

## FLAC3D 实例分析教程

刘 波 韩彦辉（美国）编著 《FLAC 原理实例与应用指南》

北京：人民交通出版社，2005.9     **Appendix(附录)**

版权所有：人民交通出版社，Itasca Consulting Group, USA

说 明：

1. 本实例分析教程是为方便读者学习、应用 FLAC 和 FLAC3D 而编写的，作为《FLAC 原理、实例与应用指南》一书的附录。

2. 计算算例参考了 Itasca Consulting Group 的培训算例，命令流的解析旨在方便读者理解 FLAC 和 FLAC3D 建模及求解问题的一般原则与步骤。

3. 实例分析的算例中，FLAC 算例是基于 FLAC 5.00 版本、FLAC 3D 算例是基于 FLAC 3D 3.00 版本实现计算分析的。读者在学习和研究相关算例时，请务必采用 Itasca 授权的合法版本进行分析计算。

4. 本实例分析教程仅供读者参考，读者在参考本教程算例进行工程分析时，编者对可能产生的任何问题概不负责。

编 者

2005.10.18

## 实例分析 1：基坑开挖

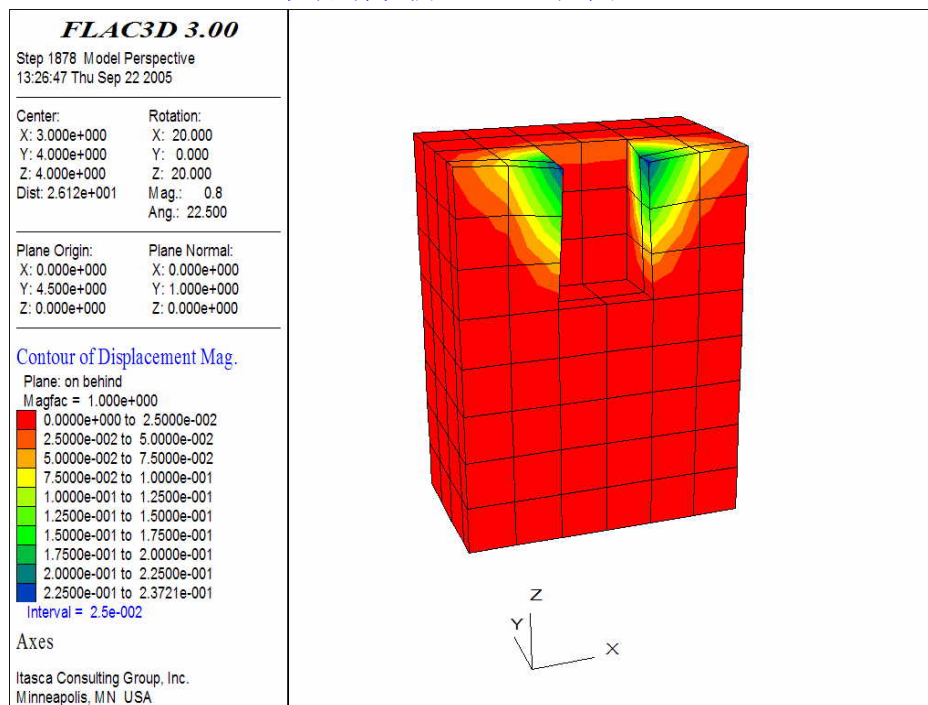


图 1：基坑开挖的位移等值线图

## 例 1：

; simple tutorial – trench excavation

;简单的隧道开挖方法

; Step 1: create initial model state

;第一步：初始模型的建立

; create grid

; 建立网格

gen zone brick size 6 8 8

;建立矩形的网格区域，其大小为  $6 \times 8 \times 8$ 

; assign material

;指定材料的性质

model mohr

;采用摩尔-库仑模型

prop bulk 1e8 shear 0.3e8

;模型的材料性质：体积模量  $1 \times 10^8$  剪切模量  $0.3 \times 10^8$ 

prop fric 35 coh 1e3 tens 1e3

;摩擦角  $35^\circ$  粘聚力  $1 \times 10^3$  抗拉强度  $1 \times 10^3$ 

; set global conditions

;设置全局参数

set grav 0,0,-9.81

```
;设置重力加速度, z 坐标正方向为正, 故为-9.81
ini dens 2000
;初始密度为 2000
; set boundary conditions
;设置初始边界情况
fix x range x -0.1 0.1
;在 x 方向上 固定 x = 0 边界, 为滚动支撑
fix x range x 5.9 6.1
;在 x 方向上 固定 x = 6 边界, 为滚动支撑
fix y range y -0.1 0.1
;在 y 方向上 固定 y = 0 边界, 为滚动支撑
fix y range y 7.9 8.1
;在 y 方向上 固定 y = 8 边界, 为滚动支撑
fix z range z -0.1 0.1
;在 z 方向上 固定 z = 0 边界, 为滚动支撑
; monitor model variables to check for equilibrium
;监控模型变量, 并控制力学平衡
hist unbal
;监测不平衡力, 并保留历史记录 (在默认情况下, 每十步做一次记录)
hist gp zdisp 4,4,8
;监测网格坐标点 (4, 4, 8) 在 z 方向的位移, 并保留历史纪录
solve
;运算, 求解(默认情况下 ratio=  $1 \times 10^{-5}$  停止运算)
pause
;暂停
save t1.sav
;形成 sav 文件, 并保存为 t1.sav
pause
;暂停
;
rest t1.sav
;恢复 t1.sav 文件
; Step 2: excavate trench
;第二步: 开挖隧道
model null range x 2,4 y 2,6 z 5,10
;在 x (2, 4) y (2, 6) z (5, 10) 范围内建立零模型 (即开挖这一部分区域)
pause
;暂停
set large
;在全局下设置大应变状态
initial xdis 0.0 ydis 0.0 zdis 0.0
;把 x, y, z 重新设置为 0 (因为我们只为了观察基坑引起的位移变化,
;而不是从施加重力荷载引起的位移变化, 这不会影响计算结果)
pause
```

;暂停

step 1000

;运算 1000 步

pause

;暂停

save t2.sav

;形成 sav 文件，并保存为 t2.sav

ret

;放在批处理文件的最后，以返回  $FLAC^{3D}$  的控制状态

---

## 实例分析 2：浅基础的稳定性分析

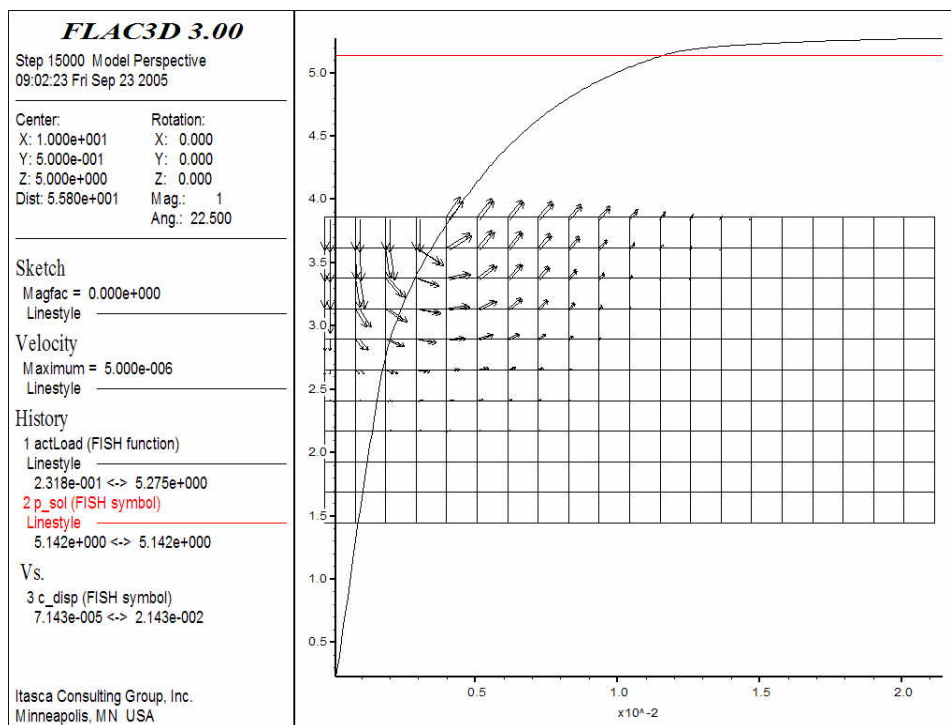


图 2：浅基础位移矢量图

## 例 2：

```

;-----
; -二维的条形基础在 Tresca 材料中的塑性流动
;-----
gen zone brick size 20 1 10
;建立矩形的网格区域，其大小为 20×1×10
model mohr
;采用摩尔-库仑模型
prop bul 2.e8 shea 1.e8 cohesion 1.e5
;模型的材料性质：体积模量  $2.0 \times 10^8$  剪切模量  $1.0 \times 10^8$  粘聚力  $1.0 \times 10^5$ 
prop friction 0. dilation 0. tension 1.e10
;内摩擦角  $0^\circ$  剪胀角  $0^\circ$  抗拉强度  $1.0 \times 10^{10}$ 
fix x range x -.1 .1
;在 x 方向上固定边界  $x=0$ （即只在 x 方向上受约束）
fix x y z range z -.1 .1
;在 x y z 方向上固定边界  $z=0$ （即为固定约束）
fix x y z range x 19.9 20.1
;在 x y z 方向上固定边界  $x=20$ 
fix y

```

```

;约束所有 y 方向上运动
fix x y z range x -.1 3.1 z 9.9 10.1
;在 x y z 方向上固定平面 x=0, 3 z= 10
ini zvel -0.5e-5 range x -.1 3.1 z 9.9 10.1
;在平面 x=0, 3 z= 10 上初始化 z 方向的速度为  $-0.5 \times 10^{-5}$ 
def p_cons
;用 fish 语言定义函数 p_cons
    pdis1 = gp_near(0.,0.,10.)
;靠近坐标 (0, 0, 10) 的节点的地址赋予 pdis1
    pdis2 = gp_near(0.,1.,10.) ;
;靠近坐标 (0, 0, 10) 的节点的地址赋予 pdis2
    p_sol = (2. + pi)
;2 +  $\pi$  赋予 p_sol
end
;用于 fish 语言的结尾
p_cons
;运行函数 p_cons
;-----
; p_load : average footing pressure / c
;场地的平均压力
; c_disp : magnitude of vertical displacement at footing center / a
;场地中心的垂直沉降量
;-----
def actLoad
;定义函数 actLoad (活荷载) (fish 语言的起始句)
    pnt = gp_head
;把网格点指针的首个地址 (gp_head) 赋予变量 pnt
    pload = 0.0
;定义变量 pload 为 0
    n = 0
;定义变量 n 为 0
    loop while pnt # null
;loop 是循环命令, 其条件是 pnt ≠ 0
        if gp_zpos(pnt) > 9.9 then
;当变量 pnt 所对应的 z 坐标大于 9.9
            if gp_xpos(pnt) < 3.1 then
;且 x 坐标小于 3.1 时
                pload = pload + gp_zfunbal(pnt)
;变量 pload 为上次循环中的值与变量 pnt 所对应的 z 方向上的不平衡力的和
            n = n + 1
;n 自我加 1
        endif
;结束里面的 if
    endif

```

```
; 结束外面的 if
    pnt = gp_next(pnt)
; 把网格点指针的下一个地址 (gp_next) 赋予变量 pnt
endloop
; 结束循环
    actLoad = pload / (3.5 * z_prop(zone_head,'cohesion'))
; 这里函数 actLoad 的值, 为后面的式子所赋予, 其中 pload 为循环结束后的最终
; 值 z_prop(zone_head,'cohesion') 为首个单元的粘聚力的数值
    c_disp = -(gp_zdisp(pdis1) + gp_zdisp(pdis2)) / 7.0
; 同理, 这里函数 c_disp 的值, 也为后面的式子所赋予, 其中 gp_zdisp(pdis1) 为
; 变量 pdis1 所对应的网格点在 z 方向的位移, gp_zdisp(pdis2) 变量 pdis2 所对应
; 的网格点在 z 方向的位移

end
; fish 语言的结束句
def p_err
; 定义函数 p_err
    p_err = 100. * (actLoad - p_sol) / p_sol
; 函数 p_err 的值为后面式子所定义
end
; fish 语言的结束句

hist n 50
; 对 n 做历史记录
hist actLoad
; 对 actLoad 做历史记录
hist p_sol
; 对 p_sol 做历史记录
hist c_disp
; 对 c_disp 做历史记录
hist unbal
; 对 unbal (不平衡力) 做历史记录

plot sketch
; 绘出轮廓线
plot add vel
; 添加矢量
plot add his 1 2 vs 3
; 绘出历史记录 n, actLoad, p_sol

cyc 15000
; 运行 15000 时步
print p_err
; 输出函数 p_err
```

save pran.sav

;形成 sav 文件，并保存为 pran.sav

ret

;放在批处理文件的最后，以返回  $FLAC^{3D}$  的控制状态

---



## FLAC3D 文件夹

*FLAC<sup>3D</sup>* 文件夹的内容

文件夹	内容
Flac3d300...*	可执行编码, DLLs, 安装数据文件
3Dshop...	六面体网格的数据文件
CommandReference	参考命令的数据文件
FISHinFLAC3D...	FLAC3D 中的 FISH 数据文件
FluidMechanicalInteraction	关于流体力学的相互作用的数据文件
OptionalFeatures	可选择属性的数据文件
StructuralElements	结构单元的数据文件
TheoryBackgroud...	理论和背景的数据文件
UserGuide	用户指南的数据文件
VerificationProblems...	有关某些确定问题的实例的数据文件
Resource	FLAC3D 的资源文件
Manuals	
Flac3d300...	<i>FLAC<sup>3D</sup></i> 用户指南 (PDF 格式)
Shared	
Drivers	驱动程序安装指南
Fishtank	<i>FISH</i> 函数的 Itasc 编码
Models...	Itasc 用户定义的模型
Resource	Itasc 资源文件
Utility...	“更新”和“动画”功能

