Modelling, Simulation, and Visualization of Heat Equation Dynamics

BABATUNDE Oluleye H
Osun State University, Osogbo, Nigeria

Abstract

Aims/ objectives: To show modelling, simulation and visualization of the dynamics of heat equation in a rod. Other PDE models and Numerical approaches are also discussed.

Study design: Cross-sectional study. This paper is simply about how to use MATLAB software to solve PDE model.

Place and Duration of Study: Australia

Methodology: A Class of PDE was uniquely modelled, simulated, and visualized using certain algorithm and MATLAB routine for Elliptic/Hyperbolic PDE. Different diagrams showing the nature of the solution are discussed.

Results: The nature of the PDE governing heat dynamic in a rod are shown in the figures. Another PDE model is also presented.

Conclusion: A simple demonstration of numerical simulation of PDE governing the popular heat equation is presented. Three other PDE models are considered. The results herein can be adapted and applied to other more complex PDE models.

Keywords: Heat Equation; Modelling; Numerical Simulation; Visualization

2010 Mathematics Subject Classification: 53C25; 83C05; 57N16

1 Introduction

Partial differential equations (PDEs) are unavoidable these days since they are applicable in many real life scenarios. Examples of these scenarios are mathematical modeling of many physical, chemical and biological phenomena and many diverse subject areas such as fluid dynamics, electromagnetism, material science, astrophysics, economy, financial modeling. Typically PDEs have been applied in fluid mechanics, general relativity, quantum mechanics, biology, tumor modeling and option pricing (American, Asian, European, etc) [McDonough (2008); Hans (2009)]. PDEs model spatio-temporal interactions and variations in the considered physical processes. Despite the general applicabilities

*BABATUNDE Oluleye H: E-mail: hezecomp@yahoo.com
of PDEs, many of them do not have analytical or closed form solutions (e.g. by Laplace and Fourier transform methods, or in the form of a power series). It is most times, either impossible or impracticable to solve majority of these PDEs analytically, and thus there is need to often resort to numerical methods and approximations to the unknown analytical solution. This paper illustrates a simple numerical solution of PDE using MATLAB.

2 Related Works and Classification of Partial Differential Equations

There are many ways in which PDEs can be classified [McDonough (2008)]. Suppose one has a function $u$ that describes the temperature at a given location $(x, y, z)$. This function will change over time as heat spreads throughout space. The heat equation is used to model the change in the function $u$ over time. The image in Figure 5 describes the way heat changes in time along a metal bar. One of the interesting properties of the heat equation is the maximum principle that says that the maximum value of $u$ is either earlier in time than the region of concern or on the edge of the region of concern. This is essentially saying that temperature comes either from some source or from earlier in time because heat permeates but is not created from nothingness. This is a property of parabolic partial differential equations and is not difficult to prove mathematically. Another interesting property is that even if $u$ has a discontinuity at an initial time $t = t_0$, the temperature becomes smooth as soon as $t > t_0$. For example, if the given bar of metal shown in Figure 5 has temperature 0 and another has temperature 100 and they are stuck together end to end, then very quickly the temperature at the point of connection will become 50 and the graph of the temperature will run smoothly from 0 to 100. The heat equation is used in probability and describes random walks. It is also applied in financial mathematics for this reason. Heat Equation PDE is applicable in Riemannian geometry and topology and it was adapted by Richard Hamilton when he defined the Ricci flow that was later used by Grigori Perelman to solve the topological Poincaré conjecture.

3 Equation type

The study of PDEs operators fits nicely into the framework of distribution theory [Casten (2013)]. These operators are being used to describe heat equation, Laplace equation, wave equation, free non-relativistic Schrodinger equation etc. Example of the operator is the Cauchy-Riemann operator. The heat, Laplace, wave and free non-relativistic Schrodinger equation can be represented by the operator of the form $P(D)$, where $P$ is a complex polynomial in $n$ variables $x_1, x_2, x_3, \ldots, x_n$ and

$$D = \left( \frac{1}{i} \frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \ldots, \frac{\partial}{\partial x_n} \right) \quad (3.1)$$

or

$$P(D)u = \sum_{|\alpha| \leq m} c_\alpha D^\alpha u \quad (3.2)$$

Applying this operator in cartesian, cylindrical, and spherical coordinates respectively, we have the following expressions:

1. Cartesian: $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}$

2. Cylindrical: $\frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial u}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} + \frac{\partial^2 u}{\partial z^2}$.

