

3D Truss Problem

huangyf15

1 Problem 2.4

Problem 2.4 Given the three-bar structure subjected to the prescribed load at point C equal to 10^3 N as shown in Figure 2.19. The Young's modulus is $E = 10^{11}$ Pa, the cross-sectional area of the bar BC is $2 \times 10^{-2} \text{ m}^2$ and that of BD and BF is 10^{-2} m^2 . Note that point D is free to move in the x-direction. Coordinates of joints are given in meters.

- Construct the global stiffness matrix and load matrix.
- Partition the matrices and solve for the unknown displacements at point B and displacement in the x-direction at point D.
- Find the stresses in the three bars.
- Find the reactions at nodes C, D and F.

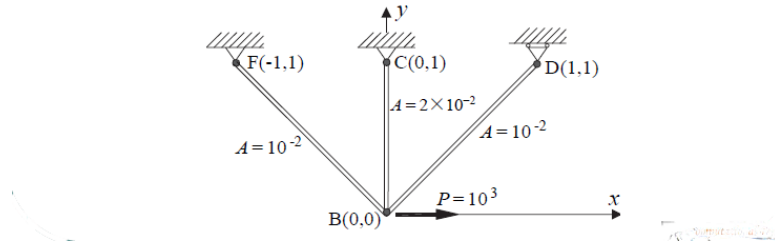


Figure 1: Problem 2.4

1.1 Preprocessing & Discussions

为实现前处理，我们需要输入单元物理参数和系统离散参数。这里我们指出如下两点：

- 由于涉及物理量的数量级大小不一，为避免由此带来较大的舍入误差，我们对所有涉及物理量均进行了正则化处理，数量级保存在相应的全局变量中；这与科学记数法的思路类似。更重要的是，这也为后续的量级估计以及可视化提供了方便。

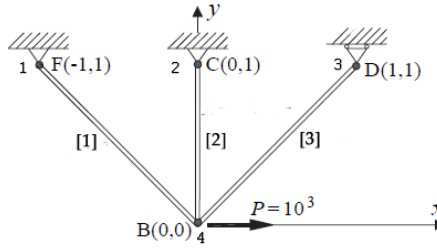


Figure 2: Numbers of the nodes and elements

- 在确定系统离散参数时，我们注意到点 D 在 x 方向自由、在 y 方向受位移约束，从而常规步骤无法使得受位移约束的自由度编号全部紧邻，于是就给后面 Reduction Technique 求解器的应用带来了小小的困难。为此我们提出如下几种备选方案：
 - 针对本题的具体情形，可以不妨将原题中坐标轴（如图 1）顺时针旋转 90 度，并采取合适的节点编号顺序使问题化归为常规情况。但这种方法不具有普适性，可解决的问题种类受到较大限制。
 - 沿用原题中已建立的坐标系并采用通常的节点编号顺序（如图 2），在求解过程中变换全局自由度号，以便应用 Reduction Technique 求解（活动列法亦要求矩阵满秩）。这时需要设置标号推前/拉回函数，标号变换可在组装总刚的同时或之后进行。
 - 沿用原题中已建立的坐标系并采用通常的节点编号顺序（如图 2），在求解过程中不进行自由度号变换，但用 Penalty Method 代替 Reduction Technique。由于这时不涉及矩阵的分块，因此无需变换全局自由度号。

如果再考虑到条件数过大时可能需要采用的预处理技术，我们便可得到如下的求解方案：

- 变换自由度号，采用分块降阶法 (+ 预处理技术) 求解刚度方程。
- 不变换自由度号，采用罚函数法 (+ 预处理技术) 求解刚度方程。

考虑到第 2 题的要求，为了方便起见，我们这里将采取方案 2 即罚函数法并在第 2 题中将其延拓至三维情形。有关罚函数法的说明见第 2 题；算法描述详见附录 A，正文中不再赘述。本题数据输入文件如下：

~\PREPRC\input_file_problem2_4.m

1.2 Assembly

利用局部标号转移阵容易组装得到如下的全局刚度阵：

$$\mathbf{K} = \begin{pmatrix} 0.3536 & -0.3536 & 0 & 0 & 0 & 0 & -0.3536 & 0.3536 \\ -0.3536 & 0.3536 & 0 & 0 & 0 & 0 & 0.3536 & -0.3536 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0.3536 & 0.3536 & -0.3536 & -0.3536 \\ 0 & 0 & 0 & 0 & 0.3536 & 0.3536 & -0.3536 & -0.3536 \\ -0.3536 & 0.3536 & 0 & 0 & -0.3536 & -0.3536 & 0.7071 & 0 \\ 0.3536 & -0.3536 & 0 & -2 & -0.3536 & -0.3536 & 0 & 2.7071 \end{pmatrix} \times 10^9$$

