An improved three phase lag model for heat transfer in human tissue

Matthew Woods (m3woods@ucsd.edu)
Alon Pavlov (alpavlov@ucsd.edu)

Abstract

Cryosurgery involves the application of extreme cold to destroy abnormal or diseased tissue using liquid nitrogen, while heat-induced techniques like heat ablation utilize rapid and extreme heat to induce necrosis in targeted tissues. Both methods benefit from precision, minimizing damage to surrounding tissues and allowing for outpatient procedures. However, challenges such as the need for multiple treatment sessions and varying tissue thermal properties exist. To address these challenges, Kumar and Kaur developed a new heat transfer model to simulate bio-heat transfer through epidermal skin tissue using a dual-phase lag modification of Fourier's law. This model incorporates relaxation times for both heat flux and temperature gradient, and further improvements were made by including evaporative cooling of sweat and enhanced boundary conditions.

Background

Heat-induced and cryosurgical techniques are emerging as innovative methods for treating various lesions throughout the body, with a primary focus on the treatment of cancerous tumors. These minimally invasive techniques offer significant advantages over traditional surgical methods, including reduced recovery times and lower risk of complications. The development and refinement of these techniques have the potential to revolutionize the field of surgical oncology and other medical specialties.

Cryosurgery involves the application of extreme cold to destroy abnormal or diseased tissue. This is typically achieved using a medical device that contains liquid nitrogen, which is cooled to temperatures ranging from -170 to -190°C [1]. The device is applied directly to the target tissue for several seconds to minutes. The extreme cold causes the water content within the cells to form ice crystals, leading to cellular damage and cell death. This process, known as cryonecrosis [1], is highly effective in destroying targeted tissues while sparing surrounding healthy tissue. One of the key advantages of cryosurgery is its precision. Cryogenic probes can be carefully controlled, allowing surgeons to target small, specific areas with minimal collateral damage. This precision is particularly beneficial in the treatment of tumors located near critical structures or within delicate organs. Furthermore, cryosurgery offers an alternative to traditional surgical methods, such as excision with scalpels and knives, which can be more invasive and carry higher risks of bleeding and infection.

In contrast to cryosurgery, heat-induced surgical techniques, such as heat ablation, utilize rapid and extreme heat application to induce necrosis in targeted tissues. Heat ablation works by raising the temperature of the target tissue to a level that causes protein denaturation and coagulative necrosis. This process effectively destroys the treated tissue while minimizing damage to surrounding areas The sources of heat in these procedures can vary, including high-intensity focused ultrasound (HIFU), radiofrequency, high-frequency lasers, and

microwaves [2]. Each of these methods has demonstrated effectiveness in clinical settings, offering various benefits depending on the specific application and target tissue. One of the significant benefits of heat ablation is its ability to achieve hemostasis through the cauterization of blood vessels, thereby reducing intraoperative bleeding and the risk of postoperative hemorrhage.

Both cryosurgery and heat-induced techniques offer several advantages over conventional surgical methods. They are generally less invasive, which leads to shorter recovery times and reduced hospital stays [2]. The precision of these techniques minimizes damage to surrounding tissues and reduces the likelihood of complications such as infection. Additionally, these methods can be performed on an outpatient basis, further decreasing the burden on healthcare systems and improving patient convenience. However, there are also several challenges and drawbacks associated with these techniques. One of the primary issues is the need for multiple treatment sessions. This is often due to the heterogeneity of human tissue, which can result in uneven treatment effects and incomplete ablation of the target area. Furthermore, the thermal properties of different tissues can vary significantly, making it difficult to predict and control the extent of thermal damage accurately [3].

To address these challenges, researchers have developed mathematical models and simulations to better understand heat transfer through human tissues. These models aim to predict the distribution of temperature within tissues during and after the application of cryogenic or thermal energy. By incorporating factors such as tissue composition, blood flow, and thermal conductivity, these models can provide insights into the optimal parameters for treatment. For instance, in heat ablation, understanding how heat propagates through various tissues allows clinicians to apply just enough heat for a specific duration to achieve the desired therapeutic effect without causing unnecessary damage to surrounding tissues. These models can also help in planning treatment protocols, optimizing the placement and movement of probes, and reducing the need for multiple treatment sessions.

In the paper by Kumar and Kaur, a new heat transfer model was created to simulate the bio heat transfer through epidermal skin tissue [4]. The model is based on a dual-phase lag modification of Fourier's law of heat conduction, initially described by Pennes [5]. This approach enhances the classical model by incorporating two relaxation times: one for the heat flux (τ_q) and another for the temperature gradient (τ_T). The traditional Fourier's law of heat conduction is expressed as:

$$\vec{q}(r,t) = -k\vec{\nabla} q(r,t), \tag{1}$$

where q(r,t) is the heat flux vector, k is the thermal conductivity, and T is the temperature field. Tzou modified this model to account for the relaxation times associated with heat flux and temperature gradient, resulting in the dual-phase lag model [6]:

$$\vec{q}(r,t+\tau_q) = -k\vec{\nabla} T(r,t), \tag{2}$$

In materials like human skin, which exhibit heterogeneous properties, these relaxation times are essential to accurately represent the delays in heat movement in both time and space. The dual-phase lag model captures the delays in heat transport caused by the application of heat

flux and the establishment of a temperature gradient [6]. However, Kumar and Kaur proposed an enhanced model by including the relaxation due to the thermal displacement gradient, predicting a more accurate representation of heat transfer. To formulate this improved model, Kumar and Kaur first utilized a two-dimensional energy balance equation (EBE) to account for the energy balance based on blood flow and metabolic heat production [4]. These factors serve as primary heat sources in the skin and depend on parameters such as blood perfusion rate, tissue density, initial temperature, and the specific heat capacity of the skin. The energy balance equation is given by:

$$\vec{q}(r, t + \tau_q) = -k\vec{\nabla} T(r, t + \tau_T), \quad (3)$$

where rho is the tissue density, c is the specific heat capacity, T is the temperature, Qb is the heat generated by blood perfusion, and Qm is the metabolic heat production. To develop a three-phase lag equation, the dual-phase lag equation was combined with the energy balance equation, resulting in the following expression:

$$\left(1 + \tau_q \frac{\partial}{\partial t}\right)\left(\rho c \frac{\partial^2 T}{\partial t^2} - \dot{Q_b} - \dot{Q_m}\right) = \left(k^* + (k + k^* \tau_v)\frac{\partial}{\partial t} + k\tau_T \frac{\partial^2}{\partial t^2}\right)\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right), \quad (4)$$

To solve the equation, initial conditions and boundary conditions were created by Kumar and Kaur. For the initial conditions the temperature profile in both dimensions is equal to the surface temperature of the skin denoted as Tw [4]. There is no change in temperature in the tissue at time zero.

$$T(x, y, 0) = T_w, \quad (8)$$

$$\frac{\partial T(x,y,0)}{\partial t} = 0, \quad (9)$$

$$\frac{\partial^2 T(x,y,0)}{\partial t^2} = 0. \quad (10)$$

The tissue also possess symmetry:

$$-k\frac{\partial T(x,L,t)}{\partial x} = 0, -k\frac{\partial T(L,y,t)}{\partial y} = 0 \quad (11)$$

Once the initial conditions were set the boundary conditions were defined based on the different possible heat ablation surgery techniques. The equation was transformed into the general form below:

$$A_1 \frac{\partial T(0,y,t)}{\partial x} + B_1 T(0, y, t) = f_1(y, t), \quad (12)$$

$$A_2 \frac{\partial T(x,0,t)}{\partial y} + B_2 T(x, 0, t) = f_2(x, t). \quad (13)$$

The first boundary condition, also known as the Dirichlet boundary condition, has the temperature at the boundary explicitly defined and fixed [4]. There is a constant temperature at the boundary regardless of the heat flux or other conditions. This boundary condition applies to many cryosurgical probes that have a constant temperature being applied to the surface of the tissue. Boundary condition of first kind:

$$A_1 = 0, B_1 = 1, f_1(y,t) = T_w, \qquad A_2 = 0, B_2 = 1, f_2(x,t) = T_w \tag{14}$$

In the second boundary condition there is not a constant temperature but alternatively a constant heat flux. This implies that temperature at the boundary is changing as heat transport is occurring but there is constant flux of energy going in the boundary [4]. This is known as the Neumann boundary condition. This simulates many heat ablation techniques in which a constant source of energy, such as high frequency lasers, are used to deliver heat to the surface of tissues. Boundary condition of second kind:

$$A_1 = -k, B_1 = 0, f_1(y,t) = q_w, \qquad A_2 = -k, B_2 = 0, f_2(x,t) = q_w \tag{15}$$

The final boundary condition used for the model is the Robin boundary condition and accounts for the convective heat transfer between the surface of the tissue and outside environment [4]. The boundary condition depends on a reference temperature Tp which is typically the temperature difference between the surface and outside fluid or environment. Boundary condition of third kind:

$$A_1 = -k, B_1 = h, f_1(y,t) = hT_p, \qquad A_2 = -k, B_2 = h, f_2(x,t) = hT_p \tag{16}$$

This new model demonstrated an enhanced ability to predict experimental results more accurately and efficiently compared to previous models [4]. Kumar and Kaur further elaborated on the solution methodology by transforming the problem into a dimensionless form. By applying finite difference methods, the problem was converted into an ordinary matrix differential equation and the Legendre wavelet Galerkin method was employed to solve the equation, enabling a novel and efficient solution for temperature distribution. Through these advancements, the three phase lag model provides a more comprehensive understanding of heat transfer in heterogeneous tissues, improving the precision and effectiveness of thermal treatments like heat ablation and cryosurgery.

