TOCHNOG PROFESSIONAL Tutorial manual
Version 23

Dennis Roddeman

December 7, 2015

Contents

1 Conditions

All conditions from the Tochnog Order form apply. See our internet page for the latest order form.

2 Basic information

The tutorials in this manual describe calculations in decreasing detail. The first tutorial discusses many aspects, whereas the following tutorials discuss less aspects. You can find these tutorials on your distribution in the directory **tochnog/test/tutorial**.

Mesh preparation and post-processing is done with Gid 7.6.0b under Linux. Gid is copyrighted by CIMNE, see http://www.gidhome.com. It is advised to learn Gid first.

The advanced user under Linux can also find examples on how specific input commands are used by grepping in the tochnog test input files in the **tochnog/test/other** directory, e.g. **grep control_timestep *.dat**.

3 Tutorial 1: slope safety factor analysis

This tutorial is taken from example 2 in [1]. The safety factor of a slope is calculated.

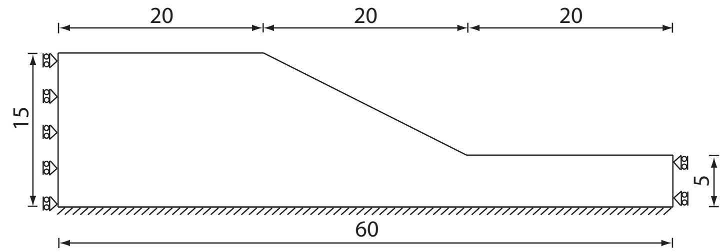


Figure 1: Slope

Figure 1 shows the slope. The lower edge is completely fixed, whereas at the left and right edge free sliding in vertical direction is allowed.

3.1 Mesh generation with Gid

Although Tochnog contains some build in mesh generation, here for generality the external Pre-and Postprocessor Gid is used. Start Gid and perform the following steps.

<u>Specify the Tochnog problem type in Gid</u>

Data - Problem type - Tochnog

This takes care that Gid understands that you want to generate a mesh for Tochnog. Some Tochnog specific input is now available, and once the mesh is generated it can be written in a file in Tochnog specific format.

<u>Create points, lines and a surface in Gid</u>

Geometry - Create - Point

Create the points along the entire edge of the slope, thus (0,0), (60,0), (60,5), (40,5), (20,15) and (0,15).

Zoom - Frame

Zoom to the total frame to see all points.

Geometry - Create - Straight Line

Use the points to create the straight lines along the entire edge.

Geometry - Create - Nurbs surface - By contour

Use the straight lines to create a surface. Figure 2 shows the Nurbs surface.

<u>Assign a material to the surface</u>

Data - Materials - Assign - Surfaces

Now assign to the surface a Tochnog element group (material). Select **GROUP1** in the dialog, and set the **group_type** to **-materi**. Then select the surface with **Assign Surfaces**.

<u>Generate a mesh with linear triangles</u>

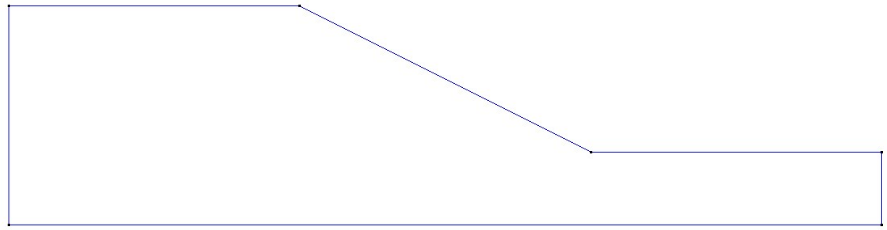


Figure 2: Nurbs surface of slope

Mesh - Generate mesh

Gid uses as default element type for surfaces linear triangles, which are OK for the slope stability analysis. Hence we don't need to change the element type, and can go straight on to generate the mesh: Setting the element size to 0.5 gives a fine mesh, good enough for an accurate solution.

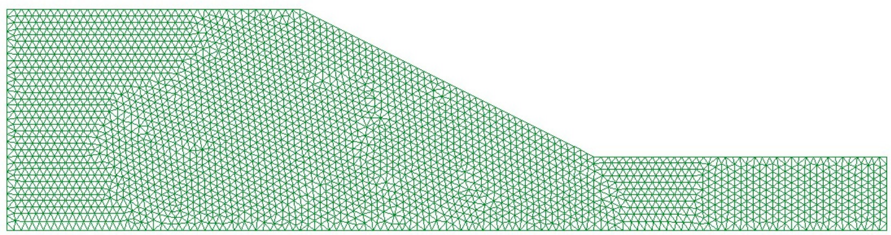


Figure 3: Mesh of slope

Save gid data

File - Save As - mesh.gid

Save everything to the directory **mesh.gid**.

Write file with tochnog elements, nodes and element groups (materials)

Calculate - Calculate

To really obtain the generated mesh in a file that can be used for Tochnog, calculate the Tochnog file. Please realize that this option in Gid only calculates the Tochnog input file for the mesh , and not does the FE calculation itself (solving equations, ...). In the directory **mesh.gid** you can find the file **mesh.dat** which contains **element** , **element_group** and **node** data.

The **element** data records contain the element number, the element name, here **-tria3**, and finally the node numbers to which the element is connected. The **element_group** records contain for each element the group number which contains material properties for the element. The **node** records contain for each node the coordinates.

3.2 Input file

The input file always contains two parts. The initialization part essentially specifies which unknown fields need to be solved. The data part specifies elements, nodes, boundary conditions, etc.

3.2.1 Initialization part

echo -yes
number_of_space_dimensions 2
materi_velocity
materi_displacement
materi_stress
materi_strain_plasti
end_initia

The first line **echo -yes** tells Tochnog that the input should be echoed when it is read. This is convenient in locating errors in your input file: you can see up to which line the input file is read before the error occurs. Use **echo -no** if you don't want to echo the input file.

The **number_of_space_dimensions 2** specifies that the dimensionality of the calculation is 2D (plane strain, plane stress, or axi-symmetric). Also 1 and 3 are available in Tochnog (thus 1D and 3D).

The **materi_velocity** together with the **materi_displacement** takes care that the velocity and displacement fields are present for solution. Since this is a 2D calculation, only the x and y components are present in the calculation.

Initialization of the stress field is done with **materi_stress**. Please realism that all the 9 components σ_{ij} of the 3D stress matrix are present in the calculation, since, by example in plane strain, displacements in x and y also lead to σ_{zz} However, the stress matrix is symmetric, so only 6 components need actually to be stored for the stress field; the other components follow from symmetry.

In this slope stability calculation an elasto-plastic material model will be used, and thus initialization of the plastic strains **materi_strain_plasti** is needed.

After you have run the calculation you can always see in the data record **dof_label** in the database file, in this case **tutorial_1.dbs**, the component names of the initialized dof's (**-disx** and **-disy** for the displacements **materi_displacement**, etc.).

3.2.2 Data part, geometry of edges

start_define
bottom_edge geometry_line 1
end_define

start_define
right_edge geometry_line 2
end_define

start_define
left_edge geometry_line 3
end_define

We want to specify the edges at which boundary conditions are imposed later. Each edge will be specified by a **geometry_line** data record, since the edges are straight lines. It is convenient to define a name for each edge, so that later that name can be used when needed and the input file remains more readable. Each **start_define** ... **end_define** specifies a word, by example **bottom_edge** which will be later substituted with its real meaning, by example **geometry_line 1**. Such defines can be used for all kinds of input.

> **bottom_edge 0.0 0.0 60.0 0.0 0.01**
> **left_edge 0.0 0.0 0.0 15.0 0.01**
> **right_edge 60.0 0.0 60.0 5.0 0.01**

The first line **bottom_edge 0.0 0.0 60.0 0.0 0.01** is read by Tochnog as **geometry_line 1 0.0 0.0 60.0 0.0 0.01**, so in fact **geometry_line 1** is specified. The other two lines specify **geometry_line 2** and **geometry_line 3**. The 0.01 indicate the tolerance of the lines; all nodes in the model with a distance not more than 0.01 are considered to be located on the geometrical lines.

Notice that we identify each **geometry_line** by a unique index; for the present geometry lines the indices 1, 2 and 3 are used. In fact, most of the data in the input file uses an index, by example **element** data records use an index which is the element number, **node** data records use an index which is the node number, etc.

3.2.3 Data part, boundary conditions

> **bounda_dof 0 -bottom_edge -disx -disy**
> **bounda_dof 1 -left_edge -disx**
> **bounda_dof 2 -right_edge -disx**

The **bounda_dof** records are used to prescribe unknowns (dof's). On the bottom edge the displacement in x-direction **-disx** and the displacement in y-direction **-disy** are suppressed, and on the left edge and right edge the displacement in x-direction **-disx** is suppressed. The nice thing about using geometries for boundary conditions is that these remain valid, even if you change the mesh (amount of elements, nodes).

First side remark: if we want to apply a non-zero displacement then also the **bounda_time** records should be specified; here however displacements on the edges are zero, and then the **bounda_time** records are not needed.

Second side remark: if displacements are prescribed, velocities automatically are calculated by Tochnog from the time derivative of the displacements. If you would prescribe velocities with **-velx** and **-vely** then displacements automatically follow from time integration of the velocities.

Third side remark: a list of all unknown names like **-disx** etc. can be found at **dof_label** in the users manual. The **dof_label** is available in the database file after the calculation.

3.2.4 Data part, gravity

> **force_gravity 0.0 -10.**
> **force_gravity_time 0.0 0.0 0.5 1.0 1.e20 1.0**

The gravity components as specified in **force_gravity** are 0 in x-directions, and $-10.\frac{m}{s^2}$ in y-direction. The **force_gravity_time** specifies at time versus factor diagram; this is the factor with which the gravity is applied. It should be read as follows: at time 0.0 the factor is 0.0, at time 0.5 the factor is 1.0 and up to time $1.e20$ the factor remains 1.0. Later in this tutorial, we will further discuss this topic; so just go straight ahead with reading the next input.

Please realize that all input in Tochnog does not have a specific dimension. The user just should take care that he/she uses consistent units for the different input data, but otherwise the units of input data is not pre-defined.

