14 Hydraulic Fracturing Simulation

14.1 Problem Statement

This is an example of using *UDEC* to simulate pressurized cracks and hydraulic fracturing. It documents the results of numerical simulations for fluid injection performed with *UDEC*. Two cases are presented. In Case 1, a uniform fluid pressure is applied inside a planar crack. The displacements of the crack surface are compared with an exact analytical solution. For Case 2, a viscous fluid is injected at a constant rate into a planar crack with zero toughness. The *UDEC* results for Case 2 are compared with both a zero toughness solution (Adachi and Detournay 2002) and a displacement discontinuity (DD) numerical solution.

14.1.1 Data for the Simulations

The medium is assumed to be elastic, Young's modulus is 40 GPa and Poisson's ratio is 0.22. The crack is straight with a length of 21.6 m.

The x-axis of reference is oriented along the fracture, with the origin located at mid-length. The *UDEC* sign convention is used in this section, whereby tension and extension are positive for the rock matrix. Joint opening is also positive. However, joint normal stress and fluid pressure are positive in compression.

The initial stress state, $\sigma_{yy} = -15$ MPa and $\sigma_{xx} = -30$ MPa, is applied. For Case 1, the fluid pressure is uniform and equal to 20 MPa. For Case 2, the injection rate is 0.0004 m^2/s, and fluid viscosity is 0.001 Pa $\cdot$ sec. The medium is initially dry. A total of 10 seconds on injection is considered for the numerical simulations, with intermediate results at 2.5, 5 and 7.5 seconds.

Table 14.1 Data for the simulations

case	E [MPa]	ν	σ_{yy} [MPa]	σ_{xx} [MPa]	a [m]	p [MPa]	Q [m$^{2/s}$]	μ [Pa s]
1	40,000	0.22	−15	−30	10.8	20	–	–
2	40,000	0.22	−15	−30	–	–	0.0004	0.001

14.2 *UDEC* Analyses

14.2.1 Case 1

The analysis for Case 1 considers the mechanical influence on stress and deformation of a uniform pressure being applied along a 21.6 meter crack. The fluid is simulated by specifying a domain pressure inside the crack.

14.2.1.1 UDEC Model

The *UDEC* model domain is 46.08 m by 46.08 m, with a zone size of 1.44 m. The matrix is elastic, with a Young's modulus of 40,000 MPa and a Poisson's ratio of 0.22. The embedded crack is modeled by preventing the ends of a throughgoing joint (located at mid-height in the model) from opening or sliding. This is achieved by assigning high strength properties to the contacts at the ends, and a value of joint stiffness equal to about 10 times the apparent stiffness of neighboring zones (see Eq. (3.1) in the **User's Guide**). The in-situ stresses are specified in the model, and a boundary element representation is selected for the far field. A uniform domain fluid pressure is assigned inside the crack. The *UDEC* model and fluid pressure in the crack are shown in Figure 14.1.

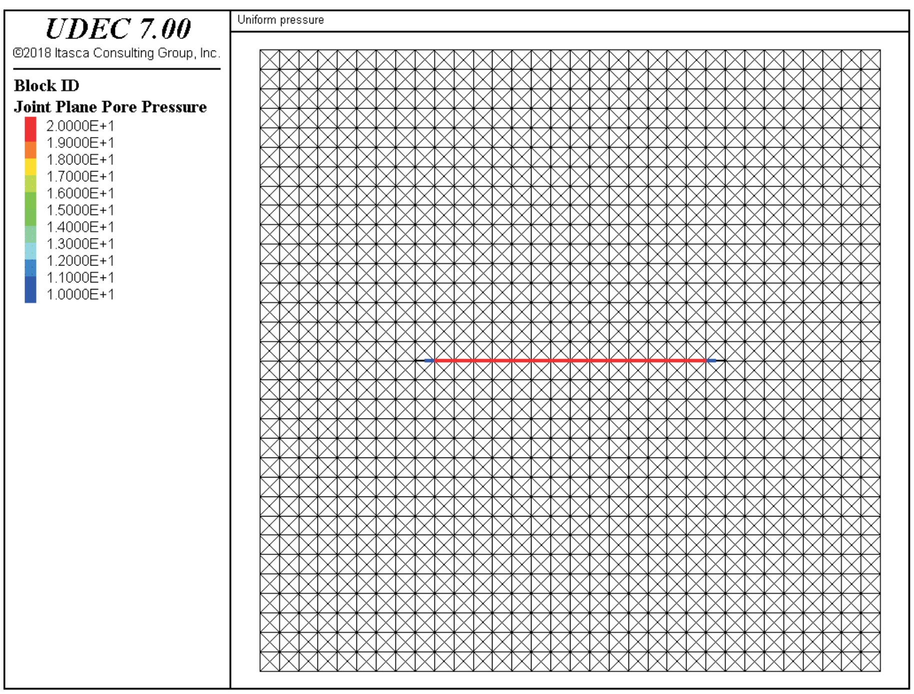

Figure 14.1 UDEC model – finite difference zones and fluid pressure in the crack

Note that the crack tip is assumed to be located at the midpoint between "glued" and "unglued" nodes at the ends of the crack. Hence, the fracture being modeled is 21.6 m long and spans 14 zones. Fluid flow is turned off and the model is cycled to mechanical equilibrium. The data file for the simulation is provided in Example 14.1.

