MUFITS
Training Course

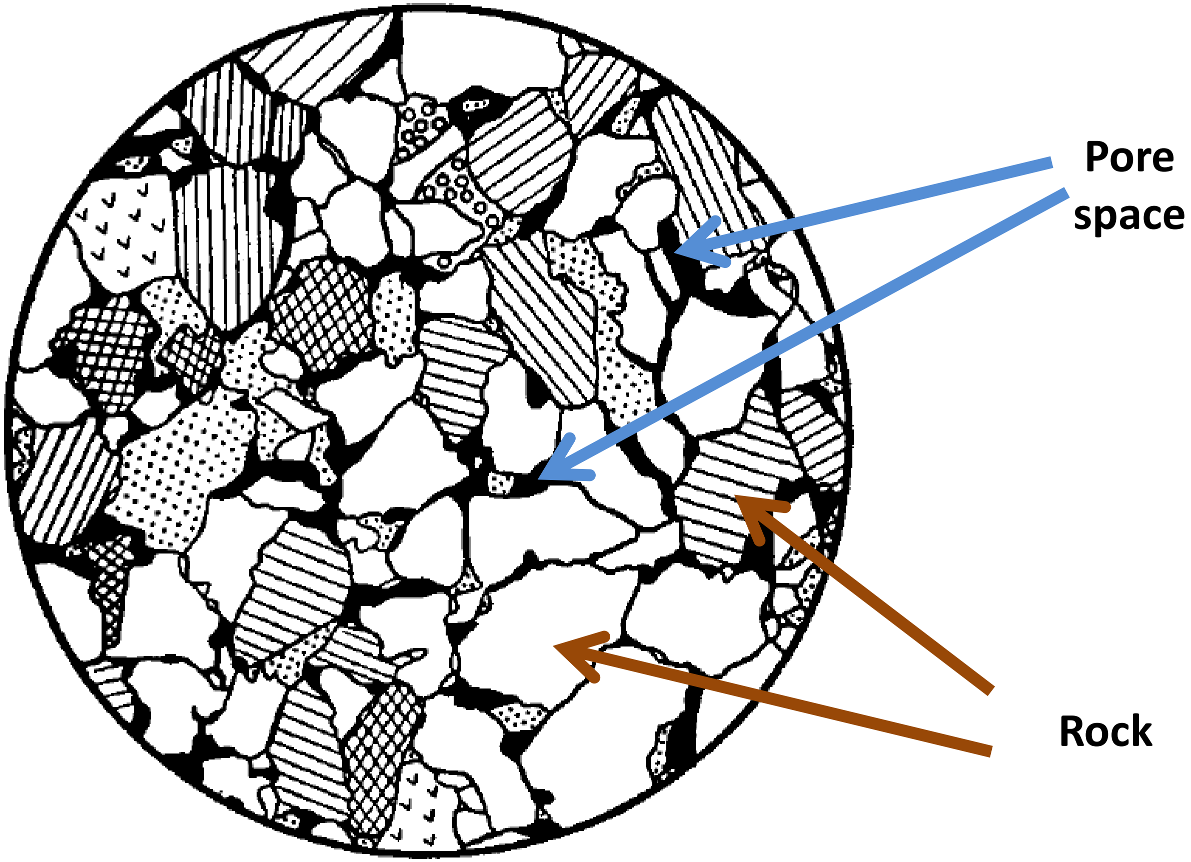
Day 1
Overview; EOS-modules; Fundamentals; SIMPLMOD module

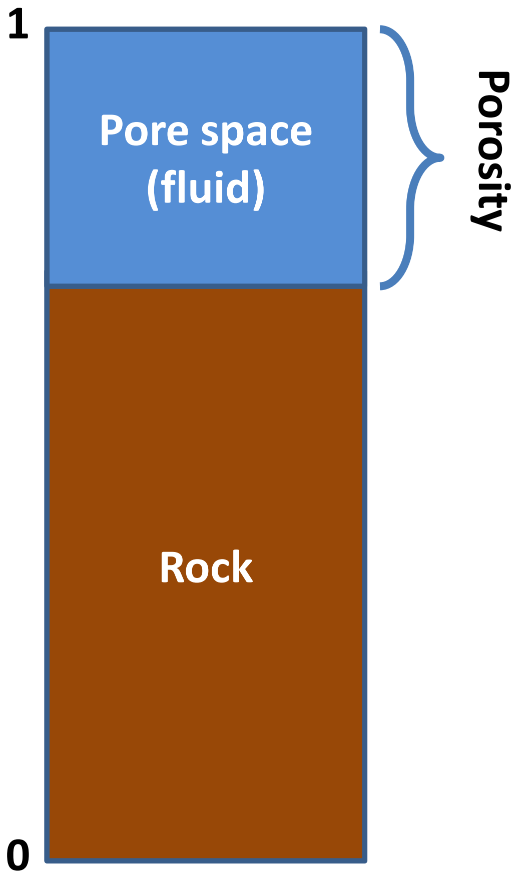
Program; Day 1

- Introduction (mathematical model, EOS-modules);
- Overview of the RUN-file;
- EOS module SIMPLMOD;
- Cartesian grids;
- Arrays loading;
- Heat conduction; Capillary pressure; Isothermal mode

Introduction

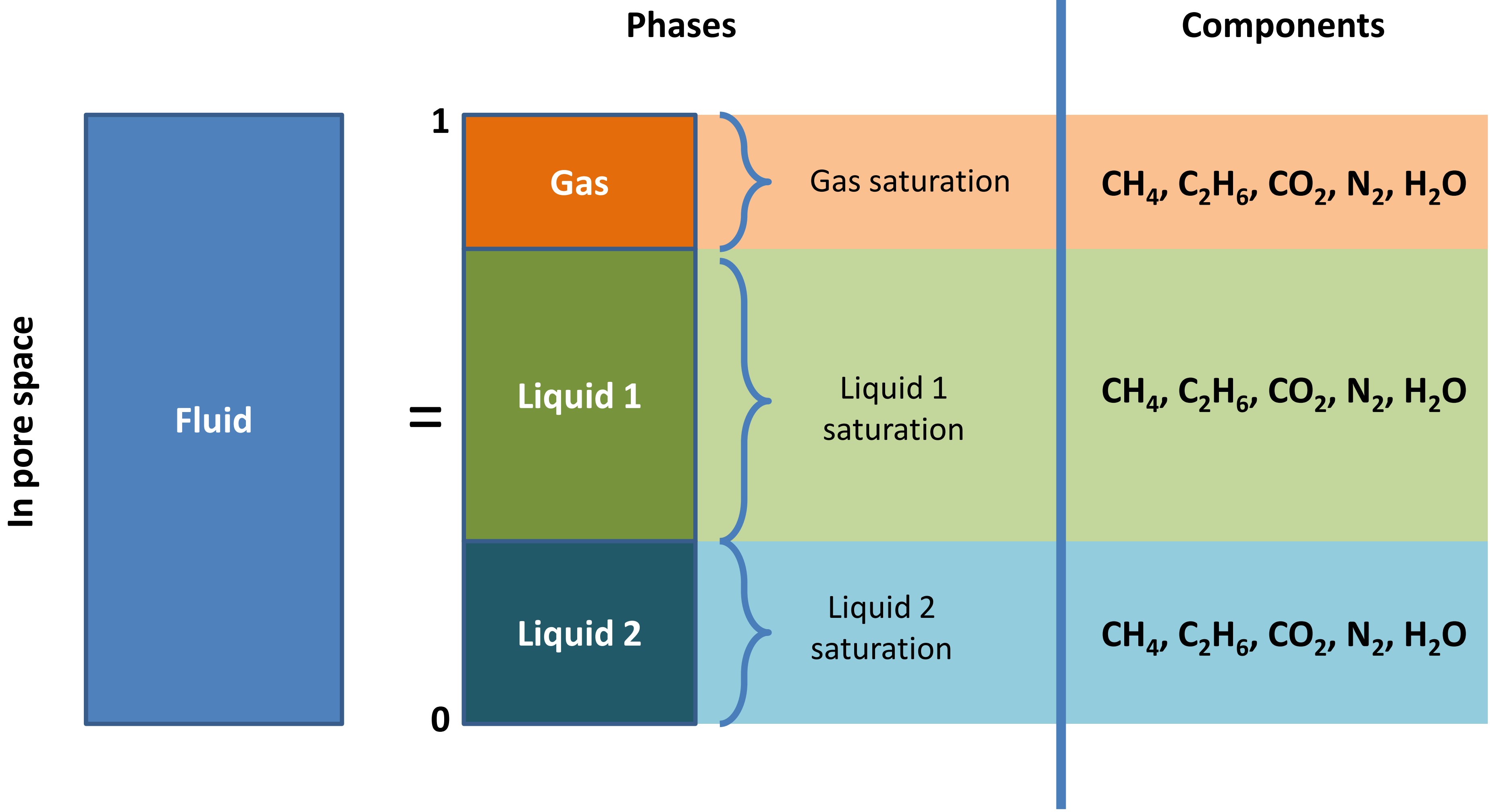
Porous medium





We will not model flows on pore scale!

Components and phase



Phases

Components

In pore space

Fluid

=

1

0

Gas

Liquid 1

Liquid 2

Gas saturation

Liquid 1 saturation

Liquid 2 saturation

CH_4, C_2H_6, CO_2, N_2, H_2O

CH_4, C_2H_6, CO_2, N_2, H_2O

CH_4, C_2H_6, CO_2, N_2, H_2O

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

$$\mathbf{w}_i = -K\frac{K_{ri}}{\mu_i}\;\mathbf{grad}P_i - \rho_i\mathbf{g}$$

- Darcy correlation

$$K_{ri} = K_{ri}(s), \quad P_i - P_k = P_{c,ik}(s)$$

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

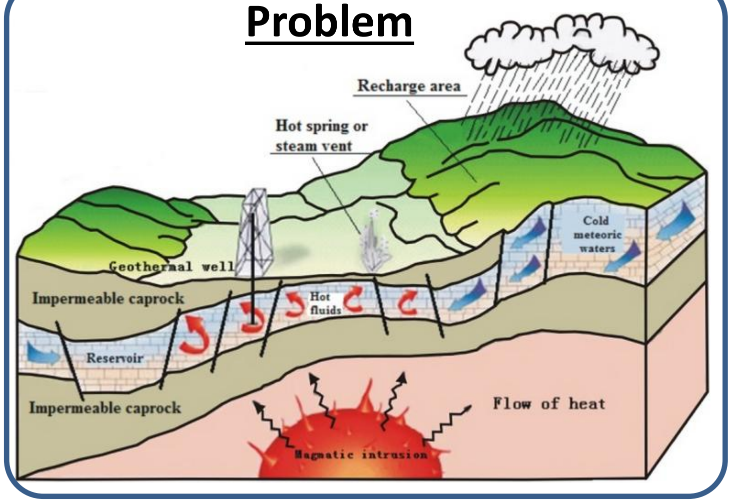
$$\sum_{i=1}^{p} s_i = 1, \qquad \sum_{j=1}^{c} c_{i(j)} = 1$$

- consistency relations

+ Equations of state (for fluid and rocks)

Data flowchart

Problem



RUN-file

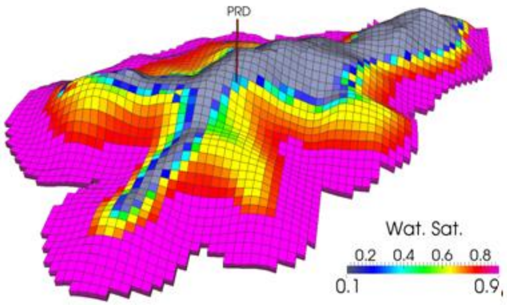
- Grid;
- Porosity and permeability;
- Initial & boundary conditions;
- Rel. permeability;
- Wells and point sources;
- Report time;
- etc.

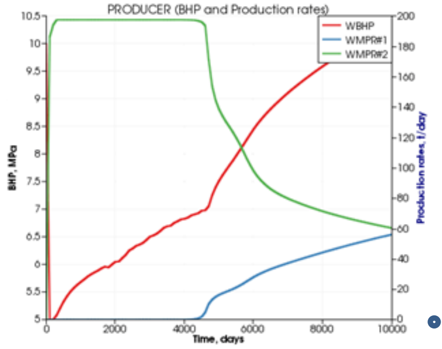
The file is prepared in a Text Editor

Runs via MPI interface

Simulator

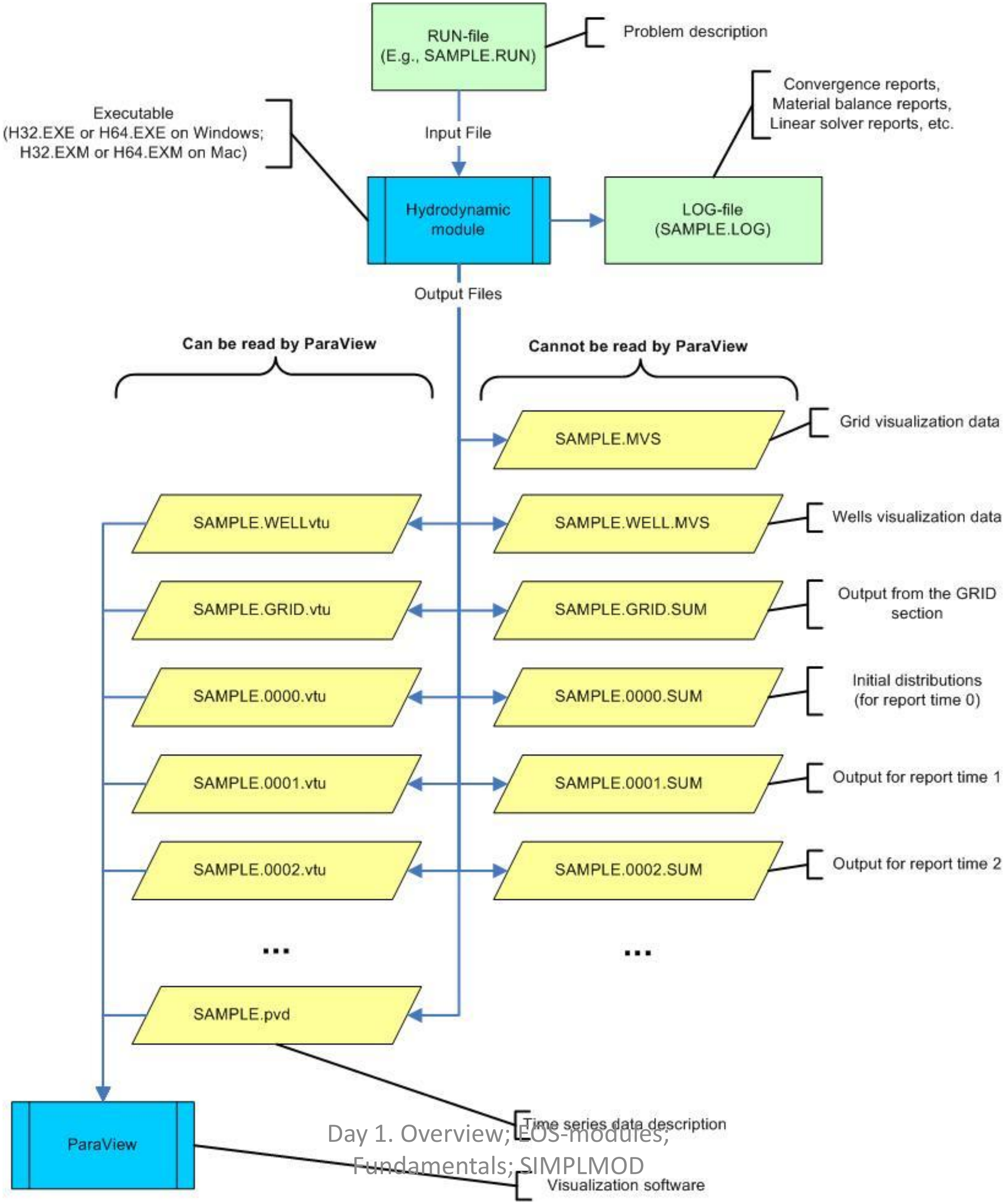
Results





In ParaView

Data flowchart



RUN-file (E.g., SAMPLE.RUN)

Problem description

Input File

Executable (H32.EXE or H64.EXE on Windows; H32.EXM or H64.EXM on Mac)

Hydrodynamic module

LOG-file (SAMPLE.LOG)

Convergence reports, Material balance reports, Linear solver reports, etc.

Output Files

Can be read by ParaView

Cannot be read by ParaView

SAMPLE.MVS

Grid visualization data

SAMPLE.WELLvtu

SAMPLE.WELL.MVS

Wells visualization data

SAMPLE.GRID.vtu

SAMPLE.GRID.SUM

Output from the GRID section

SAMPLE.0000.vtu

SAMPLE.0000.SUM

Initial distributions (for report time 0)

SAMPLE.0001.vtu

SAMPLE.0001.SUM

Output for report time 1

SAMPLE.0002.vtu

SAMPLE.0002.SUM

Output for report time 2

...

...

SAMPLE.pvd

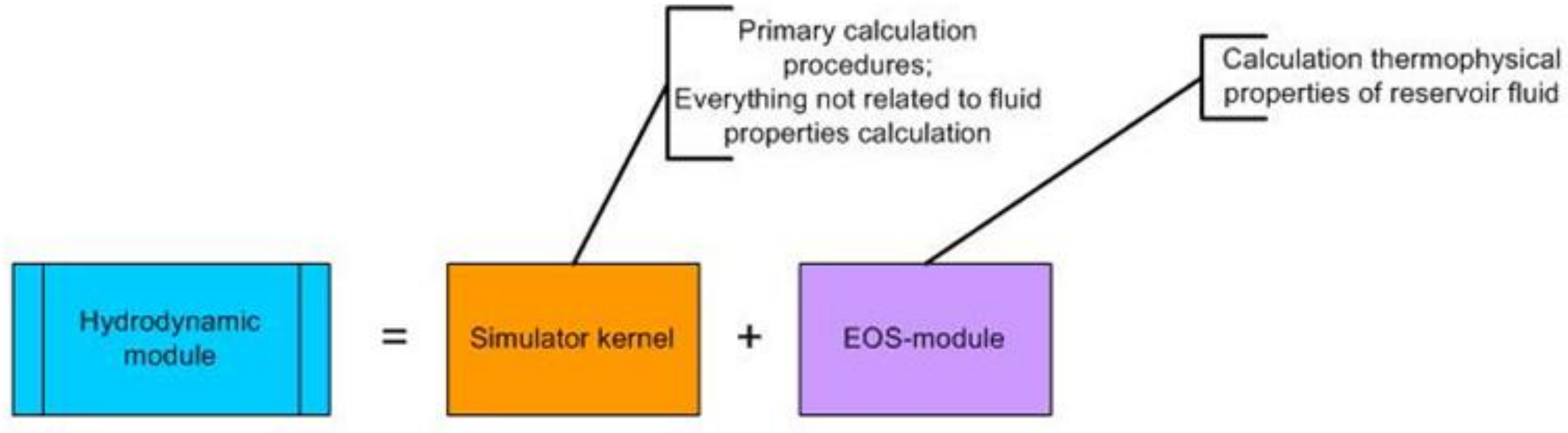
Time series data description

ParaView

Visualization software

EOS modules

EOS modules



Primary calculation procedures;
Everything not related to fluid properties calculation

Calculation thermophysical properties of reservoir fluid

Hydrodynamic module = Simulator kernel + EOS-module

EOS-modules

Name	Components /phases	Potential applications
BLACKOIL	water/oil/gas	Petroleum reservoirs (underground CO_2 storage)
BINMIXT	Binary mixture; H_2O/CO_2	Geothermal applications; underground CO_2 storage

GASSTORE (β-версия)	water/gas/salt;
	Gas = CH_4, CO_2
	Salt = NaCl
		Geothermal applications;
		underground CO_2 storage;
		Gas storage

SIMPLMOD	A single component/phase (+ an arbitrary number of components/phases)	Immiscible displacement in porous media
T2EOS1	Water/water vapour	Geothermal applications