Methods

Improvement 1:

To enhance the model, the energy balance equation can be refined to incorporate heat loss due to the evaporative cooling of sweat on the skin's surface. Sweating is an automatic physiological response to elevated body temperatures and is a highly dynamic process that helps maintain homeostasis. This process occurs continuously throughout the day, not only in response to heat but also due to emotional stress and environmental conditions. Evaporative cooling functions when sweat produced by the sweat glands accumulates on the skin's surface and subsequently evaporates. As the sweat evaporates, it absorbs heat from the skin, resulting in a cooling effect and a decrease in temperature on the surface [7]. This mechanism is particularly effective in regulating body temperature under conditions of high ambient temperature or during physical exertion, where it plays a crucial role in preventing overheating [7]. By including the effects of evaporative cooling in the energy balance equation, the model can more accurately reflect the complex interactions between heat transfer, sweat production, and evaporation. This improvement allows for a better understanding of how the body dissipates heat, especially under varying environmental and physiological conditions.

To incorporate this, a simplified model of the cooling effect of sweat was implemented using a framework developed by Aihua and Jie [8]. In this model, it is assumed that sweat generation remains constant over time after reaching a specific threshold. At resting conditions, the body begins to produce sweat at a threshold temperature of 37.38°C [9]. Tanabe and Kobayashi observed the different sweat accumulation and production in different segments of the body but for this application, a small scale analysis will be performed using the parameters of the forearm [10]. Heat loss due to evaporation is represented as a relationship between the skin surface area (*Si*), the water vapor pressure on the skin's surface (*Psk*), the environmental pressure (*Pe*), and the skin's water vapor resistance of air layer (*Rva*). This relationship can be expressed

$$Q_{sweat} = \frac{Si\,(Psk - Pe)}{Rva} \tag{17}$$

$$R_{va} = \frac{2430 * 1000}{0.1353 * \sqrt{0.11 + 0.45Vw + Ve}} \tag{18}$$

$$P_{sk} = \frac{(mrsw * Resk * Rva) + (Psat * Tsk * Rva) + (Pe * Resk)}{Resk + Rva} \tag{19}$$

Additionally, the water vapor resistance is influenced by the velocity of external air (Ve) and walking speed (Vw), and water vapor resistance of the skin (Resk). The water vapor pressure on the skin is dependent on parameters specific to each body part, such as the baseline regulatory sweating rate and the initial temperature of the skin. For this simulation, the surface area and regulatory sweating rate will be chosen to best show the characteristics of the model. Additionally, the tissue will be stationary and Ve and Vw terms will be set to equal zero. Lastly, the heat loss due to evaporation can be added to the energy balance equation.

$$\left(1 + \tau_q \frac{\partial}{\partial t}\right)\left(\rho c \frac{\partial^2 T}{\partial t^2} - \dot{Q_b} - Q_m + \dot{Q_{sweat}}\right) = \left(k^* + (k + k^* \tau_v)\frac{\partial}{\partial t} + k\tau_T \frac{\partial^2}{\partial t^2}\right)\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right), \quad (20)$$

Improvement 2:

Another enhancement to the model was to incorporate an improved boundary condition to better simulate the conduction that occurs between two different materials with different thermal conductivity values. Conduction is the process by which heat or electricity is directly transmitted through a substance where there is a difference of temperature between adjoining regions without movement of material. This has been tested by multiple groups, particularly with Yuan who utilized the bio-heat Pennes' equation to determine hyperthermia rates with a porous model equation [11]. While we are not using a porous model, we can still use the equation given in the original paper as a starting point for further analysis. One thing that this group and many others didn't do is take into account the thermal conductivity of the second material or the wall in this case. This is important to consider in bio-heat transfer since two materials in the body will almost never have the same thermal conductivity. This can occur in cases of tumor formation and even in the cases of extreme freezing or heating such as through a stove or a piece of dry ice as shown by Yuan's research, displaying an immediate need to incorporate this improved boundary condition into the model to better simulate skin tissue heating and cooling.

In [12], Slota makes an attempt at solving the heat equation with the boundary condition for the fourth kind which is continuous heat flux and temperature. This was a proposed solution based on his variational iteration method in which the sequence convergent or the solution can be derived. Other researchers have utilized this method particularly with Wazwaz who employed the method in exact solutions of Laplace and wave equation to model heat transfer[13][14]. While this equation was not used directly in a bio-heat scenario, an implementation of this extra boundary condition could still be applied to our use case as well. By including this boundary condition to account for conduction between the wall and the human skin tissue, the model can more accurately derive temperature gradients of human skin tissue when in contact with another object of a differing thermal conductivity. This boundary condition is shown below which describes both continuous temperature and heat flux at the shared common boundary, also known as a fourth kind boundary condition. ku and kv are the thermal conductivity values and u and u and v are temperature and t and x refer to time and spatial location respectively.

$$u(0,t) = v(0,t), t \in [0, t^*), \quad (21)$$

$$-k_u \frac{\partial u(x,t)}{\partial x}\Big|_{x=0} = -k_v \frac{\partial v(x,t)}{\partial x}\Big|_{x=0}, t \in [0, t^*), \quad (22)$$

This acts as a mixed boundary condition which defines a boundary value problem in which the solution of the given equation is required to satisfy different boundary conditions on different parts of the boundary of the domain where the condition is stated. This differs from a robin boundary condition which requires a linear combination of the Dirichlet and Neumann boundary value conditions to be satisfied on the whole boundary of a given domain. We then take equation (13) and perform finite difference approximation to acquire the temperature gradient at the boundaries. From here we substitute in the approximations and solve for T[0,j]. This will now serve as our conduction boundary condition that utilizes ku and kv and simulate the two different materials with differing thermal conductivities by applying these boundary conditions to the human tissue so that the shared boundary is both the (x,0) and (y,0) values. The default initial and symmetrical boundary conditions were used as in the original paper and applied to their respective boundaries. The 3rd boundary condition for constant heat coefficient was repurposed from the original paper. From here we can use established equations in order to utilize the h variable and create a boundary condition to simulate convective heat transfer.

From Newton's law of cooling:

$$Q = h(T - Tl) \tag{23}$$

And from the thermal conduction equation:

$$Q = -k\Delta T \tag{24}$$

To receive:

$$-k\Delta T = h(T - Tl) \tag{25}$$

We can combine these two equations since they share the same variable Q and perform finite difference approximations and solve for T[0,i]. These boundaries are then applied to the T(x,L,t) and T(L,y,t) locations and are applied in all experiments. In some cases, T(x,0,t) and T(0,L,t) will also have a convective boundary condition depending on the experiment as well. From here, we can apply our boundary conditions to the model to properly simulate heat transfer through conduction.

In order to test if this model improves on the simulation of conduction between two different materials, 3 experiments were enacted with established and variable parameters to see how different thermal conductivity values for the human skin tissue affect heat transfer in the case of conduction with a foreign object. 4 values were chosen for k: 0.3 W/m°C, 0.625 W/m°C, and 1 W/m°C and 1.5 W/m°C. 0.625 W/m°C was used in the original paper and was found through established research [4]. Each of these were used to calculate the ratio between the skin tissue and wall thermal conductivity values which is set to 0.625 W/m°C for the wall. From here, 3 experiments were enacted all testing for different boundary conditions with the same initial and symmetrical conditions used in the original paper as shown below:

Table 1:

Experiments Performed:

No.	Experiment	Bound. con.	Wall	4th Bound.	Time(s)	Tot. Time(s)
A.	Timed Convection	3rd	False	False	5	10
B.	Timed Direct	1st	True	False	5	10
C.	Timed Conduction	4th	True	True	5	10

1st: Dirichlet - Continuous temperature applied to shared boundary
3rd: Robin - Continuous heat coefficient applied to shared boundary
4th: Mixed - Continuous temperature and heat flux applied to shared boundary

Table 2:
Parameters and values used in the improved model:

Parameter	Values(unit)	Source
ρ	1000 kg/m^3	[4]
c	4000 J/kg°C	[4]
k	0.3, 0.625, 1, 1.5 W/m°C	[4]
k_u	0.625 W/m°C	[4]
k^*	0.1 W/m°C/s	*
h	4.5 W/m^2°C	[15]
w_b	0.0098 1/s	[4]
ρ_b	1056 kg/m^3	[4]
c_b	4000 J/kg°C	[4]
Q_{m0}	50.56W/m^3	[4]
T_b	37°C	[4]
T_0	37°C	[4]
T_l	37°C	[4]
T_w	100°C	**
τ_q	600 s	[4]
τ_T	300 s	[4]
τ_v	100 s	[4]
Pe	101325 Pa	[8]
Vw	0 m/s	[8]
Ve	0 m/s	[8]
Resk	33.6 m^{2}sPa/kg	[8]
Psat	4500 Pa	[8]
mrsw	1 g/m^{2}s	*
Si	0.025 m^2	**

* No exact value could be found within studied literature. Choose a value similar to other values.
** Value chosen by the team. Not from any literature.

