

计算动力学

课程报告

陈楷东、姚依晨、李萌、陈家琦

2019年6月

目 录

[**第1章 引言** 1](#_Toc11407111)

[1.1 子空间迭代法相关文献调研 1](#_Toc11407112)

[1.1.1 经典的子空间迭代法 2](#_Toc11407113)

[1.1.2 引入移轴的子空间迭代法 3](#_Toc11407114)

[1.1.3 新型变带宽存储方案 4](#_Toc11407115)

[1.1.4 高效LDLT分解算法 6](#_Toc11407116)

[1.1.5 旋转向量（Turning Vector） 7](#_Toc11407117)

[1.1.6 Out-of-core算法 8](#_Toc11407118)

[1.2 STAP90程序总体结构 8](#_Toc11407119)

[1.2.1 前处理 8](#_Toc11407120)

[1.2.2 Stap90程序改写 10](#_Toc11407121)

[**第2章 程序改写模块：扩展STAP90程序** 10](#_Toc11407122)

[2.1 高阶三维实体单元 10](#_Toc11407123)

[2.1.1 单元基本理论 11](#_Toc11407124)

[2.1.2 单元程序实现 14](#_Toc11407125)

[2.1.3 单元验证算例 16](#_Toc11407126)

[2.2 特征值求解高阶三维实体单元 24](#_Toc11407127)

[2.2.1 旋转向量改进的子空间迭代法 24](#_Toc11407128)

[2.2.2 程序伪代码 26](#_Toc11407129)

[2.2.3 程序实现 27](#_Toc11407130)

[2.2.4 验证算例 28](#_Toc11407131)

[2.3 后处理读取 31](#_Toc11407132)

[2.3.1 静力数据读取 31](#_Toc11407133)

[2.3.2 振型数据读取 31](#_Toc11407134)

[2.3.3 后处理操作 32](#_Toc11407135)

[**第3章 实际问题分析模块** 34](#_Toc11407136)

[3.1 桁架模型分析 34](#_Toc11407137)

[3.1.1 理想桁架的基本假设 34](#_Toc11407138)

[3.1.2 实体单元桁架模型计算 34](#_Toc11407139)

[3.2 自选结构分析 36](#_Toc11407140)

[3.2.1 自选结构介绍 36](#_Toc11407141)

[3.2.2 自选结构计算结果 37](#_Toc11407142)

[**第4章 总结与体会** 39](#_Toc11407143)

[4.1 总结与分工 39](#_Toc11407144)

[4.2 收获与体会 39](#_Toc11407145)

[**参考文献** 41](#_Toc11407146)

# 引言

## 子空间迭代法相关文献调研

在结构动力学领域，求解系统的某阶（或某几阶）特征频率和模态，是一个重要的基础问题。针对这一问题，一般的求解方法有如下两种：

(I) 子空间迭代法[1]（The Subspace Iteration Method）是由K.J.Bathe教授提出的，最初被应用于楼房、桥梁等土木结构的动力学特性分析[1,2]。

(II) Lanczos方法[3]（Lanczos Method）则是由Cornelius Lanczos教授提出的，提出这一方法时，Lanczos教授正在美国国家标准局工作，他的文章针对一般的特征值问题，从应用数学的角度提出求解方法。

一般而言，Lanczos方法的精度比子空间迭代法要高出一个数量级[4,5]，这也意味着，相较于Lanczos方法，子空间迭代法有更大提升计算效率的空间，近年来围绕子空间迭代法的改进研究也相对更多。虽然一些改进后的子空间迭代法，仍无法达到Lanczos方法的求解效率，但这些改进并非毫无意义。近年来，并行计算成为有限元方法的一种重要的发展方向，而Lanczos 方法的特性，导致了该方法需要逐个对向量进行求解计算，只能在求解每个向量时，对刚度阵进行LDLT分解、高斯消去和回代等过程进行并行处理，而无法对各个向量的计算做并行处理。相对而言，子空间迭代法则可以对多个向量做并行运算，算法的并行程度更高，可以大幅度提升计算效率[6]。

通过上面的讨论，我们不难看出，子空间迭代法相较于Lanczos方法同样是有着自身独特优势的，对子空间迭代法的改进本身就有着重要意义。

近年来，对子空间迭代法计算效率的改进，主要包括以下几个方面：

1. 引入移轴法，提高迭代的收敛速率。
2. 采用新型变带宽存储方案，降低内存占用。
3. 采用更加高效的LDLT分解算法，提高求解效率。
4. 引入Turning Vector 概念，提升单次迭代效率
5. 利用Out-of-core算法替代In-core算法，处理更大规模的特征值问题。

下面逐一对这些改进做介绍：

### 经典的子空间迭代法

为了方便对子空间迭代法的不同改进方案进行介绍，我们首先给出子空间迭代法的经典求解过程。首先，考虑如下的广义特征值问题[6,7]：



其中，、为一个有N个自由度的有限单元系统的刚度阵和质量阵（易知、均为N阶方阵）。我们的求解目标是，寻找（最小的）前个特征值和相应的特征向量满足如下条件：







其中，为克罗内克积（Kronecker delta）。一般而言，上述特征值由小到大排列，即：



针对上述广义特征值问题，经典的子空间迭代法包括以下步骤[6,7]：

Step1: 利用同时迭代法更新子空间:



一般而言，子空间维度大于待求特征向量数 ，但远小于系统总自由度数N。利用同时迭代法更新子空间的公式为：



其中，、均为阶矩阵。

Step2：将系统的刚度阵和质量阵、在子空间上投影，计算：





Step3：求解减缩的广义特征值问题：



其中，、均为阶方阵。

Step4：更新子空间，作为对特征向量的近似;



相应的特征值矩阵为：



在实际应用过程中，空间迭代法的算法如下：

表 1‑1经典子空间迭代法算法

|  |  |
| --- | --- |
| 说明 | 计算 |
| 初始化 |  |
| 确定子空间维数 | [6] |
| 分解刚度矩阵 |  |
| 开始迭代 |  |
| 矩阵乘法 |  |
| 求解 |  |
| 计算、在上的投影 |  |
| 求解广义特征值问题 |  |
| 形成新特征向量 |  |
| 判断收敛 | [6] |
| 结束迭代 |  |
| 检查strum序列 |  |

其中，在判断收敛时，为矩阵的第个列向量，与特征值相对应。

### 引入移轴的子空间迭代法

下面我们给公式引入一个移轴量，则广义特征值问题变为[8]：



则于1.1.1小节给出的子空间迭代公式应相应地更改为：



而其他公式则不需要改变。

由此可得，带移轴的子空间迭代法的算法为：

表 1‑2带移轴的子空间迭代法算法

|  |  |
| --- | --- |
| 说明 | 计算 |
| 初始化 |  |
| 确定子空间维数 | [6] |
| 分解刚度矩阵 |  |
| 开始迭代 |  |
| 矩阵乘法 |  |
| 求解 |  |
| 计算、在上的投影 |  |
| 求解广义特征值问题 |  |
| 形成新特征向量 |  |
| 判断收敛 | [6] |
| 结束迭代 |  |
| 检查strum序列 |  |

不难看出，带移轴的子空间迭代法本质上只是在进行同时迭代时将刚度阵替换为。这个替换十分简单，但是却可以在很大程度上提高迭代的收敛效率。对于某阶特征值，带移轴的子空间迭代法的收敛率为[8]：



此外数值计算结果同样表明，带移轴的子空间迭代法相较于不带移轴的子空间迭代法可有效节约求解时长[9]。

### 新型变带宽存储方案

STAP90中给出了一种经典的一维变带宽存储方案[4,7]，在存储时仅依靠每一列的列高来确定存储矩阵中的哪些元素，这会导致在进行变带宽存储时，仍会引入一些零元素，例如对于如下矩阵：



STAP90的一维变带宽存储方案为：

表 1‑3 STAP 90的一维变带宽存储

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 列 | 1 | 2 | 3 |  |  | 4 |  |  |  | 5 |
| **MAXA**(*neq*+1) | 1 | 2 | 3 |  |  | 6 |  |  |  | 10 |
| **A**(*nwk*) | 11 | 22 | 33 | 0 | a | 44 | e | 0 | b | 55 |
| 列 |  |  |  | 6 |  |  |  |  |  |  |
| **MAXA**(*neq*+1) |  |  |  | 14 |  |  |  |  | 19 |  |
| **A**(*nwk*) | g | f | d | 66 | h | 0 | 0 | 0 | c |  |

其中，存储矩阵对角元和最后一个非零元素在一维存储向量中的位置。为矩阵阶数，则为矩阵各列列高之和加矩阵阶数。

不难看出，在上述矩阵存储方式中，很容易出现零元素占用存储空间的情形，下面的稀疏矩阵存储方案则可以很好地解决这一问题[10]：

表 1‑4 典型稀疏矩阵存储方案

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 方程 | 1 |  |  |  | 2 |  | 3 |
| **ICA**(*neq*) | 4 |  |  |  | 6 |  | 9 |
| **JCA**(*nzr*) | 1 | 3 | 4 | 6 | 2 | 5 | 3 |
| **A**(*nzr*) | 11 | a | b | c | 22 | d | 33 |
| **MASTER**(*neq*) | 1 |  |  |  | 1 |  | 2 |
| 方程 |  |  | 4 |  | 5 |  | 6 |
| **ICA**(*neq*) |  |  | 11 |  | 13 |  | 14 |
| **JCA**(*nzr*) | 4 | 5 | 4 | 5 | 5 | 6 | 6 |
| **A**(*nzr*) | e | f | 44 | g | 55 | h | 66 |
| **MASTER**(*neq*) |  |  | 0 |  | 1 |  | 1 |

其中， 存储矩阵每一列中非零元素的个数，则存储每个非零元素具体位于矩阵的哪一列，为矩阵中非零元素的个数。不难看出该方案不会存储矩阵中的零元素。

最后一行引入“主方程”和“从方程”的概念[11]，观察矩阵，不难发现矩阵的第三行和第四行所占用的空间一致，这一现象的出现在有限元问题中并非偶然，例如，对于同一有限元节点，可以有多个方向的自由度，此时总矩阵中即会出现某几行占用空间相同的情况。

针对上述现象，我们可以定义第三行方程为“主方程”（Master Equation），而将第四行方程定义为“从方程”（Slave Equation）。中，非零元素表示此行方程为“主方程”，并给出“主方程”与其从属方程总数，零元素则表示此行方程为“从方程”。

利用这一特点，我们可以将“主方程”与“从方程”合并为一个“超方程”，分块存储矩阵，例如如下的“细胞数组”方案：

表 1‑5“细胞数组”方案

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 超方程 | 1 |  |  | 2 |  | 3 |  | 4 |  | 5 |
| **SUPER**(*mneq*) | 1 |  |  | 2 |  | 4 |  | 5 |  | 6 |
| **IA**(*mneq*) | 3 |  |  | 5 |  | 7 |  | 9 |  | 10 |
| **JA**(*nz*) | 1 | 3 | 5 | 2 | 4 | 3 | 4 | 4 | 5 | 5 |
| **LA**(*mneq*) | 4 |  |  | 6 |  | 11 |  | 13 |  | 14 |
| **A**(*nzr*) | 11 |  | c | 22 | d |  |  | 55 | h | 66 |

其中，**SUPER**(*s*)存储第*s*个超方程中最后一行方程对应的矩阵行数，**JA**(*nz*)存储细胞元素所对应的列，**IA**(*s*)存储前*s*个超方程存储的细胞元素总数，**LA**(*s*)则用于存储前*s*个超方程存储的单个元素总数。依据文献[11]的分析，“细胞数组”方案的存储效率可比典型稀疏矩阵存储方案提高25%以上。

### 高效LDLT分解算法

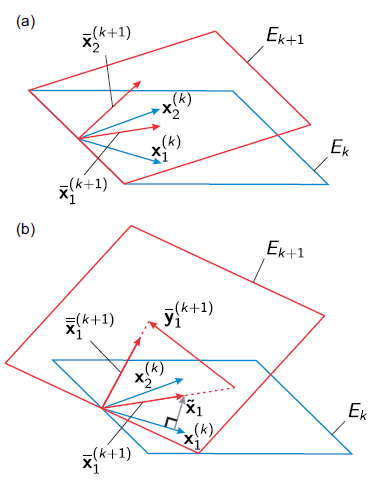
观察子空间迭代法的一般过程不难看出，在每一次更新子空间时，均需要对矩阵（刚度阵或移轴后矩阵）进行LDLT分解，因此，提高LDLT分解效率对于提高子空间迭代法效率起着至关重要的作用。

依照不同的矩阵存储方案，相应地可以提出不同的LDLT分解方案，从而提高子空间迭代法的求解效率，这一部分内容的详细介绍可在文献[11]中找到，这里不再赘述。

### 旋转向量（Turning Vector）

旋转向量（Turning Vector）的概念是由子空间迭代法的提出者，K.J.Bathe教授于2017年提出的[12]，也是本此大作业对子空间迭代法的核心改进算法，算法的具体实现过程将在2.2节做详细介绍，这里仅就算法本身特点介绍。

为简单起见，这里给出一个子空间中仅含有两个利兹基向量的情况，即：

图 1 子空间迭代法原理示意图



其对应的子空间设为，依据传统的子空间迭代法，利用同时迭代公式，可得下列新的子空间（见图 1(a)）：



下面引入旋转向量（Turning Vector）的概念，首先，对和进行正交化处理：



并进行归一化：



结合公式、，不难看出，可以表示为和的线性组合：



其中：



再对进行一次同时迭代，可求得旋转向量：



并进一步得到改进的子空间（见图 1(b)）:



记为对再进行一次同时迭代的结果（也是对连续进行两次同时迭代的结果），依据公式、，不难得到：



即旋转向量为和的线性组合，不难看出，将旋转向量引入到改进后子空间后，相当于进行了两次同时迭代，从而提高了求解效率。

### Out-of-core算法

上述各种子空间迭代法的改进思路，均属于仅利用计算机内存进行数值运算，这类算法被称为In-Core算法，其特点是求解速度快，但不足是求解问题的规模受计算机内存限制。

Out-of-Core算法则在进行数值运算时同时利用计算机硬盘存储数据[11]，这会牺牲一定的运算效率，但可以有效扩展程序可求解的有限元问题的规模。

## STAP90程序总体结构

本次大作业中所使用的STAP90 程序是基于FORTRAN 语言编写的有限元问题求解程序，具有模块化程度高、可修改性强、适应性好和计算效率高等多种优点。本次大作业的主要任务是对stap90程序进行拓展，增添一种高阶三维实体单元，同时增加特征值求解功能。由此需要对stap90的文件读入部分，单元计算部分和整体求解部分进行修改和开发。

### 前处理

本次大作业使用ansys进行前处理，前期在ansys经典界面建模和网格划分，之后通过apdl编写宏文件进行格式化输出，生成stap90程序的输入文件。由于添加了solid20单元，因此添加了有关新增单元的相关输入格式，修改后的输入文件格式如下：

1. 标题行

|  |  |  |
| --- | --- | --- |
| 列 | 变量 | 意义 |
| 1-80 | HED(80) | 标题，用于对所求解问题进行简单的描述 |

2. 控制行

|  |  |  |
| --- | --- | --- |
| 列 | 变量 | 意义 |
| 1-5 | NUMNP | 节点总数；如果为0则程序终止运行。 |
| 6-10 | NUMEG | 单元组总数，每个单元组只包含相同类型单元 |
| 11-15 | NLCASE | 荷载工况数。 |
| 16-20 | MODEX | 求解模式，等于0时只做数据检查，等于1时进行静力求解，等于2时进行模态计算。 |

3. 节点数据

|  |  |  |
| --- | --- | --- |
| 列 | 变量 | 意义 |
| 1-5 | N | 节点号 |
| 6-10 | ID(1,N) | x-平动方向边界条件代码。 0-自由；1-固定 |
| 11-15 | ID(2,N) | y-平动方向边界条件代码。 0-自由；1-固定 |
| 16-20 | ID(3,N) | z-平动方向边界条件代码。 0-自由；1-固定 |
| 21-30 | X(N) | x-坐标（字长10位，小数点后两位，F10.2） |
| 31-40 | Y(N) | y-坐标（字长10位，小数点后两位，F10.2） |
| 41-50 | Z(N) | z-坐标（字长10位，小数点后两位，F10.2） |
| 51-55 | KN | 节点数据生成增量；可缺省； |

4. 荷载数据

（当modex=2时，这部分不输入）

|  |  |  |
| --- | --- | --- |
| 列 | 变量 | 意义 |
| 1-5 | LL | 荷载工况号。 |
| 6-10 | NLOAD | 本工况中集中载荷的个数 |
| 列 | 变量 | 意义 |
| 1-5 | NOD | 集中载荷作用的节点号 |
| 6-10 | IDIRN | 载荷作用方向（1 – x方向，2 – y方向，3 – z方向） |
| 11-20 | FLOAD | 载荷值 |

5.实体单元数据

单元组控制数据

|  |  |  |
| --- | --- | --- |
| 列 | 变量 | 意义 |
| 1-5 | NPAR(1) | 单元类型（1 – 轴力杆单元，20 - 20节点实体单元） |
| 6-10 | NPAR(2) | 本单元组中的单元总数 （≥ 1） |
| 11-15 | NPAR(3) | 不同材料/截面性质组数 |
| 16-20 | NPAR(4) | 使用的高斯积分点数（2-减缩积分，3-全积分） |

