Assignment 2

Tomer Bar-Natan, Nimrod Berman

Question 1:

Section a:

We will show that the Richardson scheme is unconditionally **unstable** by using the Fourier analysis we taught in class. First, we write the discrete solution on Fourier mode: $u_i^k = \sum_{l=0}^{n_x} \alpha_l \exp\left(i2\pi l j h_x\right)$ and focus on one of these modes: $v_i^k = \exp\left(i2\pi l i h_x\right)$. Denote $v_i^{k+1} = \lambda v_i^k$ and we will proof $|\lambda| > 1$.

Notice that at the Richardson scheme: $v_i^{k+1} = v_i^{k-1} + \frac{2c^2h_t}{h_x^2}\left(v_{i+1}^k - 2v_i^k + v_{i-1}^k\right)$ the term which multiplied by $\frac{2ch_t}{h_x^2}$ is identical to the forward Euler scheme term, so using the Fourier analysis shown in class we get $v_{i+1}^k - 2v_i^k + v_{i-1}^k = v_i^k\left(-4\sin^2(\pi l h_x)\right)$.

Putting the results in Richardson scheme: $v_{i+1}^k = v_i^{k-1} + \frac{2c^2h_t}{h_x^2} v_i^k\left(-4\sin^2(\pi l h_x)\right)$.

Using $v_i^{k+1} = \lambda v_i^k = \lambda^2 v_i^{k-1}$ we conclude: $v_i^{k-1}\left(\lambda^2 + \frac{8c^2h_t}{h_x^2}\sin^2(\pi l h_x)\,\lambda - 1\right) = 0$.

Now we calculate $\lambda^2 + \varphi\lambda - 1 = 0$ for $\frac{8c^2h_t}{h_x^2} sin^2(\pi l h_x) = \varphi$ and we get $\lambda_{1,2} = \frac{-\varphi \pm \sqrt{\varphi^2+4}}{2}$.

Since $\varphi^2 + 4 > 0$ there are two distinct real roots λ_1, λ_2. Also:

$$\lambda_1 + \lambda_2 = \frac{-\varphi + \sqrt{\varphi^2+4}}{2} + \frac{-\varphi - \sqrt{\varphi^2+4}}{2} = -\varphi < 0$$

Together with: $\lambda_1\lambda_2 = \frac{-\varphi+\sqrt{\varphi^2+4}}{2} * \frac{-\varphi-\sqrt{\varphi^2+4}}{2} = -1$ so finally:

$$-1 = \lambda_1\lambda_2 > -\lambda_1\lambda_1 \Rightarrow \lambda_1 > 1 \; and \; \lambda_2 < -1$$

To conclude, we showed that at Richardson scheme $|\lambda| > 1$ no matter how we choose $h_x \; and \; h_t$, which means that the scheme is unconditionally unstable.

Now we will show that the scheme is derived from a second order finite difference method which implies that it has a second order accurate:

$$\frac{\partial u}{\partial t}(x_i, t) - \frac{c^2\partial^2 u}{\partial x^2} = 0 \xrightarrow[discretizate]{} 0 = \frac{u_i^{k+1} - u_i^{k-1}}{2h_t} - \frac{c^2}{h_x^2}\left(u_{i+1}^k - 2u_i^k + u_{i-1}^k\right) \Longrightarrow$$

which is exactly what we wanted.$\Longrightarrow u_i^{k+1} = u_i^{k-1} + \frac{2c^2h_t}{h_x^2}\left(u_{i+1}^k - 2u_i^k + u_{i-1}^k\right)$

Assignment 2

Tomer Bar-Natan, Nimrod Berman

Section b:

We will proof that the scheme: $u_i^{k+1} = u_i^{k-1} + \frac{2c^2 h_t}{h_x^2}\left(u_{i+1}^k - u_i^{k+1} - u_i^{k-1} + u_{i-1}^k\right)$ is unconditionally **stable**. First, we rewrite the scheme:

$\left(1 + \frac{2c^2 h_t}{h_x^2}\right) u_i^{k+1} = \left(1 - \frac{2c^2 h_t}{h_x^2}\right) u_i^{k-1} + \frac{2c^2 h_t}{h_x^2}\left(u_{i+1}^k + u_{i-1}^k\right)$. From here, we use again Fourier analysis and we get that for: $u_j^k = \sum_{l=0}^{n_x} \alpha_l \exp(i2\pi l j h_x)$, if $v_j^k = \exp(i2\pi l j h_x)$ then:

$$v_{j+1}^k + v_{j-1}^k = \exp(i2\pi l(j+1)h_x) + \exp(i2\pi l(j-1)h_x)$$
$$= \exp(i2\pi l j h_x)(\exp(i2\pi l h_x) + \exp(-i2\pi l h_x)) =$$

$.= v_j^k 2\cos(2\pi l h_x)$

From here if we denote $\frac{2c^2 h_t}{h_x^2} = \varphi$ so we are applying the whole scheme will yields:

$$(1+\varphi)v_i^{k+1} = (1-\varphi)v_i^{k-1} + \varphi 2\cos(2\pi l h_x)\, v_i^k.$$

Like the previous section, we denote $v_i^{k+1} = \lambda v_i^k$ and we get:

$$v_i^{k-1}((1+\varphi)\lambda^2 - 2\varphi\cos(2\pi l h_x)\,\lambda - (1-\varphi)) = 0$$

We now proof that the equation $(1+\varphi)\lambda^2 - 2\varphi\theta\lambda - (1-\varphi)) = 0$ for $\theta = \cos(2\pi l h_x)$ yields that $|\lambda| < 1$. Notice that$|\theta| \le 1$. We will solve: $\lambda = \frac{2\varphi\theta \pm \sqrt{4\varphi^2\theta^2 + 4(1+\varphi)(1-\varphi)}}{2(1+\varphi)} = \frac{\varphi\theta \pm \sqrt{\varphi^2\theta^2 + 1 - \varphi^2}}{1+\varphi}$:

Case 1: If $\varphi^2\theta^2 + 1 - \varphi^2 < 0$ so there are two complex roots λ_1, λ_2 related by $\lambda_1 = \overline{\lambda_2}$.
From here:

$$|\lambda_1|^2 = \frac{\varphi^2\theta^2 - \varphi^2\theta^2 - 1 + \varphi^2}{(1+\varphi)^2} = \frac{1-\varphi}{1+\varphi} <_{(\varphi>0)} 1 \Rightarrow |\lambda_1| = |\lambda_2| < 1$$

Case 2: If $\varphi^2\theta^2 + 1 - \varphi^2 \ge 0$, $|\lambda_1| = \left|\frac{\varphi\theta + \sqrt{\varphi^2\theta^2 + 1 - \varphi^2}}{1+\varphi}\right| \le \left|\frac{\varphi\theta + \sqrt{\varphi^2 + 1 - \varphi^2}}{1+\varphi}\right| = \left|\frac{\varphi\theta + 1}{\varphi + 1}\right|$
$\underset{|\theta|\le 1}{\Longrightarrow} |\varphi\theta + 1| \le |\varphi\theta| + 1 \le \varphi + 1 = |\varphi + 1| \Rightarrow \frac{|\varphi\theta+1|}{|\varphi+1|} \le 1.$

$|\lambda_2| \left|\frac{\varphi\theta - \sqrt{\varphi^2\theta^2 + 1 - \varphi^2}}{1+\varphi}\right| \le \frac{\left|\varphi\theta - \sqrt{\varphi^2\theta^2 - \varphi^2}\right|}{|1+\varphi|} = \frac{|\varphi\theta - \varphi\sqrt{\theta^2 - 1}|}{1+\varphi} \le \frac{\varphi|\theta - \sqrt{\theta^2 - 1}|}{\varphi} = \left|\theta - \sqrt{\theta^2 - 1}\right| =$

. That finishes case 2.$= \left|\theta - i\sqrt{1-\theta^2}\right| = \sqrt{\theta^2 + \left(-\sqrt{1-\theta^2}\right)^2} = \sqrt{\theta^2 + 1 - \theta^2} = 1$

To conclude, in either way $|\lambda| < 1$ no matter how we choose h_x, h_t, which proofs that the scheme is unconditionally stable.

