TU Delft University of Technology
Mechanical Engineering Department
3mE Faculty

Report 1
Numerical Analysis For PDE's
WI4014TU

Academic Year 2019-2020 Q1-2

Student: **Cristopher Morales Ubal (4892356)**

Contents

1 Problem 1

a) Suppose that in $x = x_0$, $f(x)$ has a local maximum, then $\exists \delta > 0$ such that

$$f(x) \leq f(x_0) \quad , \quad \forall x \in (x_0 - \delta, x_0 + \delta)$$

So let's consider $x = x_0 + h$, with $h > 0$, then

$$\begin{aligned} & f(x_0 + h) \leq f(x_0) \quad , \quad |h| < \delta \\ \iff & f(x_0 + h) - f(x_0) \leq 0 \\ \implies & \frac{f(x_0 + h) - f(x_0)}{h} \leq 0 \text{ , because } h > 0 \end{aligned}$$

Taken $h \to 0$, we have

$$\begin{aligned} & \lim_{h \to 0} \left(\frac{f(x_0 + h) - f(x_0)}{h} \right) \leq 0 \\ \iff & \qquad f'(x_0) \leq 0 \quad (1) \end{aligned}$$

On the other hand, let consider $x = x_0 - h$,with $h > 0$, then as x_0 is a local maximum, we have

$$\begin{aligned} & f(x_0 - h) \leq f(x_0) \quad , \quad |h| < \delta \\ \iff & f(x_0 - h) - f(x_0) \leq 0 \\ \implies & \frac{f(x_0 + h) - f(x_0)}{-h} \geq 0 \text{ , because } -h > 0 \end{aligned}$$

Making $u = -h$, and notice that if $h \to 0$ then $u \to 0$. So taking $\lim_{h \to 0}$ we obtain

$$\begin{aligned} & \lim_{h \to 0} \left(\frac{f(x_0 - h) - f(x_0)}{-h} \right) \geq 0 \\ \iff & \lim_{u \to 0} \left(\frac{f(x_0 + u) - f(x_0)}{u} \right) \geq 0 \\ \iff & \qquad f'(x_0) \geq 0 \quad (2) \end{aligned}$$

Thus, from (1) and (2) we can conclude

$$f'(x_0) \leq 0 \quad \wedge \quad f'(x_0) \geq 0 \implies f'(x_0) = 0$$

■

b) Assume that in $x = x_0$, $f(x)$ has a local maximum at x_0, then $\exists \delta > 0$ such that

$$f(x) \leq f(x_0) \quad , \quad \forall x \in (x_0 - \delta, x_0 + \delta)$$

Thus, Expanding $f(x)$ in its Taylor Serie around $x = x_0$ we obtain

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \sum_{k=3}^{\infty} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k$$

if we reject the terms of order up than $k = 3$ (note that we can do that because we are at the surrounding of x_0, it means that $|x - x_0| < \delta$ with $\delta << 1$(very small)), we get

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2}(x - x_0)^2$$

using part a) we have:

$$\begin{aligned} & f(x) - f(x_0) \approx \underbrace{f'(x_0)}_{=0}(x - x_0) + \frac{f''(x_0)}{2}(x - x_0)^2 \\ \iff & f(x) - f(x_0) \approx \frac{f''(x_0)}{2}(x - x_0)^2 \end{aligned}$$

As at $x = x_0$ is a maximum, then:

$$\begin{aligned} f(x) \leq f(x_0) &\implies f(x) - f(x_0) \leq 0 \\ &\implies \frac{f''(x_0)}{2}(x-x_0)^2 \leq 0 \qquad (3) \end{aligned}$$

Using the fact that $u^2 > 0$, $\forall u \neq 0$, Taking $u = x - x_0$ we obtain

$$(x-x_0)^2 > 0$$

Therefore, in (3) we can divide by $(x-x_0)^2 > 0$ and multiply by 2. In this way, we can conclude that

$$f''(x_0) \leq 0$$

■

c) Let's suppose that $u(x,y)$ has a local maximum at(x_0,y_0), it means that $\exists\delta > 0$ such that

$$u(x,y) \leq u(x_0,y_0) \quad , \quad \forall(x,y) \ : \ \sqrt{(x-x_0)^2 + (y-y_0)^2} < \delta$$

Let's consider an arbitrary curve

$$c(t) = (x(t), y(t))$$

Such that in $t = t_0$ this passes by $(x,y) = (x_0,y_0)$, that is

$$c(t_0) = (x(t_0), y(t_0)) = (x_0,y_0)$$

Thus, we have the following function of one variable

$$g(t) = u(c(t)) = u(x(t), y(t))$$

note that in $t = t_0$, $g(t)$ reaches its local maximum, because

$$g(t_0) = u(c(t_0)) = u(x_0,y_0) \geq u(x(t),y(t)) = u(c(t)) = g(t)$$

$$\forall t \in (t-\delta_1, t+\delta_1) \text{ such that } \sqrt{(x(t)-x_0)^2 + (y(t)-y_0)^2} < \delta$$

So, using Fermat's Theorem we have in $t = t_0$

$$g'(t_0) = 0 \qquad (4)$$

On the other hand, using the chain rule, we get:

$$\begin{aligned} g'(t) &= \frac{\mathrm{d}}{\mathrm{dt}}(u(x(t),y(t))) \\ &= \frac{\partial u}{\partial x} \cdot \frac{\mathrm{d}x}{\mathrm{dt}} + \frac{\partial u}{\partial y} \cdot \frac{\mathrm{d}y}{\mathrm{dt}} \\ &= \left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\right) \cdot \left(\frac{\mathrm{d}x}{\mathrm{dt}}, \frac{\mathrm{d}y}{\mathrm{dt}}\right) \\ &= \nabla u(x,y) \cdot c'(t) \qquad (5) \end{aligned}$$

Evaluating (5) in $t = t_0$ and using (4) we obtain

$$0 = \nabla u(x_0,y_0) \cdot c'(t_0) \qquad (6)$$

How this must be matched for any curve $c(t)$ that passes through $(x,y) = (x_0,y_0)$ we can conclude that

$$\nabla u(x_0,y_0) \cdot c'(t_0) = 0 \iff \nabla u(x_0,y_0) = \mathbf{0}$$

■

The Directional Derivative $D_{\mathbf{V}}u(x,y)$ can be written as

$$D_{\mathbf{V}}u(x,y) = \nabla u(x,y)\cdot \mathbf{V} \qquad (7)$$

Where $\mathbf{V}$ is a unitary vector. Therefore, replacing $(x,y)=(x_0,y_0)$ in (7) and using the last result we obtain

$$D_{\mathbf{V}}u(x_0,y_0) = \nabla u(x_0,y_0)\cdot \mathbf{V} = \mathbf{0}\cdot\mathbf{V} = 0 \ \ , \ \forall\, \mathbf{V}\in\mathbb{R}^2$$

d) First of all, the Hessian Matrix of a Scalar field $f(\mathbf{x}) : \mathbb{R}^n \longrightarrow \mathbb{R}$, $H(\mathbf{x})$,is defined as:

$$H(\mathbf{x}) : \mathbb{R}^n \longrightarrow \mathbb{R}^n \ \ : \ \ H(\mathbf{x}) = \nabla\left(\nabla\left(f(\mathbf{x})\right)\right) = \begin{pmatrix} \dfrac{\partial}{\partial x_1}\left(\nabla f(\mathbf{x})\right) \\ \dfrac{\partial}{\partial x_2}\left(\nabla f(\mathbf{x})\right) \\ \vdots \\ \dfrac{\partial}{\partial x_n}\left(\nabla f(\mathbf{x})\right) \end{pmatrix}_{(n\times n)}$$

Thus, considering a scalar field in $\mathbb{R}^2$, $u(x,y)$, and applying of the definition the Hessian Matrix we obtain:

$$H(x,y) = \nabla\left(\nabla\left(u(x,y)\right)\right) = \begin{pmatrix} \dfrac{\partial}{\partial x}\left(\nabla u(x,y)\right) \\ \dfrac{\partial}{\partial y}\left(\nabla u(x,y)\right) \end{pmatrix}_{(2\times 2)} = \begin{pmatrix} \dfrac{\partial^2}{\partial x^2}\left(u(x,y)\right) & \dfrac{\partial^2}{\partial x\partial y}\left(u(x,y)\right) \\ \dfrac{\partial^2}{\partial y\partial x}\left(u(x,y)\right) & \dfrac{\partial^2}{\partial y^2}\left(u(x,y)\right) \end{pmatrix}$$

