1-D Three Phase Lag Bio-heat Model for Targeted Hyperthermia Treatment in Multiple Skin Layers

Solana Fernandez[1] and Andrew Carpino[1]

[1]Department of Bioengineering, University of California, San Diego, 3223 Voigt Dr., La Jolla, 92093, CA, USA.

August 5, 2025

Abstract

Accurate prediction of temperature fields is critical for safe and effective hyperthermia. Based off of a single layer model, our one-dimensional three phase lag bio-heat model treats skin as three distinct layers; epidermis, dermis and subcutaneous fat. We used a targeted Gaussian heat source at tumor depth rather than at the skin surface boundary. Beyond the standard metabolic and perfusion terms, our model adds vaporization and diffusive heat components, creating a more complete representation of how energy moves and dissipates in living tissue during therapy.

1 Introduction

Hyperthermia and thermal ablation therapies are widely used in oncology to selectively destroy tumor tissue with heat while minimizing collateral damage to healthy tissue. The effectiveness of these techniques hinges on our ability to predict and control temperature distributions within living tissue, and thus the accuracy of mathematical bioheat transfer models. The foundational model in this area is the Pennes [2] bioheat equation, which treats tissue as a homogeneous medium and assumes heat propagation via classical Fourier's law [1]. While influential, this model has important limitations, including the nonphysical assumption of infinite thermal propagation speed, which can lead to significant discrepancies when simulating rapid thermal events such as those encountered during hyperthermia or ablation.

To address the shortcomings of the Pennes and classical Fourier models, Cattaneo [2] and Vernotte [3] introduced the concept of thermal relaxation, leading to the single phase lag (SPL) model, which accounts for a finite delay between heat flux and temperature gradient changes. Tzou further advanced this work with the dual phase lag (DPL) model, introducing two separate relaxation times; one for the heat flux and one for the temperature gradient—thereby more realistically capturing the lagging behavior of heat conduction in biological tissues [4, 5]. Although the DPL model improved temporal accuracy, it still could not capture all aspects of non-Fourier heat conduction, particularly in tissues with complex structure, heterogeneity, and multi-scale transport phenomena. This led to the development of the three phase lag (TPL) model [6], which incorporates an additional relaxation time associated with thermal displacement. The TPL framework has shown enhanced predictive power in modeling non-equilibrium thermal processes in tissue, especially in situations with high-intensity, short-duration heating or cooling, such as in modern ablation and cryosurgical procedures.

While previous works on DPL and TPL models offered important theoretical insights, most have been restricted to either homogeneous (single-layer) tissue or to boundary conditions representing heat applied to the tissue surface [6], conditions that do not always match clinical scenarios. In reality, human skin comprises multiple layers (epidermis, dermis, subcutaneous fat) with distinct thermal properties and biological functions. Recent numerical studies have begun to incorporate multi-layer skin models, demonstrating that such structure can critically influence both the temperature distribution and the extent of tissue damage during hyperthermia [5]. Furthermore, traditional models often apply either Dirichlet (fixed temperature), Neumann (fixed heat flux), or Robin (convective) boundary conditions at the skin surface. However, hyperthermia devices in clinical practice frequently deliver energy directly and locally, such as by focused ultrasound or lasers, resulting in a spatially localized, often Gaussian-distributed, heat source within the tissue, rather than simply at its boundary. Recent literature highlights the necessity of modeling such targeted sources to more accurately reflect clinical treatments [5]. In addition to external heating, other internal heat sources, such as metabolic activity, blood perfusion, and vaporization (evaporative cooling), play significant roles in the temperature field and thus in tissue damage or survival. Incorporating all these terms is crucial for developing truly predictive models.

Despite advances in both mathematical rigor and computational methods, few studies have fully bridged the gap between theoretical models and the real clinical context—where tissue structure is layered, external heat source is targeted, and multiple physiological phenomena (diffusion, perfusion, metabolism, vaporization) act concurrently. This integration is critical for improving the safety and efficacy of hyperthermia treatments, guiding device design, and personalizing therapy to patient-specific anatomy and pathology.

Recent numerical implementations have shown that including multiple skin layers, realistic boundary conditions, and localized heat sources produces temperature profiles that better match experimental and clinical outcomes. For example, implementing a Gaussian-distributed heat source captures the focused effect of energy deposition by hyperthermia devices, as opposed to the non-physiological scenario of heating propagating inward from the surface. Similarly, explicit inclusion of vaporization and perfusion terms is important for capturing tissue responses under high-intensity heating and for predicting cooling effects from sweat or tissue vaporization, especially in highly vascular or metabolically active regions.

The aim of this project is to develop and implement a one-dimensional, three phase lag (TPL) bioheat model for hyperthermia and heat ablation, explicitly incorporating three skin layers (epidermis, dermis, subcutaneous fat) and a Gaussian-distributed external heat source to realistically simulate a focused hyperthermia device. The model also integrates all key physiological and physical heat sources and sinks, including diffusion, blood perfusion, metabolic heat production, and vaporization, to provide a comprehensive and predictive framework for temperature distribution in living tissue under therapeutic heating.

2 Baseline Model

The three-phase-lag (TPL) bio-heat formulation introduced by Kumar & Kaur [6] we will treat here as our baseline description of heat transfer during hyperthermia. Their model represents a single, homogeneous block of skin initially at core body temperature and subsequently exposed to an external thermal load applied at the tissue surface. Within that framework they incorporate the following:

- **Finite-speed heat conduction via three relaxation times;** one for heat flux, one for the temperature gradient, and one for thermal displacement, so the model overcomes the infinite-propagation limitation of standard Fourier conduction.
- **Physiological heat sources and sinks;** metabolic heat generation, and blood-perfusion exchange with arterial blood held at a fixed core temperature.
- **Surface-only energy delivery;** implemented through one of three classical boundary types (fixed surface temperature, fixed surface flux, or convective exchange), where the opposite face of the slab is treated as adiabatic, mimicking symmetry or insulation.
- **Uniform material properties throughout the skin;** a single value each for thermal conductivity, density, and specific heat.

Kumar & Kaur express temperature, space, time, and material parameters in dimensionless form, then solve the resulting problem numerically. Their procedure couples a finite-difference discretization in space with a Legendre-wavelet Galerkin expansion in time for time marching while retaining the higher-order time derivatives introduced by the lag terms. The resulting temperature profiles show how changing a lag time, the perfusion rate, or the metabolic rate alters the transient temperature field inside the slab. Because the heat load is applied only at the surface and the slab is homogeneous, internal temperature evolution is driven solely by conduction and perfusion from that boundary inward.

This model explores a 2-D finite domain, using a hybrid approach of finite difference method (FDM) and Legendre Wavelet Galerkin Method (LWGM). FDM using the central differencing method for the second spatial derivative efficiently handles the spatial grid and complex boundary conditions. LWGM enables high-accuracy solutions and efficient handling of stiff systems, especially for time-dependent problems with higher-order derivatives and phase lag effects. The system is then solved using the Bartels–Stewart algorithm.

2.1 Governing Equations

Heat transport in tissue is described here through a sequence of constitutive laws that culminate in the three-phase-lag (TPL) formulation adopted by Kumar & Kaur. Starting with standard Fourier law,

$$q = -k\nabla T \tag{1}$$

which links the heat-flux vector q to the spatial temperature gradient through the thermal conductivity k. Combining this with conservation of energy gives

$$\rho c\left(\frac{\partial T}{dt}\right) = -k\nabla q(r,t) + Q_{tot} \tag{2}$$

where ρ and cc are tissue density and specific heat, and Q_{tot} represents all volumetric sources and sinks (metabolic heat, perfusion, external heating, etc.). For finite speed heat transfer, Cattaneo and Vernotte [2] introduced a flux relaxation time τ_q, which introduces a delay between the formation of a temperature gradient and the resulting heat flux, yielding the single-phase-lag (SPL) law:

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

Tzou [4]extended this idea by adding a temperature-gradient relaxation time, τ_T, as a response of the temperature gradient to heat flux, producing the dual-phase-lag (DPL) form. This accounts for microstructural delays within tissue, such as in cells or capillaries:

$$q(r, t+\tau_q) = -k\nabla T(r, t+\tau_T) \tag{4}$$

Kumar & Kaur [6] further refined the model by including a third delay, τ_v, that represents a lag in the way thermal energy density shifts inside tissue; the resulting TPL constitutive law is as follows:

$$q(r, t+\tau_q) = k\nabla T(r, t+\tau_T) + k^* \frac{d}{dt}\nabla T(r, t_v) \tag{5}$$

These three τ parameters correct the non physical assumption of instant heat transfer found in Fourier-based models and allow for more accurate modeling of rapid or high-intensity thermal treatments. Solving for q from **Equation 5** and substituting into the local energy balance produces the governing TPL energy equation used throughout this work:

$$\left(1+\tau_q\frac{d}{dt}\right)\left(\rho c\frac{\partial^2 T}{\partial t^2} - Q_{tot}\right) = \left(k^* + (k+k^*\tau_v)\frac{\partial}{\partial t} + k\tau_T\frac{\partial^2}{\partial t^2}\right)\left(\frac{\partial^2 T}{\partial r^2}\right) \tag{6}$$

The LHS multiplies the storage term and volumetric sources, embedding the heat-flux lag, while the RHS embeds both gradient and displacement lags. In the baseline model $Q_{tot} = Q_m + Q_b$, where Q_m is metabolic heat generation and $Q_b = \rho_b\, c_b\, w_b(T_b - T)$ represents blood-perfusion exchange. ρ_b, c_b, w_b, and T_b represent the density, specific heat, perfusion rate, and temperature of blood, respectively. These values are summarized in **Table 2**.

2.2 Initial and Boundary Conditions

Initial conditions of the baseline model are as follows, where T_0 is initial temperature of the tissue:

$$T(x, y, 0) = T_0 \tag{7}$$

$$\frac{\partial T(x, y, 0)}{\partial t} = 0 \tag{8}$$

$$\frac{\partial^2 T(x, y, 0)}{\partial t^2} = 0 \tag{9}$$

Boundary conditions (summarized in **Table 1**) can be expressed with

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

where A_1, B_1, and f_1 dictate the boundary type, and can be similarly applied in the y-dimension. Values were applied according to physical application: cryosurgery, ablation, ambient exchange, or insulated domains.

Boundary Conditions of First Kind (Dirichlet/-Constant Temperature)

$$A_1 = 0, B_1 = 1, f_1(y, t) = T_w$$

where T_w is temperature at the boundary. This scenario applies to cryosurgery or cooling probe applications; laser/ablation devices with precision in temperature control.

Boundary Conditions of Second Kind (Neumann/Constant Heat Flux)

$$A_1 = -k, B_1 = 0, f_1(y, t) = q_w$$

where q_w is the specified flux. This may relate to heat ablation where a device delivers a constant heat flux (not fixed temperature) or high-intensity focused ultrasound or laser ablation scenarios where energy delivery is specified as a rate (W/m^2) rather than a fixed temperature. Moreover, $q_w = 0$ indicates perfect insulation, which may be ideal for an internal boundary, centerline of symmetry, or insulated boundary. This is often applied at the far edge (deep tissue, core) or mid-plane for symmetric models.

Boundary Conditions of Third Kind (Robin/-Convective)

$$A_1 = -k, B_1 = h, f_1(y, t) = hT_p \tag{10}$$

with heat transfer coefficient, h, and reference temperature, T_p. This is perfect for convective cooling/heating (e.g., tissue in contact with air, water, or perfused blood vessels) and imitating physiological/realistic skin boundary - exchange with ambient air, immersion in fluid, or subdermal perfusion.

