GeMA – Modelling of Fluid Migration through fractured porous media

Version 1.0

Key Example Points

This example shows:

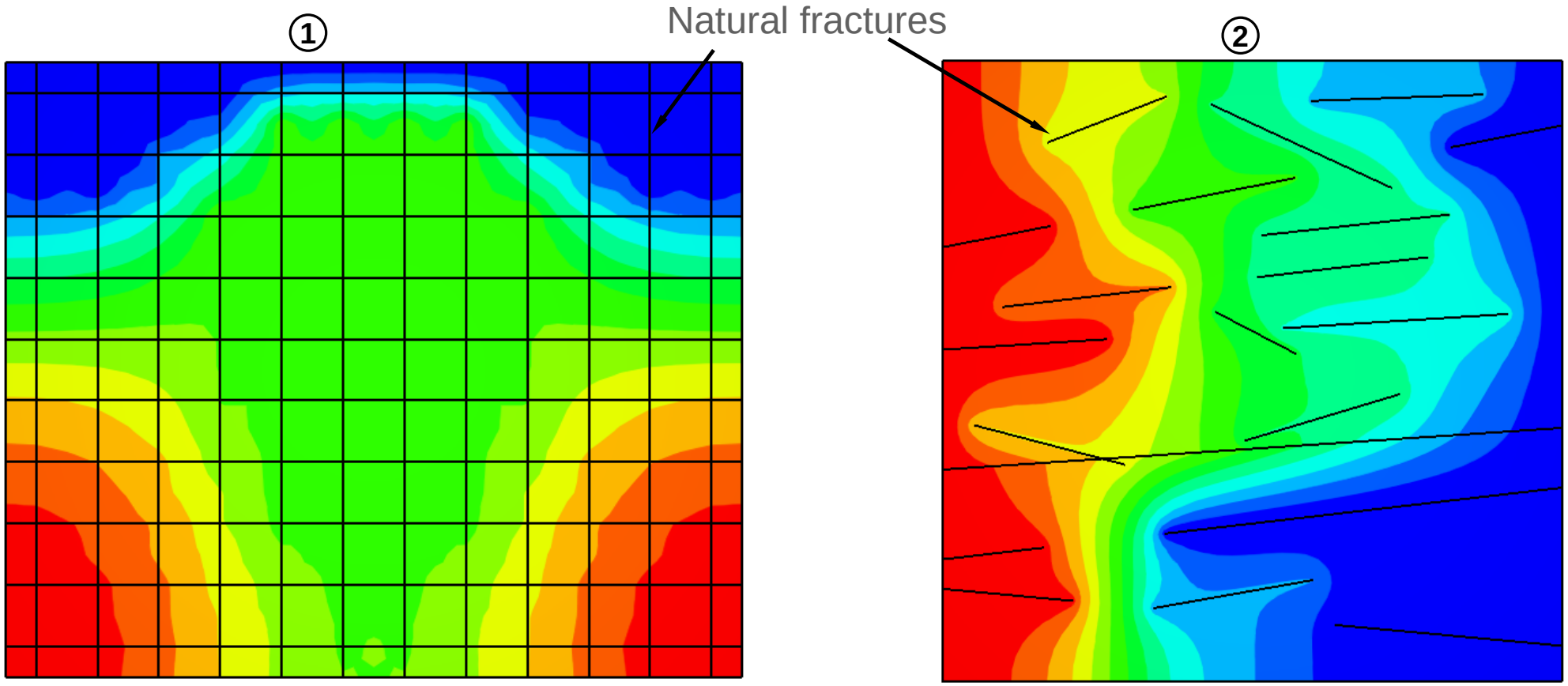
- How to setup models for transient flow analysis in fractured porous media

- How to use a coupled HM interface element as a single fracture to represent naturally fractured media

- How to create a Multiphysics simulation in GeMA where one of the physics represents explicitly the fractures and another physics represent the porous media.
- Investigate the impact of natural fractures on the flow process considering significant reductions of pore pressure.

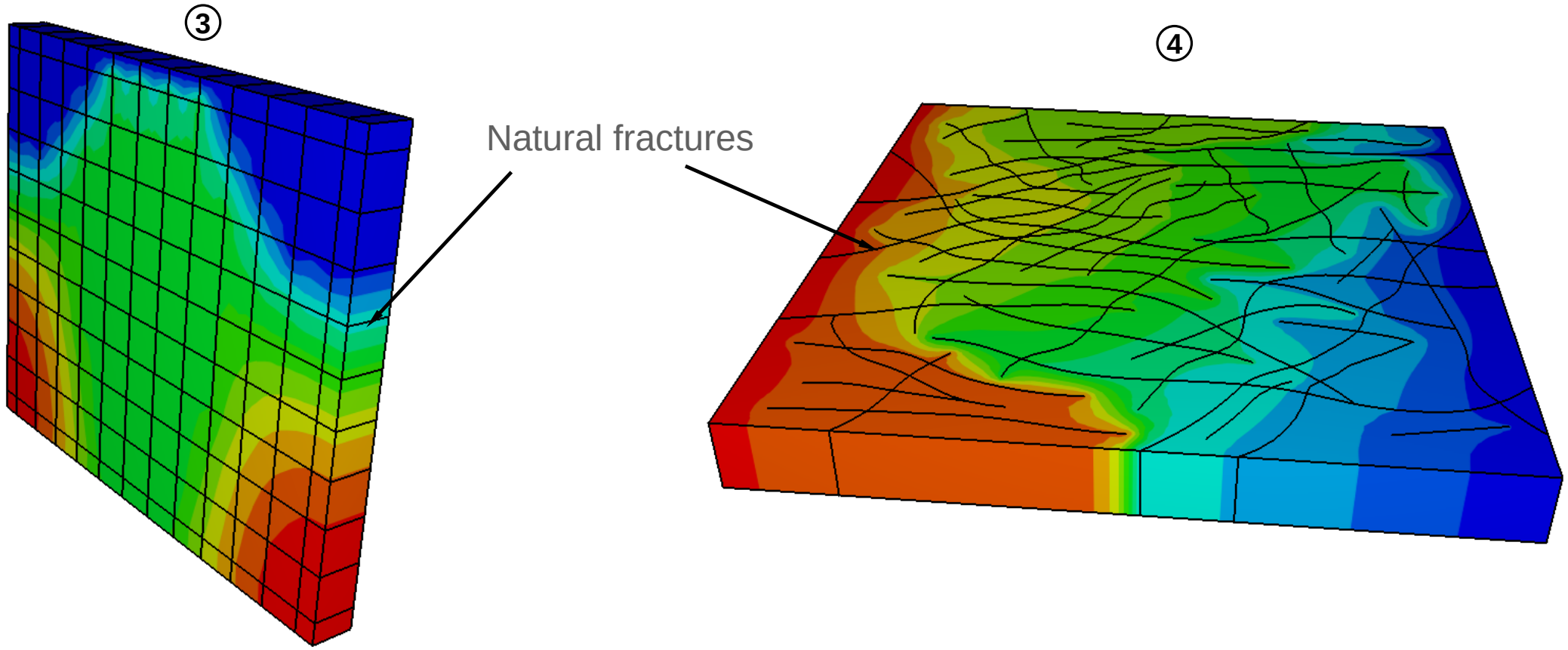
The examples

- The examples 1 and 2 present a transient analyses for consolidation and fluid flow in 2 dimensional fractured porous media, respectively.



The examples

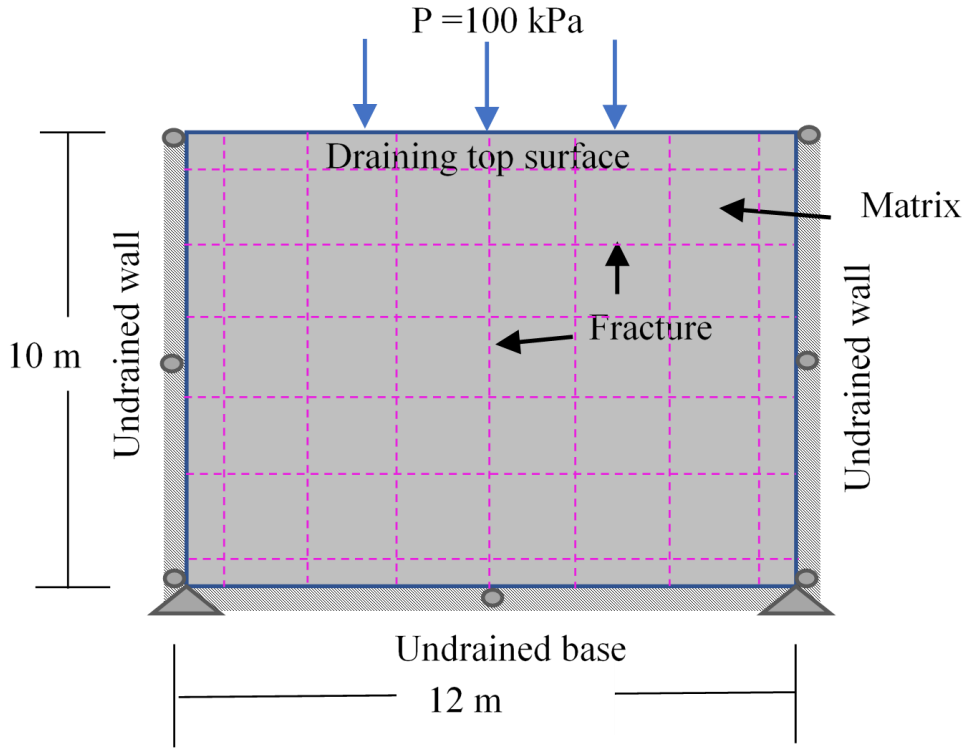
- The examples 3 and 4 present a transient analyses for consolidation and fluid flow in 3 dimensional fractured porous media, respectively.



