Project 1

November 13, 2024

Antonio Pampalone 23586519
Giuseppe Pisante 23610012
Martina Raffaelli 23616907

Task 1.0:

The boundary conditions are consistent with each other because they impose a continuous solution at the boundaries, and they set identical values at the points where they meet. However, the initial conditions are not consistent with the boundary conditions, since they impose different values on the edges of the domain than those specified by the boundary conditions. Because of the non consistency, we consider the initial conditions to be applied only on the inner part of the domain: $\Omega \setminus \partial\Omega$.

Task 1.1:

Discretization with central differencing scheme in space and Crank-Nicolson method in time:

Starting from the equation reported below, we first discretize the left-hand side term using Crank-Nicolson (C-N) method in time and then we discretize the right-hand side term following the central differencing scheme (CDS).

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \tag{1}$$

By applying C-N method to the time derivative we get the approximation:

$$\frac{\partial T}{\partial t} \approx \left(\frac{\partial T}{\partial t}\right)_{i,j} = \frac{T_{i,j}^{n+1} - T_{i,j}^{n}}{\Delta t} = \frac{1}{2}\left(\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right)^{n} + \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right)^{n+1}\right) \tag{2}$$

By applying CDS to both the spatial derivatives we get the approximations:

$$\frac{\partial^2 T}{\partial x^2} \approx \left(\frac{\partial^2 T}{\partial x^2}\right)_{i,j} = \frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{(\Delta x)^2}, \frac{\partial^2 T}{\partial y^2} \approx \left(\frac{\partial^2 T}{\partial y^2}\right)_{i,j} = \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{(\Delta y)^2} \tag{3}$$

Now we substitute (3) into (2) as follows:

$$\begin{aligned}\frac{T_{i,j}^{n+1} - T_{i,j}^{n}}{\Delta t} = \frac{1}{2}\Bigg(&\frac{T_{i+1,j}^{n} - 2T_{i,j}^{n} + T_{i-1,j}^{n}}{\Delta x^2} \\ &+ \frac{T_{i,j+1}^{n} - 2T_{i,j}^{n} + T_{i,j-1}^{n}}{\Delta y^2} + \frac{T_{i+1,j}^{n+1} - 2T_{i,j}^{n+1} + T_{i-1,j}^{n+1}}{\Delta x^2} \\ &+ \frac{T_{i,j+1}^{n+1} - 2T_{i,j}^{n+1} + T_{i,j-1}^{n+1}}{\Delta y^2}\Bigg)\end{aligned} \tag{4}$$

and we finally get:

$$\begin{aligned}T_{i,j}^{n+1} = T_{i,j}^{n} + \frac{\Delta t}{2}\Bigg(&\frac{T_{i+1,j}^{n} - 2T_{i,j}^{n} + T_{i-1,j}^{n}}{\Delta x^2} \\ &+ \frac{T_{i,j+1}^{n} - 2T_{i,j}^{n} + T_{i,j-1}^{n}}{\Delta y^2} + \frac{T_{i+1,j}^{n+1} - 2T_{i,j}^{n+1} + T_{i-1,j}^{n+1}}{\Delta x^2} \\ &+ \frac{T_{i,j+1}^{n+1} - 2T_{i,j}^{n+1} + T_{i,j-1}^{n+1}}{\Delta y^2}\Bigg)\end{aligned} \tag{5}$$

Stability of Crank-Nicolson method:

The Crank–Nicolson method is a numerically stable method that is unconditionally stable for diffusion equations, meaning that its stability does not depend on the time step size Δt. However, the method may have an oscillatory behavior when using large time steps, high initial gradients, or in cases with transport-dominated problems. These oscillations are a characteristic behavior of the method for certain problem types, not a sign of instability.

Truncation error of Crank-Nicolson method:

To compute the truncation error for the Crank–Nicolson method, we need to compare the continuous time derivative with the discrete approximation provided by the method.

