Analysis of Heat Conduction at Small Scales

Erdem Caliskan

March 2022

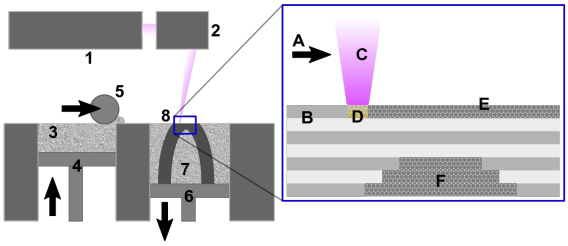
1 Introduction

The use of Fourier's law and the Maxwell-Cattaneo Equations can be applied to model the behavior of a heat change and heat gradient.

Typically, heat transfer can be modelled through Fourier's law and partial differential equations such as the heat equation. These equations tend to hold for time-scales encountered in ordinary circumstances (such as typical heat transfers over a period of seconds or minutes as opposed to (nanoseconds or milliseconds) and distributed that are not concentrated at a single point. These interactions include industrial applications of heating, turbines, or engines, etc. with some degree of uniformity.

However, the introduction of a high-intensity beam of energy in a very small point over short periods of times can result in situations in which Fourier's law can not hold sufficiently as well as other equations. One of these applications, in particular, includes SLS, or Selective Laser Sintering.

SLS involves the application of a "high-powered laser to fuse small particles of plastic, metal, cermaic, or glass powders into a mass that has a desired three dimensional shape powders into a mass that has a desired three-dimensional shape." [1]



SLS [2]

SLS has been documented from its origins and offers promising advantages in precise machining and manufacturing. Such advantages have provided promising uses for laser sintering in general and motivates the usage and analysis of this technology which has had its history described and motivation behind the procedure in [3]. SLS has been developed originally by researchers from the University of Texas, Austin.

SLS has been used in industrial and manufacturing applications and offers promising results for emerging technologies such as drug manufacturing. Furthermore, "the selective laser sintering (SLS) technique has been investigated for oral drug-delivery applications" [4]. The results have found that the technique can provide novel use cases for the "printability of pharmaceutical materials, although an important aspect for understanding the sintering process has only been properly explored in one article" [4]. In addition, "Generally, SLS shows great potential to improve compliance within fragile populations" and other medical applications" [4].

As a result, the study and analysis of SLS as a technique for manufacturing and drug delivery among possible applications is of great interest. As a result, and given the relative sparsity of new material to be explored in the applications of SLS, the analysis of SLS as a technique for heat conduction at small scales can provide very promising usage.

In general, the problem at hand has not been studied before in much depth. We seek to provide a relevant proposition to analyze the SLS laser technique in a variety of methods.

The SLS technique involves a container of powder that is heated via a laser. As the laser travels along the surface of the domain it heats the powder to super temperatures in a specific pattern over layers. Then the process continues over different layers. Since the laser itself is often several kW's at a specific point, the energy is very concentrated and may not fully be described completely by the standard Fourier Heat Equation. The laser emanates and is concentrated at the very top of the surface. As a result, there is an extremely high temperature gradient between the top of the surface of the powder where the laser is being applied and the rest of the volume, where the temperature is often assumed to be room temperature or near it.

Additionally, there is a time "lag" introduced between the body of the volume and the point of heating (laser point). As the heating is near instantaneous at the very point of heating of the powder and disperses throughout the body of the powder as time progresses, there is an additional time delay or "lag" between the heat transferring through the medium.

The effect of this laser sintering is a situation in which heat flow transfer via the Fourier Heat Law would be an edge case, and other equations governing the transfer can be compared and better model the transfer. Among these include the Maxwell-Cattaneo equation.

The real-time analysis of heat transfer is of great interest to scientists and engineers as "Heat equations can estimate the thermal distribution and phase transformation in real- time based on the operating conditions and material properties". Some of these fields are including "including laser and electron beam processing". Furthermore, "integral transform technique (ITT) is a powerful general-purpose semi-analytical/numerical method that transforms partial differential equations into a coupled system of ordinary differential equations.[5]

The significance of these methods is that they offer a more robust model and set of governing equations from which further models can be derived for heat transfer in a domain.

The previous work described involves the modelling of ultra-fast nanoparticle melting with the Maxwell-Cattaneo equation. The reason for introducing such a new equation is that the ordinary Fourier Heat equation is not tuned completely for very high-intensity beams of energy at a concentrated point over a very short time interval.