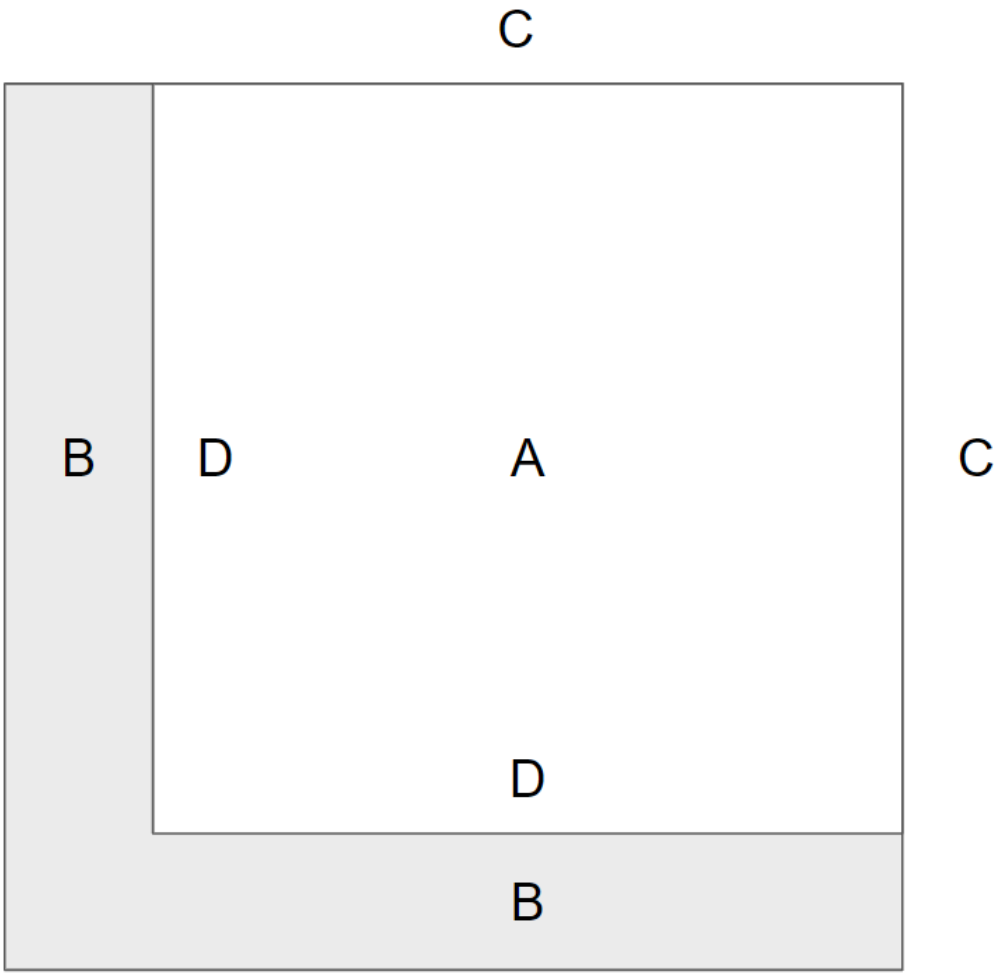

Figure 1: Schematic for skin tissue model. (A) 5x5 cm patch of human skin tissue. (B) Secondary material or wall. (C) Robin boundary condition placement for all experiments. (D) Variable boundary condition implementing Robin, Mixed, or Dirichlet for respective experiments.

Each of these experiments looks to explore how the temperature gradient changes when subjected to different boundary conditions, and heating times. The 3 experiments that we wish to perform will help us analyze the minute differences, if any, that are present when the tissue undergoes timed heating with two different materials with different thermal conductivities. These experiments involve applying heating to skin tissue for 5 seconds at 100°C under these conditions: Timed direct heating does not apply 4th boundary condition to the shared boundary and only applies a dirichlet (1st) boundary condition of 100°C to the shared boundary. Timed conduction heating applies a mixed (4th) boundary condition to the shared boundary for the same time and temperature. Timed convection applies a robin (3rd) boundary condition to the shared boundary of the same time and temperature. After the model terminates after 10 seconds, we then acquire the heat profile and heat distribution of the skin tissue to determine if the new boundary conditions had any effect on the simulating of bio-heat transfer between two separate materials of differing thermal conductivities.

Results:

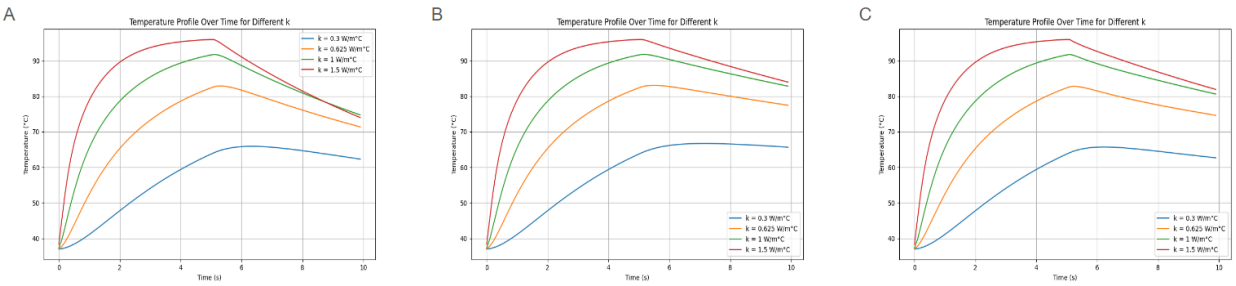

Figure 2: Heat distribution plots for all 3 experiments. (A) Plot for timed convection. (B) Plot for timed direct heating. (C) Plot for timed conduction.

In these figures we can see that the model that uses convection for the shared boundary loses heat the fastest. This is in contrast to the (B) and (C) graphs which use the 1st and 3rd boundary conditions respectively which have considerably slower cooling times. It can also be seen that the 4th boundary condition produces a higher rate of initial temperature decline within the first half of a second after the wall heat is turned off. This is seen in the dip that is produced in graph (C) at around the 5.5 second mark which is absent from graph (B) indicating that the inclusion of the 4th boundary condition better simulates heat being transferred to and from the wall after it is turned off. The rate of heat decline is also slightly steeper when compared to graph (B) indicating that loss of heat through conduction does play a role in heat transfer when there is a shared boundary between two materials with differing thermal conductivity values.

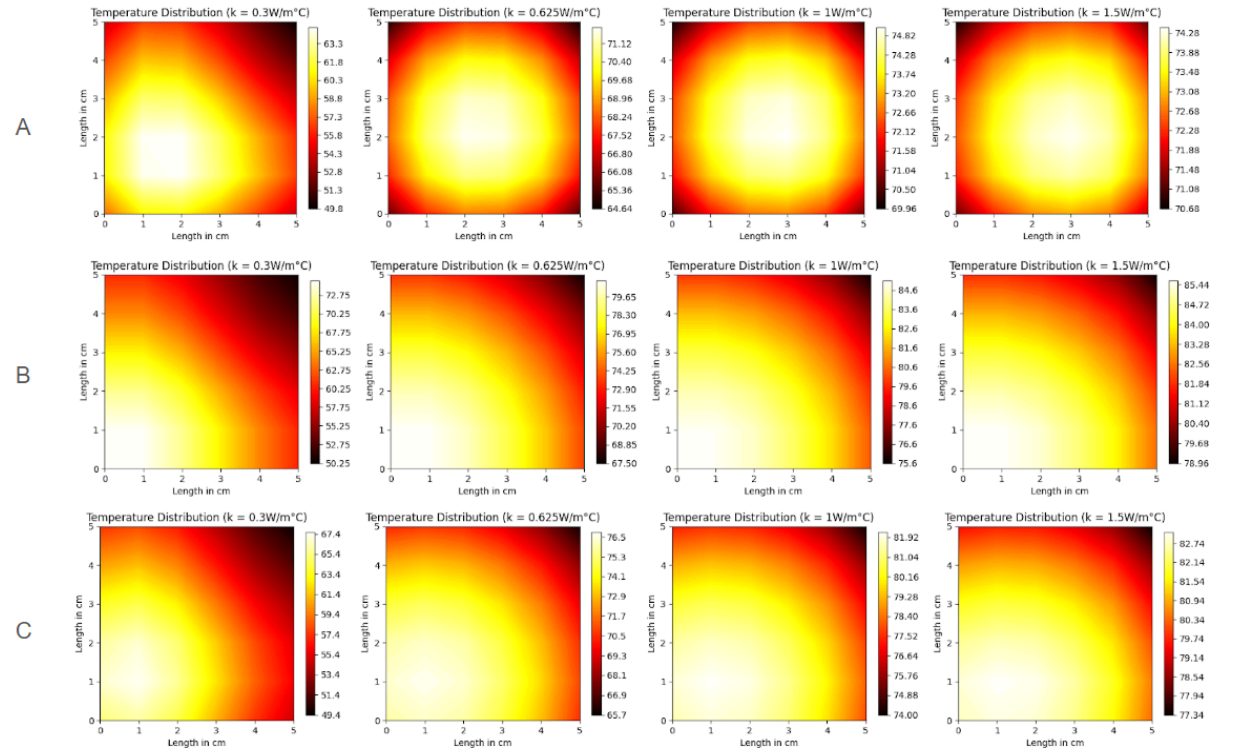

Figure 3: Heat distribution plots for all 3 experiments. (A) Plot for timed convection. (B) Plot for timed direct heating. (C) Plot for timed conduction.

