# 有限元理论基础及Abaqus内部实现方式研究系列23： 编写简单面内拉伸问题UEL Step By

## Theoretical Foundation of Finite Element Method and Research on Internal Implementation of Abaqus Series 23: Writing a Simple In-Plane Tension Problem UEL Step By

SnowWave02       2020年6月9日 13:42 June 9, 2020 13:42    浏览：3553 Views: 3553    评论：15 Comments: 15    收藏：5 Favorites: 5

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

**有限元理论基础及Abaqus内部实现方式研究系列23： 编写简单面内拉伸问题UEL Step By的图1**
**有限元理论基础及Abaqus内部实现方式研究系列23： 编写简单面内拉伸问题UEL Step By的图2**
**==概述==  ==Overview==**

有限元理论基础及Abaqus内部实现方式研究系列23： 编写简单面内拉伸问题UEL Step By的图3

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 Operation

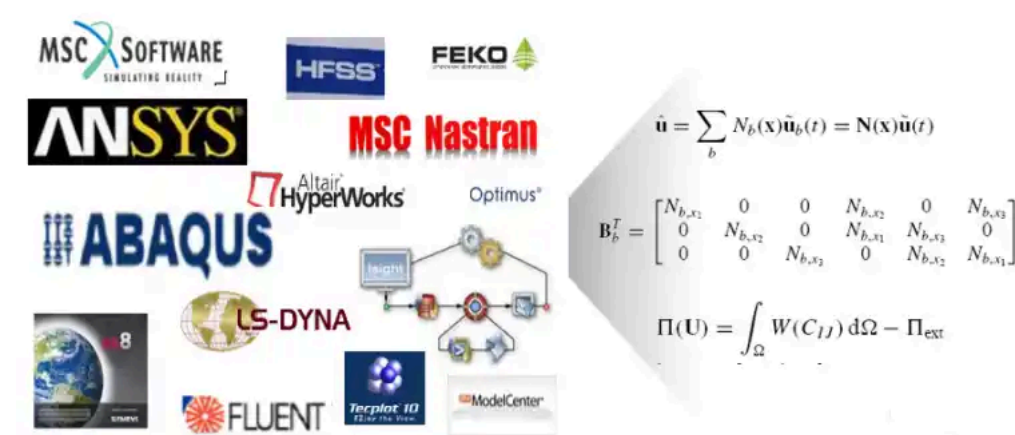(3) 自编程序  (3) Self-written program

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.



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

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

$$\Pi(\mathbf{U}) = \int_\Omega W(C_{IJ})\,d\Omega - \Pi_{ext}$$

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求解器，通过自研求解器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。

On one hand, we consult the theoretical manuals of various mainstream commercial software and guess the internal correction methods through extensive literature review. On the other hand, we program our own structural finite element solver and verify our guesses by comparing the results with those of commercial software. We study the correction methods like a glimpse through a tube, thus guessing the internal calculation methods of commercial finite element software. Since we focus on structural finite elements in CAE, we mainly choose Abaqus, which has relatively complete documentation among commercial structural finite element software, to study the internal implementation methods, and we will also involve other commercial software such as Nastran/Ansys for some issues. Many problems are not mathematically rigorous for the sake of understanding convenience, and due to our limited level, there may be many theoretical errors. We welcome discussions and look forward to more cooperation opportunities.

