numgeo

A finite element code for geotechnical simulations

Tutorials
-
Series A: Triaxial test

©Jan Machaček, Patrick Staubach
www.numgeo.de
September 29, 2022

History:

2021	Jan Machaček, Patrick Staubach	Initial version
2022	Jan Machaček, Patrick Staubach	Revised version, new option to initialize statevs

`numgeo`: Tutorials
©Jan Machaček, Patrick Staubach (Editors)
www.numgeo.de
September 29, 2022

Contents

1 Geometry

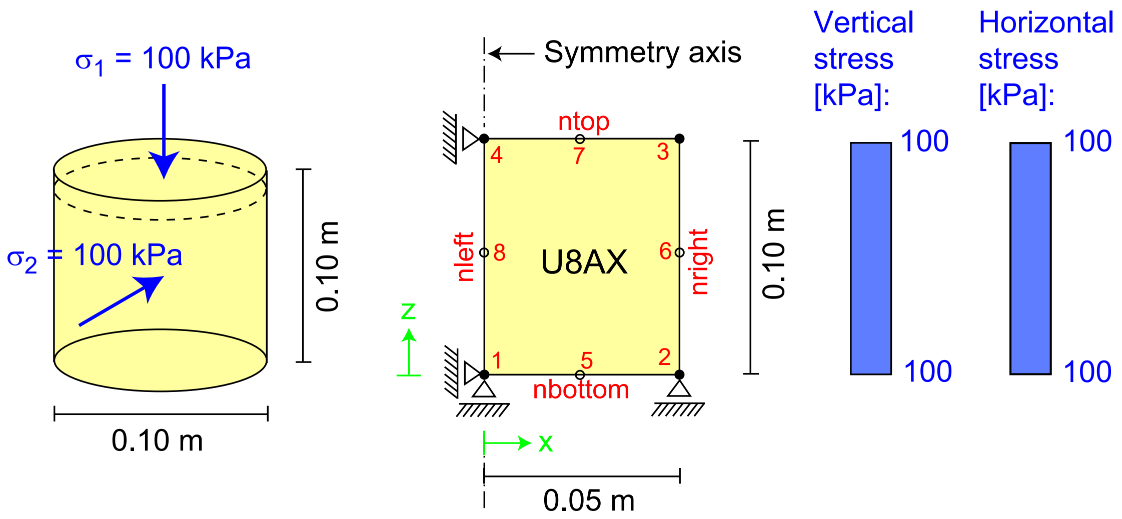

Figure 1: Model and specifications of the triaxial test

In the following the simulation of a triaxial test with a single element and displacement control is performed using the hypoplastic material model. The numerical model and its specifications are shown in Figure 1. The displayed stress state corresponds to the initial state and represents typical conditions in a triaxial test. Since the geometry and the model definition is simple, it is done without any pre-processor and completely written "by hand".

The first part is the definition of the element. Since an eight-nodded axisymmetric element is used, eight nodes are defined from line 2 to 9. Each node has a label, a x- and a y-coordinate. The element has a radius of 0.05 m and a height of 0.1 m.

```
1 *Node
2 1, 0.00, 0.00
3 2, 0.05, 0.00
4 3, 0.05, 0.10
5 4, 0.00, 0.10
6 5, 0.025,0.00
7 6, 0.05,0.05
8 7, 0.025,0.1
9 8, 0.00,0.05
10 *Nset, Nset=nall
11 1,2,3,4,5,6,7,8
12 *Nset, Nset=nleft
13 1, 4, 8
14 *Nset, Nset=nright
15 2, 3, 6
16 *Nset, Nset=nbottom
17 1, 2, 5
18 *Nset, Nset=ntop
19 3, 4, 7
20 *Element, Type=U8-AX
21 1, 1, 2, 3, 4, 5, 6, 7, 8
22 *Elset, Elset=eall
23 1
```

After the definition of the nodes, they are arranged in node sets (`NSET`) in order to apply boundary conditions later. The node set *nall* contains all nodes, *nleft* the nodes in the axisymmetry axis and *nright* the right nodes. In addition, the node sets *nbottom* and *ntop* are generated. The element is defined in line 20 to 21. An element discretising displacement (`U`) with eight nodes and axisymmetry is used. The label of the element is 1 (first entry in line 21) and the contained nodes are given subsequently. The node labels are given counter-clockwise as it is needed for the finite element method.

