有限元分析期末作业

S230200252 张秋源

1 简介

我制作了一个使用 C++的四面体四节点单元的有限元求解器，可使用自定义的模型，效率较高。处理带有 3446 个节点，12585 个单元的模型仅仅耗时 570ms。

该程序通过在 json 文件中配置单元的材料，节点位置，单元包含的节点序号，位移约束以及节点上的输入力来完成有限元分析计算。该程序通过 json 文件进行结果的输出，其中包括每一个节点的位移量，以及每个单元所受到的应力以及应变。并且通过第四强度理论（应变能理论，即 Von Mises）计算每一个节点的应变能。

$$\sigma = \sqrt{\frac{1}{2}((\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2)} \leq [\sigma]$$

1.1 输入输出文件

输入输出主要通过 json 文件来完成，现通过一个 12 节点的简单网格划分，来解释文件中的参数。

1.1.1 输入文件

```
{
    "nodes":[        （每个节点的三维坐标）
        [0, 0, 0],
        [0, 3, 0],
        [0, 3, 3],
        [0, 0, 3],
        [3, 0, 0],
        [3, 3, 0],
        [3, 3, 3],
        [3, 0, 3],
        [6, 0, 0],
        [6, 3, 0],
        [6, 3, 3],
        [6, 0, 3]
    ],
```

```
    "elements":[        （单元的序号以及对应的 4 个节点）

        {"number":100, "node":[1,2,3,6]},

        {"number":101, "node":[1,0,3,5]},

        {"number":102, "node":[1,3,5,6]},

        {"number":103, "node":[3,4,6,7]},

        {"number":104, "node":[0,3,4,5]},

        {"number":105, "node":[3,4,5,6]},

        {"number":106, "node":[5,6,7,10]},

        {"number":107, "node":[5,4,7,9]},

        {"number":108, "node":[5,7,9,10]},

        {"number":109, "node":[7,8,10,11]},

        {"number":110, "node":[4,7,8,9]},

        {"number":111, "node":[7,8,9,10]}

    ],

    "constrains":[        （位移约束）

        {"node":0, "displace":[0, 0, 0]},

        {"node":1, "displace":[0, 0, 0]},

        {"node":2, "displace":[0, 0, 0]},

        {"node":3, "displace":[0, 0, 0]}

    ],

    "forces":[        （在节点上施加的力）

        {"node":11, "forceval":[1.0, 1.0, 1.0]}

    ],

    "material":{        （材料的弹性模量和泊松比）

        "modulus-of-elasticity":1e3,

        "poisson-ratio":0.3

    }

}
```

1.1.2   输出文件

{"elements":[

Node 为节点编号，number 为单元的编号，strain 为应变，stress 为应力，vonmises 为应变能强度理论下得到的等效应力值。

{"node":[1,2,3,6],"number":100,"strain":[-
0.0007568092860647269,0.0,0.0,0.001382484880916354,0.0,0.00109738758228l9085]
,"stress":[-1.0187817312409784,-0.4366207419604193,-
0.4366207419604193,0.5317249541985977,0.0,0.4220721470315032],"vonmises":1.31
20748416504198},{"node":[1,0,3,5],"number":101,"strain":[-
0.0001728746495571639,0.0,0.0,0.0011500736965735674,0.0,0.00134265471397891 22
],"stress":[-0.23271587440387442,-0.0997353747445176,-