So, let's consider a unit vector $\mathbf{v}=(p,q)$, it means $\|\mathbf{v}\| = \sqrt{p^2+q^2} = 1$. Thus, computing $\mathbf{v}^T H\mathbf{v}$ we have:

$$\begin{aligned}
\mathbf{v}^T H(x,y)\mathbf{v} &= (p,q)\begin{pmatrix} \dfrac{\partial^2 u}{\partial x^2} & \dfrac{\partial^2 u}{\partial x\partial y} \\ \dfrac{\partial^2 u}{\partial y\partial x} & \dfrac{\partial^2 u}{\partial y^2} \end{pmatrix}\begin{pmatrix} p \\ q \end{pmatrix} \\
&= (p,q)\begin{pmatrix} p\dfrac{\partial^2 u}{\partial x^2} + q\dfrac{\partial^2 u}{\partial x\partial y} \\ p\dfrac{\partial^2 u}{\partial y\partial x} + q\dfrac{\partial^2 u}{\partial y^2} \end{pmatrix} \\
&= p\left(p\frac{\partial^2 u}{\partial x^2} + q\frac{\partial^2 u}{\partial x\partial y}\right) + q\left(p\frac{\partial^2 u}{\partial y\partial x} + q\frac{\partial^2 u}{\partial y^2}\right) \\
&= p\left(\frac{\partial}{\partial x}\left(p\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial x}\left(q\frac{\partial u}{\partial y}\right)\right) + q\left(\frac{\partial}{\partial y}\left(p\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(q\frac{\partial u}{\partial y}\right)\right) \\
&= \frac{\partial}{\partial x}\left(p\frac{\partial u}{\partial x} + q\frac{\partial u}{\partial y}\right)\cdot p + \frac{\partial}{\partial y}\left(p\frac{\partial u}{\partial x} + q\frac{\partial u}{\partial y}\right)\cdot q \\
&= \nabla\left(p\frac{\partial u}{\partial x} + q\frac{\partial u}{\partial y}\right)\cdot(p,q) \\
&= \nabla\left(\frac{\partial u}{\partial x}\cdot p + \frac{\partial u}{\partial y}\cdot q\right)\cdot(p,q) \\
&= \nabla\left(\left(\frac{\partial u}{\partial x},\frac{\partial u}{\partial y}\right)\cdot(p,q)\right)\cdot(p,q) \\
&= \nabla\left(\nabla u(x,y)\cdot(p,q)\right)\cdot(p,q) \\
&= \nabla\left(\nabla u(x,y)\cdot\mathbf{v}\right)\cdot\mathbf{v} \\
&= \nabla\left(D_{\mathbf{v}}u\right)\cdot\mathbf{v} \\
&= D_{\mathbf{v}}\left(D_{\mathbf{v}}u\right)
\end{aligned}$$

$\therefore$

$$\mathbf{v}^T H(x,y)\mathbf{v} = D_{\mathbf{v}}\left(D_{\mathbf{v}}u\right)$$

■

e) Let's consider the curve

$$x = x_0 + pt \quad \wedge \quad y(t) = y_0 + qt$$

We can consider the function:

$$f(t) = u\left(x(t), y(t)\right)$$

At $t = 0$, we have

$$f(0) = u\left(x(0), y(0)\right) = u\left(x_0, y_0\right)$$

and $u\left(x_0, y_0\right)$ is a local maximum value of $u(x, y)$, so in $t = 0$ the function $f(t)$ gets a maximum. Hence using part b) and d) we have:

$$\begin{aligned} f''(0) \leq 0 &\iff \frac{\mathrm{d}}{\mathrm{dt}}\left(\frac{\mathrm{d}}{\mathrm{dt}}\left(u(x(t), y(t))\right)\right)\bigg|_{t=0} \leq 0 \\ &\iff D_{\mathbf{v}}\left(D_{\mathbf{v}} u\right)\big|_{t=0} \leq 0 \\ &\iff \mathbf{v}^T H(x, y)\mathbf{v} \leq 0 \end{aligned}$$

where $\mathbf{v}^T = \left.\dfrac{(x(t), y(t))}{\|(x(t), y(t))\|}\right|_{t=0} = \dfrac{(x_0, y_0)}{\|(x_0, y_0)\|}$. In this way, we obtain:

$$\frac{1}{\|(x_0, y_0)\|^2} \cdot \begin{pmatrix} x_0 & y_0 \end{pmatrix} \cdot H\left(x_0, y_0\right) \cdot \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \leq 0 \iff \begin{pmatrix} x_0 & y_0 \end{pmatrix} \cdot H\left(x_0, y_0\right) \cdot \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \leq 0$$

Therefore $H(x_0, y_0)$ is negative semi-definite.

■

2 Problem 2

2.1 building 1D Grid

First of all, we create the 1D Uniform grid Ω_h in an arbitrary interval $[a, b]$ (which length $L = b - a$) given the step size h. At given step size h, we calculate the number of grid points N as follow:

$$h = \frac{b-a}{N-1} \iff N = \frac{b-a}{h} + 1 \tag{2.1}$$

Thus, the grid points are given by:

$$x_i = a + i \cdot h \ \ , \ i = 0, 1, \ldots N-1 \tag{2.2}$$

In this way, applying (2.1) and (2.2) to the interval $[a, b] = [0, 1]$, which length $L = 1$, we obtain the 1D Uniform grid Ω_h:

$$\Omega = \{x_i = ih \ , \ i = 0, 1, \ldots, N-1\} \text{ with } N = \frac{1}{h} + 1 \tag{2.3}$$

2.2 Discretization of 1D Laplace Operator

Using *central difference approximation* and zero Diriclet Boundary Conditions, we discretize the operator $\mathcal{L} = -\dfrac{d^2}{dx^2}$ in the 1D grid (2.3) as follow:

$$\mathcal{L}u = -\frac{d^2}{dx^2}(u) \approx -\mathcal{D}^{(2)}_{xx} u_i = -\frac{(\mathcal{D}^+_x u_i - \mathcal{D}^-_x u_i)}{h} = -\frac{\left(\dfrac{(u_{i+1} - u_i)}{h} - \dfrac{(u_i - u_{i-1})}{h}\right)}{h} = \frac{-u_{i+1} + 2u_i - u_{i-1}}{h^2} \tag{2.4}$$

How we are considering Dirichlet boundary conditions and noting that $x_0 = 0$ and $x_{N-1} = 1$, we obtain

$$u_0 = u(x_0) = u(0) = 0 \tag{2.5}$$
$$u_{N-1} = u(x_{N-1}) = u(1) = 0 \tag{2.6}$$

Replacing (2.5) and (2.6) in (2.4) we have:

$$\begin{aligned}
i = 1: \ \mathcal{L}u_1 &= \frac{-u_2 + 2u_1}{h^2} \\
i = 2: \ \mathcal{L}u_2 &= \frac{-u_3 + 2u_2 - u_1}{h^2} \\
&\vdots \\
i = N-3: \ \mathcal{L}u_{N-3} &= \frac{-u_{N-2} + 2u_{N-3} - u_{N-4}}{h^2} \\
i = N-2: \ \mathcal{L}u_2 &= \frac{2u_{N-2} - u_{N-3}}{h^2}
\end{aligned}$$

Writing as matrices we obtain the discretización of the operator $\mathcal{L}u$:

$$\mathcal{L}u = L\mathbf{u} = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 & 0 \\ -1 & 2 & -1 & \cdots & 0 & 0 \\ 0 & -1 & 2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2 & -1 \\ 0 & 0 & 0 & \cdots & -1 & 2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-3} \\ u_{N-2} \end{pmatrix} \tag{2.7}$$

where L is the *Laplacian Matrix*.

2.3 Results

Considering the step $h = 0.1$ we calculate the number of points using (2.3), from we obtain $N = 11$. Thus, using the command `np.linspace()` we construct Ω_h in python which is shown in the following figure:

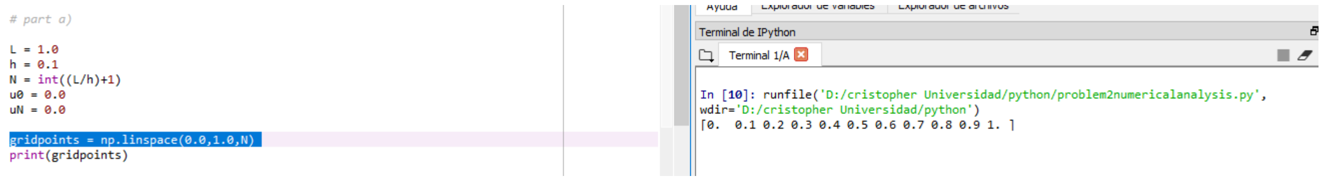

Figure 2.1: Grid Points 1D Grid

Using the command `np.diag()` we create the Laplace matrix L, and we show in the figure below the first, the second and last row of this matrix:

Figure 2.2: First, second and Last rows of L Matrix.

