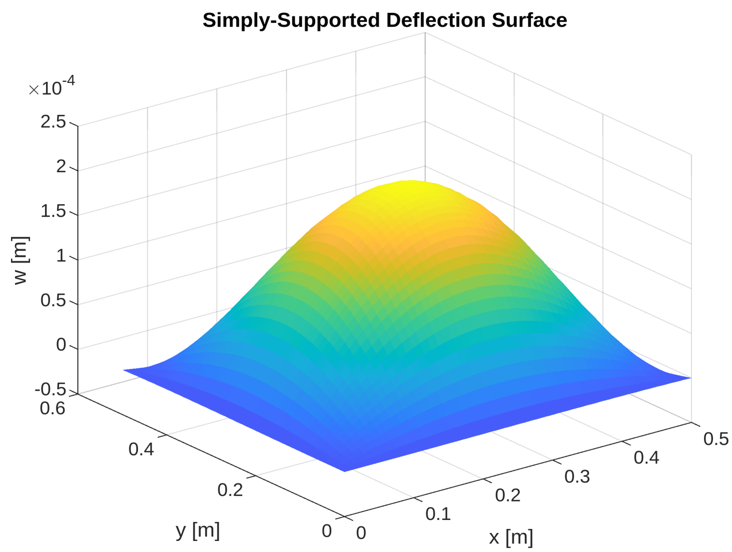
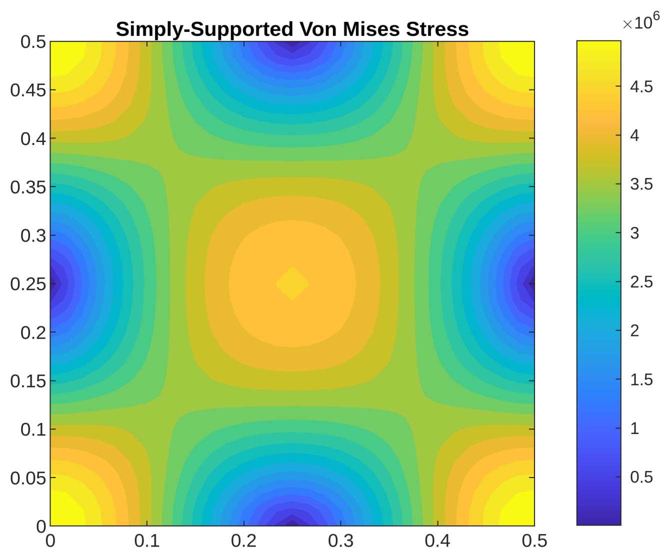
```
clear; clc; close all;

% Plate & material properties
a = 0.5;       % plate size in x [m]
b = 0.5;       % plate size in y [m]
h = 0.004;     % thickness [m]
E = 200e9;     % Young's modulus [Pa]
nu = 0.3;      % Poisson's ratio
q0 = 1000;     % uniform load [N/m^2]
D  = E*h^3/(12*(1-nu^2));
sigma_yield = 450e6; % yield stress [Pa]

% Mesh parameters
n_el = 50;
x_nodes = linspace(0,a,n_el+1);
y_nodes = linspace(0,b,n_el+1);

% Gauss‑Legendre quadrature (3‑point)
xi = [-sqrt(3/5),0,sqrt(3/5)];
wi = [5/9,8/9,5/9];

% Degrees of freedom
dps = 2*n_el + 2;
ND  = dps^2;

% Allocate global stiffness & load
K = zeros(ND,ND);
F = zeros(ND,1);

% Assemble K and F
for ex = 1:n_el
  for ey = 1:n_el
    x0 = x_nodes(ex);   x1 = x_nodes(ex+1);
    y0 = y_nodes(ey);   y1 = y_nodes(ey+1);
    J  = (x1-x0)*(y1-y0)/4;
    loc2glob = zeros(16,1);
    idx = 0;
    for i_loc = 0:3
      for j_loc = 0:3
        idx = idx + 1;
        ix = (ex-1)*2 + i_loc;
        jy = (ey-1)*2 + j_loc;
        loc2glob(idx) = ix*dps + jy + 1;
      end
    end
    for i=1:3
      for j=1:3
        ksi = xi(i);  wksi = wi(i);
        eta = xi(j);  weta = wi(j);
        x = 0.5*(x1+x0) + 0.5*(x1-x0)*ksi;
```

```
        y = 0.5*(y1+y0) + 0.5*(y1-y0)*eta;
        wght = wksi*weta*J;
        [Nx,dNx,d2Nx] = hermite1d(x,x0,x1);
        [Ny,dNy,d2Ny] = hermite1d(y,y0,y1);
        phi   = zeros(16,1);
        phixx = zeros(16,1);
        phiyy = zeros(16,1);
        phixy = zeros(16,1);
        pidx = 0;
        for ii=1:4
          for jj=1:4
            pidx = pidx + 1;
            phi(pidx)   = Nx(ii)*Ny(jj);
            phixx(pidx) = d2Nx(ii)*Ny(jj);
            phiyy(pidx) = Nx(ii)*d2Ny(jj);
            phixy(pidx) = dNx(ii)*dNy(jj);
          end
        end
        for a_loc=1:16
          A = loc2glob(a_loc);
          for b_loc=1:16
            B = loc2glob(b_loc);
            K(A,B) = K(A,B) + D*( ...
