CONSTRUCTION OF MULTIGRID SOLVER FOR 2D HEAT CONDUCTION PROBLEM

By:

MUHAMMAD AQIL BIN AZMAN

(Matrix No:120396)

Supervisor:

Dr. -Ing Muhammad Razi Abdul Rahman

June 2017

This dissertation is submitted to
Universiti Sains Malaysia
As partial fulfillment of the requirement to graduate with honors degree in
BACHELOR OF ENGINEERING (MECHANICAL ENGINEERING)

School of Mechanical Engineering
Engineering Campus
Universiti Sains Malaysia

Declaration

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed ....(MUHAMMAD AQIL BIN AZMAN)

Date ....

STATEMENT 1

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by giving explicit references.

Bibliography/references are appended.

Signed ....(MUHAMMAD AQIL BIN AZMAN)

Date ....

STATEMENT 2

I hereby give consent for my thesis, if accepted, to be available for photocopying and for interlibrary loan, and for the title and summary to be made available outside organizations.

Signed ....(MUHAMMAD AQIL BIN AZMAN)

Date ....

Acknowledgement

After undergoing countless of efforts and discussions, Alhamdulillah praise to Allah for giving me strength and guidance to me in completing my final year project as partial fulfillment of the requirement to graduate. First and foremost, I would like to express my gratitude to my project's supervisor, Dr. -Ing Muhammad Razi Abdul Rahman for giving me suggestions and advices as well as encouragements along the project. This project would not be completed without the help of my supervisor. Moreover, my special thanks to the authority of Universiti Sains Malaysia (USM) especially to School of Mechanical Engineering for providing me with all the facilities needed to complete this project. Next, I also would like to thank to my parents for always giving advice and encouragement whenever I have problems. Finally, I would like to thank all the people who involved directly or indirectly in helping me during the project.

Title of Thesis: **Construction of Multigrid Solver for 2D Heat Conduction Problem.**

Date of Submission (Academic Year): **7th June 2017 (2016/2017).**

Candidate (Matric No): **Muhammad Aqil Bin Azman (120396).**

Name of Supervisor: **Dr. -Ing Muhammad Razi Abdul Rahman.**

Tables of Content

List of Tables

List of Figures

Abstrak

Kajian ini menerangkan pembentukan dan penggunaan kaedah grid-pelbagai untuk masalah konduksi haba 2D. Satu kaedah grid-pelbagai (MG) pada asasnya adalah penyelesai matriks yang digunakan dengan kaedah pengiraan lain untuk menyelesaikan persamaan pembezaan separa (PDE) seperti kaedah unsur terhingga (FEM), kaedah unsur sempadan (BEM), terhingga berbeza kaedah (FDM) dan lain-lain. penggubalan antara FEM dan MG digunakan untuk menguji prestasi gabungan ini melalui penyelesaian. penyelesaian melibatkan separa persamaan pembezaan (PDE) persamaan Poisson 2D masalah pengaliran haba dan penyelesaian yang diselesaikan dengan menggunakan Matlab. Persamaan Poisson telah diuji dengan pelbagai jenis sumber haba dan kesilapan L2 norma dan H1 norma telah dikira untuk mengesahkan dan membuktikan penumpuan penyelesaian. Penyelesaian FEM dan FEM-MG dibandingkan dan FEM-MG mengandungi dua jenis pelicinan Gauss-Siedel dan berturut-turut lebih bersantai (SOR). Hasil kajian menunjukkan bahawa kesilapan L2 dan H1 norma dalam FEM-MG kecil berbanding FEM dengan konvensional sistem linear penyelesai.

Abstract

This research describes the formulation and application of the multigrid method for the 2D heat conduction problem. A Multigrid method (MG) is essentially a matrix solver which is used with another computational method for solving partial differential equation (PDE) such as finite element method (FEM), boundary element method (BEM), finite different method (FDM) etc. The formulation between FEM and MG is used to test the performance of this combination through the solution. The solution involves partial differential equation (PDE) of Poisson equation of 2D heat conduction problem and the solutions solved by using Matlab. The Poisson equation was tested with various types of heat source and the error L2 norm and H1 norm were computed to validate and prove the convergence of the solution. The solution of FEM and FEM-MG were compared and FEM-MG contains two types of smoother Gauss-Siedel and Successive Over Relaxation (SOR). The result shows that the error of L2 and H1 norm in FEM-MG smaller compare to FEM with conventional linear system solver.

1 Introduction

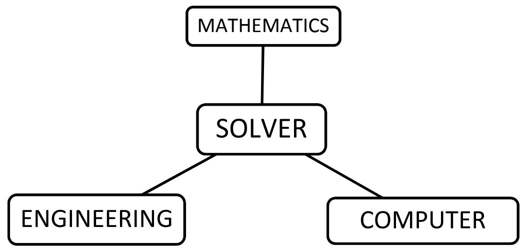


Figure 1.1: Interaction between MATHEMATICS, ENGINEERING, and COMPUTER.

Now a day, the simulation analysis widely developed every year. Whereby the combination of mathematics, engineering and computer knowledge were used. Simulation analysis is the other method besides experimental analysis to use in analysis problem which is dealing with the real problem for examples structural analysis, fluid flow analysis, fluid-structure interaction and more. The data collected from simulation need to be verified before it can be used by comparing the data between experimental and analytical solution. Simulation analysis contains several computational methods such as Finite Element Method (FEM), Finite Difference Method (FDM), Smooth Particles Hydrodynamic (SPH) and more. These methods are the way to solve the solution compare to analytical and experimental. The numbers and types of mesh or iteration are important to get the accurate result but for fine mesh or higher iteration need higher time to get the result of the solution [1]. Therefore the best choice of the mesh or iteration is important.

Multigrid method (MG) is the one of the numerical method to solve partial differential equation (PDE) and the idea of MG technique is using hierarchy discretization to solve the solution. The MG involve interpolation between finest grid and coarsest grid and the step involves are relaxation, prolongation and restriction. Relaxation step is where the first v-cycle was relaxed before the calculated variables transferring (restricting) to the next-coarser grid and after the interpolating and adding the correction at the second v-cycle stage. Prolongation step is the reverse step of restriction where coarsest grid to finest grid [2].

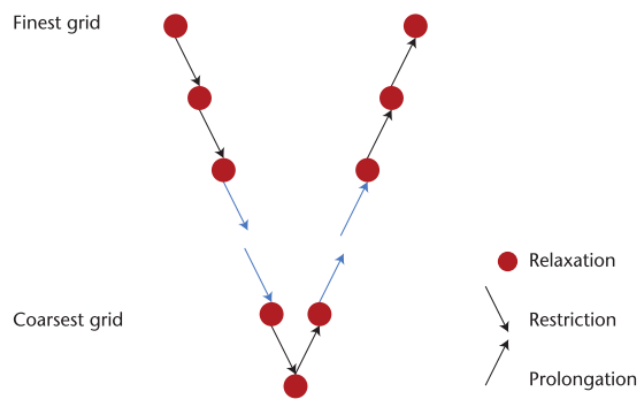


Figure 1.2: Multigrid V-cycle steps.

Multigrid provide a small error compare to other method because it such a hybrid method which combine with other method such as finite element or finite different and have the smoothing element such as Gauss-Siedel. The function of the smoothing element is to reduce high frequency occur during the cycle interpolation but there is an effect on the low frequency components [3]. There are smoothing properties such as the error after few steps are being able to smooth by using classical iterative method Gauss-Seidel (GS) and the convergence rate is good in first few steps and decrease considerably afterward [4][5]. The multigrid method with line Gauss-Seidel relaxation is found to work very well in solving fourth-order 2D Poisson equation and special multigrid methods are developed to solve the resulting sparse linear systems efficiently [3]. Coarse Grid principles are smooth function on a fine grid can be approximated satisfactorily on a grid with less discretization points, whereas oscillating function would disappear[6]. Furthermore, a coarse grid is less expensive compare to fine grid and to approximate the low frequency error components on a coarse grid.

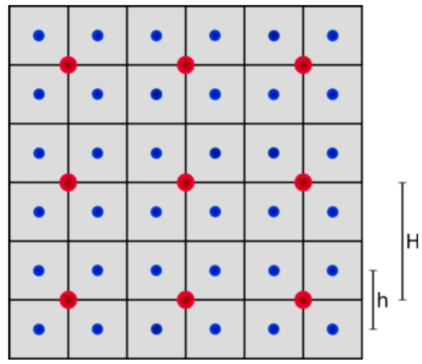


Figure 1.3: Grid in 2D with cell-centered coarsening. Small (blue) circles denote fine grid points, big (red) circles coarse grid points [7].

Multigrid iteration also known as V-*cycle* [8] and being summarized in figure 1-4 below show the algorithm of multigrid method

Algorithm 1 Recursive V-*cycle*: $u_h^{(k+1)} = V_h\left(u_h^{(k)}, A^h, f^h, v_1, v_2\right)$

if coarsest level **then**
solve $A^h u^h = f^h$ exactly or by many smoothing iterations
else
$\bar{u}_h^{(k)} = S_h^{v_1}(u_h^{(k)}, A^h, f^h)$ *//pre-smoothing*
$r^h = f^h - A^h \bar{u}_h^{(k)}$ *//compute residual*
$r^H = I_h^H r^h$ *//restrict residual*
$e^H = V_h(0, A^H, u_h^{(k)}, r^H, v_1, v_2)$ *//recursion*
$e^h = I_H^h e^H$ *//interpolate error*
$\tilde{u}_h^{(k)} = \bar{u}_h^{(k)} + e^h$ *//coarse grid correction*
$u_h^{(k+1)} = S_h^{v_2}(\tilde{u}_h^{(k)}, A^h, f^h)$ *//post-smoothing*
end if

Figure 1.4: Algorithm for multigrid [7].

