Laplace PDE Solution Using the Finite Difference Method

Numerical Method Report

Kessad Mohamed Dhia Eddine

Abstract

This report presents the numerical solution for Laplace's equation. The analytical solution of the problem was found in order to determine the error range of the method. A physical model was introduced to interpret the results, and different parameters were studied to understand the behavior of the numerical method under different conditions.

This project was prepared by members of Lithium Club

1

Overview

1.1 Introduction

Laplace's equation is one of the simplest examples of the elliptic type of PDE, taking the form:

$$\nabla^2 U = 0$$

It is widely used for modeling potential field problems in electrostatics, fluid flow, and heat conduction. In some problems, it is hard to determine the exact solution, so different numerical methods were developed to fill the gap. In this paper, a problem is solved both analytically and numerically, in order to study the error and stability of the numerical methods used to approximate solutions for partial differential equations.
When operating power electronics, elevated temperatures can be reached, which may cause damage and reduce component efficiency. To prevent this, a heat sink is used to keep the temperature at a controlled level. One such example is a heat sink attached to a transistor.

1.2 Problem

The heat distribution $u_{x,y}$ is governed by Laplace's equation:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \quad (x, y) \in (0, 1) \times (0, 1)$$

Subject to the boundary conditions:

$$\begin{aligned} u_{0,y} &= 0, \\ u_{1,y} &= 0, \\ u_{x,0} &= 0, \\ u_{x,1} &= \sin(\pi x) \end{aligned}$$

For simplicity, the dimensions are normalized. The boundary setup is:

- Bottom and sides at $0°$
- Top edge heated with $\sin(\pi x)$

Transistor with its heat sink.

1.3 Analytical Solution

The method of separation of variables is used to solve the Laplace PDE problem[?].

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = 0$$

Assuming that U is of the form:

$$U_{x,y} = X_x Y_y$$

Which, after substitution, gives:

$$X''_x Y_y + X_x Y''_y = 0$$

$$\frac{X''_x}{X_x} + \frac{Y''_y}{Y_y} = 0$$

$$\frac{X''_x}{X_x} = -\frac{Y''_y}{Y_y} = \lambda$$

λ is a constant, since the two sides of the equality are independent of each other. This leads to two ordinary differential equations:

$$X''_x - \lambda X_x = 0$$

$$Y''_y + \lambda Y_y = 0$$

To find the general solution of this ODEs, the sign of λ should be known. For that, a case analysis can be performed on λ to find the solution.

Case 1: $\lambda = 0$

$$X''_x = 0 \quad \Rightarrow \quad X_x = Ax + B$$

$$u_{0,y} = 0, \quad \Rightarrow \quad X_0 = 0 \quad \Rightarrow \quad B = 0$$

$$u_{1,y} = 0, \quad \Rightarrow \quad X_1 = 0 \quad \Rightarrow \quad A = 0$$

Which implies that $X_x = 0$. Thus, $\lambda = 0$ is not a feasible solution.

Case 2: $\lambda = \alpha^2$

The system becomes:

$$X''_x - \alpha^2 X_x = 0 \quad \text{and} \quad Y''_y + \alpha^2 Y_y = 0$$

The general solutions are:

$$X_x = A\cosh(\alpha x) + B\sinh(\alpha x) \quad \text{and} \quad Y_y = C\cos(\alpha y) + D\sin(\alpha y)$$

This case leads to exponential behavior in X_x and oscillatory behavior in Y_y.

Applying boundary conditions:

$$u_{x,0} = 0 \quad \Rightarrow \quad Y_0 = 0 \quad \Rightarrow \quad C = 0$$

$$u_{0,y} = 0 \quad \Rightarrow \quad X_0 = 0 \quad \Rightarrow \quad A = 0$$

$$u_{1,y} = 0 \quad \Rightarrow \quad X_1 = B\sinh(\alpha) = 0 \quad \Rightarrow \quad \sinh(\alpha) = 0$$

But $\sinh(\alpha) = 0$ only when $\alpha = 0$, No solution for $\lambda = \alpha^2$.

