

Multiphase Reactive Transport Module TRCHEM in IPARS

Malgorzata Peszynska and Shuyu Sun

October 29, 2001

Contents

1	Introduction	3
2	Physical Model for Multiphase Reactive Transport	3
2.1	Notation	3
2.2	Assumptions and capabilities	5
2.2.1	Assumptions and their justification	5
2.2.2	Internal unit system used	6
2.2.3	Expected model capabilities	6
2.3	Multiphase Flow Model	7
2.3.1	Special case: two phase flow model	7
2.3.2	Special case: single phase flow model	7
2.3.3	Discretization in space	8
2.3.4	Choice of flow model in input	8
2.3.5	Special case: time independent flow part	8
2.4	Derivation of Phase-Summed Reactive Transport Equations	8
2.4.1	Definitions	8
2.4.2	Phase-Summed Equations	9
2.4.3	Special Case – Single Flowing Phase	11
2.4.4	Special Case – Two Flowing Phases	11
2.5	Chemistry terms	11
2.5.1	Kinetic Reaction – Classical Rate Law	12
2.5.2	Kinetic Reaction – Monod Style Rate Law	12
2.5.3	Equilibrium type (for fast reaction)	12
2.5.4	Radionuclide Decay	13
2.5.5	Adsorption	13
3	Time splitting algorithm	13
3.1	Initial condition	13
3.2	Time stepping	13
3.3	Advection	14
3.4	Chemical Reaction	14
3.4.1	Equilibrium controlled reaction	14
3.4.2	Kinetics type reaction	14
3.4.3	Mixed kinetic and equilibrium reaction system	15

3.5	Diffusion/Dispersion	15
3.6	Radionuclide Decay	15
3.7	Adsorption	15
4	General boundary conditions	15
4.1	Flow boundary conditions	16
4.2	Boundary conditions in advection/diffusion	16
4.3	Input of boundary conditions	17
4.4	Splitting of boundary conditions	18
4.4.1	Boundary conditions in advection part	18
4.4.2	Boundary conditions in diffusion/dispersion part	18
4.4.3	Summary of boundary conditions	18
4.5	Output related to boundary conditions	19
5	Wells	19
5.1	Flow wells	19
5.2	TRCHEM wells	20
5.3	Input for wells	21
5.3.1	Example	21
5.4	Output for wells	22
6	Getting the Source Code and Executable	23
6.1	Source code	23
6.2	How to build a TRCHEM executable:	23
6.2.1	Makefile	23
6.2.2	Size file	25
6.3	Running the code	27
7	Input	27
7.1	General Remarks	27
7.2	TRCHEM Control Parameters	28
7.3	Parameters for Advection/Diffusion/Dispersion	29
7.4	Parameters for Chemistry	31
8	Implementation	33
8.1	Memory management and variables	33
8.2	Flow model postprocessing	33
8.3	Implementation of Godunov method	34
8.4	Developer's notes: TRCHEM wells	34
9	Appendix	35
9.1	The molecular diffusion tensor	35
9.2	The mechanical dispersion tensor	35
9.2.1	Derivation for single flowing phase	35
9.2.2	Derivation for multiple flowing phases	36
9.2.3	The longitudinal dispersivity d_{long} and the transverse dispersivity d_{trans}	37
9.3	Alternative for overall algorithm	37
9.3.1	Advection	38
9.3.2	Radionuclide decay	38

9.3.3	Chemical Reaction	38
9.3.4	Diffusion/Dispersion	38
9.4	Phase-summed transport equations based on total concentration	39

1 Introduction

The goal of the TRCHEM project is to develop an accurate, efficient and relatively generic multiphase reactive transport model for phenomena which occur in porous media in a number of scientific and engineering applications including chemical, petroleum and environmental engineering.

The overall multiphase multicomponent formulation of the model follows Parker's formulation [9] which we extended to arbitrary number of compressible flowing phases. The model is implemented as a part of IPARS framework (Integrated Parallel Accurate Reservoir Simulator) [1, 10, 15] and it can be used in principle with any *flow model* in IPARS provided the necessary interface routines have been written. Currently IPARS multiblock capability is not yet available under TRCHEM.

