Center for Applied Geosciences
University of Tübingen

Institute of Fluid Mechanics
and Computer Applications in Civil Engineering
University of Hannover

RockFlow Manual
RFD Input Description
Version 3.9 First Draft

O. Kolditz[(ZAG)], R. Kaiser[(ISEB)]
A. Habbar[(ISEB)], M. Kohlmeier[(ISEB)]
J. de Jonge[(ZAG)], M. Beinhorn[(ZAG)]
M. Xie[(ZAG)], T. Kalbacher[(ZAG)]
G. Ungruh[(ISEB)], S. Bauer[(ZAG)]
W. Wang[(ZAG)], Ch. McDermott[(ZAG)]
C. Chen[(ZAG)], C. Beyer[(ZAG)]
J. Gronewold[(ZAG)] D. Kemmler[(ZAG)]

ROCKFLOW – Preprint
[2003-..]

Tübingen/Hannover – July 2003

Contents

1 Preface

Previous and current work on ROCKFLOW

Version 1

Wollrath, J. (1990): Ein Strömungs- und Transportmodell für klüftiges Gestein und Untersuchungen zu homogenen Ersatzsystemen. - Bericht Nr.28/1990, Universität Hannover, Institut für Strömungsmechanik, Dissertationsschrift.

Kröhn, K.-P. (1991): Simulation von Transportvorgängen im klüftigen Gestein mit der Methode der Finiten-Elemente. - Bericht Nr. 29/1991, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen, Universität Hannover, Dissertationsschrift.

Version 2

Helmig, R. (1993): Theorie und Numerik der Mehrphasenströmungen in geklüftet-porösen Medien. - Dissertationsschrift, Bericht Nr. 34/1993, Institut für Strömungsmechanik, Universität Hannover.

Shao, H. (1994): Simulation von Strömungs- und Transportvorgängen in geklüftet porösen Medien mit gekoppelten Finite-Elemente- und Rand-Element-Methoden. - Dissertationsschrift, Bericht Nr. 37/1994, Institut für Strömungsmechanik, Universität Hannover.

Lege, T. (1995): Modellierung des Kluftgesteins als geologische Barriere für Deponien. - Dissertationsschrift, Institut für Strömungsmechanik und elektronisches Rechnen im Bauwesen, Universität Hannover, 213S.

Ratke, R. & Kolditz, O. & Zielke, W. (1996a,b): ROCKFLOW-DM2 - 3-D Dichteströmungsmodell. - Technischer Bericht, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen, Universität Han-nover.

Kolditz, O. (1996): Stoff- und Wärmetransport im Kluftgestein. - Habilitationsschrift, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen, Universität Hannover, 392S.

Barlag C. (1997): Adaptive Methoden zur Modellierung von Stofftransport im Kluftgestein. - Dissertationsschrift, Institut für Strömungsmechanik, Universität Hannover.

Version 3

Kolditz O., Kaiser R., Thorenz C. & Schulze-Ruhfus M.: ROCKFLOW User's Manual Release 3.1, June 1997, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen, Universität Han-nover.

Kolditz O., Habbar A., Kaiser R., Thorenz C.: ROCKFLOW User's Manual Release 3.2, December 1997, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen, Universität Hannover.

Kolditz O., Habbar A., Kaiser R., Thorenz C.: ROCKFLOW User's Manual Release 3.3, November 1998, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen, Universität Hannover.

Kolditz O., Habbar A., Kaiser R., Thorenz C.: ROCKFLOW User's Manual Release 3.4, November 1999, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen, Universität Hannover.

Thorenz, C.: Model Adaptive Simulation of Multiphase and Density Driven Flow in Fractured and Porous Media. Dissertation, Institut für Strömungsmechanik, Universität Hannover, Bericht Nr. 62/2001.

Kaiser, R.: Gitteradaption für die Finite-Elemente-Modellierung gekoppelter Prozesse in geklüftet-porösen Medien. Dissertation, Institut für Strömungsmechanik, Universität Hannover, Bericht Nr. 63/2001.

Rother, T.: Geometric Modelling of Geo-Systems. Dissertation, Institut für Strömungsmechanik, Universität Hannover, Bericht Nr. 64/2001.

Habbar, A.: Direkte und Inverse Modellierung reaktiver Transportprozesse in klüftig-porösen Medien. Dissertation, Institut für Strömungsmechanik, Universität Hannover, Bericht Nr. 65/2001.

Activities and persons to contact

ISEB, Hannover:

René Kaiser
head of the groundwater modelling group,
grid adaptation (RF-ADP)

Abderrahmane Habbar
reactive transport (RF-RTM), inverse modelling,
solver, data concepts

Martin Kohlmeier
deformation processes (RF-DM), mesh generation,
coupled THM-modelling, material modelling, GUI (RF-SHELL)

Sylvia Moenickes
geometric modelling, mesh generation (RF-HGM)

Gesa Ungruh
THM-modelling, mesh generation, material modelling

Alumni:

Carsten Thorenz
multiphase and density flow (RF-MM),
method of characteristics, data filter (rf2tec)

Thomas Rother
geometric modelling, mesh generation (RF-HGM)

ZAG, Tübingen:

Olaf Kolditz
Computational mechanics, software engineering

Joëlle de Jonge
Non-isothermal multi-phase / multi-componental flow

Martin Beinhorn
Density-dependent flow with free surfaces

Thomas Kalbacher
Geometric modelling, CAD interfaces

Mingliang Xie
Geochemical processes in multi-phase systems

Wenqing Wang
Inelastic deformation processes in unsaturated systems

Sebatian Bauer
Biogeochemical processes in multi-phase systems

Christopher McDermott
Hydrogeomechanics

Cui Chen
GIS interface, graphical user interfaces

Jan Gronewold
Data bank interfaces, internet information systems

Christof Beyer
Biogeochemical processes, internet information systems

Dany Kemmler
Parallel computing

Associated activities:

Prof. Taniguchi (Okayama University)
Geometric modelling and mesh generation

last modified: OK July 20, 2003

2 Control

2.1 Control (keyword: #CONTROL)

The keyword **#CONTROL** is the control unit of the general program execution (e.g. simu-lation mode, grid adaptation or parallel computing) and affects all processes defined in the process–keywords. Defaults are printed bold.

#CONTROL

The control–keyword can contain a number of optional subkeywords (labeled by a $ sign):

$MODEX

Parameter	Value	Meaning
N1		Simulation control (mode)
	-2	RFD input file will be tested. Result in RFE file.
	-1	RFD and RFI input files will be tested. Result in RFE file.
	1	Execution of the Simulation

$GRIDADAPTATION

Parameter	Value	Meaning
N1		Grid adaptation (not yet realized in the THM model)
	0	Disable
	1	Enable

$PARALLEL_COMPUTIMG

Parameter	Value	Meaning
N1		Parallel Computing (not yet realized in the THM model)
	0	Disable
	1	Enable

$NUMBER_OF_GROUPS

The keyword specifies the number of material groups (default: **1**).

last modified: RK/MK July 10, 2003

3 Models / Processes

3.1 Process (keyword: #PROCESS_*process_name*)

The keywords **#PROCESS**_*process_name* specify the thermal–hydraulic–mechanical (and chemical) model (THM(C)) where the string *process_name* defines the considered processes:

process_name	Acronym	Process
HEAT_TRANSPORT	**T**	Thermal
FLUID_FLOW	**H**	Hydraulic
SOLID_DEFORMATION	**M**	Mechanical
MASS_TRANSPORT	**C**	Chemical
(not yet realized in the THM model)		

These keywords must only be specified if the appropriate processes should be considered. Otherwise the processes are disabled and the data structures are not generated. If a process is enabled by specifying the appropriate keyword, its data structure is generated and the output file (.rfo) is accordingly adjusted.

Each of these keywords can contain a number of standard or of standard or dedicated to sub-keywords (labeled by a $ sign).

Standard sub-keywords, which are valid for each process, are described below.

$ACTIVATION

Parameter	Value	Meaning
N1		Process (in)activation
	0	Inactive (No element routines, no assembling or solving, process dependent data structures are generated)
	1	Active

$INFLUENCING_PROCESSES

Currently, these coupling specifications are not implemented in the dedicated coupling algorithms. Controllable couplings will be listed below as soon as they are available.

Parameter	Value	Meaning
N1, N2, ..., Nn	'**process_name** 1' '**process_name** 2' ... '**process_name** n' ' '	Activation of specified couplings Activation of the coupling from n specified processes No couplings activated

Sub-keywords that are appendant to a process are described separately in the ensuing sections.

3.2 Process Heat Transport (#PROCESS_HEAT_TRANSPORT)

No sub-keywords necessary.

3.3 Process Fluid Flow (#PROCESS_FLUID_FLOW)

$NUMBER_OF_PHASES

The keyword specifies the number of moving fluid phases (default: **1**).

$NUMBER_OF_FLUID_COMPONENTS

The keyword specifies the number of fluid components (e.g. air, water) contained in a fluid phase (default: **0**).

