IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE)
e-ISSN: 2278-1684,p-ISSN: 2320-334X, Volume 18, Issue 6 Ser. III (Nov. – Dec. 2021), PP 36-46
www.iosrjournals.org

Discretization of elemental modeling of soil using PFC code

Adhyatma Khare[1], Dr. S.C. Gajbhiye[2*], Dr. S.K. Jaiswal[3], Dr. R.N. Khare[4]
[1]B.Tech. (Sem.-VIII), [2]Assistant Prof. ,[3,4]Professor,
[1,4]Bhilai Institute of Tech. Durg, [2,3]Vishwavidyalaya Engg. College Ambikapur,
[2]drscgajbhiye@gmail.com, [4]rn_khare@rediffmail.com
****Corresponding Author***

Abstract:
The continuous improvement in the expansion soil properties makes helpful in the construction of flexible pavement. The stability of the soil is mainly determined by the force chains and pore channels, whereas the hydraulic conditions act as external disturbances. The erosion process is accompanied by contact failure, force chain bending, kinetic energy burst of particles, and other processes due to multi-factor coupling. This paper recaps the various researches done using the discrete element method and explores the new advanced techniques for the simulation of particles such as clay, sand, and many more, using Particle Flow Code (PFC). This PFC has two in-built programs, named as PFC2D and PFC3D. PFC3D Software is basically a generic numerical method for computing the motion of small particles like clay. With the help of this framework, the various particles have been modeled and then simulated. The simulation tests have been done on the different small modeled particles, considering the crack pattern in a bonded assembly of balls. It is observed that the flat joint contact model have the more realistic tensile to compressive ratio than that of in parallel bonded model.

***Key Word**: discrete element modeling; site characterization; erosion; calibration camber;*

Date of Submission: 14-12-2021 Date of Acceptance: 28-12-2021

I. Introduction

Soil clay particles in granular form are omnipresent in nature and widely used in various industries ranging from road construction in civil, food and pharmaceutical to agriculture and mining – among others. It has been estimated that about 10% of the world's energy consumption is used in the processing, storage and transport of granular materials [1]. Owing to complexities like dilatancy, shear band formation and anisotropy, their behavior is far from completely understood [2]. To gain an understanding of the deformation behavior, various laboratory element tests can be performed. Element tests are (ideally homogeneous) macroscopic tests in which the experimentalist can control the stress and/or strain path. One element test that can be performed is the uni-axial compression test. In order to understand the behavior of soil clay granular materials, element tests can be performed. Element tests are ideally homogeneous laboratory experiments that allow the user to control the stress/strain path [3], [4], [5]. Such macroscopic experiments are useful in developing and calibrating constitutive relations, but provide little information on the microscopic origin of the bulk flow behavior. An alternative approach is to perform discrete element simulations (DEM). It also explored that the coupling effect of the recycled ash and natural fibers to control the expansive soil's strength and durability attributes. The bottom ash (BA) was used to chemically treat the expansive soil and coir fibers (CF) as reinforcement against tensile cracking [6], [7].

Geotechnical engineering analysis and design require accurate stratigraphic characterization and estimation of engineering soil properties. However, due to the difficulties associated with obtaining undisturbed samples of coarse-grained soils, engineers often rely on field tests to estimate properties of sands and gravels [7], [8]. The cone penetration test (CPT) has emerged as one of the most widely used tools for site characterization owing to its high repeatability and accuracy, simplicity, high data resolution, rapid-ity and robustness, and industry familiarity. The CPT does not measure soil properties directly; rather, it measures soil response to the loading imposed by probe penetration. Thus, extensive research has been conducted to develop empirical correlation [9], [10].

II. FEM based literature review

Soils are composed of organic matter (stuff that used to be alive, like plants and animals) and small inorganic matter. There are three basic soil types: sand, silt, and clay. Sand is comprised of tiny rock fragments and is the roughest in texture. Clay becomes sticky or greasy when wet, and very hard when dry. Silt is between sand and clay in texture. Loam is a combination of sand, silt, and clay, and has a high proportion of organic matter. It is the best soil for most plants. Almost all soils found in nature contain some proportion of two or more of the three soil types along with some organic matter. Therefore, these soils are loam, but vary in the proportions of clay, silt, sand, and organic matter. The so-called discrete (or distinct) element method (DEM) was originally developed by Cundall [11], for the analysis of rock mechanics problems. The basic formulation of the DEM using spherical or cylindrical particles was later proposed by Cundall and his colleagues to investigate the constitutive laws for soil. Cundall and Hart [12] showed that DEM is better at modelling a discontinuous material than other numerical tools such as the finite element method. The DEM assumes that the material can be represented by an assembly of rigid particles interacting among themselves. The overall behaviour of the system is determined by the cohesive/frictional contact laws. The contact law can be seen as the formulation of the material model on the microscopic level. Cohesive bonds can be broken, which allows simulating fracture of material and its propagation.