The contents of this manual are as follows. See Section 2 for the formulation of the multiphase reactive transport model and in particular Section 2.3 for currently available flow models. Section 3 describes the numerical algorithm(s) used. Sections 4 and 5 define source terms (boundary conditions and wells) which can be used. See Section 6 for direction on how to get source code and executable. Section 7 summarizes the TRCHEM-specific input. Section 8 is written for developers and describes implementation.

We note that the formulation used in TRCHEM is essentially a superset of the (single phase) formulation in another simulator developed at CSM called ParSSim [4]. Also, parts of the chemistry module in TRCHEM have been ported from ParSSim.

An example for using IPARS-TRCHEM module is presented in a report [13].

2 Physical Model for Multiphase Reactive Transport

2.1 Notation

1. Subscript α refers to one of the chemical species,
2. Subscript p refers to one of the phases (w , o , a refers to water, organic, and air phases respectively), s to stationary phase(s)
3. Subscript e or ijk refers to grid element e or cell ijk
4. t = Time, [day]
5. x_i = The i 'th spatial coordinate, $i = 1, 2, 3$, also x, y, z , [ft]
6. Φ = Porosity, [fraction]
7. K = Permeability, [md]

Flowing Phases

8. S_p = Saturation of phase p = Fraction of the pore volume occupied by phase p , [fraction]
9. ρ_p = Density of phase p [lb/ft³]
10. μ_p = Viscosity of phase p [cp]

11. P_p = Pressure of phase p [psi]
12. k_p = Relative permeability of phase p , [$fraction$]
13. \mathbf{u}_p = Darcy velocity of phase p of components (u_{p1}, u_{p2}, u_{p3}) or (u_p^x, u_p^y, u_p^z) [ft/day]
- Species**
14. $c_{\alpha p}$ = Concentration of component α in phase p , [$lbmol/ft^3$]
15. $J_{\alpha pi}$ = The total mass flux density of component α in phase p in the direction i , [$lbmol/ft^2day$]
16. $R_{\alpha p}^I$ = The net mass transfer rate per porous medium volume of component α into (positive) or out of (negative) the p phase, [$lbmol/ft^3day$]
17. $R_{\alpha p}^C$ = The net production rate of component α in phase p per unit phase volume from chemical reactions, [$lbmol/ft^3day$]
18. $D_{\alpha pij}$ = The aggregate diffusion/dispersion tensor, [ft^2/day]
19. $d_{mol,\alpha p} = \tau_p D_{\alpha p}^0$ = The molecular diffusivity for species α in phase p , [ft^2/day]
20. $d_{long,p}$ = The longitudinal dispersivity in phase p , [ft]
21. $d_{trans,p}$ = The transverse dispersivity in phase p , [ft]
22. $\Gamma_{\alpha p} = \frac{c_{\alpha p}}{c_{\alpha r}}$ = The equilibrium partitioning coefficient for species α between phase p and reference phase r , [$dimensionless$]

Wells

23. $Q_{p,e}$ = Volumetric rate of phase p into grid cell e [ft^3/day],
24. q_p = Mass rate (pointwise): source (positive) or sink (negative) term for phase p per unit volume. It is defined as $q_p(x) = 0$ if $x \notin w$ and $q_p(x) = \rho_p(x)Q_{p,e}(x)/|e|$ if $x \in e$ for some $e \in w$. [lb/ft^3day],
25. $q_{p,e}$ = Mass rate per well element /grid cell = $\int_e q_p(x)dx = \rho_p|_e Q_{p,e}$ [lb/day],
26. q_p^w = Total mass rate per well w for phase p = $\sum_{e \in w} q_{p,e}$ [lb/day],
27. $q_{\alpha p,e}$ = Mass source/sink for component α in phase p per well element/grid cell, see equations (28),(29),(30) for definition of $q_{\alpha p,e}$ [$lbmol/day$],
28. $q_{\alpha p}$ = Pointwise source (positive) or sink (negative) term for component α in phase p per unit volume. It is defined as $q_{\alpha p}(x) = 0$ if $x \notin w$ and $q_{\alpha p}(x) = q_{\alpha p,e}(x)/|e|$ if $x \in e$ for some $e \in w$. [$lbmol/ft^3day$],
29. $q_{\alpha p}^w$ = Total mass source/sink for component α in phase p per well w , [$lbmol/day$],
30. $\bar{c}_{\alpha p}$ = Concentration of species α in phase p in injected volume[(input) $lbmol/ft^3$]
31. \bar{Q}_{α}^* = The total mass injection rate of species α (input) [$lbmol/ft^3day$].

