On verification of FEM solution to Poisson equation by comparing it to the analytical solution shown in Timoshenko MAE 207, Computational methods. UCI Spring 2006

Nasser M. Abbasi

May 28, 2006 Compiled on September 4, 2021 at 7:56pm

Contents

1 Introduction

A Matlab function was written which plots the analytical solution given by Timoshenko/-Goodier on pages 310-311 in the `Theory of Elasticity`. The analytical solution was compared to the solution generated from the FEM solution. See this page for more information and background on the problem and the analytical solution.

The absolute and percentage differences between the solutions was obtained and compared.

The matlab code used was provided thanks to Qing Wang which solves the problem by FEM. It was called to obtain the FEM solution. Minor changes made in the code to allow one to call it as a function and to use the same contour levels. The appendix contains the Matlab code.

2 Conclusion

The solution by FEM agrees to a very good approximation with the analytical solution.

21 by 41 nodes were used for FEM, we see that, in absolute value, the maximum difference was 0.00009481788675562430 At nodes (11, 10) with symmetry at the other half of the cross section as shown in the plot below (The plot is here on a separate page)

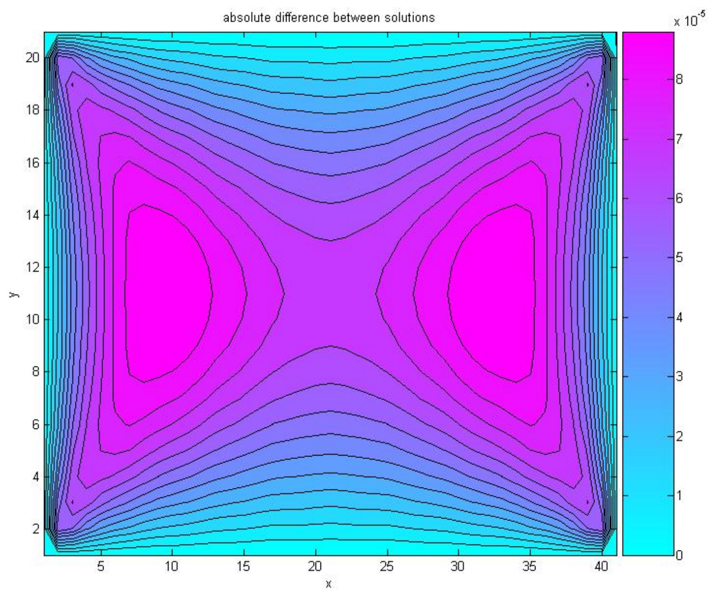


Figure 1: Solution curves

By making the grid smaller, better approximation can be obtained. This analysis was done using 21 by 41 nodes nodes for the rectangular cross section. More elements should result in better approximation.

In terms of percentage differences, the maximum percentage difference occurred at the 4 corners.

Given the above grid size, we see from the plots that there is a maximum of 1% difference between the analytical solution and the FEM solution. This occurred near the 4 corners of the cross section and was smallest in the middle. We need to better investigate why this is.

This below is a plot showing the percentage difference between both solutions.

