MUFITS
Training Course

Day 1
Fundamentals

Program

- Introduction (mathematical model, EOS-modules)
- RUN-file (input data preparation)
- Scenario1 (the first example)
- Fundamentals (Loading arrays, initial conditions, report times, etc.)
- Heat conduction (Scenario 2)

Introduction

Mathematical model

Equations for multicomponent multiphase flows in porous medium:

$$\frac{\partial}{\partial t}\left(\phi\sum_{i=1}^{p}\rho_i c_{i(j)} s_i\right)+\mathrm{div}\left(\sum_{i=1}^{p}\rho_i c_{i(j)}\mathbf{w}_i+\boldsymbol{\psi}_{(j)}\right)=q_{(j)},\quad j=1,\ldots,c$$

- mass balance equations

$$\frac{\partial}{\partial t}\left(\phi\sum_{i=1}^{p}\rho_i e_i s_i+(1-\phi)\rho_r e_r\right)+\mathrm{div}\left(\sum_{i=1}^{p}\rho_i h_i\mathbf{w}_i+\boldsymbol{\psi}_{(j)}\right)=q_{(e)}$$

- energy balance equation

$$\mathbf{w}_i=-K\frac{k_{r,i}}{\mu_i}\left(\mathbf{grad}P_i-\rho_i\mathbf{g}\right)$$

- Darcy correlation

$$k_{r,i}=k_{r,i}(s),\quad P_i-P_k=P_{c,ik}(s)$$

- saturation functions (rel. perm. & cap. pres.)

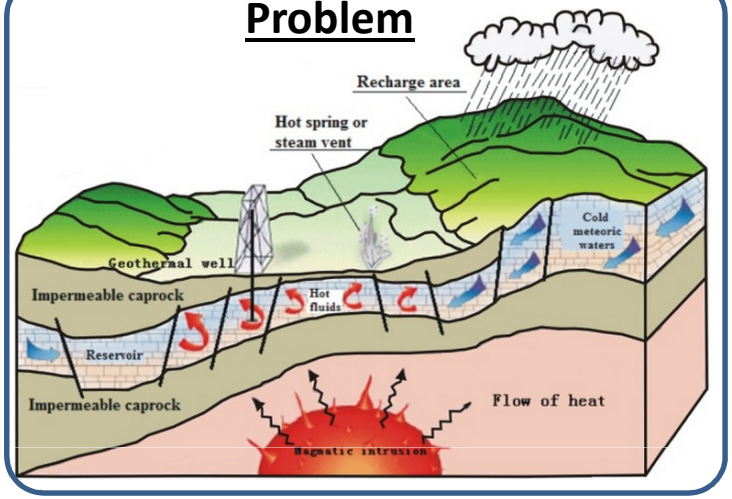
$$\sum_{i=1}^{p}s_i=1,\qquad\sum_{j=1}^{c}c_{i(j)}=1$$

- consistency relations

+ Equations of state (for fluid and rocks)

Data flowchart

Problem



RUN-file

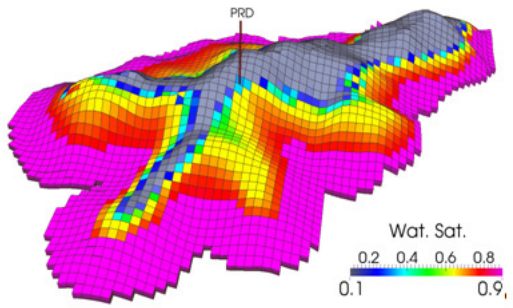
- Grid;
- Porosity and permeability;
- Initial & boundary conditions;
- Rel. permeability;
- Wells and point sources;
- Report time;
- etc.

