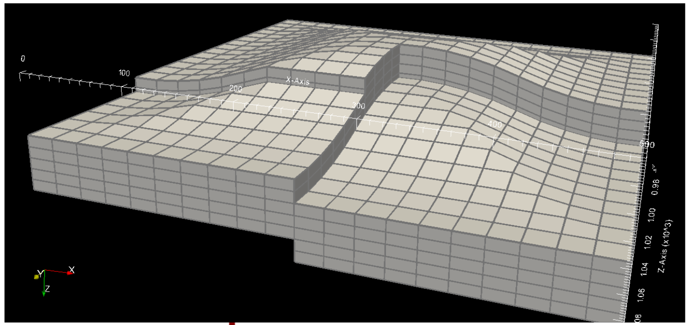
MUFITS Training Course

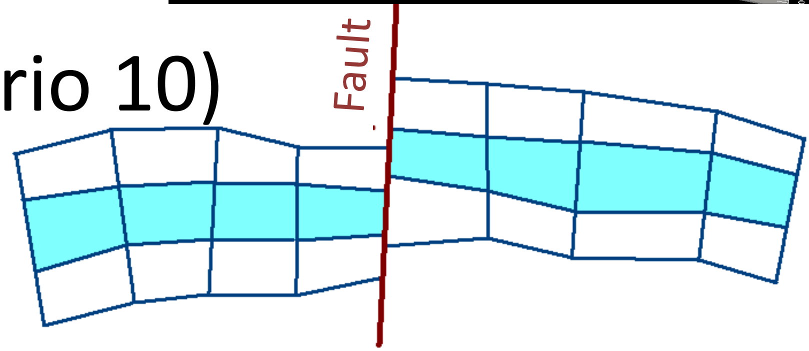
Day 5

Corner-point grids, Faults, Aquifers & Onshore/offshore

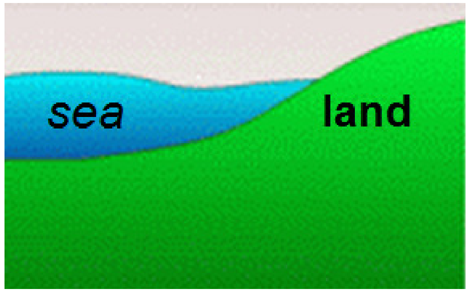
Program



- Corner-point grids (Scenario 10)

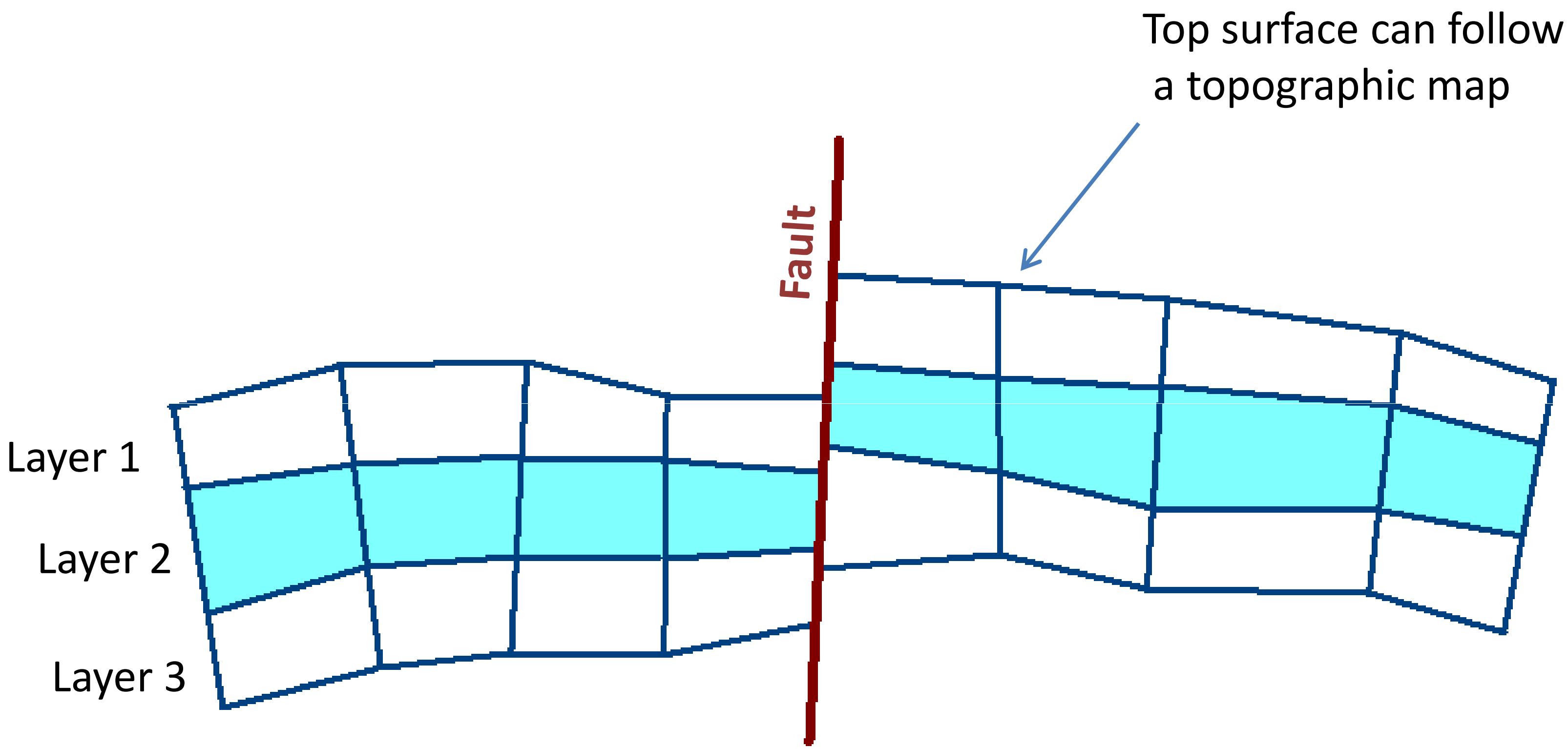


- Faults
- Modeling aquifers (Scenario 11)
- Onshore/offshore boundary conditions (Scenario 12)



Corner-point grids

Corner-point grids



Cross-section view

Corner-point grids

MAKE-ENDMAKE syntax

```
-- in GRID section

MAKE
  gridtype  ni nj nk /

-- other keywords

ENDMAKE

================================================================================


    gridtype = CART    - Cartesian Grid
             = RADIAL - Radial Grid
             = CORNER - Corner-Point grid

          ni - number of grid blocks along i-indexation axis
          nj - number of grid blocks along j-indexation axis
          nk - number of grid blocks along k-indexation axis
```

Select this option

CARTesian/RADIAL to CORNER

The simulator automatically converts the Cartesian and Radial grids into Corner-Point format within MAKE-ENDMAKE brackets. The corner-point grid can be exported by the SAVECPG keyword. The saved grid file can be later used in another simulation. The grid file can also be created/modified by a third-party software and be imported in the simulator by using INCLUDE keyword.

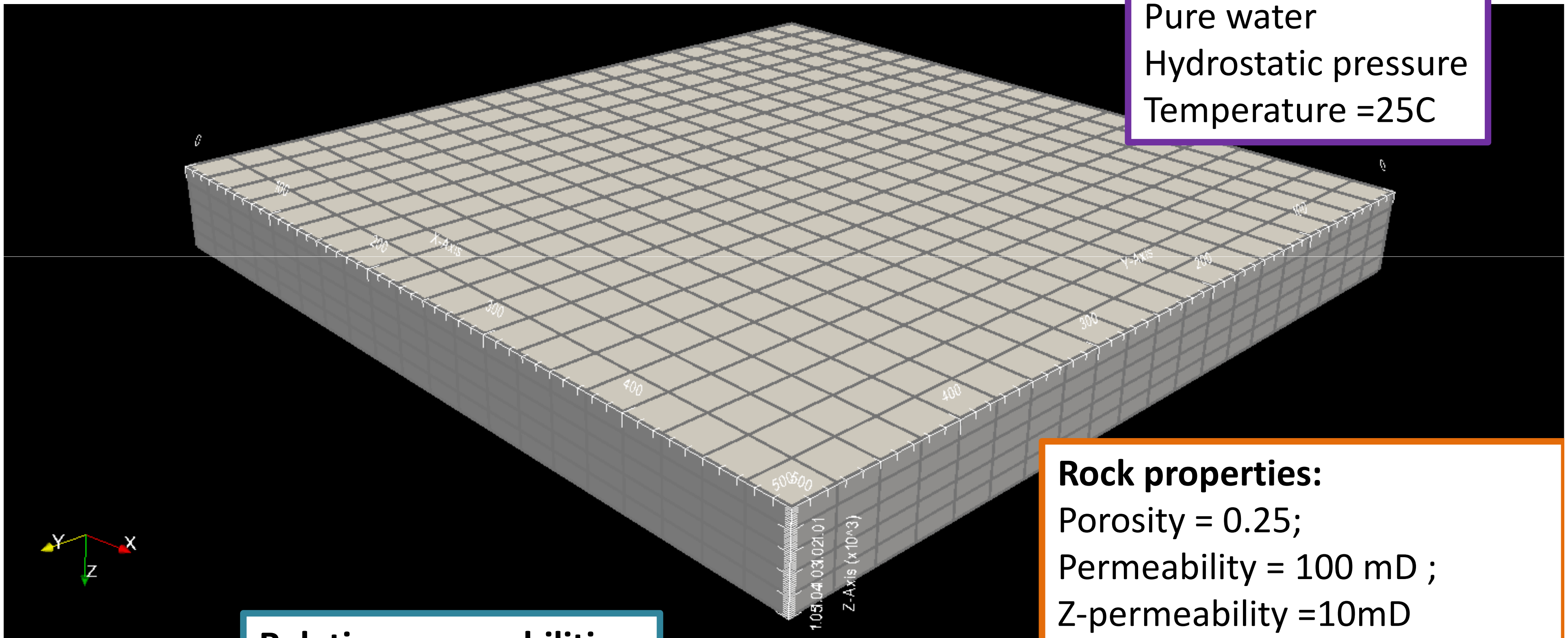
The SAVECPG keyword saves formatted grid file.