0.0997353747445176,0.44233603714367975,0.0,0.5164056592226585],"vonmises":1.1
851969191611753},{"node":[1,3,5,6],"number":102,"strain":[-
0.00017287466495571639,0.00023241118434278655,-
0.00024526713169700375,0.0005661390600660044,-1.2855947354217193e-
05,0.0007587200774713492],"stress":[-0.2401327671082305,0.07162556666096215,-
0.2958192918311842,0.21774579233307859,-
0.004944595136237382,0.29181541441205733]},"vonmises":0.7179392254475273},{"no
de":[3,4,6,7],"number":103,"strain":[-0.0004022388721066498,-
0.00050876030967610036,0.0011684578409258495,0.0015366747766343805,0.000620710
6603416503,0.000685780033253369],"stress":[-0.1608806751917905,-
0.2428202425529088,1.0473475656024396,0.5910287602439924,0.23873486936217317,
0.26376155125129574],"vonmises":1.7300779582106856},{"node":[0,3,4,5],"number
":104,"strain":[0.0007620899968209236,0.000632260486221794,0.0,-
0.00041715143602631406,0.0006610036527238193,0.0006816510612550929],"stress":
[1.3906560454638166,1.2907871911567939,0.804432970986183,-
0.16044286001012079,0.25423217412454585,0.26217348509811267],"vonmises":0.878
8733652948059},{"node":[3,4,5,6],"number":105,"strain":[0.0001781553603133605
7,0.000632260486221794,-0.00024526713169700375,-
0.0001847402516835274,0.0008934148370666058,-
0.00014755070694947399],"stress":[0.46308992033997964,0.8124015556541592,0.13
738031110123794,-0.07105394295520284,0.34362109117946377,-
0.05675027190364384],"vonmises":0.8490663435674505},{"node":[5,6,7,10],"numbe
r":106,"strain":[0.0007368315660189253,-0.00050876030967610036,-
0.00024526713169700375,6.95727936864338e-05,-
0.0005203101355562474,0.00010911202644047976],"stress":[0.5568728150025297,-
0.4012747816859539,-0.19858772170203098,0.026758766802474535,-
0.20011928290624897,0.041966164015569137],"vonmises":0.9447219144610967},{"no
de":[5,4,7,9],"number":107,"strain":[0.0013630057140168004,0.0006322604862217
94,0.0011684578409258495,-0.0009463465992533139,0.0020344356329645035,-
0.00048556296449929826],"stress":[2.8736913422231787,2.3115796285347123,2.7240
391321532167,-0.36397946125127456,0.7824752434478859,-
0.18675498634345483],"vonmises":1.6103173729492377},{"node":[5,7,9,10],"numbe
r":108,"strain":[0.0013630057140168004,-7.323514915636635e-05,-
0.00023098637318355207,-0.0009921265148311784,-7.05042165230587e-05,-
0.00053134288007084471],"stress":[1.6593029675188165,0.5545023035394576,0.4331
5520813393016,-0.38158712108891474,-0.027117006355022573,-
0.20436264618109504],"vonmises":1.390574859104951},{"node":[7,8,10,11],"numbe
r":109,"strain":[0.0006124384920140227,-
0.0007910730714540653,0.0003111100927091757,0.000476278629469298,-
0.00012637816914039604,0.0007860217088708874],"stress":[0.5475347130506711,-
0.5320895665401658,0.31574363666232724,0.1831840882574223,-
0.04860698813092155,0.3023160418734182],"vonmises":1.1623297855142354},{"node
":[4,7,8,9],"number":110,"strain":[0.0001336708565700109,-
5.4129986151544955e-
05,0.0011684578409258495,0.0009693787305668144,0.0003842708468279718,0.001164
6018216435496],"stress":[0.8228229923678825,0.6783608056589934,1.618812980333
9122,0.37283797329492857,0.14779647954921993,0.44792377755521134],"vonmises":
1.3615883486510383},{"node":[7,8,9,10],"number":111,"strain":[8.7890940992146
2e-05,-5.4129986151544955e-05,-
0.00023098637318355207,0.00026388309518865396,-0.00032122478855018907,-
0.00028062230804371674],"stress":[-0.046175479050051454,-0.15542234608366,-
0.29146572072366544,0.10149349814948229,-0.12354799559622655,-
0.10793165693989104],"vonmises":0.3961664171241216}],

