OpenFDEM Solver

Release 0.0

Grasselli's Geomechanics Group

Jan 02, 2023

CONTENTS:

CHAPTER

ONE

FEATURES AND ABOUT OPENFDEM

OpenFDEM aims to be a free finite and discrete element solver with object-oriented architecture for solving **multiscale, multiphase, and multiphysics** (3M) problems. The applications are, but not limited to, mechanical, thermal and fluid mechanics problems. The main features are:

Modular & Extensible FEM Kernel and DEM Kernel (OPENFDEMlib)

- **Fully extendable and portable** - The kernel is extendable in any "direction". The possibility of adding new element types, new material models with any element type and number of internal history parameters, new boundary conditions (time-dependent, position-dependent, state-dependent, periodic and flow-in/out) or numerical algorithms (explicit and implicit), as well as ability to add and manage arbitrary degrees of freedom.
- **Flexible element definition** - OpenFDEM is not limited only to triangular elements, but allows Q4 cohesive element and triangle-triangle contact. OpenFDEM is a more general FEM/DEM solver can be compatible to arbitrary scenarios.
- **Highly accurate and reliable** – The kernel provides high-order integration schemes and solving methods to seek more reliable numerical results which are comparable to theoretical solutions. The element type has a maximum order of three to rebuild the large deformation within the entity, and the kinematic is able to construct a nonlinear deformation. The Hilber-Hughes-Taylor (HHT) time integration scheme (second order accuracy) is employed for the explicit solver.
- **Friendly preprocessing interface** - The Gmsh API is provided to quickly create meshes from CAD, geometry file and third-party commercial software. The built-in commands are accessible to crate many basic geometries (rectangular, circle, ellipse, polygon, line, particles) and initial discontinuities (single joint, joint sets, Discrete Fracture Networks (DFNs), DFNs from image mapping). The built-in mesh module is able to quickly assess mesh quality and the local bad meshes will be further optimized by swap, node insertion, node delete, element split techniques, automatically or manually.
- **Parallel processing support** - Most modules can be operated in parallel and very good performance scalability can be obtained across various operating platforms.
- **Mesh adaptive analysis support** – Local adaptive mesh refinement (LAMR) and global adaptive mesh refinement (GAMR) are provided for mesh optimization and accuracy enhancement. It supports various error estimations based on different remeshing criteria, support for primary unknown and internal variables mapping, support for high-accuracy internal variable interpolation and fast unbalance equilibrium after refinement. The AMR supports fracture mapping before and after remeshing.
- **Rich grain-based modelling support** – Voronoi tessellations can be created with the built-in Voronoi module. The optimization is deployed to match the laboratorial mineral distribution from measurements or digital image. The realistic GBM can be reproduced directly in the project by inputting the binary sample images, the polygonal element type is available for representing the whole mineral individually, further transgranular fracturing can be realized by element splitting techniques.
- **Large material library including the state-of-the-art models for phase field of quasibrittle materials and rich element library** – currently, OpenFDEM supports 17 element materials spanning elastic, hyperelastic,

plastic, damage, nonlocal, viscous and phasefiled models, supports 7 cohesive materials spanning static, dynamic and fatigue problems. It also supports 6 contact models including Mohr-Coulomb friction, Hertz contact, rate friction, and rough dilation shear law.

- **Advanced Analysis Solvers** - Linear dynamic (implicit and explicit), linear static (using PETSC) and nonlinear dynamic (explicit) are applicable for different problems.

Particle Discrete Element Method (pDEM)

- **Rigid DEM support** - Built in module for rigid particles packing, kinematics and collision, the particle-based contact models include linear, Hertz, cohesive bond and rotation resistance model.
- **Realistic Particle Modelling** - Overlapping particles and Fourier-Voronoi-based algorithm are used to generate realistic particles having complex shapes. The realistic particles can be rigid or deformable, the breakage of the particles are also possible.

Fluid Dynamic Module

- **Analysis Procedures** - Matrix flow for pore seepage, transient incompressible fracture flow, transient compressible fracture flow and gas flow problems.
- **Element Library** - Triangle, quadratic triangle, quadrilateral and quadratic quadrilateral element types are supported for Newtonian fluid and Bingham fluid.
- **Boundary Types** - Water level, pore pressure, flow rate, steady flow and impermeable boundary conditions are supported in hydro module.

Thermal Transportation Module

- **Analysis procedures** - matrix thermal transportation, thermal resistance in fractures, heat conduction of fluid in fracture, heat advection of fluid, heat exchange between solid and fluid and contact thermal problems.
- **Element Library** - triangle, quadratic triangle, quadrilateral and quadratic quadrilateral element types are supported.
- **Boundary Types** - constant temperature, flux, conduction, advection, radiation, source and adiabatic thermal conditions are supported.

Computational Fluid Dynamics

- **Material Point Method (MPM)** is used to simulate the fluid transportation and large deformation. This mesh-free method does not encounter the drawbacks of mesh-based methods (high deformation tangling, advection errors etc.) which makes it a promising and powerful tool for large deformation problems. The coupling among FDEM and MPM makes the solid interacting with fluid is possible.

Post-Processing

- Export to VTK format is supported, allowing to use VTK based visualization tools (such as ParaView) for postprocessing on different operating platforms.
- Export to Tecplot format is supported.
- Export historic variables which are monitored at each step to csv is supported.

Third-Party Packages Used in OpenFDEM

- `GMSH` - 2D and 3D mesh generator
- `GSL` - mathematical routines
- `Eigen` - matrix calculation
- `PETSC` - Portable, Extensible Toolkit for Scientific Computation
- `ParaView` - Parallel Visualization Application (for .vtk files)

1.1 Documentation

The documentation is auto-generated from the `.of` and `.rst` files throughout the codebase and the extensive comments in the source code `.h` and `.of` files. Sphinx is used to compile the documentation in HTML and PDF formats.

CHAPTER

TWO

SYSTEM REQUIREMENTS

2.1 Build Requirements

To compile `OpenFDEM`, you neeed a complier supporting `C++ 17` and the following packages are required:

- **CMake** (>=3.5.1)
- **OpenMP** - a high-performance, freely available package for multi core acceleration
- **Gmsh** - mesh generation and pre-processing, it is optional and the kernel is implemented in the source code (4.10)
- **Eigen** - a scientific matrix computation, it is optional and the headers are included in source code (>=3.4.0)

2.2 Post-Processing

To use the post-processing outputs (optional steps):

- **ParaView** - Parallel visualization application
- **Tecplot** - Commerical software for field results

2.3 Implicit Static/Non-Linear Solvers

To use the implicit static or nonlinear solvers, at least one of the following libraries is required:

- **PETSc** - portable, extensible toolkit for scientific computation
- **LAPACK** - a standard software library for numerical linear algebra

`OpenFDEM` is flexible and can be run on Windows or Linux-like systems. The released version is for Windows x64.

CHAPTER

THREE

QUICK START FOR DEVELOPERS

This is the source code of C/C++ based OpenFDEM project developed by Dr. Xiaofeng Li, since 2017. This project is not limited to an FDEM solver for continuum-discontinuum problems, but is also capable of solving particulate DEM, material point method and phasefield problems.

The project can be complied by Visual Studio (> 2015) and compatible with Windows 7, 10 or Linux-like systems.

Tutorial examples can be found in `..src\test\..`. The main file is located at `src\solve\openfdem.cpp`. OpenFDEM is run by parsing the input file.