3.2.5 Data part, material properties

```
start_define
phi 0.349065
end_define
start_define
tanphi 0.3639
end_define

start_define
c 10.0
end_define
```

First we define the soil friction angle $\phi = 0.349065$ in radians (so 20 degrees), and the cohesion $c = 10.0\frac{kN}{m^2}$.

```
group_type 1 -materi
group_materi_memory 1 -total_linear
group_materi_density 1 2.0
group_materi_elasti_young 1 1.e5
group_materi_elasti_poisson 1 0.3
group_materi_plasti_tension_direct 1 10.0
group_materi_plasti_mohr_coul_direct 1 phi c 0.0
```

When we made the mesh with GID, all elements were assigned to group 1. Here we define the material properties of group 1. The **group_type** is set to **-materi** which means that material strains, stresses, etc. will be solved. With **group_materi_memory** set to **-total_linear** you specify a classical geometrically linear (small deformations) approximation for the soil; this is sufficient for almost all typical calculations. The soil density is set to $2.\frac{1000.kg}{m^3}$ in the data record **group_materi_density**. The Young's modulus and Poisson's ratio are $1.e5\frac{kN}{m^2}$ and 0.3, as set in the records **group_materi_elasti_young** and **group_materi_elasti_poisson** respectively. For shear failure the Mohr-Coulomb yield surface is used; the record **group_materi_plasti_mohr_coul_direct** specifies the friction angle ϕ, the cohesion c and zero dilatancy. To include tension failure (limitation of tensile stress in soils), you explicitely need to include **group_materi_plasti_tension_direct** in the input file; here the maximum tensile stress is set to $10.0\frac{kN}{m^2}$.

```
change_dataitem 10 -group_materi_plasti_mohr_coul_direct 1 0 -use
change_dataitem_time 10 0.0 tanphi 1.0 tanphi 2.0 0. 1.e20 0.0
change_dataitem_time_method 10 -tangent
change_dataitem 20 -group_materi_plasti_mohr_coul_direct 1 1 -use
```

change_dataitem_time 20 0.0 c 1.0 c 2.0 0.0 1.e20 0.0

The **change_dataitem** and **change_dataitem_time** records take care that the friction angle and cohesion are lowered in the calculation, so that we find the minimum friction angle and cohesion for stability. The ratio of the start values and the minimum values will define the safety factor.

The **change_dataitem 10** record specifies for the **group_materi_plasti_mohr_coul_direct** record with index 1 (so group 1) and 'number 0'. With 'number 0' the first value in the **group_materi_plasti_mohr_co** is meant, thus the friction angle. The diagram in **change_dataitem_time 10** specifies that at time 0.0 the friction angle is the predefined word **phi** (thus 0.349065), at time 1.0 it is again the predefined word **phi**, at time 2.0 it is 0.0 and at time $1.0e20$ it is 0.0. Between the specified time points the value will be linearly interpolated. The **-use** specifies that the given values should actually be used (as opposed to other options, see the users manual).

Similarly the cohesion is varied in time with **change_dataitem 20** and **change_dataitem_time 20**.

3.2.6 Data part, prepare post-processing

post_point 10 20.0 15.0

With **post_point 10** a point in the domain is defined, at which Tochnog will monitor during the calculation solution fields. The solution field at the point will be placed in the record **post_point_dof 10**. Such **post_point_dof** record in turn can be used in several printing options.

3.2.7 Data part, apply gravity time-steps

control_timestep 10 1.e-1 1.0
control_print 10 -time_current -post_node_rhside_ratio

control_print_gid 20 -yes

control_reset_dof 30 -disx -disy -eppxx -eppxy -eppxz -eppyy -eppyz -eppzz

Time-steps are actually started with **control_timestep 10**. There is one *important* topic about all **control_*** records that we need to discuss first. These **control_*** records also have an index, here the indices are 10, 20 and 30. Specifically for **control_*** records these indices are important: these records will be performed in order of increasing index. Here that means that the records **control_timestep 10** and **control_print 10** are performed first, after that the record **control_print_gid 20** is performed, and after that the record **control_reset_dof 30** is performed.

The **control_timestep 10** record tells Tochnog to take time-steps of size 0.1 until a total time increment of 1.0 is obtained. The **control_print 10** record specifies that some data records need to be printed, specifically the current time **-time_current** and the accuracy ratio **-post_node_rhside_ratio**. The accuracy ratio contains the maximum out-of-balance force on a free node (that is a node without prescribed displacement) divided by the maximum external force on a node with prescribed displacement. Normally all printing during a calculation goes to the computer monitor (but the advanced user can redirect printed output to a file).

Now look back at how we defined the gravity in time. It was increased from 0.0 at time 0.0 to its final value of 10 at time 0.5. Thus, the time increment of 1.0 is enough for getting gravity imposed (from time 0.0 to time 0.5), and also allows the equilibrium with full gravity to be established accurately with additional time-steps (from time 0.5 to time 1.0).

The **control_print_gid 20** record takes care that the results at the end of gravity are printed in the **tochnog_flavia.msh** and **tochnog_flavia.res** files. These can be post-processed with Gid.

Finally, to prepare the safety factor calculation itself, some results are set to 0.0 in order to be able to distinguish clearly after the safety factor calculation between safety factor results (instability results) and gravity results. To be specific, the displacement components **-disx**, **-disy** are set to 0.0 and the components of the plastic strains **-eppxx -eppxy -eppxz -eppyy -eppyz -eppzz** are set to 0.0.

3.2.8 Data part, apply safety factor calculation time-steps

```
control_timestep 40 1.e-4 1.0
control_timestep_iterations_automatic 40 1.e-3 1.e-5 1.e-2
control_print 40 -time_current -post_node_rhside_ratio
control_print_history 40 -post_point_dof 10 -disy

control_print_gid 50 -yes
```

Look first back at the material definition where the friction angle and cohesion are lowered between time 1 and time 2. We will now use time-steps to determine where between time 1 and 2 the slope becomes unstable. The records **control_timestep 40** with **control_timestep_iterations_automatic 40** together define automatic time-steps, which will be taken till instability occurs. The record **control_timestep** defines an initial timestep of 1.e-4 and total time increment of 1.0. The record **control_timestep_iterations_automatic** specifies that the maximum allowed value of the error ratio **-post_node_rhside_ratio** is 1.e-3, the minimum allowed time-step size is 1.e-5 and the maximum allowed time-step size is 1.e-2. Tochnog will decrease the time-step size if that is needed to keep the error ratio below 1.e-3. If that is not possible anymore, due to instability caused by low friction angle and cohesion, the calculation is terminated. Later we will discuss how, using the current time of the terminated calculation, the safety factor can be calculated.

3.2.9 Data part, include mesh generated with Gid

```
include mesh.gid/mesh.dat
```

The **include** command allows you to include other files containing data records in the current input file. Here the mesh data as generated with Gid is included.

3.2.10 Data part, final remarks for advanced users

```
solver_matrix_save -always
```

Default Tochnog will completely setup and decompose the system matrix when it thinks the matrix is changed. In the present calculation some material data is changed (the friction angle and cohesion), and so Tochnog will setup and decompose the system matrix at all time-steps between time 1.0 and 2.0. You can impose, however, that the matrix will not be setup and decomposed each time-step by using **solver_matrix_save -always**; then the decomposed matrix is saved once, and used again at the next time-step. This can save considerable computing time. This typically is useful for safety factor calculations, but otherwise should only be used with care.

target_item 10 -time_current 0 0
target_value 10 1.0 2.e-2

The records **target_item** and **target_value** are a convenient method to check if critical results of a calculation do change. With **target_item** you specify which result should be checked, and with **target_value** you specify the expected value maximum allowed deviation from the result. In case the real value differs more from the expected value, here 1.0, then the maximum deviation, here 2.e-2, an error message is printed in the file **tochnog.log**.

In this way you can make your own set of calculations which are important for your applications, and check with each new tochnog release if results are still like you expect them to be.

3.3 Run calculation

On Linux open a window, and type in the directory with the tutorial 1 input file the command **tochnog tutorial_1**. On Microsoft Windows open a DOS command prompt, and type in the directory with the tutorial 1 input file the command **tochnog.exe tutorial_1**.

After the calculation is ready you will see the following extra files in the directory:

- **tutorial_1.dbs**
  This is the database after the calculation. It contains all information from the calculation (input data, results, etc.).
- **tutorial_1_flavia.msh** and **tutorial_1_flavia.res**
  Post-processing files which can be opened with Gid.
- **tochnog.log**
  Log file from Tochnog which lists the calculations that have been run.
- **post_point_dof_10_disy.his**
  History file with time versus y-displacement for the **post_point 10**.

3.4 Output results

3.4.1 Determine safety factor

You will see that at time 1.36 the friction angle and cohesion have become such low that the slope becomes unstable. At that time the cohesion has decreased from $10.0\frac{kN}{m^2}$ to $(1. - 0.36) * 10.0 = 6.4\frac{kN}{m^2}$. The safety factor thus is $\frac{10}{6.4} = 1.56$.

3.4.2 Plot displacement history in Gnuplot

A history plot of the vertical displacement of the **post_point** in the file **"post_point_dof_10_disy.his"** shows what is happening when the cohesion and friction angle are lowered in time between 1.0 and 2.0. Here we use **Gnuplot** as plotting program, but since the file **"post_point_dof_10_disy.his"** is a simple ASCII text file you can use any program of your preference for plotting. With **Gnuplot** do the following for getting a postscript result of the displacement history:

gnuplot

p "post_point_dof_10_disy.his" with lines

set term postscript

set output "post_point_dof_10_disy.ps"

replot

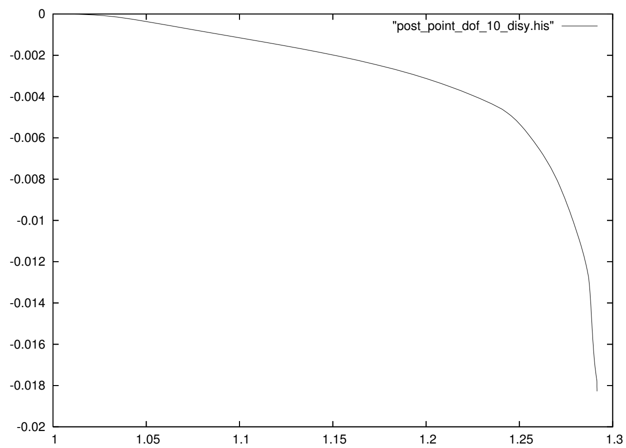


Figure 4: Vertical displacement of post_point in time

Clearly finally the slope becomes unstable.

3.4.3 Plot results in Gid

Open in Gid results

View - Postprocess

File - Open

tutorial_1_flavia.msh

View - Zoom - In

In Gid you first need to open the Gid results files as written by Tochnog, and then zoom in to get the slope full in the figure.

Plot boundary conditions in Gid

View results - Display Vectors - materi bounda dof - All

Tochnog also writes information about boundary conditions, external forces, etc. in the Gid post-processing files. In the present example, you can look in Gid at the boundary conditions with the command as given above.

Plot gravity results

View results - Default Analysis/Step - time - 1.

