5. UDEC

5.1 Introduction

On a conference in 1971 Peter A. Cundall, the doctorial student of Imperial College, introduced a novel software called Universal Distinct Element Code (UDEC). This software (Cundall, 1971) aimed at simulating the behaviour of fractured rocks around tunnels, excavations, landslopes etc. This was the first discrete element software in the world.

Applying polygonal/polyhedral elements, UDEC and its three-dimensional version 3DEC is particularly suitable for the analysis of masonry structures too. The commercial versions of the 2D and 3D code are widely applied in the engineering practice. In comparison to the DDA models (see in Section 6) which are implicit but also use polyhedral elements, UDEC (which follows an explicit time integration scheme) is less accurate, and numerical stability problems more often occur. On the other hand, many users enjoy the well-developed input and output system, which is indeed a great advantage over the rather troublesme DDA research codes.

This introduction will first focus on the two-dimensional version, in the hope that it will give a good impression on the logic of UDEC, but avoids the complicated problems of three-dimensional geometrical analysis of the contacts. Then we shall turn the attention on the three-dimensional version.

5.2 UDEC: Two-dimensional modelling

5.2.1 The elements and the nodes

The elements in UDEC may have arbitrary polygonal shapes. They can be perfectly rigid: in this case they have a reference point, so the degrees of freedom and the equations of motion are in principle the same as those in BALL-type models. Alternatively, they can be deformable, and this option is more important for practical applications.

The elements are made deformable by being subdivided into uniform-strain simplexes, triangles in 2D (see Figure . The vertices of the simplexes will be those ***nodes*** in the analysis whose equations of motion will be solved. Nodes can take place either in the interior or on the boundary of the elements.

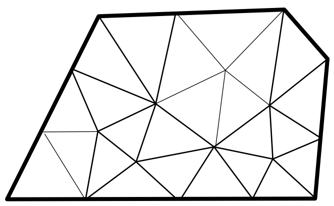


Figure 1.
Deformable element in UDEC, subdivided into simplexes

Since contacts between sharp corners, or between a corner and an edge, would cause stress peaks which are on one hand difficult to treat numerically, and on the other hand, would cause non-realistic damages, the corners of the elements are modified in UDEC: their neighbourhood is replaced with a rounded domain (see Figure 2). The rounding length is defined by the user (the same value is applied for every corner), but it should be definitely smaller than the smallest simplex edge, to make sure that only the corners are affected by rounding. This length gives the distance between the corner and between the point on the edge where the rounding arc touches the edge. (The difference between the younger or older, less or more damaged stone blocks can nicely be simulated with the help of appropriate rounding length.)

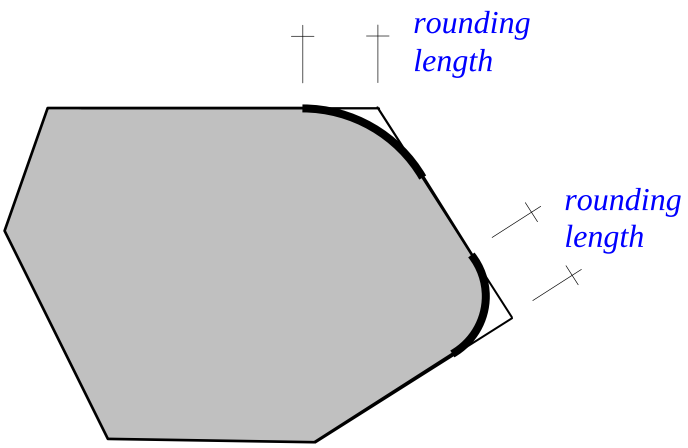


Figure 2.
Rounding of the corners in UDEC

The nodes only have translational degrees of freedom. The displacement vector of node p consists of three scalar component:

$$\mathbf{u}^p(t) = \begin{bmatrix} u_x^p(t) \\ u_y^p(t) \\ u_z^p(t) \end{bmatrix} ,$$

and these are collected into the total displacement vector of the system containing altogether N nodes:

$$\mathbf{u}(t) = \begin{bmatrix} \mathbf{u}^1(t) \\ \mathbf{u}^2(t) \\ \vdots \\ \mathbf{u}^N(t) \end{bmatrix} .$$

Within one simplex (called “zone” in UDEC terminology) the translation vector in any point can uniquely be determined through the linear interpolation of the translations of the vertices (this is the reason why simplexes are applied for the subdivision). Because of the linearity of the translation field, the gradient of this field is constant, hence the strain, i.e. the symmetric part of the translation gradient, is also uniform, and uniquely determined by the translations of the vertices.

