《计算动力学》
大作业报告

小组成员：王 若 溪　　舒 炫 博
冯　　伟　　韦 淞 瀚

二〇一七年六月

《计算动力学》大作业报告

目　录

第 1 章 引　言 1
1.1 针对改进特征值求解方法调研 1
1.2 STAP90 程序结构 3
1.3 本文主要内容 5
第 2 章 拓展 STAP90 程序 1
2.1 STAP90 程序的三维实体单元 1
2.1.1 8 节点 6 面体单元理论 1
2.1.2 8 节点 6 面体单元在 STAP90 程序中实现方法 5
2.1.3 8 节点 6 面体单元的验证算例 6
2.2 STAP90 程序的特征值求解功能 9
2.2.1 特征值求解算法设计 9
2.2.2 特征值求解算例验证 11
2.3 STAP90 程序的结果输出与可视化 12
第 3 章 涡轮增压机的模态分析 1
3.1 模型的简化分析 1
3.1.1 各个零件的简化 3
3.1.2 转子系统单元参数 4
3.2 模态仿真结果 5
3.2.1 特征频率与振型 5
3.2.2 临界频率和无阻尼特征频率图 7
3.3 结果的合理性分析 8
第 4 章 总　结 1
参考文献 1
附录 A ANSYS 命令流代码 1

摘　要

本文主要介绍了本小组在《计算动力学》大作业中完成的工作内容。本次大作业主要针对 STAP90 程序的拓展内容与有限元商业软件的对比展开，主要工作如下：

1. 分析了 STAP90 程序的结构，增加了 8 节点 6 面体单元，使其能够求解三维实体单元的静力学问题，并将结果与 ABAQUS 软件的结果进行对比；

2. 调研了多种特征值求解的方法，基于一种改进的特征值求解算法拓展了 STAP90 程序，使其能够求解特征值问题；

3. 分析了前后处理软件 Gmsh 的读入格式，修改了 STAP90 程序的输出文件格式，使其能够在 Gmsh 中可视化显示位移场和应力场结果；

4. 在 ANSYS 中分析了涡轮增压机结构的振型和频率，基于一定程度的简化模型，讨论了结果的合理性。

第1章 引　言

1.1 针对改进特征值求解方法调研

代数特征值问题是数值代数的一个重要研究领域，在许多科学和工程计算，如信号处理和控制、计算流体力学、机械和结构振动等问题中经常要求解矩阵特征值问题

$$Ax = \lambda x \tag{1.1}$$

或广义特征值问题

$$Ax = \lambda Bx \tag{1.2}$$

其中 A 和 B 为 n 阶实矩阵，(λ, x) 为特征对。

当矩阵阶数比较小时，可以利用矩阵特征值的相似不变性原理，通过一系列的变换，使之变成一系列容易求解的特征值的形式，如 Householder 方法，QR 方法等[1]。QR 方法是计算一般矩阵（中小型矩阵）全部特征值的一种有效方法，只适合求解阶数较低的矩阵。而实际应用中特征值问题往往是规模比较大的，并且只需要求解部分的特征值和特征向量，此时用变换法代价就会比较高。因此有必要研究合适的降阶方法把大型结构矩阵的计算降阶为小型的矩阵计算问题，其中投影法[1]就是一类重要的降阶方法。投影法的主要思想是，通过投影算子将原大规模矩阵 A 投影到适当低维子空间中，即将大规模问题转化为中、小型矩阵特征值问题，然后采用标准方法计算其特征对用来作为大型矩阵的特征对的近似。投影法的特点是仅使用矩阵 A 形成矩阵-向量积，不需对矩阵 A 本身做任何变换，从而可以减少内存需求，节省计算开支。由于投影法可采用压缩存储技术, 因而它适合求解大型矩阵特别是大型稀疏矩阵的特征值问题。目前常用的计算大型稀疏对称矩阵特征值问题的正交投影法有子空间迭代法[2]，Lanczos 方法[3]，Arnoldi 方法[4]，Davidson 方法[5]等。

幂法是计算矩阵特征值及其相应特征向量的基本方法之一。它是对一个初始向量进行迭代，推广到多个初始向量的情况就得到子空间迭代法，可同时求解几个极端特征值和相应的特征向量，但是在解决某些问题或在一些条件限制下收敛较慢，运算量较大，并且随着迭代次数的增加，对舍入误差的影响也较大。随后，经过大量的理论分析与数值实验，人们充分认识 Krylov 子空间方法是求解大型稀疏对称矩阵特征值问题有效的方法之一。当所要计算的矩阵阶数很大时，我们需要选择合适的整数 k，应用重新开始的 Krylov 子空间方法，由于收敛速度与初始向量的选取有关，因此有人专门对此进行了研究，提出利用 Chebyshev 多项式来加速重新开始 Krylov 子空间方法的一个技巧[6]。

另外，在求解的矩阵特征值有重特征值或者比较密集时，Krylov 子空间方法及其变形在计算中的可靠性与有效性就会下降。因此为了修正算法的可靠性与有效性，Underwood[7] 提出了求解对称矩阵特征值问题密集特征对的块 Lanczos 方法。

赵中华[8]等研究了计算大型稀疏对称矩阵的加速子空间迭代法的 Chebyshev 迭代法和预处理技术。既缩小了矩阵特征值的分布范围，又改善了每次循环的初始矩阵。加速了迭代法的收敛速度，减少了计算量和计算时间。

R. B. Lehoucq[9]提出了一种隐含重启的 Arnoldi 方法和子空间迭代的新方法。当在等尺寸的子空间上迭代时，隐式重新启动的方法可以以比同时迭代更快的收敛。

近些年,有些人对 Krylov 子空间法进行了改进。聂永明[10]提出了精化形式的不含逆的 Krylov 子空间方法，避免了应用预条件来加速收敛的冗余。提高了通常的 Krylov 子空间方法在计算特征向量时的稳定性。孟红燕[11]提出了带位移矩阵的块 Inverse-free Krylov 子空间方法并研究其正交基的形成、收敛性分析和算法的实现，可以有效地求解靠近给定位移附近的近似特征值。ROEL VAN BEEUMEN[12]提出了一种新的 Krylov 方法来解决非线性特征值问题。该方法通过未固定插值点和多项式阶数的 Hermite 插值近似目标矩阵。目前积极研究的领域是开发新的和更有效的预处理器，特别是针对特定类型的应用。

Inverse-free Krylov 子空间方法对于广义特征值问题

$$Ax = \lambda Bx \tag{1.3}$$

其中 A 和 B 为 n 阶实矩阵，求解的基本思想是把广义特征值问题转化为标准特征值问题

$$B^{-1}Ax = \lambda x \tag{1.4}$$

然后再利用计算标准的特征值问题的算法进行求解。经典的 Lanczos 算法、Arnoldi 算法是广泛应用的有效算法。针对大型矩阵密集特征值的计算，将广义特征值问题 $Ax = \lambda Bx$ 转化为带位移的逆变换（shift-invert）形式

$$\frac{1}{A-\sigma}x = \left(A-\sigma B\right)^{-1}Bx \tag{1.5}$$

应用 Krylov 子空间方法计算投影矩阵的特征对时，每一步迭代都需要解线性系统 $By = x$ 或者带位移的线性系统 $\left(A-\sigma B\right)y = Bx$。对于大型矩阵 $\left(A-\sigma B\right)^{-1}B$ 的计算已有不少的研究，对大规模矩阵进行分解的做法，如 QR 分解、LU 分解、Cholesky 分解等将变得不太适用。Inverse-free Krylov 子空间方法是一种求解广义对称特征值问题极端特征值的降阶方法，它应用 Rayleigh-Ritz 正交投影改善初始近似特征对，通过极小化 Rayleigh 商的思想，把包含特征信息的搜索子空间扩充为 Krylov 子空间

$$K_m = span\left\{x_k, (A-\rho_k B)x_k, \cdots, (A-\rho_k B)^{m-1}x_k\right\} \tag{1.6}$$

Inverse-free Krylov 子空间方法迭代地改善近似特征对，应用内外迭代计算 Rayleigh-Ritz 投影，避免了 Lanczos（Arnoldi）方法计算矩阵逆的问题。应用 Inverse-free Krylov 子空间方法，可以有效地计算广义对称特征值问题的极端特征值，并有很好的收敛性。

给定初始近似特征对(ρ_0, x_0)，在特定的子空间上应用 Rayleigh-Ritz 正交投影改善初始近似特征对，即极小化 Rayleigh 商

$$\rho(x) = \frac{x^T Ax}{x^T Bx} \tag{1.7}$$

Rayleigh 商的梯度

$$\Delta\rho(x) = \frac{(A-\rho(x)B)x}{x^T Bx} \tag{1.8}$$

最速下降法的思想在一些特征值问题求解中得到了很好的应用，并且一些算法的加速方法也是由此给出的。根据最速下降法的思想，选取 x_1 满足 $x_1 \in span\{x_0, r_0\}$ 且负梯度方向是最速下降的方向，因此可以看作是在子空间 $K_1 = span\{x_0, (A-\rho_0 B)x_0\}$ 上的 Rayleigh-Ritz 投影。以此类推，选取 x_k 满足 $x_k \in span\{x_{k-1}, r_{k-1}\}$ 可以看作是在子空间

$$K_m = span\left\{x_{k-1}, (A-\rho_{k-1}B)x_{k-1}, \cdots, x_{k-1}, (A-\rho_{k-1}B)^{m-1}x_{k-1}\right\} \tag{1.9}$$

上的 Rayleigh-Ritz 投影。通过迭代，在 Krylov 子空间 K_m 上投影求解广义特征值问题。这样利用 Rayleigh-Ritz 投影计算得到的最小特征值所在的投影子空间与 Krylov 子空间相同，因此所得到的解也是相同的。所以广义特征值问题转化为在 Krylov 子空间上利用标准的计算方法进行求解。

1.2 STAP90程序结构

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。原 STAP90 程序仅能针对一维拉伸梁单元的集中力载荷问题进行静力学求解，计算位移场、应变场和应力场。但容易对 STAP90 程序进行模块拓展，使其可以处理不同的梁、板、壳和实体单元的问题，也可以施加集中或分布的力和力矩载荷，并且可以处

理静力学、特征值和冲击动力学问题，对 STAP90 程序进行合理拓展，使之能够处理我们需要解决的问题，并与商业软件对比结果以验证拓展模块的正确性，是本次大作业中的一项重点。