Likewise, using the command `np.spy()`, we can visualize the structure of the matrix L as shown in the figure below:

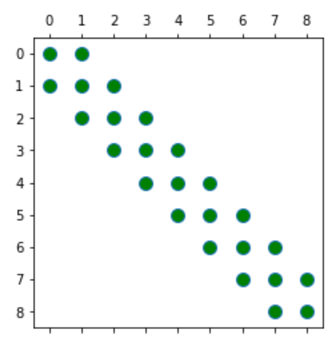

Figure 2.3: Structure of Laplace Matrix.

On the other Hand, we calculate the eigenvalues of the operator $\mathcal{L}$ given by the expression:

$$\widehat{\lambda}_i = \left(\frac{\pi i}{L}\right)^2 \quad , \; i = 1, 2, \ldots, N-2 \tag{2.8}$$

Similarly, we compute the eigenvalues of L given by

$$\lambda_i = \frac{4}{h^2} \sin^2\left(\frac{\pi i h}{2L}\right) \quad , \; i = 1, 2, \ldots, N-2 \tag{2.9}$$

Therefore, computing those expressions on python we obtain:

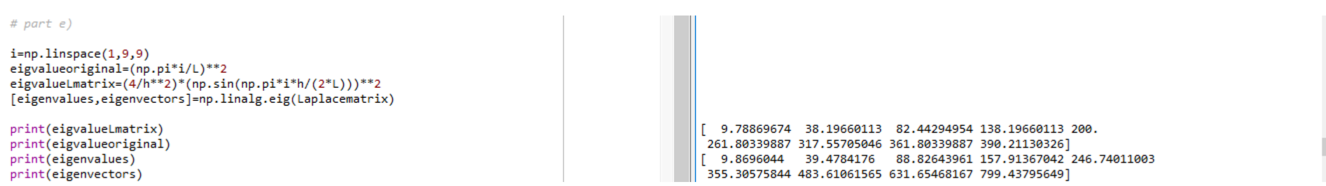

Figure 2.4: Eigenvalues of Laplace Matrix and Operator $\mathcal{L}$.

Additionally, using the command `np.linalg.eig()` we compute the eigenvalues and eigenvector of the matrix L numerically. In this way, we compare the eigenvalues of the operator $\mathcal{L}$ (2.8) with the eigenvalues of L calculated numerically, this is shown in the table below:

Table 2.1: Eigenvalues of Operator $\mathcal{L}$ vs Eigenvalues of matrix L numerically

i	$\widehat{\lambda_i}$	λ_i
1	9.8696044	9.78869674
2	39.4784176	38.19660113
3	88.82643961	82.44294954
4	157.91367042	138.19660113
5	246.74011003	200.
6	355.30575844	390.21130326
7	483.61061565	361.80339887
8	799.43795649	261.80339887
9	355.30575844	317.557050462

In order to compare in a better way, we use the command `np.plot()` in order to plot the eigenvalues in the complex plane. Hence, we obtain

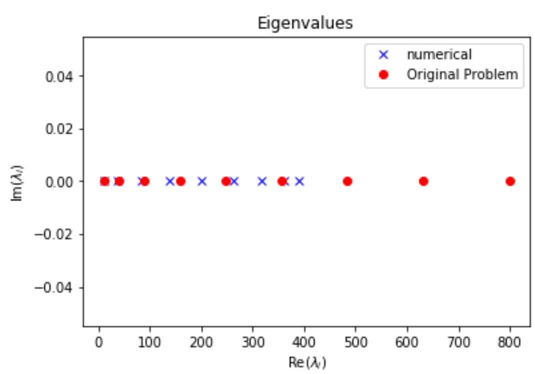

Figure 2.5: Eigenvalues of Laplace Matrix obtained numerically and Operator $\mathcal{L}$.

2.3.1 Discussion

From the results we can observe:

1. Using the expression (3.3) lead us to built a uniform grid as we can observe in figure 2.1. Additionally, from figures 2.2 and 3.3 we can visualize that the Laplacian Matrix has effectively a form of a tridiagonal matrix and , at the same time, conclude that the Laplacian Matrix is symmetric .

2. Comparing the eigenvalues obtained numerically for the Laplacian matrix and the eigenvalues of the operator $\mathcal{L}$ (table 2.1 and figure 2.5), we can deduce that the eigenvalues in both cases have only real part, it means that the imaginary part for them is zero.This result is a consequences of the symmetry of the Laplacian matrix. Nevertheless, the eigenvalues numerically and the eigenvalues of the original problem differs (figure 2.5), this can be due to the step size, if the step size becomes smaller the eingenvalues calculated numerically should be closer to the eigenvalues of the original problem.

3 Problem 3

3.1 Computing Analytical Solution 1D Poisson Equation

First of all, we calculate the Analytical solution of the *1D Poisson Equation* (show below) for two different source functions: $f_1(x) = 1$ and $f_2(x) = e^x$.

$$-\frac{\mathrm{d}^2 u}{\mathrm{dx}^2} = f_i \ , \ x \in (0,1) \ ,$$
$$u(0) = 1 \ , \ u(1) = 2 \quad \text{(B.C.)}$$

Thus, integranting twice for each case we have:

- $f_1(x) = 1$:

$$-\frac{\mathrm{d}^2 u}{\mathrm{dx}^2} = 1 \implies \frac{\mathrm{d}u}{\mathrm{dx}} = -\int dx + C_1 \implies u(x) = -\int x dx + \int C_1 dx + C_2 = -\frac{1}{2}x^2 + C_1 x + C_2$$

  Imposing boundary conditions (B.C.) we obtain:

$$\begin{matrix} u(0) = 1 \\ u(1) = 2 \end{matrix} \iff \begin{matrix} C_2 = 1 \\ -\frac{1}{2} + C_1 + C_2 = 2 \end{matrix} \implies C_1 = \frac{3}{2} \ \wedge \ C_2 = 1$$

  Hence, the exact solution is

$$u_1(x) = -\frac{1}{2}x^2 + \frac{3}{2}x + 1 \tag{3.1}$$

- $f_2(x) = e^x$: Proceeding of analogous way as we did before we obtain the following exact solution:

$$u_2(x) = -e^x + ex + 2 \tag{3.2}$$

3.2 Discretizing 1D-Poisson Equation

3.2.1 Uniform Grid

In order to solve Numerically the *1D Poisson Equation*, we build the *1D uniform grid* in the interval $[a, b]$ as follow:

$$\Omega = \{x_i = a + ih \ , \ i = 0, 1, \ldots, N\} \tag{3.3}$$

where the step h and the Number of Intervals N (note that there are $N + 1$ grid points) are related as:

$$h = \frac{b-a}{N} \iff N = \frac{b-a}{h} \tag{3.4}$$

Applying those relations to the interval $[a, b] = [0, 1]$ we build the following grid point:

$$\Omega = \{x_i = ih \ , \ i = 0, 1, \ldots, N\} \ , \ N = \frac{1}{h} \tag{3.5}$$

Likewise, we use FDM (Finite Difference Method) to approximate the second derivative of $u(x)$ in the interior nodes as follow:

$$u''(x_i) \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} \ , \ \forall i = 1, 2, \ldots, N-1 \tag{3.6}$$

Thus, replacing the second FD derivative in the Poisson Equation we obtain:

$$-\frac{(u_{i+1} - 2u_i + u_{i-1})}{h^2} = f_i \ , \ \forall i = 1, 2, \ldots, N-1 \tag{3.7}$$

where $u_i = u(x_i)$ and $f_i = f(x_i)$.

3.2.2 Non Uniform Grid

In order to compare how much the construction of the grid affects the results, we build a 1D *Non Uniform Grid* join the two first point of the 1D Uniform Grid created before (3.3). It means, the grid is given by:

$$\Omega = \{x_0 = a, x_1 = a + 2h, x_i = a + (i+1)h \ , \ i = 2, 3, \ldots, N-1\} \tag{3.8}$$

Likewise, we can notice that the step grid h and the Number of Grid Points N is given by the relation (3.4). Thus, Applying this to the interval $[a, b] = [0, 1]$ we obtain the following 1D Non Uniform Grid:

$$\Omega = \{x_0 = 0, x_1 = 2h, x_i = (i+1)h \ , \ i = 2, 3, \ldots, N-1\} \ \wedge \ N = \frac{1}{h} \tag{3.9}$$

Likewise, proceeding of analogous way, we use FDM (Finite Difference Method) to approximate the second derivative of $u(x)$, but we can observe that the approximation of the second derivative of u_i in $i = 1$ changes its form, This is because the width between x_0 and x_1 is now $2h$. Therefore, the second derivative in x_1 must be recompute as follow:

$$u''(x_1) \approx \frac{D_x^+ u_1 - D_x^{-1} u_1}{2h} = \frac{\frac{u_2 - u_1}{h} - \frac{u_1 - u_0}{2h}}{2h} = \frac{2u_2 - 3u_1 + u_0}{4h^2} \tag{3.10}$$

And, for the rest of inner points, the expression (3.6) is still valid because the distance between two consecutive nodes is the same, h.
Thus, replacing the second FD derivative in the Poisson Equation we obtain:

$$-\frac{(2u_2 - 3u_1 + u_0)}{4h^2}) = f_1 \tag{3.11}$$

$$-\frac{(u_{i+1} - 2u_i + u_{i-1})}{h^2} = f_i \ , \ \forall i = 2, 3, \ldots, N-2 \tag{3.12}$$

where $u_i = u(x_i)$ and $f_i = f(x_i)$.