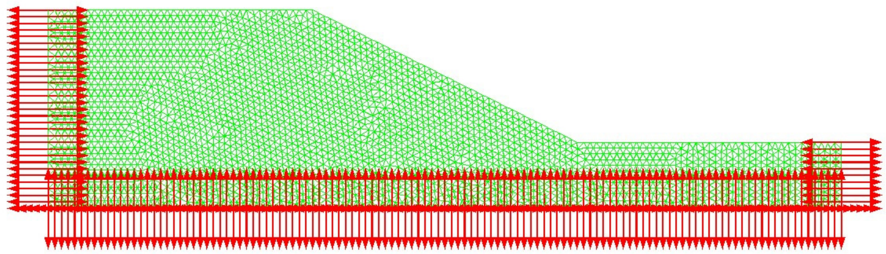


Figure 5: Boundary conditions

View results - Contour Fill - materi stress - sigyy

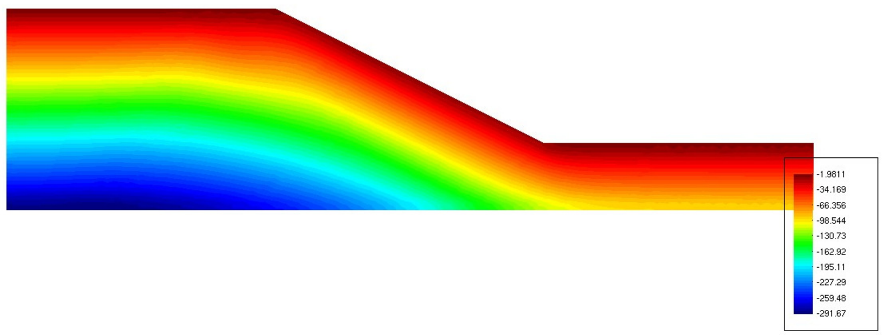


Figure 6: Vertical stress

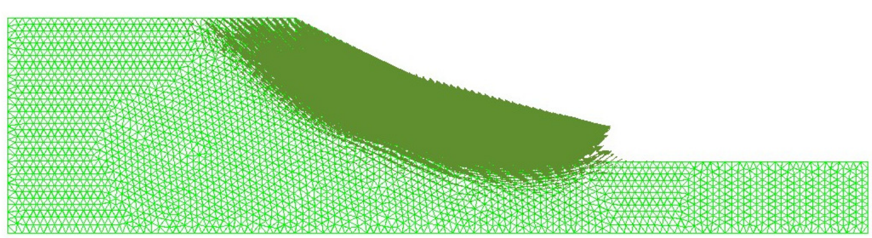
These commands show how you obtain in Gid a contour filled plot of the vertical stress field after gravity is imposed (time 1.0).

Plot safety results

View results - Default Analysis/Step - time - 1.32

Windows - View Style - Style - All Lines

View results - Display Vectors - materi velocity - | materi velocity |



The velocity profile at instability clearly shows how a circular segment slides away. It is evident that the Mohr-Coulomb shear failure condition with reduced cohesion and friction angle causes this instability.

4 Tutorial 2: non-saturated dam with seepage edge

This tutorial shows how to perform a non-saturated groundflow analysis in a dam. Figure 7 shows the dam.

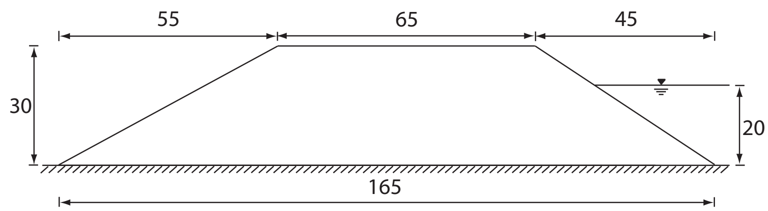


Figure 7: Dam

At the lower part of the right edge a water level of $20m$ is present, by example because of a lake. The lower edge of the dam does not allow for inward or outward flow of ground water, it is non-permeable. The left edge is a seepage edge, it only allows for outward water flow and not for inward water flow.

4.1 Mesh generation with Gid

The mesh generation in Gid is very similar to tutorial 1, so it is not repeated here. We just remark that the element size, as required in the Gid **Mesh - Generate** dialog should be set to 1.0 to get a reasonably fine mesh.

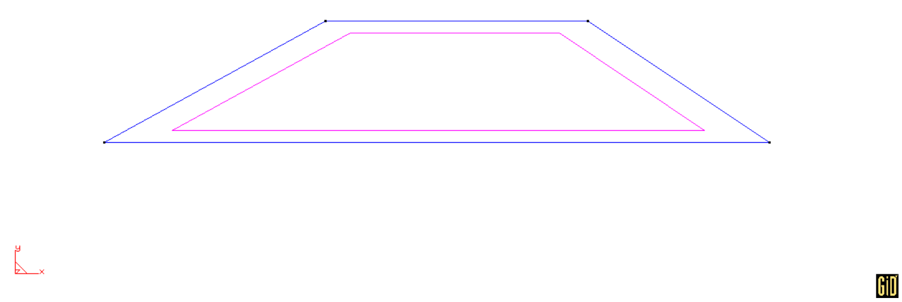


Figure 8: Nurbs surface of dam

Figure 8 shows the Nurbs surface. Figure 9 shows the mesh.

4.2 Input file

4.2.1 Initialization part

echo -yes
number_of_space_dimensions 2
groundflow_pressure
groundflow_saturation

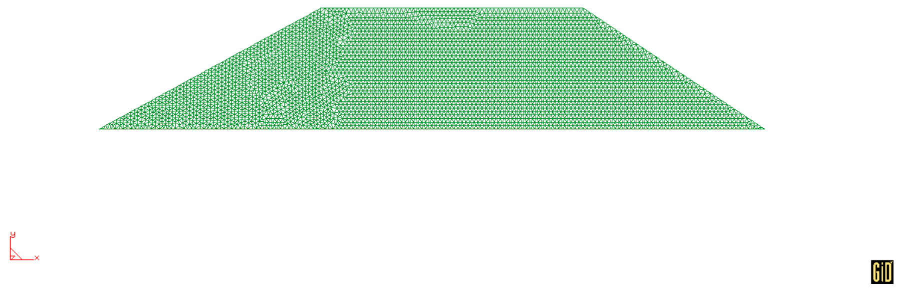


Figure 9: Mesh of dam

groundflow_velocity
end_initia

The **groundflow_pressure** takes care that the hydraulic head field becomes available over the entire domain. Initialization of the saturation rate is done with **groundflow_saturation**. This is needed since a non-saturated water model will be used. It is convenient to have the groundflow velocity vectors available for interpretation of the results, and thus we initialize **groundflow_velocity**.

4.2.2 Data part, geometry of edges

start_define
eps_geometry 1.e-1
end_define

start_define
left_edge geometry_line 10
end_define
left_edge 0 0 55 30 eps_geometry

start_define
right_dry_edge geometry_line 20
end_define
right_dry_edge 135 20 120 30 eps_geometry

We define geometrical lines where later the water boundary conditions will be applied.

4.2.3 Data part, some arithmetic expressions

start_arithmetic
rho 1.0
end_arithmetic

```
start_arithmetic
g -10.0
end_arithmetic
start_arithmetic
g_absolute 10.0
end_arithmetic
```

Here some parameters are specified with **start_arithmetic** ... **end_arithmetic**. The advantage of such arithmetic definitions for floating point parameters (that is real values), is that these later can be used with build in arithmetic operators in the Tochnog input file reader to calculate new parameters, which in turn can be used in the input file. Just read on, to see how this is used here.

4.2.4 Data part, gravity

```
force_gravity 0.0 g
```

The gravity components as specified in **force_gravity** are 0 in x-directions, and $-10.\frac{m}{s^2}$ in y-direction. Notice that the **g** as specified above with the arithmetic has been used here.

4.2.5 Data part, groundwater

```
groundflow_density rho
groundflow_seepage_geometry 0 -left_edge
```

The groundwater density ρ as specified in **groundflow_density** is $1.0\frac{1000kg}{m^3}$. On the total left edge there is air, so later we will prescribe in the boundary conditions that the pore pressure is zero at the left edge. However, we also need to be sure that the flux of water can only be directed outward the dam at the left edge, and water cannot enter the dam at the left edge. This is accomplished be using a seepage condition as specified in **groundflow_seepage_geometry**; with such condition Tochnog will take care that water flux can only be outward, and not inward.

4.2.6 Data part, boundary conditions on edges

```
bounda_dof 0 -right_wet_edge -pres
bounda_time 0 -200.0
bounda_dof 1 -left_edge -total_pressure
bounda_time 1 0.0
```

The **bounda_dof** records are used to prescribe conditions for the water pressures. On the part of the right edge below the water level a hydraulic head h of -200.0 is prescribed (the word **-pres** means hydraulic head). You can understand that is OK, by looking at the equation for the total pressure (pore pressure)

$$p_{\text{total}} = h - \rho g y$$

With density $\rho = 1.0$, gravity $g = -10.0$ we find at $y = 0.0$ that the pore pressure is -200.0, and at $y = 20.0$ that the pore pressure is 0.0.

On the total left edge we prescribe a zero pore pressure.

On the bottom edge we don't allow for any groundflow flux. In Tochnog a zero normal flux is imposed automatically if you don't specify anything, thus you don't see conditions at the bottom edge in the **bounda_*** records.

4.2.7 Data part, material properties

```
start_arithmetic
pe 1.e-5 DIVIDE rho DIVIDE g_absolute
end_arithmetic
```

The saturated permeability is $10^{-5}\frac{\mathrm{m}}{\mathrm{s}}$. Look, however, at the definition of the permeability for the input file in the user's manual. A division by **rho** and **g_absolute** is needed; this is easily accomplished with **start_arithmetic** ... **end_arithmetic**, so that the input file remains readable.

```
group_type 1 -groundflow
group_porosity 1 0.5
group_groundflow_permeability 1 pe pe
group_groundflow_nonsaturated_vangenuchten 0 0.0 1. 2.49 -0.140 1.507
```

The **group_type** is set to **-groundflow** which means that the groundflow storage equation will solved in the elements. The porosity is set to 0.5 with **group_porosity**. For the saturated permeability the **pe** as calculated in the **start_arithmetic** ... **end_arithmetic** is used. The **group_groundflow_nonsaturated_vangenuchten** records specifies the Van-Genuchten law for non-saturated groundwater properties.

4.2.8 Data part, prepare post-processing

```
post_calcul -groundflow_pressure -total_pressure
post_node 0 -node_rhside -sum -left_edge
```

The **post_calcul** record takes care of calculating extra post-processing data. This calculated data becomes available for printing or for plotting in Gid. See the user's manual for all possibilities of **post_calcul**. Here **post_calcul** is used to calculate the total pressure (=pore pressure); remember that primarily the hydraulic head is calculated in a groundflow analysis, so that the total pressure is only available as extra post processing option.