The above figures display the temperature distribution for the 5 x 5 cm patch of human skin tissue. Here we can see that lower k values across all experiments produce the largest gradient of heat while the higher k values produce smaller gradients. Experiment (A) shows that using a convection or Robin (3rd) boundary for the shared boundary causes an even heat distribution across the entire patch. Experiments (C) and (B) display a shifted distribution due to the fact that the wall is using a mixed (4th) or dirichlet (1) boundary condition respectively. The main difference we see between experiments (C) and (B) is that (B) is more linear in its distribution while (C) produces a more radial effect that is consistently cooler in temperature and with smaller total heat gradients. This is due to the fact that the patch is losing heat through conduction to the wall which produces the curvature around the shared boundary compared to the dirichlet boundary condition that experiment (B) employs that enforces a continuous temperature across the boundary, disregarding heat transfer phenomena that could be occurring.

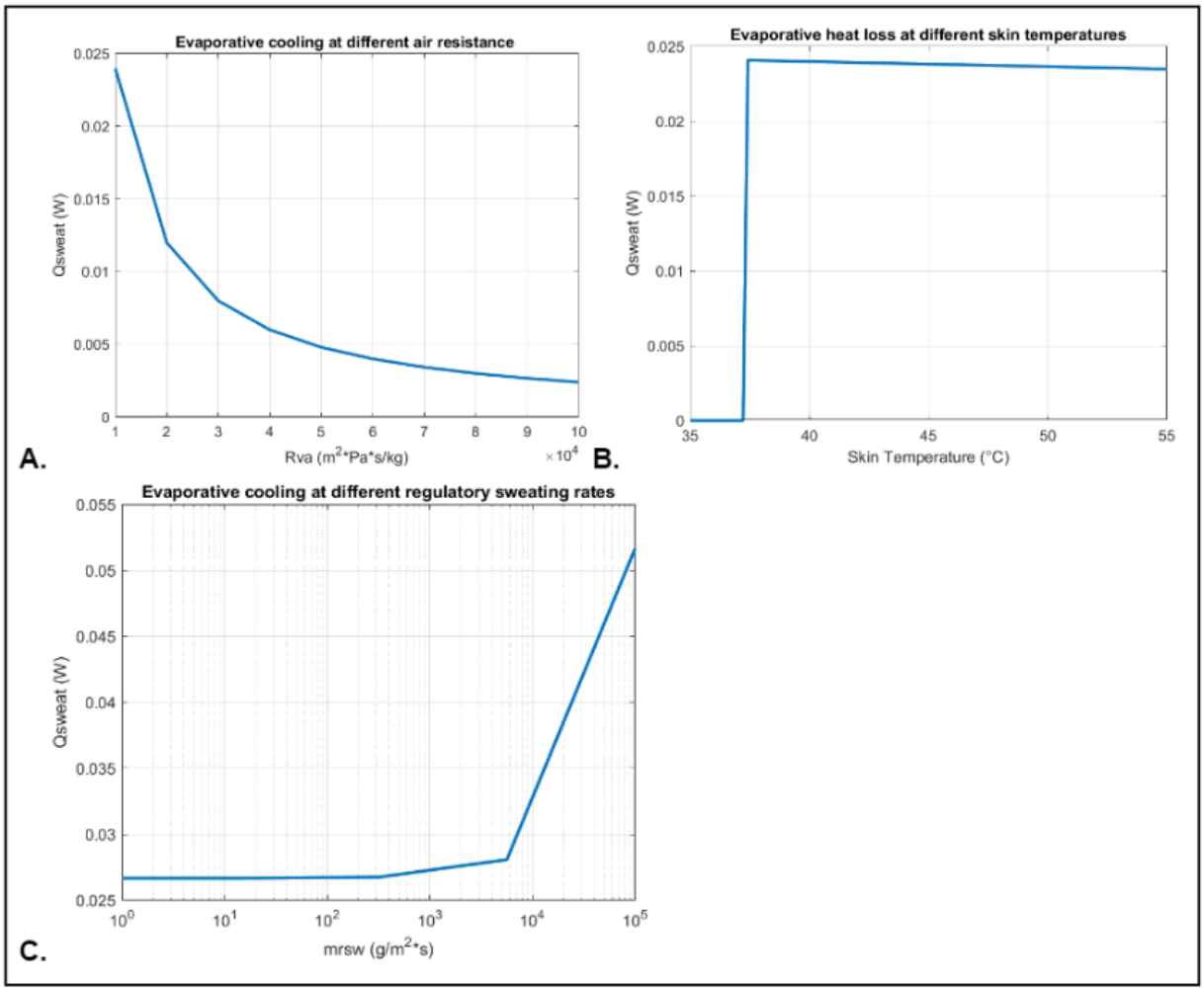

Figure 4: Effect of evaporative cooling in skin tissue at different air resistances (A), skin temperatures (B), and regulatory sweating rate (C).

The heat loss due to evaporative cooling is influenced by several factors, primarily the skin temperature and the water vapor resistance of the air as shown in Figure 4. When examining various water vapor resistance values, the energy lost due to evaporative cooling exhibits an exponential decrease as resistance increases. Similarly, increasing the skin's regulatory sweating rate results in a greater amount of heat loss due to evaporative cooling. The overall energy and heat lost to evaporative cooling remain zero until a threshold temperature is reached. Beyond this threshold, heat loss remains relatively constant, with minimal decrease, at higher temperatures. Heat loss from sweat evaporation is several magnitudes lower than metabolic heat production and blood perfusion heat production.

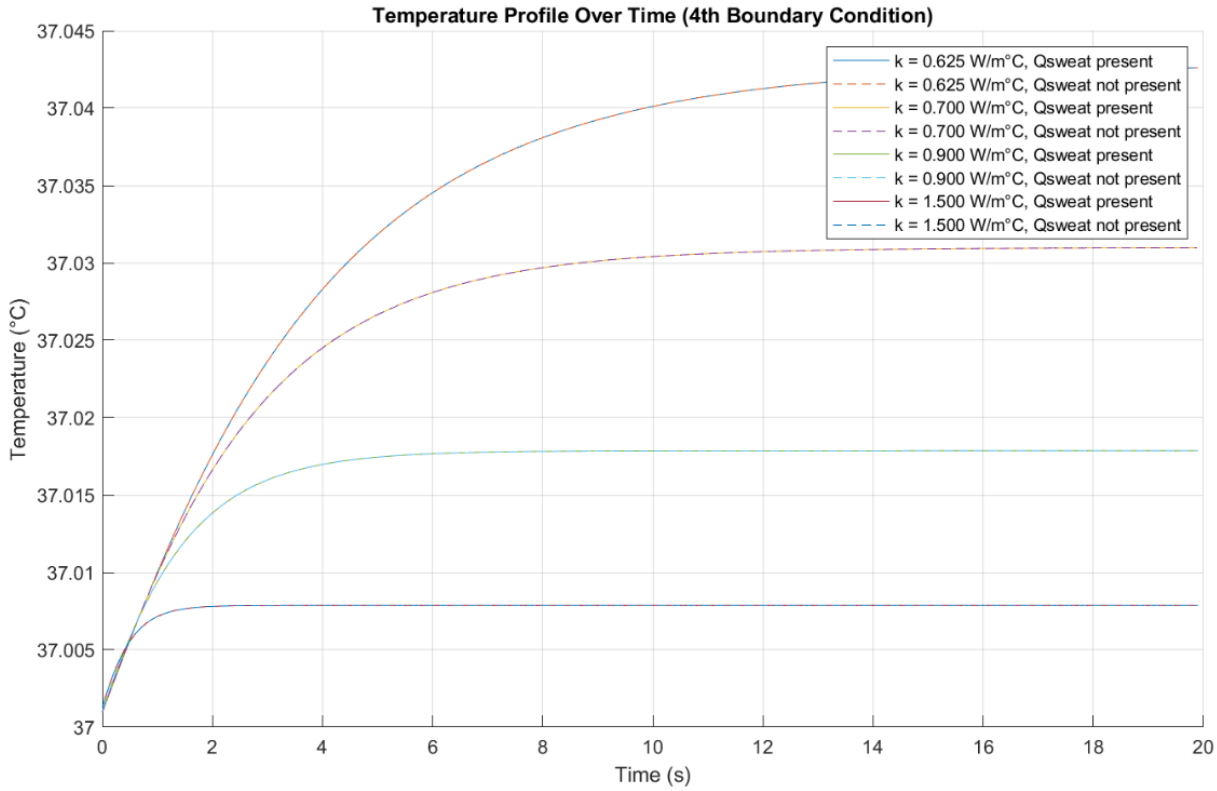

Figure 5. Effect of heat loss due to cooling on temperature profile in skin tissue with fourth boundary condition.

When adding the heat loss associated with evaporative cooling due to sweat evaporation there are minimal changes to the temperature profile in the fourth boundary condition. In the energy balance equation the value of heat loss due to sweat (Qsweat) is very minimal compared to the other variables and subsequently does not impact the temperature profile curves in a significant way. The temperature of the dotted curves are slightly lower but on an order of magnitude much smaller essentially reproducing the original curve.

