FLAC3D Gitbook

Kyeong Sun Kim

Civil and Environmental Engineering

Seoul National University

Sept, 2021

Contents

There is grandeur in this view of life, with its several powers, having been originally breathed into a few forms or into one; and that, whilst this planet has gone cycling on according to the fixed law of gravity, from so simple a beginning endless forms most beautiful and most wonderful have been, and are being, evolved.

Charles Darwin(1809-1882)

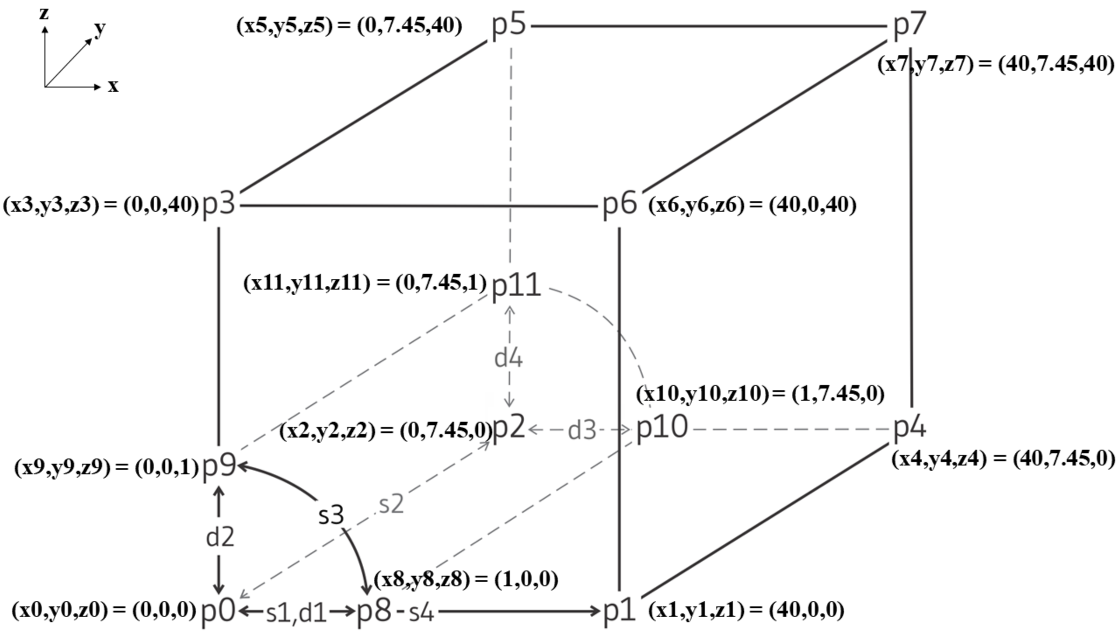
1

KAIST Model

1.1 Initial Configuration

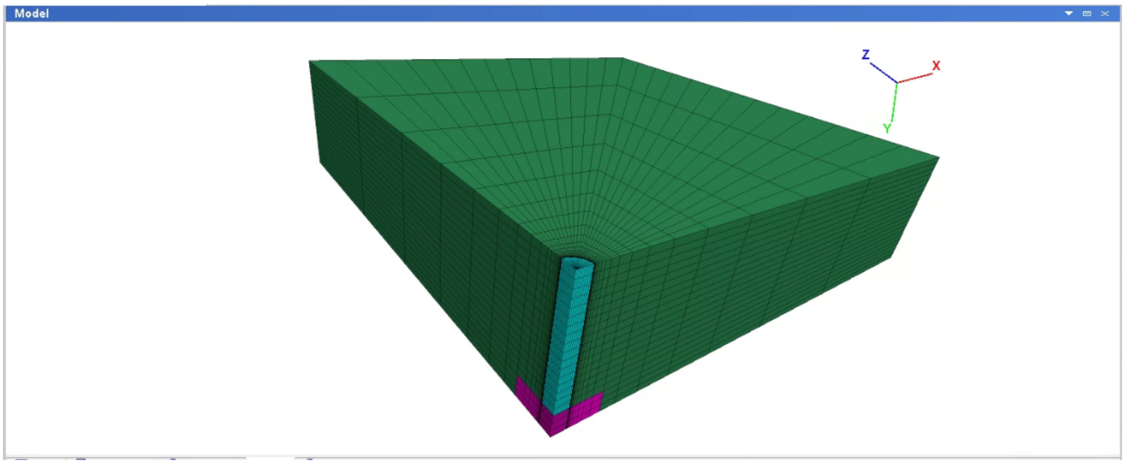
```
import itasca as it
import numpy as np
np.set_printoptions(threshold=20)
it.command("python-reset-state false")
from itasca import zonearray as za
from itasca import gridpointarray as gpa
```

1.2 Zones




```
model new
zone create radial-cylinder point 0 (0,0,0) ...
                            point 1 (40,0,0) ...
                            point 2 (0,10,0) ...
                            point 3 (0,0,40) ...
                            point 4 (40,10,0) ...
                            point 5 (0,10,40) ...
                            point 6 (40,0,40) ...
                            point 7 (40,10,40) ...
                            point 8 (1,0,0) ...
                            point 9 (0,0,1) ...
                            point 10 (1,10,0) ...
                            point 11 (0,10,1) ...
                            size 10 20 26 40 ...
                            rat 1 1 1 1.5 ...
                            fill group "shaft"
```

```
zone group "plate" range position-x 0 3.2 position-y 8.5 10 position-z 0 3.2
```



