```
function solvePlateHermiteFinal()


    E  = 200e9;
    nu = 0.3;
    t  = 4e-3;
    a  = 0.5;
    b  = 0.5;
    q0 = 1000;
    D  = E*(t^3)/(12*(1 - nu^2));

    pList   = [3,4,5];
    bcTypes = {'SSSS','CFFF'};


    sigma_yield = 450e6;

    for p = pList
        for iBC = 1:length(bcTypes)
            bcType = bcTypes{iBC};


            [K, F] = buildPlateSystem_local(p,a,b,q0,D,nu,bcType);


            alpha = K \ F;


            varK     = sprintf('K_p%d_%s',     p, bcType);
            varF     = sprintf('F_p%d_%s',     p, bcType);
            varAlpha = sprintf('alpha_p%d_%s', p, bcType);
            assignin('base', varK,     K);
            assignin('base', varF,     F);
            assignin('base', varAlpha, alpha);


            [xx,yy,W] = computeDeflection_local(p, alpha, a, b);
            figure('Name', sprintf('p=%d, bc=%s',p,bcType));
            surf(xx,yy,W,'EdgeColor','none');
            title(sprintf('Deflection w, p=%d, bc=%s', p, bcType));
            xlabel('x'); ylabel('y'); zlabel('w');
            colorbar;
            xlim([0 a]);
            ylim([0 b]);
            view(140,30);


            figure('Name', sprintf('Deflection Contour p=%d bc=%s',p,bcType));
            contourf(xx, yy, W, 30, 'LineColor','none');
            colorbar; colormap(jet);
            title(sprintf('Deflection Contour, p=%d, bc=%s', p, bcType));
            xlabel('x'); ylabel('y');
            axis equal; axis([0 a 0 b]);


            Nx = 31; Ny = 31;
            xvec = linspace(0,a,Nx);
            yvec = linspace(0,b,Ny);
            [Xv, Yv] = meshgrid(xvec, yvec);
            z_top = +t/2;
```

```
sigmaVM = zeros(Ny, Nx);


sigma_vm_max = -inf;
maxPos = [0,0];


for ii=1:Nx
    for jj=1:Ny

        [~, wxx, wyy, wxy] = computeDerivsAtPoint(p, alpha, xvec(ii), yvec(jj), a, b);


        sxx = -(E/(1-nu^2)) * z_top * ( wxx + nu*wyy );
        syy = -(E/(1-nu^2)) * z_top * ( wyy + nu*wxx );
        sxy = -(E/(1+nu))   * z_top * wxy;


        sigma_vm = sqrt( 0.5*((sxx - syy)^2 + sxx^2 + syy^2) + 3*(sxy^2) );
        sigmaVM(jj, ii) = sigma_vm;
        if sigma_vm>sigma_vm_max
            sigma_vm_max = sigma_vm;
            maxPos = [xvec(ii), yvec(jj)];
        end
    end
end

figure('Name', sprintf('VonMises p=%d bc=%s',p,bcType));
contourf(Xv, Yv, sigmaVM, 20, 'LineColor','none');
colormap(jet); colorbar;
title(sprintf('Von Mises Stress (top fiber), p=%d, bc=%s',p,bcType));
xlabel('x'); ylabel('y');
axis equal; axis([0 a 0 b]);


if sigma_vm_max>1e-12

    q_yield = (sigma_yield / sigma_vm_max)* q0;
    fprintf('p=%d, bc=%s:  Max sigma_vm=%.3e at (%.3f,%.3f).  => q_yield=%.3e N/m^2\n',...
        p, bcType, sigma_vm_max, maxPos(1), maxPos(2), q_yield);
else
    fprintf('p=%d, bc=%s: sigma_vm was near zero?\n', p, bcType);
end


xc = a/2;  yc=b/2;
z_vals = linspace(-t/2, t/2, 51);
[~, wxxC, wyyC, wxyC] = computeDerivsAtPoint(p, alpha, xc, yc, a, b);
sig_xxZ = zeros(size(z_vals));
sig_yyZ = zeros(size(z_vals));
sig_xyZ = zeros(size(z_vals));
for iz=1:length(z_vals)
    zCur = z_vals(iz);
    sig_xxZ(iz) = -(E/(1-nu^2)) * zCur * ( wxxC + nu*wyyC );
    sig_yyZ(iz) = -(E/(1-nu^2)) * zCur * ( wyyC + nu*wxxC );
    sig_xyZ(iz) = -(E/(1+nu))   * zCur * wxyC;
end
figure('Name', sprintf('Through-thickness p=%d bc=%s',p,bcType));
subplot(1,3,1);
plot(sig_xxZ, z_vals, 'LineWidth',2); grid on;
xlabel('\sigma_{xx}'); ylabel('z (m)');
title('\sigma_{xx} vs z');
```

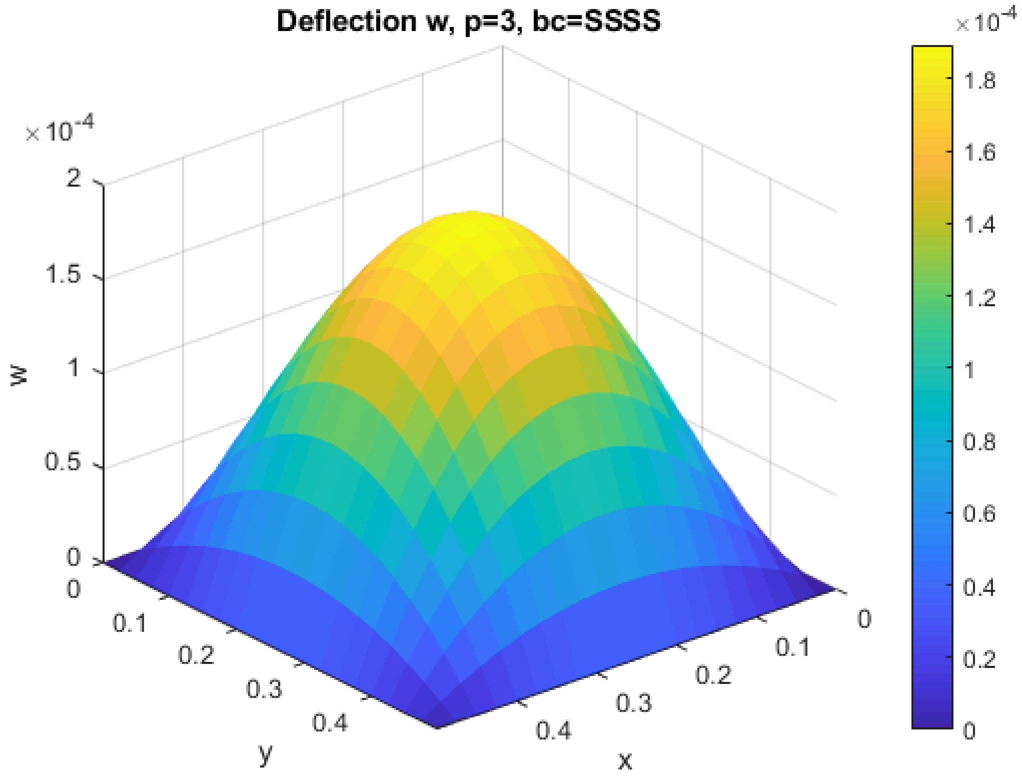
```
            subplot(1,3,2);
            plot(sig_yyZ, z_vals, 'LineWidth',2); grid on;
            xlabel('\sigma_{yy}'); ylabel('z (m)');
            title('\sigma_{yy} vs z');

            subplot(1,3,3);
            plot(sig_xyZ, z_vals, 'LineWidth',2); grid on;
            xlabel('\sigma_{xy}'); ylabel('z (m)');
            title('\sigma_{xy} vs z');

            sgtitle(sprintf('Through-thickness stresses at center, p=%d, bc=%s',p,bcType));


            strainEnergy = 0.5 * (alpha')*K*alpha;
            fprintf('p=%d, bc=%s:  Strain Energy = %.6e J\n', p, bcType, strainEnergy);

        end
    end

    disp('All done! Check your workspace for the new results, plots, etc.');
end

%%======================================================================

function [K,F] = buildPlateSystem_local(p,a,b,q0,D,nu,bcType)
n1D = p+1;
N   = n1D*n1D;
K   = zeros(N,N);
F   = zeros(N,1);

[gx,wx] = gauss1D(5,0,a);
[gy,wy] = gauss1D(5,0,b);

for igx=1:length(gx)
    x   = gx(igx);
    wtx = wx(igx);

    phiX   = zeros(n1D,1);
    d2phiX = zeros(n1D,1);
    for ix=1:n1D
        [val,~,d2val] = shapeHermite_local(p, ix, x, a);
        phiX(ix)   = val;
        d2phiX(ix) = d2val;
    end

    for igy=1:length(gy)
        y   = gy(igy);
        wty = wy(igy);
        w2D = wtx*wty;

        phiY   = zeros(n1D,1);
        d2phiY = zeros(n1D,1);
        for iy=1:n1D
            [val,~,d2val] = shapeHermite_local(p, iy, y, b);
            phiY(iy)   = val;
            d2phiY(iy) = d2val;
        end

        for ix1=1:n1D
            for iy1=1:n1D
                I = (ix1-1)*n1D + iy1;
                phiXX_i = d2phiX(ix1)* phiY(iy1);
                phiYY_i = phiX(ix1)* d2phiY(iy1);
                phiXY_i = d1phi_local(p, ix1, x, a)* ...
                          d1phi_local(p, iy1, y, b);

                F(I) = F(I) + q0*(phiX(ix1)*phiY(iy1))* w2D;

                for ix2=1:n1D
                    for iy2=1:n1D
                        J = (ix2-1)*n1D + iy2;

                        phiXX_j = d2phiX(ix2)* phiY(iy2);
                        phiYY_j = phiX(ix2)* d2phiY(iy2);
                        phiXY_j = d1phi_local(p, ix2, x, a)* ...
                                  d1phi_local(p, iy2, y, b);

                        valK = ( phiXX_i*phiXX_j + phiYY_i*phiYY_j ...
                               + 2*(1-nu)*phiXY_i*phiXY_j ...
                               + nu*(phiXX_i*phiYY_j + phiYY_i*phiXX_j ) );
                        K(I,J) = K(I,J) + D*valK*w2D;
                    end
                end
            end
        end
    end
end

BCidx = identifyBC_local(p, bcType);
for ibc=1:length(BCidx)
    ii = BCidx(ibc);
    K(ii,:)=0;
    K(:,ii)=0;
    K(ii,ii)=1;
    F(ii)=0;
end
end

function [xx,yy,W] = computeDeflection_local(p, alpha, a, b)
Nx=21; Ny=21;
xx = linspace(0,a,Nx);
yy = linspace(0,b,Ny);
W  = zeros(Ny,Nx);

n1D = p+1;
for ix=1:Nx
    x= xx(ix);
    phiX = zeros(n1D,1);
    for iLx=1:n1D
        [val,~,~] = shapeHermite_local(p,iLx,x,a);
        phiX(iLx)= val;
    end
    for iy=1:Ny
        y= yy(iy);
        phiY = zeros(n1D,1);
        for iLy=1:n1D
            [val,~,~] = shapeHermite_local(p,iLy,y,b);
            phiY(iLy)= val;
        end
        wsum=0; idx=1;
        for iLx=1:n1D
            for iLy=1:n1D
```

