2 FLUID FLOW IN JOINTS

2.1 Introduction

UDEC has the capability to perform the analysis of fluid flow through the fractures of a system of impermeable blocks. A fully coupled mechanical-hydraulic analysis is performed, in which fracture conductivity is dependent on mechanical deformation and, conversely, joint fluid pressures affect the mechanical computations. The effects modeled in *UDEC* are summarized in Figure 2.1:

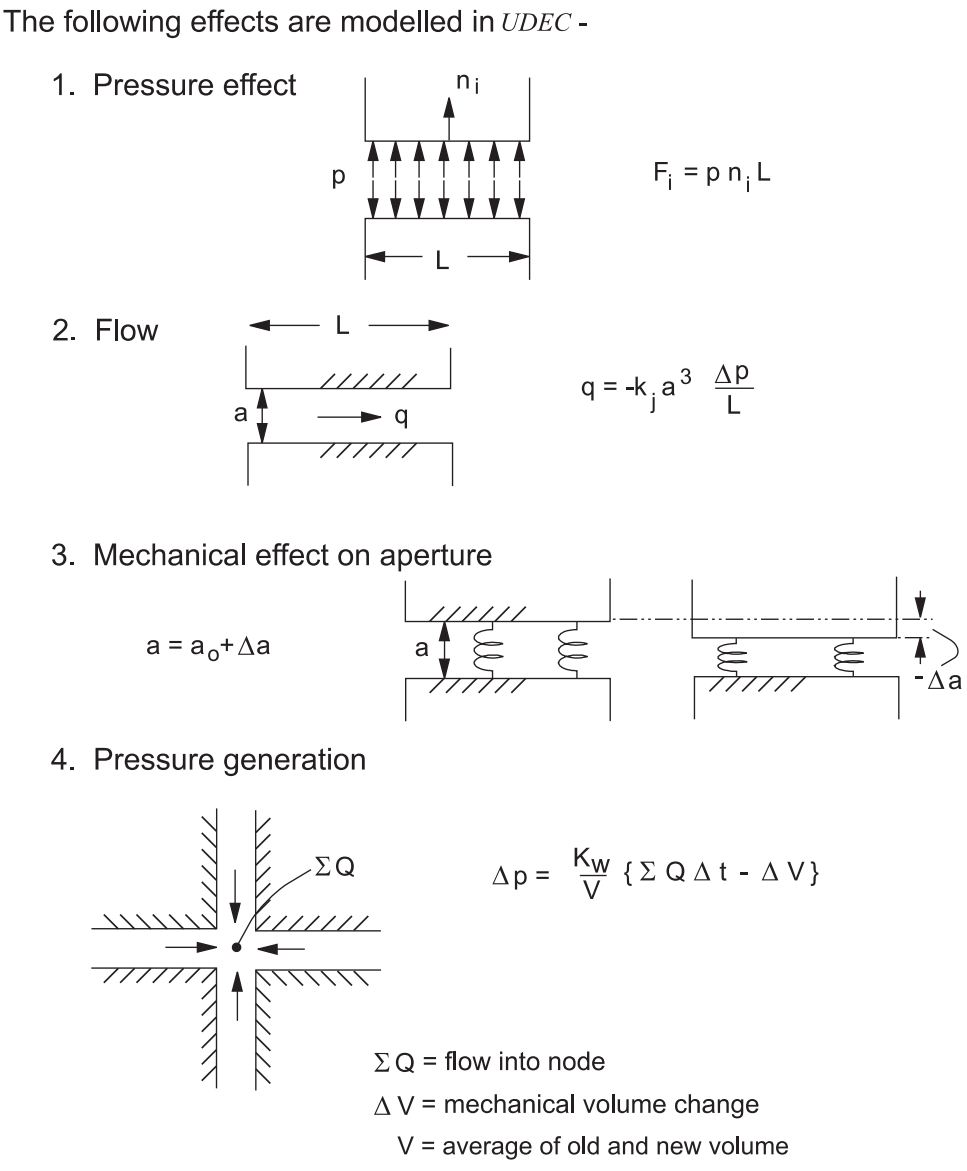

Figure 2.1 Fluid/solid interaction in discontinua

This fluid-flow formulation is called *compressible flow*, and is the basic formulation for flow of a fluid in *UDEC*. Both confined flow and flow with a free surface can be modeled with this formulation. The fluid-flow calculation can also be run either coupled or uncoupled with the mechanical stress calculation. The formulation is described in Section 2.2.1.

In addition to the basic formulation, there are four other fluid-flow modes in *UDEC*:

A computationally faster solution scheme than the basic formulation has been developed specifically for steady-state flow problems. This scheme, called the *steady-state flow*, is described in Section 2.2.2.

An alternative algorithm is available for transient flow, assuming an incompressible fluid. This formulation can greatly speed the calculation for transient analysis, but is restricted to simulations of the mechanical-fluid coupled response of systems undergoing quasi-static loading. This formation, called *incompressible* or *fast-flow*, is described in Section 2.2.3.

A special algorithm for modeling transient gas flow is also provided. The *gas-flow* is described in Section 2.2.4.

Finally, two-phase liquid/gas flow in joints can be modeled. The *two-phase flow* formulation is described in Section 2.2.5.

Special fluid-flow features are provided in *UDEC*. The joint permeability relation can be modified, and viscoplastic fluid flow can be simulated; this is described in Section 2.2.6. A porous medium can be created around the *UDEC* block model to simulate a regional flow field. This algorithm is described in Section 2.2.7. One-way thermal-hydraulic coupling of flow in joints in which temperature variations induce changes in the viscosity and density of water can also be simulated. This logic is discussed in Section 2.2.8.

Background information on the hydraulic behavior of rock joints and the associated fluid flow properties are described in Section 2.3. The *UDEC* commands required to perform a fluid-flow analysis are listed in Table 2.2, and discussed in detail in Section 2.4.

Examples and verification problems illustrating each of the fluid-flow modes are given in Section 2.5.

*.

---

* The data files in this section are stored in the folder "ITASCA\UDEC700\Datafiles\Fluid" with the extension ".DAT." A project file is also provided for each example. For the *GIIC*, open the project file by clicking on the FILE / OPEN PROJECT menu item and select the project file name (with ".PRJ" extension). Then click on the *Project Options* icon at the top of the *Project Tree Record*, select *Rebuild unsaved states*. For the GUI, open the project file by clicking on the FILE / OPEN PROJECT menu item and select the project file name (with ".UDPRJ" extension). Then click on the *Project* tab and select the "Master.dat" and run it

2.2 Fluid-Flow Formulations

2.2.1 Basic Algorithm – Transient Flow of a Compressible Fluid

The numerical implementation for fluid flow makes use of the *domain* structure described in Section 1 in **Theory and Background** (see Figure 2.2). For a closely packed system, there is a network of domains, each of which is assumed to be filled with fluid at uniform pressure and which communicates with its neighbors through contacts. Referring to Figure 2.2, domains are numbered 1 to 5: domains 1, 3 and 4 represent joints; domain 2 is located at the intersection of two joints; and domain 5 is a void space. As described in Section 1 in **Theory and Background**, domains are separated by the contact points (designated by letters A to F in the figure), which are the points at which the forces of mechanical interaction between blocks are applied. Because deformable blocks are discretized into a mesh of triangular elements, gridpoints may exist not only at the vertices of the block, but also along the edges. A contact point will be placed wherever a gridpoint meets an edge or a gridpoint of another block. For example, in the same figure, contact D implies the existence of a gridpoint along one of the edges in contact. As a consequence, the joint between the two blocks is represented by two domains: 3 and 4. If a finer internal mesh were adopted, the joint would be represented by a larger number of contiguous domains. Therefore, the degree of refinement of the numerical representation of the flow network is linked to the mechanical discretization adopted, and can be defined by the user.

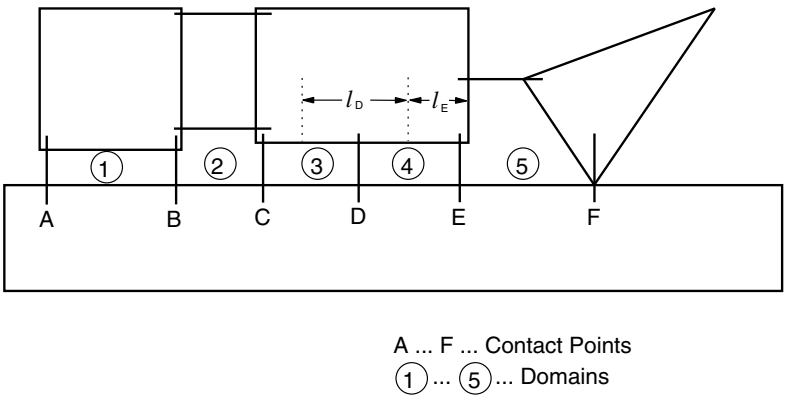

Figure 2.2 Flow in joints modeled as flow between domains

In the absence of gravity, a uniform fluid pressure is assumed to exist within each domain. For problems with gravity, the pressure is assumed to vary linearly according to the hydrostatic gradient, and the domain pressure is defined as the value at the center of the domain.

Flow is governed by the pressure differential between adjacent domains. The flow rate is calculated in two different ways, depending on the type of contact. For a point contact (i.e., corner-edge, as contact F in Figure 2.2, or corner-corner), the flow rate from a domain with pressure p_1 to a domain with pressure p_2 is given by

$$q = -k_c \, \Delta p \tag{2.1}$$

where k_c = a point contact permeability factor, and

$$\Delta p = p_2 - p_1 + \rho_w \, g \, (y_2 - y_1) \tag{2.2}$$

where ρ_w is the fluid density;
g is the acceleration of gravity (assumed to act in the negative y-direction); and
y_1, y_2 are the y-coordinates of the domain centers.

In the case of an edge-edge contact, a contact length can be defined (e.g., in Figure 2.2, l_D and l_E denote the lengths of contacts D and E, respectively). The length is defined as half the distance to the nearest contact to the left plus half the distance to the nearest contact to the right. In this case, the cubic law for flow in a planar fracture (e.g., Witherspoon et al. 1980) can be used (see Section 2.3). The flow rate is then given by

$$q = -k_j \, a^3 \, \frac{\Delta p}{l} \tag{2.3}$$

where k_j is a joint permeability factor (whose theoretical value is $1/12\mu$);
μ is the dynamic viscosity of the fluid;
a is the contact hydraulic aperture; and
l is the length assigned to the contact between the domains.

In *UDEC*, the user may change the permeability factor and exponent in Eq. (2.3). (See Section 2.2.6.) The above expression may be used for point contacts, provided a minimum length is assigned to these contacts.

Eq. (2.2) indicates that flow may take place at a contact even when both domain pressures are zero; in this case, gravity may cause fluid to migrate from a domain that is not fully saturated. However, there are two factors to consider:

(a) the apparent permeability should decrease as the saturation decreases – in particular, permeability should be zero for zero saturation; and

(b) fluid cannot be extracted from a domain of zero saturation.

To address point (a), the flow rates (and, thereby, apparent permeability) computed by Eqs. (2.1) and (2.3) are multiplied by a factor f_s, a function of saturation, s,

$$f_s = s^2(3 - 2s) \tag{2.4}$$

The function is empirical but has the property that $f_s = 0$ if $s = 0$, and $f_s = 1$ if $s = 1$ (i.e., permeability is unchanged for full saturation, and zero for zero saturation). Further, the derivative

of Eq. (2.4) is zero at $s = 0$ and $s = 1$, which is reasonable to expect, physically. The value of s used in Eq. (2.4) is taken as the saturation of the domain from which inflow takes place; hence, inflow cannot occur from a completely unsaturated domain.

The hydraulic aperture is given, in general, by

$$a = a_o + u_n \tag{2.5}$$

where a_o is the joint aperture at zero normal stress; and
u_n is the joint normal displacement (positive denoting opening).

A minimum value, a_{res}, is assumed for the aperture, below which mechanical closure does not affect the contact permeability. A maximum value, a_{max}, is also assumed, for efficiency, in the explicit calculation (arbitrarily set to five times a_{res}, but it can be changed by the user). The variation of aperture with normal stress on the joint is depicted in Figure 2.3. The above expression is a very simple relation between joint mechanical and hydraulic apertures; more elaborate relations, such as the empirical law proposed by Barton et al. (1985), might also be used.

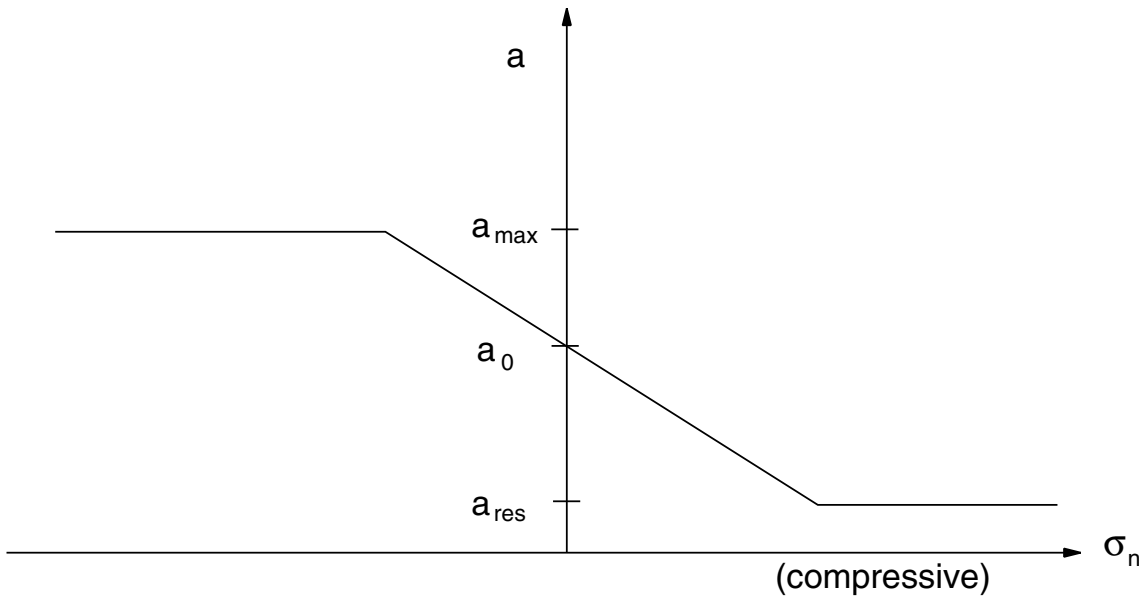

Figure 2.3 Relation between hydraulic aperture, a, and joint normal stress, σ_n, in UDEC

At each timestep in the mechanical calculation in *UDEC*, the computations determine the updated geometry of the system, thus yielding the new values of apertures for all contacts and volumes of all domains. Flow rates through the contacts can be calculated from the above formulas. Then, domain pressures are updated, taking into account the net flow into the domain, and possible changes in

domain volume due to the incremental motion of the surrounding blocks. The new domain pressure becomes

$$p = p_o + K_w \, Q \, \frac{\Delta t}{V} - K_w \, \frac{\Delta V}{V_m} \tag{2.6}$$

where p_o is the domain pressure in the preceding timestep;
Q is the sum of flow rates into the domain from all surrounding contacts;
K_w is the bulk modulus of the fluid;
ΔV = $V - V_o$; and
V_m = $(V + V_o)/2$,
where V and V_o are the new and old domain areas, respectively.

If the new domain pressure computed by Eq. (2.6) is negative, then the pressure is set to zero, and the domain outflow is used to reduce saturation, s, as follows.

$$s = s_o + Q \frac{\Delta t}{V} - \frac{\Delta V}{V_m} \tag{2.7}$$

where s_o is the domain saturation at the preceding timestep. The pressure remains at zero as long as $s < 1$; in this case, Eq. (2.7) is applied instead of Eq. (2.6). If the computed s is greater than 1, then s is set to 1, and Eq. (2.6) is used again. This scheme ensures that fluid mass is conserved; the excess domain volume is either used to change pressure or to change saturation. The phreatic surface (boundary between nodes with $s = 1$, and nodes of $s < 1$) arises naturally from the algorithms described above.

Given the new domain pressures, the forces exerted by the fluid on the edges of the surrounding blocks can be obtained. These forces are then added to the other forces to be applied to the block gridpoints, such as the mechanical contact forces and external loads. As a consequence of this procedure, *total* stresses will result inside the impermeable blocks, and *effective* normal stresses will be obtained for the mechanical contacts.

Numerical stability of the present explicit fluid flow algorithm requires that the timestep be limited to

$$\Delta t_f \; = \; \min \left[\frac{V}{K_w \; \sum_i \; k_i} \right] \tag{2.8}$$

where V is domain volume, and the summation of permeability factors, k_i, is extended to all contacts surrounding the domain (i.e., $k_i = \max(k_c, k_j \; a^3/l)$.

The minimum value of Δt over all domains is used in the analysis.

For transient flow analysis, the numerical stability requirements may be rather severe, and may make some analyses very time-consuming or impractical, especially if large contact apertures and very small domain areas are present. Furthermore, the fluid filling a joint also increases the apparent joint stiffness by K_w/a, thus possibly requiring a reduction of the timestep used in the mechanical calculation.

An optional method for calculating the transient compressible fluid-flow timestep has been added. The default method is to base the timestep on the lowest domain area and the highest actual contact aperture. For most models, this is overly conservative. The new option invoked by **block fluid timestep mode = 3** changes the calculation to consider the domain area and the aperture for each contact. This usually results in a much higher fluid timestep.

2.2.2 Steady-State Flow Algorithm

In many studies, only the final steady-state condition is of interest. In this case, several simplifications that make the present algorithm very efficient for many practical problems are possible. The steady-state condition does not involve the domain volumes. Thus, these can be scaled to improve the convergence to the solution. A scheme that was found to produce good results consists of assigning to a given domain a volume V which, inserted in the timestep expression above, leads to the same timestep for all domains. The contribution of the change in domain volume to the pressure variation can also be neglected, thus eliminating the influence of the fluid stiffness in the mechanical timestep, and making it unnecessary to specify fluid bulk modulus. Furthermore, as the steady-state condition is approached, the pressure variation in each fluid step becomes very small, allowing the execution of several fluid steps for each mechanical step without loss of accuracy. An adaptive procedure that "triggers" the update of the mechanical quantities whenever the maximum increment of pressure in any domain exceeds some prescribed tolerance (for example, 1% of the maximum pressure) was implemented in *UDEC*.

2.2.3 Transient Flow of an Incompressible Fluid

In the standard formulation described in Section 2.2.1, the fluid pressure increments are calculated from the joint volume variation and the net inflow into the domain (Eq. (2.6)). For small joint apertures, the fluid appears to be a stiff spring, with a stiffness higher than the typical joint stiffness. In an explicit algorithm, this implies that the mechanical timestep must be reduced. The fluid timestep, calculated by Eq. (2.8), is inversely proportional to the bulk modulus and joint conductivity. For typical joint apertures, fluid timesteps on the order of milliseconds are obtained. Therefore, this algorithm can only be applied in practical problem settings to short-duration simulations.

A different procedure, which overcomes the difficulties described above, has been developed. Before presenting this scheme, it is worthwhile to review the essential characteristics of a fluid-rock system and the particular conditions to be modeled.

The characteristics of a rock-fluid system can be summarized as follows.

1. There are two distinct difficulties that modelers confront:

a. the fluid trapped in a joint appears to be very stiff, owing to the small aperture; and

b. permeability varies rapidly with changing aperture, owing to the cubic term in the flow equation.

The two difficulties are separate and can be addressed separately. For example, if the imposed pressure changes are small compared to the existing pressures (and rock stresses), then item (b) is unimportant.

2. It is the *rock*, rather than the fluid, that determines fluid pressure. In a conventional pipe network, for example, the fluid determines its own pressure, via the flow and continuity equations. However, a typical rock block is so soft compared to the fluid trapped in a joint (factors of 10^3 to 10^4 are common) that significant changes in fluid volume hardly affect rock stresses. Since the rock's normal stresses must balance the fluid pressure in the neighboring joints, the fluid pressure is determined by the rock stress. It then follows that spatial variations in rock stress are directly responsible for the direction and magnitude of flow since flow occurs in response to pressure gradients.

3. If the interest is in nonsteady flow but not dynamic flow (i.e., the model must accurately capture the transmission delay as pressure fluctuations migrate from one part of the system to another), then inertial effects or wave-propagation effects can be neglected. We confine our attention to a liquid, such as water, contained in joints of small aperture (in the range 10 to 100 micrometers) and length in the range 1 m to 10 m.

4. Time constants (e.g., *time to reach a given fraction of the final pressure* or *time for a pressure change to propagate a certain distance*) depend more on the compliance of the whole system and the mode of deformation than on local compliance; the bulk modulus of the fluid is almost irrelevant. As an extreme example, imagine fluid injected into a series of joints bounded by blocks loaded by deadweight only (see Figure 2.4). The speed of propagation in this case is zero, since no pressure gradients develop between blocks. As stiffness is added between blocks, the propagation speed increases.

Keeping these points in mind, the following scheme is proposed. Flow rate is calculated from pressure difference in the usual way, as expressed previously in Eq. (2.3). At each "domain" (intersection of several joints or the middle part of a joint), the flow contributed by each joint is added algebraically and multiplied by the fluid timestep to obtain the net fluid volume entering the domain:

$$\Delta V_f = \sum q \ \Delta t_f \tag{2.9}$$

Instead of trying to translate this volume immediately into rock displacements, we imagine that the excess fluid is stored in a "balloon" attached to the domain. We then hold flow time constant, and

allow the contents of each balloon to leak into its associated domain. This leakage stops when the increase in domain volume becomes equal to the volume stored in the balloon. The process involves the usual dynamic relaxation of the equations of motion of the gridpoints, but with an additional pressure boundary condition supplied by the leakage of fluid into the domain. The following leakage scheme was found to be satisfactory.

$$p' = p^0 + F_p(\Delta V_{stored} - \Delta V_{domain}) \tag{2.10}$$

where p' and p^0 are the domain pressures at the new and old (mechanical) timesteps, respectively, ΔV_{stored} is the volume originally stored in the balloon, ΔV_{domain} is the volume increase of the domain, and F_p is a constant factor. Eq. (2.10) can be viewed as a kind of servo control that adjusts pressure until the domain volume has increased by exactly the volume stored in the balloon. There are no problems of compatibility, as we are simply applying a pressure boundary condition, rather than enforcing a velocity condition. During the leakage phase, we are solving all the mechanical equations by dynamic relaxation, so condition 4 (given above) is satisfied – the fluid "sees" the compliance of the entire system.

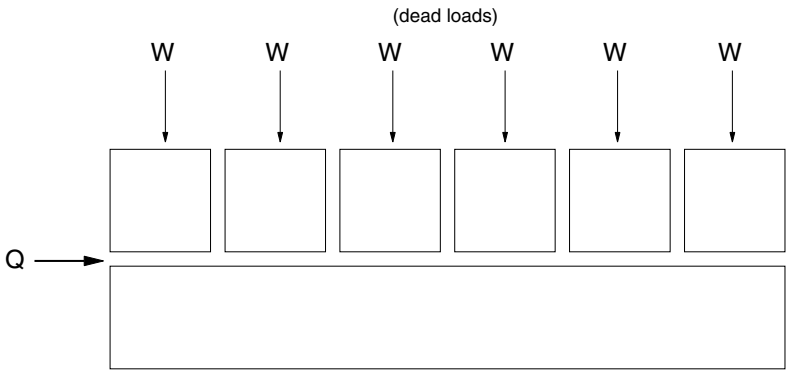

Figure 2.4 Extreme example in which the speed of propagation depends on system stiffness

The algorithm proceeds by performing a sequence of fluid steps, the timestep being defined by the user. For each step, a series of mechanical relaxation steps is performed in order to achieve continuity of flow at each domain. Given the assumption of fluid incompressibility, the net flow into a domain during a fluid step must equal the increment of domain volume. The unbalanced fluid volume, being the difference between the two, is gradually reduced during the relaxation procedure. For this purpose, the domain pressure is increased or reduced proportionally to the unbalanced volume for each domain. The proportionality factor is controlled by an adaptive scheme, and therefore varies during the iterations to provide better convergence.

2.2.4 Transient Flow of a Compressible Gas

Transient flow of a compressible gas through joints can also be simulated in *UDEC*. The flow algorithm is based upon the algorithm used to model compressible flow of a fluid (see Section 2.2.1), with the following modifications. Modeling the flow of a gas requires consideration of the strong density dependence on pressure, which can usually be neglected when dealing with slightly compressible fluids, as is done with the other flow modes in *UDEC*. For an ideal gas, we can consider the flow under isothermal conditions, for which

$$\rho_g = Bp \tag{2.11}$$

and adiabatic flow, for which

$$\rho_g = Bp^{1/\gamma} \tag{2.12}$$

where ρ_g is gas density, p is pressure, and B and γ are constants (see Chan et al. 1993).

As a consequence, the gas bulk modulus is also a function of pressure. In the isothermal case, we have, from Eq. (2.11), the bulk modulus given by

$$K_g = p \tag{2.13}$$

For the adiabatic case, gas bulk modulus is

$$K_g = \frac{p}{\gamma} \tag{2.14}$$

The dependence of density on pressure requires that the spatial variation of density be taken into account in the fluid mass balance

$$\frac{\partial(n\rho_g)}{\partial t} + \frac{\partial(\rho_g q_i)}{\partial x_i} = 0 \tag{2.15}$$

In contrast with the existing logic in *UDEC* for slightly compressible fluids, the analysis of gas flow requires the consideration of the convective terms, resulting from the density gradients in the second term of Eq. (2.15).

Currently, for water flow in *UDEC*, the domain pressures are updated at every step via Eq. (2.6), which can be written as

$$p_{new} = p_{old} + K_g \frac{1}{V} \sum_i q_i \Delta t - K_g \frac{\Delta V}{V} \tag{2.16}$$

where the summation extends over all the contacts that allow flow into or out of the domain.

For gas flow, the density varies from domain to domain. As flow rates are calculated at the contacts that separate the domains, it is necessary to assign a density to the flow in and out of each domain. Assuming that the fluid density at a contact is given by the average of the domain fluid densities, we have, at contact i,

$$\rho_{im} = \frac{\rho_{i1} + \rho_{i2}}{2} \tag{2.17}$$

where ρ_{i1} and ρ_{i2} are the densities of the two domains.

The mass balance of the flow in and out of the domain implies that the second term on the right-hand side of Eq. (2.16) needs to be adjusted by a factor equal to the ratio of the density ρ_{im} at the contact and the old domain density, ρ.

$$p_{new} = p_{old} + K_g \frac{1}{V} \sum_i q_i \frac{\rho_{im}}{\rho} \Delta t - K_g \frac{\Delta V}{V} \tag{2.18}$$

Density and bulk modulus for each domain are updated as a function of domain pressure, according to Eqs. (2.11) to (2.14).

2.2.5 Two-Phase Flow in Joints

Two-phase flow, fully coupled with deformation of solid blocks, can be simulated in *UDEC*. The flow is considered to take place in the joints only. The blocks are assumed to be completely impermeable. Both phases are active in this formulation (i.e., *UDEC* calculates pressure changes and flow for both wetting and non-wetting fluid). Capillary effects can be included. The capillary pressure curve and relative permeability of phases (as a function of saturation) have a predefined functional form, with parameters that can be set by the user. The two-phase logic is based upon the default, compressible transient flow logic, as described in Section 2.2.1.

The default flow logic in *UDEC* can simulate (in two dimensions) flow of a compressible fluid through fractures in a deformable rock mass. The code can analyze both confined and unconfined flows. Modeling of unconfined flow requires logic for determining the phreatic surface as a function of the saturation of rock joints by a wetting phase (e.g., the water is the wetting phase in a contact between the water and the air). Thus, a simple two-phase flow can be simulated with the default fluid-flow logic in *UDEC*. However, the assumption of this model is that the air (the non-wetting phase) is inactive: there is no movement of the air since the pressure in the air is uniform (under atmospheric conditions).

The two-phase flow logic allows for independent specification of general pressure and saturation boundary and initial conditions for each of the fluid phases. The algorithm does not rely on any assumptions about the number or location of interfaces between phases. The shape, the number and the connectivity of the interfaces between two phases are completely general. The length scale in the calculation is the length of the joint in the model. Rock joint roughness is not considered explicitly, but is taken into account by constitutive relations imbedded into the relations between (1) flow rates and pressure gradient, (2) the capillary pressure and the saturation and joint opening, and (3) the phase-relative permeability and the saturation.

