Contents

1 Basic information .................................................. 44
   1.1 How to perform a calculation and how to get started ............. 44
   1.2 Pre- and postprocessing ..................................... 44
   1.3 Space discretization, time discretization ....................... 45
   1.4 Program capabilities ........................................ 45
   1.5 Files used by Tochnog ....................................... 46

2 Equations .......................................................... 47
   2.1 Convection and diffusion of heat ................................ 47
      2.1.1 Convection-diffusion equation ........................... 47
      2.1.2 Convection to environment ................................ 47
      2.1.3 Radiation to environment ................................ 47
   2.2 Material deformation and flow ................................ 48
      2.2.1 Memory .................................................... 48
         Total Lagrange ............................................... 48
         Updated Lagrange ............................................ 48
      2.2.2 Elasticity ................................................. 50
         Isotropy ....................................................... 50
         Transverse Isotropy ......................................... 50
         Nonlinear elasticity polynomials ............................ 50
         Power law nonlinear elasticity ............................... 50
         Borja Tamagnini nonlinear elasticity ....................... 51
         Lade nonlinear elasticity .................................... 51
      2.2.3 Elasto-Plasticity ......................................... 52
         CamClay plasticity model ..................................... 52
         Cap1 plasticity model ......................................... 54
         Cap2 plasticity model ......................................... 54
         Compression limiting plasticity model ....................... 55
         di Prisco plasticity model ................................... 55
         di Prisco plasticity model with varying density ............ 56
Drucker-Prager plasticity model ............................................. 57
Generalised Non Associate CamClay for Bonded Soils plasticity model ............................................. 57
Gurson plasticity model ......................................................... 57
Hardening-Soil plasticity model ................................................ 57
Mohr-Coulomb plasticity model ................................................. 58
Mohr-Coulomb hardening-softening plasticity model ................. 59
Multilaminate plasticity model ................................................ 59
Tension limiting plasticity model ............................................. 60
Von-Mises plasticity model ..................................................... 61
Isotropic Hardening and softening ........................................... 61
Kinematic Hardening ............................................................ 61
Plastic heat generation ........................................................... 61

2.2.4 Hypo-Plasticity .............................................................. 62
Masin law ........................................................................... 62
Masin clay law .................................................................... 63
Wolffersdorff law ............................................................... 66
Wolffersdorff law - extended Niemunis version ...................... 67
Wolffersdorff pressure dependent initial void ratio extension ..... 67
Niemunis visco law ............................................................... 67
Cohesion extension ............................................................... 69
Intergranular strains extension .............................................. 69
ISA-Intergranular strains extension ....................................... 70

2.2.5 Damage ................................................................. 71
Mazars ............................................................................... 71

2.2.6 Average stress (hydrostatic compressibility) .................. 72
Compressibility contribution .................................................. 72

2.2.7 Undrained groundflow analysis ..................................... 72
Local capacity ................................................................. 72

2.2.8 Thermal stresses ......................................................... 73
Expansion ................................................................. 73

2.2.9 Hyper elasticity .......................................................... 73
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3</td>
<td>derivatives (third record of initialization part, if specified)</td>
<td>88</td>
</tr>
<tr>
<td>4.4</td>
<td>beam_rotation</td>
<td>88</td>
</tr>
<tr>
<td>4.5</td>
<td>condif_temperature</td>
<td>89</td>
</tr>
<tr>
<td>4.6</td>
<td>groundflow_pressure</td>
<td>89</td>
</tr>
<tr>
<td>4.7</td>
<td>groundflow_pressure_gradient</td>
<td>89</td>
</tr>
<tr>
<td>4.8</td>
<td>groundflow_saturation</td>
<td>89</td>
</tr>
<tr>
<td>4.9</td>
<td>groundflow_velocity</td>
<td>89</td>
</tr>
<tr>
<td>4.10</td>
<td>materi_damage</td>
<td>89</td>
</tr>
<tr>
<td>4.11</td>
<td>materi_acceleration</td>
<td>89</td>
</tr>
<tr>
<td>4.12</td>
<td>materi_displacement</td>
<td>89</td>
</tr>
<tr>
<td>4.13</td>
<td>materi_displacement_relative</td>
<td>90</td>
</tr>
<tr>
<td>4.14</td>
<td>materi_history_variable number_of_variables</td>
<td>90</td>
</tr>
<tr>
<td>4.15</td>
<td>materi_maxwell_stress number_of_chains</td>
<td>90</td>
</tr>
<tr>
<td>4.16</td>
<td>materi_plasti_camclay_history</td>
<td>90</td>
</tr>
<tr>
<td>4.17</td>
<td>materi_plasti_cap1_history</td>
<td>90</td>
</tr>
<tr>
<td>4.18</td>
<td>materi_plasti_diprisco_history number_of_history_variables</td>
<td>90</td>
</tr>
<tr>
<td>4.19</td>
<td>materi_plasti_f</td>
<td>90</td>
</tr>
<tr>
<td>4.20</td>
<td>materi_plasti_f_nonlocal</td>
<td>90</td>
</tr>
<tr>
<td>4.21</td>
<td>materi_plasti_generalised_non_associate_cam_clay_for_bonded_soils_history</td>
<td>91</td>
</tr>
<tr>
<td>4.22</td>
<td>materi_plasti_hardsoil_history</td>
<td>91</td>
</tr>
<tr>
<td>4.23</td>
<td>materi_plasti_hypo_history</td>
<td>91</td>
</tr>
<tr>
<td>4.24</td>
<td>materi_plasti_kappa</td>
<td>91</td>
</tr>
<tr>
<td>4.25</td>
<td>materi_plasti_kappa_shear</td>
<td>91</td>
</tr>
<tr>
<td>4.26</td>
<td>materi_plasti_laminate number_of_laminates</td>
<td>92</td>
</tr>
<tr>
<td>4.27</td>
<td>materi_plasti_phimob</td>
<td>92</td>
</tr>
<tr>
<td>4.28</td>
<td>materi_plasti_rho</td>
<td>92</td>
</tr>
<tr>
<td>4.29</td>
<td>materi_strain_energy</td>
<td>92</td>
</tr>
<tr>
<td>4.30</td>
<td>materi_strain_elasti</td>
<td>92</td>
</tr>
<tr>
<td>4.31</td>
<td>materi_strain_intergranular</td>
<td>92</td>
</tr>
<tr>
<td>4.32</td>
<td>materi_strain_isa_c</td>
<td>92</td>
</tr>
<tr>
<td>4.33</td>
<td>materi_strain_isa_eacc</td>
<td>93</td>
</tr>
</tbody>
</table>
4.34 materi_strain_plasti
4.35 materi_strain_plasti_camclay
4.36 materi_strain_plasti_cap
4.37 materi_strain_plasti_compression
4.38 materi_strain_plasti_diprisco
4.39 materi_strain_plasti_generalised_non_associate_cam_clay_for_bonded_soils
4.40 materi_strain_plasti_druckprag
4.41 materi_strain_plasti_hardsoil
4.42 materi_strain_plasti_laminate_mohr_coul
4.43 materi_strain_plasti_laminate_tension
4.44 materi_strain_plasti_mohr_coul
4.45 materi_strain_plasti_tension
4.46 materi_strain_plasti_vonmises
4.47 materi_strain_total
4.48 materi_strain_total_kappa
4.49 materi_strain_total_compression_kappa
4.50 materi_strain_total_shear_kappa
4.51 materi_strain_total_tension_kappa
4.52 materi_stress
4.53 materi_stress_pressure_history
4.54 materi_velocity
4.55 materi_velocity_integrated
4.56 materi_void_fraction
4.57 materi_work
4.58 mrange maximum_range_length
4.59 mstring maximum_number_of_strings
4.60 wave_scalar
4.61 wave_fscalar
4.62 end_initia (last record of initialization part)

5 Input file, data part, introduction

Arithmetic blocks start_arithmetic ...end_arithmetic
Automatic counting: counter_a, etc. .......................................................... 98
Conditional blocks start_if ... end_if and start_if_not ... end_if_not .... 98
Control indices .................................................................................. 99
Define blocks start_define ... end_define ............................................. 99
Include files ..................................................................................... 100
Numbering of values in records .......................................................... 100
Ranges -ra ... -ra ............................................................................ 101
Types of dof’s .................................................................................. 101

6 Input file, data part, data records ....................................................... 103

6.1 area_element_group index geometry_entity_item geometry_entity_index element_group .......................................................... 103
6.2 area_element_group_element index name .................................................. 104
6.3 area_element_group_interface index switch .......................................... 104
6.4 area_element_group_method index method .......................................... 104
6.5 area_element_group_node index node_0 node_1 ... element_group .... 104
6.6 area_element_group_time index switch ............................................... 104
6.7 area_element_group_sequence index element_0 element_1 ... .... 104
6.8 area_element_group_sequence_element index name ......................... 105
6.9 area_element_group_sequence_element_group index group_0 group_1 ... 105
6.10 area_element_group_sequence_geometry index geometry_entity_item geometry_entity_index .................................................. 107
6.11 area_element_group_sequence_geometry_method index method .......... 107
6.12 area_element_group_sequence_interface index switch ....................... 107
6.13 area_element_group_sequence_time index time_0 time_1 ... ............... 107
6.14 area_node_dataitem index geometry_entity_item geometry_entity_index index data_item_name .......................................................... 107
6.15 area_node_dataitem_double index value_0 value_1 ... ....................... 107
6.16 area_node_dataitem_integer index value_0 value_1 ... ..................... 107
6.17 axisymmetric switch .................................................................. 108
6.18 bounda_alternate index bounda_index_0 bounda_index_1 ... ............ 108
6.19 bounda_baseline_correction time_start time_end ............................... 109
6.20 bounda_baseline_correction_parameters index ... ............................ 109
6.21 bounda_constant index switch ....................................................... 109
6.22 `bounda_dof index node_range dof_0 dof_1 ...` .......................................... 110
6.23 `bounda_dof_cylindrical index x_first y_first z_first x_second y_second z_second` 111
6.24 `bounda_dof_radial index x y z` ................................................................. 111
6.25 `bounda_factor index a_o a_1 ...a_n` ......................................................... 111
6.26 `bounda_factor_parabolic_x index a_o a_1 a_2` ........................................... 112
6.27 `bounda_force index node_range dof_0 dof_1 ...` ...................................... 112
6.28 `bounda_found index found` ................................................................. 112
6.29 `bounda_geometry_method index node_type` .............................................. 112
6.30 `bounda_normal index normal_x normal_y normal_z` .................................. 112
6.31 `bounda_print_mesh_dof dof_0 dof_1 ...` .................................................. 113
6.32 `bounda_print_mesh_dof_geometry geometry_item_name geometry_item_index` 113
6.33 `bounda_print_mesh_dof_values value_dof_0 value_dof_1 ...` ...................... 113
6.34 `bounda_sine index start_time end_time freq_0 amp_0 freq_1 amp_1 ...` ...... 113
6.35 `bounda_time index time load time load ...` .............................................. 113
6.36 `bounda_time_factor index factor` ............................................................. 114
6.37 `bounda_time_offset index time_offset` ...................................................... 114
6.38 `bounda_time_increment index time_increment` ......................................... 114
6.39 `bounda_time_units factor_time factor_length` .......................................... 114
6.40 `bounda_time_smc index switch` ............................................................... 114
6.41 `bounda_time_smc_offset index time_offset` .............................................. 116
6.42 `bounda_time_smc_units factor_time factor_length` .................................. 116
6.43 `bounda_time_user index switch` .............................................................. 116
6.44 `bounda_water index switch` ................................................................. 116
6.45 `change_dataitem index data_item_name data_item_index data_item_number_0` data_item_number_1 ... operation ......................................................... 116
6.46 `change_dataitem_geometry index geometry_entity_name geometry_entity_index` 117
6.47 `change_dataitem_time index time value ...` .............................................. 117
6.48 `change_dataitem_time_discrete index switch` ........................................... 117
6.49 `change_dataitem_time_method index method` ............................................ 117
6.50 `change_dataitem_time_user index switch` ............................................... 118
6.51 `check_data switch` .............................................................................. 118
6.52 `check_error switch` .............................................................................. 118
6.80 condif_heat_edge_normal_time index time load time load ... .......................... 123
6.81 condif_heat_volume index heat ................................................................. 123
6.82 condif_heat_volume_element index element_0 element_1 ... ....................... 123
6.83 condif_heat_volume_element_group index element_group ............................. 123
6.84 condif_heat_volume_factor index a_0 a_1 ... a_n ....................................... 123
6.85 condif_heat_volume_geometry index geometry_name geometry_index ............ 123
6.86 condif_heat_volume_sine index start_time end_time freq_0 amp_0 freq_1 ... 123
6.87 condif_heat_volume_time index time load time load ... .............................. 123
6.88 condif_heat_volume_user index switch ....................................................... 124
6.89 condif_heat_volume_user_parameters index ... .......................................... 124
6.90 condif_radiation_edge_normal index α_r T_r ........................................... 124
6.91 condif_radiation_edge_normal_element index element_0 element_1 ... ........ 124
6.92 condif_radiation_edge_normal_element_node index element node_0 node_1 ... 124
6.93 condif_radiation_edge_normal_element_group index element_group_0 element_group_1 ... 124
6.94 condif_radiation_edge_normal_element_side index element_0 element_1 ... side ... 124
6.95 condif_radiation_edge_normal_geometry index geometry_entity_name geometry_entity_index ................................................................. 125
6.96 condif_radiation_edge_normal_node index node_0 node_1 ... ......................... 125
6.97 contact_apply index switch ................................................................. 125
6.98 contact_heat_generation factor ................................................................. 125
6.99 contact_penalty_pressure pressure_penalty ................................................ 125
6.100contact_penalty_temperature temperature_penalty ..................................... 125
6.101contact_penalty_velocity velocity_penalty ................................................ 126
6.102contact_plasti_friction friction ................................................................. 126
6.103contact_target_element_group element_group_0 element_group_1 ... 126
6.104contact_target_geometry index geometry_entity_item geometry_entity_index 126
6.105contact_target_geometry_switch index switch ........................................... 127
6.106contact_bounda_relax index switch .......................................................... 127
6.107contact_bounda_relax_geometry index geometry_item_name geometry_item_index 127
6.108 control_check_data index switch .............................. 127
6.109 control_contact_apply index switch .......................... 127
6.110 control_convection_apply index switch ........................ 128
6.111 control_data_activate index data_item_name_0 data_item_name_1 ... switch 128
6.112 control_data_arithmetic index data_item_name data_item_index data_item_number operat .............................. 128
6.113 control_data_arithmetic_double index val .............................. 128
6.114 control_data_copy index data_item_from data_item_to .............................. 128
6.115 control_data_copy_factor index factor .............................. 128
6.116 control_data_copy_index index data_item_from index_from data_item_to index_to .............................. 129
6.117 control_data_copy_index_factor index factor .............................. 129
6.118 control_data_delete index data_item_name index_range .............................. 129
6.119 control_data_put index data_item_name index_range number_0 number_1 ... 129
6.120 control_data_put_double index ... .............................. 130
6.121 control_data_put_integer index ... .............................. 130
6.122 control_data_save index switch .............................. 130
6.123 control_dependency_apply index switch .............................. 130
6.124 control_distribute index distribution_type data_item_name data_item_index data_item_number .............................. 131
6.125 control_distribute_correlation_distance index maximum_distance .............................. 132
6.126 control_distribute_correlation_length index correlation_length ... .............................. 132
6.127 control_distribute_minimum_maximum index minimum maximum .............................. 132
6.128 control_distribute_parameters index mean_value standard_deviation .............................. 132
6.129 control_distribute_seed index seed .............................. 132
6.130 control_element_group_apply index number .............................. 132
6.131 control_geometry_moving index initialise .............................. 132
6.132 control_groundflow_consolidation_apply index switch .............................. 133
6.133 control_groundflow_nonsaturated_apply index switch .............................. 133
6.134 control_inertia_apply index switch_0 switch_1 ... .............................. 133
6.135 control_input index switch .............................. 133
6.136 control_interface_gap_apply index switch .............................. 133
6.137 control_materi_damage_apply index switch ........................................... 134
6.138 control_materi_dynamic index factor .................................................... 134
6.139 control_materi_elasti_k0 index switch ............................................... 134
6.140 control_materi_failure_apply index switch ........................................... 134
6.141 control_materi_plasti_hypo_masin_ocr_apply index switch ..................... 134
6.142 control_materi_plasti_hypo_masin_clay_ocr_apply index switch ............... 134
6.143 control_materi_plasti_hardsoil_gammap_initial index switch .................. 134
6.144 control_materi_plasti_hypo_pressure_dependent_void_ratio index switch ... 134
6.145 control_materi_plasti_hypo_niemunis_visco_ocr_apply index switch ........ 135
6.146 control_materi_plasti_hypo_substepping index switch ......................... 135
6.147 control_materi_plasti_tension_apply index switch ................................ 135
6.148 control_materi_plasti_visco_apply index switch .................................. 135
6.149 control_materi_updated_apply index switch ........................................ 135
6.150 control_materi_undrained_apply index switch ................................... 135
6.151 control_materi_viscosity_apply index switch ...................................... 135
6.152 control_mesh_activate_gravity_apply index index_0 index_1 ...................... 135
6.153 control_mesh_adjust_geometry index geometry_entity_item_0 
    geometry_entity_index_0 geometry_entity_item_1 
    geometry_entity_index_1 .................................................................. 136
6.154 control_mesh_change_element_group index element_group_0 element_group_1
6.155 control_mesh_convert index switch ...................................................... 136
6.156 control_mesh_convert_element_group index element_group_0 element_group_1
6.157 control_mesh_convert_quad9_quad6 index dir ........................................ 137
6.158 control_mesh_convert_tria6_tria3 index switch ................................... 138
6.159 control_mesh_copy index move_x move_y move_z ................................. 138
6.160 control_mesh_cut_geometry index geometry_item_name geometry_item_index
6.161 control_mesh_cut_force index switch_0 switch_1 switch_2 ....................... 138
6.162 control_mesh_delete_element index number_0 number_1 ....................... 138
6.163 control_mesh_delete_geometry index geometry_entity_item geometry_entity_index
6.164 control_mesh_...
6.164 control_mesh_delete_geometry_direct index switch

6.165 control_mesh_delete_geometry_element index element_name_0 element_name_0

6.166 control_mesh_delete_geometry_element_group index element_group_0 element_group_1 ...

6.167 control_mesh_delete_geometry_factor index factor_0 factor_1 ...

6.168 control_mesh_delete_geometry_method index method

6.169 control_mesh_delete_geometry_move_node index switch

6.170 control_mesh_delete_geometry_projection_type index type

6.171 control_mesh_delete_geometry_stop index switch

6.172 control_mesh_delete_geometry_stop_geometry index geometry_entity_name

6.173 control_mesh_delete_small index eps

6.174 control_mesh_duplicate_element_group index element_group_old element_group_new

6.175 control_mesh_element_group_apply index group_0 group_1 ...

6.176 control_mesh_extrude index z0 z1 z2 ...

6.177 control_mesh_extrude_direction index dir

6.178 control_mesh_extrude_element index name

6.179 control_mesh_extrude_contact_spring_element_group index element_group_0 element_group_1 ...

6.180 control_mesh_extrude_contact_spring_element_group_new index element_group_new_0 element_group_new_1 ...

6.181 control_mesh_extrude_element_group_new index element_group_old_0 element_group_old_1 ... element_group_new_00 element_group_new_01 ... element_group_new_10 element_group_new_11 ...

6.182 control_mesh_extrude_n index n0 n1 n2 ...

6.183 control_mesh_generate_beam index element_group geometry_entity_item geometry_entity_index

6.184 control_mesh_generate_contact_spring index element_group geometry_entity_item geometry_entity_index

6.185 control_mesh_generate_contact_spring_element index element_0 element_1

6.186 control_mesh_generate_contact_spring_element_group index element_group_0 element_group_1 ...

6.187 control_mesh_generate_interface index element_group_0 element_group_00 element_group_01 element_group_1 element_group_10 element_group_11 ...

6.188 control_mesh_generate_interface_method index method_select method_generate
6.189 `control_mesh_generate_spring1` index element_group geometry_entity_item geometry_entity_index
6.190 `control_mesh_generate_spring2` index element_group geometry_entity_item geometry_entity_index
6.191 `control_mesh_generate_truss` index element_group geometry_entity_item geometry_entity_index
6.192 `control_mesh_generate_truss_beam` index element_group geometry_entity_item geometry_entity_index
6.193 `control_mesh_generate_truss_beam_loose` index switch
6.194 `control_mesh_generate_truss_beam_macro` index macro_0 macro_1 ...
6.195 `control_mesh_generate_truss_beam_separate` index switch
6.196 `control_mesh_gid_batch` index switch
6.197 `control_mesh_interface_triangle` index switch
6.198 `control_mesh_keep_element` index element_0 element_1 ...
6.199 `control_mesh_keep_element_group` index element_group_0 element_group_1 ...
6.200 `control_mesh_keep_geometry` index geometry_item_name geometry_item_index
6.201 `control_mesh_keep_node` index node_0 node_1 ...
6.202 `control_mesh_macro` index macro_item element_group
6.203 `control_mesh_macro_concentrate` index ...
6.204 `control_mesh_macro_element` index element_type
6.205 `control_mesh_macro_parameters` index x y ...
6.206 `control_mesh_map` index switch
6.207 `control_mesh_merge` index switch
6.208 `control_mesh_merge_eps_coord` index epsilon
6.209 `control_mesh_merge_macro_generate` index macro_0 ...
6.210 `control_mesh_merge_geometry` index geometry_entity_item geometry_entity_index
6.211 `control_mesh_merge_geometry_not` index geometry_entity_item geometry_entity_index
6.212 `control_mesh_mirror` index axis
6.213 `control_mesh_move` index move_x move_y move_z
6.214 `control_mesh_multiply` index number_of_multiplications
6.215 `control_mesh_refine_globally` index refinement_type
6.216 control mesh refine globally geometry index geometry_entity_item
   geometry_entity_index ................................................... 152

6.217 control mesh refine locally index percentage ............................. 152

6.218 control mesh refine locally dof index dof ............................... 152

6.219 control mesh refine locally geometry index geometry_entity_item
   geometry_entity_index ................................................... 153

6.220 control mesh refine locally minimal size index minimal_size ............. 153

6.221 control mesh refine locally not index geometry_entity_0
   geometry_entity_index ................................................... 153

6.222 control mesh refine locally not method index method .......................... 153

6.223 control mesh refine locally only index geometry_entity_0
   geometry_entity_index ................................................... 153

6.224 control mesh refine locally only method index method .......................... 153

6.225 control_mesh_remove index method element_group_0 element_group_1
   element_group_2 ... ...................................................... 153

6.226 control mesh remove_geometry index geometry_item_name
   geometry_item_index ................................................... 154

6.227 control mesh remove_keep_geometry index geometry_item_name
   geometry_item_index ................................................... 154

6.228 control mesh remove_really index switch ....................................... 154

6.229 control mesh remove_really_activate_all index switch .......................... 154

6.230 control mesh remove_really_activate_factor index factor ...................... 154

6.231 control mesh renumber index lowest_element lowest_node .................... 155

6.232 control mesh renumber_element_geometry_offset index offset ................. 155

6.233 control mesh renumber_element_group_offset index offset ...................... 155

6.234 control mesh rotate index n ............................................. 155

6.235 control mesh rotate_angle index angle ..................................... 155

6.236 control mesh split index switch .......................................... 156

6.237 control mesh split_element_from index name ................................... 156

6.238 control mesh split_element_to index name ................................... 156

6.239 control mesh split only index geometry_entity geometry_entity_index ......... 156

6.240 control mesh truss distribute mpc index switch ............................... 156

6.241 control mesh truss distribute mpc air index switch .............................. 157

6.242 control mesh truss distribute mpc dof index dof_0 dof_1 ... .................. 157

6.243 control mesh truss distribute mpc_element_group_truss index element_group_0
   element_group_1 ... ...................................................... 158
6.244 control_mpc_distribute_mpc_element_group_isoparametric
   index element_group_0 element_group_1 ... ........................................ 158

6.245 control_mpc_distribute_mpc_exact index switch ..................................... 158

6.246 control_mpc_distribute_mpc_exact_minimal_length index tolerance
   ........................................................................................................ 158

6.247 control_mpc_distribute_mpc_exact_minimal_length_connect
   index switch ...................................................................................... 158

6.248 control_mpc_distribute_mpcgeometry_truss
   index geometry_entity_name_0 geometry_entity_index_0 geometry_entity_name_1
game_entity_index_1 ... ........................................................................ 158

6.249 control_mpc_distribute_mpcgeometry_isoparametric
   index geometry_entity_name_0 geometry_entity_index_0 geometry_entity_name_1
game_entity_index_1 ... ........................................................................ 158

6.250 control_mpc_apply index switch ............................................................. 158

6.251 control_mpc_element_group index switch ................................................. 159

6.252 control_plasti_apply index switch .......................................................... 159

6.253 control_post index switch ...................................................................... 159

6.254 control_post_apply index switch ............................................................ 159

6.255 control_post_element_force index switch ................................................. 159

6.256 control_print index data_item_name_0 data_item_name_1 ...
   ........................................................................................................ 159

6.257 control_print_beam_force_moment index switch ........................................ 160

6.258 control_print_beam_force_moment_coordinates index x_start y_start z_start
   x_end y_end z_end ................................................................................ 160

6.259 control_print_beam_force_moment_switch index switch ............................... 160

6.260 control_print_database index switch ......................................................... 160

6.261 control_print_database_method index method .......................................... 160

6.262 control_print_data_versus_data index data_item_name_0 index_0 number_0
data_item_name_1 index_1 number_1 ... .................................................. 161

6.263 control_print_dof index switch .................................................................. 161

6.264 control_print_dof_id index switch ............................................................ 162

6.265 control_print_dof_smooth_dof index dof_0 dof_1 ...
   ........................................................................................................ 162

6.266 control_print_dof_smooth_n index number_of_smoothings .......................... 162

6.267 control_print_dof_line index switch ........................................................ 162

6.268 control_print_dof_line_coordinates index x_0 y_0 z_0 x_1 y_1 z_1 x_2
   y_2 z_2 ... .......................................................................................... 163
control_print_dof_line_element_group
index element_group_0 element_group_1

control_print_dof_line_eps_iso
index eps_iso

control_print_dof_line_method
index node_type

control_print_dof_line_n
index n

control_print_dof_line_time
index switch

control_print_dof_point
index switch

control_print_dof_point_coordinates
index x y z

control_print_dof_point_time
index switch

control_print_dof_rhside
index switch

control_print_element
index data_item_name

control_print_element_method
index method

control_print_filter
index print_filter_index_0 print_filter_index_1 ...

control_print_frd
index switch

control_print_frd_freecad
index switch

control_print_frd_prepomax
index switch

control_print_frequency_timeinterval
index timeinterval

control_print_frequency_timestep
index timestep

control_print_gid
index switch

control_print_gid_batch
index switch

control_print_gid_beam_vectors
index switch

control_print_gid_beam_vectors_normal
index normal_x normal_y normal_z

control_print_gid_contact_spring2
index number_of_nodes

control_print_gid_coord
index switch

control_print_gid_dof
index initialisation_name_0 initialisation_name_1 ...

control_print_gid_dof_calcul
index calcul_0 calcul_1 ...

control_print_gid_element_group
index element_group_0 element_group_1

control_print_gid_element_mpc
index switch

control_print_gid_empty
index switch

control_print_gid_group
index switch
6.298 control_print_gid_mesh_activate_gravity index switch .......................... 170
6.299 control_print_gid_method index method ............................................. 170
6.300 control_print_gid_node_method index method ...................................... 170
6.301 control_print_gid_other index switch .................................................. 170
6.302 control_print_gid_save_difference index switch .................................. 170
6.303 control_print_gid_safety_slip_critical index switch ............................. 171
6.304 control_print_gid_smooth_dof index dof_0 dof_1 ... ............................ 171
6.305 control_print_gid_smooth_n index number_of_smoothings ...................... 171
6.306 control_print_gid_spring2 index number_of_nodes ................................ 171
6.307 control_print_gid_truss_vector index switch ....................................... 171
6.308 control_print_gid_truss_vector_normal index normal_x normal_y normal_z 171
6.309 control_print_gmsh index switch ......................................................... 171
6.310 control_print_gmsh_dummy index switch .............................................. 172
6.311 control_print_gmsh_element_data index switch .................................... 172
6.312 control_print_gmsh_node_method index method .................................... 172
6.313 control_print_history index data_item_name_0 data_item_index_0 number_0 173
6.314 control_print_history_relative_time index tr ....................................... 173
6.315 control_print_interface_stress index switch ........................................ 173
6.316 control_print_interface_stress_2d_coordinates index x_start y_start x_end y_end 174
6.317 control_print_interface_stress_3d_geometry index geometry_item_name geometry_item_index ................................................................. 174
6.318 control_print_interface_stress_3d_order index order ............................ 174
6.319 control_print_materi_stress_force index method ................................... 174
6.320 control_print_mesh_dof index switch .................................................. 174
6.321 control_print_node index data_item_name number_0 number_1 ... ............ 174
6.322 control_print_node_angular index switch_x switch_y switch_z ............... 175
6.323 control_print_node_angular_middle index x_middle y_middle z_middle ....... 175
6.324 control_print_node_geometry index geometry_item_name geometry_item_index 175
6.325 control_print_node_sort index sort_method ......................................... 175
6.326 control_print_node_zero index switch ................................................. 175
6.327 control_print_number_iterations index switch ............... 175
6.328 control_print_partialname index data_item_name_0 data_item_name_1 ...
6.329 control_print_tecplot index switch .......................... 176
6.330 control_printvtk index switch ................................. 176
6.331 control_printvtk_coord index switch ......................... 177
6.332 control_printvtk_dof index initialisation_name_0 initialisation_name_1 ...
6.333 control_printvtk_dof_calcul index calcul_0 calcul_1 ...
6.334 control_printvtk_empty index switch ......................... 178
6.335 control_print_gid_group index switch ....................... 178
6.336 control_printvtk_node_method index node_type .............. 178
6.337 control_printvtk_other index switch ......................... 178
6.338 control_relaxation index relax_0 relax_1 ................. 178
6.339 control_repeat index number_of_repeats control_index .... 179
6.340 control_repeat_save index data_item_name_0 data_item_index_0 data_item_number_0 data_item_name_1 data_item_index_1 data_item_number_1 ...
6.341 control_repeat_save_calculate index switch ............... 179
6.342 control_reset_dof index dof_0 dof_1 .......................... 179
6.343 control_reset_element_group index element_group_number_0 element_group_number_1 ...
6.344 control_reset_geometry index geometry_item_name geometry_item_index ...
6.345 control_reset_interface index geometry_item_name geometry_item_index ...
6.346 control_reset_interface_strain index geometry_item_name geometry_item_index ...
6.347 control_reset_node index node_0 node_1 ..................... 180
6.348 control_reset_value_constant index value .................. 181
6.349 control_reset_value_exponent index a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_10 ...
6.350 control_reset_value_linear index a_2 a_3 .......................... 181
6.351 control_reset_value_logarithmic_first index a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_10 ...
6.352 control_reset_value_logarithmic_second index a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_10 ...
6.353 control_reset_value_multi_linear index z_0 z_1 value ...
6.354 control_reset_value_power index a_2 a_3 a_4 a_9 a_10 ...
6.355 control_reset_value_method index method .............. 182
6.356 control_reset_value_square_root index a_2 a_3 a_4 a_9 a_10 a_11 ...
6.357 control_restart index switch ........................................ 182
6.358 control_safety_slip index switch .................................... 182
6.359 control_slide_damping_apply index switch ...................... 183
6.360 control_slide_plasti_apply index switch ......................... 183
6.361 control_slide_stiffness_apply index switch .................... 183
6.362 control_solver index solver_type .................................. 183
6.363 control_solver_bicg_error index error ............................... 184
6.364 control_solver_bicg_restart index nrestart ..................... 184
6.365 control_solver_bicg_stop index switch ............................ 184
6.366 control_solver_matrix_save index switch ....................... 184
6.367 control_solver_pardiso_out_of_core index switch ............. 185
6.368 control_solver_pardiso_ordering index ordering ................ 185
6.369 control_support_edge_normal_damping_apply index switch ..... 185
6.370 control_support_edge_normal_stiffness_freeze index switch . 185
6.371 control_system_call index integer_value .......................... 186
6.372 control_timestep index step_size time_increment step_size time_increment ... 186
6.373 control_timestep_adjust_minimum_iterations index switch ...... 186
6.374 control_timestep_iterations index number_of_iterations ...... 186
6.375 control_timestep_iterations_automatic index ratio_criterium minimal_timestep maximum_timestep ........................................ 186
6.376 control_timestep_iterations_automatic_minimum_maximum_wished index minimum_iterations maximum_iterations wished_iterations ........................................ 187
6.377 control_timestep_iterations_automatic_stop index switch ...... 187
6.378 control_timestep_multiplier index multiplier .................... 187
6.379 control_timestep_until_data index data_item_name_0 data_item_index_0 data_item_number_0 data_item_name_1 data_item_index_1 data_item_number_1 ... ........................................ 187
6.380 control_timestep_until_maximum index maximum_0 maximum_1 ... 188
6.381 control_timestep_until_minimum index minimum_0 minimum_1 ... 188
6.382 control_truss_rope_apply index switch ............................ 188
6.383 control_zip index switch ............................................. 188
6.384 convection_apply switch ............................................. 188
6.385 convection_stabilization switch ................................... 188
6.386 **data_activate**

*index* data_item_name_0 data_item_name_1 ... switch

6.387 **data_activate_time**

*index* time

6.388 **data_delete**

*index* data_item_name index_range

6.389 **data_delete_time**

*index* time

6.390 **data_ignore**

*data_item_name*

6.391 **dependency_apply**

*switch*

6.392 **dependency_diagram**

*index* dof_value_0 ...data_item_value_0 ...

6.393 **dependency_method**

*index* method

6.394 **dependency_geometry**

*index* geometry_item_name geometry_item_index

6.395 **dependency_item**

*index* data_item element_group dofn

6.396 **dependency_number**

*index* number

6.397 **dependency_type**

*index* type

6.398 **dof_element_dof**

dof_per_element_0 dof_per_element_1 ...

6.399 **dof_label**

dof_0 dof_1 ...

6.400 **dof_limit**

lower_dof_0 upper_dof_0 lower_dof_1 upper_dof_1 ...

6.401 **dtime**

*dt*

6.402 **element**

*index* element_name node_0 node_1 node_2 ...

6.403 **element_beam_direction**

*index* dir_x dir_y dir_z dir_x dir_y dir_z dir_x dir_y dir_z dir_x dir_y dir_z dir_x dir_y dir_z dir_x dir_y dir_z dir_x dir_y dir_z dir_x dir_y dir_z

6.404 **element_beam_direction_z**

*index* dir_z dir_y dir_z dir_z dir_y dir_z

6.405 **element_beam_force_moment**

*index* force_x_first_node force_y_first_node force_z_first_node moment_x_first_node moment_y_first_node moment_z_first_node force_x_second_node force_y_second_node force_z_second_node moment_x_second_node moment_y_second_node moment_z_second_node...

6.406 **element_boundary**

*index* switch

6.407 **element_contact_spring_direction**

*index* dirN_x dirN_y dirN_z dirT1_x dirT1_y dirT2_x dirT2_y dirT2_z

6.408 **element_contact_spring_strain**

*index* strain_N strain_T1 strain_T2

6.409 **element_contact_spring_force**

*index* force_N force_T1 force_T2

6.410 **element_dof**

*index* dof_0 dof_1 ...

6.411 **element_dof_initial**

*index* dof_0 dof_1 ...

6.412 **element_dof_initial_specific_number**

*index* number

6.413 **element_dof_initial_specific_value**

*index* value_0 value_grad_x value_grad_y value_grad_z
6.441 element_truss_direction index dir_x dir_y dir_z
6.442 element_truss_force index force
6.443 element_truss_strain index strain
6.444 element_truss_strain_temperature index strain
6.445 element_volume index volume
6.446 force_edge index force_0 force_1 ...
6.447 force_edge_diagram index vertical_displacement_0 factor_0 vertical_displacement_factor_1 ...
6.448 force_edge_element index element_0 element_1 ...
6.449 force_edge_element_group index element_group_0 element_group_1 ...
6.450 force_edge_element_node index element node_0 node_1 ...
6.451 force_edge_element_side index element_0 element_1 ... side
6.452 force_edge_factor index a_0 a_1 ... a_n
6.453 force_edge_geometry index geometry_entity_name geometry_entity_index ...
6.454 force_edge_node index node_0 node_1 ...
6.455 force_edge_node_factor index factor_0 factor_1 ...
6.456 force_edge_sine index start_time end_time freq_0 amp_0 freq_1 amp_1 ...
6.457 force_edge_time index time load time load ...
6.458 force_edge_normal index force
6.459 force_edge_normal_element index element_0 element_1 ...
6.460 force_edge_normal_element_node index element node_0 node_1 ...
6.461 force_edge_normal_element_group index element_group_0 element_group_1 ...
6.462 force_edge_normal_element_side index element_0 element_1 ... side
6.463 force_edge_normal_factor index a_0 a_1 ... a_{n-1}
6.464 force_edge_normal_geometry index geometry_entity_name geometry_entity_index ...
6.465 force_edge_normal_node index node_0 node_1 node_2 ...
6.466 force_edge_normal_node_factor index factor_0 factor_1 ...
6.467 force_edge_normal_sine index start_time end_time freq_0 amp_0 freq_1 amp_1 ...
6.468 force_edge_normal_time index time load time load ...
6.497 force_volume_geometry index geometry_item_name geometry_item_index 214
6.498 force_volume_sine index start_time freq_0 amp_0 freq_1 amp_1 ... 214
6.499 force_volume_time index time load time load ... 215
6.500 geometry_boundary index switch 215
6.501 geometry_bounda_sine_x index a b 215
6.502 geometry_bounda_sine_y index a b 215
6.503 geometry_bounda_sine_z index a b 215
6.504 geometry_bricks index x_c y_c z_c l_x l_y l_z tolerance 215
6.505 geometry_circle index x_c y_c ...radius tolerance 215
6.506 geometry_circle_part index x_c y_c angle_start angle_end radius tolerance 216
6.507 geometry_circle_segment index x_c y_c radius side_x side_y tolerance 216
6.508 geometry_cylinder index x_0 y_0 z_0 x_1 y_1 z_1 radius tolerance 216
6.509 geometry_cylinder_part index x_0 y_0 z_0 x_1 y_1 z_1 radius angle_start_0 angle_end_0 angle_start_1 angle_end_1 ...tolerance 216
6.510 geometry_brick start_point_0 ...start_point_z tolerance 217
6.511 geometry_cylinder_part_start_vector index v_x v_y v_z 217
6.512 geometry_cylinder_segment index x_0 y_0 z_0 x_1 y_1 z_1 radius side_x side_y side_z tolerance 217
6.513 geometry_element_geometry index element_geometry_0 element_geometry_1 217
6.514 geometry_element_geometry_method index method 218
6.515 geometry_element_group index element_group_0 element_group_1 ... 218
6.516 geometry_element_group_method index method 218
6.517 geometry_ellipse index x_c y_c a b tolerance 218
6.518 geometry_factor index factor_0 ... 218
6.519 geometry_hexahedral index x_0 y_0 z_0 x_1 y_1 z_1 x_2 y_2 z_2 x_3 y_3 z_3 219
6.520 geometry_line index x_0 y_0 z_0 x_1 y_1 z_1 radius 219
6.521 geometry_line_eps_iso index iso_tolerance 220
6.522 geometry_list index number_0 number_1 ... 220
6.523 geometry_method index method 220
6.524 geometry_moving index geometry_entity 220
6.525 geometry_moving_parameter index parameters of entity

6.526 geometry_moving_operat index operator

6.527 geometry_moving_operat_parameter index parameters of operator

6.528 geometry_moving_operat_time index start_time end_time

6.529 geometry_moving_n index ntime nspace

6.530 geometry_mpc index switch

6.531 geometry_node_type node_type

6.532 geometry_point index x y z radius

6.533 geometry_polynomial index a_0 a_1 ... a_n x_0 x_1 y_0 y_1 tolerance

6.534 geometry_projection_type index type

6.535 geometry_quadrilateral index x_0 y_0 z_0 x_1 y_1 z_1 x_2 y_2 z_2 x_3 y_3 z_3 tolerance

6.536 geometry_quadrilateral_eps_iso index iso_tolerance

6.537 geometry_set index geometry_entity_0 geometry_entity_index_0 geometry_entity_1 geometry_entity_index_1 ...

6.538 geometry_sphere index x_c y_c z_c radius tolerance

6.539 geometry_sphere_segment index x_c y_c z_c radius side_x side_y side_z tolerance

6.540 geometry_tetrahedral index x_0 y_0 z_0 x_1 y_1 z_1 x_2 y_2 z_2 x_3 y_3 z_3

6.541 geometry_triangle index x_0 y_0 z_0 x_1 y_1 z_1 x_2 y_2 z_2 x_3 y_3 z_3 tolerance

6.542 geometry_triangle_eps_iso index iso_tolerance

6.543 global_node_dof_empty switch

6.544 global_element_dof_apply switch

6.545 global_element_dof_from_node_dof switch

6.546 global_post_point_node_type node_type

6.547 groundflow_apply switch

6.548 groundflow_consolidation_apply switch

6.549 groundflow_density \( \rho \)

6.550 groundflow_flux_edge_normal index flux

6.551 groundflow_flux_edge_normal_element index element_0 element_1 ...

6.552 groundflow_flux_edge_normal_element_group index element_group_0 element_group_1 ...

221

221

221

222

222

222

222

222

222

222

223

223

223

224

224

224

224

224

224
6.608 group groundflow permeability index $p_{ex}$ $p_{ey}$ $p_{ez}$
6.609 group groundflow total pressure tension index plastic tension minimum
   water height
6.610 group integration method index method
6.611 group integration method reduced factor index factor
6.612 group integration points index type
6.613 group interface index switch
6.614 group interface confi conductivity index $k$
6.615 group interface gap index gap
6.616 group interface groundflow capacity index $C$
6.617 group interface groundflow permeability index $p_{e}$
6.618 group interface materi elasti stiffness index $k_n$ $k_t$,first $k_t$,second
6.619 group interface materi expansion normal index expansion coefficient normal
6.620 group interface materi memory index memory type
6.621 group interface materi plasti mohr coul direct index $\phi$ $c$ $\phi$flow
6.622 group interface materi plasti tension direct index switch
6.623 group interface materi residual stiffness index factor
6.624 group interface groundflow total pressure tension index strain normal minimum
   water height
6.625 group interface tangential reference point index point $x$ point $y$ point $z$
6.626 group materi damage mazars index $\epsilon_{0}$ $a$ $b$ $c$
6.627 group materi damping index $d$
6.628 group materi damping method index method
6.629 group materi density index density
6.630 group materi density groundflow index density wet density dry
6.631 group materi elasti borja tamagnini index $G_0$ $\alpha$ $k$ $p$
6.632 group materi elasti $c$ index 81 values
6.633 group materi elasti $c$ direction index dir_0 dir_1 dir_2
6.634 group materi elasti camclay $g$ index $G$
6.635 group materi elasti camclay poisson index $\nu$
6.636 group materi elasti camclay pressure min index pressure_min
6.637 group materi elasti compressibility index $co$
<table>
<thead>
<tr>
<th>Group Material</th>
<th>Elastic Properties</th>
<th>Indexes/Parameters</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardsoil</td>
<td>$E_{50}^{ref}$, $\sigma_{50}^{ref}$, $n$, $E_{ur}^{ref}$, $\sigma_{ur}^{ref}$, $\nu_{ur}$</td>
<td>242</td>
<td></td>
</tr>
<tr>
<td>$K_0$</td>
<td>$K_0$</td>
<td>242</td>
<td></td>
</tr>
<tr>
<td>Lade</td>
<td>$B$, $R$, $\lambda$</td>
<td>243</td>
<td></td>
</tr>
<tr>
<td>Poisson</td>
<td>$\nu_{1}$, $\nu_{2}$</td>
<td>243</td>
<td></td>
</tr>
<tr>
<td>Poisson Power</td>
<td>$\nu_{0}$, $\nu_{1}$, $p_{1}$, $\alpha$</td>
<td>243</td>
<td></td>
</tr>
<tr>
<td>Shear Factor</td>
<td>$\alpha$</td>
<td>243</td>
<td></td>
</tr>
<tr>
<td>Stress Pressure History Factor</td>
<td>$\alpha$</td>
<td>243</td>
<td></td>
</tr>
<tr>
<td>Transverse Isotropy</td>
<td>$E_1$, $E_2$, $\nu_1$, $\nu_2$, $G_2$, dir_x, dir_y</td>
<td>244</td>
<td></td>
</tr>
<tr>
<td>Volumetric Poisson</td>
<td>$\nu$</td>
<td>244</td>
<td></td>
</tr>
<tr>
<td>Volumetric Young Order</td>
<td>$n$</td>
<td>244</td>
<td></td>
</tr>
<tr>
<td>Volumetric Young Values</td>
<td>$\epsilon_0$, $\sigma_0$, $\epsilon_1$, $\sigma_1$</td>
<td>244</td>
<td></td>
</tr>
<tr>
<td>Young</td>
<td>$E$</td>
<td>244</td>
<td></td>
</tr>
<tr>
<td>Young Polynomial</td>
<td>$E_0$, $E_1$</td>
<td>244</td>
<td></td>
</tr>
<tr>
<td>Young Power</td>
<td>$E_0$, $E_1$, $p_{1}$, $\alpha$</td>
<td>244</td>
<td></td>
</tr>
<tr>
<td>Young User</td>
<td>$\alpha$</td>
<td>245</td>
<td></td>
</tr>
<tr>
<td>Volumetric Expansion Linear</td>
<td>$\alpha$</td>
<td>245</td>
<td></td>
</tr>
<tr>
<td>Volumetric Expansion Volume</td>
<td>$\beta$</td>
<td>245</td>
<td></td>
</tr>
<tr>
<td>Factor</td>
<td>$\alpha$</td>
<td>245</td>
<td></td>
</tr>
<tr>
<td>Failure Crunching</td>
<td>Threshold, delete_time</td>
<td>245</td>
<td></td>
</tr>
<tr>
<td>Failure Damage</td>
<td>Threshold, delete_time</td>
<td>245</td>
<td></td>
</tr>
<tr>
<td>Failure Plastix</td>
<td>Kappa, $\alpha$</td>
<td>246</td>
<td></td>
</tr>
<tr>
<td>Failure Rupture</td>
<td>Threshold, delete_time</td>
<td>246</td>
<td></td>
</tr>
<tr>
<td>Failure Void Fraction</td>
<td>Threshold, delete_time</td>
<td>246</td>
<td></td>
</tr>
<tr>
<td>History Variable User</td>
<td>Switch</td>
<td>246</td>
<td></td>
</tr>
<tr>
<td>History Variable User Parameters</td>
<td>Indexes...</td>
<td>246</td>
<td></td>
</tr>
<tr>
<td>Hyper Besselning</td>
<td>$K_1$, $K_2$</td>
<td>246</td>
<td></td>
</tr>
<tr>
<td>Hyper Blatz Ko</td>
<td>$G_{B}$</td>
<td>246</td>
<td></td>
</tr>
<tr>
<td>Hyper Mooney Rivlin</td>
<td>$K_1$, $K_2$</td>
<td>246</td>
<td></td>
</tr>
<tr>
<td>Hyper NeoHookean</td>
<td>$K_1$</td>
<td>247</td>
<td></td>
</tr>
<tr>
<td>Hyper Reduced Polynomial</td>
<td>$K_1$, $K_2$...</td>
<td>247</td>
<td></td>
</tr>
<tr>
<td>Hyper Volumetric Linear</td>
<td>Index $K$</td>
<td>247</td>
<td></td>
</tr>
</tbody>
</table>
6.690 group_materi_plasti_generalised_non_associate_cam_clay_for_bonded_soils
index ...

6.691 group_materi_plasti_gurson
index sigy q1 q2 q3

6.692 group_materi_plasti_hardsoil
index φ c ψ R

6.693 group_materi_plasti_heat_generation
factor

6.694 group_materi_plasti_hypo_cohesion
index c

6.695 group_materi_plasti_hypo_masin
index φ c λ∗ κ∗ N r

6.696 group_materi_plasti_hypo_masin_clay
index φ c λ∗ κ∗ N ν pp

6.697 group_materi_plasti_hypo_masin_clay_advanced_parameters
index α G α f a y α c

6.698 group_materi_plasti_diprisco_density
index γ f b p c p t p θ c ξ c ξ e β F γ d β q b p c p t p θ c ξ c ξ e

6.699 group_materi_plasti_druck_prag
index phi c phiflow

6.700 group_materi_plasti_element_group
index group_0 group_1 ...

6.701 group_materi_plasti_element_group_factor
index factor_0 factor_1 ...

6.702 group_materi_hyper_volumetric_murnaghan
index Kβ

6.703 group_materi_hyper_volumetric Ogden
index Kβ

6.704 group_materi_hyper_volumetric_polynomial
index K_0 K_1 ...

6.705 group_materi_hyper_volumetric_simo_taylor
index K

6.706 group_materi_maxwell_chain
index E_0 t_0 ... E_n-1 t_n-1

6.707 group_materi_membrane
index switch

6.708 group_materi_memory
index memory_type

6.709 group_materi_plasti_bounda
index index_0 index_1 ...

6.710 group_materi_plasti_bounda_factor
index factor

6.711 group_materi_plasti_coord_limit
index coord_limit

6.712 group_materi_plasti_camclay
index M κ λ

6.713 group_materi_plasti_cap1
index φ c M λ∗ κ∗ K⁎ e K⁎ f p e f m

6.714 group_materi_plasti_cap2
index c φ α R epsilon p v p b ...

6.715 group_materi_plasti_compression
index sigy

6.716 group_materi_plasti_compression_direct
index sigy

6.717 group_materi_plasti_compression_direct_visco
index tm

6.718 group_materi_plasti_diprisco
index γ f b p c p t p θ c ξ c ξ e β F γ d β q b p c p t p θ c ξ c ξ e

6.719 group_materi_plasti_diprisco_density
index γ f b p c p t p θ c ξ c ξ e

6.720 group_materi_plasti_diprisco_density
index γ f b p c p t p θ c ξ c ξ e

6.721 group_materi_plasti_element_group
index group_0 group_1 ...

6.722 group_materi_plasti_element_group_factor
index factor_0 factor_1 ...

6.723 group_materi_plasti_gurson
index sigy q1 q2 q3

6.724 group_materi_plasti_hardsoil
index φ c ψ R

6.725 group_materi_plasti_heat_generation
factor

6.726 group_materi_plasti_hypo_cohesion
index c

6.727 group_materi_plasti_hypo_masin
index φ c λ∗ κ∗ N r

6.728 group_materi_plasti_hypo_masin_clay
index φ c λ∗ κ∗ N ν pp

6.729 group_materi_plasti_hypo_masin_clay_advanced_parameters
index α G α f a y α c

6.730 group_materi_plasti_hypo_masin
index φ c λ∗ κ∗ N r

6.731 group_materi_plasti_hypo_masin_clay
index φ c λ∗ κ∗ N ν pp
6.727 group_materi_plasti_tension_direct index sigy .......................... 259
6.728 group_materi_plasti_tension_direct_automatic index switch ......... 259
6.729 group_materi_plasti_tension_direct_visco index tm .................. 259
6.730 group_materi_plasti_tension_direct_wall index sigy .................. 259
6.731 group_materi_plasti_user index switch ............................... 260
6.732 group_materi_plasti_visco_exponential index γ α .................... 260
6.733 group_materi_plasti_visco_exponential_limit index limit ............. 260
6.734 group_materi_plasti_visco_exponential_name index name_0 name_1 ... 260
6.735 group_materi_plasti_visco_exponential_values index γ₀ α₀ γ₁ α₁ ... 260
6.736 group_materi_plasti_visco_power index η p ............................ 260
6.737 group_materi_plasti_visco_power_name index name_0 name_1 ... 261
6.738 group_materi_plasti_visco_power_value index η₀ p₀ η₁ p₁ ... .... 261
6.739 group_materi_plasti_vonmises index sigma_y0 .......................... 261
6.740 group_materi_plasti_vonmises_nadai index C₀ n ..................... 261
6.741 group_materi_stokes index switch ..................................... 261
6.742 group_materi_umat index switch ....................................... 261
6.743 group_materi_umat_parameters index parameter_0 parameter_1 ... 262
6.744 group_materi_umat_pardiso_decompose index switch ................. 262
6.745 group_materi_undrained_capacity index C .............................. 262
6.746 group_materi_viscosity index ν ................................. 262
6.747 group_materi_viscosity_heatgeneration switch ........................ 262
6.748 group_materi_viscosity_user index switch ............................ 262
6.749 group_plasti_apply index switch ..................................... 262
6.750 group_porosity index n ............................................ 262
6.751 group_spherical index switch .................................... 263
6.752 group_spring_direction index dir_x dir_y dir_z .................... 263
6.753 group_spring_memory index memory_type ............................ 263
6.754 group_spring_plasti index F_y ...................................... 263
6.755 group_spring_stiffness index k ...................................... 263
6.756 group_spring_stiffness_nonlinear index epsilon₀ k₀ epsilon₁ k₁ ... 263
6.757 group_time index birth death ........................................ 263
6.758 group_time_fill index birth_empty birth_filled death
6.759 group_truss_area index A
6.760 group_truss_density index \( \rho \)
6.761 group_truss_elast_elongation_force_diagram index \( l_0 F_0 l_1 F_1 \)
6.762 group_truss_elast_young index \( E \)
6.763 group_truss_expansion index \( \alpha \)
6.764 group_truss_initial_force index initial_force
6.765 group_truss_memory index memory_type
6.766 group_truss_rope index switch
6.767 group_truss_plasti index \( \sigma_c \sigma_t \)
6.768 group_type index type_name_0 type_name_1 ...
6.769 group_volume_factor index factor
6.770 group_wave_speed_of_sound index \( c \)
6.771 icontrol icontrol
6.772 incremental_driver ...
6.773 inertia_apply switch_0 switch_1 ...
6.774 input_abaqus switch
6.775 input_abaqus_continue switch
6.776 input_abaqus_group switch
6.777 input_abaqus_mesh switch
6.778 input_abaqus_set set_0 set_1 ...
6.779 input_abaqus_name name_0 name_1 ...
6.780 input_feflow_mesh switch
6.781 input_feflow_fem switch
6.782 input_feflow_mesh_hydraulic_head switch
6.783 input_gmsh switch
6.784 interface_gap_apply switch
6.785 license_check switch
6.786 license_wait switch
6.787 linear_calculation_apply switch
6.788 materi_damage_apply switch
6.789 `materi_dynamic factor` ................................. 274
6.790 `materi_elasti_young_power_apply switch` ............................. 274
6.791 `materi_failure_apply switch` ...................................... 274
6.792 `materi_plasti_hypo_substepping index switch` ..................... 275
6.793 `materi_plasti_tension_apply switch` .................................. 275
6.794 `materi_plasti_visco_apply switch` ........................................ 275
6.795 `mesh specifier_x specifier_y specifier_z` .............................. 275
6.796 `mesh_activate_gravity_element index element_range` ............. 275
6.797 `mesh_activate_gravity_element_group index element_group_0 element_group_1` .............................. 275
6.798 `mesh_activate_gravity_geometry index geometry_item_name geometry_item_index` ............................. 276
6.799 `mesh_activate_gravity_method index method` .......................... 276
6.800 `mesh_activate_gravity_stiffness_factor index factor` ................ 276
6.801 `mesh_activate_gravity_time index time_start time_end` ............. 276
6.802 `mesh_activate_gravity_time_initial index time_of_birth` ........... 277
6.803 `mesh_activate_gravity_time_strain_settlement index switch` .......... 277
6.804 `mesh_boundary switch` .................................................. 277
6.805 `mesh_correct switch` .................................................... 277
6.806 `mesh_delete_geometry_moving index geometry_moving_index` ............. 277
6.807 `mesh_element_group_apply index group_0 group_1 ...` .............. 278
6.808 `mesh_gid_assign_conditions_line index line_0 line_1 ...` ............ 278
6.809 `mesh_gid_assign_conditions_point index point_0 point_1 ...` .......... 278
6.810 `mesh_gid_assign_conditions_surface index surface_0 surface_1 ...` ............................. 278
6.811 `mesh_gid_arc_coord index x_0 y_0 x_1 y_1 x_2 y_2` .......................... 278
6.812 `mesh_gid_circle_coord index x y` ........................................... 278
6.813 `mesh_gid_circle_element_group index group` ............................. 279
6.814 `mesh_gid_circle_hollow index switch` ...................................... 279
6.815 `mesh_gid_circle_radius index radius` ..................................... 279
6.816 `mesh_gid_cylinder_coord index x y z` ...................................... 279
6.817 `mesh_gid_cylinder_element_group index group` .......................... 279
6.818 `mesh_gid_cylinder_height index height` ..................................... 279
6.819 `mesh_gid_cylinder_hollow index switch` ...................................... 279
6.820 mesh_gid_cylinder_normal index normal_x normal_y normal_z ........ 279
6.821 mesh_gid_cylinder_radius index radius .......................... 279
6.822 mesh_gid_line_structured_concentrate index weight_start weight_end ........ 279
6.823 mesh_gid_line_element_group index group ......................... 280
6.824 mesh_gid_line_point index point_0 point_1 ........................ 280
6.825 mesh_gid_line_size index size .................................... 280
6.826 mesh_gid_line_structured_nel index nel ............................ 280
6.827 mesh_gid_line_structured_size index size .......................... 280
6.828 mesh_gid_point_coord index x y z ................................ 280
6.829 mesh_gid_rectangle_coord index x_0 x_1 y_0 y_1 .................... 280
6.830 mesh_gid_rectangle_element_group index group .................... 280
6.831 mesh_gid_rectangle_hollow index switch ........................... 280
6.832 mesh_gid_size element_size .................................. 281
6.833 mesh_gid_sphere_coord index x y z ................................ 281
6.834 mesh_gid_sphere_element_group index group ....................... 281
6.835 mesh_gid_sphere_hollow index switch ............................... 281
6.836 mesh_gid_sphere_radius index radius ................................ 281
6.837 mesh_gid_surface_element index element_type ....................... 281
6.838 mesh_gid_surface_element_group index group ...................... 281
6.839 mesh_gid_surface_line index line_0 line_1 ... ....................... 281
6.840 mesh_gid_surface_structured_nel index nel_0 nel_1 ... ............. 281
6.841 mesh_gid_surface_structured_size index size ....................... 282
6.842 mesh_gid_volume_element_group index group ....................... 282
6.843 mesh_gid_volume_surface index surface_0 surface_1 ... ............ 282
6.844 mesh_interface_triangle_coordinates index coord_x_0 coord_y_0 coord_z_0 coord_x_1 coord_y_1 coord_z_1 coord_x_2 coord_y_2 coord_z_2 ... ........ 282
6.845 mesh_interface_triangle_element_group index element_group ........ 282
6.846 message switch .................................................... 283
6.847 mpc_apply switch ................................................. 283
6.848 mpc_element_group index element_group_0 element_group_1 ......... 283
6.849 mpc_element_group_always index switch ................................ 283
6.850 mpc_element_group_closest index switch ............................ 283
<table>
<thead>
<tr>
<th><strong>6.880</strong> node_rhside</th>
<th>index rhside_0 rhside_1 ...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>6.881</strong> node_slide</td>
<td>index slide_number</td>
</tr>
<tr>
<td><strong>6.882</strong> node_static_pressure</td>
<td>index value</td>
</tr>
<tr>
<td><strong>6.883</strong> node_start_refined</td>
<td>index coord_0 coord_1 coord_2</td>
</tr>
<tr>
<td><strong>6.884</strong> node_stiffness</td>
<td>index stiffness_x stiffness_y stiffness_z</td>
</tr>
<tr>
<td><strong>6.885</strong> node_support_edge_normal_plasti_tension_status</td>
<td>index status</td>
</tr>
<tr>
<td><strong>6.886</strong> node_total_pressure</td>
<td>index value</td>
</tr>
<tr>
<td><strong>6.887</strong> nonlocal</td>
<td>nonlocal_radius</td>
</tr>
<tr>
<td><strong>6.888</strong> nonlocal_name</td>
<td>name</td>
</tr>
<tr>
<td><strong>6.889</strong> plasti_apply</td>
<td>switch</td>
</tr>
<tr>
<td><strong>6.890</strong> post_apply</td>
<td>index switch</td>
</tr>
<tr>
<td><strong>6.891</strong> post_calcul</td>
<td>dof_0 operat_0 dof_1 operate_1 ...</td>
</tr>
<tr>
<td><strong>6.892</strong> post_calcul_absolute</td>
<td>switch</td>
</tr>
<tr>
<td><strong>6.893</strong> post_calcul_apparent_total</td>
<td>switch</td>
</tr>
<tr>
<td><strong>6.894</strong> post_calcul_label</td>
<td>doflabel_0 label_1 ...</td>
</tr>
<tr>
<td><strong>6.895</strong> post_calcul_limit</td>
<td>lower_0 upper_0 lower_1 upper_1 ...</td>
</tr>
<tr>
<td><strong>6.896</strong> post_calcul_materi_stress_force_average</td>
<td>switch</td>
</tr>
<tr>
<td><strong>6.897</strong> post_calcul_materi_stress_force_direction_exclude</td>
<td>dir_x dir_y dir_z</td>
</tr>
<tr>
<td><strong>6.898</strong> post_calcul_materi_stress_force_direction_exclude_epsilon</td>
<td>eps</td>
</tr>
<tr>
<td><strong>6.899</strong> post_calcul_materi_stress_force_direction_include</td>
<td>dir_x dir_y dir_z</td>
</tr>
<tr>
<td><strong>6.900</strong> post_calcul_materi_stress_force_direction_include_epsilon</td>
<td>eps</td>
</tr>
<tr>
<td><strong>6.901</strong> post_calcul_materi_stress_force_element_group</td>
<td>element_group_0 element_group_1 ...</td>
</tr>
<tr>
<td><strong>6.902</strong> post_calcul_materi_stress_force_reference_point</td>
<td>x_0 y_0 z_0 x_1 y_1 ...</td>
</tr>
<tr>
<td><strong>6.903</strong> post_calcul_materi_stress_force_outer</td>
<td>switch</td>
</tr>
<tr>
<td><strong>6.904</strong> post_calcul_materi_stress_force_plot_switch</td>
<td>switch_0 switch_1 ...</td>
</tr>
<tr>
<td><strong>6.905</strong> post_calcul_materi_stress_force_thickness_switch</td>
<td>switch_element_group_0 switch_element_group_1 ...</td>
</tr>
<tr>
<td><strong>6.906</strong> post_calcul_multiply</td>
<td>factor_0 factor_1 ...</td>
</tr>
<tr>
<td><strong>6.907</strong> post_calcul_safety_default</td>
<td>eps value</td>
</tr>
<tr>
<td><strong>6.908</strong> post_calcul_safety_maximum</td>
<td>value</td>
</tr>
</tbody>
</table>
6.937 post_node_rhside_fixed value_0 value_1 . . . 306
6.938 post_node_rhside_free value_0 value_1 . . . 306
6.939 post_node_rhside_ratio ratio . . . . . . . . . . . 306
6.940 post_node_rhside_ratio_dof_type dof_type_0 . . 307
6.941 post_node_rhside_ratio_method method . . . . 307
6.942 post_point index x y z . . . . . . . . . . . . . . . 307
6.943 post_point_element_group index element_group . 307
6.944 post_point_dof index dof_0 dof_1 . . . . . . . 307
6.945 post_point_dof_calcul . . . . . . . . . . . . . . 307
6.946 post_point_eps_iso index eps . . . . . . . . . . . 307
6.947 post_quadrilateral index x_0 y_0 z_0 x_1 y_1 z_1 x_2 y_2 z_2 x_3 y_3 z_3 . 308
6.948 post_quadrilateral_dof index dof_0 dof_1 . . . 308
6.949 post_quadrilateral_dof_calcul . . . . . . . . . . 308
6.950 post_quadrilateral_element_group index element_group . 308
6.951 post_quadrilateral_n index n . . . . . . . . . . . 308
6.952 post_strain_volume_absolute index volume_increase_absolute . 308
6.953 post_strain_volume_initial index volume_initial . 308
6.954 post_strain_volume_relative index volume_strain_relative . 308
6.955 print_apply switch . . . . . . . . . . . . . . . . . 309
6.956 print_arithmetic switch . . . . . . . . . . . . . . 309
6.957 print_control switch . . . . . . . . . . . . . . . . 309
6.958 print_data_name switch . . . . . . . . . . . . . . 309
6.959 print_database_calculation switch . . . . . . . . 309
6.960 print_define switch . . . . . . . . . . . . . . . . . 309
6.961 print_element_geometry_present switch . . . . 309
6.962 print_element_geometry_present_node_type node_type . 310
6.963 print_failure switch . . . . . . . . . . . . . . . . . 310
6.964 print_filter index data_item_name data_item_index number_0 number_1 . . . . . . . . . . . . . . . . . 310
6.965 print_gid_calculation switch . . . . . . . . . . . 311
6.966 print_frd_freecad switch . . . . . . . . . . . . . . 311
6.967 print_frd_prepomax switch . . . . . . . . . . . . . 311
6.968 print_gid_contact_spring2 number_of_nodes .......................... 311
6.969 print_gid_coord switch .................................................. 311
6.970 print_gid_define switch .................................................. 312
6.971 print_gid_group switch .................................................. 312
6.972 print_gid_mesh_activate_gravity switch ............................... 312
6.973 print_gid_node_method method ........................................... 312
6.974 print_gid_spring2 number_of_nodes ................................... 312
6.975 print_group_data dataitem_name_0 dataitem_name_1 ... ............ 312
6.976 print_gmsh_calculation switch .......................................... 312
6.977 print_gmsh_dummy switch ............................................... 313
6.978 print_gmsh_node_method method ....................................... 313
6.979 print_mesh_dof dof_0 dof_1 ... ........................................ 313
6.980 print_node_geometry_present switch .................................... 314
6.981 print_node_geometry_present_node_type node_type ................. 314
6.982 print_precision number_of_values .................................... 314
6.983 print_tecplot_calculation switch ....................................... 314
6.984 print_vtk_calculation switch ........................................... 314
6.985 print_vtk_coord switch .................................................. 315
6.986 print_vtk_group switch .................................................. 315
6.987 print_vtk_node_method method ......................................... 315
6.988 print_where switch ....................................................... 315
6.989 processors nproc .......................................................... 315
6.990 processors_maximum switch .............................................. 315
6.991 processors_partition npartition ........................................ 315
6.992 relaxation relax_0 relax_1 ... ........................................ 316
6.993 repeat_save_result index result_0 result_1 ... ........................ 316
6.994 repeat_save_calculate_result average_0 variance_0 average_1 variance_1 ... 316
6.995 safety_slip_circle_grid_middle index x_first y_first x_last y_last ... 316
6.996 safety_slip_circle_grid_middle_n index n ............................... 316
6.997 safety_slip_circle_grid_radius index r_first r_last .................. 316
6.998 safety_slip_circle_grid_radius_n index n ............................... 317
6.102 slide_plasti_friction index phi_c ............................................. 322
6.102 slide_plasti_tension index sig_t ............................................. 322
6.102 slide_user index switch ............................................................ 323
6.103 slide_damping index damping_n damping_t ................................. 323
6.103 slide_stiffness index stiffness_n stiffness_t ................................ 323
6.103 solver solver_type ................................................................. 323
6.103 solver_bicg_error error ............................................................ 323
6.103 solver_bicg_restart nrestart .................................................... 323
6.103 solver_bicg_stop switch ............................................................ 323
6.103 solver_matrix_save switch ....................................................... 323
6.103 solver_matrix_symmetric switch .............................................. 324
6.103 solver_pardiso_ordering ordering ............................................. 324
6.103 solver_pardiso_out_of_core switch .......................................... 324
6.103 solver_pardiso_processors nproc .............................................. 324
6.103 solver_pardiso_processors_maximum switch ............................... 324
6.104 strain_settlement_parameters index time_global,start time_plus reference_creep_strain_rate reference_time power_n lateral_factor .............................................................. 324
6.104 strain_settlement_element_group index element_group_0 element_group_1 ................................................................. 325
6.104 strain_volume_absolute_time index time_0 volume_increase_absolute_0 time_1 volume_increase_absolute_1 ................................................................. 325
6.104 strain_volume_element index element_0 element_1 ...................... 325
6.104 strain_volume_element_group index element_group_0 element_group_1 ................................................................. 326
6.104 strain_volume_geometry index geometry_item_name geometry_item_index ................................................................. 326
6.104 strain_volume_relative_time index time_0 relative_volume_strain_0 time_1 relative_volume_strain_1 ................................................................. 326
6.105 support_edge_normal index stiffness_normal stiffness_tangential ........ 326
6.105 support_edge_normal_damping index damping_normal damping_tangential ................................................................. 327
6.105 support_edge_normal_damping_automatic index switch ................. 327
6.105 support_edge_normal_damping_automatic_apparent index switch .... 327
6.105 support_edge_normal_density index density_normal density_tangential ........ 327
6.105 support_edge_normal_element_node index element_0 element_1 ........ 328
6.105 support_edge_normal_element_group index element_group ......... 328
6.105 $\text{support_edge_normal_element_side} \ index \ element_0 \ element_1 \ldots \text{side}$  328
6.105 $\text{support_edge_normal_factor} \ index \ a_0 \ a_1 \ldots a_n$  328
6.105 $\text{support_edge_normal_force_initial} \ index \ a_0 \ a_1$  328
6.105 $\text{support_edge_normal_geometry} \ index \ geometry_entity_name \ geometry_entity_index$  328
6.106 $\text{support_edge_normal_node} \ index \ node_0 \ node_1 \ node_2 \ldots$  328
6.106 $\text{support_edge_normal_plasti_compression} \ index \ normal_force_minimum \ tangential_force_factor$  328
6.106 $\text{support_edge_normal_plasti_friction} \ index \ cohesion \ friction_coefficient$  329
6.106 $\text{support_edge_normal_plasti_tension} \ index \ switch$  329
6.106 $\text{support_edge_normal_plasti_tension_double} \ index \ normal_force_maximum$  329
6.106 $\text{support_edge_normal_plasti_residual_stiffness} \ index \ factor$  329
6.106 $\text{support_edge_normal_time} \ index \ time \ load \ time \ load \ldots$  329
6.106 $\text{target_item} \ index \ data_item_name \ data_item_index \ number$  330
6.106 $\text{target_value} \ index \ value \ tolerance$  330
6.106 $\text{time_calculation} \ elapsed_time_in_seconds$  330
6.107 $\text{time_current} \ current_time$  330
6.107 $\text{timestep_predict_velocity} \ switch$  330
6.107 $\text{timestep_iterations_automatic_apply} \ switch$  331
6.107 $\text{tochnog_version} \ index \ day \ month \ year$  331
6.107 $\text{truss_rope_apply} \ switch$  331
6.107 $\text{volume_factor} \ a_0 \ a_1 \ldots a_n$  331
6.107 $\text{volume_factor_x} \ x_0 \ fac_01 \ x_1 \ fac_12 \ldots x_n$  331
6.107 $\text{Zip} \ switch$  331
6.107 $\text{end_data} \ (\text{last record of data part})$  331

7 Runtime file  332

8 Interaction analyzes and advanced analyzes  333
8.1 Fluid-structure interaction  333
8.2 Consolidation analysis: ground water flow in deforming solid  333
8.3 Heat transport in ground water flow  334
8.4 Heat transport in materials  334
8.5 Restart a calculation  335

44
9 Final topics (input trouble, save memory /CPU time, ...)

9.1 Environment symbols

9.2 Checking your geometry_* records

9.3 Continuing an analysis

9.4 Use -node as geometry entity.

9.5 Use -geometry_list as geometry entity.

9.6 List input files with options

9.7 Geometrically linear material

9.8 Dynamic calculations

9.9 Input file syntax

9.10 Check large calculations

9.11 Diverging calculations

9.12 Saving CPU time

9.13 Saving computer memory

9.14 Inaccurate results

9.15 Element sides

9.16 Badly shaped elements

9.17 For selected customers only

9.18 Youtube

9.19 External programs

9.20 Forces are setup in the element loop in timesteps

9.21 Running in a Microsoft Windows bash shell

10 User supplied subroutines
1  Basic information

1.1  How to perform a calculation and how to get started

Create an input file, e.g. problem.dat. The default input file is tochnog.dat, which will be used if no other input file is specified. Thus the command tochnog tochnog.dat yields output on the screen while tochnog tochnog.dat > tochnog.out redirects the output to a file.

So to get started do, for example, the following:

- cd test/other
- tochnog condif1.dat

Use the condif1.dat test to get started.

- Copy condif1.dat to tochnog.dat.
- Use your favorite editor to open the file tochnog.dat and study it.
- Change echo to -yes.
- Remove the parentheses (...) surrounding the control_print statement and save the file.
- Run by typing tochnog or tochnog tochnog or tochnog tochnog.dat.
- Study the output on the screen.
- Study the tochnog.log file.
- Study the tochnog.dbs file. It contains the database after the calculation, and is an input file itself!

Read at least once the start of the data part introduction section.

1.2  Pre- and postprocessing

You can use GID both for preprocessing (mesh generation) and post processing (plotting). GID is commercially available at the www.gidhome.com Internet page. A free demo version of is available for download.

Alternatively to GID you can use Mecway for preprocessing input_abaqus and post processing control_print_gmsh. Mecway is commercially available at the mecway.com Internet page. It is very affordable, and also has build in FE calculations (mostly for mechanical engineering). A free demo version of Mecway is available for download.

You can also use GMSH both for preprocessing and post processing. GMSH is freely available at www.geuz.org/gmsh.

Postprocessing files are written for the visualization program PARAVIEW. The PARAVIEW program is freely available at www.paraview.org.

Furthermore, postprocessing files are written for the visualization program TECPLLOT. These TECPLLOT are less well maintained then the files for other postprocessing programs.
With **GNUPlot** you can plot files resulting from `control_print_history` and `control_print_data_versus_data`. Also any other x-y plotting program can be used for such files.

1.3 **Space discretization, time discretization**

The computational domain is divided into finite elements. The elements connect at nodes. Either one-dimensional (1D), two-dimensional (2D), three-dimensional (3D) or axi-symmetrical (2D) domains can be used.

Only first order in time equations are solved. Time derivatives are approximated with Euler backward time discretization.

Tochnog professional can store strains, stresses etc. either in element integration points (jumps between elements possible) or in nodes (continuous fields between elements); see `global_element_dof_apply`.

1.4 **Program capabilities**

- **Input**
  Format free input. Words and no 'magic numbers' in rigidly defined columns are used.
  Boundary conditions can be imposed onto at geometrical entities, as well as onto elements and nodes.

- **Output/plotting**
  Output can be printed over user-specified geometrical objects (points, lines, quadrilaterals,...) as well as at nodes.
  The history of each variable, and for functions of variables, can be printed over user-specified geometrical objects as well as at nodes.
  Interface files for the GID pre- and post processor.

- **Finite elements**
  1D, 2D and 3D. Tochnog mostly uses isoparametric elements. There are also springs, trusses, beams and contact-springs however.
  Linear and quadratic simplex elements (triangles, tetrahedrons). Linear and quadratic prism elements. A full family of first to fourth order bar, quadrilateral and brick elements.

- **Mesh generation/refining/etc.**
  Macro regions are automatically divided into finite elements.
  Local h-refinement
  Global h-refinement (more elements).
  Global p-refinement (polynomial refinement).

- **Differential equations (materials)**
  Convection-diffusion equation:
  - Temperature calculations.
  Fluids:
  - Stokes and Navier-Stokes.
  Solids:
  - Elasticity (isotropy and transverse isotropy).
  - Elasto-Plasticity (Von-Mises, Mohr-Coulomb, Gurson, etc.; plasticity surfaces can be arbitrarily combined).
- Hypo-Plasticity (Von-Woffersdorff, Masin, cohesion, intergranular strains, pressure dependent initial void ratio).
- Damage.
- Thermal stresses.
- Hypoelasticity.
- Viscoelasticity.
- Viscoplasticity.
- Viscosity.

Ground water flow equation:
- Storage equation. - saturated and non-saturated - multiple phreatic levels - piping and lifting safety - fully coupled consolidation analysis

Wave equation.

• Interaction analysis
  Automatic fluid-solid interaction.
  Temperature effects on fluids, solids.

• Contact analysis
  Contact with and without friction.
  Frictional heat generation.

• Frames of description
  Lagrangian and Eulerian

• Types of analysis
  Static, quasi-static and dynamic analysis.

• Parallelization
  Full shared memory parallelization node and element loops

• Special features
  Automatic time-stepping (large steps for good iteration behavior, small steps for bad iteration behavior).
  Automatic distribution of tendon trusses over finite elements (automatic embedment).
  Restart possibility.
  Convection wiggle stabilization (both for low and high order elements).

### 1.5 Files used by Tochnog

- Input file. For example **condif1.dat**. The input file consists of an initialization part (which dof’s should be solved, etc.) and a data part (elements, nodes, etc.).

- Runtime input file. For example **condif1.run**. Use it to give Tochnog data records on the fly (while it is running).

- Plot files. For example **condif1_flavia.msh** and **condif1_flavia.res**.

- Database file. For example, after the calculation with input file **condif1.dat** the database file **condif1.dbs** will be written. It contains everything (nodes, elements, solutions fields, etc.). On error exit for example **condif1_error.dbs** will be generated.

- Scratch file **tochnog_tmp.txt**. Don’t use this name yourself.

- Log file **tochnog.log**. Contains log messages of calculations.
2 Equations

2.1 Convection and diffusion of heat

2.1.1 Convection-diffusion equation

\[ \rho C (\dot{T} + \beta_i \frac{\partial T}{\partial x_i}) = k_i \frac{\partial^2 T}{\partial x_i^2} - aT + f \]

The primary dof is the condif_temperature \( T \). Further notation: \( \rho \) group_condif_density; \( C \) group_condif_capacity; \( x \) space coordinate; \( \beta_i \) group_condif_flow in \( i \)-direction; \( k_i \) group_condif_conductivity in \( i \)-direction; \( a \) group_condif_absorption; \( f \) condif_heat_volume. Typical applications are heat conduction and heat conduction in a flow.

2.1.2 Convection to environment

\[ q_c = \alpha_c (T - T_c) \]

Here \( q_c \) is the condif_convection_edge_normal heat flux, \( \alpha_c \) is the convection coefficient and \( T_c \) is the environmental temperature for convection.

2.1.3 Radiation to environment

\[ q_r = \alpha_r (T^4 - T_r^4) \]

Here \( q_r \) is the condif_radiation_edge_normal heat flux, \( \alpha_r \) is the radiation coefficient and \( T_r \) is the environmental temperature for radiation.
2.2 Material deformation and flow

\[ \rho \dot{v}_i = \frac{\partial \sigma_{ij}}{\partial x_j} + (1 - \beta T) \rho g_i - dv_i + f_i \]

Notations: \( \rho \) group_material_density; \( v_i \) materi_velocity in \( i \)-direction; \( \sigma_{ij} \) materi_stress matrix; \( x \) space coordinate; \( \beta \) group_material_expansion_volume; \( T \) (optional) condif_temperature; \( g_i \) force_gravity; \( d \) is the group_material_damping coefficient (see also the dynamics section near the end of this manual); \( f_i \) force_volume. The equation is given for space coordinates following the material velocities \( v_i \).

TOCHNOG allows you to build your favorite material, by adding separate contributions to the stresses \( \sigma_{ij} \). In this way you can build solids or fluids or things in between. The separate contributions will be listed below. First two typical examples are given.

Nearly incompressible Navier Stokes:

```
... materi_velocity materi_stress end_initia ...
... mesh -fixed_in_space -fixed_in_space
timestep_predict_velocity 0 -yes ...
... group_type 0 -materi group_material_elasti_compressibility 0 1.0 group_material_viscosity 0 1.2 ...
```

Linear solid:

```
... materi_velocity materi_velocity_integrated materi_stress end_initia ...
... group_type 0 -materi group_material_elasti_young 0 1.e10 group_material_elasti_poisson 0 0.2 group_material_memory 0 -updated_linear ...
```

2.2.1 Memory

The -updated Lagrange formulation

Deformations (i.e. the incremental deformation matrix \( F \)) refers to the previous time point. TOCHNOG decomposes the incremental deformation tensor with a polar decomposition into \( F = RU \) with \( F \) the incremental deformation matrix, \( R \) the incremental rotation matrix and
\( U \) the incremental stretch matrix. The incremental stretch matrix \( U \) is used to determine the incremental strain matrix \( 0.5(U + U^T) - I \) with \( I \) the identity tensor. The stresses at a new timepoint are calculated as:

- calculate extra stresses due to incremental strain matrix
- add these extra stresses to the stresses of the previous time point
- rotate these new stresses by the matrix \( R \)

The \texttt{-updated\_jaumann} Lagrange formulation

Deformations (i.e. the incremental deformation matrix \( F \)) refers to the previous time point. The incremental stretch matrix \( U \) is used to determine the incremental strain matrix \( 0.5(F + F^T) - I \) with \( I \) the identity tensor. The incremental rotation matrix \( R \) is \( 0.5(F - F^T) + I \). The stresses at a new timepoint are calculated as:

- calculate extra stresses due to incremental strain matrix
- add these extra stresses to the stresses of the previous time point
- rotate these new stresses by the matrix \( R \)

The \texttt{-updated\_linear} Lagrange formulation

Deformations (i.e. the incremental deformation matrix \( F \)) refers to the previous time point. Any rigid body rotation between the two time points are neglected, so TOCHNOG decomposes the incremental deformation tensor with a polar decomposition into \( F = U \) with \( F \) the incremental deformation matrix, and \( U \) the incremental stretch matrix. The linear engineering strains in the deformed configuration are used as incremental strain matrix \( 0.5(F + F^T) - I \). The stresses at a new timepoint are calculated as:

- calculate extra stresses due to incremental strain matrix
- add these extra stresses to the stresses of the previous time point

The \texttt{-total} Lagrange formulation

Deformations (i.e. the total deformation matrix \( F \)) refers to the time 0. TOCHNOG decomposes the total deformation tensor with a polar decomposition into \( F = RU \) with \( F \) the total deformation matrix, \( R \) the total rotation matrix and \( U \) the total stretch matrix. The total stretch matrix \( U \) is used to determine the total strain matrix \( 0.5(U + U^T) - I \) with \( I \) the identity tensor. The stresses at a new timepoint are calculated as:

- back-rotate the old stresses at the previous time point to time 0 with the old rotation matrix
- calculate extra stresses due to incremental strain matrix
- add these extra stresses to the back-rotated old stresses of the previous time point
- rotate the added stresses with the new rotation matrix \( R \) to the new configuration

The \texttt{-total\_linear} Lagrange formulation

Deformations (i.e. the total deformation matrix \( F \)) refers to the time 0. TOCHNOG neglects any rigid body rotations and uses linear engineering strains \( 0.5(F + F^T) - I \). The difference in these linear engineering strains between two time points are the incremental strains.