SAVECPG syntax

```
-- within MAKE-ENDMAKE brackets

SAVECPG
   filename  imin imax   jmin jmax  kmin kmax  /

==================================================================================

     filename  - output file name
     imin/imax - the boundaries of the box along i-index axis, for which the
                 grid file is saved.
     jmin/jmax - the boundaries of the box along j-index axis, for which the
                 grid file is saved.
     kmin/kmax - the boundaries of the box along k-index axis, for which the
                 grid file is saved.
```

Scenario 10

Grid: 20*20*5
Domain: [0,500]*[0,500]*[1000,1050] m.
T2EOS1 module



Init. cond:
Pure water
Hydrostatic pressure
Temperature =25C

Rock properties:
Porosity = 0.25;
Permeability = 100 mD ;
Z-permeability =10mD
Rock density = 2900 kg/m^3;
Heat capacity = 1.0 kJ/kg/K;
Heat conduct. = 2 W/m/K.

Relative permeabilities:
See the RUN-file

RUN-file (scenario 10)

1. Open RUN-file in text editor
2. Run the simulation
3. Open results in ParaView

COORD keyword

COORD syntax

```
-- within brackets MAKE-ENDMAKE

COORD
   xa1 ya1 za1   xb1 yb1 zb1 /
   xa2 ya2 za2   xb2 yb2 zb2 /
   xa3 ya3 za3   xb3 yb3 zb3 /
...
   xaN yaN zaN   xbN ybN zbN /
/

================================================================================

    xa#-ya#-zb# and  - coordinates of two different points on a pillar
    xb#-yb#-zb#
                  N  - the total number of pillars in the current input box.
                       N=(imax-imin+2)*(jmax-jmin+2). The i-index is cycling
                       the fastest following by the j-index.
```

ZCORN keyword

ZCORN syntax

```
-- within MAKE-ENDMAKE brackets

ZCORN
  depth1 depth2 depth3 ... depthN /

=================================================================================


    depth# - depth of a grid block corner.
         N - the total number of the grid block corners in the current input
             box. N=2*(imax-imin+1)*2*(jmax-jmin+1)*2*(kmax-kmin+1).
```

Exercise

Exercise: Re-simulate scenario 10 exporting the grid in Corner-point format. Open the grid file and inspect it.

ADDZCORN keyword

Can be used to alter the corner depths in a box of grid blocks

ADDZCORN syntax

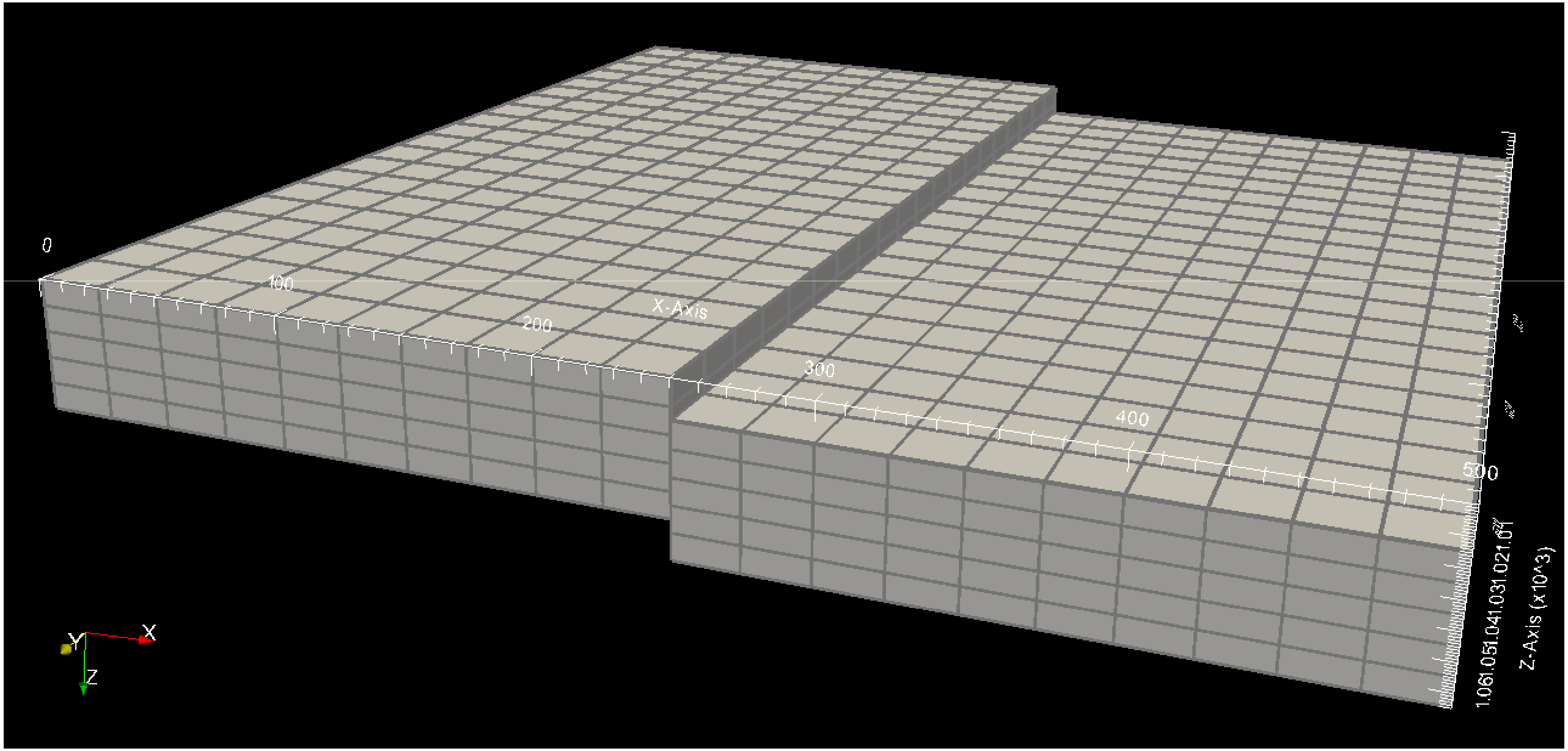
```
-- within MAKE-ENDMAKE brackets

ADDZCORN
  value1  imin1 imax1 jmin1 jmax1 kmin1 kmax1 m1_1 m2_1 m3_1 m4_1 m5_1 m6_1 /
  value2  imin2 imax2 jmin2 jmax2 kmin2 kmax2 m1_2 m2_2 m3_2 m4_2 m5_2 m6_2 /
  value3  imin3 imax3 jmin3 jmax3 kmin3 kmax3 m1_3 m2_3 m3_3 m4_3 m5_3 m6_3 /
  ...
/

================================================================================


   value#      - the value added to ZCORN array in the input box;
   imin#/imax# - the boundaries of the input box along i-indexation axis.
                 By default these values are equal to the arguments 1 and 2
                 of the keyword BOX.
   jmin#/jmax# - the boundaries of the input box along j-indexation axis.
                 By default these values are equal to the arguments 3 and 4
                 of the keyword BOX.
   kmin#/kmax# - the boundaries of the input box along k-indexation axis.
                 By default these values are equal to the arguments 5 and 6
                 of the keyword BOX.
   mi_#        - (i=1,...,6). The mode:
                 'I-' - the operation is also applied to the adjacent face of
                        the grid block connected to the considered block in
                        the input box in the negative direction of i-index
                        axis.
```

