

# OpenRadioss 研发报告

赵吉辰

版本：1.00

更新：2023 年 8 月 25 日

# 目录

<b>1</b>	<b>项目背景</b>	<b>4</b>
<b>2</b>	<b>OpenRadioss 编译</b>	<b>4</b>
2.1	Linux 下 OpenRadioss 的编译 . . . . .	4
2.1.1	环境 . . . . .	4
2.1.2	编译指令 . . . . .	5
<b>3</b>	<b>OpenRadioss 使用</b>	<b>9</b>
3.1	前处理 . . . . .	9
3.2	计算 . . . . .	10
3.3	后处理 . . . . .	11
3.4	使用 LSDYNA 格式输入 . . . . .	12
3.5	总结 . . . . .	15
3.6	To Do . . . . .	15

<b>4</b>	<b>OpenRadioss 材料</b>	<b>15</b>
4.1	READ_MATERIAL_MODELS.F . . . . .	16
4.2	HM_READ_MAT.F . . . . .	17
4.3	Radioss 中材料本构 . . . . .	18
4.3.1	各向同性弹性 . . . . .	18
4.3.2	爆炸 . . . . .	19
4.3.3	失效模型 . . . . .	19
4.4	实例分析——各向同性线弹性材料 . . . . .	19
4.4.1	弹塑性应力——应变曲线 . . . . .	20
4.4.2	MAT1 定义方式 . . . . .	22
4.4.3	原理 . . . . .	23
4.4.4	MAT1 代码解读 . . . . .	24
<b>5</b>	<b>FEniCS</b>	<b>28</b>

# 1 项目背景

OpenRadioss 是公开可用的开源代码库。OpenRadioss 可以使开发人员应对快速发展的技术带来的最新挑战，如电池开发、轻质材料和复合材料、人体模型和生物材料、自动驾驶和飞行，以及通过虚拟测试为乘客提供最安全环境的愿望。用户可以模拟汽车碰撞和安全、冲击和冲击分析、电子和消费品跌落测试以及流体结构相互作用。OpenRadioss 使用户能够依靠先进的 MPI 和 OpenMP 并行结构对复杂环境中的组合多物理场行为进行高效、稳健的预测。

- 官网: <https://www.openradioss.org/>
- 官方 Wiki: <https://openradioss.atlassian.net/wiki/spaces/OPENRADIOSS/overview>
- 在线文档: <https://github.com/jczhao1022/OpenRadiossUsage>

## 2 OpenRadioss 编译

### 2.1 Linux 下 OpenRadioss 的编译

#### 2.1.1 环境

1. Ubuntu 20.0.4 or higher

```
apt-get update  
apt-get upgrade
```

```
apt-get install build-essential
apt-get install gfortran
apt-get install cmake
apt-get install perl
apt-get install python3
apt-get install python-is-python3
apt-get install git-lfs
apt-get install openmpi-bin openmpi-doc libopenmpi-dev
apt-get install paraview
```

推荐版本为 gcc-11, g++-11, gfortran-11。paraview 用于后期可视化。

## 2.1.2 编译指令

### 1. 获取源码

(a). LFS: ‘git lfs install‘

(b). 运行 (这里需要先绑定 SSH key) ‘git clone git@github.com:OpenRadioss/OpenRadioss.git‘.

### 2. OpenRadioss Starter

(a). 进入 OpenRadioss/starter 目录

```
cd OpenRadioss/starter
```

(b). 运行 ‘build\_script.sh‘

```
./build_script.sh -arch=linux64_gf
```

‘./build\_script.sh’ 参数:

```
[]$ ./build_script.sh
```

build\_script

-----

Use with arguments :

-arch=[build architecture]

    -arch=linux64\_gf (SMP executable / Gfortran compiler)

-prec=[dp|sp] : set precision - dp (default) |sp

-static-link : Fortran, C & C++ runtime are linked in  
    binary

-debug=[0|1] : debug **version** 0 no debug flags (default),  
    1 usual debug flag