The stresses at a new timepoint are calculated as:
• calculate extra stresses due to incremental strain matrix
• add these extra stresses to the stresses of the previous time point

See also group_materi_memory.

2.2.2 Elasticity

The elastic stress rate is

\[ C_{ijkl} \dot{\epsilon}^{el}_{kl} \]

where \( C_{ijkl} \) is the elastic modulus tensor (which is a doubly symmetric tensor: \( C_{ijkl} = C_{jikl} \), \( C_{ijkl} = C_{ijlk} \) and \( C_{ijkl} = C_{jilk} \)), and \( \dot{\epsilon}^{el}_{kl} \) is the elastic strain rate. See the plasticity section for a definition of the elastic strain rate.

For an isotropic material

\[ C_{0000} = C_{1111} = C_{2222} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \]
\[ C_{0011} = C_{0022} = C_{1122} = \frac{E\nu}{(1+\nu)(1-2\nu)} \]
\[ C_{0101} = C_{0202} = C_{1212} = \frac{E}{1+\nu} \]

with \( E \) group_materi_elasti_young modulus and \( \nu \) group_materi_elasti_poisson ratio (the remaining non-zero moduli follow from the double symmetry conditions).

For a transverse isotropic material the material has one unique direction (think of an material with fibers in one direction). Here we take '2' as the unique direction; '1' and '3' are the transverse directions. The material is fully defined by \( E_1, E_2, \nu_1, \nu_2 \), and \( G_2 \). This set of parameters leads directly to a set of elasticity coefficients \( C_{ijkl} \). The parameters can be given in group_materi_elasti_transverse_isotropy.

The nonlinear elasticity polynomials is a strain dependent model. In this model, the ‘young’s stiffness’ modulus is made dependent of the size of the strains via a series of polynomials

\[ E = E_0 + E_1 \epsilon^1 + E_2 \epsilon^2 + \ldots \]

(1)

where

\[ \epsilon = \sqrt{\epsilon_{ij} \epsilon_{ij}} \]

(2)

with \( \epsilon_{ij} \) the components of the strain matrix. The parameters \( E_0 \) etc. need to be specified in the group_materi_elasti_young_polynomial record.

If the pressure state in a point is positive, we apply \( E = E_0 \).

The power law nonlinear elasticity is a stress dependent model which typically is used to model the elastic behavior of granular materials. It can be combined with plastic models, for example with the di Prisco plasticity model for soils, and with a poisson ratio.
In this model, the 'young’s stiffness’ modulus is made a function of the average stress state:

\[ E = E_0 + E_1 (p/p_1)^\alpha \]  

(3)

with condition however:

\[ E \leq E_2 \] 

(4)

where \( p \) is the pressure. The parameters \( E_0, E_1, E_2, p_1, \) and \( \alpha \) need to be specified in the *group_materi_elasti_young_power* record.

In this model, the poisson ratio is made a function of the average stress state:

\[ \nu = \nu_0 + \nu_1 (p/p_1)^\alpha \]  

(5)

with condition however:

\[ \nu \leq \nu_2 \]  

(6)

where \( p \) is the pressure. The parameters \( \nu_0, \nu_1, \nu_2, p_1, \) and \( \alpha \) need to be specified in the *group_materi_elasti_poisson_power* record.

The stiffness matrix \( C_{ijkl} \) for the Borja Tamagnini nonlinear elasticity model is specified in

The model contains \( G_0, \alpha, \hat{k} \) and \( p_r \) as user specified constants which need to be specified in the *group_materi_elasti_borja_tamagnini* record.

The Lade nonlinear elasticity is a stress dependent model which typically is used to model the elastic behavior of granular materials. It can be combined with plastic models, for example with the di Prisco plasticity model for soils.

The stress rates are linked to the strain rates by the equation:

\[ \dot{\epsilon}_{ij} = \frac{\partial W^2}{\partial \sigma_{ij} \partial \sigma_{hk}} \sigma_{hk} \]  

(7)

where the function \( W \) is

\[ W = \frac{X^{1-\lambda}}{2B(1-\lambda)} \]

where

\[ X = p^2 + R^* \text{abs}(s_{ij}s_{ij}) \]

with pressure \( p = (\sigma_{11} + \sigma_{22} + \sigma_{33})/3 \) and deviatoric stresses \( s_{ij} = \sigma_{ij} - p\delta_{ij} \).

The model contains three user specified constants \( B, R, \lambda \) which need to be specified in the *group_materi_elasti_lade* record. \( B \) and \( \lambda \) are defined by means of an isotropic unloading test, and \( R \) by means of an unloading-standard-triaxial-compression test. For example for a loose sand \( B = 1028, R = 0.25, \lambda = 0.28 \). See [10] for the details.

The model cannot be used in combination with a poisson ratio.
2.2.3 Elasto-Plasticity

Plastic strain

In plastic analysis, the \texttt{materi\_strain\_elasti} rate follows by subtracting from the \texttt{materi\_strain\_total} rate the \texttt{materi\_strain\_plasti} rate

\[ \epsilon_{ij}^{\text{elas}} = \epsilon_{ij} - \epsilon_{ij}^{\text{plas}} \]

where the \texttt{materi\_strain\_total} rate is

\[ \epsilon_{ij} = 0.5(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}) \]

The \texttt{materi\_strain\_plasti} rate follows from the condition that the stress cannot exceed the yield surface. This condition is specified by a yield function \( f^\text{yield}(\sigma_{ij}) = 0 \). The direction of the plastic strain rate is specified by the stress gradient of a flow function \( \frac{\partial f^\text{flow}}{\partial \sigma_{ij}} \). If the yield function and flow function are chosen to be the same, the plasticity is called associative, otherwise it is non-associative.

Von-Mises is typically used for metal plasticity. Mohr-Coulomb and Drucker-Prager are typically used for soils and other frictional materials. The plasticity models can freely be combined; the combination of the plasticity surfaces defines the total plasticity surface.

Typically, if you use Mohr-Coulomb or Drucker-Prager to model shear failure for soils, you should use the tension limiting model to limit tension stresses, preferably \texttt{group\_materi\_plasti\_tension\_direct}.

First some stress quantities which are used in most of the plasticity models are listed.

Equivalent Von-Mises stress:

\[ \overline{\sigma} = \sqrt{\frac{s_{ij}s_{ij}}{2}} \]

Mean stress:

\[ \sigma_m = \frac{\sigma_{11} + \sigma_{22} + \sigma_{33}}{3} \]

Deviatoric stress:

\[ s_{ij} = \sigma_{ij} - \sigma_m \delta_{ij} \]

CamClay plasticity model

Here we provide the equations of the Cam Clay model (Roscoe and Burland, 1968, summarized e.g. by Wood, 1990, see [20]). All stresses are effective (geotechnical) stresses, i.e. compression is positive! Definitions of variables:

\[ p = (\sigma_1 + \sigma_2 + \sigma_3)/3 \]

\[ q = \left\{ \frac{1}{2}[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2] \right\}^{1/2} \]

in the principal stress axes. The CamClay yield rule, which is also the flow rule, reads:
\[ f = g = q^2 - M^2[p(p_0 - p)] = 0 \]

\( M \) is a soil constant and \( p_0 \) is a history (hidden) variable which corresponds to the preconsolidation mean pressure. The hardening function, evolution, of \( p_0 \) reads:

\[ dp_0 = \frac{p_0(1 + \varepsilon)d\varepsilon_v^p}{\lambda - \kappa} \]

in which

\[ d\varepsilon_v^p = d\varepsilon_{11}^p + d\varepsilon_{22}^p + d\varepsilon_{33}^p \]

and \( \lambda \) and \( \kappa \) are user specified soil constants. Further \( \varepsilon \) is the void ratio with the evolution equation:

\[ d\varepsilon = -d\varepsilon_v(1 + \varepsilon) \]

in which

\[ d\varepsilon_v = d\varepsilon_{11} + d\varepsilon_{22} + d\varepsilon_{33} \]

Notice that this is a geometrical linear approximation for void ratio changes. The poisson ratio \( \nu \) reads:

\[ \nu = \frac{3K - 2G}{2G + 6K} \]

in which the elastic bulk modulus \( K \) is given by:

\[ K = (1 + \varepsilon)p/\kappa \]

and the Young’s modulus \( E \):

\[ E = 2. \ast G \ast (1 + \nu) \]

in which \( G \) is a user specified soil constant, By using this \( \nu \) and \( E \) the classical isotropic stress-strain law is used to calculate the stresses.

The soil constants \( M, \kappa, \lambda \) need to be specified in **group_materi_plasti_camclay**. The soil constant \( G \), need to be specified in **group_materi_elasti_camclay_g**. For an alternative see **group_materi_elasti_camclay_poisson**. The history variables \( \varepsilon, p_0 \) need to be initialized by **materi_plasti_camclay_history** record (and given initial values in node_dof records).

Remark 1: An additional parameter \( N \) can be often found in textbooks on the Cam Clay model. We don’t include it since it is linked to other model parameters via:

\[ 1 + \varepsilon = N - \lambda \ln p_0 + \kappa \ln(p_0/p) \]

Remark 2: If you apply a geometrical linear analysis, see section 8.4, then also the calculation of \( d\varepsilon \) void ratio development is linearized, and so will contain some error as compared to the exact
void ratio change. Hence for very large deformations, say above 10 percent or so, don’t use such geometrical linear analysis.

Remark 3: This camclay law is very sensitive for near-zero stresses. It is best to only use it in combination with application of initial gravity stresses by means of control_reset_dof.

Cap1 plasticity model

This group_materi_plasti_cap1 model is the first cap model that accounts for permanent plastic deformations under high pressures for granular materials. It is intended to be used in combination with shear plasticity models like Drucker-Prager, Mohr-Coulomb, etc.

First the average stress $p$ and the equivalent shear stress $q$ are introduced:

$$p = -(\sigma_{11} + \sigma_{22} + \sigma_{33})/3$$

$$q = \left\{ \frac{1}{2}[(\sigma_{11} - \sigma_{22})^2 + (\sigma_{22} - \sigma_{33})^2 + (\sigma_{33} - \sigma_{11})^2] + 3(\sigma_{12}^2 + \sigma_{23}^2 + \sigma_{31}^2) \right\}^{1/2}$$

These are used to define the cap plastic yield function:

$$f = \frac{q^2}{M^2} + p^* (p^* - p^*_c)$$

where

$$p^* = p + c \cot \phi \quad p^*_c = p_c + c \cot \phi$$

The parameter $p_c$ is a history variable of this model. The parameter $\phi$ is the coulomb friction angle, and $c$ is the cohesion. The parameter $M$ denotes the tangent of the Critical State Line in the model. Typically you can use:

$$M = \frac{6 \sin \phi}{3 - \sin \phi}$$

The history parameter $p_c$ is assumed to harden with the cap plastic volume strain rate according to the rate form:

$$\dot{\epsilon}_{cv}^{p} = \frac{\lambda^*}{\kappa^*} \left( \frac{p^{ref}}{p^*_c} \right)^m \dot{p}_c$$

Here $\kappa^*$ is the swelling index (e.g. 0.03), $\lambda^*$ is the compression index (e.g. 0.15), $K^{ref}$ is the bulk modulus at stress $p^{ref}$ (typically 100kPa), which typically can be taken as: $K^{ref} = \frac{E^{ref}}{3(1-2\nu)}$, and finally $m$ is an exponent (e.g. 0.6).

Initialize materi_plasti_cap1_history in the initialization part. The state variable $p_c$ for this hardening soil model enters the node_dof records. You need to give an initial value for it in the node_dof records. See also [2].

Cap2 plasticity model
This is the second cap model that accounts for permanent plastic deformations under high pressures for granular materials. It is intended to be used in combination with shear plasticity models like Drucker-Prager, Mohr-Coulomb, etc.

First a deviatoric stress measure \( t \) and hydrostatic stress measure \( p \) are defined

\[
\begin{align*}
    t &= \sqrt{3\bar{\sigma}} \\
    p &= -\sigma_m
\end{align*}
\]

See above for \( \bar{\sigma} \) and \( \sigma_m \). The yield rule for the \texttt{group\_materi\_plasti\_cap2} model reads:

\[
f = \sqrt{(p - p_a)^2 + \left[ \frac{Rt}{(1 + \alpha - \frac{Rt}{\cos\phi})^2} - R(c + p_a\tan\phi) \right]^2}
\]

Here \( c \) is the cohesion and \( \phi \) is the friction angle which should be taken equal to the values in the shear flow rule which you use. The parameter \( p_a \) follows from

\[
p_a = \frac{p_b - Rc}{1 + R \tan\phi}
\]

where the hydrostatic compression yield stress \( p_b \) is to be defined with an table of volumetric plastic strains \( \epsilon^p_v \) versus \( p_b \) with \( \epsilon^p_v = \epsilon^p_{11} + \epsilon^p_{22} + \epsilon^p_{33} \). As always, positive strain denote extension whereas negative strains denote compression.

Associative flow is used, so the flow rule is taken equal to the yield rule.

Summarizing the \texttt{group\_materi\_plasti\_cap2} model needs as input the cohesion \( c \), the friction angle \( \phi \), the parameter \( \alpha \) (typically \( 1.10^{-2} \) up to \( 5.10^{-2} \)), and a table \( \epsilon^p_v \) versus \( p_b \).

Compression limiting plasticity model

This \texttt{group\_materi\_plasti\_compression} model uses a special definition for the equivalent stress

\[
\bar{\sigma} = \sqrt{\sigma_{min}^2}
\]

where \( \sigma_{min} \) is the largest compressive principal stress. The model now reads

\[
\bar{\sigma} - \sigma_y = 0
\]

This plasticity surface limits the allowed compressive stresses.

di Prisco plasticity model

The di Prisco model is an non-associative plastic model for soils, which can be typically combined with the 'Lade elastic model'. This di Prisco model is a rather advanced soil model, which is explained in more detail in \cite{3} and \cite{9}. The yield rule reads:

\[
f = 3\beta f(\gamma - 3) \ln \left( \frac{r}{r_c} \right) - \gamma J_{3\eta^*} + \frac{9}{4}(\gamma - 1)J_{2\eta^*}
\]
and the flow rule yields:

\[ g = 9(\gamma - 3) \ln \left( \frac{r_g}{r_g} \right) - \gamma J_{3\eta^*} + \frac{9}{4}(\gamma - 1) J_{2\eta^*} \]

This is an anisotropic model in which the first and second invariant of the stress rate \( \eta^* \) are defined relative to the rotation axes \( \chi \).

\[
\begin{align*}
\eta^*_{hkl} &= \sqrt{3} s_{hkk}^* / r \\
\end{align*}
\]

where \( s^* \) follows from

\[
\begin{align*}
s_{hkk}^* &= \sigma_{hkk}^* - r \chi_{hkk} \\
\end{align*}
\]

Further \( r_g = 1 \).

The history variables are \( \chi_{ij} \) (rotation axes, 9 values), \( \beta \) (yield surface form factor), and \( r_c \) (preconsolidation mean pressure). The evolution laws for these history variables can be found in the papers listed above. The history variables \( \chi_{ij} \) (9 values), \( \beta \), \( r_c \) need to be initialized by the group_plasti_diprisco_history record (and should be given initial values in node_dof records). In a normally consolidated sand with isotropic initial conditions \( \chi_{ij} = \delta_{ij} \sqrt{3} \), \( \beta = 0.0001 \) and \( r_c \) equals \( \sqrt{3} \) times the means pressure.

The total model, yield rule and flow rule and evolution laws for history variables, contains a set of soil specific constants. In group_materi_plasti_diprisco you need to specify these constants. These constants are explained in more detail in the papers mentioned above, but here we give a short explanation. The constants \( \hat{\theta}_c \), \( \hat{\theta}_e \), \( \xi_c \) and \( \xi_e \) are linked to the dilatancy and the stress state during failure (standard triaxial compression and extension test in drained conditions). The constants \( \gamma \), \( c_p \), \( \beta_f \) and \( \beta_0^f \) are defined by means of the experimental curves (\( q-\epsilon_{axial}, \epsilon_{vol}-\epsilon_{axial} \)) obtained by performing a standard compression test in drained conditions. Moreover, \( \beta_f \), \( \beta_0^f \) and \( t_p \) can also be determined by means of the effective-stress path obtained by performing a standard triaxial compression test in undrained conditions.

Finally \( b_p \) can determined from an isotropic compression test. For a loose sand \( \hat{\theta}_c = 0.253, \hat{\theta}_e = 0.0398, \xi_c = -0.2585, \xi_e = -0.0394, \gamma = 3.7, c_p = 18., \beta_f = 0.5, \beta_0^f = 1.1, t_p = 10. \), and \( b_p = 0.0049 \).

**di Prisco plasticity model with varying density**

This essentially is the same as the normal di Prisco model, but instead of one set of parameters you need to specify two sets of parameters, one of loose soil and one for dense soil. The actual applied parameters will then be interpolated from the loose parameters and dense parameters depending on the actual density of the soil. The parameters need to be specified in group_materi_plasti_diprisco_density.

The history variables are those of group_materi_plasti_diprisco and finally extra the relative density (for example 20 or 40). So there are 12 history variables in total.
Drucker-Prager plasticity model

The `group_materi_plasti_druck_prag` model reads

\[ 3\alpha \sigma_m + \bar{\sigma} - K = 0 \]
\[ \alpha = \frac{2 \sin(\phi)}{\sqrt{3}(3 - \sin(\phi))} \]
\[ K = \frac{6c \cos(\phi)}{\sqrt{3}(3 - \sin(\phi))} \]

Here \( c \) is the cohesion, which needs to be specified both for the yield function and the flow rule; by choosing different values non-associative plasticity is obtained.

You should also include tension cut-off, preferably with `group_materi_plasti_tension_direct`.

Generalised Non Associate CamClay for Bonded Soils plasticity model

It is a modification of the 'Milan' model of Prof. Roberto Nova.

Gurson plasticity model

The `group_materi_plasti_gurson` model reads

\[ \frac{3\bar{\sigma}^2}{\sigma_y^2} + 2q_1 f^* \cosh(q_2 \frac{3\sigma_m}{2\sigma_y}) - (1 + (q_3 f^*)^2) = 0 \]

Here \( f^* \) is the volume fraction of voids. The rate equation

\[ f^* = (1 - f^*) f^{*\text{plas}}_{kk} \]

defines the evolution of \( f^* \) if the start value for \( f^* \) is specified. Furthermore, \( q_1, q_2 \) and \( q_3 \) are model parameters.

Hardening-Soil model

In this section, the principal stresses are ordered such that

\[ \sigma_3 > \sigma_2 > \sigma_1 \]

so that \( \sigma_1 \) is the largest compressive stress. Likewise for the principal plastic strains:

\[ \epsilon_3^p > \epsilon_2^p > \epsilon_1^p \]

First the elasticity parameters are defined. The elasticity parameters for the first loading are:

Young’s modulus = \( E_{50} = E_{50}^{ref} \left( \frac{\sigma_3 + c \cot \phi}{\sigma_{50}^{ref} + c \cot \phi} \right)^m \) and Poisson’s ratio = \( \nu_{50} \)

The elasticity parameters for the elastic unloading and reloading are:
Young’s modulus = \( E_{ur} = E_{ur}^{ref} \left( \frac{\sigma + c \cot \phi}{\sigma_{ur}^{ref} + c \cot \phi} \right)^m \) and Poisson’s ratio = \( \nu_{ur} \)

The yield function reads:

\[
f = \frac{1}{E_{50}} \frac{q}{1 - q/q_a} - \frac{2q}{E_{ur}} - \gamma^p
\]

where \( q \) is the equivalent shear stress and \( \gamma^p \) is the equivalent plastic shear strain.

The equivalent asymptotic shear stress reads

\[
q_a = \frac{q_f}{R_f}
\]

in which \( q_f \) is the shear failure stress, and \( R_f \) is the failure ratio.

Specify all elasticity parameters in group_materi_elasti_hardsoil. Typically you have:

- \( E_{50}^{ref} \) from experiment at stress \( \sigma_{50}^{ref} \)
- \( \nu_{ur} \) from experiment or the typical undrained value 0.495 or the typical drained value 0.3
- \( m \) from experiment or the typical value 0.5
- \( E_{ur}^{ref} \) from experiment at stress \( \sigma_{ur}^{ref} \), or the typical value \( 3E_{50}^{ref} \)
- \( \nu_{ur} \) from experiment or the typical undrained value 0.495 or the typical drained value 0.2

Specify all plasticity parameters in group_materi_plasti_hardsoil.

- \( \phi \) from experiment (maximum friction angle)
- \( c \) from experiment (cohesion)
- \( \psi \) from experiment (maximum dilatancy angle)
- \( R_f \) from experiment or the typical value 0.9 (failure ratio)

Initialize materi_strain_plasti_hardsoil in the initialization part. This causes that the node_dof records will be filled with the shear plastic strains. Also initialize materi_plasti_hardsoil_history.

You can add an initial contribution to the \( \gamma^p \) by setting control_materi_plasti_hardsoil_gammap_initial to -yes. This tells tochnog to create an extra contribution to \( \gamma^p \) exactly such that the yield function is zero-valued. This is convenient to start the calculation with hardsoil with deviatoric stresses which would have been outside the yield surface without this extra contribution. The extra addition to \( \gamma^p \) is saved in the record element_intpnt_materi_plasti_hardsoil_gammap_initial for each integration point of elements. The creation of this extra initial contribution is done in the first timestep of the timesteps of the corresponding control_timestep record with the same index.

See also [18] for some details. Especially notice that the model is more suited for monotonic loading than for load cycling (since it violates thermodynamics and tends to generate energy).

Mohr-Coulomb plasticity model
The `group_materi_plasti_mohr_coul` model reads

\[ 0.5(\sigma_1 - \sigma_3) + 0.5(\sigma_1 + \sigma_3) \sin(\phi) - c \cos(\phi) = 0 \]

Here \( c \) is the cohesion, \( \sigma_1 \) is the largest principal stress and \( \sigma_3 \) is the smallest principal stress. The angle \( \phi \) needs to be specified for both the yield condition and the flow rule; by choosing different values, non-associative plasticity is obtained.

As an alternative consider using `group_materi_plasti_mohr_coul_direct`, which is more stable and fast.

You should also include tension cut-off, preferably with `group_materi_plasti_tension_direct`.

Mohr-Coulomb hardening-softening plasticity model

The `group_materi_plasti_mohr_coul_hardening_softening` model is the same as the standard Mohr-Coulomb model. Now, however, the parameters \( c \) and \( \phi \) (both for the yield rule and for the flow rule) are softened on the effective plastic strain \( \kappa^{\text{shear}} \).

For example, for the cohesion a linear variation is taken between the initial value \( c_0 \) at \( \kappa^{\text{shear}} = 0 \), up to \( c_1 \) at a specified critical value of \( \kappa^{\text{shear}} \), and constant \( c_1 \) for larger values of \( \kappa^{\text{shear}} \). The same is done for \( \phi \) for the yield rule and for the flow rule.

You should also include tension cut-off, preferably with `group_materi_plasti_tension_direct`.

Multilaminate plasticity model

**Plastic yield function.**

The multi-laminate model predefines a number of weak planes, which have reduced plasticity parameters as compared to the bulk material. The numerical model will thus have the tendency to start slipping on the weak planes first, just like physical reality with weak planes. In fact, the yield function for each laminate amounts to a standard mohr-coulomb slip condition with predefined slip plane. The model reads

\[ f_k = (|\sigma_{pq}| + \sigma_{qq} \tan(\phi) - c)_k \]

where \( p \) denotes the in-plane direction of a laminate, \( q \) denotes the normal direction of the laminate, \( \phi \) denotes the friction angle of the laminate, \( c \) is the cohesion in the laminate, and finally \( k \) is the laminate number. The direction \( p \) is taken such in the plane of the laminate, that \( \sigma_{pq} \) is the maximum shear stress in the laminate plane. The stress \( \sigma_{qq} \) is normal to the laminate plane. The user needs to specify a normal vector \( n_{qk} \) to the plane of laminate \( k \), so that the plane of the laminate is precisely defined.

**Plastic flow rule.**

To allow for non-associated plastic flow, a dilatancy angle \( \psi \) is used:

\[ g_k = (|\sigma_{pq}| + \sigma_{qq} \tan(\psi) - c)_k \]

where again \( k \) denotes the number of the Multilaminate.
Elasto-plastic versus elasto-viscoplastic.

The multi-laminate plasticity model can be used elasto-plastic, but can also be used with viscoplasticity (time-dependent plasticity). In the latter case, you can apply the input data group_materi_plasti_visco_power_name and group_materi_plasti_visco_power_value.

Tension cutoff in laminates

To allow for laminate crack opening, you can specify a tension cutoff limit as yield function:

\[ f_k = (\sigma_{qq} - \sigma_t)_k \]

where \( \sigma_t \) is the maximum allowable tension stress, and \( k \) is again the laminate number. Specify this model with the input data group_materi_plasti_laminate0_tension.

Initialisation multi-laminate model

You always need to initialise materi_plasti_laminate with the number of required laminates. Optionally initialise materi_strain_plasti_laminate_mohr_coul etc. if you want to see the mohr-coulomb slip strains in the laminates. Optionally initialise materi_strain_plasti_laminate_tension etc. if you want to see tension cutoff strains in the laminates,

Status of laminates

The status of the mohr-coulomb yield condition in the integration points of elements can be found after a calculation in element_intpnt_plasti_laminate0_mohr_coul_status etc. Likewise, the status of the tension yield condition can be found in element_intpnt_plasti_laminate0_tension_status etc.

Tension limiting plasticity model

This group_materi_plasti_tension model uses a special definition for the equivalent stress

\[ \bar{\sigma} = \sqrt{\sigma_{\text{max}}^2} \]

where \( \sigma_{\text{max}} \) is the largest principal tension stress.

\[ \bar{\sigma} - \sigma_y = 0 \]

This plasticity surface limits the allowable tension stresses.

A simple model for concrete can be obtained as follows. Use group_materi_plasti_tension to limit the tension strength \( ft \). Use group_materi_plasti_vonmises to limit the compressive strength \( fc \). The tension strength could be softened to zero over an effective plastic strain \( \kappa \) of, say, 1 percent. The compressive strength could be softened to zero over an effective plastic strain \( \kappa \) of, say, 10 percent.
Von-Mises plasticity model

The *group_materi_plasti_vonmises* model reads

\[ \sqrt{3} \sigma - \sigma_y = 0 \]

where without hardening the yield value is fixed \( \sigma_y = \sigma_{y0} \).

If however the *group_materi_plasti_vonmises_nadai* hardening law for Von-Mises plasticity is specified then

\[ \sigma_y = \sigma_{y0} + C(\kappa_0 + \kappa)^n \]

where \( C, \kappa_0 \) and \( n \) are parameters for the hardening law, and \( \kappa \) is the isotropic hardening parameter (see later). The parameter \( \sigma_{y0} \) is specified by *group_materi_plasti_vonmises*.

Isotropic Hardening and softening

The size of the total plastic strains rate is measured by the *materi_plasti_kappa* parameter

\[ \dot{\kappa} = \sqrt{0.5 \epsilon_{ij}^{\text{plas},\text{plas}} \epsilon_{ij}^{\text{plas}}} \]

The size of the shear plastic strains rate is measured by the *materi_plasti_kappa_shear* parameter

\[ \dot{\kappa}^{\text{shear}} = \sqrt{0.5 \epsilon_{ij}^{\text{shear,plas},\text{shear,plas}} \epsilon_{ij}^{\text{shear,plas}}} \]

where the plastic shear strains are defined by

\[ \epsilon_{ij}^{\text{shear,plas}} = \epsilon_{ij}^{\text{plas}} - \delta_{ij} (\epsilon_{11}^{\text{plas}} + \epsilon_{22}^{\text{plas}} + \epsilon_{33}^{\text{plas}})/3 \]

These parameters \( \kappa \) and \( \kappa^{\text{shear}} \) can be used for isotropic hardening. Use the *dependency_diagram* for this.

Kinematic Hardening

The *materi_plasti_rho* matrix \( \rho_{ij} \), governs the kinematic hardening in the plasticity models. It is used in the yield rule and flow rule to get a new origin by using the argument \( \sigma_{ij} - \rho_{ij} \):

\[ f^{\text{yield}} = f^{\text{yield}}(\sigma_{ij} - \rho_{ij}) \]
\[ f^{\text{flow}} = f^{\text{flow}}(\sigma_{ij} - \rho_{ij}) \]

where the rate of the matrix \( \rho_{ij} \) is taken to be

\[ \rho_{ij} = a \epsilon_{ij}^{\text{plas}} \]

where \( a \) is a user specified factor (see *group_materi_plasti_kinematic_hardening*).

Plastic heat generation

The plastic energy loss can be partially turned into heat rate per unit volume \( q \):
\[ q = \eta \sigma_{ij} \epsilon_{ij}^{\text{plas}} \]

where \( \eta \) is a user specified parameter (between 0 and 1) specifying which part of the plastic energy loss is turned into heat (see \texttt{group\_materi\_plasti\_heat\_generation}).

### 2.2.4 Hypo-Plasticity

In hypoplasticity a direct relation is used between strain rates and effective stress rates. Rigid body rotations (objectivity) are treated elsewhere (see the section on memory). The effective stress tensor \( \sigma_{ij} \) follows from the total stress tensor \( \sigma_{ij} \) minus any pore pressures (see groundflow).

The Masin law is tuned to clays. The Wolffersdorff law is tuned to sands but can also be used for clays. The Niemunis visco law describes time dependent soil behaviour. If you need a cyclic law, you should use the Wolffersdorff law with intergranular strains and especially specify the correct theta. For many cycles the is-a-intergranular strain formulation can be used.

#### Masin law

The law proposed by MASIN [11] and [12] is used. This law is formulated in kPa; you need to make the remainder of the input file consistent with that.

The constitutive equation in rate form reads:

\[ \dot{T} = \mathcal{L} : D + f_d \|D\| \]  \hspace{1cm} (8)

where \( D \) is the Euler’s stretching tensor, \( T \) is the Cauchy stress tensor and

\[ \mathcal{L} = 3f_s \left( c_1 I + c_2 a^2 \hat{T} \otimes \hat{T} \right) \]

\[ N = \mathcal{L} : \left( -Y \frac{m}{\|m\|} \right) \]

\[ \hat{T} = \frac{T}{\text{tr}T} \]  \hspace{1cm} (9)

\( I \) is the second–order identity tensor and \( \mathcal{I} \) is the fourth–order identity tensor, with components:

\[ (\mathcal{I})_{ijkl} = \frac{1}{2} (1_{ik}1_{jl} + 1_{il}1_{jk}) \]  \hspace{1cm} (10)

The functions \( f_s(\text{tr}T) \) (barotropy factor) and \( f_d(\text{tr}T, e) \) (pyknentropy factor) are given by:

\[ f_s = -S_i \frac{\text{tr}T}{\lambda^*} \left( 3 + a^2 - 2a\sqrt{3} \right)^{-1} \]

\[ f_d = \left[ -\frac{2\text{tr}T}{3sp_r} \exp \left( \ln \left( 1 + e \right) - \frac{N}{\lambda^*} \right) \right]^\alpha \]  \hspace{1cm} (11)

where \( p_r \) is the reference stress for the parameter \( N \), typically taken as 1 kPa, and the factor \( S_i \) is a function of sensitivity \( s \):

\[ S_i = \frac{s - k(s - s_f)}{s} \]  \hspace{1cm} (12)

The scalar function \( Y \) and the second–order tensor \( m \) are given, respectively, by:

\[ Y = \left( \frac{\sqrt{3}a}{3 + a^2} - 1 \right) \frac{(I_1I_2 + 9I_3)(1 - \sin^2 \varphi_c)}{8I_3 \sin^2 \varphi_c} + \frac{\sqrt{3}a}{3 + a^2} \]  \hspace{1cm} (13)

in which:

\[ I_1 = \text{tr}T \]

\[ I_2 = \frac{1}{2} \left[ T : T - (I_1)^2 \right] \]

\[ I_3 = \det T \]

and

\[ m = - \frac{a}{F} \left[ \dot{T} + \dot{T}^* - \frac{T}{3} \left( \frac{6}{(F/a)^2 + T : T} \right) \right] \]  \hspace{1cm} (14)
in which:
\[
\hat{T}^* = \hat{T} - \frac{1}{3}
\]
\[
F = \sqrt{\frac{1}{8} \tan^2 \psi + \frac{2 - \tan^2 \psi}{2 + \sqrt{2} \tan \psi \cos 3\theta}} - \frac{1}{2\sqrt{2}} \tan \psi
\]  
(15)
\[
\tan \psi = \sqrt{3} \left\| \hat{T}^* \right\|
\]
\[
\cos 3\theta = -\sqrt{6} \frac{\text{tr} \left( \hat{T}^* \cdot \hat{T}^* \cdot \hat{T}^* \right)}{\left( \hat{T}^* \cdot \hat{T}^* \right)^{3/2}}
\]  
(16)

Finally, the scalars \(a\), \(\alpha\), \(c_1\) and \(c_2\) are given as functions of the material parameters \(\varphi_c\), \(\lambda^*\), \(\kappa^*\) and \(r\) by the following relations:
\[
a = \frac{\sqrt{3}(3 - \sin \varphi_c)}{2\sqrt{2}\sin \varphi_c}
\]
\[
\alpha = \frac{1}{\ln 2} \ln \left[ \frac{\lambda^* - \kappa^* S_i}{\lambda^* + \kappa^* S_i} \left( \frac{3 + a^2}{a\sqrt{3}} \right) \right]
\]
\[
c_1 = \frac{2(3 + a^2 - 2a\sqrt{3})}{9rS_i}
\]
\[
c_2 = 1 + (1 - c_1) \frac{3}{a^2}
\]  
(17)

Evolution of the state variables \(e\) (void ratio) and \(s\) (sensitivity) is governed by
\[
\dot{e} = (1 + e) \text{tr} \mathbf{D}
\]
(19)
\[
\dot{s} = -\frac{k}{\lambda^*} (s - s_f) \sqrt{\left( \dot{\epsilon}_v \right)^2 + \frac{A}{1 - A} \left( \dot{\epsilon}_s \right)^2}
\]  
(20)

where \(\dot{\epsilon}_v = \text{tr} \mathbf{D}\) and \(\dot{\epsilon}_s = \sqrt{2/3} \||\text{dev} \mathbf{D}||\).

The basic hypoplastic model requires five constitutive parameters, namely \(\varphi_c\), \(\lambda^*\), \(\kappa^*\), \(N\) and \(r\), state is characterised by the Cauchy stress \(\mathbf{T}\) and void ratio \(e\).

An extended model allows us to take into account the effects of meta-stable structure of natural clays. This extension requires three additional parameters \((k, A, s_f)\), and one additional state variable \(s\). A basic model without the structure effects is recovered if \(s = s_f = 1\) and \(A \neq 1\). The \(s\) should be always greater or equal to 1.

<table>
<thead>
<tr>
<th></th>
<th>(\varphi_c)</th>
<th>(\lambda^*)</th>
<th>(\kappa^*)</th>
<th>(N)</th>
<th>(r)</th>
<th>(k)</th>
<th>(A)</th>
<th>(s_f)</th>
</tr>
</thead>
<tbody>
<tr>
<td>London</td>
<td>22.6°</td>
<td>0.11</td>
<td>0.016</td>
<td>1.375</td>
<td>0.4</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Pisa</td>
<td>21.9°</td>
<td>0.14</td>
<td>0.0075</td>
<td>1.56</td>
<td>0.3</td>
<td>0.4</td>
<td>0.1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1: Typical parameters of the hypoplastic model for clays.

The basic law parameters should be specified in `group_materi_plasti_hypo_masin`. The extended parameters for the structure should be specified in `group_materi_plasti_hypo_masin_structure`. The hypoplastic history variables, \(e\) for this basic model, and \(e\) and \(s\) for the extended model, should be initialised with `materi_plasti_hypo_history`. As an alternative to specify the \(e\) you can specify the OCR at the start of the calculation in `group_materi_plasti_hypo_masin_ocr` (which is used to determine the initial \(e\) via \(e = \exp(N - \lambda^* \ln(|\text{OCR}|) - \lambda^* \ln(|p/p_r|)) - 1)\).

Masin clay law

This law is an improved version of the Masin law of the previous section. This law is formulated in kPa, so you should use that as units in your input file.

Anisotropy of sedimentary clays is such a significant feature of their mechanical behaviour that it cannot be ignored in finite element calculations. The hypoplastic model for clays with explicit formulation of the asymptotic state boundary surface is combined with an anisotropic form of the stiffness tensor. The resulting model predicts correctly the very small strain stiffness anisotropy. Anisotropy influences the undrained stress path, and this model is able to model such influence. For theoretical background see [13] and [14].
User input can be defined with the following records: `control_materi_plasti_hypo_masin_clay_ocr_advanced_parameters`, `group_materi_plasti_hypo_masin_clay`, `group_materi_plasti_hypo_masin_clay_advanced_direction`, `group_materi_plasti_hypo_masin_clay_ocr`, `group_materi_plasti_hypo_masin_clay_structure`, `group_materi_plasti_hypo_masin_clay_strain_intergranular_masin_clay`.

A biaxial input file showing all available options is given here below:

```
echo -no
number_of_space_dimensions 2
materi_velocity
materi_displacement
materi_strain_intergranular
materi_strain_total
materi_plasti_hypo_history
materi_stress
end_initia

node 1 0.00 0.00
node 2 1.00 0.00
node 3 0.00 1.00
node 4 1.00 1.00

element 1 -quad4 1 2 3 4

start_define
  left_edge geometry_line 0
end_define
left_edge 0.00 0.00 0.00 1.00 1.e-5
start_define
  lower_edge geometry_line 1
end_define
lower_edge 0.00 0.00 1.00 0.00 1.e-5
start_define
  upper_edge geometry_line 2
end_define
upper_edge 0.00 1.00 1.00 1.00 1.e-5
start_define
  right_edge geometry_line 3
end_define
right_edge 1.00 0.00 1.00 1.00 1.e-5

start_define
  sig0 -50.
end_define

force_edge_normal_geometry 0 -geometry_line 3
force_edge_normal 0 sig0

bounda_dof 0 -left_edge -velx
bounda_dof 1 -lower_edge -vely
bounda_dof 2 -upper_edge -vely
bounda_time 2 -1.
```
group_type 0 -materi

group_materi_plasti_hypo_masin_clay_ocr 0 1.2

group_materi_plasti_hypo_cohesion 0 0.

group_materi_plasti_hypo_masin_clay
  25 (phic in degrees)
  0.1 (lambda_star)
  0.01 (kappa_star)
  1 (N)
  0.2 (nu_pp)

(

group_materi_plasti_hypo_masin_clay_structure 0
  0.4 (k)
  0.1 (A)
  1.5 (s_f)


group_materi_plasti_hypo_strain_intergranular_masin_clay 0
  5.e-5 (R)
  270. (A_g)
  1.0 (n_g)
  0.5 (m_rat)
  0.08 (beta_r)
  7.0 (chi)
  7.0 (theta)


group_materi_plasti_hypo_masin_clay_visco 0
  1.e-6 (D_r)
  0.1 (I_v)


group_materi_plasti_hypo_masin_clay_ocr 0
  1.5 (ocr)


group_materi_plasti_hypo_masin_clay_advanced_double 0
  1.0 (alpha_g)
  equation 40 (alpha_f)
  0.3 (a_y)
  2.0 (o_c)

)

post_point 1 0.5 0.5

control_reset_dof 10 -hyhis0
control_reset_value_constant 10 0.6
control_reset_dof 20 -sigxx -sigzz
control_reset_value_constant 20 sig0
control_reset_dof 30 -sigyy
control_reset_value_constant 30 sig0
control_reset_dof 40 -hyhis4 (structure)
control_reset_value_constant 40 2
control_reset_dof 50 -epixx
control_reset_value_constant 50 -0.00001
control_reset_dof 60 -epiyy
control_reset_value_constant 60 -0.00001
control_reset_dof 70 -epizz
control_reset_value_constant 70 -0.00001

collection_print_history 80 -post_point_dof 1 -sigyy

control_timestep 90 1.e-4 0.1
control_materi_plasti_hypo_masin_clay_ocr_apply 90 -no
A final notice: this law only functions well if the OCR does not become below 1. If Tochnog detects a void ratio violating this condition, it will automatically change the void ratio in such a way that the OCR becomes 1, and the calculation can continue.

Wolffersdorff law

The law proposed by Wolffersdorff [19] is used.

\[
\dot{\sigma}_{ij} = L_{ijkl} \dot{\epsilon}_{ij} + f_d N_{ij} \sqrt{\dot{e}_{kl}} \dot{e}_{kl} = L_{ijkl} (d_{kl} - f_d Y m_{kl} ||d||)
\]

Here the part with \( L_{ijkl} \) gives a linear relation between strain rates and stress rates and the part with \( N_{ij} \) gives a nonlinear relation. The constitutive tensors \( L_{ijkl} \) and \( f_d N_{ij} \) are functions of the effective stress tensor \( \sigma_{ij} \) and void ratio \( e \). In the above \( d \) denotes the strain rate tensor \( \dot{\epsilon} \), \( Y \) denotes the degree of nonlinearity \( Y = ||L^{-1} : N|| \) and the flowrule \( m \) is defined by \( m = -(L^{-1} : N)^{-1} \) where \( \rightarrow \) denotes euclidian normalisation.

\[
L_{ijkl} = f_b f_e \frac{1}{\sigma_{mn} \sigma_{mn}} L^\ast_{ijkl}
\]

\[
N_{ij} = f_b f_e \frac{F a}{\sigma_{kl} \sigma_{kl}} (\dot{\sigma}_{ij} + \dot{\sigma}_{ij}^*)
\]

and \( \dot{\sigma}_{ij} = \sigma_{ij} / (\sigma_{mn} \sigma_{mn}) \), \( \dot{\sigma}_{ij}^* = \sigma_{ij} - \frac{1}{4} \delta_{ij} \), \( I_{ijkl} = \delta_{ik} \delta_{jl} \), \( a = \frac{\sqrt{3}(3 - \sin \varphi_c)}{2 \sqrt{2} \sin \varphi_c} \), \( F = \sqrt{1 - \tan^2 \psi} + \frac{2 - \tan^2 \psi}{2 + \sqrt{2} \tan \psi \cos 3\theta} - \frac{1}{2 \sqrt{2}} \tan \psi \), \( \tan \psi = \sqrt{3} \sqrt{\sigma_{ij}^* \sigma_{ij}} \), \( \cos 3\theta = -\sqrt{6} \frac{\sigma_{ij}^* \sigma_{jk} \sigma_{ki}}{[\sigma_{mn} \sigma_{mn}]^{3/2}} \).

For the \( L^\ast_{ijkl} \) above we have:

\[
L^\ast_{ijkl} = (F^2 I_{ijkl} + a^2 \dot{\sigma}_{ij} \dot{\sigma}_{kl})
\]

For \( \dot{\sigma}_{ij}^* = 0 \) is \( F = 1 \).

The scalar factors \( f_b, f_e \) and \( f_d \) take into account the influence of mean pressure and density:

\[
f_b = \frac{h_s}{n} \left( \frac{e_{i0}}{e_{c0}} \right)^\beta \left( 1 + e_i \right) \left( -\sigma_{ij} \delta_{ij} / h_s \right)^{1-n} \left[ 3 + a^2 - a \sqrt{3} \left( \frac{e_{i0} - e_{d0}}{e_{c0} - e_{d0}} \right) \right]^{-1}, \]

\[
f_d = \left( \frac{e - e_d}{e_c - e_d} \right)^\alpha.
\]

and \( f_e = (\frac{e_c}{e})^\beta \).

Three characteristic void ratios – \( e_i \) (during isotropic compression at the minimum density), \( e_c \) (critical void ratio) and \( e_d \) (maximum density) – decrease with mean stress:

\[
\frac{e_i}{e_{i0}} = \frac{e_c}{e_{c0}} = \frac{e_d}{e_{d0}} = \exp \left[ - \left( -\sigma_{ij} \delta_{ij} / h_s \right)^n \right]
\]
The range of admissible void ratios is limited by $e_i$ and $e_d$. The model parameters can be found in Tab. 2. They correspond to Hochstetten sand from the vicinity of Karlsruhe, Germany [19].

<table>
<thead>
<tr>
<th>$\varphi$ [°]</th>
<th>$h_s$ [MPa]</th>
<th>$n$</th>
<th>$e_{i0}$</th>
<th>$e_{d0}$</th>
<th>$e_{i0}$</th>
<th>$\alpha$</th>
<th>$\beta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>33</td>
<td>1000</td>
<td>0.25</td>
<td>0.95</td>
<td>0.55</td>
<td>1.05</td>
<td>0.25</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Table 2: Basic hypoplastic parameters of Hochstetten sand.

The basic law parameters should be specified in group_materi_plasti_hypo_wolffersdorff. The hypoplastic history variables should be initialised with materi_plasti_hypo_history.

Wolffersdorff law - extended Niemunis version

An extended version of this hypoplastic law is given in [16]. An extra term is added to $\hat{L}_{ijkl}$, in index notation:

$$b^2(\delta_{ik}\delta_{jl} - \frac{1}{3}\delta_{ij}\delta_{kl})$$

Here $b^2$ is given in [19], depending on the input parameter $\nu$, which can be seen as the classical Poisson ratio, typically 0.2 or 0.3. The first part of $f_d$ is determined:

- if $e \geq e_d$ then $f_d = \left(\frac{e - e_d}{e - e_c}\right)^\alpha$
- if $e < e_d$ then $f_d = -\left(\frac{e_d - e}{e - e_c}\right)^\alpha$

Then $\overline{f}_d$ is determined:

$$\overline{f}_d \text{ equation 4.221 in [16]}$$

where $M_e^{(d)}$ and $M_T^{(d)}$ equation 4.211 in [16]

Finally $f_d$ is checked

- if $(f_d < 1)$ then $f_d := f_d + (1 - f_d)^2 \overline{f}_d$

where $z$ is an input parameter specified, typically 5.

You need to specify the parameters $\nu$ and $z$ in group_materi_plasti_hypo_wolffersdorff_niemunis.

Wolffersdorff pressure dependent initial void ratio extension

You can correct the initial void ratio $e_0$, as specified in the initial value for the history variable in the node_dof records, for the initial pressure to obtain a corrected initial void ratio $e$.

$$\frac{e}{e_0} = \exp\left[-\left(-\frac{\sigma_{ij}\delta_{ij}}{h_s}\right)^n\right]$$

See the basic law description for the parameters $h_s$ and $n$. The $\sigma_{ij}$ denotes the effective stress tensor (total stresses minus any groundflow pressure). This pressure dependent initial void ratio correction can be activated by control_materi_plasti_hypo_pressure_dependent_void_ratio. After the initial void ratio has been established, the development of the void ratio is governed by volumetric compression or extension of the granular skeleton.

Niemunis visco law

For visco hypoplasticity with intergranular strains the stress rate reads:
\[ \dot{\sigma}_{ij} = M_{ijkl} \dot{\epsilon}_{kl} - L_{ijkl} \dot{\epsilon}_{kl}^{vis} \]

For visco hypoplasticity the \( L_{ijkl} \) reads:

\[ L_{ijkl} = f_b \dot{\epsilon}_{ij} \]

where

\[ f_b = \frac{-\sigma_{kk}}{(1 + a^2/3)\kappa} \]

where \( \kappa \) is a user specified material constant \( \kappa \) (= Butterfield’s swelling index upon isotropic unloading), and \( a \) relates to the user specified residual (=critical) friction angle \( \varphi_c \) as:

\[ a = \frac{\sqrt{3} (3 - \sin \varphi_c)}{2 \sqrt{2} \sin \varphi_c} \]

The pressure normalised stiffness is:

\[ \dot{L}_{ijkl} = F^2 I_{ijkl} + a^2 \dot{\sigma}_{ij} \dot{\sigma}_{kl} + b^2 (I_{ijkl} - \frac{1}{3} I_{ikjl}) \]

where

\[ b^2 = \frac{(1 + \frac{1}{3} a^2)(1 - 2\nu)}{1 + \nu} - 1 \]

Notice that the equation for \( \dot{e} \) only holds true for non-negative right-hand-side, so that puts limits on the allowed values for \( \varphi_c \) and \( \nu \).

For visco hypoplasticity the \( M_{ijkl} \) reads:

\[ M_{ijkl} = [\rho^x m_T + (1 - \rho^x) m_R] L_{ijkl} + \]

\[ + \begin{cases} 
\rho^x (1 - m_T) L_{ijkl} \dot{S}_{mn} \dot{S}_{kl} & \text{for } \dot{S}_{ij} \dot{\epsilon}_{ij} > 0 \\
\rho^x (m_R - m_T) L_{ijkl} \dot{S}_{mn} \dot{S}_{kl} & \text{for } \dot{S}_{ij} \dot{\epsilon}_{ij} \leq 0 
\end{cases} \]

where \( \dot{S} \) intergranular strains are the same as in the formulation without viscosity.

The viscosity strain rate is assumed to be:

\[ \dot{\epsilon}_{ij}^{vis} = D_r \dot{\epsilon}_{ij} \left( \frac{1}{OCR} \right)^{\frac{1}{\tau}} \]

where the normalised flow rule \( \dot{\epsilon}_{ij} \) is

\[ \dot{m}_{ij} = \frac{m_{ij}}{\sqrt{m_{ij} m_{ij}}} \]

with

\[ m_{ij} = - \left[ \frac{F^2}{a^2} (\dot{\sigma}_{ij} + \dot{\sigma}_{ij}^*) + \dot{\sigma}_{kl} \dot{\sigma}_{kl} \dot{\sigma}_{ij}^* - \dot{\sigma}_{ij} \dot{\sigma}_{kl} \dot{\sigma}_{kl}^* \right] \]

The over-consolidation ratio OCR appearing in the expression for the viscous creep rate is a function of the effective stress \( \sigma_{ij} \) and of the void ratio \( \epsilon_c \)

\[ OCR = \frac{p_{c}}{p_{e}^+} \]

wherein the void ratio is hidden in the equivalent pressure \( p_{e} \) and \( p_{e}^+ \) is a special stress invariant.

The equivalent pressure \( p_{e} \) is calculated from

\[ \ln \left( \frac{1 + e_{c0}}{1 + e} \right) = \lambda \ln \left( \frac{p_{e}}{p_{e0}} \right) \]
with a user specified material constant \( \lambda \) (= Butterfield’s first compression index) and also user-specified reference parameters \( e_0, p_0 \) which describe any pair of the void ratio and the effective pressure registered upon an isotropic \( D_r \)-isotach, i.e. during an isotropic first (= virgin) compression test with a constant volumetric rate of deformation equal to \(-\sqrt{3}D_r\frac{\lambda}{\sqrt{\lambda}}\).

The stress invariant \( p_e^+ \) is calculated using

\[
p_e^+ = \begin{cases} \frac{p}{p_R-1} \left[ \beta R \sqrt{1 + \eta^2 (\beta R^2 - 1)} - 1 \right] & \text{if } \eta < 1 \\ p(1 + \eta^2)^{1+\beta R} & \text{otherwise} \end{cases}
\]

wherein

\[
\eta = q/(Mp) \quad \text{and} \quad M = \frac{6 F \sin \varphi_c}{3 - \sin \varphi_c}
\]

where \( p = -\sigma_{kk}/3 \) and \( q = \sqrt{\frac{2}{3}} \sigma_{kl} \sigma_{kl} \) are the popular Roscoe’s stress invariants, and \( \beta_R \) (= flattening factor for the Rendulic’s cap) are the user supplied material constants.

You can specify an initial value of the void ratio \( e_0 \) in -hyhis0 with \texttt{control_reset_dof}. Then the OCR can be calculated with the above equations. As an alternative you can specify the OCR at the start of the calculation in \texttt{group_materi_plasti_hypo_niemunis_visco_ocr}; then the initial void ratio will be calculated as follows: \( p_e^+ \) will be determined from the equation above, then \( p_e^+ \) is determined from \( p_e = \text{OCR} p_e^+ \) and then the initial void ratio \( e_0 \) is determined from \( e_0 = (1 + e_e,0) \ast (p_e/p_e^+) \lambda - 1 \), (reference: Niemunis communications). Application of the specified OCR is triggered by \texttt{control_materi_plasti_hypo_niemunis_visco_ocr_apply}.

User parameters should be specified in \texttt{group_materi_plasti_hypo_niemunis_visco}. 

Cohesion extension

A simplistic approach to include cohesion is used here. Instead of feeding the real effective stress state \( \sigma_{ij} \) into the hypoplastic law, an alternative effective stress state \( \sigma_{cij} \) is used. Cohesion is modeled by subtracting in each of the normal stress components a value \( c \) representing cohesion:

\[
\sigma_{c11} = \sigma_{11} - c, \quad \sigma_{c22} = \sigma_{22} - c \quad \text{and} \quad \sigma_{c33} = \sigma_{33} - c.
\]

The shear stresses are not altered: \( \sigma_{c12} = \sigma_{12}, \) etc.

The cohesion value should be specified in \texttt{group_materi_plasti_hypo_cohesion}.

Intergranular strains extension

In order to take into account the recent deformation history, an additional tensorial state variable \( S_{ij} \) is introduced.

Denoting the normalized magnitude of \( S_{ij} \)

\[
\rho = \frac{\sqrt{S_{ij} S_{ij}}}{R}
\]

(R is a material parameter) and the direction of \( S_{ij} \)

\[
\hat{S}_{ij} = \frac{S_{ij}}{\sqrt{S_{kl} S_{kl}}}
\]

\( (\hat{S}_{ij} = 0 \text{ for } S_{ij} = 0) \), the evolution equation for the intergranular strain tensor reads:

\[
\dot{S}_{ij} = \begin{cases} (I_{ijkl} - \rho^{\delta r} \hat{S}_{ij} \hat{S}_{kl}) \dot{\epsilon}_{kl} & \text{for } \dot{S}_{ij} \dot{\epsilon}_{ij} > 0 \\ \dot{\epsilon}_{ij} & \text{for } \dot{S}_{ij} \dot{\epsilon}_{ij} \leq 0 \end{cases}
\]

\( S_{ij} \) is denoted \( \delta_{ij} \) in the paper [15]. However, in order to avoid confusion with Kronecker delta, another symbol is used here.
where \( \dot{S}_{ij} \) is the objective rate of intergranular strain. Rigid body rotations are treated elsewhere (see the section on memory). From the evolution equation \[2.2.4\] it follows that \( \rho \) must remain between 0 and 1.

The general stress-strain relation is now written as

\[
\dot{\sigma}_{ij} = M_{ijkl} \dot{\epsilon}_{kl} .
\]

The fourth order tensor \( M_{ijkl} \) represents the incremental stiffness and is calculated from the hypoplastic tensors \( L_{ijkl} \) and \( N_{ij} \) which may be modified by scalar multipliers \( m_T \) and \( m_R \), depending on \( \rho \) and on the product \( \dot{S}_{ij} \dot{\epsilon}_{ij} \):

\[
M_{ijkl} = [\rho \chi m_T + (1 - \rho \chi)m_R]L_{ijkl} + \\
\left\{ \begin{array}{ll}
\rho \chi (1 - m_T)L_{ijmn}\dot{S}_{mn}\dot{S}_{kl} + \rho \theta f_d N_{ij}\dot{S}_{kl} & \text{for } \dot{S}_{ij} \dot{\epsilon}_{ij} > 0 \\
\rho \chi (m_R - m_T)L_{ijmn}\dot{S}_{mn}\dot{S}_{kl} & \text{for } \dot{S}_{ij} \dot{\epsilon}_{ij} \leq 0
\end{array} \right.
\]

\( \chi \) and \( \theta \) are additional material parameters.

An example intergranular parameters can be found in Tab. 3.

<table>
<thead>
<tr>
<th>( R )</th>
<th>( m_R )</th>
<th>( m_T )</th>
<th>( \beta_r )</th>
<th>( \chi )</th>
<th>( \theta )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 \cdot 10^{-4}</td>
<td>5.0</td>
<td>2.0</td>
<td>0.50</td>
<td>6.0</td>
<td>6.0</td>
</tr>
</tbody>
</table>

Table 3: Example of Intergranular hypoplastic parameters.

The intergranular parameters should be specified in `group_materi_plasti_hypo_strain_intergranular`. Additionally you need to include `materi_strain_intergranular` in the initialisation part.

The additional parameter theta is very important only for the accumulation of permanent displacements or pore pressures in cyclic or dynamic analysis with small strains. For monotonic loading or higher strains theta is not very important. And thus for such monotonic loading or higher strains you should take \( \theta = \chi \).

ISA-Intergranular strains extension

Author: William Fuentes

The ISA plasticity \[6\] stands from Intergranular Strain Anisotropy and refers to a mathematical platform which allows to formulate new constitutive models or to extend existing hypoplastic models for soils. This family of models enables the simulation of small strain effects, such as the increase of stiffness and the reduction of the plastic accumulation under repetitive loops. Similar to the original intergranular strain concept by Niemunis and Herle (1996), it allows to couple existing hypoplastic relations to extend their capabilities for cyclic loading. However, the ISA plasticity contrasts with the previous formulation with the incorporation of an elastic locus related to a specific strain amplitude. This new feature makes its formulation elastoplastic, but the yield surface is now defined within the intergranular strain space.

The general stress-strain relation is now written according to:

\[
\sigma_{ij} = M_{ijkl} \epsilon_{kl}
\]

whereby \( M_{ijkl} \) is the stiffness tensor. The yield surface \( F \) of the model is defined within the space
of the intergranular strain $S_{ij}$ and distinguishes elastic conditions $F < 0$ from plastic conditions $F = 0$. The formulation of $M_{ijkl}$ reads:

$$M_{ijkl} = \begin{cases} m(L_{ijkl} + \rho f_d N_{ij} \hat{n}_{kl}) & \text{for } F \geq 0 \\ m_R L_{ijkl} & \text{for } F = 0 \end{cases}$$

whereby $m$, $\rho$, $f_d$ and $m_R$ are scalar factors, $N_{ij}$ is the non-linear stiffness tensor, $L_{ijkl}$ is the linear stiffness tensor and $\hat{n}_{kl}$ is the intergranular strain model flow rule. The ISA model proposes an evolution equation for the intergranular strain $S_{ij}$ with an elastoplastic formulation:

$$\dot{S}_{ij} = \dot{\varepsilon}_{ij} - \dot{\lambda} H \hat{n}_{ij}$$

whereby $\dot{\lambda} H$ is a plastic multiplier ($\dot{\lambda} H = 0$ for $F < 0$). The model may be enhanced to capture the plastic accumulation for larger number of cycles with an additional state variable $\varepsilon_{acc}$. This variable is able to distinguish between consecutive cycles or non-consecutive cycles [7]:

$$\varepsilon_{acc} = \varepsilon_{acc} + \frac{Ca}{R} (1 - y_h - \varepsilon_{acc}) \sqrt{\Delta \varepsilon_{ij} \Delta \varepsilon_{ij}}$$

The parameter $Ca$ controls the rate at which the plastic accumulation reduces upon consecutive cycles. The exponent $\chi$ is increased to $\chi_{max}$ to produce this effect on the constitutive equation. Typical parameters are $\chi_{max} = 20$ and $Ca = 0.017$. See also group_materi_plasti_hypo_strain_isa.

### 2.2.5 Damage

In the presence of materi_damage $d$, the materi_stress follows:

$$\sigma_{ij}^{\text{damaged}} = (1 - d)\sigma_{ij}^{\text{undamaged}}$$

For the damage, the group_materi_damage_mazars model is available:

$$d = d_t \alpha^\beta + d_c (1 - \alpha)^\beta$$

where

$$d_t = 1. - (1 - a_t) \frac{\epsilon^0}{\epsilon_{eq}} - a_t e^{-b_t (\epsilon^{eq} - \epsilon^0)}$$

and

$$d_c = 1. - (1 - a_c) \frac{\epsilon^0}{\epsilon_{eq}} - a_c e^{-b_t (\epsilon^{eq} - \epsilon^0)}$$

Here $\epsilon^{eq}$ contains the positive principal strains. The parameter $\alpha$ is given by the ratio $\epsilon^{eq} / \epsilon$, where $\epsilon$ contains the total strains (both negative and positive). The parameter $\epsilon^0$ is the strain threshold for damage; other material parameters are $\beta$ , $a_t$ , $b_t$ , $a_c$ , $b_c$. Typically for concrete:

$$1.e^{-4} < \epsilon^0 < 3.e^{-4} \; ; \; \beta = 1. \; ; \; 1 < a_t < 1.5 \; ; \; 500 < b_c < 2000 \; ; \; 0.7 < a_c < 1.2 \; ; \; e^4 < b_t < 5e^4$$

You can combine damage freely with plasticity models or other material behavior.
2.2.6 Average stress (hydrostatic compressibility)

An extra average stress contribution on each of $\sigma_{11}$, $\sigma_{22}$ and $\sigma_{33}$ is

$$\frac{1}{co} \frac{\partial v_i}{\partial x_i}$$

where $co$ is the `group_materi_elasti_compressibility`, which should not be 0. This pressure term can e.g. be used to model nearly incompressible fluids. The compressibility contribution should be combined with a contribution for the deviatoric stresses (e.g. `group_materi_viscosity`).

2.2.7 Undrained groundflow analysis

In case you want to perform an undrained groundflow analysis, but do not want to have both the material velocity and groundflow equations at the same time in system matrix, you can use `group_materi_undrained_capacity`. Then the following equation will be used to determine the total groundwater pressure changes in an element:

$$C \dot{p}_{\text{total}} = \frac{\partial v_i}{\partial x_i}$$

which actually is the groundflow storage equation without permeability. The above equation can be solved on an element-by-element level, so that the groundflow hydraulic pressure head and the storage equation do not need to be added to the complete system matrix. The capacity $C$ should be specified in `group_materi_undrained_capacity`. Results for the pressure in a element will be written to `element_intpnt_materi_undrained_pressure`. Application of this undrained analysis can be switched off and on with `control_materi_undrained_apply`.

This option is convenient to prevent the need for large, and ill-conditioned, system matrices in coupled soil - groundwater analysis. Typically the computational strategy may be like this:

```
... (include capacity for undrained analysis in relevant groups)
  group_materi_undrained_capacity ... 
...
  (set the hydraulic pressure heads, and fix them for the remainder of the calculation)
control_reset_dof ...-pres 
  bounda_dof ...-tpres 
...
  (solve material displacements in the remainder of the calculation)
control_timesteps ...
  control_materi_undrained_apply ... -yes 
...
```

The advantage of the above computational strategy is that never a system matrix with both material velocities and groundflow pressures needs to be solved. When solving the remainder of the calculation Tochnog uses the fixed total pressure from the hydraulic pressure heads plus the excessive undrained pressure of the remainder of the calculation as the full total pressure (when determining total stresses from effective stresses plus full total pressure). Alternatively to setting the hydraulic pressure head at the start with the `control_reset_dof`, you can also solve the gravity state for hydraulic pressure heads and material displacements (at the expense of a system
matrix with both material velocities and groundflow hydraulic pressures in this gravity calculation; but in the gravity calculation only and the remainder of the calculation).

2.2.8 Thermal stresses

Temperature rates cause fictitious thermal strain rates

\[ -\alpha \ddot{T}\delta_{ij} \quad \text{where} \quad \delta_{ij} = 1 \quad \text{if} \quad i = j \quad \text{else} \quad \delta_{ij} = 0 \]

where \( \alpha \) is the `group_materi_expansion_linear` coefficient and \( \ddot{T} \) is the `condif_temperature`. These fictitious thermal strain rates in turn lead to stress rates.

2.2.9 Hyper elasticity

Hyper elasticity is used to model rubbers. It should be combined with a total Lagrange formulation for the memory of the material (so use `-total` for `group_materi_memory`).

The stresses follow from a strain energy function (with \( C_{ij} \) components of the matrix \( C \), and where \( F \) is the deformation tensor and \( U \) is the stretch tensor following from the polar decomposition of the deformation tensor)

\[
2 \frac{\partial W}{\partial C_{ij}}
\]

\[
C = F^T F = U^T U
\]

Deviatoric contributions

To obtain a purely deviatoric function, the following strain measures are used (with \( I_1, I_2 \) and \( I_3 \) the first, second and third invariant of the elastic strain matrix \( C \) respectively)

\[
J_1 = I_1 I_3^{\frac{1}{2}} \\
J_2 = I_2 I_3^{\frac{1}{2}}
\]

The `group_materi_hyper_besseling` function reads (with \( K_1, K_2 \) and \( \alpha \) user defined constants)

\[
W = K_1(J_1 - 3)^\alpha + K_2(J_2 - 3)
\]

The `group_materi_hyper_mooney_rivlin` function reads (with \( G \) and \( \beta \) user defined constants)

\[
W = G * 0.5 * (I_1 - 3.0 + (2.0/\beta)(J^{-\beta} - 1.0));
\]

This Blatz-Ko hyperelastic material hardens in compression, and softens slightly in tension; it models a foamlike rubber.

The `group_materi_hyper_mooney_blatz_ko` function reads (with \( K_1 \) and \( K_2 \) user defined constants)
\[ W = K_1(J_1 - 3) + K_2(J_2 - 3) \]

The \texttt{group\_materi\_hyper\_neohookean} function reads (with \( K_1 \) a user defined constant)

\[ W = K_1(J_1 - 3) \]

The \texttt{group\_materi\_hyper\_reduced\_polynomial} function reads (with \( K_i \) user defined constants)

\[ W = K_i(J_1 - 3)^i \]

where a summation over \( i = 1, 2, \ldots \) is applied.

\textbf{Volumetric contributions}

First we define \( J = \sqrt{\mathbf{I}_3} \). Then a volumetric part can be added to the strain energy.

The \texttt{group\_materi\_hyper\_volumetric\_linear} contribution reads:

\[ W = \frac{K_2}{2} (J - 1)^2 \]

The \texttt{group\_materi\_hyper\_volumetric\_murnaghan} contribution reads:

\[ W = \frac{K}{\beta} \left( \frac{1}{\beta - 1} J^{-\beta} + 1 \right) J \]

The \texttt{group\_materi\_hyper\_volumetric\_polynomial} contribution reads:

\[ W = \frac{K_i}{2} (J - 1)^{2i} \]

for \( i = 0, 1, \ldots \).

The \texttt{group\_materi\_hyper\_volumetric\_simo\_taylor} contribution reads:

\[ W = \frac{K}{2} ((J - 1)^2 + (lnJ)^2) \]

The \texttt{group\_materi\_hyper\_volumetric\_ogden} contribution reads:

\[ W = \frac{K}{\beta} \left( \frac{1}{\beta} (J^{-\beta} - 1) + lnJ \right) \]

As an example, you can combine the \texttt{group\_materi\_hyper\_mooney\_rivlin} energy function with the \texttt{group\_materi\_hyper\_volumetric\_linear} so that the total strain energy function becomes:

\[ W = K_1(J_1 - 3) + K_2(J_2 - 3) + \frac{K}{2}(J - 1)^2 \]
Here the initial shear modulus and bulk modulus are included as:

\[
\text{initial shear modulus} = 2(K_1 + K_2)
\]

and

\[
\text{initial bulk modulus} = K
\]

respectively.

2.2.10 Viscoelasticity

Viscoelasticity is modeled with \( n \) parallel \texttt{group_materi_maxwell_chains}. Each of the chains contains a spring with stiffness \( E^m \) in line with a dash pot with relaxation time \( t^m \) (\( m \) indicates the \( m \)-th maxwell chain). The viscoelastic stress rate is given by (with \( C_{ijkl}^m \) is the elastic tensor modulus of the \( m \)-th maxwell chain (depending on \( E^m \) and the poisson ratio))

\[
\sum_{m=0}^{m=n-1} (C_{ijkl}^m \epsilon_{kl}^{\text{elas}} - \frac{\sigma_{ij}^m}{t^m})
\]

2.2.11 Viscoplasticity

Viscoplasticity is a model for rate-dependent plasticity. Rate dependent plasticity is important for (high-speed) transient plasticity calculations. It should be used in combination with a plasticity law. Viscoplasticity influences the stresses via the plastic strains.

The \texttt{group_materi_plasti_visco_exponential} model reads

\[
\dot{\epsilon}_{kl}^{\text{plas}} = \gamma_p \epsilon^{\alpha_f} \frac{\partial f_{\text{flow}}}{\partial \sigma_{kl}}
\]

where \( \gamma \) and \( \alpha \) are material fluidity constants and \( p \) is the pressure. In case the \( \alpha_f \) becomes larger than a limit, it is substituted by the limit to prevent the exponent from becoming excessive large. You can set the limit with the \texttt{group_materi_plasti_visco_exponential_limit} record. This model was first developed for visco-plastic soil behavior.

The \texttt{group_materi_plasti_visco_power} model reads

\[
\dot{\epsilon}_{kl}^{\text{plas}} = \eta(f)^p \frac{\partial f_{\text{flow}}}{\partial \sigma_{kl}}
\]

where \( \eta \) (fluidity constant), and \( p \) (power) are user specified parameters.

2.2.12 Viscosity

The viscous contribution to the total stress is
where

$$D_{ij} = 0.5 \left( \frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right)$$

and divergence is neglected since we only model slightly compressible flows.

**Viscous heat generation**

The viscous energy loss is turned into heat rate per unit volume $q$:

$$q = 2\nu D_{ij} D_{ij}$$

See group_materi_viscosity_heatgeneration.
2.3 Contact analysis

2.3.1 Penalty formulation

Attention: the contact algorithm is experimental and needs more testing.

In contact analysis, normal forces $F_n$ follow from the condition that bodies cannot penetrate each other. Since we use a penalty formulation, the normal force is given by

$$F_n = \lambda u_n$$

where $u$ is the penetration and $\lambda$ is called the contact_penalty_velocity because its generates forces on the velocity dof's. You can also impose groundflow_pressure and condif_temperature contact conditions by specifying the penalty factors contact_penalty_pressure and contact_penalty_temperature.

The contact algorithm is presently under development, and not guaranteed to work for all calculations. You must check your results.

2.3.2 Friction and frictional heat generation

This normal force leads to a friction force $F_f$ which equals

$$F_f = \nu F_n$$

where $\nu$ is the friction coefficient (see contact_plasti_friction. The friction force causes heat generation rate $Q$:

$$Q = \eta F_f v_f$$

where $v_f$ is the slip velocity, and the factor $\eta$ is a user specified factor which determines which part of the frictional energy loss is transformed into heat ($\eta$ is between 0 and 1; see contact_heat_generation).
2.4 Ground water flow

2.4.1 Storage equation for fully saturated analysis

The hydraulic pressure head \( h \) follows from the storage equation:

\[
c \frac{\partial h}{\partial t} = (k_p^1 \frac{\partial^2 h}{\partial x_1^2} + k_p^2 \frac{\partial^2 h}{\partial x_2^2} + k_p^3 \frac{\partial^2 h}{\partial x_3^2}) + \frac{\partial v_i}{\partial x_i} - \alpha \frac{\partial T}{\partial t} + f
\]

Primary dof is the hydraulic pressure head \( h \), which is initialised with \texttt{groundflow\_pressure} in the initialisation part, and which gets a label \texttt{-pres} in the \texttt{node\_dof} records. Further notation: \( c \) \texttt{group\_groundflow\_capacity}; \( k_p^i \) \texttt{group\_groundflow\_permeability} in \( i \)-direction (permeability); \( x_i \) space coordinate; \( v_i \) material velocity (if present); \( \alpha \) \texttt{group\_groundflow\_expansion} is the expansion coefficient of the groundwater for temperature changes. The equation is given for space coordinates following material velocities \( v_i \) (if present). The water mass acceleration (water newton inertia) is neglected in the above equation.

The bulk modulus of a soil-water mixture can be determined from experiments or from [21]. Following [21] one would find as theoretical value for \( K^m \):

\[
K^m = \frac{(K_g \times K_w)}{(K_w + n \times (K_g - K_w))}
\]

with \( K_g \) is the bulk modulus of the pure grain particles (so not the effective bulk modulus of the soil skeleton), \( K_w \) is the water bulk modulus (for example 100 MPa if water has some dissolved air) and \( n \) is the porosity. Then the following steps lead to the \( c \) of \texttt{group\_groundflow\_capacity}:

- Calculate \( K^m \) with the equation above or determine it with experiments
- Calculate \( K^s \) from the young and poisson (or if a hypoplastic law is used calculate it from the apparent young and poisson determined with post\_calcul\_materi\_stress\_young\_apparent\_materi\_stress\_poisson\_apparent)
- Calculate \( K \) with \( K = K^m - K^s \) (this takes care that if both water capacity and soil properties are in the input file, the total bulk modulus correctly becomes \( K^m = K + K^s \))
- Calculate \( C \) with \( c = \frac{1}{K} \).

Numerically, the groundwater capacity value is difficult to choose. Too low values leads to numerically un-stable calculations. Too high values leads to overly soft capacity.

Groundflow velocities

The groundflow velocities, after initializing \texttt{groundflow\_velocity}, follow from:

\[
v_i = k_p^i \frac{\partial h}{\partial x_i}
\]

Total groundwater pressure

The total groundwater pressure, or pore-pressure, is for example needed to calculate the total stresses in soils. The total groundwater pressure follows from:

\[
p_{\text{total}} = h - \rho g z
\]

where \( g \) is the gravitational acceleration, and \( \rho \) is the \texttt{groundflow\_density} (Please notice that \( g \) and \( z \) typically are negative numbers).
Tochnog considers pressure a pore pressure of \( p = 0 \), or positive, as indication that there is in fact no water pressure, so the porous soil skeleton is filled with air. In this case, the total soil stress is only composed by the effective stress of the soil skeleton.

The total stress in soils follows from: total soil stress = effective soil stress + total groundwater pressure. This will only be done for isoparametric finite elements which have groundflow data specified.

**Static groundwater pressure**

The static pressure due to gravity is:

\[
p_{\text{static}} = \rho g \Delta z
\]

where the \( \Delta z \) is the distance to the groundwater level, the phreatic level. The phreatic level needs to be specified with the `groundflow_phreatic_level` record. Alternatively you can specify `post_calcul_static_pressure_height`. If both `groundflow_phreatic_level` and `post_calcul_static_pressure_height` are not specified, the static pressure cannot be determined, so it remains zero.

**Dynamic groundwater pressure**

The dynamic groundwater pressure follows from

\[
p_{\text{dynamic}} = p_{\text{total}} - p_{\text{static}}
\]

**Boundary conditions**

If the groundwater velocity is 0 normal to an edge (say at the interface with a rock layer it is zero), then you should prescribe nothing on that edge (Tochnog will then take care of that boundary conditions for you).

At the phreatic level where the groundflow meets free air the hydraulic pressure head should become \( \rho gz \). You can either set this yourself by using `bounda_dof` combined with `bounda_time` or else demand that Tochnog automatically does it for you by activating the option `groundflow_phreatic_bounda`.

At edges where you have some other hydraulic pressure head you need to specify that head yourself with `bounda_dof` records. If you want to set directly the hydraulic head you need to use `-pres` in `bounda_dof`, however if you want to set the total pressure (pore pressure) you need to use `-topres` in `bounda_dof`.

If gravity is not of importance, e.g. in biomechanics where the storage equation is used to model fluid transport in soft tissues, the hydraulic pressure head \( h \) is equal to the total pressure, and thus is zero at edges where the water meets the free air. In this case, set \( h \) to zero by using `bounda_dof` combined with `bounda_time`.

**Postprocessing**

For all printing, plotting etc. you normally get the hydraulic pressure head \( h \) since it is the primary dof solved in the storage equation. The total pressure, static pressure and dynamic pressure are obtained using the `post_calcul` option.

**Naming conventions**

Following conventional naming, we remind the user that the capacity depends on the porosity \( n \) and water compressibility \( \beta \):
\[
C = n \beta
\]

and for the (intrinsic) permeability:

\[
k^p_i = \frac{k_i \rho |g|}{\rho |g|}
\]

where \(k_i\) is the hydraulic conductivity in \(i\)-direction.

### 2.4.2 Non-saturated analysis

with diagrams

You can perform a non-saturated analysis by making the permeability dependent on the groundwater total pressure (= pore pressure) by a dependency diagram. The diagram accounts for high permeability at saturation and low permeability at non-saturation. For example, do something like:

```
... dependency_item 10 -group_groundflow_permeability 0 -to_pres 4 dependency_diagram 10 -100. 0.0 0.05 100. 1.e-2 1.e-2 1.e-8 1.e-8 1.e-2 1.e-2 1.e-8 1.e-8 ...
```

The atmospheric air pressure is 0, so that is where the permeability starts changing it’s value in the table. You can also specify a table for `group_groundflow_capacity` to model non-saturated capacity.

**van Genuchten**

As an alternative to specifying diagrams you can use the specific van-Genuchten model for non-saturated ground water flow. The pore-pressure head is defined by

\[
\phi_p = -\frac{p}{\rho g}
\]

with \(p\) the pore pressure (= total pressure), \(\rho\) the ground water density and \(g\) is the absolute value of the gravity acceleration (typically 9.81). De degree of saturation is a function of the pore-pressure head

\[
S = S(\phi_p)
\]

The total capacity is the sum of the saturated capacity and a non-saturated part:

\[
c = c_{sat} + n \frac{dS(\phi_p)}{d\phi_p}
\]

where where \(c_{sat}\) is the saturated groundflow capacity as specified by `group_groundflow_capacity` and \(n\) is the porosity specified by `group_porosity`. The total permeabilities \(k_i\) are written as a relative factor of the saturated permeabilities

\[
k_i = k_{rel}(S)k_{sat,i}
\]
where \( k_i \) is the total permeability in direction \( i \), \( k_{\text{rel}}(S) \) is a factor dependent on the saturation \( S \) and \( k_{\text{sat},i} \) is the saturated permeability specified by \texttt{group\_groundflow\_permeability}.

Now for the van-Genuchten model we have

\[
S(\phi_p) = S_{\text{residu}} + (S_{\text{sat}} - S_{\text{residu}}) \left(1 + (g_a|\phi_p|)g_n\right)^{(1-g_n)/g_n}
\]

which has the following model parameters: \( S_{\text{residu}} \) is the residual saturation, \( S_{\text{sat}} \) normally is 1.0 but may be less than 1.0 if in case of trapped air, and \( g_a \) and \( g_n \) are constants to be determined experimentally. The derivative of this law defines the additional non-saturated capacity as defined above. After definition of the effective saturation \( S_e \)

\[
S_e = \frac{S - S_{\text{residu}}}{S_{\text{sat}} - S_{\text{residu}}}
\]

the relative permeability factor is defined as

\[
k_{\text{rel}}(S) = (S_e)^{g_n} \left(1 - (1 - S_e^{g_n/(g_n-1)})^{(g_n-1)/g_n}\right)^2
\]

To use the model you need to specify the saturated parameters \texttt{group\_groundflow\_capacity} and \texttt{group\_groundflow\_permeability} as usual, specify the porosity in \texttt{group\_porosity}, specify specific van-Genuchten parameters in \texttt{group\_groundflow\_nonsaturated\_vangenuchten} and initialise \texttt{groundflow\_saturation} in the initialisation part.

Since the model is strongly linear it might be needed to specify a relaxation of, say, 0.1 with \texttt{control\_relaxation} to obtain convergence.

### 2.4.3 Consolidation analysis

Look in the 'Consolidation' section of the 'Interaction analyzes and advanced analyzes' chapter in the end of this manual on how to perform consolidation analyzes (combined groundwater flow with soil stress analyzes).

In case you have \texttt{groundflow\_total\_pressure\_limit} set to 0 and the total pressure is 0, then Tochnog assumes that there is no water so the consolidation part in the equations will also be skipped. In case you have \texttt{groundflow\_total\_pressure\_limit} set to a high positive value this will not be done, so the consolidation part will also be used in case the total pressure is 0 (or positive).
2.5 Wave equation

\[ \frac{\partial \dot{s}}{\partial t} = c^2 \left( \frac{\partial^2 s}{\partial x_1^2} + \frac{\partial^2 s}{\partial x_2^2} + \frac{\partial^2 s}{\partial x_3^2} \right) \]

\[ \frac{\partial s}{\partial t} = \dot{s} \]

The primary dof’s are the wave scalar \( s \) and its first time derivative wave fscalar \( \dot{s} \) (as TOCHNOG only solves first order in time equations, the first time derivative of \( s \) also becomes primary dof in order to turn this second order in time equation into a set of first order in time equations). Further notation: \( x \) space coordinate, \( t \) time and \( c \) speed of sound.
2.6 Probabilistic distributions

The section summarises mathematical formulation of the so-called random finite element method, as described, e.g. in [8].