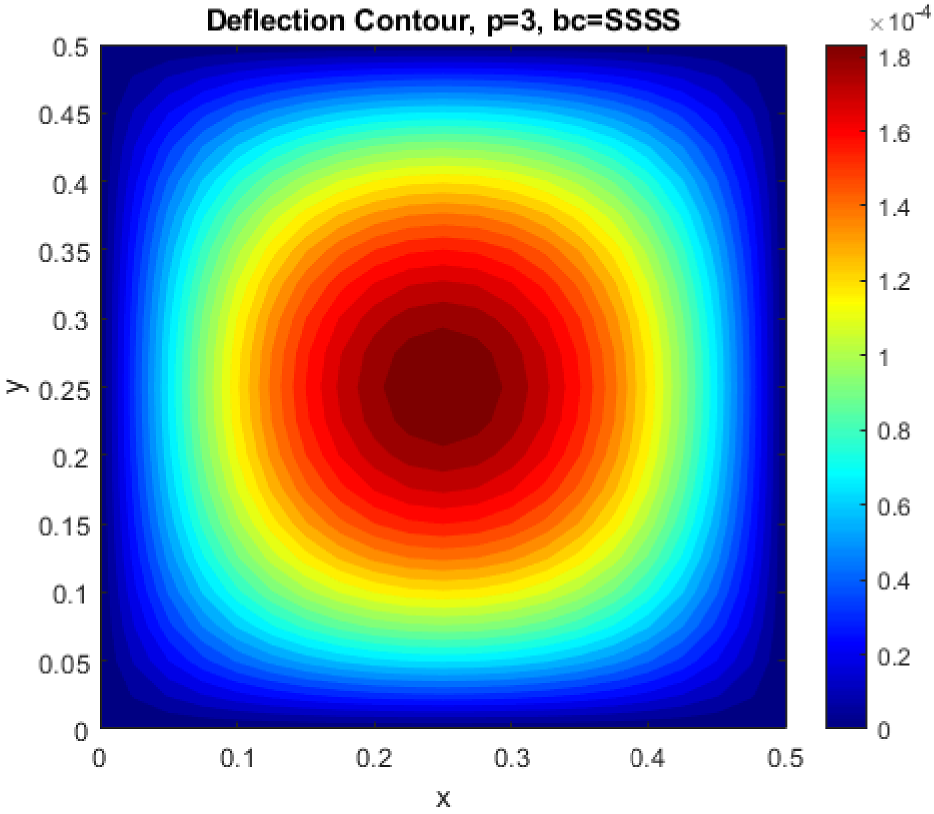
```
                wsum = wsum + alpha(idx)*phiX(iLx)*phiY(iLy);
                idx=idx+1;
            end
        end
        W(iy,ix)= wsum;
    end
end
end

function [val,dval,d2val] = shapeHermite_local(p, iLocal, X, A)
if p==3
    [val,dval,d2val] = shapeCubic(iLocal, X, A);
elseif p==4
    if iLocal<=4
        [val,dval,d2val] = shapeCubic(iLocal, X, A);
    else
        [val,dval,d2val] = polyX2Xa2(X,A);
    end
elseif p==5
    if iLocal<=4
        [val,dval,d2val] = shapeCubic(iLocal, X, A);
    elseif iLocal==5
        [val,dval,d2val] = polyX2Xa2(X,A);
    else
        [val,dval,d2val] = polyX3Xa3(X,A);
    end
end
end

function [v,dv,d2v] = shapeCubic(iLocal, x, a)
xi = x / a;
switch iLocal
    case 1
        [vRef,dvRef,d2vRef] = H1(xi);
        v   = vRef;                 dv  = dvRef*(1/a);        d2v = d2vRef*(1/a^2);
    case 2
        [vRef,dvRef,d2vRef] = H2(xi);
        v   = a * vRef;            dv  = a * dvRef*(1/a);     d2v = a * d2vRef*(1/a^2);
    case 3
        [vRef,dvRef,d2vRef] = H3(xi);
        v   = vRef;                 dv  = dvRef*(1/a);        d2v = d2vRef*(1/a^2);
    case 4
        [vRef,dvRef,d2vRef] = H4(xi);
        v   = a*vRef;              dv  = a* dvRef*(1/a);      d2v = a* d2vRef*(1/a^2);
end
end

function [v,dv,d2v] = H1(xi)
v   = 1 - 3*xi^2 + 2*xi^3;
dv  = -6*xi + 6*xi^2;
d2v = -6 + 12*xi;
end
function [v,dv,d2v] = H2(xi)
v   = xi*(1 - 2*xi + xi^2);
dv  = (1 - 2*xi + xi^2) + xi*(-2 + 2*xi);
d2v = -4 + 6*xi;
end
function [v,dv,d2v] = H3(xi)
v   = 3*xi^2 - 2*xi^3;
dv  = 6*xi - 6*xi^2;
d2v = 6 - 12*xi;
end
```

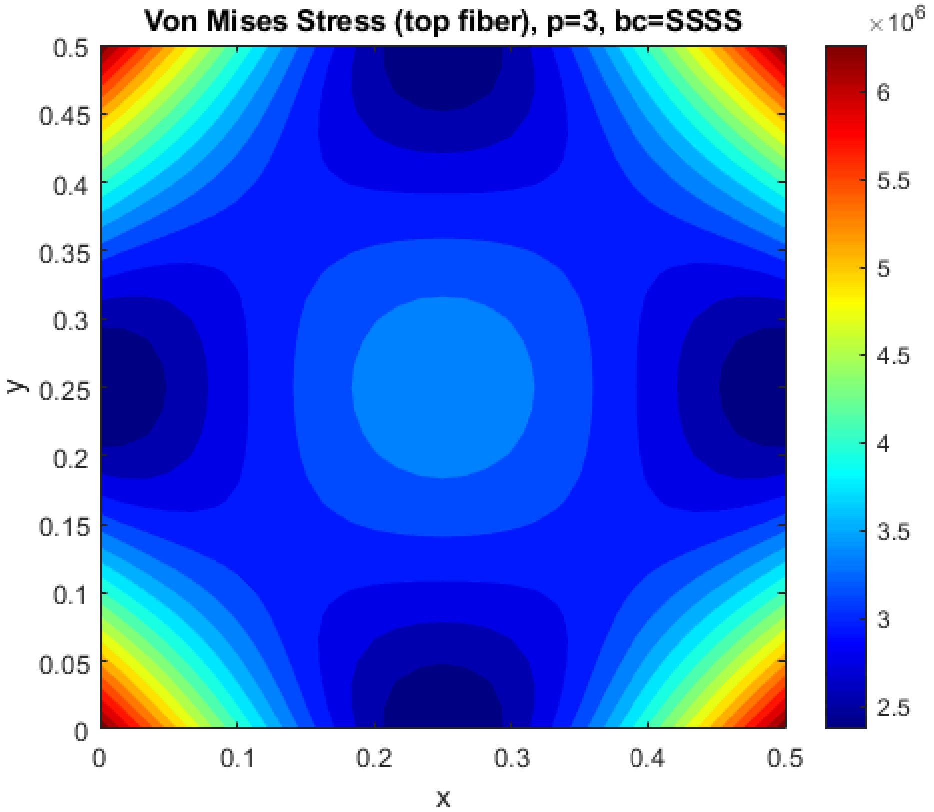
```
function [v,dv,d2v] = H4(xi)
v   = xi^2*(xi - 1);
dv  = 3*xi^2 - 2*xi;
d2v = 6*xi - 2;
end

function [v,dv,d2v] = polyX2Xa2(x,a)
f1   = x^2;       df1  = 2*x;       d2f1 = 2;
f2   = (a - x)^2; df2  = -2*(a - x); d2f2 = 2;
scale = 1/(a^4);
v   = scale* (f1 * f2);
dv  = scale* ( df1*f2 + f1*df2 );
d2v = scale* ( d2f1*f2 + 2*df1*df2 + f1*d2f2 );
end

function [v,dv,d2v] = polyX3Xa3(x,a)
f1   = x^3;         df1  = 3*x^2;      d2f1 = 6*x;
f2   = (a-x)^3;     df2  = -3*(a-x)^2; d2f2 = 6*(a-x);
scale= 1/(a^6);
vv   = f1*f2;
dvv  = df1*f2 + f1*df2;
d2vv = d2f1*f2 + 2*(df1*df2) + f1*d2f2;
v   = scale* vv;
dv  = scale* dvv;
d2v = scale* d2vv;
end

function d1 = d1phi_local(p, iLocal, x, a)
[~,dd,~] = shapeHermite_local(p,iLocal,x,a);
d1= dd;
end

function BCidx = identifyBC_local(p, bcType)
n1D = p+1;
BCidx = [];
switch bcType
    case 'SSSS'
        killX = [1,3];
        killY = [1,3];
        for ix=1:n1D
            for iy=1:n1D
                I = (ix-1)*n1D + iy;
                if ismember(ix,killX) || ismember(iy,killY)
                    BCidx(end+1)=I;
                end
            end
        end

    case 'CFFF'
        killY = 3:n1D;
        for ix=1:n1D
            for iy=killY
                I = (ix-1)*n1D + iy;
                BCidx(end+1)=I;
            end
        end
end
BCidx = unique(BCidx);
end

function [gp,gw] = gauss1D(n,x1,x2)
switch n
```