$NON_ISOTHERMAL_FLOW

Parameter	Value	Meaning
N1		Nonisothermal flow
	0	Disable
	1	Enable (corresponding phase changes are taken into account

$FLUID_FLOW_MODEL

Parameter	Value	Meaning
N1		Fluid flow model
	0	Darcy law (linear flow behavior)
	1	Forchheimer law (non-linear flow behavior)
	2	Richards formulation (unsaturated flow)

Keyword combinations for flow processes

FLUID FLOW PROCESS	NUMBER_ OF_ PHASES	NUMBER_OF_ FLUID_ COMPONENTS	NON_ISO- THERMAL_ FLOW	FLUID_ FLOW_ MODEL
Saturated liquid flow	1	0	0	0, 1
Unsaturated flow (Richards model)	1 (2)	0	0	2
Multiphase flow (two-phase flow)	2	0	0	0, 1
Non-isothermal unsaturated flow (Richards model)	2	2	1	2
Non-isothermal multiphase flow	2	2	1	0, 1

3.4 Process Solid Deformation (#PROCESS_SOLID_DEFORMATION)

No sub-keywords necessary.

last modified: RK/MK July 10, 2003

4 NOD Node Data

Data Object - Nodes

```
typedef struct {        /* Knotenkoordinaten */
    double x;           /* Koordinaten */
    double y;
    double z;
    long node_start_number;     /* Knotennummer im Startnetz */
    double source;      /* Quelle */
    long index;         /* aktueller Index nach Umnummerierung, sonst -1 */
    long *elems1d;      /* zugehoerige 1D-Elemente */
    long *elems2d;      /* zugehoerige 2D-Elemente */
    long *elems3d;      /* zugehoerige 3D-Elemente */
    int anz1d;          /* Anzahl der zugehoerigen 1D-Elemente */
    int anz2d;          /* Anzahl der zugehoerigen 2D-Elemente */
    int anz3d;          /* Anzahl der zugehoerigen 3D-Elemente */
    long *plains;       /* Wird zum Aufbau des Flaechenverzeichnisses (Startnetz) gebraucht */
    int anz_plains;     /* Waehrend der Adaption werden diese Daten nicht aktualisiert !!!!  */
    long *edges;        /* Wird zum Aufbau des Kantenverzeichnisses (Startnetz) gebraucht */
    int anz_edges;      /* Waehrend der Adaption werden diese Daten nicht aktualisiert !!!!  */
    long *newnodes;     /* benachbarte neugenerierte Knoten fuer 9-Knoten-Elemente */
    int anz_new_nodes;

    int status;         /* -1: regulaerer Innenknoten
                           -2: irregulaerer Kantenknoten
                           -3: regulaerer Randknoten
                           -4: irregulaerer Flaechenknoten */
    int corner_node;    /* corner_node flag = 1 if Eckknoten */
    int free_surface;   /* 1: bewegliche Knoten oben */
                        /* 2: bewegliche Knoten unten */
    double *nval;       /* modellabhaengige Knotenwerte, z.B. h und conc */
    void *nval_intern;            /* interne modellabhaengige Knotendaten */
    Randbedingung *randbedingung; /* Zeiger auf Randbedingungen */
} Knoten;
```

5 OUT - Data Output

5.1 Keyword

#OUTPUT_EX

5.2 Methods

M	S	Subject	Format
0	?	Contours: all node quantities at defined time interval	RFO
1	OK	Time curve: all node quantities at given node defined by coordinates (x,y,z)	
2	?	Profiles, all time steps	
3	?	Profiles, defined time steps	
4	?	Time curve: all node quantities at given node defined by number	
5	OK	Profiles: X-Y plots of specified quantities at given time points	PLT
6	?	Profiles: X-Y plots of specified quantities for all time points	PLT
7	?	Contours: specified quantities at given time points	PLT
15	OK	Profiles: along polyline	PLT

M - method, S - status

5.3 Examples

Method 0

```
Contour plots at given time intervals
#OUTPUT_EX
 0 ; method
 1e5 ; output time interval
```

Method 1

```
Time curve of all quantities at node given by coordinates
#OUTPUT_EX
 1 ; method
 file.plt ; name of output file
 1.000000e+001  0.000000e+000  0.000000e+000 ; node geometry
```

Method 5

```
Profiles: X-Y plots of specified quantities at given time points
#OUTPUT_EX
 5 ; method
 file.plt ; name of output file
 7 ; number of output times
   1.0e+3 1.0e+4 2.0e+4 4.0e+4 6.0e+4 8.0e+4 1.0e+5 ; output times
   10. ; time radius
 4 ; number of output values
   X  PRESSURE1 SATURATION2 TEMPERATURE1 ; names of output values
```

Method 6

```
Profiles: X-Y plots of specified quantities for all time points
#OUTPUT_EX
 6 ; method
 file.plt ; name of output file
 4 ; number of output values
   Z   PRESS  HEAD  SATURATION ; names of output values
```

Method 7

```
#OUTPUT_EX
 7 ; method:
 file.plt ; name of output file
 1 ; number of output times
   86400. ; output time
   1. ; output time radius
 4 ; number of output values
   X  Y  Z  PRESS ; names of output values
```

Method 10

```
#OUTPUT_EX
; Beispiel fuer Ausgabe von Knotendaten in verschiedenen Materialgruppen
10 ; method
 file.plt ; name of output file
  1 1 2 ; mode method data_output_method time
  8.64e+8 ; output time
  1.0 ; output time radius
 5 ; number of output values
   Z  PRESS CONC_1 CONC_2  CONC_3 ; names of output values
```

Method 15

```
RFD file:
#OUTPUT_EX
; Profile along polyline
15 ; method
 file.plt ; name of output file
 POLYLINE1 ; polyline name

RFM file with polyline data:
#POLYLINE
POLYLINE1
2 ; number of polyline points
2 ; polyline type
0.0 0.0 0.0 0.000000e+000 5.000000e+002
0.0 6.0 0.0 0.000000e+000 5.000000e+002
0.000000e+000 ; epsilon
1 ; closed
```

last modified: OK 21.07.2003

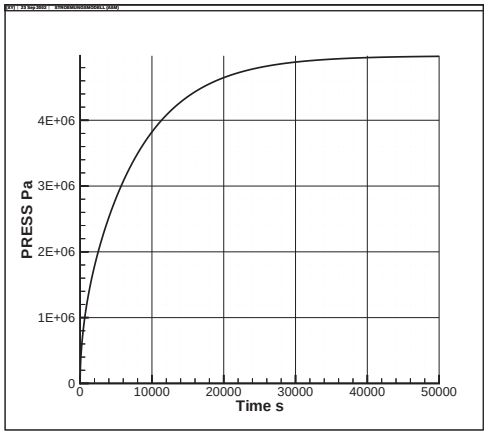

Figure 1: OUT method 1: Time curve of all quantities at node given by coordinates

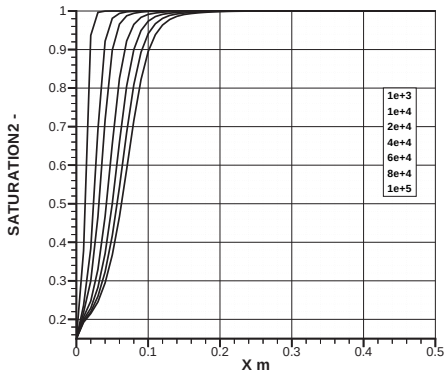

Figure 2: OUT method 5: X-Y plots of specified quantities at given time points

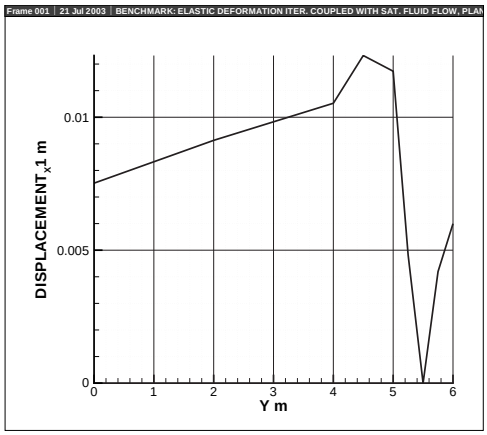

Figure 3: OUT method 15: X-Y plot of specified quantities at given time point along a polyline (profile along polyline)

6 Initial Conditions

Initial conditions can be specified for saturation, fluid pressure, temperature and concentration. Saturation, fluid pressure and temperature refer to phases and concentration refers to components. The available menus are controled by the selected physico-chemical model as well as by the specified number of phases and components.

6.1 Keywords

The following keywords are available to specify initial conditions for fluid flow, heat transport and mass transport.

```
#INITIAL_CONDITIONS_PRESSURE
#INITIAL_CONDITIONS_WATER_CONTENT
#INITIAL_CONDITIONS_SATURATION
#INITIAL_CONDITIONS_CONCENTRATION
#INITIAL_CONDITIONS_SOLUTE_CONCENTRATION
#INITIAL_CONDITIONS_SORBED_CONCENTRATION
#INITIAL_CONDITIONS_TEMPERATURE
#INITIAL_CONDITIONS_IMMOBILE_SOLUTE_CONCENTRATION
```

Initial conditions for different phases and components are specified by repeated use of the keyword. If the keyword is not repeated but more than one initial conditions are specified under one keyword these conditions are applied to the first phase.

