技术邻 首页 Home 学院 College 直播 Live Streaming 问答 Q&A 悬赏 Bounty 🔥会议 🔥Conference

有限元理论基础及Abaqus内部实现方式研究系列16： Part、Instance与Assembly

Basic Theory of Finite Element Method and Research on Internal Implementation of Abaqus Series 16: Part, Instance and Assembly

SnowWave02 关注 Focus

2020年1月17日 15:22 January 17, 2020 15:22

浏览：3499 Views: 3499

评论：7 Comments: 7

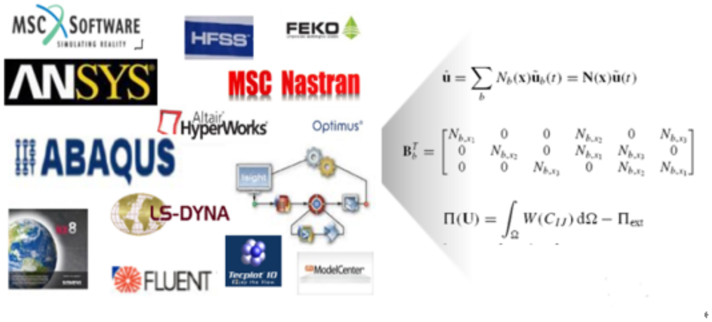
收藏：6 Favorites: 6

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

==概述== ==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。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。

This series of articles studies the mature finite element theoretical foundation and its implementation methods in commercial finite element software. The development of finite element theory 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 essentially a black box, and users can only guess its internal implementation methods from outside, except for developers.


MSC SOFTWARE
HFSS
FEKO
ANSYS
MSC Nastran
Altair HyperWorks
Optimus
ABAQUS
LS-DYNA
FLUENT
Tecplot 10
ModelCenter
$\dot{\mathbf{u}} = \sum_b N_b(\mathbf{x})\dot{\mathbf{u}}_b(t) = \mathbf{N}(\mathbf{x})\dot{\mathbf{u}}(t)$
$\Pi(\mathbf{U}) = \int_\Omega W(C_{IJ})\,d\Omega - \Pi_{ext}$

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.

iSolver介绍视频： iSolver Introduction Video:

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

==第16篇：Part、Instance与Assembly ==

==Article 16: Part, Instance and Assembly ==

有限元理论基础及Abaqus内部实现方式研究系列16： Part、Instance与Assembly的图3 1.1 三者的关系 1.1 The Relationship Among Them

与Patran不同，Abaqus引入Assembly的概念，支持在一个模型中采用多个Part。通过定义Part之间的接触和约束，能够更真实地反映复杂产品的整体结构特性。在Abaqus模型中，Assembly包含一个和多个Instance，而每个Instance都对应一个Part。

Unlike Patran, Abaqus introduces the concept of Assembly, which supports the use of multiple Parts within a single model. By defining the contact and constraints between Parts, it can more realistically reflect the overall structural characteristics of complex products. In the Abaqus model, an Assembly contains one or more Instances, and each Instance corresponds to a Part.

有限元理论基础及Abaqus内部实现方式研究系列16： Part、Instance与Assembly的图4 1.2 Instance的网格 1.2 The Mesh of Instance

在建立Instance时，除了选定对应的Part，还需要设置Instance的网格是否Dependent。如果是，则Instance直接使用Part的网格，任何对于Part网格的修改都会直接影响Instance的网格；如果否，则Instance会剪切当前Part的网格，建立仅从属于Instance的独立网格。需要注意的是，当多个Instance引用同一个Part时，这些Instance的网格必须同时独立于或者依赖Part。

When creating an Instance, in addition to selecting the corresponding Part, it is also necessary to set whether the Instance's mesh is Dependent. If it is, the Instance will directly use the mesh of the Part, and any modifications to the Part's mesh will directly affect the Instance's mesh; if not, the Instance will cut the current Part's mesh to establish an independent mesh that belongs only to the Instance. It should be noted that when multiple Instances refer to the same Part, the meshes of these Instances must either be independent of or dependent on the Part simultaneously.