14.2.1.2 Analytical Solution

The analytical solution for fracture opening is given by (see, e.g., Parker 1981)

$$w = \frac{|\sigma_{yy} + p|\, 4(1 - v^2)}{E} \sqrt{a^2 - x^2} \tag{14.1}$$

where w is fracture opening.

14.2.1.3 UDEC Results

The *UDEC* results for fracture opening are compared to the analytical prediction in Figure 14.2.

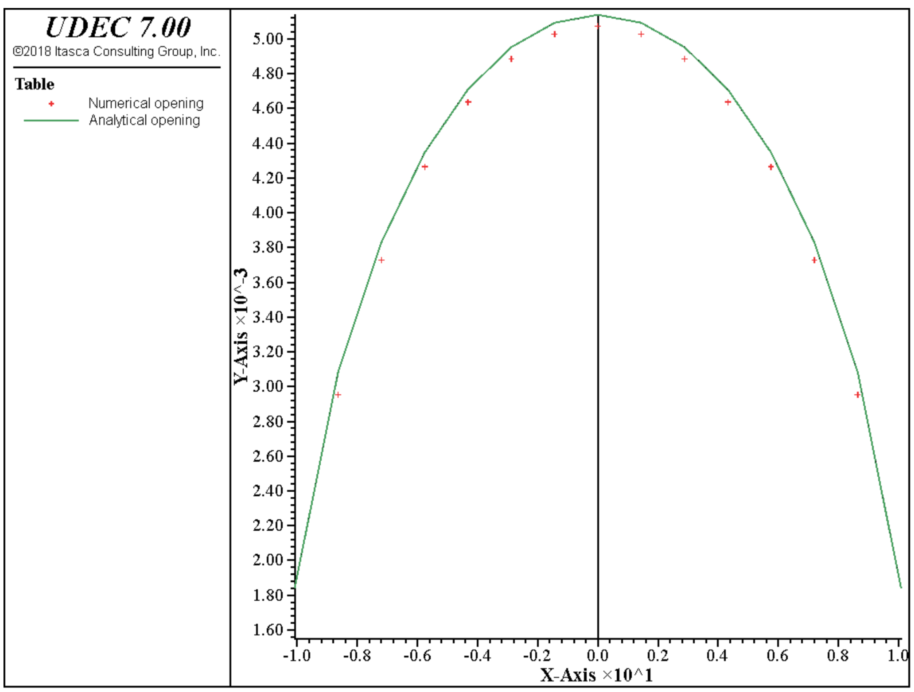

Figure 14.2 Fracture opening along the fracture for Case 1 – line: analytical solution; crosses: UDEC solution

The relative error on fracture opening at the center of the fracture is less than 1%.

The displacement vectors in the model at the end of the simulation are shown in Figure 14.3.

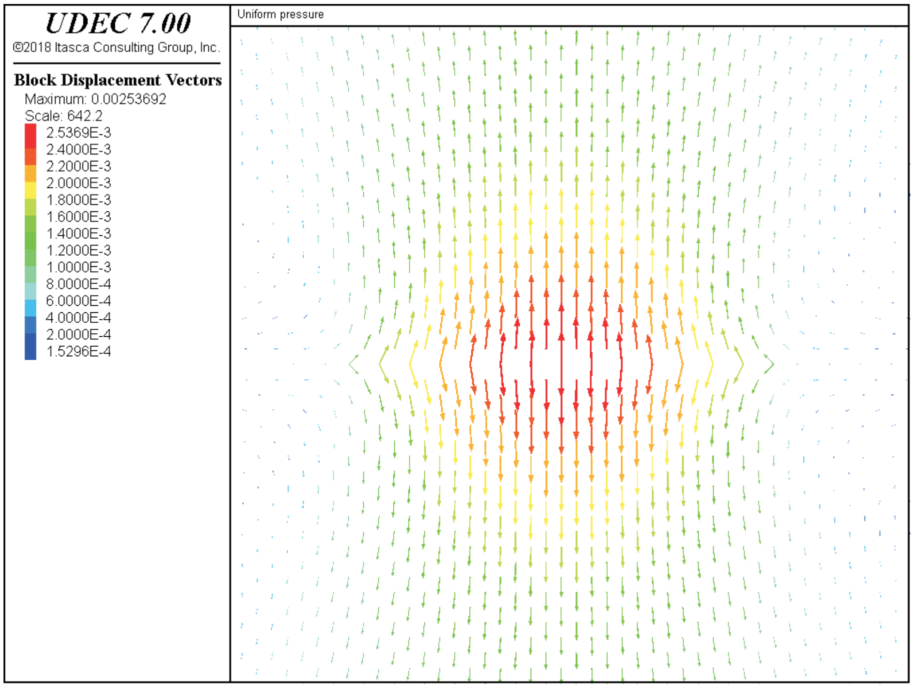

Figure 14.3 Displacement vectors at the end of the simulation – Case 1

14.2.2 Case 2

The analysis for Case 2 involves full fluid-mechanical coupling. The *UDEC* modeling methodology is based on three components: 1) the joint is initially dry (saturation is zero); 2) it offers no resistance to opening, in accordance with the zero toughness condition being investigated; and 3) the condition of zero flow ahead of the fracture tip is enforced.

