Institute of Fluid Mechanics
and Environmental Physics
in Civil Engineering

Leibniz
Universität
Hannover

RockFlow 5
User's Manual

- Keyword Description -

M. Kohlmeier, J. Maßmann,
M. Wulkau, G. Ziefle

RockFlow – Version 5.1.10

Hannover – 12th January 2009

Contents

1 Preface

Previous and current work on ROCKFLOW

Version 1

Wollrath, J.: Ein Strömungs- und Transportmodell für klüftiges Gestein und Untersuchungen zu homogenen Ersatzsystemen. Dissertation, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen, Universität Hannover, Bericht Nr. 28/1990.

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

Version 2

Helmig, R.: Theorie und Numerik der Mehrphasenströmungen in geklüftet-porösen Medien. Dissertation, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen, Universität Hannover, Bericht Nr. 34/1993.

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

Lege, T.: Modellierung des Kluftgesteins als geologische Barriere für Deponien. Dissertation, Institut für Strömungsmechanik und elektronisches Rechnen im Bauwesen, Universität Hannover, Bericht Nr. 45/1995.

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

Kolditz, O.: Stoff- und Wärmetransport im Kluftgestein. Habilitation, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen, Universität Hannover, Bericht Nr. 47/1996.

Barlag C.: Adaptive Methoden zur Modellierung von Stofftransport im Kluftgestein. Dissertation, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen, Universität Hannover, Bericht Nr. 52/1997.

Version 3

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

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

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

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

Moenickes, S.: Grid Generation for Simulation of Flow and Transport Processes in Fractured-Porous Media. Dissertation, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen, Universität Hannover, Bericht Nr. 68/2004.

Version 5

Kohlmeier, M.: Coupling of thermal, hydraulic and mechanical processes for geotechnical simulations of partially saturated porous media. Dissertation, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen, Universität Hannover, 2006.

Ziefle, G.: Modeling aspects of coupled hydraulic-mechanical processes in clay material. Dissertation, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen, Leibniz Universität Hannover, 2008.

Current Developer

ISEB, Hannover:

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

Gesa Ziefle
THM-modelling, nonlinear material modelling

Jobst Maßmann
THM-modelling, mesh generation, material modelling

Meike Wulkau
Input concept

GGA, Hannover:

Stefan Weßling
Thermal and hydraulic modeling in the framework of underground coal fires

2 Data Files

RockFlow processes several files to input and output data. For each computation the program needs the two input files - the .rfd-file and the .rfi-file. At least two output files are generated during the computation - .rfe-file and .rfo-file. The .rfd-file contains a number of keywords which control all necessary adjustments, for example time stepping scheme, material properties, as well as physical and numerical parameters. The output .rfe-file contains a protocol of the .rfd-file.

2.1 Description of the RockFlow files

Input:

- .rfd-file: This file contains all information for the simulation control. It is keyword oriented, e. g. concerning the numerical parameters, material properties, considered procecces. All keywords are described in the following in detail. With the usage of the include option any ASCII-file can be included in the .rfd-file in order to increase the clearness; syntax: <<INCLUDE user_defined_file.XXX >>.
- .rfi-file: Contains the geometry and topologie of the problem. The structure is pictured in Fig. 1.
- .rfr-file (optional): contains node and element data to manage a restart. The structure is pictured in Fig. 2. An input file for a restart can be produced by copying parts of the .rfo-file.

Output:

- .rfo-file: This file contains the simulation results, subdivided in node and element data. With the keyword #OUTPUT the output times can be choosen.
- .rfe-file: This file contains the logging of reading the .rfd-file. It also generates comments to the input. Errors during the reading of the .rfd-file will be documented.
- .msg-file (optional): If this file exists (before RockFlow has been started) the terminal output will be written in this file. It contains information concerning the programm execution, as the current time step, number of iterations, etc. Also errors and warnings, as they may arise during the simulation, will be written in this file.
- .NXX-file: Output file for time profiles, defined with the keyword #OUTPUT_NODES.
- extended output: With the keyword #OUTPUT_EX exist many possibilities to specify an output, especially for *Tecplot*.

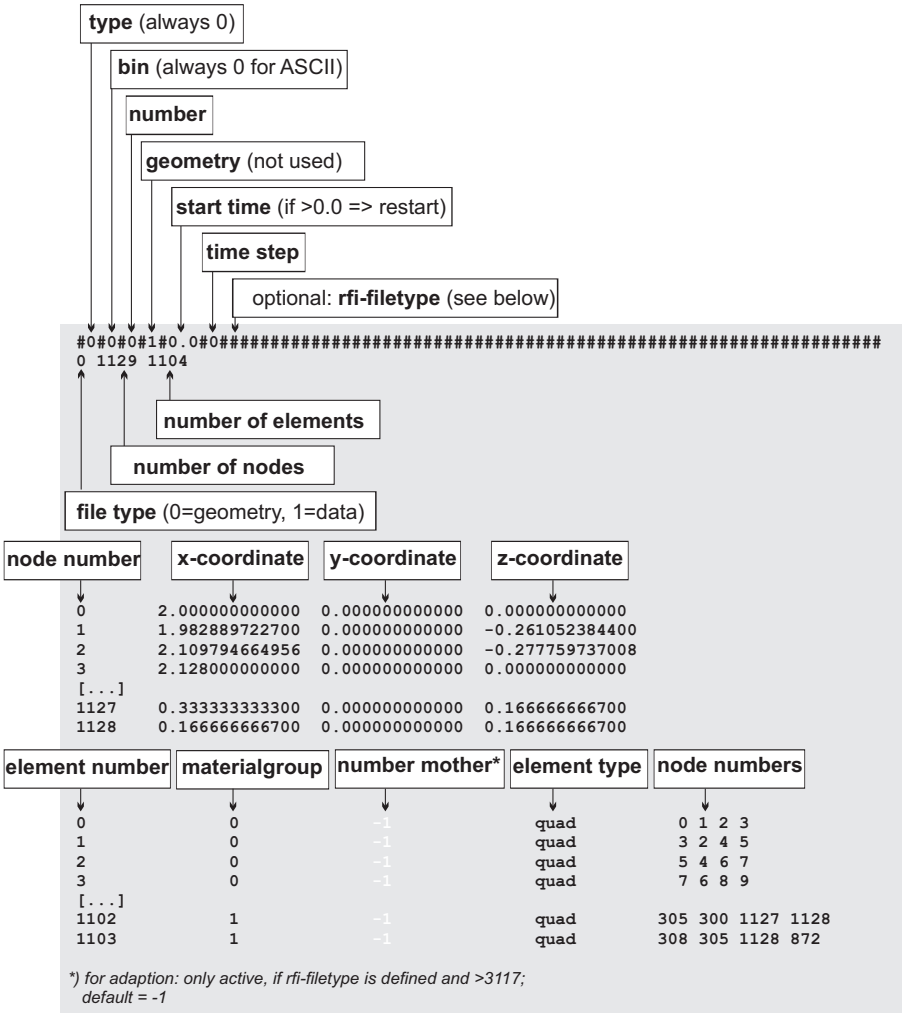

Figure 1: Structure of the `.rfi`-file.

Figure 2: Structure of the `.rfr`-file.

3 Project Title

3.1 Keyword Description

The keyword #PROJECT specifies the title of project. The text may have a maximal length of 250 characters.

#PROJECT			
Parameter	RF-Variable	Values	Meaning
N1	.project_name	[string]	Project name

3.2 Example

Input example 1:

```
#PROJECT
 2-D TRACER TRANSPORT BY UNSTEADY GAS FLOW  ; project name
```

4 Models (old concept)

4.1 Keyword Description

The keyword #MODEL controls the execution mode of the program and specifies the physico-chemical model. Alternative to #MODEL the keywords #CONTROL and #PROCESS_*process_name* can be used for coupled THM-modeling.

#MODEL			
Parameter	RF-Variable	Values	Meaning
N1	.modex	[int]	Simulation control −2: RFD input file will be tested. Result in RFE file −1: RFD and RFI input files will be tested. Result in RFE file 0: Mesh generation 1: Execution of the FEM−Simulation 2: Mesh generatio + Execution of FEM−Simulation
N2	.model	[int]	Models 00: SM Groundwater flow 01: SCM 02: GM Gas flow (compressible fluids) 06: Multiphase Flow 07: RM 10093: Heat transport by groundwater flow 10099: SM/TM 10299: GM/TM Reactive transport by gas flow 10097: SM/RTM Reactive transport by groundwater flow 10297: GM/RTM 10095: SM/MTM Multicomponent transport by ground water flow 10096: SM/TBC 10699: MM/TM Transport by multiphase model 10010: Consolidation processes
N3	.flow_model	[int]	Flow_model [0,1;0] 0: Darcy law (linear flow behavior) 1: Forchheimer law (non−linear flow behavior)
N4	.g_-iDensityDependentFlow	[int]	Density−dependent flow [0,1;0] 0: Constant density flow (flow of homogeneous liquid) 1: Density dependent flow (flow of heterogeneous liquid)
N5	.chemical_-model	[int]	Reactive transport model [0<=chemical_model<=2;0] 0: No sorption reaction 1: Equilibrium sorption reactions 2: Nonequilibrium sorption reactions
N6	.dummy	[int]	Not used
N7	.simulation	[int]	Optimization of coupled transient problems 0: Flow only (transient) 1: Unsteady both flow and transport 2: Steady flow and unsteady transport 3: Transport only (flow by IC) 4: Stroemungsfeld jeden n−ten Zeitschritt + Transport (transient)

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
if N7=4			Stroemungsfeld jeden n−ten Zeitschritt + Transport (transient)
N8	.timestep_n	[int]	Timestep_n
N9	.number_of_-groups	[long]	Number of material (soil/rock) groups [0<=;1]
N10	.number_of_-phases	[long]	Number of fluid phases [0<=;1]
N11	.number_of_-components	[long]	Number of chemical components [0<=;1]
N12	.adaptive_-mesh_-refinement_-flag	[int]	Grid adaptation flag 0: Disable adaptation 1: Enable adaptation
if next value=[int]			
N13	.chain_-reaction_-model	[int]	Chemical chain reactions 0: Disable chain reactions 1: Enable chain reactions
if next value=[int]			
N14	.heat_-reaction_-model	[int]	Heat reactions 0: Disable heat reactions 1: Enable heat reactions
if next value=[int]			
N15	.dummy	[int]	Not used
if next value=[int]			
N16	.mobile_-immobile_-model	[int]	Mobil−immobile−model 0: No mobil−immobile−model 1: Mobil−Immobil−Modell active

$NUMBER_OF_ELECTRIC_FIELDS			
N17	.number_of_-electric_fields	[int]	

$NUMBER_OF_PHASES			
N18	.number_of_-phases	[long]	Number_of_phases [0<=;1]

$NUMBER_OF_COMPONENTS			
N19	.number_of_-components	[long]	Number_of_components [0<=;1]

$NUMBER_OF_TEMPERATURES			
N20	.number_of_-temperatures	[int]	

$NUMBER_OF_GROUPS			
N21	.number_of_-groups	[long]	Number_of_groups [0<=;1]

$NUMBER_OF_CONTINUA

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N22	.number_of_-continua	[int]	Number_of_continua [0<=;1]

Default values are printed bold.

4.2 Examples

Input example 1: fluid flow (flow_singlephase/asm1d.rfd)

```
#MODEL
 1 ; simulation flag
 0 ; model identifier
 0 ; flow model flag
 0 ; convection model flag
 0 ; chemical model flag
 0 ; transport phase of multiphase model
 0 ; simulation optimizer flag
 1  1  0 ; material groups, phases, components
 0 ; adaptive mesh refinement flag
 0 ; chain_reaction_model
 0 ; heat_reaction_model
 0 ; saturation_calculation_method
 0 ; mobile immobile model flag
```

Input example 2: multicomponent transport with chain reaction (mass_transport/nitri1d.rfd)

```
#MODEL
 1 ; Steuerung
 10095 ; Rechenmodell
 0 ; Fliessgesetz(linear)
 0 ; Dichtestroemung
 1 ; Chemisches Modell
 0 ; Transportphase
 1 ; Simulationsparameter
 1 1 3 ; number_of_groups  number_of_phases  number_of_components
 0 ; adaptiv
 1 ; Kettenreaktionsmodell
 0 ; heat_reaction_model
 0 ; saturation_calculation_method
 0 ; mobile immobile model flag
```

5 Control

5.1 Keyword Description

The keyword #CONTROL is the control unit of the general program execution (e.g. simulation mode, grid adaptation or parallel computing) and affects all processes defined in the keyword #PROCESS_*process_name*. The keyword #CONTROL can contain a number of optional subkeyword (labeled by a $ sign):

#CONTROL			
Parameter	RF-Variable	Values	Meaning
$MODEX			
N1	.modex	[int]	Simulation control −2: RFD input file will be tested. Result in RFE file −1: RFD and RFI input files will be tested. Result in RFE file 0: Mesh generation 1: Execution of the FEM−Simulation 2: Mesh generation + excution of the FEM−Simulation
$MODEL			
N2	.model	[int]	Models 00: SM Groundwater flow 01: SCM 02: GM Gas flow (compressible fluids) 06: Multiphase Flow 07: RM 10093: Heat transport by groundwater flow 10099: SM/TM 10299: GM/TM Reactive transport by gas flow 10097: SM/RTM Reactive transport by groundwater flow 10297: GM/RTM 10095: SM/MTM Multicomponent transport by ground water flow 10096: SM/TBC 10699: MM/TM Transport by multiphase model 10010: Consolidation processes
$FULLY_PARTITIONED_PROCESSING			
N3	.fully_-partitioned_-model	[int]	partitioniert / monolithisch 0: partitioniert / monolithisch 1: partitioniert
$SIMULATION			
N4	.simulation	[int]	Optimization of coupled transient problems [0<=simulation<=3;2] 0: Flow only (transient) 1: Unsteady both flow and transport 2: Steady flow and unsteady transport 3: Transport only (flow by IC) 4: Stroemungsfeld jeden n−ten Zeitschritt + Transport (transient)

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
if N4=4			Stroemungsfeld jeden n−ten Zeitschritt + Transport (transient)
N5	.simulation_n	[int]	simulation_n

$GRIDADAPTATION			
N6	.adaptive_-mesh_-refinement_-flag	[int]	Grid adaption [0<=flag<=1;1] 0: Disable 1: Enable

$NUMBER_OF_GROUPS			
N7	.number_of_-groups	[long]	Number of material groups [0<=;1]

$NUMBER_OF_CONTINUA			
N8	.number_of_-continua	[int]	Number_of_continua [0<=;1]

5.2 Example

Input example 1: (thm_plus/dmsm2d.rfd)

```
#CONTROL                        ;#MODEL
$MODEX 1                        ; 1 ; simulation flag
                                ; 10010 ; model identifier
                                ; 0 ; flow model flag
                                ; 0 ; convection model flag
                                ; 0 ; chemical model flag
                                ; 0 ; transport phase of multiphase model
$SIMULATION 1                   ; 1 ; simulation optimizer flag
$NUMBER_OF_GROUPS 1             ; 1  1  1 ; material groups, phases, components
                                ; 0 ; adaptive mesh refinement flag
                                ; 0 ; chain reaction model flag
                                ; 0 ; heat reaction model flag
                                ; 0 ; don't calc phase number i
                                ; 0 ; mobile immobile model flag
```

6 Processes

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

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.

```
#PROCESS_HEAT_TRANSPORT
#PROCESS_FLUID_FLOW
#PROCESS_SOLID_DEFORMATION
#PROCESS_MASS_TRANSPORT
```

Keyword combinations for flow processes

FLUID FLOW PROCESS	**NUMBER_OF_PHASES**	**NUMBER_OF_FLUID_COMPONENTS**	**NON_ISOTHERMAL_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

6.1 Keyword Description

#PROCESS_FLUID_FLOW			
Parameter	RF-Variable	Values	Meaning
$ACTIVATION			
N1	.activation	[int]	Process (in)activation 0: Inactive (no element routines, no assembling or solving, process dependent data structures are generated) 1: Active
$CONDITIONAL_DEACTIVATION			

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N2	.cond_deact_-type	[int]	Conditional process deactivation type 1: Selection of elements by element number 2: Selection of elements by material group 10: Selection of elements by material group in a plane convex polygon / prism All criterions have to be fulfilled
if N2=1			Selection of elements by element number
N3	.number_of_-times	[long]	number of times (= number of elements to deactivate)
N3 * Loop			
N4	.time_0	[double]	Time, as from deactivation takes place [>=0.0]
N5	.ele_0	[long]	Element to deactivate [>=0]
if N2=2			Selection of elements by material group
N6	.number_of_-times	[long]	number of times (= number of elements to deactivate)
N6 * Loop			
N7	.time_0	[double]	Time, as from deactivation takes place [>=0.0]
N8	.mg_0	[long]	Material group to deactivate [>=0]
if N2=10			Selection of elements by material group in a plane convex polygon / prism All criterions have to be fulfilled
N9	.number_of_-materialgroups	[long]	number of material groups (0 => all)
N10	.materialgroups	[N9* long]	MG1, MG2, ...[count_of_mg * long]
N11	.cond_deact_-time1	[double]	Deactivation starts after this time [>=0.0;0.0]
N12	.cond_deact_-type_eq_10_-count_of_-points	[long]	count of polygone points [>2]
N12 * Loop			
N13	.x_0	[double]	Coordinate x[0] (expr)
N14	.y_0	[double]	Coordinate y[0] (expr)
N15	.z_0	[double]	Coordinate z[0] (expr)
N16	.radius	[double]	Distance perpendicular to polygon−plane (expr) [;1] If only a 2−D polygone is needed, choose the value equal to epsilon (tolerance) This value defines the extent of the right prism, perpendicular to the polygone plane. If it is >0.0, it defines the extent in both directions of the polygone. If it is <0.0, it only defines the extent in negative direction of the polygone.
N17	.epsilon	[double]	geometric tolerance to catch nodes(expr) [>0.]

$FLOW_MODEL			
N18	.flow_model	[int]	Flow model [0,1;0] 0: Darcy law (linear flow behavior) 1: Forchheimer law (non−linear flow behavior)

$FLUID_FLOW_MODEL

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N19	.fluid_flow_-model	[int]	Fluid flow model [0,1,2;0] 0: Darcy law (linear flow behavior) 1: Forchheimer law (non−linear flow behavior) 2: Richards formulation (unsaturated flow)

$DENSITY_FLOW			
N20	.g_-iDensityDependentFlow	[int]	Density−dependent flow [0,1;0] 0: Constant density flow (flow of homogeneous liquid) 1: Density dependent flow (flow of heterogeneous liquid)

$NUMBER_OF_PHASES			
N21	.number_of_-phases	[long]	Number of fluid phases [0<=;1]

$NUMBER_OF_FLUID_COMPONENTS			
N22	.number_-of_fluid_-components	[long]	number of fluid components (e.g. air, water) contained in a fluid phase [0<=;1]

$NON_ISOTHERMAL_FLOW			
N23	.non_-isothermal_-flow	[int]	Nonisothermal flow [0,1;0] 0: Disable 1: Enable (corresponding phase changes are taken into account)

$COUPLED_SOLVER_ID			
N24	.coupled_-solver_id	[int]	Activation of specified couplings: 0: Standard method −1: No monolithic solving (iterative) >0: Assign coupled solver id to process

$INFLUENCING_PROCESSES			
if next value=[string]			
N25	.process_-name	[string]	Consideration of influence of this process

#PROCESS_HEAT_TRANSPORT

Parameter	RF-Variable	Values	Meaning

$ACTIVATION			
N1	.activation	[int]	Process (in)activation 0: Inactive (no element routines, no assembling or solving, process dependent data structures are generated) 1: Active

$CONDITIONAL_DEACTIVATION

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N2	.cond_deact_-type	[int]	Conditional process deactivation type 1: Selection of elements by number 2: Selection of elements by material group 10: Selection of elements by material group in a polygon
if N2=1			Selection of elements by number
N3	.number_of_-times	[long]	number of times
N3 * Loop			
N4	.time_0	[double]	Time, as from deactivation takes place [>=0.0]
N5	.ele_0	[long]	Element to deactivate [>=0]
if N2=2			Selection of elements by material group
N6	.number_of_-times	[long]	number of times
N6 * Loop			
N7	.time_0	[double]	Time, as from deactivation takes place [>=0.0]
N8	.mg_0	[long]	Material group to deactivate [>=0]
if N2=10			Selection of elements by material group in a polygon
N9	.number_of_-materialgroups	[long]	number of material groups (0 => all)
N10	.materialgroups	[N9* long]	MG1, MG2, ...
N11	.cond_deact_-time1	[double]	Deactivation starts after this time [>=0.0;0.0]
N12	.cond_deact_-type_eq_10_-count_of_-points	[long]	count of polygon points [>=2]
N12 * Loop			
N13	.x_0	[double]	Coordinate x[0] (expr)
N14	.y_0	[double]	Coordinate y[0] (expr)
N15	.z_0	[double]	Coordinate z[0] (expr)
N16	.radius	[double]	Radius, distance normal to polygon plane (expr) [;1]
N17	.epsilon	[double]	tolerance to catch nodes(expr) [>0.]

$NUMBER_OF_TEMPERATURES			
N18	.number_of_-temperatures	[int]	Number of temperatures [0<=;1]

$COUPLED_SOLVER_ID			
N19	.coupled_-solver_id	[int]	Activation of specified couplings: 0: Standard method −1: No monolithic solving (iterative) >0: Assign coupled solver id to process

$INFLUENCING_PROCESSES			
if next value=[string]			
N20	.process_-name	[string]	Consideration of influence of this process

#PROCESS_SOLID_DEFORMATION			
Parameter	RF-Variable	Values	Meaning

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
$ACTIVATION			
N1	.activation	[int]	Process (in)activation 0: Inactive (no element routines, no assembling or solving, process dependent data structures are generated) 1: Active
$CONDITIONAL_DEACTIVATION			
N2	.cond_deact_type	[int]	Conditional process deactivation type 1: Selection of elements by number 2: Selection of elements by material group 10: Selection of elements by material group in a polygon
if N2=1			Selection of elements by number
N3	.number_of_times	[long]	number of times
N3 * Loop			
N4	.time_0	[double]	Time, as from deactivation takes place [>=0.0]
N5	.ele_0	[long]	Element to deactivate [>=0]
if N2=2			Selection of elements by material group
N6	.number_of_times	[long]	number of times
N6 * Loop			
N7	.time_0	[double]	Time, as from deactivation takes place [>=0.0]
N8	.mg_0	[long]	Material group to deactivate [>=0]
if N2=10			Selection of elements by material group in a polygon
N9	.number_of_materialgroups	[long]	number of material groups (0 => all)
N10	.materialgroups	[N9* long]	MG1, MG2, ...
N11	.cond_deact_time1	[double]	Deactivation starts after this time [>=0.0;0.0]
N12	.cond_deact_type_eq_10_count_of_points	[long]	count of polygon points [>=2]
N12 * Loop			
N13	.x_0	[double]	Coordinate x[0] (expr)
N14	.y_0	[double]	Coordinate y[0] (expr)
N15	.z_0	[double]	Coordinate z[0] (expr)
N16	.radius	[double]	Radius, distance normal to polygon plane (expr) [;1]
N17	.epsilon	[double]	tolerance to catch nodes(expr) [>0.]
$PLASTICITY			
N18	.plasticity	[int]	Selection of plasticity model 0: no plasticity 1: Drucker−Prager
$COUPLED_SOLVER_ID			

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N19	.coupled_-solver_id	[int]	Activation of specified couplings: 0: Standard method −1: No monolithic solving (iterative) >0: Assign coupled solver id to process

$INFLUENCING_PROCESSES			
if next value=[string]			
N20	.process_-name	[string]	Consideration of influence of this process

#PROCESS_MASS_TRANSPORT

Parameter	RF-Variable	Values	Meaning

$ACTIVATION			
N1	.activation	[int]	Process (in)activation 0: Inactive (no element routines, no assembling or solving, process dependent data structures are generated) 1: Active

$CONDITIONAL_DEACTIVATION			
N2	.cond_deact_-type	[int]	Conditional process deactivation type 1: Selection of elements by number 2: Selection of elements by material group 10: Selection of elements by material group in a polygon
if N2=1			Selection of elements by number
N3	.number_of_-times	[long]	number of times
N3 * Loop			
N4	.time_0	[double]	Time, as from deactivation takes place [>=0.0]
N5	.ele_0	[long]	Element to deactivate [>=0]
if N2=2			Selection of elements by material group
N6	.number_of_-times	[long]	number of times
N6 * Loop			
N7	.time_0	[double]	Time, as from deactivation takes place [>=0.0]
N8	.mg_0	[long]	Material group to deactivate [>=0]
if N2=10			Selection of elements by material group in a polygon
N9	.number_of_-materialgroups	[long]	number of material groups (0 => all)
N10	.materialgroups	[N9* long]	MG1, MG2, ...
N11	.cond_deact_-time1	[double]	Deactivation starts after this time [>=0.0;0.0]
N12	.cond_deact_-type_eq_10_-count_of_-points	[long]	count of polygon points [>=2]
N12 * Loop			
N13	.x_0	[double]	Coordinate x[0] (expr)
N14	.y_0	[double]	Coordinate y[0] (expr)

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N15	.z_0	[double]	Coordinate z[0] (expr)
N16	.radius	[double]	Radius, distance normal to polygon plane (expr) [;1]
N17	.epsilon	[double]	tolerance to catch nodes(expr) [>0.]

$CHEMICAL_MODEL			
N18	.chemical_-model	[int]	Reactive transport model [0<=chemical_model<=2;0] 0: No sorption reaction 1: Equilibrium sorption reactions 2: Nonequilibrium sorption reactions

$CHAIN_REACTION_MODEL			
N19	.chain_-reaction_-model	[int]	Chemical chain reactions 0: Disable chain reactions 1: Enable chain reactions

$MOBILE_IMMOBILE_MODEL			
N20	.mobile_-immobile_-model	[int]	Mobil−immobile−model 0: No mobil−immobile−model 1: Mobil−Immobil−Modell active

$NUMBER_OF_COMPONENTS			
N21	.number_of_-components	[long]	Number_of_components [0<=;1]

$COUPLED_SOLVER_ID			
N22	.coupled_-solver_id	[int]	Activation of specified couplings: 0: Standard method −1: No monolithic solving (iterative) >0: Assign coupled solver id to process

$INFLUENCING_PROCESSES			
if next value=[string]			
N23	.process_-name	[string]	Consideration of influence of this process

#PROCESS_GEOELECTRIC

Parameter	RF-Variable	Values	Meaning

$NUMBER_OF_ELECTRIC_FIELDS			
N1	.number_of_-electric_fields	[int]	Number of electric fields [0<=;1]

$COUPLED_SOLVER_ID			
N2	.coupled_-solver_id	[int]	Activation of specified couplings: 0: Standard method −1: No monolithic solving (iterative) >0: Assign coupled solver id to process

$INFLUENCING_PROCESSES

continued

Continuation from the previous page

Parameter		RF-Variable	Values	Meaning
if next value=[string]				
	N3	.process_-name	[string]	Consideration of influence of this process

6.2 Examples

Input example 1: Conditional deactivation by elements
(benchmarks/process_deactivation/hm_deact01_2d)

```
#PROCESS_SOLID_DEFORMATION
 $ACTIVATION 1              ; mechanic modell activ
 $CONDITIONAL_DEACTIVATION
 1 ; Conditional deactivation type
 5 ; number of times
   6.  6 ; time element
   7.  7 ; time element
   8.  8 ; time element
   9.  9 ; time element
  10. 11 ; time element

#PROCESS_FLUID_FLOW
 $ACTIVATION 1             ; hydraulic modell activ
 $CONDITIONAL_DEACTIVATION
 1 ; Conditional deactivation type
 6 ; number of times
  3.  6 ; time element
  3. 11 ; time element
  6.  6 ; time element
  7.  7 ; time element
  8.  8 ; time element
  9.  9 ; time element
```

Input example 2: Conditional Deactivation by polygone, excavation process (benchmarks/process_deactivation/hm_deact01_3d)

```
#PROCESS_SOLID_DEFORMATION
 $ACTIVATION 1              ; mechanic modell activ
 $CONDITIONAL_DEACTIVATION
 10 ; type of deactivation
  0 ; nof material groups (0=>all)
      ; 0 1 ; MG1, MG2
  0.0 ;  deactivation after this time time

  4                ; nof polygone points (marks shape of opening)
  1.0  0.2  0.0  ;xyz
  0.6  0.2  0.0  ;xyz
  0.6  0.6  0.0  ;xyz
  1.  0.6   0.0  ;xyz
  { CURVE 1 }    ; radius, distance normal to polygon plane
  0.05           ; eps

#CURVES ;1 deactivation (progress of excavation)
; time     distance normal to polygon plane
  0.0      0.0
  3.0      1.0
 14.0     10.0
```

Further examples, using #PROCESS_XXX see RockFlow Tutorial D.

7 Time Discretization

7.1 Keyword Description

Using the keyword **#TIME** , the time stepping scheme can be specified.

#TIME			
Parameter	RF-Variable	Values	Meaning
N1	.endezeit	[double]	Final simulation time in [sec] (the computation will be stopped then) [>=0.0;0.0] 0.0: no abort
N2	.anz_-zeitschritte	[long]	Maximum number of time steps [>=0;0] 0: no abort
N3	.dt_steu	[int]	Time step control [0..5;#] 0: Strictly prescribed time stepping scheme will be used 1..5: Time stepping scheme will be controled by the model
n * Loop (solange Werte folgen)			Repeating data Number of time steps [>0;#] Time step length in [sec] [>0.0;#]
N4	.timesteps_0	[long double]	Time step number [0], time step length [0]

7.2 Examples

Input example 1: Tracertransport (tutorial_a-b-c/rtm1d.rfd)

```
#TIME
 0.0     ; final simulation time
 0       ; maximum time step number
 0       ; time step control
 100     ; time step number
 86400.0 ; time step length
```

Input example 2: fluid flow, solid deformation, swelling (applications/d_iv_phase01.rfd)

```
#TIME
 32832000         ; final simulation time: 1a + 15 d
 30              ; maximum time step number
 0               ; time step control
 10   129600.0   ; ersten 15 tage in 10 schritten
 20  1576800.0   ; 20 * 1/20 a
```

8 Data Output

8.1 Standard Output of Nodal and Element Results

8.1.1 Keyword Description

Keyword **#OUTPUT** controls the standard output of nodal and element results.