              phixx(a_loc)*phixx(b_loc) + ...
              phiyy(a_loc)*phiyy(b_loc) + ...
              nu*(phixx(a_loc)*phiyy(b_loc) + phiyy(a_loc)*phixx(b_loc)) +
...
              2*(1-nu)*phixy(a_loc)*phixy(b_loc) ) * wght;
          end
          F(A) = F(A) + q0 * phi(a_loc) * wght;
        end
      end
    end
  end
end

% Simply-supported BC: w=0 on all edges
fixed_ss = [];
for i=0:n_el
  fixed_ss(end+1) = (i*2)*dps +      1;
  fixed_ss(end+1) = (i*2)*dps + (n_el*2) + 1;
end
for j=0:n_el
  fixed_ss(end+1) =      1 + (j*2);
  fixed_ss(end+1) = (n_el*2)*dps + (j*2) + 1;
end
fixed_ss = unique(fixed_ss);
free_ss  = setdiff(1:ND, fixed_ss);

% Clamped at x=0: w, dw/dx, dw/dy = 0
```

```
fixed_cl = [];
for j=0:n_el
  fixed_cl(end+1) =      1 + (j*2);
  fixed_cl(end+1) = (1)*dps + (j*2) + 1;
  fixed_cl(end+1) =      1 + (j*2) + dps*0;
end
fixed_cl = unique(fixed_cl);
free_cl  = setdiff(1:ND, fixed_cl);

% Post-process for each boundary condition
postprocess(K,F,free_ss,'Simply-Supported');
```

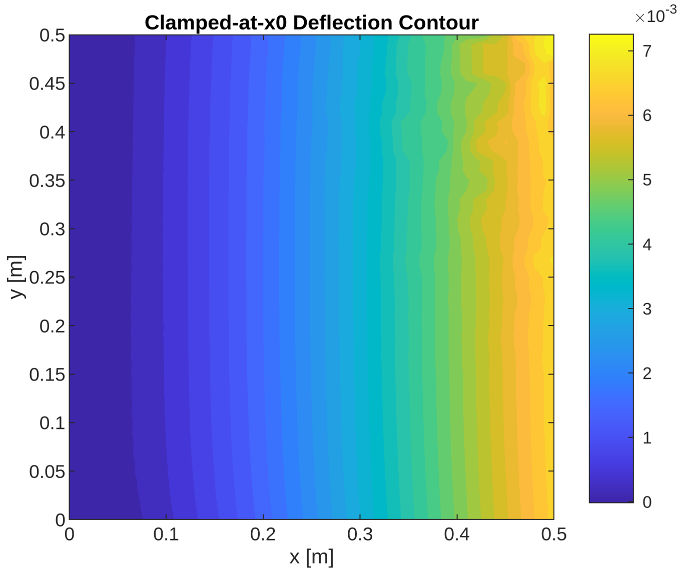


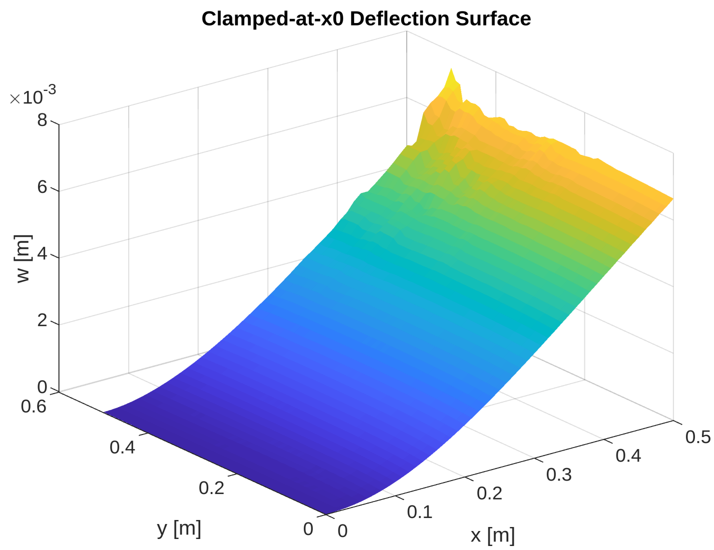

```
Simply-Supported: max σ_{vm} = 5.215e+06 Pa  at (0.000,0.000)
Simply-Supported: yield load q_{yield} = 8.629e+04 N/m^2
```



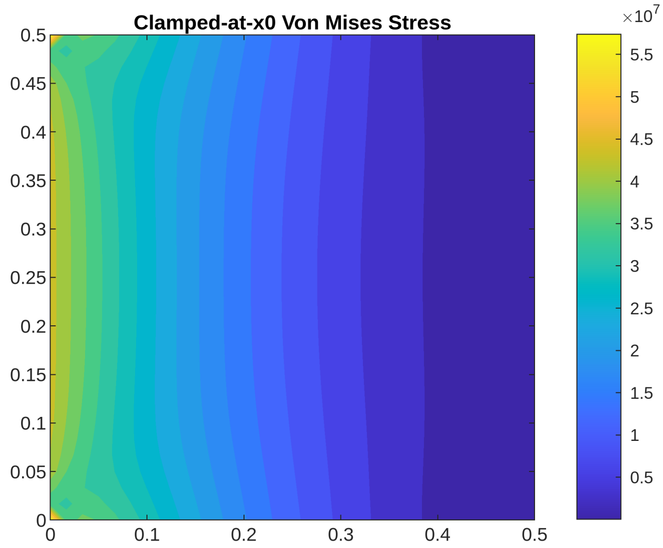

```
Simply-Supported: Strain energy U = 1.134730e-02 J
```

```
postprocess(K,F,free_cl, 'Clamped-at-x0');
```






```
Clamped-at-x0: max σ_{vm} = 6.027e+07 Pa  at (0.000,0.500)
Clamped-at-x0: yield load q_{yield} = 7.466e+03 N/m^2
```




```
Clamped-at-x0: Strain energy U = 3.408344e-01 J
```

```
%%-------------------------------------------------------------------
function [N,dN,d2N] = hermite1d(x,x0,x1)
  L  = x1-x0;  s=(x-x0)/L;