The **post_node** records are a convenient method to determine in structural calculations forces on prescribed displacement edges, or here in a groundflow calculation the groundwater flux to the environment. Using **post_node 0 -node_rhside -sum -left_edge** as shown here, causes the right-hand-side to be summed over all nodes which are located in the geometry **-left_edge**. The result will be placed in the record **post_node_result**, which can be printed during the calculation or found in the database after the calculation. You find in **post_node_result** the required flux at the position consistent with the initialized groundwater pressure variable. Look in **dof_label** after the calculation: there you see that the first variable is the hydraulic head **-pres**, and thus you find the groundwater flux also at the first position in the **post_node_result** record.

4.2.9 Data part, apply linear time-steps

control_timestep 10 1. 1.
control_groundflow_nonsaturated_apply 10 -no
control_groundflow_seepage_apply 10 -no
control_print 10 -time_current -post_node_rhside_ratio

This nonsaturated dam calculation is highly nonlinear because of the **group_groundflow_nonsaturated_vang** and the **groundflow_seepage_geometry** records. In order to get a solution a trick is commonly applied to this kind of calculation: first a linear solution is established, and afterwards the full nonlinear solution is searched iteratively with the linear solution as first guess.

We obtain a linear calculation with the extra records **control_groundflow_nonsaturated_apply 10** and **control_groundflow_seepage_apply 10**. These records de-activate the **group_groundflow_nonsatu** and **groundflow_seepage_geometry** for the time-steps of **control_timestep 10**. Notice that we print the current time **-time_current** and the error ratio **-post_node_rhside_ratio**. The error ratio is the ratio of the maximum out-of-balance flux at nodes without free hydraulic head and the maximum flux at nodes with prescribed hydraulic head.

4.2.10 Data part, apply nonlinear time-steps

control_timestep 20 1. 100.
control_print 20 -time_current -post_node_rhside_ratio

After the linear solution was obtained with **control_timestep 10**, the time-steps as specified in **control_timestep 20** are used to find the nonlinear solution including non-saturation and the seepage condition on the left edge.

4.2.11 Data part, include mesh generated with Gid

include mesh.gid/mesh.dat

4.2.12 Data part, final remarks for advanced users

target_item 1 -post_node_result 0 0
target_value 1 -1.45083e-05 1.e-7

Here we check with **target_item** and **target_value** that the flux of groundflow to the left edge has the expected value. In case a new Tochnog release would have a bug, that will be automatically detected by running this tutorial if the real value at running the calculation differs from the expected value as set in **target_item** and **target_value**. When distributing a new release, we take care that tutorial calculations, and also other calculations, meet their expected values.

4.3 Run calculation

On Linux open a window, and type **tochnog tutorial_2**. On Microsoft Windows open a DOS command prompt, and type **tochnog.exe tutorial_2**. After the calculation is ready you will see the following files in the directory:

- **tutorial_2.dbs**
  This is the database after the calculation. It contains all information from the calculation (input data, results, etc.).
- **tutorial_2_flavia.msh** and **tutorial_2_flavia.res**
  Post-processing files which can be opened with Gid.
- **tochnog.log**
  Log file from Tochnog which lists the calculations that have been run.

4.4 Output results

4.4.1 Value of the groundwater flux out of the left edge

After the calculation the **post_node_result 0** record in the **tutorial_2.dbs** contains the groundwater flux out of the left edge as first value in the record. The value is -1.45e-05 where the minus sign indicates that the flow goes outwards.

4.4.2 Plot results in Gid

Open the Gid post processing files in the usual way (see tutorial 1).

<u>Plot the saturation in Gid</u>

View results - Contour Fill - groundflow saturation

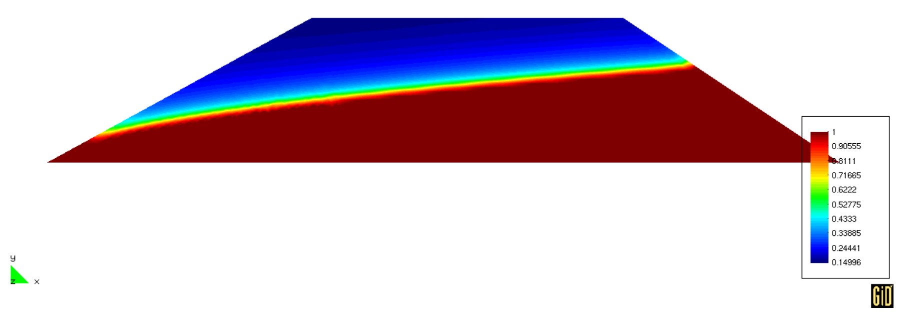


Figure 10: Groundwater saturation in dam

From figure 10 is is clear that the lower part of the dam is saturated, whereas above the phreatic level the dam is non-saturated. In this calculation with **group_groundflow_nonsaturated_vangenuchten** the phreatic level follows automatically as a result of the calculation; this is opposed to most geotechnical calculations where there are only fully saturated zones below an a priori known phreatic level which is specified in the input file.

Plot the pore pressure in Gid

View results - Contour Fill - topres

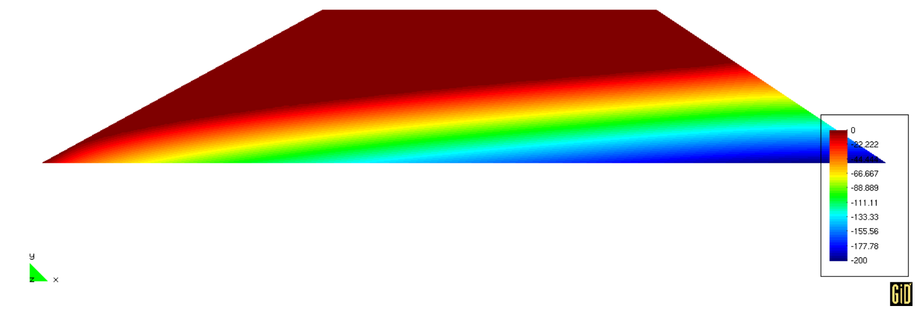


Figure 11: Groundwater pore pressure in dam

Plot the groundflow velocity in Gid

View results - Vectors - groundflow velocity

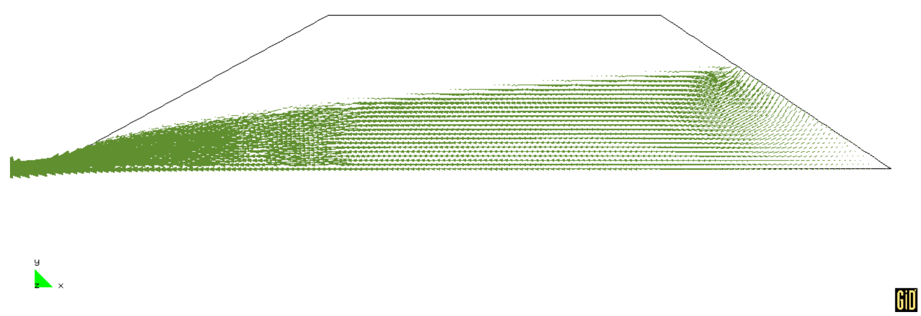


Figure 12: Groundwater velocity in dam

The vectors of groundflow velocity in figure 12 clearly show that the flux exits the dam at a relatively small part of the left edge. The highest node where groundwater starts to exit the dam is located at (6.98,3.81).

5 Tutorial 3: excavation with sheet pile, beam and contact spring elements

This tutorial is taken from [2]. An experiment is done with a sheet pile near an excavation. The following is done:

- the sheet pile is completely inserted.
- two excavations are done.
- at point A the sheet pile is fixed in x-direction (in the experiment a horizontal strut is fixed to point A).
- three more excavations are carried out
- the vertical load at the top surface is applied.
- point A of the sheet pile is moved to the left, to find when the soil at the right of the sheet-pile will become unstable.

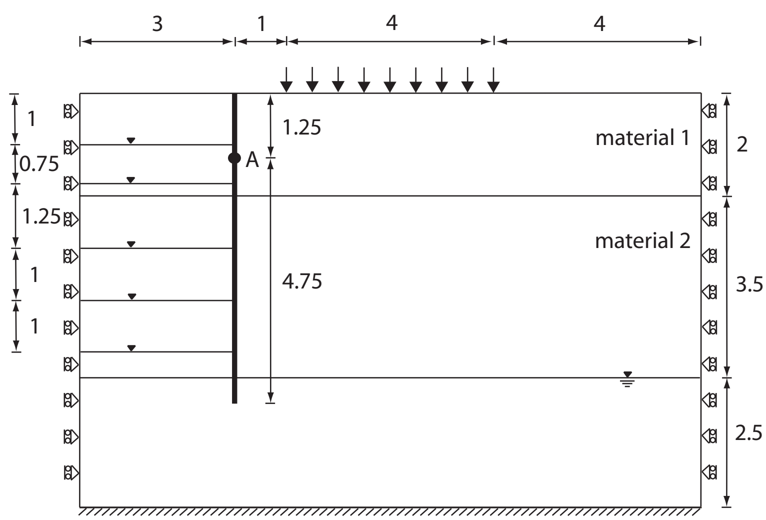


Figure 13: Excavation with sheet pile

The dimension are given in figure 13. At the left side of the sheet pile you see the five excavations levels. All excavations remain above the phreatic water level, which you can see at $y = 2.5$. Especially notice point A at which the sheet pile after the second excavation is supported, and later is moved to the left.

5.1 Input file

5.1.1 Initialization part

echo -yes
number_of_space_dimensions 2
materi_velocity
materi_displacement
materi_strain_plasti
materi_plasti_kappa

materi_stress
groundflow_pressure
beam_rotation
end_initia

Notice in the initialization part that we initialized **materi_*** for an elasto-plastic calculation, together with **groundflow_pressure** for groundwater pressure analysis. Additionally for the sheet pile we will use truss-beam elements, so that we also need to initialize **beam_rotation**.

5.1.2 Data part, tolerance on geometries

start_arithmetic
eps_geometry 1.e-5
end_arithmetic

For convenience we define **eps_geometry 1.e-5**, which will be used later as tolerance for geometries.

5.1.3 Data part, beam geometries

start_define
beam_point geometry_point counter_a
end_define
start_define
beam_line geometry_line counter_a
end_define
start_define
beam_short_line geometry_line counter_a
end_define

beam_point 3.0 6.75 eps_geometry
beam_line 3.0 2.0 3.0 8.0 eps_geometry
beam_short_line 3.0 2.01 3.0 8.0 eps_geometry

The **beam_point** defines the position of point A. With **beam_line** we define a geometrical line completely along the sheet-pile, which will be used to automatically generate truss-beam elements and contact springs. The **beam_line_short** defines a geometry line a bit shorter than the sheet-pile; the purpose of this short line will become clear later.