自主结构有限元求解器iSolver介绍视频： Introduction Video of Autonomous Structural Finite Element Solver iSolver
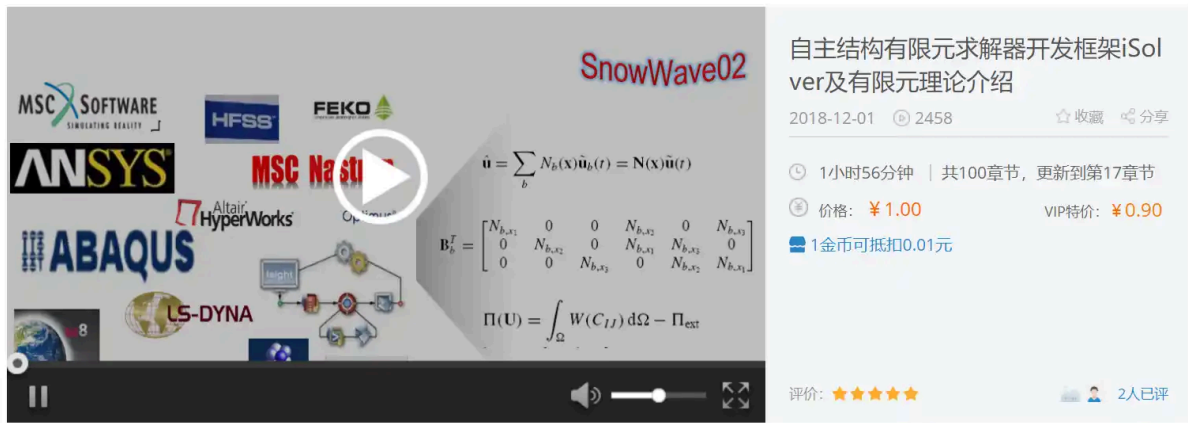
**==第23篇：编写简单面内拉伸问题UEL Step By Step==**

==23rd Article: Writing a Simple In-Plane Tension Problem UEL Step By Step==

面内拉伸问题是弹塑性力学的经典基础问题之一，即仅考虑平面里受力拉伸，而忽略力垂直于该面方向的情况。Abaqus在二维情况下采用平面应力单元，三维情况下采用壳单元。当然壳单元考虑的不仅仅是面内拉伸问题，还包括弯曲、剪切等其它问题，具体内容可以参照我们之前的文章：《有限元理论基础及Abaqus内部实现方式研究系列1：S4壳单元刚度矩阵研究》。本次我们仅考虑面内拉伸问题，并以UEL的方式实现。

The in-plane tension problem is one of the classic fundamental problems in elastoplastic mechanics, which considers tension in the plane only and ignores the case where the force is perpendicular to the plane. Abaqus uses plane stress elements in two dimensions and shell elements in three dimensions. Of course, shell elements consider not only in-plane tension problems but also other issues such as bending and shearing. For specific content, please refer to our previous article: "Series on the Theory of Finite Element Method and Internal Implementation of Abaqus 1: Research on the Stiffness Matrix of S4 Shell Element." In this article, we only consider the in-plane tension problem and implement it using the UEL method.

## 有限元理论基础及Abaqus内部实现方式研究系列23： 编写简单面内拉伸问题UEL Step By的图7

### 1.1 模型来源  1.1 Model Source

本文中所使用的模型文件是从一个简单壳单元的面内拉伸问题算例修改而来，即将单元定义和材料属性部分改为自定义单元的属性，具体修改方法，可以参照我们之前的文章：《有限元理论基础及Abaqus内部实现方式研究系列20：UEL用户子程序开发步骤》。也可直接从技术邻网页下下载，包含两个模型文件，一个是原始S4R模型的inp文件，一个是修改后使用UEL的inp文件：

The model file used in this article is modified from a simple shell element in-plane tension problem example, where the element definition and material property parts are changed to the properties of the custom element. For specific modification methods, please refer to our previous article: "Series on the Theory of Finite Element Method and Internal Implementation of Abaqus 20: Steps for UEL User Subroutine Development." You can also directly download it from the Technical Neighbor website, which includes two model files: one is the original S4R model inp file, and the other is the modified inp file using UEL:

有限元理论基础及Abaqus内部实现方式研究系列23： 编写简单面内拉伸问题UEL Step By的图8

## 1.2 算法步骤  1.2 Algorithm Steps

有限元方法有其基本步骤：   The finite element method has its basic steps:

1、       离散化和选择单元类型；  1.          Discretization and selection of element types;

2、       选择位移函数；  2.          Selection of displacement functions;