The various algorithms approaches have been developed for the generation of cylindrical (2D) and spherical (3D) particles. The algorithms can be classified in three different groups, considering how it works: constructive algorithms, dynamic algorithms and collective re-arrangement algorithms. All of them generate random packages. A fourth group can be considered for the regular structure packages, but the behaviour of materials with irregular micro-structure cannot be reproduced, and are not considered in this work [13], [14], [15]. The advanced techniques called Particle Flow Code (PFC) Modeling software has been used for the simulation of particles such as clay, sand, and many more, in which the discretization of the element have been prepared. PFC generally has two programs, namely Particle Flow Code for Two Dimension (PFC2D) and Particle Flow Cod for Three Dimension (PFC3D). This paper reflects the results and discussion of the simulation of the soil particles using PFC3D. The following are the commands used for the simulation of particles: Ball command; Contact command; Wall command; Fish command; Model command; Group command; Domain command, and Ball generate command. Using the above command the following algorithm has been prepared. Using this code, a bonded assembly of 30 balls interacting in a box with two conditions one with no energy dissipation, and other with energy dissipation via frictional slip and viscous damping at the contact, as shown above in figure 1

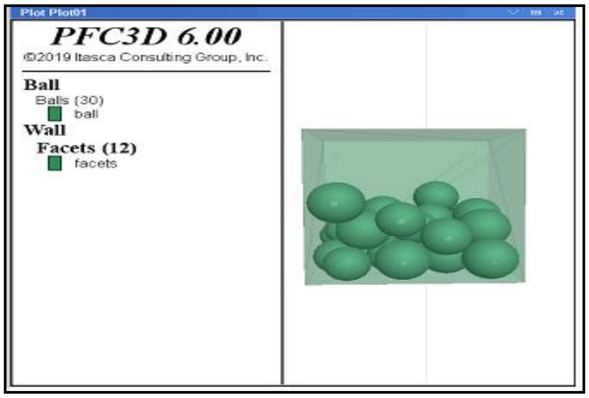


Figure 1: 30 balls in a box

Algorithms representing the ball in a box:

```
model new
model title 'Balls in a box'
model domain extent -10.0 10.0
contact cmat default model linear property kn 1.0e6
wall generate box -5.0 5.0 one-wall
model random 10001
ball generate radius 1.0 1.4 box -5.0 5.0 number 30
ball attribute density 100.0
model gravity 10.0
model save 'initial-state’
model solve time 10.0
model save 'caseA-nodamping'
model restore 'initial-state'
contact cmat default model linear ...
            property kn 1.0e6 ks 1.0e6 fric 0.25 dp_nratio 0.1
contact cmat apply
model solve time 10.0
model save 'caseB-damping'
```

III. Geometric model formulation and their working process with different test using FE analysis

The geometric formulation or modeling on PFC for clay has been done to simulate those small particles in a bonded assembly of balls (Fig. 2). Once the simulation has been done, the crack pattern has to be generated for the same soil model. The conditional test algorithm (Appendix-1) has been prepared to run the various test such as sleeved tri-axial, compression and direct tension test to derive the unconfirmed confined strength (UCS) of a modeled sample. The motion of the modeled clay has been resisted by allowing the finite length, using the flat joint contact model concept. It is observed that the flat joint contact model gives a more realistic tensile to compressive ratio than that of parallel bonded model. The various results due to the boundary restriction in terms of force and their effects between balls have been shown in figure 3 and figure 4, respectively as contact force and displacement. The corresponding stress-strain formulation has been observed and shown the test result figure 5.