Discussion

The addition of evaporative cooling did not significantly impact the temperature profile of the tissue. The heat loss due to sweating was minimal compared to the effects of metabolic heat generation and blood perfusion rate, which play a more substantial role. However, this outcome heavily depends on the experimental setup and model parameters. The energy and heat loss due to sweat is influenced by numerous factors. For simplification, the model assumed no tissue movement and no external sources of air. Among the parameters affecting Qsweat, the water vapor resistance of the air Rva was the most impactful. A higher Rva means greater resistance, resulting in less heat loss because sweat cannot evaporate as easily to cool the body. While the regulatory sweat rate also affects Qsweat, its impact is smaller and was negligible in this experiment. The sweat rate is directly related to the difference in water vapor pressure of the skin, Psk, and water vapor pressure of the environment. However, the pressure of the environment is typically much larger than Psk and thus has a small effect on Qsweat.

Sweat accumulation is influenced not only by tissue properties but also by environmental conditions. The relationship between water vapor resistance of the air and skin indicates significant variations under different conditions, such as temperature, pressure, and humidity. Therefore, although the current setup shows a minimal effect, adjusting environmental conditions can have a more substantial impact. In this model, the regulatory sweating rate was simplified to be temperature-dependent, occurring only above a specific threshold. However, this may not fully model the complex physiological conditions. It does not account for complex heat and mass transfer dynamics behavior post-threshold such as how quickly it takes for sweat to evaporate and what percentage of sweat reaches the energy threshold for evaporation. Additionally, the model was simplified by excluding factors like external clothing, ion concentration in sweat affecting vapor resistance, heterogeneity and type of sweat gland types, and different body segments. Including these factors could provide a more robust understanding of the cooling effect.

The production and evaporation of sweat had a limited impact on the model also due to the small surface area that was evaluated. Metabolic heat generation and blood perfusion rate depend on blood-specific parameters such as density, temperature, specific heat, perfusion rate coefficient, and tissue density and will remain constant regardless of surface area and body segment. However, expanding sweat production to a larger surface could significantly enhance the cooling effect.

Lastly, depending on the type of surgical technique used, sweat accumulation could have a much larger effect in clinical applications and the design of devices. In heat ablation, if a laser is directed at tissue, excessive sweat could attenuate, defocus, or refract the beam, potentially leading to misalignment and the need for higher power. This can lead to damage to the surrounding tissue and less accuracy. Understanding these interactions is crucial for optimizing surgical outcomes and minimizing unintended consequences.

The addition of a 4th boundary condition was determined to have had a noticeable effect on the minimum temperature that can be acquired given a set amount of time. As we can see, thermal conductivity plays a major role in determining heat values, especially as we reach the extreme end for thermal conductivity values. Not only does the thermal conductivity of the skin tissue needs to be taken into account, but the thermal conductivity of the heating material as

well. This is expected since heat, especially if the source is turned off, will dissipate quickly through convection or conduction, even if the shared boundary temperature is very hot. In the case of experiment (C) we see that the included mixed boundary condition causes heat to dissipate from the skin to the wall which causes slightly quicker cooling. This is noticed in the curvature of the heat distribution which is slightly lower at the shared boundary. This is in contrast to experiment (B) which does not display this curvature at the shared boundary. This is understandable since a continuous temperature is employed across the shared boundary and does not take into account various thermal mechanics that could exist that are not employed by the model.

The main issue arises when the temperature values are overestimated and are determined to be larger than they actually are. As stated in, It is important to capture the actual values of the model in order to make better predictive decisions in cases of extreme burns or hyperthermia [16]. It is also important to make sure the model is as accurate as possible in order to visualize where and how the burns occur and how they shift over time. As seen in the temperature distribution graphs, not only do differing thermal conductivity values play a huge role in heat transfer, but also the boundary conditions set in place determine how the model reacts to heat. In our case differing boundary conditions, in particular comparing (B) and (C), we do see a noticeable difference in how the temperature is distributed which could affect diagnosis and predictive capabilities when actually trying to determine if heat damage was caused to the cells or not.

Across all different thermal conductivity values for experiments (B) and (C), the heat distribution and profile plots both consistently found (B) to be hotter than C and for much longer. (C) was also able to more accurately model the heat flux shift once the heat was turned off as well as seen in the initial sharp decline in the temperature profile model when compared to (B) which has a more steady controlled decline in temperature. These can be compared to (A) which utilizes convection only and we see how much conduction plays a role in determining temperature of human skin. It should also be noted that even (A) does not display an initial sharp decline in temperature which indicates that the heat flux is not as extreme when compared to (C), indicating that conduction is showing unique qualities to cooling when compared to pure convection. It is important to incorporate conductivity and differing thermal conductivity values into the model since real-life applications utilize a wide arrange of materials to either heat or cool materials or human skin tissue.

Overall, it can be stated that the inclusion of an improved mixed boundary condition incorporating two thermal conductivity values is a much needed improvement to the original model. This incorporation better simulates heat transfer and conduction between two different objects and better captures minute differences that can be encountered when performing conduction heating and cooling. These improvements can help researchers and practitioners better analyze and predict cases of life-threatening burns or hyperthermia which in turn can help us improve our bio-heat transfer models and potentially save lives.

Conclusion

In this paper, we have enhanced the mathematical model for heat transfer in tissue, initially developed by Kumar and Kaur. Our model refines Fourier's law of heat conduction by incorporating the effects of three relaxation times under Dirichlet, Neumann, and Robin boundary conditions. We have introduced a fourth boundary condition to investigate the impact of continuous temperature and heat flux at a shared boundary on the tissue's temperature profile. This fourth boundary condition revealed notable differences in temperature profiles, particularly near the surfaces. Additionally, we integrated heat loss due to sweat evaporation into the model, though it had a minimal effect on the temperature profile due to the small surface area used in this simulation. Future work will further characterize different tissue layers to enhance the application and evaluation of cryosurgical and heat ablation procedures.

Reference

[1] Xu, Kecheng, et al. *Modern Cryosurgery for Cancer*. World Scientific, 2012. Google Books, books.google.com/books?hl=en&lr=&id=1nDNIMgjrzgC&oi=fnd&pg=PR5&dq=cryosurgery+cancer+&ots=WLYI7HWZUu&sig=OWun5yNAupfNLpSkIURgs5Dnnks#v=onepage&q=cryosurgery%20cancer&f=false. Accessed 10 June 2024.

[2] Wu, Feng. "Heat-Based Tumor Ablation: Role of the Immune Response." 1 Jan. 2016, pp. 131–153. SpringerLink, https://doi.org/10.1007/978-3-319-22536-4_8. Accessed 14 May 2023.

[3] Seror, O. “Ablative Therapies: Advantages and Disadvantages of Radiofrequency, Cryotherapy, Microwave and Electroporation Methods, or How to Choose the Right Method for an Individual Patient?” *Diagnostic and Interventional Imaging*, vol. 96, no. 6, June 2015, pp. 617–624. ScienceDirect, https://doi.org/10.1016/j.diii.2015.04.007.

[4] Kumar, Mukesh, et al. “Mathematical Modelling and Simulation of Three Phase Lag Bio-Heat Transfer Model during Cancer Treatment.” *International Journal of Thermal Sciences*, vol. 184, 1 Feb. 2023, p. 108002. ScienceDirect, https://doi.org/10.1016/j.ijthermalsci.2022.108002.

[5] Pennes, Harry H. “Analysis of Tissue and Arterial Blood Temperatures in the Resting Human Forearm.” *Journal of Applied Physiology*, vol. 1, no. 2, Aug. 1948, pp. 93–122. PubMed, https://doi.org/10.1152/jappl.1948.1.2.93.

[6] Tzou, D. Y. *Macro- to Microscale Heat Transfer: The Lagging Behavior*. John Wiley & Sons, 18 Sept. 2014. Google Books, books.google.com/books?hl=en&lr=&id=dtuSBAAAQBAJ&oi=fnd&pg=PA255&ots=vf3IN2GpeC&sig=SNm9iu-e1WCahr0AM8l0VxrwLlE#v=onepage&q&f=false. Accessed 11 June 2024.

[7]Osilla, Eva V., et al. “Physiology, Temperature Regulation.” PubMed, StatPearls Publishing, 2023, www.ncbi.nlm.nih.gov/books/NBK507838/#:~:text=Heat%20loss%20via%20evaporation%20of.

[8] Aihua, Mao, et al. “Numerical Simulation of Multiscale Heat and Moisture Transfer in the Thermal Smart Clothing System.” *Applied Mathematical Modelling*, vol. 40, no. 4, 1 Feb. 2016, pp. 3342–3364. ScienceDirect, https://doi.org/10.1016/j.apm.2015.10.038. Accessed 11 June 2024.

[9] Mekjavić, Igor B., and John Bligh. “Core Threshold Temperatures for Sweating.” *Canadian Journal of Physiology and Pharmacology*, vol. 67, no. 9, 1 Sept. 1989, pp. 1038–1044. NRC Research Press, https://doi.org/10.1139/y89-164. Accessed 2 Feb. 2023.

[10] Tanabe, Shin-ichi, et al. “Evaluation of Thermal Comfort Using Combined Multi-Node Thermoregulation (65MN) and Radiation Models and Computational Fluid Dynamics (CFD).”