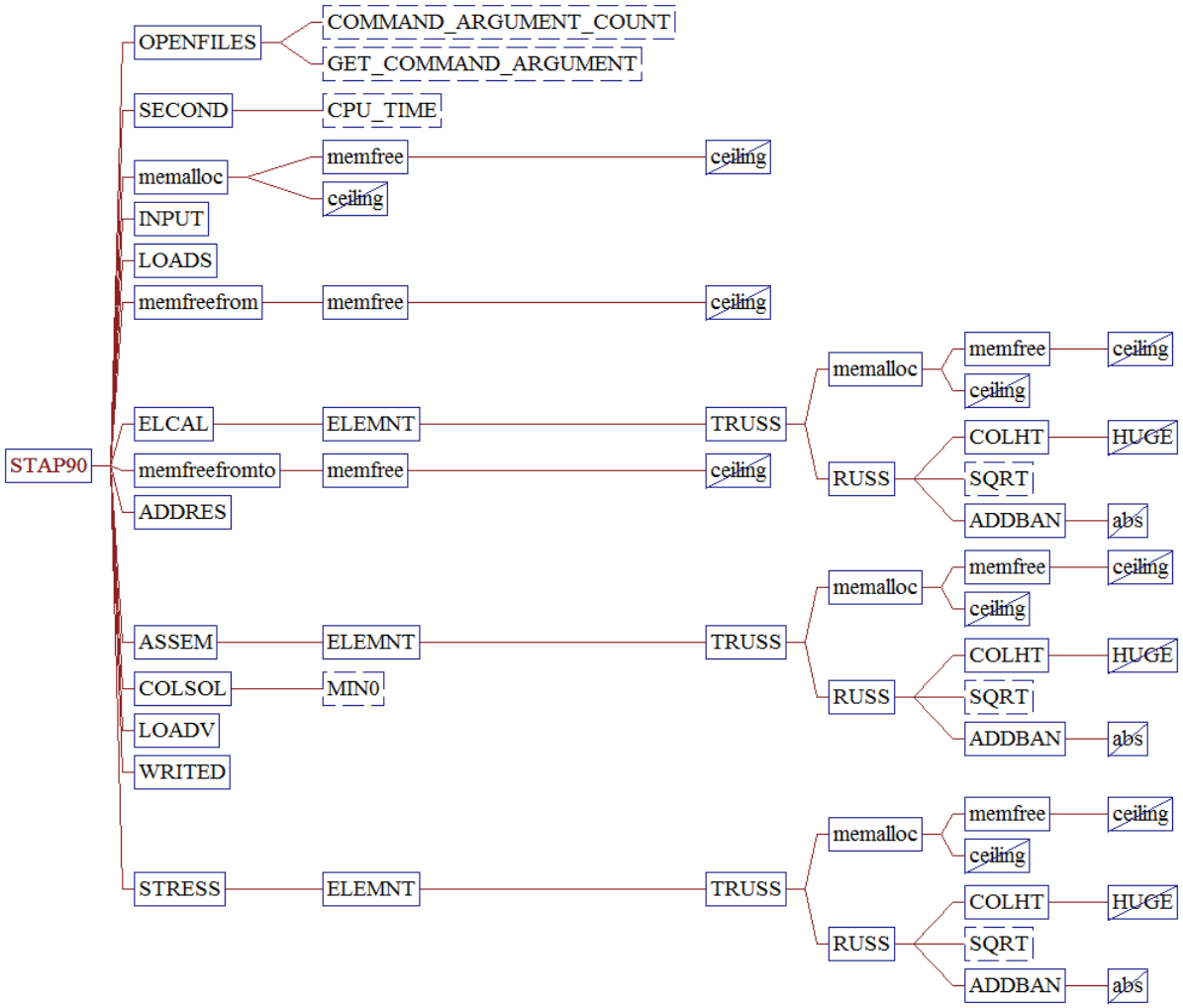


图 1.1 STAP90 程序的子程序调用关系

在此将简单介绍 STAP90 程序的主要结构，图 1.1 显示了其中子程序的调用关系。STAP90 主程序主要分为三个部分：文件读入部分、位移场求解部分和应变应力场计算部分。在这三部分中，使用变量 IND=1,2,3 指代正在运行的程序部分。在主程序 STAP90 中，首先调用子程序 MEMALLOC 为节点编号、节点坐标和载荷信息等变量分配内存。随后调用子程序 INPUT 读入全部节点信息、调用子程序 LOADS 读入全部载荷信息，调用子程序 ELCAL 读入全部单元拓扑和材料信息。其次调用子程序 ADDRES 计算一维变带宽储存总体刚度阵时需要的变量，如列高 MHT 和对角元编号 MAXA。然后调用子程序 ASSEM 组装模型的总体刚度阵 K，调用子程序 COLSOL 分

解总体刚度阵 K 并求解线性方程组$\mathbf{Kd}=\mathbf{f}$，得到总体的位移场。最后调用子程序 WRITED 输出位移场结果，以及调用子程序 STRESS 计算总体的应变应力场，并输出其结果。

其中在子程序 ELCAL、ASSEM 和 STRESS 中均需要调用子程序 ELEMNT 选择使用的单元，使用变量 NPAR(1)为单元分类，在原程序中仅有 NPAR(1)=1 时使用一维拉伸梁单元，拓展 STAP90 程序的单元类型时，可以添加相应的单元编号。在子程序 ELEMNT 中由变量 NPAR(1)选择相应单元后，调用相应单元的子程序，如一维拉伸梁单元子程序 TRUSS，在子程序 TRUSS 中分配单元信息变量的内存后调用具体计算子程序 RUSS，子程序 RUSS 包含多种功能，在主程序运行到不同部分时利用变量 IND 判断执行不同功能：1. IND=1 时，主程序 STAP90 通过子程序 ELCAL 调用子程序 TRUSS，此时子程序用于读入全部单元拓扑和材料信息，储存到变量；2. IND=2 时，主程序 STAP90 通过子程序 ASSEM 调用子程序 TRUSS，此时子程序用于计算各单元的单元刚度阵，并调用子程序 ADDBAN 将单元刚度阵组装总体刚度阵；3. IND=3 时，主程序 STAP90 通过子程序 STRESS 调用子程序 TRUSS，此时子程序用于计算单元的几何矩阵和弹性矩阵，并根据单元位移场计算单元的应变和应力场，同时输出结果。

拓展 STAP90 程序时，首先可以在子程序 ELEMNT 中添加新的单元编号和单元子程序，以实现 STAP90 程序处理各种不同单元的问题；其次可以在主程序 STAP90 中添加新的计算模式编号和相应的计算子程序，以实现 STAP90 程序求解特征值问题或动力学问题；最后可以对子程序 LOADS 或 STRESS 进行修改，以实现 STAP90 程序施加不同的载荷形式或采用不同的应力应变处理方式。

1.3 本文主要内容

本文的主要内容分为以下几个部分：

第一章是引言，综述了针对特征值问题求解的不同改进方法，并简介了 STAP90 程序的整体结构；

第二章陈述了小组对 STAP90 程序拓展的部分，包括 8 节点 6 面体实体单元和采用改进方法求解特征问题的算法设计和验证算例；

第三章分析了涡轮增压机的振型和频率，对模型进行简化后使用商业软件 ANSYS 建模，并讨论了简化结果的合理性和意义；

第四章总结了本次大作业的完成内容，以及小组内的分工合作情况。

第2章 拓展STAP90程序

2.1 STAP90程序的三维实体单元

本次大作业在 STAP90 程序中添加一种 8 节点 6 面体实体单元，本节将首先介绍该实体单元的描述方法、单元形函数、几何矩阵、弹性矩阵和单元刚度阵的计算方法，以及单元应力、应变场的计算理论；之后将介绍该三维实体单元在 STAP90 程序中的实现方法；最后给出一验证算例，并对比 STAP90 程序与商业软件 ABAQUS 对相同工况的计算结果。

2.1.1 8节点6面体单元理论

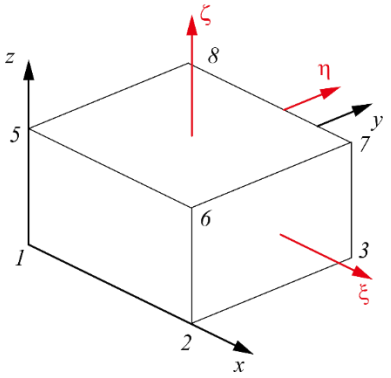


图 2.1 8 节点 6 面体单元示意图

如图 2.1 为任意一阶 6 面体单元，节点编号顺序如图，xyz 为全局坐标系，$\xi\eta\zeta$ 为单元局部坐标系，其转换关系为

$$\begin{cases} \xi = \dfrac{2}{a}\left(x - x_0\right) \\ \eta = \dfrac{2}{b}\left(y - y_0\right) \\ \zeta = \dfrac{2}{c}\left(z - z_0\right) \end{cases} \tag{2.1}$$

其中$\left(x_0, y_0, z_0\right)$为形心坐标，$a,b,c$ 为单元边长：

$$\begin{cases} x_0 = \dfrac{x_1 + x_2}{2} \\ y_0 = \dfrac{y_1 + y_4}{2} \\ z_0 = \dfrac{z_1 + z_5}{2} \end{cases} \tag{2.2}$$

$$\begin{aligned} a &= |x_1 - x_2| \\ b &= |y_1 - y_4| \\ c &= |z_1 - z_5| \end{aligned} \tag{2.3}$$

在单元的 8 个节点上，均有局部坐标$\xi,\eta,\zeta = \pm 1$。利用局部坐标可以构造该单元的形函数：

$$N_i(\xi,\eta,\zeta) = \frac{1}{8}(1+\xi_i\xi)(1+\eta_i\eta)(1+\zeta_i\zeta),(i=1,...,8) \tag{2.4}$$

形函数$N_i(\xi,\eta,\zeta)$满足特征：在任一单元节点 j 上有$N_i(\xi_j,\eta_j,\zeta_j) = \delta_{ij},(i,j=1,...,8)$。单元的位移插值函数可以写成：

$$\begin{cases} u = \sum_{i=1}^{8} N_i u_i \\ v = \sum_{i=1}^{8} N_i v_i \\ w = \sum_{i=1}^{8} N_i w_i \end{cases} \tag{2.5}$$

其中$(u_i, v_i.w_i)$为节点 i 的物理坐标。

由弹性力学相关理论，一阶单元的常应变场应为：

$$\begin{aligned}\boldsymbol{\varepsilon} &= \begin{bmatrix}\varepsilon_{xx} & \varepsilon_{yy} & \varepsilon_{zz} & \gamma_{xy} & \gamma_{yz} & \gamma_{zx}\end{bmatrix}^{\mathrm{T}} \\ &= \begin{bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial v}{\partial y} \\ \dfrac{\partial w}{\partial z} \\ \dfrac{\partial u}{\partial y}+\dfrac{\partial v}{\partial x} \\ \dfrac{\partial v}{\partial z}+\dfrac{\partial w}{\partial y} \\ \dfrac{\partial w}{\partial x}+\dfrac{\partial u}{\partial z} \end{bmatrix} = \mathbf{B}\mathbf{u}^{e}\end{aligned} \tag{2.6}$$

其中 $\mathbf{B}$ 为单元的几何矩阵，$\mathbf{u}^{e}$ 为单元的位移向量。计算可得：

$$\mathbf{B} = \begin{bmatrix}\mathbf{B}_1 & \cdots & \mathbf{B}_8\end{bmatrix}$$

$$\mathbf{B}_i = \begin{bmatrix} \dfrac{\partial N_i}{\partial x} & 0 & 0 \\ 0 & \dfrac{\partial N_i}{\partial y} & 0 \\ 0 & 0 & \dfrac{\partial N_i}{\partial z} \\ \dfrac{\partial N_i}{\partial y} & \dfrac{\partial N_i}{\partial x} & 0 \\ 0 & \dfrac{\partial N_i}{\partial z} & \dfrac{\partial N_i}{\partial y} \\ \dfrac{\partial N_i}{\partial z} & 0 & \dfrac{\partial N_i}{\partial x} \end{bmatrix}, (i = 1,\ldots,8) \tag{2.7}$$

$$\mathbf{u}^{e} = \begin{bmatrix}u_1 & v_1 & w_1 & \cdots & u_8 & v_8 & w_8\end{bmatrix}^{\mathrm{T}} \tag{2.8}$$

根据式(2.4)和(2.1)可得几何矩阵 $\mathbf{B}$ 中各个元素的值：

$$\begin{aligned} \frac{\partial N_i}{\partial x} &= \frac{\partial N_i}{\partial \xi}\frac{\partial \xi}{\partial x} = \frac{\xi_i}{4a}, (i = 1,\ldots,8) \\ \frac{\partial N_i}{\partial y} &= \frac{\partial N_i}{\partial \eta}\frac{\partial \eta}{\partial y} = \frac{\eta_i}{4b}, (i = 1,\ldots,8) \\ \frac{\partial N_i}{\partial z} &= \frac{\partial N_i}{\partial \zeta}\frac{\partial \zeta}{\partial z} = \frac{\zeta_i}{4c}, (i = 1,\ldots,8) \end{aligned} \tag{2.9}$$

对三维应力应变问题，应力场与应变场之间存在关系：

$$\boldsymbol{\sigma}=\begin{bmatrix}\sigma_{xx} & \sigma_{yy} & \sigma_{zz} & \tau_{xy} & \tau_{yz} & \tau_{zx}\end{bmatrix}^{\mathrm{T}}=\mathbf{D}\boldsymbol{\varepsilon} \tag{2.10}$$

其中 $\mathbf{D}$ 为单元弹性矩阵，定义为：

$$\mathbf{D}=\frac{E(1-\mu)}{(1+\mu)(1-2\mu)}\begin{bmatrix}1 & \frac{\mu}{1-\mu} & \frac{\mu}{1-\mu} & 0 & 0 & 0\\ \frac{\mu}{1-\mu} & 1 & \frac{\mu}{1-\mu} & 0 & 0 & 0\\ \frac{\mu}{1-\mu} & \frac{\mu}{1-\mu} & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & \frac{1-2\mu}{2(1-\mu)} & 0 & 0\\ 0 & 0 & 0 & 0 & \frac{1-2\mu}{2(1-\mu)} & 0\\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\mu}{2(1-\mu)}\end{bmatrix} \tag{2.11}$$

