有限元基础编程（终结篇）——C3D8单元程序编制

Finite Element Basic Programming (Final Part) - C3D8 Element Programming

易公子 Yi Gongzi 关注 Focus

2022年7月24日 10:48 July 24, 2022 10:48

有限元基础编程（终结篇）——C3D8单元程序编制

Finite Element Basic Programming (Final Part) - C3D8 Element Programming

本篇推文是有限元基础编程的终结篇，讲述C3D8单元的程序编制及实现。主要内容有：**C3D8单元理论基础、便于编程的"乘大数法"处理边界条件、编制程序注意事项、云图绘制函数、INP文件读取函数、Abaqus仿真对比**等，内容量大，慢慢食用~

This tweet is the final part of the Finite Element Basic Programming series, discussing the programming and implementation of the C3D8 element. Main contents include: C3D8 element theoretical foundation, "Multiplying Large Numbers" method for programming convenience to handle boundary conditions, programming 注意事项, cloud chart drawing function, INP file reading function, Abaqus simulation comparison, etc. The content is substantial, please enjoy it slowly~

特别声明：程序框架采用了**吉林大学左文杰老师**的脚本文件，计算单元刚度的核心计算程序仍延续我们以往编制程序的风格。代码文件获取方式详见文末。

Special Statement: The program framework adopts the script file of Professor Zuo Wenjie from Jilin University, and the core calculation program for element stiffness continues to follow the style of our previous program 编制. The method to obtain the code file is detailed at the end of the document.

理论基础 Theoretical Foundation

与Q4单元理论基础相同，唯一的区别就是：**每个节点的自由度由2变成了3**，代码具体变化看 `Ke` 函数和 `C3D8_cal_B` 函数的变化，理论部分可参考[有限元基础编程——Q4单元]。

The theoretical foundation is the same as that of the Q4 element, with the only difference being that the degree of freedom of each node has changed from 2 to 3. The specific changes in the code can be seen in the `Ke` function and `C3D8_cal_B` function. For the theoretical part, please refer to Finite Element Basic Programming - Q4 Element.

边界条件处理（乘大数法） Boundary Condition Handling (Multiplying by a Large Number Method)

理论： Theory:

对于边界条件 $\bar{q}r = \bar{u}$ *的情形，可将刚度方程进行如下操作：*

For the case of boundary conditions, the stiffness equation can be manipulated as follows:

$$\begin{array}{cc} & \begin{matrix} 1 & 2 & \cdots & \cdots & r & \cdots & \cdots & \cdots & \cdots \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ \vdots \\ \vdots \\ r \\ \vdots \\ \vdots \\ \vdots \\ \vdots \end{matrix} & \begin{bmatrix} & & & & & & & & \\ & & \cdots & & & & \cdots & & \\ & & & & & & & & \\ & & & & & & & & \\ k_{r1} & k_{r2} & \cdots & \cdots & \alpha k_{rr} & \cdots & \cdots & \cdots & \cdots \\ & & & & & & & & \\ & & & & & & & & \\ & & \cdots & & & & \cdots & & \\ & & & & & & & & \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ \vdots \\ \bar{q}_r \\ \vdots \\ \vdots \\ \vdots \\ \vdots \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ \vdots \\ \alpha k_{rr} \bar{u} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \end{bmatrix} \end{array}$$

考察其第 r *行的等价性：* *Examine the equivalence of the equation in the nth row:*

$$kr1q_1 + k_{r2}q_2 + \cdots + \alpha k_{rr}\bar{q}r + \cdots + krnq_n = \alpha k_{rr}\bar{u}$$

由于 $\alpha k_{rr} \gg k_n$ （ α 是一个很大的数），则上式变为 Since (is a very large number), the above equation becomes

$$\alpha k_{rr}\bar{q}r \approx \alpha krr\bar{u}$$

即 $\bar{q}_r \approx \bar{u}$

优点： Advantages:

1.

    1. 既可以处理 $\bar{q}_r = 0$ 的情形，又可以处理 $\bar{q}_r = \bar{u}$ 的情形； 1. Can handle both and situations;

2.

2. 原整体刚度矩阵求解规模不变，不需要重新排序；

2. The scale of the overall stiffness matrix solution remains unchanged, no need to reorder;

3.

3. 保持整体刚度矩阵的对称性。 3. Maintains the symmetry of the overall stiffness matrix.

相关代码： Relevant Code:

```
% 乘大数法施加位移约束
BigNumber=1e8;
ConstraintsNumber=size(Constraints,1);
FixedDof=Dof*(Constraints(:,1)-1)+Constraints(:,2);  %被约束的自由度编号(列向量)
for i=1:ConstraintsNumber
    K(FixedDof(i),FixedDof(i))=K(FixedDof(i),FixedDof(i))*BigNumber;
    Force(FixedDof(i))=Constraints(i,3)*K(FixedDof(i),FixedDof(i));
end
```

程序编制 Program Development

模型描述：一悬臂梁，左端固定，三个方向自由度均受约束，右端下侧受到均布荷载作用。

Model Description: A cantilever beam, fixed at the left end, with all three degrees of freedom constrained, and subjected to uniformly distributed loads at the lower side of the right end.