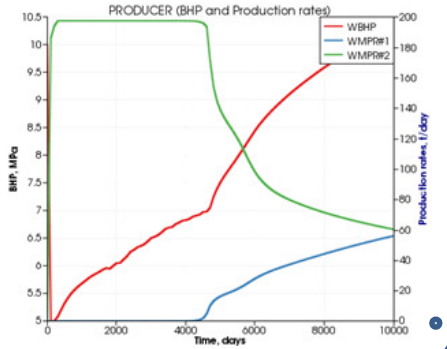
The file is prepared in a Text Editor

Runs via MPI interface

Simulator

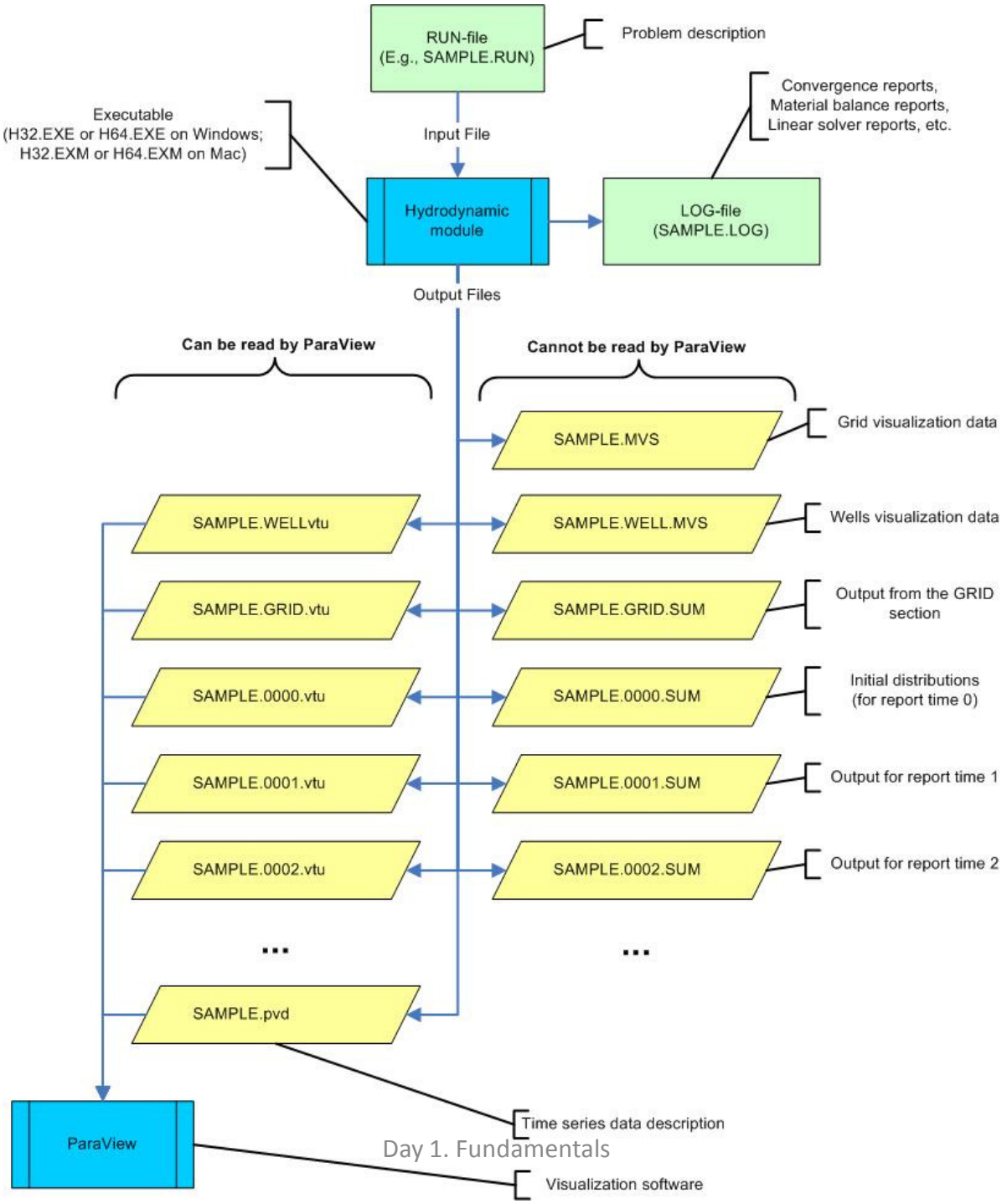
Results



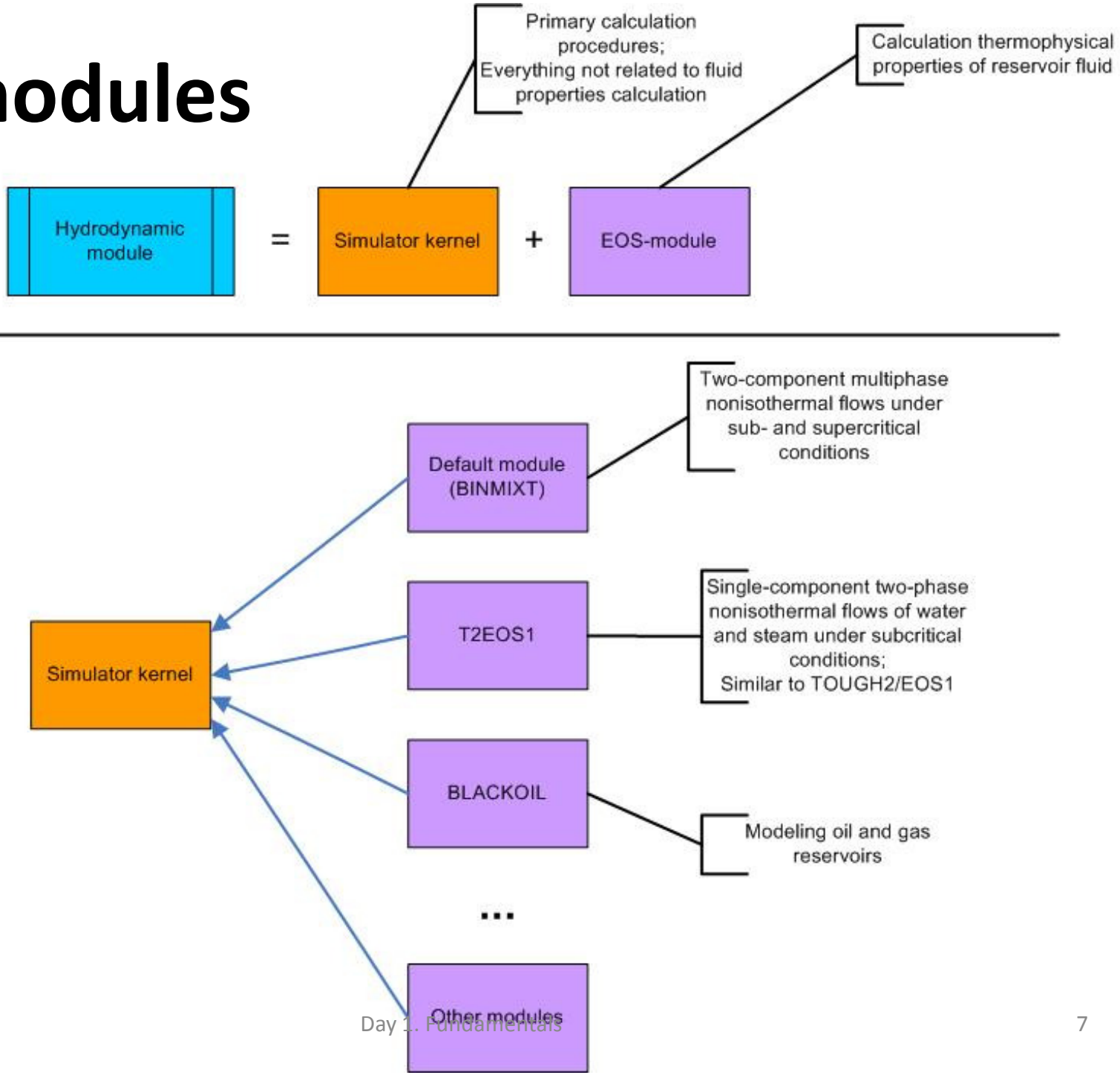


In ParaView

Data flowchart



EOS modules


Primary calculation procedures; Everything not related to fluid properties calculation
Calculation thermophysical properties of reservoir fluid
Hydrodynamic module
=
Simulator kernel
+
EOS-module
Two-component multiphase nonisothermal flows under sub- and supercritical conditions
Default module (BINMIXT)
Simulator kernel
T2EOS1
Single-component two-phase nonisothermal flows of water and steam under subcritical conditions; Similar to TOUGH2/EOS1
BLACKOIL
Modeling oil and gas reservoirs
...
Other modules

EOS module T2EOS1

Single-component (H2O) single-phase and two-phase flows of after and vapor under subcritical conditions. This module is similar to TOUGH2/EOS1.

$$\frac{\partial}{\partial t}\left(\phi(\rho_w s_w + \rho_v s_v)\right) + \mathrm{div}\left(\rho_w w_w + \rho_v w_v\right) = q$$

$$\frac{\partial}{\partial t}\left(\phi(\rho_w e_w s_w + \rho_v e_v s_v) + (1-\phi)\rho_r e_r\right) + \mathrm{div}\left(\rho_w h_w \mathbf{w}_w + \rho_v h_v \mathbf{w}_v - \overline{\lambda}\mathbf{grad}T\right) = q_{(e)}$$

$$\mathbf{w}_i = -K\frac{k_{r,i}}{\mu_i}\left(\mathbf{grad}P - \rho_i \mathbf{g}\right), \quad i = w, v$$

$$s_w + s_v = 1, \qquad \overline{\lambda} = \phi(s_w\lambda_w + s_v\lambda_v) + (1-\phi)\lambda_r\,, \qquad k_{r,i} = k_{r,i}(s_w)$$

$$\rho_i = \rho_i(P,T), \;\; e_i = e_i(P,T), \;\; h_i = h_i(P,T), \;\; \mu_i = \mu_i(P,T), \;\; \lambda_i = \lambda_i(P,T), \quad i = w, v$$

$$if\,(T < T_{eq}(P)) \quad s_w = 1$$

$$if\,(T > T_{eq}(P)) \quad s_v = 1$$

$$if\,(0 \le s_w \le 1)$$

EOS module T2EOS1

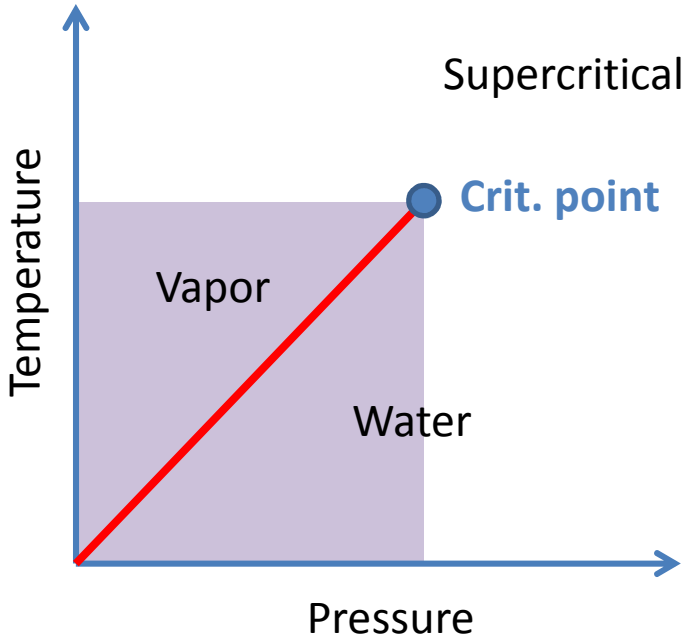
Single-component (H2O) single-phase and two-phase flows of after and vapor under subcritical conditions. This module is similar to TOUGH2/EOS1.

Closing relations:

$$if\,(T < T_{eq}(P)) \quad s_w = 1 \quad - single-phase\ water$$

$$if\,(T > T_{eq}(P)) \quad s_v = 1 \quad - single-phase\ vapor$$

$$if\,(0 \le s_w \le 1) \quad T = T_{eq}(P) - two-phase$$



RUN-file
(or data-file)

RUN-file (scenario 1)



Keywords

The keywords are the instructions in the RUN-file to the simulator. By using keywords the grid, the physical model,
the relative permeabilities, the initial and boundary conditions, etc. are specified. The keywords

- must start in column 1 of RUN-file;
- must be typed in uppercase letters;
- consist of up to 8 uppercase letters.

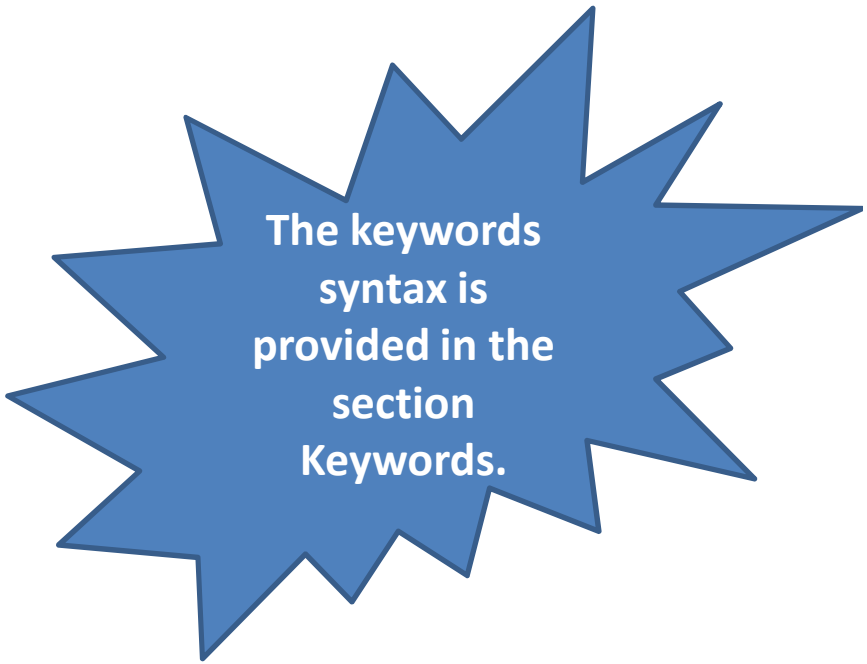
Example of keywords

```
MAKE
      CART    20  1    5    /

XYZBOUND
    0    200     -5     5    1000 1050  /

ENDMAKE

PORO
   100*0.25 /
```