The focus of this study is combination of multigrid and finite element method and to test the finite element multigrid (FEM-MG) with two types of smoother Gauss-Seidel and Successive Over Relaxation (SOR) compare with finite element method (FEM) using Matlab. The error between FEM-MG Gauss-Seidel, FEM-MG SOR and FEM were investigated to check which one is better. The graphical user interface was created to reduce human effort and make the result easily manipulated by the user.

2 Methodology

2.1 Theory

2.1.1 Finite element method

By referring [9] chapter 10,

Step 1: Choose element types.

For quadrilateral element (square) with nodal temperature t_i, t_j, t_k and t_m.

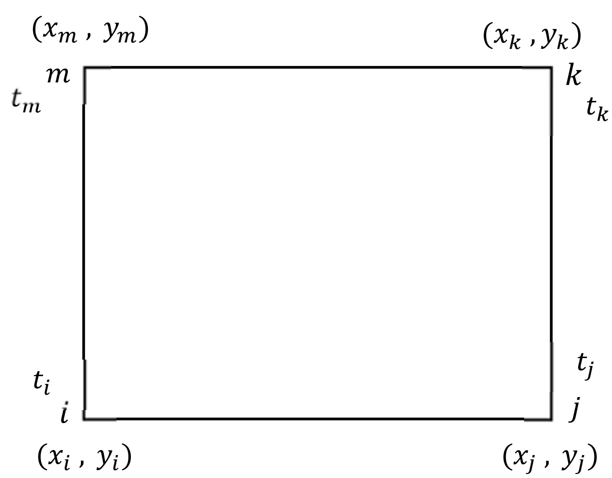


Figure 2.1: Basic quadrilateral element with nodal temperatures.

Step 2: Select the temperature function.

The temperature function is given by

$$\{T\} = [N_i \quad N_j \quad N_k \quad N_m]\begin{Bmatrix} t_i \\ t_j \\ t_k \\ t_m \end{Bmatrix} \tag{2.1}$$

Nodal t_i, t_j, t_k and t_m can moving in 2 dimensional spaces x and y direction. So, for every nodal with divide into two components x and y.

$$\{t\} = \begin{Bmatrix} t_i \\ t_j \\ t_k \\ t_m \end{Bmatrix} = \begin{Bmatrix} t_{i_x} \\ t_{i_y} \\ t_{j_x} \\ t_{j_y} \\ t_{k_x} \\ t_{k_y} \\ t_{m_x} \\ t_{m_y} \end{Bmatrix} \tag{2.2}$$

For linear displacement function

$$t_x(x,y) = a_1 + a_2x + a_3y + a_4xy \tag{2.3}$$

$$t_y(x,y) = a_5 + a_6x + a_7y + a_8xy \tag{2.4}$$

The general temperature function $\{T\}$, which stores the function of t_x and t_y

$$\{T\} = \begin{Bmatrix} a_1 + a_2x + a_3y + a_4xy \\ a_5 + a_6x + a_7y + a_8xy \end{Bmatrix} = \begin{bmatrix} 1 & x & y & xy & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & x & y & xy \end{bmatrix}\begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \end{Bmatrix} \tag{2.5}$$

To obtain values of a's, substitute nodal t_i, t_j, t_k and t_m into t_x and t_y equation 3.3 and 3.2

$$\begin{aligned}
t_{i_x} &= t_x(x_i, y_i) = a_1 + a_2 x_i + a_3 y_i + a_4 x_i y_i \\
t_{j_x} &= t_x(x_j, y_j) = a_1 + a_2 x_j + a_3 y_j + a_4 x_j y_j \\
t_{k_x} &= t_x(x_k, y_k) = a_1 + a_2 x_k + a_3 y_k + a_4 x_k y_k \\
t_{m_x} &= t_x(x_m, y_m) = a_1 + a_2 x_m + a_3 y_m + a_4 x_m y_m
\end{aligned} \tag{2.6}$$

$$\begin{aligned}
t_{i_y} &= t_y(x_i, y_i) = a_5 + a_6 x_i + a_7 y_i + a_8 x_i y_i \\
t_{j_y} &= t_y(x_j, y_j) = a_5 + a_6 x_j + a_7 y_j + a_8 x_j y_j \\
t_{k_y} &= t_y(x_k, y_k) = a_5 + a_6 x_k + a_7 y_k + a_8 x_k y_k \\
t_{m_y} &= t_y(x_m, y_m) = a_5 + a_6 x_m + a_7 y_m + a_8 x_m y_m
\end{aligned} \tag{2.7}$$

The a's at beginning can be solve with the first four

From equation above $t_x(x, y)$ and $t_y(x, y)$ by eliminate a's the equations 3.9 and 3.10 obtained

$$\begin{Bmatrix} t_i \\ t_j \\ t_k \\ t_m \end{Bmatrix} = \begin{bmatrix} 1 & x_i & y_i & x_i y_i \\ 1 & x_j & y_j & x_j y_j \\ 1 & x_k & y_k & x_k y_k \\ 1 & x_m & y_m & x_m y_m \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{Bmatrix} \tag{2.8}$$

$$\begin{aligned}
t_x(x, y) = \frac{1}{4bh}\Big[&(b - x)(h - y)t_{i_x} + (b + x)(h - y)t_{j_x} \\
&+ (b + x)(h + y)t_{k_x} + (b - x)(h + y)t_{m_x}\Big]
\end{aligned} \tag{2.9}$$

$$t_y(x,y) = \frac{1}{4bh}\Big[(b-x)(h-y)t_{i_y} + (b+x)(h-y)t_{j_y}$$

$$+ (b+x)(h+y)t_{k_y} + (b-x)(h+y)t_{m_y}\Big]$$

These equations 3.9, can be expressed equivalently in terms of the shape function and unknown nodal temperatures as

$$\{T\} = [N]\{t\} \tag{2.10}$$

The shapes functions can be obtained by rearrange equation 3.1 are given by

$$[N] = \{T\}\{t\}^{-1} \tag{2.11}$$

Where the shapes functions are given by

$$N_i = \frac{(b-x)(h-y)}{4bh}$$
$$N_j = \frac{(b+x)(h-y)}{4bh}$$
$$N_k = \frac{(b+x)(h+y)}{4bh} \tag{2.12}$$
$$N_m = \frac{(b-x)(h+y)}{4bh}$$

Step 3: Define the Temperature Gradient/Temperature and Heat Flux/Temperature Gradient Relationships

$$\{g\} = \begin{Bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{Bmatrix} \tag{2.13}$$

$$\{g\} = \begin{bmatrix} \frac{\partial N_i}{\partial x} & \frac{\partial N_j}{\partial x} & \frac{\partial N_k}{\partial x} & \frac{\partial N_m}{\partial x} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_j}{\partial y} & \frac{\partial N_k}{\partial y} & \frac{\partial N_m}{\partial y} \end{bmatrix} \begin{Bmatrix} t_i \\ t_j \\ t_k \\ t_m \end{Bmatrix} \tag{2.14}$$

Rearrange equation 3.13, $[B]$ can be obtained by substitutes equations 3.11 into $\{g\}$.

$$[B] = \{g\}\{t\}^{-1} \tag{2.15}$$

$$[B] = \begin{bmatrix} \frac{\partial N_i}{\partial x} & \frac{\partial N_j}{\partial x} & \frac{\partial N_k}{\partial x} & \frac{\partial N_m}{\partial x} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_j}{\partial y} & \frac{\partial N_k}{\partial y} & \frac{\partial N_m}{\partial y} \end{bmatrix} \{t_i \quad t_j \quad t_k \quad t_m\} \tag{2.16}$$

$$[B] = \frac{1}{4bh} \begin{bmatrix} -(h-y) & 0 & (h-y) & 0 & (h+y) & 0 & -(h+y) & 0 \\ 0 & -(b-x) & 0 & -(b+x) & 0 & (b+x) & 0 & (b-x) \\ -(b-x) & -(h-y) & -(b+x) & (h-y) & (b+x) & (h+y) & (b-x) & -(h+y) \end{bmatrix} \tag{2.17}$$

The heat flux/temperature gradient relationship is now

$$\begin{Bmatrix} q_x \\ q_y \end{Bmatrix} = -[D]\{g\} \tag{2.18}$$

Where material property matrix is

$$[D] = \begin{bmatrix} K_{xx} & 0 \\ 0 & K_{yy} \end{bmatrix} \tag{2.19}$$

Step 4 Derive the element conduction Matrix and Equations.

$$[k] = \iiint_V [B]^T[D][B]\, dV + \iint_{S_3} h[N]^T[N]dS \tag{2.20}$$

For heat conduction problem and assuming constant thickness in the element,

$$[k_c] = \iiint_V [B]^T[D][B]\, dV$$

$$= tA[B]^T[D][B]$$

$$= \frac{tA}{16b^2h^2}\begin{bmatrix} -(h-y) & 0 & -(b-x) \\ 0 & -(b-x) & -(h-y) \\ (h-y) & 0 & -(b+x) \\ 0 & -(b+x) & (h-y) \\ (h+y) & 0 & (b+x) \\ 0 & (b+x) & (h+y) \\ -(h+y) & 0 & (b-x) \\ 0 & (b-x) & -(h+y) \end{bmatrix}\begin{bmatrix} K_{xx} & 0 \\ 0 & K_{yy} \end{bmatrix} \tag{2.21}$$

$$\begin{bmatrix} -(h-y) & 0 & (h-y) & 0 & (h+y) & 0 & -(h+y) & 0 \\ 0 & -(b-x) & 0 & -(b+x) & 0 & (b+x) & 0 & (b-x) \\ -(b-x) & -(h-y) & -(b+x) & (h-y) & (b+x) & (h+y) & (b-x) & -(h+y) \end{bmatrix}$$

For heat source $\{f_Q\}$

$$\{f_Q\} = Q\iiint_V [N]^T dV \tag{2.22}$$

$$\{f_Q\} = [k]\{t\} \tag{2.23}$$

2.1.2 Finite Element-Multigrid

The variables K stiffness matrix and f_Q load vector matrix from 2.1.1 where used at figure 1.4: algorithm for multigrid and for square element the value b and h are equal size.

