ECE 7810 ASSIGNMENT REPORT

(Solution of Fields by Numerical Mtds I)

Author: Jamiu Babatunde Mojolagbe

Department: Electrical and Computer Engineering

Student ID: #7804719

Email: mojolagm@myumanitoba.ca

Course: ECE 7810

Homework: 2

Sub. Date: November 3, 2016

CODE ORGANIZATION AND DETAILS

The main codes are located in the homework folder but there are sub-folders in there. These are:

1. "**data**" folder: It contains the given matrices Q, R and T used in the programming of high order solution of the domain.
2. "**geo_files**" folder: It contains the geometry files of the domain used in this homework
3. "**mesh_files**" folder: This folder holds the meshed files used for this homework.

Functions or Methods Used

The functions used in this homework are divided into two groups and are as follows:

1. Provided functions:
    a. **GmshReadM.m**: This is for reading the generated .gmsh files from Gmsh application.
    b. **getEleIndices.m**: Its function is to get the element(s) for which its/their xy-coordinate is/are specified in the input parameter(s) to the function.
    c. **TRIangles.m**: This obtains the angles in a given triangle once the x and y co-ordinates of the three points in the triangle as supplied as parameters to the function. The function was obtained from [1].
2. Developed Functions:
    a. **getDensity.m**: This function returns the charge density of point once the x and y co-ordinates of the points are supplied as parameters to the function. It takes three parameters, x, y and type of charge distribution expected for the chosen domain.
    b. **getElementIndices.m**: This function returns the element IDs of the elements that shared a given node (supplied as argument/parameter to the function).
    c. **solveAnalytic.m:** It solves a given 2-D domain which MeshData (which is returned from GmshReadM.m) is passed to it using ANALYTICAL method.
    d. **solveFirstOrder.m**: As the name suggests, it solves a given 2-D domain passed to it as MeshData using numerical method with first order triangular elements.
    e. **solveSecondOrder.m**: As the name implies, it solves a 2-D given domain passed to it as MeshData using numerical method with second order triangular elements.

Domain/Geometry and Mesh Files Used

For the purpose of this homework, two different domains/geometry were considered and used for the purpose of this report. As previously explained the geometry files are found in the **geo_files** folder and mesh files are located in the **mesh_files** folder. And in every case, a geometry file has corresponding mesh file. The domains are:

1. "sphere"
2. "flat_plate"

Note: For the "flat_plate" geometry, the generated mesh files are three (3); "flat_plate", "flat_plate_0.1" and "flat_plate_0.02". The suffix numbers in front of the names indicates average triangle/mesh size.

QUESTION A

Implementation Note and Code Explanation

As previously explained, the Finite-Element program for solving a general 2-D electrostatic field in a closed domain with inhomogeneous dielectric ε(x,y) and arbitrary charge distribution ρ(x, y), which is Poisson's Problem, was implemented as a function using first order triangular elements with name **solveFirstOrder.m**. The details of the function as documented in the code is as follows:

```
function [phi] = solveFirstOrder(MeshData,V1, rhoType, epsilon, sides)
%solveFirstOrder.m
%   Computes scalar potential,phi(x,y) on a rectangular grid
%   using First Order Solution (Poisson's Problem)
%
%       Parameters
%           MeshData:   Returned object from ReadGMSH function
%           V1:         Applied potential
%           rhoType:    Source distribution type
%           epsilon:    Relative permittivity of the medium
%           sides:      Boundaries line physics
%
%       Returns
%           phi:        Potential distribution as column vector
```

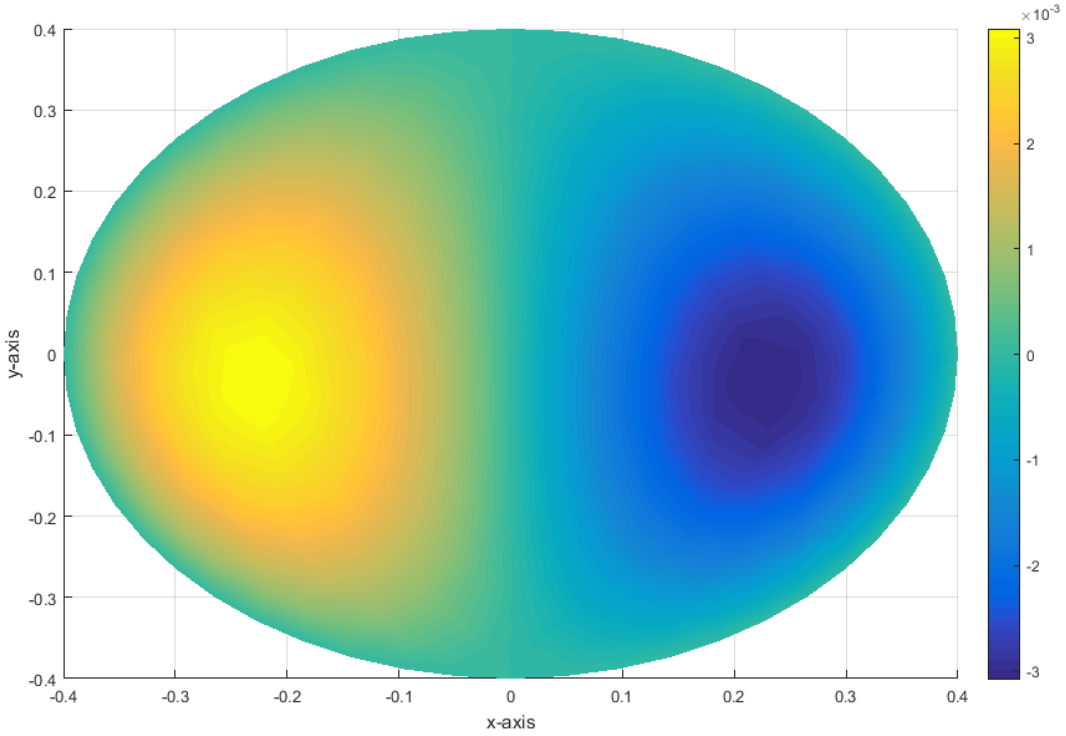
However, for the purpose of this question, this function was implemented in the file named **sphericalDomain.m**. The domain passed to this function as the name of the implementation file suggests, is a spherical domain with four (4) different distinct line physics for applying Dirichilet Boundary Condition to the domain. The passed parameters to the function depicted above in the implementation file is as follows:

```
% line physics for the domain (probe point comes last)
sides = [101 104 103 102];
% apply potential to the probe point
V1 = 100;
% set the type of excitation used (1-5)
rhoType = 5;
% dielectric constant of the domain
epsilon = 1.0;
% Use GmshreadM to read the Gmsh mesh file
MeshData = GmshReadM('mesh_files/sphere.msh');
```

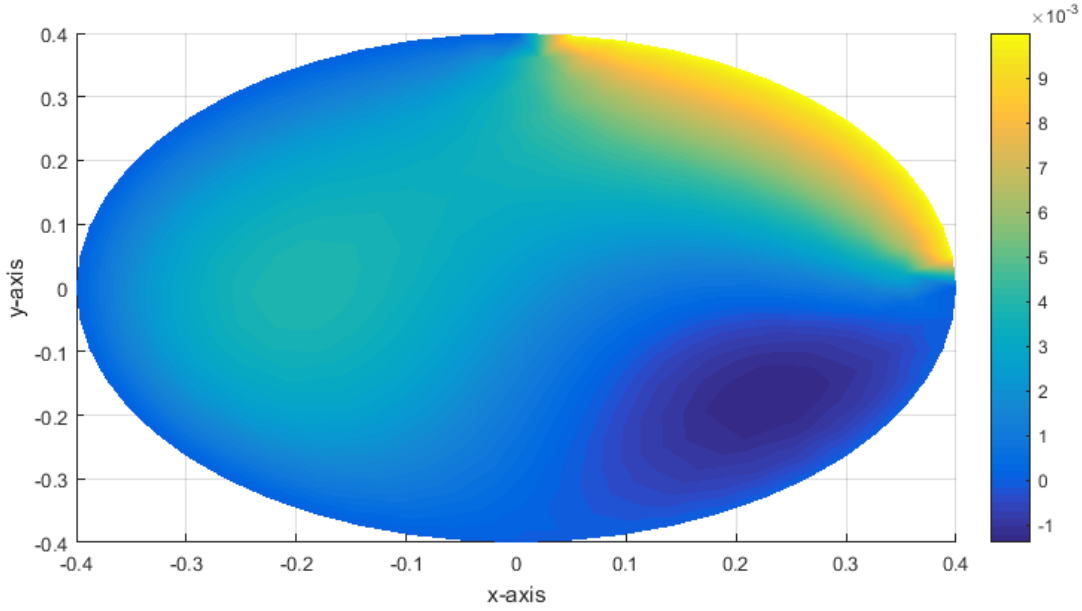
Results and Discussions

Three instances of results were recorded, firstly, when no potential is applied to the domain, when very small value of potential was applied (0.01V) and thirdly, when the applied potential is 100V.