The goal of the present work at hand is to create an effective model for analysing the transfer of heat in the surface and volume. We will employ the finite element method in 1D over a slice of the domain to analyze the heating distribution at the various positions along the unit length. We will use a unit

length to serve as the element and discretize the surface into lengths for further analysis with the finite volume method.

2 Problem Description

Heat conduction problems are mostly described using Fourier's law

$$q = -kT_x \tag{1}$$

where q is the heat flux, k is the thermal conductivity, and ∇T is the temperature gradient. When constant k is used, this equation implies infinite speed of heat propagation, which is non-physical.

To overcome this non-physicality, one can introduce single-phase-lag to the heat conduction model by replacing t with $(t + \tau_q)$ in the heat flux. Thus,

$$\mathbf{q}(\mathbf{x}, t + \tau_q) = -k\boldsymbol{\nabla} T(\mathbf{x}, t) \tag{2}$$

where τ_q is the time lag, or relaxation time of the heat flux. By expanding the left-hand-side of the equation with the Taylor series, Eq. 2 can be approximated as:

$$\tau_q q_t + q = -kT_x \tag{3}$$

Equation 3 is the so called Maxwell-Cattaneo-Vernotte (MCV) constitutive equation. This equation is used when modeling wave propagation.

Due to some of it's drawbacks, involving the violation of maximum principle [6], and some physical contradictions [7], researches came up with the equation:

$$\tau_q q_t + q = -kT_x + \kappa^2 q_{xx} \tag{4}$$

which is so called Guyer-Krumhansl (GK) equation, where κ^2 term is related with the mean free path from the aspect of kinetic theory [8].

On the other hand, when source terms are neglected for the simplicity, balance of energy is expressed as

$$\rho c_p T_t + \nabla \cdot \mathrm{q} = 0 \tag{5}$$

where ρ is the material density, c_p is the specific heat, ∂_t is the partial time derivative, ∇ is the gradient, and $\nabla\cdot$ is divergence.

To model the behavior of the system, Equations 1, 5, or 3, 5, or 4, 5 can be used to obtain temperature-only partial differential equation for the heat conduction.

Using the equations 1 and 5, we get

$$T_t = \alpha T_{xx} \tag{6}$$

where $\alpha = \dfrac{k}{\rho c_p}$ is the thermal diffusivity.
Using the equations 3 and 5, and eliminating q we get

$$\rho c_p T_{tt} + q_{xt} = 0, \tag{7}$$
$$q_x + \tau_q q_{xt} = -k T_{xx} \tag{8}$$

Thus,

$$\rho c_p T_{tt} - \frac{k}{\tau_q} T_{xx} + \frac{\rho c_p}{\tau_q} T_t = 0 \tag{9}$$
$$\tau_q T_{tt} + T_t = \alpha T_{xx} \tag{10}$$

Using the equations 4 and 5, and eliminating q we get

$$\tau_q T_{tt} + T_t = \alpha T_{xx} + \kappa^2 T_{txx} \tag{11}$$

According to experiments conducted in room temperature, deviation from the Fourier law occurs in the overdamped ($\kappa^2 > \tau_q$) region, in contrast with MCV prediction where the deviations measured in the underdamped ($\kappa^2 < \tau_q$) region when using 11.

Initial conditions

For the partial differential equations, we will use 0 temperature (dimensionless) throughout the domain as an initial condition.

Boundary conditions

Both boundaries going to have Neumann (heat flux) boundary. Heat flux will be taken as a pulse function, which have the following form

Right-hand-side boundary is going to be set to adiabatic ($q = 0$) during the simulation. Thus, the boundary conditions for Equation 6

$$\begin{aligned} q_0(t) = q_{x=0,t} &= \begin{cases} 1 - cos(2\pi \dfrac{t}{t_p}) & \text{for } 0 \le t \le t_p \\ 0 & \text{for } t > t_p \end{cases} \\ T(x,0) &= 0 \text{ for } 0 < x < 1 \\ q(L,t) &= 0 \text{ for } t > 0 \\ q(x, t=0) &= 0 \end{aligned} \tag{12}$$

where t_p is the pulse duration. Input signal with respect to time up to final time is given in Figure 1.

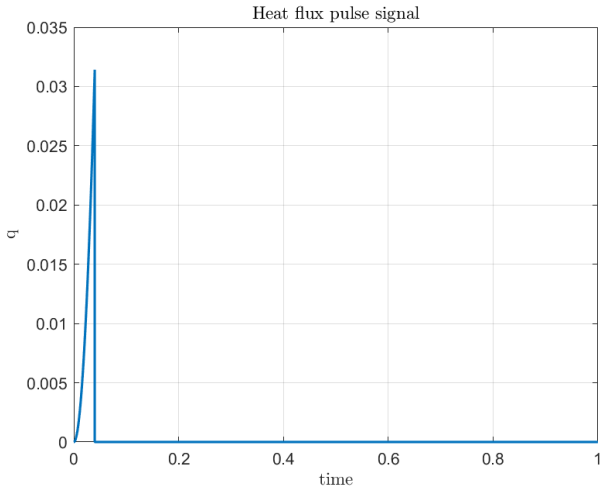


Figure 1: Pulse function for the heat flux up to t_{end}.

Simplifications and Assumptions

Domain will be modeled using a one dimensional formulation. Porosity of the material is not going to be taken into account in the calculations. Thus, material will be considered homogeneous. We will use constant specific heat, density, and thermal conductivity. Laser is applied as heat flux from the boundary instead of source term.

3 Methodology

There are multiple ways to solve the Equations 6, 10, and 11. In this project, we will utilize finite volume method to tackle with the problem.

In this work, we will use the following dimensionless parameters:

$$\tau_\Delta = \frac{\alpha t_p}{L^2}; \quad \tau_q = \frac{\alpha \tau_q}{L^2} \tag{13}$$

and all physical parameters related with the material will be assumed as 1. The details of the dimensionless parameters are given in Reference [9].

3.1 Finite Volume Scheme

For the explicit finite volume scheme the equation can be expressed as follows [10]:

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x_i \tau_\Delta}\left[q_{i+\frac{1}{2}}^{n+\beta} - q_{i-\frac{1}{2}}^{n+\beta}\right] + \Delta t S_i^n \tag{14}$$

where U_i^n is the mean value of u (temperature) over V_i, discretized volume at time t_n, $F_{i-\frac{1}{2}}^{n+\alpha}$ is the mean flux during $[t_n, t_{n+1}]$ across face $x_{i-\frac{1}{2}}$, S_i^n is the mean source over V_i and timestep, $\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$ is the volume of V_i for one dimensional problem, and $\Delta t = t_{n+1} - t_n$ is the timestep. β can take various values such as 0 for fully explicit, 1/2 for Crank-Nicolson, and 1 for implicit. We will use $\beta = 0$ throughout this work.

GK constitutive equation with no source, and taking $\beta = 0$, Equation 11 becomes

$$q_j^{n+1} = q_j^n - \frac{\Delta t}{\tau_q} q_j^n - \frac{\tau_\Delta \Delta t}{\tau_q \Delta x}\left(T_j^n - T_{j-1}^n\right) + \frac{\kappa^2 \Delta t}{\tau_q \Delta x^2}\left(q_{j+1}^n - 2q_j^n + q_{j-1}^n\right) \tag{15}$$

Equation 15 is able to reproduce the solutions of the Fourier equation with ($\kappa^2 = \tau_q$) and MCV equation with ($\kappa = 0$).

Since this is an explicit scheme, one needs to calculate the stability condition. Kovacs and Van [9] applied von Neumann method, using assumed solution of the differential equation with the form $T_j^n = \psi^n \cdot e^{ikj\Delta x}$, where ψ is the growth factor with stability condition, $|\psi| \leq 1$, i is the imaginary unit, and k is the wave number. After substitution, the system of linear algebraic equations becomes

$$\mathbf{M} \cdot \begin{pmatrix} T \\ q \\ Q \end{pmatrix} = \mathbf{0}$$

where

$$\mathbf{M} = \begin{pmatrix} 1-\psi & -\frac{\Delta t}{\tau_\Delta \Delta x}\left(e^{ik\Delta x} - 1\right) & 0 \\ -\frac{\Delta t \tau_\Delta}{\tau_q \Delta x}\left(1 - e^{-ik\Delta x}\right) & 1 - \frac{\Delta t}{\tau_q} - \psi & -\frac{\kappa \Delta t}{\tau_q \Delta x}\left(1 - e^{-ik\Delta x}\right) \\ 0 & \frac{\kappa \Delta t}{\tau_Q \Delta x}\left(e^{ik\Delta x} - 1\right) & 1 - \frac{\Delta t}{\tau_Q} - \psi \end{pmatrix}$$

is the characteristic equation of the system can be calculated from $\det\mathbf{M} = 0$ and Jury criteria can be applied for stability analysis. For simplicity, we used a step sizes of 1×10^{-5} and 1×10^{-7}, for our element size 0.003 (300 elements).