See also EQLZCORN

See full description in the Reference manual

Exercise

Exercise: Create the following grid using ADDZCORN



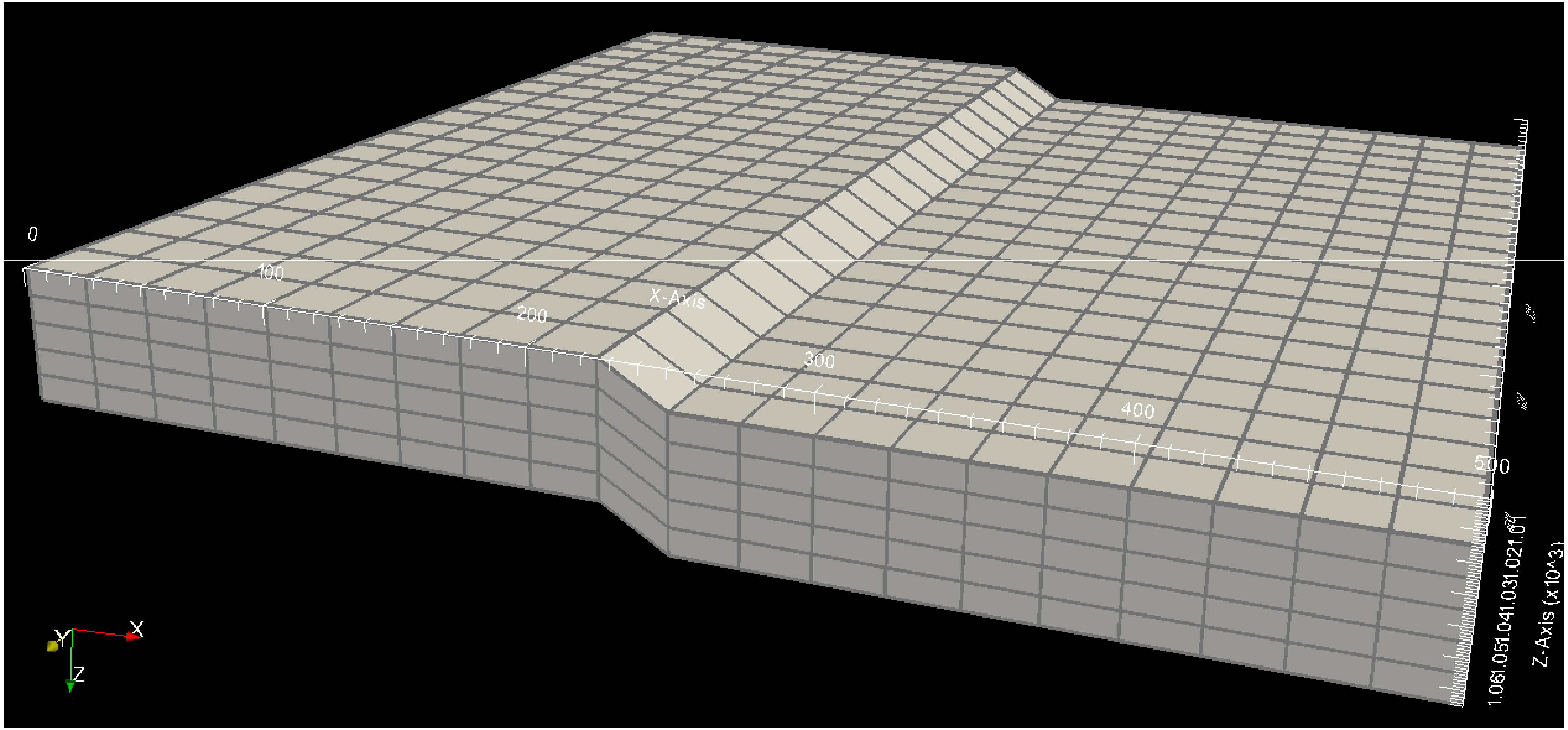
Answer

SIMULATIONS/SCENARIO10/0/TASK1.INC

```
ADDZCORN
   15.0  11 20 4*  15.0 /
/
```

Exercise

Exercise: Create the following grid using ADDZCORN



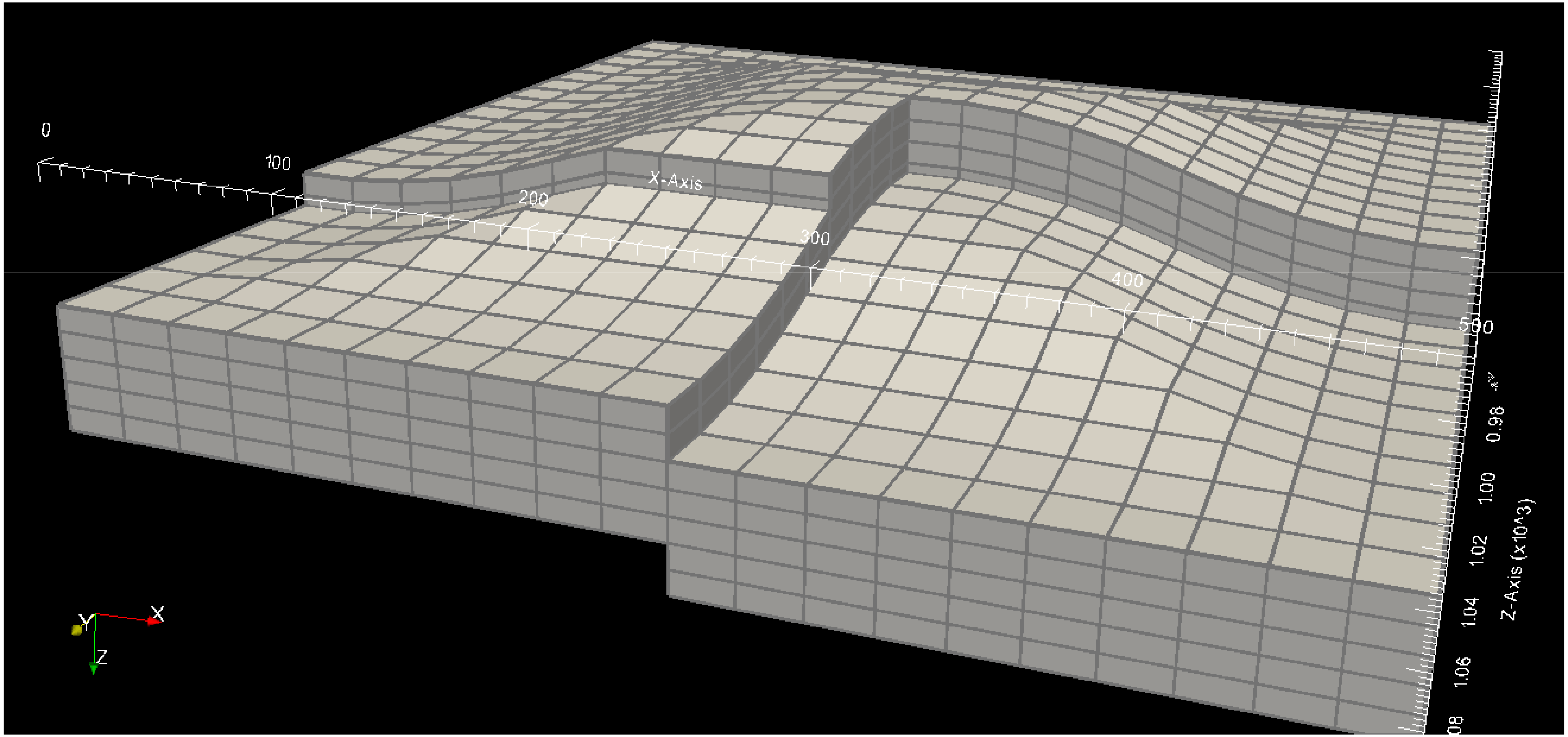
Answer

SIMULATIONS/SCENARIO10/0/TASK2.INC

```
ADDZCORN
   15.0  11 20 4*  15.0  'I-' /
/
```

Exercise

Exercise: Create the following grid using ADDZCORN



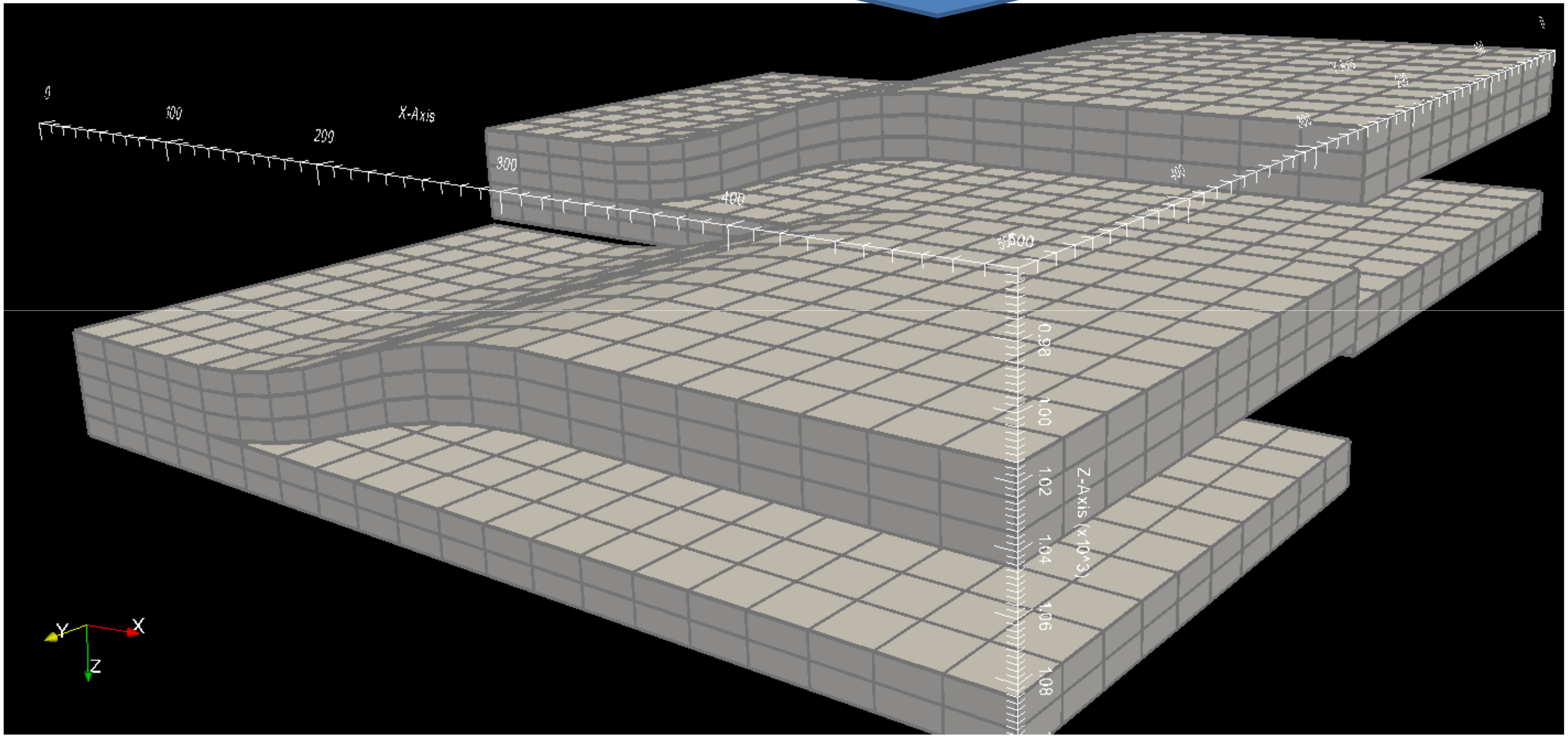
Answer

SIMULATIONS/SCENARIO10/0/TASK3.INC

```
ADDZCORN
   -2.0   3 18   3 18  2*  'I-' 'I+' 'J-' 'J+' /
   -4.0   4 17   4 17  2*  'I-' 'I+' 'J-' 'J+' /
   -6.0   5 16   5 16  2*  'I-' 'I+' 'J-' 'J+' /
   -8.0   6 15   6 15  2*  'I-' 'I+' 'J-' 'J+' /
   -8.0   7 14   7 14  2*  'I-' 'I+' 'J-' 'J+' /
   -6.0   8 13   8 13  2*  'I-' 'I+' 'J-' 'J+' /
   -4.0   9 12   9 12  2*  'I-' 'I+' 'J-' 'J+' /
   -2.0  10 11  10 11  2*  'I-' 'I+' 'J-' 'J+' /
   35.0  11 20  11 20  2* /
