SFEAS开发文档

# 1 单元类型

## 1.1 六面体单元(HEXA8)

六面体单元（Hexahedron）是一种常见的三维实体单元。线性的（低价）六面体具有8个节点，每个节点具有3个平动自由度，单元总自由度为24，积分点可取为8点高斯积分。

(1) 形函数fshape

HEXA8单元的形函数fshape为



式中：



（2）形函数对自然坐标的导数Dshape

HEXA8单元的形函数对自然坐标的导数Dshape为



（3）形函数对整体坐标的导数Dshp

HEXA8单元的形函数对整体坐标的导数Dshp为



式中：J为雅可比矩阵，由下式计算得到



式中：（*xi, yi, zi*）为节点*i*的坐标，*i*=1,2,…,8。

（4）单元的B矩阵BMat（应变-位移关系）



（5）单元的N矩阵NMat（计算质量矩阵）



（6）单元的材料矩阵DMat

各向同性线弹性材料的D矩阵为



式中：*E*为材料的弹性模量，*μ*为材料的泊松比。

（7）高斯积分点

HEXA8单元采用的是2×2×2的高斯积分，积分系数WP和积分点XP如下表所示：

表1.1 3D8I8单元的高斯积分点及系数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | WP | XP[0]=r | XP[1]=s | XP[2]=t |
| 1 | 1.0 | -0.5773502691896 | -0.5773502691896 | -0.5773502691896 |
| 2 | 1.0 | 0.5773502691896 | -0.5773502691896 | -0.5773502691896 |
| 3 | 1.0 | 0.5773502691896 | 0.5773502691896 | -0.5773502691896 |
| 4 | 1.0 | -0.5773502691896 | 0.5773502691896 | -0.5773502691896 |
| 5 | 1.0 | -0.5773502691896 | -0.5773502691896 | 0.5773502691896 |
| 6 | 1.0 | 0.5773502691896 | -0.5773502691896 | 0.5773502691896 |
| 7 | 1.0 | 0.5773502691896 | 0.5773502691896 | 0.5773502691896 |
| 8 | 1.0 | -0.5773502691896 | 0.5773502691896 | 0.5773502691896 |

HEXA8单元的刚度矩阵ElemK，质量矩阵ElemM以及阻尼矩阵ElemD为



式中：*dJ*为雅可比矩阵*J*的行列式，*GE*为单元结构阻尼（材料阻尼）。

## 1.2 五面体单元(PENTA6)

五面体单元（Pentahedron）是一种常见的三维实体单元。线性的（低价）五面体具有6个节点，每个节点具有3个平动自由度，单元总自由度为18，积分点可取为6点积分。

(1) 形函数fshape

PENTA6单元的形函数fshape为



式中：



（2）形函数对自然坐标的导数Dshape

PENTA6单元的形函数对自然坐标的导数Dshape为



（3）形函数对整体坐标的导数Dshp

PENTA6单元的形函数对整体坐标的导数Dshp为



式中：J为雅可比矩阵，由下式计算得到



式中：（*xi, yi, zi*）为节点*i*的坐标，*i*=1,2,…,6。

（4）单元的B矩阵BMat（应变-位移关系）



（5）单元的N矩阵NMat（计算质量矩阵）



（6）单元的材料矩阵DMat

各向同性线弹性材料的D矩阵为



式中：*E*为材料的弹性模量，*μ*为材料的泊松比。

（7）高斯积分点

PENTA6单元采用的是6点积分，积分系数WP和积分点XP如下表所示：

表1.1 3D8I8单元的高斯积分点及系数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | WP | XP[0]=r | XP[1]=s | XP[2]=t |
| 1 | 0.1666666666667 | 0.1666666666667 | 0.1666666666667 | -0.5773502691896 |
| 2 | 0.1666666666667 | 0.6666666666667 | 0.1666666666667 | -0.5773502691896 |
| 3 | 0.1666666666667 | 0.1666666666667 | 0.6666666666667 | -0.5773502691896 |
| 4 | 0.1666666666667 | 0.1666666666667 | 0.1666666666667 | 0.5773502691896 |
| 5 | 0.1666666666667 | 0.6666666666667 | 0.1666666666667 | 0.5773502691896 |
| 6 | 0.1666666666667 | 0.1666666666667 | 0.6666666666667 | 0.5773502691896 |

PENTA6单元的刚度矩阵ElemK，质量矩阵ElemM以及阻尼矩阵ElemD为



式中：*dJ*为雅可比矩阵*J*的行列式，*GE*为单元结构阻尼（材料阻尼）。

## 1.3 四面体单元(TETRA4)

四面体单元（Tetrahedron）是一种常见的三维实体单元。线性的（低价）四面体具有4个节点，每个节点具有3个平动自由度，单元总自由度为12，积分点可取为1点积分。

(1) 形函数fshape

TETRA4单元的形函数fshape为



式中：



（2）形函数对自然坐标的导数Dshape

TETRA4单元的形函数对自然坐标的导数Dshape为



（3）形函数对整体坐标的导数Dshp

TETRA4单元的形函数对整体坐标的导数Dshp为



式中：J为雅可比矩阵，由下式计算得到



式中：（*xi, yi, zi*）为节点*i*的坐标，*i*=1,2,…,4。

（4）单元的B矩阵BMat（应变-位移关系）



（5）单元的N矩阵NMat（计算质量矩阵）



（6）单元的材料矩阵DMat

各向同性线弹性材料的D矩阵为



式中：*E*为材料的弹性模量，*μ*为材料的泊松比。

（7）高斯积分点

TETRA4单元采用的是1点积分，积分系数WP和积分点XP如下表所示：

表1.1 3D8I8单元的高斯积分点及系数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | WP | XP[0]=r | XP[1]=s | XP[2]=t |
| 1 | 0.1666666666667 | 0.25 | 0.25 | 0.25 |