General steps to run OpenFDEM models:

1. Open `.sln` project in `\openfdem src\of\OpenFDEM\OpenFDEM.sln` by your local Visual Studio software.
2. Compile the project in Visual Studio from the `src\solve\openfdem.cpp` main file.
3. The executable code should be in `x64\Release`` (or ``Debug`), be sure to keep the `.dll` files in the same folder with `.exe` file.
4. Drag the `.of` file into the `.exe` software then the model will starts to run, or use `openfdem example.of` in terminal to run a model.

3.1 Source Code Download

The source code is hosted on the University of Toronto Gitlab: OpenFDEM Gitlab.

3.2 System Requirements

- Windows x64
- Visual Studio (> 2015)

All external dependencies are included in the OpenFDEM download.

CHAPTER FOUR

COPYRIGHT

```
             ____                   ______ _____  ______ __  __
            / __ \                 |  ____|  __ \|  ____|  \/  |
           | |  | |_ __   ___ _ __ | |__  | |  | | |__  | \  / |
           | |  | | '_ \ / _ \ '_ \|  __| | |  | |  __| | |\/| |
           | |__| | |_) |  __/ | | | |    | |__| | |____| |  | |
            \____/| .__/ \___|_| |_|_|    |_____/|______|_|  |_|
                  | |                     OpenFree Finite Element
                  |_|                         and Discrete Element Method␣
↪Solver

       OpenFDEM : Object Oriented Open Free Finite Discrete Element Code


           Copyright (C) 2017 - 2022   Xiaofeng Li
                        Email: xfli@whrsm.ac.cn

    OpenFDEM Project Website: https://openfdem.com/
        Geomechanics Group Website: https://geogroup.utoronto.ca/

    This library is free software; you can redistribute it and/or
    modify it under the terms of the GNU Lesser General Public
    License as published by the Free Software Foundation; either
    version 2.1 of the License, or any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
    Lesser General Public License for more details.
```

CHAPTER

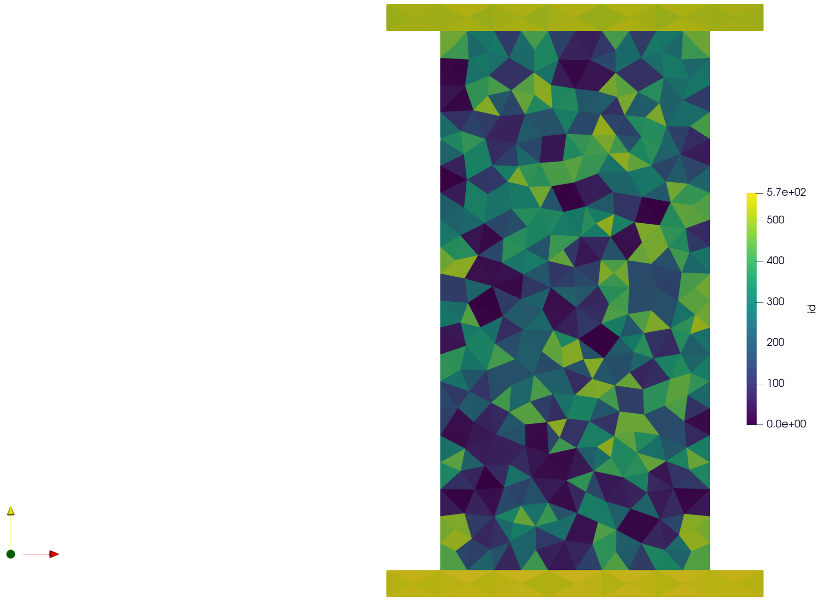
FIVE

TUTORIAL 1: UCS MODEL FROM GMSH

This example will review how to setup a uniaxial compressive test example with output from Gmsh software.

Runtime: <10 min on i9 8-core Windows 10 Machine

Expected tutorial output (visualized in ParaView):



OpenFDEM supports four main mesh pre-processing approaches:

1. A user defined command in OpenFDEM to create mesh automatically
2. Importing a `.geo` file
3. Importing a `.msh` file
4. Mesh developed from other commercial softwares, including `.inp`, `.dxf`, `.fdem`, `.tess` (for grain-based model only) and `.jpg` (for grain-based module and DFN module only).

5.1 Tutorial Prerequistes

The following files are needed to follow along the tutorial:

- example_UCS.geo (click to download from Gitlab)
- example_UCS.msh (click to download from Gitlab)

5.2 Tutorial Steps

OpenFDEM tutorials have the same main steps:

1. Mesh pre-processing steps.
2. Materials definition.
3. Define boundary conditions.
4. Specify the outputs.

A `.log` file is auto-generated by default in the same folder as the input file. To disable the logging, specify this line into the input file:

```
of.debug off
```

5.2.1 Mesh Pre-processing

Create a new empty text file (later add the *.of* extension). Begin writing the following commands:

```
of.new
```

```
# create a domain xmin =-25e-3, xmax =25e-3 , ymin=-50e-3, ymax=50e-3
# The domain is mandatory when the particles or material points are used
of.geometry.domain -25e-3 25e-3  -50e-3  50e-3
```

```
# Create a retangular block, group tag is specimen, the range is xmin =25e-3, xmax␣
↪=25e-3, ymin=-50e-3,ymax= 50e-3
of.geometry.square 'specimen' -25e-3 25e-3  -50e-3  50e-3
```

```
# Create a block for the upper plate
of.geometry.square 'up_plate' -35e-3 35e-3  50e-3  55e-3
```

```
# Create a block for the down plate
of.geometry.square 'down_plate' -35e-3 35e-3  -55e-3  -50e-3
```

After creating geometry entities, you can assign mesh size to the whole model with `all` keyword and it is also possible to assign mesh size to specific entities:

```
''' There are four methods to create mesh, 1- user defined commands in prepprocessing,
↪ it will call gmsh kernel to mesh the geometry
, 2- import .geo file for gmsh, 3- import .msh file (less than V 2.4) and 4- import .
↪inp file from other codes '''
# assign global mesh size, the default keyword is for global entities
of.geometry.mesh.size 'all' 10e-3
```

(continues on next page)

(continued from previous page)

```
# assign specific mesh size to group 'specimen'
of.geometry.mesh.size 'specimen' 5e-3
```

OpenFDEM will call the Gmsh kernel to generate mesh after the mesh size is assigned. The meshing scheme includes delaunay (default), meshadapt and frontal-delaunay . The gmsh interface will be called and you can check the mesh quality, recombine the mesh or change mesh size in gmsh pannel.

```
# starts to mesh, delaunay is optional, it is the default value
of.geometry.mesh delaunay
```

Cohesive elements can be inserted after importing the mesh. OpenFDEM supports to partially insert cohesive elements and also supports inserting extrinsic cohesive elements.

```
#insert cohesive elements, it is aviable to insert CZM in the whole model or to a␣
↪specific entity
of.mesh.insert 'specimen'
```

5.2.2 Materials Definition

Material parameters contain three parts: - The parameter for solid matrix - Solid cohesive elements and - Contacts

The paramters will be allocated to the user-defined element groups.

```
# assign  material parameters to solid elements based on the element groups
of.mat.element 'specimen' ELASTIC den 2700 E 30e9 v 0.3 damp 0.6
of.mat.element 'up_plate' ELASTIC den 2700  E 70e9 v 0.2 damp 0.9
of.mat.element 'down_plate' ELASTIC den 2700 E 70e9 v 0.2 damp 0.9

#assign material parameters to cohesive elements, the default is a reserved keyword␣
↪means the whole entities
# user are not allowed to use this keyword for the group tags
of.mat.cohesive 'all' EM ten 1e6 coh 3e6 fric 0.3 GI 10 GII 50 beta_I 0 beta_II 0

#assign material parameters to contact
of.mat.contact 'all' MC fric 0.3
```

