Question 1

The Poisson equation in two-dimensional Cartesian coordinates with zero boundary conditions is given as:

$$\bigtriangledown^2 T = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -2x \quad 0 < x, y < 1 \tag{1}$$

This is approximated on a uniform grid, such that $T_{i,j} \approx T(i\Delta x, j\Delta y)$, where $\Delta x = 1/n_x$ and $\Delta y = 1/n_y$ and i = 0, . . . , n_x, j = 0, . . . , n_y. Poissons equation therefore is approximated by

$$\delta_x^2 T_{i,j} + \delta_y^2 T_{i,j} = -2 * x_i \quad i = 1, ..., n_x - 1, \; j = 1, ..., n_y - 1 \tag{2}$$

where δ_x^2 and δ_y^2 represent the standard centered differing approximations to the second derivatives; (xi, yj) within the domain. This gives (nx 1)(ny 1) equations.This, together with the boundary conditions,

$$T_{0,j} = T_{i,0} = T_{i,M} = T_{N,j} = 0, \quad i = 1, ..., n_x - 1, \; j = 1, ..., n_y - 1, \tag{3}$$

First, this problem is extended into a larger domain 1 ¡ x ¡ 1 in x over which the odd extension is sought at each interior value of y_j so that $u(x, y_j) = u(x, y_j)$. The difference equations for the unknowns U_i, j at $x_i = -1 + i\Delta t$ for $i = 0, 1, ..., N$, where $N = 2n_x$, then become

$$\hat{\delta}_x^2 T_{i,j} + \hat{\delta}_y^2 T_{i,j} = -2 * x_i \quad i = 1, ..., N - 1, \; j = 1, ..., n_y - 1 \tag{4}$$

For i = 0 and i = N/2 the right-hand side in effect has $f_{0,j} = 0$ and $f_{N/2,j} = 0$, using the boundary conditions $U_{0,j} = U_{n_x,j} = 0$ and the properties of the odd extension. The Discrete Fourier Transforms of $T_{i,j}$ and $f_{i,j}$ for each fixed value of j, taking the sum is performed over k can be defined as

$$\hat{T}_{n,j} = \frac{1}{N} \sum_{k=0}^{N-1} T_{k,j} \exp^{-2\pi ink/N} \; and \; F_{n,j} = \frac{1}{N} \sum_{k=0}^{N-1} f_{k,j} \exp^{-2\pi ink/N} \tag{5}$$

The Poisson equation in a rectangle with zero boundary conditions can be entirely expressed in Discrete Fourier Terms as:

$$\hat{\delta}_y^2 \hat{T}_{i,j} + \frac{2}{(\Delta x)^2}[cos(2\pi n/N)1]\hat{T}_{i,j} = F_{i,j} \quad forj = 1, ..., (n_y 1) \tag{6}$$

Since $T_{i,0}$ and $T_{i,n_y} = 0$ for all i, it also follows that

$$\hat{T}_{i,0} and \hat{T}_{i,n_y} = 0$$

for all n = 0, . . . ,(N - 1), and therefore at each value of n is an $(n_y 1)(n_y 1)$ tridiagonal system for $\hat{T}_{n,1}, ..., \hat{T}_{n,n_y 1}$.

The original unknowns $T_{i,j}$ are then obtained from the inverse transform:

$$T_{i,j} = \sum_{n=0}^{N-1} \hat{T}_{n,j} \exp^{-2\pi ink/N} \quad fori = 1, ..., N - 1, \; j = 1, ..., n_y - 1 \tag{7}$$