TETRA4单元的刚度矩阵ElemK，质量矩阵ElemM以及阻尼矩阵ElemD为



式中：*dJ*为雅可比矩阵*J*的行列式，*GE*为单元结构阻尼（材料阻尼）。

## 1.4 总体矩阵的形成

稀疏矩阵的具有多种存储格式，本文仅对压缩稀疏行存储（CSR）格式进行介绍。

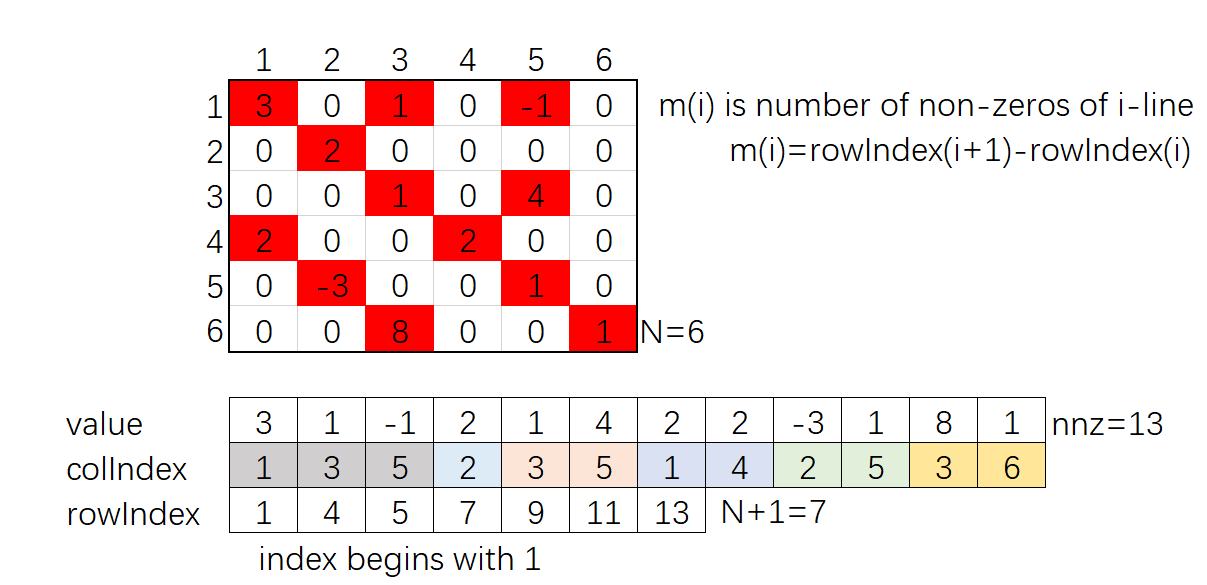


图1.4 压缩稀疏行存储（CSR）格式

如上图1.4所示，在存储稀疏矩阵的时候，需要有3个数组来表示所有信息。一个为value数组，长度为NNZ，用于存储非零元素；一个为colIndex数组，长度为NNZ，用于存储非零元素所对应的列索引信息，一个为rowIndex数组，长度为N+1，用于存储非零元素所对应的行索引信息。需要注意的是rowIndex数组还包含其他有用的信息：rowIndex(i)表示第i行的第一个元素在value中的索引，所以有rowIndex(1)=1和rowIndex(N+1)-rowIndex(1)=NNZ，此外m(i)=rowIndex(i+1)-rowIndex(i)表示第i行非零元素的数量。

在有限元中，常采用CSR格式来存储总体矩阵（总体刚度、质量和阻尼矩阵等），原因有两个：一方面可以节省内存开销，有限元的总体矩阵大多为稀疏矩阵且稀疏比较大，以某一具体案例为参考，求解自由度N=669330，NNZ=14049513，稀疏比为0.00612266%。考虑对称性，按照稠密矩阵以double类型存储，占用内存大小为669330×(669330+1)/2×8byte=1.63TB。按照稀疏矩阵CSR以double类型存储，占用内存大小为14049513×8byte=0.10GB。另一方面可以加快矩阵求解（运算）速度，且第三方数学运算库均提供稀疏矩阵CSR存储格式。

在结构有限元中，如何得到总体稀疏矩阵的全部信息，即得到总体稀疏矩阵的rowIndex数组，colIndex数组以及value数组是一大难点，也是后续求解方程的基础。其中rowIndex和colIndex仅与节点单元之间的连接相关，即网格的拓扑信息，因此其可以事先确定好。Value与单元矩阵有关，需要先计算单元矩阵，然后将其添加到value中相应位置。

以下图1.5所示的简单示例来说明总体稀疏矩阵的形成过程，示例有2个四边形单元构成，总共6个节点，假设每个节点2个自由度，则系统共有12个自由度。图1.5（a）显示了网格拓扑关系，图1.5（b）显示了系统总体矩阵的稀疏情况，红色的圆圈表示非零元素，黑色的圆圈假设其也表示非零元素，实际上其是零元素。以节点1和5关系来看，其并非直接连接，但其都属于单元1上面的节点，在通过积分点计算单元矩阵并没有判断其是否非0，直接将其添加到总体矩阵的相应位置（单元矩阵当成稠密矩阵处理了），因此在总体结构中认为其位置为非零元素。此种假设对生成总体稀疏矩阵格式和判断相关节点都带来很大的方便。

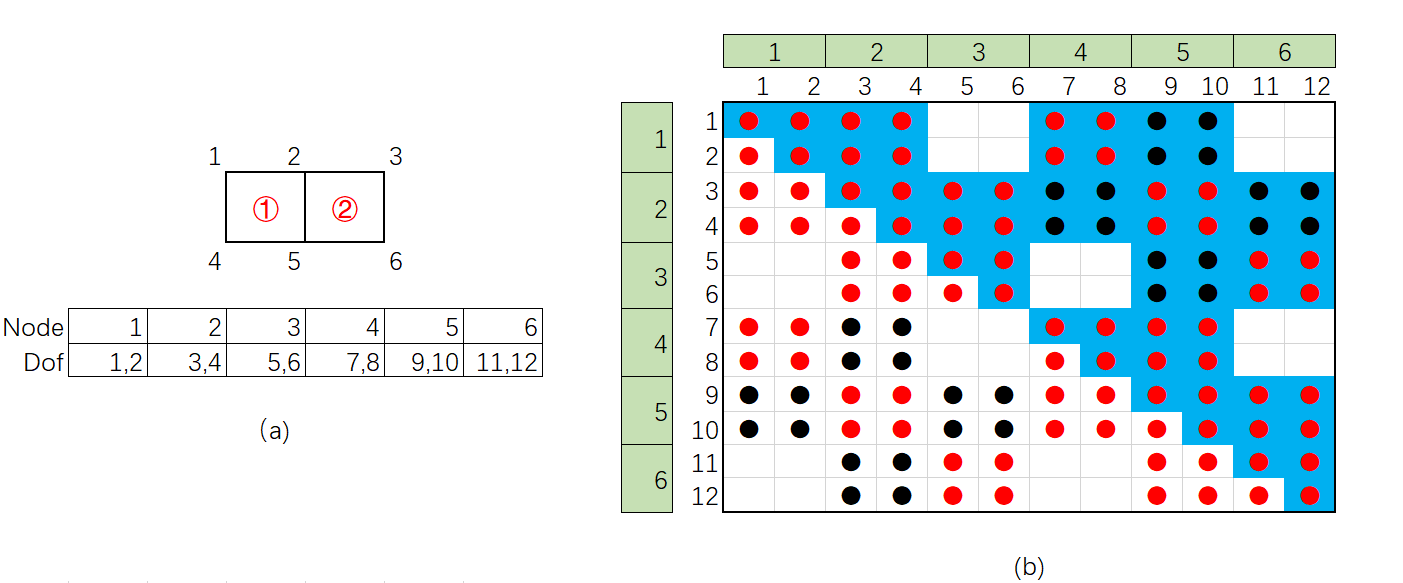


图1.5 总体稀疏矩阵的形成过程

相关节点就是指在网格信息中直接相连接的节点，需要说明的是同一单元内的节点都属于相关节点，自身与自身是也是相关节点。这样就可以通过网格拓扑信息得到每个节点的相关节点列表（不可重复且升序排列），具体做法是：先遍历每个单元，在对每个单元遍历其每个节点，得到每个节点的相关节点列表。