3、       定义应变/位移和应力/应变关系；

3. Define the relationship between strain/displacement and stress/strain;

4、       推导单元刚度矩阵和方程。

4. Derive the element stiffness matrix and equations.

5、       组装单元方程得出总体方程并引进边界条件

5. Assemble element equations to obtain the global equations and introduce boundary conditions.

6、       解未知自由度（或广义位移）

6. Solve the unknown degrees of freedom (or generalized displacements).

7、       求解单元应变和应力  7. Solution of element strain and stress

8、       解释结果  8. Explanation of the results

在Abaqus的建模过程中我们已经完成了步骤1，在求解过程中Abaqus会自动完成步骤5、6、7、8，本文的算法步骤则侧重于计算单元刚度矩阵和方程，即步骤2、3、4，也就是UEL的编写。

In the Abaqus modeling process, we have completed step 1. During the solution process, Abaqus will automatically complete steps 5, 6, 7, and 8. The algorithm steps in this article focus on calculating the element stiffness matrix and equations, i.e., steps 2, 3, and 4, which is the writing of the UEL.

结合UEL接口，针对面内拉伸问题的一般算法步骤：

Combined with the UEL interface, the general algorithm steps for the in-plane tensile problem:

1、计算形函数  1. Calculate the shape functions

2、计算B矩阵，即应变/位移关系矩阵  2. Calculate the B matrix, i.e., the strain/displacement relationship matrix

3、计算D矩阵，即应力/应变关系矩阵  3. Calculate the D matrix, i.e., the stress/strain relationship matrix

4、计算总体单元刚度矩阵  4. Calculate the overall element stiffness matrix

5、计算不平衡力  5. Calculate the unbalanced force

注：计算不平衡力用于Abaqus静力分析时进行收敛判断。

Note: The calculation of unbalanced force is used for convergence judgment in Abaqus static analysis.

**有限元理论基础及Abaqus内部实现方式研究系列23： 编写简单面内拉伸问题UEL Step By的图9**

## 1.3 UEL编写与运行  1.3 Writing and Running UEL

**有限元理论基础及Abaqus内部实现方式研究系列23： 编写简单面内拉伸问题UEL Step By的图10**

### 1.3.1 UEL编写  1.3.1 Writing UEL

UEL接口和For文件创建和使用方法可参照文章： 《有限元理论基础及Abaqus内部实现方式研究系列20： UEL用户子程序开发步骤》，接下来就按照算法步骤编写UEL文件。

The creation and usage methods of UEL interface and For files can be referred to in the article: "Series 20: Development Steps of UEL User Subroutines in the Theory of Finite Element Method and Internal Implementation of Abaqus", and then proceed to write the UEL file according to the algorithm steps.