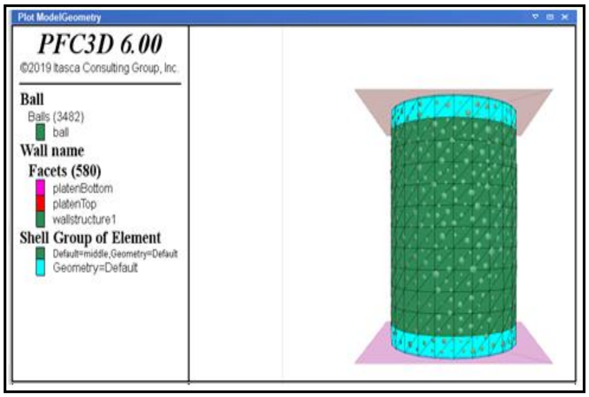


Fig.2: Geometric Model

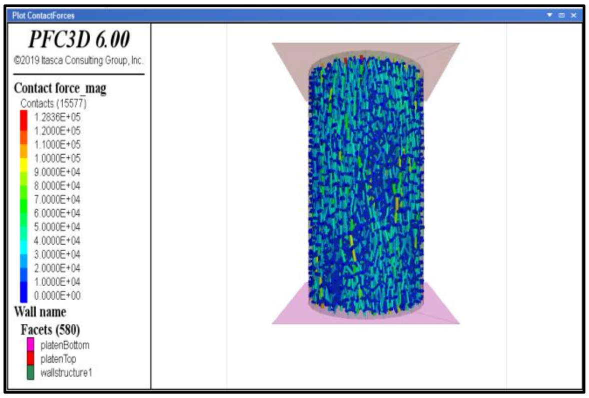


Fig. 3: Image: Contact Force

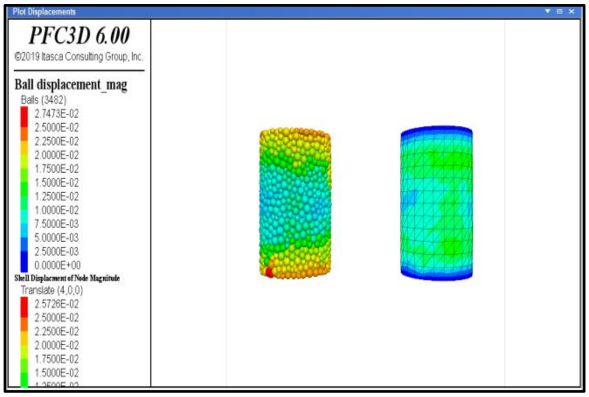


Fig. 4: Ball Displacements

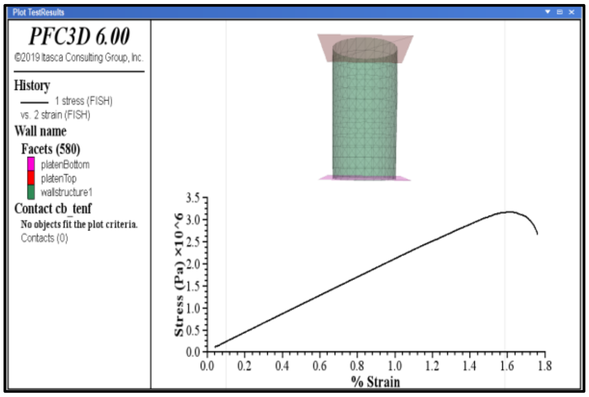
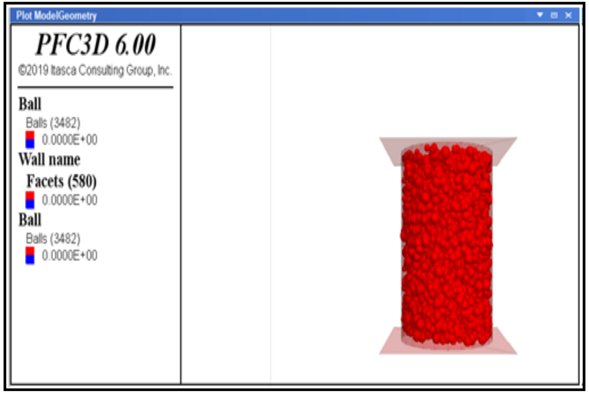


Fig. 5: Test Results

The Fragmentation Analysis have been done in PFC3D, considering the un-axial compression with crack pattern. In this, the various results have been observed at a failure condition under uni-axial unconfined compression, but only two results have been shown in figure 6 and figure 7. The three cases has been formulated as CASE-1: It generally corresponds to a regular run; CASE-2: During the simulation, the fragment logic is active, and the ball results command is used to save partial state information every 0.1; and CASE-3: A FISH function is registered with the linear parallel bond break callback event, which monitors bond breakage events and creates fractures using the Discrete Fracture Network logic to store their location, size, and orientation, in addition to the fragment logic and the ball results command. In the final state for CASE-33, a combination of tensile and shear

failures occurred in a selected bonded contacts and it resulted in the microscopic failure of the specimen (Appendix-2).