The **counter_a** is a convenient way to give a unique index to each geometry, without keeping tack of used numbers for indices yourself. Initially the **counter_a** is 0, and each time it is used it will be automatically incremented with 1 by Tochnog. So here **beam_point geometry_point counter_a** is be read as **beam_point geometry_point 0**, **beam_line geometry_line counter_a** is read as **beam_line geometry_line 1**, etc. Since we later only will refer to the defined words **beam_point**, **beam_line**, etc. those will be read as **geometry_point 0**, **geometry_line 1**, etc. at each occurrence, and there is no need to know the specific indices that are used.

5.1.4 Data part, edge geometries

```
start_define
lower_edge geometry_line counter_a
end_define
start_define
right_edge geometry_line counter_a
end_define
start_define
left_edge geometry_line counter_a
end_define
start_define
upper_edge_load geometry_line counter_a
end_define

lower_edge 0.0 0.0 12.0 0.0 eps_geometry
right_edge 12.0 0.0 12.0 8.0 eps_geometry
left_edge 0.0 0.0 0.0 8.0 eps_geometry
upper_edge_load 4.0 8.0 8.0 8.0 eps_geometry
```

The edges **lower_edge**, **right_edge** and **left_edge** will be used to impose boundary conditions. The **upper_edge_load** will be used to impose the top load.

5.1.5 Data part, excavation geometries

```
start_define
excavation_0 geometry_quadrilateral counter_a
end_define
start_define
excavation_1 geometry_quadrilateral counter_a
end_define
start_define
excavation_2 geometry_quadrilateral counter_a
end_define
start_define
excavation_3 geometry_quadrilateral counter_a
end_define
start_define
excavation_4 geometry_quadrilateral counter_a
end_define

excavation_0 0.0 7.0 3.0 7.0 0.0 8.0 3.0 8.0 eps_geometry
excavation_1 0.0 6.25 3.0 6.25 0.0 8.0 3.0 8.0 eps_geometry
excavation_2 0.0 5.0 3.0 5.0 0.0 8.0 3.0 8.0 eps_geometry
excavation_3 0.0 4.0 3.0 4.0 0.0 8.0 3.0 8.0 eps_geometry
excavation_4 0.0 3.0 3.0 3.0 0.0 8.0 3.0 8.0 eps_geometry
```

Here five **geometry_quadrilateral** are defined, which will be used for the excavations. Please pay close attention how the coordinates of the quadrilaterals are specified; for **excavation_0** the first coordinate is **0.0 7.0**, the second coordinate is **3.0 7.0**, the third coordinate is **0.0 8.0** and finally the fourth coordinate is **3.0 8.0**. If you would specify these coordinates in another order you get false results for the excavations.

5.1.6 Data part, material properties

```
start_define
soil0_group 0
end_define
group_type soil0_group -materi -groundflow
group_materi_memory soil0_group -total_linear
group_materi_density_groundflow soil0_group 1.937 1.702
group_materi_elasti_young soil0_group 20.e3
group_materi_elasti_poisson soil0_group 0.3
group_materi_plasti_tension_direct soil0_group 1.0
group_materi_plasti_mohr_coul soil0_group 0.593 3.e0 0.105
group_groundflow_permeability soil0_group 0.1 0.1
```

Later when generating elements we will assign elements between $y = 6$ and $y = 8$ to group 0, and the elements below $y = 6$ to group 1. Here, for convenience and clarity, we use a define of **soil0_group** for 0, and a define of **soil1_group** for 1. One data records is new: the **group_materi_density_groundflow** is used to specify the wet and dry density; Tochnog will automatically use the wet density $1.937 \frac{1000.kg}{m^3}$ if it detects that an element of the group is below the phreatic level, whereas the dry density $1.702 \frac{1000.kg}{m^3}$ will be used if the element is above the phreatic level.

Side remark: for the **group_groundflow_permeability** we can use arbitrary values in this specific tutorial, since the calculated groundwater pore pressure here will not be influenced by the permeability value.

```
start_define
soil1_group 1
end_define
group_type soil1_group -materi -groundflow
group_materi_memory soil1_group -total_linear
group_materi_density_groundflow soil1_group 1.937 1.702
group_materi_elasti_young soil1_group 30.e3
group_materi_elasti_poisson soil1_group 0.3
group_materi_plasti_tension_direct soil1_group 1.0
group_materi_plasti_mohr_coul soil1_group 0.698 3.e0 0.209
group_groundflow_permeability soil1_group 0.1 0.1
```

For group 1 we define a **soil1_group**.

```
start_define
sheet_pile_group 2
end_define
group_type sheet_pile_group -truss_beam
group_beam_memory sheet_pile_group -total_linear
group_beam_inertia sheet_pile_group 2.085e-4 2.085e-4 1.
group_beam_young sheet_pile_group 9.87e6
```

group_beam_shear sheet_pile_group 9.87e6
group_truss_memory sheet_pile_group -total_linear
group_truss_area sheet_pile_group 0.223
group_truss_young sheet_pile_group 9.87e6

Later truss-beam elements will be generated for the sheet pile, and these truss-beam elements will be assigned to group 2. For this group 2 we define a **sheet_pile_group**. The **group_type** is set to **-truss_beam**, so that both truss forces and beam moments will be calculated. Setting **-total_linear** for **group_beam_memory** indicates that small deformation theory needs to be used for the beam moments and forces. The beam area moments of inertia and polar moment of torsion are specified in **group_beam_inertia**; in fact the polar moment of torsion can be a dummy value since torsion is not relevant in this specific calculation. The beam young modulus and shear modulus (for the torsion moment calculation) are specified in the data records **group_beam_young** and **group_beam_shear**. Also for the truss force we apply small deformation theory, as set by **-total_linear** in **group_truss_memory**. The truss cross section area is specified in **group_truss_area**, and the young modulus in **group_truss_young**.

5.1.7 Data part, boundary conditions

bounda_dof 0 -lower_edge -vely
bounda_dof 1 -right_edge -velx
bounda_dof 2 -left_edge -velx
bounda_dof 3 -beam_point -velx
bounda_time 3 2.0 0. 5.0 0. 5.000001 -0.005 1.e8 -0.005
bounda_dof 4 -all -rotx -roty

In this calculation it is more convenient to prescribe velocities (instead of prescribing displacements), since in the experiment the velocity of point A has been prescribed. The **bounda_dof 0**, **bounda_dof 1** and **bounda_dof 2** specify conditions equal to the slope calculation of tutorial 1. The **bounda_dof 3** needs more explanation. It specifies the velocity of point A of the sheet pile in time. Up to time 2.0, so during the first two excavations, point A is not prescribed. Then at time 2.0 it is fixed till time 5.0, so till the end of the loading at the top. Then after time 5.0 point A gets a prescribed velocity of $-0.005\frac{m}{s}$, so to the left. The **bounda_dof 4** suppresses two beam rotations which are not relevant in this 2D calculation; only the **-rotz** is relevant since it corresponds with in-plane 2D bending of the beam.

There is one subtle but *important* remark to be made. We impose the boundary condition for point A at the defined **beam_point**, so in fact at a **geometry_point** (see the previous definition of **beam_point**). This will only work correctly if at that point in space there really is a finite elements and their nodes. If on that point in space there is no node, no condition will be applied at all. This remark holds for all kind of data where you use geometries, e.g. loads on edges, post-processing on geometries, etc.; in the end, such data will only be applied on finite element nodes, so these nodes should be located on the correct locations.

5.1.8 Data part, distributed force load on top

force_edge 0 0.0 -10.0
force_edge_geometry 0 -upper_edge_load
force_edge_time 0 3.0 0.0 4.0 1.0 1.e8 1.0

The **force_edge 0** records specifies a distributed edge force of $-10.0\frac{kN}{m^2}$ in y-direction (so in fact in negative y-direction). This force will be applied to the element edges which are part of the **-upper_edge_load** as specified in **force_edge_geometry 0**. Since this force is only applied in the experiment after time 3.0, we need the **force_edge_time 0** data records which contains a time versus multiplication factor diagram. Using this **force_edge_time** diagram, before time 3.0 the load is 0.0, between time 3.0 and time 4.0 the load increases from 0.0 to $-10.0\frac{kN}{m}$, and it remains $-10.0\frac{kN}{m}$ after time 4.0.

5.1.9 Data part, gravity

force_gravity 0. -9.81

The gravity acceleration is $-9.81\frac{m^2}{s}$, and is present directly from the start of the calculation. We do this because later we will directly set initial fields for the groundwater pressure and material stresses at the start of the calculation with **control_dof_reset** records, so that the gravity makes equilibrium with the pressure and stresses.

5.1.10 Data part, groundwater properties

groundflow_density 1.0
groundflow_phreatic_level 0.0 2.5 15.0 2.5

The groundwater density is $1.0\frac{1000kg}{m^3}$. The level of groundwater is constant between ($x = 0.0m$, $y = 2.5m$) and ($x = 15.0m$, $y = 2.5m$).

5.1.11 Data part, post processing and printing

post_point 0 3.0 6.75
post_node 0 -node_rhside -sum -beam_point
post_calcul -groundflow_pressure -total_pressure
print_gid_contact_spring2 1

To obtain exactly at point A information about the solved solutions fields (displacements, stresses, etc), we specify **post_point 0** at **3.0 6.75**. Tochnog will then automatically find out where in the FE mesh this post point is located, and calculates there the solution field by interpolating results from the specific element in which the post point is located. These results will be placed in the record **post_point_dof**, which in turn is available in the database after the calculation, or can be printed with **control_print** or **control_print_data_versus_data** during the calculation.

The **post_node 0** record sums the **node_rhside** record on all nodes present in the geometry **beam_point**, so in this case we directly get the **node_rhside** record at the node located at point A. The **node_rhside** contains for nodes with prescribed velocity (or displacement) the external reaction force on the prescribed node (for the positions in the **node_rhside** record that matches the velocity unknowns, so position 0 and 1 in this calculation, see **dof_label** in the database after the calculation). The result for **post_node 0** will be placed in the record **post_node_result 0**, which again can be viewed in the database after the calculation or printed during the calculation.

Similar to the dam calculation in tutorial 2 we need a **post_calcul -groundflow_pressure -total_pressure**.

Finally, with **print_gid_contact_spring2 1** we require that Gid plots the contact springs as one-node elements, even if they are in the calculation really two-node elements; this is done because Gid draws two-node elements with their length, and since the length of the contact springs is 0 the contact springs would become invisible when drawn with two nodes.