Rounding affects only the recognition and geometrical characteristics of contacts. Any other characteristics are based on the original geometry without rounding. The calculation of strains in the zones at rounded corners is a good example for this: strains are calculated from the translations of the original nodes.

The constitutive relations of the deformable simplexes specify how to calculate the stress tensor if the strain tensor (and perhaps some kind of a history data of these state variables) is known. There are several optional constitutive relations offered by UDEC:

→ The simplest type is the "***null element***", an empty domain having zero material density and zero stiffness, playing no mechanical role. This type of "material" can be used, for instance, to simulate voids and holes.

→ The elements can be ***isotropic, linearly elastic***, with infinite resistance to stresses (no plastic or fracture failure limit). Such elements are characterized with the Young-modulus (E) and the Poisson-coefficient (μ) , or, alternatively, with the bulk modulus (K, the ratio between isotropic stress and strain) and the shear modulus(G). The two pairs of quantities can easily be calculated from each other:

$$K = \frac{E}{3(1-\mu)}; \quad G = \frac{E}{2(1+\mu)} \quad .$$

→ ***Failure conditions*** can also be assigned to the elements. The Mohr-Coulomb model, the Prager-Drucker model and many others are built-in options; but the user can also prepare his or her own failure criteria if the existing options are not suitable for the problem under consideration. These failure criteria set a limit to the stresses in the deformed zones, and describe how the zone should behave if a failure criterion is met. (Inside the same element, some zones may be in plastic or damaged state while others are still elastic, but the element always remains the set of the same zones as initially.)

To summarize, from the constitutive relations the stress state can be determined if the strain (and perhaps the stress and strain history) is known.

5.2.2 The contacts

For ***rigid*** blocks, a contact in *UDEC* is created at each corner interacting with a corner or edge of an opposing block. If the blocks are ***deformable*** (internally discretized), point contacts are created at all nodes located on the block edge in contact. Being more complicated, we shall focus now on the deformable case; after understanding it, the contact treatment in the first case is rather straightforward.

Contact is formed when a node belonging to an element gets into the interior of another element. In 2D there are three different ways it can happen:

1. Contact can be formed between an internal node on an edge of an element, and the edge of another element (see Figure 3a). Figure 3b shows the two elements slightly displaced, so that the figure could better be seen.