14.2.2.1 UDEC Model

The *UDEC* model for the Case 2 simulations is 23.04 m by 23.04 m. The zone size is uniform and equal to 0.18 m. The zoning is shown in Figure 14.4 for a 2 m by 2 m portion of the *UDEC* model. The matrix is elastic, with a Young's modulus of 40,000 MPa and a Poisson's ratio of 0.22. A throughgoing joint is specified at mid-height in the model. The initial stresses are specified, and stress boundary conditions are prescribed. The joint is being assigned a "small" initial aperture of $2 \cdot 10^{-5}$ m (which is done for numerical reasons), and saturation is initialized at zero.

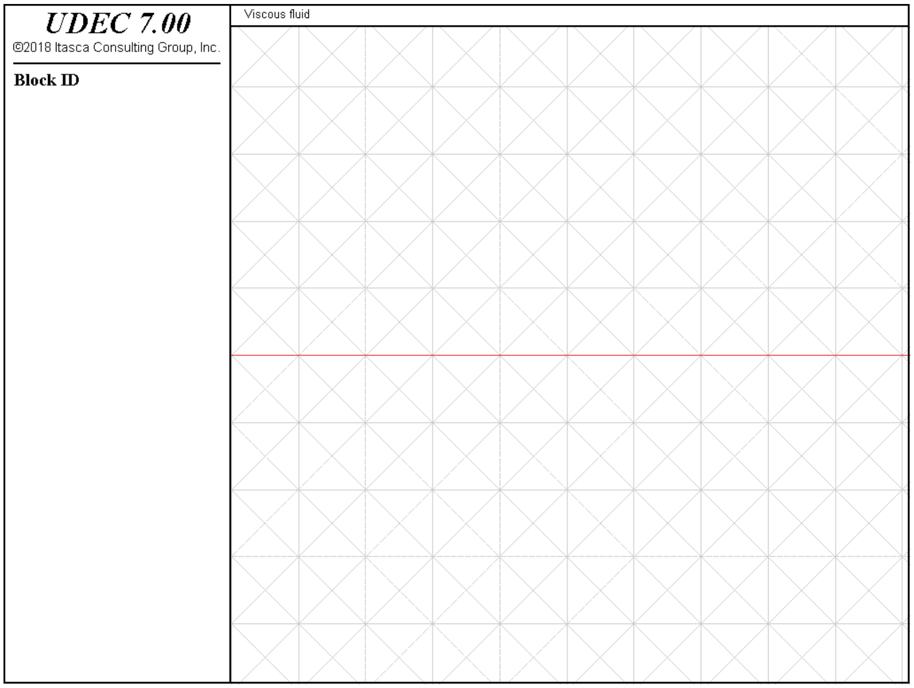

Figure 14.4 UDEC finite difference zones in a 2 m $\times$ 2 m portion of the model

A Coulomb slip with residual strength material model is assigned to the joint (joint model residual). This model has a special setting (**block fluid crack-flow on**) that is switched on in this analysis, to allow flow to only occur in fractured segments of the joint. (A joint is fractured when the joint shear or tensile strength is exceeded. Also, after a joint is fractured, the residual friction, cohesion and tensile strength values are used). The joint stiffness is equal to about 10 times the apparent stiffness of neighboring zones. The joint has no cohesion (initial and residual values are zero). A high value of joint friction (45 degree) is assigned initially, to prevent premature fracturing during the transient phase experienced by the model as it reaches a quasi-static state. The residual value of joint friction is zero. To enforce the zero toughness condition, the joint tensile strength is set to an initial (negative) value (close to the initial normal stress, which is −15 MPa, but *slightly* smaller, to prevent fracture detection along the whole joint). The initial value of tensile strength is chosen to be −14 MPa for the runs; the residual value is zero.

The compressible flow algorithm is selected. A value of fluid bulk modulus of 100 MPa is specified. The value is high compared to fluid pressure changes in the model; it is adequate to simulate an “incompressible” fluid. The numerical simulation is carried out in quasi-static mode (a servo control is used to enforce the condition that for each flow step, enough mechanical steps are taken to maintain the model in quasi-static equilibrium). Fluid injection at a constant rate is specified for the well, located at the origin of axes (center of the model). The fluid-mechanical simulation is carried out for a total of 10 seconds of injection.

14.2.2.2 UDEC Results

The fluid pressure and hydraulic aperture are monitored at the well for the total simulation time covering 10 seconds of fluid injection. The model state is saved at intervals of 2.5 seconds for analysis of fracture pressure and width at intermediate times of 2.5, 5 and 7.5 seconds.

Pressure and Width at the Well

The fracture width is calculated from the hydraulic aperture, by subtracting the initial aperture which is $2 \cdot 10^{-5}$ m for the runs. The fluid pressure and width at the well are plotted versus time in Figure 14.5 and Figure 14.6, respectively. The *UDEC* predictions are compared, in the same figure, with 1) the numerical DD solution, and 2) the first order approximation of the zero toughness solution (FMO) proposed by Adachi and Detournay (2002) (and Detournay 2004).