Nonisothermal modules

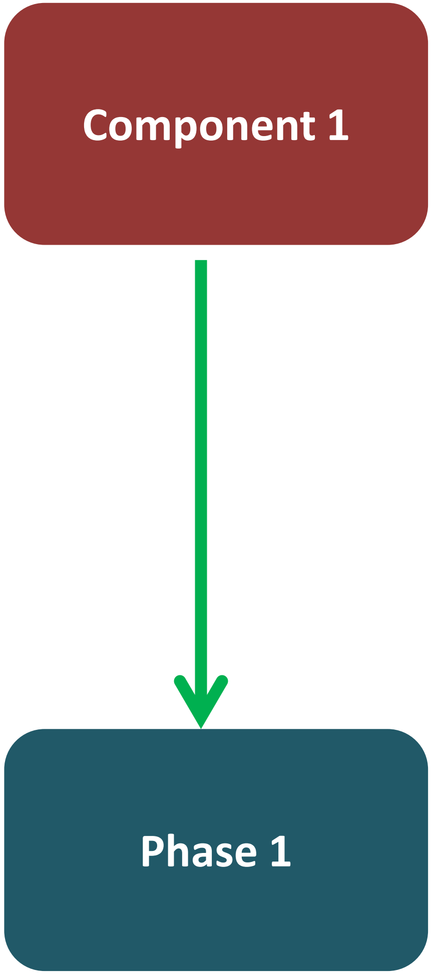
EOS module SIMPLMOD

T≠const

Components

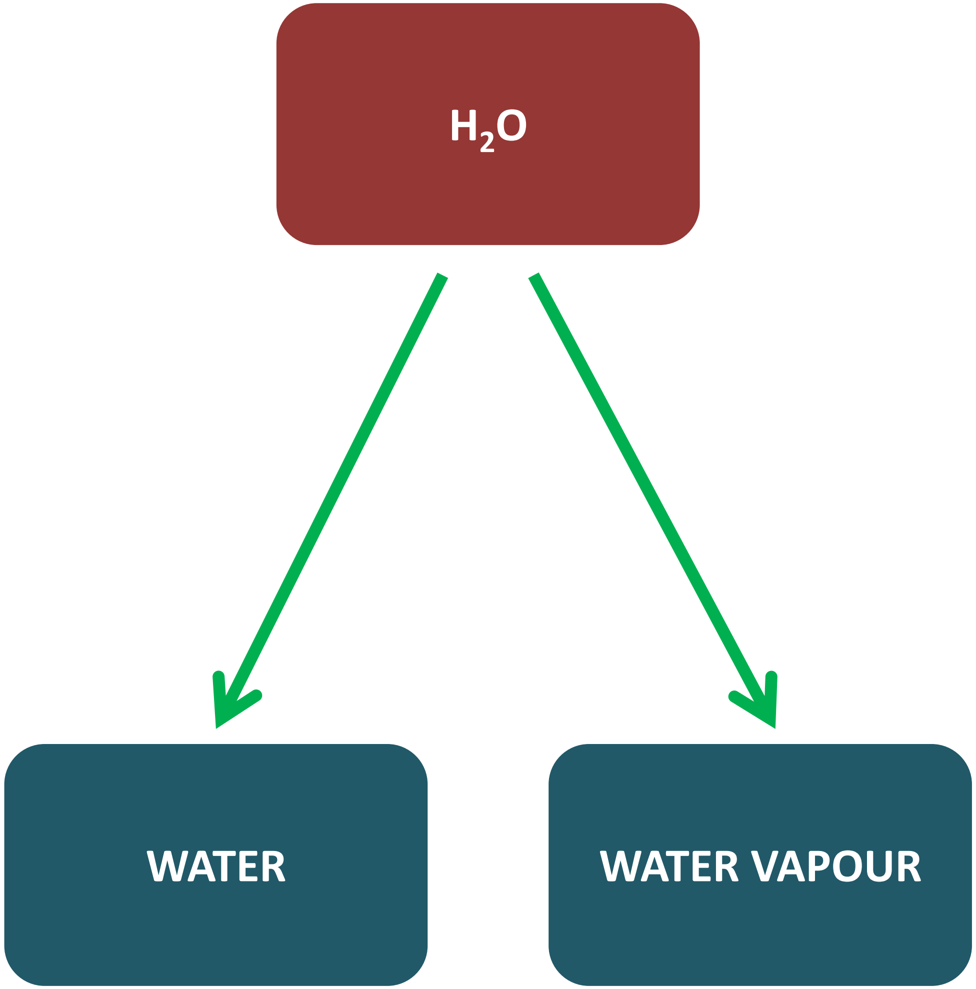
Keywords

Phases



EOS module T2EOS1

T≠const



Components

H_2O

Keywords

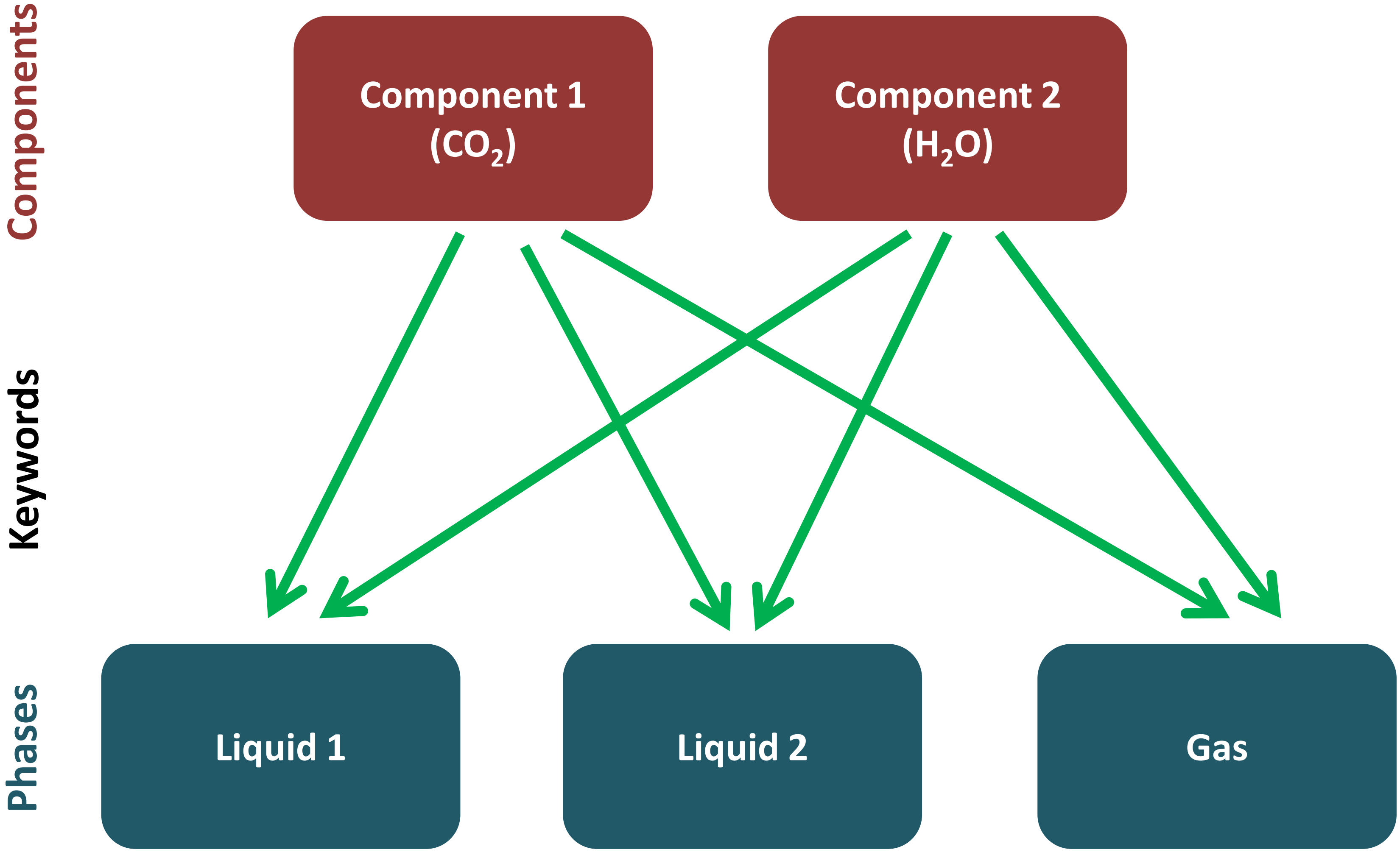
Phases

WATER

WATER VAPOUR

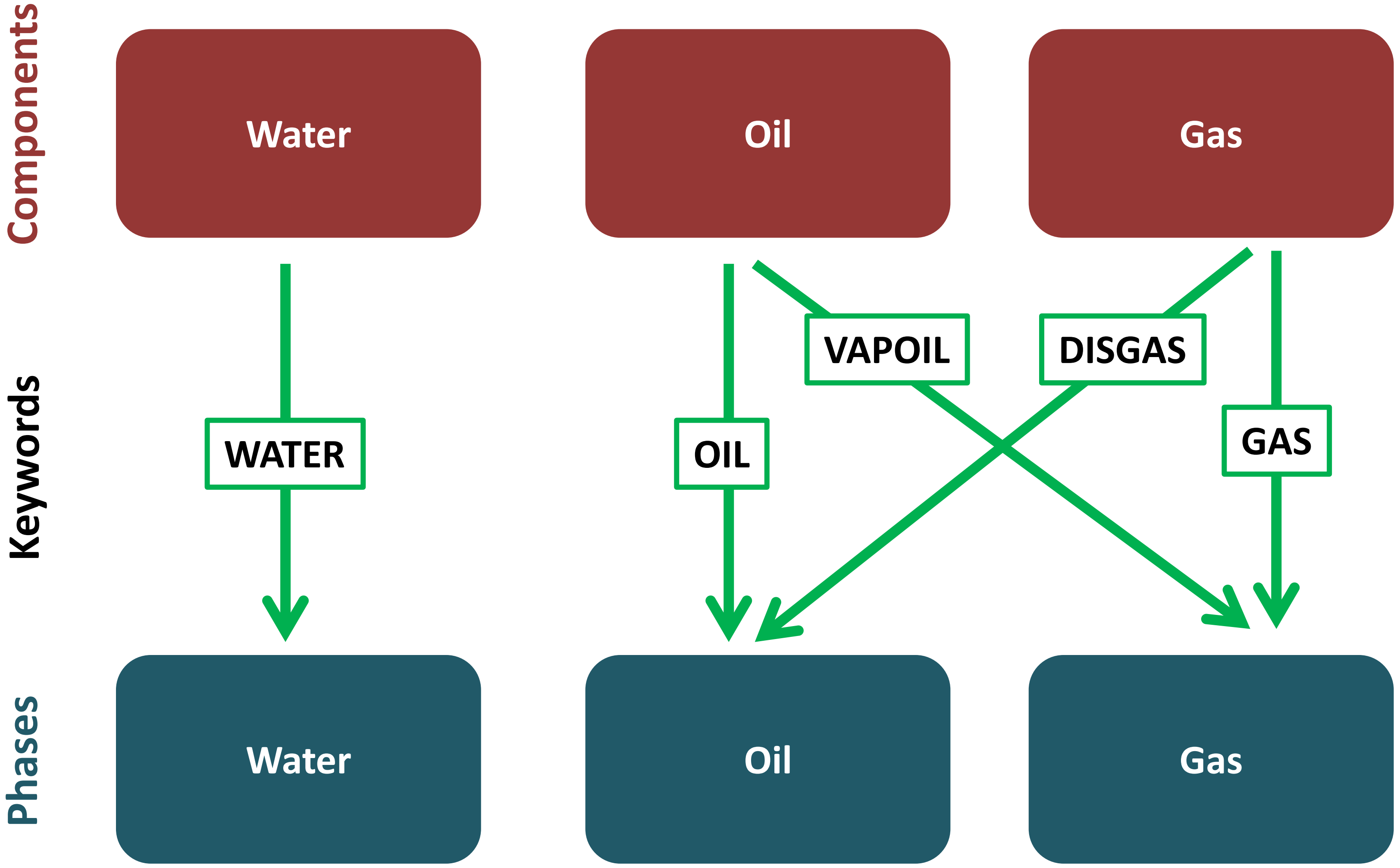
EOS module BINMIXT

T≠const



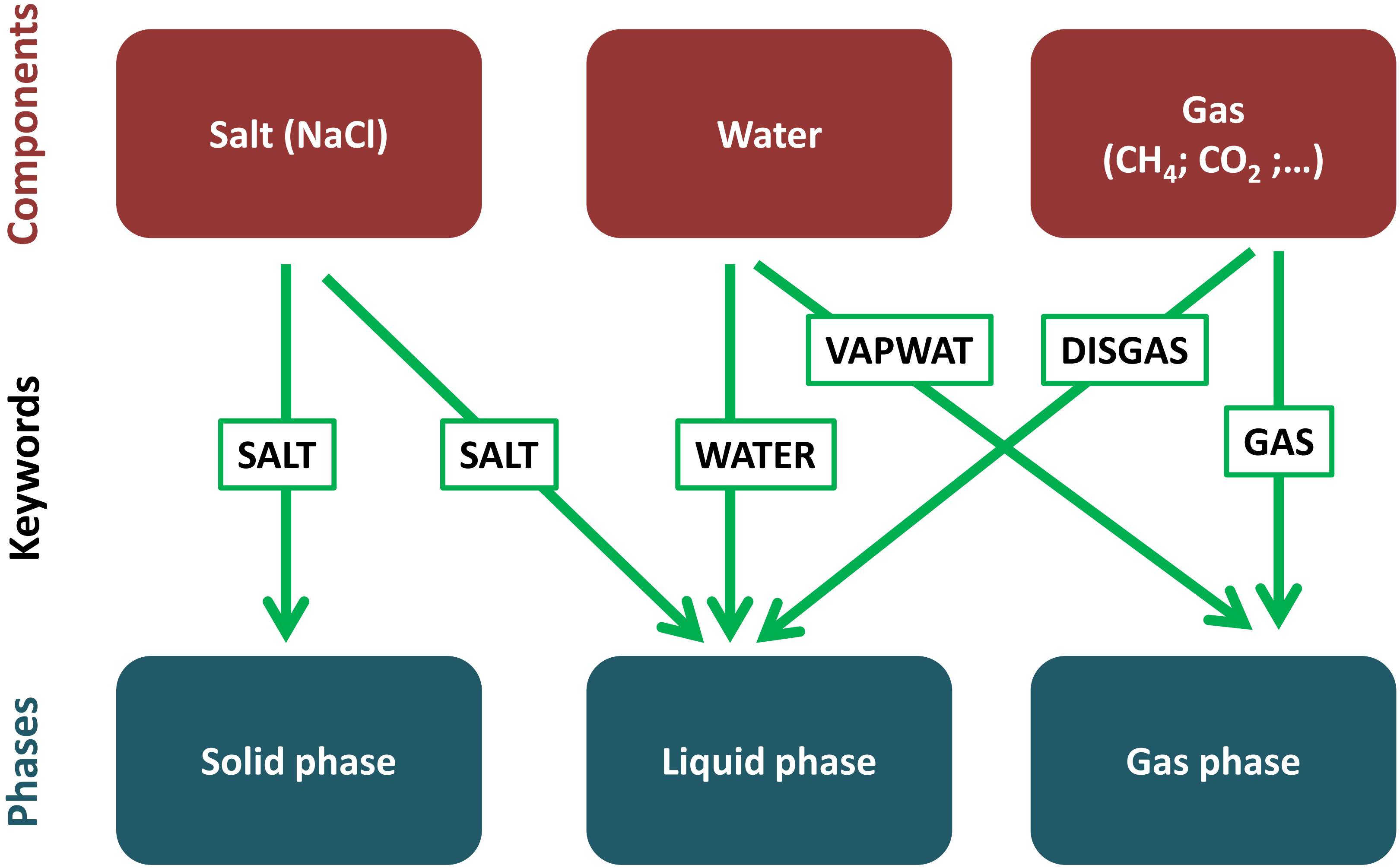
EOS module BLACKOIL

T=const



EOS module GASSTORE

T≠const



Components

Salt (NaCl)

Water

Gas
(CH_4; CO_2 ;…)

Keywords

Phases

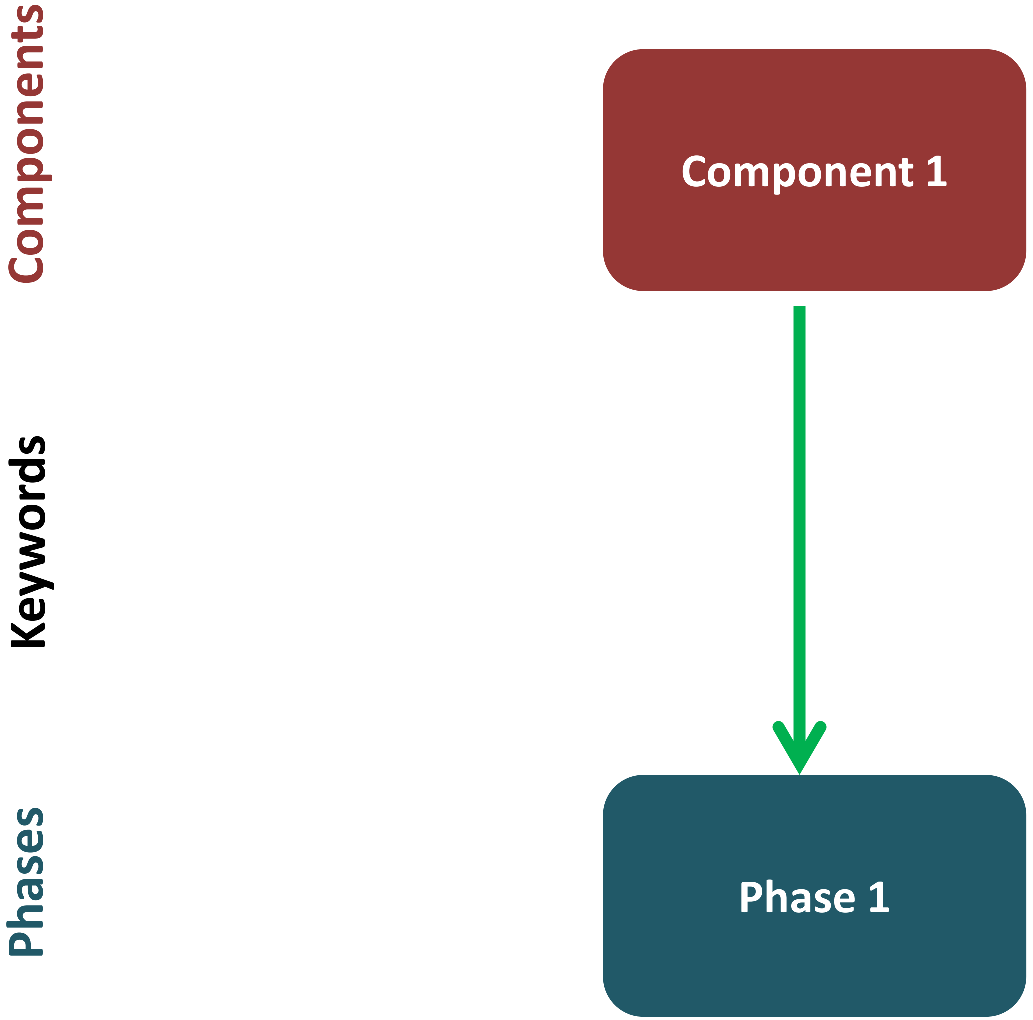
Solid phase

Liquid phase

Gas phase

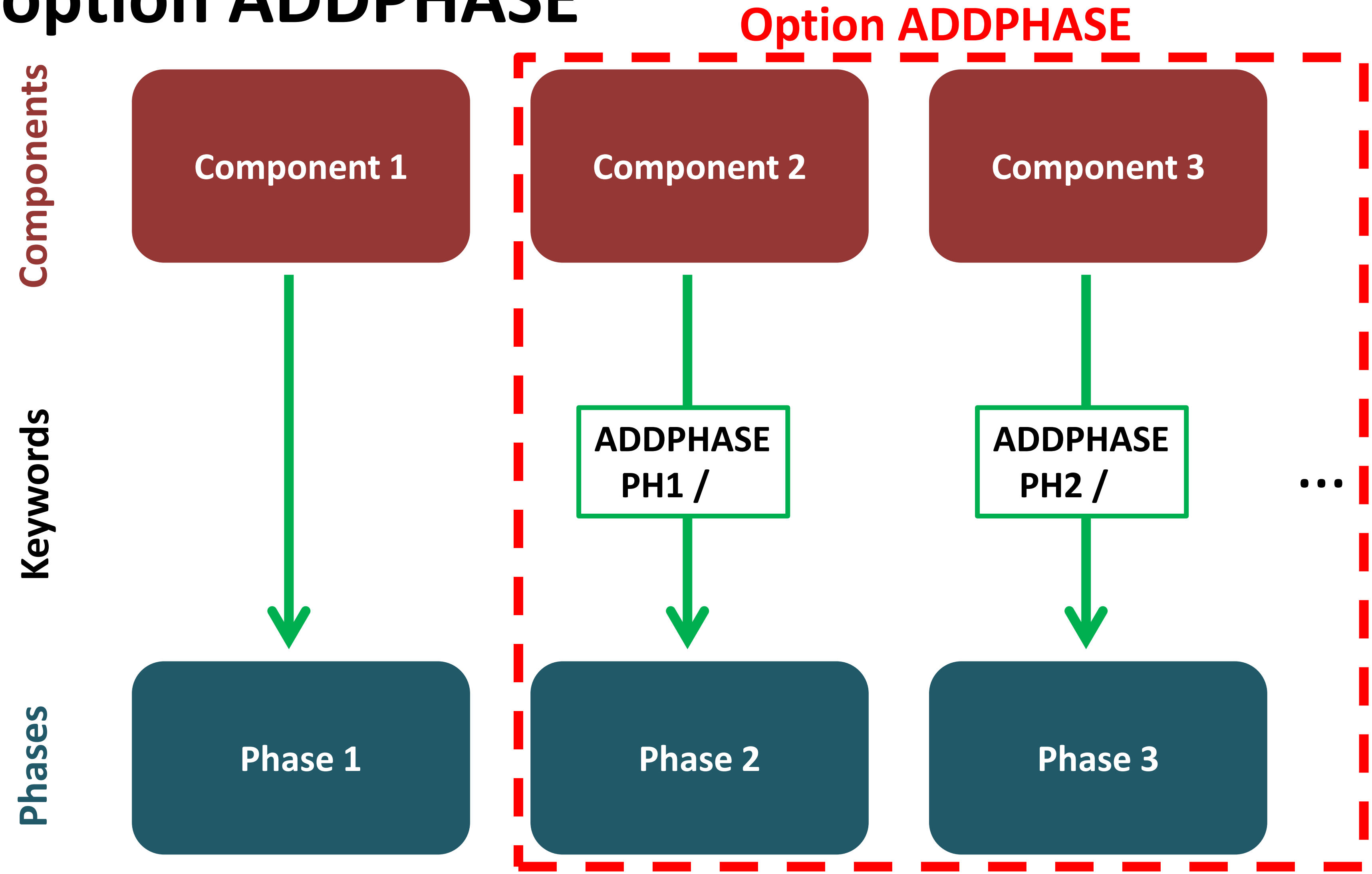
EOS module SIMPLMOD

T≠const



EOS module SIMPLMOD with option ADDPHASE

T≠const



Option ADDPHASE

Components

Keywords

Phases

Component 1

Component 2

Component 3

ADDPHASE PH1 /

ADDPHASE PH2 /

…

Phase 1

Phase 2

Phase 3

RUN-file
(or data-file)

RUN-file (Scenario 1)

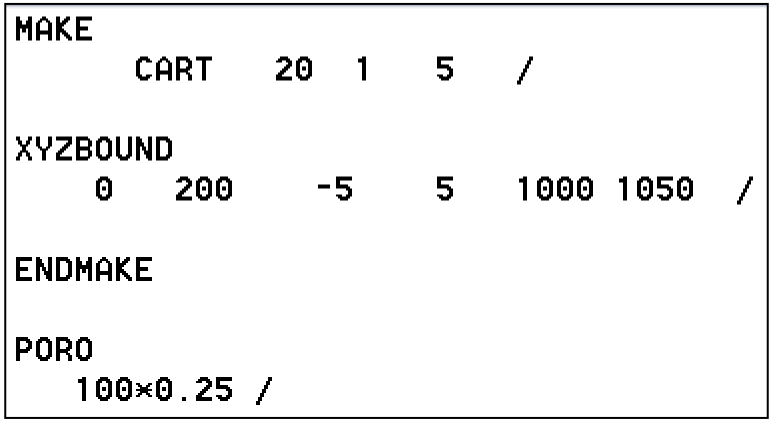
Open RUN-file (SCENARIO-C1.RUN) in a text editor

