4.4 VTK files of meshes for dynamic problems in ParaView

1. In this section, the file rules of typical VTK data of meshes developed for dynamic problems in ParaView visualization are introduced. Please save your FEM results at each time step for dynamic problems in individual VTK files, such as:
   ```
   dyna1.vtk
   dyna2.vtk
   dyna3.vtk
   .
   .
   dyna9.vtk
   dyna10.vtk
   .
   .
   ```
2. Select all the relevant VTK files and drag all of them into the *Pipeline Browser* of ParaView.
3. In the builtin, you will see "dyna*", which means the common file name for the dynamic sequential data files.
4. After choosing the variable of SCALARS and "Surface With Edges" as a display style, click "Edit Color Map", and in the "Color Map Editor" click "Rescale to data range over all timesteps".
5. Now, we are ready to check the time-varying results of the chosen variable over all time steps. Just press the "Play" on the top toolbar to see the animation of the chosen variable variation in a particular duration.

4.5 Evaluating the boundary traction integrals

In this section, we need to address how to calculate boundary traction integrals in mesh edges and mesh surfaces in 2D and 3D problems respectively. Consider a nominal traction $\boldsymbol{t}$ with the unit of force per unit area acting on part of a solid and the body force $\boldsymbol{b}$ with the unit of force per unit volume acting throughout the solid, the expression for the *external nodal forces* can therefore be written as:

$$F_i^a = \int_V b_i \, N^a \, dV + \int_{\partial_2 V} t_i \, N^a \, dA \tag{4.44}$$

In the external nodal forces, we need to deal with volume integrals (as usual) for body forces and boundary integrals for tractions separately. Please see the MATLAB code demonstrations including the four functions evaluating the global nodal forces, number of nodes for each element face, node numbers of an element face, and the nodal forces of an element by the *GlobTrac* function, *numFaceNd* function, *FaceNd* function, and *ElemLoad* function respectively.

1. *GlobTrac* function
   ```
   function rglob = GlobTrac(ndime,nnode,nelem,nelnd,ntrac,mate,coor,conn,trac)
     rglob = zeros(ndime*nnode,1);
     nfcnd = numFaceNd(ndime,nelnd);
     coorif = zeros(ndime,nfcnd);
     for j = 1:ntrac
       iel = trac(1,j);
       ifc = trac(2,j);
       fcnd = FaceNd(ndime,nelnd,ifc);
       for a = 1:nfcnd
         for i = 1:ndime
           coorif(i,a) = coor(i,conn(fcnd(a),iel));
         end
       end
       rel = ElemLoad(ndime,nelnd,nfcnd,coorif,trac(:,j));
       for a = 1:nfcnd
         for i = 1:ndime
           ir = (conn(fcnd(a),iel)-1)*ndime+i;
   ```

```
        rglob(ir) = rglob(ir)+rel((a-1)*ndime+i);
      end
    end
  end
end
```

2. *numFaceNd* function

```
function nfcnd = numFaceNd(ndime,nelnd)
  if(ndime == 2)
    if(nelnd == 3 || nelnd == 4)
      nfcnd = 2;
    elseif (nelnd == 6 || nelnd == 8)
      nfcnd = 3;
    end
  elseif(ndime == 3)
    if(nelnd == 4)
      nfcnd = 3;
    elseif (nelnd == 10)
      nfcnd = 6;
    elseif (nelnd == 8)
      nfcnd = 4;
    elseif (nelnd == 20)
      nfcnd = 8;
    end
  end
end
```

