Workshop 3

Juan David Martinez - Santiago Uribe Luna

Escuela de Ingeniería, Ciencia y Tecnología, Universidad del Rosario

March 2024

Task 1

Consider the one-dimensional heat equation:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2} \tag{1}$$

where $u(x,t)$ is the temperature at position x and time t, and α is the thermal diffusivity.

Using the finite difference method, we discretize the equation as:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \alpha \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2} \tag{2}$$

where $u_j^n = u(j\Delta x, n\Delta t)$, Δx is the spatial step, and Δt is the time step.

Let $u_j^n = \xi^n e^{ijkh}$, where $i = \sqrt{-1}$, k is the wavenumber, and $h = \Delta x$. Substituting this into the discretized equation:

$$\frac{\xi^{n+1} e^{ijkh} - \xi^n e^{ijkh}}{\Delta t} = \alpha \frac{\xi^n e^{i(j+1)kh} - 2\xi^n e^{ijkh} + \xi^n e^{i(j-1)kh}}{\Delta x^2} \tag{3}$$

Simplifying and solving for the amplification factor ξ:

$$\xi = \frac{\xi^{n+1}}{\xi^n} = 1 + \frac{\alpha \Delta t}{\Delta x^2}(e^{ikh} - 2 + e^{-ikh}) \tag{4}$$

Using Euler's formula, $e^{ikh} = \cos(kh) + i\sin(kh)$, we get:

$$\xi = 1 - \frac{4\alpha\Delta t}{\Delta x^2} \sin^2\left(\frac{kh}{2}\right) \tag{5}$$

For stability, $|\xi| \leq 1$ must hold for all wavenumbers k. The maximum value of $\sin^2\left(\frac{kh}{2}\right)$ is 1, so the stability condition is:

$$\frac{\alpha \Delta t}{\Delta x^2} \leq \frac{1}{2} \tag{6}$$

This condition is known as the Courant-Friedrichs-Lewy (CFL) condition for the heat equation.

The Von Neumann stability analysis of the heat equation shows that the discretization is stable if the CFL condition is satisfied. This condition imposes a restriction on the time step size Δt based on the spatial step size Δx and the thermal diffusivity α.

Task 2

Consider the following specific problem

$$\begin{aligned} u_t &= u_{xx}, \quad \text{for} \quad 0 < x < \pi \quad \text{and} \quad 0 < t < T, \\ u(0,t) &= u(\pi,t) = 0, \\ u(x,0) &= \phi(x) = \begin{cases} x, & \text{for } x \in (0, \pi/2). \\ \pi - x, & \text{for } \pi \in (\pi/2, \pi). \end{cases} \end{aligned}$$

Using $\Delta x = \frac{\pi}{20}$ choose a suitable value for Δt and compute the numerical solution. Plot the solution at different times from 0 to $T = \frac{3\pi^2}{80}$.

Here is the python code that we generate in order to plot the solution.

```python
# Numerical solution of the heat equation
# u_t = k * u_xx
# with Dirichlet boundary conditions
# u(0, t) = u(1, t) = 0
# and initial condition
# u(x, 0) = f(x)

import numpy as np

# Set the size of the domain
L = np.pi
tf = 3*np.pi**2 / 80
# Set dx
dx = np.pi / 20
# Set dt
dt = np.pi**2 / 1600

# Initial condition
def f(x):
    if 0 < x and x < np.pi/2:
        return x
    elif np.pi/2 <= x and x < np.pi:
        return np.pi - x

# Set the number of points in the x (space) direction
n = int(L / dx) + 1
# Set the number of points in the t (time) direction
m = int(1 / dt) + 1

# Set the grid
x = np.linspace(0, L, n)
t = np.linspace(0, tf, m)
u = np.zeros((n, m))

# Set the initial condition
for i in range(n):
    u[i, 0] = f(x[i])

# Set the boundary conditions
u[0, :] = 0
u[-1, :] = 0

# Set the diffusion constant
k = 1

# Set the coefficient matrix
r = k * dt / dx**2

# Time-stepping loop
for j in range(m - 1):
    for i in range(1, n - 1):
        u[i, j + 1] = r * u[i - 1, j] + (1 - 2 * r) * u[i, j] + r * u[i + 1, j]

# Create the meshgrid
T, X = np.meshgrid(t, x)
```