The following equations govern transient two-phase flow of immiscible, compressible fluids in a network of one-dimensional joints in a deformable, discontinuous solid.

$$
\begin{aligned}
s_w + s_{nw} &= 1 \\
p_{nw} - p_w &= p_c(s_w, a) \\
q_w &= -\kappa_w(s_w)\frac{\partial p_w}{\partial x} \\
q_{nw} &= -\kappa_{nw}(s_{nw})\frac{\partial p_{nw}}{\partial x} \\
-\frac{\partial(\rho_w q_w)}{\partial x} &= \rho_w\left(\frac{\partial s_w}{\partial t} + \frac{s_w}{a}\frac{da}{dt} + \frac{s_w}{K_w}\frac{\partial p_w}{\partial t}\right) \\
-\frac{\partial(\rho_{nw} q_{nw})}{\partial x} &= \rho_{nw}\left(\frac{\partial s_{nw}}{\partial t} + \frac{s_{nw}}{a}\frac{da}{dt} + \frac{s_{nw}}{K_{nw}}\frac{\partial p_{nw}}{\partial t}\right)
\end{aligned}
\tag{2.19}
$$

where subscripts $_w$ and $_{nw}$ denote wetting phase (e.g., water) and non-wetting phase (e.g., gas), respectively, p is the pressure, p_c is the capillary pressure, s is the saturation, κ is the joint permeability, q is the flow rate, a is the joint aperture, and K is the bulk modulus. Note that the bulk modulus of both the wetting and non-wetting fluid is considered to be constant (i.e., independent of pressure). Both fluids are considered to be "slightly compressible" – the spatial gradients of the fluid density are neglected.

In the model defined by Eq. (2.19), the "interface" between two phases is a region of localized gradient of saturation. For the purpose of interpretation of model results, the interface can be defined as a set of points in the model plane where saturation is equal to 0.5, although the transition region (of localized saturation gradient) may be spread over a number of joint segments.

Flow of each phase is governed by the pressure differential between adjacent domains. As in the single-phase flow, the flow rate is calculated in two different ways, depending on the type of contact. For a point contact, the flow rates are given by

$$
\begin{aligned}
q_w &= -f_w(s_w)\, k_{wc}\, \Delta p_w \\
q_{nw} &= -f_{nw}(s_{nw})\, k_{nwc}\, \Delta p_{nw}
\end{aligned}
\tag{2.20}
$$

where $f_w(s)$ is a wetting fluid relative permeability factor;
$f_{nw}(s)$ is a non-wetting fluid relative permeability factor;
k_{wc} is a wetting fluid point contact permeability factor;
k_{nwc} is a non-wetting fluid point contact permeability factor; and

$$\begin{aligned} \Delta p_w &= p_{w2} - p_{w1} + \rho_w \, g \, (y_2 - y_1) \\ \Delta p_{nw} &= p_{nw2} - p_{nw1} + \rho_{nw} \, g \, (y_2 - y_1) \end{aligned} \tag{2.21}$$

where ρ_w is the wetting fluid density;
ρ_{nw} is the non-wetting fluid density;
p_{w1} and p_{w2} are wetting fluid pressures in the domains 1 and 2;
p_{nw1} and p_{nw2} are non-wetting fluid pressures in the domains 1 and 2;
g is the acceleration of gravity (assumed to act in the negative y-direction); and
y_1, y_2 are the y-coordinates of the domain centers.

The flow rates along the planar fracture are given by

$$\begin{aligned} q_w &= -f_w(s_w) \, k_{wj} \, a^3 \, \frac{\Delta p_w}{l} \\ q_{nw} &= -f_{nw} \, (s_{nw}) k_{nwj} \, a^3 \, \frac{\Delta p_{nw}}{l} \end{aligned} \tag{2.22}$$

where k_{wj} is a wetting fluid joint permeability factor whose theoretical value is $(1 \,/\, 12\mu_w)$;
μ_w is the dynamic viscosity of the wetting fluid;
k_{nwj} is a non-wetting fluid joint permeability factor with theoretical value $(1 \,/\, 12\mu_{nw})$;
μ_{nw} is the dynamic viscosity of the non-wetting fluid;
a is the contact hydraulic aperture; and
l is the length assigned to the contact between the domains.

The two factors f_w and f_{nw} account for the effect of reduction in conductivity of either phase in the case when a joint is partially saturated. The functional form of the two factors and their dependence on joint properties (e.g., joint roughness) is not well-known. It has been assumed in the current formulation that the two relative permeability factors depend on joint saturation only, as it is given by the relations

$$\begin{aligned} f_w &= s_w^2(3 - 2s_w) \\ f_{nw} &= s_{nw}^2(3 - 2s_{nw}) \end{aligned} \tag{2.23}$$

Capillary pressure curves (i.e., dependence of capillary pressure to saturation) is relatively well-known for soils. The standard laboratory experiments are used for measurements of capillary pressure curves. Various authors have proposed analytical expressions for describing general shapes of capillary pressure curves. The capillary pressure curves for rock joints have not been investigated as extensively as curves for soils. Pruess and Tsang (1990) analytically derived a relation between the average joint apertures, joint roughness, saturation and the capillary pressure. Unfortunately, that curve was not convenient for implementation in *UDEC*. The curve of the following form was implemented in *UDEC* as a capillary pressure curve.

$$p_c(s_w, a) = \gamma\ \rho_w\ g\ \frac{a_0}{a}\ \left(s_w^{\frac{\beta}{1-\beta}} - 1\right)^{1/\beta} \tag{2.24}$$

where a_0, β and γ are curve parameters.

a_0 is the reference hydraulic aperture (dimension of length) for which parameters γ and β are determined.
(The capillary pressure is inversely proportional to hydraulic aperture, a.);

β controls the shape of the curve (as illustrated in Figure 2.5).

γ has dimension of length and scales the capillary pressure curve with density of wetting fluid (i.e., defines the height of capillary fringe).

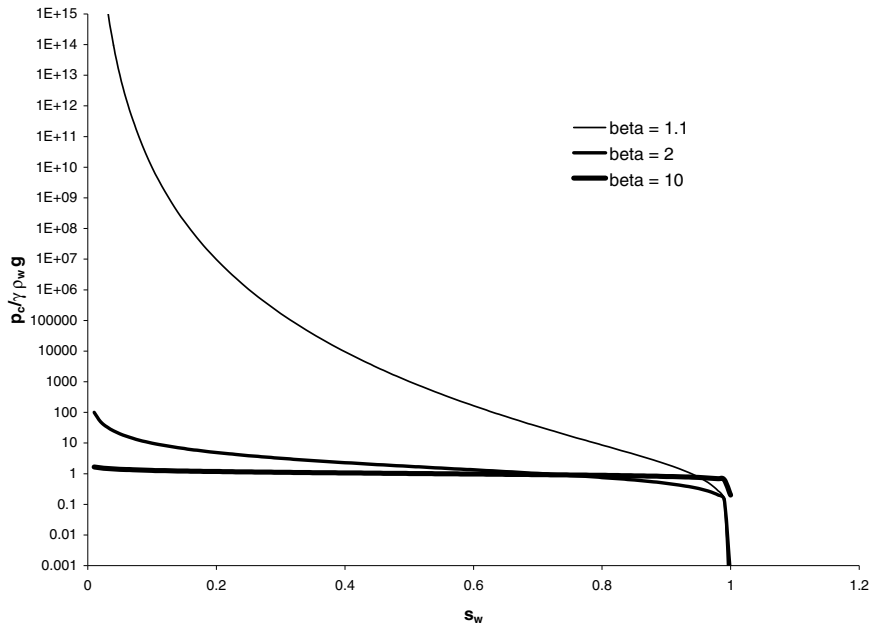

Figure 2.5 Capillary pressure curve as a function of saturation and parameter β

Eq. (2.24) has the same form as the van Genuchten curve, which is very often used for soils. The difference is that the curve defined by Eq. (2.24) takes into account the effect of hydraulic aperture. Note that the capillary pressure curve becomes singular for zero saturation. Simulation of the models with very small saturation would impose a very severe restriction on the critical calculation timestep. Therefore, the model has the lower bound on saturation defined by **block fluid two-phase saturation-minimum**. (The default value for **saturation-minimum** is 10^{-4}.) The domain values of a_0, β and γ are specified by the command **block domain property mat *n* capillary-alpha**, **capillary-beta** and **capillary-gamma**.

The domain wetting fluid pressure changes (due to flow only) can be expressed, after solving a system of four difference equations (obtained from Eq. (2.19)):

$$\Delta p_w = \frac{K_w\ K_{nw}\ (Q_w + Q_{nw}) - K_w\ S_{nw} \frac{\partial p_c}{\partial s_w} Q_w}{K_w\ s_{nw} + K_{nw}\ s_w - s_{nw}\ s_w\ \frac{\partial p_c}{\partial s_w}} \frac{\Delta t}{V} \tag{2.25}$$

The change in saturation is

$$\Delta s_w = \frac{K_w\ s_{nw}\ Q_w - K_{nw}\ S_w Q_{nw}}{K_w\ s_{nw} + K_{nw}\ s_w - s_{nw}\ s_w\ \frac{\partial p_c}{\partial s_w}} \frac{\Delta t}{V} \tag{2.26}$$

Similarly, the change in the domain wetting fluid pressure (due to joint deformation only) is

$$\Delta p_w = \frac{K_w \left[\Delta V (\frac{\partial p_c}{\partial s_w} s_w\ s_{nw} - K_{nw}) - s_{nw}\ \frac{\partial p_c}{\partial a} \Delta a\right]}{K_w\ s_{nw} + K_{nw}\ s_w - s_{nw}\ s_w\ \frac{\partial p_c}{\partial s_w}} \tag{2.27}$$

and the change in saturation is

$$\Delta s_w = \frac{s_w\ s_{nw} \left[-\Delta V (K_w - K_{nw}) + \frac{\partial p_c}{\partial a} \Delta a\right]}{K_w\ s_{nw} + K_{nw}\ s_w - s_{nw}\ s_w\ \frac{\partial p_c}{\partial s_w}} \tag{2.28}$$

Therefore, the pressures and saturation in the domain in the next timestep are

$$\begin{aligned} p_w &= p_{w0} + \Delta p_w \\ s_w &= s_{w0} + \Delta s_w \\ s_{nw} &= 1 - s_w \\ p_{nw} &= p_c(s_w) + p_w \end{aligned} \tag{2.29}$$

where
p_{w0} is the domain wetting fluid pressure in the current timestep;
p_w is the domain wetting fluid pressure in the next timestep;
s_{w0} is the domain saturation in the current timestep;
s_w is the domain saturation in the next timestep;
s_{nw} $= 1 - s_w$;
p_{nw} is the domain non-wetting fluid pressure in the next timestep;
Q_w is the sum of wetting fluid flow rates into the domain from all surrounding contacts;
Q_{nw} is the sum of non-wetting fluid flow rates into the domain from all surrounding contacts;
K_w is the bulk modulus of the wetting fluid;
K_{nw} is the bulk modulus of the non-wetting fluid;
ΔV $= V - V_o$ increment in domain volume between current and next timestep;
Δa $= a - a_o$ increment between the current and the next timestep in the average aperture of the joints that form the domain;
$\frac{\partial p_c}{\partial s_w}$ partial derivative of the capillary pressure curve with the respect to saturation; and
$\frac{\partial p_c}{\partial a}$ partial derivative of the capillary pressure curve with the respect to average aperture.

Given the new domain pressures, the average pressure exerted by the fluids on the edges of the surrounding blocks can be obtained:

$$p^{av} = p_w\, s_w + p_{nw}\, s_{nw}$$

If two-phase flow is on (**block fluid two-phase *on***), then pressures of both wetting and non-wetting fluid change as a function of domain volumetric deformation (as given by Eq. (2.27)).

Numerical stability of the present explicit fluid flow algorithm requires that the timestep be limited to

$$\Delta t_f = \min\left[\frac{V}{K_1 \sum_i k_{1i}}, \frac{V}{K_2 \sum_i k_{2i}} \right] \tag{2.30}$$

where V is domain volume, and the summation of permeability factors is extended to all contacts surrounding the domain. The symbols in Eq. (2.30) are defined:

$$
\begin{aligned}
k_{1i} &= \max(k_{wc}, k_{nwc}, k_{wj}\ a^3\ /\ l, k_{nwj}\ a^3\ /\ l) \\
K_1 &= \max(K_w, K_{nw}) \\
k_{2i} &= \max(k_{wc}, k_{wj}\ a^3\ /\ l) \\
K_2 &= \max(K_w, \frac{\partial p_c}{\partial s_w})
\end{aligned}
\tag{2.31}
$$

The minimum value of Δt over all domains is used in the analysis.

For transient flow analysis, the numerical stability requirements may be rather severe and may make some analyses very time-consuming or impractical, especially in cases when wetting fluid with large bulk modulus and non-wetting fluid with small viscosity (i.e., large conductivity of non-wetting fluid) are both in the same model.

2.2.6 Modifying Joint Flow Rate and Simulating Viscoplastic Flow in Joints

The flow of a Bingham body (or liquid), such as cement grout, is of the viscoplastic type. The major difference between this model and that for a Newtonian liquid is that, for a Bingham fluid, a yield stress, τ_y, must be exceeded to initiate flow.

For Newtonian flow, it is assumed that the flow rate per unit width, q, is related linearly to the pressure gradient, J, as shown in Figure 2.6. The general equation for fluid flow between planar surfaces is given by

$$
q = \frac{ba^x\ J}{12\mu} \tag{2.32}
$$

where a = fracture width (aperture);
b = empirical coefficient;
μ = dynamic viscosity of fluid; and
x = aperture exponent.

In the most widely used form of this relation, known as the cubic flow law, $x = 3$ and $b = 1$ (see Eq. (2.3)). The values for x and b can be specified by the user with the **block contact property**keywords **exponent** and **constant-b**, respectively.

The flow gradient relation of a Bingham body is similar to this expression, except that no flow occurs until the threshold gradient, J_0, is exceeded, as shown in Figure 2.7.

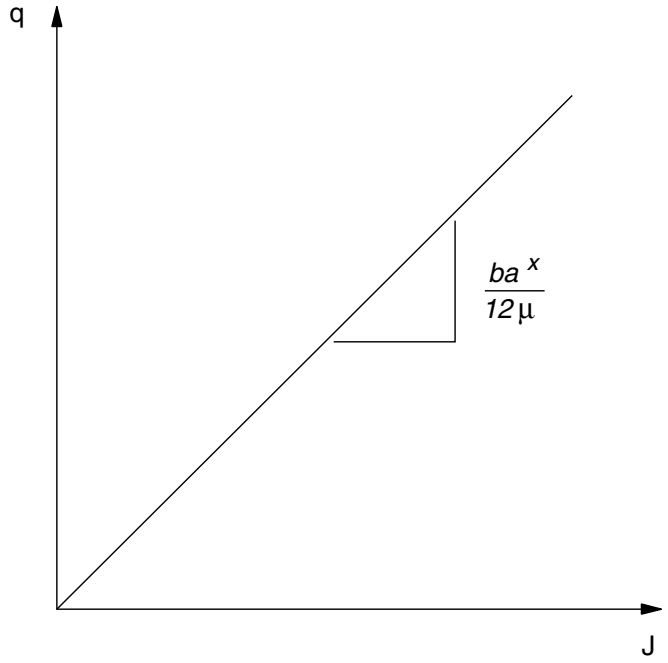

Figure 2.6 Flow-gradient relation for Newtonian fluid in UDEC

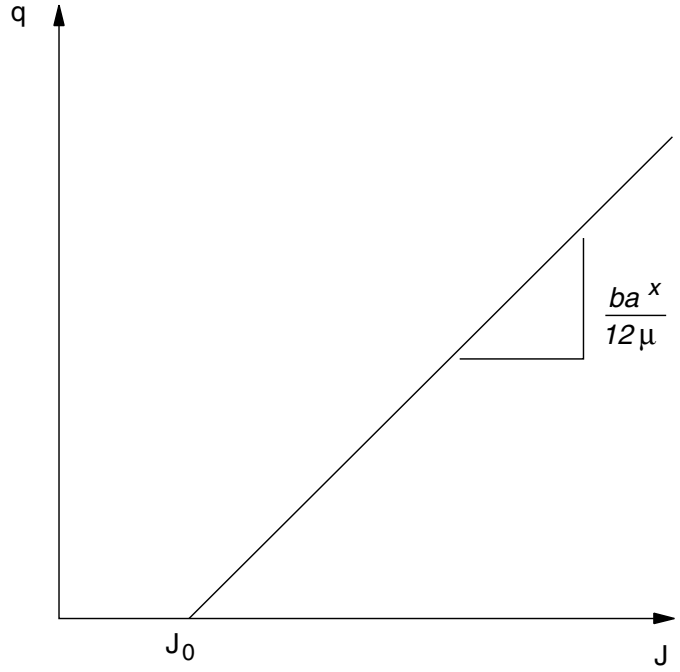

Figure 2.7 Flow-gradient relation for Bingham fluid in UDEC

Considering the balance of forces on a rectangular element of fluid, the expression for the threshold gradient for flow between parallel sides of aperture, a, is given by

$$J = \frac{2\,\tau_y}{a} \tag{2.33}$$

where τ_y is the yield stress.

The expression for the threshold gradient can also be obtained by considering the equation for steady laminar flow of a Bingham plastic in a circular pipe. This equation is known as Buckingham's equation (Wilkinson 1960):

$$Q = \frac{\pi\, r^4\, \Delta P}{8L\, \mu_p} \left[1 - \frac{4}{3} \left[\frac{2L\, \tau_y}{r\, \Delta P} \right] + \frac{1}{3} \left[\frac{2L\, \tau_y}{r\, \Delta P} \right]^4 \right] \tag{2.34}$$

where Q = volume rate of flow;

r = pipe radius;

μ_p = Bingham plastic viscosity; and

$\Delta P / L$ = pressure gradient = J.

From this expression, it can be seen that no flow occurs if the pressure gradient, J, is zero or equals $2\,\tau_y / r$. It is not clear from the equation what occurs at pressure gradients between zero and $2\,\tau_y / r$, but it is reasonable to assume that no steady flow occurs within this range. Therefore, the threshold gradient, J_0, the gradient at which steady flow is possible, is given by

$$J_0 = \frac{2\,\tau_y}{r} \tag{2.35}$$

Note that this expression can also be derived by considering the balance of forces acting on a cylindrical element of fluid with radius r and length L.

For an example application of this feature, see Section 5 in the **Example Applications**. This example applies the viscoplastic flow model to simulate cement grouting. The threshold pressure gradient, J_0, is specified with the **block fluid property *cohesion*** command.

2.2.7 Fluid Boundary Logic

A porous medium may be "wrapped" around the *UDEC* block model in order to simulate fluid flow on a larger scale while keeping the number of blocks reasonable. A radial mesh is created around the outer boundary of *UDEC*. The outer boundary of the mesh is circular and has imposed fluid pressures. The inner boundary is the outer boundary of the *UDEC* block model. The variation of pressures across this inner boundary is continuous, and flow to or from a joint is accounted for in the grid zone next to it in order to satisfy the fluid-mass balance. An example mesh is given in Figure 2.8:

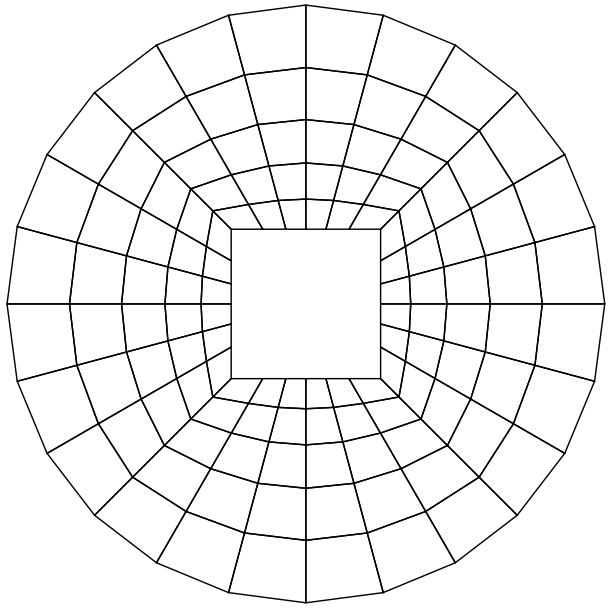

Figure 2.8 Porous medium mesh

The formulation is described below. The logic, at present, is only intended for steady-state flow; a relaxation factor is used for convergence. The formulation assumes small-strain conditions and confined flow; a free surface cannot be modeled.

Darcy's law for an anisotropic porous medium is

$$V_i = K_{ij} \frac{\partial p}{\partial x_j}$$

where V_i is the specific discharge vector, p is the pressure and K_{ij} is the permeability tensor. Each quadrilateral element in the porous medium mesh is divided into triangles in two different ways (see Figure 2.9(a)). The specific discharge vector can be derived for the generic triangle of Figure 2.9(b). By Gauss' theorem,

$$\frac{\partial p}{\partial x_i} = \frac{1}{A}\int_s pn_i\, ds \tag{2.36}$$

Hence,

$$V_i \simeq \frac{K_{ij}}{A}\sum pn_j s \tag{2.37}$$

where $\sum$ is the summation over the three sides of the triangle. For the x-component of V_i,

$$V_1 = \frac{1}{A}\left[K_{11}\sum pn_1 s + K_{12}\sum pn_2 s\right] \tag{2.38}$$

The contribution of side (ab) of the triangle to the summation is

$$V_1^{(ab)} = \frac{1}{2A}\left[-K_{11}(p^{(b)} + p^{(a)})(x_2^{(b)} - x_2^{(a)}) + K_{12}(p^{(b)} + p^{(a)})(x_1^{(b)} - x_1^{(a)})\right] \tag{2.39}$$

Similarly,

$$V_2^{(ab)} = \frac{1}{2A}\left[-K_{21}(p^{(b)} + p^{(a)})(x_2^{(b)} - x_2^{(a)}) + K_{22}(p^{(b)} + p^{(a)})(x_1^{(b)} - x_1^{(a)})\right] \tag{2.40}$$

The other two sides, (bc) and (ca), provide similar contributions to V_i. This specific discharge vector is then converted to scalar volumetric flow rates at the nodes by making dot products with the normals to the three sides of the triangle. The general expression is

$$q = \frac{V_i n_i s}{2} \tag{2.41}$$

where the factor of 2 accounts for the fact that we take the *average* of the contribution from the two triangle-pairs that make up the quadrilateral element. The flow rate into node (a) in terms of coordinates is then

$$q^{(a)} = \frac{1}{2}\left\{-V_1(x_2^{(b)} - x_2^{(c)}) + V_2(x_1^{(b)} - x_1^{(c)})\right\} \tag{2.42}$$

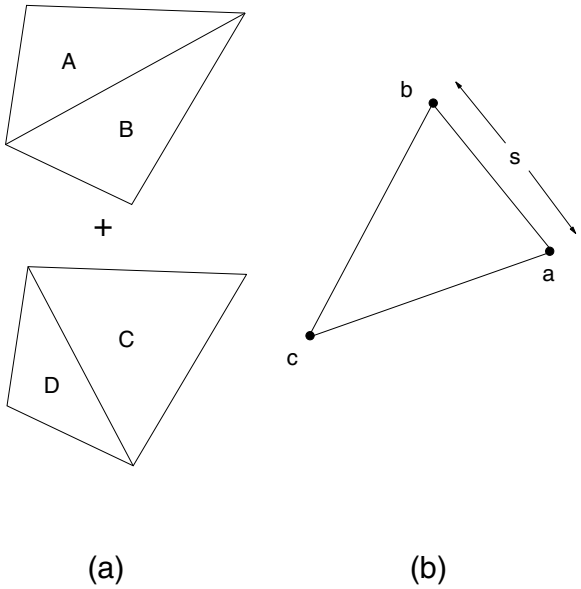

Figure 2.9 Elements in porous medium mesh

Similar expressions apply to nodes (b) and (c). Nodal flow rates are added from the other three triangles shown in Figure 2.9(a). The stiffness matrix $[K]$ of the whole quadrilateral element is defined in terms of the relation between the pressures at the four nodes and the four nodal flow-rates, as derived above:

$$\{q\} = [K]\{p\} \tag{2.43}$$

For elements along the inner boundary, contributions from the corresponding joints are added to these flow rates. An example application of the fluid boundary is given in Section 2.5.2.

2.2.8 One-Way Thermal-Hydraulic Coupling

Temperature variations induce changes in the viscosity and density of water. Because water viscosity governs the conductivity of joints, and water density governs the magnitude of hydraulic pressure gradients, these effects should be taken into account when flow is simulated in a nonhomogeneous or time-varying temperature field. Note that a full thermal-hydraulic coupling would require the fluid flow to influence heat transfer. This behavior is not included in this version of *UDEC*.* The thermal logic assumes that the rock mass is continuous, and does not account for water flow in the joints. An example application of the thermal coupling is given in Section 2.5.11.

---

* Other researchers have adapted *UDEC* for simulation of coupled thermal convection of fluid in fractures (e.g., see Abdallah et al. 1995).

2.3 Hydraulic Behavior of Rock Joints

2.3.1 Parallel Plate Model

Flow in planar rock fractures may be idealized by means of the parallel plate model. The analytic solution for laminar viscous flow between parallel plates gives the mean velocity as

$$v = k_f \, J \tag{2.44}$$

where J is the hydraulic gradient, and the fracture hydraulic conductivity is given by

$$k_f = \frac{a^2 \, g}{12\nu} \tag{2.45}$$

where a is the fracture width;
ν is the kinematic viscosity of the fluid; and
g is the acceleration of gravity.

The flow rate per unit width can thus be expressed as

$$q = v \, a = \frac{a^3 \, g}{12\nu} \, J \tag{2.46}$$

which is usually referred as the cubic flow law.

Since pressure (p) is equal to $g\rho_w h$ (where h is the head), and dynamic viscosity (μ) is equal to $\nu\rho_w$, then Eq. (2.44) can also be written as

$$q = -\frac{a^3}{12\mu} \, \frac{\Delta p}{l}$$

(i.e., see Eq. (2.3)).

Experiments conducted by Louis (1969) showed that this law is essentially valid for laminar flow in rock joints. This author proposed an empirical correction factor for the above expression in order to account for fracture roughness. Witherspoon et al. (1980) tested both open and closed joints. They reported that the cubic law is still valid for the latter, provided that the actual mechanical aperture is used. Due to the effects of roughness and tortuosity of flow, the fracture conductivity in their experiments was reduced by a factor between 1.04 and 1.65. Barton et al. (1985) proposed an empirical formula that gives the hydraulic aperture (to be used in the cubic law) as a function of the mechanical aperture and the joint roughness coefficient (JRC).

2.3.2 Fluid Flow Properties and Units

Typical SI units for the various parameters described in Sections 2.2 and 2.3 are given in Table 2.1:

Table 2.1 Typical SI units for fluid flow parameters

Parameter	SI Units
q	$m^3/sec \cdot m$
p	Pa
k_c	$m^2/sec \cdot$ Pa
k_j	$1/(Pa \cdot sec)$
a	m
l	m
Q	m^3/sec
K_w	Pa

An example calculation is given for fluid flow parameters to use in a *UDEC* model. The following discussion relates primarily to the gravity dam example problem presented in Section 4 in the **Example Applications**, but it can serve as a starting point for other problems as well. In the example, joint hydraulic apertures were calculated from Eq. (2.5).

A residual aperture, a_{res}, was taken as the minimum hydraulic aperture. A maximum value for the hydraulic aperture equal to twice the residual aperture was assumed for reasons of computational efficiency, because the timestep required for stability of the fluid flow algorithm is inversely proportional to joint conductivity. As conductivity is proportional to the cube of the aperture, considerable variation of permeability due to stress changes can still be modeled despite this constraint.

In the base run, the values a_0 = 1 mm and a_{res} = 0.5 mm were used. The water properties used were ρ_w = 1000 kg/m^3 and $\mu = 1.0 \times 10^{-3}$ Pa $\cdot$ sec. (This problem is a steady-state flow analysis, so K_w is not required as input, but is automatically defined, as discussed in Section 2.2.2.)

The dynamic viscosity of water ($\mu = 10^{-3}$ Pa·sec) was used to calculate the joint permeability factor in *UDEC* (see Eq. (2.3)):

$$k_j = \frac{1}{12\mu} = 83.3 \text{ Pa}^{-1} \text{ sec}^{-1} \tag{2.47}$$

See the data file for the gravity dam example at the end of Section 4 in the **Example Applications**.