5.1.12 Data part, generate mesh with Tochnog

control_mesh_macro 10 -rectangle soil1_group 7 25
control_mesh_macro_parameters 10 1.5 3.0 3.0 6.0
control_mesh_macro_element 10 -quad4

control_mesh_macro 20 -rectangle soil0_group 7 9
control_mesh_macro_parameters 20 1.5 7.0 3.0 2.0
control_mesh_macro_element 20 -quad4

control_mesh_merge 30 -yes
control_mesh_merge_macro_generate 30 10 20

Tochnog has some build in mesh generation possibilities which are convenient for relatively easy meshes. If the mesh is complex, Gid works better.

With **control_mesh_macro 10**, **control_mesh_macro_parameters 10** and **control_mesh_macro_element** we specify that a rectangular mesh should be generated, with **quad4** elements and the elements should be assigned to group 1. The rectangle has middle point ($x = 1.5$,$y = 3.0$), width 3.0 and height 6.0; the number of nodes in width direction, so x-direction, is 7 and the number of nodes in height direction, so y-direction, is 25. Similar, the records with index 20 generate a second rectangular mesh. These 10 and 20 records together generate the left part of the mesh, that is all **quad4** elements to the left of the sheet pile.

The two rectangular meshes each have nodes at a common line $y = 6.0$, so in fact there are duplicate nodes at this line. The **control_mesh_merge 20** takes care that these duplicate nodes are merged, so that at the common line the meshes connect to the same nodes, and solution field like velocities, displacement and water pressure are continuous. With **control_mesh_merge_macro_generate 20** we ensure that the meshes as specified with the macro's with index 10 and 20 are merged.

control_mesh_macro 40 -rectangle soil1_group 28 25
control_mesh_macro_parameters 40 7.5 3.0 9.0 6.0
control_mesh_macro_element 40 -quad4

control_mesh_macro 50 -rectangle soil0_group 28 9
control_mesh_macro_parameters 50 7.5 7.0 9.0 2.0
control_mesh_macro_element 50 -quad4

control_mesh_merge 60 -yes
control_mesh_merge_macro_generate 60 40 50

With the macro's with indices 40 and 50 rectangular meshes are generated to the right of the sheet pile. Again we merge the top and bottom mesh at the common line at $y = 6.0$.

control_mesh_merge 70 -yes

control_mesh_merge_macro_generate 70 10 40
control_mesh_merge_not 70 -beam_short_line

Up to now we only merged rectangular meshes along $y = 6.0$, so at their common horizontal line. Here we merge the left and right meshes at their common line along $x = 3.0$. However, to allow for slip along the sheet pile, we do not want to merge nodes along the location of the sheet pile, so that slip and different displacements at the sheet pile remains possible. Thus the merging with index 70, is not done at the sheet pile, by including **control_mesh_merge_not 70**. In fact we use a geometrical line a bit shorter than the actual length of the sheet pile. The reason for this is that the nodes at the end of the sheet-pile can either be considered to be part of the sheet-pile, or part of the soil underneath the sheet-pile. We prefer to consider these nodes the be part of the soil underneath the sheet-pile, and so prefer to merge these nodes.

control_mesh_generate_truss_beam 80 sheet_pile_group -beam_line
control_mesh_generate_truss_beam_loose 80 -yes
control_mesh_generate_truss_beam_macro 80 10 20

control_mesh_generate_contact_spring 90 contact_spring_group -beam_line
control_mesh_generate_contact_spring_element 90 -quad4 -truss_beam

With the previous macro and merge records we have generated **quad4** elements for the soil. Here with the **control_mesh_generate_truss_beam 80** record we generate the truss-beam elements for the sheet-pile line, and should be assigned to the group **sheet_pile_group**.. The **control_mesh_generate_truss_beam_loose 80** record tells Tochnog that the generated truss-beam elements should not be fixed to the existing nodes of the **quad4** elements, but should get their own new nodes; this will allow for slip between the sheet-pile **truss_beam** nodes and the **quad4** soil element nodes. The **control_mesh_generate_truss_beam_macro 80** record tells Tochnog that only truss elements should be placed near nodes which have been generated from a macro with index 10 or 20 (if we would not do this, then twice too much truss-beam elements).

Contact springs between **truss_beam** nodes and **quad4** nodes are generated with the **control_mesh_generate_contact_spring 90** records.

Figures 14 shows the generated mesh, and 15 gives a schematic drawing how the **truss_beam** elements are connected to the **quad4** elements. The detail drawing with the springs in figure 15 is really only for clarity; in reality the nodes of the **truss_beam** elements and the **quad4** elements have the same position in space, but that would not give a very clear drawing.

5.1.13 Data part, set gravity stresses

control_reset_dof 100 -pres
control_reset_value_constant 100 -24.525

control_reset_dof 101 -sigyy
control_reset_value_multi_linear 101 0.0 -114.825 2.5 -91.85 8.0 0.0

control_reset_dof 102 -sigxx -sigzz
control_reset_value_multi_linear 102 0.0 -49.21071429 2.5 -39.36428571 8.0 0.0

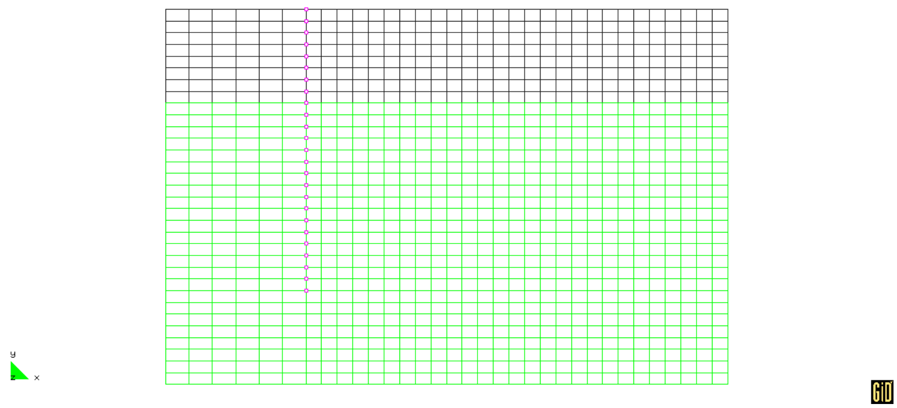


Figure 14: Mesh with quadrilateral elements, truss_beam and contact springs

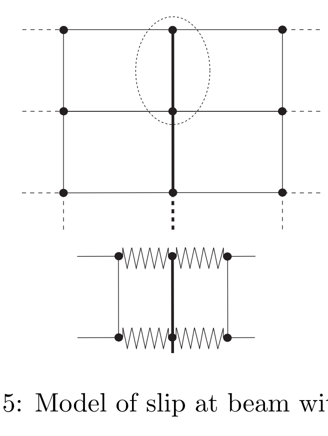


Figure 15: Model of slip at beam with springs

control_print_gid 120 -separate_index

The **control_reset_dof 100** and **control_reset_value_constant 100** records together set a constant hydraulic head of **24.525** consistent with the phreatic level at $y = 2.5m$. The **control_reset_dof 102** and **control_reset_value_multi_linear** 1012 records together set the multi-linear gravity field for the stresses.

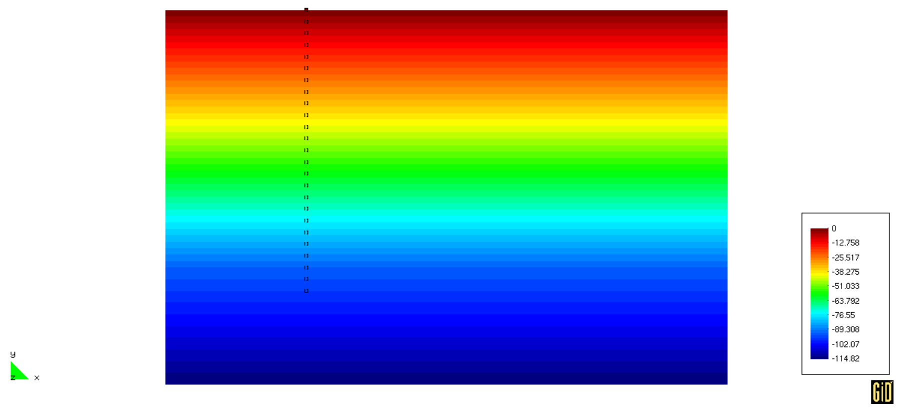


Figure 16: Gravity vertical stress field sigyy

Plot 16 shows the vertical stress, as plotted with Gid.

5.1.14 Data part, apply excavation time-steps

control_timestep 200 1.e-2 1.0
control_print 200 -time_current -post_node_rhside_ratio
control_mesh_delete_geometry 200 -excavation_0
control_mesh_delete_geometry_element 200 -quad4

control_print_data_versus_data 210 -time_current 0 0
-post_point_dof 0 -disx
-post_node_result 0 -velx

control_print_gid 220 -separate_index

We discuss here how the first excavation can be done in Tochnog; the other excavations are done quite similar, and you can find them in the input file **tochnog/test/tutorial/tutorial_3.dat**.

The **control_timestep 200** records specifies that time steps of size $1.e-2s$ should be taken until a time increment of $1.0s$ has been done. This increment value of $1.0s$ is arbitrary, since we do not use mass inertia in the calculation and so time is only a pseudo loading variable to specify the sequence of events in time. With **control_print 200** we ask for printing of the current time in the calculation, and also the error ratio (see tutorial 1 for a discussion on the error ratio, or otherwise see the users manual). The **-post_point_dof 0 -disx** denotes that from all **-post_point_dof** records the one with index 0 should be used, and from that record the first value should be taken (numbering inside records starts of 0 in Tochnog). The **-time_current 0 0** is a bit tricky: since there is only one **-time_current** record you need to specify a dummy 0 for the index, and the second 0 denotes again the first value in the record, so in fact the current time. Now the first excavation area as specified by the geometry quadrilateral **excavation_0** is actually excavated by the record

control_mesh_delete_geometry 200; record **control_mesh_delete_geometry_element** takes care that only **quad4** soil elements will be excavated, we don't want to excavate the **truss_beam** elements (which are also located in the geometry quadrilateral).

control_reset_dof 330 -disx -disy

In the experiment the reported displacements are relative to the deformations after the second excavation. In the calculation this is obtained by the **control_reset_dof** record, which sets the displacements to null after the second excavation.

5.1.15 Data part, apply load time-steps

control_timestep 700 1.e-2 1.0
control_print 700 -time_current -post_node_rhside_ratio -post_node_result -post_point_dof
control_print_data_versus_data 700 -time_current 0 0 -post_point_dof 0 -disx -post_node_result 0 -velx
control_print_gid 720 -separate_index

Loading time steps of size $1.e-1s$ are taken till a time increment of $1.0s$ is reached.

The **control_print_data_versus_data** record is a convenient method to obtain from the calculation a text file containing several columns of information, from whatever data records that you like. Here it is used to get columns with the current time in the calculation, the x-displacement at point A and the reaction force at point A.

