8. MATLAB code demonstration

(1) *main* program

Quite similar to the main program developed in a script for solving hypoelasticity problems, the script mainly deals with

a. Specifying input parameters:

(a) μ_1 and K_1 are specified in the 15-th and 16-th numbers in the `mate` array.

(b) `nstphe`, and `nprthe` are the number of load increments, and the number of increment steps for one *.vtk file and are specified in the 17-th and 18-th numbers in the `mate` array.

(c) The function `GlobStif` has to be a function of current global displacment vector denoted as `w`.

b. Modifying the global stiffness matrix

c. Calculating the global residual vector

```
.
.
.
[ndime,nnode,nelem,nelnd,npres,ntrac,mate,coor,conn,pres,trac] = ReadInput(infile);
rglob = GlobTrac(ndime,nnode,nelem,nelnd,ntrac,coor,conn,trac);
nstpho = mate(13);
tol = 1e-4;
maxit = 100;
wglob = zeros(nnode*ndime,1);
for i = 2:nstpho+1
  faci = (i-1)/nstpho;
  erri = tol*100;
  niti = 0;
  disp(['Step: ' num2str(i) ', factor: ' num2str(faci)]);
  while(erri > tol && niti < maxit)
    niti = niti+1;
    kglob = GlobStif(ndime,nnode,nelem,nelnd,mate,coor,conn,,wglob);
    fglob = GlobResi(ndime,nnode,nelem,nelnd,coor,mate,wglob);
    bglob = faci*rglob-fglob;
    for j = 1:npres
      ir = ndime*(pres(1,j)-1)+pres(2,j);
      for ic = 1:ndime*nnode
        kglob(ir,ic) = 0;
      end
      kglob(ir,ir) = 1;
      bglob(ir,ir) = -wglob(ir) + faci*pres(3,j);
    end
    dwglob = kglob\bglob;
    dwglobsq = dwglob.'*dwglob;
    wglob = wglob + dwglob;
    wglobsq = wglob.'*wglob;
    erri = sqrt(dwglobsq/wglobsq);
    disp(['Iter: ' num2str(niti) ', err: ' num2str(erri)]);
  end
end
.
.
.
```

(2) *GlobStif* function

```
function kglob = GlobStif(ndime,nnode,nelem,nelnd,mate,coor,conn,wglob)
  kglob = zeros(ndim*nnode,ndim*nnode);
  for j = 1:nelem
    kel = ElemStif(j,ndime,nelnd,coor,conn,mate,wglob);
    for a = 1:nelnd
      for i = 1:ndime
        for b = 1:nelnd
          for k = 1:ndime
            ir = ndime*(conn(a,j)-1)+i;
            ic = ndime*(conn(b,j)-1)+k;
            kglob(ir,ic) = kglob(ir,ic)+kel(ndime*(a-1)+i,ndime*(b-1)+k);
          end
        end
      end
    end
  end
end
```

(3) *ElemStif* function

```
function kel = ElemStif(iel,ndime,nelnd,coor,conn,mate,wglob)
  kel = zeros(ndime*nelnd,ndime*nelnd);
  coorie = zeros(ndime,nelnd);
  wie = zeros(ndime,nelnd);
  xii = zeros(ndime,1);
  F = zeros(ndime,ndime);
  dxdxi = zeros(ndime,ndime);
  dNdx = zeros(nelnd,ndime);
  dNdy = zeros(nelnd,ndime);
  M = numIntegPt(ndime,nelnd);
  xi = IntegPt(ndime,nelnd,M);
  w = IntegWt(ndime,nelnd,M);
  for a = 1:nelnd
    for i = 1:ndime
      coorie(i,a) = coor(i,conn(a,iel));
      wie(i,a) = wglob(ndime*(conn(a,iel)-1)+i);
    end
  end
  for im = 1:M
    for i = 1:ndime
      xii(i) = xi(i,im);
    end
    dNdxi = ShpFuncDeri(nelnd,ndime,xii);
    dxdxi(:) = 0;
    for i = 1:ndime
      for j = 1:ndime
        for a = 1:nelnd
          dxdxi(i,j) = dxdxi(i,j)+coorie(i,a)*dNdxi(a,j);
        end
      end
    end
    dxidx = inv(dxdxi);
    jcb = det(dxdxi);
    dNdx(:) = 0;
    for a = 1:nelnd
      for i = 1:ndime
        for j = 1:ndime
```

```
          dNdx(a,i) = dNdx(a,i)+dNdxi(a,j)*dxidx(j,i);
        end
      end
    end
    F(:) = 0;
    for i = 1:ndime
      for j = 1:ndime
        if(i == j)
          F(i,j) = 1;
        end
        for a = 1:nelnd
          F(i,j) = F(i,j)+wie(i,a)*dNdx(a,j);
        end
      end
    end
    jhat = det(F);
    B = F*F.';
    Finv = inv(F);
    dNdy(:) = 0;
    for a = 1:nelnd
      for i = 1:ndime
        for j = 1:ndime
          dNdy(a,i) = dNdy(a,i)+dNdx(a,j)*Finv(j,i);
        end
      end
    end
    tau = KchStrs(ndime,mate,B,jhat);
    dsde = MatStif(ndime,mate,epsi);
    for a = 1:nelnd
      for i = 1:ndime
        for b = 1:nelnd
          for k = 1:ndime
            ir = ndime*(a-1)+i;
            ic = ndime*(b-1)+k;
            for j = 1:ndime
              for l = 1:ndime
                kel(ir,ic) = kel(ir,ic)+dsde(i,j,k,l)*dNdy(a,j)*dNdy(b,l)*w(im)*jcb;
              end
              kel(ir,ic) = kel(ir,ic)-tau(i,j)*dNdy(a,k)*dNdy(b,j)*w(intpt)*jcb;
            end
          end
        end
      end
    end
  end
end
```