#OUTPUT			
Parameter	RF-Variable	Values	Meaning
N1	.output_many	[int]	Output in files [0,1;0] 0: All results are written to the same file 1: Each result is written to a single file
N2	.output_geom	[int]	Output geometry [0,1;0] 0: Geometry will be written only onces 1: Geometry will be written at each timestep
N3	.output_start	[int]	Output initial/boundary conditions [0,1;0] 0: Initial and boundary conditions will not be written 1: Initial and boundary conditions will be written
N4	.output_bin	[int]	Output format [0,1;0] 0: Results will be written in ASCII format 1: Results will be written in compact binary format (developing)
N5	.output_-restart	[int]	Output numbering [0,1;0] 0: Results are numbered subsequently 1: Results are numbered beginning with zero
N6	.output_-model	[int]	Output model [0..5;0] 0: No output at all 1: Output only after the last timestep 2: Output every output_timesteps[0] 3: Output every output_times[0] 4: Output after timesteps: output timesteps[i] 5: Output after times: output times[i]
if N6=0			No output at all
if N6=1			Output only after the last timestep
if N6=2			Output every output_timesteps[0]
N7	.output_-timesteps_0	[long]	Output_timesteps[0] [>0;#]
if N6=3			Output every output_times[0]
N8	.output_-times_0	[double]	Output_times[0] [>0.0;#] [seconds]
if N6=4			Output after timesteps: output timesteps[i]
n * Loop (solange Werte folgen)			
N9	.output_-timesteps_0	[long]	Output_timesteps[0] [>0;#]
if N6=5			Output after times: output times[i]
n * Loop (solange Werte folgen)			
N10	.output_-times_0	[double]	Output_times[0] [>0;#]

$ADDITIONAL_POINTS

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N11	.Flag_for_-output_of_-quadratic_-node_data	[int]	if activated(=1), data output also at non−corner element nodes [int;0]
N12	.Flag_for_-output_at_-gaussian_-points	[int]	if activated(=1), data output also at gaussian points [int;0]

8.1.2 Example

Input example 1: fluid flow (thm_plus_hm/effect_stress_coeff.rfd)

```
#OUTPUT
 0              ; one rfo.-file
 0              ; no geometry every timestep
 1              ; initial condition
 0              ; format ASCII
 0              ; numbering
 2  10          ; output every 10 timestep

 $ADDITIONAL_POINTS
 1 ; output at non-corner nodes ("quadratic" nodes)
 0 ; output at gaussian points (not yet implemented)
```

8.2 Temporal Profiles

8.2.1 Keyword Description

Keyword **#OUTPUT_NODES** specifies nodes, for which temporal breakthrough curves will be written.

#OUTPUT_NODES

Parameter	RF-Variable	Values	Meaning
N1	.output_-nodes	[int]	Format [;0] 0: No output >0: Node results will be written in ASCII format <0: Node results will be written in binary format Value corresponds to output time step (a maximum of 100 nodes can be specified)
if N1!=0			
n * Loop (solange Werte folgen)			
N2	.output_-node_array_0	[long]	Output node array[0] [>0;#]

Keyword **#OUTPUT_ELEMENTS** specifies elements, for which temporal breakthrough curves will be written.

#OUTPUT_ELEMENTS

Parameter	RF-Variable	Values	Meaning

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N1	.output_-elements	[int]	Format [;0] 0: No output >0: Element results will be written in ASCII format <0: Element results will be written in binary format Value corresponds to output time step (a maximum of 100 elements can be specified)
if N1!=0			
n * Loop (solange Werte folgen)			
N2	.output_-element_-array_0	[long]	Output element array[0] [>0;#]

8.3 Extended Output

8.3.1 Keyword Description

With the optional keyword an user defined extended output is possible.

#OUTPUT_EX			
Parameter	RF-Variable	Values	Meaning
N1	.type	[int]	type [>=0;0] 0: Sicherheitsausgabe 1: Knotendurchbruchskurve (Eingabe ueber Koordinaten x,y und z) 2: Profil−Ausgabe (Ausgabe aller Zeitschritte) 3: Profil−Ausgabe (Ausgabe zu bestimmten Zeiten) 5: Output of specified variables 6: Knotendaten−Ausgabe (Ausgabe aller Zeitschritte) 7: Specified quantities at given time points 10: Contour plots for Tecplot 11: Temporal breakthrough curves (readable in Tecplot) 12: like Typ 10, additional: element data projected on node locations (only uniform meshes) 13: Node values along geometric object (readable in Tecplot) 15: Tecplot: special output 20: For GIS: geometry, topology, materialgroups (generate−format) 21: For GIS / Surfer: values 30: Documentation of input parameters
if N1=0			Sicherheitsausgabe
N2	.file_name	[string]	Name of output−file
N3	.values_0	[double]	Values [0]
if N1=1			Knotendurchbruchskurve (Eingabe ueber Koordinaten x,y und z)
N4	.file_name	[string]	Name of output−file
N5	.x_0	[double]	Coordinate x [0] [>=0;−1]
N6	.y_0	[double]	Coordinate y [0] [>=0;−1]
N7	.z_0	[double]	Coordinate z [0] [>=0;−1]
if N1=2			Profil−Ausgabe (Ausgabe aller Zeitschritte)
N8	.file_name	[string]	Name of output−file
if N1=3			Profil−Ausgabe (Ausgabe zu bestimmten Zeiten)
N9	.file_name	[string]	Name of output−file
N10	.nr_of_times	[long]	Nr of output times
N11	.values	[N10* double]	Output times [sec]
N12	.time_radius	[double]	Time radius (temporal tolerance) [sec]
if N1=4			
N13	.file_name	[string]	Name of output−file
N14	.values_0	[double]	Values [0]
N15	.time_radius	[double]	Time radius (temporal tolerance) [sec]
N16	.x_0	[double]	Coordinate x [0] [>=0;−1]
N17	.y_0	[double]	Coordinate y [0] [>=0;−1]
N18	.z_0	[double]	Coordinate z [0] [>=0;−1]

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N19	.nr_of_-variables	[long]	Number of variables (0=all)
N20	.data_names	[N19* string]	Variables (e.g. X Y Z PRESSURE)
if N1=5			Output of specified variables
N21	.file_name	[string]	Name of output−file
N22	.nr_of_times	[long]	Nr of output times
N23	.values	[N22* double]	Output times [sec]
N24	.time_radius	[double]	Time radius (temporal tolerance) [sec]
N25	.nr_of_-variables	[long]	Number of variables (0=all)
N26	.data_names	[N25* string]	Variables (e.g. X Y Z PRESSURE)
if N1=6			Knotendaten−Ausgabe (Ausgabe aller Zeitschritte)
N27	.file_name	[string]	Name of output−file
N28	.nr_of_-variables	[long]	Number of variables (0=all)
N29	.data_names	[N28* string]	Variables (e.g. X Y Z PRESSURE)
if N1=7			Specified quantities at given time points
N30	.file_name	[string]	Name of output−file
N31	.nr_of_times	[long]	Nr of output times
N32	.values	[N31* double]	Output times [sec]
N33	.time_radius	[double]	Time radius (temporal tolerance) [sec]
N34	.nr_of_-variables	[long]	Number of variables (0=all)
N35	.data_names	[N34* string]	Variables (e.g. X Y Z PRESSURE)
if N1=8			
N36	.file_name	[string]	Name of output−file
N37	.nr_of_-variables	[long]	Number of variables (0=all)
N38	.data_names	[N37* string]	Variables (e.g. X Y Z PRESSURE)
if N1=10			Contour plots for Tecplot
N39	.file_name	[string]	Name of output−file
N40	.mode	[int]	mode 1: Output by times 2: Output by steps
N41	.method	[int]	method
N42	.data_-output_-method	[int]	data output method 0: node and element data in separated Tecplot Zones 1: data with different dimensions separated 2: data with different material groups separated
if N40=1			Output by times
if N41=0			Output after specified times
N43	.number_of_-values	[long]	Number of times
N44	.values	[N43* double]	Time1, time2 ... [sec]
N45	.time_radius	[double]	Time radius (temporal tolerance) [sec]
if N41=1			Output every specified time

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N46	.values_0	[double]	Values [0]
N47	.time_radius	[double]	Time radius (temporal tolerance) [sec]
if N40=2			Output by steps
if N41=0			Output after specified steps
N48	.number_of_values	[long]	Number of steps
N49	.values	[N48* double]	Step1, step2,...
if N41=1			Output every specified step
N50	.values_0	[double]	Values [0]
N51	.nr_of_variables	[long]	Number of variables (0=all)
N52	.data_names	[N51* string]	Variables (e.g. X Y Z PRESSURE)
if N1=11			Temporal breakthrough curves (readable in Tecplot)
N53	.file_name	[string]	Name of output−file
N54	.x_0	[double]	Coordinate x [0] [>=0;−1]
N55	.y_0	[double]	Coordinate y [0] [>=0;−1]
N56	.z_0	[double]	Coordinate z [0] [>=0;−1]
N57	.mode	[int]	mode 1: Output by times 2: Output by steps
N58	.method	[int]	method
N59	.data_output_method	[int]	dummy
if N57=1			Output by times
if N58=0			Output after specified times
N60	.number_of_values	[long]	Number of times
N61	.values	[N60* double]	Time1, time2 ... [sec]
N62	.time_radius	[double]	Time radius (temporal tolerance) [sec]
if N58=1			Output every specified time
N63	.values_0	[double]	Values [0]
N64	.time_radius	[double]	Time radius (temporal tolerance) [sec]
if N57=2			Output by steps
if N58=0			Output after specified steps
N65	.number_of_values	[long]	Number of steps
N66	.values	[N65* double]	Step1, step2,...
if N58=1			Output every specified step
N67	.values_0	[double]	Values [0]
N68	.nr_of_variables	[long]	Number of variables (0=all)
N69	.data_names	[N68* string]	Variables (e.g. X Y Z PRESSURE, also possible FUNCTIONS CURVES)
if N1=12			like Typ 10, additional: element data projected on node locations (only uniform meshes)
N70	.file_name	[string]	Name of output−file

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N71	.mode	[int]	mode 1: Output by times 2: Output by steps
N72	.method	[int]	method
N73	.data_-output_-method	[int]	data output method 0: node and element data in separated Tecplot Zones 1: data with different dimensions separated 2: data with different material groups separated
if N71=1			Output by times
if N72=0			Output after specified times
N74	.number_of_-values	[long]	Number of times
N75	.values	[N74* double]	Time1, time2 ... [sec]
N76	.time_radius	[double]	Time radius (temporal tolerance) [sec]
if N72=1			Output every specified time
N77	.values_0	[double]	Values [0]
N78	.time_radius	[double]	Time radius (temporal tolerance) [sec]
if N71=2			Output by steps
if N72=0			Output after specified steps
N79	.number_of_-values	[long]	Number of steps
N80	.values	[N79* double]	Step1, step2,...
if N72=1			Output every specified step
N81	.values_0	[double]	Values [0]
N82	.nr_of_-variables	[long]	Number of variables (0=all)
N83	.data_names	[N82* string]	Variables (e.g. X Y Z PRESSURE)
if N1=13			Node values along geometric object (readable in Tecplot)
N84	.file_name	[string]	Name of output−file
N85	.geo_type	[int]	Geo type 1: point, value will be interpolated (developing) 2: line
if N85=1			point, value will be interpolated (developing)
N86	.x_0	[double]	Coordinate x [0] [>=0;−1]
N87	.y_0	[double]	Coordinate y [0] [>=0;−1]
N88	.z_0	[double]	Coordinate z [0] [>=0;−1]
N89	.radius	[double]	Radius (geometric tolerance)
if N85=2			line
N90	.x_0	[double]	Coordinate x [0] [>=0;−1]
N91	.y_0	[double]	Coordinate y [0] [>=0;−1]
N92	.z_0	[double]	Coordinate z [0] [>=0;−1]
N93	.x_1	[double]	Coordinate x [1] [>=0;−1]
N94	.y_1	[double]	Coordinate y [1] [>=0;−1]
N95	.z_1	[double]	Coordinate z [1] [>=0;−1]
N96	.radius	[double]	Radius (geometric tolerance)
N97	.mode	[int]	mode 1: Output by times 2: Output by steps
N98	.method	[int]	method

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N99	.data_-output_-method	[int]	data output method 0: node and element data in separated Tecplot Zones 1: data with different dimensions separated 2: data with different material groups separated
if N97=1			Output by times
if N98=0			Output after specified times
N100	.number_of_-values	[long]	Number of times
N101	.values	[N100* double]	Time1, time2 ... [sec]
N102	.time_radius	[double]	Time radius (temporal tolerance) [sec]
if N98=1			Output every specified time
N103	.values_0	[double]	Values [0]
N104	.time_radius	[double]	Time radius (temporal tolerance) [sec]
if N97=2			Output by steps
if N98=0			Output after specified steps
N105	.number_of_-values	[long]	Number of steps
N106	.values	[N105* double]	Step1, step2,...
if N98=1			Output every specified step
N107	.values_0	[double]	Values [0]
N108	.nr_of_-variables	[long]	Number of variables (0=all)
N109	.data_names	[N108* string]	Variables (e.g. X Y Z PRESSURE)
if N1=14			
N110	.file_name	[string]	Name of output−file
if N1=15			Tecplot: special output
N111	.file_name	[string]	Name of output−file
N112	.operation_-type	[int]	Operation type 1: Convergence calculation 21: calculation of the extent of the desaturated zone 22: calculation of the extent of the seasonal influenced zone
if N112=1			Convergence calculation
N113	.x_0	[double]	Coordinate x [0] [>=0;−1]
N114	.y_0	[double]	Coordinate y [0] [>=0;−1]
N115	.z_0	[double]	Coordinate z [0] [>=0;−1]
N116	.x_1	[double]	Coordinate x [1] [>=0;−1]
N117	.y_1	[double]	Coordinate y [1] [>=0;−1]
N118	.z_1	[double]	Coordinate z [1] [>=0;−1]
if N112=21			calculation of the extent of the desaturated zone
N119	.x_0	[double]	Coordinate x [0] [>=0;−1]
N120	.y_0	[double]	Coordinate y [0] [>=0;−1]
N121	.z_0	[double]	Coordinate z [0] [>=0;−1]
N122	.x_1	[double]	Coordinate x [1] [>=0;−1]
N123	.y_1	[double]	Coordinate y [1] [>=0;−1]
N124	.z_1	[double]	Coordinate z [1] [>=0;−1]
if N112=22			calculation of the extent of the seasonal influenced zone
N125	.x_0	[double]	Coordinate x [0] [>=0;−1]
N126	.y_0	[double]	Coordinate y [0] [>=0;−1]
N127	.z_0	[double]	Coordinate z [0] [>=0;−1]
N128	.x_1	[double]	Coordinate x [1] [>=0;−1]

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N129	.y_1	[double]	Coordinate y [1] [>=0;−1]
N130	.z_1	[double]	Coordinate z [1] [>=0;−1]
N131	.mode	[int]	mode 1: Output by times 2: Output by steps
N132	.method	[int]	method
N133	.data_-output_-method	[int]	data output method 0: node and element data in separated Tecplot Zones 1: data with different dimensions separated 2: data with different material groups separated
if N131=1			Output by times
if N132=0			Output after specified times
N134	.number_of_-values	[long]	Number of times
N135	.values	[N134* double]	Time1, time2 ... [sec]
N136	.time_radius	[double]	Time radius (temporal tolerance) [sec]
if N132=1			Output every specified time
N137	.values_0	[double]	Values [0]
N138	.time_radius	[double]	Time radius (temporal tolerance) [sec]
if N131=2			Output by steps
if N132=0			Output after specified steps
N139	.number_of_-values	[long]	Number of steps
N140	.values	[N139* double]	Step1, step2,...
if N132=1			Output every specified step
N141	.values_0	[double]	Values [0]
N142	.nr_of_-variables	[long]	Number of variables (0=all)
N143	.data_names	[N142* string]	SATURATION1 for output of desaturated/seasonal influenced zone
if N1=20			For GIS: geometry, topology, materialgroups (generate−format)
N144	.file_name	[string]	Name of output−file
N145	.mode	[int]	mode 1: Output by times 2: Output by steps
N146	.method	[int]	method
N147	.data_-output_-method	[int]	data output method 0: Id = number of node / element 1: Id = dimension 2: Id = material group
if N145=1			Output by times
if N146=0			Output after specified times
N148	.number_of_-values	[long]	Number of times
N149	.values	[N148* double]	Time1, time2 ... [sec]
N150	.time_radius	[double]	Time radius (temporal tolerance) [sec]
if N146=1			Output every specified time
N151	.values_0	[double]	Values [0]
N152	.time_radius	[double]	Time radius (temporal tolerance) [sec]
if N145=2			Output by steps

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
if N146=0			Output after specified steps
N153	.number_of_values	[long]	Number of steps
N154	.values	[N153* double]	Step1, step2,...
if N146=1			Output every specified step
N155	.values_0	[double]	Values [0]
N156	.nr_of_variables	[long]	Number of variables (0=all)
N157	.data_names	[N156* string]	Variables (e.g. X Y Z)
if N1=21			For GIS / Surfer: values
N158	.file_name	[string]	Name of output−file
N159	.mode	[int]	mode 1: Output by times 2: Output by steps
N160	.method	[int]	method
N161	.data_output_method	[int]	data output method 0: Id = number of node / element 1: Id = dimension 2: Id = material group
if N159=1			Output by times
if N160=0			Output after specified times
N162	.number_of_values	[long]	Number of times
N163	.values	[N162* double]	Time1, time2 ... [sec]
N164	.time_radius	[double]	Time radius (temporal tolerance) [sec]
if N160=1			Output every specified time
N165	.values_0	[double]	Values [0]
N166	.time_radius	[double]	Time radius (temporal tolerance) [sec]
if N159=2			Output by steps
if N160=0			Output after specified steps
N167	.number_of_values	[long]	Number of steps
N168	.values	[N167* double]	Step1, step2,...
if N160=1			Output every specified step
N169	.values_0	[double]	Values [0]
N170	.nr_of_variables	[long]	Number of variables (0=all)
N171	.data_names	[N170* string]	Variables (e.g. X Y Z PRESSURE)
if N1=30			Documentation of input parameters
N172	.file_name	[string]	Name of output−file

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N173	.mode	[int]	mode 0: Show all known keywords without parameters + marks, if found in RFD−file 1: Show all keywords, read in RFD−file 2: Show all keywords, found in RFD−file with parameters if method = 1 with detailed comments 3: Show indicated keywords, if found in RFD−file with parameters 4: XML−output: all keywords, found in RFD−file with parameters 5: XML−output: Show indicated keywords, if found in RFD−file with parameters 6: XML−output: default−tree 7: XML−output: default−tree indicated keywords 8: ASCII−output: default−tree 9: LaTex−output: default−tree (In output file: LaTex−def all keyword tables) 10: LaTex−output: default−tree (executable LaTex output file with all keyword tables)
if N173=0			Show all known keywords without parameters + marks, if found in RFD−file
if N173=1			Show all keywords, read in RFD−file
if N173=2			Show all keywords, found in RFD−file with parameters if method = 1 with detailed comments
if N173=3			Show indicated keywords, if found in RFD−file with parameters
N174	.nr_of_-keywords	[long]	Number of keywords (0=all)
N175	.data_names	[N174* string]	Keywords (e.g. TIME MODEL)
if N173=4			XML−output: all keywords, found in RFD−file with parameters
if N173=5			XML−output: Show indicated keywords, if found in RFD−file with parameters
N176	.nr_of_-keywords	[long]	Number of keywords (0=all)
N177	.data_names	[N176* string]	Keywords (e.g. X Y Z PRESSURE)
if N173=6			XML−output: default−tree
if N173=7			XML−output: default−tree indicated keywords
N178	.nr_of_-keywords	[long]	Number of keywords (0=all)
N179	.data_names	[N178* string]	Keywords (e.g. TIME MODEL)
if N173=8			ASCII−output: default−tree
if N173=9			LaTex−output: default−tree (In output file: LaTex−def all keyword tables)
if N173=10			LaTex−output: default−tree (executable LaTex output file with all keyword tables)
N180	.method	[int]	Method (0: short output 1: with details)
N181	.values_0	[double]	Values [0]
if N1=100			
N182	.file_name	[string]	Name of output−file

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N183	.mode	[int]	mode 1: Output by times 2: Output by steps
N184	.method	[int]	method
N185	.data_output_method	[int]	data output method
if N183=1			Output by times
if N184=0			Output after specified times
N186	.number_of_values	[long]	Number of times
N187	.values	[N186* double]	Time1, time2 ... [sec]
N188	.time_radius	[double]	Time radius (temporal tolerance) [sec]
if N184=1			Output every specified time
N189	.values_0	[double]	Values [0]
N190	.time_radius	[double]	Time radius (temporal tolerance) [sec]
if N183=2			Output by steps
if N184=0			Output after specified steps
N191	.number_of_values	[long]	Number of steps
N192	.values	[N191* double]	Step1, step2,...
if N184=1			Output every specified step
N193	.values_0	[double]	Values [0]
N194	.nr_of_variables	[long]	Number of variables (0=all)
N195	.data_names	[N194* string]	Variables (e.g. X Y Z PRESSURE)

8.3.2 Examples

Input example 1: Ausgabe time-gesteuert, feste zeiten

```
#OUTPUT_EX
 10                              ; type
 rtm1d_10_100.plt                ; name
 1                               ; mode
 0                               ; method
 0                               ; data_output_method
 3                               ; number of times
 0.0 4320000.0  8640000.0        ; times
 100.0                           ; time radius
 3                               ; number of variables
 X CONC PRESS                    ; variables
```

Input example 2: Ausgabe time-gesteuert, vielfache zeiten

```
#OUTPUT_EX
 10                             ; type
 rtm1d_10_110.plt               ; name
 1                              ; mode
 1                              ; method
 0                              ; data_output_method
 864000.0                       ; time
 100.0                          ; time radius
 3                              ; number of variables
 X CONC PRESS                   ; variables
```

Input example 3: Ausgabe von Durchbruchskurven

```
#OUTPUT_EX
 11                             ; type
 rtm1d_11_110_01.plt            ; name
 0.0 0.0 0.0                    ; x, y, z
 1                              ; mode
 1                              ; method
 0                              ; data_output_method
 86400.0                        ; time
 100.0                          ; time radius
 2                              ; number of variables
 CONC PRESS                     ; variables

 11                             ; type
 rtm1d_11_110_10.plt            ; name
 10.0 0.0 0.0                   ; x, y, z
 1                              ; mode
 1                              ; method
 0                              ; data_output_method
 86400.0                        ; time
 100.0                          ; time radius
 2                              ; number of variables
 CONC PRESS                     ; variables
```

Input example 4: Ausgabe von Durchbruchskurven und aller FUNCTIONS / CURVES Auswertungen

```
#OUTPUT_EX
 11                             ; type
 DBK_test1.plt                  ; name
 0.0 0.0 0.0                    ; x, y, z
 1                              ; mode
 1                              ; method
 0                              ; data_output_method
 86400                          ; time
 1.0                            ; time radius
 3                              ; number of variables
 PRESSURE1                      ; variables
 FUNCTIONS
 CURVES
```

Input example 5: Ausgabe entlang einer Linie

```
#OUTPUT_EX
 13                                ; type
 H_ANISOTROP_2D_LINE_DIAGONAL.PLT  ; file name
  2                                ; geo_type  (2=line)
  -1.000000 -1.000000 0.000000     ; x y z
   1.000000 1.000000 0.000000      ; x y z
  0.001000                         ; radius
  1                                ; mode   (1=output by times, 2=by steps)
  1                                ; method (0=fixed time/step,1=multiple)
  0                                ; data_output_method (2= mg seperated)
  1.000000e+003                    ; step/time
  1.000000e-001                    ; time radius
  0                                ; number of variables (0=all)
                                   ; variables
```

Input example 6: GIS-Ausgabe

```
#OUTPUT_EX
 20              ;Methode der Ausgabe
 netz.gen        ;Name
 1               ;mode
 0               ;method:   0 => Ausgabe zu angegebenen Zeiten
                 ;          1 => Ausgabe zu Zeiten, die dem Vielfachen
                 ;               der angegebenen Zeit entsprechen
 0               ;data_output_method:
                 ;          0 => Id der ausgegebenen Element/Knotendaten
                 ;               entspricht der Element/Knotennummer
                 ;          1 => Id der ausgegebenen Element/Kontendaten
                 ;               entspricht der Dimension
                 ;          2 => Id der ausgegebenen Element/Knotendaten
                 ;               entspricht der Materialgruppe des
                 ;               Elements/Knotens

 ;wenn method = 0
 3  0.0 5.0 10.0 ;Anzahl der Zeitschritte, Zeit1 Zeit2 Zeit3
 ;-------------------------------------------------------------
 ;wenn method = 1
 ;4.0            ;Zeit
 ;-------------------------------------------------------------
 1.0             ;Zeitradius
 2               ;Anzahl der Koordinatenachsen
 X Y             ;Bezeichnung der Koordinatenachsen
```

Input example 7: GIS/Surfer-Ausgabe

```
#OUTPUT_EX
 21              ;Methode der Ausgabe
 daten.log       ;Name
 1               ;mode
 0               ;method:   0 => Ausgabe zu angegebenen Zeiten
                 ;          1 => Ausgabe zu Zeiten, die dem Vielfachen
                 ;               der angegebenen Zeit entsprechen
 0               ;data_output_method:
                 ;          0 => Id der ausgegebenen Element/Knotendaten
                 ;               entspricht der Element/Knotennummer
                 ;          1 => Id der ausgegebenen Elementdaten
                 ;               entspricht der Dimension
                 ;          2 => Id der ausgegebenen Elementdaten
                 ;               entspricht der Materialgruppe des
                 ;               Elements
 ;wenn method = 0
 3  0.0 5.0 10.0 ;Anzahl der Zeitschritte, Zeit1 Zeit2 Zeit3
 ;-------------------------------------------------------------
 ;wenn method = 1
 ;4.0           ;Zeit
 ;-------------------------------------------------------------
 1.0                    ;Zeitradius
 6                      ;Anzahl der Variablen (inklusive Koordinatenachsen)
 X Y HEAD CONC          ;Bezeichnung der Variablen (Koordinaten, Knoten-
 X-VELOCITY Y-VELOCITY  ;                          daten, Elementdaten)
```

Input example 8: Input-Data / Keyword(-parameters)-Control

```
#OUTPUT_EX
 ; all known keywords without parameters + marks, if found in RFD-file
 30                                  ; type
 allKeywords.txt                     ; name
 0                                   ; mode
 1                                   ; method
 8640.0                              ; time

 ; all keywords, found in RFD-file with parameters
 30                                  ; type
 t=0_allKeywords.rfd                 ; name
 2                                   ; mode
 1                                   ; method
 0.0                                 ; time

 ; indicated keywords with parameters
 30                                   ; type
 NUMERIC_KEYWORDS.txt                 ; name
 3                                    ; mode
 3                                    ; number of keywords
 NUMERICS_PRESSURE NUMERICS_SATURATION ; keywords
 NUMERICS_CONCENTRATION
 0                                    ; method
 8640.0                               ; time

 ; XML-output: all active parameters
 30                                  ; type
 all_active_Keywords.xml              ; name
 4                                   ; mode
 1                                   ; method
 0.0                                 ; time

 ; LaTex-output: default-tree (executable LaTex output file with all keyword tables)
 30                                  ; type
 default_tree.tex                    ; name
 10                                   ; mode
 1                                   ; method
 0.0                                 ; time
```

9 Numerical Method

There are many alternative methods to solve initial-boundary-value problems arising from flow and transport processes in subsurface systems. In general these methods can be classified into analytical and numerical ones. Analytical solutions can be obtained for a number of problems involving linear or quasi-linear equations and calculation domains of simple geometry. For non-linear equations or problems with complex geometry or boundary conditions, exact solutions usually do not exist, and approximate solutions must be obtained. For such problems the use of numerical methods is advantegeous. In this chapter we use the Finite Difference Method to approximate time derivatives. The Finite Element Method as well as the Finite Volume Method are employed for spatial discretization of the region. The Galerkin weighted residual approach is used to provide a weak formulation of the PDEs. This methodology is more general in application than variational methods. The Galerkin approach works also for problems which cannot casted in variational form.

9.1 Keyword Description

The keyword **#NUMERICS** specifies numerical parameters for the FEM solution of the correspon-ding PDE.