2.4 Calculation Modes and Commands for Fluid-Flow Analysis

A summary of the different fluid-flow calculation modes and a brief review of commands related to fluid flow analysis are presented in this section. The commands are summarized by function in Table 2.2:

Table 2.2 Summary of fluid flow commands

	Compressible Liquid, Transient Flow	Compressible Liquid, Steady-State Flow	Incompressible Liquid, Transient Flow	Compressible Gas, Transient Flow	Two-Phase Fluid, Transient Flow
set flow mode[1]	**BL FLUID compressible**	**BL FLUID steady-state**	**BL FLUID incompressible**	**BL FLUID gasflow**	**BL FLUID twophase**
fluid properties	**BL FL PRO bulk** **cohesion** [4] **density** [2] **density-table** [3] **permeability-table**[3]	**BL FL PRO cohesion** [4] **density** [2] **density-table** [3] **permeability-table**[3]	**BL FL PRO density** [2] **density-table** [3] **permeability-table**[3]	**BL FL PRO bulk density** [2] **alpha** **constant** **bulk-minimum** **density-minimum** **permeability-table**[3]	**BL FL PRO bulk** **cohesion 4** **density** [2] **density-table** [3] **nonwetting-density** **nonwetting-bulk** **permeability-table**[3]
joint flow properties properties	**BL CO PRO perm-factor** **aper-zero-load** **aper-res** **expo**[4] **const-b**[4] **BL FLUID aperture** **aperture-minimum** **aperture-max-ratio** **crack-flow**	**BL CO PRO perm-factor** **aper-zero-load** **aper-res** **expo**[4] **const-b**[4] **BL FLUID aperture** **aperture-minimum** **aperture-max-ratio** **crack-flow**	**BL CO PRO perm-factor** **aper-zero-load** **aper-res** **expo**[4] **const-b**[4] **BL FLUID aperture** **aperture-minimum** **aperture-max-ratio** **crack-flow**	**BL CO PRO perm-factor** **aper-zero-load** **aper-res** **expo**[4] **const-b**[4] **BL FLUID aperture** **aperture-minimum** **aperture-max-ratio** **crack-flow**	**BL CO PRO perm-factor** **aper-zero-load** **aper-res** **expo**[4] **const-b**[4] **nonwet-perm-factor** **BL FLUID aperture** **aperture-minimum** **aperture-max-ratio** **crack-flow** **satmin sat-max**
initialize fluid pressure[5]	**BL INSITU p-p p-grad-x p-grad-y** **water-table** **BL DO IN density** **grad-x grad-y** **history** **nonw-gr-x nonw-gr-y** **nonw-p** **pore-p** **saturation seepage**	**BL INSITU p-p p-grad-x p-grad-y** **water-table** **BL DO IN density** **grad-x grad-y** **history** **nonw-gr-x nonw-gr-y** **nonw-p** **pore-p** **saturation seepage**	**BL INSITU p-p p-grad-x p-grad-y** **water-table** **BL DO IN density** **grad-x grad-y** **history** **nonw-gr-x nonw-gr-y** **nonw-p** **pore-p** **saturation seepage**	**BL INSITU p-p p-grad-x p-grad-y** **water-table** **BL DO IN density** **grad-x grad-y** **history** **nonw-gr-x nonw-gr-y** **nonw-p** **pore-p** **saturation seepage**	**BL INSITU p-p p-grad-x p-grad-y** **water-table** **BL DO IN density** **grad-x grad-y** **history** **nonw-gr-x nonw-gr-y** **nonw-p** **pore-p** **saturation seepage**
initialize fluid flow time	**BL FL CO time**		**BL FL IN time**	**BL FL GAS time**	**BL FL T-P time**
specify fluid flow boundary conditions[5]	**BL ED AP p-p p-grad-X p-grad-Y** **impermeable**	**BL ED AP p-p p-grad-X p-grad-Y** **impermeable**	**BL ED AP p-p p-grad-X p-grad-Y** **impermeable**	**BL ED AP p-p p-grad-X p-grad-Y** **impermeable**	**BL ED AP p-p p-grad-X p-grad-Y** **impermeable**
flow-only solution	**BL MECH active off** **CYCLE *n*** [6]	**BL MECH active off** **CYCLE *n*** [6]		**BL MECH active off** **CYCLE *n*** [6]	**BL MECH active off** **CYCLE *n*** [6]
coupled mechanical-flow solution	**BL FL CO subst-mech subst-flow** **BL MECH timestep-factor** **BL CYCLE *n***	**BL CYCLE *n***	**BL FL IN tolerance-volume** **BL FL IN timestep-factor** **BL CYCLE flow-time *n***	**BL FL CO subst-mech subst-flow** **BL MECH timestep-factor** **BL CYCLE *n***	**BL FL CO subst-mech subst-flow** **BL MECH timestep-factor** **BL CYCLE *n***

[1] Fluid flow can be turned on and off by using the commands block fluid flow=on **block fluid flow=off**
[2] Use only if gravity is acting.
[3] Use only for temperature dependency.
[4] Use only for visco-plastic flow
[5] Mechanical-stress initial and boundary conditions must be specified separately from fluid flow conditions
[6] *n* =fluid flow steps
[7] *n* =calculation steps to reach steady-state flow

2.4.1 Selection of Calculation Mode

Five calculation modes are available in *UDEC* for different types of fluid-flow problems. The fluid flow modes are accessed by first setting the fluid-flow configuration in *UDEC* by specifying the **block config fluid** command.

(1) Transient Flow Analysis – Compressible Fluid (**block fluid compressible**)

This option is selected with the **block fluid compressible** command and is the basic formulation, as described in Section 2.2.1. The fluid bulk modulus given in the **block fluid property** command is used in the calculation of the transient response of the system. The explicit nature of the algorithm requires the fluid timestep to be limited to ensure numerical stability. For typical ranges of properties, the timesteps are very small in relation to the flow time scales, and this algorithm is only practical for simple problems or short flow times.

The main application of compressible fluid analysis is in the modeling of the pressure response of fluid-filled joints to a short-duration dynamic loading (e.g., an earthquake). For these short time scales, actual flow does not change significantly, and the flow calculation may be turned off. However, if the fluid bulk modulus is defined, the fluid pressure will still vary due to joint deformation, and may induce slip or separation. Note that the apparent stiffness of the fluid (given by K_w/a, the ratio of the fluid bulk modulus to the joint aperture) will reduce the mechanical timestep, so very small residual apertures may lead to large runtimes.

An optional method for calculating the transient compressible fluid-flow timestep has been added. The default method is to base the timestep on the lowest domain area and the highest actual contact aperture. For most models, this is overly conservative. The new option invoked by **block fluid timestep-mode = 3** changes the calculation to consider the domain area and the aperture for each contact. This usually results in a much higher fluid timestep.

(2) Steady-State Flow Analysis (**block fluid steady-state**)

In many applications, only the steady-state regime is of interest. For these cases, a special algorithm that provides a fast convergence to the steady-state solution (see Section 2.2.2) is available in *UDEC*. This option is selected with the **block fluid steady-state** command. The algorithm is based on the scaling of the fluid bulk modulus and the domain volumes, which are unimportant in the steady-state regime. When this option is used, the fluid flow timestep is arbitrarily set equal to the mechanical timestep, and the fluid bulk modulus is automatically defined. The flow time scale has no physical significance, and the same applies to the pressure or flow rate histories before steady state is reached.

(3) Transient Flow Analysis – Incompressible Fluid (**block fluid incompressible**)

An alternative algorithm has been implemented in *UDEC*, to make possible transient flow analysis for larger time scales than the basic algorithm allows (see Section 2.2.3). The fluid is assumed to be incompressible, and this scheme uses an iterative procedure

within each flow timestep to adjust the joint pressures and domain volumes to ensure flow continuity.

To use this option, the user must select **block fluid incompressible** and define the timestep with **block fluid incompressible timestep** ***value***. The convergence of the relaxation process in each fluid step is governed by two criteria: (1) the maximum ratio of unbalanced fluid volume to domain volume, set with **block fluid incompressible tolerance-volume** ***value*** (default = 0.001); and (2) the maximum number of mechanical iterations, defined with **block fluid incompressible substep-mechanical** ***n*** (default = 500). To check convergence, **block list max** prints out the current maximum unbalanced volume, as well as the actual number of iterations performed in the latest fluid step; **block fluid history unbalanced-volume** will record this variable. Based on these indicators, the user may adjust the convergence criteria for a specific problem.

The purpose of this algorithm is not to simulate dynamic flow or short transients, but the mechanical-fluid coupled response of a system undergoing a quasi-static process. Therefore, at each fluid step, an equilibrium must be possible. Otherwise, the mechanical iteration procedure may not converge. This scheme is not suitable for problems involving extensive failure, or when blocks may become detached. For good performance, sudden loads should not be applied to the system. For example, a cavern should not be pressurized or depressurized instantly, but a finite time should be allowed to elapse. This can be simulated by specifying a history in the **block domain fix** command. For example, domain number 100 can be assigned a pressure of 1 MPa applied over a finite time, as specified by table number 1:

```
block domain fix range domain 100
block domain init pore-pressure 1e6 hist table 1 ...
  range domain 100
```

where a **table** is used to define a simple history.

The incompressible fluid algorithm is applicable for either confined or unconfined flow conditions.

(4) Gas Flow Analysis (**block fluid gas-flow**)

The basic transient compressible flow logic is modified to simulate the special case of highly compressible gas. This logic can be accessed by specifying **block fluid gas-flow**. Gas density is now dependent upon the pressure in the domain. Gas is considerably more compressible than liquid and, consequently, pressures and stress distributions around gas-filled fractures can be significantly different than those around liquid-filled fractures.

(5) Two-Phase Flow Analysis (**block fluid two-phase**)

The basic transient compressible flow algorithm has also been adapted to simulate two-phase fluid flow in fractures. This logic is accessed via the **block fluid two-phase** command. This logic permits the independent specification of pressure and saturation conditions for each fluid phase.

The flow calculation can be switched on and off with **block fluid flow on** and **block fluid flow off**. Note that for all of the compressible fluid modes (i.e., for all options except the **incompressible** mode), fluid pressures will still change due to joint deformation. Otherwise, they remain constant. Also, flow-only analysis can be performed with all of the compressible fluid modes by selecting **block mechanical active off**, in which case joint apertures will be unchanged. **incompressible** flow requires mechanical coupling.

2.4.2 Properties

Joint Flow Properties – Joint conductivity parameters are given with the **block contact property** command. Their physical meaning is discussed in Section 2.3. The joint permeability factor is assigned by the **permeability-factor** keyword, the residual aperture by **aperture-residual**, and the nominal aperture at zero normal stress by **aperture-zero-load**. These properties are required for all five flow calculation modes. In order to modify the permeability relation, the exponent of the cubic law may be changed with **exponent** (default is 3), and the coefficient with **constant-b** (default is 1). For two-phase flow in joints, the non-wetting fluid joint permeability is also needed, and is specified with the **nonwetting-permeability-factor** keyword. For joints, flow through unfractured joints can be turned off with the **block fluid crack-flow *on*** command.

In explicit algorithms, large variations of permeabilities throughout the system generally reduce the numerical efficiency. Joint conductivity is strongly dependent on aperture, due to the cubic exponent. Therefore, for efficiency, a limit on the maximum aperture should be set. The **block fluid aperture-max-ratio** command is used to define the maximum aperture ratio allowed (i.e., the maximum hydraulic aperture equals **aperture-max-ratio** $\times$ **aperture-residual**). The default value is 5. A large **aperture-max-ratio** increases the number of cycles necessary to reach steady state (for **steady-state** mode) or reduces the fluid timestep (for **compressible**, **gas-flow** or **two-phase** mode).

Fluid Properties – The fluid properties are given with the **block fluid property** command. Fluid **density** is only required in problems with gravity, to obtain the hydrostatic pressure component. The **bulk** keyword is used for the **compressible**, **gas-flow** and **two-phase** flow modes. For two-phase flow, the density and bulk modulus of the non-wetting fluid are set with the **nonwetting-density** and **nonwetting-bulk** keywords, respectively. A threshold fluid gradient to initiate viscoplastic flow can be specified with the **cohesion** keyword.

Gas properties are specified in **gas-flow** mode to calculate gas density, ρ_g, and bulk modulus, K_g. The relation for gas density is

$$\rho_g = Bp^{\alpha} \tag{2.48}$$

and for gas bulk modulus is

$$K_g = p\alpha \tag{2.49}$$

Note that α is equal to $1/\gamma$ (compare Eq. (2.48) to Eq. (2.12) and Eq. (2.49) to Eq. (2.14)). The command **block fluid gas-flow alpha** is used to set α, and the **constant** keyword is used to set B. In addition, minimum values of bulk modulus, **bulk-minimum**, and density, **density-minimum**, are set to avoid numerical problems if the gas pressure drops to zero.

Domain Properties – A minimum hydraulic aperture is set with the **block flow aperture-minimum** command to calculate the minimum domain volume for transient flow analyses. (Note that this does not affect the joint hydraulic conductivity.) If **aperture-minimum** is not specified, it is set to the minimum value for **aperture-residual**.

For **two-phase** flow, the capillary pressure curve (Eq. (2.24)) parameters a_0, β and γ are specified with the **block domain property mat *n*** keywords **capillary-alpha**, **capillary-beta** and **cappillary-gamma**, respectively.

Also, for two-phase flow, a lower bound is specified for saturation of a domain by the wetting fluid, in order to increase the calculation efficiency. This is defined with the **block fluid two-phase saturation-minimum** command. By default, the domain is considered to be fully saturated when the wetting fluid saturation equals one. The value for full saturation can be reduced via the **block fluid two-phase saturation-maximum** command.

Temperature Dependency – Temperature-dependent fluid density and joint permeability can be specified with the **block fluid property density-table** and **block fluid property permeability-table** commands, respectively, to simulate one-way coupling of thermal and fluid flow processes. The table corresponding to **density-table** relates water density to temperature. The table corresponding to **permeability-table** relates the joint permeability factor to temperature. The **table** command is used to specify the temperature dependency. If the temperature is not within the range defined by a table, the value for the closest temperature is used (i.e., properties are constant outside the temperature range). The tables apply to *all* joint permeabilities, regardless of joint material number.

2.4.3 Boundary Conditions

Boundary conditions may be applied in terms of fluid pressures (**block edge apply pore-pressure, pressure-gradient-x, pressure-gradient-y**), or by defining an impervious boundary (**block edge apply impermeable**). Note that these keywords only supply flow boundary conditions. The mechanical pressure of the fluid in the outer domain must be given independently (with a **block edge apply stress** command, for example).

Outer boundary stresses are assumed to be total stresses. Therefore, the example in-situ stress state described in Section 2.4.4 is equilibrated by the boundary stresses

```
block edge apply stress -0.25 0 -0.25 gradient-y 0.025 0 0.025
```

Time-varying fluid pressure can be prescribed for specific boundaries with the **history** keyword to the **block edge apply** command. This is done in the same way as discussed for the **block domain fix** command, described in Section 2.4.4.

Either mechanical or fluid-flow time may be defined as reference time when applying time-varying boundaries with the **block edge apply history-time-reference mechanical** or **block edge appply history-time-reference flow** command, respectively. The default is **mechanical**. When applying histories, it is sometimes useful to change the initial time reference. This can be done with **block mechanical time-total** (mechanical time) or **block fluid compressible time** (flow time). This has no effect on the analysis.

For two-phase flow, the non-wetting boundary conditions may be applied in terms of non-wetting fluid pressures (**block edge apply nonwetting-pressure, nonwetting-gradient-x, nonwetting-gradient-y**), or by defining an impervious boundary (**block edge apply nonwetting-impermeable**).

Fluid saturation can be assigned at a boundary (**block edge apply saturation, saturation-gradient-x, saturation-gradient-y**), and a seepage boundary can be set with **block edge apply seepage**. In the latter case, wetting and non-wetting fluid pressures are fixed and equal, and an unsaturated boundary is impermeable for the wetting fluid.

A porous medium grid can also be created around the outer boundary of the *UDEC* model with the **block fluid far-field** command. The first mention of **block fluid far-field** causes the fluid boundary grid to be created. The fluid boundary must be created for a single convex block (i.e., before any joints are created). The number of fluid boundary zones in a circumferential direction is equal to the number of sides between corners on the single block. (Use the **block create corner-angle** command to increase the number of corners on the single block.) An example application of **block fluid far-field** is given in Section 2.5.2.

The following keywords define the conditions of the porous medium mesh. Once specified, the conditions cannot be changed later. The number of elements radially around the outer boundary is set by the **mesh-elements** keyword (default **mesh-elements** = 2). For Figure 2.8, **mesh-elements** = 5. The three components of the permeability tensor, K_{ij}, are given by the **permebility-xx**, **permeability-xy** and **permeability-yy** keywords. The fluid pressure at the outer boundary of the mesh is specified with **pore-pressure** (p_o). Nodes at the outer boundary of the mesh retain this fixed pressure. A pressure gradient can also be specified with **gradient-x** (pd_x) and **gradient-y** (pd_y) keywords. The pressures at nodes within the mesh are then computed from the equation

$$p = p_o + pd_x \cdot x + pd_y \cdot y \tag{2.50}$$

where ***x*** and ***y*** are the coordinates of the node. The radius of the mesh, with origin at the centroid of the block, is set by **mesh-radius**, and the ratio between radial size of adjacent elements is set by **mesh-ratio** (default **mesh-ratio** = 1).

block fluid far-field is only applicable for steady-state flow. Thus, the **block fluid steady-state** command must be given to perform a steady-state analysis.

A **block edge apply** command must be given before cycling, because this serves to transfer information between the flow calculations in the rock joints and the flow calculation in the porous medium. For example, the **block edge apply pore-pressure = 0** command will establish the boundary linked list.

2.4.4 Initialization of Fluid Pressures in Domains

The *UDEC* fluid flow logic is based on the assumption that blocks are impermeable. Block stresses are therefore total stresses, while joint stresses are effective stresses. In order to create a balanced in-situ state, the initialization of block/joint stresses and domain pressures must be done consistently. The **block insitu** command is used for this purpose. The user may define a domain pressure distribution with the **pore-pressure**, **pressure-gradient-y**, **pressure-gradient-y** and **water-table** keywords. Total stresses are specified with the **stress**, **gradient-x** and **gradient-y** keywords; these are assigned to the blocks. Joint stresses are calculated by adding the given domain pressures to the

block stresses. Note that block compressive stresses are negative, while domain pressures and joint normal compressive stresses are positive.

For example, consider the model conditions defined by the commands

```
block zone property dens 0.0025
block fluid property dens 0.001
block mechanical gravity  0 -10
```

Assume that the initial water table is located at the free surface at $y = 10$. A balanced in-situ state is created by the command

```
block insitu  stress -0.25 0 -0.25  grad-y 0.025 0 0.025 ...
        pore-p 0.1  pr-grad-y -0.01
```

Alternatively, the same state may be obtained by

```
block insitu  stress -0.25 0 -0.25  grad-y 0.025 0 0.025 ...
        water-table 10
```

The joint stresses will be calculated automatically to balance the block stresses and domain pressures.

The **block domain** command also allows the input of domain (wetting and non-wetting) pressures. However, in general, these will not be consistent with the existing block or joint stresses, so the **INSITU** command is preferred in the establishment of an initial in-situ state. The **block domain fix** command is normally used to maintain given domain pressures at a fixed value, or to prescribe a pressure history. (The **block domain well** command can also be used to assign fixed flow rates to given domains.) A time-varying pressure may be specified with the **block domain fix** command by using the **initialize history** keyword. A seepage condition can be specified on the boundaries of a domain using **block domain initialize seepage**. In this latter case, wetting and non-wetting fluid pressures are fixed and equal, and wetting fluid can inflow into the domain only from saturated domains.

2.4.5 Initialization of Pore Pressures in Blocks

UDEC does not model fluid flow through blocks. However, steady-state pore pressures can be assigned to zones within deformable blocks. This is accomplished by adding the **zone-pressure** keyword to the end of the **block insitu** command. By adding the **zone-pressure** keyword, poroelastic or poroplastic deformations associated with a new distribution of pore pressures within deformable blocks will be calculated. When this keyword is given with the **block insitu** command, the pore pressure change multiplied by the Biot coefficient is automatically subtracted from the total normal stresses in the affected zones.*

---

* Pore pressures can also be added to zones by using the **block zone initial pore-pressure** command. However, the adjustment to total stress is *not* performed automatically if the **block zone initial pore-pressure** command is used to add or change zone pore pressures.

The Biot coefficient, α, relates the compressibility of the grains to that of the drained bulk material:

$$\alpha = 1 - \frac{K}{K_s} \tag{2.51}$$

where K is the drained bulk modulus of the matrix, and K_s is the bulk modulus of the grains (see Detournay and Cheng 1993, for reference). The Biot coefficient is assigned with the **block property mat n biot-c** command.

For soils, matrix compliance is usually much higher than grain compliance (i.e., $1/K >>> 1/K_s$), and it is a valid approximation to assume that the Biot coefficient is equal to 1.

For porous rocks, however, matrix and rock compliances are most often of the same order of magnitude and, as a result, the Biot coefficient may be much smaller than 1. Consider, for example, a sample of porous elastic rock. The pores are saturated with fluid at a pressure, p, and a total external pressure, P, is applied around the periphery (i.e., on the outside of an impermeable sleeve). The problem can be analyzed by superposition of two stress states: *state a*, in which fluid pressure and external pressure are both equal to p; and *state b*, in which pore pressure is zero, and the external pressure is $P - p$ (see Figure 2.10).

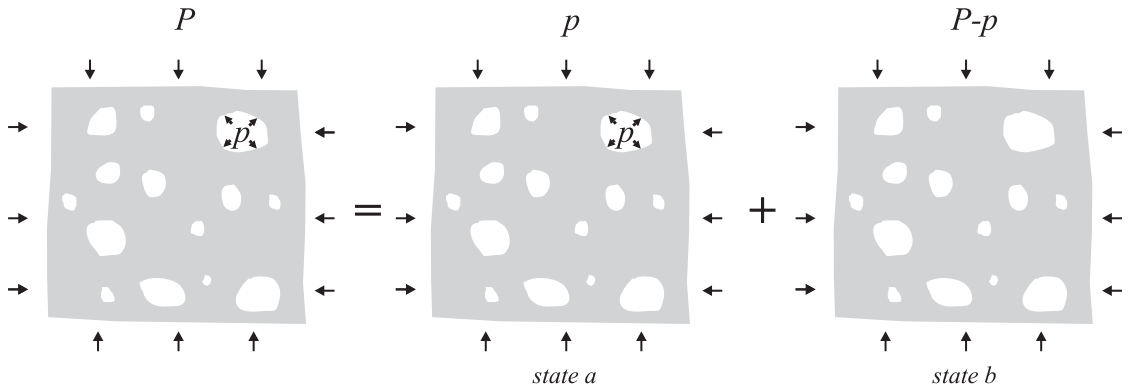

Figure 2.10 Decomposition of stresses acting on a porous, elastic rock

The stress-strain relation for *state a* may be expressed as

$$p = K_s \epsilon_a \tag{2.52}$$

For *state b* (there is no fluid), we can write

$$P - p = K \epsilon_b \tag{2.53}$$

The total strain is given by superposition of the strain in *state a* and in *state b*:

$$\epsilon = \epsilon_a + \epsilon_b \tag{2.54}$$

After substitution of ϵ_a from Eq. (2.51), and ϵ_b from Eq. (2.52), we obtain

$$\epsilon = \frac{p}{K_s} + \frac{P - p}{K} \tag{2.55}$$

After some manipulations, the stress-strain equation takes the form

$$P - \left[1 - \frac{K}{K_s}\right] p = P - \alpha p = K\epsilon \tag{2.56}$$

Clearly then, in the framework of Biot theory, a zero Biot coefficient implies that the elastic stress-strain law becomes independent of pore pressure. Of course, in general, porous rocks do not behave elastically, and pore pressure has an effect on failure. Also, if fluid flow in rocks occurs mainly in fractures, Biot theory may not be applicable. Nonetheless, there are numerous instances where the small value of the Biot coefficient may help explain why pore pressure has little effect on deformation for solid, porous (i.e., unfractured) rocks. (For example, the effect on surface settlement of raising or lowering of the water table in a solid, porous rock mass may be unnoticeable.)

Detournay and Cheng (1993) describe laboratory measurements to determine Biot coefficient for saturated rocks. Table 2.3 lists the Biot coefficient along with drained bulk modulus for several rock types.

Table 2.3 Poroelastic constants for some rocks (Detournay and Cheng 1993)

Rock	K (GPa)	α
Rule sandstone	13	0.65
Tennessee marble	40	0.19
Charcoal granite	35	0.27
Berea sandstone	8	0.79
Westerly granite	25	0.47
Weber sandstone	13	0.64
Ohio sandstone	8.4	0.74
Boise sandstone	4.6	0.85

The logic for grain compressibility, as developed in the framework of Biot theory, is provided in *UDEC*. A simple verification example is described in Section 2.5.1 to illustrate the logic.

2.4.6 Solution

For flow-only calculations (**block mechanical active off**) with compressible flow, the solution time is based upon the fluid timestep and is controlled by the **block cycle *n*** command, where ***n*** is the number of fluid flow steps.

For a coupled mechanical-flow calculation run to steady state flow (**block fluid steady-state**), the **block solve** command can be used to automatically stop the simulation when a steady state is reached. The update of mechanical quantities as a function of fluid pressure occurs by default when pressures change by 1% of the maximum pressure. This can be changed with the **block fluid steady-state tolerance-pressure** command.

For **compressible**, **gas-flow** or **two-phase** fluid-flow, the mechanical and fluid timesteps required for stability in a coupled simulation are calculated independently. In a dynamic flow analysis, these timesteps should be the same, so the user should use the **block mechanical timestep-factor *fb fz fw*** command to change either the mechanical step (***fb*** or ***fz***) or the fluid step (***fw***). If a quasi-static approximation is assumed, the mechanical time scale has no significance, so the user may use the **block fluid compressible substep-mechanical** command to perform several mechanical steps within each cycle.

For **gas-flow**, **steady-state**, **compressible** or **two-phase**, cycle number and time refer to mechanical steps and mechanical time. For **steady-state** mode, the fluid timestep is set equal to the mechanical timestep, since the fluid time scale is arbitrary. Convergence to steady state can be checked with the total inflow/outflow indicators given by **block list max**.

When using **incompressible** mode, the timestep must be defined by the user with **block fluid incompressible timestep**. In the **block cycle** command, either the number of fluid steps or the flow time increment is given (with **block cycle fluid-time**). The mechanical cycle number and the total flow time are printed to the screen when the **block cycle** command is issued.

For coupled, thermal simulations, if **SET flow-steady** is specified, the fluid flow timestep has no physical meaning; thermal time is the only relevant time measure. However, if an incompressible flow calculation is performed (**block fluid incompressible**), then the thermal and hydraulic times should be consistent. Use the **block thermal timestep** dt_t command to define the thermal timestep, and **block fluid incompressible timestep** dt_f to define the fluid-flow timestep.

The number of thermal steps is specified by **block thermal substep-thermal *nt***, and the number of fluid flow steps by **block thermal incompressible substep-fluid *nf***. When the **block thermal cycle step = *ns*** command is invoked, the program alternates between ***nt*** thermal timesteps and ***nf*** flow timesteps until the number of thermal timesteps reaches ***ns***. In order for fluid-flow and thermal times to be consistent, the following condition must be satisfied.

$$nt \times dt_t = nf \times dt_f$$

If this condition is not fulfilled, *UDEC* issues a warning and automatically modifies dt_t to satisfy the condition.

2.4.7 Output Options

The **block fluid list** command keywords **domain, contact, flow** and **max** list the main flow-related variables. Graphical output is obtained by selecting either the

Histories of flow rates at a particular contact may be recorded with **block contact history flow-rate** ***x y***, where ***x y*** is the contact location. Domain pressure histories are requested with **Block Domain history pore-pressure** ***x y***; the domain closest to (***x,y***) is selected.

Note that when using the **steady** option, the flow time scale has no physical meaning, and these histories are only worth checking if steady-state conditions have been reached. For the **compressible** or **incompressible** flow, the flow time should be recorded in a history (with **block fluid history compressible flowtime** or **block fluid history incompressible flowtime**) so that the other histories may be plotted against it(since mechanical time has no real significance).

2.5 Verification and Example Problems

Several verification problems and example exercises are presented to illustrate fluid-flow modeling in *UDEC*. The data files for these examples are located in the "Datafiles\Fluid" directory.

2.5.1 Heave of a Rock Layer

The logic for grain compressibility in a saturated rock, as developed in the framework of Biot theory, is provided in *UDEC*. To illustrate this, we consider a layer of sandstone of large lateral extent and thickness $H = 500$ meters, resting on a rigid base. The layer is elastic, the drained bulk modulus of the rock, K, is 13 GPa, and the shear modulus, G, is 8.43 GPa. The bulk density of the dry rock is 2000 kg/m^3, and the density of water, ρ_w, is 1000 kg/m^3. The porosity of the rock, n, is uniform with a value of 0.1. Gravity, g, is set to 10 m/sec^2.

The rock is initially dry. The water table is then raised to the rock surface, which induces heave in the rock. The surface heave can be evaluated analytically using Eq. (2.57),

$$u_h = -\frac{(n-\alpha)\rho_w g}{2\alpha_1} H^2 \tag{2.57}$$

where $\alpha_1 = K + 4G/3$.

A single columnar shaped block is used for this exercise. The block is 500 meters in the y-direction and 10 meters in the x-direction. The origin of axes is at the bottom of the model. The mechanical boundary conditions correspond to roller boundaries at the base and lateral sides of the model.

We first consider equilibrium of the dry layer. We initialize the stresses using the **block insitu stress** command (without the **zone-pressure** keyword), and use a value of 0.304 ($= (K - 2G/3)/(K + 4G/3)$) for the coefficient of earth pressure at rest, k_o.

There are two competing effects on deformation associated with raising the water level: first, the increase in pore pressure will generate heave of the layer; second, the increase in rock bulk density due to the presence of water in the pores will induce settlement. We proceed to model the combined effects on deformation of a rise in water level up to the rock surface as follows.

We assume the sandstone deformational behavior is similar to Weber sandstone (see Table 2.3) and set the Biot coefficient to 0.64 using the **block zone property biot-coef** command.

We specify a hydrostatic pore-pressure distribution corresponding to a new water level at the top of the model by using either the **block insitu water-table** command or the **block insitu pore-pressure** command with the **pressure-gradient-y** keyword to specify the pore-pressure gradient. The **zone-pressure** keyword is also given with both commands. A total stress correction will now be applied automatically when the pore pressure is changed with the **block insitu** command.