Assignment 2

Tomer Bar-Natan, Nimrod Berman

Section c:

To show the order of the truncation error convergence, we will evaluate the expression: $u_i^k - u(x_i, t_k)$ in terms of our scheme, and then check in what conditions it approaches to 0. First, we evaluate u_i^k of the scheme by Taylor series expression. With the same notations as the previous section we have:

$$0 = (1+\varphi)u_i^{k+1} - (1-\varphi)u_i^{k-1} - \varphi\left(u_{i+1}^k + u_{i-1}^k\right) =$$

$$= (1+\varphi)\sum_{r=0}^{\infty}\frac{\partial^r u|_i^k * h_t^r}{\partial t^r * r!} - (1-\varphi)\sum_{r=0}^{\infty}\frac{\partial^r u|_i^k * (-h_t)^r}{\partial t^r * r!}$$

$$-\varphi\left(\sum_{r=0}^{\infty}\frac{\partial^r u|_i^k * h_x^r}{\partial x^r * r!} + \sum_{r=0}^{\infty}\frac{\partial^r u|_i^k * (-h_x)^r}{\partial x^r * r!}\right) =$$

$$= 2\varphi\sum_{r\ is\ even}\frac{\partial^r u|_i^k * h_t^r}{\partial t^r * r!} + 2\sum_{r\ is\ odd}\frac{\partial^r u|_i^k * h_t^r}{\partial t^r * r!} - 2\varphi\sum_{r\ is\ even}\frac{\partial^r u|_i^k * h_x^r}{\partial x^r * r!} \Rightarrow$$

$$\Rightarrow u_i^k = -\varphi\sum_{0<r\ is\ even}\frac{\partial^r u|_i^k * h_t^r}{\partial t^r * r!} - \sum_{r\ is\ odd}\frac{\partial^r u|_i^k * h_t^r}{\partial t^r * r!} + \varphi\sum_{0<r\ is\ even}\frac{\partial^r u|_i^k * h_x^r}{\partial x^r * r!}$$

Assuming that the scheme is consistent, which means the truncation error approaches to 0, we can deduce convergence of the Taylor series above to u_i^k. It means that a necessary condition for consistent is convergence of the Taylor expression. A little more explicitly:

$$u_i^k = -\frac{c^2 h_t^3}{2h_x^2}u_{tt} - \frac{c^2 h_t^5}{h_x^2 4!}u_{tttt} - \cdots - h_t u_t - h_t^3 u_{ttt} - \cdots + \frac{c^2 h_t}{2}u_{xx} + \frac{c^2 h_t}{4!\,h_x^2}u_{xxxx} + \cdots$$

Here we can see that for convergence, as $h_t, h_x \to 0$ the terms involving the ratio $\frac{h_t}{h_x}$ must also approach to 0. So, it explains that in order to the scheme will be consistent we must require that $\frac{h_t}{h_x} \to 0$.

Finally, we will show that developing the series to an order of two, together with second order finite difference, corresponds with the initial scheme. Expanding the series into such an order gives an error of $O(h_t^2 + h_x^2 + \frac{h_t}{h_x})$ which is exactly what we want to proof.
The expansion is as follows:

$$u_i^k = -\frac{c^2 h_t^3}{2h_x^2}\left(\frac{u_i^{k-1} - 2u_i^k + u_i^{k+1}}{h_t^2} + O(h_t^2)\right) - h_t\left(\frac{u_i^{k-1} + u_i^{k+1}}{2h_t} + O(h_t^2)\right) + \frac{c^2 h_t}{2h_x^2}\left(\frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{h_x^2} + \right.$$

$$\left. =O(h_x^2)\right) + O(h_t^2 + h_x^2 + \frac{h_t}{h_x})$$

which gives us:$= \frac{c^2 h_t}{h_x^2}\left(u_{i-1}^k - u_i^{k-1} - u_i^{k+1} + u_{i+1}^k\right) + u_i^{k-1} - u_i^{k+1} + O(h_t^2 + h_x^2 + \frac{h_t}{h_x})$

And that is what we wanted to proof.$u(x_i, t_k) - u_i^k = O(h_t^2 + h_x^2 + \frac{h_t}{h_x})$

Assignment 2

Tomer Bar-Natan, Nimrod Berman

Section d

We will show now that the implicit Crank-Nicholson scheme is unconditionally **stable** using the Fourier analysis tools we learn in class and used in the previous sections.

