Math 2P40 – final project cover page

Student name(s): Jaiden Dahlke

a) written report

Description of the problem:	____________	(max 15)
Analysis of the problem:	____________	(max 40)
Conclusions and discussion:	____________	(max 15)

b) supporting software:

Originality:	____________	(max 10)
Correctness:	____________	(max 10)
Coding:	____________	(max 10)

Total:	____________	(max 100)

Numerical Simulation of Heat Diffusion using 2D Finite Difference Method

Jaiden Dahlke

March 2023

Abstract

Heat diffusion is an interesting physical process that can be modeled with the 2D diffusion equation. In this project, we present a heat diffusion simulation that uses the finite difference method to approximate the derivatives in the 2D diffusion equation. The program is made in Python using NumPy and Matplotlib libraries. In the analysis, we examine the resulting temperatures after various amounts of time steps, we also discuss limitations and possible improvements that could be made to this model.

1 Introduction

Heat diffusion is an important process that occurs in a large amount of physics and engineering based problems. Understanding heat diffusion is important for designing and optimizing heat transfer systems, say for engines, or computers, as well as determining the thermal behaviours of materials and structures. The 2D diffusion equation is a frequently used model for heat diffusion; it can be solved numerically using the finite difference method. In this report we present a program that simulates heat diffusion using finite differences, then displays the corresponding temperatures on a lattice.

2 Description of Model

The program simulates heat diffusion with the 2D diffusion equation, which is given by

$$\frac{\partial T}{\partial t} = \alpha \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right)$$

where T is the temperature, t is time, x and y are the spatial coordinates, and α is the thermal diffusivity of the material. To solve this equation numerically, we use finite difference methods to approximate the derivatives in the equation. Specifically, we use the finite difference method. We approximate the first and second derivatives with respect to x and y using the expressions given in the description. Using these approximations, we can rewrite the diffusion equation as

$$T_{i,j,t+\Delta t} = T_{i,j,t} + \alpha\Delta t \left(\frac{T_{i+1,j,t} - 2T_{i,j,t} + T_{i-1,j,t}}{\Delta x^2} + \frac{T_{i,j+1,t} - 2T_{i,j,t} + T_{i,j-1,t}}{\Delta y^2} \right)$$

where $T_{i,j,t}$ represents the temperature at node (i,j) at time t, and Δt is the time step. We initialize the lattice with zeros and set the temperature of some squares on the lattice to 100 degrees. We then set the boundary conditions to zero temperature and iterate the simulation until the temperature distribution reaches a steady state.

3 Description of the program

The program is implemented in Python using the NumPy and Matplotlib libraries. We define the lattice size and time steps using the variables nrows, ncols, num_steps, dt, dx, dy, and alpha. We initialize the lattice with zeros using the np.zeros() function and set the temperature of some squares on the lattice using array slicing. We then set the boundary conditions to zero temperature using array indexing and iterate the simulation using a for loop over the range of num_steps. Within the loop, we copy the current lattice to a new lattice for the next time step using the copy() method. Them, we update the temperature of each node on the lattice using the explicit finite difference method, as described above. Finally, we set the boundary conditions to zero temperature on the new lattice and update the current lattice for the next time step. After the simulation is complete, a GIF of the evolution of the system is outputted along with the individual frames for further analysis. Additionally the average temperature with respect to time step is plotted along with its derivative.

4 Analysis Based Off Evidence

To analyze the behavior of the heat diffusion model implemented in the code, we can observe the temperature distribution at different time steps. The following figures show the temperature distribution at different time steps, starting from an initial condition where a square region in the center of the lattice is set to a high temperature of 100 degrees, and the rest of the lattice is set to zero temperature.

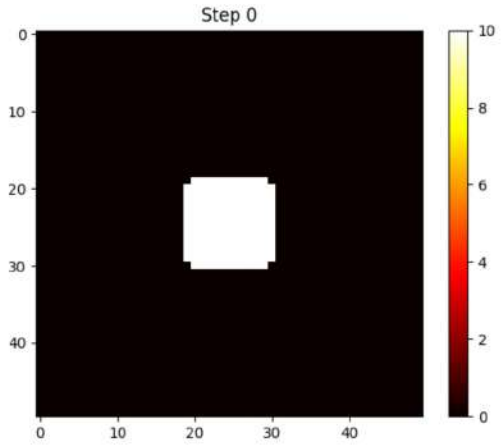

Figure 1: Temperature distribution at time step 0

At time step 0 the system hasn't evolved yet, so this only shows the initial condition of the system.