Case 3: $\lambda = -\alpha^2$

Now the system becomes:

$$X_x'' + \alpha^2 X_x = 0 \quad \text{and} \quad Y_y'' - \alpha^2 Y_y = 0$$

The general solutions are:

$$X_x = A\cos(\alpha x) + B\sin(\alpha x) \quad \text{and} \quad Y_y = C\cosh(\alpha y) + D\sinh(\alpha y)$$

Applying boundary conditions:

$$u_{0,y} = 0 \quad \Rightarrow \quad X_0 = 0 \quad \Rightarrow \quad A = 0$$

$$u_{1,y} = 0 \quad \Rightarrow \quad X_1 = B\sin(\alpha) = 0 \quad \Rightarrow \quad \alpha = n\pi \quad (\text{for } n = 1, 2, 3, \dots)$$

We take the first harmonic: $\alpha = \pi$

$$u_{x,0} = 0 \quad \Rightarrow \quad Y_0 = 0 \quad \Rightarrow \quad C = 0$$

So the solution becomes:

$$u_{x,y} = BD\sin(\pi x)\sinh(\pi y)$$

Apply the last boundary condition:

$$u_{x,1} = \sin(\pi x) \Rightarrow BD\sinh(\pi) = 1 \Rightarrow BD = \frac{1}{\sinh(\pi)}$$

So the final solution is:

$$u_{x,y} = \sin(\pi x) \cdot \frac{\sinh(\pi y)}{\sinh(\pi)}$$

2

Numerical Method

2.1 Finite Difference Method

The finite difference method is an iterative method that approximates the second derivative of the function using finite difference formulas, and builds a mesh of points in the solution domain. Each time the iteration is repeated, the boundary conditions impose their effect on the central points. Starting from the central difference formula:

$$\frac{\partial^2 u_x}{\partial x^2} \approx \frac{u_{x+h} - 2u_x + u_{x-h}}{h^2}$$

A mesh or grid is created by dividing the (x,y) domain into (nx,ny) subdomains, creating nx*ny different points (i,j). The formula then becomes:

$$\frac{\partial^2 u_{j,i}}{\partial x^2} \approx \frac{u_{j,i+1} - 2u_{j,i} + u_{j,i-1}}{h^2}$$

After inserting the formula into the Laplace PDE, we obtain the iterative formula:

$$u_{j,i} = \frac{(\Delta y)^2 \left[u_{j,i+1} + u_{j,i-1}\right] + (\Delta x)^2 \left[u_{j+1,i} + u_{j-1,i}\right]}{2\left[(\Delta x)^2 + (\Delta y)^2\right]}$$

2.2 Implementation Using Matlab

The solution accuracy is related to the number of iterations performed on the mesh. The code below is the implementation of the numerical method in Matlab.

```
C = 1/(2*(dx^2 + dy^2));
for k = 1:1000
    for i = 2:ny
        for j = 2:nx
            N(i,j) = C * (dx^2 * (N(i+1,j) + N(i-1,j)) ...
                        + dy^2 * (N(i,j+1) + N(i,j-1)));
        end
    end
end
```

3

Results and Analysis

All results were normalized for consistent graphs; 50 subdomains were created along each axis to form the grid.

3.1 Analytical Solution

The analytical solution is plotted in a contour plot below.