#NUMERICS			
Parameter	RF-Variable	Values	Meaning
N1	.numerical_-method	[int]	Numerical method 0: Bubnov−Galerkin−Method (Standard FEM) 1: Petrov−Galerkin−Method (Upwind FEM) 2: Euler−Taylor−Galerkin−Method (Donea) 3: Euler−Langrange−Method (fuer Advektion) 4: Mixed FV−FEM
N2	.name	[string]	Object identification by name
N3	.gaussian_-points	[int]	Number of Gaussian integration points [1−4;2]
N4	.time_-collocation	[double]	Time collocation factor [0−1;1]
N5	.upwind_-parameter	[double]	Upwind parameter [0−1;1]
N6	.method	[int]	Advektiven Anteil fuer 2D−Elemente mit Lagrange−Methode berechnen [0;1;2] 0: Lagrangian scheme for advective terms disabled 1: Lagrangian scheme enabled (Operator−Splitting) 2: Lagrangian scheme enabled, alle anderen Anteile vernachlaessigen
N7	.minimalquality	[double]	Parameter of Lagrangian scheme for advection terms to control approximation accuracy [0.>q>1.; 0.95]

For model 10699 the #NUMERICS keywords are no longer valid. These must be replaced by the keywords:

```
#NUMERICS_PRESSURE
#NUMERICS_SATURATION
#NUMERICS_CONCENTRATION
#NUMERICS_TEMPERATURE
#NUMERICS_GEOELECTRIC
#NUMERICS_DEFORMATION
#NUMERICS_VELOCITY
#NUMERICS_ITERATION
```

Each of these keywords can contain a number of optional subkeywords (labeled by a $ sign), which are explained in the following:

#NUMERICS_PRESSURE			
Parameter	RF-Variable	Values	Meaning
$METHOD			
N1	.numerical_-method	[int]	Numerical method 0: Constant field 1: Bubnov−Method (Standard FEM) 2: Not used 3: Transport: Euler−Langrange−Method (fuer Advektion 2D) 4: Transport: Langrange−Method nur Advektion 2D 5: Transport: Langrange−Method nur Advektion 2D, Euler fuer 1D/3D 6: Saettigung: p_cap replaced by Saturation 7: Druck/Saettigung: Richards−Approximation 8: Druck/Saettigung: Celia−Richards−Approximation 9: Transport: FEM with artificial diffusion 10: Druck/Saettigung: Saturation replaced by p_cap 11: Transport: FEM with artificial diffusion
if N1=3 ,4 ,5			
N2	.minimalquality	[double]	Minimalquality [0.>q>1.; 0.95]
N3	.maximal_-steps	[int]	Maximal steps [i>0; 1000]
N4	.local_eps	[double]	Local epsilon [f>0.; 1.e−3]
N5	.timeweighting	[int]	Timeweighting [i=0,1; 0]
N6	.minimalweight	[double]	Minimalweight [f>0.; 1.e−3]
N7	.use_matrix	[int]	Use matrix for stationary field [i=0,1; 0]
N8	.velocity_-calculation	[int]	Velocity calculation method [i=0,1,2; 2]
if N1=9			
N9	.method_-param_0	[double]	Diffusion−Threshold [>0.]
N10	.method_-param_1	[double]	Art. Diffusion [>1.; 10.]
$GAUSS_POINTS			
N11	.gaussian_-points	[int]	Nr of gauss points integration [1−4;3]
$IRR_NODES_CORRECTION			
N12	.irr_nodes_-correction	[int]	Korrektur der Geschw.berechnung an irr. Knoten
$NONLINEAR_COUPLING			
N13	.nonlinear_-coupling	[int]	Nonlinear coupling, specifies whether the regarded quantity shall be regarded within the loop for non−linear iterations. 0: schwach 1: streng

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
$TRANSPORT_IN_PHASE			
N14	.transport_-in_phase	[int]	Specifies the number of the fluid phase for which transport processes shall be regarded. −1: all phases are regarded
$PLASTIC_DEFORMATION			
N15	.plastic_-deformation	[int]	Plastic deformation
$ITERATION_METHOD			
N16	.iteration_-method	[int]	Iteration method
$REFERENCE_PRESSURE_METHOD			
N17	.reference_-pressure_-method	[int]	Reference pressure method, specifies which phase pressure shall be used as reference in multiphase flow calculations.
$CALC_OTHER_SATURATION_METHOD			
N18	.calc_other_-saturations_-method	[int]	Specifies the method to calculate phase saturations in multiphase flow −1: All phase saturations are modified, so that they sum up to unity >= 0: The saturation of this phase will be computed from the other phase saturations. It is recommended to use the index of the least moveable phase.
$UPWINDING			
N19	.upwind_-method	[int]	Upwinding method
if N19=1 ,2			
N20	.upwind_-parameter	[double]	Upwind−Parameter [0.−1.;1.]
$MASS_LUMPING			
N21	.mass_-lumping_-method	[int]	Masslumping method, specifies whether masslumping shall be used to reduce wiggles. This can be very useful for multiphase flow calculations or transport calculations with very steep fronts. Compare THORENZ (2001).
if N21=2			
N22	.mass_-lumping_-parameter	[double]	Masslumping parameter
$PREDICTOR			
N23	.predictor_-method	[int]	Prediktor method, specifies that a linear predictor shall be used to estimate values for the next timestep 1.0: full prediction is used.

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
if N23=1			
N24	.predictor_-parameter	[double]	Prediktor parameter

$RELAXATION			
N25	.relaxation_-method	[int]	Relaxation method, specifies a scheme to reduce the computed correction in non−linear iterations. 0: No relaxation 1: Fixed relaxation 2: Dynamic relaxation on the basis of field oscillations (parameter recommended: 1. 0.25 0.5 0.5) 3: Dynamic relaxation on the basis of single node oscillations (parameters recommended: 1. 0.5)
if N25=0			No relaxation
if N25=1			Fixed relaxation
N26	.relaxation_-parameter_0	[double]	Relaxation parameter [0<=x<1]
if N25=2			Dynamic relaxation on the basis of field oscillations (parameter recommended: 1. 0.25 0.5 0.5)
N27	.relaxation_-parameter_0	[double]	Relaxation parameter[0] [0<=x<1]
N28	.relaxation_-parameter_1	[double]	Relaxation parameter[1] [0<=x<1]
N29	.relaxation_-parameter_2	[double]	Relaxation parameter[2] [0<=x<1]
N30	.relaxation_-parameter_3	[double]	Relaxation parameter[3] [0<=x<1]
if N25=3			Dynamic relaxation on the basis of single node oscillations (parameters recommended: 1. 0.5)
N31	.relaxation_-parameter_0	[double]	Relaxation parameter[0] [0<=x<1]
N32	.relaxation_-parameter_1	[double]	Relaxation parameter[1] [0<=x<1]

$EXTRACT_VALUES_FROM_PDE			
N33	.val_-extraction_-method	[int]	The keyword is only valid for the pressure field. Specifies whether the results of the last timestep are extracted from the system of linear equations while assembling it. For systems with very large vertical extend this procedure reduces 'small differences of large numbers' accuracy problems. (0 or 1)

$OSCILLATION_DAMPING			
N34	.oscil_damp_-method	[int]	Oscillation damping method 0: No damping 1: min max 2: Element(max−min) 3: Element(max−min)/Total(max−min) 4: min max Element(max−min) 5: min max Element(max−min)/Total(max−min)
if N34=0			No damping

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
if N34=1			min max
N35	.oscil_damp_-parameter_0	[double]	min
N36	.oscil_damp_-parameter_1	[double]	max
if N34=2			Element(max−min)
N37	.oscil_damp_-parameter_0	[double]	Element(max−min)
if N34=3			Element(max−min)/Total(max−min)
N38	.oscil_damp_-parameter_0	[double]	Element(max−min)/Total(max−min)
if N34=4			min max Element(max−min)
N39	.oscil_damp_-parameter_0	[double]	min
N40	.oscil_damp_-parameter_1	[double]	max
N41	.oscil_damp_-parameter_2	[double]	Element(max−min)
if N34=5			min max Element(max−min)/Total(max−min)
N42	.oscil_damp_-parameter_0	[double]	min
N43	.oscil_damp_-parameter_1	[double]	max
N44	.oscil_damp_-parameter_2	[double]	Element(max−min)/Total(max−min)

$TIMESTEPPING			
N45	.name	[string]	Timestepping name [Courant, Neumann, ...]
N46	.method	[int]	Timestepping method

$TIMECOLLOCATION			
N47	.name	[string]	Timecollocation name
N48	.method	[int]	Timecollocation method
if N48=0			
N49	.param_0	[double]	Collocation [0.<t<1.]

$MATRIX_REBUILD			
N50	.name	[string]	Matrixrebuild name [MOBILITY, DENSITY, VELOCITY, ...]
N51	.method	[int]	Matrixrebuild method 1: Absolut: fabs(x−x_ref) < param0 2: Relativ: 2*fabs(x−x_ref)/fabs(x+x_ref) < param0 3: Gemischt: x < param0 2*fabs(x−x_ref)/fabs(x+x_ref) < param1 11: Absolut: fabs(x−x_ref) < param0 12: Relativ: 2*fabs(x−x_ref)/fabs(x+x_ref) < param0 13: Gemischt: x < param0 2*fabs(x−x_ref)/fabs(x+x_ref) < param1 101: Gemischt: x < param0 2*fabs(x−x_ref)/fabs(x+x_ref) < param1
if N51=1			Absolut: fabs(x−x_ref) < param0

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N52	.param_0	[double]	Parameter[0]
if N51=2			Relativ: 2*fabs(x−x_ref)/fabs(x+x_ref) < param0
N53	.param_0	[double]	Parameter [0]
if N51=3			Gemischt: x < param0 2*fabs(x−x_ref)/fabs(x+x_ref) < param1
N54	.param_0	[double]	Parameter[0]
N55	.param_1	[double]	Parameter[1]
if N51=11			Absolut: fabs(x−x_ref) < param0
N56	.param_0	[double]	Parameter[0]
N57	.param_1	[double]	Parameter[1]
if N51=12			Relativ: 2*fabs(x−x_ref)/fabs(x+x_ref) < param0
N58	.param_0	[double]	Parameter[0]
N59	.param_1	[double]	Parameter[1]
if N51=13			Gemischt: x < param0 2*fabs(x−x_ref)/fabs(x+x_ref) < param1
N60	.param_0	[double]	Parameter[0]
N61	.param_1	[double]	Parameter[1]
N62	.param_2	[double]	Parameter[2]
if N51=101			Gemischt: x < param0 2*fabs(x−x_ref)/fabs(x+x_ref) < param1
N63	.curve	[int]	Curve [int; 0]
N64	.phase	[int]	Phase [int; −1]
N65	.comp	[int]	Component [int; −1]
N66	.group	[int]	Group [int; −1]
N67	.param_0	[double]	Parameter[0]
if N51=102			
N68	.curve	[int]	Curve [int; 0]
N69	.phase	[int]	Phase [int; −1]
N70	.comp	[int]	Component [int; −1]
N71	.group	[int]	Group [int; −1]
N72	.param_0	[double]	Parameter[0]
if N51=103			
N73	.curve	[int]	Curve [int; 0]
N74	.phase	[int]	Phase [int; −1]
N75	.comp	[int]	Component [int; −1]
N76	.group	[int]	Group [int; −1]
N77	.param_0	[double]	Parameter[0]
N78	.param_1	[double]	Parameter[1]

$INTEGRATION_POINTS			
N79	.name	[string]	integrationpoints name [DENSITY, VISCOSITY, ...]
N80	.method	[int]	integrationpoints method [0: Mittelpunkt, 1: Gausspunkte

Additional information to subkeywords:

$METHOD

.numerical_method:

3 For transport calculations: Use Lagrangian scheme for advective transport in 2D-elements (fractures). Compute everything else by FEM. Requires additional parameters.

4 For transport calculations: Use Lagrangian scheme for 2D-elements, omit everything else. Requires additional parameters.

5 For transport calculations only. Use Lagrangian scheme for 2D-elements. Requires additional parameters.

7 For unsaturated flow (pressure and saturation): The Richards' formulation is used to compute pressure and saturation fields.

8 For unsaturated flow (pressure and saturation): The Celia-Richards' formulation is used to compute pressure and saturation fields.

9 For transport calculations only. Finite element calculation with artificial diffusion in critical regions. Reads additionally two double values. The first parameter p1 specifies the concentration difference which marks an element as critical. If the parameter is negative, concentrations will be normalised with the span of all concentrations in the system before comparison (recommended value is –0.25, i.e. the front should smoothed it is less than four elements wide.). The second parameter p2 specifies the additional diffusion multiplier. The diffusion multiplier is computed from $p2^{\frac{c\_max\_min - p1}{p1}}$, where c_max_min is the (normalised) concentration difference within the regarded element.

11 Same as 9.

$UPWINDING

The keyword is followed by an integer value which specifies the upwinding scheme.

For the pressure field matrices:

1 Upwinding of Gaussian integration points (scaled) , additional parameter (double) for upwinding factor is required. Compare THORENZ (2001).

2 Upwinding of Gaussian integration points (unscaled) , additional parameter (double) for upwinding factor is required. Compare THORENZ (2001).

For the concentration and temperature field matrices:

1 Streamline upwinding, additional parameter (double) for upwinding factor is required.

2 Streamline upwinding applied to advection matrices only, additional parameter (double) for upwinding factor is required. Compare THORENZ (2001).

For the saturation field matrices:

1 Streamline upwinding, additional parameter (double) for upwinding factor is required.

9.2 Examples

Input example 1: eindimensionaler advektiv-dispersiver Wärmetransport (tutorial_a-b-c/htm1d.rfd)

```
#NUMERICS
 0              ; numerical method
 PRESSURE       ; name of unknown variable
 2              ; gauss points
 1.0            ; time collocation
 0.0            ; upwind parameter
 0              ; lagrange methode
 0.0            ; quality parameter

#NUMERICS
 1              ; numerical method
 TRANSPORT      ; name of unknown variable
 2              ; gauss points
 0.5            ; time collocation
 0.0            ; upwind parameter
 0              ; lagrange methode
 0.0            ; quality parameter

#NUMERICS
 1              ; numerical method
 TEMPERATURE0   ; name of unknown variable
 2              ; gauss points
 0.6            ; time collocation
 0.0            ; upwind parameter
 0              ; lagrange methode
 0.0            ; quality parameter
```

Input example 2: (thm_plus_hm/embankment2d.rfd)

```
#NUMERICS_DEFORMATION
 $METHOD               0    ; FEM
 $ITERATION_METHOD     1    ; 0:Picard   1:Newton
 $PLASTIC_DEFORMATION  0
 $TIMECOLLOCATION
   &GLOBAL             0 1. ;  Voll implizit

#NUMERICS_PRESSURE
 $METHOD               0    ; FEM
 $TIMECOLLOCATION
   &GLOBAL             0 1. ;  Voll implizit

#NUMERICS_SATURATION
 $METHOD               7    ; Richards-Approximation
 $TIMECOLLOCATION
   &GLOBAL             0 1. ;  Voll implizit

#NUMERICS_CONCENTRATION
 $METHOD               0    ; FEM
 $TIMECOLLOCATION
   &GLOBAL             0 1. ;  Voll implizit
```

10 Renumber

Node renumbering technique is aimed to reduce the bandwidth of system matrices. This will improve the efficiency of equation solvers.

10.1 Keyword Description

#RENUMBER

#RENUMBER			
Parameter	RF-Variable	Values	Meaning
N1	.umnummerierer	[int]	Knotenumnummerierer (umnummerierer) [0,1;0] 0: Nur Loecher beseitigen 1: Cuthill McKee 2: Gibbs Pole Stockmeyer (GPS)
if N1=2			
N2	.umnummerier_laeufe	[int]	Nur bei Gibbs Pole Stockmeyer: -1 : Unbegrenzte Durchlaeufe im Nachlauf >0 : Begrenzte Durchlaeufe im Nachlauf

10.2 Examples

Input example 1:

```
#RENUMBER
 1 ; Cuthill-McKee method

#RENUMBER
 2 -1 ; Gibbs-Pole-Stockmeyer method with default criterion for iterations

#RENUMBER
 2 10 ; Gibbs-Pole-Stockmeyer method with 10 iterations
```

11 Solver

Solving linear equation systems arising from the finite element discretizations is a crucial and the most time consuming point in finite element analysis of PDE. Rockflow provides several most popular and efficient linear solvers. Nonlinear PDEs results in corresponding nonlinear equation system, which have to be linearized. Nonlinear problems of interest are: Forchheimer flow, gas flow, multi-phase flow, density driven flow, reactive transport, nonlinear deformations and so on. Two nonlinear solvers, methods to linearize the non-linear equation system, are available in Rockflow. They are Picard fix point method and Newton-Raphson method.

11.1 Nonlinear Solver

11.1.1 Keyword Description

The following keywords can be used to choose the nonlinear solver for corresponding problem:

```
#NONLINEAR_SOLVER_PROPERTIES_PRESSURE
#NONLINEAR_SOLVER_PROPERTIES_SATURATION
#NONLINEAR_SOLVER_PROPERTIES_CONCENTRATION
#NONLINEAR_SOLVER_PROPERTIES_DISPLACEMENT
```

Data specifying the property of the chosen solve have to be followed after the corresponding key word and separated by blank space. The data appear in the order as in NONLINEAR_SOLVER_PROPERTIES_-PRESSURE exemplary shown:

#NONLINEAR_SOLVER_PROPERTIES_PRESSURE			
Parameter	RF-Variable	Values	Meaning
N1	.type	[int]	Typ wird immer = 1 gesetzt!
N2	.maxiter	[long]	Maximum number of solver iterations [>=0;1000] (only iterative solver)
N3	.criterium	[long]	Type of error calculation [1,2;1] 1: difference in result vector 2: difference in residuum 4: residuum < (eps_abs + eps_rel * RHS)
N4	.abs_eps	[double]	Absolute error tolerance for non−linear solver iterations [>0.0;1.E−9]
N5	.rel_eps	[double]	Relative error tolerance for non−linear solver iterations [>0.0;1.]
N6	.assemble	[long]	Number of iterations before reassembling the system of equation
N7	.rel_cg_eps	[double]	Combined error tolerance for non−linear solver iterations respect to linear solver iterations [>0.0;1.e−3] (only Absolute error tolerance=0. !)
N8	.kind	[int]	Validity time model [0]
if N8=2			
N9	.time	[double]	Validity time [−1]

11.1.2 Example

Input example 1: (flow_multiphase/buckley1d.rfd)

```
#NONLINEAR_SOLVER_PROPERTIES_PRESSURE
 1               ; method
 10              ; maximum iterations
 3               ; criterium,
 1.000000e-003   ; absolute eps
 0.              ; relative eps
 1               ; re-assembling,
 0.000000e+000   ; adaptive eps
 0               ; time control
```

11.2 Linear Solver

Direct and iterative solvers are listed in the following box with their indeces.

1: SpGAUSS	2:SpBICGSTAB	3:SpBICG	4:SpQMRCGSTAB	5: SpCG
6: SpCGNR	7:CGS	8: SpRichard	9:SpJOR	10:SpSOR

Only the first one is direct solver.

Using pre-conditioners, which is an approximate solution of the defect equation of the linear equation solver, is very helpful to accelerate the convergence of the iterative solver. There are three kinds of pre-conditioners available in Rockflow such as Jacobi, incomplete LU decomposition (ILU) and SOR splitting of system matrix method.

11.2.1 Keyword Description

The following keywords can be used to choose the linear solver for corresponding problem:

```
#LINEAR_SOLVER_PROPERTIES_PRESSURE
#LINEAR_SOLVER_PROPERTIES_SATURATION
#LINEAR_SOLVER_PROPERTIES_CONCENTRATION
#LINEAR_SOLVER_PROPERTIES_SORBED_CONCENTRATION
#LINEAR_SOLVER_PROPERTIES_TEMPERATURE
#LINEAR_SOLVER_PROPERTIES_DISPLACEMENT
#LINEAR_SOLVER_PROPERTIES_IMMOBILE_SOLUTE_CONCENTRATION
```

The data of solver property following the keyword is in the order as

#LINEAR_SOLVER_PROPERTIES_PRESSURE			
Parameter	RF-Variable	Values	Meaning
N1	.type	[int]	Solver method 1: Sp GAUSS Gaussian (direct) 2: Sp BICGSTAB (default for transport) 3: Sp BICG 4: Sp QMRCGSTAB 5: Sp CG (default for flow) 6: Sp CGNR 7: CGS 8: Sp Richard 9: Sp JOR 10: Sp SOR 11: AMG1R5 12: UMF
if N1=2 ,3 ,4 ,5 ,6 ,7 ,11 ,12			
N2	.nom	[long]	Error norm 0: Maximum norm 1: Unity norm 2: Euclidean norm
N3	.precond	[long]	Preconditioning 0: No preconditioning 1: Diagonal preconditioning 10: Ergebnis−Extraktion 100: ILU
N4	.maxiter	[long]	Maximum number of solver iterations [>=0;1000] (only by iterative solvers)
N5	.repeat	[long]	Number of repeated solving [>=0;1000] (only by iterative solvers)

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N6	.criterium	[long]	Type of error calculation: [0..5;2] 0: absolute error 1: relative error with respect to the RHS vector b0 2: relative error with respect to the residuum r0 3: variable error (if r0< 1 then 0 else 2) 4: relative error with respect to actual residuum r/x 5: rel.to MAX(r0,b,x0)
N7	.eps	[double]	Error tolerance for equation solver iterations [>0.0;1.E−9]
if N1=8 ,9 ,10			
N8	.store	[long]	Matrix storage concept [1,2,4;2] 1: full matrix 2: sparse matrix (nur A[i,j]!=0.0 werden gespeichert) 4: sparse matrix not symmetric (nur A[i,j]!=0.0 werden gespeichert)
N9	.nom	[long]	Error norm
N10	.precond	[long]	Preconditioning 0: No preconditioning 1: Diagonal preconditioning 10: Ergebnis−Extraktion 100: ILU
N11	.maxiter	[long]	Maximum number of solver iterations [>=0;1000] (only by iterative solvers)
N12	.repeat	[long]	Number of repeated solving [>=0;1000] (only by iterative solvers)
N13	.criterium	[long]	Type of error calculation: [0..5;2] 0: absolute error 1: relative error with respect to the RHS vector b0 2: relative error with respect to the residuum R0 3: variable error (if r0< 1 then 0 else 2) 4: relative error with respect to actual residuum r/x 5: rel.to MAX(r0,b,x0)
N14	.eps	[double]	Error tolerance for equation solver iterations [>0.0;1.E−9]
N15	.theta	[double]	Relaxation factor for Richardson,JOR,and SOR solver [>0.0;1.0] (Einfluss nur bei iterativen Loesern: Richardson,JOR,SOR) JOR−Verfahren mit: Faktor 0 < theta < 1 −−> gedaempfte Jacobi−Verfahren Faktor theta = 1 −−> Jacobi oder Gesamtschrittverfahren Faktor theta > 1 −−> Jacobi Ueberrelaxation SOR−Verfahren mit: Faktor 0 < theta < 1 −−> sukzessive Unterrelaxation Faktor theta = 1 −−> Gauss−Seidel oder Einzelschrittverfahren Faktor theta > 1 −−> sukzessive Ueberrelaxation
N16	.kind	[int]	kind
if N16=0			

continued

Continuation from the previous page

Parameter		RF-Variable	Values	Meaning
	N17	.store	[long]	Matrix storage concept [1,2,4;2] 1: full matrix 2: sparse matrix (nur A[i,j]!=0.0 werden gespeichert) 4: sparse matrix not symmetric (nur A[i,j]!=0.0 werden gespeichert)
if N16=2				
	N18	.time	[double]	Gueltigkeitszeit

11.2.2 Example

Input example 1: (tutorial_d/beam-TH-2d.rfd)

```
#LINEAR_SOLVER_PROPERTIES_DISPLACEMENT
 2                                  ; method
 0                                  ; norm
 101                                ; preconditioning
 1000000                            ; maximum iterations
 0                                  ; repeating
 0                                  ; criterium
 1.0e-010                           ; absolute error
 0                                  ; kind
 4                                  ; matrix storage technique
```

11.3 Iteration Properties

11.3.1 Keyword Description

```
#ITERATION_PROPERTIES_PRESSURE
#ITERATION_PROPERTIES_SATURATION
#ITERATION_PROPERTIES_CONCENTRATION
#ITERATION_PROPERTIES_SORBED_CONCENTRATION
#ITERATION_PROPERTIES_IMMOBILE_SOLUTE_CONCENTRATION
```

#ITERATION_PROPERTIES_PRESSURE				
Parameter		RF-Variable	Values	Meaning
N1		.type	[int]	iteration method
if N1=0				
	N2	.norm	[int]	norm
	N3	.maxiter	[long]	Max. iterations
	N4	.abs_eps	[double]	Abs error tolerance
	N5	.kind	[long]	Time control
if N1=1				
	N6	.norm	[int]	norm
	N7	.maxiter	[long]	Max. iterations
	N8	.rel_eps	[double]	Rel. error tolerance
	N9	.kind	[long]	Time control
if N1=2				
	N10	.norm	[int]	norm
	N11	.maxiter	[long]	Max. iterations
	N12	.rel_eps	[double]	Eps. start
	N13	.end_eps	[double]	Eps. final
	N14	.fac_eps	[double]	Factor var. epsilon
	N15	.exp_eps	[double]	Exp. var. epsilon
	N16	.kind	[long]	Time control

11.4 Example

Input example 1: Reaktiver Stofftransport (mass_transport/decay1d.rfd)

```
#ITERATION_PROPERTIES_CONCENTRATION
 0               ; iteration method
 0               ; norm
 100             ; max iterations
 1.000000e-006   ; abs error tolerance
 0               ; time control
```

12 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.

12.1 Keyword Description

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_TEMPERATURE
#INITIAL_CONDITIONS_DISPLACEMENT_X
#INITIAL_CONDITIONS_DISPLACEMENT_Y
#INITIAL_CONDITIONS_DISPLACEMENT_Z
#INITIAL_CONDITIONS_STRESS_XX
#INITIAL_CONDITIONS_STRESS_YY
#INITIAL_CONDITIONS_STRESS_ZZ
#INITIAL_CONDITIONS_STRESS_XY
#INITIAL_CONDITIONS_STRESS_YZ
#INITIAL_CONDITIONS_STRESS_XZ
```

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.

#INITIAL_CONDITIONS_PRESSURE

Parameter	RF-Variable	Values	Meaning
N1	.type	[int]	Type [>=0;0] 0: Constant field 1: Value for individual nodes given by node number first node 2: Value for individual nodes given by coordinates 3: Linear value distribution between two given nodes: First node, second node, count of nodes in between 4: Linear value distribution on a line given by coordinates 5: Hydrostatic distribution to all nodes by pressure 6: Hydrostatic distribution to all nodes by z−coordinate 7: Block entries by node number 8: Block entries by node coordinates 9: Constant value at a plain given by coordinates 10: Lineare Verteilung ueber die Hoehe z (Eingabe in x, y und z) 11: Ebenenzuweisung (Rechteck)) 18: Wert in ebenem Polygon (mit Epsilonumgebung)
N2	.mode	[int]	Mode [0,1;0] 0 : Overwrite 1 : Superimpose

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
if N1=0			Constant field
N3	.values_0	[double]	Constant value to all nodes [;1]
if N1=1			Value for individual nodes given by node number first node
N4	.begin_node	[long]	Number begin node [>=0;−1]
N5	.values_0	[double]	Node value [;1]
if N1=2			Value for individual nodes given by coordinates
N6	.x_0	[double]	Coordinate x[0]
N7	.y_0	[double]	coordinate y[0]
N8	.z_0	[double]	coordinate z[0]
N9	.values_0	[double]	Node value [;1]
if N1=3			Linear value distribution between two given nodes: First node, second node, count of nodes in between
N10	.begin_node	[long]	number begin node [>=0;−1]
N11	.end_node	[long]	number end node [>=0;−1]
N12	.step_nodes	[long]	step_nodes [>=0;1]
N13	.values_0	[2* double]	Node values [;1]
if N1=4			Linear value distribution on a line given by coordinates
2 * Loop			
N14	.x_0	[double]	Coordinate x[0]
N15	.y_0	[double]	coordinate y[0]
N16	.z_0	[double]	coordinate z[0]
N17	.values_0	[double]	Node value [0]
if N1=5			Hydrostatic distribution to all nodes by pressure
N18	.z_0	[double]	Coordinate z [0] [>=0;−1]
N19	.values_0_1	[2* double]	Node values [;1]
if N1=6			Hydrostatic distribution to all nodes by z−coordinate
N20	.z_0	[double]	Coordinate z [0] [>=0;−1]
N21	.values_0	[double]	Node value [;1]
if N1=7			Block entries by node number
N22	.distribution_type	[long]	distribution_type [>=0;−1] 0 : Wuerfel mit konsantem Wert 1 : Diamant mit konsantem Wert 2 : Kugel mit konsantem Wert −1 : Eiswaffel mit veraenderlichem Wert −2 : Kugel mit veraenderlichem Wert −3 : Gauss−Verteilung −4 : Zylinder
N23	.begin_node	[long]	Number begin node [>=0;−1]
N24	.radius	[double]	Radius [>=0;−1]
N25	.values_0	[double]	Node value [;1]
if N1=8			Block entries by node coordinates
N26	.distribution_type	[long]	Distribution_type [>=0;−1] 0 : Wuerfel mit konsantem Wert 1 : Diamant mit konsantem Wert 2 : Kugel mit konsantem Wert −1 : Eiswaffel mit veraenderlichem Wert −2 : Kugel mit veraenderlichem Wert −3 : Gauss−Verteilung −4 : Zylinder
N27	.x_0	[double]	Coordinate x[0]
N28	.y_0	[double]	Coordinate y[0]

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N29	.z_0	[double]	Coordinate z[0]
N30	.radius	[double]	Radius [>=0;−1]
N31	.values_0	[double]	Node value [;1]
if N1=9			Constant value at a plain given by coordinates
3 * Loop			
N32	.x_0	[double]	Coordinate x[0]
N33	.y_0	[double]	Coordinate y[0]
N34	.z_0	[double]	Coordinate z[0]
N35	.values_0	[double]	Node value [;1]
if N1=10			Lineare Verteilung ueber die Hoehe z (Eingabe in x, y und z)
N36	.count_of_-points	[long]	Count of points np [>0;−1]
N36 * Loop			
N37	.z_0	[double]	Coordinate z[0]
N38	.values_0	[double]	Node value [0]
if N1=11			Ebenenzuweisung (Rechteck))
4 * Loop			
N39	.x_0	[double]	Coordinate x[0]
N40	.y_0	[double]	coordinate y[0]
N41	.z_0	[double]	coordinate z[0]
N42	.values_0	[double]	Node value [0]
if N1=18			Wert in ebenem Polygon (mit Epsilonumgebung)
N43	.count_of_-points	[long]	Count of points [>=2]
N43 * Loop			
N44	.x_0	[double]	Coordinate x[0] (expr)
N45	.y_0	[double]	Coordinate y[0] (expr)
N46	.z_0	[double]	Coordinate z[0] (expr)
N47	.radius	[double]	Radius: Abstand orthogonal zur Polygonebene (expr)
N48	.epsilon	[double]	Epsilon: Geometrische Toleranz (expr)
N49	.values_0	[double]	Node value, value for polygon area [;1] (expr)

12.2 Examples

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

```
#INITIAL_CONDITIONS_PRESSURE
 0  0  1.e5  ; method, mode, value
#INITIAL_CONDITIONS_CONCENTRATION
 0  0  0.    ; method, mode, value
```

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

```
#INITIAL_CONDITIONS_SATURATION
 1  0  0  0.15  ; method, mode, node, value
```

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

```
#INITIAL_CONDITIONS_PRESSURE
 2   0    ; method, mode
 100 1 0  ; x, y, z
 1.e5     ; 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.

```
#INITIAL_CONDITIONS_PRESSURE
 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.

```
#REFERENCE_CONDITIONS
 9.810000  0.000000  101325.000000
#INITIAL_CONDITIONS_CONCENTRATION
 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:
$value = reference\_value + (node\_elevation - reference\_elevation) * gradient$

```
#INITIAL_CONDITIONS_PRESSURE
 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 !!!!!!!

```
#INITIAL_CONDITIONS_PRESSURE
 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 ???

```
#INITIAL_CONDITIONS_CONCENTRATION
 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 ???