2 Hypoplastic material model

After the generation of the mesh, a solid section is defined in line 25.

```
24 **
25 *Solid Section , elset=eall , material=hypo_soil
26 **
```

The material name, which is defined subsequently, is chosen to be *hypo_soil*. In line 27, the material is defined and is assumed to consist of one phase (solid grains). Therefore, the soil is either assumed to be completely dry and the air having no impact on the constitutive behaviour or it is water saturated but the water does not change the effective stress (ideally drained) nor the constitutive behaviour.

```
27 **
28 ** Materials
29 **
30 *Material , name = hypo_soil , phases = 1
31 *Mechanical =Hypoplasticity
32 0.578 , 0.00 , 18930000 , 0.27 , 0.549 , 0.851 , 1.15 , 0.137
33 0.34 , 1.0 , 1.0 , 0.0001 , 0.1 , 6.0 , 0
34 *Density
35 1.65
36 *Minpressure
37 -0.1
```

Line 28 defines the constitutive law which is used as material. The hypoplastic material model is chosen here. The parameters for the hypoplastic model are given from line 29 to 30. The adopted parameters are given in Table 1. The first nine constants (φ_c to β) belong to the basic version of the hypoplastic model. Strictly

φ_c [rad]	ν	h_s [kPa]	n	e_{c0}	e_{d0}	f_{ei0}
0.578	0	18930000	0.27	0.549	0.851	1.15
α	β	m_T	m_R	R	β_R	χ
0.137	0.34	1	1	0.0001	0.1	6

Table 1: Parameters used for the hypoplastic model with intergranular strain extension

speaking, the Poisson's ratio ν is not a parameter of the original hypoplastic model of von Wolffersdorff [4], but has been introduced into the hypoplastic constitutive model by Andrzej Niemunis to give the user the possibility to increase the shear stiffness. If it is set to zero, the original model by von Wolffersdorff is preserved. The remaining constants m_T to χ are necessary for the extension by the intergranular strain extension [2]. In order to switch of this extension one can set m_T and m_R to 1 which was done in the present case.
The density of the material is given in line 31 to 32. Since no gravity is taken into account in this example (the contribution is negligible since the height of the sample is small), the density has no impact on the simulation.
Since hypoplastic models can not handle tensile mean effective stresses, a minimum pressure is defined in line 33 to 34. If at any point of the simulation the mean effective stress falls below this stress, the stress state is corrected.
The initial conditions are specified for the initial stress state from lines 36 to 37. Here, the element set *eall* is assigned -100 kPa at $y = 0.0$ m and -100 kPa at $y = 0.1$ m. The lateral stress coefficient for the horizontal direction as well as in plane are defined as 1.0. Therefore, an isotropic stress state is defined.

```
38 **--------------Initial Conditions--------------**
39 *initial conditions , type=stress , geostatic
40 eall , 0.0 , -100.0, 0.10 , -100.0, 1.0 ,1.0
41 *INITIAL CONDITIONS,TYPE=state variables ,user
42 eall
```

In addition to the initial stress, the initial void state variables (void ratio, intergranular strain) have to be set (line 38). For this purpose, an external fortran file is used:

```
subroutine user_initial_state_variables(ie,igp,ndim,nstatev,material,coords,statev) bind(c,name
    ='user_initial_state_variables')
  use, intrinsic :: iso_c_binding
  implicit none
  integer(c_int)                     , intent(in)     :: ie
  integer(c_int)                     , intent(in)     :: igp
```

```
5   integer(c_int)                      , intent(in)     :: ndim
6   integer(c_int)                      , intent(in)     :: nstatev
7   character(c_char)                   , intent(in)     :: material(*)
8   real(c_double), dimension(3)        , intent(in)     :: coords
9   real(c_double), dimension(nstatev), intent(inout) :: statev
10
11  ! user coding to define the initial state variables statev(:)
12  statev(:) = 0.0d0
13  statev(1) = 0.640 ! void ratio
14
15
16 end subroutine user_initial_state_variables
```