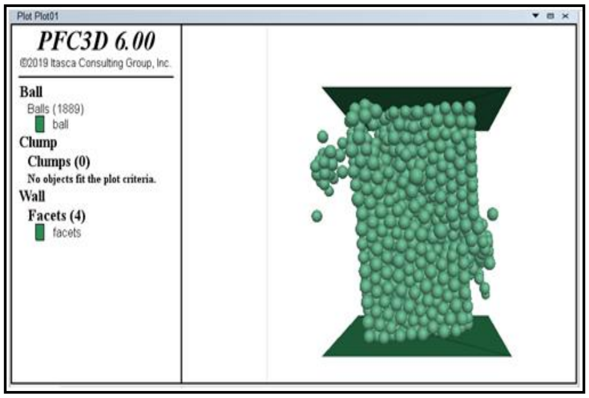


Fig. 6: Fragmentation (Part 2) in Cylindrical Shape **Fig. 7: Fragmentation (Part 1) in Rectangular Box**

IV. Result and Discussion

The model of soft clay has been prepared with the finite descritization, introducing the crack pattern in it, the UCS Tri-axial test had been performed on PFC3D. Using the conditional test algorithm, it is observed that the flat joint contact model gives a more realistic tensile to compressive ratio than that of parallel bonded model, as result in figures 3, 4 and 5. In addition to that the fragmentation analyses has been performed under the compressive load, showing the crack pattern in a combination of tensile and shear failure and thus result observed, as shown in figures 6 and 7.