材料/截面性质数据

共读入NPAR(3)行

|  |  |  |
| --- | --- | --- |
| 列 | 变量 | 意义 |
| 1-5 | N | 材料/截面性质组号（1 ≤ N ≤ NPAR(3) ） |
| 6-15 | E(N) | 杨氏模量（字长10位，小数点后两位，F10.2） |
| 16-25 | RHO(N) | 截面面积（字长10位，小数点后两位，F10.2） |
| 26-35 | MIU(N) | 泊松比（字长10位，小数点后两位，F10.2） |

单元数据

|  |  |  |
| --- | --- | --- |
| 列 | 变量 | 意义 |
| 1-5 | M | 单元号 |
| 6-10 | I1 | 单元1号节点（字长5位，I5） |
| 11-15 | I2 | 单元2号节点（字长5位，I5） |
| 101-105 | I20 | 单元20号节点（字长5位，I5） |
| 106-110 | MTYP | 本单元的材料/截面性质组号 |
| 111-115 | KG | 单元自动生成增量；可缺省 |

### Stap90程序改写

新增solid20单元，并在相应的ecal子程序中添加相应的单元接口。

新增特征值求解子程序，并在memalloc模块添加相应的动态数组。同时在主程序中更改memalloc内存分配的顺序和位置，节点位置信息得以保留。

新增静力学输出和特征值特征向量输出的后处理模块，使得计算结果能够直接导入tecplot，进行可视化后处理。

# 程序改写模块：扩展STAP90程序

## 高阶三维实体单元

作业要求在原有STAP90程序中增加一种在商业软件中已实现的高阶三维实体单元。本次大作业以商业软件ANSYS软件的solid186单元为目标，在STAP90程序中重现了20节点六面体实体单元。本节将从单元的基本理论入手，详细介绍作业中实现单元各要素的计算方法，并说明程序实现的逻辑结构，最终通过平板单轴拉伸与悬臂梁两个算例与ANSYS中solid186单元的计算结果进行比较，验证单元的正确性。

### 单元基本理论



图 2 20节点六面体实体等参元

如图 2所示为一任意的20节点六面体实体等参元， 为结构全局坐标系，为等参元的单元局部坐标系，局部坐标系下的等参元中心坐标为，坐标取值范围为。该单元具有20个节点，其中8个节点位于角点，12个节点位于棱边中点，编号顺序如图所示。

一般意义下，20节点六面体单元位移模式的普遍形式为



即



其中单元节点位移列阵表示为



形函数矩阵为



这里为三阶单位矩阵，即



是单元位移模式的插值基函数，也称为形函数。这里根据参考文献直接给出20个节点的形函数表达式与坐标变换式



对于角节点的形函数



对于棱边节点的形函数



根据弹性力学的相关理论，三维实体单元的应变分量为



其中为单元的几何矩阵，记为如下形式



每个子矩阵的形式为



由于形函数是基于局部坐标求导得到的，矩阵中每个元素的计算将涉及到全局坐标与局部坐标间的坐标转换和相应的雅克比矩阵



代入坐标转换关系与位移模式，可以得到雅克比矩阵的计算公式为



对于三维问题，单元应力应变间的本构方程如下



其中弹性矩阵为常数阵，由材料参数计算得到



根据变分原理推导与上述推导，可以计算出单元刚度阵为



其中每个子矩阵为3阶方阵，定义为



单元质量阵的子矩阵可依据相同方法计算



### 单元程序实现

基于上述20节点单元的理论推导，参照原有STAP90程序的桁架单元，对程序作出如下的主要修改：

（1）在ELEMENT子程序中添加了单元选项接口，根据变量NAPR(1)进行判断，调用相关的单元子程序。本程序中定义NAPR(1)=20时，调用20节点的实体单元子程序SOLID20。类比于桁架单元的TRUSS子程序和RUSS子程序，本次大作业分别编写了SOLID20子程序和OLID20子程序。在子程序SOLID20中，仅为单元的信息变量分配内存，之后调用OLID20子程序对上节的单元信息进行计算。

（2）当IND等于1时，OLID20子程序读入各单元的材料信息（杨氏模量、密度和泊松比）、节点坐标信息（XYZ）及联系矩阵（LM），并调用相关子程序计算列高。此外还定义了数组NELE记录每个单元中节点对应的结构总体编号，用于后续的应力磨平。

（3）当IND等于2时，计算单元刚度阵和质量阵。首先定义弹性矩阵，并根据上述理论推导过程计算雅克比矩阵、逆矩阵、行列式等，进而得到单元几何矩阵，利用高斯积分计算单元刚度阵和单元协调质量阵，并组装至相应的总体刚度阵和总体质量阵。值得注意的是，由于对标的ANSYS中的solid186单元有减缩积分和全积分两种积分模式，改写的STAP90程序也提供了2点减缩积分和3点全积分的输入选项。随后主程序中调用其他子程序完成位移求解。

（4）当IND等于3时，计算应力和应变场。首先采用和上步中相同的内容计算单元几何矩阵和弹性矩阵，随后根据联系矩阵从结构位移计算结果中提取单元节点位移，计算单元应变场和应力场。由于应力是在高斯积分点处计算的，此处涉及应力的后续处理，本次大作业主要参照王勖成《有限单元法》中的单元应力磨平方法，具体如下：



对于20节点六面体等参元实体单元，应力磨平函数与节点的单元形函数相同，上方公式中表示高斯积分点出的应力值，共计个高斯积分点；表示磨平后节点处的应力值，共计个应力插值输出节点。单元的应力磨平即是利用高斯积分点的局部坐标计算插值矩阵的值，并根据其逆矩阵求出插值节点处的应力值。

如采用减缩积分，单元内共有个高斯积分点，按照ANSYS的solid186减缩积分的输出格式，只输出了8个角点的应力插值结果。因此在改写程序中也使用8个高斯积分点的应力值来插值8个角点的应力值。此时可直接套用参考文献中的插值矩阵结果



其中



如采用全积分，单元内共有个高斯积分点，如按照ANSYS的solid186全积分的输出格式，仍然只输出了8个角点处的应力插值结果，此时应力插值矩阵并非方阵，其逆矩阵可使用矩阵的最小二乘伪逆形式，即



扩展程序中为更好绘制应力计算结果，全积分格式采用了27个高斯积分点插值20个节点的计算形式。由于该部分计算中高斯积分点处的形函数值为常数，伪逆矩阵由MATLAB计算得到。

得到节点插值结果后，由于相同节点在不同单元中插值得到的应力值可能不同，因此要对同一节点在全部所处单元的应力结果取算术平均，才能作为最终的计算结果

### 单元验证算例

本节将首先由平板单轴拉伸算例简单验证静力学求解位移与应力场的正确性，然后通过悬臂梁算例比较不同网格划分程度和高斯积分点数目下计算结果的准确性，算例验证中以ANSYS的计算结果为标准值进行比对。

1.平板单轴拉伸

如下图 3所示为平板拉伸的网格划分情况，平板长0.6m，宽0.3m，厚0.1m，，单元尺寸为0.05m，共计941个节点，144个单元。材料参数为：杨氏模量，泊松比。左侧端面节点约束全部自由度，右侧端面施加水平向右的均布载荷，等效的节点载荷由下式推导得出



假定单元右侧面积为，根据上式积分可得到角节点等效载荷为，棱边节点的等效载荷为。本算例中计算均采用3点全积分计算格式。

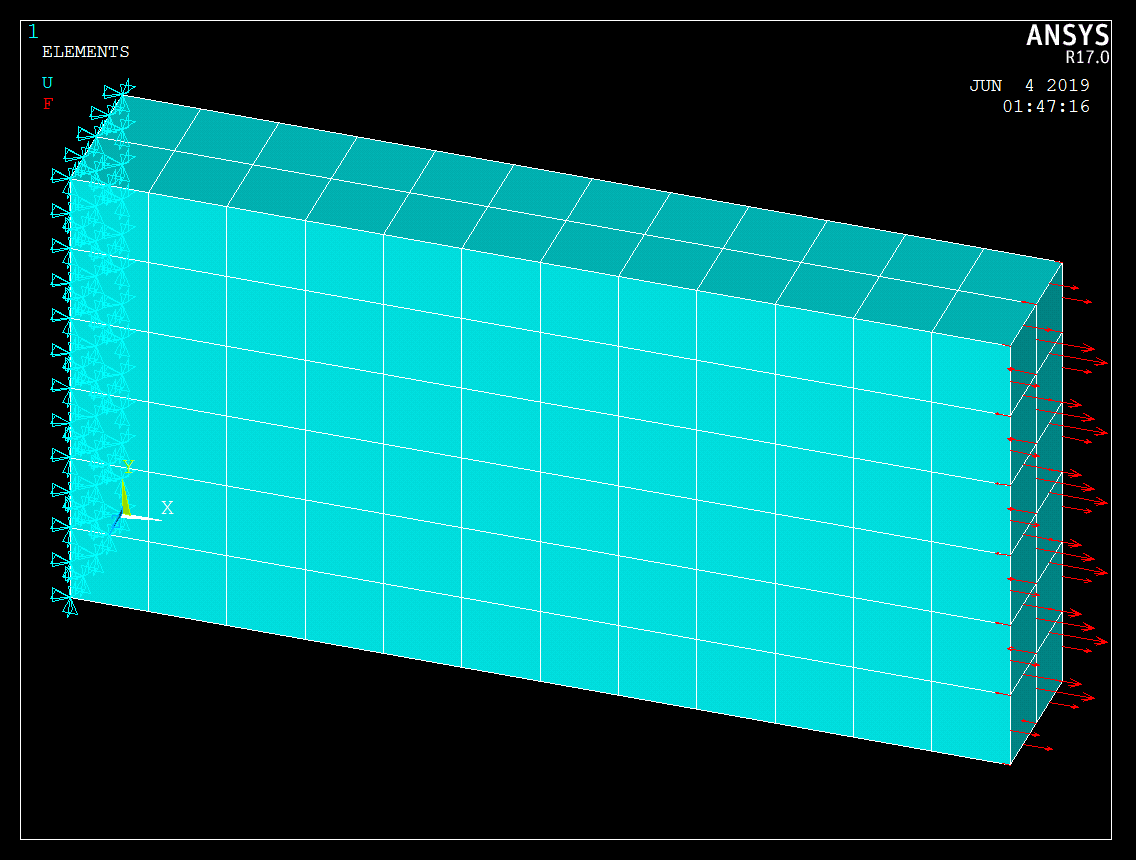
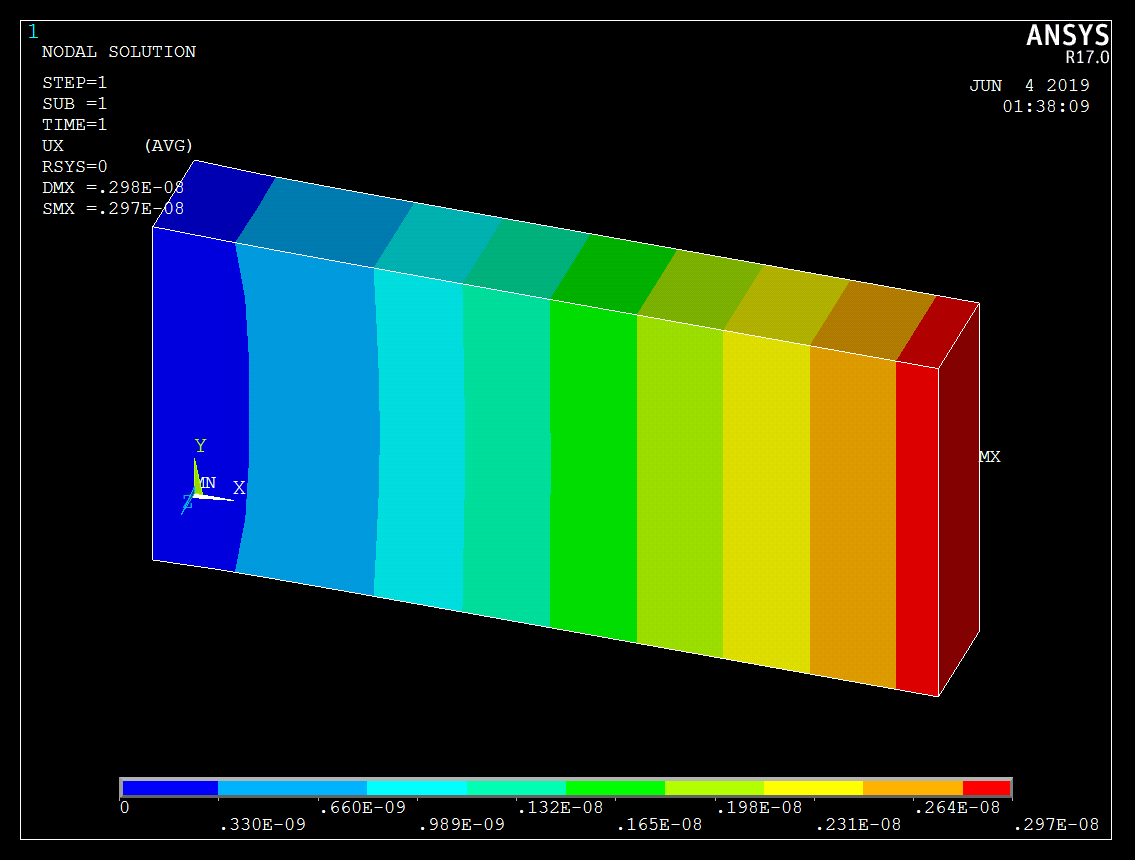
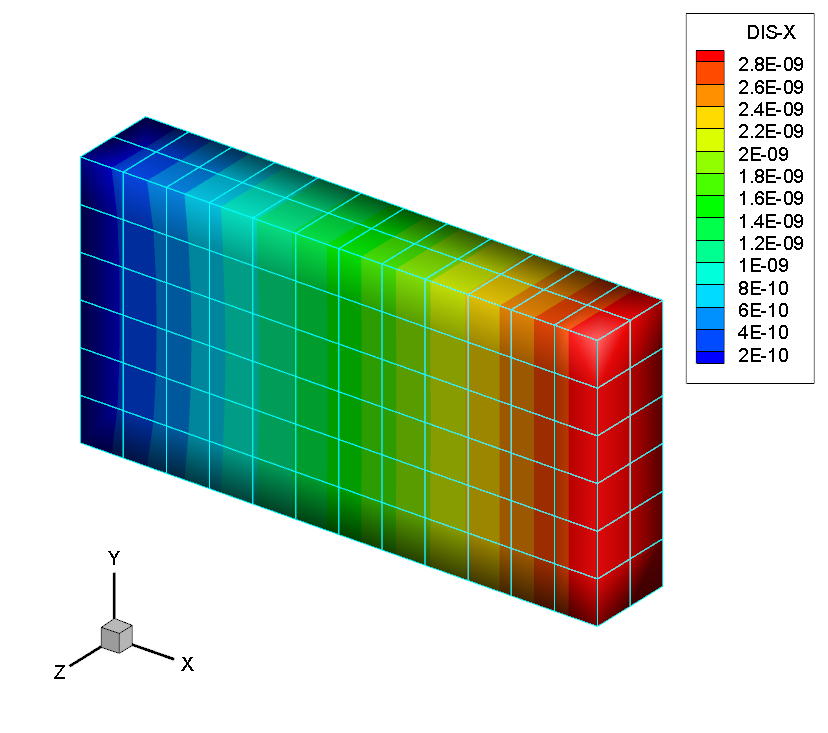
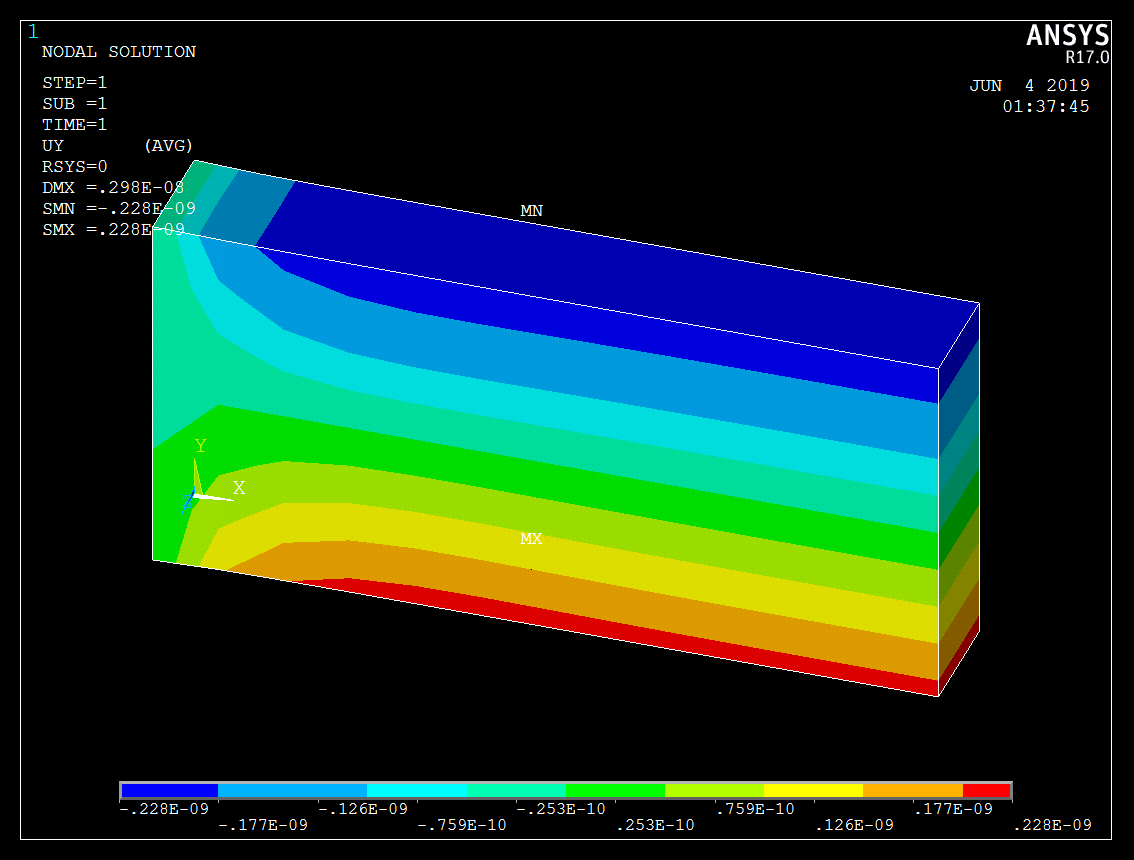