1.3 Group

```
zone group "plate" range position-x 0 3.2 position-y 8.5 10 position-z 0 3.2
```

1.4 Constitutive Model

```
#Constitutive model and properties for soil
zone cmodel assign strain-softening range group "Radial Cylinder1"
zone property density 2500 bulk 2e8 shear 1e8 range group "Radial Cylinder1"
zone property cohesion 2e6 friction 45 tension 2e5 dilation 10 range group "Radial
zone property table-friction 'fri' table-cohesion 'coh' table-dilation 'dil' range
table 'fri' add (0, 45) (.05, 42) (.1, 40) (1, 40)
table 'coh' add (0,2e6) (.05,1e6) (.1,5e5) (1,5e5)
table 'dil' add (0, 10) (.05,  3) (.1,  0)
```

Table 1: Material Properties for a Concrete Pile Foundation in Clay

	Concrete Pile	Clay
Dry density	2500 kg/m^3	1230 kg/m^3
Wet density	–	1550 kg/m^3
Elastic Properties:		
Young's modulus	25.0 GPa	100.0 MPa
Poisson's ratio	0.20	0.30
Bulk modulus	13.9 GPa	83.33 MPa
Shear modulus	10.4 GPa	38.46 MPa
Strength Properties:		
Cohesion	–	30 kPa
Friction angle	–	0.0

#Constitutive model and properties for shaft and plate zone cmodel assign elastic range group 'shaft' zone property bulk 8.333e7 shear 3.846e7 range group 'shaft' zone cmodel assign elastic range group 'plate' zone property bulk 8.333e7 shear 3.846e7 range group 'plate'

`\includegraphics[width=1\linewidth]{myfigureeeeee/d}`

`## Soil-Structure Interface`

`## Boundary Conditions`

`## Initial Equilibrium`

`## Alterations`

`## Results`

<!--->

<!--->

<!--->

\newpage

Axial Concrete Pile

Problem Description

Problem Statement

The pile is subjected to an axial load of 100 kN,

and then the top of the pile is moved horizontally for a displacement of 4 cm.

The goal is to determine relation of axial loading to the ultimate bearing capacity

And, lateral load-deflection curve is calculated.

1) origin at the top of the pile, z upward.

2) z=0: free surface

3) z=-8: fixed in z-eirection

4) x=+8, -8, y = 8: roller

5) skin friction is modeled by placing an interface

between pile concrete wall and clay.

In it, fric angle of 20 and c=30kPa are assumed.

6) toe interface is placed between pile tip and clay

*note: Zone faces are separated in a previous command so that the gridpoints common

note: include Figure of grid (geometry)

Main Parameters

Diameter = 0.6 m \

Length = 5 \

Clay \

GWT = 5.5m \

Modeling Procedure

1) equil. stress state under gravity load before install. \

1-1) water table is created at z=5.5 \

1-2) wet density of clay is assigned below this water table. \

2) equil. stress state after installation. \

2-1) change properties of pile zones from those \

representing clay to those representing concrete. \

2-2) vertical equil. stress distribution at this equil. state is shown in \

*note: include Figure of contours of vertical stress at ini state incld. pile weigh

3) apply vertical velocity at top of pile \

"ramp" = boundary condition is increased linearly \

note: critical timestep is controlled by high stiffness of concrete \

If velocity is sudden, inertial effects will dominate and renders difficulty to ide

table "ramp" is used to apply velocity to pile top gridpoints. \

*note: FISH FUNCTION vert_load calculates axial stress at the top of pile and store

For efficiency, gridpoints on cap surface are stored in symbol "cap" as a map \

*note: include plot of axial stress vs axial displ. at pile toe. ramp = (0,5e-8), s

*note: combined damping is used to remove kinetic energy for prescribed loading con

*note: FISH FUNCTION tot_reac monitors soil reaction along pile as a func of latera

note: include Figure of p-y curve at 11 equidistant points along pile \

Zones

‘‘‘python

model new

model title ’Axial and lateral loading of a concrete pile’

; create grid interactively from the extruder tool,

```
; exported to geometry.f3dat from State Record pane.
call 'geometry' suppress
zone generate from-extruder
; Reflect the grid to get a 1/2 space instead of a 1/4 space
zone reflect dip-direction 270 dip 90
```

1.5 Groups

```
; Name intersections of things named in the two extruder views
zone group 'clay' range group 'clay-c' or 'clay-s' or 'wetclay-s'
zone group 'pile' range group 'pile-c' group 'pile-s' or 'remove-s'
zone group 'remove' range group 'remove-s' group 'pile-c' not ;
zone face group 'wall' internal range group 'wall-c' group 'pile'
zone face group 'base' internal range group 'base-s' group 'pile'
zone face skin ; Name far field boundaries
; Delete the area marked for removal
zone delete range group 'remove'
;
; setup interfaces
; separate using zone separate
; all at once so common nodes are separated
zone separate by-face new-side group 'iwall' slot 'int' ...
     range group 'wall' or 'base'
; Want two different interfaces for proper normal direction at corner
zone interface 'side' create by-face range group 'wall' and 'iwall'
zone interface 'base' create by-face range group 'base' and 'iwall'
; Save initial geometric state
model save 'geometry'
```