4 Results

For the described problem, the analytical solution only exists in infinite summation form [10]. It has been shown that the results of the numerical scheme

described in Section 3.1 is matching with the the analytical solution in infinite summation form. We have qualitatively verified our results with the ones given in [10]. Qualitative comparison was quite similar in terms of peaks and numerical values, thus we proceed with the solution. However, direct comparison with the infinite sum analytical solution should be conducted later.

The parameters that are applied are as follows.

1. Fourier: $\kappa^2 = \tau_q$
2. MCV: $\tau_q = 0.02, \kappa = 0$
3. Over-diffusive GK: $\tau_q = 0.02, \kappa^2 = 10 \cdot \tau_q$

For all cases, heat pulse duration is $\tau_{Delta} = 0.04$, time step is 1×10^{-5} (except over-diffusive where , time step is 1×10^{-7}), and number of elements is 300. Following figures are compared qualitatively with the Reference [10].

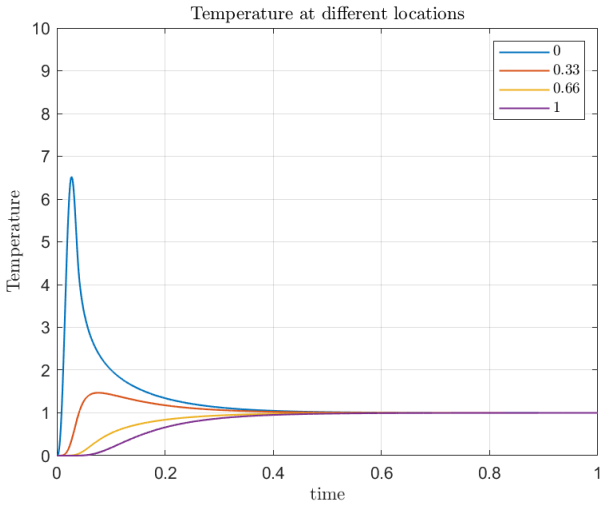


Figure 2: Fourier-like solution of the GK equation. $\kappa^2 = \tau_q$, $\kappa = 0.001$.

In Figure 2, temperature values with respect to position is given for four different location in the domain. It should be noted that the Fourier solution given in Figure 2 is an approximation to the Fourier's solution, due to the double time derivative in the equation. Thus it is only valid for near-zero values of κ.

For the solution of the MCV heat equation, one might take the coefficient of the double time derivative κ as zero and the equation turns into MCV equation. For the time-lag term, we used $\tau_q = 0.02$ in Figure 3.

In the same fashion, we applied $\tau_q = 0.02, \kappa^2 = 10\tau_q$ to get the over-diffusive GK solution. Results are given in Figure 4.

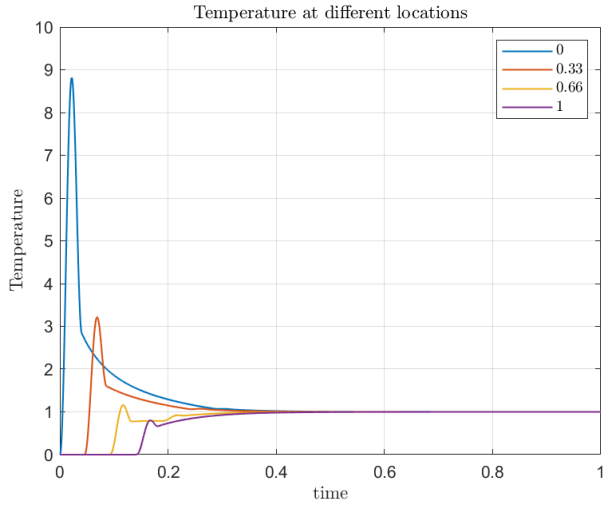


Figure 3: MCV solution of the GK equation. $\tau_q = 0.02, \kappa = 0$.