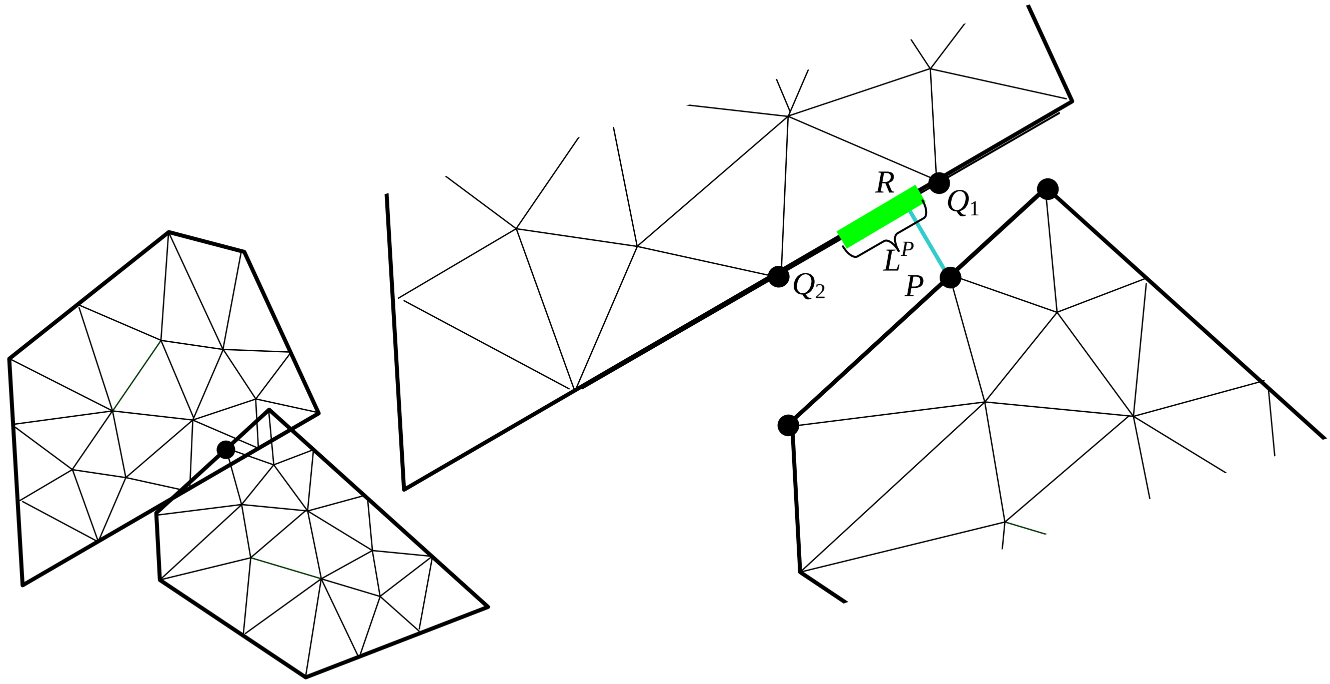


Figure 3.

a) Contact between an internal node and an edge

b) Length of the contact

The L^P length belonging to node P is defined in the following way:

– Draw a line normal to the other edge, i.e. the edge NOT containing P. This normal intersects with the edge in point R.

– The two nodes of the zone edge which contains R are Q_1 and Q_2.

– The half of the straight section RQ_1 , and the half of the straight section RQ_2 form the contact between P and the other element (see the thick green line in Figure 3b). Its length is simply

$$L^P = \frac{1}{2}\overline{RQ_1} + \frac{1}{2}\overline{RQ_2} \ .$$

– The direction of the contact normal vector is the same as that of PR so the contact normal is perpendicular to the edge Q_1Q_2.

– The normal deformation of the contact is the same as the distance between P and R, so it is the depth by which P got into the interior of the other element.

2. Contact can be formed when the rounded corner of an element gets on the internal side of an edge of another element. The contact normal vector in this case is perpendicular to the edge of the other element. The length of the contact is simply the distance between the intersection points of the rounded arc and the edge, as shown in Figure 4. (Note that the contact still does not exist if only the original corner gets into the interior of the other element, but there is no intersection between the rounded boundary of the first element and the edge of the other element.)

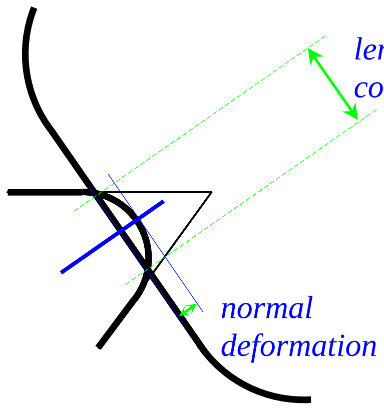


Figure 4.
Contact between a rounded corner and an edge

3. Finally, contact can be formed by two rounded corners. In this case the contact normal vector is determined by the straight line connecting the two arc centres, and the contact length is the distance between the intersection points of the two arcs. The deformation of the contact in normal direction is equal ti the magnitude of maximal overlap.