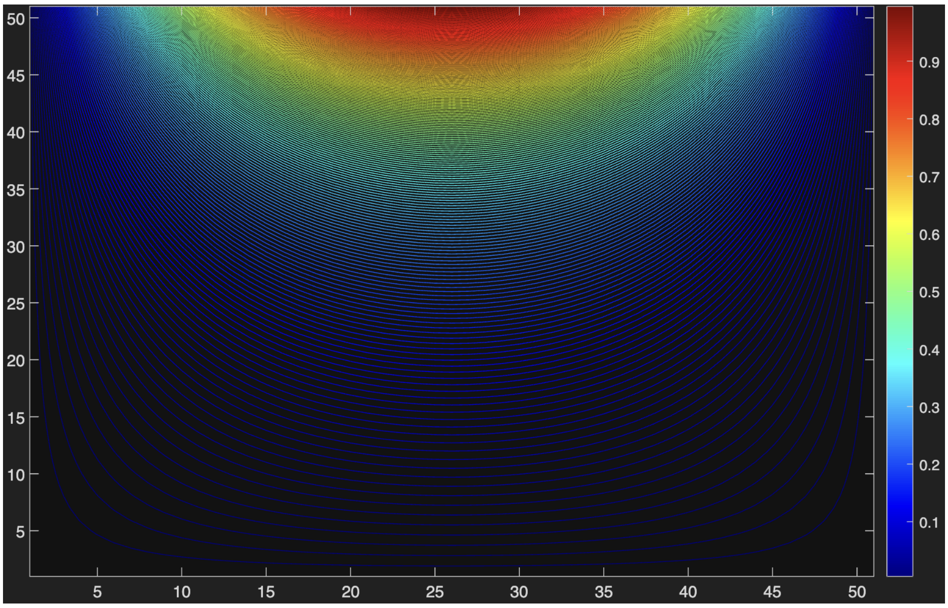

Figure 3.1: Analytical solution of the Laplace PDE.

3.2 Numerical Solution

The numerical solution is plotted in a contour plot below. 100 iterations were performed to obtain these results.

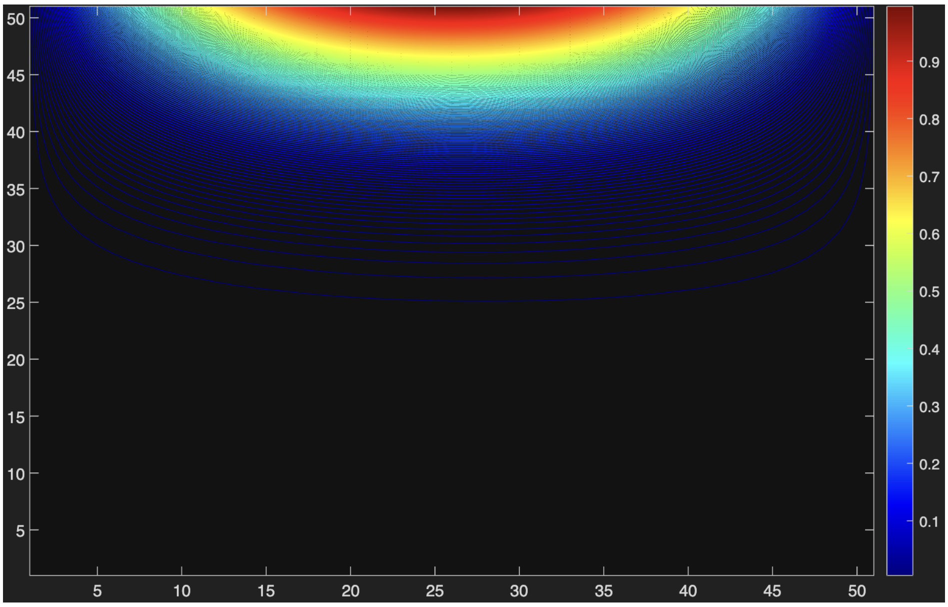

Figure 3.2: Numerical solution of the Laplace PDE with 100 iterations.

3.3 Error Analysis

The error is found by taking the absolute value of the difference between the analytical and the numerical solutions. It was then plotted in a normalized contour plot.