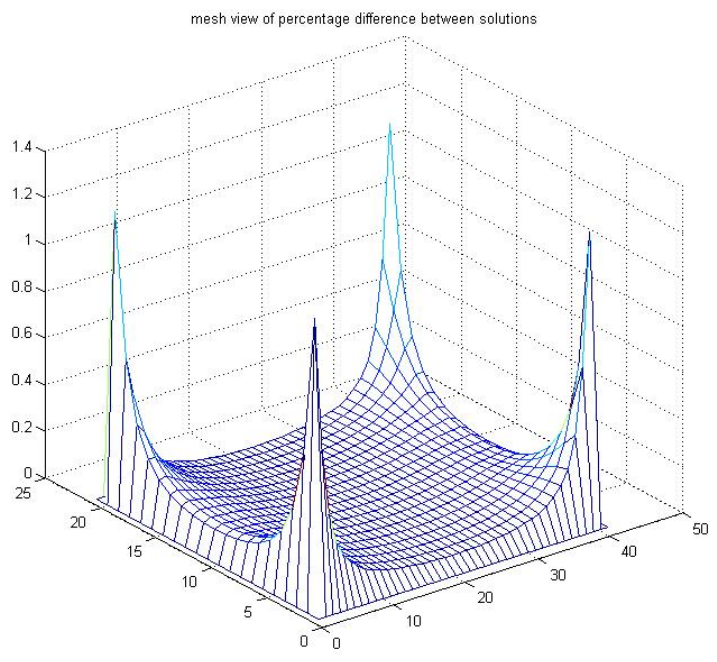
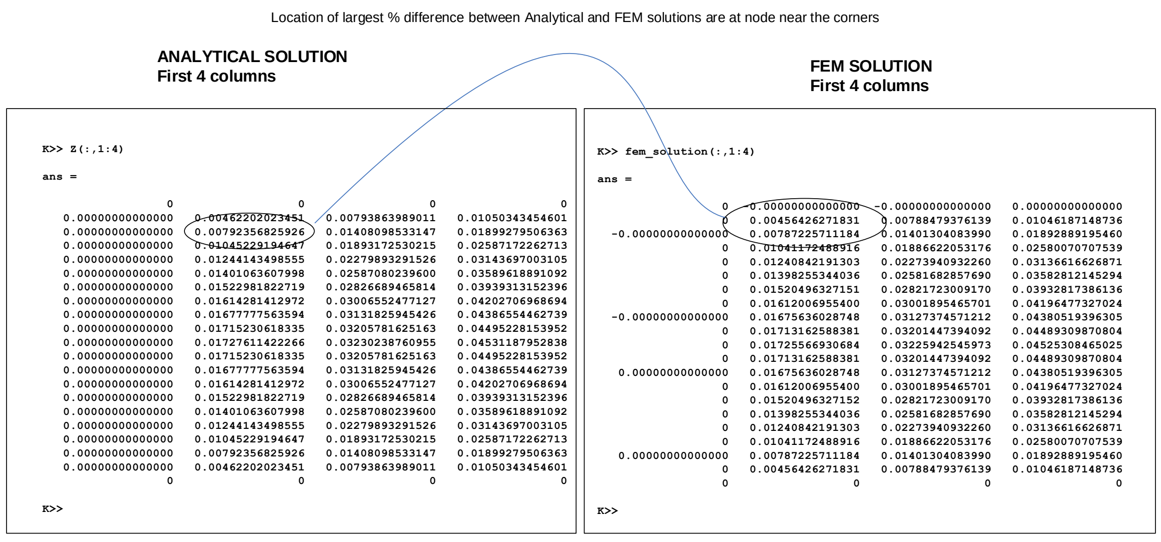


Figure 2: difference in percentage

Below shows a listing of the nodal values for the first 4 columns in the solution matrix.