Case 1: The following distribution was obtained after the spherical domain was solved with the developed function with all the sides of the domain grounded:

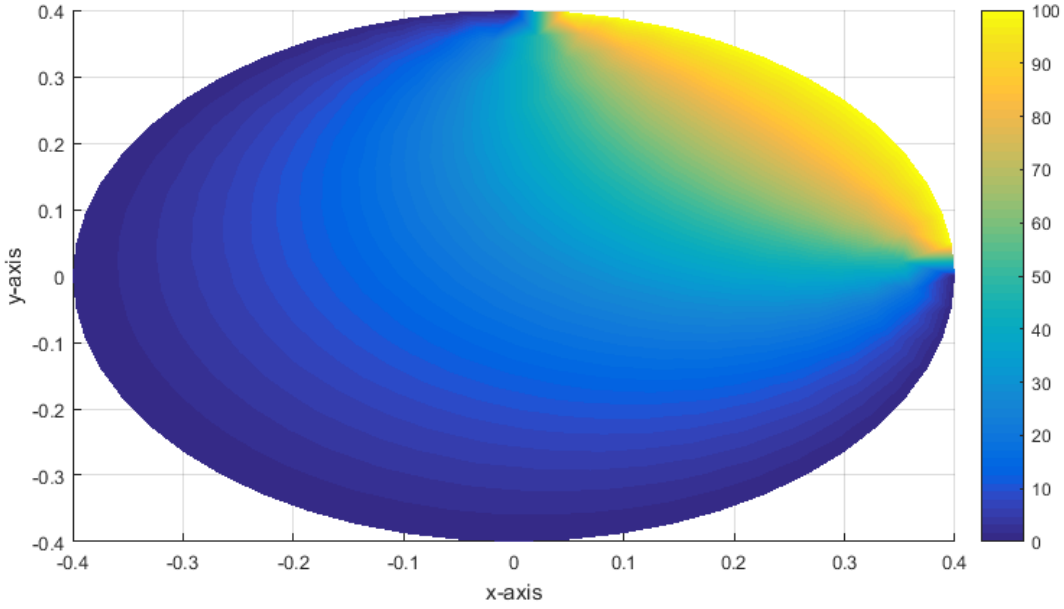


Case 2: With Dirichlet B.C set to very small value (0.01V) on one side and other sides grounded, the following distribution was obtained from the developed function after the domain was solved.



Course: ECE 7810 (Solution of Fields By Numerical Methods I)
Homework: 2
Author: Jamiu Babatunde Mojolagbe (#7804719)

Case 3: With Dirichlet B.C set to high value (100V) on one side and other sides grounded, the following distribution was obtained from the developed function after the domain was solved.



Apart from the graphical output shown above, the potential and electric field on each of the global nodes are also calculated and returned.

Observation and Conclusion

It can be seen/observed from the resulting graphs presented above that the domain is a perfect Poisson's problem as there was source of excitation in it even when all its sides are grounded that is, Case 1.

Again, when the three of its sides were grounded, and a very small amount of potential applied to one side, the applied potential tends to move from the boundary to the other parts. However, the effect of the source of excitation was still prominent and can be seen from Case 2 above.

Finally, with the applied potential on the boundary now set to be very high (100V in Case 3), the effect of the source of excitation was not that prominent, though it affected the overall result, and the potential was seen to be moving from the boundary of application to the other parts of the domain.

QUESTION B

Implementation Note and Code Explanation

As explained in the previous question, the same function (**solveFirstOrder.m**) was used to solve this question numerically. However, the domain/geometry used for this problem is a square flat plate whose analytical solution is known. Therefore, the analytical solution of this problem was implemented in a function named (**solveAnalytic.m**) as explained in the code description previously presented. The parameters and return values as contained in the function file is shown below:

```
function [phi] = solveAnalytic(width, height, MeshData,V1, maxIter, side)
%% solveAnalytic
%   Computes scalar potential,phi(x,y) on a rectangular grid
%   using Analytical Solution (Poisson's Problem)
%
%       Parameters
%           width:      Width of the domain
%           height:     Height of the domain
%           MeshData:   Returned object from ReadGMSH function
%           V1:         Applied potential
%           maxIter:    Maximum number of iterations for fourier expansion
%           side:       Boundaries line physics
%
%       Returns
%           phi:        Potential distribution as column vector
```

Hence, the driver program for this function is called **flatPlateAnalytic.m**. The passed parameters in the driver program is shown below:

```
% line physics for the domain (probe point comes last)
sides = [101 104 103 102];
% apply potential to the probe point
V1 = 100;
% width of the domain
width = 1;
% height of the doamin
height = 1;
% maximum iterations for analytic solution
maxIter = 100;
% Use GmshreadM to read the Gmsh mesh file
MeshData = GmshReadM('mesh_files/flat_plate.msh');
```

The numerical first order driver program for this problem is named "**flatPlateFirstOrder.m**" and the supplied parameters to the function developed for performing the task as contained in the program file is shown below:

```
% line physics for the domain (probe point comes last)
sides = [101 104 103 102];
% apply potential to the probe point
```

```
V1 = 100;
% set the type of excitation used (1-5)
rhoType = 5;

% dielectric constant of the domain
epsilon = 1.0;
% Use GmshreadM to read the Gmsh mesh file
MeshData = GmshReadM('mesh_files/flat_plate.msh');
```

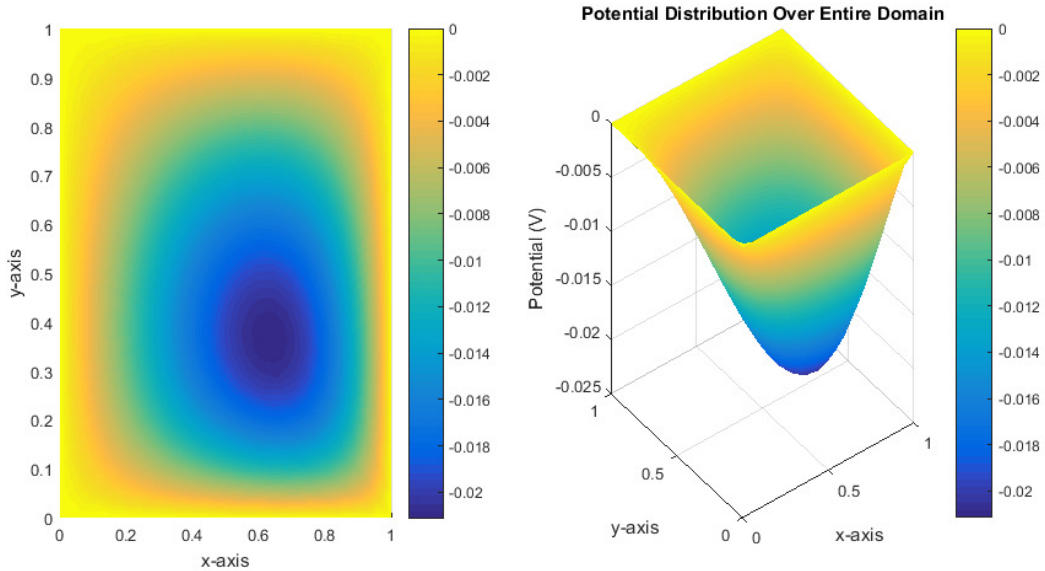
Lastly, comparison between the analytical and first order numerical solutions, tested in the two previous files mentioned and explained above, was implemented in the file named "**flatPlateCompareAnalyticVsFOrder.m**" and appropriate results were obtained and recorded.