1.6 Properties

```
; Initialize gravity, pore-pressures, density, and stres state
model gravity 10
; water table information
zone water density 1000
zone water plane origin (0,0,-5.5) normal (0,0,-1)
zone initialize density 1230
zone initialize density 1550 range group 'wetclay-s' ; Wet density
; assign properties to the soil and interfaces - temporarily remove pile cap
zone cmodel assign mohr-coulomb ...
     range group 'clay'
zone property bulk 8.333e7  shear 3.846e7  cohesion 30000 fric 0 ...
     range group 'clay'
zone cmodel assign elastic                 range group 'pile'
zone property bulk 8.333e7  shear 3.846e7  range group 'pile'
zone cmodel assign null                    range group 'remove-s'
zone interface 'side' node property stiffness-normal 1e8 ...
                           stiffness-shear 1e8 friction 20 cohesion 30000
zone interface 'base' node property stiffness-normal 1e8 ...
                           stiffness-shear 1e8 friction 20 cohesion 30000
```

1.7 B.C. and I.C.

```
; boundary and initial stress conditions
zone face apply velocity-normal 0 range group 'Bottom'
zone face apply velocity-normal 0 range group 'East' or 'West'
zone face apply velocity-normal 0 range group 'North' or 'South'
zone initialize-stress ratio 0.4286
```

```
zone interface 'side' node initialize-stresses
zone interface 'base' node initialize-stresses
```

1.8 Initial Equilibrium

```
; Solve to initial equilibrium
zone ratio local
model solve ratio 1e-4
model save 'initial'
```

1.9 Alterations

1.9.1 install the pile

```
; install the pile
model restore 'initial'
zone cmodel assign elastic                        range group 'pile'
zone property bulk 13.9e9 shear 10.4e9 density 2500 range group 'pile'
model solve ratio 1e-4
model save 'install'
```

1.9.2 vertical loading

```
; vertical loading
zone initialize state 0
zone gridpoint initialize displacement (0,0,0)
zone gridpoint initialize velocity     (0,0,0)
table 'ramp' add ([global.step],0) ([global.step+30000],-5e-8) ...
      ([global.step+58000],-5e-8) ; Increase velocity applied to pile
                                  ; over 30,000 steps
```

```
zone face apply velocity-normal 1 table 'ramp' range group 'Top'
history interval 250
zone history name 'disp' displacement-z position (0,0,0)
call 'load'
fish history name 'load' @vert_load
zone mechanical damping combined
model step 58000
model save 'vertical-loading'
```

1.9.3 vertical then lateral loading

```
; vertical loading then lateral loading
model restore 'install'
zone initialize state 0
zone gridpoint initialize displacement (0,0,0)
zone gridpoint initialize velocity     (0,0,0)
zone face apply stress-zz [-1.0e5/(math.pi*0.3*0.3)] range group 'Top'
model solve ratio 1e-4
model save 'lateral-load-start'

; apply lateral loading as x-velocity on cap
zone initialize state 0
zone gridpoint initialize displacement (0,0,0)
zone gridpoint initialize velocity     (0,0,0)
zone face apply velocity-x 1e-7 range group 'Top'
zone history name 'disp' displacement-x position 0,0,0
call 'p-y' suppress ; Calculates p-y curve for pile, when tot_reac is called
@make_pydata        ; Generate p-y curve calculation data
@output_structure   ; Sanity check of p-y curve data
fish history name 'load' @tot_reac
```

```
model step 416500
model save 'lateral-load'
```

2

Pull-Tests

2.1 Problem Description

note: FISH function force is used to sum the raction forces and monitor nodal displacement generated by the pull-test

note: free length of bolt that extends out of block + larger diameter

Perfectly plastic behavior of grout = max cohesion is exceeded +post-peak weakening of shear bond strength

note: bond strength softening of the grout is defined with keyword coupling-cohesion-table (see Rockbolt Behavior)

The relation btw shear disp. and cohesion weakening is prescribed

thru table cct. softening of friction of grout canbe defined using keyword coupling-friction-table.