由单元几何矩阵和弹性矩阵可以计算单元刚度阵：

$$\mathbf{K}^{e}=\iiint_{V}\mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}dV=\begin{bmatrix}\mathbf{K}_{11} & \cdots & \mathbf{K}_{18}\\ \vdots & \ddots & \vdots\\ \mathbf{K}_{81} & \cdots & \mathbf{K}_{88}\end{bmatrix} \tag{2.12}$$

其中 $\mathbf{K}_{ij}\,(i,j=1,...,8)$ 为 3×3 的子矩阵，定义为：

$$\mathbf{K}_{ij}=\frac{EV}{16(1+\mu)(1-2\mu)}\begin{bmatrix}k_{11} & k_{12} & k_{13}\\ k_{21} & k_{22} & k_{23}\\ k_{31} & k_{32} & k_{33}\end{bmatrix},(i,j=1,...,8) \tag{2.13}$$

其中各元素为：

$$\begin{aligned}k_{11}\quad=&(1-\mu)\frac{\xi_i\xi_j}{a^2}\left(1+\frac{\eta_i\eta_j}{3}\right)\left(1+\frac{\zeta_i\zeta_j}{3}\right)\\ &+\frac{1-2\mu}{2}\left[\frac{\eta_i\eta_j}{b^2}\left(1+\frac{\zeta_i\zeta_j}{3}\right)\left(1+\frac{\xi_i\xi_j}{3}\right)+\frac{\zeta_i\zeta_j}{c^2}\left(1+\frac{\xi_i\xi_j}{3}\right)\left(1+\frac{\eta_i\eta_j}{3}\right)\right]\end{aligned} \tag{2.14}$$

$$\begin{aligned} k_{22} &= (1-\mu)\frac{\eta_i\eta_j}{b^2}\left(1+\frac{\zeta_i\zeta_j}{3}\right)\left(1+\frac{\xi_i\xi_j}{3}\right) \\ &+\frac{1-2\mu}{2}\left[\frac{\xi_i\xi_j}{a^2}\left(1+\frac{\eta_i\eta_j}{3}\right)\left(1+\frac{\zeta_i\zeta_j}{3}\right)+\frac{\zeta_i\zeta_j}{c^2}\left(1+\frac{\xi_i\xi_j}{3}\right)\left(1+\frac{\eta_i\eta_j}{3}\right)\right] \end{aligned} \tag{2.15}$$

$$\begin{aligned} k_{33} &= (1-\mu)\frac{\zeta_i\zeta_j}{a^2}\left(1+\frac{\xi_i\xi_j}{3}\right)\left(1+\frac{\eta_i\eta_j}{3}\right) \\ &+\frac{1-2\mu}{2}\left[\frac{\xi_i\xi_j}{a^2}\left(1+\frac{\eta_i\eta_j}{3}\right)\left(1+\frac{\zeta_i\zeta_j}{3}\right)+\frac{\eta_i\eta_j}{b^2}\left(1+\frac{\xi_i\xi_j}{3}\right)\left(1+\frac{\zeta_i\zeta_j}{3}\right)\right] \end{aligned} \tag{2.16}$$

$$k_{12}=\frac{1}{ab}\left(1+\frac{\zeta_i\zeta_j}{3}\right)\left(\mu\xi_i\eta_j+\frac{1-2\mu}{2}\eta_i\xi_j\right) \tag{2.17}$$

$$k_{13}=\frac{1}{ac}\left(1+\frac{\eta_i\eta_j}{3}\right)\left(\mu\xi_i\zeta_j+\frac{1-2\mu}{2}\zeta_i\xi_j\right) \tag{2.18}$$

$$k_{23}=\frac{1}{bc}\left(1+\frac{\xi_i\xi_j}{3}\right)\left(\mu\eta_i\zeta_j+\frac{1-2\mu}{2}\zeta_i\eta_j\right) \tag{2.19}$$

$$k_{21}=\frac{1}{ab}\left(1+\frac{\zeta_i\zeta_j}{3}\right)\left(\mu\eta_i\xi_j+\frac{1-2\mu}{2}\xi_i\eta_j\right) \tag{2.20}$$

$$k_{31}=\frac{1}{ac}\left(1+\frac{\eta_i\eta_j}{3}\right)\left(\mu\zeta_i\xi_j+\frac{1-2\mu}{2}\xi_i\zeta_j\right) \tag{2.21}$$

$$k_{32}=\frac{1}{bc}\left(1+\frac{\xi_i\xi_j}{3}\right)\left(\mu\zeta_i\eta_j+\frac{1-2\mu}{2}\eta_i\zeta_j\right) \tag{2.22}$$

此即为 8 节点 6 面体单元的单元刚度阵公式。

2.1.2 8节点6面体单元在STAP90程序中实现方法

为能够在 STAP90 程序中求解该 8 节点 6 面体单元，需要对 STAP90 程序在以下 4 个方面进行修改：

(1) 与主程序的接口：在 STAP90 主程序中通过在子程序 ELEMNT 中对单元类型编号变量 NPAR(1)进行判断，再调用相应的单元子程序。故首先需要在子程序 ELEMNT 中添加该实体单元的类型编号、新建该单元子程序，使读入单元类型编号符

合时调用该实体单元子程序。在本程序中定义单元类型编号 NPAR(1)=8 时，调用实体单元子程序 SOLID；

(2) 读入单元信息：在子程序 SOLID 中，初次调用应为单元信息变量分配内存，之后调用单元具体计算子程序 OLID。当主程序运行在文件读入部分时，即变量 IND=1 时，子程序 OLID 需要读入各单元的材料信息（包括杨氏模量 E、密度ρ和泊松比μ）、节点坐标（变量 XYZ(24, NPAR(2))）和自由度编号矩阵（变量 LM(24, NPAR(2))），以及调用子程序 ADDM 计算单元集中质量阵并组装成总体集中质量阵、调用子程序 COLHT 计算总体刚度阵的列高 MHT；

(3) 计算单元刚度阵：当主程序运行在位移场求解部分时，即变量 IND=2 时，子程序 OLID 需要根据式(2.12)~(2.22)计算各单元刚度阵，再调用子程序 ADDBAN 将单元刚度阵组装成总体刚度阵；

(4) 计算单元应变应力向量：当主程序运行在应力应变场计算部分时，即变量 IND=3 时，子程序 OLID 需要读入各单元的位移向量 $\mathbf{u}^e$，并根据式(2.7)和(2.11)计算单元的几何矩阵 $\mathbf{B}$ 和弹性矩阵 $\mathbf{D}$，根据式(2.6)和(2.10)计算并输出单元的应力向量 STR。

2.1.3 8节点6面体单元的验证算例

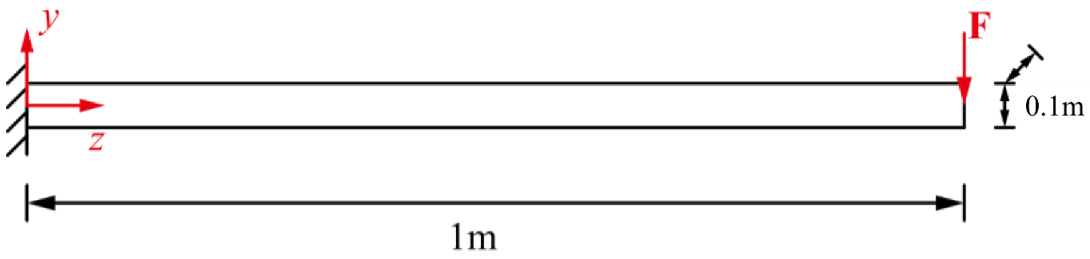


图 2.2 三维实体单元悬臂梁验证算例

为验证 STAP90 程序该 8 节点 6 面体单元的计算准确性，使用一悬臂梁模型，几何参数如图 2.2。材料参数为：杨氏模量 $E=2.1\times10^{11}Pa$、泊松比 $\nu=0.3$。边界约束条件为：一端固支、另一端自由，工况为在自由端施加的一集中恒力载荷 $F_y=-10N$，比较不同网格尺寸下 STAP90 程序与 ABAQUS 软件对该梁位移场和应力场的计算结果。验证中所使用的不同网格尺寸对应的模型节点数和单元数见表 2.1：

表 2.1 不同网格尺寸对应的模型节点数和单元数

网格尺寸(m)	节点数量	单元数量
0.1	44	10
0.05	189	80
0.025	1025	640
0.0125	6561	5120

对比不同网格尺寸下悬臂梁自由端的弯曲位移结果，见表 2.2：

表 2.2 不同网格尺寸自由端位移对比

网格尺寸(m)	STAP90 位移(m)	ABAQUS 位移(m)	相对误差
0.1	-1.23423e-6	-2.29853e-6	46.30%
0.05	-1.67051e-6	-1.87632e-6	10.97%
0.025	-1.84407e-6	-1.89955e-6	2.92%
0.0125	-1.90259e-6	-1.91896e-6	0.85%

从相对误差可以明显看出，当网格尺寸足够细密的时候，两者的相对误差在 1%以下，说明 STAP90 程序的计算结果足够准确。而当网格数较少时，STAP90 程序与 ABAQUS 软件的结果相差较多，这一方面因为网格数较少，有限元网格并未收敛，结果不具有可信性；另一方面因为使用的 ABAQUS 完全积分单元 C3D8 应用于这种弯曲问题时，往往呈现出过于刚硬的性质，使网格稀疏时 ABAQUS 的计算结果也并不准确。