   15.0   1 10  15 20  2* /
/
```

Exercise

Exercise: Create the following grid using ADDZCORN



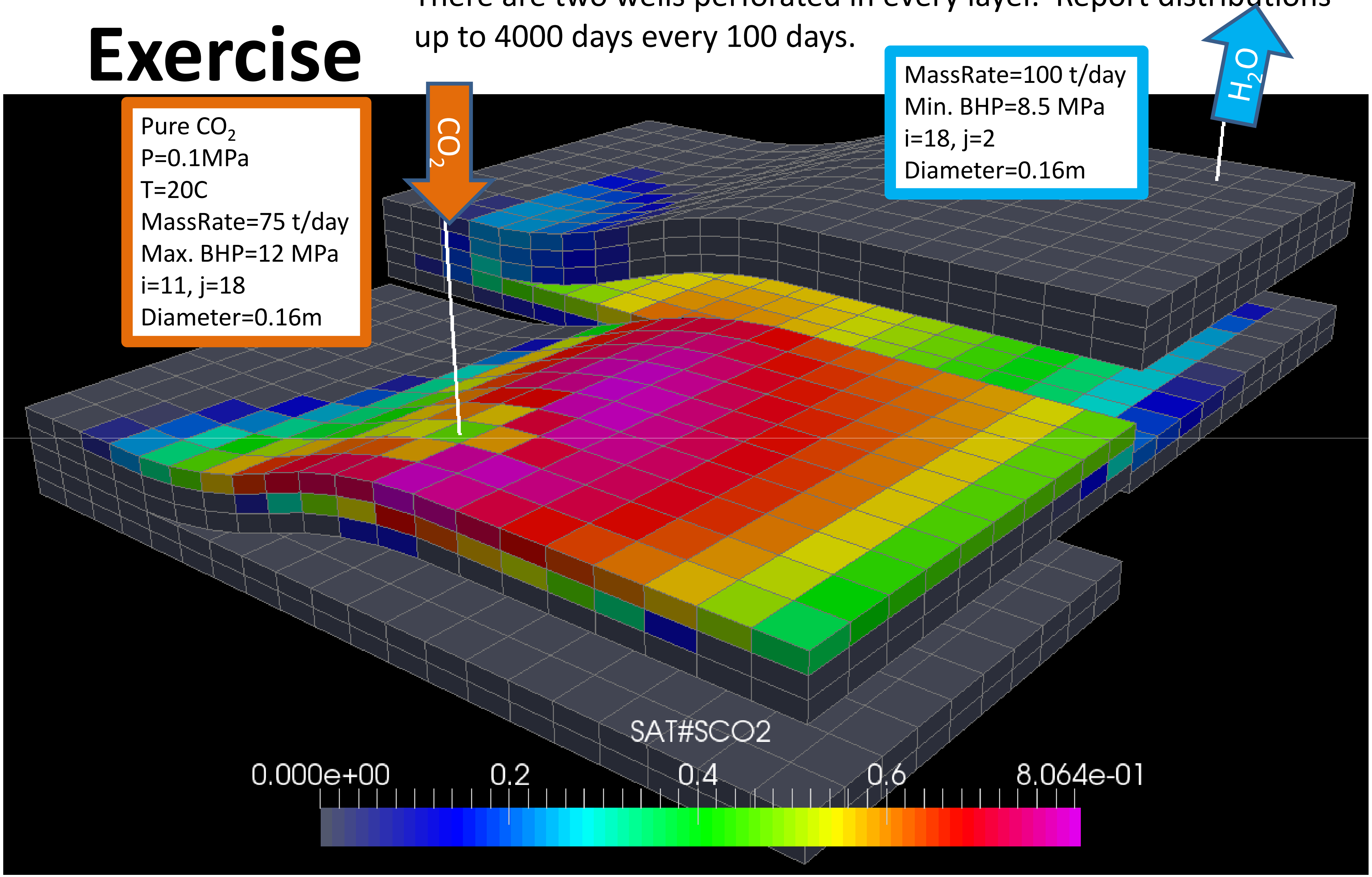
Answer

SIMULATIONS/SCENARIO10/0/TASK4.INC

```
ADDZCORN
   -2.0   6 20  2*    1 3 'I-' 'I+'/
   -4.0   7 30  2*    1 3 'I-' 'I+'/
   -6.0   8 20  2*    1 3 'I-' 'I+'/
   -8.0   9 20  2*    1 3 'I-' 'I+'/
   -8.0  10 20  2*    1 3 'I-' 'I+'/
   -6.0  11 20  2*    1 3 'I-' 'I+'/
   -4.0  12 20  2*    1 3 'I-' 'I+'/
   -2.0  13 20  2*    1 3 'I-' 'I+'/
   55.0   1 20  11 20 1 5 /
/
```

Exercise

Using the last grid simulate injection of CO_2 using BINMIXT module. There are two wells perforated in every layer. Report distributions up to 4000 days every 100 days.



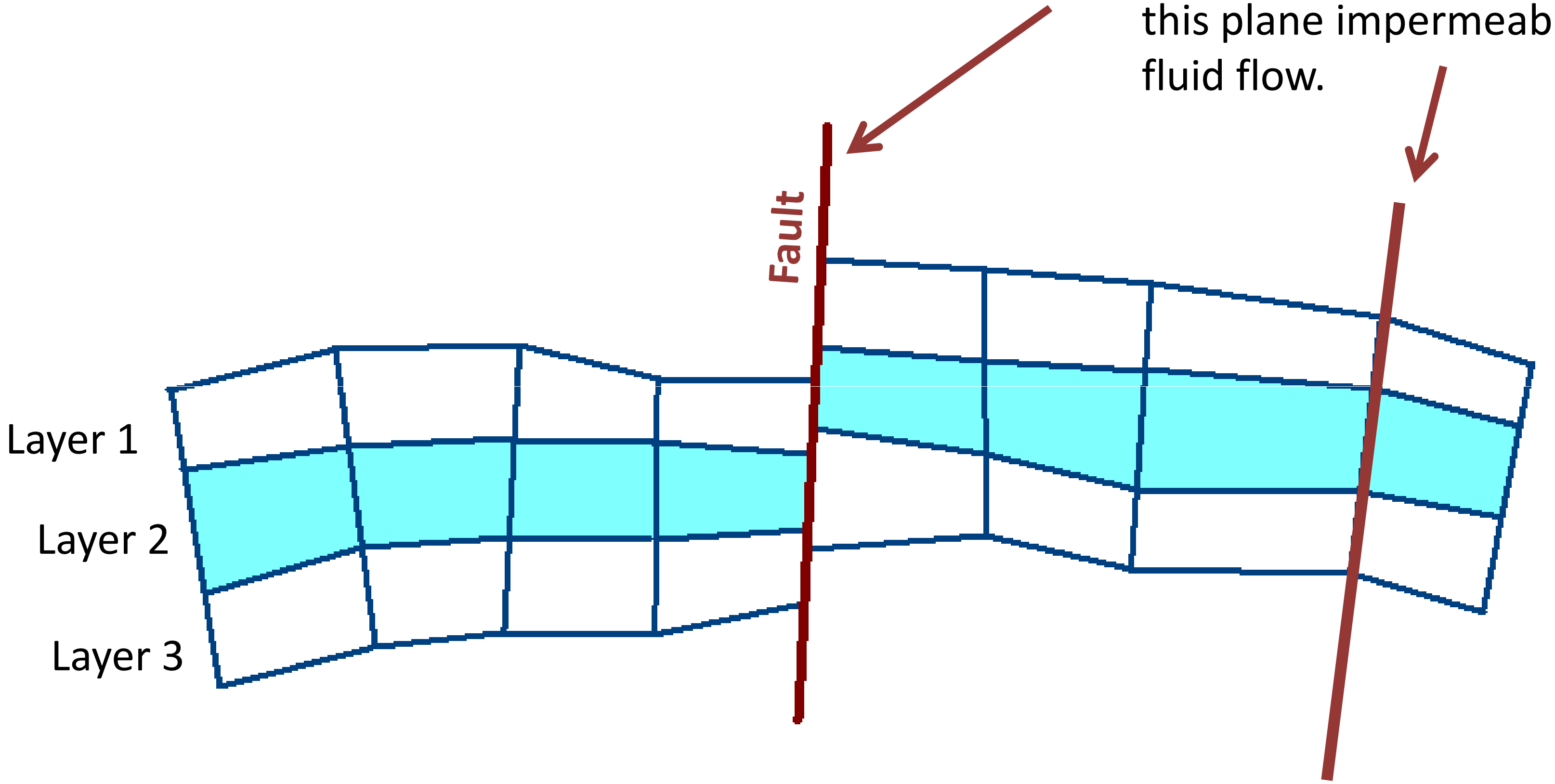
RUN-file (scenario 10, Exercise)

1. Open RUN-file in text editor
2. Run the simulation
3. Open results in ParaView

Faults

Faults

Using the Faults option you can reduce permeability (transmissibility) across the fault plane, or you can make this plane impermeable for fluid flow.