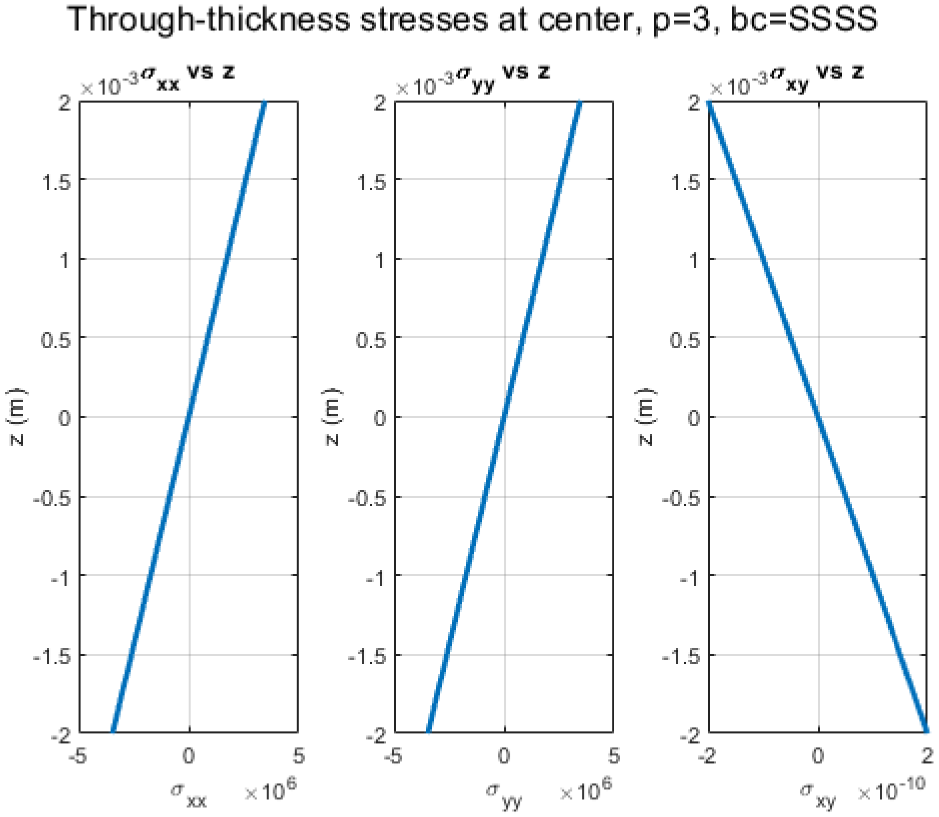
```
    case 1
        xr=0;      wr=2;
    case 2
        xr=[-1/sqrt(3);1/sqrt(3)];
        wr=[1;1];
    case 3
        xr=[-sqrt(3/5);0;sqrt(3/5)];
        wr=[5/9;8/9;5/9];
    case 4
        xr=[-0.8611363116;-0.3399810436;0.3399810436;0.8611363116];
        wr=[0.3478548451;0.6521451549;0.6521451549;0.3478548451];
    case 5
        xr=[-0.9061798459;-0.5384693101;0;0.5384693101;0.9061798459];
        wr=[0.2369268850;0.4786286705;0.5688888889;0.4786286705;0.2369268850];
    otherwise
        error('Gauss rule up to n=5 only.');
end
mid = 0.5*(x1+x2);
ht  = 0.5*(x2 - x1);
gp  = mid + ht*xr;
gw  = ht*wr;
end

%%=====================================================================

function [wVal, wxx, wyy, wxy] = computeDerivsAtPoint(p, alpha, x, y, a, b)


n1D = p+1;
wVal=0; wxx=0; wyy=0; wxy=0;



phiX   = zeros(n1D,1);
d2phiX = zeros(n1D,1);
dphiX  = zeros(n1D,1);

for ix=1:n1D
    [v1, dv1, d2v1] = shapeHermite_local(p, ix, x, a);
    phiX(ix)    = v1;
    dphiX(ix)   = dv1;
    d2phiX(ix) = d2v1;
end


phiY   = zeros(n1D,1);
d2phiY = zeros(n1D,1);
dphiY  = zeros(n1D,1);

for iy=1:n1D
    [v2, dv2, d2v2] = shapeHermite_local(p, iy, y, b);
    phiY(iy)    = v2;
    dphiY(iy)   = dv2;
    d2phiY(iy) = d2v2;
end

index=1;
for ix=1:n1D
    for iy=1:n1D

        c = alpha(index);
        phiXX = d2phiX(ix)*phiY(iy);
        phiYY = phiX(ix)*d2phiY(iy);
        phiXY = dphiX(ix)*dphiY(iy);
        phi   = phiX(ix)*phiY(iy);

        wVal = wVal + c*phi;
        wxx  = wxx  + c*phiXX;
        wyy  = wyy  + c*phiYY;
        wxy  = wxy  + c*phiXY;

        index=index+1;
    end
end
end
```

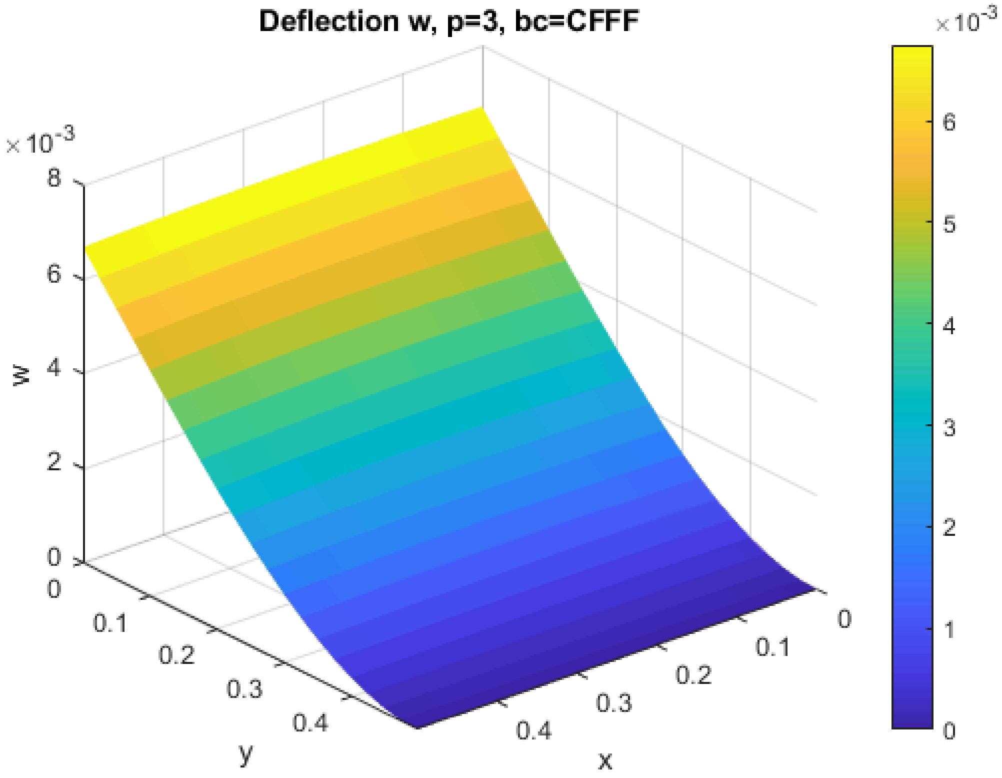
```
p=3, bc=SSSS:  Max sigma_vm=6.458e+06 at (0.000,0.000).  => q_yield=6.968e+04 N/m^2
p=3, bc=SSSS:  Strain Energy = 1.051932e-02 J
p=3, bc=CFFF:  Max sigma_vm=3.548e+07 at (0.250,0.500).  => q_yield=1.268e+04 N/m^2
p=3, bc=CFFF:  Strain Energy = 3.261147e-01 J
p=4, bc=SSSS:  Max sigma_vm=5.339e+06 at (0.000,0.000).  => q_yield=8.429e+04 N/m^2
p=4, bc=SSSS:  Strain Energy = 1.132910e-02 J
p=4, bc=CFFF:  Max sigma_vm=3.528e+07 at (0.250,0.500).  => q_yield=1.275e+04 N/m^2
p=4, bc=CFFF:  Strain Energy = 3.269084e-01 J
p=5, bc=SSSS:  Max sigma_vm=5.314e+06 at (0.000,0.000).  => q_yield=8.469e+04 N/m^2
p=5, bc=SSSS:  Strain Energy = 1.134775e-02 J
p=5, bc=CFFF:  Max sigma_vm=3.534e+07 at (0.250,0.500).  => q_yield=1.273e+04 N/m^2
p=5, bc=CFFF:  Strain Energy = 3.269537e-01 J
All done! Check your workspace for the new results, plots, etc.
```