图 3平板单轴拉伸

三方向位移计算结果如下图 4所示，可以看出ANSYS计算的位移云图与改写程序计算出的位移云图分布基本完全一致，基本验证了程序的正确性。

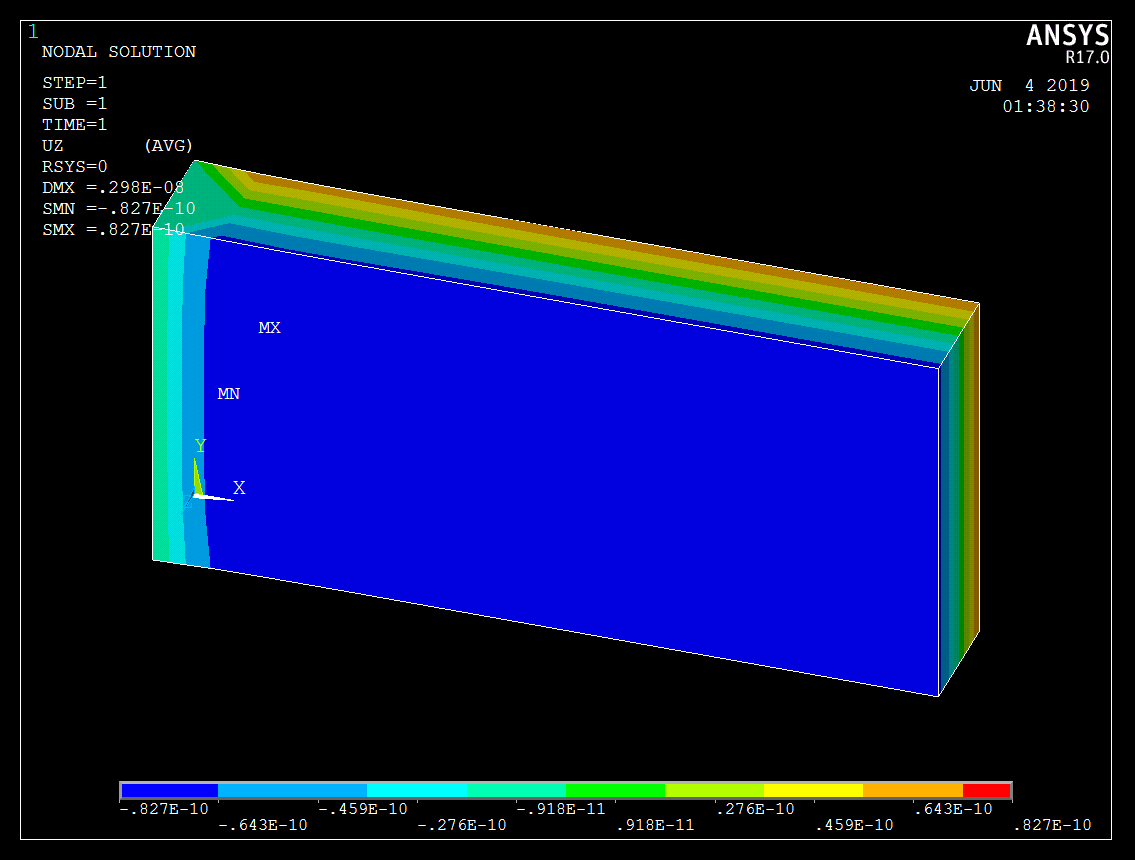
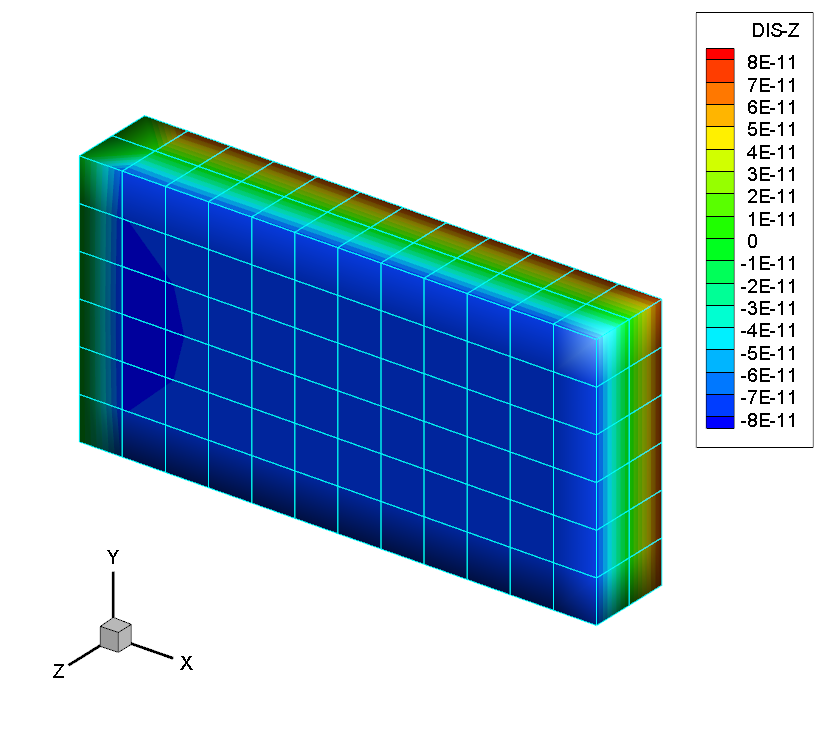
 

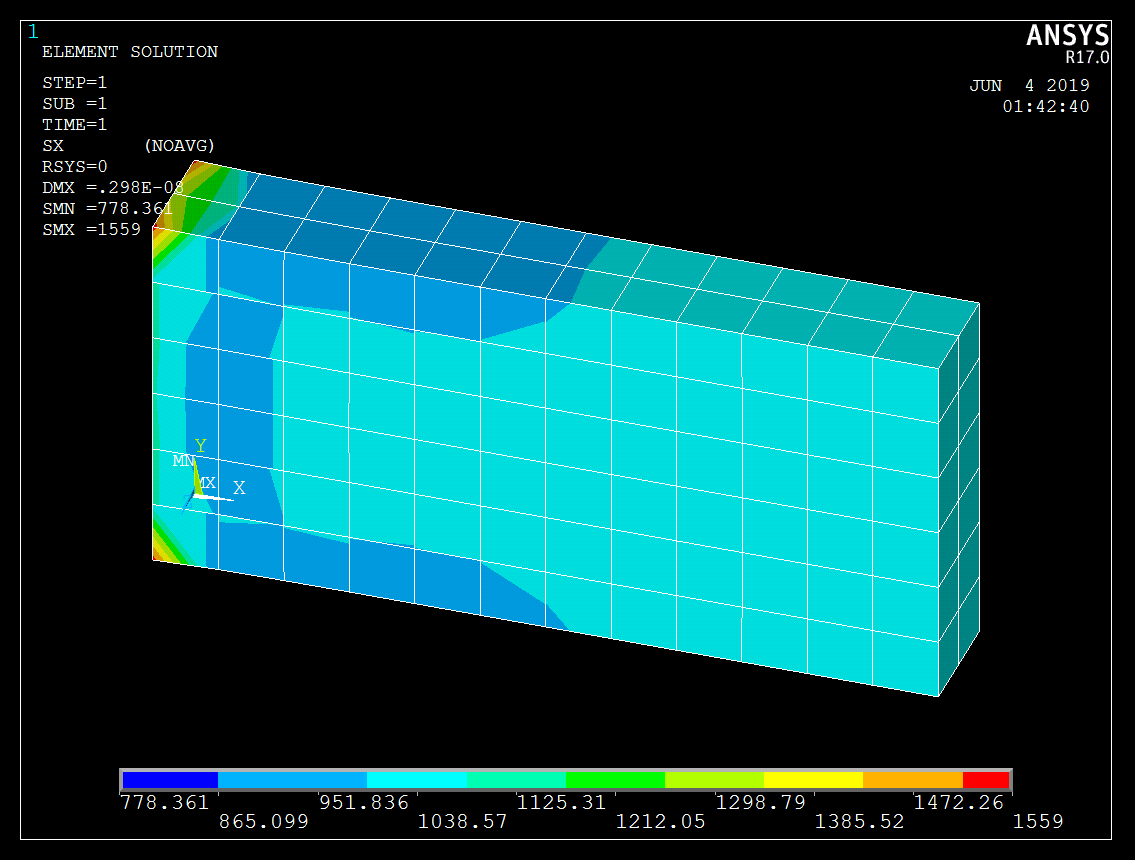
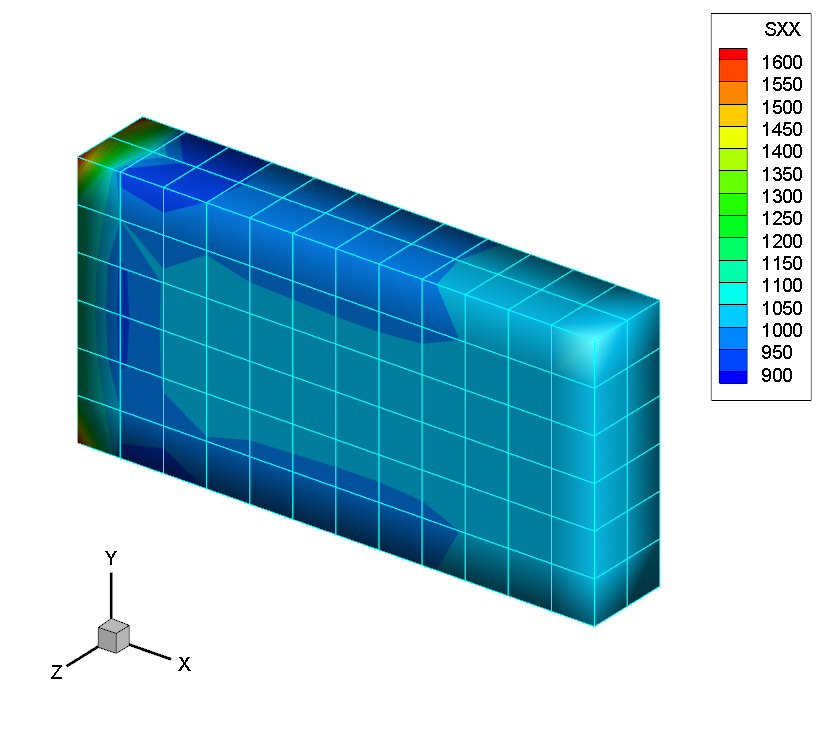
图 4平板单轴拉伸结果对比

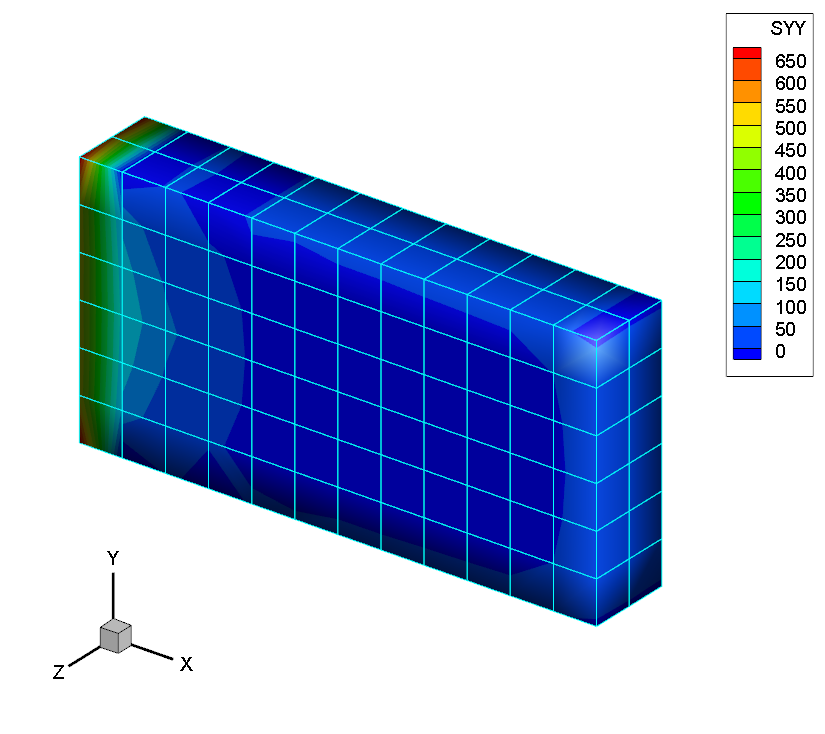
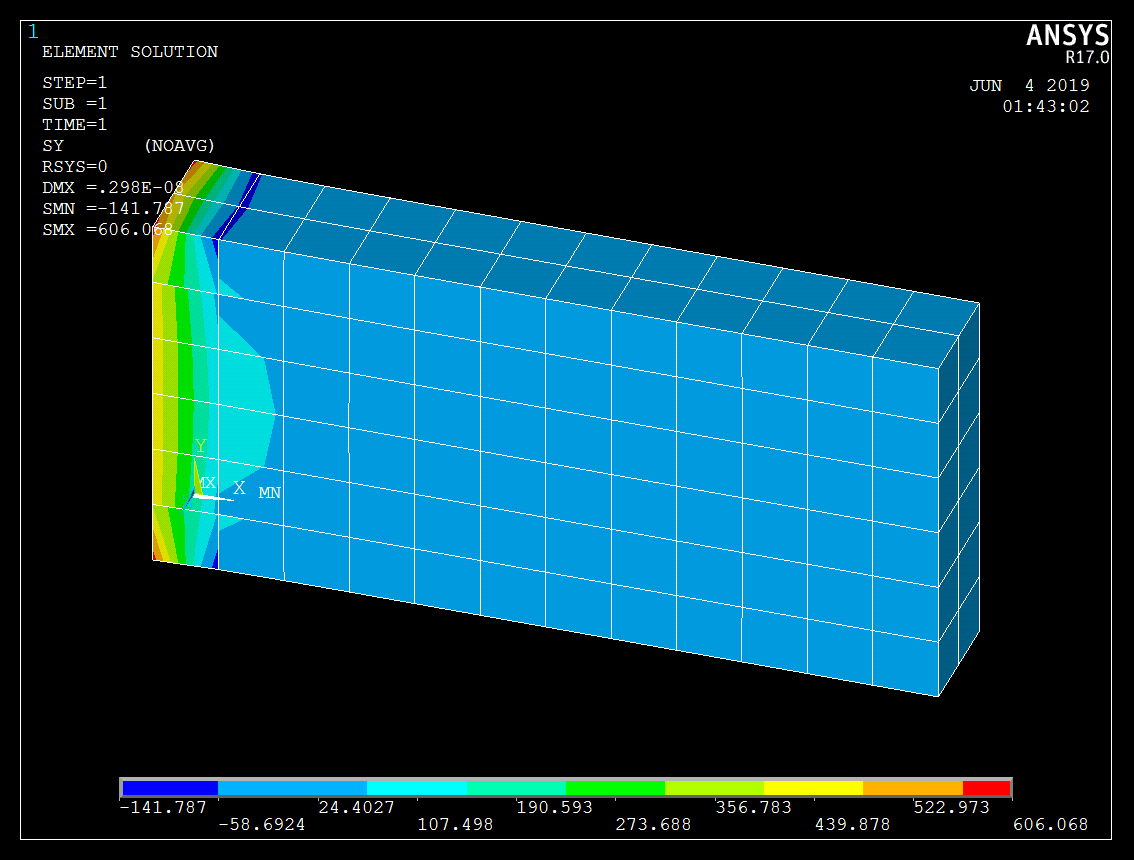
为进一步比较位移求解的数值，选取了沿x方向某条棱边的所有节点沿拉伸方向的位移值，将ANSYS与改写程序的计算结果绘制如下图 5，可以看出两者基本完全吻合，充分说明了STAP90程序实体单元位移求解的正确性。