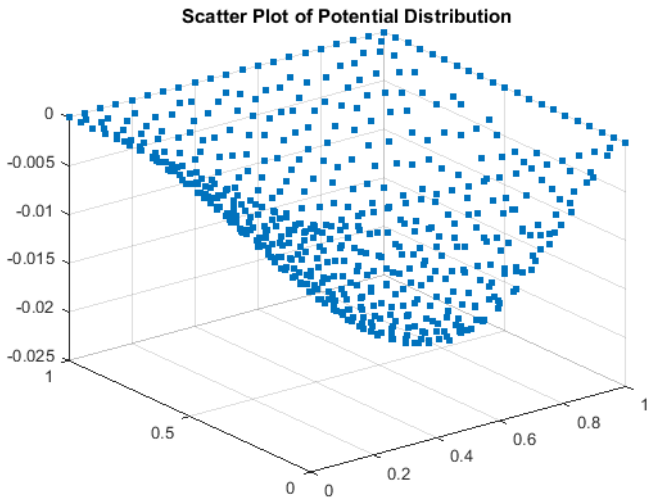
Note: *For this question three distinct mesh sizes were used for this given domain. This implies that there are three (3) mesh files associated with this problem; each mesh file has different average triangle sizes or number of triangular elements. The mesh file "**flat_plate.msh**" has 0.05, "**flat_plate_0.1.msh**" has size 0.1 and "**flat_plate_0.02.msh**" has average triangle size of 0.02. Meaning that "**flat_plate_0.1.msh**" has fewest number of triangular elements followed by "**flat_plate.msh**" while "**flat_plate_0.02.msh**" has the highest number of triangular elements.*

Results and Discussions

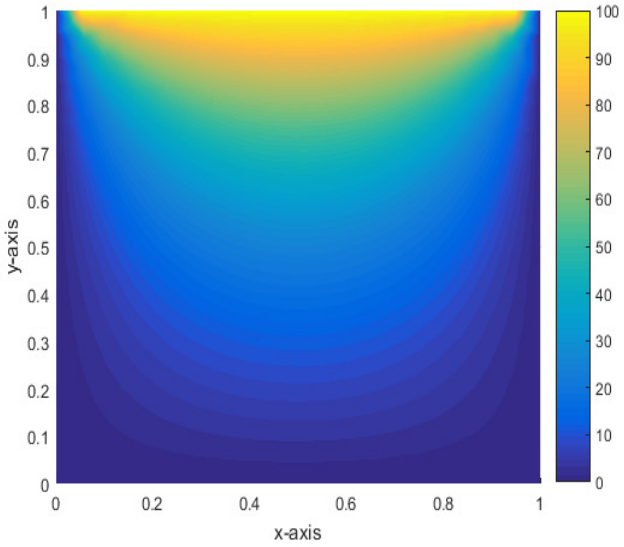
The results of the distribution obtained from the first order numerical solution is shown below:

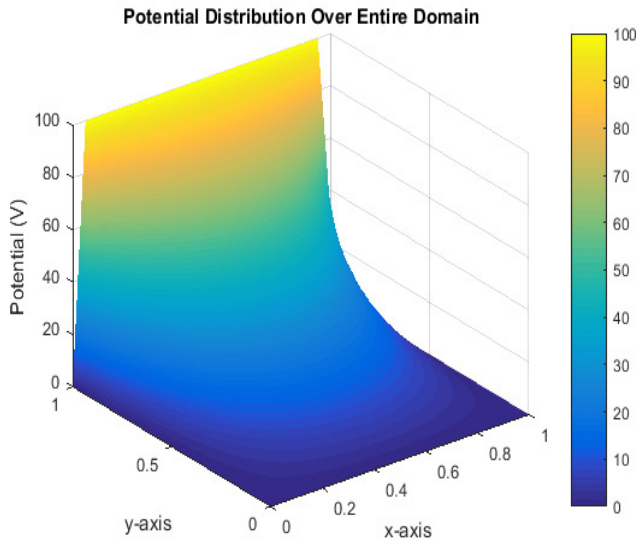
Case 1: When all the sides of the flat plate domain ("**flat_plate.msh**") was grounded the following distribution was obtained:



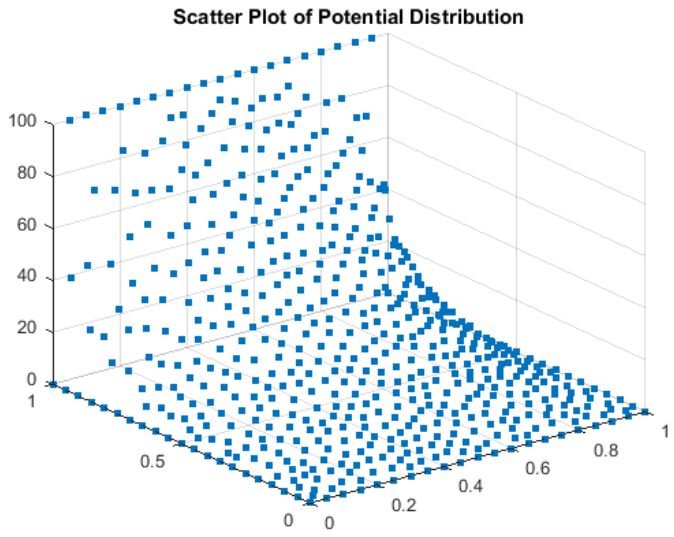


Case 2: When three (3) of the sides of the flat plate domain ("**flat_plate.msh**") was grounded and potential of 100V applied to one side, the following distribution was obtained:

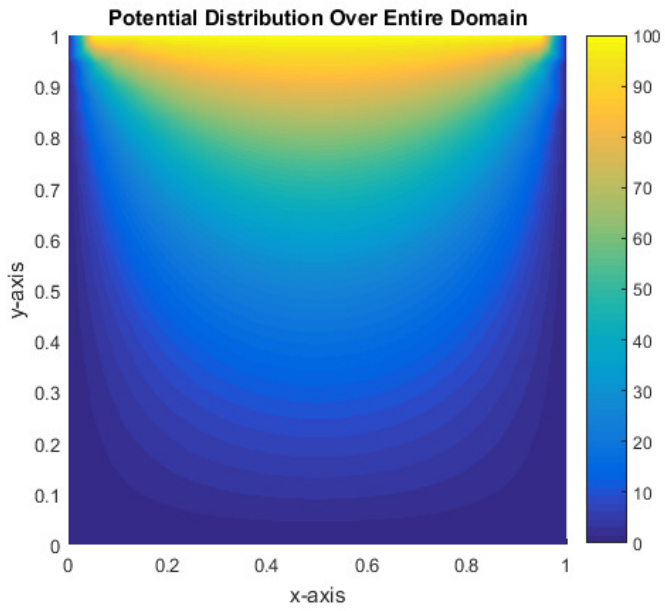


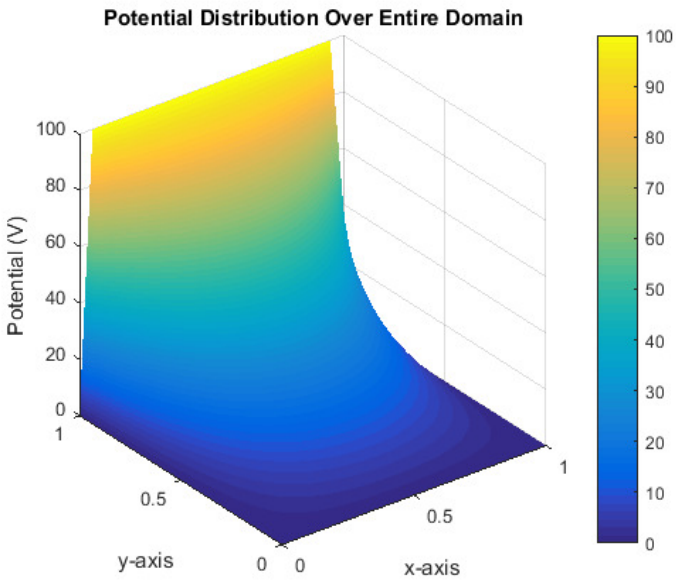


Course: ECE 7810 (Solution of Fields By Numerical Methods I)
Homework: 2
Author: Jamiu Babatunde Mojolagbe (#7804719)