1、计算形函数  1. Calculate the shape functions
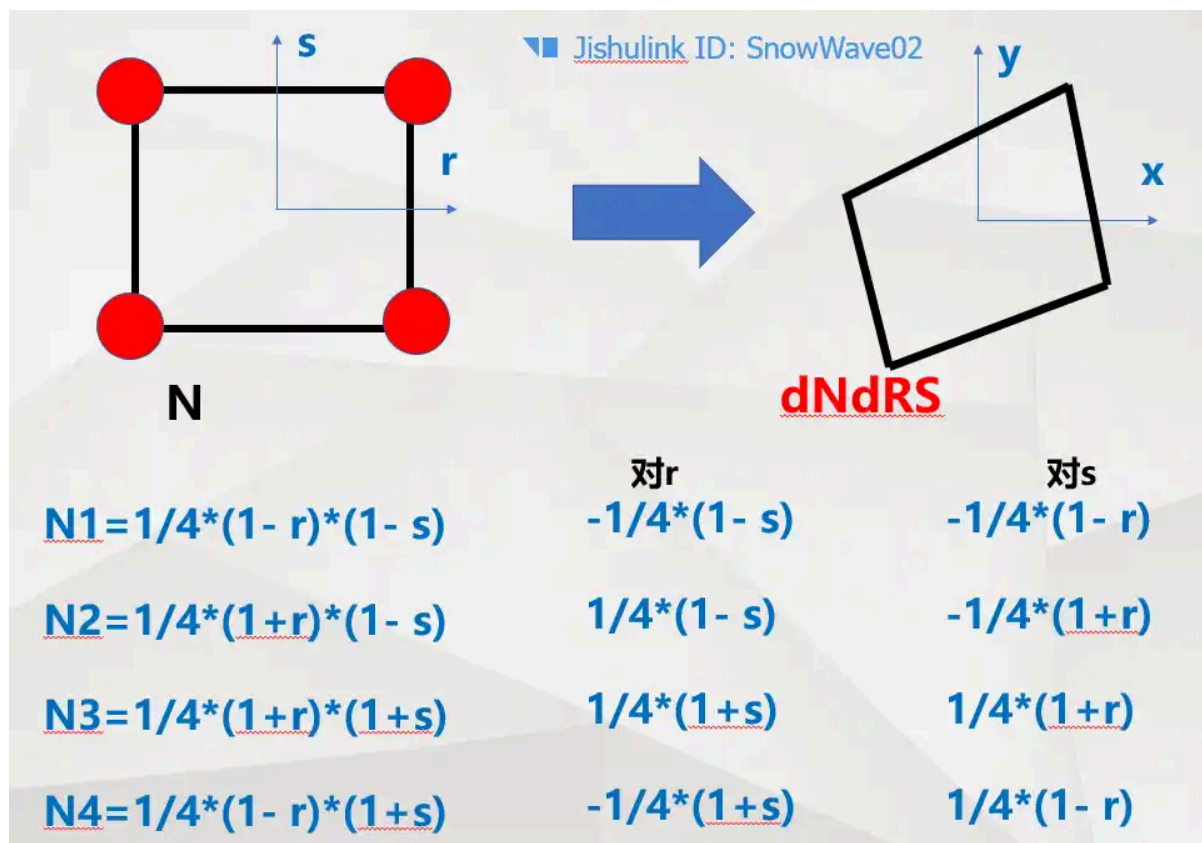


2、计算B矩阵  2. Calculate the B matrix

$$B = \begin{pmatrix} \dfrac{\partial N(i)}{\partial x} & 0 \\ 0 & \dfrac{\partial N(i)}{\partial y} \\ \dfrac{\partial N(i)}{\partial y} & \dfrac{\partial N(i)}{\partial x} \end{pmatrix}$$

3、计算D矩阵  3. Calculate the D matrix

$$D = \frac{E}{(1-\mu^2)}\begin{bmatrix} 1 & \mu & 0 \\ \mu & 1 & 0 \\ 0 & 0 & \frac{(1-\mu)}{2} \end{bmatrix}$$

4、计算刚度矩阵  4. Calculate the stiffness matrix

**K=B′*D*B*wt*h*det(Jacob)**

**K = B' * D * B * wt * h * det(Jacob)**

5、计算不平衡力  5. Calculate unbalanced force

**RHS=RHS-K*U  RHS = RHS - K * U**

完成以后保存for文件。编程过程中有不明白的问题可以参照我们的视频教程：

Save the for file after completion. If you have any questions during the programming process, refer to our video tutorials:

http://www.jishulink.com/college/video/c14948

![有限元理论基础及Abaqus内部实现方式研究系列23：编写简单面内拉伸问题UEL Step By的图17]
**1.3.2 UEL运行和调试  1.3.2 UEL Execution and Debugging**

UEL运行和单步调试的方法也可见我们之前的文章：《有限元理论基础及Abaqus内部实现方式研究系列20：UEL用户子程序开发步骤》，涉及到的单步调试插件DUS，见帖子：http://www.jishulink.com/content/post/424513。

The methods for running and single-step debugging of UEL can also be found in our previous articles: "Series 20: Steps for UEL User Subroutine Development in the Theory of Finite Element Method and Internal Implementation of Abaqus," involving the single-step debugging plugin DUS, see post: http://www.jishulink.com/content/post/424513.