3 Improved Model Rationale

The baseline model in Kumar, M. et al. [6] shows finite-speed thermal waves and basic physiological heating/-cooling, but it excludes two clinically important and two physiological realities that will be addressed in our improved formulation:

1. **Layered skin architecture;** epidermis, dermis, and subcutaneous fat each possess distinct conductivities, perfusion rates, and metabolic activity.
2. **Localized internal heating;** many modern hyperthermia devices deposit energy within the tissue rather than exclusively on the exterior surface.
3. **Effects of sweat evaporation:** Oon the skin surface (epidermis), heat is lost to the environment by water vaporization.
4. **Heat due to diffusion:** in all skin layers, there is an unaccounted-for heat loss due to water diffusion that depends on parameters intrinsic to each tissue type.

3.1 Mathematical Formulation

Our model adopts the TPL representation of Kumar, M. et al.'s bio-heat model in one dimension. Accordingly, after taking **Equation 5**, performing a first-order Taylor series expansion of each term and solving for $q(r, t)$, we yield **Equation 6**.

Like Kumar, M. et al., Q_{tot} represents the sum of all heat sources. However, where it previously consisted of the sum of perfusion and metabolic heat parameters (Q_b and Q_m, respectively), the improved model additionally considers water vaporization and diffusive heat (Q_v and Q_{d_i}, respectively):

$$Q_v = \frac{\Delta m \Delta H_{vap}}{\delta_c} \tag{11}$$

$$\Delta m = \frac{D_a M_w}{R_a} \left(\frac{P_w}{T_w} \right)_s \frac{RH}{\delta_c} \tag{12}$$

$$Q_{d_i} = \frac{D_{f_i} c_w (\rho_s - \rho_c)}{(\nabla r)^2} (T_i - T_{0_i}) \tag{13}$$

as well as a targeted, Gaussian distributed heat source, represented by Q_r:

$$Q_r = Q_{r_0} e^{-a_0 (r - r^*)^2} \tag{14}$$

$$Q_{r_0} = \rho_i SP \tag{15}$$

Where $\Delta m, \Delta H_{vap}$, and δ_c are the rate of water vaporization, the enthalpy of water vaporization, and average boundary layer difference, respectively. $D_{f_i}, c_w, \rho_s, \rho_c$, and ∇r are the diffusivity of water, water content on the layer skin, water content on the core body, and characteristic diffusion distance. S, P, a_0, r, r^* are the antenna constant, power, scattering parameter, depth of location, and depth of the tumor. Optimized and empirical values can be found in **Table 2**.

Initial conditions were adopted from the original model (**Equations 7- 9**), where the first and second

Boundary Condition	Application/Scenario	Example
First Kind (Dirichlet)	Cryosurgery probe, fixed-temp ablation	Fixed probe at 100°C or −180°C
Second Kind (Neumann)	Fixed heat flux ablation, laser energy delivery	Laser delivering 5000 W/m^2
Third Kind (Robin/Convective)	Skin–air/water interface, convective boundary	Skin exposed to air ($h \approx$ 5–25 W/m^{2}K)
Fourth Kind (Mixed Interface)*	Tissue–device/tumor–normal interface (advanced models)	Tumor boundary, prosthetic interface
Symmetry/Zero-Flux (Neumann)	Core boundary, centerline, deep insulation	Deep tissue, center symmetry

Table 1: *Boundary Conditions in Kumar, M. et al. [6] and Their Physical Applications. *Not included in paper, but used in advanced implementations.*

temporal derivatives of temperature are equal to zero at time zero. Additionally, initial temperature at time zero (T_0) throughout each layer of tissue is considered to be $37°C$.

Boundary Conditions of Third Kind (Robin/Convective) were employed at either end:

$$-k\frac{\partial T}{\partial t}\bigg|_{x=0|L} = h(T_{x=0|L} - T_{amb})$$

Accordingly, using a forward Euler approximation at the superficial end:

$$T(x=0,t) = \frac{(\Delta x h T_{amb}) + k_i T(\Delta x, t)}{\Delta x h + k_i} \tag{16}$$

and backward Euler approximation at the deep end:

$$T(x=L,t) = \frac{\Delta x h T_{amb} - k_i T(L-\Delta x, t)}{\Delta x h - k_i} \tag{17}$$

At the interface/shared boundaries of the epidermis, dermis, and subcutaneous tissues temperature and heat flux were maintained/equivalent.

Central differencing is used to achieve the second spatial derivative, and forward difference for the first time derivative. Higher-order time derivatives were approximated with a psuedo-identity due to their near-negligible effects:

$$\frac{\partial^2 T}{\partial x^2} = \frac{T(x+\Delta x,t) + T(x-\Delta x,t) - 2T(x,t)}{\Delta x^2}$$

$$\frac{\partial T}{\partial t} = \frac{T(x,t+\Delta t) - T(x,t)}{\Delta x}$$

$$\frac{\partial^2 T}{\partial t^2} = \frac{k^*(\frac{\partial^2 T}{\partial x^2})}{\rho c}$$

Which, when substituted into **Equation 6** and rearranged for $T(x,t+\Delta t)$ (or T_i^{n+1} in discretized form) yields the time update rule:

$$T_i^{n+1} = T_i^n + \Delta t\Bigg(\frac{(k\frac{\partial^2 T}{\partial x^2} + Q_{tot})}{\rho_i c_i} + \tau_q\frac{(k\frac{\partial^2 T}{\partial x^2} + Q_{tot})}{\rho_i c_i} - \tau_T\frac{k^*}{\rho_i c_i}\frac{\partial^2 T}{\partial x^2} + (k+\tau_v k^*)\frac{(k\frac{\partial^2 T}{\partial x^2} + Q_{tot})}{\rho_i c_i}\Bigg) \tag{18}$$

3.2 Implementation in MATLAB

Our improved model was executed in MATLAB using the time-step update step described by **Equation 18)** for internal points, and the exterior nodes with **Equations 16- 17**. Physical parameters, such as ρ_i, c_i, and others, were recovered from literature, and those that were unavailable (τ_q, τ_v, and τ_T) or are clinically tunable (P, S, and a_0) were optimized for our implementation. See empirical and optimized values in **Table 2**.

4 Results

After implementing the improvements to our model, we first assessed the influence of the three phase-lag parameters on the spatial temperature profile at $t = 10$s (**Figure 1a**). Varying behaviors of the temperature profiles guided our selection of $[\tau_q, \tau_T, \tau_v] = [10, 20, 1]$. Next, we varied the applied power P of the Gaussian source Q_r from 10W to 50W and tuned P to 35W to achieve a maximum value of around 45 °C (**Figure 2a**), then adjusted the antenna constant a_0 to 1×10^6 to balance localization and peak temperature (**Figure 2b**). Finally, we observed the location of Q_r's influence on the temperature distribution curve, entirely within the different layers and comparing the full model (with Q_v and Q_d) against different variations. We found that omitting Q_v or Q_d does not significantly change the curve for a tumor at $r^* = 8$ mm (**Figure 4a**).

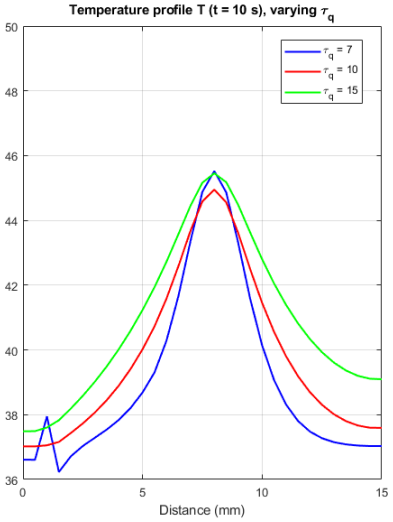
4.1 Varying Tau Parameters

To observe the differences in tau values, we generated temperature distributions over the distance into the skin with varying parameters. Each curve was generated at t = 10 seconds with default parameters equivalent to our original heat distribution. Default parameters for tau are 10, 20, and 1 for τ_q, τ_T, and τ_v, respectively. Results shown in **Figure 1**.

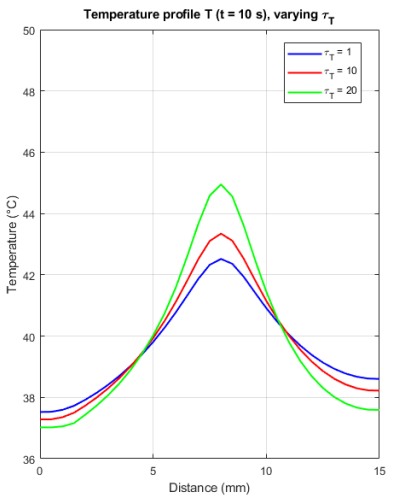
τ_Q: Values below 7 corrupt the model and temperature values are inconsistent. Values around 15 flatten out and also decrease localization of the heat too much; value of 10 was kept.

τ_T: Higher values tightened the heat distribution at the cost of exponentially increasing maximum temperature; we kept a value of 20 and tuned power P of the Q_r term (**See Table 2**) for an acceptable temp value. See [power section] for comparisons.

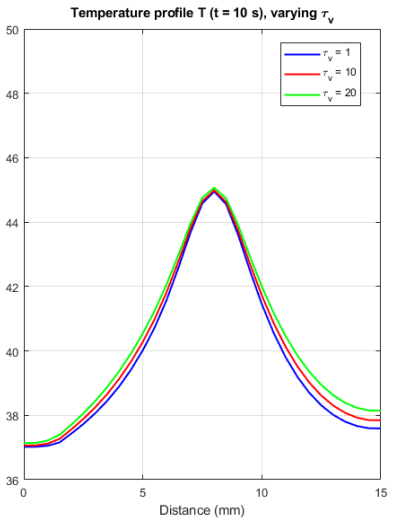
τ_V: Remained stable for increasing values, but increasing values decreased localization of the hyperthermia treatment. Minimizing τ_v value had best results.



(a)



(b)



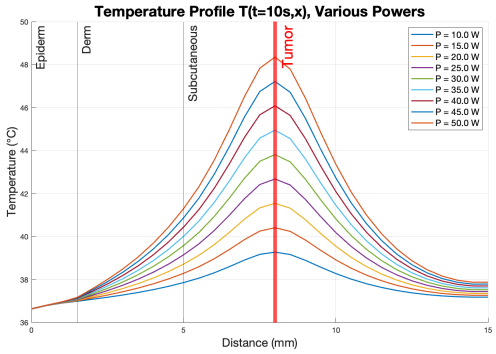
(c)

Figure 1: *Altering τ_Q* ***(a)****, τ_T* ***(b)****, and τ_V* ***(c)*** *at $t = 10$ seconds and 35W in the multilayer model.*

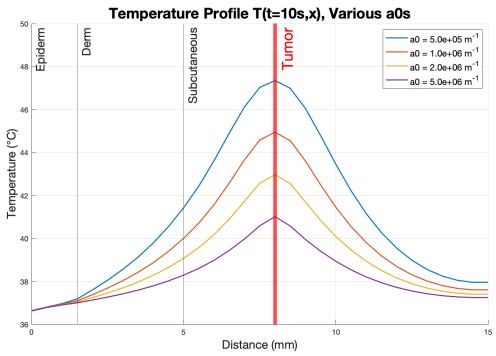
4.2 Varying Targeted Heat Source Parameters

To evaluate how external thermal energy influences spatial temperature profiles across multilayer skin tissue, we varied the power term P in our model. This analysis allowed us to examine how different levels of applied energy alter heat propagation and accumulation at the tumor site embedded within the subcutaneous layer.

The range of power inputs tested (10–50 W) reflects clinically realistic values used in localized hyperthermia. Shellock and Crivelli [7] demonstrated that 60W of microwave energy applied for 30 minutes resulted in safe, tolerable heating of muscle tissue without irreversible thermal injury. This guided our selection of 50W as the upper bound, providing confidence that the modeled heating conditions are consistent with clinical tolerability.



