# 有限元理论基础及Abaqus内部实现方式研究系列45：约束关系（1）-统一形式

## Finite Element Theory and the Internal Implementation of Abaqus: Series 45 - Constraint Relationships (1) - Unified Form

SnowWave02 关注 Focus

更新于2024年3月25日 15:10 Updated on March 25, 2024, 15:10

浏览：2405 Views: 2405 评论：3 Comment: 3 收藏：7 Favorited: 7

（原创，转载请注明出处） (Original, please indicate the source for redistribution)

1 概述 1 Overview

本系列文章研究成熟的有限元理论基础及在商用有限元软件的实现方式，通过

This series of articles studies the mature finite element theory foundation and its implementation in commercial finite element software, through

（1） 基础理论 (1) Basic Theory

（2） 商软操作 (2) Commercial Software Operations

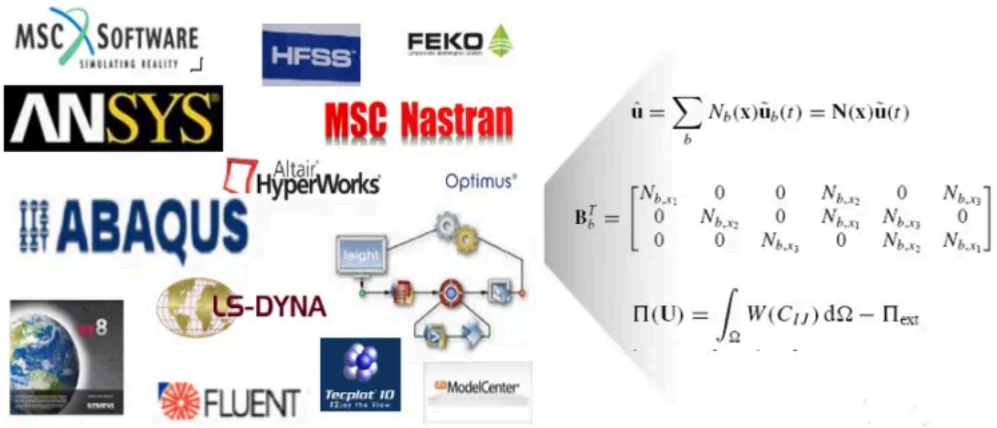
（3） 自编程序 (3) Self-written Programs

三者结合的方式将复杂繁琐的结构有限元理论通过简单直观的方式展现出来，同时深层次的学习有限元理论和商业软件的内部实现原理。

The combination of the three methods presents the complex and cumbersome structural finite element theory in a simple and intuitive way, while also deeply studying the internal implementation principles of finite element theory and commercial software.

有限元的理论发展了几十年已经相当成熟，商用有限元软件同样也是采用这些成熟的有限元理论，只是在实际应用过程中，商用CAE软件在传统的理论基础上会做相应的修正以解决工程中遇到的不同问题，且各家软件的修正方法都不一样，每个主流商用软件手册中都会注明各个单元的理论采用了哪种理论公式，但都只是提一下用什么方法修正，很多没有具体的实现公式。商用软件对外就是一个黑盒子，除了开发人员，使用人员只能在黑盒子外猜测内部实现方式。

The theoretical development of finite elements has matured over decades, and commercial finite element software also adopts these mature finite element theories. However, in the actual application process, commercial CAE software will make corresponding corrections on the basis of traditional theories to solve different problems encountered in engineering, and the correction methods of each software are different. Each mainstream commercial software manual specifies which theoretical formula each element uses, but only mentions the correction method, and many do not provide specific implementation formulas. Commercial software is a black box to the outside, and users can only guess the internal implementation methods from outside, except for developers.

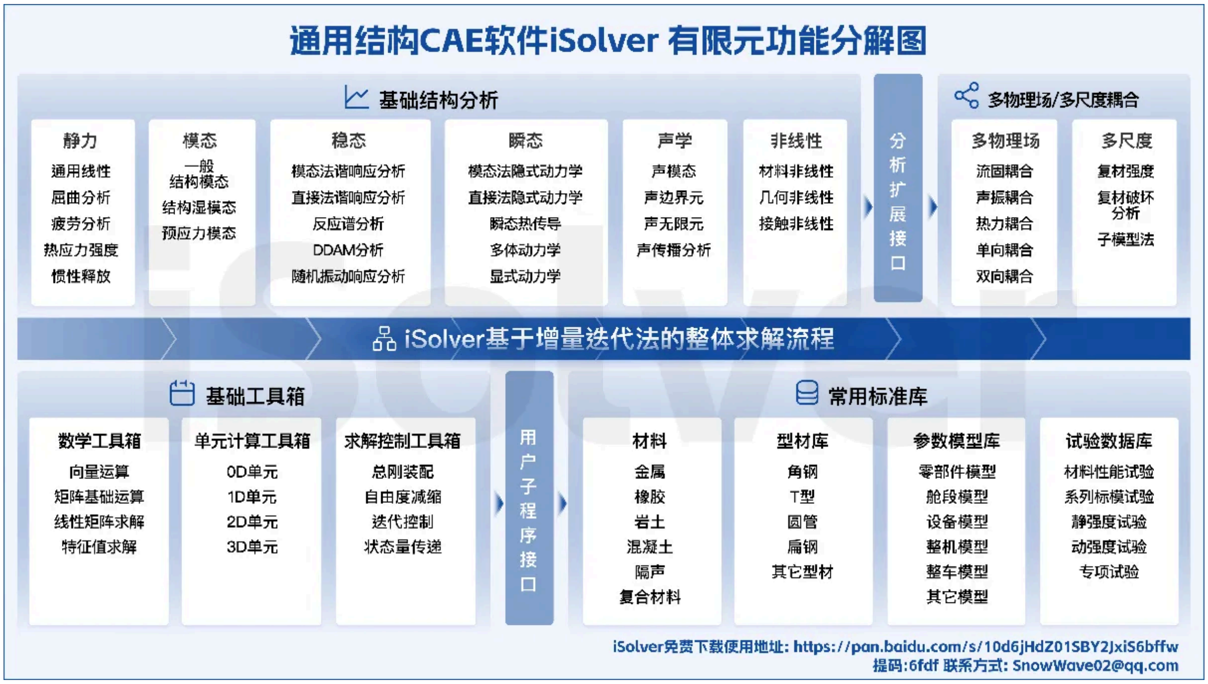

MSC Software
SIMULATING REALITY
HFSS
FEKO
ANSYS
MSC Nastran
Altair HyperWorks
Optimus
ABAQUS
Isight
LS-DYNA
8
FLUENT
Tecplot 10
ModelCenter
$\hat{\mathbf{u}} = \sum_b N_b(\mathbf{x})\tilde{\mathbf{u}}_b(t) = \mathbf{N}(\mathbf{x})\tilde{\mathbf{u}}(t)$
$\mathbf{B}_b^T = \begin{bmatrix} N_{b,x_1} & 0 & 0 & N_{b,x_2} & 0 & N_{b,x_3} \\ 0 & N_{b,x_2} & 0 & N_{b,x_1} & N_{b,x_3} & 0 \\ 0 & 0 & N_{b,x_3} & 0 & N_{b,x_2} & N_{b,x_1} \end{bmatrix}$
$\Pi(\mathbf{U}) = \int_\Omega W(C_{IJ})\,d\Omega - \Pi_{ext}$

一方面我们查阅各个主流商用软件的理论手册并通过进行大量的资料查阅猜测内部修正方法，另一方面我们自己编程实现结构有限元软件iSolver，通过自研CAE软件和商软的结果比较来验证我们的猜测，如同管中窥豹一般来研究的修正方法，从而猜测商用有限元软件的内部计算方法。我们关注CAE中的结构有限元，所以主要选择了商用结构有限元软件中文档相对较完备的Abaqus来研究内部实现方式，同时对某些问题也会涉及其它的Nastran/Ansys等商软。为了理解方便有很多问题在数学上其实并不严谨，同时由于水平有限可能有许多的理论错误，欢迎交流讨论，也期待有更多的合作机会。iSolver包括完整的前后处理和有限元求解器，功能如下，有兴趣可直接在下面网址下载：

