "Thermal-mechanical response modelling and thermal damage prediction of soft tissues during thermal ablation," 2017.p7

[17] S. Karaa, J. Zhang, and F. Yang, "A numerical study of a 3D bioheat transfer problem with different spatial heating," *Mathematics and Computers in Simulation,* vol. 68, no. 4, pp. 375-388, 2005.

[18] B. Mochnacki and A. Piasecka-Belkhayat, "Numerical modeling of skin tissue heating using the interval finite difference method," *MCB: Molecular & Cellular Biomechanics,* vol. 10, no. 3, pp. 233-244, 2013.

[19] W. Shen, J. Zhang, and F. Yang, "Skin thermal injury prediction with strain energy," ed: De Gruyter, 2005.

Appendix

Code for Pennes equation

Top of the model

```
void Solve::init_data()
{

    for (int i = 0; i < width; i++)
        for (int j = 0; j < height; j++)
            surface_data[i][j][0] = 25;

    for (int t = 1; t < time; t++) {
        double y = (-(48840 + 2 * h * p) / 3600.0) * exp(-36.0 / 80000.0 * t) + (138840 + 2 * h * p) / 3600.0;
        for (int i = 0; i < height; i++) {
            surface_data[i][0][t] = y;
            surface_data[i][height - 1][t] = y;
        }
        for (int j = 0; j < width; j++) {
            surface_data[0][j][t] = y;
            surface_data[width - 1][j][t] = y;
        }
    }

    for (int t = 1; t < time; t++)
        for (int i = 1; i < width - 1; i++)
            for (int j = 1; j < height - 1; j++) {
                surface_data[i][j][t] = (1 / 8.0) * (surface_data[i + 1][j][t - 1] + surface_data[i - 1][j][t - 1]
                + surface_data[i][j + 1][t - 1] + surface_data[i][j - 1][t - 1] - 4 * surface_data[i][j][t - 1]) +
                    (1 / 8000000.0) * (-3600 * surface_data[i][j][t - 1] + 2 * h * p + 138840) + surface_data[i][j][t - 1];
            }

    //for (int i = 0; i < width/2; i++)
    //  for (int j = 0; j < height/2; j++)
    //      cout << surface_data[i][j][time-1]<<" ";


    ifstream in("color.txt");
    for (int i = 0; i < 60; i++) {
        in >> color[i].r;
        in >> color[i].g;
        in >> color[i].b;
    }
    in.close();

    //cout << surface_data[50][50][time-1] << endl;
```

Side of the model

```
for (int t = 0; t < time; t++) {
    for (int i = 0; i < width; i++) {
        around_data[i][0][t] = 37;
        around_data[i][L - 1][t] = surface_data[i][height - 1][t];
    }
}

for (int i = 1; i < L - 1; i++) {

    around_data[0][i][0] = 37.0 - (37.0 - 25.0) * i / L;
    for (int j = 1; j < width; j++)
        around_data[j][i][0] = around_data[0][i][0];
}

for (int t = 1; t < time; t++) {
    for (int i = 1; i < L - 1; i++) {
        around_data[0][i][t] = 1 / 8.0 * (around_data[0][i + 1][t - 1]
        + around_data[0][i - 1][t - 1] - 2 * around_data[0][i][t - 1])
        + (1 / 8000000.0) * ((h * p) * exp(-1000 * h * (L - i)) - 4000 * around_data[0][i][t - 1]
        + 148840) + around_data[0][i][t - 1];
        around_data[width - 1][i][t] = around_data[0][i][t];
    }
}




for (int t = 1; t < time; t++)
    for (int i = 1; i < width - 1; i++)
        for (int j = 1; j < L - 1; j++) {
            around_data[i][j][t] = (1 / 8.0) * (around_data[i + 1][j][t - 1] + around_data[i - 1][j][t - 1]
            + around_data[i][j + 1][t - 1] + around_data[i][j - 1][t - 1] - 4 * around_data[i][j][t - 1]) +
                (1 / 8000000.0) * (-4000 * around_data[i][j][t - 1]
            + 2 * h * p * exp(-h * (L - j)) + 148840) + around_data[i][j][t - 1];
        }

/*
for (int i = 0; i < width; i++) {
    for (int j = 0; j < L; j++)
        cout << around_data[i][j][0] << " ";
    cout << endl;
```

