GeoSys/RockFlow
Version 4.5.10(WW)

Open Source
Software Design Proposal

Zentrum für Angewandte Geowissenschaften
Lehrstuhl für GeoSystemForschung
Universität Tübingen

O Kolditz, M Beinhorn, M Xie, T Kalbacher,
S Bauer, W Wang, Ch McDermott, C Chen, C Beyer,
J Gronewold, D Kemmler, R Walsh, Y Du, C-H Park,
M Hess, C Bürger, J-O Delfs

Leipzig – Jan 2008

Contents

1 Preface

RockFlow history

text from Joelle (Development of numerical tools for modeling THM coupling processes in nuclear waste storage applications, common manuscript by ZAG,ISEB and BGR)

last modified: OK January 17, 2008

2 Software Concept

New features

- object-orientation
- C++ implementation
- libraries (GEOLib, MSHLib, FEMLib)
- file concept (object-oriented)
- user interface (Windows-MDI/OpenGL application)
- software interfaces (ArcGIS,gOcad,SURFER,EXCEL)
- PHREEQC interface for chemical reactions
- data base for material properties
- parallel computing (#define PARALLEL)

2.1 File Concept

Object	Acronym	File extension	Implementation	Resp
Processes	PCS	*.pcs	4.0.01	OK
Initial conditions	IC	*.ic	4.0.04	MX
Boundary conditions	BC	*.bc	4.0.03	PCH
Source/sink terms	ST	*.st	4.0.02	MB
Fluid properties	MFP	* mfp	4.0.08	YD
Solid properties	MSP	*.msp	4.0.10	WW
Medium properties	MMP	*.mmp	4.0.09	CMCD
Component properties	MCP	*.mcp	4.0.23	SB
Reactions	REC	*.rec	4.0.13	SB/MX
Time discretization	TIM	*.tim	4.0.07	OK
Numerical properties	NUM	*.num	4.0.XX	OK/WW
Grid adaptation	ADP	*.adp	todo	
Output parameter	OUT	*.out	4.0.06	OK
Parallel computing	DOM	*.ddc	3.9.18	DK
Geometric data	GEO	*.gli	3.9	CC
Mesh data	MSH	*.rfi	3.X	TK
Restart data	RFR	*.rfr	3.X	
Output data	RFO	*.rfo	3.X	

last modified: OK January 17, 2008

3 GEOLib Data

`filename.gli`

The GLI file is the input file for GeoLib. Geometric objects are:

Objects	Keyword	Class	Dimension
Points	#POINT	CGLPoint	0
Lines	#LINES	CGLLine	1
Polylines	#POLYLINE	CGLPolyline	1
Surfaces	#SURFACE	CGLSurface	2
Volumes	#VOLUME	CGLSurface	3

3.1 GEO types

Parameter	Meaning
POINT	Name of point
POLYLINE	Name of polyline
SURFACE	Name of surface
VOLUME	Name of volume
DOMAIN	whole domain

3.2 Points

Keyword: #POINTS

```
#POINTS
0 0.0 0.0 0.0 $MD 0.3 $NAME P0
1 1.0 0.0 0.0 $MD 0.3
2 1.0 3.0 0.0 $MD 0.3
3 0.3 0.0 0.0 $MD 0.3
4 0.0 0.0 3.0 $MD 0.3
5 1.0 0.0 3.0 $MD 0.3
6 1.0 3.0 3.0 $MD 0.3
7 0.0 3.0 3.0 $MD 0.3
```

```
point number | x   y   z   | point properties with subkeywords

0              0.0 0.0 0.0   $MD 0.3 $NAME P0
```

Properties	Meaning
$MD	Mesh density
$NAME	Name of the point

3.3 Lines

Keyword: #LINES

Lines are built from 2 points.

```
#LINES
0  0 1
1  1 2
2  2 3
...
9  7 2
```

```
line number | point1 point2 (numbers)

0             0      1
```

3.4 Polylines

Keyword: #POLYLINE

Polylines are built from lines specified in sub-keyword $LINES

```
#POLYLINE
 $NAME
  LOWER_FACE
 $EPSILON
  0.000000e+000
 $POINTS ; List of points identified by ID number
  22
  23
  24
  25
  22
 $POINT_VECTOR
  file_name.ply
 $MAT_GROUP
  mat_group_number
```

Sub-keyword	Objective
$NAME	name for identification
$TYPE	type for use
$EPSILON	ϵ environment
$POINTS	list of points building the polyline
$LINES	list of lines building the polyline
$POINT_VECTOR	the name of one PLY file
$MAT_GROUP	connected material group

3.5 Surfaces

Keyword: #SURFACE

Surfaces are built from polylines specified in sub-keyword $POLYLINES. Surfaces should be completely closed by the set of polylines.

```
#SURFACE
 $NAME
  SOURROUND
 $POLYLINES ; List of polylines identified by name
  NORTHERN_FACE
  EASTERN_FACE
```

```
  SOUTHERN_FACE
  WESTERN_FACE
 $TIN
  SOURROUND.tin
 $MAT_GROUP
  mat_group_number
```

Sub-keyword	Objective
$NAME	name for identification
$TYPE	type for use
$POLYLINES	list of polylines building the surface
$TIN	file name of a TIN belonging to the surface
$MAT_GROUP	connected material group

Where $TYPE takes integer value. Two special values of it are:

- 3: Flat surface with any normal direction
- 100: Cylindrical surface between two cross round sections A and B. Four extra data are required to determine this surface as: gli points index of center of sections A and sections B, the radius of the cylinder, and a tolerance to select the element nodes close to the cylindrical surface.

Here is an example of a cylindrical surface with radius of 0.485:

```
#SURFACE
 $NAME
  SFS_TUNNEL
 $TYPE
   100
    0 4 0.485 1.0e-4
```

3.6 Volumes

Keyword: #VOLUME

Volumes are built from surfaces specified in sub-keyword $SURFACES. Volumes should be completely closed by the set of surfaces.

```
#VOLUME
 $NAME
  CUBOID_DOMAIN
 $SURFACES ; List of surfaces identified by name
  BOTTOM
  TOP
  SOURROUND
 $MAT_GROUP
  mat_group_number
```

Sub-keyword	Objective
$NAME	name for identification
$TYPE	type for use
$SURFACES	list of surfaces building the volume
$MAT_GROUP	connected material group

last modified: OK 17.11.2004

4 MSHLib Data

4.1 RFI Data

This is an old file format, which will be no more developed and supported.

`filename.rfi`

The RFI file is finite element mesh (topologic) information. It is the input file for node and element data. The RFI file contains 3 blocks: data control, data area for nodes and data area for elements.

4.1.1 Example

```
// Data control - this line is not to put in the RFI file
#0#0#0#1#0.0#0#####################################################
 0   1733   3296
// Data block for nodes  - this line is not to put in the RFI file
 0    4075.3790000000    1237.6610000000   -10000.000000000
 1    3248.3140000000    1550.1820000000   -8799.0780000000
 2    134.19960000000    1216.1690000000   -5316.9770000000
 3   -3434.3590000000   -1243.2360000000   -2755.8600000000
 4    3348.2520000000   -21.761480000000   -10000.000000000
 5   -2412.6870000000   -10000.000000000   -10000.000000000
 6   -10000.000000000   -10000.000000000   -956.06250000000
 7   -10000.000000000   -8610.7600000000    0.00020700000000000
 8   -6014.6410000000   -1707.9160000000    0.00000000000000
 9   -2160.1140000000    970.08530000000   -2751.5490000000
 ...
 1730   -2557.9790848525    2911.3944854435   -941.30925539516
 1731   -1408.0551330575    2982.2914976413   -2263.2068487507
 1732   -3228.0124258355    2264.7447395733   -587.66012303999
// Data block for elements - this line is not to put in the RFI file
 0   0  line  321    60
 1   4  tri    58   328    59
 2   5  quad  304     1   351    17
 3   2  tet   339   294   337   219
 4   1  pris  328   311   297    54    53   307
 5   3  hex   344   314   343    28   319    29   307   316
 6   0  line_radial 321    60              // radial-symmetric element
 7   4  tri_axial    58   328    59        // axial-symmetric element
 8   5  quad_axial  304     1   351    17  // axial-symmetric element
 ...
 3295   6  tri  1729   1732   1711
```

4.1.2 Data control area

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

Variable name	Parameter meaning
art	0: has to be zero
bin	0: ASCII format
nr	0: file number
geom	1: output of geometry
startzeit	0.0: start time of the simulation
zeitschritt	0: time step number of previous simulation
rfi_filetype	3831: RF/RM version number // alternatively

The parameters of the second line are as follows.

Variable name	Parameter meaning
d	0: file type
anz_n	1733: number of nodes
anz_e	3296: number of elements

4.1.3 Node data

Geometric node data are: node number (has to start with 0) and node coordinates.

```
node number | x                  | y               | z

       1732   -3228.0124258355    2264.7447395733   -587.66012303999
```

4.1.4 Element data

Topologic element data are: element number (has to start with 0), material group, geometric element type and element nodes. Number of nodes per element depends on the geometric element type.

Element type	etyp	Name	Number of nodes
Line	1	line	2
Quadrilateral	2	quad	4
Hexahedron	3	hex	8
Triangle	4	tri	3
Tetrahedron	5	tet	4
Prism	6	pris	6

```
element number | material group | element type | element nodes

 0               0                line           321    60
 ...
 3295            6                tri            1729   1732   1711
```

4.2 MultiMSH Data

since 4.1.15

Object acronym	MSH
C++ class	CFEMesh
Source files	rf_ele_msh.h/cpp
File extension	*.msh
Object keyword	#FEM_MSH

Each process can be provided with a different mesh by means of the $PCS_TYPE subkeyword. If only one mesh is provided it is used for all processes defined.

4.2.1 Example

```
GeoSys-MSH: Mesh ------------------------------------------------
#FEM_MSH
 $PCS_TYPE
  GROUNDWATER_FLOW
 $GEO_TYPE //4.3.20
  geo_type_name geo_type
 $NODES
  808
  0 0.000000000000e+000 1.000000000000e+000 -5.000000000000e+000
  ...
 $ELEMENTS
  300
  0 1 -1 hex 0 4 103 3 202 206 305 205
 $LAYER
  5
 $PCS_TYPE
  RICHARDS_FLOW
  ...
 $PCS_TYPE
  OVERLAND_FLOW
  ...
#STOP
```

If keyword $AXISYMMETRY is used, the mesh data are ready for the simulation of an axisymmetrical problem with coordinate (r, z, 0.0) of each node. More information of axisymmetry is given in section 6.

```
$GEO_TYPE //4.3.20
```

This subkeyword describes a MSH-GEO relationship. It can be used e.g. for a regional soil model. The MSH contains numerous soil columns, which are treated as several local soil problems.

4.2.2 MSH-PCS relation

```
 $PCS_TYPE
  GROUNDWATER_FLOW
 ...
 $PCS_TYPE
  RICHARDS_FLOW
  ...
```

```
$PCS_TYPE
 OVERLAND_FLOW
 ...
```

4.2.3 MSH node data

```
$NODES
 808
 0 0.000000000000e+000 1.000000000000e+000 -5.000000000000e+000
 ...
```

4.2.4 MSH element data

```
$ELEMENTS
 300
 0 1 -1 hex 0 4 103 3 202 206 305 205
 ...
```

4.2.5 MSH AddOn data

```
$LAYER
 5
```

5 Domain Decomposition

Object acronym	DDC
C++ class	CPARDomain
Source files	par_ddc.h/cpp
File extension	*.ddc
Object keyword	#DOMAIN

5.1 #DOMAIN

```
#DOMAIN
 $ELEMENTS
 element_numbers
 $NODES_INNER
 inner_node_numbers
 $NODES_HALO
 halo_node_numbers
```

Subkeyword	Meaning
ELEMENTS	Element numbers of this domain
NODES_INNER	Numbers of inner nodes
NODES_BORDER	Numbers of domain boundary nodes

5.2 Examples

5.2.1 (confined) Groundwater flow

```
benchmarks: h_tri.ddc
#DOMAIN
 $ELEMENTS
 0
 1
 2
 3
 8
 9
 10
 11
 16
 17
 18
 19
 $NODES_INNER
 0
 1
 5
 6
 10
 11
 15
 16
 $NODES_BORDER
```

```
 2
 7
 12
 17
#STOP
```

When high order interpolation is required, additional element nodes except vertex node will be intrdoduced. Under such situdation, nodes on so called borders, nodes on interfaces between adjecent domains, will be computed in the programm. Therefore, as an alternative, all nodes of a domain can be listed after keyword $NODES_INNER as

```
benchmarks: h_tri.ddc#DOMAIN
 $ELEMENTS
 0
 1
 2
 3
 8
 9
 10
 11
 16
 17
 18
 19
 $NODES_INNER
 0
 1
 5
 6
 10
 11
 15
 16
 2
 7
 12
 17
#STOP
```

last modified: WW January 17, 2008

6 Processes

Object acronym	PCS
C++ class	CRFProcess
Source files	rf_pcs.h/cpp
File extension	*.pcs
Object keyword	#PROCESS

The control–keyword #PROCESS can be used to specify physical/bio/chemical processes.