(a)



(b)

Figure 2: *Spatial profile of temperature at various powers* ***(a)*** *and various scattering parameters, a_0* ***(b)****.*

As shown in **Figure 2a**, increasing power leads to greater temperature elevation within the tissue, with 35W producing a tumor-centered peak just above 45°C. This temperature falls within the optimal therapeutic hyperthermia range, as shown by Zahra et al. [8], who emphasized that temperatures exceeding 43°C reliably induce cancer cell death without compromising adjacent healthy tissue. This considers the tumor only in the subcutaneous layer; in the next section we discuss the effects of the skin layers and show how the wattage chosen in therapy varies based on the location of the

tumor.

Gaussian width control was also a factor in alteration of parameters. The antenna constant a_0 directly contributed to the speread of the heat source Q_r at the site of the tumor. Shown in **Figure 2b**, narrowing the source represented an ideal heat source and maintained a more localized heat distribution, however increased peak temperature to levels beyond our desired threshold. Finalizing a value of $1e6$ (**Table 2**) allowed for the proper temperature at a constant power of 35W determined previously.

4.3 Impact of Skin Layers & Heat Sources

We observed the impact of skin layers to be significant in both location of the tumor and overall heat distribution. Overall, accounting for heterogeneity of tissues (and therefore different values of ρ_i and c_i) results in a flattened spatial profile. This is consistent with the fact that the epidermis and dermis have larger and smaller specific heats, causing temperatures to be hotter and cooler under the same conditions in either tissue, respectively (**Figure 3**). Because the subcutaneous tissue in our model is predominant (10mm) and the tumor is located within this domain, subcutaneous homogeneity looks similar to our 3-layer model. However, when the subcutaneous tissue thickness changes (as it does throughout various locations of the human body and varies individual-to-individual) and the tumor moves locations, the spatial distribution of temperature is highly modulating. This variance demonstrates the need for precise measurements, targeting, and control of device heat source parameters in the clinical setting.

Exclusion of our additional heat sources Q_v and Q_d shows minimal effect on temperature profile for a tumor located at $r^* = 8mm$ within the subcutaneous tissue. However, when the tumor moves locations, such as into the dermis as seen in **Figure 4**, there is a more significant difference between our improved skin-layer TPL model and the baseline.

5 Discussion

In this work, we have presented a three-phase-lag bioheat model for multilayered skin that captures important characteristics of hyperthermia treatment. By tuning the phase-lag delays, the power input, and the spread of the Gaussian heat source, we identified the key parameters influencing treatment localization and peak temperature. These findings highlight the critical role of parameter selection in designing safe and effective hyperthermia treatments. The ability to predict how changes in time lags, applied power, and source geometry affect internal heating has the potential for improving treatment options and planning.

Several different advancements can further enhance our model fidelity and clinical applicability. Incorporating heat-sink effects will improve predictions of temperature dynamics over longer treatment durations, and incorporating a feedback control system that ensures the tumor remains within the therapeutic temperature range will provide more realistic simulations. We also plan to investigate regional differences in skin thickness and thermal properties based on location of the body. The largest extension of our model would be applying it into two- and three-dimensional geometries to capture complex tissue interfaces that are more biologically relevant.

6 Code Availability

MATLAB code for the mathematical model and for generating the figures and plots is available on GitHub: TPL_bioheat_model.

Scripts:

- changing_tau_final.m: Performs the analysis under *'Varying Tau Parameters'*
- a0_test_final.m & power_test_final.m: Performs the analysis under *'Varying Targeted Heat Source Parameters'*
- skin_layer_comparison.m: Performs the analysis under texit'Impact of Skin Layers and Heat Sources'
- main_final.m: Runs the model with optimized parameters

References

[1] Harry H Pennes. Analysis of tissue and arterial blood temperatures in the resting human forearm. *Journal of applied physiology*, 1(2):93–122, 1948.

[2] Carlo Cattaneo. A form of heat-conduction equations which eliminates the paradox of instantaneous propagation. *Comptes rendus*, 247:431, 1958.

[3] Pierre Vernotte. Les paradoxes de la theorie continue de l'equation de la chaleur. *Comptes rendus*, 246:3154, 1958.

[4] Da Yu Tzou. *Macro-to microscale heat transfer: the lagging behavior*. John Wiley & Sons, 2014.

[5] Tejaswini Kumari, Dinesh Kumar, KN Rai, and SK Singh. Numerical solution of dpl heat transfer model in multi-layer biological skin tissue of the living body during hyperthermia treatment. *Mechanics Based Design of Structures and Machines*, 51(1):159–178, 2023.

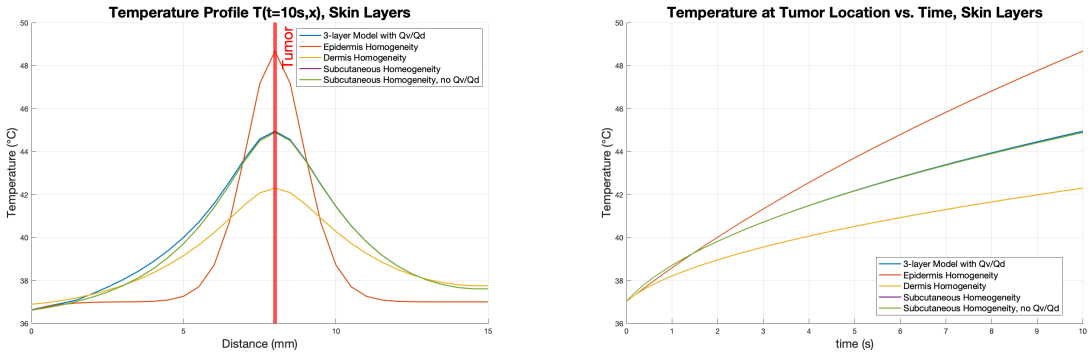
[6] Mukesh Kumar, Subrahamanyam Upadhyay, Surjan Singh, KN Rai, et al. Mathematical modelling and simulation of three phase lag bio-heat transfer model during cancer treatment. *International Journal of Thermal Sciences*, 184:108002, 2023.

[7] Noriko Ichinoseki-Sekine, Hisashi Naito, Norio Saga, Yuji Ogura, Minoru Shiraishi, Arrigo Giombini, Valentina Giovannini, and Shizuo Katamoto. Changes in muscle temperature induced by 434 mhz microwave hyperthermia. *British Journal of Sports Medicine*, 41(7):425–429, 2007.

[8] Weiwei Zhu, Siwei Pan, Jiaqing Zhang, Jingli Xu, Ruolan Zhang, Yanqiang Zhang, Zhenjie Fu, Yuqi Wang, Can Hu, and Zhiyuan Xu. The role of hyperthermia in the treatment of tumor. *Critical Reviews in Oncology/Hematology*, page 104541, 2024.

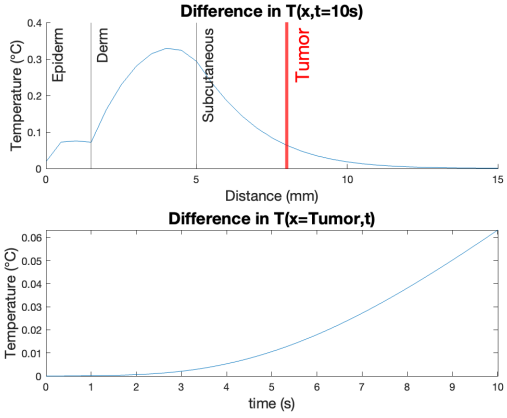
Variable	Name	Units	Value	Reference
ρ_1	epidermis denisty	kg/m^3	1150	[9]
ρ_2	dermis density	kg/m^3	1116	[10]
ρ_3	subcutaneous density	kg/m^3	900	[11]
c_1	epidermis specific heat	$J/kg \cdot C$	3590	[12]
c_2	dermis specific heat	$J/kg \cdot C$	3330	[12]
c_3	subcutaneous specific heat	$J/kg \cdot C$	2500	[12]
L_1	thickness of epidermis	mm	1.5	[5]
L_2	thickness of dermis	mm	3.5	[5]
L_3	thickness of subcutaneous	mm	10	[5]
k_1	epidermis thermal conductivity	$W/m^\circ C$	0.2	[13]
k_2	dermis thermal conductivity	$W/m^\circ C$	0.45	[13]
k_3	subcutaneous thermal conductivity	$W/m^\circ C$	0.3	[13]
r^*	tumor depth/location	mm	0.8	**
P	transmitted power	W	35	[14]
S	antenna constant	1/kg	15	[14]
a_0	scattering parameter	m^{-1}	1e6	[14]
τ_T	relaxation time due to temperature gradient	s	20	[15] **
τ_v	relaxation time due to thermal displacement	s	1	[15] **
τ_q	relaxation time due to heat flux	s	10	[15] **
k^*	hyperbolic-term coefficient	$W/m^\circ Cs$	0.1	*
h	convective coefficient at boundaries	$W/m^{2\circ}Cs$	4.5	[5]
w_b	blood perfusion rate	s^{-1}	0.8	[6]
ρ_b	density of blood	kg/m^3	1056	[5]
c_b	blood specific heat	$J/(kg^\circ C)$	4000	[5]
Q_{m_0}	metabolic heat baseline	W/m^3	50.65	[6]
T_b	arterial blood temperature	$^\circ C$	37	[6]
T_l	ambient/tissue temperature	$^\circ C$	37	[6]
D_a	average water vapor diffusivity	m^2/s	2.5e-5	[16]
M_w	molar mass of water	kg/mol	0.018	[17]
R_a	universal gas constant	$J/mol * K$	8.314	[17]
T_w	absolute temperature of skin surface	K	306	[18]
R_h	Relative Humidity	unit-less fraction	0.5	[19]
δ_c	average boundary layer difference	m	1e-4	[18]
c_{air}	specific heat of air	$J/kg^\circ C$	1005	[20]
D_f	diffusivity of water	m^2/s	2e-9	[21]
c_w	specific heat of water	$J/kg^\circ C$	4180	[5]
ρ_s	water content on the epidermis	kg/m^3	1100	[5]
ρ_c	water content on the core body	$J/kg^\circ C$	1000	[5]
$(\nabla r)^2$	characteristic distance squared	m^2	${L_i}^2$	[5]

Table 2: *A summary of the optimal parameters. (* An exact value could not be found within literature and was estimated. ** Variables changed and optimized throughout model development.)*

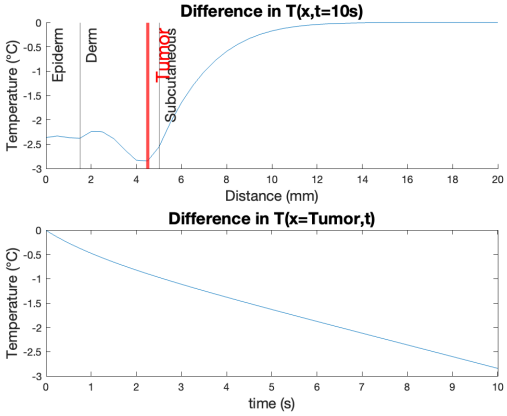


(a) ***(b)***

Figure 3: *Spatial profile of temperature* ***(a)*** *and temperature at tumor location ($r^* = 8mm$)* ***(b)*** *considering the physical parameters of various skin layers. The green line considers subcutaneous homogeneity without the added heat sources of vaporization or diffusion.*



(a)



(b)

Figure 4: *Difference between the improved model and baseline model (assuming subcutaneous homogeneity without Q_v or Q_d, but using the targeted heat source, Q_r) for the temperature profile at 10s and temperature at the tumor over time. Figure* ***(a)*** *illustrates $r^* = 8mm$ in the subcutaneous tissue and* ***(b)*** *is $r^* = 4.5mm$ in the dermis.*

[9] John Cameron. Physical properties of tissue. a comprehensive reference book, edited by francis a. duck, 1991.