This fortran file (`user_initial_state_variables.f90`) has to be compiled before running the program since it is used dynamically by the program (Dynamic-Linked-Library, DLL or Shared Object, SO). It can be found in the folder user-subroutines supplied together with `numgeo`. The void ratio is then assigned a value of $e = 0.640$ which corresponds to a dense initial state ($D_{r0} = 0.7$) via the first entry of the vector `statev` (line 13).
The fortran file is compiled using the supplied shell script:

```
./user_initial_state_variables.sh
```

Note that you have to change the directory to your folder first. The files `user_initial_state_variables.o` and `user_initial_state_variables.so` should have been created.
For the compression of the sample, a linear increase in the displacement has to be defined. Therefore, a linear increasing amplitude is defined in line 1 to 2.

```
0 **———————Amplitudes———————**
1 *AMPLITUDE, NAME = LoadingRamp , TYPE = RAMP
2 0.0, 0.0, 1.0, 1.0
```

The amplitude *LoadingRamp* defines a linear increase of an arbitrary quantity (it will be used for the displacement here) beginning with the relative value 0 at the time $t_1 = 0$ and the relative value 1 at time $t_2 = 1$.
In `numgeo`, the loading history is defined using steps. The first step for most geotechnical problems is the initialisation of the stress distribution (`Geostatic` step). This means that the stress defined in the initial conditions (internal forces) is imposed by external loads (body forces, distributed loads or concentrated loads) to bring the system into equilibrium (internal forces equal external forces).

```
0  **———————Steps———————**
1  *Step, name=step1, inc = 1
2  *Geostatic
3  *Solver, mumps
4  *Body force, instant
5  eall, GRAV, 0.0, 0, -1, 0
6  *Dload,instant
7  eall, P3, -100.0
8  *Dload,instant
9  eall, P2, -100.0
10 *Boundary
11 nleft, u1, 0.0d0
12 nbottom, u2, 0.0d0
13 **
14 *Output, field, vtk, ASCII
15 *Node output, nset = all
16 U
17 *element output, elset = eall
18 S,Void
19 *End Step
```

Listing 1: Definition of the Geostatic step

Line 1 starts the step environment and defines the name of the step. The analysis type of this step is given in line 2. As mentioned before, a Geostatic step is performed. In line 3, the solver is specified, which will be the MUMPS solver in this calculation. The keyword `Body force` imposes a gravitational force which is applied instantaneous (`Instant`). The element set *eall* is loaded by the gravity (amplitude 0.0 m$/s^2$, directed downwards with the normalized vector of the gravity $\vec{\mathbf{b}} = \{0, -1, 0\}$). As mentioned before, no gravity is taken into account, therefore the magnitude is zero. The load due to the cell pressure of the trixial device is defined from line 6 to 9. The upper as well as the right face of the element is loaded by 100 kPa. The boundary conditions are specified

