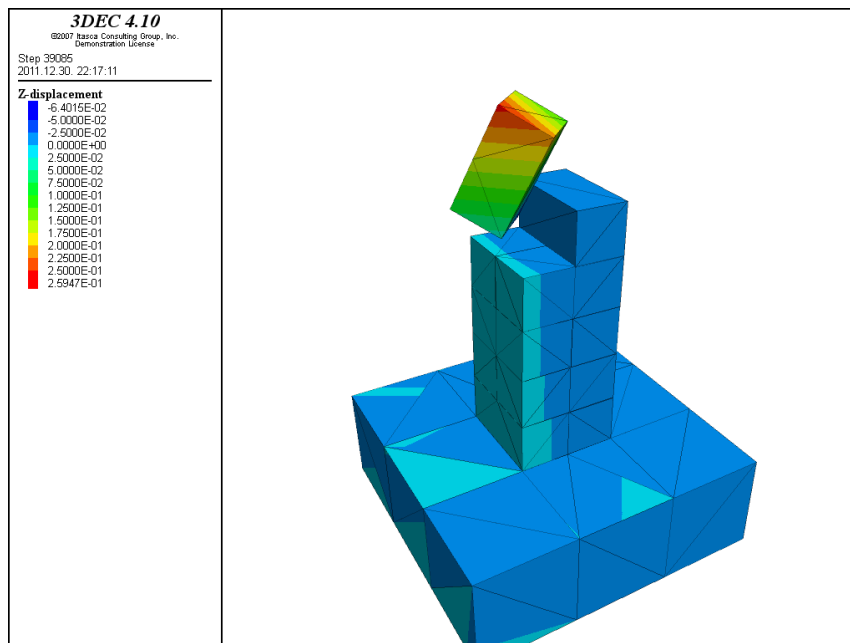


Boundary conditions in 3DEC



Written by:

Norbert Turi

BUTE Faculty of Civil Engineering MSc

Supervised by:

Katalin Bagi

BUTE Department of Structural Mechanics

CONTENTS

1. Introduction	2
2. Initial model: Masonry column loaded by self-weight	2
3. Types of boundaries	11
3.1. Load Boundary	12
3.2. Traction Boundary	17
3.3. Stress Boundary	22
3.4. Velocity (Displacement) Boundary	26
3.5. Free Boundary	56
3.6. Non-Reflecting (Viscous) Boundary	56
3.7. Reaction Boundary	56
4. Other possible functions of „Boundary”	57
5. Displaying results: PLOT command	60

Boundary conditions

1. Introduction

Basically two main categories of boundaries are used in 3DEC:

1. real: surfaces or edges that physically exist in the investigated area or structure (e.g.: tunnel surface, ground surface, stone ashlar, bricks – blocks)
2. artificial: boundaries that do not exist, but have important role in working of the numerical model. Sometimes they make it more simple, without decrease the efficiency of it (e.g.: symmetry).

Boundary command can be applied only after geometry and – in case of deformable blocks – mesh are generated. Defined boundaries do not change during the calculation until we won't overwrite them. There is only one exception, when at first **velocity boundaries** are defined and later we apply **stress boundary** then velocities change automatically according to the stress values. The stress boundary affects all degrees-of-freedom.

Loads applied by boundary command are constant but time-varying loads can be defined by using **history** keyword.

The values defined by boundary commands are always acting on the boundaries of the model. If we do not use the command **range** then the chosen values will be applied for all the gridpoints taking place on the boundaries, except the ones are hidden by the command **hide**. In case of using **range** we can specify a closer part of the structure or area where the boundaries working.

Boundary	stress	total stress applied (in case of rigid or deformable blocks)
	xload	load applied in x-direction (in case of rigid or deformable blocks)
	yload	load applied in y-direction (in case of rigid or deformable blocks)
	zload	load applied in z-direction (in case of rigid or deformable blocks)
	xvel	x-velocity applied (only in case of deformable blocks)
	yvel	y-velocity applied (only in case of deformable blocks)
	zvel	z-velocity applied (only in case of deformable blocks)

2. Initial model: Masonry column loaded by self-weight

In the examples of this script at first I always run the following command, which is defining the geometry, material properties (blocks and joints), support and applies the gravitational load for the column. In the commands there are several lines where the first character is „semicolon” (;), in this case the software do not cares what is written after the sign (;), so you can add notes.

```
new
;base-block 60x60x20cm (blue colored block on Fig. 1)
;origin is defined in the middle of the upper surface of the base-block
poly brick -0.3,0.3 -0.3,0.3 -0.2,0.
;
;block-levels made up with blocks sized: 10x10x20cm
;first level (blocks orientated in y-direction)
poly brick -0.1,0. -0.1,0.1 0.,0.1
poly brick 0.,0.1 -0.1,0.1 0.,0.1
;second level (blocks orientated in x-direction)
poly brick -0.1,0.1 -0.1,0. 0.1,0.2
poly brick -0.1,0.1 0.,0.1 0.1,0.2
```

```

;third level (blocks orientated in y-direction)
  poly brick -0.1,0. -0.1,0.1 0.2,0.3
  poly brick 0.,0.1 -0.1,0.1 0.2,0.3
;fourth level (blocks orientated in x-direction)
  poly brick -0.1,0.1 -0.1,0. 0.3,0.4
  poly brick -0.1,0.1 0.,0.1 0.3,0.4
;fifth level (blocks orientated in y-direction)
  poly brick -0.1,0. -0.1,0.1 0.4,0.5
  poly brick 0.,0.1 -0.1,0.1 0.4,0.5
;
;displaying of geometry
  plot block
  plot reset
  plot set dip 60. dd 210.
;
;mesh generation with 20cm average mesh-size
  gen edge 0.2
;
;material properties (sandstone):
;density: 2400 kg/m3
;E=19.3GPa Poisson-ratio: 0.38
;K=26.8GPa G=7GPa
  prop mat=1 dens=2400. k=2.68e10 g=7.e9
;
;joint properties (Coulomb-slip model)
;normal stiffness (1000 MPa/m) and shear stiffness (100 MPa/m)
;friction angle 25°
  change jcons=1
  prop jmat=1 jkn 1.e9 jks 1.e8 fric 25.
;
;supported on the lower surface of the base-block (velocity boundary with zero values in every direction)
  bound xvel=0.0 range x -0.31,0.31 y -0.31,0.31 z -0.21,-0.1
  bound yvel=0.0 range x -0.31,0.31 y -0.31,0.31 z -0.21,-0.1
  bound zvel=0.0 range x -0.31,0.31 y -0.31,0.31 z -0.21,-0.1
;
;for cycle is applied for rising the value of gravitational force
;this is necessary because some structures collapse if the gravitation is applied in one step
;"grav_lep" is the value that defines the number of cycles gravitation is applied
;the value of „gr” is rising with the value „g/grav_lep” in every cycle
;Calculation process is running until the structure reaches the desired accuracy
;this is measured by the ratio of unbalanced and applied forces
;now I set the limitvalue for 1e-5
  def gra
    grav_lep=10
    gr=0
  loop n (1,grav_lep)
    gr=gr-10/grav_lep
    command
      gravity 0.0,0.0,@gr
      solve rat 1.e-5
    end_command
  end_loop
  end;
;calling the „gra” command
  @gra

```

Besides the previously demonstrated „poly brick” command and „jset” command can be used for defining the geometry too. In this case the upper levels are made by a single block, that is later sliced by horizontal and vertical cutting-planes, that can be defined by **jset** command:

```

new
; base-block 60x60x20cm (blue colored block on Fig. 1)
; origin is defined in the middle of the upper surface of the base-block
  poly brick -0.3,0.3 -0.3,0.3 -0.2,0
;
; single block of upper leveles, size: 20x20x60cm
  poly brick -0.1,0.1 -0.1,0.1 0,0.6
;
;brick-levles (10x10x20cm) created by 'jset' command
;
;the 6 horizontal cutting-planes
  jset dip 0 dd 0 spac 0.1 num 5 origin 0,0,0.3
;
;vertical cutting-plane in y-direction
;the base-block and every second horizontal brick levels had
;to get hidden before the vertical cutting is made
;
;hiding of base-block:
  hide range x -0.3,0.3 y -0.3,0.3 z -0.2,0
;hiding of second brick-level:
  hide range x -0.1,0.1 y -0.1,0.1 z 0.1,0.2
;hiding of fourth brick-level:
  hide range x -0.1,0.1 y -0.1,0.1 z 0.3,0.4
;hiding of sixth brick-level:
  hide range x -0.1,0.1 y -0.1,0.1 z 0.5,0.6
;
;vertical cutting-plane in y-direction
  jset dip 90 dd 90 origin 0,0,0
;
;make every blocks visible
  seek

```

In this stage a very useful symplification can be used. After the horizontal block-levels had been hidden we can name the actually visible blocks (odd levels: 1, 3, 5) with the **mark region** command e.g.:

```
mark region 1
```

In this case we do not have to use the following command-line, because a shorter command can replace it:

```
hide reg 1
;vertical cutting planes (x-direction)
  jset dip 90 dd 0 origin 0,0,0
```

In the complicated version vertically unsliced blocks had to be visible:

```

;the base-block and every second horizontal brick levels had
;to get hidden before the vertical cutting is made
;
;hiding of base-block:
  hide range x -0.3,0.3 y -0.3,0.3 z -0.2,0
;hiding of first brick-level:
  hide range x -0.1,0.1 y -0.1,0.1 z 0,0.1
;hiding of third brick-level:
  hide range x -0.1,0.1 y -0.1,0.1 z 0.2,0.3
;hiding of fifth brick-level:
  hide range x -0.1,0.1 y -0.1,0.1 z 0.4,0.5

```

```

;vertical cutting-plane in x-direction
  jset dip 90 dd 0 origin 0,0,0
;make every blocks visible
  seek
;displaying of geometry
  plot block
  plot reset
  plot set dip 60 dd 210

```

It is obvious that the use of mark region command can make the command-line much shorter. The geometry of the column defined by the previously discussed command-lines can be seen in Fig. 1.

The cycle called „gra” can be written differently from the previously viewed one:

```

def gra
  grav_lep=10
  gr=0
  loop n (1,grav_lep)
    gr=(-10*n)/grav_lep
    command
      gravity 0.0,0.0,@gr
      solve rat 1.e-5
    end_command
  end_loop
end;

```

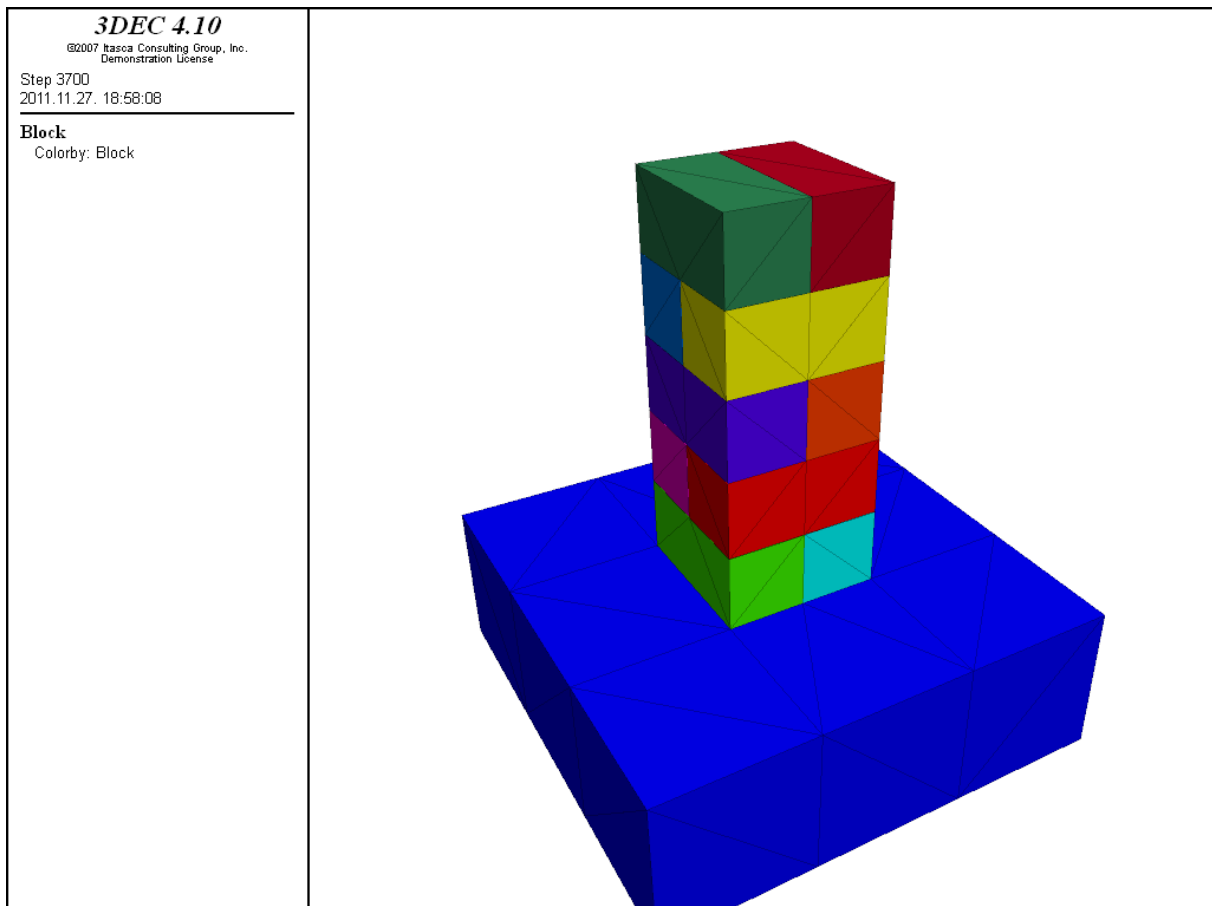


Fig. 1: Geometry of the column

```

;listing the change of the unbalanced forces during the calculation
  hist unbal

```

```

;unbalanced forces are plotted on a graph where the horizontal axle displays the number of cycles
;and the vertical axle contains the change in the value of unballanced forces
  plot hist 1 yaxis label 'change of unballanced forces'
;the calculation stops automatically, if the ratio of unbalanced and applied forces reaches
;the determined value (1e-5)
  solve rat 1.e-5

```

If the gravitational loading is applied for the structure in smaller parts, with the cycle called „gra”, the maximal settlement at the top of the column was 0.03864 mm (Fig. 2.).

Fig. 2. can be generated by the following code:

```

plot contour zdisp above au

```

Plot contour command defines different colors for the surface of the structure, according to the intensity of the displayed feature. In this case the displayed feature is z-displacement. If above command is used, the user can define the color that is used for the lowest displayed value. The software uses a special colour chart with individual codes for each colors. If „au” command is used with above, the software chooses the best (most visible, impressive) color automatically. Studying Fig. 2. it can be seen that the settlement of base block is zero, because I fixed the lower plane of the base.

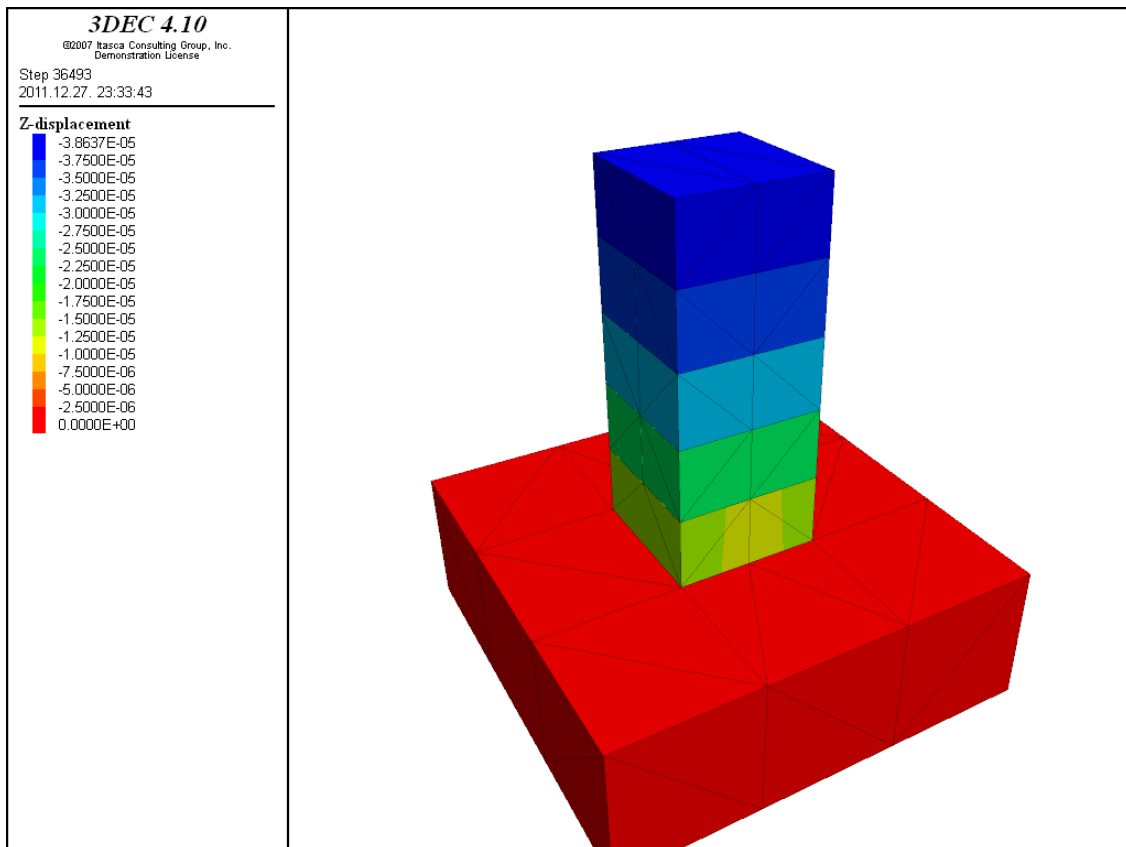


Fig. 2: Z-displacements [m] under gravitational load used by the „plot contour” command, gravity is applied in 10 cycles

In Fig. 3. the change of z-displacement of the top of the column can be seen over the cycles while ratio of unbalanced and applied forces reaches the chosen limit.

```

;listing the change of displacement in z-direction during the calculation
  hist zdisp (0.,0.,0.5)
  plot hist 1

```

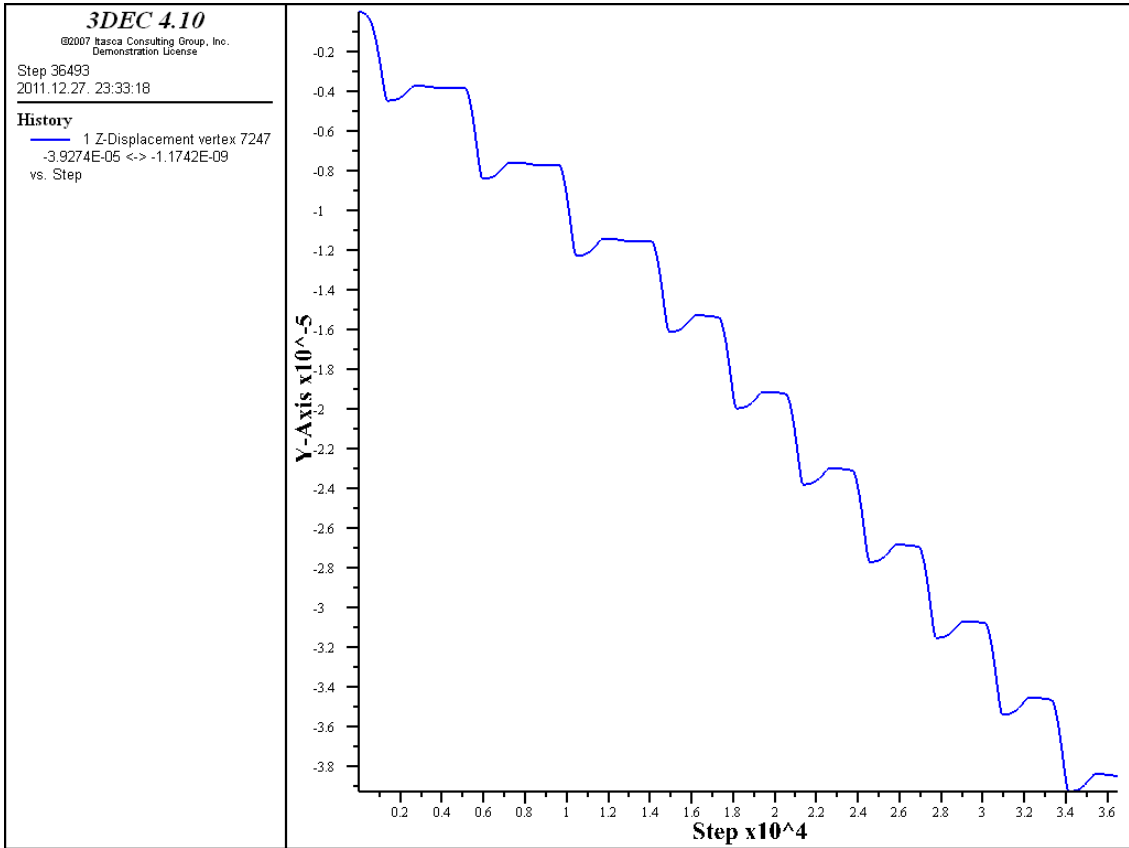


Fig. 3: Z-displacement of the top of the column [m] in case gravity is applied in ten cycles

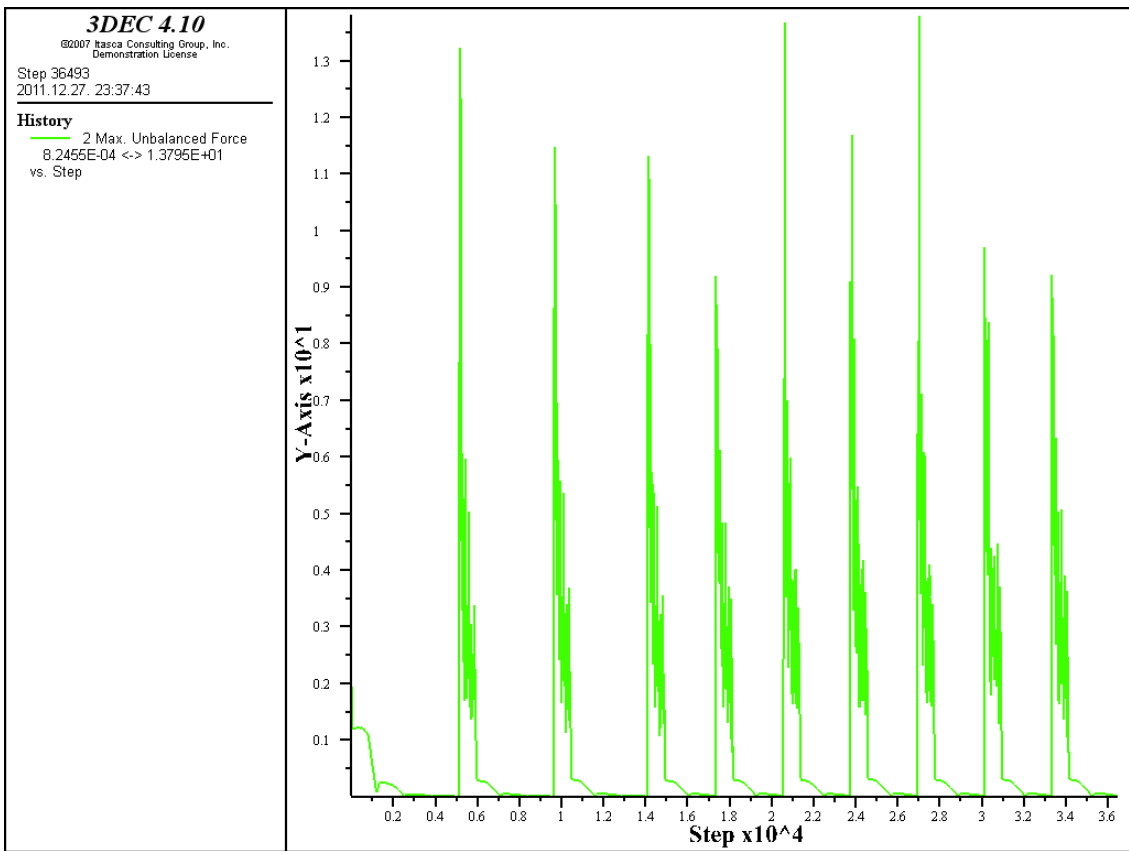


Fig. 4: Change of unbalanced forces [N] during the calculation, gravity is applied in 10 cycles

According to the graph after every load-impulse at first the displacements were higher, but after a few cycles the column rebounded. The settlement belongs to the balanced position is indicated by a plateau. The higher the number of cycles (applied gravitational force) is getting the shorter the length of the plateaus are. This can be attributed to the balancing effect of the increasing normal force.

On Fig. 4. change of unbalanced forces is displayed, the maximal unbalanced force was over 13.5 N, final balanced state was reached after 36000 cycles.

In Fig. 2., 3. and 4. gravity was applied in 10 cycles, however in Fig. 5. and 6. five cycles were used.

According to the graphs for the final balanced state the number of cycles have to be over 20000. So the reduced number of load-impulses had not resulted the same decline in the necessary solve-steps. Another important result can be determined by comparing Fig. 4. and 6., namely the value of maximal unbalanced forces had grown from ~13.5 N to ~26 N. So the higher number of load-impulses cause significant decrease in the maximal value of unbalanced forces. The maximal displacement was recorded by the value: 0.03863 mm, what is almost exactly the same as the results of calculation was carried out by 10 load-impulses. So in case of this structure the effect of increasing number of load-steps (load-impulses) can be neglected on values of displacements and on number of time-steps, but the maximal unbalanced forces are in inverse proportion to the number of load-steps.

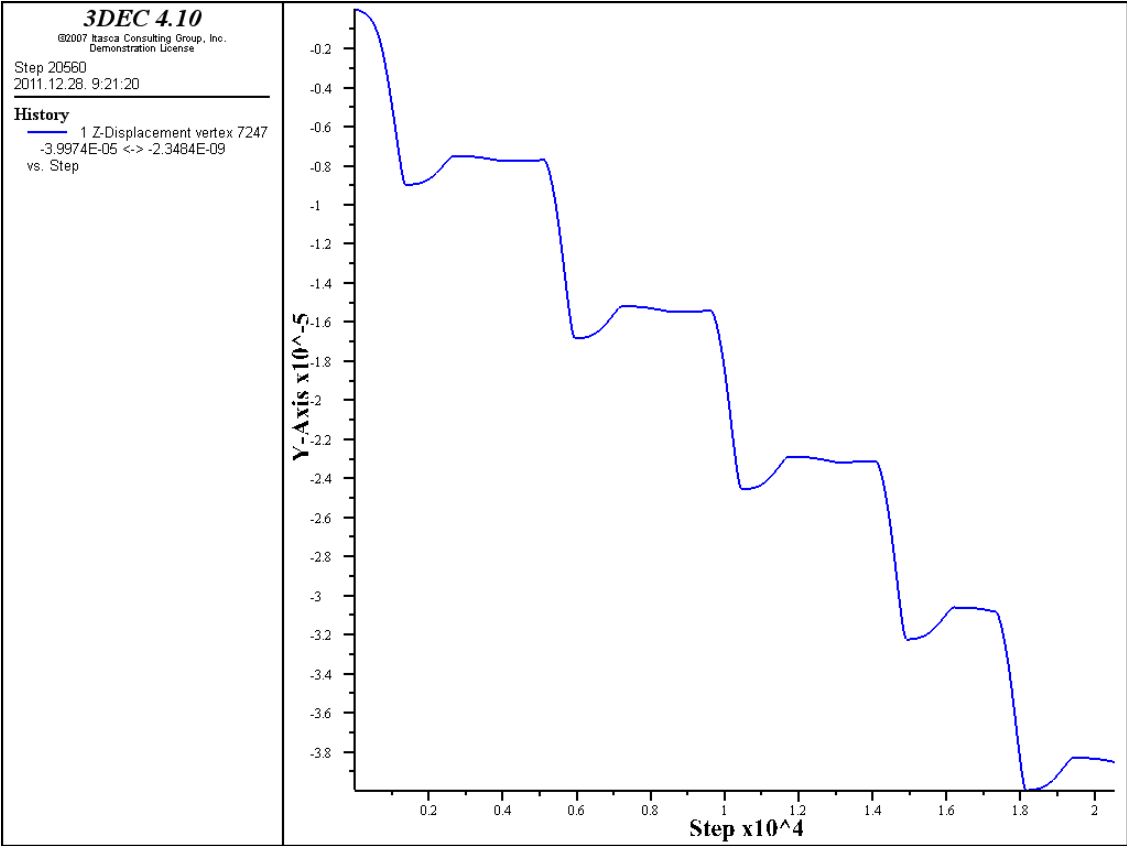


Fig. 5.: Z-displacement of the top of the column [m] in case gravity is applied in 5 cycles

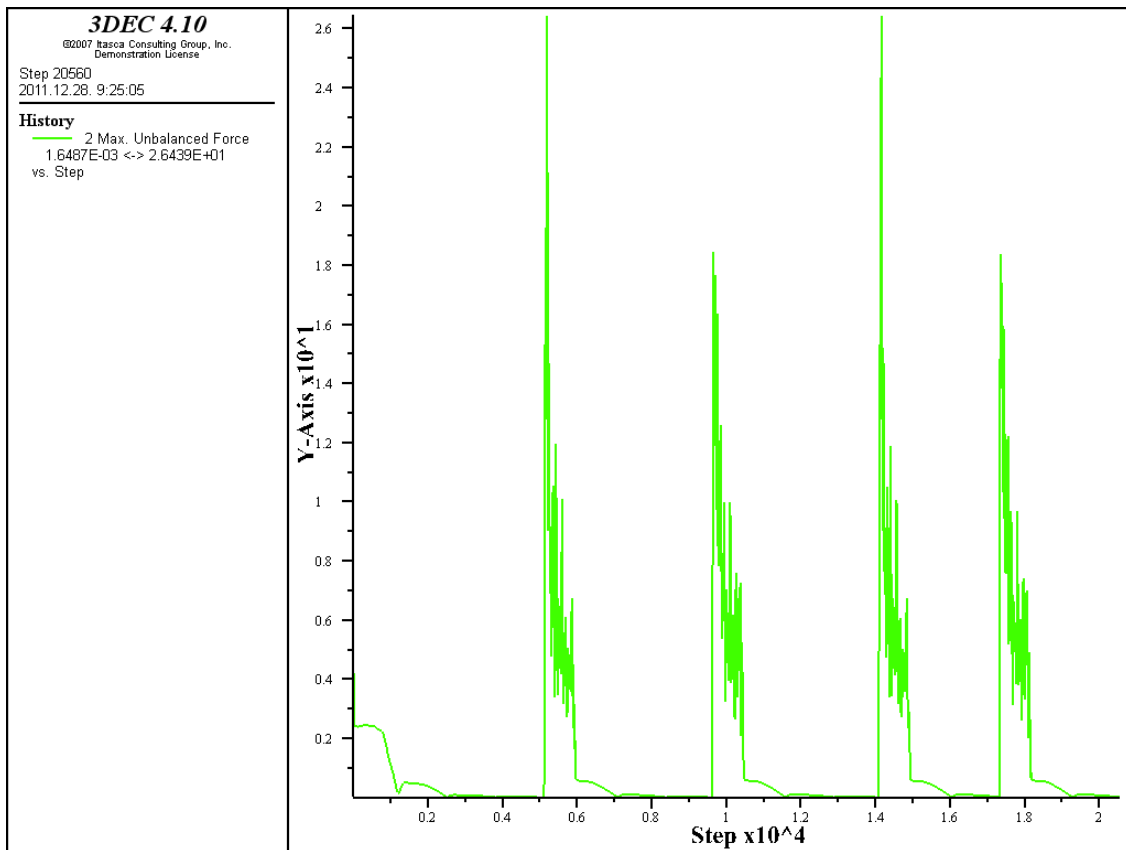


Fig. 6.: Change of unbalanced forces [N] during the calculation, gravity is applied in 5 cycles

If gravity is applied in one step to the model, the code is the following:

```

;gravity applied in one step
  gravity 0.0,0.0,-10.
;
; listing the change of unbalanced forces and z-dipslacement of x=0, y=0, z=0.5 point during the calculation
  hist zdisp (0.,0.,0.5)
  hist unbal
;
  plot hist 1
  plot hist 2
;
; the calculation stops automatically, if the ratio of unbalanced and applied forces reaches
;the determined value (1e-5)
  solve rat 1.e-5

```

The unbalanced forces acting during the calculation (gravity applied in one step) and caused by the previously demonstrated code can be seen on Fig. 7. Unbalanced forces acting during the first calculation-steps reached 22.951 N, but this value almost suddenly had dropped to ~12 N, that almost remained constant for about 800 calculation-steps. At the end of the plateau unbalanced forces nearly fell to zero. Over ~1100 calculation-steps unbalanced forces had started to grow again, but during antoher 800 steps the maximal force decreased to zero again. According to the graph over ~3400 steps the values of unbalanced forces can be neglected. Investigation of unbalanced forces can give informations about the point where we reach the required balanced-state. Unbalanced forces acting on gridpoints of structure will never get to zero because of the numerical faults, but applying „solve” commands we can require the desired accuracy. For example with the use of „solve rat” command the desired ratio of

unbalanced and applied forces can be given to the software. If the unbalanced force approaches a constant nonzero value, this probably indicates that joint slip or block failure and plastic flow are occurring within the model.

Weight of the structure, without the weight of the base-block:

$$(0.2\text{m}\cdot 0.2\text{m}\cdot 0.5\text{m}) \cdot 2400\text{kg}/\text{m}^3 = 48 \text{ kg} = 480 \text{ N}$$

During this short calculation the value of gravitational acceleration is supposed to be $10\text{m}/\text{s}^2$. Comparing the value of initial unbalanced force ($\sim 22\text{N}$) and applied gravitational forces (480 N). At rough guess according to the „ratio” (4,58%) - even at the very beginning of the calculation - the structure can be considered to be in balanced state. As it had been mentioned previously the „ratio” had dropped to $\sim 12 \text{ N}$ after the first few calculation-steps and after ~ 1200 cycles the value of unbalanced force was only $\sim 0.5 \text{ N}$ (0.10% of applied gravitational forces).

$$(22\text{N}/480\text{N}) \cdot 100 = 4.58 \%$$

$$(12\text{N}/480\text{N}) \cdot 100 = 2.50 \%$$

$$(0.5\text{N}/480\text{N}) \cdot 100 = 0.10 \%$$

The column was rebounded a few times, but finally over ~ 5000 cycles it reached to equilibrium.

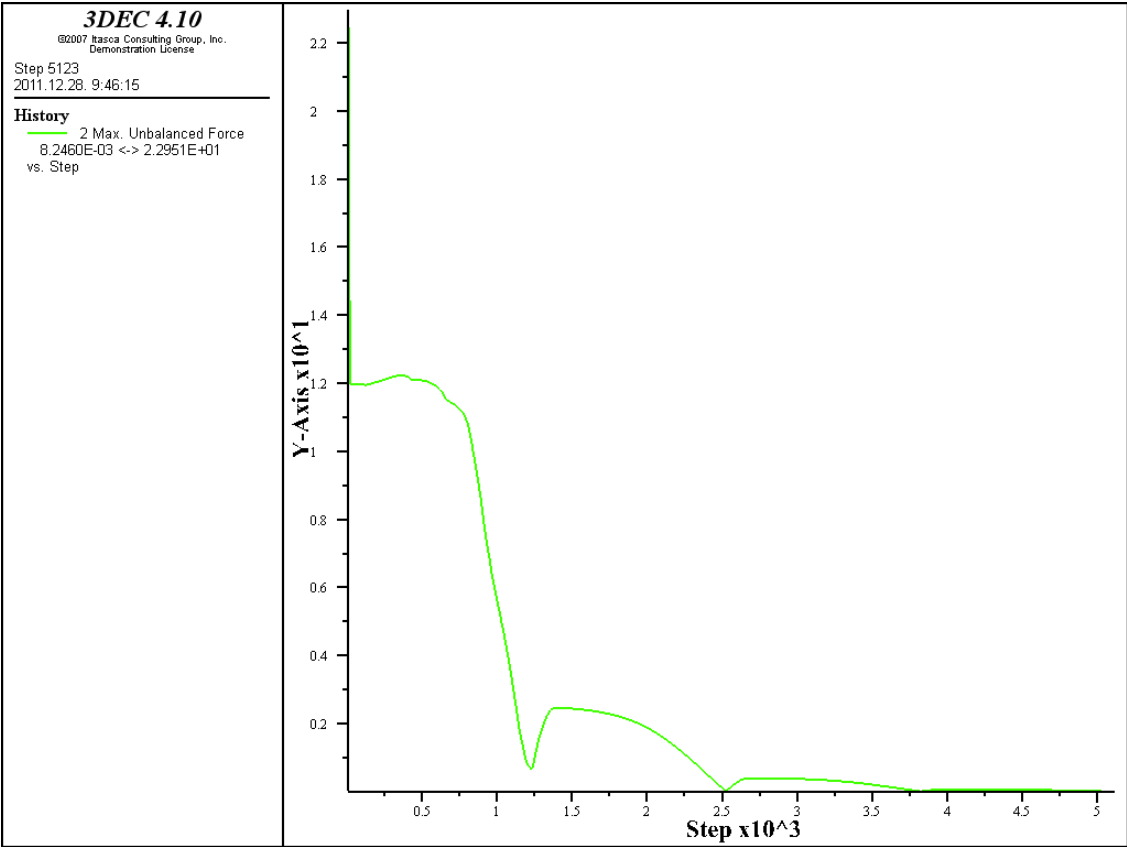


Fig. 7.: Change of unbalanced forces [N] during the calculation, gravity is applied in 1 cycle

In Fig. 8. z-displacements at the top of the column are represented in case of gravitational loading. The maximal displacement value is registered at the top of the column, namely: 0.03863 mm. This settlement is caused by the compression of the bricks. The maximal value of the settlement is

without reference to the number of load-steps. Phenomenon of rebounding occurs again, but it's not periodical because gravity was applied only in one step. As a consequence the modeled structure does not sensitive to gravitational loading, because the results hardly show an imperceptible change in displacements according to the number of time-steps.

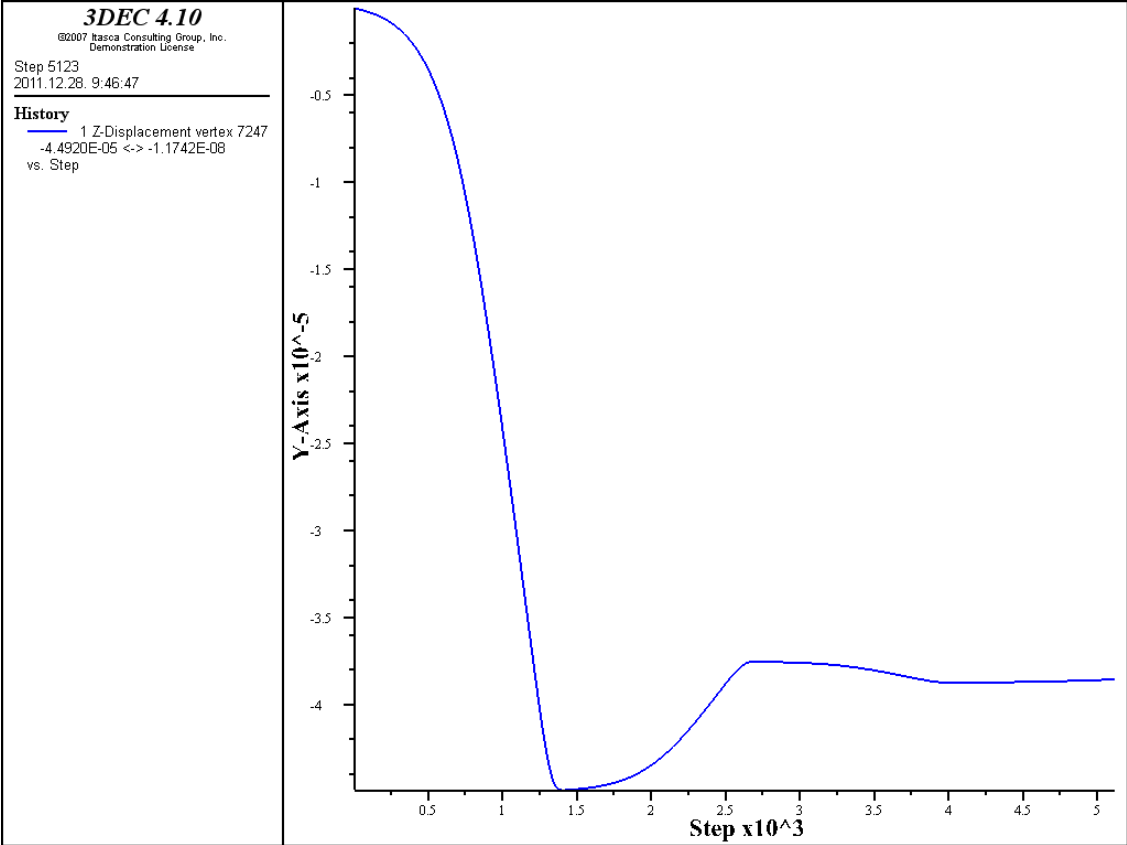


Fig. 8.: Z-displacement of the top of the column [m] in case gravity is applied in 1 cycle

3. Types of boundaries

The previously studied scripts besides geometry, materials and boundaries dealt with the gravitational loads. In the following sections different types of boundaries going to be demonstrated according to the order of Command Reference's „Boundary” section. From this point the previous sections supposed to be known and some of the boundary commands are dealt cursorily. In aim of a more effective guide detailed explanations, scripts and figures are announced at every command.

3.1. Load Boundary (concentrated load)

The user can define concentrated loads in every gridpoints (in real gridpoints or points of the generated mesh) of the structure's boundary with the commands: xload, yload, zload. After the „bound point” command the user have to give the point where the loads working - it can be done by the use of x, y, z coordinates - and the load-intensity. Sometimes it causes problems that the software cannot adopt direct coordinates of a gridpoint even if the user knows that it is accurate. In this cases the use of command „range” can be very useful, the user can define a range what contains the gridpoint. If the closer range of the point is not known by the user „list gridpoint location” command can be used to list all of the gridpoints

coordinates in the model. In huge models this command cannot help because of the too many search results, but if we use it with command „range” the searched area can be tapered. The actually studied structure is very simple so the accurate coordinates are known, but it always has to be considered that under loads the structure is deforming, that causes the moving of gridpoints. So the range has to be big enough to keep the point in it after deformations.

At first I tried to define the load in the middle of the upper plane of the column, but as it can be seen on Fig. 1. there are not any mesh-points in the middle of that range. It can be considered a drawback of the software that the user cannot manipulate the mesh directly but the geometry of the mesh is an important feature in consideration of calculation. In the demo version of the software number of gridpoints and blocks are limited, so I cannot generate gridpoints in the middle of the plane. However in the full-version where the number of elements are not limited the user can manipulate the mesh if he uses „slicing planes” and „join on” command. The method is the following: if we want to generate a more detailed mesh, or rise the number of elements in a range, use of „join on” command can help because the software generates mesh-points in intersections of slicing-planes. This method is sometimes complicated to use because these artificial slicing-planes (generated only for mesh manipulation) can influence the behaviour of the structure. For example if we do not give a high value for the joint stiffness properties along these artificial discontinuities than the block parts will fall apart or slide on each other under loads. Use of „join on” command is necessary because this ensures that joined blocks act as one block with a common centroid and volume.

Next command line lists every gridpoints’ coordinates within the range enclosed by the $z=0.45$ and $z=0.5$ horizontal planes. I had chosen these borders because I supposed that the gridpoints remain within this range after deformations.

```
list gridpoint location range z 0.45,0.5
```

Coordinates of gridpoints:

Block	grid-point	typ	x	y	z
6872	7308	7	9.9999890E-02	9.9999800E-02	4.9996179E-01
6872	7247	7	-1.1694335E-07	9.9999778E-02	4.9996147E-01
6872	7176	7	9.9999934E-02	-1.0000020E-01	4.9996170E-01
6872	7132	7	-6.7729495E-08	-1.0000023E-01	4.9996138E-01
6250	6686	7	-4.2096424E-08	9.9999793E-02	4.9996147E-01
6250	6625	7	-1.0000005E-01	9.9999785E-02	4.9996176E-01
6250	6554	7	-3.0301454E-08	-1.0000021E-01	4.9996138E-01
6250	6510	7	-1.0000003E-01	-1.0000022E-01	4.9996167E-01

The coordinates of points lying on the corners of the blocks are highlighted with grey color.

For the next script I only used the listed coordinates for orientation. With range command I defined ranges that containing the loaded points.

```
;concentrated loads in the corners of the upper plane of the column
  boundary zload -1000. range x -0.11 -0.09 y -0.11 -0.09 z 0.45 0.5
  boundary zload -1000. range x -0.11 -0.09 y 0.09 0.11 z 0.45 0.5
  boundary zload -1000. range x 0.09 0.11 y -0.11 -0.09 z 0.45 0.5
  boundary zload -1000. range x 0.09 0.11 y 0.09 0.11 z 0.45 0.5
;
hist unbal
  plot hist 2 yaxis label 'change of unbalanced forces'
  solve rat 1.e-6
```

The perpendicular (z-direction) loads are acting on the 4 peaks of the top of the column, their value is: -1000 N.

Change of unbalanced forces had been plotted during the calculation (Fig. 9.). Concentrated loads were applied only after the structure reached in balanced state under it's self-weight. Gravity was applied in 5 load-steps. According to Fig. 9. at the beginning of calculation value of unbalanced forces reached 900N. Structure got in ballanced state after ~2700 solve-steps.

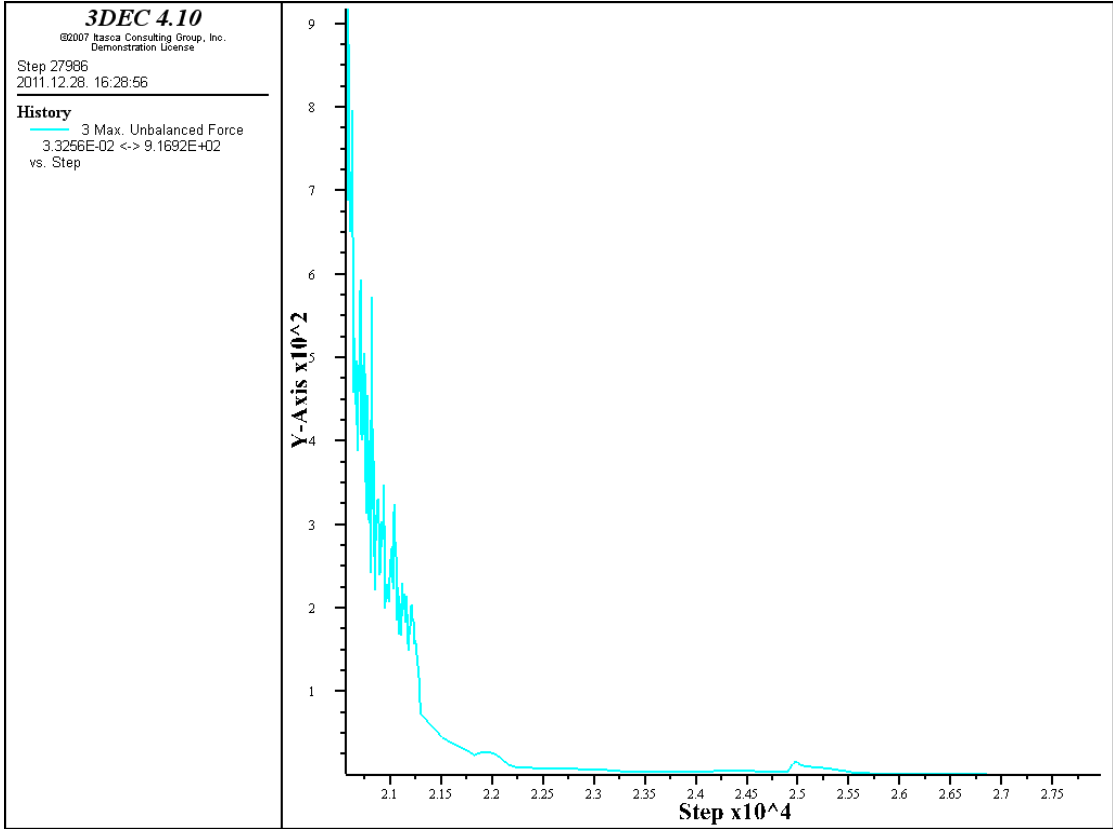


Fig. 9.: Change of unbalanced forces [N] during the calculation in case gravity and concentrated loads were applied

In Fig 10. z-displacements are displayed with „plot contour” command. Maximal perpendicular settlement was registered at the top of the column, namely: 0.68283 mm. Displacement value is higher than in case of gravitational loads (0.03863 mm). It could have been predicted without the software, but in Fig. 10. it can be seen on an expressive picture that the sides (where concentrated loads acting) of the top bricks had moved more than the middle-part of them. On the other hand it can be seen that the applied rough mesh is incapable to express the real settlements, where only the four peaks of the top plane and their closer range would have been moved. The evolving shape is semi-circular, because intermediate gridpoints can only be found on the edges parallel to the x-axle. The joint between the two top-bricks is parallel to y-axle, in this direction there aren't intermediate gridpoints so these edges remain linear. On Fig. 11. and 12. movement of a joint-edge intersection-point and a corner-point can be seen.

Fig. 11.	at teh end of the joint	0.0, -0.1, 0.5	(7132)
Fig. 12.	at corner	0.1, 0.1, 0.5	(7308)

As it can be seen on Fig 11. movements of joint-edge intersection-point is -0.4453 mm while in case of corner-point it's -0.6812 mm (Fig. 12.).

Just for curiosity movements in y- and x-directions were plotted too with „plot contour” command the results can be seen on Fig. 13. and 14. According to results of graph 13. the

joint between the two top-bricks is opening up. This horizontal movement is undoubtedly a consequence of the vertical movements resulted by the applied loads.