On one hand, we consult the theoretical manuals of various mainstream commercial software and guess the internal correction methods through a large amount of literature review. On the other hand, we ourselves program the structural finite element software iSolver, verify our guesses through comparisons between our self-developed CAE software and commercial software, and study the correction methods as if we were peering through a bamboo tube to understand the internal calculation methods of commercial finite element software. Since we focus on structural finite elements in CAE, we mainly choose the commercial structural finite element software Abaqus, which has relatively complete documentation, to study its internal implementation methods, and will also involve other commercial software such as Nastran/Ansys for certain issues. Many problems are not mathematically rigorous for the sake of convenience in understanding, and due to our limited level, there may be many theoretical errors. We welcome communication and discussion and look forward to more cooperation opportunities. iSolver includes complete pre- and post-processing and a finite element solver, with the following functions. Those who are interested can directly download it from the following website:

百度网盘链接: https://pan.baidu.com/s/10d6jHdZ01SBY2JxiS6bffw 提取码: 6fdf

Baidu Pan link: https://pan.baidu.com/s/10d6jHdZ01SBY2JxiS6bffw Password: 6fdf



通用结构CAE软件iSolver 有限元功能分解图

基础结构分析

静力
通用线性
屈曲分析
疲劳分析
热应力强度
惯性释放

模态
一般结构模态
结构湿模态
预应力模态

稳态
模态法谐响应分析
直接法谐响应分析
反应谱分析
DDAM分析
随机振动响应分析

瞬态
模态法隐式动力学
直接法隐式动力学
瞬态热传导
多体动力学
显式动力学

声学
声模态
声边界元
声无限元
声传播分析

非线性
材料非线性
几何非线性
接触非线性

分析扩展接口

多物理场/多尺度耦合

多物理场
流固耦合
声振耦合
热力耦合
单向耦合
双向耦合

多尺度
复材强度
复材破坏分析
子模型法

iSolver基于增量迭代法的整体求解流程

基础工具箱

数学工具箱
向量运算
矩阵基础运算
线性矩阵求解
特征值求解

单元计算工具箱
0D单元
1D单元
2D单元
3D单元

求解控制工具箱
总刚装配
自由度减缩
迭代控制
状态量传递

用户子程序接口

常用标准库

材料
金属
橡胶
岩土
混凝土
隔声
复合材料

型材库
角钢
T型
圆管
扁钢
其它型材

参数模型库
零部件模型
舱段模型
设备模型
整机模型
整车模型
其它模型

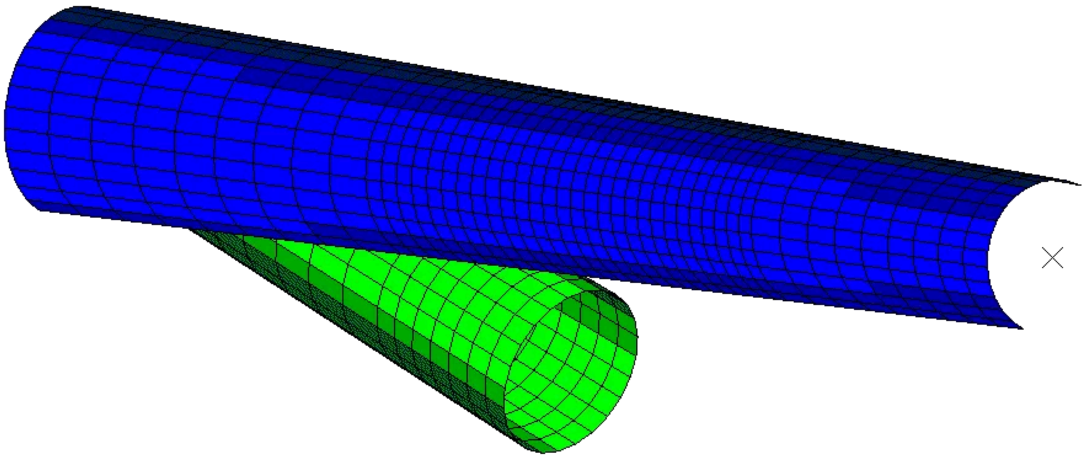
试验数据库
材料性能试验
系列标模试验
静强度试验
动强度试验
专项试验

iSolver免费下载使用地址：https://pan.baidu.com/s/10d6jHdZ01SBY2JxiS6bffw
提码:6fdf 联系方式：SnowWave02@qq.com

2 约束关系 2 Constraint Relationship

我们在系列文章第35章介绍了接触分析及常用的基本算法。现在Abaqus、LS-DYNA、Ansys等结构商软都说可以处理复杂的上万零部件接触的整车、整机等模型仿真，没做过实际的这种仿真分析，很好奇，接触分析算法往往涉及大变形、边界不连续，只要输入条件或者算法稍微变化一些，两个零部件算出来的接触结果就可能差异很大，更不用说上万个零部件的接触结果了，对这种大规模组装模型的仿真结果不知如何来判断它的可靠性，像普通的只校核一下材料的应力还是看一下动画是否和试验一致？毕竟仿真只有简单的标准来判断结果的正确性才能在企业中起到真正辅助设计的作用，如果你恰好做过，不妨也简单介绍一下你的经验。对自研CAE软件开发者来说，自研结构CAE软件是否真的要和商软去比拼接触等复杂算法还是花更多时间在精雕细琢那些常用功能上，这也是开发者需要慎重考虑的问题，而且很多自主CAE软件连常规线性问题都算的不对，或者都没法用鼠标稳定的走完那些材料、属性、边界、加载等流程，用户又怎么会相信你能算对接触这种复杂问题的？

We introduced contact analysis and common basic algorithms in Chapter 35 of our series of articles. Now, structural software such as Abaqus, LS-DYNA, and Ansys claim that they can handle complex simulations of whole vehicles and whole machines with tens of thousands of parts. I am very curious about this. Contact analysis algorithms often involve large deformations and boundary discontinuities. Even if the input conditions or algorithms are slightly changed, the contact results of two parts may differ greatly. Not to mention the contact results of tens of thousands of parts. How can we judge the reliability of the simulation results of such large-scale assembly models? Is it just to check the material stress or to see if the animation is consistent with the test? After all, only simple standards can be used to judge the correctness of the simulation results, which can truly assist in design in enterprises. If you have done it, please also briefly introduce your experience. For self-developed CAE software developers, is it really necessary to compete with commercial software in terms of complex algorithms such as contact, or should more time be spent on refining common functions? This is also a problem that developers need to consider carefully. Moreover, many self-developed CAE software cannot even solve routine linear problems correctly, or users cannot even complete the processes of materials, properties, boundaries, and loading with the mouse. How can users believe that you can solve complex problems like contact?



不管怎么样，从有限元实现的角度来讲，如果想做真正实际工程中的接触分析，那么首先需要去研究约束关系，接触分析在有限元中也仅是约束关系的一种。有限元中的约束很多场景大家用的是边界中的简支、固支等约束，但从更广泛的角度上讲，只要表示一个节点的某个自由度依赖于其它的节点自由度或者取某个特定值，就可以称为约束关系。只不过对固支、简支等直接自由度=0，在有限元中直接减缩刚度阵就行，很容易求，但对节点自由度相互依赖的约束关系就比较复杂了。约束关系主要有两类。

In any case, from the perspective of finite element implementation, if you want to do actual contact analysis in engineering, you first need to study constraint relationships. Contact analysis in finite elements is just one type of constraint. In many finite element scenarios, people use simple support and fixed support constraints in the boundary. However, from a broader perspective, as long as a node's degree of freedom depends on another node's degree of freedom or takes a specific value, it can be called a constraint relationship. It's just that for fixed support, simple support, and other direct degrees of freedom = 0, it is easy to reduce the stiffness matrix directly in finite elements, which is easy to calculate. But for constraints where node degrees of freedom are interdependent, it is more complex. There are mainly two types of constraint relationships.