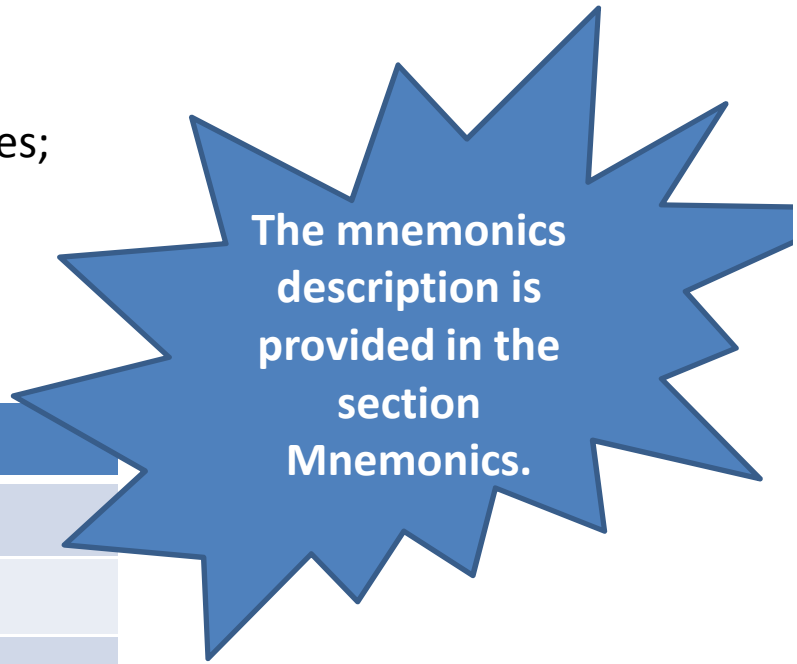
Mnemonics

Mnemonics are a short character references to physical, geological, geometrical, logical data, etc. Using mnemonics you can

- load data in simulation;
- perform operations on arrays;
- specify regions for thermophysical properties;
- specify initial or boundary conditions;
- specify output data.

Examples of Mnemonics

Mnemonic	Parameter
PRES	Pressure (MPa)
TEMP	Temperature (K)
TEMPC	Temperature (C)
SWAT	Water saturation
SVAP	Vapor saturation



Comments in RUN-file

Commented out lines begin with: '--', '!', ' ', lowercase letters.

Examples of commented out lines

```
GRID      ##########################################################################

-- this is a comment line
!  this is a comment line too
   this is another comment line

MAKE      <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
-- cartesian                  These are comment
--    grid    nx  ny  nz        lines
      CART    10  10  1  /

XYZBOUND
!   xmin xmax  ymin ymax  zmin zmax      Also a comment line
    100   200  100   200   0    10    /

...

ENDMAKE   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
```

Repeated counts

A data item can be repeated in a keyword a number of times by using asterisk symbol.

Example of repeated counts

```
BOX
-- imin-imax  jmin-jmax  kmin-kmax
    1    4     1    3     1    1   /

-- This keyword

PORO
   0.25  0.25  0.25  0.25
   0.25  0.25  0.25  0.25
   0.15  0.15  0.15  0.15 /

-- is equivalent to any of the following keywords

PORO
   4*0.25
   4*0.25
   4*0.15 /

PORO
   8*0.25 4*0.15 /

PORO
   7*0.25 0.25 2*0.15 2*0.15 /
```

Default values

There are two options for defaulted parameters: by using asterisk or by the slash symbol.

Example of defaulted parameters

```
BOX
-- imin-imax  jmin-jmax  kmin-kmax
     1   5      1    6    1    2   /

-- keywords

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

-- The keyword

BOX
-- imin-imax  jmin-jmax  kmin-kmax
      2*        3    1*    2  /

-- is equivalent to

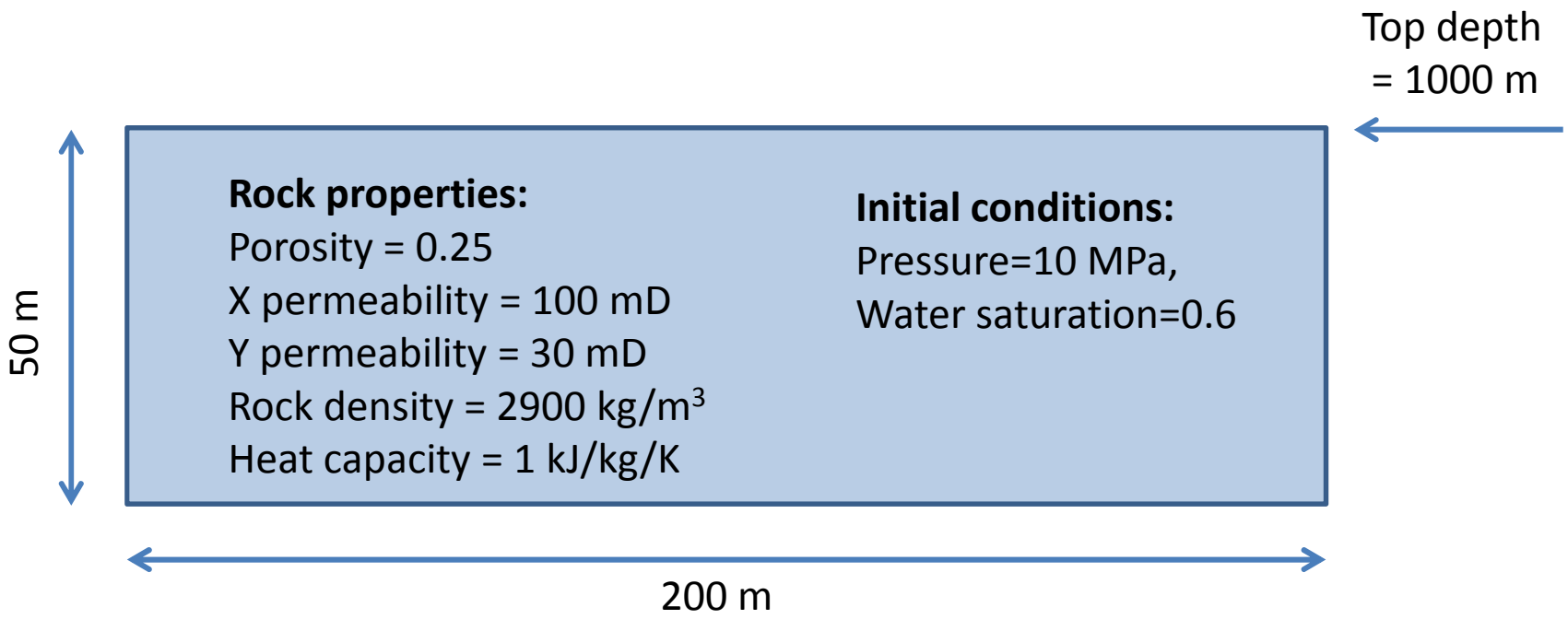
BOX
-- imin-imax  jmin-jmax  kmin-kmax
    1    5     3    6     2    2  /

-- because the parameters imin,imax,jmax,kmax are not altered
```

Scenario 1

Formulation of the problem

A simple gravity segregation 2D problem:

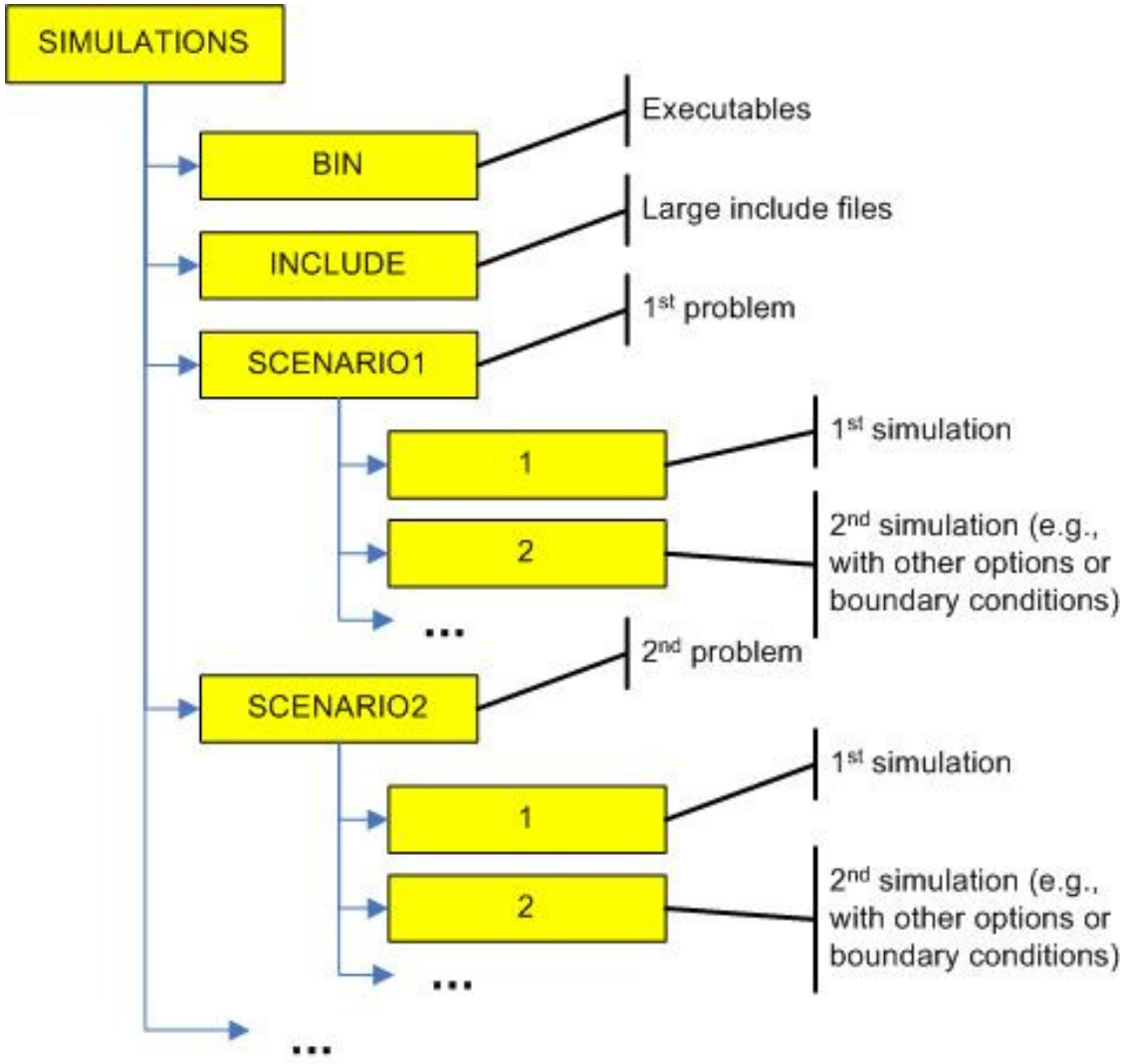


All boundaries of the domain are impermeable and insulated

RUN-file (scenario 1)



Folders hierarchy



Running the problem

In the folder SIMULATIONS/SCENARIO1/0/ execute

Mac:

```
mpirun –n 1 ./../../BIN/H64.EXM SCENARIO1.RUN > SCENARIO1.LOG
```

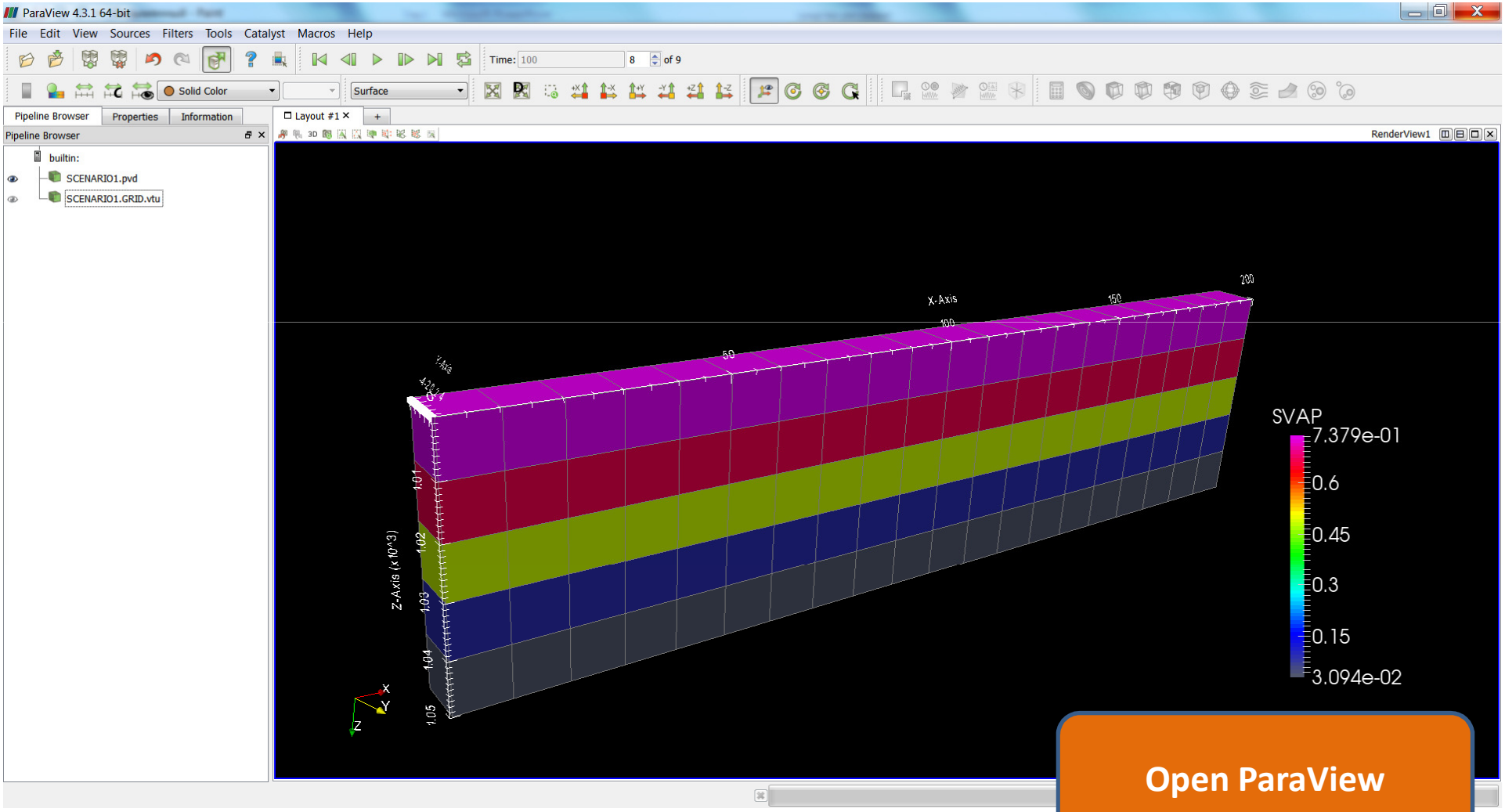
Linux:

```
mpirun –n 1 ./../../BIN/H64.EXL SCENARIO1.RUN > SCENARIO1.LOG
```

Windows:

```
"…mpiexec.exe" –n 1 ../../BIN/H64.EXE SCENARIO1.RUN > SCENARIO1.LOG
```

Results in ParaView



Fundamentals

Control on output data

The output in the file SCENARIO1.GRID.SUM is controlled by the RPTGRID keyword

RPTGRID syntax

```
1  -- in GRID section
2
3  RPTGRID
4     mnemonic1   mnemonic2   mnemonic3   ...   /
5
6  ================================================================================
7
8        mnemonic# - is the mnemonic of a property saved in the file *.GRID.SUM.
9                    If one of the mnemonics is ASCII then the formatted file
10                   is saved.
```

Exercise: Add output of grid blocks coordinates

The output in the file SCENARIO1.####.SUM is controlled by the RPTSUM keyword

RPTSUM syntax

```
1  -- in INIT or SCHEDULE section
2
3  RPTSUM
4     mnemonic1   mnemonic2   mnemonic3   ...   /
5
6  ================================================================================
7
8        mnemonic# - is the mnemonic of a property saved in the files *.0000.SUM,
9                    *.0001.SUM, *.0000.SUM, etc. If one of the mnemonics is ASCII
10                   then the formatted file is saved.
```

Exercise : Add output of water and vapor densities

Cartesian grids

The number of grid blocks along every axis is defined by the keyword **MAKE**

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
```

Exercise : Re-simulate the Scenario 1 by using grid 25*10 blocks

Cartesian grids

The domain boundaries are defined by the keyword **XYZBOUND**

XYZBOUND syntax

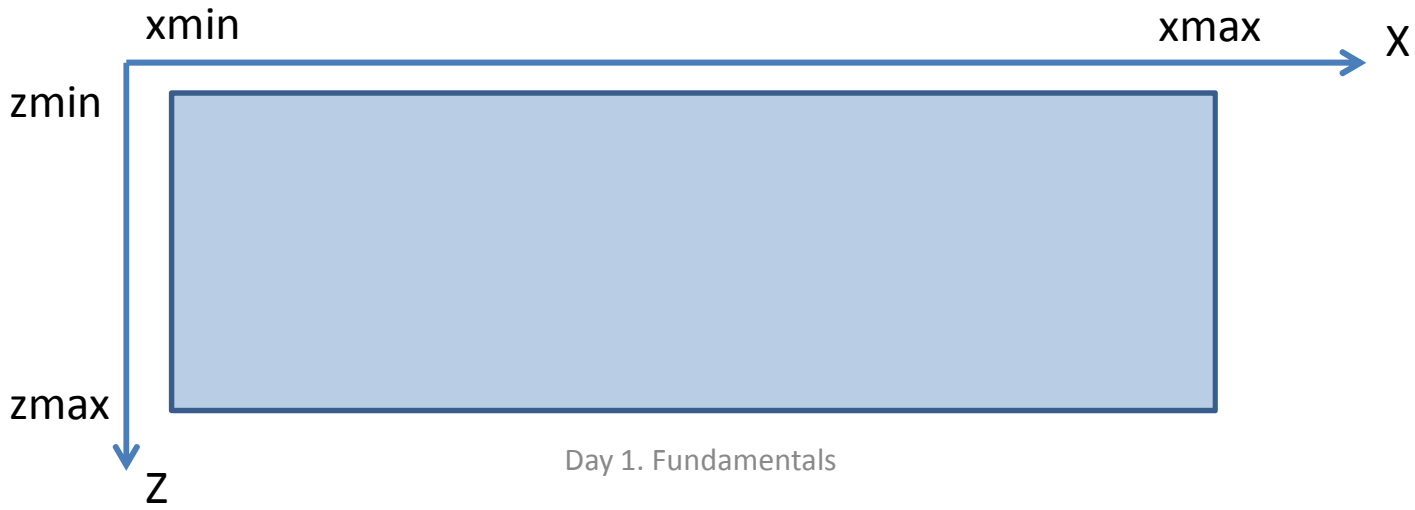
```
-- within MAKE/ENDMAKE brackets.