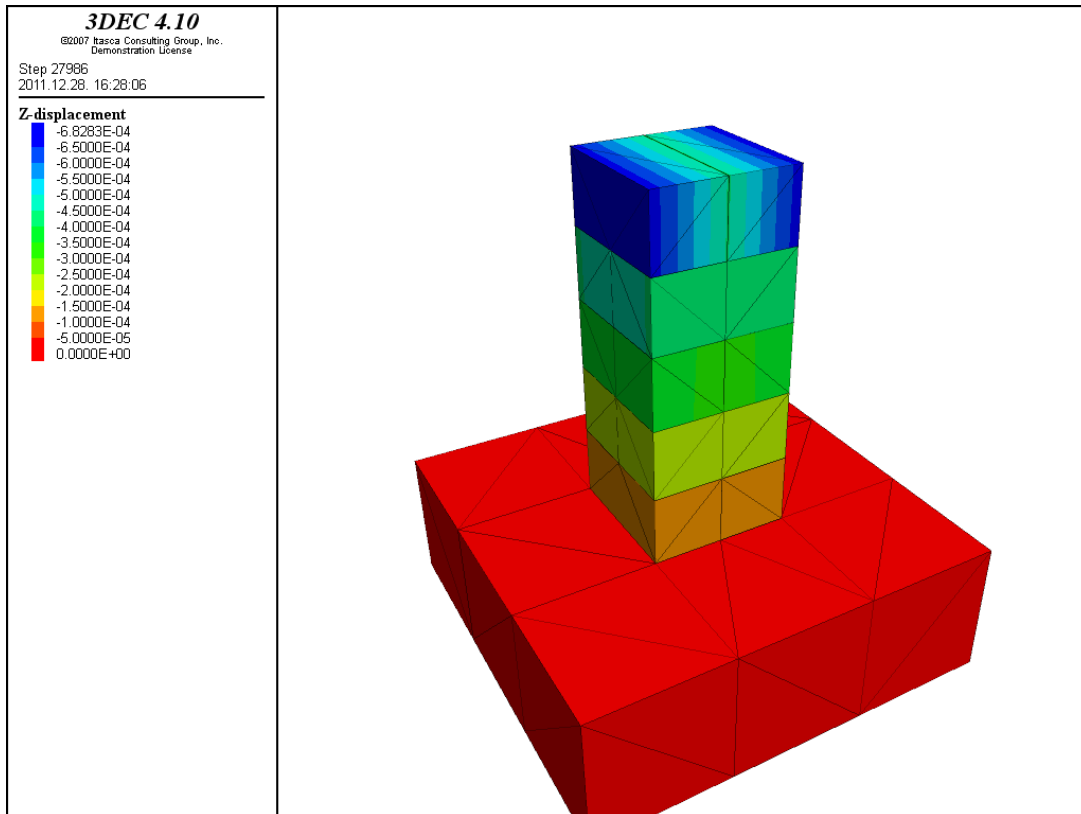


Fig. 10.: Z-displacements of the column [m] in case gravity and concentrated loads were applied

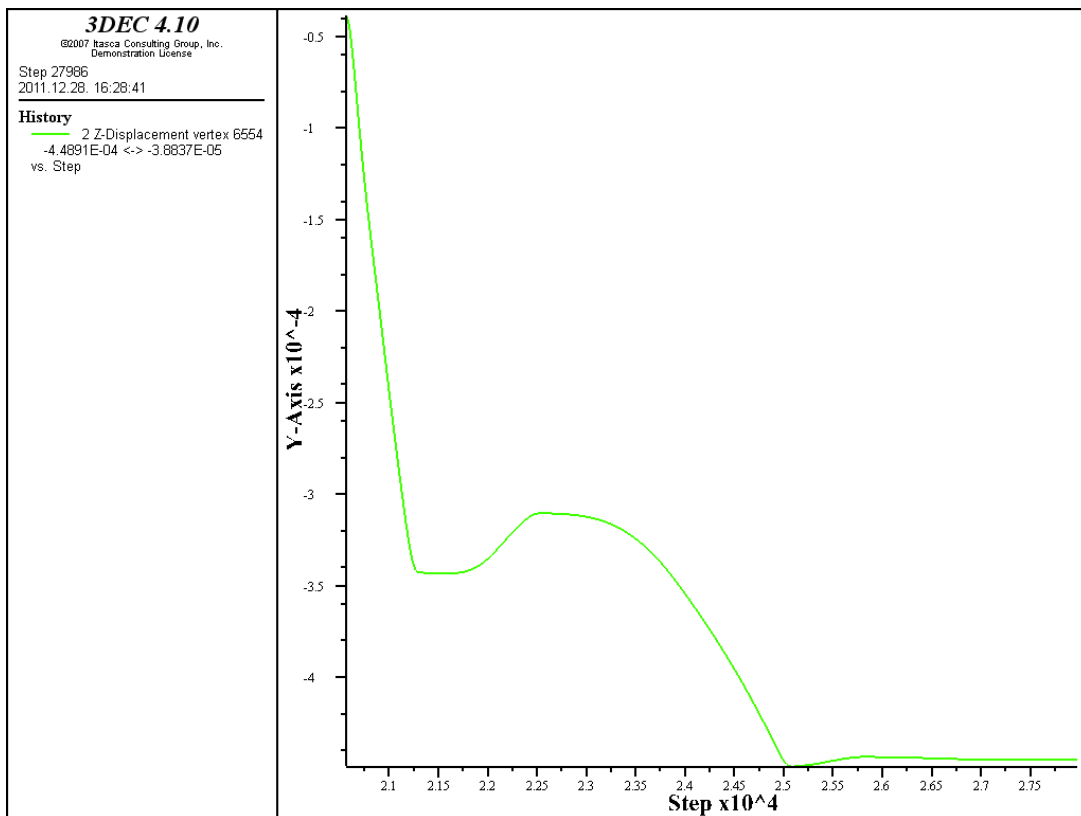


Fig. 11.: Z-displacements of the column [m] at the intersection of joint and edge (in case gravity and concentrated loads were applied)

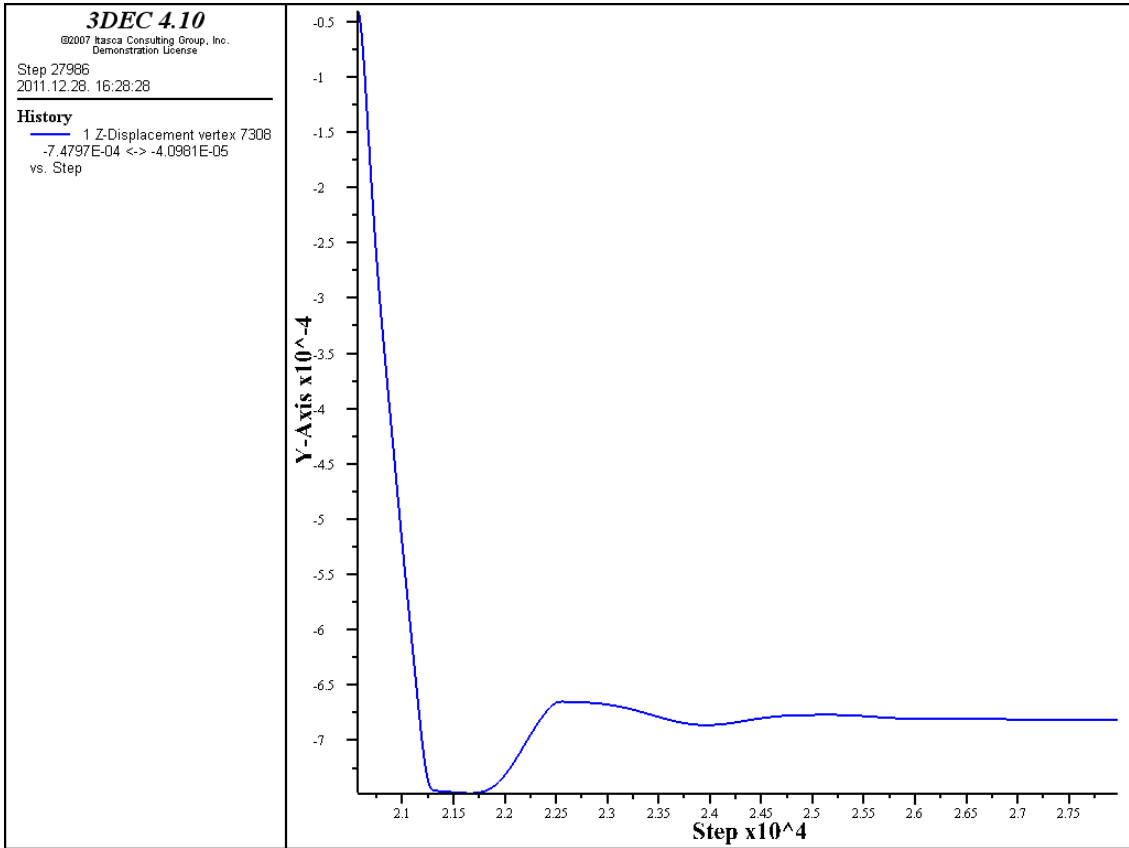


Fig. 12.: Z-displacements of the column [m] at the corners of top plane (in case gravity and concentrated loads were applied)

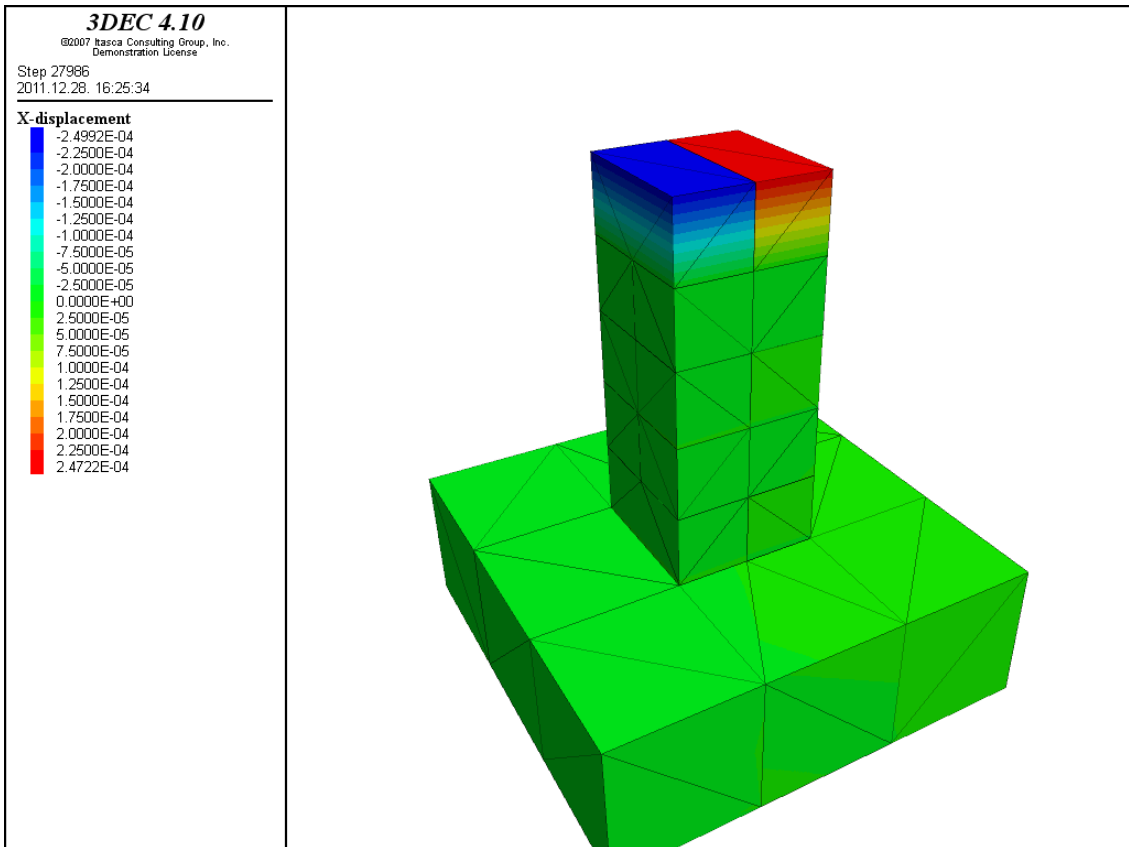


Fig. 13.: x-displacements of the column [m] in case gravity and concentrated loads were applied

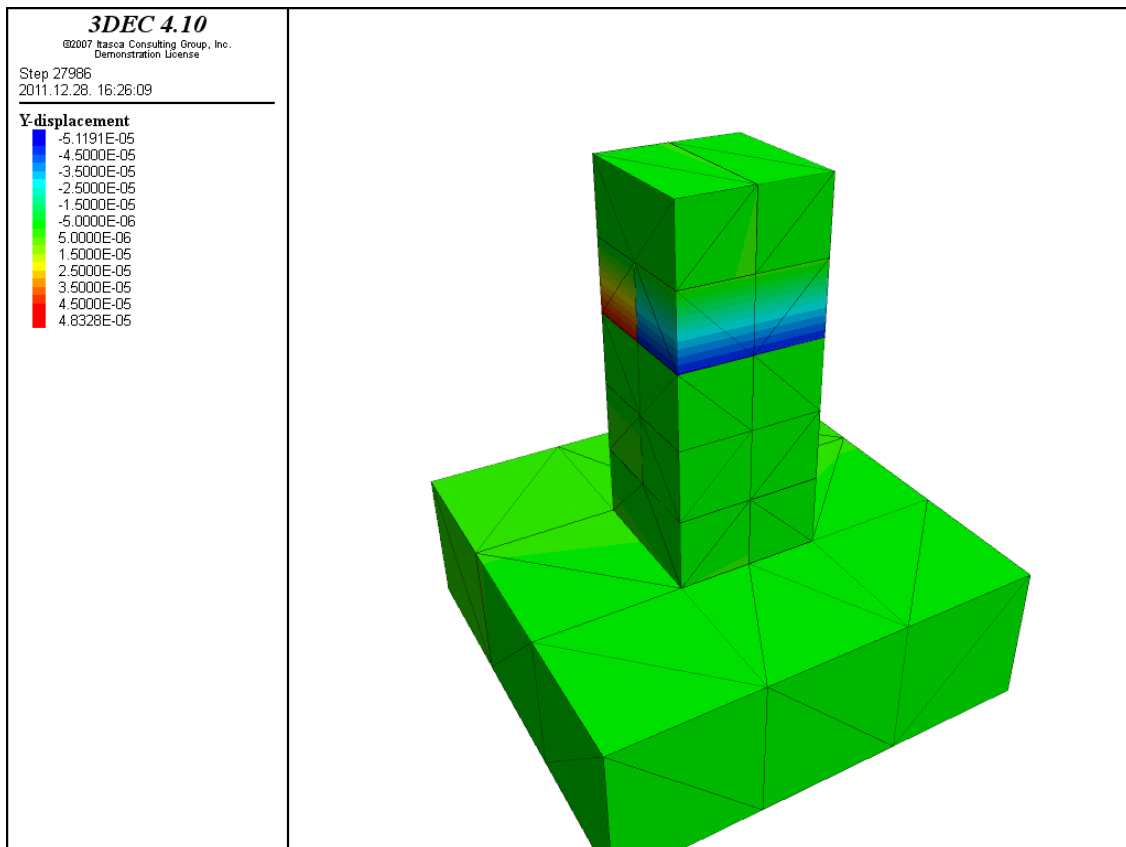


Fig. 14.: y-displacements of the column [m] in case gravity and concentrated loads were applied

On Fig. 14. the opening is acting on joint parallel to x-axle (in the brick-layer under the top bricks). The opening of joints can be explained by the well-known phenomenon of mechanics that openings usually came up on the plane where „tensile strength” acting. Another important feature of the structure is the existence of discontinuities.

3.2. Traction Boundary

During a real engineering project analysis of rock-mass response usually involves several different scales. It’s particularly true for problems, where location of the far-field boundary in the model can have a significant influence on results. In spite of the benefits of more precise and detailed models the large number of elements, gridpoints and joints usually demands a reasonable modeling approach. This involves starting with a large global model and proceeding through reasonable models to the smallest size required, with increasing complexity and detail added at each stage. This method can reduce the required calculation-time and can help to avoid some modeling mistakes, because the applied modeling-levels give better understand of the behaviour of structures or investigated fields. Furthermore the more levels reduce the complexity of each model.

3DEC has an automatic method of recording stresses at specified locations so that they can be applied as boundary tractions on smaller problems (higher modeling-levels). This unique feature ensures stress compatibility between larger and smaller models.

Boundary tractions applied to the detailed model are transferred from the regional model at points corresponding to the locations of the detailed model boundaries by the following procedure (Fig. 15.):

1. Run Model A.

2. Enter the command „set log on”.
3. Enter the command „list brick xl xu yl yu zl zu” where (xl, xu), (yl, yu) and (zl, zu) correspond to the boundaries of Model B.
4. The loads printed to the log file for the „list brick” command can be applied to Model B by using the „bound xtraction”, „bound ytraction” and „bound ztraction” commands.

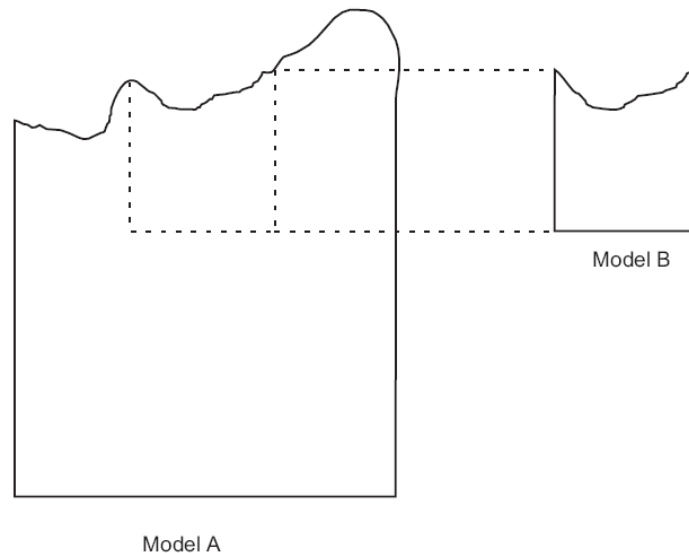


Fig. 15.: Theory of „boundary traction” [1]

„set log on” command:

This command sets global condition variables which remain constant during a run. If we set „log” command to „on”, a text file „lfname” opens in the current working directory. If lfname has not been specified with the „set logfile” command, then „lfname” is set equal to “3DEC.LOG”. Log-file involves every commands and calculation steps.

„list brick xl xu yl yu zl zu” command:

This command provides an alternative to the „list bound reaction” command. It is used when the stresses inside a global model are to be transferred as boundary conditions to a smaller model. The smaller model must be defined as a brick-shaped polyhedron. The command creates a file with „bound point”, „bound xload”, „bound yload” and „bound zload”, which are used as input in the smaller model data file. The global model need not have joints corresponding to the boundaries of the smaller model, as required by the „list bound reaction” command. The forces are calculated by sampling stresses in a regular grid located on the brick faces. The grid size may be defined by the user. The forces are corrected so that the force sums balance the gravity forces acting inside the brick region. It is assumed that no holes exist inside the brick. The default filename is “BRICK.PRN.” The default grid-spacing is (0.1 x largest brick dimension).

According to „Command Reference” at first the user has to define an optional triangular surface on the face of a deformable block, defined by the three vertices at (x1,y1,z1), (x2,y2,z2), (x3,y3,z3):

```
triangle x1 y1 z1 x2 y2 z2 x3 y3 z3
```

The place of the triangle is optional on the surface of the structure, so it is independent of the gridpoints or points of the generated mesh. According to the „Command Reference” the three points of the triangle must be entered in counterclockwise order, looking at the block-face from outside the block. In spite of that my tests were proved that **the load is only applied for the surface if it is given in clockwise order, looking at the block-face from outside the block.**

Load can be applied for the triangles with the use of „bound xtraction”, „bound ytraction” and „bound ztraction” commands. Intensity can be given automatically by the listed value of boundary-loads (imported from an other modeling-level) but any particular value is available too.

The following script is applying $1000\text{kN/m}^2 = 1000000\text{ kg/ms}^2 = 1\text{e}6\text{ kg/ms}^2$ distributed load for the upper plane of the column. If the generated mesh is taken into account then 4 triangles can be found on the top of the column. At first I defined the loads according to these triangles.

```

boundary triangle &
    0.000E+00 1.000E-01 5.000E-01 &
    1.000E-01 1.000E-01 5.000E-01 &
    1.000E-01 -1.000E-01 5.000E-01 &
xtraction 0.0 &
ytraction 0.0 &
ztraction -1.0E+06
boundary triangle &
    1.000E-01 -1.000E-01 5.000E-01 &
    0.000E+00 -1.000E-01 5.000E-01 &
    0.000E+00 1.000E-01 5.000E-01 &
xtraction 0.0 &
ytraction 0.0 &
ztraction -1.0E+06
boundary triangle &
    0.000E+00 1.000E-01 5.000E-01 &
    0.000E+00 -1.000E-01 5.000E-01 &
    -1.000E-01 1.000E-01 5.000E-01 &
xtraction 0.0 &
ytraction 0.0 &
ztraction -1.0E+06
boundary triangle &
    0.000E+00 -1.000E-01 5.000E-01 &
    -1.000E-01 -1.000E-01 5.000E-01 &
    -1.000E-01 1.000E-01 5.000E-01 &
xtraction 0.0 &
ytraction 0.0 &
ztraction -1.0E+06

```

The following script works with only two triangles but it loads the whole upper plane of the column like the previous one.

```

boundary triangle &
    -1.000E-01 1.000E-01 5.000E-01 &
    1.000E-01 1.000E-01 5.000E-01 &
    1.000E-01 -1.000E-01 5.000E-01 &
xtraction 0.0 &
ytraction 0.0 &
ztraction -1.0E+06
boundary triangle &

```

```

-1.000E-01  1.000E-01  5.000E-01 &
 1.000E-01 -1.000E-01  5.000E-01 &
-1.000E-01 -1.000E-01  5.000E-01 &
xtraction 0.0 &
ytraction 0.0 &
ztraction -1.0E+06

```

So the result were the same in case of both scripts.

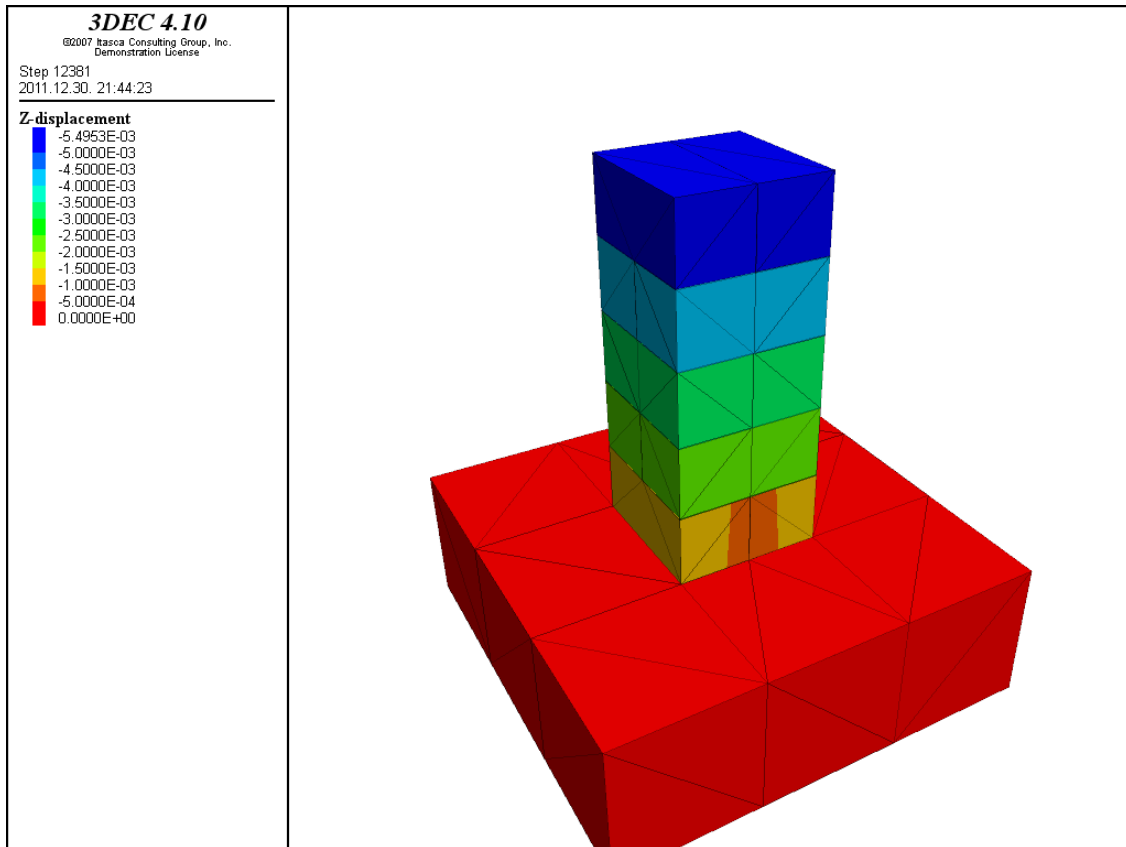


Fig. 16.: z-displacements of the column [m] in case distributed load was applied for the top of the column with traction command

Finally I demonstrate that traction command is irrespective of the gridpoints and points of mesh. For the sake of it the next script is defining a square (sized: 0.06 m x 0.06 m) with two triangles on the top of the structure.

```

boundary triangle &
  3.000E-02  3.000E-02  5.000E-01 &
  3.000E-02 -3.000E-02  5.000E-01 &
 -3.000E-02  3.000E-02  5.000E-01 &
xtraction 0.0 &
ytraction 0.0 &
ztraction -1.0E+06
boundary triangle &
 -3.000E-02  3.000E-02  5.000E-01 &
  3.000E-02 -3.000E-02  5.000E-01 &
 -3.000E-02 -3.000E-02  5.000E-01 &
xtraction 0.0 &
ytraction 0.0 &
ztraction -1.0E+06

```

The results can be seen on Fig. 17. Studying the picture it's obvious that the center of the column is loaded, because the values of displacements are higher at that part. The displacements show the same phenomenon that was represented in the previous section (Load boundary). Namely the loose mesh cannot give the real behaviour of the blocks because the software cannot represent deflections of edges where internal gridpoints do not exist (rigid-block behaviour in y-direction). On the other hand transversal direction of brick levels influencing the displacements and it decreasing the settlements of the upper bricks in y-direction. On the whole a mesh with higher density would give more realistic results.

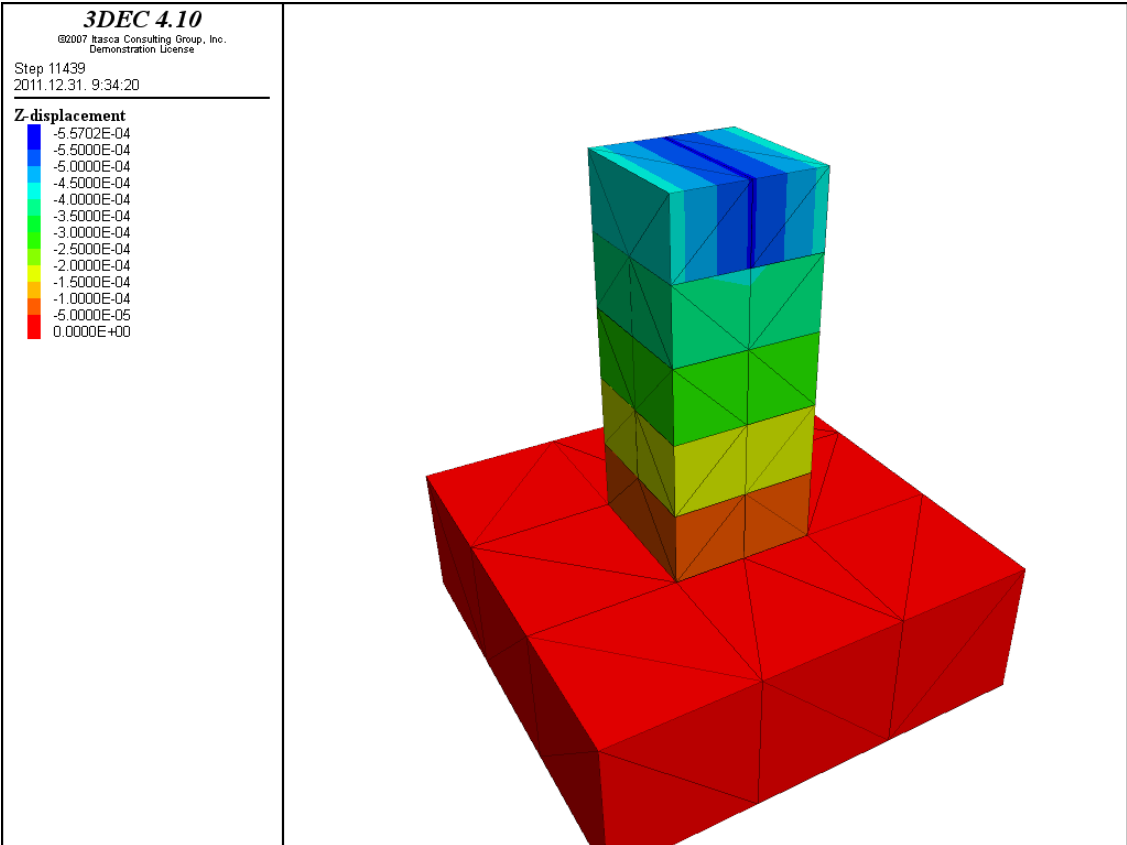


Fig. 17.: z-displacements of the column [m] in case distributed load was applied for a 0.06x0.06 m square on the top of the column with traction command

In case distributed load is applied with traction command, the direction of load (tension or compression) depends on the direction of global coordinate system. For instance if the applied load is perpendicular and pointing downwards (like in the previous example), the user have to define the value of ztraction with a negative number. The z-axis of the software's global coordinate system is pointing upwards.

On Fig 18. displacements can be seen when the applied load is pointing upwards, the script is the following.

```

boundary triangle &
  0.000E+00 1.000E-01 5.000E-01 &
  1.000E-01 1.000E-01 5.000E-01 &
  1.000E-01 -1.000E-01 5.000E-01 &
xtraction 0.0 &
ytraction 0.0 &
ztraction 1.0E+06
boundary triangle &
  1.000E-01 -1.000E-01 5.000E-01 &

```

```

0.000E+00 -1.000E-01 5.000E-01 &
0.000E+00 1.000E-01 5.000E-01 &
xtraction 0.0 &
ytraction 0.0 &
ztraction 1.0E+06
boundary triangle &
0.000E+00 1.000E-01 5.000E-01 &
0.000E+00 -1.000E-01 5.000E-01 &
-1.000E-01 1.000E-01 5.000E-01 &
xtraction 0.0 &
ytraction 0.0 &
ztraction 1.0E+06
boundary triangle &
0.000E+00 -1.000E-01 5.000E-01 &
-1.000E-01 -1.000E-01 5.000E-01 &
-1.000E-01 1.000E-01 5.000E-01 &
xtraction 0.0 &
ytraction 0.0 &
ztraction 1.0E+06

```

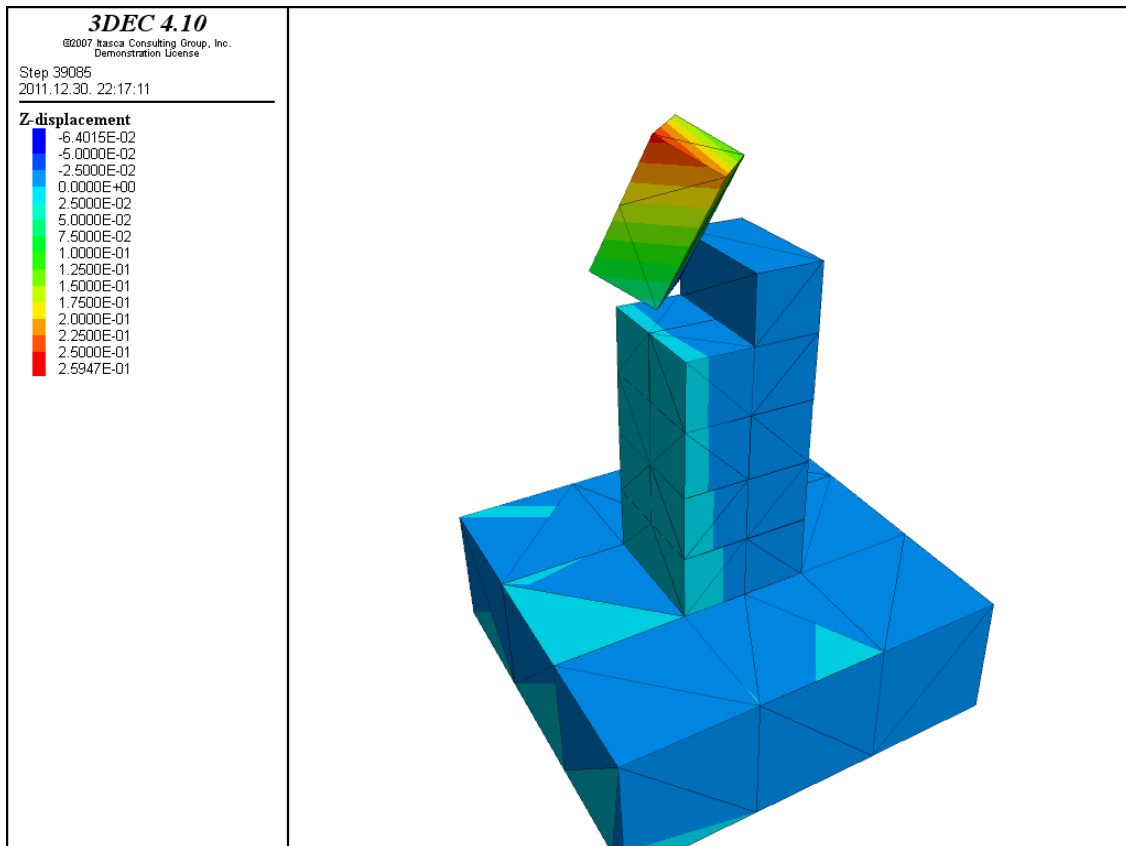


Fig. 18.: z-displacements of the column [m] in case distributed load (pointing upwards in z-direction) was applied for the top of the column with traction command

3.3. Stress Boundary

It does not count if loads are applied with „stress boundary” or ”traction boundary”, the results remains the same in both cases. As in the previous chapter, distributed load was acting on the whole upper plane of the column, with the value: 1000kN/m^2 .

Rigid or deformable blocks are both available with the stress command, user can define every items of the stress tensor (σ_{xx} , σ_{yy} , σ_{zz} , σ_{xy} , σ_{xz} , σ_{yz}).

```
boundary x y z stress  $\sigma_{xx}$ ,  $\sigma_{yy}$ ,  $\sigma_{zz}$ ,  $\sigma_{xy}$ ,  $\sigma_{xz}$ ,  $\sigma_{yz}$ 
```

```
;distributed load on the upper plane of the column
```

```
;loads were applied with "bound stress" command
```

```
;value of the perpendicular distributed load: 1000kN/m2=1000000 kg/ms2
```

```
bound (-0.11,0.11) (-0.11,0.11) (0.45,0.5) stress 0.,0.,-1.e6,0.,0.,0.
```

Next script gives the same results as the previous one:

```
bound stress 0.,0.,-1.e6 0.,0.,0. range x -0.11,0.11 y -0.11,0.11 z 0.45,0.5
```

Change of unbalanced forces and z-displacements are plotted on following pictures. In the beginning of calculation process value of unbalanced forces was higher than 4500 N, after ~1200 solve-steps it nearly dropped to zero (Fig. 19.).

Maximal z-displacement of the column was ~5.5 mm (see Fig. 20.).

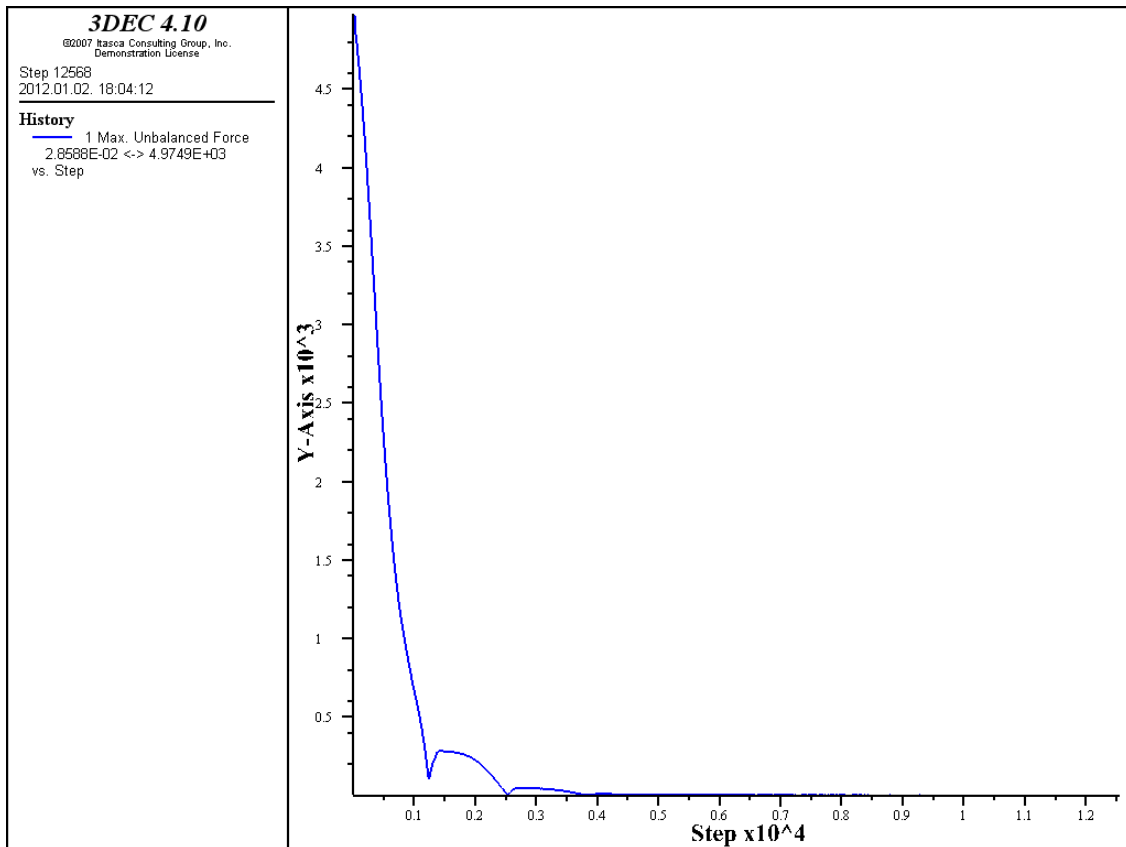


Fig. 19.: Change of unbalanced forces [N] during the calculation in case distributed load was applied with „stress” command

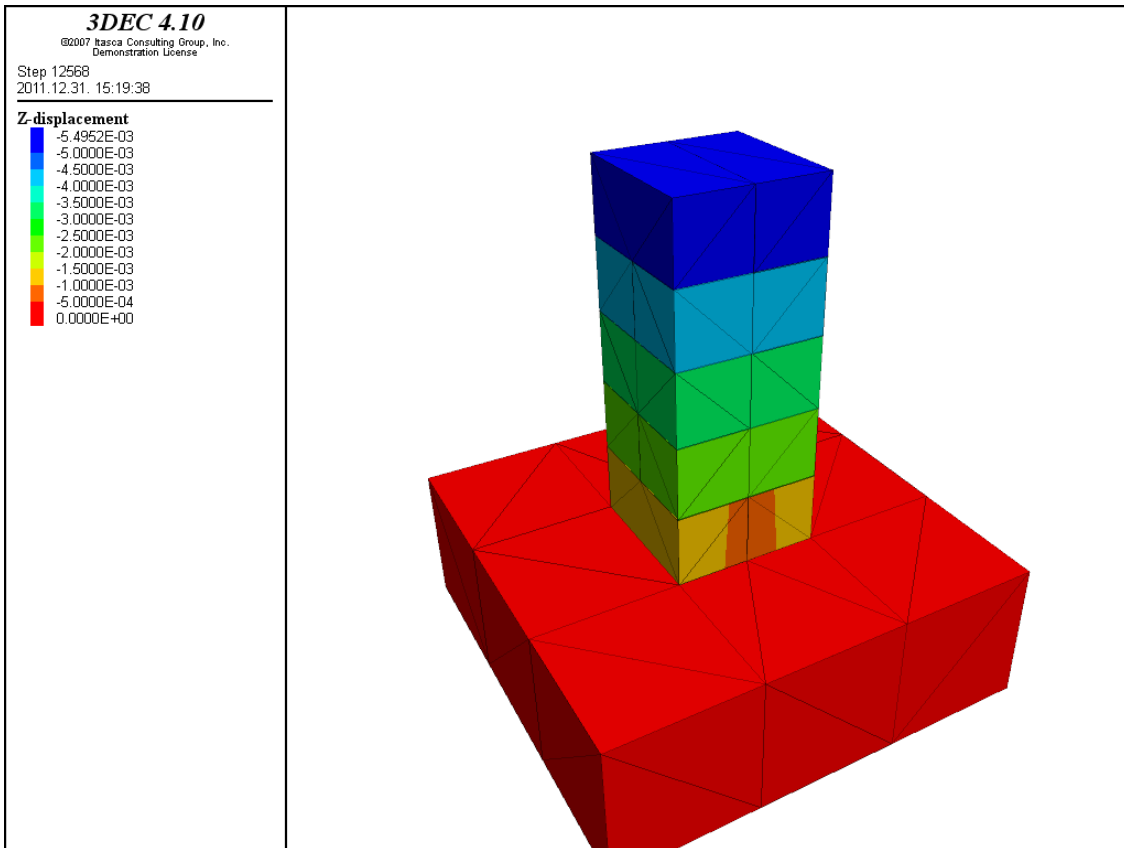


Fig. 20.: z-displacements of the column [m] in case distributed load was applied with „stress” command

„Boundary stress” command is appropriate for define linearly varying distributed load, in this case xgrad, ygrad and zgrad commands are used, according to the next section:

xgrad → sxxx syyx szzx sxyx sxzx syzx
ygrad → sxyy syyy szzy sxyy sxzy syzy
zgrad → sxxz syyz szzz sxyz sxzz syzz

$$\begin{aligned} \sigma_{xx} &= \sigma_{xx}^{\circ} + (sxxx)x + (sxyy)y + (sxxz)z \\ \sigma_{yy} &= \sigma_{yy}^{\circ} + (syyx)x + (syyy)y + (syyz)z \\ \sigma_{zz} &= \sigma_{zz}^{\circ} + (szzx)x + (szzz)y + (szzz)z \\ \sigma_{xy} &= \sigma_{xy}^{\circ} + (sxyx)x + (sxxy)y + (sxyz)z \\ \sigma_{xz} &= \sigma_{xz}^{\circ} + (sxzx)x + (sxzy)y + (sxzz)z \\ \sigma_{yz} &= \sigma_{yz}^{\circ} + (syzx)x + (syzy)y + (syzz)z \end{aligned}$$

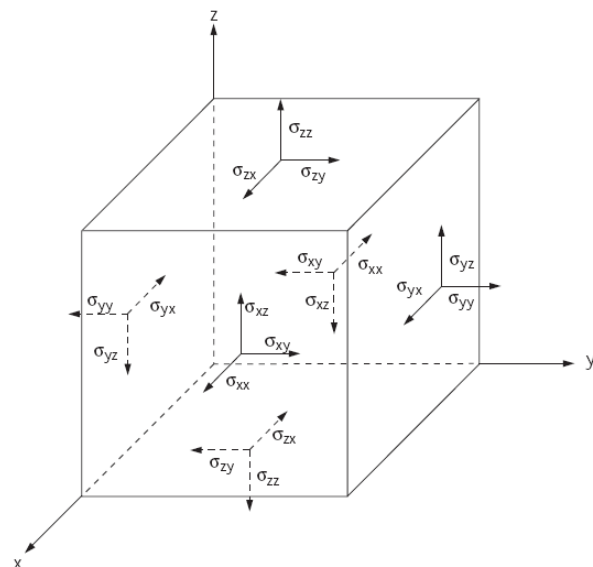


Fig. 21.: Positive orientation and point of acting of stress vectors

$\sigma_{xx}^{\circ}, \sigma_{yy}^{\circ}, \sigma_{zz}^{\circ}, \sigma_{xy}^{\circ}, \sigma_{xz}^{\circ}, \sigma_{yz}^{\circ}$ – stresses at origin ($x=0, y=0, z=0$), from this point the boundary stress is linearly varying.

If stresses at origin are not defined, the software automatically sets them to zero.

```
boundary -0.11,0.11 -0.11,0.11 0.45,0.51 &
stress 0.,0.,-1.e4 0.,0.,0. &
xgrad 0.,0.,1.e5 0.,0.,0.
```

After „boundary” ranges (where loads acting) have to be defined. On the other hand these coordinate boundaries (x1,x2 y1,y2 z1,z2) has another role, because they are influencing the stress values according to the following equations. In the previous script it can be seen that the defined range is a 0.01 m bigger in every direction than the geometry of the structure. It is very important, because the software sometimes cannot apprehend the loads without these „extra ranges”. These „extra areas” have an insignificant role on the value of stresses because the previously mentioned calculation method. Anyway the difference between the accurate and the applied values is negligible.

According to the previous script stresses at the origin are the following:

$$\begin{aligned}\sigma_{xx}^{\circ} &= 0 \\ \sigma_{yy}^{\circ} &= 0 \\ \sigma_{zz}^{\circ} &= -1 \times 10^4 \\ \sigma_{xy}^{\circ} &= 0 \\ \sigma_{xz}^{\circ} &= 0 \\ \sigma_{yz}^{\circ} &= 0\end{aligned}$$

Stress in z-direction (perpendicular) is linearly varying according to the following equation:

$$\sigma_{zz} = \sigma_{zz}^{\circ} + (s_{zzx})x + (s_{zzy})y + (s_{zzz})z$$

$$\text{xgrad} \rightarrow \text{sxxx syyx } s_{zzx} \text{ sxyx sxzx syzx}$$

So approaching to the edge $x=-0.1$ stress value is rising, while towards $x=0.1$ it's decreasing, because of the negative sign. In this example at the edge ($x=0.1$) stress value is 0, while at $x=-0.1$ it's twice of the value given in the middle of the column (0,0,0).

$$\sigma_{zz} = -1 \cdot 10^4 + (10^5) \cdot x$$

Value of σ_{zz} at $z = -0.1$ coordinate is $-1 \cdot 10^4 + (10^5) \cdot (-0.1) = -2 \cdot 10^4$ and at $z=0.1$ coordinate is 0 between these two values it change lineary. This tendence is certificated by Fig. 22. where the displacements of the structure can be seen.

The highest settlement values were reported around the negative coordinates of the x-axis, while at positive x-coordinates it heading towards zero. The load causes compression in the whole crosssection.

So with the use of „grad” command several kind of loadtypes can be created.

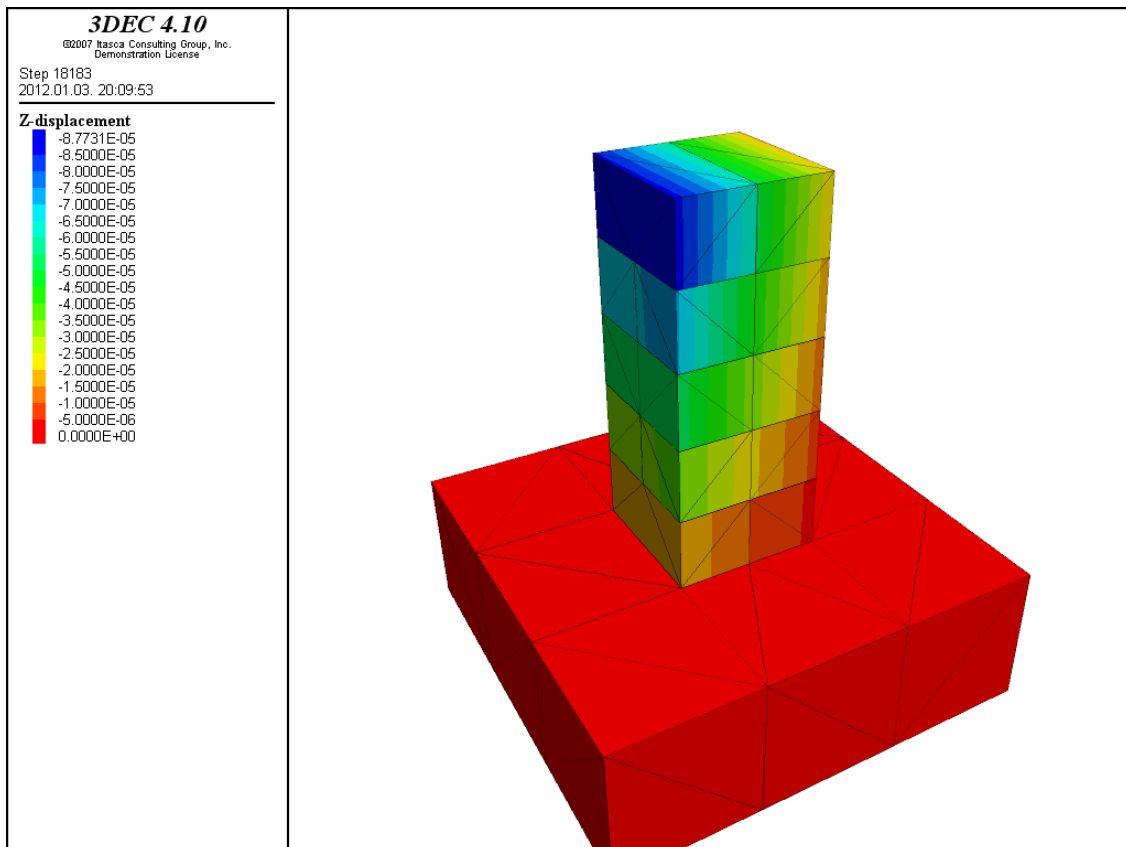


Fig. 22.: z-displacements of the column [m] in case „boundary grad” command was applied

3.4. Velocity (Displacement) Boundary

Displacements cannot be controlled directly in 3DEC; in fact, they play no part in the calculation process. In order to apply a given displacement to a boundary of a deformable block, it is necessary to fix the boundary and prescribe the boundary’s velocity for a given number of steps. The commands „boundary xvel, yvel and zvel” fix the x-, y- and z components of velocity at selected boundary gridpoints.

If the desired displacement is „d” and a velocity (v) is applied for a time increment (t), then displacement is calculated with $d = v \cdot t$ equation. Where $t = \Delta t \cdot n$, „ Δt ” is the timestep and „n” is the number of steps (or cycles). In practice, „v” should be kept small and „n” large, in order to minimize shocks to the system being modeled.

The „apply xvel” („yvel”, „zvel”) commands have the same role as „boundary velocity”, but they specify a velocity for rigid blocks. „Apply” defines constant velocities for rigid blocks centroids that can be found in range where „apply” command is used

In this section FIX and VELOCITY BOUNDARY commands were used at the same time. Although it is not important, because only one of these commands is enough to define zero velocities for gridpoints at the boundaries. On the other hand this is not obvious and some tests were provided that certain Itasca versions cannot get on with these commands. They had not given the same results if only one of the previously mentioned commands were used. So in the following examples I try to demonstrate that in 3DEC 4.10.121 (in demo and in full version) these commands are equal and only one of them is enough to use. To prove this phenomenon I run the whole column loaded with its self-weight and at first I use only the

BOUND XVEL, YVEL, ZVEL commands. I defined velocities with zero value for gridpoints at the lower plane of the base-block.

After that definition of geometry and material properties BOUNDARY commands can be seen. Now velocities were set to zero in every direction, after the gravitational loading was applied to the structure.

```

new
;base-block 60x60x20cm (blue colored block on Fig. 1)
;origin is defined in the middle of the upper surface of the base-block
  poly brick -0.3,0.3 -0.3,0.3 -0.2,0.
;
;block-levels made up with blocks sized: 10x10x20cm
;first level (blocks orientated in y-direction)
  poly brick -0.1,0. -0.1,0.1 0.,0.1
  poly brick 0.,0.1 -0.1,0.1 0.,0.1
;second level (blocks orientated in x-direction)
  poly brick -0.1,0.1 -0.1,0. 0.1,0.2
  poly brick -0.1,0.1 0.,0.1 0.1,0.2
;third level (blocks orientated in y-direction)
  poly brick -0.1,0. -0.1,0.1 0.2,0.3
  poly brick 0.,0.1 -0.1,0.1 0.2,0.3
;fourth level (blocks orientated in x-direction)
  poly brick -0.1,0.1 -0.1,0. 0.3,0.4
  poly brick -0.1,0.1 0.,0.1 0.3,0.4
;fifth level (blocks orientated in y-direction)
  poly brick -0.1,0. -0.1,0.1 0.4,0.5
  poly brick 0.,0.1 -0.1,0.1 0.4,0.5
;
;displaying of geometry
  plot block
  plot reset
  plot set dip 60. dd 210.
;
;mesh generation with 20cm average mesh-size
  gen edge 0.2
;
;material properties (sandstone):
;density: 2400 kg/m3
;E=19.3GPa Poisson-ratio: 0.38
;K=26.8GPa G=7GPa
  prop mat=1 dens=2400. k=2.68e10 g=7.e9
;
;joint properties (Coulomb-slip model)
;normal stiffness (1000 MPa/m) and shear stiffness (100 MPa/m)
;friction angle 25°
  change jcons=1
  prop jmat=1 jkn 1.e9 jks 1.e8 fric 25.
;
;supported on the lower surface of the base-block (velocity boundary with zero values in every direction)
  bound xvel=0.0 range x -0.31,0.31 y -0.31,0.31 z -0.21,-0.18
  bound yvel=0.0 range x -0.31,0.31 y -0.31,0.31 z -0.21,-0.18
  bound zvel=0.0 range x -0.31,0.31 y -0.31,0.31 z -0.21,-0.18
;
;for cycle is applied for rising the value of gravitational force
;this is necessary because some structures collapse if the gravitation is applied in one step
;"grav_lep" is the value that defines the number of cycles gravitation is applied
;the value of „gr” is rising with the value „g/grav_lep” in every cycle
;Calculation process is running until the structure reaches the desired accuracy

```

```

;this is measured by the ratio of unbalanced and applied forces
;now I set the limitvalue for 1e-5
  def gra
  grav_lep=10
  gr=0
  loop n (1,grav_lep)
  gr=gr-10/grav_lep
  command
  gravity 0.0,0.0,@gr
  solve rat 1.e-5
  end_command
  end_loop
  end;
;
hist zvel -0.3 -0.3 0
plot hist 1
;
;calling the „gra” command
  @gra

```

The working of supports can be determined according to the displacements of the gridpoints or the velocities of them.

At first let us see how can we check the displacements: the commands „hist zvel -0.3 -0.3 0” and „plot hist 1” were used to create the following graph (Fig. 23.):

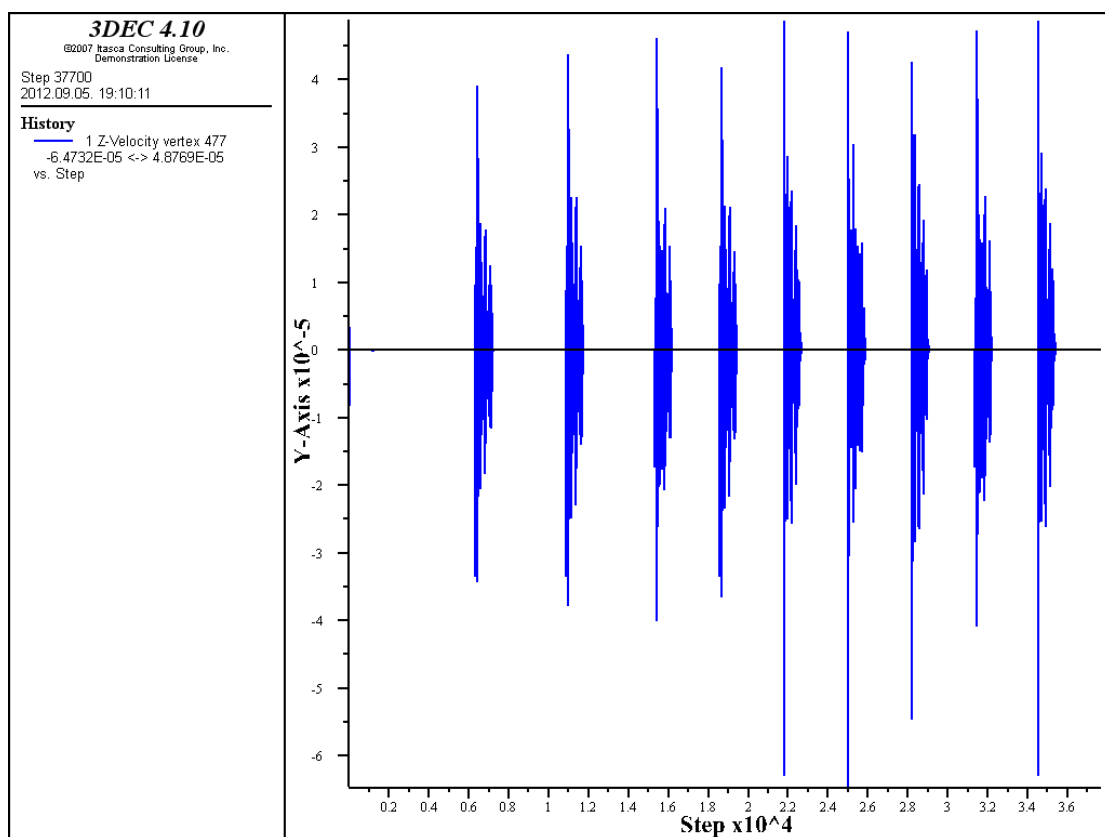


Fig. 23.: z-velocities of point: $x = -0.3$ $y = -0.3$ $z = 0.0$ [m] in case „boundary velocity” command was applied

As it can be seen on Fig. 23. the velocity of the registered gridpoint had reached the highest values at calculation steps when new gravitational load-step was applied to the structure. When the registered point was chosen to $(-0.3 -0.3 0)$ this „history” had given the following results (Fig. 24).

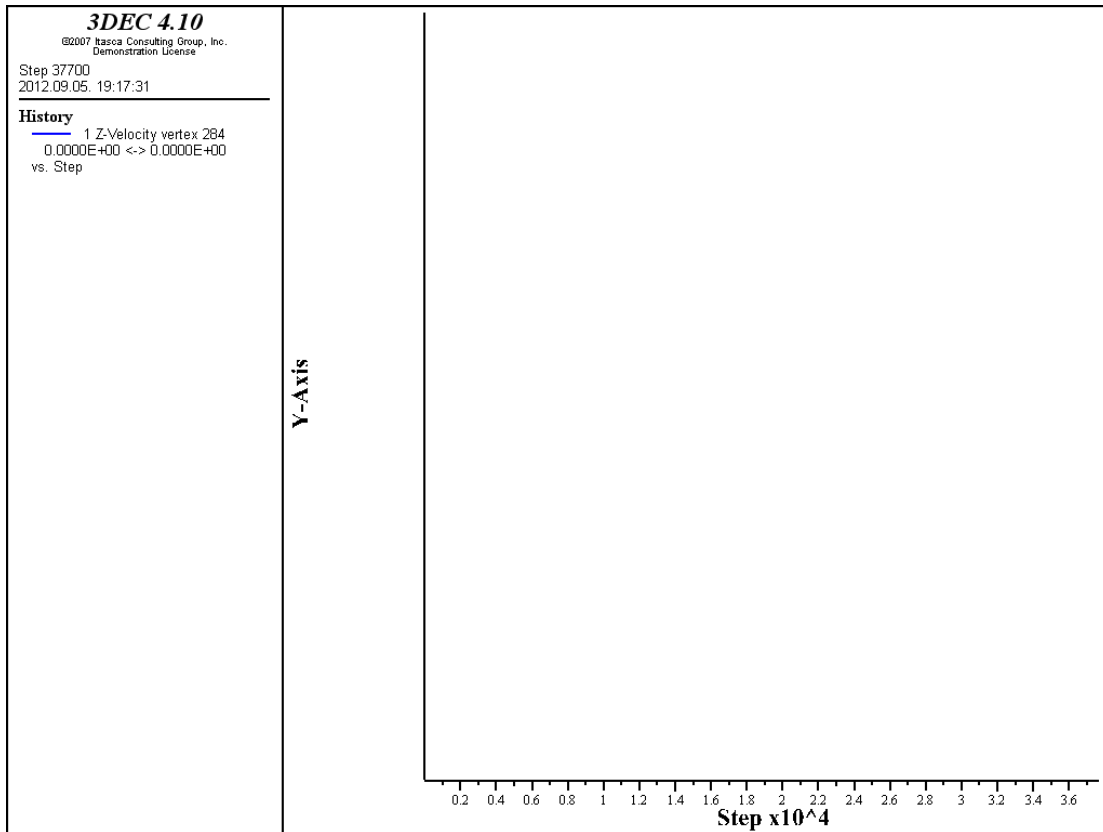


Fig. 24.: z-velocities of point: $x = -0.3$ $y = -0.3$ $z = -0.2$ [m] in case „boundary velocity” command was applied

Comparing the two graphs it can be announced that support can be defined only with the boundary velocity command, because velocities of the lower plane (-0.3, -0.3, -0.2) remained zero during the whole calculation process. On the other hand the upper plane of the base-block moved so the supports had to work.

There is a very useful command that can help identifying positions of vertexes: „list rf ve”. On Fig. 25. resulted chart of the listing can be seen. The first column of the chart contains the face where vertex can be found, the second contains the ID of the block, the third is the ID of the vertex and the last three columns are the original x, y and z coordinates of the vertexes.

Face	Block	Vertex	Position
1693	1203	1639	(-8.4475e-007, 1.0000e-001, 9.9988e-002)
1693	1203	1358	(4.6843e-008, 1.0000e-001, -1.1471e-005)
1693	1203	1314	(9.1750e-008, -1.0000e-001, -1.1708e-005)
1693	1203	1507	(-7.9937e-007, -1.0000e-001, 9.9988e-002)
1725	1203	1639	(-8.4475e-007, 1.0000e-001, 9.9988e-002)
1725	1203	1578	(-1.0000e-001, 1.0000e-001, 9.9988e-002)
1725	1203	1402	(-1.0000e-001, 1.0000e-001, -1.2361e-005)
1725	1203	1358	(4.6843e-008, 1.0000e-001, -1.1471e-005)
267	217	284	(-3.0000e-001, -3.0000e-001, -2.0000e-001)
267	217	328	(3.0000e-001, -3.0000e-001, -2.0000e-001)
267	217	372	(3.0000e-001, 3.0000e-001, -2.0000e-001)
267	217	416	(-3.0000e-001, 3.0000e-001, -2.0000e-001)
455	217	284	(-3.0000e-001, -3.0000e-001, -2.0000e-001)
455	217	477	(-3.0000e-001, -3.0000e-001, -2.2542e-008)
455	217	521	(3.0000e-001, -3.0000e-001, -2.2980e-008)
455	217	328	(3.0000e-001, -3.0000e-001, -2.0000e-001)

Fig. 25.: list of some vertexes of the structure

During the probe of several models I realised that the software generates a totally different mesh in every new running of the code. This is a very important moment, because it indicates

that after every new running the user should check the coordinates of the mesh-grids if the identification of gridpoint locations is relevant in aspect of calculation.

