

## 卷首语

要学好 FLAC3D，必须学好 FISH，FISH 身为 FLAC3D 的内嵌语言，也就是是一种嵌入式编程语言，其编写短小精干实时，同时兼顾大家操作习惯，不但可以嵌入命令流文件里工作，而且还可以引用 FLAC3D 本身的任何命令，突破了一般标准程序代码的限制，实现了读者对 FLAC3D 软件的完美控制。

类似于 ANSYS 的 APDL 语言，FISH 主要是用来处理现有命令程序难以实现（较难或较烦）的一些问题，对于没有编程经验的用户，可以编一些简单的 FISH 函数，最好是修改 3DSHOP 中的一些比较现成的函数以便使用；当然，既然是一门编程语言，也可以编复杂程序。

写这一卷的主要目的就是使大家对 FISH 有个较为充分的认识和理解，因为相关资料甚少，故卷中的有些资料是作者在 Simwe, Yantubbs 等论坛上收集后整理所得，在此表示感谢。本卷分为 4 章，第一章为关于 FISH 语言法则、变量、函数等得一个综述，主要是想让大家从全局认识 FISH 编程语言；第二章介绍了 FISH 语句的类别和各自特点和用法；第三章则较为细致的向大家介绍了各变量，函数的具体含义，部分配合了用法实例；第四章补充了一些比较不错的 FISH 应用实例。

希望大家读后会有所收获，也欢迎大家批评指正，共同提高！

Sunshine

[ssw1216@163.com](mailto:ssw1216@163.com)

QQ 6192525

2007-9-20  
于铁道科学研究院  
北京·中国

# 第一章 Fish 的语言法则

## 一、四个概念

**函数与变量**—Fish 函数由 define 定义，end 结束定义。如下例为定义一个名叫 abc 的函数。

```
new
def abc
  hh = 22
  abc = hh * 3 + 5
end
```

上述函数中 hh 即为“变量”。函数和变量是 FISH 语言中非常基本和重要的两个对象，区别如下：

当我们调用一个 FISH 对象 XXX 时，如果 XXX 是函数，系统就会执行该函数；而如果 XXX 不是函数而是变量，系统则会简简单单的调用其当前置。大家注意继续输以下命令行，注意输出结果，便可对这两个有个较为清晰的认识。

```
set abc=0 hh=0
print hh
print abc
print hh
```

**语句**—在不少实际问题中有许多具有指示说明性的、控制性的、重复性操作，程序中需要执行他们，诸如 Array, Define, end, Case, endcase, If ,else, endif 等即为语句。我们举例来说明选择语句 if..else 的用法，如下：

昨天晚上俺和俺老婆在床上就商量，我俩掰腕子，我赢了我老婆亲我一下，我老婆赢了，我亲我老婆一下。

这个就是选择语句，如果 (if) 我赢了 (表达式) 我老婆亲我一下 (语句 1)，如果我输了 (else)，我亲我老婆一下 (语句 2)。

**代码行**—FISH 语言中的代码行，含义类似于命令流中的“命令行”。只是他们的写法规则有很大的不同，如下节所述。

## 二、两个法则

### 1. FISH 语言中的代码行的写法

FISH 语言中的代码行，其写法要求严格，必须是以下格式之一：

- I 代码行由语句打头，如 DEFINE, IF 等。
- II 代码行为 FISH 中定义的函数，需要说明的是多个函数间要用空格隔开。

比较好的一点是，FISH 代码中的函数名不用提前定义，只用在执行前定义即可。

III 代码行由赋值语句组成。

IV COMMAND-ENDCOMMAND 语句中间的 FLAC3D 命令行。

V 空行。

VI 分号 “;” 打头的行。

需注意的是 FISH 中的变量名、函数名以及语句，都要写全，不能象 FLAC3D 命令中那样缩写。此外，代码行不能续行，不过可以通过引入新变量解决此问题。还有一点，FISH 不区分大小写。

## 2 FISH 中函数名、变量名的取法

FISH 中的函数名或变量名必须以非 0 开始，并且名字中不能含有以下字符：

. , \* / + - ^ = < > # ( ) [ ] @ ; ' "