[10] Chunhui Li, Guangying Guan, Roberto Reif, Zhihong Huang, and Ruikang K Wang. Determining elastic properties of skin by measuring surface waves from an impulse mechanical stimulus using phase-sensitive optical coherence tomography. *Journal of The Royal Society Interface*, 9(70):831–841, 2012.

[11] F Fidanza. Body fat in adult man: semicentenary of fat density and skinfolds. *Acta diabetologica*, 40:s242–s245, 2003.

[12] Ming Fu, Wenguo Weng, and Hongyong Yuan. Numerical simulation of the effects of blood perfusion, water diffusion, and vaporization on the skin temperature and burn injuries. *Numerical Heat Transfer, Part A: Applications*, 65(12):1187–1203, 2014.

[13] Zerin Jahan Tasnim and R Nasrin. Thermal wave and pennes' models of bioheat transfer in human skin: A transient comparative analysis. *Heliyon*, 10(21), 2024.

[14] Praveen Kumar Gupta, Jitendra Singh, Kabindra Nath Rai, and SK Rai. Solution of the heat transfer problem in tissues during hyperthermia by finite difference–decomposition method. *Applied Mathematics and Computation*, 219(12):6882–6892, 2013.

[15] Dinesh Kumar, Surjan Singh, and KN Rai. Analysis of classical fourier, spl and dpl heat transfer model in biological tissues in presence of metabolic and external heat source. *Heat and Mass Transfer*, 52(6):1089–1107, 2016.

[16] Charles V Paganelli and Fred K Kurata. Diffusion of water vapor in binary and ternary gas mixtures at increased pressures. *Respiration Physiology*, 30(1-2):15–26, 1977.

[17] G.A. Truskey, F. Yuan, and D.F. Katz. *Transport Phenomena in Biological Systems*. Pearson Prentice Hall bioengineering. Pearson Prentice Hall, 2009.

[18] Ming Fu, Wenguo Weng, and Hongyong Yuan. Numerical simulation of the effects of blood perfusion, water diffusion, and vaporization on the skin temperature and burn injuries. *Numerical Heat Transfer, Part A: Applications*, 65(12):1187–1203, 2014.

[19] Nelson GC Astrath, Jun Shen, Datong Song, Jurandir H Rohling, Francine BG Astrath, Jianqin Zhou, Titichai Navessin, Zhong Sheng Liu, Caikang E Gu, and Xinsheng Zhao. The effect of relative humidity on binary gas diffusion. *The Journal of Physical Chemistry B*, 113(24):8369–8374, 2009.

[20] Gases-Specific Heats and Individual Gas Constants. The engineering toolbox. *URL https://www.engineeringtoolbox.com/specific-heat-capacity-water-d_660. html*, 2004.

[21] COMSOL Multiphysics Cyclopedia. Understanding the diffusion coefficient. *URL https://www.comsol.com/multiphysics/diffusion-coefficient*, 2017.

6.1 main_final.m

```
1
2  %
3  % main_final.m
4  % 1D hyperbolic bioheat w/ Gaussian tumor source, 3 skin layers, finer mesh
5  % and reduced dt for stability; snapshots at 2.5, 5, 7.5, 10 s in one plot.
6  %
7
8  clear; clc;
9
10 %% 1) DISCRETIZE TIME & SPATIAL DOMAINS
11 % Skin Layer thickness
12 L_epi = 0.0015;    % Epidermis (1.5 mm)
13 L_derm = 0.0035;   % Dermis (3.5 mm)
14 L_subq = 0.01;     % Subcutaneous (10 mm)
15 Lx = L_epi + L_derm + L_subq; % Domain Length
16
17 x_ast  = 0.008;    % [m]   Tumor center (midpoint of 0.05 m)
18
19 dx     = 0.0005;  % [m]   Spatial step     now 21 nodes from 0 to 0.05
20 dt     = 0.015;     % [s]   Time step (smaller for stability)
21 max_time = 10;    % plot up to 10s
22 time_steps = round(max_time/dt) + 1; % ensures we reach exactly 10 s: ( n 1 )*dt =
       10
23 time = 0:dt:max_time+dt;
24
25 x  = 0:dx:Lx;          % x = [0, 0.0025, 0.005,    , 0.05]
26 nx = numel(x);
27
28 % Define Layers
29 layer = zeros(1, nx);
30 layer(x <= L_epi) = 1;                                    % Epidermis
31 layer(x > L_epi & x <= L_epi + L_derm) = 2;               % Dermis
32 layer(x > L_epi + L_derm) = 3;                           % Subcutaneous
33
34 % Find index of tumor center (closest grid point to x_ast)
35 [~, iTumor] = min(abs(x - x_ast));
36
37 %% 2) CONSTANTS AND PARAMETERS
38 % ------------ Physical Parameters ------------
39 % Skin Layer params
40 rho    = [1150 1116 900];   % [kg/m^3]  Tissue density (layers)
41 c      = [3590 3300 2500];  % [J/(k g   C)]  Tissue specific heat
42 k      = [0.2 0.45 0.3];    % [W/(m   C)]  Thermal conductivity
43 k_star = [0.1 0.1 0.1];     % [W/(m   C  s)]  Hyperbolicterm coefficient
44
45 % Blood params
46 h      = 4.5;      % [W/(m^2  C  )]    Convective coefficient at boundaries
47 wb     = 0.0098;   % [1/s]           Blood perfusion
48 rho_b  = 1056;     % [kg/m^3]        Blood density
49 cb     = 4000;     % [J/(k g   C)]     Blood specific heat
50
51 Qm0    = 50.65;    % [W/m^3]         Metabolic heat (uniform)
52 Tb     = 37;       % [ C ]            Arterial blood temperature
53 Tl     = [37 37 37];      % [ C ]    Initial Ambient/tissue reference
54
55 % ------------ Gaussiansource parameters ------------
56 % Q_r(i) = rho * S * P * exp( -a0*( x(i) - x_ast )^2 )
57 S      = 15;        %      perkg      scaling factor
58 P      = 35;       %    power    factor (tune as needed)
59 a0     = 1e6;    % [1/m]  Gaussian width control
60
```

```
% ------------ Water vaporization and diffusion ------------
Da = 2.5e-5;     % m^2/s (air)
Mw = 0.018;      % kg/mol
Ra = 8.314;      % J/mol K
Tw = 306;        % K (~33 C )
Pw = 5600;       % Pa (sat. vapor pressure at 33 C )
RH = 0.5;        % Relative humidity (fraction)
delta = 1e-4;    % m
c_air = 1005;    % J/kg K
Df = [2e-9 2e-9 2e-9];  % m^2/s
cw = 4180;       % J/kg C
rho_s = 1100;    % kg/m^3
rho_c = 1000;    % kg/m^3
nabla_r2 = (Lx/3)^2;

% ---------- Hyperbolic relaxation times ---------------
tau_q = 10;    % [s]
tau_T = 20;    % [s]
tau_v = 1;    % [s]

%% 3) INITIALIZE STORAGE FOR    SNAPSHOTS
% Snapshot times (in seconds) and their corresponding loop indices
snapshot_times = linspace(2,max_time,5); % [2.5, 5.0, 7.5, 10.0];
snap_idx = round(snapshot_times/dt) + 1;
% e.g., 2.5/0.01 = 250     +1 = 251     t = (251  1  )*0.01 = 2.50 s

T_snapshots = zeros(nx, numel(snapshot_times));
% Each column j holds T(x) at t = snapshot_times(j)

% Tumor temp array
T_tumor = zeros(1, time_steps);

% --------- Initialize temperature fields at t=0 -------------
T     = ones(nx,1);
T(find(layer == 1)) = Tl(1); T(find(layer == 2)) = Tl(2);
T(find(layer == 3)) = Tl(3);
T_new = T;
dTdt   = zeros(nx,1);
d2Tdt2 = zeros(nx,1);

%% 4)  TIMEMARCHING  LOOP
for n = 1:time_steps

    % (a) Update interior nodes i=2:( n x 1 )
    for i = 2:(nx-1)
        L = layer(i);

        % (b) Calculate Heat Source values for given spatial step
        % Metabolic Heat Source:
        Qm = Qm0 * (1 + (T(i) - Tl(L))/10);         % [W/m^3] metabolic

        % Water diffusion:
        Qd = (Df(L) * cw * (rho_s - rho_c) / nabla_r2) * (T(i) - Tl(L));

        % Blood Perfusion (dermis, subq):
        if L > 1
            Qb = wb * rho_b * cb * (Tb - Tl(L));      % [W/m^3] perfusion (constant
                )
        else
            Qb = 0;
        end

        % Water vaporization (epidermis only)
        if L == 1
            Delta_m = (Da*Mw/(Ra*Tw)) * (Pw/Tw) * RH/(delta*c_air);
            Delta_Hvap = 2400e3; % J/kg
            Qv = Delta_m * Delta_Hvap / (delta * c_air);
```

```
        else
            Qv = 0;
        end

        % Gaussian tumor heat source at node i:
        Qr = rho(L) * S * P * exp( -a0 * (x(i) - x_ast)^2 );

        % Add up all heat sources to consolidate
        Q_tot = Qm + Qd + Qb + Qv + Qr;

        % 2 n d spatial  derivative (finite difference)
        d2Tdx2 = (T(i+1) - 2*T(i) + T(i-1)) / dx^2;

        % Hyperbolic bioheat terms:
        dTdt(i)   = (k(L)*d2Tdx2 + Q_tot) / (rho(L) * c(L));
        d2Tdt2(i) = (k_star(L)*d2Tdx2) / (rho(L) * c(L));

        % Update temperature at node i (eq. 7):
        T_new(i) = T(i) + dt * ( ...
           dTdt(i) + tau_q*dTdt(i) ...
         - tau_T*d2Tdt2(i) ...
         + (k(L) + k_star(L)*tau_v)*dTdt(i) );
    end

    % (c) Convective (Robin) BC at left boundary (i = 1):
    % Forward Euler Approx dx/t
    T_new(1) = (k(1) * T(2) + h*dx*22) / (h*dx + k(1)); % 22C is RT

    % (d) Convective (Robin) BC at right boundary (i = nx):
    % Backward Euler Approx dx/dt
    T_new(nx) = (h*dx*37 - k(3) * T(end-1)) / (h*dx - k(3)); % 37C is body temp

    % (e) Advance to next timestep
    T = T_new;

    % (f) If current    t    matches any snapshot index, store T(x)
    idx_this = find(snap_idx == n);
    if ~isempty(idx_this)
        T_snapshots(:, idx_this) = T;
    end

    % (g) Store T at tumor location
    T_tumor(n) = T_new(iTumor);
end

%% 5) PLOT ALL T(x) PROFILES IN A SINGLE FIGURE
figure('Position',[200,200,800,500]);
hold on;

colors = lines(numel(snapshot_times));
for j = 1:numel(snapshot_times)
    plot(x*1000, T_snapshots(:,j), 'Color', colors(j,:), 'LineWidth',1.5, ...
         'DisplayName', sprintf('t = %.1f s', snapshot_times(j)));
end

ylim([36 50])
xlim([0 Lx*10^3])
xline(0, '-',{'Epiderm'},'HandleVisibility','off','Fontsize',16)
xline((L_epi)*10^3,'-',{'Derm'},'HandleVisibility','off','Fontsize',16)
xline((L_epi+L_derm)*10^3,'-',{'Subcutaneous'},'HandleVisibility','off','Fontsize'
    ,16)
xline(x_ast*10^3,'-r',{'Tumor'},'LineWidth',5,'HandleVisibility','off','Fontsize'
    ,20)
xlabel('Distance (mm)','Fontsize',16);
ylabel('Temperature ( C )','Fontsize',16);
title(['Temperature Profile T(t=10s,x), Various times'],'Fontsize',20);
legend('Location','northeast','FontSize',16);
```

```
grid on;
hold off;

%% 6. PLOT: TUMOR TEMPERATURE VS TIME
figure;
plot(time(1:length(T_tumor)), T_tumor, 'r', 'LineWidth', 2);
xlim([0 time(end)])
ylim([36 50])
xlabel('time (s)','Fontsize',16);
ylabel('Temperature ( C )','Fontsize',16);
title('Temperature at Tumor Location vs. Time','FontSize',20);
grid on;
```