如下图1.6所示，图1.6（a）所示的conNodeList数组为每个节点的相关节点列表，需要说明的是conNodeList数组的第一维度为总节点数nNode，第二维度大小事先并不知道，有两种处理方法，一是事先假设一个较大的维度，二是采用数据结构，在C++可采用set关联式容器，每次将相关节点insert即可，set属性满足了不可重复且默认升序排列的特性。图1.6（b）所示的conDofList数组为每个节点的相关节点自由度列表，需要说明的是conDofList数组的第一维度为总自由度数nDof（也为总系统方程数量），第二维度为每个节点自由度相关联的自由度数量，也就是每个方程相关的方程数量，即总体刚度矩阵中每行的非零元数量，且每个元素的具体数值表明了其在总体矩阵中的列位置。对于对称矩阵，只想要存储上三角情况（列>=行），则只需选择数组大于行号的部分即可。这样，通过对相关节点自由度列表conDofList数组的解析，就可以得到总体稀疏矩阵的rowIndex和colIndex信息。

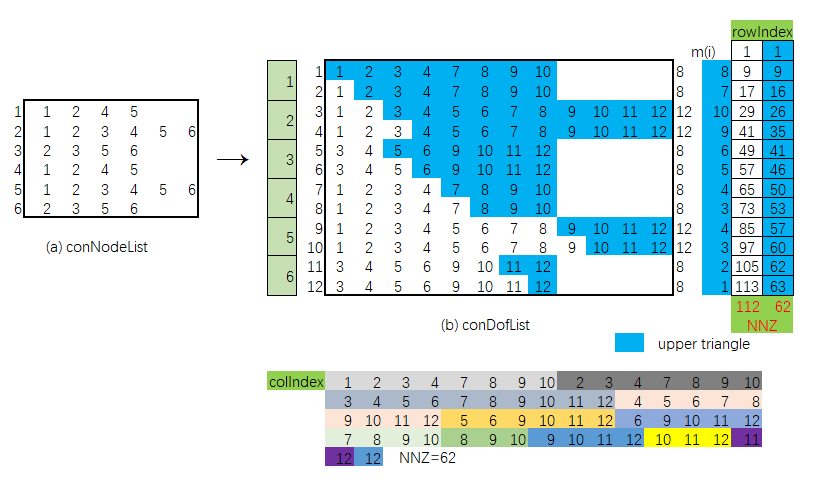


图1.6 相关节点及相关自由度列表

最后就是确定总体稀疏矩阵的value信息，如下图所示，需要先计算单元矩阵，然后将其添加到value中相应位置。计算完单元刚度矩阵后，可根据该单元上节点对应的自由度判断其该元素在总体矩阵中的行列信息，然后在通过rowIndex和colIndex数组即可确定其在value中的具体位置。具体做法为：得到行列信息(iR, iC)后，通过遍历第iR行所有元素，找到colIndex(IP)=iC的IP，则(iR, iC)在总体稀疏矩阵中对应的位置为IP，最后将单元矩阵中的值添加在value(IP)即可。

# 2 分析类型

动力学（含静力学，以下均以动力学表述）的主要理论为空间离散后得到的半离散格式的动力学方程



式中：*M*为系统的总体质量矩阵，*C*为系统的总体阻尼矩阵，*K*为系统的总体刚度矩阵，*x*为位移向量，*F(t)*为载荷向量。

根据具体问题的不同，可将动力学分析分为以下几种：静力学分析，模态分析，瞬态动力学分析，频率响应分析等。

静力学分析：是指在不随时间变化的载荷下，分析物体的刚强度情况。由于载荷不随时间变化，因此动力学方程简化为



模态分析：是指对结构的固有特性进行分析，是在无外载荷的情况下进行的。由于没有外载荷，因此动力学方程简化为：



瞬态动力学分析：是指在随时间变化的载荷下，分析物体的时域动态响应情况。动力学方程为：



频率响应分析：是指在频率上求解动力学方程，分析物体的频域动态响应情况，因此动力学方程简化为：



式中：*ω*为扫频的圆频率。

## 2.1 静力分析(STA)

针对静力学分析的方程，考虑了载荷条件（位移和力载荷）后，可得：



式中：*uf*为待求解的自由位移；*ur*为给定的位移；*F*为施加的力载荷；*R*为待求解的支反力。当给定的位移*ur*=0，即表示固定约束。

按照式中第一式展开，得到：



因此，通过求解式可得到待求解位移*uf*。

按照式中第二式展开，得到：



因此，可通过求解式可得到待求支反力*R*。

静力学分析的重点是总体刚度矩阵的组装和求解线性方程组，这两部分也是结构有限元中最为耗时的部分。

在结构有限元中，一般情况下刚度矩阵*K*是一个稀疏、对称的矩阵，一般常采用稀疏矩阵格式存储，如何组装形成稀疏格式的总体刚度矩阵是重点，具体将在后续介绍。刚度矩阵*K*的维度即为系统的求解自由度，对于大型有限元模型，多达几百万，上千万级的自由度，因此如何高效地求解这一大型稀疏线性方程组尤为重要。

对于这样的稀疏线性方程组的求解主要有两种方式：直接法和迭代法。直接法的核心思想就是高斯消元，根据矩阵性质的不同，进行LU分解（非奇异）或LDLT（非奇异对称）或LLT（也称为cholesky分解，非奇异对称正定）分解。直接法求解的结果从理论上来说是精确解（不考虑计算机本身所产生的截断误差），但直接法的缺点就是将K分解成L和U矩阵（以LU分解为例），会占用大量的内存。由于不确定L和U矩阵的稀疏程度（取决于K矩阵的性质），因此L和U所占用的内存可能会远大于K占用的内存，这就使得直接法需要更多的内存。当然直接法优点在于分解之后可以多次回代求解，这一特性在多工况（多右端项）求解以及在瞬态分析中应用较广。多工况问题是刚度矩阵K固定，右端项有多个，每个工况表示一种工况；瞬态分析中在线性问题以及时间步长不变的情况下，每个时间步只有右端项在变化，等效刚度矩阵是不变的。因此这两种情况下直接法只需要分解一次。迭代法的核心思想就是不断迭代计算*K*{*xi*}=*Fi*，直至{*Fi}*接近于{*F*}，则{*xi*}就接近于真实解{*x*}。相较于直接法，不需要对矩阵进行分解操作，取而代之的是进行多次的稀疏矩阵乘以向量的运算，当然迭代法求解关键的一个操作是对矩阵*K*进行预处理操作。

## 2.2 模态分析(MOA)

针对模态分析的方程，假设{*x*}={*Φ*}*eμt*对上述方程进行处理，可得:



式中：-*μ2*为特征值，可记为*λ*，*Φ*为特征向量。

一般来说，上式所表述的特征值为复特征值系统（*μ*和*Φ*为复数）。在结构有限元中，结构的质量矩阵*M*为对称、正定矩阵，刚度矩阵*K*为不一定为对称矩阵、也不一定为正定矩阵（存在非线性情况）。当忽略结构的阻尼矩阵*C*时，且刚度矩阵*K*为对称时，此时方程具有特征值*λ*为实数（即*μ*为实数或纯虚数），特征向量*Φ*为实数的特性。