一般而言，除此要求外，用户可以随意取名，但应该避开 FISH 中预定义的变量和函数名，最好也避开 FLAC3D 中的命令以及其缩写形式。

还要注意一点，如果变量名没有被赋值，则默认为 0。

## 三、四种格式

这里所指的是，FISH 中变量以及函数的四种数据类型：

1. 整型数据（介于 -2147483648 和 +2147483648 之间的整数）。
2. 浮点型数据（精确到小数点后 15 位，介于  $10^{-300}$  到  $10^{300}$  之间）。
3. 字符型数据（也就是字符串，可以是任意长度，FLAC3D 中，其表示通常为加上单引号的一串字符）
4. 指示字（扫描单元、网格点等清单时的机器编址）。

通过赋值可以动态的改变 FISH 中变量的数据类型，这一点与 FORTRON, C 语言相比，更加便于新手理解和掌握。

## 四、两点说明

### 1. FISH 语言中的变量和函数名的时效

与 Basic 语言类似，FISH 语言中的变量名以及函数名在整个程序中都有效，不管你是在 FISH 代码中还是在 FLAC3D 命令行中。

### 2. 变量和函数名的调用格式

函数是 FISH 语言执行的唯一对象，它没有自变量，仅仅在调用时通过提前设置变量的值来传递变量。函数 XXX 可以通过以下方式调用：

I FISH 代码行直接引用函数名 XXX

II FISH 公式中被用到，如

```
new_var = (sqrt(xxx) / 5.6)^4;
```

III FLAC3D 命令行中直接引用

IV 在代码输入行中，作为替换对象

V 作为 SET, PRINT, HISTORY 等命令的参数。

## 五、一点补充

我们可以删除或重新定义 FISH 函数，方法是用同名的新代码取而代之，如果在 DEFINE 后直接跟 END 行，那么也就删除了原定义的函数。需补充的是：

I 本构模型不能删除或重新定义。

II 当函数被删除了，但原有变量依然存在，因为变量是全局性的，如同在其他位置一样。

III 替换后，对原函数的调用会自动为新函数所取代。

## 第二章 Fish 语句用法精讲

FISH 语言中的语句不多，可以分为几类：指示说明语句；控制语句；FLAC3D 的命令执行语句。本章重点介绍语句的用法。

### 一、两条指示语句

1. **ARRAY**      *var1(n1, n2 ...)* <*var2(m1, m2 ...)*> <*var3(p1, p2 ...)*> ...

在这个语句中，Var1 是任意有效的变量名，n1,n2...可以是整数，也可以是标量变量（维数）或 ARRAY 执行过程中为整型数值的单个变量。同一代码行也可以定义多个矩阵（如 Var 2）。每个矩阵可以有不同的维数。

ARRAY 语句是一个指示说明语句，它只在汇编过程中起作用，在执行过程中被忽略。使用时，需要注意以下几点：

1. 矩阵名可以是一个存在的单个变量，这样一来，变量便转化为一个矩阵并且原变量值会丢失。如果是一个不存在的变量名，就会新建。
2. 矩阵名不能是一个函数名或已经存在的矩阵名（也就是所，矩阵不能被重新定义）。
3. 矩阵维数（n1,n2...）一定要是正整数。
4. 矩阵的维数和元素个数没有限制。

矩阵变量的使用很类似于单个变量，不同的是，其后常常接自变量清单（带圆括号）。FLAC3D 命令行中对矩阵名的使用也有一定的限制，如 Print XXX (XXX 为一 FISH 矩阵名)，只会打印出最大的矩阵指数；SET grav=XXX 是不允许的。接下来介绍几个关于此条语句的实例，虽然很简单，但希望大家小心输入，注意输入后的变化。

实例 1：定义一个一维矩阵，并进行赋值。

```
def fuzhi
array num(20)
loop i(1,20)
num(i)=i^2+1
"wanjiabaodian"=out(num(i))
endloop
end
fuzhi
```

