

有限元理论基础及 **Abaqus** 内部实现方式研究系列 8:

UMAT 用户子程序开发步骤

--基于 Matlab 实现 UMAT 子程序高效开发

V2019-0130



作者: SnowWave02

From www.jishulink.com

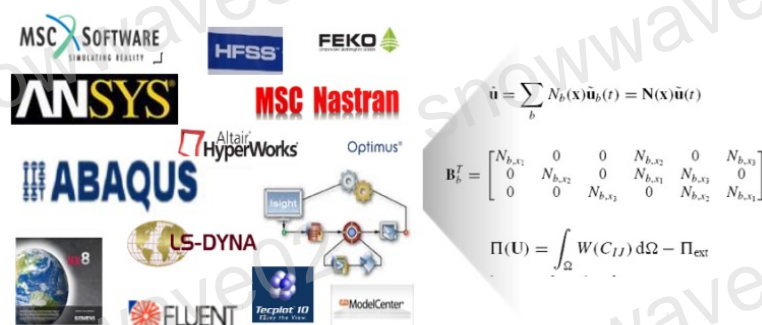
email: snowwave02@qq.com

目录

UMAT 用户子程序开发步骤.....	1
1 概述.....	3
2 第八篇：UMAT 用户子程序开发步骤.....	4
2.1 UMAT 的简单含义.....	6
2.1.1 DDSDDDE.....	6
2.1.2 STRESS.....	7
2.2 基于 Fortran 的 Abaqus 的 UMAT 的开发步骤.....	7
2.2.1 材料参数设置.....	7
2.2.2 编写.....	7
2.2.3 编译（可选）.....	8
2.2.4 调试（可选）.....	8
2.2.5 运行.....	9
2.3 基于 Matlab 的 Abaqus 的 UMAT 开发步骤.....	10
2.3.1 材料参数设置.....	10
2.3.2 编写.....	11
2.3.3 编译（无）.....	11
2.3.4 调试（可选）.....	11
2.3.5 运行.....	13
2.3.6 关联 Abaqus.....	13
2.4 算例验证.....	14
2.4.1 弯曲梁的算例.....	14
2.5 总结.....	17

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。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。



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 A 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。

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>

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。

<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。

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

本文为第八篇：**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。

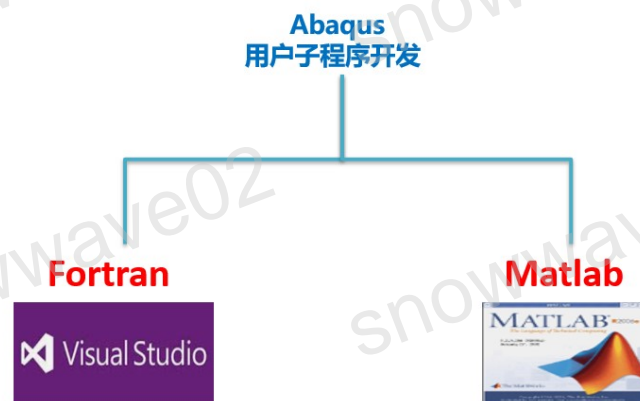
2 第八篇：UMAT 用户子程序开发步骤

用户子程序主要是将用户特定的材料本构模型和单元算法等公式编写为计算机语言表示的公式，并实现和商软求解器之间的交互迭代，UMAT 用户自定义材料是其中比较重要的一类子程序。常用的商业有限元软件都提供了用户自定义子程序的功能，且一般都是 Fortran 语言开发，Fortran 是上世纪 70 年代的语言，相对现代化的流行语言编写，格式要求非常严格，编译调试都比较繁琐，使得开发效率低下，而且接口限制较多，除了商软提供的功能外用户基本没法改动，灵活性较差。由于用户子程序很多都涉及复杂的公式编写，用户除了需要扎实的理论基础外，还需要较强的能将公式表达为 Fortran

语言的编程能力，这对非计算机专业出身的人来说往往在浪费了很多额外精力，使得很多理论高手都对用户子程序望而却步，难以入门。

商软的用户子程序的内部都是采用 **dll** 动态链接库的形式实现的，商软只是规定好接口并设置触发寻找动态链接库的机制，譬如当发现关键词***User Material**，主程序就调用指定 **dll** 中 **UMAT** 这个函数，至于这个 **dll** 是用什么语言编写的，主程序其实不会管，只要这个函数包括相同的变量名就行，主程序就会将数据传递到这些变量中，同时从规定的变量名中读取子程序的运行结果，和一般的混编语言开发类似。

如果了解了商软的用户子程序的实现原理后，完全可以用 **Matlab** 来代替 **Fortran** 开发子程序。在实际工作中，很多工程师用 **Matlab** 来编写和推导公式，**Matlab** 被认为是市面上最接近草稿纸上推导公式的一款软件了，而且有限元在数值层面上的计算其实就是矩阵运算，所以 **Matlab** 这种数据按矩阵来组织非常适合用来开发有限元相关的程序。而现在市面上还没有采用 **Matlab** 来开发商软子程序的案例并不是 **dll** 混编语言的实现方式有多难，而是需要一种基于 **Matlab** 的调试方式，在 **Matlab** 中要重复商软的有限元流程，并实现和商软的双向接口。**iSolver** 是市面上第一款基于 **Matlab** 来开发商软用户子程序的软件工具，支持用 **Matlab** 编写和调试用户子程序，并实现和 **Abaqus** 求解器的迭代调用。