\* 这里和下面的(...) 表示此文件夹下还有子文件夹。

## FLAC3D 网格生成

### 实例分析 4：网格的合并联结

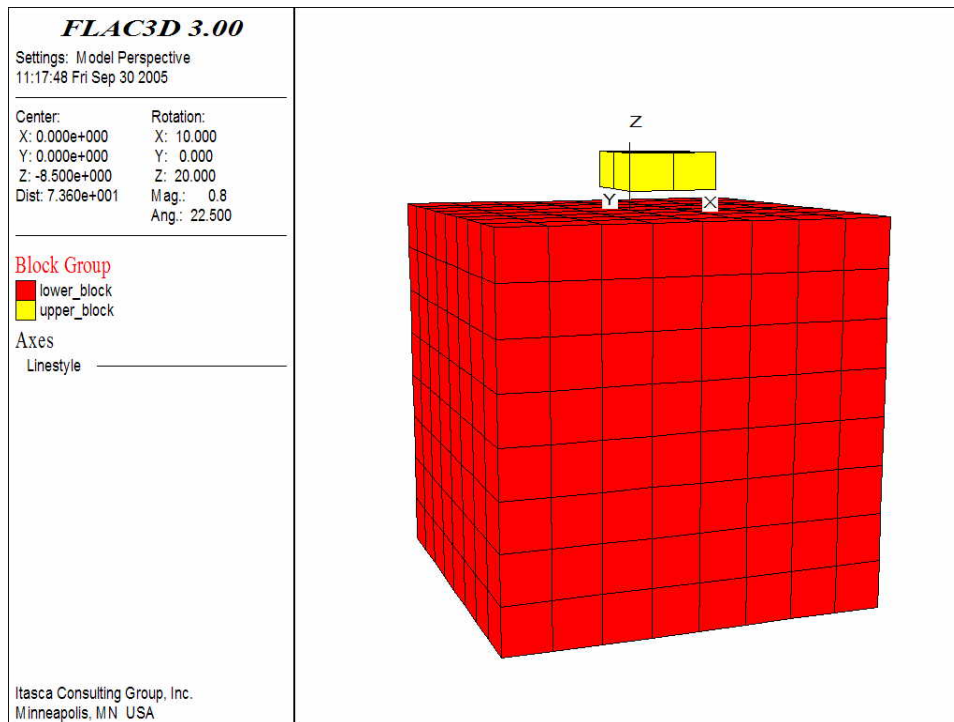


图 3：建立两个处于分离状态的块体材料

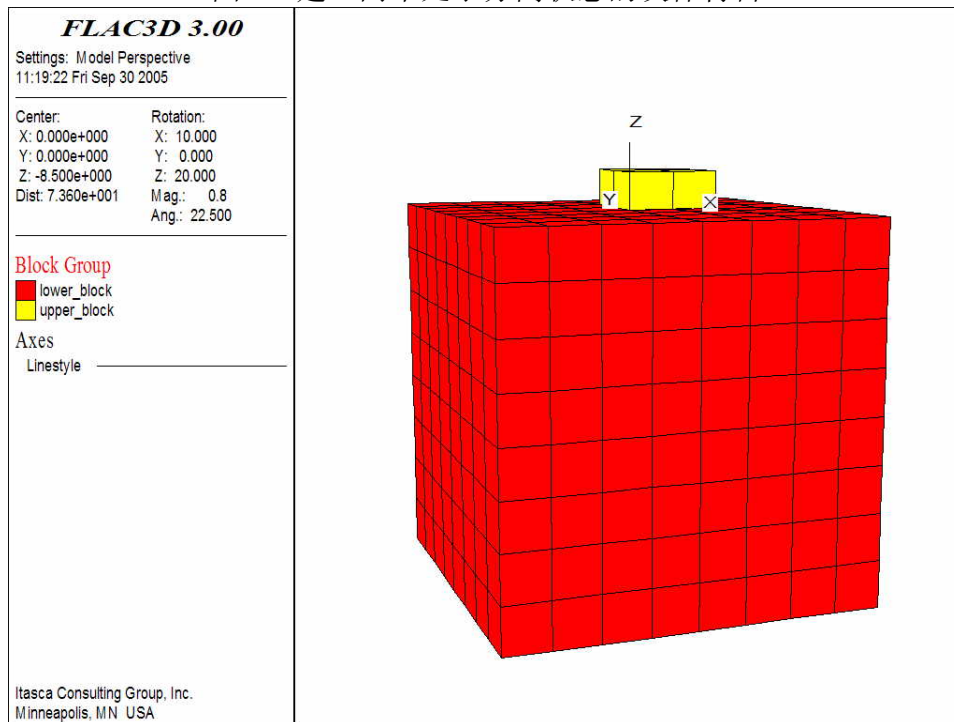


图 4：连接在一起的块体材料

**例 4:**

```
;
plot create grid4
plot add block group red yellow
;把在 group 中的部分染成红色和黄色
plot add axes black
;添加坐标轴线为黑色
;
gen zone brick size 8 8 8 p0 -10,-10,-20 p1 10,-10,-20 &
                        p2 -10,10,-20 p3 -10,-10,0 &
                        group lower_block
```

;建立块体网格，大小为  $8 \times 8 \times 8$ ，其尺寸是  $p_0$ 、 $p_1$ 、 $p_2$ 、 $p_3$  来决定，其中  $p_0$  为起始点的坐标， $p_1$  为  $x$  方向， $p_2$  为  $y$  方向， $p_3$  为  $z$  方向的坐标，并把这些区域建立成一个群，名为 `lower_block`。

```
gen zone brick size 2 ,2,1 p0 0,0,1 p1 5,0,1 &
                        p2 0,5,1 p3 0,0,3 &
                        group upper_block
```

;建立另外一个块体网格，其大小为  $2 \times 2 \times 1$ ，其尺寸及位置是  $p_0$ 、 $p_1$ 、 $p_2$ 、 $p_3$  来决定，具体如上所述，并把这个区域归为一个群，名为 `upper_block`。

```
plot show
;在视图窗中显示
pause
;暂停
ini z add -1 range group upper_block
;群 upper_block 的所有单元，在 z 方向上向下移动 1m
;
;gen zone brick size 2 ,2,1 p0 0,0,0 p1 5,0,0 &
;                        p2 0,5,0 p3 0,0,2 &
;                        group upper_block
```

;重新建立另外一个块体网格，其大小为  $2 \times 2 \times 1$ ，其尺寸及位置是  $p_0$ 、 $p_1$ 、 $p_2$ 、 $p_3$  来决定，具体如上所述，并把这个区域归为一个群，名为 `upper_block`。

```
gen merge 1e-5 range z 0
;此命令是接触面单元合并成一个整体。
```

## FLAC3D 网格生成

### 实例分析 5：界面（interface）的生成

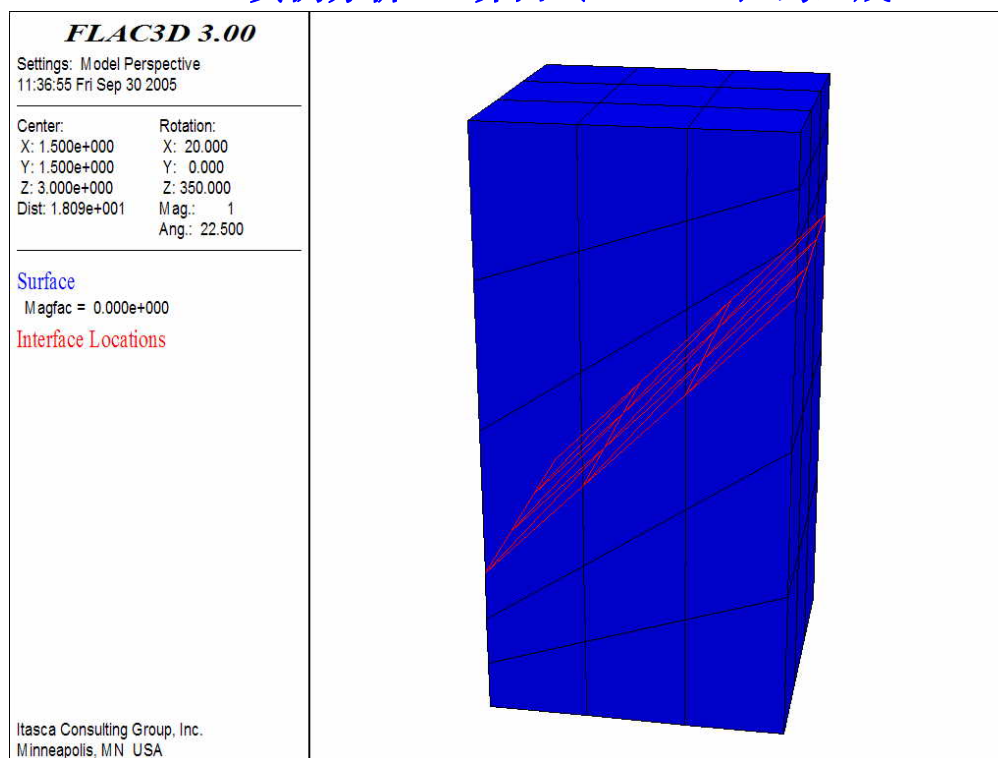


图 5：两个物体之间界面的生成

#### 例 5：

; Create Base

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

;建立另外一个块体网格，其大小为  $3 \times 3 \times 3$ ，其尺寸及位置是 p0、p1、p2、p3、p4、p5、p6 和 p7 等楔型体的 8 个角点坐标来决定，并把这个区域归为一个群，名为 Base（即下面的底座）。

; Create Top - 1 unit high for initial spacing

gen zone brick size 3 3 3 &

p0 (0,0,1.5) p1 (3,0,4.5) p2 (0,3,1.5) p3 (0,0,6) &

p4 (3,3,4.5) p5 (0,3,6) p6 (3,0,6) p7 (3,3,6)

group Top range group Base not

;建立另外一个块体网格，其大小为  $3 \times 3 \times 3$ ，其尺寸及位置是 p0、p1、p2、p3、p4、p5、p6 和 p7 等楔型体的 8 个角点坐标来决定，并把这个区域归为一个群，名为 Top（即上面的部分）。

```
gen separate Top
;使两部分的接触网格分离为两部分

; Create interface elements on the top surface of the base
;在两部分之间添加交界面单元
interface 1 wrap Base Top
;在这两部分之间添加接触单元
;
plot create view_int
;显示，并创建标题 view_int
plot add surface
;显示表面
plot add interface red
;显示交互面为红色
plot show
;打开图形
save int.sav
;形成 sav 文件，并保存为 int.sav
```