```
#INITIAL_CONDITIONS_CONCENTRATION
 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.

```
#INITIAL_CONDITIONS_PRESSURE
 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:
$value = value1 + |node\_elevation - z1| * (value2 - value1) \quad / \quad |z2 - z1|$

```
#INITIAL_CONDITIONS_PRESSURE
 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.

```
#INITIAL_CONDITIONS_CONCENTRATION
 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
```

13 Boundary Conditions

13.1 Keyword Description

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

```
#BOUNDARY_CONDITIONS_PRESSURE
#BOUNDARY_CONDITIONS_HEAD
#BOUNDARY_CONDITIONS_SATURATION
#BOUNDARY_CONDITIONS_CONDITIONAL_SATURATION
#BOUNDARY_CONDITIONS_WATER_CONTENT
#BOUNDARY_CONDITIONS_CONCENTRATION
#BOUNDARY_CONDITIONS_CONDITIONAL_CONCENTRATION
#BOUNDARY_CONDITIONS_TEMPERATURE
#BOUNDARY_CONDITIONS_DISPLACEMENT_X
#BOUNDARY_CONDITIONS_DISPLACEMENT_Y
#BOUNDARY_CONDITIONS_DISPLACEMENT_Z
#BOUNDARY_CONDITIONS_FREE_OUTFLOW
#BOUNDARY_CONDITIONS_VOLTAGE_DC
#BOUNDARY_CONDITIONS_SORBED_CONCENTRATION
```

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

#BOUNDARY_CONDITIONS_PRESSURE

Parameter	RF-Variable	Values	Meaning
N1	.type	[int]	Type [>=0;0] 0: Einzelknotenzuweisung 1: Einzelknotenzuweisung 2: Linienzuweisung 3: Linienzuweisung 4: Ebenezuweisung (Eingabe in x, y und z) 5: Polygonzuweisung 6: Polygon−berandete Flaechenzuweisung 7: Ebenenzuweisung (Eingabe in x, y und z) Hydrostatische Druckverteilung 8: Polygon−berandete Flaechenzuweisung 9: Ebenenzuweisung (Rechteck) 10: Ebenenzuweisung (beliebige Ebene) 11: Punktzuweisung (mit Epsilonumgebung)) 12: Polygonzuweisung (mit Epsilonumgebung) 13: Polygonzuweisung (mit Epsilonumgebung) 14: Hydrostatischer Druck ueber Polygonzuweisung (mit Epsilonumgebung) 15: Hydrostatischer Druck ueber Polygonzuweisung (mit Epsilonumgebung) 16: Interpolierter Wert in ebenem Viereck (mit Epsilonumgebung) 17: Hydrostatischer Druck in ebenem Viereck (mit Epsilonumgebung) 18: Value in a plane convex polygone or right prism 100: Equal to type 0, additional expressions available 101: Equal to type 1, additional expressions available 102: Equal to type 2, additional expressions available

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N2	.mode	[int]	mode [0,1;0] mode=0 : Ueberschreibmodus mode=1 : Einfuegemodus
N3	.curve	[long]	Curve [0,1;0]
if N1=0			Einzelknotenzuweisung
N4	.begin_node	[long]	Node number begin node [>=0;−1]
N5	.values_0	[double]	Nodal value [;1]
if N1=1			Einzelknotenzuweisung
N6	.x_0	[double]	Coordinate x[0]
N7	.y_0	[double]	Coordinate y[0]
N8	.z_0	[double]	Coordinate z[0]
N9	.radius	[double]	Umgebungsradius [;1]
N10	.values_0	[double]	Node value [0] [;1]
if N1=2			Linienzuweisung
2 * Loop			
N11	.x_0	[double]	Coordinate x[0]
N12	.y_0	[double]	Coordinate y[0]
N13	.z_0	[double]	Coordinate z[0]
N14	.radius	[double]	Umgebungsradius (radius) [;1]
N15	.values_0_1	[2* double]	Nodal values (values[0], values[1]) [;1]
if N1=3			Linienzuweisung
N16	.begin_node	[long]	Node number begin node [>=0;−1]
N17	.end_node	[long]	Node number end node [>=0;−1]
N18	.step_nodes	[long]	Step nodes number [>=0;1]
N19	.values_0_1	[2* double]	Nodal values (values[0], values[1]) [;1]
if N1=4			Ebenezuweisung (Eingabe in x, y und z)
3 * Loop			
N20	.x_0	[double]	Coordinate x[0]
N21	.y_0	[double]	Coordinate y[0]
N22	.z_0	[double]	Coordinate z[0]
N23	.radius	[double]	Radius [;1]
N24	.values_0	[double]	Node value [0] [;1]
if N1=5			Polygonzuweisung
N25	.count_of_-points	[long]	Count of points [>=0;−1]
N26	.nodes	[N25* long]	Node numbers
N27	.values	[N25* double]	Node values
if N1=6			Polygon−berandete Flaechenzuweisung
N28	.count_of_-points	[long]	Count of points [>=0;−1]
N29	.nodes	[N28* long]	Node numbers
N30	.radius	[double]	Radius [;1]
N31	.values_0	[double]	Node value [0] [;1]
if N1=7			Ebenenzuweisung (Eingabe in x, y und z) Hydrostatische Druckverteilung
3 * Loop			
N32	.x_0	[double]	Coordinate x[0]
N33	.y_0	[double]	Coordinate y[0]

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N34	.z_0	[double]	Coordinate z[0]
N35	.radius	[double]	Radius [;1]
N36	.z_3	[double]	Reference elevation [>=0;−1]
N37	.values_0	[double]	Node value [0] [;1]
if N1=8			Polygon−berandete Flaechenzuweisung
N38	.count_of_-points	[long]	Count of points [>=0;−1]
N38 * Loop			
N39	.x_0	[double]	Coordinate x[0]
N40	.y_0	[double]	Coordinate y[0]
N41	.z_0	[double]	Coordinate z[0]
N42	.radius	[double]	Radius [;1]
N43	.values_0	[double]	Node value [0] [;1]
if N1=9			Ebenenzuweisung (Rechteck)
4 * Loop			
N44	.x_0	[double]	Coordinate x[0]
N45	.y_0	[double]	Coordinate y[0]
N46	.z_0	[double]	Coordinate z[0]
N47	.values_0	[double]	Node value [0] [;1]
if N1=10			Ebenenzuweisung (beliebige Ebene)
4 * Loop			
N48	.x_0	[double]	Coordinate x[0]
N49	.y_0	[double]	Coordinate y[0]
N50	.z_0	[double]	Coordinate z[0]
N51	.values_0	[double]	Node value [0] [;1]
if N1=11			Punktzuweisung (mit Epsilonumgebung))
N52	.x_0	[double]	Coordinate x[0]
N53	.y_0	[double]	Coordinate y[0]
N54	.z_0	[double]	Coordinate z[0]
N55	.values_0	[double]	Node value [0] [;1]
N56	.epsilon	[double]	Epsilon [>0.]
if N1=12			Polygonzuweisung (mit Epsilonumgebung)
N57	.count_of_-points	[long]	Count of points [>=2]
N57 * Loop			
N58	.x_0	[double]	Coordinate x[0]
N59	.y_0	[double]	Coordinate y[0]
N60	.z_0	[double]	Coordinate z[0]
N61	.values_0	[double]	Node value [0] [;1]
N62	.epsilon	[double]	Epsilon [>0.]
if N1=13			Polygonzuweisung (mit Epsilonumgebung)
N63	.count_of_-points	[long]	Count of points [>=2]
N63 * Loop			
N64	.nodes_0	[long]	node numbers [0] [>=0;−1]
N65	.values_0	[double]	Node value [0] [;1]
N66	.epsilon	[double]	Epsilon [>0.]
if N1=14			Hydrostatischer Druck ueber Polygonzuweisung (mit Epsilonumgebung)
N67	.values_0	[double]	reference height []
N68	.values_1	[double]	Reference pressure []
N69	.number_of_-fluid_phase	[long]	Number of fluid phase [0,...,] fuer Dichte verwenden

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N70	.count_of_-points	[long]	Count of points [>=2]
N70 * Loop			
N71	.x_0	[double]	Coordinate x[0]
N72	.y_0	[double]	Coordinate y[0]
N73	.z_0	[double]	Coordinate z[0]
N74	.epsilon	[double]	Epsilon [>0.]
if N1=15			Hydrostatischer Druck ueber Polygonzuweisung (mit Epsilonumgebung)
N75	.values_0	[double]	reference height []
N76	.values_1	[double]	Reference pressure []
N77	.number_-fluid_phase	[long]	Number fluid phase [0,...,] fuer Dichte verwenden
N78	.count_of_-points	[long]	Count of points [>=2]
N79	.nodes	[N78* long]	Node numbers
N80	.epsilon	[double]	Epsilon [>0.]
if N1=16			Interpolierter Wert in ebenem Viereck (mit Epsilon-umgebung)
4 * Loop			
N81	.x_0	[double]	Coordinate x[0]
N82	.y_0	[double]	Coordinate y[0]
N83	.z_0	[double]	Coordinate z[0]
N84	.values_0	[double]	Node value [0] [;1]
N85	.epsilon	[double]	Epsilon [>0.]
if N1=17			Hydrostatischer Druck in ebenem Viereck (mit Epsilon-umgebung)
N86	.values_0	[double]	Reference height []
N87	.values_1	[double]	Reference pressure []
N88	.nummber_-fluid_phase	[long]	Nummber fluid phase [0,...,] fuer Dichte verwenden
4 * Loop			
N89	.x_0	[double]	Coordinate x[0]
N90	.y_0	[double]	Coordinate y[0]
N91	.z_0	[double]	Coordinate z[0]
N92	.epsilon	[double]	Epsilon [>0.]
if N1=18			Value in a plane convex polygone or right prism
N93	.count_of_-points	[long]	Count of polygone points [>2]
N93 * Loop			
N94	.x_0	[double]	Coordinate x[0] (expr)
N95	.y_0	[double]	Coordinate y[0] (expr)
N96	.z_0	[double]	Coordinate z[0] (expr)
N97	.radius	[double]	Distance perpendicular to polygon−plane (expr) [;1] If only a 2−D polygone is needed, choose the value equal to epsilon (tolerance) This value defines the extent of the right prism, perpendicular to the polygone plane. If it is >0.0, it defines the extent in both directions of the polygone. If it is <0.0, it only defines the extent in negative direction of the polygone.
N98	.epsilon	[double]	geometric tolerance (expr) [>0.]

continued

Continuation from the previous page

Parameter		RF-Variable	Values	Meaning
	N99	.values_0	[double]	Node value [0] (expr) [;1]
if N1=33				
	N100	.x_0	[double]	Coordinate x[0]
	N101	.y_0	[double]	Coordinate y[0]
	N102	.z_0	[double]	Coordinate z[0]
	N103	.x_1	[double]	Coordinate x[1]
	N104	.y_1	[double]	Coordinate y[1]
	N105	.z_1	[double]	Coordinate z[1]
	N106	.radius	[double]	Radius [;1]
	N107	.values_0	[double]	Node value [0] [;1]
	N108	.values_1	[double]	Node value [1] [;1]
if N1=34				
	N109	.x_0	[double]	Coordinate x[0]
	N110	.y_0	[double]	Coordinate y[0]
	N111	.z_0	[double]	Coordinate z[0]
	N112	.x_1	[double]	Coordinate x[1]
	N113	.y_1	[double]	Coordinate y[1]
	N114	.z_1	[double]	Coordinate z[1]
	N115	.radius	[double]	Radius [;1]
	N116	.values_0	[double]	Node value [0] [;1]
	N117	.values_1	[double]	Node value [1] [;1]
if N1=20				
	N118	.polyline_-name	[string]	Polyline name
	N119	.dummy	[long]	Dummy
	N120	.values_0	[double]	Node value [0] [;1]
if N1=21				
	N121	.polyline_-name	[string]	Polyline name
	N122	.dummy	[long]	Dummy
	N123	.values_0	[double]	Node value [0] [;1]
if N1=100				Equal to type 0, additional expressions available
	N124	.begin_node	[long]	Begin node number [>=0;−1]
	N125	.values_0	[double]	Node value (expr) [;1]
if N1=101				Equal to type 1, additional expressions available
	N126	.x_0	[double]	Coordinate x[0] (expr)
	N127	.y_0	[double]	Coordinate y[0] (expr)
	N128	.z_0	[double]	Coordinate z[0] (expr)
	N129	.radius	[double]	Radius [;1]
	N130	.values_0	[double]	Node value [0] (expr) [;1]
if N1=102				Equal to type 2, additional expressions available
	2 * Loop			
	N131	.x_0	[double]	Coordinate x[0] (expr)
	N132	.y_0	[double]	Coordinate y[0] (expr)
	N133	.z_0	[double]	Coordinate z[0] (expr)
	N134	.radius	[double]	Radius [;1]
	N135	.values_0_1	[2* double]	Node values (values[0], values[1]) [;1]

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.

13.2 Examples

Input example 1: 2D Stroemungsmodell (tutorial_a-b-c/asm2d.rfd)

```
#BOUNDARY_CONDITIONS_PRESSURE
 2               ; type
 0               ; mode
 0               ; curve
 100.0  0.0  0.0 ; x0,y0,z0
 100.0  1.0  0.0 ; x1,y1,z1
 0.001           ; radius
 20000.0         ; value0
 20000.0         ; value1
```

Input example 2: (applications/d_iv_phase02.rfd)

```
#BOUNDARY_CONDITIONS_FREE_OUTFLOW
 18                           ; type    (18=polygon plane)
 0                            ; mode
 0                            ; curve
 3                            ; count_of_points
 ;  x       y       z         ; (gekürzt)
 0.0000  0.0000   3.5650
 0.4810  0.0000   3.5437
 0.9440  0.0000   3.4050
 0.1                          ; distance in y-direction (radius)
 0.01                         ; geometric tolerance parameter epsilon
 { 'BC_FREE_OUTFLOW_TUNNEL' } ; value {EXPRESSION}


#FUNCTIONS
 0 BC_FREE_OUTFLOW_TUNNEL           ; BC at the tunnel
 { CURVE 4 + CURVE 5 * 'p_kelvin' } ; = excavation phase + seasonal fluctuation
```

14 Source Terms

14.1 Keyword Description

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_MASS_TRACER_COMPONENT
#SOURCE_VOLUME_FLUID_PHASE
#SOURCE_HEAT_PHASE
#SOURCE_ELECTRIC_CURRENT_DC
#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_MASS_TRACER_COMPONENT
#SOURCE_VOLUME_FLUID_PHASE
```

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 s̈ingle nodeïnjections 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.

#SOURCE_MASS_FLUID_PHASE			
Parameter	RF-Variable	Values	Meaning
N1	.type	[int]	Type [>=0;0] 0: Einzelknotenzuweisung 1: Einzelknotenzuweisung 2: Linienzuweisung 3: Linienzuweisung 4: Ebenezuweisung (Eingabe in x, y und z) 6: Polygonzuweisung (mit Epsilonumgebung) 7: Polygonzuweisung (mit Epsilonumgebung) 8: Punktzuweisung (mit Epsilonumgebung) 9: Punktzuweisung mit Lastverteilung (mit Epsilonumgebung) 10: Punktzuweisung in allen Knoten 11: Polygonzuweisung (mit Epsilonumgebung) 12: Polygonzuweisung (mit Epsilonumgebung) 13: Polygonzuweisung (mit Epsilonumgebung) 106, 206: Wie Typ 6 (linear/quadratisch), mit zusaetzlichen Optionen: anstatt der erwarteten double−Werte bei den Parametern value, x, y, z und epsilon kann eine Expression−Nr. in der Form: { FUNCTION 1 } angegeben werden. Es wird dann diese Funktion (hier z.B. Nr 1) bei jedem Aufruf der Neumann−Randbedingung ausgewertet. Der Ausdruck muss von { } eingerahmt sein.
N2	.mode	[int]	Mode [0,1;0] 0: Overwrite 1: Superimpose
N3	.curve	[long]	Number of time curve [0,1;0]
if N1=0			Einzelknotenzuweisung
N4	.begin_node	[long]	Node number begin node [>=0;−1]
N5	.values_0	[double]	Nodal value [;1]
if N1=1			Einzelknotenzuweisung
N6	.x_0	[double]	Coordinate x[0]
N7	.y_0	[double]	Coordinate y[0]
N8	.z_0	[double]	Coordinate z[0]
N9	.values_0	[double]	node value [0] [;1]
if N1=2			Linienzuweisung
2 * Loop			
N10	.x_0	[double]	Coordinate x[0]
N11	.y_0	[double]	Coordinate y[0]
N12	.z_0	[double]	Coordinate z[0]
N13	.values_0	[double]	node value [0] [;1]
if N1=3			Linienzuweisung
2 * Loop			
N14	.x_0	[double]	Coordinate x[0]
N15	.y_0	[double]	Coordinate y[0]
N16	.z_0	[double]	Coordinate z[0]
N17	.values_0	[double]	Flux (distributed to all nodes along this line) [;1]
if N1=4			Ebenezuweisung (Eingabe in x, y und z)
3 * Loop			

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N18	.x_0	[double]	Coordinate x[0]
N19	.y_0	[double]	Coordinate y[0]
N20	.z_0	[double]	Coordinate z[0]
N21	.values_0	[double]	node value [0] [;1]
if N1=6			Polygonzuweisung (mit Epsilonumgebung)
N22	.count_of_-points	[long]	Number of nodes [>=2]
N22 * Loop			
N23	.x_0	[double]	Coordinate x[0]
N24	.y_0	[double]	Coordinate y[0]
N25	.z_0	[double]	Coordinate z[0]
N26	.epsilon	[double]	Geometric tolerance to catch nodes [>0.]
N27	.values_0	[double]	Value (to be integrated over polygone)
if N1=7			Polygonzuweisung (mit Epsilonumgebung)
N28	.count_of_-points	[long]	Number of nodes [>=2]
N29	.nodes	[N28* long]	Node numbers
N30	.epsilon	[double]	Geometric tolerance to catch nodes [>0.]
N31	.values_0	[double]	Value (to be integrated over polygone)
if N1=8			Punktzuweisung (mit Epsilonumgebung)
N32	.x_0	[double]	Node coordinate x
N33	.y_0	[double]	Node coordinate y
N34	.z_0	[double]	Node coordinate z
N35	.epsilon	[double]	Geometric tolerance to catch nodes [>0.]
N36	.values_0	[double]	Nodal value
if N1=9			Punktzuweisung mit Lastverteilung (mit Epsilonumgebung)
N37	.x_0	[double]	Node coordinate x
N38	.y_0	[double]	Node coordinate y
N39	.z_0	[double]	Node coordinate z
N40	.epsilon	[double]	Geometric tolerance to catch nodes [>0.]
N41	.values_0	[double]	Distribution function [>0.]
N42	.values_1	[double]	Nodal value
if N1=10			Punktzuweisung in allen Knoten
N43	.values_0	[double]	Transfer coefficient
if N1=11			Polygonzuweisung (mit Epsilonumgebung)
N44	.count_of_-points	[long]	Count of points [>=2]
N44 * Loop			
N45	.x_0	[double]	Coordinate x[0]
N46	.y_0	[double]	Coordinate y[0]
N47	.z_0	[double]	Coordinate z[0]
N48	.epsilon	[double]	Geometric tolerance to catch nodes [>0.]
N49	.values_0	[double]	value
if N1=12			Polygonzuweisung (mit Epsilonumgebung)
N50	.polyline_-name	[string]	Name of polyline
N51	.epsilon	[double]	Geometric tolerance to catch nodes [>0.]
N52	.values_0	[double]	value
if N1=13			Polygonzuweisung (mit Epsilonumgebung)
N53	.polyline_-name	[string]	Name of polyline
N54	.epsilon	[double]	Geometric tolerance to catch nodes [>0.]

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N55	.values_0	[double]	value
if N1=106 ,206			Wie Typ 6 (linear/quadratisch), mit zusaetzlichen Optionen: anstatt der erwarteten double−Werte bei den Parametern value, x, y, z und epsilon kann eine Expression−Nr. in der Form: { FUNCTION 1 } angegeben werden. Es wird dann diese Funktion (hier z.B. Nr 1) bei jedem Aufruf der Neumann−Randbedingung ausgewertet. Der Ausdruck muss von { } eingerahmt sein.
N56	.count_of_-points	[long]	Number of nodes [>=2]
N56 * Loop			
N57	.x_0	[double]	Coordinate x[0] (expr)
N58	.y_0	[double]	Coordinate y[0] (expr)
N59	.z_0	[double]	Coordinate z[0] (expr)
N60	.epsilon	[double]	Geometric tolerance to catch nodes [>0.]
N61	.values_0	[double]	Value (expr) (to be integrated over polygone)

#SINK_MASS_FLUID_MIXTURE

Parameter	RF-Variable	Values	Meaning
N1	.type	[int]	Type [>=0;0] 0: Einzelknotenzuweisung 1: Einzelknotenzuweisung 2: Linienzuweisung 3: Linienzuweisung 4: Ebenezuweisung (Eingabe in x, y und z)
N2	.mode	[int]	Mode [0,1;0]
N3	.curve	[long]	Curve [0,1;0]
if N1=0			Einzelknotenzuweisung
N4	.begin_node	[long]	Node number begin node [>=0;−1]
N5	.values_0	[double]	Node value [;1]
if N1=1			Einzelknotenzuweisung
N6	.x_0	[double]	Coordinate x[0]
N7	.y_0	[double]	Coordinate y[0]
N8	.z_0	[double]	Coordinate z[0]
N9	.values_0	[double]	node value [0] [;1]
N10	.values_0	[double]	node value [0] [;1]
if N1=2			Linienzuweisung
2 * Loop			
N11	.x_0	[double]	Coordinate x[0]
N12	.y_0	[double]	Coordinate y[0]
N13	.z_0	[double]	Coordinate z[0]
if N1=3			Linienzuweisung
2 * Loop			
N14	.x_0	[double]	Coordinate x[0]
N15	.y_0	[double]	Coordinate y[0]
N16	.z_0	[double]	Coordinate z[0]
N17	.values_0	[double]	Nodal value [;1]

continued

Continuation from the previous page

Parameter			RF-Variable	Values	Meaning
if N1=4					Ebenezuweisung (Eingabe in x, y und z)
	3 * Loop				
		N18	.x_0	[double]	Coordinate x[0]
		N19	.y_0	[double]	Coordinate y[0]
		N20	.z_0	[double]	Coordinate z[0]
	N21		.values_0	[double]	node value [0] [;1]

14.2 Example

Input example 1: (applications/d_iv_phase02.rfd)

```
#LOAD_SOLID_PHASE_X
 ;typ mode curve c_o_p   x0    y0   z0      x1     y1     z1    eps    val
  206  0     0     2    40.0  0.0  40.0    40.0   0.0   -40.0  0.01  -4.32e+6
```

15 Reference Conditions

15.1 Keyword Description

#REFERENCE_CONDITIONS			
Parameter	RF-Variable	Values	Meaning
N1	.gravity_-constant	[double]	Gravity constant g [9.,10.;9.81] [msˆ−2]
N2	.T_0	[double]	ReferenceTemperature T_0 [>0.;293.] [K]
N3	.p_reference	[double]	Reference Pressure p_0 [0.<p_0;101325.] [Pa]

$REFERENCE_TEMPERATURE			
N4	.temp_type	[int]	Reference temperature type [0,1,2;0] 0: T_0, defined above, is used 1: Reference temperature defined by curve−no. 2: Reference temperature set equal to current temperature at every node
if N4=0			T_0, defined above, is used
if N4=1			Reference temperature defined by curve−no.
N5	.temp_curve	[int]	Curve number to define reference temperature
if N4=2			Reference temperature set equal to current temperature at every node
N6	.temp_mode	[int]	Mode to specify times, at which T_0=T 0: Reference temperature set equal to current temperature at all times 1: Reference temperature set equal to current temperature at specified times 2: Reference temperature set equal to current temperature between specified times
if N6=0			Reference temperature set equal to current temperature at all times
if N6=1			Reference temperature set equal to current temperature at specified times
N7	.temp_nof_-times	[long]	Number of times, at which T_0=T
N8	.temp_times	[N7* double]	Time_1, time_2, ...
N9	.temp_time_-radius	[double]	Time radius (tolerance parameter)
if N6=2			Reference temperature set equal to current temperature between specified times
N10	.temp_nof_-times	[int]	Number of times, between which the reference temperature set equal to current temperature
N11	.temp_times	[N10* double]	Time_1_start, time_1_end, time_2_start, time_2_end, ...
N12	.temp_time_-radius	[double]	Time radius (tolerance parameter)

15.2 Example

Input example 1: TM-Coupling with reset of reference-temperature
(thm_plus_tm/reference_temperature_2d.rfd)

```
#REFERENCE_CONDITIONS
 9.81  273.0  101325.0  ;g, T_0, p_0

 $REFERENCE_TEMPERATURE
   ; Set node/time oriented reference temperature
   ; Initialized with T_0, defined above (#REFERENCE_CONDITIONS)
   ; (Applied in model 8460, deformation)

 2 ; type  0-> all nodes set to T_0 at all times
   ;       1-> curve value is used to set T_0 (same value at all nodes)
   ;       2-> reference temperature set to current temperature at every node (at specified time

   ; (if type==1)
   ;1  ; curve number

   ; (if type==2)
 1 ; mode   0-> every timestep
   ;        1-> at given times
   ;        2-> between given times

   ; (if mode==1 || mode==2)
 3 ; number of times
 2000000.0 4000000.0 6000000.0 ; times
 1.0 ; time_radius (lolerance parameter to catch times)
```

16 Fluid Properties

16.1 Keyword Description #FLUID_PROPERTIES

Fluid properties (gas or liquid) can be specified using the keyword:

#FLUID_PROPERTIES

By repeatedly use the data for several phases are specified.

#FLUID_PROPERTIES			
Parameter	RF-Variable	Values	Meaning
N1	.density_model	[int]	Density model [0,1;0] 0: Constant density 1: Linear density function 11: Model 11: (rho=rho_0+max(C,0.)*drho_dC+T*drho_dT) 15: Temperature dependent density for an ideal gas: rho=rho_0*T_0/T; valid for model 0699 16: Temperature and pressure dependent density for an ideal gas: rho=(P/T)*(M/R); valid for model 0699
N2	.rho_0	[double]	Fluid density (reference value) [rho_0=>0.0;1000.0] [kg/m3]
if N1=0			Constant density
if N1=1			Linear density function
N3	.drho_dp	[double]	Compressibility [d_rho/d_p>0.0;0.0] [1/Pa]
if N1=10			
N4	.drho_dC	[double]	Coefficient of density variation due to concentration change [m3 1/kg]
if N1=11			Model 11: (rho=rho_0+max(C,0.)*drho_dC+T*drho_dT)
N5	.drho_dC	[double]	Coefficient of density variation due to concentration change [m3 1/kg]
N6	.drho_dT	[double]	Coefficient of density variation due to temperature change
if N1=15			Temperature dependent density for an ideal gas: rho=rho_0*T_0/T; valid for model 0699
if N1=16			Temperature and pressure dependent density for an ideal gas: rho=(P/T)*(M/R); valid for model 0699
N7	.viscosity_model	[int]	Viscosity model [0,1;0] 0: Constant viscosity 1: Linear viscosity function 2: Linear viscosity function
N8	.my_0	[double]	Fluid viscosity (reference value) [my_0>0.0;1.e−3] [Pa s]
if N7=0			Constant viscosity
if N7=1			Linear viscosity function
N9	.dmy_dp	[double]	Coefficient of viscosity variation due to pressure change [d_my/d_p>0.0;0.0] [1/Pa 1/s]
if N7=2			Linear viscosity function
N10	.dmy_dp	[double]	Coefficient of viscosity variation due to pressure change [d_my/d_p>0.0;0.0] [1/Pa 1/s]
N11	.Z	[double]	Real gas factor [0<=Z<=1;0]

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N12	.heat_-capacity	[double]	Heat capacity of fluid [;4200] [J 1/kg 1/K]
N13	.heat_-conductivity	[double]	Heat conductivity of fluid (expr) [;0.6] [W 1/m 1/K]

$ELECTRIC_CONDUCTIVITY			
N14	.electric_-conductivity_-model	[int]	Electric conductivity model 0: Constant conductivity
if N14=0			Constant conductivity
N15	.electric_-conductivity_-val_0	[double]	Constant conductivity [>0.]
if N14=1			
N16	.electric_-conductivity_-val_0	[double]	Constant conductivity [>0.]
N17	.electric_-conductivity_-val_1	[double]	dConductivity/dConcentration [>0.]

16.2 Examples #FLUID_PROPERTIES

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
```

16.3 Keyword Description #FLUID_PROPERTIES_NEW

The following keywords have been introduced for fluid properties:

#FLUID_PROPERTIES_NEW			
Parameter	RF-Variable	Values	Meaning
$DENSITY			
N1	.density_-model	[int]	Density model [0,1;0] 0: Constant density 1: Linear density function 11: Model=11: rho=rho_0+Max(C,0.)*drho_-dC+T*drho_dT)

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
if N1=0			Constant density
N2	.rho_0	[double]	Fluid density [rho_0=>0.0;1000.0] [kg/m3]
if N1=1			Linear density function
N3	.rho_0	[double]	Fluid density [rho_0=>0.0;1000.0] [kg/m3]
N4	.drho_dp	[double]	Compressibility [d_rho/d_p>0.0;0.0]
N5	.reference_p	[double]	
if N1=4			
N6	.rho_0	[double]	Fluid density [rho_0=>0.0;1000.0] [kg/m3]
N7	.drho_dp	[double]	Compressibility [d_rho/d_p>0.0;0.0]
if N1=10			
N8	.rho_0	[double]	Fluid density [rho_0=>0.0;1000.0] [kg/m3]
N9	.drho_dC	[double]	Coefficient of density variation due to concentration change [m3 1/kg]
if N1=11			Model=11: rho=rho_0+Max(C,0.)*drho_dC+T*drho_dT)
N10	.rho_0	[double]	Fluid density [rho_0=>0.0;1000.0] [kg/m3]
N11	.drho_dC	[double]	Coefficient of density variation due to concentration change [m3 1/kg]
N12	.drho_dT	[double]	Coefficient of density variation due to temperature change
if N1=12			
N13	.rho_0	[double]	Fluid density [rho_0=>0.0;1000.0] [kg/m3]
N14	.T_0	[double]	T_0
N15	.drho_dT	[double]	Coefficient of density variation due to temperature change
if N1=13			
N16	.rho_0	[double]	Fluid density [rho_0=>0.0;1000.0] [kg/m3]
N17	.drho_dp	[double]	Compressibility [d_rho/d_p>0.0;0.0]
N18	.drho_dT	[double]	Coefficient of density variation due to temperature change
if N1=14			
N19	.rho_0	[double]	Fluid density [rho_0=>0.0;1000.0] [kg/m3]
N20	.reference_p	[double]	
N21	.drho_dp	[double]	Compressibility [d_rho/d_p>0.0;0.0]
N22	.T_0	[double]	T_0
N23	.drho_dT	[double]	Coefficient of density variation due to temperature change

$VISCOSITY			
N24	.viscosity_model	[int]	Viscosity_model [0,1;0] 0: Constant viscosity 1: Linear viscosity function
if N24=0			Constant viscosity
N25	.my_0	[double]	Fluid viscosity [my_0>0.0;1.e−3] [Pa s]
if N24=1			Linear viscosity function
N26	.my_0	[double]	Fluid viscosity [my_0>0.0;1.e−3] [Pa s]
N27	.dmy_dp	[double]	Compressibility [d_my/d_p>0.0;0.0] [1/Pa 1/s]

$HEAT_CAPACITY			
N28	.heat_capacity_model	[int]	Heat capacity model

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
if N28=0			
N29	.heat_-capacity	[double]	Heat capacity of fluid [;4200] [J 1/kg 1/K]

$HEAT_CONDUCTIVITY			
N30	.heat_-conductivity_-model	[int]	Heat conductivity model
if N30=0			
N31	.heat_-conductivity	[double]	Heat conductivity of fluid [;0.6] [W 1/m 1/K]

$ELECTRIC_CONDUCTIVITY			
N32	.electric_-conductivity_-model	[int]	Electric conductivity model 0: Constant conductivity
if N32=0			Constant conductivity
N33	.electric_-conductivity_-val_0	[double]	Constant electric conductivity [>0.]
if N32=1			
N34	.electric_-conductivity_-val_0	[double]	Constant electric conductivity [>0.]
N35	.electric_-conductivity_-val_1	[double]	dConductivity/dConcentration [>0.]