5.2.3 Define Boundary Conditions

The boundary conditions are defined by the entity groups.

```
#create up_plate and up_plate nodal physical groups from element groups
# the element physcial groups are inherited from the msh file in geometry
of.group.nodal.from.element 'up_plate' 'up_plate'
of.group.nodal.from.element 'down_plate' 'down_plate'

#assign Dirichlet boundaries, the fixed velocities in X and y directions are added to␣
↪the
# nodal groups
of.boundary.nodal.velocity  'up_plate' XY 0.0 -0.05
of.boundary.nodal.velocity  'down_plate' XY 0.0 0.05
```

5.2.4 Post-Processing Settings

```
#set for post-processing; how often to output the results + variables (of.history.all␣
↪would export all variables, but large file....)
# interval to write history
of.history.interval 10
# interval to write paraview field results
of.history.pv.interval 5000
'''Export field variables, default is  to export all variables, the user can choose␣
↪to export specific variables using keywords'''
of.history.pv.field all
of.history.pv.fracture all

# monitor the average nodal displacemnt in upper plate every step
of.history.nodal.group.dis 1 'up_plate'
# monitor the average stress (tensor) in the specimen
of.history.element.group.stress 2 'specimen'
```

5.2.5 Run models

Finally, define the number of time-steps:

```
# total run steps
of.step 150000
```

5.2.6 Full Tutorial Script

To run the model, save your text file with the *.of* extension. Rebuild the openfdem solution and drag your *.of* file into the *OpenFDEM.exe*. It will automatically run and save the outputs.

Complete script below:

```
of.geometry.square 'specimen' -25e-3 25e-3  -50e-3  50e-3
# Create a block for the upper plate
of.geometry.square 'up_plate' -35e-3 35e-3  50e-3  55e-3
# Create a block for the down plate
of.geometry.square 'down_plate' -35e-3 35e-3  -55e-3  -50e-3

''' There are four methods to create mesh, 1- user defined commands in prepprocessing,
↪ it will call gmsh kernel to mesh the geometry
, 2- import .geo file for gmsh, 3- import .msh file (less than V 2.4) and 4- import .
↪inp file from other codes '''
# assign global mesh size, the default keyword is for global entities
of.geometry.mesh.size 'all' 10e-3
# assign specific mesh size to group 'specimen'
of.geometry.mesh.size 'specimen' 5e-3

# starts to mesh, delaunay is optional, it is the default value
of.geometry.mesh delaunay

#insert cohesive elements, it is aviable to insert CZM in the whole model or to a␣
↪specific entity
of.mesh.insert 'specimen'
```

(continues on next page)

(continued from previous page)

```
# assign  material parameters to solid elements based on the element groups
of.mat.element 'specimen' ELASTIC den 2700 E 30e9 v 0.3 damp 0.6
of.mat.element 'up_plate' ELASTIC den 2700  E 70e9 v 0.2 damp 0.9
of.mat.element 'down_plate' ELASTIC den 2700 E 70e9 v 0.2 damp 0.9

#assign material parameters to cohesive elements, the default is a reserved keyword␣
↪means the whole entities
# user are not allowed to use this keyword for the group tags
of.mat.cohesive 'all' EM ten 1e6 coh 3e6 fric 0.3 GI 10 GII 50 beta_I 0 beta_II 0

#assign material parameters to contact
of.mat.contact 'all' MC fric 0.3

#create up_plate and up_plate nodal physical groups from element groups
# the element physcial groups are inherited from the msh file in geometry
of.group.nodal.from.element 'up_plate' 'up_plate'
of.group.nodal.from.element 'down_plate' 'down_plate'

#assign Dirichlet boundaries, the fixed velocities in X and y directions are added to␣
↪the
# nodal groups
of.boundary.nodal.velocity  'up_plate' XY 0.0 -0.05
of.boundary.nodal.velocity  'down_plate' XY 0.0 0.05

#assign global damping value, the value should not be over 1.0
# the default value is 0.7
of.damp.global 0.5

#set for post-processing; how often to output the results + variables (of.history.all␣
↪would export all variables, but large file....)
# interval to write history
of.history.interval 10
# interval to write paraview field results
of.history.pv.interval 5000
'''Export field variables, default is  to export all variables, the user can choose␣
↪to export specific variables using keywords'''
of.history.pv.field all
of.history.pv.fracture all

# monitor the average nodal displacemnt in upper plate every step
of.history.nodal.group.dis 1 'up_plate'
# monitor the average stress (tensor) in the specimen
of.history.element.group.stress 2 'specimen'

# total run steps
of.step 150000
```

CHAPTER

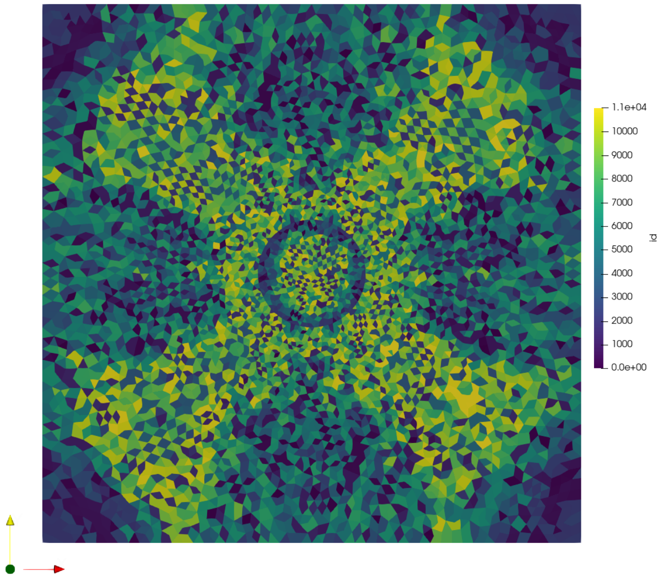
SIX

TUTORIAL 2: IN SITU STRESS TUTORIAL

This example will review how to setup an insitu excavation example, defining the mesh in the *.of* file.

Runtime: <10 min on i9 8-core Windows 10 Machine

Expected tutorial output (visualized in ParaView):



6.1 Tutorial Resources

The result mesh file will be created during the tutorial, but can also be viewed on Gitlab: mesh.msh.

6.2 Tutorial Steps

6.2.1 Mesh Pre-Processing

In a new text-file, write the following commands. These will define the geometry as a square with a circle in the center for excavation:

```
# Create a retangular block, group tag is rock, the range is xmin =0, xmax =1, ymin=0,
↪ymax= 1
of.geometry.square 'rock' 0 1 0 1
# Create a hole in the block, cut is to fragment the block and seprate the rock block␣
↪to hole block and
# new rock block (out of the hole block)
of.geometry.cut.circle 'hole' 'rock' 0.5 0.5 0.1 70
# assign global mesh size, the default keyword is for global entities
of.geometry.mesh.size 'all' 0.02
# starts to mesh using auto method, delaunay the default value
of.geometry.mesh auto
```

```
#create excavation element group using cricle tool, based on the centric point and␣
↪radius
of.group.element.circle 'excavation' 0.5 0.5 0.1
```

6.2.2 Materials Definition

Assign the following material properties (density, Young’s modulus and damping coefficient):

```
# assign  material parameters to solid elements based on the element groups
of.mat.element 'all' ELASTIC den 2000 E 30e9 v 0.3 damp 0.9
of.mat.contact 'all' MC fric 0.3
```