3.3 Constructing the Linear Algebraic Problem

3.3.1 Uniform Grid

In order to derive the Linear Algebraic problem, we can note that the boundary values correspond to $u_0 = u(x_0) = u(0) = 1$ and $u_N = u(x_N) = u(1) = 2$. Therefore, in (3.7) we have

$$\begin{aligned}
-\frac{(u_2 - 2u_1 + u_0)}{h^2} &= f_1 \\
-\frac{(u_3 - 2u_2 + u_1)}{h^2} &= f_2 \\
\vdots \quad & \quad \vdots \\
-\frac{(u_{N-1} - 2u_{N-2} + u_{N-3})}{h^2} &= f_{N-2} \\
-\frac{(u_N - 2u_{N-1} + u_{N-2})}{h^2} &= f_{N-1}
\end{aligned}$$

Rearranging these equations, we obtain

$$\frac{1}{h^2}(2u_1 - u_2) = f_1 + \frac{1}{h^2} \cdot u_0 \tag{3.13}$$

$$\frac{1}{h^2}(-u_1 + 2u_2 - u_3) = f_2 \tag{3.14}$$

$$\vdots \qquad \vdots \tag{3.15}$$

$$\frac{1}{h^2}(-u_{N-3} + 2u_{N-2} - u_{N-1}) = f_{N-2} \tag{3.16}$$

$$\frac{1}{h^2}(-u_{N-2} + 2u_{N-1}) = f_{N-1} + \frac{1}{h^2} u_N \tag{3.17}$$

which lead us to the following Linear Algebraic Problem:

$$L\mathbf{u} = \mathbf{f} \tag{3.18}$$

where

$$L = \frac{1}{h^2}\begin{pmatrix} 2 & -1 & 0 & \dots & 0 & 0 \\ -1 & 2 & -1 & \dots & 0 & 0 \\ 0 & -1 & 2 & \dots & 0 & 0 \\ & \vdots & & \ddots & \vdots & \\ 0 & 0 & 0 & \dots & 2 & -1 \\ 0 & 0 & 0 & \dots & -1 & 2 \end{pmatrix}_{(N-1)\times(N-1)}, \ \mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-2} \\ u_{N-1} \end{pmatrix}_{(N-1)\times 1}, \ \mathbf{f} = \begin{pmatrix} f_1 + \frac{1}{h^2}\cdot u_0 \\ f_2 \\ f_3 \\ \vdots \\ f_{N-2} \\ f_{N-1} + \frac{1}{h^2}\cdot u_N \end{pmatrix}_{(N-1)\times 1} \tag{3.19}$$

3.3.2 Non Uniform Grid

Following the same procedure as we did in section (3.3.1), Nevertheless, we take into account that in this case the boundary values corresponds to $u_0 = u(x_0) = u(0)$ and $u_{N-1} = u(x_{N-1}) = u(1)$. Hence, we have in (3.11) and (3.12)

$$\begin{aligned} -\frac{(2u_2 - 3u_1 + u_0)}{4h^2} &= f_1 \\ -\frac{(u_3 - 2u_2 + u_1)}{h^2} &= f_2 \\ \vdots \quad & \quad \vdots \\ -\frac{(u_{N-2} - 2u_{N-3} + u_{N-4})}{h^2} &= f_{N-3} \\ -\frac{(u_{N-1} - 2u_{N-2} + u_{N-3})}{h^2} &= f_{N-2} \end{aligned}$$

Rearranging these equations, we get:

$$\frac{1}{h^2}(3u_1 - 2u_2) = 4f_1 + \frac{1}{h^2}\cdot u_0 \tag{3.20}$$

$$\frac{1}{h^2}(-u_1 + 2u_2 - u_3) = f_2 \tag{3.21}$$

$$\vdots \qquad \vdots \tag{3.22}$$

$$\frac{1}{h^2}(-u_{N-4} + 2u_{N-3} - u_{N-2}) = f_{N-3} \tag{3.23}$$

$$\frac{1}{h^2}(-u_{N-3} + 2u_{N-2}) = f_{N-2} + \frac{1}{h^2}u_{N-1} \tag{3.24}$$

which lead us to the following Linear Algebraic Problem:

$$L\mathbf{u} = \mathbf{f} \tag{3.25}$$

where

$$L = \frac{1}{h^2}\begin{pmatrix} 3 & -2 & 0 & \dots & 0 & 0 \\ -1 & 2 & -1 & \dots & 0 & 0 \\ 0 & -1 & 2 & \dots & 0 & 0 \\ & \vdots & & \ddots & \vdots & \\ 0 & 0 & 0 & \dots & 2 & -1 \\ 0 & 0 & 0 & \dots & -1 & 2 \end{pmatrix}_{(N-2)\times(N-2)}, \ \mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-3} \\ u_{N-2} \end{pmatrix}_{(N-2)\times 1}, \ \mathbf{f} = \begin{pmatrix} 4f_1 + \frac{1}{h^2}\cdot u_0 \\ f_2 \\ f_3 \\ \vdots \\ f_{N-2} \\ f_{N-1} + \frac{1}{h^2}\cdot u_N \end{pmatrix}_{(N-2)\times 1} \tag{3.26}$$

3.4 Results

3.4.1 Uniform Grid

Considering the step size $h = 0.2$, we compute the matrix L and $\mathbf{f}$ for the two sources functions $f_1(x) = 1$ and $f_2(x) = e^x$ obtaining the following matrices:

- $f_1(x) = 1$:

$$L = \begin{pmatrix} 50 & -25 & 0 & 0 \\ -25 & 50 & -25 & 0 \\ 0 & -25 & 50 & -25 \\ 0 & 0 & -25 & 50 \end{pmatrix} \quad , \quad \mathbf{f} = \begin{pmatrix} 26. \\ 1. \\ 1. \\ 51. \end{pmatrix} \tag{3.27}$$

- $f_2(x) = e^x$:

$$L = \begin{pmatrix} 50 & -25 & 0 & 0 \\ -25 & 50 & -25 & 0 \\ 0 & -25 & 50 & -25 \\ 0 & 0 & -25 & 50 \end{pmatrix} \quad , \quad \mathbf{f} = \begin{pmatrix} 26. \\ 1.4918247 \\ 1.8221188 \\ 51.8221188 \end{pmatrix} \tag{3.28}$$

Using the command `np.linalg.solve`, we find the numerical solution for the cases mentioned before solving the Linear Algebraic Problem (3.18) . Likewise, we compare and plot the exact solution with the numerical solution, obtaining the following results:

Table 3.1: Exact and numerical solution for $f_1(x) = 1$.

Grid Point x_i	Numerical Solution $u_1(x_i)$	Exact Solution $u_1(x_i)$
0	1.	1
0.2	1.28	1.28
0.4	1.52	1.52
0.6	1.72	1.72
0.8	1.88	1.88
1	2	2

Table 3.2: Exact and numerical solution for $f_2(x) = e^x$.

Grid Point x_i	Numerical Solution $u_2(x_i)$	Exact Solution $u_2(x_i)$
0	1.	1.
0.2	1.31153464	1.32225361
0.4	1.58306929	1.59548803
0.6	1.79493094	1.8088503
0.8	1.93390785	1.94908453
1	2.	2

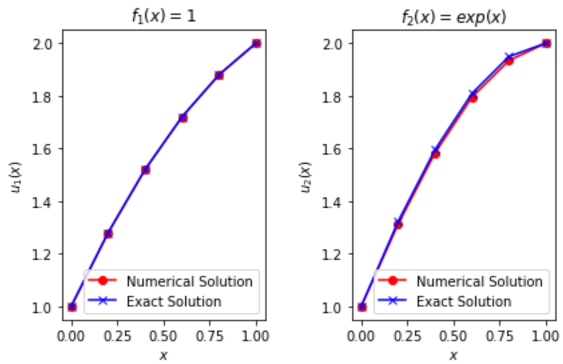

Figure 3.1: Numerical and Analytical Solutions for two different source functions.

