

有限元理论基础及 **Abaqus** 内部实现方式研究系列 6:

## General 梁单元的刚度矩阵

--如何编程实现和 **Abaqus** 的梁单元完全一致的刚度矩阵

V2018-0823



作者: SnowWave02

From [www.jishulink.com](http://www.jishulink.com)

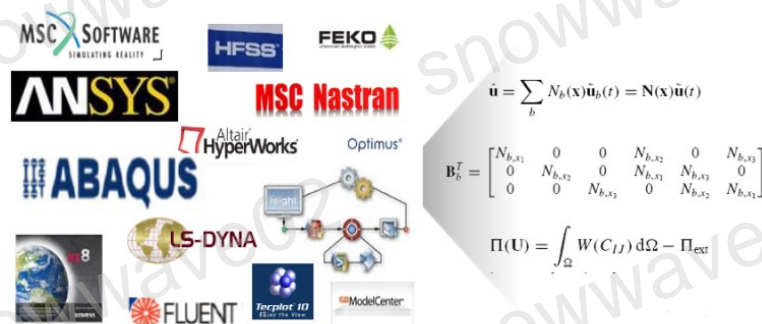
email: [snowwave02@qq.com](mailto:snowwave02@qq.com)

# 目 录

General 梁单元的刚度矩阵 .....	1
1 概述 .....	3
2 第六篇：General 梁单元的刚度矩阵 .....	4
2.1 梁单元的基本理论 .....	5
2.1.1 梁单元的受力方式分类 .....	5
2.1.2 梁单元的刚度矩阵基本理论 .....	10
2.2 Abaqus 的实现方式 .....	13
2.2.1 Abaqus 单元类型 .....	13
2.2.2 Abaqus 的 General 梁单元的刚度矩阵 .....	14
2.3 General 梁单元刚度矩阵的算例考核 .....	19
2.3.1 模型描述 .....	19
2.3.2 分析结果 .....	21
2.4 总结 .....	22
2.5 附录 .....	23
2.5.1 有限元的刚度矩阵基本理论 .....	23
2.5.2 自研有限元程序开发框架 iSolver 介绍 .....	24

# 1 概述

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。我们关注 CAE 中的结构有限元，所以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。



在实际应用过程中，商用 CAE 软件在传统的理论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，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，一方面我们查阅 Abaqus 软件手册得到修正方法的说明，另一方面我们自己编程实现简单的结构有限元求解器，通过自研求解器和 Abaqus 的结果比较结合理论手册如同管中窥豹一般来研究 Abaqus 的修正方法，从而猜测商用有限元软件的内部计算方法。为了理解方便有很多问题在数学上其实并不严谨，同时由于水平有限也可能有许多的理论错误，欢迎交流讨论。

以往的系列文章：

第一篇：**S4 壳单元刚度矩阵研究**。介绍 Abaqus 的 S4 刚度矩阵在普通厚壳理论上的修正。

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

第二篇：**S4 壳单元质量矩阵研究**。介绍 Abaqus 的 S4 和 Nastran 的 Quad4 单元的质量矩阵。

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

第三篇：**S4 壳单元的剪切自锁和沙漏控制**。介绍 Abaqus 的 S4 单元如何来消除剪切自锁以及 S4R 如何来抑制沙漏的。

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

第四篇：非线性问题的求解。介绍 Abaqus 在非线形分析中采用的数值计算的求解方法。

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

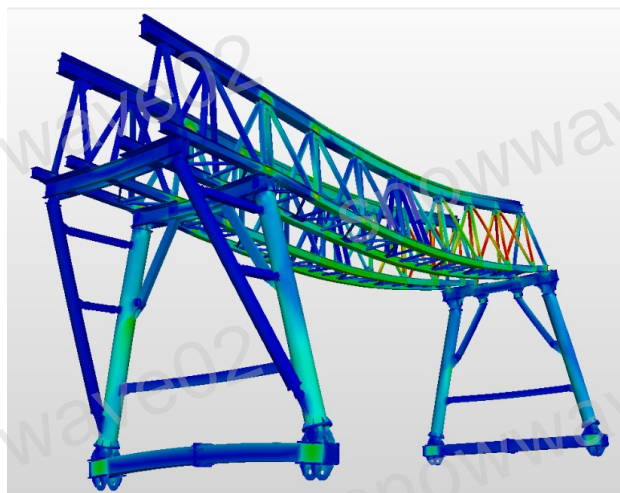
第五篇：单元正确性验证。介绍有限元单元正确性的验证方法，通过多个实例比较自研结构求解器程序 iSolver 与 Abaqus 的分析结果，从而说明整个正确性验证的过程和 iSolver 结果的正确性。

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

本文为第六篇：**General 梁单元的刚度矩阵**。介绍梁单元的基础理论和 Abaqus 中 General 梁单元的刚度矩阵的修正方式，采用这些修正方式可以得到和 Abaqus 完全一致的刚度矩阵。

## 2 第六篇：General 梁单元的刚度矩阵

杆梁在楼房、桥梁、船舶和其它很多结构中都是常见的结构类型，很多实际工程问题都是仅仅由杆梁组合构成的框架、板架等，而杆单元其实就是它的一个特殊情况，对实际工程问题来说，杆单元不能承受弯曲载荷，但对程序来说杆单元完全可以在梁单元基础上稍加改造，甚至可以直接用一个单元函数，通过类型来控制得到的是梁还是杆的刚度。



相对来说，在实际应用过程中，商业软件对梁的修正方式相对较少，如果自己编程序，采用这些修正方式可以得到和商业软件完全一致的刚度矩阵，如果刚度矩阵完全一致，那么对任何的算例都可以得到和商业软件完全一致的结果了。

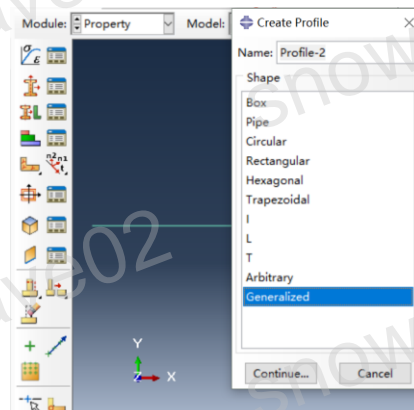
实际的梁都是有截面形状的，商用软件分析时都采用两步走的形式：

(1) 第一步：通过这些截面形状类型和参数得到构建梁单元所需的基本截面属性参数，譬如矩形面积=长\*宽等。

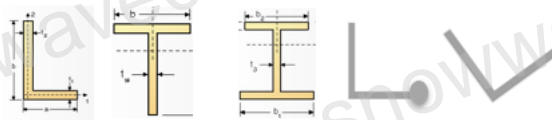
(2) 第二步：利用上面得到的截面属性参数组成梁单元的刚度矩阵。

相对应的，一般商用软件的梁都有两类：

(1) 一类是已知截面属性参数的梁，在 **Abaqus** 中创建梁时选择 **General** 就是 **General** 梁。



(2) 另一类是已知截面形状类型和几何尺寸的梁，在 **Abaqus** 创建梁截面时选择除 **General** 外的其它选项都是 **Geometry** 梁。



本篇先讨论 **General** 梁单元的刚度矩阵的基本理论和 **Abaqus** 的修正方式，**Geometry** 梁的计算方法只是比第一类梁多了一步怎么从截面几何参数得到截面属性参数，当然针对不同形状类型，**Abaqus** 也做了许多的修正，将放到下一篇中讨论。