It is also necessary to specify a wet (or saturated) density for the rock below the water table. The saturated density, ρ_s, is calculated from the equation

$$\rho_s = \rho_d + n\rho_w \tag{2.58}$$

where ρ_d is the bulk density of the dry rock, n is the porosity of the rock and ρ_w is the density of water. For the given properties in this example, the saturated density of the rock is 2100 kg/m^3. This density is specified with the **block zone property density** command. The data file for this simulation is listed in Example 2.1.

Using Eq. (2.57), the heave of the rock after raising the water table is calculated to be 2.78 $\times 10^{-2}$ meters. This compares well with the displacement calculated by *UDEC*. Figure 2.11 plots displacement vectors that show the heave for this model. Note that if the Biot effect is not included (i.e., α assumed equal to 1), then the calculated heave is 4.67 $\times 10^{-2}$ meters.

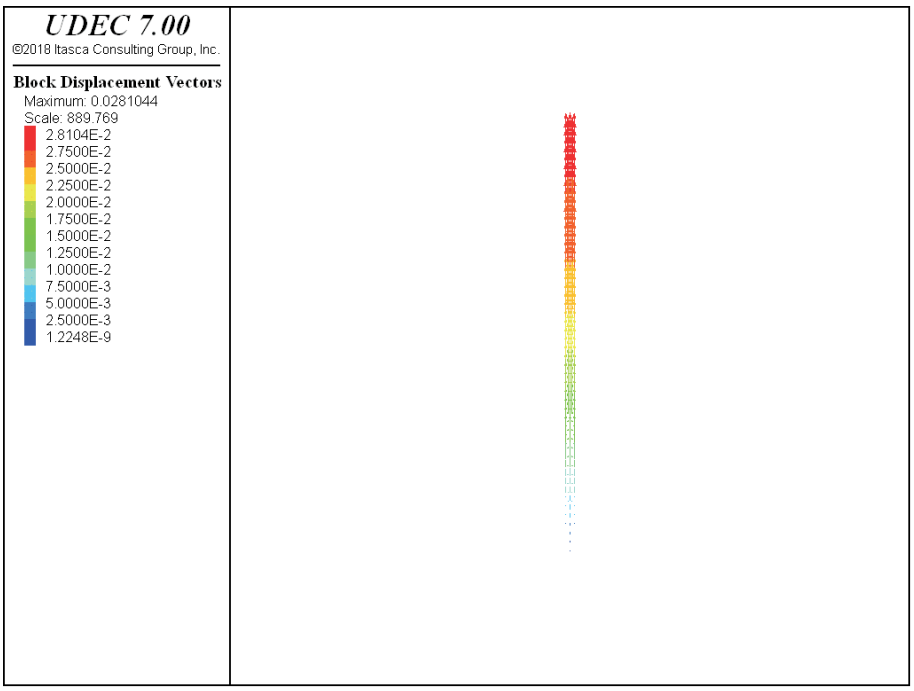

Figure 2.11 Heave of a rock layer

Example 2.1 Heave of a rock layer

```
model new
;file 'heave.dat
model title 'Fluid Pressure Heave'
block config fluid
block fluid clear steady-state off
block tolerance corner-round-length 0.1
block tolerance minimum-edge-length 0.2
block create polygon 0 0 0 500 10 500 10 0
block zone gen quad 10.0
block zone group 'sandstone'
block zone cmodel assign elastic dens 2000 bulk 13e9 ...
  shear 8.43e9 range group 'sandstone'
block fluid  property density 1000.0
block gridpoint apply velocity-x 0 range pos-x -0.1 0.1 pos-y -0.1 500.1
block gridpoint apply velocity-x 0 range pos-x 9.9 10.1 pos-y -0.1 500.1
block gridpoint apply velocity-y 0 range pos-x -0.1 10.1 pos-y -0.1 0.1
model gravity 0 -10
block gridpoint history displacement-y 5.0 250.0
block gridpoint history displacement-y 5.0 500.0
block insitu  stress -3040000.0 0.0 -10000000.0 gradient-x 0.0 0.0 0.0 ...
  gradient-y 6080.0 0.0 20000.0 stress-ZZ -3040000.0 gradient-z 0.0 6080.0
block mechanical history unbalanced-maximum
block solve ratio 1.0E-5
model save 'h1.sav'
;
block zone property density 2.1E3 range group 'sandstone'
block zone property biot 0.64 range group 'sandstone'
block insitu  water-table 500 zone-pressure
;  or install pore pressure distribution with block INSITU Pore-P
;  block insitu pore-p 5000000 pressure-grad-y -10000 zone_pressure
;
block gridpoint init displacement-x 0
block gridpoint init displacement-y 0
block solve ratio 1.0E-5
model save 'h2.sav'
return
```

2.5.2 Example Application of the Fluid Boundary

In order to validate the fluid boundary logic, the mesh shown in Figure 2.8 was applied as a fluid boundary for the *UDEC* block model shown in Figure 2.12.

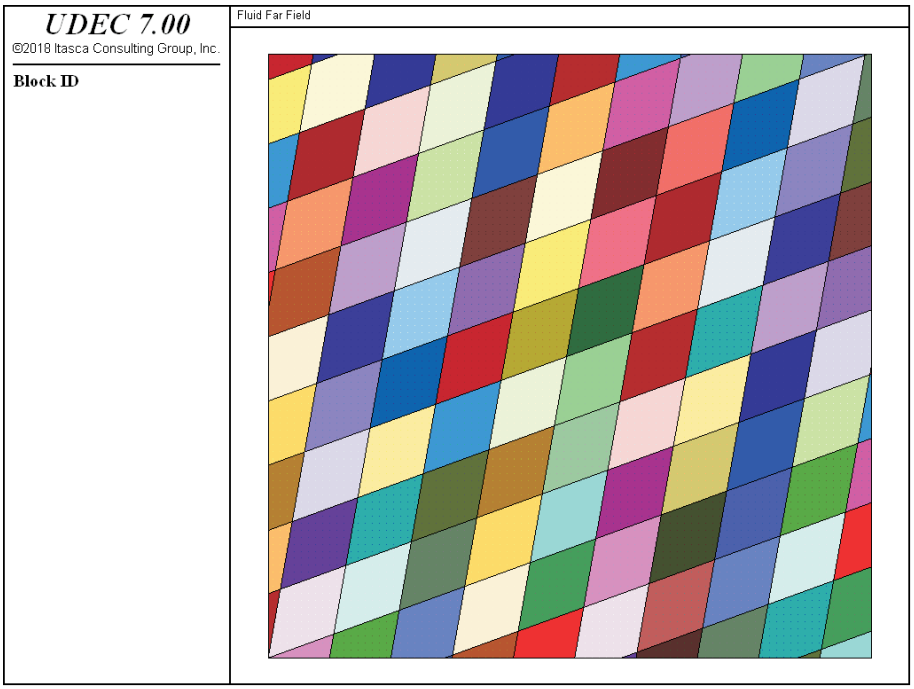

Figure 2.12 UDEC model for fluid boundary example

The joint characteristics for this example are

aperture	constant and equal to 10^{-4}m
joint permeability factor	100 $Pa^{-1}s^{-1}$
spacing	1 m

Two joint sets, with orientations 20° and 80° from the x-axis, are generated. The equivalent permeability tensor for any continuous joint set with spacing, s, orientation, α, from the x-axis, aperture, a, and joint permeability factor, k_j, can be derived:

$$C = k_j a^3 \tag{2.59}$$

where C is the conductivity of a joint.

$$K_j = \frac{C}{s} = \frac{k_j a^3}{s} \tag{2.60}$$

where K_j is the equivalent permeability in the direction of the joint set.

$$\begin{aligned} K_{11} &= K_j \ \cos^2 \alpha \\ K_{22} &= K_j \ \sin^2 \alpha \\ K_{12} &= K_j \ \sin \alpha \ \cos \alpha \end{aligned} \tag{2.61}$$

where K_{11}, K_{22} and K_{12} are the three components of the equivalent permeability tensor. If several joint sets are superposed, their contributions to the permeability tensor should be summed. In our example, this yields

$$K_{11} = 0.913\ 10^{-10}\ \text{m}^3/(\text{Pa s})$$

$$K_{22} = 1.087\ 10^{-10}\ \text{m}^3/(\text{Pa s})$$

$$K_{12} = 0.492\ 10^{-10}\ \text{m}^3/(\text{Pa s})$$

A horizontal pressure gradient, $p = 20 \times 10^4 + 10^4 x$, in Pa/m is imposed on the outer fluid boundary. The corresponding data file is listed in Example 2.2. This problem is run as an uncoupled analysis. First, a flow-only run is made to steady state, and then a mechanical-only run is made.

The fluid pressure field in the model at steady state is shown in Figure 2.13.

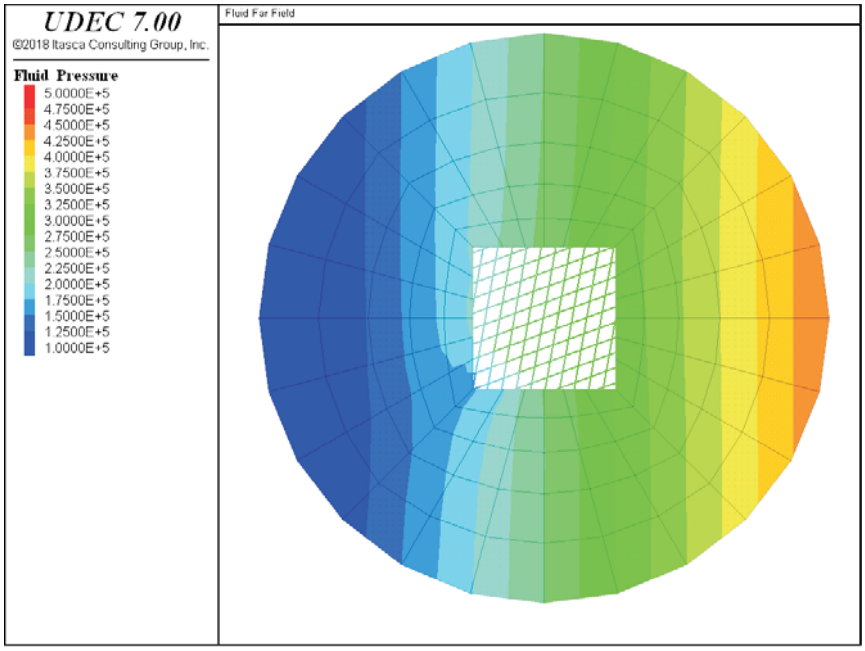

Figure 2.13 Fluid pressure field

Example 2.2 Application of the fluid boundary logic

```
model new
; test of fluid boundary logic
model title 'Fluid Far Field'
block config fluid
block fluid clear steady-state on
block tolerance corner-round-length 0.01
block tolerance minimum-edge-length 0.02
block create (corner-angle 15) polygon 0,0 0,10 10,10 10,0
block fluid far-field permeability-xx 9.13E-11 ...
  permeability-xy 4.92E-11 permeability-yy 1.087E-10 ...
  pore-pressure 200000.0 gradient-x 10000.0 mesh-radius 20.0 ...
  mesh-ratio 1.2 mesh-elements 5
block cut joint-set angle 20 spacing 1 origin 0 0
block cut joint-set angle 80 spacing 1 origin 0 0
block delete range area 0.01
block joint-delete
block zone gen edge 1.0
block zone group 'rock'
block zone cmodel assign elastic density 1E3 bulk 1E9 ...
  shear 5E8 range group 'rock'
block contact group 'joint'
block contact cmodel assign area stiffness-shear 1E8 ...
```

```
  stiffness-normal 1E8 friction 45 cohesion 1E4 tension 1E4 ...
  permeability-factor 100 aperture-residual 1E-4 ...
  aperture-zero-load 1E-4 range group 'joint'
block contact cmodel default material 1
block contact property material 1 stiffness-shear 1E8 ...
  stiffness-normal 1E8 friction 45 cohesion 1E4 tension 1E4 ...
  permeability-factor 100 aperture-residual 1E-4 aperture-zero-load 1E-4
block fluid  property density 1000.0
block edge apply stress -1000000.0 0.0 -1000000.0
block edge apply pore-pressure 200000.0 pressure-gradient-y -10000.0
block insitu  stress -1000000.0 0.0 -1000000.0 stress-ZZ -1000000.0 ...
  pore-pressure 20e4 pressure-gradient-x 0 pressure-gradient-y -1e4 ...
  zone-pressure
model gravity 0.0 -10.0
block domain history pore-pressure 0.0 0.0
block domain history pore-pressure 10.0 10.0
block fluid aperture-max-ratio 1.0
block mechanical active off
block mechanical history unbalanced-maximum
; cycle with fluid boundary to steady state (no mechanical)
block cycle 500
model save 'sf1.sav'
;
block gridpoint apply velocity-x 0 range pos-x -0.1 0.1 pos-y -0.1 10.1
block gridpoint apply velocity-x 0 range pos-x 9.9 10.1 pos-y -0.1 10.1
block gridpoint apply velocity-y 0 range pos-x -0.1 10.1 pos-y -0.1 0.1
block mechanical active on
block fluid flow off
; cycle to mechanical equilibrium (no flow)
block solve ratio 1.0E-5
model save 'sf2.sav'
;
block mechanical active off
block fluid flow on
; check that fluid is at steady state
block cycle 1000
model save 'sf3.sav'
;
```

2.5.3 Steady-State Fluid Flow with Free Surface

The fluid flow logic in *UDEC* allows for situations with a free surface in addition to confined flow problems. This test compares the *UDEC* results with a simple analytical solution for 2D flow in a homogeneous aquifer governed by Darcy's law:

$$v = k\, i \tag{2.62}$$

where v = discharge velocity;
k = coefficient of permeability (length / time); and
i = hydraulic gradient.

The problem is shown in Figure 2.14:

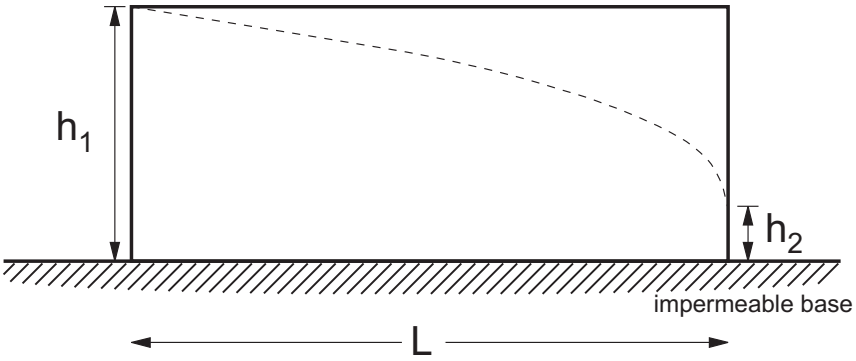

Figure 2.14 Figure showing definition of terms in Dupuit's formula

Dupuit's formula (see, for example, Harr 1962, p. 42) gives the total discharge (per unit width) as

$$Q = k\, \frac{h_1^2 - h_2^2}{2\, L} \tag{2.63}$$

The *UDEC* model is shown in Figure 2.15 (block plot). A system of two sets of joints with constant aperture (i.e., stress-independent) and constant joint spacing, s, was used to simulate the homogeneous isotropic medium.

The dimensions are

L = 8 m
h_1 = 4 m
h_2 = 1 m
s = 0.5 m

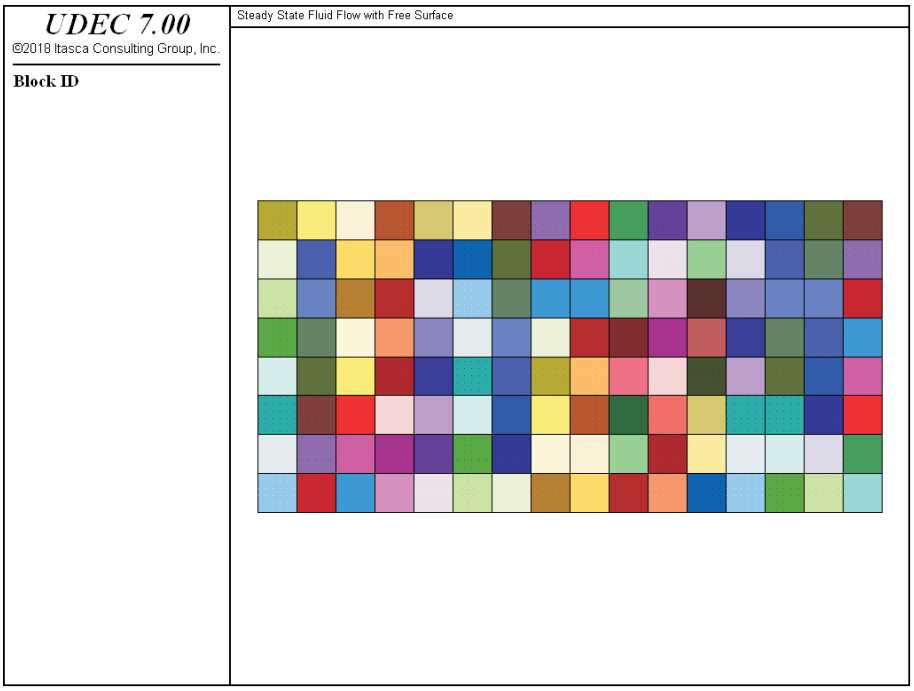

Figure 2.15 UDEC problem geometry for verification of fluid flow logic

The flow rate in a single joint of length, L, subject to a pressure difference of ΔP, is calculated in *UDEC* as

$$q = k_j \, a^3 \, \frac{\Delta P}{L} \tag{2.64}$$

where $k_j = (1/12\mu)$;

μ = dynamic viscosity; and

a = aperture.

For a system of joints with spacing, s, the average velocity for an equivalent porous medium would be

$$\begin{aligned} v &= \frac{q}{s} = \frac{1}{12\,\mu}\, a^3\, \frac{\Delta P}{L}\, \frac{1}{S} \\ &= \frac{1}{12\,\mu}\, a^3\, \rho_w\, g\, \frac{\Delta h}{L}\, \frac{1}{S} \end{aligned} \tag{2.65}$$

where ρ_w is the fluid mass density (e.g., ρ_w = 1000 kg/m^3 for water); and

g is the gravitational acceleration.

Because $\Delta h/L$ is the hydraulic gradient i in Darcy's law, the coefficient k corresponds to

$$k = \frac{\rho_w\, g}{12\,\mu}\, \frac{a^3}{S} \tag{2.66}$$

For the conditions

a = 0.0001 m

ρ_w = 1000 kg/m^3

g = 10 m/sec^2

S = 0.5 m

$k_j = \frac{1}{12\,\mu}$ = 83.3 Pa^{-1} sec^{-1}

we have $k = 1.67 \times 10^{-6}$ m/sec.

Then, Dupuit's formula gives

$$Q = 1.565 \times 10^{-6}\ \text{m}^3/\text{sec}$$

The *UDEC* model gives (from the sum of discharge flow rates given by the **block list max** command)

$$Q = 1.464 \times 10^{-6}\ \text{m}^3/\text{sec}$$

The error is 6.4%.

Figure 2.16 shows the flow rates for the steady state. Figure 2.17 shows the domain pressures.

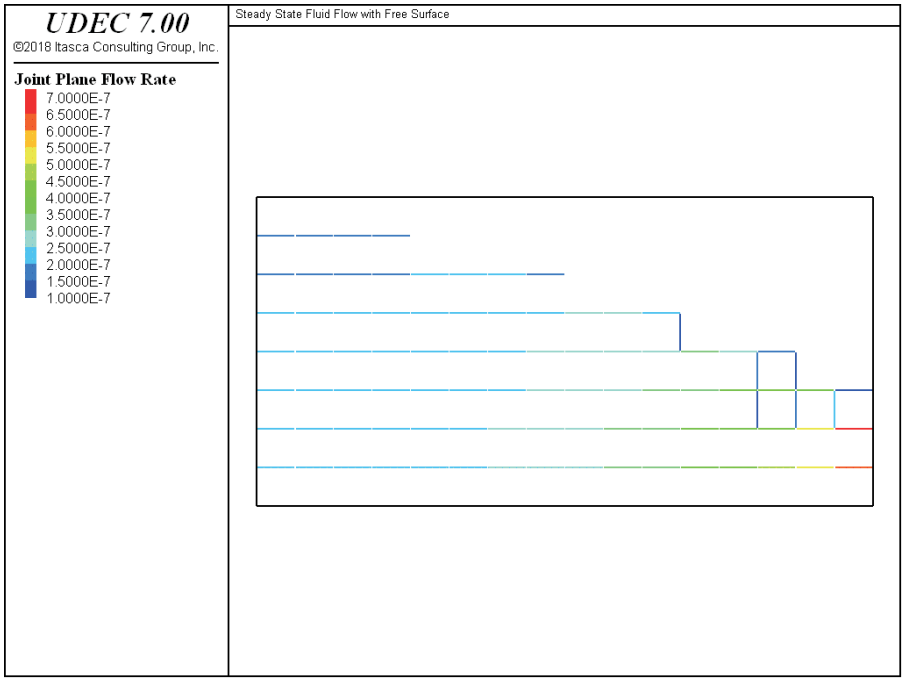

Figure 2.16 UDEC steady-state flow rates

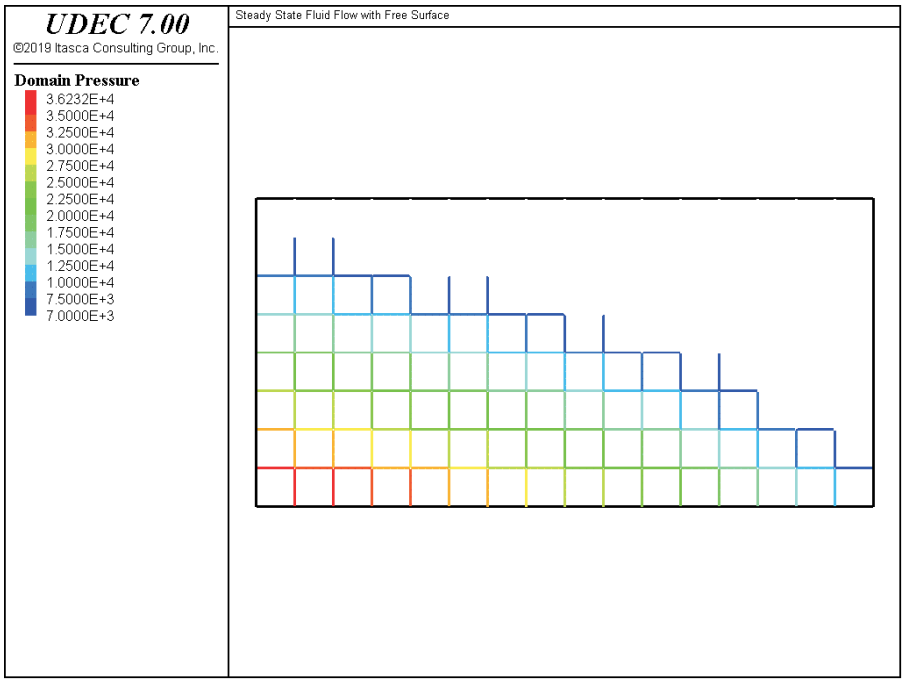

Figure 2.17 UDEC domain pressures

Example 2.3 Steady-state fluid flow with a free surface

```
model new
model title 'Steady State Fluid Flow with Free Surface'
block config fluid
block fluid steady-state on
block tolerance corner-round-length 0.01
block tolerance minimum-edge-length 0.02
block create polygon -4 0 -4 4 4 4 4 0
block cut joint-set angle 0 trace 100 spacing 0.5 origin 0 0
block cut joint-set angle 90 trace 100 spacing 0.5 origin 0 0
block change material 1
block property material 1 density 1
block contact group 'joint'
block contact cmodel assign area stiffness-shear 1 ...
  stiffness-normal 1 permeability-factor 83.3 aperture-residual 0.0001 ...
  aperture-zero-load 0.0001 range group 'joint'
; set new contact default
block contact cmodel default material 1
block contact property material 1 stiffness-shear 1 stiffness-normal 1 ...
  permeability-factor 83.3 aperture-residual 0.0001 ...
  aperture-zero-load 0.0001
block fluid  property density 1000.0
model gravity 0 -10
block edge apply pore-pressure 40000.0 pressure-gradient-y -10000.0 ...
  range position-x -4.01 -3.99 position-y -0.1 4.1
block edge apply pore-pressure 10000.0 pressure-gradient-y -10000.0 ...
  range position-x 3.99 4.01 position-y -0.1 1.1
block edge apply impermeable range position-x -4.1 4.1 position-y -0.1 0.1
block domain history pore-pressure -4.0 2.0
block domain history pore-pressure 4.0 0.0
block mechanical active off
block cycle 1000
model save 'steady.sav'
return
```

2.5.4 Pressure Distribution in a Fracture with Uniform Permeability (Aperture)

In this problem, time variations of pressure distribution in a fracture are examined as a result of a fluid pressure, P_o, being suddenly applied to one end of a fracture (see Figure 2.18).

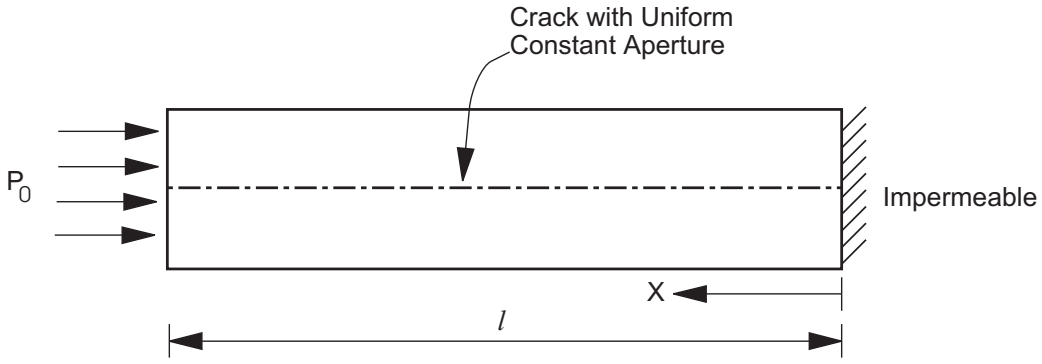

Figure 2.18 Model geometry and boundary conditions

The pressure in a finite length fracture that is suddenly pressurized at one end can be found from an analogous solution given for a 1D heat conduction problem. (See, for example, Hardy and Asgian 1989.)

The solution satisfies the differential equation

$$\left(\frac{\alpha^2}{12\,\mu} \right) \left(\frac{d^2 P}{dx^2} \right) = \beta \frac{dP}{dt} \tag{2.67}$$

where α is the hydraulic aperture (m);

β is the fluid compressibility (Pa^{-1}); and

μ is the dynamic viscosity (Pa/sec).

The values selected for this problem are

$\ell = 1$ m

$P_o = 9.5$ MPa

$\alpha = 3 \times 10^{-5}$ m

$\beta = 5 \times 10^{-8}\ \text{Pa}^{-1}$

$\mu = 10^{-3}\ \text{Pa} \cdot \text{sec}$

A dimensionless time, T, for Eq. (2.67) can be defined as

$$T = \frac{(\alpha^2 \,/\, 12\,\mu)\, t}{\beta\, \ell^2} \tag{2.68}$$

The solution of Eq. (2.67) with the boundary conditions defined in Figure 2.18 is given by Carslaw and Jaeger (1959):

$$\frac{P}{P_o} = 1 + \frac{4}{\pi} \sum_{n=0}^{\infty} \left[e^{-(2n+1)^2 (T/4)\pi^2} \cdot \cos\left(\frac{(2n+1)\,\pi}{2}\, \zeta \right) \cdot \left(\frac{(-1)^{n+1}}{2n+1} \right) \right] \tag{2.69}$$

where T = the dimensionless time;

ζ = x/ℓ (see Figure 2.18);

P = pore pressure at a distance x from the impermeable side; and

P_o = pore pressure at a distance $x = \ell$.

The fluid flow properties are set in *UDEC*:

bulkw = $1/\beta$ = 20 MPa

jperm = $1/(12\,\mu)$ = 8.33×10^7 MPa$^{-1}\cdot$ s^{-1}

The hydraulic aperture is maintained at a constant value of α by setting the *UDEC* parameters

azero = α = 3×10^{-5} m

ares = α = 3×10^{-5} m

Example 2.4 contains the data file for this problem.

Figure 2.19 compares both the analytical solution and the *UDEC* solution for three values of dimensionless time, T. The *FISH* function **ana_sol** calculates the analytical solution for times, T = 0.1, 0.3 and 0.5, and stores the results in Tables 2, 4 and 6, respectively. The function **num_sol** collects the *UDEC* results in Tables 1, 3 and 5 for comparison to the analytical results.