Code for model

```
using namespace std;
GLuint texture[3];//6 textures for 6 faces of the cube
GLfloat xRot, yRot, zRot;//control cube's rotation
//load the bitmap and convert it into a texture

Solve image;
int currentTime = 1;
void LoadGLTextures()
{
    char surface[width * height * 3];
    char around[width * L * 3];
    char bottom[width * height * 3];

    image.getImage(surface, around, bottom, currentTime%time);
    currentTime += 1;

    //int width11, height11, nrChannels;

    //unsigned char* data = stbi_load("surface.bmp", &width11, &height11, &nrChannels, 0);
    glBindTexture(GL_TEXTURE_2D, texture[0]);
    //glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, width11, height11, 0, GL_RGB, GL_UNSIGNED_BYTE, data);
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, width, height, 0, GL_RGB, GL_UNSIGNED_BYTE, surface);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);

    //data = stbi_load("around.bmp", &width11, &height11, &nrChannels, 0);
    glBindTexture(GL_TEXTURE_2D, texture[1]);
    //glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, width11, height11, 0, GL_RGB, GL_UNSIGNED_BYTE, data);
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, width, L, 0, GL_RGB, GL_UNSIGNED_BYTE, around);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);

    //data = stbi_load("surface.bmp", &width11, &height11, &nrChannels, 0);
    glBindTexture(GL_TEXTURE_2D, texture[2]);
    //glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, width11, height11, 0, GL_RGB, GL_UNSIGNED_BYTE, data);
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, width, height, 0, GL_RGB, GL_UNSIGNED_BYTE, bottom);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);


}
```

```
int init()
{
    glEnable(GL_TEXTURE_2D);
    image.init_data();
    glGenTextures(3, texture);
    LoadGLTextures();

    glShadeModel(GL_SMOOTH);
    glClearColor(0.0f, 0.0f, 0.0f, 0.5f);
    glClearDepth(1.0f);
    glEnable(GL_DEPTH_TEST);
    glDepthFunc(GL_LEQUAL);
    glHint(GL_PERSPECTIVE_CORRECTION_HINT, GL_NICEST);
    xRot += 25.0f;
    yRot += 25.0f;





    return 1;
}
void display()
{
    LoadGLTextures();
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();
    glTranslatef(0.0f, 0.0f, -5.0f);
    glRotatef(xRot, 1.0f, 0.0f, 0.0f);
    glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glRotatef(zRot, 0.0f, 0.0f, 1.0f);
    //glRotatef(45,1.0f,0.0f,0.0f);
    //glRotatef(45,0.0f,1.0f,0.0f);

    glBindTexture(GL_TEXTURE_2D, texture[1]);
    glBegin(GL_QUADS);
    // Front Face
    glTexCoord2f(0.0f, 0.0f); glVertex3f(-1.0f, -L * 1.0 / width, 1.0f);
    glTexCoord2f(1.0f, 0.0f); glVertex3f(1.0f, -L * 1.0 / width, 1.0f);
    glTexCoord2f(1.0f, 1.0f); glVertex3f(1.0f, L * 1.0 / width, 1.0f);
    glTexCoord2f(0.0f, 1.0f); glVertex3f(-1.0f, L * 1.0 / width, 1.0f);
```

```
    glEnd();

    glBindTexture(GL_TEXTURE_2D, texture[1]);
    glBegin(GL_QUADS);
    // Back Face
    glTexCoord2f(1.0f, 0.0f); glVertex3f(-1.0f, -L * 1.0 / width, -1.0f);