The scheme: $\frac{u_i^{k+1}-u_i^k}{h_t} = \frac{c^2}{2}\left(\frac{u_{i+1}^{k+1}-2u_i^{k+1}+u_{i-1}^{k+1}}{h_x^2} + \frac{u_{i+1}^{k}-2u_i^{k}+u_{i-1}^{k}}{h_x^2}\right)$.

Any discrete solution can be a combination of Fourier modes. Let us focus on one of these modes and assume the test vector $v_j^{(k)} = \exp(i2\pi ljh_x)\ \ j \in 0,\dots,n_x$.

By what we have studied on class about forward Euler and Fourier analysis, we will Denote $v_i^{k+1} = \lambda v_i^k$ and we will substitute the Fourier modes into the scheme and we will get the get the next equation: $\frac{v_i^{k+1}-v_i^k}{h_t} = \frac{c^2}{2}\left(\frac{v_{i+1}^{k+1}-2v_i^{k+1}+v_{i-1}^{k+1}}{h_x^2} + \frac{v_{i+1}^{k}-2v_i^{k}+v_{i-1}^{k}}{h_x^2}\right)$.

note that after rearranging the equation we get:

$$v_i^{k+1} - v_i^k = \frac{c^{2`}h_t}{2h_x^2}\left(v_{i+1}^{k+1} - 2v_i^{k+1} + v_{i-1}^{k+1} + v_{i+1}^{k} - 2v_i^{k} + v_{i-1}^{k}\right)$$

Denote $\psi = \frac{c^2h_t}{h_x^2}$ we see:

$$(1+\psi)v_i^{k+1} = (1-\psi)v_i^k + \frac{\psi}{2}(v_{i+1}^{k+1} + v_{i-1}^{k+1} + v_{i+1}^{k} + v_{i-1}^{k})$$

We recall Euler analysis we did on class and get the next identity:

$$v_{i+1}^k - 2v_i^k + v_{i-1}^k = v_i^k(-4\sin^2(\pi l h_x))$$

And now we rearrange the equation again s.t it will be similar to the above formula:

$$-\left(\psi v_{i+1}^{k+1} - 2(1+\psi)v_i^{k+1} + \psi v_{i-1}^{k+1}\right) = \left(\psi v_{i+1}^{k} - 2(1-\psi)v_i^{k} + \psi v_{i-1}^{k}\right)$$

Then using our definition $v_i^{k+1} = \lambda v_i^k$ and the last 2 equations we get that:

$$|\lambda| = \left|\frac{1-2\psi \sin^2(\pi l h_x)}{1+2\psi \sin^2(\pi l h_x)}\right| < 1$$

Note that $\psi > 0$ therefore we get the last equation.

Thus, to conclude, in either way $|\lambda| < 1$ no matter how we choose h_x, h_t, which proofs that the Crank-Nicolson scheme is unconditionally stable.

Assignment 2

Tomer Bar-Natan, Nimrod Berman

Section e

We will show that Crank-Nicholson scheme is second order accurate. Both the implicit and the explicit Euler methods are first order in time and second order in space. The Crank-Nicholson achieves second order accurate in time and in space. We will show how we develop the method and by that present its accuracy in time and space. The Crank-Nicholson scheme goes as follow:

$$\frac{u_i^{k+1} - u_i^k}{h_t} = \frac{c^2}{2}\left(\frac{u_{i+1}^{k+1} - 2u_i^{k+1} + u_{i-1}^{k+1}}{h_x^2} + \frac{u_{i+1}^k - 2u_i^k + u_{i-1}^k}{h_x^2}\right)$$

We will use two Tylor series expansion to derive the time derivate. We going to expand our series one in the forward direction and one on the backward direction.

$$u_i^{k+1} = u_i^{k+\frac{1}{2}} + \frac{\Delta t}{2}\frac{\partial u}{\partial t}|^{k+\frac{1}{2}} + \frac{\left(\frac{\Delta t}{2}\right)^2}{2!}\frac{\partial^2 u}{\partial t^2}|^{k+\frac{1}{2}} + \frac{\left(\frac{\Delta t}{2}\right)^3}{3!}\frac{\partial^3 u}{\partial t^3}|^{k+\frac{1}{2}} + \cdots$$