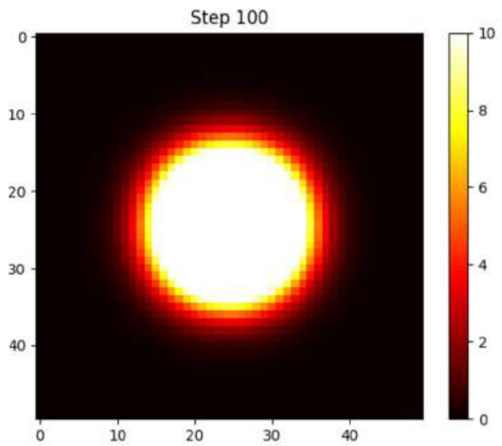

Figure 2: Temperature distribution at time step 100

At time step 100, the high temperature region has started to diffuse. The rate at which the temperature drops off as we move out of the center is rapid.

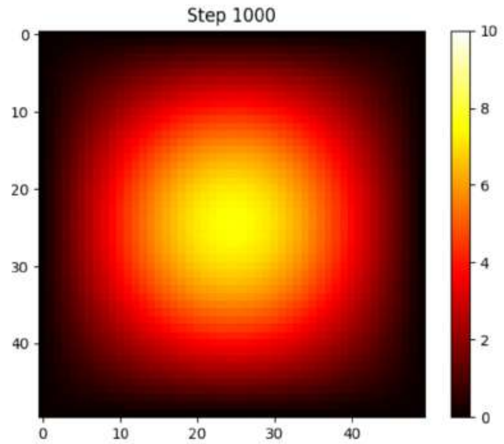

Figure 3: Temperature distribution at time step 1000

At time step 1000, the temperature at the center has started to decrease and spread to further surrounding regions.

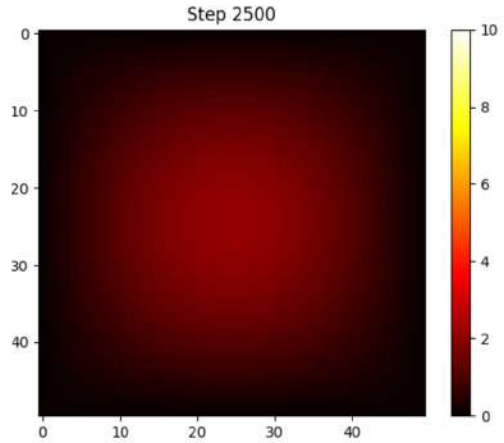

Figure 4: Temperature distribution at time step 2500

At time step 2500, the temperature is a relatively homogeneous red in the $[10,40]\times[10,40]$ region.

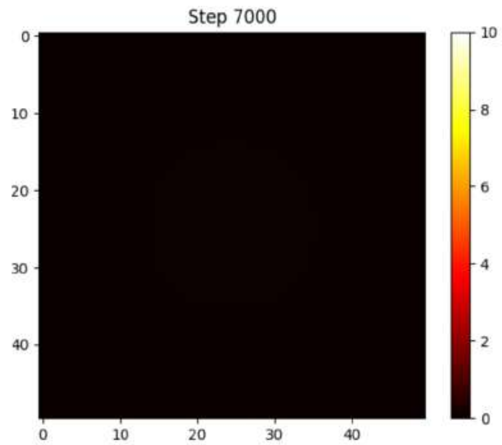

Figure 5: Temperature distribution at time step 7000

By time step 7000, the temperature has reached an equilibrium of 0. This accurately reflects the real world because, from observation, we know heat stops diffusing when the object reaches the temperature of the ambient space.

Another question: how rapidly does the heat dissipate? Clearly, the temperature diffuses more rapidly at the beginning than at the end, but knowing the exact rate of cooling would be informative. To answer this question we plot the average lattice temperature every 50 steps.