## 实例分析 6：隧道的生成

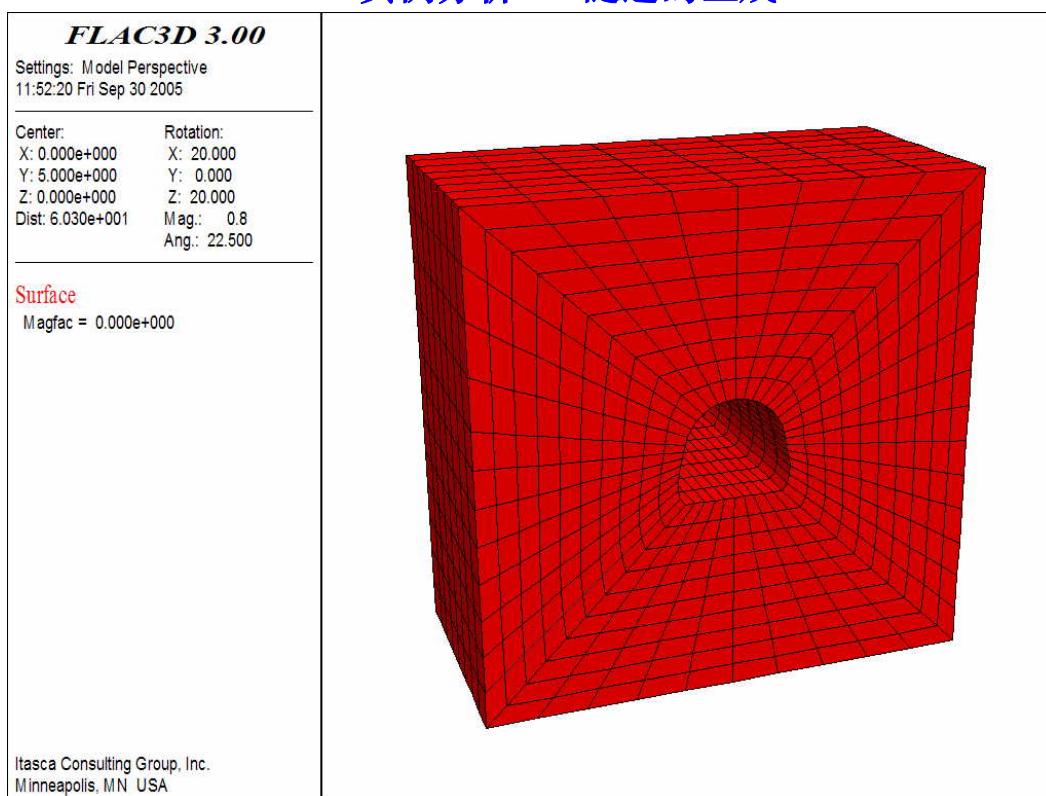


图 6：隧道的图样

## 例 6:

new

;新建

def setup

;定义函数 setup

numy = 8

;定义常量 numy 为 8

depth = 10.0

;depth 为 10

end

;结束对函数的定义

setup

;运行函数 setup

gen zone radcyl size 5 numy 8 10 &amp;

p0 0 0 0 p1 10 0 2 p2 0 depth 0 p3 0 0 10 &amp;

p4 10 depth 2 p5 0 depth 10 p6 10 0 10 p7 10 depth 10

;建立模型的单元为放射性圆柱，其内部圆柱大小尺寸为  $5 \times \text{numy} \times 8$ ，其总体的

;具体的空间位置由 p0,p1,p2,p3,p4,p5,p6,p7（它们的具体意思参看 generate 命令）

;决定。

```
;pause ;  
;暂停  
gen zone radcyl size 5 numy 8 10 &  
  p0 0 0 0 p1 10 0 -10 p2 0 depth 0 p3 10 0 2 &  
  p4 10 depth -10 p5 10 depth 2 p6 10 0 -4 p7 10 depth -4  
;建立模型的单元为放射性圆柱，其内部圆柱大小尺寸为  $5 \times unmy \times 8$ ，其中数值  
;10 的意义是扩展网格的个数。其总体的具体的空间位置由  
;p0,p1,p2,p3,p4,p5,p6,p7（它们的具体意思参看 generate 命令）决定。
```

```
;pause ;  
;暂停  
gen zone brick size 5 numy 10 &  
  p0 0 0 -10 p1 10 0 -10 p2 0 depth -10 p3 0 0 -2 p4 10 depth -10 &  
  p5 0 depth -2 p6 2 0 -2 p7 2 depth -2  
;建立块体单元网格，其大小为  $5 \times unmy \times 10$ ，空间位置由 p0,p1,p2,p3,p4,p5,p6,p7  
;（它们的具体意思参看 generate 命令）决定。  
gen zone reflect orig 0 0 0 norm 1 0 0  
;对单元进行镜像，镜像面为以 (0, 0, 0) 为原点，以 (1, 0, 0) 为法向向量。  
plo crea qqg  
;创建名为 qqg  
plo add surf red  
;表面颜色设置为红色  
plo sho  
;显示图形
```