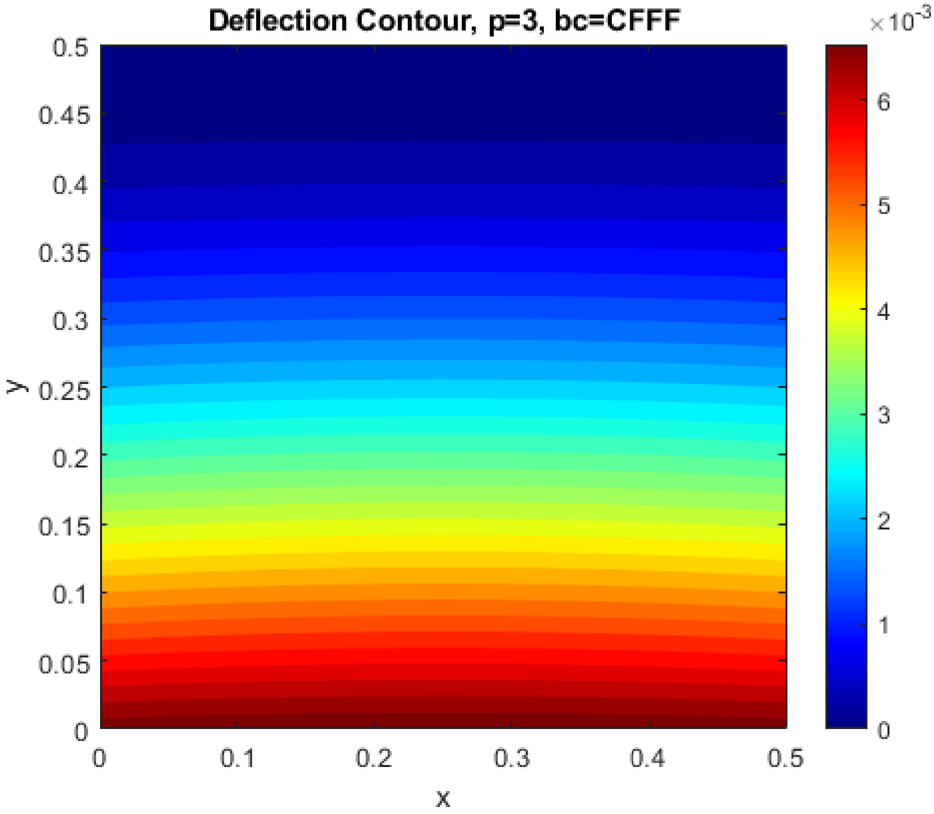

Deflection w, p=3, bc=SSSS
×10^-4
2
1.5
1
0.5
0
w
0
0.1
0.2
0.3
0.4
y
0
0.1
0.2
0.3
0.4
x
×10^-4
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0

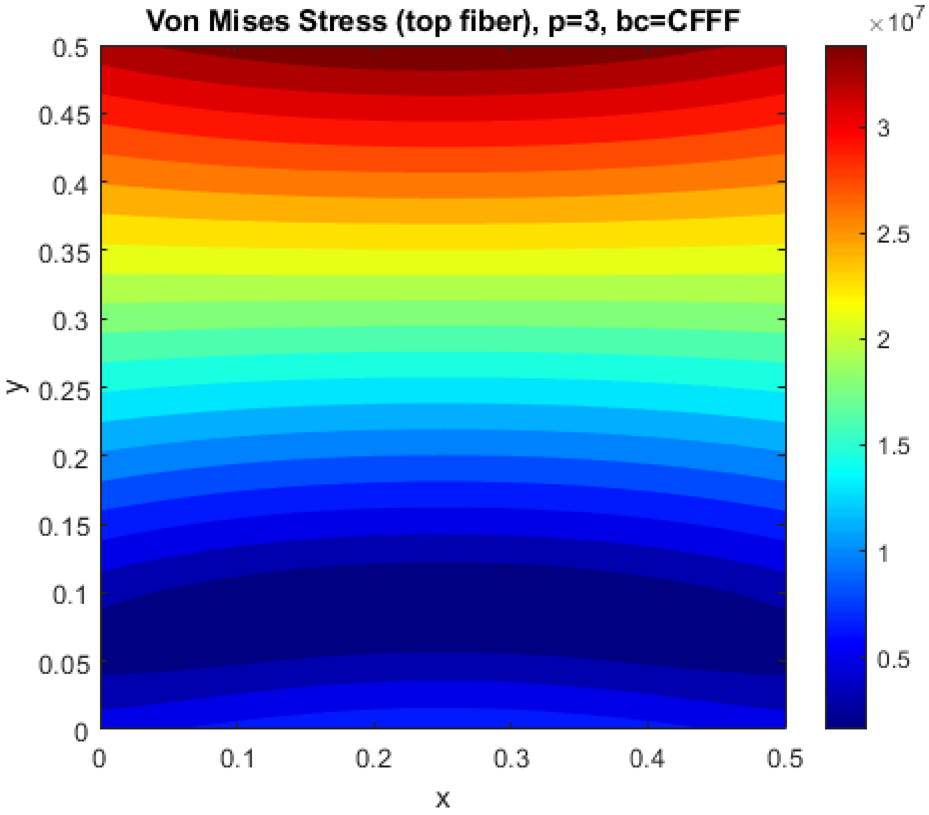

Deflection Contour, p=3, bc=SSSS
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
y
0
0.1
0.2
0.3
0.4
0.5
x
×10^-4
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0

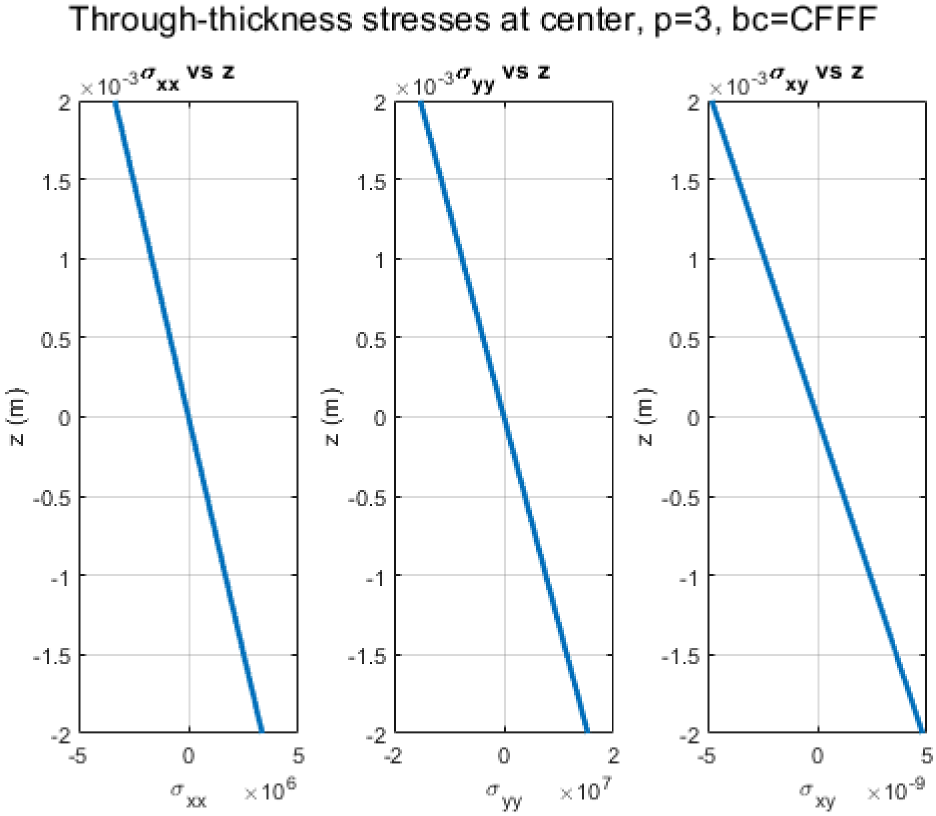

Von Mises Stress (top fiber), p=3, bc=SSSS
×10^6
6
5.5
5
4.5
4
3.5
3
2.5
y
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
x
0
0.1
0.2
0.3
0.4
0.5

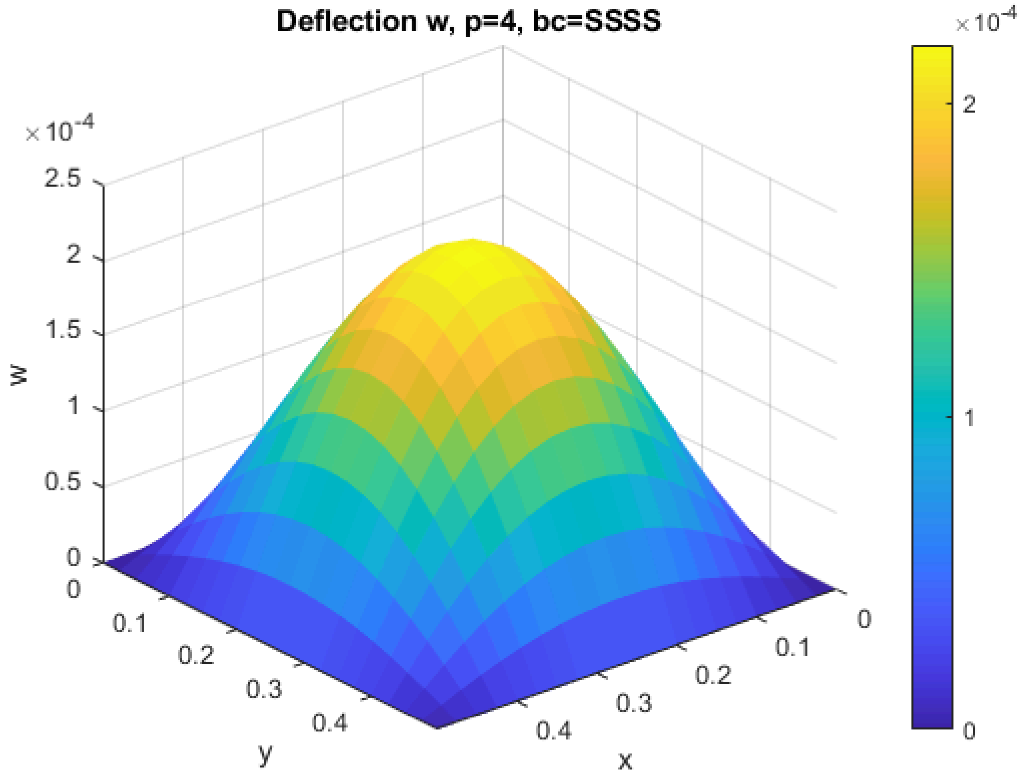

Through-thickness stresses at center, p=3, bc=SSSS
σxx vs z
σyy vs z
σxy vs z
×10^-3
z (m)
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-5
0
5
σxx
×10^6
σyy
×10^6
σxy
×10^-10