For locating coordinates of gridpoints the following command can be used:

```
list grid location range z -0.3,-0.18
```

Or a lot of command can give these coordinates too, even if the main purpose of them is not listing of the original coordinates. For example:

```
list gridpoint displacement range z -0.3,-0.18
```

gives the same coordinates as the previous command, but after the original coordinates displacements of gridpoints are listed too. See the following results: these tables (Fig. 26.) were all generated with the same geometry and the same average mesh-size (0.2 m), in spite of that every listing resulted different coordinates for gridpoints 9036, 8997, 8958, 8919.

```
3dec>list grid disp range z -0.3,-0.18
```

block	grid-point	x	y	z	dx	dy	dz
217	9036	1.076E-01	1.073E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	8997	1.074E-01	-1.075E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	8958	-1.078E-01	1.073E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	8919	-1.074E-01	-1.082E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	8022	-3.000E-01	1.000E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	7983	-3.000E-01	-1.000E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	7866	-1.000E-01	3.000E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	7827	1.000E-01	3.000E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	7788	3.000E-01	1.000E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	7749	3.000E-01	-1.000E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	7710	1.000E-01	-3.000E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	7671	-1.000E-01	-3.000E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	416	-3.000E-01	3.000E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	372	3.000E-01	3.000E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	328	3.000E-01	-3.000E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	284	-3.000E-01	-3.000E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00

```
3dec>list grid disp range z -0.3,-0.18
```

block	grid-point	x	y	z	dx	dy	dz
217	9036	-1.065E-01	1.074E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	8997	1.087E-01	1.065E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	8958	1.088E-01	-1.082E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	8919	-1.080E-01	-1.085E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	8022	-3.000E-01	1.000E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	7983	-3.000E-01	-1.000E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	7866	-1.000E-01	3.000E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	7827	1.000E-01	3.000E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	7788	3.000E-01	1.000E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	7749	3.000E-01	-1.000E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	7710	1.000E-01	-3.000E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	7671	-1.000E-01	-3.000E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	416	-3.000E-01	3.000E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	372	3.000E-01	3.000E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	328	3.000E-01	-3.000E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00
217	284	-3.000E-01	-3.000E-01	-2.000E-01	0.000E+00	0.000E+00	0.000E+00

```

3dec>list grid disp range z -0.3,-0.18
  block  grid-point    x          y          z          dx          dy          dz
  217    9036  1.087E-01 -1.093E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    8997 -1.073E-01  1.068E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    8958 -1.054E-01 -1.084E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    8919  1.067E-01  1.070E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    8022 -3.000E-01  1.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    7983 -3.000E-01 -1.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    7866 -1.000E-01  3.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    7827  1.000E-01  3.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    7788  3.000E-01  1.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    7749  3.000E-01 -1.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    7710  1.000E-01 -3.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    7671 -1.000E-01 -3.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217     416 -3.000E-01  3.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217     372  3.000E-01  3.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217     328  3.000E-01 -3.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217     284 -3.000E-01 -3.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00

3dec>list grid disp range z -0.3,-0.18
  block  grid-point    x          y          z          dx          dy          dz
  217    9036  1.081E-01  1.066E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    8997  1.077E-01 -1.080E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    8958 -1.082E-01  1.085E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    8919 -1.064E-01 -1.075E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    8022 -3.000E-01  1.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    7983 -3.000E-01 -1.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    7866 -1.000E-01  3.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    7827  1.000E-01  3.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    7788  3.000E-01  1.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    7749  3.000E-01 -1.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    7710  1.000E-01 -3.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    7671 -1.000E-01 -3.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217     416 -3.000E-01  3.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217     372  3.000E-01  3.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217     328  3.000E-01 -3.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217     284 -3.000E-01 -3.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00

```

Fig. 26.: Change of mesh-grids' location according to different runnings

In the previous examples I used BOUND VEL command to support the lower plane of the base-block. Now I demonstrate the phenomenon mentioned at first, namely: „FIX and BOUND VEL commands give the same results individually they don't have to be used together”. As it can be seen on Fig. 27. displacements at the lower plane of the base block were zero in case only FIX command was used.

```

3dec>list grid disp range z -0.22,-0.15
  block  grid-point    x          y          z          dx          dy          dz
  217    9036  1.077E-01  1.071E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    8997 -1.078E-01  1.073E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    8958  1.076E-01 -1.071E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    8919 -1.077E-01 -1.077E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    8022 -3.000E-01  1.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    7983 -3.000E-01 -1.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    7866 -1.000E-01  3.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    7827  1.000E-01  3.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    7788  3.000E-01  1.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    7749  3.000E-01 -1.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    7710  1.000E-01 -3.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217    7671 -1.000E-01 -3.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217     416 -3.000E-01  3.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217     372  3.000E-01  3.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217     328  3.000E-01 -3.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00
  217     284 -3.000E-01 -3.000E-01 -2.000E-01  0.000E+00  0.000E+00  0.000E+00

```

Fig. 27.: Displacements at the lower plane of the base block in case support was defined by FIX command

On the other hand it is true that in case velocity of gridpoints were set to a certain value earlier than FIX command is applied we can fix only the previously defined velocity. So in these cases at first the BOUND VEL command has to be applied and zero velocity is necessary to be defined. FIX command automatically refers to every direction, while in case of BOUND VEL command velocities can be changed separately in each direction. So this indicates that in most cases the two commands has to be used together. In the previously demonstrated examples the applied supports were fixed in every direction (velocities were set to zero in x, y and z directions).

In the following section a script is demonstrated that **generates base-settlement. At first out of the four corner of the lower plane of the base-block only one moves, the other three is fixed.** To specify the movement of the corner I defined a „loop” cycle in the „FISH code”. The final displacement is -0.01 m, velocity is -0.1 m/s. The „gp_near” command registrates the actual coordinates of 0.3,-0.3,-0.2 point and stores it in „ph” variable. The moved point’s velocity is equals to variable „v” (@v), so when calculation starts the points starts to move. In order to the cycle stop at the proper displacement value an „if” command had to be applied. Z-displacments of „ph” gridpoint is stored in „elm” variable with „gp_zdis(ph)” command, as a result „elm” is giving the actual displacement in every calculation cycles. This value is enough to decide whether value of „elm” reached the desired displacement „s”. If „elm” reaches or higher than „s”, velocity of moving gridpoint is turned into zero. The whole monitoring cycle with „loop” command runs 100 times, but if the conditions sufficed earlier the cycle stops. During one loop the software makes 100 solve-steps (cycles), this is defined with the „cycle 100” command.

```

;uneven settlement with one moving gridpoint
;velocity of fixed gridpoints choosen to zero
    bound zvel=0. range x -0.31,-0.29 y 0.31,0.29 z -0.21,-0.1
    bound zvel=0. range x 0.31,0.29 y 0.31,0.29 z -0.21,-0.1
    bound zvel=0. range x -0.31,-0.29 y -0.31,-0.29 z -0.21,-0.1
;fixing boundary velocities
    fix range x -0.31,-0.29 y 0.31,0.29 z -0.21,-0.1
    fix range x 0.31,0.29 y 0.31,0.29 z -0.21,-0.1
    fix range x -0.31,-0.29 y -0.31,-0.29 z -0.21,-0.1
;
;definition of displacement of the moving gridpoint
def tám
    v=-0.1 ;velocity of gridpoint
    s=-0.01 ;prescribed displacement
;
    ph=gp_near(0.3,-0.3,-0.2)
;
    command
        bound zvel=@v range x 0.33,0.27 y -0.33,-0.27 z -0.27,-0.18
    end_command
;
    loop n (1,100)
        elm=gp_zdis(ph)
;
        if elm > s then
            command
                cycle 100
            end_command
        end_if
    end_loop
;
    command

```



```

bound zvel=0. range x 0.33,0.27 y -0.33,-0.27 z -0.27,-0.18
end_command
end;
;calling „tám” command
@tám

```

The previously mentioned statement is come p again, namely: the gridpoints always has to be identified with ranges because the software cannot come up with direct coordinates. The main reason of it is the constant moving of gridpoints. Unfortunately the represented „gp_near” command, that identifies gridpoints with their identification number (ID) works only within FISH code.

The resulted displacements can be seen on Fig. 28..

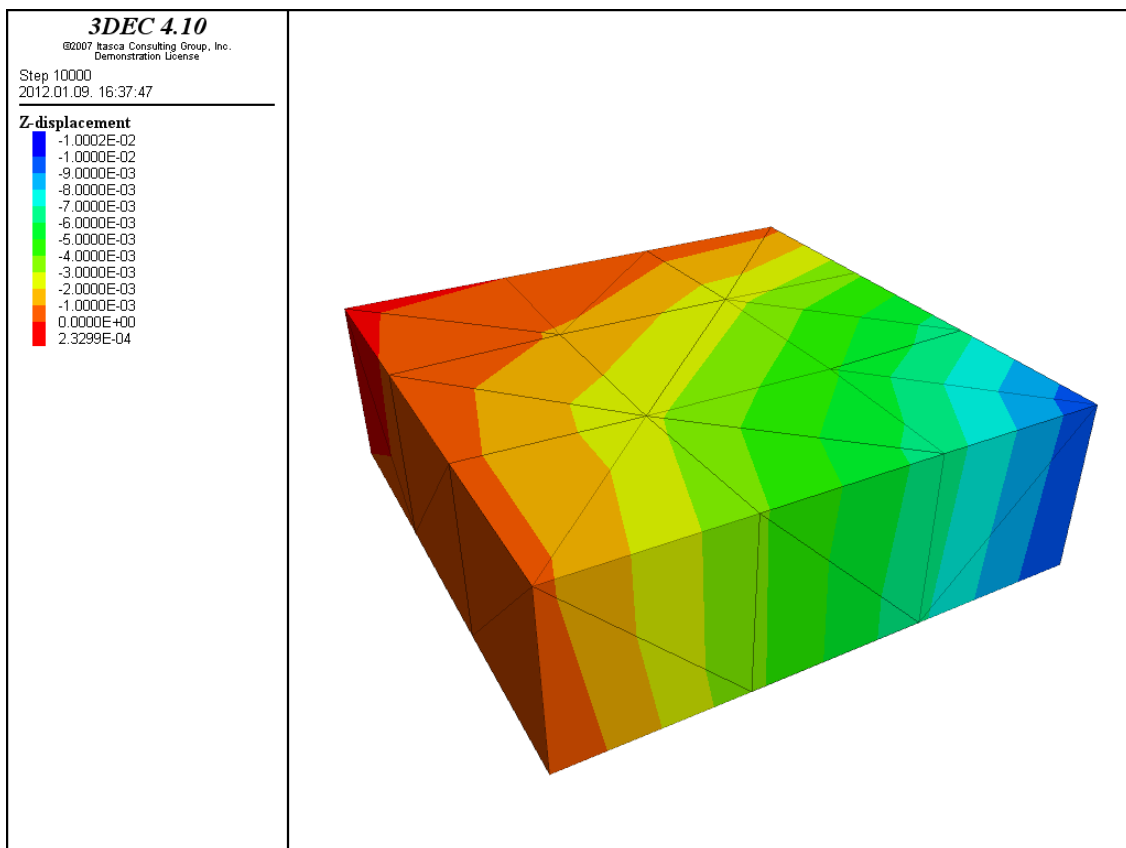


Fig. 28.: displacements in case uneven settlement applied for the model

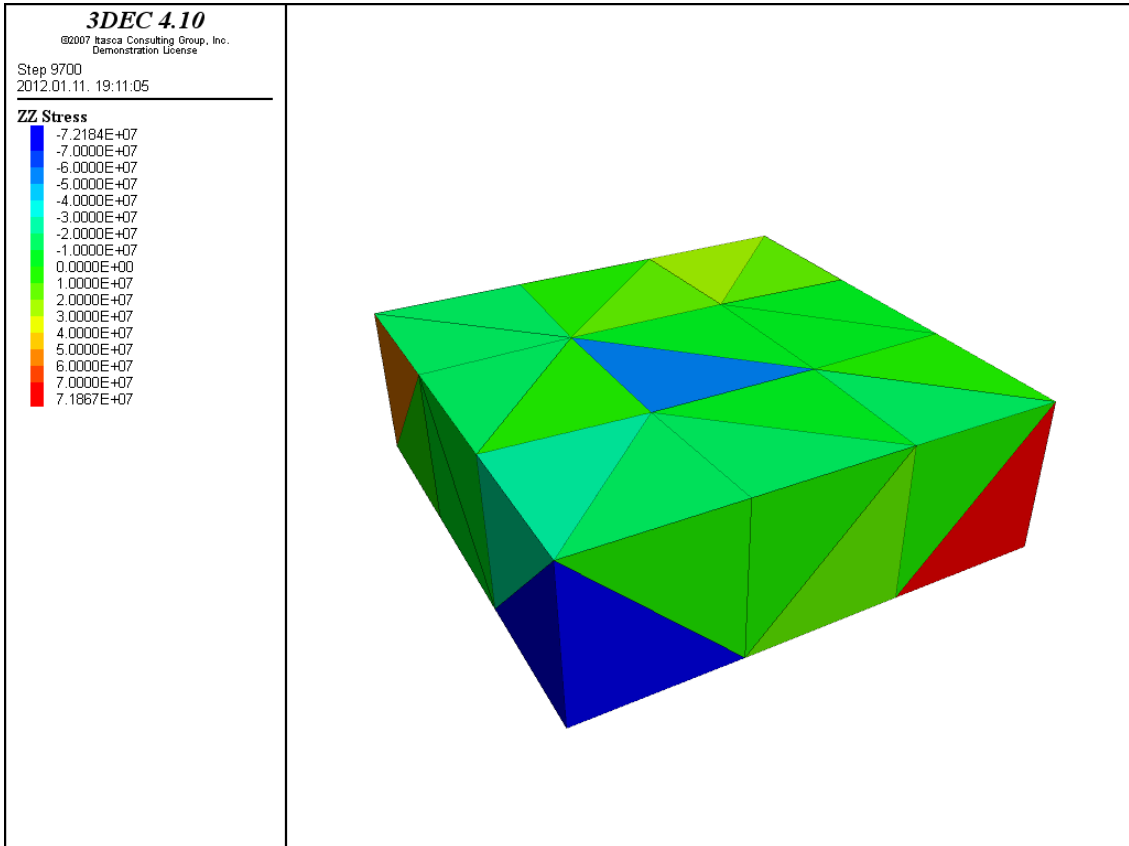


Fig. 29.: σ_{zz} normal stresses in z-direction in case uneven settlement applied for the model

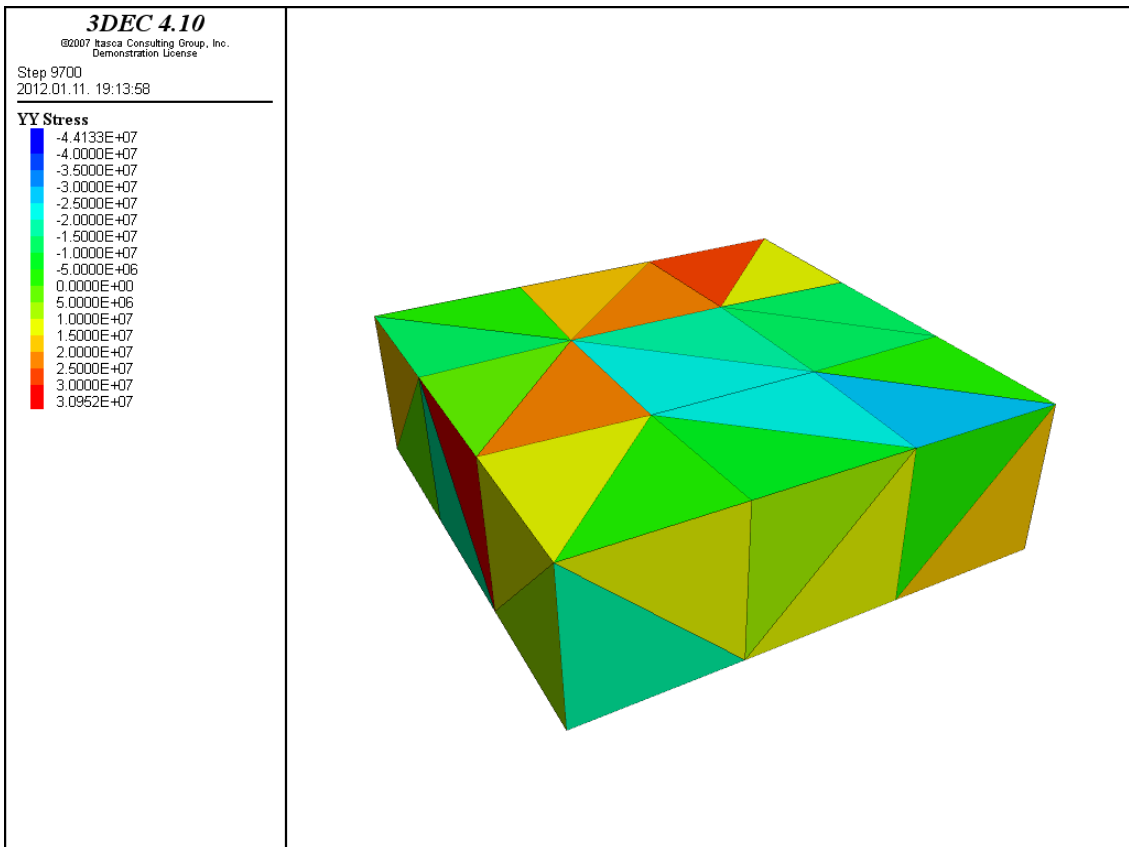


Fig. 30.: σ_{yy} normal stresses in y-direction in case uneven settlement applied for the model

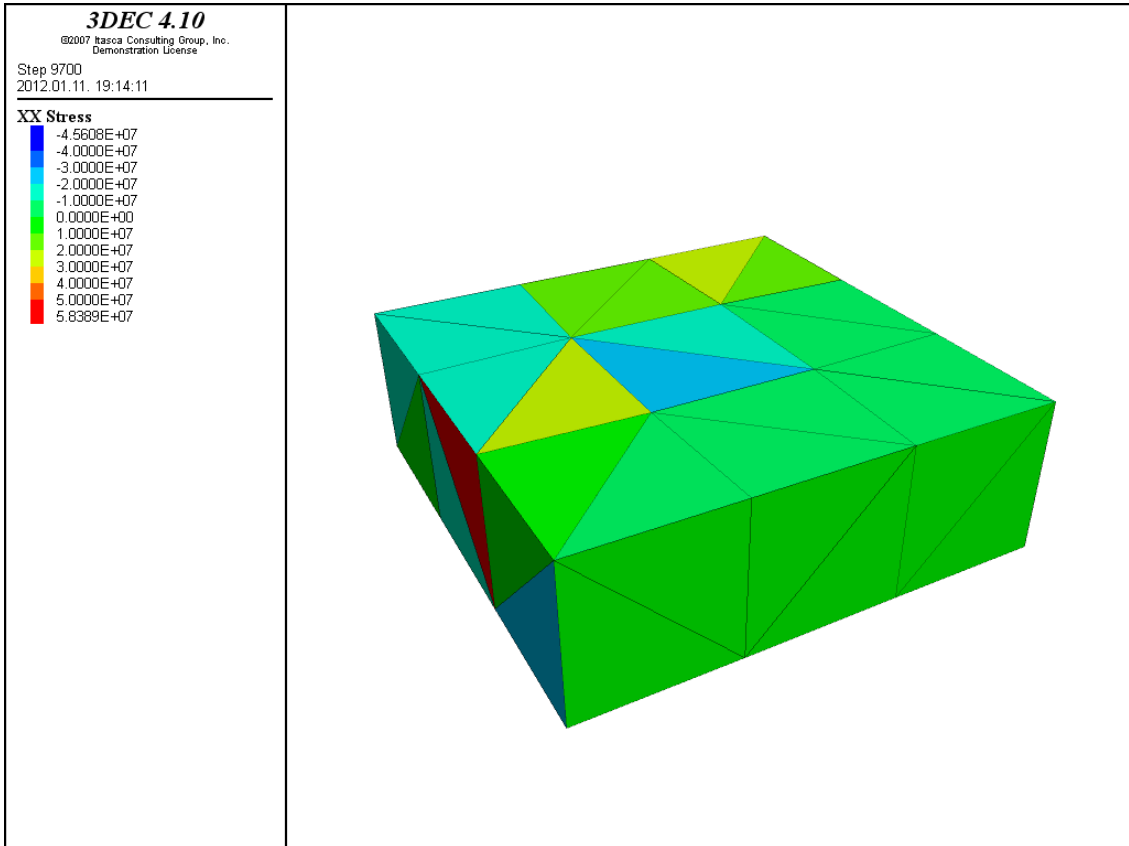


Fig. 31.: σ_{xx} normal stresses in x-direction in case uneven settlement applied for the model

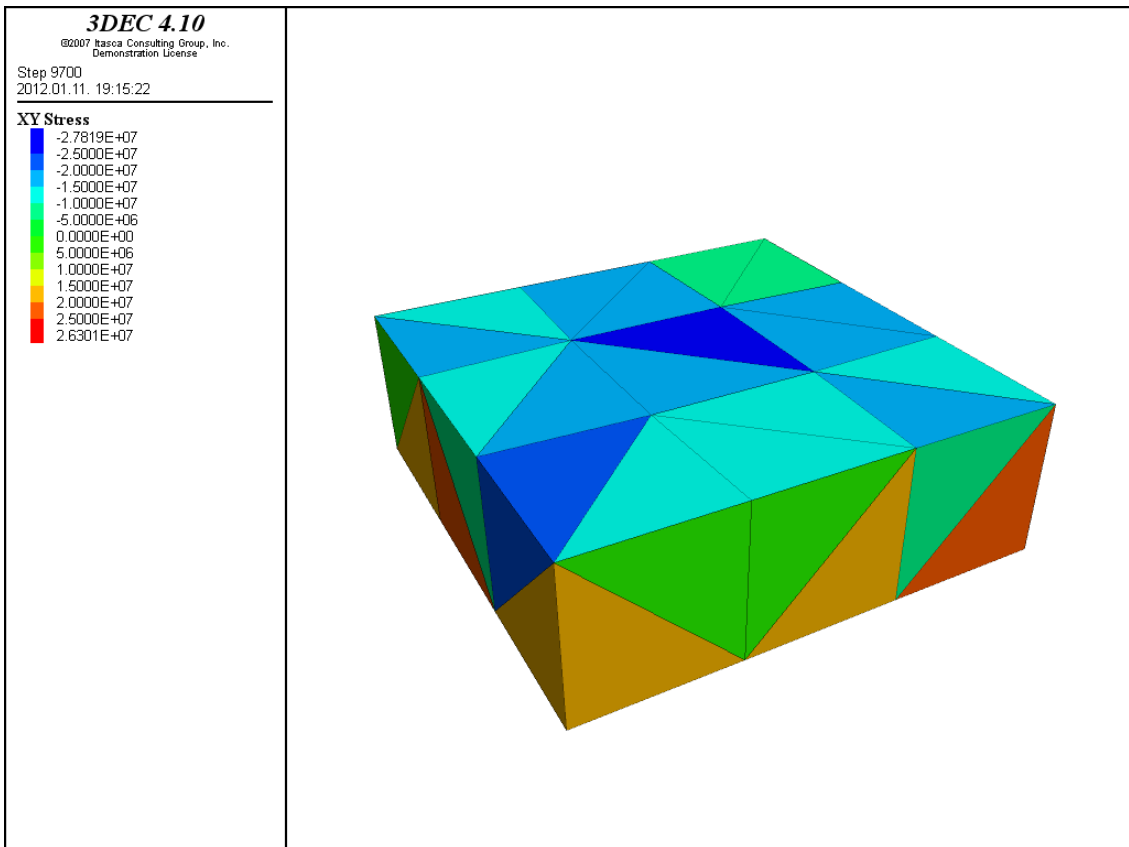


Fig. 32.: σ_{xy} shear stresses in z-direction in case uneven settlement applied for the model

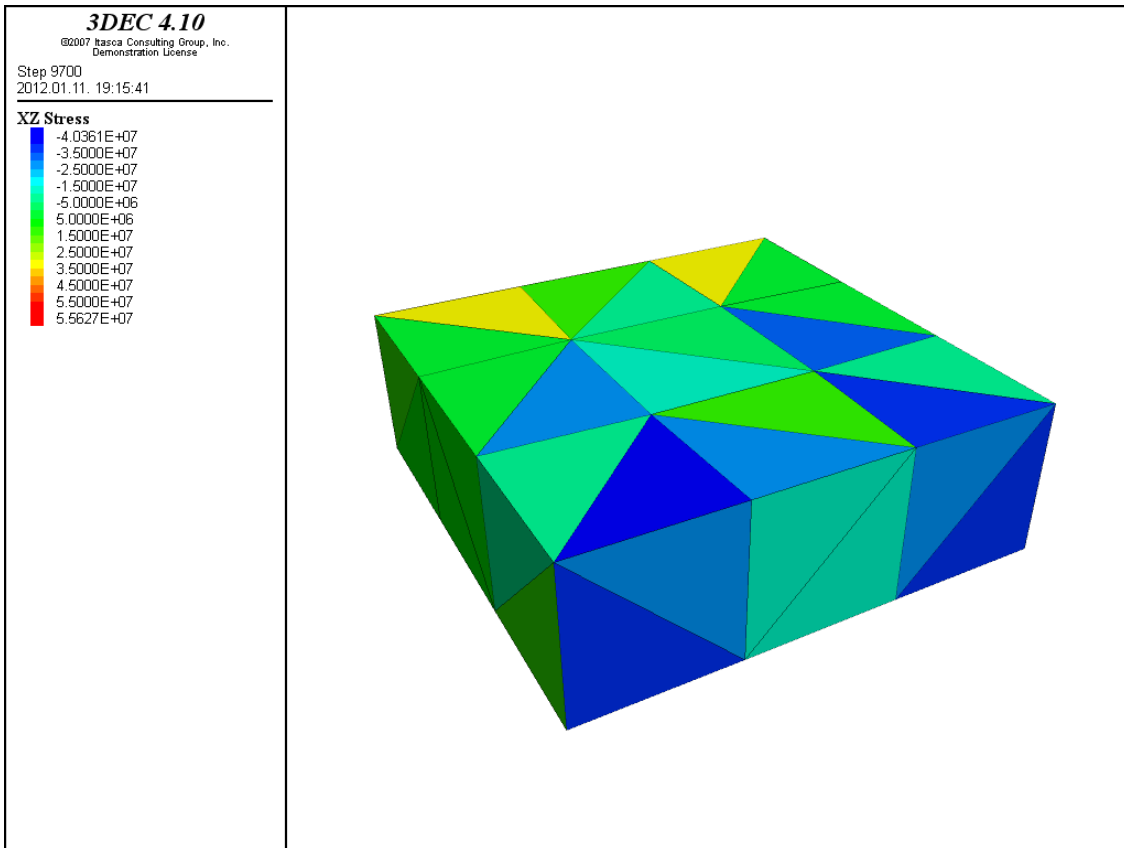


Fig. 33.: σ_{xz} shear stresses in case uneven settlement applied for the model

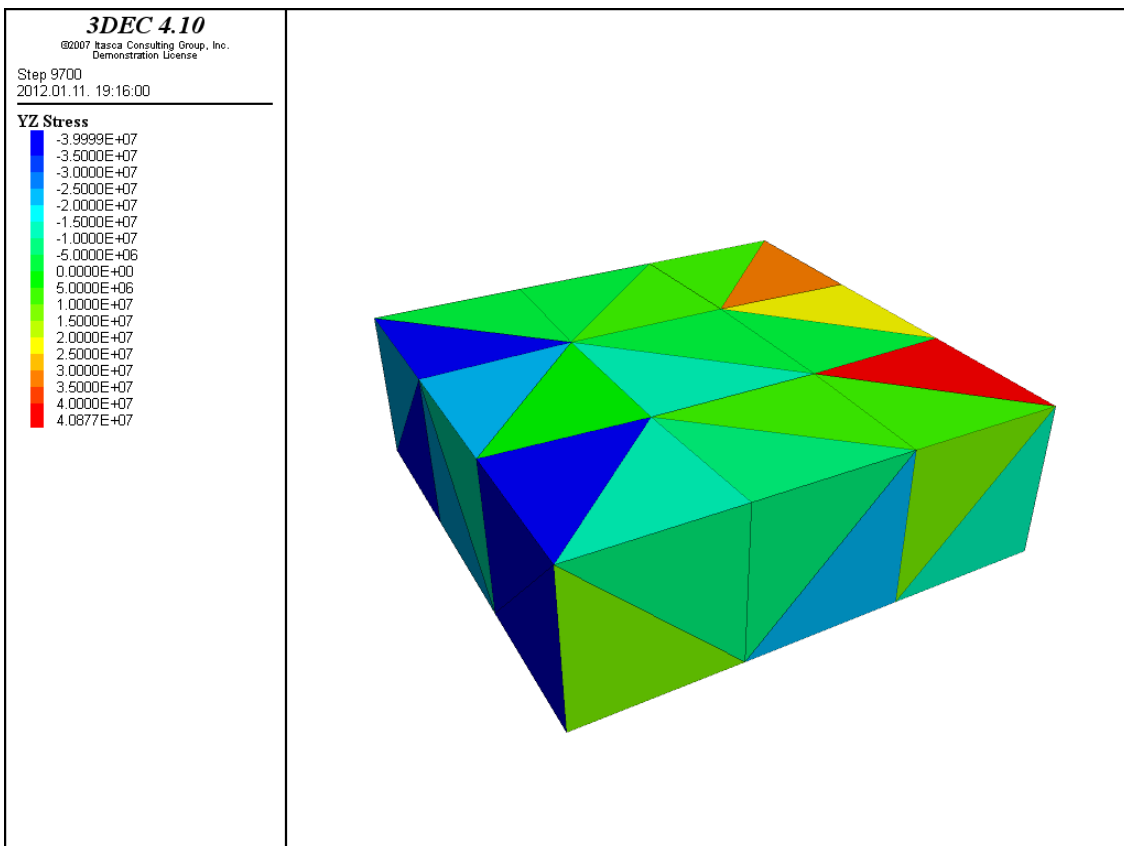


Fig. 34.: σ_{yz} shear stresses in case uneven settlement applied for the model

On Fig. 29.-34. normal and shear stresses of base-block can be seen, the pictures were taken with the following command:

```
plot bcontour syz above au
```

Command „bcontour” is similar to the previously used „contour” command, the automatic color settings were applied („above au”). „Bcontour” command displays the choosen property with the most expressing color characterizing the investigated zone. Now the displayed values are sxx, syy, szz, sxy, sxz, syz stresses.

Studying Fig. 29. the expected behaviour is ocured, at the moved corner tensile stresses were acting while at the neighbouring two corners compression was registered. At the corner opposite to the moved one, tension was ocured too, because the gridpoint tried to rise up but it was fixed. Studying the results by take into account the displacements, behaviour of the structure is proper.

According to other figures the zones show a high variety of stress values. In case of a more detailed mesh the interaction between zones could be expressed. Stress values are nearly the same, but the highest values were registered at normal stresses acting in z-direction.

Using „list grid disp range z -0.21,-0.18” command displacements of lower part of base-block were listed. Resulting table is the following:

```
3dec>list grid disp range z -0.21,-0.18
```

block	grid-point	x	y	z	dx	dy	dz
217	1960	1.069E-01	1.082E-01	-2.021E-01	-6.532E-04	9.480E-04	-2.087E-03
217	1921	1.064E-01	-1.065E-01	-2.042E-01	-1.032E-03	1.020E-03	-4.215E-03
217	1882	-1.084E-01	1.080E-01	-2.007E-01	-5.431E-04	6.944E-04	-6.998E-04
217	1843	-1.084E-01	-1.075E-01	-2.023E-01	-1.043E-03	6.967E-04	-2.349E-03
217	1570	-3.005E-01	1.006E-01	-1.998E-01	-5.235E-04	5.939E-04	1.829E-04
217	1531	-3.010E-01	-9.929E-02	-2.002E-01	-9.939E-04	7.101E-04	-1.877E-04
217	1414	-1.003E-01	3.009E-01	-1.993E-01	-2.614E-04	9.144E-04	6.926E-04
217	1375	9.955E-02	3.008E-01	-2.001E-01	-4.512E-04	8.275E-04	-8.260E-05
217	1336	2.994E-01	1.010E-01	-2.032E-01	-5.934E-04	9.784E-04	-3.235E-03
217	1297	2.989E-01	-9.885E-02	-2.062E-01	-1.082E-03	1.151E-03	-6.176E-03
217	1258	9.881E-02	-2.989E-01	-2.063E-01	-1.194E-03	1.123E-03	-6.319E-03
217	1219	-1.012E-01	-2.993E-01	-2.037E-01	-1.150E-03	6.557E-04	-3.666E-03
217	416	-3.002E-01	3.006E-01	-2.000E-01	-1.838E-04	6.017E-04	0.000E+00
217	372	2.996E-01	3.009E-01	-2.000E-01	-4.109E-04	9.420E-04	0.000E+00
217	284	-3.015E-01	-2.998E-01	-2.000E-01	-1.534E-03	2.262E-04	0.000E+00

Fig. 35.: Displacements of the lower part of the base-block in case uneven settlements were set to it

In the following sections I am studying displacements of the gridpoints lying on edges leaning to the moved corner. The edge parallel to x-axis contains the following gridpoints (IDs enumerated in order approaching to the moved corner): 284, 1219, 1258, 328. The edge parallel to y-axis contains the following gridpoints (IDs enumerated in order approaching to the moved corner): 372, 1336, 1297, 328.

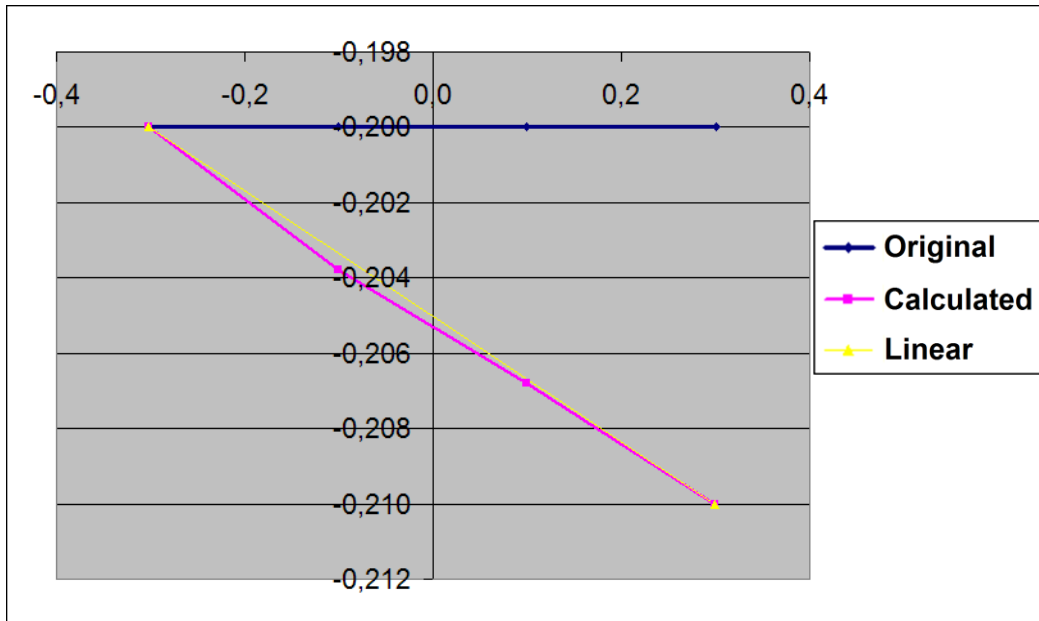


Fig. 36.: Displacements of the gridpoints parallel to x-axle (284, 1219, 1258, 328) contained by lower part of the base-block in case uneven settlements were set to them

On Fig. 36. displacements of the gridpoints parallel to x-axle contained by lower part of the base-block can be seen. Thick blue line demonstrates the original (unmoved) state, every gridpoint is horizontal and lying at $z=-0.2$ coordinate. Pink (purple) polyline represents the calculated displacements of edge, yellow line can be considered as reference, it shows linear displacements (without internal gridpoints). According to the graph displacements of gridpoints lying on the edge is not linear. On Fig. 37. displacements of edge parallel to y-axle are represented. Three of the gridpoints represented on Fig. 36. and 37. were created by me, because they build up the geometry: 284, 372, 328, but the followings were automatically created by the software, during mesh generation: 1219, 1258, 1336, 1297 („gen” command).

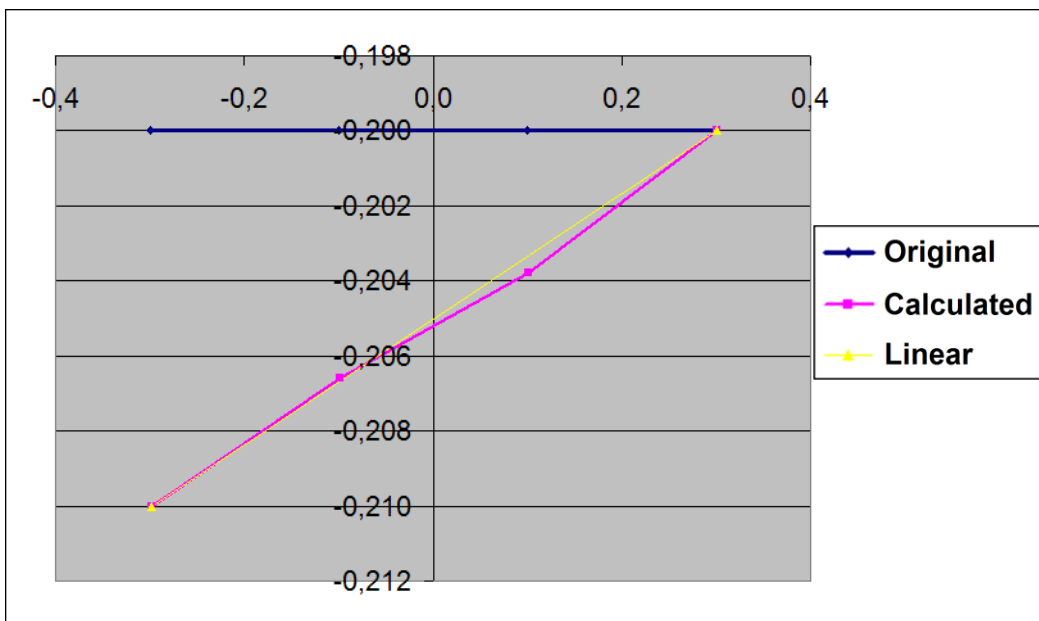


Fig. 37.: Displacements of the gridpoints parallel to y-axle (372, 1336, 1297, 328) contained by lower part of the base-block in case uneven settlements were set to them

On Fig. 38. and 39. displacements are represented in case mesh was generated with a higher density (the original 0.2 m average mesh-size was changed to 0.1 m). According to Fig. 38. and 39. the settlement of the moving-corner is more concentrated than in case of rough mesh (0.2 m) furthermore the change of displacements is refined.

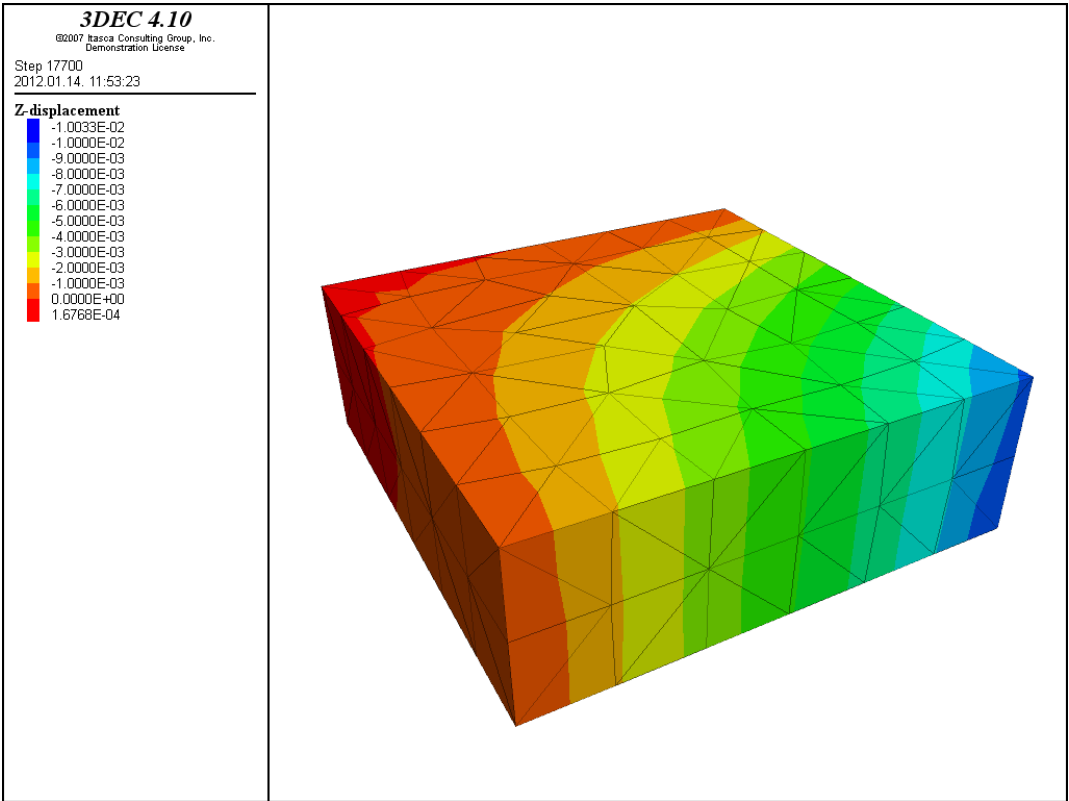


Fig. 38.: z-displacements of base-block in case uneven settlement was applied for the model, mesh was generated with 0.1 m average mesh-size

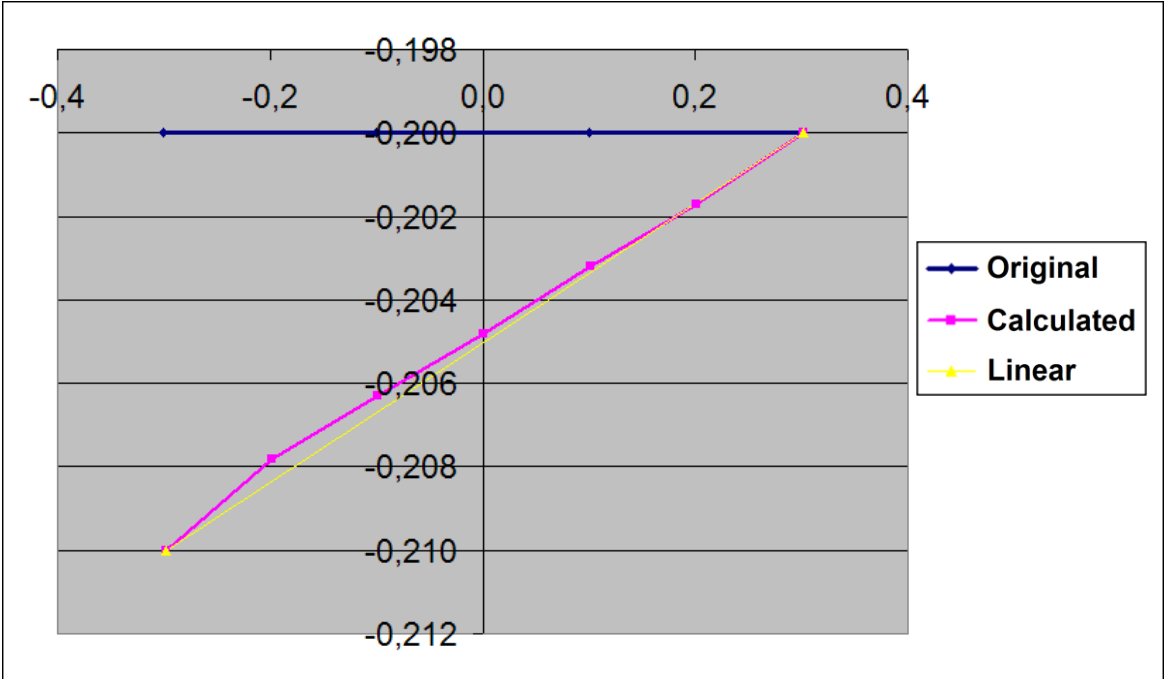


Fig. 39.: Displacements of the gridpoints parallel to y-axe (372, 1570, 1531, 1492, 1453, 1414, 328) contained by lower part of the base-block in case uneven settlements were set to them, mesh was generated with 0.1 m average mesh-size

In case velocity boundary is **applied for four ranges containing the whole area of the lower part of base-block not for the gridpoints**, the script is the following:

```

;uneven base-settlement movement is defined with ranges containing the whole plane of the base-block
;velocities of fixed regions is defined to zero
    bound zvel=0. range x -0.31,0. y 0.31,0. z -0.21,-0.18
    bound zvel=0. range x 0.31,0. y 0.31,0. z -0.21,-0.18
    bound zvel=0. range x -0.31,0. y -0.31,0. z -0.21,-0.18
;fixing of boundary ranges
    fix range x -0.31,0. y 0.31,0. z -0.21,-0.18
    fix range x 0.31,0. y 0.31,0. z -0.21,-0.18
    fix range x -0.31,0. y -0.31,0. z -0.21,-0.18
;
;applying movement for the region
def tam
    v=-0.1    ;velocity of region
    s=-0.01   ;prescribed displacement
;
    ph=gp_near(0.3,-0.3,-0.2)
;
    command
        bound zvel=@v range x 0.31,0. y -0.31,0. z -0.27,-0.18
    end_command
;
    loop n (1,100)
        elm=gp_zdis(ph)
;
        if elm > s then
            command
                cycle 100
            end_command
        end_if
    end_loop
;
    command
        bound zvel=0. range x 0.31,0. y -0.31,0. z -0.27,-0.18
    end_command
end;

```

Ranges were defined only for the lower plane of the base-block and they split the block into four equal parts. The parts are connecting in the center of the block. On Fig. 40. displacements are represented with use of „plot contour” command. According to the results were announced in the previous section the gradual change is not registered in case of ranges including the whole plane of the block. The prescribed displacements are acting on bigger areas and because of this, settlements are concentrated on an exact range. Actually these results are not surprising, the influence of changed model-parameters can be seen more impressive on Fig. 41. and 42. where phenomenon of shear failure occurs. The graphs are containing displacements of gridpoints parallel to x- and y-axes. Studying the results, in the middle of the block, displacements are evolving concentrated and changing linearly. At ranges where specified displacements acting, the graph is horizontal at the exact values. This phenomenon occurs more intense if the average mesh-size is reduced see Fig. 43. and 44.

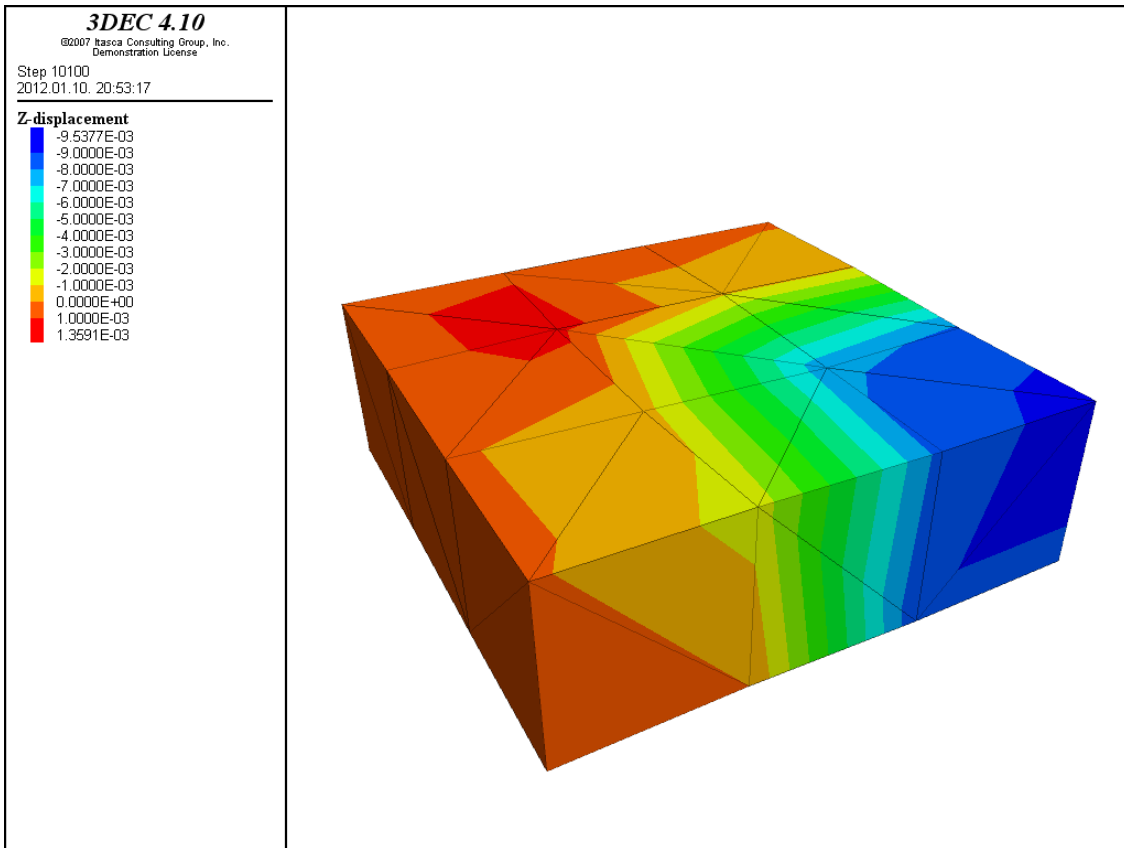


Fig. 40.: z-displacements in case uneven settlement was applied for four ranges containing the whole area of the lower part of base-block

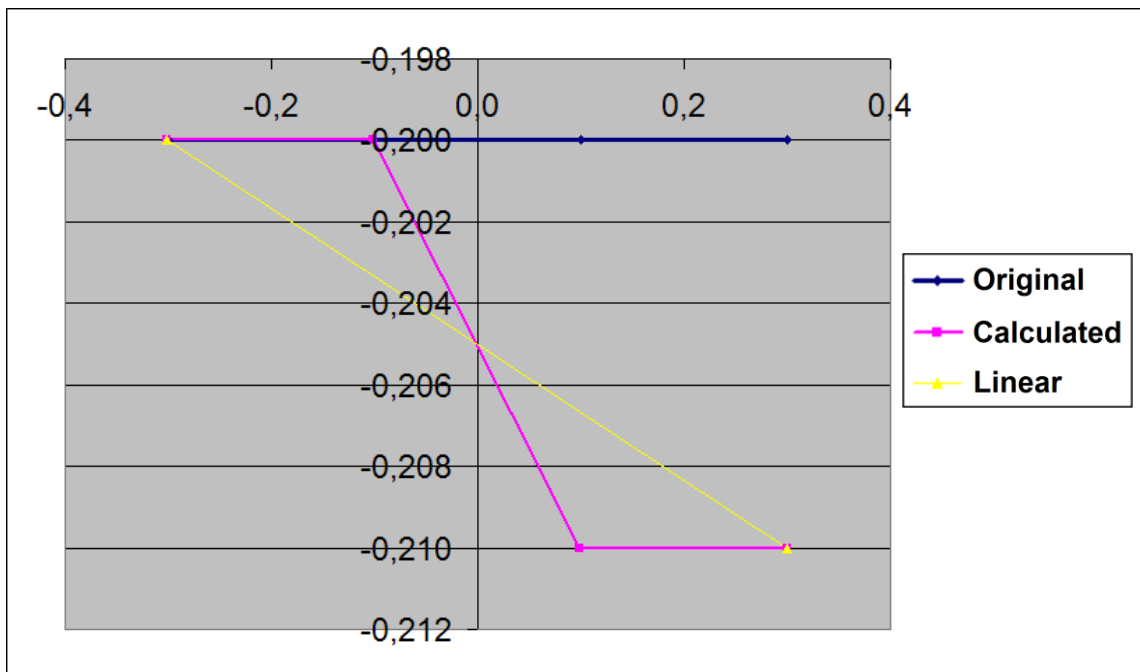


Fig. 41.: Displacements of the gridpoints parallel to x-axle (284, 1219, 1258, 328) in case uneven settlement was applied for four ranges containing the whole area of the lower part of base-block

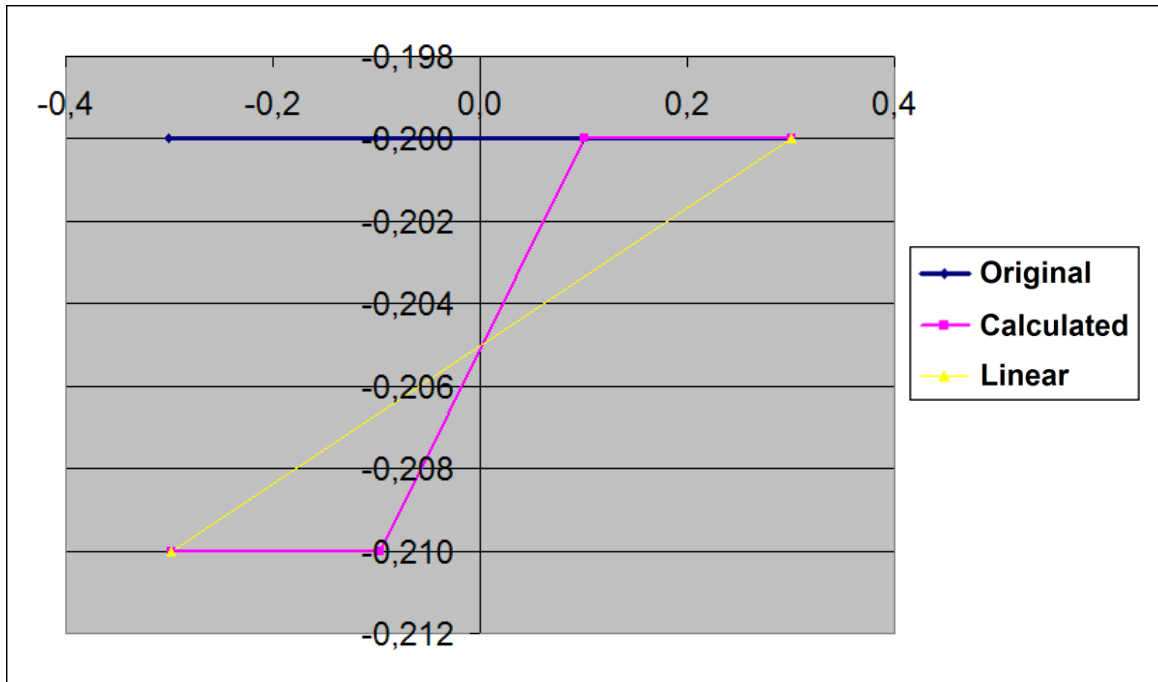


Fig. 42.: Displacements of the gridpoints parallel to y-axis (372, 1336, 1297, 328) in case uneven settlement was applied for four ranges containing the whole area of the lower part of base-block

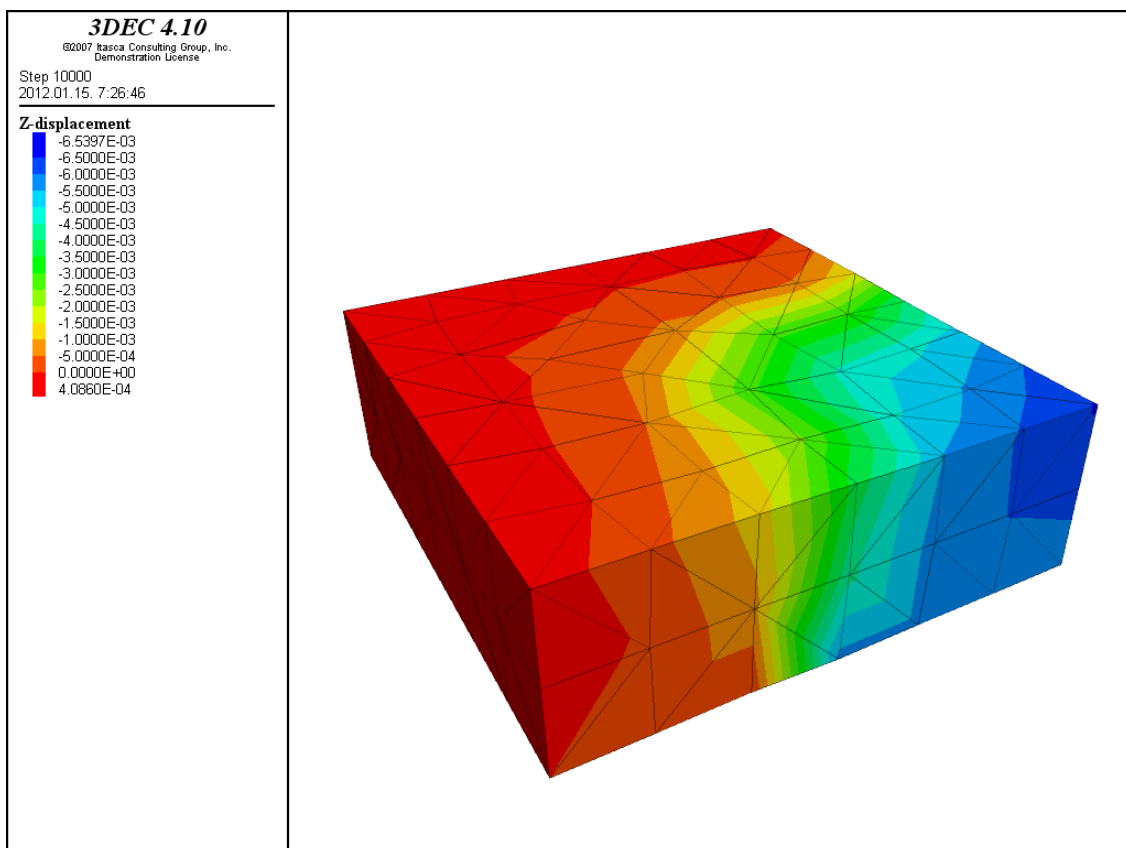


Fig. 43.: z-displacements in case uneven settlement was applied for four ranges containing the whole area of the lower part of base-block, mesh was generated with 0.1 m average mesh-size

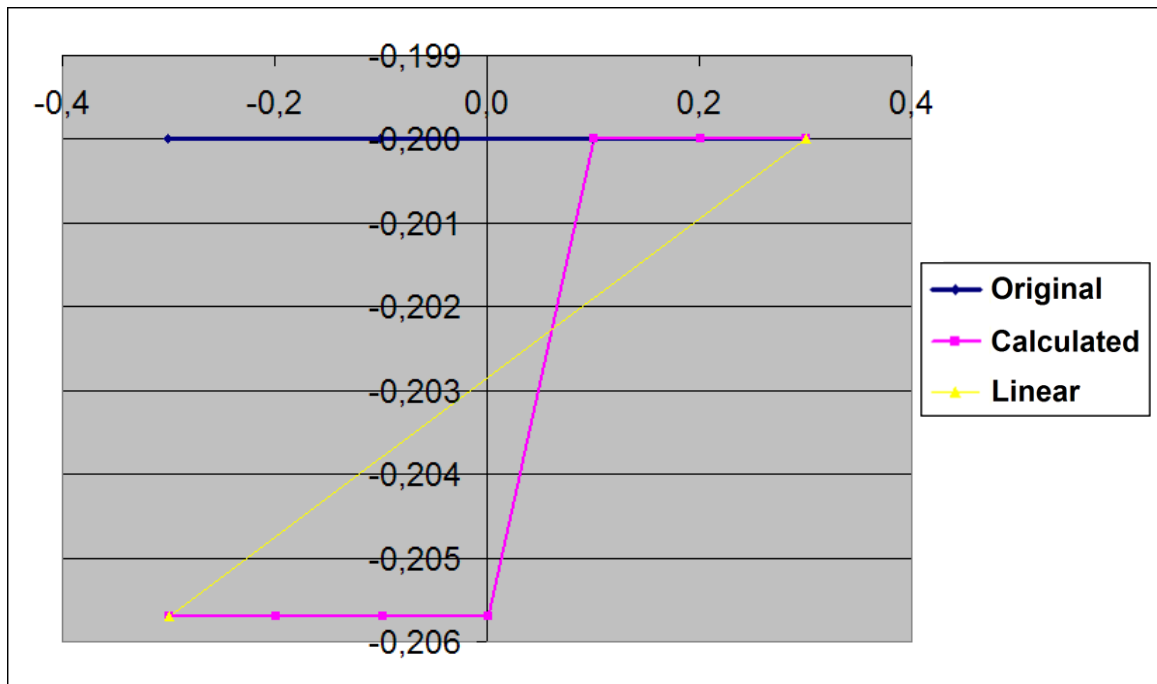


Fig. 44.: Displacements of the gridpoints parallel to y-axle (372, 1570, 1531, 1492, 1453, 1414, 328) in case uneven settlement was applied for four ranges containing the whole area of the lower part of base-block, mesh was generated with 0.1 m average mesh-size

So on Fig. 43. and 44. the average mesh-size were changed from 0.2 m to 0.1 m. Comparing the results the length of linear part where displacements occurs had decreased, this tendency is indicating that the more-fined-mesh occurs the more concentrated displacements and finally a shear-plane develops. So in case supports are defined with ranges covering the whole base-block, settlements occur in block shape instead of gradual change.