1 – CONSOLIDATION OF 2D FRACTURED POROUS MEDIA

The problem

- In this problem, a rectangular fractured porous media is subjected to pressure along 4 m on the top of the model. Base is constrained in y-direction and left and right side are constrained in x-direction. Natural fractures are represented by zero-thickness interface elements.



Simulation file: DFMkvar.lua

The result is presented in: Rueda, C. J et al. “Discrete and smeared fracture model to simulate fluid flow in naturally fractured reservoirs”

Model file: State variable

For a coupled Hydro-Mechanical analysis, displacement and pore pressure state variables are required.

```
-- State variables
StateVar{id = 'u', dim = 2, description = 'Displacements in the X and Y directions', unit = 'm',
         format = '8.4f', groupName = 'mechanic'}

StateVar{id = 'P',  description = 'Pore pressure', unit = 'kPa',
         format = '8.4f', groupName = 'hydraulic'}
```

The group name is used by the fem solver to group types of state variables and to define tolerances for each one

Both physics, continuum and interface, use the same state variables

Model file: Hydromechanical properties

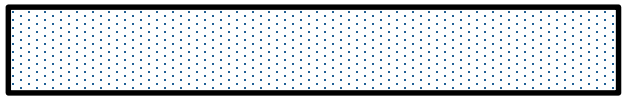
```
PropertySet
{
  id          = 'HydroProp',
  typeName    = 'GemaPropertySet',
  description = 'Material properties',
  properties  = {
    {id = 'E',   description = 'Elasticity modulus',       unit = 'kPa'},
    {id = 'nu',  description = 'Poisson ratio'},
    {id = 'K',   description = 'Hydraulic permeability',   unit = 'm/s'},
    {id = 'gw',  description = 'Specific weight of water', unit = 'kN/m3'},
    {id = 'Kww', description = 'Bulk modulus of water',    unit = 'kPa'},
    {id = 'Pht', description = 'Porosity'},
    {id = 'Kn',  description = 'Normal elastic stiffness', unit = 'kPa/m'},
    {id = 'Ks',  description = 'Shear elastic stiffness',  unit = 'kPa/m'},
    {id = 'Gap', description = 'Initial gap opening',      unit = 'm'},
    {id = 'Ufw', description = 'Dynamic fluid viscosity',  unit = 'kPa*s'},
    {id = 'Lkt', description = 'Leakoff at top',           unit = 'm/(kPa*s)'},
    {id = 'Lkb', description = 'Leakoff at bottom',        unit = 'm/(kPa*s)'},
    {id = 'material',  description = 'Mechanical material type',
     constMap = constants.CoupledHMFemPhysics.materialModels}
  },
  values = {
    {E = 1e5, nu = 0.2, K = 1e-11 , gw = 10, Pht = 0.1, Kww = 2.20e+6,  material = 'poroElastic'},
    {gw = 10, Kn = 2e4, Ks = 1e4, Gap = 0.00004933, Ufw = 1e-6,  Lkt =1.0, Lkb = 1.0,
     material = 'elasticInterface', Mfw = 0}
  }
}
```

Model file: Hydromechanical properties

```
PropertySet
{
  id          = 'SecProp',
  typeName    = 'GemaPropertySet',
  description = 'Section properties',

  properties  = {
    {id = 'h',     description = 'Element thickness', unit = 'm'},
    {id = 'Iopen', description = 'Intially open interface'},
  },

  values = {
    {h =  1.0, Iopen = 0}, -- Iopen = 0 close fracture
    {h =  1.0, Iopen = 1}, -- Iopen = 1 open fracture
  }
}
```

Defines the initial fracture condition (open or closed).

Open fracture considers longitudinal fluid



No flow inside closed fracture

Solution file: Physic definition

The coupled interface element is available through the interface object from the CoupledHMFemPhysics plugin.

```
PhysicalMethod {
  id       = 'interface',
  typeName = 'CoupledHMFemPhysics.Interface',
  type     = 'fem',

  mesh               = 'mesh',
  materials          = {'elasticInterface'},
  ruleSet            = 1,
  isoParametric      = true,
  permeabilityUpdate = true,
}
```

Coupled interface object from coupled HM Fem physic

Option to allow permeability as a function of the fracture aperture update for longitudinal fluid flow

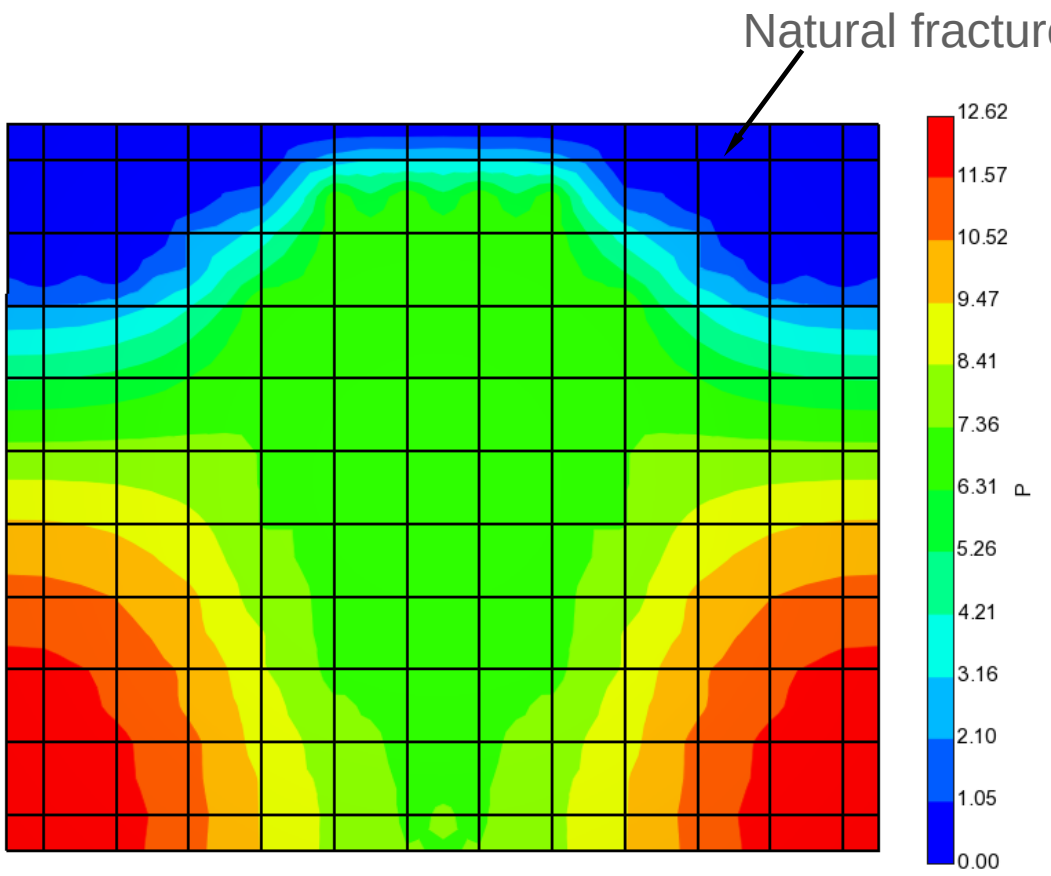
Solution file: Orchestration

```
function ProcessScript()
  -- Create the solver model
  local solver = fem.init({'Consolidation','interface'}, 'solver', solverOptions)

  -- Prepare the file where results will be saved
  local file = io.prepareMeshFile('mesh', '$SIMULATIONDIR/out/DFMkvar', 'nf', {'u','P'},{'S', 'E',},
                                  {split = true, saveDisplacements = true})

  -- Saves initial state to the file
  io.addResultToMeshFile(file, 0.0)

  ---------------------------------
  -- Consolidation step
  ---------------------------------
  local dt        = solverOptions.timeInitIncrement
  local FinalTime = solverOptions.timeMax
  local Time      = dt
  local LastStep  = false

  while (Time <= FinalTime) do
    print('------------------')
    print(tr('Consolidation step - time = %1 s'):num(Time))
    print('------------------')
```