$u_h^{(k+1)} = V_h\left(u_h^{(k)}, K^h, {f_Q}^h, v_1, v_2\right)$

if coarsest level **then**

solve $K^h u^h = {f_Q}^h$ exactly or by many smoothing iterations

else

$\bar{u}_h^{(k)} = S_h^{v_1}(u_h^{(k)}, K^h, {f_Q}^h)$ *//pre-smoothing*

$r^h = {f_Q}^h - K^h \bar{u}_h^{(k)}$ *//compute residual*

$r^H = I_h^H r^h$ *//restrict residual*

$e^H = V_h(0, K^H, u_h^{(k)}, r^H, v_1, v_2)$ *//recursion*

$e^h = I_H^h e^H$ *//interpolate error*

$\tilde{u}_h^{(k)} = \bar{u}_h^{(k)} + e^h$ *//coarse grid correction*

$u_h^{(k+1)} = S_h^{v_2}(\tilde{u}_h^{(k)}, A^h, f^h)$ *//post-smoothing*

end if

2.2 Case study

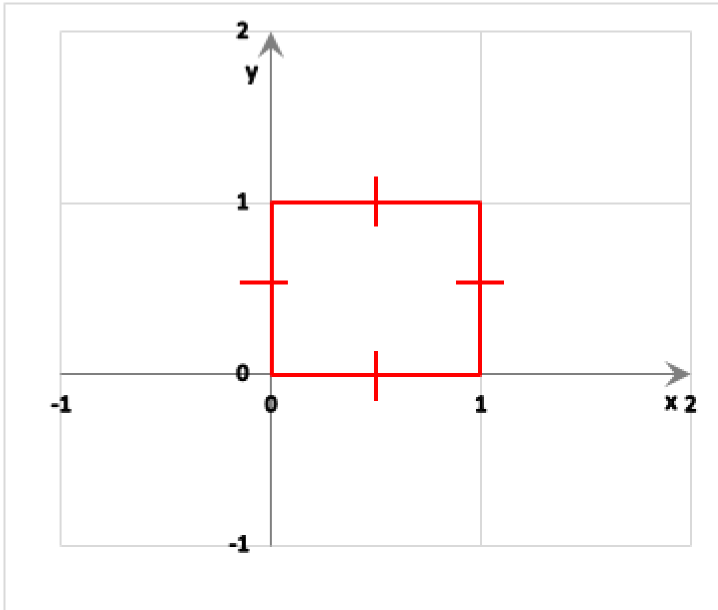


Figure 2.2: Dimension of problem (square)

The heat equation $\nabla^2 u = -f(x,y)$ where $\nabla^2 u$ are equal to $K_{xx}\frac{\partial^2 u}{dx^2} + K_{yy}\frac{\partial^2 u}{dy^2} + K_{zz}\frac{\partial^2 u}{dz^2}$ but for 2 dimensional cases the $\frac{\partial^2 u}{dz^2}$ is equal to zero. Therefore the $\nabla^2 u$ will become $K_{xx}\frac{\partial^2 u}{dx^2} + K_{yy}\frac{\partial^2 u}{dy^2}$ and the equation is $K_{xx}\frac{\partial^2 u}{dx^2} + K_{yy}\frac{\partial^2 u}{dy^2} = -f(x,y)$. Whereby $f(x,y)$ is the source of heat that supply to the equation. Assume the type of material is constant in x and y direction, $K_{xx} = K_{yy} = K$ which is thermal conductivity equal to 1 $W/(unit \cdot °C)$ then the full equation is equal to $\frac{\partial^2 u}{dx^2} + \frac{\partial^2 u}{dy^2} = -f(x,y)$. All boundaries west, east, south, north were fixed at zero.

The solution will apply various type of heat source:

Case 1: $f(x,y) = -[2x(x-1) + 2y(y-1)]$

Case 2: $f(x,y) = -[2x-2]$

Case 3: $f(x,y) = -[4-2y-2x]$

2.3 Implementation

The performance of the FEM-MG was test by using Matlab 2014b code by Hardik Kothari [10] where the codes were summarize in figure 2.3 below. The graphical user interface (GUI) was constructed. The Matlab was running at personal laptop and computer at CAD lab School of Mechanical Engineering, Engineering Campus, Universiti Sains Malaysia.

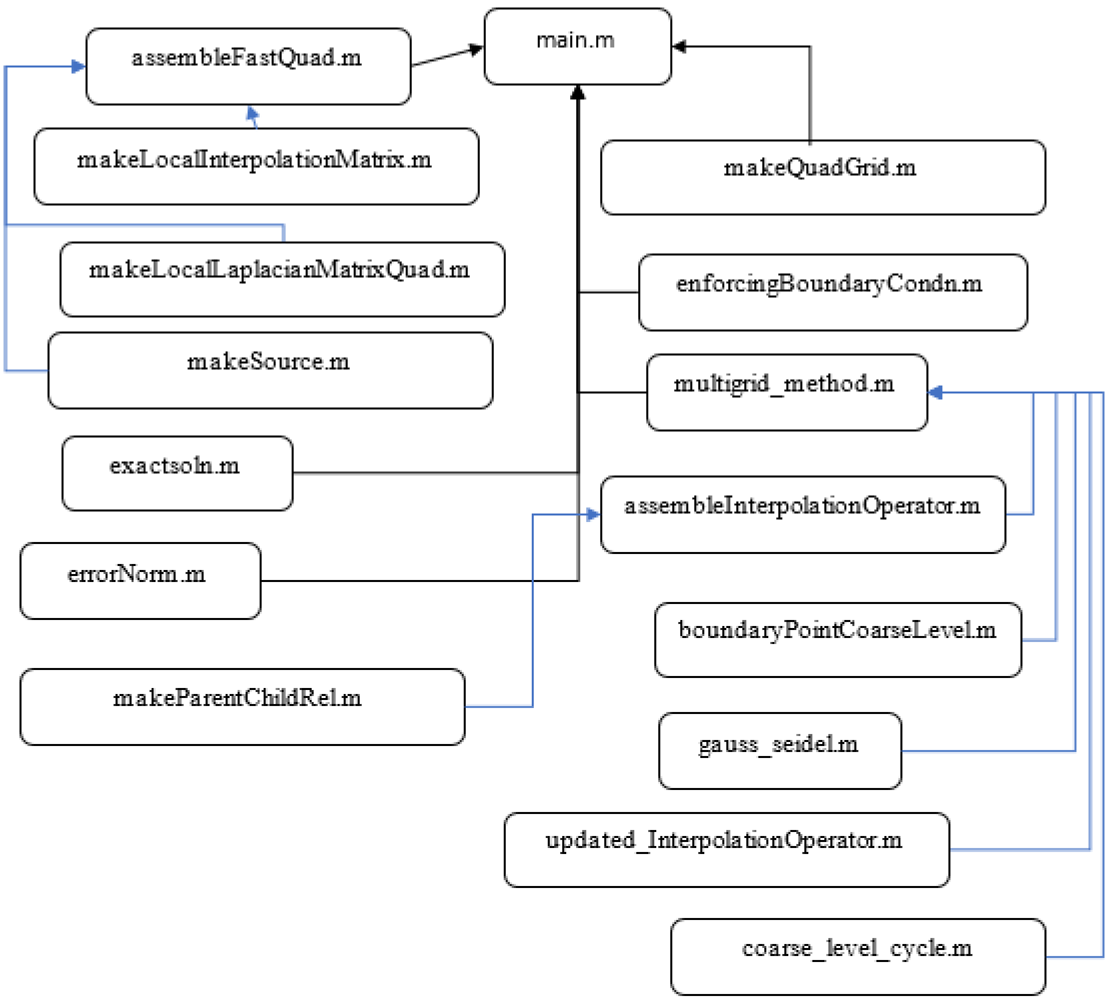


Figure 2.3: Flow chart of Matlab script and function of Multigrid-FEM solver [10].

The function of the creation of GUI is to reduce human effort and make the result easy to watch and manipulate. The GUI contain the variables that need to key in by user to formulate the result. User can choose the smoother that being generate (code) easily without changing at the main code.