本文首先简单的讨论了 **UMAT** 的一般含义，并详细的介绍了基于 **Fortran** 和 **Matlab** 两种方式的 **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。最后采用同一个算例验证两者分析结果完全一致，从而证明基于 **Matlab** 的 **UMAT** 的流程和结果的正确性。具体验证过程也可以参考下面我们的演示录像：

Abaqus 基本操作和自研有限元求解器开发框架 **iSolver** 介绍：章节 5

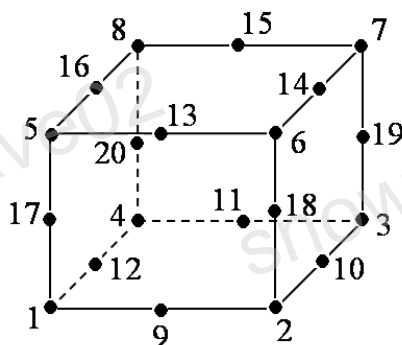
<https://www.jishulink.com/college/video/c12884?chapter=5>

2.1 UMAT 的简单含义

UMAT 网上资料很多，大家可以很容易查看，但大部分资料都只介绍 UMAT 的内部流程，我们在这儿对最重要的两个输出参数 **DDSDDE** 和 **STRESS** 在 UMAT 外面也就是 **Abaqus** 内部的用途稍微说明一下：

2.1.1 DDSDDE

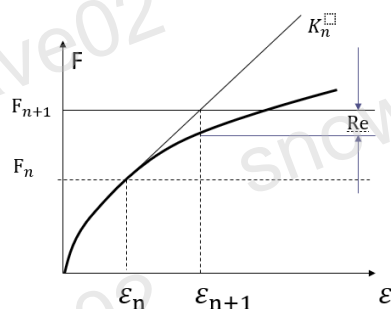
计算的 **DDSDDE** 为应力应变关系矩阵，也就是本构关系，用于计算单元刚度矩阵中，譬如线性材料的六面体的刚度为：



$$[K]^e = \iiint_{V^e} [B]^T [D] [B] dV = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 [B]^T [D] [B] |J| dg dh dr$$
$$= \sum_{i=1}^{Pi} \sum_{j=1}^{Pj} \sum_{m=1}^{Pm} \left\{ W_i W_j W_m \cdot ([B]^T [D] [B] |J|) |_{g_i, h_j, r_m} \right\}$$

其中 **D** 就是 **DDSDDE**。可见是在每个积分点上计算的。

这个计算出来的 **K** 就是 **Abaqus** 增量迭代法中当前迭代步的斜率 K_n ，在 **Newton** 迭代中，这个斜率只决定迭代的次数，也就是逼近真实值的快慢，不会影响结果，如果 **DDSDDE** 矩阵计算有稍微，结果还是收敛到精确值的（想象一下下面这个图的 **K** 有些偏差会出现什么情况？）。



2.1.2 STRESS

STRESS 用于计算等效节点力 F ，也就是

$$[F]^e = \iiint_{V^e} [B]^T S dV$$

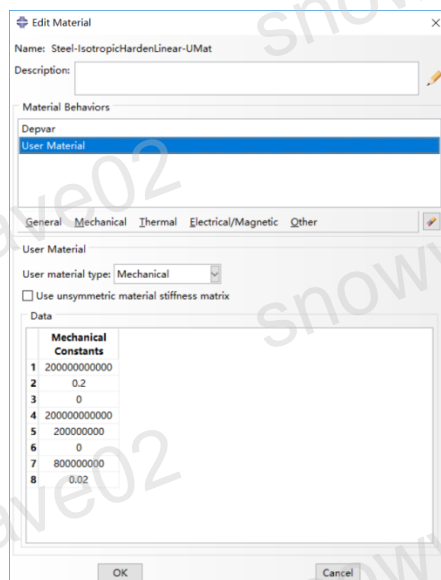
这个 S 就是 STRESS。可见也是在每个积分点上计算的。

F 最后会用来计算不平衡力 Re ，表示和精确值逼近的程序，最后收敛时 Re 接近 0，也就是 F =外力，显然这个值如果计算有问题，那直接导致结果不正确。

2.2 基于 Fortran 的 Abaqus 的 UMAT 的开发步骤

2.2.1 材料参数设置

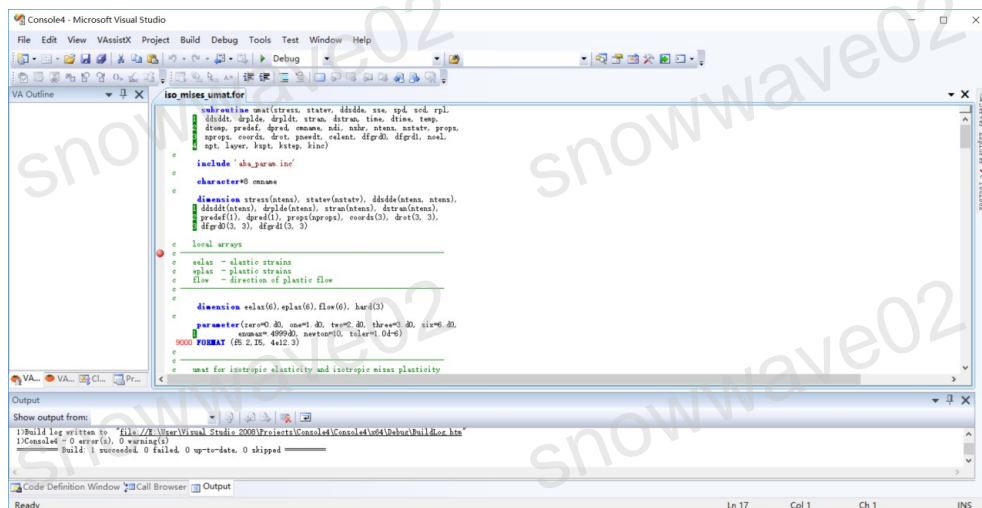
在 Abaqus 的 Property 模块，创建材料的 User Material 属性，表格中设置各种自定义参数值，如图所示：



