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

- Please note that the example code is based on the *.tet mesh generated by the previous step and it should be located in the **FileIO.cpp**
- 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();

}
```

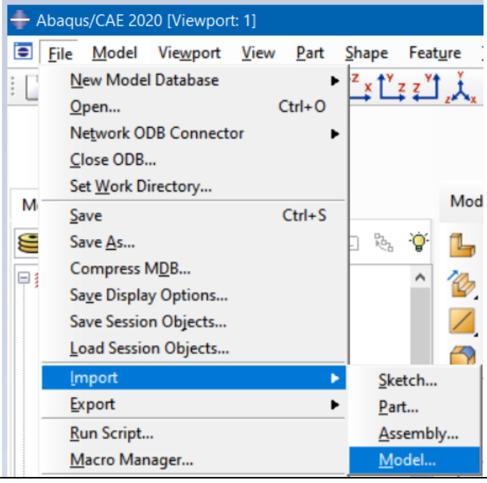
- Format of *.INP. Please check below link for more details
- https://classes.engineering.wustl.edu/2009/spring/mase5513/abaqus/docs/v6.5/books/gss/default.htm?startat=ch02s02.html

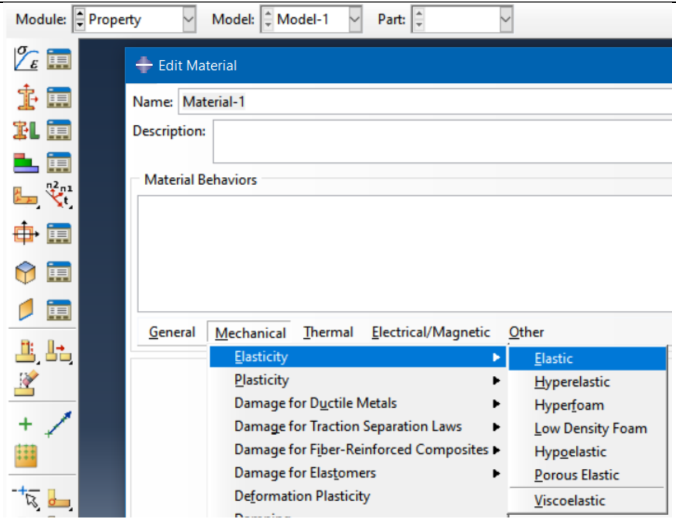
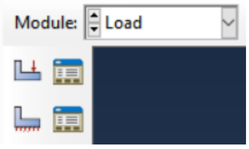
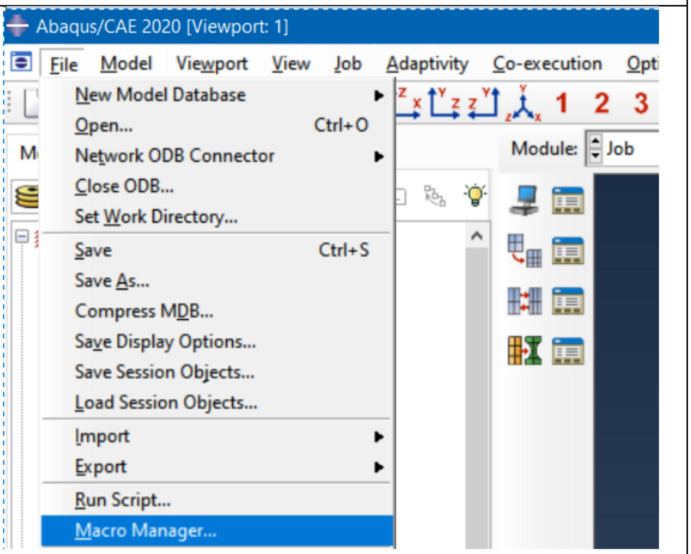
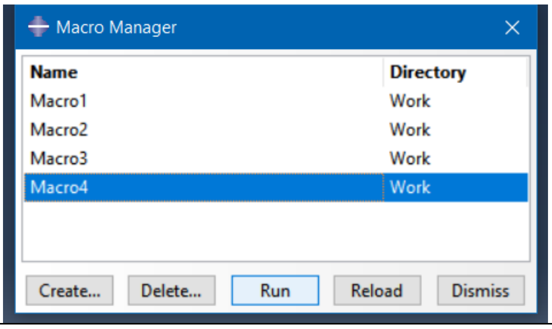
```
*Part, name=test1
*Node
1,0.126982,48,37.5583
2,-0.163012,48,38.5667
…
*Element, type = C3D4
1,1505,1509,12319,814
2,21390,23232,22444,23475
```

Step3: conduct isotropic FEA

Please go through the basic FEA course before the below operation.

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

Input *.inp model	

Define the parameters of material	 Module: Property Model: Model-1 Part: Name: Material-1 Edit Material Description: Material Behaviors General Mechanical Thermal Electrical/Magnetic Other Elasticity Plasticity Damage for Ductile Metals Damage for Traction Separation Laws Damage for Fiber-Reinforced Composites Damage for Elastomers Deformation Plasticity Elastic Hyperelastic Hyperfoam Low Density Foam Hypoelastic Porous Elastic Viscoelastic
Define the boundary condition of FEA	 Module: Load
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.	 Module: Job Abaqus/CAE 2020 [Viewport: 1] File Model Viewport View Job Adaptivity Co-execution Opti New Model Database Open... Ctrl+O Network ODB Connector Close ODB... Set Work Directory... Save Ctrl+S Save As... Compress MDB... Save Display Options... Save Session Objects... Load Session Objects... Import Export Run Script... Macro Manager...  Macro Manager Create... Delete... Run Reload Dismiss Name Directory Macro1 Work Macro2 Work Macro3 Work Macro4 Work

- 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!