5 Spherical Cavity in an Infinite Elastic Medium

5.1 Problem Statement

Stresses and displacements are determined for the case of a spherical cavity in an infinite elastic medium subjected to isotropic in-situ stresses. This problem provides a more rigorous test of the axisymmetry logic in FLAC than the cylindrical hole in an elastic medium does (see Section 1).

The elastic material surrounding the cavity is assumed to have the following properties:

- density ($\rho$) 2500 kg/m$^3$
- shear modulus ($G$) 2.9 GPa
- bulk modulus ($K$) 3.9 GPa

The cavity has a radius of 1 m. The in-situ stress state is $-30$ MPa (tension positive).

5.2 Closed-Form Solution

The radial displacement around a spherical cavity in an infinite elastic body under an isotropic stress field is given by Goodman (1980, p. 220):

$$u_r = -\frac{P_o a^3}{4r^2 G}$$  \hspace{1cm} (5.1)

where $P_o$ is the external pressure, $a$ is the spherical hole radius, and $G$ is the shear modulus of the body.

Timoshenko and Goodier (1970, p. 395) provide a solution for the stress field in a hollow spherical container subjected to internal and external pressure:

$$\sigma_r = \frac{P_o b^3 (r^3 - a^3)}{r^3 (a^3 - b^3)} + \frac{P_i a^3 (b^3 - r^3)}{r^3 (a^3 - b^3)}$$

$$\sigma_\theta = \frac{P_o b^3 (2r^3 + a^3)}{2r^3 (a^3 - b^3)} - \frac{P_i a^3 (2r^3 + b^3)}{2r^3 (a^3 - b^3)}$$  \hspace{1cm} (5.2)

where $P_i$ is the internal pressure, and $b$ is the outside radius of the container.
The solution to the problem of a spherical cavity in an infinite medium is determined by setting \( P_i = 0 \), and finding the limit as \( b \) approaches infinity. Normalizing with the in-situ stress value \( (P_o) \), the final solution is:

\[
\frac{\sigma_r}{P_o} = -\frac{r^3 - a^3}{r^3} \\
\frac{\sigma_\theta}{P_o} = -\frac{a^3 + 2r^3}{2r^3}
\] (5.3)

**5.3 FLAC Model**

Figure 5.1 shows the model used for the ***FLAC*** analysis. The model grid is identical to that used for the model of the cylindrical hole in Section 1. The ***FISH*** function “HOLE.FIS” is again used to generate the grid (see Section 3 in the ***FISH*** volume). By specifying **CONFIG axi** (i.e., stipulating axisymmetric geometry) with this grid, the model represents a spherical top-half section of the cavity. The grid is shown in Figures 5.2 and 5.3. The in-situ stress state is applied to the model first; then the cavity is made.

---

**Figure 5.1**   ***FLAC*** model for a spherical cavity in an infinite elastic medium
**Figure 5.2** FLAC zone geometry

**Figure 5.3** FLAC zone geometry in region around cavity
5.4 Discussion and Results

Figures 5.4 and 5.5 show a direct comparison between FLAC and the analytical solution for radial and tangential stresses and radial displacements. The plots compare normalized stresses, $-\sigma_r/P_o$ and $-\sigma_\theta/P_o$, and normalized displacement, $-u_r/r$, versus normalized radius, $r/a$. These figures are created with the FISH function “TABM5.FIS.” FISH functions (in file “SPHERE.DAT”) calculate the stress and displacement error throughout the model. The functions are similar to those used in Section 1. Figures 5.6 through 5.8 show contours of errors in tangential stress, radial stress and radial displacement, respectively.

The average error over the entire model was also calculated, with a result of 0.23% for tangential stress, 0.22% for radial stress and 0.94% for radial displacement.
Figure 5.5  Displacement comparison – spherical cavity

Figure 5.6  Error distributions for tangential stress
**Figure 5.7** Error distributions for radial stress

**Figure 5.8** Error distributions for radial displacement
5.5 References


5.6 Data File “SPHERE.DAT”

;Project Record Tree export

;... STATE: M5 ....
config ax extra=8
el 30 30
model
call hole.fis
;
set rmin=1 rmul=10 gratio=1.1
hole
prop shear=2.9e9 bulk=3.9e9 dens=2500
ini sxx=-30e6 syy=-30e6 szz=-30e6
fix y j 1
fix x j 31
app sxx -30e6 syy -30e6 i 31
hist unbal
hist syy i 1 j 1
hist sxx i 1 j 1
hist szz i 1 j 1
solve
save m5.sav

;... STATE: M5_V ....
;set echo off
;****************** define the constants *****************
def parm
p0=30e6
bm=bulk_mod(1,1)
sm=shear_mod(1,1)
nu=(3.0*bm-2.0*sm)/(6.0*bm+2.0*sm)
end
;
set rmin = 1.0
parm
;
;******** calculate the theoretical results ***********