6.1 #PROCESS

```
#PROCESS
 $PCS_TYPE
  LIQUID_FLOW       // H process (incompressible flow)
  GROUNDWATER_FLOW  // H process (incompressible flow)
  RIVER_FLOW        // H process (incompressible flow)
  RICHARDS_FLOW     // H process (incompressible flow)
  OVERLAND_FLOW     // H process (incompressible flow)
  GAS_FLOW          // H process (compressible flow)
  TWO_PHASE_FLOW    // H2 process (incompressible/compressible flow)
  MULTI_PHASE_FLOW  // H2 process (Non-isothermal two-phase flow)
  COMPONENTAL_FLOW  // H2 process (incompressible/compressible flow)
  HEAT_TRANSPORT    // T process (single/multi-phase flow)
  DEFORMATION       // M process (single/multi-phase flow)
  MASS_TRANSPORT    // C process (single/multi-phase flow)
  FLUID_MOMENTUM
 $CPL_TYPE
  PARTITIONED       // default
  MONOLITHIC
 $NUM_TYPE
  FEM               // Finite-Element-Method default
  FDM               // Finite-Difference-Method (for line elements)
  NEW               // New FEM assemblier
  EXCAVATION        // Works with DEFORMATION process up to now
 $PRIMARY_VARIABLE  // to specify the primary variable name
  HEAD
 $MEMORY_TYPE       // Works with $NUM_TYPE=NEW
  1                 //0  Do not save local matrices and vectors in RAM; 1, Do
 $ELEMENT_MATRIX_OUTPUT  // Element output
   1                //0  Do not output local matrices and vectors; 1, Do
 $BOUNDARY_CONDITION_OUTPUT  // Given: output the boundary condition and source term nod
 $DEACTIVATED_SUBDOMAIN  // Select the elements of a subdomain that not required by this
   1                     // Number of sub-domain
   ...                   // List of the indices of the selected subdomain. The indices a
                         //The indices are given in element data of mesh file.
   ...
$WRITE_READ_SOURCE_OR_NEUMANN_RHS
  // To write soure term or Neumann
  // BC after face or domain intergration
  // Use this keyword will save time if one exemple is run more frenquently.
```

```
  // e.g. First run: 1. Next: 2. Then you skip finding nodes on the specfied polylines,
  // faces or domains of source or Neumann BC definition and skip the corresponding
  // numerical integation as well.
   0     // Do nothing
   1     // Write
   2     // Read what has already written
$TIME_UNIT
  //HOUR, DAY, MONTH, YEAR
$MEDIUM_TYPE
  CONTINUUM  0.95
   ...
```

The axisymmetrical problems can be solved by give keyword *$AXISYMMETRY* in .msh file (c.f. section 4.2, 10.1, 9.1 and 11).

The gravity term for hydraulic analysis is assumed to be always from the vertical direction.

6.1.1 Remarks

A sub-keyword $RELOAD is also available to control output/input for test purpose. If value of $RELOAD is 1, i.e.,

```
.
.
.
$RELOAD
 1
```

Files named by process_name +"_"+pcs_type_name+"_primary_value.asc" will be produced after simulation being finished. They contain the results of primary viarables of the last time step. For deformation analysis, this value will make all Gauss point stresses stored in a binary file named as process_name +".sts". Such initial value will be read with $RELOAD=2 if the simulation is restarted. Note, this is a test subkeyword, we do not warrant or guarantee the stability for its usage in current version.

6.2 Node values

Parameter	Value	Meaning
PRESSUREx	phase	Pressure of fluid phase x
DISPLACEMENTx_X	phase	Displacement of solid phase x
DISPLACEMENTx_Y	phase	...
DISPLACEMENTx_Z	phase	...
TEMPERATUREx	phase	Temperature
CONCENTRATIONx	component	Mass concentration on component x

6.3 Coupling of processes

6.3.1 Partitioning

Partitioning is the default scheme. Each specified process is executed. The order of process execution (i.e. flow, mass transport, heat transport and deformation) depends on the order in the PCS input file.

6.4 Examples

6.4.1 (confined) Groundwater flow

```
benchmarks: h_*.*
#PROCESS
 $PCS_TYPE
  LIQUID_FLOW
#STOP
```

6.4.2 (unconfined) Groundwater flow

```
benchmarks: h_uc_*.* #PROCESS
 $PCS_TYPE
  GROUNDWATER_FLOW
#STOP
```

6.4.3 Gas flow

```
benchmarks: h_gas_*.* #PROCESS
 $PCS_TYPE
  GAS_FLOW
#STOP
```

6.4.4 Two phase flow (partitioned)

```
benchmarks: h2_*.*
#PROCESS
 $PCS_TYPE
  TWO_PHASE_FLOW
#PROCESS
 $PCS_TYPE
  TWO_PHASE_FLOW
#STOP
```

6.4.5 Multi-phase flow (monolithic)

```
benchmarks: h2_*.*
#PROCESS
 $PCS_TYPE
  MULTI_PHASE_FLOW
#STOP
```

This process has two primary variables, capillary pressure and gaseous pressure. In the code, capillary pressure gaseous pressure are denoted as PRESSURE1 and PRESSURE1, respectively.

6.4.6 Heat transport

```
benchmarks: ht_*.*
#PROCESS
 $PCS_TYPE
  LIQUID_FLOW
#PROCESS
 $PCS_TYPE
  HEAT_TRANSPORT
#STOP
```

6.4.7 Deformation

```
benchmarks: m_*.*
#PROCESS
 $PCS_TYPE
  DEFORMATION
#STOP
```

Note: For the monolithic approach of HM coupled problem, the process name for boundary conditions and source terms must be DEFORMATION_FLOW

For the case of excavation simulation, the deformation process data is given as

```
#PROCESS
 $PCS_TYPE
  DEFORMATION
  $NUM_TYPE
  EXCAVATION        // Works with DEFORMATION process up to now
#STOP
```

Consequently, user has to specify the elements in the domain to be excavated. This is can be done in .st file (see Section10.2.1). The current version can deal with linear elastic deformation of excavation.

6.4.8 Non-isothermal two-phase flow (partitioned)

```
benchmarks: th2_*.*
#PROCESS
 $PCS_TYPE
  COMPONENTAL_FLOW
#PROCESS
 $PCS_TYPE
  COMPONENTAL_FLOW
#PROCESS
 $PCS_TYPE
  HEAT_TRANSPORT
#STOP
```

6.4.9 Mass transport

```
benchmarks: c_*.*
```

```
#PROCESS
 $PCS_TYPE
  MASS_TRANSPORT
#STOP
```

6.4.10 River flow

```
GeoSys-PCS: Processes ----- 3.9.09_6OK4_SV1
#PROCESS
 $PCS_TYPE
  RIVER_FLOW
 $NUM_TYPE
  FDM
#STOP
```

6.4.11 Unsaturated flow

```
GeoSys-PCS: Processes
 $PCS_TYPE
  RICHARDS_FLOW
#STOP
```

6.4.12 Unsaturated flow with the dual porosity model

```
GeoSys-PCS: Processes
 $PCS_TYPE
  RICHARDS_FLOW
$MEDIUM_TYPE
  CONTINUUM  0.95
#STOP
```

Keyword $MEDIUM_TYPE indicates that the process is for a unsaturated flow process with the dual porosity model. While subkeyword CONTINUUM specifies the volumetric factor for matrix w_m, which is 0.95 for this example.

6.4.13 Monolithic schemes

Is more than one PCS type specified, than these processes are treated in a monolithic coupling scheme (see Example 6.4.14)

6.4.14 Consolidation (monolithic)

```
benchmarks: hm_*.*
#PROCESS
 $PCS_TYPE
  LIQUID_FLOW
  DEFORMATION
 $CPL_TYPE
  MONOLITHIC
#STOP
```

last modified: PCH January 17, 2008

7 Numerics

Object acronym	NUM
C++ class	Numerics
Source files	rf_num_new.h/cpp
File extension	*.num
Object keyword	#NUMERICS

This object is used to handle parameters of linear/non-linear solver, numerical integration and numerical relaxation.

7.1 #NUMERICS

```
#NUMERICS
 $PCS_TYPE
  LIQUID_FLOW
 $LINEAR_SOLVER
; method error_tolerance max_iterations theta precond storage
  2      0 1.e-008       1000           1.0   100       4
$COUPLING_ITERATIONS
 CPL_NAME1 3 1.0e-3
#NUMERICS
 $PCS_TYPE
  HEAT_TRANSPORT
 $LINEAR_SOLVER
; method error_tolerance max_iterations theta precond storage
  2      0 1.e-012       1000           0.5   100       4
CPL_NAME2 3 1.0e-3
#NUMERICS
 $PCS_TYPE
  DEFORMATION
 $NON_LINEAR_SOLVER
; method error_tolerance max_iterations relaxation
  NEWTON 1e-2 1e-10      100            0.0
 $LINEAR_SOLVER
; method error_tolerance max_iterations  theta   precond  storage
  2      0      1.e-011       5000          1.0   100       4
#STOP
```

where PCS_TYPE refers to a process. Data followed after keyword LINEAR_SOLVER define parameters to control the convergence of a linear solver as given in Table 1. Note: In current version, only SpBICGSTAB is well tested.

The coupling loop is controlled by keyword $COUPLING_ITERATIONS followed by acronym of the names of the coupled processes, maximum number of iterations, and the tolerance. For example, if a THM coupled problem is being modeled, the CPL_NAME1 can be THM and the CPL_NAME2 can be TH. If keyword $COUPLING_ITERATIONS is not given for a simulation, the default value, maximum number of iterations being 1, will be used.

Keyword **$NON_LINEAR_SOLVER** leads the configuration of the basic nonlinear solver as given in Table 2. Note: In current version, Newton-Raphson method is only valid for deformation and overland flow analysis.