from line 10 to line 12. The left nodes are constrained in horizontal direction and the nodes at the bottom in vertical direction (the displacements are assigned a value of zero which is equivalent to a fixed boundary).
The output demand is specified from line 14 to line 18. The output is written in the vtk format (suitable for `paraview`) and is of ASCII type. The node output includes the displacement (`U`) of all nodes and the element output includes the stress (`S`) as well as the void ratio (`Void`).
In the second step, the soil sample is compressed monotonically. Therefore, a linearly increasing displacement boundary condition is defined.

```
0  **
1  ** Steps
2  **
3  *Step, name=step2, inc = 1000
4  *Static
5  0.001,1.0,0.0001,0.001
6  *Solver, mumps
7  *Body force, instant
8  eall, GRAV, 0.0, 0, -1, 0
9  *Dload,instant
10 eall, P3, -100.0
11 *Dload,instant
12 eall, P2, -100.0
13 **
14 *Boundary
15 nleft, u1, 0.0d0
16 nbottom, u2, 0.0d0
17 *Boundary,amplitude =LoadingRamp
18 ntop, u2, -0.03d0
19 **
20 *Output, field, vtk, ASCII
21 *Frequency=5
22 *Node output, nset = all
23 U
24 *Element output, elset = eall
25 S,VOID
26 *output,print
27 *element output, elset=eall
28 E,S,VOID
29 *End Step
30 *End Input
```

Listing 2: Linear compression of the soil sample in the second step

Since the hypoplasticity is non-linear (the relation between strain and stress is non-linear), an incrementation scheme with small increment size is specified in line 3 (the solution can not directly be obtained for an arbitrary strain because hypoplasticity is path-dependent. A simulation with small steps in strain is necessary). The initial increment as well as the largest allowed increment are 0.001. The total step size is 1, which means that 100 % of the specified displacement is applied at the end of the step. In one increment $0.001 \cdot 100\% = 0.1\%$ of the displacement is applied. The minimum increment size is 0.0001, which is only used by `numgeo` if convergence (force equilibrium) is achieved very slowly (with a large number of iterations per increment).
The loading definitions remain the same here as in the first step. Since `numgeo` does not carry loads from previous steps automatically, every acting loads have to be re-defined in a new step. The compression of the element is defined in line 15 to 16. The top nodes are displaced vertically by a magnitude of 0.03 m which leads to an axial strain of 30 % for the sample at the end of the step. As explained earlier, the displacement is linearly increased using the amplitude *LoadingRamp*. For further investigations the strain (`E`), the stress (`S`) and the void ratio (`VOID`) of the integration point of the element are saved in a separate text file by defining a print output in line 25 to 27.
The calculation is started by opening a command window, navigating to the directory and executing numgeo in the command line. Subsequently specifying the name of the input file (without ending), confirming by pressing enter, specifying the number of CPUs (1 in this case) and pressing enter again. Figure 2 shows the output of the vertical stress from a calculation with a relative density of $D_{r0} = 0.7$, i.e. an initial void ratio of $e_0 = 0.640$. In order to get the same graph, follow the steps shown in Figure 2.
The data can be saved and further post-processed with python for instance. This has been done for the deviatoric stress $q = \sigma_{22} - \sigma_{11}$ and the change in void ratio Δe for two different initial relative densities shown in Figure 3. The simulation shows the typical behaviour of a granular media, indicating increased shear resistance for less void space and greater dilatancy for an initially denser packings.