实例 2：定义一个用 FISH 定义奇数和偶数组的例子。

```

def number
  array even(21),odd(21);even为偶数数组, odd为奇数数组
  m1=0
  m2=0
  loop n(0,41)
    a=n/2.0
    b=int(n/2)
    if a = b then
      m2=m2+1
      even(m2)=n
    else
      m1=m1+1
      odd(m1)=n
    endif
  endloop
end
number
def wri
  loop k(1,21)
    "wanjiabaodian"=out(even(k))
  endloop
command
  pause
endcommand
  loop k(1,21)
    "wanjiabaodian"=out(odd(k))
  endloop
end
wri

```

## 2. WHILESTEPPING 语句

在一个用户自定义函数中，只要出现 whilestepping 语句，在每一个 FLAC3D 计算步开始时，函数都会自动执行。该语句的属性可以通过 SET fishcall 0 remove 命令修改失效。FISHCALL 语句可以提供更为自由的计算控制，所以该语句应该不多，此处不作详细介绍。

### 二、五对控制性语句

控制语句在 FISH 中十分重要，如下进行一一介绍。

#### 1. 函数语句

```
DEFINE          function-name
```

```
END
```

#### 2. 条件语句

```

CASEOF      expr
CASE        n
ENDCASE

```

该语句的语法结构为：

```

CASEOF  expr
;..... default code here
CASE i1
;..... case i1 code here
CASE i2
;..... case i2 code here
CASE i3
;..... case i3 code here
ENDCASE

```

在这个结构中，CASEOF 后的 Expr 对象可以是任何有效的数学表达式，经过计算后转化为整型数据；而词条 i1, i2.....等则必须是 0~255 中间的整型，不能为符号。如果 Expr 等于 i1，系统会自动跳到 CASE i1 后语句执行，直到碰到下一条 CASE 语句，系统才会到 ENDCASE 完成执行任务，类似 i2, i3, i4.....时原理完全一样，只是要说明一点，如果 Expr 不等于 iN 中的任何一个，系统会执行第 1 个 CASE，如果 N 小于 0 或大于 255，系统就会出错停止执行。i1, i2....并不要求一定是连续或相邻整数。

下例为含有该句型用法的部分 FISH 流：

```

caseof _tielevel

case 1
_tslope = -0.466 ; slope of tieback
_length1 = 9.674 ; distance from beam to break in tieback
_length2 = 9.897 ; distance from beam to other side of break in tieback
_lengthtot = 21.897 ; total length of tieback
_tload = 800000 ; pretension force
_tsegs1 = 5 ; number of tieback segments in section next to beam
_tsegs2 = 10 ; number of tieback segments in section away from beam

case 2
.....

set _tielevel = 1
_createTie
; cable 1 - tieback1 section closest to the beam
; cable 2 - tieback1 section furthest from the beam
sel cable id 1 prop emod 2.05e11 gr_coh 0.0 gr_fric 0.0 gr_k 0.0
sel cable id 1 ycomp 1.0e5 ytens 15.34e5 xcarea 5.54e-3
sel cable id 2 prop emod 2.05e11 gr_coh 1.5e10 gr_fric 25.0 gr_per 0.264
sel cable id 2 gr_k 0.56e9 ycomp 1.0e5 ytens 15.34e5 xcarea 5.54e-3

```

### 3. 条件语句

```

IF                expr1 test expr2 THEN

ELSE

ENDIF

```

条件语句 **IF** 是所有计算机语言中较为常用的语句，在该对语句中，通常 **ELSE** 为可选项，**THEN** 可以省略不写。该语应用较为普遍，本节只简单介绍 FISH 语言中几个需要注意的问题：

I **TEST** 词条常常由以下符号组成：

= # > < >= <=      其中，# 号意为  $\neq$ 。

II **EXPR1** 和 **EXPR2** 可以是任意的有效代数式，可以包括函数，FLAC3D 变量等等。

III 如果 **TEST** 结果为真，**IF** 后语句就会执行，直到执行下一条 **ELSE** 或者 **ENDIF**；但如果测试结果为假，系统就会执行 **ELSE** 和 **ENDIF** 之间的语句，当然了，前提是存在 **ELSE** 语句，呵呵。另外，**ENDIF** 执行后，系统自动跳回第一行。