Cross-section view

FAULTS keyword

The fault faces are introduced using the FAULTS keyword.

FAULTS syntax

```
-- within MAKE-ENDMAKE brackets

FAULTS
   name1 imin1 imax1 jmin1 jmax1 kmin1 kmax1 face1 /
   name2 imin2 imax2 jmin2 jmax2 kmin2 kmax2 face2 /
   name3 imin3 imax3 jmin3 jmax3 kmin3 kmax3 face3 /
   ...
/

================================================================================

   name#       - a character ID of the fault;
   imin#-imax# - the boundaries of the input box along i-indexation axis.
                 By default these values are equal to '1' and the 2nd
                 argument of the keyword MAKE, respectively;
   jmin#-jmax# - the boundaries of the input box along j-indexation axis.
                 By default these values are equal to '1' and the 3rd
                 argument of the keyword MAKE, respectively;
   kmin#-kmax# - the boundaries of the input box along k-indexation axis.
                 By default these values are equal to '1' and the 4th
                 argument of the keyword MAKE, respectively;
   face#       - falut face tag. Must be one of
                 'I-','X-' - fault face coincides with the grid block fac
                             the negative direction of the i-index coordinate
                             line;
```

See full description in the Reference manual

MULTFLT keyword

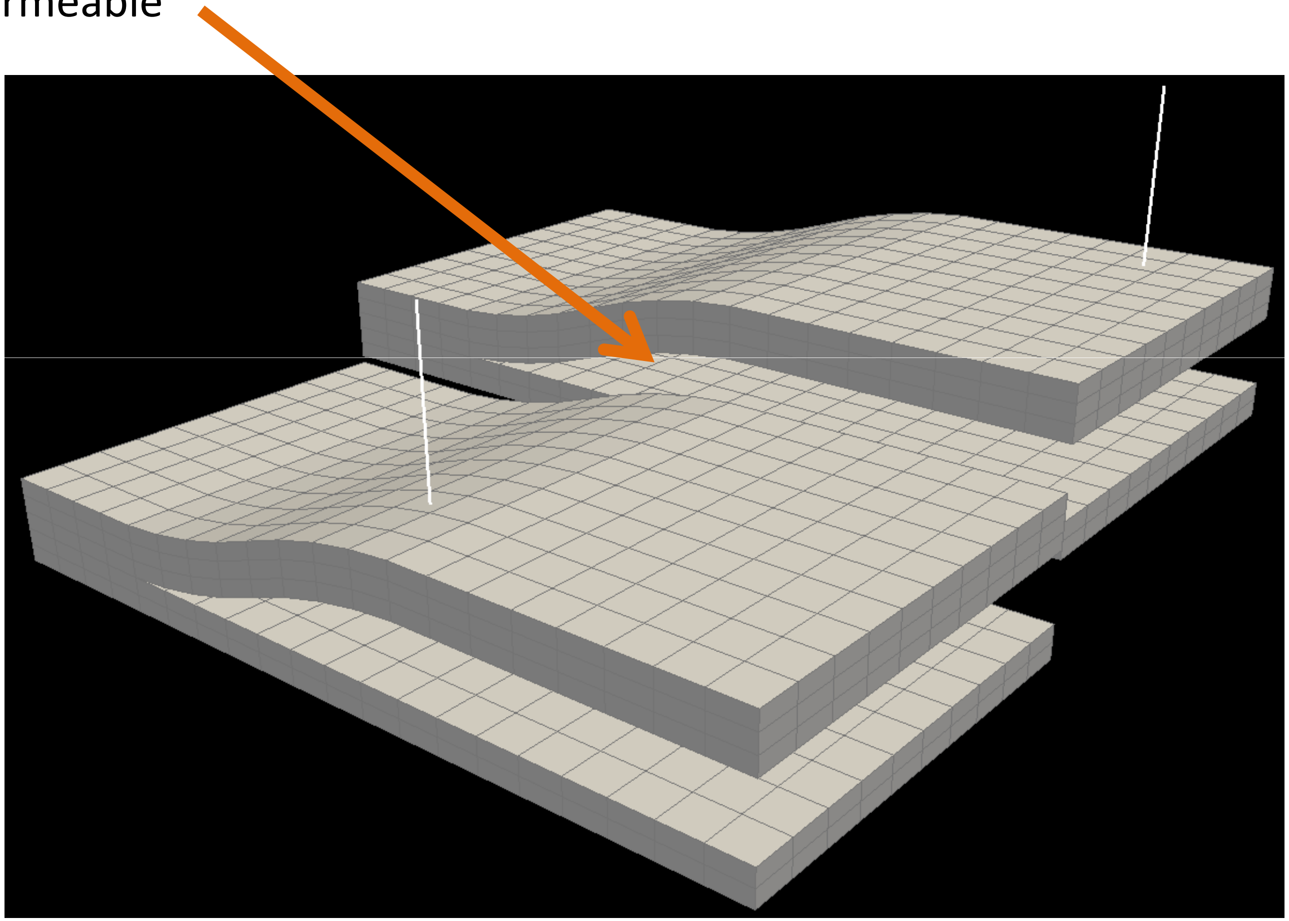
The transmissibility multipliers across the fault are introduced using the MULTFLT keyword.

MULTFLT syntax

```
-- within MAKE-ENDMAKE brackets

FAULT
  name1  mult1 /
  name2  mult2 /
  name3  mult3 /
  ...
/

================================================================================

    name# - character ID of the fault;
    mult# - transmissibility multiplier across the fault.
```

Exercise

Re-simulate the last version of the Scenario 10 (with wells) making the Fault impermeable