In the following scripts **support-ranges are covering** the whole lower plane of the base-block, like in the previous section. The difference is the following: ranges are containing **the whole block**, so **they were extended in vertical (z) direction up to the top of the base-block**. Results were show that displacements cannot develop in case fixed ranges are containing the lower and upper plane of the block. In case the mesh is generated with 0.2 m mesh size gridpoints are located on two planes (lower $z=-0.2$ and upper $z=0.0$ plane of the block, defined by geometry). If the upper boundary of the range is defined with a few mm-s under the upper plane displacements were occurred.

Another script was ran with 0.1 m mesh size, this means that an internal level has developed (besides the lower and upper plane of the block) where gridpoints are located. The new plane can be found at $z=-0.1$ height. According to the results displacements can develop if the ranges do not contain the gridpoints at level $z=-0.1$. Studying the results availability of ranges depends on the mesh-density and a more intense mesh (mesh with smaller mesh-size) would cause that displacements can develop only in case ranges were defined for the lower plane of base-block.

Following script is defines a base-settlement where **three corner of the lower plane of base-block is moving and only one is fixed. Velocity of gridpoint opposite to the fixed one is 0.1 m/s, while the other two gridpoints moving 0.05 m/s fast.**

The described script is the following:

```

;"stress-free" base-block, only one corner of the block is fixed (velocity of gridpoint is zero)
;the other three corners are moving. Velocity of two opposite moving-corners
;is half of the third moving gridpoint.
    bound zvel=0. range x -0.31,-0.29 y 0.31,0.29 z -0.21,-0.1
;fixing of boundary
    fix range x -0.31,-0.29 y 0.31,0.29 z -0.21,-0.1
;
;defining movement of gridpoint opposite to the fixed one
def tam
    v1=-0.1    ;velocity of gridpoint
    v2=-0.05
    s=-0.01    ;prescribed displacement
;
    ph=gp_near(0.3,-0.3,-0.2)
;
    command
        bound zvel=@v1 range x 0.33,0.27 y -0.33,-0.27 z -0.27,-0.18
        bound zvel=@v2 range x 0.33,0.27 y 0.33,0.27 z -0.27,-0.18
        bound zvel=@v2 range x -0.33,-0.27 y -0.33,-0.27 z -0.27,-0.18
    end_command
;
    loop n (1,100)
        elm=gp_zdis(ph)
;
        if elm > s then
            command
;                pr @elm
                cycle 100
            end_command
        end_if
    end_loop
;
    command
        bound zvel=0. range x 0.33,0.27 y -0.33,-0.27 z -0.27,-0.18
        bound zvel=0. range x 0.33,0.27 y 0.33,0.27 z -0.27,-0.18
        bound zvel=0. range x -0.33,-0.27 y -0.33,-0.27 z -0.27,-0.18
    end_command
end;

```

Displacement results of the generated base-settlement can be seen on Fig. 45. For the applied velocity boundaries the block is moving like a rigid block that is fixed only with one thing. The parallel color strips are proving this phenomenon, the settlement is linear. This kind of movement should not indicate stresses in the block because deformations weren't occurred. Stress results can be seen on Fig. 46.– 51.

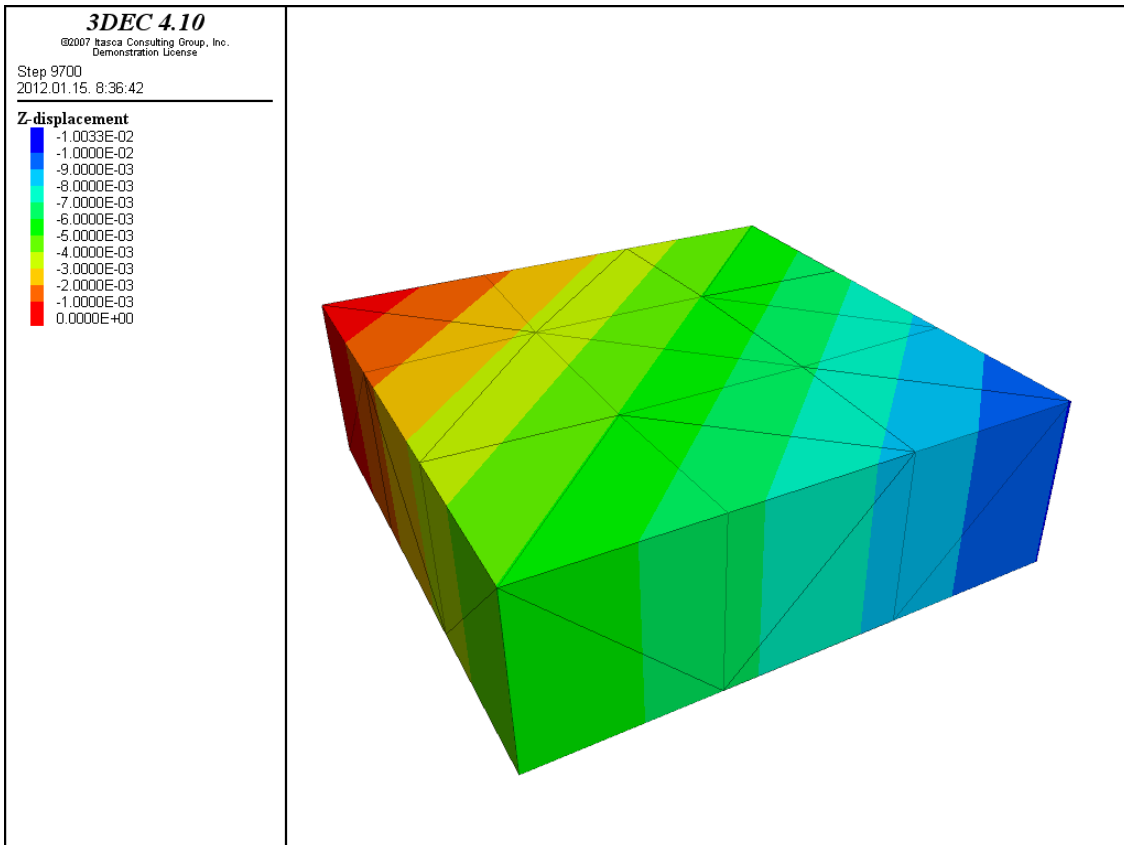


Fig. 45.: z-displacements of base-block in case only one corner is fixed on the lower part of it

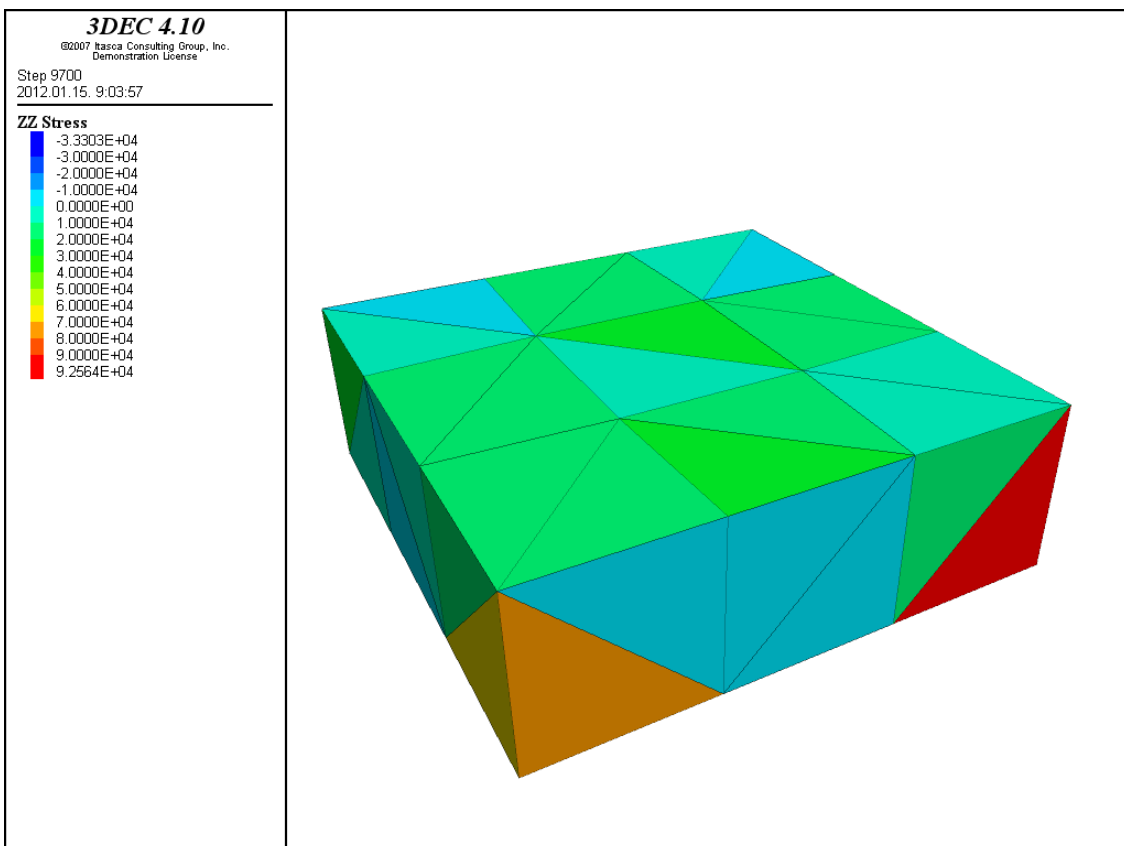


Fig. 46.: σ_{zz} normal stresses of base-block in case only one corner is fixed on the lower part of it

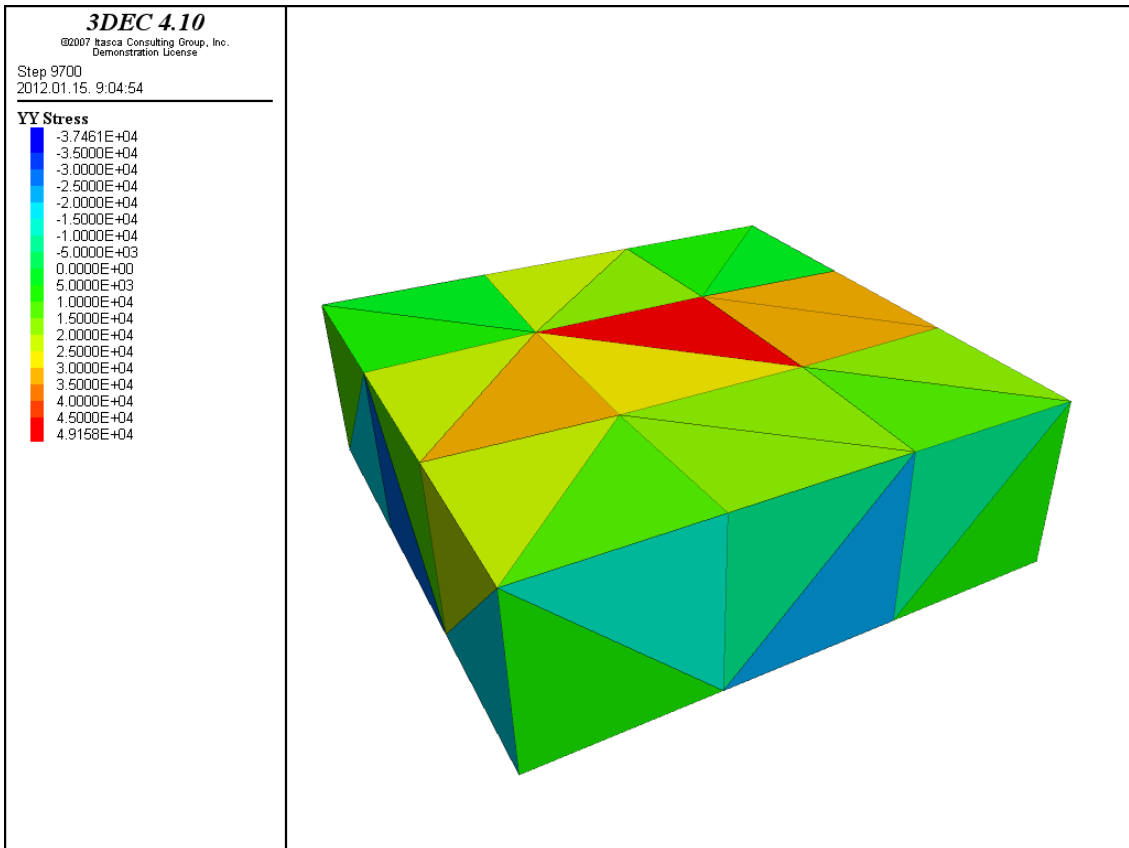


Fig. 47.: σ_{yy} normal stresses of base-block in case only one corner is fixed on the lower part of it

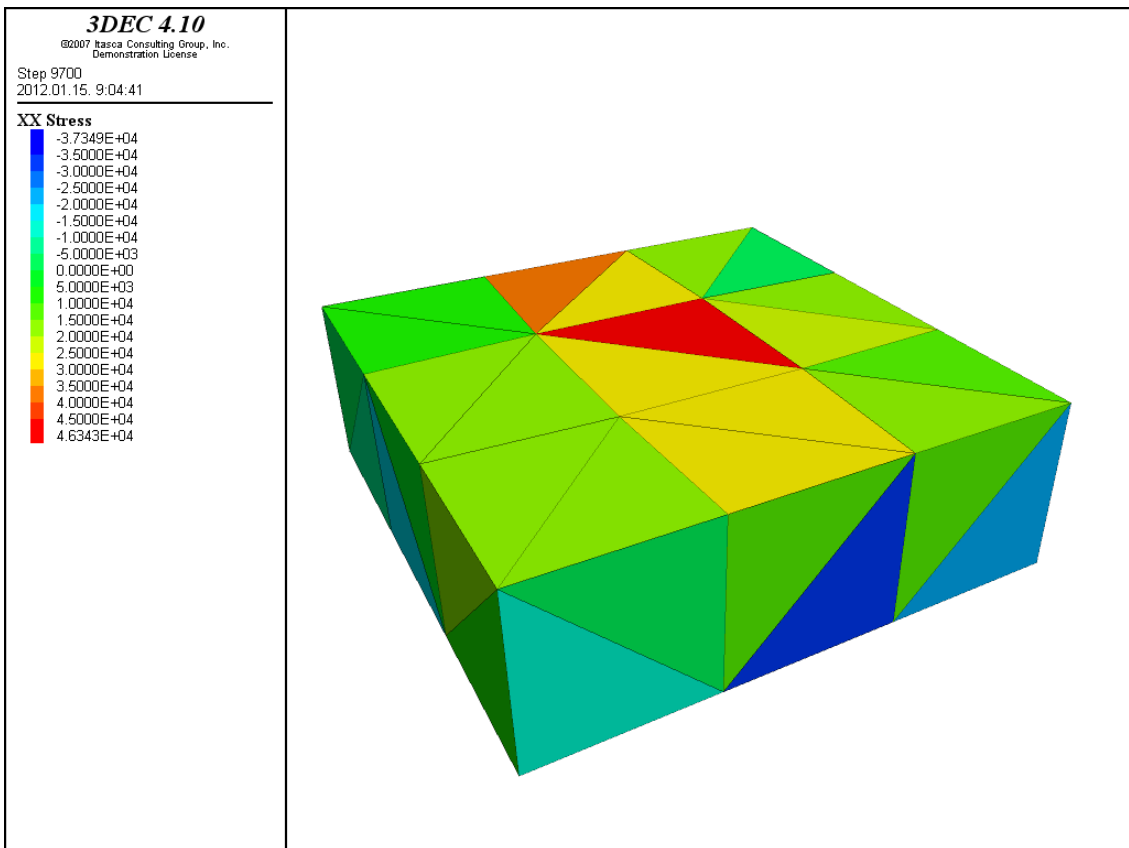


Fig. 48.: σ_{xx} normal stresses of base-block in case only one corner is fixed on the lower part of it

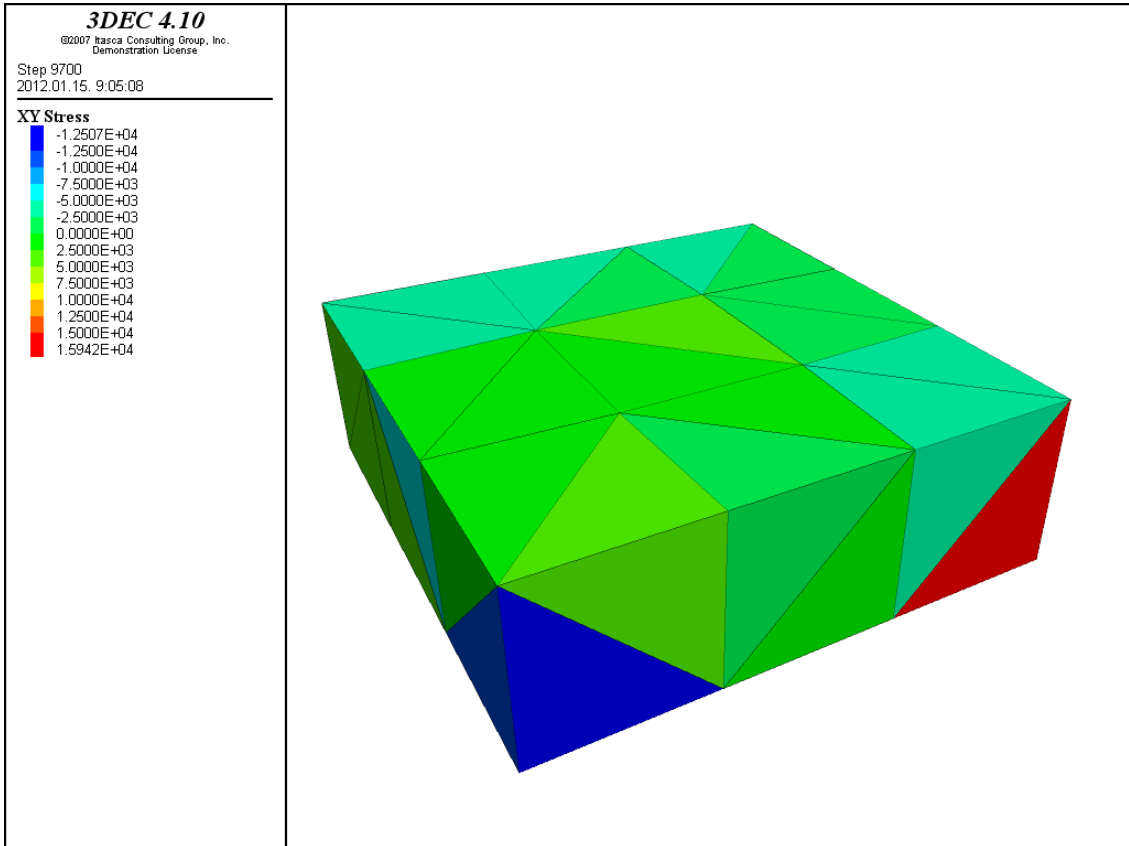


Fig. 49.: σ_{xy} shear stresses of base-block in case only one corner is fixed on the lower part of it

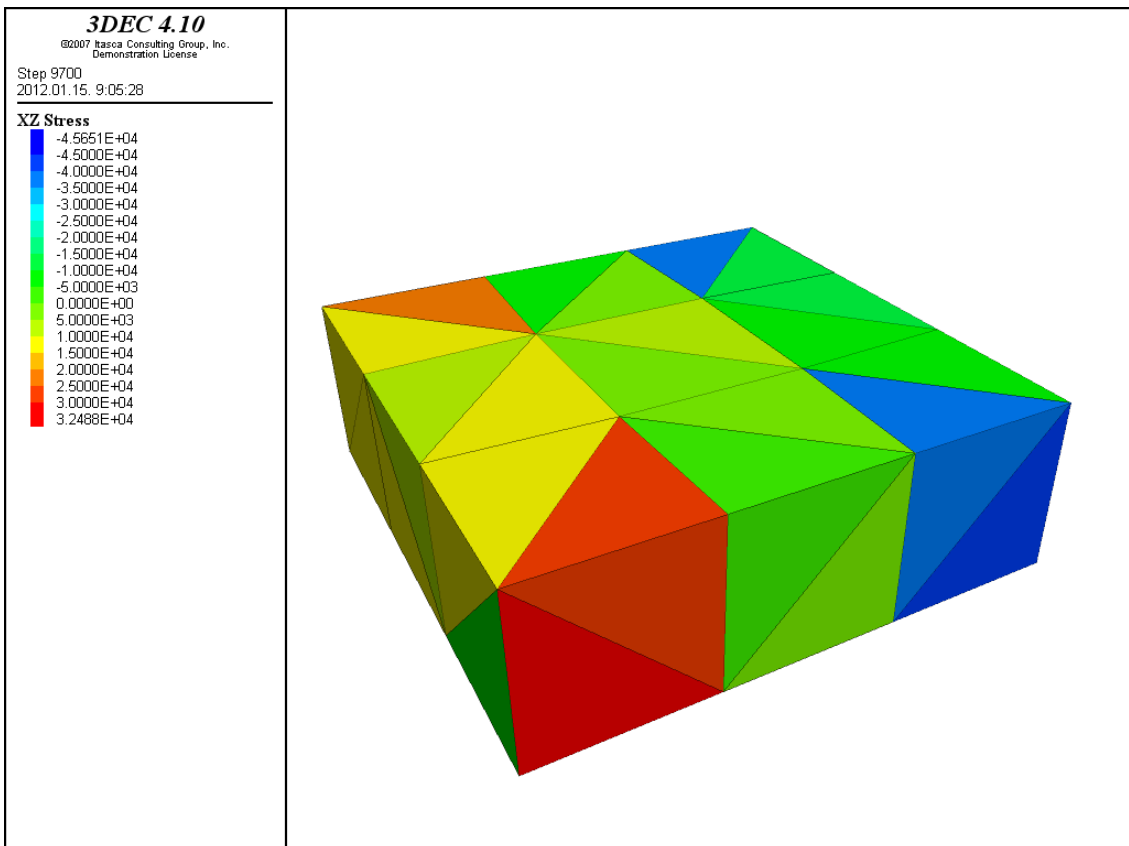


Fig. 50.: σ_{xz} shear stresses of base-block in case only one corner is fixed on the lower part of it

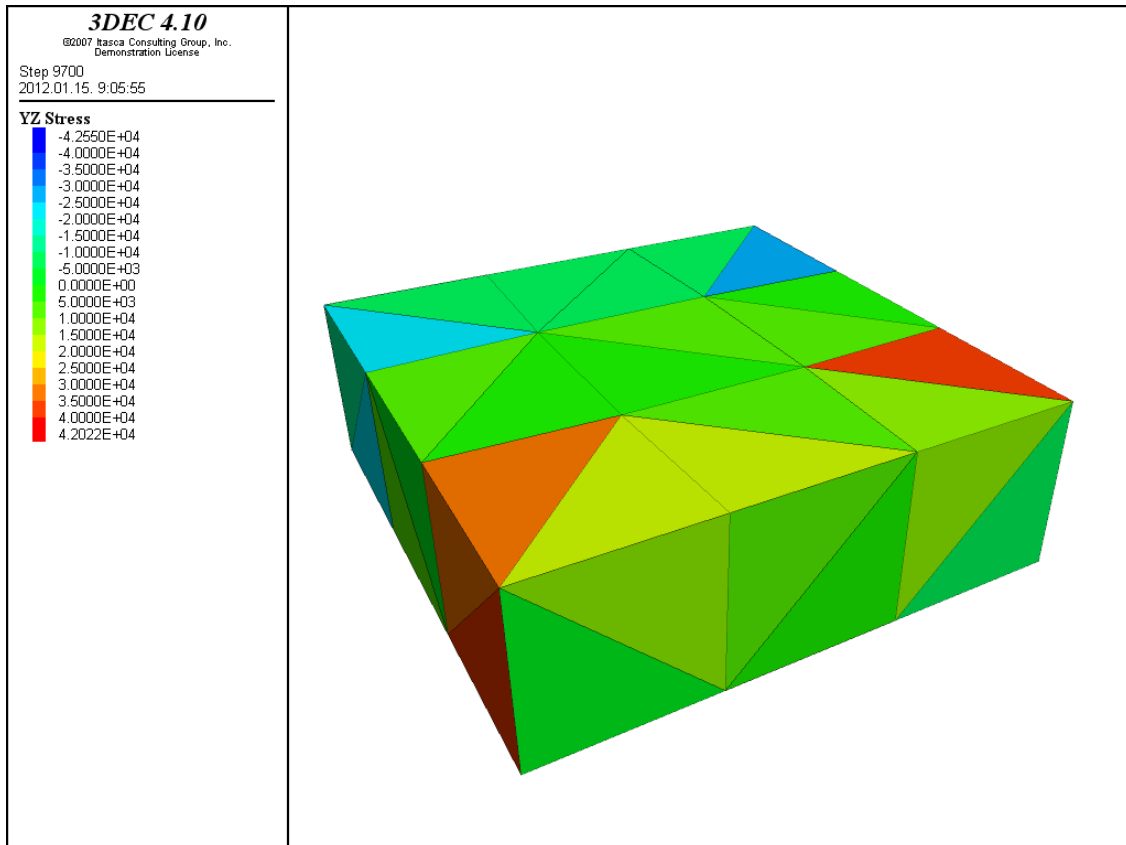


Fig. 51.: σ_{yz} shear stresses of base-block in case only one corner is fixed on the lower part of it

In spite of the previously mentioned phenomenon stress results had shown that the block is not free from stresses. On the other hand the value of registered stresses compared to the previous stress results (indicated by perpendicular loads) is not too high, on the other hand they cannot be neglected. Now I'm going to investigate deformations of the base block to find out the cause of stresses.

```
3dec>list grid disp range z -0.25,-0.18
```

block	grid-point	x	y	z	dx	dy	dz
217	1960	1.067E-01	1.081E-01	-2.050E-01	-8.107E-04	7.938E-04	-5.010E-03
217	1921	1.066E-01	-1.067E-01	-2.068E-01	-8.865E-04	8.015E-04	-6.803E-03
217	1882	-1.086E-01	1.082E-01	-2.032E-01	-8.030E-04	8.548E-04	-3.212E-03
217	1843	-1.083E-01	-1.073E-01	-2.050E-01	-8.791E-04	8.623E-04	-5.014E-03
217	1570	-3.008E-01	1.009E-01	-2.017E-01	-7.988E-04	9.093E-04	-1.669E-03
217	1531	-3.009E-01	-9.908E-02	-2.033E-01	-8.694E-04	9.165E-04	-3.338E-03
217	1414	-1.007E-01	3.008E-01	-2.017E-01	-7.354E-04	8.458E-04	-1.669E-03
217	1375	9.926E-02	3.008E-01	-2.033E-01	-7.425E-04	7.892E-04	-3.339E-03
217	1336	2.992E-01	1.007E-01	-2.067E-01	-8.201E-04	7.395E-04	-6.678E-03
217	1297	2.991E-01	-9.925E-02	-2.083E-01	-8.908E-04	7.467E-04	-8.348E-03
217	1258	9.905E-02	-2.992E-01	-2.083E-01	-9.542E-04	8.105E-04	-8.348E-03
217	1219	-1.009E-01	-2.991E-01	-2.067E-01	-9.470E-04	8.670E-04	-6.678E-03
217	416	-3.007E-01	3.009E-01	-2.000E-01	-7.283E-04	9.022E-04	0.000E+00
217	372	2.993E-01	3.007E-01	-2.050E-01	-7.495E-04	7.325E-04	-5.010E-03
217	328	2.990E-01	-2.992E-01	-2.100E-01	-9.618E-04	7.543E-04	-1.002E-02
217	284	-3.009E-01	-2.991E-01	-2.050E-01	-9.396E-04	9.240E-04	-5.010E-03

Fig. 52.: Displacements of the lower part of the base-block in case only one corner is fixed on it

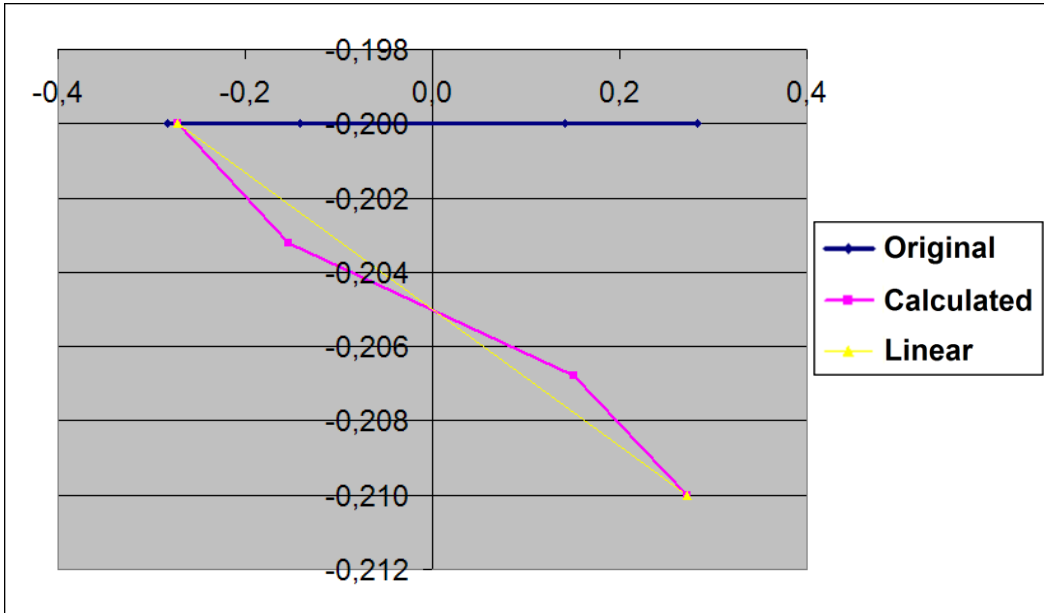


Fig. 53.: Displacement of gridpoints (416, 1882, 1921, 328) taking place on diagonal connecting the fixed and the opposite moving corners

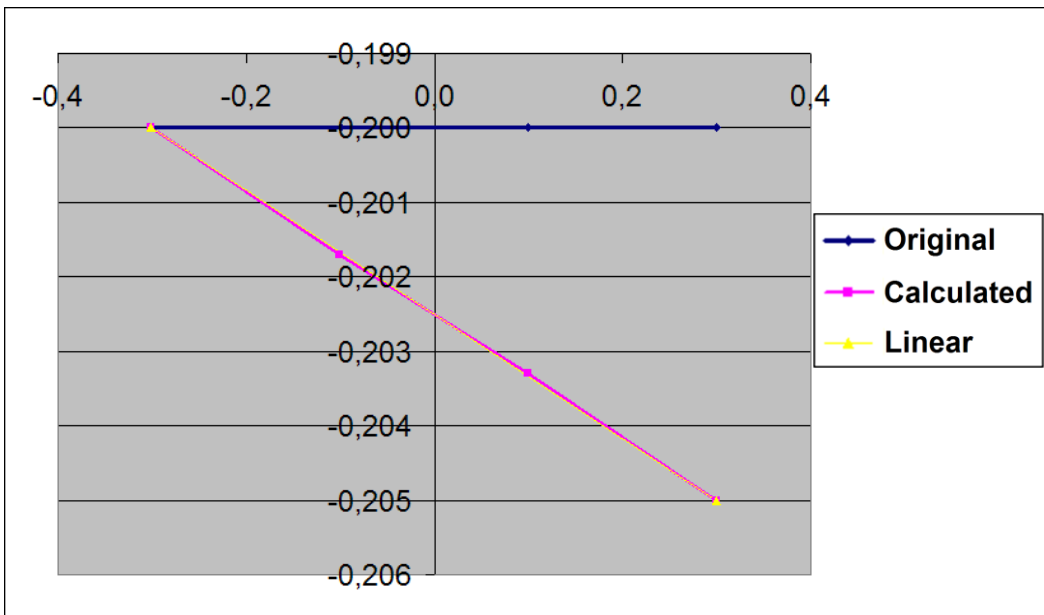


Fig. 54.: Displacements of gridpoints (416, 1414, 1375, 372) parallel to x-axe taking place on the edge connecting the fixed and moving corners

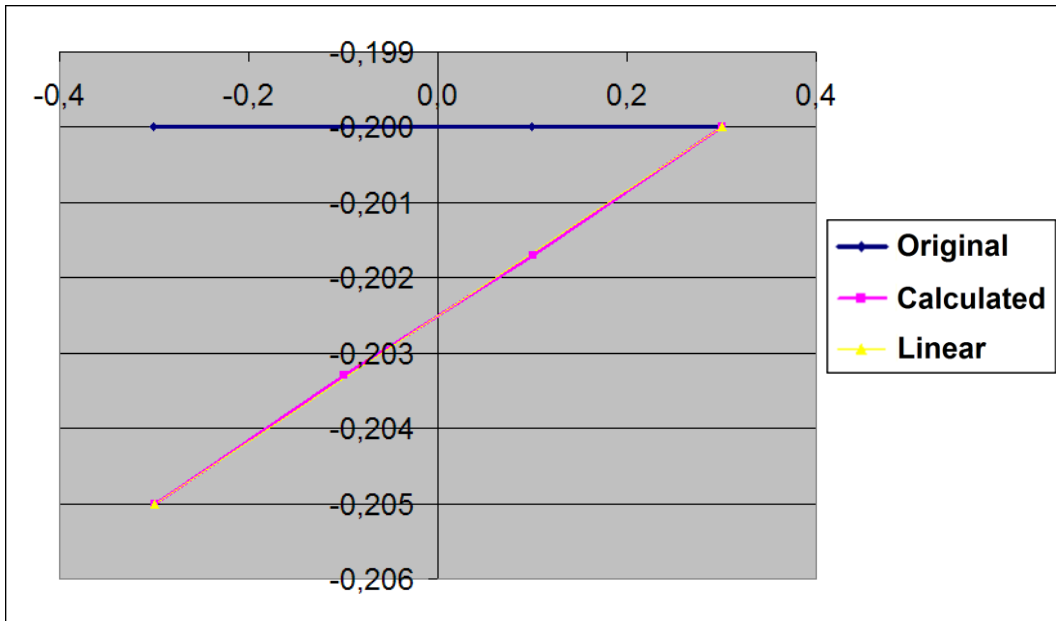


Fig. 55.: Displacements of gridpoints (416, 1570, 1531, 284) parallel to y-axle taking place on the edge connecting the fixed and moving corners

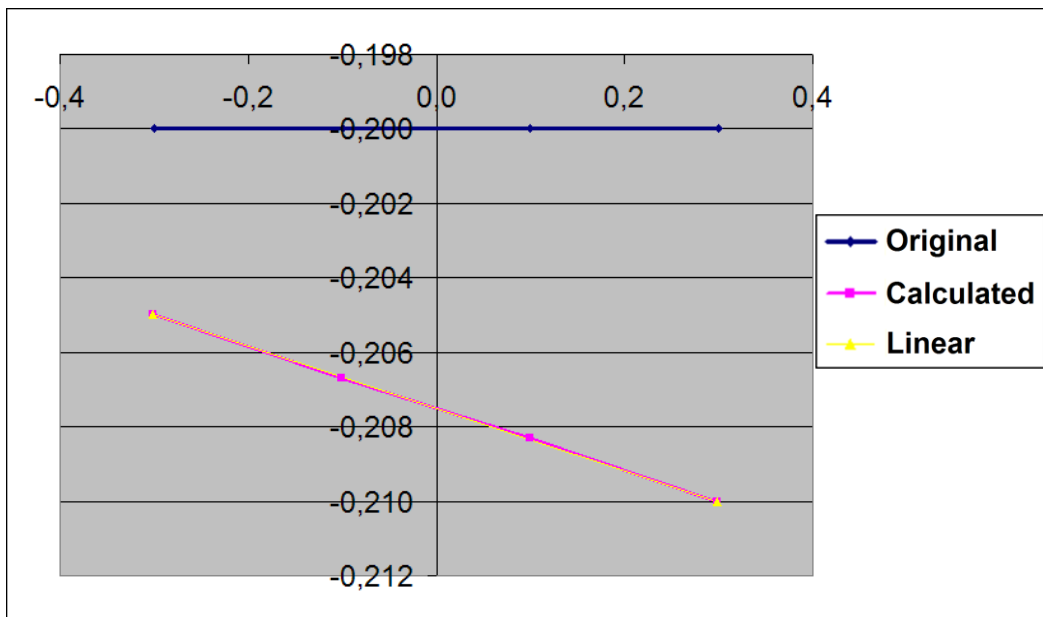


Fig. 56.: Displacements of gridpoints (284, 1219, 1258, 328) parallel to x-axle taking place on the edge connecting two moving corners

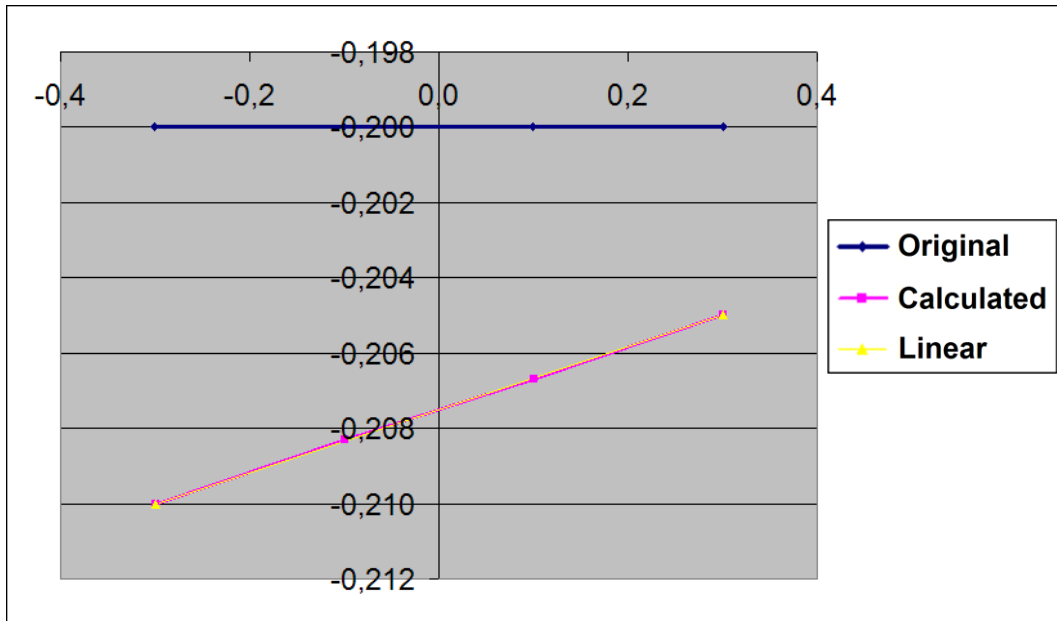


Fig. 57.: Displacements of gridpoints (372, 1336, 1297, 328) parallel to y-axis taking place on the edge connecting two moving corners

On Fig. 53.- 57. displacements of gridpoints lying on edges and diagonal of base-block are recorded. On Fig 54.- 57. displacements of edges can be seen, according to these graphs the block can be determined to be stress-free. On the other hand on Fig. 53. deformation of the diagonal is represented, studying this graph the recorded stress values can be explained.

The previously studied scripts included base-settlements generated with velocity boundaries applied to gridpoints, ranges covering the whole lower plane and block. Last but not at least three corner of the lower plane of base-block moved and only one was fixed. Now these cases are applied to the whole structure when not only the base block but bricks of column were considered too. Displacements were generated slower (0.005m/s) and faster (0.1 m/s).

Results are announced in order that cases were mentioned previously.

In case **velocity boundaries were applied to gridpoints**, displacements can be seen on Fig. 58. The script is the following:

```

new
;base-block 60x60x20cm (blue colored block on Fig. 1)
;origin is defined in the middle of the upper surface of the base-block
poly brick -0.3,0.3 -0.3,0.3 -0.2,0.
;
;block-levels made up with blocks sized: 10x10x20cm
;first level (blocks orientated in y-direction)
poly brick -0.1,0. -0.1,0.1 0.,0.1
poly brick 0.,0.1 -0.1,0.1 0.,0.1
;second level (blocks orientated in x-direction)
poly brick -0.1,0.1 -0.1,0. 0.1,0.2
poly brick -0.1,0.1 0.,0.1 0.1,0.2
;third level (blocks orientated in y-direction)
poly brick -0.1,0. -0.1,0.1 0.2,0.3
poly brick 0.,0.1 -0.1,0.1 0.2,0.3
;fourth level (blocks orientated in x-direction)
poly brick -0.1,0.1 -0.1,0. 0.3,0.4

```

```

poly brick -0.1,0.1 0.,0.1 0.3,0.4
;fifth level (blocks orientated in y-direction)
poly brick -0.1,0. -0.1,0.1 0.4,0.5
poly brick 0.,0.1 -0.1,0.1 0.4,0.5
;
;displaying of geometry
plot block
plot reset
plot set dip 60. dd 210.
;
;mesh generation with 20cm average mesh-size
gen edge 0.2
;
;material properties (sandstone):
;density: 2400 kg/m3
;E=19.3GPa Poisson-ratio: 0.38
;K=26.8GPa G=7GPa
prop mat=1 dens=2400. k=2.68e10 g=7.e9
;
;joint properties (Coulomb-slip model)
;normal stiffness (1000 MPa/m) and shear stiffness (100 MPa/m)
;friction angle 25°
change jcons=1
prop jmat=1 jkn 1.e9 jks 1.e8 fric 25.
;
;supported on the lower surface of the base-block (velocity boundary with zero values in every direction)
bound xvel=0.0 range x -0.31,0.31 y -0.31,0.31 z -0.21,-0.1
bound yvel=0.0 range x -0.31,0.31 y -0.31,0.31 z -0.21,-0.1
bound zvel=0.0 range x -0.31,0.31 y -0.31,0.31 z -0.21,-0.1
;
;for cycle is applied for rising the value of gravitational force this is necessary because some structures
;collapse if the gravitation is applied in one step. "grav_lep" is the value that defines the number of cycles
;gravitation is applied the value of „gr” is rising with the value „g/grav_lep” in every cycle
;Calculation process is running until the structure reaches the desired accuracy
;this is measured by the ratio of unbalanced and applied forces, now I set the limitvalue for 1e-5
def gra
grav_lep=10
gr=0
loop n (1,grav_lep)
gr=gr-10/grav_lep
command
gravity 0.0,0.0,@gr
solve rat 1.e-5
end_command
end_loop
end;
;calling the „gra” command
@gra
;uneven settlement with one moving gridpoint
;fixing boundary velocities
fix range x -0.31,-0.29 y 0.31,0.29 z -0.21,-0.1
fix range x 0.31,0.29 y 0.31,0.29 z -0.21,-0.1
fix range x -0.31,-0.29 y -0.31,-0.29 z -0.21,-0.1
;
;definition of displacement of the moving gridpoint
def tam
v=-0.005 ;velocity of gridpoint
s=-0.01 ;prescribed displacement
;
ph=gp_near(0.3,-0.3,-0.2)

```

```

;
command
  bound zvel=@v range x 0.33,0.27 y -0.33,-0.27 z -0.27,-0.18
end_command
;
loop n (1,100)
  elm=gp_zdis(ph)
;
  if elm > s then
    command
      cycle 100
    end_command
  end_if
end_loop
;
command
  bound zvel=0. range x 0.33,0.27 y -0.33,-0.27 z -0.27,-0.18
end_command
end;
;calling „tám” command
@tám

```

Taking a look at the script it can be seen, that at first gravitational load was applied to the structure, during this calculation-period every gridpoints' velocity at the corner of the base block were defined to zero. After the column got into balanced state three of the gridpoints (at the corners) were turned to fixed. The fourth moving gridpoint's settlement was defined by the previously demonstrated script. Velocity of moving gridpoint was defined with 0.005 m/s, this value turned out to be slow enough to get real displacement results. In case of faster movements a horizontal gap occurred between the base-block and the lowest block level. If movements were generated with 0.01 m/s the mentioned horizontal gap occurred and this cause that displacements cannot develop on the upper part of the structure (column remains unmoved and settlements can realised only on the base block). On Fig. 58. the case is represented when the base-block and the upper levels are moving together. In spite of this on Fig. 59. velocity was defined to 0.1 m/s, this case the column cannot follow the settlements of the base-block.

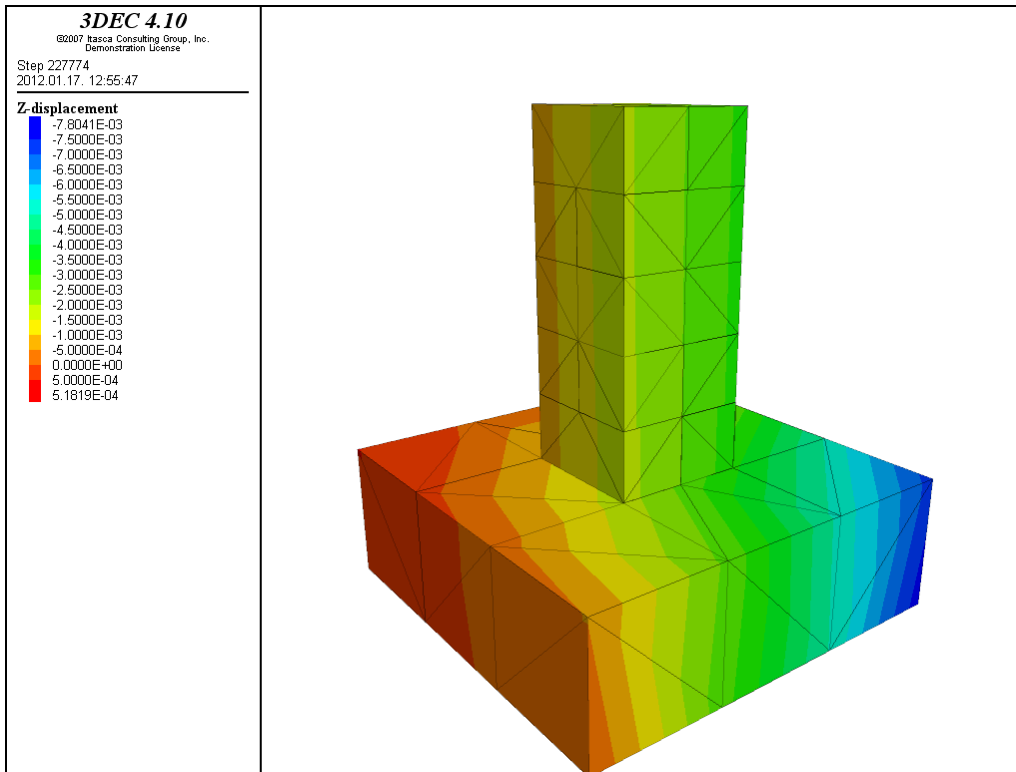


Fig. 58.: z-displacements of the whole structure in case uneven settlement was generated with 0.005 m/s velocity, one of the four corners is fixed the others are moving

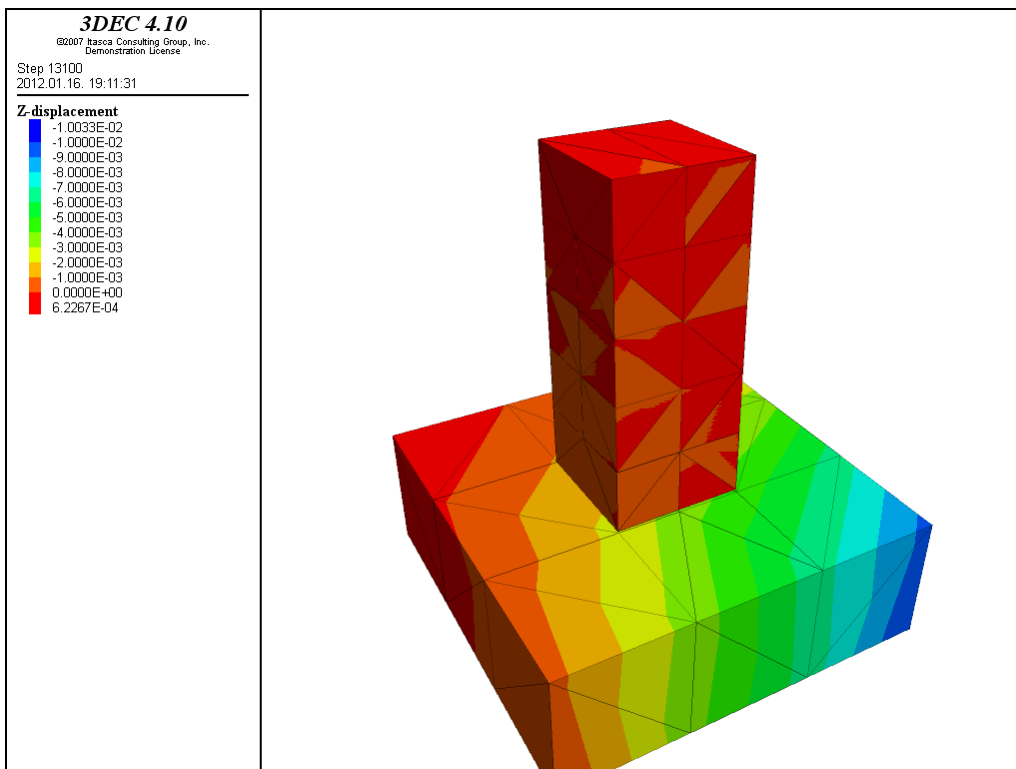


Fig. 59.: z-displacements of the whole structure in case uneven settlement was generated with 0.1 m/s velocity, one of the four corners is fixed the others are moving

If base-settlement is defined with **ranges covering the whole plane of the base-block**, bricks of the lowest block-level is supported at almost the whole plane of them. This is noticeable on Fig. 60. (velocity: 0.1 m/s), where the left side of the column is almost motionless, but on the

right handside tilting is recorded. The movement is caused by the lowest brick what is the most closer to the concentrated settlement.