IV 当 **EXPR1** 和 **EXPR2** 为字符串时，测试附后只能时两种，即 = # ，对字符串而言，其他的操作时无效的，必须很精确的对等；指示字也一样。

V 该语句可以嵌套到任何路径。

举个简单的实例，以理解其用法：

```

new
def abc
  if xx > 0 then
    abc = 33
  else
    abc = 11
  end_if
end
set xx = 1
print abc
set xx = -1
print abc

```

### 4. 循环结构语句

FISH 中采用的循环结构语句只有 **LOOP** 语句，没有 **FOR-NEXT** 等语句，其结构为

```

LOOP          var (expr1, expr2)

ENDLOOP

or

LOOP          WHILE expr1 test expr2

ENDLOOP

```

第 1 种 LOOP 形式中，采用整型计数，VAR 为 EXPR1 的初值，每次循环后加 1，直到 EXPR2 值，注意的是 EXPR 可以是任意的表达式。第 2 种 LOOP 形式中，当条件为真时，执行循环体，否则，控制路径转到 ENDLOOP。

对比两种形式，第 1 种 LOOP 形式在循环结束后进行条件测试，所以循环至少要执行 1 次；第二种形式在循环开始时进行条件测试，所以一旦初始条件为加，直接跳出循环。

举个简单的实例，以理解其用法：

```

new
def xxx
  sum = 0
  prod = 1
  loop n (1,10)
    sum = sum + n
    prod = prod * n
  end_loop
end
xxx
print sum, prod

```

## 5 跳转语句

```

SECTION

ENDSECTION

```

FISH 语言中没有 GO TO 命令，SECTION 允许程序以某种可控制的方式跳转，SECTION.....ENDSECTION 语句可包含任意行的 FISH 代码，且不影响代码的运行。然而，EXIT SECTION 语句会导致 SECTION 语句无条件的直接跳转到代码段结束的位置。

这对语句我没有用过，呵呵。其实控制语句还有 EXIT，会使程序无条件跳到当前函数的结束处。

### 三、一对命令执行语句

FISH 中可以调用 FLAC3D 命令，那么你用到了以下这对语句：

**COMMAND****ENDCOMMAND**

从一个 FISH 程序中直接调用 FLAC3D 命令通常有两个原因:一个是用 FISH 函数的预定义变量（见第 3 章介绍）实现很难，另一个原因是我们可以通过 FISH 来控制 FLAC3D 的整个运行过程。

**四、FISH 语言与 FLAC3D 命令的对接****1. 几个直接调用 FISH 的 FLAC3D 命令**

可以直接调用 FISH 的 FLAC3D 命令有：HISTORY, PRINT, SET, TITLE。这几个命令的用法都很简单，不一一介绍。语法结构为：

<b>HISTORY</b>	<i>var</i>
<b>PLOT</b>	<b>fish name</b>
<b>PRINT</b>	<i>var</i>
<b>PRINT</b>	<b>fish</b>
<b>PRINT</b>	<b>fishcall</b>
<b>SET</b>	<b>fishcall <i>n</i> &lt;remove&gt; name</b>
<b>SET</b>	<i>var value</i>
<b>TITLE</b>	<i>str</i>

通过实例说明各自用法：

例 1 HISTORY 命令在 FISH 中的用法：

```

new
gen zone brick size 1 2 1
model mohr
prop shear=1e8 bulk=2e8 &
    cohes=1e5 tens=1e10
fix x y z          range y -0.1 0.1
apply yvel -1e-5 range y 1.9 2.1
def get_ad
    ad1 = gp_near(0,2,0)
    ad2 = gp_near(1,2,0)
    ad3 = gp_near(0,2,1)
    ad4 = gp_near(1,2,1)
end

get_ad
def load
    load=gp_yfunbal(ad1)+gp_yfunbal(ad2)+gp_yfunbal(ad3)+gp_yfunbal(ad4)
end
hist load
hist gp ydis 0,6,0
step 1000
plot his 1 vs -2