2.2 Zones

```
; ====================================================================
;   Simulation of pull-test for grouted reinforcement
;   using modified pile elements - Softening of cohesion
; ====================================================================
model new
fish automatic-create off
model title 'Pull-test using modified pile elements - cohesion softening'
; Create a single rock block and set its material properties.
zone create brick size 4 4 6 point 1 (0.4,0,0) point 2 (0,0.4,0) ...
                             point 3 (0,0,0.6)
```

2.3 Properties

```
zone cmodel assign elastic
zone property bulk 5e9 shear 3e9
zone face apply velocity-normal 0.0 range position-z 0.6
; Create a pile element and assign properties
struct pile create by-line (0.2,0.2,0.1) (0.2,0.2,0.7) segments 12
struct pile property rockbolt-flag on
struct pile property young 200e9 poisson 0.25 cross-sectional-area 5e-4 ...
                     perimeter 0.08
struct pile property tensile-yield   2.25e5   ; ultimate tensile strength
struct pile property moi-y 2.0e-8 moi-z 2.0e-8 moi-polar 4.0e-8 ; 0.25*pi*r^4
struct pile property coupling-cohesion-shear 1.75e5 ...
                     coupling-stiffness-shear 1.12e7
struct pile property coupling-cohesion-normal 1.75e5 ...
                     coupling-stiffness-normal 1.12e7
```

```
struct pile property coupling-cohesion-table 'cct'
; change in cohesion with relative shear displacement
table 'cct' add (0,1.75e5) (0.025,1.75e4)
```

2.4 Initial Equilibrium

```
struct node fix velocity-x range position-z 0.7
struct node initialize velocity-x 1e-6 local range position-z 0.7
call 'pileforce' suppress ; FISH function calculates reaction force on zones
```

2.5 Alterations

```
; Set up histories for monitoring model behavior
history interval 10
fish history name 'force' @force
struct node history name 'disp' displacement-z position (0.2,0.2,0.7)
; Achieve a total displacement of 4.0 cm
model cycle 40000
;
model save 'pull-5'
```

2.6 Some other notes

2.3. pull test with confinement "Pulltest06.f3dat" +modified pile logic.(see Behavior of Shear Coupling Springs) linear law is implemented.whereby reinforcement shear strength is defined as constant

*(coupling-cohesion-shear)+ effective pressure*perimeter*fric angle(coupling-friction-shear)*

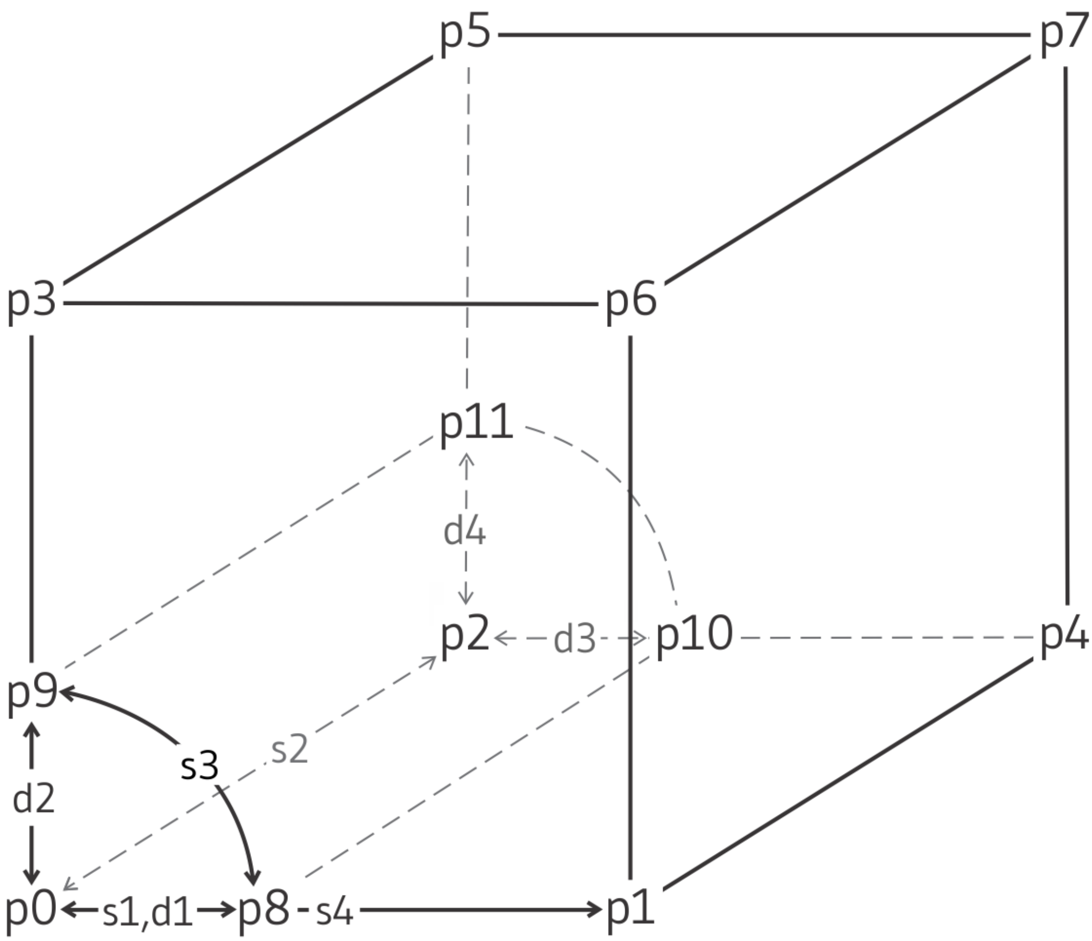
This pressure dependence is activated automatically by issuing reinforcement properties(perimeter) and

