有限元理论基础及Abaqus内部实现方式研究系列33： 线性瞬态动力学

Theoretical Foundation of Finite Element Method and Research on the Internal Implementation of Abaqus Series 33: Linear Transient Dynamics

SnowWave02

2021年1月26日 12:18 January 26, 2021 12:18

（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33： 线性瞬态动力学的图1

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 Operation

（3） 自编程序 (3) Self-written program

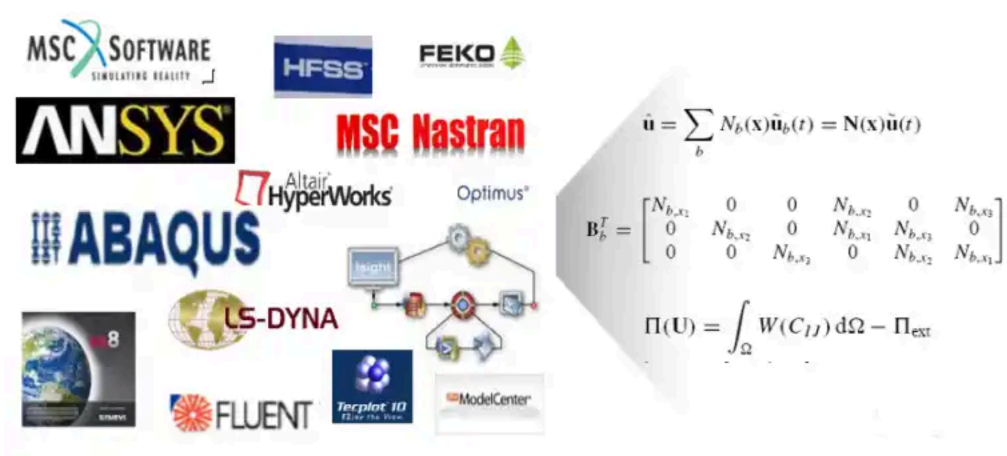
三者结合的方式将复杂繁琐的结构有限元理论通过简单直观的方式展现出来，同时深层次的学习有限元理论和商业软件的内部实现原理。

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。

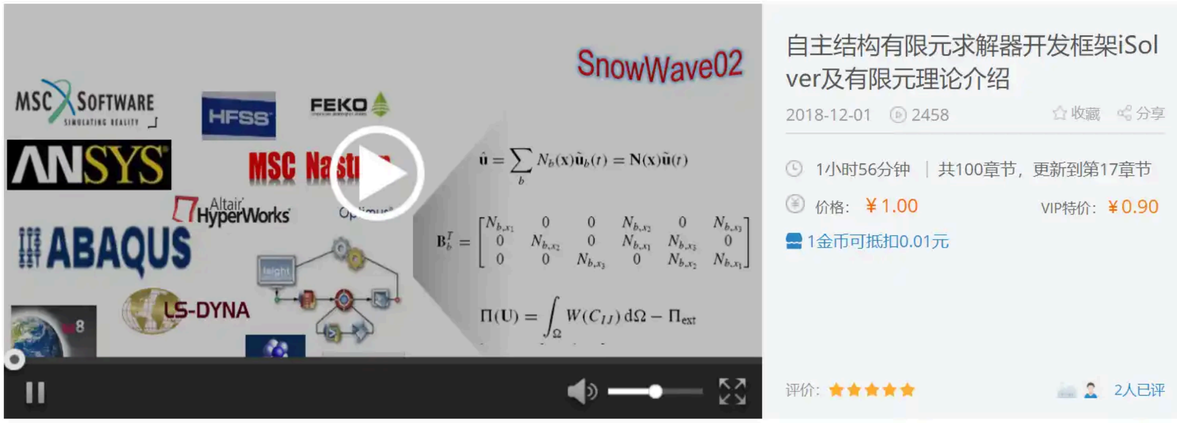
The theoretical development of finite elements has matured over decades, and commercial finite element software also adopts these mature finite element theories. However, in the actual application process, commercial CAE software will make corresponding corrections on the basis of traditional theories 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.



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.

通用结构有限元软件iSolver介绍视频： Introduction video of the general finite element software iSolver