```

例 2 PRINT 命令调用 FISH 变量和函数的方法:

```

new
def abc
    abc = hh * 3 + 5
end
set hh=22
print abc
set abc=0 hh=0
print hh
print abc
print hh

```

例 3 SET 命令的赋值调用方法:

```

set abc=0 hh=0
print hh
print abc
print hh

```

例 4 SET FISH 调用 FISH 函数。

**SET**            **fishcall** *n* <remove> *name*

FISH 函数名 NAME 可以被 FLAC3D 调用,调用位置由 FISHCALL 的识别码 ID 号 n 决定, REMOVE 为可选项, 当会使函数名为 NAME 的 FISH 函数从清单中被删除。

Location in Calculation Cycle	CONFIG Mode
initialization	
geometry update in large strain	
<b>fishcall 1</b>	all modes
top of main cycle loop (after incrementing cycle count)	
<b>fishcall 2</b>	all modes
timestep calculated	
<b>fishcall 0</b>	all modes
begin mechanical calculation	
<b>fishcall 3</b>	mechanical only
constitutive models called; gridpoint forces calculated	mechanical only
<b>fishcall 4</b>	mechanical only
motion calculations	
<b>fishcall 5</b>	mechanical only
velocities passed to slaves	
<b>fishcall 6</b>	mechanical only
<b>fishcall 7</b>	thermal only
thermal-stress calculation	if thermal mode active
<b>fishcall 8</b>	thermal only
<b>fishcall 9</b>	fluid-flow only
groundwater-flow calculation	if fluid-flow mode active
<b>fishcall 10</b>	fluid-flow only
total stress update	if fluid-flow mode active
<b>fishcall 12</b>	fluid-flow only
<b>fishcall 13</b>	all modes
end of main cycle loop	
<b>fishcall 14</b>	left double-click with mouse
<b>fishcall 15</b>	right double-click with mouse

## 2. FISH 函数的执行

一般而言，FLAC3D 和 FISH 操作是“各走各的道”，FISH 语句不能象 FLAC3D 命令那样直接给出，而 FLAC3D 命令也不能象 FISH 程序那样直接运行，两个体系可通过以下的方法完成相互的作用：

- I 函数的直接调用
- II 利用 HISTORY 变量
- III 在计算过程中自动执行
- IV 利用函数去控制一个运行过程

有一点值得注意，就是 FLAC3D 命令中任何位置的数字可以由 FISH 符号（变量或函数）来代替，这就了不起了，我们可以由 FISH 符号建立数据文件，而不是通过具体的数值，具体要通过实例来说明。

例 1 该例向大家展示的是如何创建一个数据文件，我们只要通过改变两个 SET 命令中的数值就可以实现不同单元密度隧道模型的创建。

```
;--- function definition ...
def make_hole
  x_bou = 400.0
  y_bou = 200.0
  z_bou = 200.0
  rad   = 20.0
  n_1   = n_inner
  n_2   = n_total
  n_3   = 2 * n_inner
  n_4   = n_total - n_inner
end
; FLAC3D input ---
set n_inner 5 n_total 10
make_hole
; create grid
gen zone radcylinder size n_1,n_2,n_3,n_4 &
  p0 (0,0,0) p1 add (x_bou,0,0) p2 add (0,y_bou,0) p3 (0,0,z_bou) &
  dim (rad,rad,rad,rad)
```

在这个例子中，MAKE\_HOLE 函数作为单独代码行来直接调用，我们通过提前进行 SET 命令实现了对他的控制，这里在 FLAC3D 中没有一个数值的输入，因为他们被我们用 FISH 符号取代了。