载荷矩阵只有第 7 个自由度对应的元素非零，其值为 1×10^3 。

1.3 Solution & Postprocessing

利用罚函数法，容易解出系统各自由度的位移：

$$\mathbf{d} = \begin{pmatrix} 0.0000 & 0.0000 & 0.0000 & 0.0000 & 3.8288 & 0.0000 & 3.3287 & 0.5001 \end{pmatrix}^T \times 10^{-6}$$

再利用应变矩阵以及受约束自由度的平衡方程，即可以获得各单元应力值以及受约束自由度方向的约束反力：

$$\sigma = \begin{pmatrix} 1.4142 & -0.5000 & 0.0000 \end{pmatrix}^T \times 10^5$$

$$\mathbf{r} = \begin{pmatrix} -1.0000 & 1.0000 & 0.0000 & -1.0000 & 0.0000 \end{pmatrix}^T \times 10^3$$

变形前后的桁架结构如图 3 所示，其节点位移放大倍数为 10^4 。

上述计算结果连同其作图结果亦可参考如下文件：

~\POSTPRC\solution_prb2_4.m
~\POSTPRC\plotdisp_prb2_4.fig

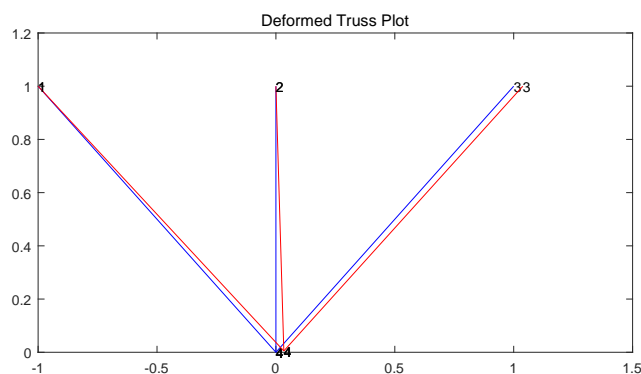
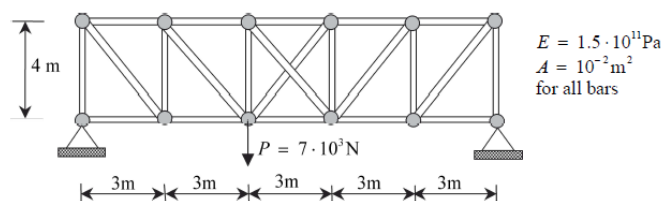


Figure 3: Deformed Truss (Problem 2.4)

2 Problem 2.7

Problem 2.7 Modify the MATLAB finite element code to enforce displacement boundary conditions using *the penalty method*.

- Solve for the nodal displacements and stresses of the structure shown in the Figure.
- *Plot the deformed structure* with MATLAB. For this purpose, add the $\text{mag} \times$ displacement to the nodal coordinates. The factor mag is to magnify the displacements so that they are visible



Problem Modify the MATLAB finite element code to solve *3D truss* problems. Can we use this code to solve the 2D truss problem shown in above figure ?

Figure 4: Problem 2.7

2.1 Preprocessing & Method

单元物理参数设置与第 1 题类似，节点编号方案如图 5 所示。本题数据输入文件如下：

```
~\PREPRC\input_file_problem2_7.m
```

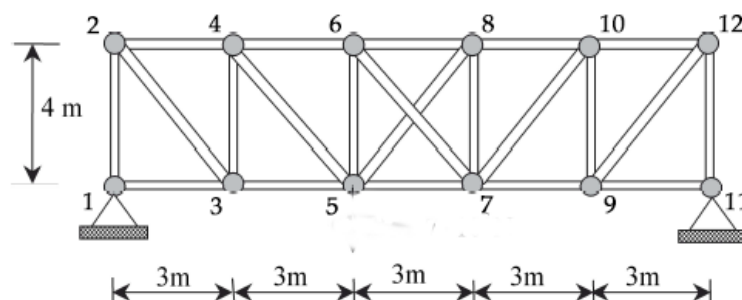


Figure 5: Numbers of nodes (Problem 2.7)

这里采用的罚函数法，具体来讲就是将受约束自由度对应项的刚度赋一个足够大的值，这样在使总刚变为非奇异阵的同时自然满足了本质约束条件。值得注意的是，若在给对角元赋值的同时将同行同列的元素清零，则此时的罚函数法与分块降阶法除自由度顺序的差别外并无二致。

