University of Amsterdam

Exercise 2: Simulating and visualising temperature distribution in a material

Visualising Heat Diffusion in a Material

November 21, 2024

Lecturer:
dr. Rob Belleman

Student:
Kattelijn Bouma
12862444

Course:
Scientific Visualization and Virtual Reality

Course code:
5284SVVR6Y

Abstract

This report presents an animation of heat diffusion in a material, modelled using a numerical simulation based on the heat equation. The simulation captures the temperature distribution in a cross-section of a pipe system with known boundary conditions. Results are visualised using ParaView, illustrating the progression to a steady state.

1 Introduction

Heat diffusion is a fundamental physical process that describes how thermal energy spreads within a medium over time. The process is described by the heat equation, which was first developed by mathematician and physicist Joseph Fourier [Fourier, 1822]. This report presents the implementation and visualisation of a heat diffusion simulation outlined in a cartoon by Neal Wagner, University of Texas, San Antonio, which is a description of a program from the book A FORTRAN Coloring Book [Kaufman, 1978].

The problem setting from the cartoon is illustrated in Figure 1. We are looking at a cross section of a long square pipe with a second square duct running down the middle. The space between the two pipes is packed with fermented rye mash. The inner pipe is filled with steam, heated to 212 °F. The top surface is kept at a constant 100 °F. The lower part of the outer pipe is submerged in a constant temperature bath, keeping the outer wall at 32 °F. From there upwards the pipe wall increases linearly in temperature up to 100 °F.

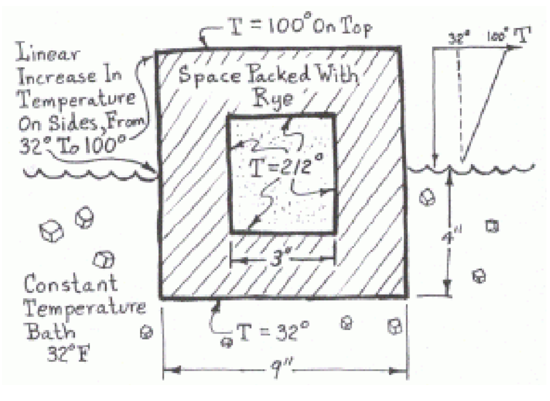

T = 100° On Top
Linear Increase In Temperature On Sides, From 32° To 100°
Space Packed With Rye
T=212°
3"
32° 100° T
4"
Constant Temperature Bath 32°F
T = 32°
9"

Figure 1: Illustration of problem setting from the cartoon by Neal Wagner.

Given the temperatures at all boundaries, the goal is to determine the exact temperature distribution within the material and create a visualisation using ParaView.

2 Methods

In this section we describe both the methods used to simulate the temperature distribution process and the visualisation pipeline.

2.1 Simulating the Diffusion Process

In order to simulate the temperature distribution, we follow the process outlined in the cartoon by Neal Wagner. The cartoon suggests discretizing the domain as a 20 × 20 grid, setting the temperature of the boundaries and inner pipe as described above, initializing the rest of the grid at 90 °F and then iteratively applying the "Magic Formula"

$$T_{i,j} = \frac{1}{4}(T_{i-1,j} + T_{i+1,j} + T_{i,j-1} + T_{i,j+1}), \tag{1}$$

which is actually equivalent to solving the steady state of the heat equation, i.e. solving Laplace's equation

$$\nabla^2 T = 0. \tag{2}$$

We have summarized the diffusion algorithm we implemented in more detail below.