Listing 1: Python code

```python
import matplotlib.pyplot as plt

# Create the 3D plot and 2D plots over time side by side
fig = plt.figure(figsize=(14, 6))
fig.suptitle('Temperature evolution')

# 3D plot
ax = fig.add_subplot(1, 2, 1, projection='3d')
ax.view_init(30, 30)
ax.plot_surface(X, T, u, cmap='autumn_r')
ax.set_xlabel('x')
ax.set_ylabel('t')
ax.set_zlabel('Temperature')

# 2D plots of t = 0, t = 0.5, t = 1
ax = fig.add_subplot(1, 2, 2)
ax.plot(x, u[:, 0], label='t=0')
ax.plot(x, u[:, int(m/2)], label='t=0.5')
ax.plot(x, u[:, -1], label='t=1')
ax.set_xlabel('x')
ax.set_ylabel('Temperature')
ax.legend()

plt.show()
```

Listing 2: Python code for plotting

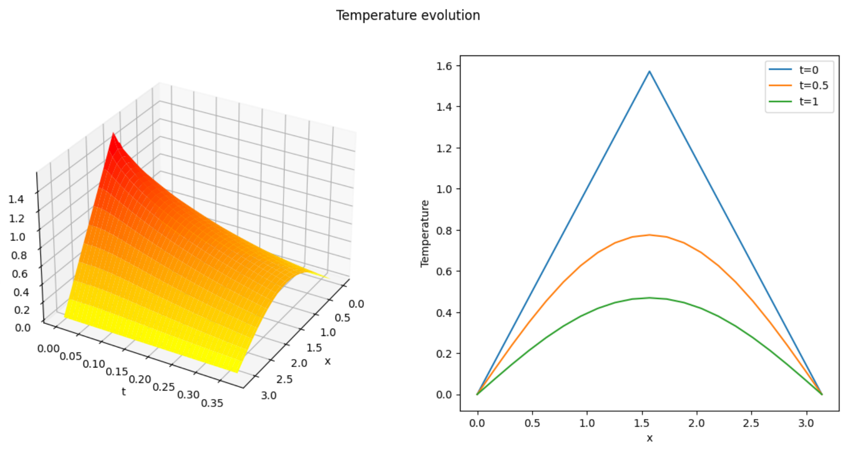

Figura 1: Task 2 plot

Task 3

What is the probability that at $T = 1$ the particle lies in the interval (a, b)?

Initially, we analyze the average and the dispersion of the issue. Given that the average is 0, we proceed to compute the dispersion using the provided values of T and Δt. The total dispersion is calculated as $n * \operatorname{Var}(X_i)$, where, $n = \frac{T}{\Delta t}$ and $\operatorname{Var}(X_i) = \Delta t$. Consequently, the total dispersion equals T. Subsequently, we determine the probability density function (PDF) to derive the outcome.

We designate W as the summation of the stochastic variables denoting the particles' positions.

$$P(a < W < b) = \int_a^b \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dz$$

$= \int_a^b \frac{e^{-\frac{z^2}{2}}}{\sqrt{2}-\sqrt{\pi}} dz$, we substitute with $u = \frac{z}{\sqrt{2}}$ and $du = \frac{1}{\sqrt{2}}$ $\frac{1}{2}\int_a^b \frac{2e^{-u^2}}{\sqrt{\pi}} du$, now we consider the error function of gauss

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

we replace in the function $\frac{1}{2}\int_a^b \frac{2e^{-u^2}}{\sqrt{\pi}} du$ we have that $\frac{er\,f(u)}{2} = \frac{er\,f\left(\frac{\pi}{\sqrt{2}}\right)}{2}$ Finally we evaluate in a and b :

$$P(a < W < b) = \frac{\operatorname{erf}\left(\frac{b}{\sqrt{2}}\right) - \operatorname{erf}\left(\frac{a}{\sqrt{2}}\right)}{2}$$

Task 4

What is the value of the diffusion coefficient (the constant k) corresponding to this case?

Solution

Given the heat kernel and the normal distribution for this case, we have that:

$$K_t(x) = \frac{1}{\sqrt{4\pi k t}} e^{-\frac{x^2}{4kt}}, N(0, \Delta t) = \frac{1}{\sqrt{2\pi \Delta t}} e^{-\frac{x^2}{2\Delta t}}$$

So we can see that with $k = \frac{1}{2}$ the equations are equal, therefore, that is the constant corresponding to this case.

Monte Carlo method for the heat equation

Use a random number generator to sample N (make a sample of the order of 10^3) numbers from the uniform distribution on the interval, $|x| \leq \frac{1}{2}$ Each point now undergoes a random walk exactly as explained above (note that the initial position of the walk is no longer at $x=0$).