$$u_i^{k} = u_i^{k+\frac{1}{2}} - \frac{\Delta t}{2}\frac{\partial u}{\partial t}|^{k+\frac{1}{2}} + \frac{\left(\frac{\Delta t}{2}\right)^2}{2!}\frac{\partial^2 u}{\partial t^2}|^{k+\frac{1}{2}} - \frac{\left(\frac{\Delta t}{2}\right)^3}{3!}\frac{\partial^3 u}{\partial t^3}|^{k+\frac{1}{2}} + \cdots$$

then we will subtract the two series and get:

$$u_i^{k+1} - u_i^k = \Delta t\frac{\partial u}{\partial t}|^{k+\frac{1}{2}} + \frac{1}{24}(\Delta t)^2\frac{\partial^3 u}{\partial t^3}|^{k+\frac{1}{2}} + \cdots$$

After rearranging we get:

$$\frac{\partial u}{\partial t}|^{k+\frac{1}{2}} = \frac{u_i^{k+1} - u_i^k}{\Delta t} - \frac{1}{24}(\Delta t)^2\frac{\partial^3 u}{\partial t^3}|^{k+\frac{1}{2}}$$

So therefore we get a second order accuracy in time $(\Delta t)^2$.

since the equation is satisfied at $k + \frac{1}{2}$ the right hand side also must be derived at that point. Since we don't have this point on the grid we will do an average of two steps k and $k + 1$ using second order central difference for the second derivative in space like we did on class for the Euler scheme:

$$\frac{c^2 \partial^2 u_i^k}{\partial x^2} = \frac{c^2}{h_x^2}\left(u_{i+1}^k - 2u_i^k + u_{i-1}^k\right) + (\Delta x)^2 \quad , \frac{c^2 \partial^2 u_i^{k+1}}{\partial x^2} = \frac{c^2}{h_x^2}\left(u_{i+1}^{k+1} - 2u_i^{k+1} + u_{i-1}^{k+1}\right) + (\Delta x)^2$$

Then we average: $\frac{\partial^2 u_i^{k+\frac{1}{2}}}{\partial x^2} = \frac{c^2}{2}\left(\left(u_{i+1}^k - 2u_i^k + u_{i-1}^k + u_{i+1}^{k+1} - 2u_i^{k+1} + u_{i-1}^{k+1}\right) + (\Delta x)^2\right.$

Finally we get both side of the Crank-Nicholson scheme. We can notice now that the accuracy is $O((\Delta x)^2, (\Delta t)^2)$ and thus it is a second order accurate method.

Assignment 2

Tomer Bar-Natan, Nimrod Berman

Question 2:

Section a:

Let $n_x = \frac{2}{h_x}$ and $n_y = \frac{2}{h_y}$ be the number of points in the discretization of $(-1,1)$ at x axis and y axis respectively. Namely, $x_i = -1 + ih_x, y_i = -1 + h_y$. Discretization of the PDE leads us to solving the linear system: $L\boldsymbol{u} = b + BC$ where the notations will be as follows:

for $A_j = \begin{pmatrix} \frac{2}{h_x^2} + \frac{2}{h_y^2} + 5u_{1,j} & -\frac{1}{h_x^2} & & \\ -\frac{1}{h_x^2} & \frac{2}{h_x^2} + \frac{2}{h_y^2} + 5u_{2,j} & -\frac{1}{h_x^2} & \\ & \ddots & \ddots & \ddots \\ & & -\frac{1}{h_x^2} & \frac{2}{h_x^2} + \frac{2}{h_y^2} + 5u_{n_x-1,j} \end{pmatrix} L = \begin{pmatrix} A_1 & I_{h_y} & & & \\ I_{h_y} & A_2 & I_{h_y} & & \\ & I_{h_y} & A_3 & I_{h_y} & \\ & \ddots & \ddots & \ddots & \\ & & I_{h_y} & A_{n_y-2} & I_{h_y} \\ & & & I_{h_y} & A_{n_y-1} \end{pmatrix}$

and I_{h_y} is the identity matrix divided by h_y^2.

$$\mathbf{u} = \left(u_{1,1}, u_{2,1}, \dots u_{n_x-1,1}, u_{1,2}, \dots, \dots, u_{n_x-1,n_y-1}\right)^T$$

for $f_{i,j} = \frac{10(1+y_j)}{(3+x_i)^2 + (1+y_j)^2} b = \left(f_{1,1}, u_{2,1}, \dots f_{n_x-1,1}, u_{1,2}, \dots, \dots, f_{n_x-1,n_y-1}\right)^T$