Algorithm 1 Heat Diffusion Simulation

```
1: Input: $grid\_size$, $tolerance$, $max\_iterations$
2: Output: Steady-state temperature grid
3: Initialize grid
4: Set $iteration \leftarrow 0$
5: while $max\_change > tolerance$ and $iteration < max\_iterations$ do
6:     $grid \leftarrow copy(grid)$
7:     $max\_change \leftarrow 0$
8:     for each cell in left half of the grid do
9:         if cell is part of inner pipe then
10:            skip
11:        end if
12:        Update cell temperature using Equation (1)
13:        if $change > max\_change$ then
14:            Update $max\_change \leftarrow change$
15:        end if
16:    end for
17:    Mirror left half temperatures onto the right half
18:    Increment $iteration \leftarrow iteration + 1$
19: end while
```

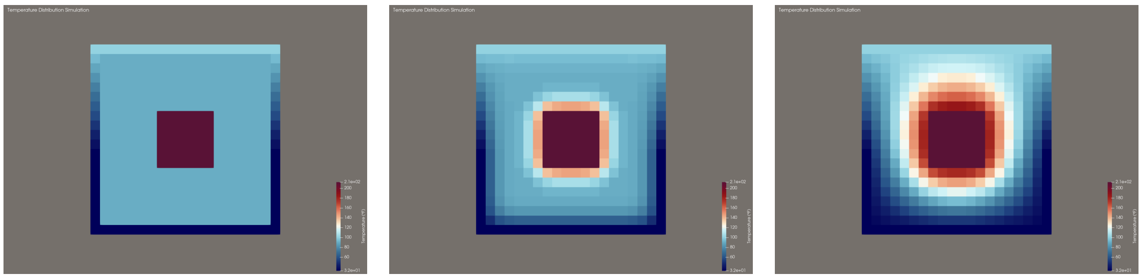
The algorithm updates the temperature of each cell based on the average temperature of its four neighbours as described by Equation (1). The largest temperature change is tracked during each iteration. Once the largest temperature change falls below a given tolerance, or a maximum number of iterations is reached, the process stops. For our simulation we use a tolerance of 0.5 as suggested by Neal Wagner. We also make use of the symmetry of the problem setting by only calculating the new temperature values for the left half of the domain and mirroring these onto the right half at the end of each iteration.

2.2 Visualisation Pipeline

After implementing and running the algorithm described in the previous section, we visualise the results of our simulation using ParaView. In order to achieve this, we make sure that our Python code produces output files that are supported by ParaView using the VTK library. We use the `vtkImageData` class to save our grid at each iteration and the `vtkXMLImageDataWriter` class to write the result to .vti files, making sure to name our files in such a way that ParaView automatically detects them as a time series. Using an appropriate colormap (cool to warm (extended)) we map temperatures to a color gradient. We also add a Text source to our visualisation pipeline to include the title. Finally, we export the time series as an animated .avi file to illustrate the progression of diffusion. We chose to save the animation using a frame rate of 3 frames per second.

3 Results

Figure 2 shows snapshots of our animation at different stages.



(a) Initial state $i = 0$ (b) Intermediate state $i = 3$ (c) Steady state $i = 37$

Figure 2: Snapshots of the heat diffusion animation at different stages: (a) initial state, (b) intermediate state, and (c) steady state.

In the animation, we observe the temperature distribution evolving over time. The central pipe transfers heat to the surrounding material, while the outer pipe is gradually cooled by the temperature bath. As the temperature difference between neighbouring cells decreases, leading to smaller changes in temperature at each iteration, the rate of diffusion slows down. The animation ends when the system reaches a steady state after 37 iterations of the algorithm.

4 Discussion

We successfully implemented the heat diffusion simulation outlined in Neal Wagner's cartoon, which models temperature distribution within a 2D grid. Furthermore, our animation effectively illustrates how the temperature in the material gradually stabilizes, with the diffusion process slowing down as the system approaches the steady state.

For future work, we could increase the grid size to improve the resolution and accuracy of the simulation, allowing for finer details in the temperature distribution. Additionally, performance improvements could be achieved by parallelizing the computation, particularly for larger grids, to reduce computation time.

References

[Fourier, 1822] Fourier, J. (1822). Théorie analytique de la chaleur.

[Kaufman, 1978] Kaufman, R. E. (1978). *A Fortran coloring book*. Mit Press.