进一步当刚度矩阵*K*为对称、半正定矩阵（不考虑非线性情况），特征值*λ*为正实数（即*μ*为纯虚数）。此处说明一点，当采用拉格朗日乘子法处理连接、约束关系时，得到的刚度矩阵*K*可能是非正定，但质量矩阵*M*中对应的拉格朗日乘子矩阵为0，此种情况也满足特征值*λ*为正实数（即*μ*为纯虚数）。此时，可令*μ=iω*，*ω*为固有圆频率，其满足*ω=2πf*，*f*称为固有频率，单位为Hz。方程进一步简化为：



模态分析一般可分为实模态分析和复模态分析。实模态分析通常也简称为模态分析，即特征值为实数，特征向量为实数，对应数学表达式为方程。复模态分析，即特征值为和特征向量为复数，对应数学表达式为方程。比如阻尼矩阵*C*不能忽视（转子动力学分析），刚度矩阵*K*不对称（存在非线性情况）等情况。

通常结构有限元中模态分析指的是实模态分析，数学本质就是求解方程，得到固有圆频率*ωi*和对应的振型向量*φi。*振型向量*φi*与固有圆频率*ωi*构成一对，常称为特征对。固有圆频率（习惯上采用固有频率来表达）顾名思义就是该结构的固有的一个特性，当外载荷的频率接近于固有频率时，结构就可能发生共振现象。振型向量就表示在该频率下结构的每个节点的位移大小，但由于模态分析无外载荷作用，该值并不表示实际情况下的位移大小，不具备实际意义，且通常振型向量进行了对质量矩阵的归一化处理。此外振型向量还具有正交特性，此特性是基于模态叠加法求解动力学问题的核心思想。即：



自由模态和约束模态的区别在于：自由模态无约束边界条件，结构刚度矩阵*K*为对称、半正定、奇异矩阵，此时特征值*λ*为正实数，可写成*μ=iω*形式，但存在*ω*等于0的情况，在物理上称之为刚体模态。约束模态存在约束边界条件，结构刚度矩阵*K*为对称、正定、非奇异矩阵，保证了特征值写成*μ=iω*形式，且*ω*大于0的情况。

关于模态分析的输出结果，除了固有频率及其振型向量（振型云图）外，还有模态参与因子，模态有效质量、累计质量分数等。以ANSYS关于模态分析的输出为例，介绍其各个输出量的含义及计算过程。

（1）固有频率FREQUENCY

调用特征值求解器之后即可求得固有圆频率*ωi*，因此只需要将固有圆频率*ωi*转化为固有频率*fi*即可，两者关系为



（2）周期PERIOD

周期*pi*的值为固有频率*fi*的倒数，即



式中：*pi*为第*i*阶固有频率所对应的周期。

（3）模态参与因子PARTIC.FACTOR

模态参与因子*γ*是一种描述模态和某一向量激励相互作用关系的参数，值越大，代表该模态对动态响应的贡献越大。注意，模态参与因子是存在正值和负值的。在计算模态参与因子之前，先介绍其广义质量*Mi*，定义为



式中：*Mi*为第*i*阶模态的广义质量，一般振型向量进行了对质量矩阵的归一化处理，所以广义质量*Mi* =1。

模态参与因子*γij*定义为



式中：*γij*为第*i*阶模态对应的*j*方向的模态参与因子，*j=1,2,3,4,5,6*，下同，*Tnj*表示施加了*j*类型刚体运动的模型中自由度*n*的刚体相应大小。



式中：*ei*为单位向量，*x,y,z*为节点坐标，*x0,y0,z0*为旋转中心的坐标。

这样，就可以得到某个模态下平动自由度和转动自由度的参与因子。

（4）模态参与因子比例RATIO

模态参与因子比例*r*是用于描述该阶模态参与因子*γ*在所求的模态参与因子的最大值的占比，计算公式为



式中：*rij*为第*i*阶模态对应的*j*方向的模态参与因子比例。

（5）模态有效质量EFFEC. MASS

模态有效质量指在某一向量激励下，某一模态参与的系统质量。模态有效质量*Meij*定义为



式中： *Meij*为第*i*阶模态对应的*j*方向的模态有效质量。

各方向所有的模态有效质量之和等于系统总质量。如果在分析中所使用的模态有效质量之和远小于系统总质量，表明该阶模态振型向量对系统的响应参与甚少。

（6）累计质量分数CUMU.MASS FRAC.

累计质量分数的定义为当前模态的之前所有模态有效质量和在总有效质量的占比，计算公式为



式中：*m*为所求的模态阶数。

（7）模态有效质量占比EFFEC/TOTAL MASS

模态有效质量的定义为当前阶的模态有效质量与总质量的比值，计算公式为



式中：*Mrig*为整个结构的总刚体质量属性，其维度为6×6的矩阵，计算公式为



式中：*nElem*为结构的单元总数，*Tk*的含义与式是相同的，*Mk*为第*k*个单元的质量矩阵。

需要说明一点，上式中的求和就是矩阵加法求和，并不表示“单元的组装求解”，关于*Mrig*的计算可放在单元质量矩阵组装过程进行。

## 2.3 瞬态分析(DTA&MTA)

针对瞬态分析的方程，考虑了载荷条件（位移和力载荷）后，可得：



式中：*uf*为待求解的自由位移；*ur*为给定的位移；*F*为施加的力载荷；*R*为待求解的支反力。当给定的位移*ur=*0，即表示固定约束。

按照上式第一行展开，得到：



式中：三者知其一，可求其二。具体的求解方法和时间积分方法中求解三者之间关系的方法一样。

因此，可通过求解得到*uf*，然后结合给定的位移*ur*，可得到最终位移*u*。



按照上式第二行展开，得到：



因此，可通过求解式得到可以支反力*R*。

对于上式，在生成节点自由度编号时就将给定位移*ur*的自由度放在最后，如式所示。在矩阵的组装过程就根据节点自由度的编号生成式中的4个矩阵。求解上式可采用直接法和模态法进行求解。

（1）直接法：所谓直接法就是直接对时间积分，亦称为完全法。

直接积分法可分为显式积分方法和隐式积分方法，显式积分方法有中心差分法等，隐式积分方法有Newmark算法、HHT算法等。

在时间积分方法中，显式与隐式积分方法的区别在于等效刚度矩阵中是否显含刚度矩阵*K*，显式积分方法中一般是不含*K*，且采用集中质量矩阵，当合理地处理阻尼矩阵*C*后，这就使得等效刚度矩阵为对角阵。这使得每个时间步求解的线性方程组解耦了，变成了对角矩阵的方程组求解，极大地节省了时间。

从稳定性角度来看，显式积分算法一般是有条件稳定的，即稳定性取决于时间步长大小，一般来说，显式积分算法的时间步长要求很小，具体可参考*CFL*条件。而隐式积分算法一般来说可以做到无条件收敛，即时间步长不影响积分的收敛性。从精确度来看，显式和隐式积分算法均无振幅误差，显式积分算法存在周期缩短误差，而集中质量矩阵常常会延长周期，这样显示积分算法中使用集中质量矩阵不仅对计算速度有益，也对计算精度有益。相反隐式积分算法存在周期延长误差，因此常搭配一致质量矩阵。