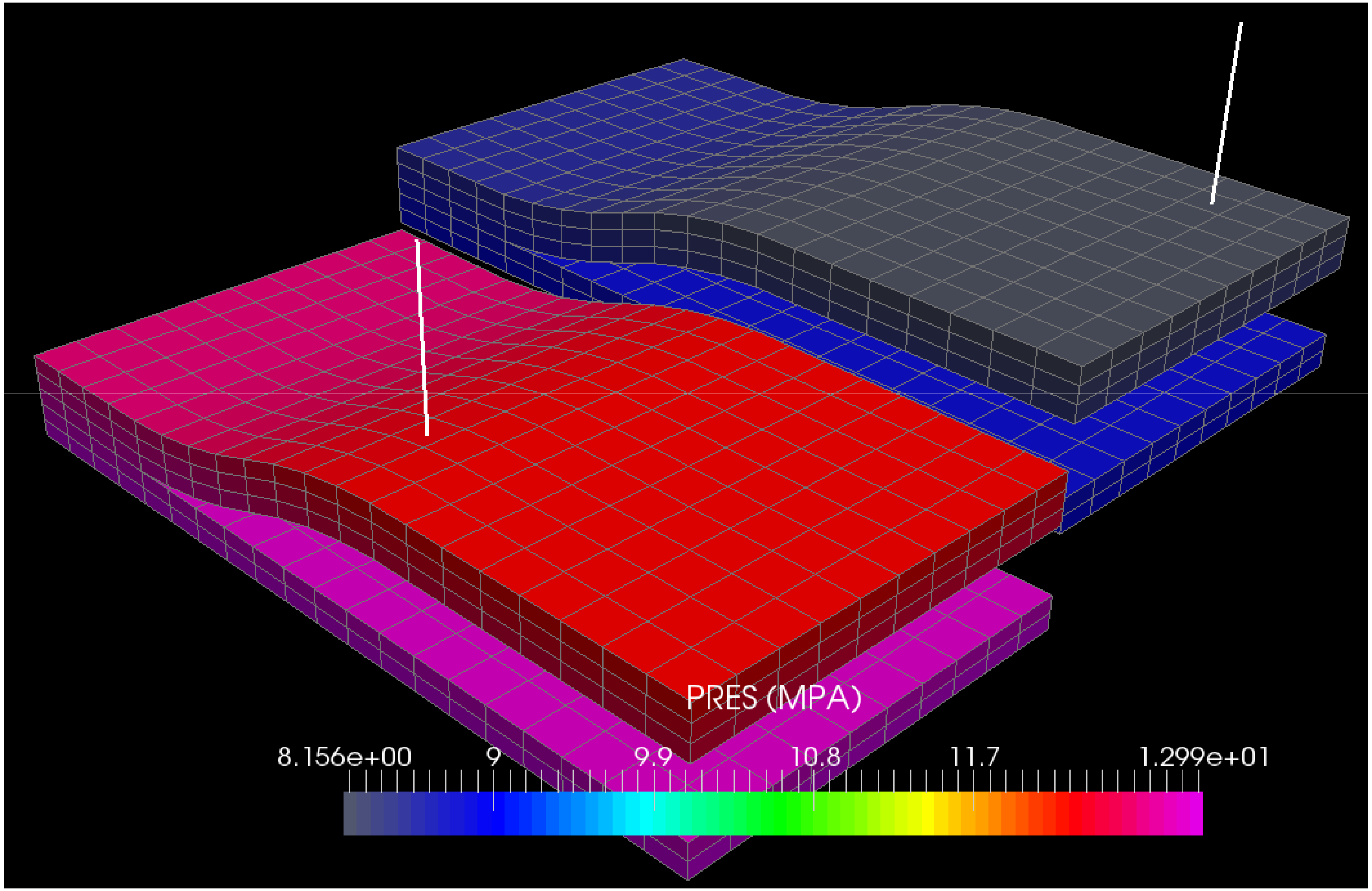


Answer

SIMULATIONS/SCENARIO10/EXERCISE/FAULTS.INC

```
-- within MAKE-ENDMAKE brackets

FAULTS
   'MYFAULT'  1 20  10 10 1 5 'Y+' /
/

MULTFLT
  'MYFAULT' 0.0 /
/
```

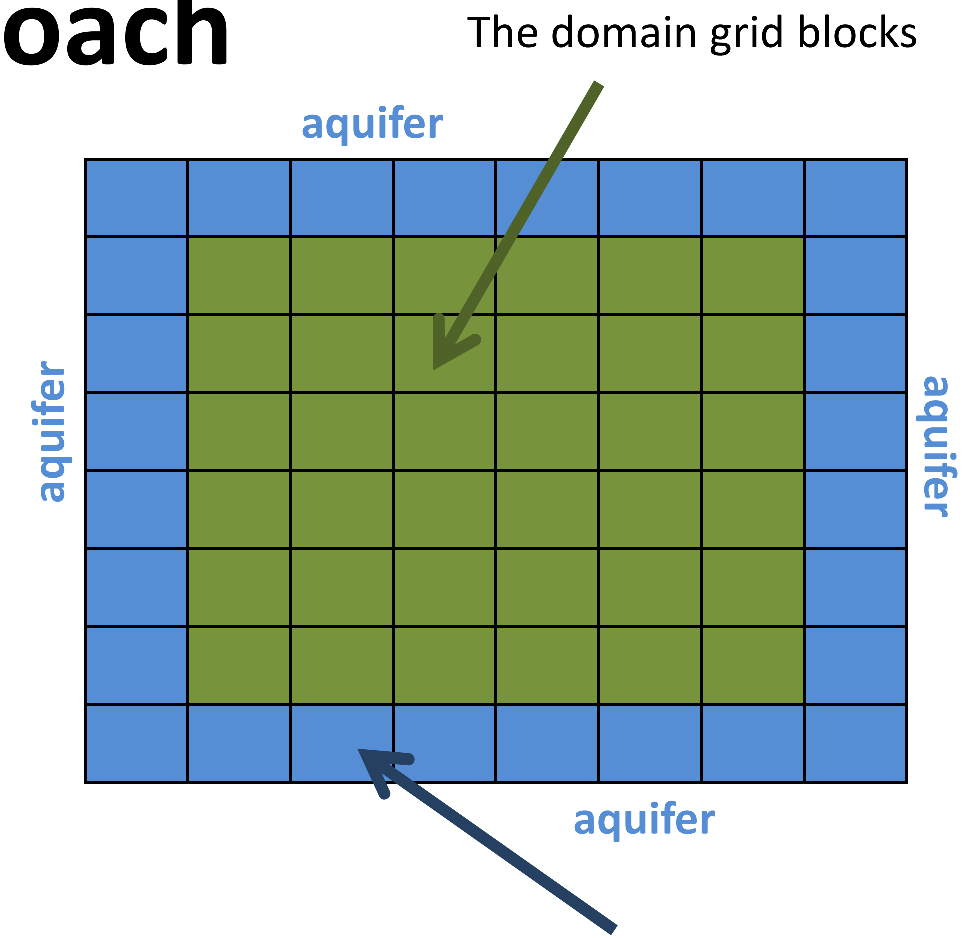
Result of the simulation



Modeling aquifers

Modeling approach

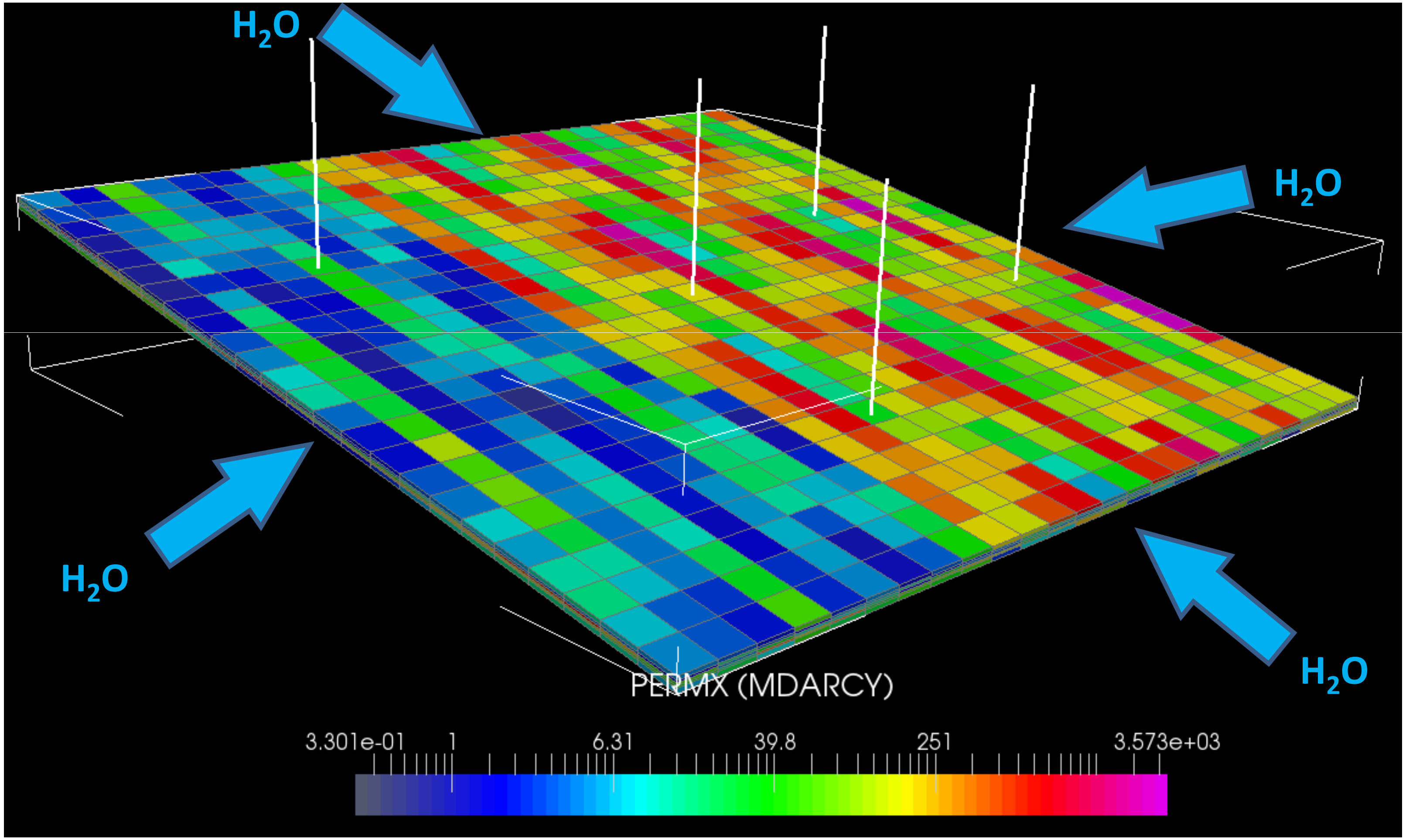
1. Encompass the domain in a circle of boundary grid blocks using SAMESIZE option. The grid blocks must be active.
2. Specify an estimate of initial distribution of pressure (and temperature).
3. Simulate the flow (without modeling injection/production) for a period of time until hydrostatic distribution forms.
4. Fix parameters in boundary grid blocks (set ACTNUM=2).
5. Farther, you can model the injection/production.



The boundary grid blocks used for modeling influx from the aquifer must be created using SAMESIZE option of the BOUNDARY keyword.