16.4 Examples #FLUID_PROPERTIES_NEW

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                ; desity model
  1000.            ; rho_0: reference density
 $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
```

17 Soil Properties

17.1 Keyword Description

Medium properties (soil or rock) can be specified by use of keyword

#SOIL_PROPERTIES

By repeatedly use data for several medium properties material groups are specified.

#SOIL_PROPERTIES			
Parameter	RF-Variable	Values	Meaning
N1	.dimension	[int]	Element dimension [1,2,3;1] 1: 1−D elements 2: 2−D elements
if N1=1			1−D elements
N2	.area	[double]	Exchange area [>0;1.0] [m2]
if N1=2			2−D elements
N3	.area	[double]	Element thickness [>0;1.0] [m]
N4	.porosity_-model	[int]	Porosity model [0,1,4;0] 0: Constant porosity 1: Saline solution 2: CSM swelling model 4: Porosity influenced by deformation
if N4=0			Constant porosity
N5	.porosity	[double]	(Initial) Porosity [>0;1.0]
if N4=1			Saline solution
N6	.porosity_-dependence_-model	[int]	Porosity dependence model 0: constant 1: concentration dependence
N7	.porosity	[double]	Porosity [>0;1.0]
if N4=2			CSM swelling model
N8	.porosity_-model_-values_0_4	[4* double]	Porosity model values
N9	.porosity	[double]	Porosity [>0;1.0]
if N4=4			Porosity influenced by deformation
N10	.porosity	[double]	Porosity [>0;1.0]
N11	.tortuosity	[double]	Medium tortuosity [>=0.0;1.0]
N12	.mobile_-immobile_-model	[int]	Mobil−Immobil−Modelle [0,..;0] 0: No mobile immobile site model 1: First order mobile immobile site model with constant transfer rate 2: First order mobile immobile site model with variable transfer rate 3: First order mobile immobile two site model with constant transfer rate 4: First order mobile immobile two site model with variable transfer rate
if N12=0			No mobile immobile site model
if N12=1			First order mobile immobile site model with constant transfer rate

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N13	.mobile_-immobile_-model_-values_0	[double]	Constant max. value
if N12=2			First order mobile immobile site model with variable transfer rate
N14	.mobile_-immobile_-model_-values_0_1	[2* double]	[0] variable max. value [1] curve
if N12=3			First order mobile immobile two site model with constant transfer rate
N15	.mobile_-immobile_-model_-value_0_1	[2* double]	Mobile immobile model values [0−1]
if N12=4			First order mobile immobile two site model with variable transfer rate
N16	.mobile_-immobile_-model_-values_0_1_2	[3* double]	Mobile immobile model values [0−2]
N17	.number_-grain_classes	[int]	Number of lithological components [1,..;1]
N17 * Loop			
N18	.grain_-radius_0	[double]	Grain class radius [0] [>0.0;1.0]
N19	.grain_-percent_0	[double]	Grain class percentage [0] [>=0.0;1.0]
N20	.maximum_-sorption_-model	[int]	Maximum sorption model [0,..;0] 0: No max. sorption model 1: Constant max. sorption concentration 2: Variable max. sorption concentration (curve)
if N20=0			No max. sorption model
if N20=1			Constant max. sorption concentration
N21	.maximum_-sorption_-model_-value_0	[double]	Constant max value
if N20=2			Variable max. sorption concentration (curve)
N22	.maximum_-sorption_-model_-value_0_1	[2* double]	[0] Variable max value [1] curve
N23	.nonlinearflowelement	[int]	Flow_model 0: Darcy 1: Forchheimer (nonlinear flow model)
if N23=0			Darcy
if N23=1			Forchheimer (nonlinear flow model)
N24	.nonlinear_-flowparameter	[double]	Alpha, non linear flow parameter [0.0−1.0]

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N25	.fluid_-storativity_-val_0	[double]	Specific storage coefficient [>=0.0;0.0] [1/Pa]
N26	.permeability_-model	[int]	Permeability model [0,1,4;0] 0: Constant permeability 1: Saline solution model 4: Permeability as function of porosity 5: Permeability as function of strains
if N26=0			Constant permeability
if N26=1			Saline solution model
if N26=4			Permeability as function of porosity
N27	.perm_poro_-val	[int]	Curve number, that defines k(n) [>0]
if N26=5			Permeability as function of strains
N28	.perm_-strain_-model_type	[int]	Type of the strain−permeability relation [>=0] 2: calculation of permeability by void fabric tensor M and porosity n fissure: k_x = b ((n+M_y)^3+(n+M_z)^3) tube : k_x = b (2n+M_y+M_z)^2) M quantifies the deviatoric part of the porosity. tube modell: c=1., d=2. fissure modell: c=3., d=1. 3: calculation of permeability by void distribution tensor N k_x = b*(N_y^c + N_z^c)^d k_y = b*(N_x^c + N_y^c)^d k_z = b*(N_x^c + N_y^c)^d N_x quantifies the pores, which are connected in the y−z−plane. tube modell: c=1., d=2. fissure modell: c=3., d=1. 1002: calculation of permeability by void fabric tensor M, porosity n and damage D fissure: k_x = b ((n+M_y)^3+(n+M_z)^3) + k_x_D tube : k_x = b (2n+M_y+M_z)^2) + k_y_D M quantifies the deviatoric part of the porosity. tube modell: c=1., d=2. fissure modell: c=3., d=1.
N29	.strain_-perm_-anisotropy	[double]	factor of anisotropy [0..1], 0=isotrop, 1=anisotrop
N30	.strain_-perm_k_min	[double]	k_min is used, if porosity equals zero [double;>0.0;1.e−25]
if N28=2			calculation of permeability by void fabric tensor M and porosity n fissure: k_x = b ((n+M_y)^3+(n+M_z)^3) tube : k_x = b (2n+M_y+M_z)^2) M quantifies the deviatoric part of the porosity. tube modell: c=1., d=2. fissure modell: c=3., d=1.

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N31	.perm_-strain_-model_-value_c	[double]	value c for strain perm relation
N32	.perm_-strain_-model_-value_d	[double]	value d for strain perm relation
if N28=3			calculation of permeability by void distribution tensor N k_x = b*(N_y^c + N_z^c)^d k_y = b*(N_x^c + N_y^c)^d k_z = b*(N_x^c + N_y^c)^d N_x quantifies the pores, which are connected in the y−z−plane. tube modell: c=1., d=2. fissure modell: c=3., d=1.
N33	.perm_-strain_-model_-value_c	[double]	value c for strain perm relation
N34	.perm_-strain_-model_-value_d	[double]	value d for strain perm relation
if N28=1002			calculation of permeability by void fabric tensor M, porosity n and damage D fissure: k_x = b ((n+M_y)^3+(n+M_z)^3) + k_x_D tube : k_x = b (2n+M_y+M_z)^2) + k_y_D M quantifies the deviatoric part of the porosity. tube modell: c=1., d=2. fissure modell: c=3., d=1.
N35	.perm_-strain_-model_-value_c	[double]	value c for strain perm relation
N36	.perm_-strain_-model_-value_d	[double]	value d for strain perm relation
N37	.perm_-strain_-model_frac_-dist	[double]	Distance between damage induced fractures (m)
N38	.permeability_-tensor_type	[int]	Permeability tensor type [0,1,2;0] 0: Isotropic permeability 1: Isotropic permeability 2: Anisotropic permeability with arbitrary coordinate system
if N38=0			Isotropic permeability
N39	.k	[double]	Medium permeability K [m^2]
if N38=1			Isotropic permeability
if N1=2			Permeabiliy tensor 2D k[0]=k_xx k[1]=k_xy k[2]=k_yx k[3]=k_yy

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N40	.k_0	[double]	permeability K[0] [m^2]
N41	.k_1	[double]	permeability K[1] [m^2]
N42	.k_2	[double]	permeability K[2] [m^2]
N43	.k_3	[double]	permeability K[3] [m^2]
if N1=3			Permeabiliy tensor 3D k[0]=k_xx k[1]=k_xy k[2]=k_xz k[3]=k_yx k[4]=k_yy k[5]=k_yz k[6]=k_zx k[7]=k_zy k[8]=k_zz
N44	.k_0	[double]	permeability K[0] [m^2]
N45	.k_1	[double]	permeability K[1] [m^2]
N46	.k_2	[double]	permeability K[2] [m^2]
N47	.k_3	[double]	permeability K[3] [m^2]
N48	.k_4	[double]	permeability K[4] [m^2]
N49	.k_5	[double]	permeability K[5] [m^2]
N50	.k_6	[double]	permeability K[6] [m^2]
N51	.k_7	[double]	permeability K[7] [m^2]
N52	.k_8	[double]	permeability K[8] [m^2]
if N38=2			Anisotropic permeability with arbitrary coordinate system
if N1=2			Permeabiliy tensor 2D k[0]=k_xx k[1]=k_xy k[2]=k_yx k[3]=k_yy
N53	.k_0	[double]	permeability K[0] [m^2]
N54	.k_1	[double]	permeability K[1] [m^2]
N55	.k_2	[double]	permeability K[2] [m^2]
N56	.k_3	[double]	permeability K[3] [m^2]
if N1=3			Permeabiliy tensor 3D k[0]=k_xx k[1]=k_xy k[2]=k_xz k[3]=k_yx k[4]=k_yy k[5]=k_yz k[6]=k_zx k[7]=k_zy k[8]=k_zz
N57	.k_0	[double]	permeability K[0] [m^2]
N58	.k_1	[double]	permeability K[1] [m^2]
N59	.k_2	[double]	permeability K[2] [m^2]
N60	.k_3	[double]	permeability K[3] [m^2]
N61	.k_4	[double]	permeability K[4] [m^2]
N62	.k_5	[double]	permeability K[5] [m^2]
N63	.k_6	[double]	permeability K[6] [m^2]
N64	.k_7	[double]	permeability K[7] [m^2]
N65	.k_8	[double]	permeability K[8] [m^2]
N66	.permeability_csys	[int]	Coordinate system number (0 -> global CS)
N67	.rel_perm_satu_model	[double]	Saturation permeability model ([INT] should be used instead)

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning

N68	.rel_perm_-satu_val_0_5	[6* double]	relative Permeability−saturation
			A1=0. : rel. Permeabilitaet=1. fuer beide Phasen
			A1=1. : rel. Permeabilitaeten ueber Kurven (Schluesselwort #CURVES)
			A2=Nummer der Kurve fuer 1. Phase, A3 fuer 2. Phase
			A1=2. : lin. Anstieg zwischen A1 und A2 von 0. auf 1. fuer Phase1
			lin. Anstieg zwischen A4 und A5 von 0. auf 1. fuer Phase2
			A1=3. : parab. Anstieg zwischen A1 und A2 von 0. auf 1. mit Exponent
			A3 fuer Phase1
			parab. Anstieg zwischen A4 und A5 von 0. auf 1. mit Exponent
			A6 fuer Phase2
			A1>=4.: Empirische Funktionen nach Van Genuchten, Haverkamp, Brooks/Corey
			(Parameter siehe Manual)

N69	.cap_press_-satu_model	[double]	Capillary pressure saturation model ([INT] should be used instead) !!!Use $CAP_PRESSURE_SATURATION

N70	.cap_press_-satu_val_0_5	[6* double]	Relative Kapillardruck−Saettigungs−Beziehung
			A1=0. : Kapiallardruck = 0.
			A1=1. : Kapiallardruck ueber Kurve (Schluesselwort #CURVES),
			A2=Nummer der Kurve
			A1=2. : lin. Anstieg zwischen A3 und A2 von 0. auf A4
			A1=3. : parab. Anstieg zwischen A3 und A2 von 0. auf A4 mit Exponent
			A5 fuer Phase1
			A1>=4.: Beziehungen nach Van Genuchten, Haverkamp, Corey:
			Parameter siehe Manual
			Fuer A1>1. gilt:
			A2 : Residualsaettigung der benetzenden (2.) Phase
			A3 : Vollsaettigung der benetzenden (2.) Phase

N71	.massdispL	[double]	Longitudinal mass dispersion length [>=0.0;0.0]
N72	.massdispT	[double]	Transverse mass dispersion length [>=0.0;0.0]
N73	.heatdispL	[double]	Longitudinal heat dispersion length [>=0.0;0.0]
N74	.heatdispT	[double]	Transverse heat dispersion length [>=0.0;0.0]
N75	.densityrock	[double]	Density of rock [>0.0; 2500] [kg/m^3]
N76	.heatcapacityrock	[double]	Heat capacity of rock [>0.0; 1000] [J/(kg K)]
N77	.heat_-conductivity_-model	[int]	Heat conductivity model of rock [0,1;#] 0: Isotrope heatconductivity 1: Orthotrope heatconductivity
if N77=0			Isotrope heatconductivity
N78	.lambda_0	[double]	Heatconductivity value (expr)[>0.0] [W/(m K)]
if N77=1			Orthotrope heatconductivity
if N1=2			2D − tensor, lambda_xx, lambda_yy

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N79	.heatconductivitytensor[0]	[double]	heatconductivitytensor[0] (expr) [>0.0] [W/(m K)]
N80	.heatconductivitytensor[1]	[double]	heatconductivitytensor[1] (expr) [>0.0] [W/(m K)]
if N1=3			3D − 3tensor, lambda_xx, lambda_yy, lambda_zz
N81	.heatconductivitytensor[0]	[double]	heatconductivitytensor[0] (expr) [>0.0] [W/(m K)]
N82	.heatconductivitytensor[1]	[double]	heatconductivitytensor[1] (expr) [>0.0] [W/(m K)]
N83	.heatconductivitytensor[2]	[double]	heatconductivitytensor[2] (expr) [>0.0] [W/(m K)]

$ELECTRIC_CONDUCTIVITY			
N84	.electric_-conductivity_-model	[int]	Electric conductivity model 0: Constant conductivity
if N84=0			Constant conductivity
N85	.electric_-conductivity_-value_0	[double]	Constant electric conductivity [>0.]

$REL_PERMEABILITY_PRESSURE			
N86	.rel_perm_-press_model	[int]	Relative permeability pressure model 0: No influence 1: Pressure dependent rel. permeability via exponentiell function 2: Head(!) gradient dependent 4: Pressure dependent rel. permeability via curve
if N86=0			No influence
if N86=1			Pressure dependent rel. permeability via exponentiell function
N87	.rel_perm_-press_val_0_3	[4* double]	[0] increase factor1 [1] increase factor2 [2] sigma(z=0) [3] d_sigma/d_z
if N86=2			Head(!) gradient dependent
N88	.rel_perm_-press_val_0_3	[4* double]	k_rel = (grad(h)*alpha1)^(alpha2−1) k_rel = Max(Min(k_rel, alpha4),alpha3 [0] alpha1 [1] alpha2 [2] alpha3 [3] alpha4
if N86=4			Pressure dependent rel. permeability via curve
N89	.rel_perm_-press_val_0_-1_2	[3* double]	[0] curve number (as real number) [1] sigma(z=0) [2] d_sigma/d_z

$REL_PERMEABILITY_VELOCITY			
N90	.rel_perm_-veloc_model	[int]	Relative permeability velocity model 0: No influence
if N90=0			No influence
if N90=1			

continued

Continuation from the previous page

Parameter		RF-Variable	Values	Meaning
	N91	.rel_perm_-velo_val_0_1	[2* double]	rel. perm. velocity values [0−1]

$REL_PERMEABILITY_SATURATION

N92		.rel_perm_-satu_model	[int]	Relative permeability saturation model 0: No influence 1: Curves 2: Linear 3: Exponent 4: V.Genuchten 5: Haverkamp 6: Brooks/Corey 8: Brooks/Corey 11: 3Phase, curves 12: 3Phase, exponent 14: V.Genuchten + CSM
if N92=0				No influence
if N92=1				Curves
	N93	.rel_perm_-satu_val_0_1	[2* double]	Curves
if N92=2				Linear
	N94	.rel_perm_-satu_val_0_5	[6* double]	Linear rel. perm. saturation values
if N92=3				Exponent
	N95	.rel_perm_-satu_val_0_5	[6* double]	Exponent, rel. perm. saturation values
if N92=4				V.Genuchten
	N96	.rel_perm_-satu_val_0_4	[5* double]	V.Genuchten, rel. perm. saturation values
if N92=5				Haverkamp
	N97	.rel_perm_-satu_val_0_4	[5* double]	Haverkamp, rel. perm. saturation values
if N92=6				Brooks/Corey
	N98	.rel_perm_-satu_val_0_1	[2* double]	Brooks/Corey (simple), rel. perm. saturation values
if N92=7				
	N99	.rel_perm_-satu_val_0_4	[5* double]	
if N92=8				Brooks/Corey
	N100	.rel_perm_-satu_val_0_2	[3* double]	Brooks/Corey, rel. perm. saturation values
if N92=11				3Phase, curves
	N101	.rel_perm_-satu_val_0_-1_2	[3* double]	3Phase, curves
if N92=12				3Phase, exponent
	N102	.rel_perm_-satu_val_0_8	[9* double]	3Phase, exponent, perm. saturation values
if N92=14				V.Genuchten + CSM
	N103	.rel_perm_-satu_val_0_7	[8* double]	V.Genuchten+ CSM, rel. perm. saturation values

$REL_PERMEABILITY_DEFORMATION

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N104	.rel_perm_-defo_model	[int]	Relative permeability deformation model 0: No influence 1: Swelling pressure
if N104=0			No influence
if N104=1			Swelling pressure
N105	.rel_perm_-defo_val_0_5	[6* double]	Swelling pressure, rel. perm. deformation values
if N104=2			
N106	.rel_perm_-defo_val_0_1	[2* double]	rel. perm. deformation values

$CAP_PRESSURE_SATURATION

N107	.cap_press_-satu_model	[int]	Capillary pressure saturation model 0: No influence 1: Curve 2: Linear 3: Exponent 4: V.Genuchten 5: Haverkamp 6: Brooks/Corey 7: V.Genuchten limited 11: 3Phase, curves
if N107=0			No influence
if N107=1			Curve
N108	.cap_press_-satu_val_0	[double]	Curve
if N107=2			Linear
N109	.cap_press_-satu_val_0_-1_2	[3* double]	Linear, cap. pressure. saturation values
if N107=3			Exponent
N110	.cap_press_-satu_val_0_3	[4* double]	Exponent, cap. pressure. saturation values
if N107=4			V.Genuchten
N111	.cap_press_-satu_val_0_4	[5* double]	V.Genuchten, cap. pressure. saturation values
if N107=5			Haverkamp
N112	.cap_press_-satu_val_0_4	[5* double]	Haverkamp, cap. pressure. saturation values
if N107=6			Brooks/Corey
N113	.cap_press_-satu_val_0_-1_2	[3* double]	Brooks/Corey, cap. pressure. saturation values
if N107=7			V.Genuchten limited
N114	.cps_values_-5x_0_4	[5* double]	V.Genuchten limited, cap. pressure. saturation values
if N107=11			3Phase, curves
N115	.cps_values_-2x_0_1	[2* double]	3Phase, curves, cap. pressure. saturation values

$FLUID_STORATIVITY

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N116	.fluid_-storativity_-model	[int]	Fluid storativity model 0: Constant storativity 1: Constant storativity 2: Pressure dependent storativity 3: Pressure dependent storativity via curve 4: Storativity is a function of effective stress defined by #CURVES
if N116=0			Constant storativity
N117	.fluid_-storativity_-val_0	[double]	Constant storativity value
if N116=1			Constant storativity
N118	.fluid_-storativity_-val_0	[double]	Constant storativity value
if N116=2			Pressure dependent storativity
N119	.fluid_-storativity_-val_0_3	[4* double]	Pressure dependent storativity values [0] S0 [1] increase [2] sigma(z0) [3] d_sigma/d_z
if N116=3			Pressure dependent storativity via curve
N120	.fluid_-storativity_-val_0_1_2	[3* double]	Pressure dependent storativity via curve [0] curve number (as real number) [1] sigma(z0) [2] d_sigma/d_z
if N116=4			Storativity is a function of effective stress defined by #CURVES
N121	.fluid_-storativity_-val_0_1_2	[3* double]	Storativity is a function of effective stress defined by #CURVES [3 double values] [0] curve number (as real number) [1] time collocation [2] solid density -> SOLID_PROPERTIES

$FLUID_EXCHANGE_WITH_OTHER_CONTINUA			
N122	.flu_sto_-model	[int]	Fluid exchange model 0: No exchange 1: Linear exchange
if N122=0			No exchange
if N122=1			Linear exchange
N123	.fluid_-exchange_-val_0_1	[2* double]	Linear exchange values

$UNCONFINED_FLOW_GROUP			
N124	.unconfined_-flow_group	[int]	Unconfined flow group

17.2 Example

Input example 1: (applications/d_iv_phase01.rfd)

```
#SOIL_PROPERTIES
 2                  ; element dimension
 1.0                ; element exchange area
 ;POROSITAET--------------------------------------
 4                  ; porosity-model (swelling and deformation)
 0.05               ; initial porosity
 ;POROSITAET-end-----------------------------------

 1.                 ; tortuosity
 0                  ; mobile immobile model
 0                  ; lithological component
 0                  ; maximum sorption model
 0                  ; nonlinear flow parameter
 0.0                ; storativity

 ;---- PERMEABILITAET--------------------------------
 ; porosity-dependent, isotropic
 4                  ; permeability model (4=permeability as function of porosity)
 3                  ; poro-perm relation given by CURVE 3

 0                  ; permeability tensor type (0=isotrop)
 1.0                ; initial permeability (CURVE values will be multiplied by this value)
 ;---- PERMEABILITAET-end----------------------------

 1.0  0.1  2.0  1.0  0.0  0.9  2.0  ; k_r - S relation: given by CURVE 2
 1.0  1.0  1.0  1.0  1.0  1.0  1.0  ; pc  - S relation: given by CURVE 1

 0.1   0.02         ; long. and trans. mass dispersion lengths
 0.    0.           ; long. and trans. heat dispersion lengths
 2550. 0.           ; density, heat capacity
 0     0.           ; heat conductivity model, conductivities
```

Input example 2: (benchmarks/perm_strain/strain_perm_test2d_01.rfd)

```
#SOIL_PROPERTIES
 2                    ; element dimension
 1.0                  ; element exchange area
;POROSITAET-----------------------------------------
 4                    ; porosity-model (swelling and deformation)
  0.3                 ; initial porosity
;POROSITAET-end-------------------------------------
 1.                   ; tortuosity
 0                    ; mobile immobile model
 0                    ; lithological component
 0                    ; maximum sorption model
 0                    ; nonlinear flow parameter
 0.0                  ; storativity

 ;---- PERMEABILITAET--------------------------------
 ; permeability model

 ;  4                  ; perm model (0=const, 4=porositaetsabhaengig)
 ;    3                ; Curve number for poro-perm-relation (if perm model == 4)

   5                  ;  perm model (0=const, 5=strain-dependency)
    2                 ; type of strain-permeability relation
    1.0               ; m, factor of anisotropy, fitting parameter (0=isotrop,1=anisotrop)
    1.e-30            ; k_min, minimal permeability

       ; strain-perm model 2: permeability as function of void fabric tensor
                        ; k = f(M,n,b,c,d), n = porosity, M= void fabric tensor
                        ; if c=1,d=2 => tube model
  ; k_x = b(2n + M_y + M_z)^2
  ; k_y = b(2n + M_x + M_z)^2
  ; k_z = b(2n + M_x + M_y)^2

 ; if c=3,d=1 => fissure model
  ; k_x = b[(2n + M_y)^3 + (n + M_z)^3]
  ; k_y = b[(2n + M_x)^3 + (n + M_z)^3]
  ; k_z = b[(2n + M_x)^3 + (n + M_y)^3]

                        ; c,d - benutzerdefiniert
                        ;   if c=1,d=2 => tube model
                        ;   if c=3,d=1 => fissure model
                        ; k_min - benutzerdefiniert (wird verwendet wenn k<k_min)
   ;  1.0 2.0           ; c, d (tube model)
     3.0 1.0            ; c, d (fissure model)

 ; initial permeability tensor type and data
   2                    ; initial permeability tensor type
                         ; (2=anisotropy, needs 4 values in 2D and 9 values in 3D)
      1.0e-11  0.0      ; initial permeability tensor (kxx kxy)
      0.0      1.0e-11  ; initial permeability tensor (kyx kyy)

       0                ; number of coordinate system (0=global)
 ;---- PERMEABILITAET-end---------------------------

  1.000000  0.10000  2.0000  1.000000  0.0000  .90000  2.000000  ; k_r - S relation: given by CURVE 2
  1.0 1.0 1.0 1.0 1.0 1.0 1.0                                    ; pc  - S relation: given by CURVE 1

 0.1   0.02           ; long. and trans. mass dispersion lengths
 0.    0.             ; long. and trans. heat dispersion lengths
 2.75e3 0.            ; density, heat capacity
 0     0.             ; heat conductivity model, conductivities
```

Input example 3: strain dependent anisotropic permeability, considering damage

```
#SOIL_PROPERTIES
 2                    ; element dimension
 1.0                  ; element exchange area
;POROSITAET-----------------------------------------
 4                    ; porosity model (4=considering strains)
 0.09                 ; initial porosity
;POROSITAET-end-------------------------------------
 1.                   ; tortuosity
 0                    ; mobile immobile model
 0                    ; lithological component
 0                    ; maximum sorption model
 0                    ; nonlinear flow parameter
 0.0                  ; storativity

;PERMEABILITY---------------------------------------
   5                  ;  perm model (0=const, 5=strain-dependency)
   1002               ; type of strain-permeability relation (1002=considering damage)
    1.0                ; m, factor of anisotropy (0=isotrop,1=anisotrop)
    1.e-30             ; k_min, minimal permeability
     ; strain-perm model 1002: permeability as function of void fabric tensor and damage
           ; k = k_intact + k_damage, n = porosity, M= void fabric tensor, D = damage state
         ; separation of the strain:
   ; strain = strain_frac + strain_intact
   ;          strain_frac = D*strain
   ;        strain_intact = (1-D)*strain

         ; k_frac_x = (strain_frac_y^3 * s^2)/12 + (strain_frac_z^3 * s^2)/12
         ; k_frac_y = (strain_frac_x^3 * s^2)/12 + (strain_frac_z^3 * s^2)/12
         ; k_frac_z = (strain_frac_x^3 * s^2)/12 + (strain_frac_y^3 * s^2)/12

 ; M = f(strain_intact, n_initial); n = f(strain_intact, n_init)
 ; if c=1,d=2 => tube model
  ; k_intact_x = b(2n + M_y + M_z)^2
  ; k_intact_y = b(2n + M_x + M_z)^2
  ; k_intact_z = b(2n + M_x + M_y)^2

 ; if c=3,d=1 => fissure model
  ; k_intact_x = b[(2n + M_y)^3 + (n + M_z)^3]
  ; k_intact_y = b[(2n + M_x)^3 + (n + M_z)^3]
  ; k_intact_z = b[(2n + M_x)^3 + (n + M_y)^3]

    3.0 1.0             ; c, d  (fissure model)
    1.e-6               ; s, frac-dist

 ; initial permeability tensor type and data
   2                   ; initial permeability tensor type
                        ; (2=anisotropy, needs 4 values in 2D and 9 values in 3D)
    0.2e-20 0.0        ; initial permeability tensor (kxx kxy)
     0.0   0.2e-20     ; initial permeability tensor (kyx kyy)

       1                ; number of coordinate system (0=global)
 ;---- PERMEABILITAET-end-------------------------------

  1.000000  0.10000  2.0000  1.000000  0.0000  .90000  2.000000  ; k_r - S relation: given by CURVE 2
  1.0 1.0 1.0 1.0 1.0 1.0 1.0                                    ; pc  - S relation: given by CURVE 1