（a）Newmark算法

Newmark算法是一种常见的隐式积分方法，其将*t+Δt*时刻的位移和速度离散化为：



该算法引入了2个积分常数*α*和*δ*，将上式代入*t+Δt*时刻的动力学方程可得：



式中，*Ai*为积分系数。



由上式可以进一步得到Newmark法的迭代格式：



式中，*Bi*为积分系数。



当积分常数*α*和*δ*满足下式，Newmark法为无条件收敛。



针对上式的条件，一般给定默认值为*α*=0.25250625，*δ*=0.505。

需要说明的是，迭代格式写成上式是为了将各种积分方法的统一。

（b）HHT算法

在Newmark法基础上引入积分常数*α*m和*α*F，从而引入算法的数值阻尼特性，得到的算法成为广义-*α*法，具体如下：



结合式，带入如下的动力学方程：



可得：



式中，*Ai*为积分系数。



由上式可以进一步得到广义-*α*法的迭代格式：



式中，*Bi*为积分系数。



当积分常数*α*、*δ、α*m和*α*F满足下式，广义-*α*法为无条件收敛。



在广义-*α*法的基础上，当*α*m=0时，称为HHT-*α*算法。针对上式的条件，一般给定默认值为*α*=0.25250625，*δ*=0.505，*α*F=0.005。

上述按照时间步长进行推进求解时，时间步长Δt一般是固定的。通过式和式可以看出，最终求解的线性方程组的系数矩阵与时间步长Δt相关，当其固定不变时，系数矩阵是固定的，此时采用直接法求解只需分解系数矩阵一次，每个时间步下只需计算右端项进行回代求解即可，可节省大量时间。当分析非线性问题时，系数矩阵在每个时间点是需要重新计算的，此时，可考虑变时间步长Δt技术，也成为自动时间步长。其基本思想为：判断*tn*时刻和*tn+1*时刻的中间时刻是否达到平衡（残差*R1/2*与给定容差*tol*比较），当残差小于给定容差时，可考虑增加时间步长，反之则反。



式中：*R1/2*为*tn*时刻和*tn+1*时刻的中间时刻外载荷和内载荷的差值，*factor*为时间步长变化系数，*tol*为给定的容差，Δtn和Δtn+1为*tn*时刻和*tn+1*时刻的时间步长，Δtmin和Δtmax为给定的时间步长范围，对时间步长进行限制，避免时间步长过大造成精度的丢失，也避免时间步长太小造成时间的浪费。

*tn*时刻和*tn+1*时刻之间任意时刻的残差可参考下式进行计算：



上式：为*ι*时刻的位移、速度和加速度。



式中：*Ci*为系数。

当积分方法为Newmark算法时，系数*Ci*为：



当积分方法为HHT算法时，系数*Ci*为：



（2）模态法：所谓模态法就是利用结构的振型矩阵（选取有限阶数）进行模态坐标变换，把动力学方程从物理空间转化到模态空间上去求的模态解，然后返回物理空间得到最终物理解，亦称为振型叠加法。

模态法最大的优势为提升了计算效率，其降低了方程求解的维度，从原有的*n*维度降低为*m*维度，其中*n*为系统的总自由度，*m*为选取的模态阶数，通常*m*远小于*n*。当然，求解结构的振型矩阵（模态分析）也需要时间，但好在其只需要进行一次（系统的刚度*K*和质量矩阵*M*不变，这也是模态叠加法不能处理非线性瞬态问题的一方面原因）。另一方面选取多大的截断模态*m*是合适的，过小的*m*会带来截断误差，理论上*m*越接近于*n*越好，但过大的*m*会使得模态求解的时间增大。针对截断误差，可采相应的修正方法来弥补，比如静力学修正法（模态叠加法）等。

通过模态振型矩阵将动力学方程从物理空间转化到模态空间上，物理空间下方程为：



假设可得模态空间下方程为：



式中：



因此，只需在模态空间求解式即可。其中CM不一定为对角阵，取决于阻尼的类型。当存在单元结构阻尼*D*（材料阻尼）时，其不是对角阵。针对此方程可采用时域积分方法求解模态空间解*y*，然后通过下式求得物理空间解*x*。



式中：*φi*为第*i*阶振型，*yi*为模态空间下的坐标。

采用模态法进行求解上式，需要说明几点。式右端项中-*K12ur*所产生的响应在物理上与振型关系不密切，不能由低阶振型组合确定。因此在使用模态叠加法求解时，会产生较大的误差。因此需要对式中的右端项进行分类。

由基础位移*ur*所引起的响应称为拟静态位移*us*，由基础加速度和速度所引起的响应称为动态位移*ud*。则上式可写成：



拟静态位移*us*不能由低阶振型组合所确定，动态位移*ud*与振型密切相关，可由低阶振型组合所确定。具体由下式计算：





求解的基本思路是先由给定位移*ur*通过求解式得到拟静态位移*us*，然后由式通过模态叠加法计算得到动态位移*ud*。

由于上述位移都是时间t的函数，因此在每个时间步下都需要求解上述位移。如在每个时间步下求解式的方程得到拟静态位移*us*，这样就失去了模态叠加法计算速度快的优势了。一种方法是求出*K11*的逆矩阵，得到矩阵*G12*，即



然后在每个时间步下通过乘法运算得到拟静态位移*us*。但这种方法涉及到稀疏矩阵*K11*的求逆，计算量很大，且*K11*的逆稀疏性未知，此方法并不容易实现。

可采用如下方法进行处理，利用式线性方程组的叠加特性。考虑给定位移*ur(t)*形式如下:



式中：为基底向量，*l*可以理解为给定位移施加的数量。基地向量*ub*可通过位移载荷输入获取，即当前给定位移对应的自由度为1.0，其余为0。

事先求解如下的多右端项方程，得到基底解向量*ubb*。



然后基于线性方程组的叠加性可得到拟静态位移*us*：



这样只需要在每个时间步下通过式得到拟静态位移*us*，然后再使用模态叠加法求解式得到动态位移*ud*。这样就保留了模态叠加法计算速度快的优势。

针对瞬态动力学方程中的阻尼矩阵C进行进一步说明，其是综合阻尼矩阵，瞬态分析中的阻尼可有以下几个部分构成：



式中：C*1*为瑞丽阻尼，C*2*为全局结构阻尼，C*3*为材料结构阻尼，C*4*为模态阻尼。

四种阻尼形式的定义如下：



式中：*α*，*β*为刚度和质量参数，均为输入参数。



式中：*Gf*为全局结构阻尼系数，*W3*为转化频率，均为输入参数。



式中：*GEf*为材料结构阻尼系数，*W4*为转化频率，*GE*为材料阻尼系数，均为输入参数。*KE*为单元的刚度矩阵。



式中：为模态阻尼在频域下表现形式，为对角线矩阵，用于记录各频率下的模态阻尼，一般可通过模态试验测量得到。*C4*为模态阻尼在频域下表现形式，可通过最小二乘法求解式得到，由于模态阻尼一般仅在采用模态法中才使用到，因此很少计算*C4*。

针对模态叠加法截断模态*m*带来截断误差，可采相应的修正方法来弥补，比如静力学修正法（模态加速度法）等。 对式进行修正



式中：*φi*为第*i*阶振型，*ωi*为第*i*阶固有圆频率，*yi*为模态空间下的坐标，*F*为外载荷，*K*为系统的刚度矩阵。

式中关于*K*求逆的操作，与上式处理方式一致，利用线性方程组的叠加特性进行处理，只需求解一次静力学分析即可。为了更加方便理解修正方法在模态叠加法的基础上修正了什么，将式改写成如下形式