XYZBOUND
   xmin xmax  ymin ymax  zmin zmax   xincr  yincr  zincr /

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

     xmin/xmax - the domain boundaries along axis x (xmin<xmax)
     ymin/ymax - the domain boundaries along axis y (ymin<ymax)
     zmin/zmax - the domain boundaries along axis z (zmin<zmax)
```

Exercise : Locate the top boundary of the reservoir at depth 900 meters, whereas the bottom boundary at depth 950 meters.



Cartesian grids

The grid block extensions can be redefined using the **DXV**, **DYV**, **DZV** keywords

DXV syntax

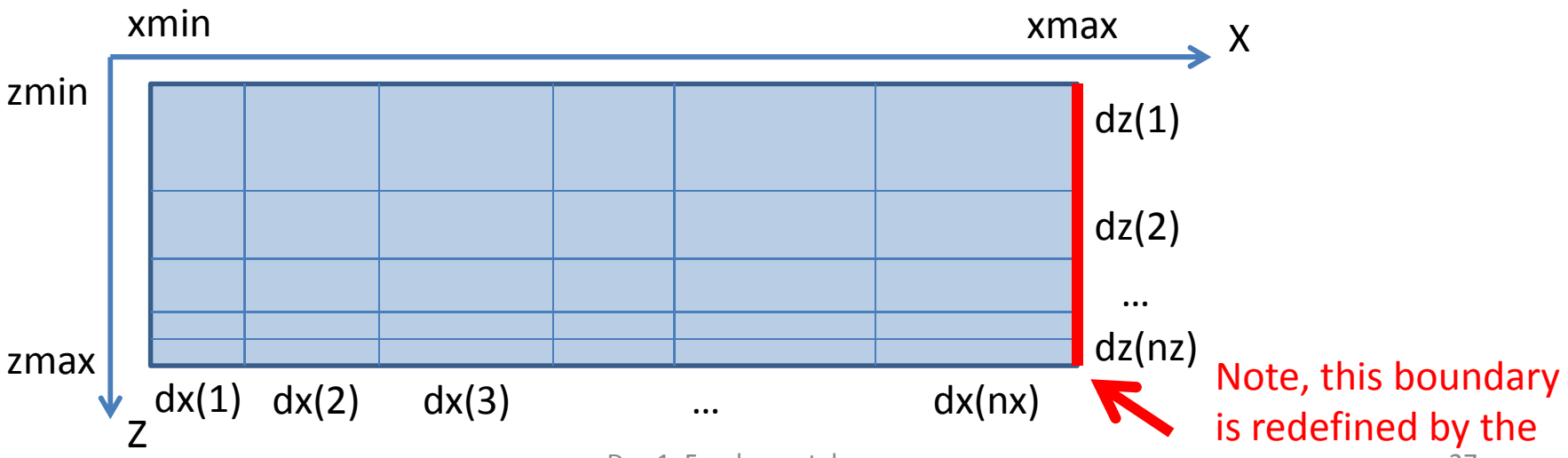
```
-- within MAKE-ENDMAKE brackets

DXV
  dx(1) dx(2) dx(3) ... dx(nx) /

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

   dx(#) - grid blocks extensions along axis X.
   nx    - number of grid block along axis X. nx is the 2nd argument of the
           keyword MAKE.
```

Exercise : Introduce a refined grid at the bottom boundary and re-simulate the Scenario 1



Note, this boundary is redefined by the keyword DXV

Arrays loading

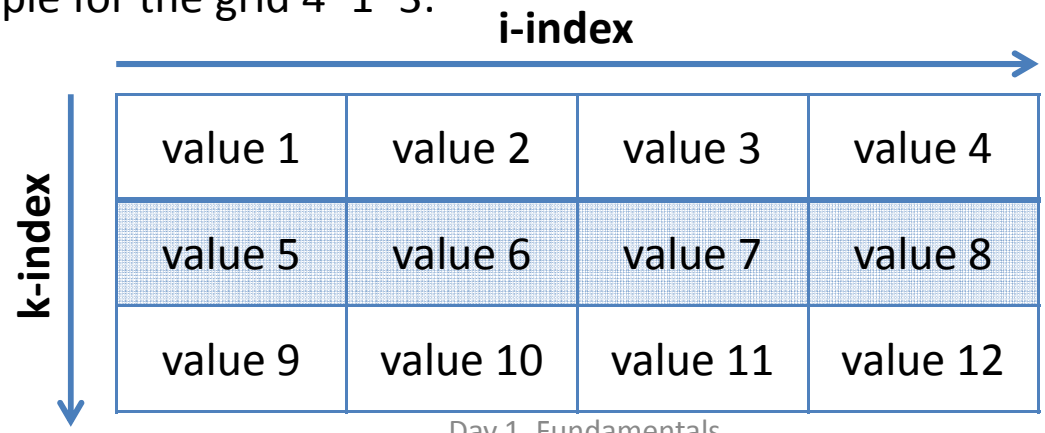
PORO syntax

```
-- in GRID section
Mnemonic  of  Array to be loaded (e.g., PERMX, PERMY, PRES)
PORO
   value1  value2  value3  ... value(nbox) /

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

    value(#) - value assigned to the corresponding grid block in the input box.
               The grid blocks are ordered with i index cycling fastest,
               followed by j and k indexes.
    nbox     - number of grid blocks in the current input box.
               nbox=(imax-imin+1)*(jmax-jmin+1)*(kmax-kmin+1), where
               imin, imax, jmin ,jmax, kmin, kmax are defined by the keyword
               BOX.
```

An example for the grid 4*1*3:

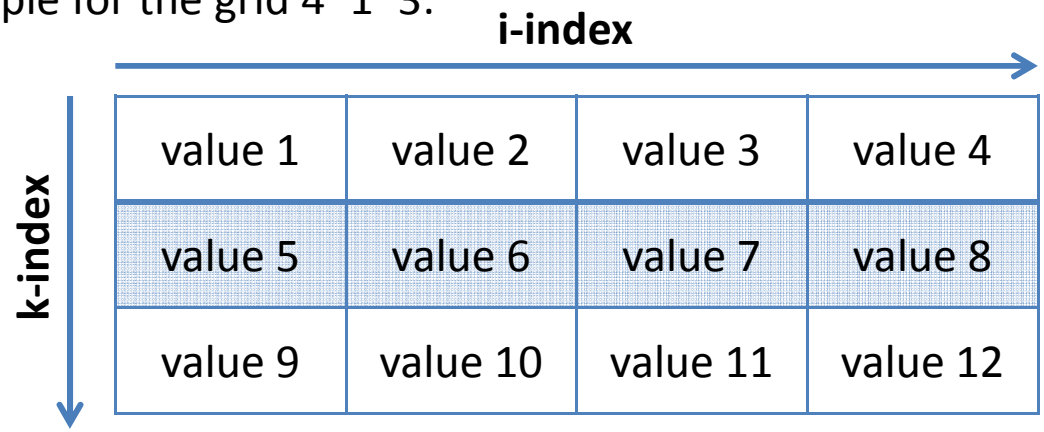


Note:

i-index – X axis
j-index – Y axis
k-index – Z axis

Arrays loading

An example for the grid 4*1*3:



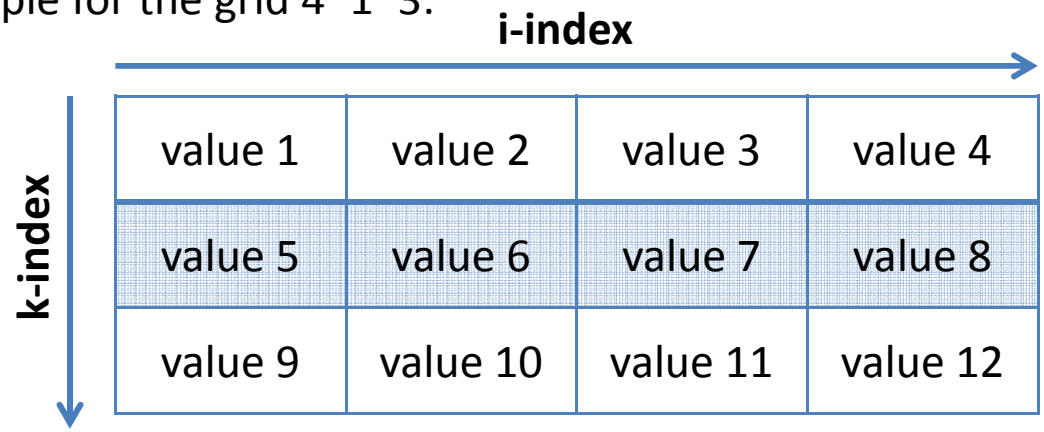
Note:

i-index – X axis
j-index – Y axis
k-index – Z axis

Exercise : Increase fivefold the permeability of the layers 3 and 4 and re-simulate scenario 1

Arrays loading

An example for the grid 4*1*3:



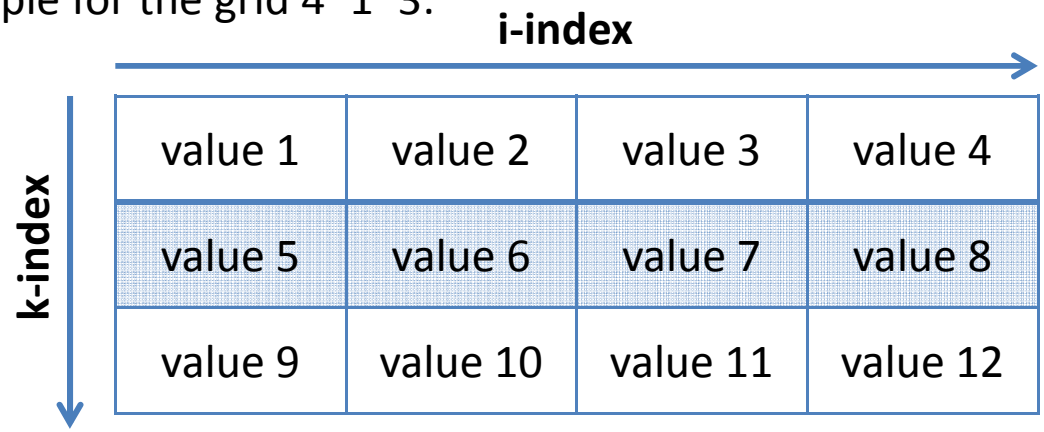
Note:

i-index – X axis
j-index – Y axis
k-index – Z axis

Exercise : Increase fivefold the permeability in the columns i=8-13 and re-simulate the scenario 1

Arrays loading

An example for the grid 4*1*3:



Note:

i-index – X axis
j-index – Y axis
k-index – Z axis

Exercise : In the left half of the domain the initial pressure is 11 MPa and the water saturation is 0.35. In the right half of the domain the initial pressure is 10 Mpa and the water saturation is 0.7. Impose these initial conditions and re-simulate scenario 1.

Keyword BOX

BOX syntax