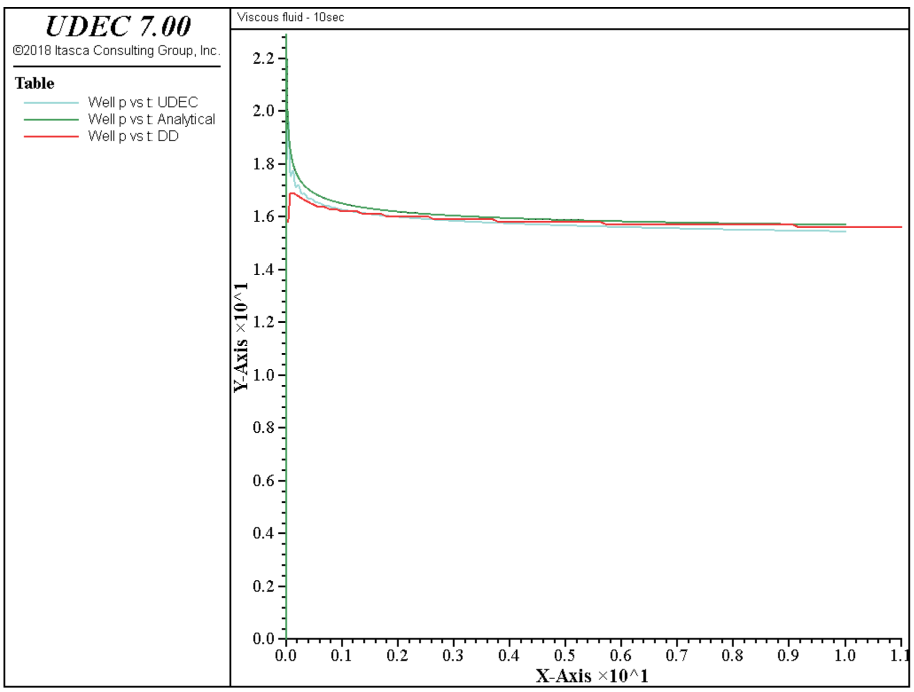

Figure 14.5 Fluid pressure at the well (MPa) versus time (sec)

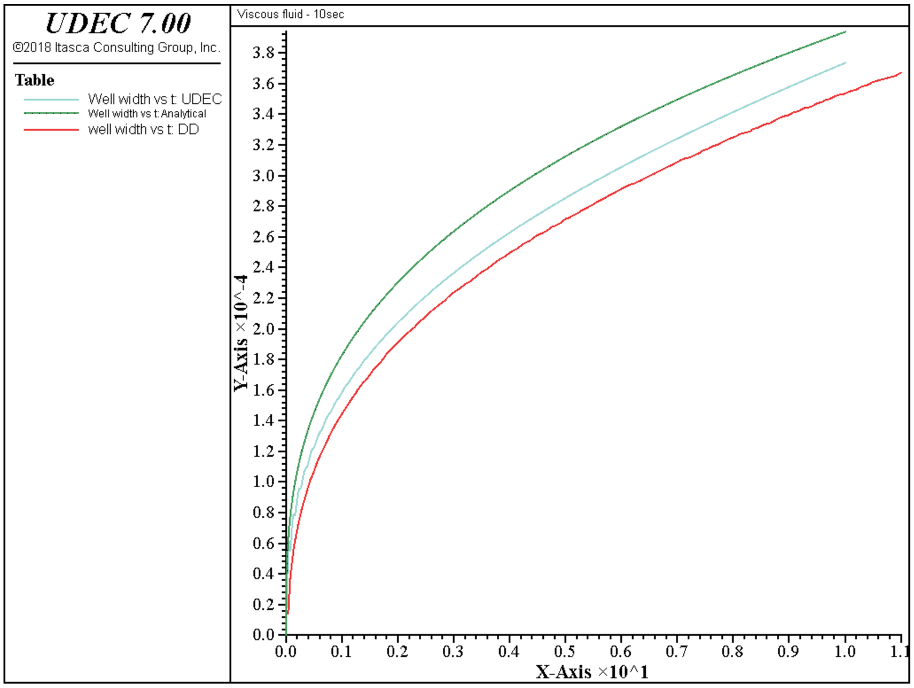

Figure 14.6 Width at the well (m) versus time (sec)

As may be observed from Figure 14.6, the *UDEC* solution for width at the well is bounded above by the analytical solution, and below by the DD solution.

Pressure and Width Distribution in the Fracture

The fracture pressure prediction from *UDEC* is compared to the first order approximation of the zero toughness solution (Adachi and Detournay 2002, and Detournay 2004) at 10 seconds of fluid injection in Figure 14.7.