-addflag="list of additionnal flags" : add compiler flags to usual set

Execution control

-nt=[threads] : number of threads for build

-verbose : Verbose build

-clean : clean build directory

### 3. OpenRadioss Engine

(a). 进入 OpenRadioss/engine 目录

(b). 运行 ‘build\_script.sh’

```
./build_script.sh -arch=linux64_gf -mpi=mpi
```

(c). './build\_script.sh' 参数:

```
[]$ ./build_script.sh
```

```
build_script
```

```
-----
```

```
Use with arguments :
```

```
-arch=[build architecture]
```

```
-arch=linux64_gf                (SMP executable / Gfortran compiler)
```

```
-arch=linux64_gf -mpi=mpi       (OpenMPI executable / Gfortran compiler)
```

```
MPI libraries
```

```
-mpi=[mpi]
```

```
not set      : SMP (default)
```

```
-mpi=mpi     : OpenMPI
```

```
Controlling MPI Libraries - if need choose one of the 3 Option Set
```

```
If no options set, recommended OpenMPI
```

```
directories are uses (default)
```

```
1. -mpi-os                                     : link with default MPI version
```

```
installed on system
```

```
libraries are in default installation
```

2. `-mpi-root=[directory]` : set rootname to link with specific MPI installation
3. `-mpi-include=[directory]` : set include directory where to find `mpif.h` and `mpi.h`
- `-mpi-libdir=[directory]` : set library directory where to find mpi libraries

#### Other control

- `-prec=[dp|sp]` : set precision - dp (default) | sp
- `-static-link` : Fortran, C & C++ runtime are linked in binary
- `-debug=[0|1]` : debug version 0 no debug flags (default), 1 usual debug flag
- `-addflag="list of additionnal flags"` : add compiler flags to usual set

#### Build control

- `-nt=[threads]` : number of threads for build
- `-verbose` : Verbose build
- `-clean` : clean build directory
- MUMPS linear solver: available only for dp, with mpi"
- `-mumps_root=[path_to_mumps]` : path\_to\_mumps/lib/libdmumps.a must exist
- `-scalapack_root=[path to scalapack]` : path\_to\_scalapack/libscalapack.a must exist
- `-lapack_root=[path to lapack]` : path\_to\_lapack/liblapack.a must exist



#### 4. 自动化脚本（待定）

```
cd starter
./build_script.sh -arch=linux64_gf
cd ../engine
./build_script.sh -arch=linux64_gf -mpi=mpi
```

## 3 OpenRadioss 使用

### 3.1 前处理

- gmsh
  - 只能生成网格信息
- Hypermesh
  - 付费软件
  - 功能最全
- ls-prepost
  - 只能生成.k 文件，OpenRadioss 不能完全支持其格式
  - 完成使用 ls-prepost 建立小球击穿铝板的前处理，并可以使用 OpenRadioss 仿真

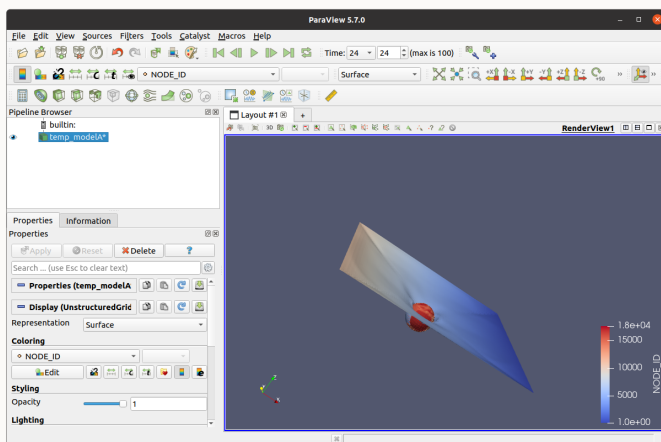


图 1: 使用 ls-prepost 建立小球击穿铝板模型前处理

## 3.2 计算

OpenRadioss 的仿真流程分为两步，其可以用下图表示。xxxx\_0000.rad 文件中记录了模型的材料，接触，状态，网格等信息。starter 接受输入之后，会生成日志，处理器计算控制文件，可以控制并行计算的流程。xxxx\_0001.rad 文件中记录了一些控制参数，比如如何生成三维动画，三维对象，历史数据等。