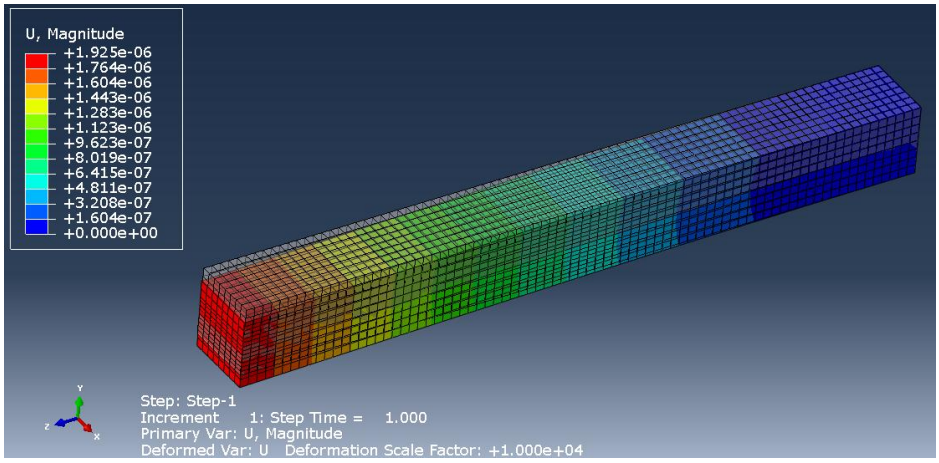


图 2.3 ABAQUS 软件计算的位移场结果

使用 0.0125m 的网格尺寸时，ABAQUS 软件的位移场结果如图 2.3，STAP90 程序与之沿梁伸长方向各节点的位移比较如图 2.4：

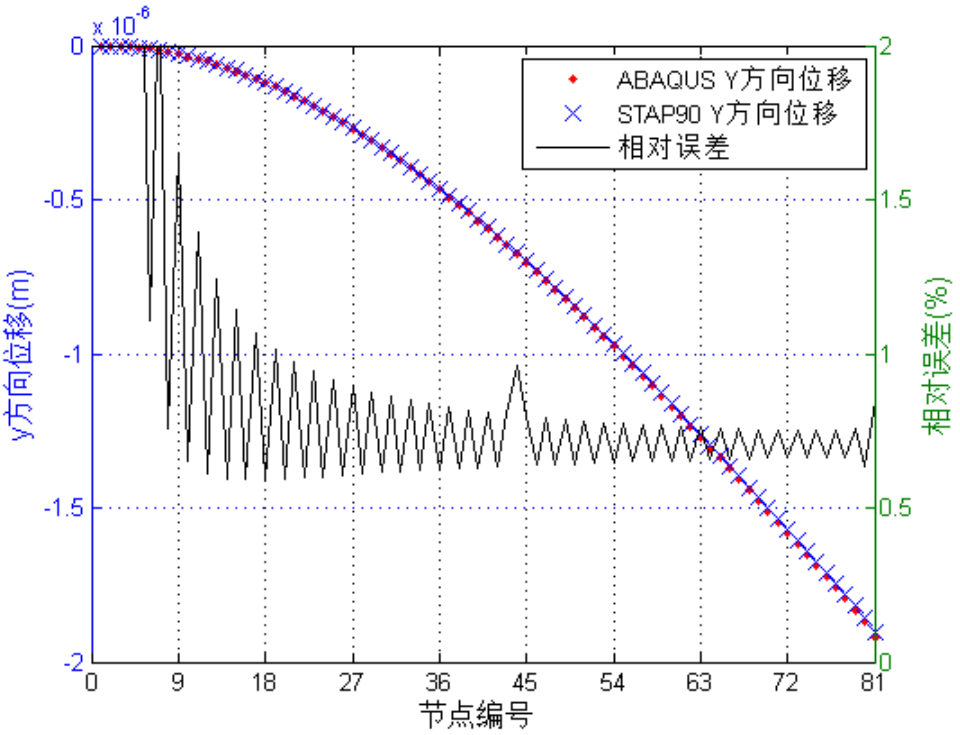


图 2.4 沿梁伸长方向的位移场比较

从图中可见，除固支端梁的挠度很小、相对误差计算不准确外，其余节点位移的相对误差均在 1%以下，结果比较准确。

表 2.3 不同网格尺寸相同单元应力场比较

网格尺寸	0.025m			0.0125m		
计算方式	STAP90 应力	ABAQUS 应力	相对误差	STAP90 应力	ABAQUS 应力	相对误差
σ_{xx}(Pa)	-9.82012e3	-9.59033e3	2.40%	-1.29343e4	-1.20026e4	7.76%
σ_{yy}(Pa)	-8.88462e3	-7.53697e3	17.88%	-1.20704e4	-1.06892e4	12.92%
σ_{zz}(Pa)	-4.40975e4	-4.57581e4	3.63%	-5.57374e4	-5.70461e4	2.29%
τ_{xy}(Pa)	-2.46096e1	-2.37014e1	3.83%	-1.76393e1	-1.67729e1	5.17%
τ_{yz}(Pa)	-1.39068e3	-1.72533e3	19.40%	-3.20817e3	-3.68407e3	12.92%
τ_{zx}(Pa)	-4.76574e3	-4.61416e3	3.29%	-8.52654e3	-8.64375e3	1.36%

取固支端的一个单元比较不同网格尺寸下的应力场结果，见表 2.3。应力场的相对误差均大于位移场的误差，这是因为应力场由位移场的结果计算得到，在一定程度上会放大相对误差。大部分应力分量的相对误差均在 3%左右，部分分量相对误差稍大，这是由于该单元与固支边界相邻，两者应力场计算结果有一定区别。在梁中部应力计算结果更加准确，相对误差会更小。比较 0.0125m 网格尺寸下沿梁伸长方向各单元的正应力结果，如图 2.5：

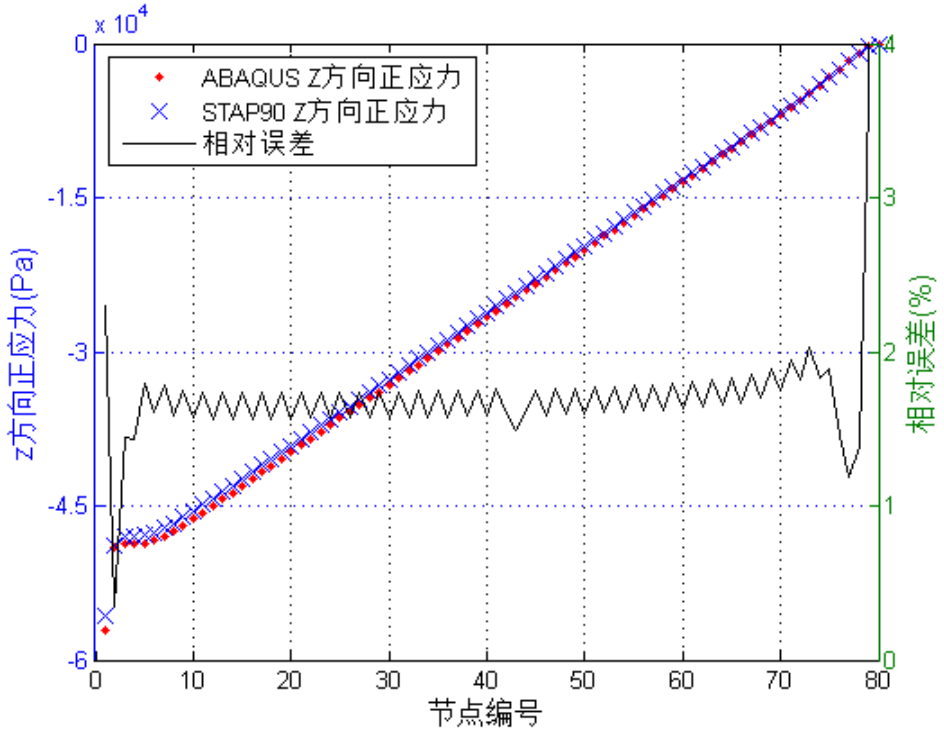


图 2.5 沿梁伸长方向各单元正应力比较

从图中可见，除梁自由端正应力很小、相对误差计算不准确，及固支端应力计算不准确外，其余单元的正应力相对误差均在 2%以下，可以认为 STAP90 程序的应力计算比较准确。

2.2 STAP90程序的特征值求解功能

2.2.1 特征值求解算法设计

根据文献调研的结果，采用 Inverse-free Krylov 子空间方法求解广义特征值问题，编写相应的求解程序扩展 STAP90 程序。

算法：Invrese-free Krylov 子空间方法

1. 选择初始向量 x_0 使得 $\|x_0\|=1$， $m\geq 1$

2. 计算 $\rho_0=\rho(x_0;A,B)$

3. 对 $k=0,1,2\cdots$，计算

（1） 子空间 $K_m\left(x_k,(A-\rho_k B)x_k\right)$ 的正交基 Z_m

（2） 投影矩阵 $A_m=Z_m^T(A-\rho_k B)Z_m$ 和 $B_m=Z_m^T B Z_m$

（3） 矩阵束 (A_m,B_m) 的最小特征值对 (θ,u)

（4） $\rho_{k+1}=\rho_k+\theta, x_{k+1}=Z_m u$

算法说明：

（1）在 Inverse-free Krylov 子空间方法中，利用标准的 Lanczos 方法或者修正的加权 Arnoldi 方法形成带位移的矩阵 $(A-\rho_k B)$ 构造的 Krylov 子空间 $K_m\left(x_k,(A-\rho_k B)x_k\right)$ 的一组正交基，比直接形成 Krylov 子空间 $K_m\left(B^{-1}A,x_k\right)$ 的一组正交基避免了计算 B^{-1} 增加了数值稳定性，并减少了计算复杂性。

（2）利用逆迭代法计算矩阵束(A, B)的最小特征值。

标准正交基的形成：

利用标准特征值问题的 Lanczos 方法或者修正的加权 Arnoldi 方法，形成算法中的正交基 Z_m，将矩阵束 (A,B) 投影、降阶为 (A_m,B_m)。计算矩阵束 (A_m,B_m) 的特征对，形成内外二层迭代，从而获得矩阵束 (A,B) 的近似特征对：

$$C_k=A-\rho_k B \tag{2.23}$$

Lanczos 过程：

1. 选择初始向量 x_k， $z_0=\dfrac{x_k}{\|x_k\|_2}$，令 $\beta_0=0$， $z_{-1}=0$；

2. 对 $i=0,1,2,\cdots,m-1$：

$$
\begin{aligned}
&w = C_k z_i \\
&w = w - \beta_i z_{i-1} \\
&\alpha_i = z_i^T w \\
&\hat{w} = w - \alpha_i z_i \\
&\beta_{i+1} = \left\| \hat{w} \right\|_2 , \\
&z_{i+1} = \frac{\hat{w}}{\beta_{i+1}}
\end{aligned}
\tag{2.24}
$$

3. 形成正交基 $Z_m = [z_1, z_2, \cdots, z_m]$

2.2.2 特征值求解算例验证

1. 算例一：

条件：正方形板的边长为 100，厚度为 1，弹性模量 2E5，泊松比为 0.3。下方简支。最上面边的中点作用一个向下的值为 1 的荷载。

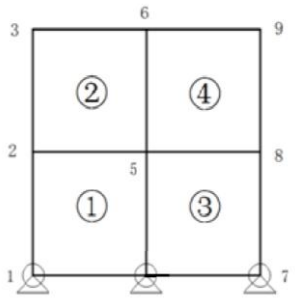


图 2.6 特征值求解的算例一示意图

ANSYS 计算得到的基频为： 0.5115Hz

利用 STAP90 程序所得到的基频为： 0.4698Hz

将 STAP90 程序中的总体刚度矩阵和质量矩阵在 MATLAB 中进行计算，求得频率为： 0.4698Hz

ANSYS 和 STAP90 程序所得到的基频误差：8.16%

2. 算例二：

条件：平面应力板的边长 20*10，厚度为 1，密度为 1，弹性模量 2E5 ，泊松比为 0.3。左边固定，单元尺寸为 1*1，有 200 个单元，231 个节点，440 个自由度。

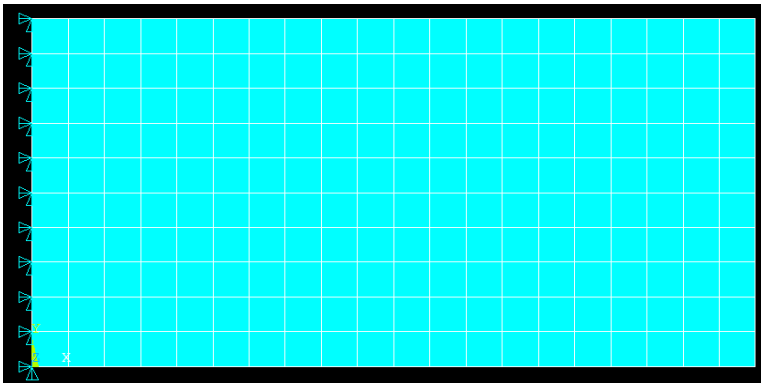


图 2.7 特征值求解的算例二示意图

ANSYS 计算结果： 1.5533Hz

STAP90 程序计算结果： 1.6188Hz

ANSYS 和 STAP90 程序所得到的基频误差：4.0%

结果分析：通过 STAP90 计算得到的特征值与 MATLAB 根据导出的刚度、质量矩阵求得的特征值在误差允许范围内相等，可以看出 Krylov 子空间法求解特征值的正确性。同时加快了收敛速度。经过与 ANSYS 建模计算结果进行对比，基频结果有误差，我们认为是由于质量阵构造的不同所导致，STAP90 使用的是集中质量阵。随着网格的加密，计算结果越来越接近。

2.3 STAP90程序的结果输出与可视化

我们使用 Gmsh 软件进行后处理。Gmsh 可以读入 MSH ASCII 格式的有限元结果文件，并且用一些可以自定义的方式显示与节点或单元关联的标量或向量。MSH ASCII 格式如下：

```
$MeshFormat
version-number file-type data-size
$EndMeshFormat
$PhysicalNames
number-of-names
physical-dimension physical-number "physical-name"
…
$EndPhysicalNames
$Nodes
number-of-nodes
node-number x-coord y-coord z-coord
…
$EndNodes
```

```
$Elements
number-of-elements
elm-number elm-type number-of-tags < tag > … node-number-list
…
$EndElements
$Periodic
number-of-periodic-entities
dimension slave-entity-tag master-entity-tag
number-of-nodes
slave-node-number master-node-number
…
$EndPeriodic
$NodeData
number-of-string-tags
< "string-tag" >
…
number-of-real-tags
< real-tag >
…
number-of-integer-tags
< integer-tag >
…
node-number value …
…
$EndNodeData
$ElementData
number-of-string-tags
< "string-tag" >
…
number-of-real-tags
< real-tag >
…
number-of-integer-tags
< integer-tag >
…
elm-number value …
…
$EndElementData
$ElementNodeData
number-of-string-tags
< "string-tag" >
```

```
…
number-of-real-tags
< real-tag >
…
number-of-integer-tags
< integer-tag >
…
elm-number number-of-nodes-per-element value …
…
$EndElementNodeData
$InterpolationScheme
"name"
number-of-element-topologies
elm-topology
number-of-interpolation-matrices
num-rows num-columns value …
…
$EndInterpolationScheme
```