2.2 Solution & Postprocessing

考虑到计算结果的存储矩阵规模较大，我们仅指出其存储路径而不再此处一一列举它们的值：

```
~\POSTPRC\solution_prb2_7.m
```

变形前后的桁架结构如图 6 所示，其节点位移放大倍数为 $10^{3.5}$ 。相应文件的存储路径如下：

```
~\POSTPRC\plotdisp_prb2_7_2D.fig
```

2.3 Extension to the 3D case & Discussions

将原程序改写为适用于 3D 情况是容易的；事实上，只要在相应程序中各添加一个 3D 分支即可。由于我们采用的求解方案 2（即罚函数法）对于受约束自由度是否紧邻没有特别要求，故我们修改后的程序能够解决 3D 问题。这时得到的计算结果可参考如下文件：

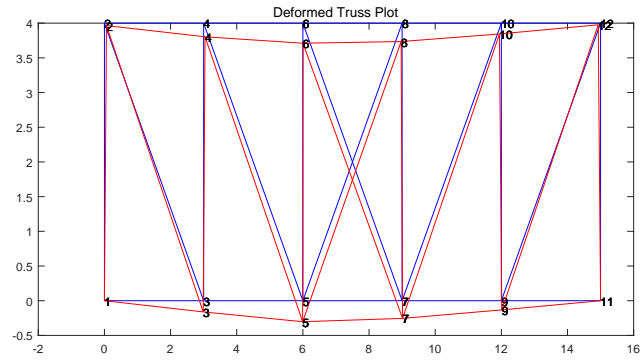


Figure 6: Deformed Truss (Problem 2.7)
(2D)

~\POSTPRC\solution_prb2_7.m

变形前后的桁架结构如图 7~8 所示，其节点位移放大倍数为 $10^{3.5}$ 。相应文件的存储路径如下：

~\POSTPRC\plotdisp_prb2_7_3D.fig

~\POSTPRC\plotdisp_prb2_7_3D_1.fig

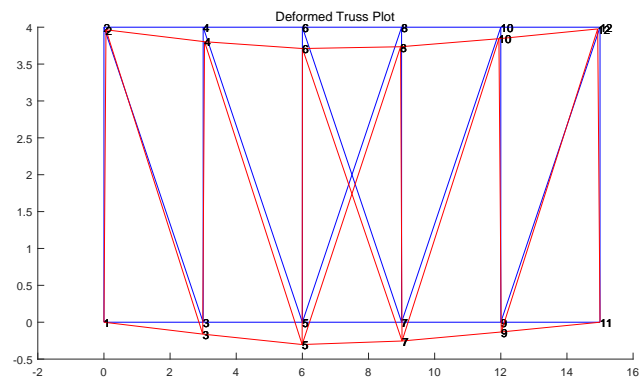


Figure 7: Deformed Truss (Problem 2.7)
(3D, from z-axis)

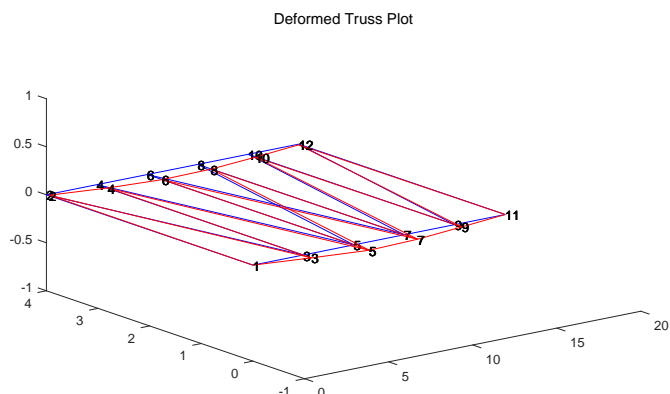


Figure 8: Deformed Truss (Problem 2.7)
(3D)

结合以上结果，我们指出如下几点经验教训：

1. 有限元本质上应用的是平衡方程（虚功原理），这要求结构静定或超静定。由于背景空间维数的增加，为了使结构保持静定，必须增加约束的个数，即增加 z 方向的约束。实际上，这等价于要求载荷给定的自由度对应的刚度阵满秩。
2. 在计算单元方向时，需要注意以下两点：
 - (a) 应用反三角函数计算角度值时应务必小心，这是因为原则上单元的方向可以是 $0 \sim 2\pi$ 内的任意值，使用不当则会造成错误。
 - (b) 当反三角函数与三角函数进行复合运算时会产生一定的舍入误差，例如 $\cos(\arccos(0))$ 的数值计算结果一般不会等于 0，因此直接利用单元的方向向量而非其角度值更为合理。
3. 刚度阵的条件数对计算结果可靠性的影响是重要的；若刚度阵的条件数过大而不采取适当的矩阵预处理手法，将无法得到有意义的结果。