6.2 Methods

Method	Subject	Au
0	Constant value to all nodes	AH
1	Constant value for individual node given by node number	AH
2	Constant value for individual nodes given by coordinates	AH
3	Linear distribution between two nodes given by node numbers	AH
4	Linear distribution between two nodes given by coordinates	AH
5	Linear depth distribution to all nodes	AH/OK
6	Hydrostatic distribution to all nodes by z-coordinate ????	AH
7	Specified distribution around node given by node number	AH
8	Specified distribution around node given by node coordinates	AH
9	Constant value at a plain given by coordinates	AH
10	Linear distribution between horizontal planes given by z-coordinates	AH
11	Bilinear distribution to all nodes within a rectangle given by node coordinates	RK
20	Constant value inside polygon, 2-D method $\Subset\Supset$ POLYLINE object	OK
21	Constant value along a polygon, 3-D method $\Subset\Supset$ POLYLINE object	OK

6.3 Examples

Method 0: A constant value is applied to all nodes.

```
RFD - File:
#INITIAL_CONDITIONS_PRESSURE ; keyword
0 0 1.e5 ; method, mode, value
#INITIAL_CONDITIONS_CONCENTRATION ; keyword
0 0 0. ; method, mode, value
```

Method 1: A constant value is applied to a node given by node number.

```
RFD - File:
#INITIAL_CONDITIONS_SATURATION ; keyword
1 0 0 0.15 ; method, mode, node, value
```

Method 2: A constant value is applied to a node given by coordinates.

```
RFD - File:
#INITIAL_CONDITIONS_PRESSURE ; keyword
2 0 ; method, mode
100 1 0 1.e5 ; x, y, z, value
```

Method 3: A linear value distribution is applied to nodes between two nodes given by node numbers. Skipmode specifies to which nodes no values are assigned:
If skipmode $= 1$: value is applied to every node on line
if skipmode $= 2$: value is applied to 1., 3., 5.....node on line etc.

```
RFD - File:
#INITIAL_CONDITIONS_PRESSURE ; keyword
3  0 ; method, mode
0  100 ; node1, node2
1  1.693567e+005  1.693567e+005 ; skipmode, value1, value2
```

Method 4: A linear value distribution is applied to all nodes between two nodes given by coordinates.

```
RFD - File:
#REFERENCE_CONDITIONS
 9.810000  0.000000  101325.000000
#INITIAL_CONDITIONS_CONCENTRATION ; keyword
 4 0 ; method, mode
 1.500000e+02  0.000000e+00  1.500000e+02 200. ; x1, y1, z1, value1
 4.500000e+02  0.000000e+00  1.500000e+02 200. ; x2, y2, z2, value2
```

Method 5: A linear depth distribution is applied to all nodes according to:
$value = reference\_value + (node\_elevation - reference\_elevation) * gradient$

```
RFD - File:
#INITIAL_CONDITIONS_PRESSURE  ; keyword
5 0 ; method, mode
150. 1.e5  -9810. ; reference_elevation, reference_value, gradient
```

Method 6: Hydrostatic distribution to all nodes by z-coordinate according to:
$value = (value1 - z1) * rho * g) - (node\_elevation - z1) * rho * g$
gekürzt: $value = rho * g * (value1 - node\_elevation)$
ist nie hydrostatisch !!!!!!!

```
RFD - File:
#INITIAL_CONDITIONS_PRESSURE  ; keyword
6 0 ; method, mode
100. 100. ; z1, value1
```

Method 7: A distribution is assigned to nodes around a node given by node coordinates according to the distribution_type chosen:
0: Cube with constant value
1: Diamond with constant value
2: Sphere with constant value
-1: Triangle or cone with value distribution $value = value1 - value1 * dist/radius$
-2: Circle or sphere with value distribution
-3: Gaussian distribution $value = value1 * exp((-0.5) * dist * dist/radius/radius)$
-4: Cylinder ???

```
RFD - File:
#INITIAL_CONDITIONS_CONCENTRATION ; keyword
7 0 ; method, mode
-1 ; distribution_type,
0 ; node
100 50 ;  radius, value1
```

Method 8: A distribution is assigned to nodes around a node given by node coordinates according to the distribution_type chosen:
0: Cube with constant value
1: Diamond with constant value
2: Sphere with constant value
-1: Triangle or cone with value distribution $value = value1 - value1 * dist/radius$
-2: Circle or sphere with value distribution
-3: Gaussian distribution $value = value1 * exp((-0.5) * dist * dist/radius/radius)$
-4: Cylinder ???

```
RFD - File:
#INITIAL_CONDITIONS_CONCENTRATION ; keyword
8 0 ; method, mode
-1 ; distribution_type
0 10 10 ; x, y, z,
2. 100. ;  radius, value1
```

Method 9: A constant value is applied to all nodes in a plain given by coordinates.

```
RFD - File:
#INITIAL_CONDITIONS_PRESSURE ; keyword
9 0 ; method, mode
0 0 0 ; x1, y1, z1
0 0 1 ; x2, y2, z2
0 1 0 ; x3, y3, z3
1.e5 ; value1
```

Method 10: A linear depth distribution is applied to all nodes in between a specified number of horizontal planes given by the z coordinates according to:
$value = value1 + |node\_elevation - z1| * (value2 - value1) \quad / \quad |z2 - z1|$

```
RFD - File:
#INITIAL_CONDITIONS_PRESSURE ; keyword
10 0 ; method, mode
2 0 100 50 0 ; number_of_plains, z1, value1, z2, value2
```

Method 11: A bilinear distribution is applied to all nodes within a rectangle given by node coordinates.

```
RFD - File:
#INITIAL_CONDITIONS_CONCENTRATION ; keyword
11 0 ; method, mode
 0 0 0 100 ; x1, y1, z1, value1
50 0 0  50 ; x2, y2, z2, value2
 0 1 0 100 ; x3, y3, z3, value3
50 1 0  50 ; x4, y4, z4, value4
```

Method 20: A constant value is assigned to all nodes within and close to a polyline (see method 21). Initial conditions data are in the RFD file, whereas POLYLINE data are in the RFM file.
(mode and distribution_type are not used).

```
RFD - File:
#INITIAL_CONDITIONS_TEMPERATURE ; keyword
 20 0 Conc1 0 340.0 ; method, mode, polyline_name, distribution_type, value
 20 0 Conc2 0 340.0 ; method, mode, polyline_name, distribution_type, value
RFM - File:
#POLYLINE
Conc1
4 ; number of polyline points
2 ; polyline type
2.566993e+005 -8.162393e+002 0.000000e+000 0.000000e+000 5.000000e+001
2.566993e+005 -2.264957e+002 0.000000e+000 0.000000e+000 5.000000e+001
2.515904e+005 -2.111111e+002 0.000000e+000 0.000000e+000 5.000000e+001
2.514379e+005 -8.111111e+002 0.000000e+000 0.000000e+000 1.000000e-001
0.000000e+000 ; epsilon
1 ; closed
#POLYLINE
Conc2
4 ; number of polyline points
2 ; polyline type
2.619608e+005 -8.213675e+002 0.000000e+000 0.000000e+000 1.000000e-001
...
```

Method 21: A constant value is assigned to all nodes close to a polyline (3-D method ????). Initial conditions data are in the RFD file, whereas POLYLINE data are in the RFM file.
(mode, distribution_type and radius are not used(epsilon in rfm)).

```
RFD - File:
#INITIAL_CONDITIONS_TEMPERATURE ; keyword
 21 0 Conc1 0 340.0 1 ; method, mode, polyline_name, distribution_type, value, radius
 21 0 Conc2 0 340.0 1 ; method, mode, polyline_name, distribution_type, value, radius
```

6.4 GUI

last modified: MB - July 25, 2003

7 Boundary Conditions

7.1 Keyword Description

The following keywords are available to specify boundary conditions for fluid flow, heat transport and mass transport.

#BOUNDARY_CONDITIONS_PRESSURE
#BOUNDARY_CONDITIONS_SATURATION
#BOUNDARY_CONDITIONS_TEMPERATURE
#BOUNDARY_CONDITIONS_CONCENTRATION
#BOUNDARY_CONDITIONS_SORBED_CONCENTRATION
#BOUNDARY_CONDITIONS_SOLUTE_CONCENTRATION
#BOUNDARY_CONDITIONS_DISPLACEMENT_X
#BOUNDARY_CONDITIONS_DISPLACEMENT_Y
#BOUNDARY_CONDITIONS_DISPLACEMENT_Z
#BOUNDARY_CONDITIONS_FREE_OUTFLOW

By repeatedly use of the keywords, boundary conditions for corresponding phases and components are specified.