(4) *KchStrs* function

```
function tau = KchStrs(ndime,mate,B,jhat)
  tau = zeros(ndime,ndime);
  dlt = [ [1,0,0]; [0,1,0]; [0,0,1] ];
  mu1 = materialprops(15);
  K1 = materialprops(16);
  Bkk = trace(B);
  if (ndime == 2)
    Bkk = Bkk+1;
  end
  for i = 1:ndime
    for j = 1 : ndime
      tau(i,j) = mu1/jhat^(2/3)*(B(i,j)-Bkk/3*dlt(i,j))+K1*jhat*(jhat-1)*dlt(i,j);
    end
  end
end
```

(5) *MatStif* function

```
function dsde = MatStif(ndime,mate,B,jhat)
  dsde = zeros(ndime,ndime,ndime,ndime);
  dlt = [ [1,0,0]; [0,1,0]; [0,0,1] ];
  mu1 = materialprops(15);
  K1 = materialprops(16);
  Bqq = trace(B);
  if (ndime == 2)
    Bqq = Bqq+1;
  end
  for i = 1:ndime
    for j = 1:ndime
      for k = 1:ndime
        for l = 1:ndime
          dsde(i,j,k,l) = mu1/jhat^(2/3)*( dlt(i,k)*B(j,l)+B(i,l)*dlt(j,k)-...
            (2/3)*(B(i,j)*dlt(k,l)+B(k,l)*dlt(i,j))+...
            (2/3)*Bqq/3*dlt(i,j)*dlt(k,l) )+...
            K1*(2*jhat-1)*jhat* dlt(i,j) * dlt(k,l);
        end
      end
    end
  end
end
```

(6) *GlobResi* function

```
function fglob = GlobResi(ndime,nnode,nelem,nelnd,mate,coor,conn,wglob)
  fglob = zeros(ndime*nnode,1);
  for iel = 1:nelem
    fel = ElemResi(iel,ndime,nelnd,coor,conn,mate,wglob);
    for a = 1:nelnd
      for i = 1:ndime
        ir = ndime*(conn(a,ie)-1)+i;
        fglob(ir) = fglob(ir)+fel(ndime*(a-1)+i);
      end
    end
  end
end
```

(7) *ElemResi* function

```
function fel = ElemResi(iel,ndime,nelnd,coor,conn,mate,wglob)
  fel = zeros(ndime*nelnd,1);
  coorie = zeros(ndime,nelnd);
  wie = zeros(ndime,nelnd);
  xii = zeros(ndime,1);
```

```
F = zeros(ndime,ndime);
dxdxi = zeros(ndime,ndime);
dNdx = zeros(nelnd,ndime);
dNdy = zeros(nelnd,ndime);
M = numIntegPt(ndime,nelnd);
xi = IntegPt(ndime,nelnd,M);
w = IntegWt(ndime,nelnd,M);
for a = 1:nelnd
  for i = 1:ndime
    coorie(i,a) = coor(i,conn(a,iel));
    wie(i,a) = wglob(ndime*(conn(a,iel)-1)+i);
  end
end
for im = 1:M
  for i = 1:ndime
    xii(i) = xi(i,im);
  end
  dNdxi = ShpFuncDeri(nelnd,ndime,xii);
  dxdxi(:) = 0;
  for i = 1:ndime
    for j = 1:ndime
      for a = 1:nelnd
        dxdxi(i,j) = dxdxi(i,j)+coorie(i,a)*dNdxi(a,j);
      end
    end
  end
  dxidx = inv(dxdxi);
  jcb = det(dxdxi);
  dNdx(:) = 0;
  for a = 1:nelnd
    for i = 1:ndime
      for j = 1:ndime
        dNdx(a,i) = dNdx(a,i)+dNdxi(a,j)*dxidx(j,i);
      end
    end
  end
  F(:) = 0;
  for i = 1:ndime
    for j = 1:ndime
      if(i == j)
        F(i,j) = 1;
      end
      for a = 1:nelnd
        F(i,j) = F(i,j)+wie(i,a)*dNdx(a,j);
      end
    end
  end
  jhat = det(F);
  B = F*F.';
  Finv = inv(F);
  dNdy(:) = 0;
  for a = 1:nelnd
    for i = 1:ndime
      for j = 1:ndime
```

```
          dNdy(a,i) = dNdy(a,i)+dNdx(a,j)*Finv(j,i);
        end
      end
    end
    tau = KchStrs(ndime,mate,B,jhat);
    for a = 1:nelnd
      for i = 1:ndime
        ir = ndime*(a-1)+i;
          for j = 1:ndime
            fel(ir) = fel(ir)+tau(i,j)*dNdy(a,j)*w(im)*jcb;
          end
        end
      end
    end
  end
end
```