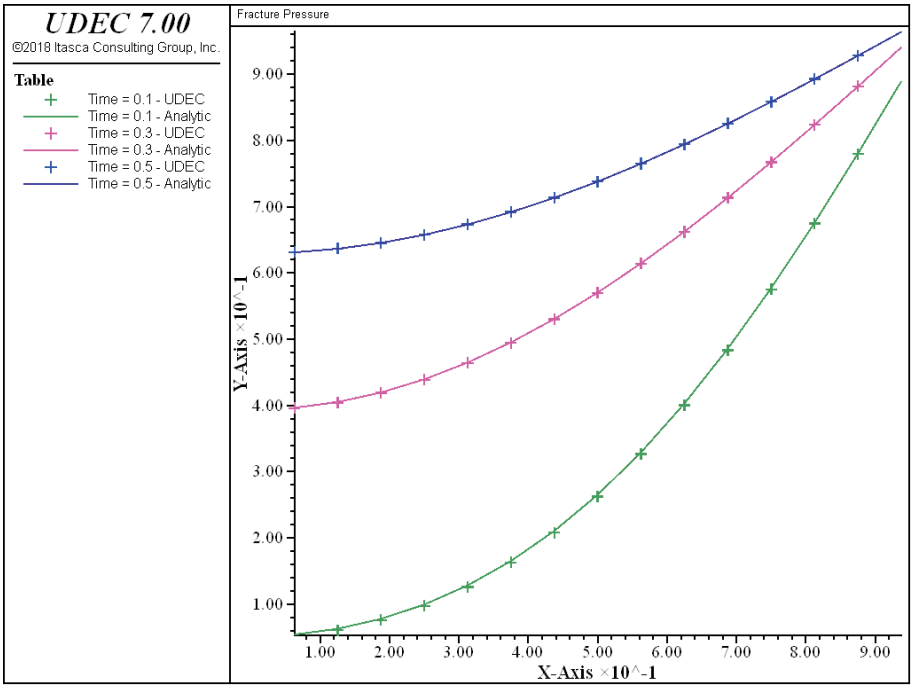

Figure 2.19 Comparison of analytical and UDEC solutions for joint fluid pressure (P / P_o) at various distances (x/l) in a fracture with zero initial pressure at x = 0

Example 2.4 Pressure distribution in a fracture with uniform permeability

```
model new
model title 'Fracture Pressure'
fish define constants
   tabo = -1
   tabe = 0
   length = 1.0
   p_0 = 9.5
   alpha = 3.0e-5
   beta  = 5.0e-8
   dy_vis = 1.0e-3
   t_cons = (alpha * alpha) / (12.0 * dy_vis * beta * length * length)
   overl = 1.0 / length
   n_max = 100
   fop = 4.0 / math.pi
   teps = 1.0e-4
end
@constants
fish define num_sol
   tabo = tabo + 2
   pnt = block.domain.head
```

```
    loop while pnt # 0
       x_pos = length - block.domain.pos.x(pnt)
       if block.domain.volume(pnt) > 0.0 then
          table(tabo,x_pos) = block.domain.pp(pnt) / p_0
       endif
       pnt = block.domain.next(pnt)
    endloop
end
fish define ana_sol
    tabe = tabe + 2
    t_cap = t_cons * block.fluid.time.total
    pnt = block.domain.head
    loop while pnt # 0
       if block.domain.volume(pnt) > 0.0 then
          x_pos = block.domain.pos.x(pnt)
          x_l   = x_pos * overl
          n     = 0
          tsum  = 0.0
          tsumo = 0.0
          sgnn = -1.0
          converge = 0
          loop while n < n_max
             fn = float(n)
             n = n + 1
             tnp1 = 2.0 * fn + 1.0
             term1 = math.exp(-0.25*(tnp1^2)*t_cap*math.pi*math.pi)
             term2 = math.cos(0.5*tnp1*math.pi*x_l)
             term3 = sgnn / tnp1
             term  = term1 * term2 * term3
             tsum = tsumo + term
             if math.abs(tsum - tsumo) < teps then
                table(tabe,x_pos) = 1.0 + fop * tsum
                converge = 1
                n = n_max
             else
                tsumo = tsum
             endif
             sgnn = -1.0 * sgnn
          endloop
          if converge = 0 then
  ii = io.out(' not converged x= ' + string(x_pos) + ' t= ' + ...
    string(block.fluid.time.total))
             exit
          endif
       endif
       pnt = block.domain.next(pnt)
```

```
   endloop
end
;
block config fluid
block fluid compressible on
block tolerance corner-round-length 0.001
block tolerance minimum-edge-length 0.002
block create polygon 0 0 0 0.1 1 0.1 1 0
block cut crack 0 0.05 1 0.05
block zone gen edge 0.2
block zone group 'dummy'
block zone cmodel assign elastic density 1 bulk 1 shear 1 ...
  range group 'dummy'
block contact group 'joint'
block contact cmodel assign area stiffness-shear 1 ...
  stiffness-normal 1 permeability-factor 8.33E7 ...
  aperture-residual 3E-5 aperture-zero-load 3E-5 range group 'joint'
block contact cmodel default material 1
block contact property material 1 stiffness-shear 1 stiffness-normal 1 ...
  permeability-factor 8.33E7 aperture-residual 3E-5 ...
  aperture-zero-load 3E-5
block fluid  property density 0.0010
block fluid  property bulk 20.0
block edge apply pore-pressure 9.5 ...
  range position-x -0.001 0.001 position-y -0.001 0.101
block edge apply impermeable ...
  range position-x 0.999 1.001 position-y -0.001 0.101
;
; print status to find out fluid flow time step, t,
; for the determination of dimensionless time T
;
block cycle 0
block list info
block mechanical active off
;
;  flow time = 0.0669  (T = 0.1)
block cycle 104
@num_sol
@ana_sol
model save 'time1a.sav'
;
;
;  flow time = 0.20   (T = 0.3)
block cycle 207
@num_sol
@ana_sol
```

```
model save 'time2a.sav'
;
;
;  flow time = 0.33  (T = 0.5)
block cycle 207
@num_sol
@ana_sol
;
table 1 label 'Time = 0.1 - UDEC'
table 2 label 'Time = 0.1 - Analytic'
table 3 label 'Time = 0.3 - UDEC'
table 4 label 'Time = 0.3 - Analytic'
table 5 label 'Time = 0.5 - UDEC'
table 6 label 'Time = 0.5 - Analytic'
model save 'time3a.sav'
return
```

2.5.5 Transient Fluid Flow in a Single Joint in an Elastic Medium

A simple example of a single joint embedded in an elastic medium was used to test the incompressible flow algorithm (**SET flow incompressible**) and the compressible flow algorithm (**SET flow compressible**). Figure 2.20 shows the problem geometry. The joint extends from $x = 0$ to $x = 10$ m, and is discretized into 10 segments. The 2-m end segments on either side are assumed to be fictitious (construction) joints (i.e., high strength and impermeable). Flow is injected at a constant rate of 5×10^{-7} m^3/sec into the left of the actual joint (A). Pressure is held at the in-situ value $\boldsymbol{p}_o$ at the right end (D). The following properties were used.

Blocks

Young's modulus	$E = 5 \times 10^{10}$ Pa
Poisson's ratio	$v = 0.25$

Joint

normal and shear stiffness	$k_n = k_s = 2 \times 10^{10}$ Pa/m
permeability factor	$k_j = 300$ $Pa^{-1}sec^{-1}$
initial aperture (at zero stress)	$a_o = 170 \times 10^{-6}$ m
residual aperture	$a_{res} = 20 \times 10^{-6}$ m

In-situ stresses

vertical stress	$\sigma_{yy} = -3.1 \times 10^6$ Pa
fluid pressure	$p_o = 1 \times 10^5$ Pa

Simulations with fluid timesteps of 1 second and 10 seconds were performed. Figure 2.21 shows the evolution of the pressure at the domains A, B and C during the first 100 seconds, obtained in the run with the smaller timestep. Figure 2.22 shows the corresponding curves for a timestep of 10 seconds. It can be seen that the results agree very well, except for the initial transient, which obviously cannot be represented with the larger timestep. This second run was continued until steady state was reached, as shown in Figure 2.23.

As a comparison, the same problem was run with the **block fluid compressible** option. A fluid bulk modulus, K_w, of 20 MPa was assumed. This is about 1/100 of the real modulus for pure water (i.e., no entrained air). But, given the joint apertures, the effective fluid stiffness (K_w/a) is on the order of 10^{12} (i.e., two orders of magnitude higher than the joint normal stiffness). The pressure histories at the same three points, displayed in Figures 2.24 and 2.25, are very close to those from the run with **block fluid incompressible** (Figures 2.21 and 2.23), indicating that the assumption of fluid incompressibility is justified for this type of problem.

The **incompressible** simulation with the 1 second timestep was run for 100 fluid steps. The total number of mechanical steps performed was 1937. The **compressible** simulation, however, required a much smaller fluid timestep for numerical stability. Initially, the calculated timestep was 2×10^{-2} sec.; subsequently, as the joint opened, the timestep was gradually reduced to a value of 4×10^{-3} sec. Thus, the 100-second simulation required about 10^4 fluid steps. To agree with the assumption of a quasi-static process implied in the **incompressible** option, 10 mechanical iterations were performed per cycle (i.e., **block fluid incompressible substep-mechanical** = 10 was used). Therefore, a total of about 10^5 mechanical steps was required in the **compressible** run, showing that, in practice, the consideration of fluid compressibility is only possible for short-term simulations.

Example 2.5 contains the data file for the **incompressible** flow calculation. Example 2.6 contains the data file for the **compressible** flow calculation.

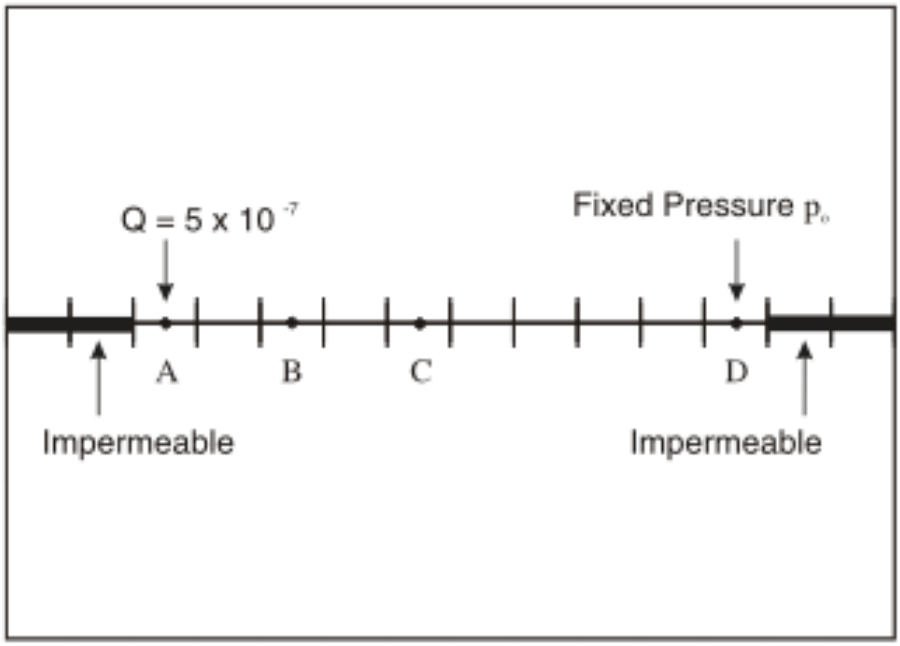

Figure 2.20 UDEC model for incompressible flow in a single joint

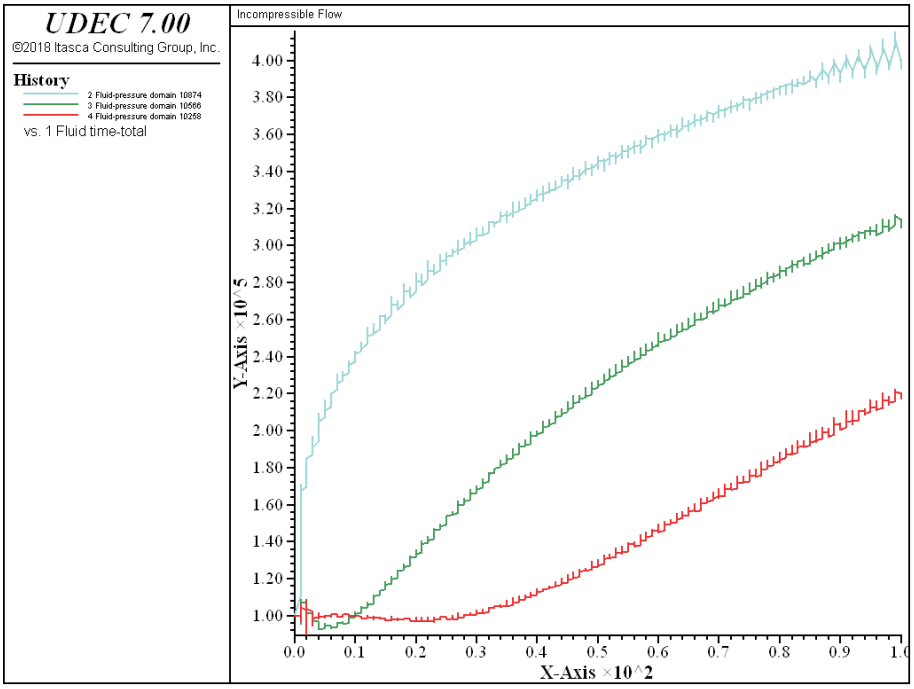

Figure 2.21 Fluid pressure histories at A, B and C for a 1-second fluid timestep

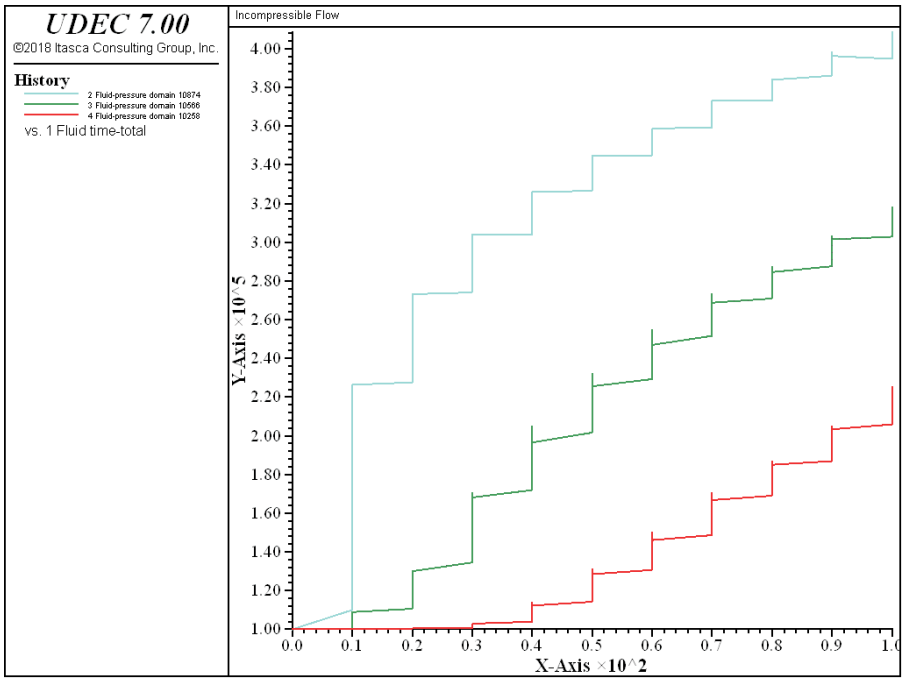

Figure 2.22 Fluid pressure histories at A, B and C for a 10-second fluid timestep

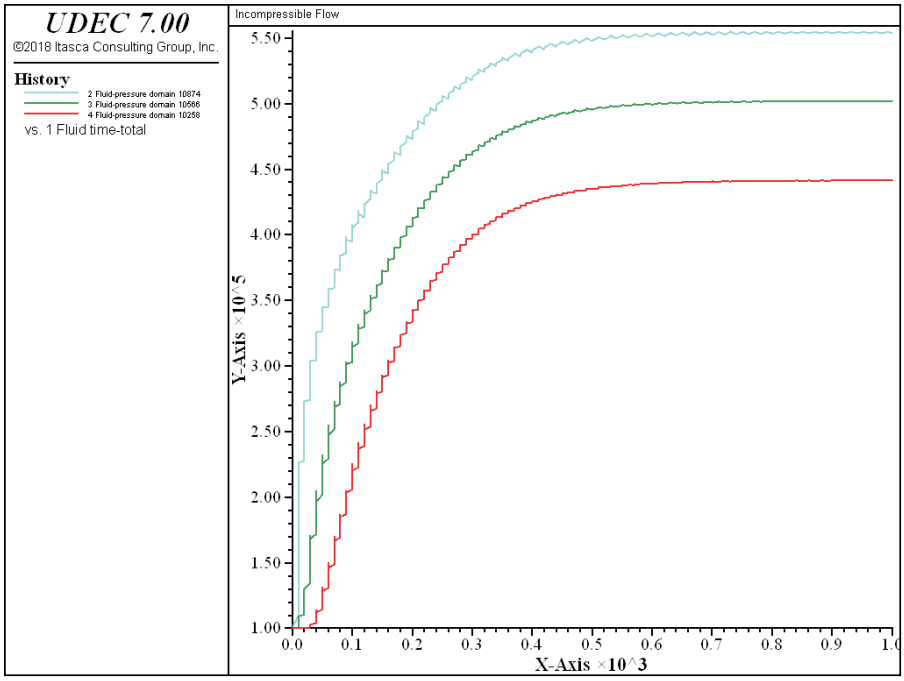

Figure 2.23 Fluid pressure histories at A, B and C for a 10-second fluid timestep; run to steady-state flow

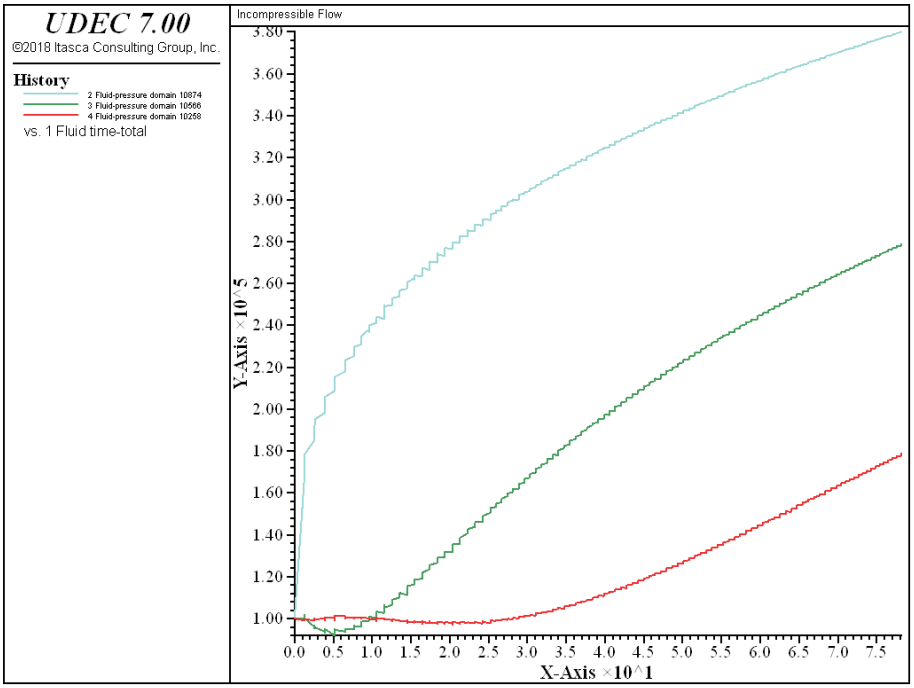

Figure 2.24 Fluid pressure histories at A, B and C for compressible flow

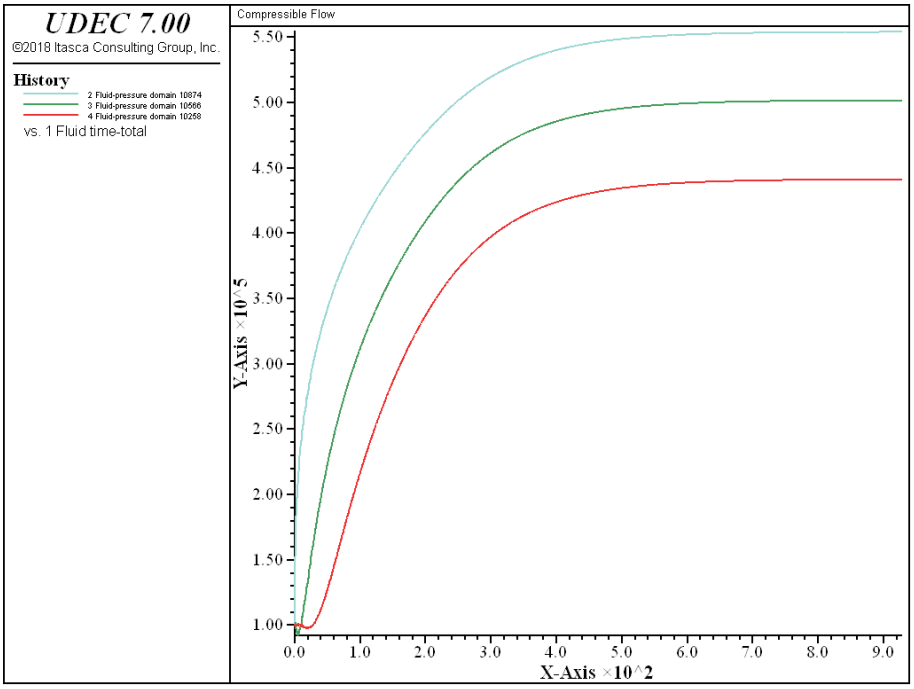

Figure 2.25 Fluid pressure histories at A, B and C for compressible flow; run to steady-state

Example 2.5 Incompressible transient flow in a single joint in an elastic medium

```
model new
; ------------------------------------------------
;  run with SET FLOW INCOMPRESSIBLE option
; ------------------------------------------------
model title 'Incompressible Flow'
block config fluid
block fluid incompressible on
block tolerance corner-round-length 0.001
block tolerance minimum-edge-length 0.002
block create polygon -2 -5 -2 5 12 5 12 -5
block cut crack -2 0 12 0
block zone gen quad 1.01 2.51
block zone group 'block'
block zone cmodel assign elastic density 1E3 bulk 3.3333E9 shear 2E9 ...
  range group 'block'
block contact join by-contact range pos-x -2.1 0.1 pos-y -0.1 0.1
block contact join by-contact range pos-x 9.9 12.1 pos-y -0.1 0.1
; crack
block contact group 'crack'
block contact cmodel assign area stiffness-shear 2E10 ...
```

```
  stiffness-normal 2E10 cohesion 1E20 tension 1E20 ...
  permeability-factor 300 aperture-residual 0.00002 ...
  aperture-zero-load 1.7E-4 range group 'crack'
; set new contact default
block contact cmodel default material 1
block contact property material 1 stiffness-shear 2E10 ...
  stiffness-normal 2E10 cohesion 1E20 tension 1E20 ...
  permeability-factor 300 aperture-residual 0.00002 ...
  aperture-zero-load 1.7E-4
block fluid  property density 1000.0
block insitu  stress 0.0 0.0 -3100000.0 pore-pressure 1e5
block gridpoint apply velocity-x 0 range pos-x -2.1 12.1 pos-y -5.1 -4.9
block gridpoint apply velocity-y 0 range pos-x -2.1 12.1 pos-y -5.1 -4.9
block gridpoint apply velocity-y 0 range pos-x -2.1 12.1 pos-y 4.9 5.1
block gridpoint apply velocity-x 0 range pos-x -2.1 12.1 pos-y 4.9 5.1
; sink at pos-x = 9.5
block domain fix range pos-x 9.4 9.6 pos-y -0.1 0.1
block domain init pore-pressure 1E5 range pos-x 9.4 9.6 pos-y -0.1 0.1
; well at pos-x = 0.5
block domain fix range pos-x -2.1 0.1 pos-y -0.1 0.1
block domain initialize pore-pressure 1E5 ...
  range pos-x -2.1 0.1 pos-y -0.1 0.1
; keep insitu pore-p in impermeable joints
block domain fix range pos-x 9.9 12.1 pos-y -0.1 0.1
block domain initialize pore-pressure 1E5 ...
  range pos-x 9.9 12.1 pos-y -0.1 0.1
block domain well flow 5.0E-7 atdomain 0.5 0.0
block fluid history flow-time
block domain history pore-pressure 0.5 0.0
block domain history pore-pressure 2.5 0.0
block domain history pore-pressure 4.5 0.0
block fluid aperture-max-ratio 100.0
block domain update 100000
history unbvol
model save 'ft0.sav'
;
;   1 second timestep
history interval 1
block fluid incompressible substep-mechanical 100
block fluid incompressible timestep 1.0
block cycle ftime 100
model save 'ft1.sav'
;
;   10 second timestep
model restore 'ft0.sav'
history interval 1
```

```
block fluid incompressible substep-mechanical 100
block fluid incompressible timestep 10.0
block cycle ftime 100
model save 'ft10.sav'
;
; run to steady state
block cycle ftime 900
model save 'ft100.sav'
```

Example 2.6 Compressible transient flow in a single joint in an elastic medium

```
model new
; ------------------------------------------------
;  run with BLOCK FLUID COMPRESSIBLE option
; ------------------------------------------------
model title 'Compressible Flow'
block config fluid
block fluid compressible on
block tolerance corner-round-length 0.001
block tolerance minimum-edge-length 0.002
block create polygon -2 -5 -2 5 12 5 12 -5
block cut crack -2 0 12 0
block zone gen quad 1.01 2.51
block zone group 'block'
block zone cmodel assign elastic density 1E3 bulk 3.3333E9 shear 2E9 ...
  range group 'block'
block contact join by-contact range pos-x -2.1 0.1 pos-y -0.1 0.1
block contact join by-contact range pos-x 9.9 12.1 pos-y -0.1 0.1
; crack
block contact group 'crack'
block contact cmodel assign area stiffness-shear 2E10 ...
  stiffness-normal 2E10 cohesion 1E20 tension 1E20 ...
  permeability-factor 300 aperture-residual 0.00002 ...
  aperture-zero-load 1.7E-4 range group 'crack'
; set new contact default
block contact cmodel default material 1
block contact property material 1 stiffness-shear 2E10 ...
  stiffness-normal 2E10 cohesion 1E20 tension 1E20 ...
  permeability-factor 300 aperture-residual 0.00002 ...
  aperture-zero-load 1.7E-4
block fluid  property density 1000.0
block fluid  property bulk 2.0E7
block insitu  stress 0.0 0.0 -3100000.0 pore-pressure 1e5
block gridpoint apply velocity-x 0 range pos-x -2.1 12.1 pos-y -5.1 -4.9
block gridpoint apply velocity-y 0 range pos-x -2.1 12.1 pos-y -5.1 -4.9
block gridpoint apply velocity-y 0 range pos-x -2.1 12.1 pos-y 4.9 5.1
block gridpoint apply velocity-x 0 range pos-x -2.1 12.1 pos-y 4.9 5.1
; sink at pos-x = 9.5
block domain fix range pos-x 9.4 9.6 pos-y -0.1 0.1
block domain initialize pore-pressure 1E5 ...
  range pos-x 9.4 9.6 pos-y -0.1 0.1
; well at pos-x = 0.5
block domain well flow 5.0E-7 atdomain 0.5 0.0
; keep insitu pp in impermeable joints
```

```
block domain fix range pos-x -2.1 0.1 pos-y -0.1 0.1
block domain initialize pore-pressure 1E5 ...
  range pos-x -2.1 0.1 pos-y -0.1 0.1
block domain fix range pos-x 9.9 12.1 pos-y -0.1 0.1
block domain initialize pore-pressure 1E5 ...
  range pos-x 9.9 12.1 pos-y -0.1 0.1
block fluid history flow-time
block domain history pore-pressure 0.5 0.0
block domain history pore-pressure 2.5 0.0
block domain history pore-pressure 4.5 0.0
history interval 100
block fluid aperture-max-ratio 100.0
block domain update 100000
model save 'fc0.sav'
;
;
block fluid substep-mechanical 10
block fluid substep-mechanical 10
block cycle 10000
model save 'fcc100.sav'
;
;
block cycle 200000
model save 'fcss.sav'
```

2.5.6 Transient One-Dimensional Gas Flow

Chan et al. (1993) present the solution for a transient one-dimensional gas flow problem: the time-evolution of the pressure of an ideal gas draining from a joint into a cavity. The problem conditions involve one-dimensional flow along the joint at constant volume in a semi-infinite domain. At $x = 0$, the gas pressure $p = 0$; at $x = \infty$, the gas pressure is set to p_1.