Parameter	RF Variable	Values	Meaning
N1	type	0	Individual nodes by node number
		1	Individual nodes by coordinates
		2	Linear distribution between two nodes by coordinates
		3	Linear distribution between two nodes by node number
		4	Constant value at plain given by 3 points
		5	Linear distribution along polygon
		6	Constant value at plain given by polygon node numbers
		7	Hydrostatic distribution at plain given by 3 points
		8	Constant value at plain given by polygon coordinates
		9	Bilinear distribution within rectangle by coordinates
		10	Bilinear distribution at plain given by coordinates
		11/0	Individual nodes by node number (eps)
		12/5	Linear distribution along polygon / coordinates (eps)
		13	Linear distribution along polygon / nodes (eps)
		14	Hydrostatic distribution in polygon / coordinates (eps)
		15	Hydrostatic distribution in polygon / nodes (eps)
		16	Bilinear distribution within rectangle by coordinates
		17	Hydrostatic distribution in rectangle / coordinates (eps)
N2	node	0	Overwrite
		1	Superimpose
N3	curve	int	Curve number
if N1=0	begin_node		Node number
	values[0]		Nodal value (AH)
if N1=1	x[0] y[0] z[0]		Node coordinates
	radius[0]		Radius (0.0)
	values[0]		Nodal value (AH)
if N1=2	x[0] y[0] z[0]		First node coordinates
	x[1] y[1] z[1]		First nodal value
	radius[0]		"Radius", epsilon
	values[0]		Second node coordinates
	values[1]		Second nodal value RFE-Data incorrect (AH)

Continuation from the previous page

if N1=3	begin_node end_node step_nodes values[0] value[1]		First node Second node Increment First nodal value Second nodal value (RFE-Data has to be adjusted) (AH)
if N1=4	x[0] y[0] z[0] x[1] y[1] z[1] x[2] y[2] z[2] radius[0] values[0]		First node coordinates Second node coordinates Third node coordinates "Radius", epsilon Areal value (AH)
if N1=5	count_of_points nodes[count_of_points] values[count_of_points]		Number of polygon points Polygon nodes Nodal values (AH/CT)
if N1=6	count_of_points nodes[count_of_points] radius [0] value[0]		Number of surface polygon points Surface polygon nodes "Radius", epsilon (does not work properly!) Areal value (RFE-Data not correct) (AH)
if N1=7	x[0] y[0] z[0] x[1] y[1] z[1] x[2] y[2] z[2] z[3] values[0]		First node coordinates Second node coordinates Third node coordinates Reference elevation Reference value (not yet tested) (AH)
if N1=8	count_of_points x[0] y[0] z[0] ... x[count_of_points-1] y[count_of_points-1] z[count_of_points-1] radius[0] values[0]		Number of polygon points Polygon node coordinates ... "Radius", epsilon (does not work properly!) Value (RFE-Data not correct) (AH)
if N1=9	x[0] y[0] z[0] values[0] x[1] y[1] z[1] values[1] x[2] y[2] z[2] values[2] x[3] y[3] z[3] values[3]		Rectangle node coordinates Value ... (does not work properly!) (RK)
if N1=10	x[0] y[0] z[0] values[0] x[1] y[1] z[1] values[1] x[2] y[2] z[2] values[2] x[3] y[3] z[3] values[3]		Node coordinates Value ... (does not work properly!) (RK)
if N1=11	x[0] y[0] z[0] values[0] epsilon		 (CT)

Continuation from the previous page

if N1=12	count_of_points x[count_of_points] y[count_of_points] z[count_of_points] values[count_of_points] epsilon		(CT)
if N1=13	count_of_points nodes[count_of_points] values[count_of_points] epsilon		(CT)
if N1=14	values[0] values[1] phase count_of_points x[count_of_points] y[count_of_points] z[count_of_points] epsilon		(not yet tested) (CT)
if N1=15	values[0] values[1] phase count_of_points nodes[count_of_points] epsilon		(not yet tested) (CT)
if N1=16	x[0] y[0] z[0] values[0] x[1] y[1] z[1] values[1] x[2] y[2] z[2] values[2] x[3] y[3] z[3] values[3] epsilon		(CT)
if N1=17	values[0] values[1] x[0] y[0] z[0] x[1] y[1] z[1] x[2] y[2] z[2] x[3] y[3] z[3] epsilon		(not yet tested) (CT)

Specific boundary conditions

#BOUNDARY_CONDITIONS_FREE_OUTFLOW

This keyword is intended for open boundaries under multiphase flow conditions.

For a system with two or more phases it is necessary to specify BCs for the saturations on each boundary which is regarded "open" by specifying a pressure BC. If the user knows in advance that this will be a free outflow boundary, the free outflow BC can replace the saturation BCs and the saturations at the boundary will adjust freely in the simulation due to the conditions inside the system.

If the simplified Richards solution is used, this BC specifies the free outflow of fluid over a boundary with no pressure BC specified. In the iterative process the pressure will be set to zero (which is the reference pressure for the transition between full and partially saturated conditions) if thereby an outflow is enabled, otherwise the boundary is regarded as impermeable.

last modified: MK June 13, 2003

8 Source Terms

In terms of the general balance equation for a quantity ψ, source terms Q^{ψ} are defined in the following way (Kolditz 2002, Chapter 1.2.3).

$$\frac{d\psi}{dt} = \frac{\partial \psi}{\partial t} + \nabla \cdot \mathbf{\Phi}^{\psi} = Q^{\psi} \tag{1}$$

Subject	Symbol	Unit
Fluid phase mass source term	$Q^{\rho^{\gamma}}$	kgs^{-1}
Fluid phase volume source term	$1/\rho^{\gamma} Q^{\rho^{\gamma}}$	$m^3 s^{-1}$
Heat source term	Q^{h}	Ws^{-1}
Mass source term (tracer, soluted, sorbed)	Q^{C_k}	kgs^{-1}
Momentum source term (load, force)	Q^{m}	N
Fluid mixture mass source term	$\sum_{\gamma} Q^{\rho^{\gamma}}$	kgs^{-1}
Fluid mixture volume source term	$\sum_{\gamma} 1/\rho^{\gamma} Q^{\rho^{\gamma}}$	$m^3 s^{-1}$

with: γ - fluid phase, k - component.

8.1 Keywords

The following keywords are available to specify source-sink terms for fluid flow, heat and mass transport. Object names are defined for object identification.

```
#SOURCE_MASS_FLUID_PHASE
#SOURCE_VOLUME_FLUID_PHASE
#SOURCE_HEAT_PHASE
#SOURCE_MASS_TRACER_COMPONENT
#SOURCE_MASS_SOLUTE_COMPONENT
#SOURCE_MASS_SORBED_COMPONENT
#LOAD_SOLID_PHASE_X
#LOAD_SOLID_PHASE_Y
#LOAD_SOLID_PHASE_Z
#SINK_MASS_FLUID_MIXTURE
#SINK_VOLUME_FLUID_MIXTURE
```

For multiphase or multicomponent computations the keywords

```
#SOURCE_MASS_FLUID_PHASE
#SOURCE_VOLUME_FLUID_PHASE
#SOURCE_MASS_TRACER_COMPONENT
```

can be repeated, according to the phase/component they shall apply to. For multiphase flow problems the resulting flux of the phases can differ from the specified ones, especially if the grid resolution around the source is insufficient. This means that "single node" injections must be avoided in multiphase flow computations.

For multiphase-multicomponent computations, the keyword

```
#SOURCE_MASS_TRACER_COMPONENT
```

can be specified (number of phases * number of components) times. Then, the first keyword applies to the first component in the first phase, the second one to the second component in the first phase, until number of components is reached. Afterwards, the same cycle starts for the second phase. This is repeated until number of phases is reached.

In multiphase flow computations the

```
#SOURCE_...FLUID_PHASE
```

keywords must not have negative values, as it is not allowed to extract a single phase from the system. Thus, the keywords

```
#SINK_..._FLUID_MIXTURE
```

can be used to extract all fluids simultaneously at the given composition.

In model 10699 (compare #MODELS) the keyword

```
#SOURCE_MASS_TRACER_COMPONENT
```

has a special meaning. If fluid fluxes into the system occur in the area for which the keyword is specified, the mass balance will be adjusted so that the tracer mass fluxes by the fluids equals the one that is specified by this keyword. I.e. it is possible to simulate the process of thinning out (e.g. desalinization) by specifying a low tracer mass flow on a boundary where fluid influxes take place.

8.2 Methods

Method	Subject	Au
0	Nodal source given by node number	AH
1	Nodal source given by coordinates	AH
2	Nodal sources along line defined by two nodes (coordinates)	AH
3	Flux distributed along line defined by two nodes (coordinates)	AH
4	Nodal sources at plain surface given by 3 nodes (coordinates)	AH
5		
6	Nodal sources at a polygon given by node coordinates (eps)	CT
7	Nodal sources at a polygon given by node numbers (eps)	CT
8	Nodal source given by node coordinates (eps)	CT
9	Distributed nodal source given by coordinates (eps)	CT
10	Transfer function by nodal sources	OK
11	Source along polygon / divided by number of nodes	CT
12	Source along polygon $\in\ni$ POLYLINE object	OK

8.3 Examples

Method 12: Source along polygon linked to a POLYLINE object (value is divided by number of nodes). Source data are in the RFD file, whereas POLYLINE data are in the RFM file.

```
RFD - File:
#SOURCE_HEAT_PHASE ; keyword
 12 0 1 ; method, mode, multiplier curve number
```

```
  HEAT_SOURCES_1 ; name of POLYLINE
  1. ; epsilon range
  1.2345e-3 ; value

RFM - File:
#POLYLINE ; keyword
 HEAT_SOURCES_1 ; name of POLYLINE
 4         ; number of polyline nodes
  ;           x             y             z         value
  7.222222e-001 3.083144e+000 1.000000e+000 1.000000e+000
  6.818783e+000 3.559226e+000 2.000000e+000 2.000000e+000
  8.783069e+000 8.570615e+000 1.000000e+000 4.000000e+000
  4.534392e+000 8.921412e+000 3.000000e+000 3.000000e+000
...
```

8.4 GUI

last modified: OK - July 25, 2003

9 Fluid Properties

9.1 Keywords

The following keywords have been introduced for fluid properties:

#FLUID_PROPERTIES_NEW	Announces fluid properties input, indicates that the formulation including keywords is used. Generally, if more than one phase is described, first the gas phase is described, then the liquid phase.
$DENSITY	Starts the input for the density case for the phase density. The first number is the density case that will be used, followed by the parameters needed for that model.
$VISCOSITY	Starts the input for the viscosity case. The first number is the density case that will be used, followed by the parameters needed for that model.
$HEAT_CAPACITY	The first number, 0, stands for constant model, the second is the heat capacity of the fluid.
$HEAT_CONDUCTIVITY	The first number, 0, stands for constant model, the second is the heat conductivity of the fluid.