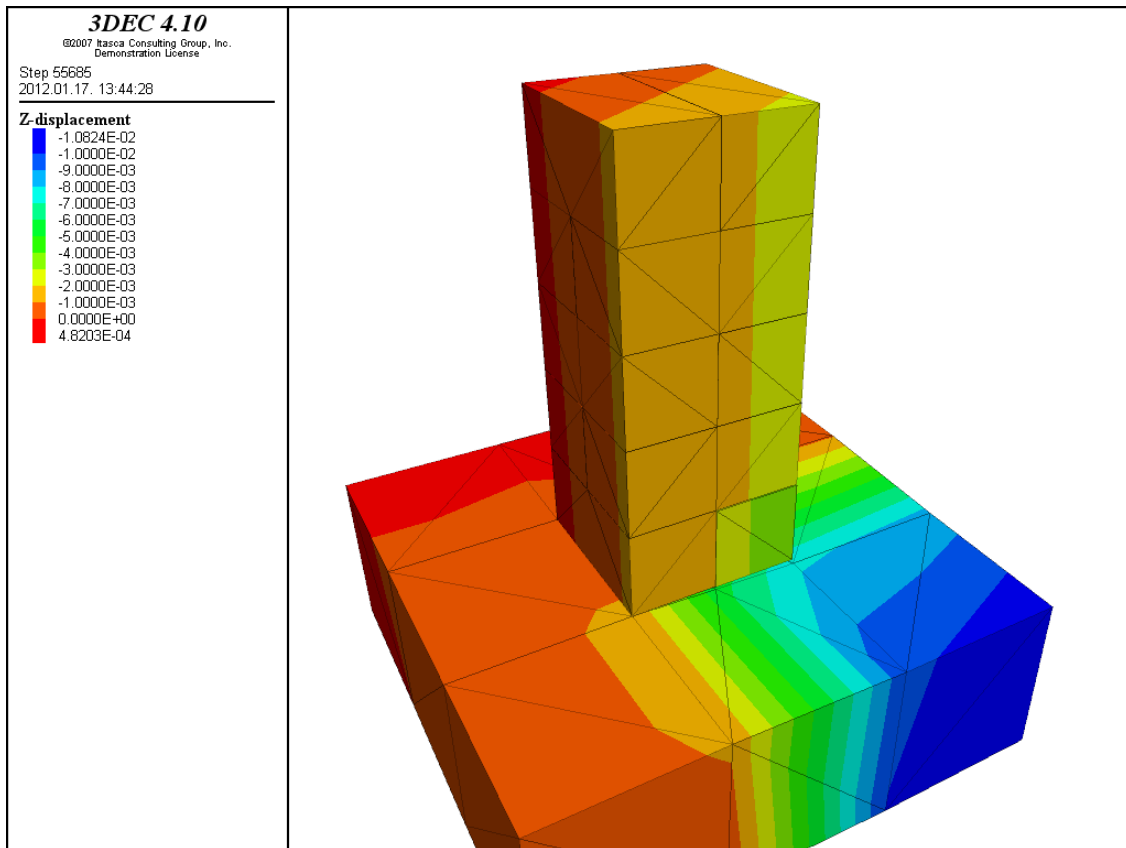


Fig. 60.: z-displacements of the whole structure in case uneven settlement was generated with 0.1 m/s velocity, settlement was applied for four ranges containing the whole area of the lower part of base-block

On Fig. 61. movements were generated with 0.005 m/s velocity, in this case tilting was recorded too, but the base-block and the upper levels of the column moved together. Script is the following:

```

new
;base-block 60x60x20cm (blue colored block on Fig. 1)
;origin is defined in the middle of the upper surface of the base-block
  poly brick -0.3,0.3 -0.3,0.3 -0.2,0.
;
;block-levels made up with blocks sized: 10x10x20cm
;first level (blocks orientated in y-direction)
  poly brick -0.1,0. -0.1,0.1 0.,0.1
  poly brick 0.,0.1 -0.1,0.1 0.,0.1
;second level (blocks orientated in x-direction)
  poly brick -0.1,0.1 -0.1,0. 0.1,0.2
  poly brick -0.1,0.1 0.,0.1 0.1,0.2
;third level (blocks orientated in y-direction)
  poly brick -0.1,0. -0.1,0.1 0.2,0.3
  poly brick 0.,0.1 -0.1,0.1 0.2,0.3
;fourth level (blocks orientated in x-direction)
  poly brick -0.1,0.1 -0.1,0. 0.3,0.4
  poly brick -0.1,0.1 0.,0.1 0.3,0.4
;fifth level (blocks orientated in y-direction)
  poly brick -0.1,0. -0.1,0.1 0.4,0.5
  
```

```

poly brick 0.,0.1 -0.1,0.1 0.4,0.5
;
;displaying of geometry
plot block
plot reset
plot set dip 60. dd 210.
;
;mesh generation with 20cm average mesh-size
gen edge 0.2
;
;material properties (sandstone):
;density: 2400 kg/m3
;E=19.3GPa Poisson-ratio: 0.38
;K=26.8GPa G=7GPa
prop mat=1 dens=2400. k=2.68e10 g=7.e9
;
;joint properties (Coulomb-slip model)
;normal stiffness (1000 MPa/m) and shear stiffness (100 MPa/m)
;friction angle 25°
change jcons=1
prop jmat=1 jkn 1.e9 jks 1.e8 fric 25.
;
;supported on the lower surface of the base-block (velocity boundary with zero values in every direction)
bound xvel=0.0 range x -0.31,0.31 y -0.31,0.31 z -0.21,-0.1
bound yvel=0.0 range x -0.31,0.31 y -0.31,0.31 z -0.21,-0.1
bound zvel=0.0 range x -0.31,0.31 y -0.31,0.31 z -0.21,-0.1
;
;for cycle is applied for rising the value of gravitational force this is necessary because some structures
;collapse if the gravitation is applied in one step. "grav_lep" is the value that defines the number of cycles
;gravitation is applied the value of „gr” is rising with the value „g/grav_lep” in every cycle
;Calculation process is running until the structure reaches the desired accuracy
;this is measured by the ratio of unbalanced and applied forces, now I set the limitvalue for 1e-5
def gra
grav_lep=10
gr=0
loop n (1,grav_lep)
gr=gr-10/grav_lep
command
gravity 0.0,0.0,@gr
solve rat 1.e-5
end_command
end_loop
end;
;calling the „gra” command
@gra
;uneven base-settlement is defined with ranges containing the whole plane of the base-block
;velocities of fixed regions is defined to zero
bound zvel=0. range x -0.31,0. y 0.31,0. z -0.21,-0.18
bound zvel=0. range x 0.31,0. y 0.31,0. z -0.21,-0.18
bound zvel=0. range x -0.31,0. y -0.31,0. z -0.21,-0.18
;fixing of boundary ranges
fix range x -0.31,0. y 0.31,0. z -0.21,-0.18
fix range x 0.31,0. y 0.31,0. z -0.21,-0.18
fix range x -0.31,0. y -0.31,0. z -0.21,-0.18
;
;applying movement for the region
def tam
v=-0.005 ;velocity of region
s=-0.01 ;prescribed displacement
;

```



```

ph=gp_near(0.3,-0.3,-0.2)
;
command
  bound zvel=@v range x 0.33,0. y -0.33,0. z -0.27,-0.18
end_command
;
loop n (1,3000)
  elm=gp_zdis(ph)
;
  if elm > s then
    command
      cycle 100
    end_command
  end_if
end_loop
;
command
  bound zvel=0. range x 0.33,0. y -0.33,0. z -0.27,-0.18
end_command
end;
;calling „tám” command
@tám

```

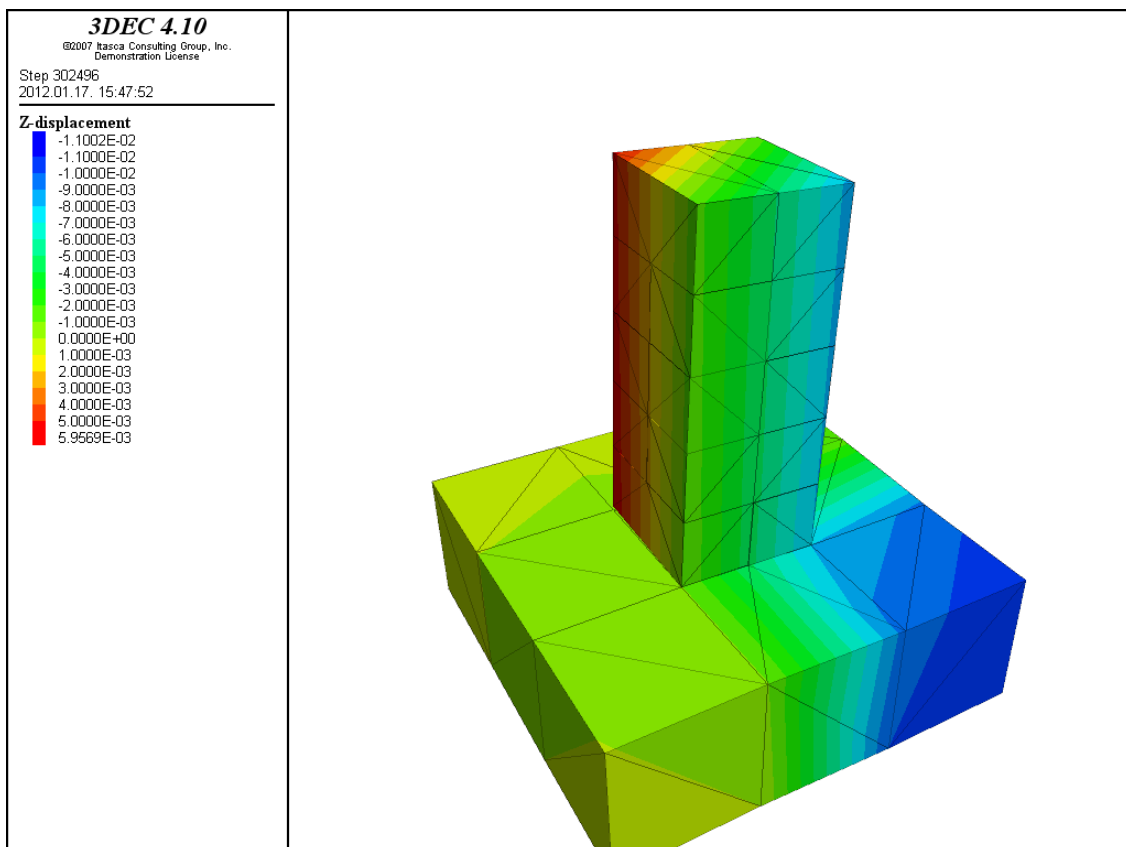


Fig. 61.: z-displacements of the whole structure in case uneven settlement was generated with 0.005 m/s velocity, settlement was applied for four ranges containing the whole area of the lower part of base-block

If **three corner of the lower plane of base-block moved and only one was fixed**, displacements of the column can be seen on Fig. 62. and 63.

On Fig. 62. results are displayed when velocity of gridpoint with the most higher displacement values is 0.1 m/s and the other two corners moving with 0.05 m/s. These fast

movements caused the previously demonstrated phenomenon (opening of the horizontal joints). In spite of this on Fig. 63. velocities were changed to 0.01 m/s and 0.005 m/s. These reduced velocities gave the proper results. In my opinion apply of slower movements are not reasonable, because they would elongate calculation-time.

Script is the following:

```

new
;base-block 60x60x20cm (blue colored block on Fig. 1)
;origin is defined in the middle of the upper surface of the base-block
poly brick -0.3,0.3 -0.3,0.3 -0.2,0.
;
;block-levels made up with blocks sized: 10x10x20cm
;first level (blocks orientated in y-direction)
poly brick -0.1,0. -0.1,0.1 0.,0.1
poly brick 0.,0.1 -0.1,0.1 0.,0.1
;second level (blocks orientated in x-direction)
poly brick -0.1,0.1 -0.1,0. 0.1,0.2
poly brick -0.1,0.1 0.,0.1 0.1,0.2
;third level (blocks orientated in y-direction)
poly brick -0.1,0. -0.1,0.1 0.2,0.3
poly brick 0.,0.1 -0.1,0.1 0.2,0.3
;fourth level (blocks orientated in x-direction)
poly brick -0.1,0.1 -0.1,0. 0.3,0.4
poly brick -0.1,0.1 0.,0.1 0.3,0.4
;fifth level (blocks orientated in y-direction)
poly brick -0.1,0. -0.1,0.1 0.4,0.5
poly brick 0.,0.1 -0.1,0.1 0.4,0.5
;
;displaying of geometry
plot block
plot reset
plot set dip 60. dd 210.
;
;mesh generation with 20cm average mesh-size
gen edge 0.2
;
;material properties (sandstone):
;density: 2400 kg/m3
;E=19.3GPa Poisson-ratio: 0.38
;K=26.8GPa G=7GPa
prop mat=1 dens=2400. k=2.68e10 g=7.e9
;
;joint properties (Coulomb-slip model)
;normal stiffness (1000 MPa/m) and shear stiffness (100 MPa/m)
;friction angle 25°
change jcons=1
prop jmat=1 jkn 1.e9 jks 1.e8 fric 25.
;
;supported on the lower surface of the base-block (velocity boundary with zero values in every direction)
bound xvel=0.0 range x -0.31,0.31 y -0.31,0.31 z -0.21,-0.1
bound yvel=0.0 range x -0.31,0.31 y -0.31,0.31 z -0.21,-0.1
bound zvel=0.0 range x -0.31,0.31 y -0.31,0.31 z -0.21,-0.1
;for cycle is applied for rising the value of gravitational force this is necessary because some structures
;collapse if the gravitation is applied in one step. "grav_lep" is the value that defines the number of cycles
;gravitation is applied the value of „gr” is rising with the value „g/grav_lep” in every cycle
;Calculation process is running until the structure reaches the desired accuracy
;this is measured by the ratio of unbalanced and applied forces, now I set the limitvalue for 1e-5
def gra

```

```

    grav_lep=10
    gr=0
    loop n (1,grav_lep)
        gr=gr-10/grav_lep
        command
            gravity 0,0,0,@gr
            solve rat 1.e-5
        end_command
    end_loop
end;
;calling the „gra” command
    @gra
;”stress-free” base-block, only one corner of the block is fixed (velocity of gridpoint is zero)
;the other three corners are moving. Velocity of two opposite moving-corners
;is half of the third moving gridpoint.
    bound zvel=0. range x -0.33,-0.27 y 0.33,0.27 z -0.21,-0.18
;fixing of boundary
    fix range x -0.33,-0.27 y 0.33,0.27 z -0.21,-0.18
;
;defining movement of gridpoint opposite to the fixed one
def tám
    v1=-0.01 ;velocity of gridpoint
    v2=-0.005 ;velocity of gridpoint
    s=-0.01 ;prescribed displacement
;
    ph=gp_near(0.3,-0.3,-0.2)
;
    command
        bound zvel=@v1 range x 0.33,0.27 y -0.33,-0.27 z -0.27,-0.18
        bound zvel=@v2 range x 0.33,0.27 y 0.33,0.27 z -0.27,-0.18
        bound zvel=@v2 range x -0.33,-0.27 y -0.33,-0.27 z -0.27,-0.18
    end_command
;
    loop n (1,3000)
        elm=gp_zdis(ph)
;
        if elm > s then
            command
;                pr @elm
                cycle 100
            end_command
        end_if
    end_loop
;
    command
        bound zvel=0. range x 0.33,0.27 y -0.33,-0.27 z -0.27,-0.18
        bound zvel=0. range x 0.33,0.27 y 0.33,0.27 z -0.27,-0.18
        bound zvel=0. range x -0.33,-0.27 y -0.33,-0.27 z -0.27,-0.18
    end_command
end;
;calling „tám” command
    @tám

```

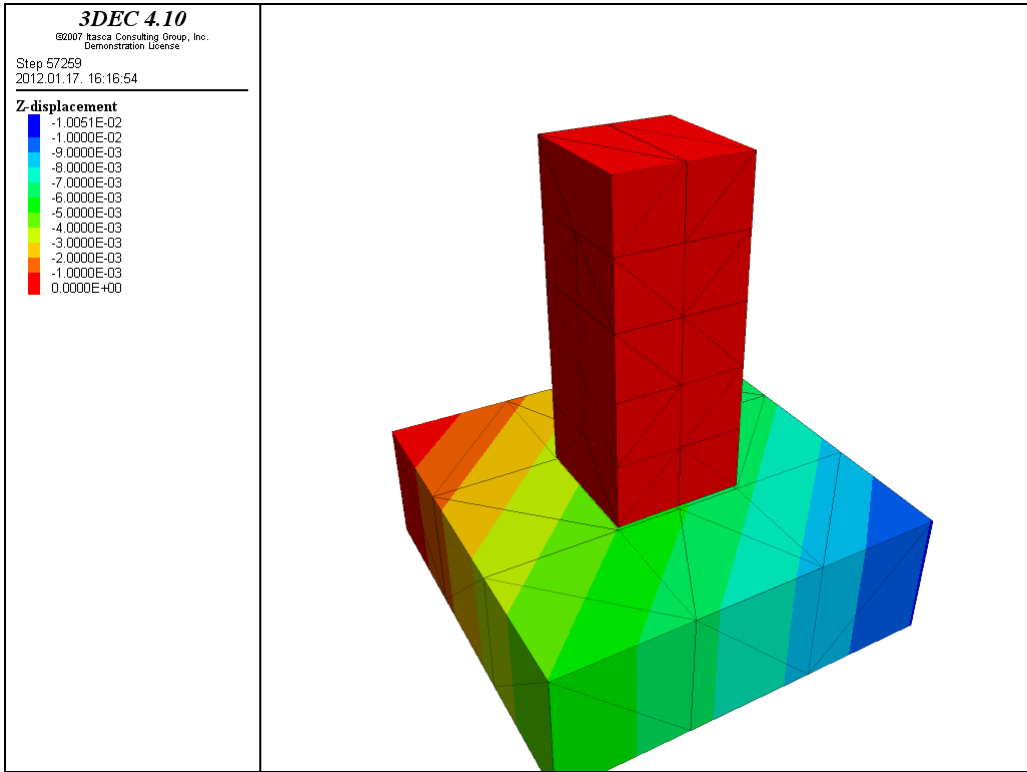


Fig. 62.: z-displacements of the whole structure in case one corner is fixed, the moving corner's velocity opposite to the fixed one is 0.1 m/s and the rest has 0.05 m/s velocity

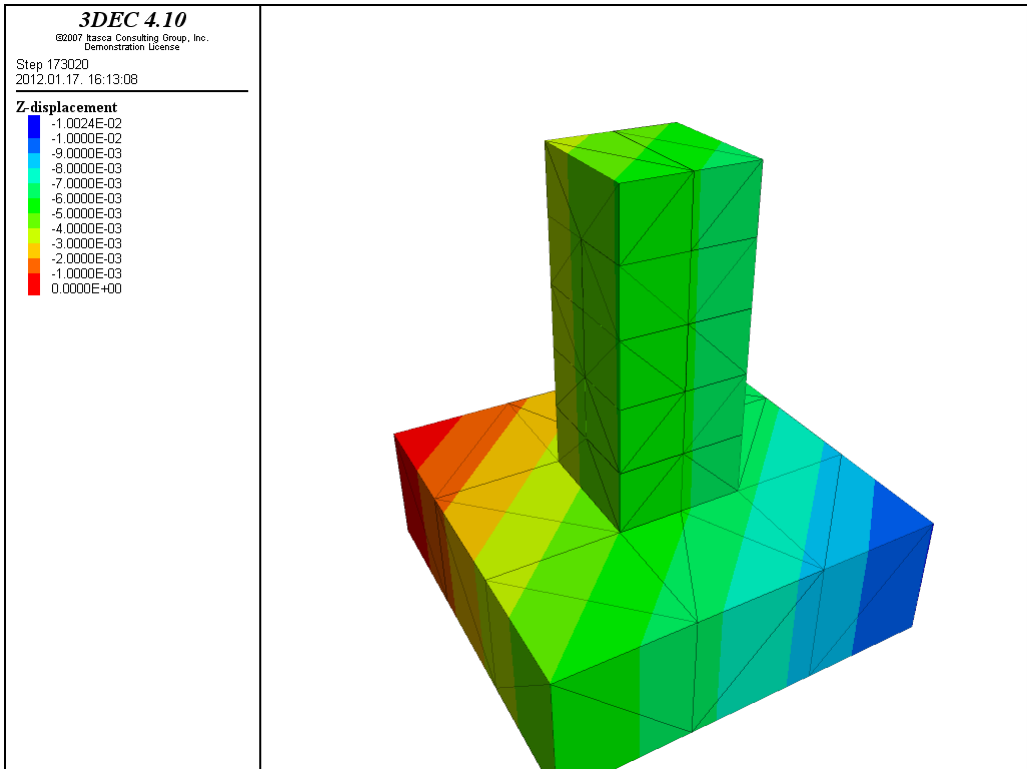


Fig. 63.: z-displacements of the whole structure in case one corner is fixed, the moving corner's velocity opposite to the fixed one is 0.01 m/s and the rest has 0.005 m/s velocity

According to the previously represented results if movements defined with too fast velocities horizontal gaps are occurred between structural parts with different boundary conditions (supports). In case ranges covered the whole plane of the base-block movements of the upper

parts could develop at higher velocities too, but on the whole smaller movements give more realistic results.

3.5. Free Boundary

Removes applied boundary conditions in the chosen direction (x, y, z). Furthermore the FIXED velocities can get relieved with FREE command.

```
boundary xfree  
boundary yfree  
boundary zfree
```

3.6. Non-reflecting (Viscous) Boundary (only for dynamic analyses)

The modeling of geomechanics problems involves media which, at the scale of the analysis, are better represented as unbounded. Deep underground excavations are normally assumed to be surrounded by an infinite medium, while surface and near-surface structures are assumed to lie on a half-space. Numerical methods relying on the discretization of a finite region of space require that appropriate conditions be enforced at the artificial numerical boundaries. In static analyses, fixed or elastic boundaries (e.g., represented by boundary element techniques) can be realistically placed at some distance from the region of interest. In dynamic problems, however, such boundary conditions cause the reflection of outward propagating waves back into the model, and do not allow the necessary energy radiation. The use of a larger model can minimize the problem, since material damping will absorb most of the energy in the waves reflected from distant boundaries. However, this solution leads to large computational costs. The alternative is to use non-reflecting (or absorbing) boundaries. Several formulations have been proposed. The viscous boundary developed by Lysmer and Kuhlemeyer (1969) is used in *3DEC*. Viscous boundaries cannot be used together with a velocity boundary. Viscous boundaries may be combined with stress boundaries.

3.7. Reaction Boundary

Point loads are applied in the opposite direction to unbalanced forces at all vertices within the range. The command „cycle 1” should be given before „bound reaction” so that unbalanced forces are calculated. Any velocity boundary conditions will be removed when this command is applied.

Using „reaction boundary” reactions can be determined in time structure collapsing. In order to determine a collapse load, it often is better to use “strain-controlled” boundary conditions instead of “stress-controlled” boundary conditions (i.e., apply a constant velocity and measure the boundary reaction forces, rather than apply forces and measure displacements). A system that collapses becomes difficult to control as the applied load approaches the collapse load. This is true of a real system as well as a model system. If you imagine a normal stress-strain curve (σ - ϵ) of a structure, it is obvious that in the range where collapse occurs the same value of loads generates higher displacements. If we applying loads for the structure and try to find the point where it collapse, it can easily cause that the results won't be accurate and the last load-blast turns the structure into collapse-state instead of reaching the accurate point where it starts.

I did not care the following types of boundaries: *Fluid Boundary, Fluid-Thermal Coupling, Histories for Loads and Velocities.*

4. Other possible functions of „Boundary”

„list boundary state”:

The command lists the state of all the gridpoints within defined „range”. Boundary condition state specified by boundary code according to the coordinate axes where it acts (ix, iy, iz), the codes are the following:

0	free
1	stress (or force)
2	viscous
3	velocity

For demonstrating the working of command I used the following script:

```
new
;base-block 60x60x20cm (blue colored block on Fig. 1)
;origin is defined in the middle of the upper surface of the base-block
  poly brick -0.3,0.3 -0.3,0.3 -0.2,0.
;
;block-levels made up with blocks sized: 10x10x20cm
;first level (blocks orientated in y-direction)
  poly brick -0.1,0. -0.1,0.1 0.,0.1
  poly brick 0.,0.1 -0.1,0.1 0.,0.1
;second level (blocks orientated in x-direction)
  poly brick -0.1,0.1 -0.1,0. 0.1,0.2
  poly brick -0.1,0.1 0.,0.1 0.1,0.2
;third level (blocks orientated in y-direction)
  poly brick -0.1,0. -0.1,0.1 0.2,0.3
  poly brick 0.,0.1 -0.1,0.1 0.2,0.3
;fourth level (blocks orientated in x-direction)
  poly brick -0.1,0.1 -0.1,0. 0.3,0.4
  poly brick -0.1,0.1 0.,0.1 0.3,0.4
;fifth level (blocks orientated in y-direction)
  poly brick -0.1,0. -0.1,0.1 0.4,0.5
  poly brick 0.,0.1 -0.1,0.1 0.4,0.5
;
;displaying of geometry
  plot block
  plot reset
  plot set dip 60. dd 210.
;
;mesh generation with 20cm average mesh-size
  gen edge 0.2
;
;material properties (sandstone):
;density: 2400 kg/m3
;E=19.3GPa Poisson-ratio: 0.38
;K=26.8GPa G=7GPa
  prop mat=1 dens=2400. k=2.68e10 g=7.e9
;
;joint properties (Coulomb-slip model)
;normal stiffness (1000 MPa/m) and shear stiffness (100 MPa/m)
;friction angle 25°
  change jcons=1
```

```

prop jmat=1 jkn 1.e9 jks 1.e8 fric 25.
;
;distributed load on the upper plane of the column
;loads were applied with "bound stress" command
;value of the perpendicular distributed load: 10kN/m2=10000 kg/ms2
bound (-0.11,0.11) (-0.11,0.11) (0.45,0.5) stress 0.,0.,-1.e4,0.,0.,0.
;
;base-block is fixed, velocities of gridpoints are zero for the whole block
bound xvel=0. range x -0.3,0.3
bound yvel=0. range y -0.3,0.3
bound zvel=0. range z -0.2,0.
;fixing the boundary conditions according to displacements
fix range x -0.3,0.3 y -0.3,0.3 z -0.2,0.

```

The bound state command was applied for the range $z_1=0.45$ and $z_2=0.5$ (in direction of z-axis), see the command below:

```
list bound state range z 0.45,0.5
```

boundary data: state

bound vert	vertex	x	y	z	ix	iy	iz
26149	7308	1.000E-01	1.000E-01	5.000E-01	3	3	1
26185	7176	1.000E-01	-1.000E-01	5.000E-01	3	3	1
26221	7247	0.000E+00	1.000E-01	5.000E-01	3	3	1
26257	7132	0.000E+00	-1.000E-01	5.000E-01	3	3	1
26293	6686	0.000E+00	1.000E-01	5.000E-01	3	3	1
26329	6554	0.000E+00	-1.000E-01	5.000E-01	3	3	1
26365	6625	-1.000E-01	1.000E-01	5.000E-01	3	3	1
26401	6510	-1.000E-01	-1.000E-01	5.000E-01	3	3	1

As it can be seen, the chart contains the accurate coordinates and identification numbers of gridpoints within defined range. The last three column of the chart gives the boundary condition state specified by boundary code according to the coordinate axes (ix, iy, iz). Chart is containing gridpoints take place on the upper part of the column. According to the results in case of every gridpoint velocities (code: 3) and stress or force (code: 1) are defined. It can be realised that velocities of gridpoints are zero for the whole structure, this is caused by misdefinition of supports of base-block. This example had shown that the command can be used for controlling conditions like this. The proper definition of base-block's support is given with the following script, this type of boundary sets the upper part of the column free in x and y directions, and 0 velocity refers only to the base block.

```

;base-block is fixed, velocities of gridpoints are zero for the whole block
bound xvel=0.0 range x -0.3,0.3 y -0.3,0.3 z -0.2,0.0
bound yvel=0.0 range x -0.3,0.3 y -0.3,0.3 z -0.2,0.0
bound zvel=0.0 range x -0.3,0.3 y -0.3,0.3 z -0.2,0.0
;fixing the boundary conditions
fix range x -0.3,0.3 y -0.3,0.3 z -0.2,0.0

```

Results given after proper boundaries were set to the base-block:

boundary data: state

bound vert	vertex	x	y	z	ix	iy	iz
26149	7308	1.000E-01	1.000E-01	5.000E-01	0	0	1
26185	7176	1.000E-01	-1.000E-01	5.000E-01	0	0	1
26221	7247	0.000E+00	1.000E-01	5.000E-01	0	0	1
26257	7132	0.000E+00	-1.000E-01	5.000E-01	0	0	1
26293	6686	0.000E+00	1.000E-01	5.000E-01	0	0	1
26329	6554	0.000E+00	-1.000E-01	5.000E-01	0	0	1
26365	6625	-1.000E-01	1.000E-01	5.000E-01	0	0	1
26401	6510	-1.000E-01	-1.000E-01	5.000E-01	0	0	1

„list boundary displacement”:

The command listing displacements of boundary gridpoints within the defined range.

„list boundary force”:

The command listing forces acting at boundary gridpoints within the defined range. Results are the following: f_x , f_y , f_z (permanent forces) és a f_{xi} , f_{yi} , f_{zi} (incremental forces – additional forces defined in the actual load-phase).

„list boundary reactions forces”:

Reaction forces at any gridpoint with a fixed velocity (output is in a format that can be used directly in the „boundary point x y z load” command).

„list boundary reactions tractions”:

Reactions at fixed velocity points in the form of average tractions on the surrounding faces (the output can be used directly in the „boundary triangle” command).

„list boundary summary”:

The command listing the total value of forces applied on the x -, y - and z -boundaries.

„list boundary velocity”:

The command listing the velocities applied to boundary.

5. Display results: PLOT command

In the previous sections a few options connected with plot command had been demonstrated, but this chapter is more detailed.

As „plot” command has a lot of subcommands and the user can find a several ways to manipulate the displayed items, so for more details see Command Reference Chapter of 3DEC Online manual.

The following keywords assign view settings:

`„plot reset“:`

This command resets the current view settings to default values.

`„plot set“:`

Most of the results can get by subcommands connected with „plot set” command can be produced in easier way because the graphical interface of the software let the user to modify the current view. Switches and buttons can be found on the left handside of the software-window. On the other hand during script writing use of these commands can be crucial. On Fig. 64. definiton of view settings can be seen.

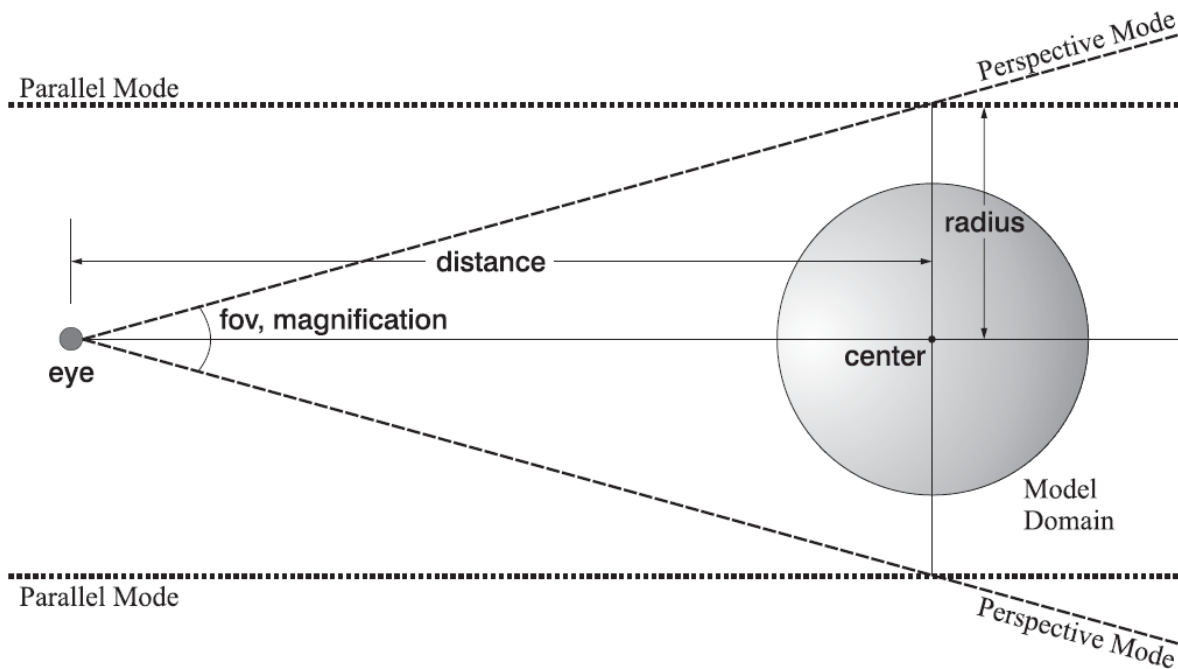


Fig. 64.: Definition of view setting keywords

Some of the „plot set” subcommands are the following: background, center, dip direction, dip, distance, eye, fov, jobtitle, legend, magnification, printsize, projection, radius, viewtitle.

For example the following script is plotting the defined blocks and changing the dip and dip direction of the default view. Finally „plot reset” command resets the current view settings to default values.

```
plot block
```

If this script is applied the „plot block” command plots the blocks were defined previously for example in case of the script „gravity” (on page number 2 and 3) this gives the following result (see Fig. 65.).

If the following command-line is applied the result can be seen on Fig. 66.:

```
plot set dip 60. dd 210.
```

If „plot reset” command is used the result is the same as on Fig. 65.. It is obvious that by default the blocks are displayed according to the three-dimensional cartesian axes.

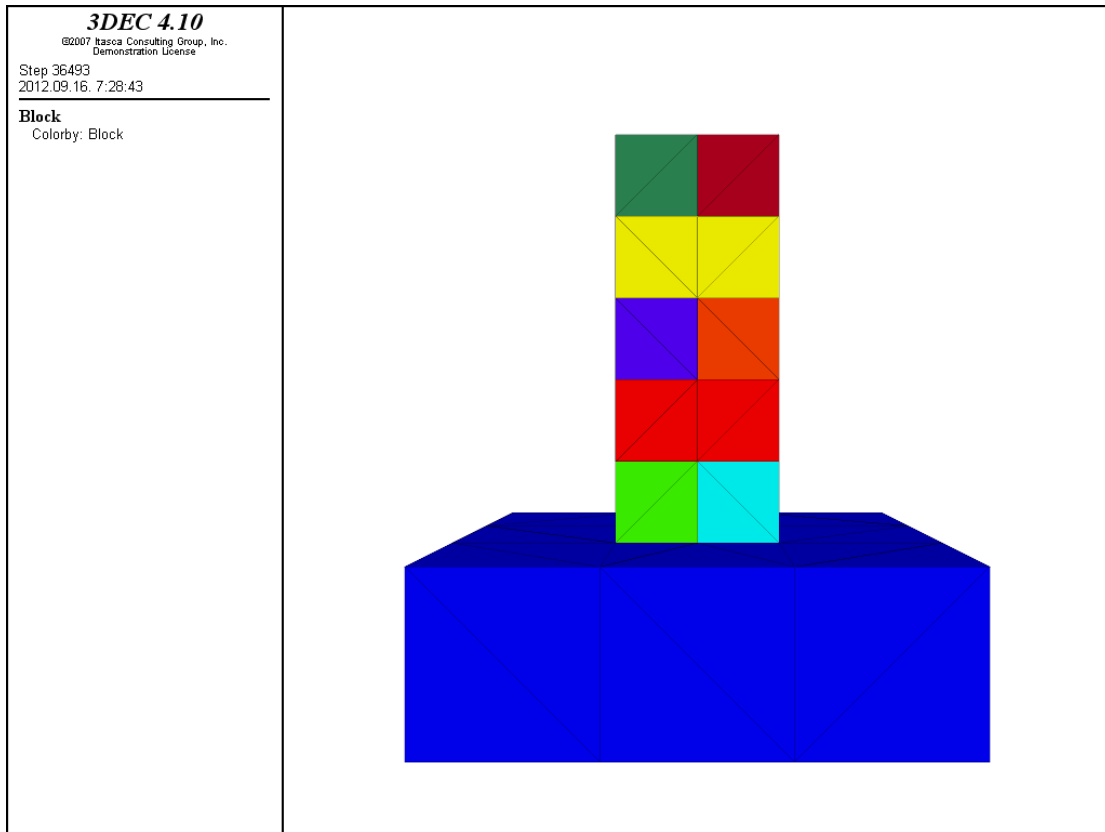


Fig. 65.: Result of „plot block” command in case plot sets are not applied

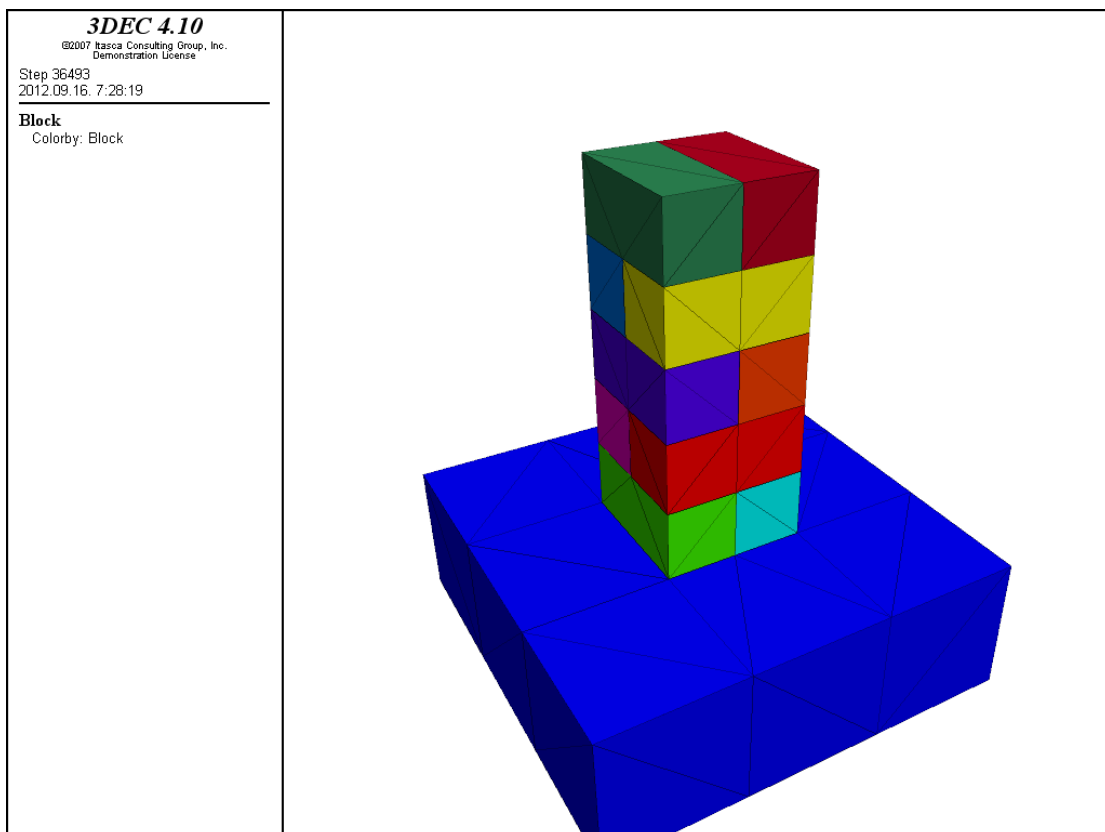


Fig. 66.: Result of „plot block” command in case plot sets are applied

By default in case „plot block” command is applied, the colors are chosen automatically by the software. If „plot block clear” command is applied the automatic colors are cleared so the structure turns into black (Fig. 67.).

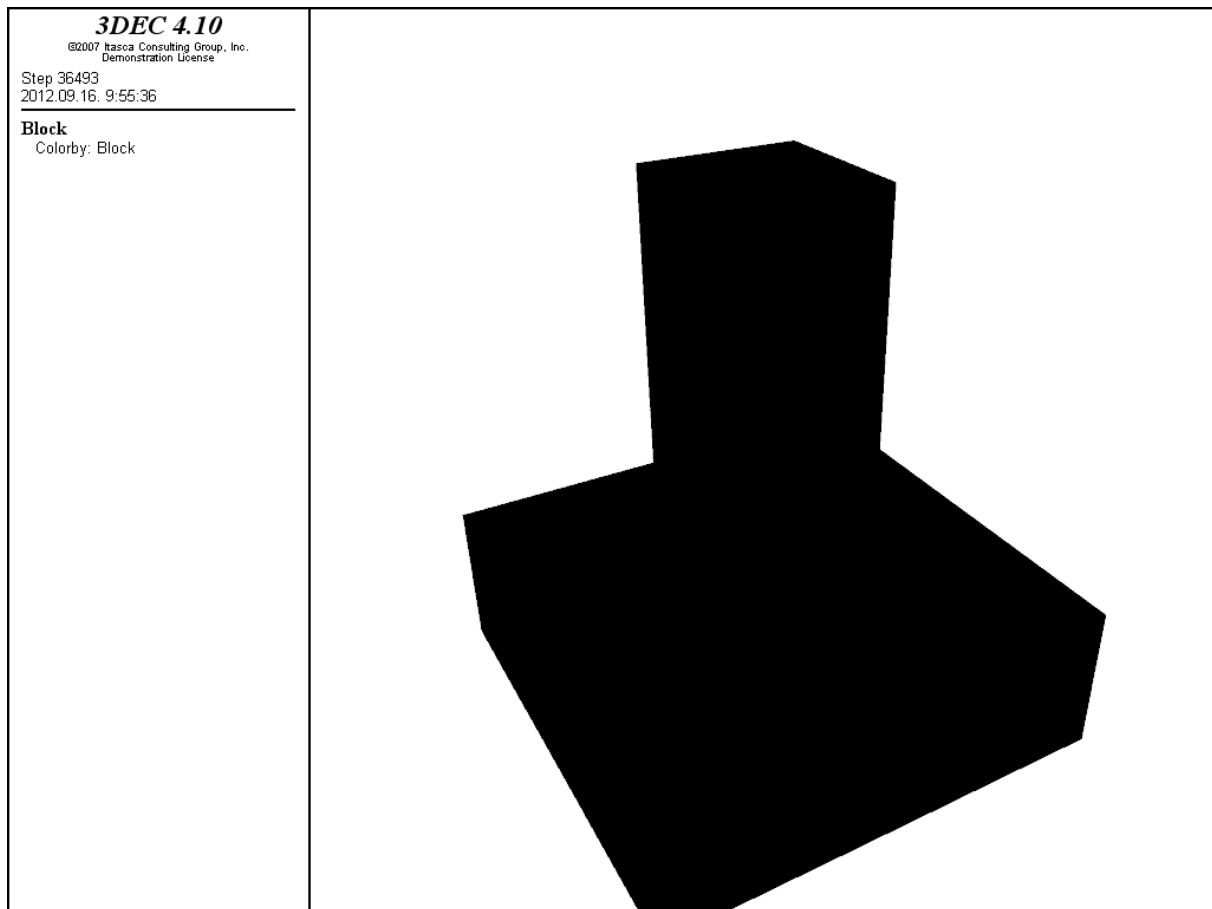


Fig. 67.: Result of „plot block clear” command

If regions are defined for the structure the created parts are treated as one block, to represent this they have the same color.

For example if I name the upper levels of the column „region 1” and join the blocks, only two colours can be seen at the resulted picture (Fig. 68.).

```

;hiding the base-block
    hide range z -0.2,0.
;marking the upper levels of the column
    mark region 1
;joining the blocks
    join on
;make the hidden parts visionable
    seek

```

Pictures like Fig. 65. and Fig. 66. can be generated with „plot block colourby block” command. So it gives the same result as „plot block” command. The picture on Fig. 68. can be generated with command „plot block colorby region” too. But for this the joining have to be released before with using the following command (Fig. 69.):

```

join off

```

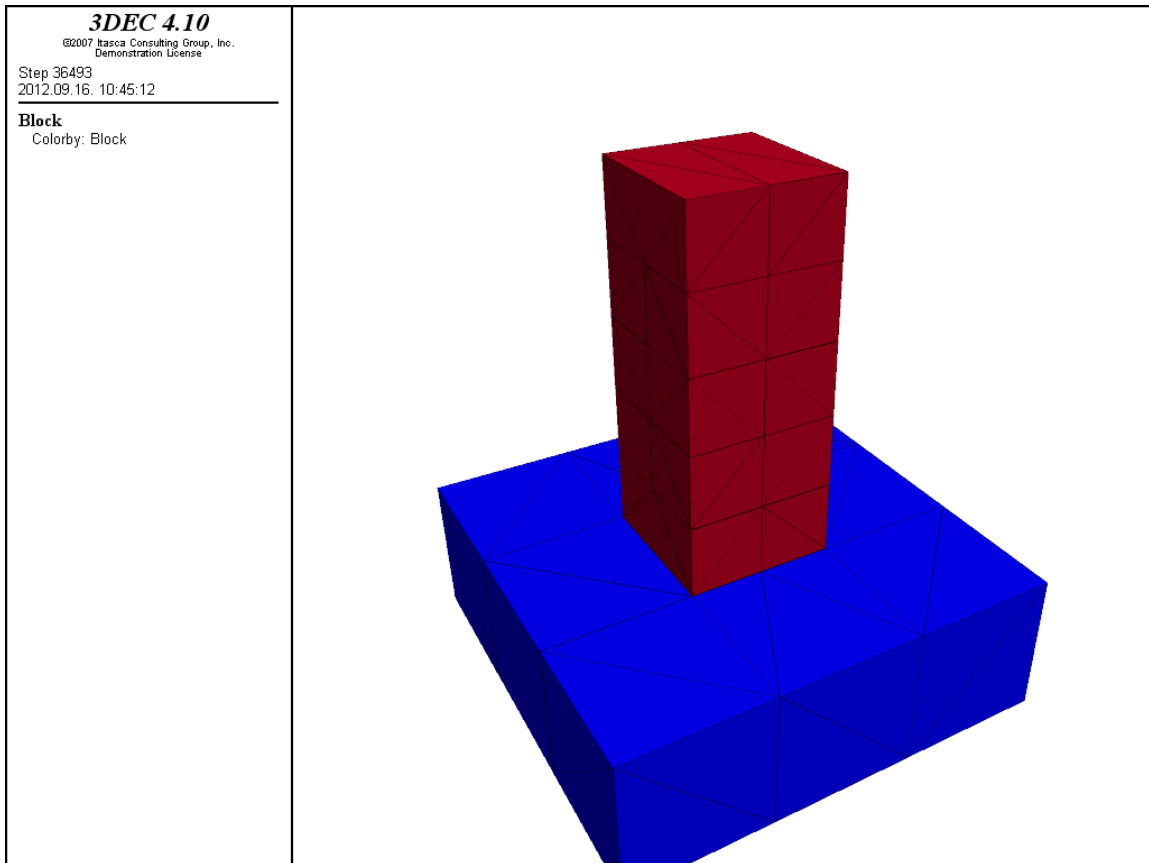


Fig. 68.: Result of „plot block” command in case only two regions are defined

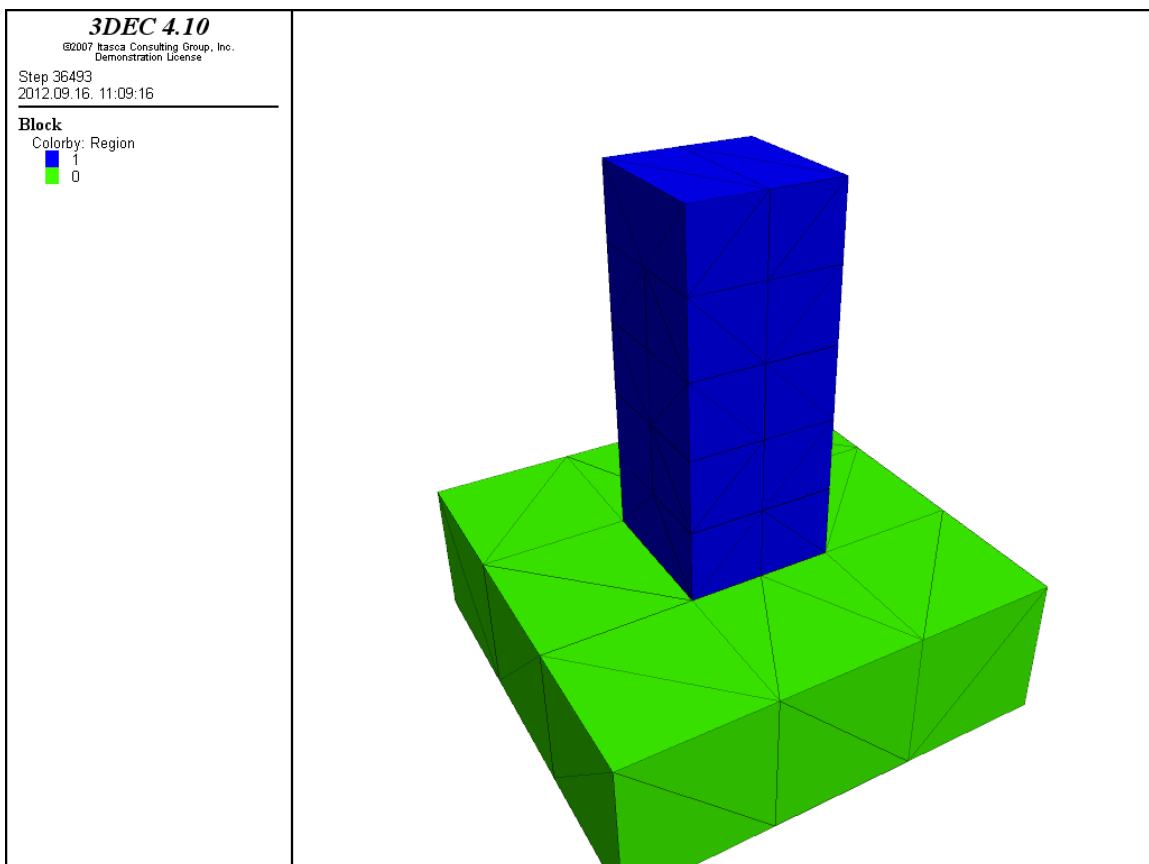


Fig. 69.: Result of „plot block colourby region” command in case joining is released

„plot applied”:

This command plots vectors indicating applied quantities on the model. Each Cartesian component may be indicated separately. Available types are: **load** (applied force vectors) and **velocity** (applied velocity vectors).

As in the previous sections „above au” switch keyword can be added to the main command. Furthermore the following subcommands can be used for manipulation of vectors displayed by „applied load or velocity” (the list is not complete, see Command Reference for further commands):

- „below au” or „below value” – specify which color to use for values below the minimum
- „colorbymagnitude bool” – vectors are drawn in a color determined by a contour color ramp based on magnitude (bool means „on” or „off”)
- „interval au” or „interval value” – specifies the size of the interval between colors
- „ramp” – specifies which color ramp the contour will use to go from minimum to maximum (greyscale or rainbow)
- „range” – the range where represented objects can be found

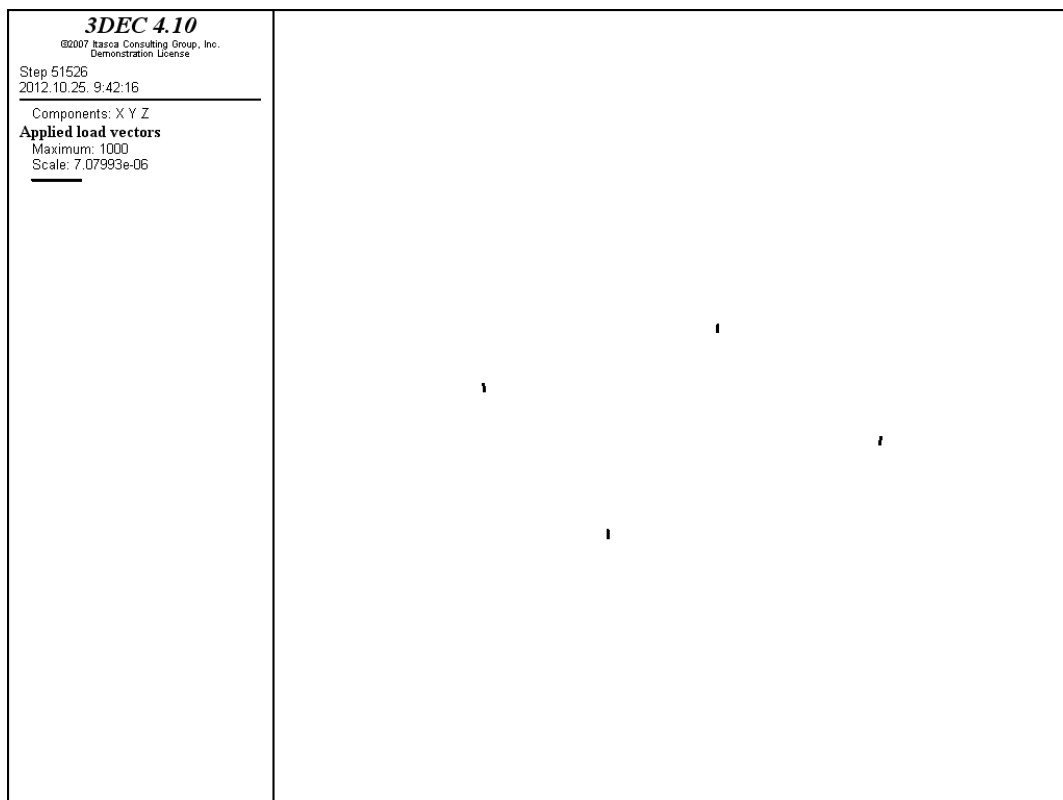


Fig. 70.: Result of „plot applied load above au” command

As it can be seen on Fig. 70. the result of „plot block applied load above au” command is not expressive enough, because the geometry of the structure is not represented. Besides the lack of the contour of geometry the size of the vectors is too small. This causes that in case contour is represented the vectors can't be seen even if they are plotted in some kind of color. So besides the change of the color of the vectors their width can be manipulated. By default the width of lines is 1, on Fig. 72. size is set to 3. The structure was loaded with four pointed-loads on the top plane each of them is 1000N. A more expressive picture can be created if „plot block fill off” command is applied and the vectors are added to the framework. The vectors can be represented with the command „plot add applied load line color red”. In this

case vectors are drawn with red lines and the „add” command ensures that the red line is represented on the same picture as the framework. (Fig. 71.)

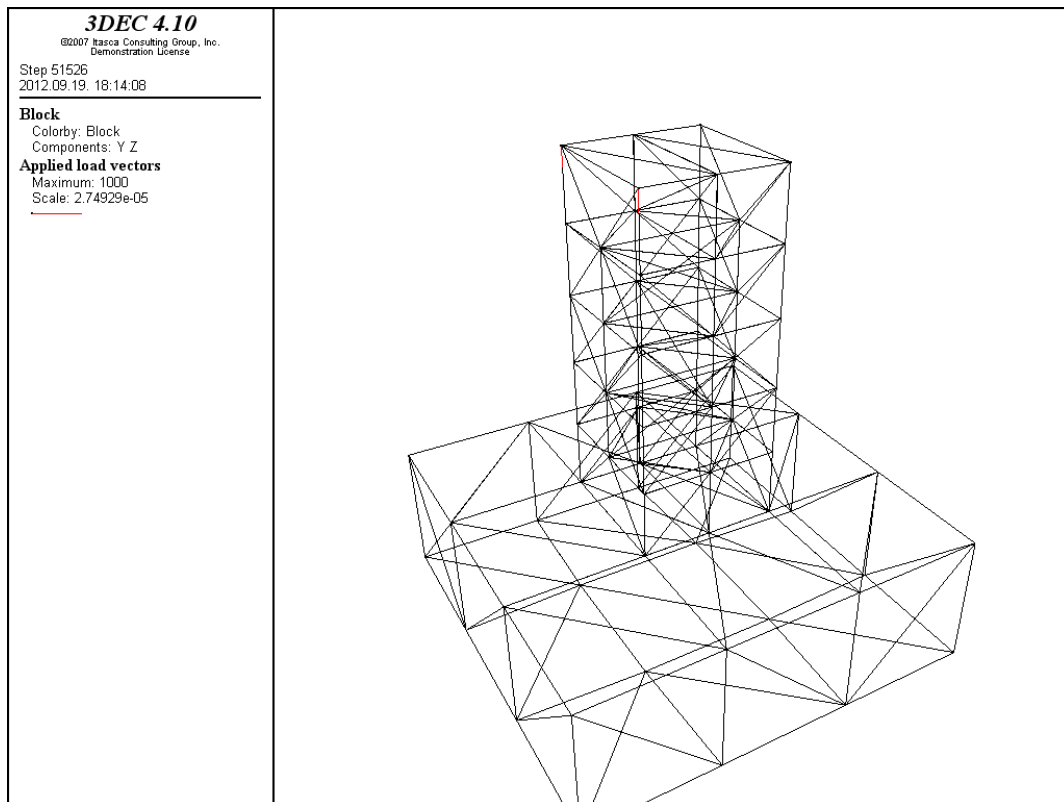


Fig. 71.: Result of „plot block fill off” and „plot add applied load line color red” command width value is set to default (1)

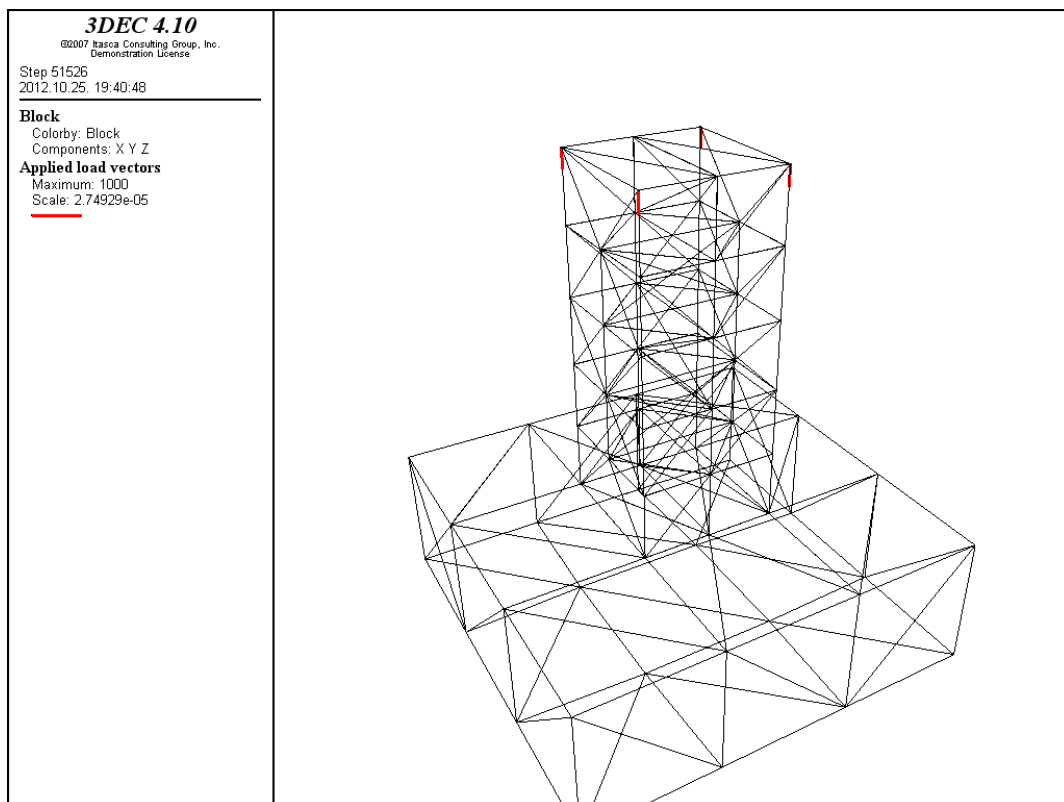


Fig. 72.: Result of „plot block fill off” and „plot add applied load line color red width 3” commands

„plot axes”:

This command sets the Cartesian coordinate axes on the screen. When a plot item is represented on the screen, axes can be set only with the „plot add axes” command. In case „add” command is missed only the axes are represented.