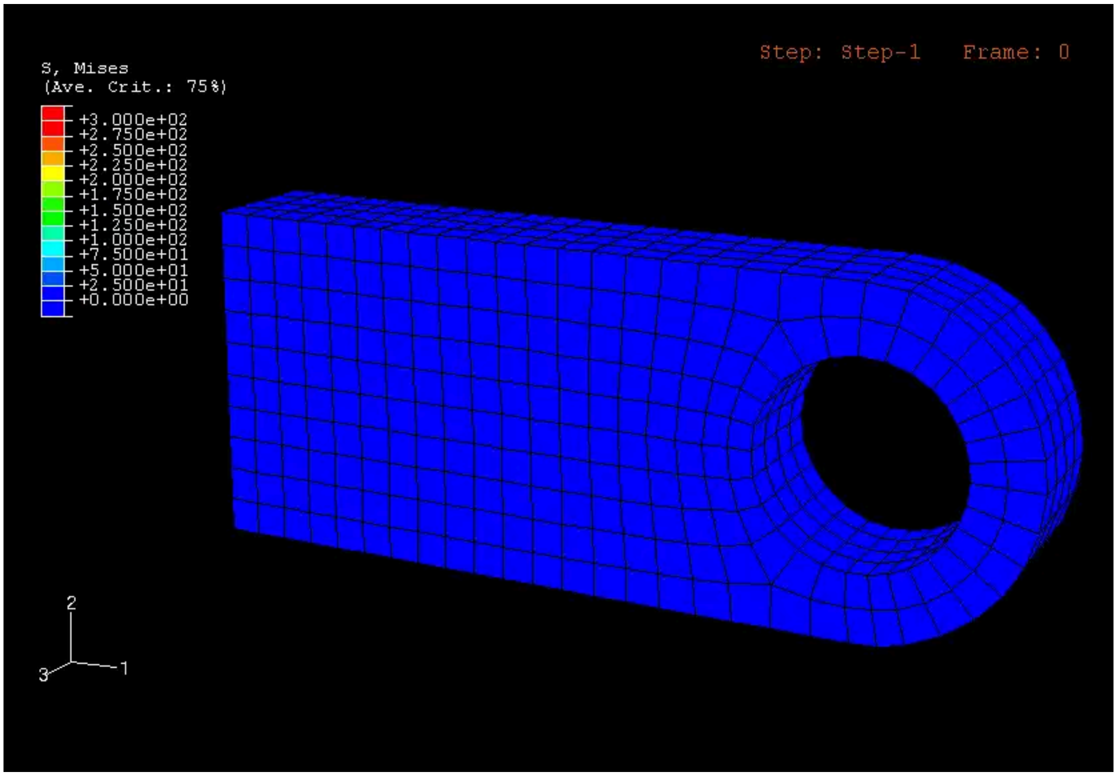


http://www.jishulink.com/college/video/c12884

==第32篇：线性瞬态动力学== ==Article 32: Linear Transient Dynamics==

在系列文章第三十篇、三十二篇时，我们介绍了稳态动力学的原理和算法，本章继续介绍瞬态动力学。稳态和瞬态都是在随时间变化的激励作用下产生的运动形式，其中，稳态是时间足够长后，系统慢慢趋于稳定后的运动，譬如当不考虑阻尼时弹簧受瞬间载荷力作用时间足够长后最后会做简谐振动，由于简谐振动的理论公式比较简单，一个简谐振动在时间域上虽然很长，但只要有频率和振幅参数就能完全表示，所以可以跳过前面的不稳定的过程而直接求解最终的稳态形式。但达到稳态前的瞬态过程中间每个时刻点的运动状态是不一样的，无法简单的用几个参数表示，当前时刻的运动状态需要前面运动状态往前推进得到，此时只能用瞬态分析。

In the 30th and 32nd articles of the series, we introduced the principles and algorithms of steady-state dynamics. This chapter continues to introduce transient dynamics. Both steady-state and transient dynamics are motion forms produced under the action of time-varying excitation. Steady-state is the motion that the system gradually tends to stabilize after a sufficiently long time, for example, when the damping is not considered, after a spring is subjected to an instantaneous load force for a sufficiently long time, it will finally perform simple harmonic vibration. Since the theoretical formula of simple harmonic vibration is relatively simple, a simple harmonic vibration can be completely represented by the frequency and amplitude parameters, even if it is very long in the time domain. Therefore, it is possible to skip the previous unstable process and directly solve the final steady-state form. However, the motion state at each moment during the transient process before reaching the steady state is different and cannot be simply represented by a few parameters. The motion state at the current moment needs to be pushed forward from the previous motion state, and at this time, only transient analysis can be used.



瞬态分析有线性和非线性之分，从线性瞬态动力学出发其实更容易理解瞬态动力学的求解方式。本章我们将简单介绍一下瞬态动力学的求解公式，并以一个单摆例子来说明线性瞬态动力学在有限元软件中的内部实现原理。非线性瞬态动力学的原理将在后面章节介绍。

Transient analysis has linear and nonlinear types. Starting from linear transient dynamics is actually easier to understand the solution method of transient dynamics. In this chapter, we will briefly introduce the solution formulas of transient dynamics and illustrate the internal implementation principle of linear transient dynamics in finite element software with an example of a simple pendulum. The principles of nonlinear transient dynamics will be introduced in later chapters.

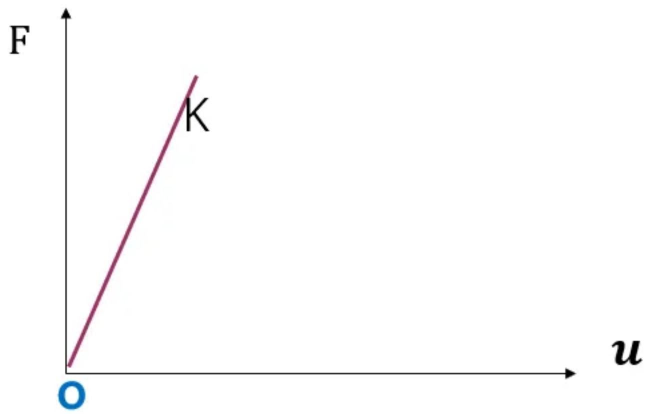
1.1 线性瞬态动力学的理论 1.1 Theory of Linear Transient Dynamics

1.1.1 线性瞬态动力学的方程 1.1.1 Equations of linear transient dynamics

线性瞬态动力学的运动方程和线性静力平衡方程很类似，对于线性静力问题载荷F和位移u是直线关系，可以非常简单的由一个状态类乘以一个系数推到另一个状态。因为静力学可以简单的表示为下方的线性表达形式：

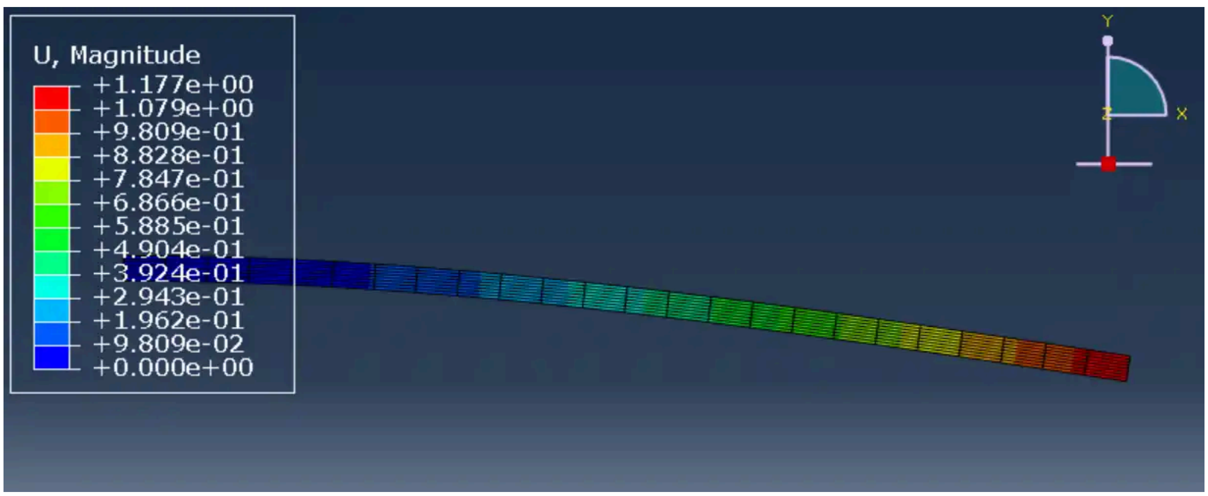
The motion equations of linear transient dynamics are very similar to those of linear static equilibrium, as for linear static problems, the load F and displacement u are in a linear relationship, which can be very simply derived from one state by multiplying a coefficient to another state. Because statics can be simply represented by the following linear expression:

$$Ku=R$$



譬如下面的悬臂梁问题，载荷1N的时候计算得到最大位移时1.177mm。

For example, in the case of a cantilever beam problem, when the load is 1N, the maximum displacement is calculated to be 1.177mm.