Abaqus关于Instance的设计也是与现实情况相吻合的。例如，经常会有这样的情况，机械产品内部使用很多个相同零件，在仿真时就可以只建一个Part，而建立多个Instance，每个Instance的网格都依赖于Part，这样可以极大程度降低仿真的计算量。但如果这些零件的受力情况差异较大，就需要创建不同的网格，即Instance的网格独立于Part，从而反映实际的结构特性。

The design of Abaqus regarding Instance is also in line with the real-world situation. For example, it is often the case that many identical parts are used internally in mechanical products, and during simulation, only one Part needs to be built, while multiple Instances can be created. Each Instance's mesh depends on the Part, which can greatly reduce the computational load of the simulation. However, if the parts have significantly different loading conditions, different meshes need to be created, i.e., the mesh of the Instance is independent of the Part, thus reflecting the actual structural characteristics.

有限元理论基础及Abaqus内部实现方式研究系列16： Part、Instance与Assembly的图5

1.3 Abaqus内部组装流程猜测 1.3 Guessing the Assembly Process within Abaqus

从上述分析可以对Abaqus内部组装流程进行猜测，以总装刚度矩阵组装流程为例：

Based on the above analysis, a guess can be made about the internal assembly process of Abaqus, taking the assembly process of the overall stiffness matrix as an example:

1． 取一个Instance； 1. Select an Instance;

2． 判断Instance网格是否为Dependent；

2. Determine if the Instance mesh is Dependent;

3． 如果是，取Instance对应的Part；如果否，组装Instance的刚度矩阵，跳转6；

3. If so, take the Instance's corresponding Part; if not, assemble the stiffness matrix of the Instance and jump to 6;

4． 判断Part的刚度矩阵是否已经组装； 4. Determine if the Part's stiffness matrix has already been assembled;

5． 如果是，跳转6；如果否，组装Part的刚度矩阵；

5. If so, jump to 6; if not, assemble the Part's stiffness matrix;

6． 判断是否还有Instance未组装，如果是，跳转1；

6. Determine if there are any unassembled Instances; if so, jump to step 1;

7． 将所有Instance的刚度矩阵组装； 7. Assemble the stiffness matrices of all Instances;

8． 组装Assembly下的单元刚度矩阵； 8. Assemble the element stiffness matrices under the Assembly;

9． 最终形成整个Assembly的总体刚度矩阵。

9. Finally, form the overall stiffness matrix of the entire Assembly.

有限元理论基础及Abaqus内部实现方式研究系列16： Part、Instance与Assembly的图6

1.4 算例 1.4 Example

模态分析结果的正确性直接受总体刚度矩阵和总体质量矩阵影响，因此我们采用一个多Part手机模型的模态分析来验证本节内容的正确性。此次验证，依然使用自研求解器iSolver与Abaqus计算结果对比的方式。