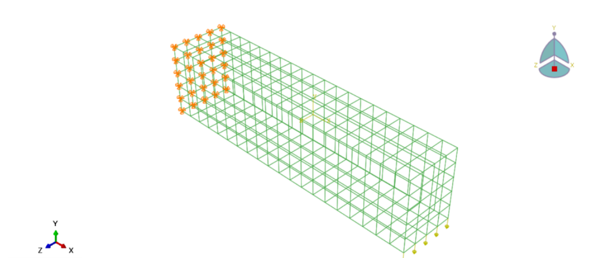

图2 悬臂梁示意图 Figure 2 Schematic of a cantilever beam

主程序代码 Main Program Code

```
%*****************************************************
%       公众号：易木木响叮当
%*****************************************************
function Main()
%读取inp文件获得节点坐标信息Nodes及单元信息Elements
[Nodes, Elements] = Readmesh( 'BC_Force.inp' );
Forces=[601 2 -100;607 2 -100;613 2 -100;619 2 -100;625 2 -100];
ConNumber=1:30;
```

```
Constraints=zeros(size(ConNumber,2)*3,3);
for i=1:size(ConNumber,2)
Constraints(3*i-2:3*i,:)=[ConNumber(i) 1 0;ConNumber(i) 2 0;ConNumber(i) 3 0;];
end
E=210000; %弹性模量
u=0.3;    %泊松比
U =StaticsSolver(E,u,Forces,Constraints,Nodes,Elements);
% 输出结果
OutputTXT = fopen('Results.txt','w'); %打开一个可写文件，用于写入计算结果
OutputResults(OutputTXT,Nodes,Elements,U)%调用输出结果文件
fclose(OutputTXT);
edit('Results.txt')
end
```

以上程序采用了**吉林大学左文杰老师**编制的脚本文件，需要注意以下几个方面：

The above program uses the script file compiled by Professor Zuo Wenjie from Jilin University. The following aspects should be noted:

1.

1. 程序中使用 `Readmesh` 函数（仅需调用即可）读取INP文件中节点、单元信息；

1. The program uses the `Readmesh` function (only needs to be called) to read node and element information from the INP file;

2.

2. 节点外载荷和边界条件信息需要在INP文件中寻找，较为麻烦，方便起见可采用ANSYS建模，导出节点、单元、载荷、边界条件信息，相应的APDL命令流如下：

2. The information of node external loads and boundary conditions needs to be searched for in the INP file, which can be 麻烦 (troublesome). For convenience, ANSYS modeling can be used to export node, element, load, and boundary condition information. The corresponding APDL command stream is as follows:

```
 nwrite,node,txt,
ewrite,element,txt,
Dlist,ALL
Flist,ALL
```

1.

1. 云图绘制采用 `PlotContour` 函数，本程序仅仅绘制出网格图、位移云图，因为研究对象是线弹性体，**位移**是第一求解未知量，最为准确，应力是通过应力矩阵变换而得，线弹性状态很简单，不是求解核心，所以本

程序仅展示位移云图，有关应力云图显示详情请观看一阶六面体(C3D8)的线弹性有限元编程[1]

1. The cloud diagram is drawn using the `PlotContour` function, this program only draws the mesh diagram and displacement cloud diagram, because the research object is a linear elastic body, displacement is the first unknown quantity to be solved, which is the most accurate. Stress is obtained through the transformation of the stress matrix. The linear elastic state is very simple and not the core of the solution, so this program only displays the displacement cloud diagram. For details on the stress cloud diagram display, please refer to the linear elastic finite element programming of the first-order hexahedron (C3D8) [1] .

Abaqus对比 Abaqus Comparison

云图对比 Cloud Chart Comparison

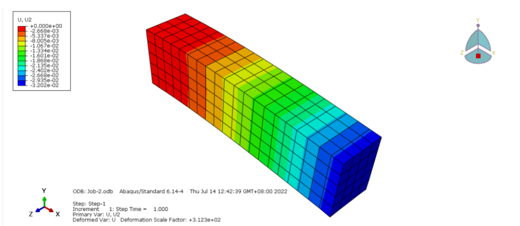

图3 Abaqus-U2 位移云图 Figure 3 Abaqus-U2 Displacement Cloud Chart

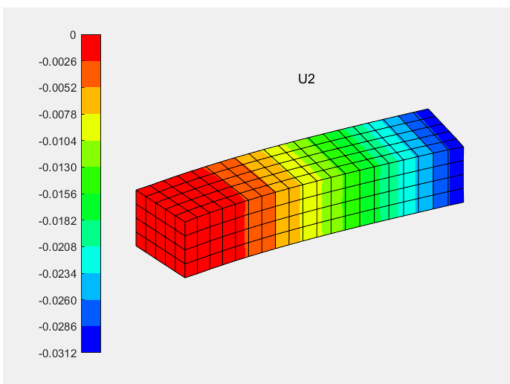

图4 Matlab-U2 位移云图 **Figure 4 Displacement Cloud Map of Matlab-U2**

Matlab最大U2值为-0.0312，Abaqus最大U2值为-0.0320，结果相差2.5%

The maximum U2 value in Matlab is -0.0312, and the maximum U2 value in Abaqus is -0.0320, with a difference of 2.5%

代码文件：在公众号：**易木木响叮当**，后台回复**C3D8**，即可自动获取~

Code file: Reply "C3D8" in the back end of the official account: Yimu Mu Xiang Ding Dang to automatically obtain it~

引用链接 Reference link

[1] 一 阶 六 面 体 (C3D8) 的 线 弹 性 有 限 元 编 程 : *https://www.bilibili.com/video/BV1KU4y1U7Mz?spm_id_from=333.999.0.0&vd_source=21e509b445f316000076bb38c1f19a05*

First-order hexahedral (C3D8) linear elastic finite element programming: https://www.bilibili.com/video/BV1KU4y1U7Mz?spm_id_from=333.999.0.0&vd_source=21e509b445f316000076bb38c1f19a05