 0.1   0.02           ; long. and trans. mass dispersion lengths
 0.    0.             ; long. and trans. heat dispersion lengths
 0.0   0.             ; density, heat capacity
 0     0.             ; heat conductivity model, conductivities
 0     0.             ; heat conductivity model, conductivities
```

18 Solid Properties

18.1 Keyword Description

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 material groups are defined.

#SOLID_PROPERTIES			
Parameter	RF-Variable	Values	Meaning
N1	.material_-model	[int]	Material model 1: Linear elasticity 2: Nonlinear elasticity, compressible (Required for usage of $COMPRESSIBILITY_MODEL) 10: Anisotropic linear elasticity 101: J2 von Mises Plasticity 102: (unofficial) Drucker−Prager Plasticity (coefficients) 103: (unofficial) Drucker−Prager Plasticity (Mohr−Coulomb values) 110: CamClayModel Version 1 (hardening,linear elastic,associated 1001: Linear isotropic elasticity with damage 1010: Isotrop Damage Model with linear anisotropic elasticity based on Ju:1989) energy based, exp. growing progressive damage
if N1=1			Linear elasticity
N2	.e_modulus	[double]	Young's modulus [E>0;20000.0] [Pa]
N3	.poissons_-ratio	[double]	Poisson's ratio [nue=0.0,0.5;0.3] [−]
if N1=2			Nonlinear elasticity, compressible (Required for usage of $COMPRESSIBILITY_MODEL)
N4	.e_modulus	[double]	Young's modulus [E>0;20000.0] [Pa]
N5	.poissons_-ratio	[double]	Poisson's ratio [nue=0.0,0.5;0.3] [−]
if N1=10			Anisotropic linear elasticity
N6	.elasticity_-representation_-type	[int]	Representation of coefficients [1,2,3,4] 1: Plane of isotropy: 23 (a=e1) 2: Plane of isotropy: 13 (a=e2) 3: Plane of isotropy: 12 (a=e3) 4: Plane of isotropy: 23 (a=e1)
if N6=1			Plane of isotropy: 23 (a=e1)
N7	.elasticity_-parameter_0	[double]	E1 Young's modulus in direction (1) [>0] [Pa]
N8	.elasticity_-parameter_1	[double]	E2 = E3 Young's modulus in plane of isotropy [>0] [Pa]
N9	.elasticity_-parameter_2	[double]	nue12 Poisson´s ratio [0.0,0.5;0.3] [−]

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N10	.elasticity_-parameter_3	[double]	nue23 Poisson´s ratio [0.0,0.5;0.3] [−]
N11	.elasticity_-parameter_4	[double]	G12 Cross shear modulus [>0] [Pa]
if N6=2			Plane of isotropy: 13 (a=e2)
N12	.elasticity_-parameter_0	[double]	E2 Young's modulus in direction (2) [>0] [Pa]
N13	.elasticity_-parameter_1	[double]	E1 = E3 Young's modulus in plane of isotropy [>0] [Pa]
N14	.elasticity_-parameter_2	[double]	nue21 Poisson´s ratio [0.0,0.5;0.3] [−]
N15	.elasticity_-parameter_3	[double]	nue13 Poisson´s ratio [0.0,0.5;0.3] [−]
N16	.elasticity_-parameter_4	[double]	G21 Cross shear modulus [>0] [Pa]
if N6=3			Plane of isotropy: 12 (a=e3)
N17	.elasticity_-parameter_0	[double]	E3 Young's modulus in direction (3) [>0] [Pa]
N18	.elasticity_-parameter_1	[double]	E1 = E2 Young's modulus in plane of isotropy [>0] [Pa]
N19	.elasticity_-parameter_2	[double]	nue31 Poisson´s ratio [0.0,0.5;0.3] [−]
N20	.elasticity_-parameter_3	[double]	nue12 Poisson´s ratio [0.0,0.5;0.3] [−]
N21	.elasticity_-parameter_4	[double]	G31 Cross shear modulus [>0] [Pa]
if N6=4			Plane of isotropy: 23 (a=e1)
N22	.elasticity_-parameter_0	[double]	lambda [Pa]
N23	.elasticity_-parameter_1	[double]	alpha [Pa]
N24	.elasticity_-parameter_2	[double]	beta [Pa]
N25	.elasticity_-parameter_3	[double]	mue_l [Pa]
N26	.elasticity_-parameter_4	[double]	mue_t [Pa]
N27	.elasticity_-csys	[int]	number of dedicated coordinate system [elasticity_-csys>0, 0]
if N1=101			J2 von Mises Plasticity
N28	.e_modulus	[double]	Young's modulus [E>0;20000.0] [Pa]
N29	.poissons_-ratio	[double]	Poisson's ratio [0.0,0.5;0.3] [−]
N30	.yield_stress	[double]	Yield stress [>0;0.0] [Pa]
if N1=102			(unofficial) Drucker−Prager Plasticity (coefficients)
N31	.e_modulus	[double]	Young's modulus [E>0;20000.0] [Pa]
N32	.poissons_-ratio	[double]	Poisson's ratio [0.0,0.5;0.3] [−]
N33	.yield_func_-parameter_0	[double]	kappa [Pa]
N34	.yield_func_-parameter_1	[double]	alpha (internal friction) coefficient [−]
N35	.yield_func_-parameter_2	[double]	alpha_(dilatancy) coefficient [−]

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
if N1=103			(unofficial) Drucker−Prager Plasticity (Mohr−Coulomb values)
N36	.e_modulus	[double]	Young's modulus [E>0;20000.0] [Pa]
N37	.poissons_-ratio	[double]	Poisson's ratio [nue=0.0,0.5;0.3] [−]
N38	.yield_func_-parameter_0	[double]	cohesion [Pa]
N39	.yield_func_-parameter_1	[double]	internal friction angle [°]
N40	.yield_func_-parameter_2	[double]	dilatancy angle [°]
if N1=110			CamClayModel Version 1 (hardening,linear elastic,associated
N41	.e_modulus	[double]	Young's modulus [E>0;20000.0] [Pa]
N42	.poissons_-ratio	[double]	Poisson's ratio [nue=0.0,0.5;0.3] [−]
N43	.yield_func_-parameter_0	[double]	CamClayParameter related to friction angle
N44	.yield_func_-parameter_1	[double]	CamClayParameter: consolidation pressure (Pa)
N45	.yield_func_-parameter_2	[double]	CamClayParameter: virgin compression index (−)
N46	.yield_func_-parameter_3	[double]	CamClayParameter: recompression index (−)
if N1=1001			Linear isotropic elasticity with damage
N47	.e_modulus	[double]	Young's modulus [E>0;20000.0] [Pa]
N48	.poissons_-ratio	[double]	Poisson's ratio [nue=0.0,0.5;0.3] [−]
N49	.damage_-model	[int]	Damage model model 1: scalar damage model model 2: tensorial damage model
N50	.damage_-criterion	[int]	Type of damage criterion type 1: 0.5 eps_el : C_0 : eps_el − r <= 0.0 type 2: sqrt(eps_el : C_0 : eps_el) − r <= 0.0
N51	.damage_-brittle	[double]	Tensile sensitivity 0.0 => tension and compression causes damage (ductile), 1.0 => only tension causes damage (brittle) in case of 1.0, only tensile strains cause damage
N52	.damage_-parameter_0	[double]	r_0, initial damage threshold
N53	.damage_-evolution	[int]	Type of damage evolution type 1: Marigo, 1981 type 2: Mazars, 1986
N54	.damage_-parameter 1	[double]	damage evolution function, parameter A
N55	.damage_-parameter 2	[double]	damage evolution function, parameter B
N56	.damage_-parameter 3	[double]	maximal damage state

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N57	.damage_-parameter 4	[double]	Microcrack opening / closing 1.0 => considering opening/closing mechanism
N58	.damage_-parameter 5	[double]	mue, damage viscosity coefficient −1 0 => inactive infinity => inviscid damage 0.0 => instantaneous elastic response
if N1=1010			Isotrop Damage Model with linear anisotropic elasticity based on Ju:1989) energy based, exp. growing progressive damage
N59	.elasticity_-representation_-type	[int]	Representation of coefficients [1,2,3,4] 1: Plane of isotropy: 23 (a=e1) 2: Plane of isotropy: 13 (a=e2) 3: Plane of isotropy: 12 (a=e3) 4: Plane of isotropy: 23 (a=e1)
if N59=1			Plane of isotropy: 23 (a=e1)
N60	.elasticity_-parameter_0	[double]	E1 Young's modulus in direction (1) [>0] [Pa]
N61	.elasticity_-parameter_1	[double]	E2 = E3 Young's modulus in plane of isotropy [>0] [Pa]
N62	.elasticity_-parameter_2	[double]	nue12 Poisson´s ratio [0.0,0.5;0.3] [−]
N63	.elasticity_-parameter_3	[double]	nue23 Poisson´s ratio [0.0,0.5;0.3] [−]
N64	.elasticity_-parameter_4	[double]	G12 Cross shear modulus [>0] [Pa]
if N59=2			Plane of isotropy: 13 (a=e2)
N65	.elasticity_-parameter_0	[double]	E2 Young's modulus in direction (2) [>0] [Pa]
N66	.elasticity_-parameter_1	[double]	E1 = E3 Young's modulus in plane of isotropy [>0] [Pa]
N67	.elasticity_-parameter_2	[double]	nue21 Poisson´s ratio [0.0,0.5;0.3] [−]
N68	.elasticity_-parameter_3	[double]	nue13 Poisson´s ratio [0.0,0.5;0.3] [−]
N69	.elasticity_-parameter_4	[double]	G21 Cross shear modulus [>0] [Pa]
if N59=3			Plane of isotropy: 12 (a=e3)
N70	.elasticity_-parameter_0	[double]	E3 Young's modulus in direction (3) [>0] [Pa]
N71	.elasticity_-parameter_1	[double]	E1 = E2 Young's modulus in plane of isotropy [>0] [Pa]
N72	.elasticity_-parameter_2	[double]	nue31 Poisson´s ratio [0.0,0.5;0.3] [−]
N73	.elasticity_-parameter_3	[double]	nue12 Poisson´s ratio [0.0,0.5;0.3] [−]
N74	.elasticity_-parameter_4	[double]	G31 Cross shear modulus [>0] [Pa]
if N59=4			Plane of isotropy: 23 (a=e1)
N75	.elasticity_-parameter_0	[double]	lambda [Pa]
N76	.elasticity_-parameter_1	[double]	alpha [Pa]

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N77	.elasticity_-parameter_2	[double]	beta [Pa]
N78	.elasticity_-parameter_3	[double]	mue_l [Pa]
N79	.elasticity_-parameter_4	[double]	mue_t [Pa]
N80	.elasticity_-csys	[int]	number of dedicated coordinate system [elasticity_-csys>0, 0]
N81	.damage_-model	[int]	Damage model model 1: scalar damage model model 2: tensorial damage model
N82	.damage_-criterion	[int]	Type of damage criterion type 1: 0.5 eps_el : C_0 : eps_el − r <= 0.0 type 2: sqrt(eps_el : C_0 : eps_el) − r <= 0.0
N83	.damage_-brittle	[double]	Tensile sensitivity 0.0 => tension and compression causes damage (ductile), 1.0 => only tension causes damage (brittle) in case of 1.0, only tensile strains cause damage
N84	.damage_-parameter_0	[double]	r_0, initial damage threshold
N85	.damage_-evolution	[int]	Type of damage evolution type 1: Marigo, 1981 type 2: Mazars, 1986
N86	.damage_-parameter 1	[double]	damage evolution function, parameter A
N87	.damage_-parameter 2	[double]	damage evolution function, parameter B
N88	.damage_-parameter 3	[double]	maximal damage state
N89	.damage_-parameter 4	[double]	Microcrack opening / closing 1.0 => considering opening/closing mechanism
N90	.damage_-parameter 5	[double]	mue, damage viscosity coefficient −1 0 => inactive infinity => inviscid damage 0.0 => instantaneous elastic response

$THERMAL_EXPANSION_COEFFICIENT			
N91	.thermal_-expansion_-coefficient	[double]	Thermal expansion coeff. [>0.0;0.0] (Applied in material models: [1])

$MOISTURE_SWELLING

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N92	.moisture_-swelling_-model	[int]	swelling / shrinkage model (Applied in material models: [1], [10]) 0: Swelling Model inactive 1: linear isotropic swelling model (eps_vol_sw = eps_vol_sw_0 + beta * (Satu − S_0) 2: non−linear isotropic swelling model (eps_vol_sw = f(S)) 11: linear anisotropic swelling model (eps_sw_xx = eps_sw_0_xx + beta_xx * (Satu − S_0) 12: non−linear anisotropic swelling model (eps_vol_xx = f_x(S); (x=1,2,3)) 13: exponential anisotropic swelling model For each direction (1,2,3) one exponential curve defines the swelling strain. (eps_sw_xx = eps_sw_0 + a[x] * (S^b[x] − S_0^b[x])); (x=1,2,3)
if N92=0			Swelling Model inactive
if N92=1			linear isotropic swelling model (eps_vol_sw = eps_vol_sw_0 + beta * (Satu − S_0)
N93	.moisture_-swelling_-parameter_0	[double]	beta_sw, volum. swelling coefficient [−]
N94	.moisture_-swelling_-parameter_1	[double]	Smin, region of swelling: min. saturation [−]
N95	.moisture_-swelling_-parameter_2	[double]	Smax, region of swelling: max. saturation [−]
N96	.moisture_-swelling_-parameter_3	[double]	S_0, reference water saturation [−]
N97	.moisture_-swelling_-parameter_4	[double]	eps_vol_sw_0, volum. reference strain [−]
if N92=2			non−linear isotropic swelling model (eps_vol_sw = f(S))
N98	.moisture_-swelling_-parameter_0	[double]	CURVE number, (Saturation <−> eps_vol_sw)
if N92=11			linear anisotropic swelling model (eps_sw_xx = eps_sw_0_xx + beta_xx * (Satu − S_0)
N99	.moisture_-swelling_-parameter_0	[double]	beta_sw_11, swelling coefficient in direction 1 [−]
N100	.moisture_-swelling_-parameter_1	[double]	beta_sw_22, swelling coefficient in direction 2 [−]
N101	.moisture_-swelling_-parameter_2	[double]	beta_sw_33, swelling coefficient in direction 3 [−]
N102	.moisture_-swelling_-parameter_3	[double]	Number of coordinate system (0=global)

continued

Continuation from the previous page

Parameter		RF-Variable	Values	Meaning
	N103	.moisture_-swelling_-parameter_4	[double]	Smin, region of swelling: min. saturation [−]
	N104	.moisture_-swelling_-parameter_5	[double]	Smax, region of swelling: max. saturation [−]
	N105	.moisture_-swelling_-parameter_6	[double]	S_0, reference water saturation [−]
	N106	.moisture_-swelling_-parameter_7	[double]	eps_sw_0_11, reference strain in direction 1 [−]
	N107	.moisture_-swelling_-parameter_8	[double]	eps_sw_0_22, reference strain in direction 2 [−]
	N108	.moisture_-swelling_-parameter_9	[double]	eps_sw_0_33, reference strain in direction 3 [−]
if N92=12				non−linear anisotropic swelling model (eps_vol_xx = f_x(S); (x=1,2,3))
	N109	.moisture_-swelling_-parameter_0	[double]	CURVE number, (Saturation <−> eps_sw_11)
	N110	.moisture_-swelling_-parameter_1	[double]	CURVE number, (Saturation <−> eps_sw_22)
	N111	.moisture_-swelling_-parameter_2	[double]	CURVE number, (Saturation <−> eps_sw_33)
	N112	.moisture_-swelling_-parameter_3	[double]	Number of coordinate system (0=global)
if N92=13				exponential anisotropic swelling model For each direction (1,2,3) one exponential curve defines the swelling strain. (eps_sw_xx = eps_sw_0 + a[x] * (S^b[x] − S_0^b[x])); (x=1,2,3)
	N113	.moisture_-swelling_-parameter_0	[double]	a[1]; material parameter a for curve 1
	N114	.moisture_-swelling_-parameter_1	[double]	a[2]; material parameter a for curve 2
	N115	.moisture_-swelling_-parameter_2	[double]	a[3]; material parameter a for curve 3
	N116	.moisture_-swelling_-parameter_3	[double]	b[1]; material parameter b for curve 1
	N117	.moisture_-swelling_-parameter_4	[double]	b[2]; material parameter b for curve 2

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N118	.moisture_-swelling_-parameter_5	[double]	b[3]; material parameter b for curve 3
N119	.moisture_-swelling_-parameter_6	[double]	S_0, reference water saturation [−]
N120	.moisture_-swelling_-parameter_7	[double]	eps_sw_0, reference strain [−]
N121	.moisture_-swelling_-parameter_8	[double]	Number of coordinate system (0=global)

$YIELD_FUNCTION_EVALUATION			
N122	.yield_-function_-evaluation_-type	[int]	yield function evaluation type 0: No evaluation 1: Evaluation equal to material model 2: Von Mises 3: Drucker−Prager 4: Tresa 5: Mohr−Coulomb 103: Drucker−Prager: (c, phi) werden in (kappa, alpha) umgerechnet 6: Ehlers
if N122=0			No evaluation
if N122=1			Evaluation equal to material model
if N122=2			Von Mises
N123	.yield_-func_eval_-parameter_0	[double]	sigma yield [Pa]
if N122=3			Drucker−Prager
N124	.yield_-func_eval_-parameter_0	[double]	kappa [Pa]
N125	.yield_-func_eval_-parameter_1	[double]	alpha [−]
if N122=4			Tresa
N126	.yield_-func_eval_-parameter_0	[double]	kappa [Pa]
if N122=5			Mohr−Coulomb
N127	.yield_-func_eval_-parameter_0	[double]	cohesion c [Pa]
N128	.yield_-func_eval_-parameter_1	[double]	phi, angle of internal friction [°]
if N122=103			Drucker−Prager: (c, phi) werden in (kappa, alpha) umgerechnet
N129	.yield_-func_eval_-parameter_0	[double]	cohesion c [Pa]

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N130	.yield_-func_eval_-parameter_1	[double]	phi, angle of internal friction [°]
if N122=6			Ehlers
N131	.yield_-func_eval_-parameter_0	[double]	alpha [−]
N132	.yield_-func_eval_-parameter_1	[double]	beta [−]
N133	.yield_-func_eval_-parameter_2	[double]	gamma [−]
N134	.yield_-func_eval_-parameter_3	[double]	delta [m2/N]
N135	.yield_-func_eval_-parameter_4	[double]	epsilon [m2/N]
N136	.yield_-func_eval_-parameter_5	[double]	kappa [N/m2]
N137	.yield_-func_eval_-parameter_6	[double]	m [−]

$DRUCKER−PRAGER			
N138	.PlasticParameter-0	[double]	Initial yield stress
N139	.PlasticParameter-1	[double]	Platic Hardening modulus
N140	.PlasticParameter-2	[double]	Internal frection angle
N141	.PlasticParameter-3	[double]	Dilatancy angle

$EHLERS_WEIMAR			
N142	.PlasticParameter-0	[double]	alpha0
N143	.PlasticParameter-1	[double]	beta0
N144	.PlasticParameter-2	[double]	delta0
N145	.PlasticParameter-3	[double]	epsilon0
N146	.PlasticParameter-4	[double]	kappa0
N147	.PlasticParameter-5	[double]	gamma0
N148	.PlasticParameter-6	[double]	m0

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N149	.PlasticParameter[double] 7		alpha1
N150	.PlasticParameter[double] 8		beta1
N151	.PlasticParameter[double] 9		delta1
N152	.PlasticParameter[double] 10		epsilon1
N153	.PlasticParameter[double] 11		kappa1
N154	.PlasticParameter[double] 12		gamma1
N155	.PlasticParameter[double] 13		m1
N156	.PlasticParameter[double] 14		Psi1
N157	.PlasticParameter[double] 15		Psi2
N158	.PlasticParameter[double] 16		Ch
N159	.PlasticParameter[double] 17		Cd
N160	.PlasticParameter[double] 18		br
N161	.PlasticParameter[double] 19		mr

$BIOT_COUPLING_COEFFICIENT			
N162	.biot_coupling_model	[int]	biot_coupling_model 0: Input of Biot Parameter 1: Input of bulk modulus of grains
if N162=0			Input of Biot Parameter
N163	.biot_coupling_parameter_0	[double]	alpha, Biot coefficient [>0.0,1.0;1.0]
if N162=1			Input of bulk modulus of grains
N164	.biot_coupling_parameter_0	[double]	Ks, bulk modulus of grains [>0;1.0e−8]

$EFFECTIVE_STRESS_COEFFICIENT			
N165	.effective_stress_coeff_model	[int]	Sigma_tot = Sigma_eff − 1 *BIOT * effect_stress_coeff * pressure 0: effect stress coefficient = 0.0 (no inluence from hydraulic to mechanic) 1: default: effect stress coefficient = Saturation, S 2: effect stress coefficient = [(S−S_res)/(1−S_res)]^kappa ,if S_res<S<1 effect stress coefficient = 1 ,if S=1 effect stress coefficient = 0 ,if S<=S_res, S!=1

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
if N165=0			effect stress coefficient = 0.0 (no inluence from hydraulic to mechanic)
if N165=1			default: effect stress coefficient = Saturation, S
if N165=2			effect stress coefficient = [(S−S_res)/(1−S_res)]^kappa ,if S_res<S<1 effect stress coefficient = 1 ,if S=1 effect stress coefficient = 0 ,if S<=S_res, S!=1
N166	.effect_stress_coeff_satu_res	[double]	residual saturation [>=0.0,1.0;0.0]
N167	.effect_stress_coeff_expo	[double]	Exponent kappa [>=0.0]

$COMPRESSIBILITY_MODEL			
N168	.compressibility_model	[int]	Extended Compressibility Models (Applied in material models: [2]) 0: (no additional parameters) 1: (1 parameter)
if N168=0			(no additional parameters)
if N168=1			(1 parameter)
N169	.stress_free_porosity	[double]	Stress−free porosity. [>0.0;0.5] (Applied in material models: [2])

$YOUNGS_MODULUS_SATURATION_DEPENDENCY			
N170	.youngs_modulus_saturation_dependency_model	[int]	Youngs Modulus depends on Saturation (Applied in material models: [1],[2]) 0: (no additional parameters) 1: (1 parameter)
if N170=0			(no additional parameters)
if N170=1			(1 parameter)
N171	.Dependency_given_by_Curve	[double]	Dependency given by Curve

18.2 Effective stress coefficient

Extension of Terzaghi´s effective stress law:

$$\boldsymbol{\sigma}_{\mathrm{tot}} = \boldsymbol{\sigma}_{\mathrm{eff}} - \mathbf{1}\alpha p\chi \tag{1}$$

With the effective stress $\boldsymbol{\sigma}_{\mathrm{eff}}$, the total stress $\boldsymbol{\sigma}_{\mathrm{tot}}$, the Biot coefficient α, the water saturation S^{w} and the pore water pressure p. The effective stress coefficient χ is a function of saturation of the soil and reflects the contribution of matric suction to effective stress.

$$\chi = \begin{cases} 0 & S^{\mathrm{w}} \leq S_{\mathrm{r}}^{\mathrm{w}}, S^{\mathrm{w}} \neq 1 \\ \left(\frac{S^{\mathrm{w}} - S_{\mathrm{r}}^{\mathrm{w}}}{1 - S_{\mathrm{r}}^{\mathrm{w}}}\right)^{\kappa} & S_{\mathrm{r}}^{\mathrm{w}} < S^{\mathrm{w}} < 1 \\ 1 & S^{\mathrm{w}} = 1 \end{cases} \tag{2}$$

Where κ and the residual saturation $S_{\mathrm{r}}^{\mathrm{w}}$ are material constants.

18.3 Compressibility Model

To use a nonlinear compressibility model, the material model 2 has to be chosen. If no additional input is made, it is assumed, that there is no preconsolidation of the material. Consequently the initial porosity equals the stress-free porosity. Additionally, some extended material models to define the initial conditions can be used by defining the subkeyword COMPRESSIBILIY_MODEL. For the definition of an initial condition by an initial porosity that differs from the stress-free porosity please refer to Example 18.5-3.

18.4 Isotropic continuum damage model

18.4.1 Damage criterion

The progressive degradation of mechanical properties due to damage is characterized by an isotropic mechanism. The state of damage in the material is then defined by a damage criterion in the following functional form

$$g(\overline{\tau}_t, r_t) = \overline{\tau}_t - r_t \leq 0. \tag{3}$$

The index t refers to the value at current time, $\overline{\tau}$ is the equivalent strain, and r defines the damage threshold (energy barrier). The criterion states that damage increases, if the equivalent strain exceeds the threshold. In the initial (undamaged) state the damage threshold r_{init} defines the damage initiation; later the relation $r_t \geq r_{\text{init}}$ has to be held. The equivalent strain $\overline{\tau}$ is introduced in order to define the damage criterion in a more general way as a function of the strains. For brittle damage mechanism, the equivalent strain can be defined with respect to a positive (tensile) or negative (compressive) state of strain. Two definitions are implemented:

$$\overline{\tau}_1 = \psi^{\text{el}}_{\text{init}}(\boldsymbol{\varepsilon}^{\text{el}}) = \frac{1}{2}\,\boldsymbol{\varepsilon}^{\text{el}} : \mathbf{C}_{\text{init}} : \boldsymbol{\varepsilon}^{\text{el}} \tag{4}$$

$$\overline{\tau}_2 = \sqrt{2\,\psi^{\text{el}}_{\text{init}}(\boldsymbol{\varepsilon}^{\text{el}}} = \sqrt{\boldsymbol{\varepsilon}^{\text{el}} : \mathbf{C}_{\text{init}} : \boldsymbol{\varepsilon}^{\text{el}}} \quad . \tag{5}$$

18.4.2 Damage evolution

Two models are implemented. In the *Marigo* model the damage variable can be stated as

$$D_t = 1 - \frac{r_{\text{init}}(1-A)}{r_t} - A\,\exp(B(r_{\text{init}} - r_t)), \quad \text{if} \quad r_t > r_{\text{init}}. \tag{6}$$

Another simple but general damage evolution law is given by *Marigo*:

$$\acute{D}_t = \left(\frac{r_t - r_{\text{init}}}{\acute{A}}\right)^{\acute{B}} \tag{7}$$

with the damage parameters r_{init}, $\acute{A}$, and $\acute{B}$.

18.5 Examples

Input example 1: Effective stress coefficient
(benchmarks/applications/MineByTest/Phase01/case011/test)

```
#SOLID_PROPERTIES ; linear elasticity
 1          ; mat.-model (elastic)
  18.0e9    ; elasticity_modul
  0.185     ; poissons_ratio

$EFFECTIVE_STRESS_COEFFICIENT ; coupling from hydraulics to
mechanics ;  Sigma_tot = Sigma_eff - 1 * BIOT *
effect_stress_coeff * pressure
                   ; model=0 =>  effect_stress_coeff = 0.0
                   ; model=1 =>  effect_stress_coeff = Saturation (default)
 2    ; model      ; model=2 =>  effect_stress_coeff = [(Satu-Satu_res)/(1-Satu_res)]^kappa
0.5  ; residual Saturation 1.5  ; kappa
```

Input example 2: Swelling model (applications/d_iv_phase01.rfd)

```
#SOLID_PROPERTIES
 1          ; mat.-model (elastic)
 15.0e9     ; elasticity_modul
 0.19       ; poissons_ratio

 ; SWELLING-----------------------------------------------------
 $MOISTURE_SWELLING
  1         ; type
  0.02      ; volumetric swelling coefficient (beta = delta_sw_strain / delta_Saturation)
  0.1 1.0   ; S_min, S_max (swelling range)
  1.0       ; refernece water saturation
  0.0       ; volumetric reference strain
 ; SWELLING-----------------------------------------------------

 $YIELD_FUNCTION_EVALUATION
  5         ; 5 = Mohr-Coulomb F=F(I1, J2, J3)
  1.4e06    ; cohesion (Pa)
  20        ; angle of internal friction ()]
```

Input example 3: Anisotropic non-linear swelling model
(benchmarks/shrinkage/exponential_anisotropic_swelling.rfd)

```
$MOISTURE_SWELLING
;exponential anisotropic swelling model
;For each direction (1,2,3) one power function defines the swelling strain.
;eps_sw_xx = eps_sw_0 + a[x] * (S^b[x] - S_0^b[x]); (x=1,2,3)
 13
 0.0022     ; a[1] material parameter a for curve 1
 0.0022     ; a[2] material parameter a for curve 2
 0.0096     ; a[3] material parameter a for curve 3
 4.5518     ; b[1] material parameter b for curve 1
 4.5518     ; b[2] material parameter b for curve 2
 9.3992     ; b[3] material parameter b for curve 3
 0.893      ;S_0 (reference water saturation )
 0.0       ;eps_sw_0 , initial swelling strain (corresponding to reference water saturation)
 0         ;number of coordinate system(0=global)
; SWELLING-----------------------------------------------------
```

Input example 4: Rotating transversely isotropic elasticity (benchmarks/thm_plus_m/aniso_tensile_plate_eng_e1.rfd)

```
#SOLID_PROPERTIES
10 1                                        ; a=e1
   1311829787.2340    561121223.15253       ; E1, E2 = E3
   0.18383691299600   0.60320349472151      ; nue12, nue23
   375000000.0                              ; G12
   1                                        ; coordinate system

#CURVES
  0.0   0.0
180.0 180.0

#COORDINATE_SYSTEM
 0          ; method axis
 1          ; method rotation
  1         ; number of rotation axes
  Z { CURVE 1 }   ; axis, angle
 0          ; method translation
```

Input example 5: Compressibility model (deformation/nl_comp_consol_diff_n.rfd)

```
#SOLID_PROPERTIES
  2              ;mat_model (nonlinear compressible elasticity)
  2.50000e+005   ;elasticity_modul
  2.000000e-001  ;poissons_ratio

 $COMPRESSIBILITY_MODEL
   1             ;compressibility_model
   0.15          ;stress-free porosity

#SOIL_PROPERTIES
  .
  .
  .
  4              ;porosity_model
  0.05           ;initial_porosity
  .
  .
  .
```

Input example 6: Isotropic damage model (damage/damage_desaturation.rfd)

```
#SOLID_PROPERTIES
1001     ; model (isotropic elasticity with damage)
16.73e9  ; E-Modul
0.185    ; Poissons_ratio
1        ; damage model (1=scalar)
1        ; damage criterion
1.0      ; damage mechanism (in case of 1, only tensile strains cause damage)
10000.   ; para_0; inititial damage threshold
1        ; damage evolution model (1=Marigo)
4.e5     ; para_1; damage evolution parameter A
0.2      ; para_2; damage evolution parameter B
0.99     ; para_3; max damage state
0.       ; para_4; microcrack opening/closing mechanism (0=inactive)
0.01     ; para 5; mue, damage viscosity coefficient
```

Input example 7: Transversely isotropic elasticity with isotropic damage considering microcrack opening-closing