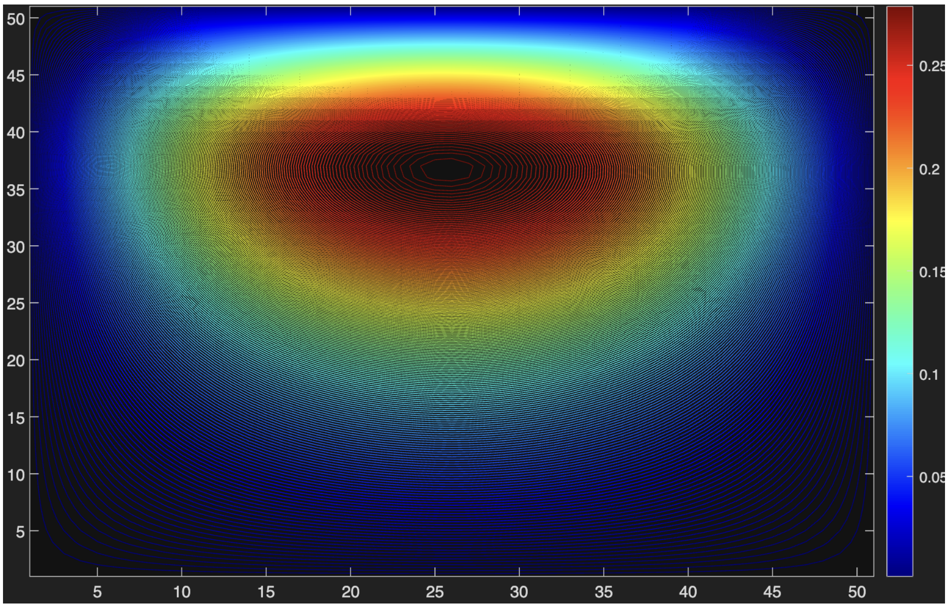

Figure 3.3: The error in a 100 iterations solution.

It is evident that the error is minimal at the boundaries and maximal in the middle, indicating that the data propagates from the boundary conditions towards the central points. A study of the effect of the iteration number, and the step size will enhance our understanding of the behavior of this iterative method.

Iteration Number Effect

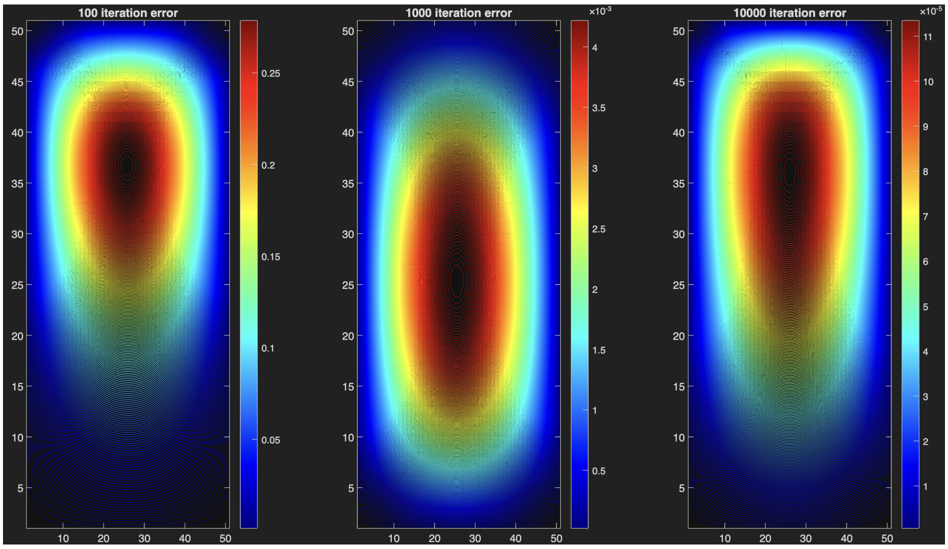

Figure 3.4: The error for different numbers of iterations.

Each time we increase the number of iterations by a factor of 10 (from left to right), the error decreases by a factor of 100, indicating that the error in the finite difference method is quadratic.

Step Size Effect

To compare between different step sizes, the number of iterations was kept constant at 500, Also the maximum error was recorded each time to study the worst case senario. The step size was varied by changing the number of subdomains multiple times to record enough data points for a data fitting.