Distribution of a random variable (e.g., $C$) is controlled by these basic parameters: parameters of the statistical distribution (typically mean value $\mu_C$ and standard deviation $\sigma_C$) and so-called correlation length $\theta_C$ that controls spatial variability of variable $C$.

Two probabilistic distributions are available in Tochnog: normal distribution and log-normal distribution. Probability function $P(C)$ of normal distribution is defined as:

$$P(C) = \frac{1}{\sigma_C \sqrt{2\pi}} \exp \left[ -\frac{(C - \mu_C)^2}{2\sigma_C^2} \right]$$  \hspace{1cm} (21)

where $\mu_C$ is a mean value and $\sigma_C$ is standard deviation. Probability function $P(C)$ of log-normal distribution is defined as:

$$P(C) = \frac{1}{C \sigma_{\ln C} \sqrt{2\pi}} \exp \left[ -\frac{(\ln C - \mu_{\ln C})^2}{2\sigma_{\ln C}^2} \right]$$  \hspace{1cm} (22)

Quantities $\mu_{\ln C}$ and $\sigma_{\ln C}$ may be calculated from $\mu_C$ and $\sigma_C$ using

$$\sigma_{\ln C} = \sqrt{\ln \left[ 1 + \left( \frac{\sigma_C}{\mu_C} \right)^2 \right]}$$

$$\mu_{\ln C} = \ln \mu_C - \frac{1}{2} \sigma_{\ln C}^2$$  \hspace{1cm} (23)

2.6.1 Generation of random field

A number of different techniques to generate random fields is available (see, e.g., [5]). In this following, the most simple method based on Cholesky decomposition of the correlation matrix.

First, vector $X$ of statistically independent random numbers $x_1, x_2, \ldots, x_n$ (where $n$ is number of elements in the FE mesh) with a standard normal distribution (i.e., with probability function of Eq. (21) with $\mu_C = 0$ and $\sigma_C = 1$) is generated.

A correlation matrix $K$, which represents the correlation coefficient between each of the element used in the finite element analysis, is assembled. The correlation matrix $K$ has the following form:

$$K = \begin{bmatrix} 1 & \rho_{12} & \ldots & \rho_{1n} \\ \rho_{21} & 1 & \ldots & \rho_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{n1} & \rho_{n2} & \ldots & 1 \end{bmatrix}$$  \hspace{1cm} (24)

where $\rho_{ij}$ is the correlation coefficient between elements $i$ and $j$, calculated using Markov function:

$$\rho_{ij} = \exp \left[ -\frac{2x_{ij}}{\theta_C} \right]$$  \hspace{1cm} (25)

where $x_{ij}$ is absolute distance between elements $i$ and $j$ (distance between centers of gravity of elements $i$ and $j$). For anisotropic case Eq. (25) reads

$$\rho_{ij} = \exp \left[ -2 \sqrt{\left( \frac{\tau_{xij}}{\theta_{C_x}} \right)^2 + \left( \frac{\tau_{yij}}{\theta_{C_y}} \right)^2 + \left( \frac{\tau_{zij}}{\theta_{C_z}} \right)^2} \right]$$  \hspace{1cm} (26)

where $\theta_{C_x}$ is a correlation coefficient in direction of $x$-axis and $\tau_{xij}$ is a distance between two elements $i$ and $j$ in $x$ direction. The same notation applies for $y$ and $z$ directions.
The matrix $K$ is positive definite and hence, the standard Cholesky decomposition algorithm can be used to factor the matrix into upper and lower triangular forms, $S$ and $S^T$, respectively:

$$S^T S = K$$  \hfill (27)

The vector of correlated random variables $G$ (i.e., $G_1, G_2, \ldots, G_n$, where $G_i$ specifies the random component of variable $C$ in element $i$) is calculated by

$$G = S^T X$$  \hfill (28)

Vector $X$ is generated as described above.

Finally, value of the variable $C$ is assigned to each element ($C_i$) by the following transformation:

- for normally distributed variable $C$:
  $$C_i = \mu_C + \sigma_{CA} G_i$$  \hfill (29)

  where $\sigma_{CA}$ is calculated from $\sigma_C$ as described in the following section.

- for log-normally distributed variable $C$:
  $$C_i = \exp (\mu_{\ln C} + \sigma_{\ln CA} G_i)$$  \hfill (30)

  where $\mu_{\ln C}$ is calculated by Eq. (23)b using $\sigma_{\ln CA}$ instead of $\sigma_{\ln C}$; $\sigma_{\ln CA}$ is calculated from $\sigma_{\ln C}$ as described in the following section.

### 2.6.2 Local averaging

The input parameters of $C$ that relate to the mean, standard deviation and spatial correlation length are assumed to be defined at the point level. Due to the finite size of each finite element, point statistical distribution must be averaged over the element. This results in reduced $\sigma_{\ln C}$ in the case of log-normal distribution and reduced $\sigma_C$ in the case of normal distribution. $\mu_{\ln C}$ in the first case and $\mu_C$ in the second case remain unaffected.

The locally-averaged standard deviations ($\sigma_{\ln CA}, \sigma_{CA}$), which are used in Eqns. (29, 30), are calculated from their point values using

$$\sigma_{\ln CA}^2 = \gamma \sigma_{\ln C}^2 \quad \quad \sigma_{CA}^2 = \gamma \sigma_C^2$$  \hfill (31)

where $\gamma$ is the variance reduction factor calculated by integration of the Markov function \textsuperscript{23}. In 1D for a finite element of side length $\alpha \theta_C$:

$$\gamma = \frac{2}{(\alpha \theta_C)^2} \int_0^{\alpha \theta_C} \exp \left( -\frac{2}{\theta_C} \sqrt{x^2} \right) (\alpha \theta_C - x) dx$$  \hfill (32)

In 2D for square finite element of side length $\alpha \theta_C$:

$$\gamma = \frac{4}{(\alpha \theta_C)^4} \int_0^{\alpha \theta_C} \int_0^{\alpha \theta_C} \exp \left( -\frac{2}{\theta_C} \sqrt{x^2 + y^2} \right) (\alpha \theta_C - x)(\alpha \theta_C - y) dxdy$$  \hfill (33)

In 3D for hexahedral finite element of side length $\alpha \theta_C$:

$$\gamma = \frac{8}{(\alpha \theta_C)^5} \int_0^{\alpha \theta_C} \int_0^{\alpha \theta_C} \int_0^{\alpha \theta_C} \exp \left( -\frac{2}{\theta_C} \sqrt{x^2 + y^2 + z^2} \right) (\alpha \theta_C - x)(\alpha \theta_C - y)(\alpha \theta_C - z) dxdydz$$  \hfill (34)

For the anisotropic case in 2D:

$$\gamma = \frac{4}{l^4} \int_0^l \int_0^l \left[ -2 \sqrt{\frac{x}{\theta_C x}} + \left( \frac{y}{\theta_C y} \right)^2 \right] (l - x)(l - y) dxdy$$  \hfill (35)
and for the anisotropic case in 3D:

$$\gamma = \frac{8}{l^6} \int_0^1 \int_0^1 \int_0^1 \exp \left[ -2 \sqrt{\left( \frac{x}{\theta_{C_x}} \right)^2 + \left( \frac{y}{\theta_{C_y}} \right)^2 + \left( \frac{z}{\theta_{C_z}} \right)^2} \right] (l-x)(l-y)(l-z) \, dx \, dy \, dz \quad (36)$$

In order to calculate the variance reduction due to local averaging correctly, all elements in the mesh should be of the same size and all elements should be regular squares. If irregular elements are used, exact value of $\gamma$ is in Tochnog approximated by calculation of $\gamma$ for an equivalent square element using Eq. (34) with area equal to an average area of all elements in the mesh.

The approximate value of $\gamma$ requires that you use as much as possible elements of the same size and shape in the complete calculation domain.

### 2.6.3 Monte Carlo simulations

The most simple but very powerful technique to solve the probabilistic problem is a Monte Carlo technique. The same problem is solved many times, each time with different fields of random variables generated according to prescribed parameters.

The whole problem is solved in the following steps:

1. Generate random fields according to Sec. 2.6.1 using `control_distribute` command as many times as many variables are treated as random. In principle, any variable can be related as random. For example material parameters, dof’s (e.g., history variables), etc.

2. Solve the problem using finite element method. Collect required results of each Monte Carlo realisation into an output file. The user can prescribe any result to be collected into an output file using `control_repeat_save` command (e.g., `time_current`, final displacement of a selected point, etc.).

3. Repeat items 1. and 2. $m$-times, where $m$ is a prescribed number of Monte Carlo realisations. $m$ value is specified in Tochnog input file using `control_repeat` command.

4. Evaluate results statistically. More complex statistical evaluation is done by the user, calculation of mean value and standard deviation can be done in Tochnog using `control_repeat_save_calculate` command.

### 2.6.4 Input data records

A typical piece of input file could be like this:

```plaintext
print_group_data ... (print in the gid files distributed group data so that you get a plot of it)
...
control_distribute 10 ... (distribute something with correlation in space)
control_distribute_parameters 10 ...
control_distribute_correlation_length 10 ...
control_distribute 20 ... (distribute something else without correlation in space)
control_distribute_parameters 20 ...
...
control_timestep 30 ... (do timesteps)
control_timestep_iterations_automatic 30 ... (with automatic timestepping)
```
control_timestep_iterations_automatic_stop 30 -continue (don’t abort the calculation if the minimum step size is reached, e.g. in a stability calculation)

... control_print_data_versus_data 40 ... (save data for repeats in a dvd file)

... control_repeat 50 100 10 (jump 100 times back to control index 10)
control_repeat_save 50 ... (select results to be saved for each repeat)
control_repeat_save_calculate 50 -yes (perform statistical analysis on saved results)

... control_print_gid 100 -yes
control_print 100 -repeat_save_result -repeat_save_calculate_result

...
3 Input file, general remarks

The input is free format. Comments are enclosed between ( ), e.g. (this is comment only); a comma , is not allowed inside comments. The input should consist of an initialization part and a data part, separated by `end_initia` and ended by `end_data`

```
initialization
...
initialization
end_initia
data_item index data_values
...
end_data
```

**Bold** printed data in this manual can be used literally. *Italic* printed data is only symbolic (it represents a number or a word).
4 Input file, initialization part

The initialization part contains initialization records and an end_initia record

```
initialization

... 
initialization
end_initia
```

The example below initializes a solid material

```
echo -yes
number_of_space_dimensions 2
materi_velocity
materi_strain_total
materi_stress
end_initia
```

The echo (always the first record), number_of_space_dimensions (always the second record), and end_initia record should be supplied always. Use echo -yes to echo the input and echo -no to not echo the input. Use number_of_space_dimensions 1 for 1D problems, etc.. The records materi_velocity, materi_strain_total and materi_stress create a velocity, strain and stress field in the entire domain. In the following sections, all possible initialization records are discussed. Most of these records create a dofield, a physical field like a temperature field or a strain field, over the computational domain.

4.1 echo switch (first record of initialization part)

If switch is -yes the input will be echoed. If switch is -no the input will not be echoed. This needs to be the first record.

4.2 number_of_space_dimensions number_of_space_dimensions (second record of initialization part)

Set number_of_space_dimensions to 1 in 1D, etc.. This needs to be the second record.

4.3 derivatives (third record of initialization part, if specified)

If this record is included, the time derivative and the space derivatives will be stored in the node_dof records. This is only required for a limited number of models. The model description will specify if this derivatives initialization is needed.

4.4 beam_rotation

The beam rotations \(\phi_x\), \(\phi_y\) and \(\phi_z\) are added to the node_dof records.
Please notice that always all three rotations are included. Typically for a 2D calculation you may want to fix the $\phi_x$ and $\phi_y$ to 0, by using a `bounda_dof` record.

4.5 condif_temperature

The temperature $T$ is added to the `node_dof` records.

4.6 groundflow_pressure

The pressure $p$ is added to the `node_dof` records.

4.7 groundflow_pressure_gradient

The gradient of the hydraulic pressure $\frac{dh}{dx} \frac{dh}{dy} \frac{dh}{dz}$ is added to the `node_dof` records.

4.8 groundflow_saturation

The groundwater saturation $S$ is added to the `node_dof` records.

4.9 groundflow_velocity

The ground water flow velocity $v_i^k$ is added to the `node_dof` records.

4.10 materi_damage

The damage $d$ is added to the `node_dof` records. Also `materi_velocity` and `materi_strain_total` should be initialized.

4.11 materi_acceleration

The accelerations $a_i$ are added to the `node_dof` records.

4.12 materi_displacement

The displacements $u, v, w$ are added to the `node_dof` records. If `materi_displacement` is initialized, then equations like the convection and diffusion of heat equation or the ground water flow equation are evaluated over the deformed volume, which is the sum of the nodal coordinates plus its displacements. Also if `materi_displacement` is initialized, the total Lagrange model will be used in stress analysis.

Condition: also `materi_velocity` should be initialized because the displacement follows from integration of the velocity.
4.13 materi_displacement_relative

Displacement relative to a previous point in the calculation. These are the current displacements minus the displacements before these were changed with timesteps in control_timestep or a displacement reset in control_reset_dof.

For example, this option comes handy when you want to understand the extra displacements caused by the last timesteps.

4.14 materi_history_variable number_of_variables

Generic history variables which can e.g. be used in some user supplied routines or otherwise.

4.15 materi_maxwell_stress number_of_chains

Maxwell stress $\sigma_{11}^m \sigma_{12}^m \sigma_{13}^m \sigma_{22}^m \sigma_{23}^m \sigma_{33}^m$ is added to the node_dof records. The parameter number_of_chains should match data item group_materi_maxwell_chains. The number of maxwell stresses is $6 \times$ number_of_chains.

4.16 materi_plasti_camclay_history

The history variables $e_0$ and $p_0$ for the camclay plasticity models are added to the node_dof records.

4.17 materi_plasti_cap1_history

The history variable $p_c$ for the cap1 plasticity models is added to the node_dof records.

4.18 materi_plasti_diprisco_history number_of_history_variables

The history variable di Prisco plasticity models are added to the node_dof records. For the group_materi_plasti_diprisco model you need to set number_of_history_variables to 11. For the group_materi_plasti_diprisco_density model you need to set number_of_history_variables to 12.

4.19 materi_plasti_f

The plastic yield rule $f$ is added to the node_dof records. This should only be used for elasto-plastic calculations, and not for visco-plastic calculations.

4.20 materi_plasti_f_nonlocal

The nonlocal plastic yield rule $fn$ is added to the node_dof records. See also: nonlocal.
4.21 materi_plasti_generalised_non_associate_cam_clay_for_bonded_soils_history

The history variables for the Generalised Non Associate Cam Clay for Bonded Soils plasticity model are added to the node_dof records.

4.22 materi_plasti_hardsoil_history

The history variable $abs(p)$ for the hardsoil plasticity model is added to the node_dof records. It contains the maximum pressure history.

4.23 materi_plasti_hypo_history

Eight history variables for the hypo-plasticity models are added to the node_dof records.

The first history variable -hyhis0 contains the void ratio, and should be initialized by initially specifying node_dof records, or optionally specifying an OCR value with a group_materi_plasti_hypo....._ocr record.

The second history variable -hyhis1 will be filled with the time step size of the hypoplastic substepping scheme; this is meant for postprocessing only.

The third history variable -hyhis2 will be filled with the mobilized friction angle in degrees; this is meant for postprocessing only.

The fourth history variable -hyhis3 will be filled with the a measure of the effective stiffness following from the hypoplasticity law ($\sqrt{M_{ijkl}M_{ijkl}}$); this is meant for postprocessing only.

The fifth history variable -hyhis4 for all laws with the structure option, will be filled with the structure variable $s$ and should be initialized by initially specifying node_dof records.

The sixth history variable -hyhis5, will be filled with the OCR value; this is meant for postprocessing only. This is only available for the group_materi_plasti_hypo_masin_clay and group_materi_plasti_hypo_niemunis_visco laws.

The seventh history variable -hyhis6 for the group_materi_plasti_hypo_wolffersdorff, will be filled with the density index $I_d = \frac{ec - e}{ec - ed}$; this is meant for postprocessing only.

The eight history variable -hyhis7 for hypoplasticity laws with intergranular strains, will be filled with the intergranular strain variable $\rho$; this is meant for postprocessing only.

4.24 materi_plasti_kappa

The size of the plastic strain $\kappa$ is added to the node_dof records. See the theory section.

4.25 materi_plasti_kappa_shear

The size of the shear part of the plastic strain $\kappa^{\text{shear}}$ is added to the node_dof records. See the theory section.
4.26 **materi_plasti_laminate** *number_of_laminates*

This initialises the number of laminates for the multilaminate plasticity model. At most 6 is allowed for *number_of_laminates*.

4.27 **materi_plasti_phimob**

The mobilized friction angle $\phi_{mob}$ is added to the *node_dof* records. It is defined as the angle, in radians, for which the yield function

$$ f = 0.5(\sigma_2 - \sigma_0) + 0.5(\sigma_2 + \sigma_0) \ast \sin(\phi_{mob}) - c \cos(\phi_{mob}) $$

becomes zero. This is available for mohr-coulomb plasticity only. Please realise that in regions with substantial cohesion the mobilized friction angle $\phi_{mob}$ can exceed the friction angle $\phi$ from the plasticity law. In case of zero cohesion, or cohesion small relative to the stresses, yield is reached if the $\phi_{mob}$ reaches the friction angle $\phi$. The definition above can give either negative or positive values for $\phi_{mob}$; negative values simply indicate that the stress state is far away from yielding. The phimob is calculated in degrees.

4.28 **materi_plasti_rho**

The plastic kinematic hardening vector $\rho_{11} \rho_{12} \rho_{13} \rho_{22} \rho_{23} \rho_{33}$ is added to the *node_dof* records. See also *group_materi_plasti_kinematic_hardening*.

4.29 **materi_strain_energy**

The material strain energy $0.5\sigma_{ij}\epsilon_{ij}^{elas}$ is added to the *node_dof* records. You can print or plot it to see where energy is stored after loading. Also *materi_stress* and *materi_strain_elasti* should be initialised.

4.30 **materi_strain_elasti**

The elastic strain $\epsilon_{kl}^{elas}$ is added to the *node_dof* records. See also: *materi_strain_total*.

4.31 **materi_strain_intergranular**

The intergranular strain $S_{ij}$ is added to the *node_dof* records. This can be used by hypoplasticity laws, see the theory section.

4.32 **materi_strain_isa_c**

The ISA intergranular back-strain $C_{ij}$ is added to the *node_dof* records. This can be used by hypoplasticity laws with ISA-intergranular strains, see the theory section.
4.33 materi_strain_isa_eacc

The ISA intergranular accumulated-strain $\epsilon_{\text{acc}}$ is added to the node_dof records. This can be used by hypoplasticity laws with ISA-intergranular strains, see the theory section.

4.34 materi_strain_plasti

The plastic strain $\epsilon_{kl}^{\text{plas}}$ is added to the node_dof records. See also: materi_strain_total.

4.35 materi_strain_plasti_camclay

The plastic strain $\epsilon_{kl}^{\text{plas}}$ specifically for the camclay model is added to the node_dof records. See also: materi_strain_plasti.

4.36 materi_strain_plasti_cap

The plastic strain $\epsilon_{kl}^{\text{plas}}$ specifically for cap models is added to the node_dof records. See also: materi_strain_plasti.

4.37 materi_strain_plasti_compression

The plastic strain $\epsilon_{kl}^{\text{plas}}$ specifically for the compression model is added to the node_dof records. See also: materi_strain_plasti.

4.38 materi_strain_plasti_diprisco

The plastic strain $\epsilon_{kl}^{\text{plas}}$ specifically for the diprisco model is added to the node_dof records. See also: materi_strain_plasti.

4.39 materi_strain_plasti_generalised_non_associate_cam_clay_for_bonded

The plastic strain $\epsilon_{kl}^{\text{plas}}$ specifically for the generalised non associate cam clay for bonded soils model is added to the node_dof records. See also: materi_strain_plasti.

4.40 materi_strain_plasti_druckprag

The plastic strain $\epsilon_{kl}^{\text{plas}}$ specifically for the drucker-prager model is added to the node_dof records. See also: materi_strain_plasti.

4.41 materi_strain_plasti_hardsoil

The plastic strain $\epsilon_{kl}^{\text{plas}}$ specifically for the hardsoil model is added to the node_dof records. See also: materi_strain_plasti.
4.42 materi_strain_plasti_laminate_mohr_coul

This record initialises for the laminates materi_strain_plasti_laminate0_mohr_coul, materi_strain_plasti_laminate1_mohr_coul, etc up to materi_strain_plasti_laminate_mohr_coul.

The materi_strain_plasti_laminate0_mohr_coul is the mohr-coulomb plastic strain specifically for laminate 0,
the materi_strain_plasti_laminate1_mohr_coul is the mohr-coulomb plastic strain specifically for laminate 1, etc.
The materi_strain_plasti_laminate_mohr_coul is the mohr-coulomb plastic strain for all laminates together.
See also: materi_strain_plasti.

4.43 materi_strain_plasti_laminate_tension

This record initialises for the laminates materi_strain_plasti_laminate0_tension, materi_strain_plasti_laminate1_tension, etc up to materi_strain_plasti_laminate_tension.

The materi_strain_plasti_laminate0_tension is the tension cutoff plastic strain specifically for laminate 0,
the materi_strain_plasti_laminate1_tension is the tension cutoff plastic strain specifically for laminate 1, etc.
The materi_strain_plasti_laminate_tension is the tension cutoff plastic strain for all laminates together.
See also: materi_strain_plasti.

4.44 materi_strain_plasti_mohr_coul

The plastic strain $\epsilon_{kl}^{\text{plas}}$ specifically for the mohr_coulomb models is added to the node_dof records. See also: materi_strain_plasti.

4.45 materi_strain_plasti_tension

The plastic strain $\epsilon_{kl}^{\text{plas}}$ specifically for the tension model is added to the node_dof records. See also: materi_strain_plasti.

4.46 materi_strain_plasti_vonmises

The plastic strain $\epsilon_{kl}^{\text{plas}}$ specifically for the von-mises model is added to the node_dof records. See also: materi_strain_plasti.

4.47 materi_strain_total

The total strain $\epsilon_{kl}$ is added to the node_dof records. All strains are time integrals of the strain rates for a specific material particle which happens to be present in the node.
4.48 materi_strain_total_kappa

The maximum strain size is added to the node_dof records.

4.49 materi_strain_total_compression_kappa

The maximum principal compression total strain as occurred in history is added to the node_dof records.

4.50 materi_strain_total_shear_kappa

The maximum size of the deviatoric part of the total strain as occurred in history is added to the node_dof records.

4.51 materi_strain_total_tension_kappa

The maximum principal tensional total strain as occurred in history is added to the node_dof records.

4.52 materi_stress

The stresses $\sigma_{11}$ $\sigma_{12}$ $\sigma_{13}$ $\sigma_{22}$ $\sigma_{23}$ $\sigma_{33}$ are added to the node_dof records.

4.53 materi_stress_pressure_history

The maximum of the absolute value of the pressure which occurs over time is added to the node_dof records. See also group_materi_elasti_stress_pressure_history_factor in the data part.

4.54 materi_velocity

The velocities $v_i$ are added to the node_dof records.

4.55 materi_velocity_integrated

The integrated velocities $v_{ii}$ are added to the node_dof records. The integration of nodal velocities in fact results in displacements. But asking for these integrated velocities doesn’t activate automatically that the calculation is done over the total deformed volume (as is the case when you initialize materi_displacement), and not automatically a total Lagrange model is used in stress analysis.
4.56 materi_void_fraction

The material void fraction $f^*$ is added to the node_dof records. This is required for the group_materi_plasti_gurson model.

4.57 materi_work

The material second order work $\dot{\sigma}_{ij}\dot{\epsilon}_{ij}$ is added to the node_dof records. You can print or plot it to see where material instabilities are present.

4.58 mrange maximum_range_length

Sets the maximum length of ranges -ra . . . -ra.

4.59 mstring maximum_number_of_strings

Sets the maximum number of strings in a define block.

4.60 wave_scalar

Scalar in wave equation is node_dof records. Condition: also wave_fscalar should be initialized.

4.61 wave_fscalar

The first time derivative in the wave equation is added to the node_dof records. Condition: also wave_scalar should be initialized.

4.62 end_initia (last record of initialization part)
5 Input file, data part, introduction

Data items in the data part are used to control the calculation, select required output, give dof’s initial values, etc.. Note that an end_data record is needed.

```
data_item index data_values
...
data_item index data_values
end_data
```

Consider the following example

```
element 0 -tria3 0 1 2
element 1 -tria3 1 2 3
node 0 0. 0.
note 1 1. 0.
note 2 0. 1.
note 3 1. 1.
...
end_data
```

Note that the data items element and node are indexed. In fact most data items need to be indexed. Indexing starts at 0 (all numbering in TOCHNOG starts at 0). Indices need not strictly be sequential (e.g. only the indices 1,2 and 5 of a data item may be specified).

The following sections first treat some extras that can be used in the data part. After that, all possible data items are specified.

Arithmetic blocks

You also can use the arithmetic expressions plus, minus, multiply and divide. We show some examples:

```
(make A equal to 4.1)
start_arithmetic
A 1.1 plus 3
end_arithmetic
....
(make B equal to 3.2)
start_arithmetic
B 3.2
end_arithmetic
....
(make C equal to 7.3)
start_arithmetic
C A plus B
end_arithmetic
....
(make D equal to 14.6)
start_arithmetic
```

99
D A plus B multiply 2.
end_arithmetic

Expressions will be evaluated from left to right. Words from define blocks will not be recognized in arithmetic blocks.

Automatic counting: the counters

The words counter_a, counter_b, counter_c and counter_d are reserved words in the data part. If they are found, they will be substituted by their integer value. After its value is substituted, the counter will be incremented by 1. Initially the value for counters is 0. The example below shows a typical application.

start_define
left_edge geometry_line counter_a
end_define
start_define
right_edge geometry_line counter_a
end_define
...
left_edge 0. 0. 0. 10. 1.e-4
right_edge 2. 0. 2. 10. 1.e-4
...
bounda_dof 1 -left_edge -velx
bounda_time 1 0.
bounda_dof 2 -right_edge -velx
bounda_time 2 1.3
...

Notice that we automatically give the geometry lines a unique number in this way; the unique number is not really of interest in the remainder of the input file, so the application of a counter is convenient.

Finally, also the words counter_a_apply, counter_b_apply, counter_c_apply and counter_d_apply are available. They will be substituted by the current value of the counters, without that the counters are incremented.

Conditional blocks

Parts of the input file can be processed conditionally within start_if ... end_if blocks. This is illustrated below with an example:

Example:

start_define
do_complete_calculation true
end_define
...
start_if do_complete_calculation
...

start_if do_complete_calculation
end_if

...  

The part in the start_if ... end_if block is only done if do_complete_calculation is set to true, like in the example. If do_complete_calculation is set to false that part will be skipped. You also can use start_if_not ... end_if_not blocks, so that actions are NOT taken if the defined variable is set to true.

Control indices

All possible data items are defined in the following sections. It only makes sense to specify some of the data items before the calculation; the other data items are only meant to be printed after the calculation. The example below specifies a 1D temperature calculation.

```
echo -no
  number_of_space_dimensions 1
  condif_temperature
end_initia
	node 1 0
	node 2 1
	node 3 2

element 1 -bar2 1 2

element 2 -bar2 2 3

bounda_dof 0 1 -temp
bounda_time 0 0.0 0. 1. 1. 100. 1.
bounda_dof 1 3 -temp
bounda_time 1 0.0 0.0 100.0 0.

group_type 0 -condif
  group_condif_density 0 1.0
  group_condif_capacity 0 0.1
  group_condif_conductivity 0 0.1
  group_condif_flow 0 0.

ccontrol_timestep 0 0.1 10.0
ccontrol_print 0 -time_current -node_dof
ccontrol_print_database 1 -separate_index
ccontrol_timestep 2 0.2 10.0
ccontrol_print 2 -time_current -node_dof

end_data
```

Note how the indices of control items like control_timestep and control_print are used to control the sequence of events. First, (index=0) time steps of size 0.1 are taken and for each time step results are printed. Then (index=1) the database is printed which can serve as a point of restart. Finally (index=2) time steps of size 0.2 are taken and for each time step results are printed.

Define blocks
You can define a word to represent a set of strings. For each word defined, you need to specify a `start_define ... end_define` block. Within the block, you first specify the word, and then you specify the set of strings. Later in the data part, you can use the defined words as the replacement of the set of strings.

Example:

```plaintext
start_define
velocity 1.34
end_define
start_define
left_edge geometry_line 1
end_define
... left_edge 0. 0. 0. 10. 1.e-4
... bounda_dof 1 -left_edge -velx
bounda_time 1 0. 0. 100. velocity
...```

The words `plus`, `minus`, `multiply` and `divide` as used in arithmetic blocks are prohibited in define blocks.

Include files

You can use `include filename` in the data part, to request that the file with name `filename` is included. This is handy to include often used data parts, or include a mesh generated by a pre-processor, etc.

The included file itself is not allowed to have an `include` in the data part.

The included file should not contain comments ( ... ). The included file needs to be ended with an `end_data`. On some MS windows computers two `end_data` records are needed, so try that in case of trouble. On MS windows 32 bit computers include is not available.

Numbering of values in records

The numbering of values in records in illustrated by `node_dof` records. Look at the following piece of input file:

```plaintext...
number_of_space_dimensions 2
materi_velocity
materi_stress
end_initia
...
node_dof 1 0.0 0.0 -1.0 0.0 0.0 -1.0 0.0 -1.0
node_dof 2 0.0 0.0 -1.0 0.0 0.0 -1.0 0.0 -1.0
...
end_data
...```
Here \texttt{node\_dof} records 1 and 2 are initialized. The initial velocities are 0, and for the initial stresses we use $\sigma_{xx} = -1$, $\sigma_{yy} = -1$ and $\sigma_{zz} = -1$. The total list of dof’s in the \texttt{node\_dof} record is \texttt{-velx, -vely, -sigxx, -sigxy, -sigxz, -sigyy, -sigyz} and \texttt{-sigzz}.

We refer to \texttt{-velx} as the 0’th value in the \texttt{node\_dof} record, \texttt{-vely} as the 1’th value, etc. So printing the history of the \texttt{-sigxx} stress of \texttt{node\_dof} record 1 is obtained by this:

\begin{verbatim}
... control_timestep 10 ...
control_print_history 10 -node_dof 1 2 ...
end_data
\end{verbatim}

where the number 2 refers to the \texttt{-sigxx} stress. See also the definition of the \texttt{control\_print\_history} record for this. As an alternative, sometimes you can use names instead of numbers, for example here:

\begin{verbatim}
... control_timestep 10 ...
control_print_history 10 -node_dof 1 -sigxx ...
end_data
\end{verbatim}

Ranges

Ranges are general input formats used for indices and data values. Possible ranges are illustrated by the following examples

- \texttt{-all}
  - \texttt{-ra 12 32 44 -ra}
  - \texttt{-ra -from 5 -to 16 -ra}
  - \texttt{-ra -from 5 -to 25 -step 2 -ra}

The \texttt{-all} range is not available for indices.

The data values for a data item can be specified as a range if this is allowed for in the description of the data item. All words in the data part (or part of an index) need to be proceeded with a ‘tic’ (-). In the example the \texttt{node\_dof} records 1 to 100 are initialized

\begin{verbatim}
node_dof -ra -from 1 -to 100 -ra 1. 0. 0.
\end{verbatim}

Types of dof’s

Some of the dof’s are principal dof’s: these are \texttt{materi\_velocity, condif\_temperature, ground-flow\_pressure, wave\_scalar}. These are the dof’s which are solved by the equilibrium equations (conservation laws).
The other dof’s, like `materi_stress` and so, follow from these principal dof’s (strains follow from displacement derivatives, stresses follow from strains by material laws, etc.).

Furthermore, for all the dof’s we have primary values, which are the dof’s themselves, and derived dof’s, which are the space and time derivatives of the primary dof’s.
6 Input file, data part, data records

6.1 area_element_group

This record is used to generate element_group records. Each element, all of whose nodes are part of the geometry_item, will get an element_group record with value element_group. Please realise that the geometry entity can be a two-dimensional area, a volume, etc.

This option comes handy whenever a part of the domain gets some specific element data. For example, this would be the case if different areas in the structure have different material properties like stiffness, etc.

Beware: any directly specified element_group records will be overwritten. Attention: default this area_element_group will only evaluated at the very start of the calculation. By setting switch in area_element_group_time to -yes however you can require that this record is evaluated at all times.

We show here two ways to get different element data in different regions: Both ways give elements with young 1.2 from x=0 to x=1, and elements with young 3.3 from x=1 to x=2.

First way:

```
.. node 1 0.
node 2 1.
ode 3 2.
element 1 -bar2 1 2
element 2 -bar2 2 3
element_group 1 0
element_group 2 1
.. group_type 0 -materi
group_materi_elasti_young 0 1.2
group_type 1 -materi
group_materi_elasti_young 1 3.3
..<control_mesh_refine_globally 10 -h_refinement
..<
```

Second way:

```
.. node 1 0.
node 2 1.
ode 3 2.
element 1 -bar2 1 2
element 2 -bar2 2 3
..<
group_type 0 -materi
group_materi_elasti_young 0 1.2
```
group_type 1 -materi
  group_materi_elasti_young 1 3.3
  ...
  geometry_line 1 0. 1. 1.e-4
  geometry_line 2 1. 2. 1.e-4
  area_element_group 1 -geometry_line 1 0
  area_element_group 2 -geometry_line 2 1
  ..

See also area_element_group_method, area_element_group_sequence_element_group etc.

6.2 area_element_group_element index name

With area_element_group_element you select the name of the elements for which the area_element_group will be used; if this area_element_group_element is not specified then all elements will be used.

6.3 area_element_group_interface index switch

If switch is set to -yes the area_element_group record with the same index will also be used for interface elements. If switch is set to -no the area_element_group record with the same index will not be used for interface elements. Presently the switch can be only set to -no.

6.4 area_element_group_method index method

Set method to -all or -any. If method is set to -all, then the corresponding area_element_group is applied to elements for which all nodes are inside the specified geometry. If method is set to -any, then the corresponding area_element_group is applied to elements for which any of the nodes is inside the specified geometry. Default method is -all.

6.5 area_element_group_node index node_0 node_1 ...element_group

Similar to area_element_group. Now, however, directly the global node numbers are specified.

6.6 area_element_group_time index switch

If switch is set to -yes the corresponding area_element_group is evaluated at all times (as opposed to only at the start of the calculation).

6.7 area_element_group_sequence index element_0 element_1 ...

See area_element_group_sequence_element_group.
6.8  **area_element_group_sequence_element**  *index name*

See **area_element_group_sequence_element_group**.

6.9  **area_element_group_sequence_element_group**  *index group_0 group_1 ...

**General description**

This option works more or less the same as the **area_element_group** option. Read that description first.

With this option however, you can specify what the element group numbers of an area (geometry), or set of element numbers, will be in time. This allows for an easy modeling of change of material models.

This option works in combination with the **area_element_group_sequence_** records (with the same index).

See also **control_mesh_change_element_group**.

**Selection of elements for which the element group changes over time**

With **area_element_group_sequence_geometry** you select the area (geometry) for which the time sequence of group numbers should be used.

With **area_element_group_sequence** you select the elements for which the time sequence of group numbers should be used.

You can use both **area_element_group_sequence_geometry** and **area_element_group_sequence** to select a combination of elements in a geometry and directly specified element numbers. As a completely separate option do not use any of **area_element_group_sequence_geometry** and **area_element_group_sequence** at all. Then at a time point time_i the elements which have group number group_(i-1) will get new group number group_i. So the previous group number of elements is used to set the current group number of elements (and geometries are not used to change the group numbers).

With **area_element_group_sequence_element** you select the name of the elements for which the sequence of time versus group will be used; if this **area_element_group_sequence_element** is not specified then all elements will be used.

**Specification of new element group numbers in time**

With **area_element_group_sequence_time** and **area_element_group_sequence_element_group** you select time points at which groups should become active; for example, group_0 becomes active at time_0 etc.

**Remarks**

Remark 1: If you want the stresses, strains, etc. to be reset to 0. when the element group changes, then use a **control_reset_geometry** record for that.

Remark 2: It is more convenient and clear to use the **start_define end_define** option to define the geometries.

**Examples**
Example:

```
area_element_group_sequence_geometry 0 -geometry_brick 1
area_element_group_sequence_element 0 -hex8
area_element_group_sequence_time 0 0. 2. 3.
area_element_group_sequence_element_group 0 1 5 4

group_type 1 ...
...
group_type 5 ...
...
group_type 4 ...
...
control_reset_geometry 10 -geometry_brick 1
...
```

In the selected geometry element group 1 will be used starting from time 0 for elements -hex8. Starting from time 2 element group 5 will be used, etc. Same example, now with defines however:

```
start_define
soil_empty_wall geometry_brick 1
end_define
...

area_element_group_sequence_geometry 0 -soil_empty_wall
area_element_group_sequence_element 0 -hex8
area_element_group_sequence_time 0 0. 2. 3.
area_element_group_sequence_element_group 0 1 5 4

group_type 1 ...
...
group_type 5 ...
...
group_type 4 ...
...
control_reset_geometry 10 -soil_empty_wall
...
```

Now an example of the separate option:

```
area_element_group_sequence_time 0 0. 2. 3.
area_element_group_sequence_element_group 0 1 5 4

element_group 77 1
element_group 78 1

group_type 1 ...
...
group_type 5 ...
...
```
At time 0, elements 77 and 78 have group number 1. At time 2, the elements with group number 1 get group number 5. At time 3, the elements with group number 5 get group number 4.

6.10 area_element_group_sequence_geometry index geometry_entity_item geometry_entity_index

See area_element_group_sequence_element_group.

6.11 area_element_group_sequence_geometry_method index method

Set method to -all or -any. If method is set to -all, then the corresponding area_element_group_sequence_geometry is applied to elements for which all nodes are inside the specified geometry. If method is set to -any, then the corresponding area_element_group_sequence_geometry is applied to elements for which any of the nodes is inside the specified geometry. Default method is -all.

6.12 area_element_group_sequence_interface index switch

If switch is set to -yes the area_element_group_sequence_* will be used for interface elements also. If switch is set to -no the area_element_group_sequence_* will not be used for interface elements. Default switch is set to -no.

6.13 area_element_group_sequence_time index time_0 time_1 ...

See area_element_group_sequence_element_group.

6.14 area_node_dataitem index geometry_entity_item geometry_entity_index data_item_name

This record is used to generate data_item_name records on all nodes located on the specified geometrical entity. The values for the data_item_name should be specified in the area_node_dataitem_double record for real precision values, or in the area_node_dataitem_integer record for integer values (or words).

6.15 area_node_dataitem_double index value_0 value_1 ...

See area_node_dataitem.

6.16 area_node_dataitem_integer index value_0 value_1 ...

See area_node_dataitem.
6.17 axisymmetric switch

If switch is set to -yes, the calculation becomes axi-symmetrical. If switch is set to -no, the calculation becomes not axi-symmetrical. In case a group_axisymmetric is specified for some group, that overrules this axisymmetric record.

6.18 bounda_alternate index bounda_index_0 bounda_index_1 ...

This option takes care that between successive iterations only one of the specified bounda_dof is not used. For example if bounda_dof records with index 10, 20 and 30 are present in the input file, and you use bounda_alternate 10 20 30 then in subsequent iterations the following index is not used: 10, 20, 30, 10, 20, 30, 10, ... etc.

This option comes handy to allow for very large calculations on a computer with limited memory. By putting alternating boundary conditions on velocities, pressures or temperatures the system of active equations to be solved in each iterations is only of a limited size. And then using enough iterations the solutions for all dof’s can slowly converge to the actual coupled solution.

As example consider a large 3d calculation where displacements and hydraulic pressure heads need to be solved:

```
solver_matrix_symmetric -yes
... bounda_alternate 10 20 30 40
bounda_dof 10 -all -velx
bounda_dof 20 -all -vely
bounda_dof 30 -all -velz
bounda_dof 40 -all -pres
... control_timestep 100 ..
control_timestep_iterations 100 20
```

The above bounda_dof records are additional to the normally present records, like fixing displacements at sides of the domain, boundary conditions on hydraulic pressure, etc. The bounda_alternate record instructs tochnog to subsequently neglect the record 10, 20, 30, 40, 10, ..., etc. When a record is neglected the corresponding solution field can be solved. For example in the first iteration the solution field for the x-displacement can be solved, while the y-displacement and z-displacement and hydraulic pressure head are kept fixed. And thus the total system of equations is much smaller, approximately 4 times less dof’s need to be solved by the pardiso solver, which in fact is the bottleneck in computer memory usage for very large calculations. Notice that we asked tochnog to use the symmetric equation solver, since the pressures and displacements are not used simultaneously, so we don’t have the disadvantage of a non-symmetric matrix with displacement and pressure contributions.

As another example we use a classical staggered solution for displacements and water pressures:

```
solver_matrix_symmetric -yes
... bounda_alternate 10 20
```
bounda_dof 10 -all -velx -vely -velz
bounda_dof 20 -all -pres
...
control_timestep 100 ..
control_timestep_iterations 100 20

You should not specify bounda_time records i.c.w. bounda_dof records which are used in bounda_alternate. The bounda_time records will not be used.

6.19 bounda_baseline_correction time_start time_end

If this record is specified baseline correction is performed after one of:

- reading SMC files with uncorrected accelerations in bounda_dof i.c.w. bounda_time_smc.
- direct specification of acceleration in bounda_dof i.c.w. bounda_time.

Such baseline correction is needed to suppress artificial drift in velocity signals following from the acceleration signal.

The correction actually is done by adding a parabolic acceleration signal to the specified accelerations, thus giving a corrected acceleration in time. The parabolic (second order) signal contains three constant coefficients. These are determined by demanding that the corrected acceleration signal leads to a minimal sum of squared velocities over the considered time interval.

This correction is done over the time interval from time_start up to time_end. Typically time_start time_end are the start time and the end time of the time interval in which you apply base excitation. You need to specify these times in units that you actually use in your Tochnog calculation (so not in the units of the SMC file).

If this bounda_baseline_correction is not specified the data will be used directly without a correction.

See also bounda_baseline_correction_parameters.

6.20 bounda_baseline_correction_parameters index ...

The parameters for the parabolic baseline correction are written in this record. In future calculations you can use the parameters yourself by setting this record in the input file; then the parameters will not be determined again by the baseline correction algorithm; the parameters in the specified record will be used instead.

6.21 bounda_constant index switch

This record can be used i.s.o. the bounda_time record. If switch is set to -yes the prescribed dofs kept constant. This is only available for velocities, pressures and temperatures. This is not available for time derivatives tttemp, tpres and tttotal_pres.
6.22  **bounda_dof index node_range dof_0 dof_1 . . .**

States which dof’s in which nodes get prescribed values by adjustment of the **node_dof** records. The item **node_range** represents a range of node numbers. In stead of a node range also, by example, `-geometry_line 1` can be used, indicating that the nodes on line 1 get the prescribed boundary values. The items `dof_0` etc. are one of the primary dof’s listed at **dof_label**.

For a specific **index**, only one of **bounda_force** or **bounda_dof** can be specified (thus either Neumann conditions or Dirichlet conditions).

Example for discrete node forces in y-direction on the nodes on a line:

```
bounda_dof 0 -geometry_line 1 -vely
bounda_time 0 0. 0. 1. 1. 100. 1.
```

Normally you only should specify boundary conditions on principal dof’s (like velocity, temperature, etc.) and not on strain, stresses, etc.!

Specially for velocity (displacement) dof’s, you can prescribe that nodes should not move in a direction normal to a plane. For this, specify `-veln` for `dof_0` to indicate that the normal velocity to a plane is 0. The normal direction should be given with **bounda_normal**; if however a geometrical entity is used to specify the nodes, you do not necessarily need to specify the **bounda_normal**, thus the normal from the geometrical entity is then used instead. The **bounda_time** record should not be specified (it is irrelevant). Internally in Tochnog a multi-point-constraint will be generated to accomplish this condition of zero velocity in normal direction.

Specially for velocity (displacement) dof’s, you can prescribe a rotation around either the x-axis, y-axis or z-axis. In 1D you cannot use this record. In 2D you can only specify a rotation around the z-axis. In 3D you can specify each of the three axis. Example of an x-axis rotation of node 12 with angular velocity of 0.33 [degrees per unit time]:

```
bounda_dof 0 12 -rotation_x_axis
bounda_time 0 0.33
```

For the rotation 0.33 the rotation vector points in the positive x-axis direction.

Specially for the groundflow phreatic head `h`, you can prescribe the physical pore pressure `-total_pressure` and Tochnog will automatically calculate the corresponding hydraulic pressure head `h`. Also specially for the groundflow phreatic head `h`, you can prescribe the time rate of the physical pore pressure `-total_pressure` and Tochnog will automatically calculate the corresponding hydraulic pressure head `h`. Also specially for the groundflow phreatic head `h`, you can prescribe the time rate of the hydraulic pressure head `-tpres`. Specially for the temperature you can prescribe the time rate of the temperature `-temp`.

As a special option you can specify also, for example, `-element_group 1` in stead of a node range. Then nodes of elements which have **element_group** set to 1 will get the prescribed boundary values.

As a special option you can specify also, for example, `-element_geometry 1` in stead of a node range. Then nodes of elements which have **element_geometry** set to 1 will get the prescribed boundary values.
As a special option you can specify also, for example, `geometry_set 1` in stead of a node range. Then nodes of elements which have any of the elements belonging to `geometry_set 1` will get the prescribed boundary values.

Notice: if several `bounda_dof` records act on a node, only the record with the highest index will be used.

See also: `bounda_time`, `bounda_sine`, `bounda_constant`, `bounda_dof_radial`, `bounda_dof_cylindrical`, `force_edge` and `force_volume`.

### 6.23 bounda_dof_cylindrical

Index $x_{first}$ $y_{first}$ $z_{first}$ $x_{second}$ $y_{second}$ $z_{second}

Specially for velocity (displacement) dof’s, you can prescribe velocities cylindrical to a line specified with the point $x_{first}$, $y_{first}$, $z_{first}$ and $x_{second}$, $y_{second}$, $z_{second}$; in 1D only $x$ values should be specified, and in 2D only $x$, $y$ values should be specified. Example:

```
bounda_dof 10 -ra ...-ra -velx -vely -velz
bounda_dof_cylindrical 10 1.23 3.43 5.12 1.23 3.43 15.12
bounda_time 10 0. 0. 1. 1. 100. 1.
```

The velocity increases linearly in size away from the specified line (at unit distance away from the line the velocity has size 1; you can scale it by the `bounda_time` record).

### 6.24 bounda_dof_radial

Index $x$ $y$ $z$

Specially for velocity (displacement) dof’s, you can prescribe velocities radial to a specified point $x$, $y$, $z$; in 1D only $x$ should be specified, and in 2D only $x$, $y$ should be specified. Example:

```
bounda_dof 10 -ra -ldots -ra -velx -vely -velz
bounda_dof_radial 10 1.23 3.43 5.12
bounda_time 10 0. 0. 1. 1. 100. 1.
```

A radial velocity is prescribed on nodes in a specified range, relative to point 1.23, 3.43, 5.12 and with the time table given by `bounda_time`. The velocity increases linearly in size away from the specified point $x$, $y$, $z$ (at unit distance away from the specified point $x$, $y$, $z$ the velocity has size 1; you can scale it by the `bounda_time` record).

### 6.25 bounda_factor

Index $a_0$ $a_1$ ... $a_n$

This data item defines a polynomial in space. This polynomial gives a factor which is used as a multiplication factor for `bounda_time` records (with the same index). In this way, you can obtain coordinate dependent boundary conditions.

In 1D the polynomial is $a_0 + a_1 x$ (specify 2 values). In 2D the polynomial is $a_0 + a_1 x + a_2 y$ (specify 3 values). In 3D the polynomial is $a_0 + a_1 x + a_2 y + a_3 z$ (specify 4 values).
6.26 **bounda_factor_parabolic_x** *index* $a_0$ $a_1$ $a_2$

This data item defines a quadratic polynomial in x-direction.

The polynomial is $a_0 + a_1x + a_2x^2$ (specify 3 values).

6.27 **bounda_force** *index* node_range dof_0 dof_1 ... 

States which ones from the list of dof’s in which nodes get prescribed nodal forces. The item *node_range* represents a range of node numbers. In stead of a node range also, for example, `-geometry_line 1` can be used, indicating that the nodes on line 1 get the prescribed nodal forces. The items *dof_0* etc. can be one of the items listed at *dof_label*. However, neither `-dis` and `-scal` can be used.

For a specific *index*, only one of **bounda_force** and **bounda_dof** can be specified; thus, either Neumann conditions or Dirichlet conditions can be applied to a particular node, but nor both.

**Attention**: with this option you get the same nodal force on all the specified nodes. If you want to apply a distributed force on a edge, however, you should use **force_edge**. That option gives forces consistent with the displacement field, so not necessarily the same for all nodes. For example the nodes on the side of linear elements on a edge get only half the force.

As a special option you can specify also, for example, **element_geometry 1** in stead of a node range. Then nodes of elements which have **element_group** set to 1 will get the prescribed nodal forces.

Notice: if several **bounda_force** records act on a node, the imposed forces are summed.

See also: **bounda_time**, **bounda_sine** and **force_edge**.

6.28 **bounda_found** *index* found

This record is meant for printing only. A value of `-yes` indicates that the corresponding **bounda_*** records are indeed used at some nodes. A value of `-no` indicates that the corresponding **bounda_*** records are not used at all at some nodes.

6.29 **bounda_geometry_method** *index* node_type

If boundary conditions are imposed on a geometry, you can set with this record which node type should be used. If *node_type* is set to `-node_start_refined` the values of `-node_start_refined` are used to determine if nodes are located on the geometry. If *node_type* is set to `-node` the values of `-node` are used to determine if nodes are located on the geometry. If *node_type* is set to `-plus_displacement` the values of `-node` plus nodal displacements are used to determine if nodes are located on the geometry.

6.30 **bounda_normal** *index* normal_x normal_y normal_z

This record specifies the components of a normal vector to a plane on which nodes should slide (the nodes are not allowed to move normal to the plane). In 3D you need to specify all of normal_x normal_y normal_z. In 2D you need to specify only normal_x normal_y. In 1D you need to specify only normal_x.
See also `bounda_dof`.

6.31 `bounda_print_mesh_dof` `dof_0` `dof_1` ...

See `print_mesh_dof`.

6.32 `bounda_print_mesh_dof_geometry` `geometry_item_name` `geometry_item_index`

See `print_mesh_dof`.

6.33 `bounda_print_mesh_dof_values` `value_dof_0` `value_dof_1` ...

See `print_mesh_dof`.

6.34 `bounda_sine` `index` `start_time` `end_time` `freq_0` `amp_0` `freq_1` `amp_1` ...

The `bounda_dof` or `bounda_force` record with the same `index` is imposed with the sum of the sine functions; the first sine function has frequency `freq_0` and amplitude `amp_0`, the second sine function has frequency `freq_1` and amplitude `amp_1`, etc. More general behavior in time can be imposed by using `bounda_time` records. For a specific `index` only one of `bounda_time` and `bounda_sine` can be specified.

As a typical application the response due to the excitation with a frequency spectrum can be analyzed. Just print the relevant response by `control_print_history` and extract the frequency spectrum of that response signal.

The sine loads will be only imposed after `start_time`, and will not be imposed anymore after `end_time`. The sine functions start at time `start_time` (then they have value 0).

As a special option setting a frequency to 0 enforces tochnog to use a constant static value of the specified amplitude.

6.35 `bounda_time` `index` `time` `load` `time` `load` ...

This record specifies a multi linear time-load diagram for the `bounda_dof` or `bounda_force` record with the same `index`. Between two time points in the diagram, the load is interpolated linearly (ramp function between the two points).

At all times that an dof is not prescribed in such way, it is free and determined with the governing differential equations. For a specific `index` only one of `bounda_time`, `bounda_sine` and `bounda_time_user` can be specified.

As a special option, you can specify only one value in the `bounda_time` record if the load is constant over time (so not time-load sets but directly the constant load value).

As a further special option, you can specify no `bounda_time` and no `bounda_sine` at all; then a 0 value is assumed.
6.36  **bounda_time_factor**  index  factor

With this record you can specify an multiplication factor to be used for loads specified by `bounda_time`. This option comes handy when you import a time-load table from some external data source, which uses some other definition of the load as you do in the tochnog input file. For example, if you specify accelerations in metric units but the external source specifies the accelerations as part of the gravity acceleration, you can convert the load in the time-load table with this factor.

Default, if `bounda_time_factor` is not specified, the factor is set to 1.

6.37  **bounda_time_offset**  index  time_offset

With this record you can specify an offset to be used for times specified by `bounda_time`. The actual times will become time offset added to the specified times in `bounda_time`. This option comes handy when you import a time-load table from some external data source, but would like to apply the table at a different moment in time in the calculation. You need to specify `time_offset` in the units that you actually use in your calculation.

6.38  **bounda_time_increment**  index  time_increment

With this record you can specify that the data as specified in `bounda_time` is only the load data, so not time points anymore. The time points are automatically calculated from a fixed time increments (and optionally an initial offset as specified in `bounda_time_offset`). For example:

```plaintext
...  
bounda_dof 10 -geometry_line -accx  
bounda_time 10 0.2 0.78 1.33 ... (acceleration data only)  
bounda_time_offset 10 1. (the accelerations start at time 1)  
bounda_time_increment 10 0.05 (the increments in time are 0.05)  
(thus the acceleration is spcified at times 1.0, 1.05, 1.10, 1.15, etc.)  
...  
```

In this example the acceleration is 0.2 at time 1, it is 0.78 at time 1.05, etc.

6.39  **bounda_time_units**  factor_time  factor_length

The specified times and data in `bounda_time` may have other units then you actually apply in your calculation. With `factor_time` you correct the time in `bounda_time` to get times consistent with your calculation. With `factor_length` you can correct the data in `bounda_time` to get data consistent with your calculation. For example, if `bounda_time` contains [sec] and [cm] and if your actual calculation uses [hour] and [m] then set `factor_time` to 3600, and set `factor_length` to 100. This option is presently only available for prescribed accelerations.

6.40  **bounda_time_smc**  index  switch

If `switch` is set to `-yes` the SMC file `index.smc` will be read. Such Strong Motion CD file (SMC file) contains base acceleration time data. This option can be used to read SMC files strictly following
the definition from http://nsmp.wr.usgs.gov/smcfmt.html. A typical input example for a SMC file looks like:

```
... materi_velocity materi_stress ...
end_initia ...
bounda_baseline_correction 1. 1.1 (correct acceleration for time 1 to 1.1 ...
... bounda_dof 10 -geometry_line -accx
bounda_time_smc 10 -yes
bounda_time_smc_offset 10 1. (the base excitation starts at time 1)
bounda_time_smc_units 10 3600. 100. (we use hours and meters)
... control_timestep 10 1.e-2 1. (gravity from time 0 to 1)
... control_timestep 20 1.e-6 0.1 (base excitation from time 1 to 1.1)
... 
```

In case the SMC file does not strictly follow the definition from http://nsmp.wr.usgs.gov/smcfmt.html, the option bounda_time_smc cannot be used. In such case you can use the actual data lines in a bounda_time record as follows:

```
... materi_velocity materi_stress ...
end_initia ...
bounda_baseline_correction 1. 1.1 (correct acceleration for time 1 to 1.1 ...
... bounda_dof 10 -geometry_line -accx
include acceleration.dat (include file containing bounda_time 10 ..., the dots ...
represents acceleration data)
bounda_time_offset 10 1. (the base excitation starts at time 1)
bounda_time_units 10 3600. 100. (we use hours and meters)
... control_timestep 10 1.e-2 1. (gravity from time 0 to 1)
... control_timestep 20 1.e-6 0.1 (base excitation from time 1 to 1.1)
... 
```

Be sure that you take sufficient small time increments while performing the base acceleration steps. See also http://nsmp.wr.usgs.gov/.
6.41  **bounda_time_smc_offset**  \textit{index time_offset}

The times of the SMC file are incremented with \textit{time_offset}, such that you can use the acceleration data starting from any time point in a calculation. If this record is not specified then \textit{time_offset} is set to 0.

6.42  **bounda_time_smc_units**  \textit{factor_time factor_length}

The SMC files have units \([\text{cm}]\) for length and \([\text{sec}]\) for time. You input file may have other units however. With \textit{factor_time} you correct the time read from the SMC file to get times consistent with your input file. With \textit{factor_length} you can correct the data (acceleration, velocity or displacement) read from the SMC file to get data consistent with your input file. For example, if you use \([\text{hour}]\) and \([\text{m}]\) in your calculation then set \textit{factor_time} to 3600. and set \textit{factor_length} to 100.

6.43  **bounda_time_user**  \textit{index switch}

If \textit{switch} is set to \texttt{-yes} a user supplied routine for the time-load diagram will be used.

See also the file \texttt{user.cpp} in the distribution.

6.44  **bounda_water**  \textit{index switch}

If \textit{switch} is set to \texttt{-yes}, and you specify the pore pressure \texttt{-total_pressure} as dof, the pore pressure is actually determined from the height of the water column between the node and the phreatic level. In fact the pore pressure is set to \(\text{density}\_\text{water} \ g \ \Delta z\) where \(g\) is the gravitational acceleration, and \(\Delta z\) is the distance to the phreatic level.

The water density is given by \texttt{groundflow\_density}. The gravity acceleration is given by the vertical component of \texttt{force\_gravity}. The water height is relative to the water height is given by \texttt{groundflow\_phreatic\_level}.

In this case the record \texttt{bounda\_time} does not contain the actual value of the pore pressure, but instead it only contains a multiplication factor for the static water pressure as calculated above.

This \texttt{bounda\_water} is convenient when the phreatic level is located above the FE mesh. Then this option allows you to impose a pressure boundary condition for the nodes in the FE mesh at the top boundary of the mesh, automatically using a specified phreatic level record.

6.45  **change\_dataitem**  \textit{index data\_item\_name data\_item\_index data\_item\_number_0 data\_item\_number_1 \ldots operat}

With this record you can specify a data item which should be changed over time. The time table should be given in the \texttt{change\_dataitem\_time} table as time-value sets; at least two sets should be specified.

The \texttt{operat} determines how the time-value sets are used. If \texttt{operat} is set to \texttt{-use}, then the value of the time-value sets is directly used. If \texttt{operat} is set to \texttt{-add}, then the value of the time-value sets is interpreted as a rate of change, so that the value is multiplied with the time step and then added to the old value.

Notice that you can change multiple numbers at once.
As a typical example you can use this to prescribe the displacement of a contact geometry over time. Below the y-coordinates of a geometry line which is used in the contact algorithm is changed over time:

```
contact_target_geometry 0 -geometry_line 1
... geometry_line 1 0. 10. 2. 10.
... change_dataitem 0 -geometry_line 1 1 -use
change_dataitem_time 0 0. 10. 100. 0.
change_dataitem 1 -geometry_line 1 3 -use
change_dataitem_time 1 0. 10. 100. 0.
...```

The specified values are evaluated at the start of each timestep.

### 6.46 change_dataitem_geometry

```
index geometry_entity_name geometry_entity_index
```

For element group data `group_*` you can restrict the application for the `change_dataitem` to only those elements which are part of the geometry specified by `geometry_entity_name geometry_entity_index`.

### 6.47 change_dataitem_time

```
index time value ...
```

See `change_dataitem` and `change_dataitem_time_user`.

### 6.48 change_dataitem_time_discrete

```
index switch
```

If `switch` is set to `-yes` then the changes applied by the `change_dataitem` and `change_dataitem_time` records (with the same index), will be applied at the discrete time points given in `change_dataitem_time`. Between those time points, no interpolation is used.

More precise, the change of the data item will be applied directly after the time point has passed.

If you don’t specify this `change_dataitem_time_discrete` record then interpolation is used.

### 6.49 change_dataitem_time_method

```
index method
```

With this record you can require that the cosinus, sinus or tangent of a data value will be changed (instead of the data value directly itself). The `method` can be set to either `-cosinus`, `-sinus` or `-tangent`. This is typically convenient for geotechnical safety factor calculations where you want that for a mohr coulomb law the cohesion and tangent of the friction angle are decreased at the same ratio in time.

Example:
... 

```
group_materi_plasti_mohr_coul_direct 10 ...
```

(tangent of friction angle reduction)
```
change_dataitem 10 -group_materi_plasti_mohr_coul_direct 10 0 -use
change_dataitem_time 10 ...(specify tangent values here)
change_dataitem_time_method 10 -tangent
```

(cohesion reduction)
```
change_dataitem 20 -group_materi_plasti_mohr_coul_direct 10 1 -use
change_dataitem_time 20 ...
```

As an extra remark on such 'phi - c' reduction: if you want to calculate the safety factor of the initial geometry you should not use the `mesh -follow_material` ... option, since with that the soil would be able to deform to new stable configurations.

**6.50 change_dataitem_time_user index switch**

If `switch` is set to `-yes` a user supplied subroutine is used instead of the `change_dataitem_time` table.

See also the `user.cpp` routine included in the distribution.

**6.51 check_data switch**

If `switch` is set to `-yes` the in-core database is checked at some moments during the calculation. You can try this option in case you experience unexpected behavior.

**6.52 check_error switch**

Tochnog will does some error checking which you can suppress by setting `switch` to `-no`.

**6.53 check_element_node index switch**

Tochnog will check that elements do not have duplicate nodes. If you want to have duplicate nodes on purpose however, you can set `switch` to `-no` so that this checking is suppressed.

**6.54 check_element_shape index factor**

Isoparametric elements are mapped from the isoparametric space to the real coordinate space with shape functions. The determinant of the Jacobian of the mapping will have the same value in each integration point if elements are not distorted by the mapping. Thus the relative difference \( \frac{\det_{ip} - \det_{average}}{\det_{average}} \) in each integration point of an element measures the distortion.

Tochnog determines the average of the relative difference for all the integration points in an element.
If this average is larger then factor a warning message will be printed. Furthermore, if check_element_shape is specified the average will be stored in a record element_shape in the database dbs file; the average will be plotted in the GID post-processing files so that you can visually inspect where the elements are most distorted.

Perfectly non-distorted isoparametric elements have average 0.

Severely distorted elements have a high average, e.g. larger than 0.25.

6.55 check_memory index switch

If switch is set to -yes, Tochnog checks memory usage of the calculation. If switch is set to -no, Tochnog does not check memory usage of the calculation.

When checking memory usage Tochnog checks that the calculation fits in the computer RAM memory. Furthermore, on 32 bit systems Tochnog checks that array sizes do not exceed 2Gb.

Default, if check_memory is not specified, the switch is set to -no.

6.56 check_memory_usage index switch

If switch is set to -yes Tochnog keeps record of the highest memory used by the calculation. It will put that highest usage, expressed in GB, in the record check_memory_usage_result. This option comes convenient to keep an eye on the memory usage of a calculation, in case you are reaching the limit on your computer. You need to prevent that memory usage exceeds the amount of RAM memory, since swapping to disk is extremely slow.

This option is only available on 64 bit linux. Default, if check_memory_usage is not specified, then switch is set to -yes.

6.57 check_memory_usage_result index memory

See check_memory_usage.

6.58 check_nan switch

If switch is set to -yes some internal result (stresses, etc.) are check for being NAN. NAN represents Not A Number, meaning that the computer cannot represent the result by a number. This means that something is wrong: the solution may have diverged, or you may have a programming error in a user supplied routine, or etc.

6.59 check_solver eps

If this record is set the solver checks if diagonal terms are smaller than eps. That normally indicates some problem in your input file if eps is very small.
6.60 check_target switch

If you set switch to -no, any target_ * records will be neglected. This allows to run the input file without getting error messages in the log file, for example when testing variations of the input file.

6.61 check_used switch

If switch is set to -yes, Tochnog will check if input data is really used. It will give a message if some input data is not used. In fact, this option will only check if any index of a certain data item is used; thus if any index of a certain data item in the input file has been used in the calculation, all indices of that data item are considered to be used, and no message will be given.

This option comes handy when looking for errors in input files.

Default, when check_used is not specified, switch is set to -no.

6.62 check_warning switch

Tochnog will does some warning checking which you can suppress by setting switch to -no.

6.63 condif_convection_edge_normal index αc Tr

Convection coefficient and convection environmental temperature. Also the record condif_convection_edge should be specified.

Attention: this option is only available for linear and quadratic isoparametric elements.

6.64 condif_convection_edge_normal_element index element_0 element_1 ...

Selects the elements for which the condif_convection_edge_normal record with the same index should be applied.

6.65 condif_convection_edge_normal_element_group index element_group_0 element_group_1 ...

Selects the element groups for which the condif_convection_edge_normal record with the same index should be applied.

6.66 condif_convection_edge_normal_element_node index element node_0 node_1 ...

Selects the element and local node numbers for which the condif_convection_edge_normal record with the same index should be applied.
6.67  condif_convection_edge_normal_element_side  
index element_0 element_1 ... side

Selects the elements and local side number for which the condif_convection_edge_normal record with the same index should be applied.

6.68  condif_convection_edge_normal_geometry  
index geometry_entity_name geometry_entity_index

Selects the area for which the condif_convection_edge_normal record with the same index should be applied.

Instead of a number of nodes also, for example, -geometry_line 1 can be used in 2D, indicating that the nodes on line 1 start to convect. The total edge of an element must be inside the geometry for the force to become active. For 2D elements the border lines are edges. For 3D elements the border surfaces are edges. See also: condif_convection_edge_normal.

6.69  condif_convection_edge_normal_node  
index node_0 node_1 ...

Selects the nodes for which the condif_convection_edge_normal record with the same index should be applied. The node_0 etc. specifies the global node numbers.

6.70  condif_heat_edge_normal  
index heat

Distributed prescribed heat flux normal normal to the edge of a element. This distributed heat is translated into equivalent nodal heat on the edges of elements. Also the record condif_heat_edge_normal should be specified, and optionally the record condif_heat_edge_normal_time can be specified.

Attention: this option is only available for linear and quadratic isoparametric elements.

Attention: if this option is used INSIDE a FE mesh, then the elements on each side will get the distributed heat. So the total heat flux will normally become zero since the normals of the elements at the side of the geometry are opposite.

6.71  condif_heat_edge_normal_element  
index element_0 element_1 ...

Restricts the elements to which the condif_heat_edge_normal record with the same index should be applied.
6.72 condif_heat_edge_normal_element_group index element_group_0
element_group_1 ... 

Restricts the element groups to which the condif_heat_edge_normal record with the same index should be applied.

6.73 condif_heat_edge_normal_element_node index element node_0 node_1
...

Selects the element and local node numbers for which the condif_heat_edge_normal record with the same index should be applied.

6.74 condif_heat_edge_normal_element_node_factor index factor_0 factor_1
...

Nodal multiplication factors with which the condif_heat_edge_normal will be applied to the element of condif_heat_edge_normal_element_node. You need to specify a factor for each node on the side. Here factor_0 is the multiplication factor for the first node on the side, etc.

6.75 condif_heat_edge_normal_element_side index element_0 element_1
... side

Selects the elements and local side number for which the condif_heat_edge_normal record with the same index should be applied.

6.76 condif_heat_edge_normal_factor index a_0 a_1 ... a_n

This data item defines a polynomial in space. This polynomial gives a factor which is used as a multiplication factor for condif_heat_edge_normal records (with the same index). In this way, you can obtain coordinate dependent heat fluxes.

In 1D the polynomial is \( a_0 + a_1 x \) (specify 2 values). In 2D the polynomial is \( a_0 + a_1 x + a_2 y \) (specify 3 values). In 3D the polynomial is \( a_0 + a_1 x + a_2 y + a_3 z \) (specify 4 values).

6.77 condif_heat_edge_normal_geometry index geometry_entity_name geometry_entity_index

Selects the area for which the condif_heat_edge_normal record with the same index should be applied. For example, -geometry_line 1 can be used in 2D, indicating that the nodes on line 1 get the distributed heat. The total edge of an element must be inside the geometry for the force to become active. For 2D elements the border lines are edges. For 3D elements the border surfaces are edges.

6.78 condif_heat_edge_normal_node index node_0 node_1 node_2 ...

Selects the nodes for which the condif_heat_edge_normal record with the same index should be applied. The node_0 etc. specify global node numbers.
6.79  condif_heat_edge_normal_sine index start_time end_time freq_0 amp_0
defreq_1 amp_1 ...  

Similar to force_edge_sine, now for heat flux however.

6.80  condif_heat_edge_normal_time index time load time load ...  

This record specifies a diagram which contains the factors with which the condif_heat_edge_normal record with the same index is applied. Linear interpolation is used to extend the time load values to the intervals between these pairs. Outside the specified time range a factor 0 is used. If this record is not specified, the heat flux is applied at all times with a factor 1.

6.81  condif_heat_volume index heat  

Distributed volume heat source. Here heat is the distributed heat source value. See also condif_heat_volume_factor, condif_heat_volume_geometry, and condif_heat_volume_time.

6.82  condif_heat_volume_element index element_0 element_1 ...  

Specifies the elements for which the condif_heat_volume record with the same index should be applied.

6.83  condif_heat_volume_element_group index element_group  

Specifies the element group for which the condif_heat_volume record with the same index should be applied.

6.84  condif_heat_volume_factor index a_0 a_1 ... a_n  

This polynomial gives a factor which is used as a multiplication factor for condif_heat_volume records (with the same index). In this way, you can obtain coordinate dependent forces.

In 1D the polynomial is a_0 + a_1 x (specify 2 values). In 2D the polynomial is a_0 + a_1 x + a_2 y (specify 3 values). In 3D the polynomial is a_0 + a_1 x + a_2 y + a_3 z (specify 4 values).

6.85  condif_heat_volume_geometry index geometry_name geometry_index  

Specifies the geometry for which the condif_heat_volume record with the same index should be applied.

6.86  condif_heat_volume_sine index start_time end_time freq_0 amp_0
defreq_1 amp_1 ...  

Similar to force_edge_sine, now for volume heat source however.
6.87 **condif_heat_volume_time** *index time load time load ...*

This record specifies a multi-linear diagram which contains the factors with which the **condif_heat_volume** record with the same index is applied.

If this record is not specified, the heat source is applied at all times with a factor 1.

6.88 **condif_heat_volume_user** *index switch*

Set `switch` to `-yes` if you want to call the user supplied routine for heat.

6.89 **condif_heat_volume_user_parameters** *index ...*

Specify the parameters for the user supplied routine for heat.

6.90 **condif_radiation_edge_normal** *index \(\alpha_r\), \(T_r\)*

Radiation coefficient and radiation environmental temperature. Also the record **condif_radiation_edge_normal_geometry** should be specified.

**Attention**: this option is only available for linear and quadratic isoparametric elements.

6.91 **condif_radiation_edge_normal_element** *index element_0 element_1 ...

Selects the elements for which the **condif_radiation_normal_edge** record with the same *index* should be applied.

6.92 **condif_radiation_edge_normal_element_node** *index element node_0 node_1 ...

Selects the element and local node numbers for which the **condif_radiation_edge_normal** record with the same *index* should be applied.

6.93 **condif_radiation_edge_normal_element_group** *index element_group_0 element_group_1 ...

Selects the element groups for which the **condif_radiation_normal_edge** record with the same *index* should be applied.

6.94 **condif_radiation_edge_normal_element_side** *index element_0 element_1 ... side*

Selects the elements and side number for which the **condif_radiation_edge_normal** record with the same *index* should be applied.
6.95  \textit{condif\_radiation\_edge\_normal\_geometry index geometry\_entity\_name geometry\_entity\_index}

Selects the area for which the \textit{condif\_radiation\_edge\_normal} record with the same \textit{index} should be applied.

\begin{center}
\includegraphics[width=0.5\textwidth]{condif_radiation_edge_normal_geometry.png}
\end{center}

In stead of a number of nodes also, for example, \textit{-geometry\_line 1} can be used in 2D, indicating that the nodes on line 1 radiate heat. The total edge of an element must be inside the geometry for the force to become active. For 2D elements the border lines are edges. For 3D elements the border surfaces are edges. See also: \textit{condif\_radiation\_edge\_normal}.

6.96  \textit{condif\_radiation\_edge\_normal\_node index node\_0 node\_1 \ldots}

Selects the nodes for which the \textit{condif\_radiation\_edge\_normal} record with the same \textit{index} should be applied. This is only available for linear elements. The \textit{node\_0} etc. specifies the global node numbers.

6.97  \textit{contact\_apply index switch}

If \textit{switch} is set to \textit{-yes}, the contact algorithm is used. If \textit{switch} is set to \textit{-no}, the contact algorithm is not used. This is done for all timestep records.

See also \textit{control\_contact\_apply}.

6.98  \textit{contact\_heat\_generation factor}

This \textit{factor} specifies how much of the frictional energy is transformed into heat (this only makes sense if \textit{friction} in \textit{contact\_plasti\_friction} is not zero, and if \textit{condif\_temperature} is initialized). The \textit{factor} should be between 0 and 1. See also \textit{contact\_target\_geometry}.

6.99  \textit{contact\_penalty\_pressure pressure\_penalty}

The \textit{pressure\_penalty} should be given some high value if the pressure is freely linked at the surfaces of contactor and target. See also \textit{contact\_target\_geometry}.

6.100  \textit{contact\_penalty\_temperature temperature\_penalty}

The \textit{temperature\_penalty} should be given some high value if free heat exchange between contactor and target is possible. See also \textit{contact\_target\_geometry}. 

127
6.101 contact_penalty_velocity velocity_penalty

The velocity_penalty essentially puts a spring between the contactor and the target if penetration occurs. Iterations (see control_timestep_iterations) are needed; more iterations are needed if the penalty factor is higher. See also contact_target_geometry.

6.102 contact_plasti_friction friction

See contact_target_geometry.

6.103 contact_target_element_group element_group_0 element_group_1 ...

This record defines the element groups for which the elements function as target in a contact analysis. It is advised to use different element groups for the contacting and target elements, so that the contact algorithm can distinguish between both. The target element group should consist of more than 1 layer of elements in contact direction (so only one layer of target elements is not allowed). The contacter should be smaller of size than the target.

See also contact_target_geometry.

6.104 contact_target_geometry index geometry_entity_item geometry_entity_index

Attention: the contact algorithm is experimental up to now, and may not work for all calculations.

This record specifies a contact geometry. Contacting nodes are forced to stay at the outward normal side of the contact geometry.

The allowed geometries and their material outward normals are listed below

- If a geometry_point is used in 1D, the normal is in positive x-direction.
- If a geometry_line is used in 2D, the normal is the outer product of 3-direction and the line direction (from point 0 to point 1).
- If a geometry_circle is used in 2D, the normal is the outward direction at the circle.
- If a geometry_circle is used in 3D, the normal is the outward direction on the circle surface.
- If a geometry_ellipse is used in 2D, the normal is the outward direction at the ellipse.
- If a geometry_sphere is used in 3D, the normal is the outward direction at the sphere.
- If a geometry_polynomial is used in 2D, the normal is in positive y-direction.
- If a geometry_polynomial is used in 3D, the normal is in positive z-direction.
- If a geometry_triangle is used in 3D, the normal is in direction of the outer product v01 * v02 where v01 is the vector from node 0 to node 1 and v02 is the vector from node 0 to node 2.
- If a geometry_quadrilateral is used in 3D, the normal is in direction of the outer product v01 * v02 where v1 is the vector from node 0 to node 1 and v02 is the vector from node 0 to node 2. Only non-distorted quadrilaterals should be used.
This normal can be switched sign by setting the `contact_target_geometry_switch` with the same `index` to `-yes`.

In stead of geometries, also contact with target elements will be checked. Only contact with the elements `-bar2`, `-quad4`, and `-hex8` can be detected. Specify `contact_target_element_group` for this.

The time steps should be such small, that contacting nodes penetrate the other elements only in small steps.

If a `contact_target_geometry` is used, then the contacting node should also be within the tolerance of the geometrical entity to be noticed!

If contact is detected, normal contact forces of size `contact_penalty_velocity` * penetration are generated between the contacting node and the other element. Moreover, also a frictional force of size `friction` * normal force is generated (see `contact_plasti_friction`).

With contact you need more iterations the normal, say 5 or more. See `control_timestep_iterations` how to define the number of iterations.

6.105 `contact_target_geometry_switch` index switch

See `contact_target_geometry`.

6.106 `control_bounda_relax` index switch

With this `control_bounda_relax` you can require Tochnog to store the nodal right-hand-sides; for example external nodal forces for nodes with prescribed velocities. These stored nodal right-hand-sides can later be used to relax prescribed boundary conditions; for example a prescribed velocity is removed and substituted by the stored external right-hand-side (external force) and slowly set to zero by multiplication with a time function as specified with `bounda_force` in combination with `bounda_time`. With the `control_bounda_relax_geometry` record with the same index you can select a specific geometry for which the storing will be done.

A typical example can be found in the `relax1.dat` file in your distribution.

6.107 `control_bounda_relax_geometry` index geometry_item_name geometry_item_index

See `control_bounda_relax`.

6.108 `control_check_data` index switch

If `switch` is set to `-yes` the in-core database is checked at some moments during the calculation, for the specified control index. You can try this option in case you experience unexpected behavior.

6.109 `control_contact_apply` index switch

If `switch` is set to `-yes`, the contact algorithm is used. If `switch` is set to `-no`, the contact algorithm is not used. This is done for timestep records with the same index.
Default switch is set to -yes. See also contact_apply.

6.110 control_convection_apply index switch

If switch is set to -yes, the convection of a material with respect to the mesh is allowed. If switch is set to -no, the convection of a material with respect to the mesh is not allowed. This is done for timestep records with the same index. See also convection_apply.

6.111 control_data_activate index data_item_name_0 data_item_name_1 ... switch

With this record you can set data items to become activated if switch is set to -yes or de-activated if switch is set to -no. The data_item_name specifies a data record name.

6.112 control_data_arithmetic index data_item_name data_item_index data_item_number operat

This record allows you to change a data item. With data_item_name data_item_index data_item_number you select which data item to change. It will be changed with value val as specified in the corresponding control_data_arithmetic_double record. With operat you select how to change the data item; possibilities are -plus, -minus, -multiply and -divide.

In stead of a specific index data_item_index you can also specify a range -ra ... -ra.

In case you specify -all for data_item_number the specified value will be used for all numbers of the record.

6.113 control_data_arithmetic_double index val

See control_data_arithmetic.

6.114 control_data_copy index data_item_from data_item_to

Copy data item data_item_from to data_item_to. The user is responsible to apply only logic copy actions.

Normally the data_item_from and data_item_to should have the same length. As a special option however, you can copy node_inertia to node_force records, while using a control_data_copy_factor of -1. This allows you to substitute material mass inertia by static nodal forces, for the remainder of the calculation. This in fact is the d’Alembert principle.

6.115 control_data_copy_factor index factor

Multiplication factor for control_data_copy.
6.116 **control_data_copy_index** \( index \) \( data \_item \_from \) \( index \_from \) \( data \_item \_to \) \( index \_to \)

Copy data item \( data \_item \_from \) with index \( index \_from \) to \( data \_item \_to \) with \( index \_to \). The user is responsible to apply only logic copy actions.

6.117 **control_data_copy_index_factor** \( index \) \( factor \)

Multiplication factor for **control_data_copy_index**.

6.118 **control_data_delete** \( index \) \( data \_item \_name \) \( index \_range \)

Delete one or more data items. The \( index \_range \) is a number (e.g. 3) or a range (-\( ra \) ... -\( ra \), or -\( all \)).

If \( index \_data \_item \_name \) is a nodal item (for example \( node \) or \( node \_dof \)) then \( index \_range \) can also be a geometrical entity (for example -geometry_line 1 or so), and the item will be deleted for nodes located on the geometrical entity.

If \( index \_data \_item \_name \) is an element item then \( index \_range \) can also be a geometrical entity (for example -geometry_line 1 or so), and the item will be deleted for elements with all nodes located on the geometrical entity.

In the example below element 1-10 and nodes 1-100 are deleted after some time in the calculation; this simulates dismantling a part of a structure somewhere in its lifetime. First, time steps with the total structure are taken; then a part of the structure is dismantled; then time steps with the remaining part of the structure are taken.

```plaintext
... control_timestep 10 ...
...
control_data_delete 20 -element -ra -from 1 -to 10 -ra
control_data_delete 21 -node -ra -from 1 -to 100 -ra
...
control_timestep 30 ...
...
```

If an element or node is deleted, then also the corresponding records will be deleted. See also **control_data_put**.

6.119 **control_data_put** \( index \) \( data \_item \_name \) \( index \_range \) \( number \_0 \) \( number \_1 \)... 

Puts one or more data items.

The \( index \_range \) is a number (e.g. 3) or a range (-\( ra \) ... -\( ra \), or -\( all \)). The -\( all \) option for \( index \_range \) is only available for nodal data items (like \( node \) or \( node \_dof \)). If \( data \_item \_name \) is a nodal item then \( index \_range \) can also be a geometrical entity (for example -geometry_line 1 or so), and the item will be put for nodes located on the geometrical entity. If \( data \_item \_name \) is
a element item then *index_range* can also be a geometrical entity (for example *-geometry_line* 1 or so), and the item will be put for elements with all nodes located on the geometrical entity.

With *number_0 number_1* etc. you can set which value should be put. For example only using 3 for *number_0* then you only want to set the fourth value for the data item (remember that numbering starts at 0). To specify the numbers for dof’s you can also specify names like *-velx, -sigxx*, etc. In case you specify *-all*, then all values should be given in *control_data_put_double* or *control_data_put_integer*.

The values to be put should be specified in a *control_data_put_double* record for real data or in a *control_data_put_integer* record otherwise. You should specify a value for each and every specified number.

If the data item already exists it is overwritten; else a new record will be generated.

See also *control_data_delete*.

### 6.120 control_data_put_double index ...

See *control_data_put*.

### 6.121 control_data_put_integer index ...

See *control_data_put*.

### 6.122 control_data_save index switch

If *switch* is set to *-yes* save the status of strains, stresses, displacements, etc. At a later point in the calculation you can plot with gid data relative to these saved data with *control_print_gid_save_difference*.