```
-- in all sections except RUNSPEC and POST

BOX
   imin imax  jmin jmax  kmin kmax /

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

   imin/imax - the boundaries of the input box along i-indexation axis.
               By default imin=1 and imax=ni, where ni is the 2nd argument
               of the keyword MAKE.
   jmin/jmax - the boundaries of the input box along j-indexation axis.
               By default jmin=1 and jmax=nj, where nj is the 3rd argument
               of the keyword MAKE.
   kmin/kmax - the boundaries of the input box along k-indexation axis.
               By default kmin=1 and kmax=nk, where nk is the 4th argument
               of the keyword MAKE.
```

The **BOX** keyword allows to select a region of the reservoir for arrays input. The arrays in the selected region are loaded on a block by block basis, as this region is the whole domain.

The **ENDBOX** keyword resets the input box so that it encompasses the whole domain.

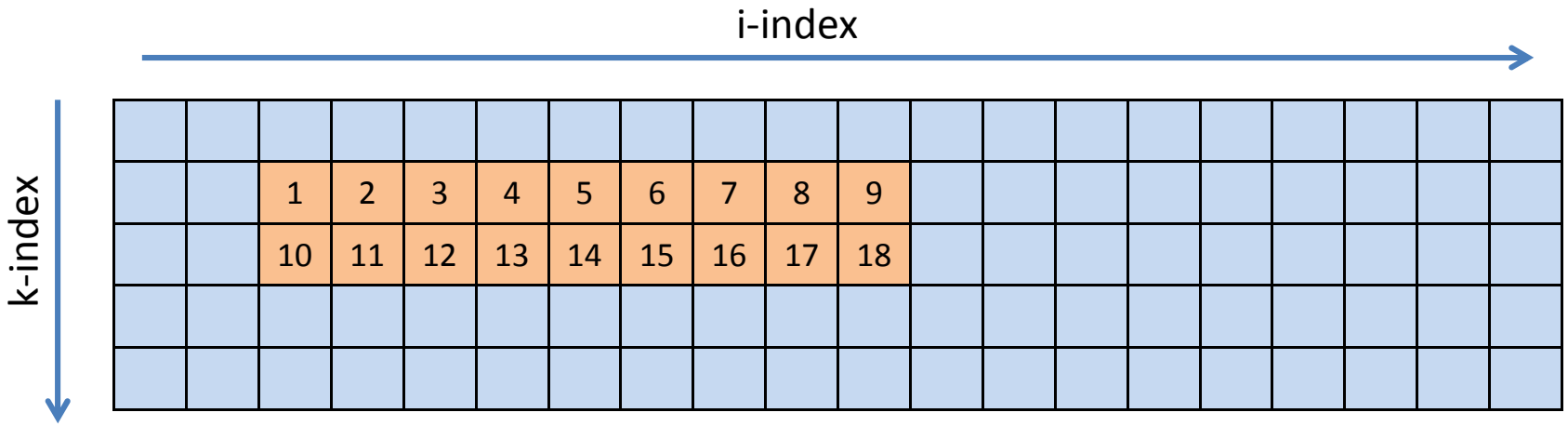
Note that at the beginning of every section the input box is reset by the program.

Keyword BOX

For example, the keyword

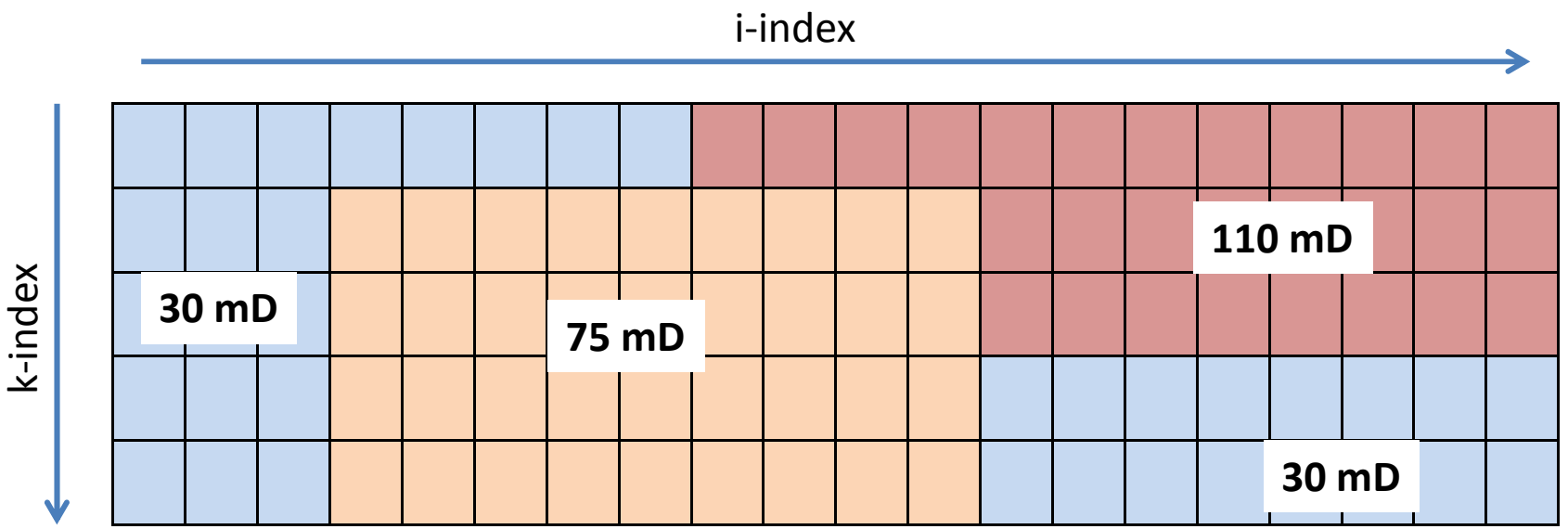
```
BOX
  3  11    2*    2  3 /
```

selects the following region for arrays input



Keyword BOX (Task)

Exercise : Create the following initial distribution of Z direction permeability and re-simulate the Scenario 1.



Rock properties

The rock thermophysical properties must be specified within brackets **ROCK-ENDROCK.**
The rock density and heat capacity can be defined using the **ROCKDH** keyword:

ROCKDH syntax

```
-- within ROCK-ENDROCK brackets

ROCKDH
   dens  heatcap /

/
------------------------------------------------------------------------------

      dens    - rock density
      heatcap - rock heat capacity
```

Relative permeabilities

The saturation functions (e.g., the relative permeabilities and capillary pressure) must be specified within brackets **SAT-ENDSAT**. They can be defined using the **SATTAB** keyword.

SATTAB syntax

```
-- within SAT-ENDSAT brackets

SATTAB
  sliq1  krliq1  krgas1  pcap1 /
  sliq2  krliq2  krgas2  pcap2 /
  sliq3  krliq3  krgas3  pcap3 /
...
/