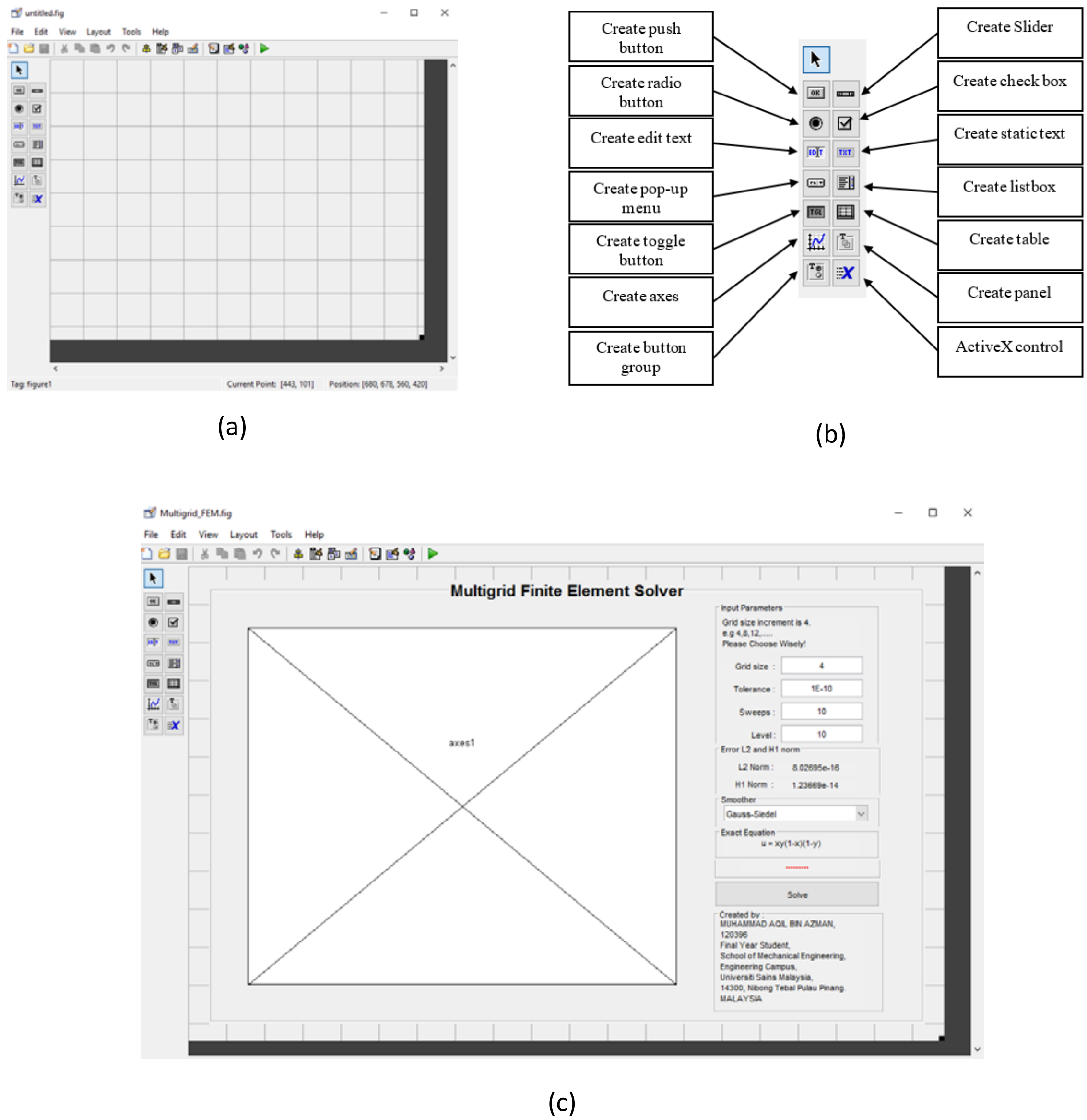
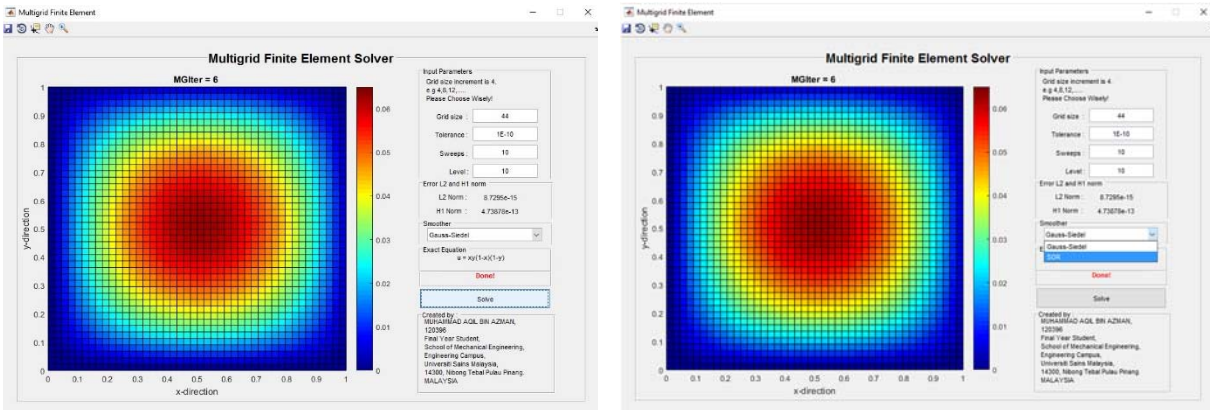


Figure 2.4: Setup Graphical User Interface Using Matlab: (a) Workplace (b) tools (c) Design layout



(d). (e)

Figure 2.5: Graphical User Interface:(d) after result (e)smoother pick

3 Result and discussion

3.1 Case 1

Case 1: $f(x,y) = -[2x(x-1) + 2y(y-1)]$

Based on table 3.1 below, the different of the contour hardly to be seen because the error of the solution by MG-FEM with Gauss-Seidel smoother and MG-FEM with Successive Over Relaxation (SOR) smoother very small and the different that can be seen located at second Multigrid iteration where MG-FEM with SOR contain nine contour lines compare to MG-FEM with Gauss-Seidel come out with seven contour lines. The differences occur based on the formulation of the Gauss-Seidel and SOR at equation 3.1 and 3.2 where the value of ω at SOR were set $\frac{2}{3}$ between $0 < \omega < 2$, if $\omega = 1$ the SOR equation will be Gauss-Seidel. Multigrid with Gauss-Seidel as smoother effective on problems of practical, and on model problems powerful smoothing characteristics cost little to implement in serial computation context [11].

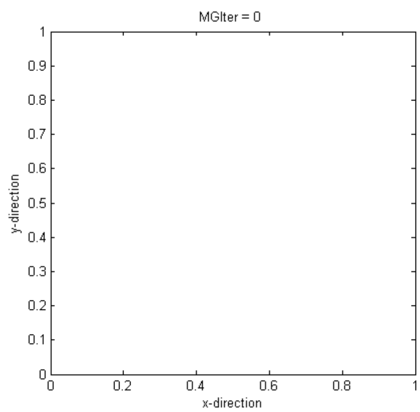
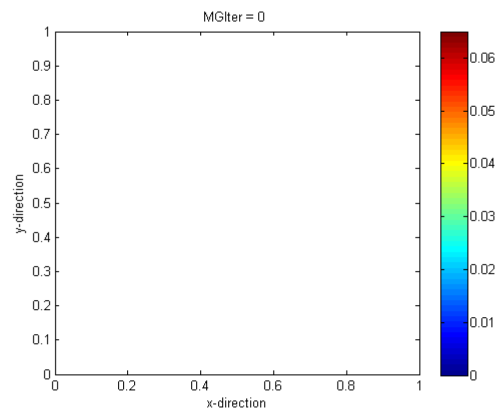
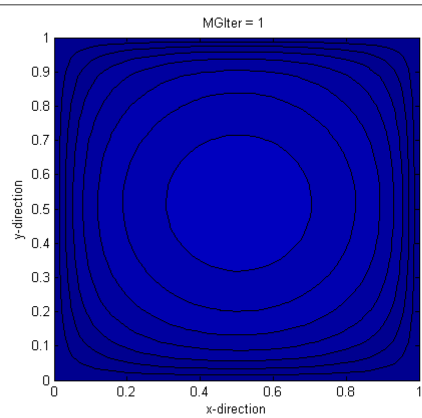
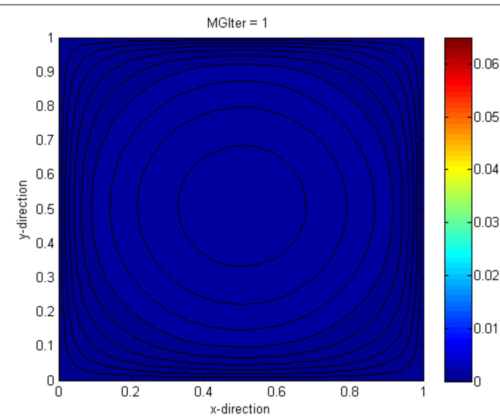
Gauss-Seidel:

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j<i} a_{ij}x_j^{(k+1)} - \sum_{j>i} a_{ij}x_j^{(k)}\right) \quad (3.1)$$

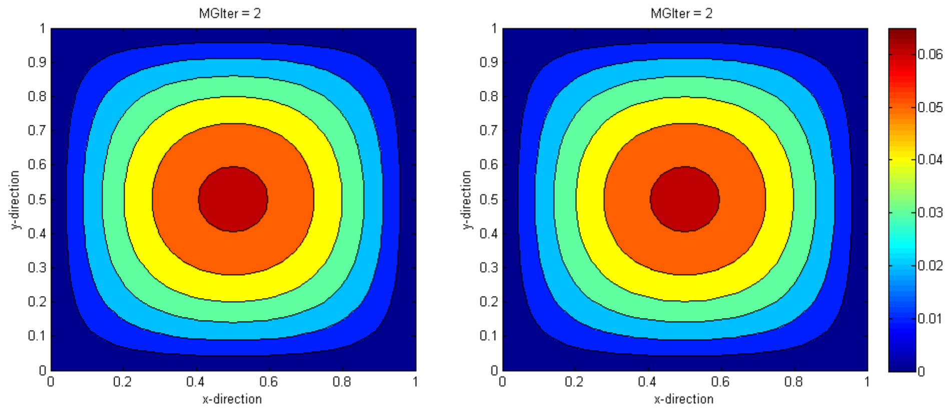
SOR: $$x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j<i} a_{ij}x_j^{(k+1)} - \sum_{j>i} a_{ij}x_j^{(k)}\right) \tag{3.2}$$

The calculations of errors are needed to show the different for the MG-FEM with Successive Over Relaxation smoother and MG-FEM with Gauss-Seidel smoother.