如果载荷是1000N时，不用计算也知道就是最大位移会变为1000倍=1177mm。

If the load is 1000N, it is unnecessary to calculate to know that the maximum displacement will become 1000 times = 1177mm.

但在线性动力学问题中分析复杂一点就是，如果力扩大了1000倍，显然最终的位移不是简单的乘上1000，瞬态动力学和上式原理上很类似，只不过包括了质量阵M相关的惯性力项和阻尼阵C相关的阻尼力项，如下：

In linear dynamic problems, analyzing something more complex is that if the force is increased by 1000 times, the final displacement is not simply multiplied by 1000. The principle of transient dynamics is similar to the above, but it includes inertia force terms related to the mass matrix M and damping force terms related to the damping matrix C, as follows:

$$M * \ddot{u} + C(t,u)\dot{u} + K(t,u)u = \mathrm{R(t)}$$

一般质量矩阵与时间和位移无关，而K和C如果和位移和时间无关，那么就是线性瞬态问题。为简单起见，我们不考虑阻尼项。得到如下表达式：

Generally, the mass matrix is independent of time and displacement, and if K and C are independent of displacement and time, then it is a linear transient problem. For simplicity, we do not consider the damping term. The following expression is obtained:

$$M * \ddot{u} + Ku = \mathrm{R(t)}$$

位移和加速度与t相关，上述方程就有两个未知数位移和加速度，只有一个方程是无法求出的，所以还有一个加速度和位移的约束关系，实际中必然是加速度先和速度相关，速度再和位移相关。

Displacement and acceleration are related to t, so the above equation has two unknowns: displacement and acceleration. With only one equation, it is impossible to solve for them. Therefore, there is another constraint relationship between acceleration and displacement. In practice, acceleration is necessarily related to velocity first, and velocity is then related to displacement.

1.1.2 线性瞬态动力学的有限元求解 1.1.2 Finite Element Solution of Linear Transient Dynamics

在理论上当前时刻的加速度就应该只和当前时刻的位移相关了，但在计算机的数值求解时，无法求出无限个连续时刻的结果，所以只能把整个时间人为的分成多个时间段，方程变为如下：

In theory, the acceleration at the current moment should only be related to the displacement at the current moment, but in the numerical solution on the computer, it is impossible to obtain the results of an infinite number of continuous moments, so the entire time must be divided into multiple time periods artificially, and the equation becomes as follows:

$$M * {}^{t+\Delta t}\ddot{u} + K\,{}^{t+\Delta t}u = {}^{t+\Delta t}\mathrm{R}$$

此时，加速度就必须要通过起码两个时刻的速度或者位移才能计算出来的，可以有不同的取法，取前时刻还是后时刻的位移来决定加速度就有隐式和显式之分（具体可看系列文章13：显式和隐式的区别），任何一种分析理论上都可以得到正确解，只不过显式不用求刚度，隐式需要求解刚度阵，而通过刚度的解释可以更容易理解Abaqus内部的实现原理，在本章中，我们选用隐式瞬态分析。

At this point, the acceleration must be calculated through at least two moments of velocity or displacement, which can be done in different ways. Whether to use the displacement of the previous moment or the next moment to determine the acceleration results in implicit and explicit methods (specifically see Series Article 13: The Difference Between Implicit and Explicit Methods). Any analysis can theoretically obtain the correct solution, but the explicit method does not require the calculation of stiffness, while the implicit method requires the solution of the stiffness matrix. Through the explanation of stiffness, it can be easier to understand the internal implementation principles of Abaqus. In this chapter, we choose implicit transient analysis.

在隐式方法中，将时间离散，加速度表示为位移的表达式后，得到下方的公式

In the implicit method, after discretizing time and expressing the acceleration as an expression of displacement, the following formula is obtained

$$\widehat{K}^{i,j+1} * \Delta u = R^i - \widehat{F}^{i,j}$$

其中K和F在静力学的增量迭代法中表示切线刚度阵和内力，只不过这边需要加上M项：

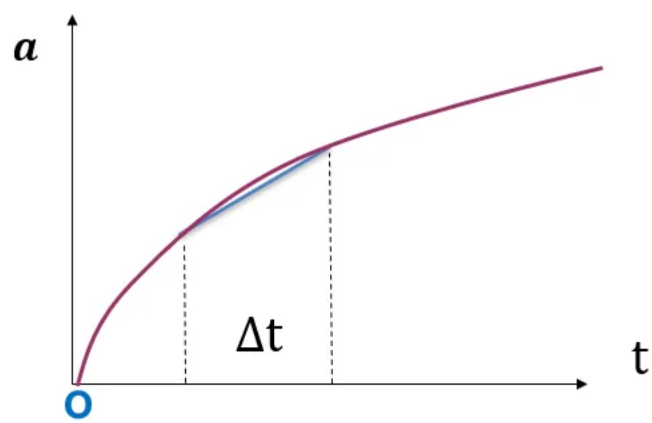
In the incremental iterative method of statics, K and F represent the tangent stiffness matrix and internal forces, respectively, but here the M term needs to be added

$$\widehat{K}^{i,j+1} = K + A * M$$

其中A是只与时间增量步相关的量，对不同的隐式算法A的值不同，譬如对最简单的Newmark方法中的梯形隐式算法，加速度在增量步内线性变化，下方红线是在输入的载荷转换为加速度的值，但计算中没法处理连续的曲线，所以Abaqus或者iSolver等有限元程序实际上只会认为增量步