To illustrate this, two input examples are shown below. The first input example is an example of the old input, without sub-keywords. The second example is an example with the new keywords. All new files should be written with the keywords, as some model additions require them. A description of the individual phase density and viscosity cases can be found in the following sections.

Input example 1: No sub-keywords

```
#FLUID_PROPERTIES ; gas
 0  1.15 ; density function, parameter
 0  1.800000e-005 ; viscosity function, parameter
 3.000000e+003  1.000000e+000 ; heat capacity, heat conductivity
#FLUID_PROPERTIES ; liquid
 0  1.050000e+003 ; density function, parameter
 0  0.0012 ; viscosity function, parameter
 0.000000 ; real gas factor
 4.000000e+003  2.000000e+000 ; heat capacity, heat conductivity
```

Input example 2: With sub-keywords

```
#FLUID_PROPERTIES_NEW
$DENSITY
 0   1000.                      ; rho_0: reference density,
; 1   1000.  1e-4               ; rho_0,drho/dp:compressibility
; 2 3 10 1000.  0.2             ; rho_0,drho/dC:expansion coeff.
; 11 1000.  0.2  1e-4           ; rho_0,drho/dC,drho/dT:therm. expan. coeff.
; 12 1000.  1e-4                ; rho_0,drho/dT
; 13                            ; no parameters needed
; 14 1000. 10100 1e-4 285 1e-4  ; rho_0, p_0, drho_/dp, T_0, drho/dT
$VISCOSITY
 0   1e-3            ; my_0: reference viscosity
$HEAT_CAPACITY
 0   4800.           ; c_0: reference heat capacity
$HEAT_CONDUCTIVITY
 0   0.6             ; lamda_0: reference heat conductivity
```

The following sections describe the choice of models for the keywords $DENSITY and $VISCOSITY.

9.2 Fluid phase density - $DENSITY

Nomenclature

C	concentration
M_a	molar mass of air
M_w	molar mass of water
p	pressure
$p^g_{w,sat}$	saturated vapor pressure
R	ideal gas constant
T	temperature
β_p	compressibility
β_T	thermal expansion coefficient
γ	phase
ρ	density
subscript 0	reference value

Cases

It is important to state that there are two functions calculating density, `CalcFluidDensity` and `MATCalcFluidDensity`. The function `MATCalcFluidDensity` does not have a variable parameter list. There are more density computation choices available through this function. Up to now, only models 100102 (TH non-isotherm) and 10333 (unconfined flow) use this function.

Table 1 shows the available fluid density calculation methods. It is important to state that for multiphase applications, it is assumed that phase numbering starts from 0, and that phase 0 is a gaseous phase. Single phase calculations are not influenced by this. Case 0 is the constant density case. Case 1 computes density as a function of pressure. Cases 2, 3, and 10 compute density as a function of concentration. In case 3, more than one component can be present and case 10 uses a reference concentration and a concentration expansion coefficient. Case 11 computes density as a function of concentration and temperature.Case 12 computes density as a function of temperature.

Case 13 computes density as a function of pressure and temperature. Case 13 is limited to calculations of the gas phase density. Additionally, one has to take care that the difference between gas pressure and saturated vapour pressure is positive.

Table 1: Fluid density calculation methods
⊙ : Cases only available with model 100102 and 10333

Case	Function
0	$\rho^\gamma = \rho_0^\gamma$
1	$\rho^\gamma(p) = \rho_0^\gamma + \beta_p^\gamma p^\gamma$
2	$\rho^\gamma(C) = \rho_0^\gamma + max(C)$
3	$\rho^\gamma(C_i) = \rho_0^\gamma + max(\rho_i^\gamma(C_i), \rho_0^\gamma$
10	$\rho^\gamma(C) = \rho_0^\gamma + \frac{\partial \rho^\gamma}{\partial C}(C - C_0)$
11	$\rho^\gamma(C,T) = \rho_0^\gamma + max(C,0)\beta_p + max(T,0)\beta_T$
⊙ 12	$\rho^\gamma(T) = \rho_0^\gamma + \frac{\partial \rho}{\partial T}(T - T_0)$
⊙ 13	$\rho^g(p^g,T) = \frac{M_a}{RT}p^g + \frac{(M_w - M_a)}{RT}p^g_{w,sat}(T)$
⊙ 14	$\rho^\gamma(p,T) = \rho_0^\gamma + \frac{\partial \rho}{\partial T}(T - T_0) + \frac{\partial \rho}{\partial p}(p - p_0)$

Data Input

The table below summarizes the input for the keyword $DENSITY. One of the cases with corresponding values can be chosen.

Table 2: Parameters for fluid density calculations

Case	Parameters		
0	ρ_0 $[kg/m^3]$		
1	ρ_0 $[kg/m^3]$	$\beta_p[1/Pa]$	
2	ρ_0 $[kg/m^3]$		
3	ρ_0 $[kg/m^3]$	$\partial\rho/\partial C[-]$	
10	ρ_0 $[kg/m^3]$	$\partial\rho/\partial C[-]$	
11	$\rho_0[kg/m^3]$	$\beta_p[1/Pa]$	$\partial\rho/\partial C[1/mol]$
12	ρ_0 $[kg/m^3]$	$\beta_T[1/K]$	
13			

Data access

The implementation of fluid density functions into the Rockflow code is as follows.

Description	Function	RF Object
Access to fluid density in kernel	rho=MATCalcFluidDensity()	MPC/ENT
Calculation of density	`MATCalcFluidDensity`	MAT
Overwriting	`MATGetNodeIndexTemperature=` `MODGetNodeIndexTemperature_XX`	MOD
Definition of model function for access to node index	`MODGetNodeIndexTemperature_XX`	MOD

Table 3: Fluid density functions in Rockflow

9.3 Fluid Viscosity

Nomenclature

C	concentration
p	pressure
T	temperature
γ	phase
μ	viscosity
subscript 0	reference value
subscript avg	average value

Cases

Fluid viscosity can be computed in one of the ways illustrated in Table 4. As for density, it is important to note that viscosity can be calculated with two different functions in the code, `CalcFluidViscosity` and `MATCalcFluidViscosity`. `MATCalcFluidViscosity` is only accessed by models 100102 and 10333 so far. Case 0 is the constant viscosity case. Cases 1 and 2 compute viscosity as a function of pressure. Case 1 uses a curve, whereas case 2 uses a coefficient for the variation of viscosity with pressure. Case 7 computes the viscosity after Reichenberg (1971) as shown in Reid et al. (1988), p. 420 as a function of pressure and temperature. Equation 2 complete the equation shown in Table 4 for this case.

Additional Equations for Case 7

$$
\begin{aligned}
\mu_0 &= 26.69\sqrt{28.96}\frac{\sqrt{T}}{3.7^2}1.010^{-6}0.1 \\
A &= \frac{1.982410^{-3}}{\frac{T}{126.2}}\exp(5.2683(\frac{T}{126.2})^{-0.5767}) \\
B &= A(1.6552\frac{T}{126.2}-1.2760) \\
C &= \frac{0.1319}{\frac{T}{126.2}}\exp(3.7035(\frac{T}{126.2})^{-79.8678}) \\
D &= \frac{2.9496}{\frac{T}{126.2}}\exp(2.9190(\frac{T}{126.2})^{-16.6169})
\end{aligned}
\tag{2}
$$

Case 8 computes viscosity as a function of temperature following Yaws et al. (1976) as shown in Reid et al. (1988), p. 441/455. Case 9 also computes viscosity as a function of temperature following a logarithmical distribution. Case 10 computes viscosity as a function of concentration and temperature.