Keywords

The keywords are the instructions in the RUN-file to the simulator. By using keywords the grid, the physical model,
the relative permeabilitiy, 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 /
```

The keywords syntax is provided in the section Keywords of the Reference manual

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.

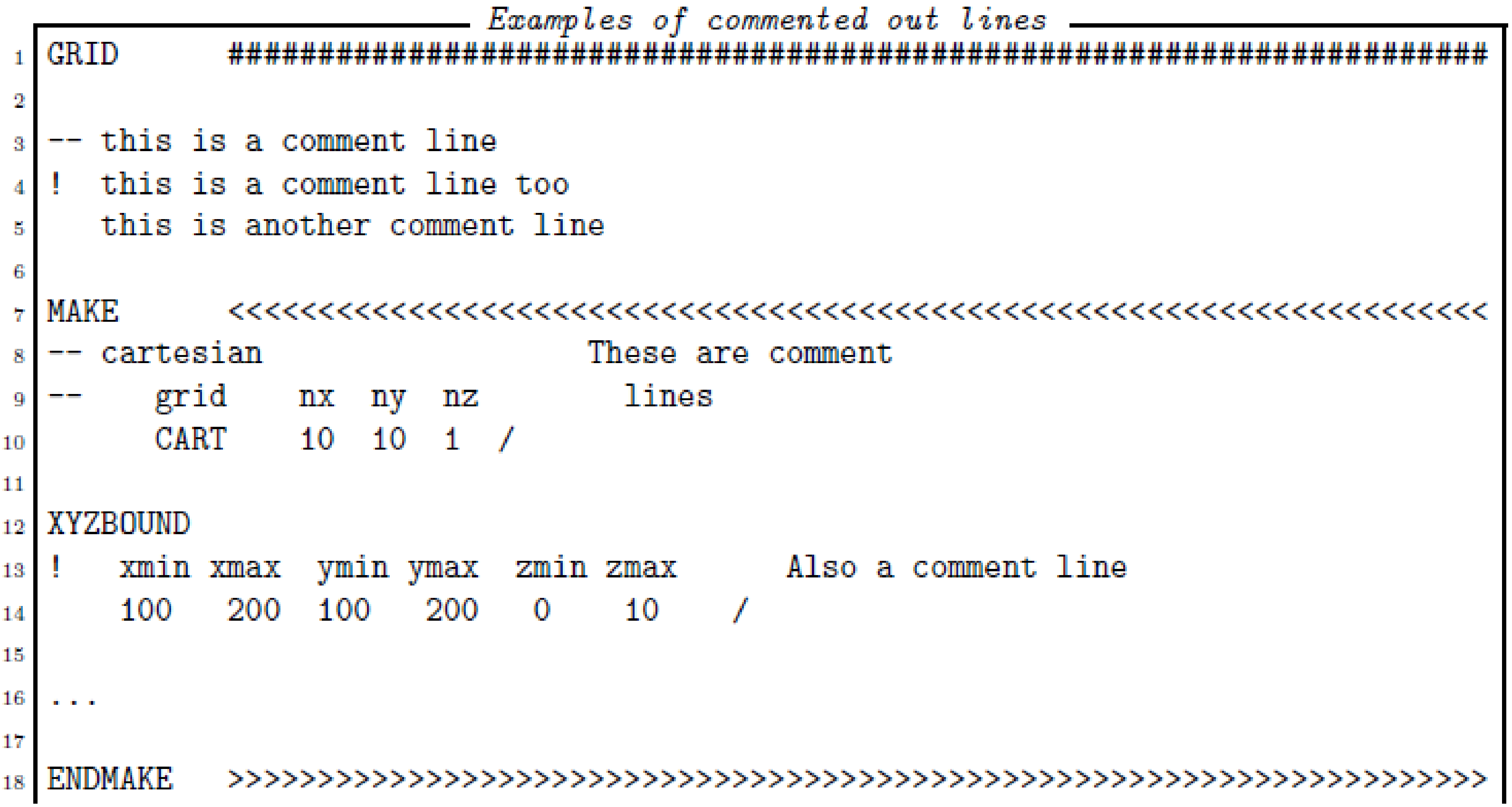
Examples of Mnemonics

Mnemonic	Parameter
PRES	Pressure (MPa)
TEMP	Temperature (K)
TEMPC	Temperature (C)
SWAT	Water saturation
SGAS	Gas saturation

The mnemonics description is provided in the section Mnemonics.

Comments in RUN-file

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

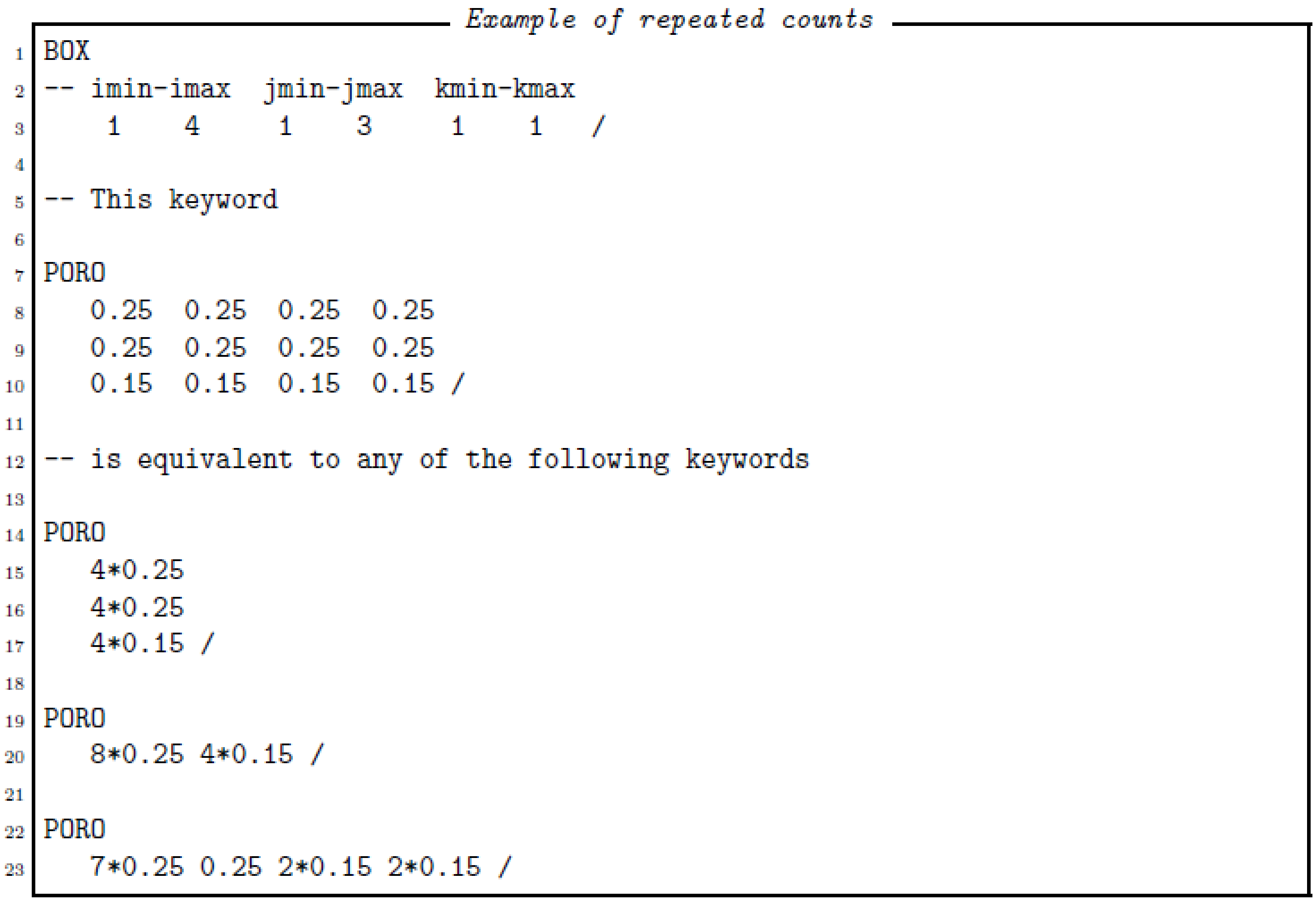


Examples of commented out lines

```
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.

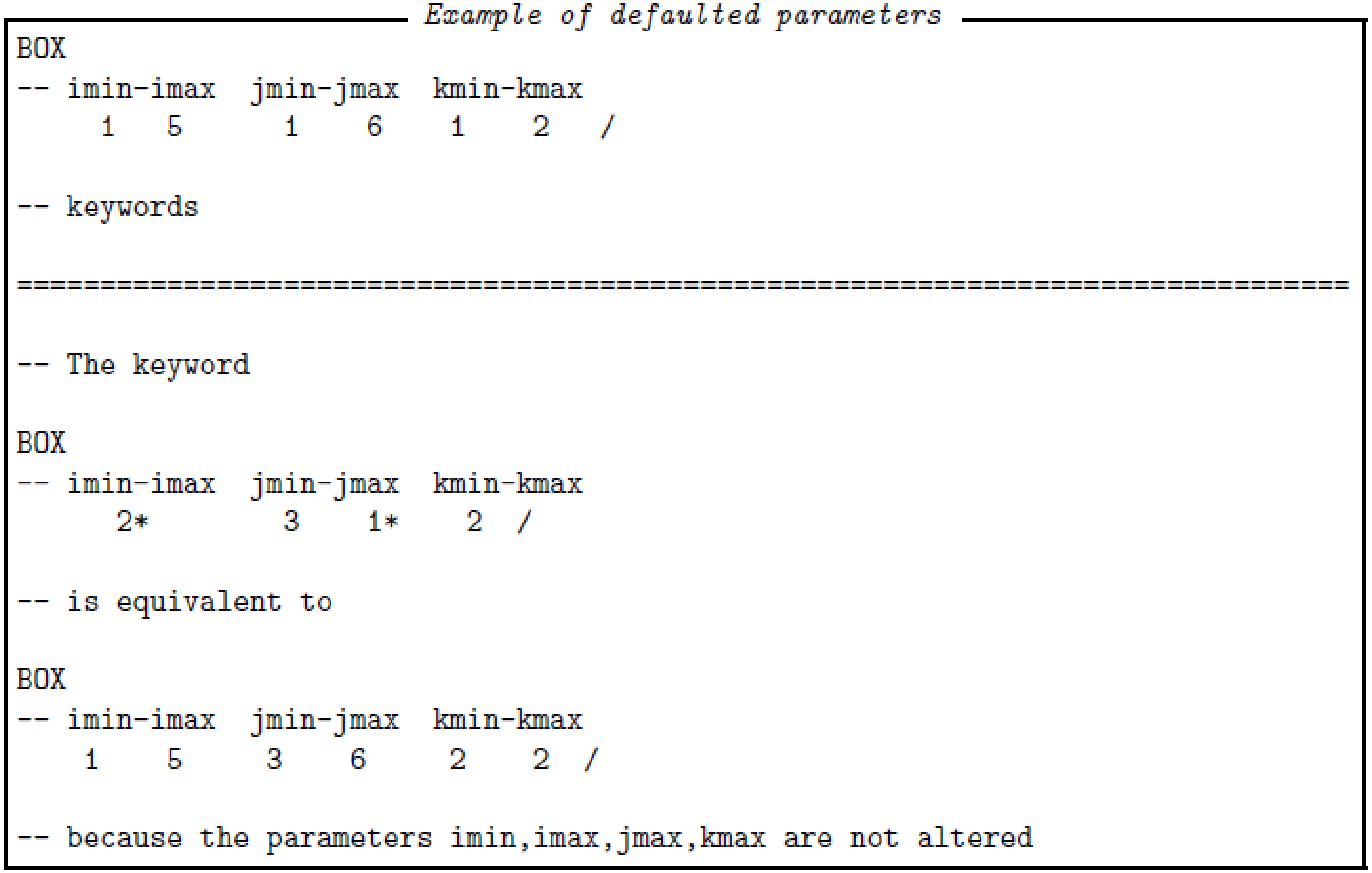


Example of repeated counts

```
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
```

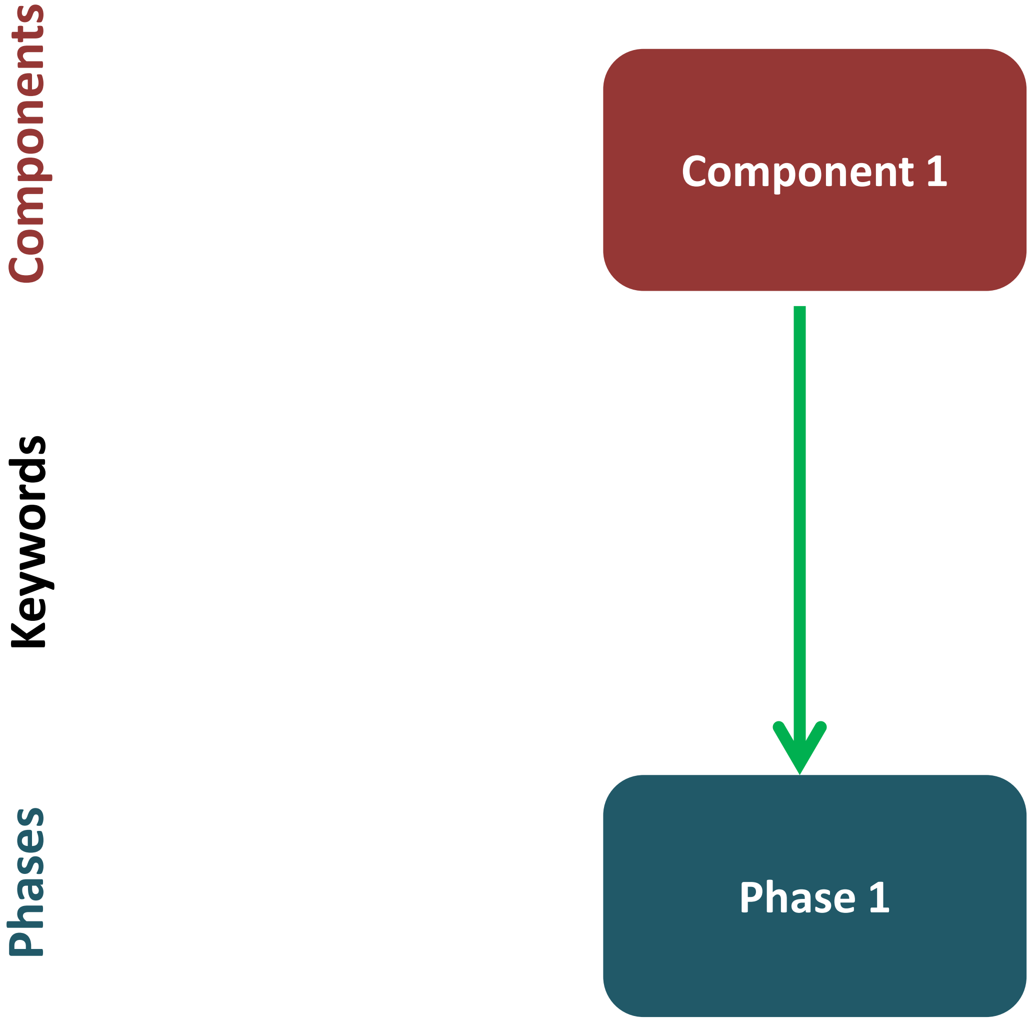
RUN-file sections

Name	Data to be specified	
RUNSPEC	Global properties; EOS module; Units	**Options**
GRID	Grid; Porosity; Absolute permeability; Rock heat conductivity	**Input data loading**
EDIT (optional)	We will not use this section in the course	
PROPS	Thermophysical properties of the fluid and rock; Relative permeability; Capillary pressure	
INIT	Initial conditions; Data to be saved at every report time	
SCHEDULE	Report times; Tuning of the simulation	**Simulating**
POST (optional)	Conversion to ParaView format; Consolidated reports for wells; grid blocks, etc.	**Post-processing**

SIMPLMOD; Mathematical mode; First example

EOS module SIMPLMOD

T≠const



Mathematical model

By default, this term is disabled

$$\frac{\partial \phi\rho}{\partial t} + \mathrm{div}(\rho\mathbf{w}) = 0$$

- mass balance equation

$$\frac{\partial}{\partial t}(\phi\rho e + (1-\phi)\rho_r e_r) + \mathrm{div}(\rho h\mathbf{w}) = \mathrm{div}(\lambda\,\mathbf{grad}T)$$

- energy balance equation

$$\mathbf{w} = -\frac{K}{\mu}\left(\mathbf{grad}P - \rho\mathbf{g}\right)$$

- Darcy’s law

$$\rho(P,T); \quad h(P,T); \quad \mu(P,T)$$

$$\rho_r = const; \; e_r = C_r T$$

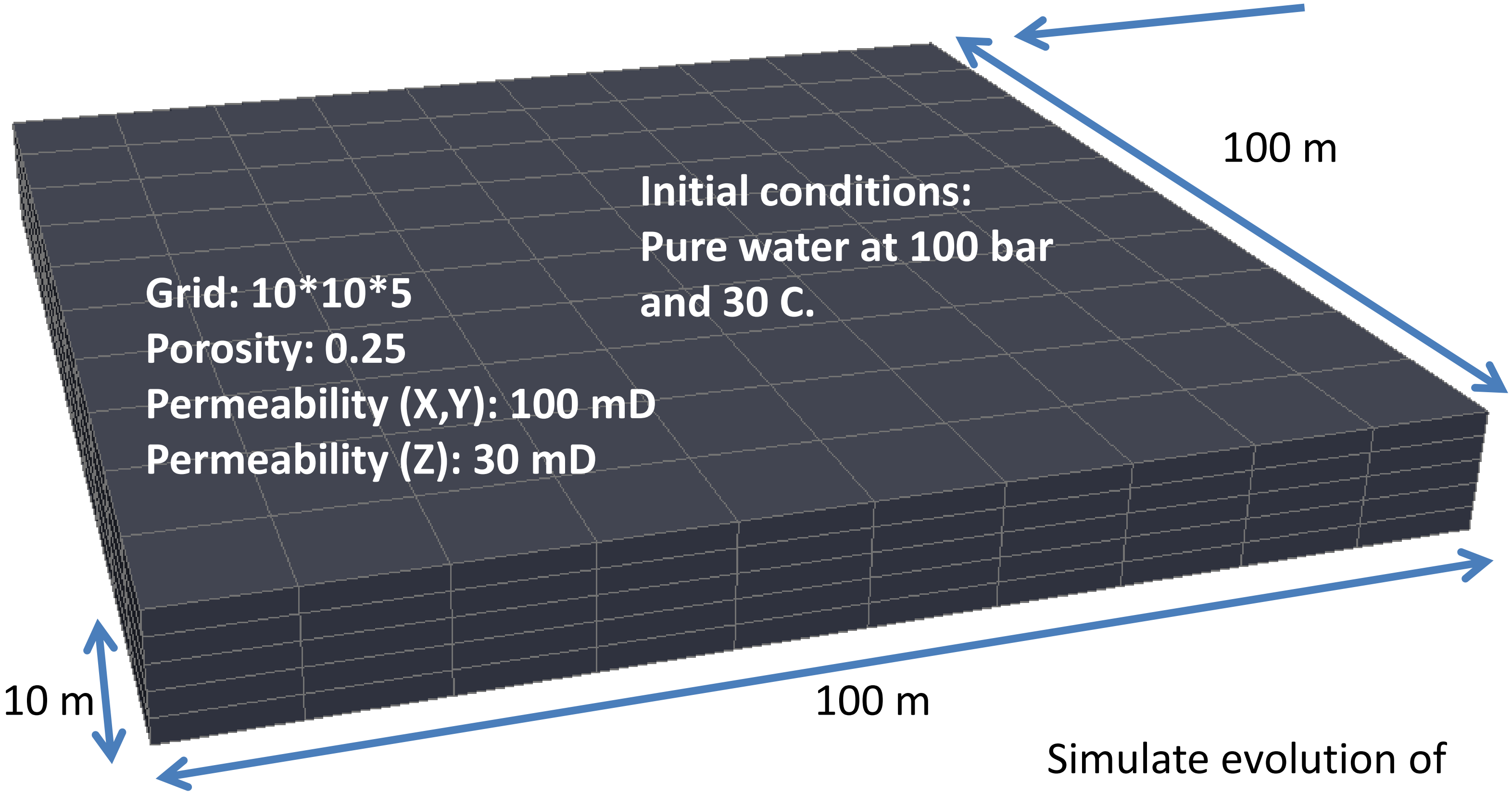
- EoS

$$h = e + \frac{P}{\rho}$$

- a thermodynamic relation

Formulation of the problem (Scenario-C1)

Top surface at 1000 m depth



Initial conditions:
Pure water at 100 bar and 30 C.

Grid: 10*10*5
Porosity: 0.25
Permeability (X,Y): 100 mD
Permeability (Z): 30 mD

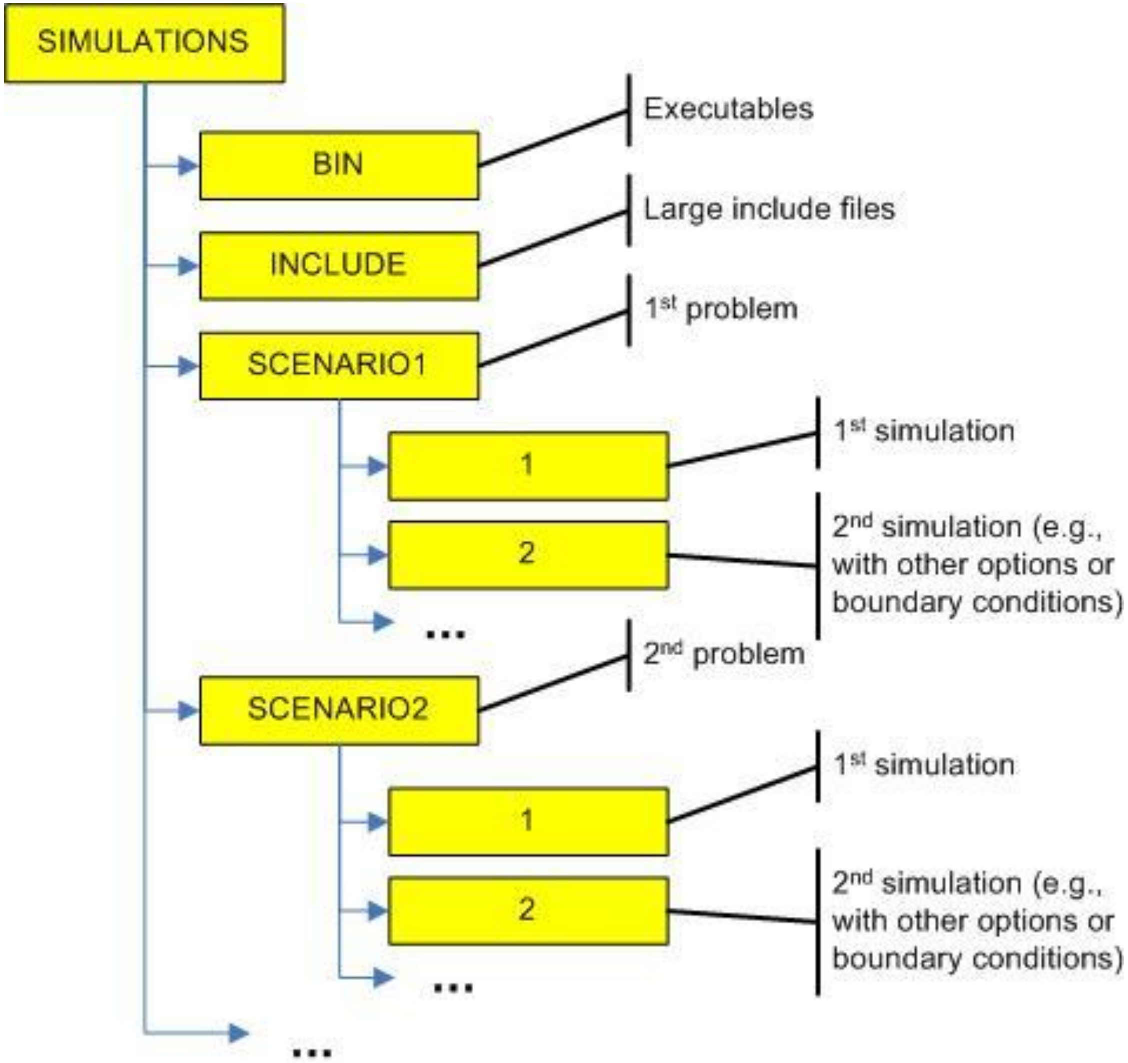
Simulate evolution of parameters for 100 days.

All boundaries are impermeable.

RUN-file (Scenario 1)

Open RUN-file (SCENARIO-C1.RUN) in a text editor

Folders hierarchy



Running the problem

Option 1

Windows:
Copy the H64.BAT (command file) in your working folder (should be SIMULATIONS/SCENARIO-C1/0/). Double click on this file to start simulation.