（1） 一类是MPC点之间的约束。Nastran的MPC的灵活度要远远超过Abaqus，Nastran的主节点可以选择123自由度，也可以对每个从节点设置不同的自由度，还能主节点和从节点互相包含，Abaqus更多的是只负责80%的常用应用场景，复杂功能让你编子程序，但事实上一线仿真工程师又有多少人愿意编子程序呢？这种做法导致虽然Abaqus无论从用户体验、非线性还是商业化都比Nastran好很多，但很多线性的工程复杂问题还是没法替代

Nastran。

(1) One type is the constraint between MPC points. The flexibility of Nastran's MPC is much greater than that of Abaqus. Nastran's master node can choose 123 degrees of freedom, and can also set different degrees of freedom for each slave node, as well as have the master node and slave node mutually inclusive. Abaqus is more focused on handling 80% of common application scenarios, and complex functions require writing subroutines, but in fact, how many simulation engineers on the front line are willing to write subroutines? This approach leads to the fact that although Abaqus is much better than Nastran in terms of user experience, nonlinear behavior, and commercialization, many linear engineering complex problems still cannot be replaced by Nastran.

（2） 另一类是Contact、Tie等的面之间的约束关系。在这方面Abaqus要明显强于Nastran了。

Another type is the constraint relationship between surfaces such as Contact, Tie, etc. In this aspect, Abaqus is significantly superior to Nastran.

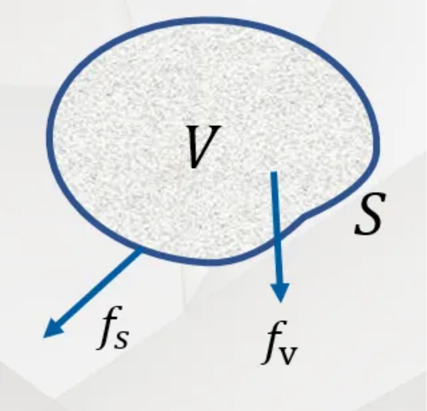
我们将用统一的公式来求解这两类关系，同时也从软件实现层面说明一下针对这两类情况的各自差异。分几篇文章来介绍约束关系，本篇是约束关系（1）-统一形式，既然接触仅是约束关系的一种，那么MPC、Tie、接触等的求解过程也是很类似的，这里将介绍一下这些约束关系如何表达为统一形式。

We will solve these two types of relationships using a unified formula and also explain the respective differences from the software implementation perspective. This series will introduce constraint relationships in several articles, and this article is the first part - Unified Form. Since contact is just one type of constraint relationship, the solution process for MPC, Tie, and contact is also very similar. Here, we will introduce how these constraint relationships are expressed in a unified form.

3 统一形式的约束关系 3 Unified Form of Constraint Relationships

在没有约束关系时，如下图情况，物体在体外力和面外力作用下变化。

In the case without constraint relationships, as shown in the figure below, the object changes under the action of external forces and surface forces.



有限元方程按照虚功原理求解，在物理上可解释能量守恒原理，即在某一个时刻点，假定在外力作用下有个虚拟的位移，那么外力在虚拟位移下做的虚功=内部应变能的变化相同。

The finite element equations are solved according to the principle of virtual work, which can physically explain the principle of energy conservation, that is, at a certain moment, assuming there is a virtual displacement under the action of external forces, the virtual work done by the external force under the virtual displacement is equal to the change in internal strain energy.

$$\delta \Pi_0 = \int_v {}^{t+\Delta t}_{t}\sigma * \delta {}^{t+\Delta t}_{t}\varepsilon \ dV - \int_v \rho {}^{t+\Delta t}_{\square} f_v * \delta u \, dV - \int_S {}^{t+\Delta t}_{\square} f_s * \delta u \, dS = 0$$

虚功原理中的每项都表示各自区域在虚位移下的能量变换

Each term in the virtual work principle represents the energy transformation of each region under virtual displacement

（1） fv是每单位体积内的力，外力fv和位移相乘表示单位体积内的虚功，所以对体积积分

(1) Fv is the force per unit volume, and the product of the external force Fv and displacement represents the virtual work per unit volume, so the volume integral

（2） fs是每单位面积上的力，外力fv和位移相乘表示单位面积上的虚功，所以对面积积分，推论就是最后一项应力和应变是单位体积内的内能，所以对体积积分。

(2) Fs is the force per unit area, and the product of the external force Fv and displacement represents the virtual work per unit area, so when integrating over the area, the conclusion is that the last term, stress and strain, represents the internal energy per unit volume, so it is integrated over the volume.

当存在约束关系时，在能量中加入约束关系相关的一项：

When constraint relationships exist, an additional term related to the constraint relationships is added to the energy.

$$\Pi_2 = \Pi_0(u_i, \ u_j) + \Pi_1(u_i, \ u_j, \ \lambda_{ij})$$

显然的单位也是该约束关系所在区域在虚位移下的能量单位。

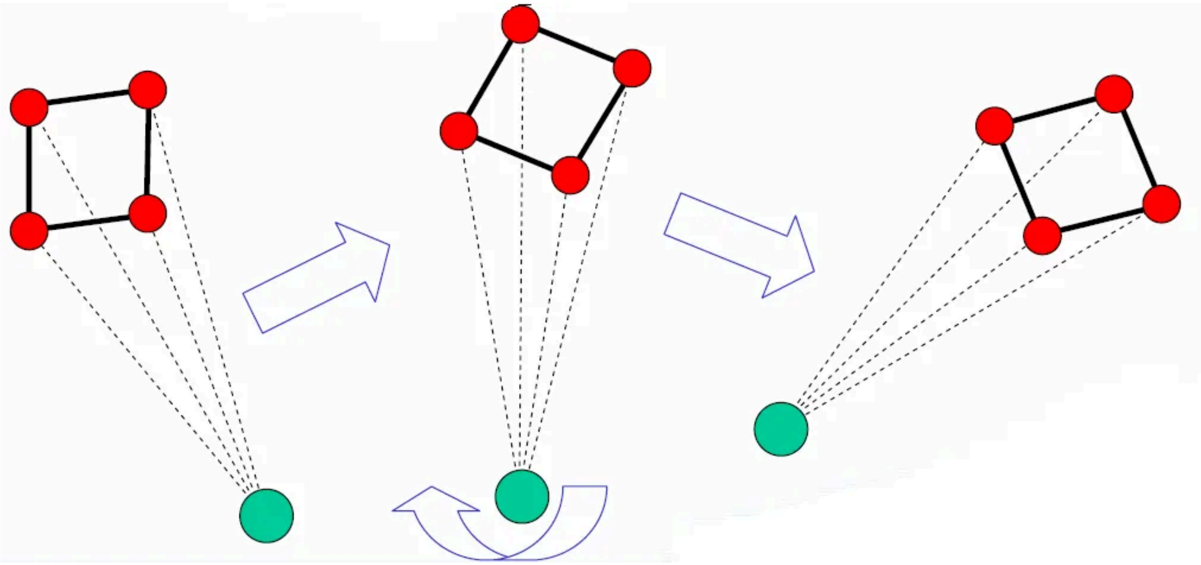
The unit is also the energy unit per unit area under virtual displacement in the region where the constraint relationship exists.