---



实例分析 7：复杂交叉隧道（NATM）的生成

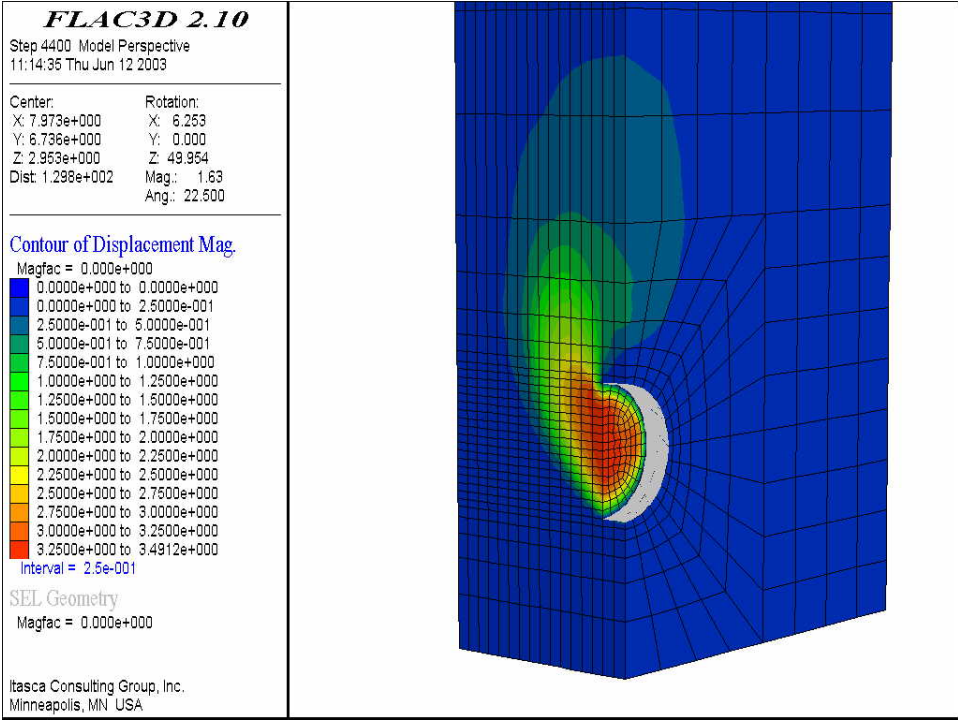


图 7：圆型衬砌在破坏时位移等值线图

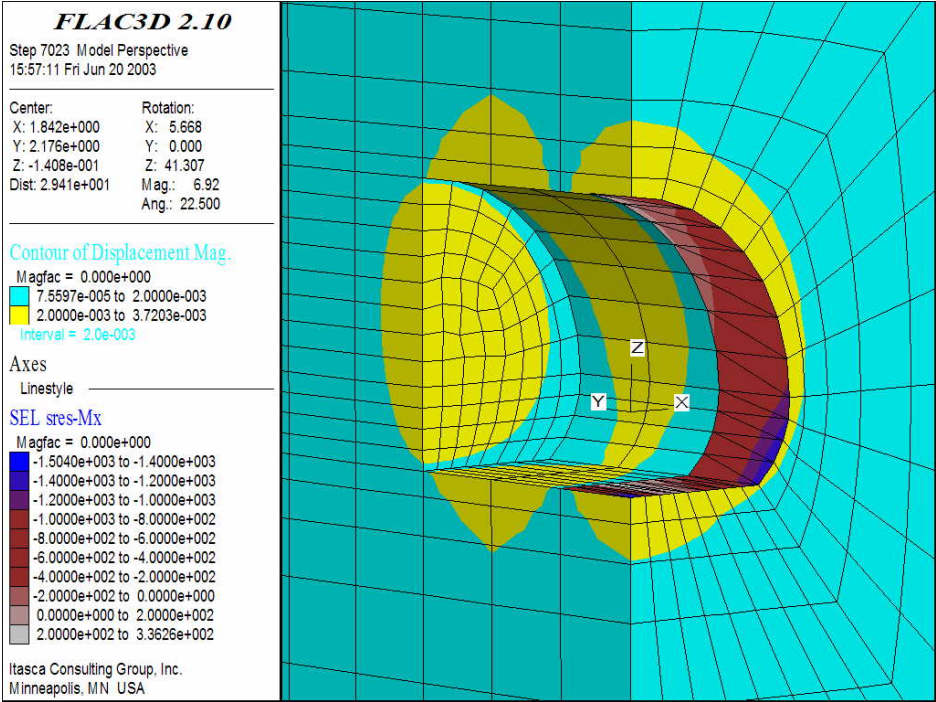


图 8：在部分喷射混凝土下的连续开挖



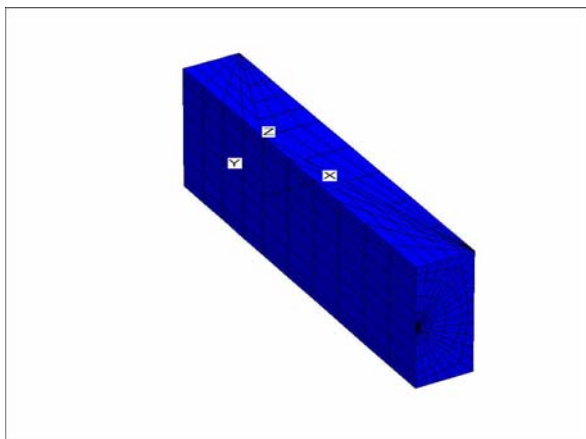


图 9：在交叉点建立放射性网格

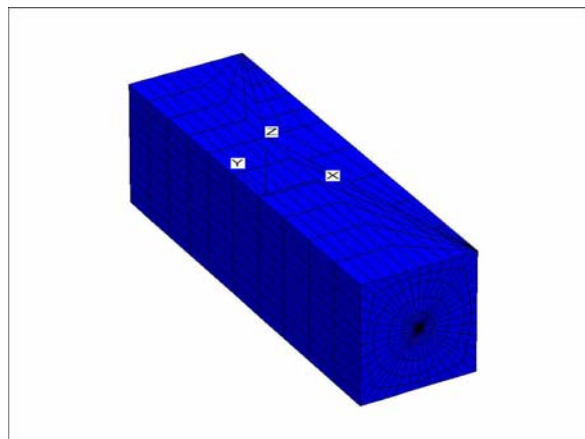


图 10：交叉点的完整隧道

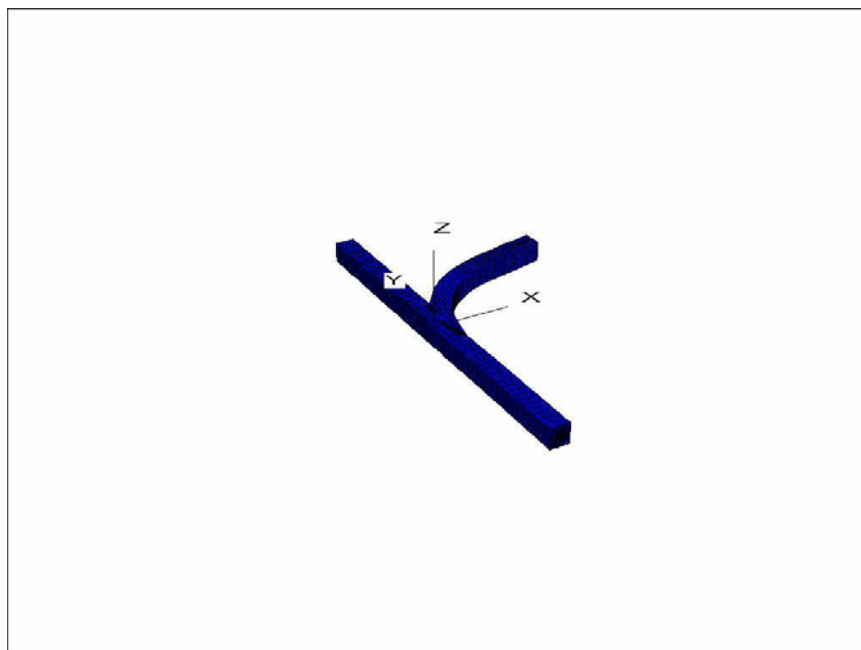


图 11：直型隧道的扩展部分

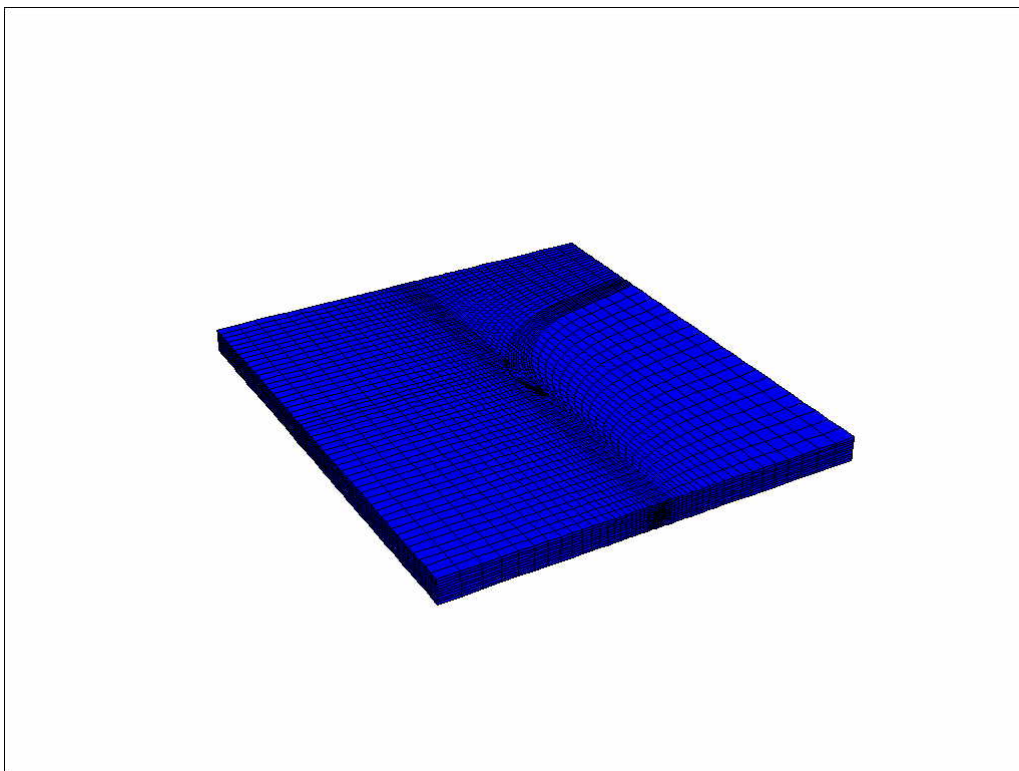


图 12：在隧道平面周围增加的网格

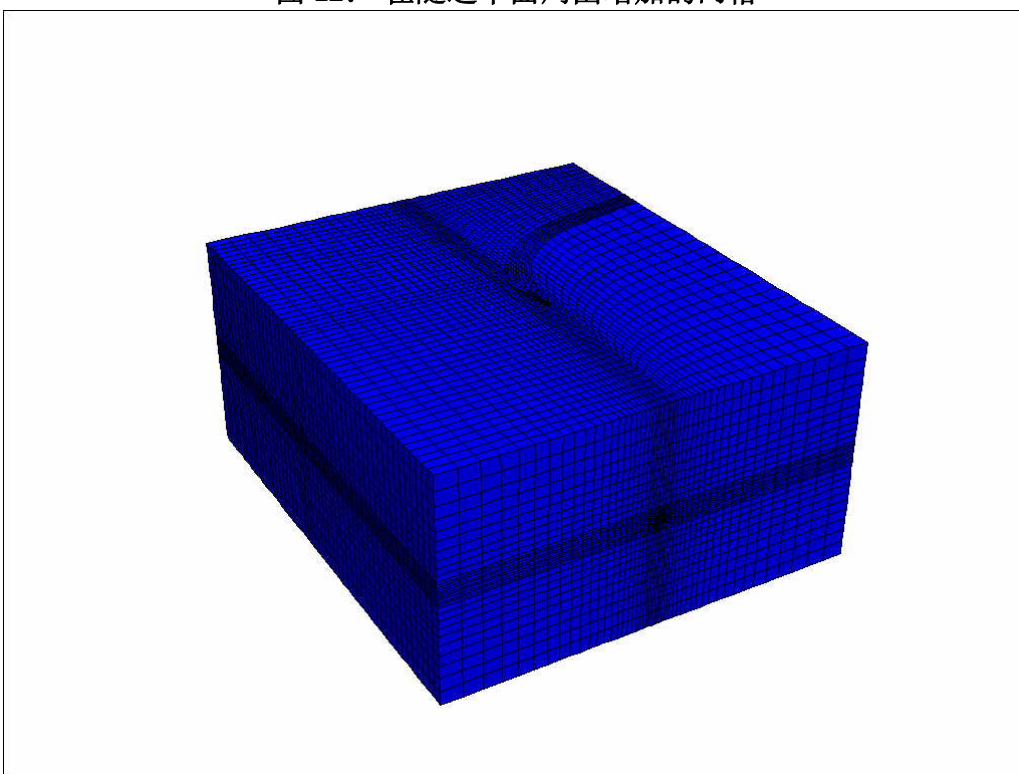


图 13：添加上部边界网格

**例 7：**

```

new
;新建文本
;
def in_def
;用 fish 语言定义函数 in_def（此函数产生的对输入参数进行交互式控制）
  xx = in(msg+'('+default:string(default)+'):')
;对 xx 赋予的值为字符串、数字，它主要用于下面的一些命令中
  if type(xx) = 3
    in_def = default
  else
    in_def = xx
  endif
;条件句的应用，这里的 type（）是用户的 fish 函数决定的返回值。
end
;用于 in_def 函数的结尾，表结束。
def tunnel_data
;fish 定义函数 tunnel_data
  default = 0.0
  msg=' tunnel center x-coordinate (x0) '
  x0 = in_def
;
;交互式输入 x0（隧道中心 x 的坐标）的坐标（输入时，会在命令窗口中产生提示，默认时为 0）
;默认值由上面的 default = 0.0 决定，注意：命令 x0 = in_def 调用了 in_def 函数。
;下面的命令和此处一样。
;
  default = 0.0
  msg=' tunnel minimum y-coordinate (y1) '
  y1 = in_def
;交互式输入 y1（隧道 y 坐标最小值）的坐标（同理，默认时为 0）
;
  default = 0.0
  msg=' tunnel center z-coordinate (z0) '
  z0 = in_def
;交互式输入 z0（隧道中心 z 的坐标）的坐标（同理，默认时为 0）
;
  default = 1.0
  msg=' tunnel radius above tunnel center (rad) '
  rad = in_def
;交互式输入 rad（隧道中心上部的半径）的坐标（同理，默认时为 1）
;
  default = .7
  msg=' tunnel base half-width below tunnel center (base : fails if base >= rad) '
  base = in_def

```

```

;交互式输入隧道中心以下的底部宽度，注意：其值不能大于隧道半径。
;
default = 0.5
msg=' tunnel "height" from base to center (height) '
height = in_def
;交互式输入模型底部到隧道中心的的高度，默认为 0.5
;
default = 2.0
msg=' extent of inner primitive in x-direction (dx: extent = x0-dx to x0+dx) '
dx = in_def
;在 x 方向上对内部的原始坐标进行扩展，分别增加、减少 dx
;
default = 2.0
msg=' extent of inner primitive in z-direction (dz: extent = z0-dz to z0+dz) '
dz = in_def
;在 z 方向上对内部的原始坐标进行扩展，分别增加、减少 dz
;
default = 5.0
msg=' uniform-grid tunnel maximum in y-direction (y2) '
y2 = in_def
;统一网格坐标下，y 坐标方向上的最大值
;
default = 15.0
msg=' radial-grid tunnel maximum in y-direction (y3) '
y3 = in_def
;放射性网格在 y 方向上的最到值
;
dummy=out(' model inner width = '+string(rad+dx))
default = 4.0
msg=' extent of entire tunnel in x-direction from tunnel center (xr: extent = xr-x0) `
xr = in_def + x0
;执行命令 out (s) 是为了在屏幕上显示其后面的信息。string(var)是转化变量
;为字符串。并在 x 方向上从隧道中心扩展整个坐标系
;
dummy=out(' model inner height = '+string(rad+dz))
default = 4.0
msg=' extent of entire tunnel in z-direction above tunnel center (zt: extent = zt-z0) '
zt = in_def + z0
;和上面的命令一样，在 z 方向上从隧道中心扩展整个坐标系
;
default = 4
msg=' number of zones along top of tunnel primitive (nx) '
nx = in_def
;初始隧道顶部的单元数
;

```

```

default = 3
msg=' number of zones in radial direction (nr) '
nr = in_def
;放射性网格在放射方向上的单元个数
;
default = 4
msg=' number of zones in tunnel u-shaped wrap (nu) '
nu = in_def
;u 型隧道单元的数量

default = 4
msg=' number of zones along bottom rightside (nz1 : must be even number) '
nz1 = in_def
if (nz1/2)*2#nz1 then
  nz1=nz1+1
end_if
;隧道底部的单元数量，这里用 if 语句控制 nz1 的输入
;
if xr-x0<1e-3 then
  nx=0
else
  default = 5
  msg=' number of zones on far right region (nxr) '
  nxr = in_def
end_if
;此 if 语句主要是控制隧道上部右侧的单元数量，和网格的合理性
;
if zt-z0<1e-3 then
  nzt=0
else
  default = 5
  msg=' number of zones on far top region (nzt) '
  nzt = in_def
end_if
;此 if 语句主要是控制隧道上部的单元数量和网格的合理性
;
default = 8
msg=' number of zones along y-axis in tunnel primitive (ny1) '
ny1 = in_def
;在 y 轴线上，最初隧道的单元数量
;
if y3-y2<1e-3 then
  ny2=0
else
  default = 4
  msg=' number of zones along y-axis in far region (ny2) '

```

```

ny2 = in_def
;用来修正 y 轴线的单元数量
;
  default = 1.2
  msg=' radial zoning ratio for ny2 zones (raty) '
  raty = in_def
end_if
;y 轴线上单元的依次放大比率
;
  default = 1.5
  msg=' radial zoning ratio for nr zones (ratr) '
  ratr = in_def
;放射性单元的依次放大比率
;
;*** DERIVED PARAMETERS ***
;初始的参数
nz2 = 2*nx ; upper-right and top zones
;此式为隧道上部右边和顶部的关系
ratr2=1/ratr
;Compute x-coordinates
;以下为计算 x 坐标
x1 = x0+rad
x2 = x0+dx
x3 = x0+base
; Compute arc center (x4,z0) to fit arc-points (x1,z0) and (x0+base,z0-height)
;下式是为了使中心坐标(x4,z0)he 弧形处的坐标相适应
x4=x0+(rad*rad-base*base-height*height)/(2.0*(rad-base))
; Compute radius of lower arc
;计算半径的弧度
rad2=rad-(x4-x0)
x7=x0+sqrt(2.0)/2*rad
; compute inner points for radtunnel - adjust based on nx,nu zones
;计算放射性网格点—调整 nx,nu 单元数
x5=(x0*nu+x7*nx)/(nx+nu)
x5c=(nu*x0+nx*x7)/(nx+nu)
x5b=(nu*x0+2*nx*x7)/(2*nx+nu)
x6=(x0*nu+x2*nx)/(nx+nu)
;Compute z coordinates
;计算 z 坐标
z1 = z0+rad
z2 = z0+dz
z4 = z0-dz
z5 = z0-height
z6 = (z0+z4)/2
z8=z0+sqrt(2.0)/2*rad

```

```

z7=(z0*nu+z8*nx)/(nx+nu)
z7c=(nu*z0+nx*z8)/(nx+nu)
z7b=(nu*z0+2*nx*z8)/(2*nx+nu)
;slope constant of lower arc
;光滑下部的弧线
dxz=(x3-x1)/(z5-z0)
;slope constants of upper arc
;光滑上部的弧线
dxz2=(z8-z0)/(x7-x1)
dxz3=(x7-x0)/(z8-z1)
end
;函数 tunnel_data 结束，这一部分主要是参数的输入

def make1
;fish 语言定义的 make1 函数，此函数是为了建立隧道顶部和上边右侧外边的网格
if y1#y2 then
command
; top and upper-right outside
;下面的命令是在 y1 ≠ y2 时，建立隧道的放射性网格
gen zone radcyl size nx ny1 nz2 nr ratio 1 1 1 ratr group 'g1' &
p0 x0 y1 z0 &
p1 x2 y1 z0 &
p2 x0 y2 z0 &
p3 x0 y1 z2 &
p4 x2 y2 z0 &
p5 x0 y2 z2 &
p6 x2 y1 z2 &
p7 x2 y2 z2 &
p8 x1 y1 z0 &
p9 x0 y1 z1 &
p10 x1 y2 z0 &
p11 x0 y2 z1
end_command
end_if
;下面的命令是在 y2 ≠ y3 时，建立隧道的放射性网格
if y2#y3 then
command
gen zone radcyl size nx ny2 nz2 nr ratio 1 raty 1 ratr group 'g1' &
p0 x0 y2 z0 &
p1 x2 y2 z0 &
p2 x0 y3 z0 &
p3 x0 y2 z2 &
p4 x2 y3 z0 &
p5 x0 y3 z2 &
p6 x2 y2 z2 &