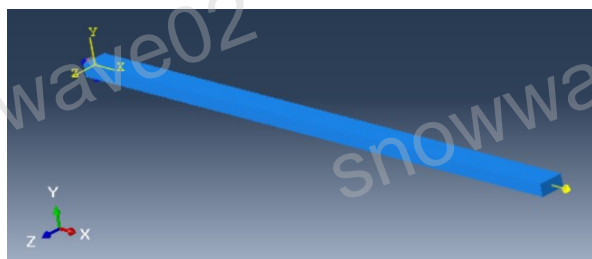
## 2.1 梁单元的基本理论

### 2.1.1 梁单元的受力方式分类

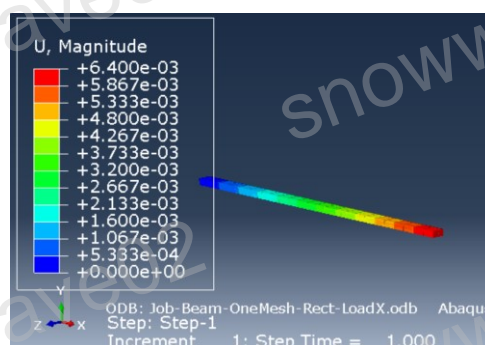
梁单元可能的受力方式有三种，譬如，对一个 **x** 方向的梁来说：

#### 2.1.1.1 轴向拉伸力

当力的方向沿 **x** 方向时。

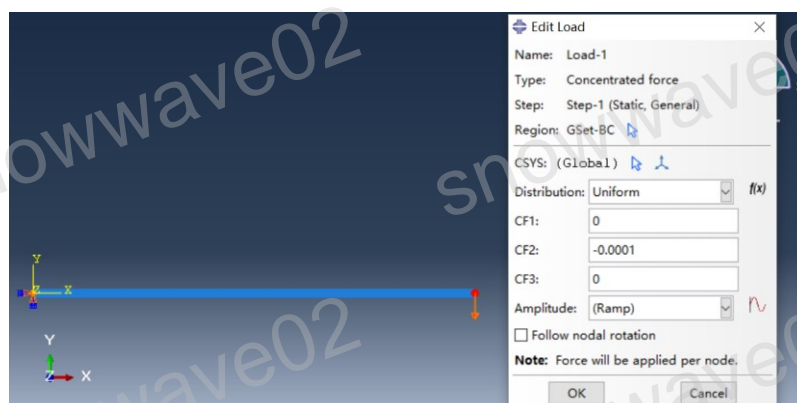


此时得到的位移也是只有轴向  $x$  方向的平动位移。

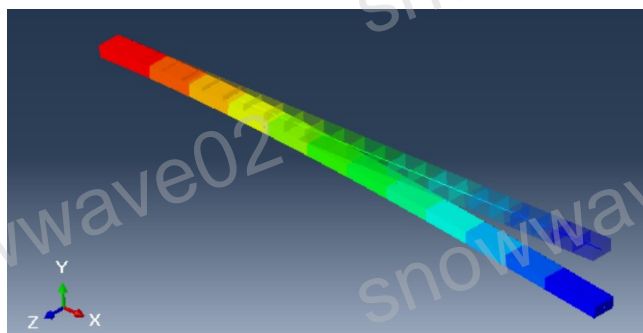


### 2.1.1.2 横向弯曲力

集中力的方向沿  $y$  或者  $z$  方向，或者采用以  $y$  或者  $z$  轴为转动轴的力矩加载时。

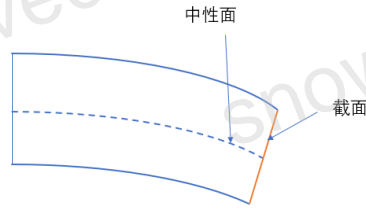


此时得到的位移沿受力方向弯曲。



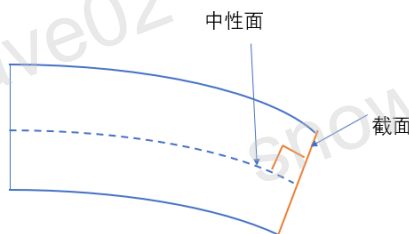
梁单元假设原来的截面受弯曲力后依然是平面。



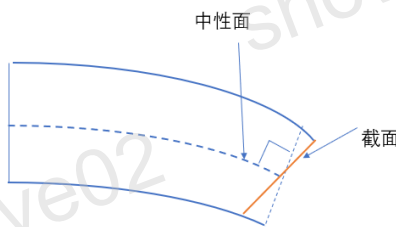


在弯曲时，梁的下端面变短被压缩了，上端面变长被拉长了，那么中间必然有个长度不变的平面，就称为中性面。受力前截面和中性面是垂直的，那么弯曲后的截面和中性面就有两种可能：

- (1) 如果是细长梁，那么可以认为依然垂直，此时就称为 **Euler 梁**。



- (2) 如果是厚梁，那么不再垂直，此时就称为 **Timoshenko 梁**。显然 Timoshenko 梁覆盖了 Euler 梁的功能，在细长梁的情况应该就和 Euler 梁完全一致。



对一个三维体是否能简化为梁主要看截面尺寸与长度的比，如果是线性材料，简单判据如下：

- (1) 截面尺寸/长度  $< 1/15$ ，认为是 **Euler 梁**；
- (2) 截面尺寸/长度  $< 1/8$ ，认为是 **Timoshenko 梁**；
- (3) 截面尺寸/长度  $> 1/8$ ，不可以近似为梁。

✧ 详见 Abaqus: Analysis User's Manual v6.12: 29.3.3 Choosing a beam

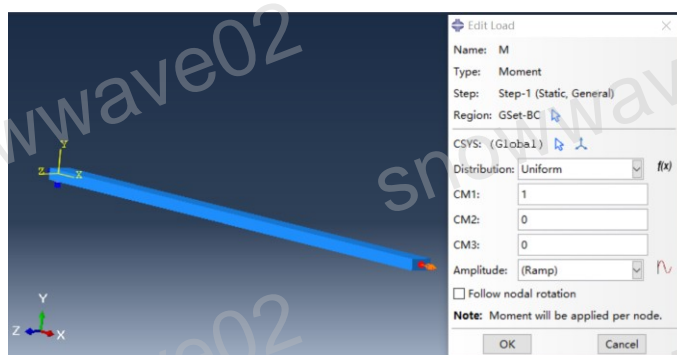
(1) For beams made of uniform material, typical dimensions in the cross-section should be less than about 1/15 of typical axial distances for transverse shear flexibility to be negligible.

(2) shear flexible beam theory can provide useful results for cross-sectional dimensions up to  $1/8$  of typical axial distances or the wavelength of the highest natural mode that contributes significantly to the response.

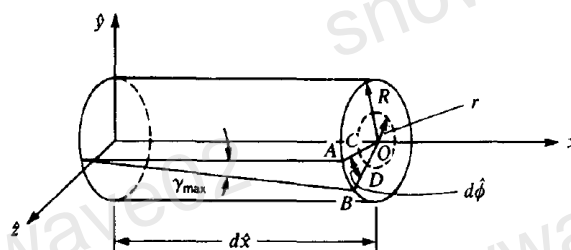
(3) Beyond this ratio the approximations that allow the member's behavior to be described solely as a function of axial position no longer provide adequate accuracy.