```plaintext
... control_timestep 10 ...
... control_data_save 20 -yes ...
... control_timestep 30 ...
... control_print_gid 40 -separate_sequential
control_print_gid_save_difference 40 -yes ...
```

### 6.123 control_dependency_apply index switch

If *switch* is set to *-yes*, dependencies as specified with *dependency_diagram* and *dependency_item* are included in the calculation. If *switch* is set to *-no*, these dependencies are not included. This is done for timestep records with the same index.

Default, if *control_dependency_apply* is not specified, then *dependency_apply* will be used.
6.124  control_distribute  

Apply a random number, based on a -lognormal or -normal distribution, to the data_item_name records. This is done for the index data_item_index and the data_item_number value in those records (0 for the first value, 1 for the second value, etc.). The data_item_index can optionally be set to -all in stead of a specific index, so that the distribution will be applied to all existing indices.

The distribution_type should be set to -lognormal or -normal. Use the control_distribute_parameters record to set the mean value and standard deviation.

The data_item_name can be one of group_* or node_. If you specify a group_, for example group_materi_elasti_young or so, then not the group item record self will be changed, but the item will be changed for the elements which use this record; in this way you can give a random distribution to group data like Young modulus, plastic properties, etc.

It is optionally possible to require a distribution that is correlated in space. To obtain such a correlated distribution, you need to specify the control_distribute_correlation_length record. If the specified correlation length is larger than 1.e12 then Tochnog uses a constant G (all components have the same value). As a special option, you can specify a different distribution length in each space direction (in 2D specify 2 values, and in 3D specify 3 values).

With control_distribute_correlation_distance you can set the maximum distance below which data will be correlated. Above that distance Tochnog will not correlate the data. Default, if control_distribute_correlation_distance is not specified it will be taken to be 4 times the correlation length.

With control_distribute_minimum_maximum you can set the minimum and maximum value which the random numbers are allowed to take. Numbers outside that range will be cutoff to the minimum or maximum value. A typical application would be limiting the void ratio to a range which is needed by a hypoplasticity law.

In the first example, an lognormal distribution with average 100 and standard deviation 1.2 is used to the nodal temperatures:

```plaintext
... materi_velocity condif_temperature ...
control_distribute 10 -lognormal -node_dof -all -temp
control_distribute_parameters 10 100. 1.2
...
```

In the second example, a normal distribution with average 1 and standard deviation 1.e-3 is used to the y coordinate of the nodes:

```plaintext
... control_distribute 10 -normal -node -all 1
control_distribute_parameters 10 1. 1.e-3
...
```
In the third example, a normal distribution with average 10 and standard deviation 1 is used to
the young’s modulus of group 7:

\[
\begin{align*}
\text{control\_distribute}\ &10\ -\text{normal}\ -\text{group\_materi\_elasti\_young}\ 7\ 0 \\
\text{control\_distribute\_parameters}\ &10\ 10.\ 1. \\
\end{align*}
\]

This control\_distribute\_* is presently only available on linux computers. The control\_distribute\_* should be before all control\_reset\_dof (thus have a higher index).

6.125 control\_distribute\_correlation\_distance index maximum\_distance

6.126 control\_distribute\_correlation\_length index correlation\_length . . .

See control\_distribute.

6.127 control\_distribute\_minimum\_maximum index minimum maximum

See control\_distribute.

6.128 control\_distribute\_parameters index mean\_value standard\_deviation

See control\_distribute.

6.129 control\_distribute\_seed index seed

For experts only. With this record you can specify the seed which will be used to start the random
series of numbers. Use a positive integer value.

As a special option you can set seed to -new then Tochnog will self choose a seed. As a special
option you can set seed to -old then Tochnog will use the previous seed.

6.130 control\_element\_group\_apply index number

See element\_group\_apply.

6.131 control\_geometry\_moving index -initialise

Initialise all geometry\_moving records. That is, determine for all elements at which time they
will be excavated by the geometry\_moving entities in the remainder of the calculation.
6.132 control_groundflow_consolidation_apply index switch

If switch is set to -no, then the material divergence part in the groundwater equation is skipped.

Attention: If you want consolidation in geotechnics then set the switch to -yes. If you do not want consolidation in geotechnics then set the switch to -no.

This is done for timestep records with the same index.

Default, if control_groundflow_consolidation_apply is not specified, then groundflow_consolidation will be used.

6.133 control_groundflow_nonsaturated_apply index switch

If switch is set to -no, then nonsaturated groundwater data (eg van Genuchten) will not be applied; only saturated data will be used.

Default, if control_groundflow_nonsaturated_apply is not specified, then groundflow_nonsaturated will be used.

6.134 control_inertia_apply index switch_0 switch_1 ...

If switch_0 is set to -yes, the corresponding inertia term is included (material mass, heat capacity, ..). The same for the other switches. A switch should be specified for each of the principal dof’s. See the 'input file - data part - introduction - types of dof’s' section for an explanation about principal dof’s. The sequence of the principal dof’s is in the order as initialised in the initia ... end_initia part.

As a special option you can specify only one switch, and then the specified value will automatically be used for all principal dof’s.

This control_inertia_apply record is applied for timestep records with the same index.

Default, if control_inertia_apply is not specified, then inertia_apply will be used.

6.135 control_input index switch

If switch is set to -yes Tochnog reads an extra piece of input from the file index.dat. The piece of input needs to be closed by two end_data statements. Comments ( ... ) are not allowed. All defines and arithmetics cannot be used.

6.136 control_interface_gap_apply index switch

If switch is set to -yes then any group_interface_gap will be applied. If switch is set to -no then any group_interface_gap will be ignored.

Default, if control_interface_gap_apply is not specified, switch is set to -yes.
6.137 **control_materi_damage_apply** *index switch*

If *switch* is set to `-no`, any damage data in the input file will be ignored. This is done for timestep records with the same index.

This option is convenient for testing your input file just linear, without the need to outcomment each and every part with damage data. See also **materi_damage_apply**.

6.138 **control_materi_dynamic** *index factor*

Same as **materi_dynamic** but now only for timesteps with the same control index.

6.139 **control_materi_elasti_k0** *index switch*

See **group_materi_elasti_k0**.

6.140 **control_materi_failure_apply** *index switch*

If *switch* is set to `-no`, any damage data in the input file will be ignored. This is done for timestep records with the same index.

This option is convenient for testing your input file just linear, without the need to outcomment each and every part with failure data. See also **materi_failure_apply**.

6.141 **control_materi_plasti_hypo_masin_ocr_apply** *index switch*

If *switch* is set to `-yes` the OCR will be applied. If *switch* is set to `-no` the OCR will not be applied. Default *switch* is `-no`.

6.142 **control_materi_plasti_hypo_masin_clay_ocr_apply** *index switch*

If *switch* is set to `-yes` the OCR will be applied. If *switch* is set to `-no` the OCR will not be applied. Default *switch* is `-no`.

6.143 **control_materi_plasti_hardsoil_gammap_initial** *index switch*

See theory section on hardsoil.

6.144 **control_materi_plasti_hypo_pressure_dependent_void_ratio** *index switch*

If *switch* is set to `-yes` the initial void ratio is corrected for pressure dependency; see the theory section. This is done for the first timestep in the corresponding **control_timestep** record with the same *index*. Default *switch* is set to `-no`.
control_materi_plasti_hypo_niemunis_viscoocr_apply \textit{index switch}

If \textit{switch} is set to -\textit{yes} the OCR will be applied. If \textit{switch} is set to -\textit{no} the OCR will not be applied.

Default \textit{switch} is -\textit{no}.

control_materi_plasti_hypo_substepping \textit{index switch}

If \textit{switch} is set to -\textit{yes} substepping will be applied in hypoplasticity routines. If \textit{switch} is set to -\textit{no} substepping will not be applied in hypoplasticity routines.

If this record is not specified the record \texttt{materi_plasti_hypo_substepping} will be used.

control_materi_plasti_tension_apply \textit{index switch}

If \textit{switch} is set to -\textit{no}, any tension-plasticity data in the input file will be ignored. This is done for timestep records with the same index.

See also \texttt{materi_plasti_tension_apply}.

control_materi_plasti_visco_apply \textit{index switch}

If \textit{switch} is set to -\textit{no}, any visco-plasticity data in the input file will be ignored. This is done for timestep records with the same index. See also \texttt{materi_plasti_visco_apply}.

control_materi_updated_apply \textit{index switch}

If \textit{switch} is set to -\textit{no}, any -\texttt{updated} material memory will be set to -\texttt{updated_linear}. If \textit{switch} is set to -\textit{yes}, any non-specified material memory will be set to -\texttt{updated}. This is done for timestep records with the same index.

control_materi_undrained_apply \textit{index switch}

See \texttt{group_materi_undrained_capacity}. Default, if \texttt{control_materi_undrained_apply} is not specified, \textit{switch} is set to -\textit{yes}.

control_materi_viscosity_apply \textit{index switch}

If \textit{switch} is set to -\textit{no}, any viscosity in the input file will be ignored. This is done for timestep records with the same index.

control_mesh_activate_gravity_apply \textit{index index_0 index_1 \ldots}

With this record you can specify which of the \texttt{mesh_activate_gravity_*} records should be applied, by specifying the indices of the records that should be applied. In case this \texttt{control_mesh_activate_gravity_apply} is not given, all \texttt{mesh_activate_gravity_*} records
will be applied. As a special option you can use -all indicating that all of the mesh_activate_gravity_* records should be applied (this is the same as not specifying the control_mesh_activate_gravity_apply record at all). As another special option you can use -none indicating that none of the mesh_activate_gravity_* records should be applied.

See also mesh_activate_gravity_time.

6.153 control_mesh_adjust_geometry index geometry_entity_item_0
   geometry_entity_index_0 geometry_entity_item_1
   geometry_entity_index_1

The nodes of the geometry entity 0 are replaced such that they neatly follow the boundary of geometry 1. In this way, it is easy to make a mesh with elements precisely in specific regions, if this is required to give separate element_group data (e.g. materials) to the geometry and it is too difficult to make the mesh at once OK for this.

The created mesh may be quite distorted.

6.154 control_mesh_change_element_group index element_group_0 element_group_1

Change the group number element_group of elements from element_group_0 to element_group_1.

6.155 control_mesh_convert index switch

If switch is set to -yes, tochnog will automatically convert elements:

- -bar2 in 2D to -quad4 if the element is an interface
- -bar3 in 2D to -quad6 if the element is an interface
- -tria3 in 3D to -prism6 if the element is an interface
- -tria6 in 3D to -prism12 if the element is an interface
- -quad4 in 3D to -hex8 if the element is an interface
- -quad8 in 2D to -quad6 if the element is an interface
- -quad8 in 3D to -hex18 if the element is an interface
- -quad9 in 2D to -quad6 if the element is an interface
- -quad9 in 3D to -hex18 if the element is an interface
- -hex20 in 3D to -hex18 if the element is an interface
- -hex20 in 3D to -hex27 if the element is not an interface
- -prism15 in 3D to -prism12 if the element is an interface
- -prism15 in 3D to -prism18 if the element is not an interface
For an interface you need to specify interface data in the `group_interface`... For example the `-bar2` is connected to two nodes, whereas the converted `-quad4` is connected to four nodes. In a similar manner all other converted elements also get extra nodes. This options makes it easy to obtain a mesh with interface elements. For example generate with GID in a 2d mesh bar elements, insert group data, and use `control_mesh_convert` to generate the interface elements. This generation of interfaces only works properly if certain conditions are satisfied:

- Each interface needs to have only isoparametric neighbours which have a total side in common with the interface. For example a `-hex8` interface should only have `-hex8` neighbours.
- Surfaces with interface elements should not intersect with another surface with interface elements.

The new generated nodes will be connected to existing neighbouring element at the interfaces. The `control_mesh_convert` tries to do that automatically correct. You can help however by specifying in the record `control_mesh_convert_element_group` element groups which are located at one side of the interfaces (for example the groups of a pile in soil when an interface is generated between pile and soil).

Example in which a `-bar2` interface becomes a `-hex8` interface:

```plaintext
... number_of_space_dimension 3 ...
... end_initia ...
... element 1 -bar2 101 102
... element_group 1 10 ...
... group_interface 10 -yes ...
... control_mesh_extrude 100 ...
... control_mesh_convert 110 -yes ...
... 
```

If `switch` is set to `-no`, tochnog will not convert elements.

### 6.156 control_mesh_convert_element_group

```plaintext
6.156 control_mesh_convert_element_group index element_group_0 element_group_1 ...
```

See `control_mesh_convert`.

### 6.157 control_mesh_convert_quad9_quad6

```plaintext
6.157 control_mesh_convert_quad9_quad6 index dir
```

Convert `quad9` into `quad6` is a 2D calculation. With `dir` you can decide in which isoparametric direction of the `quad9` nodes should be deleted (so that becomes the linear direction in the `quad6` element). Set `dir` either to `-x` or to `-y`. 
6.158  control_mesh_convert_tria6_tria3 index switch

Convert tria6 into tria3 is a 2D calculation. This is done if switch is set to -yes.

6.159  control_mesh_copy index move_x move_y move_z

This option copies the mesh. Thus you get twice as much elements and nodes. Each new node is moved move_x move_y move_z relative to the corresponding old node. In 1d you only should specify move_x. In 2d you only should specify move_x move_y.

6.160  control_mesh_cut_geometry index geometry_item_name geometry_item_index

This command cuts away a part of the mesh, as defined by geometry_item_name geometry_item_index. The cut away mesh can be substituted by its nodal forces. Actually, with control_mesh_cut_force you can set for each direction if the nodal force of the cut away mesh should be applied to the remaining mesh. This will be done if you set the corresponding switch in control_mesh_cut_force to -yes. In 2D you should set two switches, and in 3D you should set 3 switches.

See mesh_cut_1.dat and mesh_cut_2.dat in the test directory for examples. See earthquake_2.dat how this control_mesh_cut_geometry command can save you computing time in dynamic calculations with many timesteps.

6.161  control_mesh_cut_force index switch_0 switch_1 switch_2

See control_mesh_cut_geometry.

6.162  control_mesh_delete_element index number_0 number_1 . . .

The elements with numbers number_0 number_1 will be deleted. Otherwise the same as control_mesh_delete_geometry.

6.163  control_mesh_delete_geometry index geometry_entity_item geometry_entity_index

All elements which are part of the geometry item are deleted. In this way, it is easy to make a mesh with holes, tunneling systems in ground, etc. Remaining nodes in the geometry, are moved onto the edge of the geometry if the corresponding control_mesh_delete_geometry_move_node record with the same index is set to -yes; (otherwise, the remaining nodes are left inside the geometry).

For a geometry_point, elements inside the tolerance distance of the point will be deleted. For a geometry_circle, elements in the total inner area of the circle radius plus its tolerance will be deleted. Likewise for other geometries.

If you combine this record with a control_timestep record, then the element will be slowly deleted, starting from a complete element at the start of the timestep up to no element at the end of the timestep. This is accomplished by reducing the nodal forces of the elements slowly to zero; at the end of the timestep, the element is deleted completely. This might be useful for a better convergence behavior of the iterative process.
If an element is being deleted, `element_empty` is automatically set to `-empty`, even if the element is not completely deleted yet. This allows you to look with GID 'behind elements that are being deleted' (see also `element_empty` and `control_print_gid_empty`).

See also `control_mesh_delete_geometry_move_node`, `control_mesh_delete_geometry_element`, and `control_mesh_delete_geometry_element_group`.

6.164 `control_mesh_delete_geometry_direct` index switch

If `method` is set to `-yes` the elements inside the geometry will be deleted direct, as opposed to slowly over the timesteps.

6.165 `control_mesh_delete_geometry_element` index `element_name_0` element_name_0 ...

Only elements with names `element_name_0` etc. will be deleted if the `control_mesh_delete_geometry` (with the same index) is used. For example, `element_name_0` is `-quad4`, `-beam`, etc.

If this record is not specified all elements in the geometry will be deleted.

6.166 `control_mesh_delete_geometry_element_group` index `element_group_0` element_group_0 ...

Only elements from group `element_group_0` etc. will be deleted if the `control_mesh_delete_geometry` (with the same index) is used.

6.167 `control_mesh_delete_geometry_factor` index `factor_0` factor_0 factor_1 ...

The elements deleted by `control_mesh_delete_geometry` (with the same index), will be deleted by a factor `factor_0` at the start of the timesteps up to a factor `factor_1` at the end of the timesteps. If the `control_mesh_delete_geometry` is not used in combination with timesteps, then directly `factor_1` will be applied.

If `factor_1` exceeds $1.0 - 1.0 \times 10^{-10}$ an element will be completely deleted from the calculation, that is the `element` record will be removed and cannot be reactivated in any way later in the calculation.

If this record is not specified then `factor_0 = 0` and `factor_1 = 1`.

6.168 `control_mesh_delete_geometry_method` index method

Determines the condition on which an element will be considered part of the geometry to be deleted. If `method` is set to `-all` then all element nodes should be part of the geometry. If `method` is set to `-any` then any of the element nodes should be part of the geometry. If `method` is set to `-average` then the average element coordinate should be part of the geometry.

Default this record is `-all`.

See also `control_mesh_delete_geometry`.
6.169  control_mesh_delete_geometry_move_node index switch

Determines if remaining nodes inside a deleted geometry, are moved onto the edge of the geometry (-yes) or not (-no). Moving nodes makes that the element mesh exactly fits the deleted geometry, but may also lead to heavily distorted elements. Default this record is -no.

See also control_mesh_delete_geometry.

6.170  control_mesh_delete_geometry_projection_type index type

This record allows you to control what geometry will actually be deleted. Set type to -project_inside or -project_exact. For example if the geometry is a geometry_circle then -project_inside means that everything inside the circle will be deleted, whereas -project_exact means that everything within a tolerance from the circle edge will be deleted. Default type is -project_exact.

6.171  control_mesh_delete_geometry_stop index switch

If switch is set to -yes, any deleting of elements in geometries will be stopped. That is, all remaining delete factors from control_mesh_delete_geometry_factor will be destroyed and all elements will become fully active again.

In combination with global_element_dof_apply -yes, the elements which become active again will take their strains, stresses etc. of the moment just before being deleted! If you want to lower the stresses or strains or so, then consider using control_reset_dof.

In combination with global_element_dof_apply -no, the elements which become active again will take their strains, stresses etc. from the nodes.

6.172  control_mesh_delete_geometry_stop_geometry index geometry_entity_name

Only do the control_mesh_delete_geometry_stop for elements part of the geometrical entity specified in this control_mesh_delete_geometry_stop_geometry.

6.173  control_mesh_delete_small index eps

At the end of a timestep, an element will be deleted when its volume has become smaller than eps.

6.174  control_mesh_duplicate_element_group index element_group_old element_group_new

Use this command to duplicate elements from group element_group_old to new elements with group element_group_new. The new elements get the same nodes as the old original elements.

6.175  control_mesh_element_group_apply index group_0 group_1 ...

If you specify this record, only the element groups specified will be evaluated in the timesteps with the same index. Default, if control_mesh_element_group_apply is not specified, all
elements groups will be used.

6.176 \texttt{control_mesh_extrude} index z0 z1 z2 \ldots

Option to extrude a 2D mesh to 3D. The 2D mesh has $x,y,z$ coordinates, with $z=0$. The 3D mesh will have $x,y,z$ coordinates. You need to specify in the initialisation part \texttt{number_of_space_dimensions} to 3.

With $z0, z1, z2$ etc. you specify the coordinates of the layers to which the 2D coordinates will be extruded. With $n0, n1, n2$ you specify the number of elements that will be generated in each layer; $n0$ specifies the number of elements between $z0$ and $z1$, $n1$ specifies the number of elements between $z1$ and $z2$, etc.; for the last $n$-value you always should use a 1 (this is a dummy value, that is not used for any layer at all).

Extrusion must be done before doing mesh refinements, mesh splitting, etc.

6.177 \texttt{control_mesh_extrude_direction} index dir

Default extrusion is done in the global $z$-direction. Optionally you can set $dir$ to \texttt{y} and then extrusion is done in global $y$-direction.

6.178 \texttt{control_mesh_extrude_element} index name

If you extrude \texttt{-tria6} elements, you can set \texttt{name} either to \texttt{-prism12} or \texttt{-prism18}. Then either the 12 node or 18 node prismatic elements will be generated. Default, if this \texttt{control_mesh_extrude_element} is not set, then \texttt{-prism18} is used for \texttt{name}.

See also \texttt{control_mesh_extrude_n}.

6.179 \texttt{control_mesh_extrude_contact_spring_element_group} index element_group_0 element_group_1 \ldots

See \texttt{control_mesh_extrude_contact_spring_element_group_new}.

6.180 \texttt{control_mesh_extrude_contact_spring_element_group_new} index element_group_new_0 element_group_new_1 \ldots

If this record is specified, then a contact spring is generated between each start node and end node in the extrude direction. This option comes handy, when you want to use these contact springs to enforce that the nodes on the start plane get the same displacements as the nodes on the end plane, which models that the extruded mesh is in fact part of a very long domain with no variations in the longitudinal direction of the domain.

The contact springs get group number $element\_group\_new\_0$ when its node is attached to an element with old group $element\_group\_0$. The contact springs get group number $element\_group\_new\_1$ when its node is attached to an element with old group $element\_group\_1$. Etc. The old groups are specified in the \texttt{control_mesh_extrude_contact_spring_element_group} record. If the contact spring’s node is attached to more than one old group, the first specified old group, and corresponding new group, will be used.
As a special option, if you specify in `control_mesh_extrude_contact_spring_element_group_new` only one new element group number, then all contact springs will be placed on that group.

6.181 `control_mesh_extrude_element_group_new` index `element_group_old_0` `element_group_old_1` ... `element_group_new_00` `element_group_new_01` ... `element_group_new_10` `element_group_new_11` ...

With this option you set the element_group number of the new extruded elements.

With `element_group_old_0`, `element_group_old_1` etc. you specify the old element_group numbers of the 2D elements (which will be extruded). For these old groups, you specify for each layer in z-direction what the new element_group numbers of the extruded 3D elements should be. For example, `element_group_new_00`, `element_group_new_01` etc. give for `element_group_old_0` what the element_group numbers of the new extruded elements will be (for each z layer).

You need to specify element_group numbers for each and every z layer.

If a new element group is set to a negative number in this `control_mesh_extrude_element_group_new` the elements will not be generated.

See also `control_mesh_extrude`.

6.182 `control_mesh_extrude` index `n0` `n1` `n2` ...

See `control_mesh_extrude`.

6.183 `control_mesh_generate_beam` index `element_group` `geometry_entity_item` `geometry_entity_index`

The same as `control_mesh_generate_truss`, now for beams however.

6.184 `control_mesh_generate_contact_spring` index `element_group` `geometry_entity_item` `geometry_entity_index`

Generate `-contact_spring2` springs for nodes which have the same position in space. This can be used to connect these nodes with spring elements, so to model a contact area. Only nodes located on the specified geometry entity will be used.

The generated springs will get an `element_group` record with value `element_group`. So in that element group you can put the properties of the contact springs.

With the `control_mesh_generate_contact_spring_element` record you can set between which elements the contact_springs should be generated. For example use `-quad4` and `-truss_beam` if you want to generate contact_springs between those elements.

If `control_mesh_generate_contact_spring_element_group` (with the same index) is used, contact springs will only be generated between elements of the groups `element_group_0`, `element_group_1` etc.
6.185 `control_mesh_generate_contact_spring_element` index `element_0` `element_1`

See `control_mesh_generate_contact_spring`.

6.186 `control_mesh_generate_contact_spring_element_group` index `element_group_0` `element_group_1`

See `control_mesh_generate_contact_spring`.

6.187 `control_mesh_generate_interface` index `element_group_0` `element_group_00` `element_group_01` `element_group_1` `element_group_10` `element_group_11` ...

With this record you can generate interface elements.

The interface elements will be given an `element_group` record `element_group_0` if the interface is between `element_group_00` and `element_group_01`. The interface elements will be given an `element_group` record `element_group_1` if the interface is between `element_group_10` and `element_group_11`. The interface elements will be given an `element_group` record `element_group_2` if the interface is between `element_group_20` and `element_group_21`. Etc, etc.

The groups `element_group_00`, `element_group_10`, `element_group_20`, etc. should be on one side. The groups `element_group_01`, `element_group_11`, `element_group_21`, etc. should be on the opposite side.

Between two linear 2d elements `-quad4` interfaces will be generated. Between two quadratic 2d elements `-quad6` interfaces will be generated. Between two `-hex8` elements a `-hex8` interface will be generated. Between two `-hex27` elements a `-quad18` interface will be generated. Between two `-tet4` elements a `-prism6` interface will be generated. Between two `-tet10` elements a `-tria12` interface will be generated. Between two `-prism6` elements a `-hex8` interface will be generated on sides with 3 nodes. Between two `-prism6` elements a `-hex8` interface will be generated on sides with 4 nodes. For other situations no interface element will be generated.

Crossing interfaces are not allowed, eg in 2d you should not have locally two connecting lines with interfaces and in 3d you should not have locally two connecting surfaces with interfaces.

Interfaces can only be generated between exactly two elements. You cannot generate interface where three elements connect; for example you cannot generate an interface at the common side of two `quad4` elements if there is also a truss along that common side.

If you want the interface to connect, you really should do for example:

```
... control_mesh_generate_interface 10 20 30 31 20 40 41 ...
```

which takes care that the interfaces generated by this command are connected together. If you would have used the following:

```
... control_mesh_generate_interface 10 20 30 31
```
control_mesh_generate_interface 11 20 40 41
...

The interfaces generated by the two commands will not connect.

See also control_mesh_generate_interface_method.

6.188 control_mesh_generate_interface_method index method_select method_generate

If you set method_select to -element_geometry the control_mesh_generate_interface will select with element_geometry between which elements interfaces will be generated.

If you set method_generate to -element_geometry the control_mesh_generate_interface will generate element_geometry records for the interface elements, in stead of element_group records.

So for example using -element_geometry -element_geometry tells that the control_mesh_generate_interface in fact is index element_geometry_0 element_geometry_00 element_geometry_01 element_geometry_1 element_geometry_10 element_geometry_11 ... .

Default, if control_mesh_generate_interface_method is not specified, it is set to -element_group -element_group.

6.189 control_mesh_generate_spring1 index element_group geometry_entity_item geometry_entity_index

Generate -spring1 springs for nodes. Only nodes located on the specified geometry entity will be used.

The generated springs will get an element_group record with value element_group. So in that element group you can put the properties of the springs (see group_spring_stiffness etc.).

6.190 control_mesh_generate_spring2 index element_group geometry_entity_item geometry_entity_index

Generate -spring2 springs for nodes which have the same position in space. This can be used to connect these nodes with spring elements. Only nodes located on the specified geometry entity will be used.

The generated springs will get an element_group record with value element_group. So in that element group you can put the properties of the springs (see group_spring_stiffness etc.).

Typically you can use this option to connect meshes which were generated with different control_mesh_macro records or so.

If you need interfaces, then afterwards use a control_mesh_convert to turn the generated surface elements into real interface elements.
6.191 control_mesh_generate_truss index element_group geometry_entity_item geometry_entity_index

Generate trusses for nodes which are neighbor in space (that is, for nodes which are connected by an isoparametric finite element). Only nodes located on the specified geometry entity will be used.

The generated trusses will get an element_group record with value element_group. So in that element group you can put the properties of the trusses (see group_truss_elasti_young etc.).

Typically you can use this option to put easy trusses somewhere in a mesh with isoparametric elements.

6.192 control_mesh_generate_truss_beam index element_group geometry_entity_item geometry_entity_index

The same as control_mesh_generate_truss, now for truss_beams however.

6.193 control_mesh_generate_truss_beam_loose index switch

This record works together with the control_mesh_generate_truss, control_mesh_generate_beam and control_mesh_generate_truss_beam records.

If switch is set to -yes, the truss or beam of truss_beam will not be connected to the existing nodes, but new nodes will be generated for the generated element.

Afterwards you can typically connect the truss or beam of truss_beam to the existing mesh with constactsprings, so that the end result is that you can model frictional slip between isoparametric elements and structural elements.

See also control_mesh_generate_contact_spring.

6.194 control_mesh_generate_truss_beam_macro index macro_0 macro_1 ...

This record works together with the control_mesh_generate_truss, control_mesh_generate_beam and control_mesh_generate_truss_beam records.

With macro_0 etc. you can specify the indices of control_mesh_macro_* records. Then the trusses (or beams or truss_beams) will only be generated for nodes coming from the mesh generated by the macro records with the specified indices.

This is handy in case you generate two neighboring meshes with macro's, and want to generate the elements (trusses or beams or truss_beams) in between these two meshes. Normally, both the meshes would get the extra truss (or ..) in case you use a geometry_line or so to specify that the new elements should be generated between the two meshes (this is so, since the nodes of both meshes are located on the geometry_line). With the present control_mesh_generate_truss_beam_macro record however you can specify that the new elements should only be generated by looking at the nodes of some of the meshes, and so no double new elements will be generated in between the two meshes.
This record works together with the control_mesh_generate_truss, control_mesh_generate_truss_beam, control_mesh_generate_beam and control_mesh_generate_truss_beam records.

If switch is set to -yes, the truss or beam of truss_beam will be generated for separate regions, not necessary connected by isoparametric finite elements.

A typical example is the generation of exactly one truss between two end points (thus no trusses along all of the isoparametric elements between the end points). For this, put the end points in a geometry set, and also use -yes for this control_mesh_generate_truss_beam_separate record.

6.196 control_mesh_gid_batch index switch

If switch is set to -yes all mesh_gid_* records are used to generate elements using GiD as preprocessor. In fact, the GiD program is called from inside Tochnog as batch. This makes it convenient to do everything that you need from inside the Tochnog input file, without even opening GiD as preprocessor. A simple example is as follows:

```
echo -yes
number_of_space_dimensions 2
end_initia

( define points )
mesh_gid_point_coord 1 0. 0.
mesh_gid_point_coord 2 1. 0.
mesh_gid_point_coord 3 0. 1.
mesh_gid_point_coord 4 1. 1.

( define lines, each line connects to two points )
mesh_gid_line_point 1 1 2
mesh_gid_line_point 2 2 4
mesh_gid_line_point 3 4 3
mesh_gid_line_point 4 3 1

( define closed surfaces, each surfaces connects to multiple lines )
mesh_gid_surface_line 1 1 2 3 4
( define that the elements in this surface get a group number )
mesh_gid_surface_element_group 1 -yes

( size of elements to be generated )
mesh_gid_size 0.1

( call gid in batch, this generates an extra file index.dat so 10.dat in this example )
control_mesh_gid 10 -yes
( read the mesh in 10.dat )
control_input 10 -yes

( add extra data )
(...)

end_data
```

A more complex example which creates a circular hole in a rectangular zone is as follows:
echo -yes
number_of_space_dimensions 2
end_initia

( the circle will be used hollow )
mesh_gid_circle_coord 1 0.5 0.5
mesh_gid_circle_radius 1 0.3
mesh_gid_circle_hollow 1 -yes

mesh_gid_point_coord 1 0. 0.
mesh_gid_point_coord 2 1. 0.
mesh_gid_point_coord 3 0. 1.
mesh_gid_point_coord 4 1. 1.

mesh_gid_line_point 1 1 2
mesh_gid_line_point 2 2 4
mesh_gid_line_point 3 4 3
mesh_gid_line_point 4 3 1

( the surface edges are the line of the circle and the 4 specified lines )
mesh_gid_surface_line 1 1 2 3 4 5
mesh_gid_surface_element_group 1 1

mesh_gid_size 0.05

ccontrol_mesh_gid 10 -yes
ccontrol_input 10 -yes

target_item 0 -node 10 1
target_value 0 0.8715 1.e-2
end_data

For more examples see in your distribution test/other/mesh_gid_*_.dat.

Attention: Please realise that when generating gid objects (circle, cylinder, etc) also automatically points, lines, surfaces and volumes are generated. So you get more points, lines, surfaces and volumes then you specify yourself with mesh_gid_point_coord, mesh_gid_line_point, mesh_gid_surface_line and mesh_gid_volume_surface. Thus, you need to pay attention to which point, line, surface and volume numbers you specify in mesh_gid_line_point, mesh_gid_surface_line and mesh_gid_volume_surface. To help you, this option control_mesh_gid_batch will print for all mesh_gid_* data the points, lines, surfaces and volumes numbers that it generates. So for each thing that you specify you get the corresponding point, line and surface numbers. It is these printed numbers you should use in the records which need these numbers: mesh_gid_line_point, mesh_gid_surface_line and mesh_gid_volume_surface!

Attention: When you specify group numbers for mesh_gid_..._group records, you should apply the numbers 1, 2, 3, etc for the groups, and don’t use gaps in the numbers.

Attention: For this option to work you need to have GiD (of CIMNE) installed on your computer. Take care that the GiD path where the GiD executable is stored is set in your PATH environment symbol. Also take care that the tochnog.gid directory from your Tochnog distributions is copied to the GiD problemtypes directory.
6.197 control_mesh_interface_triangle index switch

See mesh_interface_triangle_coordinates.

6.198 control_mesh_keep_element index element_0 element_1 ...

With this option you can delete all elements except for the elements with numbers element_0, element_1, etc. This enables you to clearly view some specific elements and nodes in a plot.

6.199 control_mesh_keep_element_group index element_group_0 element_group_1 ...

With this option you can delete all elements except for the elements with group numbers element_group_0, element_group_1, etc. This enables you to clearly view some specific elements and nodes in a plot.

6.200 control_mesh_keep_geometry index geometry_item_name geometry_item_index

With this option you can delete all elements except for the elements present in the specified geometry. This enables you to clearly view some specific elements and nodes in a plot.

6.201 control_mesh_keep_node index node_0 node_1 ...

With this option you can delete all nodes except for the nodes with numbers node_0, node_1, etc. This enables you to clearly view some specific elements and nodes in a plot.

6.202 control_mesh_macro index macro_item element_group n ...

With this record and the control_mesh_macro_parameters record you define a macro region. The macro region will automatically be divided into finite elements.

The type of macro region is defined by macro_item. You can set macro_item to a -sphere (3D), -cylinder (3D), -cylinder_hollow (3D), -brick (3D), -rectangle (2D), -circle (2D), -circle_hollow (2D), -truss (1D/2D/3D), -truss_beam (1D/2D/3D) and -bar (1D).

The elements to be generated will get element_group element_group.

With n ... you define how much nodes and elements will be generated. For a -cylinder, you need to specify the number of nodes in the length direction, the number of nodes in radial direction and the number of nodes in circ. direction (there is always only one element in radial direction). For a -cylinder_hollow, you need to specify the number of nodes in the length direction, the number of nodes over the wall thickness and the number of nodes in circ. direction. For a -brick, you need to specify the number of nodes in x-direction, the number of nodes in y-direction and the number of nodes in z-direction. For a -circle and -sphere, you need to specify ‘fineness’ of the mesh, which is a number 0, 1, 2, 3, ...; a higher number gives a higher fineness; typically use 3 or so. For a -circle_hollow, you need to specify the number of nodes over the wall thickness, the number of elements in tangential direction. For a -rectangle, you need to specify the number of nodes in first direction and the number of nodes in second direction. For a -bar, you need to specify the
number of nodes. For a **truss**, you need to specify the number of nodes. For a **truss beam**, you need to specify the number of nodes.

In the following example a sphere is generated

```plaintext
... number_of_space_dimension 2 ...
... end_initia ...
control_mesh_macro 20 -sphere ...
control_mesh_macro_parameters 20 ...
... 6.203 control_mesh_macro_concentrate index ...
```

For the **rectangle** macro you can specify with this `control_mesh_macro_concentrate` record a mesh fineness concentration factor in the first direction and in the second direction. In each direction give a mesh fineness factor at the beginning and at the end (so two factors per direction). A smaller factor means smaller elements. The relative size of the factor determines where elements are concentrated, at the start or at the end.

```plaintext
6.204 control_mesh_macro_element index element_type
```

With this option you can set the element type which will be generated with `control_mesh_macro` (with the same index). This option is only available in 2d and 3d.

For **element** you can use **tria3**, **tria6**, **quad4** and **quad9** in 2d. For **element** you can use **tet4**, **tet10**, **hex8** and **hex27** in 2d.

If this record is not specified then **bar2** (1d), **quad4** (2d) or **hex8** (3d) will be generated.

Attention: in case you choose a quadratic element the macro geometry may not be exactly followed. In this case, leave the default linear elements, and use a global mesh refinement to quadratic elements afterwards, including the geometry to follow.

```plaintext
6.205 control_mesh_macro_parameters index x y ...
```

With this record you can specify the dimensions of the `control_mesh_macro` region.

For a **sphere**, you need to specify the x, y, z coordinates of the middle of the sphere and the radius of the sphere. For a **cylinder**, you need to specify the x, y, z coordinates at the start of the cylinder, the x, y, z coordinates at the end of the cylinder, the radius, the start angle and the end angle in degrees (which allows for an open section). For a **cylinder hollow**, you need to specify the x, y, z coordinates at the start of the cylinder, the x, y, z coordinates at the end of the cylinder, the middle radius, the wall thickness, the start angle and the end angle in degrees (which allows for an open section). For a **brick**, you need to specify the x, y, z coordinates at the middle, the length in x-direction, the length in y-direction, and the length in z-direction. For a **circle**, you need to specify the x, y coordinates of the middle and also the radius. For a **circle hollow**, you need to specify the same as for the `circle` and additionally the wall thickness, the start angle
and the end angle in degrees (which allows for an open section). For a \textit{-rectangle}, you need to specify the x, y coordinates of the middle, the width and the height respectively. For a \textit{-bar}, you need to specify the x coordinate of the middle and the length respectively. For a \textit{-truss}, you need to specify the coordinates of the start and of the end respectively.

\textbf{6.206 control\_mesh\_map index switch}

A typical piece of input file is

```plaintext
... global_element_dof_apply -no ...
... (input file with quadratic elements -hex20 or -hex27 or -tet10 or -prism15)
... control_mesh_map ...-yes (map to linear elements -hex8 or -prism6 or -tet4)
... control_timestep ... (calculate with linear elements)
control_solver ...-matrix_pardiso (with pardiso solver)
... control_mesh_map -yes (map back to quadratic elements)
... control_timestep ... (calculate with quadratic elements)
control_solver ...-matrix_iterative_bicg (with bicg solver)
...```

In this way, the last calculation with the quadratic elements gets as first guess for the bicg solver the solution field of the linear elements with the pardiso solver. This saves much computing time for bicg, especially in very large calculations. This strategy normally should only be used for large linear calculations. For this option always set \texttt{global\_element\_dof\_apply -no}.

\textbf{6.207 control\_mesh\_merge index switch}

If \textit{switch} is set to \texttt{-yes}, then nodes with the same coordinates are merged into one node.

\textbf{6.208 control\_mesh\_merge\_eps\_coord index epsilon}

Distance below which nodes will be merged. Default some small value.

\textbf{6.209 control\_mesh\_merge\_macro\_generate index macro_0 ...}

This record works together with the \texttt{control\_mesh\_merge} record.

With \texttt{macro_0} etc. you can specify the indices of \texttt{control\_mesh\_macro\_*} or \texttt{control\_mesh\_generate\_*} records. Then the merging will only be done for nodes coming from the mesh generated by the macro or generate records with the specified indices.
6.210 **control_mesh_merge_geometry** index geometry_entity_item geometry_entity_index

The mesh merging from **control_mesh_merge**, with the same index, will only be used for nodes in the geometry specified by `geometry_entity_item geometry_entity_index`.

6.211 **control_mesh_merge_geometry_not** index geometry_entity_item geometry_entity_index

The mesh merging from **control_mesh_merge**, with the same index, will not be used for nodes in the geometry specified by `geometry_entity_item geometry_entity_index`.

6.212 **control_mesh_mirror** index axis

This option mirrors the mesh. Thus you get twice as much elements and nodes. The `axis` should either be set to `-x`, `-y` or `-z`.

6.213 **control_mesh_move** index move_x move_y move_z

This option moves the mesh. Thus you get the same amount of elements and nodes, but just moved in space. In 1d you only should specify `move_x`. In 2d you only should specify `move_x` `move_y`.

6.214 **control_mesh_multiply** index number_of_multiplications

The mesh is multiplied `number_of_multiplications` times. In each multiplication the mesh gets double the amount of elements, because for each element a new element is generated with the same nodes.

6.215 **control_mesh_refine_globally** index refinement_type

This record activates global mesh refinement or global mesh coarsening. This is not available for `-tria3` and `-tet4` elements. Either `refinement_method` is `-h_refinement` (more of the same elements) or `refinement_method` is `-p_refinement` (higher order elements) or `refine_method` is `-p_coarsen` (lower order elements).

As a special option for the `-h_refinement` method, the format `refine_globally index -h_refinement switch_ξ switch_η switch_ζ` can be used. For example in 1D, only `refine_globally index -h_refinement switch_ξ` should be specified. For example in the `-hex8` element, `ξ` is the isoparametric coordinate running from the first node to the second node, `η` runs from the first node the third node and `ζ` runs from the first node to the fifth node. A isoparametric direction will be refined if the corresponding switch is set to `-yes`. This option allows for refinement in specific directions. It should be used with care however, and only gives proper results if the `ξ`, `η` and `ζ` directions of the elements match.

The **control_mesh_refine_globally** will automatically merge nodes which have the same position in space.

Rules for old and new:
• A new generated element inherits its data items from the old element it is generated from.

• If a new generated node is placed on an old element edge it inherits those data items of the old nodes on that old edge that have a property in common; then arbitrarily the data item of one of the old nodes is taken.

• If a new generated node is placed inside an old element it inherits those data items of the old nodes of that old element that have a property in common; then arbitrarily the data item of one of the old nodes is taken.

• For all new nodes the node_dof records are interpolated from the old element nodes node_dof records by using the old element interpolation functions.

See also control_mesh_refine_globally_geometry.

6.216 control_mesh_refine_globally_geometry index geometry_entity_item geometry_entity_index

This record can be used together with the control_mesh_refine_globally record with the same index. If all nodes of an element edge is part of the geometrical entity, the new generated nodes are also placed on that geometrical entity. This typically is used to follow curved edges of the domain.

The control_mesh_refine_locally will automatically merge nodes which have the same position in space.

6.217 control_mesh_refine_locally index percentage

An elements will be refined depending on the size of a solution variable. The solution variable can be chosen via control_mesh_refine_locally_dof.

The percentage of elements which will be refined is specified by percentage. Typically percentage is 10 or so.

This local mesh refinement is only available for -bar2, -bar3, -tria3, -tria6, -tet4 and -tet10 elements; there should be no other elements in the mesh.

See the rules for old and new at control_mesh_refine_globally.

6.218 control_mesh_refine_locally_dof index dof

With dof you can set which dof will be used for deciding if an element should be refined. The size of the doffield will be used.

Possibilities for dof are: -materi_damage, -materi_displacement, -materi_plasti_kappa, -materi_plasti_kappa_shear, -materi_strain_elasti, -materi_strain_plasti, -materi_strain_total, -materi_stress, -materi_velocity, -materi_void_fraction and.

As a special option you can set dof to -nothing; then an element is refined always.

For finding localization zones (e.g. shear bands) choosing -materi_strain_plasti or -materi_damage seems to be most robust.

See also control_mesh_refine_locally_geometry.
6.219  **control_mesh_refine_locally_geomet**y  *index*  geometry_entity_item  geometry_entity_index

This record can be used together with the **control_mesh_refine_locally** record with the same index. If all nodes of an element edge is part of the geometrical entity, the new generated nodes are also placed on that geometrical entity. This typically is used to follow curved edges of the domain.

6.220  **control_mesh_refine_locally_minimal_size**  *index*  minimal_size

Element with minimal size below the specified *minimal_size* will not be refined. The minimal element size is defined as the largest node distance between nodes of the element. Default the minimal allowed size is 0.

6.221  **control_mesh_refine_locally_not**  *index*  geometry_entity_index 0 geometry_entity_index

The refinement as specified in the **control_mesh_refine_locally** record with the same index, will not be applied on the geometry specified by *geometry_entity_index_0*.

6.222  **control_mesh_refine_locally_not_method**  *index*  method

Set *method* to -all or -any. If *method* is set to -all, then the corresponding **control_mesh_refine_locally_not** is applied to elements for which all nodes are inside the specified geometry. If *method* is set to -any, then the corresponding **control_mesh_refine_locally_not** is applied to elements for which any of the nodes is inside the specified geometry. Default *method* is -all.

6.223  **control_mesh_refine_locally_only**  *index*  geometry_entity_index 0 geometry_entity_index

The refinement as specified in the **control_mesh_refine_locally** record with the same index, will only be applied on the geometry specified by *geometry_entity_index_0*.

6.224  **control_mesh_refine_locally_only_method**  *index*  method

Set *method* to -all or -any. If *method* is set to -all, then the corresponding **control_mesh_refine_locally_only** is applied to elements for which all nodes are inside the specified geometry. If *method* is set to -any, then the corresponding **control_mesh_refine_locally_only** is applied to elements for which any of the nodes is inside the specified geometry. Default *method* is -all.

6.225  **control_mesh_remove**  *index*  method  element_group 0 element_group 1 element_group 2 ...

With *method* set to -method1 you can remove elements of *element_group_0* if they are completely located inside a elements of groups *element_group_1, element_group_2* etc.
With *method set* to -method3 you can remove elements if all nodes of these elements have an mpc (that is, node_mpc exists for all nodes of an element). Thus for *method set* to -method3 you do not need to specify element_group_0, element_group_1, element_group_2 etc.

6.226 control_mesh_remove_geometry index geometry_item_name geometry_item_index

With this record you can restrict to which geometry the control_mesh_remove will be applied.

6.227 control_mesh_remove_keep_geometry index geometry_item_name geometry_item_index

If elements are being removed once with the control_mesh_remove command, they keep on being removed in the future if they are part of the geometry as specified in this control_mesh_remove_keep_geometry command.

6.228 control_mesh_remove_really index switch

If *switch* is set to -yes elements removed by the control_mesh_remove command are really removed. If *switch* is set to -no elements removed by the control_mesh_remove command are not really removed; they are made inactive instead, and will become active again at the moment that they are no longer being removed by the control_mesh_remove command.

6.229 control_mesh_remove_really_activate_all index switch

See control_mesh_remove_really_activate_factor.

6.230 control_mesh_remove_really_activate_factor index factor

When an element becomes active again, after being removed before, we need to decide what the stresses and pore pressures are in nodes of the activated element.

For nodes of the element which were not connected to already active elements nothing extra is done. For nodes of the element which were not connected to already active elements, the average value of the stresses and pore pressures from already active nodes (connected to already active elements) are taken. However, with this *factor*, you can specify to which part the stresses and pore pressures of already nodes will actually be used. Set *factor* to 0 if you want to reactivate the nodal stresses and pore pressures with 0. Set *factor* to 1 if you want to reactivate the nodal stresses and pore pressures completely with the stresses and pore pressures from already active neighbour nodes. Or set a *factor* between 0 and 1 if you do something in between.

As an extra option you can set the *switch* in control_mesh_remove_really_activate_all to -yes. Then the stresses and pore pressures in ALL nodes of the activated element will be reactivated using the *factor* (so not only the nodes which were not active before).

Default, if control_mesh_remove_really_activate_factor is not specified, the *factor* is set to 1.
6.231 control_mesh_renumber index lowest_element lowest_node

The element numbers are made strictly sequential starting from lowest_element and the node numbers are made strictly sequential starting from lowest_node. Beware using control_renumber in combination with, for example, node numbers in printing of node_dof records; use post_point records instead.

6.232 control_mesh_renumber_element_geometry_offset index offset

While renumbering elements the element geometry number will be offset with offset.

6.233 control_mesh_renumber_element_group_offset index offset

While renumbering elements the element group number will be offset with offset.

6.234 control_mesh_rotate index n

After rotation n is the number of elements in rotational direction for a rotation over 360 degrees. After rotation the old y direction becomes the new z direction. The following data is transferred in the rotation process: element, element_group, node and node_dof. A 2D -tria3 element becomes a 3D -prism6 element and a 2D -quad4 element becomes a 3D -hex8 element; other 2D elements can presently not be rotated. All data that is not valid in 3D, like for example a 2D line etc, will be deleted in the rotation process.

This control_mesh_rotate is convenient when the first part of calculation is axisymmetric, for example loading a pile vertically in a soil, and the second part of the calculation is 3D, for example loading the top of the pile in some horizontal direction. Then first an axi-symmetric calculation can be performed, and the results can be used to start a 3D calculation.

If a -quad4 elements has a side on the y-axis in the 2D mesh, the element is rotated to a -prism6 element; the -quad4 element should have the side with local node numbers 0 and 1 on the y-axis, which is the case if you generated the elements with a control_mesh_macro You should not use other elements with a side on the y-axis when rotating the mesh.

This control_mesh_rotate deletes all data, except element, element_group, node, node_dof, element_interface_strain and element_interface_stress will be rotated to 3D. Furthermore, control_input will available afterwards, so that all 3d data can be set in an extra input file, which is read after the mesh rotation.

If you use any history variables in the model, these should be scalars (and thus not vectors or matrices); otherwise rotation will not go ok for the history variables.

6.235 control_mesh_rotate_angle index angle

With angle you can specify an angle in degrees up to which the mesh rotation should be done for the control_mesh_rotate with the same index. Typically you could use 90 degrees or 180 degrees for angle. Default, if this control_mesh_rotate_angle is not specified, angle will be set to 360.
6.236 control_mesh_split index switch

If switch is set to -yes then each -quad4 element is split into four -tria3 elements and each -hex8 element is split into twelve -tet4 elements. Further, each -quad9 element is split into four -tria6 elements and each -hex27 element is split into six -tet10 elements. Further, each -tria6 element is split into four -tria3 elements.

See the rules for old and new at control_mesh_refine_globally. See also control_mesh_split_element_to and control_mesh_split_only.

Splitting a 3D mesh will only work correctly on certain regular grids; you need to check the splitted mesh.

6.237 control_mesh_split_element_from index name

Split only elements with the specified name.

6.238 control_mesh_split_element_to index name

If you are splitting -hex8 elements, then you can set name either to -tet4 or -prism6. Default, if control_mesh_split_element_to is not specified, -tet4 is used.

If you are splitting -hex27 elements, then you can set name either to -tet10 or -prism18. Default, if control_mesh_split_element_to is not specified, -tet10 is used.

6.239 control_mesh_split_only index geometry_entity geometry_entity_index

If this record is used, the corresponding control_mesh_split record will only be applied on elements which have at least one node on the geometry specified by index geometry_entity_name geometry_entity_index.

6.240 control_mesh_truss_distribute_mpc index switch

If switch is set to -yes the nodes of truss elements are fixed with multi point constraints (mpc’s) to the isoparametric elements through which the trusses run. This typically can be used for modeling reinforcement bars in a concrete embedment, where the bars follow the displacements (and temperatures if present) of the concrete.

If control_mesh_truss_distribute_mpc_exact switch is set to -yes, truss elements are redistributed (that is, more small truss elements will be made), in such way that each truss gets a node when it enters an isoparametric element or ends internally in an isoparametric element. This control_mesh_truss_distribute_mpc_exact comes handy when you initially have large trusses relative to the isoparametric elements.

Truss below a minimum length as specified in control_mesh_truss_distribute_mpc_exact_minimal_length will not be generated; default the minimal length tolerance is set to some small value. With control_mesh_truss_distribute_mpc_exact_minimal_length_connect you can determine if the generated trusses jumping a space below the minimal length will be connected or will be not-connected (loose); set the switch to -yes if you want the truss to be connected in such case. Please realise that the connection is ensured only for the trusses generated from 1 old truss; connection is not ensured for trusses generated from different old truss elements.
This `control_mesh_truss_distribute_mpc` option is done for truss groups as specified in `control_mesh_truss_distribute_mpc_element_group_truss` or in `control_mesh_truss_distribute_mpc_geometry_truss`. Only one of `control_mesh_truss_distribute_mpc_element_group_truss` and `control_mesh_truss_distribute_mpc_geometry_truss` can be specified. If none of `control_mesh_truss_distribute_mpc_element_group_truss` and `control_mesh_truss_distribute_mpc_geometry_truss` is specified the distribution will be done for all trusses.

Default Tochnog will look for all isoparametric elements how to distribute the trusses. To save computer time you can restrict the geometry or element group of the isoparametric elements where Tochnog will look with `control_mesh_truss_distribute_mpc_element_group_isoparametric` end `control_mesh_truss_distribute_mpc_geometry_isoparametric`.

Please notice that if you are using geometries in `control_mesh_truss_distribute_mpc_geometry_truss` or `control_mesh_truss_distribute_mpc_geometry_isoparametric` these can in fact be a `geometry_set`.

In case you specify both of the above `*_truss` and `*_isoparametric`, the number of specified values (groups or geometries) should be the same. Then the first value specified for the truss will be combined with the first value specified for the isoparametric elements, the second value specified for the truss will be combined with the second value specified for the isoparametric elements, etc. For example, if you specify two groups for `control_mesh_truss_distribute_mpc_element_group_truss` and two groups for `control_mesh_truss_distribute_mpc_element_group_isoparametric` the first specified truss group will be distributed over the first specified isoparametric group, and the second specified truss group will be distributed over the first specified isoparametric group.

If `switch` in `control_mesh_truss_distribute_mpc_air` is set to `-yes`, trusses will also be generated in the center of the truss is not inside an isoparametric element. If `switch` in `control_mesh_truss_distribute_mpc_air` is set to `-no`, trusses will not be generated in the center of the truss is not inside an isoparametric element. Default `switch` is `-yes`.

A typical input file looks like:

```
control_mesh_truss_distribute_mpc 10 -yes
control_mesh_truss_distribute_mpc_exact 10 -yes
control_mesh_truss_distribute_mpc_geometry 10 -element_geometry 123
```

Only one `control_mesh_truss_distribute_mpc` record is allowed in the input file. As a special option you can also generate `truss_beam` elements in stead of `truss` elements.

### 6.241 control_mesh_truss_distribute_mpc_air

`index switch`

See `control_mesh_truss_distribute_mpc`.

### 6.242 control_mesh_truss_distribute_mpc_dof
dof_0 dof_1 ...

The `dof_0 dof_1 ...` specify the dof’s that should be set equal, e.g. `-velx, -vely` etc.
6.243  control_mesh_truss_distribute_mpc_element_group_truss index element_group_0 element_group_1 ...

See control_mesh_truss_distribute_mpc.

6.244  control_mesh_truss_distribute_mpc_element_group_isoparametric index element_group_0 element_group_1 ...

See control_mesh_truss_distribute_mpc.

6.245  control_mesh_truss_distribute_mpc_exact index switch

See control_mesh_truss_distribute_mpc.

6.246  control_mesh_truss_distribute_mpc_exact_minimal_length index tolerance

See control_mesh_truss_distribute_mpc.

6.247  control_mesh_truss_distribute_mpc_exact_minimal_length_connect index switch

See control_mesh_truss_distribute_mpc.

6.248  control_mesh_truss_distribute_mpc_geometry_truss index geometry_entity_name_0 geometry_entity_index_0 geometry_entity_name_1 geometry_entity_index_1 ...

See control_mesh_truss_distribute_mpc.

6.249  control_mesh_truss_distribute_mpc_geometry_isoparametric index geometry_entity_name_0 geometry_entity_index_0 geometry_entity_name_1 geometry_entity_index_1 ...

See control_mesh_truss_distribute_mpc.

6.250  control_mpc_apply index switch

If switch is set to -yes then mpc conditions will be used for the control index. If switch is set to -no then mpc conditions will not be used for the control index. Default, if control_mpc_apply is not specified, switch is set to -yes.
6.251  control_mpc_element_group  index switch

If switch is set to -yes the mpc_element_group records will be evaluated at all timesteps for the current control index. If switch is set to -no the mpc_element_group records will only be evaluated when the mesh has been changed. This option can only be used if there are only mpc’s coming from this option, and not from other mpc data.

Default, if control_mpc_element_group is not specified, the switch is set to -no.

6.252  control_plasti_apply  index switch

If switch is set to -no, any plasticity data in the input file will be ignored. This is done for timestep records with the same index.

This option is convenient for testing your input file just linear, without the need to outcomment each and every part with plasticity data. See also plasti_apply.

6.253  control_post  index switch

If switch is set to -yes all post processing commands are evaluated for index index. This enforces evaluation of post processing command even if no timesteps are performed or so.

6.254  control_post_apply  index switch

Setting switch to -no prevents post processing commands to be evaluated for control commands with the same index. Postprocessing commands have post in the name (only the post_node_rhside_ratio will be evaluated always, independent of control_post_apply).

6.255  control_post_element_force  index switch

You can save CPU time in timesteps with the same index by setting switch to -no, which prevents post_element_force commands to be evaluated in timesteps with the same index.

6.256  control_print  index data_item_name_0 data_item_name_1 . . .

This is the normal printing command. A control_print record causes the data items with name data_item_name_0, etc. to be printed. Example

control_print 1 -node -node_dof

See also: print_filter.
6.257 control_print_beam_force_moment index switch

This option prints the beam forces and moments through a set of beams starting at place $x_{\text{start}}$, $y_{\text{start}}$, $z_{\text{start}}$ and ending at $x_{\text{end}}$, $y_{\text{end}}$, $z_{\text{end}}$ as specified in control_print_beam_force_moment_coordinates. In 2D only $x$ and $y$ coordinates need to be specified. The forces and moments are printed in the file beam_force_moment.$\text{index}$. In fact, if the element contains a truss (either a truss element or a truss-beam element), the truss force will be used for the axial force. The first column in the file is the distance from the start point. The following columns contain in the local beam axes force_x_first_node force_y_first_node force_z_first_node moment_x_first_node moment_y_first_node moment_z_first_node force_x_second_node force_y_second_node force_z_second_node moment_x_second_node moment_y_second_node moment_z_second_node. The switch needs to be set to -separate_index or -separate_sequential. See also control_print_beam_force_moment_coordinates.

6.258 control_print_beam_force_moment_coordinates index $x_{\text{start}}$ $y_{\text{start}}$ $z_{\text{start}}$ $x_{\text{end}}$ $y_{\text{end}}$ $z_{\text{end}}$

See control_print_beam_force_moment.

6.259 control_print_beam_force_moment_switch index switch

If you set switch to -yes, the definition of the beam forces and moments is changed (multiplied with a -1). So you can get exactly the definition that you want.

6.260 control_print_database index switch

If switch is set to -separate_index, the complete database is be printed. See the example below

control_print_database 6 -separate_index

This database contains the complete status of the calculation. For example if index is 6, the database is printed in the file input_file_name6.dbs. As a special option, you can print databases with sequential numbers by setting switch to -separate_sequential.

If tochnog exists with an error, for example due to non-convergence, a complete database is printed in input_file_name_error.dbs. Otherwise, a complete database will be printed at the end of the calculation.

6.261 control_print_database_method index method

If method is set to -all then all database base records will be printed in the database. If method is set to -size_tot then the size of all database base records will be printed in the database. If method is set to -size_tot_large then the size of database base records larger then 1 Mb will be printed in the database.

When using -size_tot or -size_tot_large also the size of the system matrix is printed in the database.

Default, if control_print_database_method is not specified, the method is set to -all.
6.262 control_print_data_versus_data

This option prints columns of data for each time step. Print in the first column the number_0 value of data_item_name_0 with index_index_0. Similar in the second column for data_item_name_1 with index_index_1, Etc. (for all values). All results will be printed in the file problemname.dvd.

Typically, the data item names can be -node_dof such that dof’s can be printed against each other in time. If the data item names are -node_dof, then number_0 and number_1, etc. can be names of dof_label (eg -velx).

Also typically, the data item names can be -node_dof_calcul such that post calculation results can be printed against each other in time. If the data item names are -node_dof_calcul or post_point_dof_calcul or so, then number_0 and number_1, etc. can be names of post_calcul_label (eg -aept).

Otherwise, for example, if number_0 is 3 then the fourth value of data_item_name_0 is printed.

Example:

```
control_print_data_versus_data 0 -node_dof 2 -temp
-node_dof 2 -sigxx -node_dof 2 -sigxx
```

Another example:

```
post_point 0 0.0 1.0
post_calcul -materi_stress -average -materi_stress -size_dev
control_print_data_versus_data 20
-time_current 0 0
-post_point_dof_calcul 0 0 -post_point_dof_calcul 0 1
```

In the last example, the -post_point_dof_calcul 0 0 stands for the post_point_dof_record with index 0 and the 0’th number which is the first value so the average of the stresses.

For data that is not present Tochnog will print a 0.

See also: control_print.

6.263 control_print_dof

Results for the primary dof’s will be printed, including also the coordinates at which the results hold. Also results for node_dof_calcul records will be printed. The printed files will contain lines like x, y, z and dof (where dof is the dof, e.g. temp). In 1D only x will be printed, etc.

As extra also the coordinates themselves are printed in files.

If switch is set to -separate_index the filenames will be like dof.index.

If switch is set to -separate_sequential then the filenames will be sequentially numbered, like dof.0, dof.1, etc.
6.264 control_print_dof_id index switch

This record works in combination with the control_print_dof record. If switch is set to -yes also the node number ('identity') is written. So for example in 3D the file will contain lines with \( x, y, z, \text{dof} \) and node number.

Default, if control_print_dof_id is not specified, switch is set to -yes.

6.265 control_print_dof_smooth_dof index dof_0 dof_1 ...

This option allows you to smooth results for control_print_dof. With dof_0 dof_1 ... you specify the dof's to be smoothed. As a special option you can specify -all so that all dof's will be smoothed.

The smoothing is done a number of times, with increasing smooth results. You can specify this number of times with the optional control_print_dof_smooth_n; if you don’t specify this optional number of smoothings it is done 10 times.

6.266 control_print_dof_smooth_n index number_of_smoothings

See control_print_dof_smooth_dof.

6.267 control_print_dof_line index switch

This control_print_dof_line record together with the control_print_dof_line_coordinates and control_print_dof_line_n records print values of the node_dof records and node_dof_calcul records along a line in space to files.

The start point of the first line segment is given by \( x_0 y_0 z_0 \), and the end point of the first line segment is given by \( x_1 y_1 z_1 \), the start point of the second line segment is given by \( x_1 y_1 z_1 \), and the end point of the second line segment is given by \( x_2 y_2 z_2 \), etc.

In 1D only the x-coordinates of the start point and end point need to be specified, etc. The parameter \( n \) determines how many points will be printed along the line.

The printed files will contain lines like \( x, y, z \) and \text{dof} (where \text{dof} is the dof, e.g. temp). In 1D only \text{x} will be printed, etc.

If switch is set to -separate_index the filenames will be like dof.index.

If switch is set to -separate_sequential then the filenames will be sequentially numbered, like dof.0, dof.1, etc.

In control_print_dof_line_method you can set node_type either to -node or -node_start_refined. Then the coordinates in the printed file will contain either the values of node or the values of node_start_refined. In case you use an updated lagrange formulation where the mesh nodes follow the material the values of node and node_start_refined will differ; in case you do a geometrically linear analysis the values will not differ. Default node_type is set to -node_start_refined.

With control_print_dof_line_element_group you can set a specific element groups for which the printing should be done.
With control_print_dof_line_eps_iso you set the tolerance with which a point of the specified line is accepted to be part of an element. The default value is 1.e-3. You can increase the default value if if the mesh is not exactly adjusted to the line.

6.268 control_print_dof_line_coordinates index x_0 y_0 z_0 x_1 y_1 z_1 ...

See control_print_dof_line.

6.269 control_print_dof_line_element_group index element_group_0 element_group_1 ...

See control_print_dof_line.

6.270 control_print_dof_line_eps_iso index eps_iso

See control_print_dof_line.

6.271 control_print_dof_line_method index node_type

See control_print_dof_line.

6.272 control_print_dof_line_n index n

See control_print_dof_line.

6.273 control_print_dof_line_time index switch

If switch is set to -yes the first line of each file will specify the time_current at which the file is written (in gnuplot comment format).

6.274 control_print_dof_point index switch

This control_print_dof_point record prints values of the node_dof records and node_dof_calcul records in a point in space to files.

The point is given by x y z,

In 1D only the x-coordinates of the start point and end point need to be specified, etc.

The printed files will contain lines like x, y, z and dof (where dof is the dof, e.g. temp). In 1D only x will be printed, etc.

If switch is set to -separate_index the filenames will be like dof.index.

If switch is set to -separate_sequential then the filenames will be sequentially numbered, like dof.0, dof.1, etc.
6.275 `control_print_dof_point_coordinates index x y z`

See `control_print_dof_line`.

6.276 `control_print_dof_point_time index switch`

If `switch` is set to `-yes` the first line of each file will specify the `time_current` at which the file is written (in gnuplot comment format).

6.277 `control_print_dof_rhside index switch`

If `switch` is set to `-yes` then results for right-hand-side for the primary dof’s will be printed, including also the coordinates at which the results hold.

For example, for the file `temp_rhside.index` will contain lines containing `x`, `y`, `z` and right-hand-side of `-temp` (that is, heat flux). In 1D only `x` will be printed, etc.

6.278 `control_print_element index data_item_name`

With this option you can print values from element data versus coordinates. Select either `-element_truss_force` or `-element_beam_force_moment` for `data_item_name`.

The normal truss forces of the `-element_truss_force` records will be printed in the file `element_truss_force_n.index`. This file will contain lines containing `x`, `y`, `z` and normal truss force. In 1D only `x` will be printed, etc.

The lateral beam shear forces of the `-element_beam_force_moment` records will be printed in the file `element_beam_force_moment_q.index`. This file will contain lines containing `x`, `y`, `z` and lateral beam shear force. In 1D only `x` will be printed, etc. The shear force will always be calculated as an absolute value.

The beam moments of the `-element_beam_force_moment` records will be printed in the file `element_beam_force_moment_m.index`. This file will contain lines containing `x`, `y`, `z` and beam moment. In 1D only `x` will be printed, etc.

How the data is printed depends on how `method` is set in `control_print_element_method`. If method is set to `-middle` then only the average value of the element data and the coordinate of the middle of the element is printed for each element. If method is set to `-node` then the two nodal values and nodal coordinates are printed for each element.

6.279 `control_print_element_method index method`

Set `method` to `-middle` or `-node`. If `control_print_element_method` is not specified then `-middle` is used. See also `control_print_element`.

6.280 `control_print_filter index print_filter_index_0 print_filter_index_1 ...`

See `print_filter`. 

166
6.281 control_print_frd index switch

Activate printing of results in Calculix output format frd. Only results for 2D and 3D isoparametric elements are printed presently. For structural elements (trusses, beams, ...) nothing is printed.

For example if the input file name is excavation.dat and index is 100 and switch is set to -separate_index then results are printed in the file is excavation_100.frd.

For example if the input file name is excavation.dat and switch is set to -separate_sequential then results are printed in the files is excavation_0.frd, excavation1.frd, etc.

For example if the input file name is excavation.dat and switch is set to -yes_sequential then results are printed in the files is excavation.frd.

The frd files can be plotted by the prepmmax pre- and postprocessor, see http://lace.fs.unimb.si/wordpress/borovinsek/?page_id=41. Since prepmmax presently cannot plot 2d elements, Tochnog extrudes 2d calculations to a fictive thickness of 1 when printing the frd files.

You also can use the CGX postprocessor of calculix itself, see http://www.bconverged.com/download.php. Since Freecad and Prepmmax like specific names of results we write the following names:

- DISP for materi_displacement or materi_velocity_integrated
- STRESS for materi_stress
- TOSTRAIN for materi_strain_total
- NDTEMP for condif_temperature
- PEEQ for post_calcul-groundflow_pressure-total_pressure (if print_frd_freecad is -yes)

For other results the Tochnog names are used, up to the allowed 8 characters in frd files.

Freecad presently only reads frd files which contain displacements and stresses and total strains. Thus, if these are not initialised in your tochnog input file and if print_frd_freecad is -yes then tochnog will not write the frd file for freecad.

See also control_print_frd_prepmmax.

6.282 control_print_frd_freecad index switch

If switch is set to -yes the frd files are written specifically suited for the freecad program. This is done for the control_print_frd command with the same index. If switch is set to -no the frd files are not written specifically suited for the freecad program. See also print_frd_freecad.

6.283 control_print_frd_prepmmax index switch

If switch is set to -yes the frd files are written specifically suited for the prepmmax program. This is done for the control_print_frd command with the same index. If switch is set to -no the frd files are not written specifically suited for the prepmmax program. See also print_frd_prepmmax.
6.284  control_print_frequency_timeinterval index timeinterval

This control_print_frequency_timeinterval record causes control_print_* other than
control_print, control_print_history and control_print_data_versus_data to be done
each time after a time interval has passed, and always also at the end of the time increment. This
control_print_frequency_timeinterval record should only be used in combination with control_timestep (with the same index). All control_print_* printing will be influenced except
control_print, control_print_history and control_print_data_versus_data printing.

Example:

    control_timestep 10 0.04 0.41
    control_print_gid 10 -yes
    control_print_frequency_timeinterval 10 0.15

In this example gid data is written at times 0.16, 0.32, 0.41

6.285  control_print_frequency_timestep index timestep

This control_print_frequency_timestep record causes control_print_* other than control_print, control_print_history and control_print_data_versus_data to be done
each time after a number of time timesteps has passed, and always also at the end of the time in-
crement. This control_print_frequency_timestep record should only be used in combination
with control_timestep (with the same index). All control_print_* printing will be influenced except
control_print, control_print_history and control_print_data_versus_data printing.

Example:

    control_timestep 10 0.04 0.41
    control_print_gid 10 -yes
    control_print_frequency_timestep 10 5

In this example gid data is written at times 0.20, 0.40, 0.41

6.286  control_print_gid index switch

Print the mesh and the dof's in a file which can be plotted with the GID pre-post processor if
switch is set to -yes. For example, if the input file is called turbine.dat then the mesh is written
in the turbine_flavia.msh file. The results are written in the turbine_flavia.res.

The mesh and results for dof's will always be written at the end of the calculation.

Since GID gets confused when the number of elements changes between several control_print_gid
records, Tochnog will only print GID results for the last mesh used.

Coordinates for nodes will be written in the original configuration. If materi_velocity is ini-
tialized, also a vector mesh_deform will be written for GID which contains the deformation
between the original mesh configuration and the deformed mesh configuration. Use the deform
mesh menu in GID, to draw the deformed configuration by applying the vector mesh_deform
with a factor 1. If `materi_displacement` is initialised the `mesh_deform` contains the nodal displacements. If `materi_velocity_integrated` is initialised the `mesh_deform` contains the integrated nodal velocities. Else `mesh_deform` contains the current value of `node` minus the start value of the nodal coordinates.

For 2D interface elements which have strains and stresses, the normal stress `interface_sign`, the tangential shear stress `interface_sigt`, the normal strain `interface_epsn` and the tangential shear strain `interface_epst`, are written to the GID results file.

The following data is written also to the gid file and can serve as a help to check the validity of your input file. This data is only available after one or more time steps are taken.

- `condif_bounda_dof`, boundary conditions on temperature.
- `condif_heat_edge_normal`, distributed prescribed heat flow on an edge.
- `condif_convection_edge_normal`, distributed convection heat flow on an edge.
- `condif_radiation_edge_normal`, distributed convection heat flow on an edge.
- `groundflow_bounda_dof`, boundary conditions on groundflow hydraulic pressure head.
- `materi_bounda_force`, discrete forces on nodes.
- `materi_force_edge`, distributed forces on nodes.
- `materi_force_edge_normal`, distributed normal forces on nodes.
- `materi_force_edge_projected`, distributed projected forces on nodes.
- `materi_force_edge_water`, distributed water forces on nodes.
- `materi_force_volume`, distributed volume forces on nodes.
- `materi_force_gravity`, distributed gravity forces on nodes.
- `materi_bounda_dof`, boundary conditions on materi velocity on nodes.
- `materi_support_edge_normal`, distributed support forces on nodes.
- `materi_rhside_free`, unbalance forces for `materi_velocity` (for free displacements) on nodex.
- `materi_rhside_fixed`, reaction forces for `materi_velocity` (for fixed displacements) on nodex.
- `element_materi_force_edge`, norm of distributed forces on elements.
- `element_materi_force_edge_normal`, norm of distributed normal forces on an edge on elements.
- `element_materi_force_edge_water`, norm of distributed water forces on an edge on elements.
- `plasti_reduction_factor`, reduction factor for plasticity parameters from `group_materi_plasti_element_group`, etc.

If you have specified `print_node_geometry_present` then the gid files will contain `geometry_...` values which are 1 on nodes present in a geometry.

The `materi_bounda_dof` you can view in gid with View results, Display vectors, materi bounda dof, All. The other data you can view in GID for example with View results, Display
vectors, force edge normal, | force edge normal | . Above with 'distributed' we mean that results are per unit area.

For isoparametric elements the element group number will be printed.

For all post_point a point will be plotted in the gid files, with element group number 1000 + the index of the post point.

As a special option, you can set switch to -separate_index. Then the mesh and results will be printed in separate files for GID, numbered with index. The option comes handy when the mesh changes during the calculation; GID cannot plot that if the mesh and results are in the same file.

As a further special option, you can set switch to -separate_sequential. Then the mesh and results will be printed in separate files for GID, number sequentially.

6.287 control_print_gid_batch index switch

If switch is set to -yes Tochnog calls GiD as batch program to generate some typical png files of results. This works i.e.w. a control_print_gid with the same index.

The following files are generated: mesh__index.png, element_group__index.png, element_group__index.png, materi_velocity_integrated_veliy__index.png, materi_stress_sigy_y__index.png and groundflow__pressure_total.png.

For this GiD needs to be installed on your computer. You also need to take care that GiD can be found, so you need to adjust your PATH environment symbol.

6.288 control_print_gid_beam_vectors index switch

If switch is set to -yes, force and moment vectors will be plotted for -beam and -truss_beam elements. The force and moment vectors will be plotted perpendicular to the length direction and a user specified plane-normal vector, see control_print_gid_beam_vectors_normal. The length of the plotted vectors measures the size of the forces and moment.

The vectors will be plotted as element result, so not as nodally averaged result.

Attention: this control_print_gid Beam vectors is a special plotting option, to get each beam force and moment result as vector plot, with the possibility to influence the direction of the vectors with control_print_gid Beam_vectors_normal. Default Tochnog plots the beam result already as scalar values for each beam element.

6.289 control_print_gid Beam_vectors_normal index normal_x normal_y normal_z

This record gives you the possibility to influence the plane in which the moment vectors generated by the control_print_gid Beam_vectors will be plotted. In fact this control_print_gid Beam_vector specifies the normal to the plotting plane. If this control_print_gid Beam_vectors_normal is not specified then 0 0 1 is taken as normal.

6.290 control_print_gid_contact_spring2 index number_of_nodes

Set number_of_nodes to 2 if you want to draw contact_spring2 with two nodes, and to 1 if you want to draw contact_spring2 with one node. Default, if control_print_gid_contact_spring2
is not specified, then 1 node is used.

6.291  control_print_gid_coord index switch

If switch is set to -yes the coordinates of nodes is plotted in gid. If switch is set to -no the coordinates of nodes is not plotted in gid. Default switch is set to -yes.

6.292  control_print_gid_dof index initialisation_name_0 initialisation_name_1 ... 

When you specify this record only the solution fields initialisation_name_0, initialisation_name_1 etc will be printed to the gid files. So the files become smaller in size. This is especially convenient for very large calculations. The names initialisation_name_0, initialisation_name_1 are names from the initialisation part like -condif_temperature, -materi_velocity, -materi_stress etc. In case you do not want any field to be printed in the gid file use control_print_gid_dof index -none.

See also control_print_gid_other.

6.293  control_print_gid_dof_calcul index calcul_0 calcul_1 ... 

When you specify this record only the post fields calcul_0, calcul_1 etc will be printed to the gid files. So the files become smaller in size. This is especially convenient for very large calculations. See post_calcul_label for the allowed names of calcul_0, calcul_1 etc. In case you do not want any post field to be printed in the gid file use control_print_gid_dof_calcul index -none.

See also control_print_gid_other.

6.294  control_print_gid_element_group index element_group_0 element_group_1 ... 

Select specific element groups for the gid files. If this record is not specified all element groups will be used.

6.295  control_print_gid_element_mpc index switch

If switch is set to -yes also elements which have mpc’d nodes will be printed in the gid files. If switch is set to -no a elements which have mpc’d nodes will not be printed in the gid files. Default, if control_print_gid_element_mpc is not specified, switch is set to -yes.

6.296  control_print_gid_empty index switch

If switch is set to -yes, empty elements will be show in GID plots. If switch is set to -no, empty elements will not be shown. Default switch is set to -no.

See also element_empty.
6.297 control_print_gid_group index switch

If switch is set to -yes the element groups are plotted in gid as result field; in gid you can do a contour_fill to visualize this result field. If switch is set to -no the element groups are not plotted in gid as result. Default switch is set to -yes.

6.298 control_print_gid_mesh_activate_gravity index switch

See also mesh_activate_gravity_time.

6.299 control_print_gid_method index method

If method is set to -node, results will be written for global nodes to the gid files. Gid will interpolate between the nodes, to fill contour plots, etc. Hence, you get continuous plots fields.

If method is set to -element, results will be written element-by-element to the gid files, so that any discontinuity in fields can be seen. The results will be written using the integration point values.

If method is set to -element_average, results will be written element-by-element to the gid files, so that any discontinuity in fields can be seen. The results will be written using the average of the integration point values.

If method is set to -node_elemen, results will be written with continuous fields to the gid files, but at group jumps discontinuous fields are allowed.

For -element and -node_elemen gid cannot plot some results like ’contour fill’ for all elements if there are several type of elements (quad4, tria3, ...) in the mesh. You can only select on specific element type for the plot.

If this control_print_gid_method record is not specified then method is set to -node. If global_element_dof is set to -no the element-by-element stresses are not available, so then -element_average and -node_elemen should not be used.

6.300 control_print_gid_node_method index method

If method is set to -node, these coordinates are written. If method is set to -node_start_refined, these coordinates are written. If method is set to -node_deformed_mesh, these coordinates are written. Default method is -node_start_refined.

6.301 control_print_gid_other index switch

If switch is set to -yes also other things like boundary conditions, mesh deformation etc. are printed in the gid files. If switch is set to -no these other things are not printed in the gid files. Default switch is set to -yes.

6.302 control_print_gid_save_difference index switch

If switch is set to -yes then data differences relative to a saved status will be plotted. See control_data_save.
6.303 control_print_gid_safety_slip_critical index switch

If switch is set to -yes, then for a safety analysis with control_safety_slip only the critical slip surface will be plotted. Default, if switch is not set, all evaluated slip surfaces will be plotted. The critical surface is either determined over all safety surfaces, or otherwise in case sets are specified a critical surface is determined for each set.

Furthermore, always the normal stresses and shear stresses on the slip surfaces will be plotted.

6.304 control_print_gid_smooth_dof index dof_0 dof_1 ...

This option allows you to smooth results in gid files. With dof_0 dof_1 ... you specify the dof’s to be smoothed. As a special option you can specify -all so that all dof’s will be smoothed.

The smoothing is done a number of times, with increasing smooth results. You can specify this number of times with the optional control_print_gid_smooth_n; if you don’t specify this optional number of smoothings it is done 10 times.

6.305 control_print_gid_smooth_n index number_of_smoothings

See control_print_gid_smooth_dof.

6.306 control_print_gid_spring2 index number_of_nodes

Set number_of_nodes to 2 if you want to draw spring2 with two nodes, and to 1 if you want to draw spring2 with one node. Default, if control_print_gid_spring2 is not specified, then print_gid_spring2 is used.

6.307 control_print_gid_truss_vector index switch

Same as control_print_gid_beam_vector, however now for the normal force in -truss and -truss_beam elements.

Attention: this control_print_gid_truss_vector is a special plotting option, to get the truss force result as vector plot, with the possibility to influence the direction of the vectors with control_print_gid_truss_vector_normal. Default Tochnog plots the truss force result already as scalar values for each truss element.

6.308 control_print_gid_truss_vector_normal index normal_x normal_y normal_z

Same as control_print_gid_beam_vectors_normal, however now for the normal force in -truss and -truss_beam elements.

6.309 control_print_gmsh index switch

We discuss as an example the printed file naming convention if the input file name is excavation.dat
If \textit{switch} is set to \texttt{-yes} the results are printed into the \texttt{excavation.msh} file. In case the mesh (elements and nodes) have not been printed before in this file, the file will be emptied, and the mesh will be printed. This will also be done if the mesh is changed.

If \textit{index} is 100 and \textit{switch} is set to \texttt{-separate\_index} then the mesh and results are printed in the file is \texttt{excavation\_100.msh}.

If \textit{switch} is set to \texttt{-separate\_sequential} then the mesh and results are printed in the files \texttt{excavation\_0.msh}, \texttt{excavation\_1.msh}, etc. So each time that a \texttt{control\_print\_gmsh} with \texttt{-separate\_sequential} is evaluated a new file is generated with number increased by one.