3. Spherical coordinates: $\frac{1}{r^2} \frac{\partial}{\partial r} \left( r^2 \frac{\partial u}{\partial r} \right) + \frac{1}{r^2 (\sin \theta)} \frac{\partial}{\partial \theta} \left( \sin \theta \frac{\partial u}{\partial \theta} \right) + \frac{1}{r^2 (\sin^2 \theta)} \frac{\partial^2 u}{\partial \phi^2}$

101
The most general form of linear second-order partial differential equations, when restricted to two independent variables and constant coefficients, is given in Equation 3.3:

\[ A(x, y)u_{xx} + B(x, y)u_{xy} + C(x, y)u_{yy} + Du_x + Eu_y + Fu = G(u, u_x, u_y, x, y) \]  

(3.3)

where \( G \) is a known forcing function; \( A, B, C, D, E \) are given constants, and subscripts denote partial differentiation. In the homogeneous case, i.e. \( G = 0 \); this form is reminiscent of the general quadratic form from high school analytic geometry:

\[ Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \]  

(3.4)

A linear second-order PDE with two independent variables on a domain in the form (3.3) is said to be:

- **parabolic**: if \( \forall x, y \in \Omega, B^2 - 4AC = 0 \)
- **hyperbolic**: if \( \forall x, y \in \Omega, B^2 - 4AC > 0 \)
- **elliptic**: if \( \forall x, y \in \Omega, B^2 - 4AC < 0 \)

As an example, the following PDE is hyperbolic:

\[ (1 + x^2) \frac{\partial^2 U}{\partial x^2} + (5 + 2x^2) \frac{\partial^2 U}{\partial x \partial t} + (4 + x^2) \frac{\partial^2 U}{\partial t^2} = 0 \]  

(3.5)

since \( B^2 - 4AC > 0 \), i.e. \( (5+2x^2)^2 - 4(1+x^2)(4+x^2) = 25 - 6 = 9 > 0 \). This same classification elliptic, parabolic, or hyperbolic is employed for the PDE (3.3), independent of the nature of \( G(x, y) \). In fact, it is clear that the classification of linear PDEs depends only on the coefficients of the highest-order derivatives [AMS (2002)]. The grouping of terms represented as \( Au_{xx} + Bu_{xy} + Cu_{yy} \), is called the principal part of the differential operator in Equation (3.3), i.e., the collection of highest-order derivative terms with respect to each independent variable; and this notion can be extended in a natural way to more complicated operators. Thus, the type of a linear equation is completely determined by its principal part.

It should be noted that if the coefficients of (3.3) are permitted to vary with \( x \) and \( y \), its type may change from point to point within the solution domain. This may result in significant difficulties from both analytical and numerical point of view. Secondarily, it is important to note that, corresponding to each of the three types of equations, there is a unique canonical form to which (3.3) can always be reduced. The details of the transformations needed to achieve these reductions are not presented in this paper, as they can be found in literatures and in many standard texts on PDEs [Berg & McGregor (1966)]. On the other hand, it is important to be consider or take note of the possibility of simplifying (3.3), since this may also simplify the numerical analysis required to construct a solution algorithm.

**Elliptic**: Whenever \( B^2 - 4AC < 0 \), the elliptic case of (3.3) can be written as

\[ u_{xx} + u_{yy} + Au = G(x, y) \]  

(3.6)

and with \( A = 0, -1, +1 \). Whenever \( A = 0 \) and \( G = 0 \), Poisson’s or Laplace’s equation is obtained; otherwise, the result is usually termed the Helmholtz equation.

**Parabolic**: For this case \( B^2 - 4AC = 0 \) and we have

\[ u_x - u_{yy} = G(x, y) \]  

(3.7)

which is the heat equation, or the diffusion equation to be simulated in this work.

