# Step1: generate the \*.Tet model by using the TetGen

https://github.com/zhangty019/S3\_DeformFDM/blob/main/Tet\_generation\_S3.zip

# Step 2: generate the \*.INP file as the input model for ABAQUS

* Example code

|  |
| --- |
| void fileIO::exportMeshtoAbaqusFEM(QMeshPatch\* model, std::string path) {  std::string filename = model->patchName;  const char\* c = filename.c\_str();  char\* cstr = new char[filename.length() + 1];  strcpy(cstr, filename.c\_str());  const char\* split = ".";  char\* p = strtok(cstr, split);  char output\_filename[256];  strcpy(output\_filename, path.c\_str());  strcat(output\_filename, cstr);  char filetype[64];  strcpy(filetype, ".inp");  strcat(output\_filename, filetype);  std::ofstream abaqusOutput(output\_filename);  if (!abaqusOutput)  std::cerr << "Sorry!We were unable to build the file NodeSelect!\n";  abaqusOutput << "\*Part, name=" << filename << std::endl << "\*Node" << std::endl;  //First go through all the nodes  double pp[3] = { 0 }, ppl[3] = { 0 }, ppr[3] = { 0 };  int node\_index = 0;  for (GLKPOSITION Pos = model->GetNodeList().GetHeadPosition(); Pos;) {  QMeshNode\* CheckNode = (QMeshNode\*)model->GetNodeList().GetNext(Pos);  CheckNode->GetCoord3D(pp[0], pp[1], pp[2]);  node\_index++;  abaqusOutput << node\_index << "," << pp[0] << "," << pp[1] << "," << pp[2] << std::endl;  }  abaqusOutput << "\*Element, type = C3D4" << std::endl;  int node\_indexTet[4] = { 0 };  int tet\_index = 0;  for (GLKPOSITION Pos = model->GetTetraList().GetHeadPosition(); Pos;) {  QMeshTetra\* Tetra = (QMeshTetra\*)model->GetTetraList().GetNext(Pos);  tet\_index++;  for (int i = 0; i < 4; i++) node\_indexTet[i] = Tetra->GetNodeRecordPtr(i + 1)->GetIndexNo();  abaqusOutput << tet\_index << ",";  for (int i = 0; i < 3; i++) abaqusOutput << node\_indexTet[i] << ",";  abaqusOutput << node\_indexTet[3];  abaqusOutput << std::endl;  }  abaqusOutput << "\*End Part" << std::endl;  abaqusOutput.close();  } |

* Format of \*.INP

|  |
| --- |
| \*Part, name=test1  \*Node  1,0.126982,48,37.5583  2,-0.163012,48,38.5667  …  \*Element, type = C3D4  1,1505,1509,12319,814  2,21390,23232,22444,23475 |

# Step3: conduct isotropic FEA

Notes: please go through the FEA course before the below operation.

https://www.youtube.com/watch?v=SOiBbmGw02Q&t=20s

|  |  |
| --- | --- |
| Input \*.inp model | A screenshot of a computer  Description automatically generated |
| Define the parameters of material |  |
| Define the boundary condition of FEA |  |
| Run the FEA |  |
| Add and run the Macro to output the stress field of model, then store the file into “../DataSet/fem\_result” with the same name of the model. |  |

* Macro (python code)

|  |
| --- |
| * # output isotropic FEA result * def Macro4(): * import section * import regionToolset * import displayGroupMdbToolset as dgm * import part * import material * import assembly * import step * import interaction * import load * import mesh * import optimization * import job * import sketch * import visualization * import xyPlot * import displayGroupOdbToolset as dgo * import connectorBehavior * import sys * import os * file1 = open("C:/Users/zhang/Desktop/FEM\_result.txt","w+") * print(os.getcwd) * odb = session.odbs['C:/SIMULIA/temp/Job-1.odb'] * Stress = odb.steps['Step-1'].frames[1].fieldOutputs['S'].getSubset(position=CENTROID).values * sz = len(Stress) * for ip in range(0,sz): * file1.write(str(ip+1)+',') * douout = Stress[ip].maxPrincipal * strout = "%G" %(douout) * file1.write(strout) * for index in range(0,6):   + file1.write(',')   + douout = Stress[ip].data[index]   + strout = "%G" %(Stress[ip].data[index])   + file1.write(strout) * file1.write('\n') * file1.close; |

# Done!