### 2.1.1.3 轴向扭转

当力是以  $x$  方向为转动轴的力矩或者在截面四周加力载荷时。



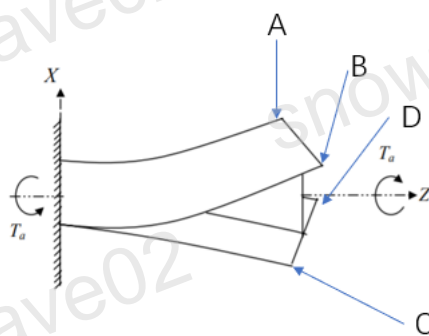
此时得到的绕  $x$  轴转动的扭转变形。



#### 2.1.1.3.1 翘曲

扭转对非圆形截面，那么截面可能会不再是一个平面，而边角会产生翘曲效应 (Warping)。类似下面图的 A 和 C 点明显在  $x$  方向收缩了，而 B 和 D 点在  $x$  方向延长了，导致 ABCD 不在一个平面了。





### 2.1.1.3.2 剪切中心

如果是一个集中力加在截面上，一般由于力矩作用都会导致截面扭转。譬如下面的槽型梁截面做的实验，力加在左边截面就逆时针扭转，加在右边，截面顺时间扭转。



那么必然中间有个从逆时针到顺时针的过渡点，这个点上加集中力将不发生扭转。

该点就称为剪切中心（**Shear Center**）。



注意：剪切中心和形心不一定重合。譬如上面的槽形截面，剪切中心在腹板的左边，而形心必然在腹板的右边。

## 2.1.2 梁单元的刚度矩阵基本理论

刚度的就是指材料或结构在受力时抵抗弹性变形的能力。梁的每种受力类型都会产生对应的刚度，也就是将产生梁的轴向拉伸刚度、弯曲刚度、轴向扭转刚度，除了这三个基本刚度外，还有它们之间的耦合刚度。对 B31 梁，为 2 节点一次梁，每个节点 6 个自由度，那么 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，以第一个节点自由度相关的 6X6 的小矩阵做研究对象。

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

K 的任意一项  $K_{ij}$  都是 i 和 j 两个自由度的乘积，其中对角元素为自身的自由度的乘积，譬如  $K_{11}$  就是 u 和 u 的乘积，而非对角元素都是耦合项， $K_{15}$  就是 u 和  $\theta_y$  的乘积。

上面就是一个 General 梁的刚度矩阵，可以发现很多耦合项都取了 0，这是因为梁在实际应用中进行了简化，具体就要涉及刚度矩阵每个元素怎么求的。

由附录[有限元的刚度矩阵基本理论](#)可知，K 的任意一项  $K_{ij}$  都是应变的表达式中 i 和 j 两个自由度的积分，如果想要分析梁单元的刚度矩阵每个元素的来源，那么只要分析梁的应变的表达式就行了。

梁单元的应变分为：

### 2.1.2.1 正应变

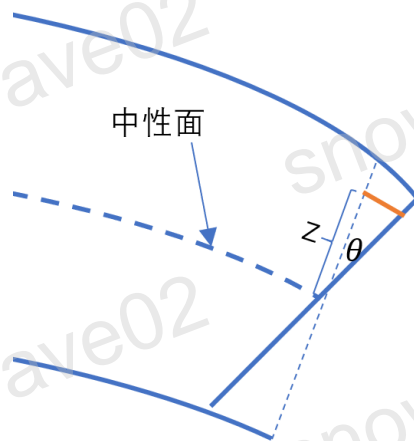
$$\varepsilon_x = \frac{\partial u}{\partial x} + z \frac{\partial \theta_y}{\partial x} - y \frac{\partial \theta_z}{\partial x}$$

分两部分组成：

第一部分  $\frac{\partial u}{\partial x}$  是 x 方向自由度本身的平移，由轴向拉伸力导致的。

第二部分的  $z \frac{\partial \theta_y}{\partial x}$  是弯曲时转动角度  $\theta_y$  造成的轴向平移  $Z * \theta_y$  产生。Z 就是相对中性面

的距离。  $-y \frac{\partial \theta_z}{\partial x}$  类似。



✧ 注意，这里由于没有考虑翘曲效应，所以没有翘曲项。

对应到刚度矩阵中，将有三部分刚度：

① 轴向拉伸刚度：  $\frac{\partial u}{\partial x}$  和  $\frac{\partial u}{\partial x}$  的乘积得到  $K_{11}$ ，也就是轴向拉伸刚度。此时积分因子为  $\int dA = A$ ，也就是拉伸刚度与  $A$  有关。下图红色部分。

② 横向弯曲刚度：  $z \frac{\partial \theta_y}{\partial x} - y \frac{\partial \theta_z}{\partial x}$  和  $z \frac{\partial \theta_y}{\partial x} - y \frac{\partial \theta_z}{\partial x}$  的乘积得到  $K$  的右下角  $2 \times 2$  的矩阵  $K_{55}, K_{56}, K_{65}, K_{66}$  横向弯曲刚度，此时积分因子为  $\int z^2 dA = I_{11}, \int y^2 dA = I_{22}, \int yz dA = I_{12}$ ，也就是常说的弯曲惯量。下图蓝色部分。

③ 轴向拉伸和横向弯曲耦合刚度：  $\frac{\partial u}{\partial x}$  和  $z \frac{\partial \theta_y}{\partial x} - y \frac{\partial \theta_z}{\partial x}$  的乘积得  $K$  的右上角  $K_{15}, K_{16}$  和左下角  $K_{51}, K_{61}$ 。此时积分因子为  $\int z dA = S_y$  和  $\int y dA = S_z$ ，也就是常说的静矩，该值和形心坐标相关。下图绿色部分。

	u	v	w	$\theta_x$	$\theta_y$	$\theta_z$
	1	2	3	4	5	6
u	625000	0	0	0	-625000	-125000
v	0	1.6578e+03	0	1.6578e+03	0	1.9894e+05
w	0	0	3.2929e+03	6.5859e+03	-3.9515e+05	0
$\theta_x$	0	1.6578e+03	6.5859e+03	4.0212e+06	-7.9030e+05	1.9894e+05
$\theta_y$	-625000	0	-3.9515e+05	-7.9030e+05	5.8460e+07	3.8542e+06
$\theta_z$	-125000	1.9894e+05	0	1.9894e+05	3.8542e+06	3.1581e+07

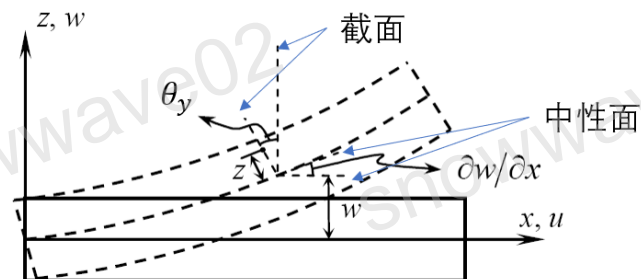
### 2.1.2.2 剪切应变

剪切应变有两个值  $\gamma_{xy}$  和  $\gamma_{xz}$ ，我们仅以  $\gamma_{xz}$  为例：

$$\gamma_{xz} = \left( \frac{\partial w}{\partial x} + \theta_y \right) + y \frac{\partial \theta_x}{\partial x}$$

分两部分组成：

第一部分 $\left(\frac{\partial w}{\partial x} + \theta_y\right)$ 是  $z$  方向弯曲导致的。受  $Z$  方向弯曲变形后，中性面的剪切角度为 $\frac{\partial w}{\partial x}$ ，截面的剪切角度为 $\theta_y$ ，当这两个值相等时，那么截面依然垂直中性面，也就是 Euler 梁，不相等时，存在一个剪切角度为 $\left(\frac{\partial w}{\partial x} + \theta_y\right)$ 。



第二部分的 $y \frac{\partial \theta_x}{\partial x}$ 是轴向扭转时转动角度 $\theta_x$ 造成的剪切角度在  $xz$  方向的分量，具体可看有限元理论书，其中  $y$  是相对剪切中心的值。

对应到刚度矩阵中，也将有三部分刚度：

④ 轴向扭转刚度：  $y \frac{\partial \theta_x}{\partial x}$ 和 $y \frac{\partial \theta_x}{\partial x}$ 的乘积得到  $K_{44}$ ，也就是轴向扭转刚度。对应下图红色部分。此时积分因子为 $\int y^2 dA$ ，如果加上  $z$  方向的， $\int y^2 + z^2 dA = \int r^2 dA$ ，也就是抗扭刚度系数  $J$ 。这个  $J$  与极惯性矩不是一回事，极惯性矩是以形心来计算  $r$  的，而这边  $J$  是以剪切中心来计算的，只不过对圆截面时，剪切中心和形心重合，这两个系数也恰好相同。