6.2.3 Define Boundary Conditions

```
# assign the in-situ stress in the domain
of.boundary.element.stress -35e6 0.0 -35e6

# create nodal groups on edges
of.group.nodal.plane 'bottom_edge' 0.0 0.0 1.0 0.0
of.group.nodal.plane 'up_edge' 0.0 1.0 1.0 1.0
of.group.nodal.plane 'left_edge' 0.0 0.0 0.0 1.0
of.group.nodal.plane 'right_edge' 1.0 0.0 1.0 1.0
```

6.2.4 Run Model, Set Paraview Parameters

Define parameters for paraview export:

```
# set interval to write paraview field results
of.history.pv.interval 500
of.history.pv.field all
of.history.pv.fracture all
of.history.pv.cohesive all

# run steps to equlibrum the in-situ stress, it should be fast
of.step 1000
```

6.2.5 Set Tunnel Excavation

```
# excavte the tunnel
of.boundary.excavation 'excavation'
# fix the outer boundaries
of.boundary.nodal.velocity 'bottom' Y 0
of.boundary.nodal.velocity 'up' Y 0
of.boundary.nodal.velocity 'left' X 0
of.boundary.nodal.velocity 'right' X 0
```

6.2.6 Execute Model

Define the number of model time-steps:

```
of.step 50000
# terminate the run and step out solver
of.stop
```

6.2.7 Full Tutorial Script

To run the model, save your text file with the *.of* extension. Rebuild the openfdem solution and drag your *.of* file into the *OpenFDEM.exe*. It will automatically run and save the outputs.

```
# Create a retangular block, group tag is rock, the range is xmin =0, xmax =1, ymin=0,
↪ymax= 1
of.geometry.square 'rock' 0 1 0 1
# Create a hole in the block, cut is to fragment the block and seprate the rock block␣
↪to hole block and
# new rock block (out of the hole block)
of.geometry.cut.circle 'hole' 'rock' 0.5 0.5 0.1 70
# assign global mesh size, the default keyword is for global entities
of.geometry.mesh.size 'all' 0.02
# starts to mesh using auto method, delaunay the default value
of.geometry.mesh auto

#create excavation element group using cricle tool, based on the centric point and␣
↪radius
of.group.element.circle 'excavation' 0.5 0.5 0.1
```

(continues on next page)

(continued from previous page)

```
# assign  material parameters to solid elements based on the element groups
of.mat.element 'all' ELASTIC den 2000 E 30e9 v 0.3 damp 0.9
of.mat.contact 'all' MC fric 0.3

# assign the in-situ stress in the domain
of.boundary.element.stress -35e6 0.0 -35e6

# create nodal groups on edges
of.group.nodal.plane 'bottom_edge' 0.0 0.0 1.0 0.0
of.group.nodal.plane 'up_edge' 0.0 1.0 1.0 1.0
of.group.nodal.plane 'left_edge' 0.0 0.0 0.0 1.0
of.group.nodal.plane 'right_edge' 1.0 0.0 1.0 1.0

# set interval to write paraview field results
of.history.pv.interval 500
of.history.pv.field all
of.history.pv.fracture all
of.history.pv.cohesive all

# run steps to equlibrum the in-situ stress, it should be fast
of.step 1000

# insert cohesive elements, the cohesive elements should be inserted fater in-situ␣
↪stress
of.mesh.insert 'all'
# assign material parameters for cohesive elements
of.mat.cohesive 'all' EM ten 10e6 coh 20e6 fric 0.3 GI 20 GII 40

# excavte the tunnel
of.boundary.excavation 'excavation'
# fix the outer boundaries
of.boundary.nodal.velocity 'bottom' Y 0
of.boundary.nodal.velocity 'up' Y 0
of.boundary.nodal.velocity 'left' X 0
of.boundary.nodal.velocity 'right' X 0

of.step 50000
# terminate the run and step out solver
of.stop
```

CHAPTER

SEVEN

TUTORIAL 3: PRESSURE METHOD

Runtime: ~2 hours on i9 8-core Windows 10 Machine

7.1 Full Tutorial Script

To run the model, save your text file with the *.of* extension. Rebuild the openfdem solution and drag your *.of* file into the *OpenFDEM.exe*. It will automatically run and save the outputs.

```
# clear old memories, be optional
of.new

# Create a retangular block, group tag is rock, the range is xmin =0, xmax =1, ymin=0,
↪ymax= 1
of.geometry.square 'rock' 0 1 0 1
# Create a hole in the block, cut is to fragment the block and seprate the rock block␣
↪to hole block and
# new rock block (out of the hole block)
of.geometry.cut.circle 'hole' 'rock' 0.5 0.5 0.1 70
# assign global mesh size, the default keyword is for global entities
of.geometry.mesh.size 'all' 0.02
# starts to mesh, delaunay is optional, it is the default value
of.geometry.mesh delaunay

#insert cohesive elements, globally
of.mesh.insert 'all'

# assign  material parameters to solid elements based on the element groups
of.mat.element 'all' elastic den 2000 E 30e9 v 0.3 damp 0.9
of.mat.cohesive 'all' EM ten 10e6 coh 20e6 fric 0.3 GI 20 GII 40
of.mat.contact 'all' MC fric 0.3

#create excavation element group using cricle tool, based on the centric point and␣
↪radius
of.group.element.circle 'excavation' 0.5 0.5 0.1
#create bottom_edge nodal group using plane tool, based on the start point coord and␣
↪end point coord
of.group.nodal.plane 'bottom_edge' 0.0 0.0 1.0 0.0
#create up_edge nodal group using plane tool, based on the start point coord and end␣
↪point coord
of.group.nodal.plane 'up_edge' 0.0 1.0 1.0 1.0
#create left_edge nodal group using plane tool, based on the start point coord and␣
↪end point coord
of.group.nodal.plane 'left_edge' 0.0 0.0 0.0 1.0
```

(continues on next page)

(continued from previous page)

```
#create right_edge nodal group using plane tool, based on the start point coord and end point coord
of.group.nodal.plane 'right_edge' 1.0 0.0 1.0 1.0

# set interval to write paraview field results
of.history.pv.interval 500
of.history.pv.field all
of.history.pv.fracture all
of.history.pv.cohesive all

# assign pressure boundary on edges, comprssion is postive
of.boundary.pressure.normal 'bottom_edge' 15e6
of.boundary.pressure.normal 'up_edge' 15e6
of.boundary.pressure.normal 'left_edge' 5e6
of.boundary.pressure.normal 'right_edge' 5e6

# total run steps to equlibrium the insitu stress
of.step 100000

# excavted the hole
of.boundary.excavation 'excavation'
# run steps to compute the tunnel deformation
of.step 500000
# terminate the run and step out solver
of.stop
```

OpenFDEM input files use a Python-like format and the dot is used to clarify the hierarchy of different classes.

The first layer is OpenFDEM, declared as `.of` or `.OpenFDEM`, but the commands in other layers cannot be shortened (for example, of.nodal or OpenFDEM.nodal).