2.2 Assumptions and capabilities

2.2.1 Assumptions and their justification

We make the following simplifying assumptions in this work.

1. Interphase mass transfer between flowing phases is assumed to be locally equilibrium controlled. In this work we consider only the case of linear partitioning between flowing phases and constant partitioning coefficients. Interphase mass transfer between flowing phases and stationary phases is handled by the Adsorption Routine or the General Chemistry Routine.
2. The chemical species are divided into two groups: the species in flowing phases and the species in stationary phases. If some species exists in a stationary and in a flowing phase, it will be treated as two different species in the input file and in the code.

This remark does not apply to linear or non-linear scalar adsorption which can be handled directly in TRCHEM.

3. The density ρ_p and the viscosity μ_p for each phase are currently assumed to be independent of the concentration of species.
4. It is assumed that the rock (solid phase) is not strongly involved in chemical reaction (such as dissolution of the rock by acid), so that the permeability tensor K and the porosity Φ are not affected by the reactive transport process.
5. The bulk source-sink term due to mass transfer between phases is assumed to be small so that it can be ignored in the flow equation.

Justification for these assumptions follows.

Assumption #1 is the key assumption in our model and it enables us to obtain the phase-summed transport equations. For many problems we are interested in, in particular, for many environmental subsurface problems, the mass transfer rate between flowing phases is much faster than the rate of change of concentrations in flowing phases with time, hence the local equilibrium assumption. The kinetic model in which the mass transfer rate between flowing phases is explicitly computed is in theory more generic than the equilibrium model. However, kinetic model is very inefficient when used to simulate fast interphase mass transfer problems. In addition, it is so different from the equilibrium model that it requires separate implementation. Therefore, only the local equilibrium model is used in our project.

Moreover, we use the linear partitioning for phase equilibrium because it is the simplest and computationally the cheapest phase equilibrium relation [5, 9]. The more general non-linear phase equilibrium thermodynamics cannot be incorporated in this model because it will in general make it impossible to obtain the phase-summed transport equations.

Assumption #2 is just for convenience of book-keeping. It does not require any restriction on the physical model.

Assumptions #3 through #5 ensure that the reactive-transport process will not noticeably influence the flow process(es). The structure of the code allows to relax these restrictions in future through a feedback loop from transport back to flow. However, for strong coupling between transport and flow equations one needs a coupled compositional formulation which is outside the scope of TRCHEM.

2.2.2 Internal unit system used

The specific internal unit system was selected for the following reasons.

1. Computational accuracy: We want to avoid units which can make a quantity too large or too small in the floating point representation.
2. Computational efficiency: We want the units to be consistent internally among all physical quantities in order to eliminate or reduce the computational cost for internal unit conversion. We also want the units in TRCHEM to be as consistent as possible with those in existing flow models in order to reduce costs of inter-model unit conversion.
3. Development and maintenance: Another reason for a consistent internal unit system is to reduce the programmer's effort in code development and maintenance. The same unit should be used for the same physical quantity in all subroutines.
4. Compatibility: We choose internal units commonly used in engineering for as many quantities as we can.
5. Convenience in handling chemistry: This criterium influences the choice of units for concentrations of species in each phase. Two of the common options are: i) units based on molar concentrations defined as amount of the species (in moles) per unit volume of phase; and ii) units based on mole fractions defined as ratio of amount of the individual species over total amount of all species in the phase. The latter unit (mole fraction) is convenient for handling non-linear phase equilibrium thermodynamics and for this reason it is frequently used in compositional models. On the other hand, the former unit (molar concentration) is convenient to handle chemistry, in particular the mass-action kinetics type of chemistry. In TRCHEM we use molar concentrations. This choice requires that velocities u_p are in volumetric rather than mass units.