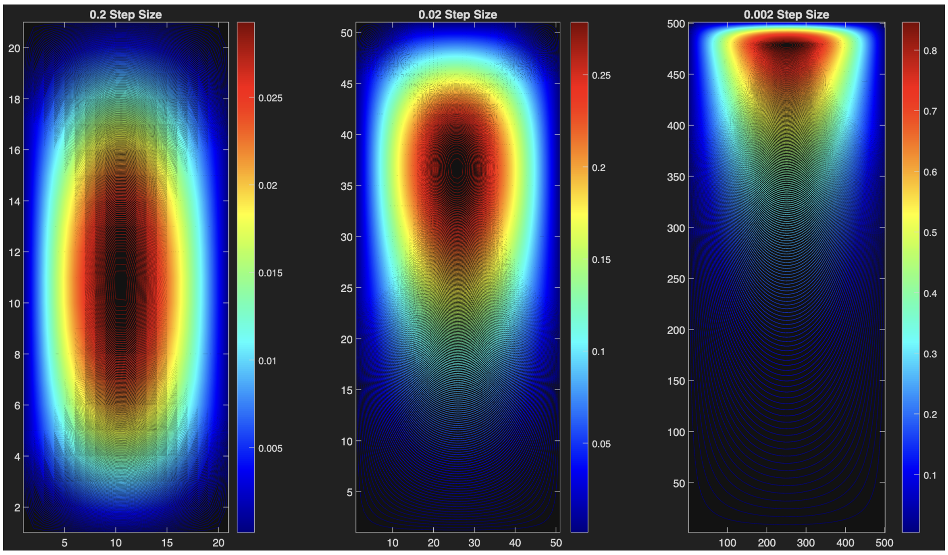

Figure 3.5: The error for different step sizes.

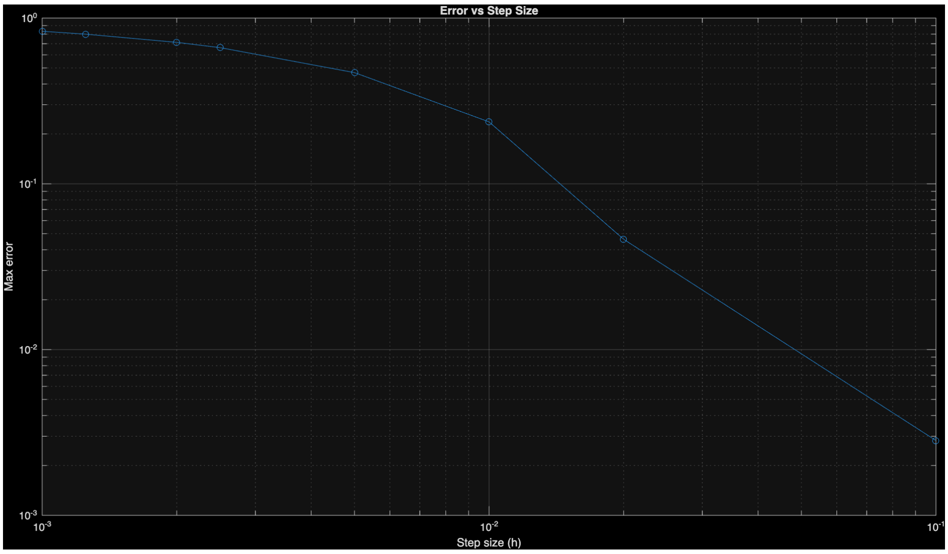

Figure 3.6: The relation Between step size and maximum error.

the error increases as the step size h become closer to 0. So it's clear that the error is inversely proportional to square of the step size.

3.4 Conclusion

The finite difference method can be a very accurate numerical method at a high number of iterations. With a quadratic error behaviour to the number of the iterations, and inversely to the step size.

$$\text{Error} = \mathcal{O}\left(\frac{k^2}{h^2}\right)$$

where k is the number of iterations and h is the step size.
It is important to decide the step size and the number of iteration based on the error that the problem can handle, in order to avoid unexpected fallouts. The error is minimal at the boundaries and increases as it moves towards the center of the domain, indicating that the data is built progressively from the boundary conditions to the center points.

Bibliography

[1] Richard L. Burden and J. Douglas Faires. *Numerical Analysis*. Cengage Learning, 9th edition, 2011.