⑤ 横向剪切刚度：  $\frac{\partial w}{\partial x} + \theta_y$ 和 $\frac{\partial w}{\partial x} + \theta_y$ 的乘积得到  $K$  的矩阵  $K_{33}$ ,  $K_{35}$ ,  $K_{53}$ ,  $K_{55}$ ，也就是横向剪切刚度，下图蓝色部分。此时积分因子为 $\int 1 dA = A$ 。类似另一个应变 $\gamma_{xy}$ 也能得到另外四个刚度元素。

⑥ 轴向扭转和横向剪切耦合刚度：  $y \frac{\partial \theta_x}{\partial x}$ 和 $\frac{\partial w}{\partial x} + \theta_y$ 的乘积得  $K$  的  $K_{43}$ ,  $K_{34}$  和  $K_{45}$ ,  $K_{54}$ 。下图绿色部分。此时积分因子为 $\int y dA$ ，该值和剪切中心坐标相关。类似另一个应变 $\gamma_{xy}$ 也能得到另外四个刚度元素。

	u	v	w	$\theta_x$	$\theta_y$	$\theta_z$
	1	2	3	4	5	6
u	1	0	0	0	-625000	-1250000
v	2	1.6578e+03	0	1.6578e+03	0	1.9894e+05
w	3	0	3.292e+03	6.585e+03	-3.951e+05	0
$\theta_x$	4	1.6578e+03	6.5859e+03	4.0212e+04	7.9030e+06	1.9894e+05
$\theta_y$	5	-625000	0	7.903e+06	5.846e+07	3.8542e+06
$\theta_z$	6	-1250000	1.9894e+05	1.9894e+05	3.8542e+06	3.1581e+07

除了上述元素，刚度矩阵其它元素均为 0。

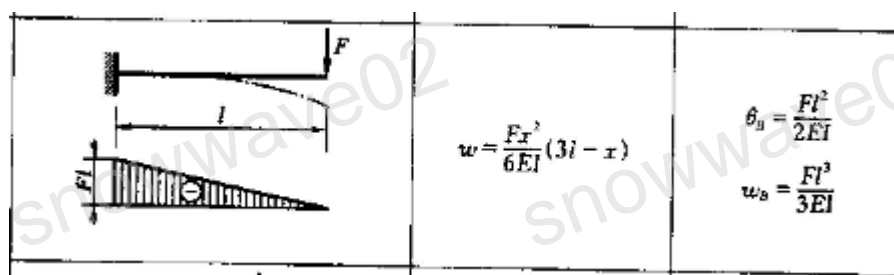
## 2.2 Abaqus 的实现方式

### 2.2.1 Abaqus 单元类型

#### 2.2.1.1 横向剪切效应

上面按弯曲时是否考虑剪切应变可把梁分成 Euler 和 Timoshenko 梁, 对应 Abaqus 梁单元类型分别为 B33/B23 和 B31/B21, 默认值是 Timoshenko 梁。

可以做一个悬臂梁模型证明 B33 和 B31 分别是 Euler 梁和 Timoshenko 的理论, 如下:



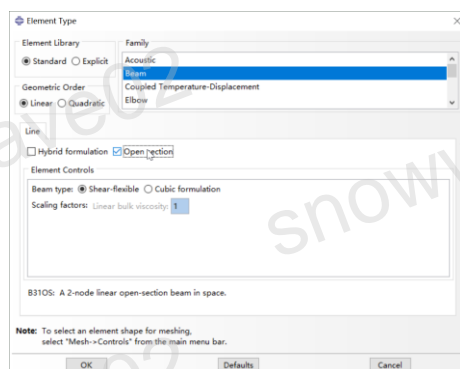
悬臂梁的理论计算公式最大的  $U = (F \cdot L^3) / (3 \cdot E \cdot I)$ 。

其中 E 指的是杨氏模量(弹性系数), L 指的是悬臂梁的长度,  $I = bh^3/12$  (b 为梁宽, h 为梁高), 显然这个值就是没有考虑剪切应变得到的。

在 Abaqus 中, 取一个单元, 如果用 B31, 可以发现结果和理论值差别很大, 但如果选 B33, 那么和理论值完全一致。

#### 2.2.1.2 翘曲效应

在 Abaqus 中, 只有开环 (Open Section) 截面譬如 I、T、L 等才会有可能考虑翘曲效应, 此时 Abaqus 中需要选择 B31OS 单元类型, 默认的 B31 没有考虑翘曲效应, 后面的讨论可以证明 General 梁的翘曲因子对 B31 单元没有作用。





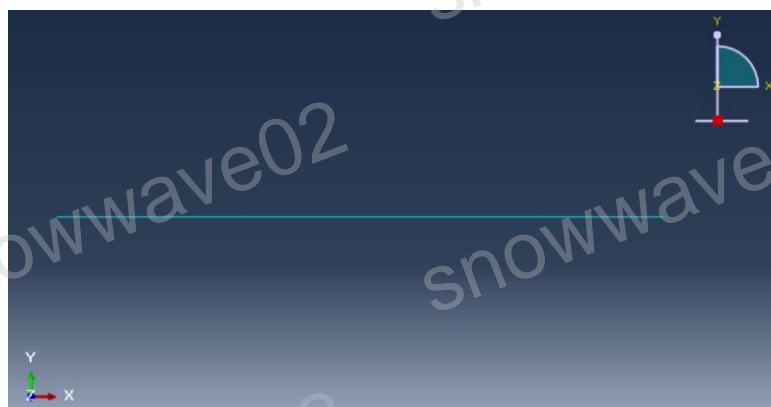
## 2.2.2 Abaqus 的 General 梁单元的刚度矩阵

我们本文只讨论 Abaqus 默认的梁单元类型 B31，不讨论 B33 或者 B31OS，下面将采用一个最简单的 General 梁的例子说明 Abaqus 中 B31 梁单元的刚度矩阵的实现方式和修正。

### 2.2.2.1 模型描述

模型文件：inp 文件为 Job-Beam-OneMesh-GeneralYBend.inp

模型参数如下



几何：长 240。

用 General 截面，截面参数修改。

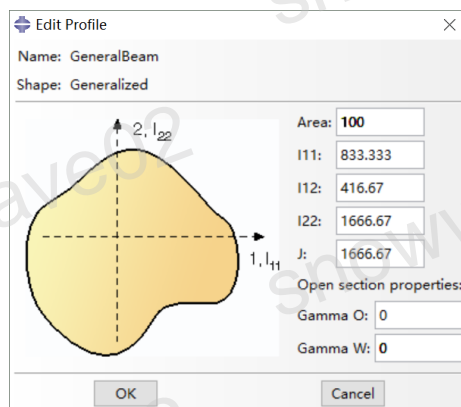
材料：杨氏模量 1500000，泊松比 0.3。

网格：取 B31 单元类型，只划分一个网格。

边界和载荷：任意，不影响刚度矩阵结果。

### 2.2.2.2 无偏置的刚度矩阵

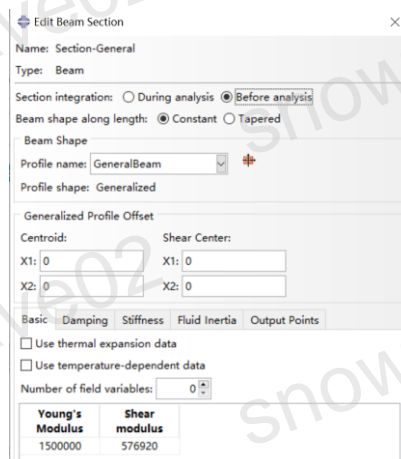
General 梁参数如下：



Area、I11、I12、I22、J 的说明可参考上一章[梁单元的刚度矩阵基本理论](#)。同时 Gamma O 和 Gamma W 为翘曲的两个参数。在这儿先设为 0。