## 第三章 常用 Fish 实例

### 一、生成空间水面得有效方法（君之枫原创）

```

new
gen zone brick size 3 3 3 &
p0 (0,0,0) p1 (3,0,0) p2 (0,3,0) p3 (0,0,1.5) &
p4 (3,3,0) p5 (0,3,1.5) p6 (3,0,4.5) p7 (3,3,4.5)
group Base
; Create Top - 1 unit high for initial spacing
gen zone brick size 3 3 3 &
p0 (0,0,2.5) p1 (3,0,5.5) p2 (0,3,2.5) p3 (0,0,7) &
p4 (3,3,5.5) p5 (0,3,7) p6 (3,0,7) p7 (3,3,7)
group Top range group Base not
;
; Create interface elements on the top surface of the base
interface 1 face range plane norm (-1,0,1) origin (1.5,1.5,3) dist 0.1
ini z add -1.0 range group Top

water dens 1000
set grav 0 0 -10

def water_table
  p_i=i_head
  p_ie=i_elem_head(p_i) ;界面 1 单元指针赋予 p_ie
  loop while p_ie # null
;返回界面单元的三个顶点的地址
    p_gp1=ie_vert(p_ie,1)
    p_gp2=ie_vert(p_ie,2)
    p_gp3=ie_vert(p_ie,3)

```

;以这三个界面单元顶点为水面的三个顶点，生成水面  
;=====interface 的节点和网格节点的地址是分别存放的

```
x1=in_pos(p_gp1,1)
y1=in_pos(p_gp1,2)
z1=in_pos(p_gp1,3)
x2=in_pos(p_gp2,1)
y2=in_pos(p_gp2,2)
z2=in_pos(p_gp2,3)
x3=in_pos(p_gp3,1)
y3=in_pos(p_gp3,2)
z3=in_pos(p_gp3,3)
command
    water table face x1,y1,z1  x2,y2,z2  x3,y3,z3
endcommand
    p_ie=ie_next(p_ie)
endloop
end
model null range group Base
model null range group Top
water_table
int 1 del

plo water
```

## 二、边坡稳定性计算通用程序

**new**

**;set mech ratio 1e-6**

**def \_geoparm**

**k=1.130** ;安全系数

**size1=40**

**size2=10**

**filecal0=string(size1)+'\_'+string(size2)+'\_cal'+'.sav'**

**fosfile0=string(size1)+'\_'+string(size2)+'\_fos'+'.sav'**

**bf=45** ; 改变此处坡角

**af=bf\*pi/180** ;

**h1=20**

**;**

**x00=30.0**

**x01=105.0**

**x02=x00**

**x03=x00+h1/tan(af)**

**x04=x01**

**x05=x03**

**x06=105.0**

**x07=x06**

**;**

**y00=0.0**

**y01=y00**

**y02=1.0**

**y03=y00**

**y04=y02**

**y05=y02**

**y06=y00**

**y07=y02**

;

**z00=20.0**

**z01=z00**

**z02=z00**

**z03=40.0**

**z04=z00**

**z05=z03**

**z06=z03**

**z07=z03**

**end**

**\_geoparm**

**gen zone brick p0 0 0 0 p1 30 0 0 p2 0 1 0 p3 0 0 20 size 12 1 8**

**gen zone brick p0 30 0 0 p1 105 0 0 p2 30 1 0 p3 30 0 20 size size1 1 8**

**gen zone brick p0 x00 y00 z00 p1 x01 y01 z01 p2 x02 y02 z02 p3 x03 y03 z03 &**

**p4 x04 y04 z04 p5 x05 y05 z05 p6 x06 y06 z06 p7 x07 y07 z07 size**

**size1 1 size2**

**def modelMaterial**

**; 弹性模量(Pa)**

**flag=0**

**E1=100e6**

**; 泊松比**

**poi1=0.35**

**;**

**; 粘结力(Pa)**

**coh1=42e3/k**

**;**

**; 容重(N/m3)**

**weight1=25e3**

**; 膨胀角(度)**

```

dila1=0.0
;
; 内摩擦角(度)
fri1=(atan((tan(20.0*pi/180))/k))*180/pi
beta1=(sin(20.0*pi/180))/(1-2*poi1)
;
afa1=E1*poi1
poi1=0.5*(1-(sin(fri1*pi/180))/beta1)
E1=afa1/poi1
;抗拉强度
ten1=0.01e6
grav0=-9.80 ; 重力加速度(N/kg)
dens1=-weight1/grav0
K1=E1/(3*(1-2*poi1))
G1=E1/(2*(1+poi1))
end
modelMaterial
model mohr ;采用莫尔库仑模型
pro bulk K1 she G1 dens dens1 coh coh1 &
friction fri1 dilation 0.
fix x range x -0.1 0.1
fix x range x 104.9 105.1
fix y
fix x y z range z -0.1 0.1
set grav 0 0 grav0
solve FOS
save 1_130.sav