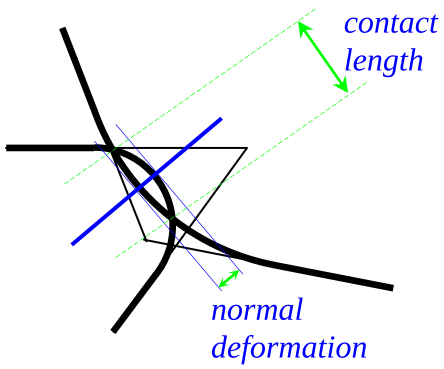


Figure 5.
Contact between two corners

When two elements get so close to each other that their boundary domains overlap, usually a series of contacts is formed: several contacts belonging to the neighbouring nodes on both elements exist. The interaction between the two elements is given by a set of contacts belonging to the neighbouring nodes along the edges of the two elements.

Independently of how the contact is formed, a distributed force – assumed to be of constant intensity – is transmitted from one element to the other. This distributed force has a normal and a tangential component, σ_n^k and σ_s^k respectively. The superscript k denotes that the k-th contact (which is perhaps just one part of the interaction of two elements). Two consecutive contacts may carry different distributed forces, so a stair-like distribution may exist along the complete interaction.

The contact area is the L^k length of the k-th contact times the unit width of the whole 2D model.

Let Δu_n^k and Δu_s^k denote the increment of the normal and tangentional relative translation in contact k. The intensity of the normal and tangemtional components of the distributed contact force is modified:

$$\Delta\sigma_n^k = -k_n^k \Delta u_n^k$$

(negative sign expresses that compression is positive in UDEC), and

$$\Delta\sigma_s^k = k_s^k \Delta u_s^k \quad .$$

The stiffnesses k_n and k_s are constant in the case of linear relation, but theoretically they can depend in any way on the actual contact forces and deformations. UDEC offers several constitutive relations for the contacts (but the user can also prepare his/her own relations):

→ linearly elastic, non-cohesional contact with Couomb-friction ("*fric*" is the friction angle):

$$\Delta\sigma_n^k = -k_n^k \Delta u_n^k \; ; \quad \sigma_n^k > 0$$
$$\Delta\sigma_s^k = k_s^k \Delta u_s^k \; ; \quad \left|\sigma_s^k\right| \le \tan(fric)\cdot\sigma_n^k$$

→ Coulomb-friction with cohesion and non-zero tensional failure limit:

$$\Delta\sigma_n^k = -k_n^k \Delta u_n^k \; ; \quad \sigma_n^k > -\sigma_n^{húzó}$$
$$\Delta\sigma_s^k = k_s^k \Delta u_s^k \; ; \quad \left|\sigma_s^k\right| \le (coh) + \tan(fric)\cdot\sigma_n^k$$

→ different models for the behaviour after sliding has started (e.g. plastic models).

To summarize, these models describe how to determine the distributed forces between the elements, when the displacements are already known at the end of the time step. Then at the beginning of the next time step these forces are applied when calculating the reduced force vectors.

5.2.3 Time integration of the equations of motion

It was explained in Section 2.3 that the equations of motion of node p can be written in the form

$$m^p \mathbf{a}_i^p = \mathbf{f}_i^p$$

where m^p is the mass of the Voronoi-cell of the node, and $\mathbf{f}^p$ is the resultant of the distributed forces acting on the Voronoi-cell. This force is the result of different effects:

→ external loads (e.g. weight or drag force);

→ stresses inside the simplexes cut by the boundaries of the Voronoi-cell of the p-th node;

→ if the node is on the edge of the element, contact force may also act on p.

The force $\mathbf{f}^p$ is assumed to go through the node, and its eccentricity is neglected.

According to the method of central differences, the discretized form of the equations of motion is:

$$m^p \frac{\mathbf{v}^p_{i+1/2} - \mathbf{v}^p_{i-1/2}}{\Delta t} = \mathbf{f}^p_i$$

Using this, the analysis of the time interval (t_i, t_{i+1}) of length Δt is done in the following way:

- At t_i the positions, and the reduced forces acting on the nodes is known.
- The velocities belonging to the middle of the previous time interval is known (which are considered to be the average velocity of the previous interval).
- The new characteristics:

$$\mathbf{v}^p_{i+1/2} = \mathbf{v}^p_{i-1/2} + \Delta t \cdot \frac{1}{m^p} \mathbf{f}^p_i$$
$$\mathbf{u}^p_{i+1} = \mathbf{u}^p_i + \Delta t \cdot \mathbf{v}^p_{i+1/2}$$

Similarly to the BALL-type models, different damping methods (see below) are applied to decrease the numerical instability problems. In addition, the time step length is limited:

$$\Delta t \le 2 \cdot \min_{(p)} \left\{ \sqrt{\frac{m^p}{k^p}} \right\}$$

where k^p is the translational stiffness of node p (see the manual of UDEC for details, Itasca (2011)).