Energy and Buildings, vol. 34, no. 6, July 2002, pp. 637–646. ScienceDirect, https://doi.org/10.1016/s0378-7788(02)00014-2. Accessed 1 Oct. 2022.

[11] Yuan, Ping. “Numerical Analysis of an Equivalent Heat Transfer Coefficient in a Porous Model for Simulating a Biological Tissue in a Hyperthermia Therapy.” *International Journal of Heat and Mass Transfer*, vol. 52, no. 7-8, Mar. 2009, pp. 1734–1740. ScienceDirect, https://doi.org/10.1016/j.ijheatmasstransfer.2008.09.033. Accessed 3 Aug. 2021.

[12] Słota, Damian. “Exact Solution of the Heat Equation with Boundary Condition of the Fourth Kind by He’s Variational Iteration Method.” *Computers & Mathematics with Applications*, vol. 58, no. 11-12, Dec. 2009, pp. 2495–2503. ScienceDirect, https://doi.org/10.1016/j.camwa.2009.03.052. Accessed 9 June 2022.

[13] Wazwaz, Abdul-Majid. “The Variational Iteration Method for Exact Solutions of Laplace Equation.” *Physics Letters A*, vol. 363, no. 4, Apr. 2007, pp. 260–262. ScienceDirect, https://doi.org/10.1016/j.physleta.2006.11.014. Accessed 10 Sept. 2022.

[14] Wazwaz, Abdul-Majid. “The Variational Iteration Method: A Reliable Analytic Tool for Solving Linear and Nonlinear Wave Equations.” *Computers & Mathematics with Applications*, vol. 54, no. 7-8, 1 Oct. 2007, pp. 926–932. ScienceDirect, https://doi.org/10.1016/j.camwa.2006.12.038. Accessed 2 Nov. 2023.

[15] de Dear, R., Arens, E., Hui, Z., et al. “Convective and Radiative Heat Transfer Coefficients for Individual Human Body Segments.” *International Journal of Biometeorology*, vol. 40, 1997, pp. 141–156. SpringerLink, https://doi.org/10.1007/s004840050035.

[16] Aihua, M., Jie, L. et al. “Numerical Simulation of Multiscale Heat and Moisture Transfer in the Thermal Smart Clothing System.” *Applied Mathematical Modelling*, vol. 40, no. 4, 1 Feb. 2016, pp. 3342–3364. ScienceDirect, https://doi.org/10.1016/j.apm.2015.10.038.

Code:

Python Code for Figures 2 and 3:

```
import numpy as np
import matplotlib.pyplot as plt

# Constants and parameters
rho = 1000  # Tissue density (kg/m^3)
c = 4000  # Specific heat of tissue (J/kg°C)
k_list = [0.3, 0.625, 1, 1.5]  # Thermal conductivity of tissue (W/m°C)
k_star = 0.1 # Additional thermal conductivity term (W/m°C/s)
h = 4.5  # Heat transfer coefficient for Robin boundary condition (W/m^2°C) -
Typical for large blood vessels
wb = 0.0098  # Blood perfusion rate coefficient (1/s) - Typical for skin tissue
rho_b = 1056  # Density of blood (kg/m^3)
cb = 4000  # Specific heat of blood (J/kg°C)
Qm0 = 50.65  # Metabolic heat generation (W/m^3)
Tb = 37  # Temperature of arterial blood (°C)
T0 = 37  # Initial temperature of the body (°C)
Tl = 37  # Temperature of Tissue (°C)
Tw = 100 # Fixed temperature at left and bottom boundary
Tw0 = 37
Lx = 0.05  # Length of the skin tissue in x direction (m)
Ly = 0.05  # Length of the skin tissue in y direction (m)
dx = 0.01  # Space step in x direction (m)
dy = 0.01  # Space step in y direction (m)
dt = 0.1  # Time step (s)
wall_temp_duration = 50 # Number of Wall Temperature 'ON' time steps
remove_wall_after = False # Remove Wall if True, disables fourth boundry condition
on boundry border
fourth_boundary_on = True
time_steps = 100  # Number of time steps
tau_q = 600  # Relaxation time due to heat flux (s)
tau_T = 300  # Relaxation time due to temperature gradient (s)
tau_v = 100  # Relaxation time due to thermal displacement (s)
ambient_temp = 37 # Ambient temperature of space without initialized wall temp (°C)

# Constants for the second material (assuming for the boundary condition of fourth
kind)
ku = 0.625   # Thermal conductivity of left material (W/m°C)

DOI_1 = 'https://doi.org/10.1016/j.ijthermalsci.2022.108002'

if wall_temp_duration == time_steps:
    remove_wall_after = False
```

```
# Boundry between wall and tissue
def wall_boundary(T_array, wall_temp):
    T_array[-1, :] = wall_temp  # x = 0 (bottom boundary)
    T_array[:, 0] = wall_temp  # y = 0 (left boundary)

# Symmetrical boundary conditions from eq. 16
def symmetric_boundary(T_array):
    T_array[0, :] = T_array[1, :]  # x = Lx
    T_array[-1, :] = T_array[-2, :]  # x = 0
    T_array[:, 0] = T_array[:, 1]  # y = 0
    T_array[:, -1] = T_array[:, -2]  # y = Ly

# Convective boundary condition to introduce a constant heat coefficient
def convective_boundary(T_array):
    T_array[0, :] = (h * dx * Tl + k * T_array[1, :]) / (h * dx + k)  # x = Lx
    T_array[:, -1] = (h * dx * Tl + k * T_array[:, -2]) / (h * dx + k)  # y = Ly

# Fourth boundary using constant temperature and heat flux for two thermal
conductivity terms from secondary paper to model conduction.
def fourth_boundary(T_array):
    if fourth_boundary_on is True:
        T_array[-1, :] = T_array[-2, :] - (k / ku) * (T_array[-2, :] - T_array[-3,
:])  # x = 0
        T_array[:, 0] = T_array[:, 1] - (ku / k) * (T_array[:, 1] - T_array[:, 2])
# y = 0

# Third boundary using constant heat coefficient
def third_boundary(T_array):
    T_array[-1, :] = (h * dx * Tl + k * T_array[-2, :]) / (h * dx + k)  # x = 0
    T_array[:, 0] = (h * dx * Tl + k * T_array[:, 1]) / (h * dx + k)  # y = 0

# Discretization
x = np.arange(0, Lx + dx, dx)
y = np.arange(0, Ly + dy, dy)
nx = len(x)
ny = len(y)

# Initialize time derivatives
dTdt_initial = np.zeros((nx, ny))  # Initial first time derivative of temperature
d2Tdt2_initial = np.zeros((nx, ny))  # Initial second time derivative of
temperature

# Initialize temperature fields
T_initial = np.ones((nx, ny)) * T0  # Initialize entire temperature field to T0
T_new_initial = np.ones((nx, ny)) * T0  # Initialize entire temperature field to T0
```

```
# Plotting the results
time = np.arange(0, time_steps * dt, dt)
plt.figure(figsize=(10, 6))

for k in k_list:
    # Copy time derivatives
    dTdt = dTdt_initial.copy()
    d2Tdt2 = d2Tdt2_initial.copy()

    # Copy temperature fields
    T = T_initial.copy()
    T_new = T_new_initial.copy()

    # Check if wall is initialized
    if wall_temp_duration > 0:
        wall_boundary(T, Tw)
        wall_boundary(T_new, Tw)

    # Else use ambient temperature of air
    elif wall_temp_duration == 0 and remove_wall_after is False:
        wall_boundary(T, Tw0)
        wall_boundary(T_new, Tw0)

    elif wall_temp_duration == 0 and remove_wall_after is True:
        wall_boundary(T, ambient_temp)
        wall_boundary(T_new, ambient_temp)

    # Store temperature profile at each time step
    temperature_profile = []

    # Time integration
    for t in range(time_steps):
        for i in range(1, nx - 1):
            for j in range(1, ny - 1):
                # From eq. 5
                Qm = Qm0 * (1 + (Tl - T0) / 10)

                #From eq. 6
                Qb = wb * rho_b * cb * (Tb - Tl)

                # Discretization using finite difference method
                d2Tdx2 = (T_new[i + 1, j] - 2 * T_new[i, j] + T_new[i - 1, j]) / dx
** 2
                d2Tdy2 = (T_new[i, j + 1] - 2 * T_new[i, j] + T_new[i, j - 1]) / dy
** 2

                # From eq. 4
```

```
                dTdt[i, j] = (k * (d2Tdx2 + d2Tdy2) + Qb + Qm) / (rho * c)

                # Derivative of eq. 4 with k* integrated
                d2Tdt2[i, j] = (k_star * (d2Tdx2 + d2Tdy2)) / (rho * c)

                # Substitute finite differences from discretization into eq. 7 and
solve for T_n+1
                T_new[i, j] = T[i, j] + dt * (dTdt[i, j] + tau_q * dTdt[i, j] -
tau_T * d2Tdt2[i, j] + (k + k_star * tau_v) * dTdt[i, j])

        # Reapply fixed temperature boundary condition at each time step
        if t < wall_temp_duration:
            wall_boundary(T_new, Tw)

        # Symmetric boundary conditions (Neumann conditions with zero gradient)
        symmetric_boundary(T_new)

        # Reapply fixed temperature boundary condition at each time step
        if t < wall_temp_duration:
            wall_boundary(T_new, Tw)

        convective_boundary(T_new)

        # Robin boundary condition on all boundaries (convective)
        if t >= wall_temp_duration and remove_wall_after is True:
            print("both off")
            third_boundary(T_new)

        elif t >= wall_temp_duration and remove_wall_after is False:
            print("off")
            fourth_boundary(T_new)

        # Apply 4th boundry condition at boundry border between wall and tissue
        elif t < wall_temp_duration or remove_wall_after is False:
            print("on")
            fourth_boundary(T_new)

        elif t < wall_temp_duration or remove_wall_after is True:
            print("on")
            third_boundary(T_new)

        # Reapply fixed temperature boundary condition at each time step
        if t < wall_temp_duration:
            wall_boundary(T_new, Tw)

        # Update temperature
        T = T_new.copy()
```

```
        # Store temperature profile
        temperature_profile.append(T_new.copy())

    # Convert temperature profile to numpy array for easy slicing
    temperature_profile = np.array(temperature_profile)

    # Plot temperature at a specific point over time
    plt.plot(time, temperature_profile[:, nx // 2, ny // 2], label=f'k = {k}
W/m°C')

plt.xlabel('Time (s)')
plt.ylabel('Temperature (°C)')
plt.title('Temperature Profile Over Time for Different k')
plt.legend()
plt.grid(True)
plt.savefig(f'temp_dur-{wall_temp_duration}_removewallafter-{remove_wall_after}_fou
rth-{fourth_boundary_on}.png')
# plt.show()

# Plot heatmaps for each k value
fig, axes = plt.subplots(1, len(k_list), figsize=(18, 6))

for idx, k in enumerate(k_list):
    # Copy time derivatives
    dTdt = dTdt_initial.copy()
    d2Tdt2 = d2Tdt2_initial.copy()

    # Copy temperature fields
    T = T_initial.copy()
    T_new = T_new_initial.copy()

    # Check if wall is initialized
    if wall_temp_duration > 0:
        wall_boundary(T, Tw)
        wall_boundary(T_new, Tw)

    # Else use ambient temperature of air
    elif wall_temp_duration == 0 and remove_wall_after is False:
        wall_boundary(T, Tw0)
        wall_boundary(T_new, Tw0)

    elif wall_temp_duration == 0 and remove_wall_after is True:
        wall_boundary(T, ambient_temp)
        wall_boundary(T_new, ambient_temp)

    # Store temperature profile at each time step
```

```
    temperature_profile = []

    # Time integration
    for t in range(time_steps):
        for i in range(1, nx - 1):
            for j in range(1, ny - 1):
                # From eq. 5
                Qm = Qm0 * (1 + (Tl - T0) / 10)

                #From eq. 6
                Qb = wb * rho_b * cb * (Tb - Tl)

                # Discretization using finite difference method
                d2Tdx2 = (T_new[i + 1, j] - 2 * T_new[i, j] + T_new[i - 1, j]) / dx
** 2
                d2Tdy2 = (T_new[i, j + 1] - 2 * T_new[i, j] + T_new[i, j - 1]) / dy
** 2


                # From eq. 4
                dTdt[i, j] = (k * (d2Tdx2 + d2Tdy2) + Qb + Qm) / (rho * c)

                # Derivative of eq. 4 with k* integrated
                d2Tdt2[i, j] = (k_star * (d2Tdx2 + d2Tdy2)) / (rho * c)

                # Substitute finite differences from discretization into eq. 7 and
solve for T_n+1
                T_new[i, j] = T[i, j] + dt * (dTdt[i, j] + tau_q * dTdt[i, j] -
tau_T * d2Tdt2[i, j] + (k + k_star * tau_v) * dTdt[i, j])

        # Reapply fixed temperature boundary condition at each time step
        if t < wall_temp_duration:
            wall_boundary(T_new, Tw)

        # Symmetric boundary conditions (Neumann conditions with zero gradient)
        symmetric_boundary(T_new)

        # Reapply fixed temperature boundary condition at each time step
        if t < wall_temp_duration:
            wall_boundary(T_new, Tw)

        convective_boundary(T_new)

        # Robin boundary condition on all boundaries (convective)
        if t >= wall_temp_duration and remove_wall_after is True:
            third_boundary(T_new)

        elif t >= wall_temp_duration and remove_wall_after is False:
```

```
            fourth_boundary(T_new)

        # Apply 4th boundry condition at boundry border between wall and tissue
        elif t < wall_temp_duration or remove_wall_after is False:
            fourth_boundary(T_new)

        # Reapply fixed temperature boundary condition at each time step
        if t < wall_temp_duration:
            wall_boundary(T_new, Tw)

        # Update temperature
        T = T_new.copy()

    ax = axes[idx]  # Access the correct subplot
    T_rotated = np.rot90(T, -1)

    contour = ax.contourf(T_rotated, 100, cmap='hot')  # Transpose T for correct
orientation
    fig.colorbar(contour, ax=ax, shrink=0.5)
    ax.set_aspect('equal', 'box')
    ax.set_xlabel('Length in cm')
    ax.set_ylabel('Length in cm')
    ax.set_title(f'Temperature Distribution (k = {k}W/m°C)')

plt.tight_layout()
plt.savefig(f'temp_dur-{wall_temp_duration}_removewallafter-{remove_wall_after}_fou
rth-{fourth_boundary_on}_HM.png')
# plt.show()
```