内加速度时线性变化的：

Among them, A is a quantity related only to the time increment step, and the value of A is different for different implicit algorithms. For example, in the simplest Newmark method's trapezoidal implicit algorithm, the acceleration varies linearly within the increment step. The red line below represents the value of the load converted to acceleration, but the calculation cannot handle continuous curves, so finite element programs like Abaqus or iSolver actually only recognize that the acceleration varies linearly within the increment step:



此时得到的修正刚度为： The corrected stiffness obtained at this point is:

$$\widehat{K}^{i,j+1} = K + \frac{4}{\Delta t^2} M$$

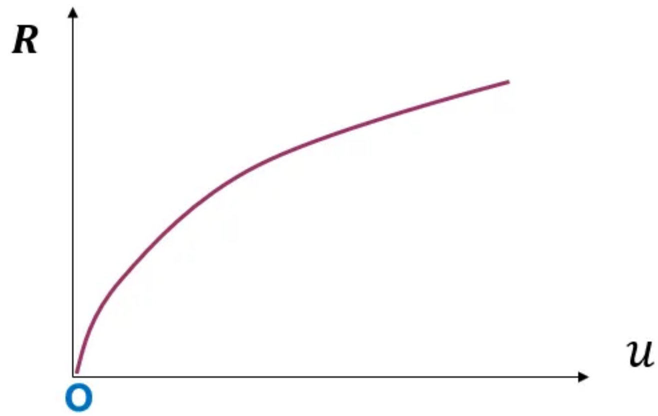
显然当时间增量步固定时，此时的K也是个定值。而F与上一时刻的包括惯性力在内的内力，与位移、速度、加速度相关，可以表示为：

It is obvious that when the time increment step is fixed, the K at this point is also a constant. The F, including the internal forces such as inertial forces at the previous moment, is related to displacement, velocity, and acceleration and can be expressed as:

$$\widehat{F}^{i,j} = \left(B1 * u^{i,j} + B2 * \dot{u}^{i,j} + B3 * \ddot{u}^{i,j}\right) * M + F^{i,j}$$

其中B是只与时间增量步相关的量。虽然固定步长K是常量，但右端量还包括了上一个时刻位移、速度等变量，R和u依然为非线性曲线，如下：

Among them, B is a quantity related only to the time increment step. Although the fixed step length K is a constant, the right-hand side also includes variables such as displacement and velocity at the previous moment, and R and u are still nonlinear curves, as follows:

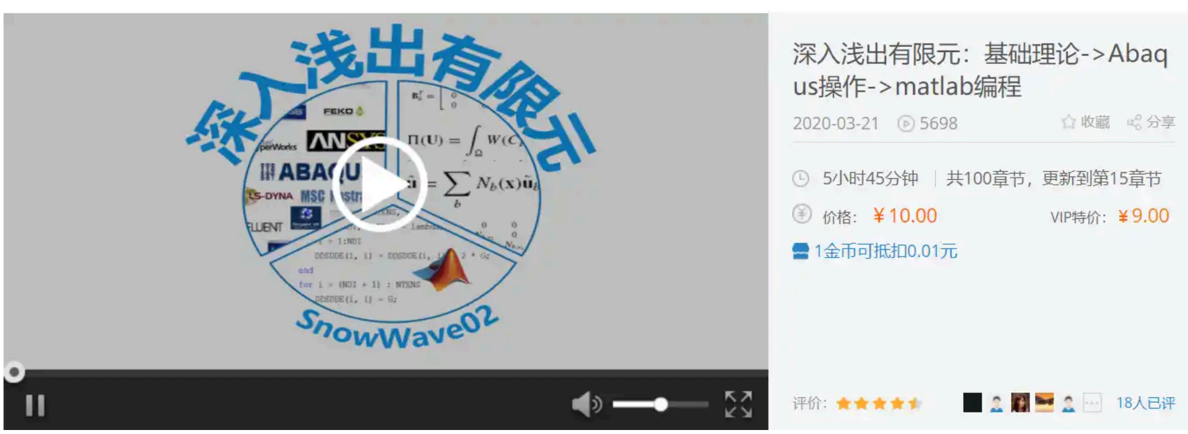


由上面的表达式，K和F只与增量步I,j相关，但实际上只要一次迭代左右就能相等平衡了，这个原理就和线性弹性力学的内力F=KU，到时右端在第一个迭代完成后就和外力R一样了，具体可看下面的课程推导：

From the above expression, K and F are only related to the increment steps I and j, but in fact, they can be balanced equally after just one iteration on both sides. This principle is similar to the internal force F=KU in linear elastic mechanics, where after the first iteration, the right side will be the same as the external force R. The specific derivation can be seen in the following course

https://www.jishulink.com/college/video/c14948 深入浅出有限元：基础理论->Abaqus操作->matlab编程

https://www.jishulink.com/college/video/c14948 An Easy-to-Understand Introduction to Finite Element Method: Basic Theory -> Abaqus Operation -> Matlab Programming



也就是说线性瞬态动力学方程与迭代步j无关，无须迭代，方程可以简单的写成：

That is, the linear transient dynamics equation is independent of the iteration step j, and there is no need for iteration; the equation can be simply written as:

$$\widehat{K}^{i} * \Delta u = R^{i} - \widehat{F}^{i-1}$$

而且只要迭代一次，而且内力项在第一次是位移为0，那么右端F就可以简单的写成下方表达式：

Moreover, if only one iteration is performed, and the internal force term is zero at the first iteration, then the right-hand side F can be simply written as the following expression:

$$\widehat{F}^{i,j} = \left(B1 * u^{i,j} + B2 * \dot{u}^{i,j} + B3 * \ddot{u}^{i,j}\right) * M$$

1.2 线性瞬态动力学的单摆模型 1.2 Single pendulum model of linear transient dynamics

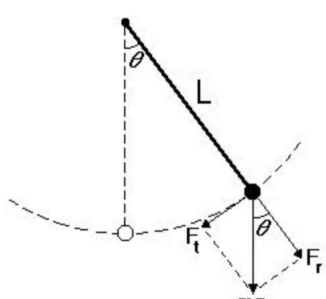
下面我们将以一个简单的单摆模型例子来说明上述线性瞬态动力学的解法。

Next, we will illustrate the solution method of the above linear transient dynamics with a simple pendulum model example.