Listing 1: main_final.m

6.2 changing_tau_final.m

```
%


% 1D hyperbolic bioheat w/ Gaussian tumor source, 3 skin layers, finer mesh
% Computes T(x) at t=10 s for three different tau_T / tau_v / tau_q values.
% Then plots each triplet in a 1 3  subplot.
%



clear; clc;

%% 1) DISCRETIZE TIME & SPATIAL DOMAINS
% Skin Layer thicknesses
L_epi  = 0.0015;   % Epidermis (1.5 mm)
L_derm = 0.0035;   % Dermis   (3.5 mm)
L_subq = 0.01;     % Subcutaneous (10 mm)
Lx     = L_epi + L_derm + L_subq;  % Total slab thickness (m)

x_ast  = 0.008;    % [m] Tumor center (8 mm from surface)

dx         = 0.0005;          % [m] spatial step      (Lx/dx + 1) nodes
dt         = 0.015;           % [s] time step (for stability)
max_time   = 10;              % [s] we want solution at t=10 s
time_steps = round(max_time/dt) + 1;  % ensures (time_steps1)*dt     10
time       = (0:(time_steps-1)) * dt;

x     = 0:dx:Lx;    % grid from 0 to Lx in steps of dx
nx    = numel(x);

% Build a    layer    index for each x(i):
layer = zeros(1,nx);
layer(x <= L_epi) = 1;                                  % Epidermis
layer(x > L_epi & x <= L_epi + L_derm) = 2;             % Dermis
layer(x > L_epi + L_derm) = 3;                          % Subcutaneous

% Find the grid index closest to x_ast for tumorlocation tracking:
[~, iTumor] = min(abs(x - x_ast));

%% 2) CONSTANTS AND PARAMETERS (unchanged except   s  )
% Physical parameters per layer:
rho    = [1150, 1116, 900];    % [kg/m^3] densities for [epi, derm, subq]
c      = [3590, 3300, 2500];   % [J/(kg  C)] specific heats
k      = [0.2, 0.45, 0.3];     % [W/(m  C)] thermal conductivities
k_star = [0.1, 0.1, 0.1];       % [W/(m  C  s)] hyperbolicterm coefficient

% Blood and convection:
h     = 4.5;       % [W/(m^2  C  )] convectiveBC coefficient
wb    = 0.0098;    % [1/s] blood perfusion rate
```

```
rho_b = 1056;       % [kg/m^3] blood density
cb    = 4000;       % [J/(kg  C)] blood specific heat

Qm0 = 50.65;        % [W/m^3] metabolic heat
Tb  = 37;           % [ C ]   blood inlet temperature
Tl  = [37, 37, 37];% [ C ]   ambient/tissue reference for each layer
T0  = 37;           % [ C ]   initial tissue temperature

% Gaussian source parameters (same for all runs):
S     = 15;         % [unitless]     perkg     scaling
P     = 35;         % [unitless]    power    factor
a0    = 1e6;        % [1/m^2] Gaussian width control
                    % Q_r(i) = rho(layer(i)) * S * P * exp(-a0*(x(i)-x_ast)^2)

%% 3) PREALLOCATE    SNAPSHOT    MATRICES
% Each final time profile has 3 columns, one for each tau value in that set:
T_tauT_snap = zeros(nx, 3);
T_tauV_snap = zeros(nx, 3);
T_tauQ_snap = zeros(nx, 3);

% We will hold the    default    values for the taus not being varied:
tauT_default = 20;    % [s]
tauV_default = 1;     % [s]
tauQ_default = 10;    % [s]

%% 4) VARY  _T      {2,3,4}  (fix  _v =1,  _q =10)
tauT_list = [1, 10, 20]; % values over 25 begin to look unstable
for j = 1:3
    tau_T = tauT_list(j);
    tau_v = tauV_default;
    tau_q = tauQ_default;

    % Initialize temperature fields at t=0:
    T     = ones(nx,1) * T0;
    T_new = T;

    % Time marching loop to t=10 s:
    for n = 1:time_steps
        for i = 2:(nx-1)
            L = layer(i);

            % 1) Metabolic heat (layer-dependent Tl):
            Qm = Qm0 * (1 + (T(i) - Tl(L))/10);
            % 2) Water diffusion (not varying here, same as original):
            Da     = 2.5e-5;  Mw = 0.018;  Ra  = 8.314;
            Tw     = 306;     Pw = 5600;   RH  = 0.5;
            delta  = 1e-4;    c_air = 1005;
            Df     = [2e-9, 2e-9, 2e-9];  cw = 4180;
            rho_s  = 1100;    rho_c = 1000; nabla_r2 = (Lx/3)^2;
            Qd     = (Df(L)*cw*(rho_s-rho_c)/nabla_r2)*(T(i)-Tl(L));

            % 3) Perfusion (dermis+subq only):
            if L > 1
                Qb = wb * rho_b * cb * (Tb - Tl(L));
            else
                Qb = 0;
            end

            % 4) Vaporization (epidermis only):
            if L == 1
                Delta_m    = (Da*Mw/(Ra*Tw)) * (Pw/Tw) * RH/(delta*c_air);
                Delta_Hvap = 2400e3; % J/kg
                Qv         = Delta_m * Delta_Hvap / (delta*c_air);
            else
                Qv = 0;
            end

```

```
            % 5) Gaussian tumor source:
            Qr = rho(L) * S * P * exp(-a0*(x(i)-x_ast)^2);

            % 6) Sum all sources:
            Q_tot = Qm + Qd + Qb + Qv + Qr;

            % 7) Seconddderivative in x:
            d2Tdx2 = (T(i+1) - 2*T(i) + T(i-1)) / dx^2;

            % 8) Hyperbolic bioheat:
            dTdt(i)   = (k(L)*d2Tdx2 + Q_tot) / (rho(L)*c(L));
            d2Tdt2(i) = (k_star(L)*d2Tdx2) / (rho(L)*c(L));

            T_new(i) = T(i) + dt * ( ...
               dTdt(i) + tau_q*dTdt(i) ...
             - tau_T*d2Tdt2(i) ...
             + (k(L) + k_star(L)*tau_v) * dTdt(i) );
        end

        % Convective BC at i=1 (left):
        L = layer(1);
        T_new(1)  = (h*dx*Tl(L) + k(L)*T_new(2)) / (h*dx + k(L));
        % Convective BC at i=nx (right):
        L = layer(nx);
        T_new(nx) = (h*dx*Tl(L) + k(L)*T_new(nx-1)) / (h*dx + k(L));

        % Advance:
        T = T_new;
    end

    % After 10 s, store the final profile in column j:
    T_tauT_snap(:,j) = T;
end

%% 5) VARY _v      {1,2,3}  (fix _T =2,  _q =10)
tauV_list = [1, 10, 20]; % minimizing works best
for j = 1:3
    tau_T = tauT_default;
    tau_v = tauV_list(j);
    tau_q = tauQ_default;

    % Initialize:
    T     = ones(nx,1) * T0;
    T_new = T;

    for n = 1:time_steps
        for i = 2:(nx-1)
            L = layer(i);

            % (Same source calculations as above)
            Qm = Qm0 * (1 + (T(i) - Tl(L))/10);
            Da     = 2.5e-5;  Mw = 0.018;  Ra  = 8.314;
            Tw     = 306;     Pw = 5600;  RH  = 0.5;
            delta  = 1e-4;    c_air = 1005;
            Df     = [2e-9,2e-9,2e-9];  cw = 4180;
            rho_s  = 1100;   rho_c = 1000; nabla_r2 = (Lx/3)^2;
            Qd     = (Df(L)*cw*(rho_s - rho_c)/nabla_r2)*(T(i) - Tl(L));

            if L > 1
                Qb = wb * rho_b * cb * (Tb - Tl(L));
            else
                Qb = 0;
            end

            if L == 1
                Delta_m    = (Da*Mw/(Ra*Tw)) * (Pw/Tw) * RH/(delta*c_air);
                Delta_Hvap = 2400e3;
```

```
                Qv         = Delta_m * Delta_Hvap / (delta*c_air);
            else
                Qv = 0;
            end

            Qr = rho(L) * S * P * exp(-a0*(x(i) - x_ast)^2);
            Q_tot = Qm + Qd + Qb + Qv + Qr;

            d2Tdx2 = (T(i+1) - 2*T(i) + T(i-1)) / dx^2;
            dTdt(i)   = (k(L)*d2Tdx2 + Q_tot) / (rho(L)*c(L));
            d2Tdt2(i) = (k_star(L)*d2Tdx2) / (rho(L)*c(L));

            T_new(i) = T(i) + dt * ( ...
               dTdt(i) + tau_q*dTdt(i) ...
             - tau_T*d2Tdt2(i) ...
             + (k(L) + k_star(L)*tau_v) * dTdt(i) );
        end

        % BCs at i=1 and i=nx:
        L = layer(1);
        T_new(1)  = (h*dx*Tl(L) + k(L)*T_new(2))   / (h*dx + k(L));
        L = layer(nx);
        T_new(nx) = (h*dx*Tl(L) + k(L)*T_new(nx-1)) / (h*dx + k(L));

        T = T_new;
    end

    T_tauV_snap(:,j) = T;
end

%% 6) VARY  _q      {5,10,15}  (fix  _T =2,  _v =1)
tauQ_list = [7, 10, 15]; % below 8, unstable -- above 15, localization decreases
for j = 1:3
    tau_T = tauT_default;
    tau_v = tauV_default;
    tau_q = tauQ_list(j);

    T     = ones(nx,1) * T0;
    T_new = T;

    for n = 1:time_steps
        for i = 2:(nx-1)
            L = layer(i);

            Qm = Qm0 * (1 + (T(i) - Tl(L))/10);
            Da     = 2.5e-5;  Mw = 0.018;  Ra  = 8.314;
            Tw     = 306;     Pw = 5600;   RH  = 0.5;
            delta  = 1e-4;    c_air = 1005;
            Df     = [2e-9,2e-9,2e-9];  cw = 4180;
            rho_s  = 1100;    rho_c = 1000; nabla_r2 = (Lx/3)^2;
            Qd     = (Df(L)*cw*(rho_s - rho_c)/nabla_r2)*(T(i)-Tl(L));

            if L > 1
                Qb = wb * rho_b * cb * (Tb - Tl(L));
            else
                Qb = 0;
            end

            if L == 1
                Delta_m    = (Da*Mw/(Ra*Tw)) * (Pw/Tw) * RH/(delta*c_air);
                Delta_Hvap = 2400e3;
                Qv         = Delta_m * Delta_Hvap / (delta*c_air);
            else
                Qv = 0;
            end

            Qr = rho(L) * S * P * exp(-a0*(x(i)-x_ast)^2);
```

```
            Q_tot = Qm + Qd + Qb + Qv + Qr;

            d2Tdx2 = (T(i+1) - 2*T(i) + T(i-1)) / dx^2;
            dTdt(i)   = (k(L)*d2Tdx2 + Q_tot) / (rho(L)*c(L));
            d2Tdt2(i) = (k_star(L)*d2Tdx2) / (rho(L)*c(L));

            T_new(i) = T(i) + dt * ( ...
               dTdt(i) + tau_q*dTdt(i) ...
             - tau_T*d2Tdt2(i) ...
             + (k(L) + k_star(L)*tau_v) * dTdt(i) );
        end

        % BCs:
        L = layer(1);
        T_new(1)  = (h*dx*Tl(L) + k(L)*T_new(2))    / (h*dx + k(L));
        L = layer(nx);
        T_new(nx) = (h*dx*Tl(L) + k(L)*T_new(nx-1)) / (h*dx + k(L));

        T = T_new;
    end

    T_tauQ_snap(:,j) = T;
end

%% 7) PLOT ALL THREE SETS IN A SINGLE 1 3  SUBPLOT FIGURE
figure('Position',[200,200,1200,400]);

% 7.1) Subplot #1: varying tau_T
subplot(1,3,1);
plot(x*1000, T_tauT_snap(:,1), 'b', 'LineWidth',1.5);  hold on;
plot(x*1000, T_tauT_snap(:,2), 'r','LineWidth',1.5);
plot(x*1000, T_tauT_snap(:,3), 'g','LineWidth',1.5);
ylim([36 50]);
xlabel('Distance (mm)');
ylabel('Temperature ( C )');
title('Temperature profile T (t = 10 s), varying \tau_T');
legend('\tau_T = 1','\tau_T = 10','\tau_T = 20','Location','best');
grid on;
hold off;

% 7.2) Subplot #2: varying tau_v
subplot(1,3,2);
plot(x*1000, T_tauV_snap(:,1), 'b', 'LineWidth',1.5);  hold on;
plot(x*1000, T_tauV_snap(:,2), 'r','LineWidth',1.5);
plot(x*1000, T_tauV_snap(:,3), 'g','LineWidth',1.5);
ylim([36 50]);
xlabel('Distance (mm)');
title('Temperature profile T (t = 10 s), varying \tau_v');
legend('\tau_v = 1','\tau_v = 10','\tau_v = 20','Location','best');
grid on;
hold off;

% 7.3) Subplot #3: varying tau_q
subplot(1,3,3);
plot(x*1000, T_tauQ_snap(:,1), 'b', 'LineWidth',1.5);  hold on;
plot(x*1000, T_tauQ_snap(:,2), 'r','LineWidth',1.5);
plot(x*1000, T_tauQ_snap(:,3), 'g','LineWidth',1.5);
ylim([36 50]);
xlabel('Distance (mm)');
title('Temperature profile T (t = 10 s), varying \tau_q');
legend('\tau_q = 7','\tau_q = 10','\tau_q = 15','Location','best');
grid on;
hold off;
```