The authors present both an approximate analytical solution and an empirical fit to a numerical solution obtained with an implicit time-marching scheme. They derive an approximate analytical solution for gas pressure along the joint:

$$\Psi \simeq 1 - exp(-\xi/\sqrt{2} - \xi^2/4) \tag{2.70}$$

They also present an empirical fit to a numerical solution:

$$\Psi = 1 - e^{-(0.625\xi + 0.186\xi^2)} \tag{2.71}$$

In Eqs. (2.70) and (2.71), dimensionless parameters $\Psi = p^2/p_1^2$, and $\xi = \sqrt{(vnx^2)/(tkp_1)}$, where v is the shear viscosity of the gas, n is porosity, k is permeability and t is time. The solutions are converted to a *FISH* function with results written to tables for comparison to *UDEC* results. This *FISH* function, **ppan**, is listed in Example 2.7:

Example 2.7 Analytical solution and empirical fit for 1D gas flow

```
def ppan
;  analytical solution at time=0.5
;   table 8 : pp vs. x
;   table 28 : psi vs. ksi
;   table 10 : empirical pp vs. x
;   table 30 : empirical psi vs. ksi
;
;  analytical solution at time=2.0
;   table 9 : pp vs. x
;   table 29 : psi vs. ksi
;   table 11 : empirical pp vs. x
;   table 31 : empirical psi vs. ksi
;
; fac = k/(visc.*poros.) = (jperm*a3) / poros
; poros = a
; fac = 1e9 * 1e-12 / 1e-4 = 10
;
  fac = 10.
  pp1 = 1.0
  tt = 0.5
```

```
  loop j (1, 10)
    xx = 0.5 + (j-1)*1.0
    ksi = (1.0/fac) * xx*xx / (tt * pp1)
    ksi = sqrt(ksi)
    psi = 1.0 - exp(-ksi/sqrt(2.0)-0.25*ksi*ksi)
    ppj = sqrt(psi*pp1*pp1)
    xtable(28,j) = ksi
    ytable(28,j) = psi
    xtable(8,j) = xx
    ytable(8,j) = ppj
    psi2 = 1.0 - exp(-0.625*ksi-0.186*ksi*ksi)
    ppj2 = sqrt(psi2*pp1*pp1)
    xtable(30,j) = ksi
    ytable(30,j) = psi2
    xtable(10,j) = xx
    ytable(10,j) = ppj2
  endloop
;
  tt = 2.0
  loop j (1, 10)
    xx = 0.5 + (j-1)*1.0
    ksi = (1.0/fac) * xx*xx / (tt * pp1)
    ksi = sqrt(ksi)
    psi = 1.0 - exp(-ksi/sqrt(2.0)-ksi*ksi/4.0)
    ppj = sqrt(psi*pp1*pp1)
    xtable(29,j) = ksi
    ytable(29,j) = psi
    xtable(9,j) = xx
    ytable(9,j) = ppj
    psi2 = 1.0 - exp(-0.625*ksi-0.186*ksi*ksi)
    ppj2 = sqrt(psi2*pp1*pp1)
    xtable(31,j) = ksi
    ytable(31,j) = psi2
    xtable(11,j) = xx
    ytable(11,j) = ppj2
  endloop
end
```

The *UDEC* model contains a single horizontal joint, 10 m long, divided into 10 domains. The model geometry is shown in Figure 2.26.

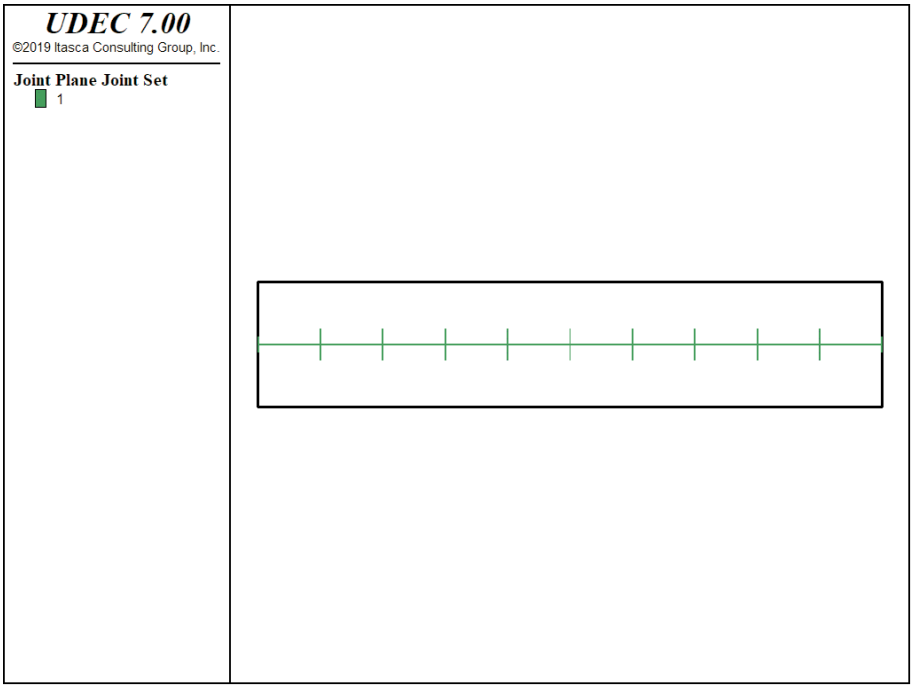

Figure 2.26 UDEC model for 1D gas flow

At $x = 0$, gas pressure is fixed at zero. At $x = 10$, the pressure is set to the value of p defined by Eq. (2.71) for $p_1 = 1$ MPa. The boundary condition is controlled by applying pressure with a *FISH* history multiplier, **`ppanx10`** (see Example 2.8). The problem parameters are $k = 10^9$, $\nu = 10^{12}$ and $n = 10^{-4}$.

Example 2.8 Analytical solution for pressure at $x = 10$

```
 def ppanx10
; analytical solution at x=10
;   fac = k/(visc.*poros.) = (jperm*a3) / poros
;   poros = a
;   fac = 1e9 * 1e-12 / 1e-4 = 10
  fac = 10.
  pp1 = 1.0
  tt = ftime
  xx = 10.0
  ppj = 1.0
  if tt > 0.0
    ksi = (1.0/fac) * xx*xx / (tt * pp1)
    ksi = sqrt(ksi)
    psi2 = 1.0 - exp(-0.625*ksi-0.186*ksi*ksi)
    ppj2 = sqrt(psi2*pp1*pp1)
    psi = psi2
```

```
    ppj = ppj2
  endif
  ppanx10 = ppj
end
```

The *UDEC* data file is listed in Example 2.9. A comparison of the *UDEC* results to the analytical solution (Eq. (2.70)) and the empirical fit (Eq. (2.71)) at time = 0.5 is shown in Figure 2.27, and at time = 2.0 in Figure 2.28.

Example 2.9 Transient one-dimensional gas flow

```
model new
Model title 'Transient One-dimensional Gas Flow'
block config fluid
block fluid clear compressible gasflow on
block tolerance corner-round-length 1E-3
block tolerance minimum-edge-length 2E-3
block create polygon 0 -1 0 1 10 1 10 -1
block cut crack 0 0 10 0
block zone gen quad 1.01 1.01 range pos-x 0 10 pos-y 0 1
block zone gen quad 10.01 1.01 range pos-x 0 10 pos-y -1 0
block zone group 'block'
block zone cmodel assign elastic density 2.7E-3 bulk 0.5555 ...
  shear 0.4167 range group 'block'
block contact group 'joint'
block contact cmodel assign area stiffness-shear 1 stiffness-normal 1 ...
  cohesion 1E20 tension 1E20 permeability-factor 1E9 ...
  aperture-residual 5E-5 aperture-zero-load 1E-4 range group 'joint'
; new contact default
block contact cmodel default area stiffness-shear 1 stiffness-normal 1 ...
  cohesion 1E20 tension 1E20 permeability-factor 1E9 ...
  aperture-residual 5E-5 aperture-zero-load 1E-4
block fluid  gasflow bulk-minimum 0.0010
block fluid  gasflow constant 1.0
block gridpoint apply velocity-x 0
block gridpoint apply velocity-y 0
block edge apply pore-pressure 0.0 range pos-x -0.1 0.1 pos-y -0.1 0.1
block edge apply pore-pressure 1.0 range pos-x 9.9 10.1 pos-y -0.1 0.1
block insitu  pore-pressure 1
block mechanical timestep-factor 0.1 1.0 0.5
block fluid history unbalanced-volume
block domain history pore-pressure 0.5 0.0
block domain history pore-pressure 1.5 0.0
block domain history pore-pressure 4.5 0.0
```

```
block domain history pore-pressure 8.5 0.0
block domain history pore-pressure 9.5 0.0
block contact history flow-rate 0.0 0.0
block contact history flow-rate 5.0 0.0
block contact history flow-rate 10.0 0.0
call 'ppan.fis'
@ppan
call 'ppanx10.fis'
@ppanx10
block edge apply pore-pressure 1.0 history @ppanx10 ...
  range pos-x 9.9 10.1 pos-y -0.1 0.1
call 'pp1dini.fis'
@pp1dini
call 'getpp1d.fis'
@getpp1d
block mechanical active off
block cycle 80
call 'getpp1d.fis'
@getpp1d
model save 'gg12ga.sav'
;
;
block cycle 240
call 'getpp1d.fis'
fish set @jjtab=4
@getpp1d
model save 'gg12g2a.sav'
;
```

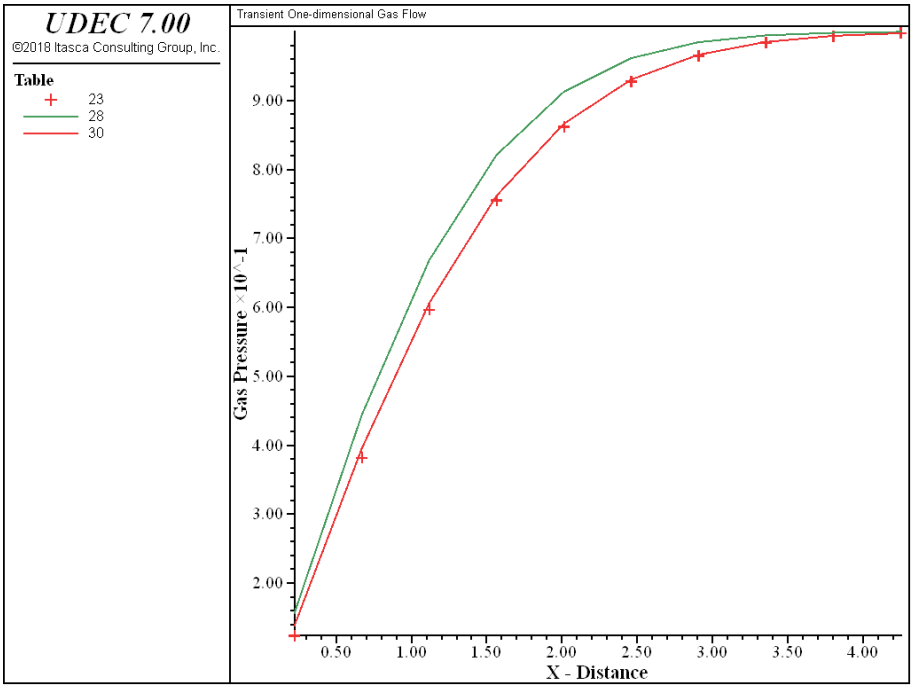

Figure 2.27 Gas pressure along joint (Ψ versus ξ) at time = 0.5

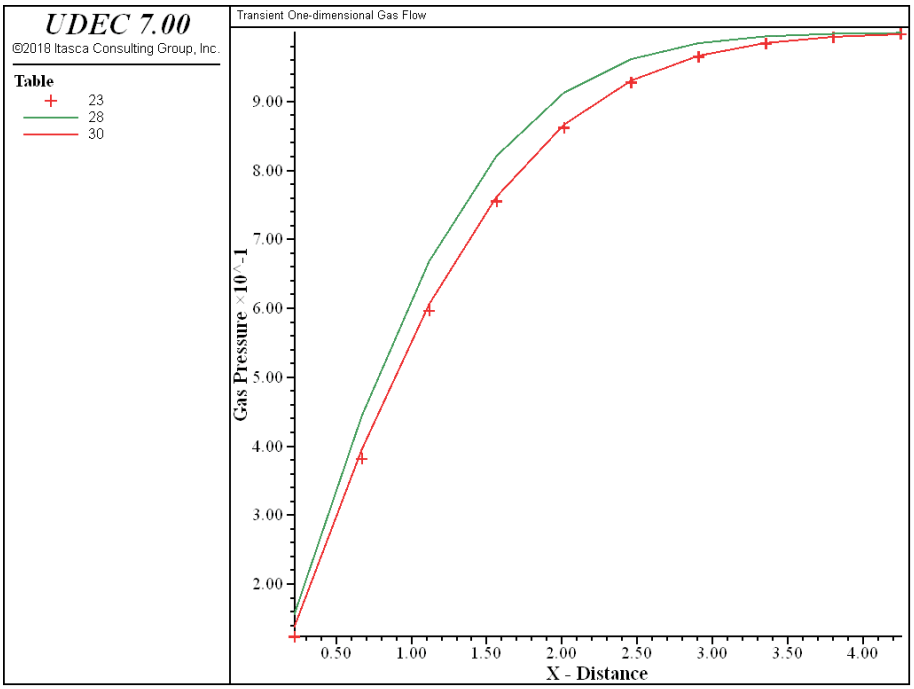

Figure 2.28 Gas pressure along joint (Ψ versus ξ) at time = 2.0

2.5.7 Filling of a Horizontal Joint – No Capillary Effects

In this two-phase flow example, the water is injected at one side (at constant pressure P_0) of a horizontal joint filled with air. The air is at atmospheric pressure. The analytical solution for the location of the interface $X(t)$ at time t is (Voller et al. 1996)

$$X(t) = \sqrt{(2\overline{P}t)}$$

where $\overline{P} = kP_0$; k is the permeability of the joint.

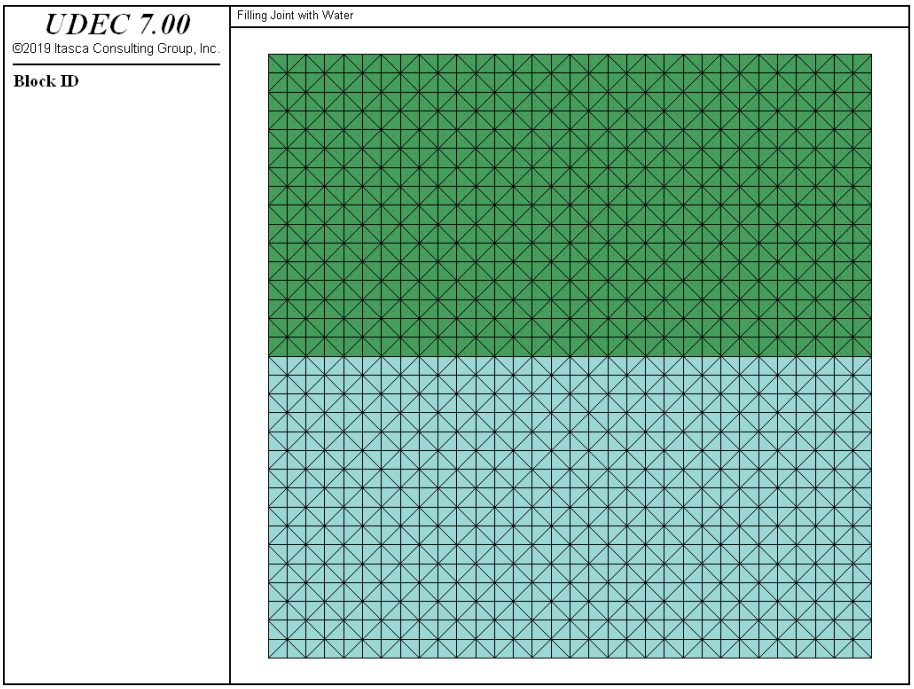

Figure 2.29 Geometry of the UDEC model

The problem is simulated using the *UDEC* model shown in Figure 2.29. The model is 2 m by 2 m in size, and has a single, horizontal joint. It was assumed that capillary forces are zero. The initial saturation of joint is 1%, and the initial pressures in both wetting and non-wetting fluid are zero. A pore pressure equal to 10 MPa is applied to the left model boundary (which is fully saturated). The following properties were used in the simulation:

a = 0.00001 m

ρ_w = 1000 kg / m^3

ρ_{nw} = 1 kg / m^3

k_{wj} = 83.3 Pa^{-1} sec^{-1}

k_{nwj} = 83.3 Pa^{-1} sec^{-1}

K_w = 2000 MPa

K_{nw} = 1 MPa

The state of the model after 1.4 seconds is illustrated in Figures 2.30 – 2.33. Comparison of the position of the fluid front as a function of time, as calculated by *UDEC* with the analytical solution, is shown in Figure 2.34. The agreement is very good.

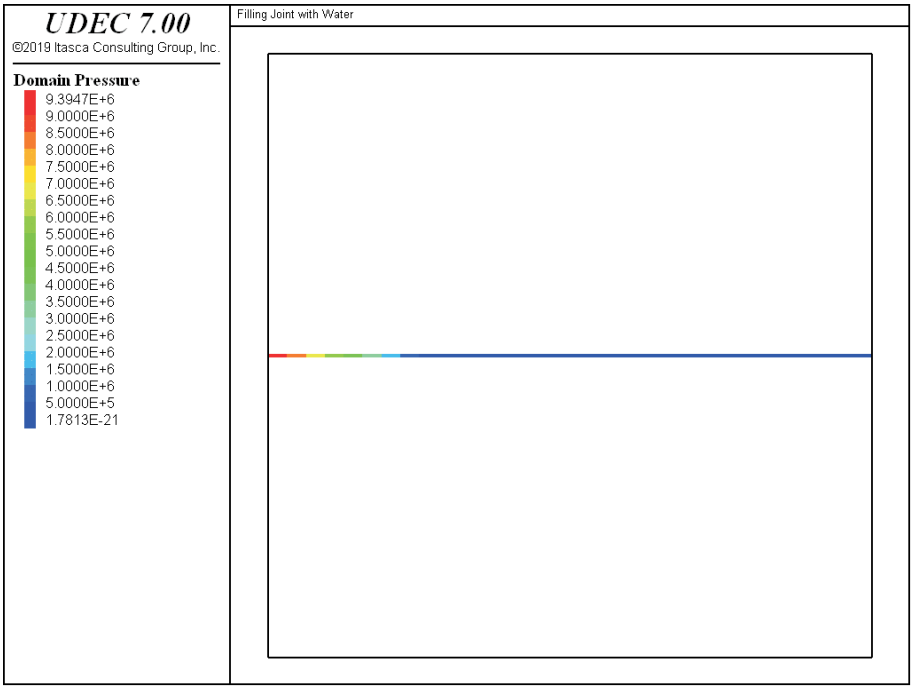

Figure 2.30 Wetting fluid pressure in the joint after 1.43 s

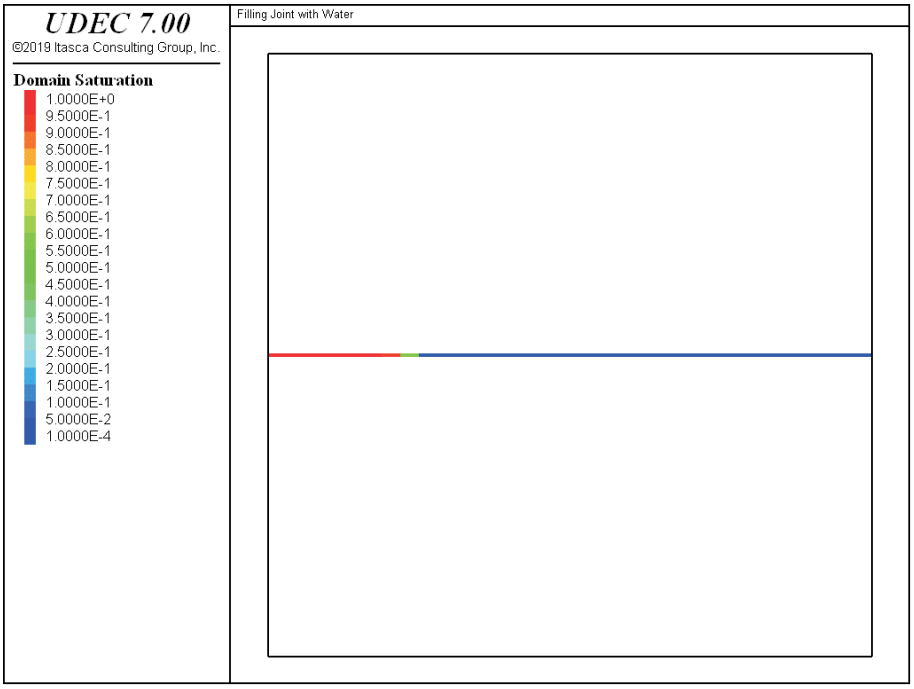

Figure 2.31 Saturation of the joint after 1.43 s

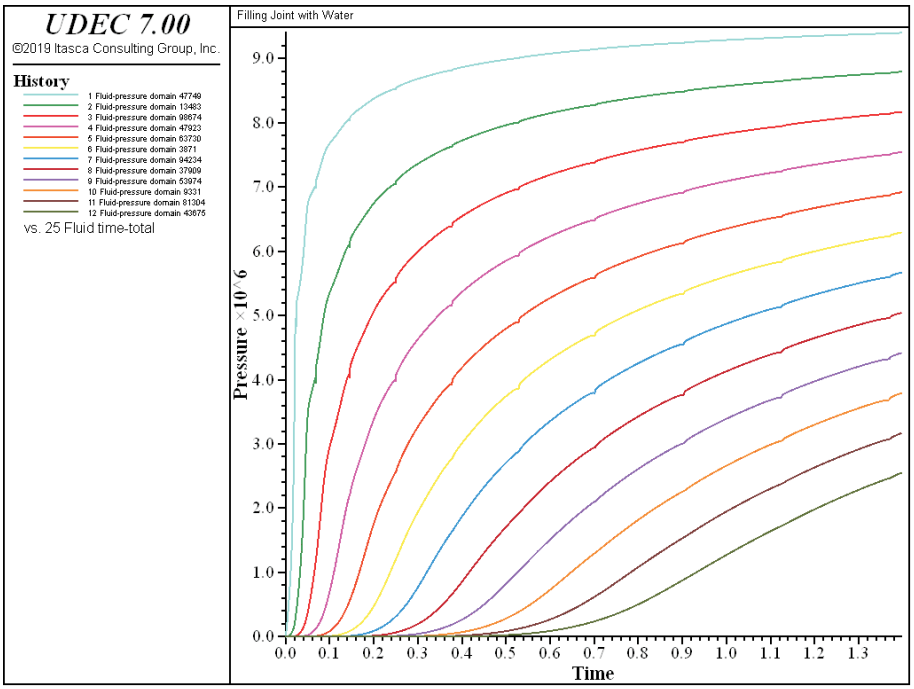

Figure 2.32 Wetting fluid pressure histories at the points along the joint

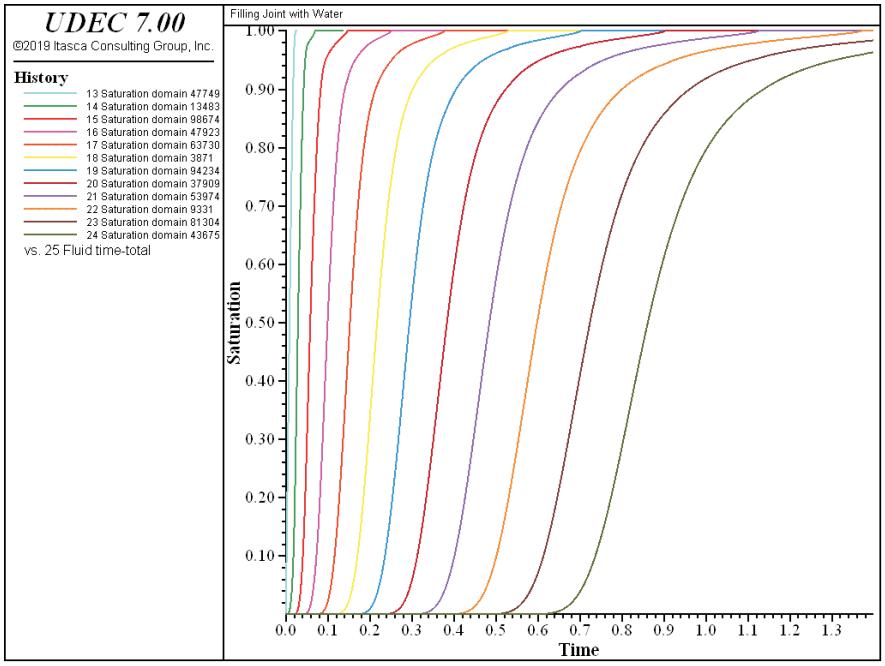

Figure 2.33 Saturation histories at the points along the joint

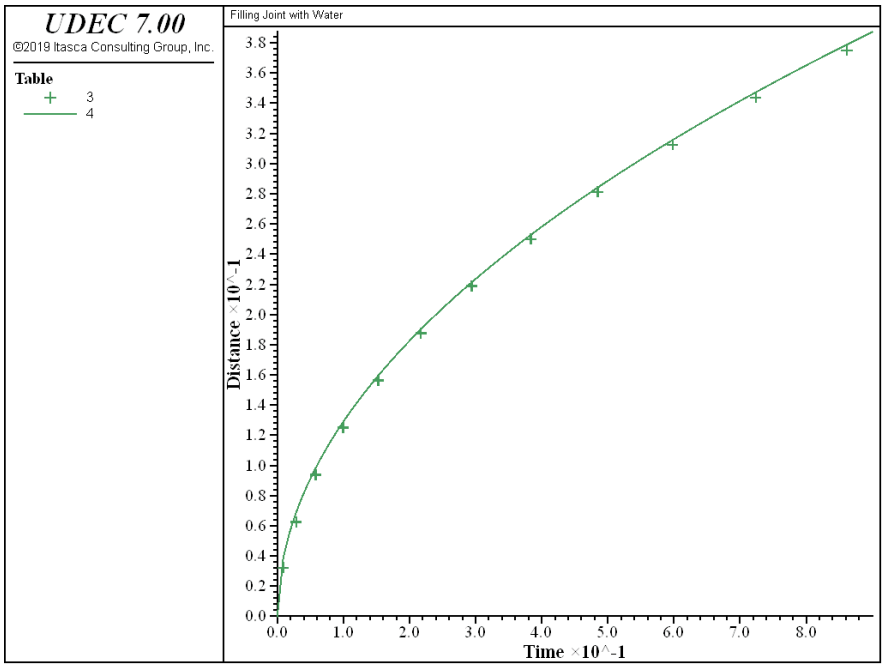

Figure 2.34 Location of fluid front (distance from the left boundary in meters) as a function of time (seconds). Comparison of the UDEC solution (crosses) with analytical solution (line).