**Hyperbolic**: For this case \( B^2 - 4AC > 0 \) and we have

\[ u_{xx} - u_{yy} + Bu = G(x, y) \]  

(3.8)

where \( B = 0 \) or 1. If \( B = 0 \), we have the wave equation, and when \( B = 1 \), the linear Klein Gordon equation is obtained [Loveque (2007)].
Finally, we note that determination of equation type in dimensions greater than two requires a different approach. The details are rather technical but basically involve the fact that elliptic and hyperbolic operators have definitions that are independent of dimension, and usual parabolic operators can then be identified as a combination of an elliptic "spatial" operator and a first-order evolution operator. A problem consisting of a partial differential equation and boundary and/or initial conditions is said to be well posed in the sense of Hadamard if it satisfies the following conditions:

1. a solution exists;
2. the solution is unique;
3. the solution depends continuously on given data.

The well-posedness property is fundamentally crucial in solving problems by numerical methods because essentially all numerical algorithms are based on assumptions that their associated problems are well posed. This means that a numerical method is not likely to work correctly on an ill-posed (i.e., a not well-posed) problem. The result may be failure to obtain a solution; but a more serious outcome may be generation of numbers that do not comply with reality. It behooves the user of numerical methods to sufficiently understand the mathematics of any considered problem to be aware of the possible difficulties and symptoms of these difficulties associated with problems that are not well posed. Next, a particular problem that is not well posed is discussed. This is the so-called "backward" heat equation problem which is applicable to geophysical studies in which it is desired to predict the temperature distribution within the Earth at some earlier geological time by integrating backward from the (presumed-known) temperature distribution of the present. To demonstrate the difficulties that arise, a simple one-dimensional initial-value problem for the heat equation is considered:

\[ u_t = \kappa u_{xx}, \quad x \in (-\infty, \infty), \quad t \in [-T, 0] \]

with

\[ u(x, 0) = f(x) \]

The exact solution of this model is given as

\[ u(x, t) = \frac{1}{\sqrt{4\pi \kappa t}} \int_{-\infty}^{\infty} f(\psi) e^{-\frac{(x-\psi)^2}{4\kappa t}} d\psi \]

Obviously, it is seen here that if \( t < 0 \), \( u(x, t) \) (if it exists at all), is imaginary (since \( \kappa \), the thermal diffusivity is always greater than zero). In fact, unless \( f \) decays to zero faster than exponentially at \( \pm \infty \), there is no solution because the integral in (3.10) does not exist. It turns out that behavior of heat equation solutions places restrictions on the form of difference approximations that can be used to numerically solve the equation. In particular, schemes that are multilevel in time with a backward (in time) contribution can fail. An example of this is the well-known second-order centered (in time) method due to Richardson; it is unconditionally unstable.

### 3.1 Some Examples of PDE

Table 1 below shows examples of some common PDE, type, nature and their applications.

<p>| Table 1: Examples and Application of PDE |</p>
<table>
<thead>
<tr>
<th>S/N</th>
<th>PDE Name</th>
<th>PDE Equation</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$y \partial_x u + x \partial_y u = 0$</td>
<td>Variable coefficient transport equation</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$u_t + u_x$</td>
<td>Transport equation, first order, linear, homogeneous</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>$i \partial_t u + \partial_x^2 u$</td>
<td>Schrödinger’s equation, 2nd-order, linear, homogeneous</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>$u_t + uu_x = 0$</td>
<td>Burger’s equation, first order, non-linear, homogeneous</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>$-\partial^2_t u + \partial^2_x u$</td>
<td>Wave equation, first-order, non-linear, homogeneous</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>$-\partial_t u + \partial^2_x u$</td>
<td>Heat equation, 2nd-order, linear homogenous</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>$\partial^2_t u + \partial^2_x u + \partial^2_y u = h(x, y, z)$</td>
<td>Poisson’s equation with source function $h$, 2nd order, linear, inhomogenous</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>$\partial^2_x u + \partial^2_y u + \partial^2_z u = 0$</td>
<td>Laplace’s equation, 2nd order, linear, homogenous</td>
<td></td>
</tr>
</tbody>
</table>

### 4 Numerical Solutions of PDEs

The main idea behind numerical method is to solve a PDE on a discrete set of points representing the solution domain. This is also called discretization. Normally, the solution domain is divided into subdomains having discretization points as vertices. The distance between two adjacent vertices is the mesh size. The time interval, say $[a,b]$ is also subdivided into discrete intervals commonly called timesteps. The given PDE is then discretized accordingly to obtain a system of algebraic equations whose unknowns are the solution candidates at the discretization points. As seen in Bargallo (2006), some lists of common numerical methods commonly employed are:

1. **Finite Differences:** This is the most popular discretization technique due to its simplicity. Herein, the finite difference (FD) approximations of the associated derivatives are obtained using truncated taylor series. The FD could be **forward**, **backward** or **central**.