```

### 三、为水面上下指定干湿密度

```
new
```

```
restore kw3-hou3.sav
```

```
ini xdis=0 ydis=0 zdis=0
```

```
water dens 1000
```

```
water table origin 114.13,3,115.50 normal 16.05,0,43.46 range x 103 199.1 group qht
```

```
water table origin 202.85,3, 81.54 normal 15.99,0,24.80 range x 199.1 206.52 group qht
```

```
water table origin 24.08,3,139.44 normal 11.01,0,43.46 range x -1.24 103 group zht
```

```
water table origin 114.13,3,115.50 normal 16.05,0,43.46 range x 103 199.1 group zht
```

```
water table origin 202.85,3, 81.54 normal 15.99,0,24.80 range x 199.1 207.7 group zht
```

```
;ini state 0
```

```
:: initialize density: saturated and unsaturated
```

```
def ini_dens
```

```
  pnt = zone_head
```

```
  loop while pnt # null
```

```
    if z_pp(pnt) # 0.0 then
```

```
      z_density(pnt) = 1960
```

```
    else
```

```
      z_density(pnt) = 1900
```

```
    endif
```

```
    pnt = z_next(pnt)
```

```
  endloop
```

```
end
```

---

```
ini_dens range group qht
```

```
def ini_dens1
```

```
  pnt = zone_head
```

```
  loop while pnt # null
```

```
    if z_pp(pnt) # 0.0 then
```

```
      z_density(pnt) = 2060
```

```
    else
```

```
      z_density(pnt) = 2000
```

```
    endif
```

```
    pnt = z_next(pnt)
```

```
  endloop
```

```
end
```

```
ini_dens1 range group zht
```

#### 四、施加应力（从小到大）

```
new
gen zone brick size 6 6 6
model elas
prop bulk 1e8 shear 7e7
fix x range x -0.1 0.1
def x_stress
    x_stress = stress_inc * step
end
set stress_inc = -1e3
apply sxx 1.0 hist x_stress range x 5.9,6.1 y 0,6 z 0,2
hist zone sxx 6,0,0
hist x_stress
step 100
```

#### 五、施加应力（从小到大）

```
new
gen zone brick size 6 6 6
model elas
prop bulk 1e8 shear 7e7
fix x range x -0.1 0.1
table 1 0,0 100,-1e5
apply sxx 1.0 hist table 1 range x 5.9,6.1 y 0,6 z 0,2
hist zone sxx 6,0,0
step 100
```

## 六、施加松弛应力（从大到小）

```
new
gen zone radcyl p0 0 0 0 p1 add 18 0 0 p2 add 0 8 0 p3 add 0 0 18 &
    dim 1.75 1.75 1.75 1.75 ratio 1.0 .83 1.0 1.2 size 6 6 6 10
gen zone radcyl p0 0 8 0 p1 add 18 0 0 p2 add 0 8 0 p3 add 0 0 18 &
    dim 1.75 1.75 1.75 1.75 ratio 1.0 1.2 1.0 1.2 size 6 6 6 10 fill
mo mo
pro bulk 33.33e9 she 25e9 fric 45 coh 15e6 ten 5e6
ini sxx -65e6 syy -65e6 szz -65e6
fix x range x -.1 .1
fix x range x 17.9 18.1
fix y range y -.1 .1
fix y range y 15.9 16.1
fix z range z -.1 .1
fix z range z 17.9 18.1
def relax    ;;:定义一个随时间改变的应力条件（应力释放）
    if step<ncyc then
        relax=1.0-(float(step)/float(ncyc))
    else
        relax=0.0
    end_if
end
set ncyc=1000
app nstress -65e6 hist relax range cyl end1 0,0,0 end2 0,8,0 r 1.75
hist unbal
hist gp xdisp 1.75 0 0
hist gp zdisp 0 0 1.75
cyc 2000
```