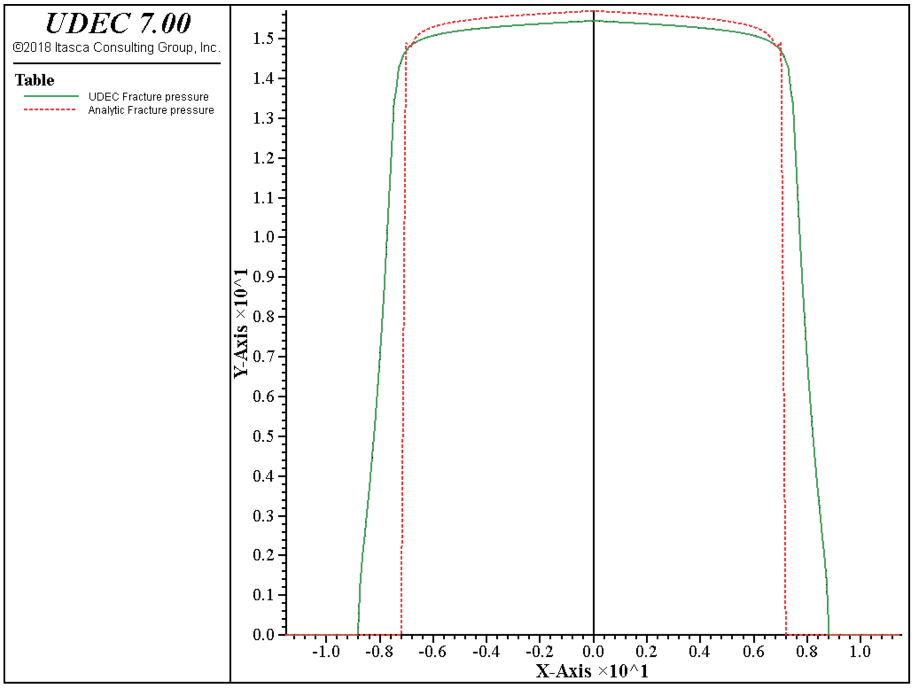

Figure 14.7 Fluid pressure (MPa) versus distance (m) along the fracture at 10 sec

It appears from Figure 14.7 that the *UDEC* pressure solution underestimates the value predicted by the analytical solution by about 7%.

The *UDEC* prediction for fracture width is compared to the FMO solution (Adachi and Detournay 2002, and Detournay 2004) at 10 seconds of fluid injection in Figure 14.8.

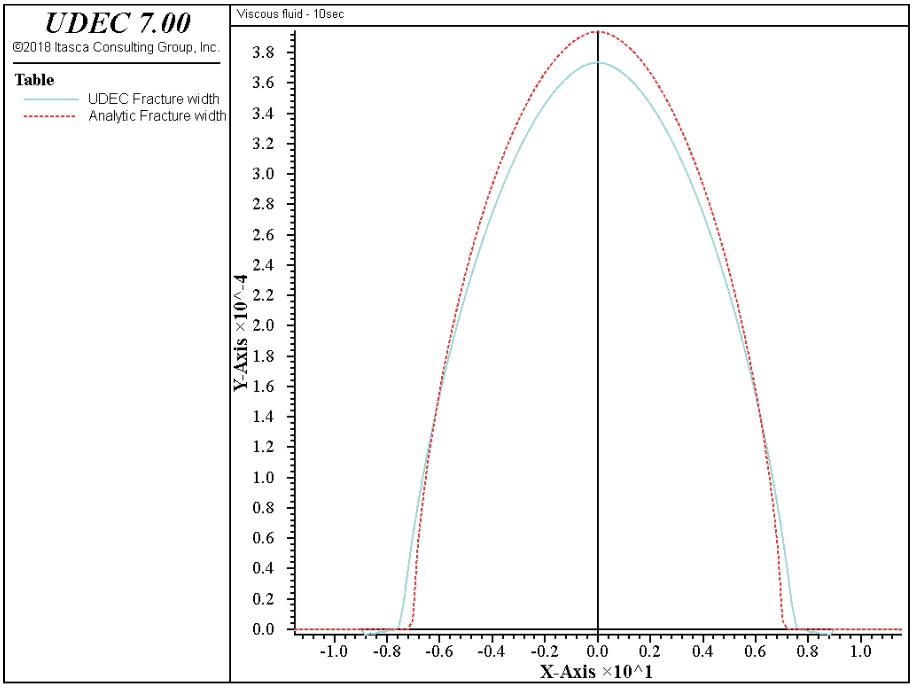

Figure 14.8 Fracture width (m) versus distance (m) along the fracture at 10 sec

The fracture width predicted by *UDEC* in the reported simulations is underestimated, when compared to the FMO solution, by about 1%. However, the discrepancy for maximum value of width does not seem to grow with time in the reported results. On the other hand, the match for fracture length between the *UDEC* prediction and the FMO solution appears to be very good.

Displacement vectors in the vicinity of the fracture after 10 seconds of injection are shown in Figure 14.9.

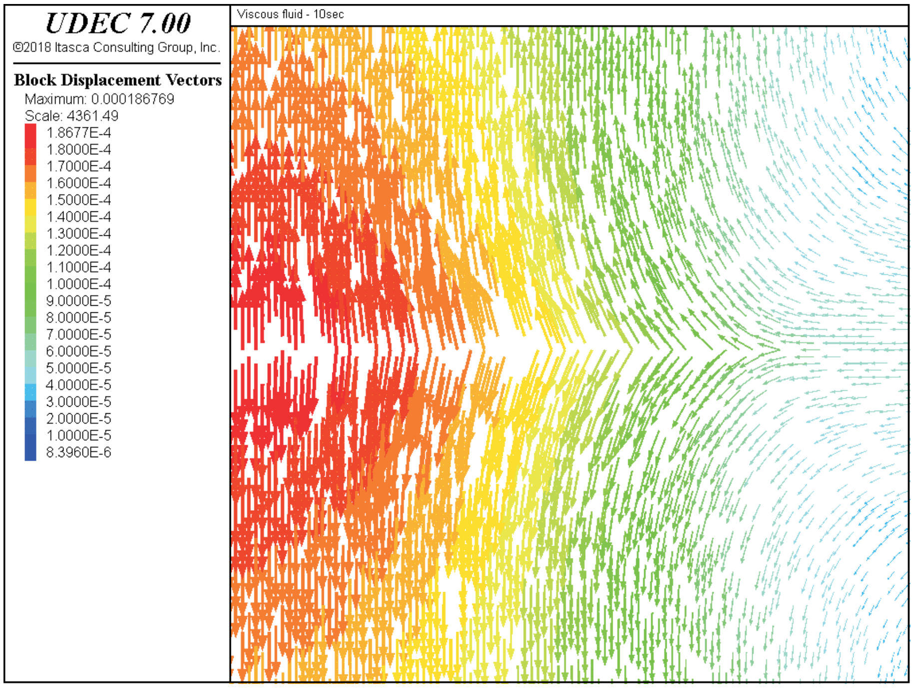

Figure 14.9 Displacement vectors near the fracture (half fracture shown) at 10 sec.