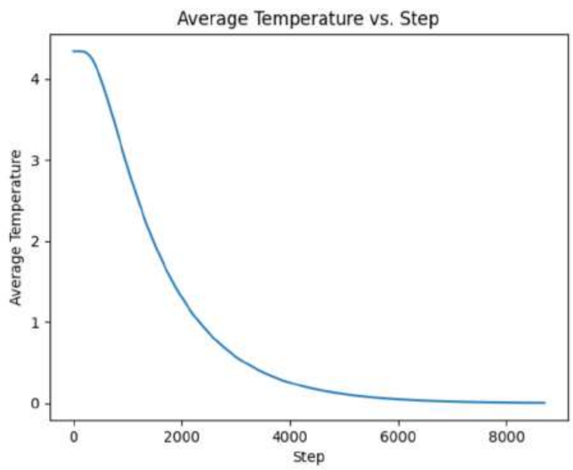

Figure 6: Average Temperature

As we can see, the median lattice temperature decreases roughly 50% by step 2000.

Approximating the derivative we can see that the maximum rate of change of average temperature occurs when n = 800

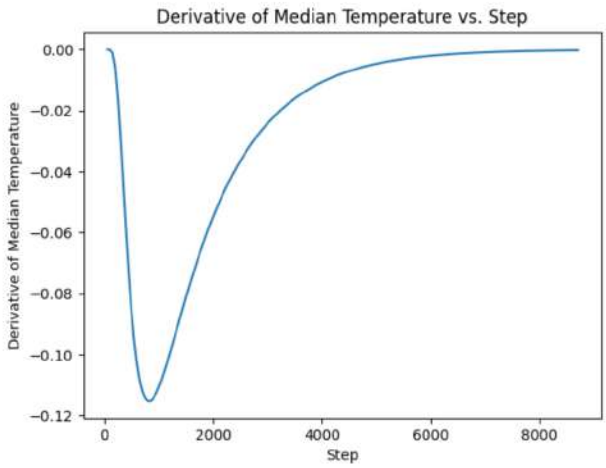

Figure 7: Rate of Change of Average Temperature

Again, generally speaking, these observations agree with reality, assuming constant ambient conditions. We should expect an object to rapidly cool after being heated. One of the counterintuitive results is that an objects doesn't cool the fastest immediately after being heated, its most rapid cooling occurs slightly after that point.

5 Limitations

The limitations of the program include the assumption of constant thermal diffusivity, which may not hold in certain materials. Additionally, the program assumes that the material is homogeneous and isotropic, which may not be the case in real-world scenarios.

To extend the program, we could incorporate more sophisticated diffusion equations that take into account variations in thermal diffusivity and anisotropy. We could also introduce boundary conditions that better reflect the physical conditions of the problem, such as a non-zero temperature gradient or a convective heat transfer at the boundaries.

Another potential extension is to model heat diffusion in three dimensions, which would require solving the 3D diffusion equation. This would allow us to study more complex geometries and scenarios, such as heat diffusion in a cylindrical or spherical object.

Overall, while the program has some limitations, it provides a useful tool for studying heat diffusion in 2D systems. By extending the program to incorporate more realistic physical conditions and scenarios, we can gain valuable insights into the behavior of heat diffusion in a wide range of materials and systems.

6 Conclusion

In this report, we have described a simple implementation of a 2D heat diffusion model using the finite difference method. The model was implemented in Python using the NumPy and Matplotlib libraries, and was used to simulate the diffusion of heat through a lattice. We observed that the high temperature region gradually diffused throughout the lattice, with the temperature decreasing as the heat spread out. Eventually, the temperature distribution reached a steady state, where the it was approximately uniform throughout the lattice. Despite being an approximation, we observed that the model demonstrates what we'd expect to happen. The diffusion rapidly occurs immediately after being exposed to the heat source, then begins to slow down. An interesting result of our model is it demonstrates that the most rapid decrease in median temperature occurs about 800 steps into the model. This is interesting because intuition may tell you that the most rapid cooling occurs at the very start of the simulation. This simple model can be extended to simulate more complex heat diffusion phenomena, and can be used as a starting point for further research in this field.

References

[1] LeVeque, R. J. (2007). Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems. SIAM.

[2] Wikipedia. (2023). Finite difference method. Retrieved March 19, 2023, from `https://en.wikipedia.org/wiki/Finite_difference_method`

[3] Wikipedia. (2023). Heat equation. Retrieved March 19, 2023, from `https://en.wikipedia.org/wiki/Heat_equation`

[4] Wikipedia. (2023). Numerical differentiation. Retrieved March 19, 2023, from `https://en.wikipedia.org/wiki/Numerical_differentiation`