5.2.4 Damping

Like BALL-type models, UDEC applies different types of damping, partly to simulate energy dissipation, and partly to improve convergence.

Adaptive global damping works in such a way that a velocity-proportional damping is used, but the viscosity constant is continuously adjusted to ensure that the calculated change of kinetic energy during the actual timestep is cut back by a user-defined ratio, for every degree of freedom. If this ratio is, for example, 0.55, it means that every reduced force component is modified separately in such a way that the change of kinetic energy belonging to that degree of freedom would be 55% less than calculated by the explicit solver. If the system tends to an equilibrium state (e.g. to a state with constant velocities), this kind of damping gradually disappears, and becomes zero in the equilibrium state. Adaptive global damping is very helpful and efficient if the system strongly oscillates around the equilibrium state.

Local damping is the same as that explained in Section 4. for the BALL-type models: a damping force component is added to the reduced forces corresponding to every degree of freedom:

$$v^p_{x,i+1/2} = v^p_{x,i-1/2} + \Delta t \cdot \frac{1}{m^p} \left(f^p_{x,i} - \alpha \cdot \left| f^p_{x,i} \right| \cdot \frac{v^p_{x,i-1/2}}{\left| v^p_{x,i-1/2} \right|} \right) \quad .$$

The α coefficient is chosen by the user (e.g. 0.8). This kind of damping is particularly advantageous if some parts of the system are already close to equilibrium while other others are just collapsing or strongly oscillating; or if the loads quickly change.

Strangely enough, UDEC does not offer a built-in contact damping option. The user can, however, prepare and code any arbitrary individual constitutive relation, so contact damping can be simulated this way.

5.3 3DEC: Three-dimensional modelling

3DEC, the three-dimensional version of UDEC, works in basically the same way as UDEC. According to Itasca (2007) and (2011), the most important differences are the following:

5.3.1 The elements

Similarly to UDEC, there are two types of polyhedral block that can be modelled in 3DEC: rigid blocks, which have now six degrees of freedom (three translational and three rotational); and deformable blocks, which are subdivided internally into tetrahedra that have three (instead of two like in UDEC) translational degrees of freedom at each vertex (node). Rigid as well as deformable blocks must have planar faces that are polygons of any number of sides. For deformable blocks each original planar polygonal face is discretized into triangular sub-faces, in accordance with the internal discretization into tetrahedra. (In some versions there is an alternative option to apply 10-node tetrahedral elements. based on quadratic displacement interpolations functions. For this purpose, new nodes are created at the midpoint of every zone edge. The higher-order element formulation allows a quadratic displacement field to be represented inside the zone and also on the zone faces. However, for purposes of contact calculations, and for plotting, in this case the block boundary is still approximated by a mesh of triangular faces. Each face of a 10-node tetrahedron is divided into 4 plane triangles.)

In general, elements in 3DEC may be convex or concave, may contain holes, may be multiply connected. However, there are so many advantages to convex blocks that within the program concave blocks are decomposed into two or more convex blocks: one is termed a “master block”; the others are “slave blocks.” In contact detection and contact analysis the slave blocks are treated in exactly the same way as master blocks, in order to take advantage of convexity. However, in the mechanical calculations, the whole block (master and slaves) is treated as one unit: a common centre of gravity, a common mass, etc. are determined.

5.3.2 The contacts

An important difference from UDEC is that while in UDEC rounding is applied to identify and characterize contacts of corners, in 3DEC contact recognition and the definition of contact characteristics happen in a different manner.