A dummy point element is put in each node in the gmsh file. Gmsh needs that for plotting vector fields in the nodes. The dummy element group 1234 is used for these dummy point elements. You can suppress these dummy point elements by setting \texttt{control\_print\_gmsh\_dummy} to \texttt{-no}.

All element data starts with \texttt{element\_} in the plots. All node data starts with \texttt{node\_} in the plots.

Scalar data with more then one value is given the extension \_0, \_1 etc. for each of the values. For example the record \texttt{node} (which contains coordinates in each space direction) is plotted as scalar \texttt{node\_0}, \texttt{node\_1} and \texttt{node\_2} which contain the x-coordinate, y-coordinate and z-coordinate respectively. For example the record \texttt{group\_groundflow\_permeability} (which contains permeability in each space direction) is plotted as scalar \texttt{group\_groundflow\_0}, \texttt{group\_groundflow\_1} and \texttt{group\_groundflow\_2} which contain the x-permeability, y-permeability and z-permeability respectively.

For nodes the presence in geometries is plotted as \texttt{node\_geometry\_*}. For elements the presence in geometries is plotted as \texttt{element\_geometry\_*}.

You can plot this file with the program \texttt{gmsh}; see \url{http://www.geuz.org/gmsh}. You can also plot this file with the program \texttt{mecway}; see \url{https://mecway.com/}.

See also \texttt{input\_gmsh}.

6.310 \texttt{control\_print\_gmsh\_dummy} \texttt{index switch}

See \texttt{control\_print\_gmsh}.

Default, if this record is not set and \texttt{print\_gmsh\_dummy} is not specified, \texttt{switch} is set \texttt{-yes}.

6.311 \texttt{control\_print\_gmsh\_element\_data} \texttt{index switch}

If you set \texttt{switch} to \texttt{-yes} data for elements (like element strains, stresses, etc.) is written averaged over the element; this corresponds to \texttt{ElementData} in the gmsh format.

If you set \texttt{switch} to \texttt{-no} this data is written for all element nodes; this corresponds to \texttt{ElementNodeData} in the gmsh format.

Default, if this record is not set, \texttt{switch} is set \texttt{-yes}.

6.312 \texttt{control\_print\_gmsh\_node\_method} \texttt{index method}

If \texttt{method} is set to \texttt{-node}, these coordinates are written. If \texttt{method} is set to \texttt{-node\_start\_refined}, these coordinates are written. If \texttt{method} is set to \texttt{-node\_deformed\_mesh}, these coordinates
are written.

**6.313 control_print_history**

*index data_item_name_0 data_item_index_0 number_0 ...*

Print the time history for each of the sets *data_item_name_0 data_item_index_0 number_0 ...*

For example, if `-node_dof` is used, *number_0* is one of the names of `dof_label` (eg `-velx`).

For example, if `-node_dof_calcul` is used, *number_0* is one of the names of `post_calcul_label` (eg `-aept`).

Otherwise, *number_0* should be an integer specifying the number of the value in the record (for instance number 2 means the third value).

The following history is printed in the file node_dof_112_temp.his

```
control_print_history 0 -node_dof 112 -temp
```

**6.314 control_print_history_relative_time**

*index tr*

If you specify this record, the time printed in the history files will not be the actual time, but instead the actual time minus the relative time *tr*. This is especially convenient when something happens suddenly after a long time, in which case the time-axis in the history plot would be not clear. Using this relative time the time-axis in the plots will become clear (since the long initial time is not part of the time axis).

**6.315 control_print_interface_stress**

*index switch*

**2D analysis**

This option prints in 2D the interface stresses through a set of interfaces starting at place *x_{start}, y_{start}* and ending at *x_{end}, y_{end}* as specified in `control_print_interface_stress_coordinates`. The `switch` needs to be set to `-separate_index` or `-separate_sequential`. The stresses are printed in the file `interface_stress.index`. The first column in the file is the distance from the start point. The following columns contain `interface_sign` and `interface_sigt`. A line is written for each node of each interface element. Crossing interfaces are not allowed. From the start point up to the end point the interfaces needs to be connected without gaps.

**3D analysis**

This option prints in 3D the average interface stresses in the middle of interface elements. The `switch` needs to be set to `-separate_index` or `-separate_sequential`. The interface element middles and average stresses are printed in the file `interface_stress.index`. The first three columns in the file are the coordinates of the middle of the interface element. The following columns contain `interface_sign` and `interface_sigt1` and `interface_sigt2`. A line is written for each interface element. If you specify `control_print_interface_stress_3d_geometry` then only interfaces elements located on the geometry will be printed. If you don’t specify `control_print_interface_stress_3d_geometry` then all interfaces elements will be printed. You can specify the order of printing of the interfaces in the file with `method` in `control_print_interface_stress_3d_order`. If you set `order` to `-x` the interfaces will be ordered according to x-coordinate. If you set `order` to `-y` the interfaces will be
ordered according to y-coordinate. If you set order to -z the interfaces will be ordered according to z-coordinate. If you don’t use control_print_interface_stress_3d_order the interfaces will be ordered according to element number.

6.316 control_print_interface_stress_2d_coordinates
index x_start y_start x_end y_end

For 2D only. See control_print_interface_stress.

6.317 control_print_interface_stress_3d_geometry
index geometry_item_name geometry_item_index

For 3D only. See control_print_interface_stress.

6.318 control_print_interface_stress_3d_order
index order

For 3D only. See control_print_interface_stress.

6.319 control_print_materi_stress_force
index method

This option prints forces and moments as calculated by post_calcul_materi_stress_force. It prints in special purpose ascii files, convenient for further external postprocessing. For example, the name of the file will be materi_stress_force.100 if the index is 100. The files themselves will contain comments explaining the detailed structure of the files.

The method can be set either to -all if all results should printed in the file (so including the averaged results) or to -primary if only the primarily calculated results should be printed in the file (so not including the averaged results).

6.320 control_print_mesh_dof
index switch

See print_mesh_dof.

6.321 control_print_node
index data_item_name number_0 number_1 ...

With this record you can print nodal data like node_dof, node_dof_calcul etc. to files. As an example in 2D you can use control_print_node index -node_dof -velx -velx to get the files velx.index and vely.index; these files contain in columns for all nodes x y velx and x y vely.

For data_item_name you can apply any nodal data record for which the name starts with node. For number_0 number_1 you can specify which parts of the data record should be printed; you can either specify numbers 0, 1, etc. or for node_dof you can specify the names of dof_label like -vely, -vely etc., or for node_dof_calcul you can specify the names of post_calcul_label like -to_pres, -dy_pres etc.
6.322  **control_print_node_angular** index switch_x switch_y switch_z

With this record you can specify that an angle will be included in the files (instead of coordinates). With `switch_x switch_y switch_z` set to `-yes -yes -no` the angle will measure the number of degrees from the positive global x-coordinate directed to the positive global y-direction. With `switch_x switch_y switch_z` set to `-no -yes -yes` the angle will measure the number of degrees from the positive global y-coordinate directed to the positive global z-direction. With `switch_x switch_y switch_z` set to `-yes -no -yes` the angle will measure the number of degrees from the positive global x-coordinate directed to the positive global z-direction. In 1D you cannot use this `control_print_node_angular` record. In 2D you should not specify `switch_z` and you can only use `-yes -yes`.

The middle point of the axes in which the angle is determined should be specified with `control_print_node_angular_middle`. For example in 2D the angle follows from \( \tan(\text{angle}) = \frac{y - y_{\text{middle}}}{x - x_{\text{middle}}} \). In 1D you cannot use this `control_print_node_angular_middle` record. In 2D you should not specify `z_middle` and you should only specify `x_middle y_middle`.

See also `control_print_node`.

6.323  **control_print_node_angular_middle** index `x_middle y_middle z_middle`

See `control_print_node_angular`.

6.324  **control_print_node_geometry** index geometry_item_name geometry_item_index

With `control_print_node_geometry` you can restrict the printing to be done only on nodes located on the specified geometry. See also `control_print_node`.

6.325  **control_print_node_sort** index sort_method

With `control_print_node_sort` you can set if the printed results should be sorted. In case you use `-angular` for `control_print_node_method`, you can set the `sort_method` to `-angle`. Otherwise you can set the sort method to `-x`, `-y` or `-z` (`-y` is only allowed for 2D or 3D, and `-z` is only allowed for 3D). The results will be sorted starting from small values (of the `-angle`, `-x`, `-y` or `-z`) up to high values.

6.326  **control_print_node_zero** index switch

With `control_print_node_zero` you can suppress or activate printing of results with value zero. If you set `switch` to `-yes` then zero valued results will also be printed. If you set `switch` to `-no` then zero valued results will not be printed. Default `switch` is `-yes`. See also `control_print_node`.

6.327  **control_print_number_iterations** index switch

If `switch` is set to `-bf-yes`, Tochnog will print the iteration number while doing equilibrium iterations in a time step. This comes convenient in very large calculations, where you want to monitor the evolution of the calculation.
6.328 control_print_partialname index data_item_name_0 data_item_name_1
...

This printing command is similar to the normal control_print command. With this record, however, all records starting with data_item_name_0, data_item_name_1, etc. will be printed. Thus, for example, control_print_partialname 10 -element will print all records stating with element (as opposed to control_print 10 -element will only print the element record).

6.329 control_print_tecplot index switch

If switch is set to -yes a tecplot plot file is printed, and each time results are added to the same file. You can also set switch to -separate_index; then a new file using the index number will be printed. And also you can set switch to -seperate_sequential; then sequential tecplot files will be printed.

These files contain:

- the primary doffields from node_dof
- post calculated results from node_dof_calcul

Tecplot uses zones to collect data. Zones with nodal results are given names nodal_.... Zones with element averaged results are given names element_averaged_.... Tecplot uses a strandid integer to select which data is visualised. Tochnog generates in the tecplot file this strandid as follows:

- for nodal results the strandid is the group number and extra 1 is placed at the end
- for element averaged results the strandid is the group number and extra 2 is placed at the end

For example for group 100, the strandid is 1001 for nodal results, and the strandid is 1002 for element averaged results.

Tecplot files are less complete as GID files and GMSH files. Tecplot files can be plotted with the tecplot program, a trademark of Amtec Eng., Inc.

6.330 control_print_vtk index switch

Activate printing of results in the Visual Toolkit unstructured grid format, which can be plotted by the paraview plotting program. See www.paraview.org.

For example, if the input file name is excavation.dat and index is 100 and switch is set to -separate_index then results are printed in the file is excavation100.vtk.

For example, if the input file name is excavation.dat and switch is set to -seperate_sequential then results are printed in the files is excavation0.vtk, excavation1.vtk, etc.

In paraview elements are called 'cells' and nodes are called 'points'.

How to get a nice contour plot for the yy-stress:

- File open ..... choose file and hit apply button
• Coloring ..... choose node_materi_stress and set 4 in stead of magnitude

• Edit hit the Choose preset button and select something nice.

• Edit set number of table values to e.g. 80

• Color Legend change legend text etc.

• File Save Screenshot save picture

How to get a vector plots for velocities:

• File open ..... choose file and hit apply button

• Glyph add glyphs for vectors

• Glyph type choose arrow

• Scale mode choose vectors

• Set scale factor choose factor to get nice vector lengths

• Coloring choose node_materi_velocity and choose magnitude

How to find the number of elements depicted in the plot:

• Split the screen at the top right of the layout window, and select spreadsheet view on the second screen

• View and then Selection display inspector

• In the inspector select ID for Cell labels and Point labels

• Activate the small select cells on button in the layout

• With the left mouse button click and drag to select the cells

How to see only elements of a certain groups:

• In Filters select Common and then select Threshold

• In Scalars select element_group

• In Minimum set the minimum group number that you want to see

• In Maximum set the maximum group number that you want to see

• In Coloring select the data that you want to see

6.331 control_print_vtk_coord index switch

If switch is set to -yes the coordinates of nodes is plotted in vtk. If switch is set to -no the coordinates of nodes is not plotted in vtk. Default switch is set to -yes.
6.332 control_print_vtk_dof index initialisation_name_0 initialisation_name_1
 ...

When you specify this record only the solution fields initialisation_name_0, initialisation_name_1 etc will be printed to the vtk files. So the files become smaller in size. This is especially convenient for very large calculations. The names initialisation_name_0, initialisation_name_1 are names from the initialisation part like -condif_temperature, -materi_velocity, -materi_stress etc. In case you do not want any field to be printed in the vtk file use control_print_vtk_dof index -none.

See also control_print_vtk_other.

6.333 control_print_vtk_dof_calcul index calcul_0 calcul_1 ...

When you specify this record only the post fields calcul_0, calcul_1 etc will be printed to the vtk files. So the files become smaller in size. This is especially convenient for very large calculations. See post_calcul_label for the allowed names of calcul_0, calcul_1 etc. In case you do not want any post field to be printed in the vtk file use control_print_vtk_dof_calcul index -none.

See also control_print_vtk_other.

6.334 control_print_vtk_empty index switch

If switch is set to -yes, empty elements are included in the vtk file. If switch is set to -no, empty elements are not included in the vtk file. Default, if control_print_vtk_empty is not specified, switch is set to -yes.

6.335 control_print_gid_group index switch

If switch is set to -yes the element groups are plotted in vtk as result field. If switch is set to -no the element groups are not plotted in vtk as result. Default switch is set to -yes.

6.336 control_print_vtk_node_method index node_type

If method is set to -node, these coordinates are written. If method is set to -node_start_refined, these coordinates are written. If method is set to -node_deformed_mesh, these coordinates are written.

6.337 control_print_vtk_other index switch

If switch is set to -yes also other things like boundary conditions, mesh deformation etc. are printed in the vtk files. If switch is set to -no these other things are not printed in the vtk files. Default switch is set to -yes.

6.338 control_relaxation index relax_0 relax_1 ...

Relaxation parameters for adjusting dof’s in iterations. This can stabilize the calculation. For example, a relaxation parameter of 0.1 means that the corresponding dof is not completely updated
with the iterative change, but only 10 percent of the change is actually applied in a iteration.

If enough iterations are used, the relaxation parameters with not influence the final solution.

You should specify a relaxation parameter term for each principal dof which is present in the calculation (see the start of the data part description for a list of principal dof’s; these are velocities, temperature, etc.).

This relaxation done for timestep records with the same index. See also relaxation.

6.339 control_repeat index number_of_repeats control_index

If number_of_repeats is larger than 0 the calculation repeats from the control_index. The value of number_of_repeats is decreased by 1.

A first application is to do many time steps, but print only once in a while:

```
control_timestep 10 1. 100. 
control_print 20 -node_dof 
control_repeat 30 80 10
```

In the latter example, first 100 timesteps are taken, then results for node_dof are printed; this is repeated 80 times.

Also, this control_repeat can typically be used to perform a number of refinements combined with time stepping to a new, refined, solution. This is done a fixed number of times.

In case the repeat jumps back to a control_timestep record for which the index equals control_index, then that the previous timestep will be used (instead of the timestep specified by the control_timestep record).

See also control_repeat_until_item.

6.340 control_repeat_save index data_item_name_0 data_item_index_0 data_item_number_0 data_item_name_1 data_item_index_1 data_item_number_1 ...

This record specifies data that should be saved while repeats are performed with control_repeat. The saved results are stored in the records repeat_save_result (subsequent repeats write in subsequent indices of repeat_save_result).

6.341 control_repeat_save_calculate index switch

Perform a statistical analysis on data of repeat_save_result. The statistical results are placed in repeat_calculate_result. The average value and variance will be calculated.

6.342 control_reset_dof index dof_0 dof_1 ...

The dof’s as specified in this record are set to a some new value. For example, dof_0 is -sigxx, etc. As a special option you can use -all to reset all dof’s.
With **control_reset_value_constant** you can specify the new value to which the dof's should be set. Additionally you can specify values depending on space coordinates with **control_reset_value_linear** etc. The records **control_reset_value_constant, control_reset_value_linear** etc. can be arbitrarily combined so that complex dependency of the value of space coordinates is possible. If none of these records is specified then a new value 0 is used.

As a typical example, you can set displacements and strains to zero in a geotechnical calculation, with an *updated* material description, after the gravity load has been applied. In this way the strains for further deformations can be distinguished more clearly.

The dof's will be reset on all nodes of elements which are completely inside the geometry specified in **control_reset_geometry**, or of elements which have all their nodes specified in **control_reset_node**.

As a special option for groundwater calculations, you can set a dof to **total_pressure** to reset the physical groundwater pore pressure (total pressure).

Attention: this **control_reset_dof** should not be used to reset displacements (or velocity integrated) if also **support_edge_normal** is present. This is because those **support_edge_normal** supports calculate forces directly from total displacements, and so you would in fact set the support forces also to zero. Also in the presence of interface elements the displacements (or velocity integrated) should not be reset. Normal isoparametric elements use an incremental formulation for stresses however (new stress = old stress + incremental stress from stiffness), so that resetting displacements to zero does not influence the stresses.

Attention: with this **control_reset_dof** option you cannot reset the strains, stresses, forces, etc. in structural elements (springs, interfaces, trusses, ...) This option only works for isoparametric elements.

### 6.343 control_reset_element_group

```
index element_group_number_0 element_group_number_1 . . .
```

Specifies the specific element groups on which the **control_reset_dof** record with the same index should be applied. If this record is not specified, the **control_reset_dof** record will be done for all element groups (in the specified geometry).

### 6.344 control_reset_geometry

```
index geometry_item_name geometry_item_index
```

Specifies the geometry on which the **control_reset_dof** record with the same index should be applied.

### 6.345 control_reset_interface

```
index geometry_item_name geometry_item_index
```

Reset all interface data like strains, stresses, etc. to 0 for interface elements located in the geometry with name **geometry_item_name** and index **geometry_item_index**.

### 6.346 control_reset_interface_strain

```
index geometry_item_name geometry_item_index
```

Reset all interface strains to 0 for interface elements located in the geometry with name **geometry_item_name** and index **geometry_item_index**. The interface stresses at this moment of resetting will be remembered by Tochnog. In the next time steps the new interface strains start with
0, and change when the interfaces deform further. And in the next time steps the new interface stresses are calculated from the interface stresses at this moment of resetting plus stress due to additional deformation (from the specified stiffness properties).

6.347 control_reset_node index node_0 node_1 ...  

Specifies nodes on which the control_reset_dof record with the same index should be applied.

6.348 control_reset_value_constant index value  

Specifies the value to which dof’s of the control_reset_dof record are reset. A constant value will be used.

6.349 control_reset_value_exponent index a_x b_x c_x d_x e_x a_y b_y c_y d_y e_y a_x b_x c_x d_x e_x 

Specifies the exponential space distribution to which dof’s of the control_reset_dof record are reset. The dependency \( a_x e^{b_x c_x d_x e_x} + a_y e^{b_y c_y d_y e_y} + a_z e^{b_z c_z d_z e_z} \) will be used. In 1D only \( a_x b_x c_x d_x e_x \) should be specified. In 2D only \( a_x b_x c_x d_x e_y a_y b_y c_y d_y e_y \) should be specified.

6.350 control_reset_value_linear index a_x a_y a_z  

Specifies the linear space distribution to which the dof’s of the control_reset_dof record are reset. The dependency \( a_x x + a_y y + a_z z \) will be used. In 1D only \( a_x \) should be specified. In 2D only \( a_x a_y \) should be specified.

6.351 control_reset_value_logarithmic_first index a_x b_x c_x d_x e_x a_y b_y c_y d_y e_y a_x b_x c_x d_x e_x 

Specifies the logarithmic space distribution to which dof’s of the control_reset_dof record are reset. The dependency \( a_x \ln(b_x c_x d_x e_x) + a_y \ln(b_y c_y d_y e_y) + a_z \ln(b_z c_z d_z e_z) \) will be used. In 1D only \( a_x b_x c_x d_x e_x \) should be specified. In 2D only \( a_x b_x c_x d_x e_y a_y b_y c_y d_y e_y \) should be specified.

6.352 control_reset_value_logarithmic_second index a_x b_x c_x d_x e_x f_x g_x a_y b_y c_y d_y e_y f_y g_y a_x b_x c_x d_x e_x f_x g_x 

Specifies the logarithmic space distribution to which dof’s of the control_reset_dof record are reset. The dependency \( (a_x + b_x) \ln(d_x) + g_x + (a_y + b_y) \ln(d_y) + g_y + (a_z + b_z) \ln(d_z) + g_z \) will be used. In 1D only \( a_x b_x c_x d_x e_x f_x g_x \) should be specified. In 2D only \( a_x b_x c_x d_x e_x f_x g_x a_y b_y c_y d_y e_y f_y g_y \) should be specified.

6.353 control_reset_value_multi_linear index z_0 value_0 z_1 value_1 ...  

Specifies the multi-linear space distribution in vertical direction to which the dof’s of the control_reset_dof record are reset. A multilinear table of value versus z coordinate should be given; at \( z_0 \) the value is \( value_0 \) etc. The \( z_0, z_1 \) etc. should have increasing values from low to high; the values should cover all coordinates in the FE mesh for with the reset is done. In 1D not a z coordinate but x coordinate is used instead. In 2D not a z coordinate but y coordinate is used instead.
6.354  control_reset_value_power  index  \(a_x b_x a_y b_y a_z b_z\)

Specifies the power space distribution to which the dof’s of the control_reset_dof record are reset. The dependency \(a_x x^{b_x} + a_y y^{b_y} + a_z z^{b_z}\) will be used. In 1D only \(a_x b_x\) should be specified. In 2D only \(a_x b_x a_y b_y\) should be specified.

6.355  control_reset_value_method  index  method

If switch is set to -multiply the values as specified by control_reset_value etc. are used as relative factor by which the dof’s are changed. So for example if 0.1 is given in control_reset_value_constant, then the dof’s will be multiplied with 0.1.

If switch is set to -add the values as specified by control_reset_value etc. are used as additional values by which the dof’s are changed. So for example if 0.1 is given in control_reset_value_constant, then the dof’s will be added with 0.1.

6.356  control_reset_value_square_root  index  \(a_x b_x c_x a_y b_y c_y a_z b_z c_z\)

Specifies the power space distribution to which the to which dof’s of the control_reset_dof record are reset. The dependency \(a_x \sqrt{b_x} + c_x x^{b_x} + a_y \sqrt{b_y} + c_y y^{b_y} + a_z \sqrt{b_z} + c_z z^{b_z}\) will be used. In 1D only \(a_x b_x\) should be specified. In 2D only \(a_x b_x a_y b_y\) should be specified.

6.357  control_restart  index  switch

If switch is set to -yes then the calculation continues with the undeformed mesh. The dof’s (in the node_dof records) are reset to the initial values. And time_current is set to the initial time.

This allows you to calculate some path dependent behavior completely from the start with a refined mesh.

6.358  control_safety_slip  index  switch

If switch is set to -yes a slip safety factor calculation will be performed with the method as described in [4]. The calculated safety factor \(F_s\) is:

\[
F_s = \frac{\int \tau_{mc} dA}{\int \tau dA}
\]

where \(\tau_{mc}\) is the maximum possible shear stress according to the mohr-coulomb condition using the actual normal stress, \(\tau\) is the actual shear stress and \(dA\) is the surface area in the integral. The advantage of this safety factor definition is that it can be evaluated at any stress state, for example the gravity stress state, without any further timesteps with friction angle and cohesion reduction. The definition simply compares the actual current shear stress relative to the maximum possible shear stress following from mohr-coulomb and the current normal stresses.

The user needs to specify over which surface the integration of the safety factor needs to be performed. See safety_slip_circle_grid_*, etc.

A critical slip surface will be calculated for each set of safety_slip_circle_grid_*, etc. (thus for each separate index of these a critical surface will be calculated). You can specify also
safety_slip_set however, which defines the indices of safety_slip_circle_grid_*, etc. belonging to a specific set. The overall minimal safety factor will be determined for all safety geometries belonging to the set.

This control_safety_slip is available for group_materi_plasti_mohr_coul, group_materi_plasti_mohr_coul_direct, group_materi_plasti_druck_prag and group_materi_plasti_hypo_wolffersdorff.

As a special option you can set the switch not to -yes but to a number 1, 2, 3, .. instead. Then this number 1, 2, 3, .. is used by tochnog as the number of automatic safety calculations of the critical slip surface. For example if you use slip circles (specified by middle points and radii) after the first safety calculations a specific middle point and radius will have the lowest safety factor. Then in the next safety calculation tochnog will reduce the area of middle points and the set of radii to a smaller zone around that critical middle point and radius. With this smaller zone a new safety analysis will lead to a new critical middle point and radius somewhere in the reduced zone. Then again a smaller zone will be used, leading to again a new critical middle point and radius, etc. etc. This repetition of reducing the zone of middle points and radii will done such many times as set in the number, so 1, 2, 3, .. Typically the number 2 could be used.

Slip surfaces will be drawn in GID plots (see control_print_gid for GID plotting). For each slip surfaces the safety factor can be plot. Moreover, also a local safety factor can be plot, which is the local ratio of shear stress and maximum possible shear stress.

Slip surfaces crossing a boundary with prescribed displacements (or velocities) non valid since the slip velocities are in general not compatible with the prescribed velocities on such boundary.

6.359 control_slide_damping_apply index switch

If switch is set to -yes then any slide_damping records will be applied. If switch is set to -no then any slide_damping records will be not applied. Default if control_slide_damping_apply is not specified then switch is -yes.

6.360 control_slide_plasti_apply index switch

If switch is set to -yes then any slide_plasti_* records will be applied. If switch is set to -no then any slide_plasti_* records will be not applied. Default if control_slide_plasti_apply is not specified, plasti_apply or control_plasti_apply will be used.

6.361 control_slide_stiffness_apply index switch

If switch is set to -yes then any slide_stiffness records will be applied. If switch is set to -no then any slide_stiffness records will be not applied. Default if control_slide_stiffness_apply is not specified then switch is -yes.

6.362 control_solver index solver_type

If solver_type is set to -diagonal then only the main diagonal of the system matrix will be used for the solution of all dof's. This gives the program an explicit like structure. In fact, if control_timestep_iterations is set to 1, then a classical explicit finite element program is obtained.
If `solver_type` is set to `-matrix_iterative_bicg` then the complete system matrix will be used for solution of the principal dof’s (see the initialization section for an explanation on principal dof’s). A diagonal Preconditioned Biconjugate Gradient method is applied.

If `solver_type` is set to `-matrix_pardiso` then the pardiso solver will be used for solution of the principal dof’s.

If `solver_type` is set to `-none` then only the matrices and right-hand sides are setup, but the equations are not really solved.

6.363 **control_solver_bicg_error** *index error*

With `error` you set the termination error ratio between the initial and final error in the bicg iterations. Default `error` is set to $1.e-13$.

See also `solver_bicg_error`. This `control_solver_bicg_error` record overrules `solver_bicg_error` if both are specified.

6.364 **control_solver_bicg_restart** *index nrestart*

With `nrestart` you set the number of restarts in the bicg iterations. Default `nrestart` is set to 0.

See also `solver_bicg_restart`. This `control_solver_bicg_restart` record overrules `solver_bicg_restart` if both are specified.

6.365 **control_solver_bicg_stop** *index switch*

If `switch` is set to `-yes`, the calculation is stopped if the bicg solver does not converge. If `switch` is set to `-no`, the calculation is not stopped if the bicg solver does not converge. Default `switch` is set to `-yes`.

See also `solver_bicg_stop`. This `control_solver_bicg_stop` record overrules `solver_bicg_stop` if both are specified.

6.366 **control_solver_matrix_save** *index switch*

If `switch` is set to `-yes`, the solver saves and applies the decomposed matrix, but not in case Tochnog thinks for some reason that the matrix needs to be decomposed at each timestep. This can save CPU time, since further decompositions of the matrix are not required anymore (only backsubstitution to find the solution vector).

If `switch` is set to `-no`, the solver does not save the decomposed matrix.

If `switch` is set to `-always`, the solver saves and applies the decomposed matrix, even in case Tochnog thinks for some reason that the matrix needs to be decomposed at each timestep.

This option is only available in combination with the pardiso solver.

Side remark: Tochnog mostly uses a linear matrix in iterations (no plasticity effect in the matrix). Only in special cases like hypoplasticity, user supplied routines, etc. the current stiffness matrix is used.
6.367 control_solver_pardiso_out_of_core index switch

If switch is set to -yes the pardiso solver is called with a 'out of core' option. See for further the pardiso solver in the the intel mkl library. Default switch is -no.

6.368 control_solver_pardiso_ordering index ordering

Set the number ordering to one of the following:

- 0 The minimum degree algorithm.
- 2 The nested dissection algorithm from the METIS package.
- 3 The parallel (OpenMP) version of the nested dissection algorithm.

Default ordering is 3. For more information see pardiso info at intel.

6.369 control_support_edge_normal_damping_apply index switch

If switch is set to -yes then all support_edge_normal_damping records will be applied. If switch is set to -no then all support_edge_normal_damping records will not be applied. Default, if control_support_edge_normal_damping_apply is not specified, then switch is set to -yes.

6.370 control_support_edge_normal_stiffness_freeze index switch

If switch is set to -yes, tochnog freezes the stiffness forces generated by support_edge_normal. The stiffness forces remain at their present value and will not change anymore. A typical application is earthquake or vibration analysis where you first impose gravity including stiffness at supports, then freeze the forces at the supports, and then in the earthquake or vibration analysis use only damping at the supports to model absorbing boundaries which absorb further force changes at the boundaries.

```plaintext
( support properties )
support_edge_normal 10 ...
support_edge_normal_damping 10 ...
...
( calculate gravity stresses )
control_timestep 10 ...
control_support_edge_normal_damping_apply 10 -no
...
( freeze stiffness forces at boundary )
control_support_edge_normal_stiffness_freeze 20 -yes
...
( calculate earthquake or vibrations )
control_timestep 30 ...
control_support_edge_normal_damping_apply 30 -yes
control_inertia_apply 30 -yes
```
6.371 control_system_call index integer_value

Specifying this record tochnog calls a system command. You need to program that command yourself. On Linux provide a tochnog_system_call.sh file which is executable. On MS Windows provide a tochnog_system_call.bat file.

In the command you can place commands that you want to be executed. For example, if you put in the Linux file the command date >> system_call.out you get the output of the date command appended to system_call.out. Another example is sending you an automatic email indicating that the calculation reached a certain point or is almost finished.

The command is called with integer_value as first argument. You can use this integer value in your command (eg by using $1 in the Linux shell script command).

This command is executed at the end of each timestep and also at the end of each control index.

6.372 control_timestep index step_size time_increment step_size time_increment ...

These records define sets of time steps of size step_size which are to be taken till the time is increased by time_increment. In the example below time steps of 0.1 are taken from time 0.0 up to time 1.0. Then time steps of 0.2 are taken up to time 2.0

control_timestep 0 0.1 1. 0.2 1.

6.373 control_timestep_adjust_minimum_iterations index switch

If switch is set to -yes Tochnog will increase the minimum number of iterations in a timestep if it thinks that is helpful for the specific input file that you are running; this is done in combination with control_timestep_iterations_automatic. If switch is set to -no Tochnog will not do so, and keep 2 as the minimum number of iterations. Default, if control_timestep_adjust_minimum_iterations is not specified, switch is set to -yes.

6.374 control_timestep_iterations index number_of_iterations

This sets a fixed number of equilibrium iterations in each time step (for time steps of the control_timestep record with the same index). For many iterations, the time stepping is Euler implicit. For few iterations the time stepping becomes explicit. Default number_of_iterations is 2.

In dynamic analysis, with the default number of 2 iterations you gain numerical stability, at the expense of numerical damping however. To prevent this numerical damping use 1 iteration instead.

As an alternative, you can use control_timestep_iterations_automatic.

6.375 control_timestep_iterations_automatic index ratio_criterium minimal_timestep maximum_timestep

After specification of this record, iterations will be performed until ratio in post_node_rhside_ratio is less than ratio_criterium. Typically, set ratio_criterium to 0.001.
The time step size is increased if the number of iterations is substantially lower than the wished (preferred) number of iterations. The time step size is decreased if the number of iterations is substantially larger than the wished (preferred) number of iterations.

The time step specified in `control_timestep` is used as initial step. The time step is not allowed to become higher than `maximum_timestep`. The time step is not allowed to become lower than `maximum_timestep`.

The initial step as specified in `control_timestep`, should be sufficiently small so that this automatic algorithm can fulfill the `ratio_criterium` in that initial step.

After the iterations in a step are finished, Tochnog performs one extra iteration to update strains, stresses, etc with the last velocity fields. In this extra iteration also the `post_node_rhside_ratio` will be recalculated, and thus may become different from the previous value that was used to determine if the iterations should be stopped.

See also `control_timestep_iterations_automatic_stop`, and `control_timestep_iterations_automatic_minimum_maximum_wished`.

6.376 `control_timestep_iterations_automatic_minimum_maximum_wished` index minimum_iterations maximum_iterations wished_iterations

This sets the minimum number of allowed iterations, the maximum number of allowed iterations, and the wished (preferred) number of iterations for the automatic time stepping mechanism as specified by `control_timestep_iterations_automatic` with the same index. The default for this record is 2 8 4. The maximum number of allowed iteration should be 2 or larger.

6.377 `control_timestep_iterations_automatic_stop` index switch

If you set `switch` in `control_timestep_iterations_automatic_stop` to -yes then the calculation does stop if the minimal timestep size is reached. If you set `switch` in `control_timestep_iterations_automatic_stop` to -no then the calculation does not stop if the minimal timestep size is reached, and the present time stepping will be finished. If you set `switch` in `control_timestep_iterations_automatic_stop` to -continue then the calculation does not stop if the minimal timestep size is reached, and the present time stepping will not be finished. Default, if `control_timestep_iterations_automatic_stop` is not specified, then `switch` is set to -yes.

6.378 `control_timestep_multiplier` index multiplier

If this record is specified, each new time step size is `multiplier` * old time step size. The `step_size` as specified in `control_timestep` will only be used as the initial time step.

This option is handy to study physical processes which develop more slowly when time proceeds. A typical example is consolidation analysis in geotechnics.

6.379 `control_timestep_until_data` index data_item_name_0 data_item_index_0 data_item_number_0 data_item_name_1 data_item_index_1 data_item_number_1 ... 

With this record you can specify conditions for which the timesteps with the same index should be stopped. For each specified data item name, index and number you can specify a minimum value in
control_timestep_until_minimum and a maximum value in control_timestep_until_maximum. A typical example:

control_timestep 10 ...
control_timestep_until_data 10 -post_point_dof 3 -velx
control_timestep_until_minimum 10 -120.
control_timestep_until_maximum 10 +120.

6.380 control_timestep_until_maximum index maximum_0 maximum_1

\ldots

See control_timestep_until_data.

6.381 control_timestep_until_minimum index minimum_0 minimum_1 \ldots

See control_timestep_until_data.

6.382 control_truss_rope_apply index switch

If switch is set to -no, any truss rope data in the input file will be ignored. This is done for timestep records with the same index.

This option is convenient for testing your input file just linear, without the need to outcomment each and every part with truss rope data. See also truss_rope_apply.

6.383 control_zip index switch

If switch is set to -yes all *flavia*, *msh, vtk, *.plt and *dbs files are zipped with the gzip program. The gzip program should be installed on your computer.

This comes convenient in large calculation with lots of output, where you want to use results later and save disk space during the calculation.

6.384 convection_apply switch

If switch is set to -yes, the convection of a material with respect to the mesh is allowed. If switch is set to -no, the convection of a material with respect to the mesh is not allowed. This is done for all timesteps.

Default switch is set to -no. See also control_convection_apply.

6.385 convection_stabilization switch

Because of finite discretisation sometimes unrealistic results may be obtained (wiggles, etc.). If switch is set to -yes results are stabilized with a minimal amount of artificial diffusion. If switch
is set to **-maximal** results are stabilized with a maximal amount of artificial diffusion. If *switch* is set to **-no** results are not stabilized.

Default, if **convection_stabilization** is not specified, *switch* is set to **-yes**.

### 6.386 data_activate

`index data_item_name_0 data_item_name_1 ... switch`

With this record you can set data items to become activated if *switch* is set to **-yes** or de-activated if *switch* is set to **-no**. The `data_item_name` specifies a data record name.

### 6.387 data_activate_time

`index time`

Time point at which the record `data_activate` with the same index is evaluated. If this record is not specified, the `data_activate` is evaluated at the start of the calculation.

### 6.388 data_delete

`index data_item_name index_range`

Similar to **control_data_delete**, but now not as control record however.

### 6.389 data_delete_time

`index time`

Time point at which the record `data_delete` with the same index is evaluated. If this record is not specified, the `data_delete` is evaluated at the start of the calculation.

### 6.390 data_ignore

`data_item_name`

With this option Tochnog will ignore all data items with name `data_item_name`. This `data_ignore` can also be set as environment symbol. As a typical example you can set the environment symbol `data_ignore` to **-print_apply**. Then when you run the tochnog example tests all **-print_apply** `-no` in the tests will be ignored, so you you get all printed output for the tests.

### 6.391 dependency_apply

`switch`

If *switch* is set to **-yes**, dependencies like specified in **dependency_diagram** and **dependency_item** are included in the calculation. If *switch* is set to **-no**, these dependencies are not included. This is done for all timestep records.

Default *switch* is set to **-yes**. See also **control_dependency_apply**.

### 6.392 dependency_diagram

`index dof_value_0 ... data_item_value_0 ...`

See **dependency_item**.
6.393 dependency_method index method

See dependency_item.

6.394 dependency_geometry index geometry_item_name geometry_item_index

See dependency_item.

6.395 dependency_item index data_item element_group dofn

This record allows you to make an element data item group_ * dependent on one of the dof’s, see dof_label for dofnames, or on one of the post calculation results, see post_calcul_label for post calculation names. This is done for n values of the dof(n should be at least 2). The dependency should be specified in the dependency_diagram record (same index) with a multi linear diagram. In the diagram first a set of dof’s values should be specified. Second the set of data item values for those dof values should be specified. Some examples are given below.

Temperature dependent Young’s modulus of element_group 1 ( E = 1.e10 at temperature 1, etc.):

dependency_item 1 -group_materi_elasti_young 1 -temp 4
dependency_diagram 1 1.2.3.4.1.e10 1.e9 1.e8 3.e5

Temperature dependent Young’s moduli in two maxwell chains of element_group 1 ( for the first chain the moduli 1.e10, 1.e9,… for the second chain the moduli 1.e12, 1.e11,… all relaxation times are 1.10^-2. ):

dependency_item 1 -group_materi_maxwell_chain 1 -temp 4
dependency_diagram 1 1.2.3.4.1.e10 1.e9 1.e8 3.e5
1.e-2 1.e-2 1.e-2 1.e-2
1.e12 1.e11 1.e10 3.e7
1.e-2 1.e-2 1.e-2 1.e-2

As a special option, dof can be set to -time_current. This allows for time-dependent properties (aging). The example below shows time dependent Young’s modulus of element_group 1 ( E = 1.e10 at time 0, etc.):

dependency_item 1 -group_materi_elasti_young 1 -time_current 4
dependency_diagram 1 0.1.2.3.1.e10 1.e9 1.e8 3.e5

As a special option, element_group can be set to -all, so that the dependency diagram will be used for all groups.

As another special option, dof can be set to -x, -y or -z. This allows for dependency on one of the space coordinates. The example below shows a von-mises stress dependent on the z-coordinate for element_group 1:
In 1D only -x can be used, in 2D only -x and -y can be used, and in 3D all of -x, -y and -z can be used.

The dependencies are available only for real precision data (and thus not for integer data). The dependency_diagram values should be specified from low to high values for the dof.

The dependency_method can be set to either -use or -multiply; with -use you specify that the values of dependency_diagram will overwrite specified values for the data item; with -multiply you specify that the values of dependency_diagram will multiply specified values for the data item; default, if dependency_method is not specified, -use will be used.

With the dependency_type record you can require that the cosinus, sinus or tangent of a data value is used in the dependency (in stead of the data value directly itself). The type can be set to either -cosinus, -sinus or -tangent. This is typically convenient for geotechnical safety factor calculations where you want that for a mohr coulomb law the cohesion and tangent of the friction angle are decreased at the same ratio in time. If you don’t specify dependency_type the value itself will be changed. To be clear we give the following four examples. If dependency_method is set to -use and dependency_type is not specified, then the value specified in the dependency diagram will be used for the data. If dependency_method is set to -use and dependency_type is set to -tangent, then the arc-tangent of the value specified in the dependency diagram will be used for the data. If dependency_method is set to -multiply and dependency_type is not specified, then the value specified in the dependency diagram will be multiplied with the original value for the data, and the result will be used as new value for for the data. If dependency_method is set to -multiply and dependency_type is set to -tangent, then the value specified in the dependency diagram will be multiplied with the tangent of the original value for the data, the arc-tangent of the result will be taken, and the final result will be used as new value for for the data.

With the dependency_number record you can require that you only want to make one specific number of the data (0 for the first value, 1 for the second value, etc.) dependent; in this case, you should specify only that specific value in dependency_diagram. If you don’t specify dependency_number, then all values of the record are made dependent, and thus all values should be specified in dependency_diagram.

The dependency_geometry can be set to select a geometry for which the dependency is valid; outside the geometry the dependency will not be used; default, if dependency_geometry is not specified, no geometry selection will be used.

The following gives as example lowering the tangent of the mohr coulomb friction angle with a factor in time, for the elements of all groups within a radius distance from a point:

```
geometry_point 10 . . .
.
.
dependency_item 1 -group_materi_plasti_mohr_coul -all -time_current 2

dependency_number 1 0 (only for the friction angle)
dependency_method 1 -multiply (use specified diagram as multiplication factor)
dependency_type 1 -tangent (for the tangent, so not for the value itself)
dependency_diagram 1 10. 11. 1. 0. (lower the tangent of friction angle between time 10 to time 11 from original value to 0)
dependency_geometry 1 -geometry_point 10 (do that only within a certain radius of a point)
```
You can use `print_group_data` to get the result for the calculated values using the dependency diagram. In fact, most `group_*` records can be used in the dependency diagram, but not all. Thus checking if things go like you want with the `print_group_data` is strongly advised.

6.396 dependency_number index number

See dependency_item.

6.397 dependency_type index type

See dependency_item.

6.398 dof_element_dof dof_per_element_0 dof_per_element_1 ...

This record is for printing only. It is not meant as user input record. After the calculation the `dof_per_element_0, dof_per_element_1` etc. contain a `-yes` or `-no`. In case a dof is default calculated per element, so the field is non-continuous, a `-yes` is set. In case a dof is default calculated as continuous field a `-no` is set. This default calculation can be overruled by `global_element_dof_apply`.

6.399 dof_label dof_0 dof_1 ...

This record will be filled with labels of the dof’s in the correct order. This information is required for understanding records like `node_dof` etc. The sequential order for the primary dof’s will match the order in which they are specified in the initialization part.

The total list of possible doflabels is:

- `-accx` acceleration in x-direction, `-accy`, `-accz`,
- `-cchis0, -cchis1` cam clay history variables,
- `-dam` damage,
- `-dens` density,
- `-dipriscohisv, -dipriscohis1, ...`, di prisco plasticity history variables,
- `-disx` displacement in x-direction, `-disy`, `-disz`,
- `-rdisx` relative displacement in x-direction, `-rdisy`, `-rdisz`,
- `-ener` material strain energy,
- `-epexx` xx-strain elastic, `-epexy`, `-epexz`, `-epeyy`, `-epeyz`, `-epezz`,
- `-epixx` xx-strain intergranular, `-epixy`, `-epixz`, `-epiyy`, `-epiyz`, `-epizz`,
-episa_cxx xx-strain is intergranular, -episa_cxy, -episa_cxz, -episa_cyy, -episa_cyz, -episa_czz,

-episa_eacc is intergranular accumulated strain,

eppxx xx-strain plastic, -eppxy, -eppxz, -eppyx, -eppyz, -eppyz, -eppzz,

eppcaxx xx-strain plastic cap model, -eppcaxy, -eppcaxz, -eppcayx, -eppcayz, -eppcazz,

eppcoxx xx-strain plastic compression model, -eppcox, -eppcox, -eppcoy, -eppcoyz, -eppcozz,

eppdixx xx-strain plastic diprisko model, -eppdixy, -eppdixz, -eppdiyy, -eppdiyz, -eppdizz,

eppdrxx xx-strain plastic druckprag model, -eppdryx, -eppdrxz, -eppdryy, -eppdryz, -eppdrzz,

eppgencamxx xx-strain plastic generalised non associate cam clay for bonded soils model, -eppgencamxy, -eppgencamxz, -eppgencamy, -eppgencamy, -eppgencamzz,

-epphaxx xx-strain plastic hardsoil model, -epphaxx, -epphaxx, -epphayx, -epphayx, -epphazz,

eppmoxx xx-strain plastic mohr-coulomb model, -eppmoxy, -eppmoxz, -eppmoyy, -eppmoyz, -eppmozz,

-epptexxx xx-strain plastic tension model, -epptexy, -epptexz, -epptexy, -eppteyz, -epptezz,

eppvoxx xx-strain plastic von-mises model, -eppvoxy, -eppvoxz, -eppvoyy, -eppvoyz, -eppvozz,

eppmolcxx xx-strain mohr-couloob model for all laminates, -eppmolxy, -eppmolxz, -eppmolyy, -eppmoly, -eppmolzz,

eppmolkxx xx-strain mohr-couloob model laminate k=0,...,5, -eppmolkxy, -eppmolkxz, -eppmolkyy, -eppmolkyz, -eppmolkzz,

-epptekxx xx-strain tension model for all laminates, -epptelx, -epptelx, -epptely, -epptely, -epptelzz,

-epptekkxx xx-strain tension model laminates k=0,...,5, -epptelho, -epptelho, -epptely, -epptely, -epptelzz,

-eptxx xx-strain total, -eptxy, -eptxz, -eptyy, -eptyz, -eptzz,

-f plasticity yield rule,

-fn nonlocal plasticity yield rule,

-fscal time derivative of scalar,

gvelx ground water velocity in x-direction, -gvely, -gvelz.

-hisv0, -hisv1, ..., material history variables,

-kap plastic hardening parameter kappa,

-kapsh shear plastic hardening parameter kappa,
-phimob mobilized friction angle plasticity in degrees,

-pres hydraulic pressure head,

-presgradx gradient hydraulic pressure head in x direction, -presgrady, -presgradz

-rhoxx xx plastic kinematic hardening, -rhoxy, -rhoxz, -rhooy, -rhoyz, -rhozz,

-rotx rotation around x-direction, -roty, -rotz,

-scal scalar,

-sigxx xx-stress, -sigxy, -sigxz, -sigyy, -sigyz, -sigzz,

-sigmkxx xx-stress in the k-th maxwell chain, -sigmkxy, -sigmkxz, -sigmkyy, -sigmkyz, -sigmkzz,

-strtokap total strain hardening parameter,

-strtocokap compression part of total strain hardening parameter,

-strtoshkap shear part of total strain hardening parameter,

-strtotekap tension part of total strain hardening parameter,

-temp temperature,

-velx velocity in x-direction, -vely, -velz,

-velix integrated velocity in x-direction, -veliy, -veliz,

-void material void fraction.

-work material second order work.

Furthermore, -xvelx denotes the spatial x-derivative of -velx in x-direction, etc.. Finally, -tvelx denotes the first time derivative of -velx, etc.. The time derivative and the space derivatives are only calculated if derivatives is included in the initialization part.

For example, the following might be seen after a print of the database

    echo -yes
    number_of_space_dimensions 2
    derivatives
    condif_temperature
    end_initia
    ...
    dof_label -temp -xtemp -ytemp -ttemp
    ...

Or, for example, the following might be seen after a print of the database
6.400  **dof_limit**  `lower_dof_0`  `upper_dof_0`  `lower_dof_1`  `upper_dof_1`  ...

With this record you can specify the lower and upper allowed values for all primary dof’s. With `lower_dof_0` you specify the lower allowed value for the first dof. With `upper_dof_0` you specify the upper allowed value for the first dof. Etc.

6.401  **dt ime**  `dt`

This record contains after the calculation the last timestep used in the calculation. This record is meant for printing only.

6.402  **element**  `index`  `element_name`  `node_0`  `node_1`  `node_2`  ...

Nodal connective of element `index`. In 1D, `element_name` is `-bar2` (2 noded bar), `-bar3`, `-bar4`. In 2D, `element_name` is `-tria3` (3 noded triangle), `-tria6` (6 noded triangle), `-quad4` (4 noded quadrilateral), `-quad6` (6 noded quadrilateral, 2 sides of 3 nodes), `-quad8`, `-quad9`, `-quad16`. In 3D, `element_name` is `-tet4` (4 noded tetrahedral), `-prism6` (6 noded prismatic), `-prism12` (12 noded prismatic), `-prism15` (15 noded prismatic), `-prism18` (18 noded prismatic), `-tet10` (10 noded tetrahedral), `-hex8` (8 noded hexahedral), `-hex18` (18 noded hexahedral, 2 sides of 9 nodes), `-hex20` (20 noded hexahedral, not formally available yet, still being tested, use with care), `-hex27`.

Further possibilities for `element_name` are: `-spring2` (2 noded spring), `-contact_spring1` (1 noded contact element), `-contact_spring2` (2 noded contact element), the two nodes may have the same position in space. `-truss` (truss element), `-beam` (beam element), `-truss_beam` (combined truss-beam element).

Some of the elements are drawn below.
See also: group_type and group_integration_points.

6.403  element_beam_direction

\[ \text{index dir}_x, \text{dir}_x, \text{dir}_z, \text{dir}_y, \text{dir}_y, \text{dir}_z \]

After the calculation, this record will be filled with the direction of a beam in space. The first three values give the direction of the local beam x direction, that is the beam torsion axis. The second three values give the direction of the local beam y direction, that is the beam y bending axis. The third three values give the direction of the local beam z direction, that is the beam z bending axis.

The index specifies the beam element number.

6.404  element_beam_direction_z

\[ \text{index dir}_z, \text{dir}_z, \text{dir}_y, \text{dir}_z \]

The index specifies the beam element number.

Sate as group_beam_direction_z, but now per element however.
6.405  **element_beam_force_moment**  
*index force_x_first_node force_y_first_node  
force_z_first_node moment_x_first_node moment_y_first_node moment_z_first_node  
force_x_second_node force_y_second_node force_z_second_node moment_x_second_node  
moment_y_second_node moment_z_second_node . . .*

After the calculation, this record will be filled with the forces and moments of a beam in the local 
beam axes \(x, y, z\).

The *index* specifies the beam element number.

Attention: the values at the first node have a minus in their definition as compared with the values 
in the second node. For example in a beam number 20 with constant \(z\) moment of 10 you will find:

```
element_beam_force_moment 20 0. 0. 0. -10. 0. 0. 0. 0. 10.
```

6.406  **element_boundary**  
*index switch*

The *switch* will be set to `-yes` if the element with index *index* is located on the boundary of the 
mesh.

This record will only become available if *mesh_boundary* is set to `-yes`. This record is meant 
for printing only, it should not be set by the user.

6.407  **element_contact_spring_direction**  
*index dirN_x dirN_y dirN_z dirT1_x  
dirT1_y dirT1_z dirT2_x dirT2_y dirT2_z*

In the input file, you can specify with this record the directions of a contact spring. If not specified, 
after the calculation this record will be filled with the used directions. The *index* specifies the spring 
element number.

6.408  **element_contact_spring_strain**  
*index strain_N strain_T1 strain_T2*

After the calculation, this record will be filled with the normal and tangential elongation in a 
contact_spring element. The *index* specifies the spring element number. The tangential strain 
strain_T2 only is present in 3D.

6.409  **element_contact_spring_force**  
*index force_N force_T1 force_T2*

After the calculation, this record will be filled with the normal and tangential forces in a con-
tact_spring element. The *index* specifies the spring element number. The tangential force force_T2 
only is present in 3D.

6.410  **element_dof**  
*index dof_0 dof_1 . . .*

Unknowns as saved per element in the element nodes. First dof's in the first node. Then dof's in 
the second node. Etc.

This is done optionally by tochnog, only when needed for the calculation. The *index* specifies the 
element number.
6.411 element_dof_initial index dof_0 dof_1 …

When an element comes the first time to live, it assumes that it had in the past the dof’s specified in this element_dof_initial record. You can either specify one value for each dof or you can specify values for the dof’s for all nodes (specify first all dof’s for the first node, then specify the dof’s for the second node, etc.). The index specifies the element number.

This record will influence inertia terms (like mass acceleration, temperature capacity, etc). As an example you can set so the initial temperature of a part that is connected to the mesh at some time.

6.412 element_dof_initial_specific_number index number

With this record you can an initial value for one specific dof. The number specifies the dof number, for example -velx or -sigxx, etc. The initial value for the dof needs to be specified with element_dof_initial_specific_value. The index specifies the element number.

6.413 element_dof_initial_specific_value index value_0 value_grad_x value_grad_y value_grad_z

This specifies for the element_dof_initial_specific_number record the initial value. Here value_0 is the value at coordinate x = y = z = 0, value_grad_x is the x-gradient, value_grad_y is the y-gradient and value_grad_z is the z-gradient. In 1D you only need to specify for the gradients the value_grad_x and in 2D you only need to specify for the gradients the value_grad_x and value_grad_y. As special option you can specify no gradients at all, and then a constant value in space of size value_0 will be used.

6.414 element_empty index switch

If Tochnog believes an element is empty, then it will set automatically switch to -empty for element_empty.

6.415 element_geometry index geometry_set

This data item specifies for element index a geometrical set number geometry_set. Elements with the same geometrical set number together form a geometry, which can be referenced by functionality selecting elements by a geometry. The syntax for referring is -element_geometry geometry_set.

A typical application would be changing material data (groups) in time for different sets of elements. In the example below element 1 belongs to geometrical set 10. The elements of geometrical set 10 get in time respectively groups 100, 101, 102 and 103.

```
  element 1 -bar2 1 2
  element 2 -bar2 2 3
  element_geometry 1 10
  element_geometry 2 20
  ...
  area_element_group_sequence_time 11 0. 1. 2. 3.
  area_element_group_sequence_geometry 11 -element_geometry 10
```
The `element_geometry` cannot be used in a `geometry_set`.

### 6.416 `element_geometry_present` index geometry_item_name_0 geometry_item_index_0 geometry_item_name_1 geometry_item_index_1 ...

This record lists for element `index` the geometries in which it is present. So it is a print record only, for checking if the geometries include exactly the elements that you want. You can switch on or off filling of these records by setting `print_element_geometry_present` to `-yes` or `-no`.

### 6.417 `element_group` index `element_group`

This data item is specified which element data items should be taken for the element `index`. Example: elements 0 and 1 get density 1024 while element 2 gets density 1236

```
element 0 0 1 2
element 1 1 2 3
element 2 2 3 4
...  
element_group 0 1
element_group 1 1
element_group 2 2
...  
density 1 1024.
density 2 1236.
```

If no `element_group` records are specified, all element data should use `index` is 0.

See also `area_element_group` and `element_geometry`.

### 6.418 `element_group_apply` index `element_group_0` `element_group_1` ...

This is yet another option to change the group of elements. It works in combination with `control_element_group_apply`. We explain it by means of an example:

```
element 43 ...
element 44 ...
...  
element_group_apply 43 1 7 4
element_group_apply 44 21 22 8
...  
```
control_element_group_apply 10 0 (element 43 gets group 1, element 44 gets group 21)

... control_element_group_apply 20 2 (element 43 gets group 4, element 44 gets group 8)

... control_element_group_apply 30 1 (element 43 gets group 7, element 44 gets group 22)

... 

6.419 element_interface_intpnt_direction index normal_x_0 normal_y_0 normal_z_0 first_tangential_x_0 first_tangential_y_0 first_tangential_z_0 second_tangential_x_0 second_tangential_y_0 second_tangential_z_0 ...

After the calculation this record will be filled with the direction vectors in interface element. Here normal_x_0 is the x-component of the normal direction in the first integration point, etc.

6.420 element_interface_intpnt_gap_status index status

After the calculation, this record will be filled with the gap status in an interface element. The status is either -opened or -closed. The index specifies the interface element number.

6.421 element_interface_intpnt_materi_tension_status index status

After the calculation, this record will be filled with the materi tension status in an interface element. The status is either -opened or -closed. The index specifies the interface element number.

6.422 element_interface_intpnt_strain index strain,normal,0 strain,shear,first,0 strain,shear,second,0 strain,normal,1 strain,shear,first,1 strain,shear,second,1 ...

After the calculation, this record will be filled with the normal strain, the first shear strain and second shear strain in the integration points of an interface element. The values with subscript 0 are for the first integration point; The values with subscript 1 are for the second integration point; etc. For a 2D interface element the second shear strain will not be set.

In fact, the normal strain is the normal displacement difference, and the shear strains are half of the shear displacement differences.

This element_interface_intpnt_strain record will only be filled if materi_strain_total is initialised. The index specifies the interface element number.

6.423 element_interface_intpnt_strain_average index strain,normal,0 strain,shear,first,0 strain,shear,second,0

Average of element_interface_intpnt_strain.
6.424  **element_interface_intpnt_stress** index stress,normal,0 stress,shear,first,0 stress,shear,second,0 stress,normal,1 stress,shear,first,1 stress,shear,second,1

After the calculation, this record will be filled with the normal stress, the first shear stress and the second shear stress in the integration points of an interface element. The values with subscript 0 are for the first integration point; The values with subscript 1 are for the second integration point; etc. For a 2D interface element the second shear stress will not be set.

The *index* specifies the interface element number.

See **control_reset_interface** on how to reset strains and stresses somewhere in a calculation.

6.425  **element_interface_intpnt_stress_average** index stress,normal,0 stress,shear,first,0 stress,shear,second,0

Average of **element_interface_intpnt_stress**.

6.426  **element_intpnt_dof** index dof_0 dof_1 ...

Unknowns as saved per element in the element integration points. The *index* specifies the element number.

6.427  **element_intpnt_h** index ...

This record is meant for printing only. It contains for each node of the element the value of the interpolation polynomial in the integration points.

6.428  **element_intpnt_iso_coord** index ...

This record is meant for printing only. It contains for each node of the element the value of the isoparametric coordinates in the integration points.

6.429  **element_intpnt_materi_plasti_hardsoil_gammap_initial** index gammap_initial_integration_point_0 gammap_initial_integration_point_1 ...

See theory section on hardsoil.

6.430  **element_intpnt_materi_undrained_pressure** index undrained_total_pressure

Total pressure from undrained analysis. See **group_materi_undrained_capacity**.

6.431  **element_intpnt_method** index method

This record is meant for printing only. It shows the space integration method that is actually used for element *index*. See also **group_integration_method**.
6.432 element_intpnt_npoint index npoint

This record is meant for printing only. It shows the number of space integration method points that are actually used for element index. See also group_integration_points.

6.433 element_intpnt_plasti_laminate0_mohr_coul_status index status

This record is meant for printing only. It gives for all integration points of an element the status of the mohr-coulomb yield rule of laminate 0. The status can be either -elastic or -plastic. For other laminates the records are element_intpnt_plasti_laminate1_mohr_coul_status etc.

The index is the element number.

6.434 element_intpnt_plasti_laminate0_tension_status index status

This record is meant for printing only. It gives for all integration points of an element the status of the tension cutoff yield rule of laminate 0. The status can be either -elastic or -plastic. For other laminates the records are element_intpnt_plasti_laminate1_tension_status etc.

The index is the element number.

6.435 element_materi_plasti_laminate0_apply index switch

If switch is set to -yes, laminate 0 of the multilaminate model will be applied for the element with number index (if the laminate is specified in the element group data). If switch is set to -no, laminate 0 of the multilaminate model will not be applied for the element with number index.

Default, if element_materi_plasti_laminate0_apply is not specified for an element then the switch is set to -yes.

For other laminates, element_materi_plasti_laminate1_apply should be specified.

6.436 element_materi_plasti_laminate0_direction index dir_x dir_y dir_z

If this record is specified, laminate 0 of the multilaminate model of the element with number index gets dir_x dir_y dir_z as normal for the multilaminate plane. This element_materi_plasti_laminate0_direction overrules the presence, if any, of the group_materi_plasti_laminate0_direction record for the element group.

6.437 element_middle index middle_x middle_y middle_z

After the calculation, this record will be filled with the middle coordinates of an element. The index specifies the element number.

6.438 element_print_group_data_values index ...

Values as required by print_group_data. The first value as required by print_group_data is placed in the first value of element_print_group_data_values. The second value as required
by print_group_data is placed in the second value of element_print_group_data_values.
Etc. Please realise that some group data requires more than one value, so that more than one value is filled in the element_print_group_data_values record.

6.439 element_spring_force index force

After the calculation, this record will be filled with the force in a spring element. The index specifies the spring element number.

6.440 element_spring_strain index strain

After the calculation, this record will be filled with the strain in a spring element. In fact the strain in a spring element is the elongation of the spring. The index specifies the spring element number.

In case you perform a geotechnical analysis and want to set all strains in the model to 0 after gravity has been imposed, then do a control_data_delete on all element_spring_strain records. In such way the element_spring_strain records will contain in the remaining part of the calculation strains relative to the gravity status.

6.441 element_truss_direction index dir_x dir_y dir_z

After the calculation, this record will be filled with the direction of a truss in space. The index specifies the truss element number.

6.442 element_truss_force index force

After the calculation, this record will be filled with the normal force in a truss element. The index specifies the truss element number.

6.443 element_truss_strain index strain

After the calculation, this record will be filled with the strain in a truss element (length increase divided by length). The index specifies the truss element number.

6.444 element_truss_strain_temperature index strain

After the calculation, this record will be filled with the normal thermal strain in a truss element (thermal length increase divided by length). The index specifies the truss element number.

6.445 element_volume index volume

This record contains the volume of the isoparametric element number index after the calculation. In fact for 1D elements it contains the element length, for 2D elements it contains the element area, and for 3D elements it contains the element volume.
6.446  **force_edge**  *index force_0 force_1 ...*

Distributed edge forces. These distributed forces are translated into equivalent nodal force terms on the edges of elements. You should specify a force term for each direction. Also the record **force_edge_geometry** should be specified, and optionally the records **force_edge_factor** and **force_edge_time** can be specified.

**Attention**: if this **force_edge** option is used INSIDE a FE mesh, then the elements on each side of the geometry will get the force. So you may need to specify only half of the physical force value.

**Attention**: this option is only available for linear and quadratic isoparametric elements.

6.447  **force_edge_diagram**  *index vertical_displacement_0 factor_0 vertical_displacement_1 factor_1 ...

This is a special purpose record, to make the specified forces dependent on vertical displacements. It actually contains multiplication factors for the force values as specified in the **force_edge** record with the same index. You need to specify sets of vertical displacements and corresponding multiplication values. Please realise the following:

- In many geotechnical calculations vertical displacements are negative (so eg vertical_displacement_0 is -3.e-2, vertical_displacement_1 is -2.e-2, etc.)
- The vertical displacements should be specified from low to high in the diagram.
- You need to specify all possible displacement values in the diagram.

6.448  **force_edge_element**  *index element_0 element_1 ...

Selects the element for which the **force_edge** record with the same *index* should be applied.

6.449  **force_edge_element_group**  *index element_group_0 element_group_1 ...

Selects the element group for which the **force_edge** record with the same *index* should be applied.

6.450  **force_edge_element_node**  *index element_node_0 node_1 ...

Selects the element and local node numbers for which the **force_edge** record with the same *index* should be applied.

6.451  **force_edge_element_side**  *index element_0 element_1 ... side*

Selects the elements and local side number for which the **force_edge** record with the same *index* should be applied.
6.452 force_edge_factor index \( a_0 \ a_1 \ldots a_n \)

This data item defines a polynomial in space. This polynomial gives a factor which is used as a multiplication factor for force_edge records (with the same index). In this way, you can obtain coordinate dependent forces.

In 1D the polynomial is \( a_0 + a_1 x \) (specify 2 values). In 2D the polynomial is \( a_0 + a_1 x + a_2 y \) (specify 3 values). In 3D the polynomial is \( a_0 + a_1 x + a_2 y + a_3 z \) (specify 4 values).

6.453 force_edge_geometry index geometry_entity_name geometry_entity_index

Selects the area for which the force_edge record with the same index should be applied. For example, -geometry_line 1 can be used in 2D, indicating that the nodes on line 1 get the distributed force. The total edge of an element must be inside the geometry for the force to become active. For 2D elements the border lines are edges. For 3D elements the border surfaces are edges.

6.454 force_edge_node index node_0 node_1 \ldots

Selects the nodes for which the force_edge record with the same index should be applied. The node_0 etc. specify global node numbers.

6.455 force_edge_node_factor index factor_0 factor_1 \ldots

Nodal multiplication factors with which the force of force_edge will be applied to the nodes of force_edge_node. You need to specify a factor for each node. Here factor_0 is the multiplication factor for the first node, etc.

6.456 force_edge_sine index start_time end_time freq_0 amp_0 freq_1 amp_1 \ldots

The force_edge record with the same index is imposed with the sum of the sine functions; the first sine function has frequency freq_0 and amplitude amp_0, the second sine function has frequency freq_1 and amplitude amp_1, etc.. The sine functions start at time 0. More general behavior in time can be imposed by using force_edge_time records. For a specific index only one of force_edge_sine and force_edge_time can be specified.

The sine loads will be only imposed after start_time, and only up to end_time. More general time behavior can be specified with force_edge_time.

6.457 force_edge_time index time load time load \ldots

This record specifies a diagram which contains the factors with which the force_edge record with the same index is applied. Linear interpolation is used to extend the time load values to the intervals between these pairs. Outside the specified time range a factor 0 is used.

If this record is not specified, and the force_edge_sine record is not specified, the force is applied at all times with a factor 1.
If no external forces like `force_edge_time` are specified, the internal element forces become zero at free edges to satisfy equilibrium. This causes, for example, temperature gradients to become zero at free edges in heat problems.

### 6.458 force_edge_normal index force

Distributed normal force in the direction of the outward normal at the edge of a element. This distributed term is translated into equivalent nodal force terms on the edges of elements. Also the record `force_edge_normal_geometry` should be specified, and optionally the record `force_edge_normal_time` can be specified.

**Attention:** this option is only available for linear and quadratic isoparametric elements.

**Attention:** if this `force_edge_normal` option is used INSIDE a FE mesh, then the elements on each side will get the force. So the total force will normally become zero since the normals of the elements at the side of the geometry are opposite.

### 6.459 force_edge_normal_element index element_0 element_1 ...

Restricts the element to which the `force_edge_normal` record with the same `index` should be applied.

### 6.460 force_edge_normal_element_node index element node_0 node_1

Selects the element and local node numbers for which the `force_edge_normal` record with the same `index` should be applied.

### 6.461 force_edge_normal_element_group index element_group_0 element_group_1 ...

Restricts the element group to which the `force_edge_normal` record with the same `index` should be applied.

### 6.462 force_edge_normal_element_side index element_0 element_1 ... side

Selects the elements and local side number for which the `force_edge_normal` record with the same `index` should be applied.

### 6.463 force_edge_normal_factor index a_0 a_1 ... a_{n-1}

This data item defines a polynomial in space. This polynomial gives a factor which is used as a multiplication factor for `force_edge_normal` records (with the same index). In this way, you can obtain coordinate dependent forces.

In 1D the polynomial is $a_0 + a_1 x$ (specify 2 values). In 2D the polynomial is $a_0 + a_1 x + a_2 y$ (specify 3 values). In 3D the polynomial is $a_0 + a_1 x + a_2 y + a_3 z$ (specify 4 values).
6.464 force_edge_normal_geometry index geometry_entity_name geometry_entity_index

Selects the area for which the force_edge_normal record with the same index should be applied. For example, -geometry_line 1 can be used in 2D, indicating that the nodes on line 1 get the distributed force. The total edge of an element must be inside the geometry for the force to become active. For 2D elements the border lines are edges. For 3D elements the border surfaces are edges.

6.465 force_edge_normal_node index node_0 node_1 node_2 ...

Selects the nodes for which the force_edge_normal record with the same index should be applied. The node_0 etc. specify global node numbers.

6.466 force_edge_normal_node_factor index factor_0 factor_1 ...

Nodal multiplication factors with which the force of force_edge_normal will be applied to the nodes of force_edge_normal_node. You need to specify a factor for each node. Here factor_0 is the multiplication factor for the first node on the side, etc.

6.467 force_edge_normal_sine index start_time end_time freq_0 amp_0 freq_1 amp_1 ...

Same as force_edge_sine, now for normal edge loads however.

6.468 force_edge_normal_time index time load time load ...

This record specifies a diagram which contains the factors with which the force_edge_normal record with the same index is applied. Linear interpolation is used to extend the time load values to the intervals between these pairs. Outside the specified time range a factor 0 is used.

If this record is not specified, the force is applied at all times with a factor 1.

6.469 force_edge_projected index force ph(0,0,0) ph_grad_x ph_grad_y ph_grad_z
pv(0,0,0) pv_grad_x pv_grad_y pv_grad_z factor_normal factor_tangential
vertical_dir_downward_x vertical_dir_downward_y vertical_dir_downward_z
tunnel_dir_x tunnel_dir_y tunnel_z

Distributed projected force on the edge of an element. This distributed term is translated into equivalent nodal force terms on the edges of elements.

This record typically can be used to model soil normal and tangential loading on tunnels. With ph(0,0,0) you specify the horizontal ground stress at x=0,y=0,z=0. With ph_grad_x, ph_grad_y and ph_grad_z you specify the gradients of the horizontal stress (such that a linear horizontal stress field can be modeled). With pv(0,0,0) you specify the vertical ground stress at x=0,y=0,z=0. With pv_grad_x, pv_grad_y and pv_grad_z you specify the gradients of the vertical stress (such that a linear vertical stress field can be modeled).
The vertical and horizontal stresses are projected on the edge of the element so that the radial stress $\sigma_{\text{radial}}$ and the tangential stress $\sigma_{\text{tangential}}$ of the edge of the element are obtained. You can decide to apply the radial stress $\sigma_{\text{radial}}$ only with a factor $\text{factor\_normal}$ (between 0 and 1). Likewise, you can decide to apply the tangential shear stress $\sigma_{\text{tangential}}$ only with a factor $\text{factor\_tangential}$ (between 0 and 1).

As extra information for Tochnog to determine the correct radial stress and tangential shear stress on the edge of an element you need to specify the downward vertical direction with $\text{vertical\_dir\_downward\_x}$, $\text{vertical\_dir\_downward\_y}$ and $\text{vertical\_dir\_downward\_z}$.

Only in 3D, you also need to specify the length direction of the tunnel axis with $\text{tunnel\_dir\_x}$, $\text{tunnel\_dir\_y}$ and $\text{tunnel\_z}$.

In 2D you should not specify the 3D information $\text{ph\_grad\_z}$, $\text{pv\_grad\_z}$, $\text{vertical\_dir\_downward\_z}$, $\text{tunnel\_dir\_x}$, $\text{tunnel\_dir\_y}$ and $\text{tunnel\_z}$.

Also the record $\text{force\_edge\_projected\_geometry}$ should be used to specify where the force should be applied, and optionally the record $\text{force\_edge\_projected\_time}$ can be specified.

Attention: notice that horizontal soil stress in length direction of the tunnel is not included.

Attention: this option is only available for linear and quadratic isoparametric elements.

Attention: if this $\text{force\_edge\_projected}$ option is used INSIDE a FE mesh, then the elements on each side will get the force. So the total force will protectedly become zero since the projected of the elements at the side of the geometry are opposite.

6.470  $\text{force\_edge\_projected\_element}$  $\text{index}$  $\text{element\_0}$  $\text{element\_1}$  

Restricts the element to which the $\text{force\_edge\_projected}$ record with the same $\text{index}$ should be applied.

6.471  $\text{force\_edge\_projected\_element\_node}$  $\text{index}$  $\text{element}$  $\text{node\_0}$  $\text{node\_1}$  

Selects the element and local nodes for which the $\text{force\_edge\_projected}$ record with the same $\text{index}$ should be applied.

6.472  $\text{force\_edge\_projected\_element\_group}$  $\text{index}$  $\text{element\_group\_0}$  $\text{element\_group\_1}$  

Restricts the element group to which the $\text{force\_edge\_projected}$ record with the same $\text{index}$ should be applied.

6.473  $\text{force\_edge\_projected\_element\_side}$  $\text{index}$  $\text{element\_0}$  $\text{element\_1}$  $\text{side}$  

Selects the elements and local side number for which the $\text{force\_edge\_projected}$ record with the same $\text{index}$ should be applied.
6.474 force_edge_projected_factor index $a_0 \ a_1 \ldots a_n$

This data item defines a polynomial in space. This polynomial gives a factor which is used as a multiplication factor for force_edge_projected records (with the same index). In this way, you can obtain coordinate dependent forces.

In 1D the polynomial is $a_0 + a_1 x$ (specify 2 values). In 2D the polynomial is $a_0 + a_1 x + a_2 y$ (specify 3 values). In 3D the polynomial is $a_0 + a_1 x + a_2 y + a_3 z$ (specify 4 values).

6.475 force_edge_projected_geometry index geometry_entity_name geometry_entity_index

Selects the area for which the force_edge_projected record with the same index should be applied. For example, -geometry_line 1 can be used in 2D, indicating that the nodes on line 1 get the distributed force. The total edge of an element must be inside the geometry for the force to become active. For 2D elements the border lines are edges. For 3D elements the border surfaces are edges.

6.476 force_edge_projected_node index node_0 node_1 node_2 \ldots

Selects the nodes for which the force_edge_projected record with the same index should be applied. The node_0 etc. specify global node numbers.

6.477 force_edge_projected_node_factor index factor_0 factor_1 \ldots

Nodal multiplication factors with which the force of force_edge_projected will be applied to the nodes of force_edge_projected_node. You need to specify a factor for each node. Here factor_0 is the multiplication factor for the first node, etc.

6.478 force_edge_projected_sine index start_time end_time freq_0 amp_0
defreq_1 amp_1 \ldots

Similar to force_edge_sine, now for projected edge loads however.

6.479 force_edge_projected_time index time load time load \ldots

This record specifies a diagram which contains the factors with which the force_edge_projected record with the same index is applied. Linear interpolation is used to extend the time load values to the intervals between these pairs. Outside the specified time range a factor 0 is used.

If this record is not specified, the force is applied at all times with a factor 1.

6.480 force_edge_water index switch

If switch is set to -yes, distributed water pressure force is added to the model. This distributed term is translated into equivalent nodal force terms on the edges of elements. The distributed force is automatically calculated as density_water $g$ $\Delta z$ where $g$ is the gravitational acceleration, and
\( \Delta z \) is the distance to the phreatic level. The water pressure force acts normal to the element edge, in inward direction. You need to specify also `force_edge_water Geometry`.

The water density is given by `groundflow_density`. The gravity acceleration is given by the vertical component of `force_gravity`. The water height is relative to the phreatic level.

**Attention:** if this `force_edge_water` option should be used with care INSIDE a FE mesh.

The total edge of an element must be inside the geometry for the force to become active. For 2D elements the border lines are edges. For 3D elements the border surfaces are edges.

### 6.481 force_edge_water_element

index element_0 ...

Selects the element for which the `force_edge_water` record with the same `index` should be applied.

### 6.482 force_edge_water_element_group

index element_group_0 ...

Selects the element groups for which the `force_edge_water` record with the same `index` should be applied.

### 6.483 force_edge_water_element_node

index element node_0 node_1 ...

Selects the element and local nodes for which the `force_edge_water` record with the same `index` should be applied.

### 6.484 force_edge_water_element_side

index element_0 element_1 ...side

Selects the elements and local side number for which the `force_edge_water` record with the same `index` should be applied.

### 6.485 force_edge_water_factor

index \( a_0 \) \( a_1 \) ... \( a_n \)

This data item defines a polynomial in space. This polynomial gives a factor which is used as a multiplication factor for `force_edge_water` records (with the same index). In this way, you can obtain coordinate dependent forces.

In 1D the polynomial is \( a_0 + a_1 x \) (specify 2 values). In 2D the polynomial is \( a_0 + a_1 x + a_2 y \) (specify 3 values). In 3D the polynomial is \( a_0 + a_1 x + a_2 y + a_3 z \) (specify 4 values).

### 6.486 force_edge_water_geometry

index geometry_item_name geometry_item_index

Selects the area for which the `force_edge_water` record with the same `index` should be applied. For example, `-geometry_line 1` can be used in 2D, indicating that the nodes on line 1 get the distributed water pressure force.
6.487  **force_edge_water_node**  index node_0 node_1 ...

Selects the nodes for which the **force_edge_water** record with the same *index* should be applied. The *node_0* etc. specify global node numbers.

6.488  **force_edge_water_time**  index time load time load ...

This record specifies a diagram which contains the factors with which the **force_edge_water** record with the same index is applied. Linear interpolation is used to extend the *time load* values to the intervals between these pairs. Outside the specified time range a factor 0 is used.

If this record is not specified, the force is applied at all times with a factor 1.

6.489  **force_gravity**  \( g_x \ g_y \ g_z \)

Gravitational acceleration.

In 1D, only the gravity in x-direction needs to be specified. In 2D, the gravity in x-direction and y-direction needs to be specified. In 3D, the gravity in x-direction, y-direction and z-direction needs to be specified.

See also **force_gravity_time**.

6.490  **force_gravity_geometry**  geometry_item_name geometry_item_index

With this record you can specify a geometrical entity on which the gravity force should be used. Only elements inside the geometry get the gravity force.

If this record is not specified all elements can get the gravity force.

See also **force_gravity_time**.

6.491  **force_gravity_time**  time load time load ...

This record specifies a multi-linear diagram which contains the factors with which the **force_gravity** record is applied. This allows you to impose the gravity on a structure slowly, which might be needed for path dependent problems. Outside the specified time range a factor 0 is used.

If this record is not specified, the gravity is applied at all times with a factor 1.

6.492  **force_point**  index coord_0 coord_1 coord_2 force_0 force_1 force_2

Point force in space; it needs not to be exactly in a node. The force will be distributed over the element nodes in which the point force is located. This will be done consistent with the element interpolation functions (linear or quadratic).

The *coord_0* specifies the x-coordinate in space. The *coord_1* specifies the y-coordinate in space (only in two or 3 dimensions). The *coord_2* specifies the z-coordinate in space (only in 3 dimensions). The *force_0* specifies the x-force in space. The *force_1* specifies the y-force in space (only in two or 3 dimensions). The *force_2* specifies the z-force in space (only in 3 dimensions).
You can use the usual `change_dataitem_time` to change the position (or value) of the point force in time.

6.493 **force_volume** \(\text{index} \ \text{force}_0 \ \text{force}_1 \ldots\)

Distributed volume forces for each direction. Here `force_0` is the distributed force in the x-direction, etc. Consider the example with distributed volume force in x-direction for a 2D material:

```
force_volume 0 1.0.
```

The `force_volume` record can be used in `dependency_diagram` records (just like element group data)/

See also `force_volume_factor`, `force_volume_geometry`, and `force_volume_time`.

6.494 **force_volume_element** \(\text{index} \ \text{element}_0 \ \text{element}_1 \ldots\)

Specifies the elements for which the `force_volume` record with the same `index` should be applied.

6.495 **force_volume_element_group** \(\text{index element_group} \ldots\)

Specifies the element group for which the `force_volume` record with the same `index` should be applied.

6.496 **force_volume_factor** \(\text{index} \ a_0 \ a_1 \ldots a_n\)

This polynomial gives a factor which is used as a multiplication factor for `force_volume` records (with the same index). In this way, you can obtain coordinate dependent forces.

In 1D the polynomial is \(a_0 + a_1x\) (specify 2 values). In 2D the polynomial is \(a_0 + a_1x + a_2y\) (specify 3 values). In 3D the polynomial is \(a_0 + a_1x + a_2y + a_3z\) (specify 4 values).

6.497 **force_volume_geometry** \(\text{index geometry_item_name geometry_item_index}\)

Specifies the area for which the `force_volume` record with the same `index` should be applied. For example, `-geometry_quadrilateral 1` can be used in 2D, indicating that the elements on quadrilateral 1 get the distributed force.

If both the `force_volume_element` and `force_volume_geometry` are not specified, then a geometry which encloses the whole model will be applied.

6.498 **force_volume_sine** \(\text{index start_time freq}_0 \ \text{amp}_0 \ \text{freq}_1 \ \text{amp}_1 \ldots\)

Same as `force_volume_sine`, now for volume loads however.
6.499 force_volume_time index time load time load . . .

This record specifies a multi-linear diagram which contains the factors with which the force_volume record with the same index is applied.

If this record is not specified, the force is applied at all times with a factor 1.

6.500 geometry_boundary index switch

With this record you can restrict a geometry to the boundary of the mesh, or to the inside of the mesh. If switch is set to -yes only nodes which are at the boundary of the mesh are actually used for the geometry with the same index. If switch is set to -no only nodes which are not at the boundary of the mesh are actually used for the geometry with the same index.

Attention: for this option to work correctly, the mesh should not contain badly shaped elements. See the section at the end of this manual for more information on bad element shapes.

6.501 geometry_bounda_sine_x index a b

This sets, for the geometrical entity with the same index an extra factor which is used for the bounda_dof and the bounda_force records. The factor gives a sinus variation in x-direction. The size of the factor is \( \sin(a + b \times x) \).

6.502 geometry_bounda_sine_y index a b

This sets, for the geometrical entity with the same index an extra factor which is used for the bounda_dof and the bounda_force records. The factor gives a sinus variation in y-direction. The size of the factor is \( \sin(a + b \times y) \).

6.503 geometry_bounda_sine_z index a b

This sets, for the geometrical entity with the same index an extra factor which is used for the bounda_dof and the bounda_force records. The factor gives a sinus variation in z-direction. The size of the factor is \( \sin(a + b \times z) \).