The Crank–Nicolson scheme updates the solution $T_{i,j}$ in time as follows:

$$T_{i,j}^{n+1} = T_{i,j}^{n} + \frac{\Delta t}{2}\left(f(T_{i,j}^{n+1}) + f(T_{i,j}^{n})\right), \tag{6}$$

To analyze the truncation error, we compare this discrete form to the exact time derivative using Taylor series expansion around $T(t)$ reported below

$$T(t+\Delta t) = T(t) + \frac{\partial T}{\partial t}\Delta t + \frac{\partial^2 T}{\partial t^2}\frac{\Delta t^2}{2!} + \frac{\partial^3 T}{\partial t^3}\frac{\Delta t^3}{3!} + O(\Delta t^4),$$
$$T(t-\Delta t) = T(t) - \frac{\partial T}{\partial t}\Delta t + \frac{\partial^2 T}{\partial t^2}\frac{\Delta t^2}{2!} - \frac{\partial^3 T}{\partial t^3}\frac{\Delta t^3}{3!} + O(\Delta t^4).$$

The Crank–Nicolson approximation for the time derivative is obtained by using the average of forward and backward differences:

$$\frac{T(t+\Delta t) - T(t-\Delta t)}{2} = \frac{\partial T}{\partial t} + \frac{\partial^3 T}{\partial t^3}\frac{\Delta t^2}{6} + O(\Delta t^4).$$

The truncation error τ is defined as the difference between the exact derivative $\frac{\partial T}{\partial t}$ and the Crank–Nicolson approximation:

$$\tau = \frac{\partial T}{\partial t} - \left(\frac{\partial T}{\partial t} + \frac{\partial^3 T}{\partial t^3}\frac{\Delta t^2}{6} + O(\Delta t^4)\right).$$

Simplifying this expression, we find the truncation error for the Crank–Nicolson:

$$\tau = -\frac{\partial^3 T}{\partial t^3}\frac{\Delta t^2}{6} + O(\Delta t^4). \tag{7}$$

This shows that the Crank–Nicolson method is second-order accurate in time because the leading term in the truncation error is proportional to Δt^2.

Total truncation error:

The total truncation error of the entire numerical scheme is the sum of the truncation errors from each component ($\frac{\partial^2 T}{\partial x^2}$, $\frac{\partial^2 T}{\partial y^2}$, and $\frac{\partial T}{\partial t}$), since we are combining these components to approximate the PDE as a whole.

Since the contribution of the time discretization has been previously computed, we can compute only one of the spatial contributes due to their similarity. The central difference approximation for the second derivative $\frac{\partial^2 T}{\partial x^2}$ is given by:

$$\frac{\partial^2 T}{\partial x^2} \approx \frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{\Delta x^2} \tag{8}$$

We can expand the terms $T_{i+1,j}$ and $T_{i-1,j}$ using the Taylor expansion as follows:

$$T_{i+1,j} = T_{i,j} + \Delta x\frac{\partial T}{\partial x} + \frac{\Delta x^2}{2}\frac{\partial^2 T}{\partial x^2} + \frac{\Delta x^3}{6}\frac{\partial^3 T}{\partial x^3} + \frac{\Delta x^4}{24}\frac{\partial^4 T}{\partial x^4} + O(\Delta x^5) \tag{9}$$

$$T_{i-1,j} = T_{i,j} - \Delta x\frac{\partial T}{\partial x} + \frac{\Delta x^2}{2}\frac{\partial^2 T}{\partial x^2} - \frac{\Delta x^3}{6}\frac{\partial^3 T}{\partial x^3} + \frac{\Delta x^4}{24}\frac{\partial^4 T}{\partial x^4} + O(\Delta x^5) \tag{10}$$

and then we substitute them in the finite-difference expression finding:

$$\frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{\Delta x^2} = \frac{\partial^2 T}{\partial x^2} + \frac{\Delta x^2}{12}\frac{\partial^4 T}{\partial x^4} + O(\Delta x^4) \tag{11}$$