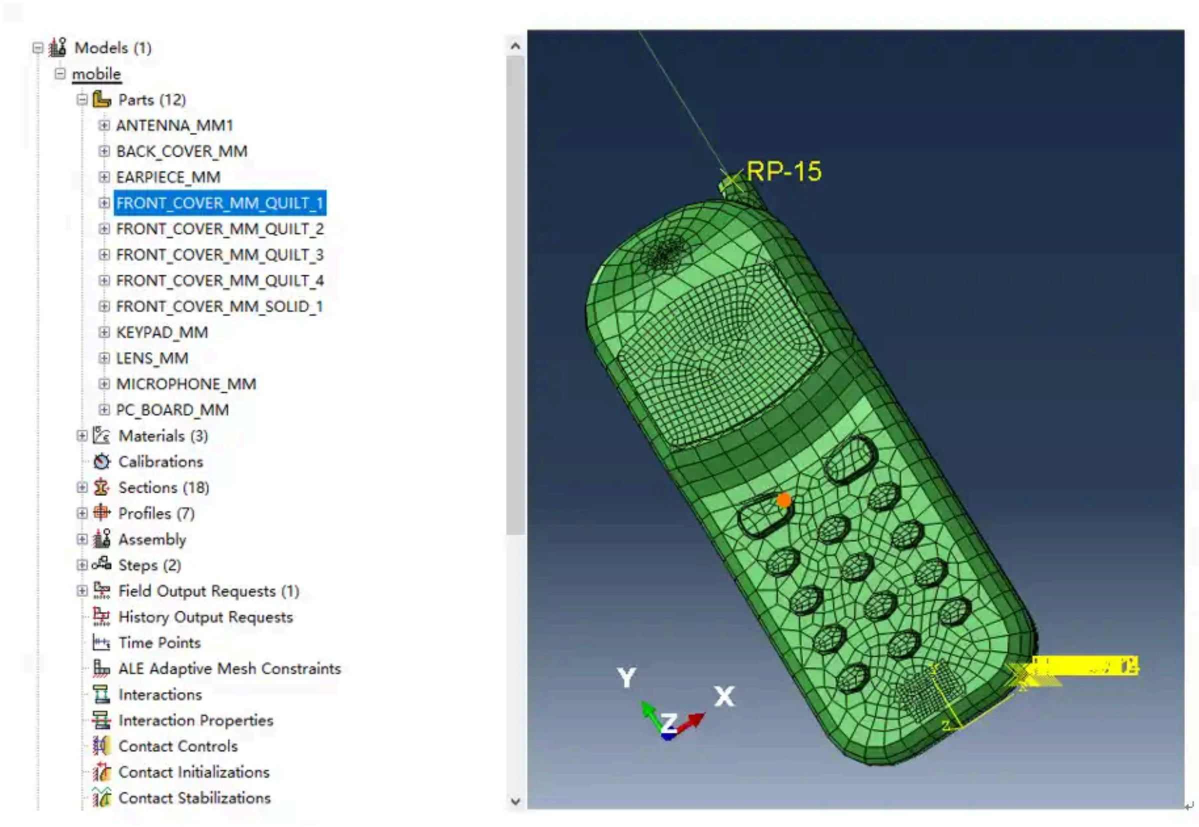
The accuracy of the modal analysis results is directly affected by the global stiffness matrix and the global mass matrix, therefore, a modal analysis of a multi-part smartphone model is used to verify the correctness of this section. For this verification, the self-developed solver iSolver is used to compare the results with those of Abaqus.

有限元理论基础及Abaqus内部实现方式研究系列16： Part、Instance与Assembly的图7

1.4.1 算例说明 1.4.1 Example Description

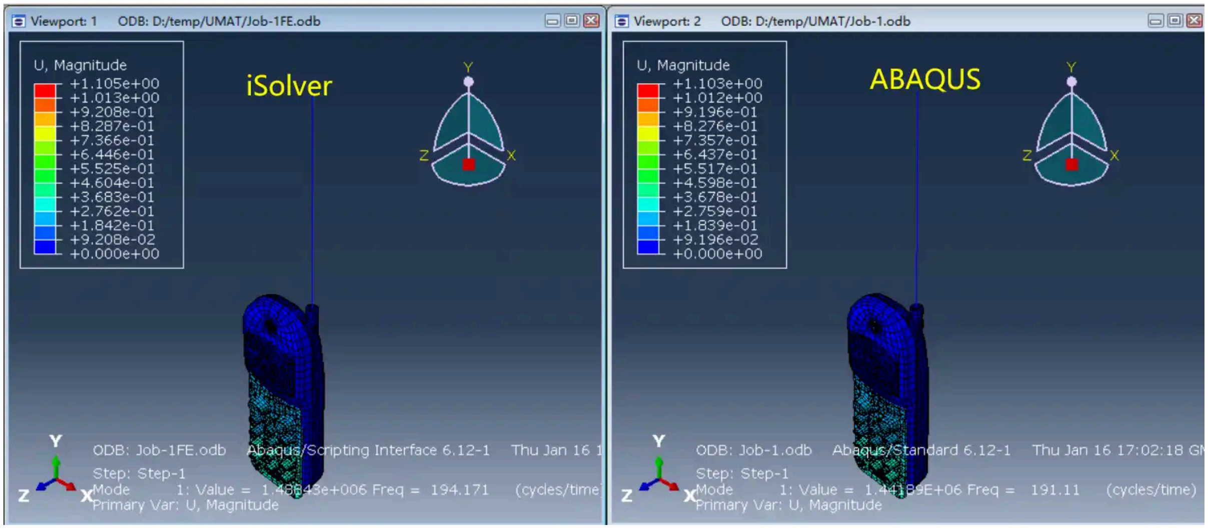
如图，此手机模型Assembly由12个Part组成，Part之间采用K-Coupling连接。