"nodes":[

每一个节点的位置、序号、以及位移。

{"displace":[0.0,0.0,0.0],"nodenum":0,"position":[0.0,0.0,0.0]},{"displace":[0.0,0.0,0.0],"nodenum":1,"position":[0.0,3.0,0.0]},{"displace":[0.0,0.0,0.0],"nodenum":2,"position":[0.0,3.0,3.0]},{"displace":[0.0,0.0,0.0],"nodenum":3,"position":[0.0,0.0,3.0]},{"displace":[0.0022862699990462771,0.0015534396310553204,0.0020449531837652790],"nodenum":4,"position":[3.0,0.0,0.0]},{"displace":[-0.0005186239486714917,0.0034502210897207025,0.004027964141936737],"nodenum":5,"position":[3.0,3.0,0.0]},{"displace":[-0.002270427858194181,0.004147454642749062,0.0032921627468457254],"nodenum":6,"position":[3.0,3.0,3.0]},{"displace":[-0.0012067166163199493,0.005673735571777373,0.005550326706542827],"nodenum":7,"position":[3.0,0.0,3.0]},{"displace":[0.0026872825601728036,0.0035784651895496584,0.009031745255478648],"nodenum":8,"position":[6.0,0.0,0.0]},{"displace":[0.00357039319337891,0.0034160752310950235,0.006064261855240511],"nodenum":9,"position":[6.0,3.0,0.0]},{"displace":[-5.993316013740481e-05,0.005419884265682595,0.005371302735689854],"nodenum":10,"position":[6.0,3.0,3.0]},{"displace":[0.0006305988597221188,0.007793103480044791,0.009965075533606175],"nodenum":11,"position":[6.0,0.0,3.0]}]}

2 前处理环节

2.1 建立三维模型
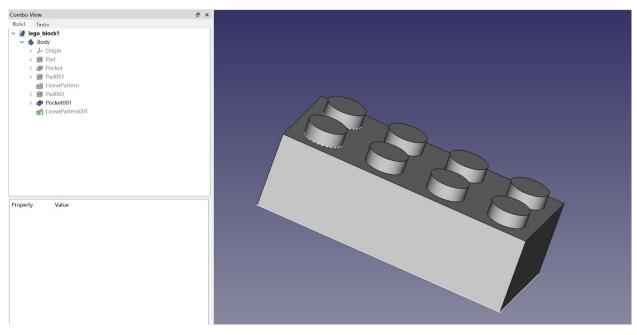
通过使用 FreeCAD 建模软件，构建一个积木模型，如图 1 所示，并导出.step 文件。

图 1 三维模型

2.2 网格划分

使用 Hypermesh 前处理软件对该模型进行处理。首先导入.step 文件模型，然后将该模型使用 tetra4 四面体网格进行自动划分。随后在模型的一侧添加位移约束，在另外一侧的 140 个节点，每一个节点添加 1N 的拉力。网格划分以及添加约束的结果如图 2 所示。
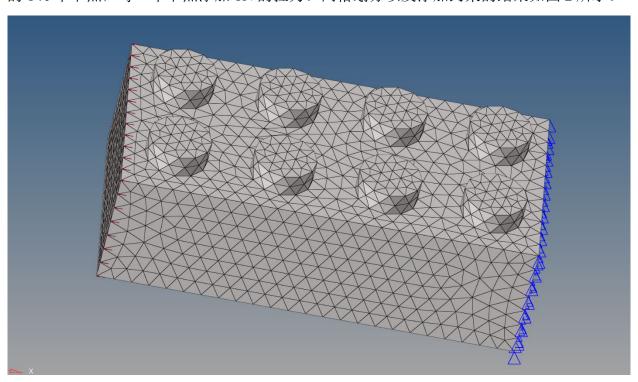

图 2 网格划分结果

将 Hypermesh 划分出的文件导出为".hmascii"格式并保存，准备进行进一步处理。

2.3 输入文件格式处理

使用自制的文件格式转换程序 parsehmascii（该文件转换仅支持四节点四面体网格），将 hypermesh 导出的文件转换成 json 文件，须输入命令：

```
./parsehmascii -i block1.hmascii -o block1-in.json
```

在输出的文件 block1-in.json 文件中，添加指定材料的参数：

```
"material":{"modulus-of-elasticity":2.1e5,"poisson-ratio":0.3},
```

2.4 有限元计算

在得到输入的 json 文件之后，使用 tetra4 程序，需要输入命令：

```
./tetra4 -i block1-in.json -o block1-out.json
```

图 3 程序运行结果

计算包含 3446 个节点，12585 个单元的模型，耗时 570ms。

在 block1-out.json 文件中，包含了所有节点的位移量，以及每一个单元的应力、应变、以及按照应变能理论计算得到的等效应力。

图 4 输出结果文件