2. **Finite Elements:** These approaches (our concern in this paper) involve subdividing the initial solution domain into finite elements (sometimes called triangulation in the case of triangular subdomains). A set of interpolation functions are then selected. These functions are used to formulate system of equations which is then solved by numerical linear algebra methods.

3. **Moment’s Methods:** These involve use of weighting functions.

4. **Monte Carlo:** These methods are probabilistic in nature since they involve application of random numbers for deriving solution candidates to the given problem. A typical example is simulation of neutron’s motions into a reactor wall.

#### 4.1 A Typical Numerical Discretization for a PDE

1. Let’s consider a PDE given as:

$$\frac{\partial U}{\partial t} = \kappa \frac{\partial^2 U}{\partial x^2}$$

To be able to solve this by Finite Difference method, the associated partial derivatives are re-written as:

$$\frac{\partial U}{\partial t} |_{(i,j)} = \frac{U_{i+1,j} - U_{i,j}}{\Delta t}$$

and

$$\frac{\partial^2 U}{\partial x^2} = \frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{\Delta x^2}$$
Therefore the PDE is given completely as:

\[
\frac{U_{i,j+1} - U_{i,j}}{\Delta t} = \frac{\kappa c \rho U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{\Delta x^2}
\]  

(4.4)

Equation (4.4) can then be solved by numerical linear algebra method such as Gauss-Seidel iterative scheme.

2. Another example is the Wave Equation expressed as:

\[
\frac{\partial^2 U}{\partial x^2} = \frac{\partial^2 U}{\partial t^2}, t > 0, 0 \leq x \leq 1
\]  

(4.5)

subject to the following Initial (ICs) and Boundary conditions (BCs):

\begin{align*}
[I.Cs] &\rightarrow U = f(x), 0 \leq x \leq 1 \\
[B.Cs] &\rightarrow U(0,t) = h_1(t), U(1,t) = h_2(t) \text{ for all } t > 0.
\end{align*}

With Central Difference (CD) scheme, the PDE is re-written as:

\[
\frac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{m^2} = \frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{n^2}
\]  

(4.6)

where \(m\) and \(n\) are grid-space along \(x\) and \(t\) axis respectively.

5 Derivation of the Heat Equation (Modelling)

In order to motivate the study of the heat equation (3.9), a derivation of this equation from physical principles is provided. Consider a metal rod of length \(L\) and cross-sectional area \(A\) that is aligned parallel to the \(x\)-axis. Assuming that the temperature gradient in the \(y\) and \(z\) directions is negligible, the temperature profile in the rod will be given by \(u(x,0)\) for \(0 \leq x \leq L\). Then starting with an initial temperature profile \(g(x) = u(x,0)\), we heat the rod in accordance with a heat source function \(h(x)\)

\[\text{Gustafson (1980); Patankar (1980).}\]

We then pose the following question

(1) What is the temperature profile \(u(x,0)\) for \(0 \leq x \leq L\) and \(t \geq 0\)? The physical quantities we are interested in are given in the following table

<table>
<thead>
<tr>
<th>S/N</th>
<th>Quantity</th>
<th>Physical Meaning</th>
<th>Physical Meaning</th>
<th>Physical Meaning</th>
<th>Dimensions</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(u(x,t))</td>
<td>Temperature</td>
<td>Temperature</td>
<td>(K)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>(h(x))</td>
<td>Heat source</td>
<td>Energy</td>
<td>(Js^{-1}m^{-3})</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>(\rho(x))</td>
<td>Mass density</td>
<td>Mass</td>
<td>(gm^{-3})</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>(c)</td>
<td>Specific heat</td>
<td>Mass</td>
<td>(Js^{-1}K^{-1})</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>(J(x))</td>
<td>Energy flux</td>
<td>Energy</td>
<td>(Js^{-1}s^{-1})</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>(q(x,t))</td>
<td>Energy density</td>
<td>Energy</td>
<td>(Js^{-1}s^{-1})</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>(\kappa)</td>
<td>Thermal conductivity</td>
<td>Energy</td>
<td>(Js^{-1}m^{-1}K^{-1})</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Now, in order to derive a physical relationship between these variables, we must rely on physical principles. A volume \(\Omega = \{x \in [a,b]\} \) along the rod (where \(a\) and \(b\) are constants) is taken into consideration. From the law of conservation of energy we have
where $A_1 = (\text{Energy Flux through } \partial \Omega)$ and