2.2.3 Expected model capabilities

The TRCHEM model is expected to have following capabilities.

1. It can handle multiple flowing phases and multiple stationary phases. The number of flowing phases is arbitrary in this model. However, most of the time only one, or two, or three flowing phases are used. The number of stationary phases is also arbitrary.
2. It can handle diffusion and dispersion. Both of the longitudinal dispersivity and the transverse dispersivity can be rock-type and phase-type dependent.

Note: In the current implementation, TRCHEM cannot handle dispersion.

3. Each of the flowing phases can be incompressible or it can be compressible. That is, the density ρ_p might vary with position and time.
4. The rock (formation) can be incompressible or can be slightly compressible. If it is slightly compressible, then the porosity Φ is taken to be a function of pressure only.
5. The TRCHEM model can be coupled easily with any flow model in IPARS framework.

Note: In the current implementation, only single phase implicit model or hydrology implicit model is coupled with TRCHEM model.

6. TRCHEM can handle general biogeochemistry including adsorption, ion-exchange, precipitation, dissolution, bioremediation and radionuclide decay.
7. Three types of chemistry are supported. These are: equilibrium controlled reaction, classical mass-action kinetics reaction and Monod type kinetics reaction.
8. The handling of equilibrium controlled reaction is relatively robust, even when mineral (stationary) phases precipitate into existence or dissolve away, because it uses an interior-point algorithm to minimize the Gibbs free energy.

2.3 Multiphase Flow Model

In this section we briefly review formulation of multiphase flow models which includes the flow models that can be coupled with TRCHEM.

2.3.1 Special case: two phase flow model

Here we consider a generic two-phase immiscible flow model with one wetting phase or component being denoted by subscript $p = w$ and the other nonwetting phase (oil or air) denoted by subscript $p = nw$. For details, see IPARS manual [2]. Other models can have different notation, see black-oil model manual [7].

The mass conservation equations written for $N_p = S_p \rho_p$ and P_p being the respective phase / component concentration and pressure for $p = w, nw$ are

$$\frac{\partial \Phi N_p}{\partial t} + \nabla \cdot (\rho_p u_p) = q_p, \quad (1)$$

with Darcy's law for multiphase flow written as

$$u_p = -K \frac{k_p}{\mu_p} (\nabla P_p - \rho_p G \nabla D). \quad (2)$$

These are complemented by

$$S_w + S_{nw} = 1, \text{ volume balance,} \quad (3)$$

$$P_{nw} = P_w + P_c(S_w) \text{ capillary pressure relationship.} \quad (4)$$

Also, both fluid phases are assumed to be compressible so that

$$\rho_p = \rho_{p,ref} \exp(c_p P_p). \quad (5)$$

In addition, if the non-wetting phase is air, then instead of Equation (5) one can use some other relationship (for example, real gas law) to describe $\rho_p = \rho_p(P_p)$. Also, Darcy's law (2) can be replaced by some other momentum equation, for example Forchheimer's.

Finally, terms q_p in equation (1) represent well (mass) rates, see Section 5.

2.3.2 Special case: single phase flow model

If there is only one flowing phase (for example, water $p = w$), then $S_p = 1, k_p = 1$ and the system (1)–(2) is reduced to

$$\frac{\partial \Phi \rho_p}{\partial t} - \nabla \cdot \left(\rho_p K \frac{1}{\mu_p} (\nabla P_p - \rho_p G \nabla D) \right) = q_p. \quad (6)$$

2.3.3 Discretization in space

The above flow model(s) can be discretized in space with the use of cell-centered finite differences or equivalently with the expanded mixed finite element methods of lowest order Raviart Thomas spaces on a rectangular grid following the procedure as in [14]. Discretization in time is implicit or explicit, with various solvers used to solve the resulting algebraic system. Other formulations (Discontinuous Galerkin, unstructured grids) are underway.