```

```

p7 x2 y3 z2 &
p8 x1 y2 z0 &
p9 x0 y2 z1 &
p10 x1 y3 z0 &
p11 x0 y3 z1
end_command
end_if
end
;此函数结束

```

```
def make2
```

;fish 语言定义的 make2 函数，此函数在是为了建立隧道下边右侧外边的网格  
;其它与函数 make1 一样

```

if y1#y2 then
command
; lower-right outside
gen zone radcyl size nx ny1 nz1 nr ratio 1 1 1 ratr group 'g2' &
p0 x4 y1 z0 &
p1 x2 y1 z4 &
p2 x4 y2 z0 &
p3 x2 y1 z0 &
p4 x2 y2 z4 &
p5 x2 y2 z0 &
p6 x2 y1 z6 &
p7 x2 y2 z6 &
p8 x3 y1 z5 &
p9 x1 y1 z0 &
p10 x3 y2 z5 &
p11 x1 y2 z0
end_command
end_if
if y2#y3 then
command
; lower-right outside
gen zone radcyl size nx ny2 nz1 nr ratio 1 raty 1 ratr group 'g2' &
p0 x4 y2 z0 &
p1 x2 y2 z4 &
p2 x4 y3 z0 &
p3 x2 y2 z0 &
p4 x2 y3 z4 &
p5 x2 y3 z0 &
p6 x2 y2 z6 &
p7 x2 y3 z6 &
p8 x3 y2 z5 &
p9 x1 y2 z0 &
p10 x3 y3 z5 &

```



```

                                p11 x1 y3 z0
end_command
end_if
end

def make3
;fish 语言定义的 make3 函数，此函数在是为了建立隧道底部外边的网格
;其它与函数 make1 一样
if y1#y2 then
command
; bottom outside
gen zone brick size nx ny1 nr ratio 1 1 ratr2 group 'g3' &
p0 x0 y1 z4 &
p1 x6 y1 z4 &
                                p2 x0 y2 z4 &
                                p3 x0 y1 z5 &
                                p4 x6 y2 z4 &
p5 x0 y2 z5 &
                                p6 x5 y1 z5 &
                                p7 x5 y2 z5
end_command
end_if
if y2#y3 then
command
; bottom outside
gen zone brick size nx ny2 nr ratio 1 raty ratr2 group 'g3' &
p0 x0 y2 z4 &
p1 x6 y2 z4 &
                                p2 x0 y3 z4 &
                                p3 x0 y2 z5 &
                                p4 x6 y3 z4 &
p5 x0 y3 z5 &
                                p6 x5 y2 z5 &
                                p7 x5 y3 z5
end_command
end_if
end

def make4
;fish 语言定义的 make4 函数，此函数在是为了建立隧道底部右侧外边的网格
;其它与函数 make1 一样
if y1#y2 then
command
; bottom-right outside
gen zone brick size nu ny1 nr ratio 1 1 ratr2 group 'g4' &
p0 x6 y1 z4 &

```

```

                                p1 x2 y1 z4 &
p2 x6 y2 z4 &
                                p3 x5 y1 z5 &
                                p4 x2 y2 z4 &
                                p5 x5 y2 z5 &
                                p6 x3 y1 z5 &
                                p7 x3 y2 z5 &
end_command
end_if
if y2#y3 then
command
; bottom-right outside
gen zone brick size nu ny2 nr ratio 1 raty ratr2 group 'g4' &
p0 x6 y2 z4 &
                                p1 x2 y2 z4 &
p2 x6 y3 z4 &
                                p3 x5 y2 z5 &
                                p4 x2 y3 z4 &
                                p5 x5 y3 z5 &
                                p6 x3 y2 z5 &
                                p7 x3 y3 z5 &
end_command
end_if
end

def make5
;fish 语言定义的 make5 函数，此函数在是为了建立隧道顶部内侧的网格
;其它与函数 make1 一样
if y1#y2 then
command
; top inside
gen zone radtunnel size nx ny1 nx nu group 'g5' &
p0 x0 y1 z0 &
p1 x7 y1 z0 &
p2 x0 y2 x0 &
p3 x0 y1 z8 &
p4 x7 y2 z0 &
p5 x0 y2 z8 &
p6 x7 y1 z8 &
p7 x7 y2 z8 &
p8 x5c y1 z0 &
p9 x0 y1 z7c &
p10 x5c y2 z0 &
p11 x0 y2 z7c &
p12 x5b y1 z7b &
p13 x5b y2 z7b &

```

```

gen zone brick size nx ny1 nx group 'g5' &
p0 x0 y1 z0 &
p1 x5c y1 z0 &
p2 x0 y2 z0 &
p3 x0 y1 z7c &
p4 x5c y2 z0 &
p5 x0 y2 z7c &
p6 x5b y1 z7b &
p7 x5b y2 z7b
end_command
end_if
if y2#y3 then
command
; top inside
gen zone radtunnel size nx ny2 nx nu ratio 1 raty 1 group 'g5' &
p0 x0 y2 z0 &
p1 x7 y2 z0 &
p2 x0 y3 x0 &
p3 x0 y2 z8 &
p4 x7 y3 z0 &
p5 x0 y3 z8 &
p6 x7 y2 z8 &
p7 x7 y3 z8 &
p8 x5c y2 z0 &
p9 x0 y2 z7c &
p10 x5c y3 z0 &
p11 x0 y3 z7c &
p12 x5b y2 z7b &
p13 x5b y3 z7b
gen zone brick size nx ny2 nx ratio 1 raty 1 group 'g5' &
p0 x0 y2 z0 &
p1 x5c y2 z0 &
p2 x0 y3 z0 &
p3 x0 y2 z7c &
p4 x5c y3 z0 &
p5 x0 y3 z7c &
p6 x5b y2 z7b &
p7 x5b y3 z7b
end_command
end_if
end

```

```
def make6
```

;fish 语言定义的 make6 函数，此函数在是为了建立隧道底部内侧的网格

;其它与函数 make1 一样

```
if y1#y2 then
```

```
command
; bottom inside
  gen zone brick size nx ny1 nz1 group 'g6' &
  p0 x0 y1 z5 &
  p1 x5 y1 z5 &
  p2 x0 y2 z5 &
  p3 x0 y1 z0 &
  p4 x5 y2 z5 &
  p5 x0 y2 z0 &
  p6 x5c y1 z0 &
  p7 x5c y2 z0
; bottom-right inside
  gen zone brick size nu ny1 nz1 ratio 1 1 1 group 'g6' &
  p0 x5 y1 z5 &
  p1 x3 y1 z5 &
  p2 x5 y2 z5 &
  p3 x5c y1 z0 &
  p4 x3 y2 z5 &
  p5 x5c y2 z0 &
  p6 x7 y1 z0 &
  p7 x7 y2 z0
end_command
end_if
if y2#y3 then
command
; bottom inside
  gen zone brick size nx ny2 nz1 ratio 1 raty 1 group 'g6' &
  p0 x0 y2 z5 &
  p1 x5 y2 z5 &
  p2 x0 y3 z5 &
  p3 x0 y2 z0 &
  p4 x5 y3 z5 &
  p5 x0 y3 z0 &
  p6 x5c y2 z0 &
  p7 x5c y3 z0
; bottom-right inside
  gen zone brick size nu ny2 nz1 ratio 1 raty 1 group 'g6' &
  p0 x5 y2 z5 &
  p1 x3 y2 z5 &
  p2 x5 y3 z5 &
  p3 x5c y2 z0 &
  p4 x3 y3 z5 &
  p5 x5c y3 z0 &
  p6 x7 y2 z0 &
  p7 x7 y3 z0
end_command
```

```

end_if
end

define make7
;fish 语言定义的 make7 函数，此函数是为了建立隧道核心部分以外的块体的网格
if nxr>0 then
  command
; primitive right of inner primitive - near - lower
  gen zone brick size nxr ny1 nz1 ratio 1.0 1.0 1.0 group 'g7' &
    p0 x2 y1 z4 &
    p1 xr y1 z4 &
    p2 x2 y2 z4 &
    p3 x2 y1 z0
; primitive right of inner primitive - far - lower
  gen zone brick size nxr ny2 nz1 ratio 1.0 raty 1.0 group 'g7' &
    p0 x2 y2 z4 &
    p1 xr y2 z4 &
    p2 x2 y3 z4 &
    p3 x2 y2 z0
; primitive right of inner primitive - near - upper
  gen zone brick size nxr ny1 nx ratio 1.0 1.0 1.0 group 'g7' &
    p0 x2 y1 z0 &
    p1 xr y1 z0 &
    p2 x2 y2 z0 &
    p3 x2 y1 z2
; primitive right of inner primitive - far - upper
  gen zone brick size nxr ny2 nx ratio 1.0 raty 1.0 group 'g7' &
    p0 x2 y2 z0 &
    p1 xr y2 z0 &
    p2 x2 y3 z0 &
    p3 x2 y2 z2
  end_command
end_if
if nzt>0 then
  command
; primitive above inner primitive - near
  gen zone brick size nx ny1 nzt p0 x0,y1,z2 p1 x2,y1,z2 p2 x0,y2,z2 p3 x0 y1 zt &
    ratio 1.0 1.0 1.0 group g7
; primitive above inner primitive - far
  gen zone brick size nx ny2 nzt p0 x0,y2,z2 p1 x2,y2,z2 p2 x0,y3,z2 p3 x0 y2 zt &
    ratio 1.0 raty 1.0 group g7
; primitive above right lower primitive - near
  end_command
if nxr>0 then
  command
    gen zone brick size nxr ny1 nzt p0 x2 y1 z2 p1 xr y1 z2 p2 x2 y2 z2 p3 x2 y1 zt &

```

```

        ratio 1.0 1.0 1.0 group g7
; primitive above right lower primitive - far
    gen zone brick size nxr ny2 nzt p0 x2 y2 z2 p1 xr y2 z2 p2 x2 y3 z2 p3 x2 y2 zt &
        ratio 1.0 raty 1.0 group g7
    end_command
end_if
end_if
end

```

def map1a ; below tunnel center (Correct need for initial points to match)

;此函数是为了使初始化的网格点相互匹配

```
kx=(x3-x0)/(x7-x0)
```

;定义一个常量

```
p_gp=gp_head
```

;把网格点的头指针的地址，赋予变量 p\_gp

```
loop while p_gp#null
```

;loop 为循环命令，其条件是  $p\_gp \neq 0$

; if p\_gp in group 'g6', apply delta-x transformation from line to circle

```
if gp_group(p_gp,1)='g6' then
```

```
xx=gp_xpos(p_gp)-x0
```

;这里的 gp\_xpos(p\_gp)是取这个地址中的网格点的 x 坐标

```
zz=z0-gp_zpos(p_gp)
```

;同理，这里的 gp\_zpos(p\_gp)是去这个地址中的网格点的 z 坐标

```
mu=zz/height
```

```
x37=x3*mu+x7*(1-mu)
```

```
kk=x3/x37
```

```
gp_xpos(p_gp)=x0+xx*kk
```

```
end_if
```

```
p_gp=gp_next(p_gp)
```

;这里的 gp\_next(p\_gp)是取下一个网格点的地址

```
end_loop
```

```
end
```

def map1 ; below tunnel center

```
tanmax=height/(base-(x4-x0))
```

```
maxang=atan(tanmax)
```

;这里，atan(tanmax)为 tanmax 的反正切值

```
kk=maxang/tanmax
```

```
cosmaxang=cos(maxang)
```