Example 2.10 Injection of water in gas-filled joint

```
model new
;file: filljoint.dat
;--------------------------------
; Filling of joint with fluid
; No capillary pressure
;--------------------------------
model title 'Filling Joint with Water'
block config fluid
block fluid clear compressible two-phase on
block tolerance corner-round-length 2E-3
block tolerance minimum-edge-length 4E-3
block create polygon -1 -1 -1 1 1 1 1 -1
block cut crack -1 0 1 0
block zone gen edge 0.1
block zone group 'block'
block zone cmodel assign elastic density 2.5E3 bulk 1E10 shear 5E9 ...
  range group 'block'
block contact group 'joint'
block contact cmodel assign area stiffness-shear 1E11 ...
  stiffness-normal 1E11 permeability-factor 83.3 ...
  nonwetting-permeability-factor 83.3 aperture-residual 1E-5 ...
  aperture-zero-load 1E-5 range group 'joint'
; new contact default
block contact cmodel default area stiffness-shear 1E11 ...
  stiffness-normal 1E11 permeability-factor 83.3 ...
  nonwetting-permeability-factor 83.3 aperture-residual 1E-5 ...
  aperture-zero-load 1E-5
block domain change material 1
block domain property material 1 capillary-alpha 0.0001 ...
  capillary-beta 2 capillary-gamma 0
block fluid  property density 1000.0
block fluid  property bulk 2.0E9
block fluid  property nonwetting-density 1.0
block fluid  property nonwetting-bulk 1000000.0
block domain initialize saturation 0.01
block gridpoint apply velocity-x 0 range pos-x -1.01 1.01 pos-y -1.01 -0.99
block gridpoint apply velocity-y 0 range pos-x -1.01 1.01 pos-y -1.01 -0.99
block gridpoint apply velocity-y 0 range pos-x -1.01 1.01 pos-y 0.99 1.01
block gridpoint apply velocity-x 0 range pos-x -1.01 1.01 pos-y 0.99 1.01
block edge apply pore-pressure 1.0E7 ...
  range pos-x -1.01 -0.99 pos-y -1.01 1.01
block edge apply nonwetting-pressure 1.0E7 ...
  range pos-x -1.01 -0.99 pos-y -1.01 1.01
```

```
block edge apply saturation 1.0 range pos-x -1.01 -0.99 pos-y -1.01 1.01
block edge apply pore-pressure 0.0 range pos-x 0.99 1.01 pos-y -1.01 1.01
block edge apply nonwetting-pressure 0.0 ...
  range pos-x 0.99 1.01 pos-y -1.01 1.01
model gravity 0.0 -10.0
block domain history pore-pressure -0.97 0.0
block domain history pore-pressure -0.94 0.0
block domain history pore-pressure -0.91 0.0
block domain history pore-pressure -0.88 0.0
block domain history pore-pressure -0.84 0.0
block domain history pore-pressure -0.81 0.0
block domain history pore-pressure -0.78 0.0
block domain history pore-pressure -0.75 0.0
block domain history pore-pressure -0.72 0.0
block domain history pore-pressure -0.69 0.0
block domain history pore-pressure -0.66 0.0
block domain history pore-pressure -0.62 0.0
block domain history saturation -0.97 0.0
block domain history saturation -0.94 0.0
block domain history saturation -0.91 0.0
block domain history saturation -0.88 0.0
block domain history saturation -0.84 0.0
block domain history saturation -0.81 0.0
block domain history saturation -0.78 0.0
block domain history saturation -0.75 0.0
block domain history saturation -0.72 0.0
block domain history saturation -0.69 0.0
block domain history saturation -0.66 0.0
block domain history saturation -0.62 0.0
block fluid history time-total
model save 'fj1.sav'
;
;
block mechanical active off
block cycle 100000
model save 'fj2.sav'
;
;
fish define _analytic
   array xd(12)
   xd(1) = -0.97
   xd(2) = -0.94
   xd(3) = -0.91
   xd(4) = -0.88
   xd(5) = -0.84
   xd(6) = -0.81
```

```
    xd(7) = -0.78
    xd(8) = -0.75
    xd(9) = -0.72
    xd(10) = -0.69
    xd(11) = -0.66
    xd(12) = -0.62
    p_bar = 0.0833
    i = 1
    command
      history export 25 table 5
    end_command
    table(4,0.0) = 0.0
    loop j (1,100)
       _tsat = 0.009*j
       _xsat =math.sqrt(2.0*p_bar*_tsat)
       table(4,_tsat) = _xsat
    end_loop
    loop i (1,12)
       ih = i+12
       command
          table 2 clear
          hist export @ih table 2
       end_command
       _index = 0
       _s = 0
       loop while _index = 0
          _s = _s+1
          _sat = table.y(2,_s)
          if _sat >= 0.5 then
             _index = _s
             _tsat = table.y(5,_s)
             _idom = block.domain.near(xd(i),0.0)
             _xsat = 1.0+block.domain.pos.x(_idom)
          endif
       end_loop
       table(3,_tsat) = _xsat
    end_loop
end
@_analytic
model save 'fj3.sav'
return
```

2.5.8 *Filling of a Horizontal Joint by the Capillary Forces Only*

Filling of an initially air-filled joint with water driven by only capillary forces may be solved by *UDEC*. The material properties are the same as those described in Section 2.5.7, with the exception that aperture, a, is now 0.0001 m. The capillary pressure curve is defined by the following parameters:

$$a_0 = 0.0001 \text{ m}$$

$$\gamma = 10 \text{ m}$$

$$\beta = 2$$

The initial saturation inside the joint is 1%. The non-wetting fluid pressure inside the joint is initialized to zero, which implies that initial pressure in the wetting fluid is -10 MPa. Boundary conditions on the left boundary of the model were zero pressure of both wetting and non-wetting fluid, and full saturation (i.e., $s_w = 100\%$). During the simulation, the fluid from the left boundary is "sucked" inside the model by the capillary forces. The state of the model after .014 seconds is shown in Figures 2.35 – 2.38.

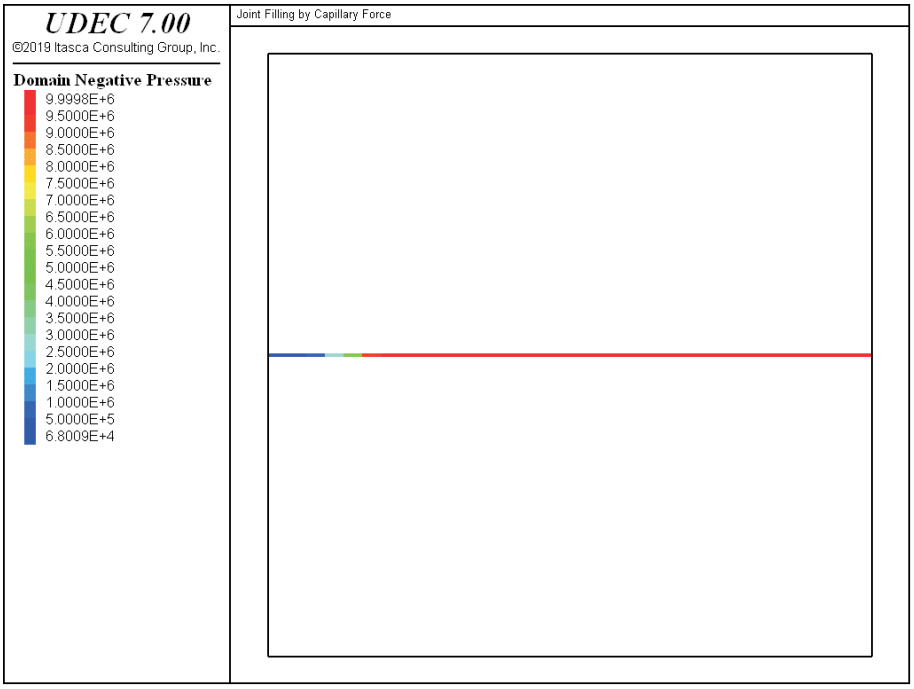

Figure 2.35 Wetting fluid pressures along the joint after .014 s

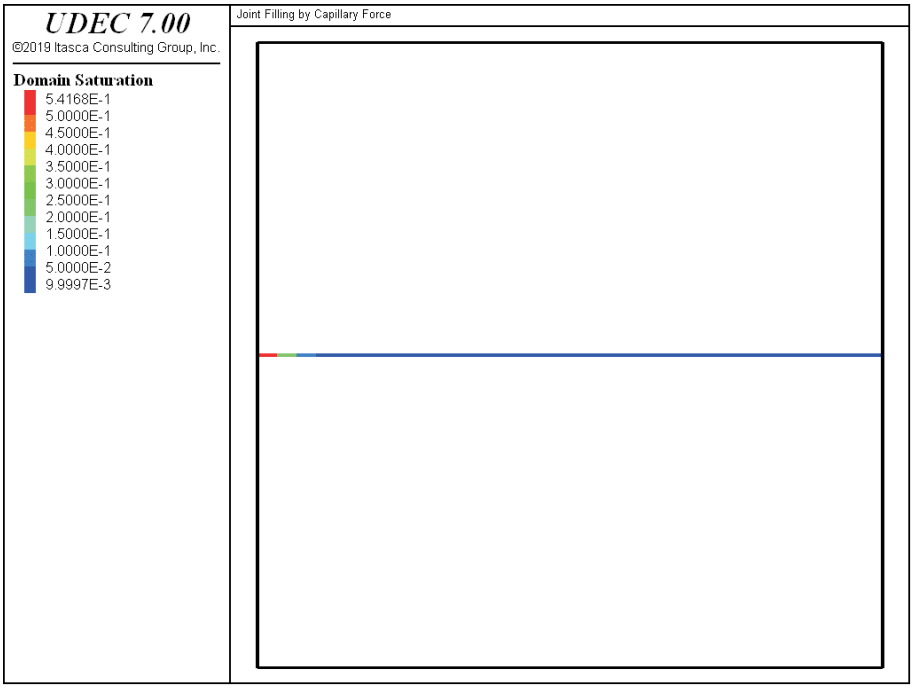

Figure 2.36 Saturation along the joint after .014 s

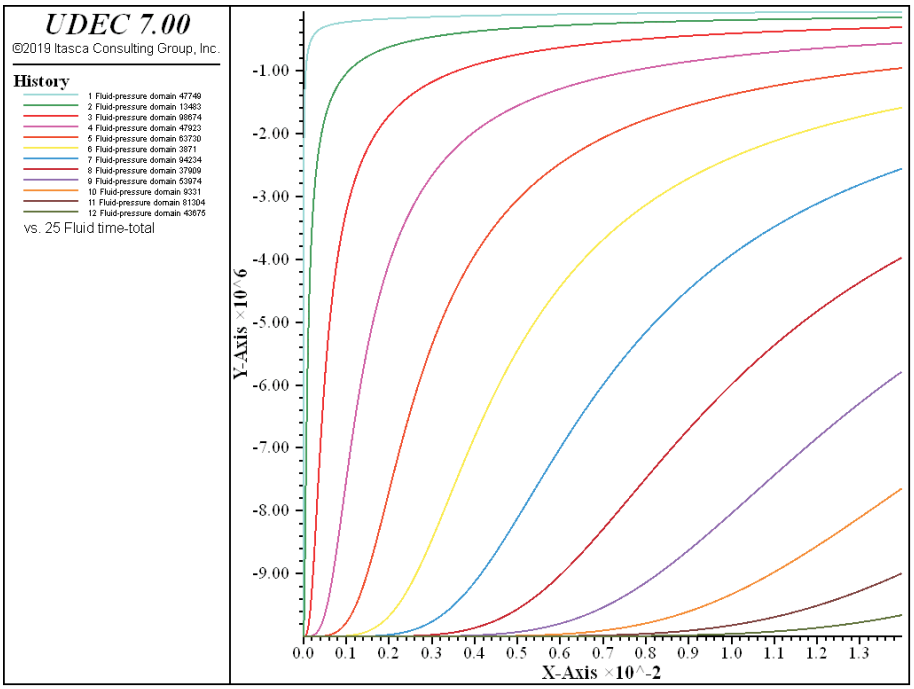

Figure 2.37 Wetting fluid pressure histories at the points along the joint

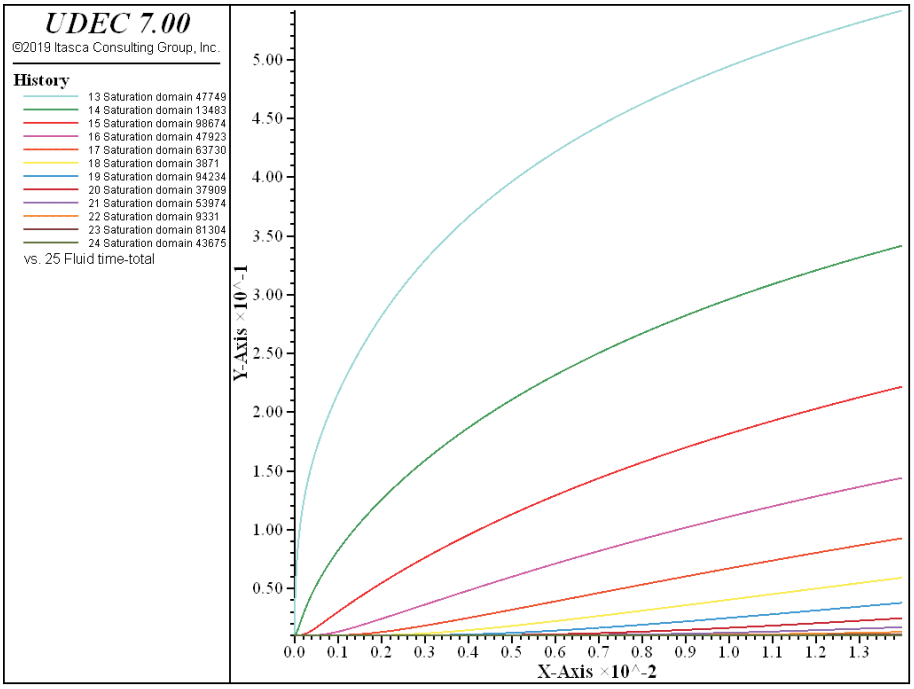

Figure 2.38 Saturation histories at the points along the joint

Example 2.11 Filling of the joint with water due to capillary forces only

```
model new
;--------------------------------
; Filling of joint with fluid
; caused by capillary forces only
;--------------------------------
model title 'Joint Filling by Capillary Force'
block config fluid
block fluid clear compressible two-phase on
block tolerance corner-round-length 2E-3
block tolerance minimum-edge-length 4E-3
block create polygon -1 -1 -1 1 1 1 1 -1
block cut crack -1 0 1 0
block zone gen edge 0.1
block zone group 'block'
block zone cmodel assign elastic density 2.5E3 bulk 1E10 shear 5E9 ...
range group 'block'
block contact group 'joint'
block contact cmodel assign area stiffness-shear 1E11 ...
  stiffness-normal 1E11 permeability-factor 83.3 ...
  nonwetting-permeability-factor 83.3 aperture-residual 1E-4 ...
  aperture-zero-load 1E-4 range group 'joint'
; new contact default
block contact cmodel default area stiffness-shear 1E11 ...
  stiffness-normal 1E11 permeability-factor 83.3 ...
  nonwetting-permeability-factor 83.3 aperture-residual 1E-4 ...
  aperture-zero-load 1E-4
block fluid  property density 1000.0
block fluid  property bulk 2.0E9
block fluid  property nonwetting-density 1.0
block fluid  property nonwetting-bulk 1000000.0
block domain change material 1
block domain property material 1 capillary-alpha 1E-4 ...
  capillary-beta 2 capillary-gamma 10
block domain initialize saturation 0.01
block domain initialize nonwetting-pressure 0.0
block gridpoint apply velocity-x 0 range pos-x -1.01 1.01 pos-y -1.01 -0.99
block gridpoint apply velocity-y 0 range pos-x -1.01 1.01 pos-y -1.01 -0.99
block gridpoint apply velocity-x 0 range pos-x -1.01 1.01 pos-y 0.99 1.01
block gridpoint apply velocity-y 0 range pos-x -1.01 1.01 pos-y 0.99 1.01
block edge apply pore-pressure 0.0 range pos-x -1.01 -0.99 pos-y -1.01 1.01
block edge apply nonwetting-pressure 0.0 ...
  range pos-x -1.01 -0.99 pos-y -1.01 1.01
block edge apply saturation 1.0 range pos-x -1.01 -0.99 pos-y -1.01 1.01
```

```
block edge apply pore-pressure -1.0E7 ...
  range pos-x 0.99 1.01 pos-y -1.01 1.01
block edge apply nonwetting-pressure 0.0 ...
  range pos-x 0.99 1.01 pos-y -1.01 1.01
block edge apply saturation 0.01 range pos-x 0.99 1.01 pos-y -1.01 1.01
model gravity 0.0 -10.0
block domain history pore-pressure -0.97 0
block domain history pore-pressure -0.94 0
block domain history pore-pressure -0.91 0
block domain history pore-pressure -0.88 0
block domain history pore-pressure -0.84 0
block domain history pore-pressure -0.81 0
block domain history pore-pressure -0.78 0
block domain history pore-pressure -0.75 0
block domain history pore-pressure -0.72 0
block domain history pore-pressure -0.69 0
block domain history pore-pressure -0.66 0
block domain history pore-pressure -0.62 0
block domain history saturation -0.97 0
block domain history saturation -0.94 0
block domain history saturation -0.91 0
block domain history saturation -0.88 0
block domain history saturation -0.84 0
block domain history saturation -0.81 0
block domain history saturation -0.78 0
block domain history saturation -0.75 0
block domain history saturation -0.72 0
block domain history saturation -0.69 0
block domain history saturation -0.66 0
block domain history saturation -0.62 0
block fluid history time-total
model save 'cfj1.sav'
;
;
block mechanical active off

block cycle 100000
model save 'cfj2.sav'
;
return
```

2.5.9 Containment of Gas Inside a Cavity – Single Horizontal Joint

A square opening, 0.4 m × 0.4 m in cross section, is pressurized by the gas to 10 MPa. The opening intersects a joint that is fully saturated with water at atmospheric pressure. The geometry of the *UDEC* model representing the stated problem is shown in Figure 2.39. The capillary forces are neglected in this problem. As the simulation is started, the non-wetting fluid enters the joint pushing the water outside (see Figures 2.40 and 2.41). The simulation of two-phase flow is fully coupled with mechanical deformation. The displacements of the model are shown in Figure 2.42. Histories of the pressure of non-wetting fluid and saturation along the joints are shown in Figures 2.43 and 2.44.

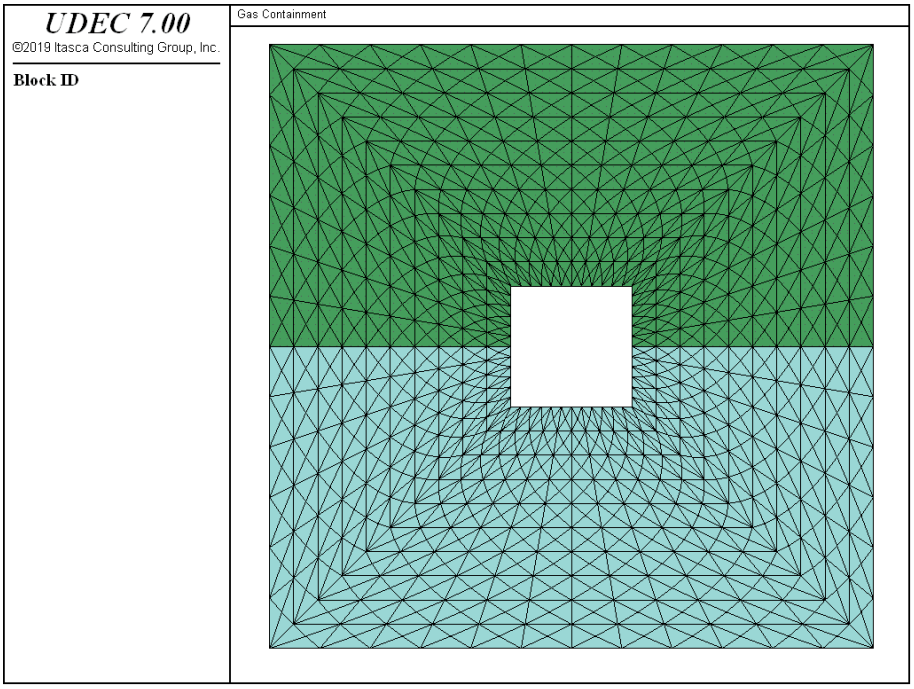

Figure 2.39 Geometry of the model

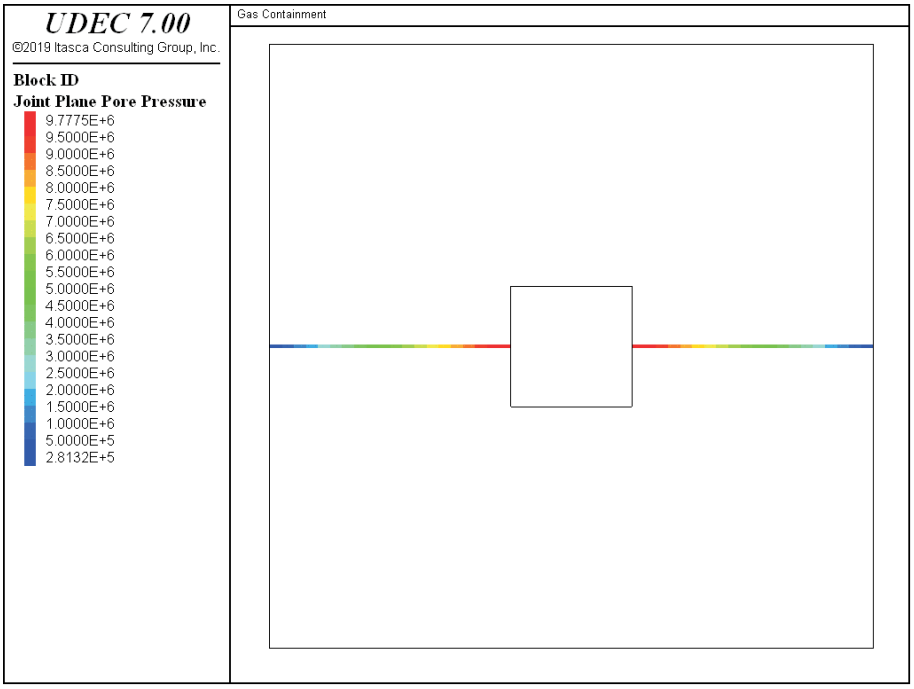

Figure 2.40 Wetting fluid pressure along the joint after .00013 s

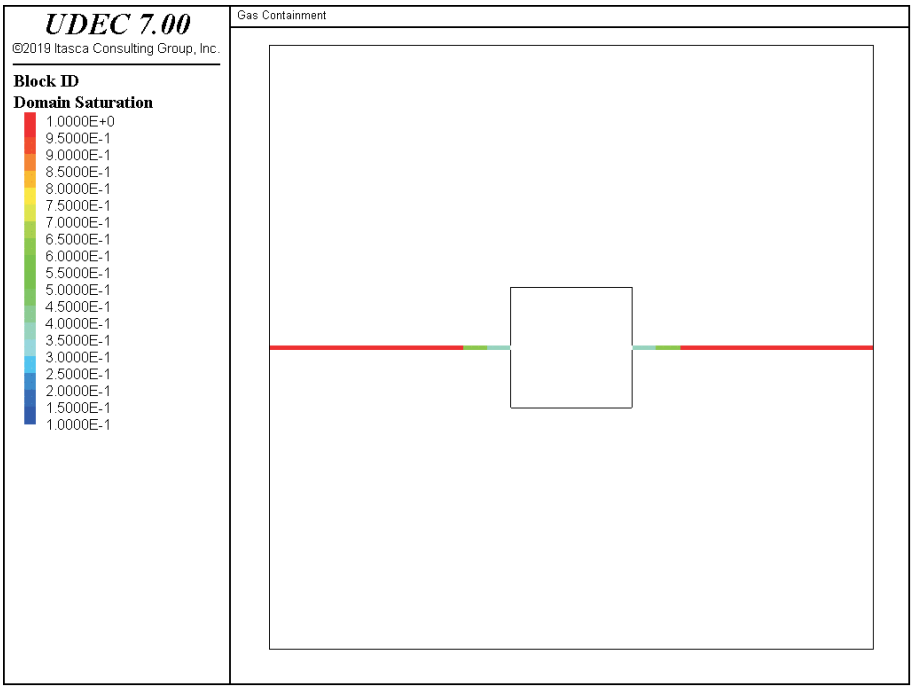

Figure 2.41 Saturation along the joint after .00013 s

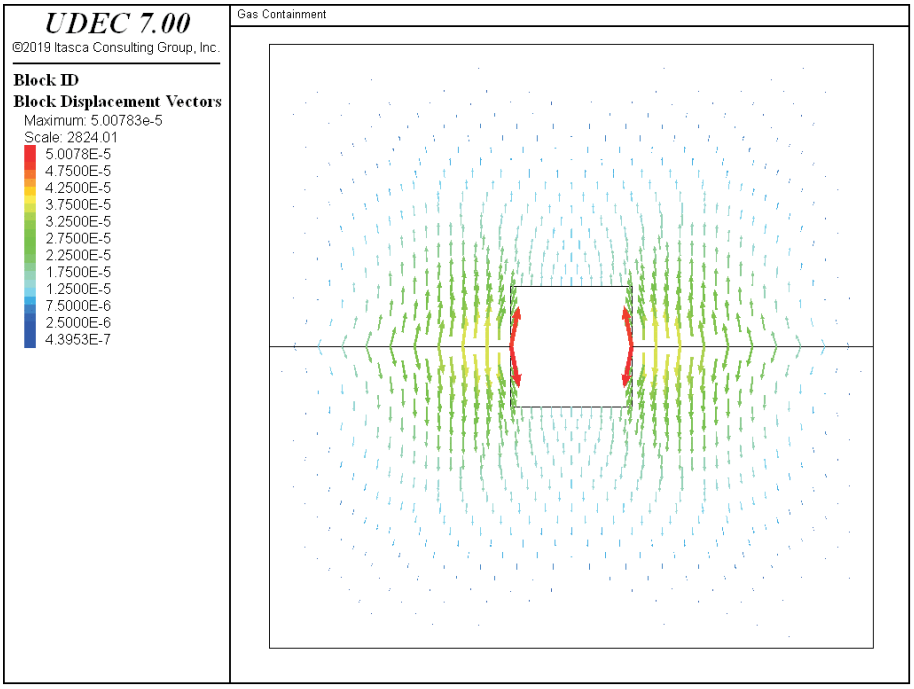

Figure 2.42 Model displacements after .00013 s

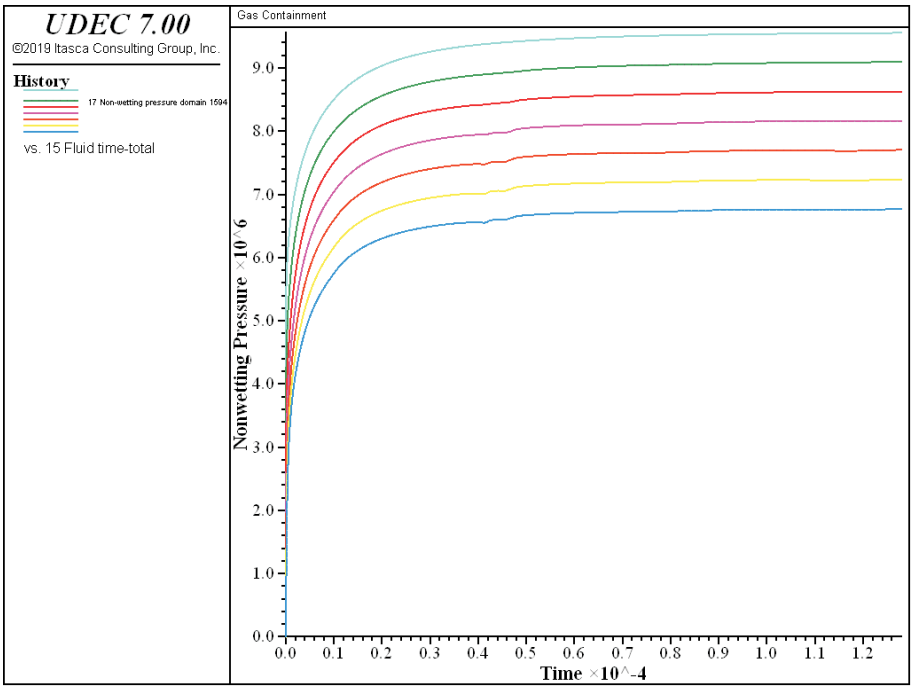

Figure 2.43 Non-wetting fluid pressure histories at the points along the joint

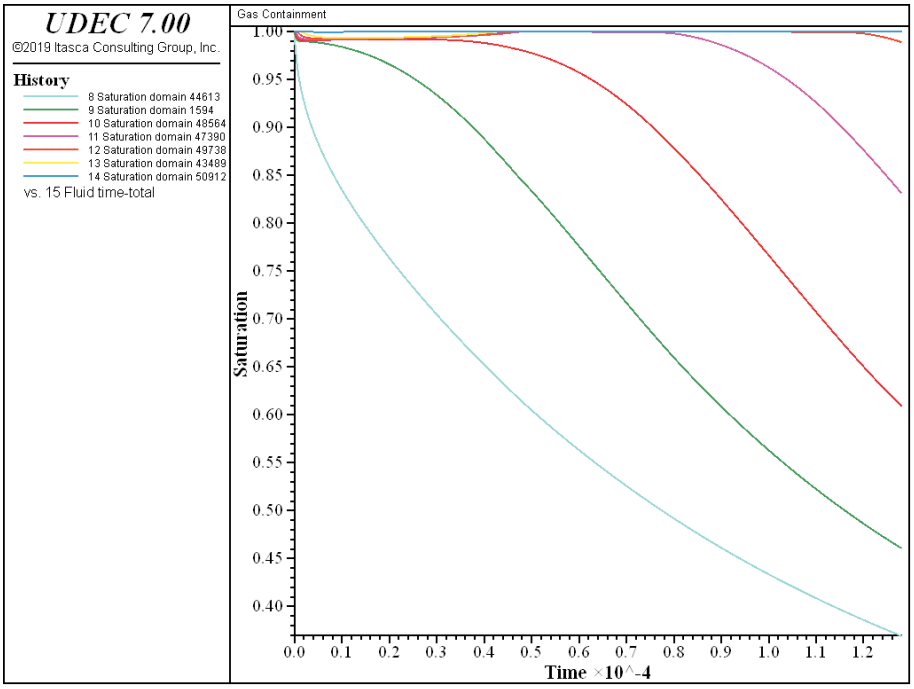

Figure 2.44 Saturation at the points along the joint

Example 2.12 Containment of gas inside a cavity – single joint

```
model new
model title 'Gas Containment'
block config fluid
block fluid clear compressible two-phase on
block tolerance corner-round-length 2E-3
block tolerance minimum-edge-length 4E-3
block create polygon -1 -1 -1 1 1 1 1 -1
block cut crack -1 0 1 0
block cut crack 0.2 -0.2 0.2 0.2
block cut crack 0.2 0.2 -0.2 0.2
block cut crack -0.2 0.2 -0.2 -0.2
block cut crack -0.2 -0.2 0.2 -0.2
block delete range pos-x -0.2 0.2 pos-y -0.2 0.2
block zone gen quad 0.1
block zone group 'block'
block zone cmodel assign elastic density 2.5E3 bulk 1E10 shear 5E9 ...
  range group 'block'
block contact group 'joint'
block contact cmodel assign area stiffness-shear 1E11 ...
  stiffness-normal 1E11 permeability-factor 83.3 ...
  nonwetting-permeability-factor 83.3 aperture-residual 1E-4 ...
  aperture-zero-load 1E-3 range group 'joint'
; new contact default
block contact cmodel default area stiffness-shear 1E11 ...
  stiffness-normal 1E11 permeability-factor 83.3 ...
  nonwetting-permeability-factor 83.3 aperture-residual 1E-4 ...
  aperture-zero-load 1E-3
block fluid  property density 1000.0
block fluid  property bulk 2.0E9
block fluid  property nonwetting-density 1.0
block fluid  property nonwetting-bulk 1000000.0
block domain change material 1
block domain property material 1 capillary-alpha 1E-4 ...
  capillary-beta 2 capillary-gamma 0
block domain initialize saturation 1.0
block domain initialize saturation 0.0 ...
  range pos-x -0.2 0.2 pos-y -0.2 0.2
block domain initialize pore-pressure 0.0
block domain fix range pos-x -0.2 0.2 pos-y -0.2 0.2
block domain initialize pore-pressure 1E7 ...
  range pos-x -0.2 0.2 pos-y -0.2 0.2
block domain fix range pos-x -0.2 0.2 pos-y -0.2 0.2
block domain initialize nonwetting-pressure 1E7 ...
```

```
  range pos-x -0.2 0.2 pos-y -0.2 0.2
block insitu  stress -1.0E7 0.0 -1.0E7
block gridpoint apply velocity-x 0
block gridpoint apply velocity-y 0
block edge apply pore-pressure 0.0 range pos-x -1.01 -0.99
block edge apply pore-pressure 0.0 range pos-x  0.99  1.01
block edge apply nonwetting-pressure 0.0 range pos-x -1.01 -0.99
block edge apply nonwetting-pressure 0.0 range pos-x 0.99 1.01
block domain history pore-pressure -0.24 0.0
block domain history pore-pressure -0.28 0.0
block domain history pore-pressure -0.32 0.0
block domain history pore-pressure -0.36 0.0
block domain history pore-pressure -0.4 0.0
block domain history pore-pressure -0.44 0.0
block domain history pore-pressure -0.48 0.0
block domain history saturation -0.24 0.0
block domain history saturation -0.28 0.0
block domain history saturation -0.32 0.0
block domain history saturation -0.36 0.0
block domain history saturation -0.4 0.0
block domain history saturation -0.44 0.0
block domain history saturation -0.48 0.0
block fluid history time-total
block domain history nonwetting-pressure -0.24 0.0
block domain history nonwetting-pressure -0.28 0.0
block domain history nonwetting-pressure -0.32 0.0
block domain history nonwetting-pressure -0.36 0.0
block domain history nonwetting-pressure -0.4 0.0
block domain history nonwetting-pressure -0.44 0.0
block domain history nonwetting-pressure -0.48 0.0
model gravity 0.0 -10.0
block fluid aperture-max-ratio 10.0
model save 'gas1.sav'
;
;
block cycle 53050
model save 'gas2.sav'
return
```

2.5.10 Containment of Gas Inside a Cavern – Jointed Rock Mass

Containment of a gas inside a cavern in a jointed rock mass is modeled using *UDEC*. The geometry of the model is shown in Figure 2.45. The same material properties as in the example from Section 2.5.9 are used in this problem. The size of the cavern is 4.2 m × 4.2 m. The joints in the rock mass are initially fully saturated with water. The pressure in the water is equal to atmospheric pressure. The non-wetting fluid pressure inside the cavern is equal to 10 MPa. The state of the model, after 0.25 seconds, as gas pushes out the water is illustrated in Figures 2.46 and 2.47.