One more thing to note here is that, when the potential is applied to one side of the domain presented above and other sides grounded for the analytical solution the same distribution was observed but the graph seemed to be more evenly distributed than the first order numerical solution distribution by closer inspection, however this does not totally justify inaccuracy or otherwise of the solution but this will be treated in more details later. The obtained graph for the analytical solution is shown below:

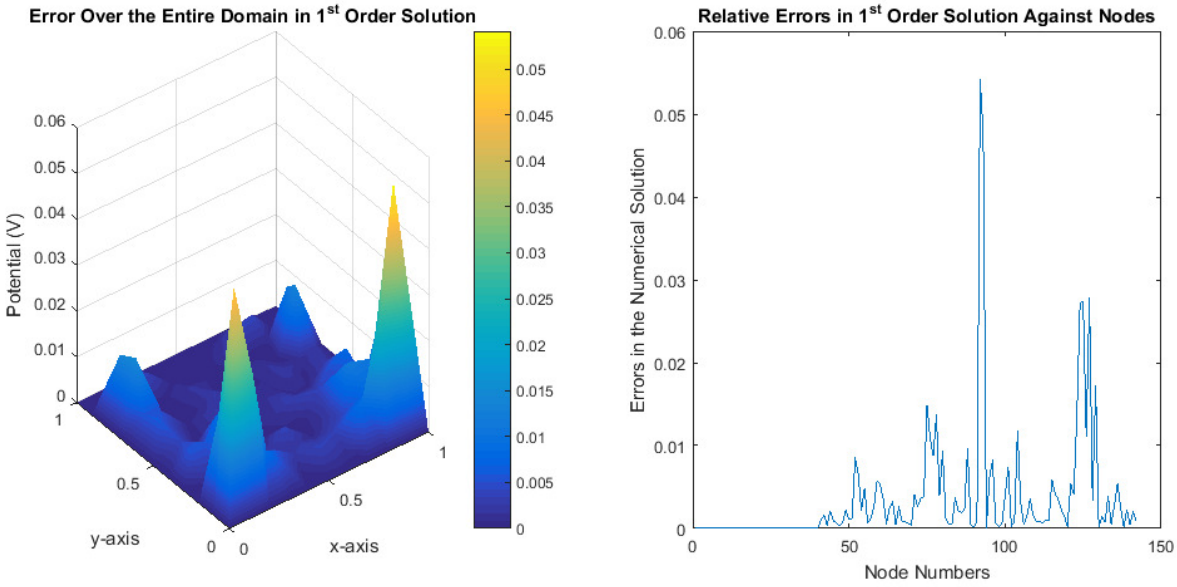




The relative error distribution of the first order solution obtained from comparison between the analytical solution and the first order equivalent over the entire domain is presented below.

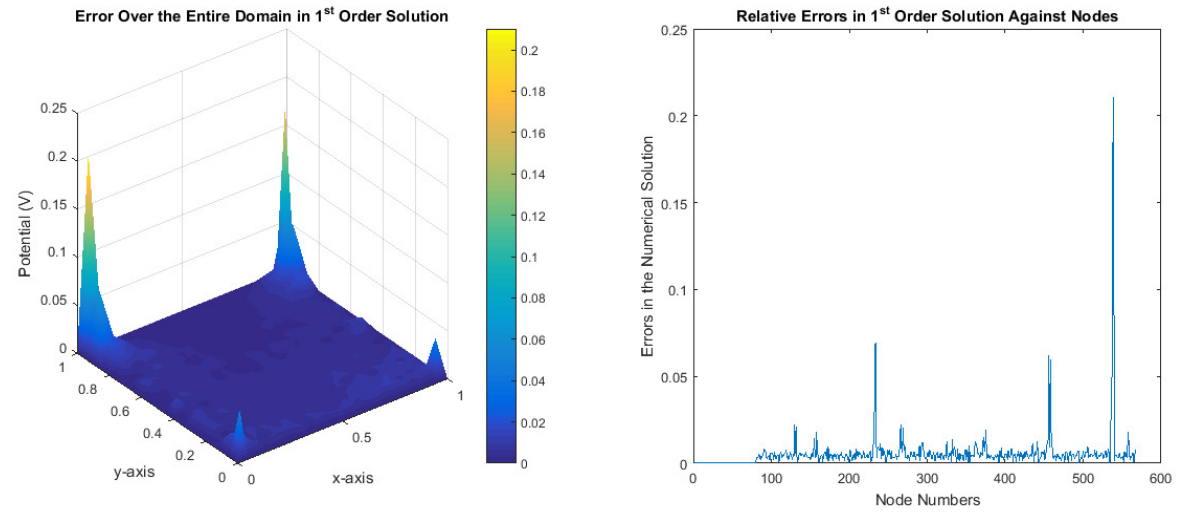
The cases for the consideration are three (3) based on the average triangular elements sizes – in other words, based on the numbers of triangular elements in each discretization as contained in the "Note" above.

Case 1: With mesh "**flat_plate_0.1.msh**" which has 242 triangular elements the following result was obtained:



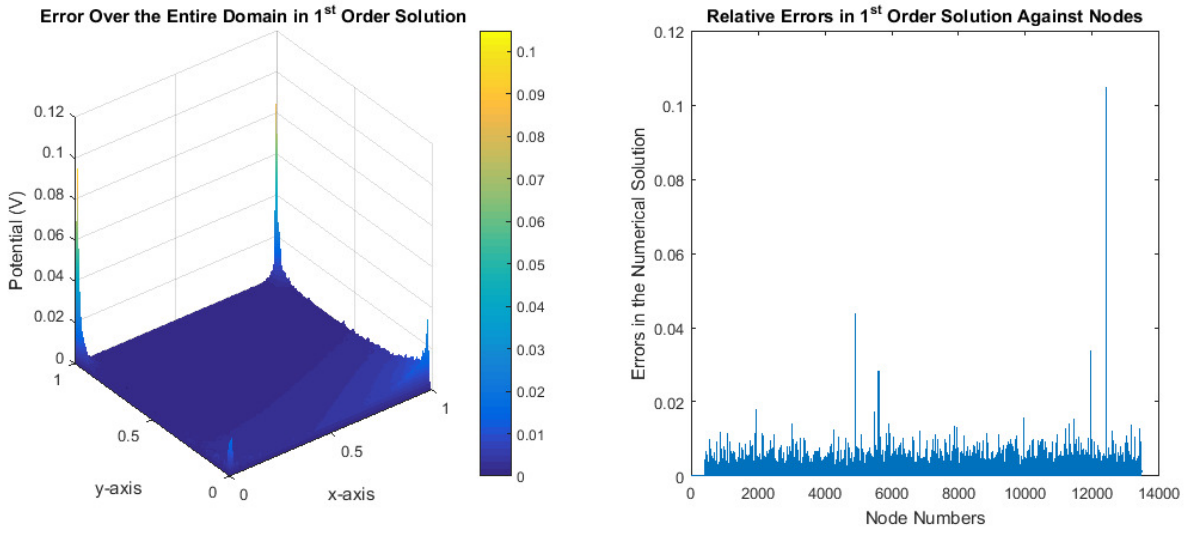
As shown above the maximum relative error is 0.0541 (5.4118%).

Case 2: With mesh "**flat_plate.msh**" which has 1054 triangular elements the following result was obtained:



As shown above the maximum relative error is 0.2102 (21.0177%).

Case 3: With mesh "**flat_plate_0.02.msh**" which has 26528 triangular elements the following result was obtained:



As shown above the maximum relative error is 0.1049 (10.4884%).

Observation and Conclusion

From the results and discussion part it can be observed that with better mesh refinement the distribution tends to be better captured and overall relative error reduced. Though, the maximum error tends to have anomalous behavior as it was lower than the others with better mesh refinements.

It can also be seen/observed from the results presented above that in all the three cases presented above with mesh refinements, the error tends to be more prominent at the four (4) angles of the domain. This error, however can be mitigated by having more meshes or better refinements at those angles because the distribution was not proper captured.

Finally, it can be concluded that with infinite mesh refinement, the numerical solution would be the same as analytical solution (though time to compute and solve the resulting matrices may be a challenge).