约束关系在有限元中可以分为两大类： Constraint relationships in finite elements can be divided into two major categories:

3.1 点之间的约束关系 3.1 Constraint Relationship Between Points

（1）点之间约束关系，最常见的是节点之间的刚性连接，Nastran中称为RBE2，在Abaqus或者iSolver中称为Kinematic Coupling，此时可以认为Master节点和Slave节点之间焊死在了一个刚性无穷大的直杆上。在实际情况中，Slave节点之间没有相对位移，但由于计算时很多时候默认为小位移，反而导致Slave节点之间是有相对位移的。

(1) The most common constraint relationship between points is the rigid connection between nodes, known as RBE2 in Nastran, and Kinematic Coupling in Abaqus or iSolver. At this point, it can be considered that the Master node and Slave node are rigidly connected to an infinitely rigid rod. In actual situations, there is no relative displacement between Slave nodes, but due to the fact that small displacements are often assumed in calculations, it may result in relative displacements between Slave nodes.



此时： At this moment:

$$h(\boldsymbol{u}_m, \boldsymbol{u}_s) = \boldsymbol{u}_s - (\boldsymbol{u}_m + \emptyset_m \times \vec{r})$$

其中， Among which,

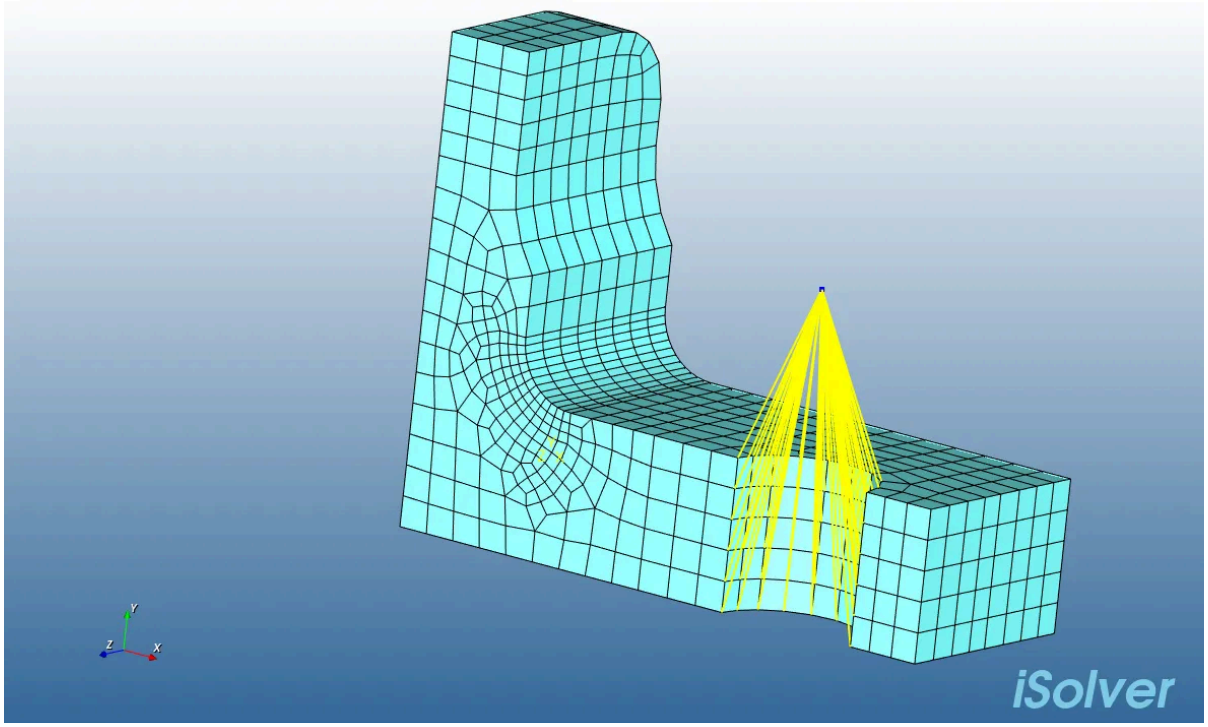
$$\emptyset_m$$

为Master节点的旋转向量，为从Slave节点到Master节点的向量。

The rotational vector for the Master node, which is the vector from the Slave node to the Master node.

这种节点之间的约束关系还有一种常见情况就是节点间的分布耦合连接RBE3，将施加在Master节点上的力和力矩按照加权系数分配到Slave节点上，从而实现载荷在单元间的传递，此时就避免了RBE2太刚的问题，典型应用场景譬如模拟圆管对壁面的压力作用。

There is also a common case of constraint relationship between nodes, which is the distributed coupling connection RBE3 between nodes. The force and torque applied to the Master node are distributed to the Slave node according to the weighting coefficient, thus realizing the transmission of load between elements, avoiding the issue of RBE2 being too rigid. Typical application scenarios include simulating the pressure exerted by a circular tube on a wall surface.



分布耦合连接是将Master节点的力和力矩按某种规则分配到Slave节点，保证力和力矩分别相等，即：

Distributed coupling connection is the distribution of the force and torque of the Master node to the Slave node according to certain rules, ensuring that the force and torque are equal respectively, i.e.:

$$\sum_{s} F_s = F_m \hookleftarrow$$

$$\sum_{s} (x_s \times F_s + M_s) = x_m \times F_m + M_m \hookleftarrow$$

上述对六自由度的Master节点来说，一共只有六个方程，但每个Slave节点都有6个位置量，所以对多个Slave节点情况解不唯一，Abaqus、iSolver、Nastran等都有各自不同的分配原则，一般都是假定Slave节点不再存在Master节点分配过来的弯矩Ms，同时，Fs等比例分配。

For a six-degree-of-freedom Master node, there are only six equations in total, but each Slave node has six position variables, so the solution is not unique for multiple Slave nodes. Abaqus, iSolver, Nastran, and others have their own allocation principles, which are generally to assume that the Slave nodes do not have the bending moments Ms allocated from the Master node, and the forces Fs are proportionally distributed.

对这种点的约束关系，虚功原理中增加的能量项表示为：

For the constraint relationship of such points, the additional energy term in the virtual work principle is expressed as:

$$\Pi_1(u_i, u_j, \lambda_{ij}) = \lambda_{ms} * \mathrm{h}(u_m, u_s)↵$$

注意，因为是点之间的约束，所以不需要对面或者体积分，类似质量点对质量阵的贡献或者集中力对载荷向量的贡献时也不需要积分一样。

Note that since it is a constraint between points, there is no need for surface or volume integrals, similar to the case where integration is not required for the contribution of mass points to the mass matrix or concentrated forces to the load vector.

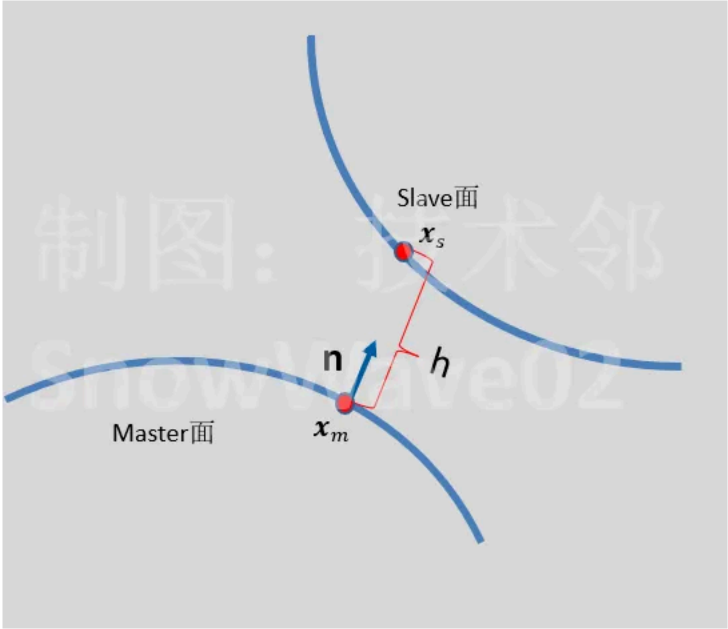
### 3.2 面之间的约束关系 3.2 Constraint relationship between surfaces

面之间的约束关系，最常见的就是Master面和Slave面/点的接触连接关系。

The most common constraint relationship between surfaces is the contact connection between Master surface and Slave surface/point.