$A_2 = \text{TotalHeatEnergy added per unit time to } \Omega$. Using the physical quantities to express this conservation principle, we write (5.1) as

$$\frac{d}{dt} \left( \int_a^b q(x,t) A(x) dx \right) = J(a,t) - J(b,t) + h(x) A(x)$$

(5.2)

Then by the fundamental theorem of calculus, we have

$$- \int_a^b \frac{\partial J}{\partial x} = J(a,t) - J(b,t)$$

(5.3)

Upon substituting (5.2) into (5.1) and bringing the derivative on the left-hand side inside the integral, we obtain

$$\int_a^b \left( \frac{\partial q}{\partial t} + \frac{\partial J}{\partial x} - h(x) \right) dx = 0$$

(5.4)

Next the energy flux is related in terms of the temperature gradient. We use Fourier’s law of heat conduction, which states that heat flows from a warm body to a cold body at a rate proportional to the temperature gradient between the two bodies [Mitchell (1969); Patankar (1980)]. Mathematically, we write

$$J(x,t) = -\kappa \frac{\partial u}{\partial x}(x,t)$$

(5.5)

Further, the energy density $q$ can be written in terms of other physical quantities as

$$q(x,t) = c\rho u(x,t)$$

(5.6)

On substituting (5.5) and (5.6) into (5.4) we obtain

$$\frac{\partial u}{\partial t} - \frac{\kappa u}{\rho c} \frac{\partial^2 u}{\partial x^2} = \frac{h(x)}{c\rho}$$

(5.7)

We now define the thermal diffusivity $D$ and temperature source $f$ by

$$D(x) = \frac{\kappa}{c\rho u(x)} = f(x)$$

(5.8)

and hence obtain the final form of the heat equation

$$\frac{\partial u}{\partial t} - D(x) \frac{\partial^2 u}{\partial x^2} = f(x)$$

(5.9)
Note that if the mass density of the rod is constant then it follows from (5.8) that \( D(x) \) is constant. Further, in the case of \( f(x) = 0 \) (i.e. no external heating) and \( D(x) = 1 \), this problem simply reduces to the homogeneous heat equation [McDonough (2008); Hans (2009)]. The domain of the heating problem is given by all points satisfying \( 0 \leq x \leq L \) and \( t \geq 0 \). Boundary conditions must be imposed at \( x = 0 \) and \( x = L \) and initial conditions imposed at \( t = 0 \). Using the heat equation (5.9), we can now formulate the heating problem as an initial value boundary value problem (IVBVP), as follows. For simplicity we assume the rod is kept at a constant temperature at either end (constant boundary conditions) and has a constant temperature initially. The IVBVP then reads

\[
\begin{align*}
\Omega &= (0, L) \times (0, \infty) & \text{: Domain} \\
u(L, t) &= u_L & \text{: Boundary Condition} \\
u(x, 0) &= C & \text{: Initial Condition} \\
u_t - Du_{xx} = f(x) & \text{: PDE}
\end{align*}
\]

(5.10)

6 Numerical Simulation

The main idea behind numerical method is to solve a PDE on a discrete set of points representing the solution domain. This is also called discretization. Normally, the solution domain is divided into subdomains having discretization points as vertices. The distance between two adjacent vertices is the mesh size. The time interval, say \([a, b]\) is also subdivided into discrete intervals commonly called timesteps. The given PDE is then discretized accordingly to obtain a system of algebraic equations whose unknowns are the solution candidates at the discretization points.

Finite

1. A specific example of this problem (i.e 5.10) is given as

\[
\pi^2 \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = 0
\]

(6.1)

which holds on an interval \( 0 \leq x \leq L \) and \( t \geq 0 \) with initial and boundary conditions below

(i) \( u(x, 0) = \tanh(\pi x) \) : Initial Condition

(ii) \( u(0, t) = 0 \) : Boundary Condition

(iii) \( \pi e^{-t} + \frac{\partial u}{\partial t}(1, t) \) : Boundary Condition

Numerical solutions of this BVP can be obtained by discretizing the domain using \( m + 2 \) distinct points \( x_i, i = 0(1)m, \) yielding \( m + 1 \) intervals.