Also, if i, j satisfy two conditions, we sum the results. $BC \epsilon \mathbb{R}^{n_x+n_y-2}$ $such\ that\ [BC]_{i,j} = \begin{cases} \frac{2(1+y_j)}{4+(1+y_j)^2}\ if\ i = 1 \\ \frac{2(1+y_j)}{4+(1+y_j)^2}\ if\ i = n_x - 1 \\ \frac{4}{(3+x_i)^2+4}\ if\ j = n_y - 1 \\ 0\ else \end{cases}$

The code in Python will be as follows:

```python
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.pyplot import legend, semilogy
import scipy.sparse as sp
import scipy.sparse.linalg as solver

def f_x_y(x, y):
    return 10 * (1 + y) / (((3 + x) ** 2) + ((1 + y) ** 2))

def boundaryR(y): #Right BC
    return 2 * (1 + y) / (4 + (1 + y) ** 2)


def boundaryL(y): #Left BC
    return 2 * (1 + y) / (16 + (1 + y) ** 2)
def boundaryB(x): #Bottom BC
    return 0


def boundaryT(x): #Top BC
    return 4 / (((3 + x) ** 2) + 4)

def actualFunction(x, y):
    return 2 * (1 + y) / ((3 + x) ** 2 + (1 + y) ** 2)
```

Assignment 2

Tomer Bar-Natan, Nimrod Berman

```
def Poisson2D(h_y, h_x):
    ##Defining and flattering the grid
    gridSizeY = int(2 / h_y)
    gridSizeX = int(2 / h_x)
    h = h_x #Assuming that h_x=h_y at this section
    gridUX = np.linspace(-1 + h, 1-h, gridSizeX-1)
    gridUY = np.linspace(-1 + h, 1-h, gridSizeY-1)
    gridX, gridY = np.meshgrid(gridUX, gridUY)
    Xu = gridX.flatten()
    Yu = gridY.flatten()

    ##Adding BC to the solution vector
    b = np.zeros(gridX.shape)
    b[:,0] +=boundary1(gridY[:,0])
    b[:,-1] += boundary2(gridY[:,-1])
    b[0,:] += boundary3(gridX[0,:])
    b[-1, :] += boundary4(gridX[-1,:])
    b = b.flatten()
```

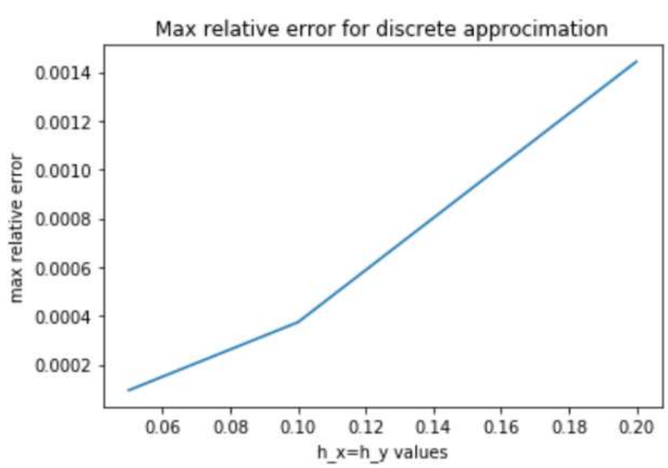
```
##Adding the given function to the solution vector
for i in range(b.shape[0]):
    b[i] += f_x_y(Xu[i], Yu[i])*(h**2)

##Define the 1D-Laplacian
main_diag = (-2)* np.ones(gridUX.shape[0])
sub_diag = np.ones(gridUX.shape[0] - 1)
lap1D = sp.coo_matrix(np.diag(sub_diag, k=-1) + np.diag(main_diag) + np.diag(sub_diag, k=1))

##Define the 2D-Laplacian
Id_mat = sp.identity(gridUX.shape[0])
lap2D = sp.kron(Id_mat, lap1D) + sp.kron(lap1D, Id_mat)
lap2D = -lap2D +sp.identity(lap2D.shape[0])*5*(h**2)

##Solving the linear system and calculating the error
approx_solution = solver.spsolve(lap2D, b)
correct_solution = np.array([actualFunction(i, j) for i, j in zip(Xu, Yu)])
error_vector = np.array(abs(correct_solution - approx_solution))
relative_err = np.divide(error_vector, correct_solution, where=correct_solution != 0)
relative_max_error = np.max(relative_err)
return relative_max_error
```