![有限元理论基础及Abaqus内部实现方式研究系列23：编写简单面内拉伸问题UEL Step By的图18]
**1.3.3 结果对比  1.3.3 Result Comparison**

我们在运行完成之后，将Abaqus的S4R单元和我们编写的UEL单元运行的结果进行对比，如下图所示。结果上存在一些差异，这是由于Abaqus的S4R单元做了很多其它的工作，譬如厚度方向积分、基于工程经验的修正等。

After the run is completed, we compare the results of Abaqus' S4R element and the UEL element we have written, as shown in the figure below. There are some differences in the results, which are due to the fact that Abaqus' S4R element performs many other operations, such as integration in the thickness direction, and corrections based on engineering experience.
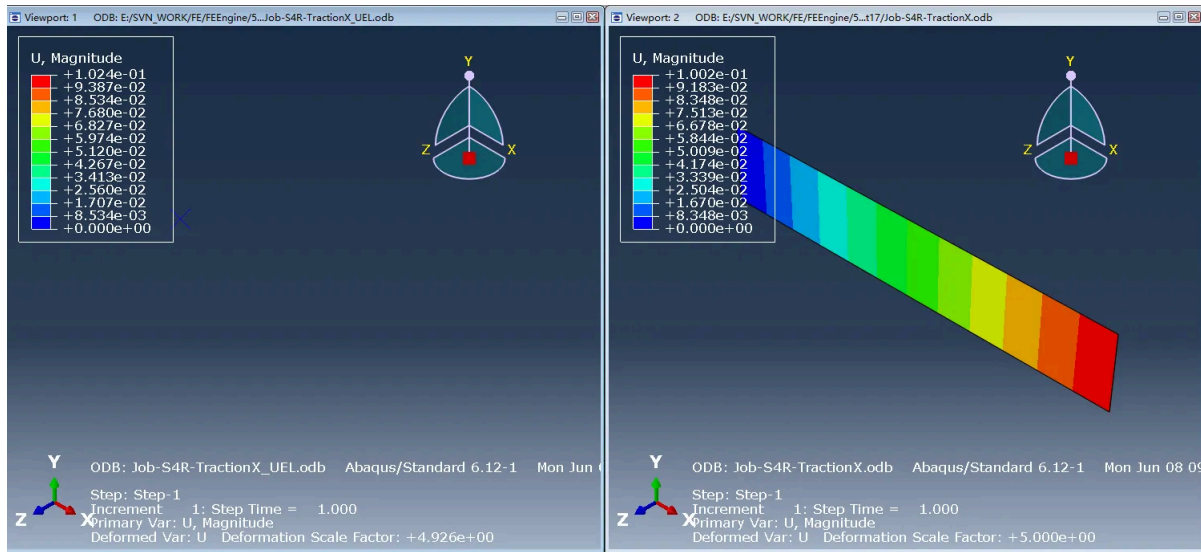
图 25 位移云图，左侧为UEL，右侧为S4R

Figure 25 displacement cloud diagram, left for UEL, right for S4R


有限元理论基础及Abaqus内部实现方式研究系列23： 编写简单面内拉伸问题UEL Step By的图21


有限元理论基础及Abaqus内部实现方式研究系列23： 编写简单面内拉伸问题UEL Step By的图22

## ==总结==  ==Summary==

本文通过简单面内拉伸问题UEL的编写，介绍解决有限元问题从理论到算法再到编程实现的一般流程。文中介绍了有限元方法的基本步骤，面内拉伸问题的基本算法步骤，旨在让大家对这些基本流程有初步的认识和了解，为后续的系列文章和视频打下基础。

This article introduces the general process from theory to algorithm and programming implementation in solving finite element problems through the writing of a simple in-plane tensile problem UEL. The article describes the basic steps of the finite element method, the basic algorithmic steps for the in-plane tensile problem, aiming to provide a preliminary understanding and knowledge of these basic processes, laying the foundation for subsequent series of articles and videos.