Appendix-1 (Code for Sleeved Triaxial Test)

```
;-----------------------------------------------------------------
;              Sleeved Triaxial Test of a Bonded  Material
;-----------------------------------------------------------------
model new
model title 'Sleeved Triaxial Test of a Bonded Material'
;make repeatable by setting the random number seed
model random 10001
;set the model domain for PFC balls and walls
model domain extent -2 2 condition destroy
;largestrain mode must always be on for coupled simulations
model largestrain on
;apply timestep scaling so the PFC timestep will be 1
model mechanical timestep scale
;define relevant parameters for the cylinder
[rad = 1.0]
[height = 3.0]
[segments = 6]
[halfLen = height/2.0]
[freeRegion = height/2.0*0.8]
;create the cylinder with the geometry logic
;first create arcs making a circle of edges
geometry edge create by-arc origin (0,0,[-halfLen]) ...
  start ([rad*(-1)],0,[-halfLen]) end (0,[rad*(-1)],[-halfLen]) ...
  segments [segments]
geometry edge create by-arc origin (0,0,[-halfLen]) ...
  start (0,[rad*(-1)],[-halfLen]) end ([rad],0,[-halfLen]) ...
  segments [segments]
geometry edge create by-arc origin (0,0,[-halfLen]) ...
  start ([rad],0,[-halfLen]) end (0,[rad],[-halfLen]) ...
  segments [segments]
geometry edge create by-arc origin (0,0,[-halfLen]) ...
  start (0,[rad],[-halfLen]) end ([rad*(-1)],0,[-halfLen]) ...
  segments [segments]
;extrude the edges to make a cylinder
geometry generate from-edges extrude (0,0,[height]) segments [segments*2]

;import the structural elements as a shell from the geometry logic
;and assign groups and properties
structure shell import from-geometry 'Default' element-type dkt-cst
structure node group 'middle' range position-z [-freeRegion] [freeRegion]
structure node group 'top' range position-z [freeRegion] [freeRegion+1]
structure node group 'bot' range position-z [-freeRegion-1] [-freeRegion]
structure shell group 'middle' range position-z [-freeRegion] [freeRegion]
structure shell property isotropic (1e6, 0.0) thick 0.25 density 930.0
;initialize the structural element related data structures
model cycle 0
;In order to use the STRUCTURE SHELL APPLY command
;set the local system of each structural
;element to be pointing to the center of the triaxial cell
fish define setLocalSystem
```

```
    loop foreach local s struct.node.list()
        local p = struct.node.pos(s)
        local nid = struct.node.id.component(s)
        local mvec = vector(0,0,comp.z(p))
        zdir = math.unit(p-mvec)
        ydir = vector(0,0,1)
        command
            structure node system-local z [zdir] y [ydir] ...
                    range component-id [nid]
        endcommand
    endloop
           command
             structure node fix system-local
           endcommand
end
@setLocalSystem
;make sure that local damping is active for the structural elements
structure damp local
;fix the structural element nodes for specimen creation
structure node fix velocity rotation

;create a wall representation of the structural element
wall-structure create
;parameter to set the platen width relative to the cylinder radius
[rad2 = rad*1.3]
;create the platens
wall generate name 'platenTop' polygon ([-rad2],[-rad2],[halfLen]) ...
                                       ([rad2],[-rad2],[halfLen])  ...
                                       ([rad2],[rad2],[halfLen])   ...
                                       ([-rad2],[rad2],[halfLen])
wall generate name 'platenBottom' polygon ([-rad2],[-rad2],[-halfLen]) ...
                                          ([rad2],[-rad2],[-halfLen])  ...
                                          ([rad2],[rad2],[-halfLen])   ...
                                          ([-rad2],[rad2],[-halfLen])
;set the wall resolution strategy to full and set the cutoff angle for
;proximity contacts
wall resolution full
wall attribute cutoff-angle 20

;set the ball modulus and generate a cloud of balls with arbitrary overlap
[ballemod = 1.0e8]
ball distribute box [-rad] [rad] [-rad] [rad] [-halfLen] [halfLen] ...
                porosity 0.3 radius 0.05 0.1 ...
                range cylinder end-1 (0,0,[-halfLen]) end-2 (0,0,[height]) ...
                    radius [rad*0.95]
;set the ball attributes
ball attribute density 2600 damp 0.8
;set the default contact behavior -
;the deformability method sets properties of the
;linear portion of the contact model
contact cmat default model linearpbond method deformability ...
                  emod [ballemod] kratio 1.0
;allow the balls to rearrange,
;nulling the linear and angular velocities every 100 cycles
model cycle 1000 calm 100

;provide some friction to kill additional energy
;and apply this to all of the contacts
contact cmat default model linearpbond method deformability ...
                  emod [ballemod] kratio 1.0 property fric 0.3
```

```
contact cmat apply
;solve to a small average ratio
model solve ratio-average 1e-8
;bond the ball-ball contacts
contact method bond gap 1.0e-2 range contact type 'ball-ball'
;change the existing contact properties
;so that the linear force is incremental and
;supply strength
contact property lin_mode 1 lin_force 0 0 0 pb_ten 2e5 pb_coh 2e6
;set the stiffness of the parallel-bond portion of the contact model