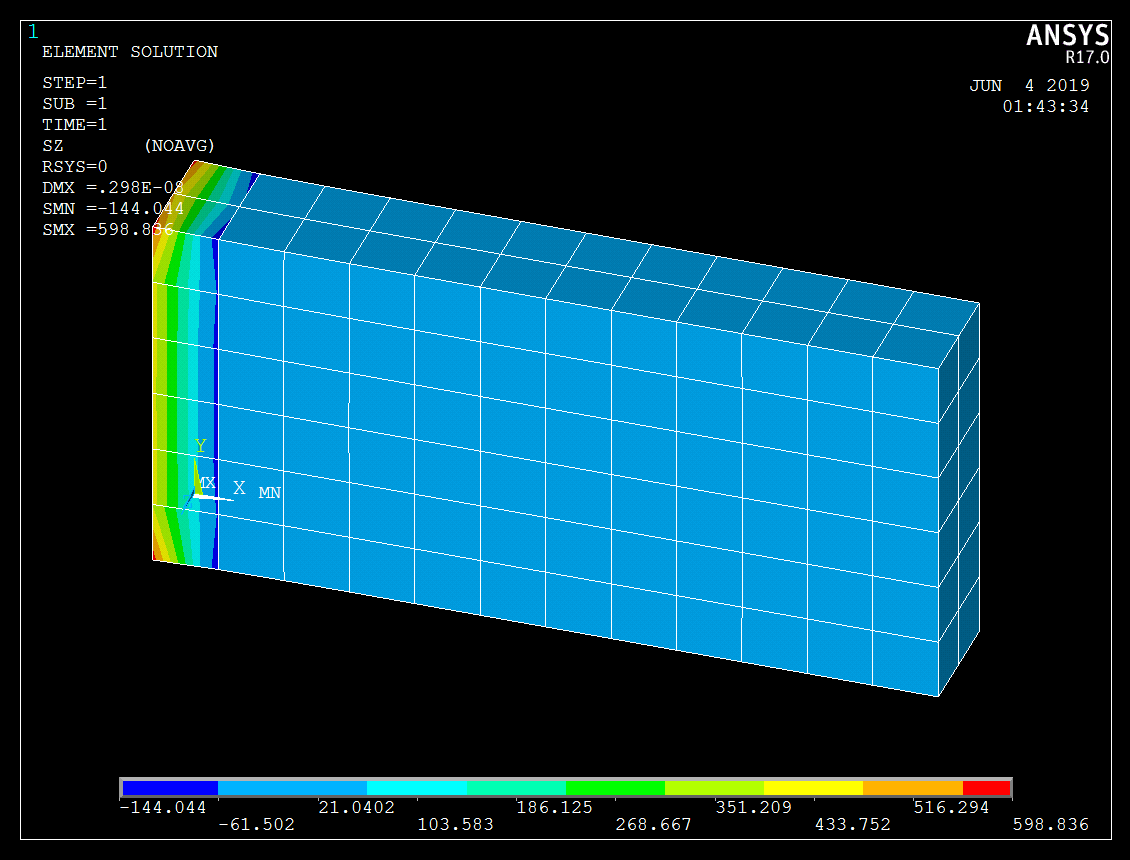
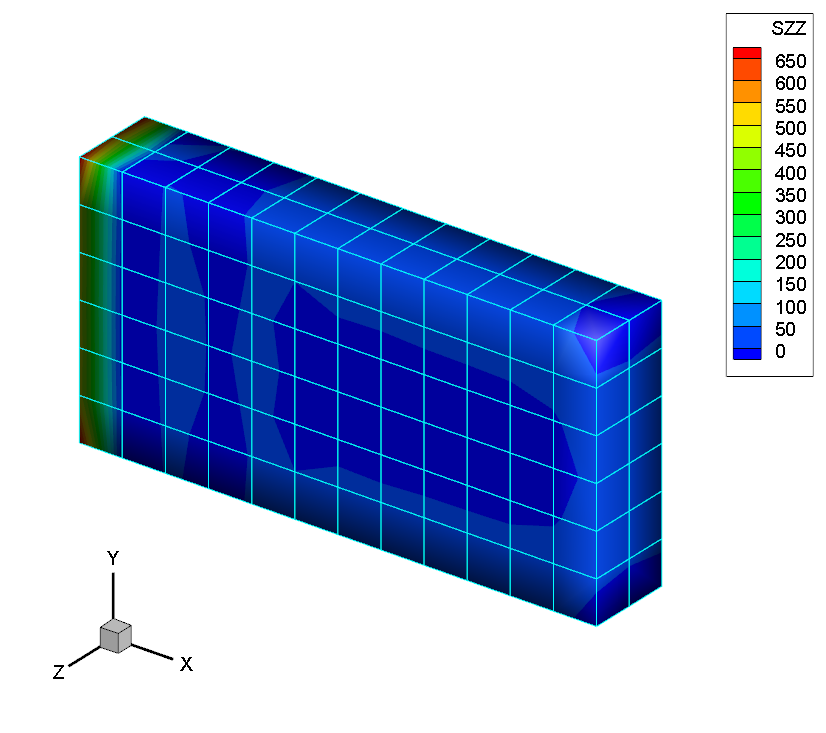


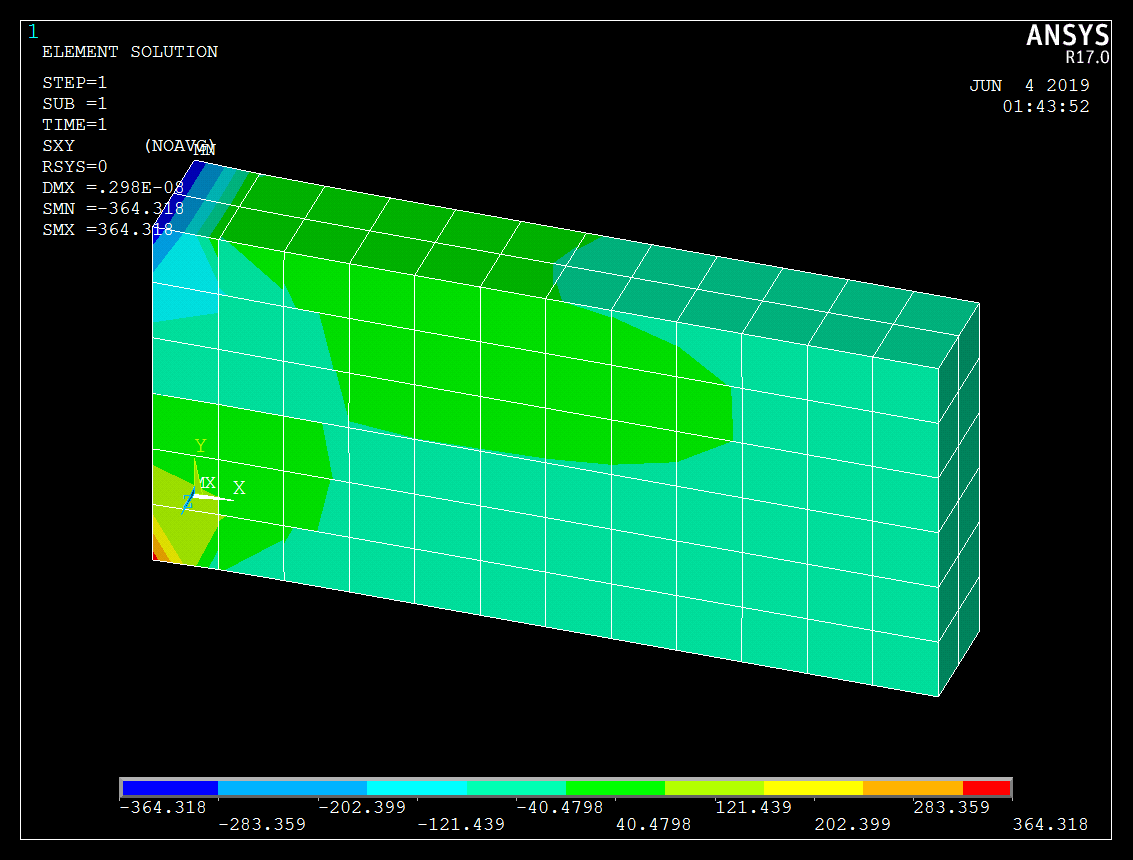
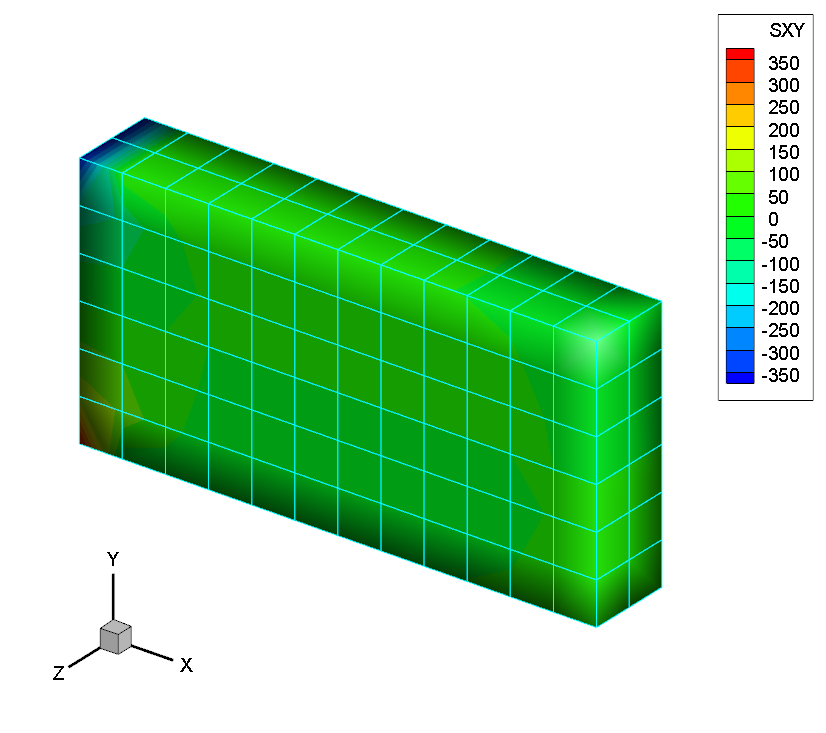
图 5 位移结果比较

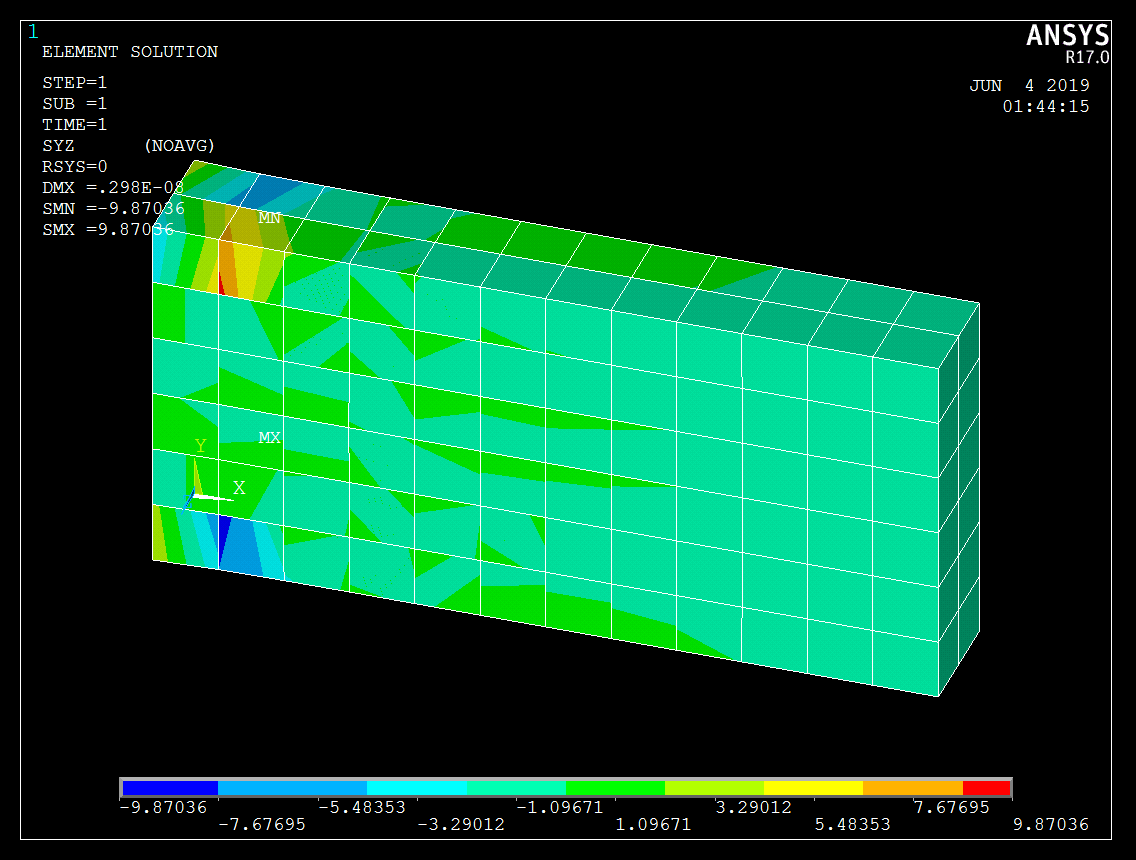
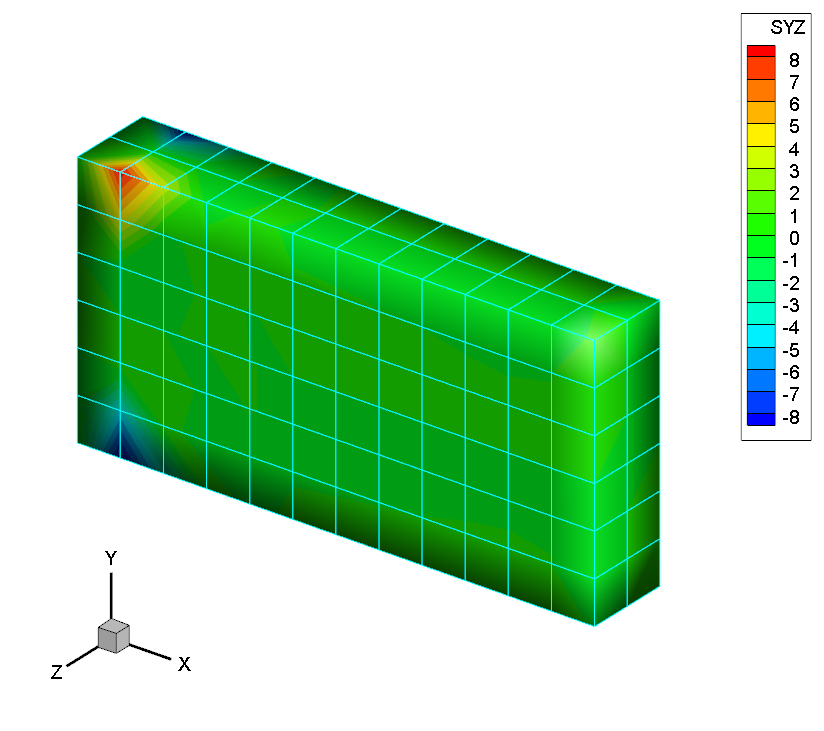
如下图 6所示，更进一步比较两种方式的应力计算结果，从云图来看，除去部分约束附近节点的应力集中问题，整体应力场分布基本保持一致，应力结果正确。