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

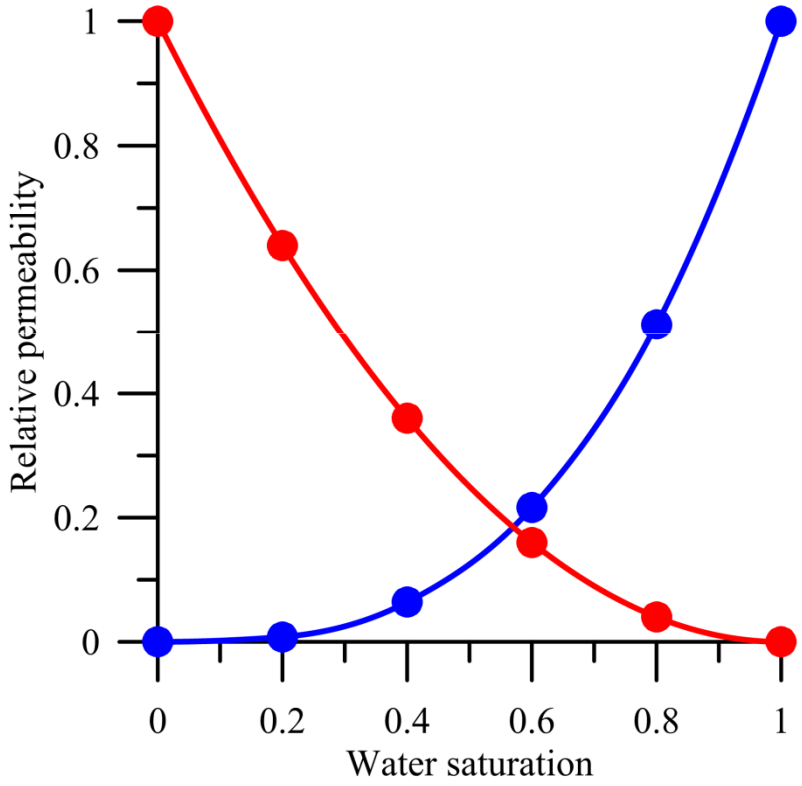
      sliq#  - liquid phase saturation
      krliq# - liquid phase relative permeability
      krgas# - gas phase relative permeability
      pcap#  - capillary pressure
```

Relative permeabilities (example)

```
SATTAB
-- sliq     krliq    krgas
    0.0      0.0      1.0    /
    0.2      0.008    0.64   /
    0.4      0.064    0.36   /
    0.6      0.216    0.16   /
    0.8      0.512    0.04   /
    1.0      1.0      0.0    /
/
```

$$k_{r,wat}(s_{wat}) = s_{wat}^3$$

$$k_{r,vap}(s_{wat}) = (1 - s_{wat})^2$$



Relative permeabilities in ParaView

The relative permeabilities can be loaded in ParaView by the **RPTSATTA** keyword. This keyword invokes the output of CSV file with a table for relative permeabilties (and capillary pressure).

RPTSATTA syntax

```
-- in PROPS section

RPTSATTA
  satnum  filename  n  smin  smax /

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

      satnum    - SATNUM region ID for which the output is requested;
      filename  - the file name in which saturation functions are saved. This
                  file should have extension .csv;
      n         - number of output points in the interval [smin,smax];
      smin-smax - the boundaries of the output interval.
```

Exercise : Load and create a plot for the relative permeability curves in ParaView

Relative permeabilities (exercise)

Exercise : Re-simulate scenario 1 by using Corey relative permeabilities
$S_{l,min}$=0.2
$S_{g,min}$=0.05
n_l=2.5
n_g=1.5.
Build the relative permeability plots in ParaView

Corey curves:

$$k_{r,liq} = \left(\frac{s_{liq} - s_{liq,\min}}{1 - s_{liq,\min} - s_{gas,\min}} \right)^{n_{liq}}$$

$$k_{r,gas} = \left(\frac{1 - s_{liq} - s_{gas,\min}}{1 - s_{liq,\min} - s_{gas,\min}} \right)^{n_{gas}}$$

Relative permeabilities (exercise)

The answer is:

```
SATTAB
    0.2    0.0       1.0     /
    0.4    0.0367    0.6279  /
    0.6    0.2077    0.3187  /
    0.8    0.5724    0.0894  /
    0.95   1.0       0.0     /
/
```

Note, that the critical saturations must be defined.

Initial conditions

The initial conditions for every cell must be specified in the **INIT** section

There are 5 options to impose the initial conditions when using T2EOS1 module:

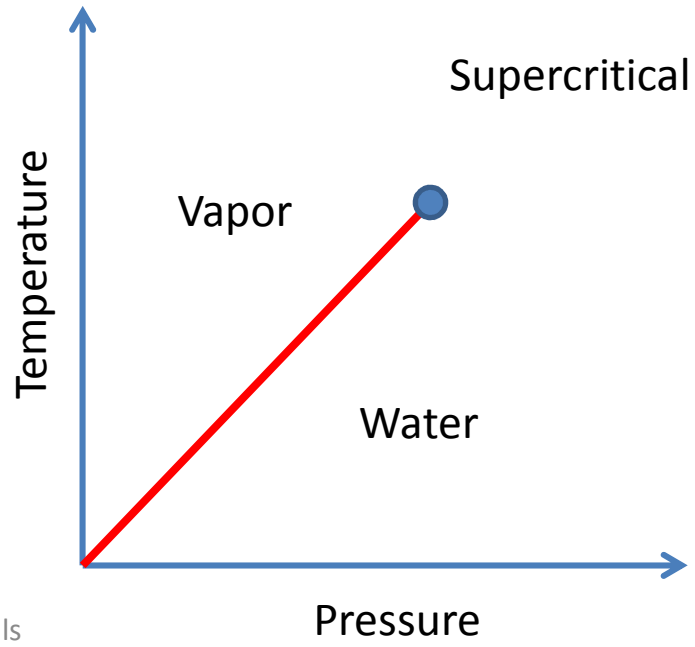
1) Specify pressure (PRES) and water saturation (SWAT) [priority 1]
2) Specify pressure (PRES) and vapor saturation (SVAP) [priority 2]
3) Specify temperature (TEMP) and water saturation (SWAT) [priority 3]
4) Specify temperature (TEMP) and vapor saturation (SVAP) [priority 4]
5) Specify pressure (PRES) and temperature (TEMP) [priority 5]

2-phase

1-phase

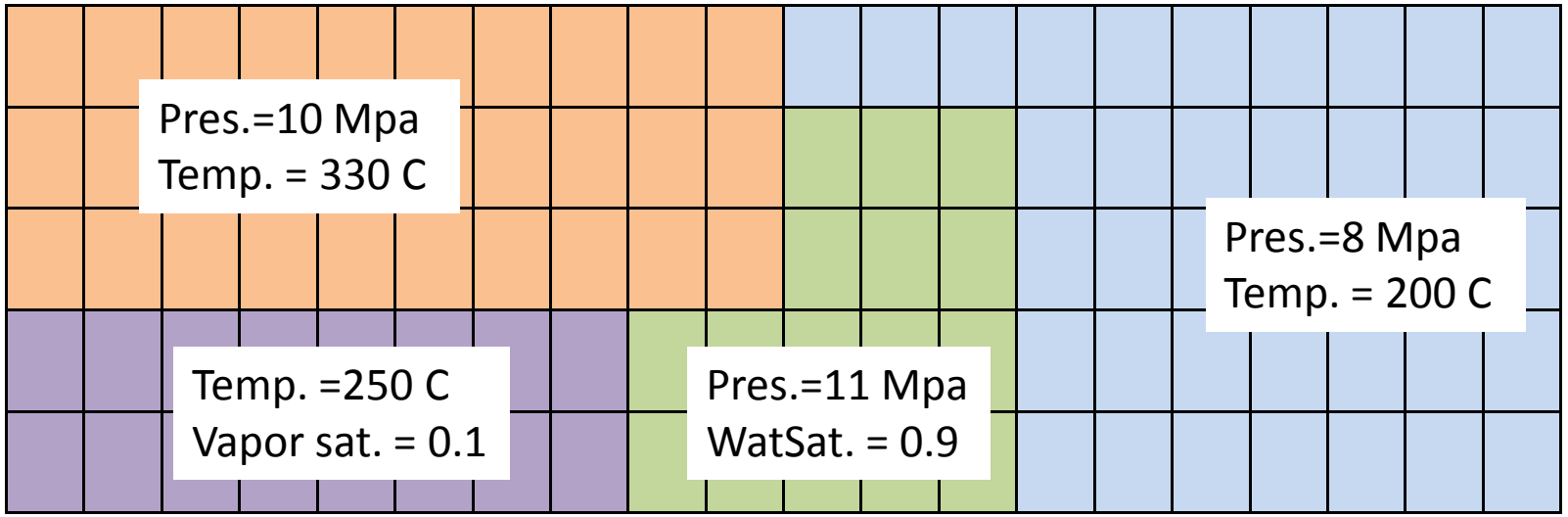
Note, that the TEMPC mnemonic can be used to specify the temperature in degrees of Celsius instead of the TEMP mnemonic which is for degrees of Kelvin.

Note, that if parameters in a cell remain unspecified the simulation will be terminated after the INIT section.



Initial conditions (Exercise)

Exercise : Specify the following initial conditions and re-simulate Scenario 1



Report times

Note that time dimension is Days.

The report times are the times at which the output is saved. These can be specified in **SCHEDULE** section by the **TSTEP** keyword. When the simulator encounters this keyword it proceeds the simulation further in time and produces the required outputs.

TSTEP syntax

```
-- in SCHEDULE section

TSTEP
   tstep1 tstep2 tstep3 ... /

/
================================================================================

     tstep# - steps between report times (in days). Every tstep# initiates the
              program to advance the simulation to time->time+tstep#. After
              every time->time+tstep# a new *.####.SUM file is saved.
```

Exercise : Re-simulate Scenario 1 reporting parameters distributions at the times: 10, 15, 30, 70, 100, 150, 500 days

Report times

TIMES syntax