截面属性设置如下，采用 **General** 梁单元，且偏置为 0：



**Abaqus** 输出的刚度矩阵结果：

Element1 Stiffness <12x12 double>												
	1	2	3	4	5	6	7	8	9	10	11	12
1	625000	0	0	0	0	0	-625000	0	0	0	0	0
2	0	1.6578e+03	0	0	0	1.9894e+05	0	-1.6578e+03	0	0	0	1.9894e+05
3	0	0	3.2929e+03	0	-3.9515e+05	0	0	-3.2929e+03	0	0	-3.9515e+05	0
4	0	0	0	4.0064e+06	0	0	0	0	0	-4.0064e+06	0	0
5	0	0	-3.9515e+05	0	5.7835e+07	2.6042e+06	0	0	3.9515e+05	0	3.7001e+07	-2.6042e+06
6	0	1.9894e+05	0	0	2.6042e+06	2.9081e+07	0	-1.9894e+05	0	0	-2.6042e+06	1.8664e+07
7	-625000	0	0	0	0	0	625000	0	0	0	0	0
8	0	-1.6578e+03	0	0	0	-1.9894e+05	0	1.6578e+03	0	0	0	-1.9894e+05
9	0	0	-3.2929e+03	0	3.9515e+05	0	0	3.2929e+03	0	0	3.9515e+05	0
10	0	0	0	-4.0064e+06	0	0	0	0	0	4.0064e+06	0	0
11	0	0	-3.9515e+05	0	3.7001e+07	-2.6042e+06	0	0	3.9515e+05	0	5.7835e+07	2.6042e+06
12	0	1.9894e+05	0	0	-2.6042e+06	1.8664e+07	0	-1.9894e+05	0	0	-2.6042e+06	2.9081e+07

不为 0 的刚度分为

① **轴向拉伸刚度**：K11 理论值= $E \cdot A / L = 1.5e6 \cdot 100 / 240 = 6.25e5$ ，Abaqus 和理论值完全一致。

② **横向弯曲刚度**：与 I11、I12、I22 有关，而且因为是积分，所以与具体的积分方式有关，在自编程序 iSolver 中编程实测如果用完全积分，那么和 Abaqus 差异较大，而采用单点减缩积分，将可以得到和 Abaqus 完全一致的结果，猜测 Abaqus 中采用了减缩积分。

④ **轴向扭转刚度**：K44 理论值= $G \cdot J / L = 576920 \cdot 1666.67 / 240 = 4.0064e+006$ ，Abaqus 和理论值完全一致。

⑤ **横向剪切刚度**：Kts 与面积 A 有关，Abaqus 除了采用减缩积分外，对这项还做了修正，增加了一个几何因子，使得当厚梁时，Kts 接近 Timoshenko 梁结果，当细长梁时，Kts 接近 0。

✧ 详见 Abaqus: Analysis User's Manual v6.12: 29.3.3 Choosing a beam

*The dimensionless factors  $f_p$  are always included in the calculation of transverse shear stiffness, is a dimensionless factor used to prevent the shear stiffness from becoming too large in slender beam elements;*

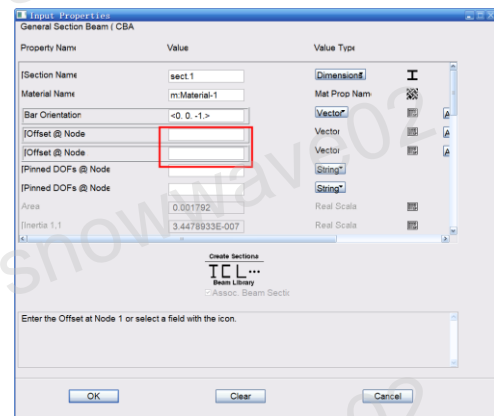
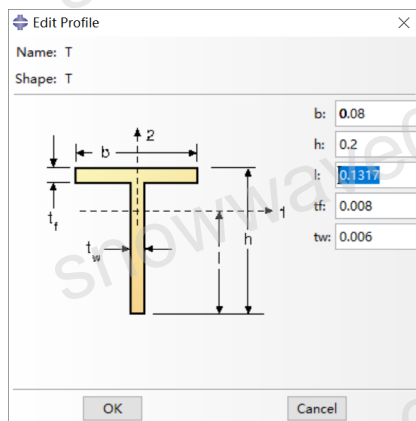
该因子如下：

$$f_p = \left(1 + 0.25 \frac{e^2 A}{12I}\right)^{-1} \quad \text{for first-order Timoshenko beam elements and}$$

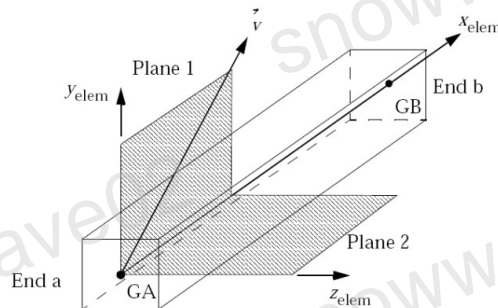
$$f_p = \left(1 + 0.25 \times 10^{-4} \frac{e^2 A}{12I}\right)^{-1} \quad \text{for all other beam elements.}$$

这里，一次梁采用的因子是 0.25，而不是 0.25e-4，不清楚为什么要这样，因为 Abaqus 对一次板单元采用的 fp 修正因子是 0.25e-4，那么对一次梁也应该是 0.25e-4 才正常。如果谁能知道为什么，欢迎交流。

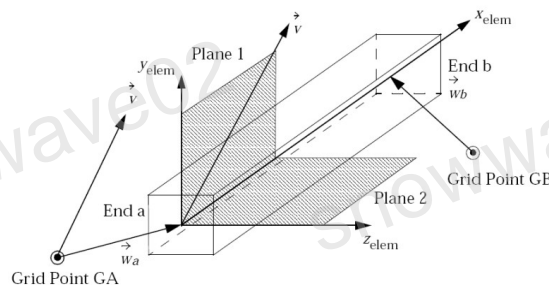
在 Abaqus 的 I、T、Trapezoid 等梁中，我们都可以设置形心坐标，Nastran 中也可以设置每个节点的 Offset，偏置是一种常见的梁的安装方式。



如果没有偏置，那么形心或者剪切中心就在原点，如下图：



偏置时表示形心或者剪切中心不再是原点了，此时会对刚度矩阵产生影响。



以上刚度矩阵缺少了③轴向拉伸和横向弯曲耦合刚度和⑥轴向扭转和横向剪切耦合刚度，这两项就是由形心偏置和剪切中心偏置才能产生。

对③轴向拉伸和横向弯曲耦合刚度，属性系数为 $S_y = \int z dA = z_c * A$ 和 $S_z = \int y dA = y_c * A$ ，如果形心在原点，那么 $z_c = y_c = 0$ ，属性系数=0，导致耦合刚度=0。

对⑥轴向扭转和横向剪切耦合刚度，类似，只不过是剪切中心没有偏置使得刚度=0。

### 2.2.2.3 增加形心偏置的刚度矩阵

截面属性修改如下，采用 Genearl 梁单元，增加形心偏置为(1,2)。

Abaqus 输出的刚度矩阵结果：

Element1Stiffness <12x12 double>						
	1	2	3	4	5	6
1	625000	0	0	0	-625000	-1250000
2	0	1.6578e+03	0	0	0	1.9894e+05
3	0	0	3.2929e+03	0	-3.9515e+05	0
4	0	0	0	4.0064e+06	0	0
5	-625000	0	-3.9515e+05	0	5.8460e+07	3.8542e+06
6	-1250000	1.9894e+05	0	0	3.8542e+06	3.1581e+07

相对没有偏置的矩阵，③轴向拉伸和横向弯曲耦合刚度结果不再为 0。只要将属性系数为 $S_y = \int z dA = z_c * A$ 和 $S_z = \int y dA = y_c * A$ ，中的  $y_c=1$ ， $z_c=2$  就可以得出 Abaqus 的  $K_{15}=-625000$ ， $K_{16}=-1250000$ 。在这里，我们特意将  $y_c$  设为 1，使得 $S_y$ 就是面积，这样  $K_{15}$  它的值就和①轴向拉伸刚度的  $K_{11}$  是一样的了。