Name	Number	Meaning
Method	1	SpGAUSS, direct solver
	2	SpBICGSTAB
	3	SpBICG
	4	SpQMRCGSTAB
	5	SpCG
	6	SpCGNR
	7	CGS
	8	SpRichard
	9	SpJOR
	10	SpSOR
Error	0	Absolutely error, $\|\boldsymbol{r}\| < \epsilon$
	1	$\|\boldsymbol{r}\| < \epsilon\|\boldsymbol{b}\|$
	2	$\|\boldsymbol{r}_n\| < \epsilon\|\boldsymbol{r}_{n-1}\|$
	3	if $\|\boldsymbol{r}_n\| > 1$, $\|\boldsymbol{r}_n\| < \epsilon\|\boldsymbol{r}_{n-1}\|$, else $\|\boldsymbol{r}\| < \epsilon$
	4	$\|\boldsymbol{r}_n\| < \epsilon\|\boldsymbol{x}\|$
	5	$\|\boldsymbol{r}_n\| < \epsilon\ max(\|\boldsymbol{x}\|, \|\boldsymbol{b}\|, \|\boldsymbol{r}_{n-1}\|)$
	6	other
Theta	0	Relaxation parameter, $\theta \in [0, 1]$.
Preconditioner	0	No preconditioner
	1	Jacobi
	100	ILU
Storage	2	Symmetry
Storage	4	Asymmetry

Table 1: Data by keyword **$LINEAR_SOLVER**

Name	Number	Meaning
Method	PICARD	Picard iteration
	NEWTON	Newton-Raphson methods
Error	float number	tolerance for global Newton-Raphson step
Tolerance	float number	tolerance for Picard or local Newton-Raphson step

Table 2: Data by keyword **$NON_LINEAR_SOLVER**

Keyword $COUPLING_ITERATIONS is always located at the first #NUMERICS object if staggered scheme is used for a coupled processes. last modified: WW - January 17, 2008

8 Initial Conditions

Object acronym	IC
C++ class	CInitialCondition
Source files	rf_ic_new.h/cpp
File extension	*.ic
Object keyword	#INITIAL_CONDITION

8.1 #INITIAL_CONDITION

```
#INITIAL_CONDITION
 $PCS_TYPE // physical process
  LIQUID_FLOW       // H process (incompressible flow)
  UNCONFINED_FLOW   // H process (incompressible flow)
  GAS_FLOW          // H process (compressible flow)
  TWO_PHASE_FLOW    // H2 process (incompressible/compressible flow)
  COMPONENTAL_FLOW  // H2 process (incompressible/compressible flow)
  RIVER_FLOW        // H process (incompressible flow)
  RICHARDS_FLOW     // H process (incompressible flow)
  OVERLAND_FLOW     // H process (incompressible flow)
  HEAT_TRANSPORT    // T process (single/multi-phase flow)
  DEFORMATION       // M process (single/multi-phase flow)
  MASS_TRANSPORT    // C process (single/multi-phase flow)
  GROUNDWATER_FLOW  // H process (incompressible flow)
 $PRIMARY_VARIABLE
  PRESSURE1      // flow (phase)
  SATURATION2
  TEMPERATURE1   // heat transport
  DISPLACEMENT_X // deformation
  DISPLACEMENT_Y
  DISPLACEMENT_Z
  CONCENTRATION1 // mass transport
  CONCENTRATIONx
 $GEO_TYPE // geometry
  POINT    name
  POLYLINE name
  SURFACE  name
  VOLUME   name
  DOMAIN
  SUB_DOMAIN
  MATERIAL_DOMAIN number (material group defined in *.rfi)
 $DIS_TYPE // value distribution
  CONSTANT     value
  DISTRIBUTED  file_name
  GRADIENT     value1, value2, value3
  GRADIENT_Z   value1, value2, value3
  RESTART      file_name.rfr
```

Parameter	Acronym	Meaning
PCS_TYPE	PCS	Reference to a process
GEO_TYPE	GEO	Reference to a geometric object
DIS_TYPE	DIS	Distribution of values

8.1.1 $PCS_TYPE

Parameter	Value	Meaning
PRESSUREx	phase	Source term for fluid phase x
DISPLACEMENTx_X	phase	Load force for solid phase x
DISPLACEMENTx_Y	phase	...
DISPLACEMENTx_Z	phase	...
TEMPERATUREx	phase	Source term for temperature
CONCENTRATIONx	component	Source term for component mass

8.1.2 $GEO_TYPE

Parameter	Meaning
POINT	Name of point
POLYLINE	Name of polyline
SURFACE	Name of surface
VOLUME	Name of volume
DOMAIN	whole domain
SUB_DOMAIN	patches of domain
MATERIAL_DOMAIN	Number of material group

Initialization of specified zones in a domain can be done by using SUB_DOMAIN for keyword $GEO_TYPE followed data as the number of sub-domains, pair data for indeces of material type (given in .rfi file) and initial values, i.e,

```
 #INITIAL_CONDITION
 $PCS_TYPE
  HEAT_TRANSPORT
 $PRIMARY_VARIABLE
  TEMPERATURE1
 $GEO_TYPE
  SUB_DOMAIN
   2
   0  25.0
   1  23.0
 $DIS_TYPE
  GRADIENT   500.0  10.0  0.03
#STOP
```

Where number 2 after keyword "SUB_DOMAIN" indicates there are two set of initial values. For each pair of data, the integer defines patch index, which is given in the file about finite element mesh data; the float one gives value. After reading these initial data, the programme will search the elements, which have the same patch indices given in "SUB_DOMAIN" section, and assign the corresponding values.

Initial stress by defining a linear function For stress variables in deformation analysis, we can give initial values of stresses by a linear expression such as

```
#INITIAL_CONDITION
 $PCS_TYPE
  DEFORMATION
 $PRIMARY_VARIABLE
  STRESS_XX
 $GEO_TYPE
```

```
  SUB_DOMAIN
   4
   0   -23.75+-0.2*y
   1   24.75+-0.5*y
   2   26.75+12.*y+-10*x
   3   27.75+14.0*y+-10*x
#INITIAL_CONDITION
 $PCS_TYPE
  DEFORMATION
 $PRIMARY_VARIABLE
  STRESS_YY
 $GEO_TYPE
  SUB_DOMAIN
  4
  0   23.75+0.2*y
   1   24.75+1.3*y
   2   26.75+16.*y+-20*x
   3   27.75+-18.0*y+-20*x
#STOP
```

Such an expression takes form

$$a + b * x + c * y + d * z$$

with a, b, c, d the float numbers. Note: There is no blank space in the expression. Operator $+$ is used as both delimiter and operator in the expression. If coefficients b, c or d is negative, symbol $+$ must be given before the coefficient, e.g. 27.75+-18.0*y+-20*x with $b = -20.0$ and $c = 18.0$.

8.1.3 $DIS_TYPE

Parameter	Meaning
CONSTANT	constant value
GRADIENT	ref position, ref value, gradient
GRADIENT_Z	ref position (z), ref value, gradient

8.2 Examples

8.2.1 Initial condition in the domain

```
benchmark: h_line.ic
#INITIAL_CONDITION
 $PCS_TYPE
  LIQUID_FLOW
 $PRIMARY_VARIABLE
  PRESSURE1
 $GEO_TYPE
  DOMAIN
 $DIS_TYPE
  CONSTANT 0.0
#STOP
```

This initial condition is defined for a process with primary variable PRESSURE1. Geometrically the initial condition is assigned to the whole domain, i.e. all mesh points.

Here the use of gradient applies temperature values according to depth z inside the domain specified. The first value gives the depth of a reference point, the second value its temperature in this case, the third value the change in temperature with depth.

```
 #INITIAL_CONDITION
 $PCS_TYPE
  TEMPERATURE1
 $GEO_TYPE
  DOMAIN
 $DIS_TYPE
  GRADIENT_Z   500.0  10.0  0.03
#STOP
```

8.2.2 Restart as initial conditions

Initial conditions can be used for restarts. A corresponding RFR file (restart.rfr) is required. We present an example for unsaturated Richards flow.

```
#INITIAL_CONDITION
 $PCS_TYPE
  RICHARDS_FLOW
 $PRIMARY_VARIABLE
  PRESSURE1
 $GEO_TYPE
  DOMAIN
 $DIS_TYPE
  RESTART restart.rfr
```

Example for an RFR file:

```
#0#0#0#1#3.236234700193E-04#0#4.2.18# // not used
1   1   4                             // not used
1   1                                 // number of node variables (first number)
PRESSURE1,  Pa                        // name of node variable, unit
0   0.000000000000000e+000            // node number, node value
1  -1.941899651235000e+001
2  -3.880014828104000e+001
3  -5.814051970137000e+001
4  -7.744430939151999e+001
...
231 -3.245527412767000e+002
232 -3.245527412767000e+002
```

Information of restart time comes from the TIM data (.tim file). Important is that number of node variables and variable names are compatible. In the data block node values for each mesh node are required.

last modified: YD - 5th July 2006

9 Boundary Conditions

Object acronym	BC
C++ class	CBoundaryCondition
Source files	rf_bc_new.h/cpp
File extension	*.bc
Object keyword	#BOUNDARY_CONDITION

9.1 #BOUNDARY_CONDITION

```
#BOUNDARY_CONDITION
 $PCS_TYPE // physical process
  LIQUID_FLOW       // H process (incompressible flow)
  UNCONFINED_FLOW   // H process (incompressible flow)
  GAS_FLOW          // H process (compressible flow)
  TWO_PHASE_FLOW    // H2 process (incompressible/compressible flow)
  COMPONENTAL_FLOW  // H2 process (incompressible/compressible flow)
  RIVER_FLOW        // H process (incompressible flow)
  RICHARDS_FLOW     // H process (incompressible flow)
  OVERLAND_FLOW     // H process (incompressible flow)
  HEAT_TRANSPORT    // T process (single/multi-phase flow)
  DEFORMATION       // M process (single/multi-phase flow)
  MASS_TRANSPORT    // C process (single/multi-phase flow)
  GROUNDWATER_FLOW  // H process (incompressible flow)
 $PRIMARY_VARIABLE
  PRESSURE1      // flow (phase)
  SATURATION2
  TEMPERATURE1   // heat transport
  DISPLACEMENT_X1 // deformation  (radial direction for axisymmetry)
  DISPLACEMENT_Y1
  DISPLACEMENT_Z1  //(axial direction for axisymmetry)
  CONCENTRATION1 // mass transport
  CONCENTRATIONx
 $GEO_TYPE // geometry
  POINT     name
  POLYLINE  name
  SURFACE   name
 $DIS_TYPE // value distribution
  CONSTANT  value
  LINEAR
  DATA_BASE file_name // EXCEL(CSV) file
 $TIM_TYPE // time dependencies
  CURVE number
 $FCT_TYPE // 4.2.14(OK)
  fct_name
```

Parameter	Acronym	Meaning
PCS_TYPE	PCS	Reference to a process
GEO_TYPE	GEO	Reference to a geometric object
DIS_TYPE	DIS	Distribution of source terms values
TIM_TYPE	TIM	Time dependencies of source terms
FCT_TYPE	FCT	Functional dependencies of source terms

9.1.1 $PCS_TYPE

Parameter	Value	Meaning
PRESSUREx	phase	Source term for fluid phase x
DISPLACEMENTx_X	phase	Load force for solid phase x
DISPLACEMENTx_Y	phase	...
DISPLACEMENTx_Z	phase	...
TEMPERATUREx	phase	Source term for temperature
CONCENTRATIONx	component	Source term for component mass

9.1.2 $GEO_TYPE

Parameter	Meaning
POINT	Name of point
POLYLINE	Name of polyline
SURFACE	Name of surface

9.1.3 $DIS_TYPE

Parameter	Meaning
CONSTANT	constant value
LINEAR	?WW
DATA_BASE	distributed data from file

9.1.4 $TIM_TYPE

Parameter	Meaning
CURVE	TIM curve number

9.2 Examples

9.2.1 Boundary condition at point

```
benchmark: h_line.bc
#BOUNDARY_CONDITION
 $PCS_TYPE
  LIQUID_FLOW
 $PRIMARY_VARIABLE
  PRESSURE1
 $GEO_TYPE
  POINT POINT1
 $DIS_TYPE
  CONSTANT 2.000000e+004
#STOP
```

This boundary condition is defined for a process with primary variable PRESSURE1. Geometrically the boundary condition is linked to point 1 named POINT1.

```
benchmark: h_line.gli
#POINTS
0 0.0 0.0 0.0  0.0
```

```
1 1.000000e+002  0.000000e+000  0.000000e+000
#STOP
```

9.2.2 Boundary condition along polyline

```
benchmark: hm_tri.bc
#BOUNDARY_CONDITION
 $PCS_TYPE
  DEFORMATION
 $PRIMARY_VARIABLE
  DISPLACEMENT_Y1
 $GEO_TYPE
  POLYLINE BOTTOM
 $DIS_TYPE
  CONSTANT 0.0
#STOP
```

This boundary condition is defined for a process with primary variable DISPLACEMENT_Y1. Geometrically the source term is linked to a polyline named BOTTOM.

```
benchmark: hm_tri.gli
#POINTS
2 0.0 -1.0 0.0
3 0.1 -1.0 0.0
#POLYLINE
 $NAME
  BOTTOM
 $TYPE
  2
 $EPSILON
  1.0e-4
 $POINTS
 2
 3
#STOP
```

The polyline BOTTOM is defined by two points 2 and 3.