When two blocks are close to each other, they are tested for contact. (If they are not in contact, the maximum gap between them must be determined so that block-pairs separated by more than a certain tolerance may be ignored.) For block-pairs separated by less than this tolerance, but not touching, a “contact” is still formed. Though such a “contact” carries no load, it is tracked at every step in the mechanical calculation, to ensure that interaction forces start to act as soon as the blocks touch. The contact-detection logic must also provide a unit normal vector, which defines the plane along which sliding can occur. This unit normal should change direction in a continuous fashion as the two blocks move relative to one another. Finally, the contact-detection logic must classify the type of the contact rapidly—e.g., face-to-edge or vertex-to-face. This information is needed in order to select the most appropriate physical law to apply at each contact. In summary, the contact-detection logic must supply, with as little delay as possible, the contact type (if touching), the gap (if not touching), and the unit normal vector. 3DEC applies a scheme based on a common plane between the two blocks.

The analysis consists of the following two parts: (1) determining a “common-plane” that, in some sense, bisects the space between the two blocks; and (2) testing both blocks separately for contact with the common-plane.

The “common-plane” is analogous to a metal plate that is held loosely between the two blocks. If the blocks are held tightly and brought together slowly, the plate will be deflected by the blocks and will become trapped at some particular angle when the blocks finally come into contact. Whatever the shape and orientation of the blocks (provided they are convex), the plate will take up a position that defines the sliding plane for the two blocks. To carry the analogy a bit further, imagine that the plate is now repelled by the blocks even when they do not touch. As the blocks are brought together, the plate will take up a position midway between them, at a maximum distance from both. Then we can easily find the gap between the blocks, simply by adding the block-to-plate distances.

The algorithm for locating and moving the common plane is based on geometry alone, and is applied at every timestep, in parallel with the mechanical calculations. The algorithm is stated as “Maximize the gap between the common-plane and the closest vertex”. For overlapping blocks, the same algorithm applies, but the words “gap” and “closest” must be used in their mathematical sense for the case of negative signs—i.e., gap means “negative overlap” and closest means “most deeply buried.” To improve readability, the algorithm may be restated for the case of overlapping blocks: “Minimize the overlap between the common-plane and the vertex with the greatest overlap”. The algorithm then applies a translation and a rotation to the common-plane in order to maximize the gap (or minimize the overlap).

Contact interaction exists if the overlap is positive, or equivalently, if the gap is negative between the two blocks. The normal vector of the common-plane is the contact normal; and the contact characteristics can easily be determined from simple geometrical considerations.

If a block face is in contact with the common-plane, then it is automatically discretized into ***sub-contacts***. For rigid blocks, faces are triangulated to create the sub-contacts. These sub-contacts are generally created at the vertices of the block face. For deformable blocks, the triangular faces of tetrahedral zones at the block surface contain a number of internal surface nodes, each of which has three independent degrees of freedom. In this case, a sub-contact is created for each node on the face.

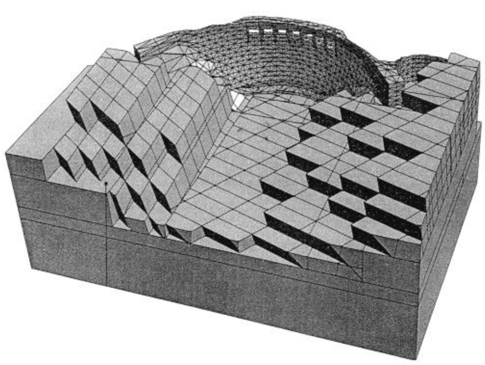
Two types of sub-contact are defined: vertex-to-face and edge-to-edge. In order to simulate face-to-face contact, each sub-contact is assigned an area allowing standard joint constitutive relations, formulated in terms of stresses and displacements, to be applied. Edge-to-edge sub-contacts model both edge-to-edge contact between blocks, and face-to-face and face-to-edge contacts at the points of intersection of edges on the common-plane. The interface displacement at each sub-contact is taken as the sub-contact displacement minus the displacement of the coincident point on the opposing face. The area "owned" by each sub-contact is, in general, equal to one-third of the area of the surrounding triangles, but this calculation must be adjusted when the sub-contact is close to one or more edges on the opposing block. If the other side of the interface is also a face, then identical conditions apply: sub-contacts are created, and relative displacements, and hence forces, are calculated.