As shown in the figure, the Assembly of this smartphone model consists of 12 Parts, which are connected using K-Coupling.



有限元理论基础及Abaqus内部实现方式研究系列16： Part、Instance与Assembly的图11

1.4.2 模态分析结果 1.4.2 Modal Analysis Results



```
ODB: Job-1FE.odb    Abaqus/Scripting Interface 6.12-1    Thu Jan 16 1
Step: Step-1
Mode        1: Value =  1.48843e+006 Freq =  194.171     (cycles/time
Primary Var: U, Magnitude
```

```
ODB: Job-1.odb    Abaqus/Standard 6.12-1    Thu Jan 16 17:02:18 GI
Step: Step-1
Mode        1: Value =  1.44189E+06 Freq =  191.11     (cycles/time)
Primary Var: U, Magnitude
```

iSolver与Abaqus计算频率结果误差小于2%，由此可以证明iSolver计算得到的总体刚度矩阵和总体质量矩阵与Abaqus一致，验证了本文对于多Part模型总装矩阵组装流程的猜测。

iSolver and Abaqus calculation frequency results have an error less than 2%, which can prove that the overall stiffness matrix and overall mass matrix obtained by iSolver are consistent with Abaqus, verifying the guess of this article for the assembly process of the total assembly matrix of the multi-Part model.

有限元理论基础及Abaqus内部部实现方式研究系列16： Part、Instance与Assembly的图14

==总结== ==Summary==

本章介绍了Part、Instance与Assembly三者之间的关系，分析了Instance的网格形成原理，也简单猜测了一下Abaqus的内部组装实现流程，最后通过一个算例来验证。

This chapter introduces the relationship between Part, Instance, and Assembly, analyzes the principle of grid formation of Instance, also makes a simple guess about the internal assembly implementation process of Abaqus, and finally verifies it through a case study.

算例视频： Case Study Video:

http://www.jishulink.com/college/video/c1288411.4 分析案例篇4：手机多Part模态分析

http://www.jishulink.com/college/video/c1288411.4 Analysis Case 4: Mobile Phone Multi-Part Modal Analysis

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:

snowwave02 From www.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16： Part、Instance与Assembly的图 15

以往的系列文章： Previous series articles:

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16： Part、Instance与Assembly的图16

有限元理论基础及Abaqus内部实现方式研究系列16： Part、Instance与Assembly的图17

有限元理论基础及Abaqus内部实现方式研究系列16： Part、Instance与Assembly的图18

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.

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 geometric beams, Abaqus can generally obtain consistent analysis results. This idea is verified by a simple example.

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/1189260

推荐阅读 Recommended Reading

Abaqus、iSolver与Nastran梁单元差异...

SnowWave02 免费 Free

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

技术邻小李 Technical Neighbor Xiao Li ¥100 100 Yuan

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...

正一算法程序 Zhengyi Algorithm Program ¥220 220 Yuan

车身设计系列视频之车身钣 正向设计实例教程...

京迪轩 Jing Di Xuan ¥1