Similarly we find the contribution of the truncation error of the second derivative $\frac{\partial^2 T}{\partial y^2}$:

$$\frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{\Delta x^2} = \frac{\partial^2 T}{\partial x^2} + \frac{\Delta x^2}{12}\frac{\partial^4 T}{\partial x^4} + O(\Delta x^4) \tag{12}$$

In the end the total truncation error is computed by adding the single contributions (7), (11), and (12):

$$\tau = -\frac{\partial^3 T}{\partial t^3}\frac{\Delta t^2}{6} + \frac{\Delta x^2}{12}\frac{\partial^4 T}{\partial x^4} + \frac{\Delta y^2}{12}\frac{\partial^4 T}{\partial y^4} + O(\Delta t^4, \Delta y^4, \Delta y^4). \tag{13}$$

Task 1.3:

The time evolution of the temperature at $x = y = 0.4$, reported below, has been generated in the file plotsgenerator.py which can be found in the Github repository [1].

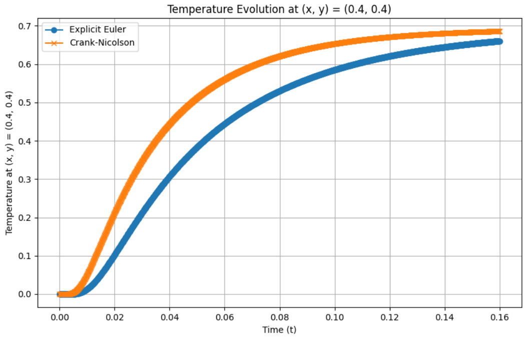


Figure 1: Temperature evolution for point (x, y) = (0.4, 0.4) with $\Delta t = 0.0001$

From the picture we can see that both methods show a gradual increase in temperature as time progresses, which is consistent with the expected behavior in heat diffusion problems, where the temperature generally rises and stabilizes as the system approaches a steady state. Crank-Nicolson solution reaches a higher temperature faster than the Explicit Euler solution, probably because it captures more accurately the diffusion process. However both solutions approach a steady state when the time progresses: Explicit Euler reaches this state more gradually, while Crank-Nicolson reaches it earlier indicating faster convergence.

The vertical temperature profile at $t = 0.16$ and $x = 0.4$ can be generated as well using the file plotsgenerator.py, and it is reported below.

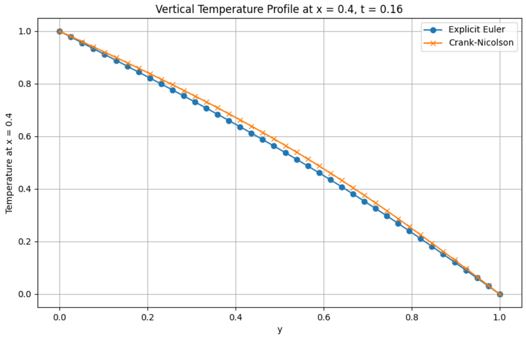


Figure 2: Vertical temperature profile at $t = 0.16$ and $x = 0.4$ with $\Delta t = 0.0001$