contact method pb_deformability emod [ballemod] kratio 1.0 ...
        range contact type 'ball-ball'
;this set of operations removes all ball velocities
;and all contact forces in the model
;so that the specimen is completely stress free and bonded
model calm
ball fix velocity spin
model cycle 2
ball free velocity spin
;free the nodes in the middle section of the sleeve
;while keeping the top and bottom edges fixed
structure node free velocity rotation range group 'middle'
;function for calculating stress and strain as the platens are displaced
;these values will be recorded as a history
;first find the top and bottom platens
[platenTop = wall.find('platenTop')]
[platenBottom = wall.find('platenBottom')]
;define some variables for the calculation
[failureStress = 0]
[currentStress = 0]
[failureStrain = 0]
[area = math.pi()*rad^2.0]
;define the stress FISH function to measure the stress and strain
fish define stress
    local topForce = math.abs(comp.z(wall.force.contact(platenTop)))
    local botForce = math.abs(comp.z(wall.force.contact(platenBottom)))
    currentStress = 0.5*(topForce+botForce)/area
    stress = currentStress
    strain = (height - (comp.z(wall.pos(platenTop)) - ...
              comp.z(wall.pos(platenBottom))))/height * 100
    if failureStress <= currentStress
        failureStress = currentStress
        failureStrain = strain
    endif
end
;define the halt FISH function to stop cycling
;as the platens displace and the material fails
fish define halt
    halt = 0
    if currentStress < failureStress * 0.85
        halt = 1
    endif
end
;define a FISH function to increase the isotropic confining pressure
;in increments so that the bonded material does not break during loading
fish define rampUp(beginIn,ending,increment)
    command
        ball attribute displacement (0,0,0)
        structure node initialize displacement (0,0,0)
  endcommand
```

```
    begin = beginIn
    loop while (math.abs(begin) < math.abs(ending))
        begin = begin + increment
        command
            ;apply the confining stress
            structure shell apply [begin] range group 'middle'
            ;apply the same confining stress on the platens
            wall servo force (0,0,[begin*area]) activate true ...
                range name 'platenTop'
            wall servo force (0,0,[-begin*area]) activate true ...
                range name 'platenBottom'
            model cycle 200
            model calm
        endcommand
    endloop
    command
        ;once the stress state has been installed cycle and turn off the servo
        ;mechanism on the walls
        model cycle 1000
        wall servo activate false
        wall attribute velocity (0,0,0) range name 'platenTop'
        wall attribute velocity (0,0,0) range name 'platenBottom'
    endcommand
end
;set the platen velocity
[platenVel = 0.000003]

;save the model for future use, including these FISH utility function, before
;any confinement has been applied
model save 'beforeApplication'
;-----------------------------------------------------------------
;test 1: perform a UCS test on the specimen
structure shell delete
wall attribute velocity-z [-platenVel] range name 'platenTop'
wall attribute velocity-z [platenVel] range name 'platenBottom'
ball attribute displacement (0,0,0)
fish history @stress
fish history @strain
model solve fish-halt halt
model save 'ucs'
[io.out(string(failureStress) + 'Pa ')]
[io.out('at' + string(failureStrain) + '% strain')]
;-----------------------------------------------------------------
;test 2: perform a triaxial test with isotropic confining stress 1e4

model restore 'beforeApplication'
@rampUp(0,-1e4,-1e3)
model save 'to1e4'
wall attribute velocity-z [-platenVel] range name 'platenTop'
wall attribute velocity-z [platenVel] range name 'platenBottom'
ball attribute displacement (0,0,0)
structure node initialize displacement (0,0,0)
fish history @stress
fish history @strain
model solve fish-halt halt
model save 'triaxial1e4'
[io.out(string(failureStress) + 'Pa ')]
[io.out('at' + string(failureStrain) + '% strain')]
;-----------------------------------------------------------------
;test 3: perform a triaxial test with isotropic confining stress 5e4
```

```
model restore 'to1e4'
@rampUp(-1e4,-5e4,-1e3)
model save 'to5e4'
wall attribute velocity-z [-platenVel] range name 'platenTop'
wall attribute velocity-z [platenVel] range name 'platenBottom'
ball attribute displacement (0,0,0)
structure node initialize displacement (0,0,0)
fish history @stress
fish history @strain
model solve fish-halt halt
model save 'triaxial5e4'
[io.out(string(failureStress) + 'Pa ')]
[io.out('at' + string(failureStrain) + '% strain')]
;-------------------------------------------------------------------
;test 4: perform a triaxial test with isotropic confining stress 1e5
model restore 'to5e4'
@rampUp(-5e4,-1e5,-1e3)
model save 'to1e5'
wall attribute velocity-z [-platenVel] range name 'platenTop'
wall attribute velocity-z [platenVel] range name 'platenBottom'
ball attribute displacement (0,0,0)
structure node initialize displacement (0,0,0)
fish history @stress
fish history @strain
model solve fish-halt halt
model save 'triaxial1e5'
[io.out(string(failureStress) + 'Pa ')]
[io.out('at' + string(failureStrain) + '% strain')]
```