Matlab Code for figure 4:

```
% Constants and parameters
rho = 1000;  % Tissue density (kg/m^3)
c = 4000;  % Specific heat of tissue (J/kg°C)
k_list = [0.625, 0.7, 0.9, 1.5];  % Thermal conductivity of tissue (W/m°C)
k_star = 1; % Additional thermal conductivity term (W/m°C/s)
h = 4.5;  % Heat transfer coefficient for Robin boundary condition (W/m^2°C)
wb = 0.0098;  % Blood perfusion rate coefficient (1/s)
rho_b = 1056;  % Density of blood (kg/m^3)
cb = 4000;  % Specific heat of blood (J/kg°C)
Qm0 = 50.65;  % Metabolic heat generation (W/m^3)
Tb = 37;  % Temperature of arterial blood (°C)
T0 = 37;  % Initial temperature of the body (°C)
Tl = 37;  % Temperature of Tissue (°C)
Tw = 90; % Fixed temperature at left and bottom boundary
Lx = 0.05;  % Length of the skin tissue in x direction (m)
Ly = 0.05;  % Length of the skin tissue in y direction (m)
```

```
dx = 0.01;  % Space step in x direction (m)
dy = 0.01;  % Space step in y direction (m)
dt = 0.1;  % Time step (s)
wall_temp_duration = 200; % Number of Wall Temperature 'ON' time steps
remove_wall_after = false; % Remove Wall if True, disables fourth boundary
condition on boundary border
time_steps = 200;  % Number of time steps
tau_q = 600;  % Relaxation time due to heat flux (s)
tau_T = 300;  % Relaxation time due to temperature gradient (s)
tau_v = 100;  % Relaxation time due to thermal displacement (s)
ambient_temp = 0; % Ambient temperature of space without initialized wall temp
(°C)
% Constants for the second material
ku = 0.7;  % Thermal conductivity of left material (W/m°C)
% Constants and parameters specific to sweat
Si = Lx * Ly;  % Area of skin surface (m^2)
Pe = 101325;  % Pressure of environment (Pa)
Resk = 33.6;  % Water vapor resistance of the skin (m^2*Pa*s/kg)
Psat = 4500;  % Saturated vapor pressure of the skin (Pa)
mrsw_list = [0.1, 1, 10, 100]; % Different values of mrsw for comparison
Rva_list = linspace(10000, 100000, 10);  % Different values of Rva for
comparison
%% 1. Comparison of Qsweat for different Tsk temperatures (35C - 55C)
% All other parameters held constant. Rva = 10,000, mrsw = 1
T_range = linspace(35, 55, 100);
Qsweat_T = zeros(size(T_range)); %intialize matrix
for i = 1:length(T_range)
   Tsk = T_range(i);
   if Tsk > 37.38
       Psk = (mrsw_list(2) * Resk * Rva_list(1) + Psat * Tsk * Rva_list(1) + Pe
* Resk) / (Resk * Rva_list(1));
       Qsweat_T(i) = -(Si * (Psk - Pe) /(Rva_list(1)));  % Heat loss due to
evaporative cooling of sweat
   else
       Qsweat_T(i) = 0;
   end
end
%plotting
figure;
plot(T_range, Qsweat_T, 'LineWidth', 2);
xlabel('Skin Temperature (°C)');
ylabel('Qsweat (W)');
title('Qsweat vs. Skin Temperature');
grid on;
%% 2. Comparison of Qsweat for different values of mrsw on the log scale
% All other parameters held constant for this loop. Rva = 10,000, Tsk = 40
mrsw_values = logspace(0, 5, 5);
Qsweat_mrsw = zeros(size(mrsw_values));
for i = 1:length(mrsw_values)
```

```
    Tsk = 40;  % Fixed skin temperature
    if Tsk > 37.38
        Psk = -(mrsw_values(i) * Resk * Rva_list(1) + Psat * Tsk * Rva_list(1) +
Pe * Resk) / (Resk * Rva_list(1));
        Qsweat_mrsw(i) = -(Si * (Psk - Pe) / Rva_list(1));  % Heat loss due to
evaporative cooling of sweat
    else
        Qsweat_mrsw(i) = 0;
    end
end
%plotting
figure;
semilogx(mrsw_values, Qsweat_mrsw, 'LineWidth', 2);
xlabel('mrsw (log scale)');
ylabel('Qsweat (W)');
title('Qsweat vs. mrsw (log scale)');
grid on;
%% 3. Comparison of Qsweat for different values of Rva
% All other parameters held constant for this loop. Tsk = 37, mrsw = 1
Tsk = 40;  % Fixed skin temperature
Qsweat_Rva = zeros(size(Rva_list));
for i = 1:length(Rva_list)
    if Tsk > 37.38
        Psk = (mrsw_list(2) * Resk * Rva_list(i) + Psat * Tsk * Rva_list(i) + Pe
* Resk) / (Resk * Rva_list(i));
        Qsweat_Rva(i) = -(Si * (Psk - Pe) / Rva_list(i));  % Heat loss due to
evaporative cooling of sweat
    else
        Qsweat_Rva(i) = 0;
    end
end
%plotting
figure;
plot(Rva_list, Qsweat_Rva, 'LineWidth', 2);
xlabel('Rva');
ylabel('Qsweat (W)');
title('Qsweat vs. Rva');
grid on;
```