在有限元软件中设置完接触关系对后，对Slave面上的任意一点xs，在Master面上寻找和xs距离最短的点，譬如xm点，此时xs和xm之间的距离为h，显然，Master面在xm点上的法向n指向xs点（要不然就不是最近邻点了）。也就是说slave面上任意一点在Master面上都有一个点来对应，这个点我们称为锚点，锚点的法向指向Slave点是一个必要条件（不是充分条件，因为有可能Master有两个锚点法向指向同一个xs点，譬如Master是个拐角的情况）。如下图所示。

After setting up the contact relationship in finite element software, for any point xs on the Slave surface, find the point on the Master surface that is closest to xs, for example, point xm. At this time, the distance between xs and xm is h. It is obvious that the normal n at point xm on the Master surface points towards xs (otherwise it wouldn't be the nearest neighbor point). That is to say, for any point on the Slave surface, there is a corresponding point on the Master surface, which we call the anchor point. The normal of the anchor point pointing towards the Slave point is a necessary condition (not a sufficient condition, because it is possible that the Master has two anchor points with normals pointing towards the same xs point, for example, when the Master is at a corner). As shown in the figure below.



此时距离(真实的距离肯定是正的，但我们这边为了方便取有向距离)

At this moment, the distance (the actual distance is definitely positive, but for convenience, we take the directed distance here)

$$\mathrm{h} = (\boldsymbol{x}_m - \boldsymbol{x}_s) * \boldsymbol{n} \hookleftarrow$$

设接触面法向压力为p，对硬接触，当两个面之间的距离h是0时，压力p>0。有距离h<0时，没有接触关系，此时压力p为0。

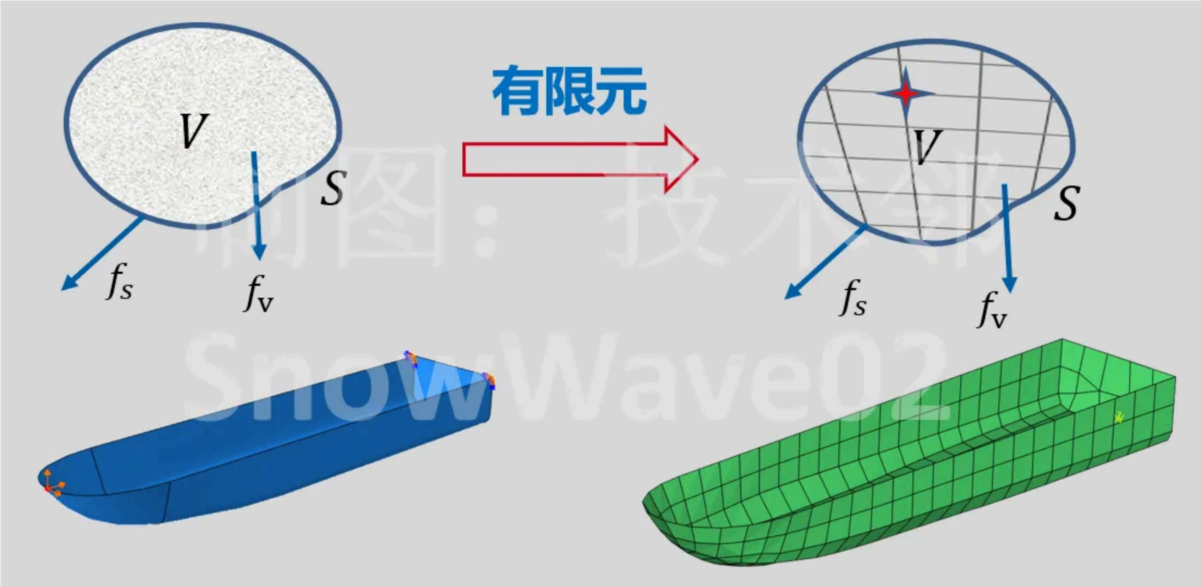
The normal pressure on the contact surface is denoted as p. For a hard contact, when the distance between the two surfaces, h, is 0, the pressure p is greater than 0. When there is a distance h less than 0, there is no contact relationship, and the pressure p is 0.

即约束方程为 The constraint equation is

$$\mathrm{hp} = \mathrm{p} * (\boldsymbol{x}_m - \boldsymbol{x}_s) * \boldsymbol{n} = 0 \hookleftarrow$$

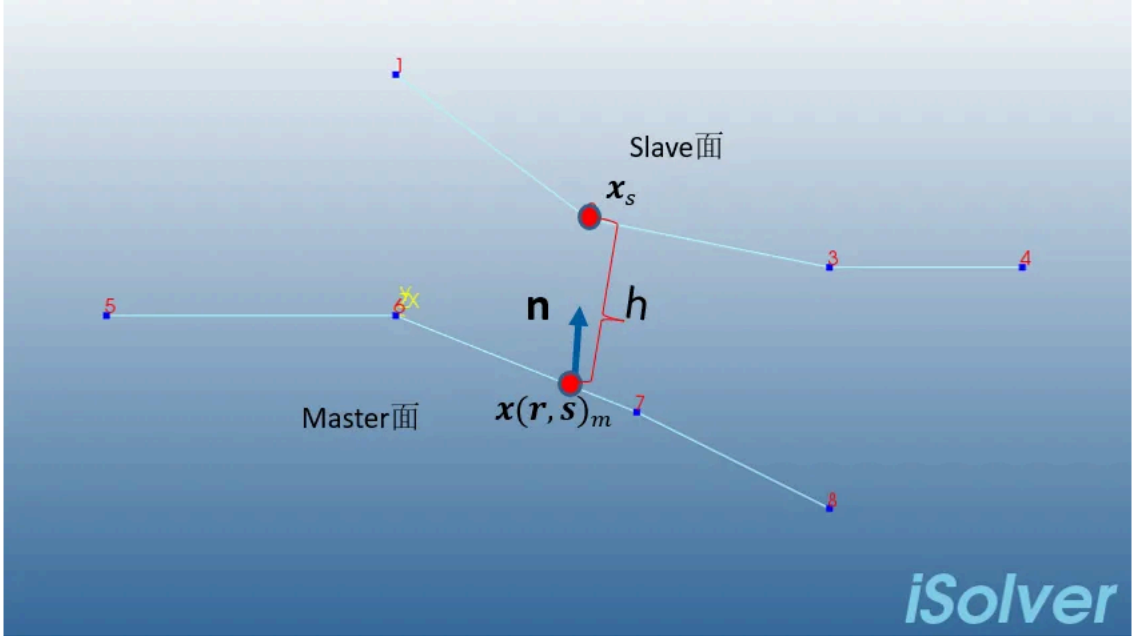
这是对面上的任意一点。计算机是无法处理面上所有点的问题的，所以也只能划分成有限单元，得到有限节点之间的关系。

This refers to any point on the surface. Computers cannot handle the problem of all points on the surface, so they can only be divided into finite elements to obtain the relationship between finite nodes.



