plots.m

```
%% Plots
% This code generates plots of grid and deflection pattern automatically.
% Then based on user input generates plots of stress distribution along
% thickness.
% ------------------------------------------------------------------
%            This portion generates the stress comparison plots
%-------------------------------------------------------------------
% This following code will plot the stress distribution along thickness for
% the middle point of plate.
ele_no=(nel_x*nel_y)/2+nel_x/2+1;
sxx_core=stress_field_plate_core(1,ele_no);
syy_core=stress_field_plate_core(2,ele_no);
szz_core=stress_field_plate_core(3,ele_no);

c_plot=-1:1/10:1;
z_plot=-c:c/10:c;
normal=(x_a/h_tot)^2;

sxx_core_sub=subs(sxx_core,{'x','y','z'},{0,0,z_plot});
syy_core_sub=subs(syy_core,{'x','y','z'},{0,0,z_plot});
szz_core_sub=subs(szz_core,{'x','y','z'},{0,0,z_plot});
sxx_core=double(sxx_core_sub/normal); syy_core=double(syy_core_sub/normal);
szz_core=double(szz_core_sub/normal);

szz_theory_5_htot=0.0199342 + 1.89609*z_plot + 0.821646*z_plot.^2 + 22.6079*z_plot.^3;
sxx_theory_5_htot=0.000582277 - 0.0073519*z_plot + 1.64365*z_plot.^2 + 42.2999*z_plot.^3;
syy_theory_5_htot=0.000579599 + 0.0589626*z_plot + 1.64294*z_plot.^2 + 48.1317*z_plot.^3;

szz_theory_20_htot=0.00125007 + 0.136564*z_plot + 0.00334897*z_plot.^2 + 0.358917*z_plot.^3;
sxx_theory_20_htot=0.0000438529 + 0.0217781*z_plot + 0.00669969*z_plot.^2 + 0.628463*z_plot.^3;
syy_theory_20_htot=0.000043641 + 0.0543721*z_plot + 0.0066962*z_plot.^2 + 0.807206*z_plot.^3;


load('stress_elast_5htot.mat') % Use 'stress_elast_20htot.mat', if want to plot stresses of plate with a = b = 20*h_tot

figure
plot(c_plot,sxx_core,'b-','LineWidth',1); hold on
plot(c_plot,sxx_theory_5_htot,'rs','LineWidth',1);
plot(c_plot,sxx_elast,'k*','LineWidth',1); hold on
xlabel('z/c','fontweight','bold')
ylabel('\sigma_x_x/\sigma_n_o_r_m','fontweight','bold')
title('Case-1 : a = b =5h_t_o_t','fontweight','bold')
legend('FEM (current)','EHSAPT','Elasticity')

figure
plot(c_plot,syy_core,'b-','LineWidth',1); hold on
plot(c_plot,syy_theory_5_htot,'rs','LineWidth',1);
plot(c_plot,syy_elast,'k*','LineWidth',1); hold on
xlabel('z/c','fontweight','bold')
ylabel('\sigma_y_y/\sigma_n_o_r_m','fontweight','bold')
title('Case-1 : a = b = 5h_t_o_t','fontweight','bold')
legend('FEM (current)','EHSAPT','Elasticity')

figure
plot(c_plot,szz_core,'b-','LineWidth',1); hold on
plot(c_plot,szz_theory_5_htot,'rs','LineWidth',1);
plot(c_plot,szz_elast,'k*','LineWidth',1); hold on
xlabel('z/c','fontweight','bold')
ylabel('\sigma_z_z/\sigma_n_o_r_m','fontweight','bold')
title('Case-1 : a = b = 5h_t_o_t','fontweight','bold')
legend('FEM (current)','EHSAPT','Elasticity')

save('stresses_5htot_20x20.mat','szz_core','szz_elast','szz_theory_5_htot','syy_core','syy_elast','syy_theory_5_htot','sxx_core','sxx_elast','c_p


%% Grid Generation
% This portion of the code generates a structured grid based on the number
% of divisions in x- & y- directions and the plate dimensions.

x_grid=0:x_a/nel_x:x_a;
y_grid=0:y_b/nel_y:y_b;
[X,Y]=meshgrid(x_grid,y_grid);
if nel_x > nel_y || nel_x < nel_y
    plate_grid=zeros(nel_y+1,nel_x+1);
    elseif nel_x==nel_y
        plate_grid=zeros(nel_x+1,nel_y+1);
end
%
surf(X,Y,plate_grid)
set(gca,'xtick',0:x_a/nel_x:x_a)
set(gca,'ytick',0:y_b/nel_y:y_b)
title(['Grid elements :   ' num2str(nel_x) ' x ' num2str(nel_y)])
xlabel('x','fontweight','bold')
ylabel('y','fontweight','bold')
set(gca,'xtick',[0,x_a])
set(gca,'ytick',[0,y_b])
set(gca,'ztick',[])

%% Displacement field plot
%
% Deflection plot : It shows the pattern of the deflection of the plate
%
figure
x_val=0; y_val=0; jj=1;
a_elem=x_a/nel_x; b_elem=y_b/nel_y;
%
```

```
x_grid=x_val:a_elem:x_val+a_elem;
y_grid=y_val:b_elem:y_val+b_elem;
%
x_loc=0:a_elem:a_elem;  y_loc=0:b_elem:b_elem;
[X_loc,Y_loc]=meshgrid(x_loc,y_loc);
%
for ii=1:nel_x*nel_y
    [X,Y]=meshgrid(x_grid,y_grid);
    eq=double(subs(u_field_plate(3,jj),{'x','y'},{X_loc,Y_loc}));
    surf(X,Y,eq)
    hold on
    if mod(ii,nel_x)==0
        x_val=0; y_val=y_val+b_elem;
    else x_val=x_val+a_elem;
    end
    jj=jj+1;
    x_grid=x_val:a_elem:x_val+a_elem;
    y_grid=y_val:b_elem:y_val+b_elem;
end

```