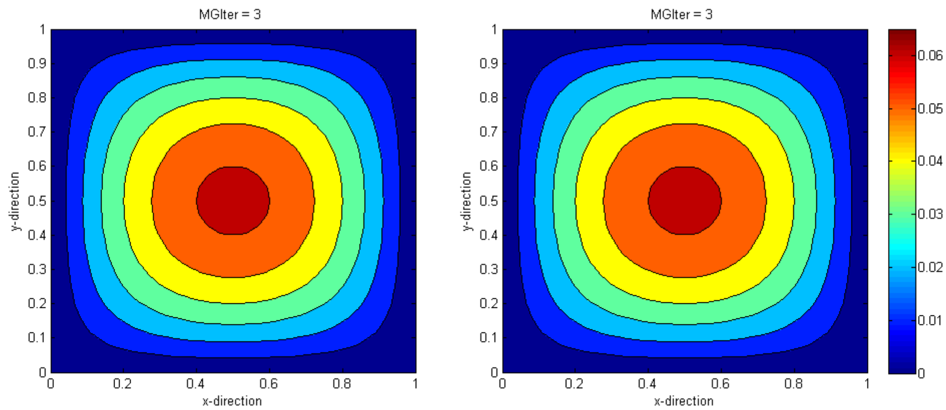
Table 3.1: Comparison Multigrid Finite Element by using Gauss-Seidel and Successive Over Relaxation(SOR), Grid size of 44 for case 1

Multigrid iteration	**Multigrid Finite Element by Using Gauss-Seidel as smoother**	**Multigrid Finite Element by Using Successive Over Relaxation (SOR) as smoother**
0		
1		