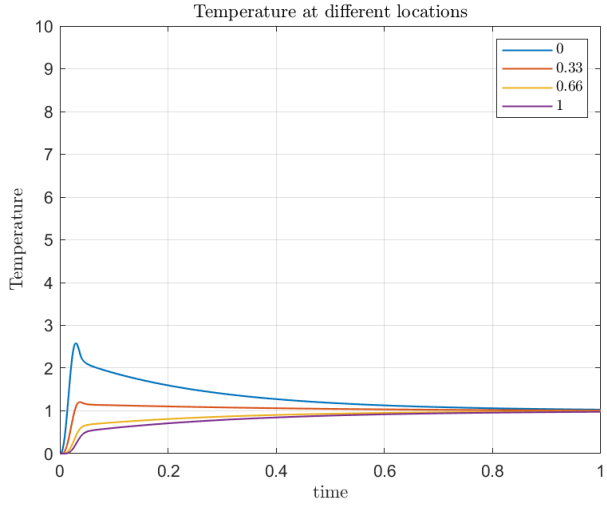
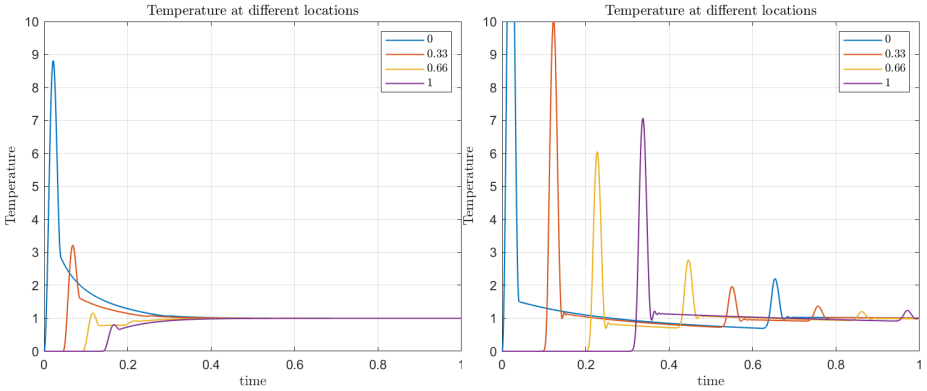


Figure 4: Over-diffusive GK solution. $\tau_q = 0.02, \kappa^2 = 10 \cdot \tau_q$.

We can compare the affects of the parameters tau_{delta}, tau_q, and *kappa* to the heat distribution of the domain.

Different time-lag values, τ_q

To see the effect of the time-lag, we can use the MCV solution as the base case, and increase the tau_q 5 times of the original value. Results are given in Figure 5.

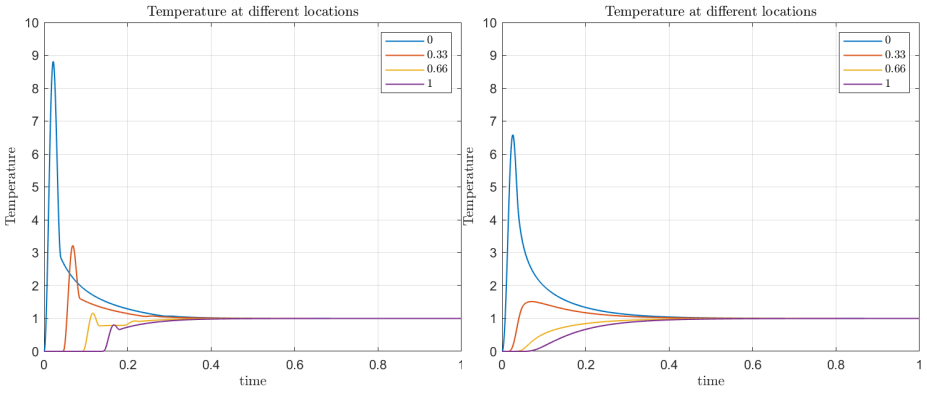


(a) MCV solution with $\tau_q = 0.02, \kappa = 0$ (b) MCV solution with $\tau_q = 0.1, \kappa = 0$

Figure 5: Difference between low and high τ_q

From the Figure 5, it can be seen that the speed of the travelling wave, and diffusivity is reduced. Thus the shape of the discontinuity is better preserved. We can see clearly see the reflected peaks from our temperature probes on 4 different locations in the domain.