官网给出了一些标准的建模例子，比如鸟撞飞机，汽车撞击等。在官网例子中可能需要对其 xxxx\_0001.rad 文件进行处理，以获得合适的可视化建模。

一个使用标准输入的脚本定义如下，其中文件命名为 test....

```
export OMP_STACKSIZE=400m
export OMP_NUM_THREADS=8
# 调用 starter 处理 test_0000.rad
./starter_linux64_gf -i ../source/test_0000.rad -np 1 -outfile="../output"
# 调用 engine 处理 test_0001.rad
mpiexec -n 1 --map-by socket:PE=$OMP_NUM_THREADS --bind-to core ./engine_linux64_gf_
    ompi -i ../source/test_0001.rad -outfile="../output"
```

### 3.3 后处理

使用 paraview 后处理的脚本如下

```
# 使用小工具将 anim 文件转换为 .vtk 文件
folder_path=../output

for file in "$folder_path"/*A[0-9][0-9][0-9]
do
    input_file="$folder_path/${file}"
    output_file="$folder_path/${file}.vtk"
    ./anim_to_vtk_linux64_gf "$input_file" > "$output_file"
done
# 调用 paraview 后处理
paraview "testA..vtk"
```

## 3.4 使用 LSDYNA 格式输入

### 1. 测试实例

测试例子是[拉伸-LSDYNA](#)。

### 2. bash 脚本

```
myjobname=$(echo "$1" )
echo $myjobname "running in OpenRadioss"
export OMP_STACKSIZE=400m
export OMP_NUM_THREADS=8
./starter_linux64_gf -i "../source/"$myjobname".k" -np $2 -outfile="../output"
#./starter_linux64_gf -i $myjobname".k" -np $2 -nt 4
# 这里是找到source文件夹下的，LS-DYNA 格式文件 (.k) 并将其传递给starter 进行计算。
mpiexec -n $2 --map-by socket:PE=$OMP_NUM_THREADS --bind-to core ./engine_linux64_
gf_ompi -i "../bin/"$myjobname"_0001.rad" -outfile="../output"
#mpiexec -n $2 .engine_linux64_gf_ompi -i $myjobname"_0001.rad" -nt 1
# 在处理.k文件时，starter 会在其目录下生成"_0001.rad"文件，需要将其传递给 engine

BASEDIR=$(dirname "$0")
for file in ../output/*;
do
    if [ -n "${file: -3}" ] && [ "${file: -3}" -eq "${file: -3}" ] 2>/dev/null && [[
        "${file: -4:1}" == "A" ]]; then
        ./anim_to_vtk_linux64_gf $file > "$file.vtk"
```

```
        echo "$file is converted"
        # anim 文件转化为 vtk 文件
    elif [[ "${file: -3}" == "T01" ]] ;then
        ./th_to_csv_linux64_gf $file
        CSV="$file.csv"
        # T01 文件转化为 csv 文件
    fi
done
cd ../output/
#paraview $myjobname"A..vtk" $CSV
paraview $myjobname"A..vtk" $CSV
exit
```