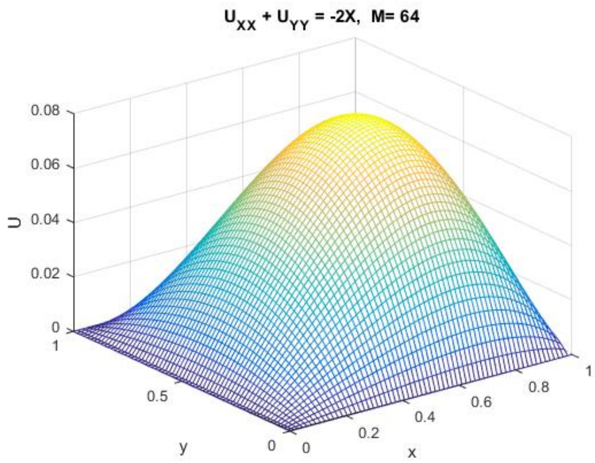

Figure 1: Poisson Equation with $N_x = N_y = 64$

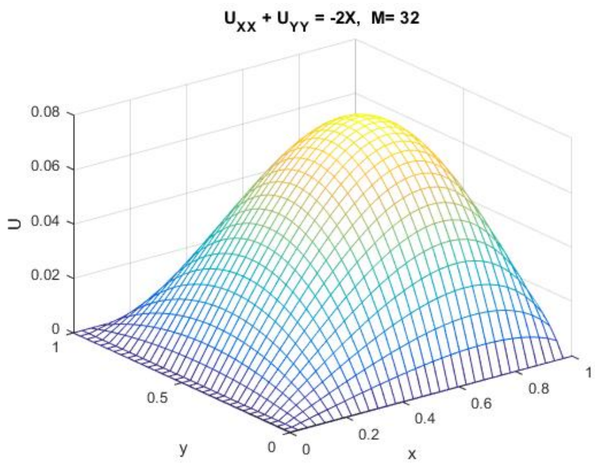

Figure 2: Poisson Equation with $N_x = N_y = 32$

The number of operations required to solve this problem using FFTs is $O(N_{n_y} logN)$ for both the $f_{i,j}$ transform and the $\hat{T}_{n,j}$ inversion, with a further $O(N_{n_y})$ operations for the tridiagonal solver.

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2n} \\ \hdotsfor{5} \\ x_{d1} & x_{d2} & x_{d3} & \dots & x_{dn} \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{d1} & x_{d2} & x_{d3} & \dots & x_{dn} \end{bmatrix}$$

The have been two plots set up, with grid size 32 and 64. Then, in order to evaluate the accuracy an exact solution is approximated with a grid size 4096. Further for computation of the norm, the matrix dimension is scaled to similar sizes by means of interpolation. The norm of plot with grid size 64 is lower than that of grid size 32, hence in accordance with theory.

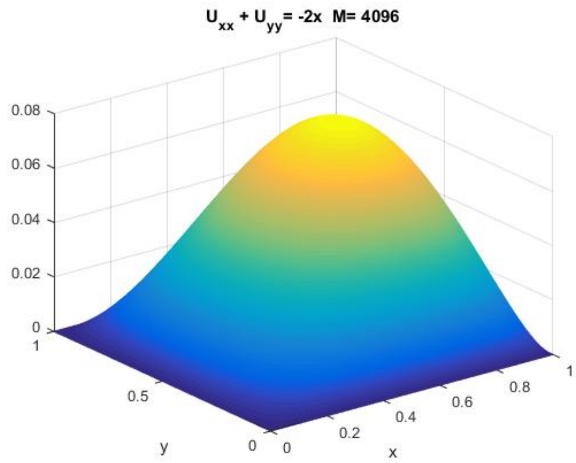

Figure 3: Approximated Exact Solution with $N_x = N_y = 4096$

Question 2

The second question of the assignment involves developing a function which implements the preconditioned gradient method and the preconditioned conjugate gradient method. The preconditioner should be of the form:

$$M_2 M_1 Z = r,$$

therefore a two-step process is implemented to calculate z . If M_2 is not set, identity matrix is considered; whereas for non-conditioned both M1 and M2 are not set , then they should be set to the identity matrix. Altogether, there are 8 cases; steepest descent method and the conjugate gradient method, using no preconditioning and preconditioning using Jacobi iteration, symmetric GaussSiedel and symmetric SOR.

$$\omega = 2 - \frac{2 * \pi}{n_x}$$

The Steepest Descent Method is a first-order iterative optimization algorithm that involves find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient of the function at the current point. Only 1 matrix-vector multiplication at each step takes place such that:

$$r^{(k)} = b - Tx^{(k)} = b - T(x^{(k-1)} + \alpha_{k-1} r^{k+1}) = r^{(k-1)} - \alpha_{k-1} T r^{(k-1)} \tag{8}$$

The problem with the method of steepest descent or gradient is that the intuitive direction for performing a line search is not necessarily the best direction. Therefore in Conjugate Gradient Method we choose the conjugate vectors p_k carefully, to obtain a good approximation to the solution x^*.

$$p^{(k)} = r^{(k-1)} + \beta_{(k-1)} p^{(k-1)}$$

where

$$\beta_{(k-1)} = \frac{(p^{(k-1)})^T T r^{(k-1)}}{(p^{(k-1)})^T T p^{(k-1)}}$$

In most cases, preconditioning is necessary to ensure fast convergence of the conjugate gradient method. The preconditioned conjugate gradient method takes the following form: