NORCE Norwegian Research Centre AS
www.norceresearch.no

User's manual for

IORCoreSim - combined EOR and SCAL simulator

(Version 1.325)

Arild Lohne

NORCE -20. May 2021

Project number:	
Project title:	
Client(s):	National IOR Centre of Norway
Research program:	
Distribution restriction:	

1 Résumé (IORCoreSim)

The IORCoreSim simulator builds on the previous MEOR simulator Bugsim reported in IRIS - 2013/015 dated 14.03.2013. The résumé from that report is included below.

2 Résumé (Bugsim)

This is the second version of this User's manual for the MEOR simulator *Bugsim*. The previous report (IRIS – 2008/84) is updated with new options included in the simulator the last year (2012) within the project *MEOR phase II*.

The *Bugsim* simulator has been built for the purpose of investigating possible MEOR mechanisms at laboratory- and in small field-scale models. The program is a three-dimensional, rectangular or radial grid model that handles multi-component flow with up to three phases (water, oil and gas). The present version has a flexible model for biomass growth. Modelled MEOR effects comprise permeability reduction and effects of metabolites as gas and polymer.

The main improvements in this last period include options for modelling effects of biofilms or biosurfactants on oil recovery, and temperature calculations.

Thanks to contributors. The authors acknowledge Wintershall, ConocoPhillips and the Ekofisk Co-venturers, including TOTAL, ENI, StatoilHydro and Petoro, for financing this second phase through the research centre COREC.

The first phase was financed by the Research Council of Norway, ConocoPhillips and the Ekofisk Co-venturers through the research centre COREC. Thanks to Aksel Hiorth (IRIS) for useful discussions.

Contents

Summary

This is the summary of the first version 30. May 2008. Description of changes in later versions can be found in the Introduction chapter.

A prototype MEOR simulator *Bugsim* has been developed for the purpose of investigating of possible MEOR mechanisms at the laboratory scale and in smaller field scale models. The program is basically a 3D, black oil, three-phase simulator (water, oil and a simplified representation of gas), extended to handle multiple components mainly dissolved in the water phase. Simulations on Cartesian, centre point (in an approximate way) and radial grid can be performed.

Bugsim has a flexible model for biomass growth based on the standard Monod equation, but is also extended to allow for an arbitrary number of bacteria types, substrates and nutrient groups. Competitive utilisation of different substrates and different nutrients within each group can be modelled. Various microbial products can be specified, each associated with one single substrate source consumed by one specific bacteria type or with several such sources.

A limited number of possible MEOR effects are included in the present version. A permeability reduction is estimated based on the amount of adsorbed biomass. The remaining mechanisms are related to microbial products, in particular produced gas and polymer. Produced gas may swell the oil and reduce its viscosity, while polymer may increase the water viscosity and thereby improve the displacement sweep.

An example of simulated results obtained from *Bugsim* is shown in Figure 1. The figure shows computed distribution of bio-mass in a vertical cross-section model after a period of external injection of food and nutrients.

This report is essential a user’s manual for *Bugsim*. It is best read online on Acrobat pdf-format, which is furnished with hyperlinks to make navigation easy. The input to the program is keyword driven and is read from a text file.

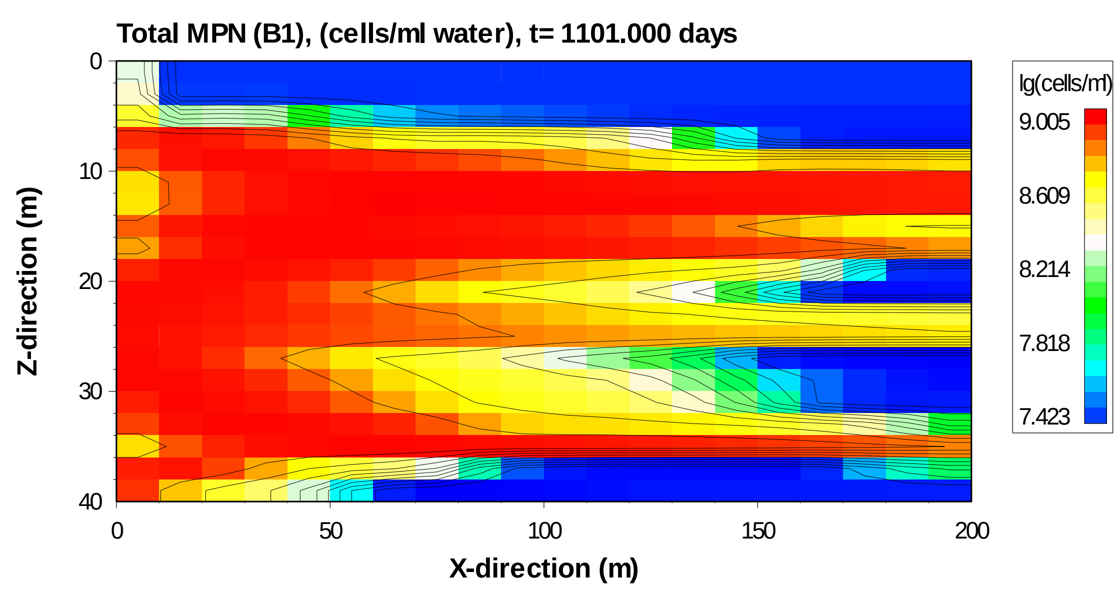


Figure 1 Total cell number distribution after injection of external food. Flow: left-to right.

List of symbols

A	adsorption [pv fraction]
b	Langmuir adsorption parameter [volume fraction]
B_f	volume factor [volume at P_{ref}/volume at surface condition]
BHP	Bottom hole pressure [bar^{-1}]
B_p	Viscosity pressure parameter [bar^{-1}]
B_T	temperature parameter [°K]
B_f	volume factor [volume at P_{ref}/volume at surface or standard condition]
c	concentration, [volume fraction] or as specified in text
E_h	heat energy [J]
f	fractional flow [volume fraction]
F_N	specific nutrient growth factor
F_S	specific substrate growth factor
h_c	specific heat capacity [J/(g·°C)]
J	Leverett *J*-function
J	joule (energy unit) [$\text{kg}\cdot\text{m}^2/\text{s}^2$]
k_d	specific decay parameter [days^{-1}]
k_h	heat conductivity [W/(m·°C)]
k_r	relative permeability
k', k''	polymer viscosity parameters
K	permeability, [mD]
K_S, K_N	half rate constants for substrate and nutrient
n	number

U_N	Utility parameter for nutrient, [amount nutrient used/new biomass]
Up	Utility parameter for product [mass ratio]
V	volume (user defined units)
X	biomass [g/ml water]
Y	yield coefficient [g biomass/g substrate]
W	watt [$\text{kg}\cdot\text{m}^2/\text{s}^3$]
ϕ	porosity [volume fraction]
$\dot{\gamma}$	shear rate [sec^{-1}]
λ_c	formation in-situ shear rate parameter
μ	viscosity [cP]
$\mu_{\max}$	specific growth rate [days^{-1}]
ρ	density [g/cm^3]
$[\mu]$	intrinsic viscosity [g/ml]

Subscripts

0	reference condition
d	dissolved, death
e	endpoint
hf	half-rate constant
j	phase index, *j*=w, o, g
g	gas
o	oil
p	polymer, product
r	residual

N	nutrient (concentration)	w	water
p	pressure, [bar]	x	x-direction
P_c	capillary pressure, [bar]	y	y-direction
pv	pore volume	t	total
q	rate [m^3/day]	z	z-direction
Q	adsorption capacity [pv]		
r_d	specific biomass decay rate [g/day]		
r_g	specific biomass growth rate [g/day]		
S	saturation [volum fraction]		
S	substrate concentration [g/ml water]		
t	time [days]		
T	temperature [°C] or [°K]		
u	Darcy velocity [m/sec]		

3 Introduction

3.1 First version

This is the introduction to the first version 30. May 2008.

The *Bugsim* simulator described in this report is intended for the investigation of possible MEOR mechanisms. It is suitable for simulating laboratory experiments and smaller field scale models. Several such programs exist already, of which the more advanced one is the chemical simulator UTCHEM[1]. It is however our experience, from other research areas at IRIS, that when using existing software researching on complex processes one will very often encounter the limitations of the program and spend much time on questionable work-arounds. With direct access to the source code, one may easily implement various algorithms and rigorously test several hypotheses. If one can establish an iterative process between simulation and the other work-tasks of the project, the result can be very fruitful in the form of improved understanding and technology progress.

In MEOR one sometimes gets the impression that the number of proposed mechanisms surpasses the number of reliable quantitative measurements. As pointed out earlier, the use of the MEOR simulator may help the interpretation of laboratory experiments and scaling the results to the oil field scale. It is also helpful in quantifying the cost, e.g., if a proposed process (like gas production) depends on external supply of food, one would like to know how much food is needed.

This first version of *Bugsim* is regarded as a prototype, and is expected to be further developed. Some important phenomena like temperature and pH are not modelled, and the number of implemented MEOR processes is limited. Still, the program has already several useful functionalities and we believe it forms a solid framework for further developments.

The program is basically a 3D, black oil, three phase (water, oil and a simplified representation of gas) simulator, extended to handle multiple components mainly dissolved in the water phase. It can simulate both Cartesian, centre point (in an approximate way) and radial grids. The flow equation is solved by the standard IMPES method[2] (implicit in pressure and explicit in saturation). Although this method requires small time steps, it is very efficient for easy implementation and testing of various models involving multi-component flow.

Bugsim has a flexible model for biomass growth based on the standard Monod equation, but extended to allow for an arbitrary number of bacteria types, substrates and nutrient groups. Competitive utilisation of different substrates and different nutrients within each group can be modelled. Various microbial products can be specified, each associated with one single substrate source consumed by one specific bacteria type or with several such sources. A brief description of the growth model is given in Chapter 6.5. A limited number of possible MEOR effects are included in the present version. A permeability reduction is estimated based on the amount of adsorbed biomass. The remaining mechanisms are related to microbial products, in particular produced gas and polymer. Produced gas may swell the oil and reduce its viscosity, while polymer may increase the water viscosity and thereby improve the oil displacement sweep.

3.2 Bugsim version 1.20

Several improvements were implemented in the program within the project *MEOR phase II* (2010-2012). These were:

- New surfactant component defined. The surfactant may be introduced externally or produced in-situ as a microbial product. The surfactant is assumed to stay in the water phase, and interfacial tension is assumed a function of the aqueous surfactant concentration, $IFT(c_{w,s})$. Adsorption is included. Multiple surfactants are handled in the IFT calculations and with competitive adsorption.
- Rate dependent relative permeability is modelled using the capillary number to interpolate, $k_r(N_c)$. This option is used for modelling the effect of surfactant (may also be used without surfactant).
- Effect of biofilm may be modelled by letting relative permeability be a function of adsorbed bacteria (biofilm), $k_r(A_b)$.
- Adsorption is included for polymer(s).
- Temperature calculations are implemented, handling heat energy transport by conduction and convection.
- The effect of temperature on the microbial growth rate is modelled. Effects on fluid viscosity were already included in the previous version, however, the equation used is modified to give a more accurate representation of the temperature dependent viscosity of the water phase.
- Some errors in the capillary pressure model are fixed. The model is extended to allow J-function scaling.

The temperature calculations and the models handling the effect of surfactant and biofilm are described in Chapter 6 Technical description.

3.3 IORCoreSim

Main new implementations

- New sequential solver allows proper handling of capillary pressure at the core-scale.
- The specification of open phase boundary conditions used in e.g. core floods and spontaneous imbibition experiments has been simplified. New boundary conditions allow simulation of spontaneous imbibition experiments. Well pressures (injectors and producers) can now be specified at a given depth (and not fixed to the first perforated cell). Finally, a boundary Pc can be specified which may limit production of the non-wetting phase.
- Geochemical model with computation of mineral dissolution and precipitation, pH calculations which is important in may chemical processes related to IOR, ion exchange and ion complexation at the rock surface. The geochemical model is an important tool for analysing low salinity and ‘Smart-water’ EOR experiments and identify underlying mechanisms.
- New polymer model covering the different flow regimes encountered in a reservoir. The model incorporates variation in reservoir properties (permeability, porosity and temperature) as well as variations in polymer concentration and salinity.

4 Overview

4.1 Input

The input part of *IORCoreSim* is in the form of keyword-driven input. The reading of input parameters used by the program is triggered by specific keywords. Most parameters are given default values, which are used if not overwritten by the user input. An example input file is listed in the Appendix B.

Some general rules for the input data are the following:

- Keywords may be in lowercase or uppercase letters (or even mixed).
- Keywords must start in the first column.
- Lines starting with '--' or '*' in the first column are treated as comments. Comments may also be added after a keyword (with space in between) and after the slash (/) symbol.
- Multiple entries may be shortened by using '*', e.g.: 5*2 is the same as 2 2 2 2 2
- The number of characters on a line should not exceed 1023.
- A record is defined as the content starting on the first non-comment line after a keyword and ending with a slash (/).
- Input record following a keyword must start on a new line (after the keyword). An example for the keyword 'griddim' is as follows:

```
griddim                          (comment after keyword: grid dimension input)
*  nx     ny    nz    irad         (2 comment lines)   (parameter names)
-- -     1     1     0                               (default values)
   50    1     1     0   /       (comment after (/) is allowed)
```

The keyword 'griddim' in this case triggers the reading of 4 parameters shown above, namely the three-dimensional grid dimensions and a radial option flag. The record must end with a slash '/' (escape character) which may be on either the same or on a new line. When / is found the reading is interrupted, and the remaining parameters in the record (if any) will be assigned their default values. In the example given above, only the first parameter, 'nx', has been changed with the remaining ones having values equal to the default ones, therefore one could instead have written:

```
griddim
*  nx     ny    nz    irad
50     /
```

The '/' is also used to terminate the reading of tables with an unknown number of rows.

Bugsim reads data from a single input file (text format), but data from other files may be included anywhere in the input file by using the keyword 'include' followed by the relative pathname of the file containing the data. Here is an example demonstrating inclusion of two files from different locations:

```
include permdata.txt   / includes a file from the same folder as the input file.
include
data/poro.txt  / includes a file (poro.txt) from the 'data' directory.
```

Nested include files are allowed (i.e., the included file may contain yet another include file). The file names cannot contain blanks.

The input is organized in seven sections, six mandatory and one optional (heatsect). The section must appear in the following predefined sequence:

```
runsect
*   section containing dimensions of the run and main governing parameters


gridsect
*   section for grid data input (permeability, porosity, etc.)


propsect
*   section for phase properties (relative permeability, viscosity, etc.)
*   and component properties


heatsect
*   section for temperature computation (heat flow by conduction and convection etc.)
*   this section is optional


initsect
*   section for initialisation of the model (pressure, saturation etc.)


compsect
*   section for computational parameters (e.g., regulating time steps) and printouts


wellsect
*   section for defining wells and injection/production constraints
```

The input sections must appear in the order listed above. Within each section, keywords may appear in any order except in some obvious cases. For example, attempting to change or copy a variable using the keyword ‘multiply’ or ‘copy’ before the addressed variable is read, will obviously provide no meaningful information to the simulator. Also, input data may be overwritten by re-entering the same (or a related) keyword. In such cases, the order of appearance may affect the final input.

4.2 Output

The results of the computations are written in text format to a number of files. The filenames contain the input file name + a prefix or suffix indicating the file content. Assuming *file.txt* is the name of the input file, the output files are:

Filename	Description of content
file_log.txt	File contains logging information from reading of the input, and data checking.
file_dist.out	File contains summary reports and grid data printouts at specified report times.

file-wellname-inje.dat file-wellname-prod.dat file-wellname-imb.dat file-hist.dat file-block.dat	History files for injection, production and average model properties. These are text files containing tabulator delimited columns which can be imported (pasted) in e.g. Excel.

An example input file demonstrating the program in a 2D cross-sectional model with two rock types is listed in the Appendix B. Results from running the example simulation case are shown in the form of production plots generated by importing the history files to Excel, and distribution plots using a utility plotting program. The example demonstrates stimulated in-situ bio-mass production and potential MEOR increased oil recovery.

4.3 Sections

The input keywords are described in alphabetic order within the six following sections:

1. runsect: Section containing dimensions of the run and main governing parameters.
2. gridsect: Section for grid data input (permeability, porosity, etc.).
3. propsect: Section for phase properties (relative permeability, viscosity, etc.).
4. heatsect: Section for temperature parameters (heat convection and conduction)
5. initsectSection for initialising the model (saturations, phase concentrations, pressure).
6. compsect: Section for computational parameters (time step, solver tolerance) and regulating printout.
7. wellsect Section for defining wells and injection schedules.

5 Keywords

5.1 RUNSECT

This is the first section of the input file containing the main data for the run.

runsect keywords

Keyword	Short description
component	Components definition
griddim	Grid dimension
hystsat	Activates hysteresis, relative permeability and capillary pressure depending on saturation history.
krflag	Relative permeability options (rate dependency, directional)
nrock	Number of rock types (satnum)
nwell	Number of wells
phases	Definition of phases present
tabdim	Maximum dimensions of tables
title	Title of the run
units	Definition of units for input and output

---

1. component

Section: **runsect**

Description:

Use this keyword to define components in your simulation project. All components must be defined in this section. Each component is defined by a *category* and a *name* followed by a slash (/). The name must be unique because it will be used to identify components later in the input file. Each record defining a new component must start on a new line. An empty record (only blanks before '/') will terminate the keyword. The component categories are:

WATER (or WAT) :	water component
OIL :	oil component
GAS :	gas component

SURFACTANT: surfactant component

POLYMER: polymer component

SUBSTRATE: substrate is the main food source (carbonate source) for bacterial growth

BACTERIA: the product of microbial growth

NUTRIENT: nutrient for microbial growth

TRACER: tracer component

The user may define an arbitrary number of components within each category. The components can be defined in any order, however, the program will sort the components according to the category list above. In later sections of the input file, e.g., when defining concentrations, this sorted list will apply. Alternatively, components' concentrations may be defined by using their names.

Components of the two last categories (NUTRIENT and TRACER) are treated as non-volumetric (not occupying any space).

Notes:

- Only the three first characters of the category name are used to identify the component type, e.g., WATER and POLYMER may be abbreviated to WAT and POL.
- Category names may be in upper- or lower-case.
- Component names are case sensitive. Later references to a component (e.g., under keyword winject) must match the spelling of the name used here.
- Each component must be defined in a separate record (starts on a new line and terminated with a slash (/)). An empty record terminates the keyword.

Example, defining six components of type water, oil, gas, substrate, bacteria and nutrient:

```
component
*category name
 wat      brine /          ‘/’ terminates record for first component
 oil      oil   /
 gas      CH4   /
 sub      S1 /
 bac      B1 /
 nut      N1 /
/            empty record terminates keyword
```

2. griddim

Section: **runsect**

Description: Defines the grid dimension in the run. Up to four items terminated with a slash ‘/’ can be specified:

Item	Name	Description	Default
1	*nx*	Number of grid blocks in the x-direction	-

2	*ny*	Number of grid blocks in the *y*-direction	1
3	*nz*	Number of grid blocks in the *z*-direction	1
4	*gtype*	Flag indicating grid geometry (Cartesian or radial) 0: Cartesian (default) 1: radial 2: corner point grid	0

Example: 2D model.

```
griddim
*   nx    ny    ny
20   50     /   nz is defaulted to 1 and gtype to 0 (cartesian grid)
```

3. hystsat

Section: **runsect**

Description: Activates the use of hysteresis in saturation functions. Reads two items terminated with a slash ‘/’:

Item	Name	Description	Default
1	*krHystFlag*	Options: 0 : Hysteresis not used for relative permeability 1 : Killough’s method 2 : Modified Killough’s method	0
2	*PcHystFlag*	0 : Hysteresis not used for capillary pressure 1 : Activates hysteresis in capillary pressure.	0

Hysteresis models are described in chapter 6.3 Hysteresis.

Note: The hysteresis model for relative permeability accounts for hysteresis in residual oil saturation which is specified using keyword hystmod in the propsect section. If hysteresis in S_{or} is activated, then hysteresis for capillary pressure is required (unless zero capillary pressure is used). The capillary pressure hysteresis model accounts for the difference in curve shape between imbibition and drainage, and can be used alone without activating hysteresis for relative permeability, assuming there is no hysteresis in s_{or}.

Example:

```
hystsat
*   krFlag   PcFlag
2        1     /
```

4. krflag

Section: **runsect**

Description: Defines directional relative permeability and interpolation method needed in e.g. surfactant flooding. Up to four items terminated with a slash '/' can be specified:

Item	Name	Description	Default
1	*mdirkr*	Maximum directional relperm sets (1,3) in x-, y-, z-direction.	1
2	*iKrMode*	Mnemonic indicating method for interpolating between relative permeability sets and capillary pressure sets (optional). Old number flag in parenthesis. Both upper and lower-case letters allowed. Options: n (0): No interpolation (constant relative permeability). Nc (1): Interpolation using the capillary number (Nc). Must be activated in surfactant flooding. adsb (2): Interpolation using amount of adsorbed bacteria. ct (3): Interpolation using total concentration of component specified in item 4 ads (4): Interpolation using adsorption of component specified in item 4 cw (5): Interpolation using water phase concentration of component specified in item 4 pH Interpolation using pH. Geochemical calculations must be activated. psi Interpolation using surface potential. Geochemical calculations with surface complexes must be activated.	n
3	*mKrSet*	Maximum number of relative permeability sets per rock type used in the interpolation.	1
4	*kcKrName*	Component name or number used for interpolation when *iKrMode* ≥ 3	-

Example: Activates interpolation of relative permeability, rate dependent kr(Nc).

```
krflag
*   mdirkr iKrMode mKrSet kcKrName
1      nc       2  /
*   example: activate interpolation using adsorption of component 'SO4'
krflag
1       ads       2       SO4  /
```

5. nrock

Section: **runsect**

Description: Defines the number of rock types. Each rock type requires separate capillary pressure (optional) and relative permeability (mandatory) curves as input (corresponds to the keyword SATNUM used by *Eclipse*).

Default value: 1

Example: two rock types.

```
nrock
2   /
```

6. nwell

Section: **runsect**

Description: The number of wells.

Default value: 0

Example: two wells.

```
nwell
2   /
```

7. phases

Section: **runsect**

Description: Flags indicating the presence of oil, water and gas phases in the computations.

Item	Name	Description	Default
1	*iph*(1)	Water flag	1
2	*iph*(2)	Oil flag	1
3	*iph*(3)	Gas flag Options for *iph*(*i*), *i*=1, 2, 3 0: The phase is not allowed 1: The phase is allowed	0

Note: Even when no gas phase is allowed (*iph*(3)=0), gas components may still be present dissolved in the oil and water phase.

Example: all phases present

```
phases
*   water oil   gas
1     1     1    /
```

8. tabdim

Section: **runsect**

Description: Optional keyword that defines maximum dimension (rows) of input tables (relative permeability and capillary pressure).

Default: 100

Example: increase default settings from 100 to 200 rows:

```
tabdim
200 /
```

9. title

Section: **runsect**

Description: Optional, description of the run.

Item: A description of the run. The length must not exceed 255 characters and must be terminated with a slash ‘/’.

Example: define the title of a run

```
title
3-phase wag test.
/
```

10. units

Section: **runsect**

Description: Flags defining parameter units in input and printout. Up to 7 integer values on a single record are read. The available units for various properties and their corresponding flag values are given in the following table:

Item	Name	Description		Default
1	*iul*	Length:	0: “cm”, 1: “m”, 2: “in”, 3: “ft”	1
2	*iut*	Time:	0: “days”, 1: “hours”, 2: “min”, 3: “sec”	1
3	*iur*	Rate:	0: “ml/sec”, 1: “ml/min”, 2: “ml/hr”, 3: “m^3/d”, 4: “cft/d”, 5: “bbl/d”	3
4	*iuk*	Permeability:	0: “mD”, 1: “D”	0
5	*iup*	Pressure:	0: ”kPa”, 1: ”MPa”, 2: ”mBar”, 3: ”Bar”, 4: ”psi”, 5: ”atm”	3
6	*iuv*	Volume:	0: “ml”, 1: “litre”, 2: “m^3”, 3: “cft”, 4: “bbl”	2
7	*iud*	Density:	0: “g/ml”, 1: “s.g.”, 2: “kg/m^3”, 3: “lbm/cft” 4: “lbm/gal”	0

Example: defining parameters units in cm, min, ml/min, mD, bar, cm^3 and g/ml.

```
units
*L  t   q   K   p   v   rho
0   2   1   0   3   0   0 /
```

5.2 GRIDSECT

This is the second section of the input file containing grid data.

gridsect keywords

Keyword	Short description
box	Define input box for grid variables
coord	Coordinate lines used with corner point geometry
copy	For copying K_x to K_y or visa versa
dtop	Grids depth and orientation
dx	Grid dimensions in *x*-direction
dy	Grid dimension in *y*-direction
dz	Grid dimension in *z*-direction
gelnum	Defines grid cells as gel with special properties
ntg	NTG net to gross ratio (compensate for shale inclusions)
multx, multy, multz	Transmissibility multiplier i *x*, *y* or *x*-direction.
permdep	Compressibility dependent permeability, $k = k_0(\phi/\phi_0)^n$
permx	Permeability in *x*-direction
permy	Permeability in *y*-direction
permz	Permeability in *z*-direction
poro	Porosity
regnum	Assigns region numbers to grid cells
rocktype	Rock type distribution
zcorn	Grid cell corner depths used in corner point geometry

---

11. box

Section: gridsect, initsect

Description: Sets input box for grid variables like permx etc.. Reads a single record with grid coordinates, terminated with a slash ‘/’:

Item	Name	Description	Default
1	*i1*	1st grid number in *x*-direction	1

2	*i2*	2nd grid number in *x*-direction	*nx*
3	*j1*	1st grid number in *y*-direction	1
4	*j2*	2nd grid number in *y*-direction	*ny*
5	*k1*	1st grid number in *z*-direction	1
6	*k2*	2nd grid number in *z*-direction	*nz*

Example: adding a low permeability barrier in a 20×20×10 cell model

```
permxc
*  constant permeability
2000 /
box
10 10 1 20 1 10 / define box region
permxb
*  reads Kx for cells defined by box (200 values)
*  if less values is found, last value will be assigned to remaining cells in box
10 /
```

12. coord

Section: **gridsect**

Description: The **coord** keyword triggers reading of (*nx*+1)·(*ny*+1) vertical coordinate lines used to define the corner point grid. Each coordinate line is specified with two points which each require 3 values, (x_1, y_1, z_1) and (x_2, y_2, z_2). The total number of values read is then (*nx*+1)·(*ny*+1)·6 items terminated with a slash '/':

The coordinate lines form a set of vertical pillars where all grid cells with the same indexes in the horizontal plane (i, j) will form a stack of cells in vertical directions with all cell corners laying on the same four coordinate lines. The data is read one coordinate line at the time (6 values) with the x-index cycling faster (*i* = 1, *nx* +1).

Notes:

- Grid type must be set to corner point geometry in the **runsect** section.
- The cell dimensions are computed combining the coordinate lines entered here and cell corners depth entered via keyword zcorn.coord

Example with *nx*·*ny* = 3·2.

```
coord
--(nx+1)*(ny+1)*6=4*3*6=72 values (12 coordinate lines)
*  x1  y1 z1      x2 y2 z2
  0   0 2000    0   0 2200
100   0 2000  100   0 2200
200   0 2000  200   0 2200
300   0 2000  300   0 2200
  0  50 2000    0  50 2200
100  50 2000  100  50 2200
200  50 2000  200  50 2200
300  50 2000  300  50 2200
0 100 2000    0 100 2200
100 100 2000  100 100 2200
200 100 2000  200 100 2200
300 100 2000  300 100 2200
/
```

13. copy

Section: **gridsect**

Description: Copies **permx** to **permy** or vica versa. A copy operation is defined on a single record terminated with a slash '/'. A slash '/' ending an empty record terminates the keyword.

Item	Name	Description	Default
1	*key1*	*Item 1* to be copied. Possible values: *pemx, permy, permz*	-
2	*key2*	Target *item 2*. Possible values: *pemx, permy, permz*	-
3	*fact*	Anisotropy factor. $item2 = item1 \cdot fact$	1

Note: Copying is done immediately. The keyword is terminated with a slash (/) on a new line.

Example: copying K_x to K_y with anisotropy factor 0.2.

```
copy
  permx permy  0.2  /
/
```

14. dtop

Section: **gridsect**

Description: Defines the grid depth and orientation. Three items terminated with a slash '/':

Item	Name	Description	Default
1	*Dtop*	Depth to the top of block (1,1,1). User defined length-units.	0
2	*anglex*	Angle between *x*-axis and horizontal direction (degrees counter-clockwise). A positive value indicates increasing depth with increasing *x*.	0
3	*angley*	Angle between *y*-axis and horizontal direction (degrees). A positive value indicates increasing depth with increasing *y*.	0

Example: depth of the top first block set to 2000m, and depth increases in *x*-direction (10°) and decreases in the *y*-direction (-5°).

```
dtop
*   Dtop anglex angley
2000  10     -5 /
```

15. dx, dxb, dxx, dxc

Section: **gridsect**

Description: Grid block dimensions in the *x*-direction. Input data for the keyword must be terminated with a slash ‘/’. Variants of keyword:

dx: Reads $nbl = nx \cdot ny \cdot nz$ values. Should not be used with radial model (*irad* flag entered via the **griddim** keyword).

dxb: Reads a value for every block in the box region. Should not be used with radial model.

dxx: Reads *nx* values, a value for each block in the *x*-direction.

dxy: Reads *ny* values, each value assigned to the corresponding *xz*-plane.

dxz: Reads *nz* values, each value assigned to the corresponding *xy*-plane.

dxc: Reads a constant value assigned to all grid blocks.

If the radial model is selected (*irad*=1, see **griddim**):

dxx: Reads *R0*, *dxx*(*i*) , *i*=1,*nx* (1+*nx* values). *R0* is the well radius and *dxx*(*i*) is assigned as radial thickness to block *i* in the radial direction.

dxc: Reads two items, *R0* and *dxc*. *R0* is the well radius and *dxc* is assigned as constant radial thickness of all grid blocks.

Note: If fewer values of block dimensions than required are read, the remaining block dimensions are set equal to the last input value.

Units: User defined (cm, m, in or ft).

Example: individual size for all blocks in the *x*-direction with *nx*=10.

```
dxx
2*10 20 25 20 5*10 /
```

16. dy, dyb, dyy, dyc

Section: **gridsect**

Description: Grid block dimensions in the *y*-direction. Input data for the keyword must be terminated with a slash ‘/’. Variants of keyword:

dy: Reads $nbl = nx \cdot ny \cdot nz$ values. Should not be used with radial model (*irad* flag entered via the **griddim** keyword).

dyb: Reads a value for every block in the box region. Should not be used with radial model.

dyc: Reads a constant value assigned to all grid blocks.

dyy: Reads *ny* values, a value for each block in the *y*-direction.

dyx: Reads *nx* values, each value assigned to the corresponding *yz*-plane.

dyz: Reads *nz* values, each value assigned to the corresponding *xy*-plane.

Note: If fewer values of block dimensions than required are read, the remaining block dimensions are set equal to the last input value.

Units: User defined (cm, m, in or ft). If the radial model is selected (*irad*=1, see **griddim**), the units are in degrees and the sum of the *ny* sectors should be ≤ 360º.

Example: individual size for all blocks in the *y*-direction with *ny*=20.

```
dyy
5*10 10*20 5*10 /
```

17. dz, dzb, dzz, dzc

Section: **gridsect**

Description: Grid block dimensions in the *z*-direction. Input data for the keyword must be terminated with a slash '/'. Variants of keyword:

dz: Reads *nbl* = *nx*·*ny*·*nz* values. Should not be used with radial model (*irad* flag entered via the **griddim** keyword).

dzb: Reads a value for every block in the box region. Should not be used with radial model.

dzz: Reads *nz* values, a value for each block in the *z*-direction.

dzx: Reads *nx* values, each value assigned to the corresponding *xz*-plane.

dzy: Reads *ny* values, each value assigned to the corresponding *yz*-plane.

dzc: Reads a constant value assigned to all grid blocks.

Note: If fewer values than required are read, the remaining block dimensions are set equal to the last input value.

Units: User defined (cm, m, in or ft).

Example: individual size for all blocks in the z-direction with *nz*=10.

```
dzz
5*10 5*5 /
```

18. gelnum

Section: **gridsect**

Description: Special keyword for simulation of laboratory experiments where part of the grid represents a gel. The gel cells are identified by assigning a positive integer. Input data for the keyword must be terminated with a slash '/'. Variants of keyword:

gelnum: Reads $nbl = nx \cdot ny \cdot nz$ numbers.

gelnumb: Reads a number for every cell in the box region.

Note: If fewer values than required are read, the remaining cells in are assigned the last number read, with a warning written to the log-file. By default, all grid cells are assigned gelnum=0, so only gel cells need to be assigned a value gelnum=1.

Example: a core submerged in gel for spontaneous imbibition from the gel is represented by a radial grid with 40*1*50 grid blocks. The 10 outer cells in radial direction, and 10 cells at the top and bottom in vertical direction represents the gel.

```
*  First set all cells as gel cells
gelnum
1 /
regnum
2 / Gel cells defined as region 2

*  Next, overwrite interior grid representing the core
box
1 30 1 1 11 40 /  the core
gelnumb
0 / Change back to non-gel cells
regnumb
1 / Core defined as region 1
box
/ resets box to total grid
```

19. multx, multy, multz

Section: **gridsect**

Description: Multiplier for interblock transmissibility. Input data for the keyword must be terminated with a slash '/'. Variants of keyword:

multx: Reads T_x-multiplier for every block, $nbl = nx \cdot ny \cdot nz$ values. Item i modifies transmissibility $T_{x,i}$ in the x-direction between cell i and i+1.

multxb: Reads T_x-multiplier for every block in the box region.

multxc: The T_x-multiplier is constant, reads a single value assigned to all grid blocks.

multxx: Reads nx values, each value assigned to corresponding yz-plane.

multxy: Reads ny values, each value assigned to corresponding xz-plane.

multxz: Reads nz values, each value assigned to corresponding xy-plane.

multy: Reads T_y-multiplier for every block, $nbl = nx \cdot ny \cdot nz$ values. Item i modifies transmissibility $T_{y,i}$ in the y-direction between cell i and $i+nx$.

Variants of keyword defined by a sixth letter (*b*, *x*, *y* or *z*), see description for *multx*.

multz: Reads T_z-multiplier for every block, $nbl = nx \cdot ny \cdot nz$ values. Item i modifies transmissibility $T_{z,i}$ in the z-direction between cell i and $i+nx \cdot ny$.

Variants of keyword defined by a sixth letter (*b*, *x*, *y* or *z*), see description for *multx*.

Note: If fewer values than required are found, the remaining block or layer values are set equal to the last input value. If the box option is used, grid cells outside the box region keep their default value 1.0.

Although *nbl* values are read, the values for the last blocks in each direction are not used, e.g., only (*nx*-1)·*ny*·*nz* values are used for *multx*.

Units: Dimensionless.

Example: grid with *nx*=10, *ny*=5 and *nz* = 10 is split into two isolated zones.

```
permxz
*  Kx(assigned for each layer
5*500
5*50  /

box
1 10 1 5 5 5 /
multzb
0 / results in zero communication between layer 5 and 6
```

20. ntg

Section:	**gridsect**
Description:	Net to gross ratio, *ntg*. Variants of keyword:
ntg:	Reads *nbl* = *nx*·*ny*·*nz* values, for all grid blocks.
ntgi:	Reads *ntg*(*ir*) for every rock type (*ir*), *nrock* values. Rock type distribution (rocktype or satnum keywords) must be defined before this variant of the keyword is used
ntgb:	Reads *ntg* for every block in the box region.
ntgc:	*ntg* is constant, reads a single value assigned to all grid blocks.
ntgl:	*ntg* by layers, reads *nz* values, *ntg* is constant within each layer.
ntgx:	Reads *nx* values, each value assigned to corresponding *yz*-plane.
ntgy:	Reads *ny* values, each value assigned to corresponding *xz*-plane.
ntgz:	Reads *nz* values, each value assigned to corresponding *xy*-plane.

By default, the horizontal transmissibilities, see Eq. (5.1), are scaled with *ntg*, while the effect in vertical direction should be included in the vertical permeability (k_z) input. An alternative version of the keyword:

ntga:	Transmissibilities in all directions are scaled with *ntg*.
ntgax:	**x** = i, b, c, x, y, z with meaning described above.

Note: If fewer values than required are read, the remaining block or layer values are set equal to the last input value.

Units: Volume fraction.

The *ntg* compensates for tight non-productive part (shale inclusions) of the formation. The grid block pore volumes and horizontal transmissibilities are reduced with the *ntg* factor.

$$v_p = \frac{\phi v_b \cdot ntg}{v_b},\ v_b = \Delta x \Delta y \Delta z,\ ntg = (1 - fraction\ shales). \tag{5.1}$$

$$T_{cx} = \frac{ntg \cdot k_x \Delta y \Delta z}{\Delta x}, T_{cy} = \frac{ntg \cdot k_y \Delta x \Delta z}{\Delta y}, T_{cz} = \frac{k_z \Delta z \Delta y}{\Delta z}$$

The use of *ntg* is better than the alternative modification of rock properties like porosity and horizontal permeability because these properties are used in various model like e.g. scaling of capillary pressure and in-situ polymer rheology. Unless the *ntga* version of the keyword is used, the transmissibility in vertical direction (*z*-direction) is not modified and the effect of inactive inclusions must be reflected in the input k_z. In various other processes affected, conductivity of the process is in general scaled with *ntg* in all spatial directions due to lack of better alternatives. These processes are diffusion of species and heat and electrical In-situ shear rate is computed from flow rate and then there is a volumetric correction obtained by dividing with *ntg*.

Example: using grid with *nx*=10, *ny*=5 and *nz* = 4.

```
ntg
*   por(i), i=1, 200
100*0.5
100*0.8  /
```

21. permdep

Section: **gridsect**

Description: Allow permeability to depend on changes in porosity. Current default behaviour is that effective phase permeabilities scales with upstream porosity.

Default value: 1

Options:

0: permeability independent on ϕ/ϕ_0
1: permeability scales with ϕ/ϕ_0 (default)
2: permeability scales with ϕ^3/ϕ_0^3
3: permeability scales with $(\phi/\phi_0)^3 \cdot (\tau_0/\tau)$, τ = tortuosity

Note: Only changes in porosity due to compressibility is considered. If the third option is used, then the variable tortuosity model (rtort) must be specified.

The change in permeability can be viewed by printing the absolute permeability reduction variable specifying the printflag pr_rkf (RKF=k_0/k).

The scaling factor depends on the fluid pressure and rock compressibility number (n_{cr}) and can be formulated in terms of the volume ratio factor b_r:

$$b_r = \frac{\phi(p)}{\phi(p_0)} = e^{n_{cr}(p-p_0)}. \tag{5.2}$$

The three scaling options results in upstream permeability factors:

$$\begin{aligned} &1:\ f = b_r, \\ &2:\ f = b_r^3, \\ &3:\ f = b_r^3 b_r^{m-1}, \end{aligned} \tag{5.3}$$

where the tortuosity is substituted with ϕ^{1-m} and where m is the rock cementation index in Archie's resistivity model.

Example: scaling with ϕ^3.

```
permdep
2   /
```

22. permx, permxc, permxl

Section:	**gridsect**
Description:	Grid block absolute permeability in the *x*-direction. Input data for the keyword must be terminated with a slash '/'. Variants of keyword:
permx:	Reads K_x for every block, $nbl = nx \cdot ny \cdot nz$ values.
permxi:	Reads K_x for every rock type, *nrock* values. (rocktype or satnum keywords cannot occur after rock type dependent input has been used)
permxb:	Reads K_x for every block in the box region.
permxc:	K_x is constant, reads a single value assigned to all grid blocks.
permxl:	K_x by layers, reads *nz* values, K_x is constant within each layer.
permxx:	Reads *nx* values, each value assigned to corresponding *yz*-plane.
permxy:	Reads *ny* values, each value assigned to corresponding *xz*-plane.
permxz:	Reads *nz* values, each value assigned to corresponding *xy*-plane.

Note: If fewer values than required are found, the remaining block or layer values are set equal to the last input value.

Units: User defined (mD or Darcy).

Example: using grid with nx=10, ny=5 and nz = 4.

```
permx
```

```
*   Kx(i), i=1, 200
100*0.5
100*0.1  /

*  Alternative input
permxc
0.5 /  assigns a constant permx to all cells
box
1 10 1 5 3 4 /
permxb
100*0.1 / overwrites all cells in the two lower layers
```

23. permy, permyc, permyl

Section: **gridsect**

Description: Grid block absolute permeability in the *y*-direction. Input data for the keyword must be terminated with a slash '/'. Variants of keyword:

permy: Reads *nbl* = *nx*·*ny*·*nz* values.

permyi: Reads K_y for every rock type, *nrock* values. (rocktype or satnum keywords cannot occur after rock type dependent input has been used)

permxb: Reads K_y for every block in the box region.

permyc: K_y is constant, reads a single value assigned to all grid blocks.

permyl: K_y by layers, reads *nz* values, K_y is constant within each layer.

permyx: Reads *nx* values, each value assigned to corresponding *yz*-plane.

permyy: Reads *ny* values, each value assigned to corresponding *xz*-plane.

permyz: Reads *nz* values, each value assigned to corresponding *xy*-plane.

Note: If fewer values than required are found, the remaining block or layer values are set equal to the last input value.

Units: User defined (mD or Darcy).

Example: using grid with *nx*=10, *ny*=5 and *nz* = 4.

```
permy
*   Ky(i), i=1, 200
100*0.5
100*0.1  /
```

24. permz, permzc, permzl

Section: **gridsect**

Description: Grid block absolute permeability in the *z*-direction. Input data for the keyword must be terminated with a slash '/'. Variants of keyword:

permz: Reads *nbl* = *nx*·*ny*·*nz* values.

permzi:	Reads K_z for every rock type, *nrock* values. (rocktype or satnum keywords cannot occur after rock type dependent input has been used)
permxb:	Reads K_z for every block in the box region.
permzc:	K_z is constant, reads a single value assigned to all grid blocks.
permzl:	K_z by layers, reads *nz* values, K_z is constant within each layer.
permzx:	Reads *nx* values, each value assigned to corresponding *yz*-plane.
permzy:	Reads *ny* values, each value assigned to corresponding *xz*-plane.
permzz:	Reads *nz* values, each value assigned to corresponding *xy*-plane.

Note: If fewer values than required are read, the remaining block or layer values are set equal to the last input value.

Units: User defined (mD or Darcy).

Example: using grid with *nx*=10, *ny*=5 and *nz* = 4.

```
permz
*   Kz(i), i=1, 200
100*0.5
100*0.1   /
```

25. poro, porob, poroc, porol

Section: **gridsect**

Description: Porosity, ϕ. Variants of keyword:

poro:	Reads $nbl = nx \cdot ny \cdot nz$ ϕ-values, for all grid blocks.
poroi:	Reads ϕ for every rock type, *nrock* values. (rocktype or satnum keywords cannot occur after rock type dependent input has been used)
porob:	Reads ϕ for every block in the box region.
poroc:	ϕ is constant, reads a single value assigned to all grid blocks.
porol:	ϕ by layers, reads *nz* values, ϕ is constant within each layer.
porox:	Reads *nx* values, each value assigned to corresponding *yz*-plane.
poroy:	Reads *ny* values, each value assigned to corresponding *xz*-plane.
poroz:	Reads *nz* values, each value assigned to corresponding *xy*-plane.

Note: If fewer values than required are read, the remaining block or layer values are set equal to the last input value.

Units: Volume fraction.

Example: using grid with *nx*=10, *ny*=5 and *nz* = 4.

```
poro
```

```
*   por(i), i=1, 200
100*0.25
100*0.19  /
```

26. regnum

Section: gridsect

Description: Divide grid into regions for reporting purposes. Each cell is assigned a region number $\geq$ 1. By default, all cells belong to region 1. Input data for the keyword must be terminated with a slash '/'. Variants of keyword:

regnum: Reads $nbl = nx \cdot ny \cdot nz$ region numbers.

regnumb: Reads a region number for every cell in the box region.

regnumx: Reads *nx* values, each value assigned to corresponding *yz*-plane.

regnumy: Reads *ny* values, each value assigned to corresponding *xz*-plane.

regnumz: Reads *nz* values, each value assigned to corresponding *xy*-plane.

Note: If fewer values than required are read, the remaining cells in are assigned the last number read, with a warning written to the log-file.

Example: a 10×10×4 grid is split into four quadrants. Two alternative methods are shown.

```
*   First method enters 400 values,
regnum
5*1 5*2 5*1 5*2 5*1 5*2 5*1 5*2 5*1 5*2 5*4 5*3 5*4 5*3 5*4 5.3 5*4 5*3 5*4 5*3
5*1 5*2 5*1 5*2 5*1 5*2 5*1 5*2 5*1 5*2 5*4 5*3 5*4 5*3 5*4 5.3 5*4 5*3 5*4 5*3
5*1 5*2 5*1 5*2 5*1 5*2 5*1 5*2 5*1 5*2 5*4 5*3 5*4 5*3 5*4 5.3 5*4 5*3 5*4 5*3
5*1 5*2 5*1 5*2 5*1 5*2 5*1 5*2 5*1 5*2 5*4 5*3 5*4 5*3 5*4 5.3 5*4 5*3 5*4 5*3
/

*   Alternative input of same data:
box
1 5 1 5 1 4 /
regnumb
100*1 / This assignment of reg 1 is not necessary, all cells is initially reg 1.
box
6 10 1 5 1 4 /
regnumb
2 / Remaining 99 cells in current box is assigned last value found (reg 2)
box
6 10 6 10 1 4 /
regnumb
3 /
box
1 5 6 10 1 4 /
regnumb
4 /
box
/ resets box to total grid
```

27. rocktype, rocktypeb, rocktypel

Section: **gridsect**

Description: Assigns rock type numbers to gridblocks. The rock type number determines the set of phase flow functions (relative permeability and capillary pressure) to be used. Reads *nbl* (*nx*·*ny*·*nz*) values (see **griddim**) terminated with a slash '/'. Variants of keyword:

rocktype: Reads *nbl* = *nx*·*ny*·*nz* values.

rocktypeb: Reads a rock type number for every block in the box region.

rocktypel: Rock type number by layers, reads *nz* values, one for each layer.

rocktypex: Reads *nx* values, each value assigned to corresponding *yz*-plane.

rocktypey: Reads *ny* values, each value assigned to corresponding *xz*-plane.

rocktypez: Reads *nz* values, each value assigned to corresponding *xy*-plane.

Value: 1, *nrock* (see nrock keyword)

Note: If fewer than *nbl* (or *nz*) values are found, the remaining block values (or layer values) are set equal to last rock type number found.

Example: with *nx*=10 and *ny*=20 and two rock types (*nbl*=*nx*·*ny*·*nz*=200).

```
rocktype
*   itype(i), i=1, 200   (rock type indicator)
100*1
100*2   /
```

28. zcorn

Section: **gridsect**

Description: The **zcorn** keyword triggers reading of cell corners depth used with the corner point grid method. Each cell has 8 corners, for which the depths are read separately for each cell. The data read is (2·*nx*)·(2·*ny*) ·(2·*ny*) items terminated with a slash '/':

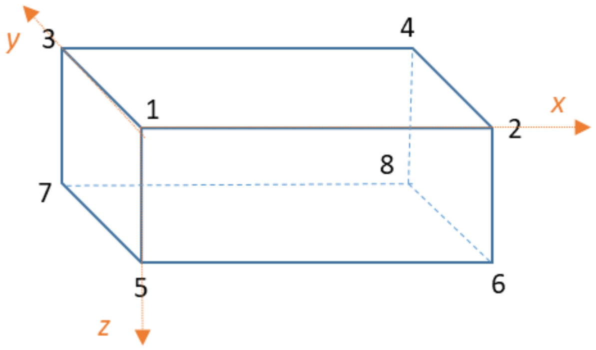


Figure 5.1 Numbering of grid cell corners.

Cell index $I = (i_z - 1) \cdot nx \cdot ny + (i_y - 1) \cdot nx + i_x$.

1. The reading of cell corner depths starts with the first line (i_y=1) in the first layer (i_z=1) reading two values (corners 1 and 2, as depicted in Figure 5.1) for each of nx cells (cell index I=1, nx). The next 2·nx values represent corners 3 and 4 for the same cells (i_z=1, i_y=1, i_x=1, nx).
2. The reading of the four upper face corner depths are repeated for the next lines, for i_y=2, ny. For each line, 4·nx values are read as described in point 1.
3. Next, the bottom phase corner depths (corners 5, 6, 7 and 8) are read in the same manner, starting with the first line (point 1) and repeated as described in point 2.
4. The procedure described by points 1-3 for the first layer (i_z=1) is repeated for the remaining layers (i_z=2, nz). A total of 8·ny·nx values are read for each layer.

Notes:

- Grid type must be set to corner point geometry in the **runsect** section.
- The cell corners x- and y-coordinates are computed from the depths entered here and the areal location of vertical cell pillars entered via keyword coord.coord

Example with $nx·ny·nz$ = 3·2·2 and depth increasing with 10 and 5 m/block in the x- and y-directions, respectively. Vertical block thickness is 4 and 8 m for layer 1 and 2:

```
zcorn
*   layer 1 - top face
*   corners 1 & 2 for the first line (I=1,nx), nx*2 values
2000 2010 2010 2020 2020 2030
*   corners 3 & 4 for the first line (I=1,nx), nx*2 values
2005 2015 2015 2025 2025 2035
*   corners 1 & 2 for the second line (I=nx+1,2*nx), nx*2 values
2005 2015 2015 2025 2025 2035
*   corners 3 & 4 for the second line (I=nx+1,2*nx), nx*2 values
2010 2020 2020 2030 2030 2040
*   layer 1 - bottom face
2004 2014 2014 2024 2024 2034
2009 2019 2019 2029 2029 2039
2009 2019 2019 2029 2029 2039
2014 2024 2024 2034 2034 2044
*   layer 2 - top face
2004 2014 2014 2024 2024 2034
2009 2019 2019 2029 2029 2039
2009 2019 2019 2029 2029 2039
2014 2024 2024 2034 2034 2044
*   bottom face
2012 2022 2022 2032 2032 2042
2017 2027 2027 2037 2037 2047
2017 2027 2027 2037 2037 2047
2022 2032 2032 2042 2042 2052
/
```

5.3 PROPSECT

This is the third section of the input file containing component and grid block properties.

propsect keywords

Keyword	Short description
adsorption	Adsorption - various models
amul	Adsorption multiplier, allows including effect of temperature, salinity etc.
bioplug	Activates biomass plugging
box	Define input box for grid variables (described in gridsect)
cmpprop	Component properties for WATER, OIL, GAS, SURFACTANT, POLYMER, SUBSTRATE, BACTERIA, NUTRIENT and TRACER.
Corey	Corey type relative permeability
diffusion	Component molecular diffusion
dispersion	Activate dispersion for species
dsprop	Phase viscosity and density, effect of dissolved species.
elcond	Electric brine conductivity and formation resistivity
gelmod	Gelation model.
gelsps	Gel specific surface area for estimating effective gel permeability.
hystmod	Hysteresis option for residual oil saturation and capillary pressure.
intkr	Relative permeability interpolation method
jscale	J-scaling options for capillary pressure
krLET	LET type relative permeability function
krpctab	Relative permeability and capillary pressure from tables
krtab	Relative permeability - table format
ncmisc	Miscibility function used for relative permeability interpolation.
nextir	Sets next relative permeability input.
pcowform **pcgoform**	Oil-water capillary pressure - formula Gas-oil capillary pressure - formula
polycse	Effective salinity model for polymer

polydegr	Polymer elongation and shear degradation
polyipv	Polymer inaccessible pore volume and depletion layer model
polyrkf	Polymer permeability reduction
pref	Reservoir conditions reference pressure
rockcmpr	Rock/formation compressibility (replaces rockprop)
rockshear	In-situ shear rate used with polymer model
rtort	Permeability and saturation dependent rock tortuosity
swof	Oil/water relative permeability and capillary pressure input, table
tlpmixpar	Todd Longstaff option for polymer

29. adsorption

Section: **propsect**

Description: Allows specification of different adsorption models for a component outside the component property keyword (cmpprop). This keyword must appear after the general component property input for the same component. The number of records for this keyword depends on the adsorption model.

The first record, terminated by a slash (/) :

1st Record, terminated with a slash '/'.			
Item	Name	Description	Default
1	*cname*	Component name for identification.	-
2	*iads*	Flag indicating adsorption model. Old number notations in parenthesis. Options: n (0): No adsorption alm (1): Langmuir type adsorption, equilibrium model. For components having an associated property (e.g. Mw for polymer), the property of adsorbent reflects the property in solution at the time of the adsorption. almt : Langmuir type adsorption, equilibrium model. For components having an associated property (e.g. Mw for polymer), the property of adsorbent is updated to match the solution property at all times. almir (2): Langmuir type, irreversible adsorption. almkin (3): Kinetic Langmuir type adsorption atab (4): Tabular adsorption atabkin (5): Kinetic tabular adsorption almasp: Langmuir type retention for associative polymer	n

		almkinasp: Kinetic LM type retention for associative polymer	
3	*ScaleQmF*	Flag indicating if scaling of *Qm* with porosity and permeability should be used (see record 3). Options: 0: No scaling 1: *Qm* scales with $(\phi/k)^{\alpha}$. $Q_m = Q_{m0}\left(\frac{k_{ref}}{\phi_{ref}}\cdot\frac{\phi}{k}\right)^{\alpha}$	0
4	*irflag*	Flag indicating different adsorption parameters should be used for each *rocktype*: 0: A single set of adsorption parameters uses for all rocktypes 1: The next adsorption record is read *nrock* times. Each record must end with a slash '/'.	0

2nd Record read only if *iads* = alm, almt, almir : Langmuir type adsorption. If *irflag*=1, this record must be repeated *nrock* times, each record terminated with '/'.

Item	Name	Description	Default
1	*b*	Adsorption parameter. Units: inverse of concentration units. The value should be chosen to get an appropriate ratio *bc*/(1+*bc*), which denotes the fraction of Q_m adsorbed at concentration *c*.	-
2	*Qm*	Maximum adsorption capacity. Units [pv fraction] for volumetric components.	-

2nd Record read only if *iads* = almkin : Kinetic Langmuir type adsorption. f *irflag*=1, this record must be repeated *nrock* times, each record terminated with '/'.

Item	Name	Description	Default
1	*ka*	Adsorption parameter. [pv vol. fraction/hr].	-
2	*kd*	Desorption parameter. [pv vol. fraction/hr].	-
3	*Qm*	Maximum adsorption capacity. Units [pv fraction] for volumetric components.	-
4	*rfg*	Shear rate dependency for adsorption/desorption rate. Botha *ka* and *kd* are multiplied with factor $f = (1+rfg\cdot gam^{0.5})$, where *gam* is the shear rate in sec^{-1}. (Currently not working).	0
5	*TFlag*	Flag indicating if *ka* and *kd* should change with temperature in the same way as molecular diffusion Options: 0: No scaling 1: Both *ka* and *kd* scale with $\frac{T\mu_0}{T_0\mu_w}$	0
6	*T0*	Reference temperature, T_0, for scaling adsorption kinetics, [°K]	293.15
7	*my0*	Reference viscosity μ_0 for scaling, Units: [cp] = [mPa·s]	1.0

2nd Record read only if *iads* = almasp : LM type retention for associative polymer.

Item	Name	Description	Default
1	*b*	Retention parameter. Units: inverse of concentration units. The value should be chosen to get an appropriate ratio $bc/(1+bc)$, which denotes the fraction of Q_m adsorbed at concentration c.	-
2	*Qm0*	Retention capacity Q_m at shear rate 1 sec^{-1}. Units [pv fraction] for volumetric components.	-
3	*na*	Shear dependency index for *Qm*. Suggested range: 0.3-0.5	0
4	*S0p*	Specific surface area for polymer. Suggested range: 4000-7000 m^2/ml polymer. Units: [m^2/ml]	5000

2nd Record read only if *iads* = almkinasp : Kinetic LM retention associative polymer

Item	Name	Description	Default
1	*ka*	Adsorption parameter. [pv vol. fraction/hr].	-
2	*kd*	Desorption parameter. [pv vol. fraction/hr].	-
3	*Qm0*	Retention capacity Q_m at shear rate 1 sec^{-1}. Units [pv fraction] for volumetric components.	-
4	*na*	Shear dependency index for Qm. Suggested range: 0.3-0.5	0
5	*rfg*	Shear rate dependency for adsorption/desorption rate. Botha *ka* and *kd* are multiplied with factor $f = (1+rfg\cdot gam^{0.5})$, where *gam* is the shear rate in sec^{-1}.	0
6	*S0p*	Specific surface area for polymer. Suggested range: 4000-7000 m^2/ml polymer. Units: [m^2/ml]	5000
7	*TFlag*	Flag indicating if *ka* and *kd* should change with temperature in the same way as molecular diffusion Options: 0: No scaling 1: Both *ka* and *kd* scale with $\frac{T\mu_0}{T_0\mu_w}$	0
8	*T0*	Reference temperature for scaling adsorption kinetics, [°K]	293.15
9	*my0*	Reference viscosity for scaling, Units: [cp] = [mPa·s]	1.0

2nd Record read only if *iads* = atab is a table of adsorption (in second column) versus component concentration (in first column). The table is terminated with a slash, ‘/’. If *irflag*=1, a seperate table must be entered for each rocktype, each terminated with ‘/’.

2nd Record read only if *iads* = atabkin : Kinetic adsorption with adsorption data entered as a table (in second column) versus component concentration (in first column). Input consists of one record containing kinetic data followed by equilibrium adsorption in a table. Both the record and the table must be terminated with a slash, '/'. If *irflag*=1, the input (record + table) must be repeated for for each rocktype.

Item	Name	Description	Default
1	*kd*	Desorption parameter. [1/hr]. Indicates the fraction of adsorbed mass that will be desorbed within an hour if all released mass in removed (C_{kc}=0). The adsorption parameter *ka* is computed internally from estimated equilibrium adsorption and *kd* assuming a local Langmuir type kinetic model.	-
2	*TFlag*	Flag indicating if *ka* and *kd* should change with temperature in the same way as molecular diffusion Options: 0: No scaling 1: Both *ka* and *kd* scale with $\frac{T\mu_0}{T_0\mu_w}$	0
3	*T0*	Reference temperature T_0 for scaling adsorption kinetics, [°K] If $T_0 < 270$, input is assumed to be in °C and increased with 273.15 °.	293.15
4	*my0*	Reference viscosity μ_0 for scaling, Units: [cp] = [mPa·s]	1.0

3rd Record read if *ScaleQmF* = 1 . If *irflag*=1 (activated input for all rocktypes), the record should be repeated nrock times if *irflag*=1, each time terminated with a slash, '/'.

Item	Name	Description	Default
1	*Q_kref*	Reference permeability for scaling, k_{ref}. Units: user defined, e.g. [mD]	-
2	*Q_firef*	Reference porosity for scaling, ϕ_{ref}. Units: volume fraction.	0.2
3	*Q_alfa*	Scaling exponent α: $Q_m = Q_{m0}\left(\frac{k_{ref}}{\phi_{ref}}\cdot\frac{\phi}{k}\right)^{\alpha}$	0.5

Example: tabular adsorption for a polymer component of name 'hpam'.

```
adsorption
*cname iads QmScale irFlag
 hpam  atab 1        0 /

*Cp     Ap(pv fraction)
0       0
```

```
0.00001 0.00002
0.00002 0.00004
0.00005 0.00010
0.00010 0.00012
0.00100 0.00014
/
*kref firef alfa
1000.0 0.25 0.5 /
```

30. amul

Section:	**propsect**
Description:	Adsorption multiplier with dependency given by variants of keyword:
amulT:	temperature
amulcsep:	effective salinity for polymer, polycse
amulcw:	water phase concentration of a specie
amulSw:	water phase saturation *Sw*

The input is one record terminated by a slash (/), followed by a table terminated with a slash (/) on a new line.

The first record: name of the adsorbing component and optional a specie name on which the adsorption depends.

Table: two columns and at least 2 rows. The first column should be the independent variable. The second column should contain the factor that the adsorption should be multiplied with at the corresponding temperature.

This keyword can be applied to all adsorption models (except the adsorption of bacteria).

The keyword can be repeated for the same component to model multiple adsorption dependency.

Note: the adsorption model must have been defined **before** this keyword is used

Example: reducing adsorption of component **hpam** at higher temperature and adding dependency on component named NaCl.

```
amulT
hpam /
-* T°C    amul
20        1
40        0.95
50        0.9
70        0.8
/
amulcw
hpam NaCl /
*cw_NaCl  amul
0.001     0.1
0.01      0.3
0.1       0.7
0.5       1.0
/
```

31. bioplug

Section: **propsect**

Description: Activates calculation of permeability reduction (plugging) by microbes.

One record with two items, terminated by a slash (/) :

Item	Name	Description	Default
1	*Ibrkf*	Flag indicating if plugging is considered. Options: 0: Permeability reduction is turned off. 1: Permeability reduction is estimated from surface area of attached (adsorbed) biomass using a Carmen-Kozeny approach. No input parameters. Surface area of the biomass is computed from bacteria properties (size and water content). 2: Permeability reduction is computed as above (*Ibrkf*=1), but only applied for the water phase.	0
2	*fbrkf*	Fraction of live microbes in the biofilm. A value below 1 will increase the plugging. An effective biomass retention, $A_{be}=A_b/fbrkf$, is used to compute the permeability reduction.	1

Example: turning biomass plugging on.

```
bioplug
1 /
```

32. cmpprop

Section: **propsect**

Description: Component properties. Properties are identified by their name entered in the runsect section (see keyword **component**). Data for multiple components can be entered.

The component is identified by its name as the first item. The number of records will depend on the component type.

Property input is described for components of types: WATER, OIL, GAS, SURFACTANT, POLYMER, SUBSTRATE, BACTERIA, NUTRIENT and TRACER.

Example: The example given for individual component types are obtained from the input file given in Appendix B.

33. cmpprop WATER

WATER properties.

Section: **propsect**

WATER properties are entered in one record terminated with '/':

Item	Name	Description	Default
1	*cname*	Component name for identification. An empty record terminates the keyword.	-
2	*Pref*	Reference pressure for compressibility, Units: user defined pressure units, e.g. [bar]	1 bar
3	*Bf*	Volume factor = Vres/Vsurface. Units: [Rm^3/Sm^3] Note; the B_f factor is currently not implemented in reported production data, and is only used if "surface conditions"-option is defined for injection wells. However, this is not recommended.	1
4	*density*	Density at *Pref.* Units: user defined density units, e.g. [g/ml]	1 g/ml
5	*compr*	Compressibility at *Pref.* Units: user defined inverse pressure units, e.g. [atm^{-1}]	$4.5 \cdot 10^{-5}$ bar^{-1}
6	*VPref*	Reference pressure used in viscosity. Units: user defined, e.g. [bar]	1 bar
7	*VTref*	Reference temperature used in viscosity. Units: [°C]	20°C
8	*Visc*	Viscosity at *Vpref* and *Vtref*. Units: [cp] =[mPa·s]	1 cp
9	*BTvisc*	Temperature parameter in viscosity. Units: [°C]	570.6 °C
10	*TCvisc*	Second temperature viscosity parameter. Units [°C]	133.15
11	*BPvisc*	Pressure parameter in viscosity. It's recommended to keep this zero for water. Pressure dependency for water viscosity is very temperature dependent, with decreasing μ_w at lower T and increasing μ_w at higher T. Units: user defined, e.g. [bar^{-1}]	0

Fluid viscosity (for water, oil and gas) is computed by a modified Arrhenius equation;

$$\mu = \mu_0 \, exp\left[B_T\left(\frac{1}{T+T_c} - \frac{1}{T_0+T_c}\right) + B_P(p - p_0)\right], \qquad (5.4)$$

where the temperature is in °C and index 0 indicates the reference conditions. Eq. (5.4) has two temperature parameters, B_T and T_C (in addition to the reference viscosity, μ_0), which gives a better fit to polar fluids like water (A2 in figure below). Using T_C = 273.15 corresponds to the traditional Arrhenius equation. Note that using μ_0 = 1.002 cp at 20°C and default B_T and T_C parameters is equivalent to the following equation for fresh water claimed to be accurate to within 2.5% from 0 °C to 370 °C.[1]

---

[1]: https://en.wikipedia.org/wiki/Temperature_dependence_of_liquid_viscosity

$$\mu = 2.544 \cdot 10^{-5} 10^{\frac{247.8}{T-140}}, \text{ with T in °K.} \qquad (5.5)$$

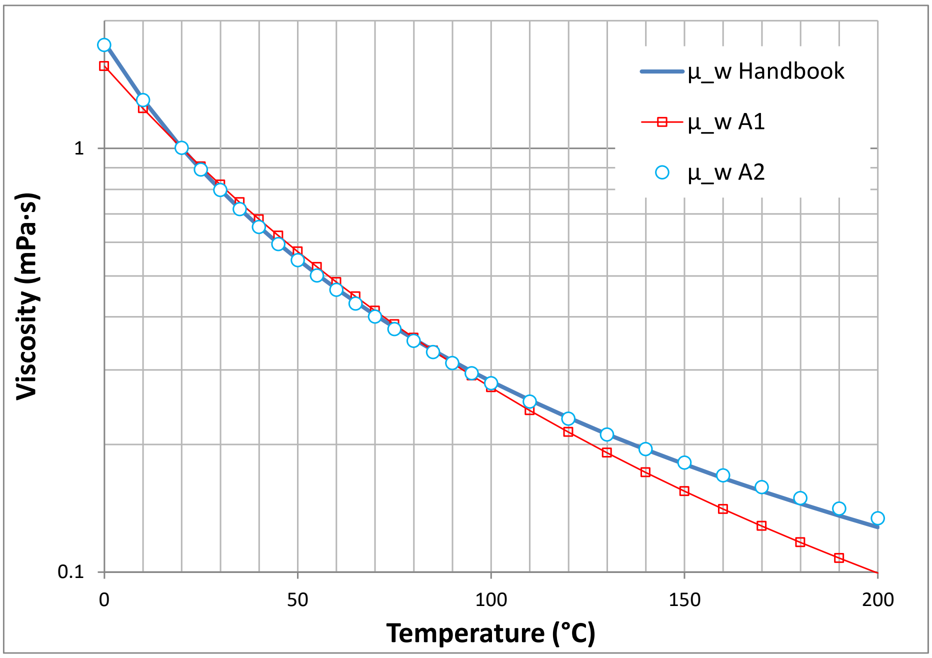


Figure 5.2 Water viscosity from Handbook of Chemistry and Physics (CRC-Press) and matched with single (A1) and dual (A2, with B_T=570.6 and T_C=133.15) parameter Arrhenius equations.

When dissolved gas is present (in either oil or water), the phase viscosity is computed by the fourth root mixing rule.

$$\mu_j = \frac{\mu_{jd}}{\left[1 - C_{g,j} + C_{g,j}\left(\frac{\mu_{jd}}{\mu_{gl}}\right)^{0.25}\right]^4}, \text{ j = w, o} \qquad (5.6)$$

Index *jd* indicates the phase property without dissolved gas and index *gl* indicates an apparent gas viscosity when dissolved in the liquid phase (see input for the GAS type below). The viscosities on the right-hand side are first computed with Eq. (5.4).

Example: defining properties for water component named WAT1.

```
cmpprop
*   Name  Prefc  Bw  density  Cw      PrefV  TrefV°C  visc   BTw    TCw     BPw
WAT1  80    1.0   1.0    4e-5    1.     90.      0.32   570.6 133.15 0.  /
/
```

34. cmpprop OIL

OIL properties.

Section: **propsect**

OIL properties are entered in one record terminated with ‘/’:

Item	Name	Description	Default
1	*cname*	Component name for identification. An empty record terminates the keyword.	
2	*Pref*	Reference pressure for compressibility, Units: user defined, e.g. [bar]	1 bar
3	*Bf*	Volume factor = Vr(at Pref)/Vsurface (NB: without gas) Note; the B_f factor is currently not implemented in reported production data, and is only used if “surface conditions”-option is defined for injection wells. However, this is not recommended.	1
4	*density*	Density at Surface condition. Units: user defined density units, e.g. [g/ml]	0.772 g/ml
5	*compr*	Compressibility at *Pref*. Units: user defined inverse pressure units, e.g. [atm^{-1}]	$1.0 \cdot 10^{-4}$ bar^{-1}
6	*VPref*	Reference pressure used in viscosity. Units: user defined, e.g. [bar]	1 bar
7	*VTref*	Reference temperature used in viscosity. Units: [°C]	20°C
8	*Visc*	Viscosity at *Vpref* and *Vtref*. Units: [cp] =[mPa·s]	1.42 cp
9	*BTvisc*	Temperature parameter in viscosity. If zero, the value is estimated from viscosity internally. Eq.(5.7). Units: [°C]	1500°
10	*TCvisc*	Second temperature viscosity parameter. Units [ºC] Default value (273.15°) corresponds to the traditional Arrhenius equation used for oils. If both *BTvisc* and *TCvisc* are zero, also *TCvisc* is computed internally.	273.15°
11	*BPvisc*	Pressure parameter in viscosity. Units: user defined, e.g. [bar^{-1}]	0.0016 bar^{-1}

The oil viscosity is computed from equations (5.4) and (5.6). If the input viscosity temperature parameter $B_T \leq 0$, then the temperature dependency is computed internally in *IORCoreSim* using the expression:

$$T^* = b/\,ln(\mu_0/a)\,, \mu^* = a \cdot exp\left(\frac{b}{T^* + c}\right), \tag{5.7}$$

$$a = 0.006861$$
$$b = 1638.9$$
$$c = 10$$

$$B_T = \frac{ln\left(\frac{\mu^*}{\mu_0}\right)}{\frac{1}{T_0+T_C+c} - \frac{1}{T_0+T_C}},$$

The temperature T is in °C and T_c would be 273.15 if the traditional Arrhenius equation is assumed.

If also input T_C is zero, then Tc is computed from the reference viscosity and temperature input and the a and b constants from Eq. (5.7):

$$\mu_o = a \cdot \exp\left(\frac{b}{T_0+T_C}\right), \qquad (5.8)$$

$$T_C = \frac{b}{\ln\left(\frac{\mu_O}{a}\right)} - T_0$$

Examples of how well the single variable model Eq. (5.8) matches real data are given in Figure 5.3.

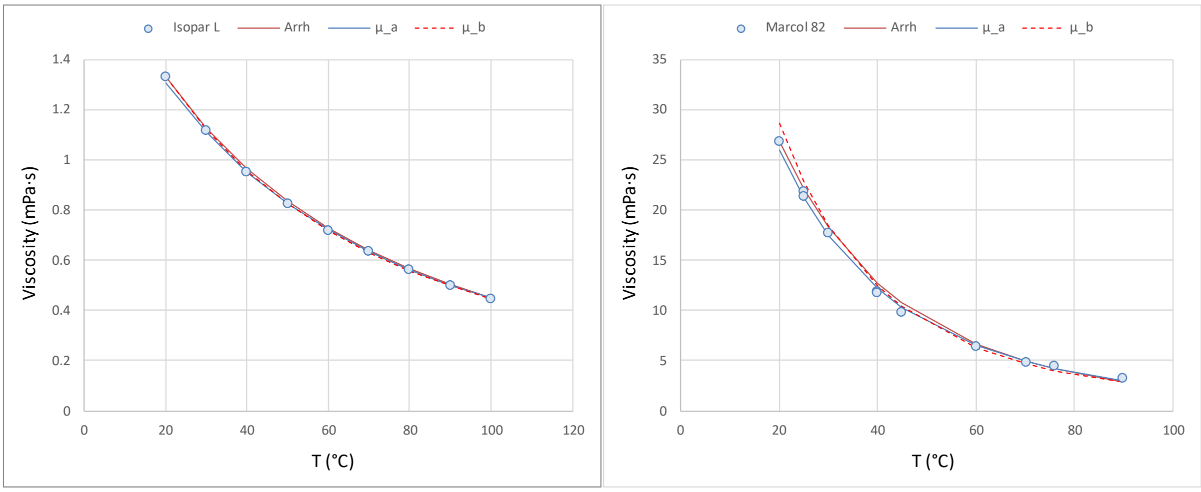


Figure 5.3 Viscosity of white oils Isopar L and Marcol 82 versus temperature, fitted with traditional Arrhenius type equation and the single parameter model in Eq. (5.8) denoted ‘Arrh’ and ‘μ_a’ in the legend. The T_C used is the average value computed for all experimental data points. The ‘μ_b’ line represents a similar method, but with a and b fitted by least square method to the Isopar L data and applied on the Marcol 82 data (fitted (a, b)=(0.010087, 1355.7)).

Example: defining properties for oil component named OIL1.

```
cmpprop
*  Name Prefc Bo density Co    PrefV TrefV°C visc  BTo   TCo    BPo
OIL1 100   1.0  0.75  1e-4  1.     90.    0.90  1500. 273.15 0.0016 /
/
```

35. cmpprop GAS

GAS properties.

Section: **propsect**

GAS properties are in two records, each terminated with ‘/’.

The first GAS record:

Item	Name	Description	Default
1	*cname*	Component name for identification. An empty record terminates the keyword.	
2	*Pref*	Reference pressure for compressibility, Units: user defined, e.g. [bar]	100 bar
3	*Bf*	Volume factor = Vres/Vsurface	0.01
4	*density*	Density at *Pref*. Units: user defined, density units, e.g. [g/ml]	0.12 g/ml
5	*compr*	Compressibility at *Pref*. Units: user defined inverse pressure units, e.g. [atm^{-1}]	$1.0\cdot10^{-2}$ bar^{-1}
6	*rgo0*	Solubility ratio (gas volume at standard condition to oil at reservoir conditions, *Pref*.). Units: e.g. [Sm^3 gas/Rm^3 oil]	0
7	*rgop*	Pressure dependency of gas solubility in oil. Units: e.g., [Sm^3 gas/Rm^3 oil /bar]	0
8	*Kwo*	Partitioning between water and oil at *Pref*. Rs_W/Rs_o in e.g. units [Sm^3 gas/Rm^3 water]/[Sm^3 gas/Rm^3 oil]	0

The second GAS record:

Item	Name	Description	Default
1	*VPref*	Reference pressure used in viscosity. Units: user defined, e.g. [bar]	100 bar
2	*VTref*	Reference temperature used in viscosity. Units: [°C]	90°C
3	*Visc*	Viscosity at *Vpref* and *Vtref*. Units: [cp] =[mPa·s]	0.015 cp
4	*BTvisc*	Temperature parameter in viscosity. If zero, the value is estimated from viscosity internally. Eq.(5.7). Units: [°K]	100°K
5	*TCvisc*	Second temperature viscosity parameter. Units [ºC] If a zero value is entered, *TCvisc* is computed internally using Eq.(5.7). This option requires also *BTvisc* to be zero.	273.15°
6	*BPvisc*	Pressure parameter in viscosity. Units: user defined, e.g. [bar^{-1}]	0.002 bar^{-1}
7	*viscl*	Apparent liquid viscosity of dissolved gas, Units: [cp]	0.1
8	*BTviscl*	Temperature parameter in apparent liquid viscosity. If *BTviscl* $\leq$ 0, the value is estimated from viscosity internally using Eq.(5.7). Units: [°K]	0
9	*TCviscl*	Second temperature viscosity parameter. Units [ºC]	273.15°

		Default value (273.15°) corresponds to the traditional Arrhenius equation used for oils. If a zero value is entered, *TCviscl* is computed internally using Eq.(5.7). This option requires also *BTviscl* to be zero.	
10	*BPviscl*	Pressure parameter in apparent liquid viscosity. Units: user defined, e.g. [bar^{-1}]	0.0015 bar^{-1}

The gas representation in the current version of the program is simplistic. The volume/pressure relation is only approximate within a narrow range in the relative pressure. It is also assumed that the gas density is the same for dissolved gas and gas present in the gas phase. If two or more gas components are present, partitioning of individual gas components between gas and liquid phase(s) are not properly computed in this version.

The gas phase viscosity is computed according to (5.4), and so is its apparent liquid viscosity. The liquid phase viscosity with dissolved gas is finally computed by Eq. (5.6).

Example: defining properties for gas component named CH4.

```
cmpprop
*   Name Prefw Bdg density Cg   rgo0 rgop   rgw0 rgwp
CH4  100   0.01  0.12  0.01 .50  0.005  .01  0.0001 /   GAS component - 1st record
*   PrefV Tref°C visc  BTg   TCg    BPg    viscl BTgl TCgl   BPgl
1.    90     0.012  200. 273.15 0.0025 0.12  0    273.15 0.001 /
/
```

36. cmpprop SURFACTANT

SURFACTANT properties.

Section: **propsect**

SURFACTANT properties are in three records, each terminated with '/'. The first SURFACTANT record defines volumetric properties:

Item	Name	Description	Default
1	*cname*	Component name for identification. An empty record terminates the keyword.	-
2	*Pref*	Reference pressure for compressibility, Units: user defined, e.g. [bar]	1 bar
3	*Bf*	Volume factor = Vres/Vsurface	1
4	*density*	Density at *Pref*. Units: user defined density units, e.g. [g/ml]	1 g/ml
5	*compr*	Compressibility at *Pref*. Units: user defined inverse pressure units, e.g. [bar^{-1}]	0
6	*iads*	Flag indicating adsorption model. Options: 0: No adsorption 1: Langmuir type adsorption, equilibrium model.	0

The second SURFACTANT record defines surfactant molecular properties.

Item	Name	Description	Default
1	*Mw*	Molecular weight, g/mol.	400
2	*ionicflag*	Ionic valence (-1: anionic, 0: nonionic, 1:cationic). Currently not used.	-1
3	*IFT0*	Reference oil/water interfacial tension at zero surfactant concentration. Units: [mN/m]=[dynes/cm]	20

The third SURFACTANT record defines a table of oil/water interfacial tension [mN/m] versus aqueous surfactant concentration. Note that blank lines before or in the table must be avoided. The table is terminated with a slash (/).

Column	Name	Description	Default
1	C_s^w	Aqueous surfactant concentration [volume fraction].	-
3	*IFT*	Interfacial tension [mN/m]	-

A fourth SURFACTANT record is read if surfactant adsorption is activated (*iads*=’alm’).

Item	Name	Description	Default
1	*b*	Adsorption parameter. [vol. fraction]. (Default value: $bc/(1+bc)$ = 0.8 at c=0.0001 (100 ppm), i.e., 80% of Q_m is adsorbed).	40000
2	*Qm*	Maximum adsorption capacity. Units [pv fraction] (Default value corresponds to approx. 0.2 mg/g at 20% porostity).	0.002

The surfactant adsorption is modelled by a Langmuir type equation, se Eq. (5.17) for polymer. If more than one surfactant is specified, competitive adsorption is assumed, see polymer Eq. (5.18), but this method is currently deactivated.

Example, surfactant input for component with name ‘Surf1’

```
cmpprop
*  Surfactant *********************************************************************
*  name  Pref    Bf   density   compr  iads
Surf1  100   1.0      1.0     0      1  /
*  Mw(g/mol)  ionicflag   IFT0(mN/m)
400.       -1          20.0 /
*  Cs        IFT
0         20.
0.000001   0.1 0.000010  0.01 0.000100  0.001
/
*  b      Qm   (adsorption)
100000 0.00005  /
/ ends cmpprop keyword
```

37. cmpprop POLYMER

POLYMER properties.

Section: **propsect**

POLYMER properties are in four records, each terminated with ‘/’.

1st POLYMER record defines volumetric properties			
Item	Name	Description	Default
1	*cname*	Component name for identification. An empty record terminates the keyword.	-
2	*Pref*	Reference pressure for compressibility, Units: user defined, e.g. [bar]	1 bar
3	*Bf*	Volume factor = V(*Pref*)/Vsurface	1
4	*density*	Density at *Pref*. Units: user defined density units, e.g. [g/ml]	1 g/ml
5	*compr*	Compressibility at *Pref*. Units: user defined inverse pressure units, e.g. [bar^{-1}]	0
6	*iads*	Flag indicating adsorption model. Old number notation in parenthesis. Options: n (0): No adsorption alm (1): Langmuir type adsorption, equilibrium model for mass adsorption. Mw of new adsorbed polymer equals polymer Mw in solution. Thus, adsorbed polymer Mw will only change when adsorption increases. Mw of released polymer equals adsorbed polymer Mw. almt : Langmuir type adsorption, equilibrium model with Mw equal for adsorbed polymer and polymer in solution. almir (2): Langmuir type, irreversible adsorption. almkin (3): Kinetic Langmuir type adsorption atab (4): Tabular adsorption atabkin (5): Kinetic tabular adsorption almasp: Langmuir type retention for associative polymer almkinasp: Kinetic LM type retention for associative polymer	n
7	*Pvismod*	Flag indicating which shear thinning model to use. Options: 1 Meter’s equation. 2 Carreau model with constant parameters 3 Carreau with variable parameters	1

2nd POLYMER record defines viscosity at low shear rate.			
Item	Name	Description	Default

1	*eta*	Reference intrinsic polymer viscosity $[\mu]_0$ in units [cm^3/g]. Typical range of values for good viscosifyers: 3000-5000 cm^3/g.	4000
2	*hug1*	The Huggin's constant, reported to be 0.4±0.1 for a range of polymers in good solvents. Units: [dimensionless]	0.4
3	*hug2*	Third viscosity parameter, k''. Units: [dimensionless] Note: entering a negative value will trigger the Martin equation.	0.1
4	*Tref*	Reference temperature T_0 for temperature dependent intrinsic viscosity $[\mu]$. Units: [°C]	20°C
5	*Tfact*	Temperature dependency factor B_{Th} for intrinsic viscosity $[\mu]$. Units: [°C]	0.005
6	*Mw*0	Polymer molecular weight [10^6 g/mol]	1
7	*alfaMw*	Exponent for dependency of $[\eta]$ om M_w. Typical range (0.5-1).	0.7

The third POLYMER record depends on the selected viscosity model, *Pvismod*. It defines the shear thinning parameters for the Meter's equation[3] or the Carreau model.

Pvismod = 1:

3rd Record read only if *Pvismod* = 1 : Meter's equation.

Item	Name	Description	Default
1	*palfa*	Shear thinning exponent P_α, possible range (1,2) where 1 corresponds to Newtonian behaviour. Typical values may be 1.75 for xanthan polymers and 1.3 for starch.	1.75
2	*ghf*	The shear rate $\dot{\gamma}_{hf}$ at which the reduced polymer viscosity (μ_p-μ_s) is one half of (μ_{p0}-μ_s). Units [sec^{-1}].	20

Pvismod = 2:

3rd Record read only if *Pvismod* = 2 : Carreau model.

Item	Name	Description	Default
1	*Carr_na*	Shear thinning exponent *n*, possible range (0,1) where 0 corresponds to Newtonian behaviour. Typical values may be 0.2-0.6 depending on concentration and molecular weight.	0.3
2	*Carr_la*	Polymer relaxation time λ_1 describing the transition from Newtonian to shear thinning fluid. Units: [sec]	1
3	*Carr_x*	Tuning parameter for the transition from Newtonian to shear thinning fluid. A higher value results in a sharper transition. Typical range: 1-2. Units: [dimensionless]	1

Pvismod = 3:

3rd Record read only if *Pvismod* = 3 : Carreau model extended.

Item	Name	Description	Default
1	*Carr_na*	1st parameter a_n for shear thinning exponent n	0.3
2	*Carr_nb*	2nd parameter b_n for shear thinning exponent n	0.8
3	*Carr_la*	1st parameter λ_a for the polymer relaxation time λ_1 describing the transition from Newtonian to shear thinning fluid. Units: [sec]	1
4	*Carr_lb*	Exponent λ_b for the polymer relaxation time λ_1 describing transition from Newtonian to shear thinning fluid.	0.00029
5	*Carr_x*	Tuning parameter for the transition from Newtonian to shear thinning fluid. A higher value results in a sharper transition. Typical range: 1-2. Units: [dimensionless]	1

A fourth POLYMER record is read if polymer adsorption is activated (*iads* $\geq$ 1). The record must end with a slash '/'.

4th Record read only if *iads* = alm, almt, almir : Langmuir type adsorption.

Item	Name	Description	Default
1	*b*	Adsorption parameter. Units: inverse of concentration units. The value should be chosen to get an appropriate ratio $bc/(1+bc)$, which denotes the fraction of Q_m adsorbed at concentration c.	-
2	*Qm*	Maximum adsorption capacity. Units [pv fraction] for volumetric components.	-
3	*ScaleQmF*	Flag indicating if scaling of *Qm* with porosity and permeability should be used. Options: 0: No scaling 1: *Qm* scales with $(\phi/k)^{\alpha}$.	0
4	*irflag*	Flag indicating different adsorption parameters should be used for each *rocktype*: 0: A single set of adsorption parameters uses for all rocktypes 1: This record is read *nrock* times. Item 3 and 4 are only read the first time. Each record must end with a slash '/'.	0

4th Record read only if *iads* = almkin : Kinetic Langmuir type adsorption.

Item	Name	Description	Default
1	*ka*	Adsorption parameter. [pv vol. fraction/hr].	-
2	*kd*	Desorption parameter. [pv vol. fraction/hr].	-
3	*Qm0*	Maximum adsorption capacity. Units [pv fraction] for volumetric components.	-

4	*rfg*	Shear rate dependency for adsorption/desorption rate. Botha *ka* and *kd* are multiplied with factor $f = (1+rfg\cdot gam^{0.5})$, where *gam* is the shear rate in $\sec^{-1}$. (Currently not working).	0
5	*TFlag*	Flag indicating if *ka* and *kd* should change with temperature in the same way as molecular diffusion Options: 0: No scaling 1: Both *ka* and *kd* scale with $\frac{T\mu_0}{T_0\mu_w}$	0
6	*T0*	Reference temperature, T_0, for scaling adsorption kinetics, [°K]	293.15
7	*my0*	Reference viscosity μ_0 for scaling, Units: [cp] = [mPa·s]	1.0
8	*ScaleQmF*	Flag indicating if scaling of adsorption capacity *Qm* with porosity and permeability should be used (record 5). Options: 0: No scaling 1: *Qm* scales with $(\phi/k)^{\alpha}$: $Q_m = Q_{m0}\left(\frac{k_{ref}}{\phi_{ref}}\cdot\frac{\phi}{k}\right)^{\alpha}$.	0
9	*irflag*	Flag indicating different adsorption parameters should be used for each *rocktype*: 0: A single set of adsorption parameters uses for all rocktypes 1: This record is read *nrock* times. Only the three first items are read from the repeated records. Each record must end with a slash '/'.	0

4th Record read only if *iads* = almasp : LM type retention for associative polymer.

Item	Name	Description	Default
1	*b*	Retention parameter. Units: inverse of concentration units. The value should be chosen to get an appropriate ratio $bc/(1+bc)$, which denotes the fraction of Q_m adsorbed at concentration *c*.	-
2	*Qm0*	Retention capacity Q_m at shear rate 1 $\sec^{-1}$. Units [pv fraction] for volumetric components.	-
3	*na*	Shear dependency index for *Qm*. Suggested range: 0.3-0.5	0
4	*S0p*	Specific surface area for polymer. Suggested range: 4000-7000 m^2/ml polymer. Units: [m^2/ml]	5000
5	*irflag*	Flag indicating different adsorption parameters should be used for each *rocktype*: 0: A single set of retention parameters used for all rocktypes 1: This record is read *nrock* times. Only the three first items are read from the repeated records. Each record must end with a slash '/'.	0

4th Record read only if *iads* = almkinasp : Kinetic LM retention associative polymer

Item	Name	Description	Default
1	*ka*	Adsorption parameter. [pv vol. fraction/hr].	-
2	*kd*	Desorption parameter. [pv vol. fraction/hr].	-
3	*Qm0*	Retention capacity Q_m at shear rate 1 sec^{-1}. Units [pv fraction] for volumetric components.	-
4	*na*	Shear dependency index for Qm. Suggested range: 0.3-0.5	0
5	*rfg*	Shear rate dependency for adsorption/desorption rate. Botha *ka* and *kd* are multiplied with factor $f = (1+rfg\cdot gam^{0.5})$, where *gam* is the shear rate in sec^{-1}.	0
6	*S0p*	Specific surface area for polymer. Suggested range: 4000-7000 m^2/ml polymer. Units: [m^2/ml]	5000
7	*TFlag*	Flag indicating if *ka* and *kd* should change with temperature in the same way as molecular diffusion Options: 0: No scaling 1: Both *ka* and *kd* scale with $\frac{T\mu_0}{T_0\mu_w}$	0
8	*T0*	Reference temperature for scaling adsorption kinetics, [°K]	293.15
9	*my0*	Reference viscosity for scaling, Units: [cp] = [mPa·s]	1.0

5th Record read if *ScaleQmF* = 1 . If *irflag*=1 (activated input for all rocktypes), the record should be repeated nrock times, each time terminated with a slash, '/'.

Item	Name	Description	Default
1	*Q_kref*	Reference permeability for scaling, k_{ref}. Units: user defined, e.g. [mD]	-
2	*Q_firef*	Reference porosity for scaling, ϕ_{ref}. Units: volume fraction.	0.2
3	*Q_alfa*	Scaling exponent α: $Q_m = Q_{m0}\left(\frac{k_{ref}}{\phi_{ref}}\cdot\frac{\phi}{k}\right)^{\alpha}$	0.5

The concentration dependent polymer viscosity at low shear rate is given by

$$\mu_{p0} = \mu_s\left[1 + \nu c_p + k'\nu^2 c_p^2 + k''\nu^3 c_p^3\right] \qquad (5.9)$$
$$\mu_{sp0} = \frac{\mu_{p0}}{\mu_s} - 1 = \nu c_p\left[1 + \nu c_p\left(k' + \nu c_p k''\right)\right],$$

or alternatively, if a negative k'' has been entered, by the Martin eq.:

$$\mu_{p0} = \mu_s\left[1 + vc_p e^{vc_p k'}\right] \qquad (5.10)$$
$$\mu_{sp0} = vc_p e^{vc_p k'},$$

where c_p is the polymer concentration, μ_s is the solvent (e.g., water) viscosity, μ_{sp0} is the specific viscosity at low shear rate and v is a dimensionless form of the intrinsic viscosity [cm^3/cm^3] accounting for eventual changes in temperature and molecular weight:

$$v = [\mu]_0 \left(\frac{M_w}{M_{w0}}\right)^{\alpha_{Mw}} \rho_p\left(1 - B_{Th}(T - T_0)\right), \qquad (5.11)$$

With *Pvismod*=1, the shear rate dependent polymer viscosity is computed with the Meter's method[3]:

$$\mu_p = \mu_s + \frac{\mu_{p0} - \mu_s}{1 + \left(\frac{\dot{\gamma}}{\dot{\gamma}_{hf}}\right)^{P_\alpha - 1}}, \qquad (5.12)$$

The Carreau model can be written:

$$\mu = \mu_s + \frac{(\mu_0 - \mu_s)}{(1 + (\lambda_1\dot{\gamma})^x)^{n/x}} = \mu_s\left(1 + \mu_{sp}\right), \qquad (5.13)$$
$$\mu_{sp} = \frac{\mu_{sp0}}{(1+(\lambda_1\dot{\gamma})^x)^{n/x}},$$

where the shear thinning exponent n is related to the Power law index n_{pl} by $n = 1\text{-}n_{pl}$ and parameter λ_1 indicates the polymer relaxation time defining the transition from Newtonian to shear thinning behaviour. The implementation of variable relaxation time in IORCoreSim takes two parameters:

$$\lambda_1 = \lambda_a \left(\frac{\eta_s \eta_{sp0} M_w}{10^6 C_p T}\right)^{\lambda_b}. \qquad (5.14)$$

The theoretical constants are $\lambda_a = 0.000144$ and $\lambda_b = 1.0$ with units [cp] for viscosity, [g/mol] for M_w and [g/ml] for c_p.

The shear thinning index n is modelled with:

$$n = 1 - n_{pl} = 1 - \frac{1}{1+\left(a_n[\eta]C_p\right)^{b_n}}. \qquad (5.15)$$

where

a_n : inverse value of $[\eta]C_p$ at which the Power law slope is -0.5. Units: [cp^{-1}].

b_n : exponent determining how fast the transition occurs from Newtonian (n = 0) to maximum shear thinning (n = -1). Example of values: a_n = 0.06, b_n = 0.8 (HPAM).

Starting with IORCoreSim version 1.307, the shear rate for the polymer containing water phase inside the porous medium is computed by:

$$\dot{\gamma} = \alpha_c \cdot \frac{4|v_w|}{R_{pw}}, v_w = \frac{u_w\sqrt{\tau}}{\phi S_w E_{pva} E_{pv0}}, \qquad R_{pw} = \sqrt{\frac{8Kk_{rw}\tau E_{pva}}{\phi S_w E_{pv0}}}, \tag{5.16}$$

$$E_{pv0} = 1 - IPV_0, \qquad E_{pva} = \frac{1}{\sqrt{R_{RF}}}$$

$$\dot{\gamma} = \alpha_c \cdot \frac{4|u_w|}{E_{pva}\sqrt{8Kk_{rw}\phi S_w E_{pva} E_{pv0}}}$$

where α_c is a constant for the formation in typical range 2-5, and is input via keyword rockshear. IPV_0 represents the fraction of the pore volume inaccessible for polymer (polyipv) and R_{RF} the residual resistance factor due to the pore volume available (E_{pva}) after subtracting the fraction effectively occupied by adsorbed polymer (polyrkf). In previous version of IORCoreSim, the effect of E_{pva} on pore flow velocity was ignored and $R_{pw} = \sqrt{8\,Kk_{rw}\tau/\phi S_w E_{pv0} R_{RF}}$ was used for the effective capillary tube radius available for the water phase. The effects of the changes are an increase in R_{pw}, reduced effect of polymer depletion layer with higher apparent viscosity at lower shear rates and a small increase in the computed in-situ shear. A corresponding correction of the elongational model results in only minor changes in simulated results at higher shear rates.

The Langmuir type adsorption of polymer is given by

$$A_p = \frac{bC_p^w Q_m}{1 + bC_p^w}, \tag{5.17}$$

where C_p^w is the volumetric polymer concentration relative to water. The adsorption capacity is optionally scaled by:

$$Q_m = Q_{m0}\left(\frac{k_{ref}}{\phi_{ref}} \cdot \frac{\phi}{k}\right)^{\alpha}. \tag{5.18}$$

Equilibrium adsorption, if more than one polymer is present, Eq. (5.18), is solved by iterations (Newton Raphson).

Example: defining properties for a polymer component named xanthan.

```
cmpprop
*  name      Pref   Bf  density  compr iads Pvismod
xanthan  100     1   1        0      n     3 / 1st record
*  eta    hug1  hug2  Tref  Tfact
4000  0.4   0.1   20    0.005 /   2nd record, viscosity at low shear
*  meter_P  meter_gh
1.75      20  /                 3rd record, shear thinning
/
```

38. cmpprop SUBSTRATE

SUBSTRATE properties

Section: **propsect**

One record terminated with '/':

Item	Name	Description	Default
1	*cname*	Component name for identification. An empty record terminates the keyword.	-
2	*Pref*	Reference pressure for compressibility, Units: user defined, e.g. [bar]	1
3	*Bf*	Volume factor = Vres/Vsurface	1
4	*density*	Density at *Pref.* Units: user defined density units, e.g. [g/ml]	1
5	*compr*	Compressibility at *Pref.* Units: user defined inverse pressure units, e.g. [bar^{-1}]	0
6	*kow*	Oil/water partitioning coefficient, range 0 to 1. $k_{ow,kc} = C_{o,kc}/(C_{w,kc} + C_{o,kc})$. Units: [volume ratio]	0

Example: defining properties for substrate component named S1.

```
cmpprop
*name Pref Bf Sdensity compr kowr
S1   1    1  1        0     0.5 / S1 is partitioning between oil and water
/
```

39. cmpprop BACTERIA

BACTERIA properties.

Section: propsect

The number of records for the BACTERIA input will depend on the input below. Each record must end with a slash '/'. The first record reads:

Item	Name	Description	Default
1	*cname*	Component name for identification. An empty record terminates the keyword.	-
2	*Pref*	Reference pressure for compressibility. Units: user defined pressure units, e.g. [bar]	1 bar
3	*Bf*	Volume factor = Vres/Vsurface	1
4	*density*	Density at *Pref.* (of dry bacteria). Units: user defined density units, e.g. [g/ml]	1 g/ml

5	*compr*	Compressibility at *Pref*. (dry bacteria) Units: user defined inverse pressure units, e.g. [bar^{-1}]	0
6	*Dpmean*	Average bacteria size (diameter). Units: [μm]	2
7	*cwater*	Water content in cells, volume fraction. Note: *cwater* and *Dpmean* (item above) are used to convert dry biomass into numbers of cells per ml.	0.8
8	*Xm*	Maximum dry cell concentration, volume fraction.	0.001
9	*mx*	Exponent used in cell growth limitation when approaching *Xm*	0.5
10	*iads*	Adsorption flag, options: 0: no adsorption 1: Langmuir type reversible adsorption	0
11	*itemp*	Temperature flag, options: 0: temperature not considered 1: Temperature included in growth kinetics	0

Adsorption input

The next record (terminated with '/') is only read if *iads*=1.

Item	Name	Description	Default
1	*ba*	Adsorption parameter *b*. Unit [vol.fraction in water phase]	0
2	*Qa*	Maximum adsorption. Unit [pv fraction]	1

The adsorption of component *i* is given by

$$A_i = \frac{bC_i^w Q_a}{1+bC_i^w} , \qquad (5.19)$$

where C_i^w is the volumetric biomass concentration relative to water. If more than one bacteria type is modelled, then competitive adsorption is assumed by weighting the numerator (bCQ_a).

Hints:

- If the product bC >>1, the adsorption will be close to Q_a.
- If bC is very small (bC<<1), then the numbers of pore volumes that must be injected before breakthrough is approximately bQ_a+1. In a 100% water saturated core and with bQ_a=10, roughly 11 pore volumes must be injected before breakthrough.

Temperature effects

The next record (terminated with '/') is only read if *TFlag*=1.

Item	Name	Description	Default

1	*LTopt*	Lower temperature limit for optimal growth range, [°C].	-
2	*UTopt*	Upper temperature limit for optimal growth range, [°C].	-
3	*Ltheta*	Temperature parameter θ_L below the optimal growth range. Valid range: $0 < \theta_L \leq 1.0$	1.0
4	*Utheta*	Temperature parameter above the optimal growth range. Valid range: $0 < \theta_U \leq 1.0$	1.0

The effect of temperature is included by multiplying the bacterial growth rate with a factor F_T given by:

$$F_T = \begin{cases} \theta_L^{T_{Lopt}-T}, & T < T_{Lopt} \\ 1, & T_{Lopt} \leq T \leq T_{Uopt} \\ \theta_U^{T-T_{Uopt}} & T > T_{Uopt} \end{cases} . \tag{5.20}$$

An example of F_T is plotted in Figure 5.4.

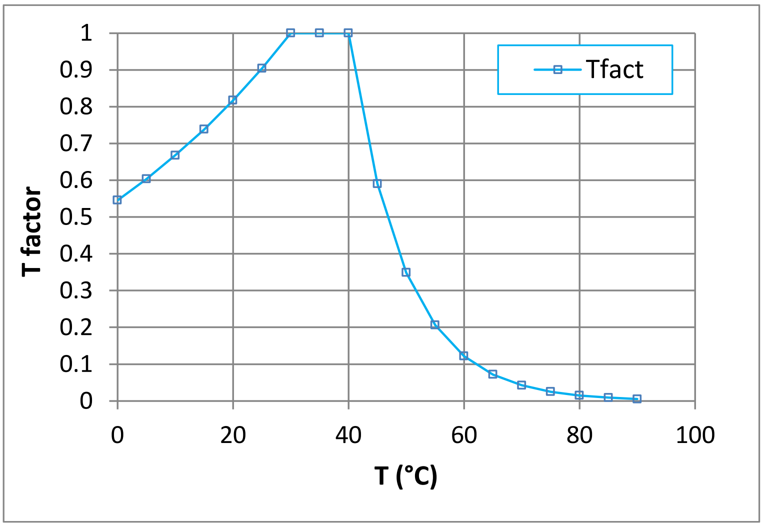


Figure 5.4 Effect of temperature on bacteria growth kinetics, (θ_L=0.98, T_L=30 °C, θ_U=0.9, T_U=40 °C).

Growth parameters

The next record describes the number of components involved in the microbial growth.

Item	Name	Description	Default
1	*nS*	The number of substrates considered. Note: any component type (not only those defined as SUBSTRATEs) can be used, e.g. oil and polymer.	0
2	*nNg*	The number of nutrient groups considered. For microbial growth to take place, at least one component must be present within each group.	0

3	*nNt*	The total number of nutrients considered. This should be the sum of the number of components within each group. If for example a component is defined in two groups, it must then be counted twice.	0
4	*nSPt*	The total number of products considered (components produced during microbial growth). Component products are associated with a substrate, and there may be several products from each substrate. This number is the sum of the product numbers considered for individual substrates. If for example methane production is defined for two substrates, then this component is counted twice.	

Substrate parameters

The substrate parameters for each substrate are entered in two records (each terminated with ‘/’), i.e., the two records are repeated as a pair *nS* times. The first record reads:

Item	Name	Description	Default
1	*Sname*	Name of the substrate component (used for identification). Note that a component defined as any type (except WATER) may be used as a substrate source here.	-
2	*mym*	Maximum specific growth rate (μ_m) with this substrate. Units: [days^{-1}]	-
3	*KS*	Half rate concentration in the Monod expression. Units: [volume fraction]	-
4	*Y*	Yield coefficient, [dry mass bacteria/mass substrate]	-
5	*nSP*	Number of component products from current substrate	-

The second record is omitted if $nSP = 0$ (no products). It contains $2{\cdot}nSP$ items, the product name and utilisation parameter repeated for all *nSP* products. A ‘/’ after the last component entrance terminates the record. For nSP=2, four items are read:

Item	Name	Description	Default
1	*Pname*(1)	Name of the 1st component produced (used for identification)	-
2	*Up*(1)	Utilisation coefficient, [mass produced/mass substrate]	-
3	*Pname*(2)	Name of the 2nd component produced (if $nSP \geq 2$)	-
4	*Up*(2)	Utilisation coefficient, [mass produced/mass substrate]. Read only if $nSP \geq 2$.	-

Nutrient parameters

The nutrient parameters within a group is entered in a single record (ended with ‘/’). The input is repeated for each group (*nNg* times).

Item	Name	Description	Default
1	*nN*(1)	Number of nutrients belonging to the first group	-
2	*Nname*	Name of the 1st nutrient component (used for identification). Note that components defined as any type (except WATER) may be used as a nutrient source here.	-
3	*KN*	Half rate concentration in the Monod expression. Units: [volume fraction] or if component is of category NUTRIENT, the user defined applies (e.g. ppm)	-
4	*UN*	Utilisation coefficient, [mass nutrient used/dry mass bacteria]. Note that mass nutrient may be in user defined units if the nutrient is a component defined as type NUTRIENT or TRACER.	-
-	-	Item 2-4 are repeated *nNg* times.	-

Decay parameters

This is the last record for BACTERIA properties.

Item	Name	Description	Default
1	*kd*	Specific decay constant (the fractional death rate of bacteria). Units: [days^{-1}]	0.01
2	*tave*	Expected average lifetime for current bacteria in [days]. If a positive number is entered, *tave* will be used in place of *kd* (item 1) for estimation of bacterial decay. Note: this model is at the testing stage.	0
3	*dName*	The dead bacteria may be specified as a component identified by *dName*. By default, no name is specified. No conversion factor is implemented here, i.e., one volume unit of dead bacteria becomes one volume unit of the new component.	none

Note: Specifying a component for the dead bacteria makes recirculation possible, by defining the dead bacteria as a substrate and/or nutrient source.

Example: defining properties for a bacteria component named B1 assuming two substrate sources and one nutrient group with two competing nutrients.

```
cmpprop
*name Pref  Bf  density  compr  Dpmean cwater  Xm    mx iads
 B1   100   1   1        0      2      0.8   .001   .25  1 /
*  b   Qmax
100000 0.0001 /  adsorption parameters (iads=1)
*  nS  nNg  nNt  nSPt   - total numbers of S, N-groups, N components and Products
2   1    2    3  /
*  Sname mym   KS   Y    nSP   - substrate 1
OIL1  .5   .01   0.5  2 /
*  Pname UP   -  product 1 from substrate 1
xanthan 0.2
*  Pname UP   -  product 2 from substrate 1
CH4  0.2 /  0.2 gram CH4/ gram OIL1 used
```

```
*   Sname mym   KS     Y    nSP    - substrate 2
S1     3.0    .0001 0.5 1 /
*   Pname UP    -   product 1 from substrate 2
CH4   0.2 /
*nNg=1  nutrient group 1
*nN[1]=2 number of nutrients in Nutrient group 1 (2 nutrients)
2
*Nname KN       UN (kcn=1,2) - name and parameters for 1st N in Ng1
 Noil  0.2      0.1
 N1    0.2      0.1 / 2nd N in Ng1

*kd   tave dName   - decay rate and no end product
0.1   0  /
/  end keyword cmpprop
```

40. cmpprop NUTRIENT

NUTRIENT properties.

Section: **propsect**

NUTRIENT properties are in one record terminated with ‘/’:

Item	Name	Description	Default
1	*cname*	Component name for identification. An empty record terminates the keyword.	-
2	*kow*	Oil/water partitioning coefficient, range 0 to 1. $k_{ow,kc} = C_{o,kc}/(C_{w,kc}+ C_{o,kc})$. Units: [volume ratio]	0
3	*cunit*	The unit associated with this component, will be used in printouts, e.g. ppm. Since this component does not occupy any volume, any unit and numbers may be used because no internal conversion is needed.	vfrac

Example: defining properties for two nutrient components named N1 and Noil.

```
cmpprop
*name kowr Unit
N1    0    ppm / represents external nutrient dissolved only in water
*name kowr Unit
Noil .999 ppm / represents nutrient from oil partitioning between oil/water
/
```

41. cmpprop TRACER

TRACER properties.

Section: **propsect**

TRACER properties are on one record terminated with ‘/’:

Item	Name	Description	Default
1	*cname*	Component name for identification.	-

		An empty record terminates the cmpprop keyword.	
2	*trtype*	Kind of tracer Note: this option is not used currently.	1
3	*iunit*	The unit associated with this component will be used in printouts, e.g. ppm. Since this component does not occupy any volume, any unit and numbers may be used because no internal conversion is needed. The units can be set with either an **integer** or a **unit keyword** as it appears in the list below. Options: 0 or v.frac: *C*: [v.frac], *V*: [user's input], *A*: [ml/ml PV] 1 or ppm: *C*: [ppm], *V*: [mg], *A*: [mg/l PV] 2 or mgpl: *C*: [mg/l], *V*: [mg], *A*: [mg/l PV] 3 or gpl: *C*: [g/l], *V*: [gram], *A*: [g/l PV] 4 or kgpm3: *C*: [kg/m^3], *V*: [kg], *A*: [kg/m^3 PV] 5 or m: *C*: [mol/l], *V*: [mol], *A*: [mol/l PV] 6 or meqpml: *C*: [meq/ml], *V*: [meq], *A*: [meq/ml PV] 7 or gp100ml: *C*: [g/100ml], *V*: [gram], *A*: [g/100 ml PV]	0
4	*iads*	Flag indicating adsorption model. Options: n : No adsorption alm : Langmuir type adsorption, equilibrium model. almir : Langmuir type, irreversible adsorption. almkin : Kinetic Langmuir type adsorption atab: Tabular adsorption aext: Adsorption activated, but model input is under another keyword.	n
5	*ireact*	Flag indicating 1st order decay. Currently not used.	0
6	*kow*	Oil/water partitioning coefficient, range 0 to 1. $k_{ow,kc} = C_{o,kc}/(C_{w,kc}+ C_{o,kc})$. Units: [volume ratio]	0

A second TRACER record is read if tracer adsorption is activated and different from 'aext':

Item	Name	Description	Default
1	*b*	Adsorption parameter. [vol. fraction]. (Default value: $bc/(1+bc) = 0.67$, i.e., 67% of Q_m is adsorbed at c=0.001 (1000 ppm)).	2000
2	*Qm*	Maximum adsorption capacity. Units [pv fraction] (Default value corresponds to approx. 10 µg/g for a 20% porosity rock).	0.0001

Example: defining properties for one tracer component named TR1.

```
cmpprop
*   name trtype iunit iads ireact
TR1   1       1     1     0   /   water tracer
*   b     Qm   (Langmuir adsorption parameters) 50% ads at 5 ppm, 90% at 50 ppm
```

```
0.2 50.0 /
/
```

42. Corey

Section: **propsect**

Description: Corey type relative permeability curve. Can be mixed with table format (see example, **krtab**). Parameters are entered for one phase at a time. The line is repeated until it is terminated by a slash (/) on a new line.

Variants of keyword:

Corey or Corey1:	Corey1 model (imbibition kr if hysteresis is activated)
Corey2:	Corey 2 model
Corey1e:	Extended Corey 1 model
Corey2e:	Extended Corey 2 model
Coreyxxd:	Drainage kr, with *Coreyxx* representing any of the variants listed above. The input of drainage kr requires the hysteresis option to be activated (hystsat).

Corey1 or Corey2 (Corey1d or Corey2d)

Item	Name	Description	Default
1	*ir*	Rock type (satnum), 1,..., *nrock*	-
2	*j*	Phase number: 0 or ‘w’: water phase 1, ‘o’ or ‘ow’: oil phase (oil/water limit) 2 or ‘g’: gas phase 3 or ‘og’: oil phase (oil/gas limit)	-
3	k_{rje}	End-point relative permeability	-
4	E_j	Corey exponent	-
5	S_{rj}	Residual saturation	-
6	*id*	Direction: 1=x-direction, 2=*y*-direction, 3=z-direction	1
7	*ik*	Relative permeability set number, must be between 1 and *mKrSet*.	1
8	*vint*	Value associated with current relative permeability set if interpolation is activated. This value could be log(N_c) or adsorbed amount of bacteria, see *iKrMode*. Alternative interpolation between two relperm sets can be entered in tabular form using keyword ncmisc.	0

Corey1e or Corey2e (Corey1ed or Corey2ed)

Item	Name	Description	Default
1	*ir*	Rock type (satnum), 1,..., *nrock*	-
2	*j*	Phase number: 0 or ‘w’: water phase 1, ‘o’ or ‘ow’: oil phase (oil/water limit) 2 or ‘g’: gas phase 3 or ‘og’: oil phase (oil/gas limit)	-
3	k_{rje}	End-point relative permeability	-
4	E_{1j}	Extended Corey exponent 1	-
5	E_{2j}	Extended Corey exponent 2	-
6	S_{rj}	Residual saturation	-
7	*id*	Direction: 1=x-direction, 2=*y*-direction, 3=z-direction	1
8	*ik*	Relative permeability set number, must be between 1 and *mKrSet*.	1
9	*vint*	Value associated with current relative permeability set if interpolation is activated. This value could be log(N_c) or adsorbed amount of bacteria, see *iKrMode*. Alternative interpolation between two relperm sets can be entered in tabular form using keyword ncmisc.	0

Notes:

- Directional relperm is optional. Relperm data must be specified for all phases and rock types in direction 1 (x).
- The oil relative permeability from oil/gas (k_{rog}) is currently not used. No three-phase model is implemented and k_{ro} in three-phase cases is obtained from the oil/water limit (k_{row}).

The Corey type 1 and 2 differ in the way the normalized saturation is computed:

$$k_{rj} = k_{rje} S_{jn}^{E_j}, \quad j = o, w, g \tag{5.21}$$

Corey1: $$S_{jn} = \frac{S_j - S_{rj}}{1 - S_{rj}},$$

Corey2: $$S_{jn} = \frac{S_j - S_{rj}}{1 - S_{wr} - S\_or}.$$

Note that the type 1 formula covers the saturation range $S_j = \left[S_{jr}, 1\right]$ Sj=[Sjr, 1] while type covers only the saturation with both phases mobile, e.g. for the water phase $S_w = [S_{wr}, 1 - S_{or}]$. For the type 2 formulae the k_r for higher saturations are obtained by linear interpolation, here exemplified for the oil phase:

$$\text{Corey2:}\ S_w < S_{wr} \ => k_{ro} = k_{roe} + (1 - k_{roe}) \cdot \frac{S_{wr} - S_w}{S_{wr}}. \qquad (5.22)$$

The extended Corey formula offers increased flexibility for better matching of experimental data. It is easy to use (the effect of changing the parameters is intuitive). If additional flexibility is needed, the LET formula or tabular format can be used. The extension is to allow a linear change of the Corey exponent over the relevant saturation interval:

$$k_{rj} = k_{rej} S_{jn}^{E_j}, \quad E_j = E_{1j} + S_{jn} E_{2j}, \quad j = o, w, g \qquad (5.23)$$

The normalized saturation is computed according to the Corey type (I or 2) defined in the keyword. There are two restrictions on the exponents, (1) the resulting exponent *Ej* must be positive and (2) if E_{2j} is positive the ratio $E_{1j}/E_{2j} > \exp(-2)$ or value 0.13534.

Example: two rock types and 2 phases.

```
corey
*   ir   j   kre   E    Sr      id
    1    o   0.8   2.5  0.15    /  oil, rock 1, id=1 is default
    1    w   0.3   2.0  0.12    /  water, rock 1
    2    o   0.7   3.0  0.10    /  oil, rock 2
    2    w   0.4   2.4  0.14    /  water, rock 2
/
```

43. diffusion

Section: **propsect**

Description: Component diffusion within the water and oil phases. Two records are read, each terminated with a slash '/'.

1st record Note: item 3-6 is not used if tortuosity keyword rtort is used.

Item	Name	Description	Default
1	*my0*	Reference viscosity in [cp], μ_o. Units: [cp] = [mPa·s]	-
2	*TD0*	Reference temperature [°C], T_{Do}. Units: [°C]	-
3	*tau*	Tortuosity of porous media [dimensionless], $\tau = (L_t/L)^2$. Only used if *m* < 1.0 (next item). Tortuosity data entered here will not be used if rock tortuosity keyword rtort is activated. Units: [dimensionless]	2
4	*m*	Cementation exponent. A positive value > 1.0 activates the alternative computation of saturation dependent tortuosity. Recommended ranges: 1.4 - 2.0 (sandstones) 2.0 – 2.6 (cabonates)	0

5	n_w	Saturation index for water. Only used if m > 1.0. Recommended ranges: 1.5 - 2.0 (water wet) 2.0 – 3.0 (mixed wet)	2.0
6	n_o	Saturation index for water. Only used if m > 1.0. Suggested range:	2.0
2nd record			
-	*D0(k)* Alternatively *name, D0(k)*	A list of bulk diffusion coefficients, $D_o(k)$, at reference condition from 1st record, starting with the first component which is not of type water or oil. Alternative input method is to indicate which component to read by its name, followed by its diffusion coefficient. This is repeated for all desired components. Missing components gets D_o defaulted to zero. Units: [cm²/s]	0

The bulk diffusion coefficient for component k dissolved in fluid phase L is:

$$D_{L,k} = D_{0,k}\frac{\mu_0}{\mu_L}\frac{T}{T_0}, \tag{5.24}$$

where temperature T_0 is in °K ($T_0 = T_{D0}$+273.15). The diffusive flux in porous media is computed

$$J_k = \frac{\nabla C_k^w (D_{0,k}\phi S_w)}{\tau_w} + \frac{\nabla C_k^o (D_{0,k}\phi S_o)}{\tau_o}. \tag{5.25}$$

If component is volumetric (occupies space, not a tracer), an opposite diffusive flux of solvent (oil or water) is assumed.

If the saturation dependent tortuosity is activated, (m>1), the tortuosity parameters in Eq. (5.25) are replaced with appropriate saturation dependent parameters computed as:

$$\begin{aligned} \tau_w &= \varphi^{1-m} S_w^{1-n_w} \\ \tau_o &= \varphi^{1-m}\left(\frac{S_o - S_{or}}{1 - S_{or}}\right)^{-n_o}. \end{aligned} \tag{5.26}$$

and diffusive flux becomes:

$$J_k = \nabla C_{w,k} D_{w,k} \phi^m S_w^{n_w} + \nabla C_{o,k} D_{o,k} \phi^m S_o^{n_o}. \tag{5.27}$$

Example: diffusive coefficients for two components.

```
diffusion
*   my0 TD0 tau
1.0  25  2.5 /
MeOH 0.8E-5 SO4 1.0E-5 /  cm²/sec
```

44. dispersion

Section: **propsect**

Description: Component dispersion within the water and oil phases. A single record terminated with a slash, '/', is repeated nrock. An empty record (no data before the slash '/') ends the input and any remaining rock types are assigned the last dispersion parameters read.

1st record	Note: item 3-6 is not used if tortuosity keyword rtort is defined.		
Item	Name	Description	Default
1	*lamDL*	Longitudinal dispersion parameter, λ_{dl} Units: [cm]	0
2	*lamDT*	Transversal dispersion parameter, λ_{dt} Units: [cm]	0

Dispersion is only considered for species assigned a diffusion constant. The molecular diffusion and dispersion are solved together. Note that while the molecular diffusivity is treated as a scalar depending on temperature, porosity and saturation, the dispersion depends only on the flow and is handled by two parameters representing longitudinal and transversal effects (should be a tensor). The total diffusive flux of specie *k* in the *x*-direction (including dispersion) is computed:

$$J_{k,x} = \frac{dC_k}{dx} D_{k,x} A_{yz}, \qquad (5.28)$$

$$D_{k,x} = \frac{D_{0,k}\phi S_w}{\tau} \cdot \frac{\mu_0 T}{\mu T_0} + \{\lambda_{dl}|u_{wx}| + \lambda_{dt}\sqrt{u_{wy}^2 + u_{wz}^2}\}.$$

The molecular diffusion, D_0, and the tortuosity, τ, are described under the *diffusion* keyword.

Notes:

- If insufficient data is found, parameters for the remaining rock types are assigned the last valid input.

Example: dispersion coefficients for two rock types appropriate for lab scale.

```
dispersion
*   lamDL   lamDT
0.1     0.05   /
0.05    0.025 /
/
```

45. dsprop

Section: **propsect**

Description: Dissolved species properties. Effect on phase viscosity and density.

The first record contains two key parameters, terminated with a slash (/).

Item	Name	Description	Default
1	*visFlag*	Integer flag indicating how viscosity is calculated when parameters for more than one species are specified. Possible range: 0-3, see the description below. Note: If *visFlag* is zero, viscosity calculations are skipped.	3
2	*phase*	Single letter indicating for which phase the parameters applies. Options: *w* : water phase *o* : oil phase	*w*

The second record, terminated with a slash (/), is repeated any number of times for new species. An empty record terminated with a slash (/) ends the keyword.

Item	Name	Description	Default
1	*cmp*	Component name, required	-
2	*avis*	First viscosity parameters a_k for dissolved specie *cmp*. Units: inverse *cmp* units, (*cmp* units)$^{-1}$.	0
3	*bvis*	Second specie viscosity parameter, b_k. Units: (*cmp* units)$^{-0.5}$.	0
4	*densf*	Density parameter, d_k, Units: (*cmp* units)$^{-1}$.	0

The effect of a dissolved specie *k* on viscosity of phase *l* is computed:

$$\mu^l = \mu_0^l \cdot \frac{1+a_k^l C_k^l}{\left(1+b_k^l {C_k^l}^2\right)^2}, \quad l = o, w. \tag{5.29}$$

The viscosity model fits viscosity data for different salts over large concentration ranges. Examples of that are given in Figure 5.5. A list of parameters for different salts is provided in Table 5.1. If the geochemical model is used, one can use the parameter values estimated for individual ions in Table 5.3. Note that computed properties for mixed brines are approximated values, because the effect of ion complexes that may form is not accounted for in the model.

If parameters for more than one specie is read, then the *visFlag* parameter determines which option is used to compute the combined effect. There are three options (phase indicator omitted):

1. $\mu = \mu_0 \cdot \frac{1+\sum_k a_k C_k}{(1+\sum_k (b_k C_k)\cdot C_t)^2}, \; C_t = \sum_k C_k$ (5.30)

2. $\mu = \mu_0 \cdot \prod_k f_k\,,\quad f_k = \frac{1+a_k C_k}{\left(1+b_k C_k^2\right)^2}$
3. $\mu = \mu_0 \cdot \frac{1+\sum a_k C_i}{(1+b_c|b_c|)^2}\,,\ b_c = \sum_k(\mathrm{sign}(b_k) C_k |b_k|^{0.5})$

The density of phase *l* is computed:

$$\rho^l = \rho_0^l \cdot \left(1 + \textstyle\sum_k d_k^l C_k^l\right),\quad l = o, w. \qquad (5.31)$$

Option 3 is correct in the two limits, 1) when two species with identical parameters are mixed in different ratios keeping the total (C_t) constant, the computed viscosity must be constant, and 2) if the parameters (*a* and *b*) of one specie in the mixture are zero, then the computed viscosity must be independent of the concentration of that specie. Options 2 or 3 should be used if there are very different concentration ranges and parameters ranges of involved species, e.g. because different concentration units are used.

Example: with two salt additives in units g/l.

```
dsprop
*   visFlag  phase
3          w /
*   name   avis      bvis       densf
NaCl  0.001419 -1.526E-06  0.000679 /
CaCl2 0.002449 -1.699E-06  0.000790 /
/
```

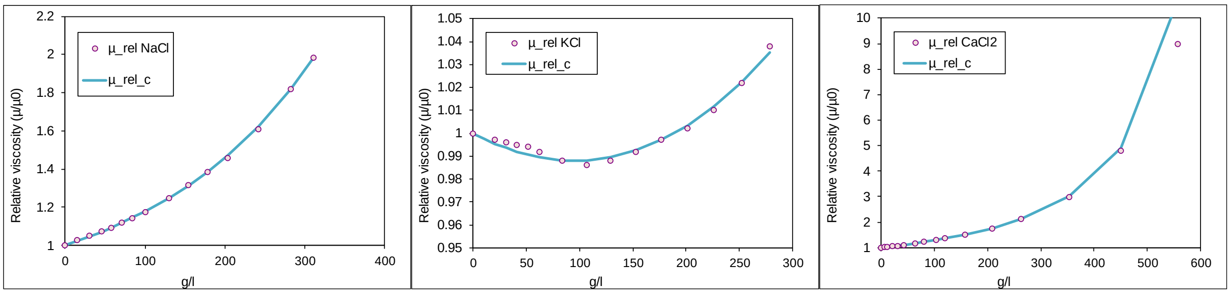


Figure 5.5 Examples of viscosity parameters fitted to different salts, from left, NaCl, KCl and $CaCl_2$. Literature data from Ref. [4].

Table 5.1 Viscosity and density parameters fitted to data from Ref. [4].

	Concentration unit: mol/L				Concentration units: g/L		
Component	Mw	a	b	d	a	b	d
HCl	36.461	0.0489	-0.00082	0.0174	0.00134	-6.20E-07	0.000478
NaOH	39.997	0.2346	-0.00746	0.0418	0.00587	-4.66E-06	0.001044
LiCl	42.394	0.1403	-0.00347	0.0234	0.00331	-1.93E-06	0.000551
NaCl	58.443	0.0829	-0.00521	0.0397	0.00142	-1.53E-06	0.000679
NaBr	102.894	0.0430	-0.00618	0.0785	0.00042	-5.84E-07	0.000763

KCl	74.551	-0.0188	-0.00375	0.0464	-0.00025	-6.74E-07	0.000622
MgCl2	95.211	0.4189	-0.02515	0.0743	0.00440	-2.77E-06	0.000781
CaCl2	110.986	0.2718	-0.02093	0.0877	0.00245	-1.70E-06	0.000790
SrCl2	158.526	0.2501	-0.02432	0.1365	0.00158	-9.68E-07	0.000861
BaCl2	208.236	0.2192	-0.01992	0.1768	0.00105	-4.59E-07	0.000849
NaHCO3	84.007	0.2194	-0.01552	0.0573	0.00261	-2.20E-06	0.000682
NaNO3	84.995	0.0453	-0.00745	0.0552	0.00053	-1.03E-06	0.000649
Na2SO4	142.037	0.4117	-0.04508	0.1204	0.00290	-2.23E-06	0.000847
KI	166.003	-0.0932	-0.01111	0.1197	-0.00056	-4.03E-07	0.000721
CH3COOH	60.052	0.1245	-0.00017	0.0082	0.00207	-4.62E-08	0.000136

Table 5.2 Viscosity and density parameters computed for individual ions assuming equal contributions from Na$^+$ and Cl$^-$ for NaCl.

	Concentration unit: mol/L			
Component	Mw	a	b	d
H$^+$	1.0079	0.0074	0.00005	-0.0024
Li$^+$	6.941	0.0988	-0.00052	0.0035
Na	22.9898	0.0415	-0.00130	0.0199
K$^+$	39.0983	-0.0602	-0.00063	0.0265
Mg^{++}	24.305	0.3360	-0.00746	0.0346
Ca^{++}	40.08	0.1889	-0.00525	0.0480
Sr^{++}	87.62	0.1672	-0.00701	0.0968
Ba^{++}	137.33	0.1363	-0.00475	0.1371
OH$^-$	17.0073	0.1932	-0.00253	0.0219
Cl$^-$	35.453	0.0415	-0.00130	0.0199
Br$^-$	79.904	0.0016	-0.00181	0.0587
I$^-$	126.9045	-0.0329	-0.00645	0.0932
NO3$^-$	62.0049	0.0038	-0.00252	0.0353
HCO3$^-$	61.0171	0.1779	-0.00783	0.0375
CH3COO$^-$	59.0445	0.1171	-0.00041	0.0106
CO$_3^{--}$	60.0092	0.4337	-0.00274	0.0619
SO$_4^{--}$	96.0576	0.3287	-0.01963	0.0806

46. elcond

Section: **propsect**

Description: Activates electric conductivity computations from aqueous composition and electric resistance for the porous medium. Electric boundary conditions should be specified using keyword elrbc.

The first record involves three items governing computation of formation resistivity, terminated with a slash (/).

Item	Name	Description	Default
1	*Tmod*	Integer flag that should indicate temperature model for brine conductivity. Currently, only Arp's equation is available and will be used whatever value is read.	0
2	*T0*	Reference temperature in °C for the conductivity input.	20
3	*ArchieF*	Integer flag indicating reading of Archie parameters. Options: 0: Archie parameters will not be read here. It should then be included using the ***rtort*** keyword. 1: cementation and saturation indexes will be read in the next record.	0

The second record is only read if *ArchieF* = 1. Then, 2·*nrock* values will be read from a single record terminated with a slash (/).

Item	Name	Description	Default
1, n_r	*m(ir)*	*nrock* (n_r) cementation indexes m_i, *i*=[1, *nrock*]	2
n_r+1, 2·n_r	*n(ir)*	*nrock* (n_r) saturation indexes, n_i,i=[1,*nrock*]	2

The third record contains species data for brine conductivity, terminated with a slash (/). One record contains data for a single component and is repeated for each specie contributing to the brine conductivity. An empty record terminated with a slash (/) ends the keyword.

Item	Name	Description	Default
1	*name*	Component name	-
2	*acw*[*k*]	First conductivity parameter for component *k*. Units: mS/cm/(*cmp* units)$^{0.5}$	0
3	*bcw*[*k*]	Second conductivity parameter for component *k*. Units: mS/cm/(*cmp* units)	0
3	*ccw*[*k*]	Third conductivity parameter for component *k*. Units: mS/cm/(*cmp* units)^2	0

The bulk conductivity for the water phase is computed:

$$\sigma_w = aC_t^{0.5} + bC_t + cC_t^2, \tag{5.32}$$

If a single component *k* is considered, then *a*=a_k=*acw*[*k*] etc., and the total concentration $C_t = C_k^w$ in the water phase. If more than one species is considered, the conductivity coefficients are weighted according to concentration and concentration exponents. The different terms are computed:

$$aC_t^{0.5} = sign(d)|d|^{0.5}, \text{where } d = \sum_k (|a_k|a_k C_k^w), \tag{5.33}$$

$$bC_t = \sum_k b_k C_k^w,$$

$$cC_t^2 = d|d|, \text{where } d = \sum_k (sign(c_k)|c_k|^{0.5} C_k^w).$$

The temperature correction for the brine conductivity is (Arps formula):

$$\sigma_w(T) = \sigma_w(T_0) \cdot \frac{T+21.5}{T_0+21.5}, \qquad (5.34)$$

with temperature *T* in °C. The brine resistivity $\rho_w = 1/\sigma_w$. The formation resistivity is computed using Archies exponents *m* and *n* for rock type *ir*:

$$\rho_f = \rho_w(\phi B_r)_i^{-m_{ir}} S_{w,i}^{-n_{ir}}, \ B_r = e^{n_{cr}(p-p_0)} \qquad (5.35)$$

Example: One salt additive in units g/l matched to concentration range 0-50 g/l, see Figure 5.6:

```
elcond
*   TRflag TR0, archieFlag;
1       22.5   2 /
*   cmp   mS/cm/conc
NaCl    1.37 1.264 0. /
/
```

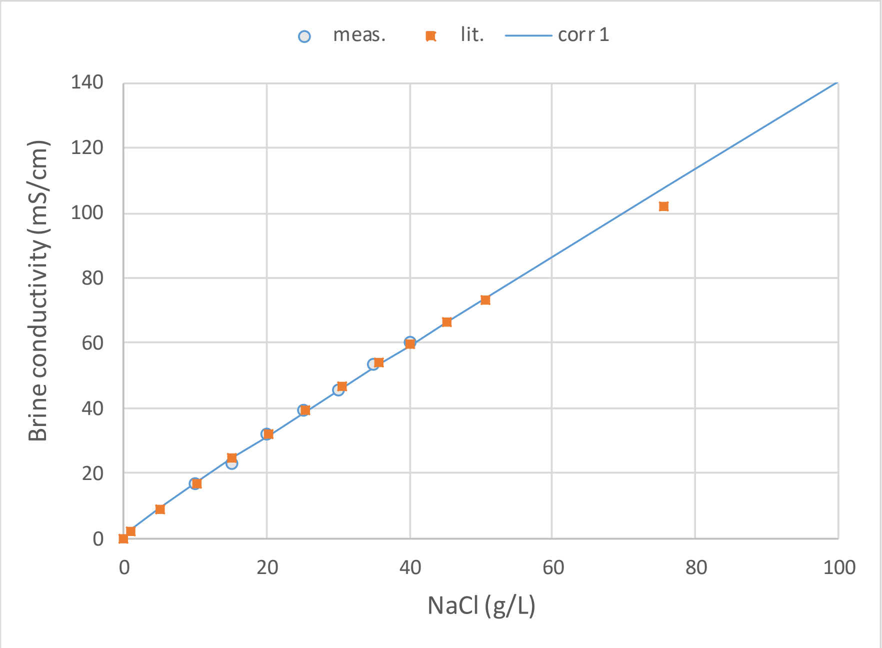


Figure 5.6 Example of conductivity parameters fitted to NaCl data, measured at 22.5 °C and literature data from Ref. [4] converted from 20 to 22.5 °C by Arps formula.

47. gelmod

Section: **propsect**

Description: Gelation model including reaction rate parameters and permeability reduction of the formed gel phase.

First record contains two parameters on a single record is read, terminated with a slash (/).

First record:

Item	Name	Description	Default
1	*gelmodel*	Integer indicating the gel model selected. Possible values: 0 : No gel model, reading remaining part of keyword is skipped. 1 : General gel model (currently the only one)	0

Gel model 1:

2nd record:

Item	Name	Description	Default
1	*gelname*	Name of the component representing gel	-
2	*ngvar*	Number of components that the gelation rate depends on, n_{gv}.	-

3rd record:

Item	Name	Description	Default
1	*gname*(*i*)	Names of the components involved in the gelation, i=1,*ngvar*. Note that the first name should represent the main component being converted to gel.	-

4th record:

Item	Name	Description	Default
1	*g_alfa*(*i*)	Gelation parameter $\alpha_{g,i}$ for component *i*, *i*=1,*ngvar*	-

5th record:

Item	Name	Description	Default
1	*g_beta*(*i*)	Gelation parameter β_i for component *i*, *i*=2,*ngvar*	1

6th record:

Item	Name	Description	Default
1	*g_Y*(*i*)	Yield coefficient $Y_{g,i}$ for component *i*, *i*=1,*ngvar*. $Y_{g,i}$ represents the consumption of component *i* per unit gel generated. Default values are 1.0 for the first and 0.0 for remaining components.	1, 0

7th record:

Item	Name	Description	Default
1	*rg*	Rate parameter in gel equation.	-

		Units: [days^{-1}]	
2	*Eag*	Activation energy. Units: [kJ/mol]	20
3	*gT0*	Reference temperature used in gel equation. Units: [°C]	20
4	*grFlag*	Gel retention flag, options: 0: A constant value for critical gel concentration is used. 1: A critical gel concentration for each *rocktype*. 2: Critical gel concentration scales with $(k/\phi)^{\alpha}$.	0

8th record with *grFlag*=0:

Item	Name	Description	Default
1	*rkfmod*	Indicates permeability reduction mode. Possible values: 1 : Only the water phase mobility is reduced. 2 : Total mobility is reduced (all phases)	-
2	*Cgcr*	Critical concentration for retention of gel. All gel above this critical concentration is irreversibly retained. Units: in gel component units (depends on component type)	-
3	*gSspm*	Specific surface area relative to gel matrix mass, S_{g0}. The units depend on the main gelation component (the first component in the 3rd record). The gel concentration can be expressed as [mass/volum]. A conversion factor is used for the volum part and the units for S_{g0} can be expressed: Units: [m²/mass gel] Examples: gel in [g/100 ml], S_{g0} units = [m²/g gel] gel in [g/L], S_{g0} units = [m²/g gel] gel in [mg/L], S_{g0} units = [m²/mg gel]	-

8th record with *grFlag*=1:

Item	Name	Description	Default
1	*rkfmod*	Indicates permeability reduction mode. Possible values: 1 : Only the water phase mobility is reduced. 2 : Total mobility is reduced (all phases)	-
2	*gSspm*	Specific surface area relative to gel matrix mass, S_{g0}. The units depend on the main gelation component (the first component in the 3rd record). The gel concentration can be expressed as [mass/volum]. A conversion factor is used for the volum part and the units for S_{g0} can be expressed: Units: [m²/mass gel]	-

3	*Cgcr*(*i*)	*i*=1,*nrock*. Critical concentration for retention of gel. All gel above this critical concentration is irreversibly retained. One value read for each *rocktype*. Units: in gel component units (depends on component type)	-

8th record with *grFlag*=2:

Item	Name	Description	Default
1	*rkfmod*	Indicates permeability reduction mode. Possible values: 1 : Only the water phase mobility is reduced. 2 : Total mobility is reduced (all phases)	-
2	*gSspm*	Specific surface area relative to gel matrix volume. Units: [m²/mass gel] (see explanation above)	-
3	*Cgcr*0	Reference critical concentration for retention of gel. $C_{gcr} = C_{gcr0}\cdot(k\cdot\phi_{g0}/k_{g0}\cdot\phi)^{\alpha_{gcr}}$, Units: in gel component units (depends on component type)	-
4	k_{g0}	Reference permeability for critical gel concentration in user defined Units, e.g., [mD].	1
5	ϕ_{g0}	Reference porosity for critical gel concentration. Units: [vol. frac.].	1
6	α_{gcr}	Exponent for calculating critical gel concentration.	0.5

Note:

- The component type representing gel and other variables in the gel equation can be user defined types. A suitable component type is tracer.
- The retention of gel component must be activated by setting the *iads* flag under component properties to 'aext', see e.g. tracer properties.

The gelation rate is:

$$\frac{dC_g}{dt} = r_g C_1^{\alpha_1} exp\left(\sum_{i=2}^{n_{gv}} \alpha_i C_i^{\beta_i} + \frac{E_{ag}}{R}\left(\frac{1}{T_{ref}} - \frac{1}{T}\right)\right), \tag{5.36}$$

$$\frac{dC_g}{dt} = -Y_{g1}\frac{dC_1}{dt},.$$

where C_1 represents the main component being converted to gel, and the temperature T is in °K and R=0.008314 kJ/(°K·mol). In order to compare with laboratory measurements, Eq. (5.36) can be integrated to give the gelation time which in the model is the time when a critical gel concentration C_{gcr} is reached.

$$t_{gel} = \frac{C_{1,0}^{1-\alpha_1}}{k(1-\alpha_1)}\left(1 - \left(1 - \frac{C_{gcr}}{Y_{g,1}C_{1,0}}\right)^{1-\alpha_1}\right), \tag{5.37}$$

$$k = r_g \, exp\left(\sum_{i=2}^{n_{gv}} \alpha_i C_i^{\beta_i} + \frac{E_{ag}}{R}\left(\frac{1}{T_{ref}} - \frac{1}{T}\right)\right),.$$

All gel above this critical concentration is irreversibly retained and denoted adsorbed gel A_g. The gel permeability is estimated using a Kozeny-Carman approach. With S_0 in m^2/cm^3 PV, the permeability in Darcy is given by:

$$k = 1.01325 \frac{\phi}{2\tau S_0^2} . \tag{5.38}$$

Rock tortuosity can be entered with keyword rtort or a default tortuosity value τ=2 is used here. The new permeability is simply computed by adding the surface area of the adsorbed gel to the surface area computed from the original rock. Note that permeability in the *x*-direction is used in the calculation of S_0. The permeability reduction is then:

$$S_0 = \sqrt{\frac{1.01325\varphi}{2k_0\tau}}, S_{gm} = A_g S_{g0}, \tag{5.39}$$

$$\frac{k}{k_0} = \left(\frac{S_0}{S_0+S_{gm}}\right)^2..$$

In the example below, the parameters are adjusted to a silicate system [ref.] where units wt% was used for silicate gel, Si and HCl (activator) and ppm for Ca concentration. In IORCoreSim, the wt% unit is approximated with g/100 ml. The specific area of adsorbed gel is estimated assuming the gel is aggregates of small silica particles with diameter 10 nm. The specific surface area is then:

$$S_{g0} = 6/(D\rho_g) = 6/(0.01\ \mu m \cdot 2.65\ g/ml) = 226\ m^2/g$$

The input units for S_{g0} depends on the units for adsorbed gel, A_g which is assumed to have the same units as C_1 (at present, this is not checked, and units specified for the gel component is currently ignored). The gel unit is typically in the form [mass/volume], for example [g/100 ml]. A unit conversion factor is used for the volumetric part, and S_{gm} is computed $S_{gm}=A_g \cdot S_{g0} \cdot f_{uv}$. With gel/Si concentration in [g/100 ml], the conversion factor f_{uv} is 0.01, and the computation of gel surface area can be written (with units):

$$A_g \left[\frac{g}{100\ ml\ pv}\right] S_{g0} \left[\frac{m^2}{g}\right] f_{uv} \left[\frac{100\ ml}{ml}\right] = 0.01 A_g S_{g0} \left[\frac{m^2}{ml\ pv}\right].$$

Example: Gelation parameters using tracer components adjusted to a silicate gel system.

```
cmpprop
*  name type iunit iads
  gel   1    7     aext /  unit g/100 ml
  Si    1    7     n /
  HCl   1    7     n /
  Ca    1   1     n /    unit ppm
/

gelmod
```

```
*  gmodel
1 /
*  gname ngvar
gel   3 /
*  variable names
Si HCl Ca /
*  alfa
4  0.7 0.1 /
*beta
1  0.5 /
*  yield
1 0 0 /
*  Rg (1/day)  Ea (kJ/mol) T0 (°C)
2.2E-7      77          20 /
*  rkfmod Ccr (critical Cgel)  Sp (m²/g gel)
2      0.3                  226 /
```

---

48. gelsps

Section: **propsect**

Description: Specific surface area of gel. Used for computation of gel permeability if special gel cell option (gelnum) is activated.

Two parameters on a single record is read, terminated with a slash (/).

Item	Name	Description	Default
1	*S0gel*	Specific surface area = Surface area/gel concentration. Units: [m²/ml]	4000
2	*taugel*	Tortuosity factor for the gel, τ_g. The tortuosity is defined here as $\tau_g = (L_t/L)^2$, where L_t is the actual flow path length through the medium and L the length of the medium.	2

Note: For a polymer gel, the polymer may be viewed as a long rod with an effective average diameter, D_p. The specific surface area would then be $S_0=4/D_p$. With D_p in µm, the units for S_0 becomes m^2/cm^3.

The gel permeability reduction is computed using a Kozeny-Carman approach. With S_0 in m^2/cm^3, the permeability in Darcy is given by:

$$k = 1.01325\frac{\varphi^3}{2\tau S_0^2(1-\varphi)^2}. \qquad (5.40)$$

Example:

```
gelsps
*  s0gel  taugel
3000    1.5      /
```

49. hystmod

Section: **propsect**

Description: Hysteresis options for history dependents residual oil saturation, description is given in chapter *6.3_Hysteresis*. Requires hysteresis to be activated with keyword hystsat in the runsect section). The keyword is optional, default values will be used if keyword is missing or if hysteresis options is not specified for all rock types.

The hysteresis option for *Sor* is set for each rock type in a single record terminated with a slash (/). The keyword is terminated with a slash on a new line.

Item	Name	Description	Default
1	*ir*	Rock type number (satnum), [1, *nrock*]	-
2	*ik*	Subset number (*ik* ≤ *mKrSet* in runsect section). NB!, present implementation does not work with interpolation activated (*mKrSet* > 1).	1
3	*sormod*	Flag indicating method for residual oil saturation, *Sor*. Options: 0 : No hysteresis in *Sor*. 1 : Modified Land's correlation. 2 : Tabular input of *Sor* versus maximum historical oil saturation. A table with two columns, *Sohmax* in the first and *Sor* in the second, must be included after this record and before the record for the next rock type (*ir*). The table must be terminated with a slash (/).	1
4	*CLmod*	Modification parameter for the Land's correlation. Default value, *CLmod* = 1, corresponds to the original model.	1
5	*Heps*	Scanning curve parameter used in Killough's hysteresis model for capillary pressure.	0.05

Note: The hysteresis option for relative permeability should only be used if residual oil saturation is history dependent.

Example: with two rock types.

```
hystmod
* ir  ik  sormod Clmod Heps
  1   1   2      1.0   0.02 / rock 1 using table
*  Sohmax  Sor
0.00  0.000
0.05  0.045
0.10  0.085
0.15  0.120
0.20  0.150
0.30  0.200
0.40  0.240
0.50  0.260
0.60  0.220
0.70  0.150
0.75  0.110
0.80  0.100
0.85  0.110
```

```
0.90  0.130
/ ends table
* ir  ik  sormod Clmod Heps
  2   1   1      0.9   0.02 / rock 2 using Land
/ ends keyword
```

50. intkr

Section: **propsect**

Description: Relative permeability interpolation method.

The interpolation method is set for each rock type in a single record terminated with a slash (/). The keyword is terminated with a slash on a new line.

Item	Name	Description	Default
1	*ir*	Rock type number (satnum), [1, *nrock*]	-
2	*ikrmod*	Indicates the method used for interpolation between *krA* and *krB* with interpolation parameter F_m. Options: 1: Linear interpolation, $kr = krA \cdot (1-Fm)+krB \cdot Fm$ 2: Interpolation parameters are modified: Water: $Fw=Fm^{Ew}$ and oil: $Fo=Fm^{Eo}$ 3: Interpolation of logarithmic relperm, $\ln(k_r)$	1
3	*ekrwint*	Water phase interpolation exponent (*Ew*), only used if *ikrmod* = 2	1
4	*ekroint*	Oil phase interpolation exponent (*Eo*). Eo <1: will increase the interpolation parameter, and more weight is put on krB. Eo=1: same result as with *ikrmod* = 1 (linear interpolation) Eo>1: puts more weight on krA. Only used if *ikrmod* = 2	1

Note: The interpolation parameter F_m can be set with keyword **ncmisc** or obtained from interpolating between *vint* associated with each relative permeability set.

Example: with two rock types.

```
intkr
* ir  ikrmod ekrw ekro
  1   1      /          rock 1
  2   2      1    0.5 / rock 2
/ ends keyword
```

51. jscale

Section: **propsect**

Description: Capillary pressure scaling options.

The first record terminated with a slash (/) reads:

Item	Name	Description	Default
1	*JscaleF*	Flag indicating J-scaling of capillary pressure. Options: 0: not used 1: J-scaling of capillary pressure 2: J-scaling using reference permeability and porosity	0
2	*IFTow*	Oil/water interfacial tension, [mN/m]=[dynes/cm].	20
3	*IFTgo*	Gas/oil interfacial tension, [mN/m]=[dynes/cm].	10

If *JscaleF*=2¸ a second record is repeated *nrock* times.

Item	Name	Description	Default
1	*Jkref*	Reference permeability in user defined units, e.g., [mD].	-
2	*Jpororef*	Reference porosity [vol. fraction].	-

If *JscaleF* = 1, the capillary pressure should be entered on dimensionless form (J-function). The capillary pressure is computed by:

$$P_c = C_f \cdot IFT \cdot J\sqrt{\frac{\phi}{k}} \, , \qquad (5.41)$$

where C_f is a unit conversion factor (e.g., C_f = 0.3183 for units set: mN/m, mD and bar). The interfacial tension is from the current keyword (IFT_{ow} or IFT_{go}), or if a surfactant component is defined, IFT_{ow} is computed from surfactant concentration.

If *JscaleF*=2, the capillary pressure obtained at some reference properties (permeability and porosity) should be entered in user defined pressure units. The capillary pressure is computed by:

$$P_c = P_{c-input}\sqrt{\frac{\phi}{k}\frac{k_{ref}}{\phi_{ref}}} \, , \qquad (5.42)$$

Note that the permeability used for the *J*-scaling is the permeability in the *x*-direction, k_x.

Example: applies J-scaling using reference permeability and porosity (*nrock*=2)

```
jscale
*   JscaleF IFTow IFTgo
2  /
*   Jkref Jpororef (nrock rows)
  500   0.2
 1000   0.3
/ ends table of reference k and poro
```

52. krLET

Section: **propsect**

Description: LET type relative permeability curve. Different models can be used for each phase (see example, **krtab**). Parameters are entered for one phase at a time in a single record terminated with a slash (/). The keyword is terminated by a slash (/) on a new line.

Variants of keyword:

krLET : imbibition kr if hysteresis is activated

krLETd : drainage kr, require hysteresis option to be activated (hystsat).

Item	Name	Description	Default
1	*ir*	Rock type (satnum), 1,..., *nrock*	-
2	*j*	Phase number: 0 or 'w': water phase 1, 'o' or 'ow': oil phase (oil/water limit) 2 or 'g': gas phase 3 or 'og': oil phase (oil/gas limit)	-
3	k_{rje}	End-point relative permeability	-
4	L_j	LET parameter *L*	-
4	E_j	LET parameter *E*	-
4	T_j	LET parameter *T*	-
5	S_{rj}	Residual saturation	-
6	*id*	Direction: 1=x-direction, 2=*y*-direction, 3=*z*-direction	1
7	*ik*	Relative permeability set number, must be between 1 and *mKrSet*.	1
8	*vint*	Value associated with current relative permeability set if interpolation is activated. This value could be log(N_c) or adsorbed amount of bacteria, see *iKrMode*. Alternative interpolation between two relative permeability sets can be entered in tabular form using keyword ncmisc.	0

Note: Directional relative permeability is optional. Relative permeability data must be specified for all phases and rock types in direction 1 (*x*).

The oil relative permeability from oil/gas (k_{rog}) is currently not used. No three-phase model is implemented and k_{ro} in three-phase cases is obtained from the oil/water limit (k_{row}).

The LET type formula used is:

$$k_{rj} = k_{rje} \cdot \frac{S_{jn}^{L_j}}{S_{jn}^{L_j} + E_j\left(1 - S_{jn}\right)^{T_j}}, \quad j = o, w, g, \tag{5.43}$$

$$S_{wn} = \frac{S_w - S_{wr}}{1 - S_{wr} - S_{or}}, S_{on} = \frac{S_o - S_{or}}{1 - S_{wr} - S_{or}}, S_{gn} = \frac{S_g - S_{gr}}{1 - S_{wr} - S_{gr}}.$$

If $S_{jn} > 1$, linear interpolation between k_{rje} and unity at $S_j = 1$ is used, e.g. if $S_w < S_{wr}$:

$$k_{ro} = k_{roe} + (1 - k_{roe}) \frac{S_{wr} - S_w}{S_{wr}}$$

Example: one rock type and 2 phases (*id*, *ik* and *vint* are defaulted):

```
krlet
*   ir   j   kre   L     E     T     Sr     id   ik vint
1    w   1.0   6.25 1.22 0.69 0.05 / B&H1) primary drainage data water-wet Berea
1    o   1.0   1.57 4.15 2.35 0    /
/
```

1) Braun, E.M. and Holland, R.F., "Relative Permeability Hysteresis: Laboratory Measurements and a Conceptual Model," *SPERE*, vol. 10 (August), pp. 222-228, 1995.

53. krpctab1, krpctab2

Section: **propsect**

Description: Use this keyword to enter relative permeability and capillary pressure from the same table. Multiple tables can be read, separated by a slash (/) on a new line. A second slash on a new line terminates the keyword.

Before each table, a record with up to 8 items is read (see table below). The first 3 items are required whereas the last 5 items are optional. NB! the last two items do not work in the current program version.

Note: No blank lines are allowed between the first record and the table, or within the table. A blank line will terminate the table reading, and subsequent rows will be lost.

Use **krpctab** (or **krpctab1)** for oil/water and **krpctab2** for oil/gas.

If hysteresis option is triggered (hystsat), drainage input is indicated with **krpctabd** while **krpctab** will indicate imbibition data.

Note: No three-phase model is currently implemented for oil, so only k_{ro} from the oil/water limit (**krpctab1**) will be used.

Item	Name	Description	Default
1	*ir*	Rock type number (satnum), [1, *nrock*]	-
2	*js*	Column number containing S_w (oil/water) or S_g (oil/gas)	-
3	*jkr1*	Column number containing k_{rw} (oil/water) or k_{rg} (oil/gas).	0

		jkr1 = 0 : no data read for water or gas	
4	*jkr2*	Column number containing k_{ro} (which can be k_{row} or k_{rog}) 0 : no data read for oil	0
5	*jpc*	Column number containing P_{cow} or P_{cgo} *jpc* = 0 : no data read	0
6	*id*	Direction: 1: *x*-direction, 2: *y*-direction Note: *id*=2 will trigger directional relperm, and data for both directions must be entered for all phases and all rock types.	1
7	*ik*	Relperm set, $1 \leq ik \leq mKrSet$, if relative permeability interpolation is activated.	1
8	*vint*	Value associated with relperm set *ik* if relative permeability interpolation is activated.	0

Notes:

- Oil/gas relperm is normally obtained in presence of connate water saturation.
- Saturations must increase downwards in the tables. Relperm must either increase or decrease monotonically, except at the ends where repeated zeros are tolerated.
- Formula (Corey) and tabular input can be mixed (see example below). If relperm is entered more than once for the same rocktype and phase, the last set read will be used.
- Separate tables may be used for e.g. oil and water, by setting *jkr1* or *jkr2* equal to zero.

Example: tabular three-phase relperms and capillary pressures for one rock type.

```
krpctab1
*  ir  js  jkr1 jkr2 jpc id ik vint - oil/water input
1   1   2    3    4    1   /
*  Sw      krw      kro       Pcow
0.09000  0.00000  1.00000   4.5
0.11100  0.00010  0.87965   0.99629
0.11763  0.00015  0.85692   0.49629
0.14586  0.00028  0.81855   0.09629
0.26330  0.00377  0.51262   0.02629
0.59976  0.11571  0.04337  -0.07071
0.78874  0.46203  0.00590  -0.12371
0.89274  0.66950  0.00099  -0.20371
0.94109  0.80677  0.00033  -0.50371
0.94682  0.84589  0.00020  -1.00371
0.95000  0.91000  0.00000  -1.50000
 /                                   ends table
 /                                   ends keyword

krpctab2
*  ir  js  jkr1 jkr2 jpc id ik vint - gas/oil (krog not used in current version)
1   1   2    3    4    1   /
*  Sg       krg       krog        Pcgo
0.00000   0.00000   1.00000    0.00000
0.05300   0.00000   0.71537    0.00072
0.07676   0.00002   0.67758    0.00467
0.10741   0.00009   0.62464    0.00664
0.86055   0.86858   0.00069    0.04809
0.87575   0.90118   0.00042    0.06783
0.88789   0.93565   0.00025    0.13100
0.89522   0.95957   0.00014    0.21686
0.91000   1.00000   0.00000    0.64150
/
```

```
/
```

54. krtab or krtab2

Section: **propsect**

Description: Relative permeability input in table format. Multiple tables can be read, separated by a slash (/) on a new line. A second slash on a new line terminates the keyword.

Before each table, a record with 6 items is read (see table below). The first 4 are required, the last 2 are optional. The record must be terminated with a slash (/).

Note: No blank lines are allowed between the first record and the table, or within the table. A blank line will terminate the table reading, and subsequent rows will be lost.

If hysteresis option is triggered (hystsat), drainage input is indicated with **krtabd** while **krtab** will indicate imbibition data.

Recommendation: use **krtab1** for oil/water and **krtab2** for oil/gas since in three-phase runs, **krtab2** must be used for oil/gas.

Item	Name	Description	Default
1	*ir*	Rock type number (satnum), [1, *nrock*]	-
2	*js*	Column number containing S_w (oil/water) or S_g (oil/gas)	-
3	*j1*	Column number containing k_{rw} (oil/water) or k_{rg} (oil/gas). *j1* = 0 : no data read for water or gas	0
4	*j2*	Column number containing k_{ro} (which can be k_{row} or k_{rog}) *j2* = 0 : no data read for oil	0
5	*id*	Direction: 1: *x*-direction, 2: *y*-direction, 3: *z*-direction Note: Relperm for *id*=1 is required. *id*=2,3 are optional.	1
6	*ik*	Relperm set, $1 \leq ik \leq mKrSet$, if relative permeability interpolation is activated.	1
7	*vint*	Value associated with relperm set *ik* if relative permeability interpolation is activated.	0

Notes:

1. Oil/gas relperm is normally obtained in presence of connate water saturation.
2. Saturations must increase downwards in the tables. Relperm must either increase or decrease monotonically, except at the ends where repeated zeros are tolerated.
3. Formula (Corey) and tabular input can be mixed (see example below). If relperm data is entered more than once for the same rocktype and phase, the last set read will be used.
4. Separate tables may be used for e.g. oil and water, by setting *jkr1* or *jkr2* equal to zero.

Example: oil/water with four rock types, mixed Corey and tabular input.

```
krtab1
*  ir   js   j1   j2   id   ik   vint
```

```
1   1   2   3   1   1   /  rock 1
*  Sw   krw   krow
0.100  0.000  0.800
0.200  0.016  0.588
0.300  0.046  0.408
0.400  0.084  0.261
0.500  0.130  0.147
0.600  0.181  0.065
0.700  0.238  0.016
0.800  0.300  0.000
/                       ends first table

*  ir  js  j1  j2  id  ik vint
2   1   2   3   1   1   /  rock 2
*  Sw  krw  kro
include krow2.dat   / includes second table from file ‘krow2.dat’
/                       ends second table

*  ir  js  j1  j2  id  ik vint
4   1   2   3   1   1   /  rock 4
*  Sw  krw  kro
0.100  0.000  0.700
0.200  0.005  0.501
0.300  0.022  0.341
0.400  0.049  0.216
0.500  0.088  0.124
0.600  0.137  0.060
0.700  0.197  0.022
0.800  0.268  0.004
0.900  0.350  0.000
/                        ends table for rock 4
/                        ends keyword

corey                         uses Corey type input for rock 3 and 4
*  ir  j  kre  E    Sr     id  ik  vint
   3   o  0.8  2.5  0.15   /  oil, rock 3, id=1 is default
   3   w  0.3  2.0  0.12   /  water, rock 3
   4   w  0.4  2.4  0.14   /  water, rock 4  - overwrite table data for water
/                                              table input for oil is kept
```

55. ncmisc

Section: **propsect**

Description: Triggers reading of a “miscibility” parameter F_m used to interpolate between two relative permeability sets. F_m is entered in a table versus $\log_{10}(N_c)$ if *iKrMode*=1, or versus bacteria adsorption if *iKrMode*=2. F_m should have a value between 0 and 1. F_m=0 (at low N_c) indicates immiscible conditions and the first relative permeability set is used. When F_m=1 (high N_c with conditions approaching miscible conditions), the second relative permeability curve is used.

Separate tables should be entered for each rock type (*nrock* tables).

Note: A maximum of two relative permeability sets (per rock type) can be used with this keyword.

The keyword takes one record and one table, both terminated with a slash (/). An additional slash (/) on a new line terminates the keyword.

1st Record			
Item	Name	Description	Default
1	*ir*	Rock type number associated with table, [1, *nrock*]	-
Table			
1	*x*	Variable used for interpolation, log(N_c) or adsorbed bacteria (A_b). See keyword *krflag* in runsect.	-
2	F_m	Miscibility function F_m used to interpolate k_r.	-

Rate dependent relative permeability

If another property is used for interpolation (see keyword krflag), the relevant property will replace $\log_{10}N_c$ in the text below).

F_m is entered as a table versus $\log_{10}(N_c)$ and takes a value between 0 (low N_c) and 1 (high N_c). The residual saturations to oil and water are interpolated:

$$S_{jr} = S_{jrA} + F_m\left(S_{jrB} - S_{jrA}\right), \quad j = w,o \tag{5.44}$$

Then k_r is interpolated. First all curves are saturation-scaled to match the new residual saturations. Let k_{rA} and k_{rB} represent the scaled curves at low and high N_c, respectively:

$$k_{rj} = k_{rjA} + F_m\left(k_{rjB} - k_{rjA}\right), \quad j = w,o \tag{5.45}$$

Alternatively, the logarithmic value of k_r can be interpolated instead (see **intkr**). The equivalent formulation used is

$$k_{rj} = k_{rjA}^{1-F_m} k_{rjB}^{F_m}, \quad j = w,o \, . \tag{5.46}$$

2 or more sets of relative permeability

If more than two sets of relative permeability is entered or if tabular F_m is missing, then F_m is linearly interpolated between N_c values that must be entered for each set.

$$F_m = \frac{\log N_c - \log N_{cA}}{\log N_{cB} - \log N_{cA}} \tag{5.47}$$

First the new endpoint saturations are determined using linear interpolation, Eq. (5.44). Then k_r is interpolated using either linear or logarithmic interpolation, Eq. (5.46) or Eq. (5.47), or alternatively the interpolation parameter F_m is modified with an exponent:

$$\begin{aligned} k_{rw} &= k_{rwA} + F_m^{ew}\left(k_{rwB} - k_{rwA}\right) \\ k_{ro} &= k_{roA} + F_m^{eo}\left(k_{roB} - k_{roA}\right) \, , \end{aligned} \tag{5.48}$$

where exponents *ew* and *eo* are entered via **intkr (***ekrwnc* and *ekronc*).

Example, with two rock types and with *iKrMode*=1:

```
ncmisc
*   ir
1    /   rock 1
*   log10(Nc)   Fm
    5          0
    4          0.2
    3          0.7
    2          0.9
    1          1
/                        ends first table
*   ir
2    /   rock 2
*   log10(Nc)   Fm
    5.5        0
    1          1
/                        ends second table
/                        ends keyword
```

Example with *iKrMode*=2:

```
Ncmisc
*   ir
1 /
*   Abac    Fm
0.0       0
0.00001   0
0.00010 1.0
/
/
```

56. nextir

Section: **propsect**

Description: Defines rock type, flow direction and set number for next relative permeability input. Useful if Eclipse-type format (SWOF or SGOF is used). One record terminated with a slash (/):

Item	Name	Description	Default
1	*nextir*	Next rock type number (satnum), [1, *nrock*]	-
2	*nextid*	Next directional relperm input (1: x, 2: y, 3:z)	old value
3	*nextik*	Next relative permeability set ≤ *mKrSet,* used if relative permeability interpolation is activated.	old value
4	*nextvint*	Value associated with next relperm set *nextik* if relative permeability interpolation is activated. This value represents a property value (e.g. logN_c) defined with the *iKrMode* flag set in the runsect section.	0

5	*drainF*	Drainage flag for next relative permeability set. Options: 0 : imbibition data 1 : drainage data (requires hysteresis activate, see hystsat).	Old value

Note: this keyword is only useful if SWOF or SGOF are used for next relative permeability input.

Example: sets next relperm input to be a second set (*ik*=2) for rock type 1.

```
nextir
*   ir   id   ik ival
1    1    2   /
```

57. pcowform, pcgoform

Section: **propsect**

Description: Capillary pressure input as formula for oil/water or gas/oil. Capillary pressure input should be specified for all rock types. Missing input for a rock type will result in zero capillary pressure for that rock type. Formula and tabular format can be mixed.

The keyword is ended by a slash (/) on a new line.

If hysteresis option is triggered (hystsat), drainage input is indicated with **pcowformd** while **pcowform** will indicate imbibition data.

Item	Name	Description	Default
1	*ir*	Rock type number (satnum), [1, *nrock*]	-
2	*CL*	Capillary pressure parameter	-
3	*C0*	Capillary pressure constant	-
4	*CR*	Capillary pressure parameter	-
5	*EL*	First capillary pressure exponent	-
6	*ER*	Second capillary pressure exponent	-
7	*SL*	Minimum saturation parameters	-
8	*SR*	Maximum saturation parameter	-

The capillary pressures for water/oil and oil/gas are calculated from the following relationship:

Oil/water

$$P_{cow} = C_{L1}(S_w - S_{L1})^{-E_{L1}} - C_{R1}(S_{R1} - S_w)^{-E_{R1}} + C_{01}. \quad (5.49)$$

Gas/oil

$$P_{cgo} = C_{L2}\left(S_g - S_{L2}\right)^{-E_{L2}} - C_{R2}\left(S_{R2} - S_g\right)^{-E_{R2}} + C_{02}. \qquad (5.50)$$

Optionally, J-scaling can be set with keyword jscale. The capillary pressure for oil/water is then:

$$P_{cow} = P_o - P_w = J_{ow} IFT_{ow}\sqrt{\varphi/K_{abs}}, \qquad (5.51)$$

$$J_{ow} = C_{L1}(S_w - S_{L1})^{-E_{L1}} - C_{R1}(S_{R1} - S_w)^{-E_{R1}} + C_{01}.$$

Note: The capillary pressures P_{cow} and P_{cgo} must be increasing with decreasing S_w or increasing S_g respectively. To accomplish that, C_{L1} and C_{R1} for oil/water must be positive, and C_{L2} and C_{R2} for gas/oil must be negative. If the wrong sign is found, these are changed as well as the signs for C_{01} or C_{02}.

Example: 3 phases and two rock types.

```
pcowform
*IR CL        CR        C0         EL     ER     SL     SR
 1  0.0005    0.00198   0.000005   3.15   3.5    0.037  0.95  /
 2  0.0014    0.0245    0.00005    3.00   1.0    0.046  0.70  /
/
pcgoform
 1  -0.0000912 -0.0000584 0.0016  0.5    3.75    0      1.0   /
 2  -0.0002    -0.000128  0.0035  0.5    3.75    0      1.0   /
/
```

58. polycse

Section: **propsect**

Description: Activates effective salinity model for polymer. Model input for a single polymer takes three records, each terminated with a slash ‘/’. If more than one polymer compounent is defined, the two last records must be repeated for all polymers (*npol* times).

Note: Keyword has been altered from version 1.298, however old input will be read correctly if not more than one polymer specie is used.

1st record: Reads a list of component names that will contribute to the effective salinity. The list must be terminated with a slash ‘/’. Assume N_{cse} components are found.

2nd record: Three items describing the effect of effective salinity, C_{SEp}, on polymer viscosity, terminated with a slash ‘/’. The 2nd and 3rd records must be repeated as a pair for all polymer components defined.

Item	Name	Description	Default
1	*csep0*	Minimum effective salinity C_{SE0}, units [M] must be a value larger than zero.	0.001

2	*acsep*	Effective salinity exponent α_{csep}, effect on intrinsic viscosity. Normally, *acsep* should be negative to give a decreasing polymer viscosity at increasing salinity. Setting *acsep*=0, will result in salinity independent viscosity.	0
3	*eta_min*	Minimum intrinsic viscosity $[\mu]_{min}$ at which salinity dependency vanishes. Units: [ml/g]	0

3rd record: Reads a list of effective salinity contribution, $\beta_{p,i}$, for all species listed in the first record, N_{cse} values. The list must be terminated with a slash '/'.

The effective salinity for polymer is computed:

In IORCoreSim, we use the expression

$$C_{SEp} = \sum_i m_i \beta_{p,i}, \qquad (5.52)$$

where m_i denotes the molar concentration of component *i*. Note that the actual units used can be different if components outside the geochemical model i used. The effective salinity will modify intrinsic viscosity using the expression:

$$[\mu] = ([\mu]_0 - [\mu]_{min}) C_{SEp}^{\alpha_{csep}} + [\mu]_{min}, \qquad (5.53)$$

with the restriction that C_{SE} cannot be lower than a critical value, $C_{SEp} \geq C_{SE0}$. Using $\alpha_{csep} = 0$, will result in a constant $[\mu]$. Ions defined using the geochemical model will have units [mole/L], however, any component defined can be set to affect the effective salinity. As an example, a passive salt component (no reactions) can be defined as a tracer component, in any user selected units (e.g., ppm). The units used will affect the input parameters in the polymer salinity model.

Note: if the polymer salinity model is triggered, the polymer intrinsic viscosity input must represent a salinity corresponding to C_{SEp}=1. If any of the species effective salinity parameter, $\beta_{p,i}$, is modified, intrinsic viscosity input at C_{SEp}=1 should be updated.

Example: Includes effect of three cations and two anions in molar units [mole/L].

```
polycse
*components affecting CSE
Na Mg  Ca Cl SO4 /
*  CSEP0 acsep eta_min
0.001  -0.5  500 /
*  betap: Na, Mg, Ca  Cl  SO4
0.5 10. 25. 0.5 2.0  /
```

59. polydegr

Section: **propsect**

Description: Activates computation of elongation (shear thickening) and polymer shear degradation. One record terminated with a slash '/'.

Item	Name	Description	Default

1	*el_lamf*	Modification factor C_{el} for estimation of relaxation time parameter λ_2 that denotes onset of elongation (shear thickening). Units: dimensionless.	4
2	*el_m2*	Slope of log(μ_{sp}) versus log($\dot{\gamma}$). Units: dimensionless.	1.5
3	*el_x2*	Tuning parameter for the transition between shear thinning and shear thickening. A higher value will give a sharper transition. Typical range: 1-3	2
4	*degr_ratef*	Degradation rate factor, r_{deg}. Units: fraction molecular ruptures/sec.	0.0015
-	*degr_Fsc*	Polymer critical shear stress constant. Units: [Pa] – NOT USED in version 1.264 or later, included in r_{deg}.	55
5	*degr_alfa*	Exponent α_d describing degradation dependency on shear stress. Units: dimensionless	3
6	*degr_beta*	Exponent β_d describing degradation dependency on molecular weight. Units: dimensionless	1

Shear thickening observed as increased effective viscosity at higher shear rate has been observed when polymer solutions are flowing in capillary tubes of varying diameter and in porous medium. The increased effective viscosity is explained by elongation of the polymer molecules due to accelerated flow through narrow pore throats. The onset of elongation is given by the relaxation parameter λ_2 (sec):

$$\lambda_2 = C_{el} \cdot 3.608 \cdot 10^{-11} \sqrt{\frac{k_{rw} E_{pva}}{S_w}} \cdot \frac{\phi}{(1-\phi)} \frac{\mu_s [\mu] M_w}{T}, \tag{5.54}$$

where units for the different parameters are; μ: mPa·s, [μ]: ml/g, M_w: g/mol and T: °K. Note that the input tuning parameter C_{el} would depend on the in-situ shear rate parameter α_c used (see keyword rockshear). Also, the constant $3.608 \cdot 10^{-11}$ is half the value used in Ref. [5] and consequently C_{el} should be twice the values used in that reference. Note that Eq. (5.54) in its present form was partly implemented in IORCoreSim version 1.299 (27. May 2019), while E_{pva} was included in version 1.308 (30. Mars 2020). E_{pva} represents the fraction of the water filled pore space available for the polymer solution after subtracting the effective space occupied by adsorbed polymer; E_{pva}=1-A_{pt}, where A_{pt} is given by Eqs. (5.73) and (5.76). E_{pva} is also related to the residual resistance factor by $E_{pva} = R_{RF}^{-2}$. The change from earlier is that the effect of two phases (S_w < 1) is more correctly represented. Note that some incorrect corrections for adsorbed polymer were implemented in v1.297-v1.298, and these versions will produce wrong results at high flow rates.

The viscosity corrected for elongated polymer is computed:

$$\mu_p = \left(\mu_{pa} - \mu_s\right)\left(1 + (\lambda_2 \dot{\gamma})^{X_2}\right)^{(m+n)/X_2} + \mu_s, \tag{5.55}$$

where μ_{pa} is the apparent viscosity after correction for a polymer depleted layer at the rock surface, Eq. (5.71). The apparent viscosity correction is done after shear thinning calculations, i.e., at increasing shear rate the correction for a depletion layer will be small. Without depletion layer, the viscosity at higher shear rate is computed:

$$\mu_p = \mu_s\left(1+\mu_{sp0}\frac{\left(1+(\lambda_2\dot{\gamma})^{x_2}\right)^{\frac{m+n}{x_2}}}{\left(1+(\lambda_1\dot{\gamma})^{x_1}\right)^{\frac{n}{x_1}}}\right). \qquad (5.56)$$

At higher shear rate, the specific viscosity can be approximated with:

$$(\lambda_2\dot{\gamma})^{x_2} >> 1\colon \mu_{sp} \approx \lambda_2^m\left(\frac{\lambda_2}{\lambda_1}\right)^n\dot{\gamma}^m. \qquad (5.57)$$

The probability for molecule rupture is related to the shear stress at the rock surface, $F_s = \mu\cdot\dot{\gamma}$. The degradation rate is also taken to depend on the formation effective pore radius, Eq. (5.65), and the molecular weight. The fraction of molecules rupturing per time unit is described by:

$$f_{rup} = (r_dF_s)^{\alpha_d}\frac{2M_w^{\beta_d}}{R_{pw}} \qquad (5.58)$$

$$F_s = \dot{\gamma}\mu$$

The units here are shear rate in s^{-1}, μ in Pa·s, M_w in ton/mol (10^6 g/mol), R_{pw} in μm and t in seconds. The change in molecular weight can be written:

$$\frac{dM_w}{dt} = -f_{rup}M_w, \qquad k = \frac{2(r_dF_s)^{\alpha_d}}{R_{pw}} \qquad (5.59)$$

$$\frac{dM_w}{dt} = -kM_w^{\beta_d+1}$$

Treating the rate parameter k in Eq. (5.59) as a constant, the integrated solution for the next timestep n+1 is:

$$\left(\frac{1}{M_w^{n+1}}\right)^{\beta_d} - \left(\frac{1}{M_w^{n}}\right)^{\beta_d} = k\Delta t \qquad (5.60)$$

The only unknown in Eq. (5.60), k, is computed as a function of the new molecular weight, $k = k(M_w^{n+1})$, and is determined implicit in time with iterations of on M_w. The new M_w at the next time step n+1 is:

$$M_w^{n+1} = \frac{M_w^n}{\left(1+k\Delta t\cdot(M_w^n)^{\beta_d}\right)^{1/\beta_d}}, k = \frac{2\left(r_dF_s(M_w^{n+1})\right)^{\alpha_d}}{R_{pw}} \qquad (5.61)$$

Note that using $\beta_d = 1$, will result in the same type of equation as used for thermal degradation in e.g. Ref. [6].

Example triggering elongation and shear degradation:

```
polydegr
*   el_lamf el_m2 el_x2 degr_ratef  degr_alfa degr_beta
4.0      1.5    3     0.0015       3.0       1.0   /
```

60. polyipv

Section: **propsect**

Description: Sets inaccessible pore volume and apparent viscosity correction for polymer. One record terminated with a slash '/'.

Item	Name	Description	Default
1	*ipv*0	Constant part of inaccessible pore volume, IPV_0, for polymer (pores less than polymer molecule size) in units, [PV fraction]. Typical range: 0 – 0.2	0
2	*dplFlag*	Flag indicating polymer depletion model to be used. Options: 0 : Not used 1 : Volume of depletion layer (IPV_d) will be added to $IPV = IPV_0+IPV_d$ 2 : Apparent polymer viscosity correction + IPV_d correction	0
3	*fdpl*	Multiplication factor or thickness of depletion layer, f_{dpl}. The thickness $\delta = f_{dpl} \cdot R_h$, where R_h is an effective radius of polymer molecule in solution.	1
4	*aM*	Exponent for modifying the viscosity ratio effect on apparent viscosity, a_M. A value <1 will reduce the effect. Note that item 7-8 offers an alternative method to achieve a similar effect; reduced effect of depletion layer at high polymer concentrations.	1.0
5	*tau*	Tortuosity parameter of the formation which is used in expression for the effective pore radius. Defined as $\tau=(L_t/L)^2$. Note: this constant tortuosity parameter will not be used if the more flexible tortuosity model is activated with keyword rtort (accounts for variation in permeability, porosity and saturation).	3
6	*Rcpd*	Relative concentration in the depletion layer from which the depletion layer viscosity is computed.	0
7	*alfadpl*	Exponent describing how the thickness of depletion layer depends on polymer concentration, suggested value $\alpha_{dpl} \leq 0$, where 0 corresponds to constant thickness and -0.75 is reported for HPAM.	0
8	*fcpd0*	Multiplication factor for polymer concentration dependency of the depletion layer thickness.	0.3

Polymer moves through a porous medium faster that its solvent (water). This can be attributed to the existence of a pore volume fraction inaccessible to the large polymer macromolecules, *IPV*. Chauveteau [7] suggested that pore wall exclusion of macromolecules should be expected to increase effective polymer transport velocity more than *IPV* in cleaner sands (where fraction *IPV* is small). Near a surface, macromolecule centers of mass are sterically excluded from the wall vicinity, which leads to a depletion layer at the solid interface.

If the depletion model is triggered (*dplFlag* > 0), the *IPV* is modelled as the sum of a constant part due to micro-porosity (IPV_0) and a variable part due to a depletion layer (IPV_d):

$$
\begin{aligned}
IPV &= IPV_0 + IPV_d, \\
IPV_d &= \left(1 - \left(\frac{R_d}{R_{fp}}\right)^2\right)(1 - IPV_0), \\
R_d &= R_{fp} - \delta,\ \delta = f_{dpl} R_{ph}.
\end{aligned}
\tag{5.62}
$$

The thickness of the depletion layer is assumed to be in the range of the hydrodynamic radius R_h, but may decrease with increasing concentration above some critical concentration, c_{pd0} [8]. It is computed by

$$
\begin{aligned}
\delta &= \begin{cases} \delta = f_{dpl} R_h, & f_{cpd0} c_p \le c_{p*} \\ \delta = f_{dpl} R_h \left(f_{cpd0}\, c_p / c_{p*}\right)^{\alpha_{dp}}, & f_{cpd0} c_p > c_{p*} \end{cases}, \\
c_{p*} &= \frac{1}{[\mu]},
\end{aligned}
\tag{5.63}
$$

where f_{cpl}, f_{cpd0} and α_{dp} are tuning parameters and c_{p*} is the overlap concentration at which molecular interactions becomes apparent. An approximate value for c_{p*} is the inverse intrinsic viscosity, $[\mu]^{-1}$. For most simulations presented in Ref. [5], $f_{dpl} = 1$. In simulations of a high viscous dataset from Howe et al. [9] with very clear depletion layer effects, the exponent was set to $\alpha_{dp} = -0.75$ and further matched with $f_{dpl} = 1$ and f_{cpd0} in the range 0.1 -0.3.

We also need to handle the polymer concentration. Injected concentration is split into a polymer rich phase with concentration c_{pp} and a water rich depleted layer at the pore surface. The average concentration in the rock is c_p, and c_p^* denotes the concentration after excluding adsorption and the constant part of the inaccessible pore volume. The relations between the different concentrations are given by

$$
\begin{aligned}
c_p^* &= \frac{c_p}{E_{PV0} E_{pva}} = c_{pp} E_{PVd} + \left(1 - E_{pvd}\right) c_{pd}, \\
R_{cpd} &= \frac{c_{pd}}{c_{pp}}, \\
c_{pp} &= \frac{c_p^*}{E_{pvd} + \left(1 - E_{pvd}\right) R_{cpd}} = \frac{c_p}{E_{PV0} E_{pva}\left(E_{pvd} + \left(1 - E_{pvd}\right) R_{cpd}\right)},
\end{aligned}
\tag{5.64}
$$

where R_{cpd} is a tuning parameter assuming a partitioning of some polymer into the depletion layer, $E_{PVd}=1-I_{PVd}$ and c_{pd} is the concentration in the depletion layer from which the depletion layer viscosity, μ_{dp}, is calculated.

The effective radius of the formation available for the polymer solution, R_{pw}, is obtained from a capillary tube model:

$$R_{pw} = \sqrt{\frac{8\tau k k_{rw} E_{pva}}{\varphi S_w E_{pv0}}}, \tag{5.65}$$

where E_{pv0}=1-IPV_0 is the effective pore volume fraction for polymer, R_{RF} is the residual resistance factor because of the effective volume fraction A_{pt} occupied by adsorbed polymer. The hydrodynamic radius of polymer in solution is computed with:

$$R_h = 5.4 \cdot 10^{-5}(M_w[\mu])^{\frac{1}{3}} \quad [\mu m] \tag{5.66}$$

This expression gives R_h in µm with units g/mol and ml/g for M_w and $[\mu]$, respectively. The effect of permeability on IPV_0 is not yet implemented, but a possible formulation could be:

$$IPV_0^* = \frac{IPV_0}{1+a_{ipv}\sqrt{k/\varphi}} \tag{5.67}$$

The effect of pore size exclusion is that polymer solution flowing into a block is split into a polymer rich part and a water rich part. The polymer viscosity must be computed using the concentration in the polymer rich part. Given the overall concentration c_p and assuming polymer concentration is zero in the water rich part, the concentration in the polymer rich part becomes:

$$c_{pp} = \frac{c_p}{E_{pv0}E_{pva}E_{pvd}}, \tag{5.68}$$

$$E_{pv0} = 1 - IPV_0,\ E_{pva} = 1 - A_{pt},\ E_{pvd} = \left(\frac{R_d}{R_{fp}}\right)^2.$$

A_{pt} is the effective pore volume fraction (after subtracting IPV_0) occupied by adsorbed polymer. The effective polymer concentrations that goes into the transport equation is

$$c_{pef} = c_{pp}f_p + c_{pd}f_w, \tag{5.69}$$

where f_p indicates the fractional flow of the polymer rich phase in the interior part of the capillary tube and fw indicates the fractional flow of water rich part in the depletion layer. If depletion model 1 (*dplFlag* = 1) is selected, f_p=1and c_{pef}=c_{pp}. If the second model is chosen (*dplFlag* = 2), a flow of polymer free water phase at the rock surface with magnitude (1-f_p)·q_w is considered. Then the effective polymer concentration becomes

$$c_{pef} = c_p \cdot \frac{E_{pvd}(2M_v(1-E_{PVd})+E_{PVd})+R_{cpd}\,M_v(1-E_{PVd})^2}{M_v-(M_v-1)E_{PVd}^2}, \tag{5.70}$$

where M_v represents the modified viscosity ratio $(\mu_p/\mu_s)^{aM}$ if that model is applied. An exponent *aM* of approximately 0.5 seems to be in line with experimental data. The existence of a polymer depleted layer at the rock surface will result in a slip effect and a reduced effective viscosity, here denoted the apparent polymer viscosity computed by:

$$\mu_{pa} = \frac{\mu_p}{M_v - (M_v - 1)E_{PVd}^2} \,. \qquad (5.71)$$

Example triggering concentration dependent depletion model with reduced apparent polymer viscosity:

```
polyIPV
*   IPV0 dplFlag  fdpl aM tau Rcpd alfadp cpd0
0.05   2      1.   1. 3   0   -0.75 0.001 /
```

61. polyrkf

Section: **propsect**

Description: Activates permeability reduction to the water phase due to adsorbed polymer. A single record terminated with '/'.

Parameter:

Item	Name	Description	Default
1	*fmrp*	Modification factor for permeability reduction factor. The assumed model value is *fmrp*=1, which can be tuned based on experiments.	1
2	*rkf_mod*	Integer indication which model is used for computing the permeability reduction. Options: 0: no permeability reduction 1: use adsorbed polymer 2: adsorbed polymer extends out from the wall 3: extended volume occupied by adsorbed polymer decrease at higher shear rate	2
3	*rp_lamf*	Parameter (C_{rp}) for onset of reduced swelling of adsorbed polymer. If *rp_lamf* > 0, then shear rate reduction in RRF (residual resistance factor) is activated. (used if *rkf_mod*=3). Units: [s^{-1}]	0.4
4	*rp_n*	Shear reduction exponent used if *rkf_mod*=3	0.5
5	*rp_x*	Shear reduction parameter used if *rkf_mod*=3	3
6	*S0p*	Specific surface area of polymer used with a Carman-Kozeny approach for estimating an upper limit for RRF. Only used if *rkf_mod* ≥ 2. Units: [m^2/ml]	7000

The residual permeability reduction factor RRF is estimated from the effective reduction of the pore space caused by adsorbed polymer, A_p. This is obtained from the relation between effective polymer size and intrinsic viscosity coupled with the relation between permeability and effective tube radius, see keyword polyipv. The final expression for RRF is:

$$RRF = \frac{1}{(1-A_{pt})^2}. \tag{5.72}$$

Depending on the chosen method (*rkf_mod*), the effective water volume fraction occupied by adsorption, A_{pt}, is computed:

$$\begin{aligned}
&1\colon A_{pt} = f_{mrp} A_p \\
&2\colon A_{pt} = f_{mrp} \frac{A[\mu]\rho}{2.5 S_w E_{pv0}} \\
&3\colon A_{pt} = f_{mrp} \frac{A}{S_w E_{pv0}} \left(\left(0.4[\mu]\rho_p - 1\right) f_{sh} + 1 \right),
\end{aligned} \tag{5.73}$$

where E_{pv0} is the effective PV fraction available for the polymer (1-IPV_0) due to size exclusion, and f_{sh} is a parameter for interpolation of the shear dependent swelling of adsorbed layer, used if *rkf_mod*=3 and modelled:

$$\begin{aligned}
&\lambda_3 = C_{rp} \cdot 3.608 \cdot 10^{-11} \sqrt{\frac{k_{rw} E_{pva}}{S_w}} \cdot \frac{\varphi}{(1-\varphi)} \frac{\mu_s [\mu] M_w}{T} \\
&f_{sh} = \left(1 + (\lambda_3 \dot{\gamma})^{X_{rp}}\right)^{-n_{rp}/X_{rp}},
\end{aligned} \tag{5.74}$$

where units for the different parameters are; μ: mPa·s, $[\mu]$: ml/g, M_w: g/mol and T: °K. E_{pva} represents a fractional decrease in available water volume for polymer flow due to adsorption, and μ_s is the solvent viscosity.

In low permeable medium A_{pt} may exceed unity, and Eq. (5.72) is not valid. An alternative method when adsorbed polymer layer extends to cover the total pore space ($A_{pt}=1$) is to use a Carman-Kozeny approach (available from version 1.297, 13 May 2019):

$$R_{RF} = \left(\frac{S_{0pv} + S_p}{S_{0pv}}\right)^2, \tag{5.75}$$

$$S_{0pv} = \sqrt{\frac{\phi S_w}{2 \tau_w k k_{rw}}},\ S_p = A_p S_{0p}.$$

To solve the problem with A_{pt} approaching and exceeding unity, we use the minimum value:

$$A_{pt} = \min\left(\frac{A_p [\mu] \rho_p}{2.5 S_w E_{pv0}}, 1 - \frac{S_{0pv}}{S_{0pv} + S_p}\right). \tag{5.76}$$

The use of Eq. (5.76) is demonstrated in Figure 5.7.

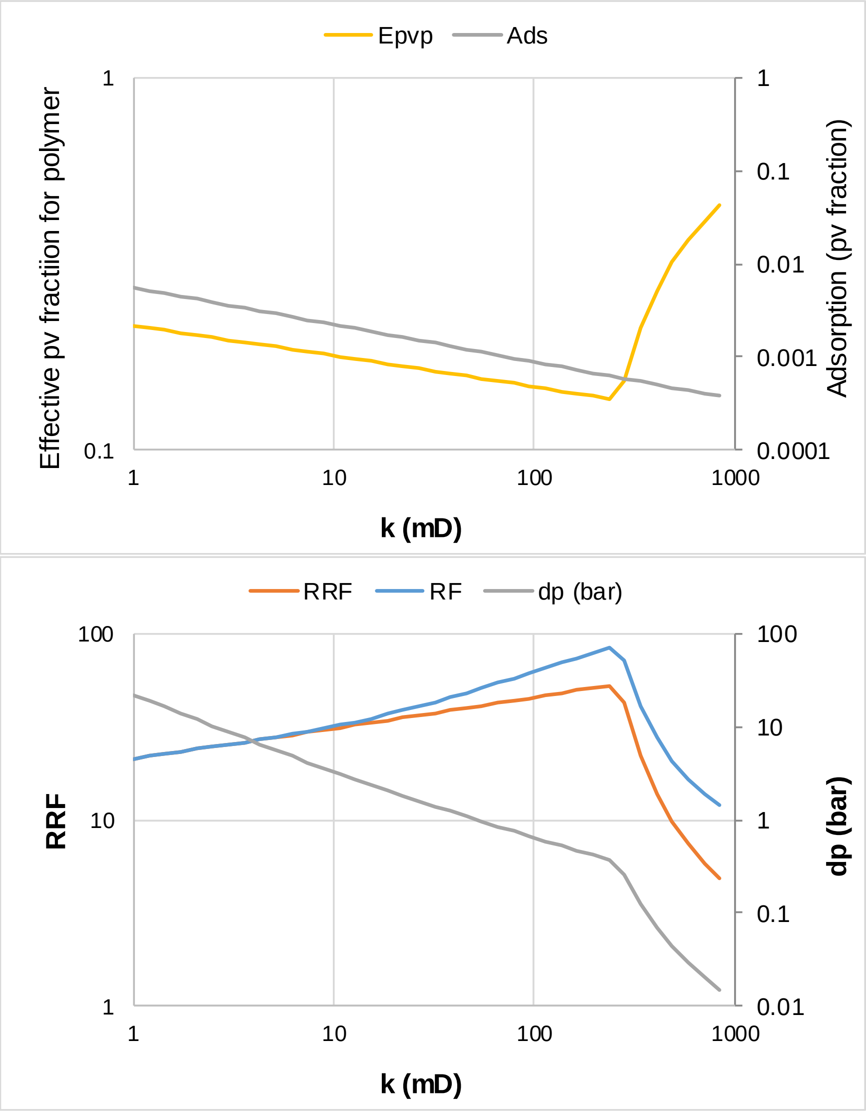


Figure 5.7 Effect of permeability on computed *RRF*, *RF* (resistance factor) and differential pressure drop over a core plug (bottom figure) when polymer retention is made dependent on permeability (upper figure). At low permeability, *RRF* is estimated using the Carman-Kozeny approach, while at higher permeability *RRF* is estimated using the decreased channel width model Eq. (5.72).

Example, triggering permeability reduction model:

```
polyrkf
*  fmrp rp_mod rp_lf rp_n  rp_x  S0p
1.0    3     0.5   1.0   4    7000 /
```

62. pref

Section: **propsect**

Description: Sets reference pressure that will be used to indicate reservoir conditions, e.g., reservoir flow rates, reservoir volumes. Must be terminated with '/'.

If this keyword is not used, reference pressure will be obtained from the rock property or alternatively from *Pref* for the first component.

Units: User defined, e.g. [bar].

Example:

```
Pref
200 /
```

63. reaction

Section: **propsect**

Description: Initiates species reaction. The type of reaction must be specified on the first record terminated with a slash '/'. Allowed options:

decay: General decay reaction (1st order reaction).

polyd: Random scission model suitable for polymer chemical degradation.

Reaction type = **decay** :

2nd Record, terminated with a slash '/'. Only read if reaction type = **decay**

Item	Name	Description	Default
1	*d_name*	Name of decaying component.	-
2	*r_nvar*	Number of components on which the reaction depends, n_{rv}.	0
3	*r_nprod*	Number of components produced in the reaction, n_{rp}.	0

3rd Record, terminated with a slash '/'. Only read if reaction type = **decay**

Item	Name	Description	Default
1	*r_k0*	Reaction rate constant. Time units [t^{-1}] is specified by the next item.	-
2	*tunit*	A single letter indicating the time unit for the rate constant. Options: s = second, m = minute, h = hour, d = day, y = year	s
3	*r_T0*	Reference temperature for the rate constant in [°C].	20
4	*r_EaR*	Activation energy for the reaction. Units [kJ/mol]	0

4th Record, a list of $n_{rv}\cdot 3$ items (3 items repeated n_{rv} times) terminated with a slash '/'. Only read if reaction type = **decay**

Item	Name	Description	Default
1	*r_name*	Name of component on which the reaction depends.	-
2	*r_alpha*	Reaction parameter α	0

3	*r_beta*	Reaction parameter β	1

5th Record, a list of n_{rp}·2 items (2 items repeated n_{rp} times) terminated with a slash ‘/’. Only read if reaction type = **decay**

Item	Name	Description	Default
1	*_name*	Name of component being produced in the reaction.	-
2	*r_Y*	Yield constant (stoichiometric constant) for the reaction,	0

The decay of component *i* is given by:

$$\frac{dc_i}{dt} = -k_i c_i, \tag{5.77}$$

$$k = k_0 \exp\left(\frac{E_a}{R}\left(\frac{1}{T_0} - \frac{1}{T}\right) + \sum_j^{ngv} \alpha_j c_j^{\beta}\right),$$

$$c_i^{n+1} = c_i^n + \Delta c_i = c_i^n \cdot e^{-k\Delta t},$$

$$c_{t,i}^{n+1} = c_{t,i}^n + \Delta c_{t,i} = c_{t,i}^n \cdot e^{-k\Delta t},$$

where the gas constant $R = 0.008314$ kJ/(°K·mol) and the temperature is in °K. The concentrations of involved species are relative to the water content, including eventual adsorption and are computed as $c_i = c_{t,i}/c_{t,w}$ with index t indicating that the overall (total) cell concentrations are used.

If i denotes the decaying component, the amount of a new component j formed in the reaction is:

$$\Delta c_{t,j} = -Y_j \Delta c_{t,i} \tag{5.78}$$

The model is demonstrated in Figure 5.8 with input given below. A single cell model with no wells is used. The model is initialized with 100 ppm of tracer TR0 and 1500 ppm hpam. TR0 degrades to a new component (tracer) TR1. For the polymer hpam, a negative reaction constant k_0 is applied to the molar concentration of hpam, named specie P-hpam (property of polymer is given the name ‘P-‘+polymer name). The volumetric concentration of hpam is kept constant (not degraded), and when the molar concentration increases the molecular weight and viscosity will decrease.

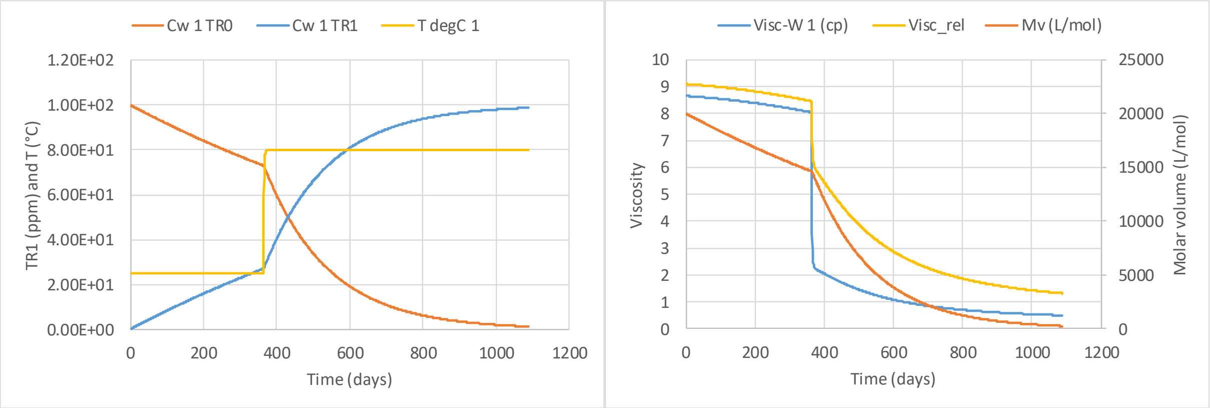


Figure 5.8 Test of decay option on polymer and tracer in a non-flow single cell model. For polymer, a negative decay constant is applied to the polymer molar concentration (x_p) which then increases with time. The volumetric concentration (c_p) remains constant and the molecular weight declines as shown in the right figure (see M_v). In a second 1st order reaction, a tracer TR0 is set to degrade to a second tracer TR1. The increasing TR1 concentration is shown in the left figure. Cell temperature and effect on the reaction rates are shown.

Example of using the decay reaction type to degrade polymer hpam and on a tracer TR0 reacting to a second tracer TR1.

```
REACTION
*   type
decay /
*   d_name, r_nvar, r_nprod
P-hpam      0        0 /
*   r_k0    tunit   r_T0   r_EaR
    1e-8   s       25     30 /
*   var_name r_alfa _beta
 /
*   product_name yield
 /

-----------------------------------------------------
reaction
*   type
decay /
*   r_kc_name, r_nvar, r_nprod
TR0         0        1 /
*   r_k0    tunit   r_T0   r_EaR
1e-8    s       25     30 /
*   var_name r_alfa _beta
 /
*   product_name yield
TR1            1. /
```

Reaction type = **polyd** :

<table>
<tr><th colspan="4">2nd Record, 2 items terminated with a slash ‘/’. Only read if reaction type = polyd</th></tr>
<tr><th>Item</th><th>Name</th><th>Description</th><th>Default</th></tr>
<tr><td>1</td><td><i>d_name</i></td><td>Name of decaying polymer.</td><td>-</td></tr>
<tr><td>2</td><td><i>r_nvar</i></td><td>Number of components on which the reaction depends, n_{rv}.</td><td>0</td></tr>
</table>

3rd Record, terminated with a slash '/'. Only read if reaction type = **polyd**			
Item	Name	Description	Default
1	*r_k0*	Reaction rate constant. Units [mol/(g·t)] with the time (*t*) units given by the next item.	-
2	*tunit*	A single letter indicating the time unit for the rate constant. Options: s = second, m = minute, h = hour, d = day, y = year	s
3	*r_T0*	Reference temperature for the rate constant in [°C].	20
4	*r_EaR*	Activation energy for the reaction, E_a. Units [kJ/mol]	0

4th Record, a list of n_{rv}·4 items (4 items repeated n_{rv} times) terminated with a slash '/'. Only read if reaction type = **polyd**			
Item	Name	Description	Default
1	*r_name*	Name of component on which the reaction depends.	-
2	*r_alpha*	Reaction parameter α	0
3	*r_beta*	Reaction parameter β	1
4	*r_Y*	Yield constant (stoichiometric constant) for consummation of present component in the reaction, Y	0

The random scission model implemented in IORCoreSim is described by:

$$\frac{dM_w}{dt} = -rM_w^2 \,, \tag{5.79}$$

$$r = k_0 \exp\left(\frac{E_a}{R}\left(\frac{1}{T_0} - \frac{1}{T}\right) + \sum_j^{n_{rv}} \alpha_j c_j^{\beta_j}\right)$$

The polymer is represented by a its volumetric concentration (c_p in units [volume fraction]), and its molar concentration (X_p [mol/L]). With M_w in units [g/mol], the density ρ_p should be in [g/L]. Substituting M_w with $M_w = c_p\rho_p/X_p$ where c_p is constant gives the following expression for change in molar concentration:

$$\frac{dX_p}{dt} = rX_pM_w = rc_p\rho_p \tag{5.80}$$

The reaction rate may depend on an arbitrary number of species (n_{rv}) which is allowed to be consumed in the reaction by the following mass balance for component j:

$$\Delta c_j = -Y_j\Delta X_p \tag{5.81}$$

A test of the implementation is done by comparing simulated and analytic solutions in Figure 5.9. The relevant simulation input is:

```
* \iorsimcases\hpam\ChemDegrad\cell\Xc2030-0D_test03.txt
cmpprop
*   Name Prefc Bw density Cw     PrefV TrefV°C visc  BTw   TC(°C) BPw
WAT1 1     1.0  1.0   4e-5  1.    25.     0.89  570.6. 133.15 0 /
---------------------------------------------------------------------------
* Polymer
*   name      Pref   Bf  density  compr iads vismodel
xanthan     100    1   1         0     n    3 /
*   eta    hug1  hug2  Tref  Tfact   Mv        alfaMv
5671   0.5  -1    20    0.000 3850000  0.827 /  viscosity at low shear
*   Carr_na Carr_nb Carr_la  Carr_le Carr_x
0.063    .804   .00035    1.0    1.0  /                 shear thinning
/  - empty slash - terminates the keyword cmpprop

IPVpol
*   IPVp_0 dplmflag dplmfact tau, turn of IPV and Depletion for bulk property sim.
0.0      0         1.0       3 /
reaction
*   type
polyd /
*   r_kc_name, r_nvar,
xanthan   0   /
*   r_k0      tunit r_T0  r_EaR
7.34E-13  s    120    173 /
*   variables
 /
*   products
 /
```

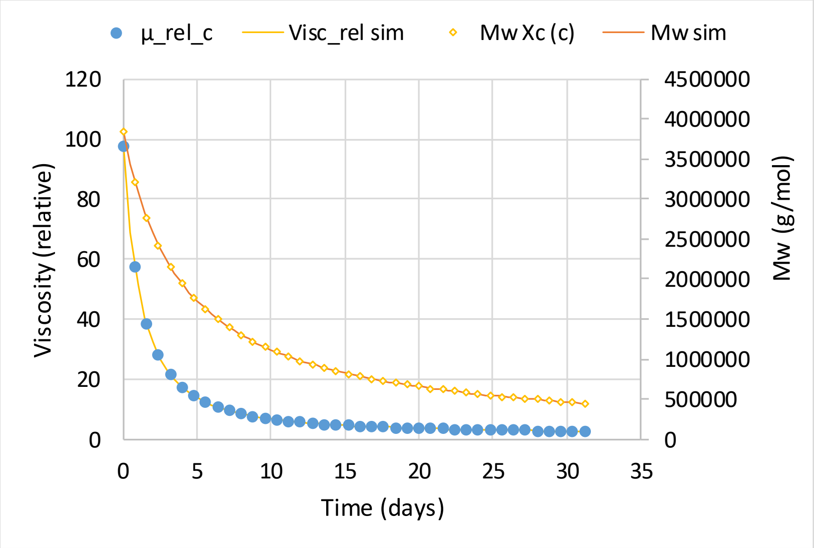


Figure 5.9 Comparing analytical and simulated viscosity and molecular weight using model and model parameters from [6]. 1000 ppm Xanthan stored at 120 °C and viscosity measured at 25 °C. Solvent viscosity is estimated to 0.89 mPa·s at 25 °C.

Thermal degradation of pam is assumed to follow the random scission model described in Eq. (5.79). Typically, stability of polymers is measured in terms of stability of the viscosity. The relation between intrinsic viscosity $[\eta]$ and M_w is given by the Houwink equation. An example of matching the model to experimental data is shown in Figure 5.10.

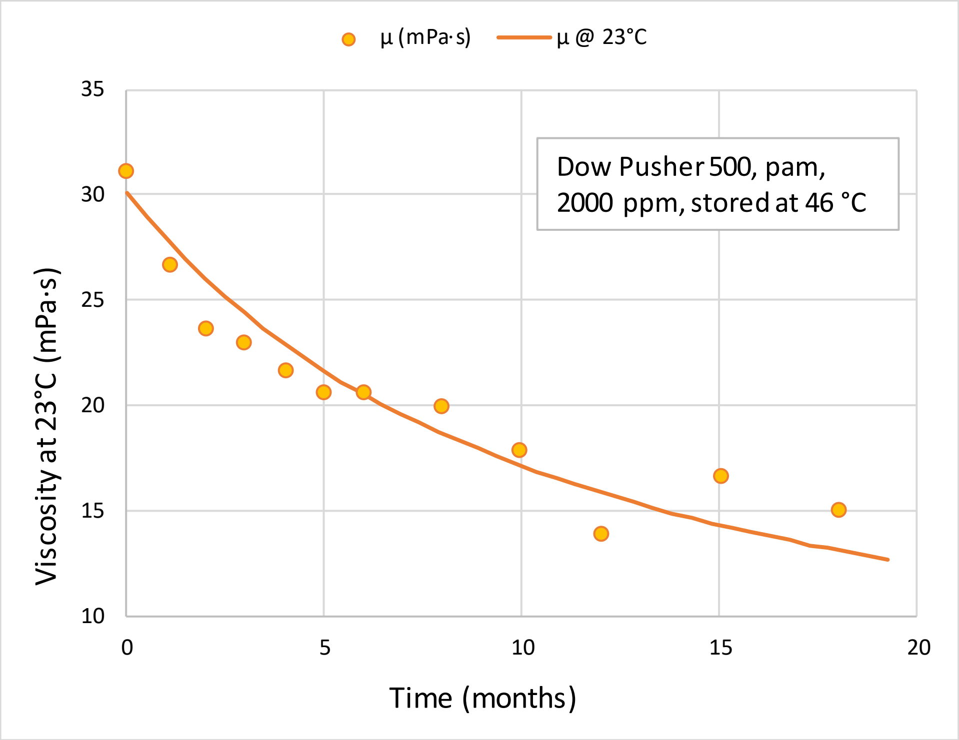


Figure 5.10 Thermal degradation of polyacrylamide, calculated with k_0=3.33E-15 s^{-1} at 46°C, M_{v0}=8.1 MDa, $[\eta] = 0.00195 \cdot M_w^{0.66}$, μ_w=0.932, k'=0.0562 (used in Martin's equation). Data from [10].

We may also assume that degradation depends on e.g. the oxygen concentration and that oxygen is spent in the reaction. An example of that, using the xanthan data presented above, is shown in Figure 5.11. The consumption of oxygen is set to one mol (16 g) for one molar increase in the polymer solution (one O-molecule spent for each polymer chain division). In the units used, the yield value for oxygen is set to 16000 mg/L. The input is:

```
* \iorsimcases\hpam\ChemDegrad\cell\Xc2030-0D_test05.txt
cmpprop
*   Name  Prefc  Bw  density  Cw     PrefV  TrefV°C  visc   BTw    TC(°C)  BPw
WAT1  1      1.0   1.0    4e-5   1.     25.      0.89   570.6. 133.15  0 /
*   Polymer
*   name      Pref    Bf   density   compr  iads  vismodel
xanthan     100     1    1         0      n     3 /
*   eta    hug1   hug2   Tref   Tfact    Mv          alfaMv
5671   0.5   -1     20    0.000  3850000   0.827 /  viscosity at low shear
*   Carr_na  Carr_nb  Carr_la   Carr_le  Carr_x
0.063    .804    .00035    1.0     1.0   /                  shear thinning
/
cmpprop
*   name  type  unit   iads
oxygen  1   mgpl   n /
/

IPVpol
*   IPVp_0  dplmflag  dplmfact  tau,  turn  of  IPV  and  Depletion  for  bulk  property  sim.
0.0     0        1.0        3 /
reaction
*   type
polyd /
*   r_kc_name,  r_nvar,
xanthan   1   /
*   r_k0      tunit  r_T0   r_EaR
```

```
7.34E-13  s    120    173 /
*   variables
*   name alfa beta yield (yield value includes unit conversion from mol/L to mg/L)
oxygen 4.0  0.1  16000 /
*   products
 /

phaseconc-w
WAT1 0.999 xanthan 0.001 oxygen 0.005 /  initial water phase concentrations
```

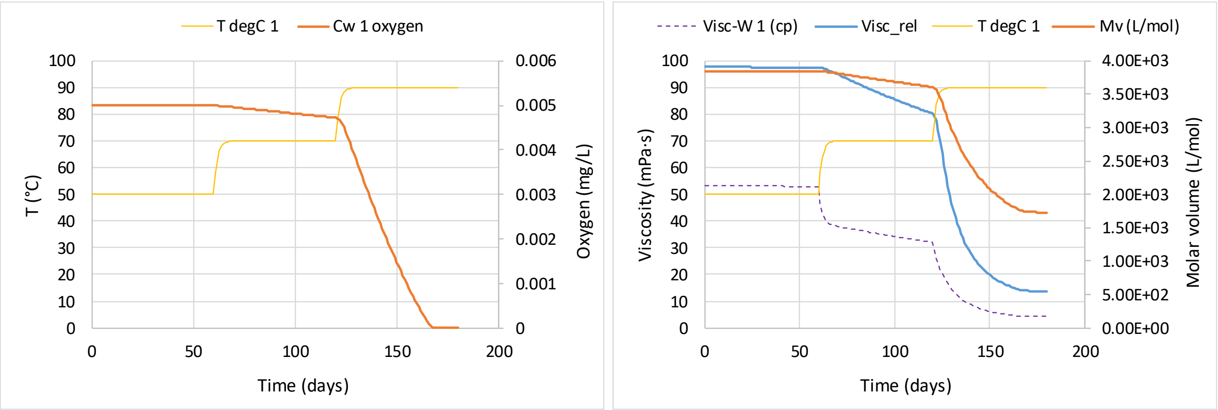


Figure 5.11 Example of simulated degradation of polymer in presence of oxygen. One oxygen is spent for each division of a polymer chain in the calculations.

64. rockcmpr

Section: **propsect**

Description: Sets the formation compressibility. One record terminated with a slash '/'. Variants of the keyword:

rockcmpr: Reads *nbl* = *nx*·*ny*·*nz* numbers.

rockcmpri: Reads a value for each rock type, *nrock* numbers.

rockcmprc: Reads a single value which is applied to all cells in the model.

rockcmprb: Reads a number for every cell in the box region.

If an 'l' or 'L' is included as the 9th letter, the phase pressure used in compression of fluids and rock can be altered. By default, from IORCoresim version 1.311 (July 2020), the water phase pressure is used (in earlier version the oil phase pressure was used). This can be altered here by preceding the compressibility data with a letter 'w' or 'o' indicating that the water or oil phase pressure shall be used.

rockcmprl: Reads phase indicator + *nbl* = *nx*·*ny*·*nz* numbers.

rockcmprli: Reads phase indicator + *nrock* numbers.

rockcmprlc: Reads phase indicator + a single value which is applied to all cells in the model.

rockcmprlb: Reads phase indicator + a number for every cell in the box region.

Item	Name	Description	Default

1	*LCpr*	Optional (see description above). Phase indicator for which phase pressure will be used in compressibility calculations. Options: W, w or 1 : water phase pressure will be used O, o or 2 : oil phase pressure will be used	w
2	*ncr*	Rock compressibility number in user defined units, e.g. [bar^{-1}]. Depending on the keyword variant used, up to *nbl=nx·ny·nz* values, one for each cell is read. If fewer values than required are read, the remaining block values are set equal to the last input value except if the rock type dependent version is used. Then *nrock* numbers are required.	0

Note: Reference pressure for compressibility is set with the keyword pref. The porosity input determines the pore volume at pref.

The porosity at fluid pressure *p* is computed:

$$\phi = \phi_0 e^{n_{cr}(p-p_0)}, \tag{5.82}$$

where p_0 is a reference pressure at which porosity is ϕ_0. In laboratory experiments, the net confinement pressure is sometimes changed. This effect can be included using the overburden keyword in the wellsect section. Defining a change in net confinement (or overburden) pressure, dP_{ob}, the porosity becomes:

$$\phi = \phi_0 e^{n_{cr}(p-p_0-dP_{ob})}. \tag{5.83}$$

The input values can be overwritten by repeating the keyword, e.g., by first using **rockcmprc**, and next **rockcmprb** to modify parts of the model.

Example: 100*10*10 cells with increased compressibility in lower layers

```
rockcmprc
*   crock   (bar)
1e-5 /
box
1 100 1 10 8 10 /
rockcmprb
3e-5 /

*Alternatively change to use oil pressure in compressibility calculations
rockcmprlc
*LCmpr crock
 o 1e-5 /
box
1 100 1 10 8 10 /
rockcmprb
3e-5 /
```

65. rockprop

Section: **propsect**

Description: Sets the formation compressibility. One record terminated with a slash '/'. Two variants of the keyword:

rockprop: Reads up to *nbl* = *nx·ny·nz* numbers.

rockpropb: Reads a number for every cell in the box region.

rockpropc: Reads a single value wich is applied to all cells in the model.

Item	Name	Description	Default
1	*rPref*	This reference pressure is no longer used. Reference pressure for rock compressibility is set using keyword pref.	1
2	*Crock*	Rock compressibility in user defined units, e.g. [bar^{-1}]. *nbl*=*nx·ny·nz* values, one for each cell is read. If fewer values than required are read, the remaining block values are set equal to the last input value.	0

Note: *rPref* is not used. Reference pressure for compressibility is set with the keyword pref. The porosity input determines the pore volume at pref.

The input values can be overwritten by repeating the keyword, e.g., by first using **rockpropc**, and next **rockpropb** to modify parts of the model.

Example: 100*10*10 cells with increased compressibility in lower layers

```
rockpropc
*   rpref crock  (bar)
200  1e-5 /
box
1 100 1 10 8 10 /
rockpropb
200 3e-5 /
```

66. rockshear

Section: **propsect**

Description: In-situ shear rate. One record terminated with a slash '/'.

Item	Name	Description	Default
1	*alfac*	Rock shear rate modification factor for use with the polymer model, units [dimensionless]. Typical range: 2-5	2.5
2	*gam_dFact*	Dampening factor for in-situ shear rate: If polymer is shear thickening in the near-well bore area, the distribution of injected polymer solution may be unstable in multi	0

		layered models. This can be solved by computing shear rate as a weighted average of the values from present and previous time steps. Range: [0, 1), where the value is the weight of the old shear rate, i.e., 0 indicates no dampening.	
3	*Well_PFlag*	Flag indicating treatment of polymer in well blocks: The shear thinning, thickening and degradation taking place close to the well bore is not captured if well is perforated in a large grid cell. This flag can be used to trigger numerical integration of the polymer behaviour for the block, which will improve simulated well pressure and degree of shear degradation. Options: 0: Option not used 1: Numerical integration of polymer equation is used.	0
4	*dRw0*	Used if *Well_PFlag*=1. The first radial interval used in numerical integration of rate dependent polymer rheology and molecular mechanical degradation for the well block. Recommended range: 0.001 - 0.01 m. Minimum value restricted to 0.0001 m. *Units:* user defined units, e.g. [m].	0.01 m
5	*dRVfact*	Volumetric incremental factor used for calculating radial intervals in the numerical integration of polymer properties in the well block. Restricted range: 1.1-2. Recommended range: 1.2-1.5	1.5

The in-situ shear rate is given in Eq. (5.16). The terms containing block velocity and water phase permeability, $u_w/\sqrt{kk_{rw}}$, used in the shear rate formula is computed:

$$\frac{u_w}{\sqrt{kk_{rw}}} = \left(\frac{u_{wx}^2}{k_x k_{rwx}} + \frac{u_{wy}^2}{k_y k_{rwy}} + \frac{u_{wz}^2}{k_z k_{rwz}}\right)^{0.5}, \tag{5.84}$$

where the directional velocity components are the average value of the flow across opposite boundaries. This is exemplified with flow in the *x*-direction, for interior and boundary cells:

$$u_{w,x,i} = \frac{\left|u_{w,x,i-\frac{1}{2}}\right| + \left|u_{w,x,i+\frac{1}{2}}\right|}{2}, i = [2, n_x - 1] \tag{5.85}$$

$$u_{wx,1} = \left|u_{wx,1+\frac{1}{2}}\right|, \qquad u_{wx,n_x} = \left|u_{wx,nx-\frac{1}{2}}\right|.$$

For cells containing a well penetrating the block in *z*-direction, the term $u_w/\sqrt{kk_{rw}}$ is computed:

$$\frac{u_w}{\sqrt{kk_{rw}}} = \left(\frac{u_{wr}^2}{\sqrt{k_x k_{rwx} k_y k_{rwy}}} + \frac{u_{wz}^2}{k_z k_{rwz}}\right)^{0.5}, \tag{5.86}$$

with the flow term along the well, u_{wz}, computed the same way as described above for cells not containing any well. The radial flow term u_{wr} represents the net horizontal into the block across all four block faces (in directions normal to the wellbore) and is computed as:

$$u_{wr} = \frac{\left(\left(u_{wx,i-\frac{1}{2}} - u_{wx,i+\frac{1}{2}}\right)\Delta x_i + \left(u_{wy,j-\frac{1}{2}} - u_{wy,j+\frac{1}{2}}\right)\Delta y_j\right)}{2\pi R_m}, \tag{5.87}$$

$$R_m = \frac{R_w + R_b}{2}, R_b = \sqrt{\frac{\Delta x_i \Delta y_j}{\pi}}.$$

The radial flow term is computed at a mid-point radius R_m. The shear rate computed at this radius is also the value that will be reported in printouts even if the numerical integration method is used. The choice of R_m for computation of the shear rate is arbitrary. It is meant as a reference point for the well block viscosity calculations and not the true shear rate. It is used this way in the numerical integration method described below. However, due to the complexity of polymer rheology, no analytic expressions have been implemented so far, and it is recommended to switch on the numerical integration method (*Well_PFlag* = 1, item 3 above). If only shear thinning is considered, analytic derivations using the Power law indicate that the shear rate should be computed at a much smaller radius in a typical range (not limited to) 10 - 50 % of R_m depending on well block size, well radius and the power law shear thinning index.

Numerical integration method (Well_PFlag = 1).

The pressure drop between the well and an outer well block radius can (for a polymer solution) be written:

$$\Delta p_{w1} = \frac{q_p}{\theta k k_{rw}} \mu_{eff} \, ln\left(\frac{R_b}{R_w}\right), \tag{5.88}$$

where θ is the radial sector width in radians and μ_{eff} is an effective shear dependent viscosity obtained from numerical integration over n intervals in the radial direction, with r_0=R_w and r_n=R_b.

$$\mu_{eff} = \frac{\sum_{i=1}^{n} \mu_i \, ln \frac{r_i}{r_{i-1}}}{ln \frac{R_b}{R_w}}. \tag{5.89}$$

The local viscosity is computed from the midpoint shear rate (index m computed for the well block):

$$\mu_i = \mu_i(\dot{\gamma}_i), \tag{5.90}$$

$$\dot{\gamma}_i = \dot{\gamma}_m \frac{r_{i-1} + r_i}{R_w + R_b}$$

The fraction of molecules rupturing during the last time step Δt is integrated:

$$f_{rup} = \sum_{i=1}^{n} \frac{df_{rup,i}}{dt} \frac{r_i^2 - r_{i-1}^2}{R_1^2 - R_w^2} \Delta t \tag{5.91}$$

$$\frac{df_{rup,i}}{dt} = \frac{df_{rup,i}}{dt}(\dot{\gamma}_{,i}, \mu_i).$$

The definition of f_{rup} and the calculation of the new molecular weight are described in Eqs. (5.58)-(5.61).

Example:

```
rockshear
*   alfac
2.5 /
```

67. rtort

Section: **propsect**

Description: Rock tortuosity using Archie's resistivity model. One record terminated with a slash '/' is optionally read for each rock type (nrock).

Item	Name	Description	Default
1	*m*	Rock cementation index, units [dimensionless]. Typical range: 1.4-2.6 (see below).	1.5
2	*nw*	Saturation index/exponent for water phase, units [dimensionless]. Typical range: 1.8-3	2
3	*no*	Saturation index/exponent for oil phase, units [dimensionless]. Typical range: Typical range: 1.8-3	2
4	*irflag*	Flag indicating if tortuosity parameters shall be read for each rock type. Options: 0: Same parameters used for all rock types. 1: Record repeated nrock times (this item is only read the first time)	0

This tortuosity model entered via this keyword will override the single value tortuosity input used in other models such as polymer rheology and degradation calculations (polyipv), and molecular diffusion.

Archie's first and second laws define the formation factor *FF* and saturation index *RI*, respectively:

$$FF = \frac{R_0}{R_w} = \varphi^{-m} \qquad (5.92)$$

$$RI = \frac{R}{R_0} = S_w^{-n},$$

where R represents the rock's electrical resistivity of a sample containing oil and brine, R_0 is the rock resistivity at 100% brine saturation and R_w is the bulk brine resistivity. The exponents m and n describe the restrictions on the geometric connection of water filled space caused by rock matrix and the presence of a non-aqueous phase. This geometric restriction on ion mobility can also be expressed in terms of the tortuosity. As m and n are routinely measured properties for significant amount of core material (needed e.g. for well log calibration), then these parameters can be used as input and at the same time offers a good representation of the effect of variation in permeability and saturation. The tortuosities for water and oil phases are computed:

$$\tau_w = \varphi^{1-m} S_w^{1-nw} \qquad (5.93)$$

$$\tau_o = \varphi^{1-m} S_o^{1-no}.$$

The cementation exponent (index) *m* is normally in the range (1-3), with low values for clean well sorted sand and higher values for consolidated rocks. Some ranges can be indicated for sand stones: 1.4-2.0, carbonates: 2.0-2.6 and igneous and metamorphic rocks that contains fractures: 1.0-1.4. The saturation exponent *n* is typically in the range 1.5-2 for water wet formations, 2-3 for mixed wet and higher for oil we formations.

Example with nrock = 2:

```
rtort
*   m  nw  no  irflag
1.8 2.5 2.5 1 /
1.6 2.2 2.7  /
```

68. swof

Section: **propsect**

Description: Allows the direct use of *Eclipse* format oil/water relative permeability and capillary pressure tables.

Up to *nrock* tables can be entered separated by a slash (/) on a new line. Two slashes (on separate lines) terminate the keyword.

Column	Name	Description
1	*Sw*	Water saturation, must be increasing downwards
2	*krw*	Water relative permeability, first entrance should be zero.
3	*kro*	Oil relative permeability, last entrance should be zero
4	*pcow*	Oil/water capillary pressure, $P_{cow} = p_o - p_w$. Units: [bar]

Note: S_w must increase downwards in the tables. P_c must decrease monotonically.

- Rock type number will be incremented before reading the next table. Use keyword nextir to set rock type and flow direction (and table number if relative permeability interpolation is activated) before this keyword.
- Tabular format and formula input can be mixed.

Example, oil/water run with two rock types using table format from *Eclipse*:

```
nextir
1 /

SWOF
*   ROCK TYPE 1
*   SW      KRW      KRO      PCOW (BAR)
```

```
0.09000  0.00000  1.00000   4.5
0.11100  0.00010  0.87965   0.99629
0.11763  0.00015  0.85692   0.49629
0.14586  0.00028  0.81855   0.09629
0.26330  0.00377  0.51262   0.02629
0.59976  0.11571  0.04337  -0.07071
0.78874  0.46203  0.00590  -0.12371
0.89274  0.66950  0.00099  -0.20371
0.94109  0.80677  0.00033  -0.50371
0.94682  0.84589  0.00020  -1.00371
0.95000  0.91000  0.00000  -1.50000
/                                   ends table

*  ROCK TYPE 2
*  SW      KRW      KRO      PCOW (BAR)
0.10000  0.00000  1.00000   5.00000
0.15000  0.00010  0.87965   1.20000
0.20000  0.00015  0.85692   0.50000
0.25000  0.00028  0.81855   0.10000
0.30000  0.00377  0.51262   0.02629
0.50000  0.11571  0.04337  -0.07071
0.70000  0.46203  0.00590  -0.12371
0.80000  0.66950  0.00099  -0.20371
0.85000  0.80677  0.00033  -0.50371
0.90000  0.84589  0.00020  -1.00371
0.95000  0.91000  0.00000  -1.50000
 /                                   ends table
 /                                   ends keyword
```

69. tlpmixpar

Section: **propsect**

Description: Activates the Todd Longstaff miscible displacement model for polymer.

Two items on a single record terminated with a slash ‘/’.

Item	Name	Description	Default
1	*TL_omegap*	Mixing parameter, ω_p, indicating the degree of segregated flow used in the miscible flow mode. Units, [dimensionless]. Typical values: 1.0 : No segregation (model has no effect) 0.7 : For moderate subgrid heterogeneity 0.5 : Significant subgrid heterogeneity 0.0 : Fully segregated flow (not recommend)	1
2	*TL_Cpmax*	Maximum polymer concentration used in the segregated flow model, C_{pmax}. Should normally be set to injected concentration. Units [vol. fraction]	0

The Todd-Longstaff-option for polymer flow assumes partially segregated flow. The gridblock is divided in a polymer rich and a water rich part. The effective viscosities in these parts are given by

$$\mu_{pe} = \mu_{pm}^{\omega} \cdot \mu_{p_{max}}^{1-\omega} \quad \text{and} \quad \mu_{we} = \mu_{pm}^{\omega} \cdot \mu_{w}^{1-\omega} . \tag{5.94}$$

where μ_{pm} represents the fully mixed viscosity and μ_{pmax} is the viscosity at C_{pmax}. The effective block viscosity for the water phase is calculated from:

$$\mu_{w_{eff}} = \left(\frac{1 - C_{pr}}{\mu_{we}} + \frac{C_{pr}}{\mu_{pe}} \right)^{-1} \quad \text{where } C_{pr} = \frac{C_p}{C_{p\max}}. \tag{5.95}$$

The correction of the polymer flow velocity is done by using an effective polymer concentration:

$$C_{p_{eff}} = \frac{\mu_{w_{eff}}}{\mu_{pe}} \cdot C_p \,. \tag{5.96}$$

Note: The TL-model results in a slower transport of polymer when concentration is less than C_{pmax}. This results in a sharpening of the polymer front, and a dispersion of the polymer tail.

Example:

```
TLPmixpar
*   TL_omegap TL_Cpmax
0.7       0.0015  /
```

5.4 HEATSECT

This section is optional. Its presence will activate temperature calculations, including convective and conductive heat flow. All parameters have default reasonable values. However, heat exchange with the surroundings is by default turned off.

heatsect keywords

Keyword	Short description
bfcond	Boundary conductivity factor
btc	Boundary temperature
extbb	External boundary blocks
fhcap	Fluids heat capacity
fhcond	Fluids heat conductivity
rhcap	Rock heat capacity
rhcondh	Rock's horizontal heat conductivity in horizontal (*xy*)
rhcondv	Rock's vertical heat conductivity (*z*)
rockdens	Rock matrix density
thexpan	Thermal expansion coefficients of water and oil
tolhc	Parameters affecting the heat transport calculations
work2heat	Include heat generated by fluid transport

---

70. bfcond

Section: heatsect

Description: Multiplier for the heat exchange between the model and the surroundings. By default, heat exchange with the surroundings is turned off, *bfcond* = 0. Variants of keyword:

bfcondc: Reads 6 values, a value for each surface of the model. The order of reading is: $yz(1)$, $yz(nx)$, $xz(1)$, $xz(ny)$, $xy(1)$, $xy(\mathrm{nz})$

bfcondyz: Reads $2\cdot ny\cdot nz$ values, a value for each boundary block at the first $yz(1)$ and last $yz(nx)$ boundaries

bfcondyzc: Reads 2 values, a constant value for the first $xz(1)$ and last $yz(nx)$ boundaries.

bfcondxz: Reads $2\cdot nx\cdot nz$ values, a value for each boundary block at the first $xz(1)$ and last $xz(ny)$ boundaries

bfcondxzc: Reads 2 values, a constant value for the first $xz(1)$ and last $xz(ny)$ boundaries.

bfcondxy: Reads $2\cdot nx\cdot ny$ values, a value for each boundary block at the first $xy(1)$ and last $xy(nz)$ boundaries

bfcondxyc: *R*eads 2 values, a constant value for the first *xy*(1) and last *xy*(*nz*) boundaries.

Default: Zero for all boundaries.

Notes:

- If too few values are found when the per-block variant of the keyword is used, the remaining boundary blocks are assigned the last input value found.
- By default, temperature is taken to be constant at half a boundary grid block distance outside the model. The heat conductivity outside the model is set equal to the rock matrix conductivity of nearest boundary block. Effective conductivity between this point and first boundary block is multiplied with the *bfcond* factor, which should normally have a value between 0 and 1.
- High temperature contrasts in combination with thin boundary blocks may result in unstable solutions of conductive heat flow. Solution: reduce *bfcond* multiplier or add layers of external boundary blocks (keyword extbb).

Example: turning heat exchange on in the z-direction (for the top and bottom surfaces), appropriate for a 2D cross section model.

```
bfcondc
0 0 0 0 1 1 /

*  Alternative input doing the same
bfcondxyc
1 1 /
```

71. btc, btyz, btyzc, btxz, btxzc, btxy, btxyc

Section: heatsect

Description: Constant boundary temperature. By default, the boundary temperature is computed from the **tinit** input. This can be overridden with this keyword. Keyword variants:

btc: Reads 6 temperatures (°C), a value for each surface of the model. The order of reading is: *yz*(1), *yz*(*nx*), *xz*(1), *xz*(*ny*), *xy*(1), *xy*(nz)

btyz: *R*eads $2 \cdot ny \cdot nz$ temperatures (°C), a value for each boundary block at the first *yz*(1) and last *yz*(*nx*) boundaries

btyzc: *R*eads 2 temperatures (°C), a constant value for the first *yz*(1) and last *yz*(*nx*) boundaries.

btxz: *R*eads $2 \cdot nx \cdot nz$ values, a value for each boundary block at the first *xz*(1) and last *xz*(*ny*) boundaries

btxzc: *R*eads 2 values, a constant value for the first *xz*(1) and last *xz*(*ny*) boundaries.

btxy: *R*eads $2 \cdot nx \cdot ny$ values, a value for each boundary block at the first *xy*(1) and last *xy*(*nz*) boundaries

btxyc: *R*eads 2 values, a constant value for the first *xy*(1) and last *xy*(*nz*) boundaries.

Default: Computed from temperature versus depth input, tinit keyword.

Units: [°C]

Notes:

- If too few values are found when the per-block variant of the keyword is used, the remaining boundary blocks are assigned the last input value found.
- By default, temperature is taken to be constant at half a boundary grid block distance outside the model.

Example: setting boundary temperature explicitly with a vertical external gradient.

```
btxc
70 70 70 70 72 68 /
```

72. extbb

Section: heatsect

Description: Adds layers of external blocks around the simulation grid to achieve more accurate computed heat exchange with the surroundings. Two records are read, each terminated with a slash (/).

1st record: Reads 6 values (nbx1, nbx2, nby1, nby2, nbz1, nbz2) representing the numbers of external grid cell layers at each surface of the model, *yz*(1), *yz*(*nx*), *xz*(1), *xz*(*ny*), *xy*(1), *xy*(nz). Default value is zero. Maximum value is 10.

2nd record: *R*eads external layer thicknes2s starting with the innermost layer at the first *xz* boundary. A total of (nbx1+nbx2+nby1+nby2+nbz1+nbz2) are read.

Units: User defined (cm, m, in or ft).

Notes:

- Using external boundary layers allows for extension of the computed temperature profile outside the internal (original) grid. The constant temperature boundary is pushed further away from the boundaries of the simulation model and computed heat exchange with the surroundings will be more accurate.
- If second record is defaulted, or less than expected values are read, external layer thickness becomes equal to the appropriate dimension of the first grid cell at each corresponding surface.
- If this option is used, heat exchange with the surroundings should be activated by using non-zero boundary conduction multipliers (bfcond).

Example: adding two boundary layers at the *xy*-boundaries, considering only conductive heat exchange in the z-direction.

```
bfcondxyc
1 1 / opens conductive heat flow in z-direction accross upper and lower xy-boundaries
extbb
0 0 0 0 2 2 /
```

```
*   external layer thickness, 2+2 values
2 4
2 4
/
```

73. fhcap

Section: heatsect

Description: Fluid heat capacities are the heat capacity per mass unit (kg) of the fluid component. Only the two main components WATER and OIL are included in the heat calculations. A single record terminated with a slash (/):

Input: Heat capacity values for all water and oil components defined in the project.

Item	Name	Description	Default
1	*fcapw*	Specific heat capacity to water ($h_{c,w}$). Units: [J/(g·°C)] = [kJ/kg·°C]	4.2
2	*fcapo*	Specific heat capacity to oil ($h_{c,o}$). Units: [J/(g·°C)] = [kJ/kg·°C]	2.0

The heat capacity of a component is converted to volumetric heat capacity ($h_{vc,k} = \rho_k h_{c,k}$) by multiplying with component density. The effective specific volumetric heat capacity of the saturated rock is computed:

$$h_{vc,e} = (1-\phi)h_{vc,r} + \varphi\left(c_{t,w}h_{vc,w} + c_{t,o}h_{vc,o}\right), \tag{5.97}$$

where $h_{vc,r}$ is the volumetric specific heat capacity to the rock matrix and $c_{t,w}$ and $c_{t,o}$ are the total volumetric concentrations of water and oil, respectively, in the rock void volume.

Example, modifying the oil specific heat capacity:

```
fhcap
4.2 1.5 /
```

74. fhcond

Section: heatsect

Description: Fluid heat conductivities. Only the two main components WATER and OIL are included in the heat calculations. A single record terminated with a slash (/):

Item	Name	Description	Default
1	*fconw*	Heat conductivity to water, $k_{h,w}$. Units: [W/m·°C)] = [J/m·s·°C]	0.6

2	*fcono*	Heat conductivity to oil, $k_{h,o}$. Units: [W/(m·°C)] = [J/ m·s· °C]	0.2

The specific heat flow is given by (W/m²):

$$u_{Eh} = \sum_l \left(u_l \sum_k \left(c_{l,k} h_{vc,k} \right) \right) - k_{h,e} \nabla T,$$
$$k_{h,e} = (1-\phi) k_{h,r} + \varphi \left(c_{t,w} k_{h,w} + c_{t,o} k_{h,o} \right). \qquad (5.98)$$

Example, modifying the oil conductivity:

```
fhcon
0.6 0.25 /
```

75. rhcap

Section: heatsect

Description: Rock matrix specific heat capacity ($h_{c,r}$) is the heat capacity per mass unit rock (kg) not including void space (pore volume). Variants of keyword:

rhcapc: Reads a single value that is assigned to the total grid.
rhcapir: Reads *nrock* values and assigns to grid based on rock type distribution.
rhcapl: Reads *nz* values. Assigned by layer.
rhcap: Reads *nbl* values, a value for each grid cell.

Default: 0.8 J/(g·°C).
Units: [J/(g·°C)] = [kJ/(kg·°C)]

Notes:

- If too few values are found, remaining input is assigned the last input value found.
- Multiplication with the rock matrix density results in the specific volumetric heat capacity for the rock matrix, $h_{vc,r} = h_{c,r} \cdot \rho_r$.
- Effective specific heat capacity for the saturated rock is given in Eq. (5.97).

Example: rock matrix specific heat capacity assigned by layers, *nz* = 10.

```
rhcapl
5*0.6 5*0.7 / horizontal direction
```

---

76. rhcondh

Section: heatsect

Description: Rock matrix heat conductivity in horizontal directions (*x* and *y*), $k_{hh,r}$. Variants of keyword:

rhcondhc: Reads a single value that is assigned to the total grid.

rhcondhir: Reads *nrock* values and assigns to grid based on rock type distribution.

rhcondhl: Reads *nz* values. Assigned by layer.

rhcondh: Reads *nbl* values, a value for each grid cell.

Default: 2.5 W/m·°C.

Units: [W/m·°C]

Notes:

- If too few values are found, remaining input is assigned the last input value found.

Example: rock matrix conductivity assigned by layers, $nz = 10$.

```
rhcondhl
2 2.5 5*3 4 4.5 5 / horizontal direction

rhcondvl
0.5 0.75 5*1.0 2.0 2.25 2.5 / vertcal direction
```

---

77. rhcondv

Section: heatsect

Description: Rock matrix heat conductivity in vertical directions (*z*), $k_{hv,r}$. Variants of keyword:

rhcondzc: Reads a single value that is assigned to the total grid.

rhcondzir: Reads *nrock* values and assigns to grid based on rock type distribution.

rhcondvl: Reads *nz* values. Assigned by layer.

rhcondz: Reads *nbl* values, a value for each grid cell.

Default: 1.25 W/m·°C.

Units: [W/m·°C]

Notes:

- If too few values are found, remaining input is assigned the last input value found.

Example: rock matrix conductivity assigned by layers, *nz* = 10.

```
rhcondhl
2 2.5 5*3 4 4.5 5 / horizontal direction

rhcondvl
0.5 0.75 5*1.0 2.0 2.25 2.5 / vertcal direction
```

---

78. rockdens

Section: heatsect

Description: Rock matrix density, ρ_r, a single value followed by a slash (/)

Default: 2.7 g/cm³

Note: The rock specific heat capacity (rhcap) is multiplied with this density to obtain a volumetric specific heat capacity, $h_{c,r}$

Units: User defined (g/cm³, kg/m³, ...).

Example with units (g/ml):

```
rockdens
2.6 /
```

---

79. thexpan

Section: heatsect

Description: Fluid thermal expansion relative to density set in the propsect. Thermal expansion is only considered for water and oil. Two records are read, both terminated with a slash '/'.

1st record: Reference temperature, T_0 in °C.

2nd record: Two values (c_{v1} and c_{v2}) describing the thermal expansion for each aqueous or oleic component.

Units: [°C^{-1}] and [°C^{-2}] respectively

The density of component *k* is computed by:

$$\rho_k = \rho_{0,k}\left[1 + c_{p,k}(p - p_0) - (T - T_0)\left(c_{v1,k} + c_{v2,k}|T - T_0|\right)\right], \qquad (5.99)$$

where c_p represent compressibility and c_v represent thermal expansion.

Notes:

- If too few values are found, missing parameters remain zero.

Example: thermal expansion for one water and one oil component.

```
thexpan
20 /                             reference temperature °C
3.0E-4 2.4E-6  1.21E-3 1.4E-6 /  water and oil expansion parameters
```

80. tolhc

Section: heatsect

Description: Sets parameters affecting the heat transport calculations. A single record terminated with a slash (/):

Input: Heat capacity values for all water and oil components defined in the project.

Item	Name	Description	Default
1	*tolhcond*	Tolerance limit for computation of heat energy transport by conduction. Units: [Dimensionless]	1E-10
2	*hcorder*	Specifies the order used in heat transport calculations during a timestep. Options: 0: Only heat transport by convection 1: Conductive heat transport is computed first 2: Conductive heat transport is computed last	1

In some special cases, unstable solutions can be made more stable by reducing error tolerance and to perform the conductive heat transport calculation last.

Example, reducing the error tolerance and moving the conductive heat exchange last to suppress instabilities in solution:

```
tolhc
*  tolhcond hcorder
1E-16    2 /
```

81. work2heat

Section: heatsect

Description: This keyword allows inclusion of heat generated by fluid flow in the energy equation. This option is turned on by single integer (1) followed by a slash (/).

Options: 1 : include work in energy equation

0 : option turned off (default).

The work required to move a fluid through a porous medium is essential energy needed to overcome frictional forces and is dissipated as heat energy. Here we neglect smaller terms due to changes in surfaces area (capillary forces) and potential energy (*mgh*). Assuming only flow in the *x*-direction, then generated heat energy in cell *i* over timestep Δ*t* is computed:

$$\Delta E_{h,i} = \left(\left|\frac{p_{i-1}-p_i}{\Delta x_{i-1}+\Delta x_i} q_{i-½}\right| + \left|\frac{p_i-p_{i+1}}{\Delta x_i+\Delta x_{i+1}} q_{i+½}\right|\right) \cdot \Delta x_i \Delta t. \tag{5.100}$$

Example: including heat generated by fluid transport.

```
work2heat
1 /
```

5.5 INITSECT

Use this section to initialize your model with saturations, pressure and phase concentrations.

initsect keywords

Keyword	Short description
box	Define input box for grid variables (described in gridsect)
phaseconc	Composition of phases initially in place
pinit	Sets initial pressure
sginit	Sets initial gas saturation
swinit	Sets initial water saturation
tinit	Sets reservoir temperature.

82. phaseconc

Section: initsect

Use: Initialise phase concentrations for water, oil and gas in the formation.

Description: There are six variants of the keyword indicating which phase to initialise and if constant or variable concentrations are read.

phaseconc-w : constant water phase composition

phaseconc-o : constant oil phase composition

phaseconc-g : constant gas phase composition

phaseconc-bw : variable water phase composition in current box region

phaseconc-bo : variable oil phase composition in current box region

phaseconc-bg : variable gas phase composition in current box region

Default: The concentration is set to one for the first component appropriate for each phase (the first WATER, OIL or GAS type defined). The default concentration of remaining components is zero. If the sum of volumetric components is different from one, the concentrations will be normalised.

Units: Volume fraction at P_{ref} (reservoir conditions), except for non-volumetric components where the units do not matter.

There are two variants of input for constant phase compositions.

1. A list of up to N_{cmp} (number of components defined) concentrations can be entered terminated with a slash '/'. The order must follow the rules defined under the **component** keyword. If less than N_{cmp} values are read, the remaining concentrations are set equal to zero.
2. An alternating list with the component name followed by a concentration value. The reading of concentrations must be terminated with a slash '/'.

Example 1: showing different methods to define initial oil composition. It is assumed one WATER component and two (or more) OIL components.

```
phaseconc-o
0 1 0 /

phaseconc-o
OIL2 0.5   OIL1 1  WAT1 0 /  Note: the concentrations will be normalised if sum ≠ 1.
phaseconc-o
*  must overwrite 1st oil component with zero if initial oil composition is OIL2
OIL1 0
OIL2 1
/
```

For variable initialization of phase concentrations, input consists of a component identifier (component number or component name) followed by *nbox* values (number of cells contained in the box region). If less than *nbox* values are found, missing values will be set equal to last input value. The list of values must be terminated with a slash '/'. A slash '/' ending an empty record will terminate keyword.

Example 2: 1D example with *nx*=100, initialized with two different oils, OIL1 in first half and OIL2 in second part.

```
phaseconc-o
OIL1 1  /  OIL1 in whole model

box
51 100 1 1 1 1 /
phaseconc-bo
*  must overwrite 1st oil component with zero in cell (51,100) defined with box
OIL1 50*0
OIL2 50*1
/
box
/ resets box to total model
```

Alternative input

```
phaseconc-o
OIL1 1  /  OIL1 in whole model

box
51 100 1 1 1 1 /
phaseconc-bo
*  Want to overwrite 1st oil component with zero in cell (51,100) defined with box.
*  If unsufficient data is found, remaining values in box is set equal to the
*  last value read, thus only one value is needed if followed by "/".
OIL1 0
/
*  Repeat keyword to enter a value for the next component within the box region.
phaseconc-bo
OIL2 1
/
```

```
box
/ resets box to total model
```

83. pinit

Section: initsect

Use: Initialise pressure in model. There are three variants of the keyword, namely:

pinit : Reads initial oil pressure for every block in user defined units, e.g. [bar]. If less than $nx \cdot ny \cdot nz$ values is found, remaining block values will be set equal to last input value.

pinitb : Initial oil pressure is read for every block in current input box. If less input values than defined by current box is found, remaining block values will be set equal to last input value. At least one input value is required. Units: user defined, e.g. [bar]

pinitc : Initial oil pressure is the same in all blocks, one single value is read. Units: user defined, e.g. [bar]

pinitd : Initial oil pressure is given at a reference depth. Two items are read:

pinitwx: Initialize the water phase pressure instead, and with **x** indicating the different options; ‘b’, ‘c’ or ‘d’.

Item	Name	Description	Default
1	*P0ref*	Initial oil pressure at reference depth in user defined units, e.g. [bar]	0
2	*DepthP0*	The reference depth in user defined units, e.g. [m]	0

Data must be terminated with a slash ‘/’.

Example: pressure specified at reference depth (bar and m).

```
pinitd
*   P0ref DepthP0
200    2000     /    200 bar at depth 2000 m
```

84. sginit

Section: initsect

Use: Initialise gas saturation, S_{gi}, in model. There are four variants of the keyword. The input must be terminated with a slash ‘/’.

sginit: Reads S_{gi} for every block, $nbl = nx \cdot ny \cdot nz$ values. If less than *nbl* values are found, the remaining saturations will be equal to the last input value.

sginitb: Reads S_{gi} for every block in current input box. If less input values than defined by current box is found, remaining block values will be set equal to last input value.

sginitc: S_{gi} is constant, reads a single value assigned to all blocks.

sginitl: S_{gi} by layers, reads *nz* values.

sginitpc: S_{gi} computed from capillary pressure, two items are read: (not working in current version).

Item	Name	Description	Default
1	*DepthGOC*	Depth to the gas/oil contact, *GOC* Units: user defined length units, e.g. [m]	0
2	*PcgoGOC*	Gas/oil capillary pressure at *GOC*: Units: user defined pressure units, e.g. [bar]	0

Example: with 10 layers, *nz*=10.

```
SginitL
2*0.5 0.4 0.3 0.2 5*0 /
```

---

85. swinit

Section: initsect

Use: Initialise water saturation, S_{wi}, in model. There are four variants of the keyword. The input must be terminated with a slash '/'.

swinit: Reads S_{wi} for every block, *nbl* = *nx·ny·nz* values. If less than *nbl* values are found, the remaining saturations will be equal to the last input value.

swinitb: Reads S_{wi} for every block in current input box. If less input values than defined by current box is found, remaining block values will be set equal to last input value.

swinitc: S_{wi} is constant, reads a single value assigned to all blocks.

swinitl: S_{wi} by layers, reads *nz* values.

swinitve: Vertical Equilibrium (*VE*) option with S_{wi} computed assuming equilibrium between capillary and gravitational forces. Two items are read, the depth to the oil-water contact (*WOC*) and the capillary pressure at this depth. The use of this option require that initial pressure varies with depth using the **pinitd** version of the pinit keyword. If the *Pc* hysteresis option has been triggered, *Swi* is computed using the drainage *Pc*.

Item	Name	Description	Default
1	*DepthWOC*	Depth to the wate/oil contact, *WOC* Units: user defined length units, e.g. [m]	0
2	*PcowWOC*	Oil/water capillary pressure at *WOC*: Units: user defined pressure units, e.g. [bar]	0

Example: with 10 layers, *nz*=10.

```
SwinitL
2*0.2 0.15 0.2 5*0.12 0.10 /
```

86. tinit

Section: initsect

Description: Sets the reservoir temperature to a constant value. Optionally, a vertical temperature gradient can be used. The temperature is fixed throughout the run in the current version. The data must be terminated with a slash, '/'.

Item	Name	Description	Default
1	*T0*	Reservoir temperature in ºC	50
2	*zT0*	Reference depth for *T0*. Units: user defined length units, e.g. [m]	0
3	*Tgrad*	Vertical temperature gradient. Units: ºC per user defined length units, e.g. [ºC/m]	0

Heat transport by convection and conduction is not implemented in the current version of *Bugsim*. Temperature effects may still be included in several models so that they will be ready for future implementation. Also, property parameters are easily reused at different temperature runs. The temperature is used in for example viscosity calculations.

Example: sets the temperature in the model to 70ºC.

```
Tinit zT0  Tgrad
70  /
```

5.6 Geochem

Activation of the geochemical model is done by including keyword geochem in the initsect section, e.g. at the end. The input structure is somewhat different from the that in the other parts.

This is a **preliminary** description of the geochemical input.

geochem keywords

Keyword	Short description
add_species	opens an include-file allowing additional definitions of species and reactions
complex	surface complexes
equilibrium_phases	specification of minerals that will be in equilibrium with solution 0
iexchange	define ion exchange capacity
rate	Defines minerals and dissolution kinetics
solution	Defines brine solutions (ion composition)

interpolate	Preliminary method to turn interpolation among previous solutions on/off. (interpolate 1 / interpolate 0). Should appear before solution 0.

87. complex

Section: geochem

Use: Model ion complexation at mineral surface.

Description: ....

88. equilibrium_phases

Section: geochem

Use: Defines mineral or other buffer phases.

Description: Reads a list of buffer phases and their concentration (weight fraction), each buffer on a new line. The keyword is ended with a slash '/' in the first column on a new line.

Up to 4 items is read from each line:

Item	Name	Description	default
1	*buffer_name_*	Name of buffer	-
2	*c_buffer_*	Concentration of buffer, weight fraction	0
3	*log_af_*	Log activity	0
4	*phase_name*	Valid input: 'oil' or 'gas', or else water is assumed.	water

Example:

```
--
equilibrium_phases
# name      wt-fraction log_a  phase
magnesite  .1           0.
calcite    .35          0.
CO2(G)     .6          -3.5
/ ends keyword
```

89. iexchange

Section: geochem

Use: Defines ion exchange capacity.

Description: Reads two items which must be on the same line. The keyword is ended with a slash '/' in the first column on a new line.

Two items are read:

Item	Name	Description	default
1	*exchange_name*	Name representing the exchange sites. The name must be defined before it is used here. A predefined name is X, representing cation exchange.	-
2	*capacity*	Exchange capacity, Q_{cec}, in units [Eq./L PV] = [meq/ml PV]	0

Cation exchange capacity (*CEC*) is commonly reported as meq/100 g solid material. Typical range for sandstone could be 0.1-1.0 meq/100 g. Converting to meq/ml PV :

$$Q_{cec} = CEC \cdot \frac{\rho_m(1-\phi)}{100\phi}, \tag{5.101}$$

where ρ_m is the matrix density in g/ml. Example: with *CEC* = 0.5 meq/100 g, porosity 0.2 and matrix density 2.7 g/ml, Q_{cec} = 0.054 meq/ml PV.

The ion exchange calculations in the geochemical model in IORCoreSim has been extended to also allow anion exchange at positively charged surfaces. The anion exchange capacity, *AEC*, in meq/100 g solid, is converted to Eq/L pore volume by

$$Q_{aec} = \frac{AEC \cdot \rho_m(1-\varphi)}{100\varphi} \tag{5.102}$$

Where ρ_m is the matrix (solid) density in g/ml and Q_{aec} (Eq/L) represents the anion exchange capacity input. Let *Z* represent the anion site name, the exchange of sorbed Cl^- with SO_4^{2-} can be written:

$$2ClZ + SO_4^{2-} = 2Cl^- + SO_4Z_2 \tag{5.103}$$

The ion exchange implemented in the geochemical model in IORCoreSim follows the Gaines & Thomas convention and can be expressed with:

$$K_{SO_4}^{Cl} = \frac{a_{ClZ}^2}{a_{SO_4X_2}} \cdot \frac{a_{SO4}}{a_{Cl}^2} = \frac{\beta_{Cl}^2}{\beta_{SO4}} \cdot \frac{[SO_4^{2-}]}{[Cl^-]^2}, \tag{5.104}$$

$$\beta_{SO_4} = 2 \cdot \frac{[SO_4Z_2]}{Q_{aec}}, \beta_{Cl} = \frac{[ClZ]}{Q_{aec}}.$$

Where β_i denotes the equivalent fraction occupied by species *i*, and a_i denotes the activity of species *i*.

The input exchange constants are in form of half reactions. As an example, for exchange between two ions, Cl^- and SO_4^{2-}, two constants are required, K_{Cl} and K_{SO4}. In the half reactions, Z^+ represents the unreacted surface sites:

$$ClZ = Cl^- + Z^+, \qquad K_{Cl} = \frac{a_{Cl}a_Z}{a_{ClZ}} = \frac{[Cl^-]\beta_Z}{\beta_{Cl}} \tag{5.105}$$

$$SO_4Z_2 = SO_4^{2-} + 2Z^+, \qquad K_{SO4} = \frac{a_{SO4}a_Z^2}{a_{SO4Z}} = \frac{[SO_4^{2-}]\beta_Z^2}{\beta_{SO4}}$$

In the above expression the activities are replaced with:

$$a_Z = \beta_Z = \frac{[Z^+]}{Q_{aec}}, \; a_{Cl} = [Cl^-]\gamma_{Cl}, \; a_{ClZ} = \frac{\beta_{Cl}\gamma_{Cl}}{Q_{aec}}, \tag{5.106}$$

$$a_{SO4} = [SO_4^{2-}]\gamma_{SO4}, \; a_{SO4Z} = \frac{\beta_{SO4}\gamma_{SO4}}{Q_{aec}},$$

where γ_i is the activity factor for specie i. The exchange constant $K_{SO_4}^{Cl}$ is obtained from the ratio:

$$K_{SO_4}^{Cl} = \frac{\beta_{Cl}^2}{\beta_{SO4}} \cdot \frac{[SO_4^{2-}]}{[Cl^-]^2} = \frac{K_{SO4}}{K_{Cl}^2}\,. \tag{5.107}$$

$K_{SO_4}^{Cl}$ can be determined from experiments. Finally, we use Cl^- as a reference ion and set the constant for this ion to unity, K_{Cl}=1, and use $K_{SO4} = K_{SO4}^{Cl}$ as input, or for other ions, $K_{A_i} = K_{A_i}^{Cl}$.

Similarly for cations, one typically uses Na^+ as a reference ion (K_{Na}=1) and $K_{M_i} = K_{M_i}^{Na}$.

Another convention, used e.g. in Utchem[2, 9], is to use the Normal [Eq./L] bulk concentration. and the cation exchange can be written (in terms of molar concentrations):

$$K_{MC_i} = \frac{\beta_{Na}^{z_i}}{\beta_i} \cdot \frac{z_i\left[M_i^{z_i}\right]}{[Na^+]^{z_i}} = Q_{cec}^{1-z_i} \cdot \frac{[NaX]^{z_i}}{\left[M_iX_{z_i}\right]} \cdot \frac{\left[M_i^{z_i}\right]}{[Na^+]^{z_i}}, \tag{5.108}$$

where M_i is a cation with valence z_i. If the half reaction constant for Na^+ is set to unity (K_{Na}=1), then the half reaction constants for other species will relate to the K_{MC} values by $K_i = K_{MC_i}/z_i$. Note that the K_{MC} values given in Table 5.3 compensate for that molar bulk concentrations used in Eq. (5.108) are not corrected for complexes:

Table 5.3 A list of ion exchange constants used with Eq. (5.108).

	K^+	Mg^{++}	Ca^{++}	Sr^{++}
K_{MC_i}	0.71	0.585	0.3	0.225
$\log_{10}(K_i)$	-0.15	-0.53	-0.82	-0.95

Example defining cation + anion exchange capacities:

```
Geochem

#include file with additional species, in this case anion exchange (Z+) is defined
add_species "ions2.txt"
--

--
iexchange
X 0.045
Z+ 0.06
/ end
```

The "ions2.txt" file includes definitions of ion exchange half reactions and the logarithmic value of the half reaction constants:

```
EXCHANGE_SPECIES
#name  (defines a name for anion exchange)
Z+
/end
SECONDARY_SPECIES
#cation exchange reaction followed by / ANA log(K) /
NaX = X- + Na+ / ANA 0. /
CaX2 = 2X- + Ca+2 / ANA -0.8 /
MgX2 = 2X- + Mg+2 / ANA -0.6 /
KX = X- + K+ / ANA -0.7 /
SrX2 = 2X- + Sr+2 / ANA -0.91 /
--
#anion exchange reaction followed by / ANA log(K) /
SO4Z2 = 2Z+ + SO4-2 / ANA  -0.8 /
ZCl   = Z+ + Cl- /ANA 0 /
ZHCO3 = Z+ + HCO3- /ANA -0.6/
/end
```

90. rate

Section: geochem

Use: Defines minerals and dissolution/precipitation kinetics.

Description: Reads mineral name and kinetic parameters for dissolution/precipitation from a single line. Several minerals can be specified, each on a new line. The keyword is ended with a slash '\' in the first column on a new line.

Ten items are read from each line:

Item	Name	Description	default
1	*min_name*	Mineral name. The name must be defined before it is used here.	-
2	*conc*	Initial mineral concentration, in units [weight fraction]	
3	*Sg*	Specific surface area (m^2/L mineral) (currently not used ?)	1
4	*log_af*	Log activity of mineral	0

5	*log_Ea1*	Log activation energy for first rate parameter (J/mol)	
6	k_1	First rate parameter, k_{10} (mol/L/sec) at 25 °C	
7	*log_Ea2*	Log activation energy for second rate parameter (J/mol)	
8	k_2	Second rate parameter, k_{10} (sec^{-1}) at 25 °C	
9	n	Saturation index exponent, n	
10	m	Second saturation exponent, m	0

Mineral dissolution and precipitation are described by the following rate law:

$$\frac{\partial \mathrm{c_i(t)}}{\partial \mathrm{t}} = \sum_j \xi_{ij}\, I_j, \tag{5.109}$$

where ξ_{ij} is a stoichiometric coefficient for the exchange of specie i between the aqueous phase and mineral j. The flux I_j is given by:

$$I_j = sgn(1-\Omega_j)(k_1 + k_2 a_H)\left|1-\Omega_j^m\right|^n. \tag{5.110}$$

where Ω_j is the saturation index (SI) in the aqueous phase, a_H is the H^+ activity and k_1 and k_2 are rate constants depending on temperature:

$$k_1 = k_{10} e^{\frac{E_{a1}}{R}\left(\frac{1}{298.15}-\frac{1}{T}\right)},\ k_2 = k_{20} e^{\frac{E_{a2}}{R}\left(\frac{1}{298.15}-\frac{1}{T}\right)}. \tag{5.111}$$

Example defining kinetics for calcite:

```
--
# rate = (k_1*exp(-Ea/Rg)(1/298.15)-1/T+k_2*aH*exp(-Ea/Rg)(1/298.15-1/T)*(1-SI^n)^m

#mineral wt-fraction Sg log_af   Ea_1   k_1      Ea2    k_2        m   n (chalk)
*  calcite    .95     1     0     37.8   3.43E-02  8.4   1.11E+03   1   1
#suggested input for Berea based on data in Ref. [11]
calcite      .01     1     0     37.8   1.2E-8    8.4   0.12       0.5 1
/end
```

91. solution

Section: geochem

Use: Defines brine ion composition.

Description: The keyword and a number on the same line defines a new solution, with its content given by the species list starting on the next line. Each row in the list has at least to items, a species names and its concentration. The species list is ended with a slash '\' in the first column on a new line (which also terminates the keyword). The first solution number should be zero, and other solutions should be defined in increasing order.

Species: only basis species should be used, Exceptions from normal (name + concentration) input:

pH 7.5 : ‘p’ in front of the component name ‘H’ indicates that the negative logarithmic value of [H$^+$] should be read next, pH=-log$_{10}$(H$^+$).

Hydrogen: ‘H’ is really representing a H$^+$/OH$^-$ balance, and added a constant 1 to avoid negative numbers (negative concentrations not liked by the transport algorithms). If ‘H’ is used in the species list (not ‘pH’) then a value 1 represents neutral (pure water) added other species not containing H or OH.

Example, fresh water + NaCl

```
solution 0
*   H = 1 + H+ - OH- = 1 (neutral mix-up water with [H+]=[OH-])
*   note final solution will not be neutral due to Na and Cl complexes.
*   ions are given in mol/l
H  1.0
Na 0.5
Cl 0.5
/ end
```

Example, fresh water + 0.1 M NaOH + 0.05 M H$_2$CO$_3$

```
solution 0
*   fresh water + 0.1 M NaOH + 0.05 M H2CO3
*   H = 1 - OH +H2CO3 = 1-0.1+0.05=0.95  (half the H in H2CO3 is specified by basis species HCO3)
      H    0.95
      Na   0.1
HCO3 0.05
/ end
```

Example using the geochemical model:

```
Geochem
#indicates a comment line
#add include file ‘ions2.txt‘ with additional definitions
add_species “ions2.txt”
solution 0
#   formation water. ions are given in mol/l
pH 7.5
Na 0.050
K  0.01
Mg 0.045
Ca 0.013
HCO3 0.002
Cl 0.125
SO4 0.024
/ end solution 0
rate
# rate = (k_1*exp(-Ea/Rg)(1/298.15)-1/T+k_2*aH*exp(-Ea/Rg)(1/298.15-1/T)*(1-SI^n)^m
#mineral wt-fraction Sg log_af logEa_1  k_1      logEa2  k_2       m   n
calcite   0.999 1   0  37.8 3.43E-03 8.4  1.11E+03  1  1
magnesite 0.0   1   0  60   0.7E-06  0.0  0.1  1  1
anhydrite 0.0   1   0  60   0.7E-08  0.0  0.0  1  1
#barite   0.0   1   0  60   0..7E-10 0.0  0.0  1  1
#witherite 0.001 1  0  37.8 3.4E-06  8.4  10.  1  1
/ end rate
iexchange
X 0.15
Z+ 0.2
/ end iexchange
solution 1
#   SW0Na0S. ions are given in mole/l
pH 7.0
```

```
Na 0.002
K  0.009
Mg 0.045
Ca 0.013
HCO3 0.002
Cl 0.125
SO4 0.0
/  end
solution 2
#   seawater.
pH 7.5
Na 0.450
K  0.01
Mg 0.045
Ca 0.013
HCO3 0.002
Cl 0.526
SO4 0.024
/  end solution 2
/ end geochem
```

The “ions2.txt” file:

```
EXCHANGE_SPECIES
#name  (defines a name for anion exchange)
Z+
/end
SECONDARY_SPECIES
#cation exchange reaction followed by / ANA log(K) /
NaX = X- + Na+ / ANA 0. /
CaX2 = 2X- + Ca+2 / ANA -0.8 /
MgX2 = 2X- + Mg+2 / ANA -0.6 /
KX = X- + K+ / ANA -0.7 /
#BaX2 = 2X- + Ba+2 / ANA -0.91 /
SrX2 = 2X- + Sr+2 / ANA -0.91 /
--
#anion exchange reaction followed by / ANA log(K) /
SO4Z2 = 2Z+ + SO4-2 / ANA  -0.8 /
ZCl   = Z+ + Cl- /ANA 0 /
ZHCO3 = Z+ + HCO3- /ANA -0.6/
/end
```

5.7 COMPSECT

This section contains computational parameters and keywords controlling printing of the simulation results.

compsect keywords

Keyword	Short description
bpc, bpr **bopr, bwpr**	History printout of block pressure values

bkcx	History printout of block component concentrations or adsorption.
bosat, **bwsat**	History printout of block saturation values
bxxx	History printout of block property values
ctransp	Component transport algorithm (explicit or implicit)
dcmax	Time step restricted by change in component concentrations
eclrst	Generate Eclipse type grid and restart files suitable for plotting distribution of properties.
gravity	Gravity flag
history	History printout frequency
hpflash	History printout of phase concentrations in production wells
phflash	Activates pH flash calculation for production wells at user defined temperatures.
pr_profile	Activates printout of properties along 1D grid profile
printcf	Component flags for phase concentration printout
printcft	Component flags for total concentration printout
printflags	Printout of various properties
printkey	Printouts of various phase properties
timestep	Timestep length parameters
solver	Solver method
tollit	Linear solver tolerance parameters
upstrc	Activates two-point upstream algorithm for species transport
upstream	Upstream weighted transmissibility
vtkprint	Activate printout of vtk files that can be viewed with program Paraview.

92. bkcx

Section:	compsect
Use:	Triggers printout of block concentration data.
Description:	Reads a single record terminated with a slash, ‘/’. The first entrance must be a component identifier (component number or name) followed by a list of block numbers.

The list of Possible keywords are:

bkcw : Block water phase concentration.

bkco : Block oil phase concentration.

bkcg : Block gas phase concentration.

bkct : Block total concentration.

bkca : Block adsorption.

Valid block numbers : 1, nbl (nx·ny·nz). Concentration data may also be printed for wells, using block numbers: *nbl*+m, where *m* is the well number in the order wells are defined (*welldef*).

Block values are printed versus time at history printout times to file *file-block.dat*.

Example showing alternative input methods:

```
*   Prints: water phase concentration of component 3 & 4 in blocks 1, 25 and 75
bkcw
3 1 25 75 /
bkcw
4 1 25 75 /

*   Prints: water phase concentration of 'hpam' component in blocks 1, 25 and 75
bkcw
hpam 1 25 75 /
```

93. bpc, bpr, bopr, bwpr

Section: **compsect**

Use: Triggers printout of block pressure data.

Description: Reads keyword followed by a list of block numbers. The list is terminated with a slash, '/'. Possible keywords are:

bpc : Block capillary pressure value.

bpr : Block oil pressure value.

bopr : Block oil pressure value.

bwpr : Block water pressure value.

Valid block numbers : 1, *nbl* (*nx·ny·nz*). Pressure data may also be printed for wells, using block numbers: *nbl*+m, where *m* is the well number in the order wells are defined (*welldef*).

Block values are printed versus time at history printout times to file *file-block.dat*.

Example:

```
*   Prints: oil and water phase pressures for block 25 and 75
bopr
25 75 /
bwpr
25 75 /
```

94. bosat, bwsat, bgsat

Section: compsect

Use: Triggers printout of block saturation data.

Description: Reads keyword followed by a list of block numbers. The list is terminated with a slash, '/'. Possible keywords are:

bosat : Block oil saturation value.

bwsat : Block water saturation value.

bgsat : Block gas saturation value.

Valid block numbers : 1, nbl (nx·ny·nz).

Block values are printed versus time at history printout times to file *file-block.dat*.

Example:

```
*  Prints: oil and water phase saturation for block 25 and 75
bosat
25 75 /
bwsat
25 75 /
```

95. bxxx

Section: compsect

Use: Triggers history printout of block property values.

Description: Reads keyword followed by a list of block numbers. The list is terminated with a slash, '/'. Possible keywords are:

bwsat, bosat, bgsat : Block saturation value, for either the water, oil or gas phase.

bwvisc, bovisc, bgvisc : Block viscosity value, for either the water, oil or gas phase.

brrfp : Block residual resistance factor from regular polymer adsorption.

brkfa: Block resistance factor from associative polymer network

brkfw: Block total resistance factor for the water phase including effect of adsorbed microbes, polymer (regular and associative) and gel.

bshear : Block in-situ shear rate value.

btemp : Block in-situ temperature.

bkrw : Block relative permeability to water.

bkro : Block relative permeability to oil.

bbw, bbo, bbr : Block volume factors, $V(p,T)\ /\ V(p_0,T_0)$, for water, oil and rock pore volume.

Valid block numbers : 1, nbl (nx·ny·nz).

Block values are printed versus time at history printout times to file *file-block.dat*.

Example:

```
*  Prints: oil and water phase saturation for block 25 and 75
bosat
25 75 /
bwsat
25 75 /
```

96. ctransp

Section: **compsect**

Description: Flag indicating how component transport algorithm is solved.

Item: *impRateF* followed by a slash '/'. Options:

0: Component transport is solved explicitly using reduced time step length if necessary (default).
1: Component transport is solved explicit or implicit depending on local residence time in each grid block. If phase residence time is smaller than time step length used in the pressure solution, implicit method is used.
2: Always use implicit solution for component transport.

Note: The implicit transport algorithm may speed up simulation, but will also increase numerical dispersion. It can be recommended if high cell throughput is experienced in limited part of the model, e.g., near the well.

Example: turning on the implicit transport algorithm, while still using explicit solution for cells with phase residence time larger than Δt.

```
ctransport
1 /
```

97. dcmax

Section: **compsect**

Description: Time step restriction by maximum relative change in component overall concentration. One record terminated with a slash '/'.

Item	Name	Description	Default
1	*compname*	Component name.	-
2	*dcrmax*	Maximum relative change in overall concentration per time step. Units: dimensionless.	-
3	*crefmin*	Minimum reference concentration.	1

		Units: depends on the component type, volume fractions for volumetric components and e.g. molar for geochemical components.	

Note: Can be used to restrict time steps for component fronts within a phase, or in single phase simulations with chemicals. The relative change is relative to the larger value of the current block concentration or *crefmin*. The use of a minimum reference value (*crefmin*) is to avoid unnecessary timestep restrictions at very low concentrations.

Example: in case with polymer injected into water flooded area.

```
dcmax
*   name   dcrmax crefmin (injected concentration: 0.0015)
hpam  0.1     0.001 /
```

98. eclrst

Section: **compsect**

Description: Triggers printout of Eclipse type grid and restart files that can be used for plotting of grid cell properties with e.g. FloViz (Schlumberger) and ResInsight (OPM; http://resinsight.org/). One record terminated with a slash '/'. An alternative keyword version (**eclrsta**) is described next.

Item	Name	Description	Default
1	*gridF*	Flag indicating type of grid. Options: 0: no printout 1: Grid file with extension .GRID (binary) or .FGRID (text) 2: Extensible Grid file .EGRID (binary) or .FEGRID (formatted text)	1
2	*grformat*	Flag indicating formatted text printout of grid file (previous item) and restart file. 0: Binary printout of restart file with extension .UNRST (produces smaller files, and files can be imported to ResInsight). 1: Formatted printout of restart file with extension .FUNRST	0
3	*gridhzF*	Flag indicating if variation of depth in the *xy*-plane shall be reported as a variation in *z*. Options: 0: constant *z*-value within the *xy*-plane 1: Depth variation reported as variation in *z*-values Note: Change in (*x*, *y*) coordinates due to rotation of the grid is not accounted for and cells become distorted if rotation (angles between horizontal plane and *x*- and *y*-axis) is significant. Using this option with 90° grid rotation will produce useless grid file.	0
4	*year*	Four digit calendar year representing simulation start	2000
5	*month*	Calendar month at simulation start, range (1,12)	1

6	*day*	Calendar day at simulation start, range (1,31)	1
7	*gridwF*	Printout of well paths to restart file. Options: 0: no printout 1: well information printed	1

Alternative record using keyword **eclrsta**, terminated with a slash '/'.

Item	Name	Description	Default
1	*gridF*	Flag indicating type of grid. Options: 0: no printout 1: Grid file with extension .GRID (binary) or .FGRID (text) 2: Extensible Grid file .EGRID (binary) or .FEGRID (formatted text)	1
2	*grformat*	Flag indicating formatted text printout of grid file (previous item) and restart file. 0: Binary printout of restart file with extension .UNRST (produces smaller files, and files can be imported to ResInsight). 1: Formatted printout of restart file with extension .FUNRST	0
3	*gridhzF*	Flag indicating if variation of depth in the *xy*-plane shall be reported as a variation in *z*. Options: 0: constant *z*-value within the *xy*-plane 1: Depth variation reported as variation in *z*-values Note: Change in (*x*, *y*) coordinates due to rotation of the grid is not accounted for and cells become distorted if rotation (angles between horizontal plane and *x*- and *y*-axis) is significant. Using this option with 90° grid rotation will produce useless grid file.	0
4	*gridwF*	Printout of well paths to restart file. Options: 0: no printout 1: well information printed	1
5	*year*	Four digit calendar year representing simulation start	2000
6	*month*	Calendar month at simulation start, range (1, 12)	1
7	*day*	Calendar day at simulation start, range (1, 31)	1
8	*hr*	Calendar hour at simulation start, range (0, 23)	0
9	*min*	Calendar minute at simulation start, range (0, 59)	0
10	*sec*	Calendar second at simulation start, range (0, 59)	0

Note: All static and dynamic grid properties from all report times are printed to a single unified restart file. History data (e.g. well production versus time) are not printed to eclipse type summary files.

Example: Generate printout of eclipse restart files.

```
eclrst
*gF form zF year month day  wF
 2  0    0  2010 1     1    1 /

* or
eclrsta
*gF form zF wF year month day  hr  min sec
 2  0    0  1  2010 1     1    12  0   0 /
```

99. gravity

Section: **compsect**

Description: Flag indicating if gravity will be used in pressure/flow calculations.

Item: *gFlag* followed by a slash '/'. Options:

0: gravity is not used
1: gravity included (default)

Example: turning gravity of.

```
gravity
0 /
```

100. history

Section: **compsect**, can be updated in wellsect

Description: History printout frequency, two items terminated with a slash '/'.

Item	Name	Description	Default
1	*nhist*	Printout will be every *nhist* timestep.	20
2	*dthist*	Maximum time between history printouts. Will not be used if value is zero or less. Units: user defined time units, e.g., [days].	0

Note: *dthist* will also act as an upper limit for maximum timesteps.

Example:

```
history
*   nhist dthist
30     0.1   / activates maximum time interval for printouts
```

101. hpflash

Section: **compsect**

Use: Triggers history printout of phase concentrations in production wells.

Description: Reads three integer printflags without space between on a single record terminated with a slash, '/'. The three printflags indicates history printout (1) or not (0) of phase concentrations for water, oil and gas phases.

Example: activating printout of only water phase concentrations

```
hpflash
100  /
```

102. pr_profile

Section: compsect

Use: Triggers the printout 1D profile data.

Description: Reads a single record describing direction and start coordinates of profile data to be printed at the same times and for the same variables as for grid distribution printouts. Profile data are printed to a separate file *name-profile.txt* in a tabulated form that can be imported (pasted) into e.g. Excel.

Record must be terminated with a slash '/'.

Item	Name	Description	Default
1	*idirpprt*	Indicates direction of profile data to be printed. Options: 1 : *x*-direction, the two next items represent (*j*, *k*) coordinates 2 : *y*-direction, the two next items represent (*i*, *k*) coordinates 3 : *z*-direction, the two next items represent (*i*, *j*) coordinates	1
2	*ipprt*	Indicates *x*-index of first cell (*i*) or *y*-index (*j*)	1
3	*jpprt*	Indicates *y*-index of first cell (*j*) or *z*-index (*k*)	1

Example: activate printout of 1D profile in the *x*-direction in second layer of a model with $nx \cdot ny \cdot nz = 20 \cdot 1 \cdot 5$:

```
pr_profile
*   idir j   k
1     1  2   /
```

103. phflash

Section: compsect

Use: Triggers flash calculation of pH for production wells.

Description: Reads a single record with up to three temperatures [°C]. Record must be terminated with a slash '/'.

Experimental pH data is typically measured at ambient conditions (room temperature and low pressure). This option allows comparing experimental and calculated pH in cases where core (or reservoir) temperature is different from the temperature at which pH is measured. This keyword requires that geochemical model has been activated.

Example, triggers flash pH calculations at three temperatures:

```
phflash_
```

```
*   T(i)
20 25 80  /
```

104. printcf

Section: **compsect**

Use: Printout of phase concentrations.

Description: Controls for which component phase concentration distribution will be printed. Three records are read, each terminated with a slash, ‘/’. The records can have up to *nt* (total number of components) integers with value 0 (not printed) or 1 (printed). The three records controls printout for the water, oil and gas phases respectively.

Alternatively, printing can be activated by component names appearing in the records.

Default: 0

Note: The printout must also be set via the *Pr_C* item under the **printkey** keyword.

Example, 4 components; Wat, Oil, xanthan and CH4

```
*   Prints: Sw,So,Sg,  Pw,Po, - CTOT, - Cwk,Cok,Cgk, - Visg
PrintKey
Pr_S 111
Pr_P 110
Pr_CT 1
Pr_C  111
Pr_Vis 001
  /
*   Sets phase concentration printflags for indiviual components
PrintCF
1 0 1 /  1st and 3rd component in the water phase
0 1 1 /  2st and 3rd component in the oil phase
0 0 1 /  3rd component in the gas phase
PrintCTF
1 1 1  / Total concentration for the first three components
```

Alternative input using component names

```
PrintCF
Wat xanthan /  1st and 3rd component in the water phase
Oil CH4 /  2st and 3rd component in the oil phase
CH4 /  3rd component in the gas phase
PrintCTF
Wat Oil xanthan  / Total concentration for the first three components
```

105. printcft

Section: **compsect**

Use: Printout of total concentrations.

Description: Controls for which component total concentration distribution will be printed. A single record terminated with a slash, '/' is read. The records can have up to *nt* (total number of components) integers with value 0 (not printed) or 1 (printed).

Alternative input, printing can be activated using a list of component names. Component names not in the record will not be printed.

Default: 0

Note: The printout must also be set via the *Pr_CT* item under the **printkey** keyword.

Example:

```
*   Prints: Sw,So,Sg,  Pw,Po, - CTOT, - Cwk,Cok,Cgk, - Visg
PrintKey
Pr_S 111
Pr_P 110
Pr_CT 1
Pr_C  111
Pr_Vis 001
  /
*   Sets phase concentration printflags for indiviual components
PrintCF
0 0 1 1 /  3rd and 4th component in the water phase
0 0 0 1 /  4th component in the oil phase
/  no printout of gas phase concentrations
PrintCTF
1 1 1 1 / Total concentration for the first four components
```

Alternative input:

```
PrintCF
xanthan CH4 /  printout of xanthan and CH4 water phase concentrations
CH4 /  printout of oil phase CH4 concentration
/  no printout of gas phase concentrations

PrintCTF
Wat Oil xanthan CH4 /  printout of four components
```

106. printkey

Section: **compsect**

Use: Triggers the reading of output keywords.

Description: Reads a list of keywords each followed by integer printflags. The list is terminated with a slash, '/'. Possible keywords and associated flags are:

Pr_S ijk : Saturation output. Three digits (0 or 1) control the printout of water, oil and gas respectively.

Pr_P ijk : Phase pressure output. Three digits (0 or 1) control the printout for the phases: water, oil and gas respectively.

Pr_Vel ijk : Phase velocity output. Three digits (0 or 1) control the printout for the phases: water, oil and gas respectively.

Pr_Vis ijk : Phase viscosity output. Three digits (0 or 1) control the printout for the phases: water, oil and gas respectively.

Pr_Dens ijk : Phase density output. Three digits (0 or 1) control the printout for the phases: water, oil and gas respectively.

Pr_C ijk : Phase concentration output. Three digits (0 or 1) control the printout for the phases: water, oil and gas respectively. Which components that are actually printed must be set via the **printcf** keyword.

Pr_CT i : Total concentration output. A single digit (0 or 1) controls the printout. Which components that are actually printed must be set via the **printcft** keyword.

Pr_kr ijk : Relative permeability output. Three digits (0 or 1) control the printout for the phases; water, oil and gas respectively.

Pr_k ijk : Effective phase permeability output including phase permeability reduction. . Three digits (0 or 1) control the printout for the phases; water, oil and gas respectively.

Pr_B ijk : Phase volume factors V(P,T)/V(P0,T0). Three digits (0 or 1) control the printout for the phases; water, oil and gas respectively.

Example:

```
*  Prints: Sw,So,Sg,  Pw,Po, - CTOT, - Cwk,Cok,Cgk, - Visg
PrintKey
Pr_S 111
Pr_P 110
Pr_CT 1
Pr_C  111
Pr_Vis 001
  /
*  Sets phase concentration printflags for indiviual components
PrintCF
1 0 1 /  1st and 3rd component in the water phase
0 1 1 /  2st and 3rd component in the oil phase
0 0 1 /  3rd component in the gas phase
PrintCTF
1 1 1  / Total concentration for the three first components
```

107. printflags

Section: **compsect**

Description: A number of printflags is activated by mnemonics in this section. No triggering keyword is used and no terminating slash (/) is needed. Each mnemonic must start on a new line. The mnemonics may be in upper, lower and mixed case letters. The following list is available:

pr_ads : Printout of adsorption in [fraction pv]. Printout require the component's *iads*-flag to be activated.

pr_elrw : Printout of electric resistivity distribution (Ohm), requires electric resistance to be activated.

pr_epot : Printout of electric field (Volt), requires electric resistance to be activated.

pr_fmisc : Printout of miscibility parameters used in relative permeability interpolation (krflag).

pr_gor : Printout of dissolved gas distribution in units [volume gas at surface conditions/volume oil at reservoir conditions]. Dissolved gas in water is not printed in current version.

pr_ift : Printout of interfacial tension. Require a surfactant component to be defined.

pr_min : Printout of mineral distribution in units [mole/L PV].

pr_mpn : Printout of bacteria number distribution in units [cells/ml water].
pr_mva : Printout of molar volume for adsorbed polymer.
pr_mvt : Printout of molar volume for total polymer.
pr_mvw : Printout of molar volume for polymer in the water phase.
pr_nc : Printout of capillary number distribution. Require rate dependent relative permeability to be activated (iKrMode=1)
pr_pcow : Prinout of oil/water capillary pressure.
pr_ph : Printout of pH, requires geochemical model to be activated.
pr_ppr : Printout of phase permeability reduction, k/k_0.
pr_rf : Printout of water phase resistance factor *RF* (mobility reduction factor). *RF* is typically used in polymer flooding where it represents $\Delta p_{polymer}/\Delta p_{water}$, the pressure drop ratio with/without polymer conducted at the same flow rate. *RF* represents the product of relative effective viscosity and phase permeability reduction factor.
pr_rkf : Printout of absolute permeability reduction factors, k_0/k.
pr_rkfa : Printout of water phase permeability reduction factor due to associative polymer
pr_rrfp : Printout of water phase permeability reduction factor due to adsorption of polymer.
pr_shear : Printout of in-situ shear rate used in the polymer model.
pr_si : Printout of logarithmic values of mineral saturation indexes (geochemical model)

pr_sigma : Printout of rock surface electric potential (geochemical model).
pr_tab : Changes distribution printout format to tabulator separated values for easier pasting into Excel.
pr_tage : Printout of average age bacteria cells [days].
pr_temp : Temperature printout [°C].
pr_br : Formation volume factor Br=PV(P)/PV(P0) [volume ratio].

108. solver

Section: **compsect**

Description: Selection of solver method

A single record with up to 3 items is read, terminated with a slash '/'.

Item	Name	Description	Default
1	*solver*	Solver method, three letters indicating IMPES (imp) or sequential (seq) method.	imp
2	*PcChordflag*	Flag indicating method for derivative P_c: 0: tangential method (recommended), 1: chord method	0
3	*krChordflag*	Flag indicating method for derivative k_r: 0: tangential method (recommended), 1: chord method	0
4	*iphPhase*	Flag indicating the phase pressure solved for. By default, the pressure equation is solved for the new oil pressure, which seems to produce more stable solutions. Alternatively, the aqueous pressure may be used by changing the flag to zero. 0: water phase pressure used	1

		1: oil phase pressure used (recommended)	

IMPES: The flow equation is solved using implicit pressure and explicit saturation formulations, i.e., when the solution is brought forward one step in time, the new pressure at time n+1 is found using relative permeabilities from the old time n.

The IMPES method requires smaller time steps than the Sequential Method, but offers less numerical dispersion of fronts. If capillary pressure is used in combination with small grid block sizes (as in simulation of laboratory core experiments and near well treatments in field) the method becomes highly unstable and the Sequential Method should be used.

In the Sequential Method, the pressure is first solved for using the IMPES method. Next, the saturation equation is solved keeping pressure and total inter-block flow rates from the IMPES solution unchanged. The results of the second solution are corrected new saturations and corrected inter-block fractional flow rates where the effects of saturation dependent phase mobilities and capillary pressure are included.

Note: The sequential method is currently only implemented for two-phase problems (oil+water).

Example:

```
solver
seq / changing to sequential solver method
```

109. timestep

Section: **compsect**, wellsect

Description: Regulates timestep calculations

A single record with up to 5 items is read, terminated with a slash '/'.

Item	Name	Description	Default
1	*dt0*	Initial timestep length. A zero entrance will keep the default value. Units: user defined time units, e.g. [days].	1 sec
2	*qxmax*	Maximum relative volume exchange of a phase per timestep in any grid-block (flow throughput). Possible range: $0 < qxmax < 1$ if IMPES solution method is used. If sequential solver option is used, try a value 5 to 100 (or larger). Units: [phase volume fraction]	0.5
3	*dsmax*	Maximum saturation change per timestep. NB! This is a main parameter for limiting time step lengths if unstable solutions are experienced. In core-scale simulations with strong capillary forces, a value of 0.01 or lower may be used.	0.1
4	*dtfmax*	Maximum increment factor for next timestep: $\Delta t^{n+1} \leq dtfmax \cdot \Delta t^{n}$	2

5	*dtmax*	Maximum timestep length. If zero or a negative number is entered, this item will not be used. Units: user defined time units, e.g. [days].	1E20 sec
6	*qxemax*	Maximum relative volume exchange of a phase per timestep used in the component transport equation, in any grid-block (flow throughput). *qxemax* should be used to stabilize transport of components travelling faster than the phase they are dissolved in. Example: polymer travels faster than water due to inaccessible pore volume and a value smaller than 1-*IPV* (inaccessible pore volume for polymer) should be used.	1

Example: In a polymer simulation with *IPV* in the range 0.25

```
timestep
*   dt0   qxmax dsmax dtfmax dtmax qxemax
0     5.0   0.2   1.5    0     0.7 /
```

110. tollit

Section: **compsect**

Description: Linear solver tolerance.

Item	Name	Description	Default
1	*zeta_imp*	Linear solver precision for the pressure [dimensionless].	1E-12
2	*itmax_imp*	Maximum number of iteration in linear solver.	2000
3	*zeta_seq*	Solver precision for the saturation. Only used if sequential solver option is selected.	1E-13
4	*itmax_seq*	Maximum number of iterations for the saturation. Only used if sequential solver option is selected.	200
5	*upstr_checkFlag*	Indicates if upstream checking is performed, options: 0: not used 1: upstream checking is performed	1
6	*upstr_dSerrmax*	Saturation error tolerance for use of wrong flow direction.	1E-6
7	*upstr_Flim*	Tolerance limit for phase mobility difference if wrong flow direction is used. A value between 0 (infinite difference allowed) and 1 (no difference allowed).	0.0001

Example, relax solver accuracy:

```
tollit
*tollit itmax
1e-10  1000 /
```

111. upstrc

Section: **compsect**

Description: Sets upstream option for species transport. The two-point upstream algorithm can be used to reduce numerical dispersion of species fronts.

A single record terminated with a slash ‘/’.

Item	Name	Description	Default
1	*iupstrC*	Indicates algorithm used for species transport, options: 1: single-point upstream concentration used 2: two-point upstream algorithm used for species transport	1

In a 1D case with flow in the *x*-direction, the change in overall species concentration in a grid cell *i* during time-step (*n*+1) computed using an explicit scheme is:

$$\Delta C^{n+1}_{t,kc,i} = \Delta t^{n+1} \sum_{L} \left(q^{n+1}_{L,i-1} C^{n}_{L,kc,i-1} - q^{n+1}_{L,i} C^{n}_{L,kc,i} \right), \tag{5.112}$$

where $C^{n}_{L,kc,i}$ indicates the concentration of component *kc* in phase *L*, cell *i* at the old time-step *n* and $q^{n+1}_{L,i}$ denotes the corresponding phase flow rate between cell *i* and (*i*+1). If the two-point upstream algorithm is activated for species transport the concentration used in transport between cell *i* and (*i*+1), denoted $C_{i+1/2}$ is computed:

$$C_{L,b} = C_{L,i} + \frac{(C_{L,i} - C_{L,i-1}) V_{L,i}}{V_{L,i-1} + V_{L,i}}, \tag{5.113}$$

$$C_{L,i+1/2} = \begin{cases} \min\left(C_{L,b}, \max\left(C_{L.i}, C_{L,i+1}\right)\right), & C_{L,b} > C_{L,i} \\ \max\left(C_{L,b}, \min\left(C_{L,i}, C_{L,i+1}\right)\right), & C_{L,b} < C_{L,i} \end{cases},$$

The ‘boundary’ concentration C_b is a linear extrapolation of the two cell concentrations based on their pore volumes. The last restrictions help to avoid some unphysical behaviour and reduce noise.

Note: The two-point upstream algorithm is only applied if the explicit transport algorithm is used, i.e., *ctransport* = 0. If the automatic transport option *ctransport* =1, then the two-point upstream algorithm may be applied to cells with low flow-through.

The time-step stability limit for the explicit transport algorithm is reduced with a factor 0.66 if the two-point upstream algorithm is used. The parameter *qxemax* (6th item in keyword *timestep*) should be reduced to less than 0.75.

Example, turn on 2-point upstream weighting algorithm:

```
upstrc
*   iupstrC
2       /
```

112. upstream

Section: **compsect**

Description: Checking of correct upstream in final solution.

Item	Name	Description	Default
1	*CheckFlag*	Indicates if upstream checking is performed, options: 0: not used 1: upstream checking is performed	1
2	*dSerrmax*	Saturation error tolerance for use of wrong flow direction.	1E-6
3	*Flim*	Tolerance limit for phase mobility if wrong flow direction is used.	0.1

Upstream phase mobilities are obtained from phase flow directions from the previous time step *n*. In the new pressure solution for time step (*n*+1), some flow directions may have changed. If *upstr_Check*Flag=1, the solution is checked and pressure computation repeated using corrected upstream mobilities if one of the two conditions occurs:

- error in saturation change exceeds *dSerrmax*
- $\lambda_{corr} \cdot \mathrm{Flim} < \lambda_{old}$ or $\lambda_{corr} > \lambda_{old} \cdot \mathrm{Flim}$

The last condition secures that there cannot be flow out of a cell when phase mobility is zero.

Example, relax solver accuracy:

```
upstream
*   upstr_checkFlag, upstr_dSerrmax, upstr_Flim
1                 1e-4            0.1 /
```

113. vtkprint

Section: **compsect**

Description: Flag indicating if vtk files will be printed at report times. Vtk-files can be plotted using Paraview. In present version, incomplete grid information is printed, and only 1D plot using cell number is possible.

Item: *vtk_prF* followed by a slash ‘/’. Options:

0: no printout (default)
1: a vtk-folder is created with grid property data printed every 1/100 of total simulation time.

Example: turning printout of vtk-files on

```
vtkprint
1 /
```

5.8 WELLSECT

Defines wells and injection/production schedules.

wellsect keywords:

Keyword	Short description
bctab	Tabular well constraints
btupdate	update boundary temperatures
dtime	Sets incremental simulation times (applies to printouts)
elrbc	Sets boundary conditions for electric resistance computations.
history	Updating history printout frequency (see **compsect**)
overburden	Updates overburden pressure.
pcwell	Sets boundary capillary pressure (different from zero)
time	Sets cumulative simulation times (applies to printouts)
timestep	Updating timestep restrictions (see **compsect**)
welldef	Defines new wells, must be defined for all wells before boundary types are specified
wellgfact	Off-centre location of well within grid-cell.
wellopen	Opens well
wellshut	Shuts well
wimb	Sets spontaneous imbibition constraints
winject	Sets injection well constraints
wprobe	Phase selective probe used for pressure measurements
wprod	Sets production well constraints

114. bctab

Section:	wellsect
Description:	Allows entering changing well constraints in form of a table instead of repeating well keywords like winject, wprod or wimb. Variants of the keyword indicates which property to be read:
bctabp:	*pressure*, limiting well pressure
bctabq:	*rate*, limiting well rate
bctabT:	*temperature*, boundary or well temperature

Input: A single record terminated with a slash (‘/’) containing one or more well names for which the new boundary conditions will be applied. Alternatively, the model boundary temperature can be modified with this keyword using mnemonics *bcT* (after keyword *bctabT*).

The record must be followed by a table terminated with a slash (‘/’) on a new line. The table must have two columns, with time in the first and the well constraint in the second column. The time in the table is assumed to start from the appearance of the keyword (table should start with *t*=0). The well constraint will be updated from the table using the integrated average value for the present timestep, until the maximum time entrance in the table is exceeded or if boundary conditions for the well has been updated using any of the available well keywords.

Units: User defined, e.g., [days] and [bar] or [m³/d] depending on selected property.

Note: Before using this keyword, full boundary conditions must first have been defined using one of the well keywords winject, wprod, wimb or wprobe.

Example, applying tabular variation of injection rate from time 20 to 70, thereafter injection rate is 90 m³/d (last table entrance). Also, injection temperature and model boundary temperature are changed using *bctabT* keyword:

```
winject
*name   r/p rlim plim cond hzw0
winj    r   40   250 /
* fw fg T°C
1  0  15 /     injects water with temperature 15 °C
* Cw
SSW 1 / brine
/ no oil injected
/ no gas injected
/ ends keyword

time
10 20 / report times

winject
*name   r/p rlim plim cond hzw0
winj    r   50   250 /
* fw fg T°C
1  0  15 /
*  Cw
SSW 1 / brine
/ no oil injected
/ no gas injected
/ ends keyword

bctabq
*names
winj /  rate for winj (50) is overwritten
*t    q
  0  50
 10  60
 20  80
 30  70
 40  60
 41  0
 42  60
 50  90
/
bctabT
*names
winj bcT / use table to change injection (winj) and boundary (bcT) teperature
```

```
*t  °C
0   15
10 15
20 20
30 40
40 40
/
time
30 50 70 90 /
```

115. btupdate

Section: wellsect

Description: Updates temperature at model boundaries. Reads a single value which must be followed by a slash ‘/’. This option requires temperature calculations to be activated (heatsect) or else this value is ignored. The use of this keywords allows simulating heat changes in laboratory experiments. The speed at which the core temperature changes can be tuned using external boundary blocks and boundary conductivity factors. Unit: [°C].

Example:

```
BTupdate
90 / Change temperature of surroundings to 90 °C
```

116. dtime

Section: wellsect

Description: Reads incremental time(s) in user defined units, e.g., [days]. Data must end with a slash ‘/’. The keyword triggers printout of a simulation summary report + grid data.

Several entrances are allowed.

Example:

```
dtime
10 2*20 10 3*50 / total simulation time is now: previous time + 210
```

117. elrbc

Section: wellsect

Description: Keyword for defining boundary conditions for electric resistance computations, defining connections of electrodes to the grid and electric potential (voltage). Require specification of brine conductivity using keyword elcond.

Input:

At least two electrodes must be defined. Each electrode requires at least one record ending with a slash ‘/’. All entries for one electrode must finish before proceeding to the next electrode.

An empty record (only blanks before '/') will terminate the keyword.

Item	Name	Description	Default
1	*ne*	Electrode number	-
2	*U*	Electrode potential. Units: [volt].	-
3	*idir*	Identification of surface to which the electrode is connected. 4: open end face x-directed flow, 1st block (*i*=1) -4: open end face in last block (*i*=*nx*) 5: open end face y-directed flow, 1st block (*j*=1) -5: open end face in last block (*j*=*ny*) 6: open end face z-directed flow, 1st block (*k*=1) -6: open end face in last block (*k*=*nz*)	-
4-7	*i1*, *j1* or *k1*	Item 4-7 are cell indexes describing the range of surface cells connected to the current electrode. Depending on *idir*: *idir*=4 (-4): j1 j2 k1 k2 *idir*=5 (-5): i1 i2 k1 k2 *idir*=6 (-6): i1 i2 j1 j2 Note that if first value is -1, the total surface is connected to the electrode and item 5-7 are not used.	-
8	*i0*, *j0* or *k0*	Optional index to define a boundary within the grid at the indicated layer (index in *x*-, *y*- or *z*-direction). E.g, with *idir*=4, *i0* will be used if a value between 1 and *Nx* is read.	0

Example, 6 electrodes are defined along a radial model with 20×1×400 cells:

```
elrbc
*   ng  U idir j1 j2 k1 k2 i0
 1    0 6    -1 /              1st electrode connected to the end
 1    0 -4   1  1  1  7 /      and at the outer ratius of layer 1-7
 2    1 -4   1  1  64 73 /     2nd electrode connected to outer radius layer 64-73
 3    2 -4   1  1  130 139 /
 4    3 -4   1  1  196 205 /
 5    4 -4   1  1  262 271 /
 6    5 -4   1  1  328 337 /
 7    6 -4   1  1  394 400 /
 7    6 -6   -1 /
/
```

118. overburden

Section: wellsect

Use: Updates overburden pressure.

Description: Use this keyword to include changes in absolute overburden pressure during a run. The keyword can be repeated.

Data is on a single record that must end with a slash '/'. An empty record (only blanks before '/') will terminate the keyword.

Item	Name	Description	Default
1	*dPob*	Change in overburden/confinement pressure compared to initial conditions. Units: user defined, e.g., [bar]	0
2	*Fpob*	Flag indicating if rock compression due to *dPob* shall depend on saturation. Saturation dependency may be relevant in special experiments like porous plate drainage where changes in surrounding oil pressure act as increased confinement pressure for the porous disk, but has less effect on the oil invaded core. Options: 0: no effect of saturation 1: effective *dPob* dpends on S_w 2: effective *dPob* dpends on S_o	0
3	*tramp*	Ramp time until the new conditions are fully applied. A linear change versus time is assumed for *dPob*. If a negative value is read, a zero ramp time is used. Units: user defined, e.g., [hours]	-1

Comments:

Changing the overburden pressure will have an effect on the rock pore volume if rock compressibility is activated. The pressure dependent porosity is computed:

$$\phi = \phi_0 \cdot e^{(p-p_0-dP_{nob})n_{cr}}, \tag{5.114}$$

$$dP_{nob} = f_{ob} dP_{ob}, \; f_{ob} = \begin{cases} 1 & ,Fpob = 0 \\ S_w & ,Fpob = 1 \\ S_o & ,Fpob = 2 \end{cases}$$

Index 0 indicates properties at reference conditions (p_0) and n_{cr} is the rock compressibility. The pressure used in the compressibility calculations is by default the aqueous phase pressure, but this can be changed using the rockcmpr keyword.

The possibility for defining a saturation dependent increment of net overburden pressure can be used in e.g. porous plate experiments where the core is surrounded partly by the least wetting phase. By e.g. specifying that the water pressure is used, the behaviour of kaolin paste typically used to ensure capillary contact between core and porous disk, can be mimicked by assigning a high compressibility to part of the model.

Example: Increasing the overburden in two steps.

```
overburden
*dPob   Fpob
1      1 /
time
1 /
overburden
*dPob   Fpob
5      1 /
```

```
time
5 /
```

119. pcwell

Section: wellsect

Use: Sets boundary capillary pressure for produced phase. Can be used if this value is significantly different from zero (default assumption).

Description: In spontaneous imbibition experiments, produced phase appears as growing bubbles on the rock surface. The actual boundary pressure is therefore not zero, but will fluctuate between a minimum values corresponding to the snap-off droplet size ($Pcb=2\sigma/R_{max}$) and the maximum droplet curvature ($Pcb_{max}=2\sigma/R_{min}$). If produced phase is a (partly) wetting phase, this boundary *Pcb* is assumed to be small (a few Pa) and can be ignored. If produced phase is a (strongly) non-wetting phase, *Pcb* can be quite significant. It is assumed that the value should be approximately the *Pc* level measured in imbibition *Pc* just as snap-off occurs (and non-wetting phase becomes discontinues in the pore network). Observations from mercury measurements indicate a value of approximately 30% to 50% of rock threshold *Pc* measured in drainage *Pc*.

Data is on a single record that must end with a slash '/'. An empty record (only blanks before '/') will terminate the keyword.

Item	Name	Description	Default
1	*wname*	Well name used to identify the well.	-
2	*Pcboil*	Boundary *Pcb* for the oil phase. Units: user defined, e.g., [bar], or dimensionless if J-scaling is defined.	-
3	*Pcbwater*	Boundary *Pcb* for the water phase. Units: user defined, e.g., [bar], or dimensionless if J-scaling is defined.	-
4	*SwPcbo*	The oil *Pcb* is only applied if $S_{w,well} > SwPcbo$. Units: [vol. fraction]	0.9
5	*SwPcbw*	The aqueous *Pcb* is only applied if $S_{w,well} < SwPcbw$. Units: [vol. fraction]	0.1

Notes:

- *Pcb* will only be applied on boundaries (wells) defined as production boundaries using keywords wimb or wprod.
- If boundaries are defined for counter-current flow (spontaneous imbibition boundary), produced phase is assumed to be removed, and only the imbibing phase is in contact with the

surface (see wimb). The saturation limits (*SwPcbo* and *SwPcbw*) are only used if boundaries are of co-current type (see wprod).

- If J-scaling is used and produced phase is very strongly non-wetting, a value in the range 0.05 – 0.15 can be suggested.
- The keyword can be repeated to change production constraints during the simulation period. Current constraints apply from the last *time* or *dtime* read.

Example: Spontaneous imbibition of water (in strongly water-wet core) from core inlet end, while outlet end is in contact with only oil. Flow is in x-direction. Pcb defined only at the inlet boundary.

```
welldef
*name pcflag (first boundary)
Inlet  1 /
*dir i1  j1   k1  k2   rw  open
4   -1 /  All boundary blocks at inlet end are open.
/

welldef
*name pcflag (second boundary)
Outlet  1 /
*dir i1  j1   k1  k2   rw  open
   4   -1 /  All boundary blocks at oulet end are open.
/

wimb
*name   phase  plim   hz0 : boundary for counter-current flow (spontaneous imbibition)
Inlet    w     1.0    0.5 /
*  Cw
1 /  Ci water
/
/  terminate keyword
wprod
*name   r/p rlim plim hz0  : boundary for co-current production
Outlet   p   500  1.0  0.5 /
/

pcwell
*name Pcboil Pcbwater SwPcbo SwPcbw :
Inlet 0.1    0        0.9    0.1 /
/

time
1  5 10 20 30 60 120 240 360 720 1080 1440  / 1440 min = 24 hrs

end
/
```

120. time

Section: wellsect

Description: Reads cumulative times in user defined units, e.g., [days]. Data must end with a slash '/'. The keyword triggers printout of a simulation summary report + grid data.

Several entrances are allowed, but the numbers must be increasing.

Example:

```
time
10 20 50 100 /
```

121. welldef

Section: wellsect

Use: Mandatory keyword for defining wells. The keyword must be repeated *nwell* times.

Description: At least two records are read.

The first record must contain a well name, which will later be used to identify the well, and for reporting purposes. The record must end with a slash '/'.

The next record(s) will contain well connection data. Each record must be terminated with a slash '/'. An empty record (only blanks before '/') will terminate the keyword and data entry for the current well.

The exact content of the record will depend on the well penetration direction. The case with the well penetrating in the *z*-direction is described below.

The first record has two items:

1st record

Item	Name	Description of 1st record	Default
1	*wname*	Well name, must be unique. Case sensitive. Will be used to identify the well in subsequent input.	-
2	*PcFlag*	Flag indicating if capillary pressure should be used in the well connection term. Options: 0: Not used 1: Capillary pressure will be used. In multi-phase injectors, fractional flow weighted capillary pressure will be used. 2: Capillary pressure will be used. In multi-phase injectors, well pressure will be the phase with the highest pressure.	0
3	*WdFlag*	Well density flag indicating how fluid density in well is computed which affect the vertical pressure gradient in the well. This flag only influence production wells and only if well extends vertically. Options: 0: Use an average fluid density for well. 1: Fluid density varies along well according to average upstream inflow. This option gives more correct vertical gradient in computed well pressure. The well section inflow order assumed is the order specified in the next record(s) where specification of well connections is assumed to start with the top perforation (closest to the wellhead). If open end face boundary conditions is specified (see next record)	1

2nd record (repeated until well description is complete)

Item	Name	Description	Default
1	*idir*	Well penetration direction for the current block(s): 1: *x*-direction 2: *y*-direction 3: *z*-direction (vertical well) must be used in radial grid 4: open end face x-directed flow, 1st block (*i*=1) -4: open end face in last block (*i*=*nx*) 5: open end face y-directed flow, 1st block (*j*=1) -5: open end face in last block (*j*=*ny*) 6: open end face z-directed flow, 1st block (*k*=1) -6: open end face in last block (*k*=*nz*)	-
2	*iw*	Note: item 2-5 depends on the value of *idir* (1st item). This is described below. The following description is valid for *idir*=3, where the 2nd item read (*iw*) represents the *x*-coordinate index for current well connection(s). Range: 1-*nx* (must be one for the first well in radial model)	-
3	*jw*	*y*-coordinate index for current well connection(s). Range: 1-*ny* Special: if *idir* = 3, for radial wells with multiple secors (*ny* > 1), setting *jw* = 0 will automatically connect all sectors to the well without more input. Otherwise, a new record for each sector that should be connected is required. (An alternative input method is to use *idir* = 4).	-
4	*k1*	First *z*-coordinate index for current set of well connections	-
5	*k2*	Second *z*-coordinate index for current set of well connections	-
6	*rw*	Well radius. Units: user defined, e.g., [m]. Note: Not used for 1st well in radial geometry. See below. If zero, open end faces will be used as e.g. in core floods. The face normal to the indicated well direction is used (e.g. xy-face is used if *idir*=3). If *idir* ≥ 4, item 6 will change, see description below.	0
7	*iopen*	Flag indicating if current well connection(s) is open: 0 : closed 1 : open Note: this option is not working. Opening and closing of well should be done using the *wellopen* and *wellshut* keywords.	1
8	*skin*	Skin factor for current well. Note: currently not used.	0

By repeating the record, the well can be perforated in arbitrary blocks in different directions, e.g., letting the first part be vertical and last part be horizontal. The first five items read for different well penetration directions are:

```
*   idir=1   i1    i2   jw   kw   / x-direction
*   idir=2   iw    j1   j2   kw   / y-direction
*   idir=3   iw    jw   k1   k2   / z-direction
```

Open end face boundary conditions are easily set using *idir* = ±[4, 6]. Such boundary conditions are typical in laboratory experiments. If rectangular grid is used, one might use *idir*=4 to indicate inlet boundary and -4 to indicate the outlet boundary. The *x*-coordinates is the automatically set to *i*=1 for the inlet boundary and to *i*=*nx* for the outlet boundary. The next four parameters (item 2-5) describe the part of the *yz*-surface open for flow:

```
*   idir= 4   j1 j2   k1 k2 i0 / x-directed flow with boundary at the inlet of i0
*   idir=-4   j1 j2   k1 k2 i0 / x-directed flow with boundary at the outlet of i0
*   idir= 5   i1 i2   k1 k2 j0 / y-directed flow with boundary at the inlet of j0
*   idir=-5   i1 i2   k1 k2 j0 / y-directed flow with boundary at the outlet of j0
*   idir= 6   i1 i2   j1 j2 k0 / z-directed flow with boundary at the inlet of k0
*   idir=-6   i1 i2   j1 j2 k0 / z-directed flow with boundary at the outlet of k0
```

Note that entering zero or a negative integer for the first surface coordinate (2nd item), the remaining values will not be used. All the surface blocks will be open (with *idir*=4: *ny*·*nz* surface blocks will be used). If *idir* is positive, the boundary will be at the inlet (first block) in the indicated direction. If *idir* is negative, the boundary will be at the outlet (last block) in the indicated direction (*nx*, *ny* or *nz*). If 5th item is omitted or negative, the first or last block in indicated direction is used.

```
*   idir   j1 j2   k1 k2   / x-directed flow with boundary at the beginning (i=1)
4    -1 / j1 < 0: all surface blocks will be opened (j1=k1=1, j2=ny, k2=nz)
```

The well direction determines how the well connection factor pi_{kw} is computed.

Note: If radial geometry is used with ***R0*** > 0, the first well is assumed radial and must be located in gridblock (1,1,*k*) where *k* is the index in *z*-direction (layer index). If more than one sector is used (*ny* > 1), only the first sector index (*jw*=1) should be specified, the well will automatically be perforated in the remaining sectors.
The *rw* entered here will be owerwritten with ***R0*** entered in the grid dimension section.
Additional wells are specified as with Cartesian grid.

Example:

```
welldef
injector  /  well name and no capillary pressure in well term (default)
*   idir   iw   jw  k1   k2   rw    iopen  skin
3      1    1    1   5    0.1   1       0   /  vertical
*   idir   i1   i2  jw   kw   rw    iopen  skin
1      2    10   1   5    0.1   1       0   /  horizontal x-direction
/  finished well data
welldef
producer  /  well name
*   idir   i1   i2  jw   kw   rw    iopen  skin
1      10   20   20  2    0.1   1       0   /  horizontal x-direction
/  finished well data
```

---

122. wellgfact

Section: wellsect

Description: Allows specification of a well at the grid boundary or corner. This option can be used in cases with a repetitive pattern, e.g., in a five-spot pattern.

Data is entered for one well at the time on a single record that must end with a slash '/'. The record can be repreated for more wells. An empty record (only blanks before '/') will terminate the keyword.

Item	Name	Description	Default
1	*wname*	Well name used to identify the well.	-
2	*Fwg*1	Geometric location of well within the grid cell in the first direction normal to the wellbore, which would be the *x*-direction for a well penetrating the grid cell in the *z*-direction (*idir* =3 in welldef) *Fwg*1 = 1-(*x*-distance between cell centre and well)/Δ*x* Units: dimensionless.	1
3	*Fwg*2	Geometric location of well within the grid cell in the second direction normal to the wellbore, e.g., the *y*-direction. *Fwg*2 = 1-(*y*-distance between cell centre and well)/Δ*y* Units: dimensionless.	1
4	*Fwg*3	Additional well productivity multiplier. Units: dimensionless.	1

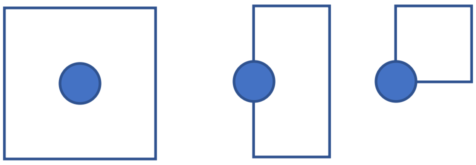


Figure 5.12 Examples of well positions within grid cell. From left: well in centre of cell (f_{wgx}, f_{wgy}) = (1, 1), well on left boundary (f_{wgx}, f_{wgy}) = (0.5, 1) and well in the corner (f_{wgx}, f_{wgy}) = (0.5, 0.5).

Off-centre location of well in grid block can be specified with factors f_{wg1} and f_{wg2}. The third factor f_{wg3} is just a well connection index multiplier. The computations of R_0 and T_{wb} (well connection term) are (for a vertical well):

$$T_{wb,k} = f_{wgb} \cdot \frac{2\pi k \Delta h_k}{\ln\left(\frac{R_0^*}{R_w}\right)+S}, \quad f_{wgb} = f_{wg1} f_{wg2} f_{wg3}, \tag{5.115}$$

$$R_0 = 0.28 \frac{\left(\left(\frac{\Delta x}{\mathrm{fwg1}}\right)^2 \left(\frac{k_y}{k_x}\right)^{0.5} + \left(\frac{\Delta y}{\mathrm{fwg2}}\right)^2 \left(\frac{k_x}{k_y}\right)^{0.5}\right)^{0.5}}{\left(\frac{k_y}{k_x}\right)^{0.25} + \left(\frac{k_x}{k_y}\right)^{0.25}} \cdot exp\left(-\frac{(f_{wg1}+f_{wg2}-2)\pi}{8}\right),$$

$$R_0^* = \sqrt{R_0^2 + R_w^2}.$$

An approximate correction of the well connection factor if the geometric location of the well is at the edge or corner of the grid (see Figure 5.13) is obtained by setting f_{wg1}=0.5 and f_{wg2}=1 for the W_1-case and f_{wg1}=f_{wg2}=0.5 for the W_2-case. If well is located on the border between two cells as indicated with case W_3, the well can be perforated in both cells in welldef and f_{wg1}=0.5 and f_{wg2}=1 should be used.

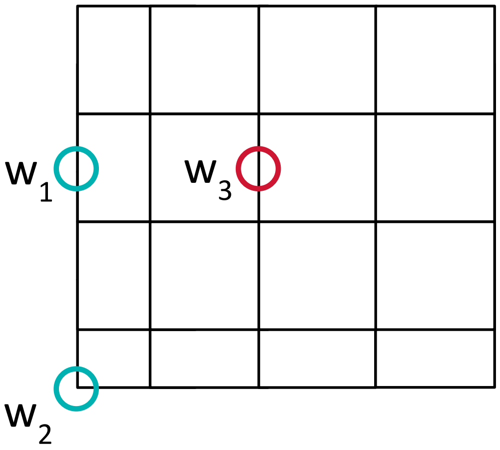


Figure 5.13 Well placement at the edge (W_1) or corner (W_2) of the grid, or at the boundary between two cells /W_3).

While the two first factors (f_{wg1} and f_{wg2}) addresses the off-centre position of a well penetrating a block (defined with *idir*=1, 2 or 3 in welldef), all the factors (including the third factor f_{wg3}) will modify the well transmiscibility T_{wb} as indicated in Eq. (5.115) also with other boundary conditions (*idir* ± 4, 5, 6).

Example in a 10×11×5 grid with two corner injectors and one boundary production well:

```
Welldef
*wname Pc-flag
WI_1   1 /  well name
*  idir  iw  jw  k1  k2  rw   iopen  skin
3     1    1   1   5   0.1   1      0  /  vertical in corner (1,1)
/
welldef
*wname Pc-flag
WI_2   1 /  well name
*  idir  iw  jw  k1  k2  rw   iopen  skin
3     1    11  1   5   0.1   1      0  /  vertical in corner (1,11)
/
welldef
*wname Pc-flag
WP_1   1 /  well name
*  idir  iw  jw  k1  k2  rw   iopen  skin
3     10    6   1   5   0.1   1      0  /  vertical at boundary (10,6)
/  finished well data
Wellgfact
*wname  fwg1 fwg2
WI_1     0.5  0.5 /
WI_2     0.5  0.5 /
WP_1     0.5  1.0 /
/
```

123. wellopen

Section: wellsect

Description: A list of well names on a record terminated with a slash ‘/’.

The wells will be opened starting from the last *time* or *dtime* read. The keyword can be repeated to change injection constraints during the simulation period. Current constraints apply from the last *time* or *dtime* read.

Example (shuts two wells for ten days, and reopens):

```
time
10 20 30 /
wellshut
winj wprod /
time
40 /
wellopen
winj
wprod
/
*  old constraints from before shutting the wells are re-applied when opened
```

124. wellshut

Section: wellsect

Description: A list of well names on a record terminated with a slash ‘/’.

The wells will be shut starting from the last *time* or *dtime* read. Printouts of well history is suppressed while wells are shut. The keyword can be repeated during the simulation period.

Example (see wellopen above).

125. wimb

Section: wellsect

Use: Sets boundary constraints for spontaneous imbibition and defines composition of imbibing fluid.

Description: Reads boundary constraints, what phase to imbibe and its composition. Three records are read. Each record must end with a slash ‘/’.

The input can be repeated for additional wells. The keyword is terminated with an extra slash ‘/’ on a new line.

1st record			
Item	Name	Description	Default
1	*wname*	Well name used to identify the well.	-
2	*imbphase*	Imbibing phase, only 1st letter is used. Options:	

		‘w’ or ‘W’: aqueous phase ‘o’ or ‘O’: oleic phase	-
3	*plim*	Boundary pressure in imbibing phase. (BHP) Units: user defined, e.g., [bar]	-
4	*hzw*0	Reference depth for *plim* (BHP). Default value is previous input or depth of the 1st grid cell connected. If a value less than -9999 is entered, the default value is used. If more than one boundaries (wells) is defined, consistent boundary conditions is easily obtained by using the same *plim* and *hzw*0 for all. Units: user defined, e.g., [cm]	1st cell
5	*gelmode*	Parameter indicating that spontaneous imbibition is from a gel explicitly present in the neighbour cell, see the *gelnum* keyword. If activated, water flux across this boundary is set to zero. Options: 0: Normal spontaneous boundary 1: Imbibing fluid (normally water) connection factor is set to zero, while the non-imbibing fluid (oil) connection factor is kept.	0
6	*tramp*	Ramp time until the new well conditions are fully applied. A linear change versus time is assumed for *rlim* and *plim*. If a negative value is read, a zero ramp time is used. Units: user defined, e.g., [hours]	-1
2nd record			
-	*C(i)* Alternatively *name, Cw(i)*	Composition of water phase, $i = 1, nt$. Up to *nt* (number of components) values are read. The record must be terminated with a slash ‘/’. Remaining items not read will be zero. Alternative input method is to indicate which component to read by its name, followed by its concentration. This is repeated for all positive concentrations, followed by a ‘/’. Units: volume fraction	0
3rd record			
1	*Vimbcell*	Volume of imbibition cell (void volume not including the core plug). A positive value triggers mass balance calculations for component exchange with the core, both advective and diffusive exchange. History file for cell is printed. Units: user defined e.g. [ml]	0
2	*wtemper*	Boundary temperature. Only used if the temperature calculations are activated by the presence of the heatsect section. Default value is the previous input or grid boundary temperature if this is the first appearance of this keyword for the current well. Note that heat exchange between model and surroundings are only through conductions. Fluids entering or	20

		leaving the core is assumed to have the same T as the boundary block. Units: [°C]	
3	*iupdat*	Flag indicating if concentrations in *Vimbcell* should be updated. Applies only if *Vimbcell* >0. Options: 0: Fluids in imbibition cell will gradually be replaced using flow through rate q_{wcell} 1: Fluids in imbibition cell are replaced	1
4	*qwcell*	Imbibition cell flow through rate, suitable for high temperature, high pressure experiments. Units: user defined e.g. [ml/min]	0
5	*Sini_wcell*	Initial saturation of imbibing phase in cell. If a value <0.99 is entered, no fluid is assumed to leave the cell. The previous rate parameter *qwcell* will not be used. This option mimics experiments under atmospheric pressure.	1

Notes: 1. If sum of concentrations for volumetric components are different from one, then the concentrations will be normalised.

2. The keyword can be repeated to change injection constraints during the simulation period. Current constraints apply from the last *time* or *dtime* read.

Example: Spontaneous imbibition of WATER. Definition of boundary condition for a radial core model. Boundary temperature is first defaulted (to that set in the initsect or heatsect) and later changed to 80 °C.

```
welldef
*name pcflag
winj1  1 /
*dir cell coordinates … (radial core model)
4   -1 / outer radial boundary
6   -1 / inlet end of core
-6   -1 / outlet end of core
/

wimb
*name   phase  plim hz0
winj1    w     10      0 /
*  Cw
1 /  Ci water
*  Vimbcell Temper (start with boundary T equal to initial Temperature)
/
/  terminate keyword
time
0.01 0.05 0.1 0.15 0.2 0.3 0.5 1 2
4 6 8 10 12 16 20 24 /  cumulative times for printout

wimb
*name   phase  plim hz0
winj1    w     10      0 /
*  Cw
1 /  Ci water
*  Vimbcell Temper
0.   80.0  /  Boundary T changed to 80°C
/  terminate keyword
dtime
2*0.01 0.08 3*0.1  3*0.2 1 1 1 2 3*6 4*12 / Incremental time steps for reporting
```

126. winject

Section: wellsect

Use: Sets well injection constraints and defines composition of injected fluid(s).

Description: Reads injection well constraints, what phase to inject and the composition of injected phases. Five records are read. Each record must end with a slash ‘/’.

The input can be repeated for additional wells. The keyword is terminated with an extra slash ‘/’ on a new line.

1st record			
Item	Name	Description	Default
1	*wname*	Well name used to identify the well.	-
2	*flag*	Flag indicating primary constraints for this well. Options: ‘r’ or ‘R’: rate constraints ‘p’ or ‘P’: pressure constraints	-
3	*rlim*	Upper rate limit (positive value). Units: user defined, e.g., [m^3/d]	-
4	*plim*	Upper pressure limit. (BHP) Units: user defined, e.g., [bar]	0
5	*cond*	Flag indicating if concentrations and fractional flows are specified at reference (*P*, *T*) or surface conditions. Options: r : volumes (concentrations) at reference (*Pref*, *Tref*) s : volumes (concentrations) at surface conditions, (option may not work properly, not recommended)	r
6	*hzw*0	Reference depth for *plim* (BHP). Default value is previous input or depth of the 1st grid cell connected. If input value is less than -9999, default value will be used. Units: user defined, e.g., [m]	1st cell
7	*ramptime*	Allows to specify a time before the new well conditions are fully applied. A linear change versus time is assumed for *plim*, *rlim*, and phase flow rates. New phase concentrations and temperature are applied instantly. If a negative value is read, a zero ramp time is used. Units: user defined, e.g., [hours]	-1
2nd record			
1	*fw*	water fraction	-
2	*fg*	gas fraction	0

3	*wtemper*	Well injection temperature. Only used if the temperature calculations are activated by the presence of the heatsect section. Default value is the previous input or 20°C if this is the first appearance of this keyword for the current well. Change of boundary temperature can be done using keyword btupdate. Units: [°C]	old value or 20°C
3rd record			
-	*Cw(i)* Alternatively *name, Cw(i)*	Composition of water phase, *i* = 1, *nt*. Up to *nt* (number of components) values are read. The record must be terminated with a slash '/'. Remaining items not read will be zero. Alternative input method is to indicate which component to read by its name, followed by its concentration. This is repeated for all positive concentrations, followed by a '/'.	0
4th and 5th records: both are required, oil and gas phase composition - see 3rd record			

Notes:

- If sum of concentrations for volumetric components is different from one, then the concentrations will be normalised.
- If the well is injecting at a constant rate equal to *qlim* and the pressure exceeds *plim*, then the well will automatically switch to pressure constraints, *plim*.
- If the well is injecting at a constant pressure equal to *plim* and the rate exceeds *qlim*, then the well will automatically switch to rate constraints, *qlim*.
- The keyword can be repeated to change injection constraints during the simulation period. Current constraints apply from the last *time* read.

Example: assuming one WATER, one OIL and one GAS component, the last named CH4.

```
winject
*name   r/p rlim plim cond hzw0
winj     r    50   250   r   /
*   fw  fg  T°C
1   0   15 /     injects water with temperature 15 °C
*   Cw
1 /
0  1  0 /
0  0  1 /
/

time
10 /
*   new well contraints starting from t=10
winject
winj  r 100 250 /
0  1  / fw fg changed to inject gas  (injection temperature is still 15 °C)
1  / water phase conc
/   oil phase conc not needed because fo=0
CH4  1 /  (used in stead of 0  0  1 /)
/
```

127. wprobe

Section: wellsect

Use: Defines a phase selective probe used for pressure measurements, e.g., for measurement of the water phase pressure in a core. The pressure in the probe chamber can lag behind the actual in-situ core pressure due to compressibility of fluids and the chamber volume. This keyword triggers the probe model which handles the compressibility driven fluid exchange between the probe chamber and the core.

Description: Reads probe properties on up to four records, each record must end with a slash '/'. The input can be repeated for additional wells. An empty record (only blanks before '/') will terminate the keyword.

1st record

Item	Name	Description	Default
1	*wname*	Well name used to identify the well.	-
2	*iupdate*	Flag indicating if this is an update. Options: 0 : Required option the first time this keyword is read for a 'well'. Reads all probe properties (and overwrite existing properties if used more than once for a 'well'). 1 : Updates a limited number of properties.	0

The next three records are read if *iupdate* = 0:

2nd record with *iupdate* = 0

Item	Name	Description	Default
1	*L*	A single letter to identify the probe fluid phase. Options: w : The probe has contact with the water phase in the core. Only water can enter or leave the probe. o : Defines an oil phase probe.	-
2	*V_probe*	Volume of probe chamber. Units: user defined, e.g., [cm³]	-
3	*kh_probe*	Defines the ceramic disk conductivity. $kh_{probe} = k_{disk}A/L$, with *A* and *L* representing the disk area and thickness. Units: user defined, e.g., [md·cm]	-
4	*bcell_nc*	Compressibility coefficient for the probe chamber volume. Units: user defined, e.g., [bar^{-1}]	0
5	*bcell_nT*	Thermal expansion coefficient for the probe chamber volume. Only used if temperature calculation is activated. Units: user defined, e.g., [bar^{-1}]	0
6	*hzw0*	Reference depth for probe pressure. Default value is previous input or depth of the 1st grid cell connected. If entered value is less than -9999, the default value is applied.	1st cell

3rd record with *iupdate* = 0

Item	Name	Description	Default
1	*pncp0*	Reference pressure for probe chamber compressibility Units: user defined, e.g., [bar]	-
2	*p_probe*	Initial probe chamber pressure. Units: user defined, e.g., [bar]	-
3	*T_probe*	Probe temperature. If value read is zero or negative, the probe temperature is set equal to the temperature in the 1st connected cell.	-9999
4	*bcell_dV*	Optional additional volume injected into the probe chamber. Units: user defined, e.g., [cm³]	0
5	*bcell_Vdt*	Time interval for including *bcell_dV*. Units: user defined, e.g., [min]	0
6	*Tcon*	Temperature conductivity parameter used if probe temperature is changing using keyword bctabT. If *Tcon* > 0 is entered, the temperature for the new timestep (n+1) is: $T^{n+1} = T^n + (T_b - T^n) \cdot \exp(-\Delta t \cdot T_{con})$. Units: user defined, e.g., [hr^{-1}]	0

4th record with *iupdate* = 0

Item	Name	Description	Default
1	*name c*	Composition of fluid in the probe in form of an alternating list of a component name followed by its concentration. The list of alternating name/concentration is terminated by a '/'. Components not present in the list are assigned zero concentration.	-

The next record is read if *iupdate* = 1:

2nd record with *iupdate* = 1

Item	Name	Description	Default
1	*p_probe*	New probe chamber pressure. If a value converted to less than -9 atm is read, the existing probe pressure is used. Units: user defined, e.g., [bar]	-100 atm
2	*V_probe*	Volume of probe chamber will be updated if *V_probe* > 0. Units: user defined, e.g., [cm³]	0
3	*pncp0*	Reference pressure for probe chamber compressibility. If a value converted to less than -9 atm is read, the old reference pressure is kept. Units: user defined, e.g., [bar]	-100 atm
4	*bcell_nc*	Compressibility coefficient for the probe chamber volume. Only updated if value read here is larger than or equal to zero. Units: user defined, e.g., [bar^{-1}]	-1
5	*bcell_dV*	Optional additional volume injected into the probe chamber.	0

		Units: user defined, e.g., [cm^3]	
6	*bcell_Vdt*	Time interval for including *bcell_dV*. Units: user defined, e.g., [min]	0

A possible use of a water-wet probe to improve experimental results is shown in Figure 5.14.

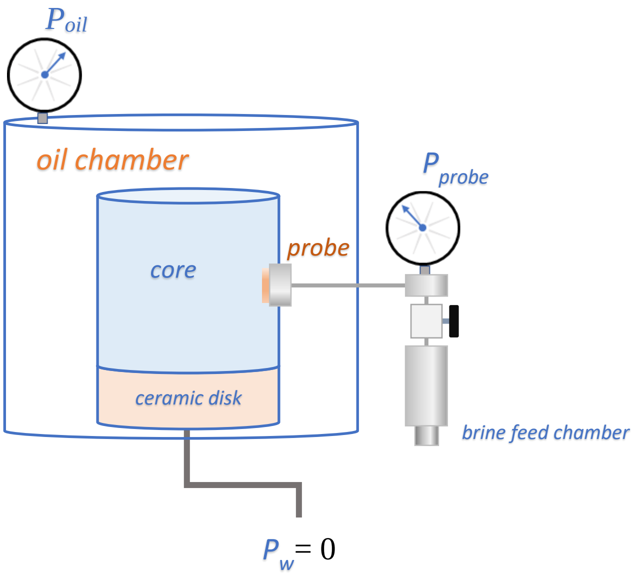


Figure 5.14 Schematic illustration of the use of a water-wet probe to measure in-situ water pressure in a porous plate drainage experiment.

Notes:

- If probe is connected to more than a single cell, the disk conductivity (kh_{probe}) is distributed among the connections based on cell connection factors calculated in absence of the ceramic disc.
- The cell connection factors (cell productivity indexes) are obtained by harmonic averaging of the cell and ceramic disc conductivities.

Example, assuming a radial model (15×8×46 cells representing half (180°) the core in Figure 5.14):

```
*  connect probe to outer radial surface of two cells (yz-surface)
welldef
probe 1 /
*dir j1  j2   k1  k2   i0
 4  1   1    19  20 /
/
*
*
*  probe properties
wprobe
*name iupdat
probe 0 /
*L_pr  Vp_pr  kh_pr nc_pr   nT_pr  hz0
 w     2.55   0.029 6E-5    0      0 /
*pncp0 pinit Tinit
 0     1.0   20  /
*  conc
FW 1 /
/
```

```
time
4 0.5 1 2 3 4 5 6 10  /

*  mimicking a 3 bar increase in surrounding oil pressure by increasing pncp0
*  resulting in a small compression of the probe chamber

wprobe
*name iupdat
probe 1 /
*p_probe, bcell_V0, pncp0, bcell_nc
 -1000       -1000     3 /
/
Time
12 /

*  mimicking a new probe touchdown (TD)with initial pressure equal to Po=4

wprobe
*name iupdat
probe 1 /
*p_probe, bcell_V0, pncp0, bcell_nc
 4      /
/
```

128. wprod

Section: wellsect

Use: Sets well production constraints.

Description: Reads production well constraints on a single record that must end with a slash ‘/’. The input can be repeated for additional wells. An empty record (only blanks before ‘/’) will terminate the keyword.

Item	Name	Description	Default
1	*wname*	Well name used to identify the well.	-
2	*flag*	Flag indicating primary constraints for this well. Options: ‘r’ or ‘R’: rate constraints ‘p’ or ‘P’: pressure constraints	-
3	*rlim*	Upper rate limit (positive value). Units: user defined, e.g., [m^3/d]	-
4	*plim*	Lower pressure limit. (BHP) Units: user defined, e.g., [bar]	0
5	*hzw*0	Reference depth for *plim* (BHP). Default value is previous input or depth of the 1st grid cell connected. If entered value is less than -9999, the default value is applied.	1st cell

6	*tramp*	Ramp time until the new well conditions are fully applied. A linear change versus time is assumed for *rlim* and *plim*. If a negative value is read, a zero ramp time is used. Units: user defined, e.g., [hours]	0

Notes: 1. Well constraints will automatically switch between rate and pressure if the upper *rlim* or lower *plim* limits are exceeded.

2. Well constraints can be changed by repeating the keyword during the simulation period. New constraints apply from the previous *time* value read.

Example:

```
wprod
*name   r/p rlim plim
wprod1  r   500  100 /
/
time
10 /
wprod
*name   r/p rlim plim
wprod1  p   1000 100 /
/
```

6 Technical description

6.1 Boundary connection factors

The boundary phase productivity index PI_k^l for cell k and connection factor $T_{wb,k}$ for a well penetrating a rectangular grid cell in z-direction is computed:

$$PI_k^l = \lambda_l \cdot T_{wb,k} \,, \qquad (6.1)$$

$$T_{wb,k} = \frac{\theta k \Delta z k_h}{\ln\left(\frac{R_0^*}{R_w}\right) + S} \,.$$

For a rectangular grid, $\theta = 2\pi$. The horizontal absolute permeability is computed as $k_h = \sqrt{k_x k_y}$. The block radius R_0 used in the expression for the boundary connection factor is the radius that gives the correct pressure difference between the well and grid blocks outside the well block at steady state with the numerical scheme used for block-to-block flow. The expression derived by Peaceman [12] is used with a small modification:

$$R_0 = 0.28 \frac{\left(\Delta x^2 \left(\frac{k_y}{k_x}\right)^{0.5} + \Delta y^2 \left(\frac{k_x}{k_y}\right)^{0.5}\right)^{0.5}}{\left(\frac{k_y}{k_x}\right)^{0.25} + \left(\frac{k_x}{k_y}\right)^{0.25}} \,, \qquad (6.2)$$

$$R_0^* = (R_0^4 + R_w^4)^{0.25}$$

In the isotropic case with $k_x = k_y$, the expression for R_0 becomes: $R_0 = 0.14\,(\Delta x^2 + \Delta y^2)^{0.5}$. The last term in Eq. (6.2) is a modification of R_0 to avoid problems if small grid blocks are used. The use of R_0^* introduces a small error in computed well pressure but secures that the block radius used in Eq. (6.1) is always larger than the well radius R_w.

If the well penetrates the grid cell in the x-direction, the same equations are used with the appropriate exchange of x and z indices. Penetration of the grid cell in the y-direction is obtained by swapping y and z indices.

If open surfaces are specified as boundary conditions, the flow between the cell and the well becomes linear and the well connection factor to cell k is (assumed cell yz-surface open for flow):

$$T_{wb,k} = \frac{2k \Delta y_k \Delta z_k}{\Delta x_k} \qquad (6.3)$$

6.2 Geometric location of well in block

Off-centre location of well in grid block can be specified with factors f_{wgb1} and f_{wgb2} using keyword *wellgfact*. The computations of R_0 and T_{wb} become:

$$T_{wb,k} = f_{wgb} \cdot \frac{2\pi k \Delta h_k}{\ln\left(\frac{R_0^*}{R_w}\right)+S}, \quad f_{wgb} = f_{wg1} f_{wg2} f_{wg3}, \tag{6.4}$$

$$R_0 = 0.28 \frac{\left(\left(\frac{\Delta \mathrm{x}}{\mathrm{f_{wg1}}}\right)^2 \left(\frac{k_y}{k_x}\right)^{0.5} + \left(\frac{\Delta \mathrm{y}}{\mathrm{f_{wg2}}}\right)^2 \left(\frac{k_x}{k_y}\right)^{0.5}\right)^{0.5}}{\left(\frac{k_y}{k_x}\right)^{0.25} + \left(\frac{k_x}{k_y}\right)^{0.25}} \cdot exp\left(-\frac{(f_{wg1}+f_{wg2}-2)\pi}{8}\right),$$

$$R_0^* = (R_0^4 + R_w^4)^{0.25}.$$

The correct well connection factor if the geometric location of the well at the edge or corner of the grid (see Figure 6.1) is obtained by setting f_{wgb1}=0.5 and f_{wgb2}=1 for the W$_1$-case and f_{wgb1}=f_{wgb2}=0.5 for the W$_2$-case. If well is located on the border between two cells (case W$_3$) f_{wgb1}=0.5 and f_{wgb2}=1 should be used for both cells.

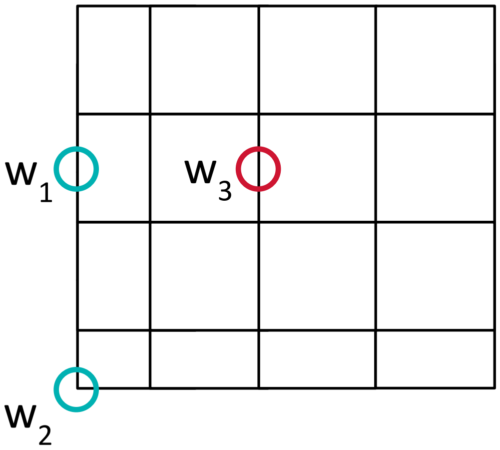


Figure 6.1 Well placement at the edge (W$_1$) or corner (W$_2$) of the grid, or at the boundary between two cells /W$_3$).

6.3 Hysteresis

Saturation functions are in general dependent on the previous saturation history. Saturation history refers to previous saturation reversals, historical maximum or minimum of phase saturations and the saturation change process (drainage or imbibition). Here we use the term drainage to describe the situation of decreasing water saturation and imbibition for increasing water saturation, in-line with the common use of these terms in the oil industry.

The hysteresis mechanisms implemented in IORCoreSim include 1) residual oil saturation (S_{or}) depending on historical maximum oil saturation, 2) relative permeability scanning curve depending on historical minimum water saturation and S_{or} and 3) capillary pressure scanning curve depending on process (drainage or imbibition) and previous saturation reversals. Implementation of hysteresis is activated with keyword hystsat in the runsect section. Parameters for computing S_{or} and the capillary interpolation parameter ε are specified using keyword hystmod in the propsect section.

6.3.1 Land's expression

Two options are available using keyword hystmod in IORCoreSim for computing residual oil saturation (S_{or}) as function of maximum historical oil saturation (S_{ohmax}). One option is free format option where S_{or} versus S_{ohmax} is entered as a table. This option allows non-monotonic behaviour associated with a wettability change where S_{or} goes through a maximum and decrease when S_{ohmax} exceeds a critical level for wettability change to take place. The other option is the expression suggested by Land [15, 16] with a modification:

$$S_{or} = \frac{C_{mL} S_{ohmx}}{1 + C_L S_{ohmx}}, \quad 0 < C_{mL} \leq 1, \tag{6.5}$$

where S_{ohmx} is the maximum historical oil saturation, C_L is the Land's trapping parameter and C_{mL} is a modification parameter increasing the flexibility of the expression. $C_{mL} = 1$ corresponds to the original expression and $C_{mL} = S_{orI}/(1-S_{wr})$ will give a linear relation between residual oil and maximum oil saturation. If we have a non-zero S_{orD} in the primary drainage, the expression will be:

$$S_{or} = S_{orD} + \frac{C_{mL}(S_{ohmx} - S_{orD})}{1 + C_L(S_{ohmx} - S_{orD})}, \quad 0 < C_{mL} \leq 1, \tag{6.6}$$

6.3.2 Capillary pressure hysteresis

The actual curves used in the simulations are scanning curves generated from the boundary curves based on the previous average saturation history and using a modified version of the expression suggested by Killough [13].

$$\begin{aligned} P_c &= P_{c,I} + F\left[P_{c,D} - P_{c,I}\right], \\ F &= F_1 + (F_2 - F_1)\frac{\left(|S_w - S_w^{H1}| + \varepsilon\right)^{-1} - \frac{1}{\varepsilon}}{\left(|S_w^{H1} - S_w^{H2}| + \varepsilon\right)^{-1} - \frac{1}{\varepsilon}}. \end{aligned} \tag{6.7}$$

The capillary pressures, $P_{c,D}$ and $P_{c,I}$, are obtained from the drainage and imbibition boundary curves at the same saturation (S_w). F is obviously equal to one along the drainage boundary curve, and equal to zero along the imbibition boundary curve. To obtain smooth transitions for any number of nested scanning curves, we use two variables that must be remembered from all n_{rev} previous reversals, S_w^{Hi} and F_i (i=1,2, ...,n_{rev}). The numbering is organized so that the last reversal is 1 and the previous is 2, and so on. A value of 0.05-0.1 was indicated for the "*curvature*" parameter ε by Killough. Unpublished work at IRIS indicates that a lower value 0.01 match experimental data better. An improved model might use a saturation dependent ε.

Computed scanning curves starting at different positions along the primary drainage curves are shown in Figure 6.2. The scanning curves (imbibition + drainage) forms closed loops. The effect of different "*curvature*" parameter ε is demonstrated.

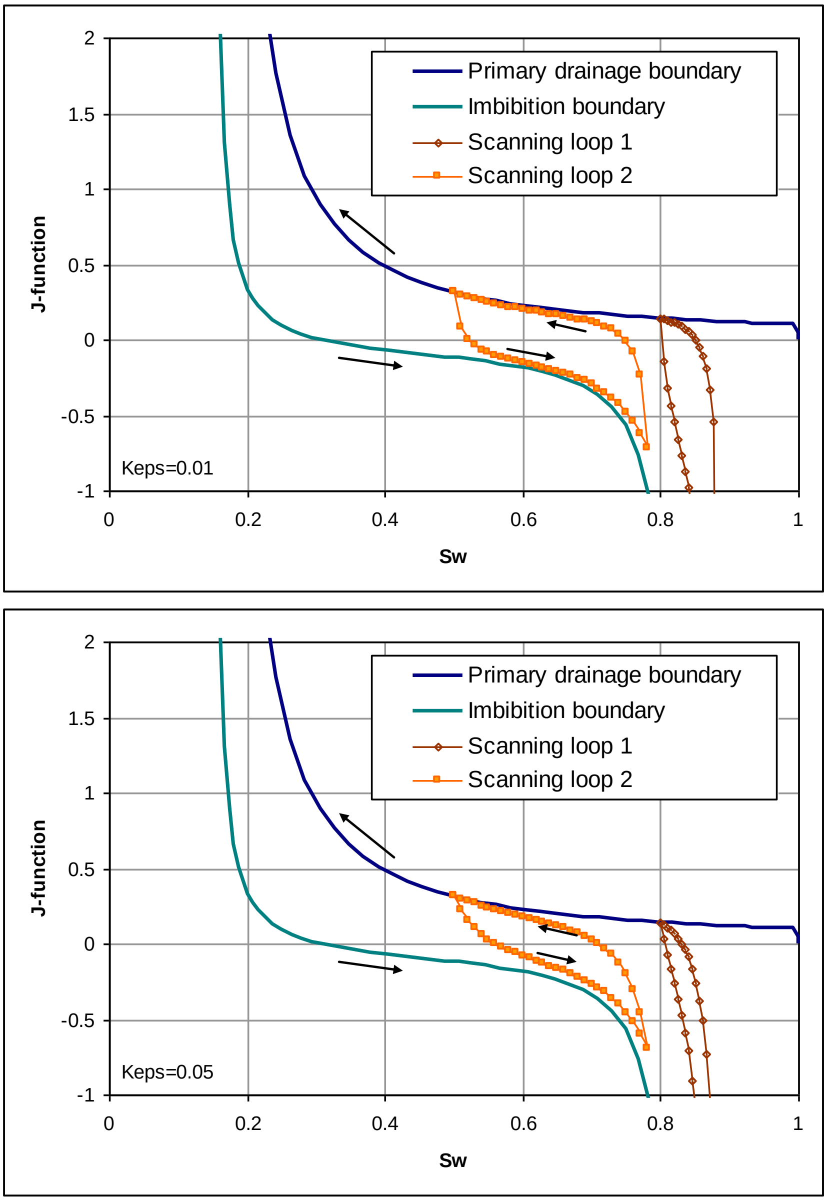


Figure 6.2 Calculation of capillary scanning curves at different saturation reversals using ε = 0.01 in upper figure and 0.05 in lower figure.

The original model assumes drainage and imbibition boundary curves have the same end point saturations, i.e., imbibition and secondary drainage. We want to address experimental procedures as well, e.g., establishing initial water saturation. That will include the use of primary drainage Pc, and a method for computation imbibition scanning curves starting from a higher water saturation than the imbibition boundary curve. The two things needed are (1) computation of residual oil saturation (from the previous saturation history) and (2) some saturation scaling for computation of boundary values (P_{cD1}, P_{cI}). The saturation for looking up the drainage $P_{cD}(S_{wD})$ is:

$$S_{wD} = S_{whmn} + (S_w - S_{whmn})\frac{S_{ohmx}-S_{orD}}{S_{ohmx}-S_{or}}, \quad (6.8)$$

where S_{whmn} is the historical minimum water saturation, S_{ohmx} the maximum historical oil saturation (obviously S_{whmn}+S_{ohmx}=1), S_{orD} is the primary drainage residual oil saturation (normally equal to zero) and S_{or} the current residual oil saturation normally obtained as a function of S_{ohmx} (see below). Obviously, if S_{or} is constant (and equal to S_{orD}) then there will be no scaling. The scaled saturation for looking up the imbibition value, $P_{cI}(S_{wI})$, must be computed differently:

$$S_{wI} = S_{wr} + (S_w - S_{wr})\frac{1-S_{orI}-S_{wr}}{1-S_{or}-S_{wr}}, \quad (6.9)$$

where S_{orI} is the residual oil saturation in from the imbibition boundary input and the residual water saturation, S_{wr}, is assumed to be the same in the drainage and imbibition boundary curves.

Derivative expression needed for the sequential solver is computed by:

$$\frac{dP_c}{d\,S_w} = \frac{dP_{\mathrm{cI}}}{d\,S_w} + F\left(\frac{dP_{\mathrm{cD}}}{d\,S_w} - \frac{dP_{\mathrm{cI}}}{d\,S_w}\right) + \frac{dF}{d\,S_w}(P_{cD} - P_{cI}). \quad (6.10)$$

F can be written

$$\begin{aligned} F &= F_1 + b(a^{-1} - \varepsilon^{-1}), \quad (6.11)\\ b &= \frac{F_2 - F_1}{(|S_w^{H1} - S_w^{H2}| + \varepsilon)^{-1} - \frac{1}{\varepsilon}},\\ a &= |S_w - S_w^{H1}| + \varepsilon. \end{aligned}$$

Then the derivative of F is

$$\begin{aligned} \frac{dF}{d\,S_w} &= -\frac{b}{a^2}\frac{da}{d\,S_w}, \quad (6.12)\\ \frac{da}{d\,S_w} &= \begin{cases} 1, & S_w^{H2} \geq S_w \geq S_w^{H1}(\text{decreasing}S_w)\\ -1, & S_w^{H2} \leq S_w \leq S_w^{H1}(\text{increasing}S_w) \end{cases} \end{aligned}$$

The dF/dS_w will always be negative.

6.3.3 Relative permeability

Killough [13] presented a simulation model utilising experimental curves and Land’s expression, Eq. (6.13) for residual non-wetting-phase saturation.

$$S_{Nr} = \frac{S_N^{Max} - S_{NrD}}{1 + C_L(S_N^{Max} - S_{NrD})} + S_{NrD} \quad (6.13)$$

The model uses experimental curves for primary drainage and imbibition. Hysteresis between primary drainage and imbibition is included for both the wetting and the non-wetting phase. Residual saturation for the non-wetting phase, S_{Nr}, is calculated by Land’s expression as a function of maximum historical saturation, S_N^{Hyst}. Scanning curves between primary drainage and imbibition (D_1 and I_1) are calculated from the imbibition curves. The model is illustrated in Figure 2.3.1 and 2.3.2, for the non-wetting and wetting phase respectively.

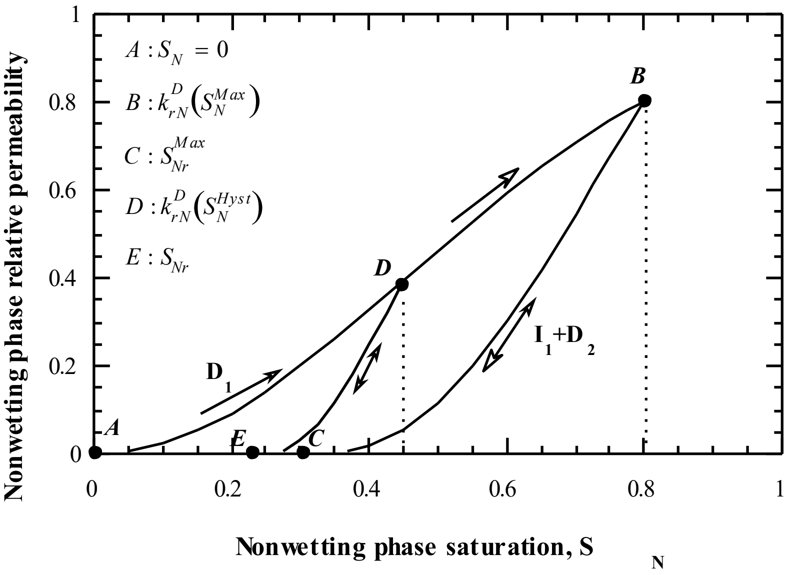


Figure 6.3 : Killough's model for non-wetting-phase relative permeability. Primary drainage, D1: ADB, imbibition, I1: BC, secondary drainage, D2: CB, scanning curve (imbibition and drainage): DED.

The scanning imbibition curve (*DE*) for the non-wetting phase are given by:

$$k_{rN}^{I}(S_N) = k_{rN}^{D}(S_N^{Hyst}) \cdot \left[\frac{k_{rN}^{Exp}(S_N^{Norm}) - k_{rN}^{Exp}(S_{Nr}^{Max})}{k_{rN}^{Exp}(S_N^{Max}) - k_{rN}^{Exp}(S_{Nr}^{Max})}\right], \tag{6.14}$$

where saturation on the scanning curve ED is scaled to the imbibition curve CB

$$S_N^{Norm} = S_{Nr}^{Max} + \left[\frac{(S_N - S_{Nr}) \cdot (S_N^{Max} - S_{Nr}^{Max})}{(S_N^{Hyst} - S_{Nr})}\right]. \tag{6.15}$$

Described by words instead; the curve ED is computed by scaling the imbibition curve BC between D and E.

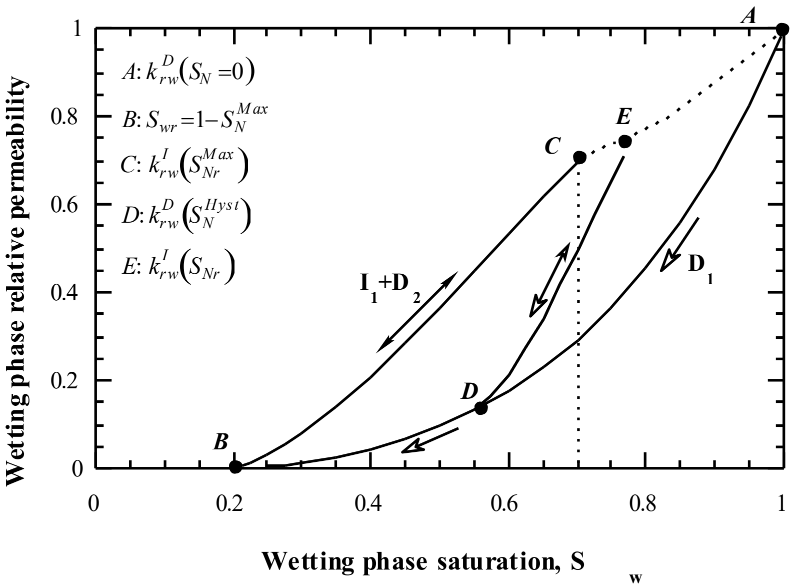


Figure 6.4: Killough's model for the wetting phase relative permeability. Primary drainage, D_1: *ADB*, imbibition, I_1: *BC*, secondary drainage, D_2: *CB*, scanning curve (imbibition and drainage): *DED*.

For the wetting phase, the end point relative permeability is interpolated between initial value at 100 % saturation ($k_{rw}^{D}(S_N = 0)$ at point A in **Figure 6.4**) and the experimental imbibition endpoint, $k_{rw}^{I}(S_{Nr}^{Max})$ at point C.

$$k_{rw}^{I}(S_{Nr}) = k_{rw}^{D}(S_{Nr}) + \left(k_{rw}^{*I}(S_{Nr}^{Max}) - k_{rw}^{D}(S_{Nr}^{Max})\right)\left(\frac{S_{Nr}}{S_{Nr}^{Max}}\right)^{\alpha} \tag{6.16}$$

where k_{rw}^{*I} represents an experimental or analytical imbibition curve in between S_N^{Max} and S_{Nr}^{Max}. The scanning curve for imbibition is then calculated by the formula:

$$k_{rw}^{I}(S_N) = k_{rw}^{D}(S_N^{Hyst}) + \left[\frac{k_{rw}^{*I}(S_N^{Norm}) - k_{rw}^{*I}(S_N^{Max})}{k_{rw}^{*I}(S_{Nr}^{Max}) - k_{rw}^{*I}(S_N^{Max})}\right] \cdot \left[k_{rw}^{I}(S_{Nr}) - k_{rw}^{D}(S_N^{Hyst})\right]. \tag{6.17}$$

So, the relative permeability is computed by scaling the imbibition curve BC between the reversal point D (historical maximum non-wetting phase saturation) and point E.

The imbibition curves for both phases are considered reversible in the model, i.e. upon phase reversal of an imbibition process, the following drainage process is assumed to follow the imbibition curve until maximum historical non-wetting saturation, S_N^{Hyst}, is reached.

6.3.3.1.1 Alternative interpolation

In short, the Killough's scanning curve is the total "imbibition curve BC" scaled between DE. In particular for water, this scaling of the total imbibition curves may result in non-intuitive behaviour as e.g. the first part of an imbibition scanning curve crossing the drainage curve. If we instead base the scaling on the mobile saturations:

$$k_{rN}^{I}(S_N) = k_{rN}^{D}\left(S_N^{Hyst}\right) \cdot \frac{k_{rN}^{Exp}(S_N^{m})}{k_{rN}^{Exp}(S_N^{Hm})} \tag{6.18}$$

where the scaled actual saturation and the reversal point saturation used to look up the imbibition (scanning) curve relperms are:

$$\begin{aligned} S_N^{m} &= S_{Nr}^{Max} + (S_N - S_{Nr})\frac{(S_N^{Max} - S_{Nr}^{Max})}{(S_N^{Max} - S_{Nr})} \\ S_N^{Hm} &= S_{Nr}^{Max} + \left(S_N^{Hyst} - S_{Nr}\right)\frac{(S_N^{Max} - S_{Nr}^{Max})}{(S_N^{Max} - S_{Nr})} \end{aligned} \tag{6.19}$$

For water:

$$\begin{aligned} k_{rw}^{I}(S_N) = k_{rw}^{D}\left(S_N^{Hyst}\right) &+ \left[\frac{k_{rw}^{*I}(S_N^{m}) - k_{rw}^{*I}(S_N^{Hm})}{k_{rw}^{*I}(S_{Nr}^{Max}) - k_{rw}^{*I}(S_N^{Hm})}\right] \\ &\cdot \left[k_{rw}^{I}(S_{Nr}) - k_{rw}^{D}\left(S_N^{Hyst}\right)\right], \\ k_{rw}^{I}(S_{Nr}) = k_{rw}^{D}(S_{Nr}) &+ \left(k_{rw}^{*I}(S_{Nr}^{Max}) - k_{rw}^{D}(S_{Nr}^{Max})\right) \cdot \frac{S_{Nr}}{S_{Nr}^{Max}} \end{aligned} \tag{6.20}$$

The two interpolation methods are compared in Figure 6.5

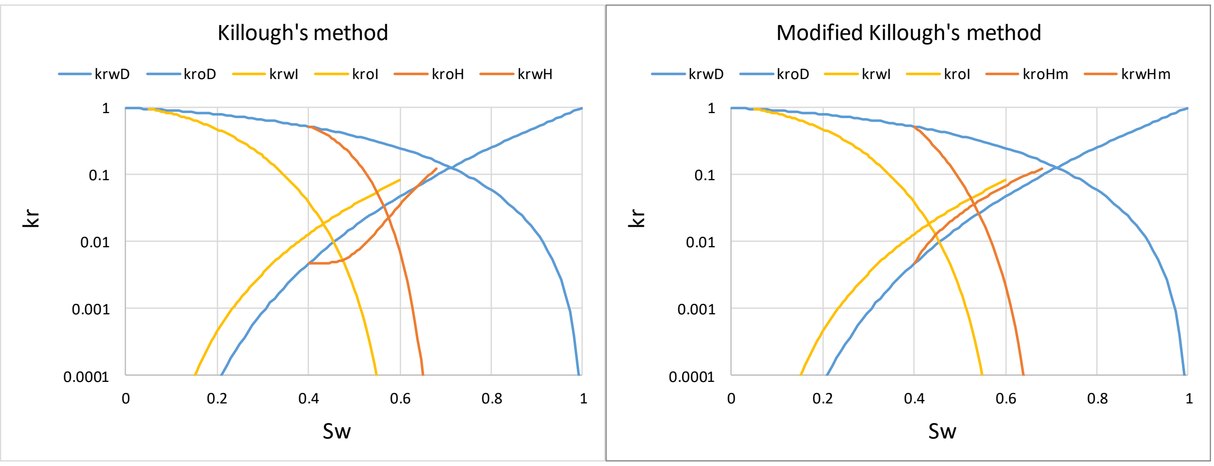


Figure 6.5 Relative permeability scanning curves (kroH, krwH) computed from primary drainage and imbibition boundary curves. Left, original Killough's method and right, modified Killough's method.

6.4 Heat model

The temperature model in *Bugsim* includes heat transport by conduction and convection. Some assumptions/limitations of the model are:

- It is assumed no transition of heat energy to other energy forms (e.g., kinetic energy and heat produced or used in chemical reactions are neglected).

- The heat conductivity tensor of the formation is assumed diagonal and with equal conductivity in the two horizontal directions (x and y). Anisotropy between horizontal and vertical directions is allowed.
- Only water and oil components and the solid matrix are considered in the temperature calculations.
- Effects of thermal expansion are ignored.
- Thermal effects are included in fluid viscosity and in bacteria growth kinetics.

With the above-mentioned limitations, the change of heat energy (E_h) versus time can be written as a sum of heat energy transported with the flowing phases and conductive heat transport due to gradients in temperature:

$$\frac{dE_h}{dt} = \nabla \cdot \left(\sum_{l=1}^{n_p} \rho_l u_l h_{c,l} T - k_h \nabla T\right) + q_{Eh}, \qquad (6.21)$$

where q_{Eh} is a sink/source term due to production or injection through wells. The specific heat capacity of phase l ($h_{c,l}$) and the effective heat conductivity (k_h) are given by:

$$\begin{aligned} \rho_l h_{c,l} &= \sum_{k=1}^{n_c} \left(\rho_k c_{l,k} h_{c,k}\right) \\ k_h &= \phi \sum_{1}^{n_c} \left(c_{t,k} k_{h,k}\right) + (1-\phi)\rho_r k_{h,r} \end{aligned} \qquad (6.22)$$

,

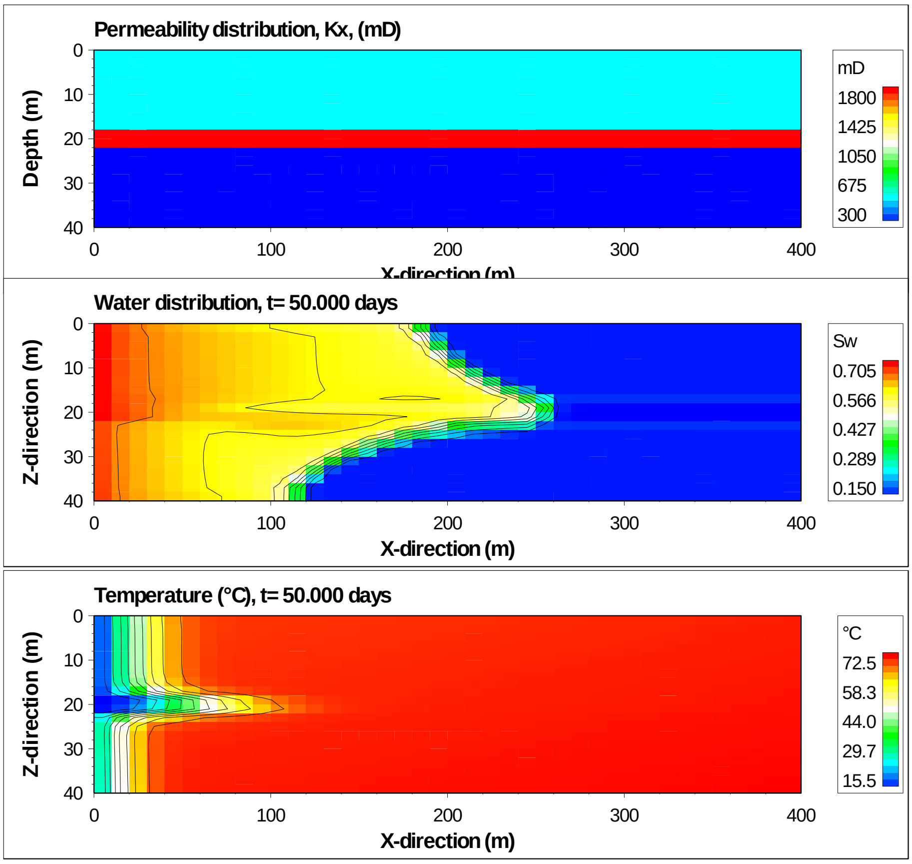
The temperature is obtained from the relation:

$$E_h = T\left[\phi \sum_{1}^{n_c} \left(\rho_k c_{t,k} h_{c,k}\right) + (1-\phi)\rho_r h_{c,r}\right]. \qquad (6.23)$$

The convective and conductive parts of Eq. (6.21) are solved separately. The solution method for the heat convection term follows the solution method selected for the phase flow (keyword ctransp), either explicit or implicit. The conductive part is solved by an either explicit or implicit method automatically selected by the program. An explicit treatment tends to be unstable in very small grid cells, which is the typical case in core scale simulations. The default order of calculations is that the conductive heat transport is solved first, and then convective heat transport is added. This can be altered with keyword tolhc.

The temperature calculations in *Bugsim* are triggered by the presence of the *heatsect* section. The essential input consists of heat capacity and conductivity values of fluids and the formation rock. Heat exchange with the surroundings is by default turned off. To activate heat exchange with the surroundings, use positive value for the boundary conductivity factor *bfcond* on the appropriate surfaces. Define external boundary layers (*extbb*) to get more accurate calculations. In e.g. a 2D cross-sectional model, it would be natural to activate external heat exchange in the vertical direction while keeping it off in the y-direction.

The down-hole injection temperature must be set for injection wells with keyword *winject*.



Permeability distribution, Kx, (mD)

Water distribution, t= 50.000 days

Temperature (°C), t= 50.000 days

Figure 6.6 Example of simulation with temperature in a 2 dimensional cross section model with permeability distribution given in the top figure. Distributions of water and temperature after 50 days of cold water injection (15°C) are shown in the middle and bottom figures, respectively.

The temperature model is demonstrated in a 2D cross sectional model in Figure 6.6. This example shows the cooling resulting from cold water (15 °C) injection into a reservoir with initial temperature of approximately 70 °C. One may observe that the temperature front advances much slower than the saturation front. The velocity of the temperature front depends on porosity and heat capacity of fluids and rock, and on the saturation behind the displacement front.

Effects of including temperature calculations are indicated in Figure 6.7. Temperature has significant influence on phase viscosities and a potential effect on microbial growth.

In the above examples, heat exchange with the surroundings was turned off. The major mechanism for heat flow in a reservoir being flooded is heat transported with the flowing phases (convective heat transport). Conductive heat transport because of temperature gradients has a much shorter range and has less influence on larger scales. The effect of including heat exchange across upper and lower reservoir boundary is demonstrated in Figure 6.8. Note that a major part of the flow in this model is along the high permeable zone in the center away from the boundaries, which results in a slower cooling of the upper and lower part when heat exchange with the surroundings is included.

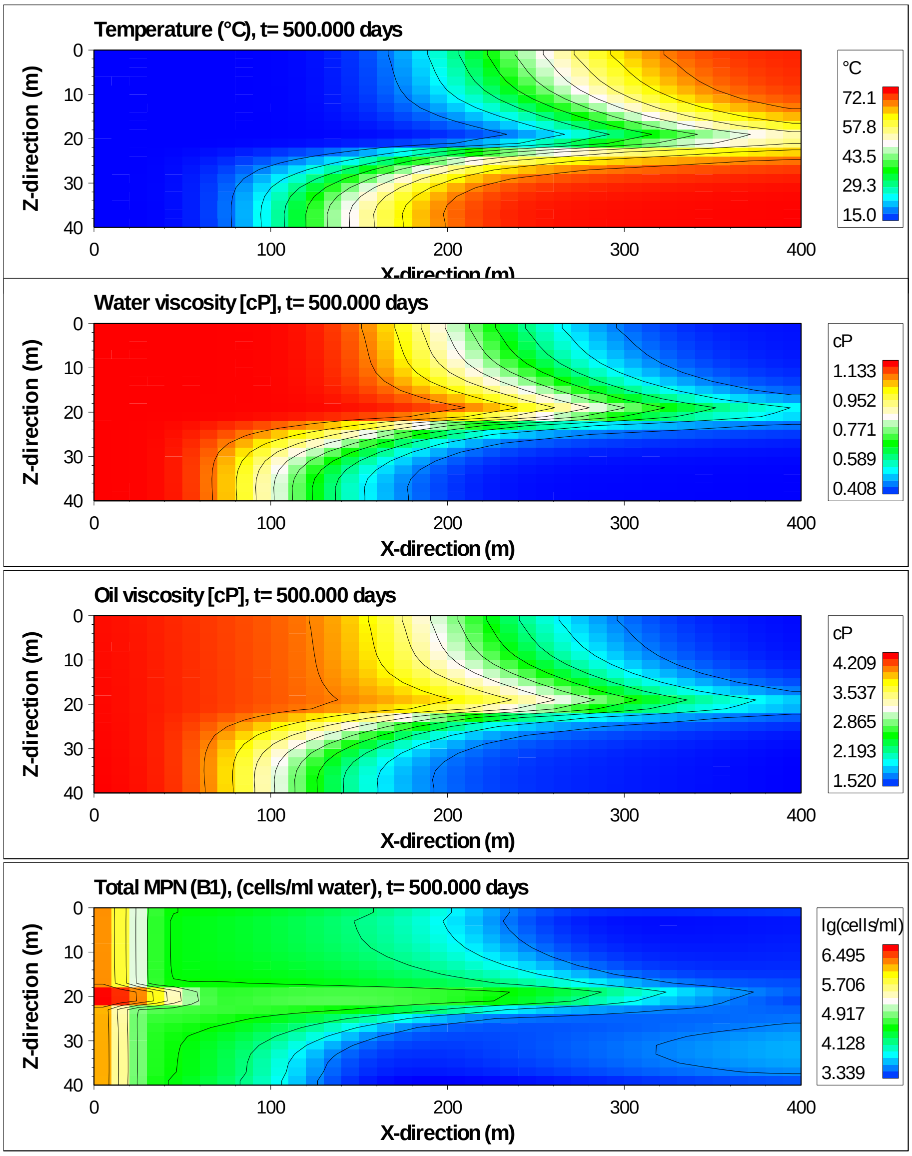


Figure 6.7 Effect of cold water injection (2D model from Figure 6.6). The temperature distribution shown at the top (after 500 days injection) influences the viscosity to water and oil, and the bacteria growth.

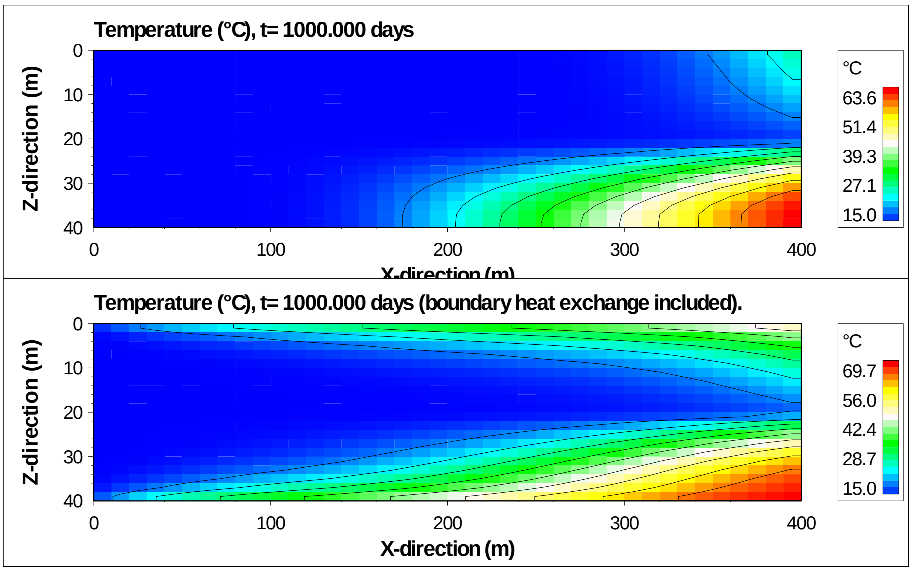


Figure 6.8 Effect of including heat exchange with the surroundings. Upper figure: zero heat flux across reservoir boundaries. Bottom figure: conductive heat exchange across upper and lower boundaries included in simulation. (2D model from Figure 6.6).

6.5 Biomass growth model

The change in biomass concentration X versus time is:

$$\frac{\mathrm{d}X}{\mathrm{d}t} = r_g - r_d \quad , \tag{6.24}$$

where r_g is the rate at which new cells are formed and r_d is the cell death rate. The X represents the current total concentration of living cells in units [gram dry biomass/ml water]. The growth term r_g is assumed to be limited by a maximum biomass concentration X_m and by limitations in substrate and various nutrients supplies.

Substrate is assumed to contain the carbon source, which constitutes the main part of the new cells (roughly 50%). The substrate will normally contain organic carbon, but it may also be CO_2.

Nutrients are inorganic compounds needed by the bacteria to generate energy and create chemical compounds (biosynthesis). Some nutrients are only needed in small quantities and others like nitrogen (N) and phosphorus (P) in relative significant amounts (the dry biomass contains roughly 14 % N and 3% P). The growth rate can then be written:

$$r_g = X \cdot \left(1 - \left(\frac{X_t}{X_m} \right)^m \right) \cdot F_T \cdot F_S \cdot F_{N1} \cdot F_{N2} \cdot \ldots. \quad , \tag{6.25}$$

where F_T represents the effect of temperature, F_S represents the specific growth rate at given substrate conditions and F_{N1}, F_{N2} etc. represents limitations in growth rate at given concentrations of various nutrients. The nutrient factors will take values between 0 (no available nutrients) and 1 (excess nutrient).

If more than one microorganism type is modelled, X_t in the above expression will represent the sum of all the types, while X is the concentration of one type. The rates will be computed for all types one at the time.

The effect of temperature is included by multiplying the bacterial growth rate with a factor F_T:

$$F_T = \begin{cases} \theta_L^{T_{Lopt}-T}, & T < T_{Lopt} \\ 1, & T_{Lopt} \le T \le T_{Uopt} \\ \theta_U^{T-T_{Uopt}} & T > T_{Uopt} \end{cases} . \tag{6.26}$$

An example of F_T is plotted in Figure 6.9.

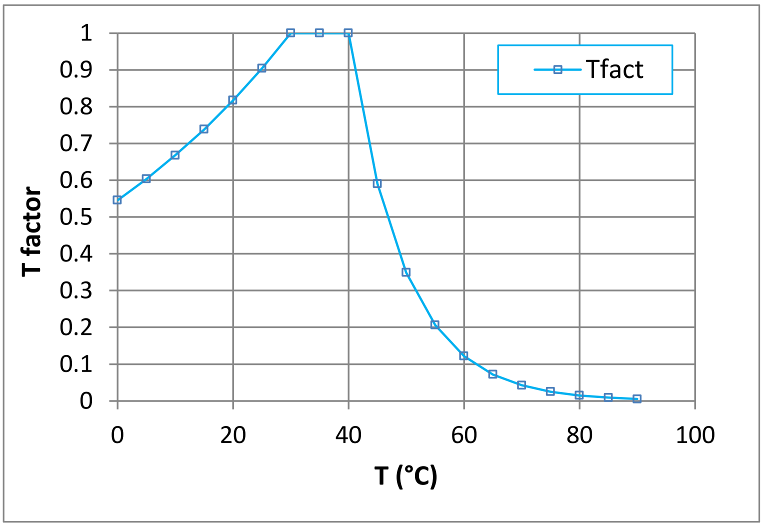


Figure 6.9 Effect of temperature on bacteria growth kinetics, (θ_L=0.98, T_L=30 °C, θ_U=0.9, T_U=40 °C).

We use the common Monod expression to describe the growth factors (F). The substrate factor if only one substrate source is considered reads:

$$F_S = \mu_{\max}\left(\frac{S}{K_S + S}\right), \tag{6.27}$$

where S represents the substrate concentration and K_S is the half rate constant (representing the concentration at which the rate is $\mu_{\max}/2$). If more than one substrate source is present, competitive growth is modelled by:

$$F_S = \frac{\sum_{i=1}^{n_S} \frac{S_i}{K_{Si}} \mu_{\max,i}}{1 + \sum_{i=1}^{n_S} \frac{S_i}{K_{Si}}} . \tag{6.28}$$

The expression (6.28) is different from what is normally used as we allow to specify different $\mu_{\max}$-values associated with different substrate sources. The usual approach is to use the same $\mu_{\max}$ for all substrate sources, and then $\mu_{\max}$ can be moved outside the summation.

A similar expression is used for the nutrient factors, here shown for the j^{th} factor, F_{Nj}.

$$F_{N_j} = \frac{\sum_{i=1}^{n_{Nj}} \frac{N_{ji}}{K_{N,ji}}}{1+\sum_{i=1}^{n_{Nj}} \frac{N_{ji}}{K_{N,ji}}} . \tag{6.29}$$

In the above expression, N_{ji} represents the concentration of nutrient component i belonging to nutrient group j. K_{Nji} denote the corresponding half rate constant.

The consumption rate of a substrate S_i is given by:

$$\frac{dS_i}{dt} = -f_{S,i} \cdot \frac{r_g}{Y_i} \tag{6.30}$$

$$f_{S,i} = \frac{\frac{S_i}{K_{Si}} \mu_{\max,i}}{\sum_{i=1}^{n_S} \frac{S_i}{K_{Si}} \mu_{\max,i}} .$$

And the production of a component C_j from the metabolism of S_i:

$$\frac{dC_j}{dt} = U_{p,j}\, f_{S,i} \cdot \frac{r_g}{Y_i} . \tag{6.31}$$

The change of nutrient N_{ji} is expressed by:

$$\frac{dN_{ji}}{dt} = -f_{N,ji} \cdot r_g U_{N,ji} \tag{6.32}$$

$$f_{N,ji} = \frac{\frac{N_i}{K_{N,i}}}{\sum_{i=1}^{n_{Nj}} \frac{N_{ji}}{K_{N,ji}}} .$$

The death rate of the bacteria is currently modelled only with a constant specific decay constant, k_d.

$$r_d = X \cdot k_d . \tag{6.33}$$

Note that while in the above expressions (6.30), (6.31) and (6.32), the change in a component's concentration is associated with only a single path. In the general way the growth model is implemented in *Bugsim*, the user may specify the same component product from more than one substrate source, and the produced component may again serve as a substrate and/or nutrient. Also, the same component may be specified as both a substrate and nutrient source.

When solving the growth equations for a time step Δt, the above differential equations are discretised ($\mathrm{d}X/\mathrm{d}t$ -> $\Delta X/\Delta t$) and solved in an implicit manner with respect to time. Let n represent the concentrations from the previous timestep and n+1 the new timestep. Then all the concentrations at the right-hand side of the equations are from time n+1. These concentrations are found using an iterative procedure starting with concentration from time n.

6.6 Surfactant and biofilm models

Possible effects of surfactant (biosurfactant) and biofilm on oil recovery are basically modelled by interpolations between sets of relative permeability curves. Because of limited knowledge about mechanisms involved in the proposed effects of biofilm, this model is rather simple with the relative permeability interpolation parameter modelled as a function of the bacteria adsorption, $F_m(A_b)$. In the surfactant model, the interpolation parameter is a function of the capillary number, $F_m(N_c)$. The surfactant model is described in more detail below. The interpolation options described for surfactant is also valid for the biofilm model (share the same relative permeability interpolation code).

6.6.1 Surfactant model

One or more component(s) must be defined as type SURFACTANT. The surfactant component can be injected, and/or be introduced in-situ as a microbial product. The present treatment of surfactant behaviour is:

- Surfactant is only present in the water phase.
- Interfacial tension is only a function of the aqueous surfactant concentration, IFT=IFT($c_{w,ks}$).
- Adsorption is only a function of aqueous surfactant concentration, $A_s = A_s(c_{w,ks})$.
- The effect of surfactant on oil production is modelled by
    i. compute *IFT*
    ii. compute the capillary number, N_c
    iii. use N_c to interpolate between relative permeability sets

The capillary number definition used in *Bugsim* is computed using the pressure gradient of the oil phase:

$$\vec{N}_c = \frac{\bar{\bar{k}}(\nabla p_o - \rho_o \vec{g})}{IFT}, \qquad (6.34)$$

where the permeability (k) is assumed to be a diagonal tensor. The same N_c is used also for the oil phase. The length of the capillary number vector,

$$N_c = \sqrt{N_{c,x}^2 + N_{c,y}^2 + N_{c,z}^2}, \qquad (6.35)$$

is used to determine the interpolation parameter $F_m(N_c)$. The functional relation between F_m and N_c must be specified by the user, either as a table (which is the method used in *Eclipse*, see keyword ncmisc) or F_m can be linearly interpolated between N_c values associated with each relative permeability set. With two data sets (A and B), linear interpolation of F_m is given by:

$$F_m = \begin{cases} 0, & N_c \le N_{cA} \\ \dfrac{\log N_c - \log N_{cA}}{\log N_{cB} - \log N_{cA}}, & N_{cA} < N_c < N_{cB} \\ 1, & N_c > N_{cB} \end{cases} \qquad (6.36)$$

An arbitrary number of relative permeability sets can be used, however, tabular $F_m(N_c)$ is only allowed with two sets. With more data sets, linear interpolation between sets is used, Eq.(6.36).

When F_m is determined, then the interpolation between curve sets A and B can be done. If the curves A and B have different end points (S_{wr} and S_{or}), then the new end points are obtained by:

$$\begin{aligned} S_{or} &= (1 - F_m)S_{orA} + F_m S_{orB} \\ S_{wr} &= (1 - F_m)S_{wrA} + F_m S_{wrB}. \end{aligned} \qquad (6.37)$$

The curve set A and B are then saturation scaled to the new endpoints. Then the interpolation between the scaled curve sets A and B can be done in three ways (see keyword intkr). The linear interpolation (used in e.g. Eclipse) is given by:

$$k_{rj} = (1 - F_m)k_{rjA} + F_m k_{rjB}, j = w, o. \qquad (6.38)$$

An alternative option is to instead interpolate the logarithmic k_r values:

$$ln(k_{rj}) = (1 - F_m)\, ln(k_{rjA}) + F_m\, ln(k_{rjB})\,, j = w, o. \qquad (6.39)$$

The third method is an option in the STARS simulator:

$$k_{rj} = k_{rjA} + F_m^{ej}(k_{rjB} - k_{rjA}), j = w, o, \qquad (6.40)$$

where the modifying exponent e_j allows faster (e_j<1) or slower (e_j>1) transition from A to B. Different exponents can be used for oil and water phases.

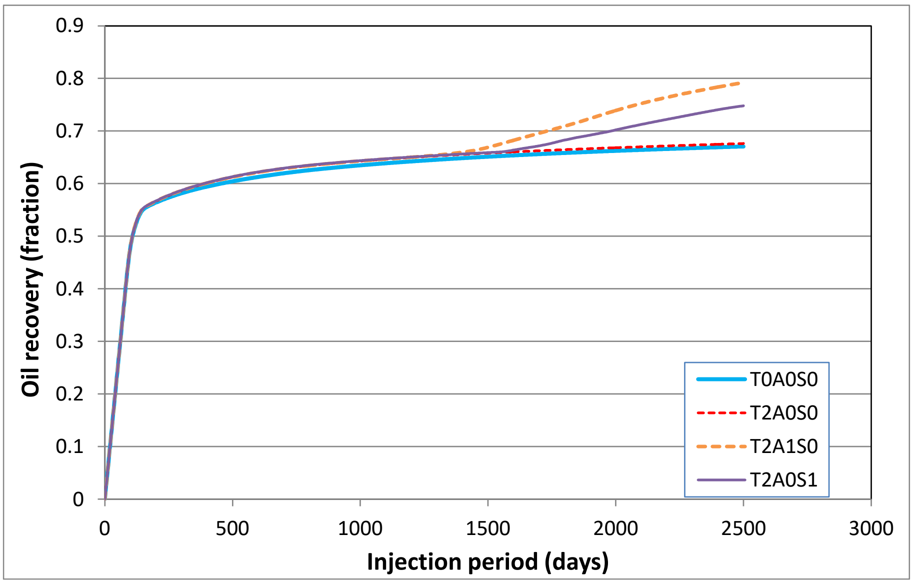


Figure 6.10 Example simulations using temperature and relative interpolation models (2D model from Figure 6.6). T0A0S0: constant temperature , T2A0S0: cold water injection, T2A1S0: adsorbed biomass used to interpolate relative permeability and T2A0S1: capillary number (increased by produced biosurfactant) is used for relative permeability interpolation.

The use of the linear interpolation model to simulate effects of biosurfactant and biofilm (i.e. adsorbed bacteria) is demonstrated in Figure 6.10. In this simulation, food (substrate+nutrient) is injected the first 500 days and from 1000 to 1250 days. In the first period there was insufficient growth due to the high initial reservoir temperature. After sufficient cooling from cold water injection, growth was much better and an effect of biofilm or produced biosurfactant is seen (with the supplied input parameters). Figure 6.11 shows distribution plots of IFT, N_c and F_m from the simulation with produced biosurfactant.

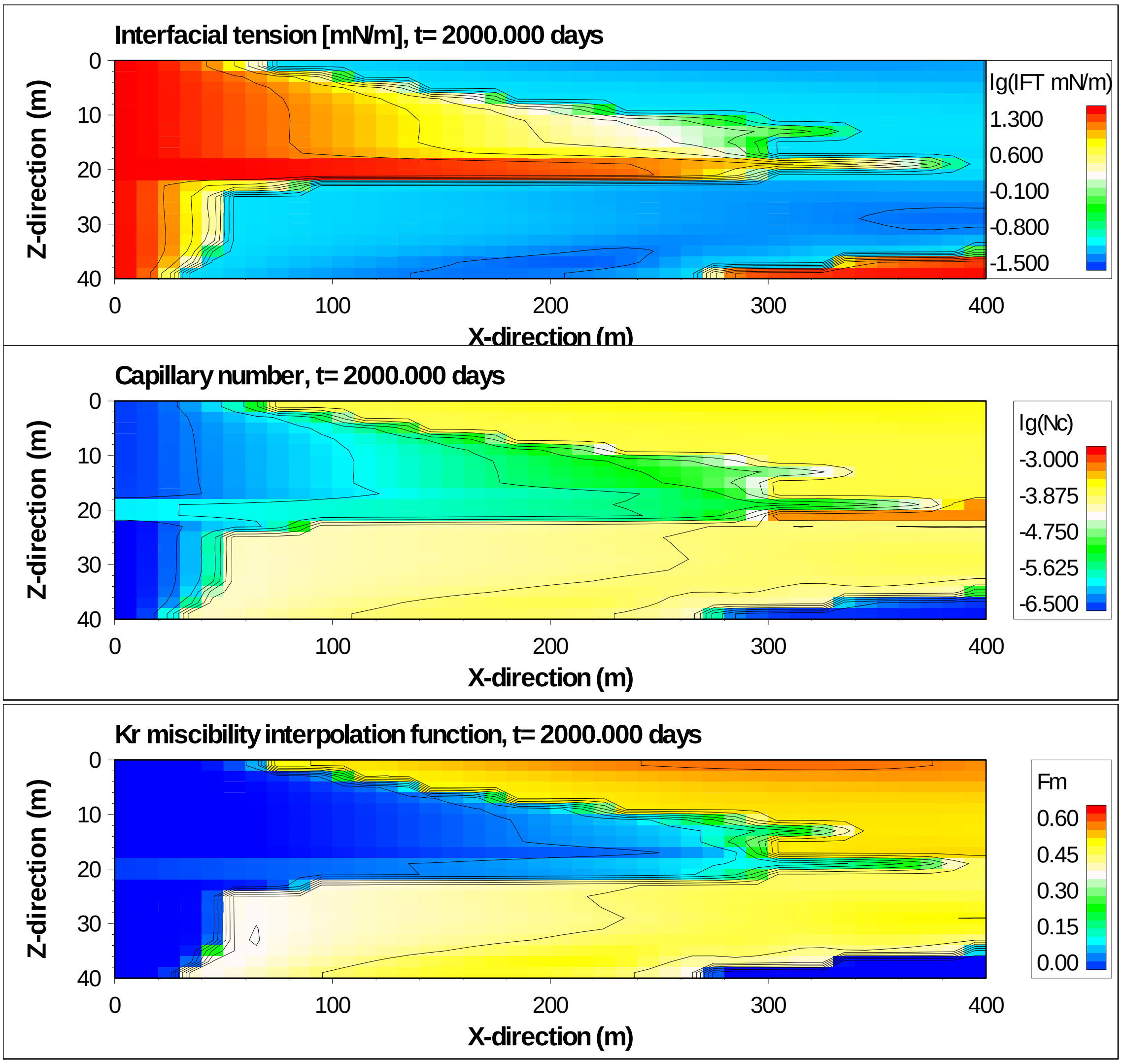
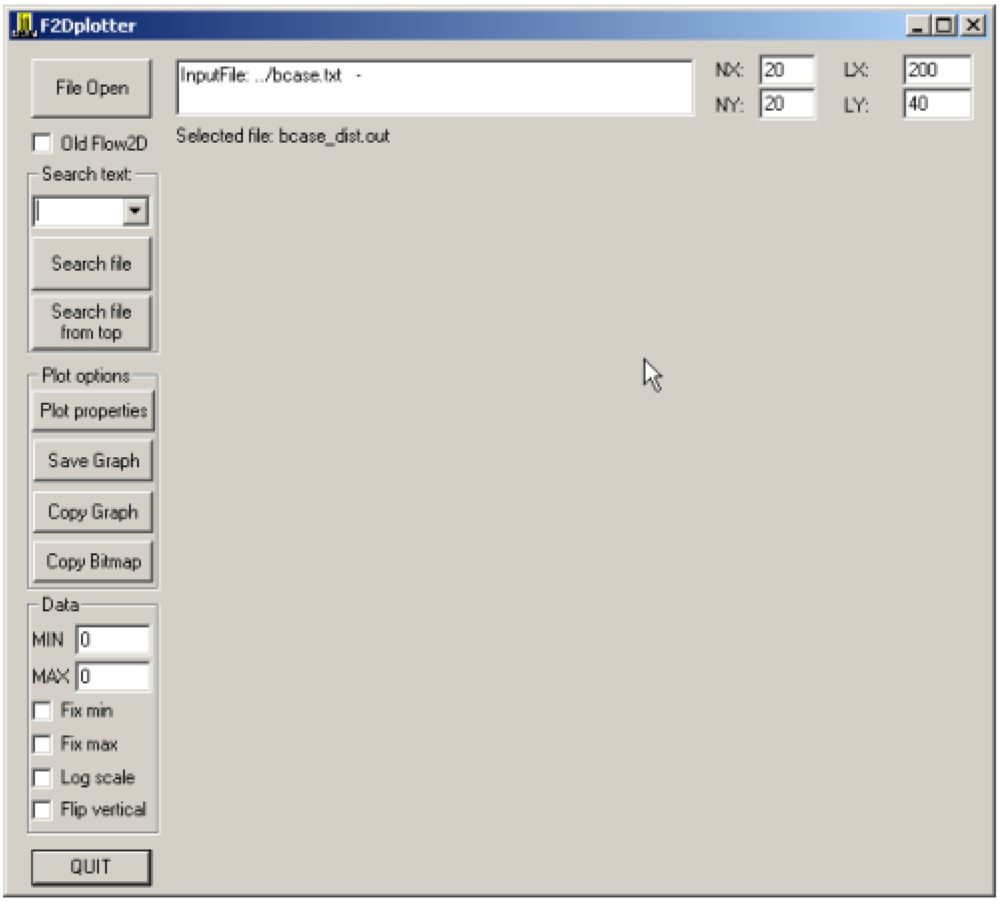


Figure 6.11 Example simulations using the surfactant model (2D model from Figure 6.6). The modeled production of biosurfactant results in reduced interfacial tension (upper figure) and increased capillary number (middle figure). The corresponding relative permeability interpolation parameter, $F_m(N_c)$, is given in the bottom figure.

7 Distribution plot utility



InputFile: ../bcase.txt -

Selected file: bcase_dist.out

NX: 20
NY: 20
LX: 200
LY: 40

☐ Old Flow2D

MIN 0
MAX 0

☐ Fix min
☐ Fix max
☐ Log scale
☐ Flip vertical

Figure 7.1 Initial F2DPlot dialog box.

The F2Dplotter.exe is a utility program made within the upscaling activity at IRIS for fast 2D distribution plotting of properties. It can be used to plot property distributions printed from *Bugsim* for vertical cross-sections (if *ny* = 1) or for one layer at the time (3D models). Properties that can be plotted are printed distribution data in file_dist.out, at specified times. Generated plots may be saved to files or transferred to other programs (e.g., Word) by copying and pasting. Some examples are given in Figure 7.3 and Figure 7.4.

Starting the F2Dplotter.exe program will bring up the 2D plot dialog box seen in Figure 7.1. In the following description of this program, key-text in dialog boxes is referred to using matching underlined text.

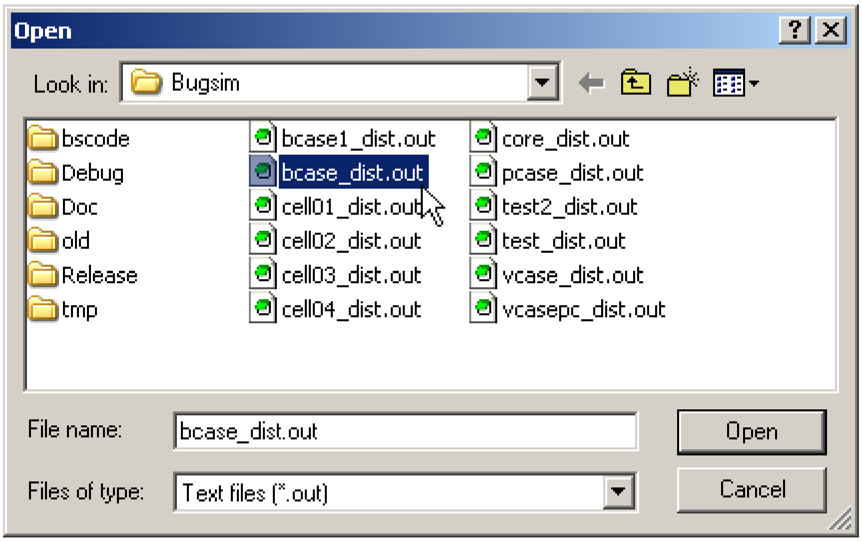
The user must first select the data to plot by pressing the File Open – button. Files generated by *Bugsim* that can be plotted are named file_dist.out (where ‘file’ denotes the name of the input file. If a valid file has been opened, grid dimensions in terms of block numbers (NX and NY) and lengths (LX and LY) will be seen in the upper right corner of the dialog box.

Note: the dialog box can be resized by dragging the lower right corner.

- File Open – click to open a new file

Possible files are

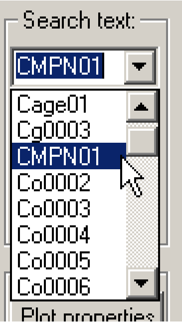
*_dist.out:



File name: bcase_dist.out

Files of type: Text files (*.out)

- Search text – select a keyword from dropdown list. A list of possible keywords and their meaning are given in Table 7.1.

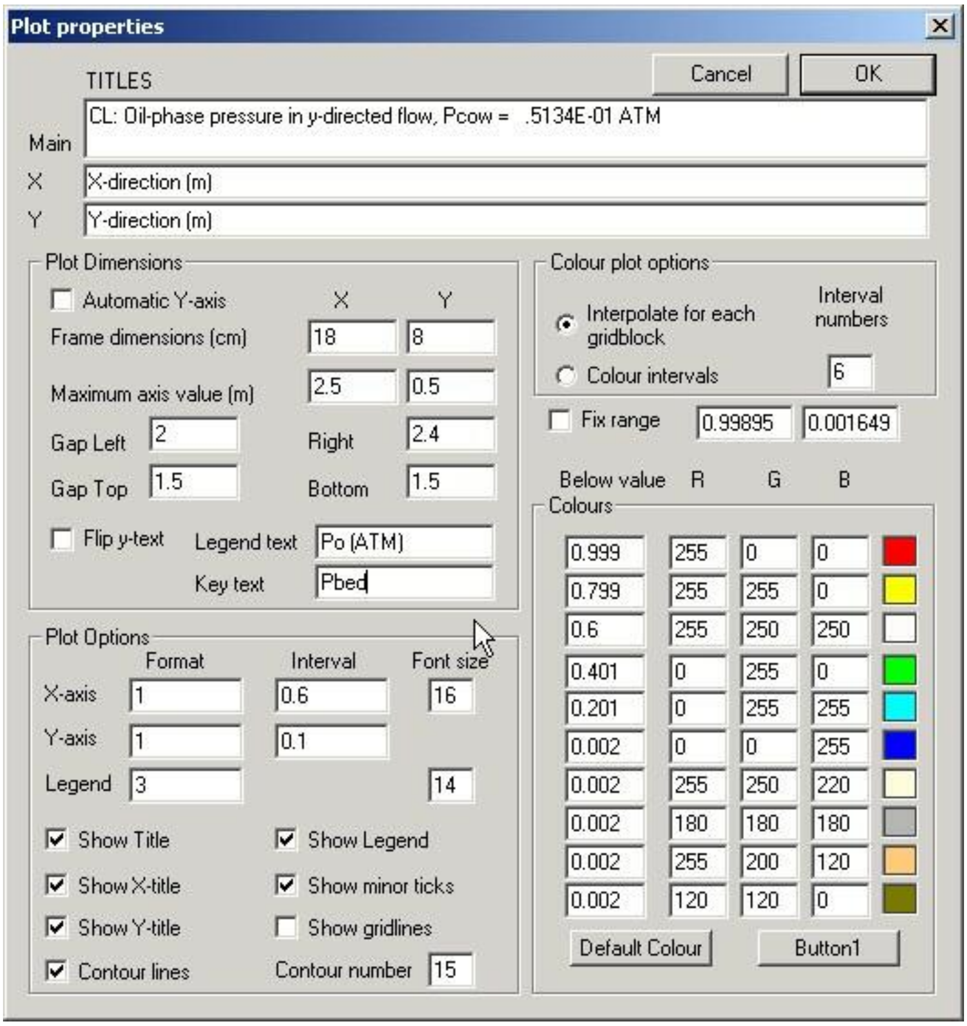


Search file – press button to start plotting. The file is searched for the selected keyword (Search text) and related data is plotted. Pressing the button again will jump to the next occurrence of the keyword. When the end of the file is reached, the message “no text found” will be displayed in the window at the top.

- Search file from top – press to restart the keyword search from the beginning of the file. No stepping back is possible, so to re-plot data one has to start all over from the beginning. Should normally be used after changing the Search text keyword.
- Plot properties – brings up a dialog box (Figure 7.2) with plot options that allows you to change the appearance of the plot.
- Save Graph – saves current plot to file in enhanced-metafile format (*.emf).
- Copy Graph – places the current plot on the clipboard in enhanced-metafile format.
- Copy Bitmap – places the current plot on the clipboard in bitmap format.
- Data – the minimum and/or maximum values for plotting can be fixed. The plotting range can be changed from linear to logarithmic. The direction of the *y*-axis can be changed (Flip vertical). The default plotting direction is positive *y*- or *z*-axis upwards, while in simulations of vertical cross-sections, the *y*- or *z*-axis is commonly chosen to be positive downwards. In that case, the Flip vertical option should be used (see Figure 7.3).

Table 7.1 Possible keywords for distribution plots.

Sor-*file*.out	
`G___Sx`	Saturation, x = G, O, W (gas, oil or water)
`G___Px`	Phase pressure, x = G, O, W (gas, oil or water)
`GVISCx`	Phase viscosity, x = G, O, W (gas, oil or water)
`Cx000i`	Phase concentration of component *i*, x = g, o, w (gas, oil or water)
`CT000i`	Total concentration of component *i*
`RS000i`	Gas/oil ratio of dissolved gas component *i* [Sm^3/Rm^3]
`CMPN0i`	Cells/ml in water phase distribution for bacteria component *i* (only un-attached biomass)
`TMPN0i`	Total cells/ml distribution for bacteria component *i* (including attached biomass)
`Cage0i`	Average cell age distribution for bacteria component *i*
`ADS00i`	Adsorption of component *i*
`G__RKF`	Permeability reduction factor
`G__KAx`	Absolute permeability distribution in direction x, x = X, Y, Z
`G_PORO`	Porosity distribution



TITLES

Main: CL: Oil-phase pressure in y-directed flow, Pcow = .5134E-01 ATM

X: X-direction (m)

Y: Y-direction (m)

Plot Dimensions

☐ Automatic Y-axis

	X	Y
Frame dimensions (cm)	18	8
Maximum axis value (m)	2.5	0.5

Gap Left 2 Right 2.4

Gap Top 1.5 Bottom 1.5

☐ Flip y-text Legend text Po (ATM)

Key text Pbed

Plot Options

	Format	Interval	Font size
X-axis	1	0.6	16
Y-axis	1	0.1	
Legend	3		14

☑ Show Title ☑ Show Legend

☑ Show X-title ☑ Show minor ticks

☑ Show Y-title ☐ Show gridlines

☑ Contour lines Contour number 15

Colour plot options

● Interpolate for each gridblock

○ Colour intervals

Interval numbers 6

☐ Fix range 0.99895 0.001649

Colours

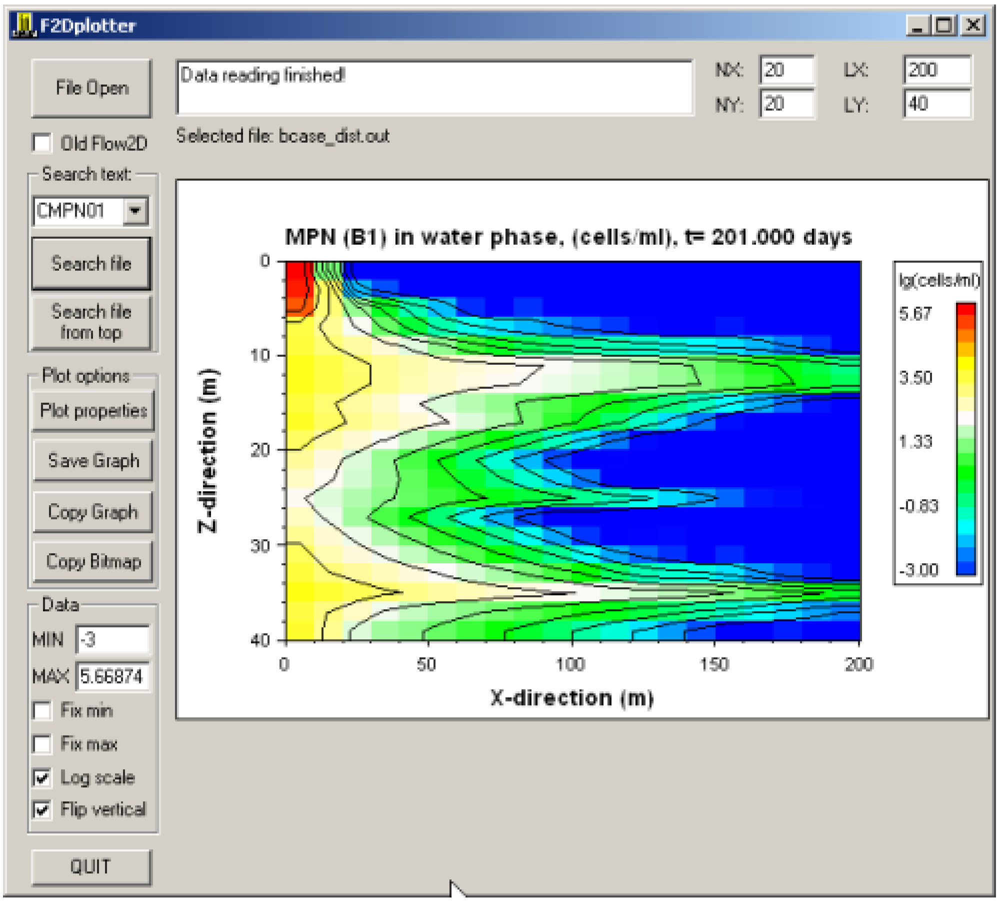
Below value	R	G	B
0.999	255	0	0
0.799	255	255	0
0.6	255	250	250
0.401	0	255	0
0.201	0	255	255
0.002	0	0	255
0.002	255	250	220
0.002	180	180	180
0.002	255	200	120
0.002	120	120	0

Figure 7.2 Plot properties dialog box.

- <u>TITLES</u> – three text windows show the current <u>Main</u>, <u>X</u>, and <u>Y</u> titles, which can be altered here. Note: \n in the <u>Main</u> text is interpreted as a line shift. The <u>Main</u> title is reread from file every time the <u>Search file</u> - button is pressed, while the <u>X</u> - and <u>Y</u> - titles are only read each time a new file is opened.
- Plot dimensions –
  - <u>Automatic Y-axis</u> – check to scale Y-axis using actual grid size (plotted ratio limited to be at least 1:5 and not bigger than 5:1).
  - <u>Frame dimensions</u> – enter the plot dimensions in cm. (The displayed plot will always be reduced to fit the available plotting area if to small. The plotting area can be increased by dragging.)
  - <u>Maximum axis value (m)</u> – reduce the maximum values to plot only the first part of the grid.
  - <u>Gap Left</u>, <u>Right</u>, <u>Top</u> and <u>Bottom</u> – distances in cm between plot and outer frame.

  - Flip y-text – check box to flip *y*-text. On different windows system, it might happen that the *y*-text is rotated when pasted into another program. Check this box if that happens.

- Legend text – shows current legend text and allows altering.
- Key text - text entered here is shown in lower right corner of the plot.
- Plot Options –
  - Format – the number of digits for the value labels for the *x*- and *y*-axes.
  - Interval – the interval between value labels
  - Font size – enter font sizes for axes and legend text
  - Show ...... – check boxes for properties to be shown, uncheck to hide property.
  - Contour lines – check to use contour lines in the plot.
  - Contour number – enter number of contours to be used.
- Colour plot options – select option, to Interpolate the colour for each grid block or to use a fixed number of Colour intervals. Enter the number of colours (Interval numbers) to be used, N_{int}.
- Colours – the current colours and their RGB (Red, Green, Blue) values are listed. The first N_{int} colours are used. The individual colours can be changed by changing the corresponding R, G and B values (in the range 0-255).



Data reading finished!

NX: 20

NY: 20

LX: 200

LY: 40

Selected file: bcase_dist.out

☐ Old Flow2D

Search text: CMPN01

Data

MIN -3

MAX 5.66874

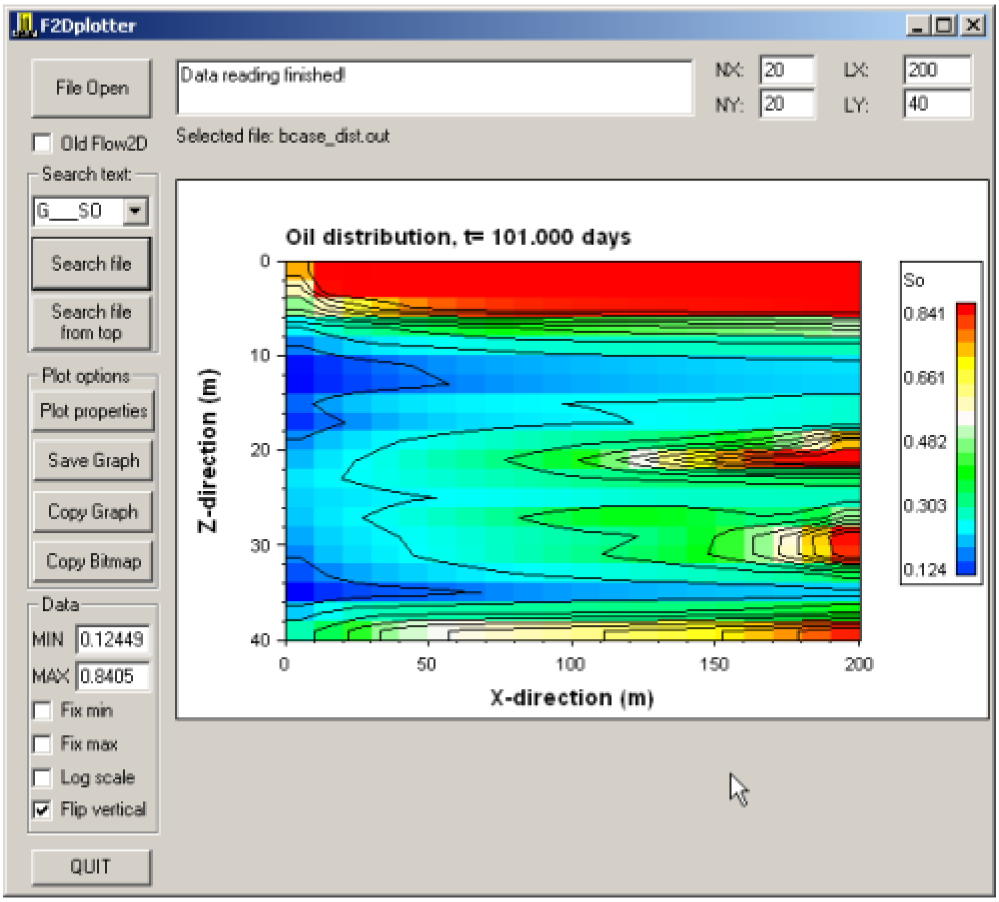
☐ Fix min

☐ Fix max

☑ Log scale

☑ Flip vertical

Figure 7.3 Example of microbial cell number distribution.



Data reading finished!

Selected file: bcase_dist.out

NX: 20
NY: 20
LX: 200
LY: 40

☐ Old Flow2D

Search text: G__SO

MIN 0.12449
MAX 0.8405

☐ Fix min
☐ Fix max
☐ Log scale
☑ Flip vertical

Figure 7.4 Oil saturation in a 2D vertical cross-section.

8 References

[1] UTCHEM-9.0, “Technical Documentation, A Three-Dimensional Chemical Flood Simulator,” The University of Texas at Austin, July 2000.

[2] K. Aziz and A. Settari, *Petroleum Reservoir Simulation*. London and New York: Elsevier Applied Science Publishers, 1979.

[3] D. M. Meter and R. B. Bird, “Tube Flow of Non-Newtonian Polymer Solutions: Part I. Laminar Flow and Rheological Models,” *A.I.Ch.E*, vol. 10, no. 6, pp. 878-881, 1964.

[4] R. C. Weast, Ed. *CRC Handbook of Chemistry and Physics*, 63rd ed. Boca Raton, Florida: CRC Press, Inc., 1982-1983.

[5] A. Lohne, O. Nødland, A. Stavland, and A. Hiorth, “A model for non-Newtonian flow in porous media at different flow regimes,” *Computational Geosciences*, journal article vol. 21, no. December, December 2017, doi: 10.1007/s10596-017-9692-6.

[6] E. A. Lange and C. Huh, “A Polymer Thermal Decomposition Model and Its Application in Chemical EOR Process Simulation,” presented at the SPE/DOE Improved Oil Recovery Symposium, Tulsa, Oklahoma, 1994/1/1/, 1994. [Online]. Available: https://doi.org/10.2118/27822-MS.

[7] G. Chauveteau, “Fundamental Criteria in Polymer Flow Through Porous Media,” in *Water-Soluble Polymers*, J. E. Glass Ed., (Adv. in Chem. Ser., no. 213), 1986.

[8] A. Omari, M. Moan, and G. Chauveteau, “Wall effects in the flow of flexible polymer solutions through small pores,” *Rheologica Acta*, vol. 28, no. 6, pp. 520-526, 1989.

[9] A. M. Howe, A. Clarke, and D. Giernalczyk, “Flow of concentrated viscoelastic polymer solutions in porous media: effect of Mw and concentration on elastic turbulence onset in various geometries,” *Soft Matter*, vol. 11, no. 32, pp. 6419-6431, 2015, doi: 10.1039/c5sm01042j.

[10] P. Khorrami and C. J. Radke, “A Random Scission Model for Chemical Degradation of Polymer Solutions,” *SPE Reservoir Engineering*, vol. 3, no. 01, pp. 349-352, 1988/2/1/ 1988, doi: 10.2118/15197-PA.

[11] A. Stavland, B. Kvanvik, and A. Lohne, “Simulation Model for Predicting Placement of Gels,” in *the SPE Annual Technical Conference & Exhibition*, New Orleans, LA., 25-28 Sept. 1994, no. 28600, doi: 10.2118/28600-MS.

[12] D. W. Peaceman, “Interpretation of Well-Block Pressures in Numerical Reservoir Simulation With Nonsquare Grid Blocks and Anisotropic Permeability,” *SPE Journal*, no. June, pp. 531-543, 1983, doi: 10.2118/10528-PA.

[13] J. E. Killough, “Reservoir Simulation With History-Dependent Saturation Functions,” in *Trans. AIME*, vol. 261, 1976, pp. 37-48, (*SPEJ* , Feb., **16**).

[14] A. Lohne, J. O. Helland, G. A. Virnovsky, and K. O. Vatne, “Statfjord three-phase steady-state core flood interpretation - Tarbert,” IRIS, - 2008/012, 2008.

[15] C. S. Land, “Calculation of Imbibition Relative Permeability for Two- and Three-Phase Flow From Rock Properties,” in *Trans. AIME*, vol. 243, 1968, pp. 149-156.

[16] C. S. Land, “Comparison of Calculated with Experimental Imbibition Relative Permeability,” in *Trans. AIME*, vol. 251, 1971, pp. 419-425.

Appendix A - Keyword list

9 Keywords

RUNSECT

Appendix B - 2D cross-sectional example

This appendix provides an example input file for microbial growth in a 2D cross-sectional model. Some results from running this file is shown in Figure B.1 to Figure B.11. This case demonstrates some of the problems that can be simulated with the *Bugsim* program. It is assumed native bacteria already exist in the reservoir exhibiting a slow growth rate on oil. Injection of external substrate and nutrient stimulates the biomass production. It is modelled that polymer and gas are produced, resulting in increased water viscosity and reduced oil viscosity. An increased oil production rate is seen in Figure B.1.

Note that the intention with this example is to demonstrate functionalities of *Bugsim* and that rather arbitrary parameters may have been used.

A problem encountered in this early stage is that the biomass growth model (which is based on the common models found in the literature) is not very suited for establishing different steady-state population densities when growing on different substrates (e.g., oil versus external substrate). This is something that must be evaluated in the future.

```
------------------------------- example ----------------------------------
runsect
--------------------------------------------------------------------------
title
This is a test in a 2D cross-section
of Bugsim with Bacterial growth
/
griddim
*nx ny nz
20 1 20 /  a total of 20*20=400 grid blocks

nrock
2 /
phases
1 1 1 /

units
*  L t Q      K  P   V    rho    (properties)
*  m d m**3/d mD bar m**3 g/cc   (units)
1 0 3      0  3   2    0 /
components
*  type name /    Defining components
oil   OIL1 /
water WAT1 /
gas   CH4 /
polymer   xanthan /
bac B1 /
substrate S1 /  external substrate
nut N1 / external nutrient, only the three first characters are used
nut Noil / nutrient from oil partitioning between oil/water
tracer TR1 /
/

*  component order will be re-arranged with type in the order:
*  WATER, OIL, GAS, POLYMER,SUBSTRATE,NUTRIENT,TRACER
*  rearranged order:
*  1     2     3     4        5   6   7   8     9
*  WAT1, OIL1, CH4 , xanthan, S1, B1, N1, Noil, TR1

nwells
2 /
```

```
----------------------------------------------------------------------
gridsect
----------------------------------------------------------------------
dxx
20*10 /  nx entries for block x-dimension
dyc
10 /  constant y-dimension 10m
dzz
20*2 /  nz entries for block z-dimension

permxl
1    5   10 100 200 500 500 200 300 10
20 1000 1500 100 200  10 100 500 100 50 /  permx by layer (nz values)

copy
permx permz 0.1 /  permz=0.1*permx
permx permy /
/

poroc
0.2 /  constant porosity

rocktype
200*1 200*2 /  80*1 100*2 40*1 40*2 80*1 20*2 40*1 /



---------------------------------------------------------------------
propsect
---------------------------------------------------------------------
tull
/    not recognized, a warning will be issued in log-file
 /
pref
200 /  make sure reference pressure 200 bar is used

cmpprop
*  Name Prefc Bw density Cw    PrefV TrefV°C visc  BTw   BPw
  WAT1 80    1.0  1.0   4e-5  1.    20.     1.02  1780. 0. /
  OIL1 100   1.0  0.75  1e-4  1.     90.    0.90  1500. 0.0016 / (same input as wat)

*  Tracer
*  name trtype cunit kow
TR1  1      ppm   0    /  water tracer (kow=0)
*  Polymer
*  name     Pref   Bf  density  compr
xanthan  100    1   1        0  /
*  eta   hug1  hug2  Tref  Tfact
4000  0.4   0.1   20    0.005 /  viscosity at low shear
*  meter_P  meter_gh
1.75      20  /                 shear thinning
-------Start Bacteria input -----------------------------------------------
*  the bacteria may grow solely on oil, but may grow faster if external substrate
*  and nutrient is supplied
*name Pref  Bf  density  compr  Dpmean cwater  Xm    mx iads
 B1   100   1   1        0      2      0.8   .001   .25  1 /
*  b   Qmax
100000 0.0001 /  adsorption parameters (iads=1)
*  nS  nNg  nNt  nSPt  - total numbers of S, N groups, N components and Products
2   1    2    3  /
*  Sname mym   KS    Y    nSP   - substrate 1
OIL1  .5   .01   0.5  2 /
*  Pname UP   -  product 1 from substrate 1
xanthan 0.2
*  Pname UP   -  product 2 from substrate 1
CH4  0.2 /  0.2 gram CH4/ gram OIL1 used
*  Sname mym   KS    Y   nSP   - substrate 2
S1    3.0   .0001 0.5 1 /
```

```
*  Pname UP   -  product 1 from substrate 2
CH4  0.2 /
*nNg=1  nutrient group 1
*nN[1]=2 number of nutrients in Nutrient group 1 (2 nutrients)
2
*Nname KN      UN (kcn=1,2) - name and parameters for 1st N in Ng1
 Noil  0.2     0.1
 N1    0.2     0.1 / 2nd N in Ngroup 1
*kd   tave dName  - decay rate and no end product
0.1  0  /
-------End Bacteria input ------------------------------------------------------
*  Substrate properties - one record
*name Pref Bf Sdensity compr kowr
S1   1    1  1        0     0.5 / S1 is partitioning between oil and water
*  Nutrient properties - one record
*name kowr Unit
N1   0    ppm /
*name kowr Unit
Noil .999 ppm /

*  Gas properties - two records (partitioning and viscosity)
*Name Prefw Bdg   density Cg   rgo0 rgop   Kwo
 CH4  100   0.01  0.0012  0.01   50  0.5  .02 /
*  PrefV Tref°C visc  BTo   BPo viscl BTviscl BPviscl
1.    90     0.012  200. 0.0025 0.1 / BTviscl=0 => computed internally form viscl

*  name - 1st oil component (will overwrite previous data entered)
OIL1
*  Pref  Bw   ro    compr
100   1.0  0.75  1e-4
*  VPo  VTo   Vo    BT    BP
1.   90.   0.90  1500. 0.0016
/ terminates input (record) for this component
/  - empty slash - terminates the keyword cmpprop
rockprop
*  Crock
100 1e-5 /
corey
*1 w .4 2.5 0.1 /   water and oil from tables below
*1 o .8 2.0 0.15 /
1 g .9  2   0 /
2 g 1.0 1.5 0 /
/

*  Pcowform
*  ir CL      CR     C0     EL  ER   SL   SR      Pcow = CL/(Sw-SL)**EL - CR/(SR-Sw)**ER
+ C0
  1  .0001   0.0001 0.    2.  1.5  0.1  0.9 /
  2   /  defaults to zero Pc
   /
--/ ends Pcow input
krpctab1
*  ir  js  jkrw  jkro  jpcow  idir krspl  pcspl
  1   1   2     3     0      1  /
0.09000  0.00000  1.00000   4.55000
0.11100  0.00010  0.87965   0.99629
0.11763  0.00015  0.85692   0.49629
0.12514  0.00020  0.83942   0.19629
0.14586  0.00028  0.81855   0.09629
0.20959  0.00112  0.69013   0.04629
0.26330  0.00377  0.51262   0.02629
0.28251  0.00600  0.45064   0.01329
0.28898  0.00669  0.43315   0.00429
0.32306  0.01041  0.33696  -0.02371
0.34028  0.01172  0.29350  -0.02771
0.35096  0.01257  0.27083  -0.02971
0.35653  0.01298  0.25991  -0.03071
0.37431  0.01444  0.22848  -0.03371
```

```
0.39243  0.01612  0.20100  -0.03671
0.41194  0.01804  0.17532  -0.03971
0.43696  0.02120  0.14689  -0.04371
0.46336  0.02563  0.12189  -0.04771
0.48869  0.03203  0.10147  -0.05171
0.53227  0.04985  0.07348  -0.05871
0.56736  0.07815  0.05622  -0.06471
0.59976  0.11571  0.04337  -0.07071
0.63499  0.16993  0.03221  -0.07771
0.70214  0.29443  0.01692  -0.09371
0.78874  0.46203  0.00590  -0.12371
0.82755  0.53657  0.00326  -0.14371
0.85639  0.59305  0.00198  -0.16371
0.89274  0.66950  0.00099  -0.20371
0.92775  0.75873  0.00048  -0.30371
0.94109  0.80677  0.00033  -0.50371
0.94682  0.84589  0.00020  -1.00371
0.95000  0.91000  0.00000  -1.50000
 /

*   ir  js  jkrw  jkro  jpcow  idir krspl  pcspl
  2   1   2     3     0      1  /
0.152000  0.000000  1.000000  14.000000
0.152670  0.000000  0.999990   4.999180
0.158550  0.000000  0.977280   0.999180
0.162530  0.000010  0.961720   0.749180
0.169020  0.000020  0.935930   0.499180
0.177650  0.000040  0.899920   0.299180
0.202400  0.000080  0.792990   0.099180
0.216510  0.000100  0.737020   0.074180
0.241760  0.000140  0.634530   0.049180
0.256390  0.000190  0.573670   0.039180
0.273750  0.000270  0.505240   0.029180
0.293050  0.000440  0.429730   0.019180
0.301680  0.000580  0.400710   0.014180
0.308360  0.000720  0.382040   0.009180
0.317300  0.000920  0.357640  -0.000820
0.321210  0.001010  0.346420  -0.005820
0.326340  0.001100  0.333470  -0.010820
0.353040  0.001360  0.271970  -0.020820
0.364220  0.001480  0.249200  -0.023320
0.376640  0.001620  0.225860  -0.025820
0.389880  0.001780  0.204280  -0.028320
0.403860  0.001970  0.183530  -0.030820
0.417800  0.002180  0.165530  -0.033320
0.432230  0.002430  0.148710  -0.035820
0.460750  0.003080  0.120610  -0.040820
0.475080  0.003500  0.108600  -0.043320
0.489300  0.004010  0.097820  -0.045820
0.503390  0.004650  0.088250  -0.048320
0.517040  0.005360  0.079740  -0.050820
0.543560  0.007230  0.065400  -0.055820
0.568860  0.009910  0.053800  -0.060820
0.614330  0.017590  0.037210  -0.070820
0.654000  0.029850  0.026260  -0.080820
0.717390  0.072480  0.013880  -0.100820
0.861430  0.434050  0.001310  -0.200820
0.931950  0.801620  0.000020  -0.500820
0.947080  0.865210  0.000000  -1.000820
0.950000  0.871930  0.000000  -9.000000
/
/

bioplug
1 /

-------------------------------------------------------------------
initsect
```

```
-------------------------------------------------------------------
swinitc
0.16 / constant initial Sw

pinitc
200 /

phaseconc-w
1  0 B1 1e-8 /  initial water phase concentration

phaseconc-o
0 1 0 Noil 100 / initial oil phase concentration
*the three first components + nutrient Noil in ppm

phaseconc-o
OIL1 1  WAT1 0 Noil 100 / alternative input method using names
phaseconc-g
CH4 1 /


tinit
*  T0 zT0 Tgrad
70 /
--------------------------------------------------------------------
compsect
--------------------------------------------------------------------

PrintKey
Pr_S 111
Pr_P 111
Pr_CT 1
Pr_C  111
Pr_Vis 111
  /

*a total of 9 components is defined
*
PrintCF
1 0 7*1 /  water concentrations
0 1 1 0 1 0 0 1 0  /   oil concentrations
0 0 1 /   gas phase concentrations
PrintCTF
9*1 /   total concentrations

Pr_MPN   - prints cell numbers (part of line after keyword is discarded)
Pr_tage
Pr_ads
Pr_GOR
Pr_rkf

history
30 0.5 / history print frequency minimum of every 30 timestep or every 0.5 hr
timestep
*dt0 qxmax dsmax dtfmax dtmax
0    0.8   0.5   4   /

tollit
*tollit itmax
1e-12  1000 /


-------------------------------------------------------------------
wellsect
-------------------------------------------------------------------

welldef
winj  /
*dir i1  j1   k1  k2  rw  open
```

```
 3   1   1    1   10   .1  1 /
 3   1   1    11  15   .1  0 / closed well connections in layer 11-15
 3   1   1    16  20   .1  1 /
/
welldef
wprod /
*dir i1  j1   k1  k2  rw
 3   20  1   1    20   .1 /
/

winject
*name   r/p rlim plim
winj    r   20   250 /
*  fw fg
1 0  /
*  Cw
1 0 /
0 1 /
0 0 1/
/

wprod
*name   r/p rlim plim
wprod p   1000 190 /
/

time
0.1  /

dtime
0.9  /
winject
*name   r/p rlim plim
winj    r   100   300 /
*  fw fg
1  0  /
*  Cw
1 B1 1e-8 /  injects a small amount of bacteria
0 1 0 /
0 0 1 /
/
dtime
5*10 9*50 /

winject
*name   r/p rlim plim
winj    r   100   300 /
*  fw fg
1  0  /
*  Cw
1 0 /
0 1 0 /
0 0 1 /
/
dtime
10*50 /

winject
*name   r/p rlim plim
winj    r   100   300 /
*  fw fg
1  /
*  Cw

1 B1 1e-10 S1 .01 N1 10 /   injects new bacteria+food
0 1 0 /
0 0 0 /
/
```

```
dtime
5*50 /

winject
*name   r/p rlim plim
winj    r   100   300 /     injects only water
*   fw fg
1 0 /
*   Cw
1 /
0 1 0 /
0 0 1 /
/
dtime
50 /
dtime
50 /
dtime
3*50 100 /

end
```

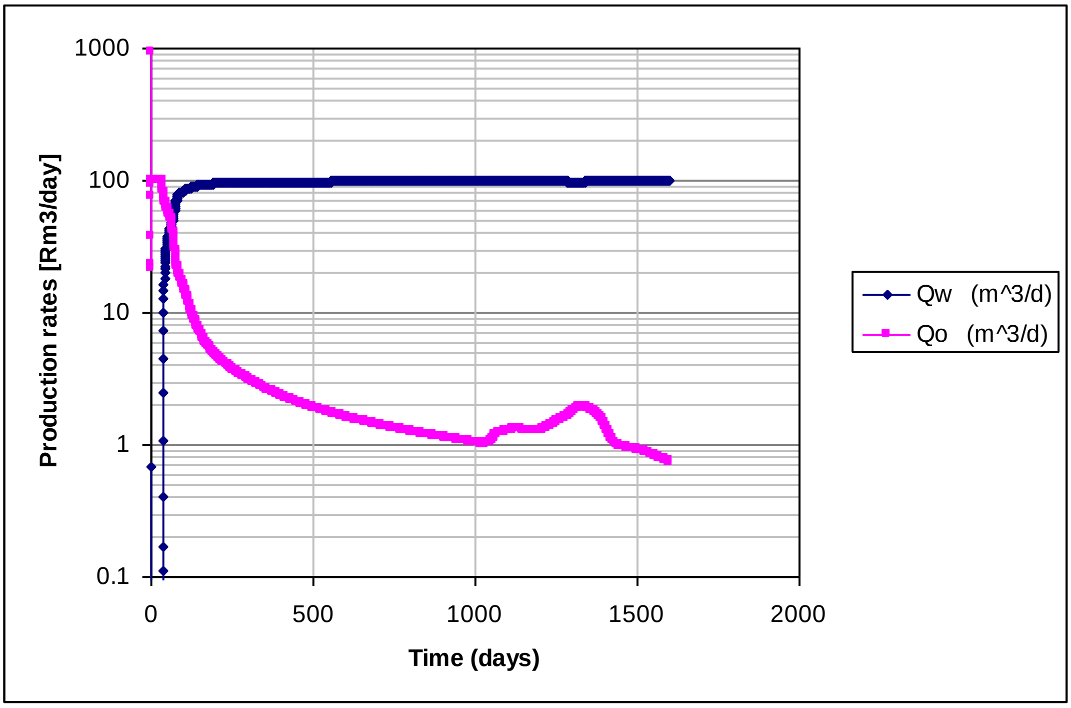


Figure B.1 Production rates in 2D cross section example. Extra substrate and nutrient injected in the period 1000-1250 days.

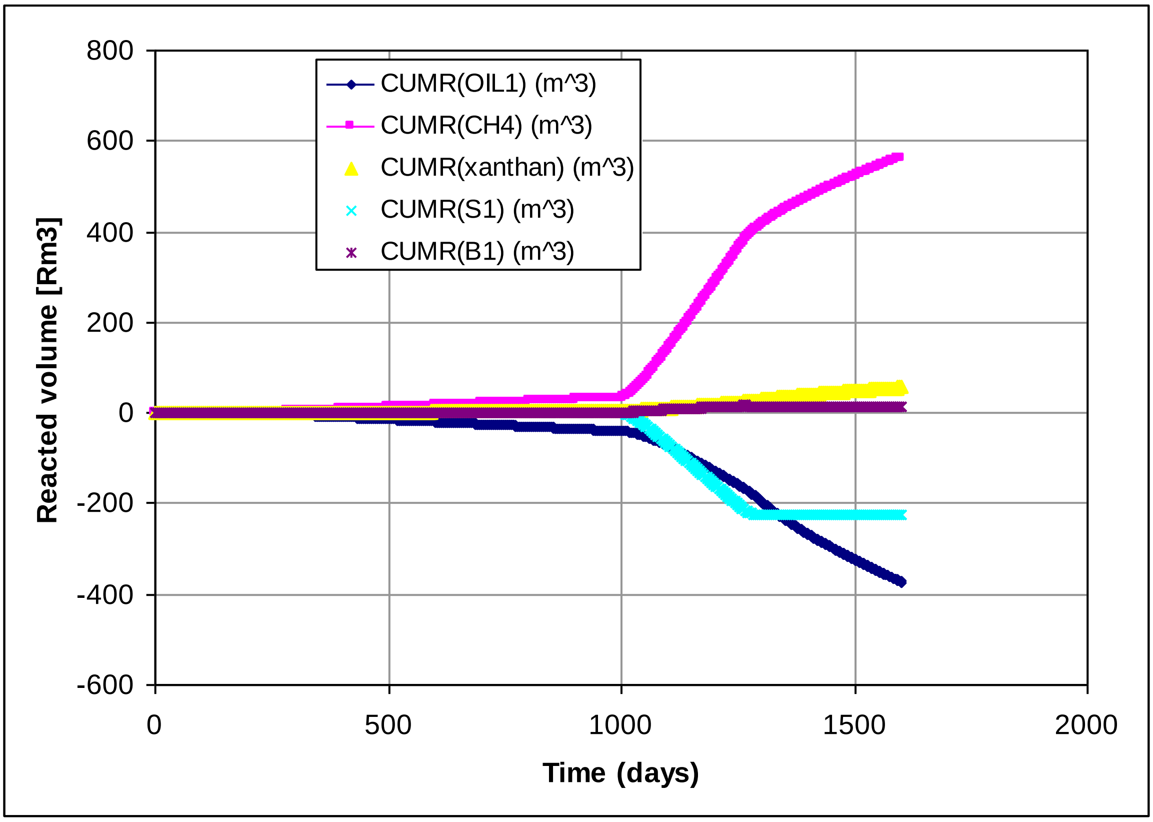


Figure B.2 Reacted volumes in 2D cross section example. Extra substrate and nutrient injected in the period 1000-1250 days.

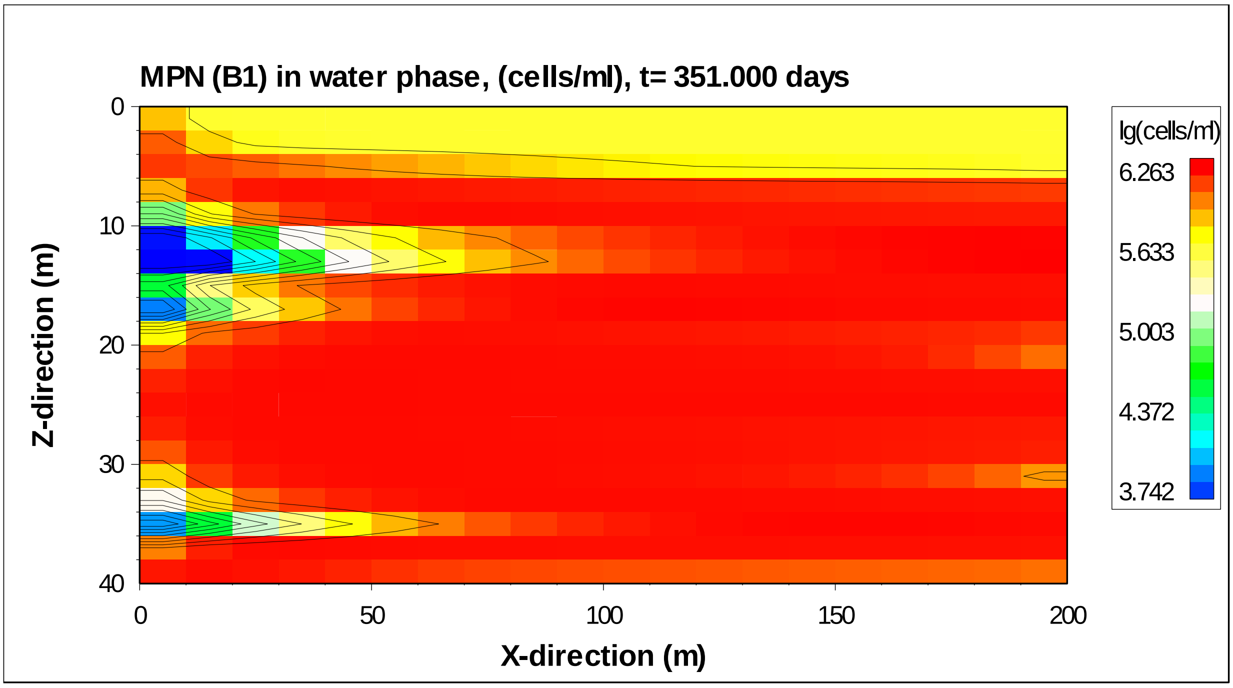


Figure B.3 Un-attached cell number distribution without external food. After 350 days of water injection.

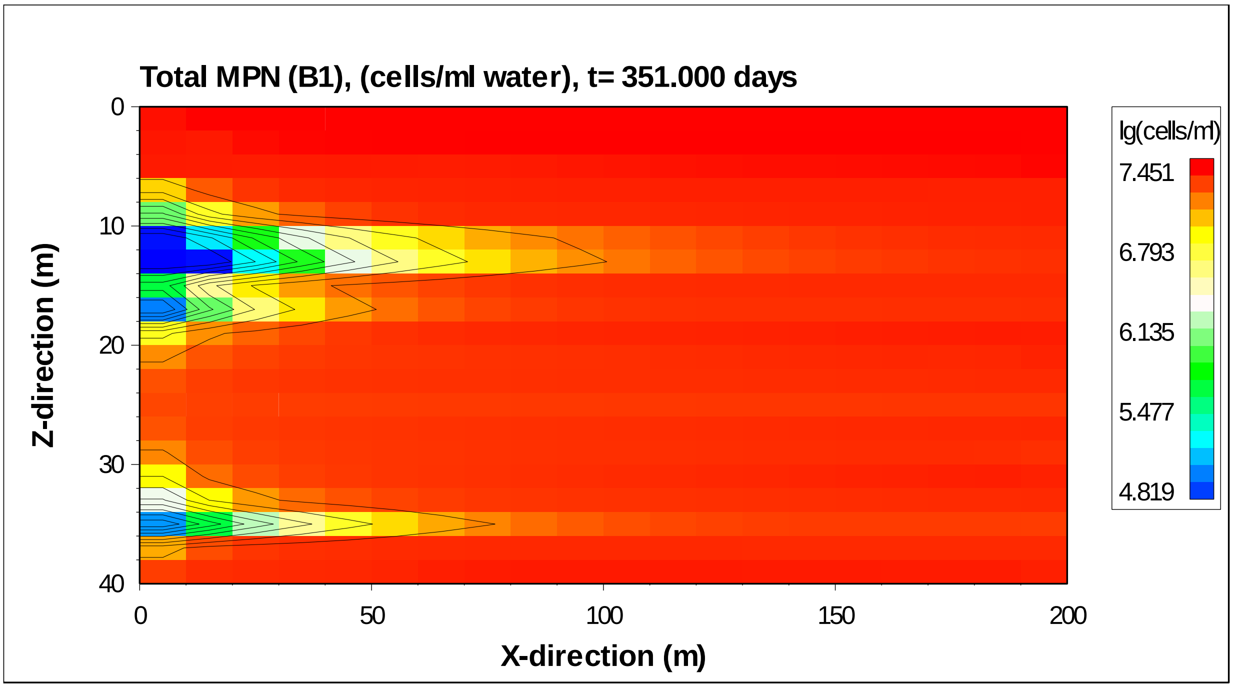


Figure B.4 Total cell number (free+attached) distribution without external food. After 350 days of water injection.

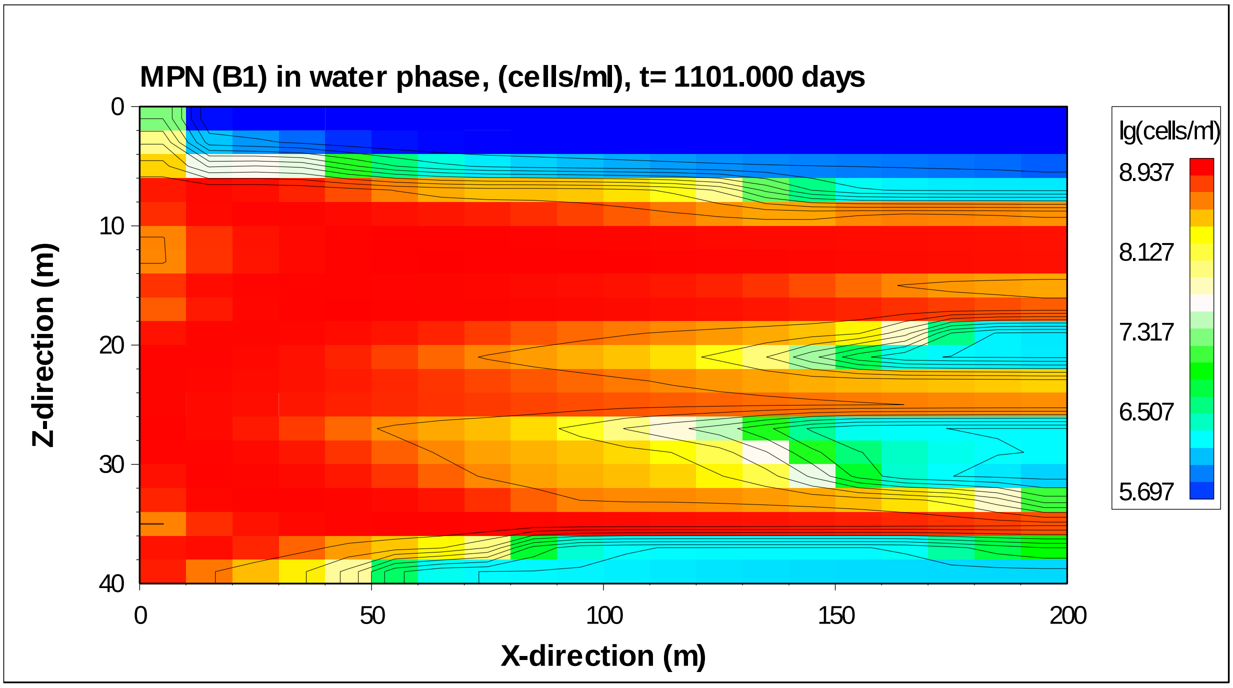


Figure B.5 Un-attached cell number distribution after injection of external food for 100 days (starting from *t*=1001 days).

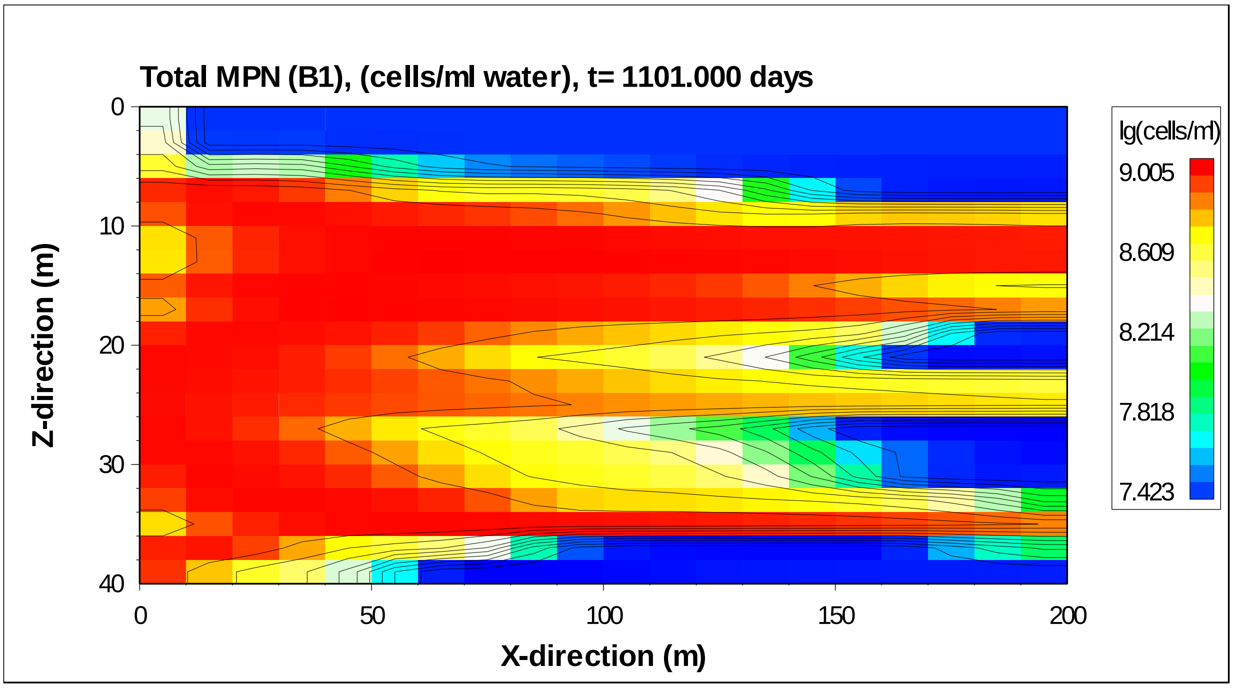


Figure B.6 Total cell number distribution after injection of external food for 100 days (starting from *t*=1001 days).

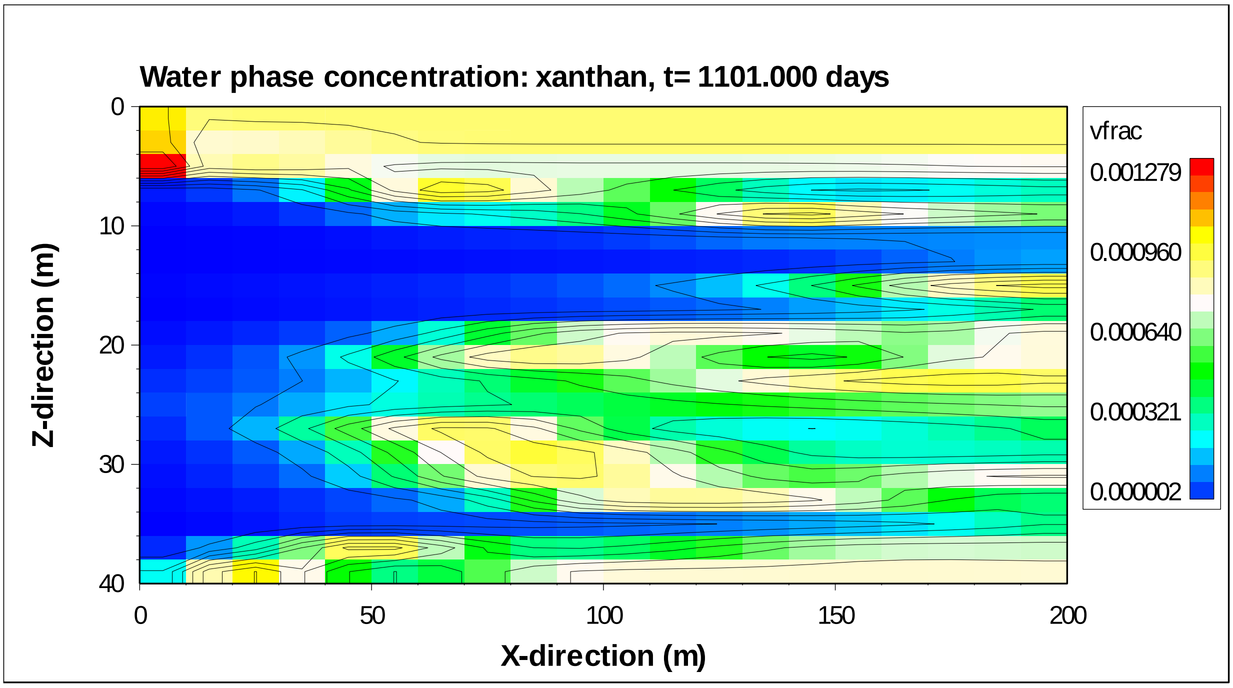


Figure B.7 Produced bio-polymer distribution after injection of external food for 100 days (starting from *t*=1001 days).

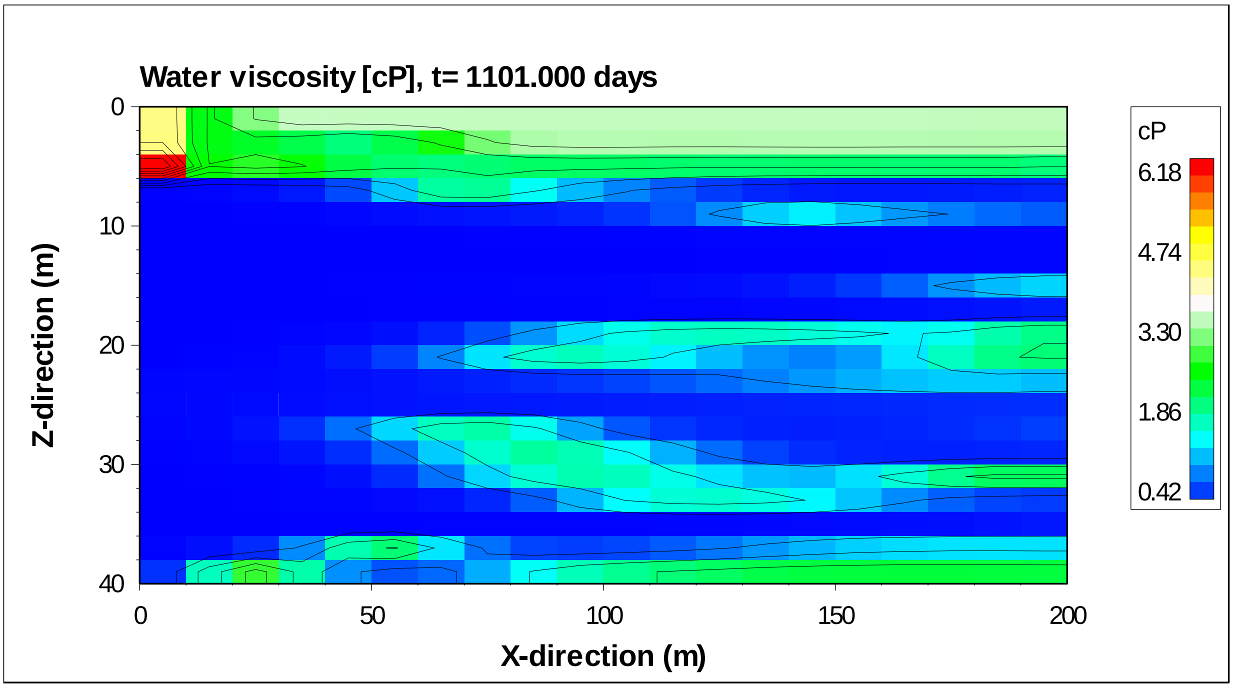


Figure B.8 Increased water viscosity because of bio-produced polymer after injection of external food for 100 days (starting from *t*=1001 days).

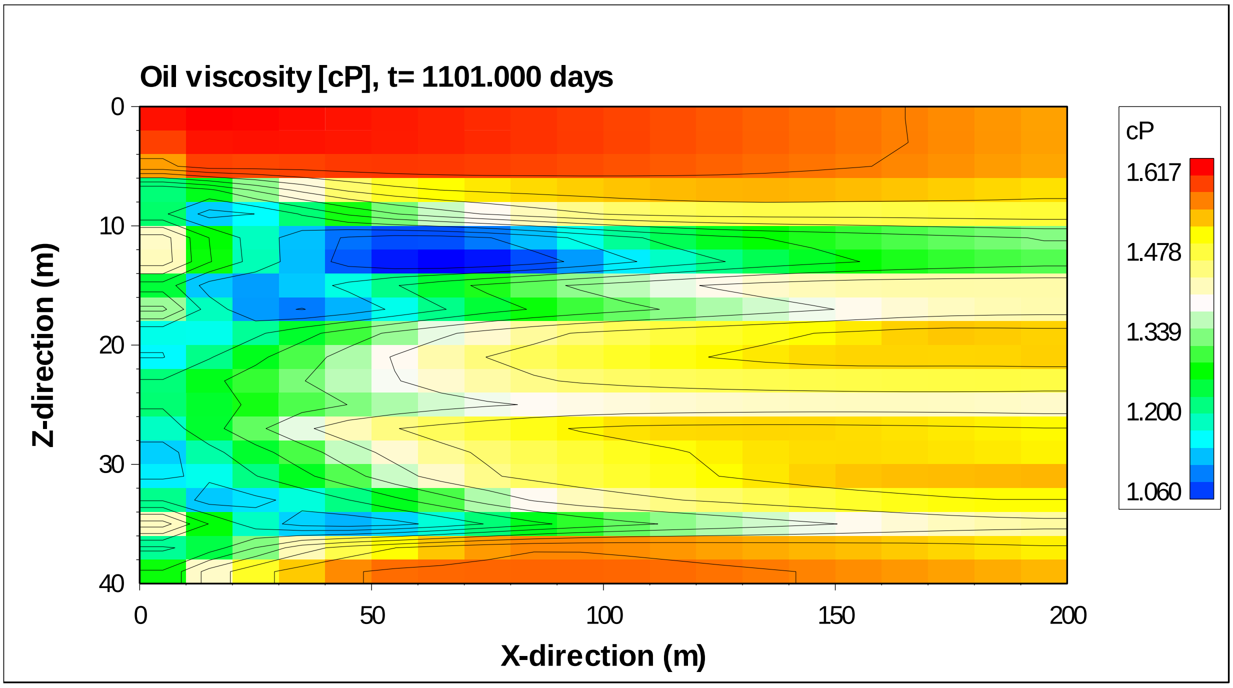


Figure B.9 Decreased oil viscosity because of bio-produced gas after injection of external food for 100 days (starting from *t*=1001 days).

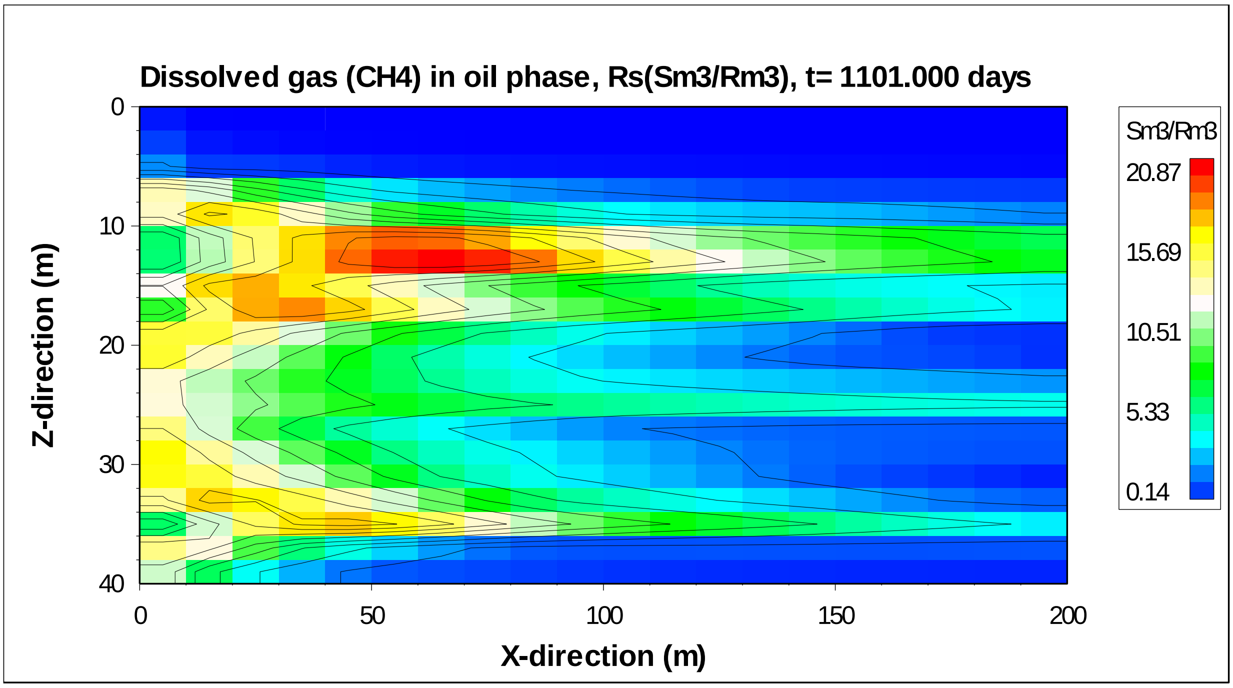


Figure B.10 Distribution of bio-produced gas after injection of external food for 100 days (starting from *t*=1001 days).

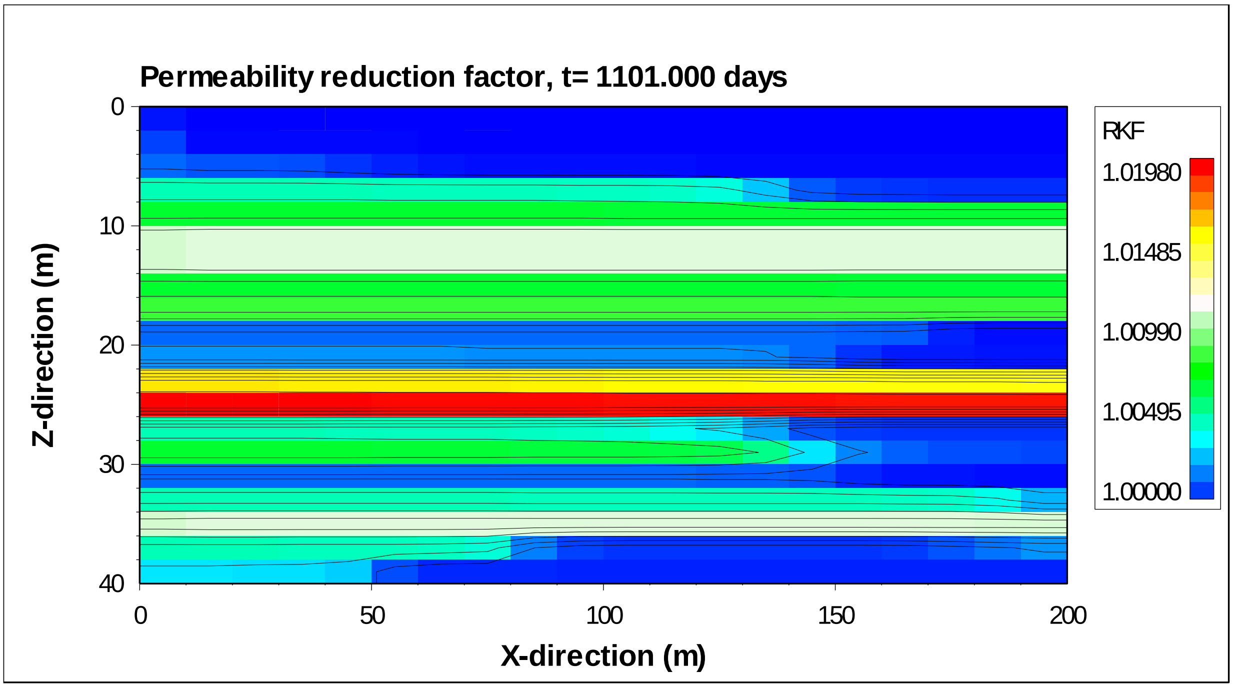


Figure B.11 Permeability reduction factor after injection of external food for 100 days (starting from *t*=1001 days).