Adittionally, using the command `np.linalg.norm()`, we compute the global error for each case obtaining the following results:

$$\text{Case 1}: \ \|\mathbf{e}\|_2 = 3.8459253727671276e - 16 = 0 \tag{3.29}$$

$$\text{Case 2}: \ \|\mathbf{e}\|_2 = 0.02632872225564738 \tag{3.30}$$

Where, we have used the norm 2, $\|\cdot\|_2$, defined by:

$$\|A\|_2 = \sqrt{\lambda_{max}(A^T A)}$$

Note that $3.8459253727671276e - 16 = 0$ because it has the maximum precision of the program, which is 10^-16.

Finally, in order to study the global error behaviour with respect to the step size h, we redefine the uniform grid changing the step size as $h = \dfrac{2}{2^k}$ with $k = 0, 1, 2, 3, 4$. For this purpose, we recompute the numerical solution of $u_2(x)$ for this 5 different steps size and calculate, in each case, the global error. Subsequently, we obtain the following figure:

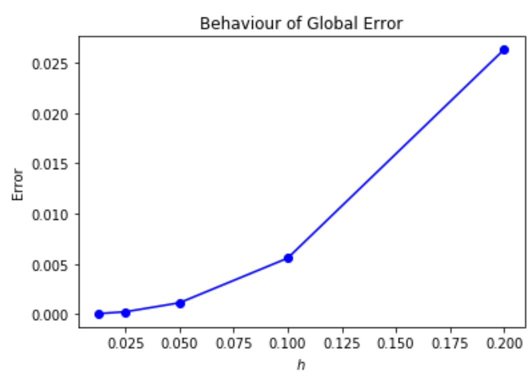

Figure 3.2: Global Error vs Step Size

Moreover, using the command `scipy.optimize.curve fit()`, we fit the global error data obtained before with the curve Ch^α. As a result, we get the following values for C and α:

$$C = 0.97810502 \quad \wedge \quad \alpha = 2.24604479 \tag{3.31}$$

3.4.2 Non Uniform Grid

Considering the step size $h = 0.2$, we compute the matrix L and $\mathbf{f}$ given by (3.26) for the two sources functions $f_1(x) = 1$ and $f_2(x) = e^x$ obtaining the following matrices:

- $f_1(x) = 1$:

$$L = \begin{pmatrix} 75 & -50 & 0 \\ -25 & 50 & -25 \\ 0 & -25 & 50 \end{pmatrix} \quad , \quad \mathbf{f} = \begin{pmatrix} 29. \\ 1. \\ 51. \end{pmatrix} \tag{3.32}$$

- $f_2(x) = e^x$:

$$L = \begin{pmatrix} 75 & -50 & 0 \\ -25 & 50 & -25 \\ 0 & -25 & 50 \end{pmatrix} \quad , \quad \mathbf{f} = \begin{pmatrix} 30.96729879 \\ 1.8221188 \\ 52.22554093 \end{pmatrix} \tag{3.33}$$

Using the command `np.linalg.solve`, we find the numerical solution for the cases $f_1(x) = 1$ and $f_2(x) = e^x$ solving the Linear Algebraic Problem (3.25) when the step size is $h = 0.2$. Likewise, we compare and plot the exact solution with the numerical solution, obtaining the following results:

Table 3.3: Exact and numerical solution for $f_1(x) = 1$.

Grid Point x_i	Numerical Solution $u_1(x_i)$	Exact Solution $u_1(x_i)$
0	1.	1
0.4	1.544	1.52
0.6	1.736	1.72
0.8	1.888	1.88
1	2	2

Table 3.4: Exact and Numerical solution for $f_2(x) = e^x$.

Grid Point x_i	Numerical Solution $u_2(x_i)$	Exact Solution $u_2(x_i)$
0	1.	1.
0.4	1.63713163	1.59548803
0.6	1.83635147	1.8088503
0.8	1.96268655	1.94908453
1	2.	2

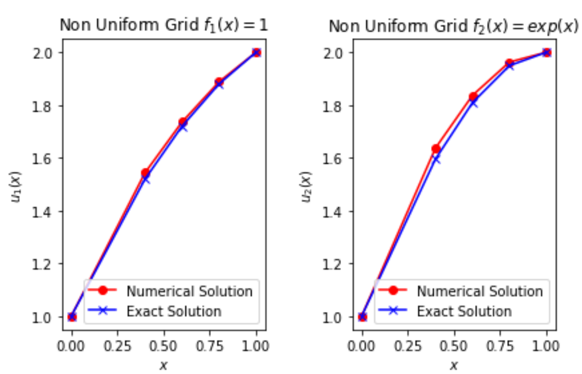

Figure 3.3: Numerical and Analytical Solutions for two different source functions.

Adittionally, using the command `np.linalg.norm()`, we compute the global error for each case obtaining the following results:

$$\text{Case 1}: \quad \|\mathbf{e}\|_2 = 0.029933259094191263 \tag{3.34}$$
$$\text{Case 2}: \quad \|\mathbf{e}\|_2 = 0.051725410002495996 \tag{3.35}$$

Where, we have used the norm 2 as we did in the case of the uniform grid.
Repeating the same procedure as we did before for the uniform grid (section 3.4.1) , we study the global error behaviour with respect to the step size h. As a result of this, we obtain the following plot:

Figure 3.4: Global Error vs Step Size

Likewise, using the command `scipy.optimize.curve fit()`, we fit the global error data obtained before with the curve Ch^{α}. As a result, we get the following values for C and α:

$$C = 0.55795288 \quad \wedge \quad \alpha = 1.47734935 \tag{3.36}$$

3.5 Discussion

From the results we can observe that:

1. From table 3.1. we observe when the source function is $f_1(x) = 1$ that the numerical solution and the analytic solution coincide in every grid point, this can also be observed in figure 3.1. This is because when we use the second FD derivative, the error introduced in this method can be calculated using Taylor's expansion, obtaining the following expression:
$$\left|D_{xx}^{(2)} u_i - u''(x_i)\right| = \frac{h^2}{12}\left|u^{(4)}(\xi_i)\right| \quad , \; \xi_i \in (x_{i-1}, x_{i+1}) \tag{3.37}$$
As the Poisson equation is
$$-\frac{d^2u}{dx^2} = 1 \quad \Longrightarrow \quad \frac{d^4u}{dx^4} = -\frac{d^2}{dx^2}(1) = 0 \tag{3.38}$$
Therefore replacing (3.36) in (3.35) we can conclude that
$$\left|D_{xx}^{(2)} u_i - u''(x_i)\right| = 0 \tag{3.39}$$
Thus, the numerical solution and the exact solution are the same for every node. This is also confirmed for the error obtained using norm 2 in (3.29) which is the resolution of the computational program. Furthermore, as the second derivative of any polynomial function of grade less or equal than 1 is zero, we can conclude that the FD method provides the exact approximation of grid values for any polynomial function with grade less or equal than 1.

2. From table 3.2 and figure 3.1, we can observe that the numerical solution for the second source function $f_2(x) = e^x$ differs respect to the exact solution. This can also explain by the expression (3.35) because for this function we have
$$-\frac{d^2u}{dx^2} = e^x \quad \Longrightarrow \quad \frac{d^4u}{dx^4} = -\frac{d^2}{x^2}(e^x) = e^x \neq 0 \tag{3.40}$$
This is reflected in the error obtained using norm 2 in (3.30).

3. When we study the behaviour of global error respect to the step size, figure 3.2, we can see that the error decreases when the step size decreases which it is the expected result. Moreover, when we fixed the global error data with the curve Ch^{α} we obtained the values given by (3.31), the exponent value is $\alpha = 2.24604479$ which is close to the exponent value expected which is 2. This confirms that the FD method has the second-order approximation.

Notwithstanding, we must notice that the values obtained in (3.31) depend on the norm used. This is because if we use, for instance the infinity norm defined by

$$|e\|_\infty = \max_{1\leq i\leq n-1} |u_i - \widehat{u_i}|$$

where u_i is the exact solution and $\widehat{u_i}$ is the numerical approximation we will likely get other values of C and α, however they must be close to those obtained in this report.

4. When we repeated the calculations for a Non Uniform Grid, we can observe that in this case from table 3.3 and figure 3.3 that FD method does not provide the exac solution for the first source function $f_1(x) = 1$. Similarly, for the second source function $f_2(x) = e^x$ the error increases, however, in order to do a fair comparison, we should have two grids with the same number of points: one uniform and the second one Non uniform.

5. Additionally, when we study the behaviour of global error respect to the step size, figure 3.4, we can observe that the form of the curve is not longer parabolic, which is confirmed with the value of α obtained which is 1.47734935. This leads us to conclude that when the grid is not uniform, the method is no longer of second order.