  H1 = 1-3*s^2+2*s^3;  H2 = s-2*s^2+s^3;
  H3 = 3*s^2-2*s^3;    H4 = -s^2+s^3;
  dH1 = -6*s+6*s^2;    dH2 = 1-4*s+3*s^2;
  dH3 = 6*s-6*s^2;     dH4 = -2*s+3*s^2;
  d2H1 = -6+12*s;      d2H2 = -4+6*s;
  d2H3 = 6-12*s;       d2H4 = -2+6*s;
  N   = [H1; L*H2; H3; L*H4];
  dN  = [dH1/L; dH2; dH3/L; dH4];
  d2N = [d2H1/L^2; d2H2/L; d2H3/L^2; d2H4/L];
end

function postprocess(K,F,free_dofs,bc_name)
  % retrieve parameters
  a = evalin('base','a');  b = evalin('base','b');
  h = evalin('base','h');  E = evalin('base','E');
  nu= evalin('base','nu'); q0= evalin('base','q0');
  D = evalin('base','D');  sigma_yield = evalin('base','sigma_yield');
  dps = evalin('base','dps');
  n_el= evalin('base','n_el');
  x_nodes = evalin('base','x_nodes');
  y_nodes = evalin('base','y_nodes');
  ND = numel(F);

  % solve reduced system
  Kff = K(free_dofs,free_dofs) + 1e-12*eye(numel(free_dofs));
  Ff  = F(free_dofs);
  af  = Kff\Ff;
  a_vec = zeros(ND,1);
```

```
a_vec(free_dofs) = af;

% Deflection contour
n_nodes = numel(x_nodes);
W = zeros(n_nodes,n_nodes);
for i=1:n_nodes
  x = x_nodes(i);
  for j=1:n_nodes
    y = y_nodes(j);
    ex = min(floor((i-1)/2)+1,n_el);
    ey = min(floor((j-1)/2)+1,n_el);
    x0 = x_nodes(ex); x1 = x_nodes(ex+1);
    y0 = y_nodes(ey); y1 = y_nodes(ey+1);
    [Nx,~,~] = hermite1d(x,x0,x1);
    [Ny,~,~] = hermite1d(y,y0,y1);
    wsum=0;
    for il=1:4
      for jl=1:4
        A = ((ex-1)*2+(il-1))*dps + ((ey-1)*2+(jl-1)) + 1;
        wsum = wsum + a_vec(A)*Nx(il)*Ny(jl);
      end
    end
    W(i,j)=wsum;
  end
end

figure;
contourf(x_nodes,y_nodes,W',30,'LineColor','none');
axis equal; axis([0 a 0 b]); colorbar;
title([bc_name ' Deflection Contour']);