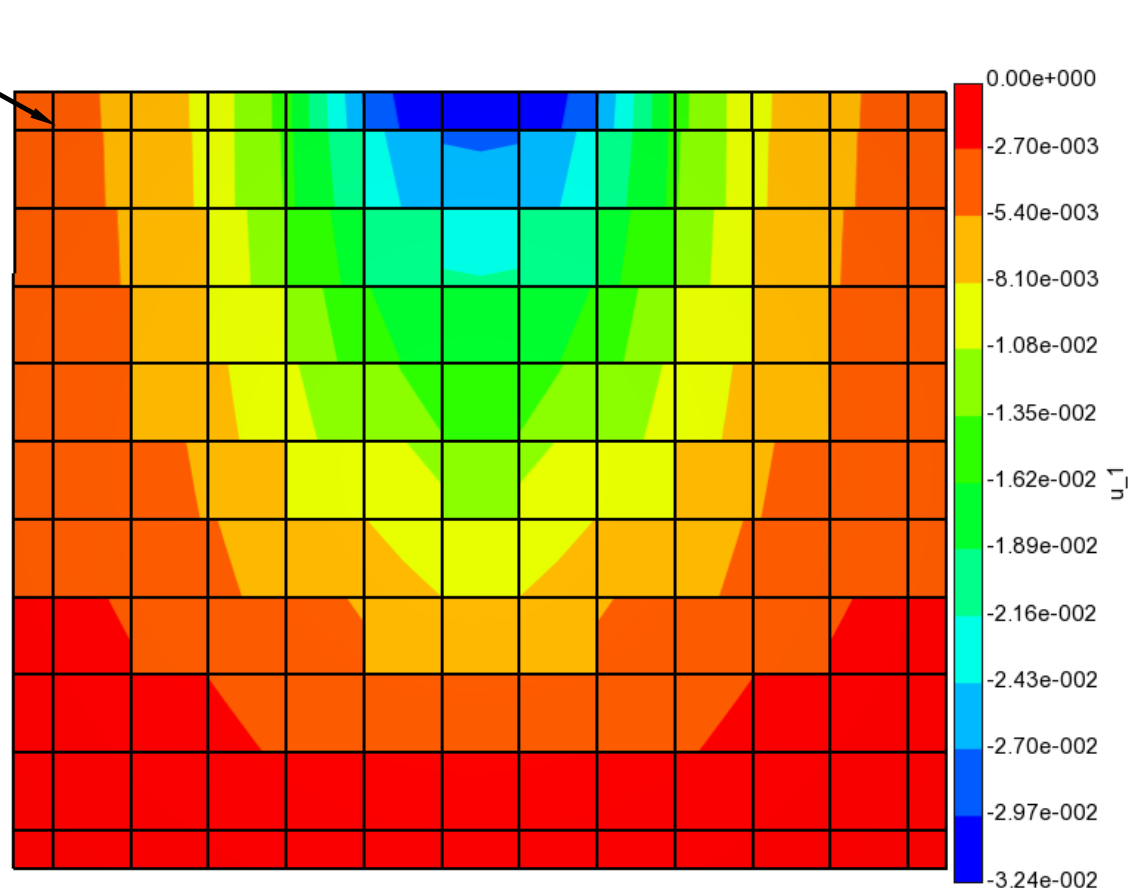
Solution file: Orchestration script

```
    -- Run transient analysis.  Solver returns a suggested time-step
    -- for the next iteration
    dt = fem.step(solver, dt)

    -- Save results
    io.addResultToMeshFile(file, Time)

    -- Adjust time to guarantee that the last iteration will be on the
    -- requested final simulation time
    if (Time + dt >= FinalTime and not LastStep) then
      dt       = FinalTime - Time
      Time     = FinalTime
      LastStep = true
      if equal(dt, 0.0) then break end -- Test special case when Time == FinalTime
    else
      Time = Time + dt
    end
    print('Time = ' .. Time)
  end

  -- Closes the result file
  io.closeMeshFile(file)
end
```

Results at time = 1.0e9 s



Pore pressure (kPa)

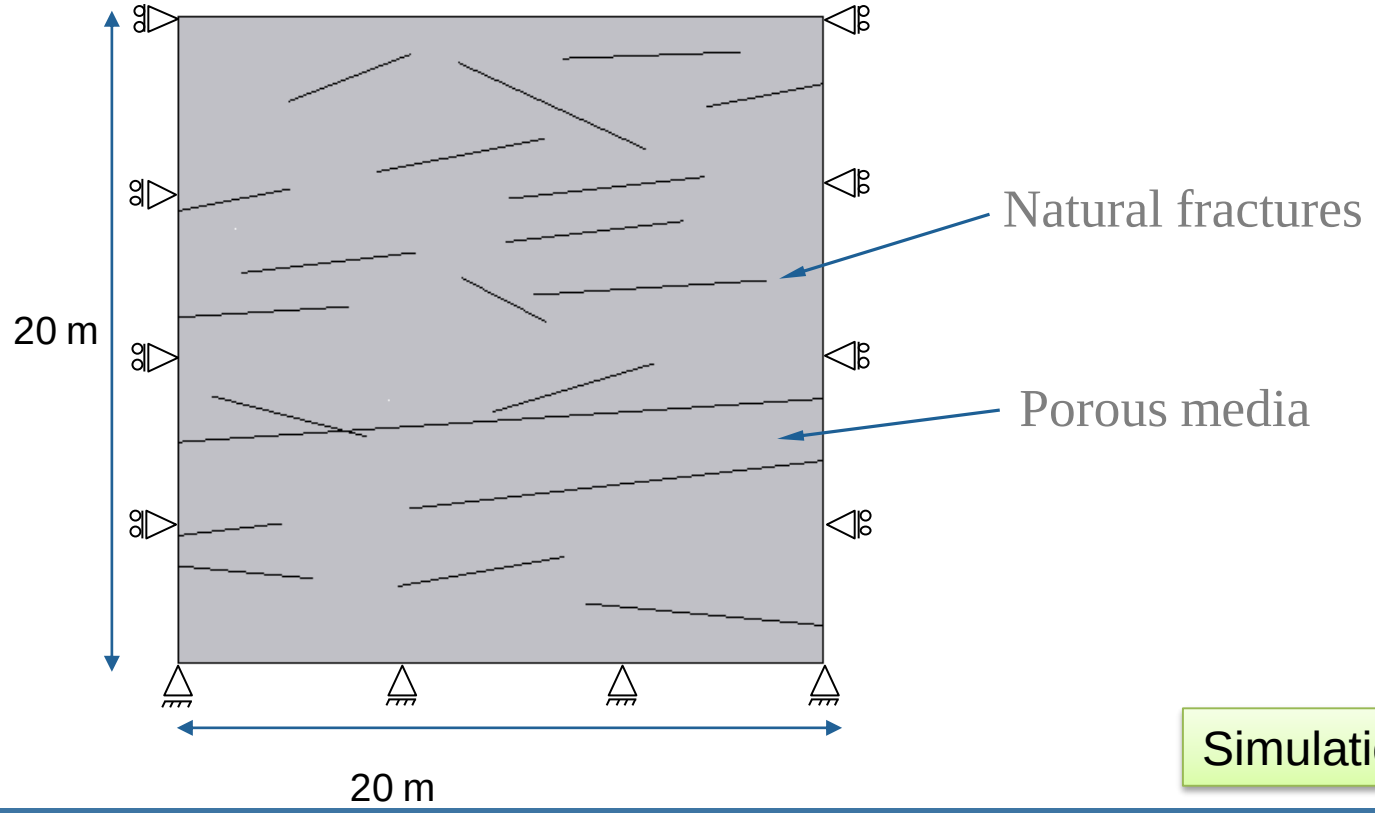


Vertical displacement (m)

2 – FLUID FLOW IN FRACTURED POROUS MEDIA

The problem

- In this problem, a rectangular fractured porous media under in-situ condition (stresses and pore pressure) is subjected to pore pressure gradient in x-direction. The impact of fractures on the fluid flow process is investigated.



k_0 = 8.63 10^{-4} m/d
n_0 = 0.1

E = 12000 MPa
ν = 0.3

p_0 = 55 MPa
σ_0 = 85 MPa

γ_w = 0.01 MN/m^3
μ_w = 10^{-9} MPa.s
β_w = 0.000458 MPa^{-1}

Simulation file: naturalFractures2D.lua

Model file: Hydromechanical properties

```
PropertySet
{ id          = 'MatProp',
  typeName    = 'GemaPropertySet',
  description = 'Material properties',
  properties  = {
    {id = 'E',     description = 'Elasticity modulus',                    unit = 'kPa'},
    {id = 'nu',    description = 'Poisson ratio'},
    {id = 'K',     description = 'Hydraulic permeability in x',           unit = 'm/s'},
    {id = 'gw',    description = 'Specific weight of water',              unit = 'kN/m^3'},
    {id = 'Kww',   description = 'Bulk modulus of water',                 unit = 'kPa'},
    {id = 'Pht',   description = 'Porosity'},
    {id = 'Kn',    description = 'Normal elastic stiffness',              unit = 'kPa/m'},
    {id = 'Ks',    description = 'Shear elastic stiffness at direction 1', unit = 'kPa/m'},
    {id = 'Gap',   description = 'Initial gap opening',                   unit = 'm'},
    {id = 'Ufw',   description = 'Dynamic fluid viscosity',               unit = 'kPa*s'},
    {id = 'Mfw',   description = 'fracture bulk compressibility',         unit = 'kPa/m'},
    {id = 'Lkt',   description = 'Leakoff at top',                        unit = 'm/kPa/s'},
    {id = 'Lkb',   description = 'Leakoff at bottom',                     unit = 'm/kPa/s'},
    {id = 'Iopen',description = 'Intially open interface'},
    {id = 'Closure',     description = 'Fracture closure model',   constMap = constants.MechanicalFemPhysics.closureModel},
    {id = 'materialHM', description = 'Hydro-mechanics material', constMap = constants.CoupledHMFemPhysics.materialModels},
    {id = 'h',           description = 'Element thickness', unit = 'm'},
  },
  values = {
    {E = 1.20e+07, nu = 0.3, h = 1, K = 9.99e-09, gw = 1.00e+01, Pht = 1.00e-01, Kww = 2.20e+06, materialHM = 'poroElastic'},
    {Kn = 1.50e+07, Ks = 1.00e+07, Gap = 5.00e-04, Ufw = 1.00e-06,  Lkt = 1.00e+00, Lkb = 1.00e+00, Mfw = 0.00e+00,
     Closure = 'normal', materialHM = 'elasticInterface', Iopen = 1, h = 1}, -- For interface element
  }
}
```

Table with properties required by the coupled interface and solid elements

Defines the initial fracture condition (open or closed).