A Simplified Documentation for the Programme

A.1 Main

该部分程序位于 `\truss` 文件夹中。

`truss.m` 这是求解桁架结构的主程序，用于计算给定载荷及约束下静定/超静定桁架的变形及应力，同时根据需要作出变形前后桁架构形的图形。由前处理(`\truss\PREPRC\preprocessor.m`)、组装总刚(`\truss\STFMAT`)、求解刚度方程(`\truss\SOLVER`)和后处理四部分(`\truss\POSTPRC\postprocessor.m`) 组成。

A.2 Include flags

该部分程序位于 `\truss` 文件夹中。

`include_flags.m` 该程序定义了程序所需的所有全局变量，供主程序和子程序载入。具体包括系统离散参数、单元物理参数及其量级参数、求解器相关参数、作图命令标签等。

A.3 Preprocessing

该部分程序位于 `\truss\PREPRC` 文件夹中。

`preprocessor.m` 该程序是前处理模块的主子程序。该模块由载入求解域参数、计算单元长度与方向、估计导出物理量量级、生成局部标号转移阵四部分组成。

Rem. 值得指出的是，这里计算单元的方向实际上得到的是单元的方向向量，这样会避免后续的舍入误差影响。

`~\input_file` 该子文件夹中的文件用于载入求解域参数，并根据需要进行初始桁架构型作图。文件格式为 `input_file_[type][number].m`。具体包括输入系统基本参数（系统离散参数、单元物理参数及其量级参数、求解器相关参数、作图命令标签），局部变量初始化（全局刚度阵、节点自由度位移、系统载荷），输入定解条件（给定载荷、位移约束），初始构型作图等。其中，初始构型作图由下面的子程序实现。

`~\truss_mesh` 该子文件夹中的文件用于载入系统的节点、单元编号，并根据需要为初始构型作图。文件格式为 `truss_mesh_[type][number].m`。

`plottruss.m` 该程序用于初始桁架构型作图。

A.4 Assembly

该部分程序位于 \truss\STFMAT 文件夹中。

trusselem.m 该程序用于计算单元刚度阵与应变矩阵，并输出单元刚度阵的值。这里应变矩阵以全局变量的形式存储，以便后处理计算应力时使用。

assembly.m 该程序用于组装全局刚度阵。输入变量为组装前总刚 K 、单元编号 e 及单元刚度阵 ke ，输出变量为组装后总刚 K 。

A.5 Solution

该部分程序位于 \truss\SOLVER 文件夹中。其中的每一类求解器都会在工作台上输出节点位移、受约束自由度方向的约束反力以及相应刚度阵 (分块降阶法为有效子刚度阵，罚函数法为惩罚后刚度阵) 的条件数。

solverdr.m 求解器标签 `solver = 1`。该程序为简单分块降阶求解器。与罚函数求解器不同，这里要求受约束自由度集中在位移矢量的前部。

solverdr_pnl.m 求解器标签为 `solver = 2`。该部分为罚函数求解器。设置了惩罚因子初值 `pnl0`、惩罚因子步长 `pnl`、迭代收敛精度 `err` 三个参数来实现其功能，同时还设置了临界循环参数以避免迭代发散导致的死循环。

~\UNDONE 内有活动列求解器(`solver_colcol.m`)，推前/拉回函数(`transfwd.m`、`transbkwd.m`)，skyline求解函数(`makeskyline.m`)。其中活动列求解器的求解器标签 `solver = 3`。

A.6 Postprocessing

该部分程序位于 \truss\POSTPRC 文件夹中。

postprocessor.m 该程序是后处理模块的主子程序。输入变量为节点位移 d ，程序将在工作台上输出单元应力。该模块由计算单元应力、变形后桁架构型作图两部分组成。

plotdisp.m 该程序用于变形后桁架构形作图。输入变量为节点位移 d 。

~\plotdisp_output 该子文件夹用于保存计算结果。文件格式为 `plotdisp_[type][number].m`。

~\solution_output 该子文件夹用于保存作图结果。文件格式为 `solution_[type][number].m`。