✧ Depvar 当需要保存中间变量时才设置，非必须。

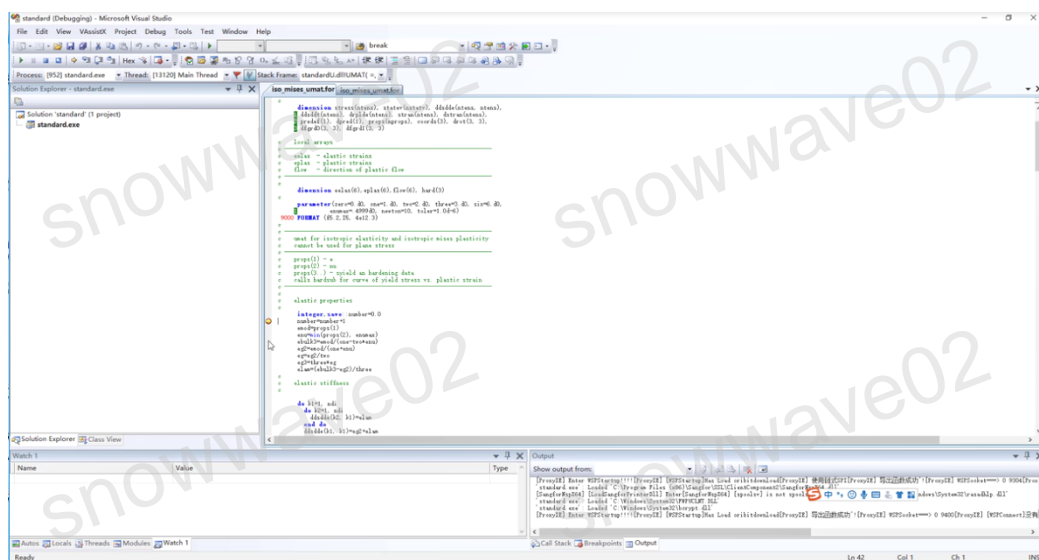
2.2.2 编写

打开任意文本编辑器编写 .for 文件，推荐使用 Visual Studio，这样可以直接进入下一步编译：



2.2.3 编译（可选）

Abaqus 没有自带 Fortran 编译器，所以用户需要自己去安装 Fortran 编译器，并用 Visual Studio 编译，在 VS 中创建一个工程，并新建或者导入老的用户自定义材料.for 文件，按 F7 编译。

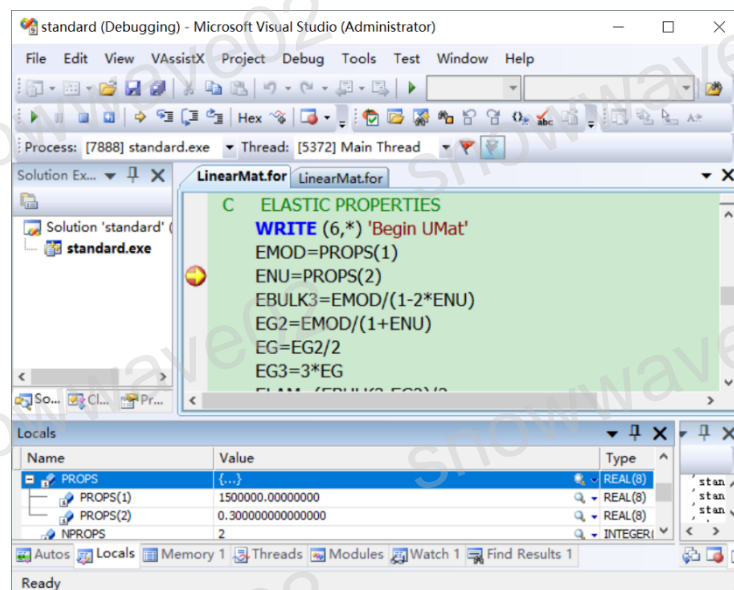


Abaqus 如果其它没问题了，那么 Visual Studio 编译结果就只剩 Cannot open include file 'aba_param.inc' 错误。这个错误不需要修改，直接进行下一步。