Appendix-2 (Code for Fragmentation)

Fragmentation (Part 1) in Rectangular Box:

```
; fname: fragment.p3dat
;; Illustrate how the Fragment, DFN and Model Result logic can be used
; to study the fragmentation of a Bonded Particle Model.
;;==============================================================
model new
model precision 15
model title 'Fragmentation of a Bonded Particle Model'
;-----------------------------------------------------------------------------
; -------------- INITIAL PHASE: CREATION OF A BONDED SPECIMEN ----------------
;-----------------------------------------------------------------------------
model domain extent -5 5 -5 5 -7.5 7.5 condition destroy
contact cmat proximity 0.1
contact cmat default model linear property kn 1e7
contact cmat default type ball-ball model linearpbond ...
                              property kn 5e6

wall generate box -2.5 2.5 -2.5 2.5 -5 5
model random 10001
ball distribute porosity 0.3 radius 0.25 0.3 box -2.5 2.5 -2.5 2.5 -5 5
ball attribute density 1.0e3 damp 0.7
model cycle 2000 calm 10
model solve calm 1000 ratio-average 5e-3
contact method bond gap 0.1
contact property pb_kn 1e7 pb_ks 1e7 pb_ten 4e5 pb_coh 4e5 pb_fa 20.0
wall delete
contact cmat default type ball-facet model linear ...
                               property kn 1e7 ks 1e7 fric 0.5

wall generate id 1 name 'top' polygon -4.0 -4.0 5.0 ...
                                -4.0  4.0 5.0 ...
```

```
                           4.0  4.0 5.0 ...
                           4.0 -4.0 5.0
wall generate id 2 name 'bottom' polygon -4.0 -4.0 -5.0 ...
                            -4.0  4.0 -5.0 ...
                             4.0  4.0 -5.0 ...
                             4.0 -4.0 -5.0
wall attribute rotation-center 0.0 0.0  5.0 range position-z  5.0
wall attribute rotation-center 0.0 0.0 -5.0 range position-z -5.0
model clean
model cycle 10
model solve elastic ratio-average 5e-3
wall attribute velocity-z 0.0 gradient 0.0 0.0 -0.001
model history name 1 mechanical time-total
wall history name 2 force-contact id 1
wall history name 3 force-contact id 2
model save 'initial'
;---------------------------------------------------------------------------
; ----------- CASE 0: NO TRACKING OF FRAGMENTS - NO BALL RESULTS -------------
;---------------------------------------------------------------------------
model solve time 50.0
model save 'final-case0'
;---------------------------------------------------------------------------
; ----------- CASE 1: CONTINUOUS UPDATE OF FRAGMENTS WITH BALL RESULTS -------
;---------------------------------------------------------------------------
model restore 'initial'

fragment register ball-ball
fragment activate time 0.1
model mechanical time-total 0.0
model result interval mechanical 0.1
model result warn off interval active true
ball result add-attribute fragment active true
wall result add-attribute velocity active true
model solve time 50.0
model save 'final-case1'

;program call 'post_treat.p3fis'
;@make_movie_case1
;---------------------------------------------------------------------------
; ----------- CASE 2: CUSTOM UPDATE OF FRAGMENTS AND BALL RESULTS ------------
;---------------------------------------------------------------------------
model restore 'initial'
program call 'fracture.p3fis'
@track_init
model solve time 50.0
@track_end
model save 'final-case2'
;program call 'post_treat.p3fis'
;@make_movie_case2
return
;=========================================================================
; eof: fragment.p3dat
```

Fragmentation (Part 2) in Cylindrical Shape:

```
; fname: fragment.p3dat
;
; Illustrate how the Fragment, DFN and Model Result logic can be used
; to study the fragmentation of a Bonded Particle Model.
;
;=========================================================================
model new
model precision 15
model title 'Fragmentation of a Bonded Particle Model'
;---------------------------------------------------------------------------
; -------------- INITIAL PHASE: CREATION OF A BONDED SPECIMEN ----------------
;---------------------------------------------------------------------------
model domain extent -5 5 -5 5 -7.5 7.5 condition destroy
contact cmat proximity 0.1
contact cmat default model linear property kn 1e7
contact cmat default type ball-ball model linearpbond ...
                     property kn 5e6
wall generate box -2.5 2.5 -2.5 2.5 -5 5
model random 10001
ball distribute porosity 0.3 radius 0.25 0.3 box -2.5 2.5 -2.5 2.5 -5 5
ball attribute density 1.0e3 damp 0.7
model cycle 2000 calm 10
model solve calm 1000 ratio-average 5e-3
contact method bond gap 0.1
contact property pb_kn 1e7 pb_ks 1e7 pb_ten 4e5 pb_coh 4e5 pb_fa 20.0
wall delete
contact cmat default type ball-facet model linear ...
                      property kn 1e7 ks 1e7 fric 0.5
wall generate id 1 name 'top' polygon -4.0 -4.0 5.0 ...
                      -4.0  4.0 5.0 ...
                       4.0  4.0 5.0 ...
                       4.0 -4.0 5.0
wall generate id 2 name 'bottom' polygon -4.0 -4.0 -5.0 ...
                        -4.0  4.0 -5.0 ...
                         4.0  4.0 -5.0 ...
                         4.0 -4.0 -5.0

wall attribute rotation-center 0.0 0.0  5.0 range position-z  5.0
wall attribute rotation-center 0.0 0.0 -5.0 range position-z -5.0
model clean
model cycle 10
model solve elastic ratio-average 5e-3
wall attribute velocity-z 0.0 gradient 0.0 0.0 -0.001
model history name 1 mechanical time-total
wall history name 2 force-contact id 1
wall history name 3 force-contact id 2
model save 'initial'
;---------------------------------------------------------------------------
; ----------- CASE 0: NO TRACKING OF FRAGMENTS - NO BALL RESULTS -------------
;---------------------------------------------------------------------------
model solve time 50.0
model save 'final-case0'
;---------------------------------------------------------------------------
; ----------- CASE 1: CONTINUOUS UPDATE OF FRAGMENTS WITH BALL RESULTS -------
;---------------------------------------------------------------------------
model restore 'initial'
fragment register ball-ball
fragment activate time 0.1
model mechanical time-total 0.0
model result interval mechanical 0.1
model result warn off interval active true
ball result add-attribute fragment active true
wall result add-attribute velocity active true
model solve time 50.0
model save 'final-case1'
;program call 'post_treat.p3fis'
;@make_movie_case1
```

```
;------------------------------------------------------------------------------
; ----------- CASE 2: CUSTOM UPDATE OF FRAGMENTS AND BALL RESULTS ------------
;------------------------------------------------------------------------------
model restore 'initial'
program call 'fracture.p3fis'
@track_init
model solve time 50.0
@track_end
model save 'final-case2'
;program call 'post_treat.p3fis'
;@make_movie_case2
return
;==============================================================================
; eof: fragment.p3dat
```