在命令行输入 ‘bash process\_k.sh zug\_test3\_RS 1 ‘即可。

### 3. 运行结果见图2

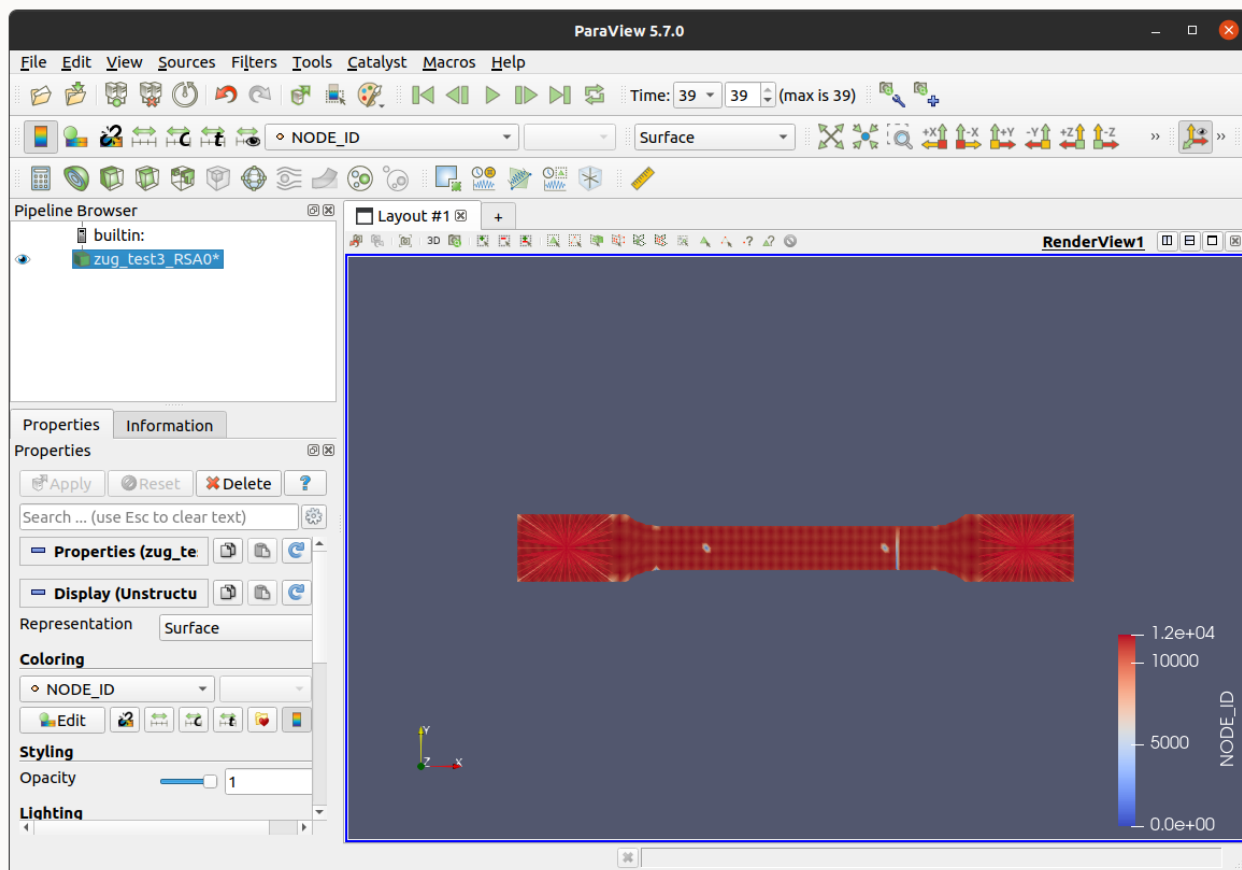


图 2: 弹性 LS-DYNA 运行结果

## 3.5 总结

在使用 Openadioss 处理 LS-DYNA 格式的文件时，需要.k 文件。这个文件其实是可以直接放在 LS-DYNA 上进行计算的。这里提供了一种其他的计算方式——使用 OpenRadioss。

## 3.6 To Do

需要验证计算的正确性：也就是使用 LS-DYNA 计算同样的例子，观察计算结果是否有出入。

- LS-DYNA 是商业软件，能找到的软件不支持 Ubuntu 系统。
  - 可以 windows 下仿真，将结果和进行对比。
- 如何衡量误差
  - 这一部分有很多需要考虑的，比如毁伤面积、毁伤深度。

## 4 OpenRadioss 材料

在 OpenRadioss 中，材料信息被记录在 xxxx\_0000.rad 文件中，包括以下内容：

材料本构定律 (Constitutive Laws) : /MAT

状态方程 (Equations of State) : /EOS

失效模型 (Failure models) : /FAIL

粘性模型 (Viscosity) : /VISC

热参数 (Thermal parameters) : /HEAT

## 4.1 READ\_MATERIAL\_MODELS.F

源码中使用 Fortran 函数 READ\_MATERIAL\_MODELS.F 来读取。这段代码定义了一个 Fortran 的子程序 READ\_MATERIAL\_MODELS，其主要功能是管理与材料模型相关的读取器，其输入包括：

MAT\_ELEM: 材料元素的参数  
MLAW\_TAG: 材料本构定律的标签  
FAIL\_TAG: 失效模型的标签  
VISC\_TAG: 粘性模型的标签  
EOS\_TAG: 状态方程的标签  
BUFMAT: 材料数据的缓存区  
BUFLEN: 缓存区的长度  
IPM: 材料参数的整型数组  
PM: 材料参数的实型数组  
UNITAB: 单位制表  
MULTI\_FVM: 多重有限体积法的数据结构  
MAXEOS: 最大状态方程数  
FAILWAVE: 失效波的数据结构  
NLOC\_DMG: 非局部损伤的数据结构  
LSUBMODEL: 子模型的数据结构  
TABLE: 表的数据结构  
NPC: 数组 NPC