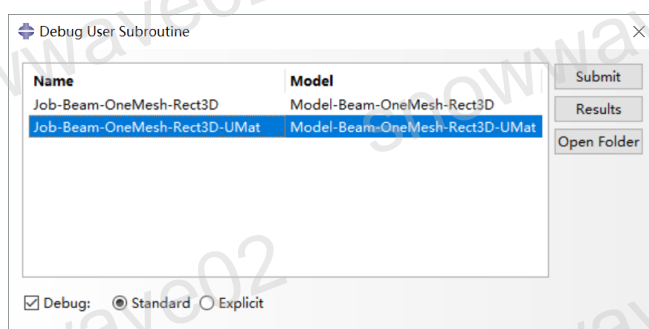
```
onfiguration: Debug Win32
Compiler: MSVC 13.0.0.0 [IA-32]
achievement\20170701 FEATheoryBasicAndAbaqus\Chapter9 Files\UMAT\UMAT\iso_mises_umat.for(77): error #5102: Cannot open include file 'aba_param.inc'
achievement\20170701 FEATheoryBasicAndAbaqus\Chapter9 Files\UMAT\UMAT\iso_mises_umat.for(220): error #5102: Cannot open include file 'aba_param.inc'
achievement\20170701 FEATheoryBasicAndAbaqus\Chapter9 Files\UMAT\UMAT\iso_mises_umat.for(220): error #5102: Cannot open include file 'aba_param.inc'
OR\FE\FEEngine\1 Document\5 Achievement\20170701 FEATheoryBasicAndAbaqus\Chapter9 Files\UMAT\UMAT\Debug\BuildLog.htm
0 up-to-date, 0 skipped
```

2.2.4 调试（可选）

如果想要知道代码的运行结果是否和预期的一致，一种笨办法是用 print 打印到 log 文件中，高效的方法是采用断点调试的方法进行运行中的调试。



进行调试需要先在 Abaqus 的启动 bat 配置 Visual Studio 和 fortran 编译器。同时在命令行调试,不过命令行反复运行也比较繁琐,用户也可选择更加方便的调试 Abaqus 的用户子程序的 DUS 插件工具, DUS (Debug User Subroutine) 是集成在 ABAQUS/CAE 中的一个插件,能够一键启动用户配置的用户子程序开发平台(如 Visual Studio 2008 等), 并进入对用户定义子程序的单步调试模式。



有兴趣的可到下面网页下载使用。

Abaqus 用户子程序调试插件:

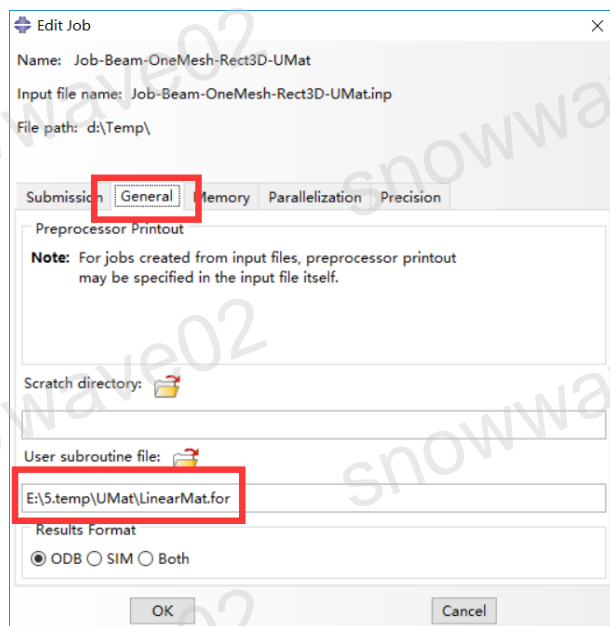
<https://www.jishulink.com/content/post/424513>

环境配置和调试插件 DUS 使用可看录像:

<https://www.jishulink.com/college/video/c13034?chapter=1>

2.2.5 运行

该步和调试一样,也需要先确认 Fortran 环境配置正确,然后在创建 Job 的时候,在 General 的 User subroutine file 指定前面编写的.for 文件。这时就可以像正常的 Job 一样 submit 了。如果运行出错,先查看.log 文件是否有 error,排除是.for 文件的错误。



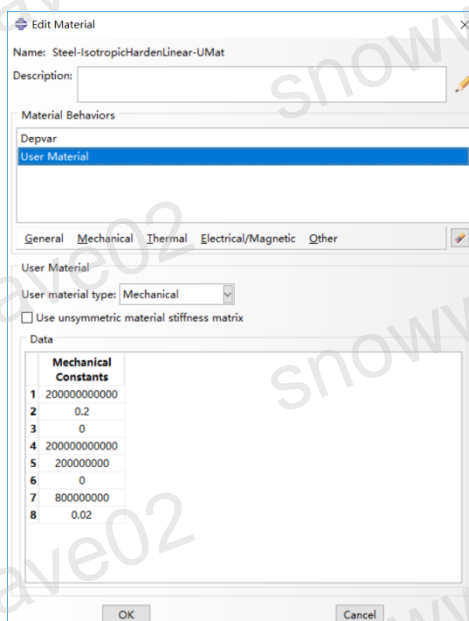
至此，基于 Fortran 的 UMAT 开发流程已经完成，但结果的正确性还需要更加细致的验证，为更方便的查找问题，建议先采用单个单元调试 UMAT，在确保单个单元正确后再将 UMAT 用于实际问题。