The programming language for the implementation is MATLAB 7.9 version. The generic parabolic PDE ((parabolic/elliptic) that Partial Differential Equation Toolbox [MATLAB (2009)] functions solve is

\[
c(x, t, u, \frac{\partial u}{\partial x}) \frac{\partial u}{\partial t} = x^{-m} \frac{\partial}{\partial x} \left( x^m f(x, t, u, \frac{\partial u}{\partial x}) \right) + s(x, t, u, \frac{\partial u}{\partial x})
\]

(6.2)

The PDE (6.2) hold for \( t_0 \leq t \leq t_f \), and \( a \leq x \leq b \) The interval\([a, b]\) must be finite. The \( m \) can be \( 0, 1, \) or \( 2 \), corresponding to slab, cylindrical, or spherical symmetry, respectively. If \( m > 0 \), then \( a \) must be at least \( 0 \), \( pl \) and \( ql \) correspond to the left boundary conditions \((x = 0)\), and \( pr \) and \( qr \) correspond to the right boundary condition \((x = 1)\). I select mesh points for the solution. I needed to specify the mesh points \((t, x)\) at which I want pdepe to evaluate the solution before I use the MATLAB PDE solver. I will specify the points as vectors \( t \) and \( x \). The vectors \( t \) and \( x \) play different roles in the solver [MathWorks (2007)] since the length of the vector \( x \) influences the cost and the accuracy of the solution. However, the computation is much less sensitive to the values in the vector \( t \).

This example requests the solution on the mesh produced by 20 equally spaced points from
the spatial interval \([0, 1]\) and five values of \(t\) from the time interval \([0, 2]\). Now, this is analogous to the heat equation I am solving in that;

**STEP1:** Transformation/Comparison

\[
c(x, t, u, \frac{\partial u}{\partial x}) = \pi^2, \quad m = 0, \quad s(x, t, u, \frac{\partial u}{\partial x}), f(x, t, u, \frac{\partial u}{\partial x}) = \frac{\partial u}{\partial x}
\]

Therefore, \(\pi^2 \frac{\partial u}{\partial t} = x_0^2 \frac{\partial}{\partial x}(x_0 \frac{\partial u}{\partial t})\)

**% STEP2:** Code the PDE as follows

function \([c, f, s] = \text{pdeHezekiah}(x, t, u, \text{DuDx})\),

**% STEP3:** Code the Initial Conditions

function \(u0 = \text{pdeHezekiah}_\text{ic}(x)\)

**% STEP4:** Code the Boundary Conditions

function \([pL, qL, pR, qR] = \text{pdeHezekiah}_\text{bc}(xL, uL, xR, uR, t)\)

### 6.1 Visualization

**% A surface plot is often a good way to study a solution**

\(m = 0;\)

\(x = \text{linspace}(0, 1, 20);\)

\(t = \text{linspace}(0, 2, 5);\)

\(\text{solutionPDE}_\text{Hezekiah} = \text{pdepe}(m, @\text{pdeHezekiah}, @\text{pdeHezekiah}_\text{ic}, @\text{pdeHezekiah}_\text{bc}, x, t);\)

\(u = \text{solutionPDE}_\text{Hezekiah}(\cdot, \cdot, 1);\)

\(\text{surf}(x, t, u);\)

\(\text{title('Numerical Solution of PDE')};\)

\(\text{xlabel('Distance x')};\)

\(\text{ylabel('Time t')};\)

**% A solution profile can also be illuminating.**

\(\text{figure};\)

\(\text{plot}(x, u(\cdot, 2));\)

\(\text{title('Solution at t = 2')};\)

\(\text{xlabel('Distance x')};\)

\(\text{ylabel('u(x,2)')};\)