QUESTION C

Implementation Note and Code Explanation

As contained in the introductory part, high order implementation of the presented square flat, whose first order solution is explained in Question B above, was implemented with second order elements as a function named "**solveSecondOrder.m**". The input parameters to the function and return values are shown below as contained in the implementation file:

```
function [phi,MeshData]=solveSecondOrder(MeshData,V1,rhoType,epsilon,sides,Qpath, Rpath, Tpath)
%solveFirstOrder.m
%   Computes scalar potential,phi(x,y) on a rectangular grid
%   using Second Order Solution (Poisson's Problem)
%
%       Parameters
%           MeshData:   Returned object from ReadGMSH function
%           V1:         Applied potential
%           rhoType:    Source distribution type
%           epsilon:    Relative permittivity of the medium
%           sides:      Boundaries line physics
%           Q|R|Tpath:  Path to the matrices Q, R and T respectively
%
%       Returns
%           phi:    Matrix of potential distribution
```

The numerical second order driver program for this problem is named "**flatPlateSecondOrder.m**" and the supplied parameters to the function developed for performing the task as contained in the program file is shown below:

```
% line physics for the domain (probe point comes last)
sides = [101 104 103 102];
% apply potential to the probe point
V1 = 100;
% set the type of excitation used (1-5)
rhoType = 5;
% dielectric constant of the domain
epsilon = 1.0;
% Use GmshreadM to read the Gmsh mesh file
MeshData = GmshReadM('mesh_files/flat_plate.msh');
```

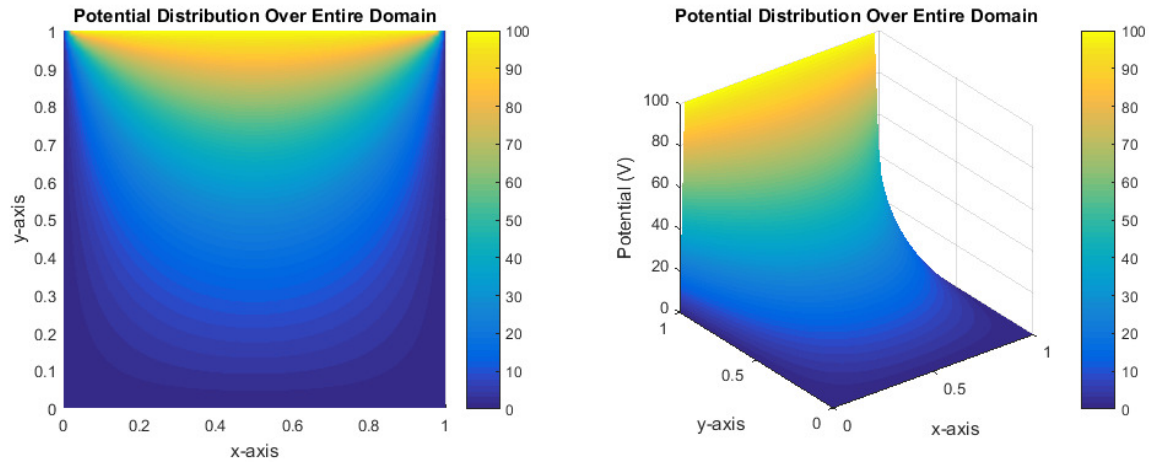
Lastly, comparison between the analytical, first, second order numerical solutions, was implemented in the file named "**flatPlateCompare.m**" and "**flatPlateCompareRandomPoints.m**". Appropriate results were obtained and recorded. In the file named "**flatPlateCompare.m**" the potentials on each of the global nodes over the entire domain were calculated using analytic, first and second order solutions and relative errors between the analytical vs first order solution and analytical vs second order were calculated and plotted. While

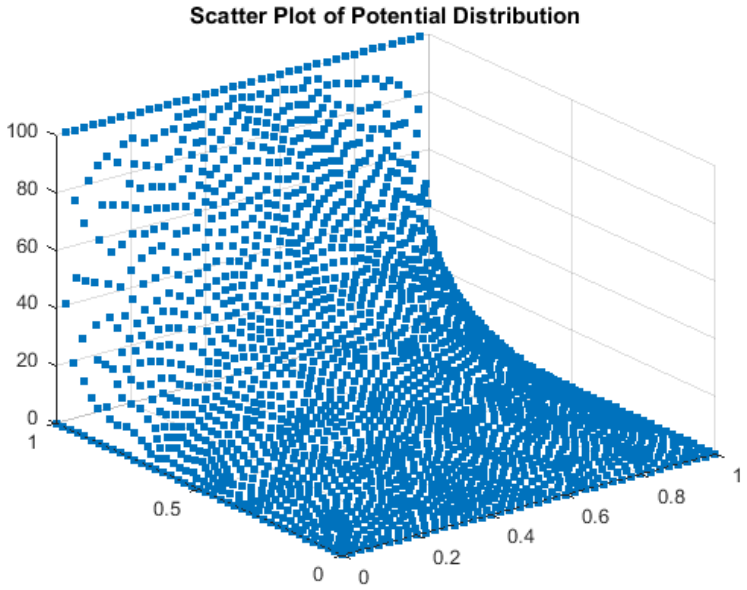
in "**flatPlateCompareRandomPoints.m**" random points are selected on the domain with random point generator and potential on the points are calculated using analytical, first and second order solutions; the relative error on those points are also plotted for the problem.

Note: *For this question three distinct mesh sizes were used for this given domain. This implies that there are three (3) mesh files associated with this problem; each mesh file has different average triangle sizes or number of triangular elements. The mesh file "**flat_plate.msh**" has 0.05, "**flat_plate_0.1.msh**" has size 0.1 and "**flat_plate_0.02.msh**" has average triangle size of 0.02. Meaning that "**flat_plate_0.1.msh**" has fewest number of triangular elements followed by "**flat_plate.msh**" while "**flat_plate_0.02.msh**" has the highest number of triangular elements.*

Results and Discussions

The results of the distribution obtained from the second order numerical solution is shown below:

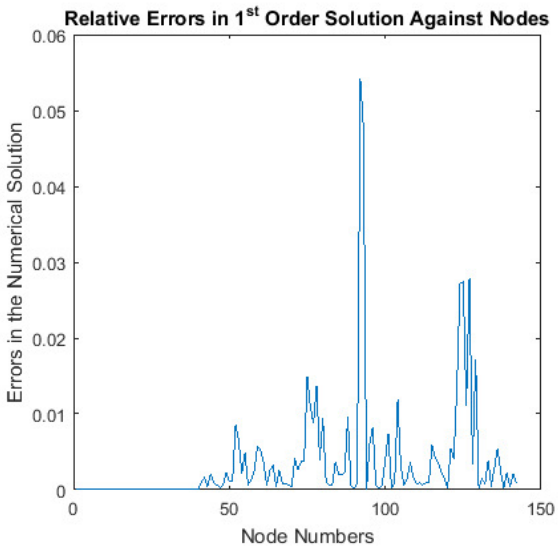


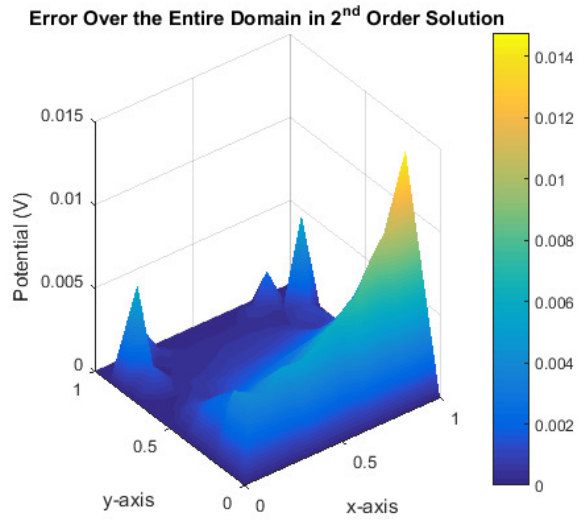


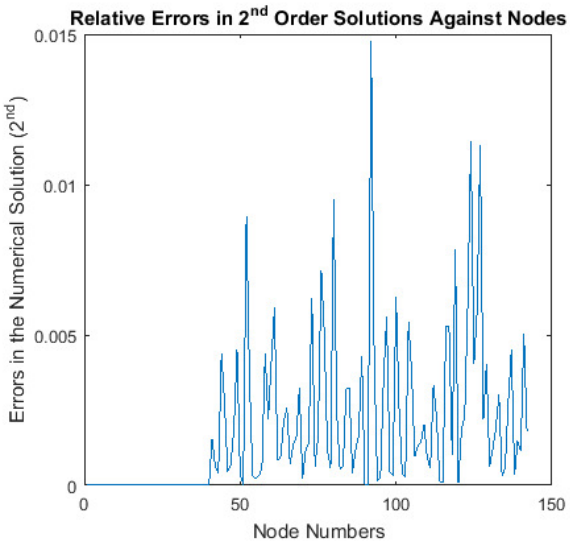
The relative error comparison between analytic solution and numerical solutions (first and second order) were presented for each of the three meshes presented in the first order solution above.