Solution file: Physic definition

Two physics are defined for solid elements. The first for the geostatic step and the second one for transient analysis.

```
PhysicalMethod {
  id         = 'Geostatic',
  typeName = 'CoupledHMFemPhysics.PlaneStrain',    -- define plane strain condition
  type       = 'fem',
  mesh                 = 'mesh',
  elementGroups        = {'TRUP_1'},
  materials            = {'poroElastic'},           -- define poroElastic material
  isoParametric        = false,
  boundaryConditions = {'bpd', 'bpmGeo', 'bc'},    -- define boundary conditions
  ruleset              = 1,
  properties           = { material = 'materialHM' }
}
```

Geostatic step

```
PhysicalMethod {
  id         = 'Consolidation',
  typeName = 'CoupledHMFemPhysics.PlaneStrain', -- define plane strain condition
  type       = 'fem',

  mesh                 = 'mesh',
  elementGroups        = {'TRUP_1'},
  materials            = {'poroElastic'},         -- define poroElastic material
  isoParametric        = false,
  boundaryConditions = {'bpd', 'bpm', 'bc',},
  ruleSet              = 1,
  properties           = { material = 'materialHM' }
}
```

Transient step

Solution file: Physic definition

The coupled interface element is available through the interface object from the CoupledHMFemPhysics plugin.

```
PhysicalMethod {
  id       = 'interfaceHM',
  typeName = 'CoupledHMFemPhysics.Interface',
  type     = 'fem',

  mesh               = 'mesh',
  elementGroups      = {'Int2UP_2'},
  materials          = {'elasticInterface'},
  isoParametric      = false,
  permeabilityUpdate = true,
  ruleSet            = 1,
  properties         = {material = 'materialHM'}
}
```

Call coupled interface object from coupled HM Fem physic

Option to allow permeability as a function of the fracture aperture update for longitudinal fluid flow

Solution file: Initial in-situ conditions

The initial stresses are computed and stored at the integration points.

```
function initialCondition()
  -- Get the needed accessors.
  local mesh = modelData:mesh('mesh')
  -- Accessor for the old stress state
  local accS = mesh:gaussAttributeAccessor('S',1, true)
  -- Node coordinate accessor
  local coordAc = mesh:nodeCoordAccessor()

  -- Integration rules
  local triIr = mesh:elementIntegrationRule('tri3', 1)
  local intIr = mesh:elementIntegrationRule('int2dl4', 1)

  -- Initial stress for each element
  for i=1, mesh:numCells() do
    local e = mesh:cell(i)  -- element

    -- solid element
    if (e:type() == 'tri3') then
      -- for each integration point
      for j=1, triIr:numPoints() do
        -- fills the insitu stress
        local v = {-12857.0, -30000, -12857.0, 0.0}
        accS:setValue(e, j, v)
      end
    end
    -- interface element
    if (e:type() == 'int2dl4') then
      -- elemental node coordinates
      local Xnode = e:nodeMatrix(coordAc, true, 'vertices')
      assert(Xnode)
      -- get fault angle in radians
      local theta = getFaultAngle(Xnode)
      -- for each integration point
      for j=1, intIr:numPoints() do
        -- fill stress acording
        local Sv = -30000.0
        local Sh = -12857.0
        -- stresses on the fracture plane
        local Sn  = Sh*math.sin(theta)*math.sin(theta) +
                    Sv*math.cos(theta)*math.cos(theta)
        local Tau = (Sv-Sh)*math.sin(theta)*math.cos(theta)
        -- fills the insitu stress vector
        local v = {0, Sn, 0, Tau}
        accS:setValue(e, j, v)
      end
    end
  end
end
```

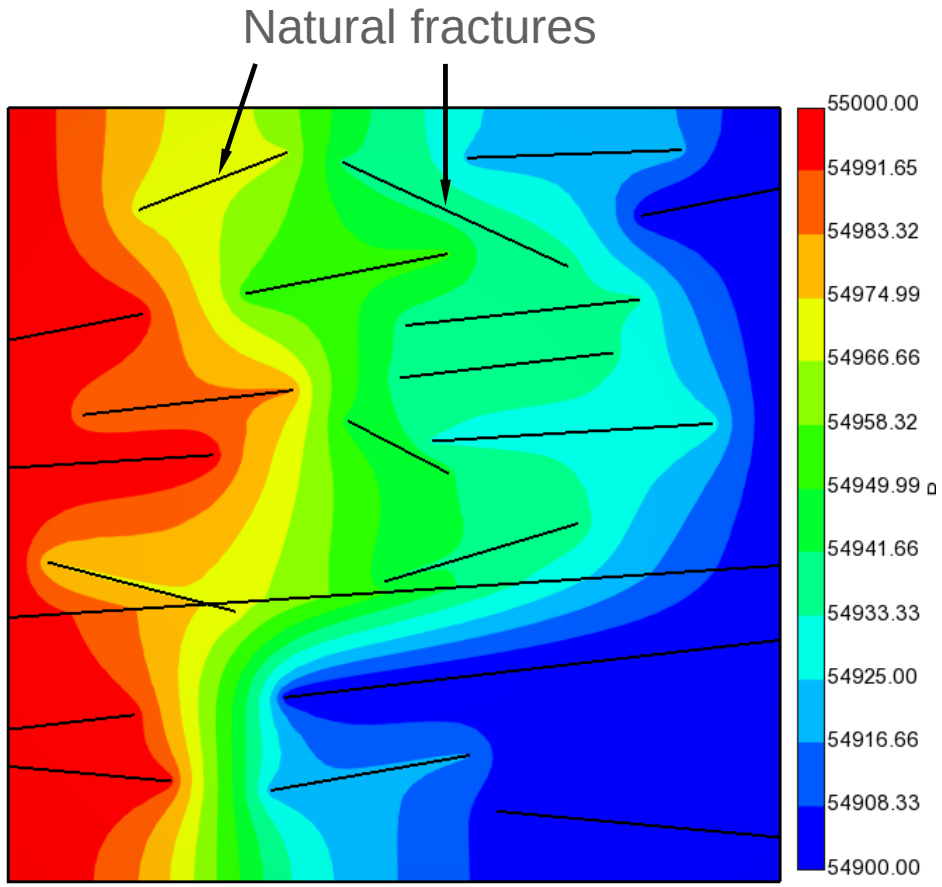
Solution file: Orchestration script for Geostatic step

```
-- Initial condition
initialCondition()
print(' GEOSTATIC Step.....\n'  )
-- Solves the Geostatic step using a transient nonlinear solver
local solver = fem.init({'interfaceHM','Geostatic'}, 'solver', solverOptions)
-- Prepare the file where results will be saved
-- Results are saved on the out sub-directory of the same directory hosting the simulation model
local file = io.prepareMeshFile('mesh', '$SIMULATIONDIR/out/naturalFractures2D', 'nf', {'u', 'P'}, {'S', 'E'},
                                {split = true, saveDisplacements = true})
local dt        = solverOptions.timeInitIncrement
local FinalTime = solverOptions.timeMax
local Time      = dt
local LastGeo   = false
while (Time <= FinalTime) do
    print('------------------')
    print(tr('Geostatic - time = %1 Step') :num(Time))
    print('------------------')
    dt = fem.step(solver, dt)
    -- Adjust time to guarantee that the last iteration will be on the requested final simulation time
    if ((Time + dt) >= FinalTime and not LastGeo) then
       dt      = FinalTime - Time
       Time    = FinalTime + 0.0001
       LastGeo = true
    else
       Time = Time +  dt
    end
    print('Time = ' .. Time)
end
-- Saves initial state after the Geostatic step
io.addResultToMeshFile(file, 0.0)
setCurrentTime(0.0)
```