% --------------------------------------------------------------

**The Content of Solution Space Contains 100 elements:** We may reshape into space of 20 by 5 or 10 by 10 or 25 by 4 etc. The Statement \(\text{reshape}(u, 25, 4)\) results in

\[
\begin{array}{cccc}
0 & 0 & 0 & 0 \\
0 & -0.0065 & -0.0524 & -0.2447 \\
0 & -0.0453 & -0.1524 & -0.3693 \\
0 & -0.0864 & -0.2171 & -0.4049 \\
0 & -0.1123 & -0.2471 & -0.4007 \\
0 & 0 & 0 & 0 \\
-0.0006 & -0.0103 & -0.0744 & -0.3127 \\
-0.0071 & -0.0598 & -0.1863 & -0.4257 \\
-0.0157 & -0.1079 & -0.2507 & -0.4464 \\
-0.0216 & -0.1371 & -0.2771 & -0.4305 \\
\end{array}
\]
<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.0013</td>
<td>-0.0160</td>
<td>-0.1035</td>
<td>-0.3915</td>
<td></td>
</tr>
<tr>
<td>-0.0147</td>
<td>-0.0773</td>
<td>-0.2250</td>
<td>-0.4849</td>
<td></td>
</tr>
<tr>
<td>-0.0318</td>
<td>-0.1316</td>
<td>-0.2865</td>
<td>-0.4876</td>
<td></td>
</tr>
<tr>
<td>-0.0434</td>
<td>-0.1629</td>
<td>-0.3078</td>
<td>-0.4590</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.0024</td>
<td>-0.0243</td>
<td>-0.1408</td>
<td>-0.4803</td>
<td></td>
</tr>
<tr>
<td>-0.0233</td>
<td>-0.0983</td>
<td>-0.2685</td>
<td>-0.5459</td>
<td></td>
</tr>
<tr>
<td>-0.0487</td>
<td>-0.1576</td>
<td>-0.3245</td>
<td>-0.5279</td>
<td></td>
</tr>
<tr>
<td>-0.0656</td>
<td>-0.1899</td>
<td>-0.3388</td>
<td>-0.4856</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.0040</td>
<td>-0.0361</td>
<td>-0.1876</td>
<td>-0.5775</td>
<td></td>
</tr>
<tr>
<td>-0.0333</td>
<td>-0.1232</td>
<td>-0.3168</td>
<td>-0.6073</td>
<td></td>
</tr>
<tr>
<td>-0.0668</td>
<td>-0.1861</td>
<td>-0.3641</td>
<td>-0.5662</td>
<td></td>
</tr>
<tr>
<td>-0.0886</td>
<td>-0.2180</td>
<td>-0.3699</td>
<td>-0.5095</td>
<td></td>
</tr>
</tbody>
</table>

2. Linear Convection Diffusion Equation

\[
\frac{\partial u}{\partial t} + \frac{\partial (h(x)u)}{\partial x} = \frac{\partial^2 u}{\partial x^2} \tag{6.3}
\]

with respective boundary and initial conditions as:

\[
u(t, -\infty) = u(t, +\infty) = 0 \tag{6.4}
\]

\[
u(0, x) = \frac{1}{1 + (x - \bar{u})^2} \tag{6.5}
\]

where \( h(x) = 3\bar{u}(x)^2 - 2\bar{u}(x) \) and \( \frac{1}{\bar{u}} + \log \left| \frac{1 - u}{1 - \bar{u}} \right| = x \). The function \( \bar{u} \) herein is an equilibrium solution to the common conservative law represented as \( \frac{\partial u}{\partial t} + \frac{\partial (u^3 - u^2)}{\partial x} = \frac{\partial^2 u}{\partial x^2} \) with conditions \( u(t, 0, -\infty) = u(t, +\infty) = 0 \).

This equation is also discretized and simulated in a similar way to the heat equation being considered as the primary PDE model in this paper.

3. Contaminant Transport PDE: This equation is given as:

\[
\frac{\partial C}{\partial t} = \frac{D}{R_d} \frac{\partial^2 C}{\partial x^2} - \frac{V_s}{R_d} \frac{\partial C}{\partial x} - \lambda_c \tag{6.6}
\]

where:

- \( \lambda_c \) = decay constant of a linear or first order decay reaction
- Initial condition is \( C = 0 \), \( x \geq 0, t = 0 \)
- First Boundary Condition is \( C = C_0, (x = 0, t > 0) \)
- Second Boundary Condition is \( \frac{\partial C}{\partial x} = 0, (x = \infty, t > 0) \)
- \( C \) = concentration of the contaminant
- \( V_s \) = seepage velocity \( (V_s = \frac{K_i}{n}) \)
- \( K \) = hydraulic conductivity of the soil
- \( n \) = porosity of soil
- \( R_d \) = Retardation factor
- \( t \) = time of transport
- \( D \) = dispersion coefficient
- \( x \) = direction of transport