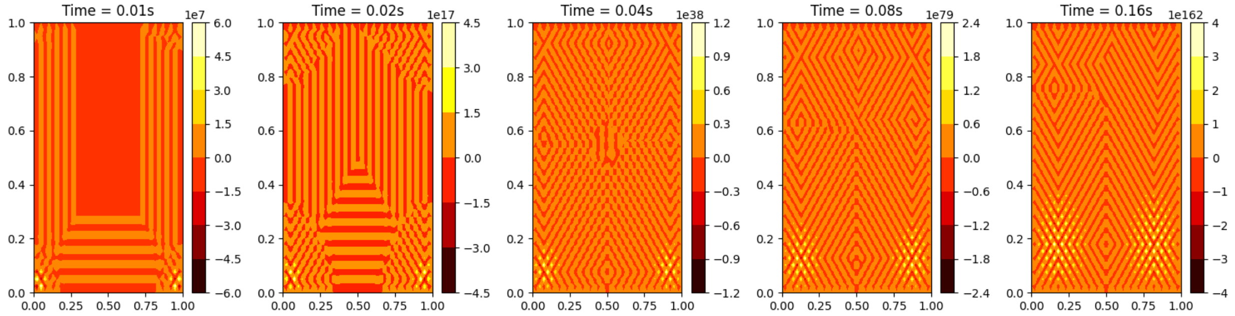
The temperature profile decreases smoothly from $y = 0$ to $y = 1$, indicating a gradient that is consistent with heat diffusion from a higher temperature boundary to a lower one. Both methods produce very similar results across the entire y-axis. This similarity suggests that by time $t = 0.16$ both methods have closely approximated the true temperature distribution, so the differences in the initial evolution of the solution reduce as the solution approaches the steady-state.

In evaluating the performance of the Explicit Euler and Crank-Nicolson methods in the implementation of a CFD solver, several observations can be made. Using a spatial discretization of $\Delta x = \Delta y = \frac{1}{40}$ and 1600 time steps, the Explicit Euler method required 0.1449 seconds and the Crank-Nicolson method took 10.9192 seconds.

While the Explicit Euler method demonstrates a significant advantage in computational speed, the Crank-Nicolson method offers higher accuracy, being a second-order method in both time and space. Additionally, a crucial advantage of the Crank-Nicolson scheme is its unconditional stability, which allows it to operate without needing to adjust the time step Δt in relation to Δx. In contrast, the Explicit Euler method requires that Δx^2 and Δt maintain a specific ratio to satisfy the CFL (Courant-Friedrichs-Lewy) condition for stability. This constraint can limit the flexibility and accuracy of the Explicit Euler approach when finer spatial discretizations are needed.

Task 1.4:

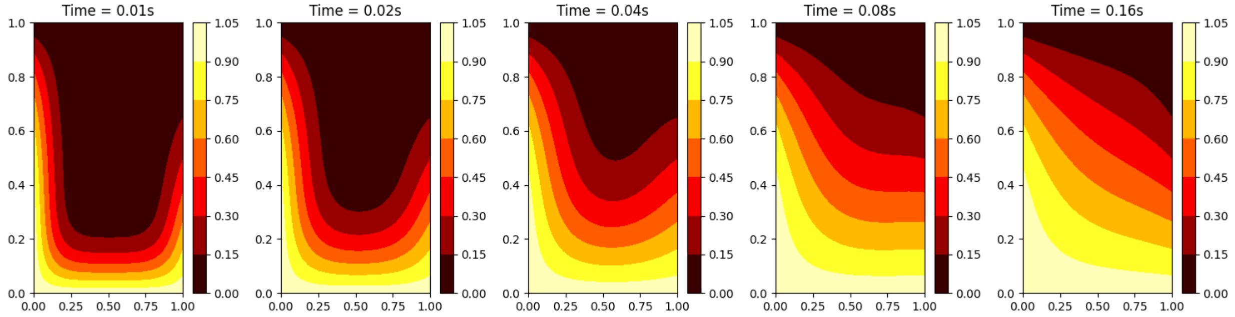
Yes, there are noticeable differences in the contour plots when changing the spatial resolution and/or the time step. Using a fixed spatial resolution of $\Delta x = \Delta y = \frac{1}{40}$ with a number of time steps of 160, we observe that the solution becomes unstable because the time step violates the Von Neumann stability criterion for the explicit Euler method.