;这里，cos(maxang)为 maxang 余弦的值

```
p_gp=gp_head
```

;同理，把网格点的头指针的地址，赋予变量 p\_gp

```
loop while p_gp#null
```

;loop 为循环命令，其条件是  $p\_gp \neq 0$

; if p\_gp in group 'g6', apply delta-r transformation from line to circle

```

if gp_group(p_gp,1)='g6' then
  xx=gp_xpos(p_gp)-x4
  if xx>0 then
    zz=z0-gp_zpos(p_gp)
;这里的 gp_zpos(p_gp)是取这个地址中的网格点的 z 坐标
    zx=zz/xx
    if zx<tanmax then
;这里的 tan 是正切函数
      ang2=kk*zx ; remap from uniform lengths to uniform angles
      rr=xx/cosmaxang ; remap radius from square to sector
      gp_xpos(p_gp)=x4+rr*cos(ang2)
      gp_zpos(p_gp)=z0-rr*sin(ang2)
;这两句话的意思是把后面数值地址放在 gp_xpos(p_gp)和 gp_zpos(p_gp)中
    end_if
  end_if
end_if
p_gp=gp_next(p_gp)
end_loop
end

def map2 ; above tunnel center
;此函数中的意义可以参看上面的例子，这里就不再赘述了
angmax=pi/4
sqrt2=sqrt(2)
cosangmax=cos(angmax)
p_gp=gp_head
loop while p_gp#null
; if p_gp in group 'g5', apply delta-r transformation from line to circle
if gp_group(p_gp,1)='g5' then
  xx=gp_xpos(p_gp)-x0
  zz=gp_zpos(p_gp)-z0
  if xx>zz then
    ang2=angmax*zz/xx ; remap direction from uniform lengths to uniform angles
    rr=xx*sqrt2
    gp_xpos(p_gp)=x0+rr*cos(ang2)
    gp_zpos(p_gp)=z0+rr*sin(ang2)
  else
    if zz>xx then
      ang2=angmax*xx/zz ; remap direction from uniform lengths to uniform angles
      rr=zz*sqrt2
      gp_xpos(p_gp)=x0+rr*sin(ang2)
      gp_zpos(p_gp)=z0+rr*cos(ang2)
    end_if
  end_if
end_if
p_gp=gp_next(p_gp)

```

```
end_loop  
end
```

```
def makeall  
  tunnel_data  
  make1 ; outside top  
  make2 ; outside lower-right  
  make3 ; outside bottom  
  make4 ; outside bottom-right  
  make5 ; inside bottom (call make5 before make6 so common points mapped with map2)  
  make6 ; inside top  
  make7 ; far top and right  
  ; remap inner tunnel walls into outer arcs  
  map1a ; inside bottom (adjust right side to vertical)  
  map1 ; inside bottom (adjust right side to sector)  
  map2 ; inside top  
end
```

```
makeall
```

;此时，执行上面的所有命令

```
gen merge 1e-4  
; gen zone reflect orig x0 y1 z0 norm 0 0 1  
;
```

```
plo crea qqq  
plo add block group  
;plo set perspective off ;easier to see w/o perspective  
plo sho
```

;显示图像，这里用户可以有一些下拉菜单的命令，找出自己想看的内容

```
pause
```

;暂停

```
group 'top heading' range group g5  
group invert      range group g6  
group rock range group 'top heading' not group invert not  
;这里的 group 命令是建立一些群，方便操作。
```

```
pause
```

;暂停

---



## 实例分析 8：模型材料问题

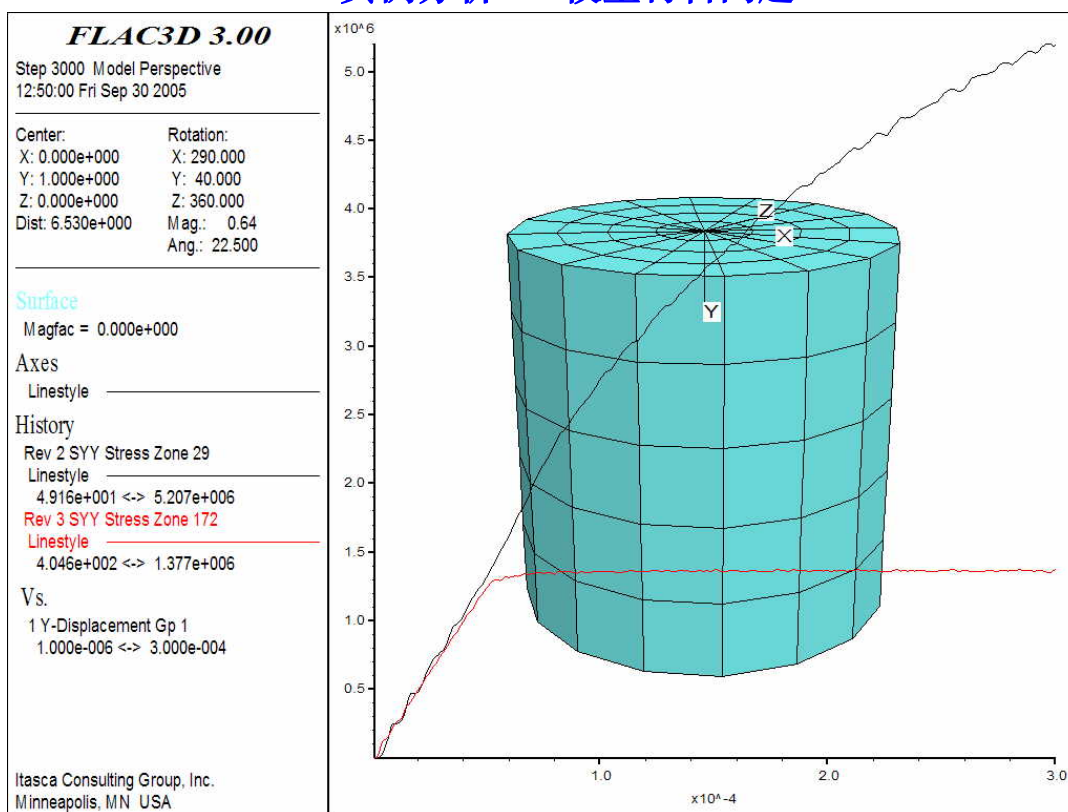


图 14: y 方向的应力曲线

## 例 8：

```
gen zone cyl p0 0 0 0 p1 1 0 0 p2 0 2 0 p3 0 0 1 size 4 5 4
```

;建立圆柱的网格单元，大小为  $4 \times 5 \times 4$

```
gen zone reflect norm 1,0,0
```

;对单元进行镜像，法线为  $(1, 0, 0)$ ，原点为  $(0, 0, 0)$

```
gen zone reflect norm 0,0,1
```

;对以上所有单元再进行镜像，法线为  $(0, 0, 1)$ ，原点为  $(0, 0, 0)$

```
model mohr
```

;采用摩尔-库仑模型

```
prop bulk 1.19e10 shear 1.1e10
```

;为模型赋予参数，体积模量为  $1.19 \times 10^{10}$ ，剪切模量为  $1.1 \times 10^{10}$ 。

```
prop coh 2.72e5 fric 44 ten 2e5
```

;内聚力为  $2.72 \times 10^5$ ，内摩擦角为  $44^\circ$ ，抗拉强的为  $2 \times 10^5$ 。

```
fix x y z range y -.1 .1
```

;边界  $y=0$  为固定支座

```
fix x y z range y 1.9 2.1
```

;边界  $y=2$  为固定支座

```
ini yvel 1e-7 range y -.1 .1
```

;在  $y=0$  的边界上，初始  $y$  方向上的速度为  $1 \times 10^{-7}$

```
ini yvel -1e-7 range y 1.9 2.1
```

---

```
;在 y=2 的边界上，初始 y 方向上的速度为 $-1\times 10^{-7}$   
; ini pp 1e5  
hist gp ydisp 0,0,0  
;监测网格点 (0, 0, 0) 点的 y 方向上的位移  
hist zone syy 0,1,0  
;监测单元 (0, 1, 0) 点的 y 方向上的应力  
hist zone syy 1,1,0  
;监测单元 (1, 1, 0) 点的 y 方向上的应力  
step 3000  
;运行 3000 步
```