其中我们用到的字段有：

- version-number：文件格式版本，一定是 2.2
- file-type：文件格式是 ASCII 还是二进制。这里是 ASCII，所以一定是 0
- data-size：浮点数所占字节数。目前只支持 sizeof(double)=8
- number-of-nodes：节点数目
- node-number：节点编号，只能是正整数
- x-coord y-coord z-coord：XYZ 坐标
- number-of-elements：单元数目
- elm-number：单元编号，只能是正整数
- elm-tpe：单元种类。1 是二节点线单元，5 是六面体实体单元
- number-of-tags：本单元有多少个 tag。默认第一个 tag 是单元所属的物理实体编号，第二个 tag 是单元所属“基本几何体”的编号（不懂是什么意思，下面的示例文件中两个编号分别是 99 和 2，就一直在用这两个数），第三个 tag 是单元所属的网格分区数目，之后的 tag 是这些分区的编号。Gmsh 要求至少要有前面两个 tag。
- node-number-list：本单元的节点列表。其中六面体单元的节点编号如下图 2.8，其中 u，v，w 为单元本地坐标：

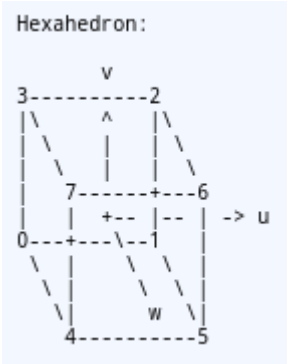


图 2.8 六面体单元本地坐标

- number-of-string-tags：字符串 tag 的数目。第一个 tag 为数据的名称，第二个是所使用的插值方式
- number-of-real-tags：实数 tag 的数目。第一个 tag 是数据对应的时刻
- number-of-integer-tags：整数 tag 的数目。第一个 tag 是时间步的序号，第二个是数据的维数：标量是 1，向量是 3，张量是 9，第三个是数据中节点或单元的数目，第四个是数据所属的“分区编号”，不懂是什么意思，设为 0（不分区）
- value：实数数据

下面是书中习题 3-1(a)的结果文件，实际文件中不能包含注释：

```
$MeshFormat
2.2 0 8
$EndMeshFormat
$Nodes
6      共有 6 个节点
1      -20.000       10.000        0.000
2        0.000       10.000        0.000
3       20.000       10.000        0.000
4      -20.000        0.000        0.000
5        0.000        0.000        0.000
6       20.000        0.000        0.000
$EndNodes
$Elements
11      共有 11 个单元
1 1 2 99 2 1 2   第一个单元，类型为 1（二节点直线），有两个 tag，第一个是 99，第二个是 2，包含 1 号和 2 号节点
2 1 2 99 2 2 3
3 1 2 99 2 1 5
4 1 2 99 2 2 4
5 1 2 99 2 2 6
```

```
6 1 2 99 2 3 5
7 1 2 99 2 1 4
8 1 2 99 2 2 5
9 1 2 99 2 3 6
10 1 2 99 2 4 5
11 1 2 99 2 5 6
$EndElements
$NodeData
1
"NodalLoad1"
1
0.0
3
0
3
1
2 0 -0.10000E+01 0
$EndNodeData
$NodeData
1       1 个字符串 tag
"NodeDisplacement1"      数据名称：载荷工况 1 下的节点位移
1       1 个浮点数 tag
0.0     时刻为 0（静态分析）
3       3 个整数 tag
0       第 0 个时间步（时间步序号从 0 开始）
3       每个节点上的数据是 3 维向量
6       有 6 个节点数据
1     0.335135E-04    -0.837838E-05     0.000000E+00
2     0.169407E-20    -0.185835E-03     0.000000E+00
3    -0.335135E-04    -0.837838E-05     0.000000E+00
4     0.000000E+00     0.000000E+00     0.000000E+00
5     0.000000E+00    -0.169079E-03     0.000000E+00
6     0.000000E+00     0.000000E+00     0.000000E+00
$EndNodeData
$ElementData
1               1 个字符串 tag
"ElementStress1"  数据名称：载荷工况 1 下的单元应力
1               1 个浮点数 tag
0.0             时刻为 0（静态分析）
3               3 个整数 tag
0               第 0 个时间步（时间步序号从 0 开始）
```

```
1                 每个节点上是 1 维标量（杆内拉压应力）
11                共有 11 个单元
1 -0.335135E+00
2 -0.335135E+00
3  0.374693E+00
4 -0.743341E+00
5 -0.743341E+00
6  0.374693E+00
7 -0.167568E+00
8 -0.335135E+00
9 -0.167568E+00
10  0.000000E+00
11  0.000000E+00
$EndElementData
```

用 Gmsh 软件打开上述结果文件，调整数据显示方式后得到如下界面：

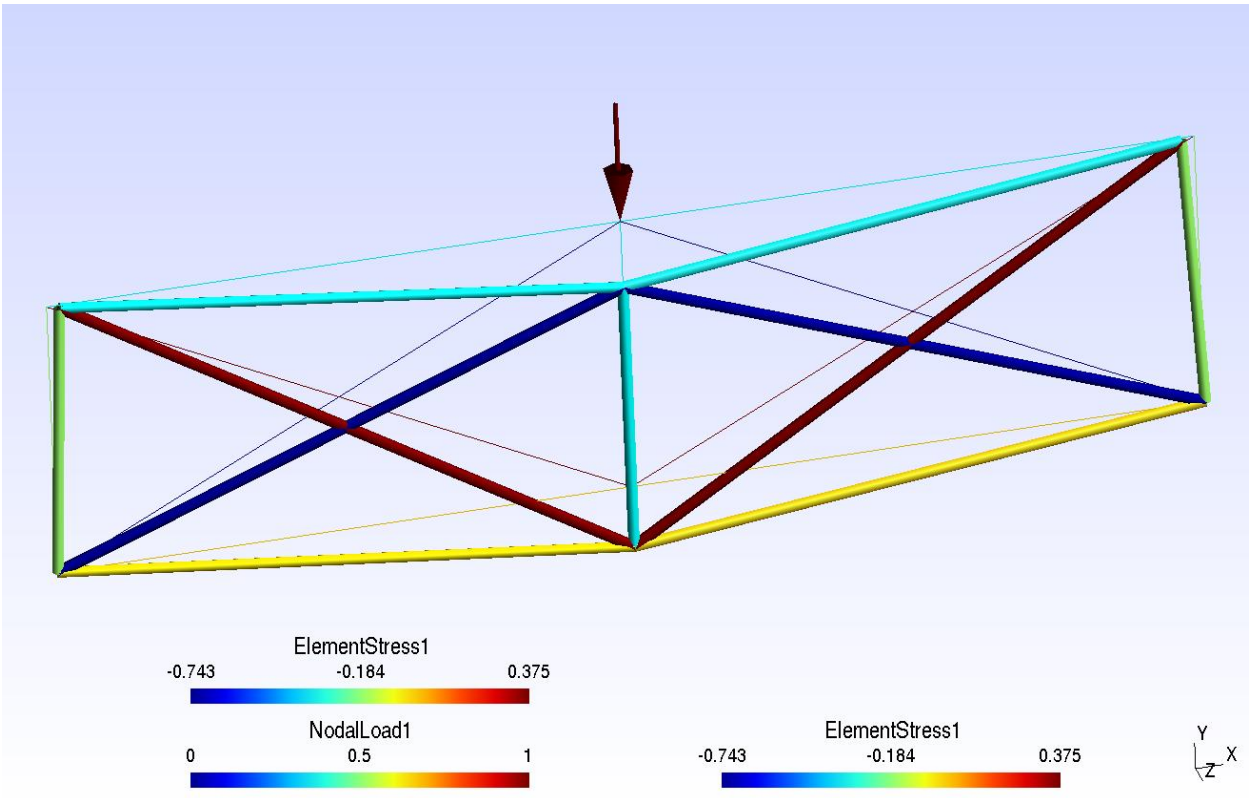


图 2.9 Gmsh 后处理显示界面

那么我们是怎样修改 STAP90 得到 MSH 格式的结果文件呢？我们修改了原有的 WRITE 函数调用，使其输出的数据格式符合上述要求。但是 STAP90 输出数据的顺序

不符合上述要求：$Elements 片段应在$NodeData 片段之前，而 STAP90 输出的顺序正好相反。所以我们用下面的 Vim 命令将这两段颠倒一下：

```
execute "normal!
/$Elements\<enter>d/$NodeData\<enter>?$EndNodes\<enter>p:wq!
STAP90.msh\<enter>"
```

对上述命令解释如下：

- execute “normal! …”是指在 Vim 的 Normal 模式下依次输入后续命令。所以这行命令相当于在 Vim 中在 Normal 模式下依次按下以下按键：
  /$Elements<回车>d/$NodeData<回车>?$EndNodes<回车>p:wq!
  STAP90.msh<回车>
- “/”是“向下查找”，“/$Elements”是向下查找“$Elements”这个字符串。紧随其后的“\<enter>”是指按下回车。此时光标处于“$Elements”的“$”处（下表黄色）：

```
…       （节点信息）
$EndNodes
$NodeData
…       （载荷数据）
$EndNodeData
$Elements
…       （单元信息）
$EndElements
$NodeData
…
```

- “d”是“剪切”。“d”之后应有一个移动光标的命令，这样 Vim 就会从光标当前位置剪切到移动到的位置之前一个字符。在这里，“d”之后的“/$NodeData\<enter>”即为将光标移至“$NodeData”的第一个字符处（上表红色），在这之前是“$EndElements”。所以至此“$Elements”到“$EndElements”这一段就被剪切了
- “?”是“向上查找”，“?$EndNodes\<enter>”将光标置于“$EndNodes”这一行
- “p”是“粘贴”，它将剪切过的内容粘贴在当前行之后。至此文件符合 MSH 格式要求
- “:wq! STAP90.msh\<enter>”是将文件保存为 STAP90.msh 并强制退出

编写 Bash 脚本 postprocessing.sh 运行 Vim 打开 STAP90.OUT 文件并执行上面这条命令，脚本内容如下：

```
#!/bin/bash
```

```
vim -c 'execute "normal!
/$Elements\<enter>d/$NodeData\<enter>?$EndNodes\<enter>p:wq!
STAP90.msh\<enter>"' STAP90.OUT
```