9.2.3 Boundary condition at surface

```
benchmark: h_tet3.bc
#BOUNDARY_CONDITION
 $PCS_TYPE
  LIQUID_FLOW
 $PRIMARY_VARIABLE
  PRESSURE1
 $GEO_TYPE
  SURFACE BC_PRESSURE
 $DIS_TYPE
  CONSTANT 0.0
#STOP
```

This boundary condition is defined for a process with primary variable PRESSURE1. Geometrically the boundary condition is linked to a surfaced named BC_PRESSURE, which is defined by a polyline.

```
benchmark: h_tet3.gli
#POINTS
2 -10.  10. 0.
3 -10. -10. 0.
4 -10. -10. 10.
5 -10.  10. 10.
#POLYLINE
 $NAME
  BC_PRESSURE
 $POINTS
  2
  3
  4
  5
#SURFACE
 $NAME
  BC_PRESSURE
 $POLYLINES
  BC_PRESSURE
 $EPSILON
  0.01
```

last modified: OK - January 17, 2008

10 Source Terms

Object acronym	ST
C++ class	CSourceTerm
Source files	rf_st_new.h/cpp
File extension	*.st
Object keyword	#SOURCE_TERM

10.1 #SOURCE_TERM

```
#SOURCE_TERM
 $PCS_TYPE // physical process
  LIQUID_FLOW       // H process (incompressible flow)
  GROUNDWATER_FLOW  // H process (incompressible flow)
  RIVER_FLOW        // H process (incompressible flow)
  RICHARDS_FLOW     // H process (incompressible flow)
  OVERLAND_FLOW     // H process (incompressible flow)
  GAS_FLOW          // H process (compressible flow)
  TWO_PHASE_FLOW    // H2 process (incompressible/compressible flow)
  COMPONENTAL_FLOW  // H2 process (incompressible/compressible flow)
  HEAT_TRANSPORT    // T process (single/multi-phase flow)
  DEFORMATION       // M process (single/multi-phase flow)
  MASS_TRANSPORT    // C process (single/multi-phase flow)
 $PRIMARY_VARIABLE
  PRESSURE1      // flow (phase)
  SATURATION2
  HEAD            // head
  TEMPERATURE1   // heat transport
  DISPLACEMENT_X1 // deformation  (radial direction for axisymmetry)
  DISPLACEMENT_Y1
  DISPLACEMENT_Z1  //(axial direction for axisymmetry)
  CONCENTRATION1 // mass transport
  CONCENTRATIONx
  EXCAVATION     // Simulation of excavation deformation
 $GEO_TYPE // geometry
  POINT    name
  POLYLINE name
  SURFACE  name
  VOLUME   name
 $DIS_TYPE // value distribution
  ANALYTICAL value1 value2 value3 // 4.2.13(CMCD)
  CONSTANT value
  CONSTANT_NEUMANN value
  LINEAR no_points
   Point1 value1
   Point2 value2
   Point3 value3
  LINEAR_NEUMANN no_points
   Point1 value1
   Point2 value2
   Point3 value3
  RIVER no_points
   Point1 HeadRiver KfRiverBed WidthRiverBed TopRiverBed BottomRiverBed
```

```
  Point2 HeadRiver KfRiverBed WidthRiverBed TopRiverBed BottomRiverBed
 CRITICALDEPTH
$TIM_TYPE // time dependencies
 CURVE number
$DIS_TYPE_CONDITION //4.3.20
 PCS pcs_type_name
```

Parameter	Acronym	Meaning
PCS_TYPE	PCS	Reference to a process
GEO_TYPE	GEO	Reference to a geometric object
DIS_TYPE	DIS	Distribution of source terms values
TIM_TYPE	TIM	Time dependencies of source terms

10.1.1 $PCS_TYPE

Parameter	Value	Meaning
PRESSUREx	phase	Source term for fluid phase x
HEAD	phase	Source term for fluid phase x
DISPLACEMENTx_X	phase	Load force for solid phase x
DISPLACEMENTx_Y	phase	...
DISPLACEMENTx_Z	phase	...
TEMPERATUREx	phase	Source term for temperature
CONCENTRATIONx	component	Source term for component mass

10.1.2 $GEO_TYPE

Parameter	Meaning
POINT	Name of point
POLYLINE	Name of polyline
SURFACE	Name of surface
VOLUME	Name of volume
DOMAIN	whole domain

10.1.3 $DIS_TYPE

Parameter	Meaning
ANALYTICAL	matrix diffusion as analytical solution
CONSTANT	value is assigned to each node found
CONSTANT_NEUMANN	value times node area/node length is assigned to each node found
LINEAR	linear distribution of values is assigned to each node found
LINEAR_NEUMANN	linear distribution of values times node length or area is assigned to each node found
RIVER	linear distribution of values times node length or area is assigned to each node found, terms depend on groundwater head and the river parameters
CRITICALDEPTH	Dynamic source for Overland Flow, term depends on head
SYSTEM_DEPENDENT	free seepage face

10.1.4 $TIM_TYPE

Parameter	Meaning
CURVE	TIM curve number

10.1.5 $DIS_TYPE_CONDITION

Parameter	Meaning
PCS pcs_type_name	source term provided from coupled process pcs_type_name

10.2 Examples

10.2.1 Excavation deformation

If a process is defined as excavation deformation simulation in .pcs file (see Section 6.4.7). The domains to be excavated can be defined in this file by polylines or surfaces. Here is an example of such input data:

```
#SOURCE_TERM
 $PCS_TYPE
  DEFORMATION
 $PRIMARY_VARIABLE
  EXCAVATION
 $GEO_TYPE
  POLYLINE PLY_9
  EXCAVATION_DOMAIN 2
```

where keywords POLYLINE or SURFACE define cave surface for 2D and 3D problems, respectively. Meanwhile keyword "EXCAVATION_DOMAIN" and the following integer specifies patch index such that all elements that have this patch index are assumed in the domain to be excavated. The reason to put these data in .st file is that such data defile surfaces of caves and released loads will be applied to these surfaces as Neumann, or traction boundary conditions for excavation simulation.

10.2.2 Source at point

```
benchmark: h_line.st
#SOURCE_TERM
 $PCS_TYPE
  LIQUID_FLOW
 $PRIMARY_VARIABLE
  PRESSURE1
 $GEO_TYPE
  POINT POINT0
 $DIS_TYPE
  CONSTANT 1.157407e-006
#STOP
```

This source term is defined for a process with primary variable PRESSURE1. Geometrically the source term is linked to point 0 named POINT0.

```
benchmark: h_line.gli
#POINTS
0 0.0 0.0 0.0
#STOP
```

10.2.3 Source along polyline

```
benchmark: h_tet3.st
#SOURCE_TERM
 $PCS_TYPE
  LIQUID_FLOW
 $PRIMARY_VARIABLE
  PRESSURE1
 $GEO_TYPE
  POLYLINE BOREHOLE
 $DIS_TYPE
  CONSTANT 1.157407e-000
#STOP
```

This source term is defined for a process with the primary variable PRESSURE1. Geometrically the source term is linked to a polyline named BOREHOLE.

```
benchmark: h_tet3.gli
#POINTS
0 0.0 0.0 -10.
1 0.0 0.0  10.
#POLYLINE
 $NAME
  BOREHOLE
 $TYPE
  0
 $EPSILON
  0.060000e+000
 $POINTS
  0
  1
#STOP
```

The polyline BOREHOLE is defined by two points 0 and 1.

10.2.4 Source along polyline

This is an example for source terms (load force) along a polyline for a deformation process. The value is equally distributed along the polyline named TOP. Additionally the source has a time dependency according to time curve 2 (see section **??**).

```
benchmark: h_cc_tri.st
#SOURCE_TERM
 $PCS_TYPE
  DEFORMATION
 $PRIMARY_VARIABLE
  DISPLACEMENT_Y1
```

```
 $GEO_TYPE
  POLYLINE TOP
 $DIS_TYPE
  CONSTANT_NEUMANN -1.0
 $TIM_TYPE
  CURVE 2
#STOP
```

10.2.5 River

This is an example for a River source terms along the polyline Channel. The polyline may contain more than 2 point; data points with river parameters only are defined for 2 points 4 and 5. A linear interpolation of all parameters is done automatically.

```
benchmark: h_riv1_pris.st
#SOURCE_TERM
 $PCS_TYPE
  HEAD
 $GEO_TYPE
  POLYLINE Channel
 $DIS_TYPE
  RIVER 2
  4  3 1.00e-6 60.0 1.3 1.0
  5  3 1.00e-6 60.0 1.3 1.0
#STOP
```

For the case of Head aquiver $>$ Bottom RiverBed the equation for the river source term is:

```
q = (RiverConductance * HRiver) - (RiverConductance * HAquifer)
```

The first term is added as RHS term to the equation system, the second term is added on the diagonal using the function MXInc(). For the case of Head aquiver $<$ Bottom RiverBed the equation for the river source term is:

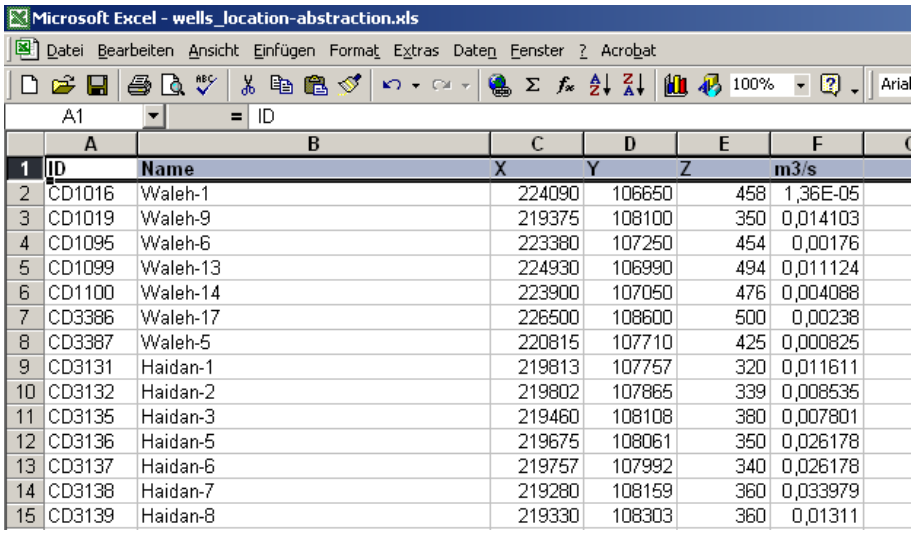
```
Haquiver  <  BRiverBed q = (RiverConductance * HRiver)   -
(RiverConductance * BRiverBed)
```

Both terms are added to the RHS as 'normal' source terms. River conductance is defined as:

```
RiverConductance = KfRiverBed* WidthRiverBed* NodeReachLength /
                   TopRiverBed - BottomRiverBed
```

10.2.6 Data input from EXCEL files

Fig. 1 shows an EXCEL file for well data. This data can be imported using the #DATA_BASE option. The EXCEL file has to be converted to a CSV file before data processing.


Microsoft Excel - wells_location-abstraction.xls

Datei Bearbeiten Ansicht Einfügen Format Extras Daten Fenster ? Acrobat

A1 = ID

	A	B	C	D	E	F
1	ID	Name	X	Y	Z	m3/s
2	CD1016	Waleh-1	224090	106650	458	1,36E-05
3	CD1019	Waleh-9	219375	108100	350	0,014103
4	CD1095	Waleh-6	223380	107250	454	0,00176
5	CD1099	Waleh-13	224930	106990	494	0,011124
6	CD1100	Waleh-14	223900	107050	476	0,004088
7	CD3386	Waleh-17	226500	108600	500	0,00238
8	CD3387	Waleh-5	220815	107710	425	0,000825
9	CD3131	Haidan-1	219813	107757	320	0,011611
10	CD3132	Haidan-2	219802	107865	339	0,008535
11	CD3135	Haidan-3	219460	108108	380	0,007801
12	CD3136	Haidan-5	219675	108061	350	0,026178
13	CD3137	Haidan-6	219757	107992	340	0,026178
14	CD3138	Haidan-7	219280	108159	360	0,033979
15	CD3139	Haidan-8	219330	108303	360	0,01311

Figure 1: EXCEL sheet for well data import

10.2.7 Analytical source term

This is an example for a fracture plane in a rock body. Here the solution is applied to two processes, heat transport and mass transport. The $PCS_TYPE defines the process for which the source term is to be applied. The $GEO_TYPE specifies DOMAIN indicating all nodes in the specified material group (value 1) act as sources.

```
benchmark: frac_an.st
#SOURCE_TERM
 $PCS_TYPE
  MASS_TRANSPORT
 $PRIMARY_VARIABLE
  CONCENTRATION1
 $GEO_TYPE
  DOMAIN
 $DIS_TYPE
  ANALYTICAL 0 1.e-6 50

#SOURCE_TERM
 $PCS_TYPE
  HEAT_TRANSPORT
 $PRIMARY_VARIABLE
  TEMPERATURE1
 $GEO_TYPE
  DOMAIN
 $DIS_TYPE
  ANALYTICAL 0 1.e-3 50
```

Three variables are required

value 1: material group of the fracture

value 2: diffusion constant in matrix

value 3: number of previous time steps to be taken into account (max 50).