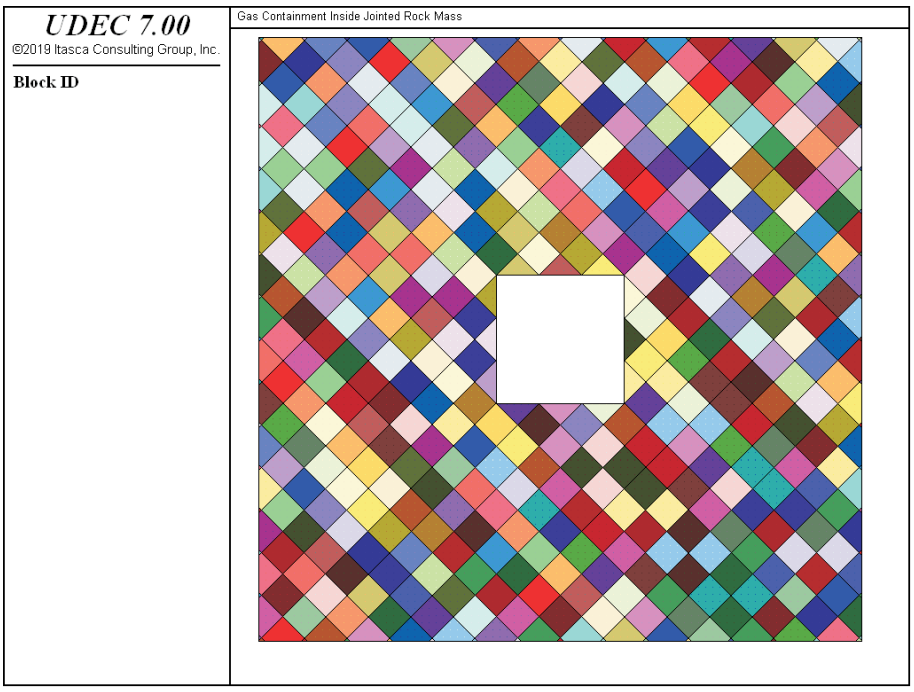

Figure 2.45 Geometry of the model

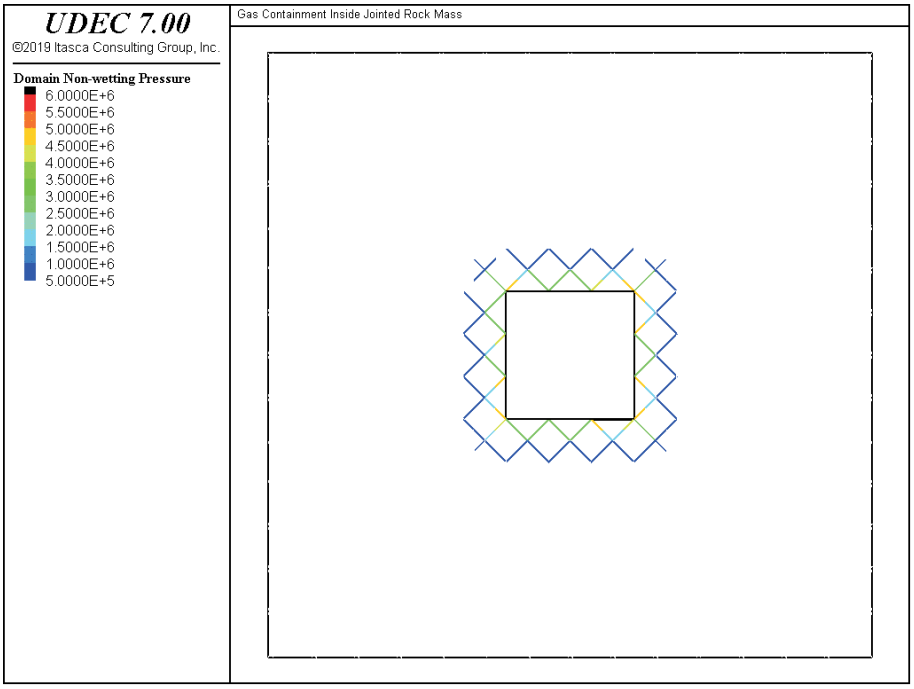

Figure 2.46 Non-wetting fluid pressures after 0.25 s

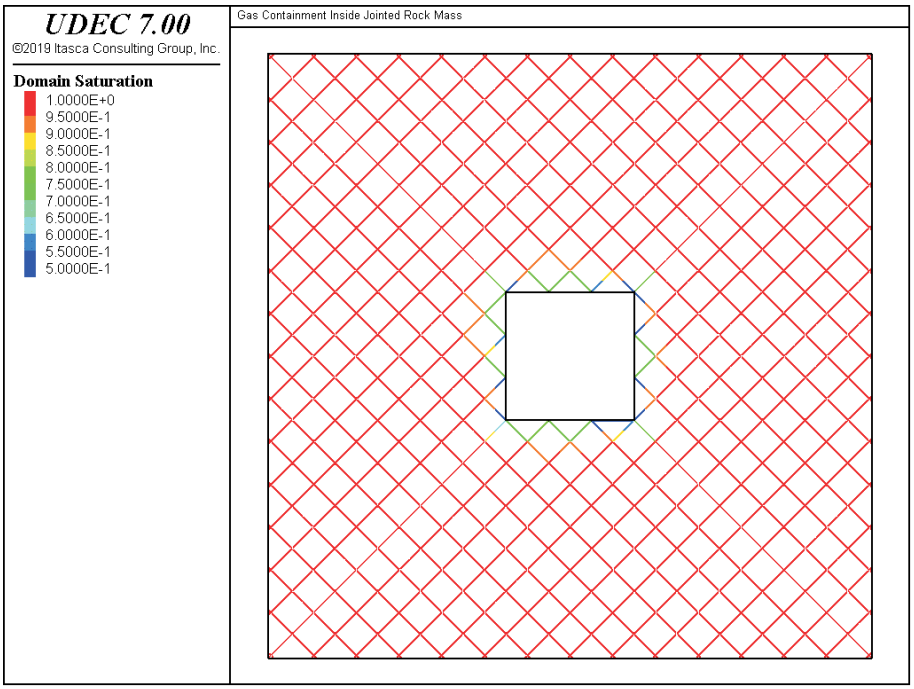

Figure 2.47 Saturation of the joints in the model after 0.25 s

Example 2.13 Containment of gas inside a cavity – jointed rock mass

```
model new
model title 'Gas Containment Inside Jointed Rock Mass'
block config fluid
block fluid clear compressible two-phase on
block tolerance corner-round-length 2E-2
block tolerance minimum-edge-length 4E-2
block create polygon -10 -10 -10 10 10 10 10 -10
block cut joint-set angle 45 spacing 1 origin 0 0
block cut joint-set angle 135 spacing 1 origin 0 0
block cut crack 2.12 -2.12 2.12 2.12
block cut crack 2.12 2.12 -2.12 2.12
block cut crack -2.12 2.12 -2.12 -2.12
block cut crack -2.12 -2.12 2.12 -2.12
block delete range pos-x -2.12 2.12 pos-y -2.12 2.12
block delete range area 0.1
block zone gen edge 1.0
block zone group 'block'
block zone cmodel assign elastic density 2.5E3 bulk 1E10 ...
  shear 5E9 range group 'block'
block contact group 'joint'
block contact cmodel assign area stiffness-shear 1E11 ...
  stiffness-normal 1E11 cohesion 1E10 tension 1E10 ...
  permeability-factor 83.3 nonwetting-permeability-factor 83.3 ...
  aperture-residual 1E-4 aperture-zero-load 1E-4 range group 'joint'
; new contact default
block contact cmodel default area stiffness-shear 1E11 ...
  stiffness-normal 1E11 cohesion 1E10 tension 1E10 ...
  permeability-factor 83.3 nonwetting-permeability-factor 83.3 ...
  aperture-residual 1E-4 aperture-zero-load 1E-4
block fluid  property density 1000.0
block fluid  property bulk 2.0E9
block fluid  property nonwetting-density 1.0
block fluid  property nonwetting-bulk 1000000.0
block domain change material 1
block domain property material 1 capillary-alpha 1E-4 capillary-beta 2 ...
  capillary-gamma 0
block domain initialize saturation 1.0
block domain initialize saturation 0.0 ...
  range pos-x -2.12 2.12 pos-y -2.12 2.12
block domain initialize pore-pressure 0.0
block domain fix range pos-x -2.12 2.12 pos-y -2.12 2.12
block domain initialize pore-pressure 1E7 ...
  range pos-x -2.12 2.12 pos-y -2.12 2.12
```

```
block domain fix range pos-x -2.12 2.12 pos-y -2.12 2.12
block domain initialize nonwetting-pressure 1E7 ...
  range pos-x -2.12 2.12 pos-y -2.12 2.12
block insitu  stress -1.0E7 0.0 -1.0E7
block gridpoint apply velocity-x 0
block gridpoint apply velocity-y 0
block edge apply pore-pressure 0.0 range pos-x -10.1 -9.8
block edge apply nonwetting-pressure 0.0 range pos-x -10.1 -9.8
block edge apply pore-pressure 0.0 range pos-x 9.8 10.1
block edge apply nonwetting-pressure 0.0 range pos-x 9.8 10.1
block edge apply pore-pressure 0.0 range  pos-y -10.1 -9.8
block edge apply nonwetting-pressure 0.0 range  pos-y -10.1 -9.8
block edge apply pore-pressure 0.0 range  pos-y 9.8 10.1
block edge apply nonwetting-pressure 0.0  pos-y 9.8 10.1
model gravity 0.0 -10.0
model save 'joint1.sav'
;
;
block domain history saturation 2.2 0.0
block domain history saturation 2.5 0.0
block cycle 28925
model save 'joint2.sav'
;
;
```

2.5.11 Thermal-Mechanical-Fluid Flow Example

This simple example shows the effect of a heat source on a saturated system initially at equilibrium. Figure 2.48 shows the jointed system with two orthogonal joint sets spaced 5 m apart. The properties are as follows.

Elastic Blocks

mass density	2500 kg/m^3
bulk modulus	6666 MPa
shear modulus	4000 MPa
thermal conductivity	5 W/m °C
thermal expansion coefficient	10^{-5} (1/C°)
specific heat	900 J/kg °C

Coulomb Joints

normal stiffness	50,000 MPa/m
shear stiffness	10,000 MPa/m
friction angle	30°
cohesion	10 MPa
residual aperture	20 μm
aperture at zero stress	68 μm

The fluid density is supposed to vary from 1000 kg/m^3 at 0°C (initial temperature) to 800 kg/m^3 at 100°C.

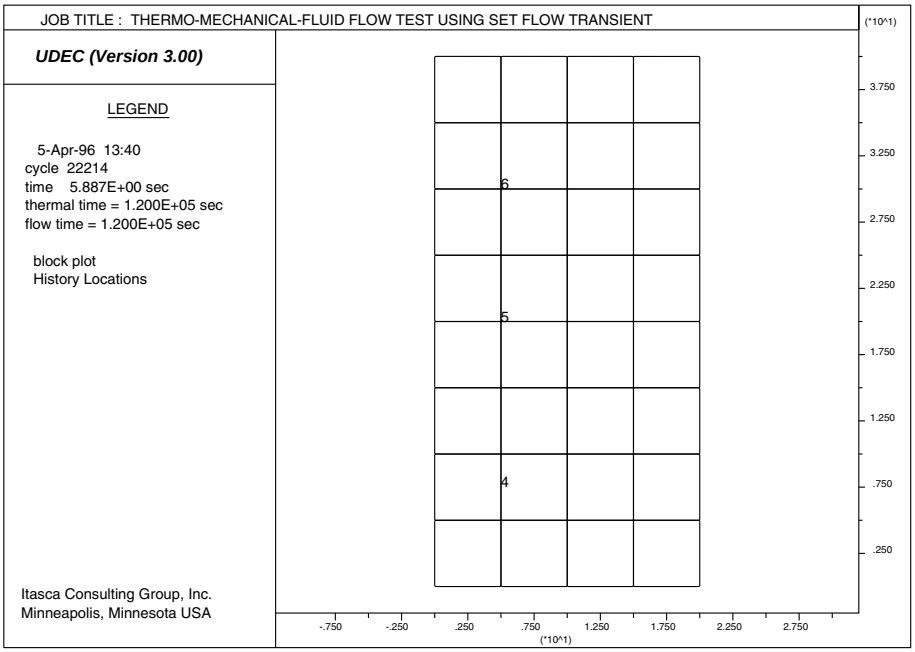

Figure 2.48 UDEC model for thermal-mechanical-fluid flow example

The model is subjected to the gravitational stress state (units in Pa)

$$\sigma_{xx} = -15 \times 10^5 + (0.125 \times 10^5)y$$

$$\sigma_{yy} = -30 \times 10^5 + (0.25 \times 10^5)y$$

The bottom boundary is fixed in the y-direction, and the side boundaries are fixed in the x-direction. The bottom and side boundaries are impermeable, and the model is fully saturated, with the water table 10 m above the top of the model. Adiabatic conditions are specified for the bottom and side boundaries, and a constant heat source of 100 kW/m is applied along the top boundary for 6000 seconds; thereafter, the boundary is adiabatic.

The problem is solved with both the steady-state and transient (incompressible fluid) flow modes. Figures 2.49 and 2.50 show the fluid pressure history at three points in the model (see Figure 2.48). When the rock is heated, large thermal stresses are generated and compress the joints. This causes a transient increase of water pressures. However, once these extra pressures have dissipated, the only remaining effect on the fluid is the decrease in its density. Thus, fluid pressures stabilize at a level lower than their initial values.

The steady-state flow algorithm directly reproduces the long-term behavior without going through the early time increase, whereas the transient flow algorithm shows the complete sequence. (Compare Figures 2.49 and 2.50.) The data files for these two cases are given in Examples 2.14 and 2.15.

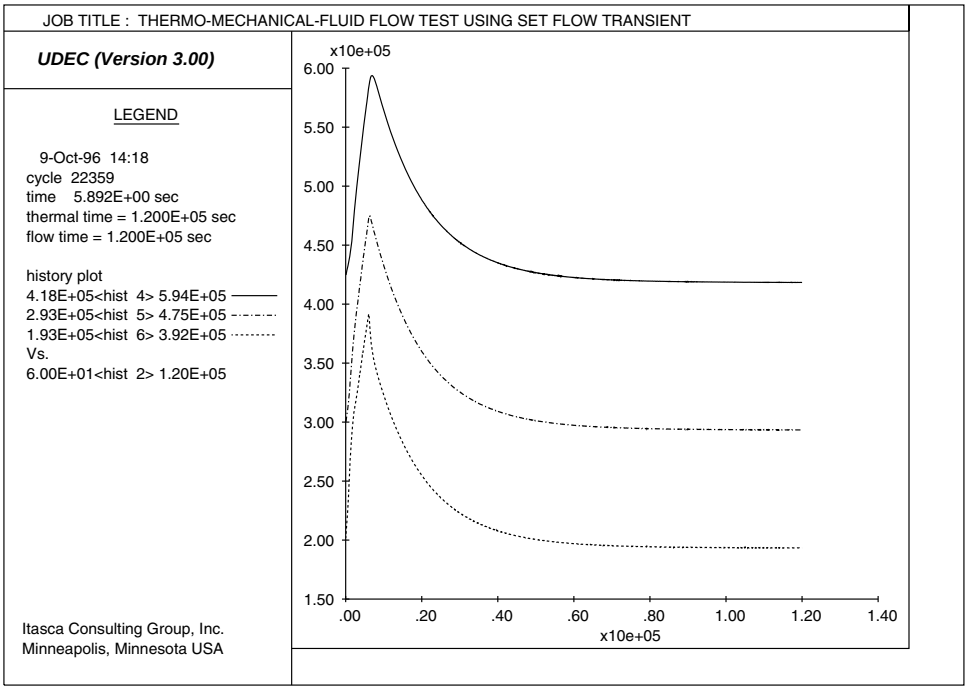

Figure 2.49 Fluid pressure versus thermal time histories for transient flow analysis

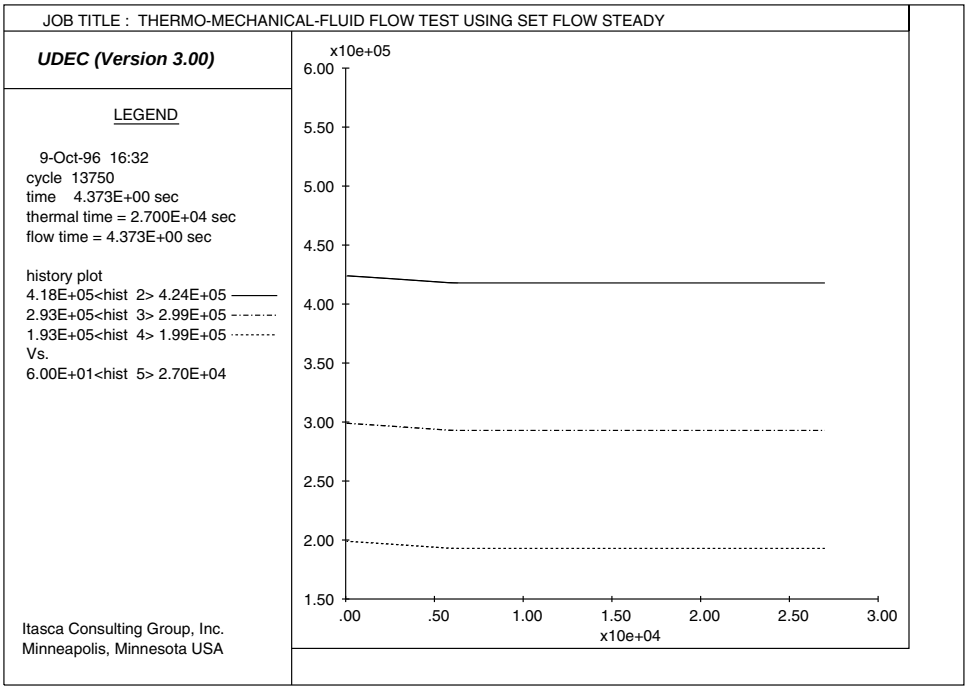

Figure 2.50 Fluid pressure versus thermal time histories for steady-state flow analysis

Example 2.14 Thermal-mechanical-fluid flow with incompressible transient flow

```
config thermal fluid
set flow clear incompressible on
round 0.1
edge 0.2
block 0,0 0,40 20,40 20,0
jset angle 90 spacing 5 origin 0,0
jset angle 0 spacing 5 origin 0,0
gen edge 10.0
;  elastic block (E=100000bars, nu=.25)
group zone 'block'
zone model elastic density 2.5E3 bulk 6.6667E9 shear 4E9 cond 5 specheat &
 900 thexp 1E-5 range group 'block'
;  Coulomb joints
group joint 'joint'
joint model area jks 1E10 jkn 5E10 jfriction 30 jcohesion 1E7 jperm 83 &
 ares 0.00002 azero 6.8E-5 range group 'joint'
; new contact default
set jcondf joint model area jks=1E10 jkn=5E10 jfriction=30 jcohesion=1E7 &
 jperm=83 ares=2E-5 azero=6.8E-5
; water
fluid density=1000.0
fluid dtable=1
table 1 delete
table 1 0 1E3 100 800
; boundary conditions
bou stress -10e5 0 -20e5 pp 10e4 range -.1 20.1 39.9 40.1
bou xvel=0 imperm                range -.1 .1 -.1 40.1
bou xvel=0 imperm                range 19.9 20.1 -.1 40.1
bou yvel=0 imperm                range -.1 20.1 -.1 .1
; gravity and initial stresses
set gravity=0 -10
insitu stres -15e5 0 -30e5 ygrad .125e5 0 .25e5 ywtable 50.
his flowtime
hist pp 5,7.5
hist pp 5,20
hist pp 5,30
hist ydisp 5,40
history ncyc 1
history ntcyc 1
set capratio=50.0
;  incompressible flow parameters
set dtflow=60.0
set voltol=0.001
```

```
; equilibrate without heat
cycle  3000
save thm_inc1.sav
;
reset  hist time
set ftime=0
history ncyc 1
history ntcyc 1
his flowtime
hist pp 5,7.5
hist pp 5,20
hist pp 5,30
hist ydisp 5,40
history temperature 10.0,40.0
history temperature 10.0,35.0
history temperature 10.0,30.0
history thtime
; heat source at top
thapp  flux 1e5 0  range -.1 20.1 39. 41.
set nmech=1
set ntherm=1
set thdt=60.0
run  temp 100 step 100
save thm_inc2.sav
; stop heat source and let pressures stabilize
thapp  flux -1e5 0  range -.1 20.1 39. 41.
run step 1900
save thm_inc3.sav
;
;*** plot commands ****
;plot name: unbal force
plot hold history 1 line
;plot name: pressure hist
plot hold history 3 4 5 vs 10
;plot name: ydisp hist
plot hold history 6 line
ret
```

Example 2.15 Thermal-mechanical-fluid flow assuming steady-state flow

```
config thermal fluid
set flow steady on
round 0.1
edge 0.2
block 0,0 0,40 20,40 20,0
jset angle 90 spacing 5 origin 0,0
jset angle 0 spacing 5 origin 0,0
gen edge 10.0
;  elastic block (E=100000bars, nu=.25)
group zone 'block'
zone model elastic density 2.5E3 bulk 6.6667E9 shear 4E9 cond 5 specheat &
 900 thexp 1E-5 range group 'block'
;  Coulomb joints
group joint 'joint'
joint model area jks 1E10 jkn 5E10 jfriction 30 jcohesion 1E7 jperm 83 &
 ares 0.00002 azero 6.8E-5 range group 'joint'
; new contact default
set jcondf joint model area jks=1E10 jkn=5E10 jfriction=30 jcohesion=1E7 &
 jperm=83 ares=2E-5 azero=6.8E-5
; water
fluid density=1000.0
fluid dtable=1
table 1 delete
table 1 0 1E3 100 800
; boundary conditions
bou stress -10e5 0 -20e5 pp 10e4 range -.1 20.1 39.9 40.1
bou xvel=0 imperm                range -.1 .1 -.1 40.1
bou xvel=0 imperm                range 19.9 20.1 -.1 40.1
bou yvel=0 imperm                range -.1 20.1 -.1 .1
; gravity and initial stresses
set gravity=0 -10
insitu stres -15e5 0 -30e5 ygrad .125e5 0 .25e5 ywtable 50.
his flowtime
hist pp 5,7.5
hist pp 5,20
hist pp 5,30
hist ydisp 5,40
history ncyc 1
history ntcyc 1
set capratio=50.0
; equilibrate without heat
solve ratio 1e-6
save thm_ss1.sav
```

```
;
reset  hist time
set ftime=0
history ncyc 1
history ntcyc 1
his flowtime
hist pp 5,7.5
hist pp 5,20
hist pp 5,30
hist ydisp 5,40
history temperature 10.0,40.0
history temperature 10.0,35.0
history temperature 10.0,30.0
hist thtime
; heat source at top
thapp  flux 1e5 0  range -.1 20.1 39. 41.
set nmech=30
set ntherm=1
set thdt=60.0
run  temp 100 step 100
save thm_ss2.sav
; stop heat source and let pressures stabilize
thapp  flux -1e5 0  range -.1 20.1 39. 41.
run step 350
save thm_ss3.sav

;*** plot commands ****
;plot name: unbal force
plot hold history 1 line
;plot name: pressure hist
plot hold history 3 4 5 vs 10
;plot name: ydisp hist
plot hold history 6 line
ret
```

2.6 References

Abdallah, G., et al. "Thermal Convection of Fluid in Fractures Media," *Int. J. Rock Mech. Min Sci. & Geomech. Abstr.*, ***32***(5), 481-490 (1995).

Barton, N., S. Bandis and K. Bakhtar. "Strength, Deformation and Conductivity Coupling of Rock Joints," *Int. J. Rock Mech. Min. Sci. & Geomech. Abstr.*, ***22***(3), 121-140 (1985).

Carslaw, H. S., and J. C. Jaeger. ***Conduction of Heat in Solids***, 2nd Ed. Oxford: Clarendon Press (1959).

Chan, D. Y. C., B. D. Hughes and L. Paterson. "Transient gas flow around boreholes," *Transport in Porous Media*, ***10***, 137-152 (1993).

Detournay, E., and A. H.-D. Cheng. "Fundamentals of Poroelasticity," in ***Comprehensive Rock Engineering***, Vol. 2, pp. 113-171. J. Hudson et al., eds. London: Pergamon Press (1993).

Hardy, M. P., and M. J. Asgian. "Fracture Reopening during Hydraulic Fracturing Stress Determinations," *Int. J. Rock Mech. Min Sci. & Geomech. Abstr.*, ***26***(6), 489-497 (1989).

Harr, M. E. ***Groundwater and Seepage***. New York: McGraw-Hill Book Company (1962).

Itasca Consulting Group Inc. ***UDEC (Universal Distinct Element Code)***, Version 6.0. Minneapolis: ICG (2014).

Louis, C. "A Study of Groundwater Flow in Jointed Rock and Its Influence on the Stability of Rock Masses," Imperial College, Rock Mech. Research Report No. 10 (1969).

Pruess, K., and Y. W. Tsang. "On Two-Phase Relative Permeability and Capillary Pressure of Rough-Walled Rock Fractures," *Water Resources Research*, ***26***(9), 1915-1926 (September 1990).

Voller, V. R., S. Peng and Y. F. Chen. "Numerical Solution of Transient, Free Surface Problems in Porous Media," *Int. J. Num. Meth. Eng.*, ***39***, 2889-2906 (1996).

Wilkinson, W. L. ***Non-Newtonian Fluids: Fluid Mechanics, Mixing and Heat Transfer***. London: Pergamon Press (1960).

Witherspoon, P. A., et al. "Validity of Cubic Law for Fluid Flow in a Deformable Rock Fracture," *Water Resources Research*, ***16***(6), 1016-1024 (1980).