Listing 2: changing_tau_final.m

6.3 a0_test_final

```
%


% a0_test_final.m
% 1D hyperbolic bioheat w/ Gaussian tumor source, 3 skin layers, finer mesh
% and reduced dt for stability; Varying a0 to find optimal parameter
%



clear; clc;

%% 1) DISCRETIZE TIME & SPATIAL DOMAINS
% Skin Layer thickness
L_epi = 0.0015;    % Epidermis (1.5 mm)
L_derm = 0.0035;   % Dermis (3.5 mm)
L_subq = 0.01;     % Subcutaneous (10 mm)
Lx = L_epi + L_derm + L_subq; % Domain Length

x_ast  = 0.008;    % [m]   Tumor center (midpoint of 0.05 m)

dx     = 0.0005;   % [m]   Spatial step     now 21 nodes from 0 to 0.05
dt     = 0.015;     % [s]   Time step (smaller for stability)
max_time = 10;    % plot up to 10s
time_steps = round(max_time/dt) + 1; % ensures we reach exactly 10 s: ( n 1 )*dt =
    10
time = 0:dt:max_time+dt;

x  = 0:dx:Lx;          % x = [0, 0.0025, 0.005,    , 0.05]
nx = numel(x);

% Define Layers
layer = zeros(1, nx);
layer(x <= L_epi) = 1;                                    % Epidermis
layer(x > L_epi & x <= L_epi + L_derm) = 2;               % Dermis
layer(x > L_epi + L_derm) = 3;                            % Subcutaneous

% Find index of tumor center (closest grid point to x_ast)
[~, iTumor] = min(abs(x - x_ast));

%% 2) CONSTANTS AND PARAMETERS
% ------------ Physical Parameters ------------
% Skin Layer params
rho    = [1150 1116 900];   % [kg/m^3]  Tissue density (layers)
c      = [3590 3300 2500];  % [J/(k g   C)]  Tissue specific heat
k      = [0.2 0.45 0.3];    % [W/(m   C)]  Thermal conductivity
k_star = [0.1 0.1 0.1];     % [W/(m   C  s)]  Hyperbolicterm coefficient

% Blood params
h      = 4.5;      % [W/(m^2  C  )]    Convective coefficient at boundaries
wb     = 0.0098;   % [1/s]           Blood perfusion
rho_b  = 1056;     % [kg/m^3]        Blood density
cb     = 4000;     % [J/(k g   C)]     Blood specific heat

Qm0    = 50.65;    % [W/m^3]         Metabolic heat (uniform)
Tb     = 37;       % [ C ]            Arterial blood temperature
Tl     = [37 37 37];      % [ C ]    Initial Ambient/tissue reference

% ------------ Gaussiansource parameters ------------
% Q_r(i) = rho * S * P * exp( -a0*( x(i) - x_ast )^2 )
S      = 15;        %      perkg      scaling factor
P      = 35;       %    power    factor (tune as needed)
% a0     = 1e6;    % [1/m]  Gaussian width control

% ------------ Water vaporization and diffusion ------------
```

```
Da = 2.5e-5;     % m^2/s (air)
Mw = 0.018;      % kg/mol
Ra = 8.314;      % J/mol K
Tw = 306;        % K (~33 C )
Pw = 5600;       % Pa (sat. vapor pressure at 33 C )
RH = 0.5;        % Relative humidity (fraction)
delta = 1e-4;    % m
c_air = 1005;    % J/kg K
Df = [2e-9 2e-9 2e-9];  % m^2/s
cw = 4180;       % J/kg C
rho_s = 1100;    % kg/m^3
rho_c = 1000;    % kg/m^3
nabla_r2 = (Lx/3)^2;

% ---------- Hyperbolic relaxation times ---------------
tau_q = 10;    % [s]
tau_T = 20;    % [s]
tau_v = 1;    % [s]


%% 3) INITIALIZE STORAGE FOR    SNAPSHOTS
% Snapshot times (in seconds) and their corresponding loop indices
snapshot_times = linspace(2,max_time,5); % [2.5, 5.0, 7.5, 10.0];
snap_idx = round(snapshot_times/dt) + 1;
% e.g., 2.5/0.01 = 250     +1 = 251     t = (251  1  )*0.01 = 2.50 s

T_a0 = zeros(nx,4); % For storing T(x,t=10s) for a0 testing
T_tumor_a0 = zeros(time_steps,4); % for storing tumor temp for a0 testing

z = 1;
a0s = [5e5 1e6 2e6 5e6]
for a0 = a0s
    T_snapshots = zeros(nx, numel(snapshot_times));
    % Each column j holds T(x) at t = snapshot_times(j)

    % Tumor temp array
    T_tumor = zeros(1, time_steps);

    % --------- Initialize temperature fields at t=0 -------------
    T = ones(nx,1);
    T(find(layer == 1)) = Tl(1); T(find(layer == 2)) = Tl(2);
    T(find(layer == 3)) = Tl(3);
    T_new = T;
    dTdt   = zeros(nx,1);
    d2Tdt2 = zeros(nx,1);


    %% 4)  TIME MARCHING LOOP
    for n = 1:time_steps

        % (a) Update interior nodes i=2:( n x 1 )
        for i = 2:(nx-1)
            L = layer(i);

            % (b) Calculate Heat Source values for given spatial step
            % Metabolic Heat Source:
            Qm = Qm0 * (1 + (T(i) - Tl(L))/10);         % [W/m^3] metabolic

            % Water diffusion:
            Qd = (Df(L) * cw * (rho_s - rho_c) / nabla_r2) * (T(i) - Tl(L));

            % Blood Perfusion (dermis, subq):
            if L > 1
                Qb = wb * rho_b * cb * (Tb - Tl(L));      % [W/m^3] perfusion (
                    constant)
            else
                Qb = 0;
```

```
            end

            % Water vaporization (epidermis only)
            if L == 1
                Delta_m = (Da*Mw/(Ra*Tw)) * (Pw/Tw) * RH/(delta*c_air);
                Delta_Hvap = 2400e3; % J/kg
                Qv = Delta_m * Delta_Hvap / (delta * c_air);
            else
                Qv = 0;
            end

            % Gaussian tumor heat source at node i:
            Qr = rho(L) * S * P * exp( -a0 * (x(i) - x_ast)^2 );

            % Add up all heat sources to consolidate
            Q_tot = Qm + Qd + Qb + Qv + Qr;

            % 2 n d spatial  derivative (finite difference)
            d2Tdx2 = (T(i+1) - 2*T(i) + T(i-1)) / dx^2;

            % Hyperbolic bioheat terms:
            dTdt(i)   = (k(L)*d2Tdx2 + Q_tot) / (rho(L) * c(L));
            d2Tdt2(i) = (k_star(L)*d2Tdx2) / (rho(L) * c(L));

            % Update temperature at node i (eq. 7):
            T_new(i) = T(i) + dt * ( ...
               dTdt(i) + tau_q*dTdt(i) ...
             - tau_T*d2Tdt2(i) ...
             + (k(L) + k_star(L)*tau_v)*dTdt(i) );
        end

        % (c) Convective (Robin) BC at left boundary (i = 1):
        % Forward Euler Approx dx/t
        T_new(1) = (k(1) * T(2) + h*dx*22) / (h*dx + k(1)); % 22C is RT

        % (d) Convective (Robin) BC at right boundary (i = nx):
        % Backward Euler Approx dx/dt
        T_new(nx) = (h*dx*37 - k(3) * T(end-1)) / (h*dx - k(3)); % 37C is body temp

        % (e) Advance to next timestep
        T = T_new;

        % (f) If current    t    matches any snapshot index, store T(x)
        idx_this = find(snap_idx == n);
        if ~isempty(idx_this)
            T_snapshots(:, idx_this) = T;
        end

        % (g) Store T at tumor location
        T_tumor(n) = T_new(iTumor);
    end

    %% Store profile for given P to be compared after
    T_a0(:,z) = T_snapshots(:,end);
    T_tumor_a0(:,z) = T_tumor;
    z = z+1;

end % end of testing parameter loop (power)

%% Plot T(x) for a0s
figure('Position',[200,200,800,500]);
hold on;

colors = lines(numel(a0s));
for j = 1:size(T_a0,2)
    plot(x*1000, T_a0(:,j), 'Color', colors(j,:), 'LineWidth',1.5, ...
         'DisplayName', sprintf('a0 = %.1e m^-^1', a0s(j)));
```

```
end

ylim([36 50])
xlim([0 Lx*10^3])
xline(0, '-',{'Epiderm'},'HandleVisibility','off','Fontsize',16)
xline((L_epi)*10^3,'-',{'Derm'},'HandleVisibility','off','Fontsize',16)
xline((L_epi+L_derm)*10^3,'-',{'Subcutaneous'},'HandleVisibility','off','Fontsize'
    ,16)
xline(x_ast*10^3,'-r',{'Tumor'},'LineWidth',5,'HandleVisibility','off','Fontsize'
    ,20)
xlabel('Distance (mm)','Fontsize',16);
ylabel('Temperature ( C )','Fontsize',16);
title(['Temperature Profile T(t=10s,x), Various a0s'],'Fontsize',20);
legend('Location','northeast','FontSize',12);
grid on;
hold off;


%% 6. PLOT: TUMOR TEMPERATURE VS TIME
figure('Position',[200,200,800,500]);
hold on;

colors = lines(numel(a0s));
for j = 1:size(T_a0,2)
    plot(time, T_tumor_a0(:,j), 'Color', colors(j,:), 'LineWidth',1.5, ...
         'DisplayName', sprintf('a0 = %.1e m^-^1', a0s(j)));
end

xlim([0 time(end)])
ylim([36 50])
xlabel('time (s)','Fontsize',16);
ylabel('Temperature ( C )','Fontsize',16);
title(['Temperature at Tumor Location vs. Time, Various a0s'],'Fontsize',20);
legend('Location','best','FontSize',12);
grid on;
hold off;
```