The numerical simulation for 10 seconds of injection takes approximately 50 minutes to run on an Intel Core i7-950 with the model setup described in this section.

14.3 Summary

Numerical simulations have been carried out with *UDEC* to simulate fluid injection in a preexisting fracture with zero toughness. The numerical predictions for pressure and width at the well have been compared with available DD results and the analytical first order approximation of the zero toughness solution (FMO) of Adachi and Detournay (2002). Also, pressure and width along the fracture have been compared to the FMO solution. The reported *UDEC* results for fracture growth show an excellent match with the FMO solution. Fluid pressure at the well predicted by *UDEC* matches well with the FMO solution. The *UDEC* results for width at the well are bounded by the FMO solution and the DD solution.

14.4 References

Adachi, J. I., and E. Detournay. "Self-similar solution of a plane strain fracture driven by a power-law fluid," *Int. J. Numer. Anal. Meth. in Geomechanics*, ***26***(6), 579-604 (2002).

Detournay, E. "Propagation regimes of fluid-driven fractures in impermeable rocks," *Int. J. Geomechanics*, ***4***(1), 35-45 (2004).

Parker, A. P. ***The Mechanics of Fracture and Fatigue***. E. & F. N. Spon (1981).

14.5 Listing of Data Files

Example 14.1 HF_CASE1.DAT

```
model new

;file 'hf_case1.dat
block config fluid
; --------------------------------
; *** Case 1: Uniform pressure ***
; --------------------------------
model title "Uniform pressure"
fish define setup
  _young =  40e3     ; MPa
  _nu    =  0.22
  _syy   = -15.
  _sxx   = -30.
  _pp    =  20.
  _hl    = 10.8      ; fracture half length
;
  _jkn   = 3e4
  _ares  = 5e-5
  _a0    = 5e-4
end
@setup
;
block tolerance corner-round-length 0.01
block create polygon -23.04 -23.04 -23.04 23.04 23.04 23.04 23.04 -23.04
block cut crack -24 0 24 0
;
block zone gen edge 1.6
;
block contact join by-contact
block contact join by-contact off range pos-x -10.5 10.5 pos-y -1 1
;
block property material 1 density 1e-3 young @_young poisson @_nu
; --- crack ---
block contact cmodel assign residual st-n @_jkn st-s @_jkn ...
  permeability-factor 300 aperture-residual @_ares
;
block fluid  property density 1e-3
;
block insitu  stress @_sxx 0 @_syy stress-ZZ 0 pore-pressure 0 aperture 0.0
;
block edge apply stress @_sxx 0 @_syy
block cycle 1
```

```
;
; --- boundary element representation of the far field ---
block boundary-element gen range pos-x -24 24 pos-y -24 24
block boundary-element material 1
block boundary-element fix 0 -24 -24 0
block boundary-element stiff
;
block domain update 100000
;  to speed calculation
; uniform pressure in fracture
bl domain fix range pos-x -10.5 10.5 pos-y -.1 .1
bl domain initialize pore-pressure @_pp range pos-x -10.5 10.5 pos-y -.1 .1
;
; keep zero pp in impermeable joints
block domain fix range pos-x -50 -10.5 pos-y -.1 .1
block domain initialize pore-pressure 0 range pos-x -50 -10.5 pos-y -.1 .1
block domain fix range pos-x 10.5 50 pos-y -.1 .1
block domain initialize pore-pressure 0 range pos-x 10.5 50 pos-y -.1 .1
;
history interval 1
block gridpoint history displacement-y 0 0
block gridpoint history displacement-y 2 0
block gridpoint history displacement-y 4 0
block gridpoint history displacement-y 6 0
block gridpoint history displacement-y 8 0
block gridpoint history displacement-y 10 0
model display hist 1
hist name 1 label 'Ydis at x=0'
hist name 2 label 'Ydis at x=2'
hist name 3 label 'Ydis at x=4'
hist name 4 label 'Ydis at x=6'
hist name 5 label 'Ydis at x=8'
hist name 6 label 'Ydis at x=10'
;
model save 'hf_case1_ft.sav'
block fluid steady-state off
block fluid aperture-max-ratio 100
block solve force 0 ratio 1e-6
model save 'hf_case1.sav'
model rest 'hf_case1.sav'
;
fish define fracop
; --- fracture opening ---
  bpnt=block.head
  loop while bpnt # 0
     pnt=block.gp(bpnt)
```

```
      loop while pnt # 0
        _x=block.gp.pos.x(pnt)
        _y=block.gp.pos.y(pnt)
        if math.abs(_y) < 0.1 then
          if math.abs(_x) < 10.5 then
            if bpnt=block.head then
              table(1,_x)=block.gp.disp.y(pnt)
            else
              table(2,_x)=-block.gp.disp.y(pnt)   ;<--(look: - sign)
            endif
          endif
        endif
        pnt=block.gp.next(pnt)
      endloop
      bpnt=block.next(bpnt)
  endloop
;
  coe = 2.*(1.-_nu*_nu)*((_pp+_syy)/_young)
  _hl = 10.8
  nitem=table.size(1)
  loop ii (1,nitem)
    _x = table.x(1,ii)
    table.x(3,ii)=_x
    table.y(3,ii)=table.y(1,ii)+table.y(2,ii)      ; numerical
    table.x(4,ii)=_x
    table.y(4,ii)=2.*coe*math.sqrt(_hl*_hl-_x*_x)     ; analytical opening
  endloop
  _omax=2.*coe*_hl
  _err=-100.*(table(3,0.)-_omax)/_omax
end
@fracop
table 3 label 'Numerical opening'
table 4 label 'Analytical opening'
fish list @_err
model save 'hf_case1_tables.sav'
return
```