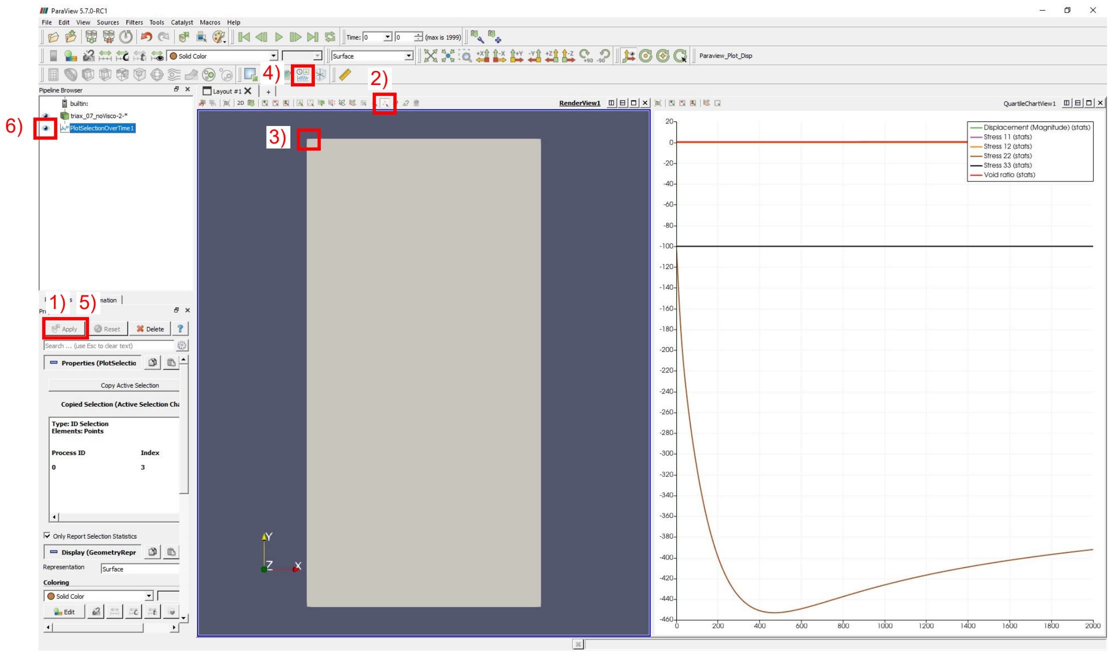

Figure 2: Results of the triaxial test and development of vertical stress with increasing displacement

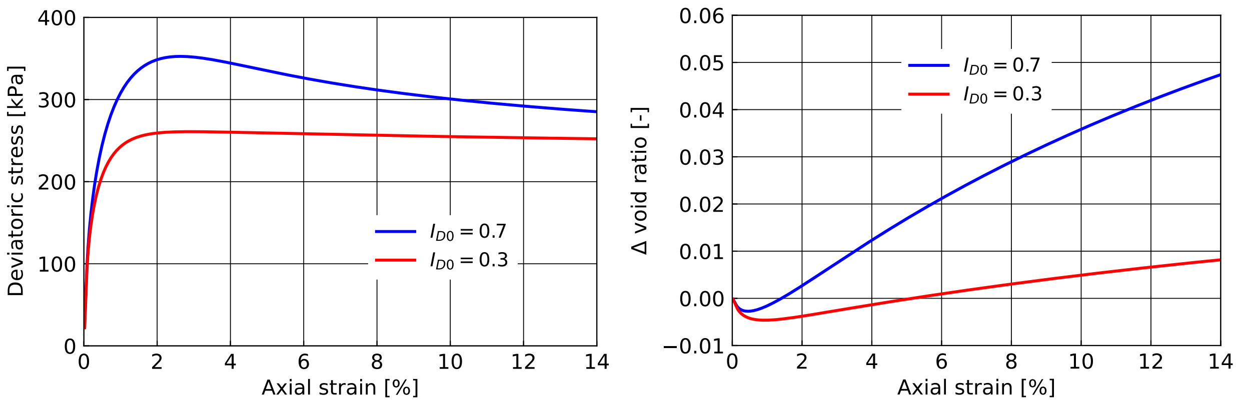

Figure 3: Deviatoric stress and change in void ratio for two different initial densities I_{D0}

3 Sanisand material model

Sanisand is an advanced elasto-plastic model for soil developed by Dafalias and Manzari [1].
In order to use the Sanisand constitutive model, the definition of the material has to be changed to:

```
**
** Solid Section
**
*Solid Section, elset=eall, material=sani_soil
**
** Materials
**
*Material, name = sani_soil, phases = 1
*Mechanical = Sanisand
100, 1.103, 0.122, 0.205, 1.34, 0.94, 0.05, 100
0.25, 4., 0.95, 1.2, 0.9, 2.0, 1, 100
*Density
1.65
```

The assigned parameters are given in Table 2 with their abbreviation. These parameters have been determined by Wichtmann [3] based on monotonic and cyclic triaxial tests.

p_{atm} [kPa]	e_0	λ_c	ξ	M_c	M_e	m	G_0
100	1.103	0.122	0.205	1.34	0.94	0.05	100
ν	h_0	c_h	n_b	A_0	n^d	c_z	$z_{\max}$
0.25	4	0.95	1.2	0.9	2.0	1	100