Location of largest % difference between Analytical and FEM solutions are at node near the corners
ANALYTICAL SOLUTION
First 4 columns
FEM SOLUTION
First 4 columns

```
K>> Z(:,1:4)

ans =

                 0                  0                  0                  0
   0.00000000000000   0.00462202023451   0.00793863989011   0.01050343454601
   0.00000000000000   0.00792356825926   0.01408098533147   0.01899279506363
   0.00000000000000   0.01045229194647   0.01893172530215   0.02587172262713
   0.00000000000000   0.01244143498555   0.02279893291526   0.03143697003105
   0.00000000000000   0.01401063607998   0.02587080239600   0.03589618891092
   0.00000000000000   0.01522981822719   0.02826689465814   0.03939313152396
   0.00000000000000   0.01614281412972   0.03006552477127   0.04202706968694
   0.00000000000000   0.01677777563594   0.03131825945426   0.04386554462739
   0.00000000000000   0.01715230618335   0.03205781625163   0.04495228153952
   0.00000000000000   0.01727611422266   0.03230238760955   0.04531187952838
   0.00000000000000   0.01715230618335   0.03205781625163   0.04495228153952
   0.00000000000000   0.01677777563594   0.03131825945426   0.04386554462739
   0.00000000000000   0.01614281412972   0.03006552477127   0.04202706968694
   0.00000000000000   0.01522981822719   0.02826689465814   0.03939313152396
   0.00000000000000   0.01401063607998   0.02587080239600   0.03589618891092
   0.00000000000000   0.01244143498555   0.02279893291526   0.03143697003105
   0.00000000000000   0.01045229194647   0.01893172530215   0.02587172262713
   0.00000000000000   0.00792356825926   0.01408098533147   0.01899279506363
   0.00000000000000   0.00462202023451   0.00793863989011   0.01050343454601
                 0                  0                  0                  0

K>>
```

```
K>> fem_solution(:,1:4)

ans =

                 0  -0.00000000000000  -0.00000000000000   0.00000000000000
                 0   0.00456426271831   0.00788479376139   0.01046187148736
 -0.00000000000000   0.00787225711184   0.01401304083990   0.01892889195460
                 0   0.01041172488916   0.01886622053176   0.02580070707539
                 0   0.01240842191303   0.02273940932260   0.03136616626871
                 0   0.01398255344036   0.02581682857690   0.03582812145294
                 0   0.01520496327151   0.02821723009170   0.03932817386136
                 0   0.01612006955400   0.03001895465701   0.04196477327024
 -0.00000000000000   0.01675636028748   0.03127374571212   0.04380519396305
                 0   0.01713162588381   0.03201447394092   0.04489309870804
                 0   0.01725566930684   0.03225942545973   0.04525308465025
                 0   0.01713162588381   0.03201447394092   0.04489309870804
  0.00000000000000   0.01675636028748   0.03127374571212   0.04380519396305
                 0   0.01612006955400   0.03001895465701   0.04196477327024
                 0   0.01520496327152   0.02821723009170   0.03932817386136
                 0   0.01398255344036   0.02581682857690   0.03582812145294
                 0   0.01240842191303   0.02273940932260   0.03136616626871
                 0   0.01041172488916   0.01886622053176   0.02580070707539
  0.00000000000000   0.00787225711184   0.01401304083990   0.01892889195460
                 0   0.00456426271831   0.00788479376139   0.01046187148736
                 0                  0                  0                  0

K>>
```

Figure 3: numerical difference

The plot below shows the absolute difference between the analytical and the FEM solutions in 3D mesh.

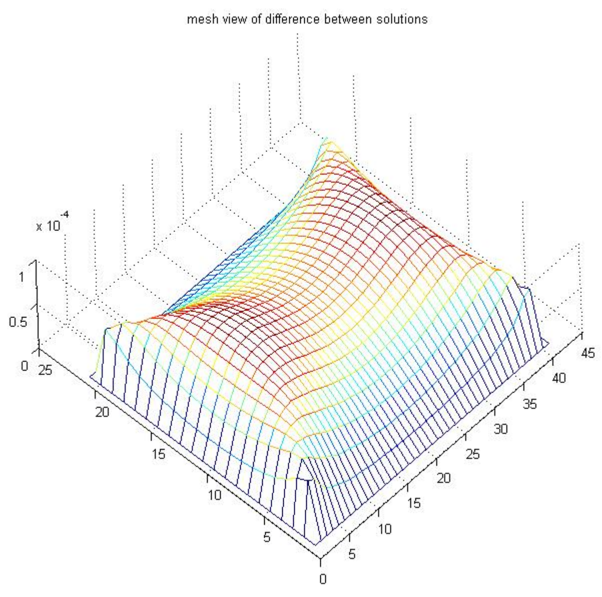


Figure 4: 3D mesh plot of the difference

The plot below is the analytical solution (The wrapping function, i.e. the solution function $\Phi(x, y)$ shown using larger number of contours)