References

[1]. J. Ai, J.-F. Chen, J. M. Rotter, and J. Y. Ooi. Assessment of rolling resistance models in discrete element simulations. Powder Technology, 206(3):269–282, 2011.

[2]. J. P. Bardet. Experimental Soil Mechanics. Prentice-Hall, Upper Saddle River, New Jersey, 1997.

[3]. P. J. Owen and P. W. Cleary. Prediction of screw conveyor performance using the Discrete Element Method (DEM). Powder Technology, 193(3):274–288, 2009.

[4]. A. Thornton, T. Weinhart, S. Luding, and O. Bokhove. Modeling of particle size segregation: Calibration using the discrete particle method. Int. J. Mod. Phys. C, 23:1240014(08), 2012.

[5]. P. A. Cundall and O. D. L. Strack. A discrete numerical model for granular assemblies. Géotechnique, 29(1):47–65, 1979.

[6]. Nitin Tiwari, Neelima Satyam and Anand J Puppala. Strength and Durability Assessment of Expansive Soil Stabilized with Recycled Ash and Natural Fibers, Journal of Transportation Geotechnics, Elsevier, 29, 100556, 2021.

[7]. Nitin Tiwari and Neelima Satyam**.** Evaluation of Strength and Water Susceptibility Performance of Polypropylene Fiber Reinforced and Silica Fume Modified Hot Mix Asphalt, Advances in Civil Engineering Materials, 10(1), 20210029, 2021.

[8]. Santamarina, J.C., andCho, G.C. Determination of critical state parameters in sandy soils-simple procedure. Geotechnical Testing Journal, 24(2): 185–192, 2001.

[9]. Jamiolkowski, M., LoPresti, D., and Manassero. M. 2001. Evaluation of relative density and shear strength of sands from cone penetration test and flat dilatometer test.In Soil behavior and soft ground construction. GSP 119. ASCE, Reston, Va., 201–238, 2001.

[10]. Robertson, P.K. Cone penetration test (CPT)-based soil behaviour type (SBT) classification system – an update. Canadian Geotechnical Journal,53(12): 1910–1927, 2016.

[11]. Butlanska, J., Arroyo, M., Gens, A., and O'Sullivan, C. Multi-scale analysis of cone penetration test (CPT) in a virtual calibration chamber. Canadian Geotechnical Journal,51(1): 51–66, 2014.

[12]. P. Cundall and O. Strack. A discrete numerical method for granular assemblies. Geotechnique, 29:47-65, 1979.

[13]. P. A. Cundall and R. D. Hart. Numerical modeling of discontinua. Engineering Computations, 9:101-113, 1992.

[14]. I. Omelyan, I. Mryglod, and R. Folk. Symplectic analytically integrable decomposition algorithms: classification, derivation, and application to molecular dynamics, quantum and celestial mechanics simulations. Computer Physics Communications, 151:272-314, 2003.

[15]. R. Gertsch, L. Gertsch, and J. Rostami. Disc cutting tests in Colorado Red Granite: Implications for TBM performance prediction. International Journal of Rock Mechanics and Mining Sciences, 44(2):238{246, 2007.