4 Problem 4

4.1 Discretizacion of the problem

First, in order to solve the 2D Poisson Equation, we must discretise the region Ω. For this purpose, we build the 2D discretization using 1D grid for x in the interval $[a, b]$ and y in the interval $[c, d]$, so we have:

$$x_i = a + ih_x, \quad i = 0, 1, 2, ..., N_x \tag{4.1}$$

where N_x and h_x are related as:

$$N_x = \frac{b-a}{h_x} \tag{4.2}$$

Proceeding in the same way, we obtain for y:

$$y_j = c + jh_y, \quad j = 0, 1, 2, ..., N_y \tag{4.3}$$

where N_y and h_y are related as:

$$N_y = \frac{d-c}{h_y} \tag{4.4}$$

Thus, considering a uniform grid $h_x = h_y = h$ and the interval for $x: \ [a, b] = [0, 2]$ and $y: \ [c, d] = [0, 1]$ we can create the 2D uniform Grid given by:

$$\Omega = \{(ih, jh)\ ,\ i = 0, 1, \dots, N_x\ ,\ j = 0, 1, \dots, N_y\} \tag{4.5}$$

where $N_x = \dfrac{2}{h}$ and $N_y = \dfrac{1}{h}$.
Additionally, we use Central difference approximation in order to approximate the second derivative of $u(x, y)$ respect to x and y, this is:

$$\frac{\partial^2 u_{i,j}}{\partial x^2} \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} \quad \wedge \quad \frac{\partial^2 u_{i,j}}{\partial y^2} \approx \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2} \tag{4.6}$$

where $u_{i,j} = u(x_i, y_j)\ ,\ \forall i = 1, 2, \dots, N_x - 1\ ,\ j = 1, 2, \dots, N_y - 1$.
Therefore, Replacing in the Poisson equation:

$$-\Delta u = f \tag{4.7}$$

we obtain the following 2D discretizacion

$$-\left(\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2}\right) = f_{i,j} \tag{4.8}$$

$$\iff \quad \frac{-u_{i+1,j} - u_{i-1,j} + 4u_{i,j} - u_{i,j+1} - u_{i,j-1}}{h^2} = f_{i,j} \tag{4.9}$$

where $f_{i,j} = f(x_i, y_j)$ with $i = 1, 2, \dots, N_x - 1\ ,\ j = 1, 2, \dots, N_y - 1$.

4.2 Construction of Sparse Matrix and Laplace Matrix

First, we consider the Sparse Matrix D_x which is obtained from the Backward First-order Derivative:

$$D_x = \frac{1}{h}\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 \\ 0 & 0 & 0 & \cdots & 0 & -1 \end{pmatrix}_{N_x \times (N_x - 1)} \tag{4.10}$$

which can be expressed in terms of its components as

$$(D_x)_{i,j} = \frac{1}{h} \cdot \begin{cases} 1 & ,\ \text{if } i = j \\ -1 & ,\ \text{if } i = j+1 \\ 0 & ,\ \text{in Other cases} \end{cases} \tag{4.11}$$

With $i = 1, 2, \ldots, N_x$ and $j = 1, 2, \ldots, N_x - 1$. Thus, its transpose is given by:

$$\left(D_x^T\right)_{i,j} = \frac{1}{h} \cdot \begin{cases} 1 & \text{, if } i = j \\ -1 & \text{, if } j = i + 1 \\ 0 & \text{, in Other cases} \end{cases} \tag{4.12}$$

with $i = 1, 2, \ldots, N_x - 1$ and $j = 1, 2, \ldots, N_x$. Therefore, computing $D_x^T D_x$ we have

$$\begin{aligned}
\left(D_x^T D_x\right)_{i,j} &= \frac{1}{h^2} \cdot \sum_{k=1}^{N_x} \left(D_x^T\right)_{i,k} \cdot (D_x)_{k,j} \\
&= \frac{1}{h^2} \cdot \begin{cases} \sum_{k=1}^{N_x} \left(D_x^T\right)_{i,k} \cdot (D_x)_{k,j} & \text{, if } i = j \\ \sum_{k=1}^{N_x} \left(D_x^T\right)_{i,k} \cdot (D_x)_{k,j} & \text{, if } i = j + 1 \\ \sum_{k=1}^{N_x} \left(D_x^T\right)_{i,k} \cdot (D_x)_{k,j} & \text{, if } j = i + 1 \\ 0 & \text{, in Other cases} \end{cases} \\
&= \frac{1}{h^2} \cdot \begin{cases} \sum\limits_{\substack{k=1 \\ k \neq i \\ k \neq i+1}}^{N_x} \left(D_x^T\right)_{i,k} \cdot (D_x)_{k,i} + \left(D_x^T\right)_{i,i} \cdot (D_x)_{i,i} + \left(D_x^T\right)_{i,i+1} \cdot (D_x)_{i+1,i} & \text{, if } i = j \\ \sum\limits_{\substack{k=1 \\ k \neq j \\ k \neq j+1}}^{N_x} \left(D_x^T\right)_{j+1,k} \cdot (D_x)_{k,j} + \left(D_x^T\right)_{j+1,j} \cdot (D_x)_{j,j} + \left(D_x^T\right)_{j+1,j+1} \cdot (D_x)_{j+1,j} & \text{, if } i = j + 1 \\ \sum\limits_{\substack{k=1 \\ k \neq i \\ k \neq i+1}}^{N_x} \left(D_x^T\right)_{i,k} \cdot (D_x)_{k,i+1} + \left(D_x^T\right)_{i,i} \cdot (D_x)_{i,i+1} + \left(D_x^T\right)_{i,i+1} \cdot (D_x)_{i+1,i+1} & \text{, if } j = i + 1 \\ 0 & \text{, in Other cases} \end{cases} \\
&= \frac{1}{h^2} \begin{cases} 0 + (1 \cdot 1) + (-1 \cdot -1) & \text{, if } i = j \\ 0 + (0 \cdot 1) + (1 \cdot -1) & \text{, if } i = j + 1 \\ 0 + (1 \cdot 0) + (-1 \cdot 1) & \text{, if } j = i + 1 \\ 0 & \text{, in Other cases} \end{cases} \\
&= \frac{1}{h^2} \begin{cases} 2 & \text{, if } i = j \\ -1 & \text{, if } i = j + 1 \\ -1 & \text{, if } j = i + 1 \\ 0 & \text{, in Other cases} \end{cases}
\end{aligned}$$

$\therefore$ we have

$$D_x^T D_x = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 & 0 \\ -1 & 2 & -1 & \cdots & 0 & 0 \\ 0 & -1 & 2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2 & -1 \\ 0 & 0 & 0 & \cdots & -1 & 2 \end{pmatrix}_{(N_x - 1) \times (N_x - 1)} = L_{xx} \tag{4.13}$$

which is the 1D Laplacian Matrix in x-direction.
Doing the same procedure for D_y, we can conclude that

$$D_y^T D_y = L_{yy} \tag{4.14}$$

Where L_{yy} is the 1D Laplacian Matrix in y-direction.

In order to construc the 2D Laplacian Matrix, let's consider the Identity Matrices $I_x = I_{N_x-1}$ and I_{N_y-1}. Additionally, the *Kronecker Product* is defined by:

$$(A \oplus B)_{i,j} = a_{ij} B \tag{4.15}$$

it means that every component of the new matrix is a_{ij} times the matrix B. Thus, the *Kronecker Product* defines a matrix defined by blocks (a matrix which every of its components is a matrix).
On the other hand, L_{xx} can be written in terms of its components as:

$$(L_{xx})_{i,j} = \frac{1}{h^2} \cdot \begin{cases} 2 & \text{, if } i = j \\ -1 & \text{, if } i = j+1 \\ -1 & \text{, if } j = i+1 \\ 0 & \text{, in Other cases} \end{cases} \tag{4.16}$$

Where $i = 1, 2, \ldots, N_x - 1$, $j = 1, 2, \ldots, N_x - 1$. Likewise:

$$(I_y)_{i,j} = \begin{cases} 1 & \text{, if } i = j \\ 0 & \text{, if } i \neq j \end{cases} \tag{4.17}$$

Where $i = 1, 2, \ldots, N_y - 1$, $j = 1, 2, \ldots, N_y - 1$. So, computing the *Kronecker Product* (4.15) between I_y and L_{xx} we have:

$$(I_y \oplus L_{xx})_{i,j} = (I_y)_{ij} L_{xx} = \begin{cases} L_{xx} & \text{, if } i = j \\ 0 & \text{, if } i \neq j \end{cases} \tag{4.18}$$

with $i, j = 1, 2, \ldots, N_y - 1$. Then

$$I_y \oplus L_{xx} = \begin{pmatrix} L_{xx} & 0 & 0 & \cdots & 0 & 0 \\ 0 & L_{xx} & 0 & \cdots & 0 & 0 \\ 0 & 0 & L_{xx} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & L_{xx} & 0 \\ 0 & 0 & 0 & \cdots & 0 & L_{xx} \end{pmatrix} \tag{4.19}$$