Scenario 11

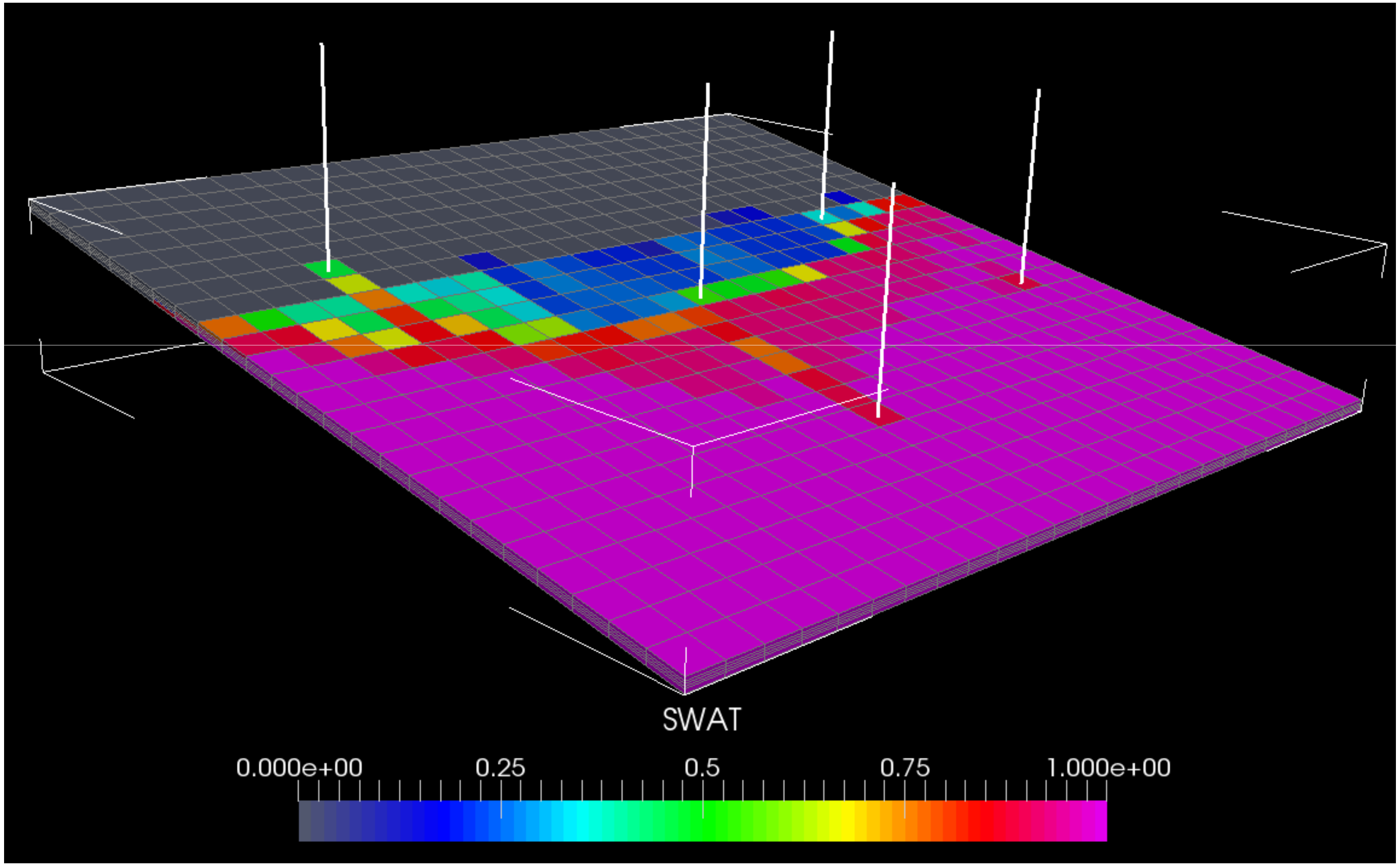
Steam production via 5 production wells. The pressure in the field maintains by the fluid inflow from lateral boundaries (aquifer). The grid, porosity and permeability are loaded via grid file. See the problem description in the RUN-file.



RUN-file (scenario 11)

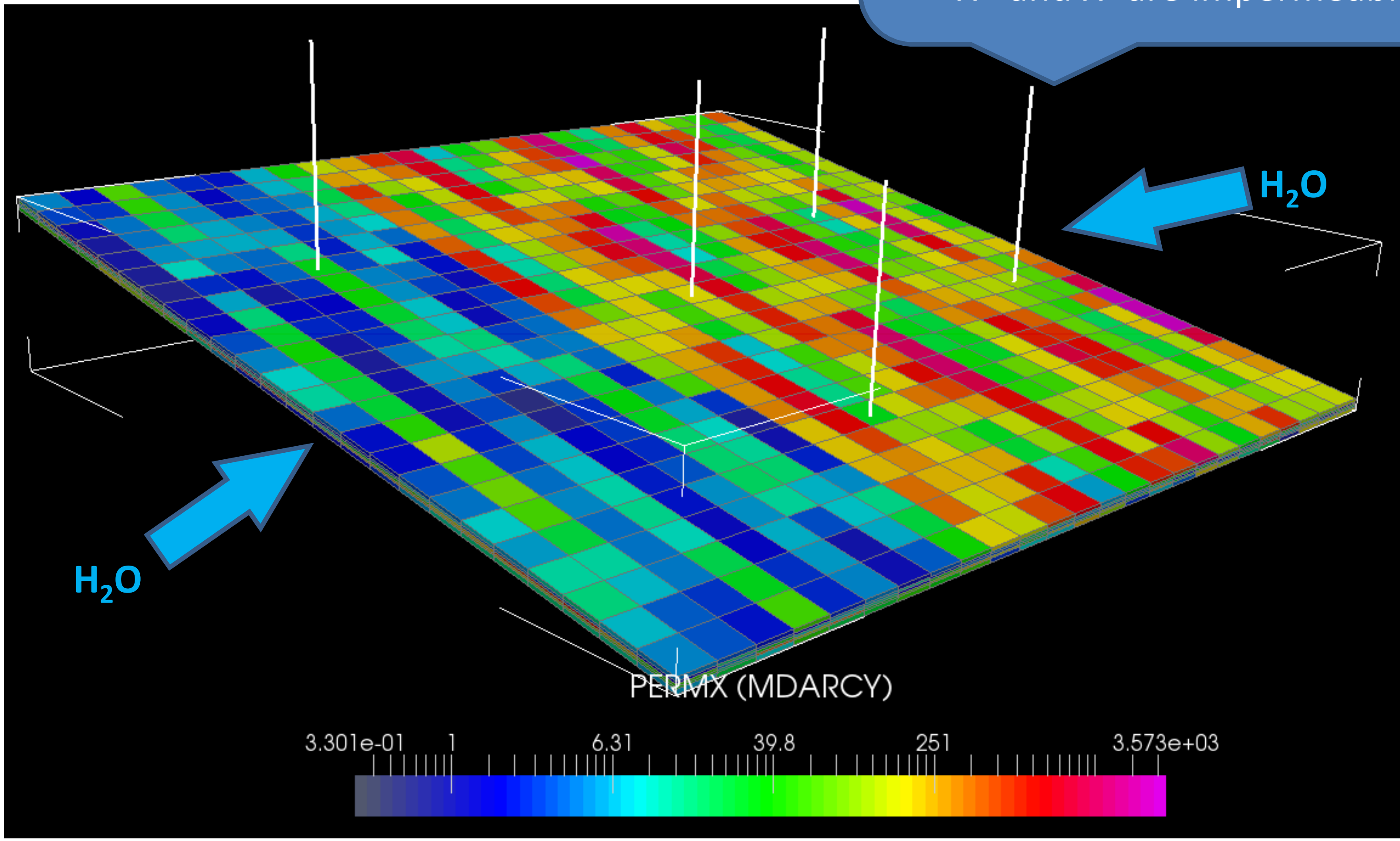
1. **Open RUN-file in text editor**
2. **Run the simulation**
3. **Open results in ParaView**

Scenario 11 (result)



Scenario 11 (exercise)

Exercise: Re-simulate scenario 11 supposing that the fluid can flow into the reservoir only through Y+ and Y- lateral boundaries, whereas X+ and X- are impermeable.

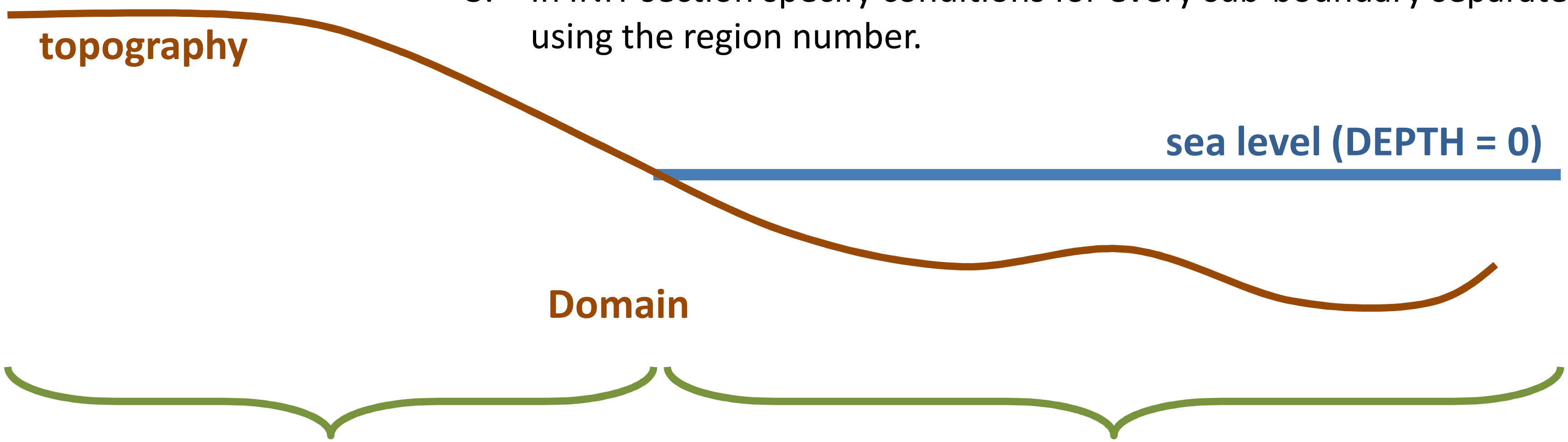


Onshore/offshore boundary conditions

Onshore/offshore conditions

To model the boundary conditions:

1. Create the boundary grid blocks for the top surface of the domain using the BOUNDARY keyword;
2. Divide the boundary into two sub-boundaries (onshore and offshore) using the DEPTH property. Use a region number (e.g., INCONUM) to distinguish the sub-boundaries.
3. In INIT section specify conditions for every sub-boundary separately using the region number.