1.2.1 实际单摆模型 1.2.1 Actual Simple Pendulum Model

实际的单摆如下图，线长L，无质量，末端绑定一个质量块m，如果没有空气阻力，在重力加速度g作用下往复运动。我们取初始时刻绳索在水平位置。假定绳索足够硬，在小球作用下变形非常小。显然，这个问题是一个大位移大转动小应变的几何非线性问题（具体可看我们的系列文章18：几何非线性的应变），这个小球位置随时间变化的运动过程就是非线性的瞬态过程。

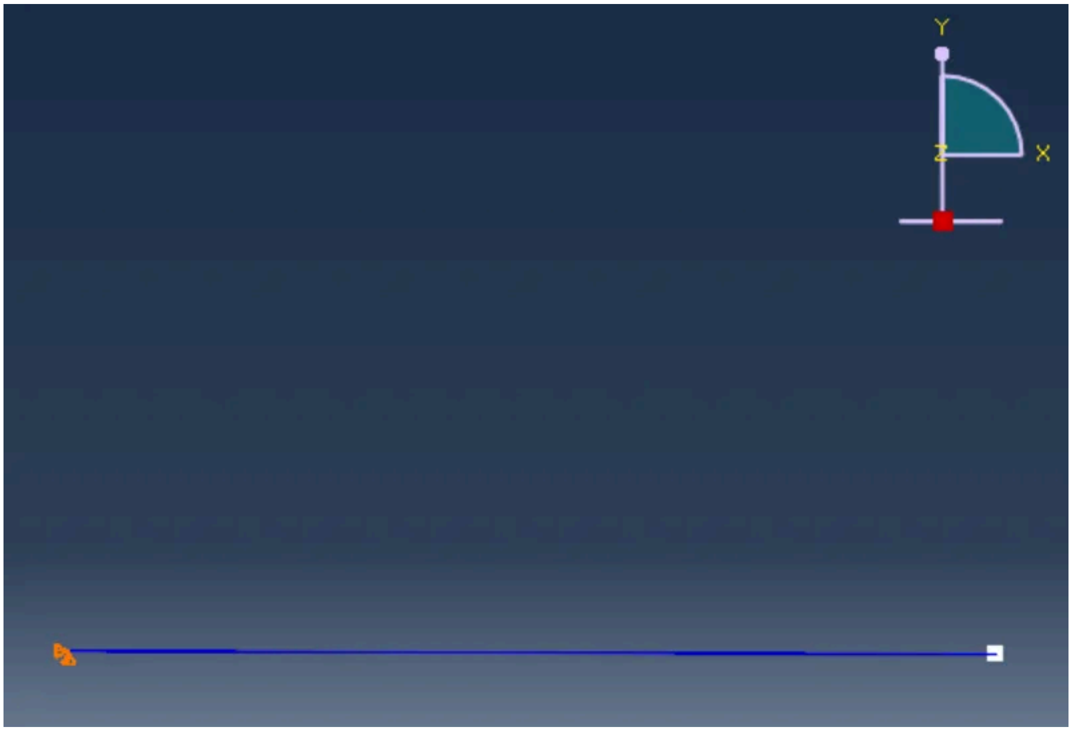
The actual simple pendulum is shown as follows, with a string length L, no mass, and an end bound to a mass block m. If there is no air resistance, it will oscillate back and forth under the action of gravity acceleration g. We take the string in a horizontal position at the initial moment. Assuming the string is sufficiently rigid, the deformation is very small under the action of the ball. It is obvious that this problem is a geometrically nonlinear problem with large displacements, large rotations, and small strains (specifically see our series article 18: Geometric Nonlinear Strain). The motion process of this ball's position changing over time is a nonlinear transient process.



1.2.2 仿真模型 1.2.2 Simulation Model

Abaqus中没有绳索单元，我们用Truss杆单元模拟

Abaqus does not have a rope element, so we use Truss bar elements to simulate



参数如下： Parameters as follows:

>尺寸：L=300，Truss截面积1。 >Size: L=300, Truss cross-sectional area 1.

>材料：Young’s Modulus 1e10， Poisson Ratio 0。

>Material: Young's Modulus 1e10, Poisson's Ratio 0.

>边界：左侧节点固支。 >Boundary: Fixed support at the left node.

>载荷：右侧节点加集中Mass=10，加速度取980，-Y方向。

>Load: Concentrated mass of 10 added to the right node, acceleration taken as 980, -Y direction.

>网格：划分为一个Truss单元。 Mesh: Divided into a Truss element.