Option 2

In the folder SIMULATIONS/SCENARIO-C1/0/   execute:

Windows:

```
  “…mpiexec.exe” –n 1 ../../BIN/H64.EXE SCENARIO-C1.RUN > SCENARIO-C1.LOG
```

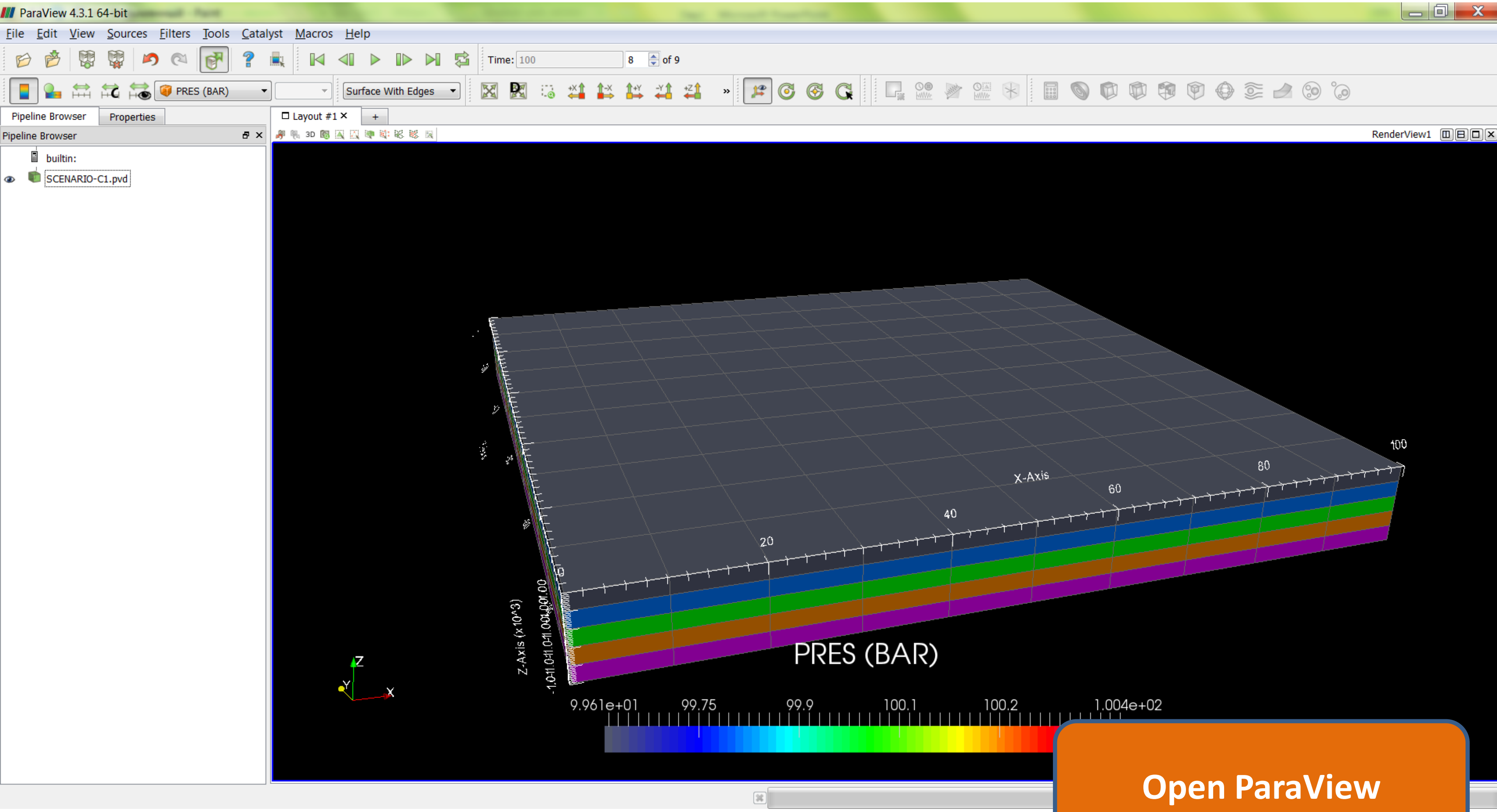
Mac:

```
   mpirun –n 1  ./../../BIN/H64.EXM SCENARIO-C1.RUN > SCENARIO-C1.LOG
```

Linux:

```
    mpirun –n 1   ./../../BIN/H64.EXL SCENARIO-C1.RUN > SCENARIO-C1.LOG
```

Results in ParaView

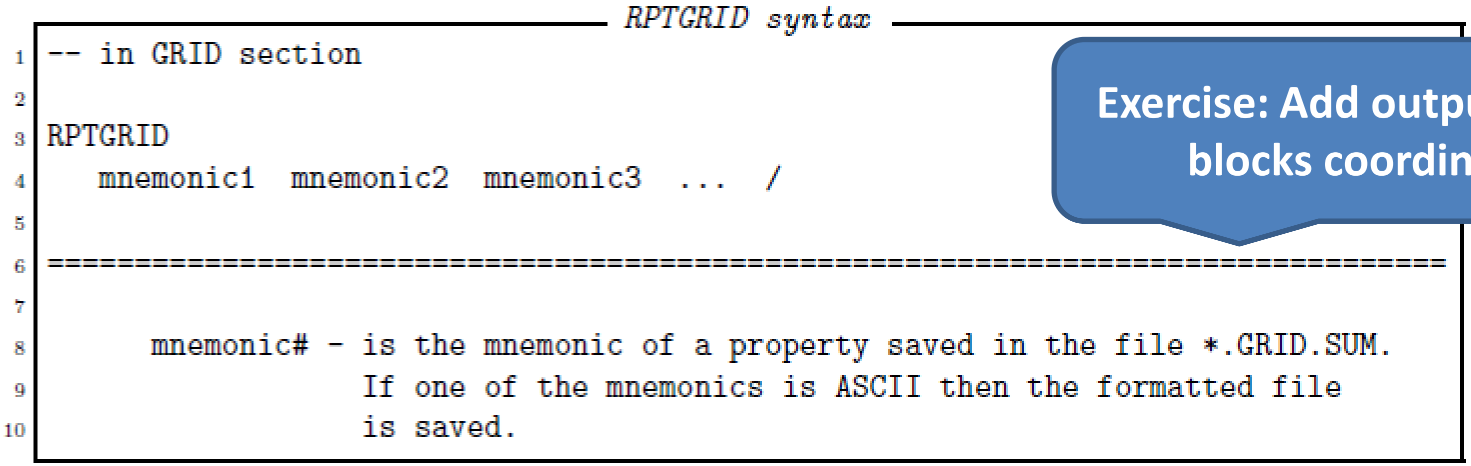


Open ParaView

Fundamentals

Control on output data

The output in the file SCENARIO-C1.GRID.SUM is controlled by the RPTGRID keyword

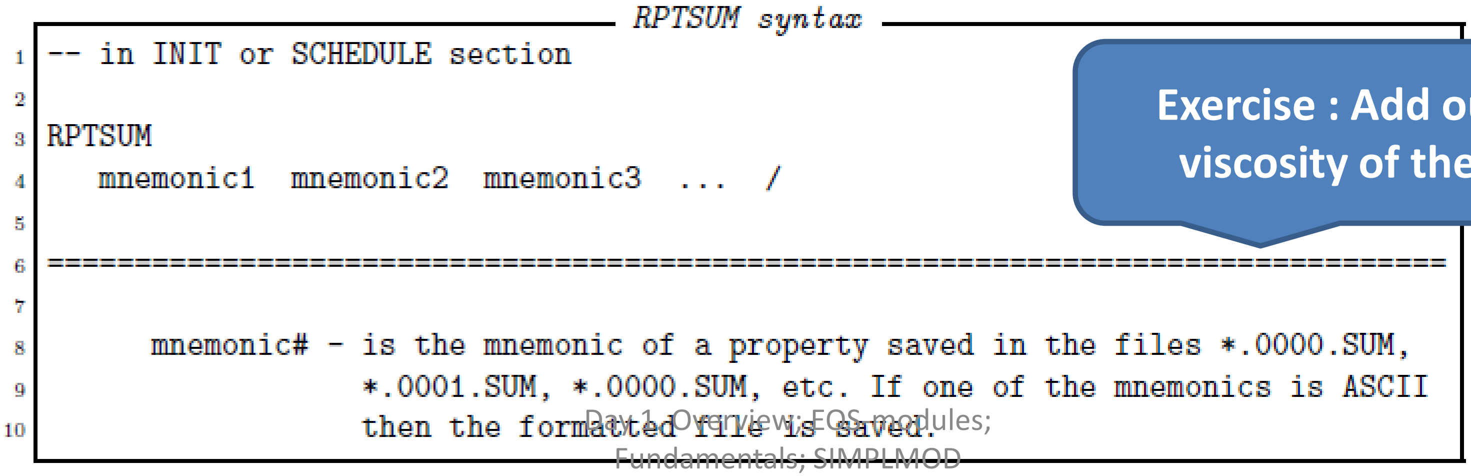


RPTGRID syntax

```
-- in GRID section

RPTGRID
   mnemonic1  mnemonic2  mnemonic3  ...  /

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

      mnemonic# - is the mnemonic of a property saved in the file *.GRID.SUM.
                  If one of the mnemonics is ASCII then the formatted file
                  is saved.
```

Exercise: Add output of grid blocks coordinates

The output in the file SCENARIO-C1.####.SUM is controlled by the RPTSUM keyword



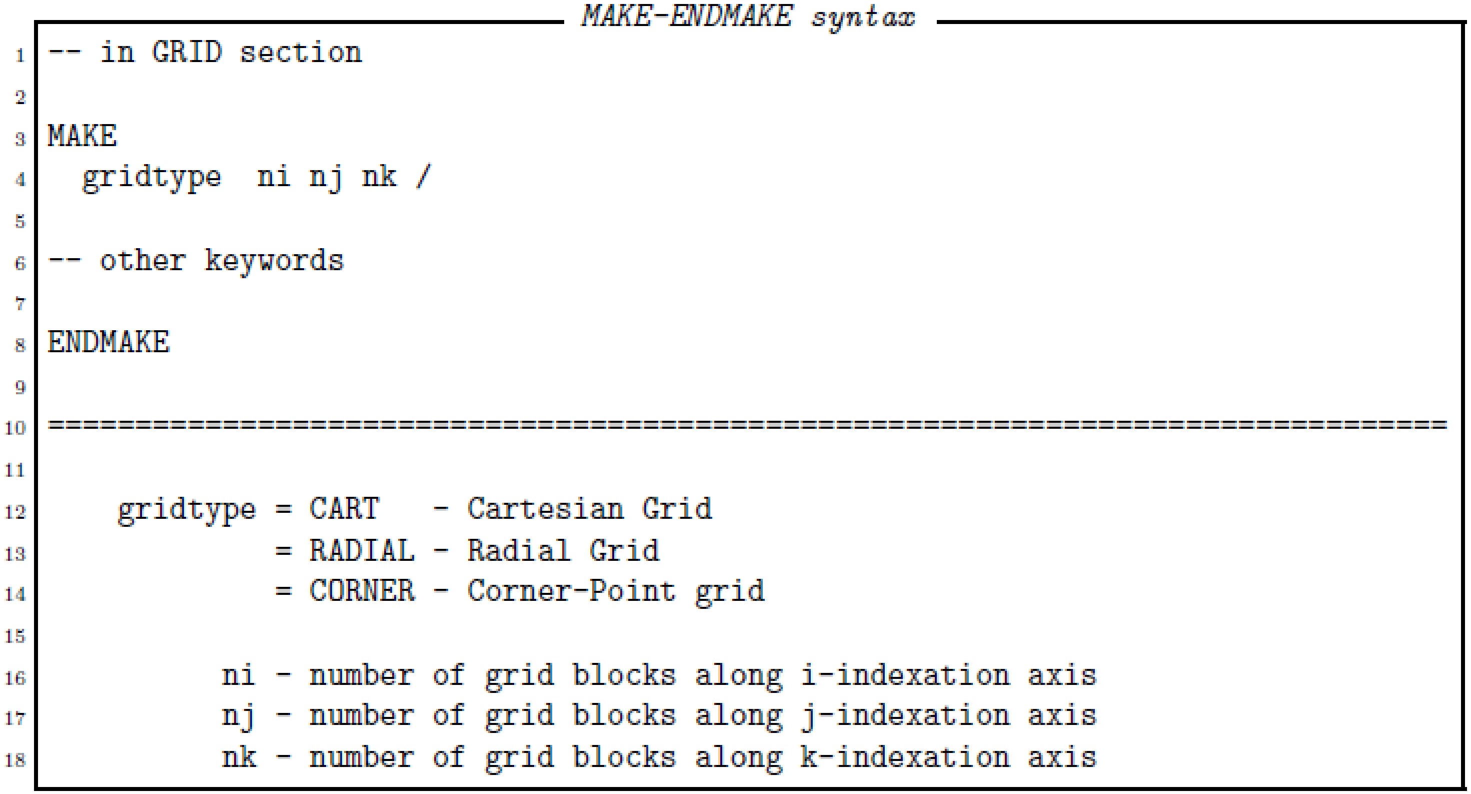
RPTSUM syntax

```
-- in INIT or SCHEDULE section

RPTSUM
   mnemonic1  mnemonic2  mnemonic3  ...  /

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

      mnemonic# - is the mnemonic of a property saved in the files *.0000.SUM,
                  *.0001.SUM, *.0000.SUM, etc. If one of the mnemonics is ASCII
                  then the formatted file is saved.
```

Exercise : Add output of viscosity of the phase

Cartesian grids

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

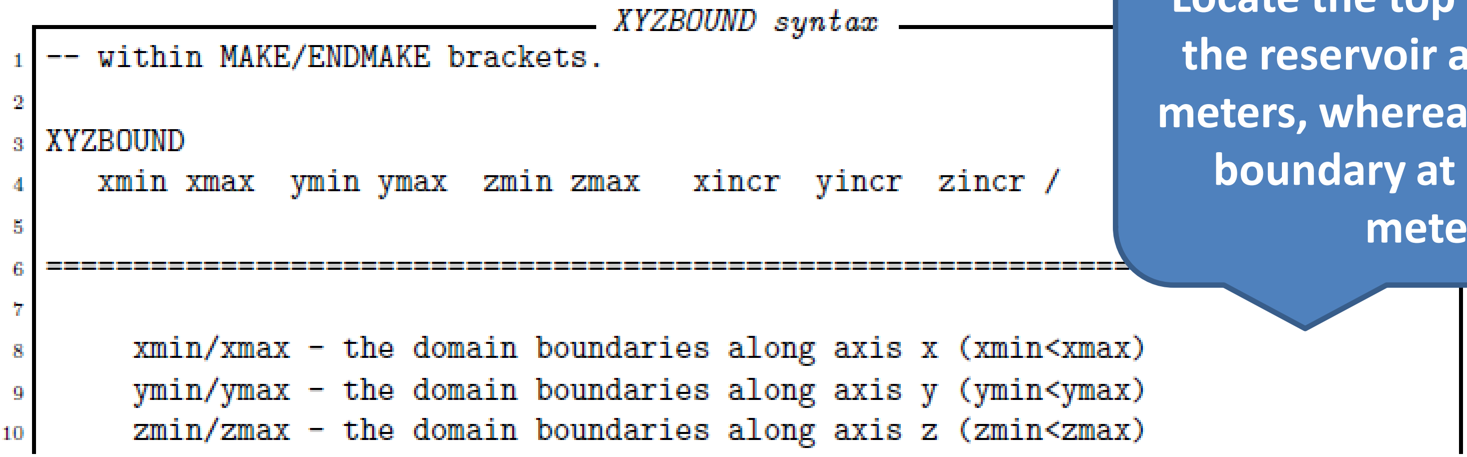


MAKE-ENDMAKE syntax

```
-- 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
```

Cartesian grids

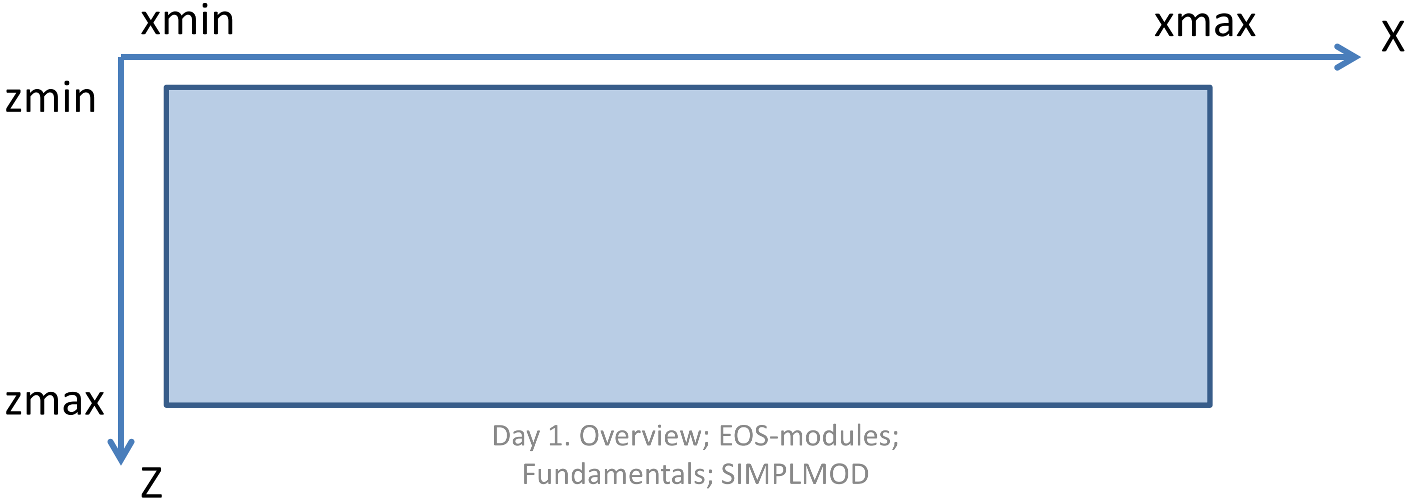
The domain boundaries are defined by the keyword **XYZBOUND**



XYZBOUND syntax

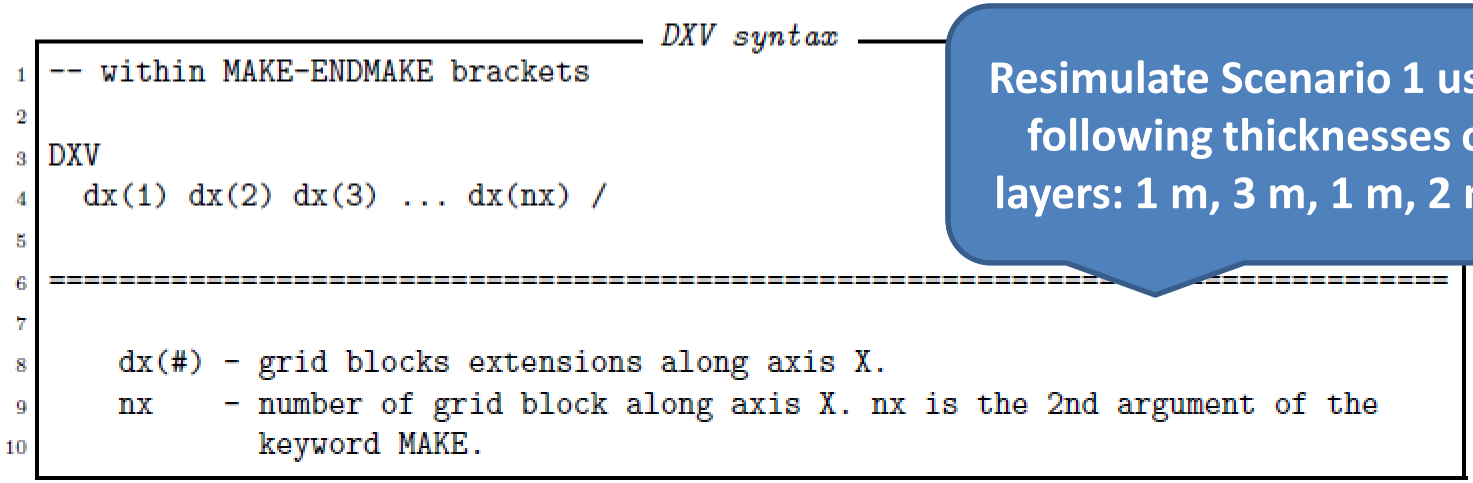
```
-- within MAKE/ENDMAKE brackets.

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)
```

Locate the top boundary of the reservoir at depth 900 meters, whereas the bottom boundary at depth 920 meters.



Cartesian grids

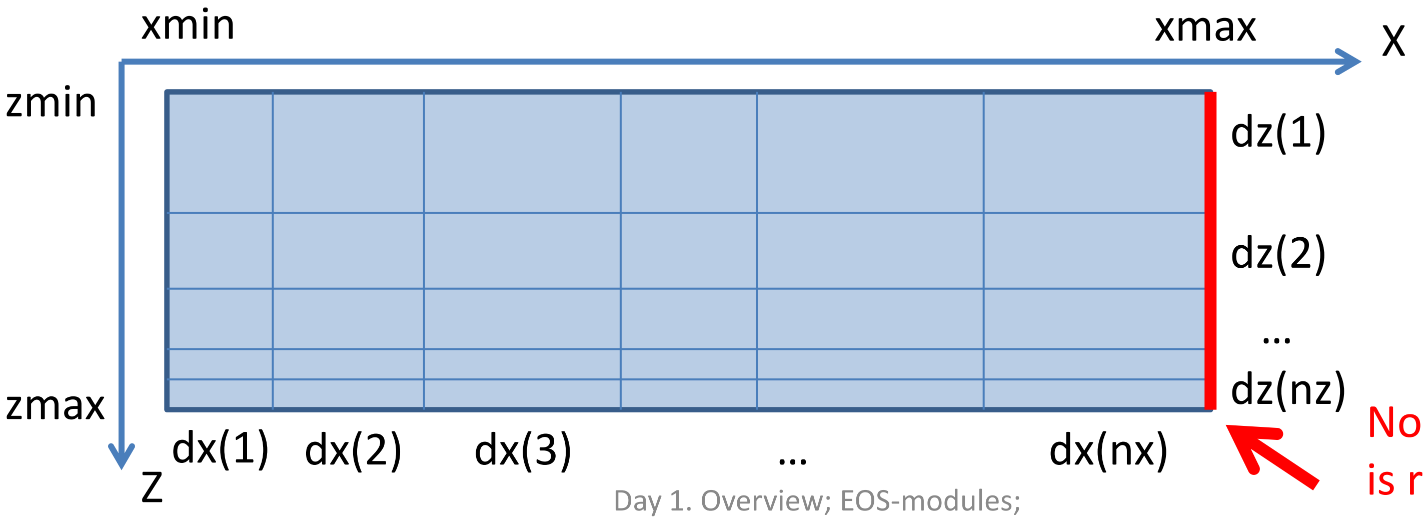
The grid block extensions can be redefined using the **DXV**, **DYV**, **DZV** keywords



DXV syntax

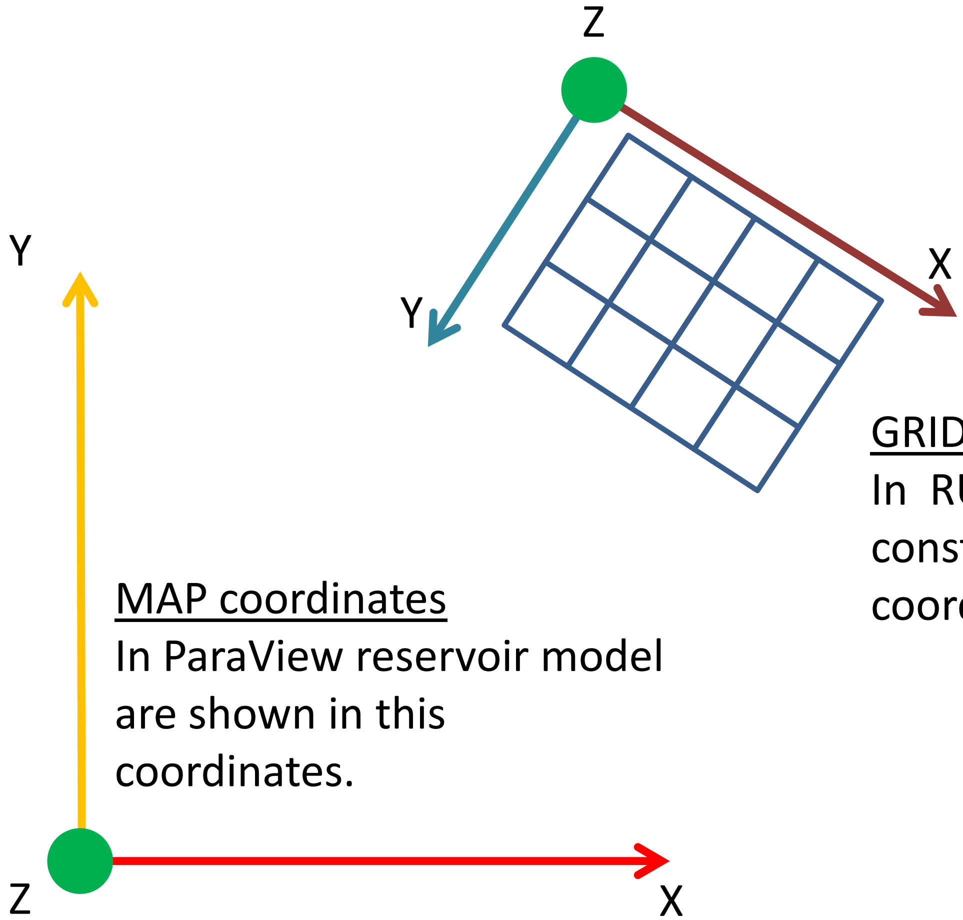
```
-- within MAKE-ENDMAKE brackets

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.
```

Resimulate Scenario 1 using the following thicknesses of the layers: 1 m, 3 m, 1 m, 2 m, 5 m.



Note, this boundary is redefined by the keyword DXV

Coordinate systems



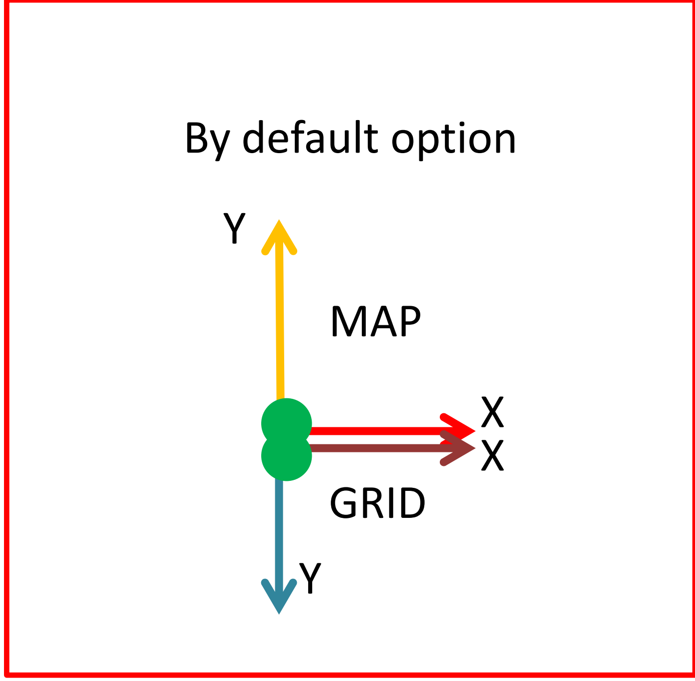
MAP coordinates

In ParaView reservoir model are shown in this coordinates.

GRID coordinates

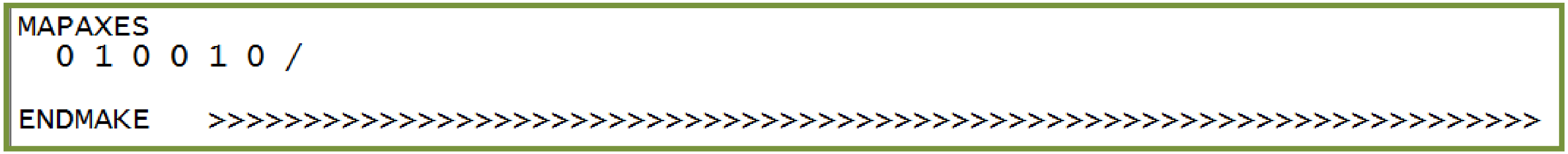
In RUN-file the grid is constructed in this coordinates (e.g. XYZBOUND)

By default option

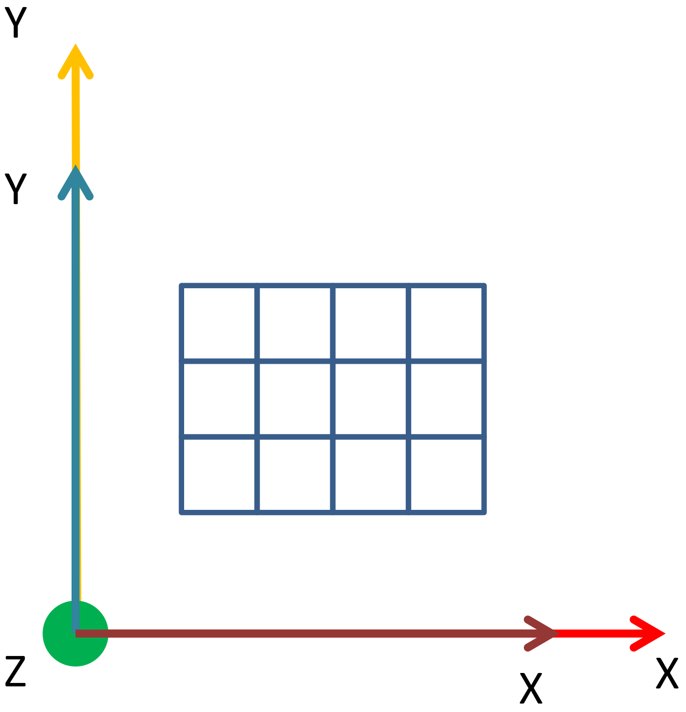


Coordinate systems

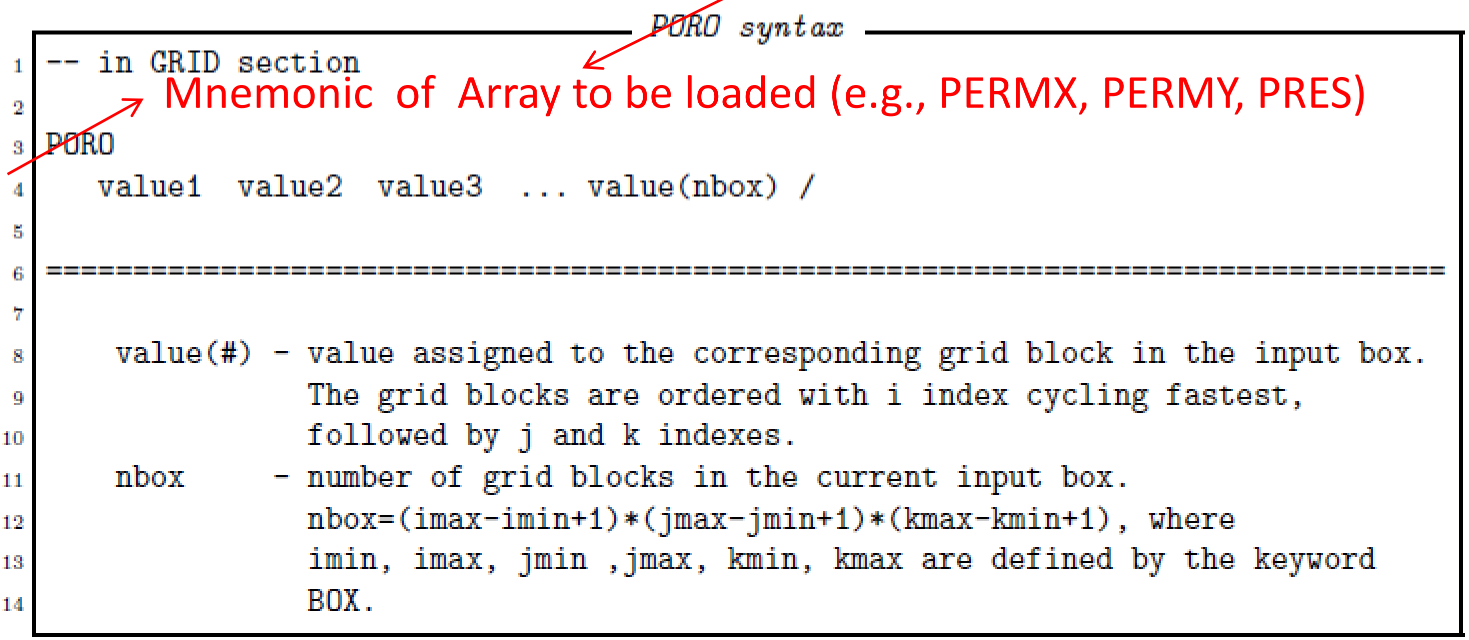
If you want to have the coordinate systems as shown below then type before ENDMAKE:



```
MAPAXES
  0 1 0 0 1 0 /

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



Arrays loading

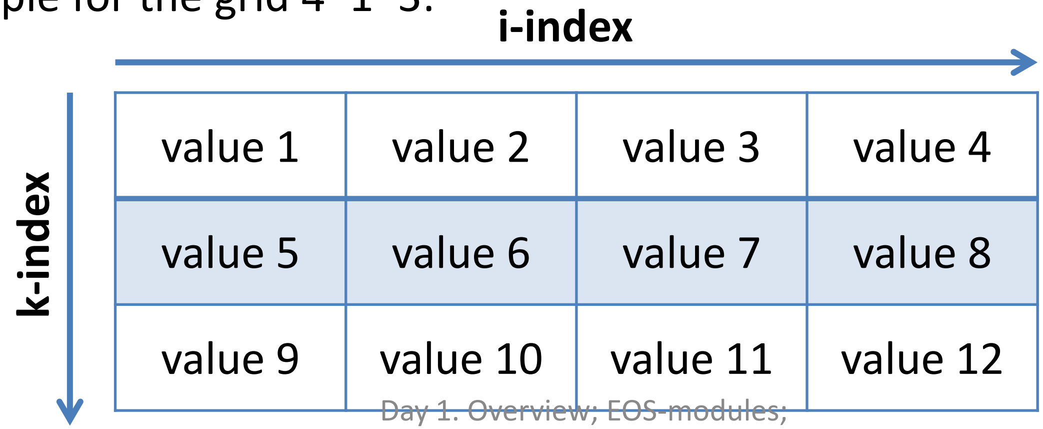


PORO syntax

```
-- in GRID section

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.
```

Mnemonic of Array to be loaded (e.g., PERMX, PERMY, PRES)

An example for the grid 4*1*3:



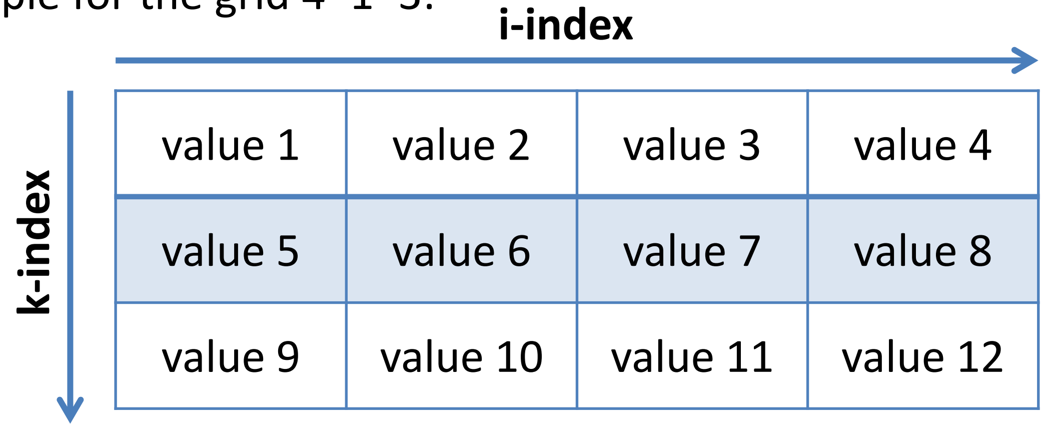
value 1	value 2	value 3	value 4
value 5	value 6	value 7	value 8
value 9	value 10	value 11	value 12

Note:

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

Arrays loading

An example for the grid 4*1*3:



value 1	value 2	value 3	value 4
value 5	value 6	value 7	value 8
value 9	value 10	value 11	value 12

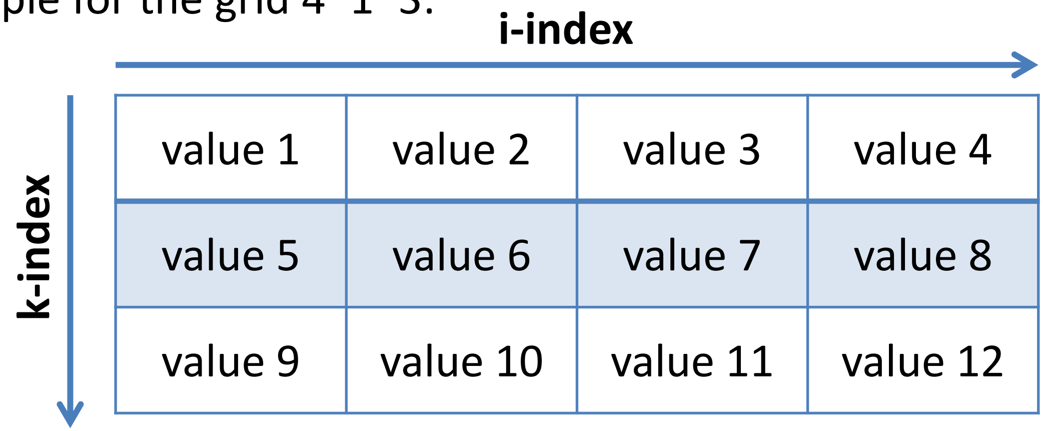
Note:

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

Layer 1: PORO=0.25; Layer 2: PORO=0.2;
Layer 3: PORO=0.15; Layer 4: PORO=0.17;
Layer 5: PORO=0.3.
Specify this distribution of porosity and resimulate Scenario 1.

Arrays loading

An example for the grid 4*1*3:



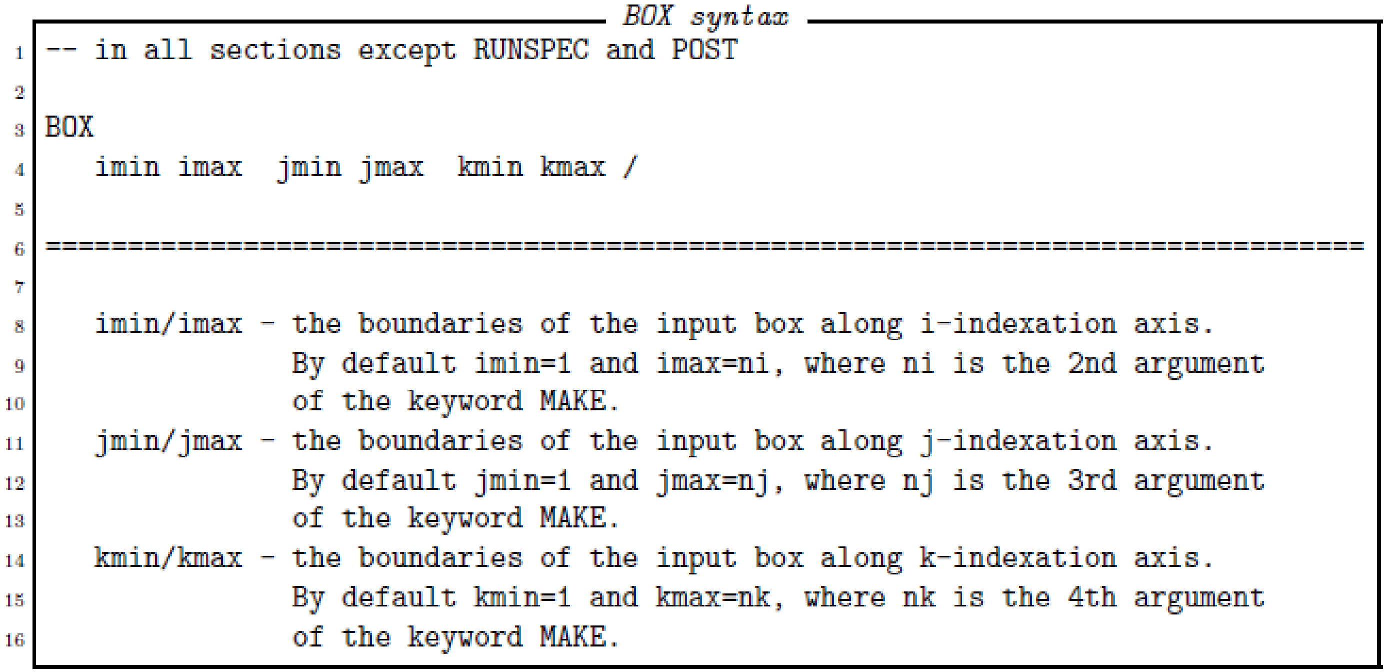
value 1	value 2	value 3	value 4
value 5	value 6	value 7	value 8
value 9	value 10	value 11	value 12

Note:

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

Initial pressure is 100 bar for j=1,...,5 , and 105 bar for j=6,...,10 . Resimulate 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.

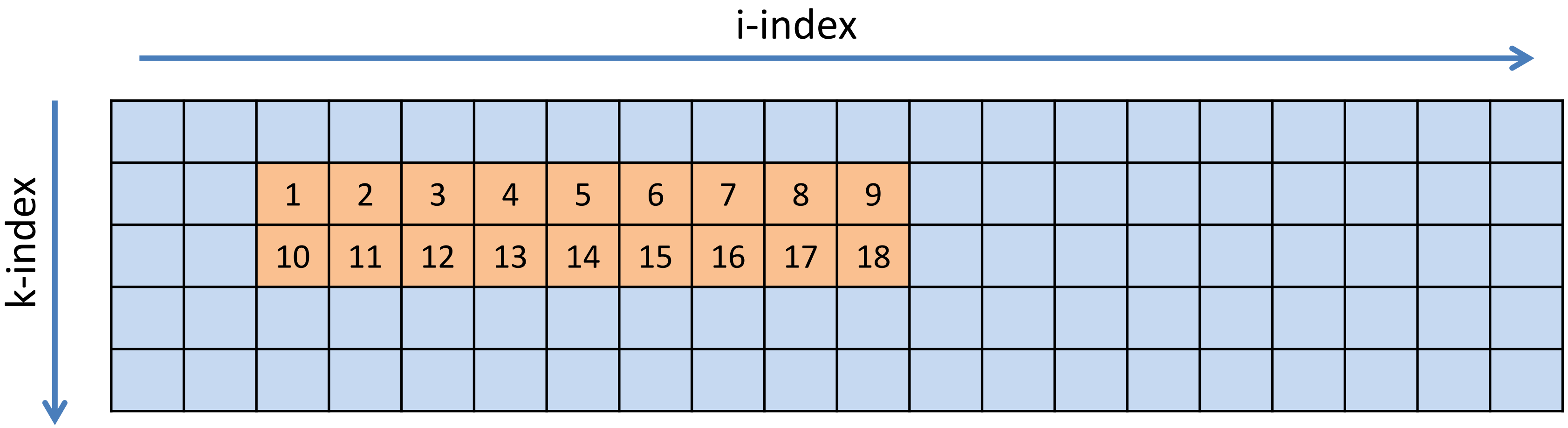
Note that at the beginning of every section the input box is reset by the program.

Keyword BOX

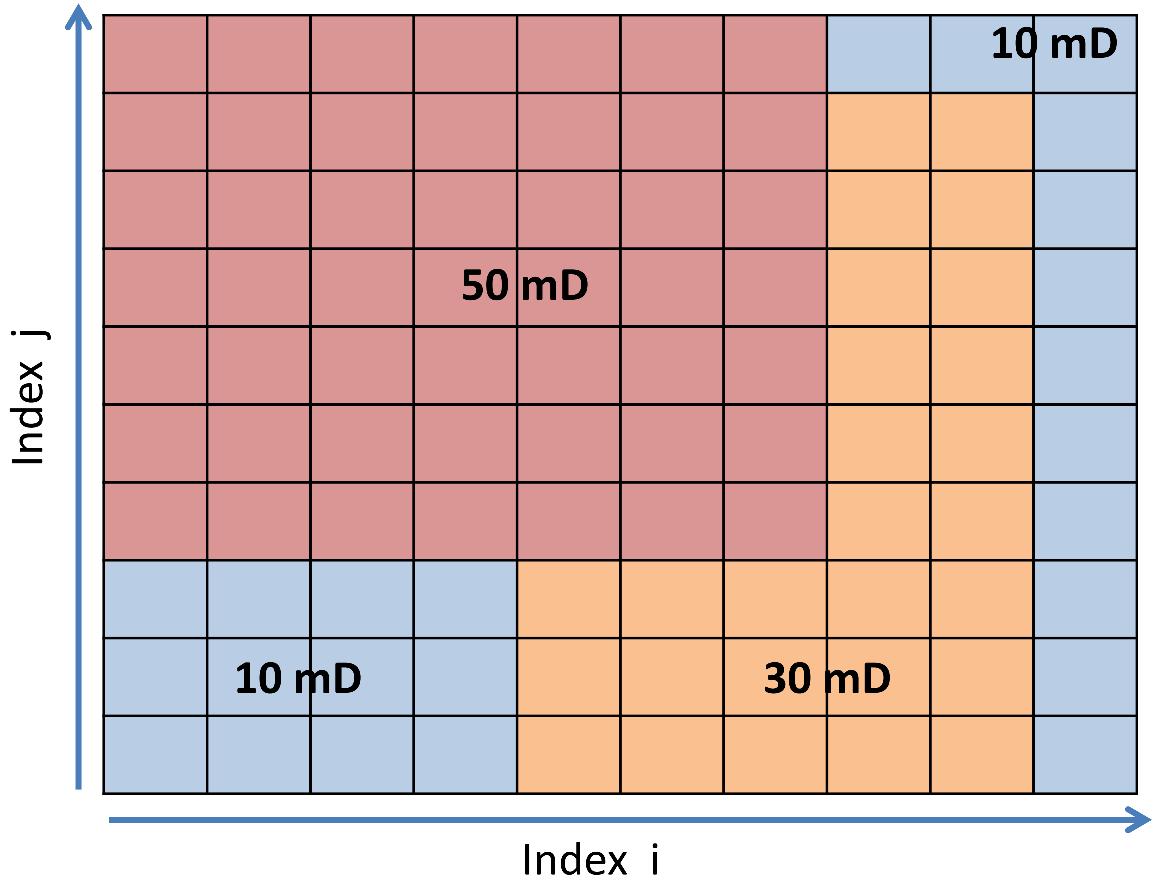
For example, the keyword

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

selects the following region for arrays input



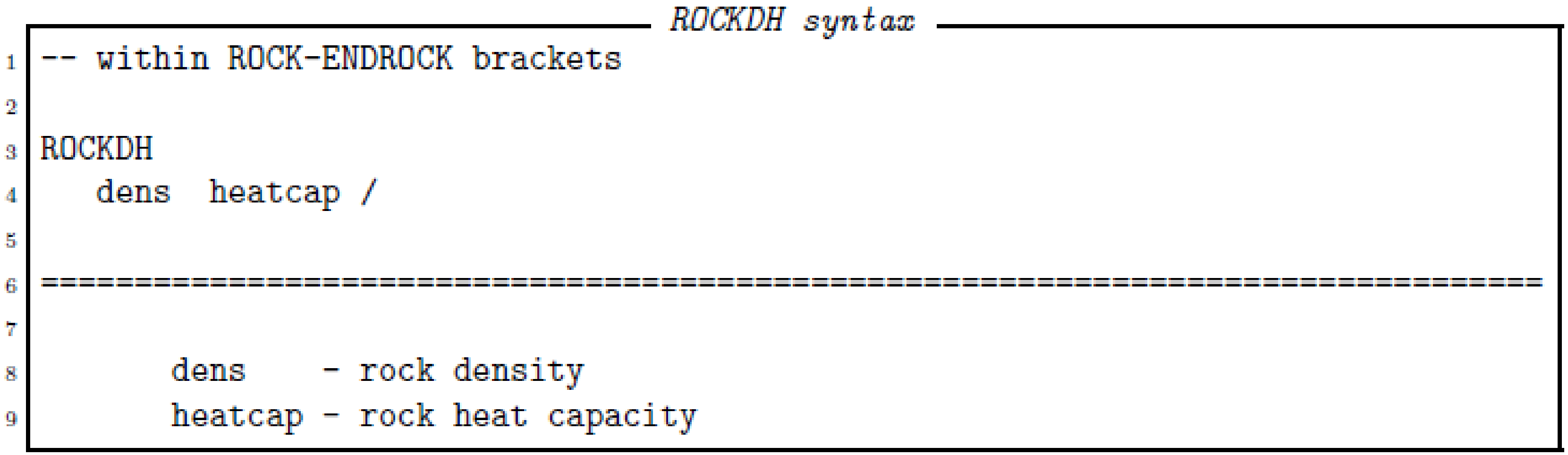
Keyword BOX



Specify this porosity distribution and resimulate 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:

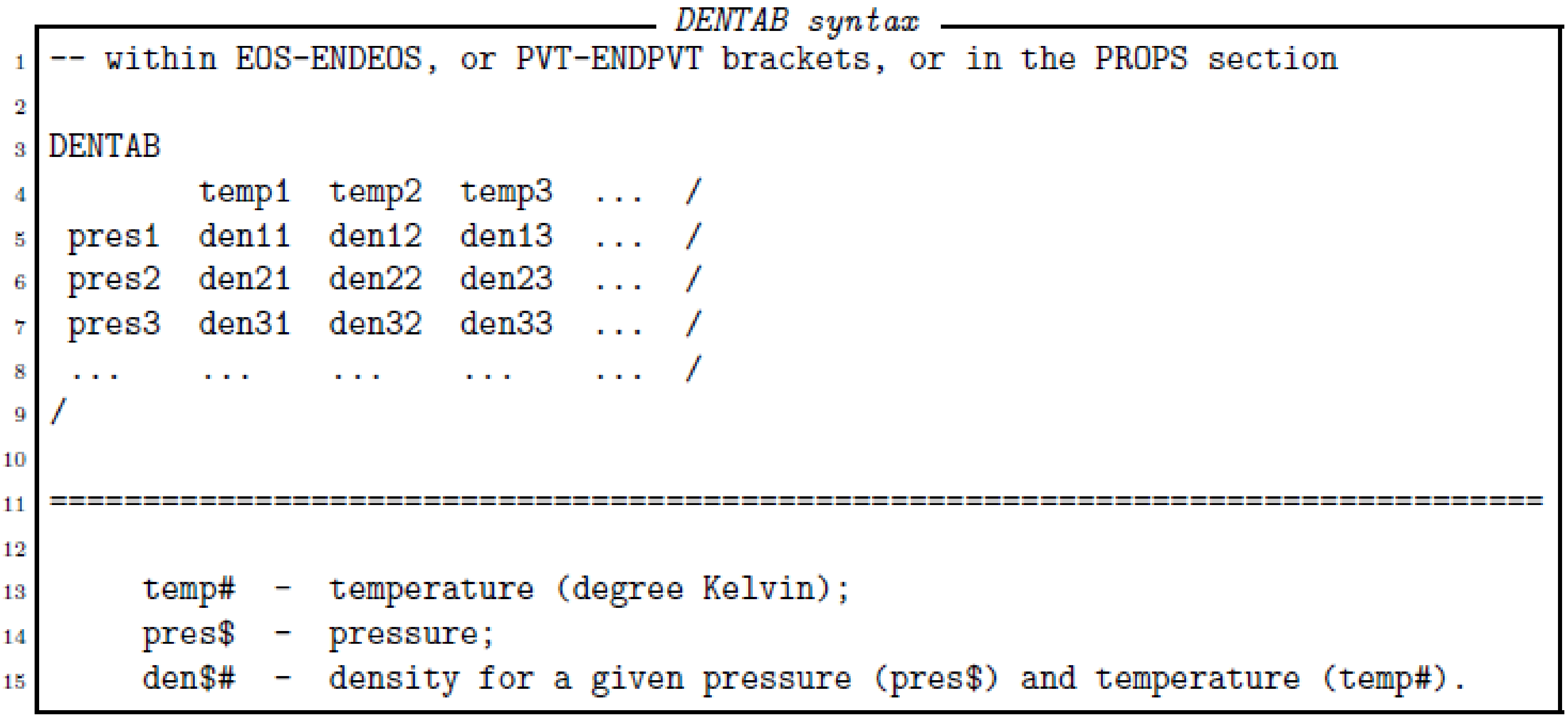


ROCKDH syntax

```
-- within ROCK-ENDROCK brackets

ROCKDH
   dens  heatcap /

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

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

Properties of fluid (DENTAB)

The function $\rho(P,T)$ is specified by the DENTAB keyword.

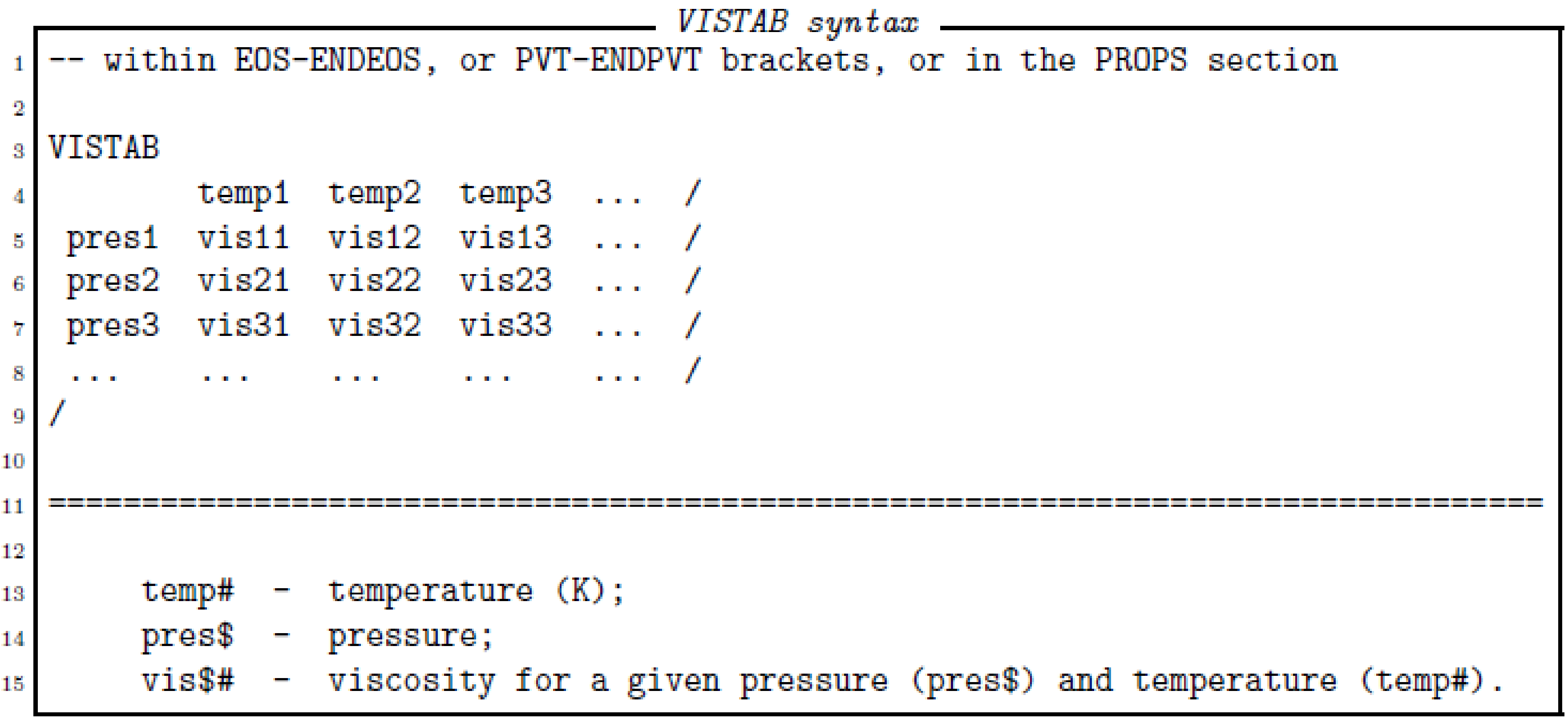


DENTAB syntax

```
-- within EOS-ENDEOS, or PVT-ENDPVT brackets, or in the PROPS section

DENTAB
        temp1  temp2  temp3  ...  /
 pres1  den11  den12  den13  ...  /
 pres2  den21  den22  den23  ...  /
 pres3  den31  den32  den33  ...  /
 ...    ...    ...    ...    ...  /
/

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

     temp#  -  temperature (degree Kelvin);
     pres$  -  pressure;
     den$#  -  density for a given pressure (pres$) and temperature (temp#).
```

Properties of fluid (VISTAB)

The function $\mu(P,T)$ is specified by the VISTAB keyword.

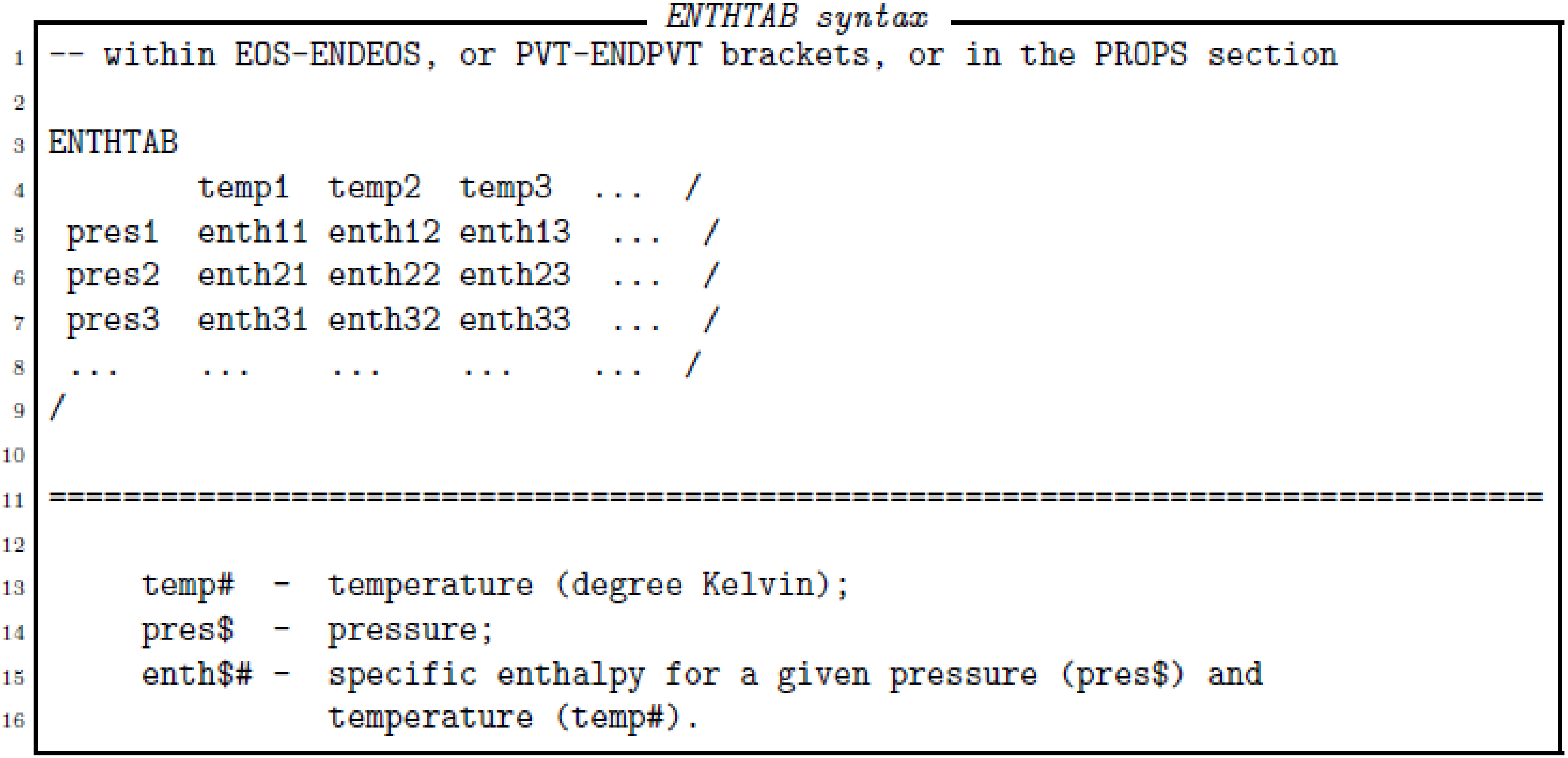


VISTAB syntax

```
-- within EOS-ENDEOS, or PVT-ENDPVT brackets, or in the PROPS section

VISTAB
        temp1  temp2  temp3  ...  /
 pres1  vis11  vis12  vis13  ...  /
 pres2  vis21  vis22  vis23  ...  /
 pres3  vis31  vis32  vis33  ...  /
 ...    ...    ...    ...    ...  /
/

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

     temp#  -  temperature (K);
     pres$  -  pressure;
     vis$#  -  viscosity for a given pressure (pres$) and temperature (temp#).
```

Properties of fluid (ENTHTAB)

The function $h(P,T)$ is specified by the ENTHTAB keyword.



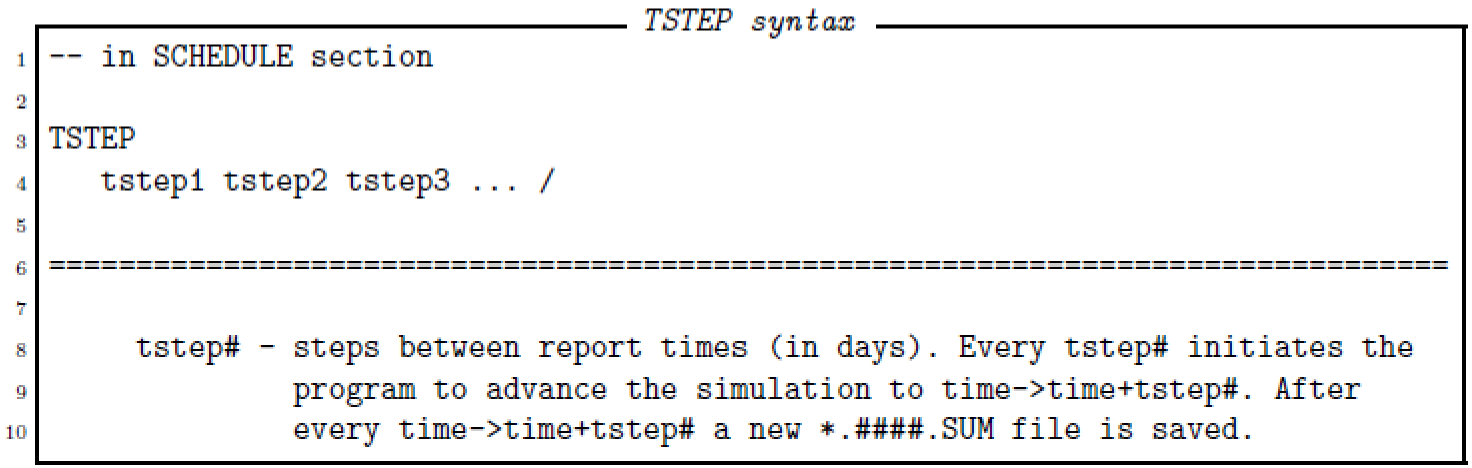
ENTHTAB syntax

```
-- within EOS-ENDEOS, or PVT-ENDPVT brackets, or in the PROPS section

ENTHTAB
        temp1  temp2  temp3   ...  /
 pres1  enth11 enth12 enth13   ...  /
 pres2  enth21 enth22 enth23   ...  /
 pres3  enth31 enth32 enth33   ...  /
 ...    ...    ...    ...     ...  /
/

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

     temp#  -  temperature (degree Kelvin);
     pres$  -  pressure;
     enth$# -  specific enthalpy for a given pressure (pres$) and
               temperature (temp#).
```

Report times

Note that time units 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 advances the simulation further in time and produces the required outputs.

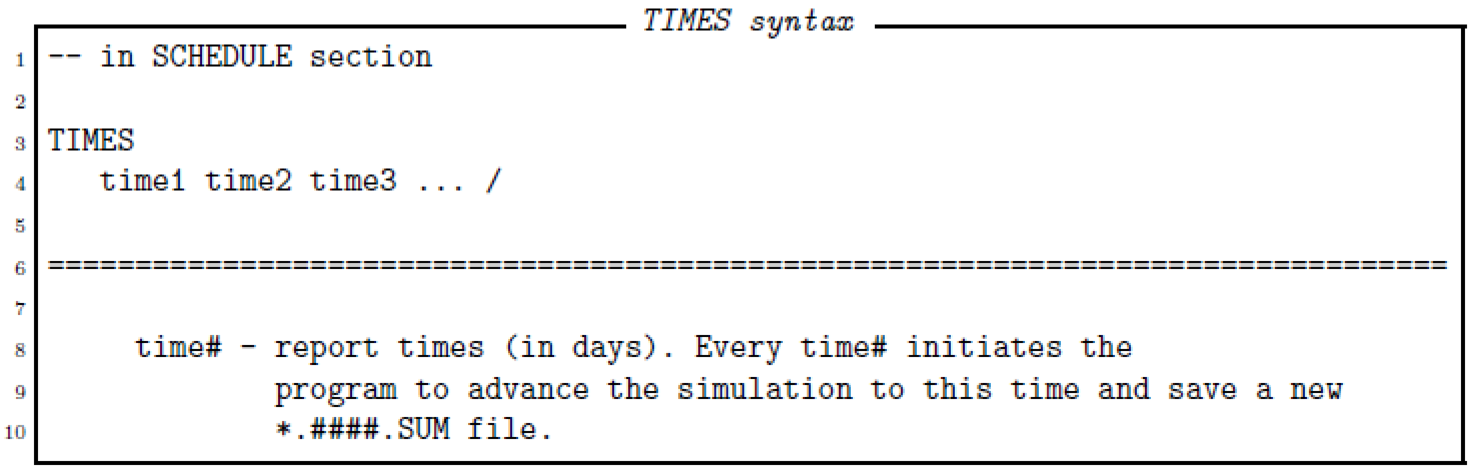


TSTEP syntax

```
-- 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

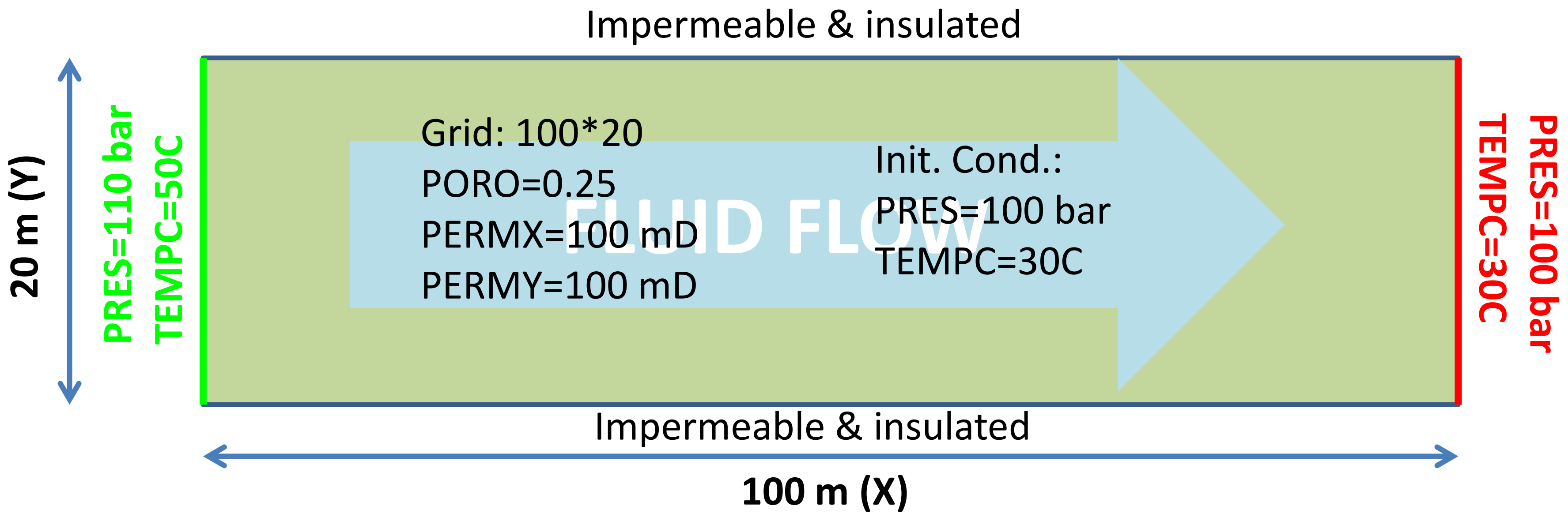
```
-- in SCHEDULE section

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: 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.

Scenario 2

Scenario 2 (a 2D problem)



Grid: 100*20
PORO=0.25
PERMX=100 mD
PERMY=100 mD

Init. Cond.:
PRES=100 bar
TEMPC=30C

Simulate the flow over 200 days reporting every 20 days.

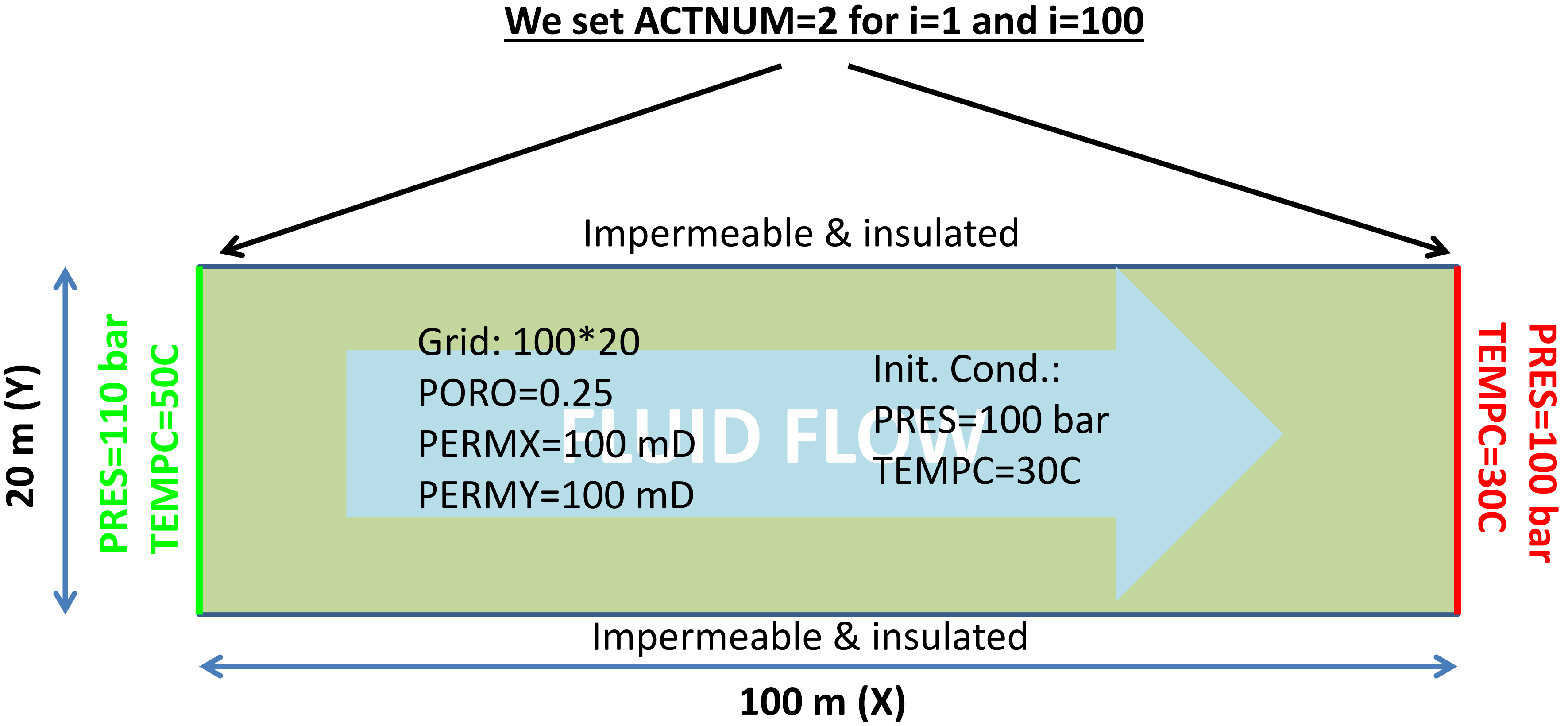
ACTNUM flag

<u>Possible value of ACTNUM:</u>

0 – cell inactive;

1 – cell active (default);

2 – fixed parameters.

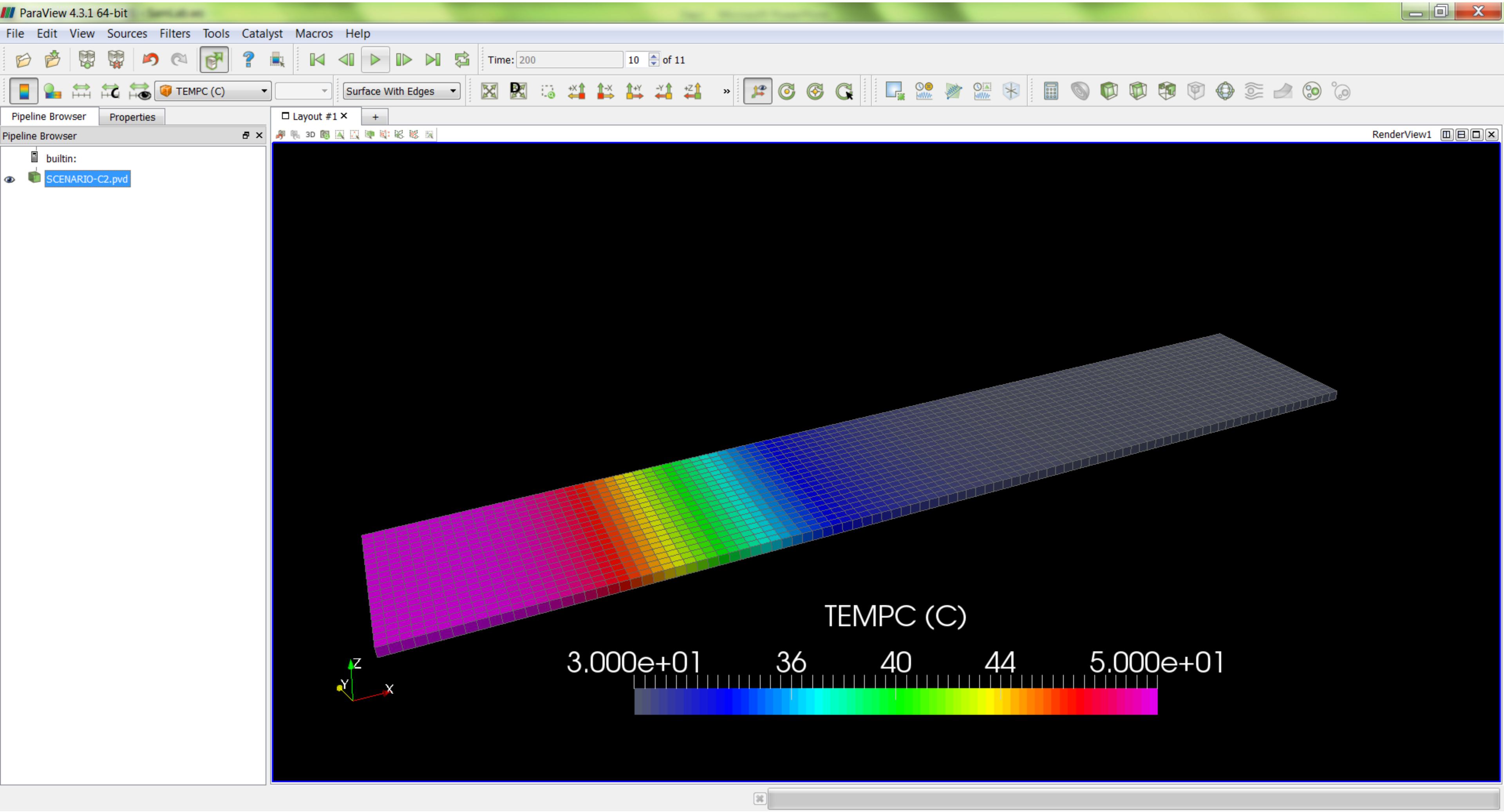


We set ACTNUM=2 for i=1 and i=100

RUN-file (Scenario 2)

Open RUN-file (SCENARIO-C2.RUN) in a text editor

Scenario 2 (result)



Heat conduction

Including physical phenomena in simulation

Keyword in RUNSPEC section	Description
CAPPRES	Enables capillary pressure
ROCKCOMP	Enables compressibility of rocks
HCROCK	Enables heat conduction through rocks
HCFLUID	Enables heat conduction through fluid
NOGRAV	Disables gravity
ISOTHERM	Simulate at constant temperature

Heat conduction option

Heat conduction coefficient:

$$\lambda = \underbrace{\phi \sum_i s_i \lambda_i}_{\textbf{HCFLUID}} + \underbrace{1 - \phi \;\; \lambda_r}_{\textbf{HCROCK}}$$

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

Scenario 2; exercise

Resimulate scenario 2 with heat conductivity of rocks 2 W/m/K .

Open RUN-file (SCENARIO-C2.RUN)
in a text editor

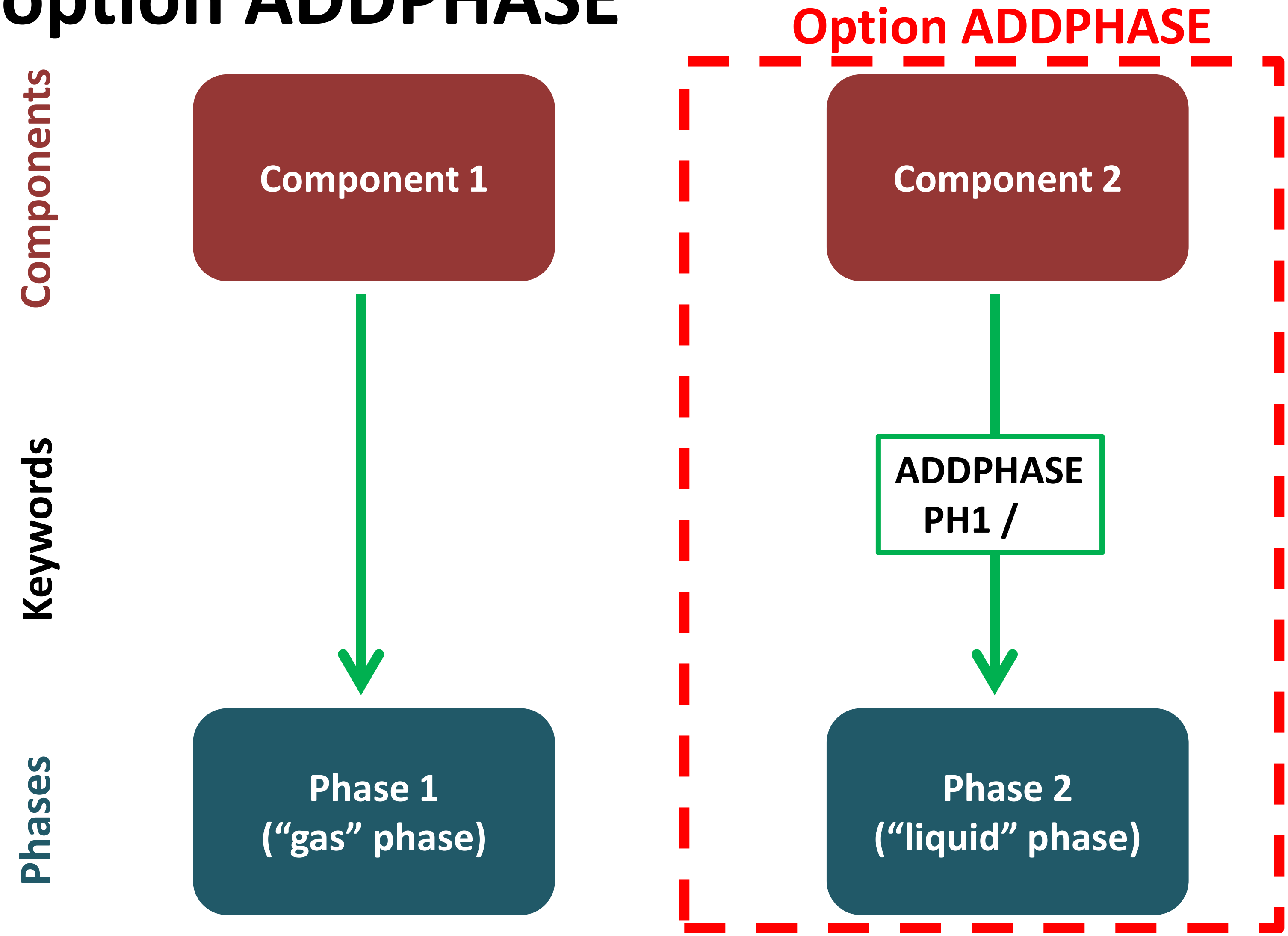
Options vs. EOS modules

	SIMPLMOD	T2EOS1	BINMIXT	BLACKOIL	GASSTORE
CAPPRES	Yes	No	No	Yes	Yes
HCROCK	Yes	Yes	Yes	No	Yes
HCFLUID	No	Yes	No	No	No
ISOTHERM	Yes	Yes	No	No	Yes
ADDPHASE	Yes	Not tested	Yes	Not tested	Not tested
EQL-ENDEQL (initial equilibration)	No	No	No	Yes	Yes
EOSNUM regions	Yes	No	No	Yes	No

Buckley-Leverett problem (modelling two-phase immiscible displacement)

EOS module SIMPLMOD with option ADDPHASE

T≠const



Mathematical model

i=1 – gas phase (CO_2)
i=2 – liquid phase (H_2O)

$$\frac{\partial \phi s_i \rho_i}{\partial t} + \mathrm{div}(\rho_i \mathbf{w}_i) = 0, \quad i = 1,2$$

- mass balance equation

$$\frac{\partial}{\partial t}\left(\phi \sum_{i=1,2} s_i \rho_i e_i + (1-\phi)\rho_r e_r \right) +$$

$$+ \mathrm{div}\left(\sum_{i=1,2} \rho_i h_i \mathbf{w}_i \right) = \mathrm{div}(\lambda \mathbf{grad} T)$$

- energy balance equation

$$\mathbf{w}_i = -K \frac{K_{ri}(s)}{\mu_i} \left(\mathbf{grad} P_i - \rho_i \mathbf{g} \right); \quad P_1 - P_2 = P_c(s)$$

- Darcy & cap. Pres.

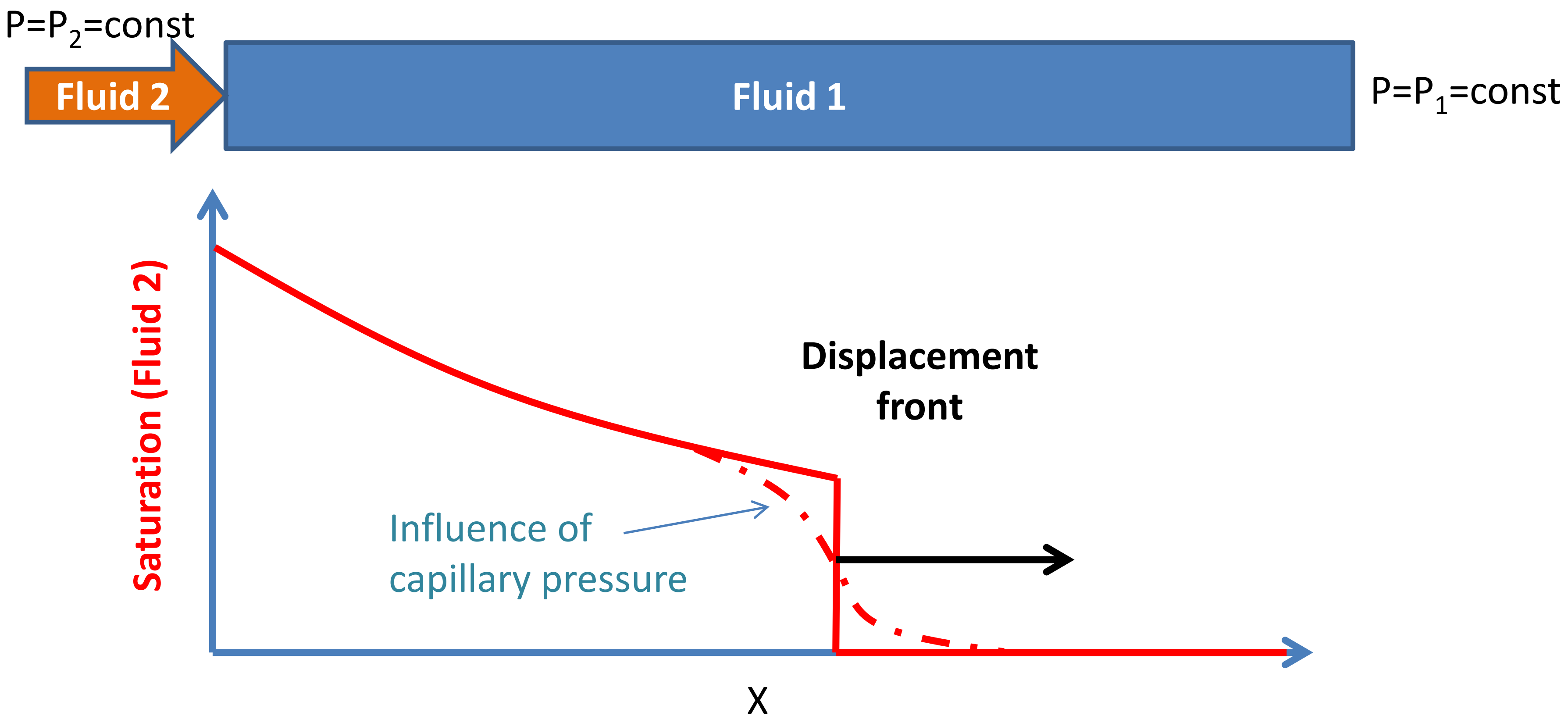
$$\rho_i(P_i, T); \quad h_i(P_i, T); \quad \mu_i(P_i, T)$$

$$h_i = e_i + \frac{P_i}{\rho_i}$$

- EoS

Some words about the fractional-flow theory

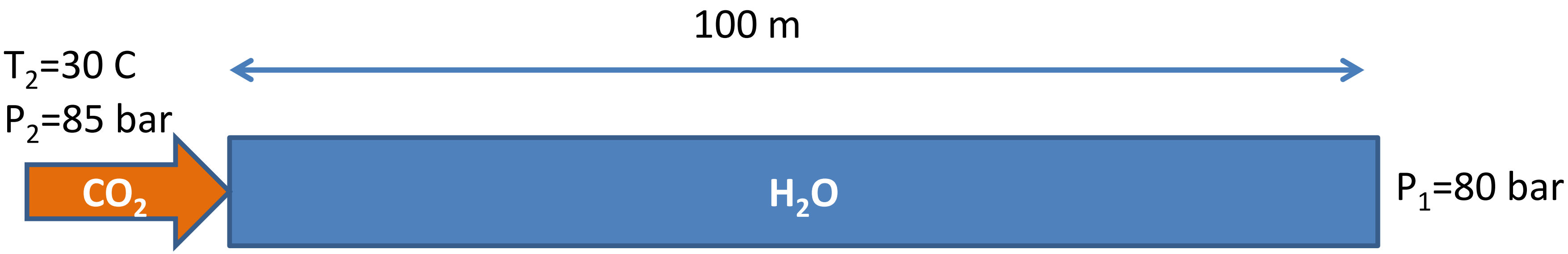
For isothermal flow



Scenario 3 (a 1D problem)

Saturation functions

See RUN-file



T_2=30 C

P_2=85 bar

100 m

CO_2

H_2O

P_1=80 bar

Rock

PORO=0.25

PERMX=100 mD

HCONDCFX=2 W/m/K

Initial conditions

PRES=80 bar

TEMPC=47 C

SAT%H2O=1

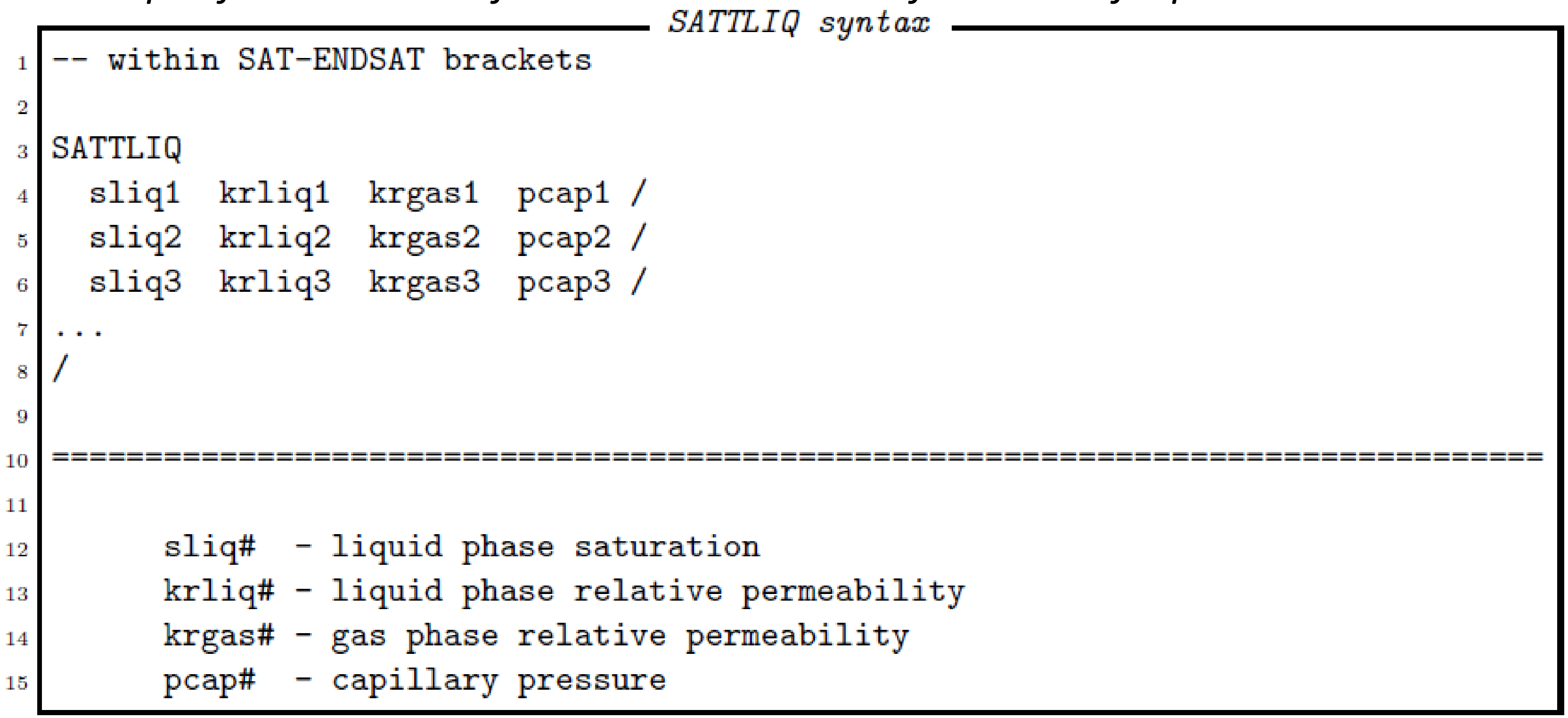
Scenario 3

1. Open RUN-file in text editor
2. Run the simulation
3. Open LOG-file to see results

Relative permeability

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

Specifies saturation functions as tabulated functions of liquid saturation



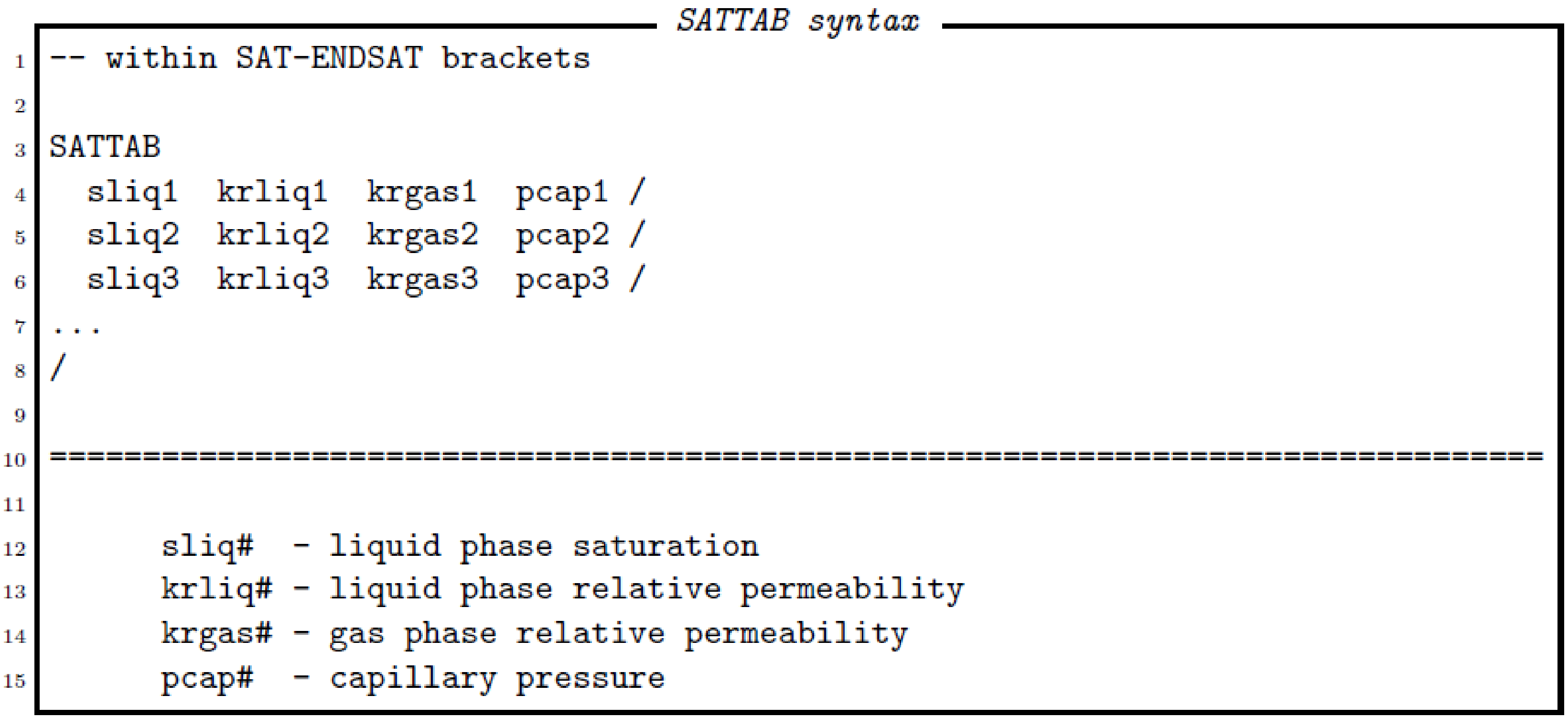
SATTLIQ syntax

```
-- within SAT-ENDSAT brackets

SATTLIQ
  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 permeability

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

Specifies saturation functions as tabulated functions of liquid saturation



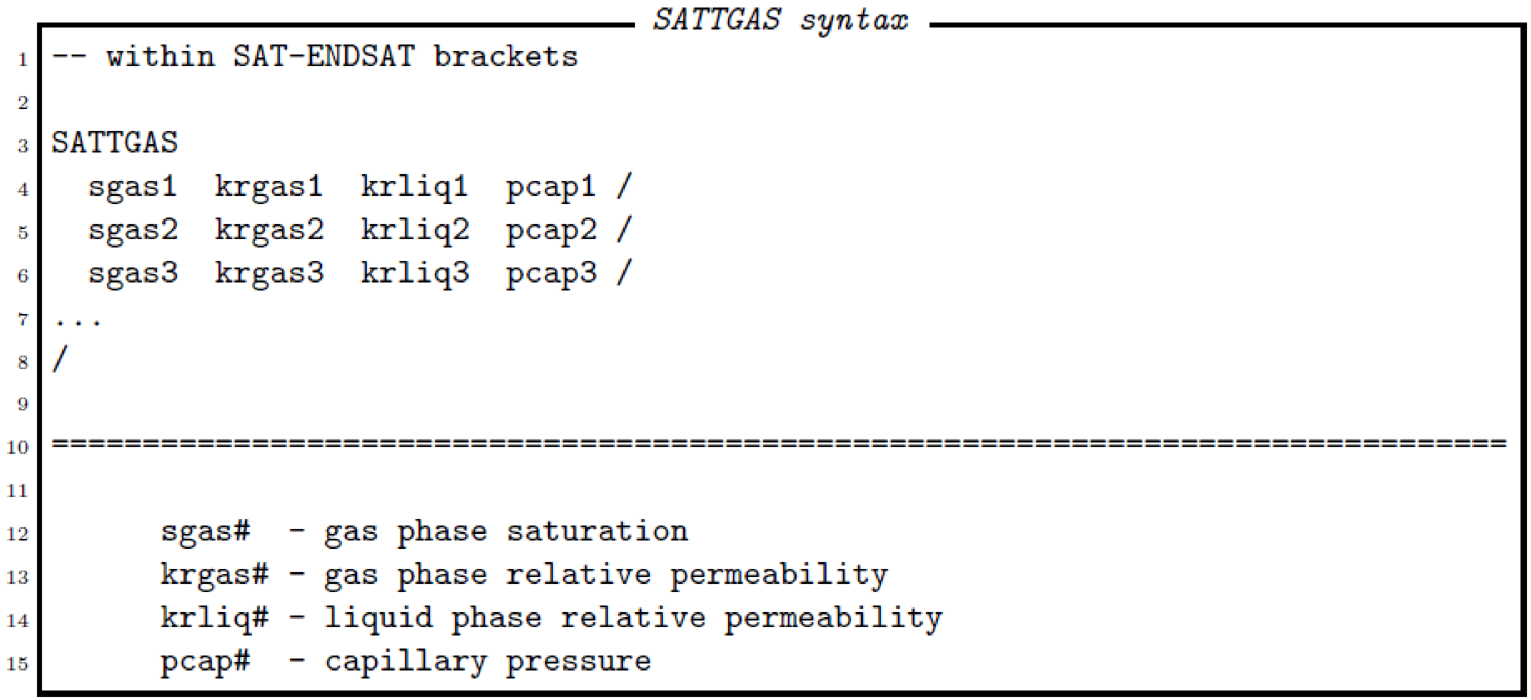
SATTAB syntax

```
-- 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 permeability

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

Specifies saturation functions as tabulated functions of gas saturation



SATTGAS syntax

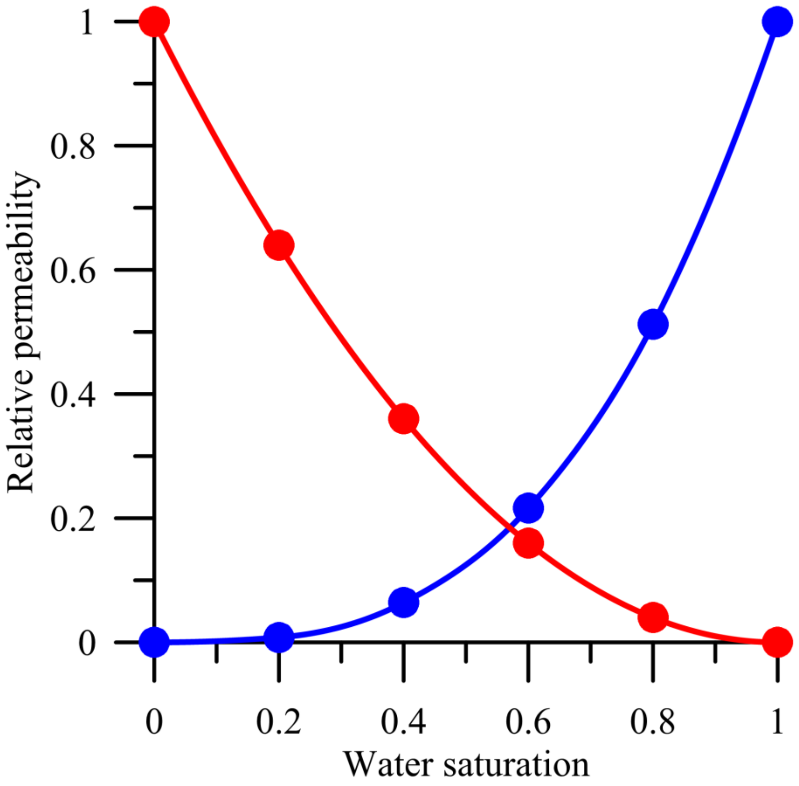
```
-- within SAT-ENDSAT brackets

SATTGAS
  sgas1  krgas1  krliq1  pcap1 /
  sgas2  krgas2  krliq2  pcap2 /
  sgas3  krgas3  krliq3  pcap3 /
...
/

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


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

Relative permeability (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    /
/
```

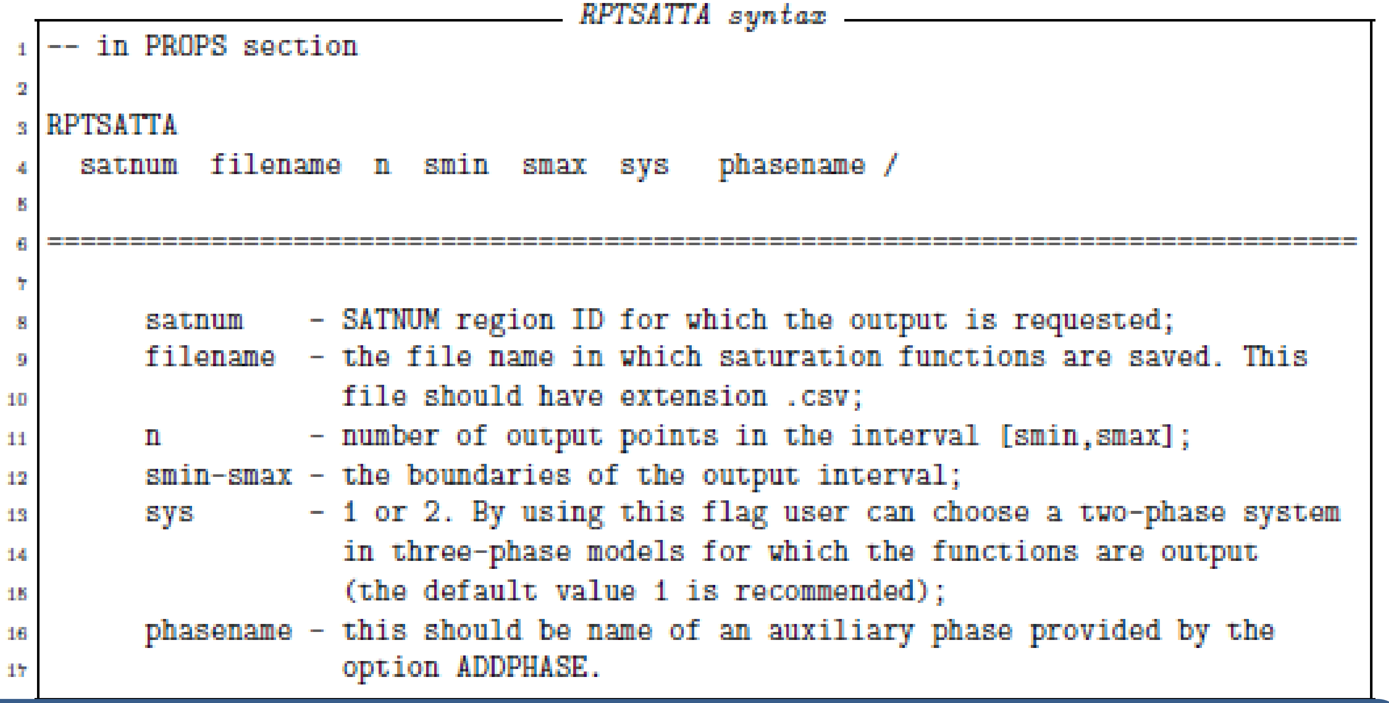
$$k_{r,wat}(s_{wat}) = s_{wat}^3$$

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



Relative permeability in ParaView

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

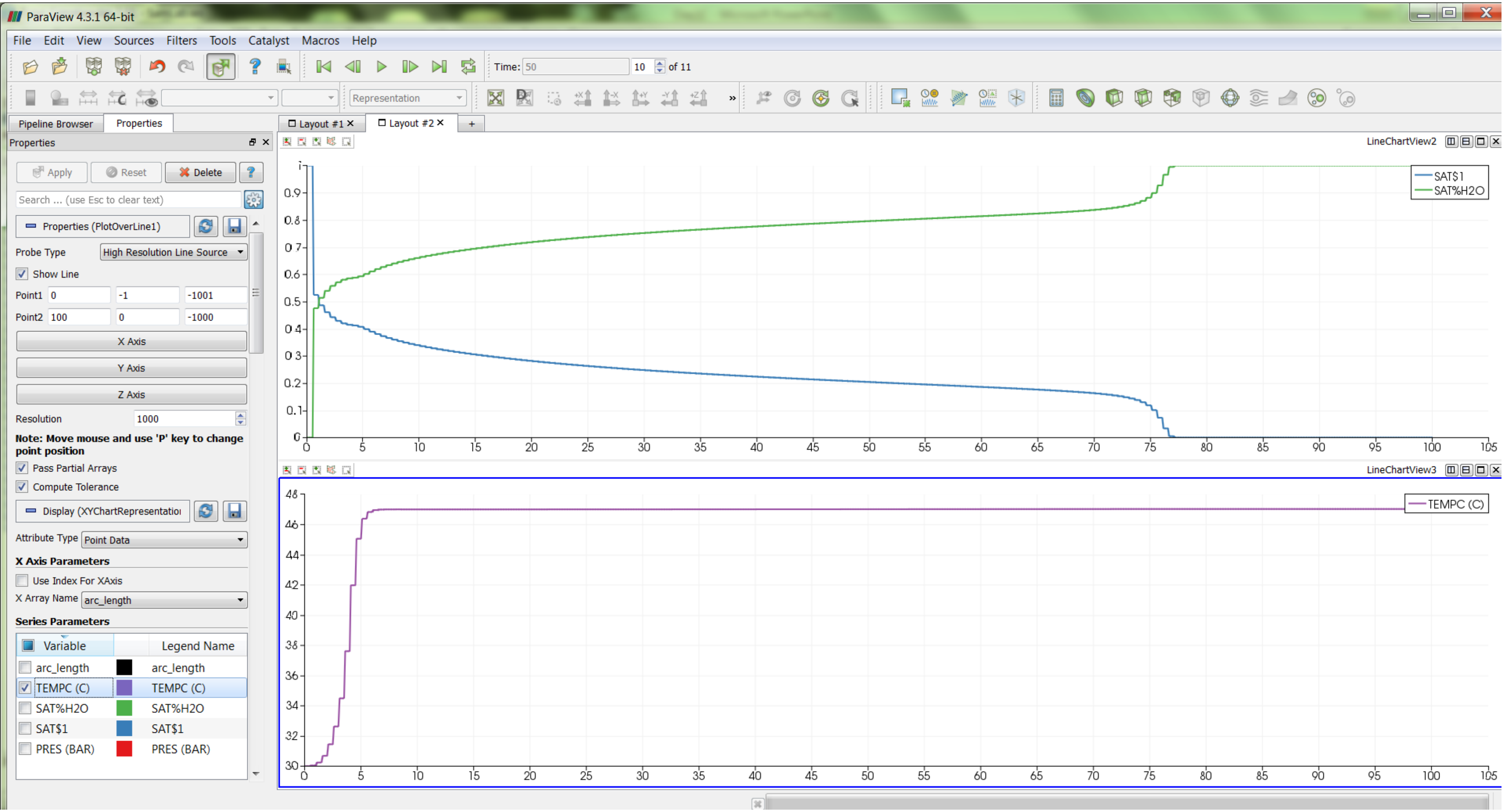


RPTSATTA syntax

```
-- in PROPS section

RPTSATTA
  satnum  filename  n  smin  smax  sys   phasename /

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

      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;
      sys       - 1 or 2. By using this flag user can choose a two-phase system
                  in three-phase models for which the functions are output
                  (the default value 1 is recommended);
      phasename - this should be name of an auxiliary phase provided by the
                  option ADDPHASE.
```

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

Result



Properties (PlotOverLine1)

Probe Type: High Resolution Line Source

☑ Show Line

Point1	0	-1	-1001
Point2	100	0	-1000

Resolution: 1000

Note: Move mouse and use 'P' key to change point position

☑ Pass Partial Arrays

☑ Compute Tolerance

Display (XYChartRepresentatio

Attribute Type: Point Data

X Axis Parameters

☐ Use Index For XAxis

X Array Name: arc_length

Series Parameters

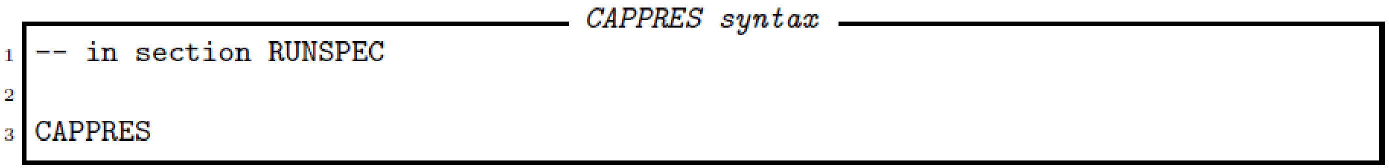
Variable		Legend Name
☐ arc_length	■	arc_length
☑ TEMPC (C)	🟪	TEMPC (C)
☐ SAT%H2O	🟩	SAT%H2O
☐ SAT$1	🟦	SAT$1
☐ PRES (BAR)	🟥	PRES (BAR)

Including physical phenomena in simulation

Keyword in RUNSPEC section	Description
CAPPRES	Enables capillary pressure
ROCKCOMP	Enables compressibility of rocks
HCROCK	Enables heat conduction through rocks
HCFLUID	Enables heat conduction through fluid
NOGRAV	Disables gravity
ISOTHERM	Simulate at constant temperature

Scenario 3; Exercise 1

Resimulate scenario 3 enabling calculation of capillary pressure.

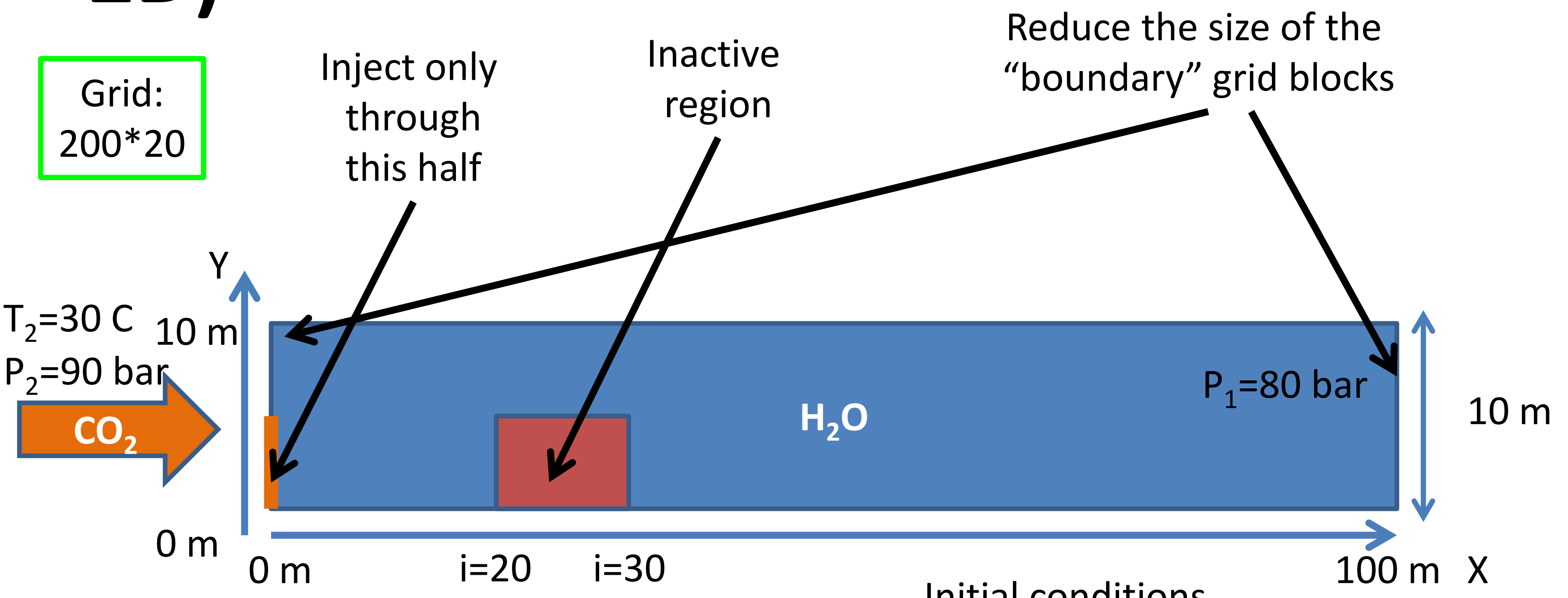


CAPPRES syntax

```
-- in section RUNSPEC

CAPPRES
```

Scenario 3 (Exercise 2; extension to 2D)

Grid: 200*20



<u>Rock</u>

PORO=0.25
PERMX=100 mD
PERMY=100 mD
HCONDCFX=2 W/m/K
HCONDCFY=2 W/m/K

<u>Initial conditions</u>

PRES=80 bar
TEMPC=47 C
SAT%H2O=1

Scenario 3; exercise 2

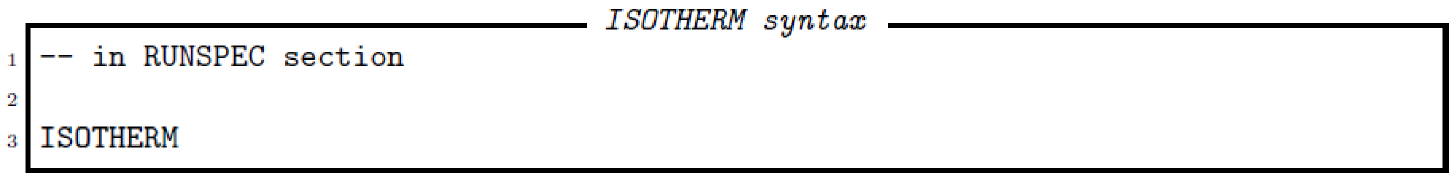
1. Open RUN-file in text editor
2. Run the simulation
3. Open LOG-file to see results

Including physical phenomena in simulation

Keyword in RUNSPEC section	Description
CAPPRES	Enables capillary pressure
ROCKCOMP	Enables compressibility of rocks
HCROCK	Enables heat conduction through rocks
HCFLUID	Enables heat conduction through fluid
NOGRAV	Disables gravity
ISOTHERM	Simulate at constant temperature

Scenario 3; Exercise 3

Resimulate scenario 3 in isothermal mode



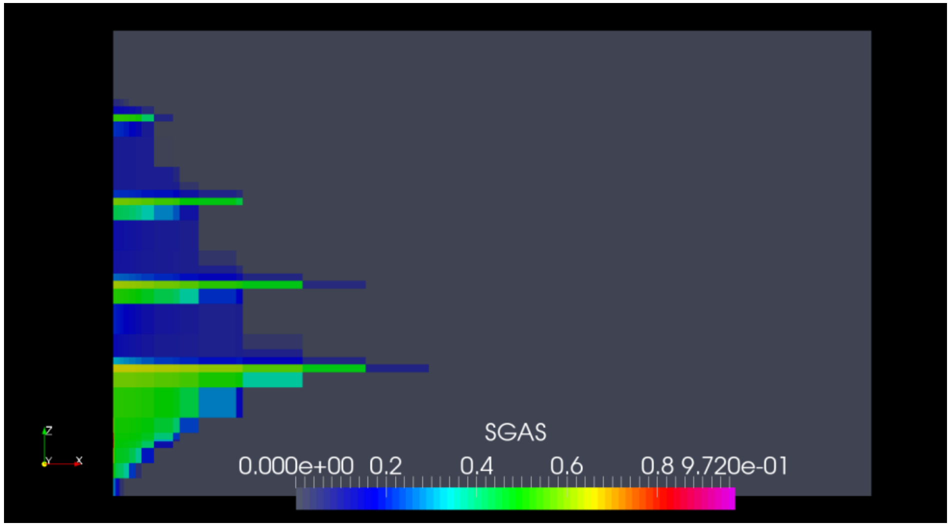
ISOTHERM syntax

```
-- in RUNSPEC section

ISOTHERM
```

Next day

- EOS module GASSTORE
- Operations on arrays
- Regions
- Boundary conditions
- Fluid-in-place regions
- Radial grids
- POST section
- Point sources

Vertical brine displacement by CO_2


SGAS
0.000e+00
9.720e-01
Z
Y
X