该子程序的输出包括：

MAT\_ELEM：材料元素的参数

MLAW\_TAG：材料本构定律的标签

FAIL\_TAG：失效模型的标签

VISC\_TAG：粘性模型的标签

EOS\_TAG：状态方程的标签

该子程序还调用了其他一些子程序，例如 HM\_READ\_MAT、HM\_READ\_EOS、HM\_READ\_FAIL、HM\_READ\_VISC、HM\_READ\_LEAK、READ\_ALE\_MAT、READ\_EULER\_MAT、FILL\_BUFFER\_51\_0、HM\_READ\_THERM、HM\_READ\_THERM\_STRESS 和 HM\_READ\_NONLOCAL 等，用于读取不同类型的材料模型数据。

接下来考虑分析 ‘HM\_READ\_MAT’ 程序，分析软件是如何运作的。

## 4.2 HM\_READ\_MAT.F

这是用于读取材料模型定义的函数。

- 对于每种材料，它根据材料定律关键字（MAT00、MAT01 等）调用特定的读取例程。这些读取例程位于 mat 子目录中的单独文件中。
- 解析输入、填充材料参数数组、设置材料所需数据的标志，并存储材料 ID、定律类型、应变率设置等内容。
- 读取材料后，它会进行一些后处理，例如计算派生值（剪切模量）、检查重复 ID 以及计算输出所

需数量的平方根。

- 每种材料的属性存储在 MAT\_PARAM 数组和各种其他公共块中。

它循环遍历材料，调用特定于定律的读取器，并将所有材料数据收集到参数数组和通用块中，以便稍后在 FEA 分析中使用。关键部分是模块化法则读取器和用于保存材料道具的中央 MAT\_PARAM 数据结构。

因此，想要定义全新的材料，所要做的事情是自定义 MATxxx 文件。

## 4.3 Radioss 中材料本构

### 4.3.1 各向同性弹性

Description	Model Name	Keyword/MAT	Law
空材料	Void	/VOID	0
线弹性	Elastic (Hooke)	/ELAST	1
超弹性	Tabulated Hyperelastic	n/a	69
超弹性	Ogden Formulation	n/a	82

表 1: 部分各向同性弹性材料

### 4.3.2 爆炸

Description	Model Name	Keyword/MAT	Law
Detonation driven	Jones Wilkins Lee	/JEL	5
Hydrodynamics	Lee-Tarver	/LEE-TARVER	41
Multi-materials	Soild, liquid,gas and explosives	/MULTIMAT	51