In order to solve this PDE model, it’s re-written as:

\[
\frac{\partial C}{\partial t} = x^\alpha \frac{\partial}{\partial x} \left( x^\alpha \frac{D}{R_d} \frac{\partial u}{\partial x} \right) + \left( -\frac{V_s}{R_d} \frac{\partial C}{\partial x} - \lambda_c \right) \tag{6.7}
\]
and the initial and boundary conditions are coded in a similar manner to the heat equation solved in example 1.

4. System of PDEs:
The last examples in this work are systems of PDEs given as:

\[
\frac{\partial U_1}{\partial t} = 0.125 \frac{\partial^2 U}{\partial x^2} - H(U_1 + 2U_2) \tag{6.8}
\]

\[
\frac{\partial U_2}{\partial t} = 0.5 \frac{\partial^2 U}{\partial x^2} + H(U_1 + 2U_2) \tag{6.9}
\]

where \( H(\tau) = 1 - \exp(2\tau) \) and \( 0 \leq x \leq 1, t \geq 0 \) with the following initial and boundary conditions (I.Cs and B.Cs):

I.Cs \( \mapsto U_1(x,0) = 1, U_2(x,0) = 0 \) & B.Cs \( \mapsto \frac{\partial U_1}{\partial x}(0,t) = 0, U_2(0,t) = 0, U_1(1,t) = 1, \frac{\partial U_2}{\partial x}(1,t) = 0 \)

Solution Approach:
STEP 1: The PDE systems are re-written in a form solvable by the PDE Toolbox. The complete code for this PDE systems are given below:

```matlab
function [u1, u2] = PDE_System
    m = 0;
    x = 0:0.005:1;
    t = 0:0.005:2;
    sol = pdepe(m, @PDE_SystemDuDx, @PDE_System_IC, @PDE_System_BC, x, t);
    u1 = sol(:,:,1);
    u2 = sol(:,:,2);
    figure(1)
    surf(x, t, u1)
    title('U1(x,t)')
    xlabel('Distance x')
    ylabel('Time t')
    figure(2)
    surf(x, t, u2)
    title('U2(x,t)')
    xlabel('Distance x')
    ylabel('Time t')
    figure(3)
    hist(u1)
    figure(4)
    hist(u1)
end
```

```matlab
function [c, f, s] = PDE_SystemDuDx(x, t, u, DuDx)
end
```

110
\[ c = [1;1]; \]
\[ f = [0.125; 0.5] \cdot DuDx; \]
\[ y = u(1)-u(2); \]
\[ H = 1 + \exp(2*y); \]
\[ s = [-H, H]; \]
end

function u0 = PDE_System_IC(x);
  u0 = [1;0];
end

function [pl, ql, pr, qr] = PDE_System_BC(xl, ul, xr, ur, t)
  pl = [0; ul(2)];
  ql = [1;0];
  pr = [ur(1)-1; 0];
  qr = [0;1];
end

Figure 2 showing histogram and surface plot of the solutions
Figure 3 showing histogram and surface plot of the solutions

Figure 4 showing histogram and surface plot of the solutions
7 Conclusion

The primary concern in this study is modelling, simulation and visualization of heat equation. A set of three(3) other PDE models are also considered. MATLAB Finite Element Method(FEM) for Elliptic/Hyperbolic PDE was used. The entire system was built using MATLAB software. Figures 2, 3, 4, & 5 show the surface and histogram plots for the PDEs considered. Specifically, for the heat equation, it is clearly shown that the temperature moves from high level to low level showing cooling effects along the rod. The second equation was simulated in a similar way to the first equation and shows similar results. The solutions of both equations are also shown as surface diagrams in the figures. The histograms of the two solutions shows high level of concentrations towards zero points along the x-axis, showing the same proof for cooling effects. Most importantly, these Figures show that the solutions of the PDEs exist, are unique and continuous. Further works that may be carried out here are many. Such may include solving 3D models and more complicated solutions.
References