Example 14.2 HF_CASE2.DAT

```
model new
;file 'hf_case2.dat
block config fluid
; ----------------------------
; *** Case 2: viscous fluid ***
; ----------------------------
model title '  Viscous fluid'
fish define setup
  _young =  40e3     ; MPa
  _nu    =  0.22
  _syy   = -15.
  _sxx   = -30.
  _hl    =  10.8     ; fracture half length
  _edge  =  0.2
;
  _bu = _young/(3.*(1.-2.*_nu))
  _sh = _young/(2.*(1.+_nu))
  _jkn  = 3e6
  _jten = -14.
;
  _ares   = 2e-5
  _a0     = -_syy/_jkn + _ares
  _amax   = 6e-3
  _caprat = _amax/_ares
end
@setup
;
block tolerance corner-round-length 0.01
block create polygon -11.52 -11.52 -11.52 11.52 11.52 11.52 11.52 -11.52
block cut crack -12 0 12 0
;
block zone gen edge @_edge
;
block property material 1 density 1e-3 bulk @_bu shear @_sh
; --- crack ---
block contact cmodel assign residual
block contact property st-n @_jkn st-s @_jkn cohesion 0. tension @_jten ...
  friction 45.
block contact property permeability-factor 83.33e6 ...
  aperture-zero-load @_a0 aperture-residual @_ares
;
block fluid  property density 1e-3
;
```

```
block insitu  stress @_sxx 0 @_syy stress-ZZ 0 pore-pressure 0
;
block edge apply stress @_sxx 0 @_syy impermeable
block domain initialize saturation 0
;
block domain update 100000
;  to speed calculation
fish define setup2
  _cp0 = block.contact.near(0.,0.)
end
@setup2
;
;
block fluid aperture-max-ratio @_caprat
;
block fluid compressible on
block fluid  property bulk 100
block cycle 0
model save 'hf_case2_ini.sav'
;
fish define _kw
  _ap = _a0 + block.contact.disp.normal(_cp0)
  _ap = math.max(_ares,_ap)
  _ap = math.min(_amax,_ap)
  _kw    = 10.*(_bu/1.44)*math.abs(_ap)
  _tdel  = block.mechanical.timestep
end
;
; injection well at x=0.
fish define in_flow
  in_flow=block.domain.near(0.,0.)
end
block domain well domain @in_flow flow 4e-4
;
history interval 100
block gridpoint history displacement-y 0 0
block gridpoint history displacement-y 2 0
block gridpoint history displacement-y 4 0
block gridpoint history displacement-y 6 0
block gridpoint history displacement-y 8 0
block gridpoint history displacement-y 10 0
block mechanical history time-total
block fluid history time-total
block domain history pore-pressure .0 0
block domain history pore-pressure .72 0
block domain history pore-pressure 2.16 0
```

```
block domain history pore-pressure 3.60 0
block domain history pore-pressure 5.04 0
block domain history pore-pressure 6.48 0
block domain history pore-pressure 7.92 0
block domain history pore-pressure 9.36 0
block domain history pore-pressure 10.8 0
fish history @_kw
fish history @_tdel
fish history @_ap
block contact history stress-normal .0 0
block contact history stress-normal .72 0
block contact history stress-normal 2.16 0
block contact history stress-normal 3.60 0
block contact history stress-normal 5.04 0
block contact history stress-normal 6.48 0
block contact history stress-normal 7.92 0
block contact history stress-normal 9.36 0
block contact history stress-normal 10.8 0
model display hist 9
hist name 1  label 'Ydis at x=0'
hist name 2  label 'Ydis at x=2'
hist name 3  label 'Ydis at x=4'
hist name 4  label 'Ydis at x=6'
hist name 5  label 'Ydis at x=8'
hist name 6  label 'Ydis at x=10'
hist name 8  label 'Flow Time'
hist name 9  label 'Fluid Pressure at x=.0'
hist name 10 label 'Fluid Pressure at x=.72'
hist name 11 label 'Fluid Pressure at x=2.16'
hist name 12 label 'Fluid Pressure at x=3.60'
hist name 13 label 'Fluid Pressure at x=5.04'
hist name 14 label 'Fluid Pressure at x=6.48'
hist name 15 label 'Fluid Pressure at x=7.92'
hist name 16 label 'Fluid Pressure at x=9.36'
hist name 17 label 'Fluid Pressure at x=10.80'
hist name 18 label 'Fluid bulk modulus'
hist name 19 label 'Fluid ? time step'
hist name 20 label 'Aperture at x=0'

block fluid crack-flow off
;
fish define _maxVel
  maxVel_ = 0.
  iBlock_ = block.head
  loop while iBlock_ # 0
    iGp_ = block.gp(iBlock_)
```

```
    loop while iGp_ # 0
      maxVel_ = math.max(maxVel_,math.sqrt(block.gp.vel.x(iGp_)^2+ ...