Table 2: Parameters of Sanisand for Karlsruher fine sand

In addition, the initial state variables have to be defined identical to the simulation with the hypoplastic model using the `user_initial_state_variables.f90` file:

```
subroutine user_initial_state_variables(ie,igp,ndim,nstatev,material,coords,statev) bind(c,name
   ='user_initial_state_variables')
  use, intrinsic :: iso_c_binding
  implicit none
  integer(c_int)                    , intent(in)    :: ie
  integer(c_int)                    , intent(in)    :: igp
  integer(c_int)                    , intent(in)    :: ndim
  integer(c_int)                    , intent(in)    :: nstatev
  character(c_char)                 , intent(in)    :: material(*)
  real(c_double), dimension(3)      , intent(in)    :: coords
  real(c_double), dimension(nstatev), intent(inout) :: statev

  ! user coding to define the initial state variables statev(:)
  statev(:) = 0.0d0
  statev(1) = 0.640 ! void ratio

end subroutine user_initial_state_variables
```

Note that the initial back-stress tensor of the Sanisand model is automatically initialized such that the initial stress state is in the centre of the elastic locus. However, the user can also define the initial back-stress tensor by setting the statevs 2:ntens+1, where ntens is the number of stress components.
As an alternative, to avoid using the user file, the initial state variables can also be directly initialized in the input file by:

```
**-----------------Initial Conditions-----------------**
*INITIAL CONDITIONS,TYPE=state variables,default
eall, void_ratio, 0.64d0
```

The development of deviatoric stress with respect to axial strain using Sanisand is given in Fig. 4.

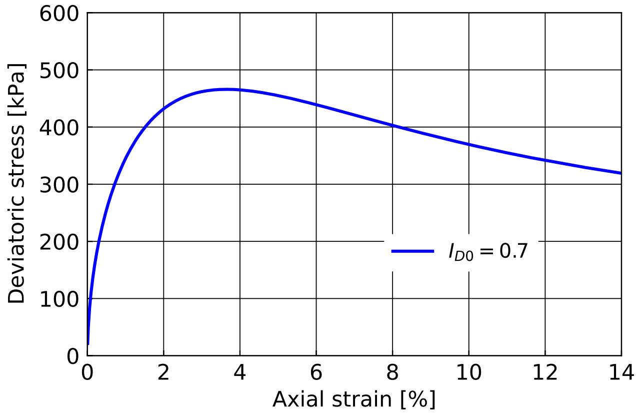

Figure 4: Deviatoric stress using the Sanisand model

References

[1] Dafalias, Y. F. and Manzari, M. T. "Simple Plasticity Sand Model Accounting for Fabric Change Effects". In: *Journal of Engineering Mechanics* 130.6 (2004), pp. 622–634. ISSN: 0733-9399. DOI: 10.1061/(asce)0733-9399(2004)130:6(622).

[2] Niemunis, A. and Herle, I. "Hypoplastic model for cohesionless soils with elastic strain range". In: *Mechanics of Cohesive-Frictional Materials* 2.4 (1997), pp. 279–299. ISSN: 10825010. DOI: 10.1002/(SICI)1099-1484(199710)2:4<279::AID-CFM29>3.0.CO;2-8.

[3] Wichtmann, T., Fuentes, W., and Triantafyllidis, T. "Inspection of three sophisticated constitutive models based on monotonic and cyclic tests on fine sand: Hypoplasticity vs. Sanisand vs. ISA". In: *Soil Dynamics and Earthquake Engineering* 124 (2019), pp. 172–183. ISSN: 02677261. DOI: 10.1016/j.soildyn.2019.05.001.

[4] Wolffersdorff, P.-A. von. "A hypoplastic relation for granular materials with a predefined limit state surface". In: *Mechanics of Cohesive-Frictional Materials* 1 (1996), pp. 251–271.