其中调用 vim 使用的-c 参数是指打开文件后运行单引号中的命令。

用同样的方法绘制 SOLID 单元组成的悬臂梁算例结果如下图 2.10：

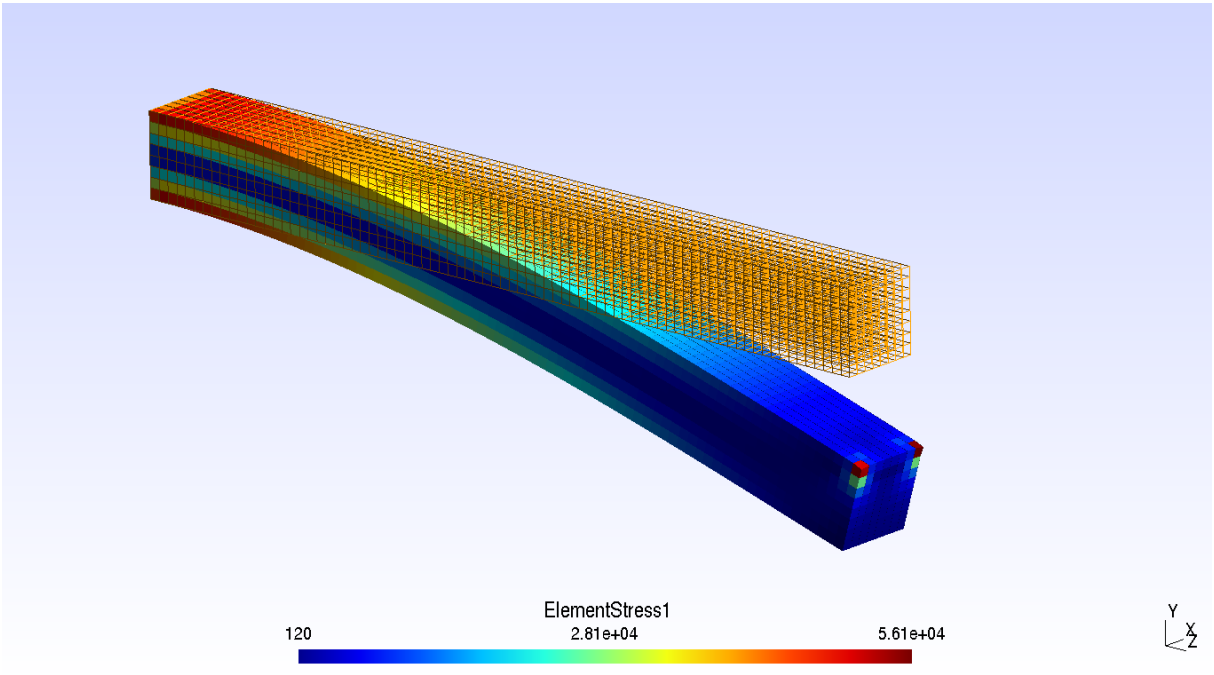


图 2.10 Gmsh 后处理显示悬臂梁应力场

第3章 涡轮增压机的模态分析

本文选择的常用设备是涡轮增压机。涡轮增压机有可能在汽车发动机或者航空发动机中看到，可以说是非常常用了。图 3.1 是一个涡轮增压机的结构原理示意图，它是利用发动机排出的废气惯性冲力来推动涡轮室内的涡轮，涡轮又带动同轴的叶轮，叶轮压送由空气滤清器管道送来的空气，使之增压进入气缸，实现气体增压。

图 3.1 涡轮增压机结构原理

3.1 模型的简化分析

由于涡轮增压机的整体结构比较复杂，而且整体结构中的很多部分是通气的管路，其受到的气体的相互作用力不大，固在分析中将其视为近似刚体的结构。增压机中主要的运转的容易变形的机构是中间的转子机构，所以模态分析中主要研究的是转子系统。

图 3.2 是某涡轮增压机的转子装配图，其中转子构件和其材料、质量等数据见表 3.1 所示，转子的运转速度最高考虑到 30000r/min。

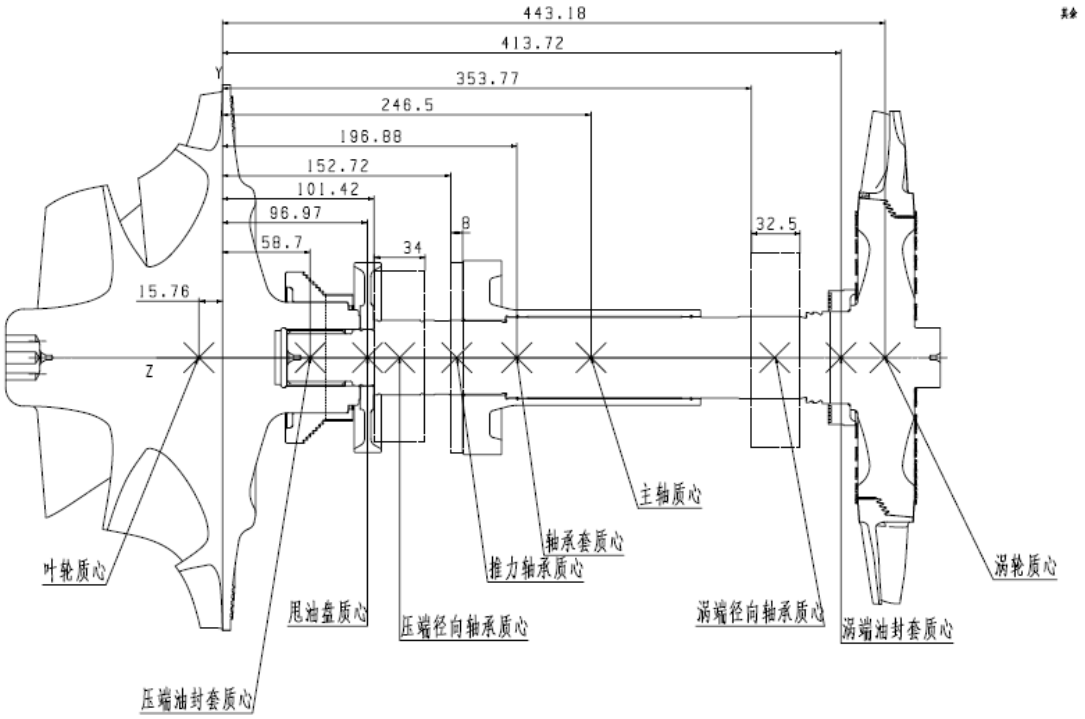


图 3.2 某涡轮增压机的转子装配图

表 3.1 转子转配图上零件信息

序号	名称	材料	密度 Kg/m^3	质量 kg	弹性模量	转动惯量
1	导风轮	2618	2800	2.881	7.45e10	0.00903
2	叶轮	2618	2800	7.660	7.45e10	0.077419
3	叶轮座	TC4	4500	13.702	1.1e11	0.184768
4	涡轮	GH4169	8240	12.81	2e11	0.09496
5	压端油封套	42CrMo	7830	1.113	2.12e11	0.0021528
6	推力轴承套	38 CrMoAlA	7830	2.856	2.13e11	0.00458
7	甩油盘	42CrMo	7830	0.6732	2.12e11	0.001241
8	涡端油封套	42CrMo	7830	0.3874	2.12e11	0.0005666
9	主轴	42CrMo	7830	5.76	2.12e11	0.056875
10	压端径向轴承	QSn6.5-0.1	8650	0.431	121.6	
11	涡端径向轴承	QSn6.5-0.1	8650	0.418	121.6	
12	推力轴承	G20CrNiMoA	7850	0.65	206	

转子的简化一般是简化成盘和轴构成的系统，轴用 Timoshenko 梁来仿真，盘用集中质量模型来仿真，轴承简化成弹簧单元。

3.1.1 各个零件的简化

1）叶轮的简化

如图 3.2 所示，叶轮不只是叶片，还包括叶片中间的轴，而且叶轮轴的材料与叶轮、主轴都不一样。一般叶轮这类不规则的旋转体有两种简化方法，一种是直接简化成集中质量附加在轴上，另一种是简化成额外的轴段附加在轴上。图 3.2 中叶轮轴与主轴通过螺纹的方式连接，但其实应该是紧连接使叶轮轴和主轴在接触面上完全固定。所以本文把叶轮分成不同部分，一个是中间有完整圆柱体的部分，此部分简化成梁单元，另一部分是完整圆柱体之外的不规整的部分，此部分分成几个集中质量加在轴段上。

2）推力轴承套的简化

推力轴承套是一个比较长的零件附在盘上，看起来应该简化成附加的轴段。但是分析此轴承套的设计意图，因为其受轴向推力，应该是用过与轴颈的配合来固定，而其他地方应该设计成间隙配合来方便安装，所以推力轴承套应该简化成集中质量加在轴颈处。

3）其他零件的简化

轴上其他零件都是比较典型盘型零件，都可以之间简化成盘单元，在有限元模型中当成集中质量来看待。最后可以将轴简化成图 3.3 的形式。

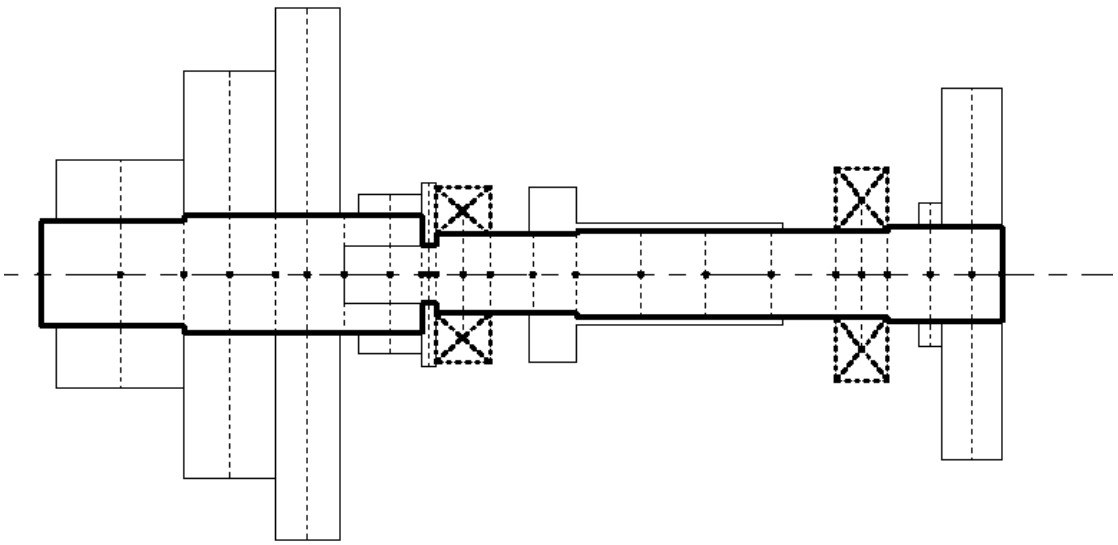


图 3.3 简化后的转子有限元模型

3.1.2 转子系统单元参数

如图 3.3 所示，使用 ansys apdl 来进行仿真，整个轴上分布有 24 个节点。用 beam188 单元来构成轴，共有 25 个 188 单元；用 mass21 单元来表示盘，共 8 个单元。此外，在节点 12 和节点 20 处还连接有若干弹簧单元来表示轴承，两个轴承的刚度认为是一样的，由于轴承刚度未知，取刚度分别为 10^6 到 10^9，阻尼为 50。所得到的单元参数如下：

表 3.2 各轴段单元 beam188

单元编号	节点	长度/mm	截面尺寸（外径）/mm	材料
1	1,2	50	66	TC4
2	2,3	40	66	TC4
3	3,4	29	74	TC4
4	4,5	29	74	TC4
5	5,6	20	74	TC4
6	6,7	23	74	TC4
7	7,8	29	外径 74，内径 36	TC4
8	8,9	20	外径 74，内径 36	TC4
9	7,8	29	36	42CrMo
10	8,9	20	36	42CrMo
11	9,10	4.5	36	42CrMo
12	10,11	4.5	36	42CrMo
13	11,12	17	50	42CrMo
14	12,13	17	50	42CrMo
15	13,14	27	50	42CrMo
16	14,15	27	50	42CrMo
17	15,16	41	54	42CrMo
18	16,17	41	54	42CrMo
19	17,18	41	54	42CrMo
20	18,19	41	54	42CrMo
21	19,20	16.25	54	42CrMo

22	20,21	16.25	54	42CrMo
23	21,22	27	60	42CrMo
24	22,23	26	60	42CrMo
25	23,24	19	60	42CrMo

表 3.3 集中质量单元 mass21

质量块编号	节点位置	质量/kg	极转动惯量/kg・m^2	径转动惯量/kg・m^2
1	2	2.881	0.0090385	0.0060562
2	4	7.660	0.067999	0.036147
3	6	13.702	0.184769	0.0942113
4	8	1.113	0.0021528	0.0012448
5	10	0.673	0.0012410	0.0006250
6	15	2.856	0.0045865	0.0077487
7	22	0.3874	0.0005666	0.0002896
8	24	12.811	0.949599	0.049021