Listing 3: a0_test_final.m

6.4 power_test_final.m

```
%


% power_test_final.m
% 1D hyperbolic bioheat w/ Gaussian tumor source, 3 skin layers, finer mesh
% and reduced dt for stability; Varying Power to find optimal parameter
%



clear; clc;

%% 1) DISCRETIZE TIME & SPATIAL DOMAINS
% Skin Layer thickness
L_epi = 0.0015;    % Epidermis (1.5 mm)
L_derm = 0.0035;   % Dermis (3.5 mm)
L_subq = 0.01;     % Subcutaneous (10 mm)
Lx = L_epi + L_derm + L_subq; % Domain Length

x_ast  = 0.008;    % [m]   Tumor center (midpoint of 0.05 m)

dx     = 0.0005;   % [m]   Spatial step     now 21 nodes from 0 to 0.05
dt     = 0.015;     % [s]   Time step (smaller for stability)
max_time = 10;    % plot up to 10s
```

```
time_steps = round(max_time/dt) + 1; % ensures we reach exactly 10 s: ( n 1 )*dt =
    10
time = 0:dt:max_time+dt;

x  = 0:dx:Lx;          % x = [0, 0.0025, 0.005,    , 0.05]
nx = numel(x);

% Define Layers
layer = zeros(1, nx);
layer(x <= L_epi) = 1;                                    % Epidermis
layer(x > L_epi & x <= L_epi + L_derm) = 2;               % Dermis
layer(x > L_epi + L_derm) = 3;                            % Subcutaneous

% Find index of tumor center (closest grid point to x_ast)
[~, iTumor] = min(abs(x - x_ast));

%% 2) CONSTANTS AND PARAMETERS
% ------------ Physical Parameters ------------
% Skin Layer params
rho    = [1150 1116 900];   % [kg/m^3]  Tissue density (layers)
c      = [3590 3300 2500];  % [J/(k g   C)]  Tissue specific heat
k      = [0.2 0.45 0.3];    % [W/(m   C)]  Thermal conductivity
k_star = [0.1 0.1 0.1];     % [W/(m   C  s)]  Hyperbolicterm coefficient

% Blood params
h      = 4.5;      % [W/(m^2  C  )]    Convective coefficient at boundaries
wb     = 0.0098;   % [1/s]           Blood perfusion
rho_b  = 1056;     % [kg/m^3]        Blood density
cb     = 4000;     % [J/(k g   C)]     Blood specific heat

Qm0    = 50.65;    % [W/m^3]         Metabolic heat (uniform)
Tb     = 37;       % [ C ]            Arterial blood temperature
Tl     = [37 37 37];      % [ C ]    Initial Ambient/tissue reference

% ------------ Gaussiansource parameters ------------
% Q_r(i) = rho * S * P * exp( -a0*( x(i) - x_ast )^2 )
S      = 15;        %     perkg     scaling factor
% P    = 30;       %    power    factor (tune as needed)
a0     = 1e6;    % [1/m]  Gaussian width control

% ------------ Water vaporization and diffusion ------------
Da = 2.5e-5;     % m^2/s (air)
Mw = 0.018;      % kg/mol
Ra = 8.314;      % J/mol K
Tw = 306;        % K (~33 C )
Pw = 5600;       % Pa (sat. vapor pressure at 33 C )
RH = 0.5;        % Relative humidity (fraction)
delta = 1e-4;    % m
c_air = 1005;    % J/kg K
Df = [2e-9 2e-9 2e-9];  % m^2/s
cw = 4180;       % J/kg C
rho_s = 1100;    % kg/m^3
rho_c = 1000;    % kg/m^3
nabla_r2 = (Lx/3)^2;

% ---------- Hyperbolic relaxation times ---------------
tau_q = 10;    % [s]
tau_T = 20;    % [s]
tau_v = 1;    % [s]


%% 3) INITIALIZE STORAGE FOR    SNAPSHOTS
% Snapshot times (in seconds) and their corresponding l o o p indices
snapshot_times = linspace(2,max_time,5); % [2.5, 5.0, 7.5, 10.0];
snap_idx = round(snapshot_times/dt) + 1;
% e.g., 2.5/0.01 = 250     +1 = 251     t = (251  1  )*0.01 = 2.50 s

```

```
T_p = zeros(nx,4); % For storing T(x,t=10s) for power testing
T_tumor_p = zeros(time_steps,4); % for storing tumor temp for power testing

z = 1;
powers = 10:5:50
for P = powers
    T_snapshots = zeros(nx, numel(snapshot_times));
    % Each column j holds T(x) at t = snapshot_times(j)

    % Tumor temp array
    T_tumor = zeros(1, time_steps);

    % --------- Initialize temperature fields at t=0 -------------
    T     = ones(nx,1) * Tl(1);
    T(find(layer == 1)) = Tl(1); T(find(layer == 2)) = Tl(2);
    T(find(layer == 3)) = Tl(3);
    T_new = T;
    dTdt   = zeros(nx,1);
    d2Tdt2 = zeros(nx,1);


    %% 4)  T I M E MARCHING  LOOP
    for n = 1:time_steps

        % (a) Update interior nodes i=2:( n x 1 )
        for i = 2:(nx-1)
            L = layer(i);

            % (b) Calculate Heat Source values for given spatial step
            % Metabolic Heat Source:
            Qm = Qm0 * (1 + (T(i) - Tl(L))/10);          % [W/m^3] metabolic

            % Water diffusion:
            Qd = (Df(L) * cw * (rho_s - rho_c) / nabla_r2) * (T(i) - Tl(L));

            % Blood Perfusion (dermis, subq):
            if L > 1
                Qb = wb * rho_b * cb * (Tb - Tl(L));      % [W/m^3] perfusion (
                    constant)
            else
                Qb = 0;
            end

            % Water vaporization (epidermis only)
            if L == 1
                Delta_m = (Da*Mw/(Ra*Tw)) * (Pw/Tw) * RH/(delta*c_air);
                Delta_Hvap = 2400e3; % J/kg
                Qv = Delta_m * Delta_Hvap / (delta * c_air);
            else
                Qv = 0;
            end

            % Gaussian tumor heat source at node i:
            Qr = rho(L) * S * P * exp( -a0 * (x(i) - x_ast)^2 );

            % Add up all heat sources to consolidate
            Q_tot = Qm + Qd + Qb + Qv + Qr;

            % 2 n d spatial  derivative (finite difference)
            d2Tdx2 = (T(i+1) - 2*T(i) + T(i-1)) / dx^2;

            % Hyperbolic bioheat terms:
            dTdt(i)   = (k(L)*d2Tdx2 + Q_tot) / (rho(L) * c(L));
            d2Tdt2(i) = (k_star(L)*d2Tdx2) / (rho(L) * c(L));

            % Update temperature at node i (eq. 7):
            T_new(i) = T(i) + dt * ( ...
```

```
                dTdt(i) + tau_q*dTdt(i) ...
            - tau_T*d2Tdt2(i) ...
            + (k(L) + k_star(L)*tau_v)*dTdt(i) );
        end

        % (c) Convective (Robin) BC at left boundary (i = 1):
        % Forward Euler Approx dx/t
        T_new(1) = (k(1) * T(2) + h*dx*22) / (h*dx + k(1)); % 22C is RT

        % (d) Convective (Robin) BC at right boundary (i = nx):
        % Backward Euler Approx dx/dt
        T_new(nx) = (h*dx*37 - k(3) * T(end-1)) / (h*dx - k(3)); % 37C is body temp

        % (e) Advance to next timestep
        T = T_new;

        % (f) If current    t    matches any snapshot index, store T(x)
        idx_this = find(snap_idx == n);
        if ~isempty(idx_this)
            T_snapshots(:, idx_this) = T;
        end

        % (g) Store T at tumor location
        T_tumor(n) = T_new(iTumor);
    end

    %% Store profile for given P to be compared after
    T_p(:,z) = T_snapshots(:,end);
    T_tumor_p(:,z) = T_tumor;
    z = z+1;

end % end of testing parameter loop (power)

%% Plot T(x) for Powers
figure('Position',[200,200,800,500]);
hold on;

colors = lines(numel(powers));
for j = 1:size(T_p,2)
    plot(x*1000, T_p(:,j), 'Color', colors(j,:), 'LineWidth',1.5, ...
         'DisplayName', sprintf('P = %.1f W', powers(j)));
end

ylim([36 50])
xlim([0 Lx*10^3])
xline(0, '-',{'Epiderm'},'HandleVisibility','off','Fontsize',16)
xline((L_epi)*10^3,'-',{'Derm'},'HandleVisibility','off','Fontsize',16)
xline((L_epi+L_derm)*10^3,'-',{'Subcutaneous'},'HandleVisibility','off','Fontsize'
    ,16)
xline(x_ast*10^3,'-r',{'Tumor'},'LineWidth',5,'HandleVisibility','off','Fontsize'
    ,20)
xlabel('Distance (mm)','Fontsize',16);
ylabel('Temperature ( C )','Fontsize',16);
title(['Temperature Profile T(t=10s,x), Various Powers'],'Fontsize',20);
legend('Location','northeast','FontSize',12);
grid on;
hold off;


%% 6. PLOT: TUMOR TEMPERATURE VS TIME
figure('Position',[200,200,800,500]);
hold on;

colors = lines(numel(powers));
for j = 1:size(T_p,2)
    plot(time, T_tumor_p(:,j), 'Color', colors(j,:), 'LineWidth',1.5, ...
         'DisplayName', sprintf('P = %.1f W', powers(j)));
```

```
end

xlim([0 time(end)])
ylim([36 50])
xlabel('time (s)','Fontsize',16);
ylabel('Temperature ( C )','Fontsize',16);
title(['Temperature at Tumor Location vs. Time, Various Powers'],'Fontsize',20);
legend('Location','best','FontSize',12);
grid on;
hold off;
```