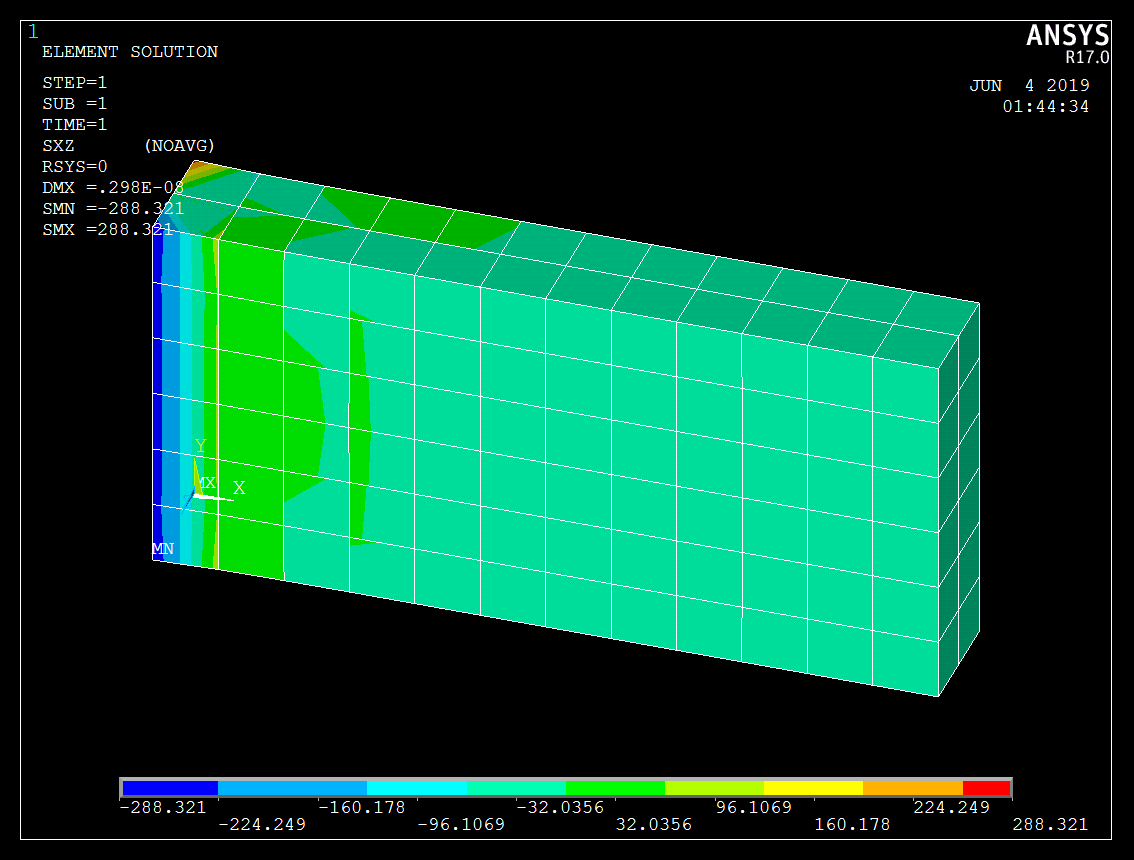
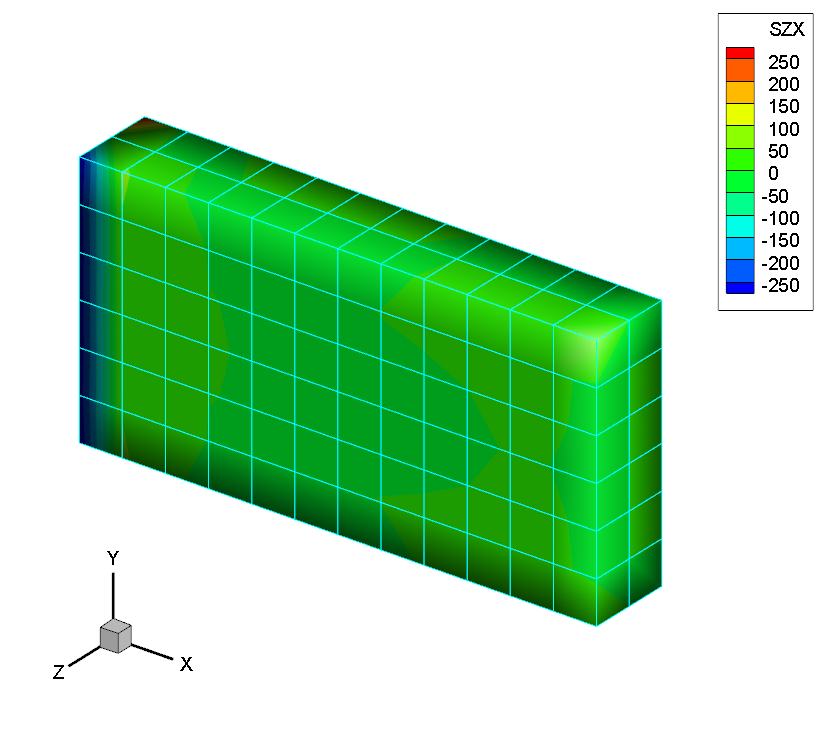
 

图 6 应力计算结果对比

与位移比较方法类似，为从数值上比较计算效果，选取正面沿x方向的中线，绘制了所有节点正应力的计算结果，如图 7，可以看出在靠近约束一侧应力存在一定的误差，但自由端的应力计算十分准确，考虑这是由应力集中所产生的应力梯度较大导致的，如加密网格可有效减小误差。



图 7 应力计算结果比较

2.悬臂梁末端受集中载荷

如下图 8图 3所示为平板拉伸的网格划分情况，悬臂梁长2m，截面为0.2m\*0.2m的正方形。为对比网格密度对计算结果的影响，我们分别用网格长度为0.1m和0.05m的网格对模型进行划分，两组划分结果的节点数量分别为621个和3665个，单元数量分别为80个和640个。材料的杨氏模量，泊松比。将模型左侧端面节点约束全部自由度，右侧端面右下角节点上施加1000N的集中载荷。

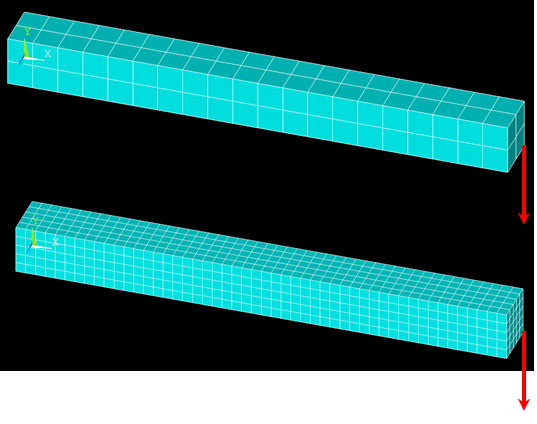


图 8 悬臂梁的两组网格划分情况

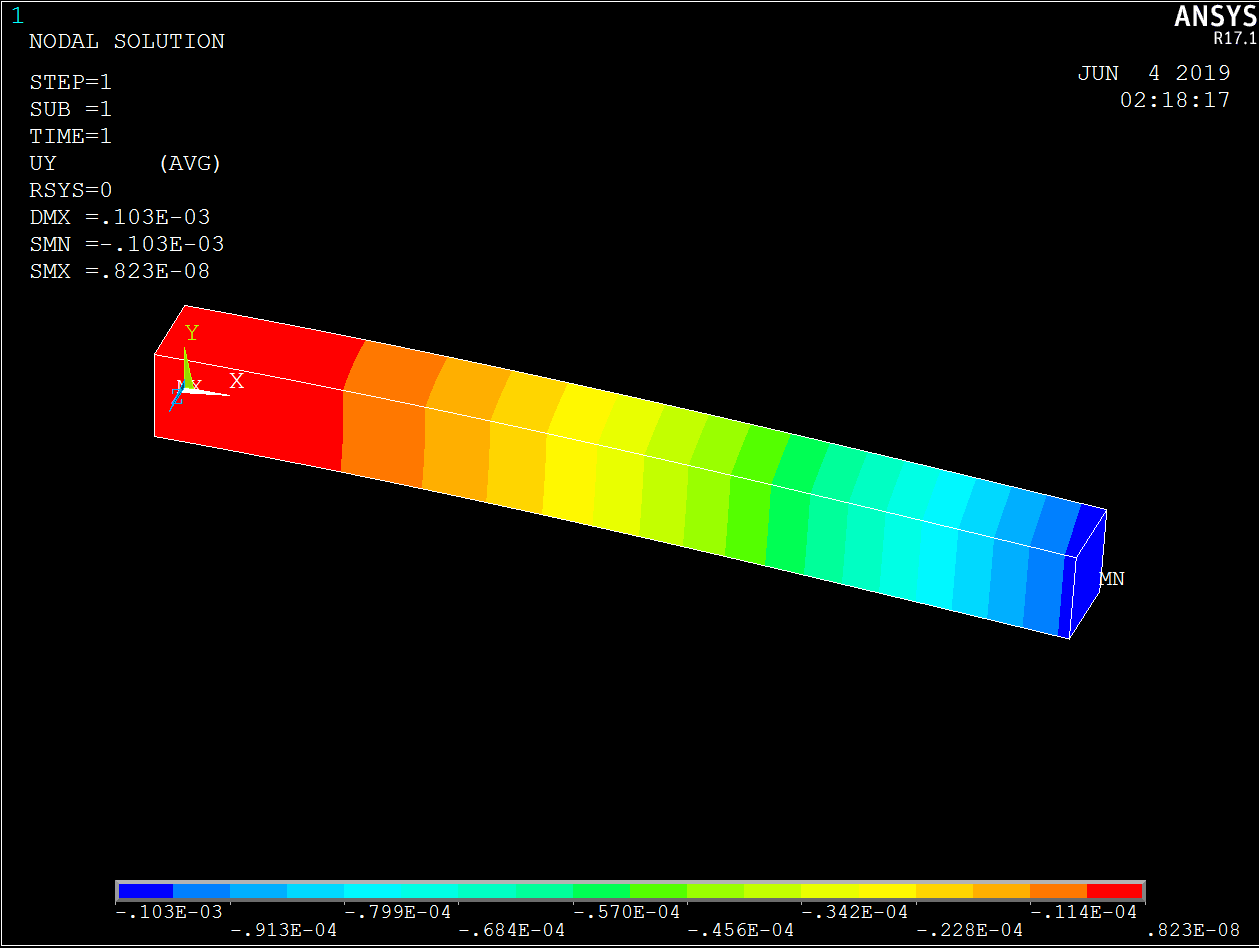
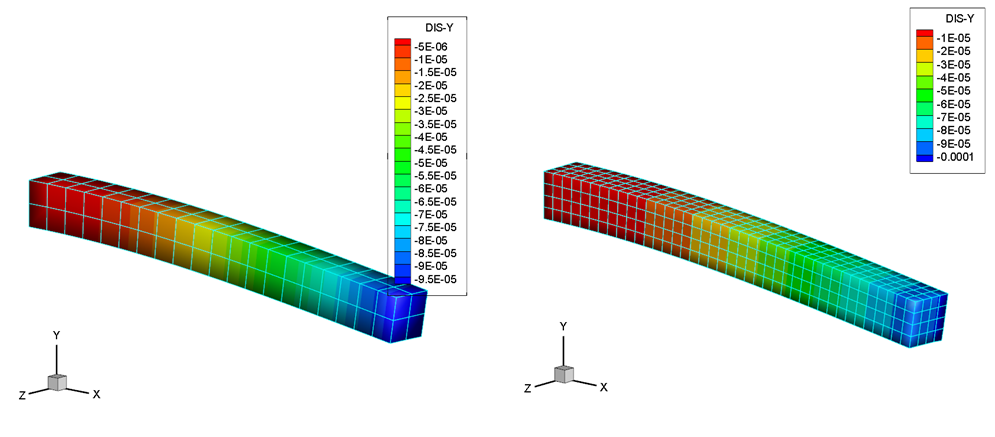
两种网格密度与ansys的640单元模型的计算所得节点y方向位移云图如

图 9所示，取其里侧竖直面中线上的一条楞作为考察对象，提取其相应节点上的y方向位移，并将其计算结果与640单元模型的ansys计算结果进行对比，如图 10所示。可以看到两种网格密度位移计算结果基本一致，且与ansys的计算结果吻合较好。



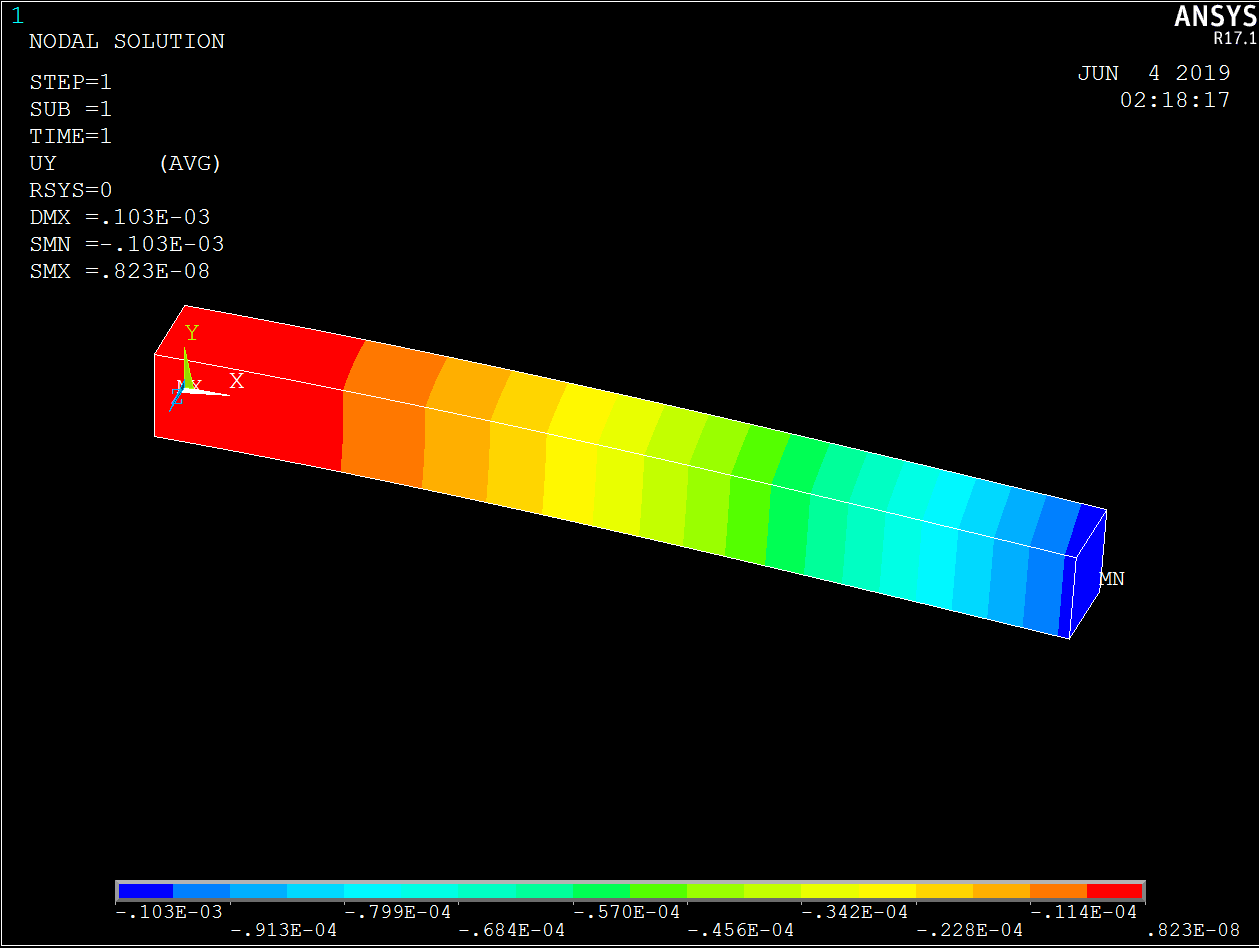


图 9 节点y方向位移云图

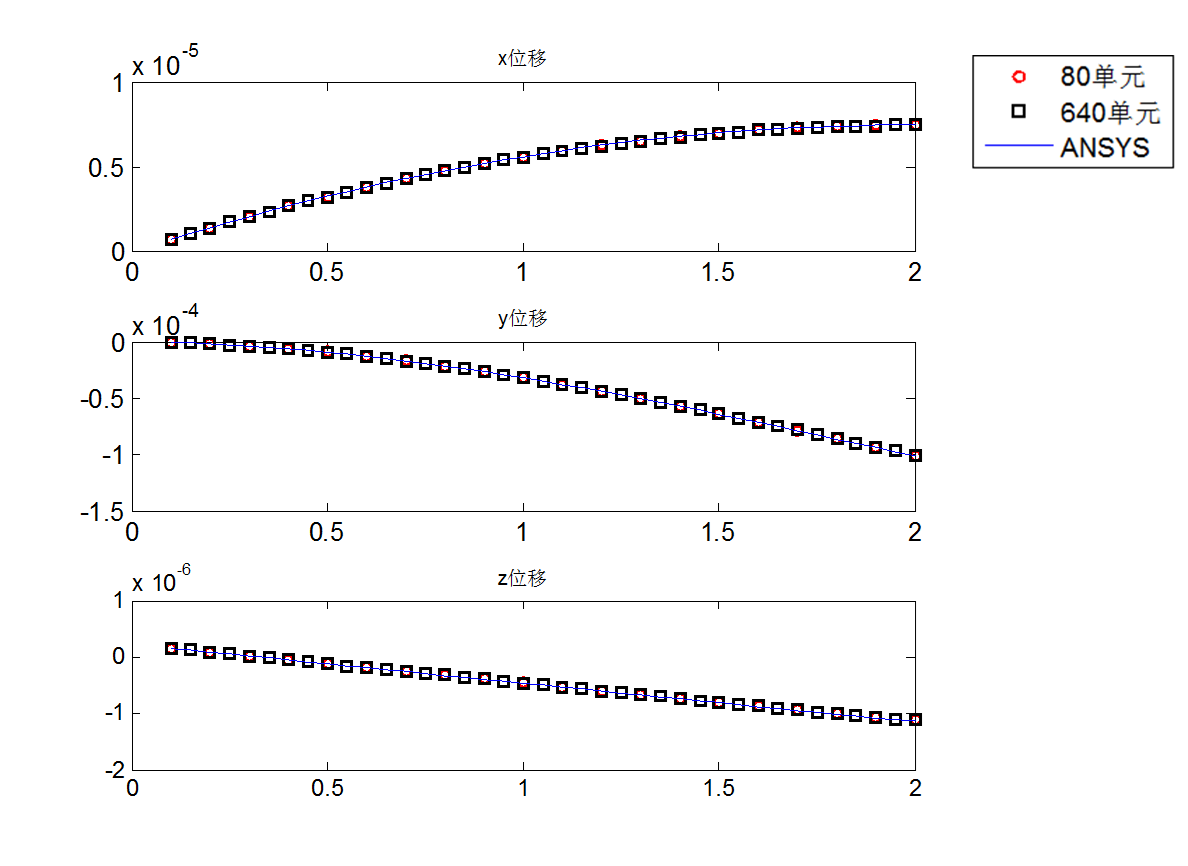


图 10 楞边上节点y方向位移计算结果

提取典型节点的位移计算结果如表 2‑1所示，这里的计算结果选用的均是680个单元的模型数据。从表中可以看出我们改写的stap90实体单元模块计算的结果与ansys计算结果基本吻合。