The method will be documented in a forthcoming publication. It is currently only available for triangular and quadratic elements acting as 2D elements in 3D space, here representing fractures in matrix.

last modified: CMCD - 19th December 2005

11 Data Output

Object acronym	OUT
C++ class	COutput
Source files	rf_out.h/cpp
File extension	*.out
Object keyword	#OUTPUT

Three types of data output are available:

- VAR_TYPE: Output for variables (primary and secondary). If no PCS_TYPE and no MSH_TYPE are specified, all PCS are screened for the given variables (standard case).
- PCS_TYPE: Output for processes. Output only for the given PCS.
- MSH_TYPE: Output for meshes. Output only for the given MSH.

11.1 #OUTPUT

```
#VERSION       //Given:  show version in all output file names
#OUTPUT
 $PCS_TYPE // physical process
  LIQUID_FLOW       // H process (incompressible flow)
  UNCONFINED_FLOW   // H process (incompressible flow)
  GAS_FLOW          // H process (compressible flow)
  TWO_PHASE_FLOW    // H2 process (incompressible/compressible flow)
  COMPONENTAL_FLOW  // H2 process (incompressible/compressible flow)
  RIVER_FLOW        // H process (incompressible flow)
  RICHARDS_FLOW     // H process (incompressible flow)
  OVERLAND_FLOW     // H process (incompressible flow)
  HEAT_TRANSPORT    // T process (single/multi-phase flow)
  DEFORMATION       // M process (single/multi-phase flow)
  MASS_TRANSPORT    // C process (single/multi-phase flow)
  GROUNDWATER_FLOW  // H process (incompressible flow)
  FLUID_MOMENTUM
 $MSH_TYPE // mesh
  msh_name //4.2.14(OK)
 $NOD_VALUES // specified node quantities
  PRESSUREx
  SATURATIONx
  TEMPERATURE1
  DISPLACEMENT1_X
  DISPLACEMENT1_Y
  DISPLACEMENT1_Z
  CONCENTRATION1
  CONCENTRATIONx
  STRESS_XX
  STRESS_XY
  STRESS_YY
  STRESS_ZZ
  STRESS_XZ
  STRESS_YZ
  STRAIN_XX
```

```
  STRAIN_XY
  STRAIN_YY
  STRAIN_ZZ
  STRAIN_XZ
  STRAIN_YZ
  STRAIN_PLS
  VELOCITY1_X
  VELOCITY1_Y
  VELOCITY1_Z
 $ELE_VALUES // specified element quantities
  VELOCITY1_X
  VELOCITY1_Y
  VELOCITY1_Z
  MASS_FLUX1_X
  MASS_FLUX1_Y
  MASS_FLUX1_Z
 $GEO_TYPE // geometry
  POINT    name
  POLYLINE name
  SURFACE  name
  VOLUME   name
  DOMAIN
  LAYER //4.3.20
 $TIM_TYPE // output times
  time1
  ...
  timex
 $DAT_TYPE // output file format
  TECPLOT
  ROCKFLOW // 4.2.14(OK)
  VTK      // 4.3.XX (GK)
 $DIS_TYPE // 4.2.14(OK)
  AVERAGE
 $AMPLIFIER // to amplify output data
  scale
#STOP
```

In the case of axisymmetrical deformation problem, the output of stresses, strains and displacements have the following meanings:

$$\sigma_{xx} = \sigma_{rr}, \quad \sigma_{yy} = \sigma_{\theta\theta}, \quad \sigma_{zz} = \sigma_{zz}, \sigma_{xy} = \sigma_{rz}$$
$$\epsilon_{xx} = \epsilon_{rr}, \quad \epsilon_{yy} = \epsilon_{\theta\theta}, \quad \epsilon_{zz} = \epsilon_{zz}, \epsilon_{xy} = \epsilon_{rz}$$
$$u_x = u_r, \quad u_z = u_z$$

Subkeyword	Acronym	Meaning
PCS_TYPE	PCS	Specified process for output
MSH_TYPE	MSH	Specified mesh for output
NOD_VALUES	NOD	Specified node values for output
ELE_VALUES	ELE	Specified element values for output
GEO_TYPE	GEO	Related geometric objects
TIM_TYPE	TIM	Specified output times
DAT_TYPE	DAT	Output file format
DIS_TYPE	DIS	Output methods (e.g. averaging)

11.1.1 $PCS_TYPE

see section 6.

11.1.2 $MSH_TYPE

see section 4.2.

11.1.3 $NOD_VALUES

see section 6.2

11.1.4 $GEO_TYPE

see section 3.1

LAYER //4.3.20

This specification produces layer output for regional processes (e.g. regional soil model).

11.1.5 $TIM_TYPE

Parameter	Meaning
...	List of output times
STEPS	Interval of output steps

11.1.6 $DAT_TYPE

Parameter	Meaning
TECPLOT	Tecplot file format (tec file)
ROCKFLOW	RockFlow file format (rfo file)
VTK	Paraview file format (vtk file)

11.1.7 $DIS_TYPE

Parameter	Meaning
AVERAGE	nodal average

11.2 Examples

11.2.1 VAR_TYPE: Output for variables

Output files

The names of the OUT files are generated automatically:

Parameter	File name
Domain	node values: filename_dom_nod.tec
Domain	element values: filename_dom_ele.tec
Time curves	filename_time_GEOName.tec
Profiles	filename_GEOName_TIMStepNumber.tec

Domain output

Data output of PRESSURE1 at t = 4.320000e+003 sec for whole domain.

```
benchmark: h_tet3.out
#OUTPUT // domain
 $NOD_VALUES
  PRESSURE1
 $GEO_TYPE
  DOMAIN
 $DAT_TYPE
  TECPLOT
 $TIM_TYPE
  4.320000e+003
#STOP
```

Data output of PRESSURE1 each time step for whole domain.

```
#OUTPUT // domain
 $NOD_VALUES
  PRESSURE1
 $GEO_TYPE
  DOMAIN
 $DAT_TYPE
  TECPLOT
 $TIM_TYPE
  STEPS 1
#STOP
```

11.2.2 Time curve output

Data output of PRESSURE1 and TEMPERATURE1 in POINT2 for all time steps.

```
benchmark: ht_line.out
#OUTPUT // time curve
 $NOD_VALUES
  PRESSURE1
  TEMPERATURE1
 $GEO_TYPE
  POINT POINT2
 $DAT_TYPE
  TECPLOT
#STOP
```

Data output of node average at surface OUT for all time steps.

```
benchmark: h2_line.out
#OUTPUT // profile
 $NOD_VALUES
  CONCENTRATION
 $GEO_TYPE
  SURFACE OUT
```

```
 $TIM_TYPE
  STEPS 1
 $DIS_TYPE
  AVERAGE
#STOP
```

11.2.3 Profile output

Data output of PRESSURE1 and SATURATION2 along polyline OUT at times t = 5e+5, 1e+6, 5e+6, 1e+7, 5e+7, 1e+8 and 2e+8 sec.

```
benchmark: h2_line.out
#OUTPUT // profile
 $NOD_VALUES
  PRESSURE1
  SATURATION2
 $GEO_TYPE
  POLYLINE OUT
 $DAT_TYPE
  TECPLOT
 $TIM_TYPE
  5e+5
  1e+6
  5e+6
  1e+7
  5e+7
  1e+8
  2e+8
#STOP
```

11.2.4 MSH_TYPE: Output for meshes

Output files

The names of the OUT files are generated automatically:

Parameter	File name
Domain	node values: `filename_dom_MSHName_nod.tec`
Domain	element values: `filename_dom_MSHName_ele.tec`
Time curves	`filename_time_MSHName_GEOName.tec`
Profiles	`filename_MSHName_GEOName_TIMStepNumber.tec`

Output for two meshes (regional soil model).

```
benchmark: 2.out
#OUTPUT
 $MSH_TYPE
  SURFACE0
 $NOD_VALUES
  PRESSURE1
  SATURATION1
 ...
#OUTPUT
```

```
 $MSH_TYPE
  SURFACE1
 $NOD_VALUES
  PRESSURE1
  SATURATION1
 ...
#STOP
```

11.2.5 PCS_TYPE: Output for meshes

Output files

The names of the OUT files are generated automatically:

Parameter	File name
Domain	node values: `filename_dom_PCSName_nod.tec`
Domain	element values: `filename_dom_PCSName_ele.tec`
Time curves	`filename_time_PCSName_GEOName.tec`
Profiles	`filename_PCSName_GEOName_TIMStepNumber.tec`

last modified: OK - 30.12.2005

12 Time Discretization

Object acronym	TIM
C++ class	CTimeDiscretization
Source files	rf_tim.h/cpp
File extension	*.tim
Object keyword	#TIME_STEPPING

12.1 #TIME_STEPPING

```
#TIME_STEPPING
 $PCS_TYPE // physical process
  LIQUID_FLOW       // H process (incompressible flow)
  UNCONFINED_FLOW   // H process (incompressible flow)
  RICHARDS_FLOW     // H process (incompressible flow)
  GAS_FLOW          // H process (compressible flow)
  TWO_PHASE_FLOW    // H2 process (incompressible/compressible flow)
  COMPONENTAL_FLOW  // H2 process (incompressible/compressible flow)
  RIVER_FLOW        // H process (incompressible flow)
  OVERLAND_FLOW     // H process (incompressible flow)
  HEAT_TRANSPORT    // T process (single/multi-phase flow)
  DEFORMATION       // M process (single/multi-phase flow)
  MASS_TRANSPORT    // C process (single/multi-phase flow)
  GROUNDWATER_FLOW  // H process (incompressible flow)
 $TIME_STEPS
  no_steps time_step_length
  ...
 $TIME_UNIT
  SECOND
  HOUR
  DAY
  YEAR
 $TIME_END
  time_end
 $TIME_START
  time_start
 $TIME_CONTROL
  COURANT
  NEUMANN           // Only available for Richards flow
  PECLET
  SELF_ADAPTIVE
  ERROR_CONTROL_ADAPTIVE     // Only available for Richards flow
 #STOP
```

Subkeyword	Data type	Meaning
PCS_TYPE	string	Specified processes for time stepping
TIME_STEPS	int,double	Number of time steps and time step length
TIME_UNIT	string	Unit of time step, default value is second
TIME_START	double	Start time
TIME_END	double	End time
TIME_CONTROL	string	Criterion for time step control

12.1.1 $PCS_TYPE

Parameter	Meaning
FLUID_FLOW	Time stepping for flow process
HEAT_TRANSPORT	Time stepping for heat transport process
DEFORMATION	Time stepping for deformation process
MASS_TRANSPORT	Time stepping for mass transport process
RIVER_FLOW	Time stepping for river flow process
RICHARDS_FLOW	Time stepping for Richards flow process

12.2 Examples

12.2.1 Domain output

Data output of PRESSURE1 at t = 4.320000e+003 sec for whole domain.

```
benchmark: th2_line.tim
#TIME_STEPPING
 $PCS_TYPE
  LIQUID_FLOW
 $TIME_STEPS
 1000  1e+0
 100  1e+1
 100  2e+1
 400  5e+1
 $TIME_START
  0.0
 $TIME_END
  2.e+4
#STOP
```

```
benchmark: h_us_line_warrick.tim
#TIME_STEPPING
 $PCS_TYPE
  RICHARDS_FLOW
 $TIME_END
  61200.0
 $TIME_START
  0.0
 $TIME_CONTROL
  NEUMANN
#STOP
```

```
benchmark: h_us_line_forsyth.tim
#TIME_STEPPING
 $PCS_TYPE
  RICHARDS_FLOW
 $TIME_UNIT
  HOUR
 $TIME_END
  240
 $TIME_START
  0.0
 $TIME_CONTROL
   SELF-ADAPTIVE
   4    1.7
   10   0.7
   MAX_TIME_STEP
   1
   MIN_TIME_STEP
   0.00001
#STOP
```

```
benchmark: h_us_line_warrick.tim #TIME_STEPPING
 $PCS_TYPE
  RICHARDS_FLOW
 $TIME_END
  61200.0
 $TIME_START
  0.0
 $TIME_CONTROL
  ERROR_CONTROL_ADAPTIVE
#STOP
```

last modified: YD - January 17, 2008

13 Fluid Properties

Object acronym	MFP
C++ class	CFluidProperties
Source files	rf_mfp.h/cpp
File extension	*.mfp
Object keyword	#FLUID_PROPERTIES