2.3 基于 Matlab 的 Abaqus 的 UMAT 开发步骤

基于 Matlab 的 Abaqus 的 UMAT 具体开发步骤和 Abaqus 类似，只不过某些步骤需要用到自研工具 iSolver。

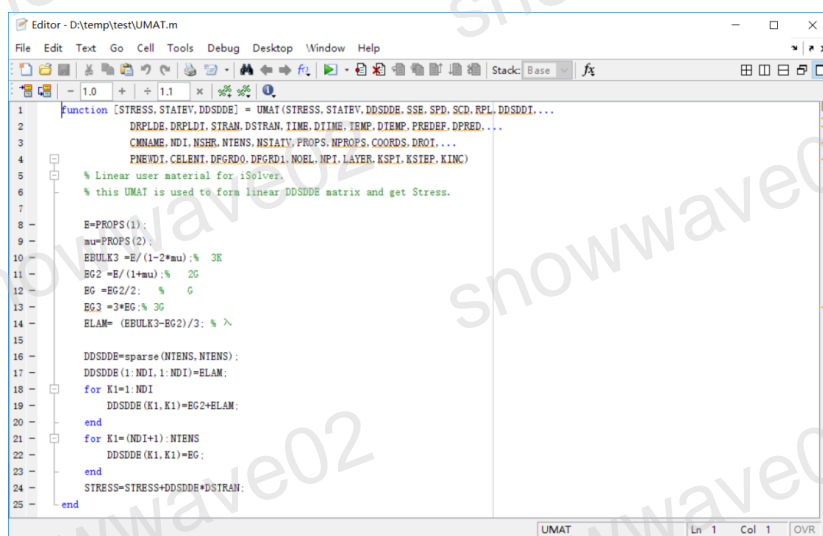
2.3.1 材料参数设置

Abaqus/CAE 的材料参数设置完全一致，如图所示：



2.3.2 编写

在 Matlab 中创建并编写 UMAT.m 的文件，放入 Abaqus 工作目录下。该文件只包括一个 UMAT 函数，接口和 Abaqus 的接口参数完全一致，功能也是计算应力应变关系和当前应力状态等，相对 Fortran，利用 Matlab 可以更容易的编写计算公式，同时可以利用 Matlab 在矩阵计算中各种强大功能和算法库。因为 Abaqus 的 UMAT 接口和计算功能各个版本相对固定，这个 matlab 的 UMAT.m 也相对固定，不会因为 iSolver 的版本不同而重新修改 UMAT.m。

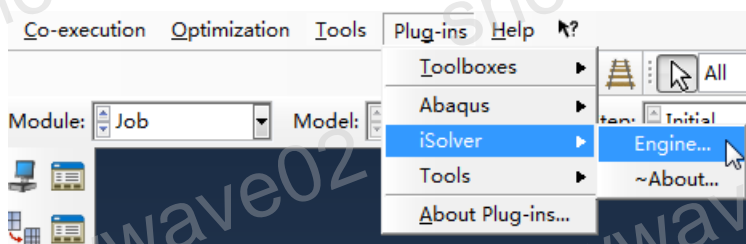


2.3.3 编译（无）

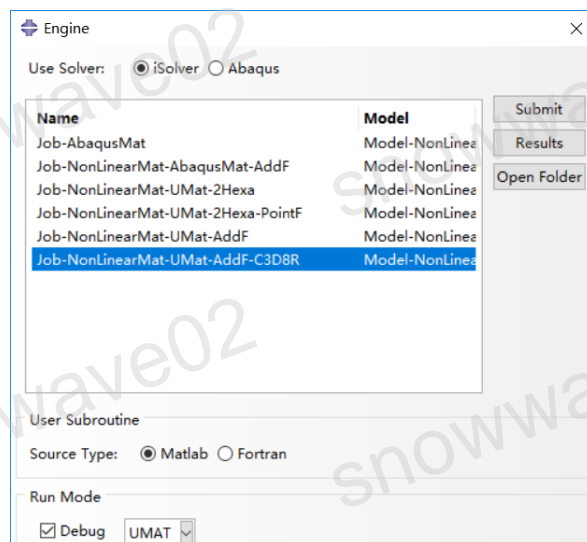
由于 matlab 是脚本语言，不需要编译。

2.3.4 调试（可选）

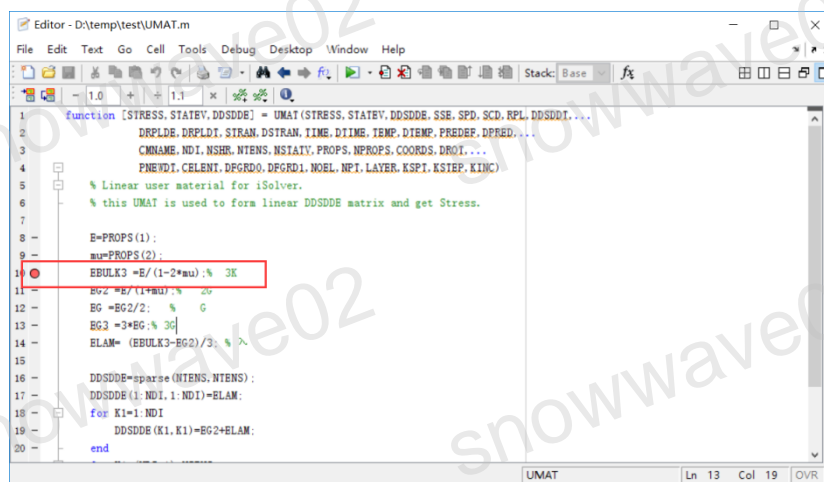
在 Abaqus 菜单栏的 Plug-ins 里选择 iSolver 插件的菜单。