式中：*K-1*可写成



式中：*n*为结构的总自由度。

因此，将式代入式中，得



式中模态叠加法修正后的响应*x*相比于未修正的响应*x*多出了一项，称之为残差向量（Residual Vector），记为*xR*，有时也称修正方法为残差向量法。



针对模态叠加法截断模态*m*带来截断误差，可采相应的修正方法来弥补，其本质在于找出舍去的高阶模态所带来的误差项（残差向量），然后通过某种方法间接求解误差项，以达到修正的效果。比如通过一次静力学求解来得到误差项，称为静力修正法。

## 2.5 频响分析(DFA&MFA)

针对频率响应分析的方程，考虑了载荷条件（位移和力载荷）后，可得：



式中：*uf*为待求解的自由位移；*ur*为给定的位移；*F*为施加的力载荷；*R*为待求解的支反力。当给定的位移*ur=*0，即表示固定约束。

其处理方法与瞬态动力学完全一致。需要注意的是频响分析的载荷为复数。在频响分析中，输入输出都是位移项，因此在求解过程中不需要处理位移、速度和加速度的关系。只需在求解前后进行处理即可，三者知其一，可求其二，具体求解方法如下：



将上式按照第一行展开得到：



因此，可通过求解式得到*uf*，然后结合给定位移*ur*，可得到最终位移*u*。



将上式按照第二行展开得到：



因此，可通过求解式得到支反力*R*。

求解上式可采用直接法和模态法进行求解。

（1）直接法：是指直接求解式得到响应。

（2）模态法：所谓模态法就是利用结构的振型矩阵（选取有限阶数）进行模态坐标变换，把动力学方程从物理空间转化到模态空间上去求的模态解，然后返回物理空间得到最终物理解，亦称为振型叠加法。

通过模态振型矩阵将动力学方程从物理空间转化到模态空间上，物理空间下方程为：



假设可得模态空间下方程为：



式中：



因此，只需在模态空间求解式即可。其中*CM*不一定为对角阵，取决于阻尼的类型。当存在单元结构阻尼*D*（材料阻尼）时，其不是对角阵。针对此方程直接求解模态空间解*y*，然后通过式求得物理空间解*x*。

频响分析中的线性方程组为复数域，该方程不是按照频率点推进迭代计算的，每个频率下的方程是独立的，下一频率点的计算与上一频率点无关系，因此对频率点的选取并没有限制。需要说明一点，频响分析得到的解为频域内的稳态解，瞬态分析得到的解为时域内的瞬态解，两者之间的关系并不是简单的傅里叶变换关系。

在模态法求解频响分析中，固有频率是事先求得，在固有频率附近，响应时会出现峰值的，为使得频率响应曲线更为光滑，可在固有频率附近设置更多的频率点。

假设扫频设置的范围为[*Fsta*, *Fend*]，中间有*N*个固有频率点，记为*Fi*，将扫频范围分成了*N+1*个子频率范围。则在第*i*个子频率范围内[*Fi-1*, *Fi*)，*i=*1,2,…,*N*+1，其中*F0=Fsta*，*FN+*1*=Fend*。在每个子频率范围内设置*NEF-*1个扫频点（不包括右端点），则全局共有(*N+*1)*\**(*NEF-*1)*+*1个扫频点（包含了起始频率）。其满足中间稀疏，两端稠密的特点，这样就达到了在固有频率点附近具有更多频率点的特性。



式中：*Fi-1*, *Fi* 为第*i*个子频率范围上下限频率，*i=*1,2,…,*N*+1，*fk*为该子频率范围内的第*k*个扫频点，*k*=1, 2,…,*NEF*-1，*c*为聚集系数（c>1.0），用于表征扫频点向两端的聚集程度，通常可取*c*=2.0，*ξk*为等参坐标，范围为[-1, 1)，计算公式为



在直接频响分析中，虽事先不知道固有频率，但可根据上一频率点的响应与当前频率点响应之间的差值进行判断，当响应差值很大时说明响应曲线变化陡峭，当前频率可能在固有频率附近，因此可缩小频率步长进行计算，称此方法为自动频率步长。

与采用模态法求解瞬态动力学问题一样，采用模态法进行求解上式，需要说明几点。式右端项中-*K12ur*所产生的响应在物理上与振型关系不密切，不能由低阶振型组合确定。因此在使用模态叠加法求解时，会产生较大的误差。因此需要对式中的右端项进行分类。

由基础位移*ur*所引起的响应称为拟静态位移*us*，由基础加速度和速度所引起的响应称为动态位移*ud*。则上式可写成：



拟静态位移*us*不能由低阶振型组合所确定，动态位移*ud*与振型密切相关，可由低阶振型组合所确定。具体由下式计算：





求解的基本思路是先由给定位移*ur*通过求解式得到拟静态位移*us*，然后由式通过模态叠加法计算得到动态位移*ud*。

由于上述位移都是频率*ω*的函数，因此在每个频率步下都需要求解上述位移。如在每个频率步下求解式的方程得到拟静态位移*us*，这样就失去了模态叠加法计算速度快的优势了。一种方法是求出*K11*的逆矩阵，得到矩阵*G12*，即



然后在每个频率步下通过乘法运算得到拟静态位移*us*。但这种方法涉及到稀疏矩阵*K11*的求逆，计算量很大，且*K11*的逆稀疏性未知，此方法并不容易实现。

可采用如下方法进行处理，利用式线性方程组的叠加特性。考虑给定位移*ur(ω)*形式如下:



式中：为基底向量，*l*可以理解为给定位移施加的数量。基地向量*ub*可通过位移载荷输入获取，即当前给定位移对应的自由度为1.0，其余为0。

事先求解如下的多右端项方程，得到基底解向量*ubb*。



然后基于线性方程组的叠加性可得到拟静态位移*us*：



这样只需要在每个频率步下通过式得到拟静态位移*us*，然后再使用模态叠加法求解式得到动态位移*ud*。这样就保留了模态叠加法计算速度快的优势。

针对频率响应动力学方程中的阻尼矩阵C进行进一步说明，其是综合阻尼矩阵，瞬态分析中的阻尼可有以下几个部分构成：



式中：C*1*为瑞丽阻尼，C*2*为全局结构阻尼，C*3*为材料结构阻尼，C*4*为模态阻尼。

四种阻尼形式的定义如下：



式中：*α*，*β*为刚度和质量参数，均为输入参数。



式中：*Gf*为全局结构阻尼系数，均为输入参数。



式中：*GEf*为材料结构阻尼系数，*GE*为材料阻尼系数，均为输入参数。*KE*为单元的刚度矩阵。



式中：为模态阻尼在频域下表现形式，为对角线矩阵，用于记录各频率下的模态阻尼，一般可通过模态试验测量得到。*C4*为模态阻尼在频域下表现形式，可通过最小二乘法求解式得到，由于模态阻尼一般仅在采用模态法中才使用到，因此很少计算*C4*。