---

实例分析 9: Interface 合并 (联结) 问题

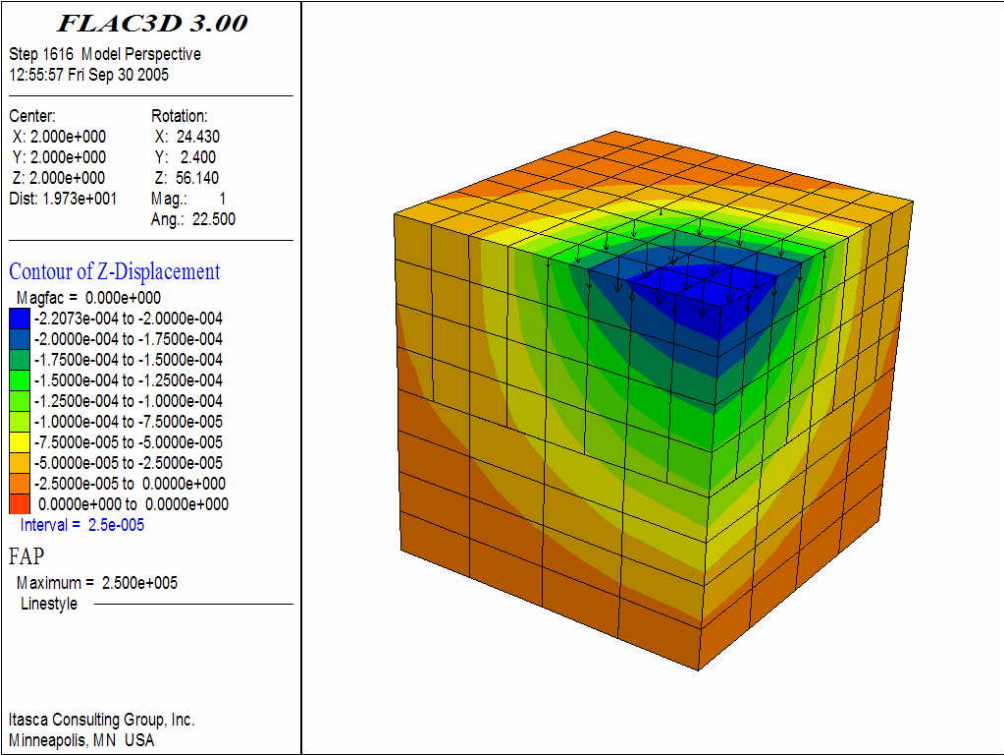


图 15: 运行 1616 步的 z 方向的等值线图

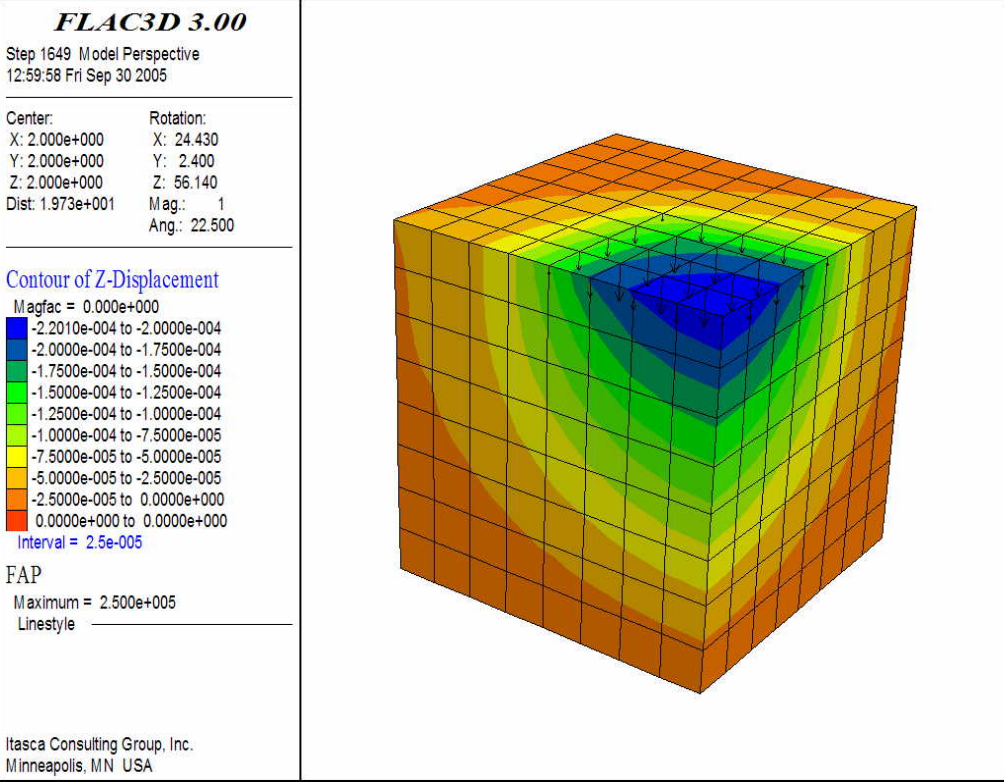


图 16: 运行 1649 步的 z 方向的等值线图

## 例 9

```
new
;新建
; glued interface grid
;粘合交互网格
gen zone brick size 4 4 4 p0 0,0,0 p1 4,0,0 p2 0,4,0 p3 0,0,2
;建立块体网格单元，大小为  $4 \times 4 \times 4$ ，其具体位置由 p0,p1,p2,p3 的坐标决定。
gen zone brick size 8 8 4 p0 0,0,3 p1 4,0,3 p2 0,4,3 p3 0,0,5
;建立块体网格单元，大小为  $8 \times 8 \times 4$ ，其具体位置由 p0,p1,p2,p3 的坐标决定。
inter 1 face range z 2.9,3.1
;在  $z = 3$  的面上建立交互单元
inter 1 prop kn 300e9 ks 300e9 tens 1e10 SBRATIO = 1
;赋予界面参数，法向刚度为  $3.0 \times 10^{11}$ ，剪切刚度为  $3.0 \times 10^{11}$ ，
;抗拉强度为  $1.0 \times 10^{10}$ ，网格的比率为 1
ini z add -1.0 range z 2.9,5.1
;z 方向上在 2.9 到 5.1 的范围内，所有 z 的坐标都减 1
model elas
;材料模型为弹性模型
prop bulk 8e9 shear 5e9
;材料的参数为，体积模量为  $8 \times 10^9$ ，剪切模量为  $5 \times 10^9$ 
fix z range z -.1 .1
;固定边界  $z=0$ ，为滚动支座
fix x range x -.1 .1
;固定边界  $x=0$ ，为滚动支座
fix x range x 3.9 4.1
;固定边界  $x=4$ ，为滚动支座
fix y range y -.1 .1
;固定边界  $y=0$ ，为滚动支座
fix y range y 3.9 4.1
;固定边界  $y=4$ ，为滚动支座
apply szz -1e6 range z 3.9 4.1 x 0,2 y 0,2
;在  $z=4$  的平面中， $x(0, 2)$ ， $y(0, 2)$  的范围内施加  $z$  方向的应力为  $-1 \times 10^6$ 
hist unbal
;监测不平衡力
solve
;计算，一般计算到  $\text{ratio}=1 \times 10^{-5}$  为止
save inter.sav
;将文件保存为 inter.sav
call plot.dat
;调用 plot.dat 文件
pause
;暂停
```

```
new
;新建文档
; attached grid
;连接网格
gen zone brick size 4 4 4 p0 0,0,0 p1 4,0,0 p2 0,4,0 p3 0,0,2
;建立块体的网格单元，其大小为  $4 \times 4 \times 4$ ，其空间位置由 p0,p1,p2,p3 决定
gen zone brick size 8 8 4 p0 0,0,2 p1 4,0,2 p2 0,4,2 p3 0,0,4
;建立块体的网格单元，其大小为  $8 \times 8 \times 4$ ，其空间位置由 p0,p1,p2,p3 决定
attach face range z 1.9 2.1
;在  $z=2$  的平面上连接交界面
model elas
;材料模型为弹性模型
prop bulk 8e9 shear 5e9
;体积模量为  $8 \times 10^9$ ，剪切模量为  $5 \times 10^9$ 
fix z range z -.1 .1
;固定边界  $z=0$ ，为滚动支座
fix x range x -.1 .1
;固定边界  $x=0$ ，为滚动支座
fix x range x 3.9 4.1
;固定边界  $x=4$ ，为滚动支座
fix y range y -.1 .1
;固定边界  $y=0$ ，为滚动支座
fix y range y 3.9 4.1
;固定边界  $y=4$ ，为滚动支座
apply szz -1e6 range z 3.9 4.1 x 0,2 y 0,2
;在  $z=4$  的平面中， $x(0, 2)$ ， $y(0, 2)$  的范围内施加  $z$  方向的应力为  $-1 \times 10^6$ 
hist unbal
;监测不平衡力
Solve
;计算，一般计算到  $\text{ratio}=1 \times 10^{-5}$  为止
save att.sav
;将文件保存为 att.sav
call plot.dat
;调用 plot.dat 文件
pause
;;暂停

new
;新建文本
; one grid
;一个网格
gen zone brick size 8 8 8 p0 0,0,0 p1 4,0,0 p2 0,4,0 p3 0,0,4
;建立块体的网格单元，其大小为  $8 \times 8 \times 8$ ，其空间位置由 p0,p1,p2,p3 决定
model elas
```

---

```
;材料模型为弹性模型
prop bulk 8e9 shear 5e9
;材料的体积模量为  $8 \times 10^9$ ，剪切模量为  $5 \times 10^9$ 
fix z range z -.1 .1
;固定边界  $z=0$ ，为滚动支座
fix x range x -.1 .1
;固定边界  $x=0$ ，为滚动支座
fix x range x 3.9 4.1
;固定边界  $x=4$ ，为滚动支座
fix y range y -.1 .1
;固定边界  $y=0$ ，为滚动支座
fix y range y 3.9 4.1
;固定边界  $y=4$ ，为滚动支座
apply szz -1e6 range z 3.9 4.1 x 0,2 y 0,2
;在  $z=4$  的平面中， $x$  (0, 2)， $y$  (0, 2) 的范围内施加  $z$  方向的应力为  $-1 \times 10^6$ 
hist unbal
;监测不平衡力
solve
;计算，一般计算到  $\text{ratio}=1 \times 10^{-6}$  为止
save noatt.sav
;将文件保存为 noatt.sav
call plot.dat
;调用 plot.dat 文件
```

---

## 实例分析 10：初始条件问题

APPLY 命令可以在其后面选用关键词 **gradient**，这时，我们就可以在一个指定的范围内线性地改变应力或应变。在关键词 **gradient** 后面的参数为  $gx$ ， $gy$ ， $gz$ ，它们可以用来说明应力或者应变在  $x$ ， $y$ ， $z$  上的变化趋势。应力或者应变线性改变的距离，是以全局坐标系下的原点  $(0, 0, 0)$  为准的。例如：

$$S = S^{(0)} + g_x x + g_y y + g_z z \quad (3.1)$$

在此公式中， $S^{(0)}$  的数值是全局坐标下的值，其原点坐标为  $(0, 0, 0)$ ， $g_x$ 、 $g_y$ 、 $g_z$  是沿  $x$ ， $y$ ， $z$  上的变化趋势值。

对于这个公式的理解，我们最好用下面的这得例子加以解释：

`apply sxx -10e6 gradient 0,0,1e5 range z -100,0`

在这个求解  $\sigma_{xx}$  的方程中， $z$  为变量，所以  $\sigma_{xx}$  为：

$$\sigma_{xx} = -10 \times 10^6 + (10^5)z$$

这里， $\sigma_{xx}$  在原点  $(0, 0, 0)$  的应力为  $\sigma_{xx} = -10 \times 10^6$ ，而在  $z = -100$  处， $\sigma_{xx} = -20 \times 10^6$ 。在这两点之间时，应力的大小随这一点距原点坐标的远近而线性地改变。

特别指出，我们可以应用此命令来模拟由重力引起的，随深度而有所增加的应力变化。我们在说明 **INITIAL** 命令时，确定应力场是随坐标而产生坡度变化的，并且重力加速度的值已有 **SET gravity** 命令给出。