```
#SOLID_PROPERTIES
1010         ; model (transversely isotropic elasticity with isotropic damage)
   2         ; a = e_2
   9.27e9    ; elasticity_modul in direction 2 [Pa]
  24.19e9    ; elasticity_modul in plane of isotropy 1  [Pa]
  0.20       ; poissons_ratio_21
  0.17       ; poissons_ratio_12
  3.94e9     ; shear modulus_21 [Pa]
  1          ; number of coordinate system (0=global)
1        ; damage model
1        ; damage criterion
1.0      ; damage mechanism (in case of 1, only tensile strains cause damage)
20000.   ; para_0; initial damage threshold
1        ; damage evolution model (1=Marigo)
2.e6     ; para_1; damage evolution parameter A
0.2      ; para_2; damage evolution parameter B
0.99     ; para_3; max damage state
1.       ; para_4; microcrack opening/closing mechanism (1=active)
0.001    ; para 5; mue, damage viscosity coefficient
```

19 Component Properties

19.1 Keyword Description

Tracer properties (chemical species) can be specified using this keyword.

#COMPONENT_PROPERTIES			
Parameter	RF-Variable	Values	Meaning
if next value!=[$]			Es kann entweder eine Parameterliste folgen oder eine Definition über Sybkeywords
N1	.diffusion_- model	[int]	diffusion model 0: No diffusion 1: Constant value for diffusion 2: Variable diffusion (timedependent) 3: Diffusionskoeffizienten in Wasser Daq [m^2/s] Worch, 1993 4: Diffusionskoeffizienten in Wasser Daq [m^2/s] Hayduk und Laudie, 1974 5: Diffusionskoeffizienten in Wasser Daq [m^2/s] Wilke und Chang, 1955 6: Diffusionskoeffizienten in Wasser Daq [m^2/s] Stokes−Einstein (fuer Partikel/Makromolekuele) 7: Diffusionskoeffizienten in Luft Dg [m^2/s] FSG−Methode, Lyman et al., 1990
if N1=0			No diffusion
if N1=1			Constant value for diffusion
N2	.diffusion_- values_0	[double]	Diffusion value [>=0;0]
if N1=2			Variable diffusion (timedependent)
N3	.diffusion_- values_0	[double]	Diffusion value [>=0;0]
if N1=3			Diffusionskoeffizienten in Wasser Daq [m^2/s] Worch, 1993
N4	.diffusion_- values_0	[double]	Diffusion value [>=0;0]
if N1=4			Diffusionskoeffizienten in Wasser Daq [m^2/s] Hayduk und Laudie, 1974
N5	.diffusion_- values_0	[double]	Diffusion value [>=0;0]
if N1=5			Diffusionskoeffizienten in Wasser Daq [m^2/s] Wilke und Chang, 1955
N6	.diffusion_- values_0_1	[2* double]	Diffusion values [>=0;0]
if N1=6			Diffusionskoeffizienten in Wasser Daq [m^2/s] Stokes−Einstein (fuer Partikel/Makromolekuele)
N7	.diffusion_- values_0	[double]	Diffusion value [>=0;0]
if N1=7			Diffusionskoeffizienten in Luft Dg [m^2/s] FSG−Methode, Lyman et al., 1990
N8	.diffusion_- values_0_1	[2* double]	Diffusion values [>=0;0]

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N9	.soluted_-decay_model	[int]	soluted decay model [>=0;0] 0: No Decay 1: First Order Decay with constant Decay Rate 2: First Order Decay with variable Decay Rate 3: Second Order Decay with constant Decay Rate 4: Second Order Decay with variable Decay Rate 5: Monod or Michaelis−Menten kinetics constant rate coefficients 6: Monod or Michaelis−Menten kinetics variable rate coefficients 7: First Order Decay with constant Decay Rate and constant fration of parent that form the daughter product
if N9=0			No Decay
if N9=1			First Order Decay with constant Decay Rate
if N-1=10095			if rockflow model = 10095
N10	.soluted_-decay_-model_-values_0_1	[2* double]	decay parameter [>=0;0]
if N-1!=10095			if rf model != 10095
N11	.soluted_-decay_-model_-values_0	[double]	decay parameter [>=0;0]
if N9=2			First Order Decay with variable Decay Rate
if N-1=10095			if rf model = 10095
N12	.soluted_-decay_-model_-values_0_1_2	[3* double]	decay parameter [>=0;0]
if N-1!=10095			if rf model != 10095
N13	.soluted_-decay_-model_-values_0_1	[2* double]	decay parameter [>=0;0]
if N9=3			Second Order Decay with constant Decay Rate
if N-1=10095			if rf model = 10095
N14	.soluted_-decay_-model_-values_0_1_2	[3* double]	decay parameter [>=0;0]
if N-1!=10095			if rf model != 10095
N15	.soluted_-decay_-model_-values_0_1	[2* double]	decay parameter [>=0;0]
if N9=4			Second Order Decay with variable Decay Rate
if N-1=10095			if rf model = 10095
N16	.soluted_-decay_-model_-values_0_3	[4* double]	decay parameter [>=0;0]
if N-1!=10095			if rf model != 10095

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N17	.soluted_-decay_-model_-values_0_1_2	[3* double]	decay parameter [>=0;0]
if N9=5			Monod or Michaelis−Menten kinetics constant rate coefficients
if N-1=10095			if rf model = 10095
N18	.soluted_-decay_-model_-values_0_1_2	[3* double]	decay parameter [>=0;0]
if N-1!=10095			if rf model != 10095
N19	.soluted_-decay_-model_-values_0_1	[2* double]	decay parameter [>=0;0]
if N9=6			Monod or Michaelis−Menten kinetics variable rate coefficients
if N-1=10095			if rf model = 10095
N20	.soluted_-decay_-model_-values_0_3	[4* double]	decay parameter [>=0;0]
if N-1!=10095			if rf model != 10095
N21	.soluted_-decay_-model_-values_0_1_2	[3* double]	decay parameter [>=0;0]
if N9=7			First Order Decay with constant Decay Rate and constant fration of parent that form the daughter product
if N-1=10095			if rf model = 10095
N22	.soluted_-decay_-model_-values_0_1_2	[3* double]	decay parameter [>=0;0]
if N-1!=10095			if rf model != 10095
N23	.soluted_-decay_-model_-values_0_1	[2* double]	decay parameter [>=0;0]
N24	.sorbed_-decay_model	[int]	sorbed decay model [>=0;0] 0: No Decay 1: First Order Decay with constant Decay Rate 2: First Order Decay with variable Decay Rate 3: Second Order Decay with constant Decay Rate 4: Second Order Decay with variable Decay Rate 5: Monod or Michaelis−Menten kinetics constant rate coefficients 6: Monod or Michaelis−Menten kinetics variable rate coefficients 7: First Order Decay with constant Decay Rate and constant fration of parent that form the daughter product

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
if N24=0			No Decay
if N24=1			First Order Decay with constant Decay Rate
if N-1=10095			rf model = 10095
N25	.sorbed_-decay_-model_-values_0_1	[2* double]	decay parameter [>=0;0]
if N-1!=10095			rf model != 10095
N26	.sorbed_-decay_-model_-values_0	[double]	decay parameter [>=0;0]
if N24=2			First Order Decay with variable Decay Rate
if N-1=10095			rf model = 10095
N27	.sorbed_-decay_-model_-values_0_1_2	[3* double]	decay parameter [>=0;0]
if N-1!=10095			rf model != 10095
N28	.sorbed_-decay_-model_-values_0_1	[2* double]	decay parameter [>=0;0]
if N24=3			Second Order Decay with constant Decay Rate
if N-1=10095			rf model = 10095
N29	.sorbed_-decay_-model_-values_0_1_2	[3* double]	decay parameter [>=0;0]
if N-1!=10095			rf model != 10095
N30	.sorbed_-decay_-model_-values_0_1	[2* double]	decay parameter [>=0;0]
if N24=4			Second Order Decay with variable Decay Rate
if N-1=10095			rf model = 10095
N31	.sorbed_-decay_-model_-values_0_3	[4* double]	decay parameter [>=0;0]
if N-1!=10095			rf model != 10095
N32	.sorbed_-decay_-model_-values_0_1_2	[3* double]	decay parameter [>=0;0]
if N24=5			Monod or Michaelis−Menten kinetics constant rate coefficients
if N-1=10095			rf model = 10095
N33	.sorbed_-decay_-model_-values_0_1_2	[3* double]	decay parameter [>=0;0]
if N-1!=10095			rf model != 10095

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N34	.sorbed_-decay_-model_-values_0_1	[2* double]	decay parameter [>=0;0]
if N24=6			Monod or Michaelis−Menten kinetics variable rate coefficients
if N-1=10095			rf model = 10095
N35	.sorbed_-decay_-model_-values_0_3	[4* double]	decay parameter [>=0;0]
if N-1!=10095			rf model != 10095
N36	.sorbed_-decay_-model_-values_0_1_2	[3* double]	decay parameter [>=0;0]
if N24=7			First Order Decay with constant Decay Rate and constant fration of parent that form the daughter product
if N-1=10095			rf model = 10095
N37	.sorbed_-decay_-model_-values_0_1_2	[3* double]	decay parameter [>=0;0]
if N-1!=10095			rf model != 10095
N38	.sorbed_-decay_-model_-values_0_1	[2* double]	decay parameter [>=0;0]
N39	.isotherm_-model	[int]	isotherm model [>=0;0] 0: No Isotherm 1: Henry Isotherm 2: Freundlich Isotherm 3: Langmuir Isotherm 4: Freundlich Langmuir Isotherm 5: Double Langmuir Isotherm 6: Extented Freundlich Isotherm 7: Gunary Isotherm 8: Fitter−Sutton Isotherm 9: Barry Isotherm 10: Power Isotherm 11: Modified Kielland Isotherm
if N39=0			No Isotherm
if N39=1			Henry Isotherm
N40	.isotherm_-model_-values_0	[double]	isotherm values k[] [>=0;0]
if N39=2			Freundlich Isotherm
N41	.isotherm_-model_-values_0_1	[2* double]	isotherm values k[] [>=0;0]
if N39=3			Langmuir Isotherm

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N42	.isotherm_-model_-values_0_1	[2* double]	isotherm values k[] [>=0;0]
if N39=4			Freundlich Langmuir Isotherm
N43	.isotherm_-model_-values_0_1_2	[3* double]	isotherm values k[] [>=0;0]
if N39=5			Double Langmuir Isotherm
N44	.isotherm_-model_-values_0_3	[4* double]	isotherm values k[] [>=0;0]
if N39=6			Extented Freundlich Isotherm
N45	.isotherm_-model_-values_0_1_2	[3* double]	isotherm values k[] [>=0;0]
if N39=7			Gunary Isotherm
N46	.isotherm_-model_-values_0_1_2	[3* double]	isotherm values k[] [>=0;0]
if N39=8			Fitter−Sutton Isotherm
N47	.isotherm_-model_-values_0_1_2	[3* double]	isotherm values k[] [>=0;0]
if N39=9			Barry Isotherm
N48	.isotherm_-model_-values_0_3	[4* double]	isotherm values k[] [>=0;0]
if N39=10			Power Isotherm
N49	.isotherm_-model_-values_0_1	[2* double]	isotherm values k[] [>=0;0]
if N39=11			Modified Kielland Isotherm
N50	.isotherm_-model_-values_0_1	[2* double]	isotherm values k[] [>=0;0]
N51	.chemical_-nonequilibrium_-model	[int]	Chemical nonequilibrium model [>=0;0] 0: Chemical equilibrium model or chemical no nonequilibrium model 1: First order reaction 2: n−th order reaction 3: reaction with langmuir kinetik 4: non reversible reaction 5: power kinetik 6: first order reaction (zweiparametrig) 7: n−th order reaction (zweiparametrig)
if N51=0			Chemical equilibrium model or chemical no nonequilibrium model
if N51=1			First order reaction
N52	.chemical_-nonequilibrium_-model_-values_0	[double]	Chemical nonequilibrium model values [>=0;0]
if N51=2			n−th order reaction

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N53	.chemical_-nonequilibrium_-model_-values_0_1	[2* double]	Chemical nonequilibrium model values [>=0;0]
if N51=3			reaction with langmuir kinetik
N54	.chemical_-nonequilibrium_-model_-values_0_1	[2* double]	Chemical nonequilibrium model values [>=0;0]
if N51=4			non reversible reaction
N55	.chemical_-nonequilibrium_-model_-values_0_1	[2* double]	Chemical nonequilibrium model values [>=0;0]
if N51=5			power kinetik
N56	.chemical_-nonequilibrium_-model_-values_0_1	[2* double]	Chemical nonequilibrium model values [>=0;0]
if N51=6			first order reaction (zweiparametrig)
N57	.chemical_-nonequilibrium_-model_-values_0_1	[2* double]	Chemical nonequilibrium model values [>=0;0]
if N51=7			n−th order reaction (zweiparametrig)
N58	.chemical_-nonequilibrium_-model_-values_0_1_2	[3* double]	Chemical nonequilibrium model values [>=0;0]
N59	.physical_-nonequilibrium_-model	[int]	Physical nonequilibrium model [>=0;0] 0: Physical equilibrium model or physical no nonequilibrium model 1: Coats + Smith (first order reaction with konstant transfer coeffizient) 2: Coats + Smith (first order reaction with variable transfer coeffizient)
if N59=0			Physical equilibrium model or physical no nonequilibrium model
if N59=1			Coats + Smith (first order reaction with konstant transfer coeffizient)
N60	.physical_-nonequilibrium_-model_-values_0	[double]	Physical nonequilibrium model values [>=0;0]
if N59=2			Coats + Smith (first order reaction with variable transfer coeffizient)
N61	.physical_-nonequilibrium_-model_-values_0_1	[2* double]	Physical nonequilibrium model values [>=0;0]
N62	.solubility_-model	[int]	solubility model 0: No solution 1: Salzloesung
if N62=0			No solution

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
if N62=1			Salzloesung
N63	.solubility_-dependence_-model	[int]	Loesungsmodell Salzloesung: 0: konstante Saettigungskonzentration 1: Saettigung druckabhaengig 2: Saettigung temperaturabhaengig 3: Saettigung druck− und temperaturabhaengig
N64	.solubility	[double]	Saettigung fuer die Loesungs− , Faellungsreaktion [>=0;0]
N65	.dissolution_-rate	[double]	Loesungs−, Faellungsrate [>=0;0]

$DIFFUSION			
N66	.diffusion_-model	[int]	diffusion model 0: No diffusion 1: Constant value for diffusion 2: Variable diffusion (timedependent) 3: Diffusionskoeffizienten in Wasser Daq [m^2/s] Worch, 1993 4: Diffusionskoeffizienten in Wasser Daq [m^2/s] Hayduk und Laudie, 1974 5: Diffusionskoeffizienten in Wasser Daq [m^2/s] Wilke und Chang, 1955 6: Diffusionskoeffizienten in Wasser Daq [m^2/s] Stokes−Einstein (fuer Partikel/Makromolekuele) 7: Diffusionskoeffizienten in Luft Dg [m^2/s] FSG−Methode, Lyman et al., 1990
if N66=0			No diffusion
if N66=1			Constant value for diffusion
N67	.diffusion_-model_-values_0	[double]	Diffusion values [>=0;0]
if N66=2			Variable diffusion (timedependent)
N68	.diffusion_-model_-values_0	[double]	Diffusion values [>=0;0]
if N66=3			Diffusionskoeffizienten in Wasser Daq [m^2/s] Worch, 1993
N69	.diffusion_-model_-values_0	[double]	Diffusion values [>=0;0]
if N66=4			Diffusionskoeffizienten in Wasser Daq [m^2/s] Hayduk und Laudie, 1974
N70	.diffusion_-model_-values_0	[double]	Diffusion values [>=0;0]
if N66=5			Diffusionskoeffizienten in Wasser Daq [m^2/s] Wilke und Chang, 1955
N71	.diffusion_-model_-values_0_1	[2* double]	Diffusion values [>=0;0]
if N66=6			Diffusionskoeffizienten in Wasser Daq [m^2/s] Stokes−Einstein (fuer Partikel/Makromolekuele)

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N72	.diffusion_-model_-values_0	[double]	Diffusion values [>=0;0]
if N66=7			Diffusionskoeffizienten in Luft Dg [m^2/s] FSG−Methode, Lyman et al., 1990
N73	.diffusion_-model_-values_0_1	[2* double]	Diffusion values [>=0;0]

$DECAY_AQUEOUS			
N74	.soluted_-decay_model	[int]	soluted decay model [>=0;0] 0: No Decay 1: First Order Decay with constant Decay Rate 2: First Order Decay with variable Decay Rate 3: Second Order Decay with constant Decay Rate 4: Second Order Decay with variable Decay Rate 5: Monod or Michaelis−Menten kinetics constant rate coefficients 6: Monod or Michaelis−Menten kinetics variable rate coefficients 7: First Order Decay with constant Decay Rate and constant fration of parent that form the daughter product
if N74=0			No Decay
if N74=1			First Order Decay with constant Decay Rate
if N-1=10095			if rockflow model = 10095
N75	.soluted_-decay_-model_-values_0_1	[2* double]	decay parameter [>=0;0]
if N-1!=10095			if rf model != 10095
N76	.soluted_-decay_-model_-values_0	[double]	decay parameter [>=0;0]
if N74=2			First Order Decay with variable Decay Rate
if N-1=10095			if rf model = 10095
N77	.soluted_-decay_-model_-values_0_1_2	[3* double]	decay parameter [>=0;0]
if N-1!=10095			if rf model != 10095
N78	.soluted_-decay_-model_-values_0_1	[2* double]	decay parameter [>=0;0]
if N74=3			Second Order Decay with constant Decay Rate
if N-1=10095			if rf model = 10095
N79	.soluted_-decay_-model_-values_0_1_2	[3* double]	decay parameter [>=0;0]
if N-1!=10095			if rf model != 10095

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N80	.soluted_-decay_-model_-values_0_1	[2* double]	decay parameter [>=0;0]
if N74=4			Second Order Decay with variable Decay Rate
if N-1=10095			if rf model = 10095
N81	.soluted_-decay_-model_-values_0_3	[4* double]	decay parameter [>=0;0]
if N-1!=10095			if rf model != 10095
N82	.soluted_-decay_-model_-values_0_1_2	[3* double]	decay parameter [>=0;0]
if N74=5			Monod or Michaelis−Menten kinetics constant rate coefficients
if N-1=10095			if rf model = 10095
N83	.soluted_-decay_-model_-values_0_1_2	[3* double]	decay parameter [>=0;0]
if N-1!=10095			if rf model != 10095
N84	.soluted_-decay_-model_-values_0_1	[2* double]	decay parameter [>=0;0]
if N74=6			Monod or Michaelis−Menten kinetics variable rate coefficients
if N-1=10095			if rf model = 10095
N85	.soluted_-decay_-model_-values_0_3	[4* double]	decay parameter [>=0;0]
if N-1!=10095			if rf model != 10095
N86	.soluted_-decay_-model_-values_0_1_2	[3* double]	decay parameter [>=0;0]
if N74=7			First Order Decay with constant Decay Rate and constant fration of parent that form the daughter product
if N-1=10095			if rf model = 10095
N87	.soluted_-decay_-model_-values_0_1_2	[3* double]	decay parameter [>=0;0]
if N-1!=10095			if rf model != 10095
N88	.soluted_-decay_-model_-values_0_1	[2* double]	decay parameter [>=0;0]

$DECAY_SORPTIVE

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N89	.sorbed_-decay_model	[int]	sorbed decay model [>=0;0] 0: No Decay 1: First Order Decay with constant Decay Rate 2: First Order Decay with variable Decay Rate 3: Second Order Decay with constant Decay Rate 4: Second Order Decay with variable Decay Rate 5: Monod or Michaelis−Menten kinetics constant rate coefficients 6: Monod or Michaelis−Menten kinetics variable rate coefficients 7: First Order Decay with constant Decay Rate and constant fration of parent that form the daughter product
if N89=0			No Decay
if N89=1			First Order Decay with constant Decay Rate
if N-1=10095			rf model = 10095
N90	.sorbed_-decay_-model_-values_0_1	[2* double]	decay parameter [>=0;0]
if N-1!=10095			rf model != 10095
N91	.sorbed_-decay_-model_-values_0	[double]	decay parameter [>=0;0]
if N89=2			First Order Decay with variable Decay Rate
if N-1=10095			rf model = 10095
N92	.sorbed_-decay_-model_-values_0_1_2	[3* double]	decay parameter [>=0;0]
if N-1!=10095			rf model != 10095
N93	.sorbed_-decay_-model_-values_0_1	[2* double]	decay parameter [>=0;0]
if N89=3			Second Order Decay with constant Decay Rate
if N-1=10095			rf model = 10095
N94	.sorbed_-decay_-model_-values_0_1_2	[3* double]	decay parameter [>=0;0]
if N-1!=10095			rf model != 10095
N95	.sorbed_-decay_-model_-values_0_1	[2* double]	decay parameter [>=0;0]
if N89=4			Second Order Decay with variable Decay Rate
if N-1=10095			rf model = 10095
N96	.sorbed_-decay_-model_-values_0_3	[4* double]	decay parameter [>=0;0]

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
if N-1!=10095			rf model != 10095
N97	.sorbed_-decay_-model_-values_0_1_2	[3* double]	decay parameter [>=0;0]
if N89=5			Monod or Michaelis−Menten kinetics constant rate coefficients
if N-1=10095			rf model = 10095
N98	.sorbed_-decay_-model_-values_0_1_2	[3* double]	decay parameter [>=0;0]
if N-1!=10095			rf model != 10095
N99	.sorbed_-decay_-model_-values_0_1	[2* double]	decay parameter [>=0;0]
if N89=6			Monod or Michaelis−Menten kinetics variable rate coefficients
if N-1=10095			rf model = 10095
N100	.sorbed_-decay_-model_-values_0_3	[4* double]	decay parameter [>=0;0]
if N-1!=10095			rf model != 10095
N101	.sorbed_-decay_-model_-values_0_1_2	[3* double]	decay parameter [>=0;0]
if N89=7			First Order Decay with constant Decay Rate and constant fration of parent that form the daughter product
if N-1=10095			rf model = 10095
N102	.sorbed_-decay_-model_-values_0_1_2	[3* double]	decay parameter [>=0;0]
if N-1!=10095			rf model != 10095
N103	.sorbed_-decay_-model_-values_0_1	[2* double]	decay parameter [>=0;0]

$ISOTHERM

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N104	.isotherm_-model	[int]	isotherm model [>=0;0] 0: No Isotherm 1: Henry Isotherm 2: Freundlich Isotherm 3: Langmuir Isotherm 4: Freundlich Langmuir Isotherm 5: Double Langmuir Isotherm 6: Extented Freundlich Isotherm 7: Gunary Isotherm 8: Fitter−Sutton Isotherm 9: Barry Isotherm 10: Power Isotherm 11: Modified Kielland Isotherm
if N104=0			No Isotherm
if N104=1			Henry Isotherm
N105	.isotherm_-model_-values_0	[double]	isotherm values k[] [>=0;0]
if N104=2			Freundlich Isotherm
N106	.isotherm_-model_-values_0_1	[2* double]	isotherm values k[] [>=0;0]
if N104=3			Langmuir Isotherm
N107	.isotherm_-model_-values_0_1	[2* double]	isotherm values k[] [>=0;0]
if N104=4			Freundlich Langmuir Isotherm
N108	.isotherm_-model_-values_0_1_2	[3* double]	isotherm values k[] [>=0;0]
if N104=5			Double Langmuir Isotherm
N109	.isotherm_-model_-values_0_3	[4* double]	isotherm values k[] [>=0;0]
if N104=6			Extented Freundlich Isotherm
N110	.isotherm_-model_-values_0_1_2	[3* double]	isotherm values k[] [>=0;0]
if N104=7			Gunary Isotherm
N111	.isotherm_-model_-values_0_1_2	[3* double]	isotherm values k[] [>=0;0]
if N104=8			Fitter−Sutton Isotherm
N112	.isotherm_-model_-values_0_1_2	[3* double]	isotherm values k[] [>=0;0]
if N104=9			Barry Isotherm
N113	.isotherm_-model_-values_0_3	[4* double]	isotherm values k[] [>=0;0]
if N104=10			Power Isotherm

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N114	.isotherm_-model_-values_0_1	[2* double]	isotherm values k[] [>=0;0]
if N104=11			Modified Kielland Isotherm
N115	.isotherm_-model_-values_0_1	[2* double]	isotherm values k[] [>=0;0]

$NONEQUILIBRIUM_CHEMICAL			
N116	.chemical_-nonequilibrium_-model	[int]	Chemical nonequilibrium model [>=0;0] 0: Chemical equilibrium model or chemical no nonequilibrium model 1: First order reaction 2: n−th order reaction 3: reaction with langmuir kinetik 4: non reversible reaction 5: power kinetik 6: first order reaction (zweiparametrig) 7: n−th order reaction (zweiparametrig)
if N116=0			Chemical equilibrium model or chemical no nonequilibrium model
if N116=1			First order reaction
N117	.chemical_-nonequilibrium_-model_-values_0	[double]	Chemical nonequilibrium model values [>=0;0]
if N116=2			n−th order reaction
N118	.chemical_-nonequilibrium_-model_-values_0_1	[2* double]	Chemical nonequilibrium model values [>=0;0]
if N116=3			reaction with langmuir kinetik
N119	.chemical_-nonequilibrium_-model_-values_0_1	[2* double]	Chemical nonequilibrium model values [>=0;0]
if N116=4			non reversible reaction
N120	.chemical_-nonequilibrium_-model_-values_0_1	[2* double]	Chemical nonequilibrium model values [>=0;0]
if N116=5			power kinetik
N121	.chemical_-nonequilibrium_-model_-values_0_1	[2* double]	Chemical nonequilibrium model values [>=0;0]
if N116=6			first order reaction (zweiparametrig)
N122	.chemical_-nonequilibrium_-model_-values_0_1	[2* double]	Chemical nonequilibrium model values [>=0;0]
if N116=7			n−th order reaction (zweiparametrig)

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N123	.chemical_-nonequilibrium_-model_-values_0_1_2	[3* double]	Chemical nonequilibrium model values [>=0;0]

$NONEQUILIBRIUM_PHYSICAL			
N124	.physical_-nonequilibrium_-model	[int]	Physical nonequilibrium model [>=0;0] 0: Physical equilibrium model or physical no nonequilibrium model 1: Coats + Smith (first order reaction with konstant transfer coeffizient) 2: Coats + Smith (first order reaction with variable transfer coeffizient)
if N124=0			Physical equilibrium model or physical no nonequilibrium model
if N124=1			Coats + Smith (first order reaction with konstant transfer coeffizient)
N125	.physical_-nonequilibrium_-model_-values_0	[double]	Physical nonequilibrium model values [>=0;0]
if N124=2			Coats + Smith (first order reaction with variable transfer coeffizient)
N126	.physical_-nonequilibrium_-model_-values_0_1	[2* double]	Physical nonequilibrium model values [>=0;0]

$SOLUTION_PRECIPITATION			
N127	.solubility_-model	[int]	solubility model 0: No solution 1: Salzloesung
if N127=0			No solution
if N127=1			Salzloesung
N128	.solubility_-dependence_-model	[int]	Loesungsmodell Salzloesung: 0: konstante Saettigungskonzentration 1: Saettigung druckabhaengig 2: Saettigung temperaturabhaengig 3: Saettigung druck− und temperaturabhaengig
N129	.solubility	[double]	Saettigung fuer die Loesungs− , Faellungsreaktion [>=0;0]
N130	.dissolution_-rate	[double]	Loesungs−, Faellungsrate [>=0;0]

19.2 Example

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)
```

20 Grid Adaptation

This menu is used to specify the required parameters for controling the automatic grid adaptation. To activiate grid adaptation, the adaptive_mesh_refinement_flag has to set (see MODELS).

20.1 Keyword Description

The following keywords are available to specify grid adaptation parameters for field functions.