When we have low values for the time-lag, the solution is quite similar to the Fourier's solution, as can be seen in the Figure 6

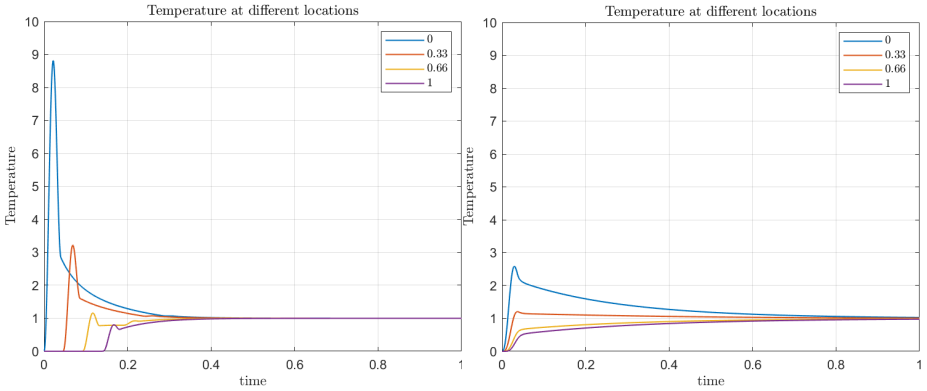


(a) MCV solution with $\tau_q = 0.02, \kappa = 0$ (b) MCV solution with $\tau_q = 0.002, \kappa = 0$

Figure 6: Difference between low and lower (Fourier-like) τ_q

Different κ values

To see the effect of the κ, we can use the over-diffusive GK solution as the base case, and compare it with the MCV solution. Comparison is given in the Figure 7.



(a) Over-diffusive GK: $\tau_q = 0.02, \kappa^2 = 10\tau_q$ (b) MCV solution with $\tau_q = 0.02, \kappa = 0$

Figure 7: Difference between low and high κ values

As we can clearly see from the the Figure 7, thermal inertia is increasing with increase in κ. Response is similar to a system with high damping when we have the high κ value. We should note that the time step we used for the other cases was not enough for over-diffusive GK, thus we decreased it 100 times to 1×10^{-7},

5 Conclusions

In this work, we formulated two alternatives to the Fourier's heat law. The two alternative constitutive equations are more appropriate for the modeling of the heat conduction process that is happening in the SLS manufacturing. For all the equations, we used dimensionless parameters to investigate the effect of the parameters, which can easily computed from physical parameters.

First order finite volume scheme is utilized for numerical solution of the heat conduction problem. This method is efficient in terms of computation time and accuracy. We used explicit time stepping for the solution. Maximum computation was around 35 seconds, for 300 elements, when we used time step equal to 1×10^{-7}, which corresponds to 1×10^{7} time steps. Even though, one targets higher accuracy with using more elements, since lower maximum time step for stability, the code should make use of parallelism in order to have reasonable computation times.

Contributions: Jan Jakowski: Introduction, Erdem Caliskan: Problem Description, Methodology, Results, Conclusion, Coding, Editing

References

[1] Wikipedia. *Selective laser sintering*. URL: https://en.wikipedia.org/wiki/Selective_laser_sintering.

[2] Wikipedia. *Selective laser sintering process*. URL: https://en.wikipedia.org/wiki/Selective_laser_sintering#/media/File:SLS_schematic.svg.

[3] Katherine Tucker et al. "Network Based Technology Roadmapping for Future Markets: Case of 3D Printing". In: *Technol. Invest.* 5.3 (Aug. 2014), pp. 137–156. ISSN: 2150-4059. DOI: 10.4236/TI.2014.53014.

[4] Yanis A. Gueche et al. "Selective Laser Sintering (SLS), a New Chapter in the Production of Solid Oral Forms (SOFs) by 3D Printing". In: *Pharmaceutics* 13.8 (Aug. 2021). ISSN: 19994923. DOI: 10.3390/PHARMACEUTICS13081212.