### 2.2.2.4 增加剪切中心偏置的刚度矩阵

截面属性修改如下，采用 Genearl 梁单元，增加剪切中心 Shear Center 偏置为(1,2)。

✚ Edit Beam Section ✕

Name: Section-General

Type: Beam

Section integration: ☐ During analysis ☒ Before analysis

Beam shape along length: ☒ Constant ☐ Tapered

Beam Shape

Profile name: GeneralBeam

Profile shape: Generalized

Generalized Profile Offset

Centroid: X1: 1 X2: 2

Shear Center: X1: 1 X2: 2

Basic Damping Stiffness Fluid Inertia Output Points

☐ Use thermal expansion data

☐ Use temperature-dependent data

Number of field variables: 0

Young's Modulus	Shear modulus
1500000	576920

**Abaqus** 输出的刚度矩阵结果:

	1	2	3	4	5	6
1	625000	0	0	0	-625000	-1250000
2	0	1.6578e+03	0	1.6578e+03	0	1.9894e+05
3	0	0	3.2929e+03	6.5859e+03	-3.9515e+05	0
4	0	1.6578e+03	6.5859e+03	4.0212e+06	-7.9030e+05	1.9894e+05
5	-625000	0	-3.9515e+05	-7.9030e+05	5.8460e+07	3.8542e+06
6	-1250000	1.9894e+05	0	1.9894e+05	3.8542e+06	3.1581e+07

相对没有剪切中心偏置的矩阵，⑥轴向扭转和横向剪切耦合刚度结果不再为 0。

同时④轴向扭转刚度的值发生了变换，因为 J 的值是相对剪切中心的，Abaqus 的修正方式和形心偏置类似。

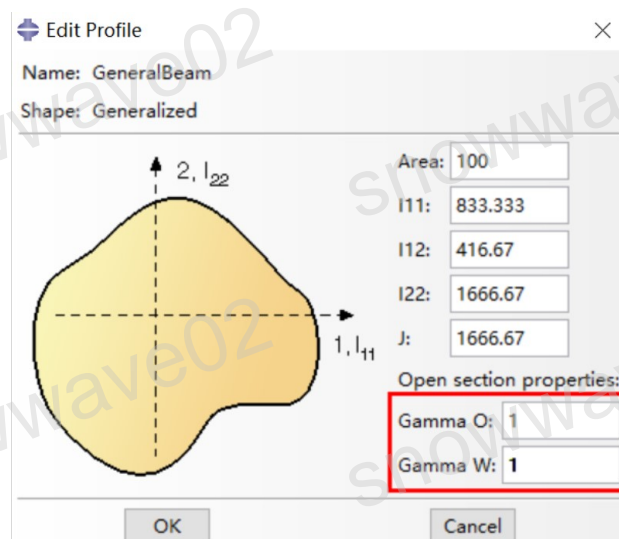
## 2.2.2.5 增加翘曲参数的刚度矩阵

这里再顺便验证一下翘曲参数的意义，前面说了翘曲参数只对 B31OS 有作用。在这两个参数为 0 时，Abaqus 的刚度矩阵如下：

Element1 Stiffness <14x14 double>						
	1	2	3	4	5	6
1	625000	0	0	0	-625000	-1250000
2	0	1.6578e+03	0	1.6578e+03	0	1.9894e+05
3	0	0	3.2929e+03	6.5859e+03	-3.9515e+...	0
4	0	1.6578e+03	6.5859e+03	4.0212e+06	-7.9030e+...	1.9894e+05
5	-625000	0	-3.9515e+...	-7.9030e+...	5.8460e+07	3.8542e+06
6	-1250000	1.9894e+05	0	1.9894e+05	3.8542e+06	3.1581e+07
7	0	0	0	0	0	0

刚度为 14X14 的矩阵，也就是每个节点增加了一个额外的翘曲自由度，该自由度在两个翘曲参数为 0 时，刚度对应为 0。

采用 Genearl 梁单元将两个翘曲参数设为 0：



**Abaqus** 输出的刚度矩阵结果:

Element1 Stiffness <14x14 double>							
	1	2	3	4	5	6	7
1	625000	0	0	0	0	0	12500
2	0	1.6578e+03	0	0	0	1.9894e+05	0
3	0	0	3.2929e+03	0	-3.9515e+05	0	0
4	0	0	0	4.8077e+06	0	0	9.6153e+07
5	0	0	-3.9515e+05	0	5.7835e+07	2.6042e+06	0
6	0	1.9894e+05	0	0	2.6042e+06	2.9081e+07	0
7	12500	0	0	9.6153e+07	0	0	1.1538e+10

可见第 7 自由度的刚度值被更改了。

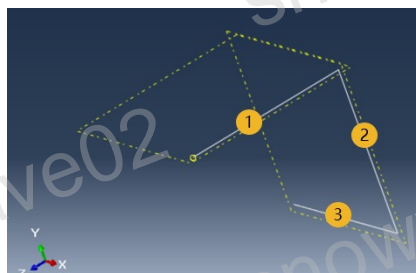
## 2.3 General 梁单元刚度矩阵的算例考核

由上面讨论可知, 对 **General** 梁, 如果采用 **Timoshenko** 梁理论+**Abaqus** 的修正, 就能得和 **Abaqus** 的 **B31** 单元类型完全一致的刚度矩阵。自研软件的正确性证明是非常困难的, 无论你测试多少标准算例或者和试验结果对比, 都很难让用户踏实的相信你的结果和商业软件是一致的, 但如果一个模型仅有 **General** 梁组成, 那么无论这个模型多么的复杂, 我们一般都可以得到和 **Abaqus** 完全一致的分析结果。下面将使用一个简单的算例综合验证, 针对该算例我们对比采用了同样修正的自编程序 **iSolver** 和 **Abaqus** 的结果, 可以发现结果如预期的一样, 没有任何误差。

### 2.3.1 模型描述

模型文件: inp 文件为 Job-Beam-3Beamin3D.inp

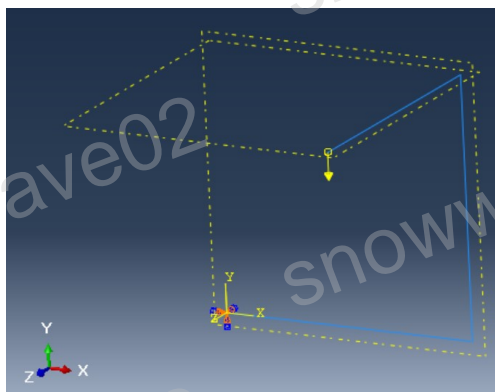
模型参数如下





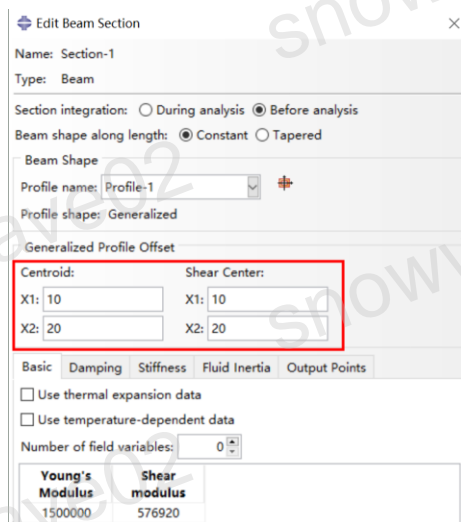
模型由三根梁组成。每根梁长 240。2 号梁垂直于 1 号梁，3 号梁垂直于 1、2 梁所在平面。