划分完网格后，上述接触面上必然也是节点组成，如下：

After the mesh is divided, the above contact surface will also be composed of nodes, as follows:



对每个Slave节点xs，在Master接触面上寻找对应的锚点，因为Master面由不连续的节点组成，所以这个锚点很多情况都不在Master节点上。譬如上面图示，此时就是该点所在单元对锚点的插值，同样满足这个锚点的法向n(r,s)（也由所在单元的节点在r，s点插值得到）指向xs。

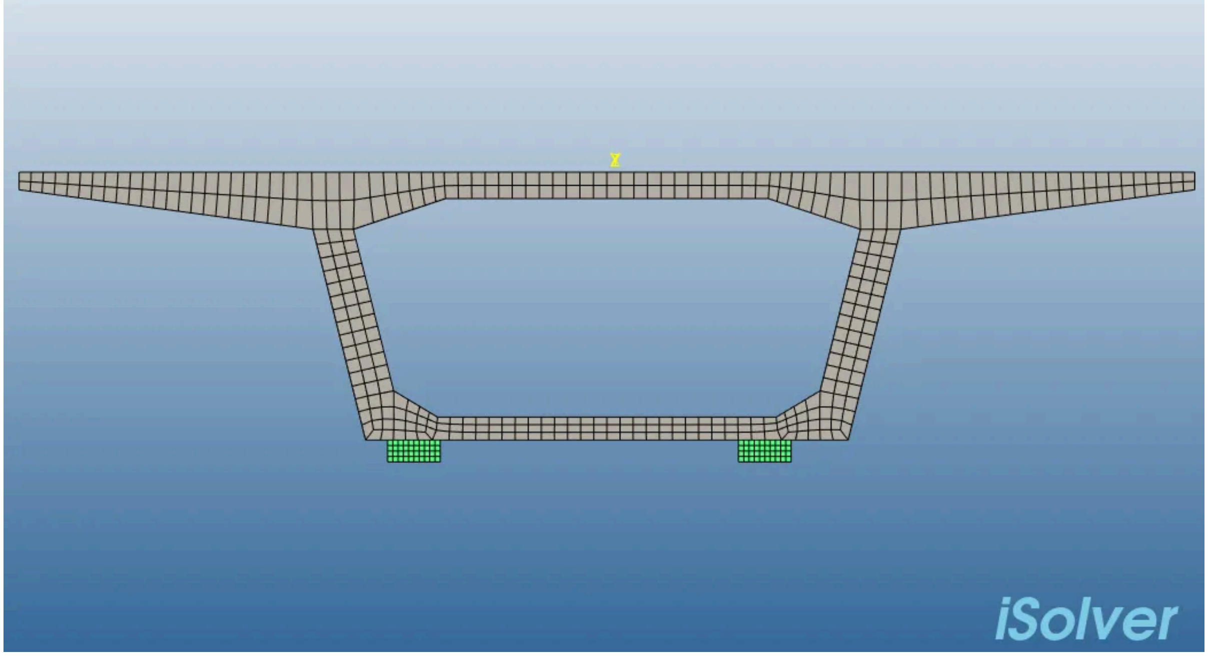
For each Slave node xs, find the corresponding anchor point on the Master contact surface. Since the Master surface is composed of discontinuous nodes, the anchor point is often not on the Master nodes. For example, as shown in the figure above, this is the interpolation of the anchor point by the element containing this point, which also satisfies the normal n(r,s) of this anchor point (also obtained by interpolation of the nodes of the element at the r, s point) pointing towards xs.

同样，此时的约束方程为： Similarly, the constraint equation at this time is:

$$\mathrm{hp} = \mathrm{p} * (\boldsymbol{x}(r,s)_m - \boldsymbol{x}_s) * \boldsymbol{n}(r,s) = 0$$

这种面面之间的一种特殊情况就是Master面和Slave面/点的粘贴连接关系，表示Slave面/点用胶水粘在Master面上，所以在Nastran中称为Glue，在Abaqus中称为Tie，一般用于两个不同网格之间边界耦合在一起，譬如下方的桥身和桥墩，实际上Slave面/点是焊死在Master面的，Slave面/点会随着Master面一起拉伸压缩移动，但Slave面/点在运动过程中不会脱落，最终反应到刚度阵依然是固定的Slave节点自由度和Master节点自由度之间的约束关系。

This is a special case of the bonding relationship between master and slave surfaces/points, indicating that the slave surface/point is glued to the master surface, hence it is called "Glue" in Nastran and "Tie" in Abaqus. It is generally used for boundary coupling between two different meshes, such as the bridge body and pier below. In fact, the slave surface/point is welded to the master surface, and the slave surface/point will move with the master surface, stretching and compressing. However, the slave surface/point will not fall off during the movement, and the final stiffness matrix still reflects the constraint relationship between the fixed slave node degrees of freedom and the master node degrees of freedom.

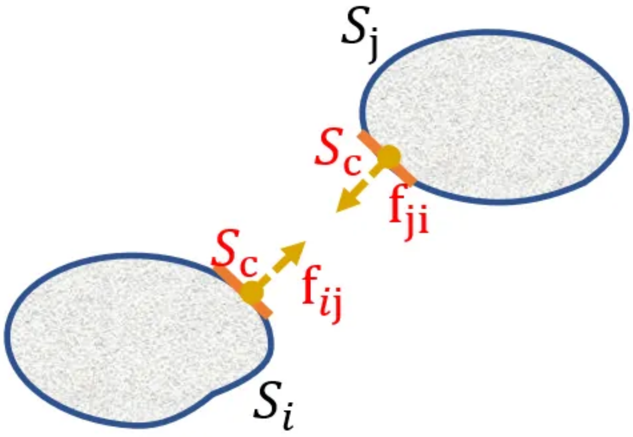


对这种面与面/点的约束关系，增加的能量项表示为：

The energy term added for this surface-to-surface/point constraint relationship is expressed as:

$$\Pi_1\left(u_i，\ u_j，\ \lambda_{ij}\right) = \int_{Sc} p_{ms} * \left(\boldsymbol{x}(r,s)_m - \boldsymbol{x}_s\right) * \boldsymbol{n}(r,s)\ dS$$

Sc为接触面。 Sc is the contact surface.



上式和节点的约束关系相比，可知pms就是Lagrange因子。

The above equation and the constraint relationship of the node show that pms is the Lagrange multiplier.

在实际三维接触分析中，接触力除了法向**n**的压力还有两个切向**v1**和**v2**的摩擦力。得到实际接触力由三个正交的力分量组成：

In practical three-dimensional contact analysis, the contact force consists of not only the normal pressure n but also two tangential friction forces v1 and v2. The actual contact force is composed of three orthogonal force components:

$$f_{ms} = p_{ms}\boldsymbol{n} + \tau_{ms1}\boldsymbol{v_1} + \tau_{ms2}\boldsymbol{v_2}$$

简单起见，我们假定不滑动，此时由于接触关系增加的能量项为：

For simplicity, we assume no sliding, and the additional energy term due to the contact relationship is:

$$\Pi_1(u_i，\ u_j，\ \lambda_{ij}) = \int_{Sc} (p_{ms}\boldsymbol{n} * \mathrm{h} + \tau_{ms1}\boldsymbol{v_1} * s_{\boldsymbol{1}} + \tau_{ms2}\boldsymbol{v_2} * s_{\boldsymbol{2}})\ dS$$

4 统一形式的约束关系 4 Unified form of constraint relationships

上述不同的仅是约束关系带来的积分范围和约束方程的个数和物理量。去掉所有的积分形式，我们可以统一写成如下形式:

The only differences are the integration range and the number of constraint equations and physical quantities brought by the constraint relationships. By removing all integral forms, we can uniformly write it in the following form:

$$\Pi_1(u_i，\ u_j，\ \lambda_{ij}) = {}^{t+\Delta t}_{\square}\lambda * {}^{t+\Delta t}_{\square}\mathrm{h}$$