如果有任何其它疑问或者项目合作意向，也欢迎联系我们：

If you have any other questions or intentions for project cooperation, feel free to contact us:

SnowWave02Fromwww.jishulink.com

email: snowwave02@qq.com


有限元理论基础及Abaqus内部实现方式研究系列23： 编写简单面内拉伸问题UEL Step By的图23


有限元理论基础及Abaqus内部实现方式研究系列23： 编写简单面内拉伸问题UEL Step By的图24

以往的系列文章：　　Previous series articles:

**有限元理论基础及Abaqus内部实现方式研究系列23： 编写简单面内拉伸问题UEL Step By的图25**

**1.5.1 ========第一阶段========**

**1.5.1 ========First Phase========**

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。

First article: Research on the Stiffness Matrix of S4 Shell Element. Introduces the correction of Abaqus' S4 stiffness matrix in the theory of ordinary thick shell.

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。

Second article: Research on the Mass Matrix of S4 Shell Element. Introduces the mass matrices of Abaqus' S4 and Nastran's Quad4 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**。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。

Third article: Shear locking and hourglass control of S4 shell elements. Introduces how Abaqus S4 elements eliminate shear locking and how S4R suppresses hourglassing.

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

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

Fourth article: Solution of nonlinear problems. This article introduces the numerical computation methods adopted by Abaqus in nonlinear analysis.

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。

Fifth article: Element correctness verification. Introduces the verification methods for finite element element correctness, compares the analysis results of the self-developed structural solver program iSolver with Abaqus through multiple examples, thereby illustrating the entire correctness verification process and the correctness of the iSolver results.

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

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梁单元完全一致的刚度矩阵。

Sixth article: Stiffness matrix of General beam element. Introduces the basic theory of beam elements and the correction methods of the General beam element stiffness matrix in Abaqus. By using these correction methods, it is possible to obtain a stiffness matrix that is completely consistent with the Abaqus beam element.

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

第七篇：**C3D8六面体单元的刚度矩阵**。介绍六面体单元的基础理论和Abaqus中C3D8R六面体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六面体单元完全一致的刚度矩阵。

Seventh article: Stiffness matrix of C3D8 hexahedral element. Introduces the basic theory of hexahedral elements and the correction methods of the C3D8R hexahedral element stiffness matrix in Abaqus. By using these correction methods, it is possible to obtain a stiffness matrix that is completely consistent with the Abaqus hexahedral element.

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

第八篇：**UMAT用户子程序开发步骤**。介绍基于Fortran和Matlab两种方式的Abaqus的UMAT的开发步骤，对比发现开发步骤基本相同，同时采用Matlab更加高效和灵活。

Eighth article: Steps for UMAT user subroutine development. Introduces the development steps of Abaqus UMAT based on both Fortran and Matlab, and finds that the development steps are basically the same. At the same time, Matlab is found to be more efficient and flexible.

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

第九篇：

有限元理论基础及Abaqus内部实现方式研究系列23： 编写简单面内拉伸问题UEL Step By的图26
有限元理论基础及Abaqus内部实现方式研究系列23： 编写简单面内拉伸问题UEL Step By的图27
有限元理论基础及Abaqus内部实现方式研究系列23： 编写简单面内拉伸问题UEL Step By的图28

**编写线性UMAT Step By Step**。介绍基于Matlab线性零基础，从零开始Step by Step的UMAT的编写和调试方法，帮助初学者UMAT入门。

Chapter 9: Writing Linear UMAT Step by Step. Introduces the writing and debugging methods of UMAT based on Matlab linear zero foundation, starting from scratch step by step to help beginners get started with UMAT.

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

第十篇：**耦合约束（Coupling constraints）的研究**。介绍Abaqus中耦合约束的原理，并使用两个简单算例加以验证。

Chapter 10: Research on Coupling Constraints. Introduce the principle of coupling constraints in Abaqus and verify it with two simple examples.