Table of Contents

  - *of.geometry.circle name x0 y0 r n*
  - *of.geometry.ellipse name x0 y0 rmax rmin theta n*
  - *of.geometry.cut.square name name2 x0 y0 x1 y1 x2 y2 x3 y3*
  - *of.geometry.cut.polygon name name2 n x0 y0 x1 y1 x2 y2 x3 y3*
  - *of.geometry.cut.table name name2 'table.tab'*
  - *of.geometry.cut.circle name name2 x0 y0 r n*
  - *of.geometry.cut.ellipse name name2 x0 y0 rmax rmin theta n*
  - *of.geometry.cut.jset name name2 dip length space gap*
  - *of.geometry.cut.dfn name name2 method dip length*
  - *of.geometry.import.rdfn pixel_ratio name name2 name*
  - *of.geometry.remove.square name name2 x0 y0 x1 y1 x2 y2 x3 y3*
  - *of.geometry.remove.polygon name name2 n x0 y0 x1 y1 x2 y2 x3 y3*
  - *of.geometry.remove.table name name2 'table.tab'*
  - *of.geometry.remove.circle name name2 x1 y1 r n*
  - *of.geometry.remove.ellipse name name2 x0 y0 rmax rmin theta n*
  - *of.geometry.mesh.size name default*
  - *of.geometry.mesh keyword*
- *mesh_insert*
  - *of.mesh.insert elementgroup*
- *nodal*
  - *of.nodal.coord number_nodes*
- *element*
  - *of.element.connectivity number_nodes number_elements*
- *contact*
  - *of.contact.detection nbs*
  - *of.contact.force lig*
- *cohelement*
  - *of.cohelement.delete cohelementgroup*
- *group*
  - *nodal groups*
  - *bool*
  - *element*
  - *cohesive element*
- *boundary*
  - *nodal*

  - *nodal local edge*
  - *nodal table*
  - *hydro*
  - *blast*
  - *element*
  - *thermal*
- *material*
  - *of.mat.element elementgroupname(all) modelname p1 p2 p3 . . .*
  - *of.mat.particle elementgroupname(all) modelname p1 p2 p3 . . .*
  - *of.mat.cohesive cohelementgroupname(all) modelname p1 p2 p3 . . .*
  - *of.mat. cohesive 'rock' em_het power 0.5 dip 35 ten 30e6 3e6 coh 30e6 3e6 fric 0.3 0 gi 100 20 gii 300 30 (power, dip, mean pn, dev pn, mean pt, dev pt, mean ten, dev ten, mean coh, dev coh, mean fri, dev fri, mean gi, dev gi, mean gii, dev gii)*
  - *of.mat.contact elementgroupname1 elementgroupname2(all) modelname p1 p2 p3 . . .*
- *gbm*
  - *of.gbm numberofminerals m1_name ratio m2_name ratio . . .*
- *history*
  - *of.history.nodal.force id x1 y1*
  - *of.history.nodal.vel id x1 y1*
  - *of.history.nodal.dis id x1 y1*
  - *of.history.nodal.fluid.pressure id x1 y1*
  - *of.history.nodal.fracture.pressure id x1 y1*
  - *of.history.nodal.matrix.pressure id x1 y1*
  - *of.history.nodal.temperature id x1 y1*
  - *of.history.nodal.group.force id groupname*
  - *of.history.nodal.group.vel id groupname*
  - *of.history.nodal.group.dis id groupname*
  - *of.history.nodal.group.fluid.pressure id groupname*
  - *of.history.nodal.group.fracture.pressure id groupname*
  - *of.history.nodal.group.matrix.pressure id groupname*
  - *of.history.nodal.group.temperature id groupname*
  - *of.history.element.stress id x1 y1*
  - *of.history.element.strain id x1 y1*
  - *of.history.element.strainrate id x1 y1*
  - *of.history.element.group.stress id groupname*
  - *of.history.element.group.strain id groupname*

CHAPTER

EIGHT

GENERAL COMMANDS

8.1 of.new

meaning [Optional]: Clear all variables from your last computation (i.e. start of a new OpenFDEM run).

demo:

```
of.new
```

8.2 of.restore

meaning [Optional]: call the saved binary computation results.

demo:

```
of.restore “filename.sav”
```

8.3 of.result

meaning [Optional]: assign the results folder, default value is `../result`.

demo:

```
of.result “c:/user/openfdem example/result”
```

8.4 of.save

meaning: save the binary computation results.

demo:

```
of.save “c:/user/openfdem example/result/insitu.sav”
```

8.5 of.import

meaning: import files, currently supports all formats of `.inp` files, `.msh` files (no more than 2.0 version), `.msh2` files, `.geo` files and `.ofdem` files (default format for OpenFDEM).

OpenFDEM read all information from your mesh, including nodal physical, group, element physical group.

OpenFDEM 4.6+ version can read `.tess` and `.stl`

demo:

```
of.import “c:/user/openfdem example/result/ucs.msh”
```

8.6 of.debug on

meaning [Optional]: turn on the debug mode, the default is off. Debugging will increase the run-time but a `.log` file will be generated when this mode is on.

demo:

```
of.debug on or of.debug off
```

CHAPTER

NINE

CONFIG

9.1 of.config.type planestress/planestrain

meaning: choosing plane stress or plane strain, the default is plane stress

demo:

```
of.config.type planestrain
```

9.2 of.config.module hydro/blast/thermal

meaning: choosing mechanical module, the default is mechanical

demo:

```
of.config.module hydro
of.config.module blast
of.config.module thermal
```

CHAPTER

TEN

GEOMETRY

You can create meshes based on the built-in command line functions. OpenFDEM also supports creating `retangular`, `circle`, `ellipse`, `arbitary polygon` from tables. The bool operators support `intersection`, `union`, `difference`, and `fragment` between different entities.

10.1 of.geometry.square name x0 y0 x1 y1 x2 y2 x3 y3

meaning: create a geometry using square range, and the geometry tag is ‘name’

demo:

```
of.geometry.square ‘rock1’ 0.0 1.0 1.0 2.0 2.0 3.0 4.0 2.0
```

10.2 of.geometry.polygon name n x0 y0 x1 y1 x2 y2 x3 y3

meaning: create a geometry using polygon range, the geometry tag is ‘name’ and the vertex count is n

demo:

```
of.geometry.polygon ‘rock1’ 5 0.0 1.0 1.0 2.0 2.0 3.0 4.0 2.0 3.0 5.2
```

10.3 of.geometry.table name ‘table.tab’

meaning: create a geometry. All point cords are put in the table file.

table format: x0, y0 x1, y1 x2, y2 . . .

demo:

```
of.geometry.table ‘rock1’ ‘tablerock.tab’
```

10.4 of.geometry.circle name x0 y0 r n

meaning: create a geometry using circle range, the geometry tag is ‘name’. The centric point is (x1, y1), radius is r and edge count is n, deafault is 50.

demo:

```
of.geometry.circle ‘tunnel’ 0.0 0.0 1.0 20
```

10.5 of.geometry.ellipse name x0 y0 rmax rmin theta n

meaning: create a geometry using ellipse range, where the geometry tag is ‘name’ and the centric point is (x0, y0). Radiuses are rmax and rmin, and edge count is n, default is 50.

demo:

```
of.geometry.ellipse ‘ellipsetunnel’ 0.0 0.0 1.0 0.5 20
```

10.6 of.geometry.cut.square name name2 x0 y0 x1 y1 x2 y2 x3 y3

meaning: cut a rectangular tunnel in existing *‘name2’* group geometry, but the *‘name’* group is not excavated.

demo:

```
of.geometry.cut.square ‘square_tunnel’ ‘rock’ 0.0 1.0 1.0 2.0 2.0 3.0 4.0 2.0
```

10.7 of.geometry.cut.polygon name name2 n x0 y0 x1 y1 x2 y2 x3 y3

meaning: cut a polygon tunnel in existing *‘name2’* group geometry, but the *‘name’* group is not excavated.

demo:

```
of.geometry.cut.polygon ‘polygon_tunnel’ ‘rock’ 4 0.0 1.0 1.0 2.0 2.0 3.0 4.0 2.0
```