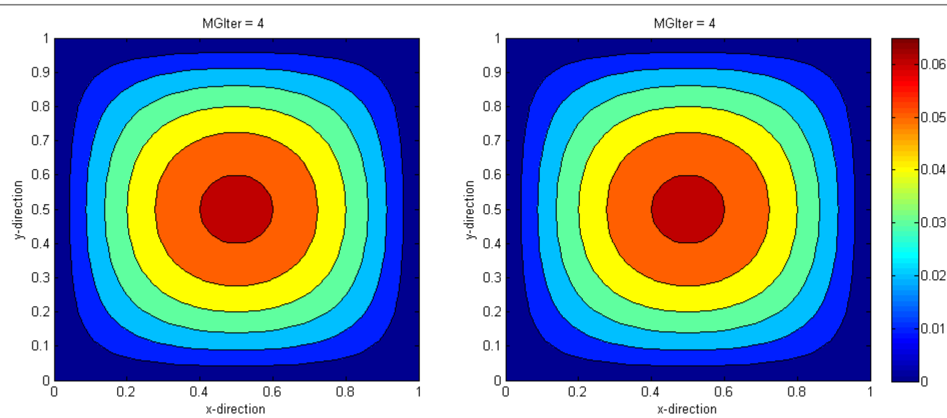
2


MGIter = 2
MGIter = 2
y-direction
x-direction

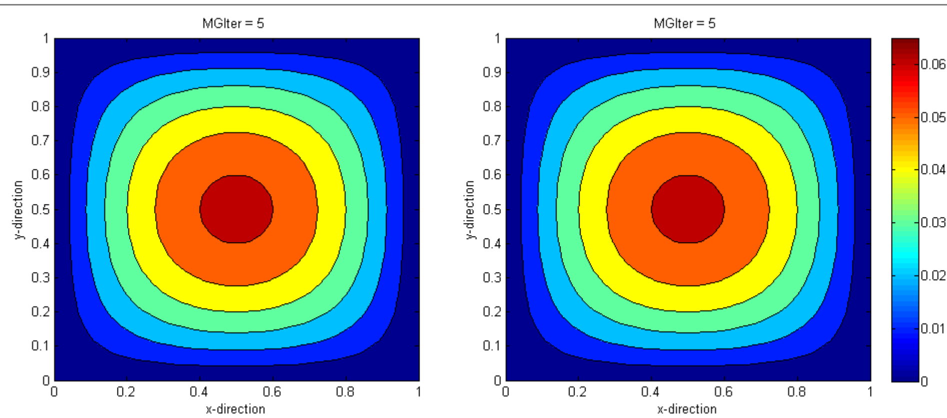
3

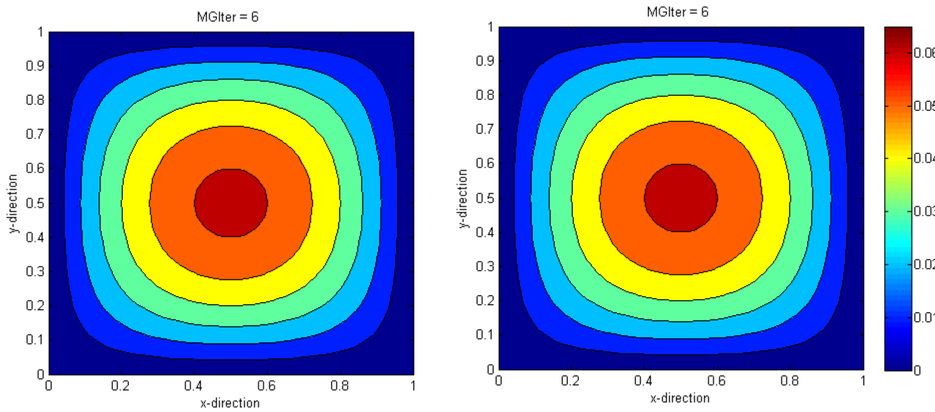

MGIter = 3
MGIter = 3
y-direction
x-direction

4


MGIter = 4
MGIter = 4
y-direction
x-direction

5


MGIter = 5
MGIter = 5
y-direction
x-direction

6	

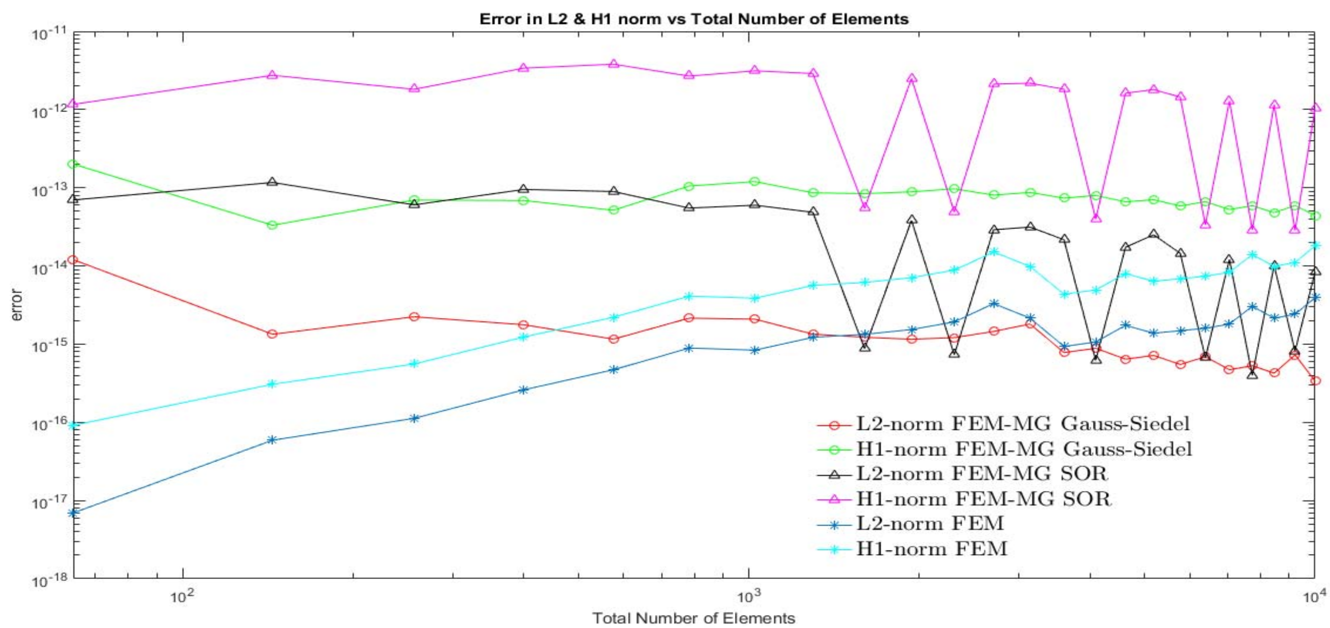


Figure 3.1: Error in L2 & H1 norm vs Total Number of elements for case 1

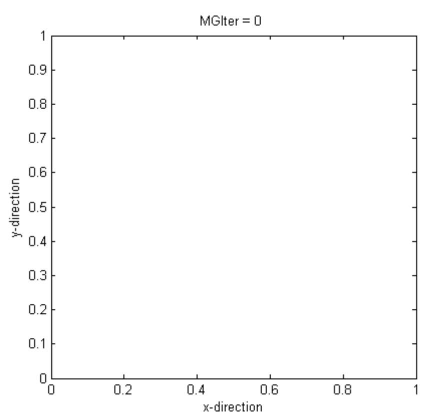
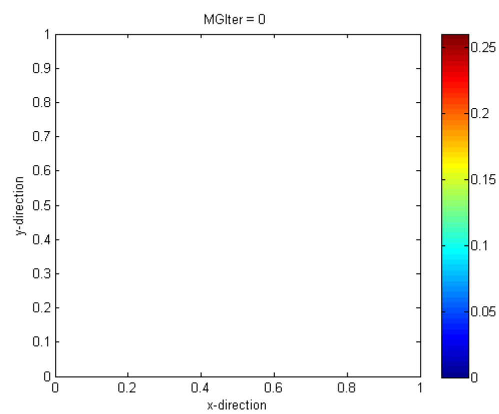
Based on figure 4.2, start at grid size of 8 by 8 which contain 64 square elements the convergence of error of Multigrid-FEM with Gauss-Seidel smoother decrease 88.88% to grid size of 12 by 12. After that, the error drop again by 15.39% at grid size of 24 by 24. However, the error slightly increases by 44.45% at grid size of 32 by 32. Suddenly, the error drop again 41.96% at grid size 48 by 48. Next, the error fluctuates and continues drop by 72.22% at grid size of 100 by 100. By comparing lines between three errors of last result (grid size 100 by 100) of L2 norm for FEM-MG Gauss-Seidel, FEM-MG SOR and FEM, the error FEM-MG Gauss-Seidel give the small error of 2.0190292847e-14 compare to FEM-MG SOR, 2.3066648816e-14 and FEM, 3.2899940541e-14. Therefore, the Multigrid-FEM with Gauss-Seidel smoother give the best result of the error compare to MG-FEM SOR smoother and FEM.

3.2 Case 2

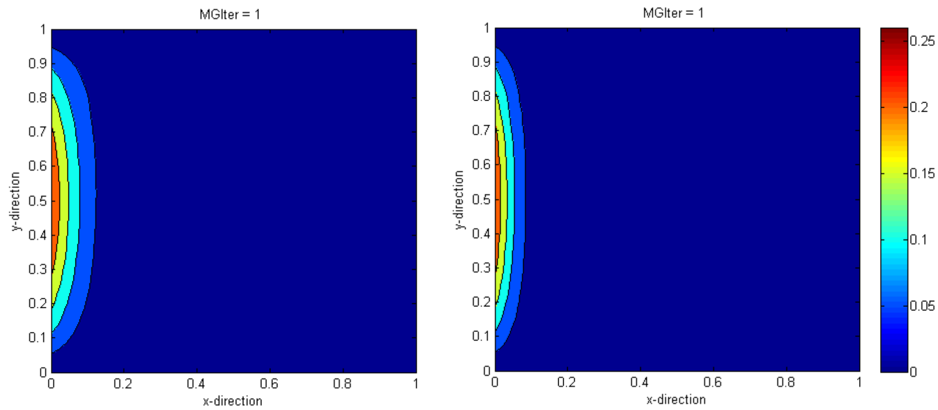
Case 2: $$f(x,y) = -[2x - 2]$$

Based on table 3.2, the contour differences clearly marked occur at MG iteration 1 where the size of MG-FEM Gauss-Seidel is bigger than size of contour of MG-FEM SOR but for others iteration the differences were hardly to differentiate with each other. Therefore, the calculations of error were needed to see the detail differences occurred. For this case the FEM-MG with SOR computed extra one iteration compare to FEM-MG with Gauss-Seidel.

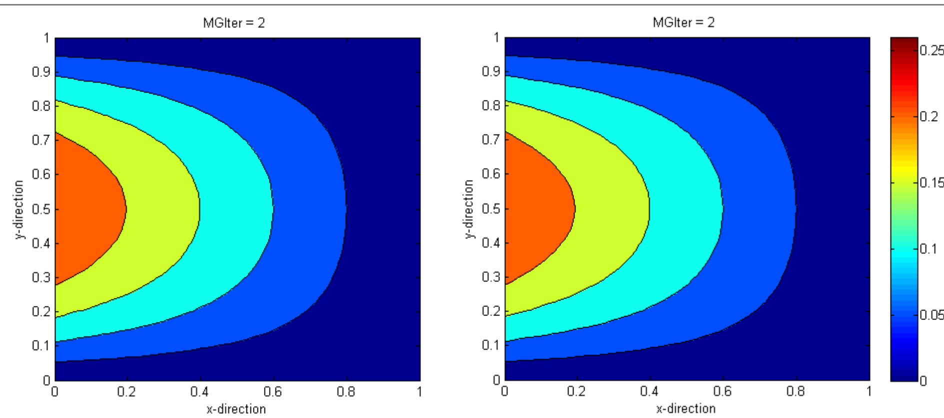
Table 3.2: Comparison Multigrid Finite Element by using Gauss-Seidel and Successive Over Relaxation(SOR), Grid size of 44 for case 2

Multigrid iteration	**Multigrid Finite Element by Using Gauss-Seidel as smoother**	**Multigrid Finite Element by Using Successive Over Relaxation (SOR) as smoother**
0		

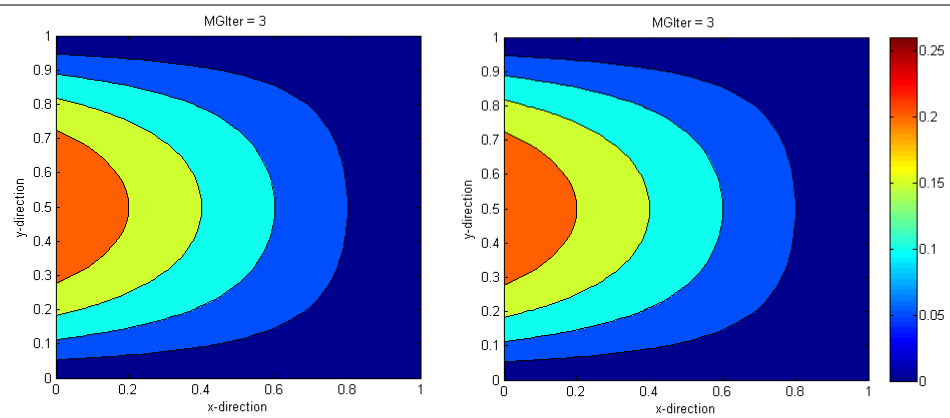
1


MGIter = 1
y-direction
x-direction
MGIter = 1
y-direction
x-direction
0.25
0.2
0.15
0.1
0.05
0

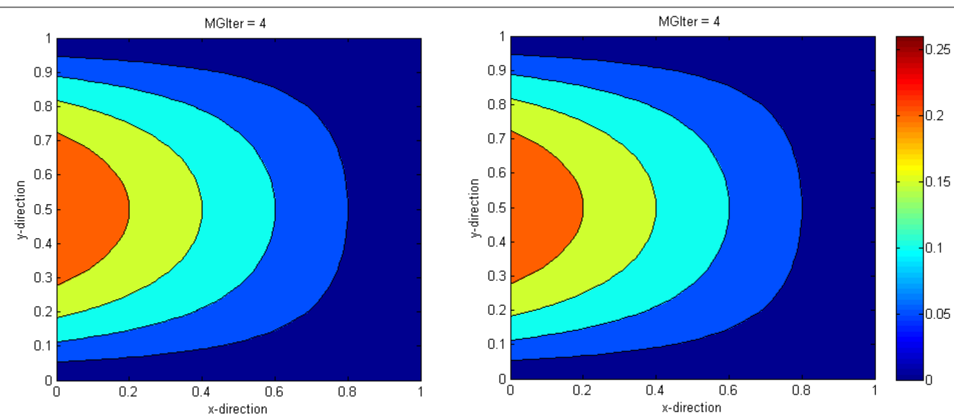
2


MGIter = 2
y-direction
x-direction
MGIter = 2
y-direction
x-direction
0.25
0.2
0.15
0.1
0.05
0

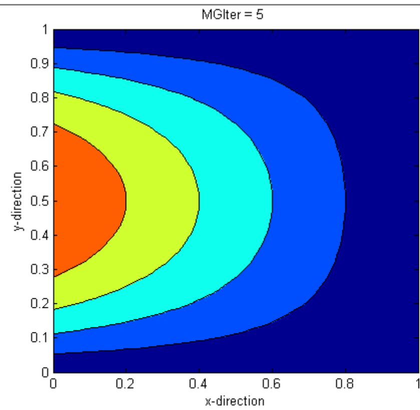
3

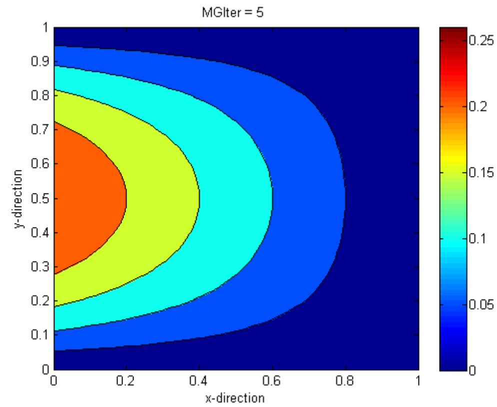

MGIter = 3
y-direction
x-direction
MGIter = 3
y-direction
x-direction
0.25
0.2
0.15
0.1
0.05
0