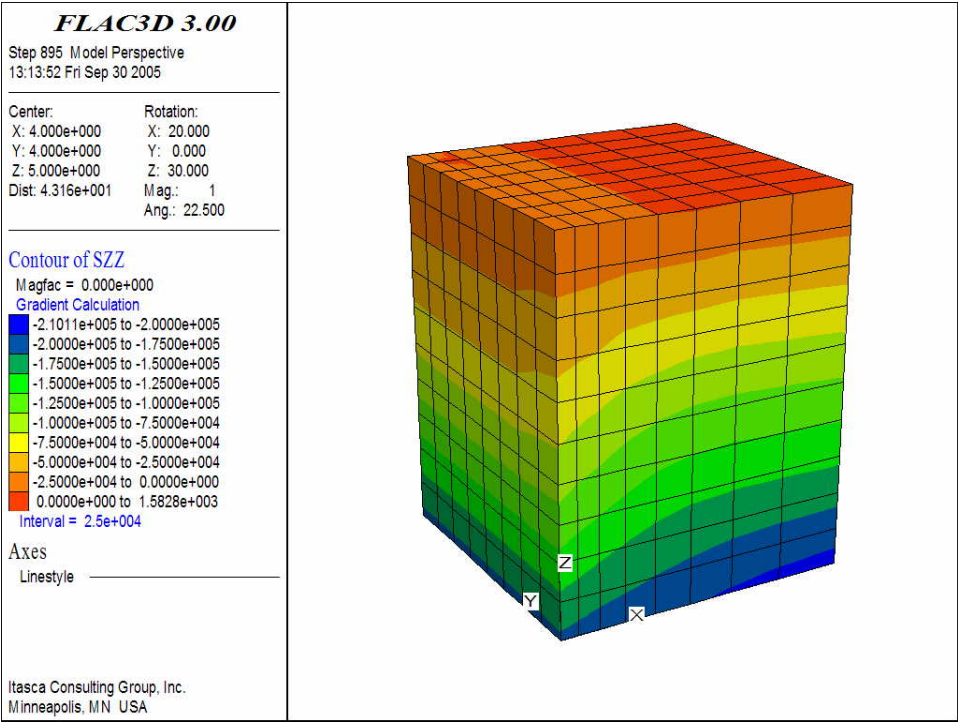


图 17：在初始阶段 z 放向应力的等值线图

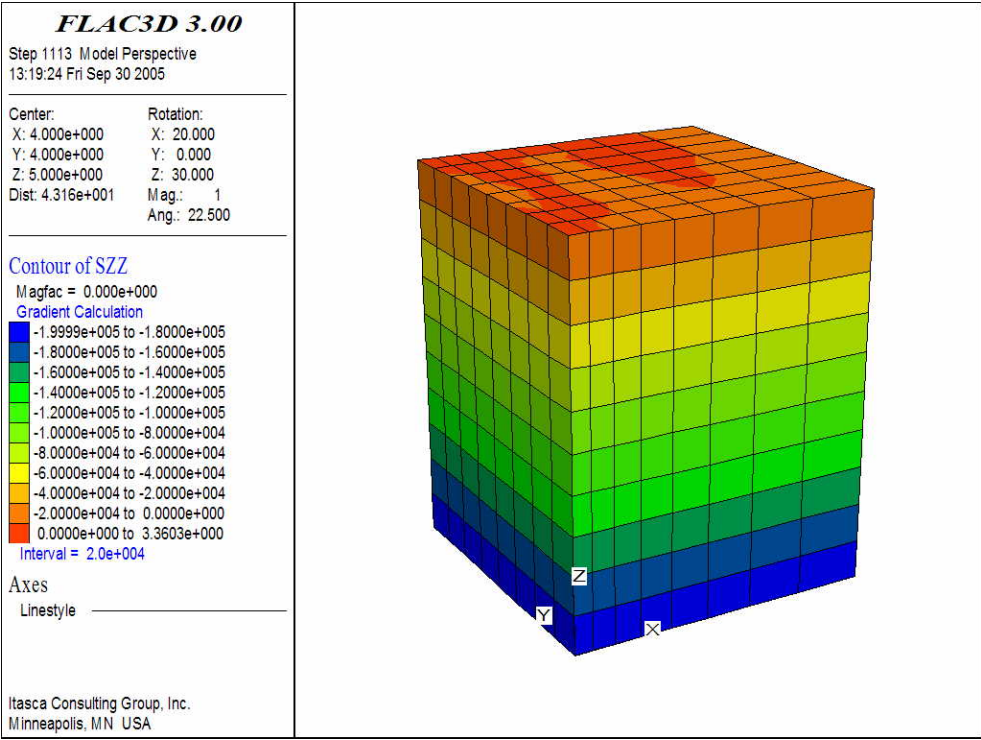


图 18：在运行 1113 步后 z 放向应力的等值线图

例 10：

gen zone brick size 8 8 10 ratio 1.2 1 1



```
;产生块体的网格单元，其到小为  $8 \times 8 \times 10$ ，在 x, y, z 方向上的放大比率依  
;次为 1.2, 1, 1  
model mohr  
;材料模型为摩尔-库仑模型  
ini dens 2000  
;所有材料的初始密度为 2000  
prop bulk 2e8 shear 1e8  
;材料的体积模量为  $2 \times 10^8$ ，剪切模量为  $1 \times 10^8$ 。  
prop fric 30  
;材料的内摩擦角为  $30^\circ$ 。  
fix x range x -.1 .1  
;固定边界  $x=0$ ，为滚动支座  
fix x range x 7.9 8.1  
;固定边界  $x=8$ ，为滚动支座  
fix y range y -.1 .1  
;固定边界  $y=0$ ，为滚动支座  
fix y range y 7.9 8.1  
;固定边界  $y=8$ ，为滚动支座  
fix z range z -.1 .1  
;固定边界  $z=0$ ，为滚动支座  
set grav 10  
;设置重力加速度为 10  
solve  
;运算  
save ini1.sav  
;形成 sav 文件，保存为 ini1.sav  
plot create szz_contour  
;创建文件名为 szz_contour (z 方向的应力云图)  
plot set cent 4 4 5  
;视图的中心坐标为 (4, 4, 5)  
plot set rot 20 0 30  
;视图的旋转角度为 (20, 0, 30)  
plot set mag 1.0  
;视图的放大倍数为 1.0  
plot add cont szz out on shade on  
;打开 szz 的等值线图  
plot add axes  
;打开坐标轴线  
plot show  
;显示窗口  
pause  
;暂停  
new
```

```
;新建文本
gen zone brick size 8 8 10 ratio 1.2 1 1
;产生块体的网格单元，其到小为 $8\times 8\times 10$ ，在 x, y, z 方向上的放大比率依
;次为 1.2, 1, 1
model mohr
;材料模型为摩尔-库仑模型
ini dens 2000
;所有材料的初始密度为 2000
prop bulk 2e8 shear 1e8
;材料的体积模量为 $2\times 10^8$ ，剪切模量为 $1\times 10^8$ 。
prop fric 30
;材料的内摩擦角为 $30^\circ$ 。
fix x range x -.1 .1
;固定边界 x=0，为滚动支座
fix x range x 7.9 8.1
;固定边界 x=8，为滚动支座
fix y range y -.1 .1
;固定边界 y=0，为滚动支座
fix y range y 7.9 8.1
;固定边界 y=8，为滚动支座
fix z range z -.1 .1
;固定边界 z=0，为滚动支座
ini szz -2.0e5 grad 0,0,2e4
;施加初始应力，因为 $\sigma_{zz} = \sigma_{zz}^0 + gx \times x + gy \times y + gz \times z$ ，所以，在此命
;令下 $\sigma_{zz} = -2.0 \times 10^5 + 2 \times 10^4 \times z$ 
ini sxx -1.5e5 grad 0,0,1.5e4
;同理， $\sigma_{xx} = -1.5 \times 10^5 + 1.5 \times 10^4 \times z$ 
ini syy -1.5e5 grad 0,0,1.5e4
;同理， $\sigma_{yy} = -1.5 \times 10^5 + 1.5 \times 10^4 \times z$ 
set grav 10
;设置重力加速度为 10
solve
;运算
save ini2.sav
;形成 sav 文件，保存为 ini2.sav
plot create szz_contour
;创建文件名为 szz_contour (z 方向的应力云图)
plot set cent 4 4 5
;视图的中心坐标为 (4, 4, 5)
plot set rot 20 0 30
;视图的旋转角度为 (20, 0, 30)
plot set mag 1.0
```

```
;视图的放大倍数为 1.0
plot add cont szz out on shade on
;打开 szz 的云图
plot add axes
;打开坐标轴线
plot show
;显示窗口
pause
;暂停

new
;新建文本
gen zone brick size 8 8 10 ratio 1.2 1 1
;产生块体的网格单元，其到小为  $8 \times 8 \times 10$ ，在 x, y, z 方向上的放大比率依
;次为 1.2，1，1
model mohr
;材料模型为摩尔-库仑模型
ini dens 2000
;所有材料的初始密度为 2000
prop bulk 2e8 shear 1e8
;材料的体积模量为  $2 \times 10^8$ ，剪切模量为  $1 \times 10^8$ 。
prop fric 30
;材料的内摩擦角为  $30^\circ$ 。
fix x range x -.1 .1
;固定边界  $x=0$ ，为滚动支座
fix x range x 7.9 8.1
;固定边界  $x=8$ ，为滚动支座
fix y range y -.1 .1
;固定边界  $y=0$ ，为滚动支座
fix y range y 7.9 8.1
;固定边界  $y=8$ ，为滚动支座
fix z range z -.1 .1
;固定边界  $z=0$ ，为滚动支座
set grav 10
;设置重力加速度为 10
solve elastic
;进行弹性计算
save ini3.sav
;形成 sav 文件，保存为 ini3.sav
plot create szz_contour
;创建文件名为 szz_contour (z 方向的应力云图)
plot set cent 4 4 5
;视图的中心坐标为 (4, 4, 5)
plot set rot 20 0 30
```

```
;视图的旋转角度为 (20, 0, 30)
plot set mag 1.0
;视图的放大倍数为 1.0
plot add cont szz out on shade on
;打开 szz 的云图
plot add axes
;打开坐标轴线
plot show
;显示窗口
```

---

实例分析 11：破坏问题分析

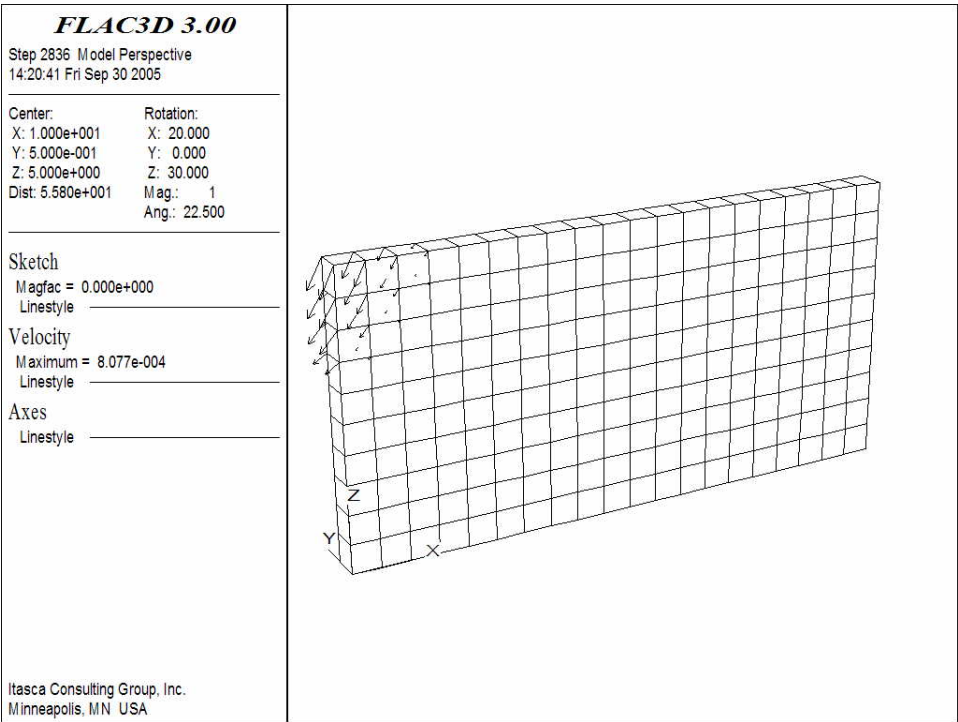


图 19：土体的速度矢量

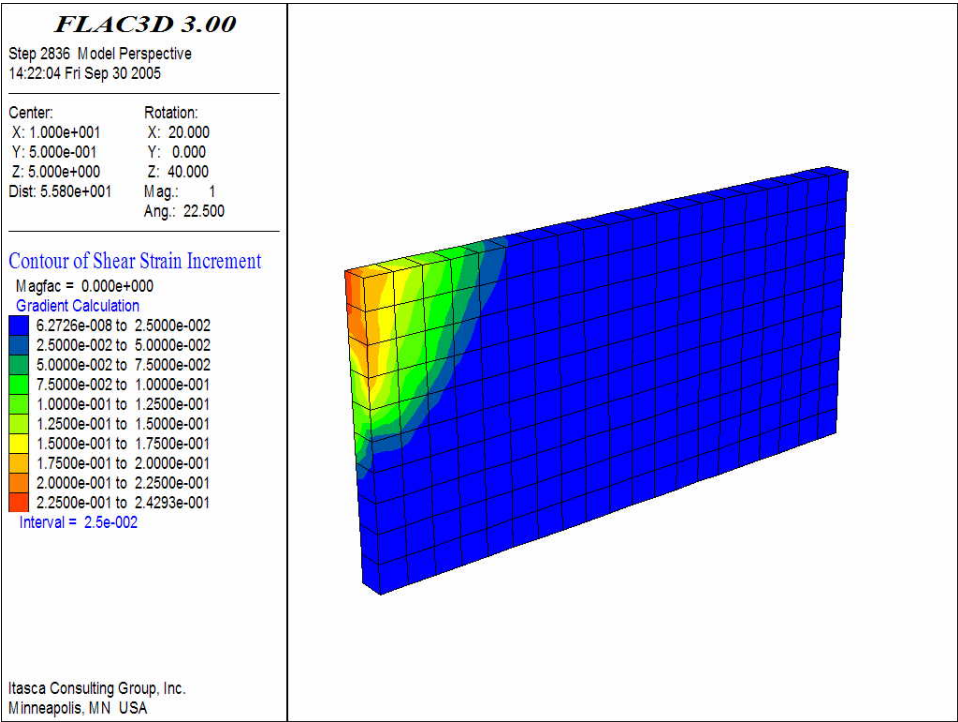


图 20：土体的剪切应力等值线图

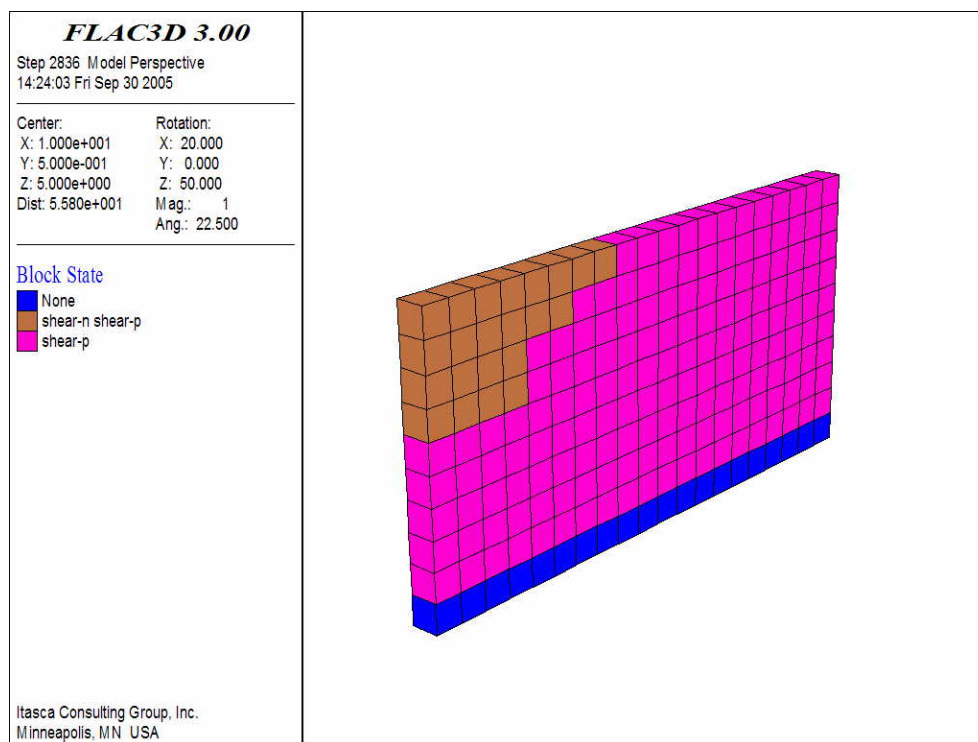


图 21：土体的塑性状态

**例 11：**

```

gen zone brick size 20 1 10
;建立块体网格，其大小为 20×1×10。
model mohr
;材料模型为摩尔-库仑模型
prop dens 2000 bul 2.e8 shea 1.e8 cohesion 0.0
;材料的性质： 密度为 2000， 体积模量为  $2.0 \times 10^8$ ， 剪切模量为  $1.0 \times 10^8$ ，
;粘聚力为 0
prop friction 30. dilation 0. tension 0
;内摩擦角为  $30^\circ$ ， 剪胀角为 0， 抗拉强度为 0
fix x range x -.1 .1
;固定边界  $x=0$ ， 为滚动支座
fix x range x 19.9 20.1
;固定边界  $x=20$ ， 为滚动支座
fix x y z range z -.1 .1
;固定边界  $z=0$ ， 为固端支座
fix y
;固定所有单元 y 方向的运动
set gravity 0 0 -10
;设置重力加速度为 (0, 0, -10)
hist unbal

```

```
;监测不平衡力
solve
;运算
save fail1.sav
;保存为 fail1.sav
free x range x -.1 .1 z 6.9 10.1
;放松 x=0 平面上，z=7，10 这一部分在 x 方向的约束
step 2000
;运算 2000 个时步
save fail2.sav
;保存为 fail2.sav
ret
;放在批处理文件的最后，以返回  $FLAC^{3D}$  的控制状态
```

---