表 3.4 轴承单元 combine14

轴承编号	单元位置	径向刚度	径向阻尼
1	12	106 到 109	50
2	20	106 到 109	50

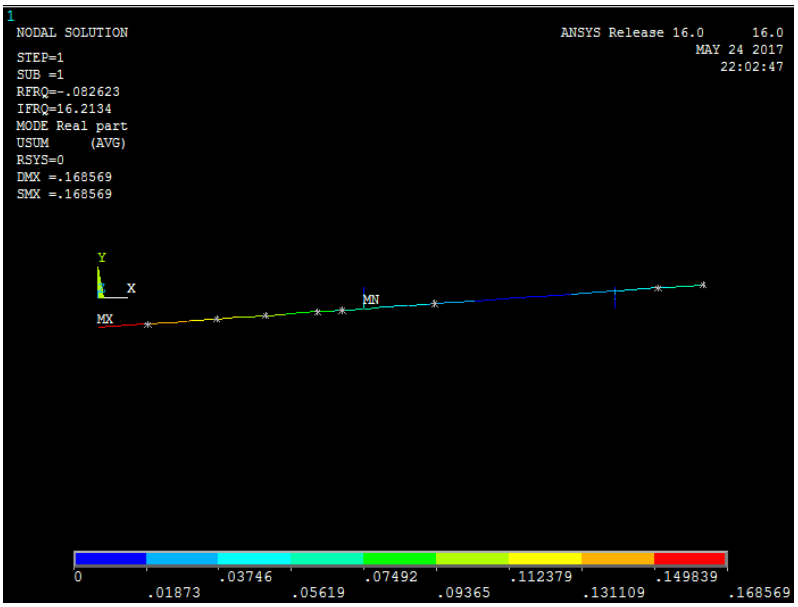
3.2 模态仿真结果

3.2.1 特征频率与振型

将单元信息输入 apdl 进行仿真，在轴承刚度为 10^6N/m 时的模态分析结果如下：

SET	TIME/FREQ(Damped)		TIME/FREQ(Undamped)	LOAD STEP	SUBSTEP	CUMULATIVE
1	-0.82623E-01	16.213	j 16.217	1	1	1
2	-0.82652E-01	16.219	j 16.217	1	2	2
3	-0.36616	34.120	j 34.123	1	3	3
4	-0.36619	34.122	j 34.123	1	4	4
5	-0.58300E-01	136.23	j 136.23	1	5	5
6	-0.42215	234.57	j 234.59	1	6	6
7	-0.42223	234.61	j 234.59	1	7	7

可以得到前两阶的无阻尼固有频率为 16.217hz 和 34.123hz。它们对应的振型如图 3.4 所示。



(a)一阶弯曲振型图

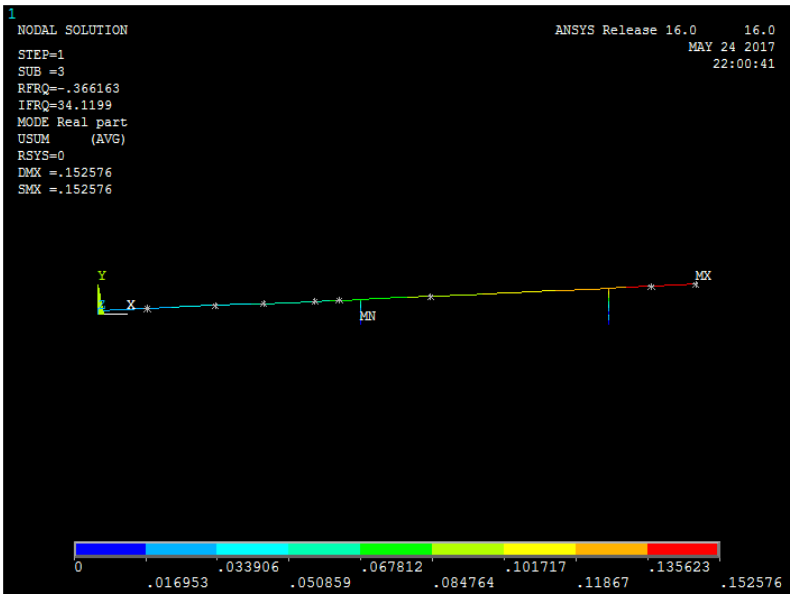


图 3.4 (b)二阶弯曲振型图

从振型图中可以看出第一阶模态的变形量最大在左侧的叶轮部分，第二阶模态的变形量最大在右侧的涡轮的部分。

3.2.2 临界频率和无阻尼特征频率图

对应转子系统，其临界转速是非常重要的部分，所以在这里还要对其临界转速进行分析。由于系统的阻尼振动频率会跟转速有关，所以一般通过坎贝尔图来分析临界转速。图 3.5 为转子的坎贝尔图，可以观察图中 F=1x spin 线与其他线的交点即为临界转速。由图中可以观察到一阶正向涡动的临界转速约为 800r/min，二阶正向涡动的临界转速约为 2400r/min。

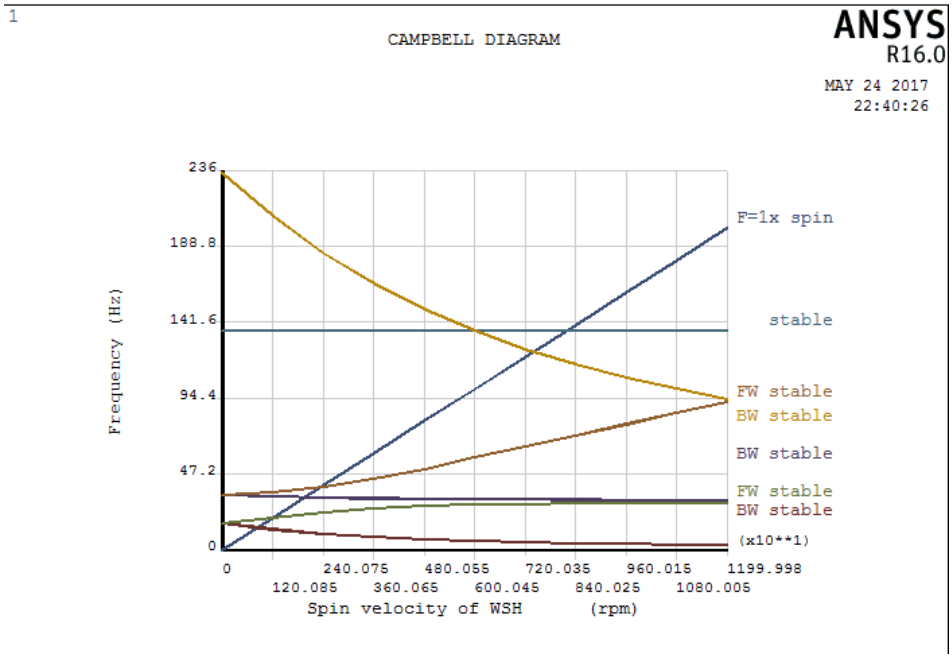


图 3.5 坎贝尔图

考虑到轴承刚度的变化，图 3.6 是前两阶无阻尼固有频率随轴承刚度变化的变化曲线：

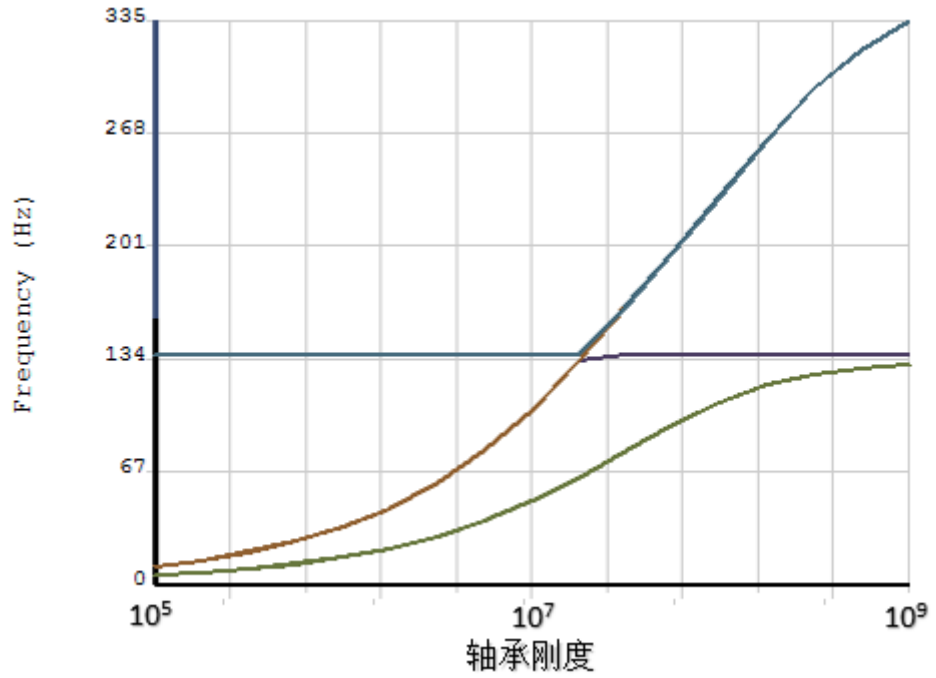


图 3.6 无阻尼固有频率随轴承刚度的变化

图中可以看到 1 阶无阻尼固有频率随着轴承刚度增大逐渐趋向于一个平稳值，在刚度为 10^9N/m 时为约 130Hz。而 2 阶无阻尼固有频率在刚度为 10^9N/m 时为约 334Hz，虽然没有明显收敛到某一值，但是也有要收敛的趋势。因为随着刚度的增大，轴承处趋向于固定，此时的特征频率不可能是无限大，所以它们最终都会要收敛。

3.3 结果的合理性分析

为了验证仿真结果的合理性，由于原转子系统的三维模型我没有，不过还是用一个简化后的三维模型来进行仿真，来验证一下用一维梁单元来进行仿真的正确性。图 3.7 为三维模型仿真得到的无阻尼模态频率和其 1 阶振型，轴承刚度取 10^6N/m。

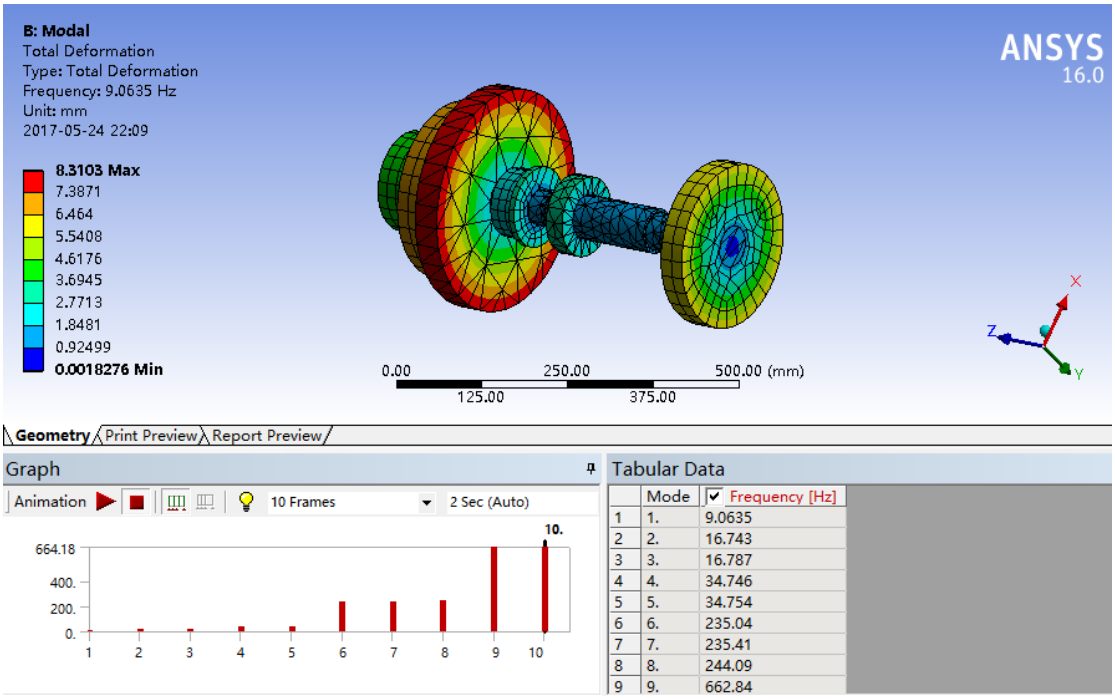
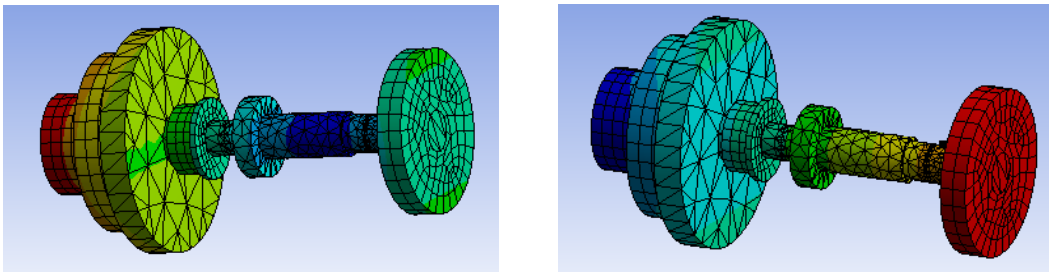