When the loading steps are ready, we print Gid plotting files with **control_print_gid 720**.

5.1.16 Data part, apply velocity point A time-steps

control_timestep 700 5.e-2 1.
control_print 700 -time_current -post_node_rhside_ratio -post_node_result -post_point_dof
control_print_data_versus_data 700 -time_current 0 0 -post_point_dof 0 -disx -post_node_result 0 -velx
control_print_gid 720 -separate_index

5.1.17 Data part, print pressure on beam

control_print_dof_line 900 -yes
control_print_dof_line_coordinates 900 3.01 2.0 3.01 8.0
control_print_dof_line_n 900 10

control_print_dof_line 910 -yes
control_print_dof_line_coordinates 910 2.99 2.0 2.99 8.0
control_print_dof_line_n 910 10

To obtain X-Y graphs of the pressure of the soil on the left side and right side of the sheet-pile, the **control_print_dof_line_*** data records are used.

With index 900, printing of all solution fields is demanded, for the line starting at point ($x = 3.01$,$y = 2.0$) and ending at ($x = 3.01$,$y = 8.0$), so that is a line just to the right of the sheet-pile. The solution fields are printed at 10 points along that line. The fields are printed in text files, by example, **velx.900**, ..., **sigxx.900**, etc.

Similar with index 910 the solution fields will be printed along a line just to the left side of the sheet-pile.

5.2 Run calculation

On Linux open a window, and type **tochnog tutorial_3**. On Microsoft Windows open a DOS command prompt, and type **tochnog.exe tutorial_3**.

See also tutorial 1 for the type of files that you can find after the calculation.

5.3 Output results

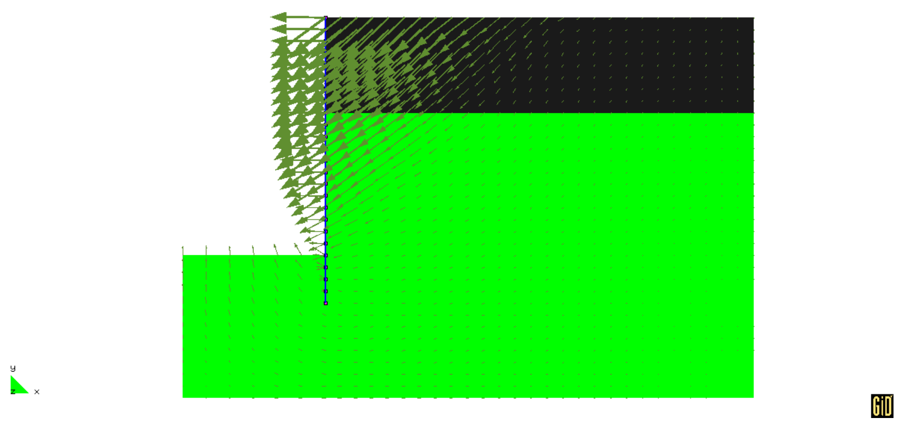


Figure 17: Material displacement mesh

View results - Display Vectors - materi displacement

In the post processor of Gid look at the displacements at the end of the calculation. Especially notice that at sheet pile you see two displacements vectors; one is a vector of the sheet-pile displacement, and the other is the vector of the soil displacement near the sheet-pile. The sheet-pile refuges to be compressed (because it is very stiff), so the sheet-pile displacement vector is nearly horizontal. The soil slips over the sheet-pile, so the displacement vector of the soil points also downwards.

Windows - View Results - Main Mesh - Deformed - Materi Mesh Deform

It is in Gid also possible to draw the deformed FE mesh. In figure 18 we used a factor of 20 to get a clear view of the deformations.

Windows - View Results - Main Mesh - Original

Set the mesh back to original to draw the undeformed mesh again.

View Results - Contour Fill - Materi plasti kappa

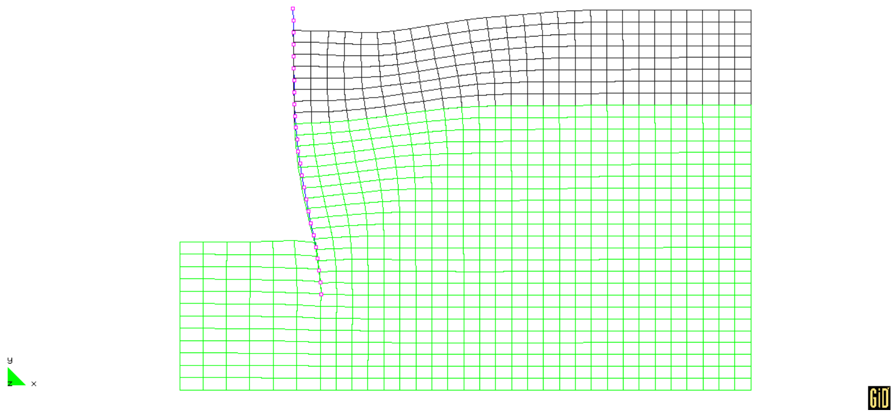


Figure 18: Deformed mesh

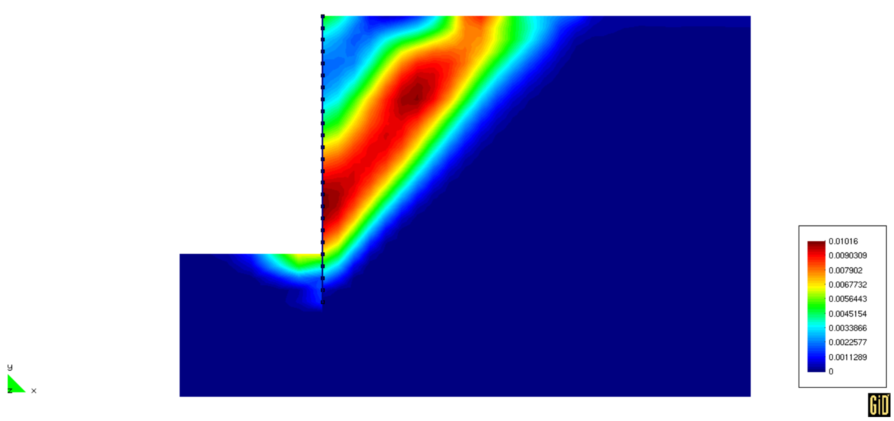


Figure 19: Material effective plastic strain, kappa

The effective plastic strain shows the Mohr-Coulomb shear failure line, caused by moving the sheet-pile at point A to the left, see 19.

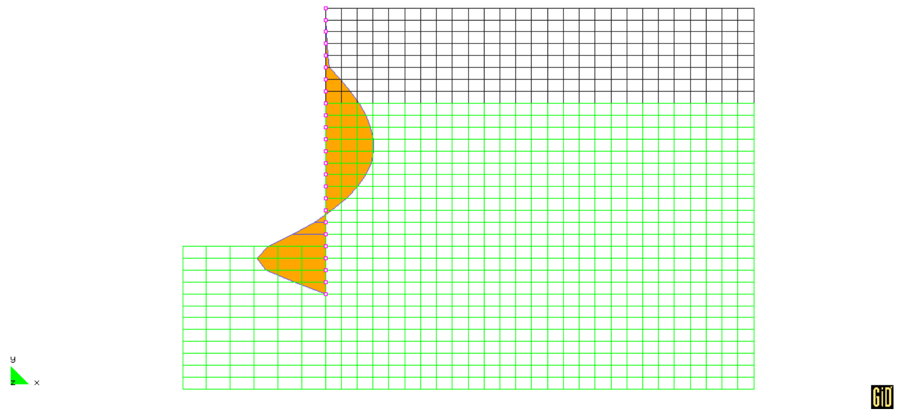


Figure 20: Beam bending moment

View Results - Line Diagram - Scalar - beam force moment 5

Using a line diagram in Gid, the beam moment can be displayed, see figure 20. You need to select **beam force moment 5**, since these values represent the moment around the z-axis in the **element_beam_force_moment** record, see the users manual.

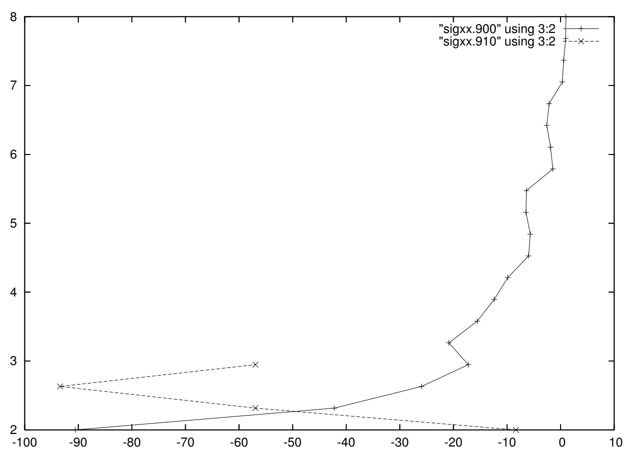


Figure 21: Pressure on sheet pile, left and right

gnuplot

p "sigxx.900" using 2:3 with linespoints, "sigxx.910" using 2:3 with linespoints

set term postscript

set output "post_line_dof_sigxx.ps"

replot

With **Gnuplot** an X-Y graph of the soil pressures on the sheet-pile is obtained, see figure 21. The second and third column from the **sigxx.900** and **sigyy.910** need to be used since those contain the y coordinate and the xx stress. Notice the increase of the pressure near the end of the sheet-pile. As an exercise you can try to run the calculation with more elements, to see how much results for the pressure on the sheet-pile change with a finer mesh.

6 Tutorial 4: excavation with sheet pile, isoparametric and interface elements

6.1 Input file

In this tutorial we show the same analysis as in tutorial 3. However, in stead of **truss_beam** elements we will use **quad4** isoparametric elements to model the sheet pile. And in stead of **contact_spring2** elements we will use **quad4** interface elements to model the slip between sheet pile and soil.

We will not repeat all the details of the analysis, only the differences with tutorial 3 will be highlighted.

6.1.1 Initialization part

```
echo -yes
number_of_space_dimensions 2
materi_velocity
materi_displacement
materi_strain_plasti
materi_plasti_kappa
materi_stress
groundflow_pressure
end_initia
```

Please notice that we did not initialize **beam_rotation**, since will be not use truss beam elements in this tutorial.

6.1.2 Data part, using linear test calculations

We included the definition of **test_calculation** , which we either set to **true** or **false**. Then using **start_if test_calculation** ... **end_if** we can perform a fast linear test calculation when **test_calculation** is set to **true**. In this way, it is easy to test whether the excavations, go well under linear elastic conditions (no plasticity, etc.), and check if linear elastic displacements, stresses etc. are plausible. Once the linear elastic calculation is verified, the **test_calculation** can be set to **false** and the full non-linear calculation can be done. You are encouraged to follow the same strategy.

```
start_define
test_calculation false
end_define
start_if test_calculation
linear_calculation_apply -yes
end_if
```

6.1.3 Data part, geometries

Since we will not use truss beams, the beam geometries in the data part do not need to be defined. Only the edge geometries and excavation geometries are defined.