**边界和载荷：**在 3 号梁根部固支。一号梁自由端加-Y 方向集中力。

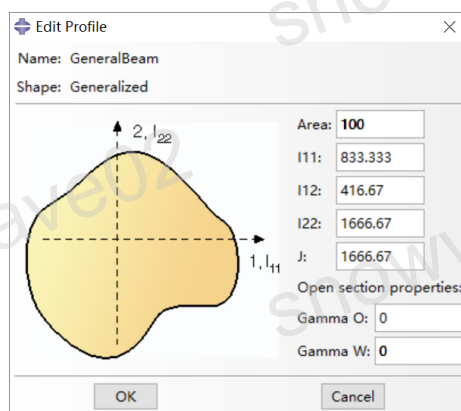


此时可知，梁将同时受到轴向拉伸、弯曲力、轴向扭转三个力的共同作用。

**梁参数：**每根梁取形心和剪切中心的偏置都为(10,20)。

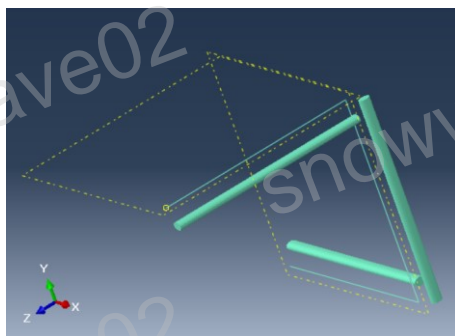


General 梁参数如下：



在 Abaqus 三维梁显示中可以看到这种偏置如下，为了和梁长 240 比拟，我们取的偏置也比较大。



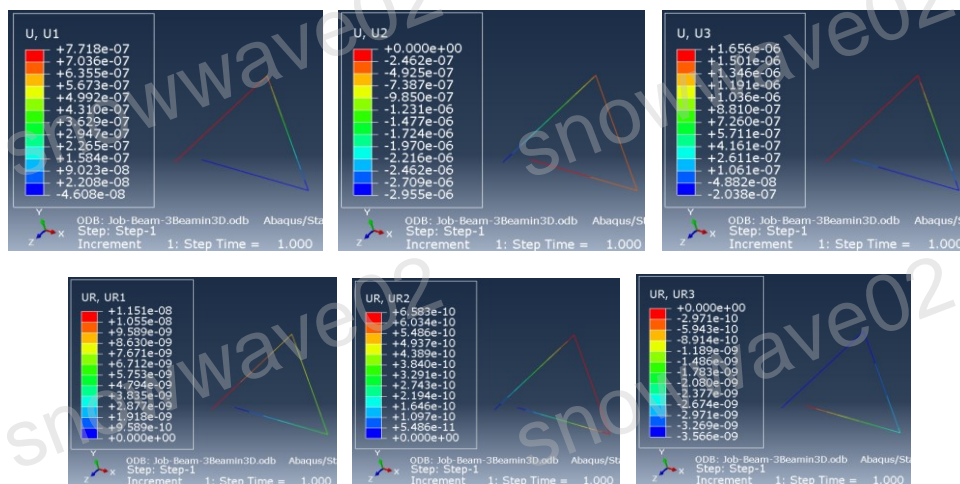


材料：杨氏模量 1500000，泊松比 0.3。

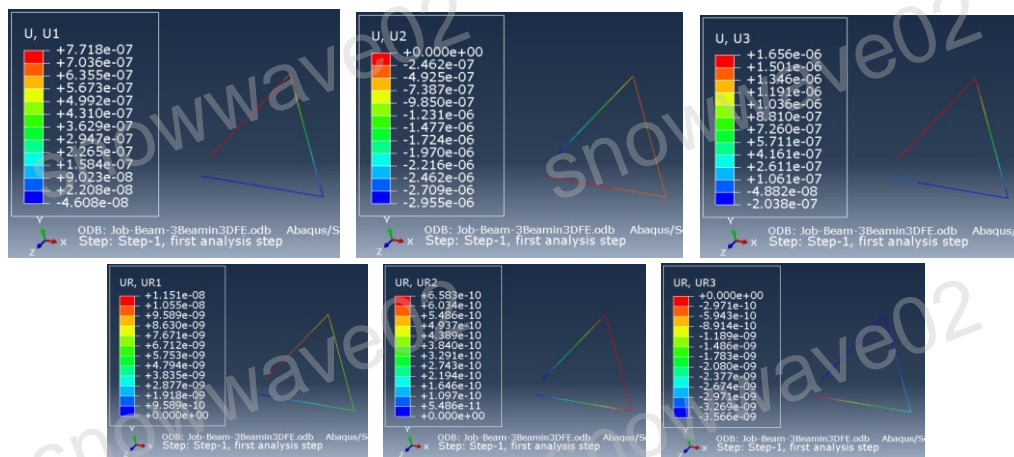
网格：取 B31 单元类型，每根梁只划分一个网格。

## 2.3.2 分析结果

采用 Abaqus 求解器，得到的 U 和 UR 分别如下



采用 iSolver 求解器，得到的 U 和 UR 分别如下：



可以看到 U 和 UR 的最大值最小值都是和 Abaqus 完全一致的，没有任何的误差。

但如果算法稍微有些差异，就会得到不一样的结果，譬如同样的模型 Nastran 的 U1 结果如下：



✧ Nastran 和 Abaqus 的结果都是正确性，在这里差异过大只不过是网格都是画了一个。

## 2.4 总结

本文首先简单介绍了一下梁单元的基本理论，分析了每一部分刚度的来源，并研究了 Abaqus 中 General 梁的 B31 单元的刚度矩阵的修正方式，采用这些修正方式可以得到和 Abaqus 完全一致的刚度矩阵，从而对只有 General 梁组成的任意模型一般都能得到 Abaqus 完全一致的分析结果，并用一个简单的算例验证了该想法。

General 梁的 B31 单元的刚度矩阵在 Timoshenko 梁理论基础上的修正如下表：

项次	刚度	修正情况		说明
		修正	不修正	
1	轴向拉伸刚度		√	
2	横向弯曲刚度	√		采用减缩积分
3	轴向拉伸和横向弯曲耦合刚度	√		根据形心偏置进行修正
4	轴向扭转刚度	√		剪切中心的偏置会影响抗扭刚度系数 J
5	横向剪切刚度	√		增加了一个几何因子，使得细长梁的时候该项趋于 0。
6	轴向扭转和横向剪切耦合刚度	√		根据剪切中心的偏置进行修正。
7	其它元素		√	都为 0

有兴趣的可以自行下载 iSolver 进行验证，因为看不到 Abaqus 的源代码，上述 B31 的修正方式也仅是猜测，如果你在使用 iSolver 测试其它的由 General 梁组成的模型结果时发现和 Abaqus 结果不一致，欢迎联系我们。

如果有任何其它疑问或者项目合作意向，也欢迎联系我们：

snowwave02 From [www.jishulink.com](http://www.jishulink.com)

email: [snowwave02@qq.com](mailto:snowwave02@qq.com)

## 2.5 附录

### 2.5.1 有限元的刚度矩阵基本理论

在此不讨论严格的数学表达，只是简单说明一下有限元单元的刚度矩阵每一元素的求解过程，后面将用来推导梁的刚度矩阵的各个元素。

#### (1) 位移

某个自由度的平动位移可写为：

$$u_p = u_t + fr * u_r$$

分两部分组成，第一部分是该自由度本身的平移，第二部分是其它自由度转动角度造成的平移。其中  $fr$  是转动半径在该自由度方向的分量，与坐标相关的值。

#### (2) 应变

按照应变的定义，应变就是平动位移的导数与坐标的导数的比。那么无论是正应变还是剪切应变，都可以写成类似下方的形式：

$$\varepsilon = \frac{\partial u_p}{\partial x} = \frac{\partial u_t}{\partial x} + A_r * u_r$$

$A_r$  与坐标  $(x, y, z)$  相关。

等参形式下，任意点的位移  $u_t$ ,  $u_r$  可以表示节点位移  $u_{ti}$ ,  $u_{ri}$  的线性组合。

$$u_t = \sum_i N_i u_{ti}$$
$$u_r = \sum_i N_i u_{ri}$$

因为  $u_{ti}$ ,  $u_{ri}$  与坐标无关，那么带入应变表达式后，就得到文章的应变表达式

$$\varepsilon = \sum_i B_i * u_i$$

其中： $u_i$  表示平移或者转动自由度的位移。

#### (3) 应变

应变可以表示成节点位移的线性组合（详见附录推导）。

$$\varepsilon = \sum_i B_i * u_i$$

其中： $u_i$  表示平移或者转动自由度的位移。

前面的系数组成在有限元中就称为 **B** 矩阵，它由  $N_i$  或者  $\partial N_i / \partial x$  和  $A_r$  组成。