10.8 of.geometry.cut.table name name2 ‘table.tab’

meaning: cut a table tunnel in existing ‘name2’ group geometry, but the ‘name’ group is not excavated.

demo:

```
of.geometry.cut.table ‘tunnel_from_table’ ‘rock’ ‘tunneltable.tab’
```

10.9 of.geometry.cut.circle name name2 x0 y0 r n

meaning: cut a circular tunnel in existing ‘name2’ group geometry, but the ‘name’ group is not excavated.

demo:

```
of.geometry.cut.circle ‘circle_tunnel’ ‘rock’ 0.0 0.0 1.0 20
```

10.10 of.geometry.cut.ellipse name name2 x0 y0 rmax rmin theta n

meaning: cut a elliptical tunnel in existing ‘name2’ group geometry, but the ‘name’ group is not excavated.

demo:

```
of.geometry.cut.ellipse ‘ellipse_tunnel’ ‘rock’ 0.0 0.0 1.0 0.5 20
```

10.11 of.geometry.cut.jset name name2 dip length space gap

meaning: create jset ‘name2’ group geometry in ‘name’ group

method: default, normal, uniform, exponential, log_normal, power

cdip ndip udip edip logdip pdip

dip: dip udip ndip

length: length nlength flength plength

space: space nspace

gap: gap ngap

demo:

```
of.geometry.cut.jset 'jset-1' 'rock_0' n_dip 60 20 n_space 0.2 0.1
```

10.12 of.geometry.cut.dfn name name2 method dip length

method: count, p21(length/area), p10 (n/edge length)

dip: dip udip ndip

length: length nlength flength

demo:

```
of.geometry.cut.dfn 'dfn-1' 'rock_0' p10 3 n_dip 60 5 n_length 0.3 0.1
```

10.13 of.geometry.import.rdfn pixel_ratio name name2 name

demo:

```
of.geometry.import.rdfn 1.0 'dfn-1' 'rock_0' 'fault.dat'
```

10.14 of.geometry.remove.square name name2 x0 y0 x1 y1 x2 y2 x3 y3

meaning: remove a rectangular tunnel in existing ‘name2’ group geometry, the ‘name’ group is excavated as null

demo:

```
of.geometry.remove.square ‘rock2’ ‘rock’ 0.0 1.0 1.0 2.0 2.0 3.0 4.0 2.0
```

10.15 of.geometry.remove.polygon name name2 n x0 y0 x1 y1 x2 y2 x3 y3

meaning: remove a polygon tunnel in existing ‘name2’ group geometry, the ‘name’ group is excavated as null

demo:

```
of.geometry.remove.polygon ‘rock2’ ‘rock’ 0.0 1.0 1.0 2.0 2.0 3.0 4.0 2.0
```

10.16 of.geometry.remove.table name name2 ‘table.tab’

meaning: remove a table tunnel in existing ‘name2’ group geometry, the ‘name’ group is excavated as null

demo:

```
of.geometry.remove.table ‘rock2’ ‘rock’ 0.0 1.0 1.0 2.0 2.0 3.0 4.0 2.0
```

10.17 of.geometry.remove.circle name name2 x1 y1 r n

meaning: remove a circle tunnel in existing ‘name2’ group geometry, the ‘name’ group is excavated as null

demo:

```
of.geometry.remove.circle ‘rock2’ ‘rock’ 0.0 1.0 1.0 2.0 2.0 3.0 4.0 2.0
```

10.18 of.geometry.remove.ellipse name name2 x0 y0 rmax rmin theta n

meaning: remove a ellipse tunnel in existing ‘name2’ group geometry, the ‘name’ group is excavated as null

demo:

```
of.geometry.remove.ellipse ‘rock2’ ‘rock’ 0.0 1.0 1.0 2.0 2.0 3.0 4.0 2.0
```

10.19 of.geometry.mesh.size name default

meaning: assign the mesh size to corresponding geometry groups, ‘all’ means the all geometries

demo:

```
of.geometry.mesh.size 'all' 0.2
```

10.20 of.geometry.mesh keyword

meaning: method used to generate mesh, default is delaunay

keyword: meshadapt(default), delaunay, frontal-delaunay

demo:

```
of.geometry.mesh meshasapt
```

CHAPTER

ELEVEN

MESH_INSERT

11.1 of.mesh.insert elementgroup

meaning: insert the CZMs for specific element groups, ‘all’ means insert CZMs in all model

demo:

```
of.mesh.insert ‘rock’

of.mesh.insert ‘all’
```

CHAPTER

TWELVE

NODAL

12.1 of.nodal.coord number_nodes

meaning: input nodes for the .fdem format

demo:

```
of.nodal.coord 3

  0.1 0.3

  0.2 0.6

  0.8 0.2
```

CHAPTER

THIRTEEN

ELEMENT

13.1 of.element.connectivity number_nodes number_elements

x1 y1 x2 y2 x3 y3…

meaning: input elements for the .fdem format, in anticlockwise order

demo:

```
of.element.connectivity 3 3

  0 3 15

  1 2 7

  2 4 6
```

CHAPTER

FOURTEEN

CONTACT

14.1 of.contact.detection nbs

meaning: the contact detection method, current methods are: Munjiza model, NBS, modified NBS and Li-Grasselli's method [default].

demo:

```
of.contact. detection nbs
```

14.2 of.contact.force lig

meaning: the contact force method, current methods are: Munjiza model, and Li-Grasselli's method [default].

demo:

```
of.contact.force lig
```

CHAPTER

FIFTEEN

COHELEMENT

15.1 of.cohelement.delete cohelementgroup

CHAPTER

SIXTEEN

GROUP

16.1 nodal groups

16.1.1 of.group.nodal.square nodalgroupname x1 x2 y1 y2

meaning: create nodal group using rectangular range

demo:

```
of.group.nodal.square 'rock' -50.0 50.0 -50.0 10.0
```

16.1.2 of.group.nodal.circle nodalgroupname x1 y1 r

meaning: create nodal group using circle range, the nodes within the circle

demo:

```
of.group.nodal.circle 'tunnel' 0.0 0.0 2.5
```

16.1.3 of.group.nodal.circle.outer nodalgroupname x1 y1 r

meaning: create nodal group out of the circle range

demo:

```
of.group.nodal.circle.outer 'tunnel' 0.0 0.0 2.5
```

16.1.4 of.group.nodal.circle.on nodalgroupname x1 y1 r

meaning: create nodal group on of the circle boundary

demo:

```
of.group.nodal.circle.on 'tunnel' 0.0 0.0 2.5
```

16.1.5 of.group.nodal.plane.left nodalgroupname x1 y1 x2 y2

meaning: create nodal group on the left of the predefined plane

demo:

```
of.group.nodal.plane.left ‘slope’ 0.0 0.0 1.0 2.7
```

16.1.6 of.group.nodal.plane.right nodalgroupname x1 y1 x2 y2

meaning: create nodal group on the right of the plane

demo:

```
of.group.nodal.plane.right ‘slope’ 0.0 0.0 1.0 2.7
```

16.1.7 of.group.nodal.plane nodalgroupname x1 y1 x2 y2

meaning: create nodal group on the plane

demo:

```
of.group.nodal.plane ‘slope’ 0.0 0.0 1.0 2.7
```

16.1.8 of.group.nodal.from.element ‘up’ ‘up’

meaning: create nodal group from the element group

demo:

```
of.group.nodal.from.element ‘rock’ ‘rock’
```

16.2 bool

16.2.1 of.group.nodal.bool.union nodalgroupnename nodalgroupnamea nodalgroupnameb