## 七、生成不规则表面

```

new
gen zone brick size 15 15 10 p0 0,0,0 edge=100.0
model elastic
prop shear 3e8 bulk 5e8
pause
def mountain
  gp = gp_head
  loop while gp # null
    zz = sqrt(gp_xpos(gp)^2 + gp_ypos(gp)^2)
    dz = 0.06 * sin(0.2 * zz + 100.0)      ; Sum Fourier terms for
    dz = dz + 0.06 * sin(0.22 * zz - 20.3) ; quasi-random surface
    dz = dz - 0.04 * sin(0.33 * zz + 33.3) ; topology.
    gp_zpos(gp) = 0.5 * gp_zpos(gp) * (1.0 + dz)
    gp = gp_next(gp)
  end_loop
end
mountain

```

## 八、求轴向力与轴向位移之间得关系

;求轴向应力

```

def ax_str
str = 0
pnt = gp_head
loop while pnt # null
if gp_zpos(pnt)<0.1 then
str = str + gp_zfunbal(pnt)
endif
endloop
ax_str = str / 100 ; cylinder radius = 1
end

```

```
ax_str

hist n 1
hist gp ydisp 0,0,0
hist ax_str
hist gp xdisp 1,1,0

plot hist -2 vs 1;;;;;;axial stress vs axial disp

step 1000
save beforeplzones.sav
```

### 九、求塑性剪应变大于 0.2 的塑性区

; Plot of plastic region as zones with strain > 0.2

```
def ShowPlasticZones
  zp = zone_head
  loop while zp # null
    if z_prop(zp,'es_plastic') > 0.2
      z_group(zp) = 'yield'
    else
      z_group(zp) = 'other'
    endif
    zp = z_next(zp)
  endLoop
end
ShowPlasticZones
plo crea qqq
plo add surf red range group yield
plo add axes green
plo set rot 123 313 3
```

```
plo set mag 1.5
```

## 十、定义安全度

安全度问题

1.计算之前 config zextra 1

2.根据计算结果，计算 zone 的点安全度 f，赋给 zextra 1

```
z_extra(p_z,1) = f
```

3.绘图 pl con zextra 1

```
p_z = zone_head
```

```
loop while p_z # null
```

```
    计算安全度 f
```

```
    z_extra(p_z,1) = f
```

```
    p_z = z_next(p_z)
```

```
endloop
```

## 十一、塑性区体积计算

```
def get_plast
```

```
    shearnow = 1
```

```
    tensionnow = 2
```

```
    shearpast = 4
```

```
    tensionpast = 8
```

```
    v_shear_now = 0
```

```
    v_tension_now = 0
```

```
    v_shear_past = 0
```

```
    v_tension_past = 0
```

```
    p_z = zone_head
```

```
    loop while p_z # null
```

```
        if and(z_state(p_z,0),shearnow) = shearnow then
```

```

        v_shear_now = v_shear_now + z_volume(p_z)
    endif
    if and(z_state(p_z,0),tensionnow) = tensionnow then
        v_tension_now = v_tension_now + z_volume(p_z)
    endif
    if and(z_state(p_z,0),shearpast) = shearpast then
        v_shear_past = v_shear_past + z_volume(p_z)
    endif
    if and(z_state(p_z,0),tensionpast) = tensionpast then
        v_tension_past = v_tension_past + z_volume(p_z)
    endif
    p_z = z_next(p_z)
endloop
ii = out('shear_now : ' + string(v_shear_now))
ii = out('tension_now : ' + string(v_tension_now))
ii = out('shear_past : ' + string(v_shear_past))
ii = out('tension_past : ' + string(v_tension_past))
end
get_plast
十二、 求算应变
;求应变
def ax_str
str1 = 0
n=1.0
array arr(6)
pnt1 = zone_head
loop while pnt1 # null
if z_zcen(pnt1) <0.1 then
oo = z_fsi(pnt1,arr)
str1 = str1 -arr(3)

```

---

```
endif  
pnt1 = z_next(pnt1)  
n=n+1.0  
endloop  
ax_strain = str1 / n ; pi 面积
```