为简单起见，假定只有两项，写成矩阵形式就是

$$\varepsilon = B1 * u1 + B2 * u2$$

#### (4) 本构关系

加入本构关系，可得到应力为：

$$\sigma = D * \varepsilon = D * (B1 * u1 + B2 * u2)$$

如果是线性材料，那么对正应力，**D** 就是杨氏模量 **E**，对剪切应力就是剪切模量 **G**。

#### (5) 刚度矩阵

应变能为

$$U = \frac{1}{2} \int \sigma^T * \varepsilon = \frac{1}{2} * (u1 \quad u2) \int B^T * D * B * \begin{pmatrix} u1 \\ u2 \end{pmatrix}$$

则刚度矩阵为（由 **U** 怎么得到 **K** 可以类比弹簧的  $U = \frac{1}{2} K x^2$ ）：

$$K = \int B^T * D * B = \begin{pmatrix} \int B_1 * D * B_1 & \int B_1 * D * B_2 \\ \int B_2 * D * B_1 & \int B_2 * D * B_2 \end{pmatrix}$$

可以知道：

**K** 的任意一项  $K_{ij}$  都是应变的表达式中 **i** 和 **j** 两个自由度的积分，其中对角元素为自身的自由度的积分，譬如  $K_{11}$  就是  $u1$  和  $u1$  的积分，而非对角元素都是耦合项， $K_{12}$  就是  $u1$  和  $u2$  的积分。

### 2.5.2 自研有限元程序开发框架 iSolver 介绍

**iSolver** 为一个完全自主的通用有限元程序开发框架，可以快速集成客户的自研有限元算法和分析流程，开发解决特定问题的、与商业软件系统建立接口的、具备良好互操作能力的专业仿真软件，帮助客户实现自研程序的商业化包装和推广。

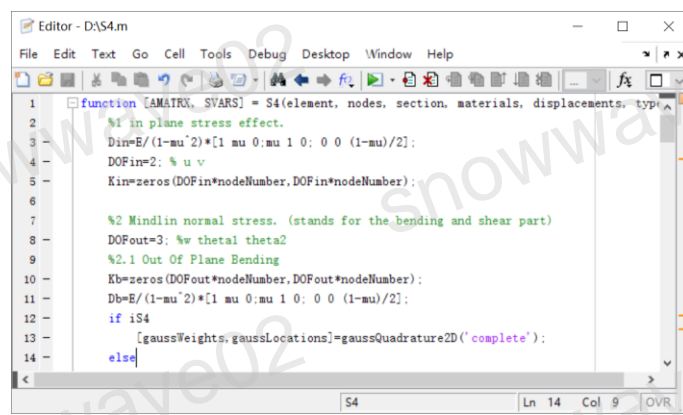


**应用场景：**在工业 4.0 时代，开发具有自主知识产权的工业 App 已是大势所趋，然而，我国工业软件中极其重要的一环仿真 CAE 软件市场一直被国外商业软件所垄断，

国内科研单位在专业领域虽然积累了不少计算方法和专业流程，在解决实际问题时，要么基于商软进行二次开发，只能做些小插件，完全依赖国外软件，要么嵌入网上的开源软件，开发难度大，且没有知识产权。iSolver 作为自主 CAE 软件研发的基础平台，可以帮助客户充分利用内部提供的一组基础 CAE 软件功能模块，避免从头开始做低水平地重复开发，高起点、高标准地开展工作。定制有专业针对性的用户界面，固化专家知识，规范操作流程，快速的形成自主品牌的 CAE 软件。

iSolver 有如下特点：

(1) 软件整体框架灵活，可快速集成各种自研有限元算法，支持用户利用 Matlab 编写自定义单元和材料等，便于扩展。



```
1 function [K, S, VARS] = S4(element, nodes, section, materials, displacements, type)
2 %1 in plane stress effect.
3 Dir=E/(1-mu^2)*[1 mu 0; mu 1 0; 0 0 (1-mu)/2];
4 DOFin=2; % u v
5 Kin=zeros(DOFin*nodeNumber, DOFin*nodeNumber);
6
7 %2 Mindlin normal stress. (stands for the bending and shear part)
8 DOFout=3; %w thetal theta2
9 %2.1 Out Of Plane Bending
10 Kb=zeros(DOFout*nodeNumber, DOFout*nodeNumber);
11 Db=E/(1-mu^2)*[1 mu 0; mu 1 0; 0 0 (1-mu)/2];
12 if iS4
13     [gaussWeights, gaussLocations]=gaussQuadrature2D('complete');
14 else
```

(2) 自带三维可视化前后处理，也可直接使用 Abaqus/CAE 做前后处理，上手容易。



(3) 自带结构有限元基本的单元类型和分析方法，内部算法在工程应用上的修正方法和 Abaqus 一致，结果和 Abaqus 接近。



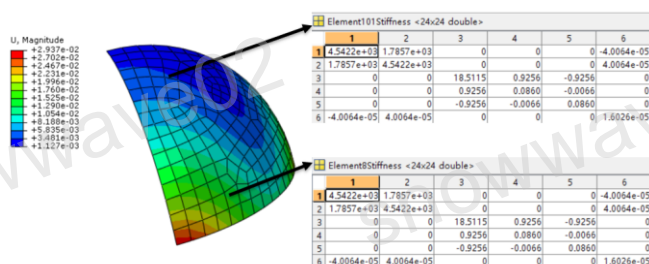


$$\ddot{\mathbf{u}} = \sum_b N_b(\mathbf{x}) \ddot{\mathbf{u}}_b(t) = \mathbf{N}(\mathbf{x}) \ddot{\mathbf{u}}(t)$$

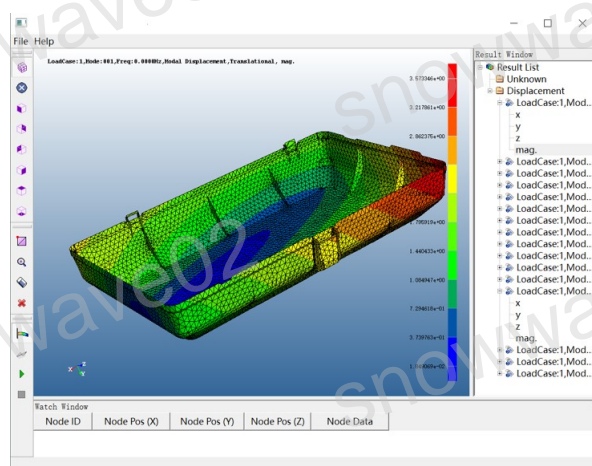
$$\mathbf{B}_b^T = \begin{bmatrix} N_{b,x_1} & 0 & 0 & N_{b,x_2} & 0 & N_{b,x_3} \\ 0 & N_{b,x_2} & 0 & N_{b,x_1} & N_{b,x_3} & 0 \\ 0 & 0 & N_{b,x_3} & 0 & N_{b,x_1} & N_{b,x_2} \end{bmatrix}$$

$$\Pi(\mathbf{U}) = \int_{\Omega} W(\mathbf{C}_{IJ}) d\Omega - \Pi_{\text{ext}}$$

(4) 支持和 Abaqus 结果一键式比较，且深入到刚度矩阵、质量矩阵各个元素的快速考核。



(5) 提供定制化开发服务，可快速集成各种自研有限元算法，并实现前后处理可视化界面，最终形成商用的客户拥有自主产权的商用化有限元软件。



具体介绍和下载请到下面网页：

<http://www.jishulink.com/content/post/337351>