At the top surface of the domain: air (we use CO_2 instead), P=1 Bar, T=20°C

At the top surface of the domain: water, P = hydrostatic in water, T=20°C (or other value).

Simulate using 3 (or 2) cores

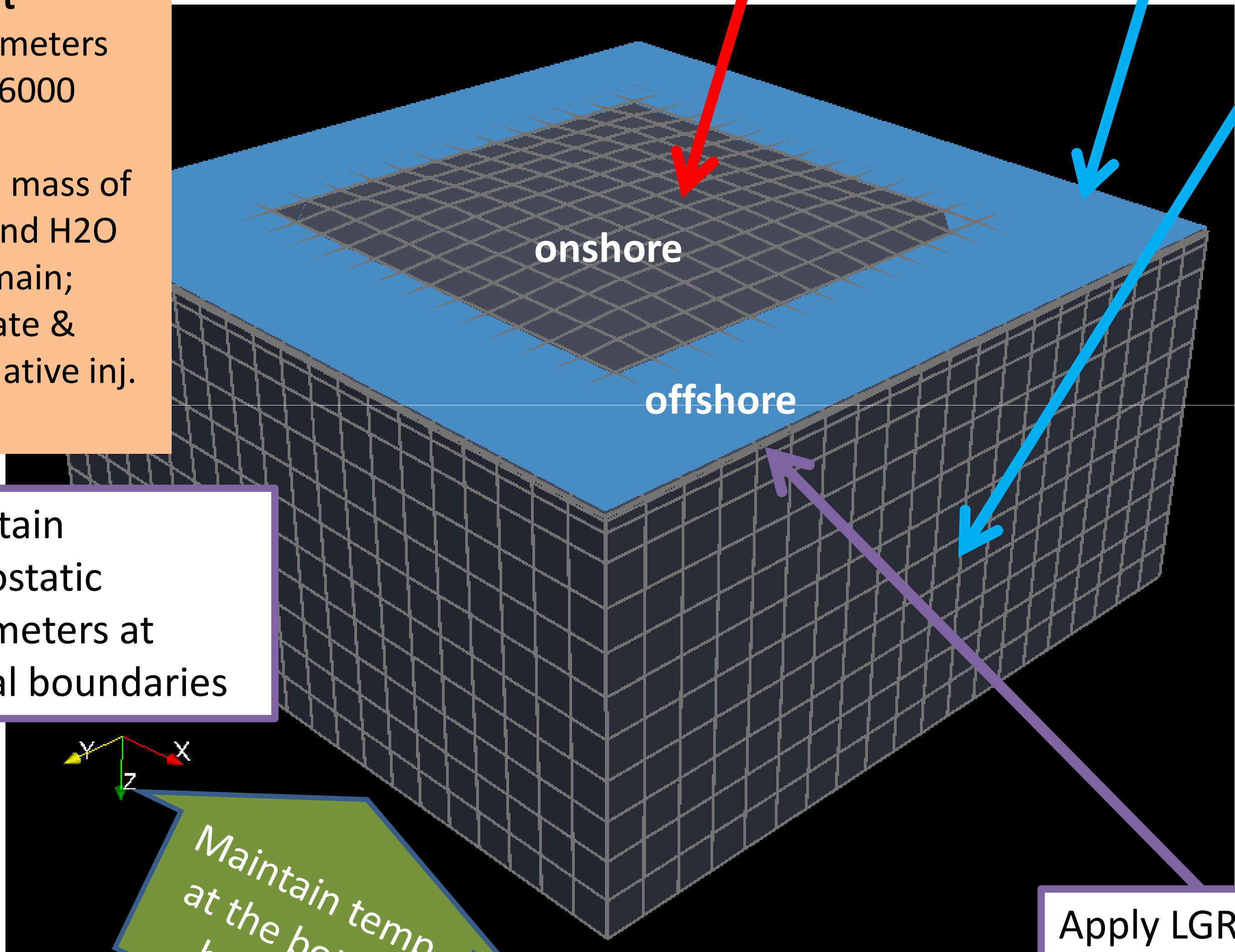
Scenario 12

“Square” island ☺
P=0.1 Mpa, T=20 C
Pure CO_2

Pressure hydrostatic
Geoth. Grad 2.5 C/100m
Pure water

Report

- parameters up to 6000 days;
- total mass of CO2 and H2O in domain;
- Inj. rate & cumulative inj. rate.



Grid file is provided.
Porosity =0.2
X-Perm=20 mD
Z-Perm=10 mD
RockDen=2900 kg/m3
RockHC =1 kJ/kg/K
RockHCcf=2W/m/K
Rel.Perm see in RUN.

Model influx of hot magmatic fluid in block (13,12,10).
Fluid T=400C at P=10MPa,
CO2 mol. frac.=50%.
Rate= 500 t/day.

Maintain hydrostatic parameters at lateral boundaries

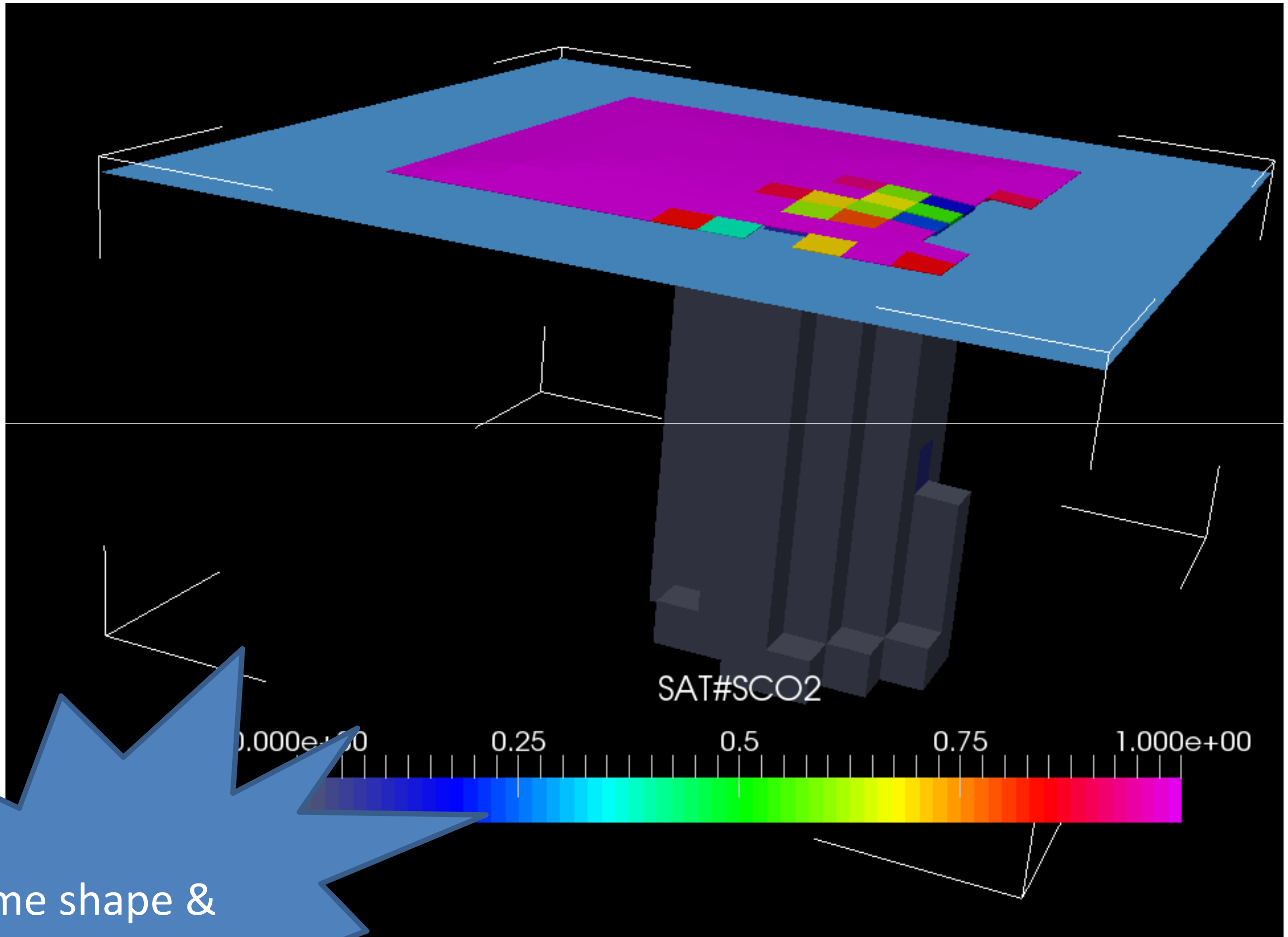
Maintain temp. at the bottom boundary

Apply LGR near surface

RUN-file (scenario 12)

1. Open RUN-file in text editor
2. Run the simulation
3. Open results in ParaView

Results (scenario 12)



Plume shape &
CO_2 saturation

Thank you!