Listing 4: power_test_final.m

6.5 skin_layer_comparison.m

```
%


% 1D hyperbolic bioheat w/ Gaussian tumor source, 3 skin layers, finer mesh
% and reduced dt for stability; TESTING EFFECT OF SKIN LAYERS & HEAT
% SOURCES
%



clear; clc;

%% 1) DISCRETIZE TIME & SPATIAL DOMAINS
% Skin Layer thickness
L_epi = 0.0015;    % Epidermis (1.5 mm)
L_derm = 0.0035;   % Dermis (3.5 mm)
L_subq = 0.010;     % Subcutaneous (10 mm)
Lx = L_epi + L_derm + L_subq; % Domain Length

x_ast  = 0.008;    % [m]   Tumor center (midpoint of 0.05 m)

dx     = 0.0005;   % [m]   Spatial step     now 21 nodes from 0 to 0.05
dt     = 0.015;     % [s]   Time step (smaller for stability)
max_time = 10;    % plot up to 10s
time_steps = round(max_time/dt) + 1; % ensures we reach exactly 10 s: ( n 1 )*dt =
    10
time = 0:dt:max_time+dt;

x  = 0:dx:Lx;           % x = [0, 0.0025, 0.005,     , 0.05]
nx = numel(x);

% Define Layers
layer = zeros(1, nx);
layer(x <= L_epi) = 1;                                    % Epidermis
layer(x > L_epi & x <= L_epi + L_derm) = 2;               % Dermis
layer(x > L_epi + L_derm) = 3;                            % Subcutaneous

% Find index of tumor center (closest grid point to x_ast)
[~, iTumor] = min(abs(x - x_ast));

%% 2) CONSTANTS AND PARAMETERS
% ------------ Physical Parameters ------------
% Skin Layer params
% rho    = [1150 1116 900];    % [kg/m^3]  Tissue density (layers)
% c      = [3590 3300 2500];   % [J/(kg   C)]  Tissue specific heat
% k      = [0.2 0.45 0.3];     % [W/(m   C)]  Thermal conductivity
k_star = [0.1 0.1 0.1];      % [W/(m   C  s)]  Hyperbolicterm coefficient

% Blood params
h      = 4.5;      % [W/(m^2  C  )]    Convective coefficient at boundaries
wb     = 0.0098;   % [1/s]           Blood perfusion
```

```
rho_b  = 1056;     % [kg/m^3]         Blood density
cb     = 4000;     % [J/(k g   C)]      Blood specific heat

Qm0    = 50.65;    % [W/m^3]          Metabolic heat (uniform)
Tb     = 37;       % [ C ]             Arterial blood temperature
Tl     = [37 37 37];       % [ C ]     Initial Ambient/tissue reference

% ------------ Gaussiansource parameters ------------
% Q_r(i) = rho * S * P * exp( -a0*( x(i) - x_ast )^2 )
S      = 15;        %      perkg      scaling factor
P      = 35;       %    power    factor (tune as needed)
a0     = 1e6;    % [1/m]  Gaussian width control

% ------------ Water vaporization and diffusion ------------
Da = 2.5e-5;     % m^2/s (air)
Mw = 0.018;      % kg/mol
Ra = 8.314;      % J/mol K
Tw = 306;        % K (~33 C )
Pw = 5600;       % Pa (sat. vapor pressure at 33 C )
RH = 0.5;        % Relative humidity (fraction)
delta = 1e-4;    % m
c_air = 1005;    % J/kg K
Df = [2e-9 2e-9 2e-9];  % m^2/s?
cw = 4180;       % J/kg C
rho_s = 1100;    % kg/m^3
rho_c = 1000;    % kg/m^3
nabla_r2 = (Lx/3)^2;

% ---------- Hyperbolic relaxation times ---------------
tau_q = 10;    % [s]
tau_T = 20;    % [s]
tau_v = 1;    % [s]


%% 3) INITIALIZE STORAGE FOR    SNAPSHOTS
% Snapshot times (in seconds) and their corresponding loop indices
snapshot_times = linspace(2,max_time,5); % [2.5, 5.0, 7.5, 10.0];
snap_idx = round(snapshot_times/dt) + 1;
% e.g., 2.5/0.01 = 250     +1 = 251     t = (251  1  )*0.01 = 2.50 s

T_layers = zeros(nx,5); % For storing T(x,t=10s) for layer testing
T_tumor_layers = zeros(time_steps,5); % for storing tumor temp for layer testing

for layer_test = 1:5

    if layer_test == 1 % Heterogeneous case
        rho    = [1150 1116 900];    % [kg/m^3]  Tissue density (layers)
        c      = [3590 3300 2500];   % [J/(k g   C)]  Tissue specific heat
        k      = [0.2 0.45 0.3];     % [W/(m   C)]  Thermal conductivity
    elseif layer_test == 2 % Treat all as epidermis
        rho    = [1150 1150 1150];    % [kg/m^3]  Tissue density (layers)
        c      = [3590 3590 3590];   % [J/(k g   C)]  Tissue specific heat
        k      = [0.2 0.2 0.2];     % [W/(m   C)]  Thermal conductivity
    elseif layer_test == 3 % Treat all as dermis
        rho    = [1116 1116 1116];    % [kg/m^3]  Tissue density (layers)
        c      = [3300 3300 3300];   % [J/(k g   C)]  Tissue specific heat
        k      = [0.45 0.45 0.45];     % [W/(m   C)]  Thermal conductivity
    elseif layer_test == 4 | 5 % Treat all as subq
        rho    = [900 900 900];    % [kg/m^3]  Tissue density (layers)
        c      = [2500 2500 2500];   % [J/(k g   C)]  Tissue specific heat
        k      = [0.3 0.3 0.3];     % [W/(m   C)]  Thermal conductivity
    end

    T_snapshots = zeros(nx, numel(snapshot_times));
    % Each column j holds T(x) at t = snapshot_times(j)

    % Tumor temp array
```

```
    T_tumor = zeros(1, time_steps);

    % --------- Initialize temperature fields at t=0 ------------
    T     = ones(nx,1);
    T(find(layer == 1)) = Tl(1); T(find(layer == 2)) = Tl(2);
    T(find(layer == 3)) = Tl(3);
    T_new = T;
    dTdt   = zeros(nx,1);
    d2Tdt2 = zeros(nx,1);


    %% 4)  T I M E MARCHING  LOOP
    for n = 1:time_steps

        % (a) Update interior nodes i=2:( n x 1 )
        for i = 2:(nx-1)
            L = layer(i);

            % (b) Calculate Heat Source values for given spatial step
            % Metabolic Heat Source:
            Qm = Qm0 * (1 + (T(i) - Tl(L))/10);          % [W/m^3] metabolic

            % Water diffusion:
            Qd = (Df(L) * cw * (rho_s - rho_c) / nabla_r2) * (T(i) - Tl(L));

            % Blood Perfusion (dermis, subq):
            if L > 1
                Qb = wb * rho_b * cb * (Tb - Tl(L));      % [W/m^3] perfusion (
                    constant)
            else
                Qb = 0;
            end

            % Water vaporization (epidermis only)
            if L == 1
                Delta_m = (Da*Mw/(Ra*Tw)) * (Pw/Tw) * RH/(delta*c_air);
                Delta_Hvap = 2400e3; % J/kg
                Qv = Delta_m * Delta_Hvap / (delta * c_air);
            else
                Qv = 0;
            end

            % Gaussian tumor heat source at node i:
            Qr = rho(L) * S * P * exp( -a0 * (x(i) - x_ast)^2 );

            % Add up all heat sources to consolidate
            if layer_test == 5
                Qd = 0; Qv = 0;
            end
            Q_tot = Qm + Qd + Qb + Qv + Qr;

            % 2 n d spatial  derivative (finite difference)
            d2Tdx2 = (T(i+1) - 2*T(i) + T(i-1)) / dx^2;

            % Hyperbolic bioheat terms:
            dTdt(i)   = (k(L)*d2Tdx2 + Q_tot) / (rho(L) * c(L));
            d2Tdt2(i) = (k_star(L)*d2Tdx2) / (rho(L) * c(L));

            % Update temperature at node i (eq. 7):
            T_new(i) = T(i) + dt * ( ...
               dTdt(i) + tau_q*dTdt(i) ...
             - tau_T*d2Tdt2(i) ...
             + (k(L) + k_star(L)*tau_v)*dTdt(i) );
        end

        % (c) Convective (Robin) BC at left boundary (i = 1):
        % Forward Euler Approx dx/t
```

```
        T_new(1) = (k(1) * T(2) + h*dx*22) / (h*dx + k(1)); % 22C is RT

        % (d) Convective (Robin) BC at right boundary (i = nx):
        % Backward Euler Approx dx/dt
        T_new(nx) = (h*dx*37 - k(3) * T(end-1)) / (h*dx - k(3)); % 37C is body temp

        % (e) Advance to next timestep
        T = T_new;

        % (f) If current    t    matches any snapshot index, store T(x)
        idx_this = find(snap_idx == n);
        if ~isempty(idx_this)
            T_snapshots(:, idx_this) = T;
        end

        % (g) Store T at tumor location
        T_tumor(n) = T_new(iTumor);
    end

    %% Store profile for given P to be compared after
    T_layers(:,layer_test) = T_snapshots(:,end);
```

```

    T_tumor_layers(:,layer_test) = T_tumor;

end % end of testing parameter loop (power)

%% Plot T(x) for Layers
legend_layers = ["3-layer Model with Qv/Qd","Epidermis Homogeneity", "Dermis
    Homogeneity", "Subcutaneous Homeogeneity", "Subcutaneous Homogeneity, no Qv/Qd
    "];

figure('Position',[200,200,800,500]);
hold on;

colors = lines(numel(legend_layers));
for j = 1:size(T_layers,2)
    plot(x*1000, T_layers(:,j), 'Color', colors(j,:), 'LineWidth',1.5, ...
         'DisplayName', num2str(legend_layers(j)));
end

ylim([36 50])
xlim([0 Lx*10^3])
% xline(0, '-',{'Epiderm'},'HandleVisibility','off','Fontsize',16)
% xline((L_epi)*10^3,'-',{'Derm'},'HandleVisibility','off','Fontsize',16)
% xline((L_epi+L_derm)*10^3,'-',{'Subcutaneous'},'HandleVisibility','off','Fontsize
    ',16)
xline(x_ast*10^3,'-r',{'Tumor'},'LineWidth',5,'HandleVisibility','off','Fontsize'
    ,20)
xlabel('Distance (mm)','Fontsize',16);
ylabel('Temperature ( C )','Fontsize',16);
title(['Temperature Profile T(t=10s,x), Skin Layers'],'Fontsize',20);
legend('Location','northeast','FontSize',12);
grid on;
hold off;


%% 6. PLOT: TUMOR TEMPERATURE VS TIME
figure('Position',[200,200,800,500]);
hold on;

colors = lines(numel(legend_layers));
for j = 1:size(T_layers,2)
    plot(time, T_tumor_layers(:,j), 'Color', colors(j,:), 'LineWidth',1.5, ...
         'DisplayName', num2str(legend_layers(j)));
end

```

```
xlim([0 time(end)])
ylim([36 50])
xlabel('time (s)','Fontsize',16);
ylabel('Temperature ( C )','Fontsize',16);
title(['Temperature at Tumor Location vs. Time, Skin Layers'],'Fontsize',20);
legend('Location','best','FontSize',12);
grid on;
hold off;

T_layers_difference = T_layers(:,1)-T_layers(:,5);
T_tumor_layers_difference = T_tumor_layers(:,1)-T_tumor_layers(:,5);

figure
subplot(2,1,1)
hold on
plot(x*10^3,T_layers_difference)
xline(0, '-',{'Epiderm'},'HandleVisibility','off','Fontsize',14)
xline((L_epi)*10^3,'-',{'Derm'},'HandleVisibility','off','Fontsize',14)
xline((L_epi+L_derm)*10^3,'-',{'Subcutaneous'},'HandleVisibility','off','Fontsize'
    ,14)
xline(x_ast*10^3,'-r',{'Tumor'},'LineWidth',3,'HandleVisibility','off','Fontsize'
    ,18)
title('Difference in T(x,t=10s)','FontSize',16)
xlabel('Distance (mm)','Fontsize',14);
ylabel('Temperature ( C )','Fontsize',14);
hold off

subplot(2,1,2)
plot(time,T_tumor_layers_difference)
xlim([0 time(end)])
title('Difference in T(x=Tumor,t)','FontSize',16)
xlabel('time (s)','Fontsize',14);
ylabel('Temperature ( C )','Fontsize',14);
```

Listing 5: skin_layer_comparison.m