13.1 #FLUID_PROPERTIES

```
#FLUID_PROPERTIES
 $FLUID_TYPE
  AIR
 $DENSITY
  model model_parameters
 $VISCOSITY
  model model_parameters
 $SPECIFIC_HEAT_CAPACITY
  model model_parameters
 $HEAT_CONDUCTIVITY
  model model_parameters
 $PHASE_DIFFUSION
  model model_parameters
 $NON_GRAVITY
  model model_parameters
```

- fluid type for data base operation
- model / model parameters

13.2 Fluid model data input

- organized by flow types

13.2.1 Fluid density

Case 0: User-defined function by #CURVE

$$\rho^{\gamma} = f(u) \tag{1}$$

```
$DENSITY
 0 curve_number
```

Case 1: Incompressible flow

$$\rho^{\gamma} = \rho_0^{\gamma} \tag{2}$$

```
$DENSITY
 1 rho_0
```

Case 2: Compressible flow

$$\rho^{\gamma}(p) = \rho_0^{\gamma}(1 + \beta_p^{\gamma}(p^{\gamma} - p_0^{\gamma})) \tag{3}$$

```
$DENSITY
 2  rho_0  beta_p  p_0
```

Case 3: Density-dependent flow, mass convection

$$\rho^{\gamma}(C) = \rho_0^{\gamma}(1 + \beta_C^{\gamma}(C - C_0)) \tag{4}$$

```
$DENSITY
 3  rho_0  beta_C  C_0
```

Case 4: Density-dependent flow, thermal convection

$$\rho^{\gamma}(T) = \rho_0^{\gamma}(1 + \beta_T^{\gamma}(T - T_0)) \tag{5}$$

```
$DENSITY
 4  rho_0  beta_T  T_0
```

Case 5: Density-dependent flow, thermohalin convection

$$\rho^{\gamma}(C,T) = \rho_0^{\gamma}(1 + \beta_C^{\gamma}(C - C_0) + \rho_0^{\gamma}(1 + \beta_T^{\gamma}(T - T_0)) \tag{6}$$

```
$DENSITY
 5  rho_0  beta_C  C_0  beta_T  T_0
```

Case 6: Compressible non-isothermal flow

$$\rho^{\gamma}(p,T) = \rho_0^{\gamma}(1 + \beta_p^{\gamma}(p^{\gamma} - p_0^{\gamma}) + \rho_0^{\gamma}(1 + \beta_T^{\gamma}(T - T_0)) \tag{7}$$

```
$DENSITY
 6  rho_0  beta_p  p_0  beta_T  T_0
```

Case 7: Compressible non-isothermal flow with phase changes

$$\rho^{g}(p^{g},T) = \frac{M_a}{RT}p^{g} + \frac{(M_w - M_a)}{RT}p_{w,sat}^{g}(T) \tag{8}$$

```
$DENSITY
 7
```

13.2.2 Fluid viscosity

Case 0: User-defined function by #CURVE

$$\mu^{\gamma} = f(u) \tag{9}$$

```
$VISCOSITY
 0  curve_number
```

Case 1: Incompressible flow

$$\mu^{\gamma}(p) = \mu_0^{\gamma} \tag{10}$$

```
$VISCOSITY
 1  my_0
```

Case 2: Compressible flow

$$\mu^{\gamma}(p) = \mu_0^{\gamma} \left(1 + \frac{d\mu}{dp}(p - p_0)\right) \tag{11}$$

```
$VISCOSITY
 2  my_0  dmy_dp  p_0
```

Case 3: Density-dependent flow, mass convection

$$\mu^{\gamma}(C,T) = \frac{\mu}{f1 + f2} \quad f1 = f(C), f2 = f(T) \tag{12}$$

```
$VISCOSITY
 3  my_0  dmy_dC  0.0
```

Case 4: Density-dependent flow, thermal convection

$$\mu^{\gamma}(C,T) = \frac{\mu}{f1 + f2} \quad f1 = f(C), f2 = f(T) \tag{13}$$

```
$VISCOSITY
 4  my_0  0.0  dmy_dT
```

Case 41: Non-isothermal liquid flow (Yaws et al. 1976)

$$\mu^{l}(T) = 10^{-3} \exp(-2.47110^{1} + \frac{4.20910^{3}}{T} + 4.52710^{-2}T - 3.37610^{-5}T^{2}) \tag{14}$$

```
$VISCOSITY
 41
```

Case 5: Density-dependent flow, thermohaline convection

$$\mu^{\gamma}(C,T) = \frac{\mu}{f1 + f2} \quad f1 = f(C), f2 = f(T) \tag{15}$$

```
$VISCOSITY
 5  my_0  dmy_dC  dmy_dT
```

Case 6: Compressible non-isothermal flow (Reichenberg 1971)

$$\mu^{\gamma}(p,T) = \mu_0[1 + \frac{A(\frac{p}{33.910^4})^{1.5}}{B(\frac{p}{33.910^4}) + \frac{1}{C(\frac{p}{33.910^4})^{D}}}] \tag{16}$$

```
$VISCOSITY
 6
```

Case 61: Non-isothermal gas flow (Marsily 1986)

$$\mu^g(T) = 2.285 \cdot 10^{-5} + 1.01 \cdot 10^{-3} \log T \tag{17}$$

```
$VISCOSITY
 61
```

Case 7: Compressible non-isothermal flow with phase changes (Reichenberg 1971)

$$\mu^\gamma(p,T) = \mu_0[1 + \frac{A(\frac{p}{33.910^4})^{1.5}}{B(\frac{p}{33.910^4}) + \frac{1}{C(\frac{p}{33.910^4})^D}}] \tag{18}$$

```
$VISCOSITY
 7
```

13.2.3 Specific heat capacity

Case 0: User-defined function by #CURVE

$$c^\gamma = f(T) \tag{19}$$

```
$SPECIFIC_HEAT_CAPACITY
 0 curve_number
```

Case 1: Constant

$$c^\gamma = c_0^\gamma \tag{20}$$

```
$SPECIFIC_HEAT_CAPACITY
 1 rho_0
```

Case 2: Simple enthalpy based phase change

$$c^\gamma = f(h,T) \tag{21}$$

$$h^\gamma = f(c,T) \tag{22}$$

```
$SPECIFIC_HEAT_CAPACITY
 2
```

Case 3: Phase change, enthalpy defined by #CURVE

$$c^\gamma = f(h,T) \tag{23}$$

$$h^\gamma = f(T) \tag{24}$$

```
$SPECIFIC_HEAT_CAPACITY
 3  T_latent1  T_latent1 curve number
```

Case 4: LBNL Phase change model

$$c = (1-n)\rho^s c^s + nS^l\rho^l c^l + nS^g\rho^g c^g + H_1\left(e^{\frac{p^l}{\rho^l}RT}\frac{\partial\rho_s^g}{\partial T} - \frac{\rho^g p^l}{RT^2}\right) \tag{25}$$

$$H_1 = nS^g(L_0 + c^g(T - T_{l1})) \tag{26}$$

```
$SPECIFIC_HEAT_CAPACITY
 4  T_latent1  T_latent1 Latent_heat
```

Nomenclature

C	concentration
c	specific heat capacity
h	enthalpy
M_a	molar mass of air
M_w	molar mass of water
L_0	Latent heat
p	pressure
$p^g_{w,sat}$	saturated vapor pressure
R	ideal gas constant
T	temperature
T_{l1}	temperature of beginning phase change
T_{l2}	temperature of ending phase change
β_C	solutal expansion coefficient
β_p	compressibility
β_T	thermal expansion coefficient
γ	phase
μ	viscosity
ρ	density
subscript 0	reference value

13.3 Examples

13.3.1 Single phase flow

```
benchmark: h_line.mfp
#FLUID_PROPERTIES
 $FLUID_TYPE
  LIQUID
 $PCS_TYPE
  PRESSURE1
 $DENSITY
  1 1.000000e+003
 $VISCOSITY
  1 1.000000e-003
#STOP
```

13.3.2 Two phase flow

```
benchmark: h2_line.mfp
#FLUID_PROPERTIES
 $FLUID_TYPE
  LIQUID
 $PCS_TYPE
  PRESSURE1
 $DENSITY
  1 1.000000e+003
 $VISCOSITY
  1 1.000000e-003
#FLUID_PROPERTIES
 $FLUID_TYPE
  LIQUID
 $PCS_TYPE
  SATURATION2
 $DENSITY
  1 1.000000e+003
 $VISCOSITY
  1 1.000000e-003
#STOP
```

13.3.3 Non-isothermal two phase flow

```
benchmark: th2_line.mfp
#FLUID_PROPERTIES // first fluid phase
 $FLUID_TYPE
  GAS
 $PCS_TYPE
  PRESSURE1
 $DENSITY
  2 1.26 1.e5 6.6667e-6
 $VISCOSITY
  1 1.8e-5
 $HEAT_CAPACITY
  1 1.01e+3
```

```
 $HEAT_CONDUCTIVITY
  1 0.026
 $PHASE_DIFFUSION
  1 2.13e-6
#FLUID_PROPERTIES // second fluid phase
 $FLUID_TYPE
  LIQUID
 $PCS_TYPE
  SATURATION2
 $DENSITY
  2 1000. 1.e5 4.7e-7
 $VISCOSITY
  1 0.0012
 $HEAT_CAPACITY
  1 4200.
 $HEAT_CONDUCTIVITY
  1 0.6
 $PHASE_DIFFUSION
  1 2.13e-6
#STOP
```

13.3.4 Consolidation

```
benchmark: hm_tri.mfp
#FLUID_PROPERTIES
 $FLUID_TYPE
  LIQUID
 $PCS_TYPE
  PRESSURE1
 $DENSITY
  1 0.0
 $VISCOSITY
  1 1.000000e-003
#STOP
```

13.3.5 Heat transport

```
benchmark: ht_line.mfp
#FLUID_PROPERTIES
 $FLUID_TYPE
  LIQUID
 $PCS_TYPE
  PRESSURE1
 $DENSITY
  1 1.000000e+003
 $VISCOSITY
  1 1.000000e-003
 $HEAT_CAPACITY
  1 4.280000e+003
 $HEAT_CONDUCTIVITY
  1 6.000000e-001
#STOP
```

last modified: YD - January 17, 2008

14 Solid Properties

Object acronym	MSP
C++ class	CSolidProperties
Source files	rf_msp.h/cpp
File extension	*.msp
Object keyword	#SOLID_PROPERTIES

14.1 #SOLID_PROPERTIES

```
#SOLID_PROPERTIES
// Data base
 $SOLID_TYPE
  BENTONITE
  CLAY
// Mechanical properties
 $DENSITY
  model model_parameters
 $ELASTICITY
  poisson_model model_parameters
  elasticity_model model_parameters
 $PLASTICITY
  model_name model_parameters
 $VISCOSITY
  model model_parameters
// Thermal properties
 $HEAT_CAPACITY
  model model_parameters
 $HEAT_CONDUCTIVITY
  model model_parameters
 $THERMAL_EXPANSION
  model model_parameters
...
#STOP
```

14.2 Heat capacity

- mode 0: User defined curve (not available)
- mode 1: Constant number
- mode 2: Boiling mode, medium property. Input format: [mode] [wet capacity] [dry capacity] [boiling temperature] [duration temperature] [heat latent]
- mode 3: Temperature and saturation dependent heat capacity, solid property. (DECOVALEX IV)

14.3 Heat conductivity

- mode 0: User defined curve (not available)
- mode 1: Constant number

- mode 2: Boiling mode, medium properties. Input format: [mode] [wet conductivity] [dry conductivity] [boiling temperature] [duration temperature]
- mode 3: Temperature and saturation dependent heat conductivity, solid property. (DECOVALEX IV)

14.4 Examples

14.4.1 Drucker-Prager elasto-plasticity

```
benchmark: m_dp_tri.msp
#SOLID_PROPERTIES
 $ELASTICITY
  1 3.0000e-001 // Poisson ratio
 $PLASTICITY
  DRUCKER-PRAGER
  1.0e6
  -1.0e+6
  20.0
  5.0
#STOP
```