表 2‑1 典型节点的位移计算结果对比

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **位移分量** | **STAP90** | **ANSYS** | **相对误差** |
| **6（自由端角点)** | X | 0.744784E-05 | 0.74482E-05 | 0.0048% |
| Y | -0.100823E-03 | -0.10082E-03 | 0.0030% |
| Z | -0.110609E-05 | -0.11063E-05 | 0.019% |
| **236（外侧棱边点）** | X | 0.452118E-05 | 0.45209E-05 | 0.0062% |
| Y | -0.179334E-04 | -0.17931E-04 | 0.0134% |
| Z | -0.564570E-06 | -0.56455E-06 | 0.0035% |
| **300（内部角点）** | X | 0.261319E-13 | -0.63659E-13 | \ |
| Y | -0.423098E-04 | -0.42307E-04 | 0.0066% |
| Z | 0.914051E-14 | 0.60686E-13 | \ |

应力云图的对比如图 11所示，80节点的结果与640节点的计算结果相近，故不再重复展示。从图中可见我们编写的实体单元应力计算结果与ansys的结果相近。

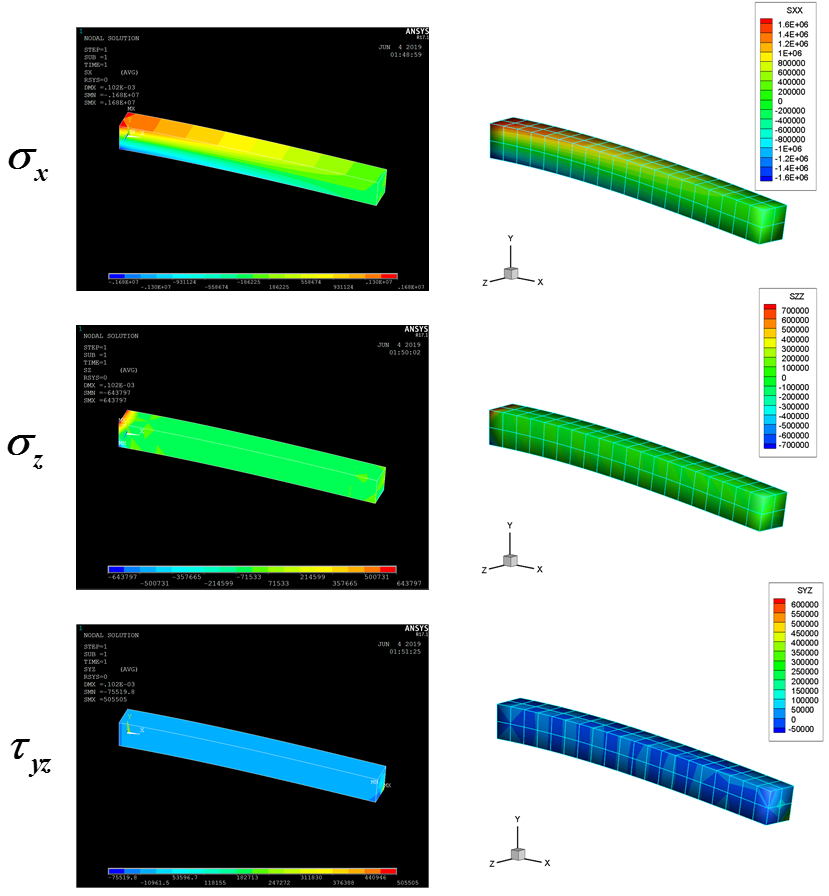


图 11 各应力云图对比图

将640单元与80单元的模型内侧中线上的节点应力计算结果与ansys的计算结果做比较（640节点的结果与80节点的结果相近，故仅仅展示640节点的计算结果），如图 12所示。从图中可得，加密网格将得到更好的应力计算结果。

综上，我们组拓展的20节点实体单元的正确的得到了验证。



图 12 不同模型节点应力对比

## 特征值求解高阶三维实体单元

作业要求结合文献调研在原有STAP90程序中增加特征值求解的功能。本次大作业在传统子空间迭代的基础上，实现了经由两次旋转向量优化后的子空间迭代法。本节将从旋转向量改进的子空间迭代法（subspace iteration method enriched by turning vectors）的算法步骤入手，详细介绍作业中实现的特征值计算子程序的计算方法，并说明程序实现的逻辑结构，最终通过胞单元与悬臂梁两个算例与ANSYS中子空间迭代法的计算结果进行比较，验证程序和算法的正确性。

### 旋转向量改进的子空间迭代法

旋转向量改进的子空间迭代法在每次迭代中主要有以下几个步骤：

1. 将特征向量分为三类：对应特征值已经收敛的特征向量；需要加速收敛的特征向量；辅助特征向量：



其中，中分别有个列向量。若记刚度（质量）矩阵维数为n，待求特征值-特征向量对个数为,则

1. 仅对第二类特征向量进行一次迭代



1. 计算
2. 按照反序计算迭代向量的旋转向量：



建立评估标准：



若，则跳转至下一个i

若，则计算：



1. 利用形成

首先初步构建：



再次构建旋转向量



利用旋转向量修正：



1. 计算和



1. 计算缩减自由度之后的刚度矩阵和质量矩阵



1. 计算广义特征值问题



1. 利用strum序列判断收敛性

### 程序伪代码

以上计算过程中有大量的矩阵相乘，为了缩减计算量，程序中引入变量A,B,C,D对多次出现的计算结果进行保存。算法的具体伪代码如下：

表 2‑2 伪代码

|  |  |
| --- | --- |
| 说明 | 计算 |
| 分解刚度矩阵 |  |
| 开始迭代 |  |
| （a） |  |
| （b） |  |
| 结束迭代 |  |
| 检查strum序列 |  |

### 程序实现

基于上述伪代码，对原有STAP90程序作出如下的主要修改：

（1）添加子程序SSPACE90ENRICHED.f90，实现旋转向量改进的子空间迭代算法。在该子程序中，质量矩阵和刚度矩阵都采用一维变带宽存储。一维变带宽的矩阵算法参见SSPACE90.f90

（2）程序需要根据问题规模和待求特征值特征向量个数来对保存特征值和特征向量及特征值计算的中间变量的矩阵进行内存分配，所以在memalloc模块中，创建可变数组，如图 13。在主程序调用求解特征值函数时，根据问题规模及需要的特征值的个数，动态分配数组大小，如图 14

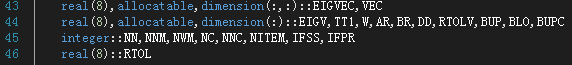


图 13 memalloc模块添加部分

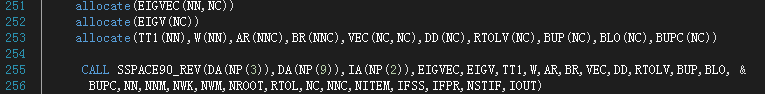


图 14 主程序中添加部分

### 验证算例

本节将首先由胞单元算例简单验证特征值计算程序的正确性，然后通过悬臂梁算例验证特征值计算程序的准确性，算例验证中以ANSYS的计算结果为标准值进行比对。

1. 胞单元算例

如图 15所示为胞单元的边界条件设计，该单元一个面的自由度被固定，特征值和特征向量计算结果及与ansys子空间迭代法的计算解过如见表 2‑3

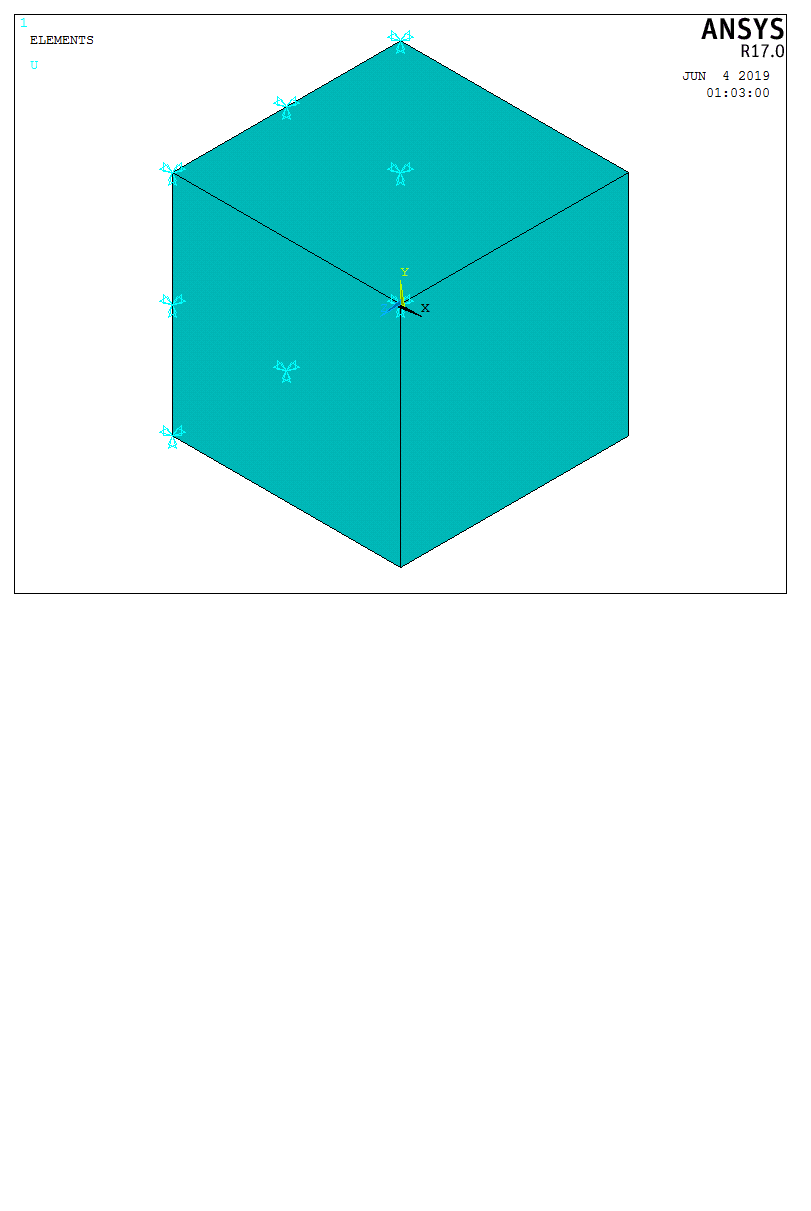


图 15 胞单元的边界条件

表 2‑3 胞单元计算结果与ansys对比

|  |  |  |
| --- | --- | --- |
| 模态 | Ansys计算结果（Hz） | 改编程序计算结果（Hz） |
| 1 | 14245 | 14232.23 |
| 2 | 14245 | 14232.23 |
| 3 | 18849 | 18843.96 |
| 4 | 32758 | 32758.24 |
| 5 | 37635 | 37582.37 |
| 6 | 37635 | 37582.37 |

|  |  |
| --- | --- |
|  |  |
|  |  |
| 图 16 胞单元特征向量计算结果 | |

2. 悬臂梁算例

如图 17所示为悬臂梁的边界条件设计。该悬臂梁由120个solid20单元纵向排列组成，其一端的自由度被固定，特征值和特征向量计算结果及与ansys子空间迭代法的计算解过如见表 2‑4

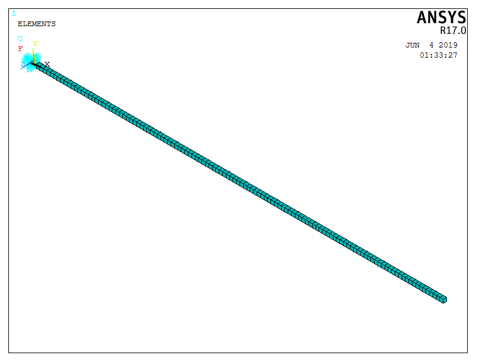
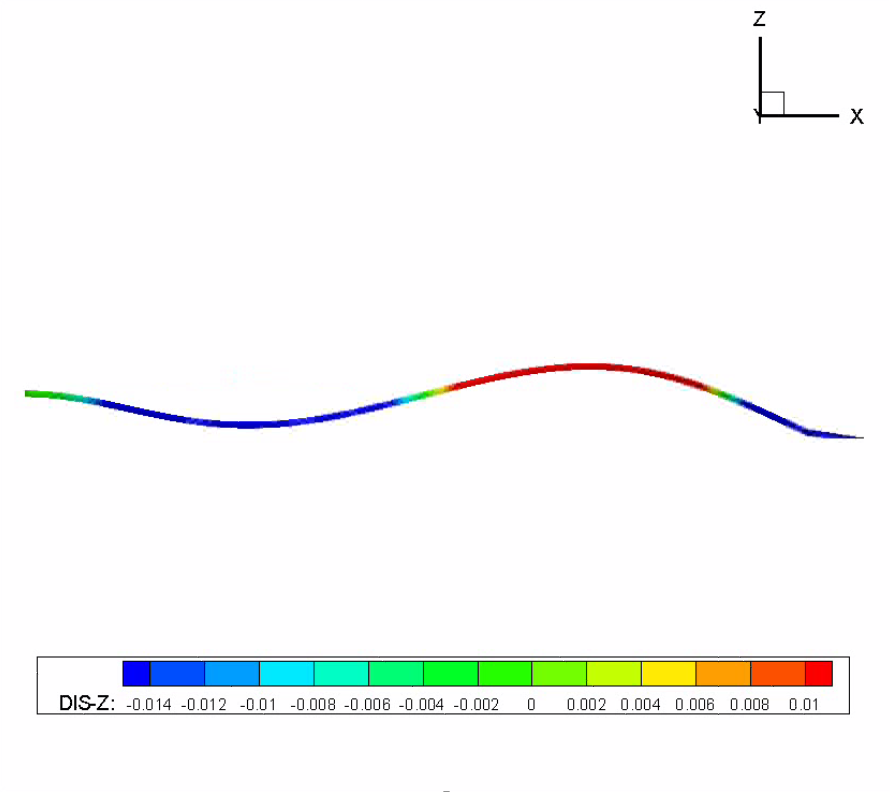
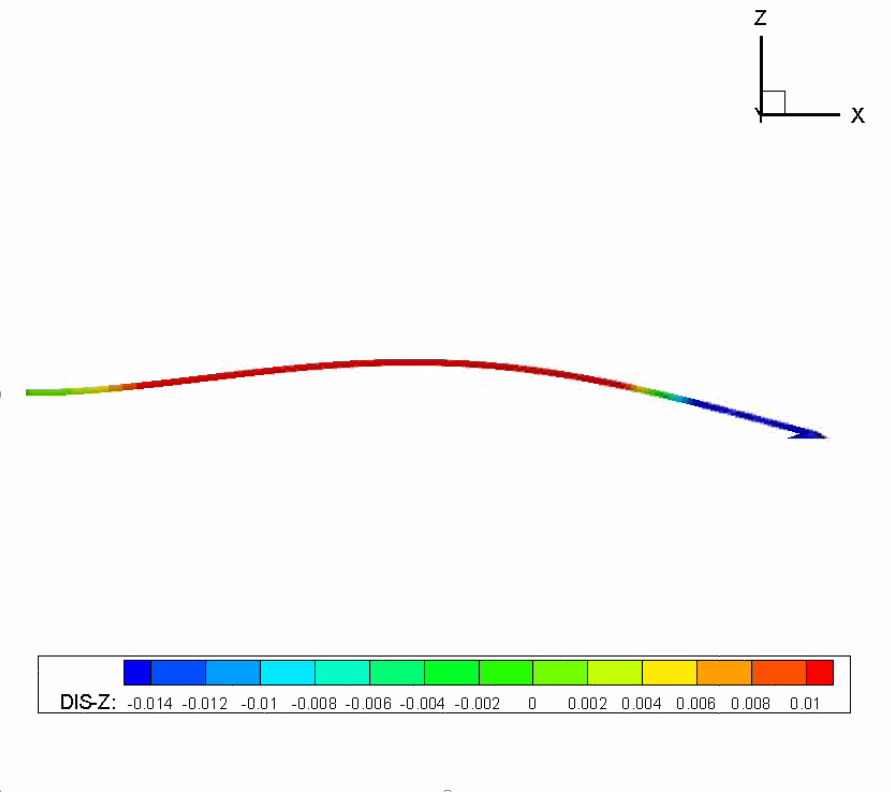