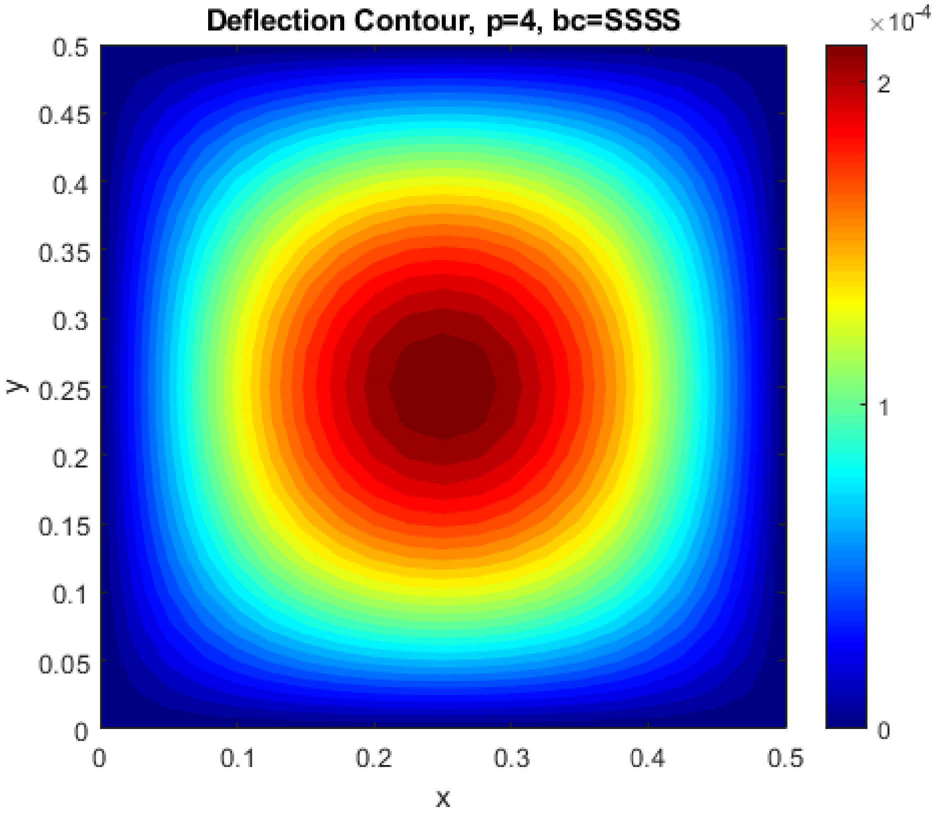

Deflection w, p=3, bc=CFFF
×10^-3
8
6
4
2
0
w
0
0.1
0.2
0.3
0.4
y
0
0.1
0.2
0.3
0.4
x
×10^-3
6
5
4
3
2
1
0

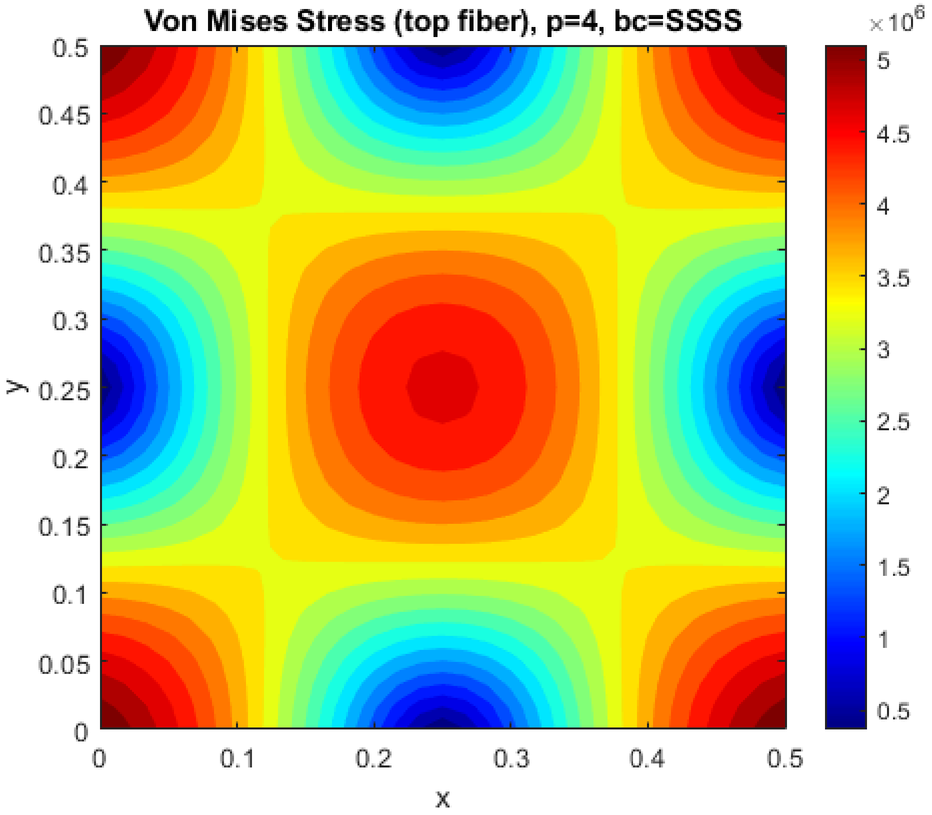

Deflection Contour, p=3, bc=CFFF
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
y
0
0.1
0.2
0.3
0.4
0.5
x
×10^-3
6
5
4
3
2
1
0

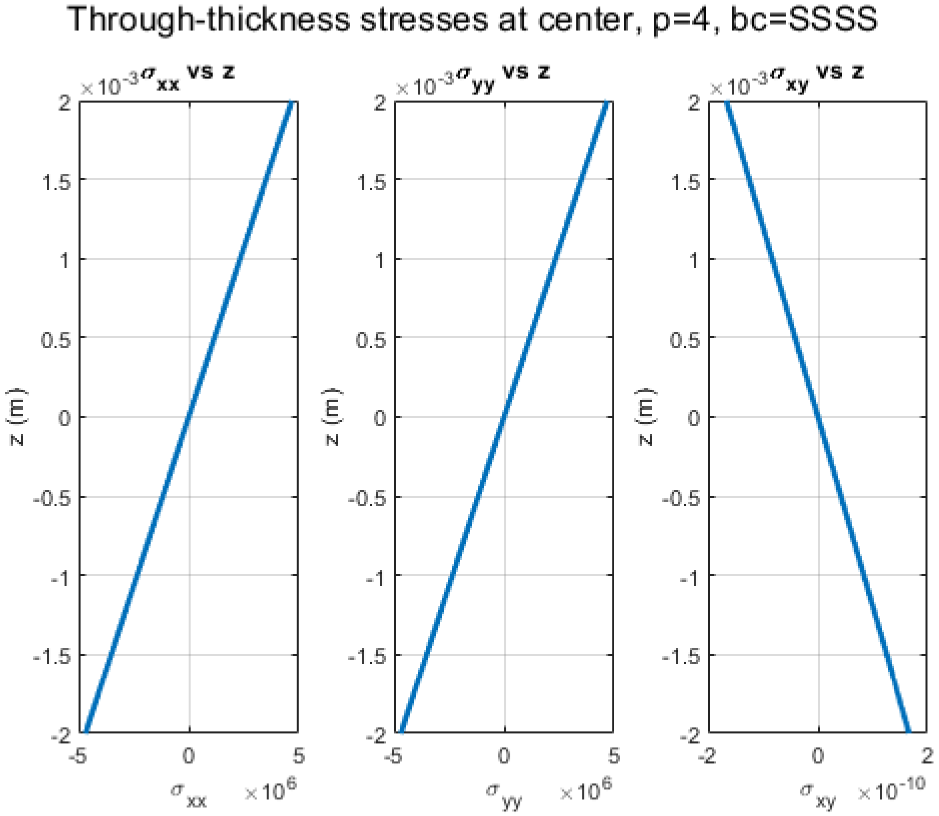

Von Mises Stress (top fiber), p=3, bc=CFFF
y
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
x
0
0.1
0.2
0.3
0.4
0.5
×10^7
3
2.5
2
1.5
1
0.5

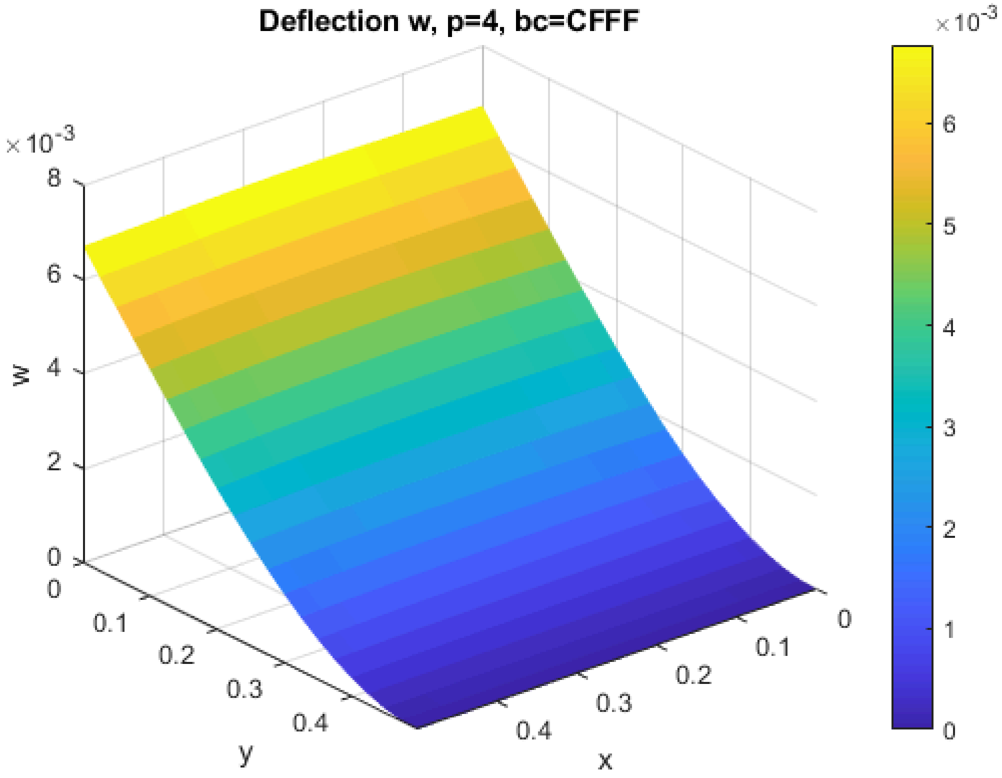

Through-thickness stresses at center, p=3, bc=CFFF
×10^-3 σxx vs z
×10^-3 σyy vs z
×10^-3 σxy vs z
z (m)
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-5
0
5
σxx ×10^6
-2
0
2
σyy ×10^7
-5
0
5
σxy ×10^-9