4

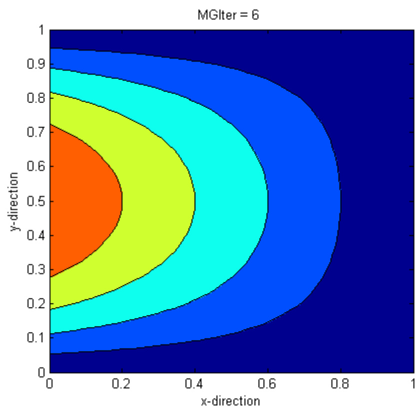

MGIter = 4
y-direction
x-direction
MGIter = 4
y-direction
x-direction
0.25
0.2
0.15
0.1
0.05
0

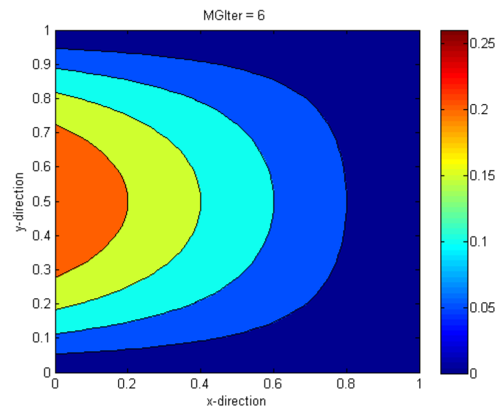
5


MGIter = 5
x-direction
y-direction

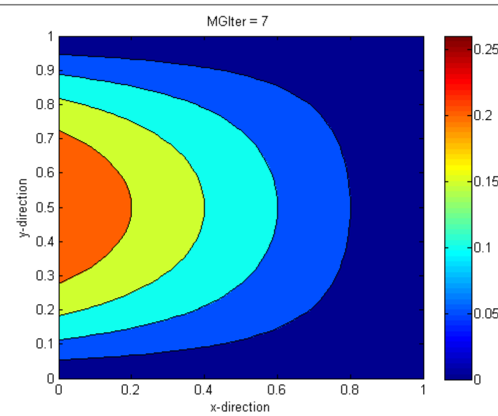

MGIter = 5
x-direction
y-direction

6


MGIter = 6
x-direction
y-direction


MGIter = 6
x-direction
y-direction

7


MGIter = 7
x-direction
y-direction

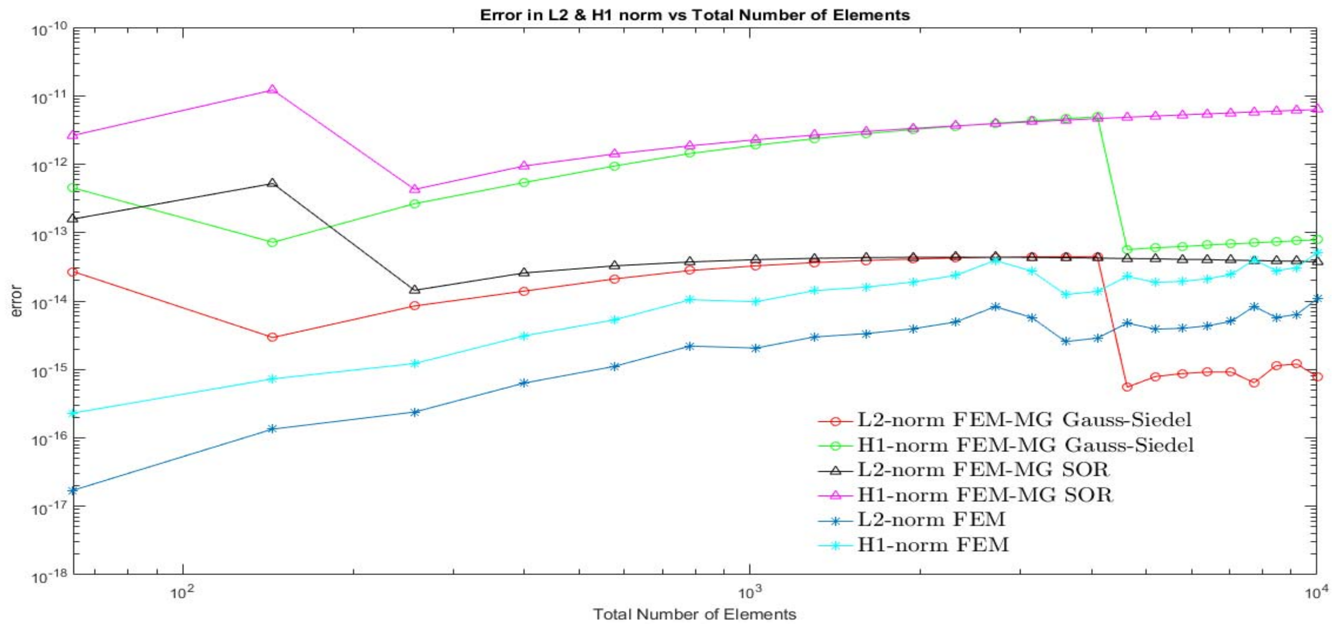


Figure 3.2: Error in L2 & H1 norm vs Total Number of elements for case 2

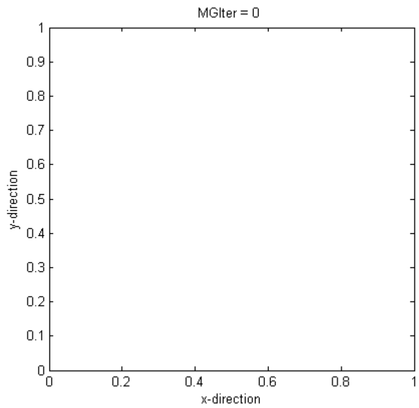
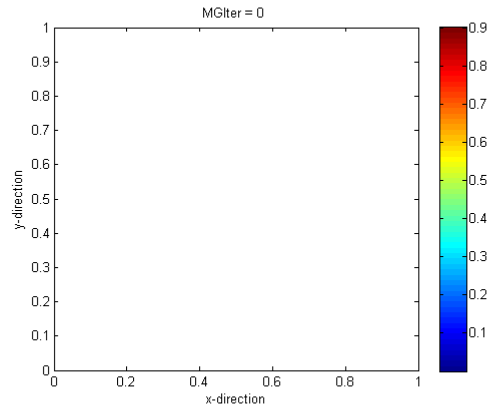
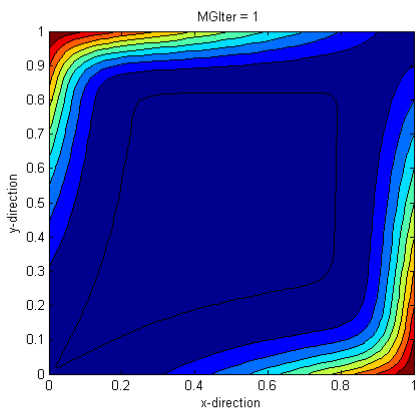
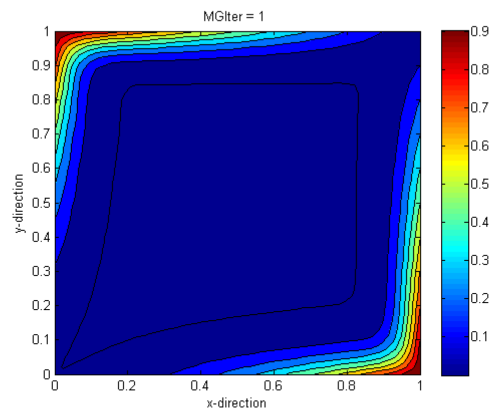
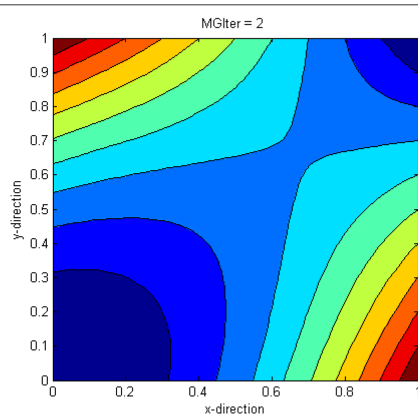
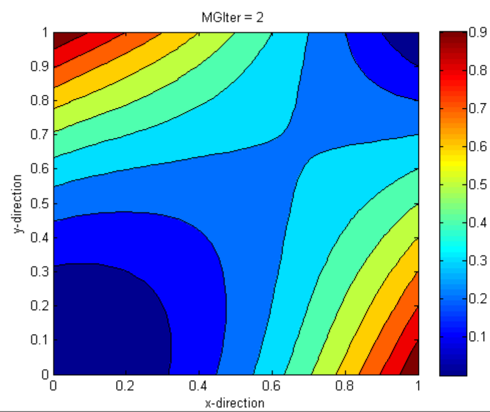
Based on figure 3.2, the FEM-MG Gauss-Seidel also give the least error compare to FEM-MG SOR by 4626.7% differences and FEM by 1273.8% differences.

3.3 Case 3

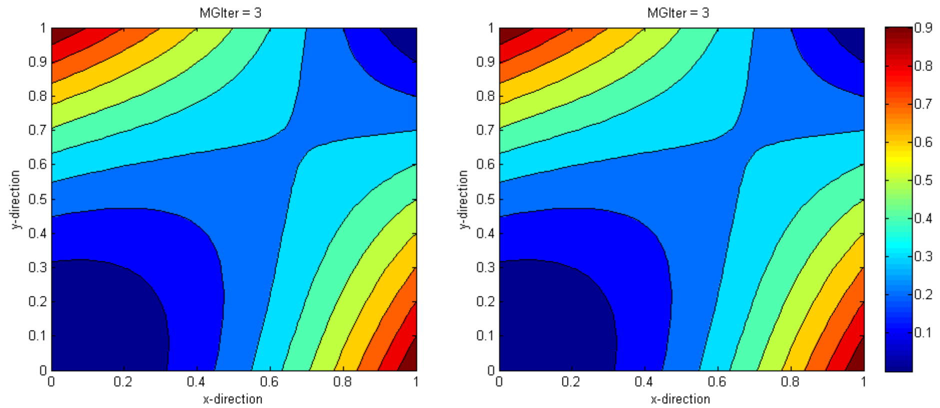
Case 3: $$f(x, y) = -[4 - 2y - 2x]$$

The contour plot at table 3.3, show that the differences occur at MG iteration 1 which the size of contour for the middle which FEM-MG SOR give bigger area compare to FEM-MG Gauss-Seidel.