[5] Mihai Oane, Muhammad Arif Mahmood, and Andrei C. Popescu. "A State-of-the-Art Review on Integral Transform Technique in Laser-Material Interaction: Fourier and Non-Fourier Heat Equations". In: *Mater. (Basel, Switzerland)* 14.16 (Aug. 2021). ISSN: 1996-1944. DOI: 10.3390/MA14164733.

[6] Soma Both et al. "Deviation from the Fourier law in room-temperature heat pulse experiments". In: *J. Non-Equilibrium Thermodyn.* 41.1 (Jan. 2016), pp. 41–48. ISSN: 14374358. DOI: 10.1515/JNET-2015-0035/MACHINEREADABLECITATION/RIS. arXiv: 1506.05764. URL: https://www.degruyter.com/document/doi/10.1515/jnet-2015-0035/html.

[7] C. Bai and A. S. Lavine. "On Hyperbolic Heat Conduction and the Second Law of Thermodynamics". In: *J. Heat Transfer* 117.2 (May 1995), pp. 256–263. ISSN: 0022-1481. DOI: 10.1115/1.2822514. URL: https://asmedigitalcollection.asme.org/heattransfer/article/117/2/256/415778/On-Hyperbolic-Heat-Conduction-and-the-Second-Law.

[8] Ingo Müller and Tommaso Ruggeri. *Rational extended thermodynamics*. Vol. 37. Springer Science & Business Media, 2013.

[9] R. Kovács and P. Ván. "Generalized heat conduction in heat pulse experiments". In: *Int. J. Heat Mass Transf.* 83 (Apr. 2015), pp. 613–620. ISSN: 0017-9310. DOI: 10.1016/J.IJHEATMASSTRANSFER.2014.12.045.

[10] R. Kovács. "Analytic solution of Guyer-Krumhansl equation for laser flash experiments". In: *Int. J. Heat Mass Transf.* 127 (Dec. 2018), pp. 631–636. ISSN: 0017-9310. DOI: 10.1016/J.IJHEATMASSTRANSFER.2018.06.082. arXiv: 1804.05225.