14.4.2 Cam-Clay elasto-plasticity

```
benchmark: m_cc_tri_s.msp, m_cc_quad_s.msp,  hm_cc_tri_s.msp
 #SOLID_PROPERTIES
 $ELASTICITY
  1 3.0000e-001 // Poisson ratio
 $PLASTICITY
  CAM-CLAY
   1.0    // M
   0.045  // Virgin compression index
   0.016  // Internal frictional angle
   4.2e4  // Initial pre-consolidation pressure
   0.285  // Initial void ratio
   1.0    // OCR
   -0.9e4 // Initial stress_xx
   -2.1e4 // Initial stress_yy
   -0.9e4 // Initial stress_zz
   0.0    // Minimum (stress_xx+stress_yy+stress_zz). Only for some special cases
#STOP
```

14.4.3 Norton creep model

```
benchmark: m_crp_tri.msp
#SOLID_PROPERTIES
  $DENSITY
  1 0.0
  $ELASTICITY
    POISSION  0.3
    YOUNGS_MODULUS
      1 100.0
```

```
  $CREEP_NORTON
    10e-10 5.0
#STOP
```

14.4.4 Single-Yield-Surface elasto-plasticity (Ehlers)

```
benchmark: m_dp_tri.msp
#SOLID_PROPERTIES
 $ELASTICITY
  1 3.0000e-001 // Poisson ratio
  1 1.90139e+08 // Youngs modulus
 $PLASTICITY
  SINGLE-YIELD-SURFACE
   0.      // alpha0
   0.26    // beta0
   3.5e-7  // delta0
   1.0e-7  // epsilon0
   0.0     // kappa0
   0.0     // gamma0
   0.569   // m0
   0.      // alpha1
   0.29    // beta1
   8.81e-9 // delta1
   1.5e-8  // epsilon1
   0.      // kappa1
   0.0     // gamma1
   1.0     // m1
   0.55    // psi1
    -0.26  // psi2
   0.81e-3 // ch
   0.60e-3 // cd
   100.0   // br
   1.0     // mr
   0.0     // s_xx
   0.0     // s_yy
   0.0     // s_zz
#STOP
```

14.4.5 Discrete Fracture Deformation

```
benchmark: frac_test.msp
#SOLID_PROPERTIES
  $ELASTICITY
    POISSION 1e-001
    YOUNGS_MODULUS:
     2 10.0e6 40.0e9 0.0006
#SOLID_PROPERTIES
  $ELASTICITY
    POISSION 1e-001
    YOUNGS_MODULUS:
     1 40.0e9
#STOP
```

The two material groups represent the elastic properties of the fracture and matrix material. For the fracture, Young's modulus type 2 is defined. The value 10.0e6 is the modulus (in Pa) of the open fracture sections, the value 40.0e9 is the elastic modulus of closed fracture segments. The final parameter is the aperture (in m) below which the fracture is considered closed. The second material group defines the properties of the rock matrix, here it is defined as an elastic continuum with a constant Young's modulus of 40.0e9 Pa.

last modified: OK - January 17, 2008

15 Porous Medium Properties

Object acronym	MMP
C++ class	CMediumProperties
Source files	rf_mmp.h/cpp
File extension	*.mmp
Object keyword	#MEDIUM_PROPERTIES

15.1 #MEDIUM_PROPERTIES - TORTUOSITY Changed

```
#MEDIUM_PROPERTIES
// Data base
 $MEDIUM_TYPE
   CLAY
   SILT
   SAND
   GRAVEL
   CRYSTALLINE
// Geometric properties
 $GEOMETRY_DIMENSION
   dim // 1,2,3-D
 $GEOMETRY_AREA
   area // area for 1D element, thickness for 2D element
   FILE file.dat // input of distributed data (node or element wise)
 $GEO_TYPE
  POINT    point_name
  POLYLINE polyline_name
  SURFACE  surface_name
  VOLUME   volume_name
  LAYER    Layer_number
 $POROSITY
  model model_parameters
 $TORTUOSITY
  model model_parameters
  or
  ISOTROPIC   tortuosity
  ANISOTROPIC tortuosities
  FILE file_name
// Hydraulic properties
 $STORATIVITY
  model model_parameters
 $PERMEABILITY_TENSOR
  ISOTROPIC   permeability
  ORTHOTROPIC permeabilities
  ANISOTROPIC permeabilities
  FILE file_name
 $UNCONFINED
 $PERMEABILITY_FUNCTION_SATURATION
  model model_parameters for first fluid phase
  model model_parameters for second fluid phase
```

```
  ...
 $PERMEABILITY_FUNCTION_DEFORMATION
  model model_parameters
 $PERMEABILITY_FUNCTION_PRESSURE
  model model_parameters
 $PERMEABILITY_FUNCTION_STRESS
  model model_parameters
 $PERMEABILITY_FUNCTION_VELOCITY
  model model_parameters
 $PERMEABILITY_FRAC_APERTURE
  average_type roughness_correction
 $CAPILLARY_PRESSURE
  model model_parameters
 // Thermal properties
 $HEAT_DISPERSION
  model model_parameters
 // Mass transport properties
 $MASS_DISPERSION
  model model_parameters
 // Electric properties
 $ELECTRIC_CONDUCTIVITY
  model model_parameters
 // Multi-continua properties
 $FLUID_EXCHANGE
  model model_parameters
 $MASS_EXCHANGE
  model model_parameters
 $PERMEABILITY_DISTRIBUTION
  file_name
 $MANNING_COEFFICIENT
  value
 $FRACTURE_DATA
  number_of_fractures names_of_fractures
#STOP
```

15.2 Medium model data input

15.2.1 Porosity

Case 11: Read in a file

```
$POROSITY
 11 poro_layer1.dat
```

15.2.2 Friction coefficient for overland or channel flow

```
$MANNING_COEFFICIENT
 0.15

$CHEZY_COEFFICIENT
 10
```

15.2.3 Permeability

PERMEABILITY_TENSOR

```
$PERMEABILITY_TENSOR
 FILE perm_layer1.dat
```

15.2.4 Discrete Fracture Permeability

```
$PERMEABILITY_FRAC_APERTURE
  average_type roughness_correction
```

- average_type: Arithmetic, Geometric, or Harmonic
- roughness_correction: corr_roughness OR no_corr_roughness

The average type describes how the average aperture (which is subsequently used to calculate the permeability) will be calculated. The permeability calculation is based on the cubic law. If $corr\_roughness$ is entered, this cubic law permeability will be corrected depending on the roughness of the fracture walls and the closure ratio of the fracture. This correction is based on:

Zimmerman RW, Bodvarsson GS (1996) Hydraulic Conductivity of Rock Fractures. *Transport in Porous Media* 23: 1-30

Note: For this function to work polylines must be defined for the upper and lower fracture surface profiles. See *frac_test.gli*.

15.2.5 Confined or unconfined flow

Standard is the confined flow is modelled. No additional keyword is required. If the mmp group is unconfined the keyword

```
$UNCONFINED
```

has to be used.

15.2.6 Relative permeability - saturation

Case 0: User-defined function by #CURVE

$$k_{\mathsf{rel}}^{\gamma} = f(u) \tag{27}$$

```
$PERMEABILITY_SATURATION
 0 curve_number
```

Case 2: Linear function

Case 21: Linear function from saturation

$$k_{\mathsf{rel}}^{\gamma} = 1 - S^{\gamma} \tag{28}$$

```
$PERMEABILITY_SATURATION
 21
```

Case 4: van Genuchten (1980)

$$S_{\text{eff}} = \frac{S^{\gamma} - S_r^{\gamma}}{S_{\text{max}}^{\gamma} - S_r^{\gamma}} \tag{29}$$

$$k_{\text{rel}}^{l} = S_{\text{eff}}^{1/2} \left[1 - (1 - S_{\text{eff}}^{1/m})^{m}\right]^2 \tag{30}$$

```
$PERMEABILITY_SATURATION
 4  s_res  s_max  m
```

Case 14: van Genuchten 2 (1980)

$$S_{\text{eff}} = \frac{S^{\gamma} - S_r^{\gamma}}{S_{\text{max}}^{\gamma} - S_r^{\gamma}} \tag{31}$$

$$k_{\text{rel}}^{l} = S_{\text{eff}}^{0.5} \left[1 - (1 - S_{\text{eff}}^{1/m})^{m}\right]^2 \tag{32}$$

```
$PERMEABILITY_SATURATION
 14  s_res  s_max  m
```

Case 14: van Genuchten (1980) for non-wettable phase

$$S_{\text{eff}} = \frac{S^{\gamma} - S_r^{\gamma}}{S_{\text{max}}^{\gamma} - S_r^{\gamma}} \tag{33}$$

$$k_{\text{rel}}^{g} = (1 - S_{\text{eff}})^{0.5} \left[1 - (1 - S_{\text{eff}})\right]^{2m} \tag{34}$$

```
$PERMEABILITY_SATURATION
 15  s_res  s_max  m
```

15.2.7 Capillary pressure

$$p_c = \frac{\rho^l g}{\alpha} (S_{\text{eff}}^{1/m} - 1)^{(1-m)} \tag{35}$$

```
$CAPILLARY_PRESSURE
4  alpha
```

15.2.8 Discrete Fracture Data

```
$FRACTURE_DATA
 number_of_fractures names_of_fractures
```

- number_of_fractures: The number of discrete fractures in the model domain. Each fracture is handled as a separate entity.
- names_of_fractures: The names of each of these fractures. For each fracture, polylines must be created representing the upper and lower fracture surface profiles. These polylines must have the names *name_of_fracture_top* and *name_of_fracture_bot*. See *frac_test.gli*.

15.3 Examples

15.3.1 Single phase flow

```
benchmark: h_line.mmp
#MEDIUM_PROPERTIES
 $GEOMETRY_DIMENSION
  1
 $GEOMETRY_AREA
  1.000000e+000
 $POROSITY
  1 2.000000e-001
 $TORTUOSITY
  1  1.000000e+000
 $PERMEABILITY_TENSOR
  ISOTROPIC 1.000000e-07
#STOP
```

15.3.2 Unconfined groundwater flow

```
benchmark: beerze-reuzel.mmp
#MEDIUM_PROPERTIES
 $NAME
  Layer1
 $GEO_TYPE
  LAYER 1
 $GEOMETRY_DIMENSION
  2
 $GEOMETRY_AREA
  1.000000000000e+000
 $POROSITY
   11 sf1_1x.dat1
 $PERMEABILITY_TENSOR
   FILE   kd1_simgroq.dat1
 $UNCONFINED
```

15.3.3 Two phase flow

```
benchmark: h2_line.mmp
```

```
#MEDIUM_PROPERTIES
 $GEOMETRY_DIMENSION
  1
 $GEOMETRY_AREA
  1.000000e+000
 $POROSITY
  1 2.000000e-001
 $TORTUOSITY
  1 1.000000e+000
 $PERMEABILITY_TENSOR
  ISOTROPIC 1.000000e-07
 $PERMEABILITY_SATURATION
  3 0.2 0.8 2.
  3 0.2 0.8 2.
 $CAPILLARY_PRESSURE
  0:
#STOP
```

15.3.4 Non-isothermal two phase flow

```
benchmark: th2_line.mmp
#MEDIUM_PROPERTIES
 $GEOMETRY_DIMENSION
  1
 $GEOMETRY_AREA
  1.000000e-2
 $POROSITY
  1 0.407407407
 $TORTUOSITY
  1 0.8
 $PERMEABILITY_TENSOR
  ISOTROPIC: 8.22854E-20
 $PERMEABILITY_SATURATION
  21 0.0 0.9
  4  0.1 1.0 1.0
 $PERMEABILITY_DEFORMATION
  1 1.0 1.0 7.0 3293673.0 -0.165 3.0
 $CAPILLARY_PRESSURE
  0 5
#STOP
```

15.3.5 Richards flow

```
benchmark: h_us_line.mmp #MEDIUM_PROPERTIES
 $GEOMETRY_DIMENSION
  3
 $GEOMETRY_AREA
  1.000000e+000
 $POROSITY
  1 2.000000e-001
 $TORTUOSITY
  1 1.000000e+000
 $PERMEABILITY_TENSOR
```

```
  ISOTROPIC 1.000000e-07
 $PERMEABILITY_SATURATION
  4  0  0.6452  0.791667
 $CAPILLARY_PRESSURE
  4  320
#STOP
```