6.504 geometry_brick index x_c y_c z_c l_x l_y l_z tolerance

This data item defines a brick in space. Other data items can check if nodes are located on this geometry. The coordinate of the center is \( x_c \ y_c \ z_c \). The length in respectively x, y and z direction are \( l_x \ l_y \ l_z \). All node within a distance tolerance are considered to be part of the brick.

6.505 geometry_circle index x_c y_c . . . radius tolerance

This data item defines a circle in space. Other data items can check if nodes are located on this geometry. The coordinate of the center is \( x_c \ y_c \). In 2D you need to specify \( x_c \ y_c \) radius tolerance. In 2D all node within a distance tolerance of the radius are considered to be part of the
circle. In 3D you need to specify $x_c \ y_c \ z_c \ \text{normal}_x \ \text{normal}_y \ \text{normal}_z \ \text{radius} \ \text{tolerance}$, where $\text{normal}_x \ \text{normal}_y \ \text{normal}_z$ specifies the direction normal to the surface. In 3D all node within a distance tolerance of the circle surface are considered to be part of the circle.

**6.506 geometry_circle_part** $\text{index} \ x_c \ y_c \ \text{angle}_\text{start} \ \text{angle}_\text{end} \ \text{radius} \ \text{tolerance}$

This data item defines a circle in 2D space. Other data items can check if nodes are located on this geometry. The coordinate of the center is $x_c \ y_c$. All node within a distance tolerance of the radius are considered to be part of the circle. The circle part starts at angle $\text{angle}_\text{start}$, measured in radians from the positive x-axis. The circle part ends at angle $\text{angle}_\text{end}$, measured in radians from the positive x-axis.

**6.507 geometry_circle_segment** $\text{index} \ x_c \ y_c \ \text{radius} \ \text{side}_x \ \text{side}_y \ \text{tolerance}$

This data item defines a circle segment in space. Other data items can check if nodes are located on this geometry. The coordinate of the center is $x_c \ y_c$. If $\text{side}_x$ is set to a positive value, say +1., then only x-values larger than $x_c$ are considered to be part of the geometry. If $\text{side}_x$ is set to a negative value, say -1., then only x-values smaller than $x_c$ are considered to be part of the geometry. If $\text{side}_x$ is set to 0, then all x-values are considered to be part of the geometry. Likewise remarks hold for y-values. All node within a distance tolerance of the radius are considered to be part of the circle segment.

**6.508 geometry_cylinder** $\text{index} \ x_0 \ y_0 \ z_0 \ x_1 \ y_1 \ z_1 \ \text{radius} \ \text{tolerance}$

This data item defines a cylinder segment in space. Other data items can check if nodes are located on this geometry. The coordinate of the center point at the bottom is $x_0 \ y_0 \ z_0$. The coordinate of the center point at the top is $x_1 \ y_1 \ z_1$. The cylinder can only be used in 3D. All node within a distance tolerance of the radius are considered to be part of the cylinder.

**6.509 geometry_cylinder_part** $\text{index} \ x_0 \ y_0 \ z_0 \ x_1 \ y_1 \ z_1 \ \text{radius} \ \text{angle}_\text{start}_0 \ \text{angle}_\text{end}_0 \ \text{angle}_\text{start}_1 \ \text{angle}_\text{end}_1 \ldots \ \text{tolerance}$

This data item defines parts of a cylinder in space. Other data items can check if nodes are located on this geometry.

The $\text{index} \ x_0 \ y_0 \ z_0 \ x_1 \ y_1 \ z_1 \ \text{radius}$ are the same as in geometry_cylinder.

The $\text{angle}_\text{start}_0 \ \text{angle}_\text{end}_0$ defines the first valid part of the cylinder, where $\text{angle}_\text{start}_0$ is the start angle of the part and $\text{angle}_\text{end}_0$ is the end angle. The angles are measured in the x-y plane, starting from the positive x-axis towards the positive y-axis. Likewise, the $\text{angle}_\text{start}_1 \ \text{angle}_\text{end}_1$ defines a second valid part of the cylinder. You should define at least one valid part, and optionally you can specify several valid parts.

Start angles and end angles should be non-negative. End angles should be larger than start angles.

Angles will be measured relative to the vector as specified in geometry_cylinder_part_start_vector, if that vector is specified. This geometry_cylinder_part_start_vector should be specified perpendicular to the cylinder axes. This geometry_cylinder_part_start_vector should be exactly in the middle of the angle range that you want to select. With geometry_cylinder_part_start_vector.
only one angle range is allowed, and the start angle should be 0. All nodes with an angle smaller or equal to the end angle are accepted as valid (thus, you get a total angle range of twice the end angle size as valid range).

If `geometry_cylinder_part_start_vector` is not specified, the `geometry_cylinder_part` should be either along the x-direction, y-direction or z-direction; then the angle is measured relative to the axes (for example for a cylinder along the z-direction the angle starts at the x-axes).

All node within a distance `tolerance` of the radius and inside a valid part are considered to be part of the cylinder part.

### 6.510 geometry_cylinder_part_start_vector

`index v_x v_y v_z`

See `geometry_cylinder_part`.

### 6.511 geometry_cylinder_segment

`index x_0 y_0 z_0 x_1 y_1 z_1 radius side_x side_y side_z tolerance`

This data item defines a cylindrical segment in space. Other data items can check if nodes are located on this geometry. The coordinate of the center point at the bottom is $x_0 \ 0 \ z_0$. The coordinate of the center point at the top is $x_1 \ y_1 \ z_1$. If `side_x` is set to a positive value, say +1., then only x-values larger then $x_c$ are considered to be part of the geometry. If `side_y` is set to a negative value, say -1., then only x-values smaller then $x_c$ are considered to be part of the geometry. If `side_x` is set to 0 , then all x-values are considered to be part of the geometry. Likewise remarks hold for y- and z-values. The cylinder segment can only be used in 3D. All node within a distance `tolerance` of the radius are considered to be part of the cylinder.

### 6.512 geometry_exclude

`index geometry_item_name_0 geometry_item_index_0 geometry_item_name_1 geometry_item_index_1 ...`

With this record you can exclude geometries from the geometry with the same index. The next 2D example excludes a circular area with radius 0.3 inside a quadrilateral:

```plaintext
geometry_quadrilateral 10 0. 0. 1. 0. 0. 1. 1.
gometry_exclude 10 -geometry_point 20
geometry_point 20 0.5 0.5 0.3
```

You are not allowed to let a `geometry_` use a `geometry_exclude` which contains itself.

### 6.513 geometry_element_geometry

`index element_geometry_0 element_geometry_1 ...`

Similar to `geometry_element_group`, but now using `element_geometry` i.s.o. `element_group` however.
6.514 geometry_element_geometry_method index method

Similar to geometry_element_group_method.

6.515 geometry_element_group index element_group_0 element_group_1 ...

With this record you can restrict the geometry as specified in the geometry record with the same index. For example for the geometry as specified by

    ... geometry_quadrilateral 10 ...
    geometry_element_group 10 ...
    ...

nodes which are located on the geometry_quadrilateral 10, but at the same time are also a node of elements of one of the specified element groups element_group_0 element_group_1 etc., belong to the geometry. Nodes which are not a node of elements of one of the groups do not belong to the geometry, even if such nodes are located on the geometry_quadrilateral 10.

See also geometry_element_group_method.

6.516 geometry_element_group_method index method

With this record you can set the method that the geometry_element_group record uses. If method is set to -all then a node should be attached to all the specified element groups, to be part of the geometry. If method is set to -any then a node should be attached to any of the specified element groups, to be part of the geometry. If method is set to -only then a node should be attached to only the specified element groups, to be part of the geometry. Default, if method is not specified then -any is assumed.

6.517 geometry_ellipse index x_c y_c a b tolerance

The coordinate of the center is x_c y_c. The equation for the ellipse is:

\[ \left( \frac{x - x_c}{a} \right)^2 + \left( \frac{y - y_c}{b} \right)^2 = 1 \]

Other data items can check if nodes are located on this geometry. The ellipse can only be used in 2D. All node within a distance tolerance of the ellipse are considered to be part of the ellipse.

6.518 geometry_factor index factor_0 ...

This sets for some geometries extra factors which are used for the bounda_dof, bounda_force and force_edge_* records. For a geometry_line either 2 or 3 factors should be specified; 2 factors define a linear variation where the factors hold at the start and end of the line respectively; 3 factors define a parabolic variation where the factors hold at the start, at the middle and at
the end of the line respectively. For a **geometry_triangle** 3 factors should be specified (a linear variation with factors for the first, second and third corner point respectively). For a **geometry_quadrilateral** 4 factors should be specified (a linear variation with factors for the first, second, third and fourth corner point respectively). For a **geometry_point** 1 factor should be specified; a multiplication with a half sine wave will be used, with the specified factor in the middle (exactly at the point) decreasing to factor 0 at a distance tolerance from the point.

In the example below, node 2 will get temperature $20 \times 1.6$ and node 3 will get temperature $20 \times 2.2$.

```plaintext
... number_of_space_dimensions 2
condif_temperature
...
end_initia
node 2 0.2 0
node 3 0.4 0.
...
geometry_line 1 0. 0. 1. 0. 0.01
gometry_factor 1 1. 4.
boundary_dof 0 -geometry_line 1 -temp
boundary_time 0 0. 20. 1.e6 20.
...
end_data
```

**6.519 geometry_hexahedral**

```plaintext
index x_0 y_0 z_0 x_1 y_1 z_1 x_2 y_2 z_2
 x_3 y_3 z_3
 x_4 y_4 z_4 x_5 y_5 z_5 x_6 y_6 z_6 x_7 y_7 z_7
```

This data item defines a hexahedral in space. Other data items can check if nodes are located on this geometry (everything inside the hexahedral belongs to the geometry). The coordinates of the corner points are $x_0$ $y_0$ $z_0$ etc.. The points of the hexahedral should be specified in the correct order; the order is clarified in the example below.

**Example**

```plaintext
... number_of_space_dimensions 3
...
geometry_hexahedral 0 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 1. 0. 1. 0. 0. 1. 0. 1. 0. 1. 0. 0. 1. 0. 1. 0. 1. 1.
0. 1. 1. 1. 1.
...
```

Notice the order in which the points are to be specified.

**6.520 geometry_line**

```plaintext
index x_0 y_0 z_0 x_1 y_1 z_1 radius
```

This data item defines a line in space. Other data items can check if nodes are located on this geometry. Coordinates of the end points are denoted by $x_0$, etc.. In 1D, only the $x$-coordinates should be specified, etc.. All node within a distance $radius$ are considered to be part of the line.
In the example, a line in 2D space is defined and is used by a `convection_geometry` record (nodes located on the line will convect heat)

... 

geometry_line 2 1. 0. 1. 1. 0.01 

... 

group_condif_convection_edge_normal_geometry 0 -geometry_line 2 

...

6.521 geometry_line_eps_iso index iso_tolerance

With this parameter you can ask Tochnog to accept points just outside the line in direction of the line. Typically try $1.e^{-3}$ for `iso_tolerance`.

6.522 geometry_list index number_0 number_1 ...

This is a list of numbers which can be used in geometry selection options.

For example

... 

geometry_list 10 1 45 43 26 27 

... 

bounda_dof 200 -geometry_list 10 ... (set the boundary condition on the nodes of the list) 

... 

6.523 geometry_method index method

For selecting elements with a geometry entity you can set the `method` either to `-all`, `-any` or `-average`. With `-all` all nodes of an element should be inside the geometry entity for the element to be selected (completely inside). With `-any` any node of an element should be inside the geometry entity for the element to be selected (at least partially inside). With `-average` the middle coordinate of an element should be inside the geometry entity for the element to be selected. Default if this record is not specified the `method` is set to `-all`.

6.524 geometry_moving index geometry_entity

This option comes handy when you want to model moving excavations in complex FE meshes. For complex geological regions it is not possible to make the FE mesh a priori in such way that the excavation zones can be easily defined in terms of element groups or otherwise. In such case this option allows geometrical entities to move through the complex FE mesh. The geometrical entity will determine automatically which elements become part of the entity at which times, and then will automatically excavate those elements. Elements which are at a moment in time only partly inside the geometrical entity, will be only partly excavated to the same amount as which
they are inside the geometrical entity. Elements which are at a moment in time completely inside
the geometrical entity, will be fully excavated.

We first explain the usage of this option by means of an example:

```
...  
gameometry_moving 10 -geometry_point (geometrical point that will move in space
and excavate the mesh)
gameometry_moving_parameter 10 -1. -1. 2.e-1 (start x-coordinate of point, start
y-coordinate of point, radius of point)
gameometry_moving_operat 10 -translate (translate the point in space)
gameometry_moving_operat_parameter 10 1. 1. (velocity of point in x-direction,
velocity of point in y-direction)
gameometry_moving_operat_time 10 0. 3. (start time of point moving, end time
of point moving)
gameometry_moving_n 10 200 200 (number of time discretisation, number of space
distretisation)
...  
mesht_delete gameometry_moving 10 10 (tell Tochnog to use the gameometry_moving
records with index 10 to excavate the mesh)
...  
control_gameometry_moving 25 -initialise (prepare the point for moving through
the mesh)
...
control_timestep 30 ...
...  
```

The geometrical entity can be one of -geometry_point, -geometry_triangle, -geometry_quadrilateral,
-geometry_hexahedral and -geometry_tethrahedral.

### 6.525 geometry_moving_parameter index parameters of entity

For each vertex of the geometry you need to specify the initial x-coordinate, y-coordinate (only
for 2D and 3D) and z-coordinate. A point has one vertex, a triangle three, a quadrilaterial four, a
hexahedral eight and a tethrahedral six. The sequence of the vertices is the same as the sequence of
nodes for the similar finite elements (see element). For a -geometry_point you need to specify
additionally the radius.

See also geometry_moving.

### 6.526 geometry_moving_operat index operator

This operator specifies how the geometry changes between time_start and time_end. If operator
is set to -translate the geometry moves in space with a constant velocity vector.

See also geometry_moving.

### 6.527 geometry_moving_operat_parameter index parameters of operator

For a -translate you need to specify the x-velocity, y-velocity (only for 2D and 3D) and z-velocity.
6.528  geometry_moving_operat_time  index start_time end_time

The start_time specifies when the geometry comes into existence and starts to move. The end_time specifies when the geometry stops moving but remains in existence.

See also geometry_moving.

6.529  geometry_moving_n  index ntime nspace

The geometry_moving command needs to find out for each element at each time point which part of the element is part of the geometrical entity. In order to find this out, it is checked for many points inside each element if the point is inside the geometrical entity. In fact, for a 1D element nspace points will be used, for a 2D element nspace * nspace points will be used and for nspace * nspace * nspace points will be used.

This checking which part of each element is inside the geometrical entity will be done at ntime time moments between time_start and time_end.

Default, if geometry_moving is not specified, we use ntime is 100 and nspace is 5.

See also geometry_moving.

6.530  geometry_mpc  index switch

With this record you can restrict the geometry as specified in the geometry record with the same index. Only nodes which have mpc’s are considered to be part of the geometry.

6.531  geometry_node_type  node_type

If node_type is set to -node_start_refined the values of -node_start_refined are used for evaluating the geometry. If node_type is set to -node the values of -node are used for evaluating the geometry. If node_type is set to -plus_displacement the values of -node plus nodal displacements are used for evaluating the geometry. Default node_type is set to -node_start_refined.

6.532  geometry_point  index x y z radius

This data item defines a point in space. Other data items can check if nodes are located on this geometry. The coordinate of the point is x y z. In 1D, only x should be specified, etc.. All node within a distance radius are considered to be part of the point.

6.533  geometry_polynomial  index a_0 a_1 \ldots a_n x_0 x_1 y_0 y_1 tolerance

This data item defines a polynomial in space in 2D or 3D. Other data items can check if nodes are located on this geometry.

In 2D the polynomial is $a_0 + a_1 x + a_2 y$ (specify 3 values). In 3D the polynomial is $a_0 + a_1 x + a_2 y + a_3 z$ (specify 4 values).
In 2D \(x_0 - x_1\) defines the domain of \(x\). In 3D \(x_0 - x_1\) defines the domain of \(x\) and \(y_0 - y_1\) defines the domain of \(y\). All node with a distance (that is the \(y\)-distance in 2D or the \(z\)-distance in 3D) not more than \(tolerance\) are considered to be part of the polynomial.

### 6.534 geometry_projection_type

\[ index \_ type \]

This record allows you to control what geometry will actually be used. Set \(type\) to -project_inside or -project_exact. For example if the geometry is a geometry_circle then -project_inside means that everything inside the circle will be used, whereas -project_exact means that everything within a tolerance from the circle edge will be used. Default \(type\) is -project_exact.

### 6.535 geometry_quadrilateral

\[ index \_ x_0 \ y_0 \ z_0 \ x_1 \ y_1 \ z_1 \ x_2 \ y_2 \ z_2 \]

This data item defines a quadrilateral in space. Other data items can check if nodes are located on this geometry. The coordinates of the corner points are \(x_0 \ y_0 \ z_0\) etc.. In 2D, only \(x_0 \ y_0\) etc. should be specified etc.. The points of the quadrilateral should be specified in the correct order; the order is clarified in the example below.

In 2D all node inside the quadrilateral (the tolerance is neglected). In 3D all node within a distance \(tolerance\) are considered to be part of the quadrilateral (this is a brick with thickness \(tolerance\)). All node within a distance \(tolerance\) are considered to be part of the quadrilateral (in 2D this gives a quadrilateral with corners nodes specified by the corners points, in 3D this gives a brick corners nodes specified by the corners points and with thickness \(tolerance\)). Internally in TOCHNOG, the quadrilateral is divided into two geometry_triangles, which is only approximately true if the quadrilateral is twisted. Example

```
...  
number_of_space_dimensions 2  
...  
geometry_quadrilateral 0 0. 0. 1. 0. 0. 1. 1. 1.e-3  
...  
```

Notice the order in which the points are to be specified.

### 6.536 geometry_quadrilateral_eps_iso

\[ index \_ iso\_tolerance \]

With this parameter you can ask Tochnog to accept points just outside the quadrilateral in direction of the quadrilateral plane. Typically try \(1.e - 3\) for iso_tolerance.

### 6.537 geometry_set

\[ index \_ geometry\_entity_0 \_ geometry\_entity\_index_0 \_ geometry\_entity_1 \_ geometry\_entity\_index_1 \ldots \]

This set combines a number of geometrical entities (e.g. geometry_circle, geometry_line, etc.) into a new entity. You cannot use another geometry set for the geometrical entities (that is, geometry sets cannot be nested).
Other data items can check if nodes are located on this geometry.

6.538 geometry_sphere \textit{index} \(x_c \ y_c \ z_c \ radius \ tolerance\)

This data item defines a sphere in space. Other data items can check if nodes are located on this geometry. The coordinate of the center is \(x_c \ y_c \ z_c\). All \textit{node} within a distance \textit{tolerance} of \textit{radius} are considered to be part of the sphere.

6.539 geometry_sphere_segment \textit{index} \(x_c \ y_c \ z_c \ radius \ side_x \ side_y \ side_z \ tolerance\)

This data item defines a spherical segment in space. Other data items can check if nodes are located on this geometry. The coordinate of the center is \(x_c \ y_c \ z_c\). If \textit{side_x} is set to a positive value, say +1., then only x-values larger then \(x_c\) are considered to be part of the geometry. If \textit{side_x} is set to a negative value, say -1., then only x-values smaller then \(x_c\) are considered to be part of the geometry. If \textit{side_x} is set to 0, then all x-values are considered to be part of the geometry. Likewise remarks hold for y- and z-values.

All \textit{node} within a distance \textit{tolerance} of \textit{radius} are considered to be part of the spherical segment.

6.540 geometry_tetrahedral \textit{index} \(x_0 \ y_0 \ z_0 \ x_1 \ y_1 \ z_1 \ x_2 \ y_2 \ z_2 \ x_3 \ y_3 \ z_3\)

This data item defines a tetrahedral in space. Other data items can check if nodes are located on this geometry. The coordinates of the corner points are \(x_0 \ y_0 \ z_0\) etc..

6.541 geometry_triangle \textit{index} \(x_0 \ y_0 \ z_0 \ x_1 \ y_1 \ z_1 \ x_2 \ y_2 \ z_2 \ tolerance\)

This data item defines a triangle in space. Other data items can check if nodes are located on this geometry. The coordinates of the corner points are \(x_0 \ y_0 \ z_0\) etc.. In 2D the z coordinates should not be specified. All \textit{node} within a distance \textit{tolerance} are considered to be part of the triangle (this gives a wedge with thickness \(2\textit{tolerance}\)).

6.542 geometry_triangle_eps_iso \textit{index} \(iso\_tolerance\)

With this parameter you can ask Tochnog to accept points just outside the triangle in direction of the triangle plane. Typically try \(1.e-3\) for \(iso\_tolerance\).

6.543 global_node_dof_empty \textit{switch}

If \textit{switch} is set to \textit{-yes}, the dof’s in a node are set to zero when the node is only connected to empty elements. If \textit{switch} is set to \textit{-no}, the dof’s in a node are not set to zero when the node is only connected to empty elements. Default \textit{switch} is set to \textit{-yes}.
6.544 global_element_dof_apply switch

If you set switch to -yes, then dof’s like strains, stresses, etc. will be saved in the element integration points in the records element_intpnt_dof. So, these dof’s will actually not be averaged over global nodes, but each element remembers its own values for these dof’s. This will be done for dof’s like strains, stresses, etc. only. Other dof’s like velocities, displacement field, temperature, etc. are not saved per element, but remain saved in the global nodes.

If you set switch to -no, then elements will actually use the averaged nodal results, and will not remember its own values.

Default, if global_element_dof_apply is not specified, global_element_dof_apply is set to -yes. See also global_element_dof_from_node_dof.

6.545 global_element_dof_from_node_dof switch

If global_element_dof_apply is set to -yes, and the element_intpnt_dof record does not exist, but node_dof records exist in the input file, you can either require that the element_intpnt_dof records will be initialised from the node_dof records, or will not be initialised from the node_dof records. If you set switch to -yes the element_intpnt_dof records will be initialised from the node_dof records. If you set switch to -no the element_intpnt_dof records will not be initialised from the node_dof records. Default, if global_element_dof_from_node_dof is not specified, switch is set to -no.

6.546 global_post_point_node_type node_type

With this record you can determine how records like post_point, control_print_dof_point and control_print_dof_line are evaluated. If node_type is set to -node the current nodal coordinates for elements are used to determine for which material point inside elements the dof’s should be determined; if you do an updated lagrange calculation in which the coordinates of nodes change, so the node records change, you get dof results for the material at the current moment presented on the point or line. If node_type is set to -node_start_refined the initial start nodal coordinates for elements are used to determine for which material point inside elements the dof’s should be determined; thus you get dof results for the material at the initial start moment presented on the point or line.

Default, if this record is not set, node_type is set to -node_start_refined.

6.547 groundflow_apply switch

If switch is set to -no, then the groundflow equation is skipped, and all groundflow data is ignored. This is done for all timesteps.

6.548 groundflow_consolidation_apply switch

If switch is set to -no, then the material divergence part in the groundflow equation is skipped. This is done for all timesteps.

Default switch is -no.
6.549  **groundflow_density** $\rho$

Density of ground water.

6.550  **groundflow_flux_edge_normal** *index flux*

Distributed prescribed water flux normal to the edge of an element. This distributed flux is translated into equivalent nodal flux on the edges of elements. Also the record **groundflow_flux_edge_normal_element** should be specified, and optionally the record **groundflow_flux_edge_normal_time** can be specified.

**Attention:** this option is only available for linear and quadratic isoparametric elements.

**Attention:** if this option is used INSIDE a FE mesh, then the elements on each side will get the distributed flux. So the total water flux will normally become zero since the normals of the elements at the side of the geometry are opposite.

6.551  **groundflow_flux_edge_normal_element** *index element_0 element_1 . . .

Restricts the elements to which the **groundflow_flux_edge_normal** record with the same *index* should be applied.

6.552  **groundflow_flux_edge_normal_element_group** *index element_group_0 element_group_1 . . .

Restricts the element groups to which the **groundflow_flux_edge_normal** record with the same *index* should be applied.

6.553  **groundflow_flux_edge_normal_element_node** *index element node_0 node_1 . . .

Selects the element and local node numbers for which the **groundflow_flux_edge_normal** record with the same *index* should be applied.

6.554  **groundflow_flux_edge_normal_element_node_factor** *index factor_0 factor_1 . . .

Nodal multiplication factors with which the **groundflow_flux_edge_normal** will be applied to the element of **groundflow_flux_edge_normal_element_node**. You need to specify a factor for each node on the side. Here *factor_0* is the multiplication factor for the first node on the side, etc.
6.555  `groundflow_flux_edge_normal_element_side index element_0 element_1 ... side`

Selects the elements and local side number for which the `groundflow_flux_edge_normal` record with the same index should be applied.

6.556  `groundflow_flux_edge_normal_factor index a_0 a_1 ... a_n`

This data item defines a polynomial in space. This polynomial gives a factor which is used as a multiplication factor for `groundflow_flux_edge_normal` records (with the same index). In this way, you can obtain coordinate dependent water fluxes.

In 1D the polynomial is \( a_0 + a_1 x \) (specify 2 values). In 2D the polynomial is \( a_0 + a_1 x + a_2 y \) (specify 3 values). In 3D the polynomial is \( a_0 + a_1 x + a_2 y + a_3 z \) (specify 4 values).

6.557  `groundflow_flux_edge_normal_geometry index geometry_entity_name geometry_entity_index`

Selects the area for which the `groundflow_flux_edge_normal` record with the same index should be applied. For example, `-geometry_line 1` can be used in 2D, indicating that the nodes on line 1 get the distributed flux. The total edge of an element must be inside the geometry for the force to become active. For 2D elements the border lines are edges. For 3D elements the border surfaces are edges.

6.558  `groundflow_flux_edge_normal_node index node_0 node_1 node_2 ...`

Selects the nodes for which the `groundflow_flux_edge_normal` record with the same index should be applied. The `node_0` etc. specify global node numbers.

6.559  `groundflow_flux_edge_normal_sine index start_time end_time freq_0 amp_0 freq_1 amp_1 ...`

Similar to `force_edge_sine`, now for water flux however.

6.560  `groundflow_flux_edge_normal_time index time load time load ...`

This record specifies a diagram which contains the factors with which the `groundflow_flux_edge_normal` record with the same index is applied. Linear interpolation is used to extend the `time load` values to the intervals between these pairs. Outside the specified time range a factor 0 is used.

If this record is not specified, the flux is applied at all times with a factor 1.

6.561  `groundflow_nonsaturated_apply index switch`

If switch is set to `-no`, then nonsaturated groundflow data (e.g., van Genuchten) will not be applied; only saturated data will be used. This is done for all timesteps.
Default switch is -yes.

6.562 groundflow_phreatic_bounda switch

If method is set to -yes, the phreatic level is used to automatically prescribe the hydraulic pressure head of nodes which are located on or above the phreatic level.

Default, if groundflow_phreatic_bounda is not specified, method is set to -yes.

6.563 groundflow_phreatic_level ...

Groundwater level.

In a 1D calculation this record should be given x value of the groundwater level. The groundwater is below that x-value.

In a 2D calculation this record should be given sets of x – y which specify the y level of the groundwater at several x locations; In 2D you need to give the x – y sets as follows:

- specify x – y sets for increasing x

In 3D the phreatic line is specified as follows. Denote the lowest x with x_0, the next higher x with x_1 etc. Denote the lowest y with y_0, the next higher y with y_1 etc. Denote the phreatic level z value for x_i y_j with z_ij. Then give the following:

- x_0 y_0 z_00 x_1 y_0 z_10 etc.
- x_0 y_1 z_01 x_1 y_1 z_11 etc.
- etc.

In 3D the number of points in x and y direction respectively should be set with nx and ny of the groundflow_phreatic_level_n record.

In nodes above the phreatic level the total pressure will be set to zero during the calculation.

As a special option in 2D and 3D, you can specify one value only, which sets a constant phreatic level of that value everywhere. In this special case, you do not need to specify groundflow_phreatic_level_n.

If you want to apply pore pressures directly following from the height under a phreatic level but not influenced be groundwater flow, then include a phreatic level and a boundary conditions for hydraulic pressure head changes:

... groundflow_phreatic_level ...
...
bounda_dof 20 ...-tpres

This has the advantage that the groundwater pressures don’t enter the system of equations, so that for combined soil - groundwater analysis a more effective solution can be obtained for the system of equations.
6.564 groundflow_phreatic_level_n nx ny

See groundflow_phreatic_level.

6.565 groundflow_phreatic_level_static switch

If switch is set to -yes, total pressures (pore pressures) in nodes for which the groundflow_phreatic_level holds, will be set equal to the static pressure.

This is convenient if the phreatic line is located above the mesh part to which it belongs; then nodes of the mesh will not get a boundary condition, so that the correct hydraulic pressure head can not be determined. And thus, it is necessary to determine the total pressure (pore pressure) from the difference of nodal coordinates and phreatic line height.

It is also convenient if you do not want to initialise and solve the hydraulic pressure heads with the groundflow storage equation. This saves computer memory and CPU time.

In the group_type for elements which should get the static groundflow pressure you need to add -groundflow.

6.566 groundflow_phreatic_level_multiple index . . .

The same as groundflow_phreatic_level, but now however several groundwater levels can be specified. For each groundflow_phreatic_level_multiple you should specify a separate value for index.

This option typically can be used if you have in vertical direction non-permeable layers separating the total domain in independent parts with each its own groundwater level.

You can specify with one of groundflow_phreatic_level_multiple_element or groundflow_phreatic_level_multiple_element_geometry or groundflow_phreatic_level_multiple_element_group or groundflow_phreatic_level_multiple_node the parts of the domain that belong to the groundwater level of groundflow_phreatic_level_multiple with the same index. Only one of these record can be used, you cannot combine them.

With groundflow_phreatic_level_multiple_n you specify nx ny in 3D again.

In the group_type for elements which should get the static groundflow pressure you need to add -groundflow.

6.567 groundflow_phreatic_level_multiple_element index element_0 element_1 . . .

Element numbers for groundflow_phreatic_level_multiple with the same index.

6.568 groundflow_phreatic_level_multiple_element_group index element_group_0 element_group_1 . . .

Element group numbers for groundflow_phreatic_level_multiple with the same index.
6.569  \textit{groundflow\_phreatic\_level\_multiple\_element\_geometry index element\_geometry\_0 element\_geometry\_1} \ldots

Element geometry numbers for \textit{groundflow\_phreatic\_level\_multiple} with the same index.

6.570  \textit{groundflow\_phreatic\_level\_multiple\_n nx ny}

See \textit{groundflow\_phreatic\_level\_multiple}.

6.571  \textit{groundflow\_phreatic\_level\_multiple\_node index node\_0 node\_1} \ldots

Node numbers for \textit{groundflow\_phreatic\_level\_multiple} with the same index.

6.572  \textit{groundflow\_phreatic\_level\_multiple\_static index switch}

If \textit{switch} is set to \textit{-yes}, total pressures (pore pressures) in nodes for which the \textit{groundflow\_phreatic\_level\_multiple\_static} holds, will be set equal to the static pressure.

This is convenient if the phreatic line is located above the mesh part to which it belongs; then nodes of the mesh will not get a boundary condition, so that the correct hydraulic pressure head can not be determined. And thus, it is necessary to determine the total pressure (pore pressure) from the difference of nodal coordinates and phreatic line height.

It is also convenient if you do not want to initialise and solve the hydraulic pressure heads with the groundflow storage equation. This saves computer memory and CPU time.

6.573  \textit{groundflow\_phreatic\_only switch}

If \textit{switch} is set to \textit{-yes} groundflow data is removed for groups which are not part of \textit{groundflow\_phreatic\_level\_multiple\_element\_group} records. Thus only groundflow data is retained for groups for which a multiple phreatic level is defined.

6.574  \textit{groundflow\_phreatic\_project switch}

If \textit{switch} is set to \textit{-yes}, the hydraulic pressure head which is imposed on nodes above the phreatic level uses the project coordinate on the phreatic level (smallest distance); thus not simply the distance in vertical direction. For some calculations that gives better groundwater velocities.

Default, it \textit{groundflow\_phreatic\_project} is not specified, \textit{switch} is set to \textit{-no}.

6.575  \textit{groundflow\_seepage\_eps eps}

The \textit{eps} specifies the tolerance if the groundflow seepage condition should be applied or not. If the inner product of the groundflow water flow direction with the normal outside the material is smaller then \textit{eps}, the seepage status will be set to closed, and the total pressure condition will not be applied (so that the boundary is really closed for water flow). If not specified, \textit{eps} is set to 0.1.
This record specifies an edge of the groundflow domain for which the groundwater is only allowed to flow outwards of the domain; flow into the domain is not allowed on that edge. The geometrical entity should be specified such that the normal of the geometry points outwards the material (so outwards the groundflow domain). This option comes handy when the point of groundwater flow exit is not known in advance of the calculation; it will be a result of the calculation instead.

Example:

```plaintext
... groundflow_seepage Geometry 10 -Geometry Line 100 ...
... Bounda_dof 20 -Geometry Line 100 -Total Pressure
 ... Bounda_time 20 0.0
```

In this example the total pressure (pore pressure) is set to 0 on the geometry line number 100, to account for free air at that edge. Since at that edge water cannot enter the domain the seepage option is applied to that edge. The result of these combined options is that on nodes with outward flow a total pressure 0 boundary condition is imposed, whereas on other nodes no boundary conditions is imposed (so that the flow is 0 at those nodes). The transition point between these outflow nodes and nodes with zero flow will be found automatically as a result of the calculation.

This record does the same as the `groundflow_seepage_geometry` record, but now however you specify node numbers at which the seepage condition holds. The `node_0` is the first node number, the `node_1` is the second node number, etc.

With this record you can specify the maximum allowed total pressure value. Any higher value resulting from the groundflow equations will be cutoff to this value. Default the `limit` is set to 0.

If `switch` is set to `-yes`, the calculation becomes axi-symmetrical for the group `index`. Each specified `x` coordinate becomes a radius and `y` becomes the length (=vertical) direction. The `z`-direction is the axi-symmetric direction. Specify only non-negative `x` coordinates, i.e. define the computational domain in the right half-plane.

This option is only available for groups with isoparametric 1D elements (bar2, ...), or isoparametric 2D elements (tria3, quad4, ...), or for 2D interface elements (quad4 interface, ...).
6.580 **group_beam_force_moment_plasti** index force_x_plasti_first_node
force_y_plasti_first_node force_z_plasti_first_node moment_x_plasti_first_node
moment_y_plasti_first_node moment_z_plasti_first_node force_x_plasti_second_node
force_y_plasti_second_node force_z_plasti_second_node moment_x_plasti_second_node
moment_y_plasti_second_node moment_z_plasti_second_node

With this record you can set ideally plastic limits on forces and moments in beam elements. You can specify different values for each of the local x, y and z-directions, so that a different plastic behavior can be specified for beams having different properties in the different directions. You can also different values for each of the two nodes.

6.581 **group_beam_inertia** index Iyy Izz J

Bending and torsion properties for beam elements. Here $I_{yy}$ is the area moment of inertia for bending along the local beam y axis, and $I_{zz}$ is the area moment of inertia for bending along the local beam z axis, and $J$ is the polar moment of inertia for torsion along the local beam x axis.

See also **beam_rotation** in the initialisation part.

The *index* specifies the element_group, see **element_group**.

6.582 **group_beam_memory** index memory_type

Memory model for beam; either -updated_linear, -updated or -total_linear. The -updated model is a geometrically nonlinear model which takes large beam rotations into account. The *index* specifies the element_group, see **element_group**.

6.583 **group_beam_direction_z** index dir_z,x dir_z,y dir_z,z

This record specifies the local beam z direction in global space. If **group_beam_direction_z** is not specified in 2D then 0 0 1 will be used. If **group_beam_direction_z** is not specified in 3D then an arbitrary direction perpendicular to the beam length axes will be used.

The local beam axes will be placed in the **element_beam_direction** record after the calculation.

The *index* specifies the element_group, see **element_group**.

See also **group_beam_direction_z_reference_point** for automatic beam z-axis towards a reference point.

6.584 **group_beam_direction_z_reference_point** index point_x point_y
point_z

This data record defines a reference point that allows you to influence the local beam z-direction. The local beam z-direction will be setup as follows:

- The length direction of the beam is determined, that is the local beam x-axis.
- A vector is taken from the beam middle point to the reference point.
- The part of this vector perpendicular to the length direction defines the local beam z-axis.
The above procedure ensures that the beam z-axis is perpendicular to the length direction, and that the z-axis points as much as possible to the reference point. As a typical example, you can use this option to take care that the local beam z-axis points to the middle of a tunnel, which is convenient if a tunnel lining with the local z-axis towards the tunnel middle; to do so specify the middle point of the tunnel axis as reference point point_x point_y point_z.

6.585 group Beam Young index E

Young’s modulus for a beam (for bending moment calculation). The index specifies the element_group, see element_group.

6.586 group Beam Shear index G

Shear modulus for a beam (for torsion moment calculation). The index specifies the element_group, see element_group.

6.587 group Condif Absorption index a

Absorption coefficient. The index specifies the element_group, see element_group.

6.588 group Condif Capacity index C

Heat capacity. The index specifies the element_group, see element_group.

6.589 group Condif Conductivity index k_x k_y k_z

Heat conductivity in x, y and z direction respectively. As a special option you can also specify one value only, which then will be used in each direction. The index specifies the element_group, see element_group.

6.590 group Condif Density index density

Density for convection-diffusion equation. The index specifies the element_group, see element_group.

6.591 group Condif Flow index beta_1 beta_2 beta_3

Known flow field. In 1D only beta_1 should be specified, etc. The index specifies the element_group, see element_group.

6.592 group Contact Spring Direction index dirN_x dirN_y dirN_z

Normal direction of a contact_spring. The index specifies the element_group, see element_group.

As an alternative, you can specify element Contact Spring Direction which allows for specification of the direction for each element separately.
As yet another alternative you can set \textit{switch} in \texttt{group\_contact\_spring\_direction\_automatic} to \texttt{-yes}. Then the contact spring will automatically determine the directions.

\begin{verbatim}
6.593  group_contact_spring_direction_automatic  index switch
\end{verbatim}

See \texttt{group\_contact\_spring\_direction}.

\begin{verbatim}
6.594  group_contact_spring_plasti_cohesion  index c
\end{verbatim}

The normal contact force $F_N$ is not allowed to become larger than cohesion $c$ in tension (positive values of $F_N$). If it would become larger, then the contact is broken, a gap is assumed and the contact force $F_N$ is put to 0. To have really a positive $F_N$ for extension of the contact spring, the order of the two nodes as specified in the \texttt{element} record for the contact spring should be correct.

Notice that when you use \texttt{control\_mesh\_generate\_contact\_spring} to obtain the contact spring elements, you are not sure what the first and what the second node of an element will be, and thus you should not use this \texttt{group\_contact\_spring\_cohesion} record. Otherwise, it is not important what you use as first and second node, so that \texttt{control\_mesh\_generate\_contact\_spring} can be used safely.

If this \texttt{group\_contact\_spring\_plasti\_cohesion} is not specified, infinite cohesion is assumed.

The \texttt{index} specifies the \texttt{element\_group}, see \texttt{element\_group}.

\begin{verbatim}
6.595  group_contact_spring_plasti_friction  index f
\end{verbatim}

With this record you can specify a fixed friction coefficient for contact springs. If this record is not specified, a very large value for $f$ will be applied.

The \texttt{index} specifies the \texttt{element\_group}, see \texttt{element\_group}.

See also \texttt{group\_contact\_spring\_stiffness} and \texttt{group\_contact\_spring\_friction\_automatic}.

\begin{verbatim}
6.596  group_contact_spring_plasti_friction_automatic  index switch
\end{verbatim}

If \texttt{switch} is set to \texttt{-yes}, the friction coefficient for contact springs will be determined from the plasticity law angle of neighboring elements. For a neighboring \texttt{group\_materi\_plasti\_mohr\_coul} the friction coefficient $f$ will be set to $f = (2/3)\tan\phi$ with $\phi$ the friction angle in the mohr-coulomb law of the neighboring elements. For a neighboring \texttt{group\_materi\_plasti\_diprisco} the friction coefficient $f$ will be set to a value depending on the parameter $\gamma$ of that law.

If no neighbor elements with appropriate material law are found, then $f$ will be set to 0.2.

The \texttt{index} specifies the \texttt{element\_group}, see \texttt{element\_group}. See also \texttt{group\_contact\_spring\_direction\_automatic}.

\begin{verbatim}
6.597  group_contact_spring_direction_automatic_planes  index switch_x switch_y switch_z
\end{verbatim}

With this option you can help the \texttt{group\_contact\_spring\_friction\_automatic} by telling in which planes the automatically determined spring direction is allowed to be. If a switch is set to
-yes, then the direction may have a component in that plane. If a switch is set to -no, then the direction may not have a component in that plane. Default all switches are -yes.

The index specifies the element_group, see element_group.

6.598 group_contact_spring_memory index memory_type

Memory model for contact_spring; either -updated_linear, -total_linear. The index specifies the element_group, see element_group.

6.599 group_contact_spring_stiffness index kN kT

Stiffnesses for contact springs. The force $F_N$ in normal direction of the contact spring is determined from $F_N = k_N u_N$ where $u_N$ is the normal displacement difference of the two nodes (that is, the displacement of the second node in normal direction minus the displacement of the first node in normal direction). The first tangential force $F_{T1}$ of the contact spring is determined from $F_{T1} = k_T u_{T1}$ where $u_{T1}$ is the tangential displacement difference of the two nodes in the first tangential direction; the same is done for the second tangential force. The total tangential force $\sqrt{F_{T1}^2 + F_{T2}^2}$ cannot exceed $f F_N$ with $f$ friction coefficient; then frictional slip occurs and the total tangential force is set to $f F_N$. To model continuing stick between two bodies just put the friction coefficient $f$ very high.

In 1D the parameters $k_T$ and $f$ will not be used (but should be specified as dummies nevertheless).

The index specifies the element_group, see element_group.

See also group_contact_spring_friction and group_contact_spring_friction_automatic.

6.600 group_dof_initial index dof_0 dof_1 \ldots

Same as element_dof_initial, now specified for a group of elements however.

6.601 group_dof_initial_specific_number index dof

Same as element_dof_initial_specific_number, now specified for a group of elements however.

6.602 group_dof_initial_specific_value index value_0 value_grad_x value_grad_y value_grad_z

Same as element_dof_initial_specific_value, now specified for a group of elements however.

6.603 group_groundflow_capacity index C

Capacity in ground water flow equation. The index specifies the element_group, see element_group.
6.604 group_groundflow_consolidation_apply index switch

If switch is set to -no consolidation will not be applied for the elements of the group.

6.605 group_groundflow_expansion index \( \alpha \)

Thermal expansion coefficient for ground water, for a combined groundwater with temperature analysis. The index specifies the element_group, see element_group.

6.606 group_groundflow_nonsaturated_eps_permeability index \( \varepsilon \)

The nonsaturated law lowers the permeability relative to the specified linear permeability as specified in group_groundflow_permeability. With this group_groundflow_nonsaturated_eps_permeability you can specify the lowest allowed factor for reducing the permeability.

6.607 group_groundflow_nonsaturated_vangenuchten index \( S_{\text{residu}} \) \( S_{\text{sat}} \)

Parameters for non-saturated van Genuchten ground water flow, see the theory section. The index specifies the element_group, see element_group.

Since the van-Genuchten law is highly nonlinear, convergence of the calculation can be difficult. Always check if the calculation converges by printing post_node_rhside_ratio. You can try including inertia to improve convergence. Alternatively for calculations without inertia you can specify a relaxation factor with control_relaxation (try a factor of 0.1 or so).

6.608 group_groundflow_permeability index \( p_{\text{x}} \) \( p_{\text{y}} \) \( p_{\text{z}} \)

Permeability coefficient in ground water flow, in each space direction. In 1D you only should specify \( p_{\text{x}} \), etc. If you specify only value, then that will be used in each direction. The index specifies the element_group, see element_group.

6.609 group_groundflow_total_pressure_tension index plastic_tension_minimum water_height

Using this option you can control that the water pressure in an element is at least the value as determined from the specified water_height. More precise, if the static water pore pressure as determined from the water density, the gravity and the water_height exceeds the pore water pressure from the groundflow equation (in absolute terms), this static water pressure actually is used. This is only done if the largest eigenvalue of materi_strain_plastic_tension exceeds plastic_tension_minimum. To calculate the eigenvalues of materi_strain_plastic_tension you need to include post_calcul -materi_strain_plastic_tension -prival in the input file.

This option comes handy to take care that in cracks in concrete actually the largest water pressure from an environment is used. It so ensures that a critical safety analysis for concrete cracking is obtained.
Here *method* sets the integration method for bars, quad en hex elements. You can either set *method* to *-gauss*, *-lobatto* or *-reduced*.

The gauss and lobatto integration method are explained in text books on finite element methods. The reduced method is a weighted combination of lobatto and 1-point cell centered integration. The weighting is set by the `group_integration_method_reduced_factor` record.

If this record is not set, the default method as described in `group_integration_points` is chosen.

It is advised to keep the default method, so not specify this `group_integration_method` record, unless you know what you are doing.

When *factor*=1 the reduced method is in fact lobatto integration. When *factor*=*infinite* the reduced method is in fact 1-point cell centered integration. When the *factor* is in between 1 and infinity the integration is done with a weighted combination of the lobatto and 1-point cell centered integration.

Default the *factor* is set to 10.

Here *type* sets the number of integration points in an element. It should be set to *-normal*, *-minimal* or to *-maximal*.

For *-tria3* elements the integration point will be located in the middle with *-minimal* integration, or a four-point integration scheme will be used with *-maximal* integration.

For *-tria6* elements a seven-point scheme will be used for *-maximal* and a four-point scheme will be used with *-minimal* integration.

For *-tet4* elements the integration point will be located in the middle with *-minimal* integration, or a five-point integration scheme will be used with *-maximal* integration.

For *-tet10* elements a five-point scheme will be used for *-minimal* and a ten-point scheme will be used with *-maximal* integration.

For other elements, if *-minimal* is used then the number of integration points in a direction is set equal to the number of nodes in the direction minus 1, and gauss integration is used. If for the other elements *-maximal* is used then the number of integration points in a direction is set equal to the number of nodes in the direction; gauss integration is used, but in case inertia is applied then lobatto integration will be used.

Default *-minimal* is used for *-bar2*, *-tria3*, and *-tet4* elements; it is default *-maximal* otherwise.

If *type* is set to *-normal*, the default integration will be used.

The above is valid for normal isoparametric elements. For interface elements default lobatto integration is used (integration points in nodes).

It is advised to keep the default method, so not specify this `group_integration_points` record.

The *index* specifies the element _group_, see `element_group`.
6.613 group_interface index switch

With this record, you set that the element with element group index will act as an interface element by setting switch to -yes. This is available for -quad4, -quad6, -hex8, -hex18, -prism6 and -prism12.

See group_interface_* which data can be set for interfaces.

In interfaces strains are displacement differences between the opposite interface sides.

6.614 group_interface_condif_conductivity index k

The 'index' specifies the group number. The conductivity $k$ specifies the heat flow in interface thickness direction per unit temperature difference. Thus the conductivity is not the material conductivity but the conductivity of the layer simulated by the interface incorporating the thermal thickness of the interface. The 'conductivity' has units [power]/[temperature*length] in 2D, and [power]/[temperature*length*length] in 3D.

6.615 group_interface_gap index gap

By specifying this record you can account for initial empty space between the sides of an interface element. Only when the sides displacements are such that this initial gap is closed, then the interface element will start to generate stresses. This is accomplished in the program by setting the stiffness of the interface element to zero or a very small value as long as the gap is not closed.

As a special case, setting gap to 0 means that the gap option is inactive and will not be used.

6.616 group_interface_groundflow_capacity index C

This record specifies the capacity for interface elements.

6.617 group_interface_groundflow_permeability index pe

This record specifies the permeability per unit length in 2D or unit area in 3D for interface elements.

6.618 group_interface_materi_elasti_stiffness index kn kt,first kt,second

This record allows you to specify a normal stiffness and tangential shear stiffnesses for discrete interface elements with -materi in group_type. Normal stresses in the interface element follow from normal strains multiplied with $kn$ (stress,normal = $kn$ * strain,normal). Shear stresses in the interface element in the first tangential direction follow from shear strains in the first tangential direction multiplied with $kt,first$ (stress,shear,first = $kt,first$ * shear,gamma,first = 2 * $kt,first$ * strain,shear,first). Shear stresses in the interface element in the second tangential direction follow from shear strains in the second tangential direction multiplied with $kt,second$ (stress,shear,second = $kt,second$ * shear,gamma,second = 2 * $kt,second$ * strain,shear,second). The $kt,second$ should be specified for 3D interfaces only.

Too high values for interface stiffness will cause convergence problems in calculations. Thus, if you are running a calculation with interface elements and you are experiencing convergence problems
please try lower values for the interface stiffnesses. Typically the normal interface stiffness can be chosen as 10 times the Young’s modulus of the neighbouring isoparametric element divided by the length of that element in normal direction. Typically the tangential interface stiffness can be chosen as half of the normal interface stiffness.

A 3d example:

```
... 
  number_of_space_dimensions 3
...
  group_interface_materi_elasti_stiffness 0 0.10000e+11 0.50000e+10 0.50000e+10
...
```

6.619 group_interface_materi_expansion_normal index expansion_coefficient_normal

The 'index' specifies the group number. The expansion_coefficient_normal specifies the thermal strain expansion in interface thickness direction per unit temperature in the interface. The temperature is the average of the temperature of the both sides at the location of the integration point. This option is only available if group_interface_materi_memory is set to -total_linear or -updated_linear. Furthermore, materi_strain_elasti should be initialised.

6.620 group_interface_materi_memory index memory_type

Either memory_type should be set to -updated_linear or -total_linear.

6.621 group_interface_materi_plasti_mohr_coul_direct index phi c phi-flow

Mohr-coulomb plasticity model for interfaces. The angles are in radians. The cohesion c has stress unit (so just the same as for group_materi_plasti_mohr_coul in normal isoparametric elements). The maximum friction force in the interface is \( c + F_n \times \tan(\phi) \) where \( c \) is the cohesion, \( \phi \) is the friction angle in radians and \( F_n \) is the normal force (which is a negative value under compression).

6.622 group_interface_materi_plasti_tension_direct index switch

If switch is set to -no then the stresses are set to 0 if the interface normal strain is positive.

This group_interface_materi_plasti_tension_direct is not allowed in combination with group_interface_gap.

Default, if group_interface_materi_plasti_tension_direct is not specified, switch is set to -yes.

6.623 group_interface_materi_residual_stiffness index factor

With factor you can set the part of the original stiffness to be used as stiffness in opened interfaces. Default, if this record is not specified, the factor is set to some small value.
6.624  
\texttt{group\_interface\_groundflow\_total\_pressure\_tension} \ \texttt{index strain\_normal\_water\_height}

Using this option you can control that the water pressure in an interface element is at least the value as determined from the specified \texttt{water\_height}. More precise, if the static water pore pressure as determined from the water density, the gravity and the \texttt{water\_height} exceeds the pore water pressure from the groundflow equation (in absolute terms), this static water pressure actually is used. This is only done if the interface normal strain (displacement different between interface sides) exceeds \texttt{strain\_normal\_minimum}.

This option comes handy to take care that in cracks in concrete actually the largest water pressure from an environment is used. It so ensures that a critical safety analysis for concrete cracking is obtained.

6.625  
\texttt{group\_interface\_tangential\_reference\_point} \ \texttt{index point\_x point\_y point\_z}

This data record defines a reference point that allows you to influence the tangential directions in a 3D interface element. The tangential directions will be setup as follows:

- The normal direction to the interface plane is determined.
- A vector is taken from the integration point in the interface element to the reference point.
- The part of this vector perpendicular to the normal direction defines the first tangential direction.
- The outer product of the normal direction and the first tangential direction gives the second tangential direction.

The above procedure ensures that the tangential directions are perpendicular to the normal direction, and that the first tangential directions points as much as possible to the reference point. As a typical example, you can use this option to take care that the first tangential direction points to the middle of a tunnel, so the first tangential interface direction equals in fact the tunnel radial direction; to do so specify the middle point on the tunnel axis as reference point \texttt{point\_x point\_y point\_z}.

If this \texttt{group\_interface\_tangential\_reference\_point} is not specified, it is only certain that the tangential directions are in plane of the interface (perpendicular to the normal direction), but are not defined otherwise.

See also \texttt{element\_interface\_intpnt\_direction}.

6.626  
\texttt{group\_materi\_damage\_mazars} \ \texttt{index epsilon0 a_t b_l a_c b_c beta}

Parameters for the Mazars damage law. The \texttt{index} specifies the element \_group, see \texttt{element\_group}.

6.627  
\texttt{group\_materi\_damping} \ \texttt{index d}

Material damping coefficient \texttt{d}. See also the dynamics section near the end of this manual.

The \texttt{index} specifies the element \_group, see \texttt{element\_group}.
6.628 **group_materi_damping_method** *index method*

See **group_materi_damping**.

6.629 **group_materi_density** *index density*

Density for material flow equation. The *index* specifies the *element_group*, see **element_group**.

6.630 **group_materi_density_groundflow** *index density_wet density_dry*

Density for material flow equation when a calculation is performed in combination with groundflow. If the element is filled with groundwater the *density_wet* will be used and otherwise the *density_dry* will be used. To determine if an element is filled with water, tochnog does not the following: If **post_calcul -groundflow_pressure -total_pressure** is put in the input file then total pressures (pore pressures) are calculated. Then if the pore pressure in an element is negative the wet density is taken. Otherwise the dry density. If **post_calcul -groundflow_pressure -total_pressure** is NOT put in the input file the total pressures are not calculated. Then tochnog looks if a phreatic level is given; if so, then if an element is below the phreatic level the wet density is used, otherwise if an element is above the phreatic level the dry density is used.

Here *density_wet* is the amount of kg of soil + water in a unit volume. And *density_dry* is the amount of kg of soil in a unit volume.

The *index* specifies the *element_group*, see **element_group**.

In case total pressures are calculated

In case total pressures are calculated from the **post_calcul groundflow_pressure -total_pressure** command, the *density_wet* will be used if the total pressure is smaller then 0, whereas *density_dry* will be used if the total pressure is larger or equal to 0.

In case total pressures are not calculated but a phreatic level is specified

In case an element is above a specified phreatic level the *density_dry* will be used. In case an element is below a specified phreatic level, the *density_wet* will be used.

In other cases

In other cases *density_dry* will be used.

6.631 **group_materi_elasti_borja_tamagnini** *index* \( G_0 \ \alpha \hat{\kappa} p_r \)

Elastic data for the modified Borja Tamagnini model, see [1]. The *index* specifies the *element_group*, see **element_group**.

6.632 **group_materi_elasti_c** *index 81 values*

With this record you can directly specify the 81 values of the linear material stiffness \( C_{ijkl} \) which will be used to calculate stresses from strains with \( \sigma_{ij} = C_{ijkl} \epsilon_{kl} \). Here \( \sigma_{ij} \) is the stress matrix and \( \epsilon_{kl} \) is the strain matrix.

The sequence of strains and stresses is \( xx, xy, xz, yx, yy, yz, zx, zy \) and \( zz \).
The 81 values should be specified row by row, where each row contains 9 values. See also group_materi_elasti_c_direction.

6.633 group_materi_elasti_c_direction index dir_0 dir_1 dir_2

This record specifies local axes for which the group_materi_elasti_c is specified. In total 9 values need to be specified, first the 3 values for dir_0, then the 3 values for dir_1 and then the 3 values for dir_2. Default, if this record is not specified, the global axes will be used.

6.634 group_materi_elasti_camclay_g index G

Elastic data G for the modified CamClay model. The index specifies the element_group, see element_group.

6.635 group_materi_elasti_camclay_poisson index ν

Elastic data ν for the modified CamClay model. This option is alternative to the group_materi_elasti_camclay_g option (so, only one of both can be defined). With this option the poisson ratio ν is assumed constant, and is used as follows:

\[ G = \frac{3}{2} K(1 - 2\nu)/(1 + \nu) \]

The index specifies the element_group, see element_group.

6.636 group_materi_elasti_camclay_pressure_min index pressure_min

This specifies a minimal allowed value for the pressure in the calculation of the bulk modulus for the camclay model. In the calculation pressures below pressure_min will be set to pressure_min. This prevents numerical problems for very low bulk modulus \(K\) values.

The index specifies the element_group, see element_group.

6.637 group_materi_elasti_compressibility index co

Compressibility for materials. A positive value should be used. The index specifies the element_group, see element_group.

6.638 group_materi_elasti_hardsoil index \(E_{50}^{ref} \) \(\sigma_{50}^{ref} \) \(\nu_{50} \) m \(E_{ur}^{ref} \) \(\sigma_{ur}^{ref} \) \(\nu_{ur} \)

Elasticity data for Hardening Soil model. The index specifies the element_group, see element_group.

6.639 group_materi_elasti_k0 index K0

Elastic data \(K0\). When this data is specified, and also control_materi_elasti_k0 is set to -yes, then the \(K0\) parameter will be used in the elastic stress law with group_materi_elasti_young or group_materi_elasti_young_power and group_materi_elasti_poisson, or with group_materi_
In fact it will be used to determine the poisson coefficient consistent with the $K_0$; this poisson coefficient is used in the elastic stress law.

This group elasti_k0 in combination with control elasti_k0 is a convenient method to get $'K_0 stresses'$ when imposing gravity in a geotechnics calculation. After gravity is imposed simply do not set the control elasti_k0 anymore, so that the normal group elasti_poisson will be used in the remaining steps.

For $K_0 > 0.95$ Tochnog will take 0.95. $K_0$ exceeding 1 (or 0.95) may lead to ill-conditioned calculations.

6.640 group_elasti_lade index $B \ R \ \lambda$

Elastic data $B = 0$, $R$, $\lambda$ for the Lade model. The index specifies the element_group, see element_group.

6.641 group_elasti_poisson index poisson

Poisson ratio for solid. The index specifies the element_group, see element_group.

6.642 group_elasti_poisson_power index $\nu_0 \ \nu_1 \ \nu_2 \ p_1 \ \alpha$

Power law poisson ratio for solid material. See the theory part. The index specifies the element_group, see element_group.

6.643 group_elasti_shear_factor index factor

Specifying this record causes the shear stiffness following from a specified young and poisson to be multiplied with factor. This provides a convenient way to test in a calculation what the effect of low shear stresses is. The index specifies the element_group, see element_group.

6.644 group_elasti_stress_pressure_history_factor index factor

This record allows you to model a different soil stiffness when first loading or unloading/reloading instead. The materi_stress_pressure_history should be initialised, which records the maximum soil pressure that occurred in history. If the current pressure is smaller than the largest pressure in history, the material is unloading or reloading, and the stiffness will be multiplied with factor. If the current pressure is the larger than the largest pressure from history, then this current pressure becomes the maximum history pressure, and the stiffness will not be multiplied with factor. The factor typically may be $\frac{1}{3}$.

This group elasti_stress_pressure_history_factor can be combined with the young as specified by group_elasti_young or the young calculated from group_elasti_young_power.
6.645 group_materi_elasti_transverse_isotropy index $E_1$ $E_2$ $\nu_1$ $\nu_2$ $G_2$ dir_x dir_y dir_z

Specifies the unique direction (dir_x dir_y dir_z) and elastic moduli in the transverse isotropic model. Here $E_2$ is the young modulus in the unique direction, $E_1$ is the young modulus in both directions perpendicular to the unique direction, etc. The index specifies the element_group, see element_group.

6.646 group_materi_elasti_volumetric_poisson index $\nu$

See group_materi_elasti_volumetric_young_values

6.647 group_materi_elasti_volumetric_young_order index $n$

See group_materi_elasti_volumetric_young_values

6.648 group_materi_elasti_volumetric_young_values index $\epsilon_0$ $\sigma_0$ $\epsilon_1$ $\sigma_1$ ...

This is a special record to model the volumetric stress part of a nonlinear material, given the experimental results of a volumetric compression test (compression in one direction, fixed size in other two directions).

The table $\epsilon_0 \sigma_0 \epsilon_1 \sigma_1 \ldots$ specifies the strain-stress results for the volumetric compression test. Together with the poisson ratio as specified in group_materi_elasti_volumetric, an isotropic law in a nonlinear Young’s modulus and a constant poisson ratio is fitted to this experiment. The Young modulus in fact is taken as the polynomial expansion $E_0 + E_1 \epsilon_0 + E_2 \epsilon_0^2 + \ldots + E_{n-1} \epsilon_0^{n-1}$ where $n$ denotes the order of the polynomial expansion (as given in group_materi_elasti_volumetric_young_order).

The poisson ratio should be taken very high, say 0.4999999 or so, to ensure that the resulting law only models volumetric stresses. Then afterwards a normal young-poisson isotropic law (group_materi_elasti_young and group_materi_elasti_poisson) can be added to get an extra deviatoric part.

6.649 group_materi_elasti_young index $E$

Young’s modulus for solid material. The index specifies the element_group, see element_group.

6.650 group_materi_elasti_young_polynomial index $E_0$ $E_1$ ...

Polynomial parameters for strain dependent Young’s modulus for solid material. See the theory part. The index specifies the element_group, see element_group.

6.651 group_materi_elasti_young_power index $E_0 E_1 E_2 p \_1 \alpha$

Power law Young’s modulus for solid material. See the theory part. The index specifies the element_group, see element_group.
If you want to get the calculated young as output, initialise with `materi_history_variable 1`; the history variable will be filled with the calculated young, and can be plotted for example in GID.

**6.652 group_materi_elasti_young_user index switch**

If `switch` is set to `-yes` the user supplied routine `user_young` will be called. There the youngs modulus should be calculated from the solution fields and the stress history. Typically degradation of material stiffness for cyclic loading can be programmed with this user specified routine.

You can plot in gid the values for the young as follows:

```plaintext
... print_group_data -group_materi_elasti_young ...
...```

**6.653 group_materi_expansion_linear index α**

Linear expansion coefficient. The `index` specifies the element_group, see `element_group`.

**6.654 group_materi_expansion_volume index β**

Volume expansion coefficient. The `index` specifies the element_group, see `element_group`.

**6.655 group_materi_factor index factor**

This factor comes convenient if your material stress law is specified in other units then you actually want in your calculation. Then you can specify `factor` to take care that your material stresses become consistent with the remaining part of the input file. For example, if you want your input file to work with kPa but your material stress law works with MPa then simply set `factor` to 1000.

**6.656 group_materi_failure_crunching index threshold delete_time**

If the compression strain in an element exceeds `threshold`, the element is considered to be fail. The element will be slowly deleted. It is totally deleted if the `delete_time` has passed.

The `index` specifies the element_group, see `element_group`.

**6.657 group_materi_failure_damage index threshold delete_time**

If the damage in an element exceeds `threshold`, the element is considered to be fail. The element will be slowly deleted. It is totally deleted if the `delete_time` has passed.

The `index` specifies the element_group, see `element_group`.
If the plastic parameter kappa in an element exceeds \textit{threshold}, the element is considered to be fail. The element will be slowly deleted. It is totally deleted if the \textit{delete\_time} has passed.

The \textit{index} specifies the \textit{element\_group}, see \textit{element\_group}.

If the tensile strain in an element exceeds \textit{threshold}, the element is considered to be fail. The element will be slowly deleted. It is totally deleted if the \textit{delete\_time} has passed.

The \textit{index} specifies the \textit{element\_group}, see \textit{element\_group}.

If the void fraction in an element exceeds \textit{threshold}, the element is considered to be fail. The element will be slowly deleted. It is totally deleted if the \textit{delete\_time} has passed.

The \textit{index} specifies the \textit{element\_group}, see \textit{element\_group}.

Set \textit{switch} to \texttt{-yes} if you want to activate the user supplied routine for material history variables. The \textit{index} specifies the \textit{element\_group}, see \textit{element\_group}.

Specify parameters for the user supplied routine for material history variables. The \textit{index} specifies the \textit{element\_group}, see \textit{element\_group}.

Parameters for Besseling Hyper elastic rubber model. The \textit{index} specifies the \textit{element\_group}, see \textit{element\_group}.

Parameters for Blatz-Ko model. The \textit{index} specifies the \textit{element\_group}, see \textit{element\_group}.

Parameters for Mooney-rivlin hyper elastic rubber model. The \textit{index} specifies the \textit{element\_group}, see \textit{element\_group}.
6.666  group_materi_hyper_neohookean  index $K_1$

Parameter for Neo-Hookean hyper elastic rubber model. The index specifies the element_group, see element_group.

6.667  group_materi_hyper_reduced_polynomial  index $K_1$ $K_2$ \ldots

Parameters for reduced polynomial hyper elastic rubber model. The index specifies the element_group, see element_group.

6.668  group_materi_hyper_volumetric_linear  index $K$

Parameter for the linear volumetric hyperelasticity model. The index specifies the element_group, see element_group.

6.669  group_materi_hyper_volumetric_murnaghan  index $K\beta$

Parameter for the murnaghan volumetric hyperelasticity model. The index specifies the element_group, see element_group.

6.670  group_materi_hyper_volumetric_ogden  index $K\beta$

Parameter for the ogden volumetric hyperelasticity model. The index specifies the element_group, see element_group.

6.671  group_materi_hyper_volumetric_polynomial  index $K_0$ $K_1$ \ldots

Parameters for the polynomial volumetric hyperelasticity model. The index specifies the element_group, see element_group.

6.672  group_materi_hyper_volumetric_simo_taylor  index $K$

Parameter for the simo-taylor volumetric hyperelasticity model. The index specifies the element_group, see element_group.

6.673  group_materi_maxwell_chain  index $E_0$ $t_0$ \ldots $E_{n-1}$ $t_{n-1}$

In total $n$ parallel maxwell chains are defined with stiffness $E_0$, relaxation time $t_0$, etc.. The number $n$ should equal materi_maxwell_stress in the input initialization part. The index specifies the element_group, see element_group.
6.674  group_materi_membrane index switch

If switch is set to -yes the $zz$ stress becomes zero in 2D and the $yy$ and $zz$ stress become zero in 1D (in combination with axi-symmetry in 1D, only the $yy$ stress becomes zero since $zz$ is the axi-symmetric direction). So this option models plane stress conditions.

If group_materi_membrane is not used the plane strain conditions are used. Always the $z$-thickness is 1. in 3D, and the $y$, and $z$-thickness are 1. in 2D; see however also volume_factor.

The group_materi_membrane option cannot be used in combination with group_materi_elasti_compressibility, group_materi_hyperelasticity and group_materi_viscosity.

The index specifies the element_group, see element_group.

6.675  group_materi_memory index memory_type

Either memory_type should be set to -updated, -updated_jaumann, -updated_linear, -total or -total_linear. See the theoretical part for some explanation.

For an linear total Lagrange solid the input file may look like, and is recommended for most solid calculations:

```
... materi_velocity materi_displacement materi_strain_total materi_stress end_initia ...
... node 1 ...
node 2 ...
... group_materi_memory 0 -total_linear group_materi_elasti_young 0 ...
... end_data
```

For a large deformation total Lagrange solid with a straightforward decomposition of the deformation tensor into a rotation tensor and a stretch tensor the input file may look like

```
... materi_velocity materi_displacement materi_strain_total
```
For an updated Lagrange solid the input file may look like

...  
materi_velocity  
materi_velocity_integrated  
materi_stress  
end_initia  
...  
mesh -follow_material ...  
...  
node 1 ...  
node 2 ...  
...  
group_materi_memory 0 -updated  
group_materi_elasti_young 0 ...  
...  
end_data  

Notice that for an updated Lagrange formulation you should always set that the mesh follows the material.

For a fluid the input file may look like

...  
materi_velocity  
materi_stress  
end_initia  
...  
(use Eulerian mesh)  
mesh -fixed_in_space ...  
timestep_predict_velocity -yes  
...  
node 1 ...  
node 2 ...  
...  
group_materi_memory 0 -updated_linear  
group_materi_viscosity 0 ...  
group_materi_elasti_compressibility 0 ...  
...  
end_data  

The index specifies the element_group, see element_group.
6.676  \texttt{group\_materi\_plasti\_bounda} index index\_0 index\_1 \ldots

With this option, you can model reduction of friction of soil material and alike granular materials on walls. Set \textit{index\_0, index\_1} etc. to the index of the \texttt{bounda\_dof} records for which you want to use this reduction. We define an element to be on a wall when at least one of the velocities (displacements) of the elements is prescribed (via \texttt{bounda\_dof}). As a special option, you can use \texttt{-all} which indicates that the \texttt{bounda\_dof} records for all indeces will be used.

The reduction of friction is done for
\begin{itemize}
  \item \texttt{group\_materi\_plasti\_mohr\_coul},
  \item \texttt{group\_materi\_plasti\_mohr\_coul\_direct},
  \item \texttt{group\_materi\_plasti\_druck\_prag},
  \item \texttt{group\_materi\_plasti\_hardsoil}, if specified,
\end{itemize}
by reducing the friction angle \textit{phi} and dilatancy angle \textit{phiflow} and cohesion \textit{c} of the granular material with a factor \((2./3.)\).

This is done for \texttt{group\_materi\_plasti\_camclay}, if specified, by reducing \textit{M} with a factor \((2./3.)\).

This is done for \texttt{group\_materi\_plasti\_hypo\_*}, if specified, by reducing deviatoric stress increments with a factor \((2./3.)\).

The \textit{index} specifies the element\_group of the granular material, see \texttt{element\_group}.

See also \texttt{group\_materi\_plasti\_bounda\_factor} and \texttt{group\_materi\_plasti\_mohr\_coul\_direct\_wall}.

6.677  \texttt{group\_materi\_plasti\_bounda\_factor} index factor

With this record you can specify a factor other then the default \((2./3.)\) used by the \texttt{group\_materi\_plasti\_bounda} record. You need to specify a factor for each of \textit{index\_0, index\_1} etc. If you specify one factor only it will be used for all boundaries.

The \textit{index} specifies the element\_group of the granular material, see \texttt{element\_group}.

6.678  \texttt{group\_materi\_plasti\_coord\_limit} index coord\_limit

To prevent plasticity problems near free surfaces, you can require that Tochnog neglects plasticity laws if the vertical coordinate exceeds \textit{coord\_limit}. This option is not available for hypoplasticity laws, since for these laws nonlinear elasticity and plasticity are defined by one law, so the plasticity part cannot be suppressed by itself.

6.679  \texttt{group\_materi\_plasti\_camclay} index \(M\ \kappa\ \lambda\)

Plastic data \(M, \kappa\) and \(\lambda\) for the modified CamClay model. The \textit{index} specifies the element\_group, see \texttt{element\_group}.

6.680  \texttt{group\_materi\_plasti\_cap1} index \(\phi\ \ c\ \ M\ \lambda^*\ \kappa^*\ K^{ref}\ p^{ref}\ m\)

Plastic data for the cap1 plasticity model.

The \textit{index} specifies the element\_group, see \texttt{element\_group}.
6.681 group_materi_plasti_cap2 index c φ α R epsilon \( \epsilon \) \( p_b \) ... 

Plastic data for the cap2 plasticity model. The \( \epsilon \) \( p_b \) ... represents a table with \( \epsilon \) \( p_b \) versus \( p_b \) values; at least two sets of values need to be specified.

The \textit{index} specifies the \textit{element_group}, see \textit{element_group}.

6.682 group_materi_plasti_compression index \( \text{sigy} \)

Yield data for compression plasticity. The \textit{index} specifies the \textit{element_group}, see \textit{element_group}. Condition: \textit{materi_strain_plasti} should be initialized.

6.683 group_materi_plasti_compression_direct index \( \text{sigy} \)

Compression limit. Principal stresses lower than \textit{sigy} are not allowed and will be cut off by Tochnog. This model uses directly a cut-off of stresses, and does not use plastic strains. The \textit{index} specifies the \textit{element_group}, see \textit{element_group}.

You can apply softening with a \textit{dependency_diagram} on \textit{materi_strain_total_compression_kappa}.

6.684 group_materi_plasti_compression_direct_visco index \( \text{tm} \)

Characteristic relaxation time for visco plasticity with \textit{group_materi_plasti_compression_direct}. Choose \( \text{tm} \) small if you want to have fast relaxation of the plasticity to the plastic state.

Thus the \textit{group_materi_plasti_compression_direct} can only use this \textit{group_materi_plasti_compression_direct_visco} for visco-plasticity, and no other visco-plasticity model.

See also \textit{group_materi_plasti_mohr_coul_direct_visco}.

6.685 group_materi_plasti_diprisco index \( \gamma \) \( \hat{\beta}_f \) \( b_p \) \( c_p \) \( t_p \) \( \hat{\theta}_c \) \( \hat{\theta}_e \) \( \xi_c \) \( \xi_e \) \( \beta_f^0 \)

Yield data for di Prisco plasticity. The \textit{index} specifies the \textit{element_group}, see \textit{element_group}. Condition: \textit{materi_strain_plasti} and \textit{materi_plasti_diprisco_history 11} should be initialized.

6.686 group_materi_plasti_diprisco_density index \( \gamma_1 \) \( \hat{\beta}_{lf} \) \( b_p \) \( c_p \) \( t_p \) \( \hat{\theta}_{lc} \) \( \hat{\theta}_{le} \) \( \xi_{lc} \) \( \xi_{le} \) \( \beta_{lf}^0 \) \( \gamma_d \) \( \hat{\beta}_{df} \) \( b_d \) \( c_d \) \( t_d \) \( \hat{\theta}_{dc} \) \( \hat{\theta}_{de} \) \( \xi_{dc} \) \( \xi_{de} \) \( \beta_{df}^0 \) \( e_l \) \( e_d \)

Yield data for di Prisco plasticity with varying density. All data with an \( l \) in the subscript holds for loose soil, whereas all data with an \( d \) in the subscript holds for dense soil. The actually used data will be interpolated between the loose and dense data using the current density.

The \textit{index} specifies the \textit{element_group}, see \textit{element_group}. Condition: \textit{materi_strain_plasti} and \textit{materi_plasti_diprisco_history 12} should be initialized.
Both yield data and flow data (indicated by the word flow) for Drucker-Prager plasticity. Choose \( \phi \) and \( \phi_{\text{flow}} \) in between 0 and \( \frac{\pi}{2} \). The \textit{index} specifies the \textit{element\_group}, see \textit{element\_group}. Condition: \textit{materi\_strain\_plasti} should be initialized.

With this record you can model frictional slip of soil material and alike granular materials on other materials like concrete, steel, etc.

This is done for \textit{group\_materi\_plasti\_mohr\_coul}, \textit{group\_materi\_plasti\_druck\_prag}, \textit{group\_materi\_plasti\_hardsoil}, if specified, by reducing the friction angle \( \phi \) and dilatancy angle \( \phi_{\text{flow}} \) and cohesion \( c \) of the granular material with a factor (2./3.).

This is done for \textit{group\_materi\_plasti\_camclay}, if specified, by reducing \( M \) with a factor (2./3.).

This is done for \textit{group\_materi\_plasti\_tension}, if specified, by reducing \( \text{sigy} \) with a factor (2./3.).

This is done for \textit{group\_materi\_plasti\_hypo\_*}, if specified, by reducing the deviatoric stress increments with a factor (2./3.).

With \textit{group\_0}, \textit{group\_1} etc. you can specify the groups of the concrete material, steel material etc. The reduction of the friction angle and dilatancy angle will only be applied to the granular elements (of \textit{element\_group}) which are a direct neighbor of an element which has one of the groups \textit{group\_0}, \textit{group\_1} etc.

Please realise that this method only works well if the finite elements are not too large.

The \textit{index} specifies the \textit{element\_group} of the granular material, see \textit{element\_group}.

See also \textit{group\_materi\_plasti\_element\_group\_factor}.

With this record you can specify a factor other then the default 2./3. used by the \textit{group\_materi\_plasti\_element\_group} record. You need to specify a factor for each group.

As a special option you can specify one value only, which will then be used for all factors.

The \textit{index} specifies the \textit{element\_group} of the granular material, see \textit{element\_group}.

Yield data for the Generalised Non Associate Cam Clay for Bonded Soils plasticity model. The \textit{index} specifies the \textit{element\_group}, see \textit{element\_group}.
6.691 group_materi_plasti_gurson index sigy q1 q2 q3

Yield data (also used as flow data) for Gurson plasticity. The index specifies the element_group, see element_group.

6.692 group_materi_plasti_hardsoil index φ c ψ R

Plasticity data for Hardening Soil model. The index specifies the element_group, see element_group. This model requires sufficient small timesteps; in case of trouble try smaller timesteps.

6.693 group_materi_plasti_heat_generation factor

This factor specifies how much of the plastic energy loss is transformed into heat (this only makes sense if condif_temperature is initialized). The factor should be between 0 and 1. The index specifies the element_group, see element_group.

6.694 group_materi_plasti_hypo_cohesion index c

Cohesion parameter in hypoplastic law. First, the cohesion in this record is subtracted from the normal stresses in the hypo law evaluation, so that the hypo law is evaluated at a lower pressure level. Second, if the pressure level then still exceeds the $-3. \ast c$, the hypo law only will use its linear contribution. These actions increase numerical stability of the calculation in zones of small stresses (free surfaces).

The index specifies the element_group, see element_group.

6.695 group_materi_plasti_hypo_masin index $\varphi_c \lambda^* \kappa^* N r$

Masin hypoplasticity parameters; see the theory section. The angle $\varphi_c$ should be specified in degrees. The $\lambda^*$ should be bigger than the $\kappa^*$.

6.696 group_materi_plasti_hypo_masin_clay index $\varphi_c \lambda^* \kappa^* N \nu_{pp}$

Masin clay hypoplasticity parameters. The angle $\varphi_c$ should be specified in degrees. The $\lambda^*$ should be bigger than the $\kappa^*$. Typical values are $\varphi_c = 25$, $\lambda^* = 0.1$, $\kappa^* = 0.01$, $N = 1$ and $\nu_{pp} = 0.2$.

6.697 group_materi_plasti_hypo_masin_clay_advanced_parameters

index $\alpha_G \alpha_f a_y o_c$

Optional advanced parameters $\alpha_G \alpha_f a_y o_c$ for masin lay law. If not specified the defaults are: $\alpha_G = 1$, $\alpha_f$ from governing equation in [13], $a_y = 0.3$ and $o_c = 2$. 

255
6.698 group_materi_plasti_hypo_masin_clay_advanced_direction index dir_i

Optional advanced integer parameter for Masin clay law. If not specified the default for the vertical direction dir_i is: 0 in 1D calculation, 1 in 2D calculation and 2 in 3D calculation.

6.699 group_materi_plasti_hypo_masin_clay_index OCR

OCR in masin hypoplastic law; the initial void ratio will be calculated from this. A typical value is OCR = 1.5. You need to set control_materi_plasti_hypo_masin_clay_index OCR apply to -yes.

6.700 group_materi_plasti_hypo_masin_clay_structure index k A s_f

Masin clay hypoplasticity structure parameters; see the theory section. The k should be at least 0. The A should be greater or equal to 0, and lower than 1. The s_f should be greater or equal to 1. Typical values are k = 0.4, A = 0.1 and s_f = 1.5.

6.701 group_materi_plasti_hypo_masin_clay_visco index D_r I_v

Masin clay hypoplasticity visco hypoplasticity parameters; see the theory section. Typical values are \( D_r = 1.e^{-6} \) and \( I_v = 0.1 \).

Remark: this law should not be combined with intergranular strains.

6.702 group_materi_plasti_hypo_masin_index OCR

OCR in masin hypoplastic law; the initial void ratio will be calculated from this. You need to set control_materi_plasti_hypo_masin_index OCR apply to -yes.

6.703 group_materi_plasti_hypo_masin_structure index k A s_f

Masin hypoplasticity structure parameters; see the theory section. The k should be at least 0. The A should be greater or equal to 0, and lower than 1. The s_f should be greater or equal to 1.

6.704 group_materi_plasti_hypo_minimum_void_ratio index minimum_void_ratio

This option sets a minimum for the void ratio as calculated by hypoplasticity laws. If the void ratio would become lower than this minimum_void_ratio value it will be reset to this minimum_void_ratio value.

This option becomes handy when in difficult nonlinear calculations the void ratio may become negative in local zones, preventing the remainder of the calculation to continue. You should check the final results of the calculation if they are what you want.

The index specifies the element_group, see element_group.
Intergranular strain parameters in hypoplastic law. The index specifies the element_group, see element_group.

Intergranular strain parameters for masin clay hypoplastic law. Typical values are $R = 5 \times 10^{-5}$, $A_g = 270$, $n_g = 1$, $m_{rat} = 0.5$, $\beta_r = 0.08$, $\chi = 7$ and $\theta = 7$. The index specifies the element_group, see element_group.

ISA-Intergranular strain parameters in hypoplastic law; see the theory section. The index specifies the element_group, see element_group.

Von-Wolffersdorff parameters in hypoplastic law; see the theory section. Here $\varphi$ is in degrees. The index specifies the element_group, see element_group.

Von-Wolffersdorff-Niemunis extended parameters in hypoplastic law; see the theory section. The index specifies the element_group, see element_group.

Parameters $\varphi$, $\nu$, $D_r$, $I_v$, $e_{c0}$, $p_{e0}$, $\lambda$, $\beta_R$, $\kappa$ for the visco part of hypoplasticity; see the theory section.

The history variables are the same as for group_materi_plasti_hypo_wolffersdorff. You also need to specify control_materi_plasti_hypo_niemunis_visco_ocr_apply.

OCR in visco hypoplastic law. The initial void ratio will be calculated from this; see the theory section.

In case you would like to have an OCR dependent on space coordinate you can use dependency_diagram and dependency_item.
The index specifies the element_group, see element_group.

6.712 group_materi_plasti_hypo_void_ratio_linear index switch

Normally the void ratio change in hypoplasticity is calculated by using the trace of the deformation tensor. That in fact uses a linear approximation of the volume change. This linear approximation is convenient to compare results with analytical theories which are based on such linear approximation, and also to use parameters in hypoplastic laws which are calibrated with such approximation.

Optionally, using -no in this group_materi_plasti_hypo_void_ratio_linear record, the exact volume change of the material is calculated using the determinant of the deformation tensor, and used to calculate the exact new void ratio. However, hypoplasticity laws are usually fitted on the linear approximation of the volume change. So you should not use -no unless you are perfectly certain what you are doing.

6.713 group_materi_plasti_kinematic_hardening index a

This record specifies the size of the rate of the kinematic hardening matrix $\rho_{ij}$. The index specifies the element_group, see element_group.

6.714 group_materi_plasti_laminate0_direction index dir_x dir_y dir_z

Specifies 3 components of the vector normal to the plane of laminate 0. For other laminates you need to use group_materi_plasti_laminate1_direction etc.

The index specifies the element_group, see element_group.

6.715 group_materi_plasti_laminate0_mohr_coul index phi c phiflow

Parameters of laminate 0 for the Mohr-Coulomb plasticity model. Here $\phi$, $c$, $phiflow$, $normal_x$, $normal_y$, $normal_z$ are the friction angle, cohesion and dilatancy angle. For other laminates you need to use group_materi_plasti_laminate1_mohr_coul etc.

The index specifies the element_group, see element_group.

6.716 group_materi_plasti_laminate0_tension index sigma_t

Tension cutoff stress for the tension plasticity model for laminate 0. For other laminates you need to use group_materi_plasti_laminate1_tension etc.

The index specifies the element_group, see element_group.

6.717 group_materi_plasti_mohr_coul index phi c phiflow

Both yield data and flow data (indicated by the word flow) for Mohr-Coulomb plasticity. Choose $\phi$ and $phiflow$ in between 0 and $\frac{\pi}{2}$. The index specifies the element_group, see element_group.
It is advised to use `group_materi_plasti_tension` or preferably with `group_materi_plasti_tension_direct` for tension cutoff of large tension stresses.

6.718 `group_materi_plasti_mohr_coul_direct` index phi c phiflow

Both yield data and flow data (indicated by the word flow) for Mohr-Coulomb plasticity. Choose `phi` and `phiflow` in between 0 and \( \frac{\pi}{2} \). The `index` specifies the element group, see `element_group`.

Principal stress differences higher than allowed by the mohr-coulomb criterium are not allowed and will be cut off by Tochnog. This model uses an alternative programming of the mohr-coulomb law, which tends to be very stable.

You must specify also `group_materi_plasti_tension_direct`.

You can apply softening with a `dependency_diagram` on `materi_strain_total_shear_kappa`.

6.719 `group_materi_plasti_mohr_coul_direct_visco` index `tm`

Characteristic relaxation time for visco plasticity with `group_materi_plasti_mohr_coul_direct`. Choose `tm` small if you want to have fast relaxation of the plasticity to the plastic state.

Thus the `group_materi_plasti_mohr_coul_direct` can only use this `group_materi_plasti_mohr_coul_direct_visco` for visco-plasticity, and no other visco-plasticity model.

For `group_materi_plasti__..._direct_visco` laws visco plasticity works as follows. Suppose the stress at time `t` is `σ_t`. Calculate the elastic response (non-plastic) at time `t+dt` as `σ_{e,t+dt}`. Calculate the plastic non-viscous response at time `t+dt` as `σ_{p,t+dt}`. Calculate the plastic viscous response at time `t+dt` as

\[
σ_{vp,t+dt} = σ_{e,t+dt} + factor \times (σ_{p,t+dt} - σ_{e,t+dt})
\]

where

\[
factor = 1 - \exp(-dt/tm)
\]

Notice that for `dt` is very small `factor = 0` so `σ_{vp,t+dt} = σ_{e,t+dt}`, thus purely elastic response (which is the stress at the previous time plus the elastic stress increment). Notice that for `dt` is very large `factor = 1` so `σ_{vp,t+dt} = σ_{p,t+dt}`, thus purely plastic response (which is the stress completely mapped to the plastic surface). And for `dt` in between the response is in between the elastic response and purely plastic response.

6.720 `group_materi_plasti_mohr_coul_direct_wall` index phi c phiflow

As a special option for `group_materi_plasti_mohr_coul_direct` you can use a record `group_materi_plasti_mohr_coul_direct_wall` index phi c phiflow. These values will be used when an element is attached to a wall. The input file needs to contain `group_materi_plasti_element_group` or `group_materi_plasti_bounda` or `group_materi_plasti_mpc` to specify when an element is at the wall. Thus a typical input file looks like:

```plaintext
... group_materi_plasti_mohr_coul_direct 0 ...
(group_materi_plasti_mohr_coul_direct_wall 0 ...
) (values for elements not attached to wall)
(group_materi_plasti_mohr_coul_direct_wall 0 ...
) (values for elements attached to wall)
```
group_materi_plasti_mpc 0 -yes (elements with mpc’s are attached to wall)

... end_data

See also group_materi_plasti_element_group, group_materi_plasti_bounda and group_materi_plasti_mpc.

6.721 group_materi_plasti_mohr_coul_hardening_softening index phi_0 c_0 phiflow_0 phi_1 c_1 phiflow_1 kappashear_crit

Both yield data and flow data (indicated by the word flow) for Mohr-Coulomb hardening-softening plasticity. See the theoretical part. Choose each of the angles phi_0 phiflow_0 phi_1 phiflow_1 in between 0 and \( \frac{\pi}{2} \). It is advised to use group_materi_plasti_tension or preferably with group_materi_plasti_tension_direct for tension cutoff of large tension stresses. The index specifies the element_group, see element_group.

6.722 group_materi_plasti_mohr_coul_direct_wall index phi c phiflow

As a special option for group_materi_plasti_mohr_coul_direct you can use a record group_materi_plasti_mohr_coul_direct_wall index phi c phiflow. These values will be used when an element is attached to a wall. The input file needs to contain group_materi_plasti_bounda or group_materi_plasti_mpc to specify when an element is at the wall. Thus a typical input file looks like:

... group_materi_plasti_mohr_coul_direct 0 ... (values for elements not attached to wall)
group_materi_plasti_mohr_coul_direct_wall 0 ... (values for elements attached to wall)
group_materi_plasti_mpc 0 -yes (elements with mpc’s are attached to wall)
... end_data

See also group_materi_plasti_bounda and group_materi_plasti_mpc.

6.723 group_materi_plasti_mpc index switch

Same as group_materi_plasti_bounda, but now for mpc ... records however. If you set switch to -yes, the reduction factor will be applied if there is any mpc at the node of an element.

See also group_materi_plasti_mpc_factor.

6.724 group_materi_plasti_mpc_factor index factor

Same as group_materi_plasti_bounda_factor, but now for group_materi_plasti_mpc however.
6.725 group_materi_plasti_pressure_limit index pressure_limit

To prevent plasticity problems near free surfaces, you can require that Tochnog neglects plasticity laws if the pressure exceeds pressure_limit. This option is not available for hypoplasticity laws, since for these laws nonlinear elasticity and plasticity are defined by one law, so the plasticity part cannot be suppressed by itself.

See also group_materi_plasti_pressure_limit_method.

6.726 group_materi_plasti_tension index sigy

Yield data for tension plasticity. The index specifies the element_group, see element_group. Condition: materi_strain_plasti should be initialized.

It is encouraged to use group_materi_plasti_tension_direct instead, which tends to give more stable calculations.

6.727 group_materi_plasti_tension_direct index sigy

Tension limit. Principal stresses higher than sigy are not allowed and will be cut of by Tochnog. This model uses directly a cut-off of stresses, and does not use plastic strains. The index specifies the element_group, see element_group.

You can apply softening with a dependency_diagram on materi_strain_total_tension_kappa. See also group_materi_plasti_tension_direct_automatic.

6.728 group_materi_plasti_tension_direct_automatic index switch

If switch is set to -yes, a plastic tension limit is set in the apex of the group_materi_plasti_mohr_coul_direct with the same index. This actually means a tension limit of $\frac{c}{\tan(\phi)}$.

If you specify group_materi_plasti_mohr_coul_direct with the same index and no group_materi_plasti_tension_direct record, then Tochnog automatically puts switch to -yes in this group_materi_plasti_tension_direct_automatic record.

6.729 group_materi_plasti_tension_direct_visco index tm

Characteristic relaxation time for visco plasticity with group_materi_plasti_tension_direct. Choose tm small if you want to have fast relaxation of the plasticity to the plastic state.

Thus the group_materi_plasti_tension_direct can only use this group_materi_plasti_tension_direct for visco-plasticity, and no other visco-plasticity model.

See also group_materi_plasti_mohr_coul_direct_visco.

6.730 group_materi_plasti_tension_direct_wall index sigy

As a special option for group_materi_plasti_tension_direct you can use a record group_materi_plasti_tension_direct_wall index sigy. This value will be used when an element is attached to a wall. The input file needs to contain group_materi_plasti_element_group,
group_materi_plasti_bounda or group_materi_plasti_mpc to specify when an element is at the wall.

6.731 group_materi_plasti_user index switch

If switch is set to -yes the user supplied routine for plasticity is called.

See also the file user.cpp in the distribution.

The index specifies the element_group, see element_group.

6.732 group_materi_plasti_visco_exponential index γ α

This record specifies visco-plasticity data for the exponential model. It should be used in combination with a plasticity model.

6.733 group_materi_plasti_visco_exponential_limit index limit

This record defines the limit for the exponential viscoplasticity argument alphaf. If the argument alphaf becomes larger than limit then actually limit will be used instead as argument for the exponent.

Default, if group_materi_plasti_visco_exponential_limit is not specified, then 3 will be used as limit.

This record specifies visco-plasticity data for the exponential model. It should be used in combination with a plasticity model.

6.734 group_materi_plasti_visco_exponential_name index name_0 name_1 ...

Same as group_materi_plasti_visco_power_names, now for the exponential law however.

6.735 group_materi_plasti_visco_exponential_values index γ₀ α₀ γ₁ α₁ ...

See group_materi_plasti_visco_exponential_name.

6.736 group_materi_plasti_visco_power index η p

This record specifies visco-plasticity data for the power model. It should be used in combination with a plasticity model.

The index specifies the element_group, see element_group.
This `group_materi_plasti_visco_power_name` together with `group_materi_plasti_visco_power_value` allows you to specify different viscoelastic parameters for each of the plasticity models.

Set each of the names `name_0`, `name_1`, etc. to the plasticity models that you use (e.g. `group_materi_plasti_mohr_coul` etc.) Set the visco parameters for `name_0` in \( \eta_0 \) and \( p_0 \), set the visco parameters for `name_1` in \( \eta_1 \) and \( p_1 \), etc.

In case a plasticity model is used, but is not present in the names `name_0`, `name_1`, etc. then that model will be evaluated elasto-plastic (and thus not elasto-viscoplastic).

The `index` specifies the element_group, see `element_group`.

See `group_materi_plasti_visco_power_name`.

Yield data for Von-Mises plasticity.

The `index` specifies the element_group, see `element_group`. Condition: `materi_strain_plasti` should be initialized.

Data for Von-Mises Nadai hardening. The \( \sigma_{y0} \) of the `group_materi_plasti_vonmises` record is taken as \( \sigma_{y0} \) in the nadai law.

The `index` specifies the element_group, see `element_group`. Condition: `materi_plasti_kappa` should be initialized.

If `switch` is set to `-yes`, then stokes flow is used. If `switch` is set to `-no`, then stokes flow is not used. Default it is set to `-yes`. The `index` specifies the element_group, see `element_group`.

If `switch` is set to `-yes` then the user supplied umat routine is called for the element group `index`. See also the section about user supplied routines at the end of this manual.
6.743 group_materi_umat_parameters index parameter_0 parameter_1 ... 

User supplied parameters for group_materi_umat.

6.744 group_materi_umat_pardiso_decompose index switch

If switch is set to -yes and an umat routine is present, Tochnog will ask the pardiso solver to decompose the system matrix each and every iteration of each and every timestep. If switch is set to -no and an umat routine is present, Tochnog will ask the pardiso solver to decompose the system matrix only once (please realise, however, that because of other input file options the decomposition possibly can be done more than once). Default, if switch is not defined, it is set to -yes.

6.745 group_materi_undrained_capacity index C

Capacity for undrained analysis. See the theory section for details on undrained analyses.

6.746 group_materi_viscosity index ν

Dynamic viscosity for nearly incompressible Newtonian flow. The index specifies the element_group, see element_group.

6.747 group_materi_viscosity_heatgeneration switch

If switch is set to -yes, then viscous dissipation will be used as a heat generation source. See also the theoretical part at the start of this manual. The index specifies the element_group, see element_group.

6.748 group_materi_viscosity_user index switch

If switch is set to -yes, the user supplied routine for the viscosity for Newtonian flow is used. The index specifies the element_group, see element_group.

6.749 group_plasti_apply index switch

If switch is set to -no any plasticity data in the group index will be neglected. Default, if group_plasti_apply is not specified, switch is set to -yes.

6.750 group_porosity index n

Porosity in material. For example needed for group_groundflow_nonsaturated_vangenuchten. The index specifies the element_group, see element_group.
6.751  group_spherical  index switch

If switch is set to -yes, the calculation becomes spherical for the group index. Each specified x coordinate becomes a radius and y becomes the φ direction and the z becomes the θ direction. Specify only positive x coordinates (thus only a radius), and no y and z coordinates.

6.752  group_spring_direction  index dir_x dir_y dir_z

Direction of a spring. If for a -spring2 this record is not specified, the direction is taken to be from the first node of the spring to the second node. The index specifies the element_group, see element_group.

6.753  group_spring_memory  index memory_type

Memory model for spring; either -updated_linear, -total_linear or -updated. The -updated model is a geometrically nonlinear model which takes large spring rotations into account for two-noded springs. The index specifies the element_group, see element_group.

6.754  group_spring_plasti  index F_y

Maximum force in a spring. The index specifies the element_group, see element_group.

6.755  group_spring_stiffness  index k

Stiffness of a spring. It is multiplied with the elongation of the spring to calculate the spring force. The index specifies the element_group, see element_group.

6.756  group_spring_stiffness_nonlinear  index epsilon0 k0 epsilon1 k1 ...

Diagram with spring stiffness dependent on total spring strain (= total spring elongation). Here epsilon0 k0 is the first point in the diagram, with epsilon0 the total spring strain and k0 the spring stiffness. Likewise for the next points in the diagram. Take care that you specify diagram values with a strain range that includes all spring strain that actually occur in the calculation.

The index specifies the element_group, see element_group.

6.757  group_time  index birth death

With this option you can set the time of birth of the elements (in group index) and the time of death of the elements.

Out of the range birth - death the elements of the group will not be used in the calculation (the starting birth limit itself is not included in the range, whereas the ending death limit itself is included).
6.758 group_time_fill index birth_empty birth_filled death

With this option you can set the time of birth of the elements (in group index) and the time of death of the elements.

Out of the range birth_empty - death the elements of the group will not be used in the calculation (the starting birth_empty limit itself is not included in the range, whereas the ending death limit itself is included).

Between birth_empty and birth_filled the elements will be 'slowly filled with material'. This means that the density of the element and the total pressure (pore pressure), in case groundflow is present, will be scaled with a factor 0 at time birth_empty up to a factor 1 at time birth_filled. To prevent numerical problems at low gravity, any plasticity data will be ignored when an element is being filled; after the element is completely filled plasticity will become active (plasticity data will be applied).

6.759 group_truss_area index A

Cross-sectional area for a truss. The index specifies the element_group, see element_group.

6.760 group_truss_density index ρ

Density for a truss. The index specifies the element_group, see element_group.

6.761 group_truss_elasti_elongation_force_diagram index l_0 F_0 l_1 F_1 ...

With this record you can specify a force versus elongation diagram for a truss. Here each l_i is the ratio of the truss elongation divided by the initial truss length. And each F_i is the corresponding force. This group_truss_elasti_elongation_force_diagram cannot be combined with group_truss_elasti_young.

6.762 group_truss_elasti_young index E

Young's modulus for a truss. The truss force F is F = EAΔu, where Δu is the elongation of the truss. The index specifies the element_group, see element_group.

See also group_truss_area.

6.763 group_truss_expansion index alpha

Thermal expansion coefficient for trusses. A temperature increment dT leads to a thermal incremental length of the size alpha * dT * initial length;

6.764 group_truss_initial_force index initial_force

Initial truss force in truss elements.
6.765  group_truss_memory  index memory_type

Memory model for truss; either -updated_linear, -updated or total_linear. The -updated
model is a geometrically nonlinear model which takes large truss rotations into account. The index
specifies the element_group, see element_group.

6.766  group_truss_rope  index switch

The truss will act as a rope if switch is set to -yes. This means that negative forces will not
be allowed (the force remains zero in compression). The index specifies the element_group, see element_group.

6.767  group_truss_plasti  index sigma_c sigma_t

Compressive and tension yield stress for truss. The actual stress cannot become lower than the
sigma_c in compression, and the actual stress cannot become higher than the sigma_t in tension.
The index specifies the element_group, see element_group.

6.768  group_type  index type_name_0 type_name_1 ...

With this record a differential equation is specified for the element group index. Allowed type
names are -condif, -groundflow, -materi, -wave, -spring, -contact_spring, -truss, -beam,
-truss_beam. Also -empty is allowed; it indicates that the element is empty.

For the -truss_beam type you need to set parameters with group_truss_* and group_beam_*
records. For the -truss type you need to set parameters with group_truss_* records. For the
-beam type you need to set parameters with group_beam_* records. For the -condif type
you need to set parameters with group_condif_* records. For the -materi type you need to
set parameters with group_materi_* records. Etc etc.

See also element_group.

6.769  group_volume_factor  index factor

In 1D or 2D you can specify the cross-section and thickness respectively, for elements of the element
group index (see element_group).

See also volume_factor.

6.770  group_wave_speed_of_sound  index c

Speed of sound in wave equation. The index specifies the element_group, see element_group.

6.771  icontrol  icontrol

With this record you can set the control index which already have been performed. Thus if you
set it to 10, all control_* records up to and including those with index 10 will be skipped, and
the control indices starting from 11 will be performed.
6.772 incremental_driver...

All incremental driver options are given in the example input file below:

echo -no
number_of_space_dimensions 2
materi_velocity
materi_velocity_integrated
materi_strain_total
materi_strain_plasti
materi_strain_intergranular
materi_strain_isa_c
materi_strain_isa_eacc
materi_plasti_hypo_history 4
materi_stress
groundflow_pressure
end_initia

( RUNNING:
Run this file with the normal tochnog executable.
Results are written in incremental_driver_result.txt.

EQUILIBRIUM:
At the start equilibrium is assumed (an external force is assumed which makes equilibrium with the internal stress and internal pore water pressure if specified).

UNITS:
The word force below means a force unit (eg kN).
The word stress below means a stress unit (eg kPa).
The incremental driver does not use specific units; all units can be used which are consistent. As an exception the hypoplastic masin law uses specific units (see the tochnog users manual).

EXPERIMENTS:

-INCREMENTAL_DRIVER_OEDOMETRIC
'group_axisymmetric index -yes' is allowed; then x=r and y=theta.
'incremental_driver_volume_constant index -yes' is not allowed
x: zero displacement in x-direction
y: zero displacement and strain in y-direction
z: compression in z-direction
user specified: displacement or force on top plane in z-direction,
  the specified values should be negative for compression

-INCREMENTAL_DRIVER_TRIAX
'group_axisymmetric index -yes' is allowed; then x=r and y=theta.
'incremental_driver_volume_constant index -yes' is allowed;
  then the volume is kept constant (undrained water pressure is not needed)
x: fixed stress in x-direction
y: if axisymmetric then y-strain comes from calculation,
  if not axisymmetric then constant stress in y direction
z: compression in z-direction
user specified: displacement or force on top plane in z-direction,
  the specified values should be negative for compression
**INCREMENTAL_DRIVER_DIRECT_SHEAR**

'group_axisymmetric index -yes' is not allowed.
'incremental_driver_volume_constant index -yes' is allowed;
then the volume is kept constant (undrained water pressure is not needed)
x: zero strain in x-direction
y: zero displacement and strain in y-direction
z: shear of top z-plane in x-direction
user specified: shear displacement or shear force on top plane in x-direction

**INCREMENTAL_DRIVER HYDROSTATIC**

'group_axisymmetric index -yes' is allowed.
'incremental_driver_volume_constant index -yes' is not allowed
equal compression in x- and y- and z-direction
user specified: volumetric displacement

\[ \frac{(x\text{-}plus y\text{-}plus z\text{-displacement})}{3} \]
or volumetric stress
notice: not a force but instead a stress is specified.

**COMBINED EXPERIMENTS:**

Only combined experiments will be analysed, and only output will be printed for these combined experiments; the experiments by themselves will not be analysed.
The start time of each experiment in a combined experiment should be equal to the end time of a previous experiment in that combined experiment.
Only for the first experiment the variables incremental_driver_void_ratio, incremental_driver_stress , incremental_driver_pressure and incremental_driver_intergranular_strain can be specified;
for all subsequent experiments these values will be taken from the end values of the previous experiment.
If no combined experiments are specified, automatically a combined experiment will be generated for each experiment, containing that experiment only.
Notice that prescribed displacements and forces in an experiment are additional to the last values of the previous experiment.
In an experiment all forces that are not prescribed and don’t belong to a prescribed displacement remain equal to those of the previous experiment.

**UNDRAINED:**

You can obtain undrained behavior in two different ways:
1. Use groundwater in the test specimen. The groundwater capacity physically causes the test specimen to behave undrained.
2. Apply 'incremental_driver_volume_constant index -yes' . This causes the deformations of the test specimen to be such that the volume remains constant.
You should choose one of both methods. You should not combine both methods.

**GROUNDFLOW:**

In case of 'group_type index -groundflow' groundwater is present in the test specimen. Then all groundwater initialisation and data needs to be specified (search for groundwater in this file to see what is needed).

**MATERIAL LAWS:**

group_materi_plasti_mohr_coul_direct*
group_materi_plasti_hypo*
group_materi_umat*
See the tochnog users manual.

**MATERIAL MEMORY:**

If you use -updated_linear then small deformation theory is used, and the area change of the surface where forces are applied is not taken into account.
If you use -updated_area then small deformation theory is used, but the area
change of the surface where forces are applied is taken into account.
If you use -updated then large deformation theory is used, and the area change of the surface where forces are applied is taken into account.

AXISYMMETRIC:
Either all experiments are axisymmetric or all experiments are not axisymmetric. Do not mix axisymmetric and non-axisymmetric experiments in a combined experiment.

)  
post_calcul -groundflow_pressure -total_pressure  
group_type 0 -materi  
group_axisymmetric 0 -no  
group_materi_memory 0 -updated_area  
group_materi_plasti_hypo_cohesion 0 1.  
group_materi_plasti_hypo_wolffersdorff 0 30. 5800.e3 0.28 0.84 0.53 1.00 0.13 1.05  
group_materi_plasti_hypo_strain_intergranular 0 1.e-4 5.0 2.0 0.50 6.0 6.0  
group_materi_plasti_hypo_strain_isa 0 20.0 0.017  
group_type 1 -materi -groundflow  
group_axisymmetric 1 -no  
group_materi_memory 1 -updated_area  
group_materi_elasti_young 1 6000.  
group_materi_elasti_poisson 1 0.4  
group_materi_plasti_mohr_coul_direct 1 0.4 1.e1 0.1  
(gamma_0 c_0 gamma_1 c_1 ..., optional)  
group_materi_plasti_mohr_coul_direct_incremental_driver_c 1 0. 10. 0.01 9. 100. 0.  
(gamma_0 phi_0 gamma_1 phi_1 ..., optional)  
group_materi_plasti_mohr_coul_direct_incremental_driver_phi 1 0. 0.4 0.01 0.3 100. 0.1  
(gamma_0 psi_0 gamma_1 psi_1 ..., optional)  
group_materi_plasti_mohr_coul_direct_incremental_driver_psi 1 0. 0.1 0.01 0.05 100. 0.02  
group_groundflow_capacity 1 1.e-4  
( etc )  
(number of timesteps to be used for each experiment, default 1000. In these steps the total time table as specified by)
incremental_driver_experiment_time is performed
incremental_netime 1000

(set to experiment name)
incremental_driver 0 -incremental_driver_direct_shear

(set to element group used)
incremental_driver_element_group 0 0

(set to -yes for force controlled experiment,
set to -no for displacement controlled)
incremental_driver_experiment_force_controlled 0 -no

(time table for additional displacement or additional force at top y-plane
in x-direction relative to start time experiment)
incremental_driver_experiment_time 0 0.0 0.0 1.0 0.02

(initial stress: sigxx sigxy sigxz sigyy sigyz sigzz)
incremental_driver_stress 0 -100 0 0 -100 0 -100

(initial pore pressure for undrained analysis for groundflow)
incremental_driver_pressure 0 -10.

(initial void_ratio for hypo laws)
incremental_driver_void_ratio 0 0.75

(initial intergranular strain for hypo: epixx epixy epixz epiyy epiyz epizz)
incremental_driver_intergranular_strain 0 -0.0000577 0. 0. -0.0000577 0. -0.0000577

(initial isa intergranular back-strain for hypo: episa_cxx episa_cxy episa_cxz episa_cyy episa_cyz episa_czz)
incremental_driver_isa_strain 0 -0.0000404145 0. 0. -0.0000404145 0. -0.0000404145

(initial length in x-direction)
incremental_driver_length_x 0 0.1

(initial length in y-direction, not needed in axisymmetric)
incremental_driver_length_y 0 0.1

(initial length in z-direction)
incremental_driver_length_z 0 0.2

(set to experiment name)
incremental_driver 1 -incremental_driver_direct_shear

(set to element group used)
incremental_driver_element_group 1 0

(set to -yes for force controlled experiment,
set to -no for displacement controlled)
incremental_driver_experiment_force_controlled 1 -no

(time table for additional displacement or additional force at top y-plane
in x-direction relative to start time experiment)
incremental_driver_experiment_time 1 1.0 0.0 2.0 0.02

(initial length in x-direction)
incremental_driver_length_x 1 0.1
(initial length in y-direction, not needed in axisymmetric)
incremental_driver_length_y 1 0.1

(initial length in z-direction)
incremental_driver_length_z 1 0.2

(a combined experiment which only contains experiment 0)
incremental_driver_combined 0 0

(a combined experiment which contains both experiment 0 and 1)
incremental_driver_combined 1 0 1

end_data

6.773 inertia_apply switch_0 switch_1 ...

If switch_0 is set to -yes, the corresponding inertia term is included (material mass, heat capacity, ..). The same for the other switches. A switch should be specified for each of the principal dof’s. See the 'input file - data part - introduction - types of dof’s' section for an explanation about principal dof’s. The sequence of the principal dof’s is in the order as initialised in the initia ... end_initia part.

As a special option you can specify only one switch, and then the specified value will automatically be used for all principal dof’s.

This inertia_apply is applied for all timestep records.

Default, if inertia_apply is not specified, then each of switch_0, switch_1 etc. is set to -no.

See also control_inertia_apply.

6.774 input_abaqus switch

Set switch to -yes for reading the abaqus input file abaqus.inp. Tochnog will use it to generate a tochnog input file tochnog_abaqus.dat. This can typically be done by making an input file like:

```bash
... 
  echo -yes
  number_of_space_dimensions 3
  materi_velocity
  materi_stress
  end_initia
  input_abaqus -yes
  input_abaqus_mesh -no
  input_abaqus_continue -yes
... 
  include tochnog_abaqus.dat
```
You need to initialise the fields like `materi_velocity`, `materi_stress`, etc that you will actually use later in the calculation. Only a limited set of data is transferred from the abaqus input file to the tochnog input file; you need to check if the Tochnog input file is like you want. Abaqus element sets and node sets are evaluated and can be used in the tochnog calculation.

ABAQUS is a registered trademark or trademark of Dassault Systemes. Abaqus input files can be generated by many programs, among which mecway is relatively affordable, see https://mecway.com/.

### 6.775 input_abaqus_continue switch

If `switch` is set to `-yes` then after `tochnog_abaqus.dat` is generated the remainder of the input file read and the calculation continues. If `switch` is set to `-no` then after `tochnog_abaqus.dat` is generated the remainder of the input file will not be read and the calculation aborts. The input_abaqus_continue record should always be present as last record of the input_abaqus_* records.

### 6.776 input_abaqus_group switch

If `switch` is set to `-yes` then also `group_*` is written to `tochnog_abaqus.dat`. If `switch` is set to `-no` then no `group_*` is written to `tochnog_abaqus.dat`. So you can set `switch` to `-no` in case you want to provide the `group_*` yourself, and don’t want it to be taken from the `abaqus.inp`.

Default, if `input_abaqus_group` is not specified, the `switch` is set to `-yes`.

### 6.777 input_abaqus_mesh switch

If `switch` is set to `-no` then only mesh data is written to `tochnog_abaqus.dat`; so timestep information and post processing prints are not written. This record should be placed before the `input_abaqus_continue` record.

Default the `switch` is set to `-no`.

### 6.778 input_abaqus_set set_0 set_1 ...

With this option you can specify for which set numbers the elements should be written. See the generated `tochnog_abaqus.dat` for the set numbers.

### 6.779 input_abaqus_name name_0 name_1 ...

With this option you can specify which abaqus element types should be converted into tochnog elements. For example specify `-tria3` if you want to include `tria3` elements in the Tochnog input file. In case you do not specify `input_abaqus_name` all elements will be converted into tochnog
elements. However, not all abaqus elements are available as tochnog element; if a non-available element is encountered it will be skipped.

6.780 input_feflow_mesh switch

If switch is set to -yes the mesh is read from a FEFLOW file. FEFLOW is a dedicated groundwater analysis program, see https://www.mikepoweredbydhi.com/products/feflow. The Tochnog input file itself should not contain a mesh (elements and nodes). Presently, in a 3D analysis the thickness of each layer should be uniform over the nodes (no varying layer thickness in x- and y-direction). Default elements of the FEFLOW file get the element_group 0 in the Tochnog calculation; however, for each set of elements specified inside the feflow file with K_xx inside MAT_I_FLOW an increasing element group number is generated (1 for the elements specified in the first K_xx, 2 for the elements specified in the second K_xx, etc.).

6.781 input_feflow_fem switch

If the switch in input_feflow_fem is set to -yes the mesh is read from the file feflow.fem. If the switch in input_feflow_fem is set to -no the mesh is read from the file feflow.dac. Both files should be in ASCII text format. Default, if input_feflow_fem is not specified, the switch is set to -no.

See also input_feflow_mesh.

6.782 input_feflow_mesh_hydraulic_head switch

If the switch in input_feflow_hydraulic_head is set to -yes then also the hydraulic head is read from the feflow.dac file. The feflow.dac should contain hydraulic pressure head results for all time points in the Tochnog calculation. The hydraulic pressure head as calculated by FEFLOW is then used as prescribed value in the Tochnog calculation, so that Tochnog only calculates stress (influenced however by the presence of the hydraulic pressure head). This option allows you to calculate hydraulic safety factors using FEFLOW hydraulic pressure heads and Tochnog stresses (see safety_piping and safety_lifting in post_calcul).

See also input_feflow_mesh.

6.783 input_gmsh switch

Set switch to -yes for reading the gmsh mesh file tochnog_in.msh. Only linear and quadratic elements are read.

The gmsh program is a free external pre- and postprocessor. See http://www.geuz.org/gmsh.

Only the data element, element_group and node is read.

6.784 interface_gap_apply switch

If switch is set to -yes then any group_interface_gap will be applied. If switch is set to -no then any group_interface_gap will be ignored.

Default, if interface_gap_apply is not specified, switch is set to -yes.
This `interface_gap_apply` record will be overruled by the `control_interface_gap_apply` record if specified.

6.785 **license_check switch**

If `switch` is set to `-yes` tochnog checks if the license of selected customers is available (which unlocks special features).

Default `switch` is set to `-no`.

6.786 **license_wait switch**

If `switch` is set to `-yes` tochnog waits 10 seconds till a valid license is found on the computer.

Default `switch` is set to `-no`.

6.787 **linear_calculation_apply switch**

If you set the `switch` to `-yes`, Tochnog will skip nonlinearities from the input file. This option is convenient for testing and problem search. Simple set `linear_calculation -yes` so that the calculation should run without any trouble, and use a `control_print` for `-post_node_rhside_ratio`. The printed `-post_node_rhside_ratio` should be very small, typically 1.e-10 or lower, since the calculation is linear now. If that is not the case, there may be a problem with the boundary conditions or some other problem.

A typical sequence for testing very large calculations may be following: first run with `solver -none` and check the mesh at all times; second run with `linear_calculation_apply -yes` to check if good linear solutions fields are obtained (check the linear results carefully); finally run your actual calculation without any special options.

The following specific actions are taken:

- Any `control_plasti_apply` is deleted, and `plasti_apply` is set to `-no`.
- `mesh` is set to `-fixed_in_space`.
- For all `group_*_memory` the memory type is set to `-total_linear` if `materi_displacement` is initialised, and it is set to `-updated_linear` otherwise.
- Any `dependency_item, dependency_diagram` containing `group_*` data depending on one of the dof’s of `dof_label` is deleted.
- Any `group_materi_elasti_hardsoil` is deleted and substituted by a `group_materi_elasti_young` with $E_{50\,ref}$ as Young’s modulus.
- Any `group_materi_elasti_polynomial` is deleted and substituted by a `group_materi_elasti_young` with $E_0$ as Young’s modulus.
- Any `group_materi_elasti_young_power` is substituted by a linear `group_materi_elasti_young`.
- Any `group_materi_plasti_hypo_wolffersdorff` is deleted and substituted by a `group_materi_elasti_young` with $h_s$ as Young’s modulus, and a `group_materi_elasti_poisson` with value 0.2.
- Any `group_spring_stiffness_nonlinear` is deleted and substituted by a `group_spring_stiffness` with the stiffness value at strain 0.
• Any `group_groundflow_nonsaturated` is deleted.
• Any `group_interface_gap` is deleted.
• Any `group_interface_materi_hardening` is deleted.
• Any `group_interface_materi_elasti_stiffness_tangential_diagram` is deleted.
• Any `group_materi_damage` is deleted.
• Any `group_materi_failure` is deleted.
• Any `group_truss_rope` is deleted.
• Any `contact_*` is deleted.
• Any `groundflow_seepage_*` is deleted.

6.788 materi_damage_apply switch

If `switch` is set to `-no`, any damage data in the input file will be ignored. This is done for all timesteps.

This option is convenient for testing your input file just linear, without the need to outcomment each and every part with damage data. See also control_materi_damage_apply.

6.789 materi_dynamic factor

This option is specifically meant for dynamic calculations. When the solution is known at time \( t \) and is being solved in the new timestep for time \( d + dt \) the stresses at time \( t \) are used with \( 1 - factor \) and the stress at time \( t + dt \) are used with \( factor \). Thus using a factor less than 1 makes the scheme less implicit, and thus less numerical damping will occur.

For quasi static calculations this option should not be used since it results in a loss of stability for strong nonlinearities. Also in case of dynamic calculations with heavy nonlinearity it is better to use the default timestepping method with sufficient small steps. You can try if by using this option the calculation converges; if the calculation converges it is better to use this option; if the calculation diverges by using this option it should be removed.

Using this `materi_dynamic` option the value of `post_node_rhside_ratio` is not usable anymore. So this option is typically used in combination with fixed timestepping. In fact, you should perform the dynamic calculation with typical timestep, and afterwards perform the calculation again with a smaller timestep to see if results change significantly.

Default, if this option is not specified, \( factor \) is set to 1 (so fully implicit).

6.790 materi_elasti_young_power_apply switch

If `switch` is set to `-no`, any nonlinearity in young dependent on a power law will be ignored; simply the constant young as encountered in the `group_materi_elasti_young_power` records will be applied at all times.
6.791 materi_failure_apply switch

If switch is set to -no, any failure data in the input file will be ignored. This is done for all timesteps.

This option is convenient for testing your input file just linear, without the need to outcomment each and every part with failure data. See also control_materi_failure_apply.

6.792 materi_plasti_hypo_substepping index switch

If switch is set to -yes substepping will be applied in hypoplasticity routines. If switch is set to -no substepping will not be applied in hypoplasticity routines.

If the record control_materi_plasti_hypo_substepping is specified that record will be used. If none record is not specified switch is set to -yes.

6.793 materi_plasti_tension_apply switch

If switch is set to -no, any tension-plasticity data in the input file will be ignored. This is done for all timesteps.

See also control_materi_plasti_tension_apply.

6.794 materi_plasti_visco_apply switch

If switch is set to -no, any visco-plasticity data in the input file will be ignored. This is done for all timesteps.

See also control_materi_plasti_visco_apply.

6.795 mesh specifier_x specifier_y specifier_z

If specifier_x is set to -fixed_in_space, the nodal points of the mesh remain fixed in space in x-direction. If a specifier_x is set to -follow_material, the nodal points of the mesh will follow material displacements in x-direction. The same holds for the other directions. In 1D, you only need to give specifier_x, etc.

Default each specifier is set to -fixed_in_space.

This record mesh only is used if materi_velocity is initialised. If materi_displacement is initialized each specifier is automatically set to -fixed_in_space.

6.796 mesh_activate_gravity_element index element_range

See mesh_activate_gravity_time.
6.797  mesh_activate_gravity_element_group  index element_group_0 element_group_1 . . .

See mesh_activate_gravity_time.

6.798  mesh_activate_gravity_geometry  index geometry_item_name geometry_item_index

See mesh_activate_gravity_time.

6.799  mesh_activate_gravity_method  index method

Set to -method1 or -method2. Default Tochnog will use -method2.

See mesh_activate_gravity_time.

6.800  mesh_activate_gravity_stiffness_factor  index factor

See mesh_activate_gravity_time.

6.801  mesh_activate_gravity_time  index time_start time_end

With this record you can slowly activate gravity for elements between time_start and time_end.

You can specify an element range with mesh_activate_gravity_element. The elements you need to specify as elements range. Possible formats for the elements are a number (eg. 5), a number range (eg. -ra 5 4 8 -ra), or all elements (-all).

Or otherwise, you can specify element group numbers with mesh_activate_gravity_element_group.

Or otherwise, you can specify a geometry with mesh_activate_gravity_geometry so that elements completely in the geometry will be used.

Tochnog will activate the elements from the bottom to the top. For each specific element the start time of activation is interpolated from the global time_start and time_end and the lowest coordinate of the element. Likewise, for the element end time of activation the highest coordinate is used. Typically take care that the timestep is so small that each timestep only about 10 percent of an element gets filled.

This option comes handy to slowly build dams or so, starting at the bottom and building upwards to the top.

If mesh_activate_gravity_method is set to -method1, before the element start time of activation, the element is not active in the calculation. After the element end time of activation, the element is fully active in the calculation. Between these times the element is active, but the gravity is 0 at the element start time of activation, the gravity gets its full value at the element end time of activation, and the gravity is interpolated in between. With this -method1 the displacements for activated nodes are 0 at the moment of activation, and grow later in time. Thus the displacements in the activation area are relative to the moment of material activation, and not relative to the moment of start of the calculation.
If `mesh_activate_gravity_method` is set to `-method2`, before the element start time of activation, the element is active in the calculation, but has no gravity force yet. After the element end time of activation, the element is fully active in the calculation with full gravity force. Between these times the element is active, but the gravity is 0 at the element start time of activation, the gravity gets its full value at the element end time of activation, and the gravity is interpolated in between. With this `-method2` the displacements for activated nodes are not 0 at the moment of activation, but already have values resulting from activation of material below. For elements which are not activated yet, Tochnog will reduce the stiffness so that it will not really influence displacements inside the elements which are already being activated; the stiffness reduction factor can be specified by `mesh_activate_gravity_stiffness_factor`, and is 1.e-6 by default. For elements which are not activated yet, Tochnog will not print the elements to the gid postprocessing files; however you can demand that these elements will also be printed by specifying `-yes` in `print_gid_mesh_activate_gravity` or `control_print_gid_mesh_activate_gravity` (default `-no`).

You can set with `mesh_activate_gravity_time_initial` when elements become active in a calculation; before the specified `time_of_birth` an element will not take part of the calculation.

You can only use this option not in combination with `materi_displacement`. See also `control_mesh_activate_gravity_apply`.

**6.802 mesh_activate_gravity_time_initial** `index time_of_birth`

See `mesh_activate_gravity_time`.

**6.803 mesh_activate_gravity_time_strain_settlement** `index switch`

If `switch` is set to `-yes` then strain settlement should be used for the `mesh_activate_gravity_time` record with the same index.

**6.804 mesh_boundary** `switch`

If `switch` is set to `-yes`, Tochnog determines the boundary of the mesh and sets `element_boundary` and `node_boundary` records. If `switch` is set to `-no`, Tochnog does not determine the boundary of the mesh. Default, if `mesh_boundary` is not specified, the `switch` is `-no`.

**6.805 mesh_correct** `switch`

If `switch` is set to `-yes`, Tochnog checks that the connectivity list for quadrilateral and hexahedral interfaces is correct. If the connectivity list would not be correct (that is, according to the required sequence in Tochnog for such elements), the connectivity list will be corrected.

Default `switch` is set to `-no`.

**6.806 mesh_delete_geometry_moving** `index geometry_moving_index`

With `geometry_moving_index` you tell Tochnog to excavate the mesh with the geometrical entity as specified by the `geometry_moving*` records with index `geometry_moving_index`. 
6.807  mesh_element_group_apply  index  group_0  group_1  

If you specify this record, only the element groups specified will be evaluated in all timesteps. Default, if mesh_element_group_apply is not specified, all elements groups will be used.

If control_mesh_element_group_apply is specified it overrules this mesh_element_group_apply record.

6.808  mesh_gid_assign_conditions_line  index  line_0  line_1  

Assign nodes on gid lines to tochnog geometry_list index. You can use geometry lists to impose geometrical conditions, see geometry_list. The index needs to be 11, 12, ...., 20 (start with index 11, then index 12, up to maximal 20).

A simple example is:

mesh_gid_assign_conditions_line 11 3 4 (assign gid line 3 and 4 to geometry_list 11)
bounda_dof 10 -geometry_list 11 -velx -vely (set velocities to 0 on geometry_list 11)

See also control_mesh_gid_batch.

6.809  mesh_gid_assign_conditions_point  index  point_0  point_1  

Assign nodes on gid points to tochnog geometry_list index. You can use geometry lists to impose geometrical conditions, see geometry_list. The index needs to be 1, 2, ...., 10 (start with index 1, then index 2, up to maximal 10).

See also control_mesh_gid_batch.

6.810  mesh_gid_assign_conditions_surface  index  surface_0  surface_1  

Assign nodes on gid surfaces to tochnog geometry_list index. You can use geometry lists to impose geometrical conditions, see geometry_list. The index needs to be 21, 22, ...., 30 (start with index 21, then index 22, up to maximal 30).

See also control_mesh_gid_batch.

6.811  mesh_gid_arc_coord  index  x_0 y_0 x_1 y_1 x_2 y_2  

Arc coordinates of first, second and third point. See also control_mesh_gid_batch.

6.812  mesh_gid_circle_coord  index  x y  

Circle middle coordinates. See also control_mesh_gid_batch.
6.813 mesh_gid_circle_element_group index group

Group number. See also control_mesh_gid_batch.

6.814 mesh_gid_circle_hollow index switch

SWitch for hollow circle.

6.815 mesh_gid_circle_radius index radius

Circle radius. See also control_mesh_gid_batch.

6.816 mesh_gid_cylinder_coord index x y z

Circle base middle coordinates. See also control_mesh_gid_batch.

6.817 mesh_gid_cylinder_element_group index group

Group number. See also control_mesh_gid_batch.

6.818 mesh gid cylinder_height index height

Cylinder height. See also control_mesh_gid_batch.

6.819 mesh gid cylinder_hollow index switch

Set switch to -yes if you want the cylinder to act as a hollow zone, typically inside another zone (for example a volume zone). So this option comes handy when you want to study, for example, cylindrical excavation zones in a soil.

6.820 mesh_gid_cylinder_normal index normal_x normal_y normal_z

Cylinder normal. See also control_mesh_gid_batch.

6.821 mesh_gid_cylinder_radius index radius

Cylinder radius. See also control_mesh_gid_batch.

6.822 mesh_gid_line_structured_concentrate index weight_start weight_end

Concetrate elements for structured mesh near start or end of line. The weight_start and weight_end can be specified between -1 and +1. These are the mesh concentration factors for the line start
and end respectively. With 0 there is no concentration. With -1 the concentration is low. With +1 the concentration is high.

Default, if mesh_gid_line_structured_concentrate is not specified, both weight factors are 0. See also control_mesh_gid_batch.

6.823 mesh_gid_line_element_group index group
Group number. See also control_mesh_gid_batch.

6.824 mesh_gid_line_point index point_0 point_1
Line points. See also control_mesh_gid_batch.

6.825 mesh_gid_line_size index size
Size of elements at line. See also control_mesh_gid_batch.

6.826 mesh_gid_line_structured_nel index nel
Number of elements for structured line. See also control_mesh_gid_batch.

6.827 mesh_gid_line_structured_size index size
Size of elements at structured line. See also control_mesh_gid_batch.

6.828 mesh_gid_point_coord index x y z
Point coordinates. In 2D you only need to specify x and y. See also control_mesh_gid_batch.

6.829 mesh_gid_rectangle_coord index x_0 x_1 y_0 y_1
Rectangle coordinates of first and second point. See also control_mesh_gid_batch.

6.830 mesh_gid_rectangle_element_group index group
Group number. See also control_mesh_gid_batch.

6.831 mesh_gid_rectangle_hollow index switch
Set switch to -yes if you want the rectangle to act as a hollow zone, typically inside another zone (for example a surface zone). So this option comes handy when you want to study, for example, rectangular excavation zones in a soil.
6.832 **mesh_gid_size**  \textit{element_size}  

Size of elements. See also \texttt{control_mesh_gid_batch}.

6.833 **mesh_gid_sphere_coord**  \textit{index x y z}  

Sphere middle coordinates. See also \texttt{control_mesh_gid_batch}.

6.834 **mesh_gid_sphere_element_group**  \textit{index group}  

Group number. See also \texttt{control_mesh_gid_batch}.

6.835 **mesh_gid_sphere_hollow**  \textit{index switch}  

Set \textit{switch} to \texttt{-yes} if you want the sphere to act as a hollow zone, typically inside another zone (for example a volume zone). So this option comes handy when you want to study, for example, a hollow space in a soil.

6.836 **mesh_gid_sphere_radius**  \textit{index radius}  

Sphere radius. See also \texttt{control_mesh_gid_batch}.

6.837 **mesh_gid_surface_element**  \textit{index element_type}  

Element type for surface. You can use one of \texttt{-tria3}, \texttt{-tria6}, \texttt{-quad4}, \texttt{-quad8}, \texttt{-quad9}, \texttt{-tet4}, \texttt{-tet10}, \texttt{-hex8}, \texttt{-hex20} and \texttt{-hex27}. If this record is not specified linear elements are generated as default, triangles in 2D and tets in 3d. See also \texttt{control_mesh_gid_batch}.

6.838 **mesh_gid_surface_element_group**  \textit{index group}  

Group number. See also \texttt{control_mesh_gid_batch}.

6.839 **mesh_gid_surface_line**  \textit{index line_0 line_1 ...}  

Surface lines. This option will generate 1 new surface. See also \texttt{control_mesh_gid_batch}.

6.840 **mesh_gid_surface_structured_nel**  \textit{index nel_0 nel_1 ...}  

Number of elements on lines of structured surface. Specify a number for each of the lines. Specify a positive number if you want to set the number of elements for the line. Specify a 0 for each of the opposing lines. See also \texttt{control_mesh_gid_batch}.
6.841 mesh_gid_surface_structured_size index size

Size of elements on structured surface. See also control_mesh_gid_batch.

6.842 mesh_gid_volume_element_group index group

Group number. See also control_mesh_gid_batch.

6.843 mesh_gid_volume_surface index surface_0 surface_1 ...

Volume surfaces. This option will generate 1 new volume. See also control_mesh_gid_batch.

6.844 mesh_interface_triangle_coordinates index coord_x_0 coord_y_0 coord_z_0 coord_x_1 coord_y_1 coord_z_1 coord_x_2 coord_y_2 coord_z_2 ...

With this option you can generate interface elements in a 3d mesh with tet4 elements. You specify the triangulated plane of the interface as sets of triangles in 3d space. For each triangle you specify for the three corner points the coordinates. For example coord_x_0 coord_y_0 coord_z_0 are the coordinates of the first corner point, coord_x_1 coord_y_1 coord_z_1 are the coordinates of the second corner point and coord_x_2 coord_y_2 coord_z_2 are the coordinates of the third corner point. The combination of all triangles specifies the plane which will be intersected with the 3d tet4 mesh to generate the interface elements.

With mesh_interface_triangle_element_group you specify the group which will be attributed to the interface elements. With control_mesh_interface_triangle you specify the control index for which the generation should be done.

A typical input file looks like:

```
... group_type 1 -materi
  group_interface 1 -yes
  group_interface_materi_memory 1 -total_linear
  group_interface_materi_elasti_stiffness 1 1.e11 0.5e11 0.5e11
  ...
  mesh_interface_triangle_coordinates 0. 0. 0.6 100. 0. 0.6 0. 100. 0.6
  mesh_interface_triangle_element_group 1
  ...
  control_mesh_interface_triangle 10 -yes
  ...
```

6.845 mesh_interface_triangle_element_group index element_group

See mesh_interface_triangle_coordinates.
6.846 message switch

If switch is set to -yes the introduction message (executable name, date, conditions) will be printed. If switch is set to -no it will not be printed. Default switch is -yes.

You can also set this message as environment symbol to -no if you want to suppress messages for all calculations.

6.847 mpc_apply switch

If switch is set to -yes then mpc conditions will be used. If switch is set to -no then mpc conditions will not be used. Default, if mpc_apply is not specified, switch is set to -yes.

See also control_mpc_apply.

6.848 mpc_element_group index element_group_0 element_group_1

Each node of element of group element_group_0 that is also located in an element of group element_group_1 will be tied to that group by means of multi point constraints. The multi point constraints will be consistent with the shape functions at the specific isoparametric coordinates of the location of that node in the element of group element_group_1. For element_group_1 you can only use isoparametric elements. As a special option you can use -all for element_group_1, so that all other element groups are selected.

See also mpc_element_group_always, mpc_element_group_closest and control_mpc_element_group.

6.849 mpc_element_group_always index switch

If switch is set to -yes the mpc’s will be generated always. If switch is set to -no the mpc’s will only be generated if the considered node is not a member of the node list of the element of group element_group_1 (this ensures that mpc’s will only be generated if the node is completely loose from the other element). You can use the switch is -no option if you are not sure if element_group_0 is connected to, or not connected to, element_group_1; with -no you will not get mpc’s if the groups are connected; see mpc_* in the dbs file to check if mpc’s are generated. So if you are not sure if surfaces are connected in gid, a typical strategy would be:

- run tochnog with mpc_element_group ... and mpc_element_group_always -no
- if you get mpc_* records in the dbs, run again with mpc_element_group ... and mpc_element_group_always -yes
- if you do not get mpc_* records remove mpc_element_group and mpc_element_group_always

Default, if mpc_element_group_always is not specified, switch is set to -yes.

6.850 mpc_element_group_closest index switch

If switch is set to -yes the mpc’s will also be generated if no element of element_group_1 is found; then the closest element of element_group_1 will be used.
6.851 mpc_element_group_coord_geometry index switch

If switch is set to -yes the mpc's will only be generated if the coordinate of the node of element_group_0 is above the lowest node of element_group_1.

A typical example is when you mpc the nodes of soil with element_group_0 to pile elements with element_group_1. Then only soil nodes above the toe of the pile should be mpc'd. A piece of input file will look like:

```plaintext
... start_define pile_group 1 end_define
start_define soil_interface_group 2 end_define ...
start_define pile_edge geometry_line 10 end_define pile_edge ......
...
mpc_element_group 10 soil_interface_group pile_group (mpc soil nodes to pile)
mpc_element_group_geometry 10 -pile_edge (for soil nodes located on the pile edge)
mpc_element_group_closest 10 -yes (the soil node should be mpc'd to the closest pile element)
mpc_element_group_coord_geometry 10 -yes (do this only for soil nodes above the toe of the pile)
mpc_element_group_dof 10 -velx -vely (in 2D mpc the velocities, and thus displacements) ...
```

6.852 mpc_element_group_dof index dof_0 dof_1 ...

The dof_0 dof_1 ... in mpc_element_group_dof specify the dof's that should be set equal, e.g. -velx, -vely etc. Default, if mpc_element_group_dof is not specified, all principal dofs will be set equal.

6.853 mpc_element_group_eps_iso index eps

With eps you can specify the tolerance on the isoparametric coordinates for the element of element_group_1 below which a node of element_group_0 is considered to be located in element_group_1. Default, if mpc_element_group_eps_iso is not specified, eps is set to 1.e-4.

6.854 mpc_element_group_geometry index geometry_entity_item geometry_entity_index

Select a geometry for nodes of element_group_0.
**6.855 mpc_element_group_keep index switch**

If `switch` is set to `-yes` the generated mpc’s will kept in the remainder of the calculation. As a typical application you can use this option to save computing time if you know that the mpc’s need not to be changed during time steps. Or as another example you can use this option if you want that the mpc remains active even if the element for which the mpc is generated gets outside the specified geometry of `mpc_element_group_geometry`.

**6.856 mpc_geometry index geometry_entity_item_0 geometry_entity_index_0 geometry_entity_item_1 geometry_entity_index_1**

See also `mpc_geometry_method`.

If `method` in `mpc_geometry_method` is set to `-method0` the following mpc’s will be generated. This record automatically generates `mpc_node_number` and `mpc_node_factor` records such that dof’s in the second geometry `geometry_entity_item_1 geometry_entity_index_1` become equal to the dofs in the first geometry `geometry_entity_item_0 geometry_entity_index_0`. The `switch_x switch_y switch_z` in `mpc_geometry_switch` specify the coordinates that should be checked to judge if a node in the second geometry is considered to have the same position as a node in the first geometry, and thus should get the same dof’s. Only the coordinate for which the corresponding switch is set to `-yes` will be checked. For example in 3D if `-yes -no -no` are used then a node in the second geometry gets the same dofs of a node in the first geometry if it has (almost) equal x-coordinate; the y and z-coordinate are irrelevant. In 2D only `switch_x switch_y` need to be specified. With `mpc_geometry_tolerance` you can set the tolerance beneath which nodes of the first geometry and second geometry are assumed to have the same coordinate. If `mpc_geometry_tolerance` is not specified then a tolerance of 1.e-4 is used.

If `method` in `mpc_geometry_method` is set to `-method1` the following mpc’s will be generated. You should only specify the first geometry. The dofs of the nodes in this first geometry become equal. The first node of this first geometry becomes the master, all other nodes in this first geometry become slave. If you want to know which node is the first node in this first geometry, use a `control_print ... -node` with a `print_filter` for the first geometry.

If `method` in `mpc_geometry_method` is set to `-method2` the following mpc’s will be generated. You should only specify the first geometry. Unknowns of the nodes with equal coordinate in this first geometry become equal.

**6.857 mpc_geometry_method index method**

See `mpc_geometry`. If this `mpc_geometry_method` is not specified then `method` will be set to `-method0`.

**6.858 mpc_geometry_switch index switch_x switch_y switch_z**

See `mpc_geometry`.

**6.859 mpc_geometry_tolerance index tolerance**

See `mpc_geometry`.
6.860 mpc_geometry_dof index dof_0 dof_1 ...

The $dof_0$ $dof_1$ ... in mpc_geometry_dof specify the dof's that should be set equal, e.g. -velx, -vely etc.

6.861 mpc_linear_quadratic switch

If switch is set to -yes this option is activated.

If you have a mesh with both linear elements and quadratic elements, the mesh is not compatible at the places where the linear elements and quadratic elements meet at a common interface. There some of the quadratic element nodes are not attached to the linear elements, and so non-compatible solution fields occur.

This mpc_linear_quadratic option allows you to automatically prevent the non-compatible solution fields. Tochnog imposes a multi point constraint on all non-compatible solution fields between the linear and quadratic elements, so that the extra nodes of the quadratic elements are forced to follow the solution field of the linear elements, and so compatibility is ensured again.

This option typically can be used to model structural parts like beams, sheet piles, tunnel shells etc with quadratic elements, and the surrounding soil with linear element. Use one quadratic element in the structural part thickness direction, and extra one quadratic soil element attached to the structural element. For the remaining soil elements use linear elements. In this way, the stiff structural elements can deform flexible enough, and you save computer time by modeling most of the soils with linear elements.

6.862 mpc_node_factor index factor_10 factor_11 ...factor_20 factor_21 ...

See mpc_node_number.

6.863 mpc_node_number index node_0 dof_0 node_1 dof_10 dof_11 ...node_2
dof_20 dof_21 ...

This Multi Point Constraint record mpc_node_number allows you to set constraints between dof’s at different nodes. The $dof_0$ specifies the dof at node number $node_0$ which will be constrained. It will be constrained to dof’s $dof_10$, $dof_11$, ... of $node_1$ and $dof_20$, $dof_21$, ... of $node_2$, etc. Only principal dof’s can be specified. Principal dof’s are material velocities, groundwater pressure, temperature in the convection diffusion equation, etc.; see the start of the data section for a definition of principal dof’s. With mpc_node_factor you can set multiplication factors for the constraints. If you don’t specify mpc_node_factor a 1 is used for all factors.

Example:

```
mpc_node_number 10 1 -velx 2 -velx 3 -vely
mpc_node_factor 10 7. 9.
```
In this example the velx_1 = 7. * velx_2 + 9. * vely_3 where velx_1 is the x-velocity at node 1 etc. Node number node_0 is this slave node which depends on nodes node_1 etc. which are the master nodes.

Boundary conditions with bounda_dof and bounda_time cannot be specified for slave nodes. See also mpc_geometry for easy generation of multi point constraints.

6.864 node index coord_0 coord_1 coord_2

Coordinates of node index. In 1D, only coord_0 should be specified, etc.

You are not allowed to put free nodes (not attached to any element) in your model. These free nodes will be removed automatically.

6.865 node_boundary index switch

The switch will be set to -yes if the node with index index is located on the boundary of the mesh.

This record will only become available if mesh_boundary is set to -yes. This record is meant for printing only, it should not be set by the user.

6.866 node_bounded index indicator_dof_0 indicator_dof_1 ...

This record is for printing only, it is not an input record. This record indicates if dof’s in the node are bounded via a bounda_dof record; then the corresponding indicator is set to 1, else it remains 0.

6.867 node_bounded_index index bounda_dof_index_0 bounda_dof_index_1 ...

This record is for printing only, it is not an input record. This record list the index of the bounda_dof record by which the dof’s are bounded. This index is only filled if the dof’s really bounded, so if the corresponding value in the node_bounded record is set to 1.

6.868 node_convection_apply index switch

If switch is set to -no convection contributions in node index are de-activated in case they are activated for the whole mesh by convection_apply or control_convection_apply.

6.869 node_damping index damping_x damping_y damping_z

This record adds a discrete damper to node index in x, y and z direction respectively. In 1D only damping_x needs to be specified, etc. The damper will lead to a nodal force of the size damping_x * v_x where v_x is the velocity in x direction. The same holds for the y and z direction.
6.870 node_deformed_mesh index coord_0 coord_1 coord_2

After the calculation, this record will contain deformed coordinates of node index.

If materi_displacement is initialised these are the initial start coordinates node_start_refined plus the displacements.

If materi_velocity_integrated is initialised and the node follows the material with -follow_material these are the current coordinates of the node node.

If materi_velocity_integrated is initialised and the node is fixed in space with -fixed_in_space these are the initial start coordinates node_start_refined plus the integrated velocities.

Else these are equal to the current node node.

In 1D, only coord_0 is filled, etc..

6.871 node_dof index dof_0 dof_1 ...

dof_0 dof_1 … are the degrees of freedom (dof’s) at the node with number index. The total number and type of the dof’s depends on the initialization part. Each node has the same dof’s.

Unknowns like pressure, temperature, etc. are primary dof’s. The other dof’s, space derivatives and the time derivative, are not primary dof’s. In the example below, -temp is 1., -xtemp is 0.2 and -ttemp is 0.1 in node 6

... number_of_space_dimensions 1
derivatives
condif_temperature
end_initia
...
node_dof 6 1.0 0.2 0.1
...

Default all values in the node_dof records are set zero at the start of the calculation.

These node_dof records contain principal dof’s for all elements (displacements, temperatures, etc). Other dof’s like strains, stresses etc. are only filled for the normal isoparametric elements; thus, for example, strain and stress results for interfaces elements are not placed in the node_dof records.

See also: dof_label and post_point.

6.872 node_dof_calcul index ...

See post_calcul.
6.873 node_dof_start_refined index dof_0 dof_1 ...

This record will be filled with dof_0 dof_1 ..., which are the degrees of freedom (dof's) as specified at the start of the calculation, at the node with number index.

If the mesh has been refined, these start values hold for the refined mesh.

See also node_dof and node_start_refined.

6.874 node_dynamic_pressure index value

With this record you can specify for node index the dynamic pressure. Thus, the dynamic pressure as normally calculated will be overruled with this value.

6.875 node_force index force_x force_y force_z

With this record you can input a discrete nodal force at node index. In 1D you only should specify the force in x-direction. In 2D you only should specify the force in x- and y-direction.

6.876 node_geometry_present index geometry_item_name_0 geometry_item_index_0 geometry_item_name_1 geometry_item_index_1 ...

This record lists for node index the geometries in which it is present. So it is a print record only, for checking if the geometries include exactly the nodes that you want. You can switch on or off filling of these records by setting print_node_geometry_present to -yes or -no.

6.877 node_inertia index inertia_dof_0 inertia_dof_1 ...

This record will be filled with calculated inertia terms degrees of freedom (dof's) as specified at the start of the calculation, at the node with number index. For material velocity that is the mass inertia term in the node.

6.878 node_mass index mass_x mass_y mass_z

This record adds a discrete mass to node index in x, y and z direction. In 1D only the x-mass needs to be specified, etc. The mass will lead to a nodal force of the size mass_x \dot{v}_x where \dot{v}_x is the acceleration, and to a gravity force if force_gravity is specified. The same holds for the y and z direction.

6.879 node_mesh index ...

Same as mesh, but now specified per node however. The index specifies the node number. If this node_mesh record is specified for a node, it overrules the mesh record.
6.880  **node_rhside index rhside_0 rhside_1 ...**

This record will contain after the calculations the unbalance forces.

For the temperature equation, this will give the heat flow normal to the outer surface (the heat flux to the environment) at prescribed temperatures. For velocity dof’s, this will give the force vector at prescribed displacements. For the pressure in the ground flow equation, this will give the ground flow to the environment at prescribed pressures. For non-prescribed dofs the record will contain the numerical unbalance, and should be close to zero.

The `index` is the node number.

6.881  **node_slide index slide_number**

With `node_slide` you can specify if a specific node `index` if it belongs to a sliding geometry with `index slide_number`. For the sliding geometry `slide_geometry` is not needed anymore because the `node_slide` already specifies which nodes belong to the sliding geometry.

6.882  **node_static_pressure index value**

With this record you can specify for node `index` the static pressure. Thus, the static pressure as normally calculated will be overruled with this `value`.

6.883  **node_start_refined index coord_0 coord_1 coord_2**

After the calculation, this record will contain coordinates of node `index` as specified at the start of the calculation. If the mesh has been refined this record will contain the start coordinates for the refined mesh. In 1D, only `coord_0` is filled, etc..

6.884  **node_stiffness index stiffness_x stiffness_y stiffness_z**

This record adds a discrete stiffness to node `index` in `x`, `y` and `z` direction respectively. In 1D only `stiffness_x` needs to be specified, etc. The stiffness will lead to a nodal force of the size `stiffness_x * u_x` where `u_x` is the displacement in `x` direction. The same holds for the `y` and `z` direction. Condition: also `materi_velocity_integrated` or `materi_displacement` should be initialized.

6.885  **node_support_edge_normal_plasti_tension_status index status**

This record will contain after a calculation the status of a node for the `support_edge_normal_plasti_tension` or `support_edge_normal_plasti_tension_double` option. If the node is opened due to tension plasticity the status is set to `-opened`. If the node is closed the status is set to `-closed`.

6.886  **node_total_pressure index value**

With this record you can specify for node `index` the total pressure. Thus, the total pressure as normally calculated will be overruled with this `value`. 
6.887 nonlocal nonlocal_radius

By specifying this record in combination with a viscoplastic model, like group_materi_plasti_visco_power, a nonlocal yield rule \( f_n \) will be used in the viscoplastic law. The nonlocal yield rule needs to be initialized as dof by the materi_plasti_f_nonlocal record in the initialization part. The nonlocal yield rule \( f_n \) in a point is determined by an averaging of the local yield rule \( f \) in neighboring points and using gauss weighting functions for this (i.e. the larger the distance the less the neighboring point contributes to the nonlocal yield rule). The averaging is done over a region with radius nonlocal_radius.

In this way, you can prevent unlimited localization and so mesh dependency, in calculations with softening plasticity.

See also nonlocal_name.

6.888 nonlocal_name name

With name you specify the name of the plasticity model that should be treated nonlocal, eg -group_materi_plasti_mohr_coul. You can only specify one name, so only one plasticity model can be used as nonlocal model.

6.889 plasti_apply switch

If switch is set to -no, any plasticity data in the input file will be ignored. This is done for all timesteps.

This option is convenient for testing your input file just linear, without the need to outcomment each and every part with plasticity data. See also control_plasti_apply.

6.890 post_apply index switch

Setting switch to -no prevents post processing commands to be evaluated. Postprocessing commands have post in the name (only the post_node_rhside_ratio will be evaluated always, independent of post_apply).

Default, if post_apply is not specified, the switch is set to -yes.

6.891 post_calcul dof_0 operat_0 dof_1 operate_1...

This records activates calculation post results. The dof_0 dof_1 etc. refers to a dof, and possibilities for dof are listed below. The operat_0 operat_1 etc. refers to an operat, and possibilities for operat are listed below.

Here dof can be one of the matrices
- materi_stress,
- materi_strain_elasti,
- materi_strain_plasti,
- materi_strain_plasti_compression,
- materi_strain_plasti_diprisco,
- materi_strain_plasti_druckprag,
- materi_strain_plasti_hardsoil,
If operat
We denote a matrix stored for each and are stored for each post_line_dof record, and are stored for each post_point_dof record in a post_point_dof_calcul record, and are stored for each post_line_dof record in a post_line_dof_calcul record, and are stored for each post_quadrilateral_dof record in a post_quadrilateral_dof_calcul record.

The results of these calculations are stored for each node_dof record in a node_dof_calcul record, and are stored for each post_point_dof record in a post_point_dof_calcul record, and are stored for each post_line_dof record in a post_line_dof_calcul record, and are stored for each post_quadrilateral_dof record in a post_quadrilateral_dof_calcul record.

We denote a matrix dof with $A_{ij}$ and denote a vector dof with $A_i$, and denote a scalar dof with $a$. If operat is -absol then the absolute value of a scalar $a$ is calculated.

If operat is -average then $\frac{1}{3}(A_{11} + A_{22} + A_{33})$ is calculated for a matrix or $\frac{1}{3}(A_1 + A_2 + A_3)$ is calculated for a vector.

If operat is -negative then the average of the negative principal values for a matrix is calculated. If materi_strain_plasti is taken for the matrix $A_{ij}$, then this operator typically can be used as a measure for the amount of compression failure (crunching).

If operat is -positive then the average of the positive principal values for a matrix is calculated. If materi_strain_plasti is taken for the matrix $A_{ij}$, then this operator typically can be used as a measure for the amount of tensile failure (cracking).

If operat is -prival then three principal values of a matrix $A_{ij}$ are calculated. Each principal value contains the size of the principal vector. The principal values are ordered (the first value is the smallest one, and the last value is the largest one).

If operat is -privec then three principal vectors of a matrix $A_{ij}$ are calculated. Each principal vector contains the $x$, $y$ and $z$ component of the principal vector. The same ordering as used for -prival is used here also.

If operat is -size_tot then $\sqrt{A_{ij}A_{ij}}$ is calculated for a matrix or $\sqrt{A_iA_i}$ is calculated for a vector. This measures the size of a matrix or the size of a vector.

If operat is -size_dev then $\sqrt{B_{ij}B_{ij}}$ is calculated where $B_{ij}$ is the deviatoric part of a matrix $A_{ij}$; $B_{ij} = A_{ij} - \delta_{ij} \frac{A_{11} + A_{22} + A_{33}}{3}$ where $\delta_{ij}$ is 1 if $i = j$ and is 0 otherwise. This measures the size of the deviatoric part of the matrix.

Specially for -quad4, -quad9, -hex8 and -hex27 elements you can set operat to -force in case dof is -materi_stress. Then forces and moments are calculated in these isoparametric elements. See also post_calcul_materi_stress_force_element_group.

Specially for geotechnics you can set operat to -total_pressure in case dof is -materi_stress. Then the total stress is calculated from the effective stress and the groundflow total pressure. This option is not valid in combination with undrained pressures as obtained by group_materi_undrained_capacity.

Specially for geotechnics you can set operat to -static_pressure in case dof is -groundflow_pressure. Then the static pressure is calculated.

Specially for geotechnics you can set operat to -dynamic_pressure in case dof is -groundflow_pressure. Then the dynamic pressure is calculated.

Specially for geotechnics you can set operat to -k0 in case dof is -materi_stress. Then the ratio of horizontal and vertical stresses is calculated. If 2D this is the ratio $0.5 \frac{\tau_{xx}}{\tau_{yy}}$. If 3D this is the...
ratio $0.5\frac{\sigma_{xx} + \sigma_{yy}}{\sigma_{xx}}$.

Specially for geotechnics you can set `operat` to `-young_apparent` in case `dof` is `-materi_stress`. Then the apparent Young modulus is calculated from the incremental strains and incremental stresses. If determination is not possible (for almost zero incremental strains) the apparent young is put to 0. See also `post_calcul_apparent_total`.

Specially for geotechnics you can set `operat` to `-poisson_apparent` in case `dof` is `-materi_stress`. Then the apparent Poisson ratio is calculated from the incremental strains and incremental stresses. If determination is not possible (for almost zero incremental strains) the apparent poisson is put to 0. See also `post_calcul_apparent_total`.

Specially for geotechnics you can set `operat` to `-total_pressure` in case `dof` is `-groundflow_pressure`. Then the total pressure is calculated.

The next piece of input file

```plaintext
... materi_stress materi_strain_plasti end_initia ...
post_calcul -materi_stress -size_dev -materi_strain_plasti -size_tot ...
control_timestep 1 ...
control_print 1 -node_dof_calcul
```

will print records like

```plaintext
node_dof_calcul index 0.2 1.1e-4
```

Here the 0.2 is the equivalent Von Mises stress and $1.1e-4$ measures the plastic strain matrix.

See also `post_calcul_absolute` and `post_calcul_label`.