    glTexCoord2f(1.0f, 1.0f); glVertex3f(-1.0f, L * 1.0 / width, -1.0f);
    glTexCoord2f(0.0f, 1.0f); glVertex3f(1.0f, L * 1.0 / width, -1.0f);
    glTexCoord2f(0.0f, 0.0f); glVertex3f(1.0f, -L * 1.0 / width, -1.0f);
    glEnd();

    glBindTexture(GL_TEXTURE_2D, texture[0]);
    glBegin(GL_QUADS);
    // Top Face
    glTexCoord2f(0.0f, 1.0f); glVertex3f(-1.0f, L * 1.0 / width, -1.0f);
    glTexCoord2f(0.0f, 0.0f); glVertex3f(-1.0f, L * 1.0 / width, 1.0f);
    glTexCoord2f(1.0f, 0.0f); glVertex3f(1.0f, L * 1.0 / width, 1.0f);
    glTexCoord2f(1.0f, 1.0f); glVertex3f(1.0f, L * 1.0 / width, -1.0f);
    glEnd();

    glBindTexture(GL_TEXTURE_2D, texture[2]);
    glBegin(GL_QUADS);
    // Bottom Face
    glTexCoord2f(1.0f, 1.0f); glVertex3f(-1.0f, -L * 1.0 / width, -1.0f);
    glTexCoord2f(0.0f, 1.0f); glVertex3f(1.0f, -L * 1.0 / width, -1.0f);
    glTexCoord2f(0.0f, 0.0f); glVertex3f(1.0f, -L * 1.0 / width, 1.0f);
    glTexCoord2f(1.0f, 0.0f); glVertex3f(-1.0f, -L * 1.0 / width, 1.0f);
    glEnd();

    glBindTexture(GL_TEXTURE_2D, texture[1]);
    glBegin(GL_QUADS);
    // Right face
    glTexCoord2f(1.0f, 0.0f); glVertex3f(1.0f, -L * 1.0 / width, -1.0f);
    glTexCoord2f(1.0f, 1.0f); glVertex3f(1.0f, L * 1.0 / width, -1.0f);
    glTexCoord2f(0.0f, 1.0f); glVertex3f(1.0f, L * 1.0 / width, 1.0f);
    glTexCoord2f(0.0f, 0.0f); glVertex3f(1.0f, -L * 1.0 / width, 1.0f);
    glEnd();

    glBindTexture(GL_TEXTURE_2D, texture[1]);
    glBegin(GL_QUADS);
    // Left Face
```

```
    glBegin(GL_QUADS);
    // Left Face
    glTexCoord2f(0.0f, 0.0f); glVertex3f(-1.0f, -L * 1.0 / width, -1.0f);
    glTexCoord2f(1.0f, 0.0f); glVertex3f(-1.0f, -L * 1.0 / width, 1.0f);
    glTexCoord2f(1.0f, 1.0f); glVertex3f(-1.0f, L * 1.0 / width, 1.0f);
    glTexCoord2f(0.0f, 1.0f); glVertex3f(-1.0f, L * 1.0 / width, -1.0f);
    glEnd();

    /*xRot += 0.3f;
    yRot += 0.4f;
    zRot += 0.5f;*/
    glutSwapBuffers();
}
void reshape(int w, int h1)
{
    if (0 == h1)
        h1 = 1;
    glViewport(0, 0, (GLsizei)w, (GLsizei)h1);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(60.0f, (GLfloat)w / (GLfloat)h1, 1, 100);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}
void keyboard(unsigned char key, int x, int y)
{
    switch (key) {
    case 'x':
        xRot += 2.5f;
        glutPostRedisplay();
        break;
    case 'y':
        yRot += 2.5f;
        glutPostRedisplay();
        break;
    case 'z':
        zRot += 2.5f;
        glutPostRedisplay();
        break;
    default:
        break;
```

```
            break;
        }
    }
void OnTimer(int value)
{
    display();

    glutTimerFunc(speed, OnTimer, 1);
}
int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
    glutInitWindowSize(400, 400);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("3D Model");
    init();
    glutDisplayFunc(display);
    glutTimerFunc(speed, OnTimer, 1);

    glutReshapeFunc(reshape);
    glutKeyboardFunc(keyboard);
    glutMainLoop();
    return 0;
}
```