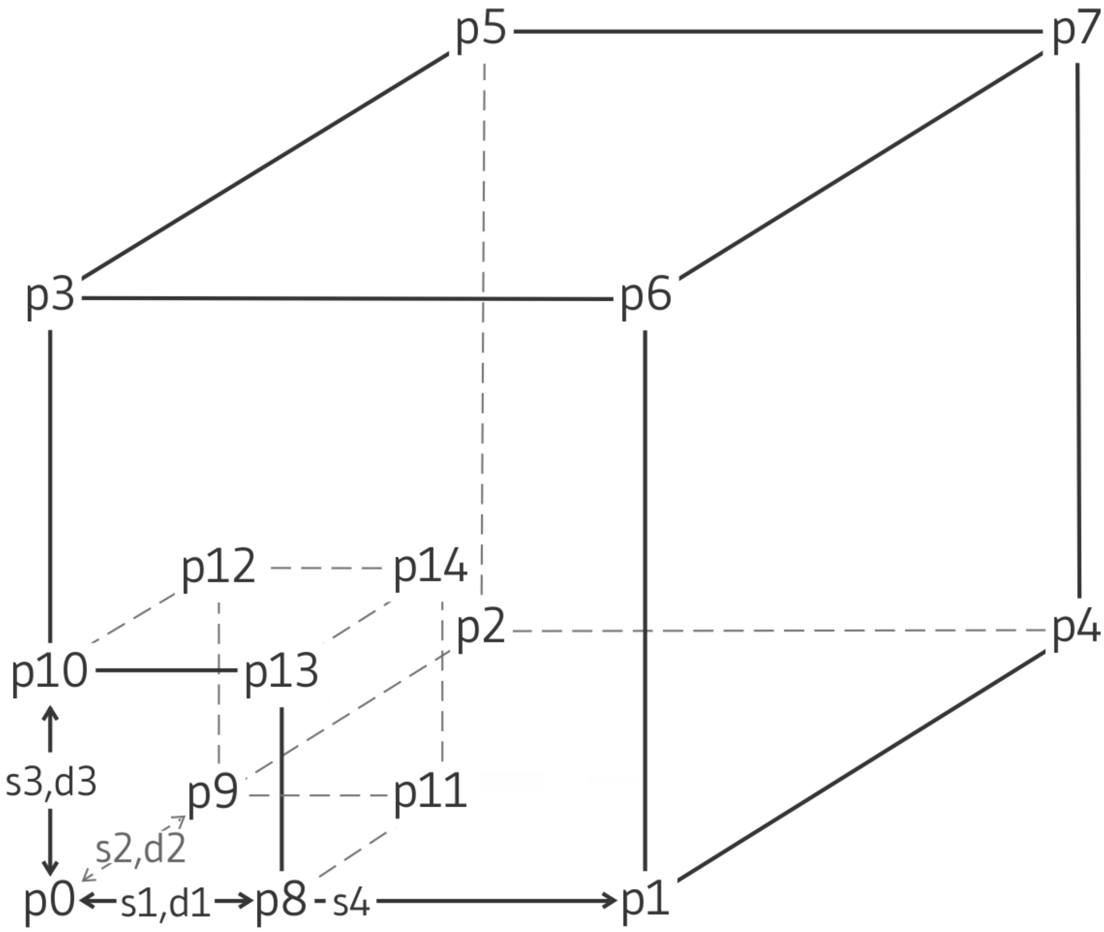
2.5. pull test with tensile rupture “Pulltest08.f3dat” *note: tensile-yield, tensile-failure-strain: for limiting axial yield force and limiting axial strain for rockbolt*