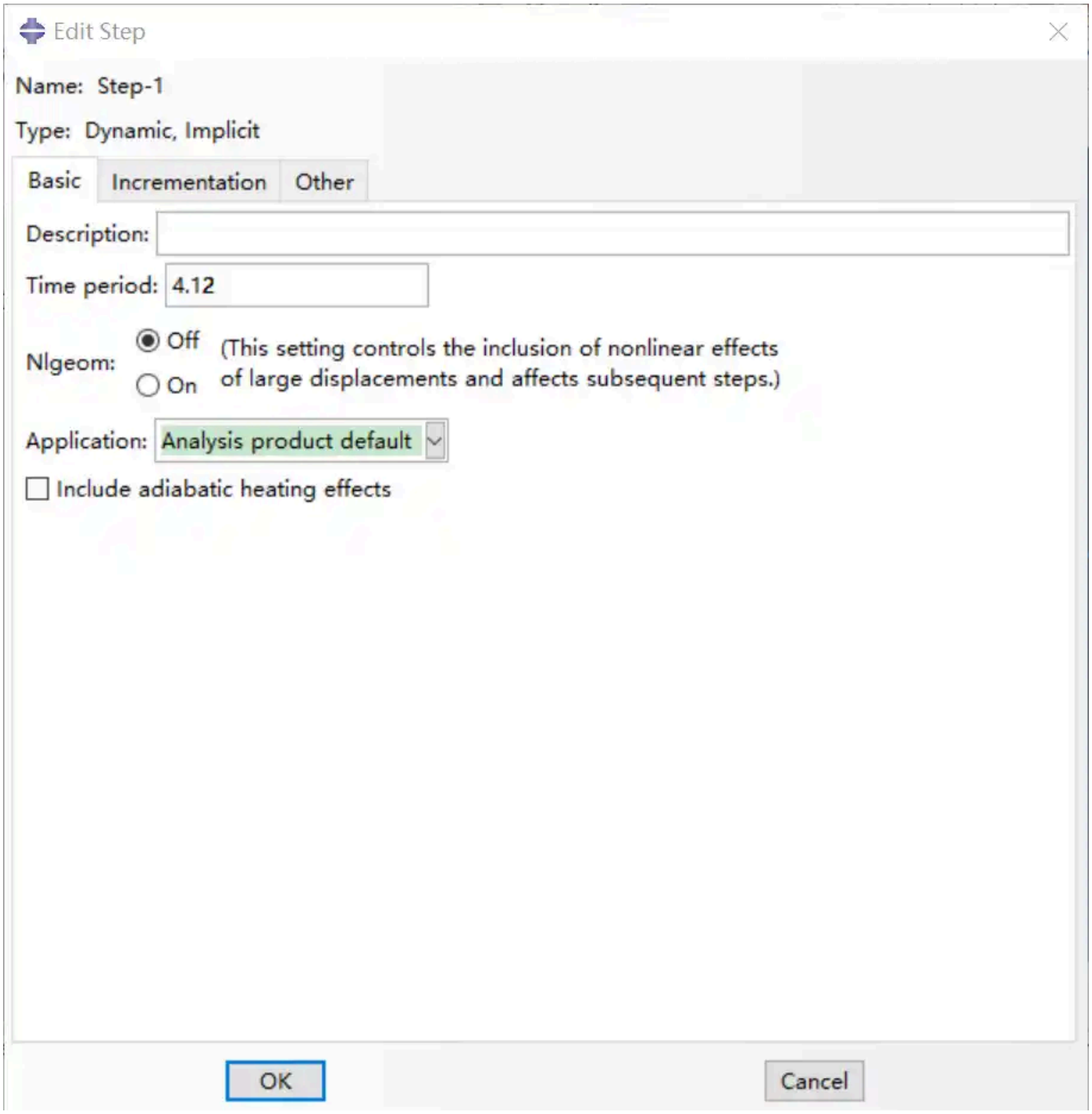
>分析步：时间长度取单摆从水平开始回到水平位置一个周期的时间。单摆运动如果摆动幅度很小时，周期可由下式得到，与材料和小球质量无关，这其实也是在地球上的所有钟摆只需通过设置绳长就能精确计时的原理。

Analysis step: The time length is taken as the time for the pendulum to swing from horizontal to horizontal position for one cycle. If the swing amplitude of the pendulum is small, the period can be obtained from the following formula, which is independent of the material and mass of the small ball. This is actually the principle that all pendulums on Earth can be precisely timed by setting the rope length.

$$T0 = 2\pi\sqrt{L/g} = 3.47s$$

在我们这个例子中，由于是从水平位置开始摆动，所以周期略大于此值，查文献可知周期T=4.12s，增量步固定为0.1。

In our example, since the pendulum starts from the horizontal position, the period is slightly larger than this value. According to the literature, the period T=4.12s, and the increment step is fixed at 0.1.



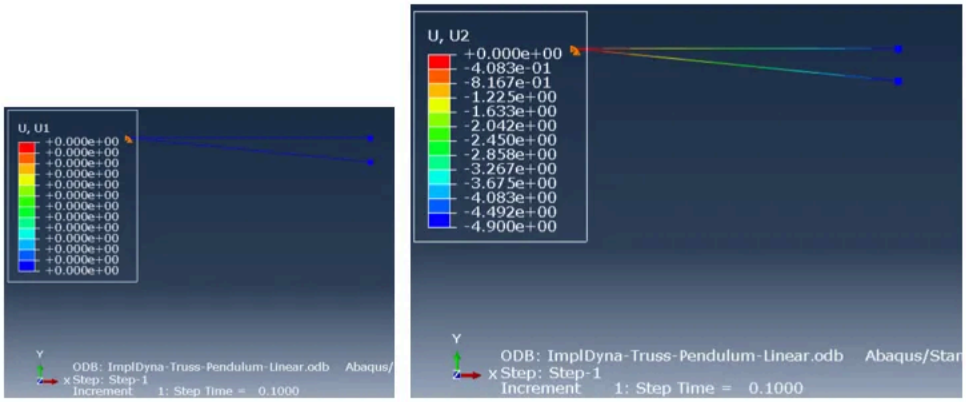
我们研究线性情况下是什么变化过程。在Abaqus中NLGeom=Off，取线性分析。

We study what the change process is under linear conditions. In Abaqus, NLGeom=Off is set, and linear analysis is taken.

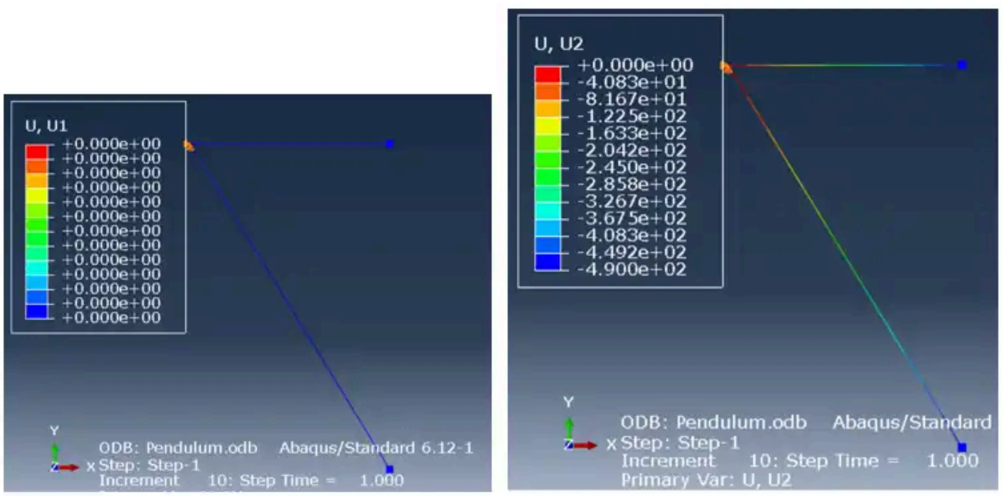
1.2.3 计算结果 1.2.3 Calculation results

分析后得到U1、U2在某两个特定时刻结果如下： The analysis results show the following values for U1 and U2 at two specific moments:

0.1s时刻： At 0.1s moment:



1s时刻： At 1s moment:



由上图可知U1永远都是0，而U2会随着时间不断的增加，也就是说不会产生实际的单摆运动。

From the figure above, it can be seen that U1 is always 0, while U2 continuously increases over time, which means that actual simple pendulum motion does not occur.