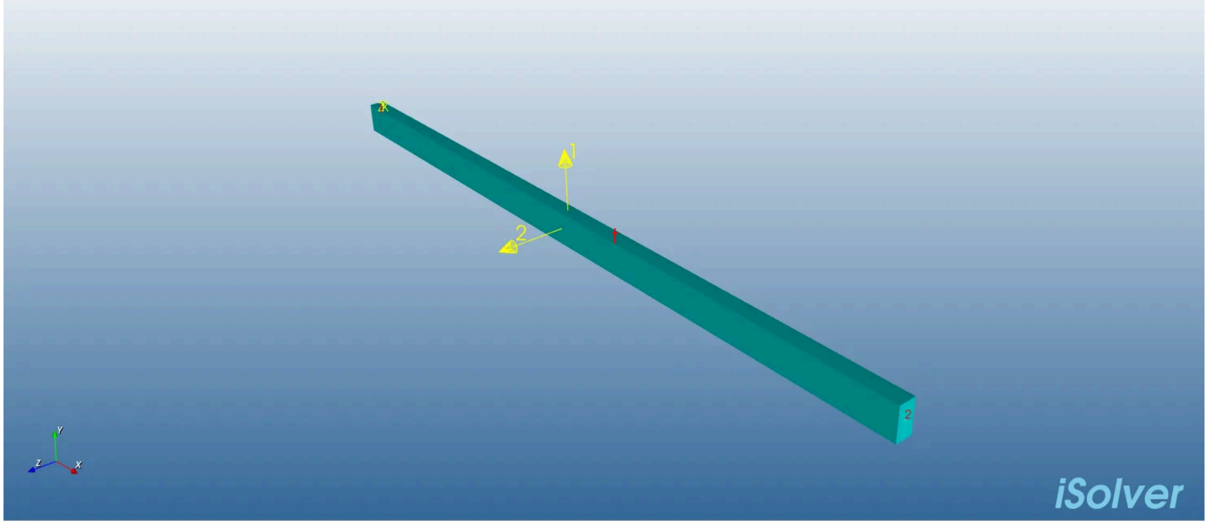
其中上面的正负不影响结果，仅仅和h的正负有关。 The positive and negative signs above do not affect the result; they are only related to the sign of h.

5 更大统一形式的有限元的约束关系 The constraint relationships of the unified form of finite elements

MPC、接触等都是某部分节点和另一部分节点之间的关系，那么有限元中梁、壳、体普通的单元是什么呢？从更大的统一形式来说，这些普通单元也仅仅是节点之间的约束关系而已，只不过这个关系可能更加复杂点。

MPC, contact, and so on are the relationships between certain nodes and other nodes. Then, what are the ordinary elements of finite elements such as beams, shells, and solids in finite element analysis? From a larger unified perspective, these ordinary elements are merely constraint relationships between nodes, although these relationships may be more complex.

以两节点梁单元为例。 Taking the two-node beam element as an example.

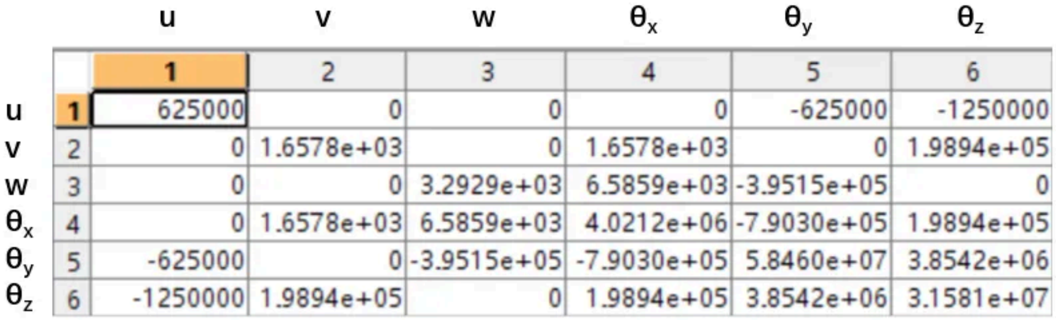


此时，每个节点6个自由度，那么单元刚度阵K为12X12的矩阵。K可以按6x6矩阵分块，每块小矩阵都比较类似，那么可以分成4块：

At this point, each node has 6 degrees of freedom, so the element stiffness matrix K is a 12x12 matrix. K can be divided into 6x6 matrix blocks, and each small matrix is quite similar, so it can be divided into 4 blocks:

（1）左上角为第一个节点自由度相关的6X6的小矩阵做研究对象。iSolver软件中内部打印如下

(1) The upper left corner is a 6x6 small matrix related to the first node's degrees of freedom, which is the subject of study. The internal printout in iSolver software is as follows



|  |  | u | v | w | $\theta_x$ | $\theta_y$ | $\theta_z$ |
|---|---|---|---|---|---|---|---|
|  |  | 1 | 2 | 3 | 4 | 5 | 6 |
| u | 1 | 625000 | 0 | 0 | 0 | -625000 | -1250000 |
| v | 2 | 0 | 1.6578e+03 | 0 | 1.6578e+03 | 0 | 1.9894e+05 |
| w | 3 | 0 | 0 | 3.2929e+03 | 6.5859e+03 | -3.9515e+05 | 0 |
| $\theta_x$ | 4 | 0 | 1.6578e+03 | 6.5859e+03 | 4.0212e+06 | -7.9030e+05 | 1.9894e+05 |
| $\theta_y$ | 5 | -625000 | 0 | -3.9515e+05 | -7.9030e+05 | 5.8460e+07 | 3.8542e+06 |
| $\theta_z$ | 6 | -1250000 | 1.9894e+05 | 0 | 1.9894e+05 | 3.8542e+06 | 3.1581e+07 |

K的任意一项Kij都是i和j两个自由度的乘积，其中对角元素为节点1和节点1自身的自由度的乘积，譬如K11就是u和u的乘积，而非对角元素都是耦合项，譬如K15就是u和θy的乘积。

Any element Kij of K is the product of the two degrees of freedom i and j, where the diagonal elements are the product of the degrees of freedom of node 1 with itself, for example, K11 is the product of u and u, rather than the off-diagonal elements, which are coupling terms, such as K15, which is the product of u and θy.

（2）右下角类似，只不过是节点2和节点2自身的关系。

(2) Similarly, the lower right corner represents the relationship between node 2 and itself.

（3）左下角和右上角类似，是节点1和节点2的关系。

(3) The lower left and upper right corners are similar, representing the relationship between node 1 and node 2.

所以，可以总结有限元的所有单元，包括MPC、接触等最终计算的都是这个单元所涉及点两两之间的关系，这个关系的最终形式都体现在刚度阵上。有限元简单来讲就是求各个点之间关系的公式，只不过有些公式简单，譬如两个点之间用线性弹簧，有些复杂，譬如几何大变形时两个节点之间或者接触时两个面之间的关系。

Therefore, all finite element units, including MPC and contact, ultimately calculate the pairwise relationships between the points involved in the unit, and the final form of this relationship is reflected in the stiffness matrix. In simple terms, finite element analysis is the formula for the relationship between points, with some formulas being simple, such as using a linear spring between two points, and some being complex, such as the relationship between two nodes or two surfaces when there is large geometric deformation or when in contact.