```
-- in SCHEDULE section

TIMES
   time1 time2 time3 ... /

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

     time# - report times (in days). Every time# initiates the
             program to advance the simulation to this time and save a new
             *.####.SUM file.
```

Exercise Task: Re-simulate Scenario 1 reporting parameters distributions at the times: 10, 15, 30, 70, 100, 150, 500 days and using the TIMES keyword

Exercise : Re-simulate Scenario 1 reporting parameters distributions at the times: 10, 15, 30, 70, 100, 150, 500 days and using both TIMES and TSTEP keywords in the same simulation.

Heat conduction

Heat conduction option

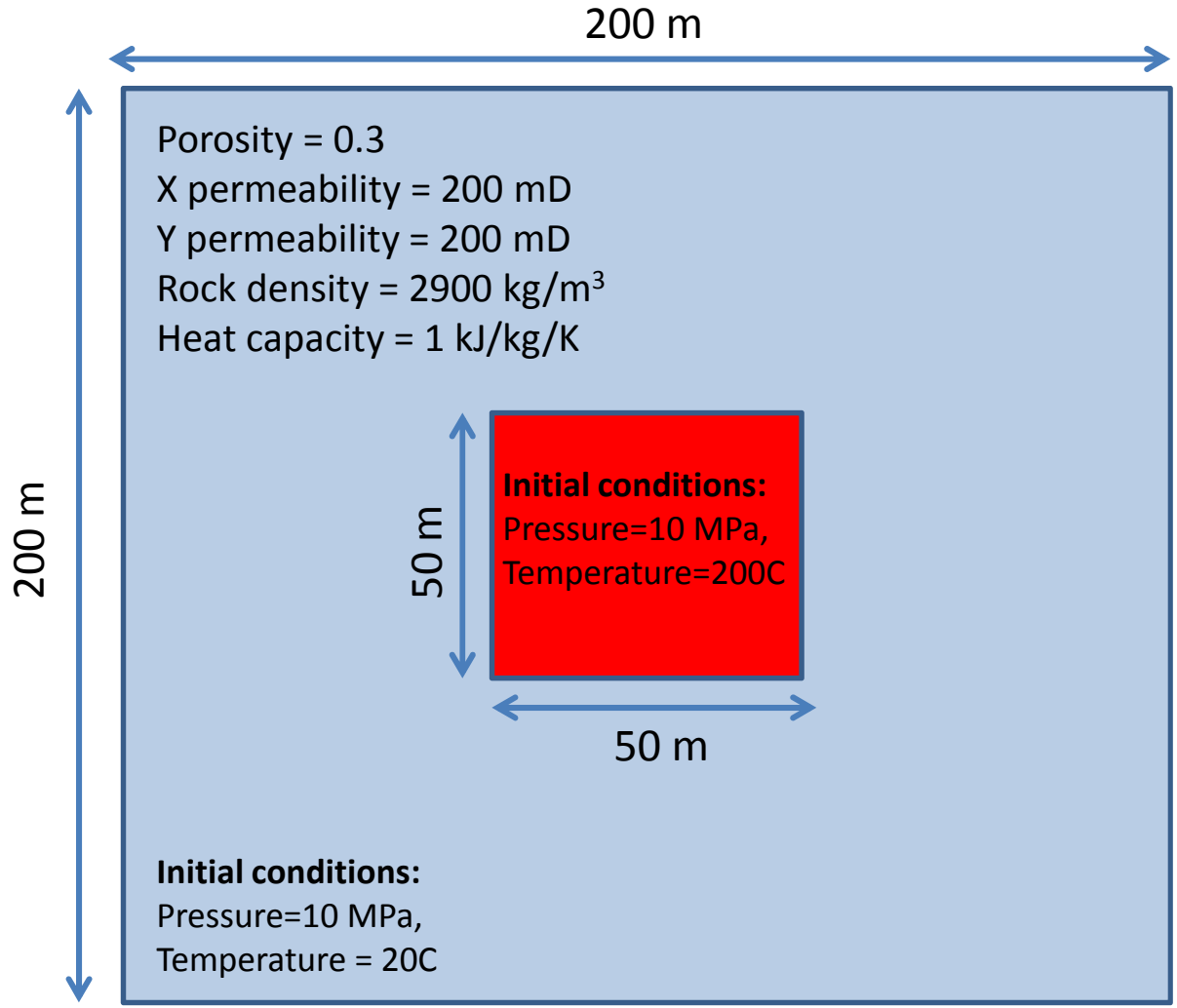
Heat conduction coefficient:

$$\lambda = \underbrace{\phi(s_w\lambda_w + s_v\lambda_v)}_{\text{HCFLUID}} + \underbrace{(1-\phi)\lambda_r}_{\text{HCROCK}}$$

The HCFLUID and HCROCK enable heat conduction modelling. The keywords must be specified in the RUNSPEC section.
The rock heat conduction coefficient distribution must be loaded in the GRID section using HCONDCFX, HCONDCFY and HCONDCFZ keywords.

Scenario 2

All boundaries of the domain are impermeable and insulated

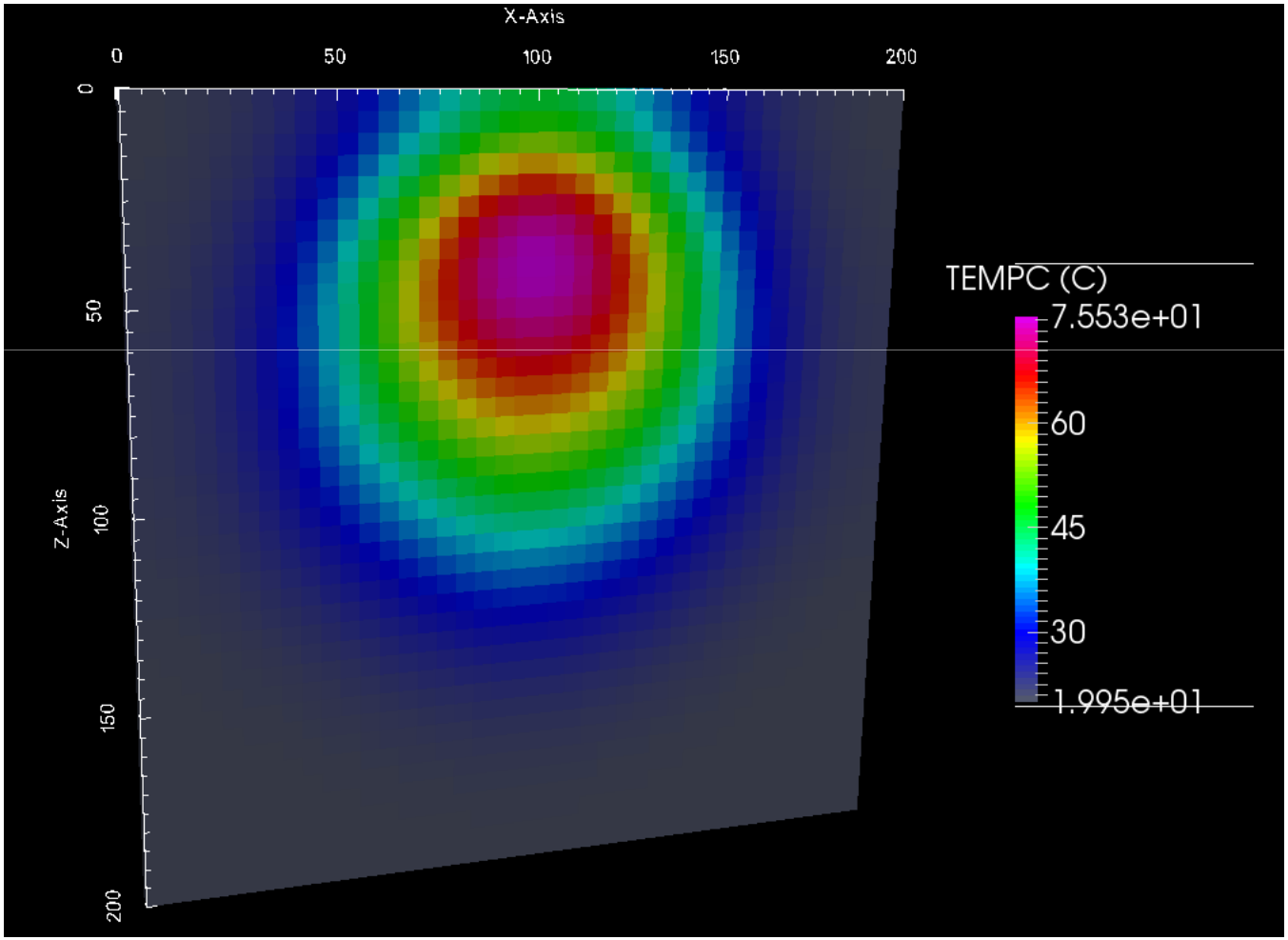


Simulate up to 20000 days

RUN-file (scenario 2)

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

Result (Scenario 2, t=10000 days)



Next day

- Operations on arrays
- Regions
- Boundary conditions
- Point sources

Arithmetic operations

PORO = 0.2
PERMZ = 0.1*PERMX
PRES = 8.5+0.01*DEPTH