Table 4: Fluid viscosity calculation methods
⊙ : Cases only available with model 100102 and 10333

Case	Function
0	$\mu^\gamma(p) = \mu_0^\gamma$
1	$\mu^\gamma(p) = GetCurveValue(\text{get\_fp\_curve(fp)}, 0, p_{\text{avg}}, gueltig)$
2	$\mu^\gamma(p) = \mu_0^\gamma + p_{avg} \cdot \frac{d\mu}{dp}$
⊙ 7	$\mu^\gamma(p,T) = \mu_0[1 + \frac{A(\frac{p}{33.910^4})^{1.5}}{B(\frac{p}{33.910^4}) + \frac{1}{C(\frac{p}{33.910^4})^D}}]$
⊙ 8	$\mu^\gamma(T) = 10^{-3}\exp(-2.47110^1 + \frac{4.20910^3}{T} + 4.52710^{-2}T - 3.37610^{-5}T^2)$
⊙ 9	$\mu^\gamma(T) = 2.285 \cdot 10^{-5} + 1.01 \cdot 10^{-3}\log T$
10	$\mu^\gamma(C,T) = \frac{\mu}{f1+f2}\ f1 = f(C), f2 = f(T)$

Data Input

Case	Parameters		
0	$\mu_0\ [Pa\,s]$		
1	$\mu_0\ [Pa\,s]$	curve number	
2	$\mu_0\ [Pa\,s]$	$\partial\mu/\partial p\ [Pa^{-1}s^{-1}]$	
10	$\mu_0\ [Pa\,s]$	$\partial\mu/\partial C\ [1/mol]$	$\partial\mu/\partial p\ [Pa^{-1}s^{-1}]$

Table 5: Parameters for viscosity calculations

Data Access

To obtain fluid viscosity results, the structure shown in Table 6 is implemented in the program.

Description	Function	RF Object
FE-Kernel needs fluid viscosity for finite element matrix calculation	`GetFluidViscosity`	MPL
Calculation of viscosity using one of the cases as described above	`CalcFluidViscosity`	MAT
Specification of model specific parameters for fluid viscosity calculation	`GetFluidViscosity = THMGetFluidViscosity`	MOD

Table 6: Fluid viscosity functions in Rockflow

10 Component Properties

10.1 Keywords

The following keywords have been introduced for component properties:

#COMPONENT_PROPERTIES_NEW	announces component (tracer) properties input, indicates that the formulation including keywords is used. Generally, if more than one component is described, the components will be numbered subsequently.
$DIFFUSION	Starts the input for the diffusion parameters for the component. The first number represents the diffusion model that will be used, followed by the parameters needed for that model.
$DECAY_AQUEOUS	Starts the input for the decay in solution case. The first number is the decay model that will be used, followed by the parameters needed for that model.
$DECAY_SORPTIVE	Starts the input for the decay in sorbed state case. The first number is the decay model that will be used, followed by the parameters needed for that model.
$ISOTHERM	Starts the input for the isotherm case. The first number is the isotherm model that will be used, followed by the parameters needed for that model.
$NONEQUILIBRIUM_CHEMICAL	Starts the input for the nonequilibrium chemical reaction. The first number is the model type that will be used, followed by the parameters needed for that model.
$NONEQUILIBRIUM_PHYSICAL	Starts the input for the physical nonequilibrium model. The first number is the model type that will be used, followed by the parameters needed for that model.
$SOLUTION_PRECIPITATION	Starts the input for the dissolution - precipitation model. The first number is the model type that will be used, followed by the parameters needed for that model.

To illustrate this, the data concept and a input example are shown in the following section. All new files should be written with the keywords mentioned above, as some model additions require them. A description of the individual models can be found in the following sections.

10.2 Data Concept

10.2.1 Data Object - MAT-CP

```
typedef struct
  {                             /* fuer Tracereigenschaften */
    long type;                  /* Item-Typ */
    char *name;                 /* Gruppenname */

    /* Transport */
    double molecular_diffusion; /* Molekulardiffusion */
    /* Chemical reactions */
    /* Decay */
    double decay_rate;          /* Zerfallsrate */
    double soluted_decay_rate;  /* Zerfallsrate in geloester phase */
    double sorbed_decay_rate;   /* Zerfallsrate in sorbierter phase */

    /* Diffusionsmodelle und zugehoerige Beschreibungswerte */
    int diffusion_model;        /* Zerfallsmodell in geloester phase */
    int count_of_diffusion_model_values;        /* Anzahl und Werte zur Spezifikation der */
    double *diffusion_model_values;     /* Parameter fuer das Diffusionsprozess */

    /* Zerfallsmodelle und zugehoerige Beschreibungswerte in der geloesten Phase */
    int soluted_decay_model;    /* Zerfallsmodell in geloester phase */
    int count_of_soluted_decay_model_values;    /* Anzahl und Werte zur Spezifikation der */
    double *soluted_decay_model_values;         /* Parameter fuer das in der Loesungsphase
                                                   Zerfallsprozess wie z.B. Zerfallsrate */
    /* Zerfallsmodelle und zugehoerige Beschreibungswerte in der sorbierten Phase */
    int sorbed_decay_model;     /* Zerfallsmodell in sorbierter phase */
    int count_of_sorbed_decay_model_values;     /* Anzahl und Werte zur Spezifikation der */
    double *sorbed_decay_model_values;  /* Parameter fuer das in der geloesten Phase
                                           Zerfallsprozess wie z.B. Zerfallsrate */

    /* Sorption */
    int isotherm_model;         /* Isothermen-Typ */
    int count_of_isotherm_model_values;         /* Anzahl der Isothermen-Koeffizienten */
    double *isotherm_model_values;      /* Isothermen-Koeffizienten */

    double partitioning_coefficient;    /* K_D-Wert */
    double sorption_k1;         /* Sorptionskoeffizient k1 */
    double sorption_k2;         /* Sorptionskoeffizient k2 */

    /* Nichtgleichgewichtskoefffizient */ /* alt */
    double nonequilibrium_coefficient;  /* Nichtgleichgewichtskoefffizient(Rate) */

    /* Chemische Nichtsgleichgewichtsmodelle und zugehoerige Beschreibungswerte  */
    int chemical_nonequilibrium_model;  /* Chemisches Nichtsgleichgewichtsmodell */
    int count_of_chemical_nonequilibrium_model_values;  /* Anzahl und Werte zur Spezifikation der */
    double *chemical_nonequilibrium_model_values;       /* Parameter fuer das chemische
                                                           Nichtsgleichgewichtsmodell */

    /* Physikalische Nichtsgleichgewichtsmodelle und zugehoerige Beschreibungswerte  */
    int physical_nonequilibrium_model;  /* Physikalisches Nichtsgleichgewichtsmodell */
    int count_of_physical_nonequilibrium_model_values;  /* Anzahl und Werte zur Spezifikation der */
    double *physical_nonequilibrium_model_values;       /* Parameter fuer das physikalische
                                                           Nichtsgleichgewichtsmodell */

    /* Solution */
    int solubility_model;       /* Mathematisches Loesungsmodell */
    int solubility_dependence_model;
    int count_of_solubility_model_field_values;         /* Anzahl und Werte zur Spezifikation des */
    double *solubility_model_field_values;      /* Arbeitsbereiches fuer das Loesungsmodell */
    double solubility;          /* Salzloeslichkeit (Saettigung) */
    double dissolution_rate;    /* Loesungsgeschwindigkeit */
  } TRACER_PROPERTIES;
```

10.2.2 Data Input - MAT-CP

One Example of the Input Data for COMPONENT_Properties (From the examples in Benchmarks like: tce1d.rfd, decay1d.rfd, noneq1d.rfd, pnm1d.rfd and saltsol.rfd)

```
#COMPONENT_PROPERTIES
$DIFFUSION
1 1.157407407407e-6 ; Diffusionsmodell und Molekulare Diffusionkonstante
                    %/(See section: diffusion_model)
$DECAY_AQUEOUS
1 0.000008125  1.0  ; Zerfallsmodell und Zerfallsrate in geloester Phase
                       f*lambda=0.3*2.708333333333e-4=0.000008125
                       %/((See section:  Aqueous_Decay_Model)
$DECAY_SORPTIVE
1 7.32e-11  1.0    ; decay model in sorbed phase, lambda_s
                       %/((See section: Sorptive_Decay_Model)
$ISOTHERM
1  6.800000e-004    ; isotherm-type, k1
                       %/((See section: Isotherm_Model)
$NONEQUILIBRIUM_CHEMICAL
1  1.000000e-007    ; chemical nonequilibrium model, alpha 0
                       %/(\(See section: Chemical_Nonequilibrium_Model)
$NONEQUILIBRIUM_PHYSICAL
1  1.e-8           ; physical nonequilibrium model
                       %/((See section: Physical_Nonequilibrium_Model)
$SOLUTION_PRECIPITATION
1 1 0.2679212 1.e-5 ; solubility model
                       %/((See section: Solution_Precipitation_Model)
```

10.3 Diffusion model

Diffusion Model					
N1	n	Description	Parameter	Equation	Remarks
0	0	without Diffusion		(3)	
1	1	constant value of diffusion	c	(4)	$[1], P_{33}$
2	1	variable value of diffusion	η, m	(5)	$[1], P_{33}$
3	1	Worch, 1993		(6)	$[1], P_{33}$
4	1	Hayduk u. Laudie, 1974		(7)	$[1], P_{33}$
5	2	Wilke and Chang, 1995		(8)	$[1], P_{33}$
6	1	Stokes-Einstein, 1993		(9)	$[1], P_{33}$
7	2	FSG-Method, Lyman et al., 1990		(10)	$[1], P_{33}$
N2	Diffusion rate $[m^2/s]$				

$$D_m = 0 \tag{3}$$

$$D_m = c \tag{4}$$

$$D_m = D_m(t) \tag{5}$$

$$D_{aq} = \frac{3.595 * 10^{-7} * T}{\eta * m^{0.53}} \tag{6}$$

$$D_{aq} = \frac{13.26 \times 10^{-5}}{\eta \times V^{0.589}} \tag{7}$$

$$D_{aq} = \frac{7.4 * 10^{-8} * T * [\sqrt{X * m_{sol}}]}{\eta * V'^{0.589}} \tag{8}$$

$$D_{aq} = \frac{K_B * T}{6 * \pi * r_m * \eta} \tag{9}$$

$$D_g = \frac{0.001 * T^{1.75} * [\frac{1}{m_g} + \frac{1}{m}]^{0.5}}{P * [V_g^{\frac{1}{3}} + V^{\frac{1}{3}}]} \tag{10}$$