Running the code for h_x=h_y in [0.2 ,0.1 ,0.05] we get:



Assignment 2

Tomer Bar-Natan, Nimrod Berman

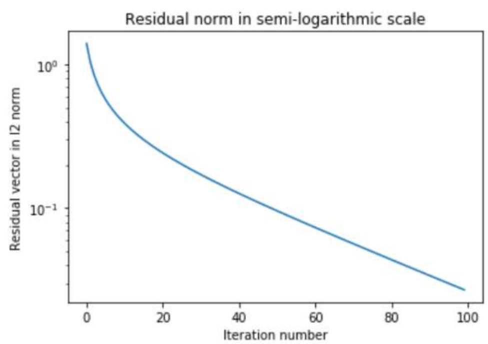
Section a:

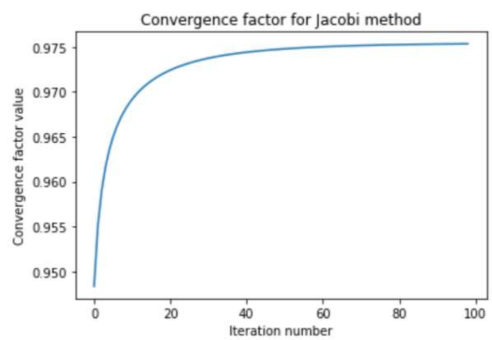
With the same notations as the previous section, we will now solve the linear equation $L\boldsymbol{u} = b + BC$ using Jacobi and Gauss-Seidel iterative method.

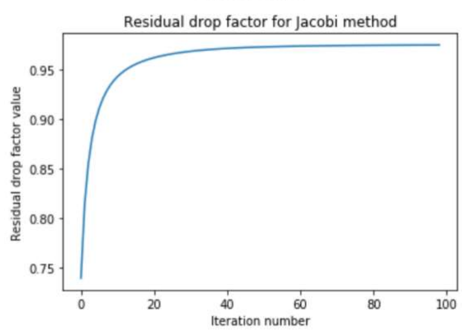
Starting with Jacobi method, the code will be as follows:

```
def JacobiIterations(lap2D, b, solution):
    residual_vector = []
    error_vector = []
    lap2D = np.array(sp.coo_matrix.todense(lap2D))
    M = np.diag(lap2D)
    N = lap2D - np.diagflat(M)
    x = np.zeros(lap2D.shape[0])
    for i in range(100):
        x= (b-np.dot(N,x))/M
        residual_vector.append(np.linalg.norm(b - np.dot(lap2D,x)))
        error_vector.append(np.linalg.norm(solution - x))
```

In this case we applied 100 iterations with initial vector $x^{(0)} = \bar{0}$ and choose h_x=h_y=0.1 for lap2D matrix to get the results:







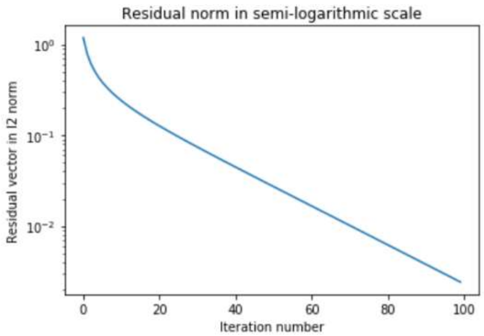
Now, for the Gauss-Seidel method the code will be as follows:

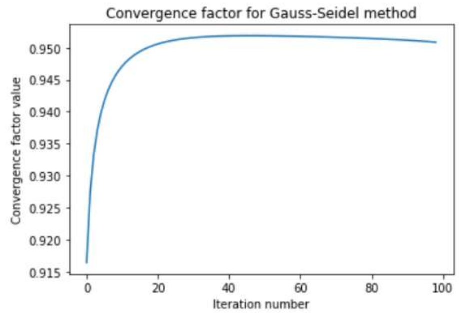
```
def GaussSeidelIterations(lap2D, b, solution):
    residual_vector = []
    error_vector = []
    lap2D = np.array(sp.coo_matrix.todense(lap2D))
    M = np.tril(lap2D)
    N = lap2D - M
    x = np.zeros(lap2D.shape[0])
    for i in range(100):
        x= np.matmul(inv(M),(b-np.matmul(N,x)))
        residual_vector.append(np.linalg.norm(b - np.dot(lap2D,x)))
        error_vector.append(np.linalg.norm(solution - x))
```