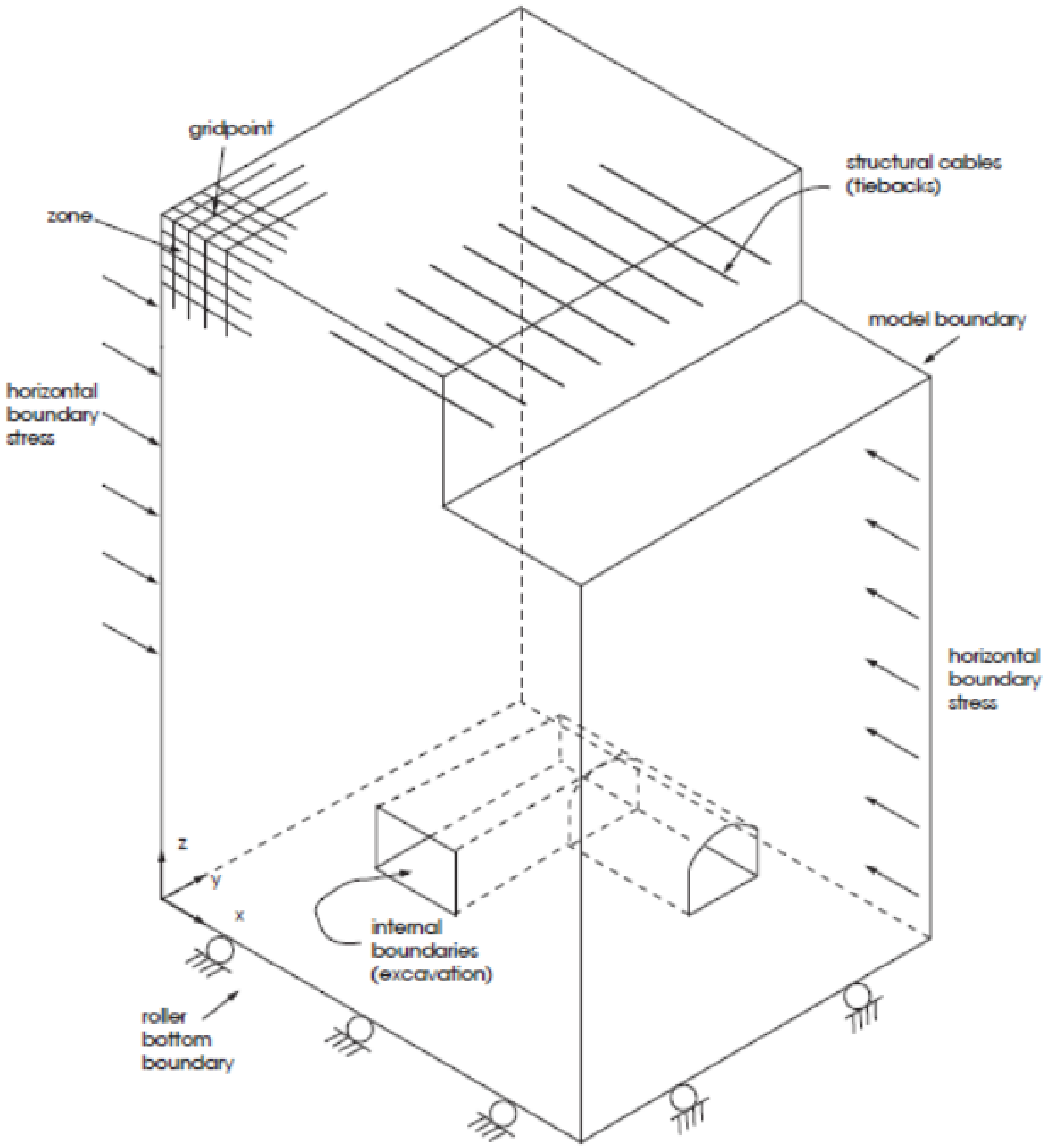
3

Grid

3.1 Primitive Shapes




p5
p7
p3
p6
p12
p14
p2
p4
p10
p13
s3,d3
p9
p11
s2,d2
p0
s1,d1
p8
s4
p1



note: zone create generates primitive grid

note: zone gridpoint create puts gridpoints at specific locations

note: zone gridpoint merge ensures separate primitives are connected properly

note: zone attach connects primitive meshes of different zone sizes.

```
zone create radial-cylinder size 5 10 6 12 fill
zone create radial-cylinder size 5 10 6 12 ratio 1 1 1 5
each size is controlled by a ratio (geometric ratio of 1.2 times preceding zone)
```

ex) 5 along innder radius of cylindrical tunnel,

10 along axis

6 along circumference of tunnel

12 between periphery of tunnel and outer boundary of model

note: size keyword defines the number of zones in the grid.

keywords for zone create:

- dimension - edge - fill - point (boundary dimensions) - ratio (coarser toward edge) - size

3.2 several primitive shapes connected:

```
zone create radial-cylinder size 5 10 6 12 rat 1 1 1 1.2 ...
                           point 0 (0,0,0) point 1 (100,0,0) ...
                           point 2 (0,200,0) point 3 (0,0,100)
zone create radial-tunnel size 5 10 5 12 rat 1 1 1 1.2 ...
                          point 0 (0,0,0) point 1 (0,0,-100) ...
                          point 2 (0,200,0) point 3 (100,0,0)
; here, model boundary dimensions are 100, 200, 100
; boundary coord are defined using point keyword

zone reflect dip 90 dip-direction 270 origin (0,0,0)
```

this adds symmetric part.

note: The symmetry plane is a vertical plane (located by the dip, dip-direction, and origin keywords) coincident with the x = 0 plane. Note that dip angle (dip) and dip direction (dip-direction) assume that x corresponds to "East," y to "North", and z to "Up."

third option, the zone gridpoint create command, is available to position single points in the model region.

note: zone gridpoint create is used for positioning reference points of primitives

During execution of a zone create command, a check is made for each boundary gridpoint against the boundary gridpoints of zones that already exist.
If two boundary gridpoints fall within a tolerance of 1 ? 10-7 (relative to the magnitude of the gridpoints position vector) of each other, they are assumed to be the same point,
If it is discovered that some gridpoints don't match, the zone gridpoint merge command can be used to merge these gridpoints after the zone create command has been applied.

Example: (zone attach) - Two unequal sub-grids

```
zone create brick size 4 4 2 point 0 (0,0,0) point 1 (4,0,0) ...
                             point 2 (0,4,0) point 3 (0,0,2)
zone create brick size 8 8 4 point 0 (0,0,2) point 1 (4,0,2) ...
                             point 2 (0,4,2) point 3 (0,0,4)
zone attach by-face range position-z 2
```

Example: (zone densify)

```
zone create brick size 4 4 4
zone densify segments 2 range position-x 2 4
```

the first two command lines can be changed to where zone densify segments 2 refines the upper zones (between the z-coordinate of 2 and 4) with the segment number of 2 on each edge.