图 17 悬臂梁边界条件

表 2‑4 悬臂梁算例的特征值计算结果比较

|  |  |  |
| --- | --- | --- |
| 模态 | Ansys计算结果（Hz） | 改编程序计算结果（Hz） |
| 1 | 4.0221 | 4.022 |
| 2 | 4.0221 | 4.022 |
| 3 | 25.199 | 25.197 |
| 4 | 25.199 | 25.197 |
| 5 | 70.524 | 70.521 |
| 6 | 70.524 | 70.521 |
| 7 | 138.10 | 138.099 |
| 8 | 138.10 | 138.099 |
| 9 | 228.10 | 228.088 |
| 10 | 228.10 | 228.088 |



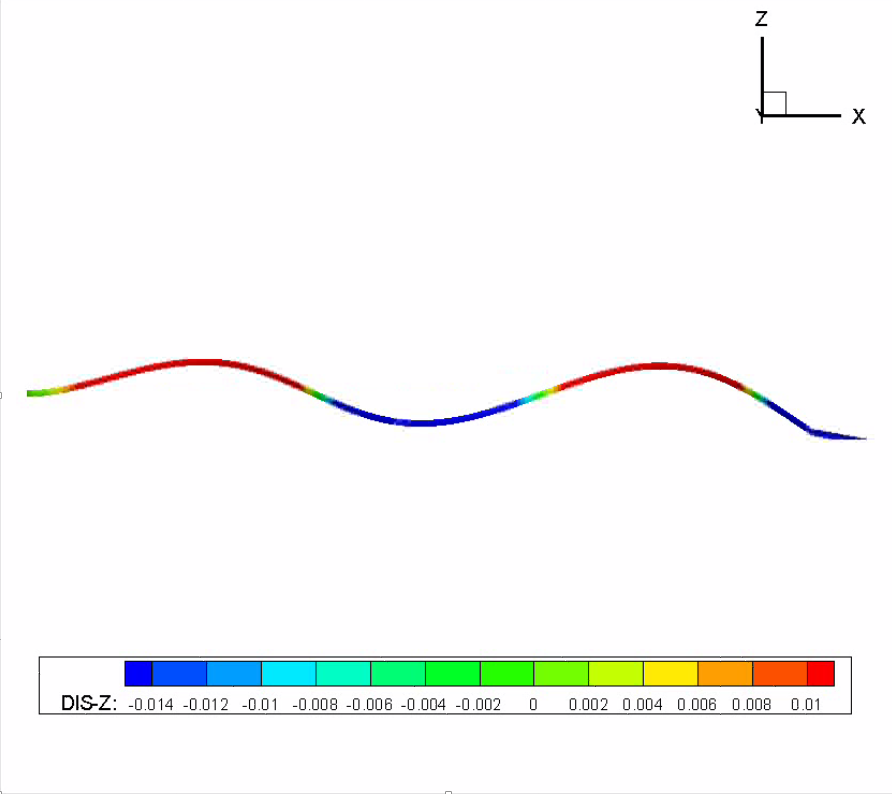


图 18 悬臂梁算例的特征向量计算结果

## 后处理读取

在STAP90程序中新增DataOut.f90，根据“Tecplot 360”软件的输出格式输出dat文件，作为改写程序的后处理。具体输出格式如下：

### 静力数据读取

以均布载荷拉伸板模型的后处理输出文件为例

TITLE = "STATIC SOLUTION "

说明：文件名字

VARIABLES= "X" "Y" "Z" "DIS-X" "DIS-Y" "DIS-Z" "SXX" "SYY" "SZZ" "SXY" "SYZ" "SZX"

说明：变量名称定义

ZONE T="Time = 0.0000E+00" F=FEPOINT N= 941 E= 144 ET=BRICK C=CYAN

%一个zone， T来定义ZONE的标题。F表示有限元节点。N表示节点个数。E表示单元个数。ET表示单元类型，这里BRICK是6面体单元的意思。C表示单元颜色。

在标题行之后根据定义的变量名称输出数据，每一列为一个变量，每一行为一个节点。完成节点数据输出后再输出单元节点编号，每一行代表一个单元。

### 振型数据读取

根据以上的格式，每个振型作为一个ZONE输出一组节点数据，输出数据格式与上述相同。（注意：单元节点编号不需要重复输出，只需要在标题行最后加上 D=(FECONNECT)）

ZONE T="Time = 3.6222E-05" F=FEPOINT N= 861 E= 640 ET=BRICK C=CYAN D=(FECONNECT)

### 后处理操作

1、点击工具栏中的“File”按钮，选择”Load data”，读入.dat文件

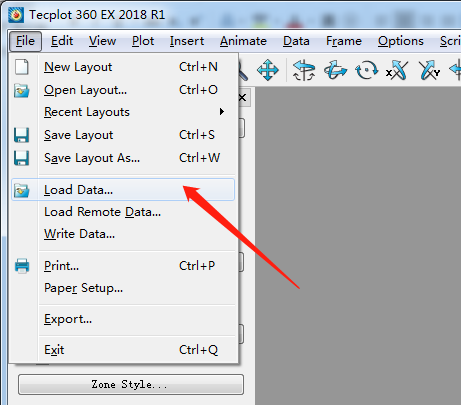


图 19 操作步骤1

2、读入数据后在左侧工具栏中勾选“Mesh”和“Contour”

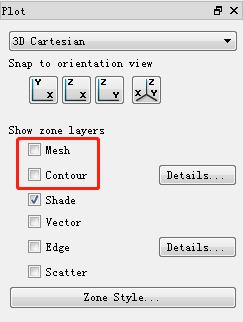


图 20 操作步骤2

3、点击上述图中的”Detailes”按钮，出现如下对话框，可以调整后处理的显示，也可以选择云图所画的数据。

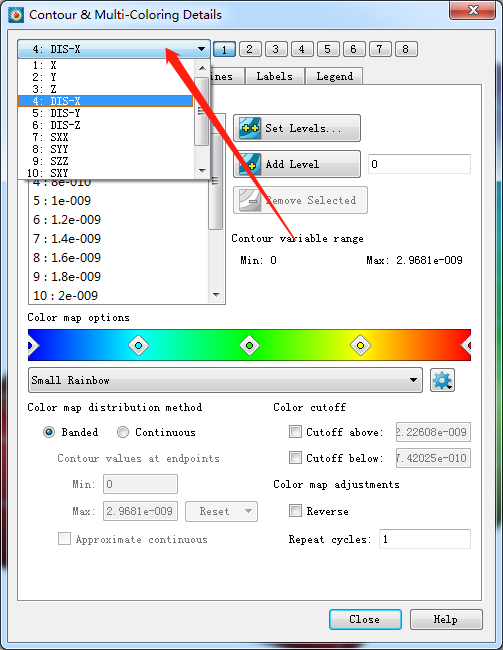


图 21 操作步骤3

4、对于振型数据，可以选择上部工具栏中“Animate”下的” Zones”选项，在对话框中选择当下的第n阶振型，也可以将各阶振型输出为视频格式。



图 22 操作步骤4.1

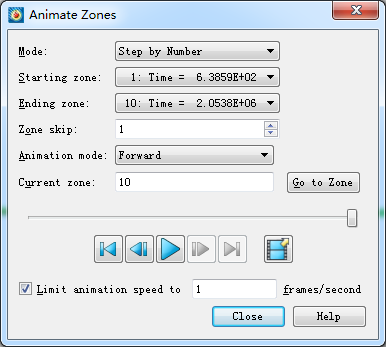


图 23 操作步骤4.2

# 实际问题分析模块

## 桁架模型分析

### 理想桁架的基本假设

在工程实际中，厂房、桥梁、起重机、油田井架、电视塔等大跨度建筑物常使用桁架结构，这种结构具有自重小、承载能力强、跨度大、可充分利用材料等优点。桁架是由若干直杆状构件在两端以一定的方式连接起来的结构，详细分析桁架中的各个构件以及连接处的受力是十分复杂的，需充分考虑材料力学、弹性力学与结构力学的知识。因此实际计算或初步设计中，一般会引入一定的假设，对桁架做适当简化后再进行分析。基于引入误差小和计算偏“保守”的原则，可对桁架作如下假设：

1. 桁架结构都是直的刚杆，杆的轴线是直线，并通过连接铰链的中心；
2. 杆件在端点处以光滑铰链相连接；
3. 杆件的自重不计，且支座的约束反力及载荷均作用于节点上。

满足以上假设的桁架称为理想桁架，在这些假设下，桁架的各个杆均为二力杆，即杆所受内力必须沿着杆的方向，且是单纯的拉力或压力。

### 实体单元桁架模型计算

选取如图所示10m\*20m的桁架结构作为分析的对象，截面为1m\*1m的正方形，材料弹性模量为200Gpa。节点载荷如图所示。

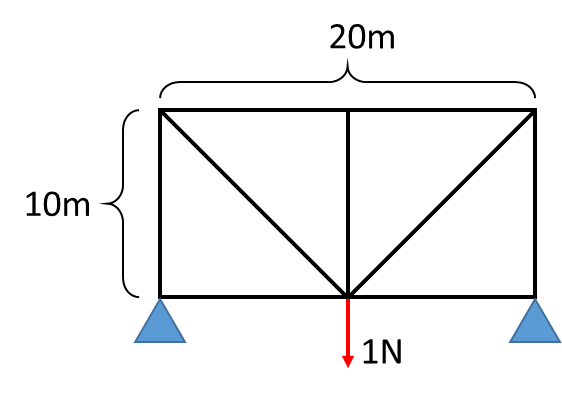


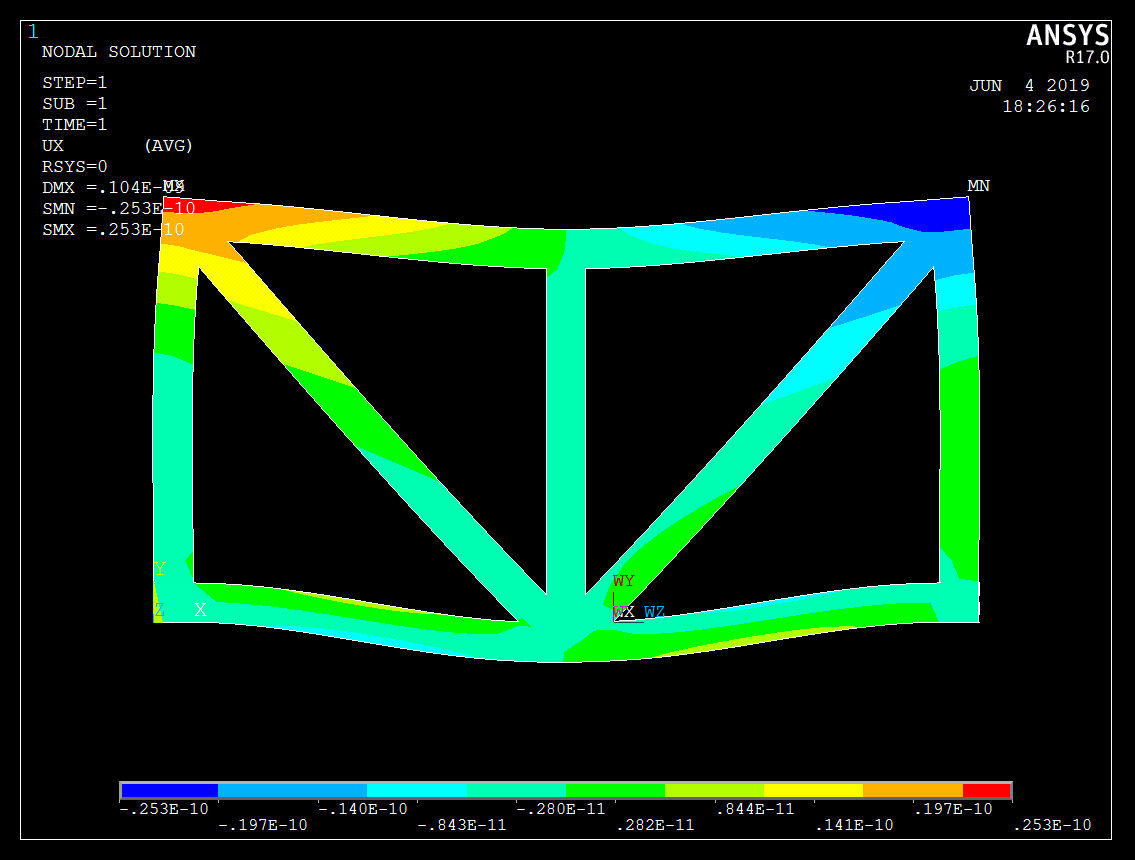
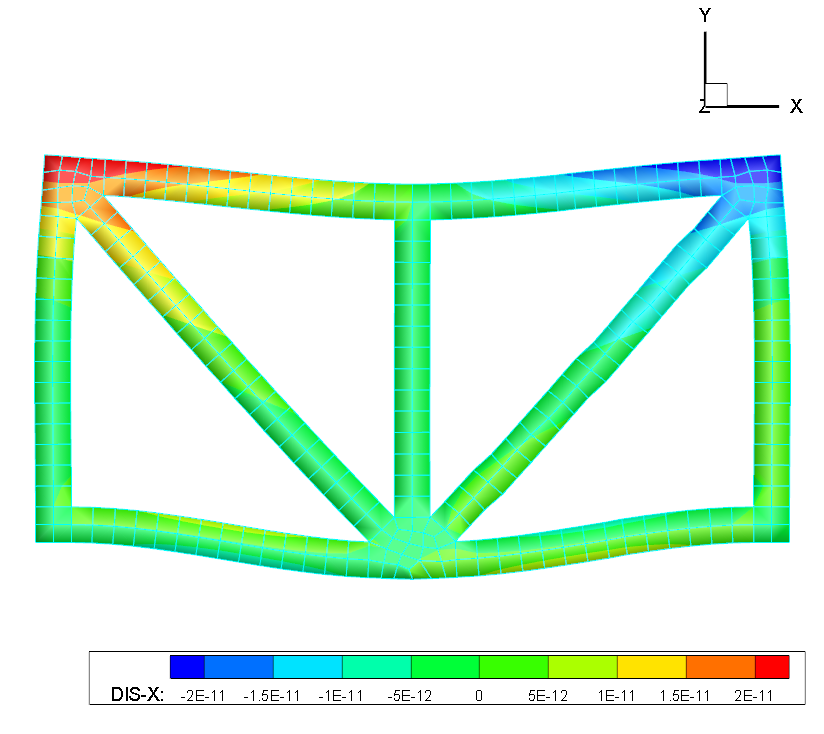
图 24 桁架结构示意图