6 Appendix

Listing 1: Matlab code for the Fourier heat law.

```
% Erdem Caliskan
% 05/10/2022
% nonFourier.m input file
% Non-equilibrium thermodynamics
%
close all
clear all


MM = 300; t0=0;  tend=1;  dtout=0.0025;  D = 1;  a=0; b=1;
dt = 1e-5;
dx = (b-a)/MM;
dtout=1;
tau_d = 0.04;
%% fourier
kappa = sqrt(0.001); tau_q = kappa^2;
GKSolver(tau_d, b, a, MM, dx, tend, t0, dt, dtout, tau_q, kappa);

%% MCV

kappa = 0; tau_q = 0.02;
GKSolver(tau_d, b, a, MM, dx, tend, t0, dt, dtout, tau_q, kappa);

%% high tau_q

kappa = 0; tau_q = 0.1;
GKSolver(tau_d, b, a, MM, dx, tend, t0, dt, dtout, tau_q, kappa);
%% low tau_q

kappa = 0; tau_q = 0.002;
GKSolver(tau_d, b, a, MM, dx, tend, t0, dt, dtout, tau_q, kappa);

%% Over-diffusive
% MM = 32;
tic
dt = 1e-7;
tau_q = 0.02; kappa = sqrt(10*tau_q);
GKSolver(tau_d, b, a, MM, dx, tend, t0, dt, dtout, tau_q, kappa);
toc
%% Under-diffusive
% MM = 32;
```

```
dt = 1e-7;
tau_q = 0.02; kappa = sqrt(0.02/5);
GKSolver(tau_d, b, a, MM, dx, tend, t0, dt, dtout, tau_q, kappa);

%%
function GKSolver(tau_d, b, a, MM, dx, tend, t0, dt, dtout, tau_q, kappa)
tpulse = tau_d;

M  = (b-a)*MM;

x = zeros(M+2,1);
x = MESH(M,a,b,dx,x);

% set the timestep
Nend   = round((tend-t0)/dt); % #of timesteps

[U,F,Q]    = INIT(M);

nsteps  = 1;
time    = 0.0;
tout    = max(dtout,dt);
% OUTPUT(x,U,time,nsteps)

U1    = zeros(Nend,1);
U3    = zeros(Nend,1);
U6    = zeros(Nend,1);
Uend  = zeros(Nend,1);

figure
xlabel('Length', 'interpreter', 'latex')
ylabel('Temperature', 'interpreter', 'latex')
titlename= sprintf('Temperature distribution across the domain');
title(titlename, 'interpreter', 'latex')
box on
grid on
ylim([0  10])
ax = gca;
ax.NextPlot = 'replaceChildren';
loops = tend/tout;
k = 1;
filename = sprintf(['tau_d_%.2f_tau_q_%.2f_kappa_%.2f_' ...
    'tend_%.2f_dtout_%.3f.avi'],tau_d,tau_q,kappa,tend,dtout);
v = VideoWriter(filename,'Motion JPEG AVI');
open(v)
for nsteps = 1:Nend
    time = nsteps*dt;
```

```
    [F,Q] =  FLUX(U,F,Q,M,tau_d,tau_q,time,dt,dx,kappa);
    U =  PDE(U,F,M,dt,dx,tau_d);
    U1(nsteps,1) = U(1);
    U3(nsteps,1) = U(floor(0.33*M));
    U6(nsteps,1) = U(round(0.66*M));
    Uend(nsteps,1) = U(end);
    if time >= tout
        %          % OUTPUT(x,U,time,nsteps)
        plot(x,U,'Color','red','LineWidth',1.5)
        ftime = sprintf('time = %.3f',time);
        text(0.8,8.5,ftime, 'interpreter', 'latex')
        frame = getframe(gcf);
        writeVideo(v,frame)
        tout = tout + dtout;
        k = k+1;
    end
end % end of time-stepping
close(v)
%%
tv = 0:1/(Nend-1):1;
figure
xlabel('time', 'interpreter', 'latex')
ylabel('Temperature', 'interpreter', 'latex')
titlename= sprintf('Temperature at different locations');
title(titlename, 'interpreter', 'latex')
box on
grid on
ylim([0 10])
hold on
plot(tv,  U1,'LineWidth',1)
plot(tv,  U3,'LineWidth',1)
plot(tv,  U6,'LineWidth',1)
plot(tv,Uend,'LineWidth',1)
legend('0','0.33','0.66','1', 'interpreter', 'latex')
filename = sprintf(['png_tau_d_%.2f_tau_q_%.2f_kappa_%.2f_' ...
    'tend_%.2f_dtout_%.3f.png'],tau_d,tau_q,kappa,tend,dtout);
grid on
exportgraphics(gcf,filename)
end
%% FLUX for MCV
function [F,Q]= FLUX(U,F,Q,M,tau_d,tau_q,time,dt,dx,kappa)
if time < tau_d
F(1)     = (1-cos(2*pi*time/tau_d)); % Neumann BC for b
else
F(1)     = 0;                        % Neumann BC for b
end
```

```
% Internal face values
for i = 2:(M)
    F(i)= 1*(F(i)-dt/tau_q*F(i)-(tau_d*dt)/(tau_q*dx)*(U(i+1) - ...
        U(i))+kappa^2*dt/tau_q/dx^2*(F(i+1)-2*F(i)+F(i-1)));
end
F(M+1)  = 0;
end

%% PDE function for MCV
%
function U = PDE(U,F,M,dt,dx,tau_d)
U(1)     = U(2)+dx/2*F(1);                  % Neumann BC for b
for i = 2:(M+1)
    U(i) = U(i)-dt/dx/tau_d*(F(i)-F(i-1)); % Internal U values
end
U(M+2)  = U(M+1)+dx/2*F(M+1);               % Neumann BC for b
end
%%
function [U,F,Q] = INIT(M)
U = zeros(M+2,1);
F = zeros(M+1,1);
Q = zeros(M+1,1);
U(1) = 0;
U(M+2)  = 0;
end

function x = MESH(M,a,b,dx,x)
x(1) = a;
x(2) = x(1) + dx/2;
for i = 3:(M+1)
    x(i) = x(2) + (i-2)*dx;
end
x(M+2) = b;
end

function OUTPUT(x,U,time,nsteps)
filename = [num2str(time),'.o.prof'];
fileID = fopen(filename,'w');   % output file
fprintf(fileID, '# Profile at time: %9.4f, nsteps=%6d \n', time,nsteps);
for i = 1:(M+2)
    fprintf(fileID, '%12.4f %16.8g\n', x(i), U(i));
end
fclose(fileID); % end of the output
end
```