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

**有限元理论基础及Abaqus内部实现方式研究系列23： 编写简单面内拉伸问题UEL Step By的图29**
**1.5.2 ========第二阶段========**
**1.5.2 ========Second Stage========**

第十一篇：**自主CAE开发实战经验第一阶段总结**。介绍了iSolver开发以来的阶段性总结，从整体角度上介绍一下自主CAE的一些实战经验，包括开发时间预估、框架设计、编程语言选择、测试、未来发展方向等。

The eleventh article: Summary of the first phase of independent CAE development experience. It introduces the phase-by-phase summary of the development of iSolver, and gives an overall introduction to some practical experiences of independent CAE, including development time estimation, framework design, programming language selection, testing, and future development directions.

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

第十二篇：**几何梁单元的刚度矩阵**。研究了Abaqus中几何梁的B31单元的刚度矩阵的求解方式，以L梁为例，介绍General梁用到的面积、惯性矩、扭转常数等参数在几何梁中是如何通过几何形状求得的，根据这些参数，可以得到和Abaqus完全一致的刚度矩阵，从而对只有几何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。

Twelfth article: Stiffness Matrix of Geometric Beam Element. This article studies the method of solving the stiffness matrix of the B31 element of geometric beam in Abaqus, taking the L beam as an example, and introduces how the parameters such as area, moment of inertia, and torsion constant used in General beam are obtained through geometric shape in geometric beam. Based on these parameters, a stiffness matrix consistent with Abaqus can be obtained, so that for any model composed only of

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

第十三篇：**显式和隐式的区别**。介绍了显式和隐式的特点，并给出一个数学算例，分别利用前向欧拉和后向欧拉求解，以求直观表现显式和隐式在求解过程中的差异，以及增量步长对求解结果的影响。

Thirteenth article: The difference between explicit and implicit. It introduces the characteristics of explicit and implicit methods, and provides a mathematical example, using forward Euler and backward Euler methods respectively to solve, in order to intuitively demonstrate the differences between explicit and implicit methods in the solution process, as well as the influence of the increment step size on the solution results.

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

第十四篇：**壳的应力方向**。简单介绍了一下数学上张量和Abaqus中壳的应力方向，并说明Abaqus这么选取的意义，最后通过自编程序iSolver来验证壳的应力方向的正确性。

14th article: Stress direction of shells. A brief introduction to the tensor of stress direction in mathematics and in Abaqus, and an explanation of the significance of Abaqus's selection, and finally, the correctness of the stress direction of shells is verified through the self-written program iSolver.

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

第十五篇：**壳的剪切应力**。介绍了壳单元中实际的和板壳近似理论中的剪切应力，也简单猜测了一下Abaqus的内部实现流程，最后通过一个算例来验算Abaqus中的真实的剪切应力。

15th article: Shear Stress of Shell. Introduces the shear stress in actual shell elements and in the plate-shell approximate theory, also makes a simple guess about the internal implementation process of Abaqus, and finally verifies the actual shear stress in Abaqus through a calculation example.

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

第十六篇：**Part、Instance与Assembly**。介绍了Part、Instance与Assembly三者之间的关系，分析了Instance的网格形成原理，并猜测Abaqus的内部组装实现流程，随后针对某手机整机多part算例，通过自编程序iSolver的结果比对验证我们的猜想。

Chapter 16: Part, Instance, and Assembly. Introduces the relationship between Part, Instance, and Assembly, analyzes the principle of grid formation of Instance, and guesses the internal assembly implementation process of Abaqus. Subsequently, for a multi-part assembly example of a mobile phone, the results of the self-written program iSolver are compared and verified to confirm our conjecture.

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

第十七篇：**几何非线性的物理含义**。介绍了几何非线性的简单的物理含义，并通过几何非线性的悬臂梁Abaqus和iSolver的小应变情况的结果，从直观上理解几何非线性和线性的差异。

Chapter 17: Physical Meaning of Geometric Nonlinearity. Introduces the simple physical meaning of geometric nonlinearity and illustrates the difference between geometric nonlinearity and linearity through the results of small strain of the cantilever beam with geometric nonlinearity in Abaqus and iSolver.