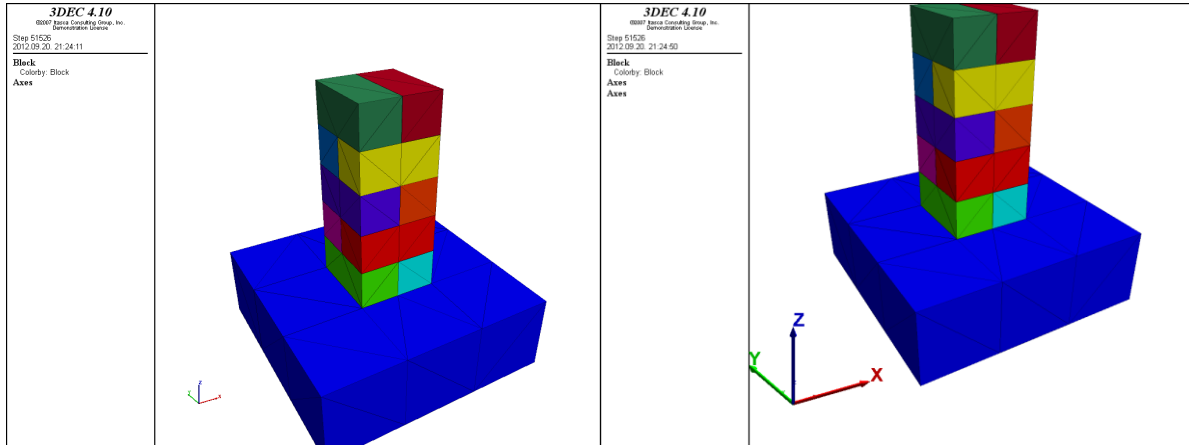


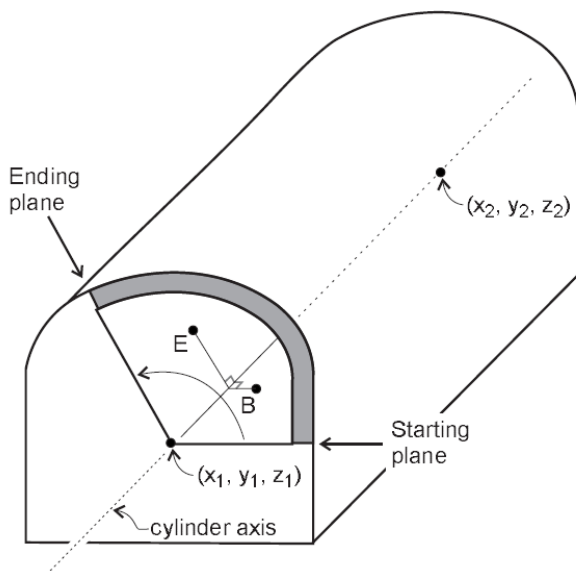
Fig. 73.: Result of „plot axes” command in case of default-sized coordinate axes and an enlarged one

As it can be seen on Fig. 73. by default the size of the axes is too small, this can be changed with the command „plot add axes scale 20”. Besides the „size” command several other keywords are available with „axes” command. For instance the most useful keywords are the following: „position”, „transparency”, „xaxis”, „yaxis”, „zaxis”.

There are some reinforcement elements that usually used at geotechnical problems, but sometimes they can be applied to structural tasks too. Now I am only mentioning them, for further details see Command Reference – Structure section:

1. axial reinforcement: this command defines a normal restraint even across discontinuities. If the model is build up with deformable blocks discretization has to be finished before reinforcements are specified. The command is the following:

```
struct axial x1 y1 z1 x2 y2 z2 prop n
struct prop n rkaxial=... rlength=... rstiffness=... rsstrain=... rstrain=... rultimate=... rushear=...
```



As it can be seen, two bunch of coordinates defines the starting (x_1, y_1, z_1) and the ending point (x_2, y_2, z_2) of the reinforcement element. The subcommand „property number” refers to the structural properties defined with the „struct prop” command. At the represented script only two types of probable element-properties were shown (rkax, rult), for further details see Command Reference.

2. beam element: the command is generating beam elements along the tunnel axis.

Fig. 74.: signs at beam command

Two types of elements' positioning can be defined: a series of beams and a single beam. In case of the „multi-beams” the following command is used:

```
radial gen x1 y1 z1 x2 y2 z2
seg naxial nradial prop n
begin xb yb zb end xe ye ze connect
struct porp n area=... density=... emod=... I=... I1=... I2=... J=... nu=... S1=... ycomp=... yield=...
```

x_1, y_1, z_1 and x_2, y_2, z_2 coordinates refer to the starting and ending point of the tunnel axis. „naxial” defines the number of sections, that enclosed by the starting and ending plane („begin”, „end”). (Fig. 74.)

„nradial” gives the number of beam elements in each section. „connect” command connects new elements to the existing nodes if they are at the same location. Single beams can be added with:

```
node x y z id id prop n tol tol
element naid nbid id id prop n
```

The first command places the beam according to the node that identified with it's „ID”, „tol” is the distance tolerance to search for a face to place the node. The second one is according to the „ID” of the element, not for the node. Possible „structural properties” are the same as in case of „multi-beams”.

3. cable element: this command allows use of tendons even if they are grouted. The required input command is the following:

```
struct cable x1 y1 z1 x2 y2 z2 &
seg ns prop n tens t
struct porp n area=... emod=... kbond=... sbond=... thexpansion=... ycomp=... yield=...
```

As it can be seen, as in case of axial reinforcement geometry of tendons are given with the coordinates of starting (x_1, y_1, z_1) and ending (x_2, y_2, z_2) point of the cable. Number of segments which cable is divided into is defined with „seg ns”. „property number” is defined in the same manner as at the previous structural commands and pretension value can be added with „tens t” subcommand.

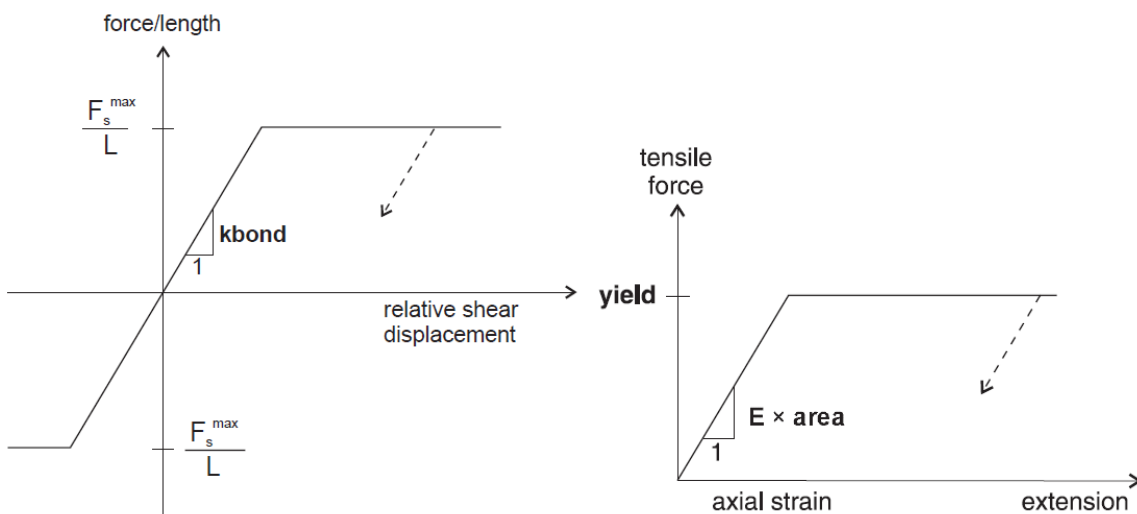


Fig. 75.: Grout material (on the left handside) and cable material (on the right handside) properties for cable elements

4. liner element: it is obvious that thickness of tunnel liners are usually negligible compared to their diameter. So the liners usually represented with their theoretical axis. 3DEC is providing a pre-defined structural element that allows this simplification to be taken into account.

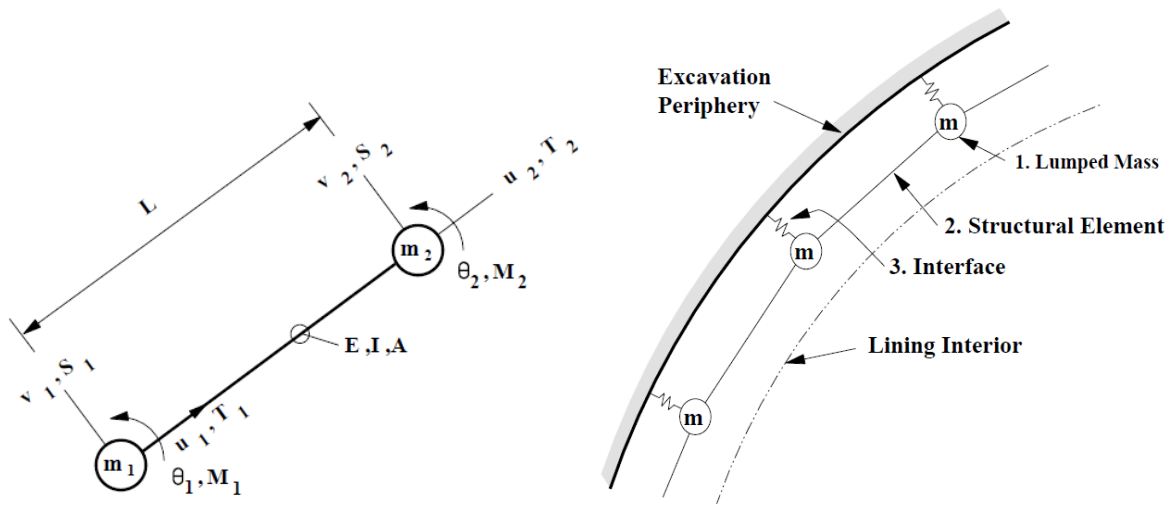


Fig. 76.: Liner elements

For two-dimensional analysis liners are beam elements which include non-linear behavior. (Fig. 76.) With these structural parts axial, transverse and flexural loads can be taken into consideration. For three-dimensional analysis triangular elements are used instead of beams (Fig. 77.)

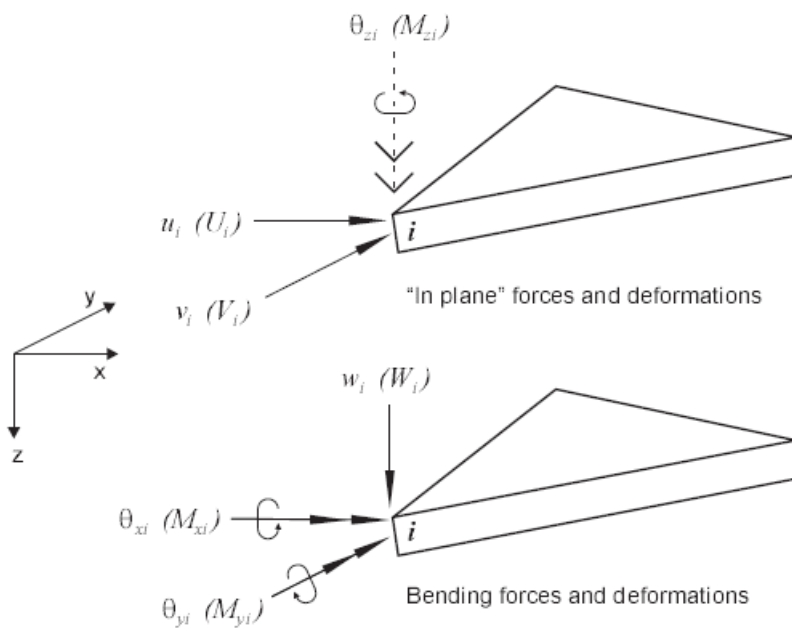


Fig. 77.: Liner elements for three-dimensional analysis

radial x1 y1 z1 x2 y2 z2 &
 seg na, nr prop n cylinder r &
 begin xb yb zb end xe ye ze connect
 struct porp n cohesion=... emod=... kn=... ks=... nu=... tensile=... thexpansion=... thick=...

x1, y1, z1 and x2, y2, z2 are the coordinates of beginning and ending points of the axis of tunnel. „nr” is the number of radial and „na” is number of axial segments into which the liner

is divided. By default the liners are defined at the whole cylindrical surface, but with the use of „begin” and „end” coordinates liners can be added for only a segment of the tunnel.

The previously demonstrated structural elements can be used in special cases for masonry structures, but now they are not taking part of this booklet. So the following plot commands according to these structures won't be discussed: „plot axial”, „plot axialcontour”, „plot axialvector”, „plot beam”, „plot beamcontour”, „plot beamvector”, „plot cable”, „plot cablecontour”, „plot cablevector”, „plot liner”, „plot linercontour”, „plot linervector”, „plot bond”.

„plot bcontour”, „plot block”, „plot contour”, „plot history” commands were demonstrated previously, so they are not mentioned in this section.

„plot boundary”

If only the gravity is added to the structure and the following script is used for displaying boundary, the result can be seen on Fig. 78. It is represented that the whole lower plane of base block is fixed (velocity displacement is defined with zero value).

```
plot boundary active  
plot add block fill off
```

First line of the script is used to represent the active boundaries, the second line was applied to show the positions of gridpoints where boundaries were active. If we are studying the Command reference it turns out, that a lot of subcommands can be used to manipulate the resulted picture.

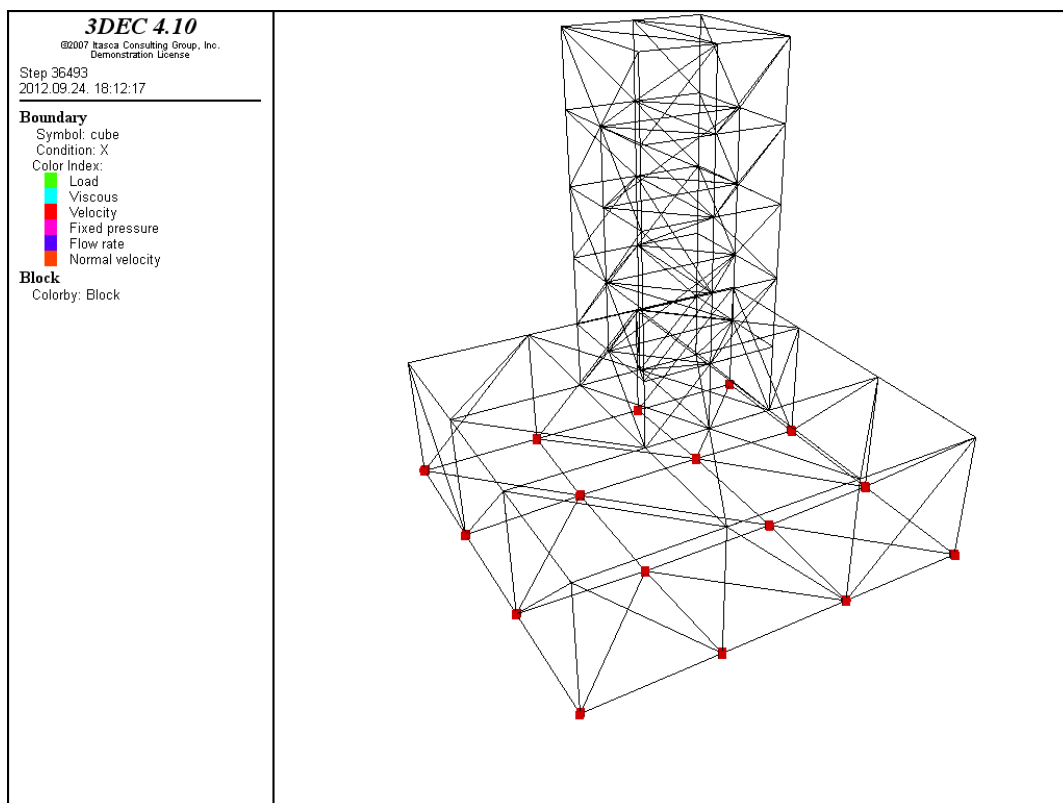


Fig. 78.: Result of plot boundary command

„plot displacement”

This command is used to display displacement vectors at vertexes. The following switch keywords can be used with the command:

plot displacement above au
plot displacement below au
plot displacement interval au
plot displacement maximum au
plot displacement minimum au

plot displacement colorbymagnitude

The first five commands give the same results if the default values are applied. Resulted plot output can be seen on Fig. 79. The resulted picture contains the maximal value of displacements, it can be seen on the left hand side. Furthermore displacement vectors at vertexes are represented with black lines proportional to the displacement values.

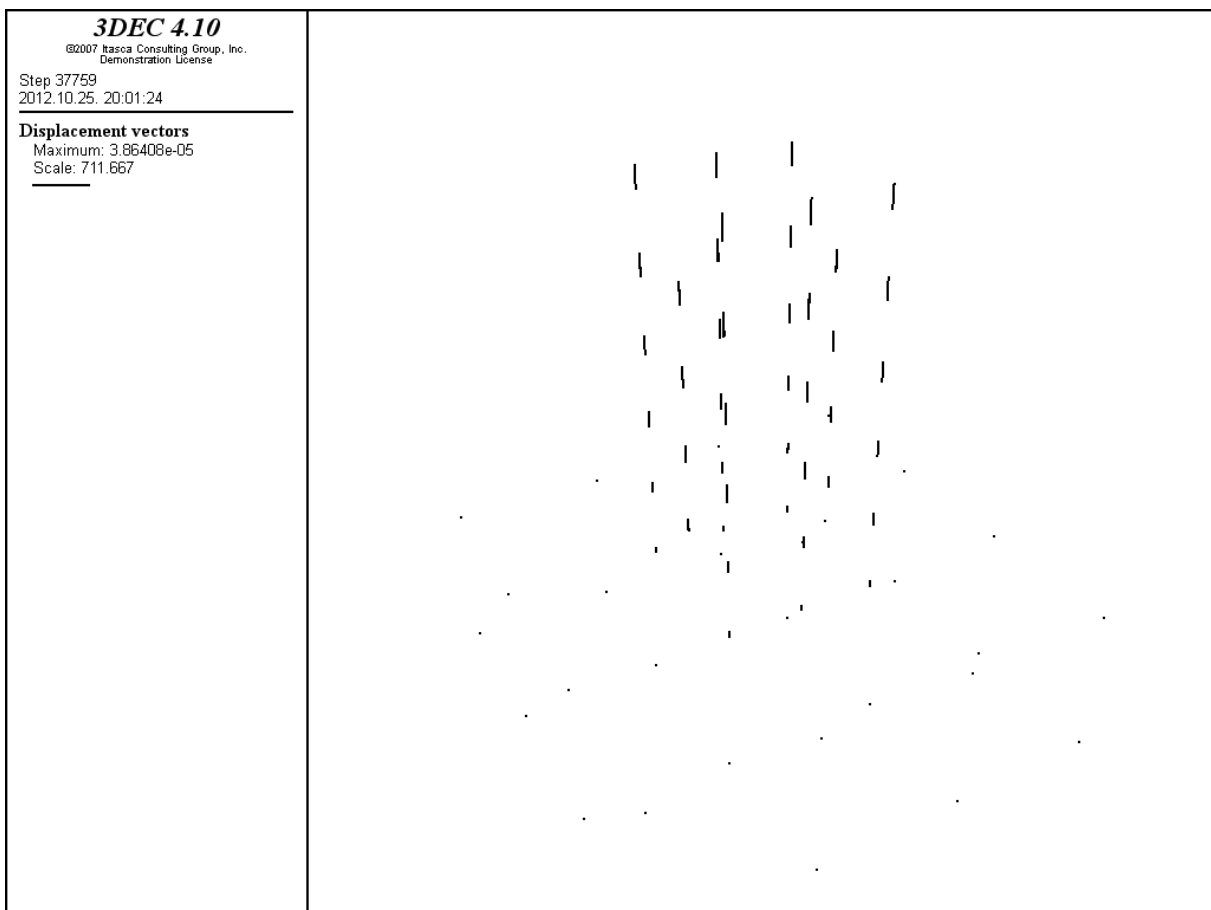


Fig. 79.: Result of „plot disp line width 2” command

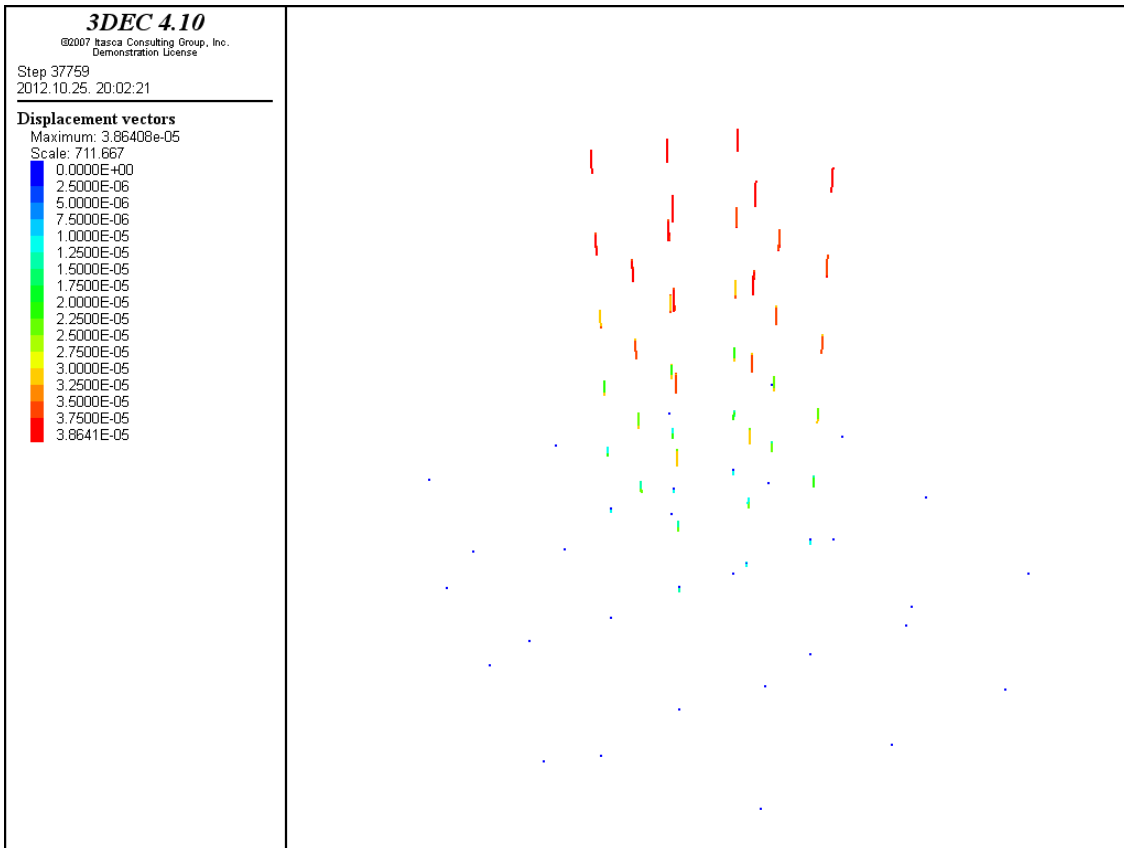


Fig. 80.: Result of „plot disp colorbymagnitude line width 2” command

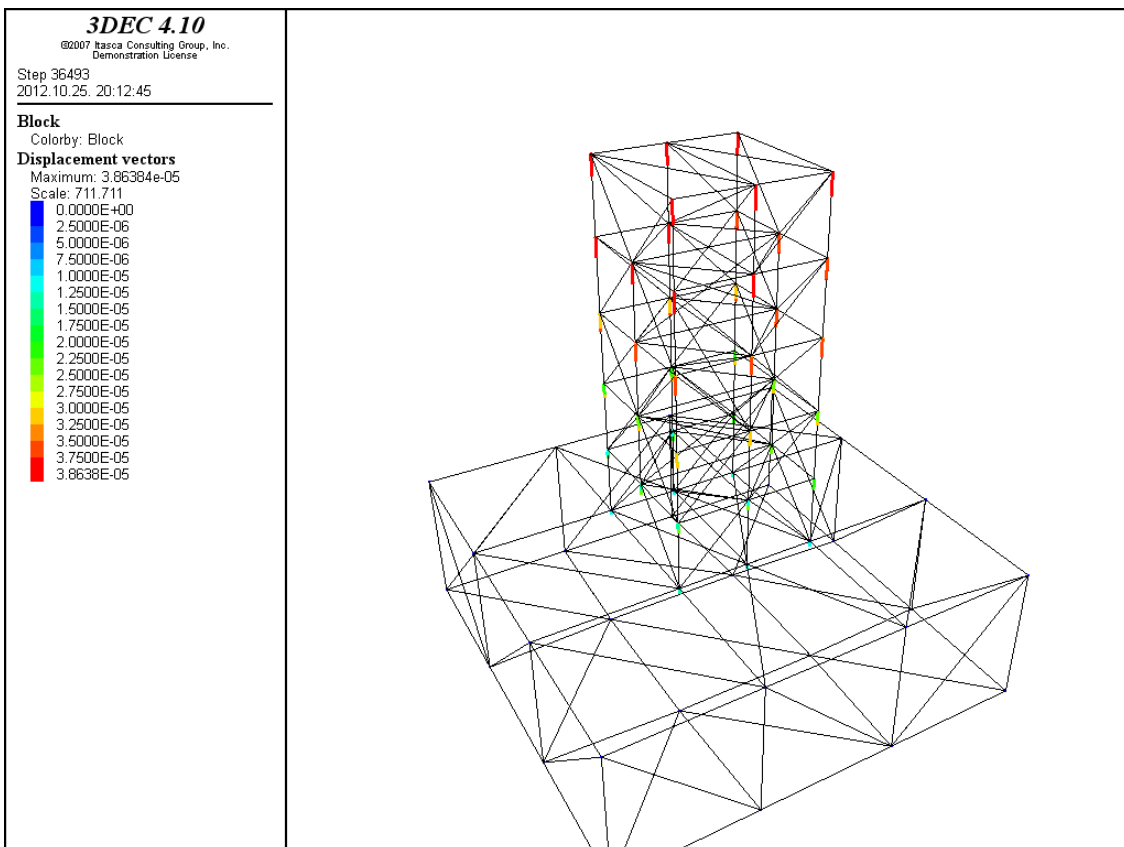


Fig. 81.: Result of „plot disp colorbymagnitude line width 3” command if „plot add block fill off” is applied too

Fig. 80. is the same as Fig. 79., the only difference between the two pictures is that Fig. 80. is more expressing, because it is representing displacement vectors with a special color ramp besides the proportional sizes. On Fig. 81. I used the previously demonstrated „fill off” command, as it can be seen now it is not so expressive as it was for example at „plot boundary” command. The generated mesh-grid covers the coloured vectors. It is obvious that in case of a mesh with normal density the vectors without the mesh-grid is resulting a well-represented geometry. Results of Fig. 79., 80., 81. are all generated under gravitational loads.

„plot dxf”

With the help of this command the loaded dxf files (converted Autocad files) can be manipulated.

„plot fknot”

This command is not reviewed in this paper because it’s connected to fluid models. If the command is applied locations of fluid flow knots can be plotted.

„plot fob”

If this command is applied to the modell the out-of-balance force vectors will be represented at the vertexes.

```
plot fob above au
plot fob below au
plot fob interval au
plot fob maximum au
plot fob minimum au
```

```
plot fob colorbymagnitude
```

According to the previously demonstrated script it is obvious that the available subcommands are the same at every command. This makes their use easier. If only gravitation is acting on the model the results can be seen on Fig. 82., 83. In case of a model where besides the gravitational loading four concentrated loads (each of them is 1000 N) are acting on the top-plane of the column, results are the same as in case of gravitational loading.

According to the pictures the demonstrated results are the following: value of maximum unbalanced force, unbalanced force vectors with proportional sizes. With the use of „color by magnitude” command the results can be made more expressive.

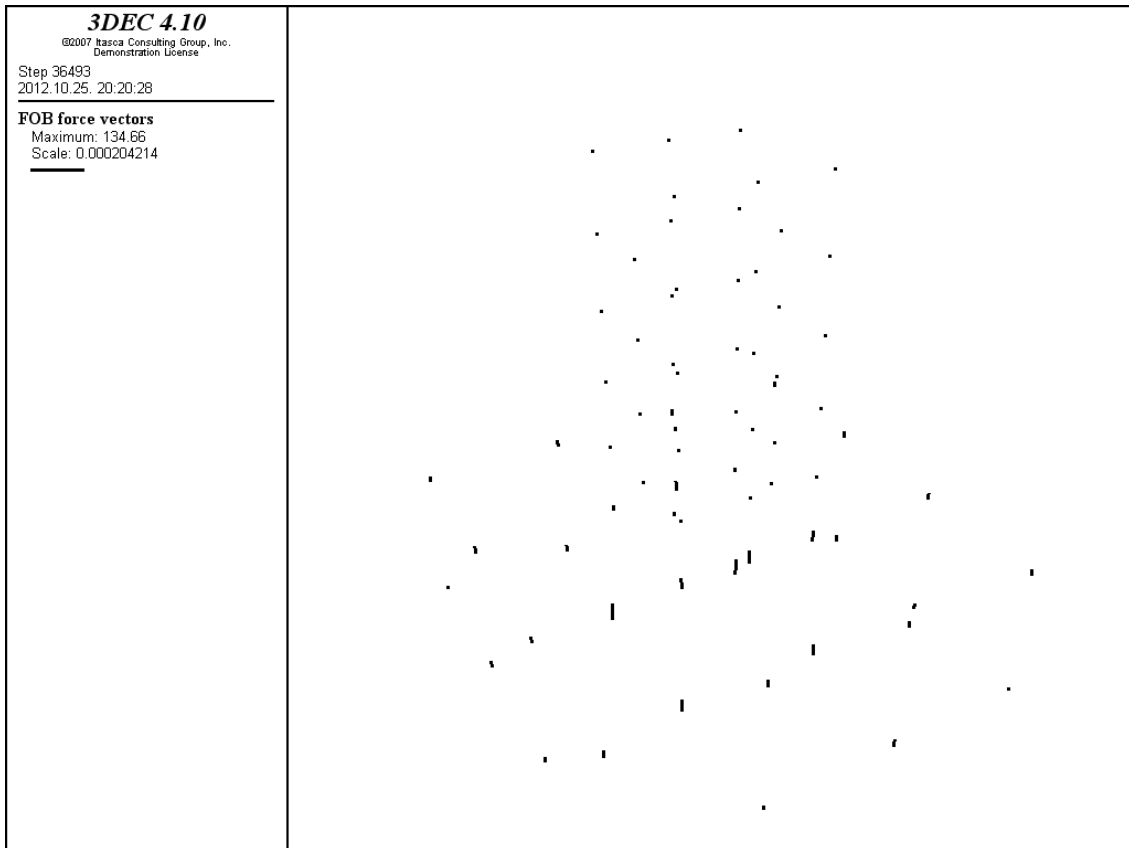


Fig. 82.: Result of „plot fob line width 2” command

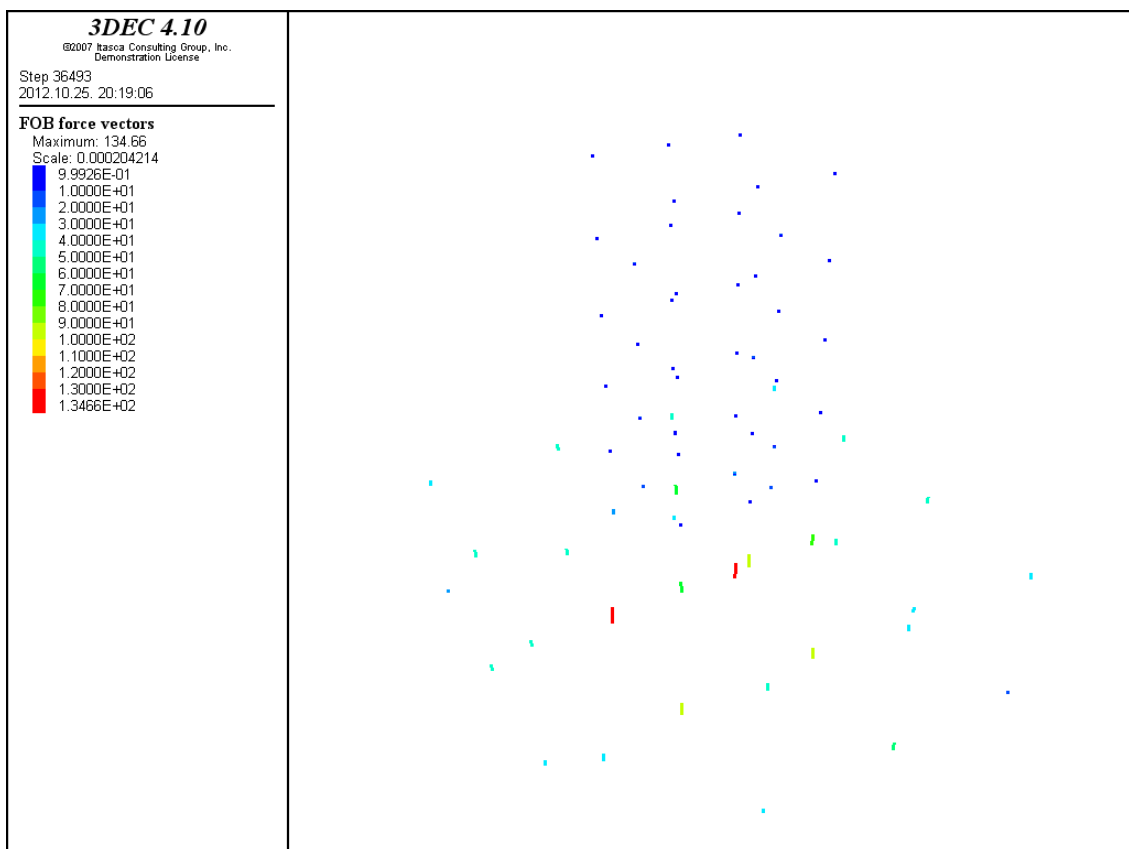


Fig. 83.: Result of „plot fob colorbymagnitude line width 3 point size 3” command

The next commands are connected to fluid-flows in the discontinuities of rocks. In 3DEC there is a built-in experimental fracture fluid-flow logic. The flow geometry consists of flow planes along the faces of blocks. There are flow pipes at the intersections of the flow planes along the edges of the faces. There are flow knots at the intersections of flow pipes (corners of blocks).

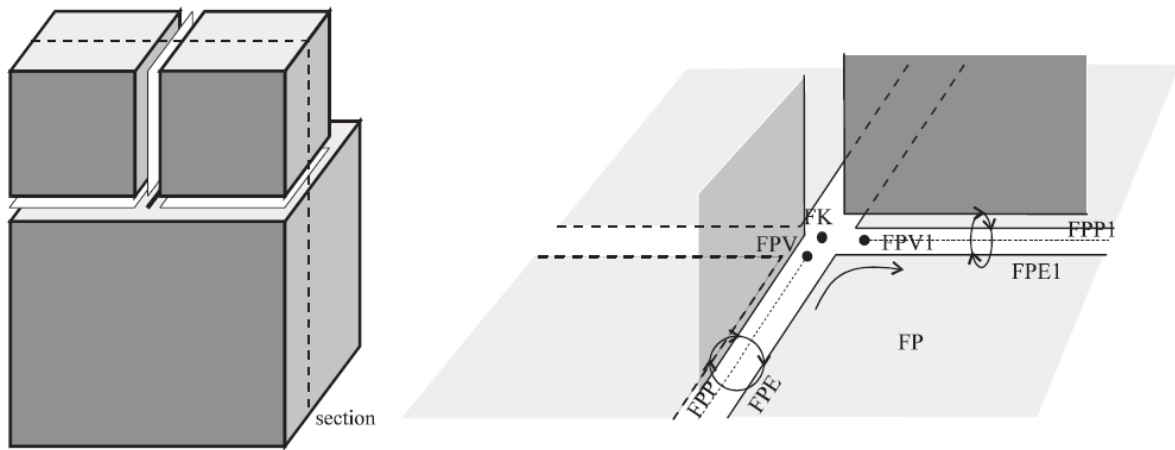


Fig. 84.: Fluid-flow planes, pipes and knots

„plot fpcontour”

This command is not reviewed in this paper because it's connected to fluid models. If the command is applied contours of various quantities on flow planes can be plotted.

„plot fplane”

This command is not reviewed in this paper because it is connected to fluid models. If the command is applied locations of flow planes in the model can be plotted.

„plot fpvector”

This command is not reviewed in this paper because it's connected to fluid models. If it is applied, fluid plane discharge vectors can be plotted.

„plot joint”

The joints in the model can be represented by the command „plot joint active”, result can be seen on Fig. 85. Colours were chosen by joint-id.

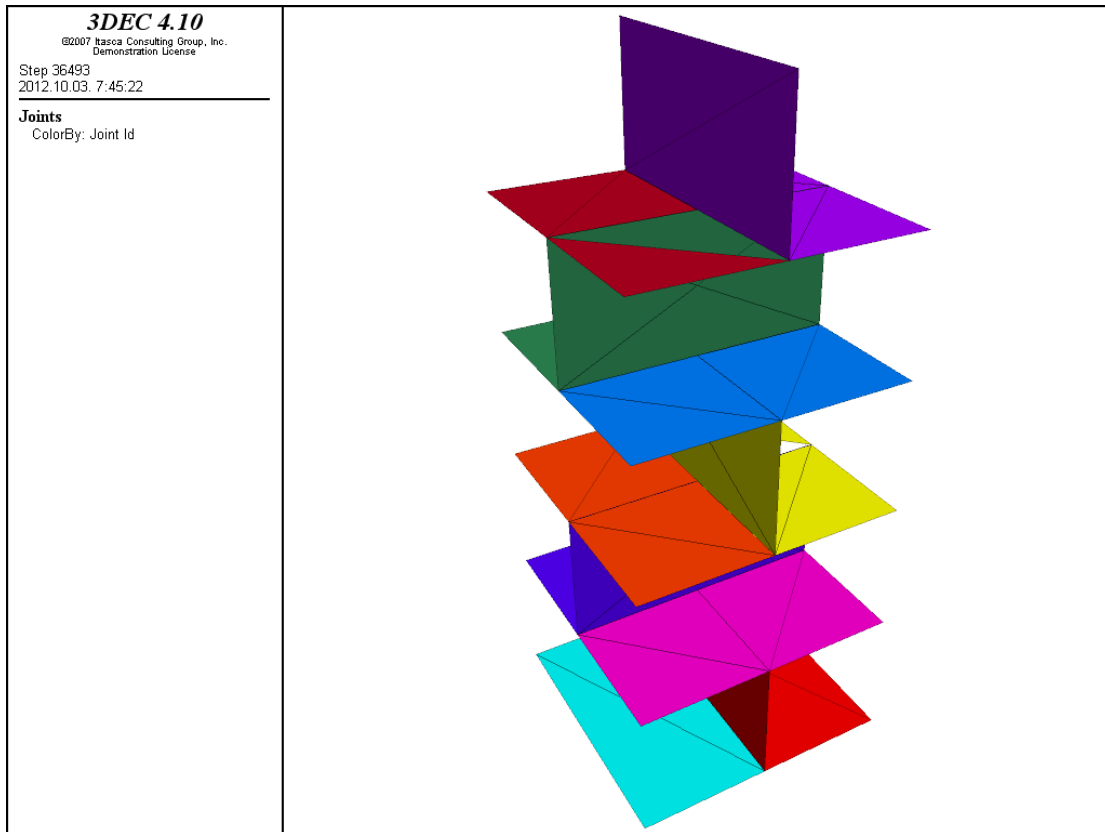


Fig. 85.: Result of „plot joint active” command

On Fig. 86. result of „plot joint backface on” command can be seen, the difference between the two figures 85 and 86 is obvious. „Backface on” command is not fill every joint-planes, but the contours of the planes are shown, sometimes this can make the result more useful.

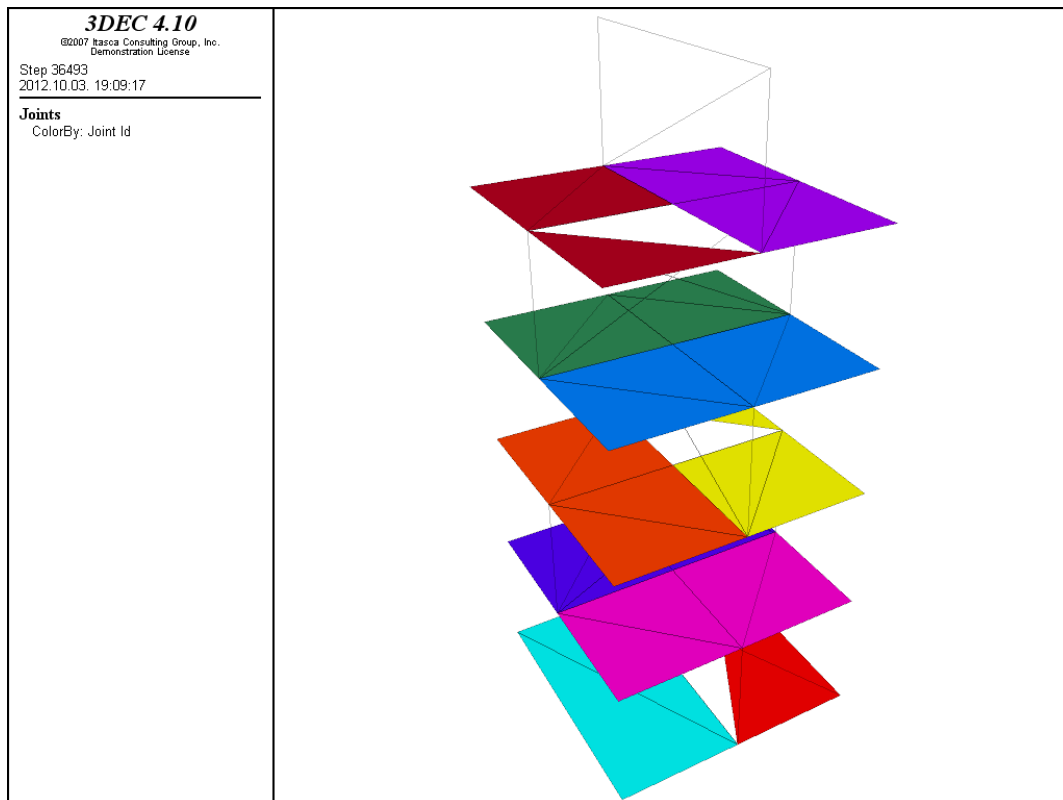


Fig. 86.: Result of „plot joint backface on” command

Plot joint clear command is similar to „plot block clear” (on page 66) both of them are clear all color information. This results a joint representation like the one on Fig. 87.

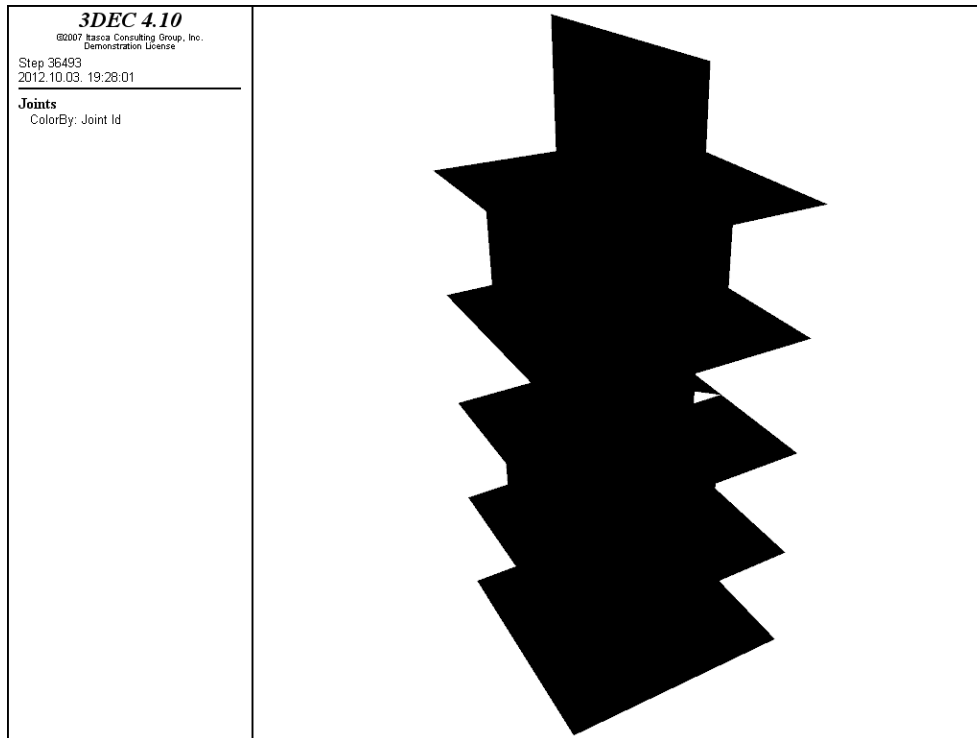


Fig. 87.: Result of „plot joint clear” command

„Plot joint colorby jointid” results the same picture as the one on Fig. 85. In case „plot joint colorby material” and „plot joint colorby model” commands are applied results can be seen on Fig. 88. and 89.

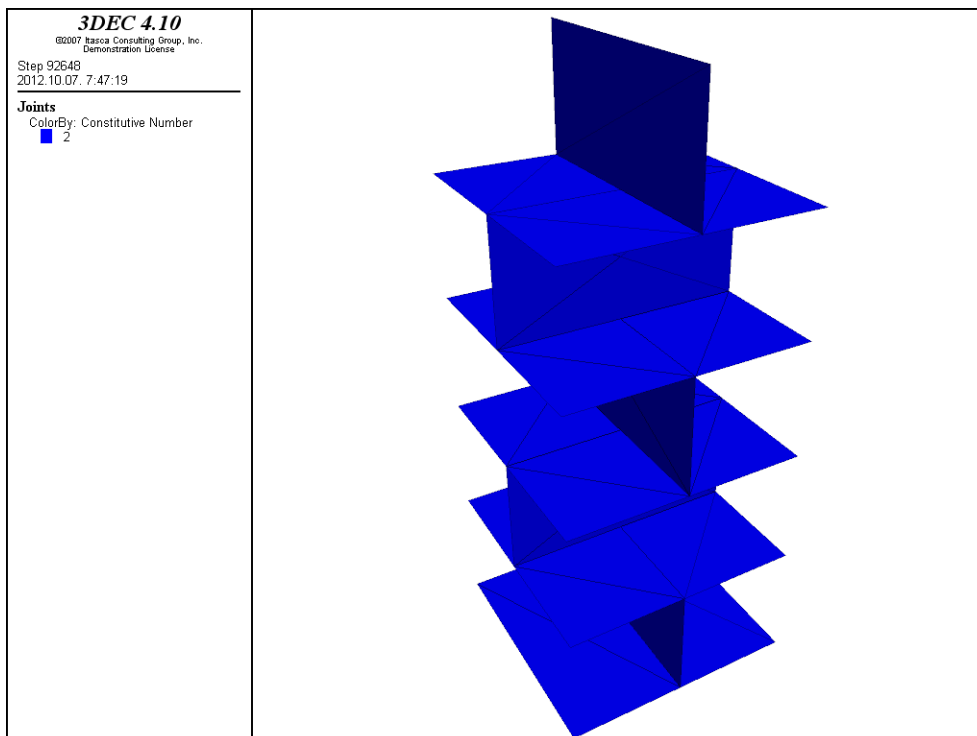


Fig. 88.: Result of „plot joint colorby material” and „plot joint colorby model” commands in case the first script was used

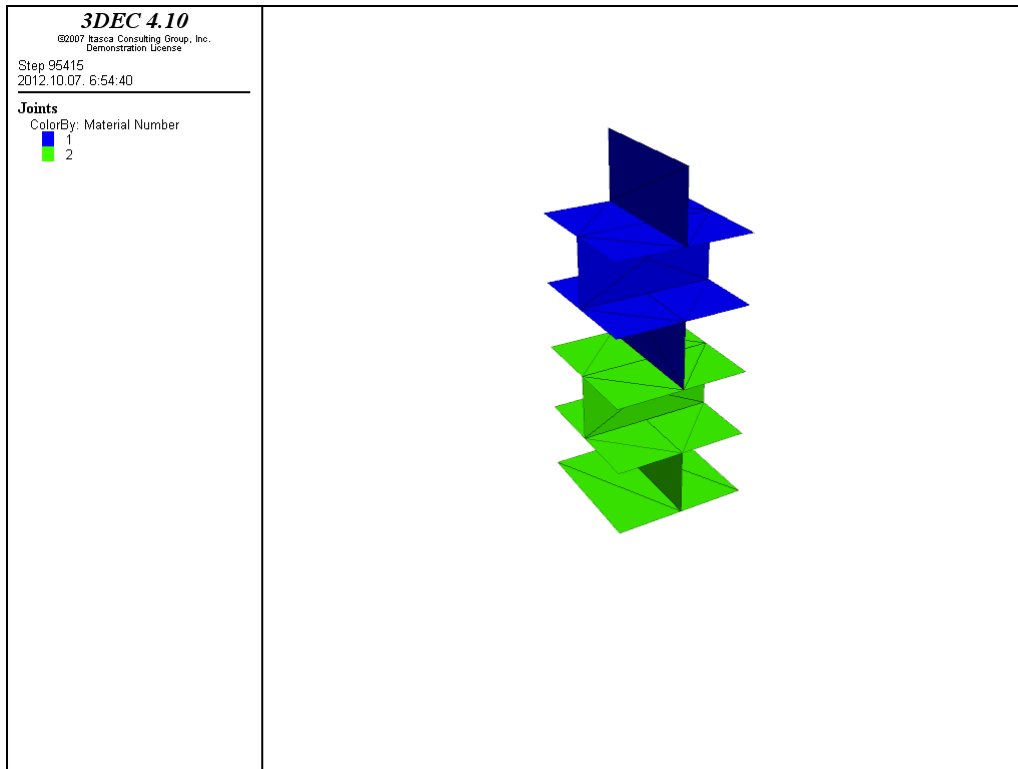


Fig. 89.: Result of „plot joint colorby material” command in case the second modified joint properties were used

```

;joint properties (Coulomb-slip model)
;normal stiffness (1000 MPa/m) and shear stiffness (100 MPa/m)
;friction angle 25°
  change jcons=1
  prop jmat=1 jkn 1.e9 jks 1.e8 fric 25.
;joint properties (Coulomb-slip model)
;normal stiffness (1000 MPa/m) and shear stiffness (100 MPa/m)
;friction angle 25°
;1st joint material
  change jcons=1
  prop jmat=1 jkn 1.e9 jks 1.e8 fric 25.
  change jmat=1 range z -0.21, 0.21
;2nd joint material
  change jcons=1
  prop jmat=2 jkn 1.e8 jks 1.e7 fric 15.
  change jmat=2 range z 0.21, 0.51

```

The colors of joint material can be changed by the „plot add joint colorlist red” command. As it can be seen on Fig. 90. the colour changed to red, but any other colors can be used.

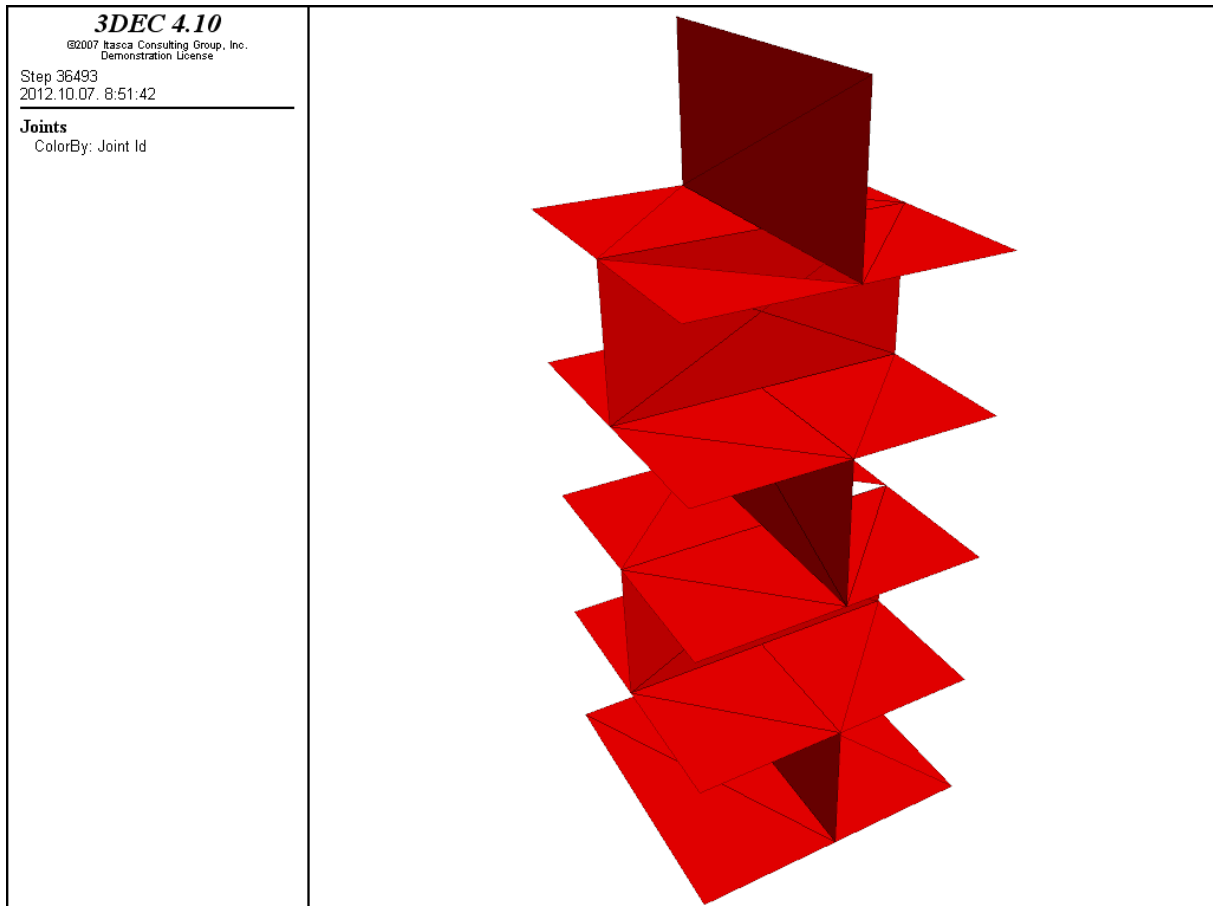


Fig. 90.: Result of „plot joint colorlist red” command

Another „plot joint” command is the „plot joint fill off”, this command plots the joint faces but ignores their color fill, so only the edges of the planes remain. (Fig. 91.) „Plot joint wire” command gives the same result as „plot joint fill off”.