3.3 Structural Element Operation

Creating a liner in the service tunnel

```
; liner
structure shell create by-face range cylinder ...
                                    end-1 (0,0,-1) end-2 (0,50,-1) ...
                                    radius 3
```

The liner contains 240 structural shell elements and is connected to the FLAC3D grid at 143 structural-node links. The grid with the liner is shown below.

3.4 Densifying grid by specifying max size length

```
model new
zone create brick size 4 4 4
plot 'Brick' export bitmap filename 'densify3.png'
;
zone densify local maximum-length (0.5,0.5,0.4) range position-z 2 4
zone attach by-face
;
plot 'Brick' export bitmap filename 'densify4.png'
```

note that in the local z-direction, the maximum size length is 0.4. FLAC3D densifies the maximum length in this direction to be 1/3 (= 0.4)

The zone attach by-face command in this example is used to attach faces of sub-grids together rigidly to form a single grid

Always use the zone attach by-face command after the zone densify command if there are different numbers of gridpoints along faces of different zones.

3.4.1 Densify a grid using geometric information

```
model new
zone create brick size 10 10 10
;
geometry set "setA" polygon create ...
                    by-positions (0,0,1) ( 5,0, 1) ( 5,10, 1) (0,10,1)
geometry set "setA" polygon create ...
                    by-positions (5,0,1) (10,0, 5) (10,10, 5) (5,10,1)
geometry set "setB" polygon create ...
                    by-positions (0,0,5) ( 5,0, 5) ( 5,10, 5) (0,10,5)
geometry set "setB" polygon create ...
                    by-positions (5,0,5) (10,0,10) (10,10,10) (5,10,5)
plot 'Brick2' export bitmap filename 'densify5.png'

zone densify segments 2 range geometry-space "setA" set "setB" count 1
zone attach by-face
;
plot 'Brick2' export bitmap filename 'densify6.png'
```

4

Python with FLAC3D

4.1 Introduction

```
import itasca as it
it.command("python-reset-state false")
it.command("""
model new
zone create brick size 10 10 10
zone cmodel assign elastic
zone property density 2950 young 12e9 poisson 0.25
cycle 1
""")
it.zone.count()
z=it.zone.find(1)
print z
z.pos()
```

```
volume_sum = 0.0
for z in it.zone.list():
  volume_sum += z.vol()

print volume_sum
print z.vol() * it.zone.count()
assert volume_sum == z.vol() * it.zone.count()

z = it.zone.near ((5,5,5))
z.pos()
```

4.2 Zones

```
it.zone.count() # 1000
z = it.zone.find(1)
for z in it.zone.list():
  z = it.zone.near((5,5,5))
z.pos()
z.vol()
```

4.3 Properties

```
z.props() or z.props()['bulk']
z.prop('shear')
z.set_prop('bulk', 8.5e9)
```

4.4 Gridpoints

```
gp = it.gridpoint.near((2,2,2))
for gp in it.gridpoint.list():
  total_mass = gp.mass_gravity()
z.vol()*z.density()*1000
```

4.5 Structural Elements

```
it.structure.list()
it.structure.find(1)
it.structure.near((0,2,2))
it.structure.node.find(1)
s_node.links()[0]
```

4.6 Extra Variables

```
z.set_extra(1, 1.23)
z.set_extra(2, "a test string")
z.set_extra(1, gp.pos())
```

4.7 Groups and B.C.

```
if z.group("default") == "lower":
  gp.set_fix(0, True)
  gp.set_fix(1, True)
  gp.set_force_load((1e6,2e6,1e6))
it.zone.near((5,5,5)).stress()
```

```
it.zone.near((5,5,5)).strain()
"""
```

4.8 Parameteric Studies

```
"*note: for modulus in [6e9, 8e9, 10e9, 12e9]:"
it.command("""
model restore 'before_cycling'
zone prop young {}
model solve
""".format(modulus))
vertical_disp = it.gridpoint.near((5,5,10)).disp_z()
print "~~~".format(modulus,vertical_disp)
```

4.9 Setting FISH variables

```
import itasca as it
it.command('python-reset-state false')
it.fish.set('x', 10)
x = it.fish.get('x') yields 10
```

4.9.1 Issuing Command

```
import itasca as it
import numpy as np
data = np.loadtext('brick-data.txt')
command_template = ;;;
zone create brick
zone cmodel assign elastic
```

```
zone property density {density} young {young} poisson {poisson}
;;;
density = data[0]
young = data[1]
poisson = data[2]

command = command_template.format(density=density, young=young, poisson=poisson)
it.command(command)
```

4.10 String

```
"The value of x is {:.2f}".format(0.3872)
"The value of x is {:.2e}".format(0.3872)
"My name is Sasha"
"My name is {}".format("Sasha")
"My name is {name}".format(name="Sasha")
```