meaning: create nodal group using bool union

demo:

```
of.group.nodal.bool.union ‘rock1’ ‘rock2’
```

16.2.2 of.group.nodal.bool.intersect nodalgroupnewname nodalgroupnamea nodalgroupnameb

meaning: create nodal group using bool intersect

demo:

```
of.group.nodal.bool.intersect ‘rock1’ ‘rock2’
```

16.2.3 of.group.nodal.bool.subtract nodalgroupnewname nodalgroupnamea nodalgroupnameb a-b

meaning: create nodal group using bool subtract

demo:

```
of.group.nodal.bool.subtract‘rock1’ ‘rock2’
```

16.3 element

16.3.1 of.group.element.square elementgroupname x1 x2 y1 y2

demo:

```
of.group.element.square ‘rock’ -50.0 50.0 -50.0 10.0
```

16.3.2 of.group.element.circle elementgroupname x1 y1 r

demo:

```
of.group.element. circle ‘tunnel’ 0.0 0.0 2.5
```

16.3.3 of.group.element.circle.outer elementgroupname x1 y1 r

16.3.4 of.group.element.plane.left elementgroupname x1 y1 x2 y2

demo:

```
of.group.element. .left ‘slope’ 0.0 0.0 1.0 2.7
```

16.3.5 of.group.element.plane.right elementgroupname x1 y1 x2 y2

16.3.6 of.group.element.bool.union elementgroupnewname elementgroupnamea elementgroupnameb

16.3.7 of.group. element.bool.intersect elementgroupnewname elementgroupnamea elementgroupnameb

16.3.8 of.group. element.bool.subtract elementgroupnewname elementgroupnamea elementgroupnameb a-b

16.4 cohesive element

16.4.1 of.group.cohelement.square cohelementgroupname x1 x2 y1 y2

demo:

```
of.group. cohelement.square ‘rock’ -50.0 50.0 -50.0 10.0
```

16.4.2 of.group.cohelement.circle cohelementgroupname x1 y1 r (default)

16.4.3 of.group.cohelement.circle.outer cohelementgroupname x1 y1 r

16.4.4 of.group.cohelement.circle.on cohelementgroupname x1 y1 r

16.4.5 of.group.cohelement.plane.left cohelementgroupname x1 y1 x2 y2 (default)

16.4.6 of.group.cohelement.plane.right cohelementgroupname x1 y1 x2 y2

16.4.7 of.group.cohelement.plane cohelementgroupname x1 y1 x2 y2 (default)

16.4.8 of.group.cohelement.gbm cohelementgroupname elementgroupname1 elementgroupname2

demo:

```
of.group.cohelement.gbm 'qtz-qtz' 'qtz' 'qtz' # quartz group
```

16.4.9 of.group.cohelement.dfn cohelementgroupname dfnname

demo:

```
of.group.cohelement.dfn 'dfn1' 'dfn1'
```

16.4.10 of.group.cohelement.bool.union cohelementgroupnewname cohelementgroupnamea cohelementgroupnameb

16.4.11 of.group. cohelement.bool.intersect cohelementgroupnewname cohelementgroupnamea cohelementgroupnameb

16.4.12 of.group. cohelement.bool.subtract cohelementgroupnewname cohelementgroupnamea elementgroupnameb a-b

CHAPTER

SEVENTEEN

BOUNDARY

17.1 nodal

17.1.1 of.boundary.nodal.force nodalgroupname xy force_x force_y

meaning: assign force boundary to nodal groups

demo:

```
of.boundary.nodal.force ‘rock’ xy -50.0 27.0
of.boundary.nodal.force ‘rock’ x -50.0
of.boundary.nodal.force ‘rock’ y 27.0
```

17.1.2 of.boundary.nodal.velocity nodalgroupname xy vel_x vel_y

meaning: assign normal velocity boundary to nodal groups (in local coords)

17.1.3 of.boundary.nodal.inivelocity nodalgroupname xy vel_x vel_y

meaning: assign initial velocity boundary to nodal groups (in local coords)

17.1.4 of.boundary.nodal.acceleration nodalgroupname xy acc_x acc_y

meaning: assign aceleration boundary to nodal groups (in local coords)

17.1.5 of.boundary.nodal.viscous nodalgroupnamexy viscous_x viscous_y

demo:

```
of.boundary.nodal.viscous ‘rock’ xy
of.boundary.nodal.viscous ‘rock’ x
of.boundary.nodal.viscous ‘rock’ y
```

17.1.6 of.boundary.nodal.viscous nodalgroupnamexy viscous_x viscous_y

17.2 nodal local edge

17.2.1 of.boundary.nodal.force.local nodalgroupname xy force_normal force_shear

17.2.2 of.boundary.nodal.velocity.local nodalgroupname xy vel_ normal vel_ shear

17.2.3 of.boundary.nodal.acceleration.local nodalgroupname xy acc_ normal acc_ shear

demo:

```
of.boundary.nodal.force.local ‘rock’ xy -50.0 27.0
of.boundary.nodal.force.local ‘rock’ x -50.0
of.boundary.nodal.force.local ‘rock’ y 27.0
```

17.2.4 of.boundary.nodal.force.local.table nodalgroupname xy force_normal force_shear tablename

17.2.5 of.boundary.nodal.velocity. table nodalgroupname xy vel_ normal vel_ shear tablename

17.2.6 of.boundary.nodal.acceleration. table nodalgroupname xy acc_ normal acc_ shear tablename

demo:

```
of.boundary.nodal.force.local ‘rock’ xy -50.0 27.0

  of.boundary.nodal.force.local ‘rock’ x -50.0

  of.boundary.nodal.force.local ‘rock’ y 27.0
```

17.3 nodal table

17.3.1 of.boundary.nodal.force.table nodalgroupname xy force_x force_y tablename

demo:

```
of.boundary.nodal.force.table ‘rock’ xy -50.0 27.0 blast
of.boundary.nodal.force.local ‘rock’ x -50.0
of.boundary.nodal.force.local ‘rock’ y 27.0
```

17.3.2 of.boundary.nodal.velocity.table nodalgroupname xy vel_x vel_y tablename tablename

17.3.3 of.boundary.nodal.acceleration.table nodalgroupname xy acc_x acc_y tablename tablename

17.3.4 of.boundary.pressure.normal nodalgroupname value

17.3.5 of.boundary.pressure.shear nodalgroupname value

demo:

```
of.boundary.pressure .normal‘rock’ -50.0
of.boundary.pressure.shear ‘rock’ -50.0
```

17.3.6 of.boundary.pressure.normal.table nodalgroupname value

17.3.7 of.boundary.pressure.shear.table nodalgroupname value

demo:

```
of.boundary.pressure.normal.table ‘rock’ -50.0 gas
```

17.4 hydro

17.4.1 of.boundary.hydro.porepressure nodalgroup p p_x p_y

17.4.2 of.boundary.hydro.waterlevel nodalgroup water_y initial_p

17.4.3 of.boundary.hydro.pressure nodalgroup value

17.4.4 of.boundary.hydro.pressure.table nodalgroup value tablename

17.4.5 of.boundary.hydro.flow nodalgroup flow_rate_value initial_p

17.4.6 of.boundary.hydro.flow.table nodalgroup value value2 tablename

17.4.7 of.boundary.hydro.impermeable nodalgroup

17.5 blast

17.5.1 of.boundary.blast nodalgroup value tablename

17.6 element

17.6.1 of.boundary.element.stress elementgroupname sxx sxy syy

demo:

```
of.boundary.element.stress ‘all’ -20e6 2e6 -2e6
of.boundary.element.stress.xgrad elementgroupname sxx.x sxy.x syy.x
of.boundary.element.stress.ygrad elementgroupname sxx.y sxy.y syy.y
```