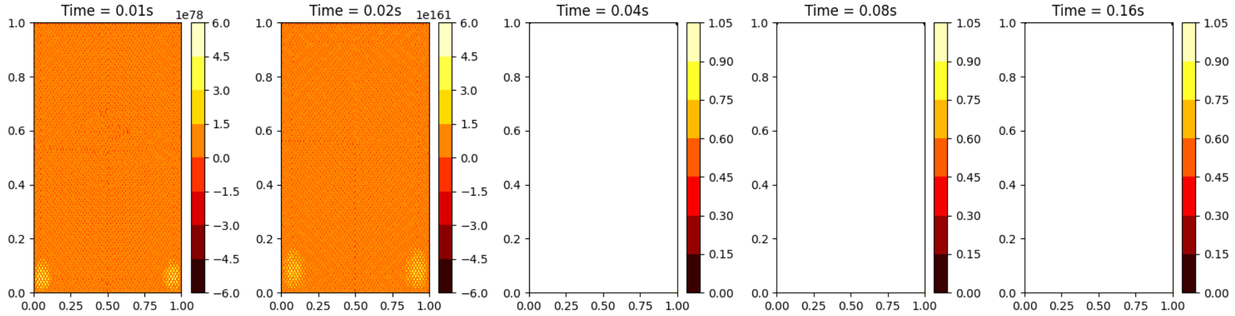
The stability condition ensures that errors introduced at each time step do not grow exponentially as the computation progresses. In particular, for a 2D heat diffusion problem, this condition can be expressed as:

$$\Delta t \leq \frac{\Delta x^2}{4}$$

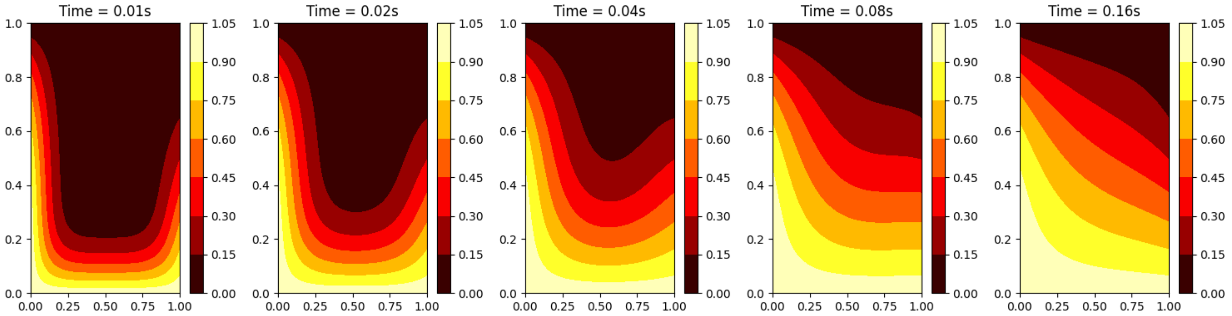
which implies that the minimum required number of time steps must be at least 1024. Infact, using a number of time steps of 1600 the solution becomes stable:



Similarly, using a finer spatial resolution of $\Delta x = \Delta y = \frac{1}{100}$ with a number of time steps of 1600 the solution becomes unstable because the minimum number of time steps for stability must be 6400.



Higher spatial resolution allows for a more accurate representation of temperature but requires smaller time steps to maintain stability. Infact, with a number of time steps of 6400, the solution becomes stable:



The stability of the solution depends on the ratio between Δt and Δx. For finer grids, a smaller Δt is required to avoid instability. The choice of time values that are factors of 2 apart ($t = 0.01, 0.02, 0.04, 0.08, 0.16$) helps us understand how heat diffusion evolves over time, allowing us to observe the progression of the temperature field at exponentially increasing intervals. Using exponentially increasing time steps provides a clear and efficient view of both the rapid initial diffusion and the slower, later stages, without requiring an excessive number of snapshots.

References

[1] *CFD Repository*,
Available at: `https://github.com/GiuseppePisante/CFD.git`

[2] *GitHub Copilot*,
GitHub. Available at: `https://github.com/features/copilot`