```
6.892 post_calcul_absolute switch
```

If `switch` is set to `-yes` all results of `post_calcul` are set to be positive values. This may be done if you prefer positive values in your presentation of results.
6.893  **post_calcul_apparent_total**  *switch*

If *switch* is set to *-yes*, the total strains and total stresses (as opposed to incremental) will be used to calculate the apparent young modulus and poisson ratio. If *switch* is set to *-no*, the incremental strains and incremental stresses will be used. Default, if *post_calcul_apparent_total* is not specified, the *switch* is set to *-no*.

6.894  **post_calcul_label**  *doflabel_0 label_1 ...*

This record will be filled with the names of the data that is calculated by means of the *post_calcul* option. The first name comes from the first *dofoperat* in *post_calcul*, the second name comes from the second *dofoperat* in *post_calcul*, etc. You can find this record in the dbs file after a calculation.

6.895  **post_calcul_limit**  *lower_0 upper_0 lower_1 upper_1 ...*

With this record you can specify the lower and upper allowed values for all calculated results. With *lower_dof_0* you specify the lower allowed value for the first result. With *upper_dof_0* you specify the upper allowed value for the first result. Etc.

6.896  **post_calcul_materi_stress_force_average**  *switch*

See first *post_calcul_materi_stress_force_element_group*.

This *post_calcul_materi_stress_force_average* option is only available for *quad9* and *hex27* elements. It can be used if forces and moments are primarily calculated in two opposing end faces of the *quad9* and *hex27* element. If *switch* set to *-yes*, the forces and moments of nodes in the plane between the two end faces will be set to the averaged values from the forces and moments on the two opposing end faces. If *switch* set to *-no* this is not done. Default *switch* is *-yes*.

6.897  **post_calcul_materi_stress_force_direction_exclude**  *dir_x dir_y dir_z*

See first *post_calcul_materi_stress_force_element_group*.

In 3D, Tochnog needs to know for which element sides it should determine forces and moments. For this purpose you need to specify this direction *dir_x dir_y dir_z*. All element sides with normals in this direction will be neglected; no forces and moments will be determined for such sides.

Typically, in a tunnel calculation you take the tunnel length direction as *dir_x dir_y dir_z*.

6.898  **post_calcul_materi_stress_force_direction_exclude_epsilon**  *eps*

With *eps* you can influence which normals are considered to be in the specified exclude direction. A small *eps* specifies that only very precise normals in the specified direction will be excluded. A large *eps* specifies that also not precise normals in the specified direction will be excluded. In
fact $\text{eps}$ is the difference from inproduct between the specified exclude direction with the normal direction and 1. Default $\text{eps}$ is $1.e^{−8}$.

6.899 post_calcul_materi_stress_force_direction_include $\text{dir}_x \text{dir}_y \text{dir}_z$

See first post_calcul_materi_stress_force_element_group.

In 3D, Tochnog needs to know for which element sides it should determine forces and moments. For this purpose you need to specify this direction $\text{dir}_x \text{dir}_y \text{dir}_z$. All element sides with normals perpendicular to this direction will be neglected; no forces and moments will be determined for such sides.

Typically, in a sheet pile calculation you take the sheet pile height direction as $\text{dir}_x \text{dir}_y \text{dir}_z$.

6.900 post_calcul_materi_stress_force_direction_include_epsilon $\text{eps}$

With $\text{eps}$ you can influence which normals are considered to be perpendicular to the specified include direction. A small $\text{eps}$ specifies that only normals precisely perpendicular to the specified direction will be excluded. A large $\text{eps}$ specifies that also normals not precisely perpendicular to the specified direction will be excluded. In fact $\text{eps}$ is the difference from inproduct between the specified include direction with the normal direction and 0. Default $\text{eps}$ is $1.e^{−8}$.

6.901 post_calcul_materi_stress_force_element_group $\text{element\_group\_0}$ $\text{element\_group\_1}$ ...

With the post_calcul -materi_stress -force option the normal force, shear force and moment(s) are calculated for the isoparametric elements -quad4, -quad9, -hex8 and -hex27. This option is meant for structures like sheet piles, tunnel shells, etc. where there is only 1 element over the thickness of the structure. Thus the element has a thickness equal to the complete thickness of the structure, and the length of the element is a part of the total length of the structure (e.g. tunnel length).

In the following definitions of forces and moments, $n$ denotes the normal to an element side, $t$ denotes the thickness direction in the side, and $l$ denotes the length direction. The 2D and 3D normal force nor results is defined by the normal stresses $\sigma_{nn}$ integrated over the thickness. The 2D and 3D shear force she results is defined by the shear stresses $\sigma_{nt}$ integrated over the thickness. The 2D moment mom and 3D moment mom1 are defined by the moment contributions of normal stresses $\sigma_{nn}$ with a distance in thickness direction $d_t$ relative to the middle of the element, integrated over thickness direction (radial bending moment in tunnel shell, thickness bending moment in sheet pile, etc.). The 3D moment mom2 is defined by the moment contributions of normal stresses $\sigma_{nn}$ with a distance in length direction $d_l$ relative to the middle of the element, integrated over thickness direction (bending moment in tunnels, sheet piles, etc.).

The forces and moments will be calculated per unit length $l$ of the isoparametric element, where $l$ is the size of the element in length direction. In a 3D calculation, the length of an element is determined from the nodal coordinates differences in length direction. In a axi-symmetric 2D calculation, the length of the elements is set to $2 * \text{PI} * \text{radius}$ by Tochnog (notice that with this definition values cannot be calculated at the symmetry axis with zero radius). In a plane 2D calculation, the length of the elements is set to 1 by Tochnog.
The normal force and moment(s) are given the proper sign (plus or minus). For example, a positive normal force means that the structure is under tension. For the shear force, however, always a positive value is calculated by Tochnog, so only the size of the shear force is available (and not the direction of the shear force).

For all of the forces and moment vectors, we want to display the vector in thickness direction of the structure, to get a clear view in postprocessors (e.g. GID). Thus, the components in global x- and y-direction are determined such that the vector direction is in thickness direction of the structure. Because of this, the components by themselves are not the real physical components of the force or moment; they are only convenient values for getting clear plots in postprocessors. However, the size of the vector formed by these components (square root of components squared), indeed is the real physical size of the force or moment, so the size can indeed be used for design purposes. For your convenience, the size of each vector is also calculated automatically be Tochnog. For example, for the normal forces Tochnog calculates $-\text{nor}_x\_\text{sig}$, $-\text{nor}_y\_\text{sig}$ and $-\text{nor}_z\_\text{sig}$ which are the global plot vector x-component, y-component and the physical real size respectively.

The enable a correct force or moment direction in either the positive or negative thickness direction, Tochnog wants you to specify **post_calcul_materi_stress_force_reference_point**.

In 3D, you need to specify either **post_calcul_materi_stress_force_direction_exclude** or **post_calcul_materi_stress_force_direction_include**. With these records you can determine for which element sides forces and moments should be determined. The direction and element should be such that for each element for which you want to determine forces and moments exactly 4 sides should be consistent with the specified direction. Otherwise the present option for determination of forces and moments is not available for the element. Only one of **post_calcul_materi_stress_force_direction_exclude** and **post_calcul_materi_stress_force_direction_include** should be specified, not both.

The element_group_0 element_group_1... of this **post_calcul_materi_stress_force_element_group** specify the groups of isoparametric elements for which the forces and moments should be determined by Tochnog.

Summary of conditions for the **post_calcul_materi_stress_force** option to work well:

- Only 1 element in thickness direction.
- Elements in 3D should be regular shaped in length direction. That is, the element sides perpendicular to the length direction should be completely parallel.
- At least 1 timestep should be done (since element forces needed for this option are setup in a timestep)

6.902 **post_calcul_materi_stress_force_reference_point** $x_0 y_0 z_0$

$\ldots x_1 y_1 z_1 \ldots$

See first **post_calcul_materi_stress_force_element_group**.

For example tunnels typically are of circular or piecewise circular geometry. To get a correct direction of the calculated forces and moments, Tochnog needs to know the approximate middle point of the tunnel, so that it can put all negative forces and moments and positive forces and moments consistently outwards or inwards in thickness direction of the structure. Thus, you need to specify with this **post_calcul_materi_stress_force_reference_point** record the
approximate middle point of the tunnel that you are evaluating for each of the element groups. In case you have a sheet pile, you should specify a reference point on a large perpendicular distance away from the sheet pile.

You need to specify a reference point for each element group specified in `post_calcul_materi_stress_force_element_group`.

In 3D you need to specify the x, y and z value for each reference point. In 2D you only need to specify the x and y value for each reference point.

See also `post_calcul_materi_stress_force_plot_switch`.

6.903 `post_calcul_materi_stress_force_outer switch`

If `switch` is set to `-yes`, the forces and moments are only calculated for the nodes at the outer sides of the elements; these are the nodes which have the furthest distance relative to the reference point. This will give a bit more nice vector plots.

Default, if `post_calcul_materi_stress_force_outer` is not specified, `switch` is set to `-no`. This will give a bit more nice contour fill plots.

6.904 `post_calcul_materi_stress_force_plot_switch switch_0 switch_1 ...`

If you don’t like the direction in which tochnog draws the vectors (outward or inward), you can switch the direction by setting the corresponding switch to `-yes`. In 2D you need to specify a switch for the normal force, shear force and moment. In 3D you need to specify a switch for the normal force, shear force and two moments.

6.905 `post_calcul_materi_stress_force_thickness_switch switch_element_group_0 switch_element_group_1 ...`

See first `post_calcul_materi_stress_force_element_group`.

In 3D Tochnog normally assumes that the shortest element direction in the side where forces and moments are calculated is the structure thickness direction. If that is not the case, e.g. if you have very short elements in a tunnel length direction, then you need to explain Tochnog that it should switch to the longest element direction as structural thickness direction, by setting a to `-yes`.

This ensures that the shear force is always really calculated over the structural thickness, and the first moment is really the moment over the structural thickness.

If you specify `post_calcul_materi_stress_force_thickness_switch` you need to give a switch for each element group of `post_calcul_materi_stress_force_element_group`.

6.906 `post_calcul_multiply factor_0 factor_1 ...`

With this record you can specify a multiplication factor for each calculated item. This comes handy when you prefer another definition. If you specify `post_calcul_multiply`, you need to give a factor for each item.
6.907  **post_calcul_safety_default**  *eps value*

Specifically for safety lifting and piping calculations division by pressure values equal to zero can occur. With this **post_calcul_safety_default** record you can prevent such division. If the absolute of the pressure value for the division is smaller than the specified *eps*, the safety factor will be set to the user specified *value*.

In case this record is not specified we set *eps* to something very small, and *value* to 0.

6.908  **post_calcul_safety_maximum**  *value*

Specifically for safety lifting and piping calculations you can limit the calculated safety factor to this specified maximum *value*. This comes convenient if you get excessive large values in a calculation due to numerical accuracy in the calculation.

In case this record is not specified we set *value* to something very large.

6.909  **post_calcul_safety_method**  *method*

You can determine with this option how the hydraulic piping and lifting safety should be determined. If you set *method* to **-vertical** the safety factors will be determined using the vertical stress (zz-stress in 3D, yy-stress in 2D, xx-stress in 1D); this is the default, as described in the **post_calcul** command. Thus you get one value for the safety piping and one value for the safety lifting.

If you set *method* to **-prival** the safety factors will be determined using the three principal stresses (principal stress 0, principal stress 1, principal stress 2). Thus you get three values for the safety piping and three values for the safety lifting.

If you set *method* to **-global** the safety factors will be determined using the three global normal stresses (xx-stress, yy-stress, zz-stress). Thus you get three values for the safety piping and three values for the safety lifting.

Setting *method* to **-vertical** is the classical definition used in most text books. However since the critical direction in complex calculations will not always be in the vertical direction, it is also of interest to study the hydraulic safety factors with the principal stress values (**-prival**). And for some situation, like soil near a retaining wall, it may be of interest to study the hydraulic safety factors with global normal stresses (**-global**).

In the dbs file after a calculation you can see in the **post_calcul_label** record the naming of the calculated hydraulic safety factors.

6.910  **post_calcul_static_pressure_height**  *coord_min,0 coord_max,0 height_ref,0 coord_min,1 coord_max,1 height_ref,1 . . .*

Using this option the static pressure as required by **post_calcul-groundflow_pressure-static_pressure** is determined relative to the reference height, and not anymore to a groundwater level. Thus, the $\Delta z$ in the equation for $p_{\text{static}} = \rho g \Delta z$ is taken relative to the specified reference height in this **post_calcul_static_pressure_height** record.

You can specify multiple regions. The first region is between vertical coordinate *coord_min,0* and *coord_max,0*. The *coord_min,0* and *coord_max,0* themselves are included as part the region. If a node is inside this region the *height_ref,0* is used as phreatic level height in the equation for the
static pressure. The second region is between vertical coordinate $\text{coord}_\text{min},1$ and $\text{coord}_\text{max},1$. The $\text{coord}_\text{min},1$ and $\text{coord}_\text{max},1$ themselves are included as part the region. If a node is inside this region the $\text{height}_\text{ref},1$ is used as phreatic level height in the equation for the static pressure.

If a node is not inside any of the regions, and if the groundwater phreatic level itself is not specified, the static pressure cannot be determined and remains zero.

See also post_calcul_static_pressure_height_element_group.

6.911 post_calcul_static_pressure_height_element_group element_group_0 element_group_1 ...

Restrict the regions of post_calcul_static_pressure_height to specific element groups. The region between $\text{coord}_\text{min},0$ and $\text{coord}_\text{max},0$ is valid for element group element_group_0. The region between $\text{coord}_\text{min},1$ and $\text{coord}_\text{max},1$ is valid for element group element_group_1. Etc.

You need to specify an element group for each and every region. As a special option you can specify -all for an element group number; then the corresponding region is valid for all element groups.

6.912 post_count dataitem_name_0 dataitem_name_1 ...

With this post_count record you can specify data items for which the number of active indices should be counted. The results will be placed in the record post_count_result.

For example count the number of active elements, nodes and geometry points by:

```
... post_count -element -node -geometry_point ...
```

6.913 post_data index dataitem_name_0 dataitem_index_0 dataitem_number_0 dataitem_name_1 dataitem_index_1 dataitem_number_1 ...

The specified data items are taken, and each is multiplied with its corresponding factor in post_data_factor and added to post_data_result. This allows you to conveniently follow the sum of data item, each multiplied with some factor.

6.914 post_data_factor index factor_0 factor_1 ...

See post_data.

6.915 post_data_result index result

See post_data.
6.916  **post_element_force**  

```plaintext
index dir_normal_x dir_normal_y dir_normal_z
  dir_shear0_x dir_shear0_y dir_shear0_z dir_shear1_x dir_shear1_y dir_shear1_z
  middle_x middle_y middle_z
```

With this record you can calculate the normal force, shear force and moments in cross sections. Only cross sections at the side of elements are allowed; so that typically is the common side between two elements, or the side at the edge of a domain; a cross section through the interior of elements is not allowed. Below we will describe how you can select elements. For the combination of selected elements nodal forces will be used to determine cross section forces and moments.

The `middle_x middle_y middle_z` should contain the exact middle coordinates of the cross section. You can use this `post_element_force` option however also to get the soil forces on the shaft or toe of a pile; then use a `middle_x middle_y middle_z` point below the bottom of the pile.

The nodal force components in the `dir_normal_x dir_normal_y dir_normal_z` direction are summed to give a normal force `normal_force`. The nodal force components in the `dir_shear0_x dir_shear0_y dir_shear0_z` direction are summed to give the first shear force `shear0_force`. The nodal force components in the `dir_shear1_x dir_shear1_y dir_shear1_z` direction are summed to give the second shear force `shear1_force`. The nodal force components in the `dir_normal_x dir_normal_y dir_normal_z` direction are multiplied with the distance in `dir_shear0_x dir_shear0_y dir_shear0_z` direction as measured from the `middle_x middle_y middle_z` vector, and this is summed to give the first bending moment `moment0`. The nodal force components in the `dir_normal_x dir_normal_y dir_normal_z` direction are multiplied with the distance in `dir_shear1_x dir_shear1_y dir_shear1_z` direction as measured from the `dir_shear0_x dir_shear0_y dir_shear0_z` vector, and this is summed to give the first bending moment `moment1`. The results for the normal force, two shear forces and two moments will be placed in the record `post_element_force_result`.

In 3D you need to specify the complete `post_element_force` record and you get the normal force, two shear forces and two bending moments in the `post_element_force_result` record. The directions `dir_shear0_x dir_shear0_y dir_shear0_z` and `dir_shear1_x dir_shear1_y dir_shear1_z` should be perpendicular.

In 2D you need to specify only a partial record `post_element_force` as `index dir_normal_x dir_normal_y dir_shear0_x dir_shear0_y middle_x middle_y` and you get the normal force, one shear force and one bending moment in the `post_element_force_result` record.

In 1D you need to specify only a partial record `post_element_force` as `index dir_normal_x middle_x` and you get the normal force in the `post_element_force_result` record.

You can restrict with `post_element_force_geometry` with the same index that the `post_element_force` is only evaluated for nodes on a specific geometry. This `post_element_force` option always checks if the initial node location is located on this geometry (so not the location of moved nodes after displacement of the nodes).

You can restrict with `post_element_force_group` with the same index that the `post_element_force` is only evaluated for certain element groups.

You can restrict with `post_element_force_number` with the same index that the `post_element_force` is only evaluated for certain element numbers.

You can restrict with `post_element_force_normal` with the same index that the `post_element_force` is only evaluated for elements in positive normal direction `dir_normal_x dir_normal_y dir_normal_z`. If you don’t specify `post_element_force_normal` elements on both sides will be used if present.

You can require by setting the `switch` in `post_element_force_force` with the same index that also the external forces (like gravity and edge loads etc.) are added to the result.

You can require by setting the `switch` in `post_element_force_inertia` with the same index
that also the inertia forces is added to the result.

If you are not happy with the sign or units with which the forces are calculated, you can use a multiply factor in `post_element_force_multiply_factor` with the same index to get what you want.

Please realise that in calculation with groundwater the calculated forces contain the force due to effective stresses and also due to groundwater total pressure (pore pressure).

We now give some examples for a 2D vertical pile driven into the soil in a dynamic inertia ... calculation, and including gravity `force_gravity` ... and an external force `force_element_edge` ... at the top of the pile. Below `x_pile` is the x-coordinate at the middle of the pile, `y_pile_middle` is the y-coordinate at the middle of the pile, `y_pile_bottom` is the y-coordinate at the bottom of the pile and `pile_group` is the group number of the pile.

The force in a cross section (force resulting from normal stress in cross section):

```plaintext
... post_element_force 10 0. 1. 1. 0. x_pile y_pile
post_element_force_geometry 10 -pile_cross_section
post_element_force_group 10 pile_group
...
```

Here `pile_cross_section` is a geometry line through the cross section of the pile,

The force along the shaft (force resulting from shear stress along shaft):

```plaintext
... post_element_force 10 0. 1. 1. 0. x_pile y_pile_bottom
post_element_force_geometry 10 -pile_shaft
post_element_force_group 10 pile_group
post_element_force_force 10 -yes
post_element_force_inertia 10 -yes
...
```

Here `pile_shaft` is a geometry line containing only nodes of the pile shaft,

The force at the pile toe (force resulting from normal stress at pile tip):

```plaintext
... post_element_force 10 0. 1. 1. 0. x_pile y_pile_bottom
post_element_force_geometry 10 -pile_toe
post_element_force_group 10 pile_group
post_element_force_force 10 -yes
post_element_force_inertia 10 -yes
...
```

...
Here \textit{pile\_toe} is a geometry line containing only nodes of the pile toe.

The complete force on the pile:

\begin{verbatim}
... 
post_element_force 10 0. 1. 1. 0. x_pile y_pile_bottom
post_element_force_geometry 10 -pile\_complete
post_element_force_group 10 pile\_group
post_element_force_force 10 -yes
post_element_force_inertia 10 -yes
... 
... 
\end{verbatim}

Here \textit{pile\_complete} is a geometry line containing all nodes of the pile.

Also see the example calculation \texttt{force14.dat} and \texttt{force17.dat}.

\textbf{6.917 post\_element\_force\_force} \textit{index switch}

See \texttt{post\_element\_force}.

\textbf{6.918 post\_element\_force\_geometry} \textit{index geometry\_item\_name geometry\_item\_index}

See \texttt{post\_element\_force}.

\textbf{6.919 post\_element\_force\_group} \textit{index element\_group\_0 element\_group\_1}

\texttt{...}

See \texttt{post\_element\_force}.

\textbf{6.920 post\_element\_force\_inertia} \textit{index switch}

See \texttt{post\_element\_force}.

\textbf{6.921 post\_element\_force\_multiply\_factor} \textit{index multiply\_factor}

See \texttt{post\_element\_force}.

\textbf{6.922 post\_element\_force\_normal} \textit{index switch}

Set \textit{switch} to \texttt{-yes} if you want to select elements in positive normal direction. See \texttt{post\_element\_force}.
6.923  **post_element_force_number**  *index number_0 number_1 ...*

See **post_element_force**.

6.924  **post_element_force_result**  *index normal_force shear0_force shear1_force moment0 moment1*

See **post_element_force**.

6.925  **post_global**  *switch*

With this **post_global** you can ask for global information to be determined if you set *switch* to *-yes*. The following information will then be determined:

- **-post_bounda_force_summed** (total force following from **-bounda_force** records, number of principal dofvalues)
- **-post_element_mass_summed** (total global mass)
- **-post_element_summed** (total number of elements)
- **-post_element_volume_summed** (total elements volume without empty elements)
- **-post_group_summed** (total number of elements in group 0, group 1, etc.)
- **-post_materi_inertia_summed** (sum of material nodal inertia, so of **node_inertia**)
- **-post_slide_force_summed** (sum of slide forces in global axes, so of **node_slide_force**)
- **-post_node_summed** (total number of nodes)
- **-post_node_dof_average** (average values for dof’s)
- **-post_node_dof_maximum** (maximum values for dof’s)
- **-post_node_dof_minimum** (minimum values for dof’s)
- **-post_force_edge_summed** (total force following from **-force_edge** integrated over edges in x,y,z directions, *number_of_space_dimensions* values)
- **-post_force_edge_normal_summed** (total force following from **-force_edge_normal** integrated over edges in x,y,z directions, *number_of_space_dimensions* values)
- **-post_force_edge_projected_summed** (total force following from **-force_edge_projected** integrated over edges in x,y,z directions, *number_of_space_dimensions* values)
- **-post_support_edge_normal** (total force following from **-support_edge_normal** integrated over edges in x,y,z directions, *number_of_space_dimensions* values)
- **-post_solver_diagonal_minimum_value** (minimum diagonal term total matrix, only for pardiso solver)
- **-post_solver_diagonal_minimum_node** (node number at which the minimum value is found)
- **-post_solver_diagonal_maximum_value** (maximum diagonal term total matrix, only for pardiso solver)
• **-post_solver_diagonal_maximum_node** (node number at which the maximum value is found)

• **-post_solver_diagonal_ratio** (ratio maximum/minimum diagonal terms total matrix, only for pardiso solver)

• **-post_solver_iterations** (total number of iterations of iterative linear equation solver, only for bicg solver)

If you set `switch` to `-no` then the information will not be determined (this saves a little bit of computer time). Default, if `post_global` is not specified, `switch` to `-yes`.

### 6.926 post_group_volume_summed

`volume_group_0 volume_group_1 ...`

This record will be filled with the total volume of the elements in each group. So `volume_group_0` is the summed volume of the elements in element group 0, etc.

### 6.927 post_integrate

`index data_item_name data_item_index data_item_number ...`

Here you can specify results that should be integrated over time. The integrated results will be placed in the `post_integrate_result` record with the same index.

An example looks like:

```plaintext
...  
groundflow_pressure  
groundflow_velocity  
end_initia  
...
post_node 1 -average -geometry_line 4  
...
post_integrate 3 -post_node_result 1 -gvely  
...
...`

Here the `post_node` record first takes care that the average groundflow y-velocity at nodes on a line are determined, among other dof’s. The `post_integrate` record integrates that average groundflow y-velocity over time. In this way the total groundflow debit volume over a line is registered.

### 6.928 post_integrate_result

`index result`

See `post_integrate`.

### 6.929 post_line

`index x_0 y_0 z_0 x_1 y_1 z_1`

This record specifies a line in space for which the average or sum of the dof values will be calculated. The values are placed in a record `post_line_dof` with the same `index`. Internally in TOCHNOG,
post_point records are used to evaluate the dof’s on the line. In 1D only $x_0$ and $x_1$ should be specified, etc. In the example below, the average of the $x$-velocity between the points (3,1) and (3,7) will be printed

```plaintext
... number_of_space_dimensions 2 materi_velocity ...
end_data ...
post_line 1 3. 1. 3. 7.
...
print_filter 0 -post_line_dof 1 -velx ...
control_timestep 1 1. 100.
control_print 1 -post_line_dof
```

The coordinates are defined in the initial mesh. See also: post_line_n and post_line_operat.

6.930 post_line_operat index operat

If operat is set to -average then the average is calculated for the post_line record with the same index. If operat is set to -sum then the sum is calculated for the post_line record with the same index.

If this post_line_operat is not specified, then operat is set to -average.

6.931 post_line_dof index dof_0 dof_1 ...

Average dofvalues at a selected line. See post_line.

6.932 post_line_dof_calcul ...

See post_calcul.

6.933 post_line_n index n

Use $n$ post_point records to evaluate the dof’s along the line. Default $n$ is 5. See post_line.

6.934 post_node index data_item operat geometry_entity_name geometry_entity_index

If operat is set to -sum, results for the nodal data_item are summed. For example, you can take for data_item the -node_rhside and for operat you take -sum so that the total external force on a boundary is determined.

If operat is set to -average, results for the nodal data_item are averaged.
If `data_item` is set to `-node_rhside` and `operat` is set to `-moment`, the moment is determined around the origin \((x=0,y=0)\) in a 2D calculation; this is done by calculating in the nodes the tangential force extracted from `-node_rhside` and multiplying it with the radial distance from the origin.

This operation is done for nodes which are placed on the geometrical entity `geometry_entity_name` `geometry_entity_index`. Instead of a geometrical entity you can also use `-all` to tell that all nodes should be used. Instead of a geometrical entity you can also use `-ra ... -ra` to tell that the nodes of the range should be used.

The result of this `post_node` record is put into the `post_node_result` record (with the same `index`).

**6.935 post_node_factor index factor**

You can multiply the result of `post_node` with `factor`. Default, if `post_node_factor` is not specified, we take `factor` equal to 1.

**6.936 post_node_result index result_0 result_1 ...**

See `post_node`.

**6.937 post_node_rhside_fixed value_0 value_1 ...**

This record will be filled with the average of the norm of `node_rhside` for those dof’s which are prescribed (eg with a `bounda_dof`). For example, in a calculation with only velocities (displacements) as primary dof’s, this record contains the average of the reaction force at the nodes in which the velocity is prescribed. Values are only filled for principal dof’s (materi velocity, groundflow pressure, condif temperature, ...).

**6.938 post_node_rhside_free value_0 value_1 ...**

Same as `post_node_rhside_fixed`, now for free values however. For example, in a calculation with only velocities (displacements) as primary dof’s, this record contains the average of the unbalance force at the nodes in which the velocity is not prescribed.

**6.939 post_node_rhside_ratio ratio**

This record gives during a calculation a measure for the inaccuracy of the calculation. For each primary dof type the ratio between the size of the corresponding parts in `post_node_rhside_fixed` and `post_node_rhside_free` is determined.

If the size `post_node_rhside_fixed` is below 1.e-10 the `ratio` is directly filled with `post_node_rhside_free`. See also `post_node_rhside_ratio_dof_type`. 
6.940  \texttt{post\_node\_rhs}ide\_ratio\_dof\_type \ dof\_type \ 0 \ldots

With this option you can specify a list of doftypes which should be used in the calculation of the \texttt{post\_node\_rhs}ide\_ratio result. For example, if both \texttt{groundflow\_pressure} and \texttt{condif\_temperature} are initialised, then you can use only the groundflow pressure in the accuracy ratio determination by specifying \texttt{post\_node\_rhs}ide\_ratio\_dof\_type \texttt{-groundflow\_pressure}.

If \texttt{post\_node\_rhs}ide\_ratio\_dof\_type is not specified and \texttt{materi\_velocity} is initialised then automatically \texttt{post\_node\_rhs}ide\_ratio\_dof\_type \texttt{-materi\_velocity} will be used.

6.941  \texttt{post\_node\_rhs}ide\_ratio\_method \ method

By setting \texttt{method} to \texttt{-post\_node\_rhs}ide\_free the ratio is directly filled with \texttt{post\_node\_rhs}ide\_free. Default, when this \texttt{post\_node\_rhs}ide\_ratio\_method record is not specified, the default definition as specified in \texttt{post\_node\_rhs}ide\_ratio is used.

6.942  \texttt{post\_point} \ index \ x \ y \ z

This record specifies a point in space for which \texttt{dof} values will be calculated. The values are placed in a record \texttt{post\_point\_dof} with the same \texttt{index}. The values are obtained by determining in which element the point is located and then using the element’s interpolation functions. In 1D only \texttt{x} should be specified, etc.. The coordinates are defined in the initial mesh. This option is only available for isoparametric elements.

6.943  \texttt{post\_point}\_element\_group \ index \ element\_group

Limit the search for the element in which the post point with the same \texttt{index} is located to the specified \texttt{element\_group}.

6.944  \texttt{post\_point\_dof} \ index \ dof\_0 \ dof\_1 \ldots

Unknown values at a selected point. See \texttt{post\_point}.

6.945  \texttt{post\_point\_dof}\_calcul \ldots

See \texttt{post\_calcul}.

6.946  \texttt{post\_point\_eps}\_iso \ index \ eps

Tolerance with which a \texttt{post\_point} is accepted to be part of an element. The default value is 1.e-3. You can increase the default value if a \texttt{post\_point} is exactly on or over the border of the mesh, so that the \texttt{post\_point} may be not found; typically try 0.1 or so.
6.947 **post_quadrilateral**  
*index x_0 y_0 z_0 x_1 y_1 z_1 x_2 y_2 z_2 x_3 y_3 z_3*

This record specifies a quadrilateral in space for which the average of the dof values will be calculated. The values are placed in a record **post_quadrilateral_dof** with the same *index*. Internally in TOCHNOG, **post_point** records are used to evaluate the dof’s on the quadrilateral. In 2D only $x_0 y_0, x_1 y_1$, etc. should be specified. The coordinates are defined in the initial mesh. See also: **post_quadrilateral_n**.

6.948 **post_quadrilateral_dof**  
*index dof_0 dof_1 ...*

Average dof values at a selected quadrilateral. See **post_quadrilateral**.

6.949 **post_quadrilateral_dof_calcul** ...

See **post_calcul**.

6.950 **post_quadrilateral_element_group**  
*index element_group*

Select the specific element group from which the dof values should be taken.

6.951 **post_quadrilateral_n**  
*index n*

Use $n$ **post_point** records in each direction to evaluate the dof’s along the quadrilateral. Default $n$ is 5. See **post_quadrilateral**.

6.952 **post_strain_volume_absolute**  
*index volume_increase_absolute*

This record will hold after the calculation the absolute volume increase summed over the elements that are selected in the **strain_volume_element**, **strain_volume_element_group** and **strain_volume_geometry** records (with the same index).

The actual volume increase which you will find in this **post_strain_volume_absolute** record will depend on the relative volume strain or absolute volume increase that you specified, but also on stiffnesses of neighboring zones, boundary conditions, etc.

You can use this **post_strain_volume_absolute** result to decide to manually change the specified relative volume strain or absolute volume increase and rerun the calculation.

6.953 **post_strain_volume_initial**  
*index volume_initial*

Initial volume of selected elements.

6.954 **post_strain_volume_relative**  
*index volume_strain_relative*

Relative volume strain percentage. Otherwise the same as **post_strain_volume_absolute**.
6.955 print_apply switch

If switch is set to -no, then all control_print_* records will not be applied. Default, if print_apply is not specified, switch is set to -yes.

6.956 print_arithmetic switch

If switch is set to -yes, all evaluated arithmetics will be printed. See the start of the data part for an explanation about arithmetics. The printing will be done to the file tochnog_arithmetic.txt.

6.957 print_control switch

If switch is set to -yes, the control index being evaluated will be printed. Handy for keeping track on what the program is doing.

6.958 print_data_name switch

If switch is set to -yes, all possible data names will be printed. The printing will be done to the file tochnog_data_name.txt.

This is convenient to search in the tochnog_data_name.txt file fast for options. For example under linux to search all options which have the word group in it do grep group tochnog_data_name.txt.

The possible data names may include also internal names that Tochnog uses during the calculation; so for each name you can check this users manual if it is a name that you can use as input in the input file, or not.

6.959 print_database_calculation switch

If switch is set to -yes, the database will be written after successful completion of a calculation to the file name.dbs, where name is the name of the input file. If switch is set to -no, the database will not be written.

Default, switch is set to -yes.

6.960 print_define switch

If switch is set to -yes, all evaluated defines will be printed. See the start of the data part for an explanation about defines. The printing will be done to the file tochnog_define.txt.

6.961 print_element_geometry_present switch

See element_geometry_present. See also print_element_geometry_present_node_type. Default switch is set to -no.
6.962 print_element_geometry_present_node_type node_type

If node_type is set to -node_start_refined the values of -node_start_refined are used. If node_type is set to -node the values of -node are used. If node_type is set to -plus_displacement the values of -node plus nodal displacements. Default node_type is set to -node_start_refined.

6.963 print_failure switch

If switch is set to -yes then failure of elements due to one of the failure criteria (group_materi_failure_rupture etc.) will be reported.

6.964 print_filter index data_item_name data_item_index number_0 number_1 ...

The data selected in the records control_print, control_print_dof, control_print_dof_rhside and control_print_element will be filtered at output. Thus only a limited amount of data will actually be printed. Here data_item_name is the name of the data item to be filtered, e.g. data_item_name is -node_dof. data_item_index is the index of the data_item_name record which passes the filter. If, for example, data_item_index is 3 then only index 3 passes the filter. If data_item_index is -all then all indices pass the filter. If, for example, data_item_index is -geometry_line 3 (valid if data_item_name is -node or another nodal item) then only records with coordinates located on line 3 pass the filter. If, for example, data_item_index is -geometry_line 3 (valid if data_item_name is -node or another nodal item) then only element with at least one coordinate located on line 3 pass the filter. If, for example, data_item_index is -ra . . . -ra then indices in this range pass the filter. If, for example, data_item_index is -macro 4 and data_item_name is data valid at a node (or element), then only nodes (or elements) generated by the macro number 4 pass the filter (see control_mesh_macro_* for macro’s). If, for example, data_item_index is -macro -none and data_item_name is data valid at a node (or element) then only nodes (or elements) not generated by any macro pass the filter (see control_mesh_macro_* for macro’s).

For example, if number_0 is 3 then the fourth value of a record passes the filter. If number_0 is -all the whole record passes the filter. If, for example, number_0 is -velx while data_item_name is -node_dof then only x-velocities pass the filter.

Some examples are

    print_filter 1 -node_dof-all -temp -sigxx (temperatures and xx-stresses)
    print_filter 2 -node -geometry_line 3 0 (x-coordinates on line 3)

With control_print_filter you can select if the records control_print, control_print_dof or control_print_dof_rhside (with the same index) should use specific filters (specify the indices of the filter for print_filter_index), should use all filters (specify -all for print_filter_index), or should use no filter at all (specify -none for print_filter_index). Default, if control_print_filter is not specified, all filters will be used for a print option.

Example:

    print_filter 1 -node_dof ...
    print_filter 2 -node_dof_all ...
All used filters are placed in-line for a data item; thus only data which passes all used filters for that data item will be printed.

6.965 print_gid_calculation switch

If you set switch to -yes the gid files will be printed at the end of the calculation. If you set switch to -no the gid files will not be printed at the end of the calculation. Default, if switch is not specified, it is set to -yes.

6.966 print_frd_freecad switch

See control_print_frd_freecad, but now for all frd printing however. If both control_print_frd_freecad and print_frd_freecad are present, the control_print_frd_freecad dictates what happens for the specific control index.

If both print_frd_freecad and control_print_frd_freecad are not specified the switch is set to -no.

6.967 print_frd_prepomax switch

See control_print_frd_prepomax, but now for all frd printing however. If both control_print_frd_prepomax and print_frd_prepomax are present, the control_print_frd_prepomax dictates what happens for the specific control index.

If both print_frd_prepomax and control_print_frd_prepomax are not specified the switch is set to -no.

6.968 print_gid_contact_spring2 number_of_nodes

Set number_of_nodes to 2 if you want to draw contact_spring2 with two nodes, and to 1 if you want to draw contact_spring2 with one node. Default, if print_gid_contact_spring2 is not specified, then 1 is used for number_of_nodes.

6.969 print_gid_coord switch

If switch is set to -yes the coordinates of nodes is plotted in gid.
6.970  print_gid_define switch

If switch is set to -yes defined names will be used i.s.o. geometry names (when a geometry is defined as a name with start_define ... end_define). If switch is set to -no that will not be done. Default, if switch is not specified, switch is set to -yes.

6.971  print_gid_group switch

If switch is set to -yes the element groups are plotted in gid as result field; in gid you can do a contour_fill to visualize this result field. If switch is set to -no the element groups are not plotted in gid as result. Default switch is set to -yes.

6.972  print_gid_mesh_activate_gravity switch

See also mesh_activate_gravity_time.

6.973  print_gid_node_method method

See print_gid_node_method.

6.974  print_gid_spring2 number_of_nodes

Set number_of_nodes to 2 if you want to draw spring2 with two nodes, and to 1 if you want to draw spring2 with one node. Default, if print_gid_spring2 is not specified, then 1 is used for number_of_nodes.

6.975  print_group_data dataitem_name_0 dataitem_name_1 ...

Print in the gid files group_* data items for isoparametric finite elements. As a typical example use -group_materi_elasti_young; then you get in the gid plot what the young model distribution is for isoparametric finite elements in the mesh.

All group data is averaged over each element, so you will see a constant value per element (even when the group data item may vary over the different integration points in an element).

For elements which do not have a specific group data item a value 0 will be plotted. Tochnog sets the gid group data information in the timesteps, so only after timesteps have been taken you will see meaningful results for the group data in gid plots.

The values will also be placed in the element_print_group_data records.

6.976  print_gmsh_calculation switch

If you set switch to -yes the gmsh files will be printed at the end of the calculation. If you set switch to -no the gmsh files will not be printed at the end of the calculation. Default, if switch is not specified, it is set to -no.
6.977 print_gmsh_dummy switch

See control_print_gmsh_dummy. This print_gmsh_dummy holds for all gmsh printing, unless it is overruled by a control_print_gmsh_dummy.

6.978 print_gmsh_node_method method

See print_gmsh_node_method.

6.979 print_mesh_dof dof_0 dof_1 ...

This option allows you to print results for dof’s (temperatures, groundflow pressures, ...) in a first calculation and use these results later in a second calculation as boundary conditions. This comes handy when you need to run the second calculation multiple times, and the results for the printed dof’s can be taken from the first calculation. In this way, the computing time of the second calculation can be smaller, and also a different FE mesh can be used in the first calculation and the second calculation for the different doffields.

In the first calculation you can print the dof’s with the command print_mesh_dof; the results will be printed in the file print_mesh_dof.txt. The dof_0 dof_1 ... of print_mesh_dof specify the dof’s which will be printed. In the first calculation printing of the dof’s to the file print_mesh_dof.txt will actually be done for when switch is set to -yes in control_print_mesh_dof.

For the second calculation rename the file print_mesh_dof.txt into bounda_mesh_dof.txt. You can specify which of the dof’s in the file bounda_mesh_dof.txt will actually be used a prescribed value ('boundary condition') with the dof_0 dof_1 ... of bounda_print_mesh_dof. You can restrict the nodes to which this will be done by bounda_print_mesh_dof_geometry (please realise using a geometry point with a very large tolerance in combination with geometry_element_group you can effectively select the geometry formed by an element group).

The FE meshes as used in the first calculation and in the second calculation need not be the same, and are also allowed to vary in time (in building processes, excavations, etc.). Nodes from the second mesh will be located in the first mesh, and doffields will be interpolated from the first mesh to the second mesh. In case a node from the second mesh is not inside any isoparametric element of the first mesh, the value for the dof’s as specified in the optional bounda_print_mesh_dof_values will be used. In bounda_print_mesh_dof_values you need to specify values for each and every dof that was specified with print_mesh_dof in the first calculation. If the node of the second mesh cannot be found in the first mesh and also bounda_print_mesh_dof_values is not specified then the dof’s will be taken from the closest node of the first mesh.

Results for the second mesh will be linearly interpolated in time from results of the first mesh.

Example first calculation in which only a temperature field is calculated:

```bash
echo -yes
number_of_space_dimensions 2
condif_temperature
end_initia
...
print_mesh_dof -temp
...
control_timestep 10 ...
control_print_mesh_dof 10 -yes (print in print_mesh_dof.txt)
```
Example second calculation in which the temperature field calculated in the first calculation is imposed:

```plaintext
  echo -yes
  number_of_space_dimensions 2
  condif_temperature
  materi_velocity
  materi_displacement
  materi_stress
  end_initia
  ...
  bounda_print_mesh_dof -temp
  bounda_print_mesh_dof_values 20. (read from bounda_mesh_dof.txt)
  ...
```

6.980  **print_node_geometry_present** switch

See `node_geometry_present`. See also `print_node_geometry_present_node_type`. Default switch is set to `-no`.

6.981  **print_node_geometry_present_node_type** node_type

If `node_type` is set to `-node_start_refined` the values of `-node_start_refined` are used. If `node_type` is set to `-node` the values of `-node` are used. If `node_type` is set to `-plus_displacement` the values of `-node` plus nodal displacements. Default `node_type` is set to `-node_start_refined`.

6.982  **print_precision** number_of_values

With `number_of_values` you can set for all printing how many values will be used at printing. For example, setting `number_of_values` to 4 the internal tochnog double `98.123456789` will be printed as `98.12` when using `control_print`, `control_print_gid` etc.

6.983  **print_tecplot_calculation** switch

If you set `switch` to `-yes` the tecplot files will be printed at the end of the calculation. If you set `switch` to `-no` the tecplot files will not be printed at the end of the calculation. Default, if `switch` is not specified, it is set to `-yes`.

6.984  **print vtk_calculation** switch

If you set `switch` to `-yes` the vtk files will be printed at the end of the calculation. If you set `switch` to `-no` the vtk files will not be printed at the end of the calculation. Default, if `switch` is not specified, it is set to `-yes`. 
6.985  **print_vtk_coord switch**

If *switch* is set to -yes the coordinates of nodes is plotted in vtk.

6.986  **print_vtk_group switch**

If *switch* is set to -yes the element groups are plotted in vtk as result field. If *switch* is set to -no the element groups are not plotted in vtk as result. Default *switch* is set to -yes.

6.987  **print_vtk_node_method method**

See print_vtk_node_method.

6.988  **print_where switch**

If *switch* is set to -yes, information will be printed about the taks that tochnog is performing (evaluation boundary conditions, loop over elements, etc.).

This is convenient for very large calculations, to see what is being done and how the calculation proceeds.

6.989  **processors nproc**

With this record you can set the number of shared memory CPUs you want to use (*nproc*). If your TOCHNOG implementation does not allow for more processors, this record is ignored. In fact, not the number of processors but the number of threads is set (that is, if you use 2 threads while your system only supports 1 processor than those threads are split over that single processor).

Error messages may become confusing when you use more than one processor.

Default *nproc* is 1.

6.990  **processors_maximum switch**

If *switch* is set to -yes, the processors record will be set to the maximum number as allowed by your computer.

Default *switch* is set to -yes. This processors_maximum record will not be used if the processors record is specified.

6.991  **processors_partition npartition**

The element loop is parallised as follows. The master process gives away small amounts of the total number of elements to child processes, and if a child process is ready it gets a new small amount of the master process. In fact, a child process gets each time an amount of \( \text{npartition} \times \text{processors} \) where \text{npartition} is the number of elements, \text{npartition} is specified in processors_partition, and processors is specified in processors. Default, if processors_partition is not specified, we set npartition to 1.
6.992 relaxation *relax_0 relax_1 ...*

Relaxation parameters for adjusting dof’s in iterations. This can stabilize the calculation. For example, a relaxation parameter of 0.1 means that the corresponding dof is now completely updated with the iterative change, but only 10 percent of the change is actually applied in a iteration.

If enough iterations are used, the relaxation parameters with not influence the final solution.

You should specify a relaxation parameter term for each principal dof which is present in the calculation (see the start of the data part description for a list of principal dof’s; these are velocities, temperature, etc.).

The relaxation is used for all timesteps. See also control_relaxation.

6.993 repeat_save_result *index result_0 result_1 ...*

See control_repeat_save. The index is the number of repetition (index 0 is repeat 0, index 1 is repeat 1, etc.)

6.994 repeat_save_calculate_result *average_0 variance_0 average_1 variance_1 ...*

See control_repeat_save_calculate.

6.995 safety_slip_circle_grid_middle *index x_first y_first x_last y_last*

This record specifies a grid with middles of a circle for safety factor calculations. With *x_first y_first* you specify the first middle. With *x_last y_last* you specify the last middle. With safety_slip_circle_grid_middle_n you specify the number of middles that should be evaluated in the safety calculation; all middles together form a equidistant grid between *x_first y_first* and *x_last y_last*.

As a special option you can only specify *x_first y_first* and not specify safety_slip_circle_grid_middle_n, then only one middle *x_first y_first* will be evaluated for the circle in the safety calculation.

See also control_safety_slip.

6.996 safety_slip_circle_grid_middle_n *index n*

See safety_slip_circle_grid_middle.

6.997 safety_slip_circle_grid_radius *index r_first r_last*

This record specifies the radius of a circle for safety factor calculations.

With *r_first* you specify the first radius. With *r_last* you specify the last radius. With safety_slip_circle_grid_radius_n you specify the number of radius that should be evaluated in the safety calculation; all radius to be evaluated will be put equidistant between *r_first* and *r_last*. 
As a special option you can only specify \textit{r\_first} and not specify \texttt{safety\_slip\_circle\_grid\_radius\_n}; then only one radius \textit{r\_first} will be evaluated for the circle in the safety calculation.

\textbf{6.998 \texttt{safety\_slip\_circle\_grid\_radius\_n} index \textit{n}}

See \texttt{safety\_slip\_circle\_grid\_radius}.

\textbf{6.999 \texttt{safety\_slip\_circle\_grid\_result} index \textit{x y r safety\_factor}}

This record will after the calculation be filled with the middle, radius and safety factor for the critical surface (for the slip circles with the same index).

\textbf{6.1000 \texttt{safety\_slip\_circle\_grid\_segment\_n} index \textit{n}}

With this record you can specify how many segments in the circle will be used in the integration of the safety factor. A high number of segments gives more accuracy but is time consuming. A low number of segments is less accurate but fast. Default, if \texttt{safety\_slip\_circle\_grid\_segment\_n} is not specified, then 90 segments will be used.

\textbf{6.1001 \texttt{safety\_slip\_circle\_line\_middle} index \textit{x\_first y\_first x\_last y\_last}}

This record specifies a line with middles of a circle for safety factor calculations. With \textit{x\_first y\_first} you specify the first middle. With \textit{x\_last y\_last} you specify the last middle. With \texttt{safety\_slip\_circle\_line\_middle\_n} you specify the number of middles that should be evaluated in the safety calculation; all middles together form a equidistant line between \textit{x\_first y\_first} and \textit{x\_last y\_last}.

As a special option you can only specify \textit{x\_first y\_first} and not specify \texttt{safety\_slip\_circle\_line\_middle\_n}; then only one middle \textit{x\_first y\_first} will be evaluated for the circle in the safety calculation.

See also \texttt{control\_safety\_slip}.

\textbf{6.1002 \texttt{safety\_slip\_circle\_line\_middle\_n} index \textit{n}}

See \texttt{safety\_slip\_circle\_line\_middle}.

\textbf{6.1003 \texttt{safety\_slip\_circle\_line\_radius} index \textit{r\_first r\_last}}

This record specifies the radius of a circle for safety factor calculations.

With \textit{r\_first} you specify the first radius. With \textit{r\_last} you specify the last radius. With \texttt{safety\_slip\_circle\_line\_radius\_n} you specify the number of radius that should be evaluated in the safety calculation; all radius to be evaluated will be put equidistant between \textit{r\_first} and \textit{r\_last}.

As a special option you can only specify \textit{r\_first} and not specify \texttt{safety\_slip\_circle\_line\_radius\_n}; then only one radius \textit{r\_first} will be evaluated for the circle in the safety calculation.
6.1004  safety_slip_circle_line_radius_n index n

See safety_slip_circle_line_radius.

6.1005  safety_slip_circle_line_result index x y r safety_factor

This record will after the calculation be filled with the middle, radius and safety factor for the critical surface (for the slip circles with the same index).

6.1006  safety_slip_circle_line_segment_n index n

With this record you can specify how many segments in the circle will be used in the integration of the safety factor. A high number of segments gives more accuracy but is time consuming. A low number of segments is less accurate but fast. Default, if safety_slip_circle_line_segment_n is not specified, then 90 segments will be used.

6.1007  safety_slip_combined_linear index x_first,0 y_first,0 x_first,1 y_first,1 
... x_last,0 y_last,0 x_last,1 y_last,1 ...

This record specifies combined linear lines along which the safety factor should be calculated.

All data with first specifies the first combined linear line. The x_first,0 y_first,0 specifies the first point of the first line piece of the first combined linear line, the x_first,1 y_first,1 specifies the second point of the first line piece of the first combined linear line The x_first,2 y_first,2 specifies the first point of the second line piece of the first combined linear line, the x_first,3 y_first,3 specifies the second point of the second line piece of the first combined linear line etc.

All data with last specifies the last combined linear line. The x_last,0 y_last,0 specifies the first point of the first line piece of the last combined linear line, the x_first,1 y_first,1 specifies the second point of the first line piece of the last combined linear line The x_last,2 y_last,2 specifies the first point of the second line piece of the last combined linear line, the x_first,3 y_first,3 specifies the second point of the second line piece of the last combined linear line etc. This last combined linear line should have an equal number of points as the first combined linear line.

With safety_slip_combined_linear_n you specify the number of combined linear lines that should be evaluated in the safety calculation; all combined linear lines to be evaluated will be put equidistant between the first combined linear line and the second combined linear line.

As a special option you can only specify data for the first combined linear line, and specify not data for the last combined linear line and not safety_slip_combined_linear_n; then only one combined linear line will be used.

See also control_safety_slip.

6.1008  safety_slip_combined_linear_n index n

See safety_slip_combined_linear.
6.1009  safety_slip_combined_linear_result index x_0 y_0 x_1 y_1 \ldots safety_factor

This record will after the calculation be filled with the combined linear line for the critical surface (for the combined linear lines circles with the same index).

6.1010  safety_slip_combined_linear_segment_n index n

With this record you can specify how many segments in a line piece of a combined linear line will be used in the integration of the safety factor. A high number of segments gives more accuracy but is time consuming. A low number of segments is less accurate but fast. Default, if safety_slip_combined_linear_segment_n is not specified, then 10 segments will be used.

6.1011  safety_slip_ellipsoide index middle_x_first middle_y_first middle_z_first

base1_x_first base1_y_first base1_z_first base2_x_first base2_y_first

base2_z_first a_first b_first c_first middle_x_last middle_y_last middle_z_last

base1_x_last base1_y_last base1_z_last base2_x_last base2_y_last

base2_z_last a_last b_last c_last

This record specifies a 3D ellipsoide for which the safety factor should be calculated. The ellipsoide equation is \[ \frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1 \], where \( x, y \) and \( z \) are local coordinates in the ellipsoide. The ellipsoide is specified by 12 parameters in tochnog.

All parameters with first specifies the first ellipsoide. The middle_x_first middle_y_first middle_z_first specifies the ellipsoide middle (for which the local coordinates are 0; \( x = 0, y = 0, z = 0 \)). The base1_x_first base1_y_first base1_z_first specifies the direction of the local \( x \) axes in space. The base2_x_first base2_y_first base2_z_first specifies the direction of the local \( y \) axes in space. Tochnog determines automatically the direction of the local \( z \) axes in space. The \( a b c \) specifies the radii in respective the local \( x, y \) and \( z \) direction.

All parameters with last specifies the last ellipsoide.

With safety_slip_ellipsoide_n you specify the number of variations that should be used for each of the specified ellipsoids parameters. All parameters will be interpolated between the values specified for the first and last ellipsoide. In case you want to keep a parameter fixed, thus it should not be varied, simply specify an equal value for the parameter in the first and last ellipsoide.

As a special option you can only specify parameters for the first ellipsoide, and specify not parameters for the last ellipsoide.

See also control_safety_slip.

6.1012  safety_slip_ellipsoide_method index method

The normal on the ellipsoide surface is uniquely defined, so that the normal stresses are uniquely defined. The slip direction in the surface is not uniquely defined however. Below several possibilities are listed.

If method is set to -safety_slip_ellipsoide, then the ellipsoide local \( x \) direction will be used as slip direction (to be more precise, the projections on the ellipsoide surface will be used everywhere).

If method is set to -materi_displacement or -materi_velocity_integrated, then the last calculated displacements will be used as slip direction (to be more precise, the projections on the
ellipsoid surface will be used everywhere).

If *method* is set to `-materi_velocity`, then the last calculated velocities will be used as slip direction (to be more precise, the projections on the ellipsoid surface will be used everywhere).

Default, if `safety_slip_ellipsoide_method` is not specified, *method* is set to `-safety_slip_ellipsoide`.

### 6.1013 safety_slip_ellipsoide_n index n

See `safety_slip_ellipsoide`.

### 6.1014 safety_slip_ellipsoide_result index middle_x middle_y middle_z base1_x base1_y base1_z base2_x base2_y base2_z a b c safety_factor

This record will after the calculation be filled with the ellipsoid for the critical surface and the safety factor.

### 6.1015 safety_slip_ellipsoide_segment_n index n

With this record you can specify how many segments in an ellipsoid will be used in the integration of the safety factor. The ellipsoid is internally in technotg integrated in a local \( \phi \) and \( \theta \) direction, over `safety_slip_combined_linear_segment_n` segments each. A high number of segments gives more accuracy but is time consuming. A low number of segments is less accurate but fast. Default, if `safety_slip_ellipsoide_n` is not specified, then 90 segments will be used.

### 6.1016 safety_slip_grd index switch

If *switch* is set to `-yes`, Tochnog will read a slip surface from the file `index.grd`. The file is in .grd format, as used by the `surfer` program from Golden Software. Thus the format is:

```
DSAA
nx ny
xmin xmax
ymin ymax
zmin zmax
...(for first y specify z values for all x)
...(for second y specify z values for all x)
...
```

This `safety_slip_grd` is only available in 3D.

### 6.1017 safety_slip_grd_method index method

With this record you can specify with which method the slip direction is chosen (this is the direction in which the slip shear force will be determined, to calculate the safety factor).
If `method` is set to `-safety_slip_grd_direction` the direction specified in `-safety_slip_grd_direction` will be used. If `method` is set to `-materi_velocity` the last calculated `-materi_velocity` directions will be used. If `method` is set to `-materi_displacement` the last calculated `-materi_displacement` directions will be used. If `method` is set to `-materi_velocity_integrated` the last calculated `-materi_velocity_integrated` directions will be used. If somewhere the direction is not specified by the above, because the used direction is a null vector, then Tochnog will ask you to specify `safety_slip_grd_method_direction` additionally, and then that direction will be used there.

Default, if `safety_slip_grd_method` is not specified, `method` is set to `-safety_slip_grd_direction`.

### 6.1018 safety_slip_grd_method_direction

Index: dir_x dir_y dir_z

See `safety_slip_grd_method`.

### 6.1019 safety_slip_grd_segment_n

Index: n

With this record you can specify how many segments in each part of the surface of the grd file will be used in the integration of the safety factor. In total the surface has `nx*ny` parts; each of these parts will be integrated with `n*n` segments. Default, if `safety_slip_grd_segment_n` is not specified, then `n` will be set to 10.

### 6.1020 safety_slip_multi_linear

Index: x_first,0 y_first,0 x_first,1 y_first,1

This record specifies multi linear lines along which the safety factor should be calculated.

All data with `first` specifies the first multi linear line. The `x_first,0 y_first,0` specifies the first point of the first line piece of the first multi linear line, the `x_first,1 y_first,1` specifies the second point of the first line piece of the first multi linear line which is also the first point of the second line piece of the first multi linear line, etc.

All data with `last` specifies the last multi linear line. The `x_last,0 y_last,0` specifies the first point of the first line piece of the last multi linear line, the `x_first,1 y_first,1` specifies the second point of the first line piece of the last multi linear line which is also the first point of the second line piece of the last multi linear line, etc. This last multi linear line should have an equal number of points as the first multi linear line.

With `safety_slip_multi_linear_n` you specify the number of multi linear lines that should be evaluated in the safety calculation; all multi linear lines to be evaluated will be put equidistant between the first multi linear line and the second multi linear line.

As a special option you can only specify data for the first multi linear line, and specify not data for the last multi linear line and not `safety_slip_multi_linear_n`; then only one multi linear line will be used.

See also `control_safety_slip`.

### 6.1021 safety_slip_multi_linear_n

Index: n

See `safety_slip_multi_linear`.
6.1022 safety_slip_multi_linear_result index x_0 y_0 x_1 y_1 ... safety_factor

This record will after the calculation be filled with the multi linear line for the critical surface (for the multi linear lines circles with the same index).

6.1023 safety_slip_multi_linear_segment_n index n

With this record you can specify how many segments in a line piece of a multi linear line will be used in the integration of the safety factor. A high number of segments gives more accuracy but is time consuming. A low number of segments is less accurate but fast. Default, if safety_slip_multi_linear_segment_n is not specified, then 10 segments will be used.

6.1024 safety_slip_set index index_0 index_1 index_1 ...

This records defines the indices of safety geometries belong to a set. For all safety geometries of a set, the minimal safety factor will be determined.

As a special option you can also define a range.

6.1025 safety_slip_set_result index index safety_factor

This record will be filled after the calculation with the minimal safety factor of the geometries in the set.

6.1026 slide_geometry index geometry_entity geometry_entity_index

This record generates slide friction forces when a material slides over the geometry specified by geometry_entity geometry_entity_index.

This option comes handy when it is a priori known at which nodes sliding will occur, which is typically the case in an Eulerian calculation.

Also slide_plasti_friction should be specified.

See also node_slide.

6.1027 slide_plasti_friction index phi c

This record specifies friction for the slide_geometry option. The maximum friction force between the material and the side surface equals $c + F_n \cdot \tan(\phi)$ where $c$ is the cohesion, $\phi$ is the friction angle in radians and $F_n$ is the normal force.

6.1028 slide_plasti_tension index sig_t

This record specifies maximum tensile force for the slide_geometry option.
6.1029  **slide_user**  *index switch*

If *switch* is set to *-yes* the user supplied routine for slide friction is called.

See also the file **user.cpp** in the distribution.

6.1030  **slide_damping**  *index damping_n damping_t*

This specifies the normal damping and tangential damping for sliding. See also **control_slide_damping_apply**.

6.1031  **slide_stiffness**  *index stiffness_n stiffness_t*

This specifies the normal stiffness and tangential stiffness for sliding. See also **control_slide_stiffness_apply**.

6.1032  **solver**  *solver_type*

You can set here the solver type to one of solvers as specified in **control_solver**. The solver set here holds for the entire calculation (as opposed to the **control_solver** which only holds for the corresponding time steps). In fact, each **control_solver** will be overwritten by a specified **solver**.

When using the bicg solver, consider also setting **solver_matrix_symmetric** to *-yes*, in order to speed up the speed of the bicg solver.

6.1033  **solver_bicg_error**  *error*

With *error* you set the termination error ratio between the initial and final error in the bicg iterations.

See also **control_solver_bicg_error**.

6.1034  **solver_bicg_restart**  *nrestart*

With *nrestart* you set the number of restarts in the bicg iterations.

See also **control_solver_bicg_restart**.

6.1035  **solver_bicg_stop**  *switch*

If *switch* is set to *-yes*, the calculation is stopped if the bicg solver does not converge. If *switch* is set to *-no*, the calculation is not stopped if the bicg solver does not converge.

See also **control_solver_bicg_stop**.

6.1036  **solver_matrix_save**  *switch*

If *switch* is set to *-yes*, the solver saves and applies the decomposed matrix, but not in case Tochnog thinks for some reason that the matrix needs to be decomposed at each timestep. This
can save CPU time, since further decompositions of the matrix are not required anymore (only backsubstitution to find the solution vector).

If switch is set to -no, the solver does not save the decomposed matrix.

If switch is set to -always, the solver saves and applies the decomposed matrix, even in case Tochnog thinks for some reason that the matrix needs to be decomposed at each timestep.

This option is only available in combination with the pardiso solver. See also control_solver_matrix_save.

6.1037 solver_matrix_symmetric switch

If switch is set to -yes then, if needed, matrices are symmetrized so that less memory will be needed and a symmetrical equation solver can be used.

6.1038 solver_pardiso_ordering ordering

See also control_solver_pardiso_ordering.

6.1039 solver_pardiso_out_of_core switch

If switch is set to -yes the pardiso solver is called with a 'out of core' option. See for further the pardiso solver in the the intel mkl library.

6.1040 solver_pardiso_processors nproc

Set the number of processors to be used by the pardiso solver. Only 1 or the maximum number of the computer is allowed, nothing in between.

To maximise parallel performance of the pardiso solver set the environment symbol MKL_DYNAMIC to FALSE.

6.1041 solver_pardiso_processors_maximum switch

If switch is set to -yes the maximum number of processors will be used for the pardiso solver. Else only 1 processor will be used for the pardiso solver.

6.1042 strain_settlement_parameters index time_global,start time_plus reference_creep_strain_rate reference_time power_n lateral_factor

With this option data items you can specify an extra vertical settlement creep strain. Think of geotechnics soil dumping as a typical example where after dumping some extra vertical straining shows up over time.

The vertical settlement creep strain of a soil particle is assumed to be:

\[ \dot{\epsilon}_{zz} = \frac{\dot{\epsilon}_r}{1 + \left(\frac{t - t_{plus}}{t_r}\right)^n} \]
The user supplied parameters are: \( \dot{\epsilon}_r \) as reference creep rate, \( t_{\text{plus}} \), \( t_r \) as reference time and \( n \) as power constant. Creep strain starts when the global time \( t_{\text{global}} \) reaches \( t_{\text{global, start}} \). So \( \text{time}_{\text{age, start}} \) can be used to set where in the creep strain curve the material will start with creeping. The time \( t \) in this equation is the time elapsed after the material has become active (so the time after dumping the material, which typically is different for each finite element).

The horizontal creep strains \( \dot{\epsilon}_{xx} = f \dot{\epsilon}_{zz} \) and \( \dot{\epsilon}_{yy} = f \dot{\epsilon}_{zz} \) are assumed to be a lateral factor \( f \) times the vertical creep strain.

This \texttt{strain_settlement_parameters} should be combined with the \texttt{mesh_gravity_activate_time} option as follows:

```
...                               mesh_activate_gravity_time 10 ...
    mesh_activate_gravity_time_strain_settlement 10 -yes
    strain_settlement_parameters 20 ...
...                                    
```

The \texttt{mesh_activate_gravity_time_strain_settlement} indicates that the mesh activation should not be used by itself, but is only used to determine element activation times needed for the \texttt{strain_settlement_parameters} option.

See also \texttt{strain_settlement_element_group}.

\textbf{6.1043 strain_settlement_element_group index element_group_0 element_group_1 ...}

This record specifies the element groups for which the \texttt{strain_settlement_parameters} with the same parameters will be used. As a special option you can use \texttt{-all}, such that all elements groups will be used.

\textbf{6.1044 strain_volume_absolute_time index time_0 volume_increase_absolute_0 time_1 volume_increase_absolute_1 ...}

See \texttt{strain_volume_element}.

\textbf{6.1045 strain_volume_element index element_0 element_1 ...}

With the \texttt{strain_volume_*} data items you can specify an extra volumetric strain component which Tochnog should add to specified elements, element groups or a geometry. Think of geotechnics grouting as a typical example.

If you specify \texttt{strain_volume_*} then \( \frac{1}{3} \) of the specified volumetric strain is imposed as \( xx \)-strain, \( yy \)-strain and \( zz \)-strain. The actual straining of elements in the calculation will depend on boundary conditions, external forces, etc. For example, if you have a calculation where all displacements are suppressed by boundary conditions, the actual deformation of elements is null, and the specified \texttt{strain_volume_*} only leads to stresses. For example, if you have a 3d calculation where elements are free to deform the actual deformation of elements will become equal to the specified \texttt{strain_volume_*}. For example, if you have a 2d plain strain calculation where elements are free
to deform the actual deformation of elements will be smaller than specified \texttt{strain\_volume\_} (since there is no deformation possible in the third direction).

Use \texttt{strain\_volume\_element} to specify element numbers for which the volume strain should be applied. Use \texttt{strain\_volume\_element\_group} to specify element group numbers for which the volume strain should be applied. Use \texttt{strain\_volume\_geometry} to specify a geometry for which the volume strain should be applied.

You can either specify relative volume strains (relative to the initial volume) or absolute volume changes. Use \texttt{strain\_volume\_relative\_time} to specify a time versus relative volume strain diagram. These relative volume strains should be specified as percentage (thus, 100 would be a volume strain equal to the initial volume, so 100 percent extra volume). Use \texttt{strain\_volume\_absolute\_time} to specify a time versus absolute volume increase diagram. These absolute volume increases should be specified as real volume (thus \( m^3 \) if you use "meter" as length unit in your input file). Exactly one of \texttt{strain\_volume\_relative\_time} or \texttt{strain\_volume\_absolute\_time} should be given in the input file, not both. At times outside \texttt{strain\_volume\_relative\_time}, or \texttt{strain\_volume\_absolute\_time}, the relative volume strain, or absolute volume increase, are assumed to be zero.

If none of \texttt{strain\_volume\_element}, \texttt{strain\_volume\_element\_group} or \texttt{strain\_volume\_geometry} is given then the \texttt{strain\_volume\_relative\_time} or \texttt{strain\_volume\_absolute\_time} will be applied to all isoparametric elements.

The volumetric strain option presently is available only for small deformation analysis. The volumetric strain can be be applied to isoparametric elements only. The volumetric strain is not available for membrane elements.

See also \texttt{post\_strain\_volume\_absolute} and \texttt{post\_strain\_volume\_relative}.

\texttt{6.1046 strain\_volume\_element\_group index element\_group\_0 element\_group\_1 ...}

See \texttt{strain\_volume\_element}.

\texttt{6.1047 strain\_volume\_geometry index geometry\_item\_name geometry\_item\_index}

See \texttt{strain\_volume\_element}.

\texttt{6.1048 strain\_volume\_relative\_time index time\_0 relative\_volume\_strain\_0 time\_1 relative\_volume\_strain\_1 ...}

See \texttt{strain\_volume\_element}.

\texttt{6.1049 support\_edge\_normal index stiffness\_normal stiffness\_tangential}

Distributed support of an edge (winkler foundation). The \texttt{stiffness\_normal} specifies the normal stiffness of the support per unit length in 2D, and per unit area in 3D. The \texttt{stiffness\_tangential} specifies the tangential stiffness. This option is meant for 2D and 3D calculations.

Also the record \texttt{support\_edge\_normal\_geometry} should be specified.

\textbf{Attention:} this option is only available for linear and quadratic isoparametric elements.
6.1050  support_edge_normal_damping index damping_normal damping_tangential

Distributed damping at an edge. The damping_normal specifies the normal damping of the support per unit length in 2D, and per unit area in 3D. The damping_tangential specifies the tangential damping. This option is meant for 2D and 3D calculations.

If you want to use support_edge_normal_damping to absorb stress wave at the boundaries of the mesh (think of vibration or earthquake analysis), there are typical values to be used for the normal and tangential damping.

For absorbing boundaries the damping_normal typically should be set to \( C_n \rho V_n \). The parameter \( C_n \) typically is chosen as 1. The \( \rho \) is the material density. The pressure wave velocity is \( V_n = \sqrt{\frac{E_{oed}}{\rho}} \) with oedometric stiffness \( E_{oed} = \frac{(1-\nu)E}{(1+\nu)(1-2\nu)} \) where \( E \) is the Young’s modulus, and \( \nu \) is Poisson’s ratio. For absorbing boundaries the damping_tangential typically should be set to \( C_t \rho V_t \). The parameter \( C_t \) typically is chosen as 0.25. The shear wave velocity is \( V_t = \sqrt{\frac{G}{\rho}} \) with shear modulus \( G = \frac{E}{2(1+\nu)} \).

Also the records support_edge_normal and support_edge_normal_geometry should be specified.

Attention: this option is only available for linear and quadratic isoparametric elements.

See also control_support_edge_normal_stiffness_freeze and node_support_edge_normal_force.

See also support_edge_normal_damping for automatic specification of damping properties.

6.1051  support_edge_normal_damping_automatic index switch

If you set switch to -yes this record generates damping just like the support_edge_normal_damping record. Now however you do not need to specify the damping properties yourself; they are calculated by Tochnog using the Young value \( E \) and the Poisson ratio \( \nu \) from the isoparametric element attached to the support.

6.1052  support_edge_normal_damping_automatic_apparent index switch

If you set switch to -yes this record generates damping just like the support_edge_normal_damping record. Now however you do not need to specify the damping properties yourself; they are calculated by Tochnog using the apparent young and poisson value from the isoparametric element attached to the support.

6.1053  support_edge_normal_density index density_normal density_tangential

Distributed density at an edge. The density_normal specifies the normal density of the support per unit length in 2D, and per unit area in 3D. The density_tangential specifies the tangential density. This option is meant for 2D and 3D calculations.

Attention: this option is only available for linear and quadratic isoparametric elements.
6.1054 support_edge_normal_element_node index element_0 element_1 ...

Selects the element and local nodes for which the support_edge_normal record with the same index should be applied.

6.1055 support_edge_normal_element_group index element_group

Restricts the element group to which the support_edge_normal record with the same index should be applied.

6.1056 support_edge_normal_element_side index element_0 element_1 ...
... side

Selects the elements and local side number for which the support_edge_normal record with the same index should be applied.

6.1057 support_edge_normal_factor index a_0 a_1 ... a_n

The same as force_edge_normal_factor, now however for the support stiffnesses (and not the force).

6.1058 support_edge_normal_force_initial index a_0 a_1

This record allows you to specify an initial normal force in the support, linear varying in depth direction. The initial normal force actually is \( a_0 + a_1 \times y \) in 2D, or \( a_0 + a_1 \times z \) in 3D.

6.1059 support_edge_normal_geometry index geometry_entity_name geometry_entity_index

Selects the area for which the support_edge_normal record with the same index should be applied. For example, -geometry_line 1 can be used in 2D, indicating that the nodes on line 1 get the distributed support.

6.1060 support_edge_normal_node index node_0 node_1 node_2 ...

Selects the nodes for which the support_edge_normal record with the same index should be applied. The node_0 etc. specify global node numbers.

6.1061 support_edge_normal_plasti_compression index normal_force_minimum tangential_force_factor

With normal_force_minimum you can limit the amount of compression force that a support can take. Any compression force lower than this normal_force_minimum will actually be set to normal_force_minimum. Typically you want to specify a negative value for index normal_force_minimum.
With *tangential_force_factor* you can model frictional slip in the tangential direction. The tangential force is limited to *tangential_force_factor* times the normal force. Larger tangential forces are not allowed.

This **support_edge_normal_plasti_compression** will only be used if the normal force does not exceed the maximum tension force as specified in **support_edge_normal_plasti_tension** or **support_edge_normal_plasti_tension_double**.

All forces are per unit length in 2D, and per unit area in 3D.

6.1062 **support_edge_normal_plasti_friction** *index cohesion friction_coefficient*

With this record you can limit the amount of friction force that a support can take. The maximum allowed friction force is the *cohesion* plus the *friction_coefficient* multiplied with the absolute value of the normal force.

All forces are per unit length in 2D, and per unit area in 3D.

6.1063 **support_edge_normal_plasti_tension** *index switch*

If *switch* is set to *-yes* and the normal force in the support is tension, then all forces are set to 0. This models gap building between the support and the element edge.

6.1064 **support_edge_normal_plasti_tension_double** *index normal_force_maximum*

With *normal_force_maximum* you can limit the amount of tension force that a support can take. As opposed to *support_edge_normal_plasti_tension*, you can specify a non-zero value with this option. If a normal force higher than this *normal_force_maximum* occurs it will be set to *normal_force_maximum*, and tangential shear forces will be set to zero. Typically you want to specify zero or a positive value for *index normal_force_maximum*, although a negative value is also allowed.

Not both of **support_edge_normal_plasti_tension** and **support_edge_normal_plasti_tension_double** can be specified.

All forces are per unit length in 2D, and per unit area in 3D.

6.1065 **support_edge_normal_plasti_residual_stiffness** *index factor*

In case of plasticity in a support you can require that Tochnog includes a part of the original elastic stiffness in the element stiffness matrix to get more stable iterations. The part of the original stiffness included needs to be specified with *factor*, between 0 and 1. The stiffness is only included in the matrix, and not in the right-hand-side; so it will only influence convergence behaviour, but not the final results if a sufficient amount of steps is taken. Default, if **support_edge_normal_plasti_residual_stiffness** is not specified, *factor* is set to 0.

6.1066 **support_edge_normal_time** *index time load time load …*

This record specifies a diagram with a multiplication factor for the support edge force. Linear interpolation is used to extend the *time load* values to the intervals between these pairs. Outside
the specified time range a factor 0 is used.

If this record is not specified, a factor of 1 is applied at all times.

6.1067  **target_item**  index  data_item_name  data_item_index  number

See also: **target_value**.

6.1068  **target_value**  index  value  tolerance

This allows for testing the results of the calculation. Typically, `data_item_name` is `-node_dof` but also other data items can be tested. The record with index `data_item_index` will be tested. If `data_item_name` is `-node_dof` then `number` can be `-velx, -temp, etc. (see dof_label)`; else, for example, `number` is 3 states that the fourth value needs to be checked. The result should not differ more from `value` than `tolerance`.

For a calculation with no problems, the tochnog.log file contains a line stating that the calculation did start followed by a line stating that the calculation did end. If this is not precisely the case, some problem did occur or the targeted results differ too much. In the example below it is checked that the pressure in node 6 does not differ more than 1. $10^{-5}$.

```
.target_item 0 -node_dof 6 -pres
.target_value 0 1.2 1.e-5
```

The checked value, 1.2 in this case, has been found from a previous computation that is regarded as reliable. The present computed value is compared with the earlier one. If they agree within the specified tolerance, 1.e-5 in this case, then Tochnog is silent. If they do not, then Tochnog writes an error message into the file "tochnog.log".

6.1069  **time_calculation**  elapsed_time_in_seconds

Elapsed computer time up to moment of printing (wall clock time).

6.1070  **time_current**  current_time

Current time in calculation.

6.1071  **timestep_predict_velocity**  switch

Normally tochnog will use as prediction for velocities in a timestep the previous calculated velocities from the previous timestep.

However, if there is no inertia, and `convection_apply` is `-no` tochnog will use as prediction for velocities in a timestep a zero velocity.

You can require that tochnog does the normal prediction from the previous timestep however by setting `switch` to `-yes`; you typically want to do that in eulerian calculations.
6.1072  timestep_iterations_automatic_apply switch

If switch is set to -no any control_timestep_iterations_automatic records will be neglected.

6.1073  tochnog_version index day month year

This record contains the build day, the build month and the build year.

6.1074  truss_rope_apply switch

If switch is set to -no, any truss rope data in the input file will be ignored. This is done for all timesteps.

This option is convenient for testing your input file just linear, without the need to outcomment each and every part with truss rope data. See also control_truss_rope_apply.

6.1075  volume_factor $a_0, a_1 \ldots a_n$

This data item defines a polynomial in space in 1D or 2D. The polynomial specifies the cross-sectional area (in 1D) or the thickness (2D) as function of the global $x$ coordinate (1D) or the global $x,y$ coordinates (2D). For example, in a 1D solid calculation it can be used to specify varying cross-sectional areas of bars, or in a 1D flow calculation it can be used to specify the cross-sectional area of a channel.

In 1D the polynomial is $a_0 + a_1 x$ (specify 2 values). In 2D the polynomial is $a_0 + a_1 x + a_2 y$ (specify 3 values).

If this record is not specified, the cross-sectional area is 1 (1D) or the thickness is 1 (2D).

See also volume_element_factor.

6.1076  volume_factor_x $x_0 \ fac_{01} \ x_1 \ fac_{12} \ldots x_n$

This specifies an in x-direction changing volume factor for elements. Left from $x_0$ the factor is 1. From $x_0$ to $x_1$ the factor is $fac_{01}$. Etcetera. And right from $x_n$ the factor is 1 again.

6.1077  zip switch

If switch is set to -yes all *flavia*, *msh, vtk, *.plt and *dbs files are zipped with the gzip program. The gzip program should be installed on your computer.

This comes convenient in large calculation with lots of output, where you want to use results later and save disk space during the calculation.

If also control_zip is specified for a certain control index that overrules this zip for that control index.

6.1078  end_data (last record of data part)
7 Runtime file

You can use a runtime file to give Tochnog data on the fly (while it is running). The runtime file will be read at the start of each time step. The runtime file needs to have the same name as the input file, with the extension run instead of dat however. Suppose the name of the normal input file is beam.dat, then the name of the runtime file is beam.run. The runtime file always needs to be ended with two end_data statements.

As a typical example you can use this runtime file when you are doing a long calculation and you decide while the calculation is running that you want extra output. Suppose the normal input file tochnog.dat contains:

```plaintext
... control_timestep 100 ... 
... 
```

Then you can decide to get some extra GID plotting files, while Tochnog is already running, by using the runtime file tochnog.run with:

```plaintext
control_print_gid 100 -yes 
end_data end_data 
```

When you want to de-activate the printing of GID files again then set the runtime file to:

```plaintext
control_print_gid 100 -no 
end_data end_data 
```

As a special option, you can use exit_tochnog -yes in the runtime file; then Tochnog will exit the calculation after printing the complete database and GID files.

After the runtime file is read, it will be automatically deleted by Tochnog.
8 Interaction analyzes and advanced analyzes

8.1 Fluid-structure interaction

If a solid construction interacts with a fluid, both the solid and fluid can be modeled with the materi equation. Interaction forces between solid and fluid will automatically be generated. If required, a temperature field may be imposed. An example of an input file is given below:

```
... materi_velocity materi_stress condif_temperature end_initia ...
...
    element_group -ra -from 0 -to 100 -ra 1
    element_group -ra -from 101 -to 200 -ra 2 ...
    type 1 -materi -condif
    group_materi_elasti_young 1 ...
    group_materi_memory -updated
    group_condif_conductivity 3 ...
    ...
    type 2 -materi
    group_materi_elasti_compressibility 2 ...
    group_materi_viscosity 2 ...
    group_materi_memory -updated_linear
    group_condif_conductivity 2 ...
    ...
```

Elements 0-100 are solids (with temperature) and elements 101-200 are fluids (with temperature).

8.2 Consolidation analysis: ground water flow in deforming solid

The ground water flow equation can be combined with the materi equations. The solid will deform due to the ground water flow pressure gradient and ground water flow pressure will change due to solid volume changes. An example of an input file is given below:

```
... materi_velocity materi_stress groundflow_pressure end_initia ...
...
    groundflow_consolidation_apply -yes ...
    groundflow_density ...
    groundflow_phreatic_level ...
...
    group_type 0 -materi -groundflow
    group_materi_elasti_young 0 ...
```
The stresses as initialized by `materi_stress` are effective stresses. Internally the program calculates with total stresses (effective stress + total pressure) in the material equilibrium equation. You can obtain the total stresses for postprocessing by means of the `post_calcul` option.

To account for the gravitational stresses, use a density $\rho_{\text{sat}}$ in the `group_materi_density` record. Here $\rho_{\text{sat}}$ is the saturated density of the groundwater-soil mixture (mass of soil + water per unit volume of the soil-water mixture). Also specify the gravitation in the `force_gravity` record and, if required, also the `force_gravity_time` record to apply the gravitation slowly.

### 8.3 Heat transport in ground water flow

Heat transport in a ground water flow can be analyzed by combining the convection and diffusion of heat equation with the groundwater flow equation. Now the velocity in the convection and diffusion of heat equation is taken from the groundwater velocity field ($\beta_i = v_i^g$) if `groundflow_velocity` is initialized. An example of a input file is given below

```plaintext
... 
  groundflow_pressure 
  groundflow_velocity 
  condif_temperature 
  end_initia 
... 
  type 0 -groundflow -condif 
  group_groundflow_compressibility 0 ... 
  group_condif_conductivity 0 ... 
... 
```

If both `materi_velocity` and `groundflow_velocity` are initialized, $\beta_i = v_i + v_i^g$.

### 8.4 Heat transport in materials

Heat transport in a material can be analyzed by combining the convection and diffusion of heat equation with the materi equations. In this way thermal stresses or heat induced convection can be analyzed. Now the velocity in the convection and diffusion of heat equation is taken from the velocity field ($\beta_i = v_i$). An example of a input file is given below

```plaintext
... 
  materi_velocity 
  materi_stress 
  condif_temperature 
  end_initia 
... 
  type 0 -materi -condif 
```
8.5 Restart a calculation

You can use a dbs file to restart a calculation. In fact, a dbs file is an input file itself. It contains the record icontrol which contains the last control index actually performed with the previous calculation. You can add more control_* records and start the file again; it will then continue with these new control_* records.

You cannot use dbs files with contain control_repeat for restarting a calculation.
9 Final topics (input trouble, save memory /CPU time, ...)

9.1 Environment symbols

Records with a length of 1, and no index, you can also set via an environment symbol. You need to use capital characters in doing so. Typical examples are

- PROCESSORS 4
- PRINT_GID_CALCULATION -no
- PRINT_GMSH_CALCULATION -yes
- PRINT_NODE_GEOMETRY_PRESENT -yes

In windows set environment symbols in your advanced system settings. In a linux bash shell set environment symbols in your .bashrc file (eg export PROCESSORS=4).

9.2 Checking your geometry_* records

Set print_node_geometry_present -yes and set print_element_geometry_present -yes. Then look with gmsh if the geometries are like you want.

9.3 Continuing an analysis

- Copy the database from the previous calculation to a new file, e.g. new.dat.
- Run a new calculation with new.dat.

This can also be done with a database that is written as intermediate database in a previous calculation, for example directly after gravity. See also icontrol.

9.4 Use -node as geometry entity.

As a special option you can use a node as a geometrical entity. For example the following imposes a boundary condition on the temperature of node 6:

bounda_dof 10 -node 6 -temp

Notice that -node 6 is used in the format of a geometry entity.

9.5 Use -geometry_list as geometry entity.

As a special option you can use a list as a geometrical entity. For example the following imposes a boundary condition on the nodes of geometry list 6:
9.6 List input files with options

You can search for input files in your distribution which contain multiple words. For example change to the test/other directory. Suppose you want to see in which files you can see transient consolidation in a deforming soil.

In linux use the following command to list input file:
```
grep -il materi_velocity *.dat | xargs grep -il groundflow_capacity | xargs grep -il groundflow_consolidation.
```

In MS Windows use:
```
winodws explorer - Search - Advanced options - File contents
and search for
materi_velocity AND groundflow_capacity AND groundflow_consolidation.
```

9.7 Geometrically linear material

Either do this:

- Initialise `-materi_velocity` and `-materi_displacement`
- Use `-total_linear` for the material.

or do this:

- Initialise `-materi_velocity` and `-materi_velocity_integrated`
- Use `-fixed_in_space` for mesh
- Use `-updated_linear` for the material.

9.8 Dynamic calculations

Dynamic calculations are triggered by setting `inertia_apply -yes`. Take care that you have specified all required data, like material density, etc.

In case you want to reduce numerical damping, you can use the following piece of input file, notice the `materi_dynamic`.

```
... 
materi_displacement
materi_velocity
end_initia
... 
inertia_apply -yes
materi_dynamic -yes
```
... control_timestep ......
...

For heavy non-linear calculations this may converge worse than the default euler backward time integration.

To get damping similar to rayleigh damping in structural dynamics use:

... group_materi_damping ... (similar to rayleigh damping mass term, use rayleigh alpha * material density )
    group_materi_viscosity ... (similar to rayleigh damping stiffness term, use rayleigh beta * material young )
...

See also materi_dynamic to influence numerical damping in dynamic calculations with solid materials.

9.9 Input file syntax

- If you don’t understand the syntax of an option, please look in the tochnog/test/other directory for example files. Under linux search for the command, eg grep control_print_filter *.dat to get example files with control_print_filter.

9.10 Check large calculations

- Set both solver -none and linear_calculation_apply -yes ; run and check in gid the boundary conditions, forces, change of element groups, etc. In a complex model you can check geometries that you use by imposing an artificial boundary on them, eg bounda_dof ... -temp with value 1, and look in gid if you see that boundary condition showing up at the correct nodes.

- Only set linear_calculation_apply -yes ; run and check linear solution fields.

- Do not set anything special ; run and check solution fields.

9.11 Diverging calculations

- Try the linear elements -bar2, -quad4, -tria3, -hex8 and -tet4 in stead of quadratic elements.

- Try solver_matrix_save -no (always setup new system matrix)

- Try group_materi_plasti_mohr_coul_direct i.s.o. group_materi_plasti_mohr_coul

- Try small fixed timesteps (do not use automatic time stepping).

- Try more iterations with control_timestep_iterations.

- Try a lower interface stiffness.
• Try higher water capacity in calculation with consolidation (so water less stiff, anyway not too stiff relative to soil).

• Set `group_interface_materi_residual_stiffness` to 1.

9.12 Saving CPU time

• Check that the computer doesn’t swap to disk (use top in linux, use task manager in windows). In case of swapping lower the memory usage (see the section ‘Saving computer memory’).

• For large calculations with many `post_*` commands: use `post_apply -no` and use `control_post_apply index -yes` only at the moment that you actually need post results.

9.13 Saving computer memory

Try the following steps, in order of priority:

• `solver_matrix_symmetric -yes`.

• `solver -matrix_iterative_bicg`.

• Use `bounda_alternate`.

• If possible, don’t use extreme large indices (since memory is allocated for all indices).

9.14 Inaccurate results

• Set the interface stiffness to about 10 times the neighbouring element young divided by the neighbour length.

• If a structure is submerged in water, e.g. a one-side submerged dam, you need to impose the correct pressure condition; but you also need to impose the water loading by a `force_edge_water`.

9.15 Element sides

This section defines node numbers for element sides 0, 1, ... respectively.

For a **bar2** element the sides have the nodes numbers 0 and 1.

For a **tria3** element the sides have the nodes numbers 0,1 and 1,2 and 2,0.

For a **tria6** element the sides have the nodes numbers 0,1,2 and 0,3,5 and 2,4,5.

For quad elements the sides are in the order below, upper, left, right; see the pictures in `elements`.

For hex elements the sides are in the order below, upper, front, back, right, left; see the pictures in `elements`.

For a **tet4** element the sides have the nodes numbers 0,1,2 and 0,1,3 and 1,2,3 and 0,2,3.

For a **tet10** element the sides have the nodes numbers 0,1,2,3,4,5 and 0,1,2,6,7,9 and 2,4,5,7,8,9 and 0,3,5,6,8,9.

For a **prism6** element the sides have the nodes numbers 1,2,3 and 4,5,6 and 1,2,4,5 and 0,2,3,5 and 0,1,3,4.
For a **prism15** element the sides have the nodes numbers 0,9,1,11,10,2 and 3,12,4,14,13,5 and
0,9,1,6,7,3,12,4 and 1,10,2,7,8,4,13,5 and 0,11,2,6,8,3,14,5.

For a **prism18** element the sides have the nodes numbers 0,1,2,3,4,5 and 12,13,14,15,16,17 and
0,1,2,6,7,8,12,13,14 and 2,4,5,8,10,11,14,16,17 and 0,3,5,6,9,11,12,15,17.

9.16 Badly shaped elements

Each element should have at maximum one common side with a neighbouring element. For example
two neighbouring quad4 elements have only one common side in a proper element mesh; if the
neighbouring quad4 elements have two sides in common, the elements are badly shaped.

Some tochnog options will not work correctly if the mesh contains badly shaped elements.

9.17 For selected customers only

The **group_materi_plasti_* ** and **group_materi_umat_* ** options in combination with **con-
vection* and/or **node_mesh** options.

The **group_materi_umat** option.

The **_element_dof_apply** option.

The **bounda_dof ... -rotation** option.

The landslide algorithm.

Selected customers can require Tochnog to check their license with the **license_check** option.

9.18 Youtube

Tutorial movies can be found on https://www.youtube.com/channel/UC7qvITX-SLwA4RuqMPYBm

9.19 External programs.

- **http://www.gidhome.com** Commercial pre- and postprocessor; easy. It can write for
  Tochnog input directly. It can read results both from **control_print_gid** and from **con-
trol_print_vtk**. See the gid directory in your distribution.

- **http://mecway.com** Commercial finite element program; easy. It can write for Tochnog
  input_abaqus. It can read results from **control_print_gmsh**.

- **http://www.gnuplot.info/** Free x-y plotter; easy. It can read results from **control_print_history**
  and **control_print_data_versus_data**.

- **http://gmsh.info** Free pre- and postprocessor; complex. It can write for Tochnog
  input_abaqus. It can read results from **control_print_gmsh**.

- **http://www.paraview.org** Free postprocessor; complex. It can read results from **con-
trol_print_vtk**.

- **http://www.freecadweb.org/** Free CAD program; complex. It can write for Tochnog
  input_abaqus and read results from **control_print_frd**.
• [http://lace.fs.uni-mb.si/wordpress/borovinsek/?page_id=41](http://lace.fs.uni-mb.si/wordpress/borovinsek/?page_id=41) Free prepomax pre- and postprocessor for MS Windows; easy. It can write for Tochnog input_abaqus and read results from control_print_frd. Isoparametric elements only.


• [https://www.mikepoweredbydhi.com/products/feflow](https://www.mikepoweredbydhi.com/products/feflow) Dedicated ground water analysis program. Tochnog can read the mesh and hydraulic pressure heads with input_feflow_.*.

• [https://ngsolve.org/](https://ngsolve.org/) Free preprocessor in 3D. Tochnog can read the nodes and elements.

### 9.20 Forces are setup in the element loop in timesteps

The records force_edge, force_edge_normal, force_edge_projected, force_volume, condif_heat_edge_normal, condif_convection_edge_normal and condif_radiation_edge_normal are evaluated inside the element loop. Hence, the resulting nodal forces only get their values after a timestep is performed (since the element loop is performed in time steps).

### 9.21 Running in a Microsoft windows bash shell

When you want to run the linux executable in a Microsoft windows bash shell you need to set the environment symbol KMP_AFFINITY to disabled. So do `export KMP_AFFINITY=disabled` in the windows bash shell.
10 User supplied subroutines

Several skeleton user supplied subroutines are available in the file user.cpp. As a special option you can use an ABAQUS umat.f (ABAQUS is a trademark of Dassault Systemes; see HTTP://www.abaqus.com for the ABAQUS homepage). See also group_materi_umat in this manual. We do not support any compilation, linking or run-time problems with user supplied routines.

We only have available user.cpp and umat.f on linux 64 bit.
We use ourselves the linux mint 64 bit version specified in linux_version.txt file in the user supplied distribution. If you want to use the user supplied routines it is convenient to also use the same operating system in order to prevent compiling and linking problems; for this distribution the needed libraries are supplied.

If you insist on using another distribution than we do, you should install yourself the needed libraries (with commands like: sudo apt-get install gcc-7 g++-7). For the distribution of linux_version.txt the needed libraries are supplied.

Do the following:

- Download the tochnog_linux_64_bit_user_supplied.tar.gz distribution.
- In the user supplied distribution read README_UMAT_USER.txt
- In the user supplied distribution look in the makefile how to compile.

References


[16] Niemunis, A. 2003 *Extended hypoplasticity models for soils* Bochum, ISSN, 1439-9342