When two blocks come together, the contact logic described above is equivalent to two sets of contact springs in parallel — in this case, the forces from both sets are divided by two, so that the overall interface behaviour is the average of that of both sets.

5.4 Applications

UDEC/3DEC is the most widely used discrete element technique to date in the engineering practice. A few interesting applications:

– earthquake analysis of the Funcho dam and the Camambe dam (Lemos, 1996; Lemos, 1999);



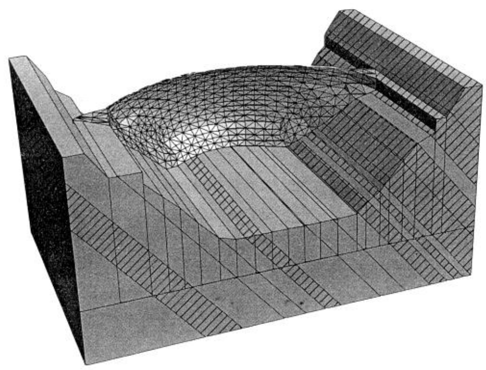


Figure 6.

Camambe dam *Funcho dam*

– Underground ice hockey stadium, Norway, Gjovik (Chryssanthakis and Barton, 1999);

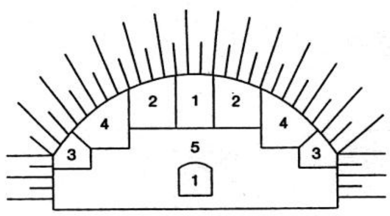


Figure 7.

Overview of the stadium in Gjovick *cross-section and excavation order*

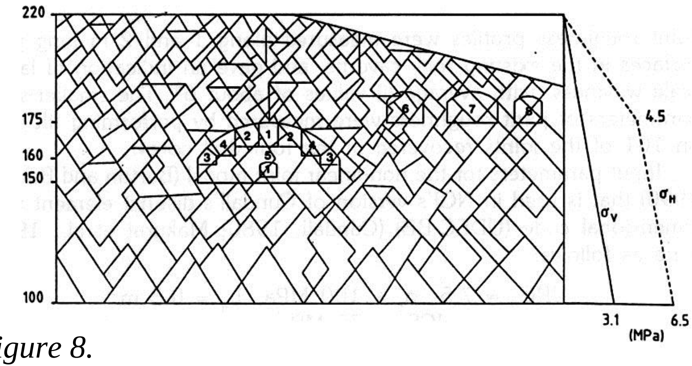


Figure 8.
UDEC model of the fractured rock in Gjovick (Chryssanthakis and Barton, 1999)

– Weathering of the Nishida Bridge, Jiang and Esaki (2002):

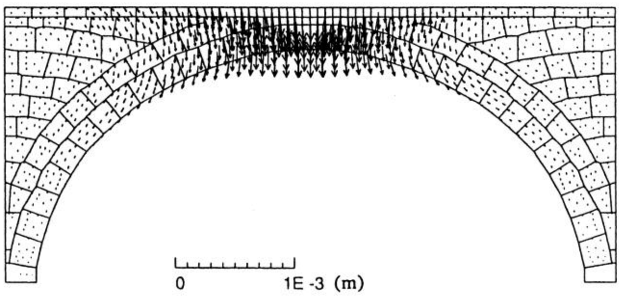


Figure 9.
Nishida bridge and it UDEC model,
Jiang and Esaki (2002)

Questions

5.1. What are the degrees of freedom of the deformable elements in UDEC and in 3DEC?
5.2. What is the meaning of the rounding length? How can a contact be formed between two elements, and what is the size and direction of the contact in the different cases?
5.3. Introduce the most important contact types of UDEC!
5.4. Explain how a time step is analysed in UDEC!
5.5. What are the most important differences between UDEC and 3DEC?
5.6. What kinds of damping are applied in UDEC, and why are they necessary?