15.3.6 Consolidation

```
benchmark: hm_tri.mmp
#MEDIUM_PROPERTIES
 $GEOMETRY_DIMENSION
  2
 $GEOMETRY_AREA
  1.0
 $POROSITY
  1 0.000000e-001
 $TORTUOSITY
  1 1.000000e+000
 $PERMEABILITY_TENSOR
  ISOTROPIC 1.000000e-10
#STOP
```

15.3.7 Heat transport

```
benchmark: ht_line.mmp
#MEDIUM_PROPERTIES
 $GEOMETRY_DIMENSION
  1
 $GEOMETRY_AREA
  1.0
 $POROSITY
  1 2.000000e-001
 $PERMEABILITY_TENSOR
  ISOTROPIC 1.000000e-11
 $HEAT_DISPERSION
  1 5.000000e+000 0.000000e+000
#STOP
```

15.3.8 Discrete Fracture Deformation

```
benchmark: frac_test.mmp
#MEDIUM_PROPERTIES
 $GEOMETRY_DIMENSION
  2
 $GEOMETRY_AREA
  1.000000e+000
 $POROSITY
  1 1.000000e-001
 $TORTUOSITY
  1 1.000000e+000
```

```
 $PERMEABILITY_TENSOR
  ORTHOTROPIC 1 1
 $PERMEABILITY_FRAC_APERTURE
  Arithmetic corr_roughness
 $FRACTURE_DATA
  1 Frac0
#MEDIUM_PROPERTIES
 $GEOMETRY_DIMENSION
  2
 $GEOMETRY_AREA
  1.000000e+000
 $POROSITY
  1 1.000000e-005
 $TORTUOSITY
  1 1.000000e+000
 $PERMEABILITY_TENSOR
  ISOTROPIC 1.000000e-16
#STOP
```

The two material groups represent the medium properties of the fracture and matrix material respectively.

15.3.9 Distributed data

```
benchmark: fracnet02.mmp
#MEDIUM_PROPERTIES
 $PERMEABILITY_DISTRIBUTION
  permeabilities.txt
```

last modified: OK - January 17, 2008

16 Component Properties

Object acronym	CP
C++ class	CComponentProperties
Source files	rfmat_cp.h/cpp
File extension	*.mcp
Object keyword	#COMPONENT_PROPERTIES

16.1 #COMPONENT_PROPERTIES

For each component, that has to be included in the model simulation, the corresponding component properties have to be specified. This includes chemical species as well as biological species. If the component appears in different phases (i.e. water and solid phase), it has to be specified for each phase. For each component, one process of type MASS_TRANSPORT has to be given.

```
#COMPONENT_PROPERTIES
 $NAME
  name
 $MOBIL
  0 / 1
 $TRANSPORT_PHASE
  phase_number
 $DIFFUSION
  model model_parameters
 $DECAY_AQUEOUS
  model model_parameters
 $ISOTHERM
  model model_parameters
..
```

16.2 Data Input

16.2.1 Component Name

The sub-keyword

```
$NAME
```

specifies a name for the component. The name is case-sensitive. The same name can be used to get component output. If this keyword is skipped, then the name CONCENTRATIONx will be used, where x is number of the component (starting with 0).

16.2.2 Component Mobility

The sub-keyword

```
$MOBILE
```

specifies, if a component is mobile (=1) or immobile (=0). If it is mobile, then the corresponding equation system is solved. If it is immobile, i.e. a sorbed species, then the component concentrations are just passed to the next timestep. Has to be specified, default is 1.

16.2.3 Component Phase

The sub-keyword

```
$TRANSPORT_PHASE
```

specifies the number of the phase in which the component is to be transported. 0 is water phase. Default is 0.

16.2.4 Component Diffusion

The sub-keyword

```
$DIFFUSION
```

specifies the diffusion model and the corresponding diffusion model values used for the diffusion coefficient.

Case 0: User-defined function by #CURVE Not implemented.

Case 1: Constant diffusion coefficient

$$D = D_0 \tag{36}$$

```
$DIFFUSION
1  D0
```

16.2.5 Component Decay

The sub-keyword

```
$DECAY
```

specifies, if the component decays in the phase in which it is transported. The decay does not account for the production of daughter products. Decay with a kinetics of any order as well as a Monod kinetics is accounted for. Default is no decay.

Case 0: User-defined function by #CURVE

$$\frac{\partial C}{\partial t} = -f(C) \tag{37}$$

```
$DECAY
 0 curve_number
```

Case 1: Decay with any-order kinetics

$$\frac{\partial C}{\partial t} = -KC^o \tag{38}$$

```
$DECAY
 1  K    o
```

Case 2: Decay with Monod-kinetics

$$\frac{\partial C}{\partial t} = -\frac{KC}{M + C} \tag{39}$$

```
$DECAY
 2  K M
```

16.2.6 Component Sorption

The sub-keyword

```
$ISOTHERM
```

specifies, if the component sorbs in the phase in which it is transported. Linear, Freundlich and Langmuir sorption are accounted for. No mass is transfered to the immobile phase. Default is no decay. Aqueous and sorbed concentration are termed C and C_S, respectively.

Case 0: User-defined function by #CURVE

$$C_S = f(C)C \tag{40}$$

```
$ISOTHERM
0 curve_number
```

Case 1: Linear Isotherm

$$C_S = K_D C \tag{41}$$

```
$ISOTHERM
1  KD
```

Case 2: Freundlich Isotherm

$$C_S = K_D C^e \tag{42}$$

```
$ISOTHERM
2  KD e
```

Case 3: Langmuir Isotherm

$$C_S = \frac{KC}{1 + LC} \tag{43}$$

```
$ISOTHERM
2  K L
```

16.3 Examples

16.3.1 Mass transport

```
benchmark: 1d_line.mcp

#COMPONENT_PROPERTIES  ; comp0
 $NAME
  Hallo1 ;  Component Name
 $MOBIL
  1;  Component is mobile
 $DIFFUSION
  1  1.0e-9 ; constant diffusion coefficient
 $DECAY_AQUEOUS
  1  1.0e-6  1.0  ; first - order decay
 $ISOTHERM
  1  1e-3 ;  linear sorption
#STOP
```

last modified: SB - January 17, 2008

CHM - ChemApp data

17 ChemApp data

Object acronym	eq
C++ class	CEqlink
Source files	EQL/eqlink.h/cpp
File extension	*.chm

A file with extension .chm should be provided for chemical reaction using ChemApp. This data is a standard file for ChemApp which can be found in the ChemApp user's manual.

An example of data is:

```
benchmark: eq.chm

System MgCl2-CaCO3-H2O, no reduced phases or phase constituents
   7   2   1  15   6
 EA                        H                         O
 Mg                        Ca                        Cl
 C
 0.000549                  1.008                     15.999
 24.305                    40.078                    35.453
 12.011
1    1 1    1
GAS IDMX CO2(g)
 1  1    0.0   0.0   2.0   0.0   0.0   0.0   1.0
 298.15       -3.94371E+05
AQUEOUS IDDZ H2O
 1  1   0.00000000e+00 0.0   2.0   1.0   0.0   0.0   0.0   0.0
 298.15       -2.3714E+05
H<+>
 1  1   1.00000000e+00 -1.0   1.0   0.0   0.0   0.0   0.0   0.0
 298.15       0.0
OH<->
 1  1   -1.00000000e+00 1.0   1.0   1.0   0.0   0.0   0.0   0.0
 298.15       -1.5723E+05
Mg<2+>
 1  1   2.00000000e+00 -2.0   0.0   0.0   1.0   0.0   0.0   0.0
 298.15       -4.55375E+05
Ca<2+>
 1  1   2.00000000e+00 -2.0   0.0   0.0   0.0   1.0   0.0   0.0
 298.15       -5.52806E+05
Cl<->
 1  1   -1.00000000e+00 1.0   0.0   0.0   0.0   0.0   1.0   0.0
 298.15       -1.31217E+05
HCO3<->
 1  1   -1.00000000e+00 1.0   1.0   3.0   0.0   0.0   0.0   1.0
 298.15       -5.86875E+05
CO3<2->
 1  1   -2.00000000e+00 2.0   0.0   3.0   0.0   0.0   0.0   1.0
 298.15       -5.27917E+05
CO2
 1  1   0.00000000e+00 0.0   0.0   2.0   0.0   0.0   0.0   1.0
 298.15       -3.85992E+05
CaCO3
```

```
 1  1   0.00000000e+00 0.0   0.0   3.0   0.0   1.0   0.0   1.0
 298.15       -1.099127E+06
CaHCO3<+>
 1  1   1.00000000e+00 -1.0   1.0   3.0   0.0   1.0   0.0   1.0
 298.15       -1.145992E+06
CaOH<+>
 1  1   1.00000000e+00 -1.0   1.0   1.0   0.0   1.0   0.0   0.0
 298.15       -7.16997E+05
MgCO3
 1  1   0.00000000e+00 0.0   0.0   3.0   1.0   0.0   0.0   1.0
 298.15       -1.0003E+06
MgHCO3<+>
 1  1   1.00000000e+00 -1.0   1.0   3.0   1.0   0.0   0.0   1.0
 298.15       -1.048347E+06
MgOH<+>
 1  1   1.00000000e+00 -1.0   1.0   1.0   1.0   0.0   0.0   0.0
 298.15       -6.27215E+05
Ca(OH)2_Portlandite
 1  1    0.0   2.0   2.0   0.0   1.0   0.0   0.0
 298.15       -8.96943E+05
CaCO3_Aragonite
 1  1    0.0   0.0   3.0   0.0   1.0   0.0   1.0
 298.15       -1.128306E+06
CaCO3_Calcite
 1  1    0.0   0.0   3.0   0.0   1.0   0.0   1.0
 298.15       -1.129127E+06
CaMg(CO3)2_Dolomite(dis)
 1  1    0.0   0.0   6.0   1.0   1.0   0.0   2.0
 298.15       -2.158425E+06
CaMg(CO3)2_Dolomite(ord)
 1  1    0.0   0.0   6.0   1.0   1.0   0.0   2.0
 298.15       -2.161565E+06
Mg(OH)2_Brucite
 1  1    0.0   2.0   2.0   1.0   0.0   0.0   0.0
 298.15       -8.33532E+05
```

last modified: MX - January 17, 2008

18 Functions

Object acronym	FCT
C++ class	CFunction
Source files	rf_fct.h/cpp
File extension	*.fct
Object keyword	#FUNCTION

18.1 #FUNCTION

```
#FUNCTION
 $TYPE
  fct_name
 $GEO_TYPE
  geo_type geo_name
 $VARIABLES
  fct_x_name fct_y_name
 $DATA
  x_1 y_1
  x_2 y_2
  ...
  x_n y_n
 #STOP
```

18.2 Examples

```
#FUNCTION
 $TYPE
  STEP_FUNCTION
 $GEO_TYPE
  POLYLINE BC02
 $VARIABLES
  TIME CONCENTRATION
 $DATA
  0.000000000000e+000 1.000000000000e+000
  5.000000000000e+005 1.000000000000e+000
  5.010000000000e+005 0.000000000000e+000
  1.000000000000e+006 0.000000000000e+000
 #STOP
```

last modified: OK - January 17, 2008

19 Unit

19.1 Richards Flow

19.1.1 Independent units

Objects	Subkeyword	Parameter	Unit
IC	$PRIMARY_VARIABLE	PRESSUREx	Pa
BC	$PRIMARY_VARIABLE	PRESSUREx	Pa
MFP	$DENSITY	ρ	$kg \cdot m^{-3}$
	$VISCOSITY	μ	$kg \cdot m^{-1} \cdot s^{-1}$
MMP	$POROSITY	n	-
	$PERMEABILITY_TENSOR		m^{-2}
		S	-
		k_rel	-
	van Genuchten 1980	S_res	-
		S_max	-
		m	-
		α	m^{-1}

19.1.2 Time dependent units

Keyword setting in *.tim :

```
$TIME_UNIT    // [T]
 SECOND       // Default
 HOUR
 DAY
 YEAR
```

Objects	Subkeyword	Parameter	Unit
TIM	$TIME_END		$[T]$
	$TIME_START		$[T]$
ST	$PRIMARY_VARIABLE	PRESSUREx	$m \cdot [T]^{-1}$
OUT	$TIM_TYPE		$[T]$

last modified: YD - January 17, 2008