Input example (tce1d.rfd)

```
#COMPONENT_PROPERTIES
 $DIFFUSION
 1 1.1574e-6 ; diffusion model type, diffusion coefficient [m^2/s]
```

Data

10.4 Aqueous Decay model

Aqueous Decay Model				
N3	Description	Parameter	Equation	Remarks
0	without Decay	-		
1	First order decay with constant decay rate	λ	(11)	$[1], P_{18}$
2	First order decay with variable decay rate	λ_1	(12)	$[1], P_{18}$
3	n-th order decay with constant power kinetics	λ_1,λ_2	(13)	$[1], P_{18}$
4	n-th order decay with variable power kinetics	λ_1,λ_2	(13)	$[1], P_{18}$
5	Monod or Michaelis-Menten Kinetics with constant coefficient		(14)	$[1], P_{24}$
6	Monod or Michaelis-Menten Kinetics with variable coefficient		(14)	$[1], P_{24}$
N4	Aqueous decay rate $[1/s]$			
N5	?			

$$\phi = -\lambda * C \tag{11}$$

$$\phi = -\lambda_1 * C \tag{12}$$

$$\phi = -\lambda_1 * C^{\lambda_2} \tag{13}$$

$$\phi = \frac{\mu_{max} * C}{K_s + C} \tag{14}$$

Input example (tce1d.rfd)

```
#COMPONENT_PROPERTIES
 $DECAY_AQUEOUS
 1 2.847e-5 1.0 ; decay model type,
                  decay rate in solution [1/s], ?order of decay
```

10.5 Sorptive Decay model

Sorptive Decay Model					
N6	n	Description	Parameter	Equation	Remarks
0	0	without Decay			
1	2	First order decay with constant decay rate		(15)	$[1], P_{24}$
2	3	First order decay with variable decay rate		(15)	$[1], P_{24}$
3	3	Second order decay with constant power kinetics		(16)	$[1], P_{24}$
4	4	Second order decay with variable power kinetics		(16)	$[1], P_{24}$
5	3	Monod or Michaelis-Menten Kinetics with constant coefficient		(17)	$[1], P_{24}$
6	4	Monod or Michaelis-Menten Kinetics with variable coefficient		(17)	$[1], P_{24}$
7	3	First order decay with constant decay rate and constant fraction of parent that form the daughter product		?	
N7	Sorptive decay rate $[1/s]$				
N8	?				

$$\phi = -\lambda 1 * C \tag{15}$$

$$\phi = -\lambda 1 * C^{\lambda 2} \tag{16}$$

$$\phi = \frac{\mu_{max} * C}{K_s + C} \tag{17}$$

Input example (decay1d.rfd)

```
#COMPONENT_PROPERTIES
 $DECAY_SORPTIVE
  1 7.32e-11  1.0     ; decay model type in sorbed phase,
                       lambda_s, ?order of decay
```

10.6 Isotherm model

Isotherm Model				
N9	Description	Parameter	Equation	Remarks
0	No Isotherm			
1	Henry Isotherm		(18)	$[1], P_{19,43}$
2	Freundlich Isotherm		(19)	$[1], P_{43}$
3	Langmuir Isotherm		(20)	$[1], P_{43}$
4	Freundlich Langmuir Isotherm		(21)	$[1], P_{19,43}$
5	Double Langmuir Isotherm		?	
6	Extended Freundlich Isotherm		?	
7	Gunary Isotherm		(22)	$[1], P_{19}$
8	No Diffusion		?	
9	Fitter-Sutton Isotherm		?	
10	Power Isotherm		?	
11	Modified Kielland Isotherm		?	
N10	Coefficient of distribution			

$$S = k_d * C \tag{18}$$

$$S = k_1 * C^{k_2} \tag{19}$$

$$S = \frac{k_1 * C}{1 + k_2 * C} \tag{20}$$

$$S = \frac{k_1 * C^{k_3}}{1 + k_2 * C^{k_3}} \tag{21}$$

$$S = \frac{k_1 * C}{1 + k_2 * C^{k_3} + k_3 * \sqrt{C}} \tag{22}$$

Input example (decay1d.rfd)

```
#COMPONENT_PROPERTIES
 $ISOTHERM
 1 6.8e-5      ; isotherm model type, kd
```

10.7 Chemical Nonequilibrium Model

Chemical Nonequilibrium Model				
N11	Description	Parameter	Eqn	Remarks
0	equilibrium model or no nonequilibrium model			
1	First order reaction with constant transfer coefficient	k1	(23)	$[1], P_{31}$
2	n-th order reaction	k1, k2	?(24)	$[1], P_{31}$
3	Reaction with langmuir kinetic		(25)	
4	Non reversible reaction		(26)	
5	power kinetic		(27)	
6	First order reaction with constant transfer coefficient (2 parameters)		(28)	$[1], P_{31}$
7	n-th order reaction with constant transfer coefficient (2 parameters)		(29)	$[1], P_{31}$
N12	Coefficient of chemical nonequilibrium reaction α			

$$\frac{\partial S}{\partial t} = k_1 * (f(C) - S) \tag{23}$$

$$\frac{\partial S}{\partial t} = k_1 * \frac{\theta}{\rho_b} * C - k_2 * S \tag{24}$$

$$\frac{\partial S}{\partial t} = k_1 * \frac{\theta}{\rho_b} * C^n - k_2 * S \tag{25}$$

$$\frac{\partial S}{\partial t} = k_s * \frac{\theta}{\rho_b} * (C - C_p) \tag{26}$$

$$\frac{\partial S}{\partial t} = k_1 * \frac{\theta}{\rho_b} * C(S_{max} - S) - k_2 * S \tag{27}$$

$$\frac{\partial S}{\partial t} = A * exp(-BS) \tag{28}$$

$$\frac{\partial S}{\partial t} = k_1 * \frac{\theta}{\rho_b} * C^n S^m \tag{29}$$

Input example (noneq1d.rfd)

```
#COMPONENT_PROPERTIES
 $NONEQUILIBRIUM_CHEMICAL
 1 1.00e-7  ;  chemical nonequilibrium model type, k1 [?-]
```

10.8 Physical Nonequilibrium Model

Physical Nonequilibrium Model				
N13	Description	Parameter	Eqn	Remarks
0	Model type (Non- or equilibrium)			
1	First order reaction with constant transfer coefficient (Coats and Smith 1964)		?(30)	$[1], P_{38-39}$
2	First order reaction with variable transfer coefficient (Coats and Smith 1964)		?(30)	$[1], P_{38-39}$
N14	Coefficient of chemical nonequilibrium reaction α			

$$\theta_{im} * R_{im} \frac{\partial C_{im}}{\partial t} = \alpha * (C_m - C_{im}) \tag{30}$$

Input examples (pnm1d.rfd)

```
#COMPONENT_PROPERTIES
 $NONEQUILIBRIUM_PHYSICAL
 1 1.00e-8  ; physical nonequilibrium model type, alpha [?-]

#COMPONENT_PROPERTIES
 $NONEQUILIBRIUM_PHYSICAL
 2 1.00e-8 1.0; physical nonequilibrium model type, alpha [?-]
```

10.9 Solution-Precipitation Model

Solution-Precipitation Model					
N11	n	Description	Parameter	Eqn	Remarks
0		No dissolution and precipitation		-	
1		With dissolution			[2]
	0	Constant solubility in case of saturation		(31)	
	1	Dependence on Pressure (P in $[Pa]$)		(32)	
	2	Dependence on temperature (T in $[^\circ C]$)		(32)	
	3	Dependence on temperature and pressure		(32)	
N12	Coefficient of dissolution ≥ 0				

$$solubility = const. \tag{31}$$

$$solubility\_coefficient = 0.26291 + 0.7448 \cdot 10^{-4} T + 0.1252 \cdot 10^{-5} T^2 + 7.5 \cdot 10^{-11} P \tag{32}$$

Input example

```
#COMPONENT_PROPERTIES
 $SOLUTION_PRECIPITATION
 1 1 0.26292 1.0e-5  ; model type, solution dependence type,
                       solubility [kg/m^3], dissolution_rate[m/s]
```

10.10 References Specific for this Chapter

Bear J & Bachmat Y (1990): *Introduction to modeling of transport phenomena in porous media*, Kluwer Academic Publishers.

Brooks R N & Corey A T (1964): *Properties of porous media affecting fluid flow* J Irrig Drain Div ASCE, 92 (IR2): 61-8.

Forchheimer, P. (1914): *Hydraulik*, Teubner Verlag, Berlin-Leipzig.

Haverkamp R, Vauclin M, Touma J, Wierenga P J & Vachaud G (1977): *A comparison of numerical simulation models for one-dimensional infiltration*, J Soil Science Society of America, 41: 284-94.