4.11 KAIST Model Code

```
import itasca as it
import numpy as np
np.set_printoptions(threshold=20)
it.command("python-reset-state false")
from itasca import zonearray as za
from itasca import gridpointarray as gpa
" ============================================ "
" ============GRID GENERATION=============== "
" ============================================ "
#
```

```
it.command("""
model new
zone create radial-cylinder point 0 (0,0,0) ...
                            point 1 (40,0,0) ...
                            point 2 (0,10,0) ...
                            point 3 (0,0,40) ...
                            point 4 (40,10,0) ...
                            point 5 (0,10,40) ...
                            point 6 (40,0,40) ...
                            point 7 (40,10,40) ...
                            point 8 (1,0,0) ...
                            point 9 (0,0,1) ...
                            point 10 (1,10,0) ...
                            point 11 (0,10,1) ...
                            size 10 20 26 40 ...
                            rat 1 1 1 1.5 ...
                            fill group "shaft"
zone group "plate" range position-x 0 3.2 position-y 8.5 10 position-z 0 3.2
""")

" ============================================ "
" ============GROUPS AND MASKS=============== "
" ============================================ "

" GROUPS AND MASK ARRAYS "
it.command("zone group \"lower\" range position-z 0 5")
za.in_group("lower")
za.in_group("lower").sum(), "zones in lower group."
corner_mask = reduce(np.logical_and, (x<3, y<3, z<3))
za.set_group(corner_mask, "corner", "geometry")
```

```
print za.in_group("corner", "geometry").sum(), "zones in corner group."

" GRIDPOINTS ARRAY FUNCTIONS "
gpos = gpa.pos()
gx, gy, gz = gpos.T
print gz
f = gpa.fixity()
print f
f[:,][gz==0] = True, True, True
print f
gpa.set_fixity(f)

top_gridpoints = gz==10
radial_distance = np.sqrt((gx-5)**2+(gy-5)**2)
central_gridpoints = radial_distance < 5
mask = np.logical_and(top_gridpoints, central_gridpoints)
print "boundary load applied to {} gridpoints".format(mask.sum())
fapp = gpa.force_app()
print fapp
fapp[:,2] = mask*1e6*(5.0-radial_distance)/5.0
gpa.set_force_app(fapp)



print "zone centroids: "
print za.pos()
za.gridpoints()
za.faces()
za.ids()
print za.neighbors()
```

```
" ============================================ "
" ===============PROPERTIES================== "
" ============================================ "

it.command("""
zone cmodel assign elastic
zone property density 2950 young 12e9 poisson 0.25
cycle 1
""")


" ============================================ "
" ==========BOUNDARY CONDITIONS============== "
" ============================================ "




" ============================================ "
" =================RESULTS=================== "
" ============================================ "

it.command("model solve")
print "gridpoint displacements:"
print gpa.disp()
print "gridpoint displacement magnitudes: "
mag = np.linalg.norm(gpa.disp(), axis=1)
print mag
```

```
max_index = np.argmax(mag)
print "Maximum displacement: {} at location {}".format(gpa.disp()[max_index],
                                                     gpa.pos()[max_index])

print "Vertical displacement along the vertical line x=5, y=5: from z=0 to z=10"
print gpa.disp()[np.logical_and(gx==5, gy==5)][:,2]

za.stress()

za.stress_flat()


" ============================================ "
" =========REFERENCE EXAMPLES================ "
" ============================================ "


""" Some Numpy Operation Examples
np.array([1,2,3,4,5])
np.linspace(0,1,15)
np.zeros((4,4))
a = np.linspace(0,1,15)
b = np.ones_like(a)
np.sin(a)
print a[0]
a[0] = 20.2
print a
c = np.array(((1,2,3),(4,5,6),(7,8,9),(10,11,12)))
print c
c[0][0]
```

```
c[:,0]
"""
""" SOME GRIDPOINTS EXAMPLES
z = it.zone.near((5,5,5))
print "central zone id: {}, position: {}".format(z.id(), z.pos())

for gp in z.gridpoints():
    print "gridpoint with id: {} at {}".format(gp.id(), gp.pos())
"""
```

Appendices

A

1. Template

A.1 Problem Description

A.2 Modeling Procedure

A.3 Zones/Groups

A.4 Properties

A.5 B.C. and I.C.

A.6 Initial Equilibrium

A.7 Alterations

A.8 Results

B

Reference Collective

B.0.1 Uplift Resistance of Anchor Plate

B.0.1.1 Before 1968

- Coulomb
- Mohr
- Kotter's equation
- Balla (1961)
- Mors
- Matsuo

B.0.1.2 Post-1968

- Meyerhof, G.G., and Adams, J.I. 1968
- Meyerhof, G.G. 1973
- Das, B.M., and Seeley, G.R. 1975
- Rowe, R.K., and Davis, H. 1982
- Dickin, E.A., and Leung, C.F. 1983

- Murray, E.J., and Geddes, J.D. 1987
- Dickin, E.A. 1988
- Koutsabeloulis, N.C., and Griffiths, D.V. 1989 #### Post-2000
- Merifield, R.S., and Sloan, S.W. 2006

B.0.2 Numerical Analysis

B.0.3 Standards

- IEEE 2001
- DS 1110, DS 1111

B.0.4 Textbook

- Das, B. M. 2013. Earth Anchors

B.0.5 Ph.D Thesis

B.0.6 Award Lecture