Keywords	Name variables
#ADAPTATION	General control parameter
#ADAPTATION_PRESSURE	Quantity parameter for fluid pressure
#ADAPTATION_COMPONENT	Quantity parameter for components
#ADAPTATION_SATURATION	Quantity parameter for fluid saturation
#ADAPTATION_SORBED_COMPONENT	Quantity parameter for sorbed components
#ADAPTATION_SOLUTE_COMPONENT	Quantity parameter for soluted components
#ADAPTATION_TEMPERATURE	Quantity parameter for temperature

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

20.1.1 Adaptation Control Parameter

#ADAPTATION

#ADAPTATION			
Parameter	RF-Variable	Values	Meaning
N1	.method_-adaptation	[int]	Adaptation loop at time level: 0: no adaptation 1: calculation – adaptation – ... – calculation, next timestep 2: adaptation – calculation – ... – calculation, next timestep 3: calculation – adaptation – ... – calculation, next timestep 4: adaptation – calculation – ... – calculation, next timestep 5: first time step methode 1, then methode 2 6: first time step methode 3, then Methode 4
if N1=3 ,4 ,6			
N2	.adaptation_-timestep	[int]	Time step number interval for adaptation
N3	.max_ref_level	[int]	Maximum refinement level [>=0;0] Wenn ein Element schon max_ref_level−mal verfeinert wurde, wird es auf keinen Fall weiter verfeinert.
N4	.timestep_-ref_number	[int]	Maximum refinement at time step
N5	.method_irr_-nodes	[int]	Regularization method 1: at element leve 2: at equation system level

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N6	.ref_-neighbours_-2d	[int]	Number of neighbours (2−D elements) to be refined (secondary criterium) [1..4;4] Der Wert hat bei reiner 1D−Verfeinerung keinen Einfluss.
N7	.ref_-neighbours_-3d	[int]	Number of neighbours (3−D elements) to be refined (secondary criterium) [1..6;6] Der Wert hat bei reiner 1D−Verfeinerung keinen Einfluss.
N8	.method_-node_limiter	[int]	Method for node limitation 0: No limitations 1: Limitation 1 2: Limitation 2
if N8=0			No limitations
if N8=1			Limitation 1
N9	.curve_node_-limiter	[int]	Time curve number
N10	.max_nodes	[int]	Maximum number of allowed nodes during grid adaptation < 0 Indikatoren werden variiert, um x Knoten nicht zu ueberschreiten 0 no limitations > 0 Indikatoren werden variiert, um x Knoten zu erhalten
if N8=2			Limitation 2
N11	.curve_node_-limiter	[int]	Time curve number
N12	.max_nodes	[int]	Maximum number of allowed nodes during grid adaptation
N13	.output_-adaptation	[int]	Extended output information during simulation

20.1.2 Adaptation Quantity Parameter

#ADAPTATION_PRESSURE

#ADAPTATION_COMPONENT

#ADAPTATION_SATURATION

#ADAPTATION_SORBED_COMPONENT

#ADAPTATION_SOLUTE_COMPONENT

#ADAPTATION_TEMPERATURE

#ADAPTATION_PRESSURE			
Parameter	RF-Variable	Values	Meaning
N1	.method_ind	[int]	Method indikator 0: Defaultwerte Es muessen keine weiteren Werte eingelesen werden!! 1: Standard 1 : For analytical estimators Unterscheidung zwischen Diffusion und Advektion bei Transportprozessen (Verwendung analytischer oder heuristischer Verfeinerungsindikatoren moeglich) 2: Standard 2: Identical parameter for both diffusion and dominated elements Keine Unterscheidung zwischen Diffusion und Advektion bei Transportprozessen (heuristische Verfeinerungsindikatoren bei Transportprozessen) (heuristische und analytische Verfeinerungsindikatoren bei Stroemung) 3: Professional: Identical parameter for both diffusion and dominated elements special features for time control
if N1=0			Defaultwerte Es muessen keine weiteren Werte eingelesen werden!!
if N1=1			Standard 1 : For analytical estimators Unterscheidung zwischen Diffusion und Advektion bei Transportprozessen (Verwendung analytischer oder heuristischer Verfeinerungsindikatoren moeglich)
N2	.coarsement_correction	[int]	Coarsement correction [0,1]
N3	.number_ind	[long]	Number of indicator data blocks
N-1 * Loop			
N4	.ref_tol	[int]	Tolerance parameter [0,1,2] 0: based on absolute values 1: based on average values 2: based on maximum values
N5	.ref_ind_0_0	[int]	Indicator for 1−D diffusion elements
N6	.ref_param_0_0_0	[double]	Parameter for 1−D diffusion elements: refinement border
N7	.ref_param_0_0_1	[double]	Parameter for 1−D diffusion elements: coarsement border
N8	.ref_param_0_0_2	[double]	Not used
N9	.ref_ind_0_1	[int]	Indicator for 1−D advektive elements

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N10	.ref_param_-0_1_0	[double]	Parameter for 1−D advektive elements: refinement border
N11	.ref_param_-0_1_1	[double]	Parameter for 1−D advektive elements: coarsement border
N12	.ref_param_-0_1_2	[double]	Not used
N13	.ref_ind_1_0	[int]	Indicator for 2−D diffusive elements
N14	.ref_param_-1_0_0	[double]	Parameter for 2−D diffusive elements: refinement border
N15	.ref_param_-1_0_1	[double]	Parameter for 1−D diffusive elements: coarsement border
N16	.ref_param_-1_0_2	[double]	Not used
N17	.ref_ind_1_1	[int]	Indicator for 2−D advektive elements
N18	.ref_param_-1_1_0	[double]	Parameter for 2−D advektive elements: refinement border
N19	.ref_param_-1_1_1	[double]	Parameter for 2−D advektive elements: coarsement border
N20	.ref_param_-1_1_2	[double]	Not used
N21	.ref_ind_2_0	[int]	Indicator for 3−D diffusive elements
N22	.ref_param_-2_0_0	[double]	Parameter for 3−D diffusive elements: refinement border
N23	.ref_param_-2_0_1	[double]	Parameter for 3−D diffusive elements: coarsement border
N24	.ref_param_-2_0_2	[double]	Not used
N25	.ref_ind_2_1	[int]	Indicator for 3−D advektive elements
N26	.ref_param_-2_1_0	[double]	Parameter for 3−D advektive elements: refinement border
N27	.ref_param_-2_1_1	[double]	Parameter for 3−D advektive elements: coarsement border
N28	.ref_param_-2_1_2	[double]	Not used
if N1=2			Standard 2: Identical parameter for both diffusion and dominated elements Keine Unterscheidung zwischen Diffusion und Advektion bei Transportprozessen (heuristische Verfeinerungsindikatoren bei Transportprozessen) (heuristische und analytische Verfeinerungsindikatoren bei Stroemung)
N29	.coarsement_-correction	[int]	Coarsement correction [0,1]
N30	.number_ind	[long]	Number of indicator data blocks
N-1 * Loop			
N31	.ref_tol	[int]	Tolerance parameter [0,1,2] 0: based on absolute values 1: based on average values 2: based on maximum values
N32	.ref_ind_0_0	[int]	Indicator for 1−D elements
N33	.ref_param_-0_0_0	[double]	Parameter for 1−D elements: refinement border

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N34	.ref_param_-0_0_1	[double]	Parameter for 1−D elements: coarsement border
N35	.ref_param_-0_0_2	[double]	Not used
N36	.ref_ind_1_0	[int]	Indicator for 2−D elements
N37	.ref_param_-1_0_0	[double]	Parameter for 2−D elements: refinement border
N38	.ref_param_-1_0_1	[double]	Parameter for 2−D elements: coarsement border
N39	.ref_param_-1_0_2	[double]	Not used
N40	.ref_ind_2_0	[int]	Indicator for 3−D elements
N41	.ref_param_-2_0_0	[double]	Parameter for 3−D elements: refinement border
N42	.ref_param_-2_0_1	[double]	Parameter for 3−D elements: coarsement border
N43	.ref_param_-2_0_2	[double]	Not used
if N1=3			Professional: Identical parameter for both diffusion and dominated elements special features for time control
N44	.quantity_-max_ref_-level_0	[int]	Maximum refinement level for 1−D elements
N45	.quantity_-max_ref_-level_1	[int]	Maximum refinement level for 2−D elements
N46	.quantity_-max_ref_-level_2	[int]	Maximum refinement level for 3−D elements
N47	.coarsement_-correction	[int]	Coarsement correction [0,1]
N48	.number_ind	[long]	Number of indicator data blocks
N-1 * Loop			
N49	.ref_tol	[int]	Tolerance parameter [0,1,2] 0: based on absolute values 1: based on average values 2: based on maximum values
N50	.ref_ind_0_0	[int]	Indicator for 1−D elements
N51	.curve_ind_0_-0_0	[int]	Time curve refinement 1d
N52	.curve_ind_0_-0_1	[int]	Time curve recoarsement 1d
N53	.ref_param_-0_0_0	[double]	Parameter for 1−D elements: refinement border
N54	.ref_param_-0_0_1	[double]	Parameter for 1−D elements: coarsement border
N55	.ref_param_-0_0_2	[double]	Not used
N56	.ref_ind_1_0	[int]	Indicator for 2−D elements
N57	.curve_ind_1_-0_0	[int]	Time curve refinement 2d
N58	.curve_ind_1_-0_1	[int]	Time curve recoarsement 2d

continued

Continuation from the previous page

Parameter			RF-Variable	Values	Meaning
		N59	.ref_param_-1_0_0	[double]	Parameter for 2−D elements: refinement border
		N60	.ref_param_-1_0_1	[double]	Parameter for 2−D elements: coarsement border
		N61	.ref_param_-1_0_2	[double]	Not used
		N62	.ref_ind_2_0	[int]	Indicator for 3−D elements
		N63	.curve_ind_2_-0_0	[int]	Time curve refinement 3d
		N64	.curve_ind_2_-0_1	[int]	Time curve recoarsement 3d
		N65	.ref_param_-2_0_0	[double]	Parameter for 3−D elements: refinement border
		N66	.ref_param_-2_0_1	[double]	parameter for 3−D elements: coarsement border
		N67	.ref_param_-2_0_2	[double]	Not used

20.2 Example

Input example 1: ADAPTION 2D Variante 13 (adaptive Netzverfeinerung in 3 Stufen) (tutorial_a-b-c/adaption2d_v13.rfd)

```
#ADAPTATION
 1          ; method
 3          ; max ref level
 1          ; timestep ref number
 1          ; method irr nodes
 4          ; number of neighbours to be refined (2D) (1...4)
 6          ; number of neighbours to be refined (3D) (1...6)
 0          ; node limitation method
 0          ; extended output

#ADAPTATION_COMPONENT
1            ; method ind (1=analytic estimators)
1            ; coarsement correction
1            ; number of indicator data blocks
1            ; basis of tolerance parameter (0=absolute, 1=average, 2=maximum)

0            ; Indicator parameter for 1-D diffusion elements
0. 0. 0.     ; refinement border, coarsement border, not used

0            ; Indicator parameter for 1-D advection elements
0. 0. 0.     ; refinement border, coarsement border, not used

3            ; Indicator parameter for 2-D diffusion elements
1.0 0.3 0.   ; refinement border, coarsement border, not used

3            ; Indicator parameter for 2-D advection elements
1.0 0.3 0.   ; refinement border, coarsement border, not used

0            ; Indicator parameter for 3-D diffusion elements
0. 0. 0.     ; refinement border, coarsement border, not used

0            ; Indicator parameter for 3-D advection elements
0. 0. 0.     ; refinement border, coarsement border, not used

#ADAPTATION_PRESSURE
1           ; method ind
1           ; coarsement correction
0           ; number ind (number of indicator data blocks)
```

21 Curves

21.1 Keyword Description

#CURVES			
Parameter	RF-Variable	Values	Meaning
n * Loop (solange Werte folgen)			
N1	.stuetzstellen_- 0	[double double]	Sampling point[0] Function value[0]

21.2 Example

Input example 1: BUCKLEY1D (1D Mehrphasenströmungs-Modell) (tutorial_a-b-c/buckley1d.rfd)

```
#CURVES
; curve1
;   time            value
 1.000000e+000  0.000000e+000
 1.000001e+007  1.000000e+000
 2.000000e+007  1.000000e+000
 2.000001e+007  0.000000e+000

#CURVES
; curve2
;   time            value
 0.000000e+000  0.000000e+000
 2.000000e+007  0.000000e+000
 2.000001e+007  2.000000e+000
 3.000000e+007  2.000000e+000
 3.000001e+007  0.000000e+000
```

22 Functions

22.1 Keyword Description

#FUNCTIONS			
Parameter	RF-Variable	Values	Meaning

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N1	.type	[int]	− Typ (type) 0: Typ = 0: User defined Functions or Variables (parser) Definition of user−variables (use a double as value) or user−functions (use an expression as value) Syntax of Expressions: write expression in brackets: e.g.: { COS(TIME) } Operators: + * / ^ () Math. functions: SIN() COS() TAN() LN() LOG() SQRT() EXP() PI Intern variables: TIME X Y Z RockFlow functions: FUNCTION [int] CURVE [int] User defined function in quotations: e.g.: { 'name_of_function' } Seperate inputs by blanks. Use − only in combination with a double: e.g.: { 4 + −2 + CURVE 3 * −1 } Examples: a) { (−1 * Z + CURVE 3) * 1000 } = (−[Z−coordinate] + [Curve Nr 3]) * 1000 b) { 50.0 } = 50.0 c) { (50.0 * CURVE 1 + 'TIEFE' * FUNCTION 1) * FUNCTION 2 + 100 } = (50.0 * [Curve 1] + [TIEFE]) * [Function 1]) * [Function 2] + 100.0 d) { FUNCTION 5 * TIME } = [Function 5] * [aktuelle Zeit] e) { X * −1 } = −[X−coordinate] f) { LN('hr') * 1000 * 8.3144 * 'T' / 0.018 } = (KELVIN−EQUATION) Remark: Coordinates only available for special evaluations: e.g.: BC Typ 18/101/102, SOSI Typ 106/206 1: Following linear functions provided f(x,y,z)= a + b(x−x0) + c(y−y0) + d(z−z0) 2: Following quad. functions provided f(x,y,z)= a + b(x−x0) + c(y−y0) + d(z−z0) + e(x−x0)^2 + f(y−y0)^2 + g(z−z0)^2 3: Following wave functions provided f(x,z,t)= Druck an Gewässersohle durch Hydrostatik und lineare Wellen 100: lineare Funktion (curve): f(...)= a + b * (c*[curve_val] − z − z0) (Berechnung von hydrostatischen Druck bei veraenderlichem Wasserstand

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
if N1=0			Typ = 0: User defined Functions or Variables (parser) Definition of user−variables (use a double as value) or user−functions (use an expression as value) Syntax of Expressions: write expression in brackets: e.g.: { COS(TIME) } Operators: + * / ^ () Math. functions: SIN() COS() TAN() LN() LOG() SQRT() EXP() PI Intern variables: TIME X Y Z RockFlow functions: FUNCTION [int] CURVE [int] User defined function in quotations: e.g.: { 'name_of_function' } Seperate inputs by blanks. Use − only in combination with a double: e.g.: { 4 + −2 + CURVE 3 * −1 } Examples: a) { (−1 * Z + CURVE 3) * 1000 } = (−[Z−coordinate] + [Curve Nr 3]) * 1000 b) { 50.0 } = 50.0 c) { (50.0 * CURVE 1 + 'TIEFE' * FUNCTION 1) * FUNCTION 2 + 100 } = (50.0 * [Curve 1] + [TIEFE]) * [Function 1]) * [Function 2] + 100.0 d) { FUNCTION 5 * TIME } = [Function 5] * [aktuelle Zeit] e) { X * −1 } = −[X−coordinate] f) { LN('hr') * 1000 * 8.3144 * 'T' / 0.018 } = (KELVIN−EQUATION) Remark: Coordinates only available for special evaluations: e.g.: BC Typ 18/101/102, SOSI Typ 106/206
N2	.name	[string]	Function name
N3	.dvalue	[double]	Function value
if N1=1			Following linear functions provided f(x,y,z)= a + b(x−x0) + c(y−y0) + d(z−z0)
N4	.values_0_6	[7* double]	Function values x0, y0, z0 Ursprung a fester Anteil b, c, d linearer Anteil
if N1=2			Following quad. functions provided f(x,y,z)= a + b(x−x0) + c(y−y0) + d(z−z0) + e(x−x0)^2 + f(y−y0)^2 + g(z−z0)^2
N5	.values_0_9	[10* double]	Function values x0, y0, z0 Ursprung a fester Anteil b, c, d linearer Anteil e, f, g quadr. Anteil
if N1=3			Following wave functions provided f(x,z,t)= Druck an Gewässersohle durch Hydrostatik und lineare Wellen

continued

Continuation from the previous page

Parameter		RF-Variable	Values	Meaning
	N6	.values_0_5	[6* double]	Function values x0, z0 Ursprung (Gewässersohle) H Wellenhöhe T Periode h Wassertiefe bei z0 rho Dichte des freien Wassers
if N1=100				lineare Funktion (curve): f(...)= a + b * (c*[curve_val] − z − z0) (Berechnung von hydrostatischen Druck bei veraenderlichem Wasserstand
	N7	.values_0_3	[4* double]	Function values z0 (z−Koord der Sohle) a (p0, zB Luftdruck) b (rho*g) c (Wasserstand=curve_val*c)
	N8	.curve	[long]	Curve nr

22.2 Example

Input example 1: Use of FUNCTIONS to evaluate the Kelvin equation with seasonal fluctuation (applications/d_iv_phase02.rfd)

```
#FUNCTIONS
 0                                ; sinus function within 365 d between 5 and 15 deg. Celsius
 T_tunnel    ; Temp(t) [K]        ; S(t)= 5.0 *(sin(t) + 283.15);
                                  ; S(t)=max. delta S * sin(2*pi*t/365) + Smean
 { 283.15 + -5.0*SIN((TIME+-1296000)/86400/365*360) }

#FUNCTIONS
 0                                ; sinus function within 365 d between 0.4 and 1.0
 hr_tunnel    ; rel huminidy(t)   ; S(t)= 0.3 *(sin(t) + 0.7
                                  ; S(t)=max. delta S * sin(2*pi*t/365) + Smean
 { 0.7 + -0.3*SIN((TIME+-1296000)/86400/365*360) }

#FUNCTIONS
 0
 p_kelvin     ; KELVIN equation ; p = ln(hr) * rho_l * R * T / Mv
 { LN('hr_tunnel') * 1000 * 8.3144 * 'T_tunnel' / 0.018 }
```

23 Coordinate System

The program code *RockFlow* allows the declaration of material properties with respect to a reference coordinate system. This coordinate system can be given in the input file.

The format of the input data is given in the following.

23.1 Keyword Description

The standard coordinate system is defined as default and has not to be given within this keyword. Its origin is in the point (0.0/0.0/0.0) and the axes lie in x-, y- and z-direction. It has the coordinate system number 0. If anisotropic material properties are used in a not standard 2D system (e.g. xz-System), be

sure to define this system (see also examples). It is possible to declare more than one coordinate system by using this keyword repeatedly.

#COORDINATE_SYSTEM			
Parameter	RF-Variable	Values	Meaning
N1	.method_axis	[int]	method axis (und ggfs. Daten zu d. Achsen) 0: Standard axes are used 1: Axes are given by three vectors 2: global axes (e.g. X Y or X Z)
if N1=0			Standard axes are used
if N1=1			Axes are given by three vectors
N2	.axis_vec1_0	[double]	x1
N3	.axis_vec1_1	[double]	y1
N4	.axis_vec1_2	[double]	z1
N5	.axis_vec2_0	[double]	x2
N6	.axis_vec2_1	[double]	y2
N7	.axis_vec2_2	[double]	z2
if N1=2			global axes (e.g. X Y or X Z)
N8	.axes_vec1_-name	[string]	Name axis 1
N9	.axes_vec2_-name	[string]	Name axis 2
N10	.method_-rotation	[int]	Methode Rotation (und ggfs. Daten zur Rotation) 0: No rotation 1: rotation about global coordinates−axis 2: rotation about lokal coordinates−axis, given above 3: matrix
if N10=0			No rotation
if N10=1			rotation about global coordinates−axis
N11	.count_of_-rot_axis	[int]	number of rotation axes (0,1,2) 1: axis1 alpha (e.g.: X 10.0) 2: axis1 alpha (e.g.: X 10.0) axis2 beta (e.g.: Y −3.0)
if N11=1			axis1 alpha (e.g.: X 10.0)
N12	.rot_axis1	[string]	first rotation axis (X, Y or Z)
N13	.alpha	[double]	first rotation angle (expr)
if N11=2			axis1 alpha (e.g.: X 10.0) axis2 beta (e.g.: Y −3.0)
N14	.rot_axis1	[string]	first rotation axis (X, Y or Z)
N15	.alpha	[double]	first rotation angle (expr)
N16	.rot_axis2	[string]	second rotation axis
N17	.beta	[double]	second rotation angle (expr)
if N10=2			rotation about lokal coordinates−axis, given above
N18	.count_of_-rot_axis	[int]	number of rotation axes (0,1,2)
if N11=1			axis1 alpha (e.g.: 1 10.0)
N19	.rot_axis1	[string]	first rotation axis (0, 1, 2)
N20	.alpha	[double]	first rotation angle (expr)
if N11=2			axis1 alpha (e.g.: 1 10.0) axis2 beta (e.g.: 3 −3.0)
N21	.rot_axis1	[string]	first rotation axis (0, 1, 2)
N22	.alpha	[double]	first rotation angle (expr)
N23	.rot_axis2	[string]	second rotation axis

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N24	.beta	[double]	first rotation angle (expr)
if N10=3			matrix
N25	.matrix_0	[double]	matrix [0]
N26	.matrix_1	[double]	matrix [1]
N27	.matrix_2	[double]	matrix [2]
N28	.matrix_3	[double]	matrix [3]
N29	.matrix_4	[double]	matrix [4]
N30	.matrix_5	[double]	matrix [5]
N31	.matrix_6	[double]	matrix [6]
N32	.matrix_7	[double]	matrix [7]
N33	.matrix_8	[double]	matrix [8]
N34	.method_-translation	[int]	– Methode Translation [int] (und ggfs. Daten zur Translation) 0: No translation 1: Translation
if N34=0			No translation
if N34=1			Translation
N35	.trans_vec_0	[double]	trans_vec [0]
N36	.trans_vec_1	[double]	trans_vec [1]
N37	.trans_vec_2	[double]	trans_vec [2]

23.2 Example

Input example 1: transient fluid flow and transport (anisotropie/h_anisotrop_2d_xz.rfd)

```
; Definition of a xzy-System (in 2D: xz-System)
#COORDINATE_SYSTEM
 1          ; method axis     0-> local axis = global axis;
            ;                 1-> local axes given by two 3D-vectors;
            ;                 2-> local axes given by name of two global axes e.g. X Y or X Z
   1. 0. 0.  ; local axes 1
   0. 0. 1.  ; local axes 2
 0          ; method rotation 0-> no rotation
 0          ; method translation


; alternative inputs for the definition of the same coordinate
system:

#COORDINATE_SYSTEM
 0          ; method axis
 1          ; method rotation
 2          ; number of rotation axes
 X 90               ; axis1, angle1
 Y { TIME/-100 }    ; axis2, angle2
 0          ; method translation

#COORDINATE_SYSTEM
 1          ; method axis
 1. 0. 0.   ; local axes 1
 0. 0. 1.   ; local axes 2
 2          ; method rotation
 1          ; number of rotation axes
 3 { TIME/100 }    ; axis, angle
 0          ; method translation

#COORDINATE_SYSTEM
 1          ; method axis
 1. 0. 0.   ; local axes 1
 0. 0. 1.   ; local axes 2
 1          ; method rotation
 1          ; number of rotation axes
 Y { TIME/-100 }    ; axis, angle
 0          ; method translation
```

24 Mesh Generation

24.1 Keyword Description

#MESH_GENERATION			
Parameter	RF-Variable	Values	Meaning
N1	.type	[int]	type
if N1=0			
N2	.filename	[string]	file_name

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
Einlesen aus: filename			
$READ_MG_DATA			
N3	.filename	[string]	file_name
Einlesen aus: filename			
$MESH_ID			
N4	.int_array	[int]	mesh id
$FIRST_INPUT_MESH_ID			
N5	.int_array	[int]	first input mesh id
$SECOND_INPUT_MESH_ID			
N6	.int_array	[int]	second input mesh id
$RECTANGLE			
N7	.long_array	[long]	Material group
N8	.double_array	[6* double]	anf−Kn (x, y, z) end−Kn (x, y, z)
N9	.int_array	[3* int]	nx, ny, nz Unterteilungen
N10	.int_array	[int]	Bezugsebene (x−y−> 0, x−z−> 1, y−z−> 2)
$CUBE			
N11	.long_array	[long]	Material group
N12	.double_array	[6* double]	anf−Kn (x, y, z) end−Kn (x, y, z)
N13	.int_array	[3* int]	nx, ny, nz Unterteilungen
$QUAD			
N14	.long_array	[long]	Material group
N15	.double_array	[2* double]	Knoten1 (x, y)
N16	.double_array	[2* double]	Knoten2 (x, y)
N17	.double_array	[2* double]	Knoten3 (x, y)
N18	.double_array	[2* double]	Knoten4 (x, y)
N19	.int_array	[int]	Unterteilungen zwischen Knoten 1 und 2, bzw. 3 und 4 (n)
N20	.int_array	[int]	Unterteilungen zwischen Knoten 1 und 4, bzw. 2 und 3 (m)
$EXPAND_2D_TO_3D			
N21	.long_array	[long]	Material group
N22	.double_array	[3* double]	Expansionsvektor
N23	.int_array	[int]	Unterteilungen
$BUILD_NODE_EL_REL_2D			
$BUILD_NODE_EL_REL_3D			
$HOLE_RING_PLATE_QUARTER			
N24	.int_array	[3* int]	material_group_circle, material_group_ring, material_group_plate
N25	.int_array	[int]	plane: 0−xy, 1−xz, 2−yz
N26	.int_array	[int]	mode 0:all 1:hole, 2:ring, 3:plate
N27	.double_array	[2* double]	x0,y0
N28	.double_array	[2* double]	x1,y1
N29	.double_array	[2* double]	radius_circle, radius_ring

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N30	.int_array	[int]	number of rings ring /no ring if(number_of_rings_ring=0)
N31	.int_array	[int]	number of rings rect /no plate if(number_of_rings_rect=0)
N32	.int_array	[int]	number of sectors
N33	.int_array	[int]	sort bound nodes (0/1)
$HOLE_RING_PLATE_QUARTER_EQUAL_ANGLES			
N34	.int_array	[3* int]	material_group_circle, material_group_ring, material_group_plate
N35	.int_array	[int]	plane: 0−xy, 1−xz, 2−yz
N36	.int_array	[int]	mode 0:all 1:hole, 2:ring, 3:plate
N37	.double_array	[2* double]	x0,y0
N38	.double_array	[2* double]	x1,y1
N39	.double_array	[2* double]	radius_circle, radius_ring
N40	.int_array	[int]	number of rings ring /no ring if(number_of_rings_ring=0)
N41	.int_array	[int]	number of rings rect /no plate if(number_of_rings_rect=0)
N42	.int_array	[int]	number of sectors
N43	.int_array	[int]	sort bound nodes (0/1)
$ROTATE_2D_TO_3D			
N44	.long_array	[long]	Material group
N45	.double_array	[6* double]	Drehachse (x1, y1, z1, x2, y2, z2)
N46	.double_array	[double]	Drehwinkel
N47	.long_array	[long]	Unterteilungen
$DISPLACE_MESH_2D			
N48	.double_array	[3* double]	Verschiebungsvektor (x, y, z)
$DISPLACE_MESH_3D			
N49	.double_array	[3* double]	Verschiebungsvektor (x, y, z)
$ROTATE_MESH_2D			
N50	.double_array	[3* double]	Drehachse (x, y, z)
N51	.double_array	[double]	Drehwinkel
$ROTATE_MESH_3D			
N52	.double_array	[3* double]	Drehachse (x, y, z)
N53	.double_array	[double]	Drehwinkel
$DEL_EQUAL_NODES_3D			
$DUPLICATE_MESH_2D			
$DUPLICATE_MESH_3D			
$MESH_COMBINE_2D			
$MESH_COMBINE_3D			
$MIN_NODE_DISTANCE_3D			
$MIN_EL_SIZE_3D			
$OUTPUT_RFI_2D			
N54	.char_array	[string]	file_name
$OUTPUT_RFI_3D			
N55	.char_array	[string]	file_name

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
$INPUT_RFI_2D			
N56	.char_array	[string]	file_name
N57	.int_array	[long]	Material group
if N1=1			
$READ_MG_DATA			
N58	.filename	[string]	file_name
Einlesen aus: filename			
$MESH_ID			
N59	.int_array	[int]	mesh id
$FIRST_INPUT_MESH_ID			
N60	.int_array	[int]	first input mesh id
$SECOND_INPUT_MESH_ID			
N61	.int_array	[int]	second input mesh id
$RECTANGLE			
N62	.long_array	[long]	Material group
N63	.double_array	[6* double]	anf−Kn (x, y, z) end−Kn (x, y, z)
N64	.int_array	[3* int]	nx, ny, nz Unterteilungen
N65	.int_array	[int]	Bezugsebene (x−y−> 0, x−z−> 1, y−z−> 2)
$CUBE			
N66	.long_array	[long]	Material group
N67	.double_array	[6* double]	anf−Kn (x, y, z) end−Kn (x, y, z)
N68	.int_array	[3* int]	nx, ny, nz Unterteilungen
$QUAD			
N69	.long_array	[long]	Material group
N70	.double_array	[2* double]	Knoten1 (x, y)
N71	.double_array	[2* double]	Knoten2 (x, y)
N72	.double_array	[2* double]	Knoten3 (x, y)
N73	.double_array	[2* double]	Knoten4 (x, y)
N74	.int_array	[int]	Unterteilungen zwischen Knoten 1 und 2, bzw. 3 und 4 (n)
N75	.int_array	[int]	Unterteilungen zwischen Knoten 1 und 4, bzw. 2 und 3 (m)
$EXPAND_2D_TO_3D			
N76	.long_array	[long]	Material group
N77	.double_array	[3* double]	Expansionsvektor
N78	.int_array	[int]	Unterteilungen
$BUILD_NODE_EL_REL_2D			
$BUILD_NODE_EL_REL_3D			
$HOLE_RING_PLATE_QUARTER			
N79	.int_array	[3* int]	material_group_circle, material_group_ring, material_group_plate
N80	.int_array	[int]	plane: 0−xy, 1−xz, 2−yz
N81	.int_array	[int]	mode 0:all 1:hole, 2:ring, 3:plate
N82	.double_array	[2* double]	x0,y0
N83	.double_array	[2* double]	x1,y1

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
N84	.double_array	[2* double]	radius_circle, radius_ring
N85	.int_array	[int]	number of rings ring /no ring if(number_of_rings_ring=0)
N86	.int_array	[int]	number of rings rect /no plate if(number_of_rings_rect=0)
N87	.int_array	[int]	number of sectors
N88	.int_array	[int]	sort bound nodes (0/1)
$HOLE_RING_PLATE_QUARTER_EQUAL_ANGLES			
N89	.int_array	[3* int]	material_group_circle, material_group_ring, material_group_plate
N90	.int_array	[int]	plane: 0−xy, 1−xz, 2−yz
N91	.int_array	[int]	mode 0:all 1:hole, 2:ring, 3:plate
N92	.double_array	[2* double]	x0,y0
N93	.double_array	[2* double]	x1,y1
N94	.double_array	[2* double]	radius_circle, radius_ring
N95	.int_array	[int]	number of rings ring /no ring if(number_of_rings_ring=0)
N96	.int_array	[int]	number of rings rect /no plate if(number_of_rings_rect=0)
N97	.int_array	[int]	number of sectors
N98	.int_array	[int]	sort bound nodes (0/1)
$ROTATE_2D_TO_3D			
N99	.long_array	[long]	Material group
N100	.double_array	[6* double]	Drehachse (x1, y1, z1, x2, y2, z2)
N101	.double_array	[double]	Drehwinkel
N102	.long_array	[long]	Unterteilungen
$DISPLACE_MESH_2D			
N103	.double_array	[3* double]	Verschiebungsvektor (x, y, z)
$DISPLACE_MESH_3D			
N104	.double_array	[3* double]	Verschiebungsvektor (x, y, z)
$ROTATE_MESH_2D			
N105	.double_array	[3* double]	Drehachse (x, y, z)
N106	.double_array	[double]	Drehwinkel
$ROTATE_MESH_3D			
N107	.double_array	[3* double]	Drehachse (x, y, z)
N108	.double_array	[double]	Drehwinkel
$DEL_EQUAL_NODES_3D			
$DUPLICATE_MESH_2D			
$DUPLICATE_MESH_3D			
$MESH_COMBINE_2D			
$MESH_COMBINE_3D			
$MIN_NODE_DISTANCE_3D			
$MIN_EL_SIZE_3D			
$OUTPUT_RFI_2D			
N109	.char_array	[string]	file_name

continued

Continuation from the previous page

Parameter	RF-Variable	Values	Meaning
$OUTPUT_RFI_3D			
N110	.char_array	[string]	file_name
$INPUT_RFI_2D			
N111	.char_array	[string]	file_name
N112	.int_array	[long]	Material group

24.2 Example

Input example 1: 3-D Strömungsmodell (mesh_generation/quartersphere.rfd)

```
#MESH_GENERATION
0 quartersphere.mgd

(mesh_generation/quartersphere.mgd):
 $MESH_ID 1
  $HOLE_RING_PLATE_QUARTER
    0 1 2    ; material_group_circle, material_group_ring, material_group_plate
    0        ; plane: 0-xy, 1-xz, 2-yz
    0        ; 0-all 1-hole, 2-ring, 3-plate
    0.0 0.0  ; x0,y0
    1.0 1.0  ; x1,y1
    0.4 0.5  ; radius_circle, radius_ring
    2        ; number_of_rings_ring    (no ring if  (number_of_rings_ring==0))
    0        ; number_of_rings_rect    (no plate if (number_of_rings_rect==0))
    8        ; number_of_sectors
    0        ; sort_bound_nodes (0/1)

  $FIRST_INPUT_MESH_ID 1
  $ROTATE_2D_TO_3D
    -1                                  ; Material
    0.0 0.0 0.0  0.0 1.0 0.0            ; Drehachse (x1, y1, z1, x2, y2, z2)
    3.14159265358979323846264338327 95 ; Drehwinkel
    16                                  ; Unterteilungen
  $OUTPUT_RFI_3D quartersphere.rfi
#STOP
```