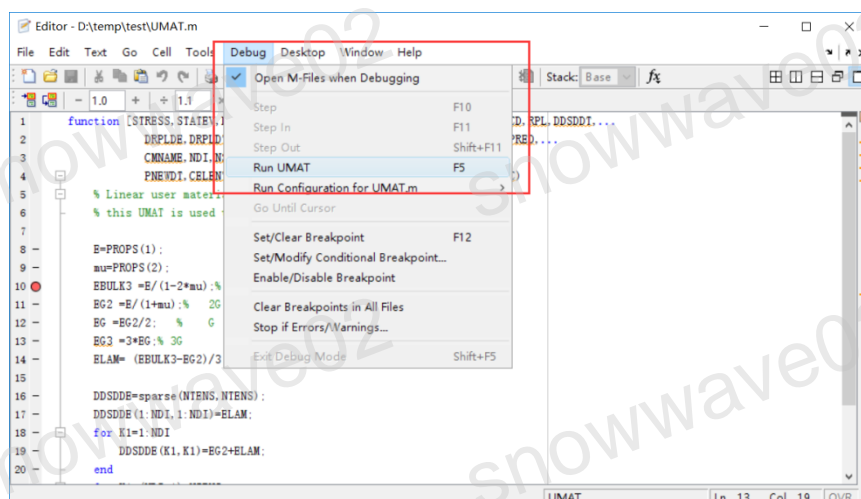
点击 iSolver->Engine，按照下图所示，在功能项 Use Solver 中选择 iSolver，在 Source Type 里面选择 Matlab，勾选 Debug。点击 Submit 进行调试运行。



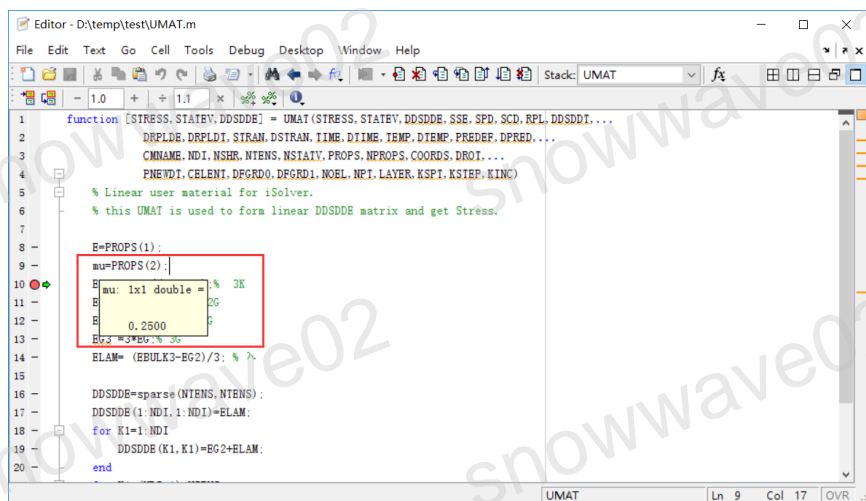
程序会自动打开 matlab 并加载 UMat.m 文件，手动打上断点



点击在 Debug 菜单下的 Run UMAT 运行 UMat 程序。



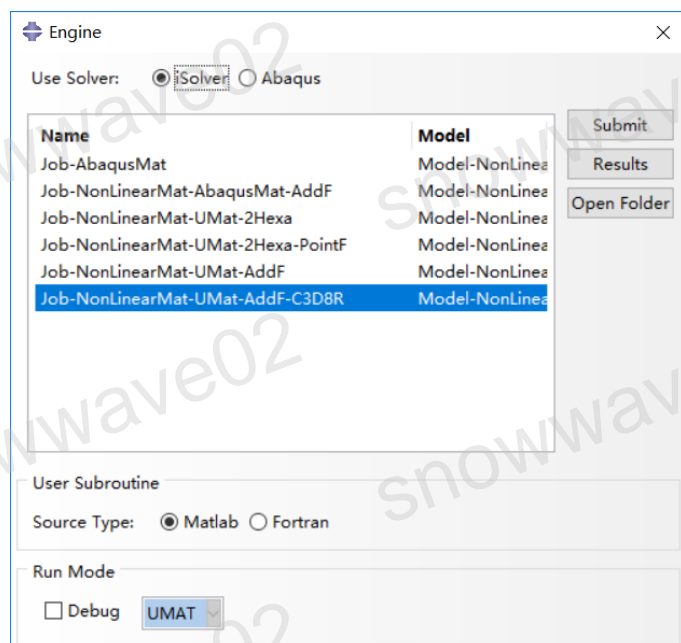
程序将在断点处停止，且将鼠标移动到需要调试查看的参数上，能够查看到对应的值。



按 F10 可以进行单步调试。

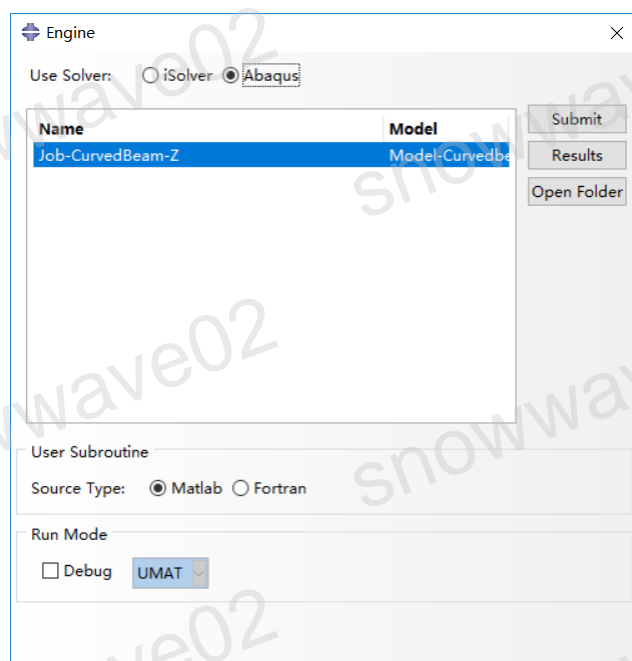
2.3.5 运行

在上述步骤的基础上去掉勾选 **Debug** 选项，点击 **Submit** 运行计算，此时将采用 iSolver 求解器联合 UMAT.m 进行求解分析，运行完毕点击 **Result** 在 Abaqus 中查看结果。