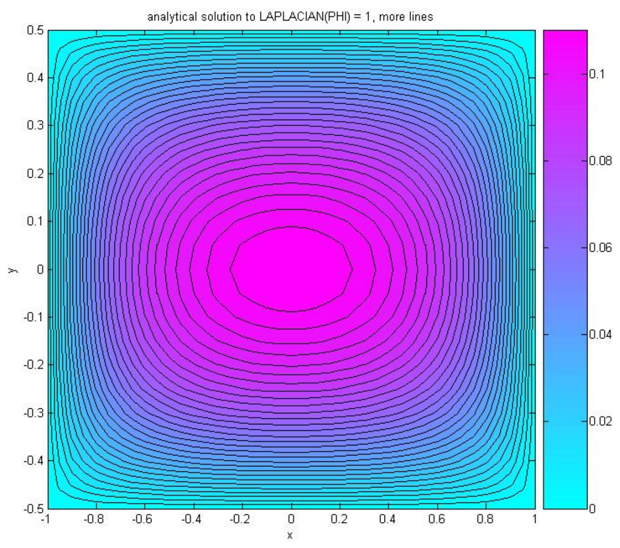


Figure 5: analytical solution

The plot below is a contour plot of the analytic solution.

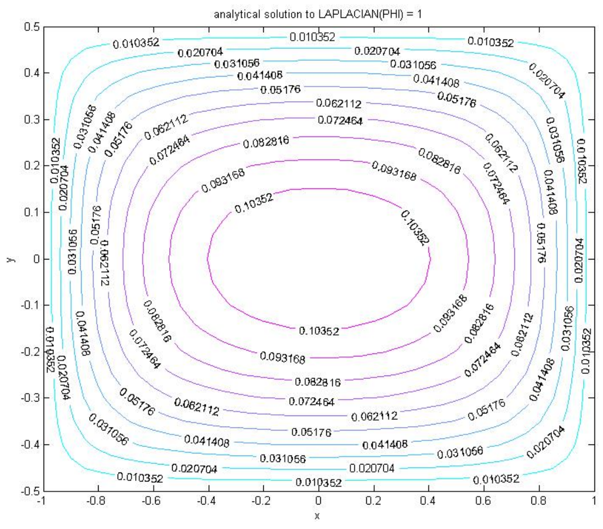


Figure 6: contour plot of the analytic solution

The plot below is a contour plot of the FEM solution to compare with the above.

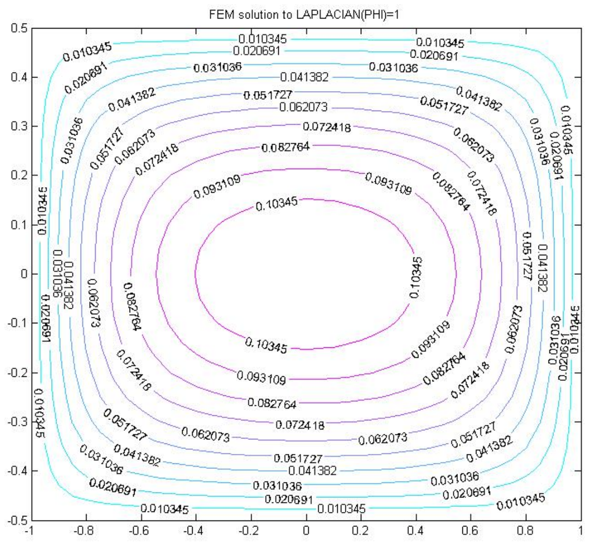


Figure 7: FEM contour plot

The plot below is the above 2 contour plots side-by-side.

