**Supplementary Material**

**1. An example of cavity length in the steady state based on a FORTRAN module** (The initial effective pressure *Pe*＝0.4 MPa and *Us*＝500 m yr-1. *L(i)*＝*L(ti), L(i+1)*＝*L(ti+1), h(i)*＝*h(ti), h(i+1)=h(ti+1)*)

do *i*＝1,30000 (The number 30000 can be change, in order to guarantee *h(k)*=AC)

Assume a value *R*.

(The size of every step is 100s.)

end do

When *h(i)*＝AC, then calculate a new value *R1* by Eqns (8)

Compare *R1* to *R*, if , then *L(k)* is the cavity length in the steady state, otherwise, assume a new value *R2* between *R* and *R1*, repeat the previous step.

**2. An example of cavity length in the dynamic state based on a FORTRAN module** (The initial effective pressure *Pe*＝0.4 MPa, the amplitude of water level fluctuations *w*＝100 m, and *Us*＝500 m yr-1. *L(i)*＝*L(ti), L(i+1)*＝*L(ti+1), h(i)*＝*h(ti), h(i+1)*＝*h(ti+1)*)

do *i*＝1,17000 (The purpose of this part is to calculate the shape of cavity roof)

*R*＝31.82 (The cavity radius is calculated by the algorithm above in the steady state)

(The size of every step is 100s)

*L(1)*＝0.0015855 (Assume *L(0)*＝0, *h(0)*＝0, *L(1)*＝100×500/365/24/3600)

*h(1)*＝0.000239 (Assume *L(0)*＝0, *h(0)*＝0, *h(1)*＝31.82×(400000/3/3.16/107) 3×100)

end do

do *k*＝1,34560 (40 days are divided by 100 s)

do *m*＝1,17000

if (*h(17002-m)*>1) then

*R*＝(1+*L(17002-m)* ×*L(17002-m)*)/2 (Calculate a new cavity roof radius *R* )

end if

end do

*LhR(k)*=(2×*R*-1) 0.5 (Calculate a new cavity length )

do *j*＝1,17000

*L(1)*＝0.0015855 (Every time, *L(1)* is the same because of *Us*＝500 m yr-1 )

*Uc/R*＝((400000+(0.5-0.5×cos(2×3.14159×100×*k*/3600/24)) ×100×1000×9.8)/3/3.16/(107)) 3

(In this model, water level fluctuate 100 m on a daily basis, continue 40 days. But these parameters can be adjusted according to the need)

*h(1)*＝*Uc*×100

end do

*x(k)*＝*real(k)*

*y(k)*＝*LhR(k)*  (It is cavity length in dynamic state )

end do

**3. An example of computer program on a FORTRAN module** (The initial effective pressure *Pe*＝0.4 MPa, the amplitude of water level fluctuations *w*＝100 m, and *Us*＝500 m yr-1. The result is Figure 6)

Program B

use aplot

implicit none

type(aplot\_t)::p

real, dimension(34560)::*x,y*

real::*L(17001)* ----------------------------------------------------------- horizontal displacement

real::*h(17001)* ----------------------------------------------------------- vertical displacement

real::*LhR(34560)* ------------------------------------------------------- LhR means cavity length

real::*Uc* -----------------------------------------------------------------

real::*R* ------------------------------------------------------------------- cavity radius

integer::*T*

integer::*i*

integer::*j*

integer::*m*

integer::*k*

*R*＝1

*T*＝0

do *i*＝1,17000 --------------------------------------------This loop calculates the cavity shape in steady state

*L(1)*＝0.0015855 ------------------------------------------------------ initial value of the first point

*L(i+1)*＝0.999992488\**L(i)*+0.0015855

*h(1)*＝31.82\*7.512\*10\*\*(-6) ----------------------------------- 31.82 m is the cavity radius in steady state

*h(i+1)*＝*h(i)*+(( 31.82\*31.82-*L(i)\*L(i)*)\*\*0.5)\*0.000007512

end do

do *k*＝1,34560 ---------------------------------------- This loop calculates the cavity shape in dynamic state

do *m*＝1,17000 --------------------------------------- This loop calculates the cavity radius

if (*h(17002-m)*>1) then

R＝(1+*L(17002-m)*\**L(17002-m)*)/2

end if

end do

*LhR(k)*＝(2\**R*-1)\*\*0.5

do *j*＝1,17000 ----------------------------------- This loop calculates the cavity shape every 100 seconds

*L(1)*＝0.0015855

*Uc*＝((400000+(0.5-0.5\*cos(2\*3.14159\*100\**k*/3600/24))\*100\*1000\*10)/3/3.16/(10\*\*(7)))\*\*3

*h(1)*＝*R\*Uc*\*100\**R/R*

*L(17002-j)*＝(1-*Uc*\*100\**R/R*)\**L(17001-j)*+0.0015855

*h(17002-j)*＝*h(17001-j)*+((*R\*R*-*L(17001-j)\*L(17001-j)*)\*\*0.5)\**Uc*\*100\**R/R*

end do

*x(k)*＝*real(k)*

*y(k)*＝*LhR(k)*

end do

p＝initialize\_plot()

call set\_yscale(p,0.0,10.0)

call set\_xlabel(p, "*k*")

call set\_ylabel(p, "*LhR*")

call add\_dataset(p,*x,y*)

call display\_plot(p)

call destroy\_plot(p)

print ………

end