## 6 以往的系列文章 6 Previous series articles

6.1 ========第一阶段========

6.1 ====== Phase One ======

第一篇：S4壳单元刚度矩阵研究。 First article: Research on the stiffness matrix of S4 shell elements.

http://www.jishulink.com/content/post/338859

第二篇：S4壳单元质量矩阵研究。 Second article: Research on the mass matrix of S4 shell elements.

http://www.jishulink.com/content/post/343905

第三篇：S4壳单元的剪切自锁和沙漏控制。 Third article: Shear locking and hourglass control of S4 shell elements.

http://www.jishulink.com/content/post/350865

第四篇：非线性问题的求解。 Fourth article: Solution of nonlinear problems.

http://www.jishulink.com/content/post/360565

第五篇：单元正确性验证。 Fifth article: Element correctness verification.

https://www.jishulink.com/content/post/373743

第六篇：General梁单元的刚度矩阵。 Sixth article: Stiffness matrix of General beam elements.

https://www.jishulink.com/content/post/403932

第七篇：C3D8六面体单元的刚度矩阵。 Chapter 7: Stiffness Matrix of C3D8 Hexahedral Element.

https://www.jishulink.com/content/post/430177

第八篇：UMAT用户子程序开发步骤。 Chapter 8: Steps for UMAT User Subroutine Development.

https://www.jishulink.com/content/post/432848

第九篇：编写线性UMAT Step By Step。

The Ninth Article: Writing Linear UMAT Step by Step.

http://www.jishulink.com/content/post/440874

第十篇：耦合约束（Coupling constraints）的研究。

The Tenth Article: Research on Coupling Constraints (Coupling constraints).

https://www.jishulink.com/content/post/531029

6.2 ========第二阶段========

6.2 Second Phase ========

第十一篇：自主CAE开发实战经验第一阶段总结。 The Eleventh Article: Summary of the First Phase of Self-developed CAE Practical Experience.

http://www.jishulink.com/content/post/532475

第十二篇：几何梁单元的刚度矩阵。 The Twelfth Article: Stiffness Matrix of Geometric Beam Elements.

http://www.jishulink.com/content/post/534362

第十三篇：显式和隐式的区别。 The Thirteenth Article: Difference between Explicit and Implicit.

http://www.jishulink.com/content/post/537154

第十四篇：壳的应力方向。 Article 14: Shell Stress Direction.

https://www.jishulink.com/content/post/1189260

第十五篇：壳的剪切应力。 The Fifteenth Article: Shear Stress of Shells.

https://www.jishulink.com/content/post/1191641

第十六篇：Part、Instance与Assembly。

The Sixteenth Part: Part, Instance, and Assembly.

https://www.jishulink.com/content/post/1195061

第十七篇：几何非线性的物理含义。 The Seventeenth Part: Physical Meaning of Geometric Nonlinearity.

https://www.jishulink.com/content/post/1198459

第十八篇：几何非线性的应变。 The Eighteenth Part: Strain of Geometric Nonlinearity.

https://www.jishulink.com/content/post/1201375

第十九篇：Abaqus几何非线性的设置和后台。 The Nineteenth Part: Settings and Background of Abaqus Geometric Nonlinearity.

http://www.jishulink.com/content/post/1203064

第二十篇：UEL用户子程序开发步骤。 Chapter 20: Steps for UEL User Subroutine Development.

https://www.jishulink.com/content/post/1204261

6.3 ========第三阶段========

6.3 ========= Third Phase ========

第二十一篇：自主CAE开发实战经验第二阶段总结。

Twenty-first article: Summary of the Second Stage of Independent CAE Development Experience.

https://www.jishulink.com/content/post/1204970

第二十二篇：几何非线性的刚度矩阵求解。 Twenty-second article: Solution of Stiffness Matrix for Geometric Nonlinearity.

http://www.jishulink.com/content/post/1254435

第二十三篇：编写简单面内拉伸问题UEL Step By Step

Twenty-third article: Step by Step Guide to Writing a Simple In-Plane Tension Problem UEL

http://www.jishulink.com/content/post/1256835

第二十四篇：显式求解Step By Step。 24th article: Explicit Solving Step by Step.

https://www.jishulink.com/content/post/1261165

第二十五篇：显式分析的稳定时间增量。 第二十五篇：Explicit Analysis Stable Time Increment

http://www.jishulink.com/content/post/1263601

第二十六篇：编写线性VUMAT Step By Step。

第二十六篇：Linear VUMAT Step By Step Guide

https://www.jishulink.com/content/post/1266640

第二十七篇：Abaqus内部计算和显示的应变。 第二十七篇：Abaqus Internal Calculation and Display of Strain

https://www.jishulink.com/content/post/1273788

第二十八篇：几何非线性的T.L.和U.L.描述方法

第二十八篇：Description Methods for Geometric Nonlinearity T.L. and U.L.

https://www.jishulink.com/content/post/1282956

第二十九篇：几何非线性的T.L.和U.L.转换关系

Twenty-ninth article: The conversion relationship between T.L. and U.L. of geometric nonlinearity

https://www.jishulink.com/content/post/1286065

第三十篇：谐响应分析原理 Thirty-first article: Principles of Harmonic Response Analysis

https://www.jishulink.com/content/post/1290151

6.4 ========第四阶段========

6.4 ========= Fourth Phase ========

第三十一篇：自主CAE开发实战经验第三阶段总结 Thirty-second article: Summary of the Third Phase of Independent CAE Development Experience

https://www.jishulink.com/content/post/1294824

第三十二篇：谐响应分析算法 32nd article: Harmonic Response Analysis Algorithm

https://www.jishulink.com/content/post/1299983

第三十三篇：线性瞬态动力学 Chapter 33: Linear Transient Dynamics

https://www.jishulink.com/content/post/1302074

第三十四篇：非线性瞬态分析 Series 34: Nonlinear Transient Analysis

https://www.jishulink.com/content/post/1787283

第三十五篇：接触求解算法 Thirty-fifth Article: Contact Solution Algorithm https://www.jishulink.com/content/post/1792869

第三十六篇：DLOAD用户子程序开发步骤 Thirty-sixth Article: DLOAD User Subroutine Development Steps

https://www.jishulink.com/content/post/1826803

第三十七篇：梁单元差异（1）-理论基础 Thirty-seventh Article: Beam Element Differences (1) - Theoretical Foundation

https://jishulink.com/content/post/1872208

第三十八篇：梁单元差异（2）-梁截面方向 Thirty-eighth Article: Beam Element Differences (2) - Beam Section Direction

https://www.jishulink.com/content/post/1874628

第三十九篇：梁单元差异（3）-剪力和弯矩 The 39th Article: Beam Element Differences (3) - Shear and Bending Moments

**https://www.jishulink.com/content/post/1876013**

第四十篇：梁单元差异（4）-形心、剪心和偏置 The 40th Article: Beam Element Differences (4) - Centroid, Shear Center, and Offset

**https://www.jishulink.com/post/1888000**

6.5 ========第五阶段========

6.5 ========= Fifth Stage ========

第四十一篇：自主CAE开发实战经验第四阶段总结

The 41st Article: Summary of the Fourth Stage of Self-developed CAE Practical Experience

**https://www.jishulink.com/post/1905963**

第四十二篇：声学分析(1)-有限元 42nd article: Acoustic Analysis (1) - Finite Element Method

**https://www.jishulink.com/post/1912491**

第四十三篇：声学分析(2)-边界元 The forty-third article: Acoustic Analysis (2) - Boundary Element

**https://www.jishulink.com/post/1923936**

第四十四篇：声学分析(3)-湿模态 44th Article: Acoustic Analysis (3) - Wet Modal

**https://www.jishulink.com/post/1928692**

推荐阅读 Recommended Reading

**Abaqus、iSolver与Nastran梁单元差异...**

**转子旋转的周期性模型-水冷电机散热仿真 Periodic Model of Rotor...**

**网架设计之SFCAD第七节终极大招 Design of Grid Structure: The...**

**ansys经典半刚性脚手架分析（2） Ansys Classic Semi**

| | |
|---|---|
| SnowWave02 | 免费 Free |
| 技术邻小李 Technical Neighbor Xiao Li | ¥100 100 Yuan |
| 高工 Senior Engineer | ¥138 |
| 冷月 Cold Moon | ¥1 |