xlabel('x [m]'); ylabel('y [m]');

[Xg,Yg] = meshgrid(x_nodes,y_nodes);
figure;
surf(Xg,Yg,W','EdgeColor','none');
title([bc_name ' Deflection Surface']);
xlabel('x [m]'); ylabel('y [m]'); zlabel('w [m]');

% Von Mises stress field and yield load
Nx_s=31; Ny_s=31;
xg=linspace(0,a,Nx_s); yg=linspace(0,b,Ny_s);
sigma_vm=zeros(Ny_s,Nx_s);
max_vm=0; pos=[0,0];
for ix=1:Nx_s
  for jy=1:Ny_s
    x=xg(ix); y=yg(jy);
    ex = min(floor(x/(x_nodes(2)-x_nodes(1)))+1,n_el);
    ey = min(floor(y/(y_nodes(2)-y_nodes(1)))+1,n_el);
    x0=x_nodes(ex); x1=x_nodes(ex+1);
    y0=y_nodes(ey); y1=y_nodes(ey+1);
```

```
      [Nx,dNx,d2Nx] = hermite1d(x,x0,x1);
      [Ny,dNy,d2Ny] = hermite1d(y,y0,y1);
      wxx=0; wyy=0; wxy=0;
      for il=1:4
        for jl=1:4
          A = ((ex-1)*2+(il-1))*dps + ((ey-1)*2+(jl-1)) + 1;
          c = a_vec(A);
          wxx = wxx + c*d2Nx(il)*Ny(jl);
          wyy = wyy + c*Nx(il)*d2Ny(jl);
          wxy = wxy + c*dNx(il)*dNy(jl);
        end
      end
      zt=h/2;
      sxx= -E/(1-nu^2)*zt*(wxx+nu*wyy);
      syy= -E/(1-nu^2)*zt*(wyy+nu*wxx);
      sxy= -E/(1+nu)*zt*wxy;
      vm = sqrt(0.5*((sxx-syy)^2+sxx^2+syy^2)+3*sxy^2);
      sigma_vm(jy,ix)=vm;
      if vm>max_vm, max_vm=vm; pos=[x,y]; end
    end
  end
  q_yield = (sigma_yield/max_vm)*q0;

  fprintf('\n%s: max σ_{vm} = %.3e Pa  at (%.3f,%.3f)\n', ...
    bc_name, max_vm, pos(1), pos(2));
  fprintf('%s: yield load q_{yield} = %.3e N/m^2\n', bc_name, q_yield);

  figure;
  contourf(xg,yg,sigma_vm,20,'LineColor','none');
  axis equal; axis([0 a 0 b]); colorbar;
  title([bc_name ' Von Mises Stress']);

  U = 0.5*a_vec' * K * a_vec;
  fprintf('%s: Strain energy U = %.6e J\n\n', bc_name, U);
end
```