表 2: 部分适用爆炸的材料

### 4.3.3 失效模型

/FAIL/Tuler-Butcher 高速碰撞失效、

## 4.4 实例分析——各向同性线弹性材料

/MAT/LAW1 或/MAT/ELAST: 弹性材料

1. LAW1 使用胡克定律对各向同性，线弹性材料进行建模
2. 可用于杆，梁，壳以及实体 solids
3. 适用于连接刚体
4. 不适用形变大的构件

#### 4.4.1 弹塑性应力——应变曲线

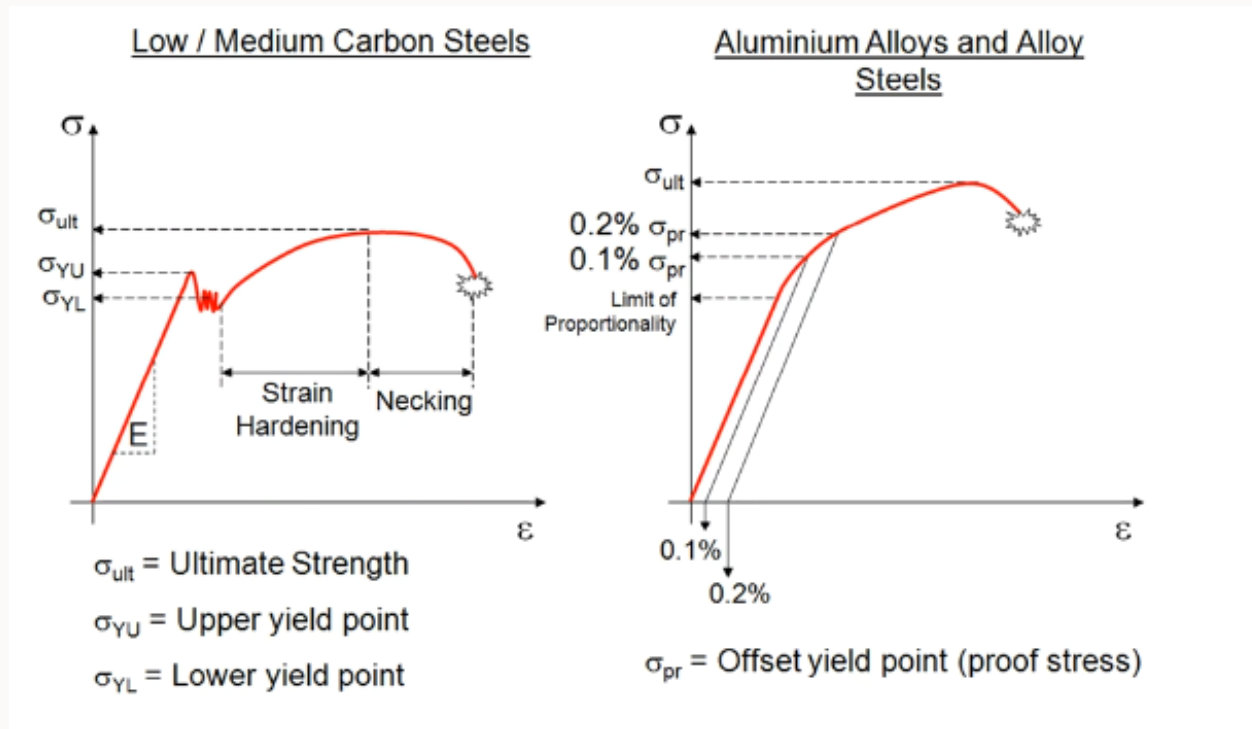


图 3: 弹塑性应力——应变曲线

真实应力应变

- 真实应力：考虑当前或真实横截面积  $A_{true}$  的应力

$$\sigma_{true} = \frac{P_x}{A_{true}}$$

- 在塑性变形范围内，塑性变形横截面积永久减小
- 继续使用工程应力不再准确

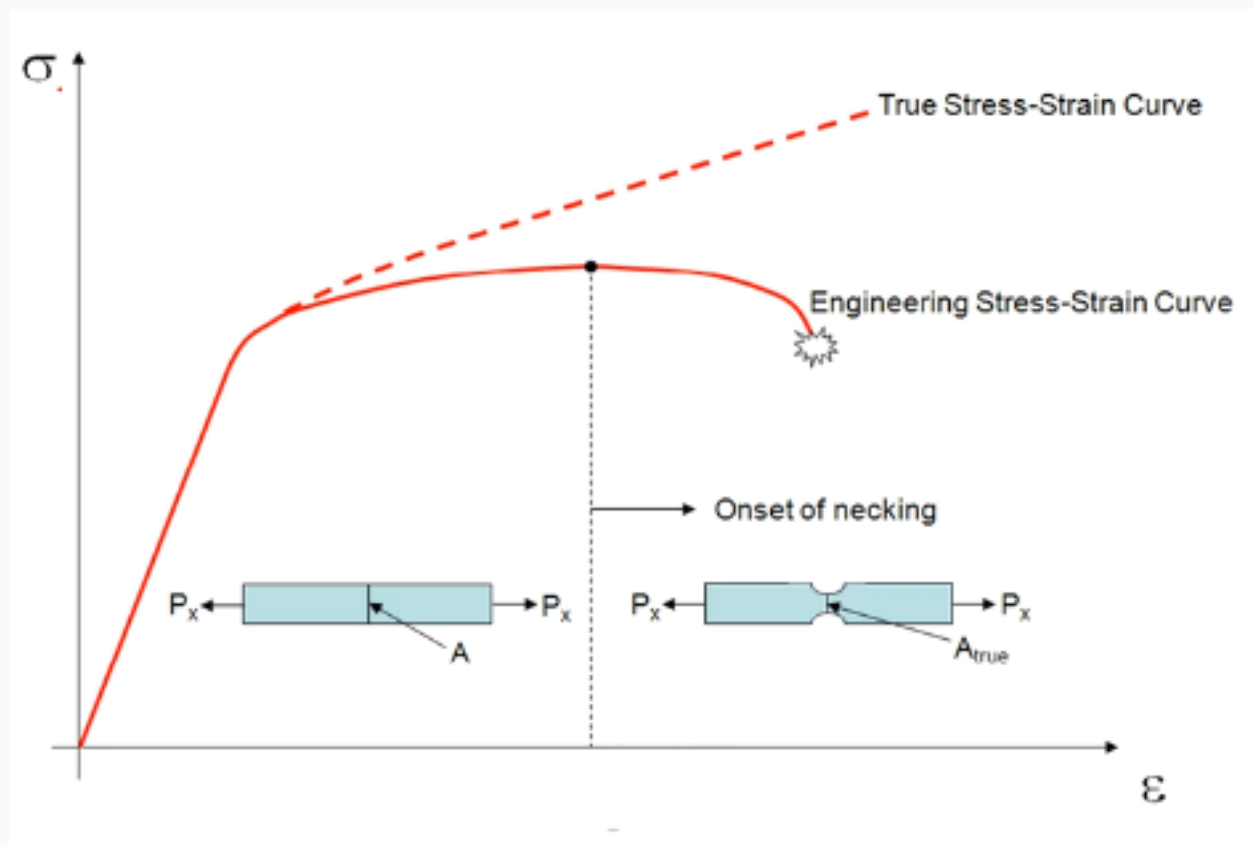


图 4: 真实应力形变

4.4.2 MAT1 定义方式

这个关键词用胡克定律定义了一种各向同性的线弹性材料。这个定律表示应力和应变之间的线性关系。它可用于桁架，梁 (仅 3 型)，壳和实体元件。格式

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
/MAT/LAW1/ <i>mat_ID</i> / <i>unit_ID</i> or /MAT/ELAST/ <i>mat_ID</i> / <i>unit_ID</i>									
<i>mat_title</i>									
	$\rho_i$								
	$E$	$\nu$							

图 5: MAT1 格式

Field	Contents	SI Unit Example
<i>mat_ID</i>	Material Identifier (Integer, maximum 10 digits)	
<i>unit_ID</i>	Unit Identifier (Integer, maximum 10 digits)	
<i>mat_title</i>	Material title (Character, maximum 100 characters)	
$\rho_i$	Initial density (Real)	$\left[\frac{\text{kg}}{\text{m}^3}\right]$
$E$	Young's modulus (Real)	[Pa]
$\nu$	Poisson's ratio (Real)	

图 6: MAT1 定义

```

#RADIOSS STARTER
#---1---|---2---|---3---|---4---|---5---|---6---|---7---|---8---|---9---|---10---|
/UNIT/1
unit for mat
      Mg      mm      s
#---1---|---2---|---3---|---4---|---5---|---6---|---7---|---8---|---9---|---10---|
#- 2. MATERIALS:
#---1---|---2---|---3---|---4---|---5---|---6---|---7---|---8---|---9---|---10---|
/MAT/ELAST/1/1
Steel
#
      RHO I
      7.85E-9      0
#
      E      nu
      210000      .3
#---1---|---2---|---3---|---4---|---5---|---6---|---7---|---8---|---9---|---10---|
#ENDDATA
/END
#---1---|---2---|---3---|---4---|---5---|---6---|---7---|---8---|---9---|---10---|