1.2.4 结果解析 1.2.4 Result Analysis

对一个truss单元来说，它的每个节点都是三个平动自由度，那么两节点truss的K就是一个6X6的矩阵，同时由于左端固支，所以，只剩右端的三个自由度相关的刚度阵，同时，因为我们几何非线性没开启，那么刚度都是初始时刻水平位置的值，此时只有K11有值，质量单元为右端节点的集中质量得到，如下：

For a truss element, each node has three translational degrees of freedom. Therefore, the stiffness matrix K for a two-node truss is a 6x6 matrix. Since the left end is fixed, only the three degrees of freedom related to the right end remain, and the stiffness matrix is the value at the initial horizontal position since geometric nonlinearity is not enabled. At this time, only K11 has a value, and the mass element is obtained from the concentrated mass at the right end node.

$$\widehat{K} = \begin{array}{ccc} K_{11} + A * M & 0 & 0 \\ 0 & A * M & 0 \\ 0 & 0 & A * M \end{array}$$

在第一个增量步，方程就变为： In the first increment step, the equation becomes:

$$\begin{array}{c} (K_{11} + A * M) * \Delta u1^1 \\ (A * M) * \Delta u2^1 \\ (A * M) * \Delta u3^1 \end{array} = \begin{array}{c} R1 - \left(B1 * u1^0 + B2 * \dot{u}1^0 + B3 * \ddot{u}1^0\right) * M \\ R2 - \left(B1 * u2^0 + B2 * \dot{u}2^0 + B3 * \ddot{u}2^0\right) * M \\ R3 - \left(B1 * u3^0 + B2 * \dot{u}3^0 + B3 * \ddot{u}3^0\right) * M \end{array}$$

由于R只有Y方向恒值，初始位置时位移、速度都是0，但Y方向存在加速度，显然，上述三个方程变为：

Since R only has a constant value in the Y direction, at the initial position, both displacement and velocity are 0, but there is acceleration in the Y direction. It is obvious that the above three equations become:

$$\begin{array}{c} (K_{11} + A * M) * \Delta u1^1 \\ (A * M) * \Delta u2^1 \\ (A * M) * \Delta u3^1 \end{array} = \begin{array}{c} 0 \\ R - B3 * \ddot{u}2^0 \\ 0 \end{array}$$

U1、U2、U3没有耦合项，可以直接解耦，分别得到三个方程。

U1, U2, and U3 have no coupling terms and can be decoupled directly, resulting in three separate equations.

由第一个和第三个方程可得位移U1和U3在第一个增量步结束时=0。第二个方程可得在第一个增量步结束时的位移是R/(A*M)，将Newmark的梯形算法的A带入后得到：

From the first and third equations, the displacements U1 and U3 at the end of the first increment step are equal to 0. The second equation gives the displacement at the end of the first increment step as R/(A*M). Substituting A from Newmark's trapezoidal algorithm gives:

$$\Delta u2^1 = \frac{1}{2} * g * \Delta t^2$$

这个公式显然和我们熟知的位移和加速度的关系式完全一致。

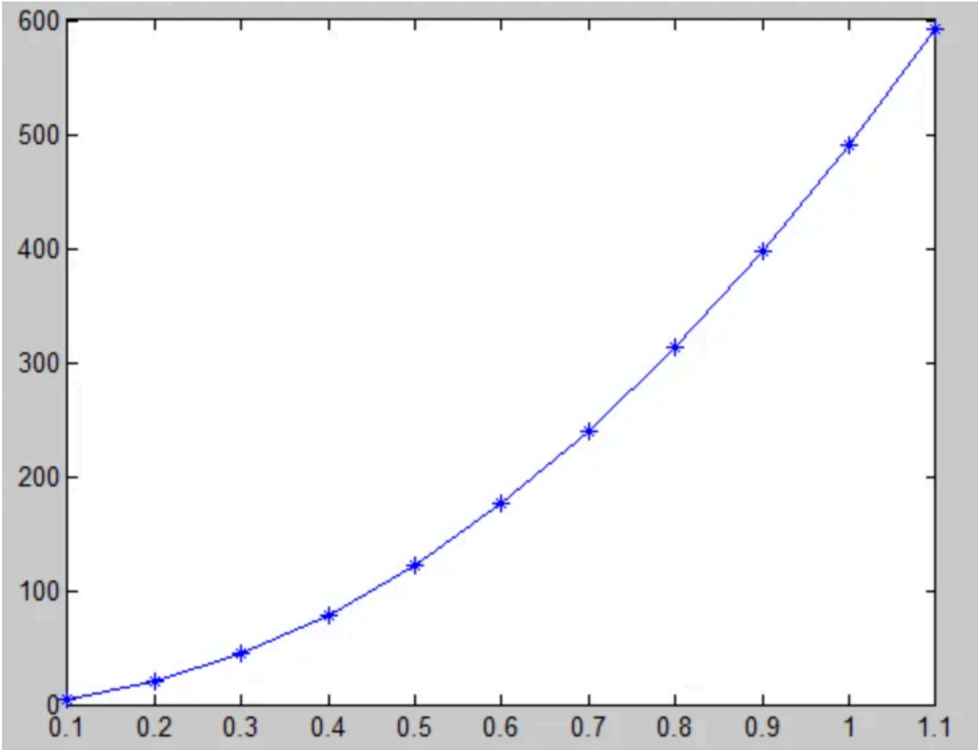
This formula is obviously completely consistent with the relationship between displacement and acceleration that we are familiar with.