                         block.gp.vel.y(iGp_)^2))
      iGp_ = block.gp.next(iGp_)
    end_loop
    iBlock_ = block.next(iBlock_)
  end_loop
end
;
fish define _cycle
  nStep_ = nStep_
  loop while block.fluid.time.total < _ftime
    command
      block cycle @nStep_
    end_command
    tfdel_ = (block.fluid.time.total-ftimeOld_)/(nStep_*nf_)
    if tfdel_ < tfdelLimit_ then
      frac_ = frac_*tfdelLimit_/tfdel_
      command
        block mechanical timestep-factor 0.1 1.0 @frac_
      end_command
    endif
    _maxVel
    if maxVel_ > upperBound_ then
      if nf_ > 1 then
        nfm_ = 1
        nf_  = nf_-1
      else
        nfm_ = nfm_+1
        nf_  = 1
      endif
    else
      if maxVel_ < lowerBound_ then
        if nfm_ > 1 then
          nfm_ = nfm_-1
          nf_  = 1
        else
          nfm_ = 1
          nf_  = nf_+1
        endif
      endif
    endif
    command
      block fluid substep-mechanical @nfm_
      block fluid gasflow substep-flow @nf_
    end_command
```

```
    ftimeOld_ = block.fluid.time.total
  end_loop
end
fish set @nStep_ 100
fish set @nfm_ 1
fish set @nf_ 1
fish set @lowerBound_ 5e-4
fish set @upperBound_ 5e-3
fish set @tfdelLimit_ 0.5e-6
fish set @frac_ 1.0

fish set @_ftime=2.5
@_cycle
model save 'hf_case2_2p5sec.sav'
fish set @_ftime=5.
@_cycle
model save 'hf_case2_5sec.sav'
fish set @_ftime=7.5
@_cycle
model save 'hf_case2_7p5sec.sav'
fish set @_ftime=10.
@_cycle
model save 'hf_case2_10sec.sav'
; -------------------------------------------------------
;    fluid pressure and opening at the well
; comparison between:
;   UDEC solution
;   DD solution
;   Analytical solution (Adachi and Detournay, 2002)
; -------------------------------------------------------
;
model res 'hf_case2_10sec.sav'
history export 8 table 1
history export 9 table 2
history export 20 table 3
fish define ana_sol
  _num=table.size(1)
  loop ii (2,_num)
    _tim = table.y(1,ii)
    _tim2 = _tim^(1./3.)
    table.x(2,ii)=_tim
    table.x(3,ii)=_tim
    table.y(3,ii)=table.y(3,ii)-_ares
    table.x(12,ii)=_tim
    table.y(12,ii)=15.+1.508/_tim2
    table.x(13,ii)=_tim
```

```
    table.y(13,ii)=1.827*_tim2*1e-4
  end_loop
end
@ana_sol
table 2  label 'Well p vs t: UDEC'
table 12 label 'Well p vs t: Analytical'
table 3  label 'Well width vs t: UDEC'
table 13 label 'Well width vs t: Analytical'
;; --- import Displacement Discontinuity results ---
call 'table22.txt'
call 'table23.txt'
;
table 22 label 'Well p vs t: DD'
table 23 label 'well width vs t: DD'
;
;plot hold table 2 12 22
;plot hold table 3 13 23
model title 'Viscous fluid - 10sec'
;
fish define _width
  pnt=block.contact.head
  loop while pnt# 0
    _x=block.contact.pos.x(pnt)
    _y=block.contact.pos.y(pnt)
    if _y < 0.1 then
      if _y > -0.1 then
        table(20,_x)=_a0+block.contact.disp.normal(pnt)-_ares
      end_if
    end_if
    pnt=block.contact.next(pnt)
  end_loop
  pnt=block.domain.head
  loop while pnt # 0
    _x=block.domain.pos.x(pnt)
    _y=block.domain.pos.y(pnt)
    if _y < 0.1 then
      if _y > -0.1 then
        if math.abs(_x) > 1e-2 then
          table(30,_x)=block.domain.pp(pnt)
        end_if
      end_if
    end_if
    pnt=block.domain.next(pnt)
  end_loop
end
@_width
```

```
;
call 'table50.txt'
call 'table51.txt'
;
fish define ana_width
  _num=table.size(20)
  _time=10.
  _l=1.516*_time^(2./3.)
  _coew=1.623*_time^(1./3.)
  _coep=2.768*_time^(-1./3.)
  loop ii (1,_num)
    _x=table.x(20,ii)
    _xi=math.abs(_x)/_l
    _om=table(51,_xi)
    table(21,_x)=_coew*_om*1e-4
    _pi=table(50,_xi)
    _press=_coep*_pi + 15.
    if _pi = 0.0 then
      _press = 0.0
    end_if
    table(31,_x)=_press
  end_loop
end
@ana_width
;
table 20 label 'UDEC Fracture width'
table 21 label 'Analytic Fracture width'
table 30 label 'UDEC Fracture pressure'
table 31 label 'Analytic Fracture pressure'
model save hf_case2_10sec_tables.sav

;plot hold table 20 21 xw -10 10 yw 0 4e-4
;plot hold table 30 31 xw -10 10 yw 0 16.5
return
```