针对模态叠加法截断模态*m*带来截断误差，可采相应的修正方法来弥补，具体过程与上述瞬态分析中一致，此处不在赘述。

# 3 关键字定义

输入文件(.dat)文件由三部分组成：关键字、数据和注释。

关键字以\*开始，后接大写字母；数据部分紧接其关键字定义的下方，数据间以空格或Tab分隔；

注释部分以#开始，表示注释该一整行。注意注释部分不要出现在关键字与其数据部分之间。各关键字之间可插入空行。

## 3.1 \*TITLE

该关键字为该仿真分析的名字，无实际意义，非必要关键字。

\* TITLE

[TitleName]

注： TitleName： string类型，分析的题目，单词间不可以空格隔开。

## 3.2 \*NODE

该关键字为节点信息输入，必要关键字。

\*NODE [nNODE]

[NodeID] [NodeCoord1] [NodeCoord2] [NodeCoord3]

注： nNODE： int类型（>0），总节点数；

NodeID： int类型（>0），节点编号；

NodeCoord1： double类型，节点的X坐标；

NodeCoord2： double类型，节点的Y坐标；

NodeCoord3： double类型，节点的Z坐标。

## 3.3 \*ELEM

该关键字为单元信息输入，必要关键字。

\*ELEM [nElem]

[ElemID] [ETypeID] [MatID] [ENodeID1-8]

注： nElem： int类型（>0），总单元数；

ElemID： int类型（>0），单元编号；

ETypeID： int类型（>0），单元类型编号，指向关键字\*ETYPE；

MatID： int类型（>0），材料类型编号，指向关键字\*MAT；

ENodeID1-8： int类型（>0），单元的8个节点编号。（顺序不可随意改变）

需要说明的是，单元类型编号和材料类型编号要从1开始逐渐递增。

## 3.4 \*ETYPE

该关键字为单元类型输入，必要关键字。

\*ETYPE [nETYPE]

[ETypeID] [ElemType] [nENode]

注： nETYPE： int类型（>0），总单元类型数；

ETypeID： int类型（>0），单元类型编号；

ElemType： string类型，单元类型标识；

=“SOLID45”：实体单元45（低阶单元）；

=“SHELL63”：壳单元63（当前未支持）；

nENode： string类型，单元的节点数量标识；

=“TETRA4”：4节点四面体；

=“PYRAM5”：5节点金字塔（当前未支持）；

=“PENTA6”：6节点三棱柱；

=“HEXA8”：8节点六面体；

需要说明的是，单元类型编号要从1开始逐渐递增。

## 3.5 \*MAT

该关键字为材料类型输入，必要关键字。

\*MAT [nMAT]

[MatID] [MatType] [EX] [MU] [DENS] [GE]

注： nMAT： int类型（>0），总材料类型数；

MatID： int类型（>0），材料类型编号；

MatType： string类型，材料类型标识；

=“ISO”：各项同性材料；

=“ANISO”：各项异性材料（当前未支持）；

EX： double类型（>0），材料的弹性模量；

MU： double类型（>0），材料的泊松比；

DENS： double类型（>0），材料的密度;

GE： double类型（>=0），材料阻尼(单元结构阻尼)。

需要说明的是，材料类型编号要从1开始逐渐递增。

## 3.6 \*NSETS

该关键字为节点集合输入，必要关键字。

\*NSETS [nNSETS]

[NSetsID] [nSetsNode]

[NodeID1-8] (only 8 data for one line)

...

注： nNSETS： int类型（>0），总节点集合数；

NSetsID： string类型，节点集合名称；

nSetsNode： int类型（>0），该节点集合的节点数量；

[NodeID1-8] ： int类型（>0），该节点集合的节点编号，每行有8个节点编号。

需要说明的是，节点集合名称可由英文字母、下划线和数字组成。

## 3.6 \*CURVE

该关键字为曲线数据输入，非必要关键字。

\* CURVE [nCurve]

[CurveID] [nCurveData]

[X] [Y]

...

注： nCurve： int类型（>0），总曲线数；

CurveID： string类型，曲线名称；

nCurveData： int类型（>0），该曲线的数据行数；

需要说明的是，曲线名称可由英文字母、下划线和数字组成，曲线名称被使用时，要用两个%表示，即%CurveID%。

## 3.7 \*ANTYPE

该关键字为分析类型输入，必要关键字。

\*ANTYPE

[AnTypeID]

注： AnTypeID： string类型，分析类型编号；

AnTypeID =“STA”： 静力学分析(STA)；

AnTypeID =“MOA”： 模态分析(MOA)；

AnTypeID =“DTA”： 直接瞬态分析(DTA)；

AnTypeID =“MTA”： 模态瞬态分析(MTA)；

AnTypeID =“DFA”： 直接频响分析(DFA)；

AnTypeID =“MFA”： 模态频响分析(MFA)。

## 3.8 \*EQUSLV

该关键字为方程组求解器类型输入，非必要关键字。

\*EQUSLV

[SolverType]

注： SolverType: string类型，分析类型编号;

SolverType=“PDS”：mkl\_pardiso求解器；

SolverType=“MPS”：mumps求解器，默认值。

## 3.9 \*EIGSLV

该关键字为特征值求解器类型输入，非必要关键字。

\*EIGSLV

[SolverType]

注： SolverType: string类型，分析类型编号;

SolverType=“FST”：mkl\_feast求解器；

SolverType=“APK”：arpack求解器，默认值。

## 3.10 \*FORCE

该关键字为节点力载荷的输入，非必要关键字。

\*FORCE [nFORCE]

[NSetsID] [DofId] [Value1] [Value2]

注： nFORCE： int类型（>0），节点力载荷数；

NSetsID： string类型，加载的节点集合名称；

DofId： string类型，加载的节点自由度编号；

=“FX”表示*x*方向节点力；

=“FY”表示*y*方向节点力；

=“FZ”表示*z*方向节点力；

=“MX”表示*rotx*方向节点力；

=“MY”表示*rotx*方向节点力；

=“MZ”表示*rotx*方向节点力；

Value1： int/double类型，表示节点力载荷的实部。

若为string类型，则表示节点力载荷实部的曲线ID（动力学分析）。

若为double类型（输入中必须含有小数点），则表示节点力载荷实部的具体值（静动力学分析）。

Value2： int/double类型，表示节点力载荷的虚部（主要为频响分析等）。

若为string类型，则表示节点力载荷虚部的曲线ID。

若为double类型（输入中必须含有小数点），则表示节点力载荷实部的具体值。

## 3.11 \*DISP

该关键字为给定位移载荷的输入，非必要关键字。

\*DISP [nDISP]

[NSetsID] [DofId] [Value1] [Value2]

注： nDISP： int类型（>0），给定位移载荷数；

NSetsID： string类型，加载的节点集合名称；

NDofId： string类型，加载的节点自由度编号；

=“UX”表示*x*方向给定位移；

=“UY”表示*y*方向给定位移；

=“UZ”表示*z*方向给定位移；

=“RX”表示*rotx*方向给定位移；

=“RY”表示*rotx*方向给定位移；

=“RZ”表示*rotx*方向给定位移；

Value1： string/double类型，表示给定位移载荷的实部。

若为string类型，则表示给定位移载荷实部的曲线名称（动力学分析中）。

若为double类型（输入中必须含有小数点），则表示给定位移载荷实部的具体值（=0.0就表示固定约束，静动力学均可）。

Value2： string/double类型，表示给定位移载荷的虚部（主要为频响分析等）。

若为string类型，则表示给定位移载荷虚部的曲线名称。

若为double类型（输入中必须含有小数点），则表示给定位移载荷实部的具体值（=0.0就表示固定约束，静动力学均可）。

## 3.12 \*VELO

该关键字为给定速度载荷的输入，非必要关键字。

\* VELO [nVELO]