Note that every component of this product is a matrix of $(N_x - 1) \times (N_x - 1)$ and the dimension of matrix I_y is $(N_y - 1) \times (N_y - 1)$. Hence, the matrix $I_y \oplus L_{xx}$ is a square matrix of $(N_x - 1) \cdot (N_y - 1)$ rows.
Proceeding of analogous way, we compute $L_{yy} \oplus I_x$ obtaining the following result:

$$(L_{yy} \oplus I_x)_{i,j} = (L_{yy})_{i,j} \cdot I_x \tag{4.20}$$

$$= \frac{1}{h^2} \cdot \begin{cases} 2I_x & \text{, if } i = j \\ -I_x & \text{, if } i = j+1 \\ -I_x & \text{, if } j = i+1 \\ 0 & \text{, in Other cases} \end{cases} \tag{4.21}$$

$$= \frac{1}{h^2} \begin{pmatrix} 2I_x & -I_x & 0 & \cdots & 0 & 0 \\ -I_x & 2I_x & -I_x & \cdots & 0 & 0 \\ 0 & -I_x & 2I_x & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2I_x & -I_x \\ 0 & 0 & 0 & \cdots & -I_x & 2I_x \end{pmatrix} \tag{4.22}$$

Where, following the same reasoning that we did before, we can conclude the $L_{yy} \oplus I_x$ is a square matrix of $(N_x - 1) \cdot (N_y - 1)$ rows.

Finally, the 2D Laplacian matrix in lexicographic order is obtained adding $I_y \oplus L_{xx}$ (4.19) with $L_{yy} \oplus I_x$ (4.22), this is

$$
\begin{aligned}
L &= I_y \oplus L_{xx} + L_{yy} \oplus I_x \\
&= \begin{pmatrix}
L_{xx} & 0 & 0 & \cdots & 0 & 0 \\
0 & L_{xx} & 0 & \cdots & 0 & 0 \\
0 & 0 & L_{xx} & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & L_{xx} & 0 \\
0 & 0 & 0 & \cdots & 0 & L_{xx}
\end{pmatrix}
+ \frac{1}{h^2}
\begin{pmatrix}
2I_x & -I_x & 0 & \cdots & 0 & 0 \\
-I_x & 2I_x & -I_x & \cdots & 0 & 0 \\
0 & -I_x & 2I_x & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 2I_x & -I_x \\
0 & 0 & 0 & \cdots & -I_x & 2I_x
\end{pmatrix} \\
&= \begin{pmatrix}
L_{xx} + \frac{2}{h^2} \cdot I_x & -\frac{1}{h^2} \cdot I_x & 0 & \cdots & 0 & 0 \\
-\frac{1}{h^2} \cdot I_x & L_{xx} + \frac{2}{h^2} I_x & -\frac{1}{h^2} \cdot I_x & \cdots & 0 & 0 \\
0 & -\frac{1}{h^2} \cdot I_x & L_{xx} + \frac{2}{h^2} I_x & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & L_{xx} + \frac{2}{h^2} I_x & -\frac{1}{h^2} \cdot I_x \\
0 & 0 & 0 & \cdots & -\frac{1}{h^2} \cdot I_x & L_{xx} + \frac{2}{h^2} I_x
\end{pmatrix}
\end{aligned}
$$

Noting that

$$
L_{xx} + \frac{2}{h^2} \cdot I_x = \frac{1}{h^2}
\begin{pmatrix}
2 & -1 & 0 & \cdots & 0 & 0 \\
-1 & 2 & -1 & \cdots & 0 & 0 \\
0 & -1 & 2 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 2 & -1 \\
0 & 0 & 0 & \cdots & -1 & 2
\end{pmatrix}
+ \frac{2}{h^2}
\begin{pmatrix}
1 & 0 & 0 & \cdots & 0 & 0 \\
0 & 1 & 0 & \cdots & 0 & 0 \\
0 & 0 & 1 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 1 & 0 \\
0 & 0 & 0 & \cdots & 0 & 1
\end{pmatrix}
= \frac{1}{h^2}
\begin{pmatrix}
4 & -1 & 0 & \cdots & 0 & 0 \\
-1 & 4 & -1 & \cdots & 0 & 0 \\
0 & -1 & 4 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 4 & -1 \\
0 & 0 & 0 & \cdots & -1 & 4
\end{pmatrix}
$$

$\therefore$

$$
L = \frac{1}{h^2}
\left(\begin{array}{ccccccccccccccccccccc}
4 & -1 & \cdots & 0 & 0 & -1 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & & 0 & 0 & \cdots & 0 & 0 \\
-1 & 4 & \cdots & 0 & 0 & 0 & -1 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & & 0 & 0 & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \cdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 4 & -1 & 0 & 0 & \cdots & -1 & 0 & 0 & 0 & \cdots & 0 & 0 & & 0 & 0 & \cdots & 0 & 0 \\
0 & 0 & \cdots & -1 & 4 & 0 & 0 & \cdots & 0 & -1 & 0 & 0 & \cdots & 0 & 0 & & 0 & 0 & \cdots & 0 & 0 \\
-1 & 0 & \cdots & 0 & 0 & 4 & -1 & \cdots & 0 & 0 & -1 & 0 & \cdots & 0 & 0 & & 0 & 0 & \cdots & 0 & 0 \\
0 & -1 & \cdots & 0 & 0 & -1 & 4 & \cdots & 0 & 0 & 0 & -1 & \cdots & 0 & 0 & & 0 & 0 & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \cdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & -1 & 0 & 0 & 0 & \cdots & 4 & -1 & 0 & 0 & \cdots & -1 & 0 & & 0 & 0 & \cdots & 0 & 0 \\
0 & 0 & \cdots & 0 & -1 & 0 & 0 & \cdots & -1 & 4 & 0 & 0 & \cdots & 0 & -1 & & 0 & 0 & \cdots & 0 & 0 \\
 & & \vdots & & & & & \vdots & & & & & \vdots & & & \ddots & & & \vdots & & \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & & 4 & -1 & \cdots & 0 & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & & -1 & 4 & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \cdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & & 0 & 0 & \cdots & 4 & -1 \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & & 0 & 0 & \cdots & -1 & 4
\end{array}\right)
\tag{4.23}
$$

4.3 Constructing the Linear Algebraic Problem

First of all, we need to order the unknown grid values $u_{i,j}$ in a single vector $\mathbf{u} \in \mathbb{R}^{(N_x-1)(N_y-1)}$. This is made writing them in lexicographic order, it means, we first fix j=1 and we write all grid values $u_{i,1}$, subsequently we do the same for

$j = 2$ and continue in this way we obtain the following vector:

$$\mathbf{u} = \begin{pmatrix} u_{1,1} \\ u_{2,1} \\ \vdots \\ u_{N_x-1,1} \\ u_{1,2} \\ u_{2,2} \\ \vdots \\ u_{N_x-1,2} \\ \vdots \\ u_{1,N_y-1} \\ u_{2,N_y-1} \\ \vdots \\ u_{N_x-1,N_y-1} \end{pmatrix} \tag{4.24}$$

On the Other hand, in (4.9) we need to add the boundary conditions, in this we can observe the following cases:

i) Nodes closest to Corners of $\overline{\Omega}$: let's consider the grid point $(i, j) = (1, 1)$ we have

$$\frac{-u_{2,1} - u_{0,1} + 4u_{1,1} - u_{1,2} - u_{1,0}}{h^2} = f_{1,1} \tag{4.25}$$

Nevertheless, the values $u_{0,1}$ and $u_{1,0}$ are boundary values, this is because:

$$u_{0,1} = u(0, h) = \sin(2\pi h) \ \wedge \ u_{1,0} = u(h, 0) = sin(0.5\pi h) \tag{4.26}$$

Thus, replacing (4.26) in (4.25) we can conclude

$$\frac{-u_{2,1} + 4u_{1,1} - u_{1,2}}{h^2} = f_{1,1} + \frac{(u_{0,1} + u_{1,0})}{h^2} \tag{4.27}$$