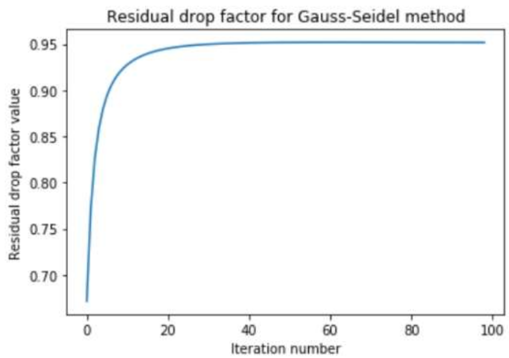
Assignment 2

Tomer Bar-Natan, Nimrod Berman

Again, we applied 100 iterations with initial vector $x^{(0)} = \bar{0}$ and choose h_x=h_y=0.1 for lap2D matrix to get the results:







Section c:

As shown in class, the Jacobi method in scalar form the iteration is:
for the equation $Ax = b,$ *and the iterations* $x^{k+1} = D^{-1}\left(b - (L+U)x^k\right)$:

$$x_i^{k+1} = \frac{1}{a_{ii}}\left(b_i - \sum_{j \neq i} a_{ij} x_j^k\right)$$

In our case, the matrix A is 2D-Laplacian operator, so Jacobi scalar form iteration is:
for $u_{l,j}^k$ is the value of $u_{l,j}$ in the k iteration.$u_{l,j}^{k+1} = -\frac{1}{4}\left(h^2 b_{l,j} - \left(u_{l+1,j}^k + u_{l-1,j}^k + u_{l,j-1}^k + u_{l,j+1}^k\right)\right)$
Denote $u(x_l, y_j)$ to be the exact solution.

So the error at iteration k at l, j is: $e_{l,j}^k = u\left(x_l, y_j\right) - u_{l,j}^k$ and from here we can calculate:

$$e_{l,j}^{k+1} = u\left(x_l, y_j\right) - u_{l,j}^{k+1} = \frac{1}{4}\left(e_{l+1,j}^k + e_{l-1,j}^k + e_{l,j-1}^k + e_{l,j+1}^k\right)$$

Using furrier analysis, denote: $v_{l,j}^k = \exp\left(i2\pi\left(lw_x + jw_y\right)h\right)$ and we calculate:

$$v_{l+1,j}^k + v_{l-1,j}^k = \exp\left(i2\pi\left((l+1)w_x + jw_y\right)h\right) + \exp\left(i2\pi\left((l-1)w_x + jw_y\right)h\right) =$$
$$\exp\left(2i\pi h\left(lw_x + jw_y\right)\right)\left(\exp(2i\pi lhw_x) + \exp(-2i\pi lhw_x)\right) = v_{l,j}^k 2\cos\left(2i\pi lhw_x\right)$$

Applying for the whole error expression yields:

$$\frac{1}{4}\left(v_{l+1,j}^k + v_{l-1,j}^k + v_{l,j-1}^k + v_{l,j+1}^k\right) = \frac{1}{4}\left(v_{l,j}^k 2\cos(2i\pi lhw_x) + v_{l,j}^k 2\cos\left(2i\pi lhw_y\right)\right) =$$
$$= v_{l,j}^k\left(\frac{1}{2}\cos(2i\pi lhw_x) + \frac{1}{2}\cos\left(2i\pi lhw_y\right)\right)$$

Assignment 2

Tomer Bar-Natan, Nimrod Berman

From here we can see that as h approaches 0, $\frac{1}{2}\cos(2i\pi lhw_x) + \frac{1}{2}\cos\left(2i\pi lhw_y\right)$ approaches to 1. It means that for a small h, $e^{k+1} = (1-\varepsilon)e^k$ for a small ε, and at the limit the Jacobi iteration is unable to reduce the error.