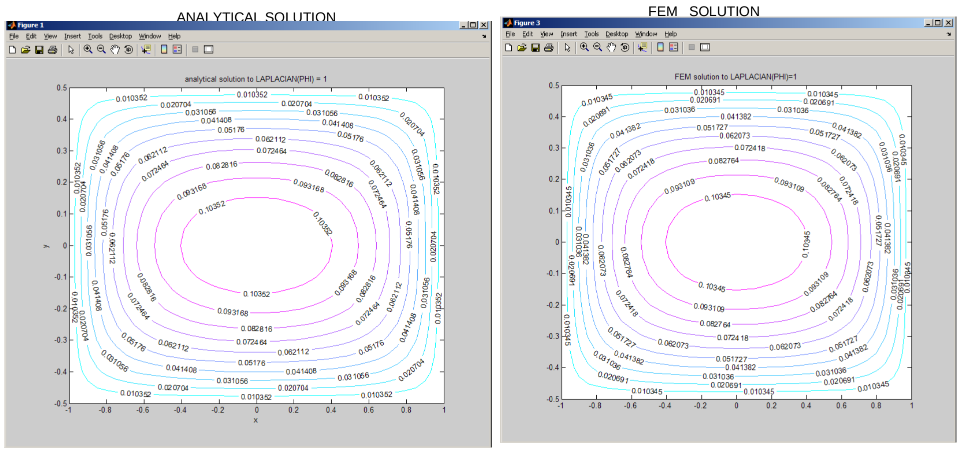

ANALYTICAL SOLUTION
FEM SOLUTION
Figure 1
Figure 3
File Edit View Insert Tools Desktop Window Help
analytical solution to LAPLACIAN(PHI) = 1
FEM solution to LAPLACIAN(PHI)=1

Figure 8: contour plots side by side

This below is a mesh plot of the analytical solution and FEM solution side by side.