Segol G (1995): *Classic groundwater simulations*, PTR Prentice Hall.

van Genuchten M (1980): *A closed-form equation for predicting the hydraulic conductivity of unsaturated soils*, J Soil Science Society of America, 44: 892-98.

11 Solid Properties

11.1 Keywords

Solid properties can be specified using the keyword

#SOLID_PROPERTIES

This keyword is mandatory for the use of the stand-alone deformation model (10) and other coupled deformation models and processes. It may be used repeatedly, then several species are defined.

Parameter	RF Variable	Values	SI-Unit	Meaning
N1	material_model			**Linear elastic models**
		1		Linear elasticity model
				Elasto-plastic models
		(100)		J^2 von Mises plasticity model
		(101)		Drucker-Prager model
		(200)		**Visco-elastic models**
if N1=1				Linear elasticity model
	e_modulus	E	Pa	Elasticity modulus
	poissons_ratio	ν	-	Poisson's Ratio
if N1=101				J^2 von Mises plasticy model
	e_modulus	E	Pa	Elasticity modulus
	poissons_ratio	ν	-	Poisson's Ratio
	yield_stress	σ_0	Pa	Yield stress

() - in preparation

Example

```
#SOLID_PROPERTIES
 1           ; Linear elasticity model
 2000.0      ; Modulus of elasticity
 0.3         ; Poisson's ratio
```

11.2 Sub-keywords

Auxiliary sub-keywords are listed in the following table.

Sub-keyword / RF Variable	Value	SI-Unit	Compatible Models / Meaning
$THERMAL_EXPANSION_COEFFICIENT thermal_expansion_coefficient	β_T	1/K	[1] Thermal expansion coeff.[1]

[1)] The reference temperature $T_{\mathrm{ref}} = T_0$ is defined using #REFERENCE_CONDITIONS

last modified: MK June 25, 2003

The following keywords govern the input of the extra material parameters of solid properties.

$ DRUCKER-PRAGER	Following this keyword, four parameters for Drucker-Prager plastic model have to be input. They are Y_0 The initial yield stress H Plastic hardening modulus ϕ Internal frictional angle ϕ dilatancy angle This keyword has to be used in the case that involves plasticity with Drucker-Prager model
$ EHLERS_WEIMAR	This keyword switches the input of 20 parameters of a new plastic model derived from Ehlers' model. Following this keyword, the 20 data are in the order: α_0, β_0, δ_0, ϵ_0, κ_0, γ_0, m_0, $\hat{\alpha}$, $\hat{\beta}$, $\hat{\delta}$, $\hat{\epsilon}$, $\hat{\kappa}$, $\hat{\gamma}$, $\hat{m}$, ψ_1, ψ_2, C_h,C_d, b_r,m_r

Here is a sample of using these keywords

```
#SOLID_PROPERTIES
  1 0.350000e+08  3.0000e-001 ; elasticity_modul, Possion ratio
  $THERMAL_EXPANSION_COEFFICIENT
    1.0e-5
  $DRUCKER-PRAGER
   1.0e6 ;Initial yield stress
   -1.0e+6 ; Plastic hardening modulus
   20.0 ;Internal frictional angle
   5.0 ;dilatancy angle
```

last modified: WW May 2003

12 File Formats

12.1 RFR File Format

The RFR file is to restart (i.e. continue) simulations. It can be produced with help of the RF2TP converter (Kunz 2002). The RFR file contains 3 blocks: data control, data definition and data area.

```
#0#0#13#1#901200000#9062#3805OK34#################################
1 22 16
```

Data control area

```
22 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
PRESSURE1, Pa
PRESSURE2, Pa
PRESSURECapillary, Pa
SATURATION1, -
SATURATION2, -
TEMPERATURE1, K
MASSFRACTION1, kg/kg
MASSFRACTION2, kg/kg
DISPLACEMENT_X0, m
DISPLACEMENT_Y0, m
DISPLACEMENT_Z0, m
DISPLACEMENT_X1, m
DISPLACEMENT_Y1, m
DISPLACEMENT_Z1, m
STRAIN_XX1,
STRAIN_YY1,
STRAIN_XY1,
STRESS_XX1, N/m2
STRESS_YY1, N/m2
STRESS_XY1, N/m2
DISPLACEMENT_XRES1, N
DISPLACEMENT_YRES1, N
```

Data definition area

```
0 101300.00000000 -110898700.00000 111000000.00000 0.85000000000000 0.15000000000000 ...
1 101300.05954008 -34067068.915785 34168368.975325 0.60290595206663 0.39709404793337 ...
2 101300.11295699 -20732808.901570 20834109.014527 0.53222739343018 0.46777260656982 ...
3 101300.33089324 -14747543.477143 14848843.808036 0.48771676189124 0.51228323810876 ...
4 101300.45920062 -11423090.965458 11524391.424659 0.45471546027399 0.54528453972601 ...
5 101300.55564668 -9590955.8857362 9692256.4413829 0.43230704356181 0.56769295643819 ...
6 101300.65236626 -8497353.6405735 8598654.2929397 0.41674852758369 0.58325147241631 ...
7 101300.71031441 -7809047.0644937 7910347.7748081 0.40576860635268 0.59423139364732 ...
8 101300.78160706 -7345426.8343881 7446727.6159951 0.39771717807398 0.60228282192602 ...
9 101300.80309042 -7021805.3615949 7123106.1646853 0.39172418823491 0.60827581176509 ...
10 101300.86740896 -6782734.5172004 6884035.3846094 0.38708070769219 0.61291929230781 ...
...
...
...
2430 0.00000000000000 5.2771464762532e-012 -2.0041375094110e-012 -2.0041842887472e-012 ...
2431 0.00000000000000 -1.9144884623022e-018 6.3468164316966e-013 -2.0041375094110e-012 ...
2432 0.00000000000000 2.9016201030660e-018 6.3468196274108e-013 6.3468164316966e-013 ...
2433 0.00000000000000 3.3496785694255e-019 6.3468111418126e-013 6.3468196274108e-013 ...
2434 0.00000000000000 2.2797634966230e-016 6.3456862808354e-013 6.3468111418126e-013 ...
2435 0.00000000000000 2.6385843839177e-012 -6.8479299992922e-013 6.3456862808354e-013 ...
```

Data area

12.1.1 Data control area

The parameters of the head line are separated by double-crosses.

Variable name	Parameter meaning
rfr_file->art	0: has to be zero
rfr_file->bin	0: ASCII format
nr	13:
rfr_file->geom	1:
startzeit	901200000: beginning from this time the simulation is continued
rfr_file->zeitschritt	9062: time step number of previous simualtion
rf_filetype	3805OK34: RF/RM version number
rfr_mesh_doesnt_fit_data	

The parameters of the second line are as follows.

Variable name	Parameter meaning
dateityp	1:
anz_n	22: 22 node values are written
anz_e	16: 16 element values are written
rfr_nodes	
rfr_elements	

12.1.2 Data definition area

The first line of the data definition area gives the number of model node values to output an their vector type.

Variable name	Parameter meaning
	22: number of node values to output 1: vector type of node value (scalar: 1 component) ...

It follows the definition of all node quantities: name and unit

Variable name	Parameter meaning
PRESSURE1, Pa	quantity name, quantity unit

The data definition area can be configured by a RF/RM model. Only those model node data will be written for which the parameter save is set.

```
in mod_xxxx.c
void MODConfigNODValues_XXXX(void)
{
  save=1;
  ModelsAddNodeValInfoStructure("PRESSURE1","Pa",save, 0, 1, 1, 0.);
}
```

12.1.3 Data area

The data area is simply a table of the specified node values.

```
NODE1 QUANTITY1 QUANTITY2 QUANTITY3 QUANTITY4 ...
NODE2 QUANTITY1 QUANTITY2 QUANTITY3 QUANTITY4 ...
NODE3 QUANTITY1 QUANTITY2 QUANTITY3 QUANTITY4 ...
...
```

last modified: OK 21.07.2003

Manual Authors

Section	Title	Object	Author		Last modified
1	Preface		OK	ZAG	20.07.2003
2	Data concept		OK	ZAG	
3	Control		RK	ISEB	
4	Models		RK	ISEB	
5	Time	TIM	CB	ZAG	
6	Output	OUT	OK	ZAG	
7	Numerics	NUM	WW	ZAG	
8	Solver	SOL	WW	ZAG	
9	Initial conditions	IC	MB	ZAG	
10	Boundary conditions	BC	MK	ISEB	
11	Source terms	ST	OK	ZAG	
12	Fluid properties	MAT-FP	JdJ	ZAG	
13	Solid properties	MAT-SP	MK	ISEB	
14	Medium properties	MAT-MP	CD	ZAG	
15	Component properties	MAT-CP	MX	ZAG	
16	Grid adaptation	ADP	RK	ISEB	
17	Geometry	GeoObj	TK/SM	ZAG/ISEB	
18	Meshing	MshObj	TK/SM	ZAG/ISEB	
19	RFI-format	RFI	TK	ZAG	
20	RFR-format	RFR	OK	ZAG	
21	RFM-format	RFM	CC	ZAG	
22	SHP-format	SHP	CC	ZAG	
23	DB interfaces	WebObj	JG	ZAG	
24	Parallel computing	ParObj	DK	ZAG	