Deflection w, p=4, bc=SSSS
×10^-4
2.5
2
1.5
1
0.5
0
w
0
0.1
0.2
0.3
0.4
y
0
0.1
0.2
0.3
0.4
x
×10^-4
2
1
0

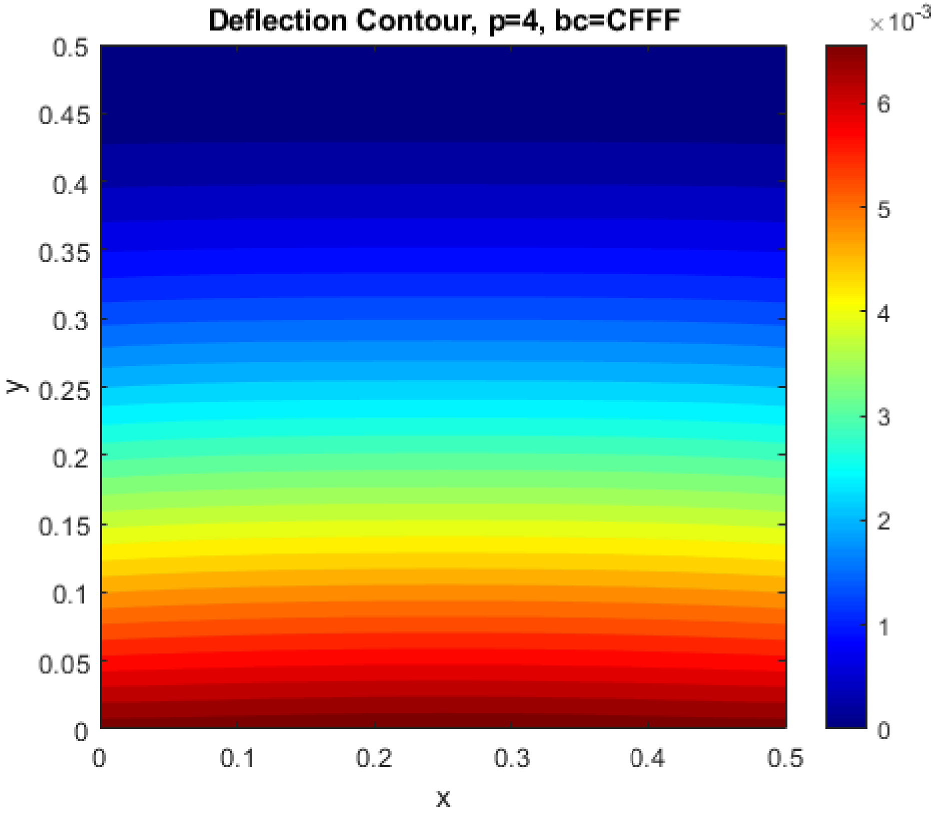

Deflection Contour, p=4, bc=SSSS
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
y
0
0.1
0.2
0.3
0.4
0.5
x
×10^-4
2
1
0


Von Mises Stress (top fiber), p=4, bc=SSSS
×10^6
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
y
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
x
0
0.1
0.2
0.3
0.4
0.5

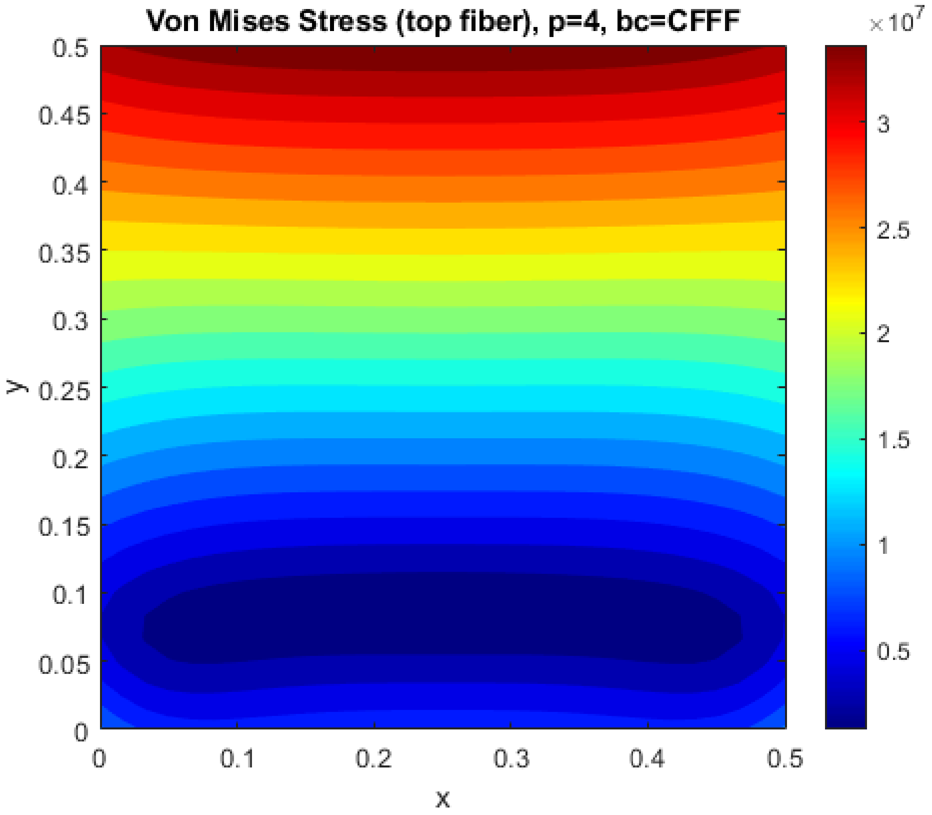

Through-thickness stresses at center, p=4, bc=SSSS
×10^-3
σxx vs z
σyy vs z
σxy vs z
z (m)
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
σxx ×10^6
σyy ×10^6
-5
0
5
σxy ×10^-10
-2
0
2