Matlab Code for Figure 5:

```
% Constants and parameters
rho = 1000;  % Tissue density (kg/m^3)
c = 4000;  % Specific heat of tissue (J/kg°C)
k_list = [0.625, 0.7, 0.9, 1.5];  % Thermal conductivity of tissue (W/m°C)
k_star = 1; % Additional thermal conductivity term (W/m°C/s)
h = 4.5;  % Heat transfer coefficient for Robin boundary condition (W/m^2°C)
wb = 0.0098;  % Blood perfusion rate coefficient (1/s)
```

```
rho_b = 1056;  % Density of blood (kg/m^3)
cb = 4000;  % Specific heat of blood (J/kg°C)
Qm0 = 50.65;  % Metabolic heat generation (W/m^3)
Tb = 37;  % Temperature of arterial blood (°C)
T0 = 37;  % Initial temperature of the body (°C)
Tl = 37;  % Temperature of Tissue (°C)
Tw = 90; % Fixed temperature at left and bottom boundary
Lx = 0.05;  % Length of the skin tissue in x direction (m)
Ly = 0.05;  % Length of the skin tissue in y direction (m)
dx = 0.01;  % Space step in x direction (m)
dy = 0.01;  % Space step in y direction (m)
dt = 0.1;  % Time step (s)
time_steps = 200;  % Number of time steps
tau_q = 600;  % Relaxation time due to heat flux (s)
tau_T = 300;  % Relaxation time due to temperature gradient (s)
tau_v = 100;  % Relaxation time due to thermal displacement (s)
ambient_temp = 0; % Ambient temperature of space without initialized wall temp (°C)
% Constants for the second material
ku = 0.7;  % Thermal conductivity of left material (W/m°C)
% Constants and parameters specific to sweat
Si = Lx * Ly;  % Area of skin surface (m^2)
Pe = 101325;  % Pressure of environment (Pa)
Psat = 4500;  % Saturated vapor pressure of the skin (Pa)
Vw = 0;
Ve = 0;
Resk = 33.6;  % Water vapor resistance of the skin (m^2*Pa*s/kg)
mrsw = 1; %regulatory sweating
Rva = 2430 * 1000 / (0.1353 * sqrt(0.11 + 0.45 * Ve + Vw));  % Water vapor resistance
of the air layer above skin (m^2*Pa*s/kg)
% Discretization
x = 0:dx:Lx;
y = 0:dy:Ly;
nx = length(x);
ny = length(y);
% Initialize time derivatives
dTdt_initial = zeros(nx, ny);  % Initial first time derivative of temperature
d2Tdt2_initial = zeros(nx, ny);  % Initial second time derivative of temperature
% Initialize temperature fields
T_initial = ones(nx, ny) * T0;  % Initialize entire temperature field to T0
T_new_initial = ones(nx, ny) * T0;  % Initialize entire temperature field to T0
% Graph generation loop for 4th boundary condition
figure('Position', [100, 100, 1000, 600]);
hold on;
for k = k_list
   % Loop for both Qsweat present and not present
   for sweat_idx = 1:2 %using index 1 to show curves when Qsweat is present. Index 2
is when Qsweat is not present
       % Copy time derivatives
```

```
        dTdt = dTdt_initial;
        d2Tdt2 = d2Tdt2_initial;
        % Copy temperature fields
        T = T_initial;
        T_new = T_new_initial;
        % Store temperature profile at each time step
        temperature_profile = [];
        % Time integration
        for t = 1:time_steps
            for i = 2:nx-1
                for j = 2:ny-1
                    % From eq. 5 in Kumar and Kaur paper, the metabolic
                    % heat product
                    Qm = Qm0 * (1 + (Tl - T0) / 10);
                    % From eq. 6 in Kumar and Kaur paper, the heat from
                    % blood perfusion
                    Qb = wb * rho_b * cb * (Tb - Tl);
                    % Calculate Qsweat if temperature exceeds 37.38°C and sweat_idx ==
1
                    Tsk = T(i, j);  % Skin temperature at the current point
                    if sweat_idx == 1 && Tsk > 37.38
                        Psk = ((mrsw * Resk * Rva) + (Psat * Tsk * Rva) + (Pe * Resk))
/ (Resk * Rva); %From Aihua, Mao paper
                        Qsweat = - (Si * (Psk - Pe) / Rva);  % Heat loss due to
evaporative cooling of sweat
                    else
                        Qsweat = 0;
                    end
                    % Discretization using finite difference method
                    d2Tdx2 = (T_new(i + 1, j) - 2 * T_new(i, j) + T_new(i - 1, j)) /
dx ^ 2;
                    d2Tdy2 = (T_new(i, j + 1) - 2 * T_new(i, j) + T_new(i, j - 1)) /
dy ^ 2;
                    % From eq. Kumar paper
                    dTdt(i, j) = (k * (d2Tdx2 + d2Tdy2) + Qb + Qm + Qsweat) / (rho *
c);
                    % Taking the derivative with k* integrated
                    d2Tdt2(i, j) = (k_star * (d2Tdx2 + d2Tdy2)) / (rho * c);
                    % Substitute finite differences from discretization. Rearranged
energy balance equation
                    T_new(i, j) = T(i, j) + dt * (dTdt(i, j) + tau_q * dTdt(i, j) -
tau_T * d2Tdt2(i, j) + (k + k_star * tau_v) * dTdt(i, j));
                end
            end
            % Boundary conditions for 4th boundary condition
            fourth_boundary(T_new, k, ku);
            % Update temperature
            T = T_new;
```

```
            % Store temperature profile
            temperature_profile(:, :, t) = T_new;
        end
        % Plot temperature at a specific point over time
        if sweat_idx == 1
            plot_style = '-';
            legend_label = 'present';
        else
            plot_style = '--';
            legend_label = 'not present';
        end
        plot((0:dt:(time_steps-1)*dt), squeeze(temperature_profile(nx / 2, ny / 2,
:)), plot_style, 'DisplayName', sprintf('k = %.3f W/m°C, Qsweat %s', k,
legend_label));
    end
end
xlabel('Time (s)');
ylabel('Temperature (°C)');
title('Temperature Profile Over Time (4th Boundary Condition)');
legend('show');
grid on;
% Functions must be at the end of the script in MATLAB
% Boundary between wall and tissue
function wall_boundary(T_array, wall_temp)
    T_array(end, :) = wall_temp;  % x = 0 (bottom boundary)
    T_array(:, 1) = wall_temp;  % y = 0 (left boundary)
end
% Symmetrical Boundary conditions from eq. 16
function symmetric_boundary(T_array)
    T_array(1, :) = T_array(2, :);  % x = Lx
    T_array(end, :) = T_array(end-1, :);  % x = 0
    T_array(:, 1) = T_array(:, 2);  % y = 0
    T_array(:, end) = T_array(:, end-1);  % y = Ly
end
% Convective boundary condition to introduce a constant heat coefficient
function convective_boundary(T_array, h, dx, Tl, k)
    T_array(1, :) = (h * dx * Tl + k * T_array(2, :)) / (h * dx + k);  % x = Lx
    T_array(:, end) = (h * dx * Tl + k * T_array(:, end-1)) / (h * dx + k);  % y = Ly
end
% Fourth boundary for constant temperature and heat flux from secondary paper
function fourth_boundary(T_array, k, ku)
    T_array(end, :) = T_array(end-1, :) - (k / ku) * (T_array(end-1, :) -
T_array(end-2, :)); % x = 0
    T_array(:, 1) = T_array(:, 2) - (ku / k) * (T_array(:, 2) - T_array(:, 3));  % y =
0
end
function third_boundary(T_array, h, dx, Tl, k)
    T_array(end, :) = (h * dx * Tl + k * T_array(end-1, :)) / (h * dx + k);  % x = 0
```

```
    T_array(:, 1) = (h * dx * Tl + k * T_array(:, 2)) / (h * dx + k);  % y = 0
end
```