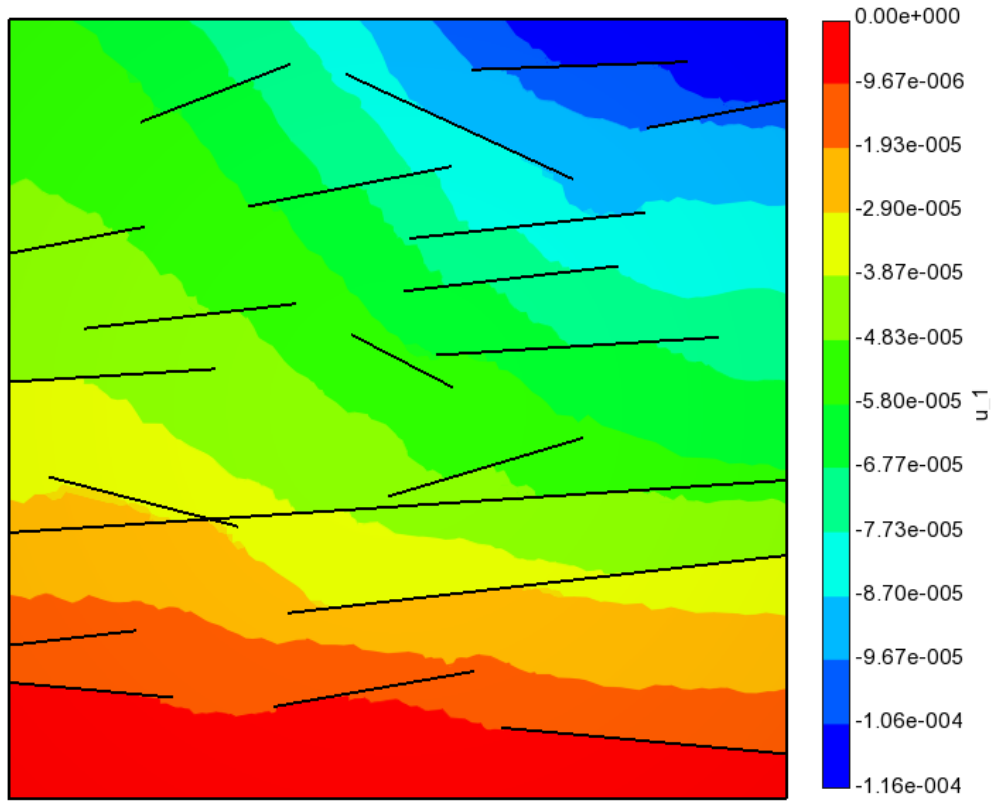
Solution file: Orchestration script for Transient step

```
-- Solves the Geostatic step using a transient nonlinear solver
local solver = fem.init({'interfaceHM', 'Consolidation'}, 'solver', solverOptions2)
local dt        = solverOptions2.timeInitIncrement
local FinalTime = solverOptions2.timeMax
local Time      = dt
local LastStep  = false
while (Time <= FinalTime) do
   print('------------------')
   print(tr('New time increment - time = %1 s'):num(Time))
   print('------------------')
   -- Run fully coupled analysis.  Solver returns a suggested time-step
   -- for the next iteration
   dt = fem.step(solver, dt)
   -- Save results
   io.addResultToMeshFile(file, Time)
   -- Adjust time to guarantee that the last iteration will be on the
   -- requested final simulation time
    if ((Time + dt) >= FinalTime and not LastStep) then
       dt       = FinalTime - Time
       Time     = FinalTime + 0.0001
       LastStep = true
   else
       Time = Time +  dt
    end
end
-- Close output file
io.closeMeshFile(file)
```

Results at time = 1.0 e6 s



Pore pressure (kPa)

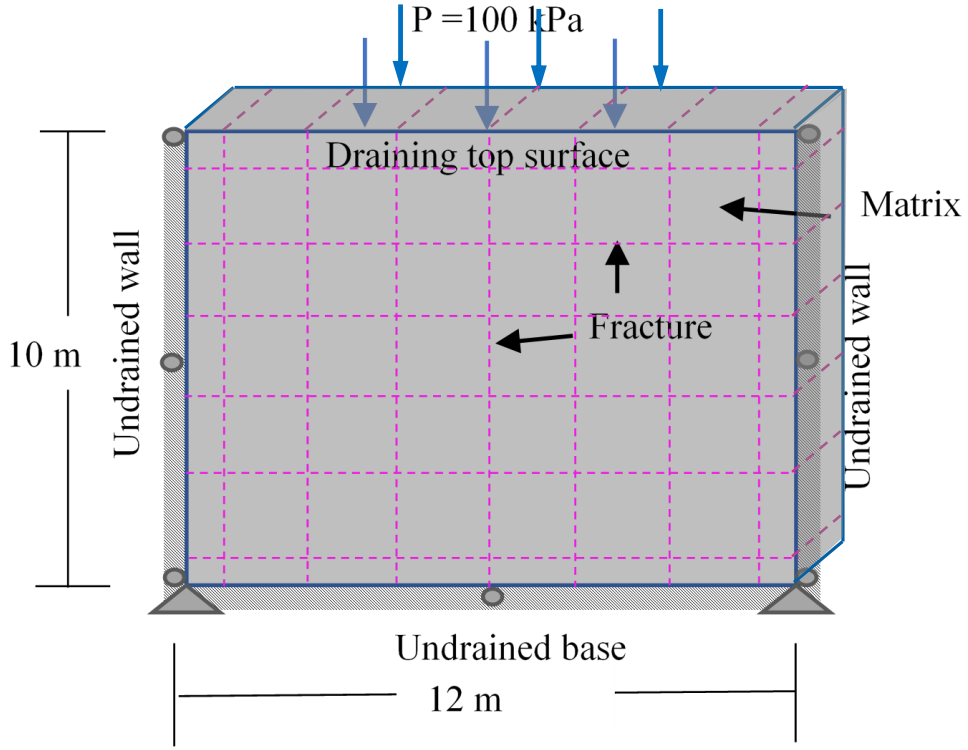


Vertical displacement (m)

3 – CONSOLIDATION OF 3D FRACTURED POROUS MEDIA

The problem

This problem is a 3D representation of problem 1, a rectangular block of fractured porous media is subjected to pressure along 4 m on the top of the model. Base is constrained in z-direction, left and right side are constrained in x-direction, and front and back side are constrained in y-direction.

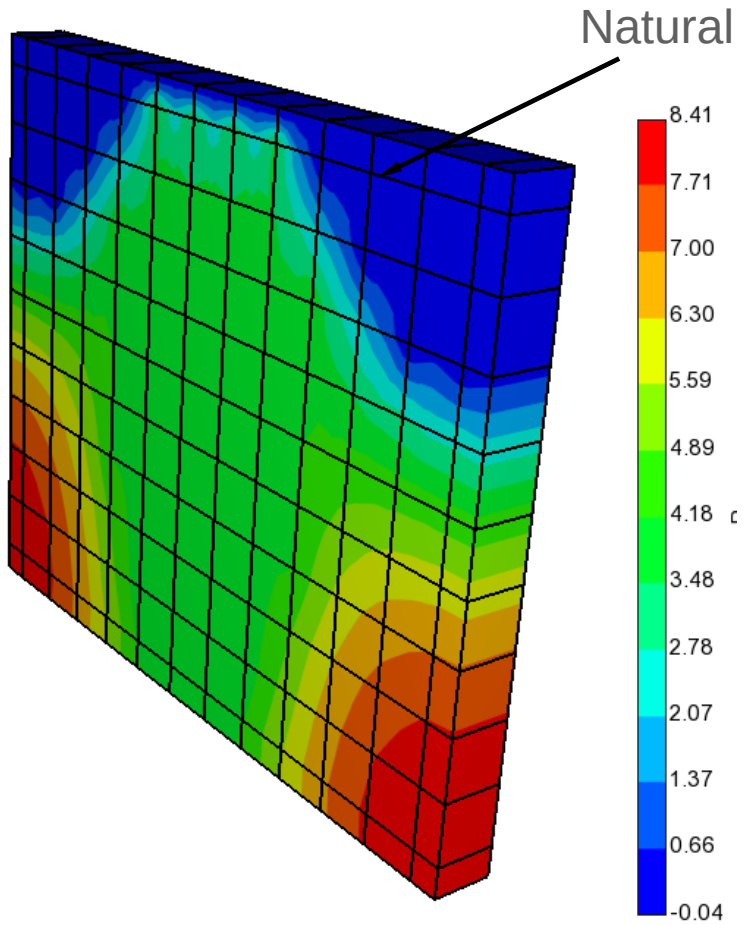


Natural fractures are represented by 3D zero-thickness interface elements.