2.3.4 Choice of flow model in input

The keyword `FLOW_MODEL` specifies the choice of flow model. Currently the following models available under `IPARStv2` can be used as a flow model in `TRCHEM`:

- This model is described by (1)-(2)

```
FLOW_MODEL = "HYDROLOGY_IMPLICIT" $$ hydroi = model 5 = two phase
```

- This model is described by (6)

```
FLOW_MODEL = "SINGLE_PHASE_IMPLICIT" $$ singlei= model 12 = single phase
```

Only one flow model can be chosen currently. This does not include `MultiModel` or `multiblock` extensions to flow models.

2.3.5 Special case: time independent flow part

In general, the conservation equations (1)-(2) describe a transient problem. Even if the system is single phase and incompressible, the source (well) terms q_p or boundary conditions may be time-dependent. This requires the solution of the flow problem in every (flow) time step. In addition, most flow problems are initialized using hydrostatic equilibrium conditions and the solution after the first flow step significantly differs from the one at equilibrium.

In some particular cases however, the solutions can be known to be time-independent and the user may wish the code to by-pass the solution of the flow step after the first one, as it may be very time consuming. This is enabled by the `TRCHEM` model with the use of flag `SKIP_FLOW`, see Section 7. When using `SKIP_FLOW`, all velocities and other flow-related parameters are computed only once and there is no need to interpolate them between time steps, see Section 3.

2.4 Derivation of Phase-Summed Reactive Transport Equations

2.4.1 Definitions

The mass conservation equations are written for each stationary phase s and for each flowing phase p .

For each species α in each stationary phase s we have

$$\frac{\partial \Phi c_{\alpha s}}{\partial t} = R_{\alpha s}^I + R_{\alpha s}^C + R_{\alpha s}^N \quad (7)$$

where the source terms include contributions from interphase transfer R^I , chemical reactions R^C and radionuclide decay R^N , all functions of the concentrations. Please note that though it would be more physically correct for the concentration of immobile species (i.e. the species existed in

stationary phase or phases) to be defined as as amount of species per unit area of soil surface, this definition is not convenient to use. Thus, for convenience of presentation and implementation, the concentration of immobile species is defined as amount of species (in pound-moles) per unit pore volume, which is the same as the definition of concentration of mobile species for single flowing phase case. There is no advective or diffusive/dispersive mass flux term in above equation as the phase s is stationary.

For the flowing phases, the amount of species α accumulated in a flowing phase p per unit volume is equal to $\Phi S_p c_{\alpha p}$. Source terms similar as above R^I, R^C, R^N are allowed but here the reaction terms are proportional to phase volume. Additionally, there is an advective and diffusive/dispersive mass flux due to phase velocities as well as to diffusion and hydrodynamic dispersion. The mass conservation equations are written as

$$\frac{\partial \Phi c_{\alpha p} S_p}{\partial t} + \sum_i \frac{\partial J_{\alpha p i}}{\partial x_i} = R_{\alpha p}^I + \Phi S_p R_{\alpha p}^C + \Phi S_p R_{\alpha p}^N + q_{\alpha p}. \quad (8)$$

The definition of mass flux density $J_{\alpha p i}$ for each flowing phase and each species α includes contributions from advection, diffusion and dispersion as follows

$$J_{\alpha p i} = c_{\alpha p} u_{p i} - \Phi S_p \sum_j D_{\alpha p i j} \frac{\partial c_{\alpha p}}{\partial x_j}, \quad (9)$$

where diffusion/dispersion tensor

$$D_{\alpha p i j} = D_{\alpha p i j}^{\text{diff}} + D_{\alpha p i j}^{\text{hyd}} \quad (10)$$

The molecular diffusion tensor $D_{\alpha p i j}^{\text{diff}}$ can be computed by (see Section 9.1),

$$D_{\alpha p i j}^{\text{diff}} = \tau_p d_{\text{mol}, \alpha p} \delta_{i j}$$

The mechanical dispersion tensor $D_{\alpha p i j}^{\text{hyd}}$ is a function of the Darcy velocities (see Section 9.2 for its derivation and typical values)

$$\Phi S_p D_{\alpha p i j}^{\text{hyd}} = d_{\text{trans}, p} \|u_p\| \delta_{i j} + (d_{\text{long}, p} - d_{\text{trans}, p}) \frac{u_{p i} u_{p j}}{\|u_p\|}$$