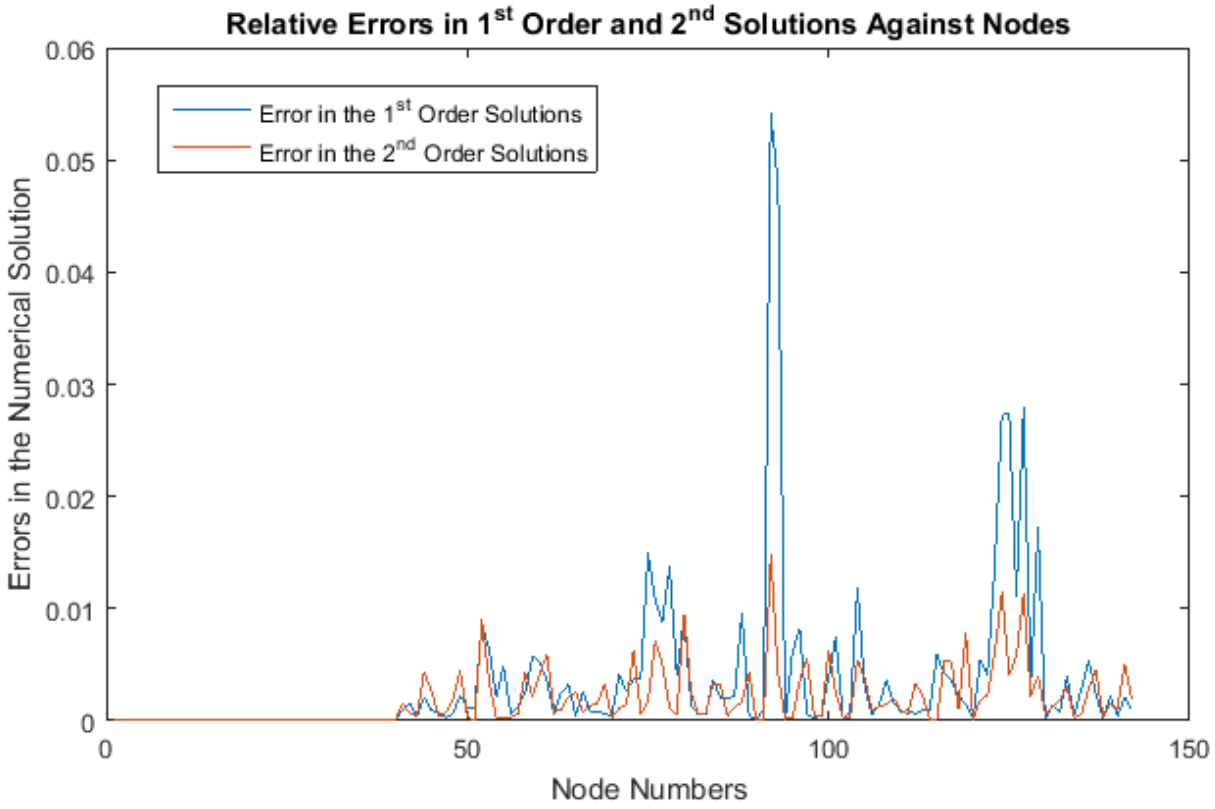
Comparing Potentials Values of Each Node over the Entire Domain

Case 1: With mesh "**flat_plate_0.1.msh**" which has 242 triangular elements the following result was obtained:

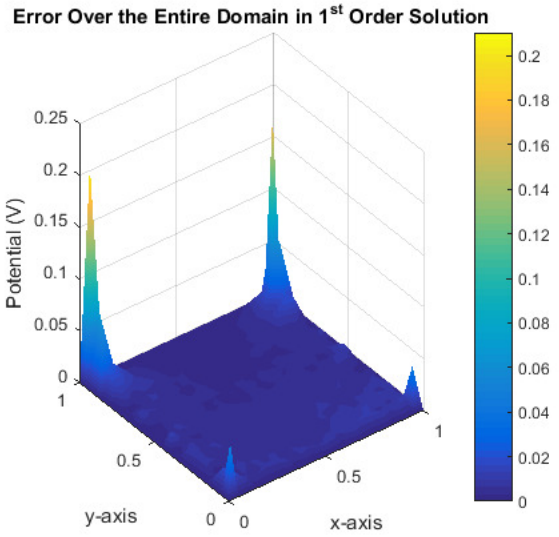


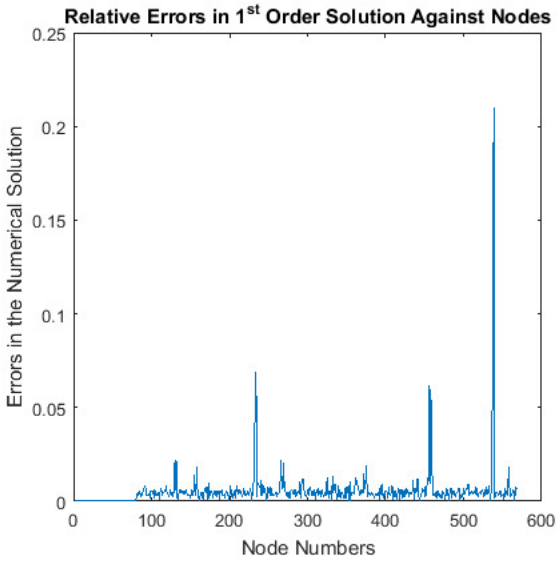


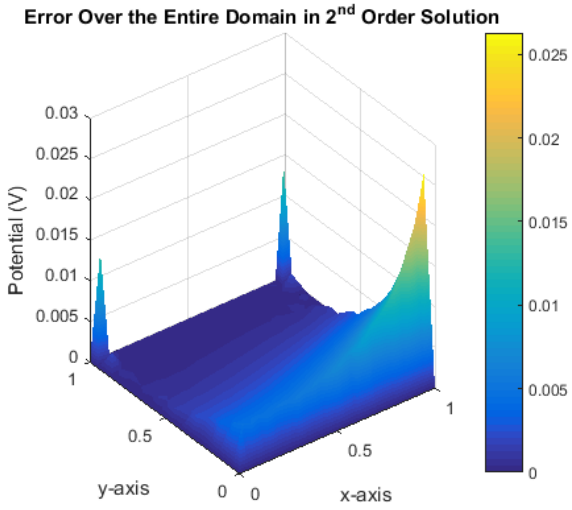


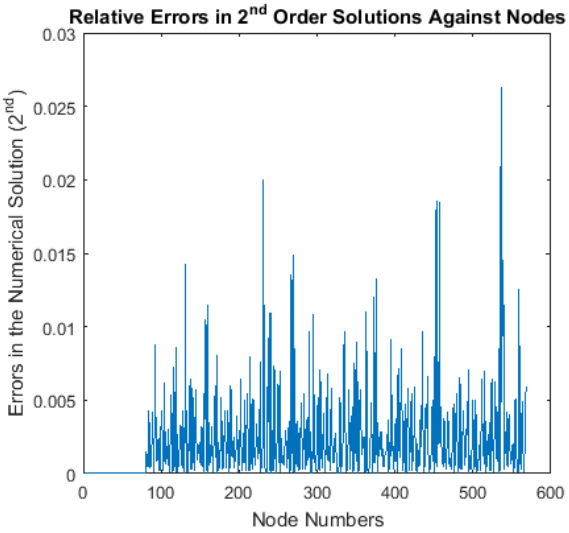


Case 2: With mesh "**flat_plate.msh**" which has 1054 triangular elements the following result was obtained:

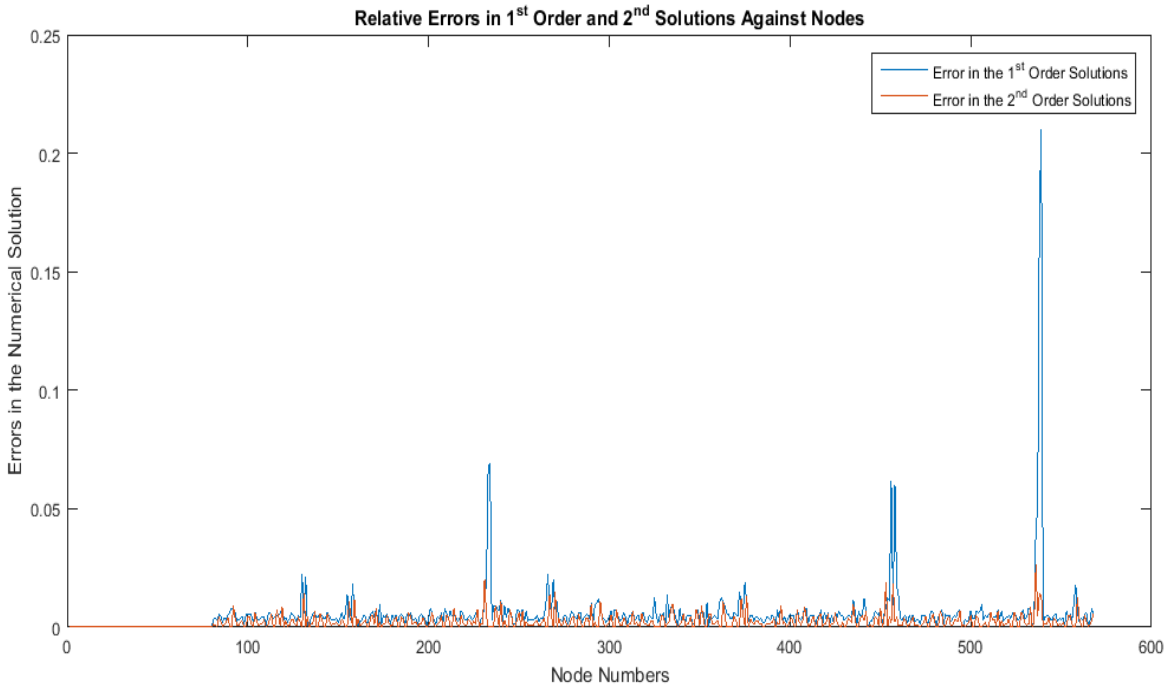




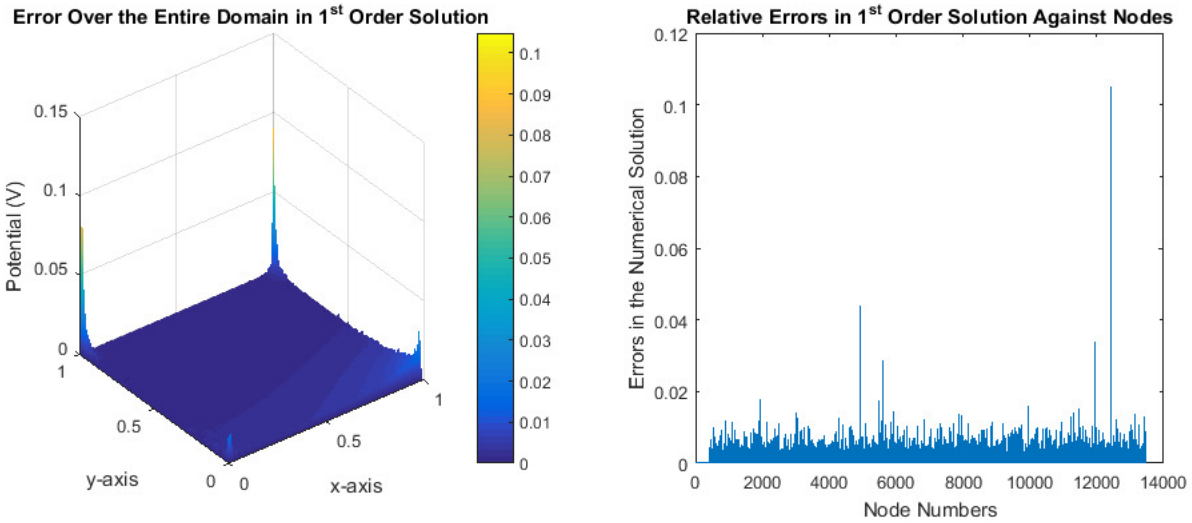




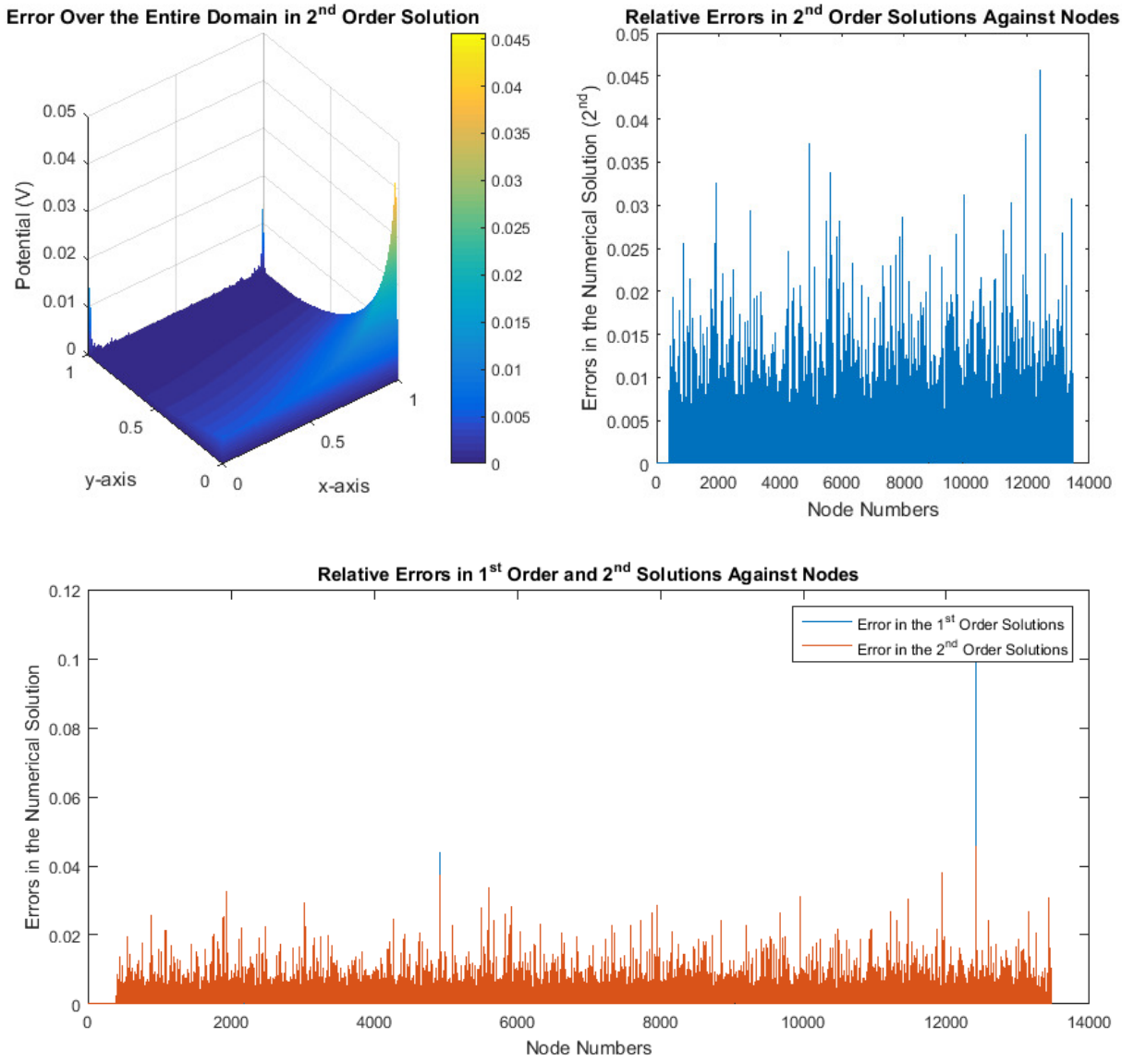
Course: ECE 7810 (Solution of Fields By Numerical Methods I)
Homework: 2
Author: Jamiu Babatunde Mojolagbe (#7804719)