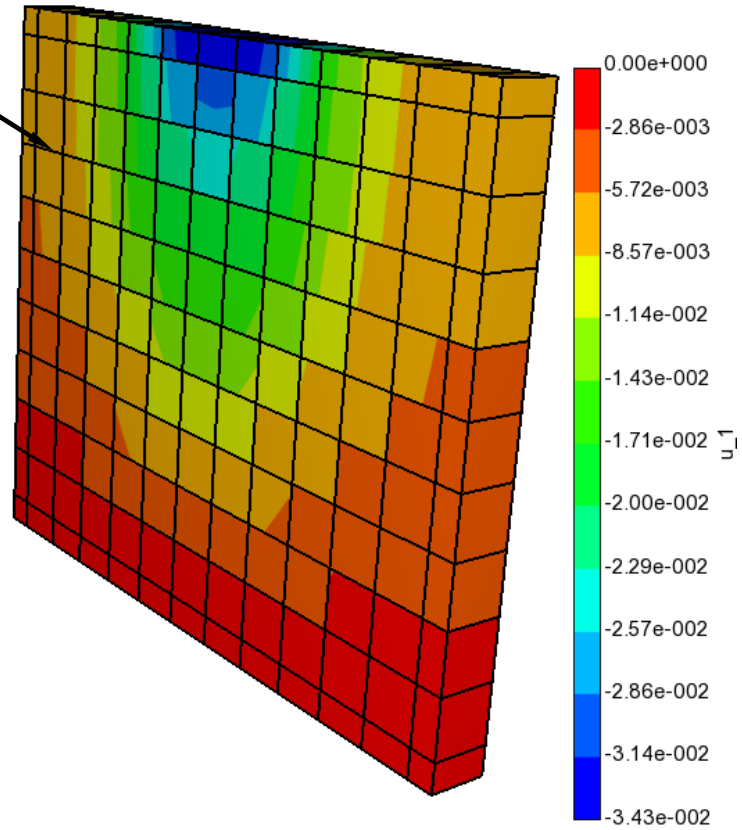
Simulation file: DFM3Dkvar.lua

Material values and the orchestration technique are essentially the same as presented on example 1. See the simulation files for details.

Results at time = 1.0e9 s



Pore pressure (kPa)

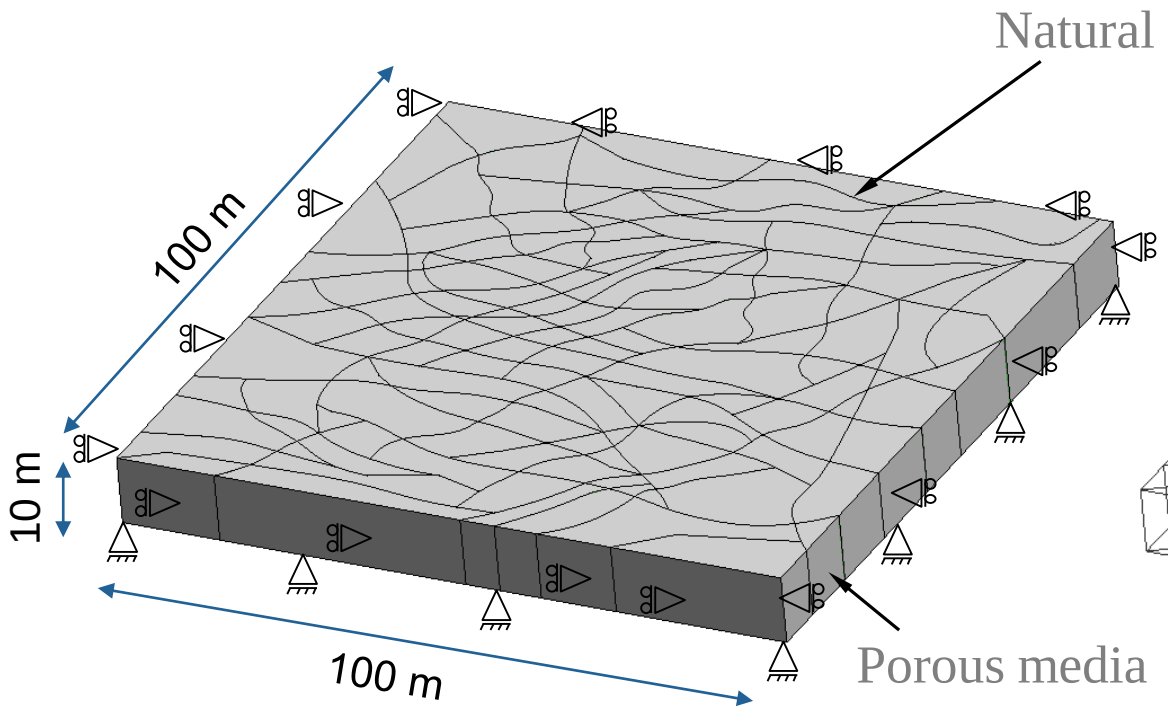


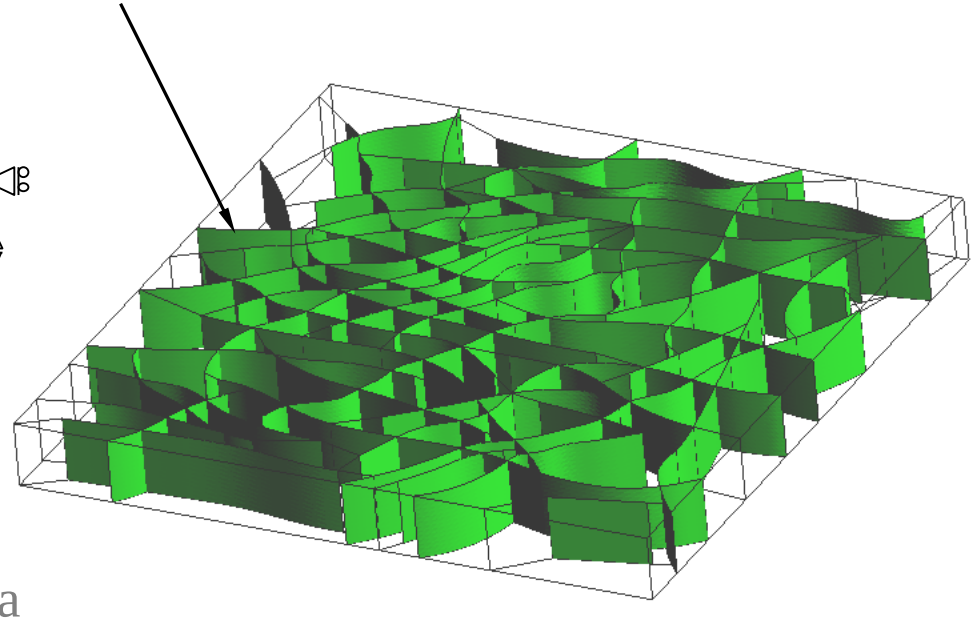
Vertical displacement (m)

4 – FLUID FLOW IN 3D FRACTURED POROUS MEDIA

The problem

A block of fractured porous media (100x100x10) under in-situ condition (stresses and pore pressure) is subjected to pore pressure gradient in x-direction. The impact of fractures on the fluid flow process is investigated.