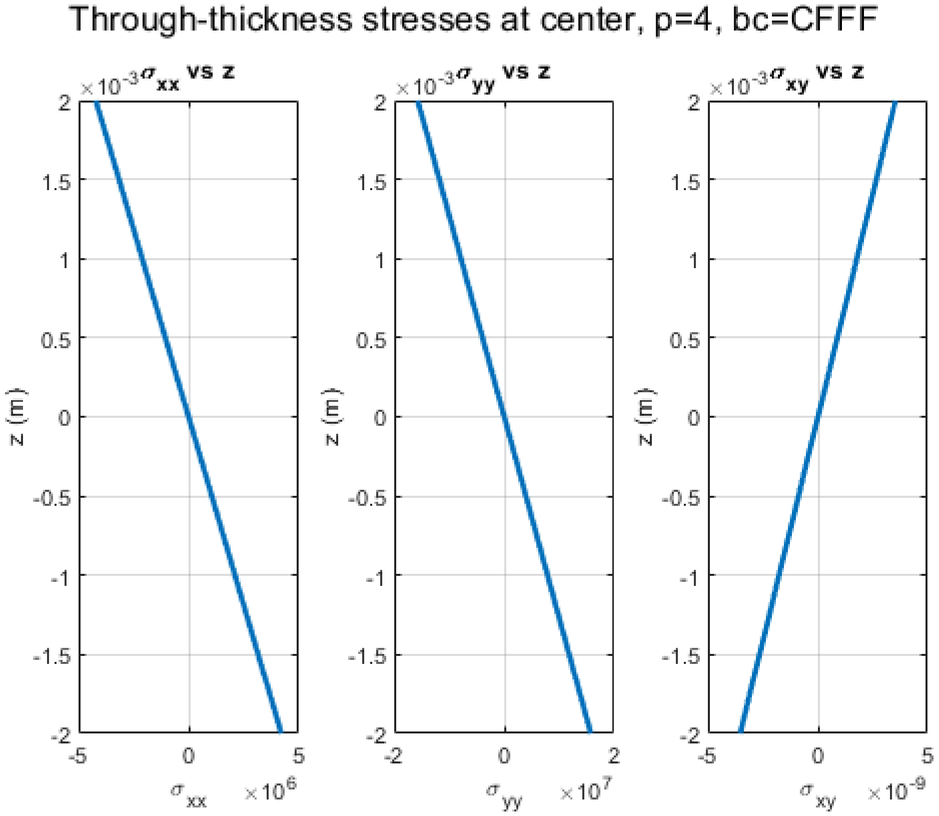


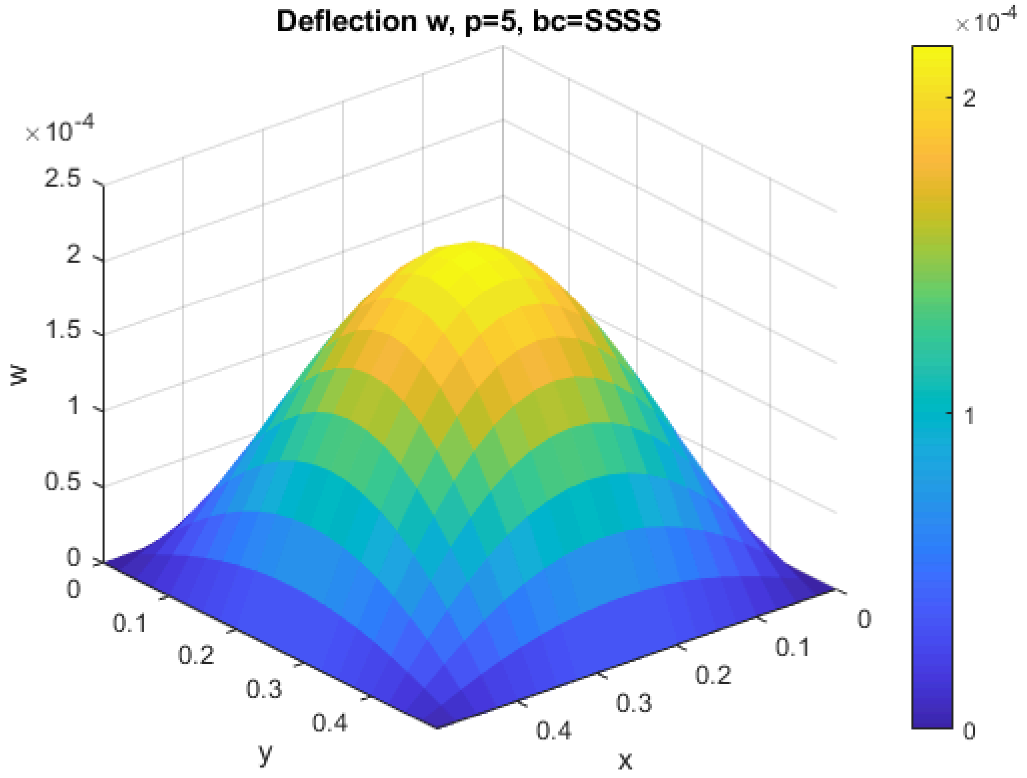
Deflection w, p=4, bc=CFFF



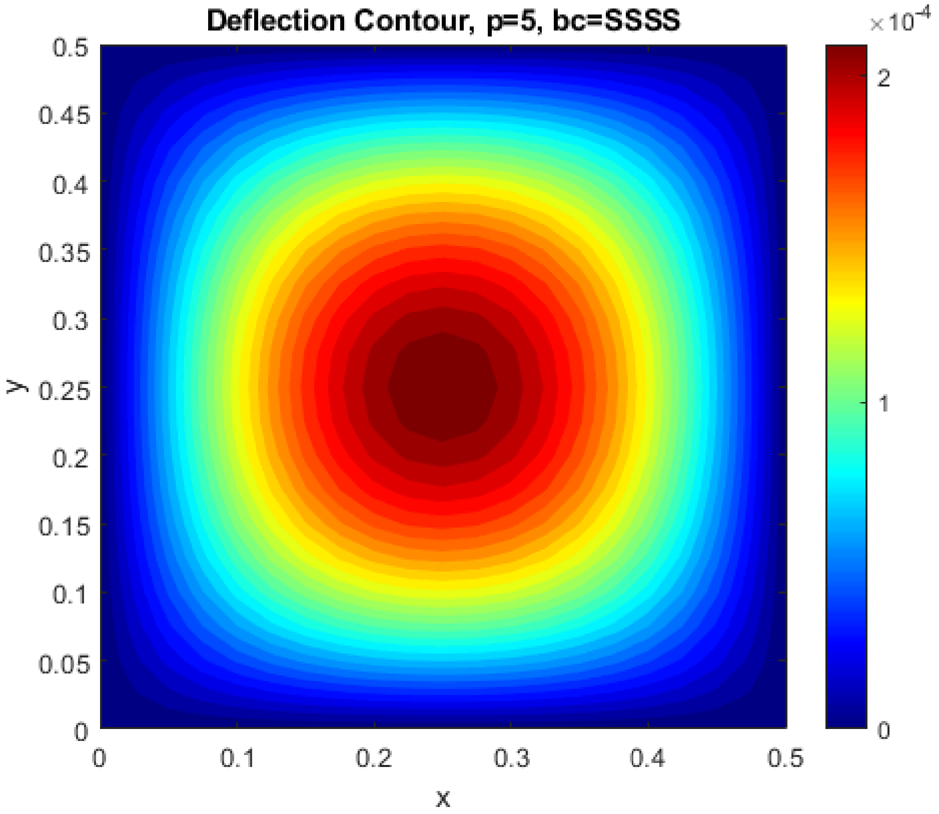
Deflection Contour, p=4, bc=CFFF


Von Mises Stress (top fiber), p=4, bc=CFFF
y
x
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0
0.1
0.2
0.3
0.4
0.5
×10^7
3
2.5
2
1.5
1
0.5

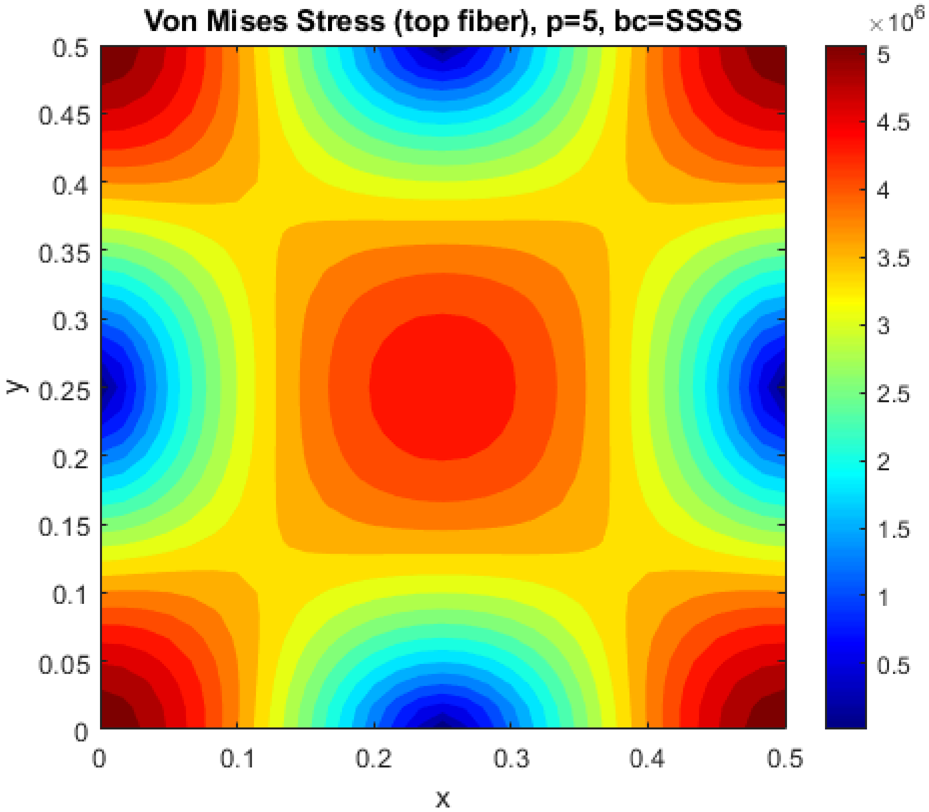

Through-thickness stresses at center, p=4, bc=CFFF
σxx vs z
σyy vs z
σxy vs z
×10^-3
z (m)
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
σxx ×10^6
-5
0
5
σyy ×10^7
-2
0
2
σxy ×10^-9
-5
0
5