Case 3: With mesh "**flat_plate_0.02.msh**" which has 26528 triangular elements the following result was obtained:

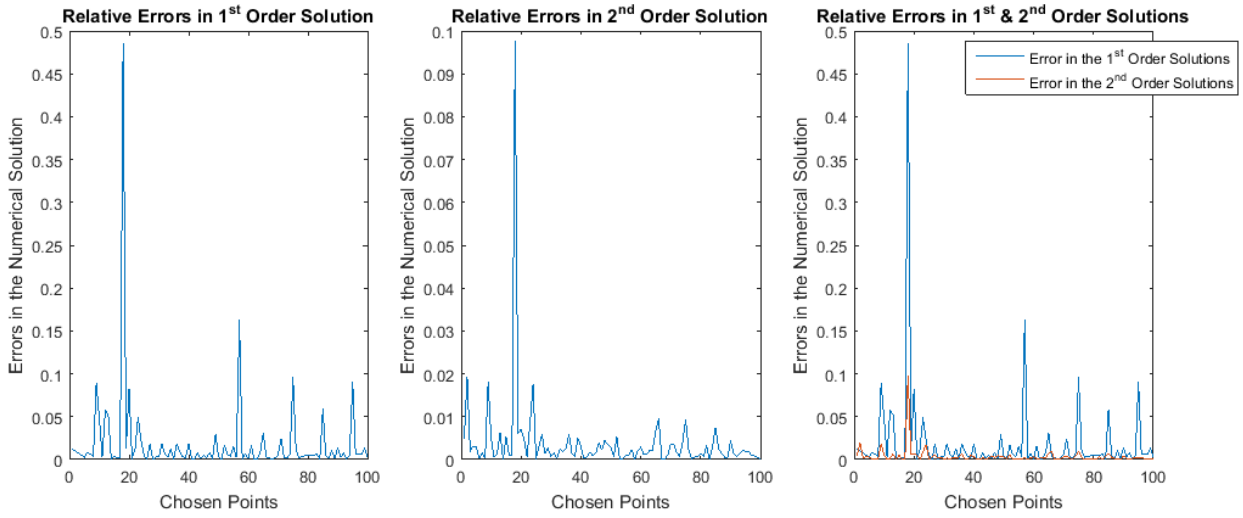


Course: ECE 7810 (Solution of Fields By Numerical Methods I)
Homework: 2
Author: Jamiu Babatunde Mojolagbe (#7804719)

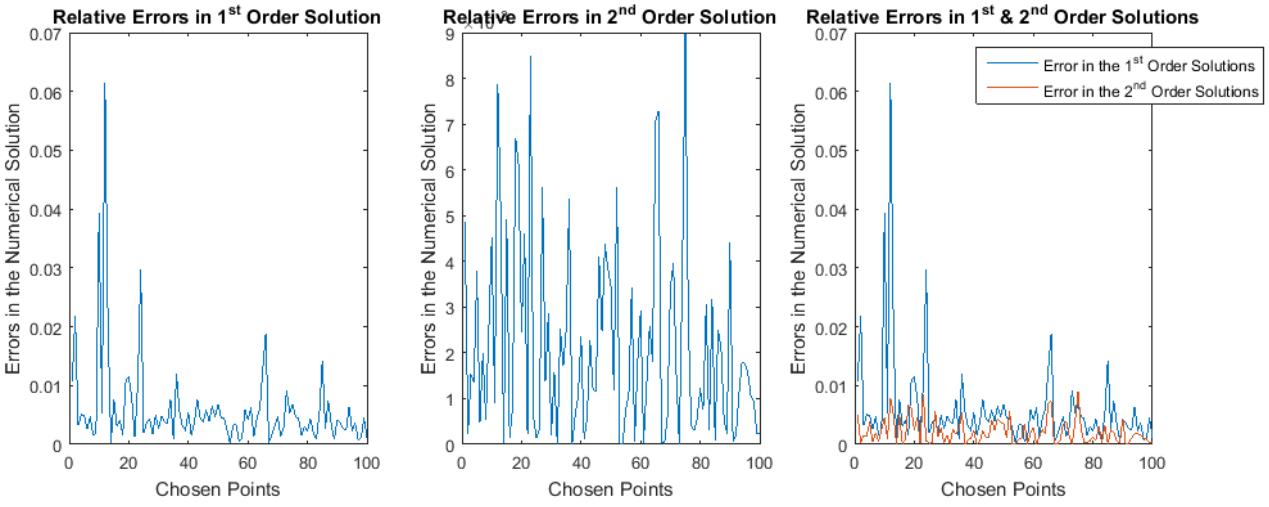


Comparing Potential Values for Random Points on the Domain

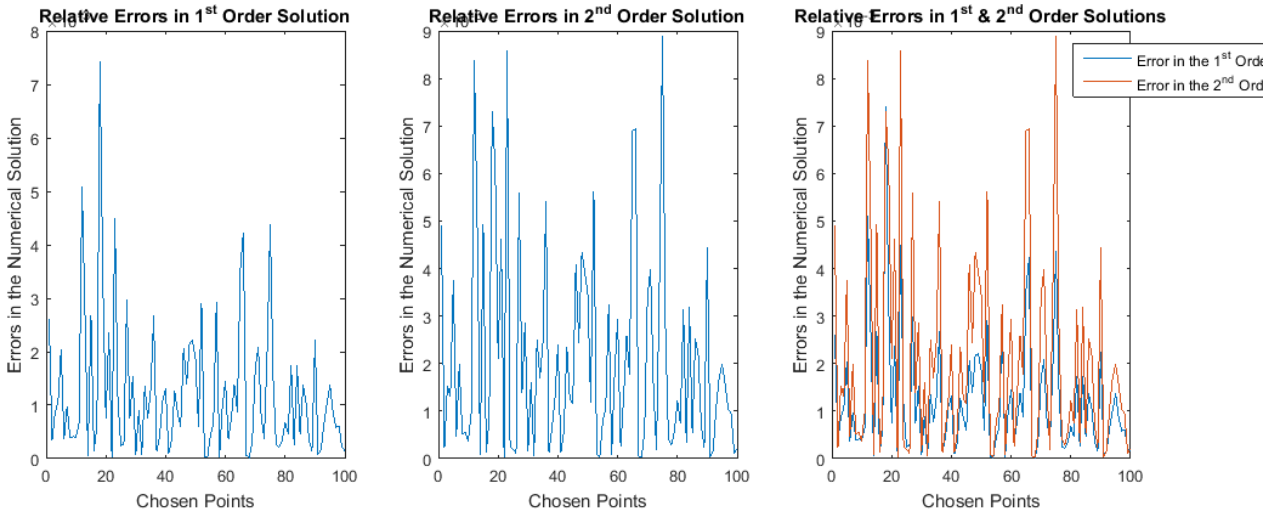
Case 1: With mesh "**flat_plate_0.1.msh**" which has 242 triangular elements the following result was obtained:



Case 2: With mesh "**flat_plate.msh**" which has 1054 triangular elements the following result was obtained:



Case 3: With mesh "**flat_plate_0.02.msh**" which has 26528 triangular elements the following result was obtained:



Observation and Conclusion

From the foregoing, it can be observed that

1. The four angles of the flat plate domain has the highest errors as previously noted
2. With mesh better refinements, even the first order solution tends to capture the distribution better though less accurate compare to the corresponding second order solution. Again, mesh refinements may not be a better option as the input file becomes heavy and matrix size becomes larger which will take considerable amount of time to assemble and finally solve the resulting matrices.
3. In all the presented cased above, the high order solution give highly accurate result compare to the corresponding first order solution; in fact the reduction in the relative error is fantastic given the same number of input triangular elements.

In conclusion, the high order solution of a problem is better in terms of accuracy and time taken to solve the problem; this produces more effective results than mesh refinements which may take considerable amount of time to solve the resulting large matrices. Again it must also be stated that implementing high order may be a little bit challenging but the result is quite more rewarding than the energy expended in the implementation.

References

1. http://www.mathworks.com/matlabcentral/fileexchange/37389-find-angles
2. M. N. O. Sadiku, "Numerical Techniques in Electromagnetics with Matlab," CRC Press, 2009.