6.1.4 Data part, material properties

The material properties for the soil are the same as in tutorial 3.

For the sheet pile we will use isoparametric **quad4** elements with a large fictive thickness. As density we simply take the soil density, we is OK since most of the fictive thickness in reality will actually consist of soil. The young modulus we fitted in such way that the bending behavior of the isoparametric **quad4** elements is the same as that of the **truss_beam** elements of tutorial 3 (in fact, we have simply put a force at the bottom of the **quad4** elements and **truss_beam** elements, and tuned the young modulus of the **quad4** elements such that the same displacement was obtained with both models).

```
start_define
sheet_pile_group 2
end_define
group_type sheet_pile_group -materi
group_materi_memory sheet_pile_group -total_linear
group_materi_density sheet_pile_group 1.702
group_materi_elasti_young sheet_pile_group 5.9874e6
```

For the slip between soil and sheet pile we will use interface elements. The stiffness of interface elements should be taken high enough, so that the deformations in the interface elements are relatively small. On the other hand, the stiffness of interface elements should not be too large, because that would lead to a bad conditioned matrix and numerical problems. The interfaces are not allowed to transfer tension stresses, and slip frictional forces are restricted by a Mohr-Coulomb law; the friction angle of the Mohr-Coulomb law is taken such that $\tan(phi_{\text{interface}}) = 0.5 * \tan(phi_{\text{soil}})$.

```
start_define
interface_group 3
end_define
group_type interface_group -materi
group_interface interface_group -yes
group_interface_materi_memory interface_group -total_linear
group_interface_materi_elasti_stiffness interface_group 10.e10 5.e10
group_interface_materi_plasti_mohr_coul_direct interface_group 0.4636 1. 0.
group_interface_materi_plasti_tension interface_group -no
```

For convergence of the calculation, it is required to use a quite small interface stiffness.

6.1.5 Data part, generate mesh with Tochnog

The commands for mesh generation are as follows:

```
control_mesh_macro 10 -rectangle soil1_group 7 9
control_mesh_macro_parameters 10 1.5 1.0 3.0 2.0
control_mesh_macro_element 10 -quad4

control_mesh_macro 20 -rectangle soil1_group 7 17
control_mesh_macro_parameters 20 1.5 4.0 3.0 4.0
control_mesh_macro_element 20 -quad4

control_mesh_macro 30 -rectangle soil0_group 7 9
control_mesh_macro_parameters 30 1.5 7.0 3.0 2.0
control_mesh_macro_element 30 -quad4

control_mesh_macro 40 -rectangle soil1_group 2 9
control_mesh_macro_parameters 40 3.05 1.0 0.1 2.0
control_mesh_macro_element 40 -quad4

control_mesh_macro 50 -rectangle sheet_pile_group 2 17
control_mesh_macro_parameters 50 3.05 4.0 0.1 4.0
control_mesh_macro_element 50 -quad4

control_mesh_macro 60 -rectangle sheet_pile_group 2 9
control_mesh_macro_parameters 60 3.05 7. 0.1 2.0
control_mesh_macro_element 60 -quad4

control_mesh_macro 70 -rectangle soil1_group 28 9
control_mesh_macro_parameters 70 7.55 1.0 8.9 2.0
control_mesh_macro_element 70 -quad4

control_mesh_macro 80 -rectangle soil1_group 28 17
control_mesh_macro_parameters 80 7.55 4.0 8.9 4.0
control_mesh_macro_element 80 -quad4

control_mesh_macro 90 -rectangle soil0_group 28 9
control_mesh_macro_parameters 90 7.55 7.0 8.9 2.0
control_mesh_macro_element 90 -quad4

control_mesh_merge 95 -yes

control_mesh_generate_interface 96 interface_group soil0_group sheet_pile_group
control_mesh_generate_interface 97 interface_group soil1_group sheet_pile_group
```

The generation of the sheet pile group is new (see control index 50 and 60); it generates **quad4** isoparametric elements which model the sheet pile. Further, look how simple the interfaces between the sheet pile group and soil group are generated (see control index 96 and 97); it generates **quad4** interface elements which model the sliding between soil and sheet pile.

Figure 22 shows the generated mesh.

6.1.6 Data part, post processing and printing

In this tutorial we use isoparametric **quad4** elements to model the sheet pile. Isoparametric elements primarily calculate stresses, so that forces and moments are not directly available. To get the forces and moments in the sheet pile, a post calculation is needed. The following **post_calcul** commands perform this calculation:

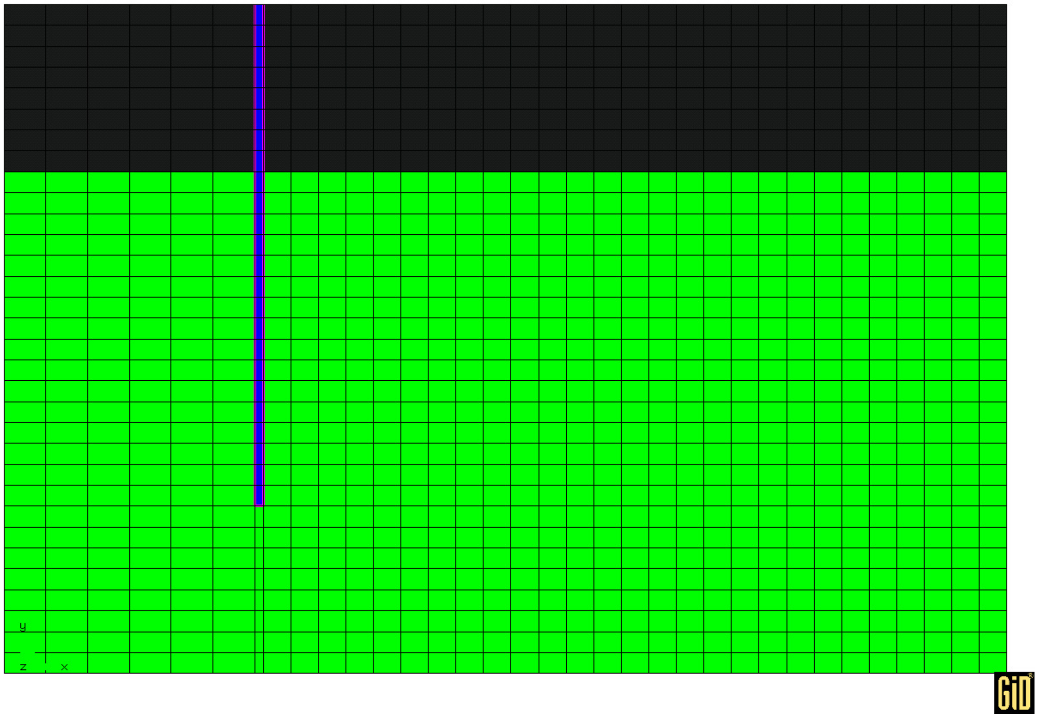


Figure 22: Mesh with soil elements (green,black), sheet pile elements (blue), and interface elements (purple)

```
post_calcul -groundflow_pressure -total_pressure -materi_stress -force
post_calcul_materi_stress_force_element_group sheet_pile_group
post_calcul_materi_stress_force_reference_point 100 4.
```

The **post_calcul -materi_stress -force** tells Tochnog that is should calculate forces and moments from initially calculated stresses. The group for which this should be done is specified by **post_calcul_materi_stress_force_element_group sheet_pile_group**. Finally, it is needed to specify a reference point for drawing plots in GID, so that positive and negative vectors for forces and moments are drawn in a unique manner.

6.1.7 Data part, timesteps

Timesteps are much the same as in the previous tutorial. We now take special action, however, in case the **test_calculation** is set to **true**. In that case simply a timestep of size 1 is taken for imposing gravity and excavations. In this way a linear test calculation can be performed very fast, and you can check if linear results are plausible. The lines below show how this is done:

```
start_define
dtime 2.e-4
end_define
start_if test_calculation
control_timestep 700 1. 1.0
end_if
```

start_if_not test_calculation
control_timestep 700 dtime 1.0
end_if_not

The remaining part of the input file equals that of tutorial 3.

6.2 Run calculation

See tutorial 3.

6.3 Output results

View Results - Display Vectors - momsig

Using a display vector in Gid, the beam moment can be displayed, see figure 23. You need to select **momsig**, which is generated by the **post_calcul** commands.

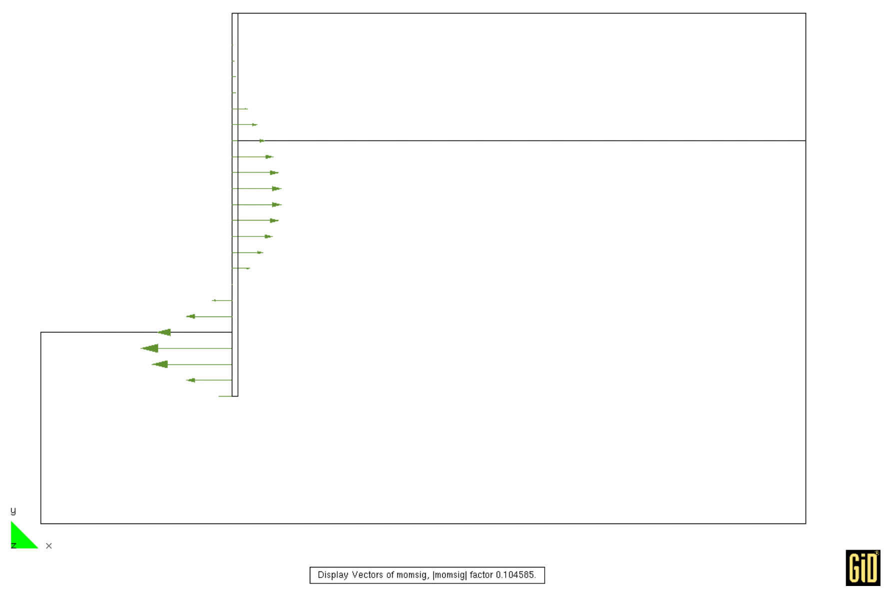


Figure 23: Beam bending moment

References

[1] D.V. Griffiths, P.A. Lane, *1999 Slope stability analysis by finite elements* Geotechnique 49, no. 3, pp. 387-403

[2] I. Shahrour, S. Ghorbanbeigi, P.A. von Wolffersdorff, *1995 Comportament des rideaux de palplanche: experimentation en vraie grandeur et predictions numeriques* Revue francaise de Geotechnique, no. 71, pp 39-47