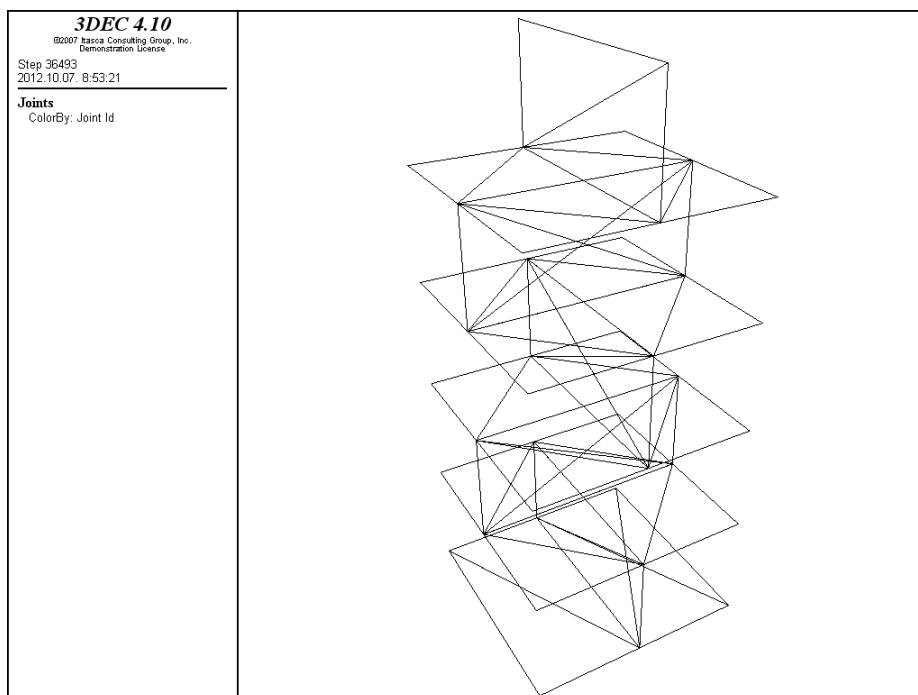


Fig. 91.: Result of „plot joint fill off” command

„Plot joint wire” command gives the same result as „plot joint fill of” (Fig. 91.). Transparency of wire can be changed by the „plot joint wiretrans” command, the default value is 75. If the value is reduced for example until 1, the wireframe is getting more robust (thickness of wires). „Plot joint wireframe off” command gives the result as if the value of wiretrans is set to 100. (Fig. 92.)

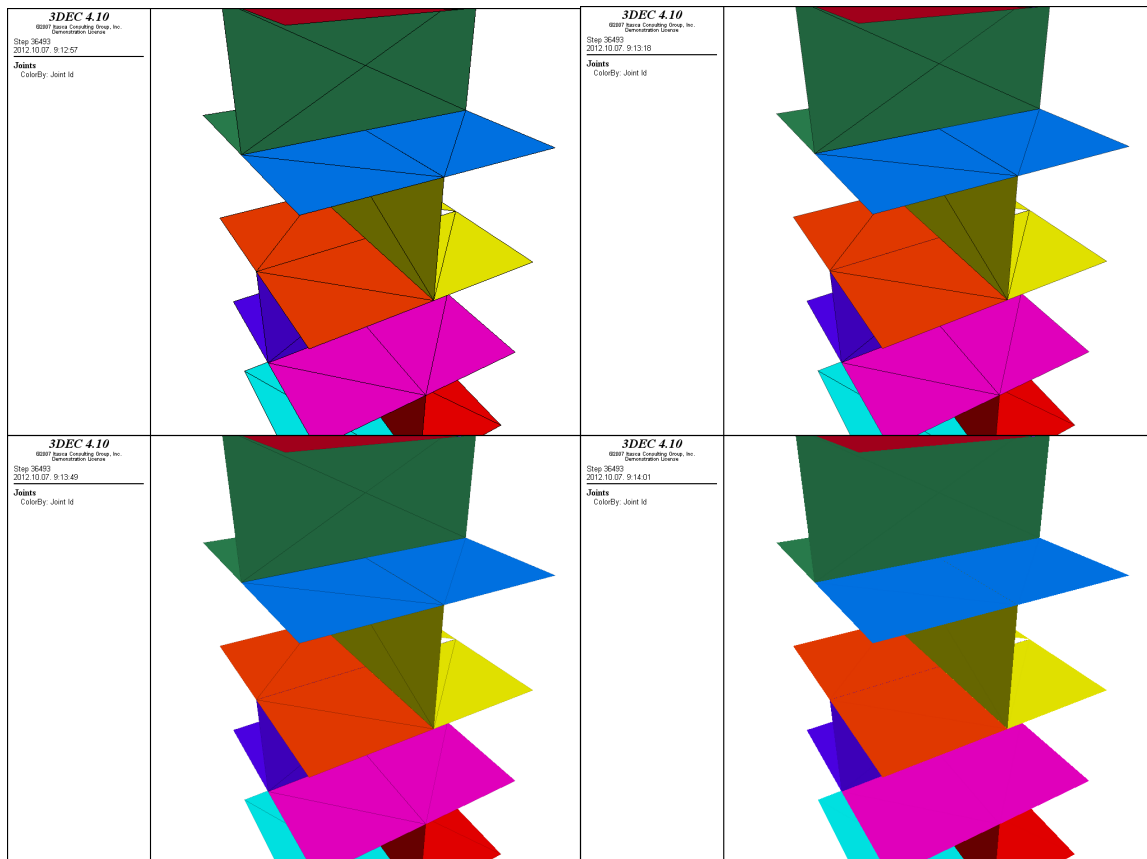


Fig. 92.: Results of „plot joint wiretrans 1”, „wiretrans 50”, wiretrans 85”, „wiretrans 100” „plot jointcontour”

„Plot contour” commands were demonstrated in previous sections, the „plot jcontour” command is working the same as the previously used ones. There are four quantities that can be represented on the surfaces of joints normal displacement (ndisplacement), normal stress (nstress), shear displacement (sdisplacement), shear stress (sstress). On the following I try all of these commands in case gravity is acting on the structure.

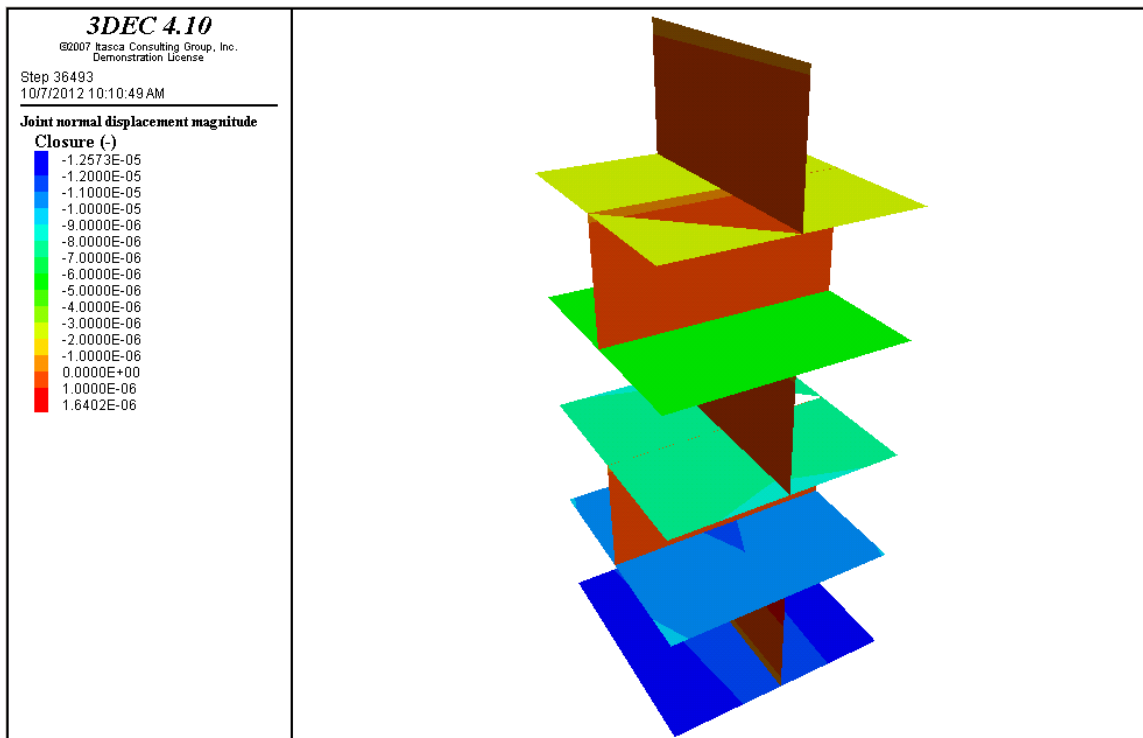


Fig. 93.: Result of „plot jointcontour ndisplacement below au”

Result of „plot jcontour ndisplacement below au” (Fig. 93.), „plot jcontour nstress max au” (Fig. 94.), „plot jcontour sdisplacement above au” (Fig. 95.), „plot jcontour sstress above au” (Fig. 96.) Studying Fig. 93. the highest normal displacement values were registered at the lower plane of the column. On the perpendicular planes registered displacement-values were very low and even positive, this means they opened during load-process. This tendency is true for the results of normal stresses too. It was expectable that normal stresses will not work on the perpendicular joint-planes, because horizontal forces hadn't been acting on the structure. Comparing results of stresses and displacements it can be considered that normal stresses and displacements are far higher than values connected to shear-effects. The registered shear-effects can be connected to slipping of joints, this phenomenon is acting on the second and fifth joint-planes (sequence is considered from lower to upper planes).

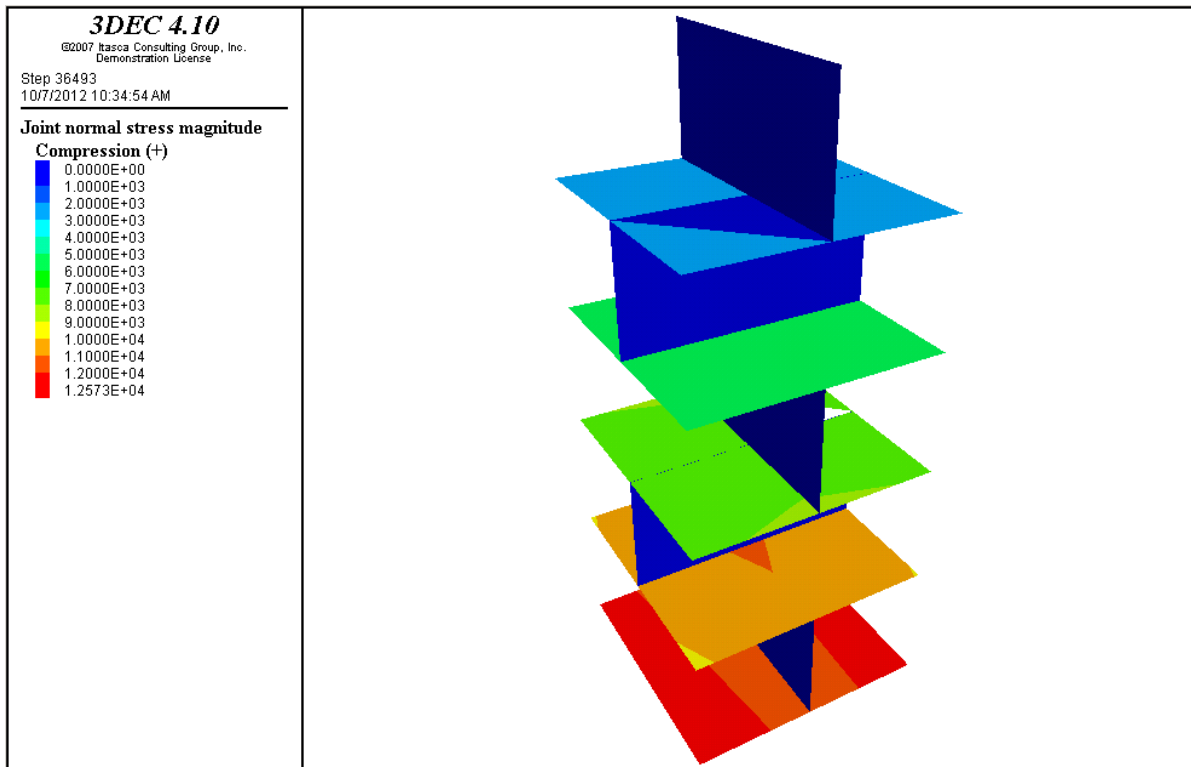


Fig. 94.: Result of „plot jointcontour nstress max au”

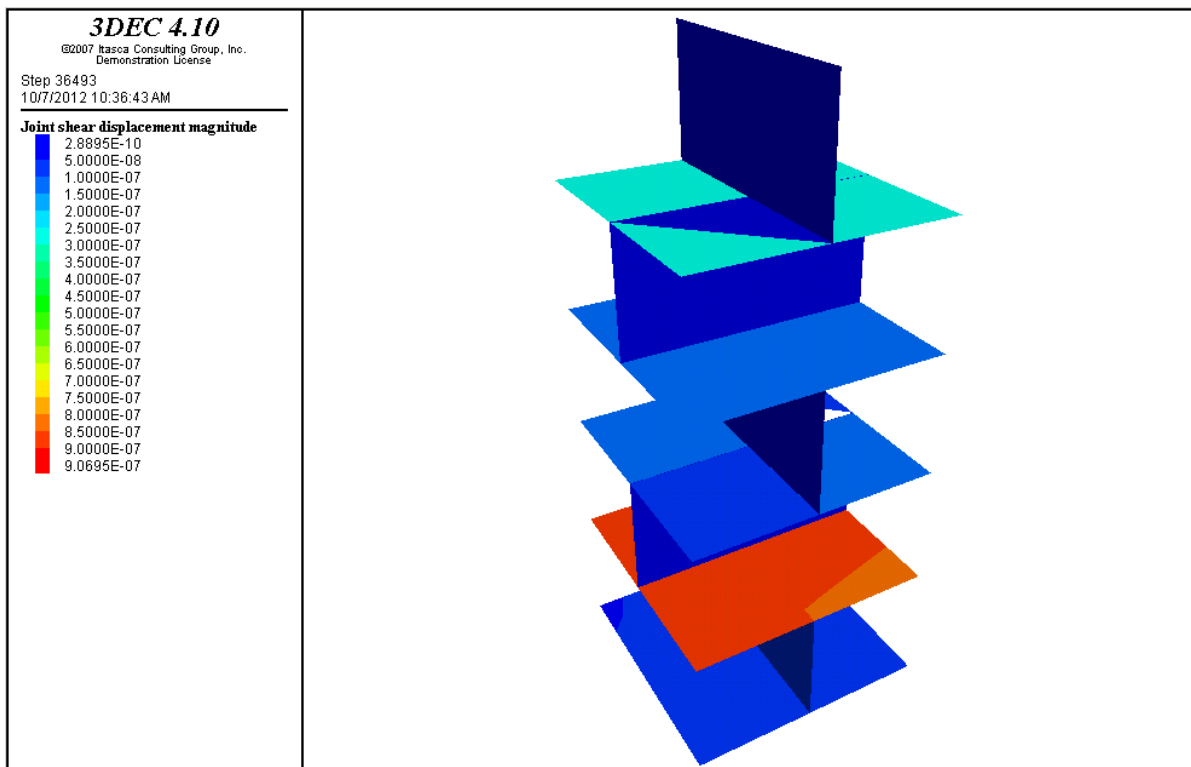


Fig. 95.: Result of „plot jointcontour sdisplacement above au”

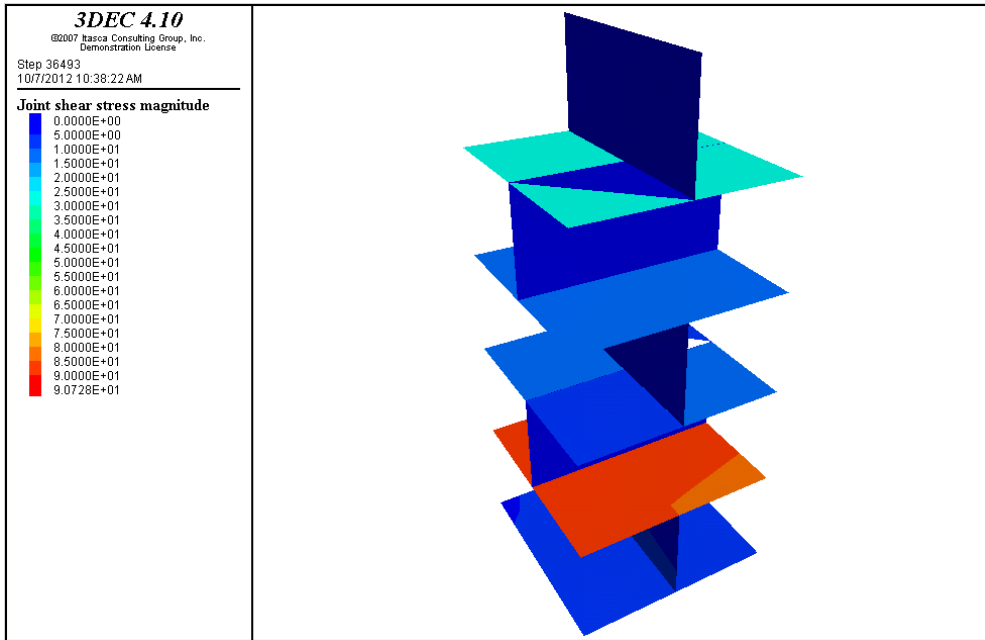


Fig. 96.: Result of „plot jointcontour sstress above au”

„plot jointvector”

Working of the command and its subcommands are the same as in case of the previous one. Shear- and normal-displacements can be represented like in case of „plot jointcontour” command. On the contrary the most important difference is connected to the type of representation; while one of them is changing the colors of joint-planes, the other one is changing the length of vectors according to the values acting on joint-planes.

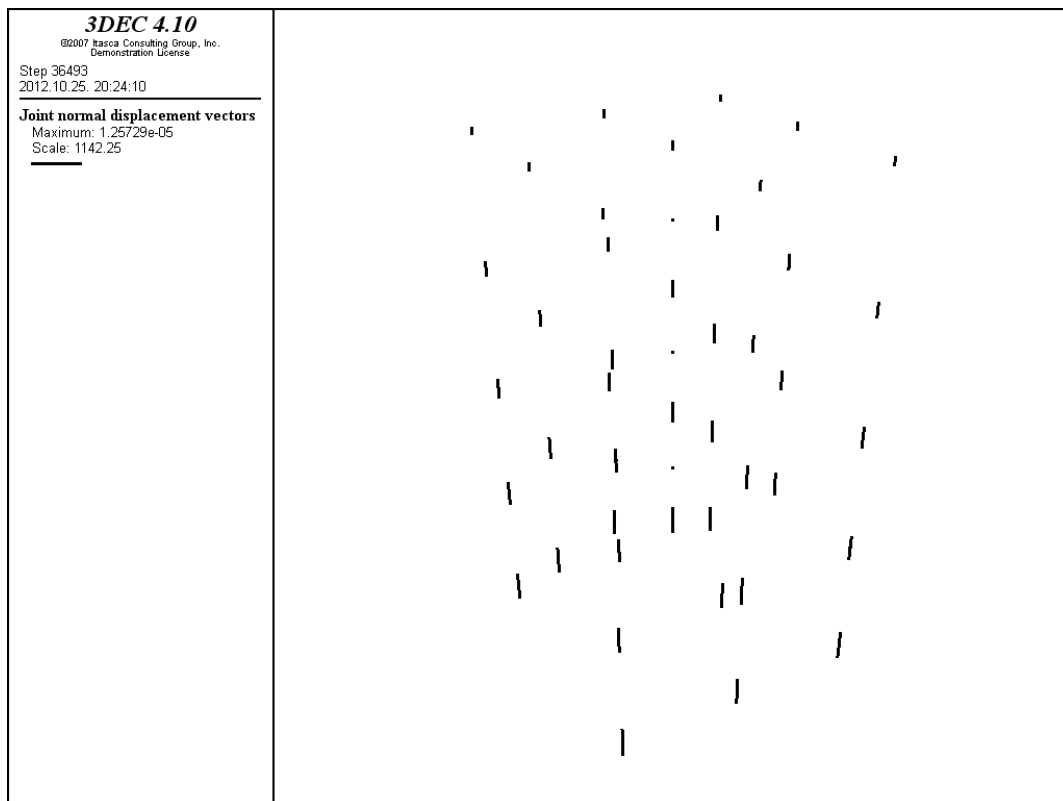


Fig. 97.: Result of „plot jointvector normal above au line width 3 point size 3” command

On Fig. 97. normal-displacement vectors (acting on joint planes) are represented. The results are the same as the ones were got at „plot jointcontour” command. On the other hand these results could be expected, because the magnitude of vectors are growing from the top of the column to the bottom of it.

On Fig. 97. contour of the column can be considered distinct but the results can be more expressive if the „plot add joint wire” or „plot add joint fill off” commands are applied too.

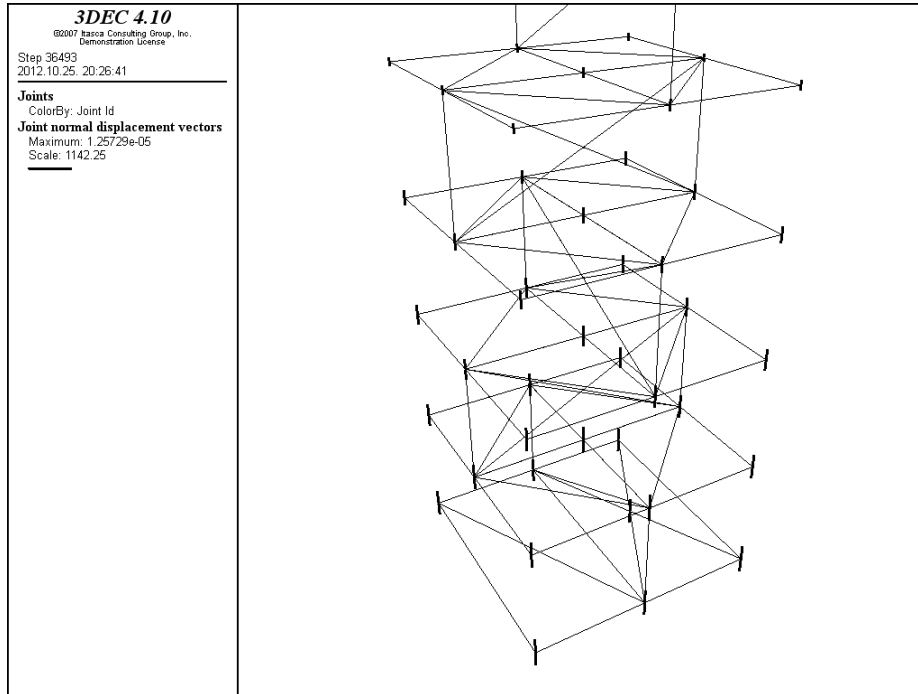


Fig. 98.: Result of „plot joint wire” and „plot add jointvector normal above au line width 3” commands

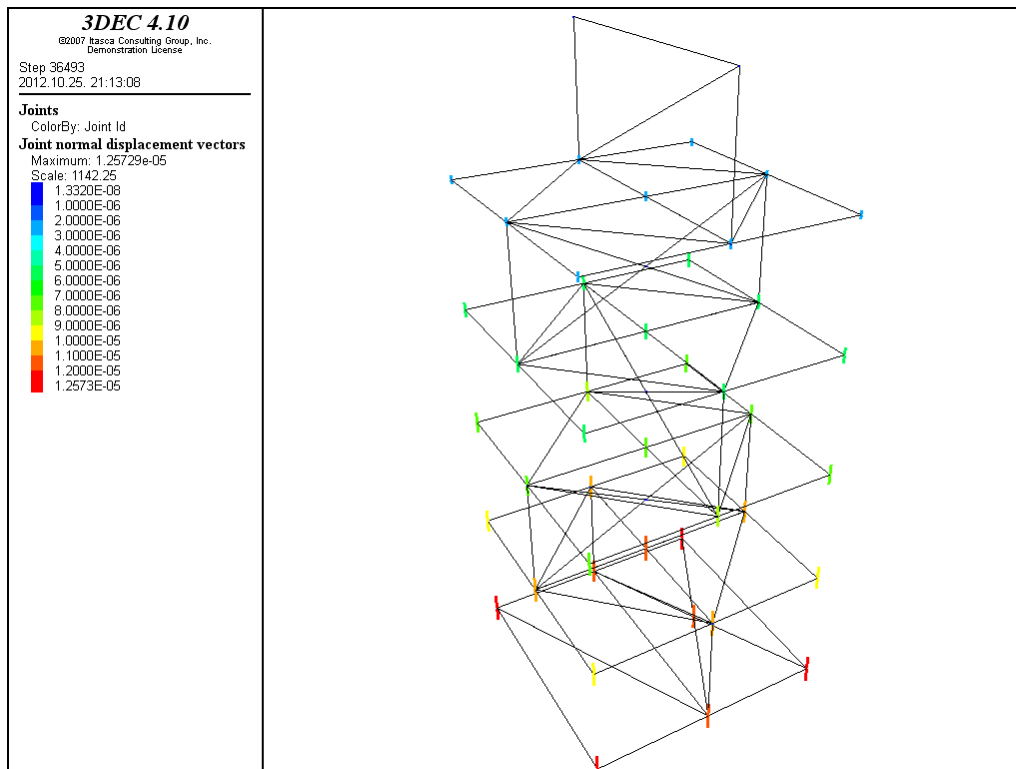


Fig. 99.: Result of „plot joint fill off” and „plot add jointvector normal colorbymagnitude on line width 3” commands

If Fig. 98. and Fig. 99. are compared, it is obvious that Fig. 99. is more advantageous, because the coloured vectors are more outstanding. The previous pictures were all generated with the „plot jointvector normal” command and its subcommands, but the jointvector command is able to display shear-displacement (Fig. 100.).

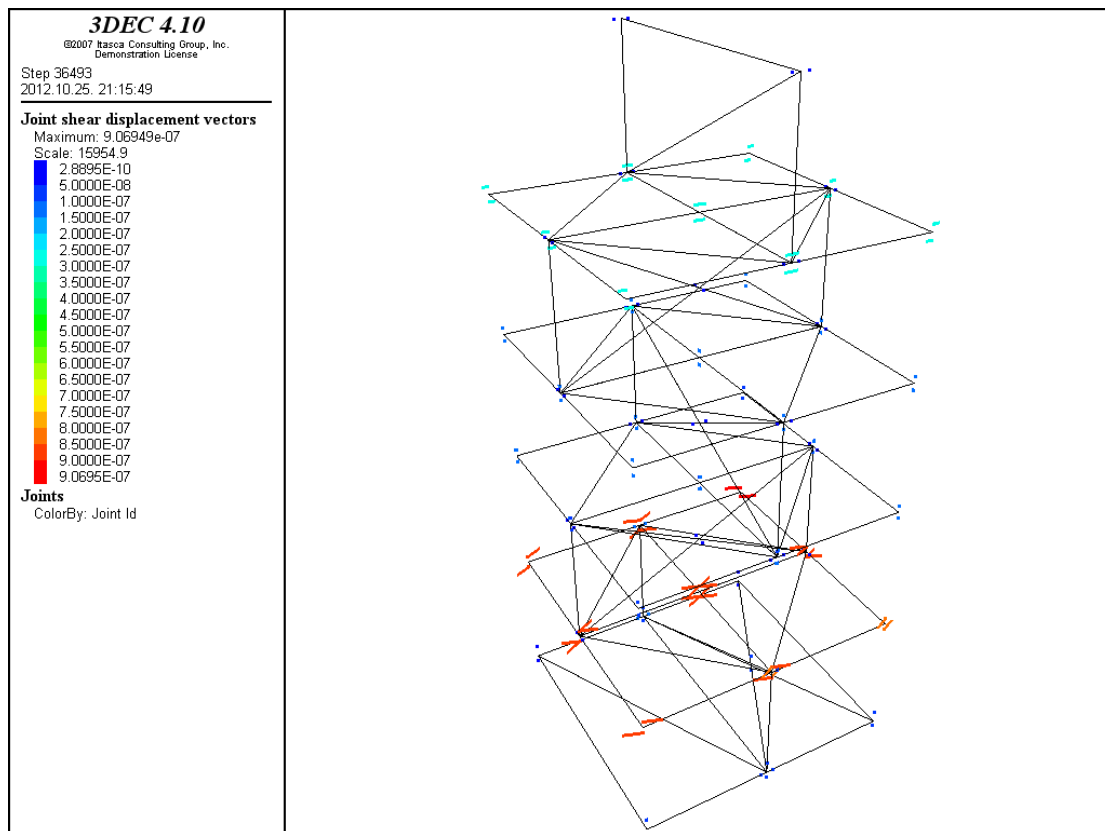


Fig. 100.: Result of „plot jointvector shear colorbymagnitude on line width 3 point size 3” and „plot add joint wire”

„plot jslip”

This command can be used to display the nodes where joint-slip occurs. The gridpoints are represented with three kind of colours according to the state of the nodes (slipping now, slipped past, tensile failure). On Fig. 101. and Fig. 102. results of the command can be seen. Displayed features can be manipulated with several subcommands.

„plot location”

„Plot location” command can be used to plot location of histories currently being taken, if the history has an associated position. For example let us see the script when column is loaded with its self-weight and the following script is added to the „base-script”. For further details and result of this script see section 3.4. As it can be seen the following history is refers to the point identified by $x=-0.3$, $y=-0.3$ and $z=0$. coordinates. According to the geometry of the column this point takes place at the corner of the upper plane of the base-block (Fig. 103.). Results of the command without „plot add block fill off” command can’t be interpreted, because only a coloured point will be displayed on the background and the geometry of the structure is invisible.

```
hist zvel -0.3 -0.3 0
plot hist 1
```

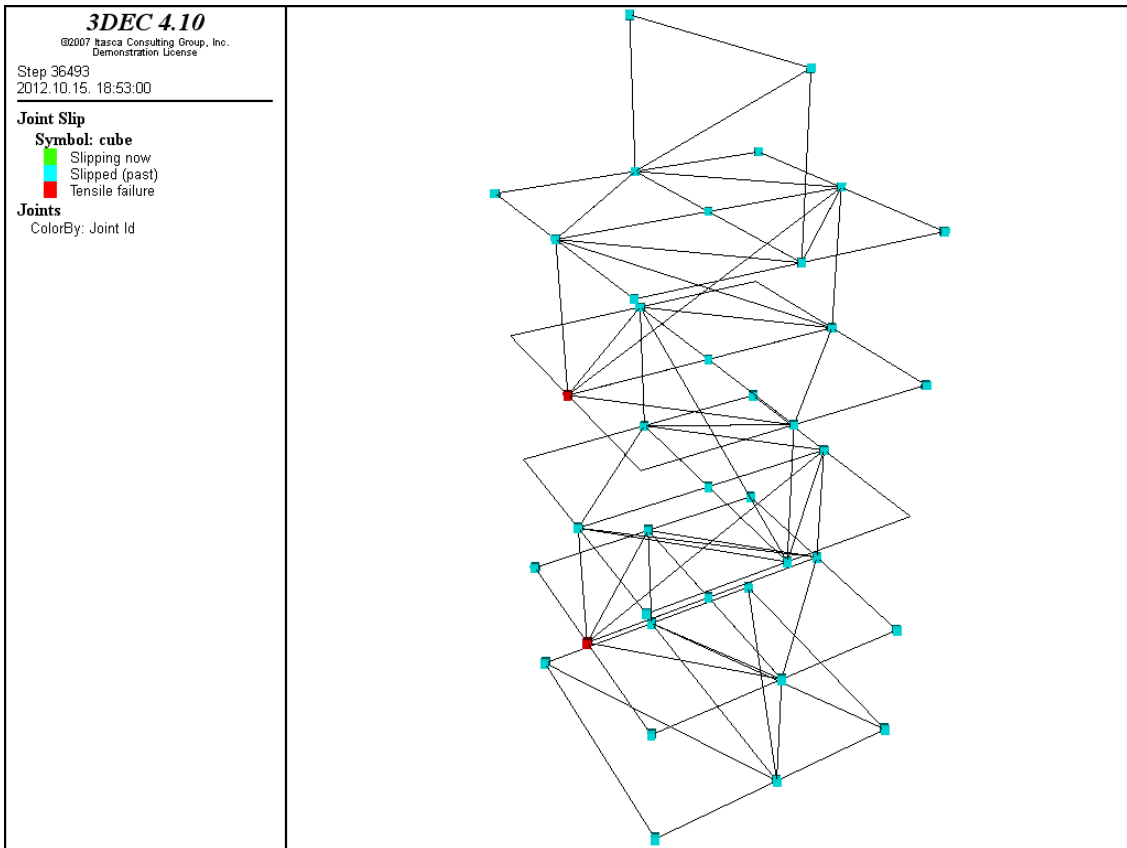


Fig. 104.: Result of „plot jslip” command in case only gravity is applied to the structure

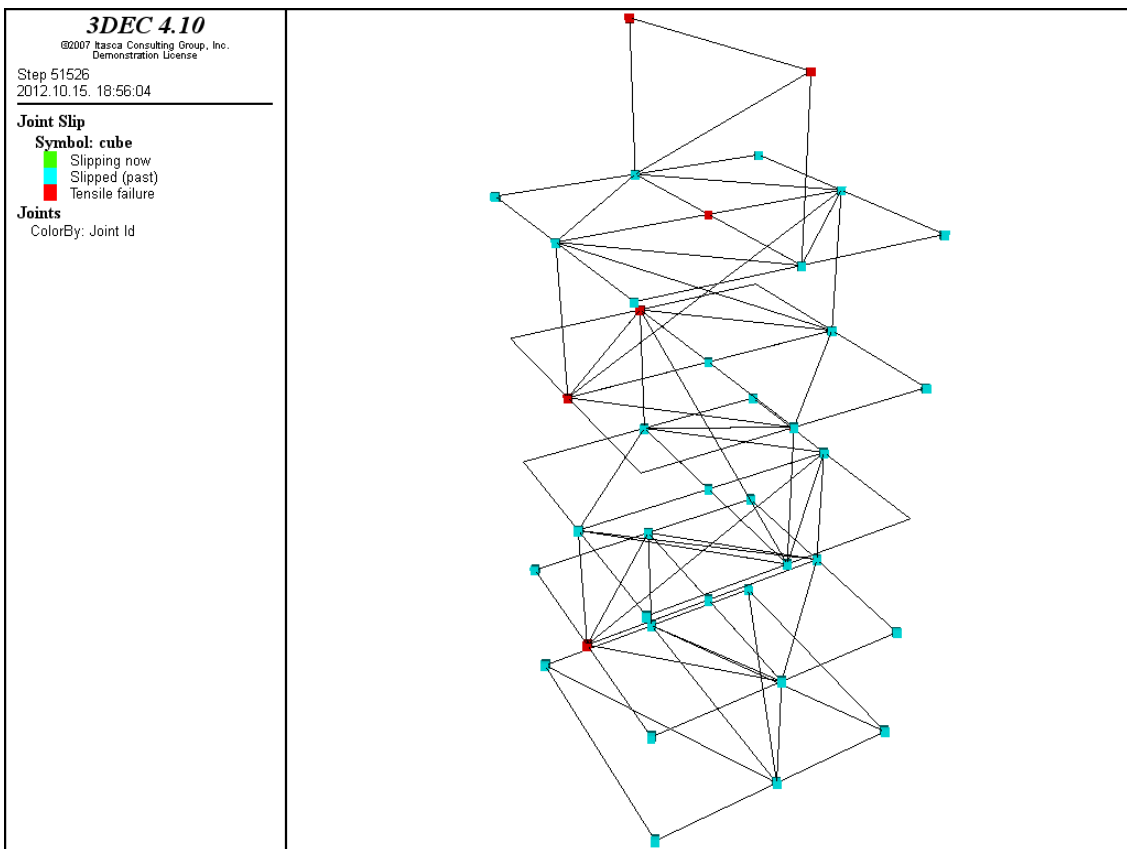


Fig. 105.: Result of „plot jslip” command in case gravity and four concentrated loads are applied for the column

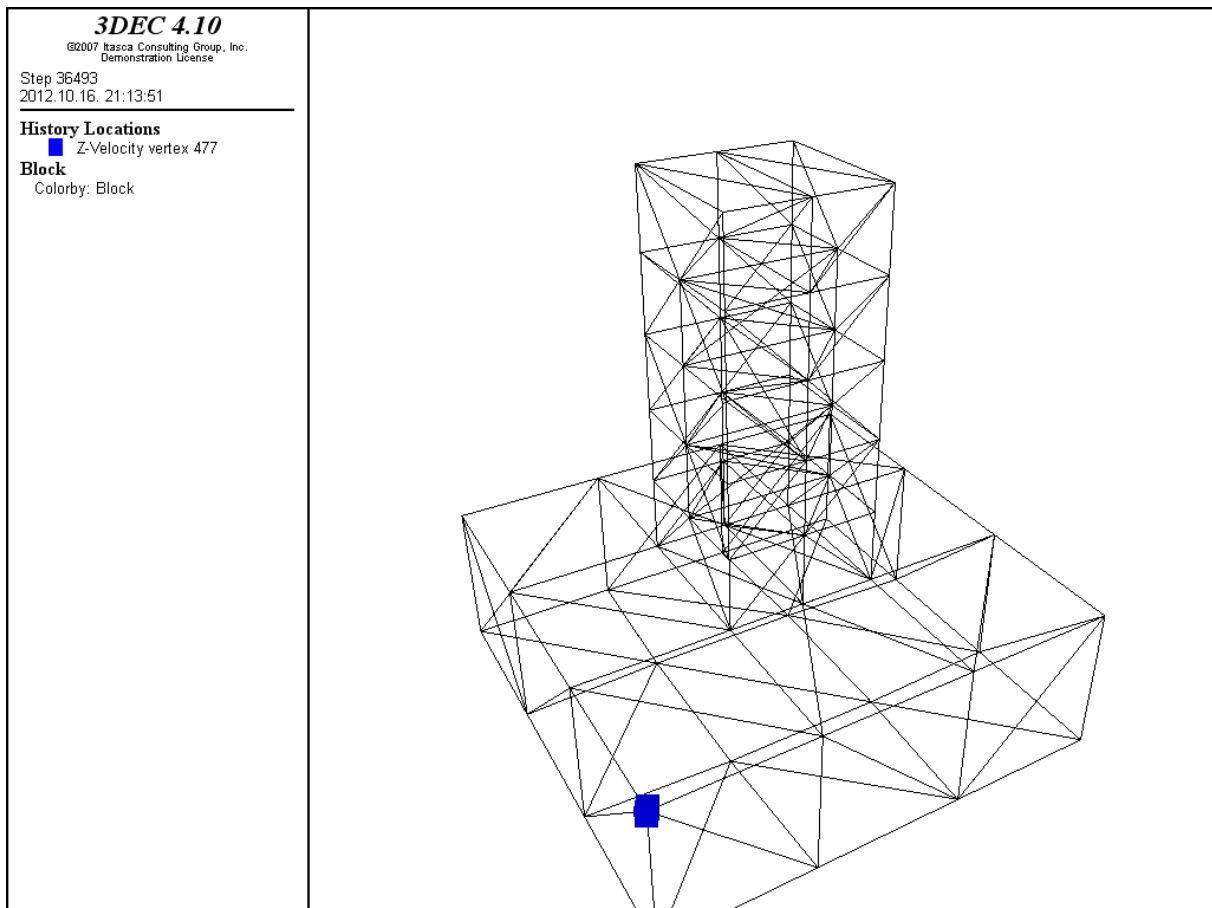


Fig. 103.: Result of „plot location” and „plot add block fill off” commands

„plot stereonet”

This command plots orientation on an equal angle stereonet projection. This kind of representation is very useful in case of geotechnical problems. If the problem axes do not align with the model axes (i.e., positive z -axis (vertical), x -axis (east) and y -axis (north)), then the orientation of geologic features will have to be transformed from the problem axes to the model axes. This can be accomplished by using a stereonet. For more information see Problem Solving with 3DEC page 23-26.

„plot table”

„Tables” are special elements of FISH programming-language. Tables can be used to store high number of datas, their structure is similar to vectors. Every table has an „id number” and two sequential vectors are belong to an „id” („xtable” and „ytable”).

Working of „table” command: „xtable (*id of the table, sequential number of values stored in th table*)=*value*”, this system is the same at „ytable (...)”. The datas stored in the vectors can be identified with their „id”-s (sequential number).

```
def arch
;radius of the arch [m]
  r1=3.0
;
;angle between horizontal line and the line pointing to the lower point of the arch (springing)
  alf1=13.5*(pi/180.)
;
```

```

;number of stones in the arch [pcs.]
i=17.
;
;angle between the lines pointed to the intradose of the arch at every stone
;if the angle between the lines pointed to the springings are divided into „i” parts
alf2=(pi-2.*alf1)/i
;
;loop for calculating the angles and filling the tables
loop ii (1,i+1)
alf=pi-(alf1+alf2*(ii-1))
;
xtable(1,ii)=r1*cos(alf) ;x coordinate at the intradose
ytable(1,ii)=r1*sin(alf) ;z coordinate at the intradose
end_loop
end
@arch

```

```
plot table 1
```

As it can be seen, the script stores the coordinates during the loop that counts the angle. The „id” of the coordinates (xtable, ytable) are named by the actual number of cycle (*ii*). This means that „id”-s of the coordinates get the same number as the sequential number of the stones in arch. The x-coordinates are stored in the xtable, while the z-coordinates are stored in ytable. The result of plot table command gives the contour of the intradose, this example represents the use of the „plot table” and „table” commands. On Fig. 104. the coherent values (x and z coordinates of intradose) are plotted.

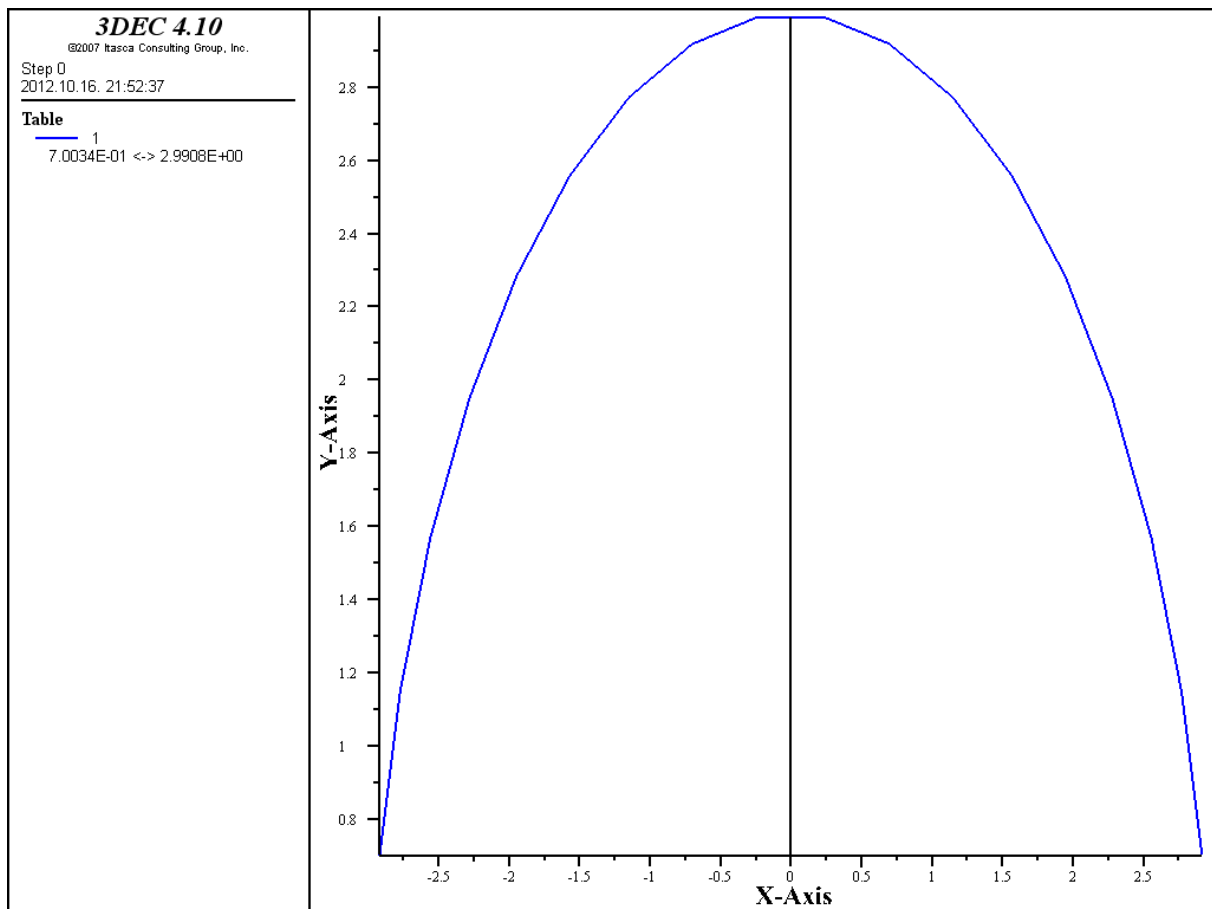


Fig. 104.: Result of „plot table” command, contour of an intradose is represented

„plot traction”

Tractions are result from stress tensors acting on the given boundary planes (see chapter 3.2.). An important aspect of boundary traction is that a more complex boundary stress distribution can be generated than by conventional linear variations (linearly varying boundary stress distributions). On Fig. 105. – 107. results of plot traction command are represented in case that only gravitational loading is applied for the column. On Fig. 108. – 110. besides gravity four concentrated loads were applied at the corners on the top-plane of the structure. Normal and shear tractions on a hypothetical plane, specified by dip, dip direction (dd) and origin coordinates. The area of output may be limited by the radius keyword. This is used to assess slip potential in different orientations. At this command orientation (normals at studied planes) come from either a default or a cut-plane. Block failure criterions are connected to traction components. Available traction results are the following:

- total – Total vector magnitude (this is the default)
- normal – Component of the vector in the normal direction
- shear – Magnitude of the vector resolved on the projection plane

As I mentioned, the following six pictures can be separated into two main categories, according to the applied loads (gravitational, gravitational+concentrated). Besides this within these two categories the order of the results are the same at every load-types (total, normal, shear).

If we are comparing the resulted pictures it can be settled that the representation of tractions give very expressive results and with the help of them potential flow and failure planes can be determined easily. It is obvious that in case of gravitational loads stresses are increasing from the top plane of the structure towards the lower plane of it (Fig. 105. – 107.). This tendence is confirmed by the results of traction commands. On the other hand in case when besides gravity four pointed loads were applied the higher values were concentrated on the top planes of the column (Fig. 108. – 110.). These results could have been expected because the applied concentrated loads are higher than self-weight of the bricks and they are acting in the top plane of the column.

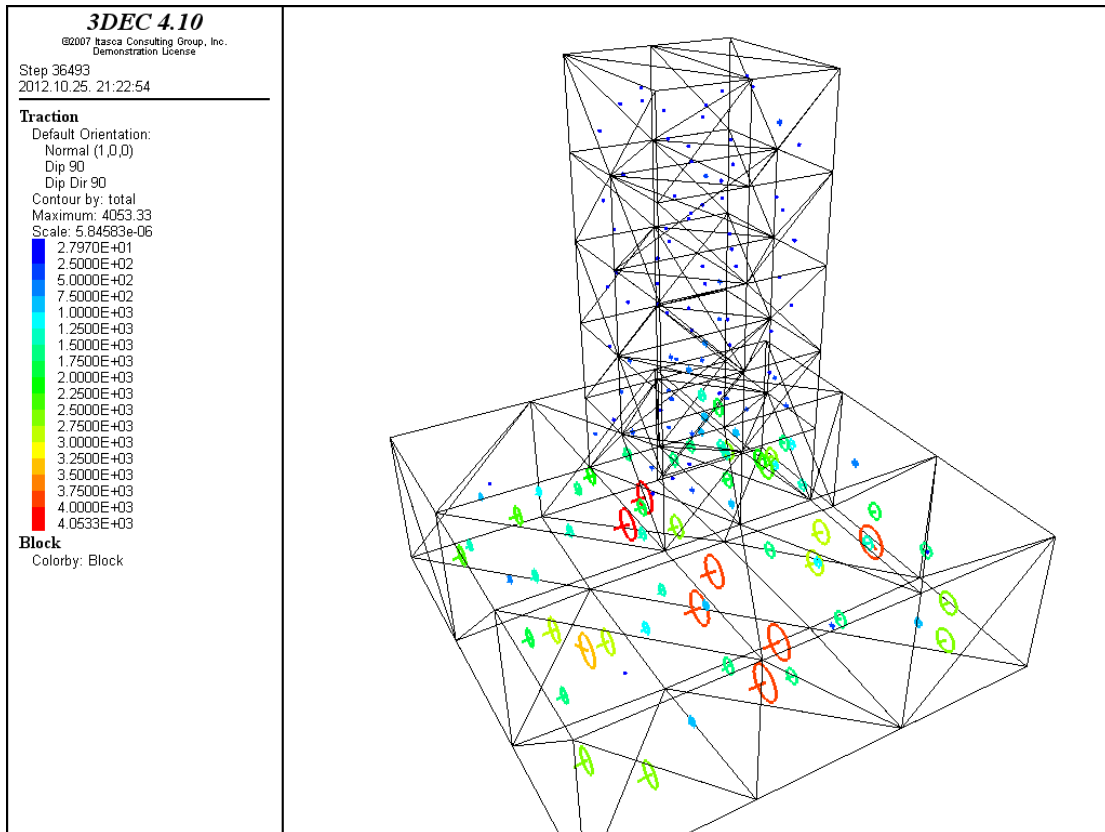


Fig. 105.: Result of „plot traction colorbymagnitude line width 3 point size 3” and „plot add block fill off” commands

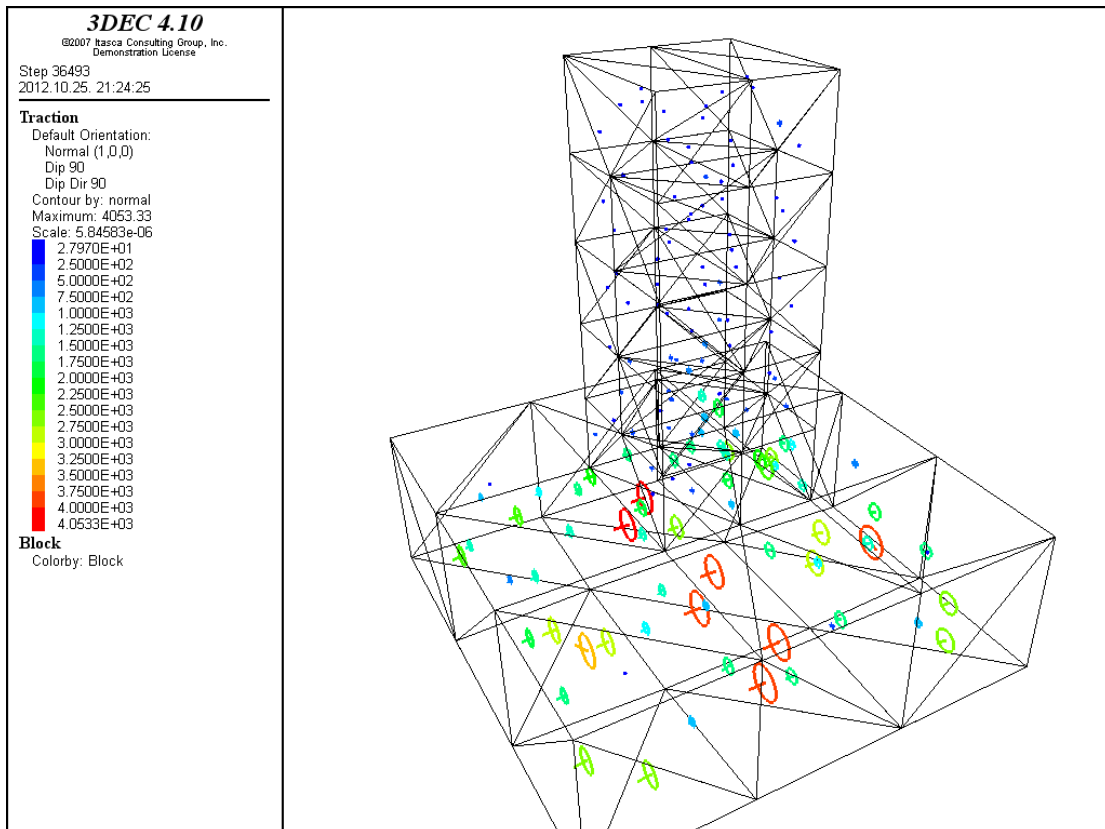


Fig. 106.: Result of „plot traction colorbymagnitude contourby normal line width 3 point size 3” and „plot add block fill off” commands

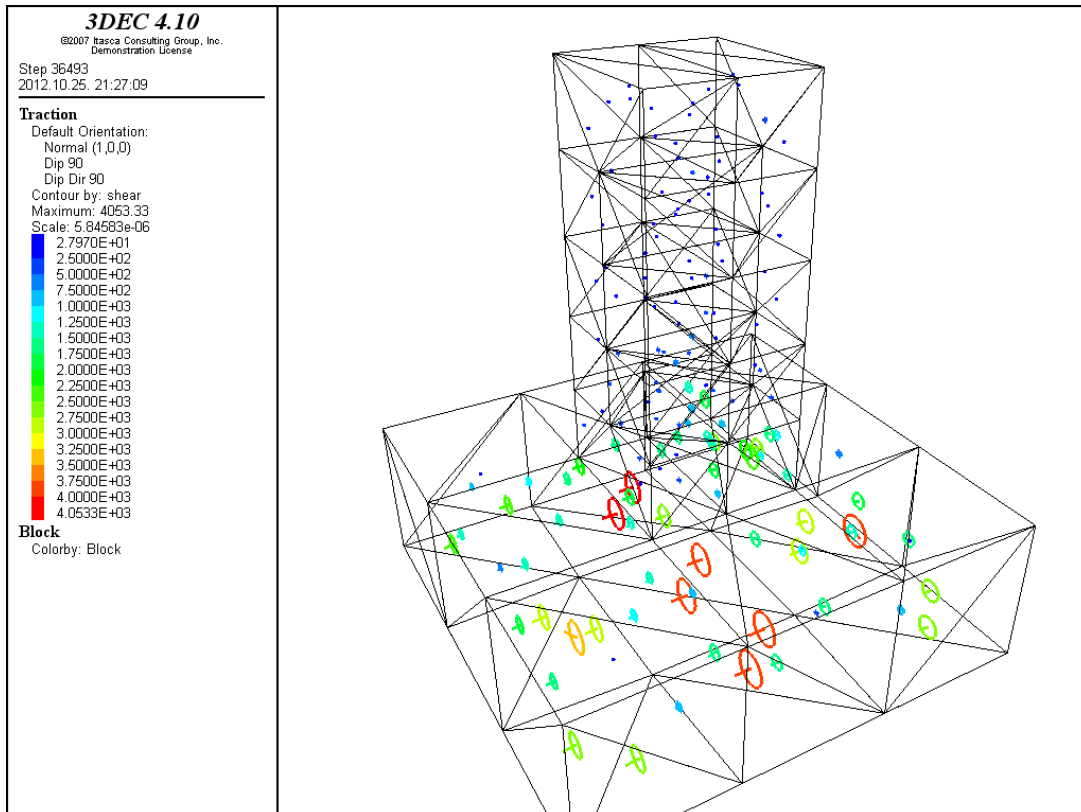


Fig. 107.: Result of „plot traction colorbymagnitude contourby shear line width 3 point size 3” and „plot add block fill off” commands in case gravity is applied