图 3.7　三维模型的无阻尼模态频率和其 1 阶振型

由图中右下角可以看到，此模型的前几阶模态中，除了第一阶模态是 9.0635Hz 外，第二阶模态 16.743Hz，与第三阶模态 34.746Hz，分别与一维梁单元构成的模型对应得很好。由图中的一阶振型图也可以看到，9Hz 对应的振型是由盘自身的振动引起的，而一维模型中直接把盘当做集中质量看待所以没有其振型，由此可以得到一维模型的一个局限性就是不能很好的分析盘自身的振动影响。图 3.8 为这两阶径向振动的振型，最大位移处也分别在叶轮段和涡轮端，说明这两阶径向振动与一维模型得到的振动相对应。



(a)第一阶径向振动振型　　(b)第二阶径向振动振型

图 3.8　径向振动振型

图 3.9 为轴承刚度取 10^9N/m 仿真得到的无阻尼模态频率和 5 阶振型（第二阶径向振动的振型）。

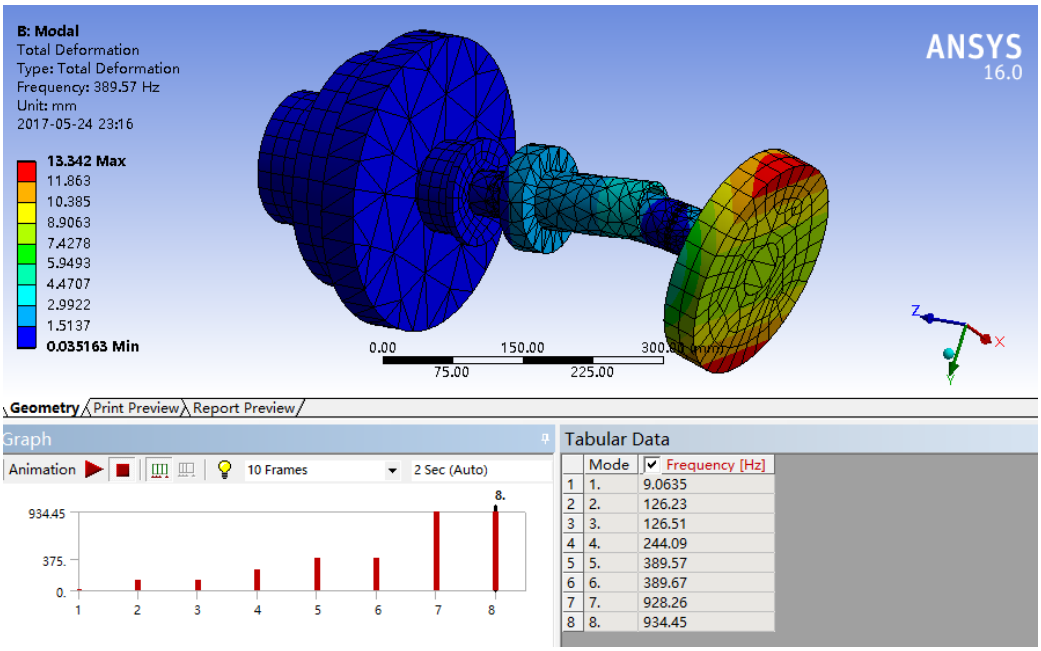


图 3.9　另一个三维模型的无阻尼模态频率和其 1 阶振型

由图中可以看出，轴承刚度变大了之后，其无阻尼的固有频率与一维模型出现了误差，第一阶径向振动频率为 126.23Hz，与一维模型的 130Hz 还算接近，但第二阶径向振动的频率 389.57Hz 就与一维模型的 334Hz 有较大的差别了。由图中的振型也可以看出，此时振型的最大位移是在盘间而不是轴端，即此时盘的自身厚度也对模态频率产生了影响。考虑到盘的厚度后，轴段在此处刚度应该增大，所以得到无阻尼固有频率就会变大。

第4章 总　结

在本次《计算动力学》大作业中，通过小组成员的共同努力，实现了以下四项主要工作：

1. 分析了 STAP90 程序的结构，增加了 8 节点 6 面体单元，使其能够求解三维实体单元的静力学问题，并将结果与 ABAQUS 软件的结果进行对比；

2. 调研了多种特征值求解的方法，基于一种改进的特征值求解算法拓展了 STAP90 程序，使其能够求解特征值问题；

3. 分析了前后处理软件 Gmsh 的读入格式，修改了 STAP90 程序的输出文件格式，使其能够在 Gmsh 中可视化显示位移场和应力场结果；

4. 在 ANSYS 中分析了涡轮增压机结构的振型和频率，基于一定程度的简化模型，讨论了结果的合理性。

其中，第一项工作由舒炫博同学完成、第二项由冯伟同学完成、第三项由王若溪同学完成、以及第四项由韦淞瀚同学完成。此外，韦淞瀚同学整合了展示 PPT、舒炫博同学整理了报告文档。尽管各部分主要工作相对独立，由个人分别完成，但在其中也针对程序各部分的接口进行了大量的讨论，比如如何改变 STAP90 程序的输出格式使之符合后处理软件读入接口等等，大作业进行过程中的很多问题均离不开小组成员的共同努力。

感谢小组的各位成员对大作业完成做出的贡献！

参考文献

[1] 李庆扬. 数值计算原理[M]. 清华大学出版社有限公司, 2000.

[2] 曹志浩. 矩阵特征值问题[M]. 上海科学技术出版社, 1980.

[3] Lanczos C. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators[M]. Los Angeles, CA: United States Governm. Press Office, 1950.

[4] Arnoldi W E. The principle of minimized iterations in the solution of the matrix eigenvalue problem[J]. Quarterly of applied mathematics, 1951, 9(1): 17-29.

[5] Davidson E R. The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices[J]. Journal of Computational Physics, 1975, 17(1): 87-94.

[6] 周树荃, 戴华. 求解大型对称特征值问题的块 Chebyshev-Lanczos 方法[J]. 南京航空航天大学学报, 1989, 21(4): 22.

[7] Underwood R R. An iterative block lanczos method for the solution of large sparse symmetric eigenproblems[J]. 1975.

[8] 赵中华. 用 Chebyshev 多项式加速的子空间迭代法[J]. 南京航空航天大学学报, 2002, 34(2): 197-200.

[9] Lehoucq R B. Implicitly restarted Arnoldi methods and subspace iteration[J]. SIAM Journal on Matrix Analysis and Applications, 2001, 23(2): 551-562.

[10] 聂永明. 对称广义特征值问题的数值求解算法的研究[D]. 南昌大学, 2012.

[11] 孟红燕. 求解矩阵特征值问题的 Inverse-free Krylov 子空间方法[D]. 南京航空航天大学, 2012.

[12] Van Beeumen R, Meerbergen K, Michiels W. A rational Krylov method based on Hermite interpolation for nonlinear eigenvalue problems[J]. SIAM Journal on Scientific Computing, 2013, 35(1): A327-A350.

附录 A ANSYS 命令流代码

```
/CLEAR
/PREP7

MP,EX,1,2.12E11
MP,PRXY,1,0.28
MP,DENS,1,7830
MP,EX,2,1.1E11
MP,PRXY,2,0.34
MP,DENS,2,4500

N,1,0
N,2,0.05
N,3,0.09
N,4,0.119
N,5,0.148
N,6,0.168
N,7,0.191
N,8,0.22
N,9,0.24
N,10,0.2445
N,11,0.249
N,12,0.266
N,13,0.283
N,14,0.31
N,15,0.337
N,16,0.378
N,17,0.419
N,18,0.46
N,19,0.501
```

```
N,20,0.51725
N,21,0.5335
N,22,0.5605
N,23,0.5865
N,24,0.6055

ET,1,BEAM188,,,3
TYPE,1
MAT,2
SECTYPE,1,BEAM,CSOLID
SECDATA,0.033,32,4
SECNUM,1
E,1,2
E,2,3
SECTYPE,2,BEAM,CSOLID
SECDATA,0.037,32,4
SECNUM,2
E,3,4
E,4,5
E,5,6
E,6,7
SECTYPE,3,BEAM,CTUBE
SECDATA,0.018,0.037,32
SECNUM,3
E,7,8
E,8,9
MAT,1
SECTYPE,4,BEAM,CSOLID
SECDATA,0.018,32,4
SECNUM,4
E,7,8
E,8,9
E,9,10
```

```
E,10,11
SECTYPE,5,BEAM,CSOLID
SECDATA,0.025,32,4
SECNUM,5
E,11,12
E,12,13
E,13,14
E,14,15
SECTYPE,6,BEAM,CSOLID
SECDATA,0.027,32,4
SECNUM,6
E,15,16
E,16,17
E,17,18
E,18,19
E,19,20
E,20,21
SECTYPE,7,BEAM,CSOLID
SECDATA,0.03,32,4
SECNUM,7
E,21,22
E,22,23
E,23,24

N,112,0.266,-0.01,0
N,120,0.51725,-0.01,0
N,212,0.266,0,-0.01
N,220,0.51725,0,-0.01
N,312,0.266,0.01,0
N,320,0.51725,0.01,0
N,412,0.266,0,0.01
N,420,0.51725,0,0.01
ET,2,COMBIN14
```

```
R,1,1e6/2,50
TYPE,2
REAL,1
E,12,112
E,12,212
E,12,312
E,12,412
E,20,120
E,20,220
E,20,320
E,20,420

ET,3,MASS21
TYPE,3
R,2,2.881,2.881,2.881,9.0385E-3,6.0562E-3,6.0562E-3
R,3,7.66,7.66,7.66,6.7999E-2,3.6147E-2,3.6147E-2
R,4,13.702,13.702,13.702,0.184769,0.0942113,0.0942113
R,5,1.113,1.113,1.113,2.1528E-3,1.2448E-3,1.2448E-3
R,6,0.673,0.673,0.673,1.241E-3,6.25E-4,6.25E-4
R,7,2.856,2.856,2.856,4.5865E-3,7.7487E-3,7.7487E-3
R,8,0.3874,0.3874,0.3874,5.666E-4,2.896E-4,2.896E-4
R,9,12.811,12.811,12.811,0.949599,0.049021,0.049021
REAL,2
E,2
REAL,3
E,4
REAL,4
E,6
REAL,5
E,8
REAL,6
E,10
REAL,7
```

```
E,15
REAL,8
E,22
REAL,9
E,24

!D,15,UX
!D,12,UY,,,,,UZ
!D,20,UY,,,,,UZ
!D,All,UX,,,,,rotx
D,112,ALl
D,212,ALL
D,312,ALl
D,412,ALL
D,120,ALL
D,220,ALL
D,320,ALL
D,420,ALL


ESEL,ALL
ESEL,S,TYPE,,1,3
CM,WSH,ELEM
ALLSEL,ALL

/SOLVE
ANTYPE,MODAL

MODOPT,QRDAMP,7,0.1,,on
!qrdopt,on
MXPAND,6
CORIOLIS,1,,,1
!beta,1e-6
```

```
CMOMEGA,WSH,0.1,0,0
SOLVE
*DO,I,1,10
CMOMEGA,WSH,100*I,0,0
SOLVE
*ENDDO

/POST1
PLCAMP,,1,RDS,,WSH
FINISH
```