[NSetsID] [NDofId] [Value1] [Value2]

注： nVELO： int类型（>0），给定速度载荷数；

NSetsID： string类型，加载的节点集合名称；

DofId： string类型，加载的节点自由度编号；

=“VX”表示*x*方向给定速度；

=“VY”表示*y*方向给定速度；

=“VZ”表示*z*方向给定速度；

=“RX”表示*rotx*方向给定速度；

=“RY”表示*rotx*方向给定速度；

=“RZ”表示*rotx*方向给定速度；

Value1： string/double类型，表示给定速度载荷的实部。

若为string类型，则表示给定速度载荷实部的曲线名称（动力学分析）。

若为double类型（输入中必须含有小数点），则表示给定速度载荷实部的具体值（动力学分析）。

Value2： string/double类型，表示给定速度载荷的虚部（主要为频响分析等）。

若为string类型，则表示给定速度载荷虚部的曲线名称。

若为double类型（输入中必须含有小数点），则表示给定速度载荷实部的具体值。

## 3.13 \*ACCE

该关键字为给定加速度载荷的输入，非必要关键字。

\* ACCE [nACCE]

[NSetsID] [DofId] [Value1] [Value2]

注： nVELO： int类型（>0），给定加速度载荷数；

NSetsID： string类型，加载的节点集合名称；

DofId： string类型，加载的节点自由度编号；

=“AX”表示*x*方向给定加速度；

=“AY”表示*y*方向给定加速度；

=“AZ”表示*z*方向给定加速度；

=“RX”表示*rotx*方向给定加速度；

=“RY”表示*rotx*方向给定加速度；

=“RZ”表示*rotx*方向给定加速度；

Value1： string/double类型，表示给定加速度载荷的实部。

若为string类型，则表示给定加速度载荷实部的曲线名称（动力学分析）。

若为double类型（输入中必须含有小数点），则表示给定加速度载荷实部的具体值（动力学分析）。

Value2： string/double类型，表示给定加速度载荷的虚部（主要为频响分析等）。

若为string类型，则表示给定加速度载荷虚部的曲线名称。

若为double类型（输入中必须含有小数点），则表示给定加速度载荷实部的具体值。

## 3.14 \*TIME

该关键字为瞬态分析的时间步长输入，对某具体分析类型是必要关键字。

\*TIME

[StartTime] [EndTime] [TimeStep]

注： StartTime： doublel类型（>=0.0），起始时间；

EndTime： doublel类型（> StartTime），结束时间；

TimeStep： doublel类型（>0.0），时间步长。

## 3.15 \*FREQ

该关键字为频响分析的频率步长输入，对某具体分析类型是必要关键字。

\*FREQ

[StartFreq] [EndFreq] [FreqStep]

注： StartFreq： doublel类型（>=0.0），起始频率；

EndFreq： doublel类型（>StartFreq），结束频率；

FreqStep： doublel类型（>0.0），频率步长。

## 3.16\*TAINTM

该关键字为瞬态分析的积分方法输入，非必要关键字。

@TAINTM

[IntegralMethod]

注： IntegralMethod： string类型，时间积分方法；

=“NKM”：Newmark方法，默认值；

=“WLS”：Wilson方法；

=“HHT”：HHT方法。

## 3.17 \*TAINTP

该关键字为瞬态分析的积分方法的参数输入，非必要关键字。

\*TAINTP

[alpha] [delta] [alphaF] [alphaM] [theta]

注： alpha： double类型（>=0.0），Newmark/HHT方法的参数，默认值为0.25250625；

delta： double类型（>=0.0），Newmark/HHT方法的参数，默认值为0.505；

alphaF： double类型（>=0.0），HHT方法的参数，默认值为0.005；

alphaM： double类型（>=0.0），HHT方法的参数，默认值为0.0；

theta： double类型（>=0.0），Wilson方法的参数，默认值为1.4。

## 3.18 \*DAMP

该关键字为阻尼的输入，非必要关键字。

\*DAMP

[alphad] [betad] [Gf] [GEf] [W3] [W4]

注： alphad：doublel类型（>=0.0），瑞丽阻尼参数，默认值为0.0；

betad： doublel类型（>=0.0），瑞丽阻尼参数，默认值为0.0；

Gf： doublel类型（>=0.0），总体结构阻尼系数，默认值为0.0。

GEf： doublel类型（>=0.0），单元结构阻尼系数，默认值为0.0。

W3： doublel类型（>=0.0），总体结构阻尼对应的频率(只对于瞬态分析) ，默认值为0.0。

W4： doublel类型（>=0.0），总体结构阻尼对应的频率(只对于瞬态分析) ，默认值为0.0。

## 3.19 \*MDAMP

该关键字为模态阻尼的输入，非必要关键字。

\*MDAMP

[CurveID]

注： CurveID： string/double类型，模态阻尼对应的曲线名称，默认为无；

若为string类型，则表示模态阻尼对应的曲线名称。

若为double类型，则表示常值模态阻尼。

## 3.20 \*MODEXT

该关键字为模态分析的模态提取的输入，对某具体分析类型是必要关键字。

\*MODEXT

[nModes] [MinFreq] [MaxFreq]

注： nModes： int类型（>0），想要提取的模态阶数；

MinFreq： doublel类型（>0.0），模态频率范围最小值；

MaxFreq： doublel类型（> MinFreq），模态频率范围最大值。

## 3.21 \*LUMPM

该关键字为质量矩阵类型的输入，非必要关键字。

\*LUMPM

[IsLump]

注： IsLump： int类型，是否为集中质量；

=1： 集中质量；

=0： 一致质量，默认值。

## 3.22 \*GENEOUT

该关键字为通用输出(云图输出)的设置，非必要关键字，默认不输出任何云图。

\*GENEOUT nGENEOUT

[OutType] [OutInterval]

注： nGENEOUT: int类型，通用输出的数量；

OutType： string类型，通用输出类型；

=“DISP”：输出位移；

=“VELO”：输出速度；

=“ACCE”：输出加速度；

=“STRA”：输出应变；

=“STRE”：输出应力；

OutInterval： int类型（>=0），通用输出的间隔；

=0表示输出最后一步。

## 3.23 \*HISTOUT

该关键字为历程输出(曲线输出)的设置，非必要关键字，默认不输出任何节点数据。

\*HISTOUT nHISTOUT

[OutType] [OutInterval] [NSetsID]

注： nGENEOUT: int类型，历程输出的数量；

OutType： int类型，历程输出类型；

=“DISP”：输出位移；

=“VELO”：输出速度；

=“ACCE”：输出加速度；

=“STRA”：输出应变；

=“STRE”：输出应力；

OutInterval： int类型（>0），历史输出的间隔。

NSetsID： string类型，输出的节点集合名称。

## 3.24 \*END

该关键字为dat输入文件结束的标志，即\*END后面的内容不再读取。非必要关键字。