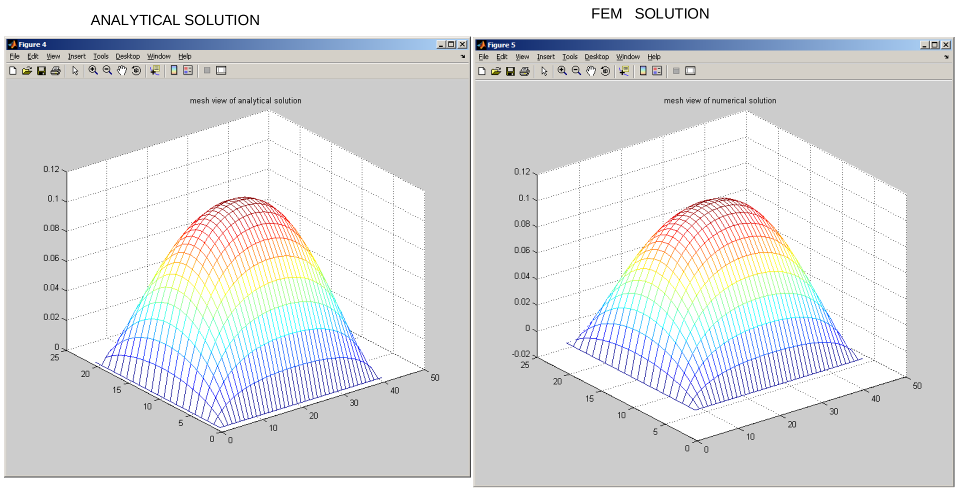


Figure 9: mesh plot side by side

3 Appendix

3.1 Modified Qing Wang matlab function

```
function fem_solution=poisson_fem_as_function
% original code by Qing Wang, UCI
%
%minor changes by Nasser M. Abbasi, UCI.  on May 28, 2006:
% - made it a function to be able to call it.
% - changes to contour levels to make it the same as analytical
%   soltution for better comparison.
% - changed the grid to go from -a/2,a/2 instead of 0,a and smiliarly
%   to the b side.
%
% SOLVE THE FOLLOWING PROBLEM USING FEM
%
%    LAPLACIAN( PHI ) = 1
%

format long;
fc=1.0;    % f constant
bc=0.;     % The Dirichlet boundary condition, constant
W=1.0;   % the width of the rectangle
L=2.0;    % the length of the rectangle
N=20;      % the number of elements in Y direction
M=40;       % the number of elements in X direction
dx = L/M;    % dx
dy = W/N;    % dy
D=(N+1)*(M+1);    % The dimension of the global stiffniss matrix or number of nodes
s=zeros(D,D);      % global stiffness matrix
f=zeros(D,1);       % load factor
v=zeros(N+1,M+1);    % the collocated results for plotting
xindex=0;
yindex=0;
x1=0; x2=0; x3=0; x4=0;   % (x1,x2,x3) and (x4,x2,x3) are summits of the two trangles in one
j1=0; j2=0;
% generating the global stiffness matrix and load vectors
for yindex = 1:N
    for xindex = 1:M
        x1 = xindex + (yindex-1)*(M+1);
        x2 = 1+ x1;
        x3 = xindex + yindex*(M+1);
        x4 = 1+ x3;

        j1 = dx*dy;  % the Jacobian of the first trangle
        s(x1,x1) = s(x1,x1)+(dx^2+dy^2)/(2*j1);
        s(x1,x2) = s(x1,x2)-dy^2/(2*j1);
        s(x1,x3) = s(x1,x3)-dx^2/(2*j1);
        s(x2,x1) = s(x2,x1)-dy^2/(2*j1);
        s(x2,x2) = s(x2,x2)+dy^2/(2*j1);
        s(x2,x3) = s(x2,x3);
        s(x3,x1) = s(x3,x1)-dx^2/(2*j1);
        s(x3,x2) = s(x3,x2);
        s(x3,x3) = s(x3,x3)+dx^2/(2*j1);
        f(x1,1) = f(x1,1)+ fc*j1/6;
        f(x2,1) = f(x2,1)+ fc*j1/6;
        f(x3,1) = f(x3,1)+ fc*j1/6;   %  Process the first trangle in one block

        j2 = dx*dy;   % the Jacobian of the second trangle
        s(x4,x4) = s(x4,x4)+(dx^2+dy^2)/(2*j2);
        s(x4,x2) = s(x4,x2)-dx^2/(2*j2);
        s(x4,x3) = s(x4,x3)-dy^2/(2*j2);
        s(x2,x4) = s(x2,x4)-dx^2/(2*j2);
        s(x2,x2) = s(x2,x2)+dx^2/(2*j2);
        s(x2,x3) = s(x2,x3);
        s(x3,x4) = s(x3,x4)-dy^2/(2*j2);
        s(x3,x2) = s(x3,x2);
        s(x3,x3) = s(x3,x3)+dy^2/(2*j2);
        f(x4,1) = f(x4,1)+ fc*j2/6;
        f(x2,1) = f(x2,1)+ fc*j2/6;
        f(x3,1) = f(x3,1)+ fc*j2/6;  %  Process the second trangle
    end
end

% applying BC
for xindex=1:M+1
    s(xindex,:) = zeros(1,D); s(xindex,xindex)=1; f(xindex,1)=bc;
    s(xindex+N*(M+1),:) = zeros(1,D);
    s(xindex+N*(M+1),xindex+N*(M+1))=1;
    f(xindex+N*(M+1),1)=bc;
end         % the bottome and upper BC
for yindex=1:N-1
    s(1+yindex*(M+1),:)=zeros(1,D);
    s(1+yindex*(M+1),1+yindex*(M+1))=1;
    f(1+yindex*(M+1),1)=bc;
    s((1+yindex)*(M+1),:)=zeros(1,D);
    s((1+yindex)*(M+1),(1+yindex)*(M+1))=1;
    f((1+yindex)*(M+1),1)=bc;
end         % the left and right side BC

q=s\f;      % solve q

% generating visulized results
for yindex=1:N+1
    for xindex=1:M+1
        v(yindex,xindex) = q((M+1)*(yindex-1)+xindex);
    end
end

figure;
[X,Y] = meshgrid(-L/2:dx:L/2,-W/2:dy:W/2);
[C,h] = contour(X,Y,v,10);
clabel(C,h)
colormap cool
title('FEM solution to LAPLACIAN(PHI)=1');
%figure;
%contour(s); title('Global stiffness matrix contour plot');

fem_solution = v;
```