$k_0 = 8.63\ 10^{-4}$ m/d
$n_0 = 0.1$

E = 12000 MPa
$\nu = 0.3$

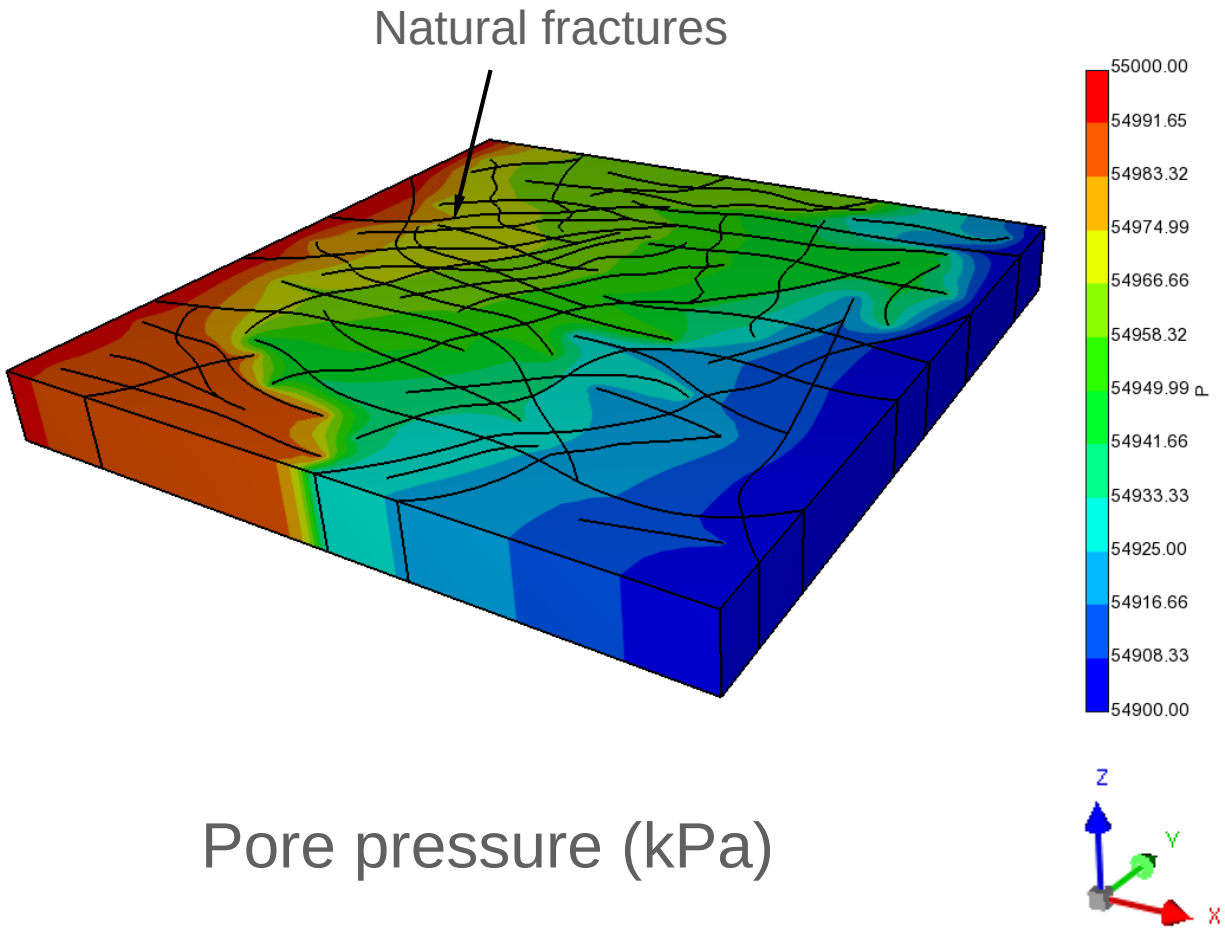
$p_0 = 55$ MPa
$\sigma_0 = 85$ MPa

$\gamma_w = 0.01$ MN/m^3
$\mu_w = 10^{-9}$ MPa.s
$\beta_w = 0.000458$ MPa^{-1}

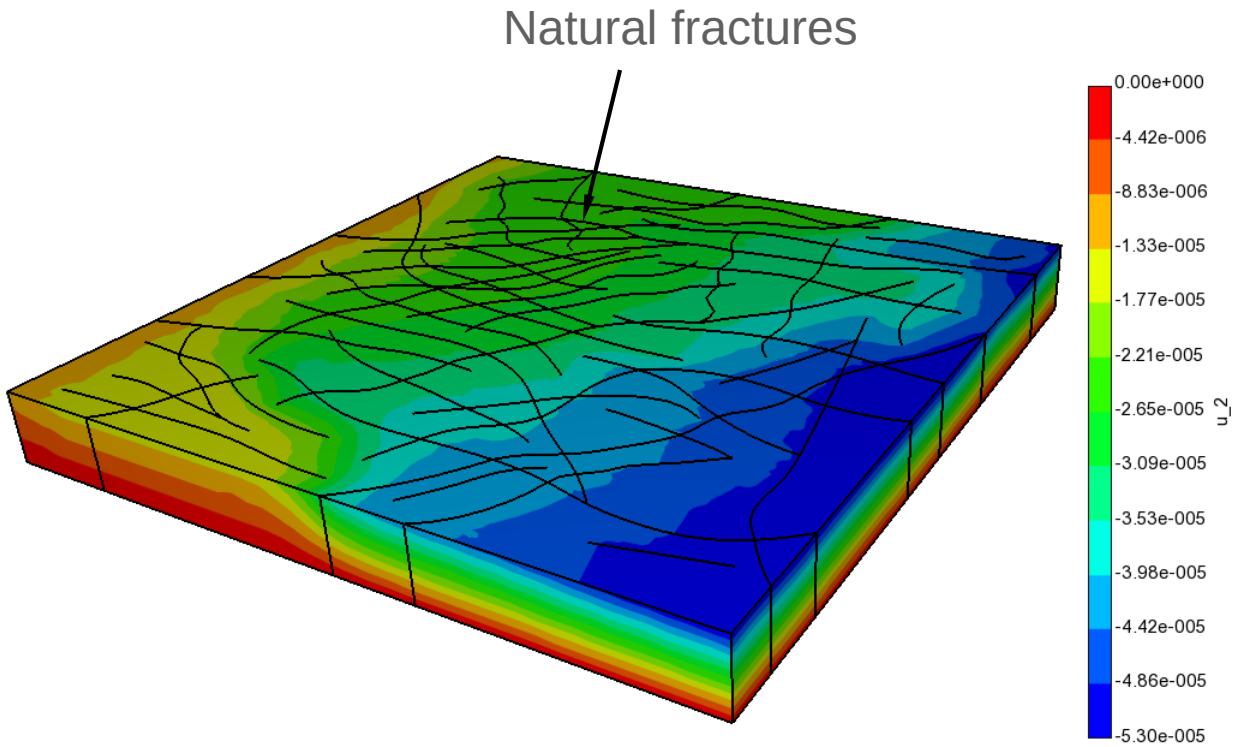
Material values and the orchestration technique are essentially the same as presented on example 2. See the simulation files for details.

Simulation file: naturalFractures3D.lua

Results at time = 1.0 e6 s

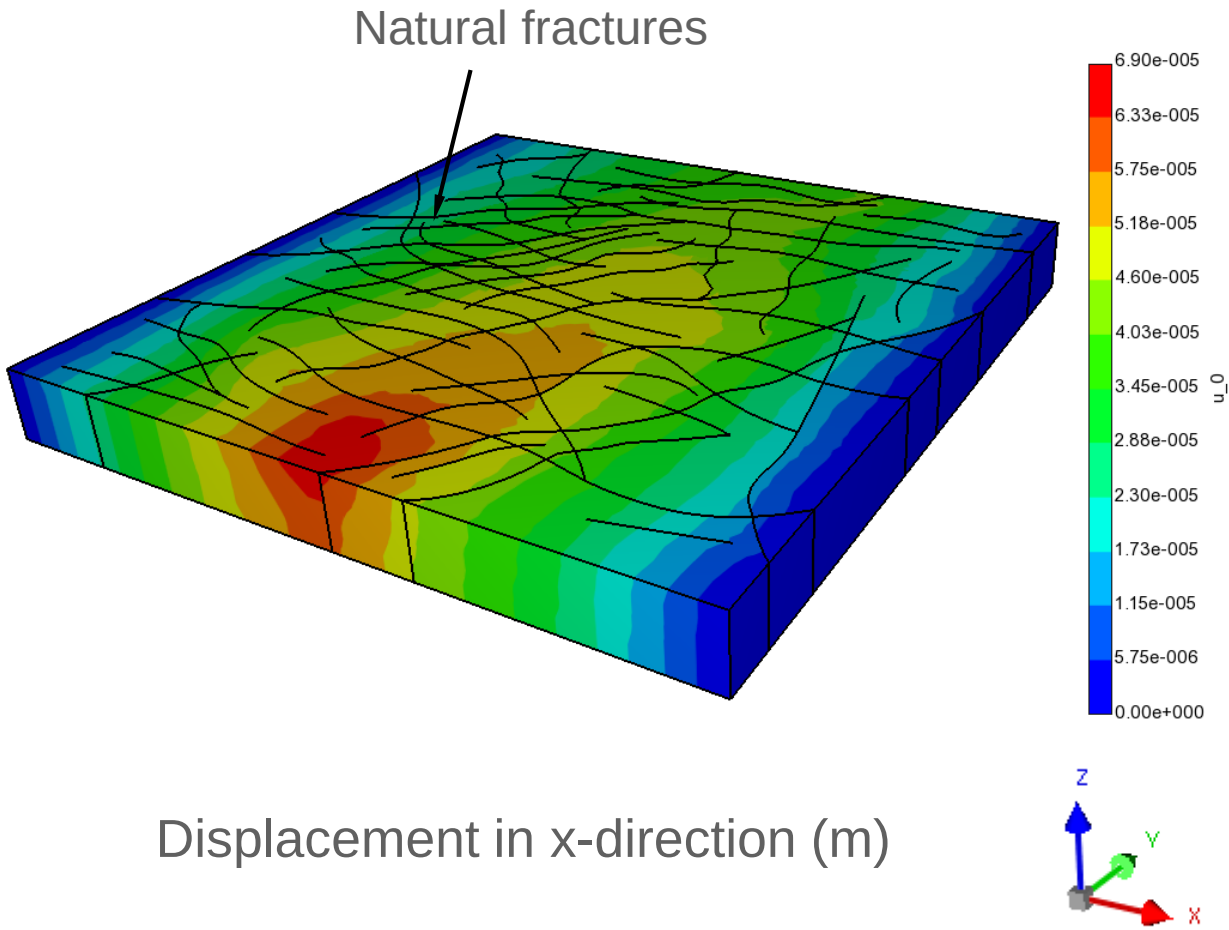


Pore pressure (kPa)