2.3.6 关联 Abaqus

上述的运行只是在 iSolver 求解器中运行，我们最终还需要在 Abaqus 求解器中使用 UMAT.m。在功能项 **Use Solver** 中选择 **Abaqus**，在 **Source Type** 里面选择 **Matlab**。点击 **Submit** 运行自动启动 Abaqus，并自动关联 Abaqus 完成迭代计算。



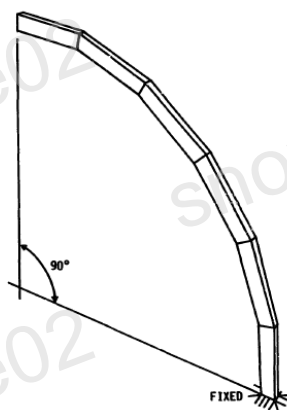
2.4 算例验证

下面我们采用 **Matlab** 编写的自定义材料，和 **Abaqus** 自带材料结果比较从而验证上述基于 **Matlab** 的 **Abaqus** 的 **UMAT** 开发步骤的正确性。

2.4.1 弯曲梁的算例

2.4.1.1 模型描述

模型：该算例如下。

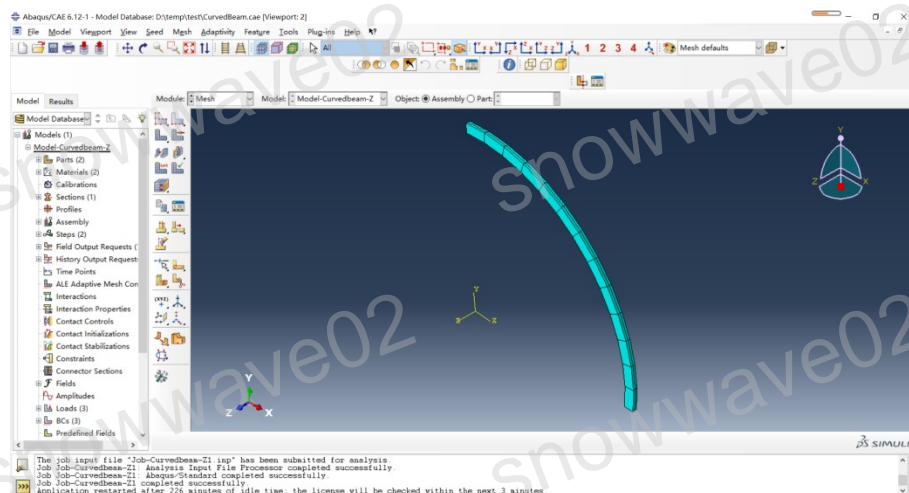


模型为一个 1/4 的半圆形的梁，一端固支，另一端受面外拉力 1N，分析受力点在面外的位移。

几何：内径 4.12，外径 4.32，厚度 $h=0.1$ 。

材料：杨氏模量 $1e7$ ，泊松比 0.25。

网格：网格取为 C3D8，划了一个简单的网格。

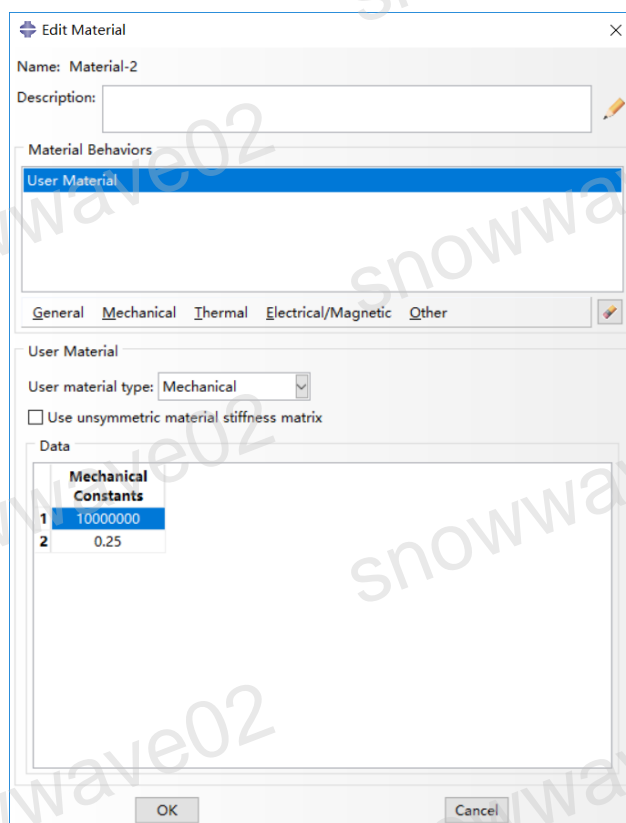


边界和载荷：一端固支，另一端受面外拉力 1N。

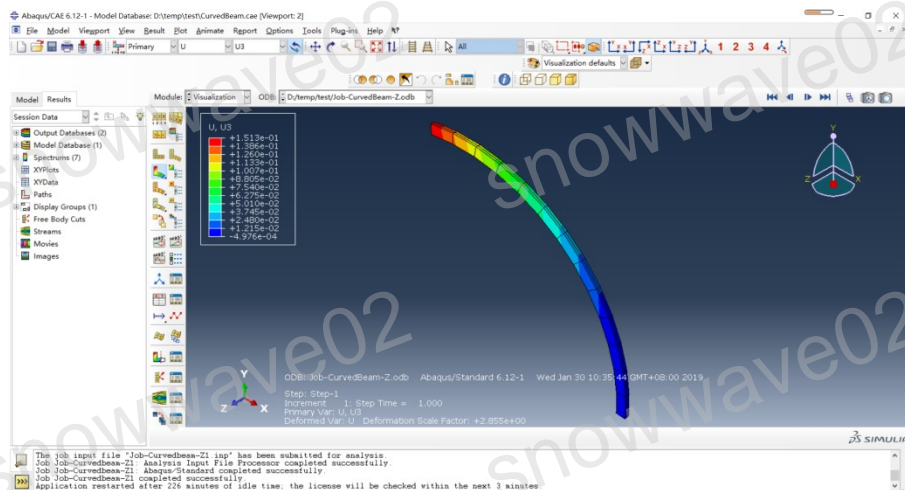
2.4.1.2 分析结果

2.4.1.2.1 Abaqus 采用 Matlab 的 UMAT 的结果

用户自定义材料设置：



采用 UMAT.m，计算结果得到 U3 最大值=0.1513：



2.4.1.2.2 Abaqus 采用自带材料结果

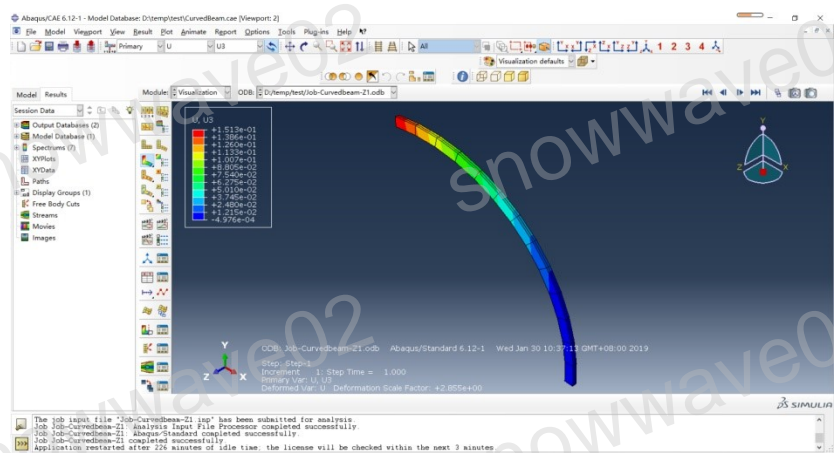
Abaqus 采用自带材料，Elastic 属性中设置：

The 'Edit Material' dialog box is shown. The 'Name' is 'Material-3' and the 'Description' is empty. Under 'Material Behaviors', 'Elastic' is selected. The 'General' tab is active. The 'Type' is 'Isotropic'. The 'Use temperature-dependent