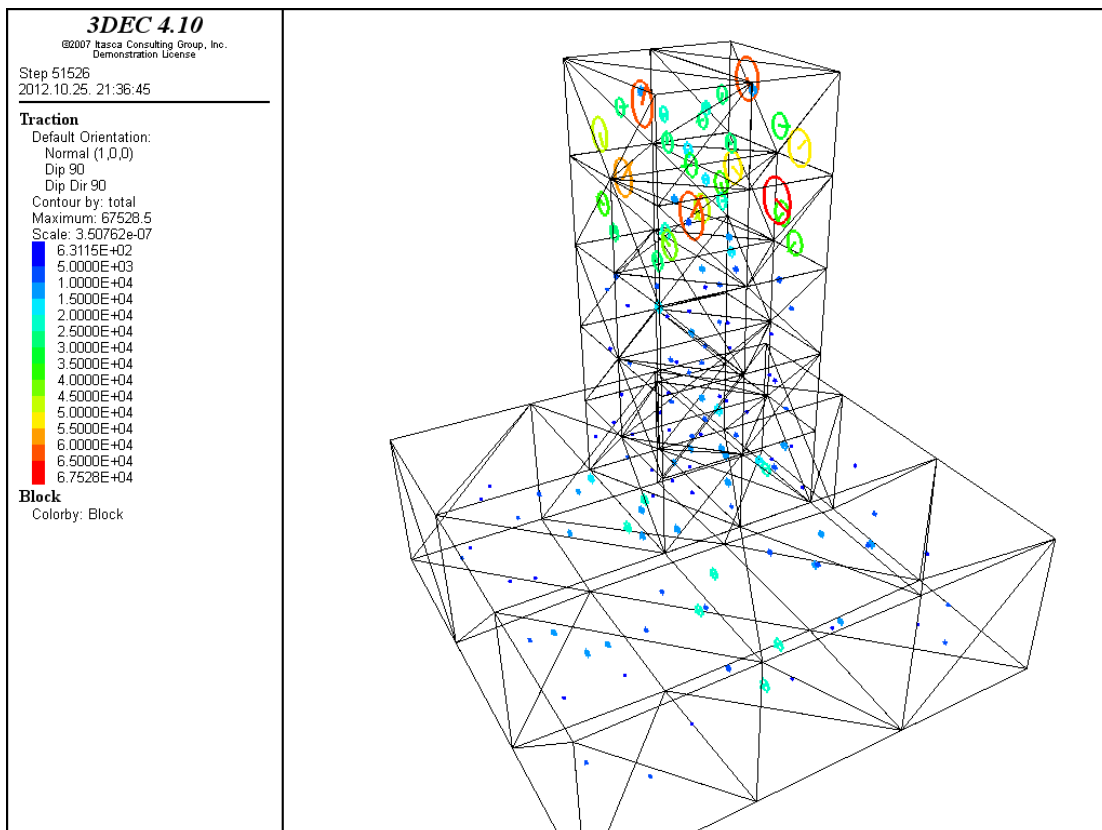


Fig. 108.: Result of „plot traction colorbymagnitude contourby total line width 3 point size 3” and „plot add block fill off” commands in case 1000N concentrated load is applied on top of the structure

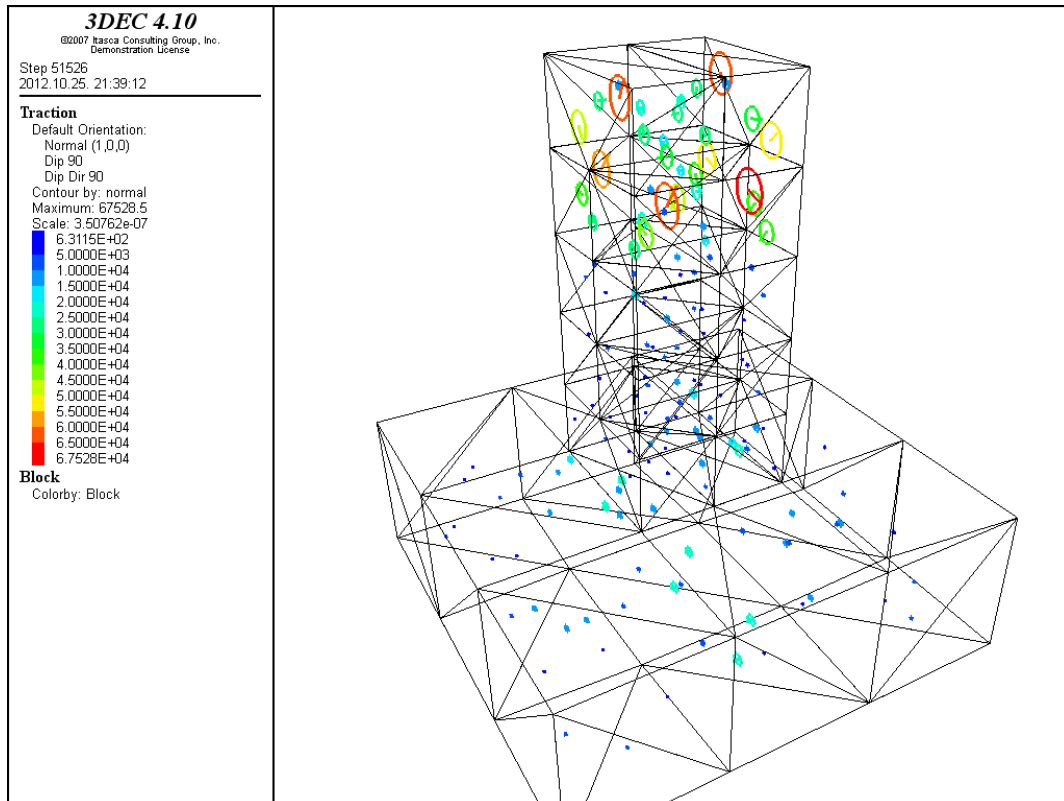


Fig. 109.: Result of „plot traction colorbymagnitude contourby normal line width 3 point size 3” and „plot add block fill off” commands in case 1000N concentrated load is applied on top of the structure

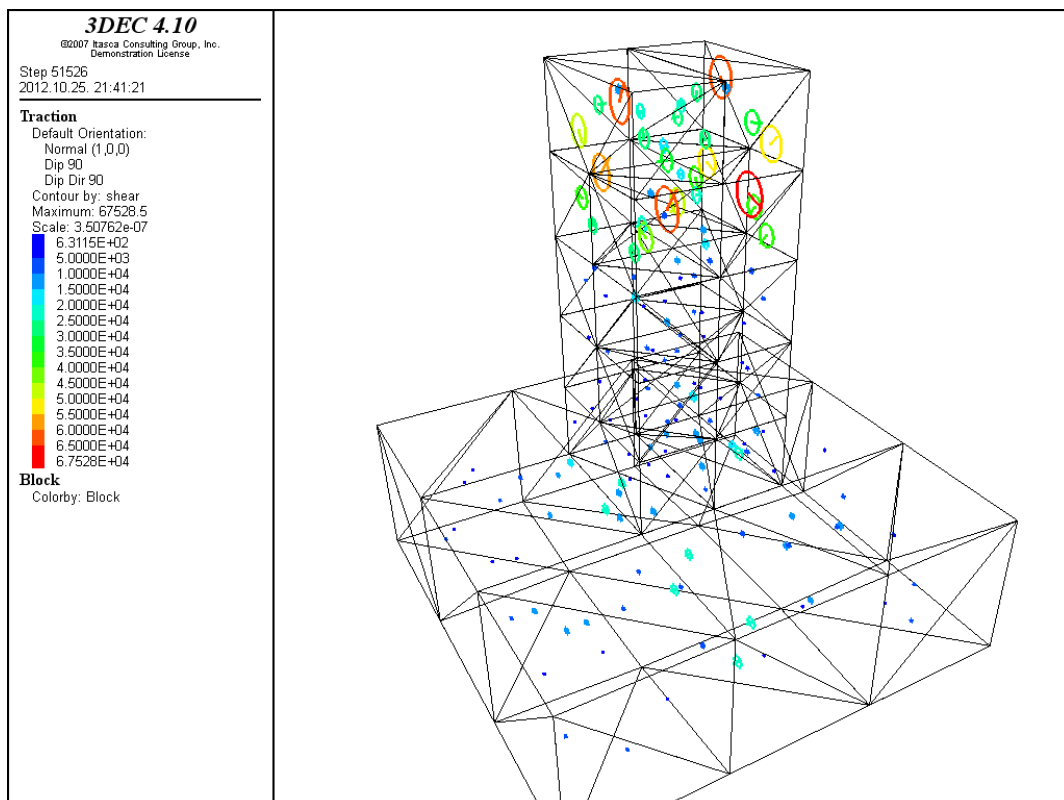


Fig. 110.: Result of „plot traction colorbymagnitude contourby shear line width 3 point size 3” and „plot block fill off” commands in case 1000N concentrated load is applied on top of the structure

„plot udscalar”, „plot udtensor”, „plot udvector”

If a scalar, tensor or vector is created with the help of FISH, results of it can be plotted in the same way as in case of built-in 3DEC commands. „Plot user-defined” commands can be used to represent these quantities.

„plot velocity”

This command is used to plot block vertex velocity vectors. Subcommands and the most effective commands and command-pairs are the same as at the previous ones. On Fig. 111. and Fig. 112. only vectors were represented, but on Fig. 113. contours of the blocks were plotted too. In my opinion in this case the first two pictures give more effective results than the third one.

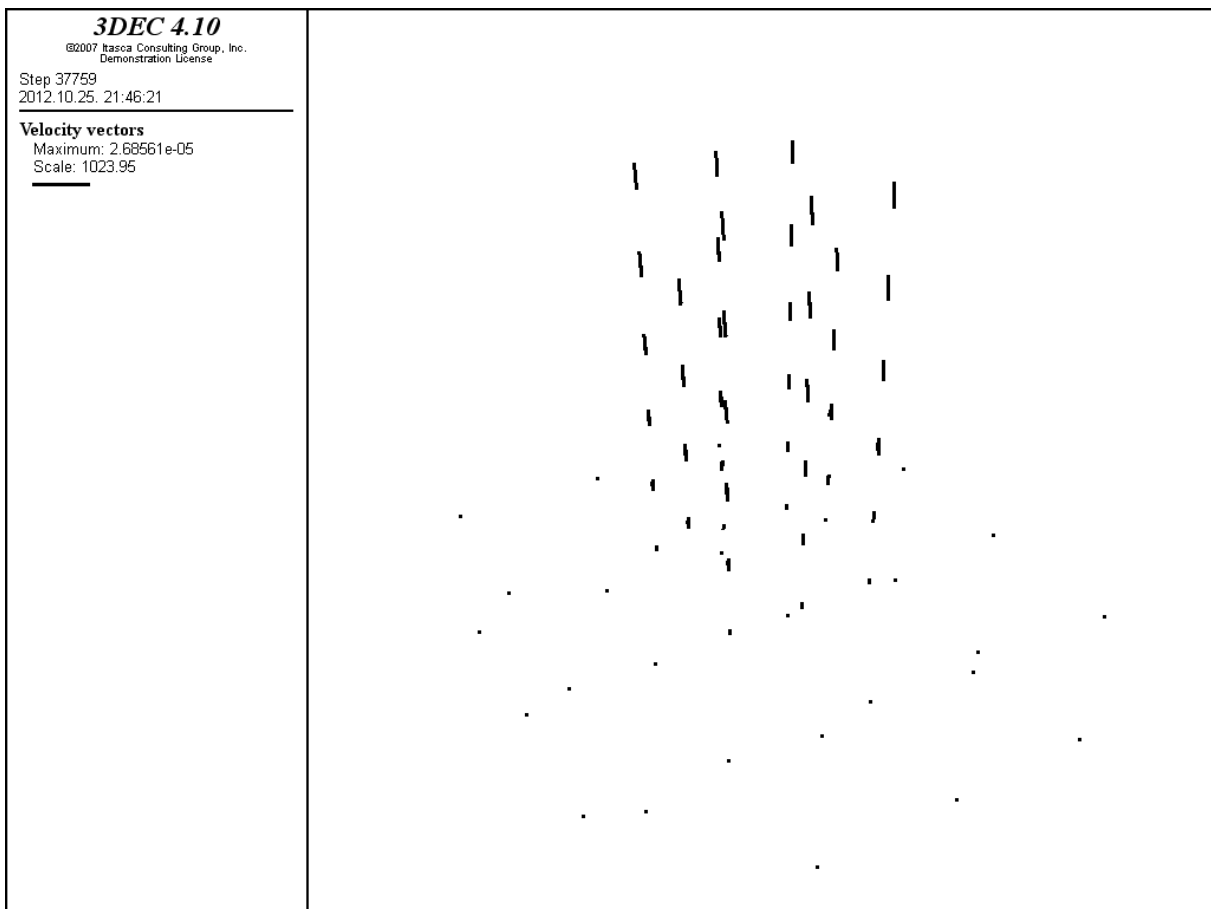


Fig. 111.: Result of „plot velocity above au line width 3 point size 3” command

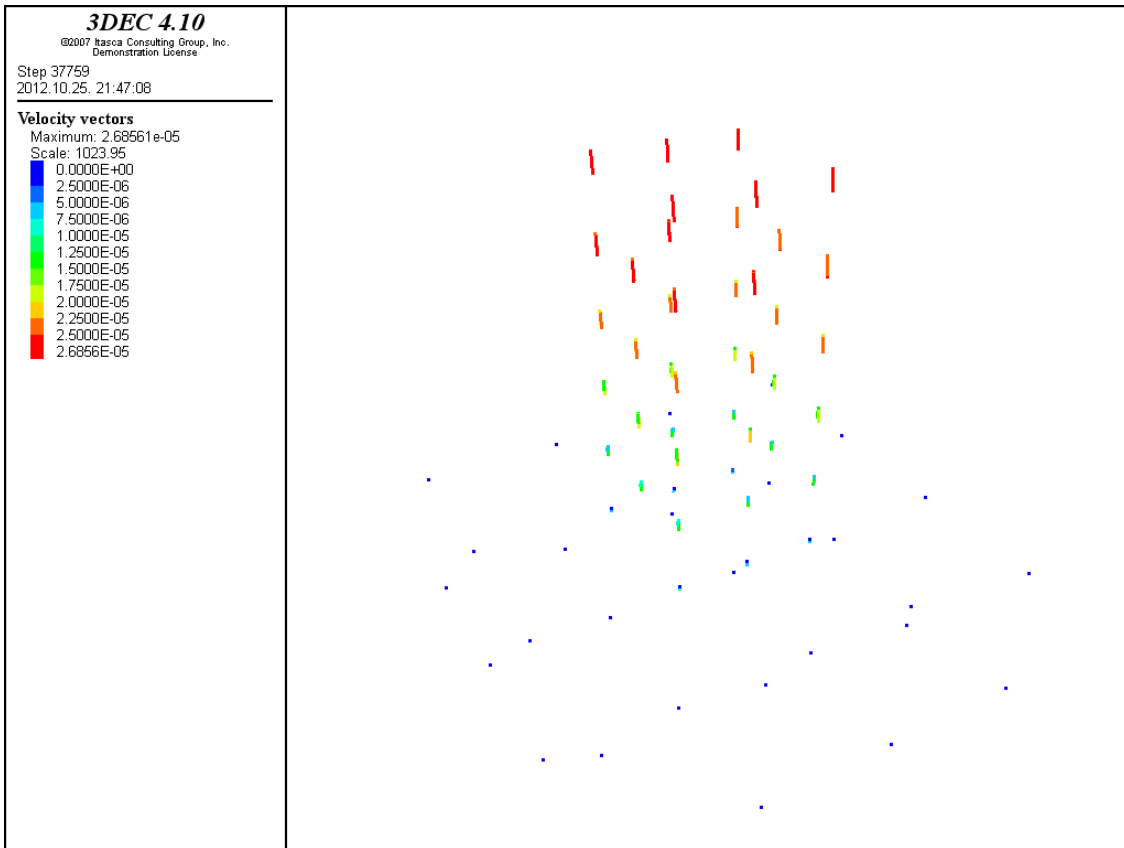


Fig. 112. : Result of „plot velocity colorbymag” command

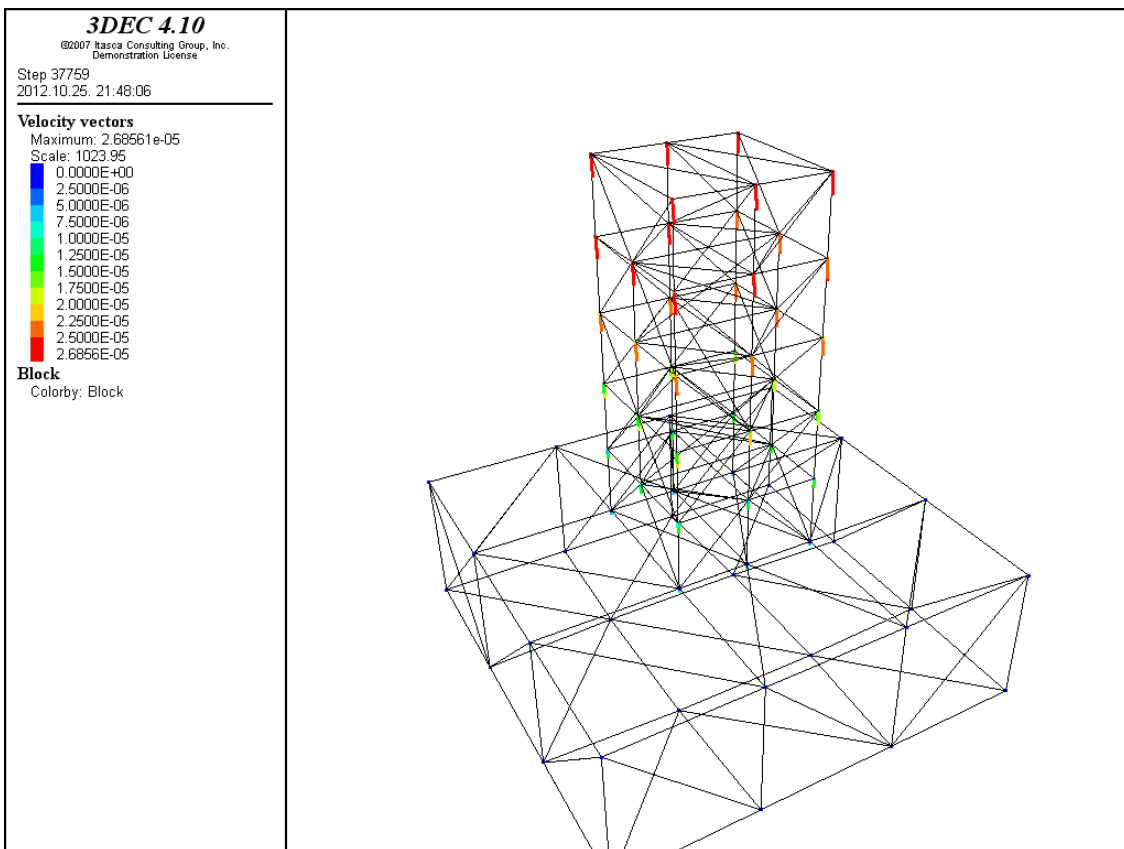


Fig. 113.: Result of „plot velocity colorbymag line width 3 point size 3” and „plot add block fill off” commands

„plot water”

This command is not reviewed in this paper because it is connected to fluid models. If it is applied, the current water table can be plotted. The water table surface is either built up with individual triangles or an infinite plane.

„plot zonetensor”

In case of this command the following quantities can be represented:

principal (stress) – zone strain tensor
strainincrement (ssi) – zone strain increment tensor
strainrate (ssr) – zone strain rate tensor

The results can be easily manipulated with the following subcommands:

colorby deviatoric – Tensors are drawn in a uniform color, determined by a contour ramp of S1 - S3

colorby intermediate – Tensors are drawn in a uniform color, determined by a contour ramp of intermediate principal stress

colorby maximum – Tensors are drawn in a uniform color, determined by a contour ramp of maximum principal stress

colorby minimum – Tensors are drawn in a uniform color, determined by a contour ramp of minimum principal stress

colorby order – Three different colors are used for each axis, representing minimum, intermediate and maximum principal stress

colorby uniform – A single color is used for all tensors

In case only gravity is acting on the structure effect of color-ramp manipulating subcommands can be seen on Fig. 114. – 121. Represented quantity is principal stress at every picture.

On Fig. 122. and 123. the strainincrement and strainrate were plotted the contour ramp was chosen by deviatoric stresses.

On Fig. 124. – 126. results of principal, strainincrement and strainrate commands are represented in case besides self-weight of the column concentrated loads were defined at the top plane of it (4x1000N). As it can be seen zonetensors give very expressive results and the previously mentioned phenomena (behaviour of the structure under different load cases) can be interpreted with them. For instance on Fig. 125. highest values of zone strain increment values were registered at the top plane of the column where concentrated loads were acting.

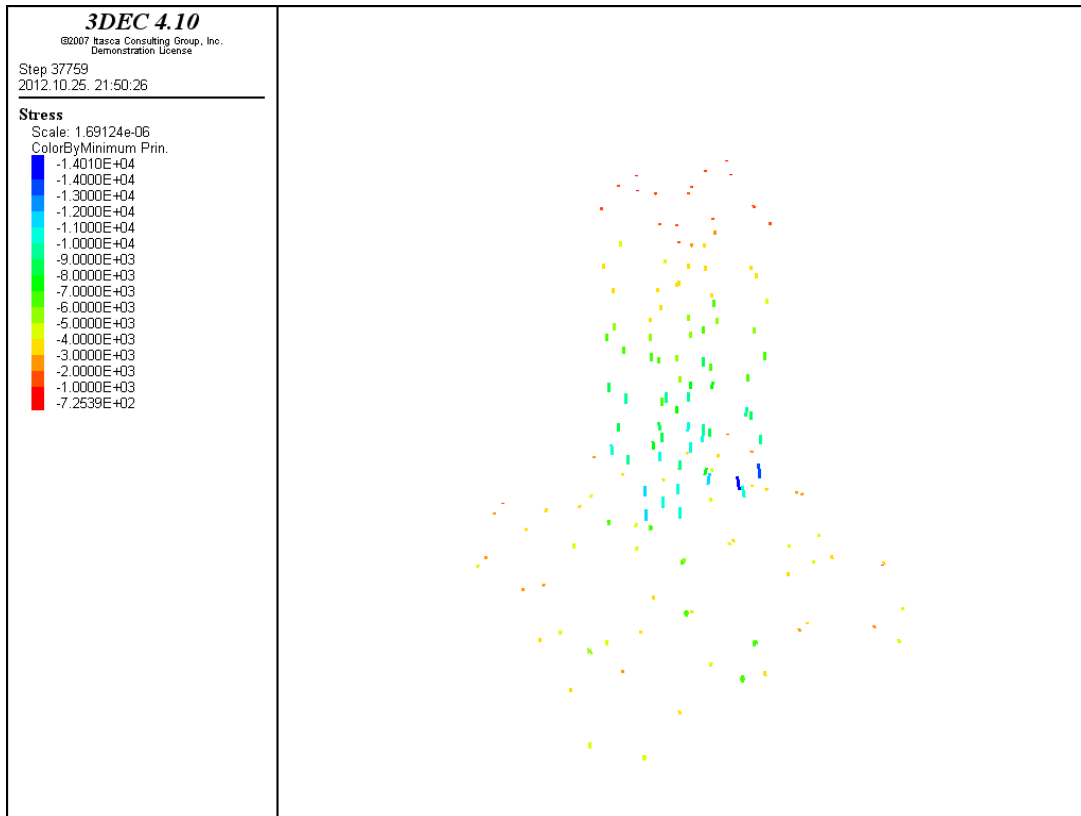


Fig. 114.: Result of „plot zonetensor stress line width 4”, by default the software represented the tensors by the color ramp of minimum principal stress in case only gravity is acting on the structure

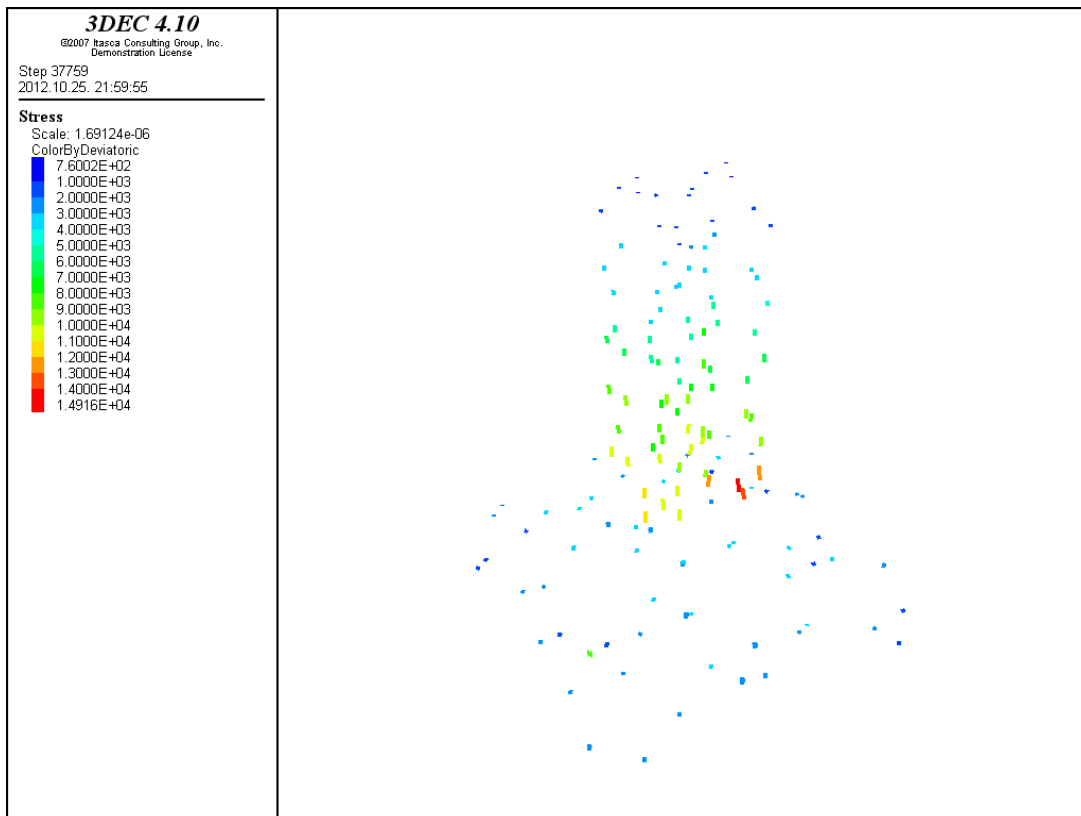


Fig. 115.: Result of „plot zonetensor stress colorby deviatoric line width 4” command in case only gravity is acting on the structure

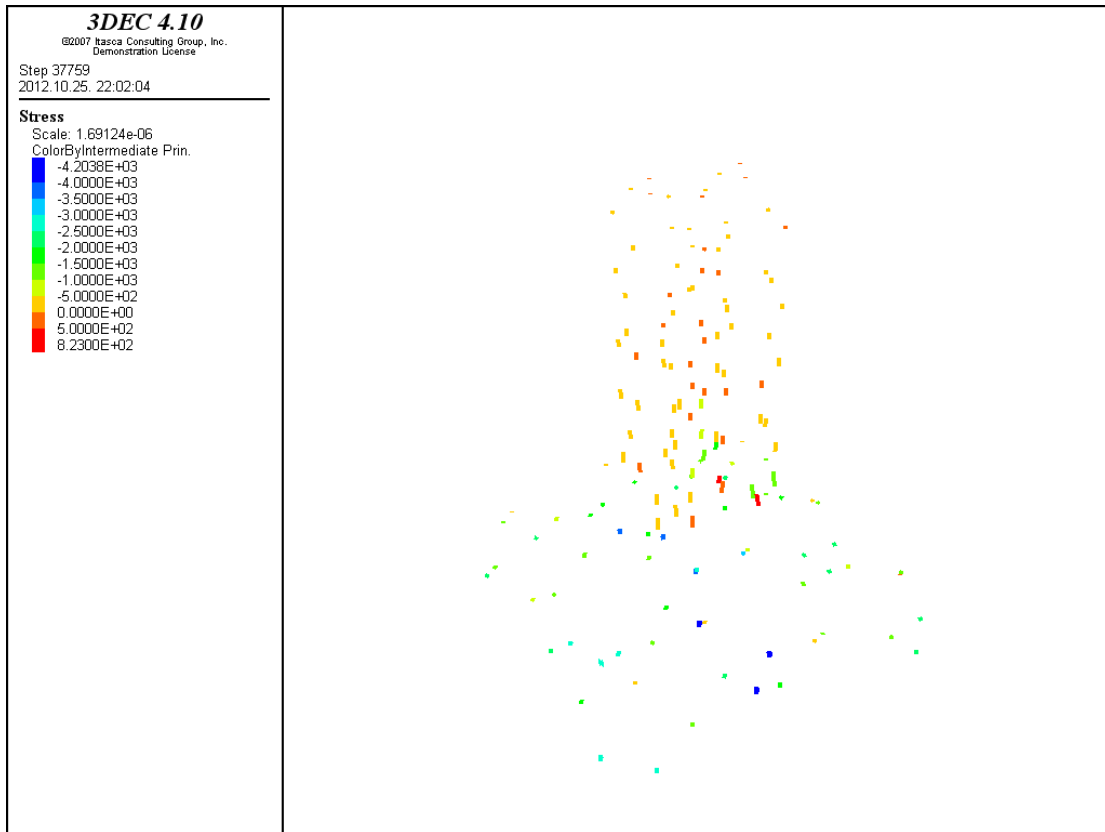


Fig. 116.: Result of „plot zonetensor stress colorby intermediate line width 4” command in case only gravity is acting on the structure

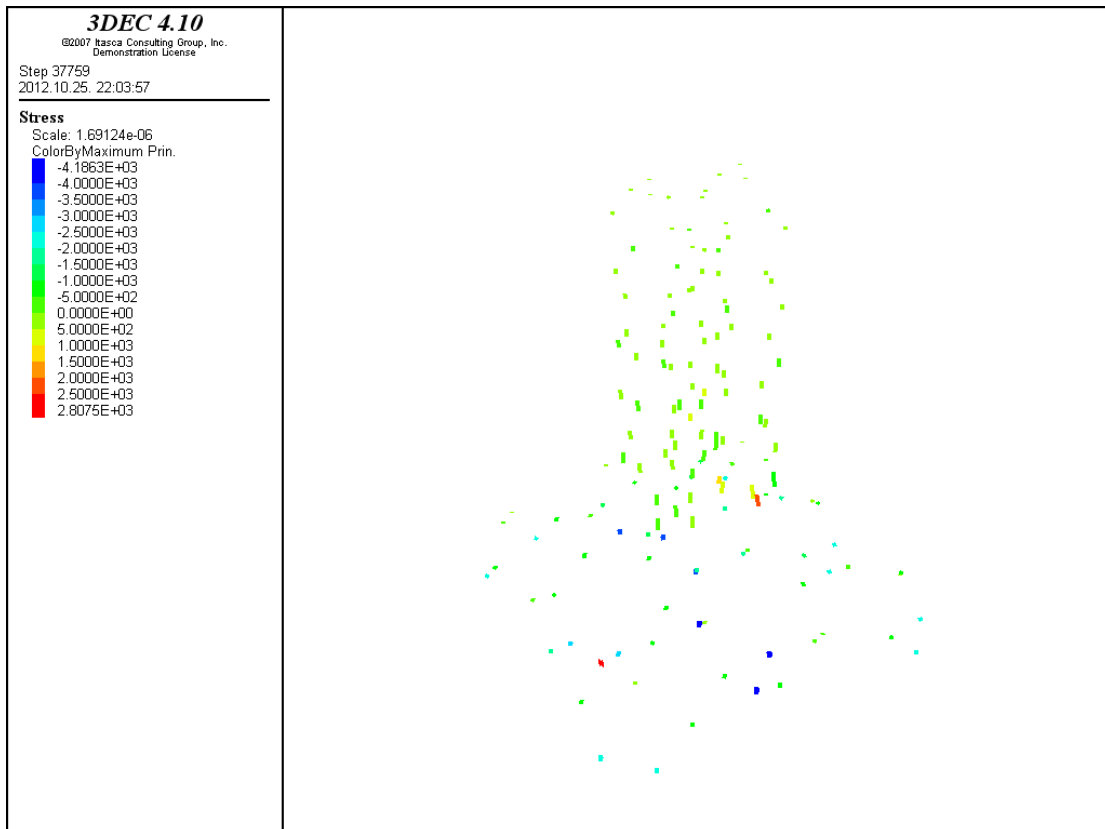


Fig. 117.: Result of „plot zonetensor stress colorby maximum line width 4” command in case only gravity is acting on the structure

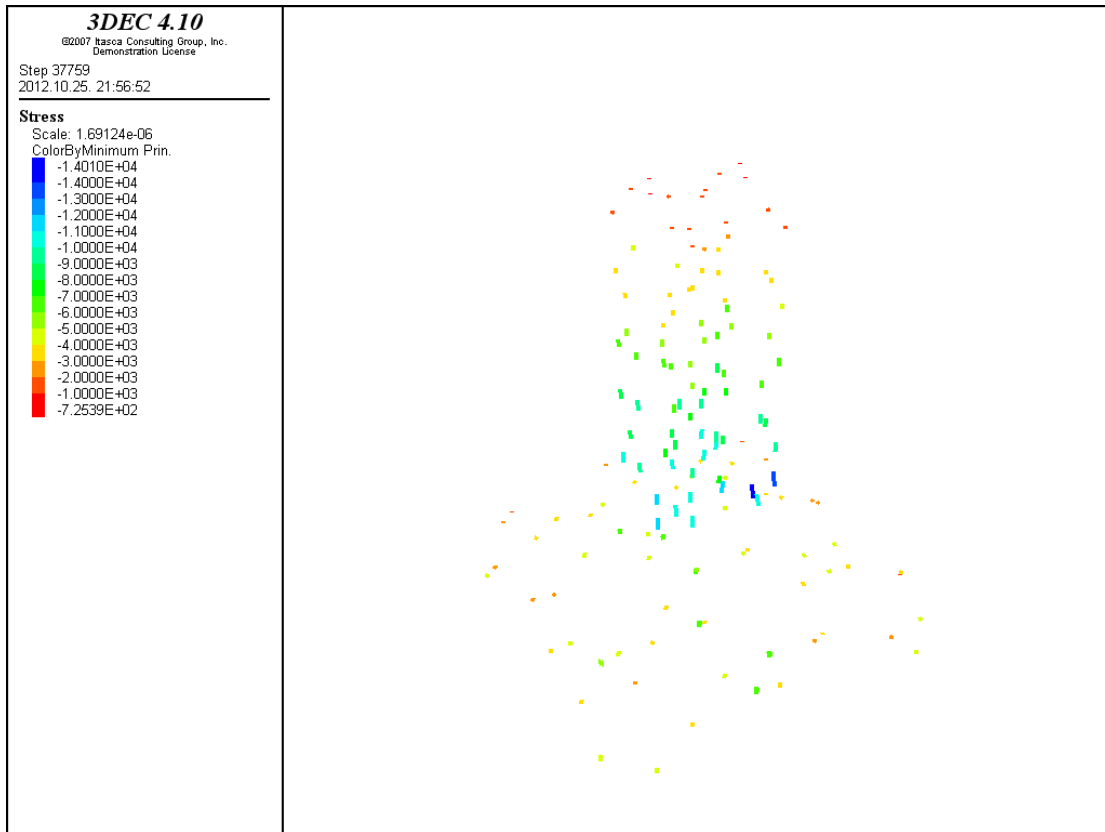


Fig. 118.: Result of „plot zonetensor stress colorby minimum line width 4” command in case only gravity is acting on the structure

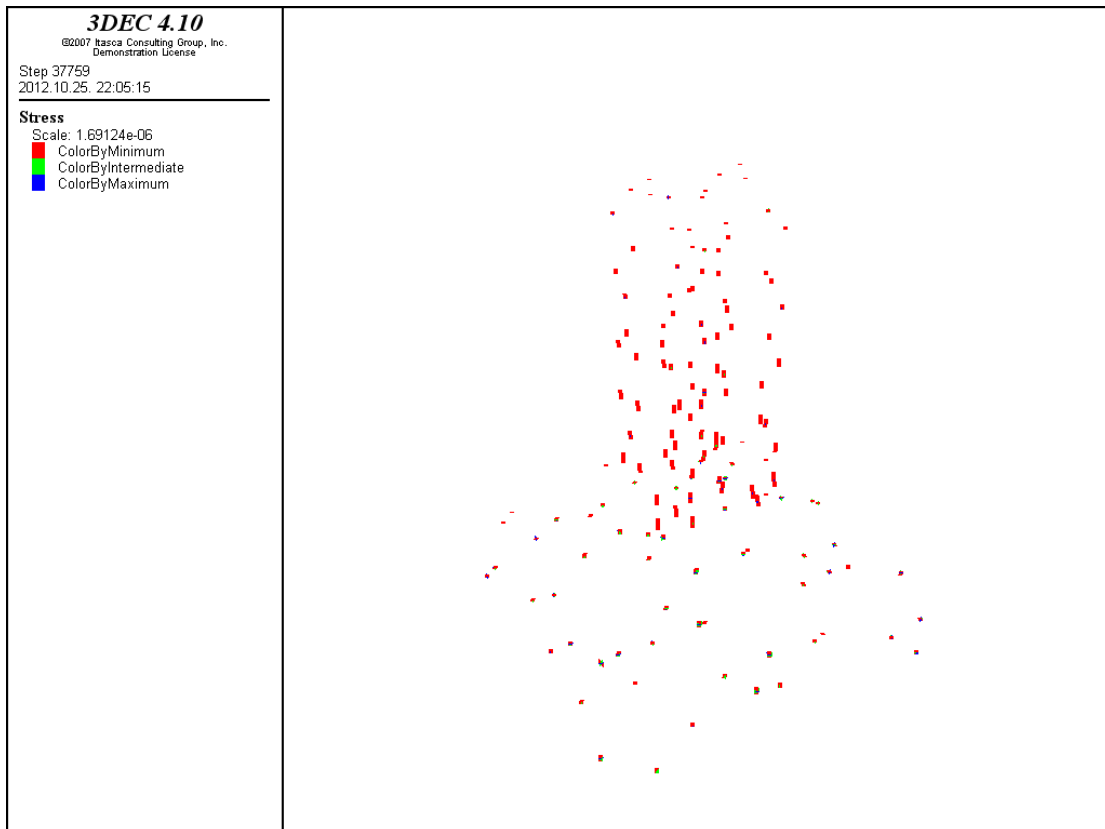


Fig. 119.: Result of „plot zonetensor stress colorby order line width 4” command in case only gravity is acting on the structure

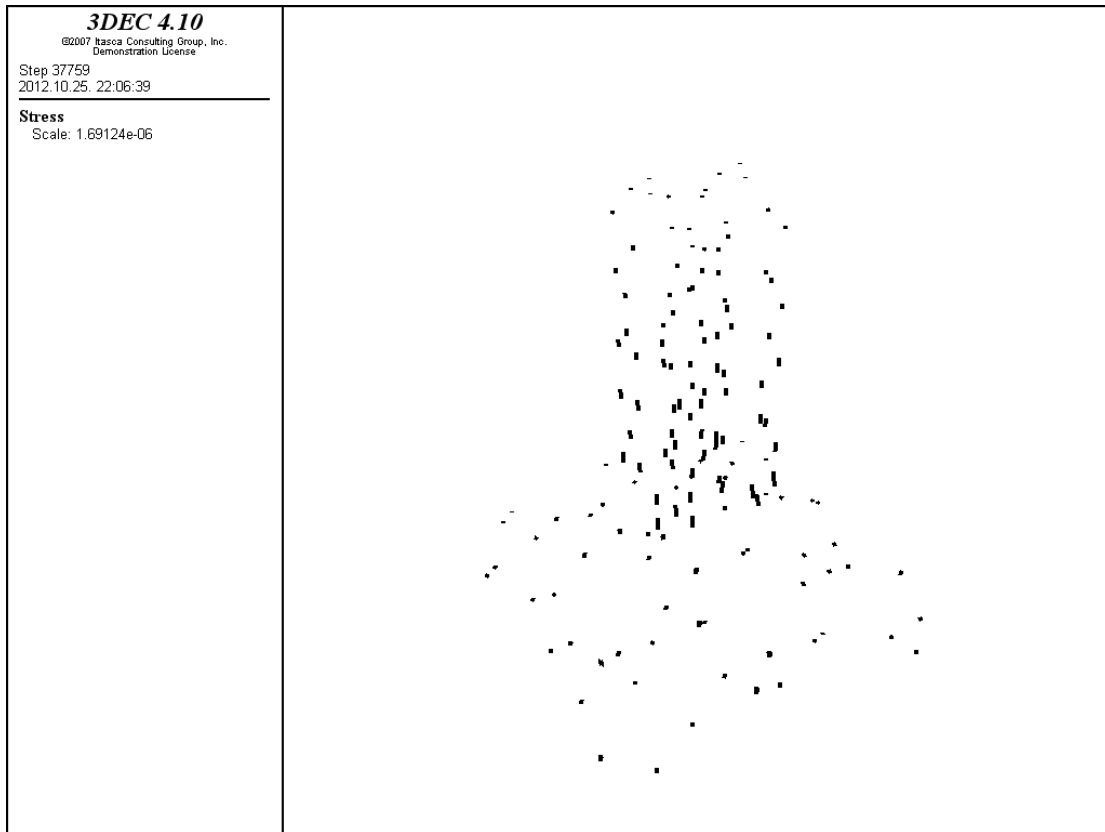


Fig. 121.: Result of „plot zonetensor stress colorby uniform line width 4” command in case only gravity is acting on the structure

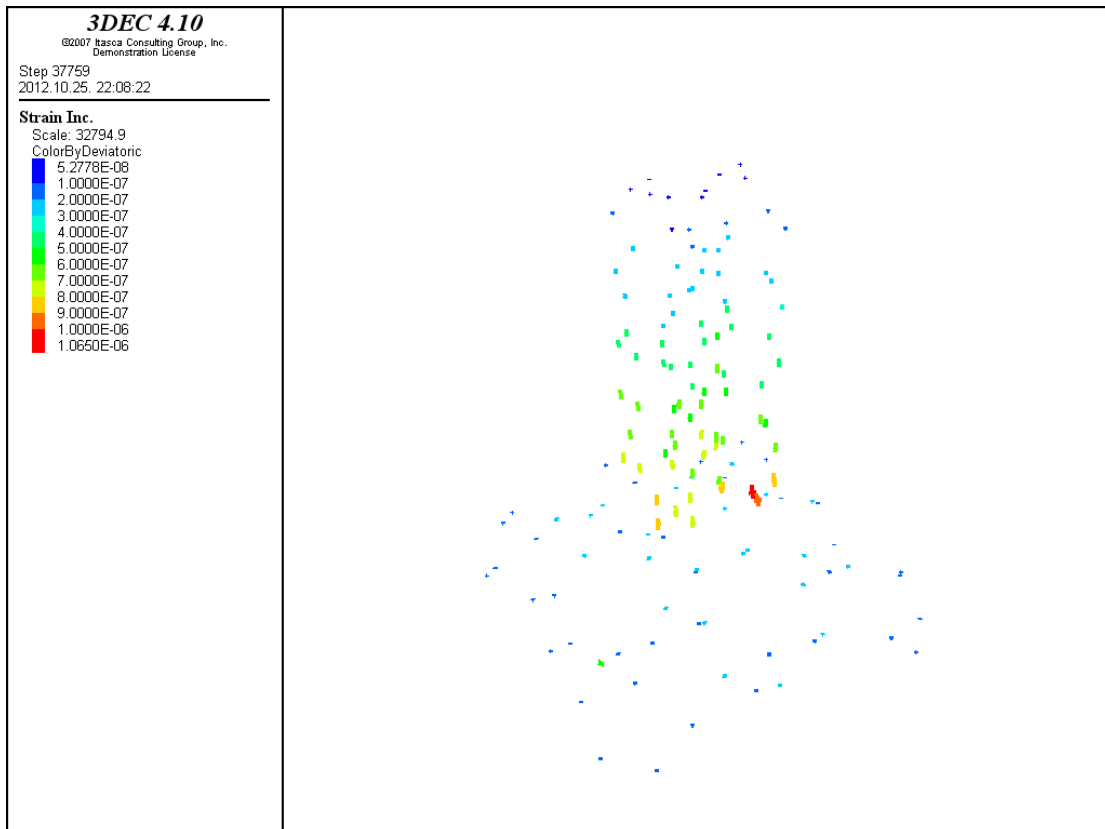


Fig. 122.: Result of „plot zonetensor strainincrement colorby deviatoric line width 4” command in case only gravity is acting on the structure

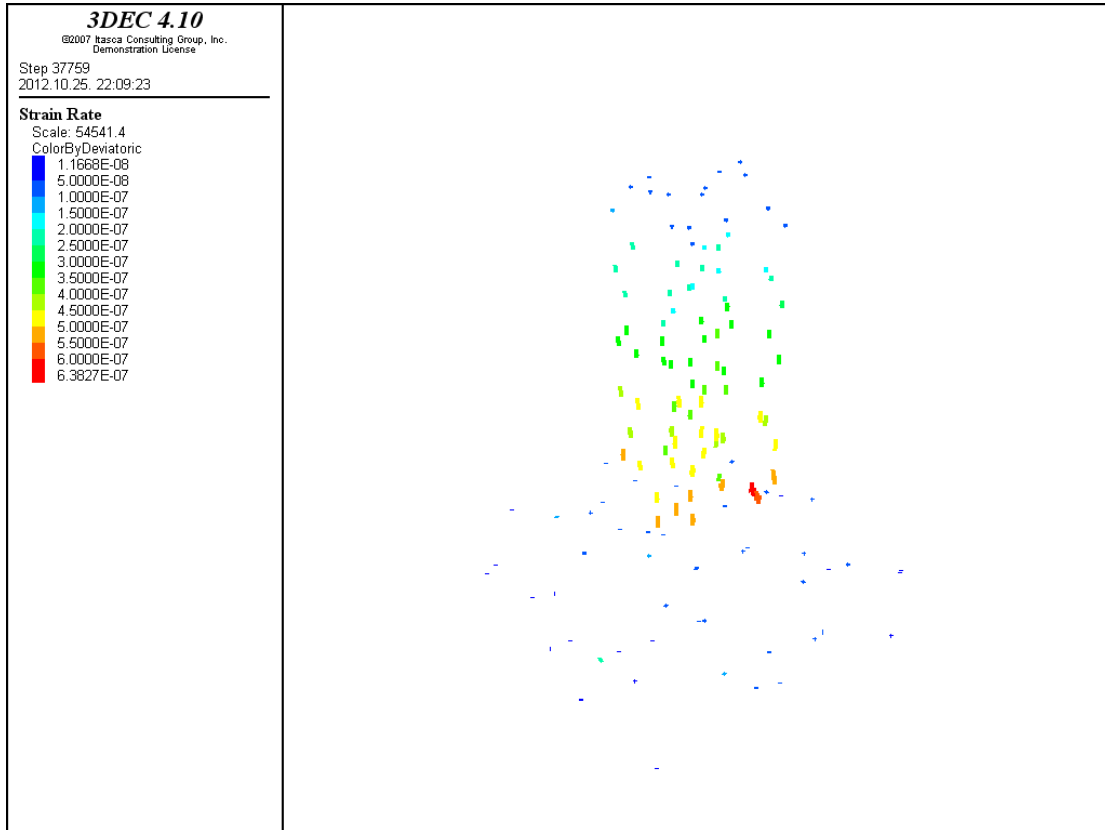


Fig. 123.: Result of „plot zonetensor strainrate colorby deviatoric line width 4” command in case only gravity is acting on the structure

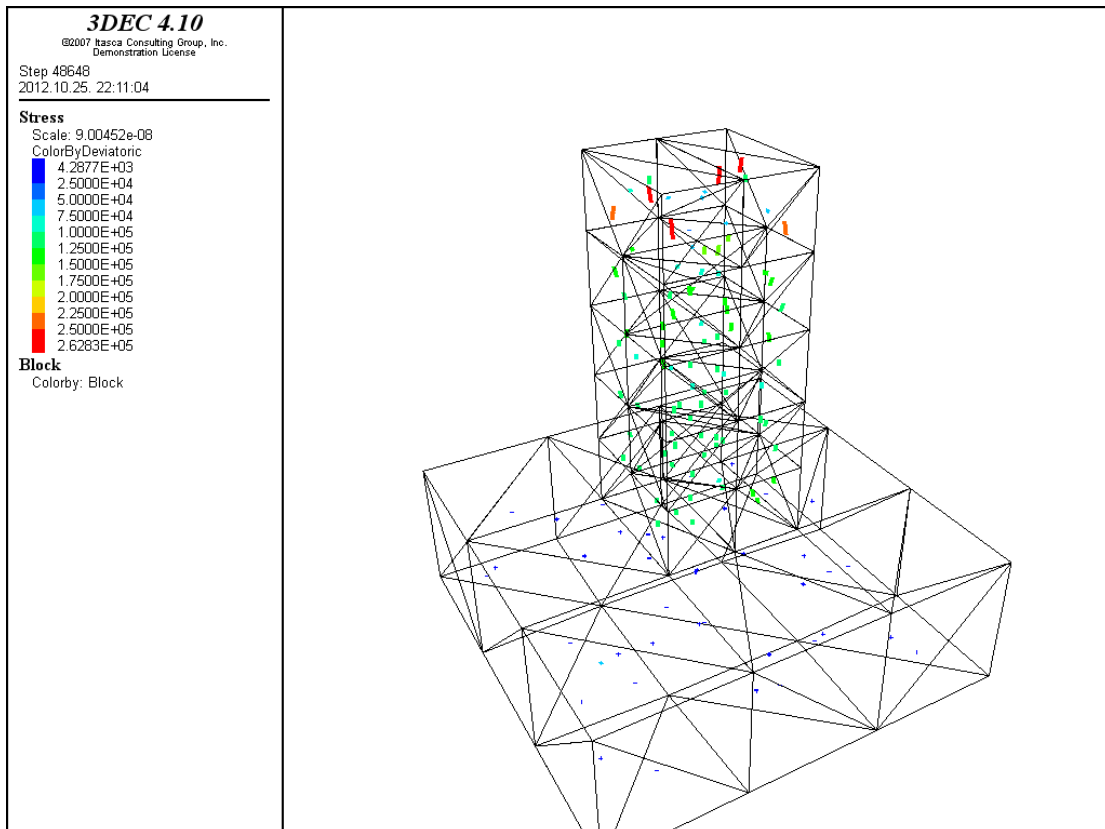


Fig. 124.: Result of „plot zonetensor stress colorby deviatoric line width 4” and „plot add block fill off” commands in case gravity and concentrated loads are acting on the structure

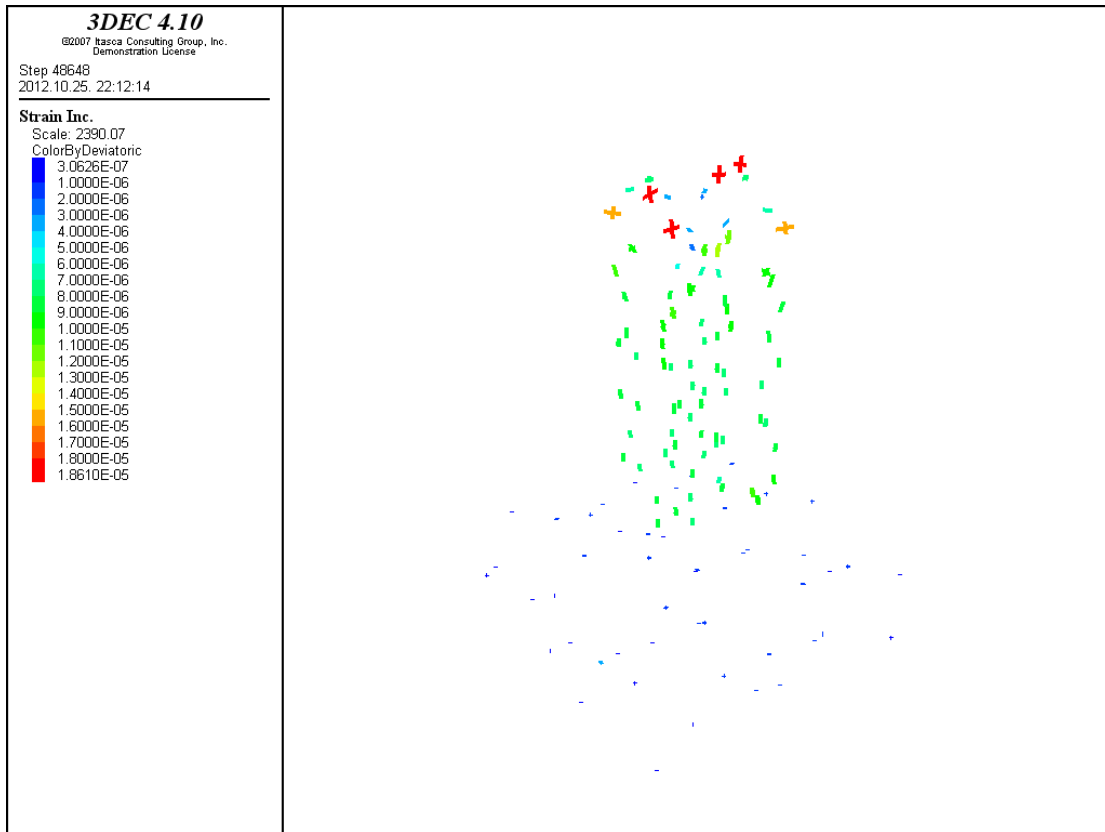


Fig. 125.: Result of „plot zonetensor ssi colorby deviatoric line width 4” command in case gravity and concentrated loads are acting on the structure

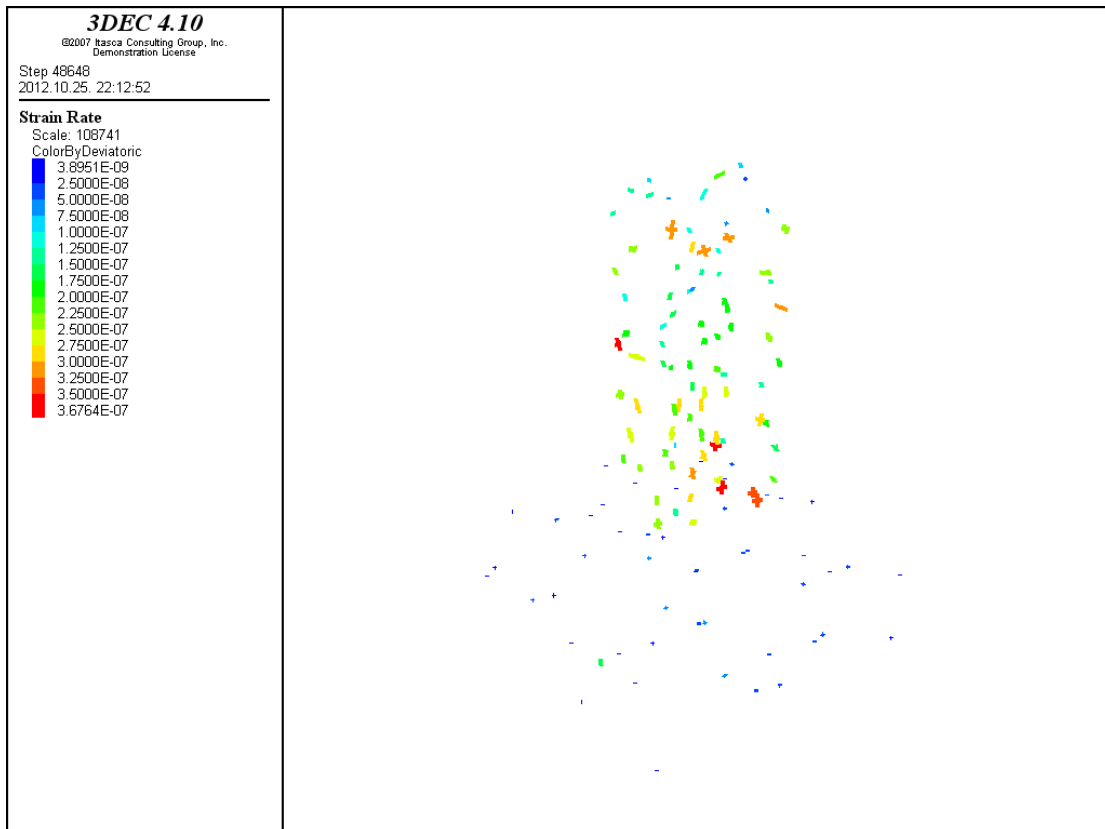


Fig. 126.: Result of „plot zonetensor ssr colorby deviatoric line width 4” command in case gravity and concentrated loads are acting on the structure

Bibliography:

[1] 3DEC Version 4.1 – Online manual