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

第十八篇：**几何非线性的应变**。首先从位移、变形和应变的区别说起，然后通过一维的简单例子具体介绍了几何非线性下的应变的度量方式，并给出了工程应变、真实应变、Green应变三者一维情况下在数学上的表达方式。

Chapter 18: Strain under Geometric Nonlinearity. Firstly, the differences between displacement, deformation, and strain are discussed, followed by a specific introduction to the measurement methods of strain under geometric nonlinearity through a one-dimensional example, and the mathematical expressions of engineering strain, true strain, and Green strain under one-dimensional conditions are given.

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

第十九篇：**Abaqus几何非线性的设置和后台**。首先介绍了几何非线性一般的分类，然后详细说明了Abaqus中几何非线性的设置方式和常用单元的分类，最后以一个壳单元的简单算例为对象，可以发现应变理论、Abaqus和iSolver三者在线性、小应变几何非线性和大应变几何非线性三种情况下都完全一致，从而验证Abaqus几何非线性后台采用的应变和我们的预想一致。

Chapter 19: Abaqus Geometric Nonlinearity Settings and Background. First, a general classification of geometric nonlinearity is introduced, followed by a detailed explanation of the setting methods for geometric nonlinearity in Abaqus and the classification of commonly used elements. Finally, taking a simple shell element example, it can be found that the strain theory, Abaqus, and iSolver are completely consistent in linear, small strain geometric nonlinearity, and large strain geometric nonlinearity, thus verifying that the strain adopted by the Abaqus geometric nonlinearity background is consistent with our expectations.

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

第二十篇：**UEL用户子程序开发步骤**。本文首先简单的讨论了UEL的一般含义，并详细的介绍了基于Fortran和Matlab两种方式的UEL的开发步骤，对比发现开发步骤基本相同，但Matlab更加高效和灵活。

第二十篇：UEL 用户子程序开发步骤。本文首先简单的讨论了 UEL 的一般含义，并详细的介绍了基于 Fortran 和 Matlab 两种方式的 UEL 的开发步骤，对比发现开发步骤基本相同，但 Matlab 更加高效和 flexible.

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

![有限元理论基础及Abaqus内部实现方式研究系列23：编写简单面内拉伸问题UEL Step By的图30]**有限元理论基础及Abaqus内部实现方式研究系列23： 编写简单面内拉伸问题UEL Step By的图30**

**1.5.3 ========第三阶段========**

第二十一篇：**自主CAE开发实战经验第二阶段总结**。从实战角度介绍自主CAE在推广和工程化应用的过程中的体会，同时说明一个CAE平台最重要的两个特点：可扩展和易维护。

Chapter 21: Summary of the Second Stage of Autonomous CAE Development Practice. Introduce the experience in promoting and engineering the application of autonomous CAE from a practical perspective, and at the same time, explain the two most important characteristics of a CAE platform: scalability and ease of maintenance.

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

第二十二篇：**几何非线性的刚度矩阵求解。**介绍几何非线性下的刚度矩阵的理论推导和计算机求解方法，最后利用一个简单的算例验证我们对Abaqus几何非线性的刚度矩阵的实现方式的猜测。

Chapter 22: Solution of Stiffness Matrix under Geometric Nonlinearity. Introduce the theoretical derivation and computer solution method of the stiffness matrix under geometric nonlinearity, and finally verify our guess about the implementation method of Abaqus geometric nonlinearity stiffness matrix through a simple example.

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

## 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… | 非局部均值滤波和MATLAB程序详解视频算法及其保留图形细节应用… | 车身设计系列视频之车身钣金正向设计实例教程… |
| SnowWave02            免费 Free | 技术邻小李  Technical Neighbor Xiao Li    ¥100  100 Yuan | 正一算法程序  Zhengyi Algorithm Program    ¥220  220 Yuan | 京迪轩  Jing Di Xuan            ¥1 |