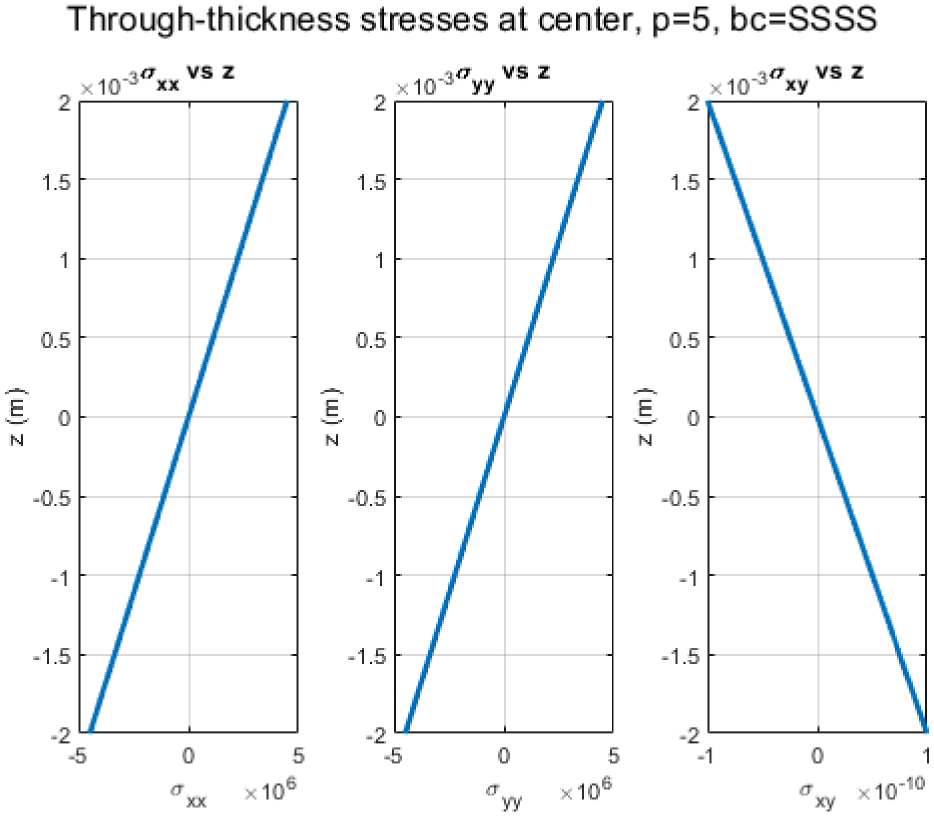


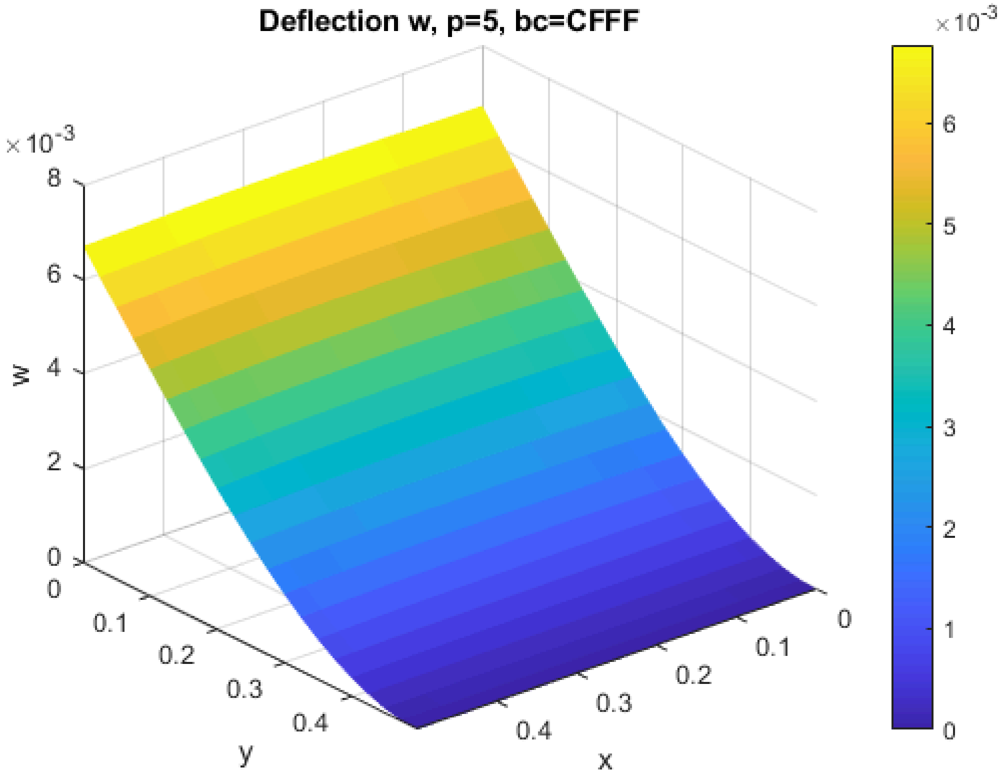
Deflection w, p=5, bc=SSSS
×10^-4
2.5
2
1.5
1
0.5
0
w
0
0.1
0.2
0.3
0.4
y
0
0.1
0.2
0.3
0.4
x
×10^-4
2
1
0

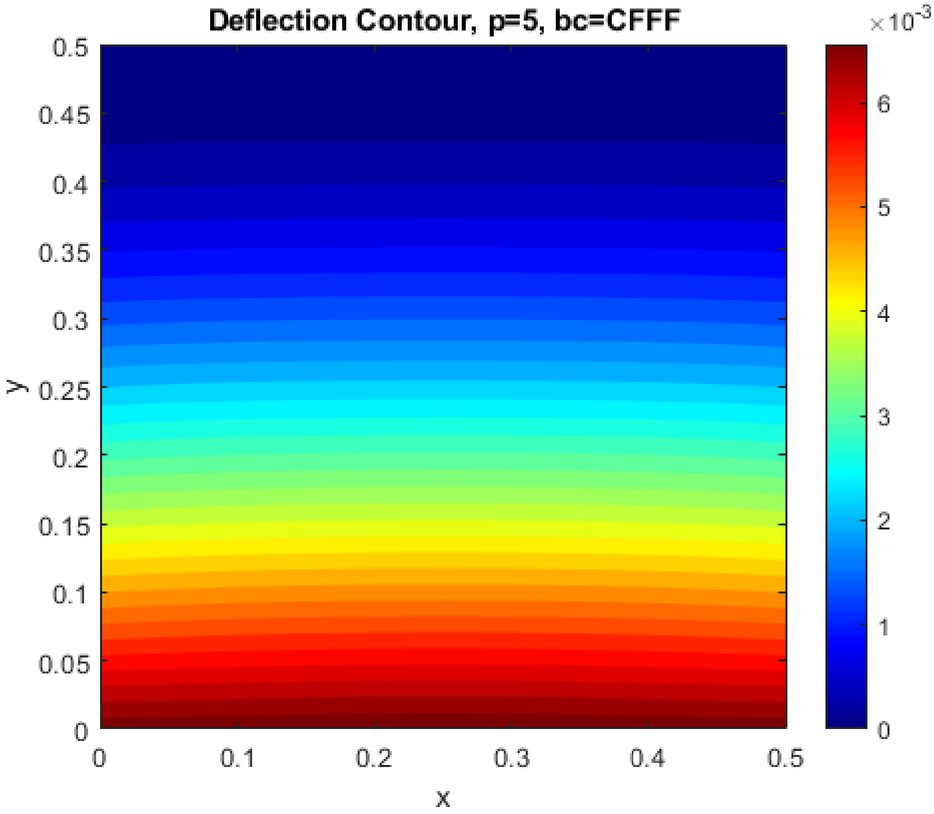


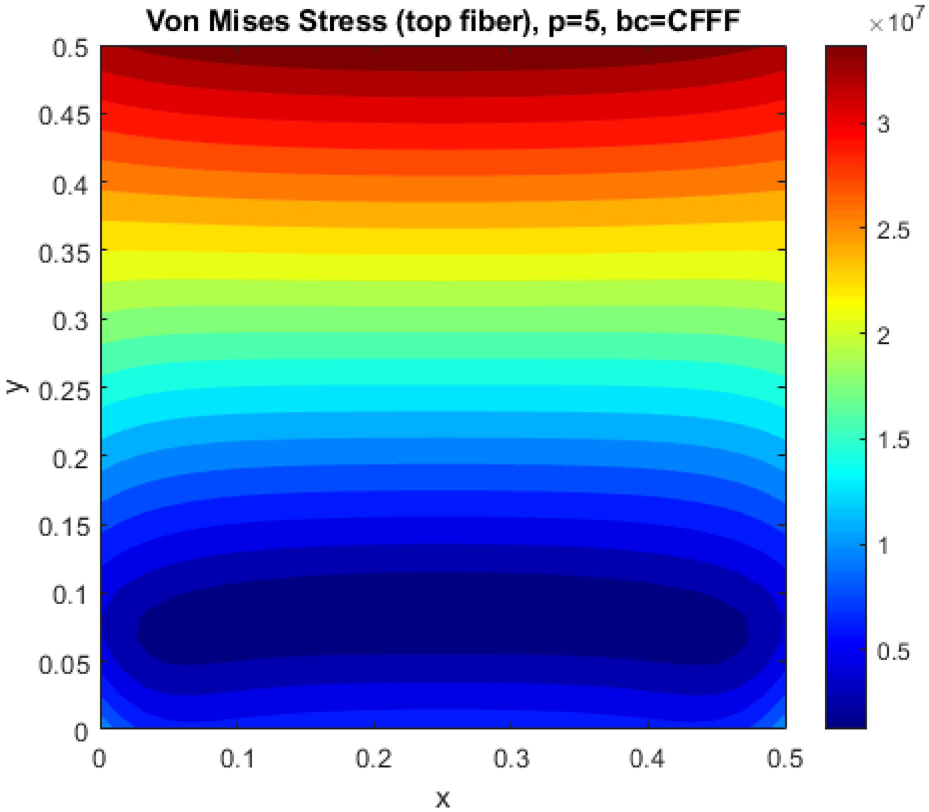
Deflection Contour, p=5, bc=SSSS
0.5
0.45
0.4
0.35
0.3
y
0.25
0.2
0.15
0.1
0.05
0
0
0.1
0.2
0.3
0.4
0.5
x
×10^-4
2
1
0

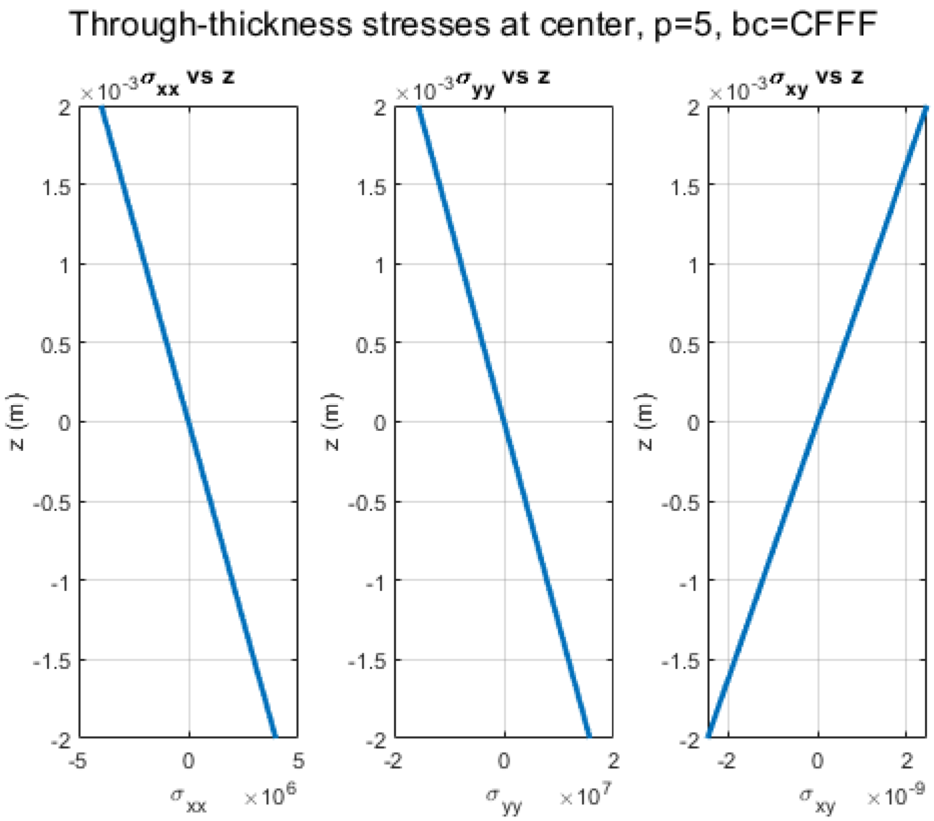

Von Mises Stress (top fiber), p=5, bc=SSSS
×10^6
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
y
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
x
0
0.1
0.2
0.3
0.4
0.5


Through-thickness stresses at center, p=5, bc=SSSS
×10^-3 σxx vs z
×10^-3 σyy vs z
×10^-3 σxy vs z
z (m)
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-5
0
5
σxx ×10^6
σyy ×10^6
-1
0
1
σxy ×10^-10


Deflection w, p=5, bc=CFFF
×10^-3
8
6
4
2
0
w
0
0.1
0.2
0.3
0.4
y
0
0.1
0.2
0.3
0.4
x
×10^-3
6
5
4
3
2
1
0


Deflection Contour, p=5, bc=CFFF
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
y
0
0.1
0.2
0.3
0.4
0.5
x
×10^-3
6
5
4
3
2
1
0





Published with MATLAB® R2023b