3.2 Code used for plotting

Here is the Matlab function used to plot the analytical solution. This function makes a call to the above FEM function.

```
function nma_verify_MAE207_solution
%function nma_verify_MAE207_solution
%
% Display the solution of the poisson equation by
% plotting the analytical solution to the torsion
% problem on a rectangular cross section.
%
% see report and references on
% http://12000.org/my_courses/spring_MAE_207/
% by Nasser Abbasi.
%
% The cross section is
%
%     a   a
%   +---+---+
%   |       | b
%   |       +
%   |       | b
%   +-------+
%
% The origin is in the middle of the cross section:
%
%       y
%       ^
%       |
%       |
%       |
%   +---+---+
%   |   |   |
%   |   +---+-----------------> x
%   |       |
%   +-------+
%
% This Analytical solution is to the following problem
%
%   LAPLACIAN( PHI ) = - 2 G K
%
% To verify the analytical solution against the FEM solution, which solves
%
%   LAPLACIAN( PHI ) = 1
%
% then in this analytical solution we set G*K=-0.5
% But k = T/(G*J), hence we need to have
%
%              T/J = -0.5
%
% change history
% name date    changes
% nma  052806  started

close all;
clear all;

a = 1;    % meter
b = 0.5;  % meter
T_OVER_J = 0.5;   % SET THIS TO MATCH FEM CONSTANT f=1, see above
% I think the Matlab FEM code should have used -1
% but it is easier to change this to +0.5 to match them
```

```
dx = 2*a/40;
dy = 2*b/20;

%applied_torque = 1 % Newton-meter   NOT USED

%J = get_torsion_constant(a,b);  NOT USED

[X,Y]    = meshgrid(-a:dx:a, -b:dy:b);
big_term = 0;

NUMBER_TERMS = 100;

for n = 1:2:NUMBER_TERMS
    big_term = big_term + (1/n^3)*(-1)^( (n-1)/2)* ...
        (1- (cosh(n*pi*Y/(2*a))./(cosh(n*pi*b/(2*a))))).*cos(n*pi*X/(2*a));
end
%Z = 32*applied_torque*a^2/(J*pi^3) * big_term;
Z = 32*T_OVER_J*a^2/(pi^3) * big_term;
%
%  DONE. Now do plots
%
figure;
[C,h] = contour(X,Y,Z,10);
clabel(C,h)
colormap cool
title('analytical solution to LAPLACIAN(PHI) = 1');
xlabel('x'); ylabel('y');
axis square

figure;
[C,h] = contourf(X,Y,Z,30);
colormap cool
title('analytical solution to LAPLACIAN(PHI) = 1, more lines');
xlabel('x'); ylabel('y');

%
% now obtain the FEM solution and compare point-to-point
%

fem_solution = poisson_fem_as_function();

difference      = fem_solution-Z;
abs_difference  = abs(difference);

figure;
[C,h] = contourf(abs_difference,13);
%clabel(C,h)
colormap cool
title('absolute difference between solutions');
xlabel('x'); ylabel('y');

figure;
mesh(abs_difference);
title('mesh view of absolute difference between solutions');


figure;
mesh(Z); title('mesh view of analytical solution');

figure;
mesh(fem_solution); title('mesh view of numerical solution');

figure;
```

```
[nRow,nCol]=size(Z);
per_diff=zeros(nRow,nCol);
for i=1:nRow
    for j=1:nCol
        if abs(Z(i,j))>eps
            per_diff(i,j)= 100* abs_difference(i,j)/Z(i,j);
        else
            per_diff(i,j)= 100* abs_difference(i,j);
        end
    end
end
mesh(per_diff); title('mesh view of percentage difference between solutions');

figure;
[C,h] = contourf(per_diff,13);
%clabel(C,h)
colormap cool
title('% difference between solutions');
xlabel('x'); ylabel('y');

[r,c]=find(abs_difference==max(abs_difference(:)));
fprintf('Maximum difference at node [%d,%d]\n',r,c);


end

%*******************
%
%
%*******************
function J = get_torsion_constant(a,b)
NUMBER_TERMS = 1000;

fixed_term = pi*b/(2*a);
J = 0;
for n=1:2:NUMBER_TERMS
    J = J + tanh(n*fixed_term)/n^5;
end
J = J * 192*a/(b*pi^5);
J = 16*a^3*b/3 * (1 - J);
end
```