; the theoretical results are stored in the following arrays
; tangential stress ........... EX_1
; radial stress ............... EX_2
; x displacements ............ EX_3
; y displacements ............ EX_4
; displacements magnitude .... EX_5
;
def theor
Spherical Cavity in an Infinite Elastic Medium

loop i (1, izones)
  loop j (1, jzones)
    xc = .25 * (x(i, j) + x(i, j+1) + x(i+1, j+1) + x(i+1, j))
    yc = .25 * (y(i, j) + y(i, j+1) + y(i+1, j+1) + y(i+1, j))
    rz = sqrt(xc^2 + yc^2)
    if rz # 0 then
      ex_1(i, j) = -((rmin^3) + (2*(rz^3)))*p0/(2*(rz^3))
      ex_2(i, j) = -((rz^3) - (rmin^3))*p0/(rz^3)
    else
      ex_1(i, j) = 0
      ex_2(i, j) = 0
    end_if
  end_loop
end_loop
loop i (1, igp)
  loop j (1, jgp)
    ro = sqrt(x(i, j)^2 + y(i, j)^2)
    if ro # 0 then
      dd = (p0*(rmin^3))/(4*(ro^2)*sm)
      ex_3(i, j) = -dd*x(i, j)/ro
      ex_4(i, j) = -dd*y(i, j)/ro
      ex_5(i, j) = -dd
    end_if
  end_loop
end_loop
end

; theor
;
; ***************** evaluate the error in stresses *****************
; the errors in stress calculations are evaluated
; for each zone and are stored in the following arrays:
;   ; the total average errors are calculated and stored in:
;   ; average ; average ;
def evals
  ert = 0
  err = 0
loop i (1, izones)
  loop j (1, jzones)
    temp1 = .5*(sxx(i, j) + syy(i, j))
    temp2 = sqrt(sxy(i, j)^2 + .25*(sxx(i, j) - syy(i, j))^2)
    stm = temp1 - temp2
    ex_6(i, j) = 100*(stm - ex_1(i, j))/p0
    ert = ert + ex_6(i, j)^2
    srm = temp1 + temp2
    ex_7(i, j) = 100*(srm - ex_2(i, j))/p0
  end_loop
end_loop
err=err+ex_7(i,j)^2
end_loop
end_loop
ert=sqrt(ert/(izones*jzones))
err=sqrt(err/(izones*jzones))
end

evals

;*************** evaluate the error in displacements ***************
; the errors in displacement calculations are evaluated
; for each zone and are stored in the following array:
; average

def evald
erd=0
loop i (1,igp)
  loop j (1,jgp)
    temp3=100*sqrt((xdisp(i,j)-ex_3(i,j))^2+(ydisp(i,j)-ex_4(i,j))^2)
    if ex_5(1,1)≠0 then
      ex_8(i,j)=temp3/ex_5(1,1)
    else
      ex_8(i,j)=0.0
    end_if
    erd=erd+ex_8(i,j)
  end_loop
end_loop
erd=erd/(igp*jgp)
end

evals

;****************** create plots *******************************
scline 1 1 .1 10 .1
scline 2 1 1 15 15
catabm5.fis
save m5_v.sav

;*** plot commands ****
;plot name: grid
plot hold grid blue
;plot name: Stress comparison
label table 10
tangential stress (anal)
label table 11
tangential stress (FLAC)
Spherical Cavity in an Infinite Elastic Medium

```
label table 20
radial stress (anal)
label table 21
radial stress (FLAC)
plot hold table 21 cross 20 line 11 cross 10 line
;plot name: Displacement comparison
label table 30
radial disp. (anal)
label table 31
radial disp. (FLAC)
plot hold table 31 cross 30 line label 30 red label 31 red
;plot name: Error distribution for tangential stress
plot hold bound ex_6 zone fill
;plot name: Error distribution for radial stress
plot hold bound ex_7 zone fill
;plot name: Error distributions for radial displacement
plot hold bound ex_8 fill
```
5.7 Data File “TABM5.DAT”

```plaintext
def tabm
    loop i(1,izones)
        j = 1
        xc = 0.25*(x(i,j) + x(i,j+1) + x(i+1,j+1) + x(i+1,j))
        yc = 0.25*(y(i,j) + y(i,j+1) + y(i+1,j+1) + y(i+1,j))
        rz = sqrt(xc^2 + yc^2)
        temp1 = 0.5*(sxx(i,j) + syy(i,j))
        temp2 = sqrt((sxy(i,j)^2 + 0.25*(sxx(i,j) - syy(i,j))^2)
        stm = temp1 - temp2
        srm = temp1 + temp2
        xtable(10,i) = rz
        ytable(10,i) = -ex_1(i,1)/p0
        xtable(11,i) = rz
        ytable(11,i) = -stm/p0
        xtable(20,i) = rz
        ytable(20,i) = -ex_2(i,1)/p0
        xtable(21,i) = rz
        ytable(21,i) = -srm/p0
    end_loop
    loop i (1,igp)
        j = 1
        rg = sqrt(x(i,j)^2 + y(i,j)^2)/rmin
        temp1 = sqrt(xdisp(i,j)^2 + ydisp(i,j)^2)/rmin
        xtable(30,i) = rg
        ytable(30,i) = -ex_5(i,j)
        xtable(31,i) = rg
        ytable(31,i) = temp1
    end_loop
end
```

plot hold table 10 line 20 line 11 21
plot hold table 30 line 31
ret

FLAC Version 6.0