在0.1s，得到位移U2=-4.9，和Abaqus完全一致。

At 0.1s, the displacement U2 is -4.9, which is consistent with Abaqus.

其它增量步类似，我们就不再累述，最终我们可以得到一个位移U2随时间的变化曲线如下：

The other incremental steps are similar, and we will not elaborate further. Finally, we can obtain a displacement U2 versus time curve as follows:



显然，都是正值，且不会往回摆动，这与实际单摆是不一致的，这将在后面章节讲到非线性瞬态动力学的问题时修正。

Clearly, all are positive values and will not swing back, which is inconsistent with the actual simple pendulum. This will be corrected in the later chapters when discussing the issues of nonlinear transient dynamics.

1.3 视频讲解和操作验证演示 1.3 Video Explanation and Operation Verification Demonstration

如果觉得上面的文字太复杂，也可以看一下视频的简要讲解，包括基于线性瞬态动力学理论和算例的操作验证，地址如下：

If the above text is too complex, you can also watch a brief video explanation, including operation verification based on linear transient dynamics theory and examples, as follows:

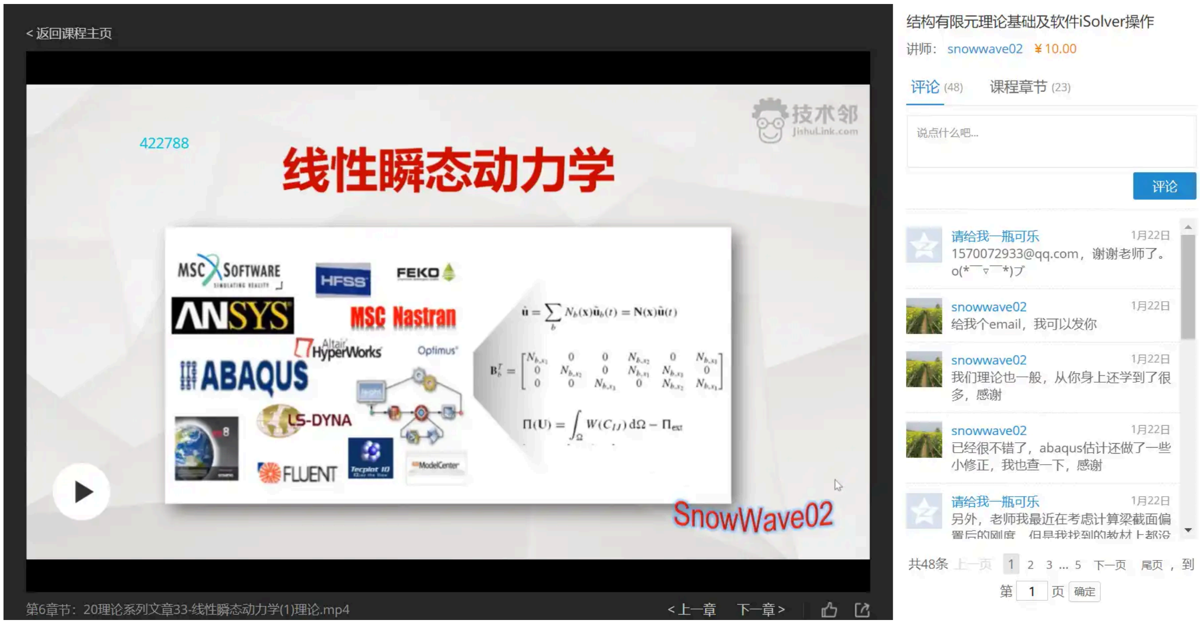
https://www.jishulink.com/college/video/c12884

20理论系列文章33-线性瞬态动力学(1)理论.mp4 及

20 Theory Series Article 33 - Linear Transient Dynamics (1) Theory.mp4 and

20理论系列文章33-线性瞬态动力学(2)算例.mp4

20 Theory Series Article 33 - Linear Transient Dynamics (2) Examples.mp4



以往的系列文章： Previous series articles:

1.6.1 ========第一阶段========

1.6.1 ========First Phase========

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。** The 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。** The Third Article: Shear Locking and Shear Band Control of S4 Shell Elements.

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

第四篇：**非线性问题的求解。** The Fourth Article: Solution of Nonlinear Problems.

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

第五篇：**单元正确性验证。** The Fifth Article: Element Accuracy 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。** Seventh Article: Stiffness Matrix of C3D8 Hexahedral Elements.

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

第八篇：**UMAT用户子程序开发步骤。** Eighth Article: Steps for Developing UMAT User Subroutines.

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

第九篇：**编写线性UMAT Step By Step。**

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.

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33：线性瞬态动力学的图64
有限元理论基础及Abaqus内部实现方式研究系列33：线性瞬态动力学的图65

1.6.2 ========第二阶段========

1.6.2 ========Second Stage========

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 independent CAE development 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 the geometric beam element.

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

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

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

第十四篇：**壳的应力方向。** The Fourteenth Article: Stress Direction of Shells.

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。**

Chapter 16: 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 17th article: The 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 18th article: 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 19th article: 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。** The 20th article: Steps for developing UEL user subroutines.

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33：线性瞬态动力学的图66
有限元理论基础及Abaqus内部实现方式研究系列33：线性瞬态动力学的图67

1.6.3 ========第三阶段========

1.6.3 ========Third Phase========

第二十一篇：**自主CAE开发实战经验第二阶段总结。** Chapter 21: Summary of the Second Stage of Autonomous CAE Development Practical 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。** The 22nd article: Solution of the 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**

Chapter 23: Writing a Simple In-Plane Tension Problem UEL Step by Step

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

第二十四篇：**显式求解Step By Step。** The 24th article: Step By Step explicit solution.

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

第二十五篇：**显式分析的稳定时间增量。** The 25th article: Stability time increment for explicit analysis.

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：Step by Step Guide to Writing Linear VUMAT.

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：Strain Calculation and Display in Abaqus Internal.

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**

Chapter 28: 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**

29th article: The conversion relationship between T.L. and U.L. in 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

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