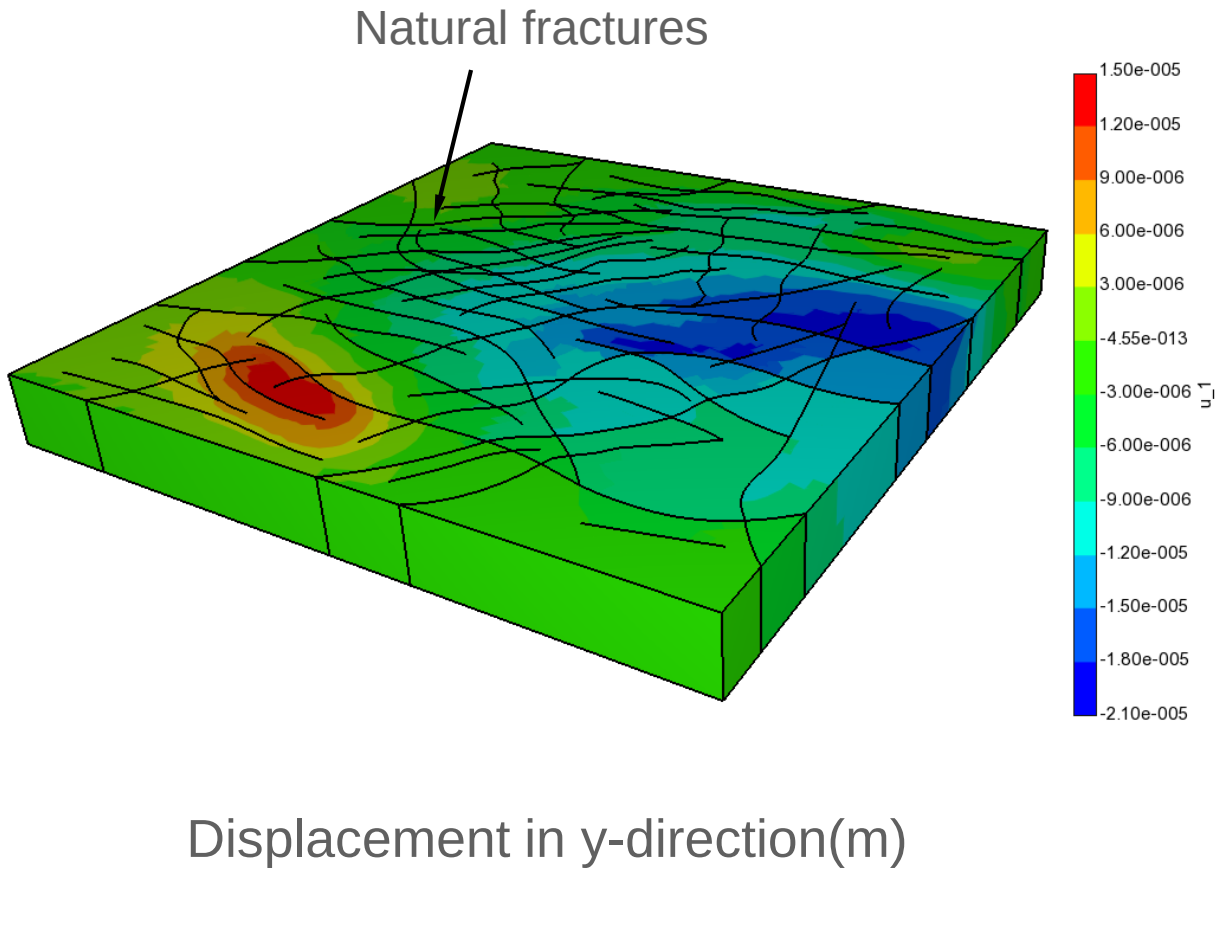


Vertical displacement (m)

Results at time = 1.0 e6 s

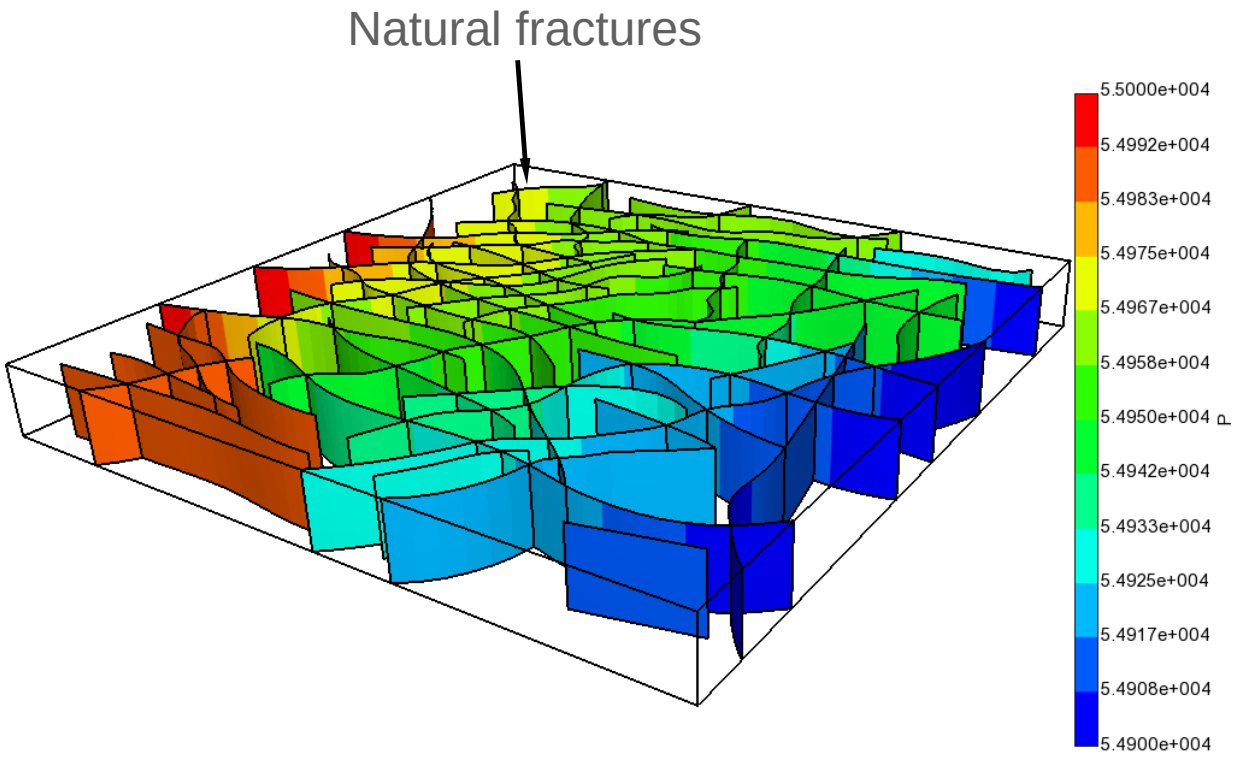


Displacement in x-direction (m)

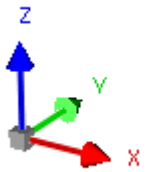


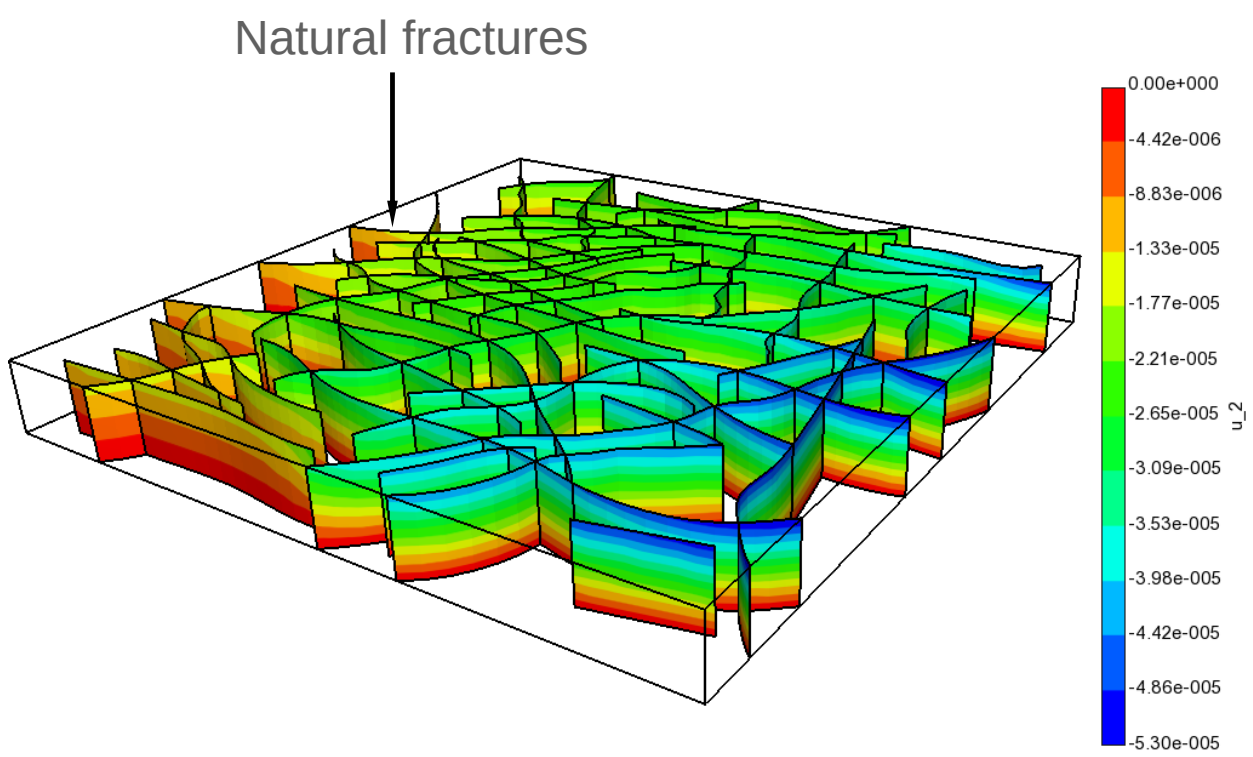
Displacement in y-direction(m)

Results at time = 1.0 e6 s



Pore pressure (kPa)





Vertical displacement (m)