data' checkbox is unchecked. The 'Number of field variables' is 0. The 'Moduli time scale (for viscoelasticity)' is 'Long-term'. The 'No compression' and 'No tension' checkboxes are unchecked. The 'Data' table shows the following values:

	Young's Modulus	Poisson's Ratio
1	10000000	0.25

The 'OK' and 'Cancel' buttons are at the bottom.

Abaqus 分析后得到 U3 最大值=0.1513:



2.4.1.2.3 结果分析

Abaqus 采用 MATLAB 的 UMAT 材料和自带材料结果完全一致,基于 MATAB 开发的 UMAT 除了实现了材料内部的算法公式外,在外部还和 Abaqus 求解器进行了双向通讯,并且确保了迭代过程和 Abaqus 内部计算完全一致。

2.5 总结

本文首先简单的讨论了 UMAT 的一般含义,并详细的介绍了基于 Fortran 和 Matlab 两种方式的 UMAT 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。最后采用同一个算例验证两者分析结果完全一致,从而证明基于 Matlab 的 UMAT 的流程和结果的正确性。

UMAT 的开发一方面要有扎实的公式推导能力,另一方面需要基础的编程能力和开发工具应用水平,后者不是重点,但往往浪费了大家很多的精力,善用工具方能提高效率,基于 Fortran 和 Matlab 两种方式的 UMAT 的开发步骤和开发工具如下表:

项次	步骤	基于 Fortran 的 开发工具	基于 Matlab 的 开发工具
1	材料参数设置	Abaqus/CAE	Abaqus/CAE
2	编写	文本编译器	Matlab
3	编译	VS+ivf	无需编译
4	调试	VS+ivf+DUS 插件	Matlab
5	运行	Abaqus+UMAT.for	iSolver+UMAT.m
6	关联 Abaqus	/	Abaqus+UMAT.m

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

email: snowwave02@qq.com