Using the same reasoning for the grid points: $(1, N_y - 1)$, $(N_x - 1, 1)$ and $(N_x - 1, N_y - 1)$, we obtain the following expressions:

$$\frac{-u_{2,N_y-1} + 4u_{1,N_y-1} - u_{1,N_y-2}}{h^2} = f_{1,N_y-1} + \frac{\left(u_{0,N_y-1} + u_{1,N_y}\right)}{h^2} \tag{4.28}$$

$$\frac{-u_{N_x-1,2} + 4u_{N_x-1,1} - u_{N_x-2,1}}{h^2} = f_{N_x-1,1} + \frac{(u_{N_x-1,0} + u_{N_x,1})}{h^2} \tag{4.29}$$

$$\frac{-u_{N_x-2,N_y-1} + 4u_{N_x-1,N_y-1} - u_{N_x-2,N_y-1}}{h^2} = f_{N_x-1,N_y-1} + \frac{\left(u_{N_x,N_y-1} + u_{N_x-1,N_y}\right)}{h^2} \tag{4.30}$$

ii) Grid points on the form $(i, 1)$, $(i, N_y - 1)$,$(1, j)$ and $(N_x - 1, j)$, with $i = 2, 3, \ldots, N_x - 2$, $j = 2, 3, \ldots, N_y - 2$: Let's consider the grid point $(i, 1)$,from (4.9) we have:

$$\frac{-u_{i+1,1} - u_{i-1,1} + 4u_{i,1} - u_{i+1,1} - u_{i,0}}{h^2} = f_{i,1} \tag{4.31}$$

However, the values $u_{i,0}$ is a boundary value, this is because:

$$u_{0,j} = u(0, jh) = \sin(2\pi jh) \tag{4.32}$$

Thus, replacing (4.32) in (4.31) we obtain

$$\frac{-u_{i+1,1} - u_{i-1,1} + 4u_{i,1} - u_{i+1,1}}{h^2} = f_{i,1} + \frac{u_{i,0}}{h^2} \tag{4.33}$$

Proceeding in the same way we obtain:

$$\frac{-u_{i+1,N_y-1} - u_{i-1,N_y-1} + 4u_{i,N_y-1} - u_{i,N_y-2}}{h^2} = f_{i,N_y-1} + \frac{u_{i,N_y}}{h^2} \tag{4.34}$$

$$\frac{-u_{2,j} - u_{1,j+1} + 4u_{1,j} - u_{1,j-2}}{h^2} = f_{1,j} + \frac{u_{0,j}}{h^2} \tag{4.35}$$

$$\frac{-u_{N_x-1,j} - u_{N_x-1,j-1} + 4u_{N_x-1,j} - u_{N_x-1,j+1}}{h^2} = f_{N_x-1,j} + \frac{u_{N_x,j}}{h^2} \tag{4.36}$$

iii) Other Inner points: the equation (4.29) is applied to those inner points, this is

$$\frac{-u_{i+1,j} - u_{i-1,j} + 4u_{i,j} - u_{i,j+1} - u_{i,j-1}}{h^2} = f_{i,j} \ , \ \forall i = 2,3,\dots,N_x-2, j = 2,3,\dots N_y - 2 \tag{4.37}$$

Hence, considering the right side of the expressions (4.27),(4.28),(4.29),(4.30),(4.33),(4.34), (4.35), (4.36) and (4.37) we obtain the 2D RHS function $\mathbf{f} \in \mathbb{R}^{(N_x-1)\cdot(N_y-1)}$ written in lexicographic order given by:

$$\mathbf{f} = \begin{pmatrix} f_{1,1} + \dfrac{(u_{0,1} + u_{1,0})}{h^2} \\ f_{2,1} + \dfrac{u_{2,0}}{h^2} \\ \vdots \\ f_{N_x-2,1} + \dfrac{u_{N_x-2,0}}{h^2} \\ f_{N_x-1,1} + \dfrac{(u_{N_x-1,0} + u_{N_x,1})}{h^2} \\ f_{1,2} + \dfrac{u_{0,2}}{h^2} \\ f_{2,2} \\ \vdots \\ f_{1,N_y-1} + \dfrac{(u_{0,N_y-1} + u_{1,N_y})}{h^2} \\ f_{2,N_y-1} + \dfrac{u_{2,N_y}}{h^2} \\ \vdots \\ f_{N_x-1,N_y-1} + \dfrac{(u_{N_x,N_y-1} + u_{N_x-1,N_y})}{h^2} \end{pmatrix} \tag{4.38}$$

Therefore, from (4.24),(4.24) and (4.38) we build the linear algebraic problem:

$$L\mathbf{u} = \mathbf{f} \tag{4.39}$$

4.4 Results

For $h = 0.2$, using the command `plt.spy()` we visualize the matrix L (4.23) obtaining the following figure:

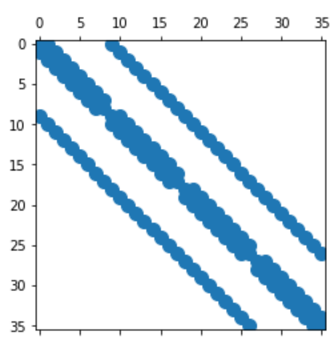

Figure 4.1: Form of 2D Laplace Matrix using plt.spy()

Note that the size of the square matrix L in this case is $(10-1)\cdot(5-1) = 36$ rows.
Since now we consider $h = 0.02$, using the command `np.mgrid[]` we create in python the 2d uniform grid. Subsequently, using the command `plt.imshow()` and `plt.colorbar()` we can visualize the source function $f(x, y)$, as shown in the following figure:

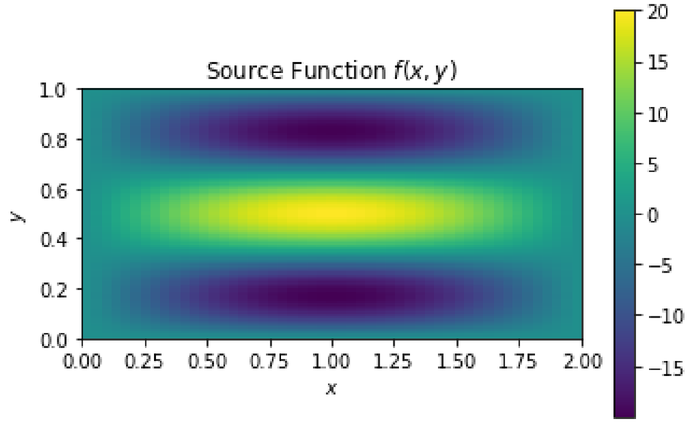

Figure 4.2: Visualization of Source function $f(x, y)$.

In order to write the grid values $u_{i,j}$ in a column vector in lexicographic order, we use the command `np.reshape()` with the option `order='F'`. Then, we use the command `la.spsolve()` in order to solve the linear algebraic problem (4.39). Finally, using `np.reshape()` in order to reshape the solution vector in the appropriate 2D array, we can visualize the solution to the 2D-Poisson problem using `plt.imshow()` and `plt.colorbar()`, obtaining the figure shown above:

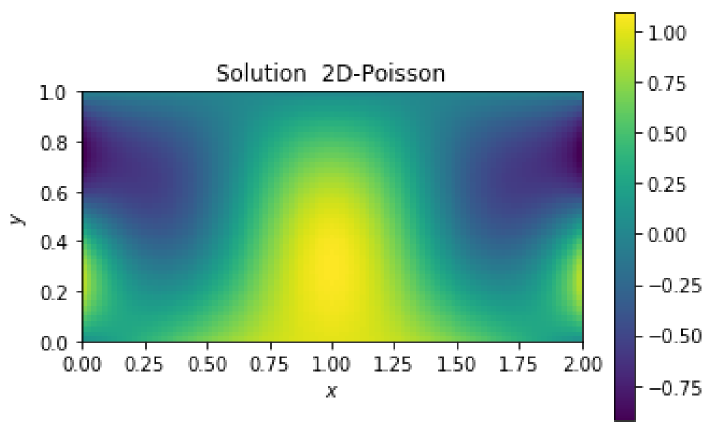

Figure 4.3: Solution of 2D Poisson Equation.

4.5 Discussion

From the results we can observe that:

1. From figure 4.1. we observe the form of the Laplace matrix in 2D matches with the form obtained algebraically in section 4.2, equation (4.23). Likewise, we can conclude that many terms of the 2D Laplace Matrix are zero. Thus, using sparse matrices is appropriate to solve this problem because them help to save memory and solving it faster as a result of only consider the elements of the matrix that are non zero.

2. From figure 4.2, we can observe that there are clearly two zones with negatives values (sink) close to the top and bottom .Conversely, it is in the middle zone where $f(x, y)$ reaches its maximum values (source). Additionally, we can observed that $f(x, y)$ is zero in the sides of Ω, it means $\alpha\Omega$.

3. From figure 4.3, we can observe how the boundary conditions and the zone of high source values afect to the other zones. Moreover, we can observe that the solution matches with the boundary conditions and we can realize that the solution is symmetric as a result of having the same boundary conditions at $x = 0$ and $x = 2$.

5 References

- Budko, N.V.(2019). WI4014TU-Numerical Analysis for PDE's, Lectures 2-3-4-5.
- Burden, Richard (9th edition). Numerical Analysis, chapters 7 and 12.