采用ansys对模型进行前处理，网格划分如下图所示，其中节点数量4957个，单元数量680个。



图 25 桁架结构网格划分图

我们改写程序（左）与ansys（右）的位移计算结果（x方向、y方向）如下图所示，可以看到我的改写的程序对本桁架结构模型计算所得的位移结果与ansys基本一致。



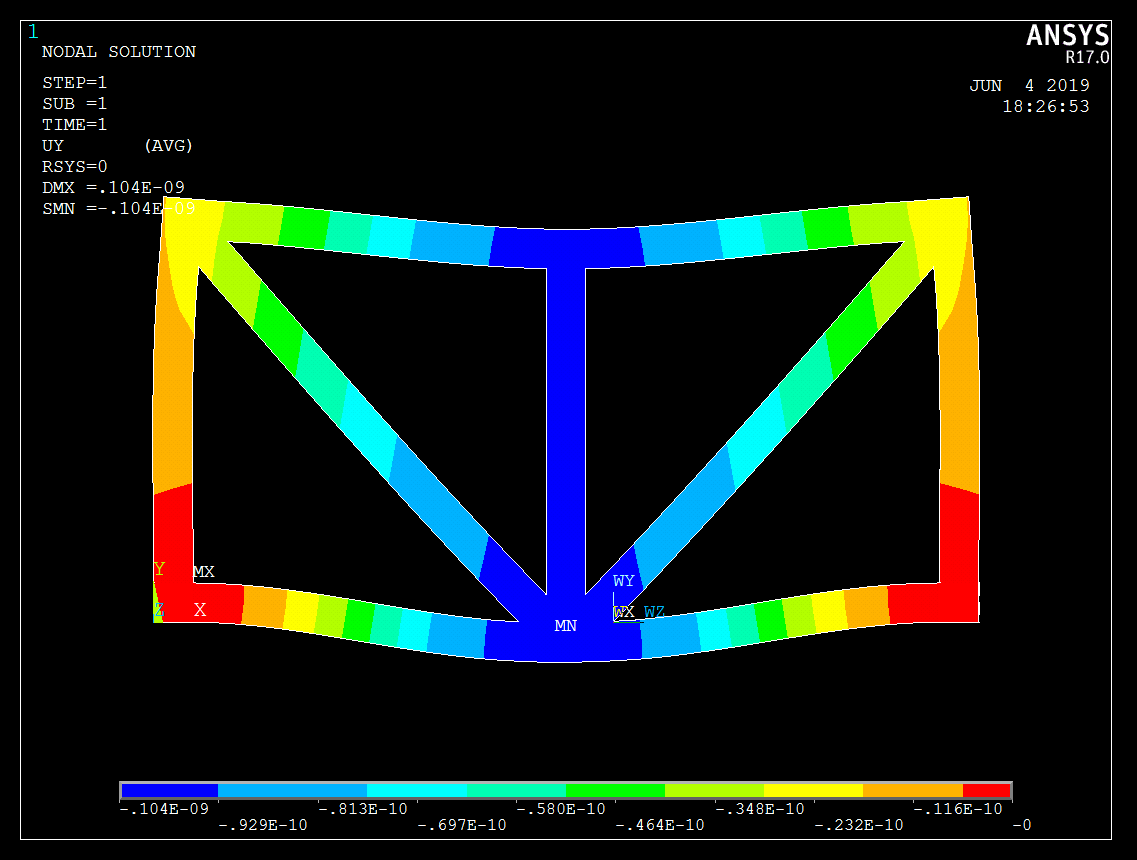
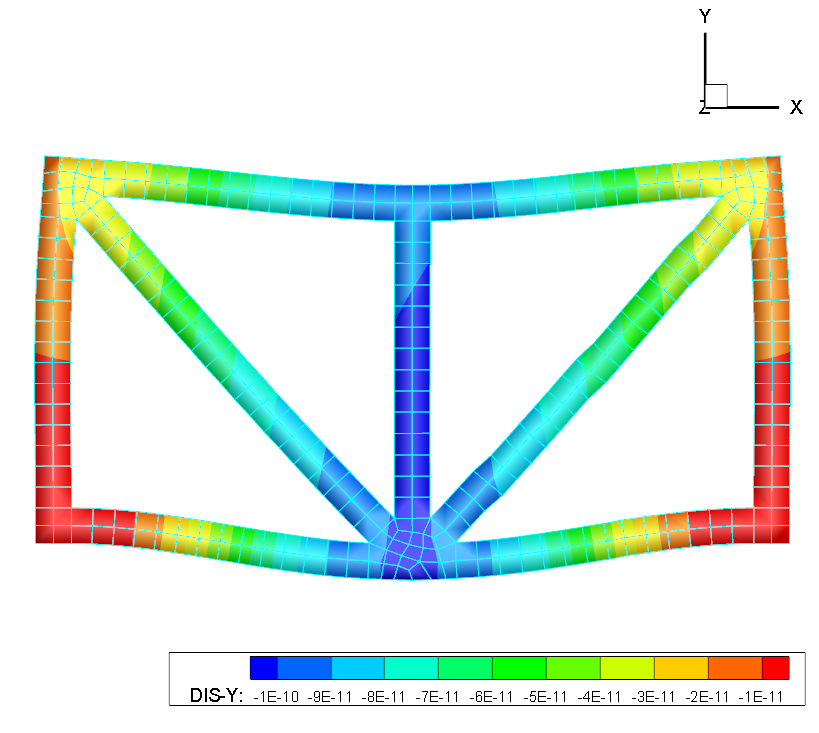


图 26 桁架结构位移计算结果对比

提取改写程序计算所得的mises应力如下图所示。该结果与杆单元的计算结果接近，即杆内应力分布均匀，其应力数值与杆单元相应应力数值接近。

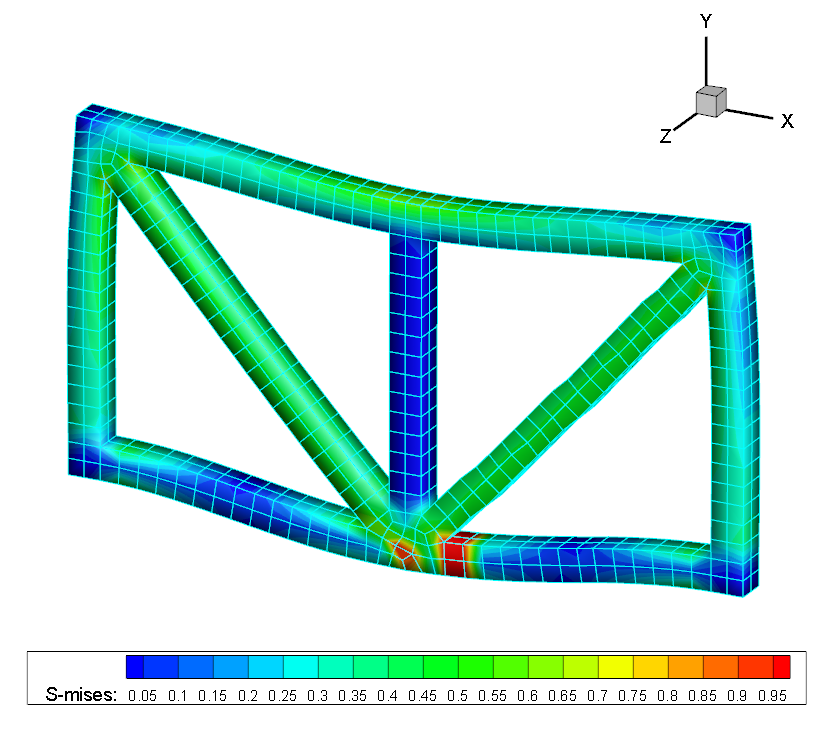


图 27 实体单元桁架结构mises应力云图

这一结果也与ABAQUS中桁架单元计算所得的应力情况基本一致。

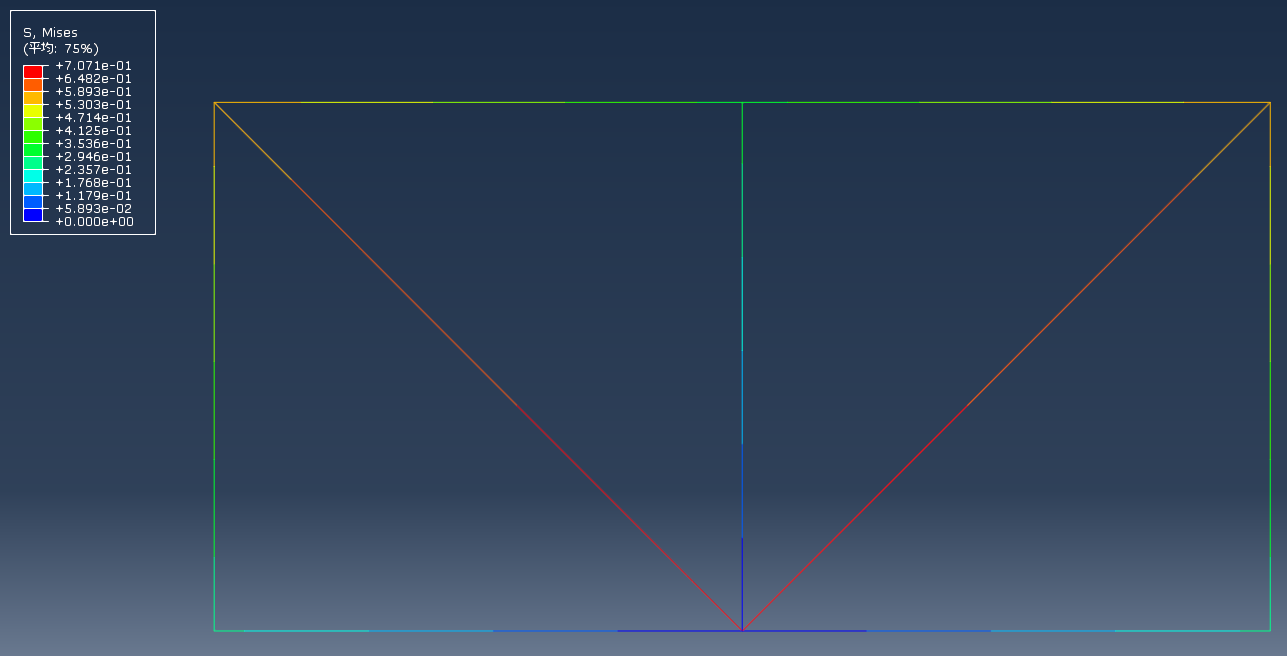


图 28 桁架单元桁架结构mises应力云图

取载荷施加点的y方向位移情况对比（实体单元：-1.1010E-10m；杆单元：-1.21E-10m），发现杆单元的载荷施加点位移绝对值比实体单元的值要稍大一些，这可能是由于桁架的基本假设中忽略了桁架在节点处的弯矩约束而产生的，即用桁架假设计算所得的位移结果将比真实情况更大，这也提高了设计的安全余量。

综上，我们认为理想桁架的基本假设是正确可行的。

## 自选结构分析

### 自选结构介绍

本次大作业选取实验室某振动模态实验结构作为自选结构进行特征值计算，该自选结构的外形及尺寸见图 29图 30，该结构的四根立柱材料为钢，三块平板材料为铝。



图 29 自选结构实物图

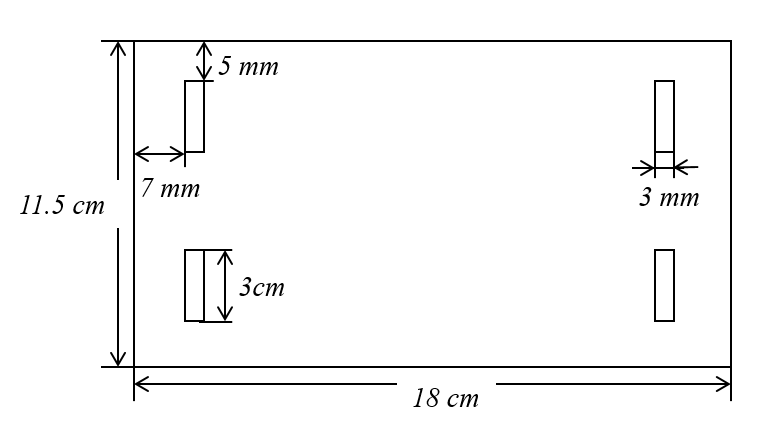
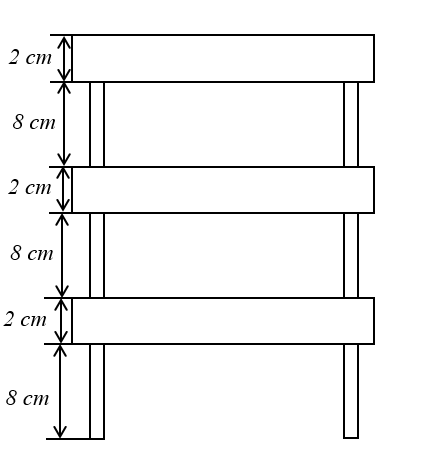
 

图 30 自选结构尺寸

### 自选结构计算结果

由于节点编号存在不合理，有些的网格划分可能会导致stap90计算出现问题，如图 31。最终选择合适的网格划分，可以得到前5阶特征值和特征向量，计算结果与实验结果对比见表 3‑1

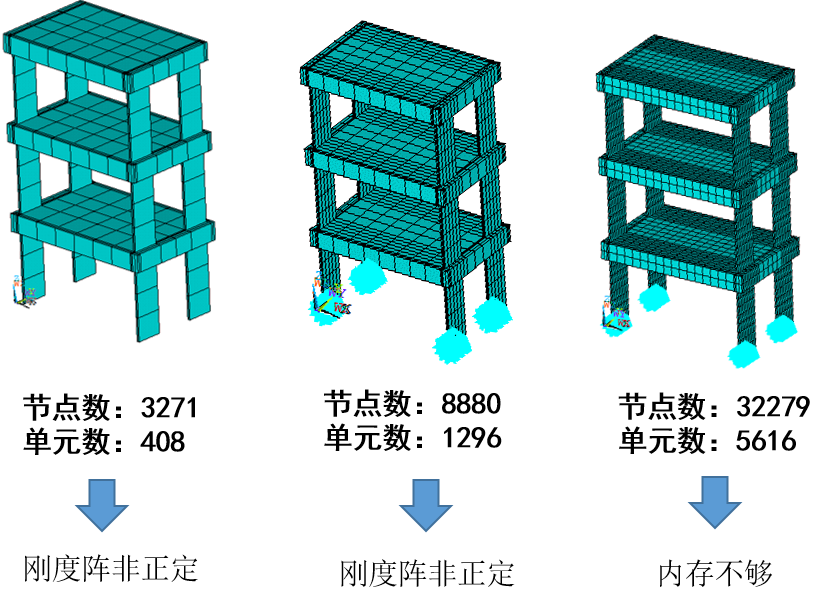


图 31 自选结构网格划分

表 3‑1 自选结构特征值计算结果与实验结果比较

|  |  |  |
| --- | --- | --- |
| 模态 | 计算结果（Hz） | 模态实验结果（Hz） |
| 1 | 16.479 | 14.0773 |
| 2 | 46.340 | 42.0137 |
| 3 | 67.317 | 63.8548 |
| 4 | 84.514 | - |
| 5 | 151.19 | - |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | |  |
|  | |  | |

图 32 自选结构特征值计算振型图

由表 3‑1可知，计算得到的特征值整体上大于实验测得结果。这是因为进行锤击实验时，结构的底部很难做到完全固支。除此之外，材料参数的选取也会造成一定误差。

# 总结与体会

## 总结与分工

在本次大作业中，我们完成了以下工作：

**文献调研模块：**

1： 调研子空间迭代法的改进方案 （陈楷东）

**程序改写模块：**

1： 增加一种高阶三维实体单元(李萌、陈家琦)

2： 增加特征值求解的功能(姚依晨、陈楷东)

3： 结果输出成后处理软件可以读取的形式(李萌、陈家琦)

**实际问题分析模块**

1：利用实体单元构建桁架的真实模型，通过对静力问题的计算验证桁架假设的合理性(李萌、陈家琦)

2：选择所在实验室的一个具体结构，讨论其动力学模型的建立，计算其频率和振型，详细讨论结果的正确性(姚依晨、陈楷东)

## 收获与体会

**陈楷东：**细节决定成败，引起失败的往往不是大错误而是小错误；基本概念和基本定义非常重要；不积跬步无以至千里，从简单结构出发测试算法和程序；不要“迷信权威”,实践是检验真理的唯一标准；抓紧时间，时间流逝飞快。

**李萌:**本次大作业是我第一次接触系统有限元程序的编写，更加深入理解了有限元的相关理论，对STAP90程序进行阅读、调试和修改是一次不错的体验，收获很大！

**姚依晨：**本次大作业，让我熟悉了fortran语言特点，增加了对stap90程序的了解，及对子空间迭代法的理解。同时非常感谢大作业期间组员和组长对我的安慰鼓励和帮助。

**陈家琦：**本次大作业让我有机会自己动手改编一个有限元的计算程序，重温了有限元的知识，让我对其计算过程更加了解。另外我们遇到的后处理、读写格式等问题，让我意识到多人协同工作时版本控制的重要性。

# 参考文献

[1] Klaus-Jürgen Bathe, Solution Methods for Large Generalized Eigenvalue Problems in Structural Engineering, Report UCSESM 71-20, Department of Civil Engineering, University of California, Berkeley, 1971.

[2] Klaus-Jürgen Bathe, Wilson E L. Solution methods for eigenvalue problems in structural mechanics [J]. International Journal for Numerical Methods in Engineering, 1973, 6(2):213-226.

[3] Lanczos C. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators [J]. Journal of research of the national bureau of standard, 1950, 45.

[4] 张雄, 王天舒, 刘岩. 计算动力学（第2版）[M]. 清华大学出版社, 2015.

[5] 宫玉才, 周洪伟, 陈璞, 等. 快速子空间迭代法、迭代Ritz向量法与迭代Lanczos法的比较[J]. 振动工程学报, 2005, 18(2):227-232.

[6] Bathe, Klaus-Jürgen. The subspace iteration method – Revisited [J]. Computers & Structures, 2013, 126:177-183.

[7] Bathe KJ. Finite Element Procedures [M]. Prentice Hall; 1996.

[8] Jung H J , Kim M C , Lee I W . An improved subspace iteration method with shifting[J]. Journal of Sound Vibration, 1999, 227(2):271-291.

[9] Zhao Q C , Chen P , Peng W B , et al. Accelerated subspace iteration with aggressive shift[J]. Computers & Structures, 2007, 85(19-20):1562-1578.

[10] Sherman A H . On the efficient solution of sparse systems of linear and nonlinear equations [J]. Journal of the American Chemical Society, 1975, 88(11):219-223.

[11] Chen P , Zheng D , Sun S , et al. High performance sparse static solver in finite element analyses with loop-unrolling[J]. Advances in Engineering Software, 2003, 34(4):203-215.

[12] Kim K T , Bathe, Klaus-Jürgen. The Bathe subspace iteration method enriched by turning vectors[J]. Computers & Structures, 2017, 186:11-21.

[13] 王勖成，有限单元法[M].清华大学出版社，2013.

[14] 卓家寿，弹性力学中的有限元法[M].高等教育出版社，1987.

[15] 秦太验，有限单元法[M].中国农业科学技术出版社，2006.

[16] 李俊峰，理论力学[M].清华大学出版社，2012