17.6.2 of.boundary.excavation elementgroupname

17.6.3 of.boundary.uncontact elementgroupname

17.7 thermal

17.7.1 of.boundary.thermal.t0 nodalgroupvalue

17.7.2 of.boundary.thermal.temperature nodalgroupvalue

CHAPTER

EIGHTEEN

MATERIAL

18.1 of.mat.element elementgroupname(all) modelname p1 p2 p3 ...

demo:

```
of.mat.element ‘all’ elastic den 2500 e 40e9 v 0.2
of.mat.element ‘rock’ mc den 2500 e 40e9 v 0.2 ten 10e6 coh 30e6 fric 0.3
```

18.2 of.mat.particle elementgroupname(all) modelname p1 p2 p3 ...

demo:

```
of.mat.particle ‘all’ rigid den 2500
```

18.3 of.mat.cohesive cohelementgroupname(all) modelname p1 p2 p3 ...

demo:

```
of.mat.cohesive ‘all’ em pn 1e10 pt 0.5e10 ten 30e6 coh 30e6 fric 0.3 gi 100 gii 300␣
↪(pn pt ten coh fri gi gii)
```

18.4 of.mat. cohesive ‘rock’ em_het power 0.5 dip 35 ten 30e6 3e6 coh 30e6 3e6 fric 0.3 0 gi 100 20 gii 300 30 (power, dip, mean pn, dev pn, mean pt, dev pt, mean ten, dev ten, mean coh, dev coh, mean fri, dev fri, mean gi, dev gi, mean gii, dev gii)

18.5 of.mat.contact elementgroupname1 elementgroupname2(all) modelname p1 p2 p3 ...

demo:

```
of.mat.contact ‘all’ mc kn 2e10 ks 1e10 fric 0.3
of.mat.contact ‘rock’ ‘plate’ mc kn 2e10 ks 1e10 fric 0.3 kn ks
```

18.5.1 of.mat.fluid den bulk viscousity cohesion

demo:

```
of.mat.fluid den 1000.0 k 3e8 viscosity 1e6 cohesion 3e6
```

18.5.2 of.mat.fluid.matrix elementgroupname(all) permeability m alpha

demo:

```
of.mat.fluid.matrix 'all' permeability 1.0e-8 biot_k 22e9 biot_c 0.1
```

18.5.3 of.mat.fluid.fracture cohelementgroupname(all) aperature_0 apreature_min para_exp para_b

demo:

```
of.mat.fluid.fracture 'joint' a0 5e-4 power 3.0 b 1.0
```

18.5.4 of.mat.gas initial_den permeability initial_bulk constant_b alpha

demo:

```
of.mat.fluid.fracture 'joint' density0 1.29 permiability 200 k0 1.01e5 b
1.0 alpha 0.1
```

CHAPTER

NINETEEN

GBM

19.1 of.gbm numberofminerals m1_name ratio m2_name ratio . . .

demo:

```
of.gbm 3 'qtz' 0.3 'fel' 0.46 'bio' 0.24 (area ratio)
```

CHAPTER TWENTY

HISTORY

20.1 of.history.nodal.force id x1 y1

20.2 of.history.nodal.vel id x1 y1

20.3 of.history.nodal.dis id x1 y1

20.4 of.history.nodal.fluid.pressure id x1 y1

20.5 of.history.nodal.fracture.pressure id x1 y1

20.6 of.history.nodal.matrix.pressure id x1 y1

20.7 of.history.nodal.temperature id x1 y1

20.8 of.history.nodal.group.force id groupname

20.9 of.history.nodal.group.vel id groupname

20.10 of.history.nodal.group.dis id groupname

20.11 of.history.nodal.group.fluid.pressure id groupname

20.12 of.history.nodal.group.fracture.pressure id groupname

20.13 of.history.nodal.group.matrix.pressure id groupname

20.14 of.history.nodal.group.temperature id groupname

note: average value in this group

20.15 of.history.element.stress id x1 y1

20.16 of.history.element.strain id x1 y1

20.17 of.history.element.strainrate id x1 y1

20.18 of.history.element.group.stress id groupname

20.19 of.history.element.group.strain id groupname

20.20 of.history.element.group.strainrate id groupname

20.21 of.history.cohelement.dis id x1 y1

20.22 of.history.cohelement.force id x1 y1

20.23 of.history.cohelement.vel id x1 y1

20.24 of.history.cohelement.shearstrength id x1 y1

20.25 of.history.cohelement.group.dis id groupname

20.26 of.history.cohelement.group.force id groupname

20.27 of.history.cohelement.group.vel id groupname

20.28 of.history.cohelement.group.shearstrength id groupname

20.29 of.history.energy

20.30 of.history.unbalance

20.31 of.history.solveratio

20.32 of.history.interval intervalvalue

20.33 of.history.pv.interval intervalvalue

20.34 of.history.pv.reduced.interval intervalvalue fracturethreshold intervalvalue

20.35 paraview

20.35.1 of.history.pv.field fieldkeywords

fieldkeywords: velocity force displacement fluid_pressure nodal_group element_group gbm_group mass stress strain strain_rate principal_stress mat_id fragment

20.35.2 of.history.pv.fracture fracturekeywords

fracturekeywords: mode sliding opening area time length energy

20.35.3 of.history.pv.damage damagekeywords

damagekeywords: mode sliding opening area time length

20.35.4 of.history.pv.cohesive cohesivekeywords

cohesivekeywords: velocity force displacement shear_strength dfn mat_id group

20.35.5 of.history.pv.ae aekeywords

aekeywords: mode time win_time win_kinetic kinetic magnitude energy

CHAPTER

TWENTYONE

DFN

21.1 of.dfn.connectivity dfnnum

n1 n2 . . .

demo:

```
of.dfn.connectivity 1000 1 15 2 19 1000 999 ...
of.dfn.group dfnname setnum dfn1 dfn4 dfn12...
```

demo:

```
of.dfn.group ‘dfn_1’ 3 1 7 12
```

21.1.1 of.dfn.type dfnname dfn_enum

demo:

```
of.dfn.type ‘jset1’ cohesive
```

CHAPTER

TWENTYTWO

TABLE

22.1 of.table tablename ‘table1.dat’

format of table:

```
t1 t2 num
n1 n2 n3 ...
struct table:
{char\* tag;
double t1;
double t2;
unsigned int num;
double\* data;}
```

demo:

```
of.table tab1 ‘table1.dat’
```

CHAPTER

TWENTYTHREE

DAMPING

23.1 of.damp.global value

demo:

```
of.damp.global 0.7
```

demo:

```
of.damp.rayleigh elementgroupname value1 value2
of.damp.rayleigh.mass elementgroupname value
of.damp.rayleigh.stiffness elementgroupname value
```

CHAPTER

TWENTYFOUR

HYDRO

24.1 of.hydro.timestep value

demo:

```
of.hydro.timestep 1e-9
of.hydro.matrix off
of.hydro.mechanical off
of.hydro.fracture on
```

CHAPTER

TWENTYFIVE

GRAVITY

25.1 of.gravity x y

demo:: of.gravity 0.0 -9.8

CHAPTER

TWENTYSIX

SEISMIC

26.1 of.seismic.window value

demo:

```
of.seismic.window 1.0e-3
```

1. of.timestep auto or fix value
2. of.step
3. of.stop