```

图 7: MAT1 例子

### 4.4.3 原理

1. 该材料适用于纯粹的弹性材料。材料的性能只取决于两个因素：

- (a). 杨氏模量:  $E$
- (b). 泊松比:  $\nu$
- (c). 剪切模量: (可计算)

$$G = \frac{E}{2(1 + \nu)}.$$

## 2. 应力应变关系:

$$\begin{bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ \gamma_{23} \\ \gamma_{31} \\ \gamma_{12} \end{bmatrix} = \frac{1}{E} \begin{bmatrix} 1 & -\nu & -\nu & 0 & 0 & 0 \\ -\nu & 1 & -\nu & 0 & 0 & 0 \\ -\nu & -\nu & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2(1+\nu) & 0 & 0 \\ 0 & 0 & 0 & 0 & 2(1+\nu) & 0 \\ 0 & 0 & 0 & 0 & 0 & 2(1+\nu) \end{bmatrix} \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{31} \\ \sigma_{12} \end{bmatrix}.$$

### 使用注意事项

- 密度值通常用于显式模拟，也可以用于静态隐式模拟，以便在拟静态分析中达到更好的收敛性。
- 当通过壳体厚度的积分点个数不同于 NP=1(膜) 时，对 LAW1 和壳单元 (/PROP/TYPE1 (shell)) 采用全局积分法。
- 失效模型在全局集成的情况下不可用。在这些情况下，具有很高屈服应力的 LAW2 和 LAW27 可以作为 LAW1 的替代品。

## 4.4.4 MAT1 代码解读

总体框架是

### 1. 获取输入

```
CALL HM_OPTION_IS_ENCRYPTED(IS_ENCRYPTED)
```



```

CALL HM_GET_FLOATV('MAT_RHO',  RHO0  ,IS_AVAILABLE,LSUBMODEL,UNITAB)
CALL HM_GET_FLOATV('Refer_Rho',RHOR  ,IS_AVAILABLE,LSUBMODEL,UNITAB)
CALL HM_GET_FLOATV('MAT_E',     YOUNG ,IS_AVAILABLE,LSUBMODEL,UNITAB)
CALL HM_GET_FLOATV('MAT_NU',    ANU   ,IS_AVAILABLE,LSUBMODEL,UNITAB)

IF(RHOR == ZERO ) RHOR=RHO0

```

```

IF (YOUNG<=ZERO) THEN
    CALL ANCMMSG(MSGID=683,
.           MSGTYPE=MSGERROR,
.           ANMODE=ANINFO,
.           I1=ID,
.           C1=TITR,
.           C2='YOUNG' 'S MODULUS')
ENDIF
IF(ANU==HALF) ANU=ZEP499

```

```

G=YOUNG/(TWO*(ONE+ANU))
CO=ZERO
C1=YOUNG/(THREE*(ONE-TWO*ANU))
EO=ZERO
E1MN2=YOUNG/(ONE-ANU**2)
EN1N2=ANU*E1MN2
SDSP  =SQRT(YOUNG/MAX(RHOR,EM20))
ISRATE = 0

```

## 3. 保存参数

```
PM(1) =RHOR
PM(20)=YOUNG
PM(21)=ANU
PM(22)=G
PM(23)=E0
PM(24)=E1MN2
PM(25)=EN1N2
PM(26)=FIVE_OVER_6
PM(27)=SDSP
PM(31)=C0
PM(32)=C1
PM(89) =RH00
```

## 4. 异常处理

```
C-----
      CALL INIT_MAT_KEYWORD(MATPARAM,"TOTAL")
      IF (ANU > 0.49) THEN
          CALL INIT_MAT_KEYWORD(MATPARAM,"INCOMPRESSIBLE")
      ELSE
          CALL INIT_MAT_KEYWORD(MATPARAM,"COMPRESSIBLE")
      END IF
      CALL INIT_MAT_KEYWORD(MATPARAM,"HOOK")
C-----
      WRITE(IOUT,1001) TRIM(TITR),ID,01
```



```
& 5X,'YOUNG''S MODULUS . . . . .=' ,E12.4/ ,  
& 5X,'POISSON''S RATIO . . . . .=' ,E12.4/ ,  
& 5X,'SHEAR MODULUS . . . . .=' ,E12.4//)
```

在数据里最重要的是‘PM’以及‘IPM’，分别存储了材料参数和材料 ID。但是暂时没有相关文档说明其具体内容。提取 mat 文件夹下所有 PM 变量的信息汇总在 [Github 在线文档](#) 里。

## 5 FEniCS

编译成功，但是调不出来。。。猜测可能没有加入环境变量。。。

直接使用二进制文件是可以的。(可以演示)

**Thank You!**