3. *FaceNd* function

```
function fcnd = FaceNd(ndime,nelnd,ifc)
  fcnd = zeros(numFaceNd(ndime,nelnd),1);
  ind3 = [2,3,1];
  ind4 = [2,3,4,1];
  if(ndime == 2)
    if(nelnd == 3)
      fcnd(1) = ifc;
      fcnd(2) = ind3(ifc);
    elseif(nelnd == 6)
      fcnd(1) = ifc;
      fcnd(2) = ind3(ifc);
      fcnd(3) = ifc+3;
    elseif(nelnd == 4)
      fcnd(1) = ifc;
      fcnd(2) = ind4(ifc);
    elseif(nelnd == 8)
      fcnd(1) = ifc;
      fcnd(2) = ind4(ifc);
      fcnd(3) = ifc+4;
    end
  elseif(ndime == 3)
    if(nelnd == 4)
      if(ifc == 1)
        fcnd = [1,2,3];
      elseif(ifc == 2)
        fcnd = [1,4,2];
      elseif(ifc == 3)
        fcnd = [2,4,3];
      elseif(ifc == 4)
        fcnd = [3,4,1];
      end
    elseif(nelnd == 10)
      if(ifc == 1)
        fcnd = [1,2,3,5,6,7];
      elseif(ifc == 2)
        fcnd = [1,4,2,8,9,5];
      elseif(ifc == 3)
        fcnd = [2,4,3,9,10,6];
      elseif(ifc == 4)
        fcnd = [3,4,1,10,8,7];
      end
    elseif(nelnd == 8)
      if(ifc == 1)
        fcnd = [1,2,3,4];
      elseif(ifc == 2)
        fcnd = [5,8,7,6];
      elseif(ifc == 3)
        fcnd = [1,5,6,2];
      elseif(ifc == 4)
        fcnd = [2,3,7,6];
      elseif(ifc == 5)
        fcnd = [3,7,8,4];
      elseif(ifc == 6)
        fcnd = [4,8,5,1];
      end
    elseif(nelnd == 20)
      if(ifc == 1)
        fcnd = [1,2,3,4,9,10,11,12];
      elseif(ifc == 2)
        fcnd = [5,8,7,6,16,15,14,13];
      elseif(ifc == 3)
        fcnd = [1,5,6,2,17,13,18,9];
      elseif(ifc == 4)
        fcnd = [2,6,7,3,18,14,19,10];
      elseif(ifc == 5)
        fcnd = [3,7,8,4,19,15,20,11];
      elseif(ifc == 6)
        fcnd = [4,8,5,1,20,16,17,12];
      end
    end
  end
end
```

4. *ElemLoad* function

```
function rel = ElemLoad(ndime,nelnd,nfcnd,coorif,tracif)
  rel = zeros(ndime*nfcnd,1);
  xii = zeros(ndime-1,1);
  dxdxi = zeros(ndime,ndime-1);
  M = numIntegPt(ndime-1,nfcnd);
  xi = IntegPt(ndime-1,nfcnd,M);
  w = integWt(ndime-1,nfcnd,M);
  for im = 1:M
    for i = 1:ndime-1
```

```
      xii(i) = xi(i,im);
    end
    N = ShpFunc(nfcnd,ndime-1,xii);
    dNdxi = ShpFuncDeri(nfcnd,ndime-1,xii);
    dxdxi(:) = 0;
    for i = 1:ndime
      for j = 1:ndime-1
        for a = 1:nfcnd
          dxdxi(i,j) = dxdxi(i,j)+coorif(i,a)*dNdxi(a,j);
        end
      end
    end
    if(ndime == 2)
      jcb = sqrt(dxdxi(1,1)^2+dxdxi(2,1)^2);
    else
      jcb = sqrt( ...
        ((dxdxi(2,1)*dxdxi(3,2))-(dxdxi(2,2)*dxdxi(3,1)))^2+...
        ((dxdxi(1,1)*dxdxi(3,2))-(dxdxi(1,2)*dxdxi(3,1)))^2+...
        ((dxdxi(1,1)*dxdxi(2,2))-(dxdxi(1,2)*dxdxi(2,1)))^2);
    end
    for a = 1:nfcnd
      for i = 1:ndime
        ir = ndime*(a-1)+i;
        rel(ir) = rel(ir)+N(a)*tracif(2+i)*w(im)*jcb;
      end
    end
  end
end
```