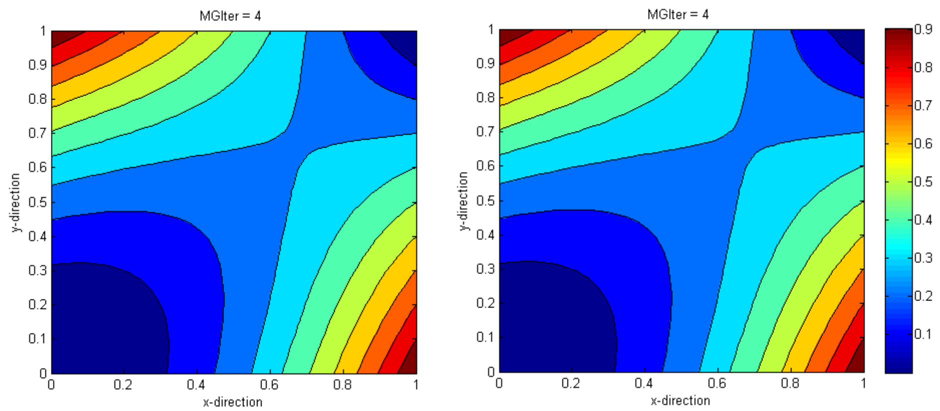
Table 3.3: Comparison Multigrid Finite Element by using Gauss-Seidel and Successive Over Relaxation(SOR), Grid size of 44 for case 3

Multigrid iteration	**Multigrid Finite Element by Using Gauss-Seidel as smoother**	**Multigrid Finite Element by Using Successive Over Relaxation (SOR) as smoother**
0		
1		
2		

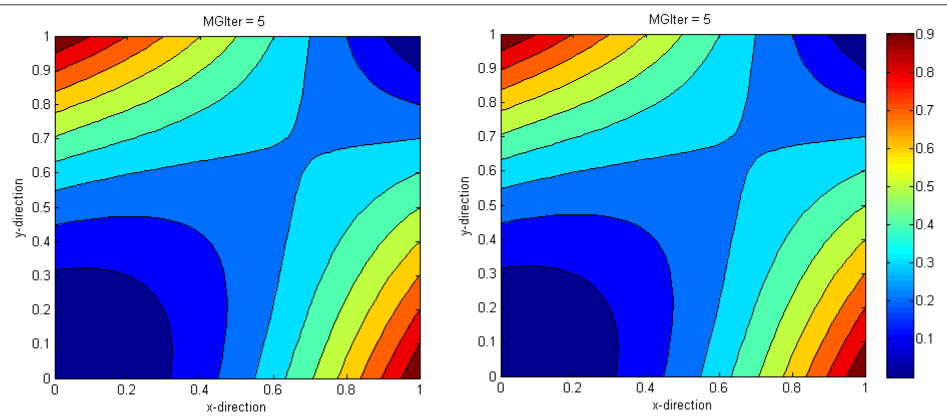
3


MGIter = 3
y-direction
x-direction
MGIter = 3
y-direction
x-direction

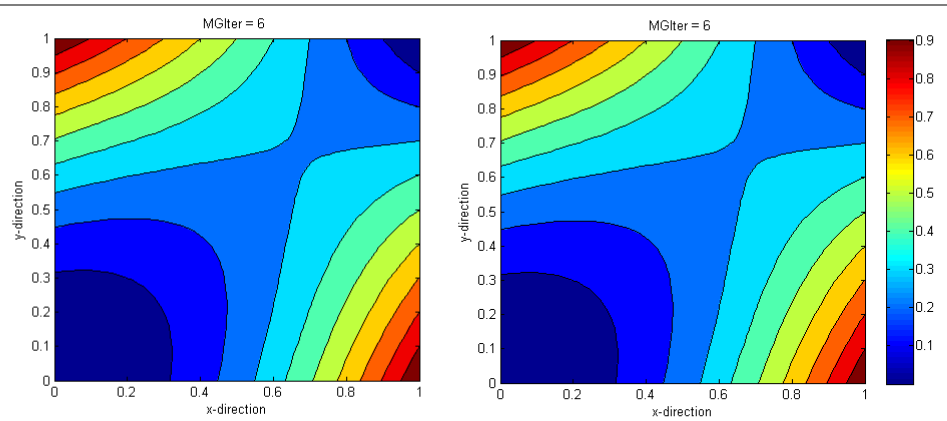
4

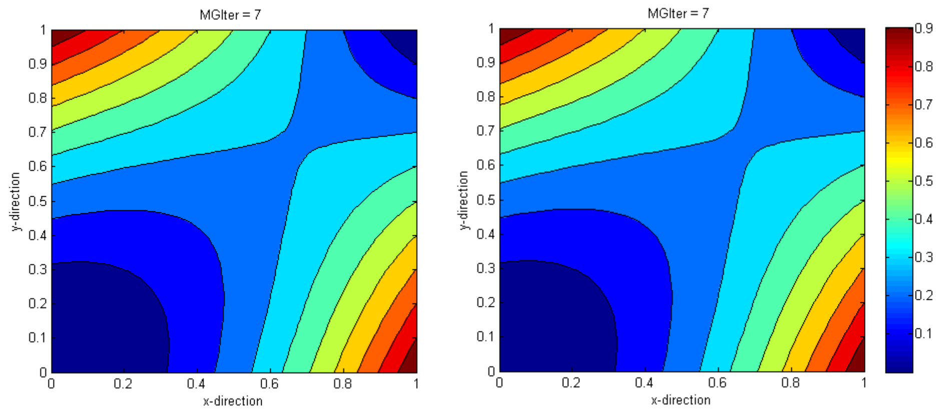

MGIter = 4
y-direction
x-direction
MGIter = 4
y-direction
x-direction

5


MGIter = 5
y-direction
x-direction
MGIter = 5
y-direction
x-direction

6


MGIter = 6
y-direction
x-direction
MGIter = 6
y-direction
x-direction

7	

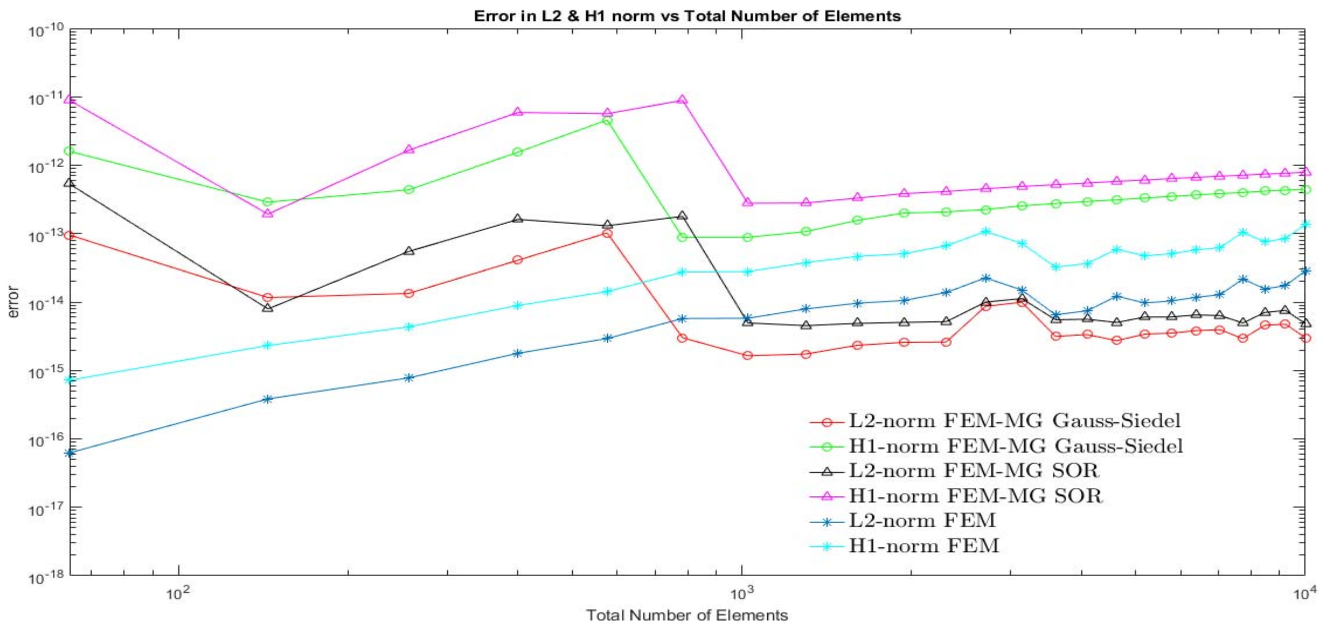


Figure 3.3: Error in L2 & H1 norm vs Total Number of elements for case 3

As shown in figure 3.3, once again the FEM-MG Gauss-Seidel give the least error compare to FEM-MG SOR by 60.63% differences and FEM by 847.39% differences.