Task 5

Compute the probability of finding a point inside the interval (a, b) at $T=1$.

```matlab
% Parameters
N = 1000; % Number of random points 10^3
T = 1; % Time
% Generate random numbers to evaluate
x_initial = rand(1, N) - 0.5; % Random points in [-0.5, 0.5]
% Perform random walks
x_final = x_initial + sqrt(2 + T) * randn(1, N);

% Define the interval (a, b)
a = -0.2;
b = 0.2;

% Create histogram
edges = linspace(-0.5, 0.5, 21); % Define edges for histogram bins
counts = histogram(x_final, edges);

% Find the bin indices for (a, b)
index_a = find(edges > a, 1);
index_b = find(edges < b, 1, 'last');

% Count the number of points inside (a, b)
count_inside = sum(counts.Values(index_a:index_b));
% Compute the probability
probability = count_inside / N;
fprintf('Probability of finding a point inside the interval (%.2f, %.2f) at
    T=1: %.4f\n', a, b, probability);

% Plot histogram with the interval highlighted
figure;
histogram(x_final, edges, 'FaceColor', [0.5, 0.5, 0.5]);
hold on;
line([a, a], [0, max(counts.Values)], 'Color', 'r', 'LineWidth', 2, '
    LineStyle', '--');
line([b, b], [0, max(counts.Values)], 'Color', 'r', 'LineWidth', 2, '
    LineStyle', '--');
xlabel('Value');
ylabel('Frequency');
title('Histogram of Final Positions');
legend('Histogram', 'Interval (a, b)', 'Location', 'best');
grid on;
hold off;
```

Listing 3: MATLAB code

Solution:
we can see in the (listing 1) that one of the examples evaluating it in the interval (-0.20,0.20) would be the following diagram:

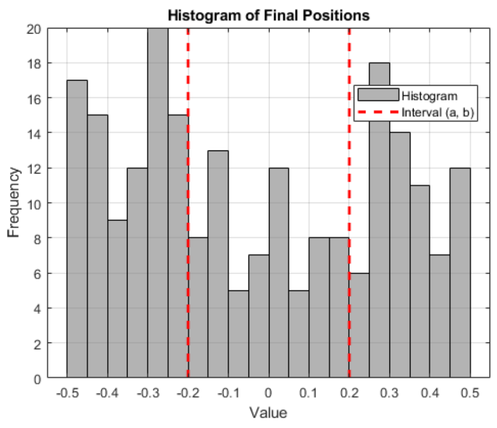

Figura 2: Diagram

and the answer it gives us is:
Probability of finding a point inside the interval (-0.20, 0.20) at T=1: 0.0580
With other random numbers:

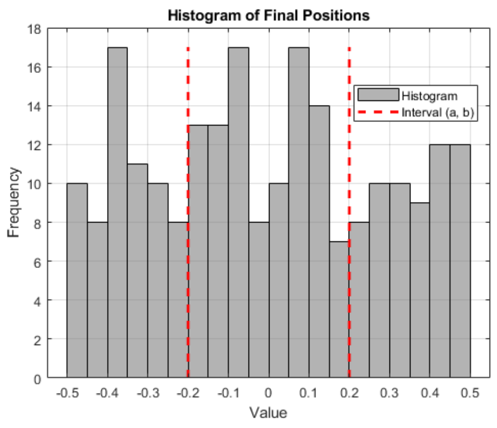

Figura 3: Diagram 2

and the answer it gives us is:
Probability of finding a point inside the interval (-0.20, 0.20) at T=1: 0.0860

Task 6

Find an exact expression for the probability distribution of the particles at $T = 1$ and compare your results.
The one-dimensional diffusion equation for the probability density $P(x,t)$ is given by:

$$\frac{\partial P(x,t)}{\partial t} = D\frac{\partial^2 P(x,t)}{\partial x^2}$$

Let D denote the diffusion coefficient, and in this particular instance, $D = 1$ (for ease of computation). The fundamental solution or diffusion kernel to this diffusion equation, given an initial condition $P(x,0) = \delta(x)$ (where $\delta(x)$ represents the Dirac delta function), is well-established and expressed as:

$$P(x,t) = \frac{1}{\sqrt{4\pi t}} \exp\left(-\frac{x^2}{4t}\right)$$

This represents the Gaussian or standard normal distribution, characterized by a mean of $\mu = 0$ and a variance of $\sigma^2 = 2t$. Hence, the distribution of probabilities for the particles at $T = 1$ corresponds straightforwardly to the evaluation of the Gaussian distribution at $t = 1$:

$$P(x,1) = \frac{1}{\sqrt{4\pi}} \exp\left(-\frac{x^2}{4}\right)$$