1. 已知条件
2. 划分单元和节点
   1. 节点数 int numNode
   2. 节点 MatrixXd nodeCoord[numNode\*3]:保存每个节点的x,y,z坐标
   3. 单元数int numEle
   4. 单元 MatrixXd eleNode[numEle \*8]:保存每个单元的节点编号
3. 确定自由度总数和被约束的自由度,则同时确定激活的自由度
   1. 自由度总数int numDOF= numNode\*3
   2. 总自由度VectorXd DOF[numDOF]
   3. 被约束的自由度数量 int numRestrainedDOF
   4. 被约束的自由度 VectorXd restrainedDOF[numRestrainedDOF]
   5. 激活的自由度数量 int numActiveDOF
   6. 激活的自由度 VectorXd activeDOF[numActiveDOF]
4. 确定载荷
   1. 体积力Vector3d fv
   2. 由体积力、面积力等计算等效节点载荷VectorXd force[numDOF](本例中只有体积力)
5. 计算过程
6. 写出刚度矩阵stiffness(numDOF\* numDOF)
   1. 写出每个单元包含的节点Vector2d noindex
   2. 写出每个单元(包含的节点)对应的自由度eleDof(4)
   3. ~~计算C,S,写出eleK(4\*4)~~
   4. Stiffness(eleDof(j),eleDof(k))=stiffness+eleK(j,k)
7. 划行划列法去掉约束自由度对应的行列
8. ~~计算节点位移displacement(activeDOF)= stiffness(activeDOF\*activeDOF)~~~~-1~~~~×force(activeDOF)~~
9. 计算单元轴向应力σ=E/L\*[-C,-S,C,S]\*displacement(eleDOF) (eleDOF为单元对应的自由度)
10. 计算支座反力reaction(restrainedDOF)=stiffness.row(restrainedDOF )\*displacement