In summary, the aggregate diffusion/dispersion tensor is calculated by

$$\Phi S_p D_{\alpha p i j} = (\Phi S_p \tau_p d_{\text{mol}, \alpha p} + d_{\text{trans}, p} \|u_p\|) \delta_{i j} + (d_{\text{long}, p} - d_{\text{trans}, p}) \frac{u_{p i} u_{p j}}{\|u_p\|}$$

Substituting the (9) into the (8), we have

$$\frac{\partial \Phi c_{\alpha p} S_p}{\partial t} + \sum_i \frac{\partial c_{\alpha p} u_{p i}}{\partial x_i} - \sum_i \frac{\partial}{\partial x_i} \left(\Phi S_p \sum_j D_{\alpha p i j} \frac{\partial c_{\alpha p}}{\partial x_j} \right) = R_{\alpha p}^I + \Phi S_p R_{\alpha p}^C + \Phi S_p R_{\alpha p}^N + q_{\alpha p}. \quad (11)$$

2.4.2 Phase-Summed Equations

In order to derive the phase-summed counterpart of (11), we first note that the net interphase transfer is zero

$$\sum_p R_{\alpha p}^I + R_{\alpha}^A = 0. \quad (12)$$

where, we only summing the flowing phases; R_α^A is net mass transfer per porous medium volume of component α into (positive) or out of (negative) rock (all stationary phases). The term R_α^A can be handled by chemistry routine for general mass transfer model between flowing phases and stationary phases. For adsorption, it can be handled by efficiently implemented routine for linear adsorption and specialized routine for some non-linear adsorption. Also, we assume that the partitioning of species between phases can be described by a fixed relationship

$$c_{\alpha p} = \Gamma_{\alpha p} c_{\alpha r} \quad (13)$$

where r is reference phase and $\Gamma_{\alpha p}$ is a constant. In this work we assume water-wet rocks and we choose water phase as the reference phase, see Remark below.

Now we sum the continuity equation (11) over all flowing phases and use equation (12) to get the phase-summed transport equation for species α in the flowing phases

$$\frac{\partial \Phi_\alpha^* c_{\alpha w}}{\partial t} + \sum_i \frac{\partial u_{\alpha i}^* c_{\alpha w}}{\partial x_i} - \sum_i \frac{\partial}{\partial x_i} \left(\sum_j D_{\alpha ij}^* \frac{\partial c_{\alpha w}}{\partial x_j} \right) = R_\alpha^{TC} + R_\alpha^A + R_\alpha^{TN} + q_\alpha^T \quad (14)$$

where we have defined

$$\begin{aligned} \Phi_\alpha^* &= \Phi \sum_p \Gamma_{\alpha p} S_p \\ u_{\alpha i}^* &= \sum_p \Gamma_{\alpha p} u_{pi} \\ D_{\alpha ij}^* &= \Phi \sum_p S_p \Gamma_{\alpha p} D_{\alpha pij} = \left(\Phi \sum_p S_p \tau_p \Gamma_{\alpha p} d_{mol, \alpha p} + \sum_p d_{trans, p} \|u_p\| \Gamma_{\alpha p} \right) \delta_{ij} \\ &\quad + \sum_p (d_{long, p} - d_{trans, p}) \frac{u_{pi} u_{pj} \Gamma_{\alpha p}}{\|u_p\|} \\ R_\alpha^{TC} &= \Phi \sum_p S_p R_{\alpha p}^C \\ R_\alpha^{TN} &= \Phi \sum_p S_p R_{\alpha p}^N \\ q_\alpha^T &= \sum_p q_{\alpha p} \end{aligned}$$

Equation (14) or

$$\frac{\partial \Phi_\alpha^* c_{\alpha w}}{\partial t} + \nabla \cdot (\mathbf{u}_\alpha^* c_{\alpha w} - \mathbf{D}_\alpha^* \nabla c_{\alpha w}) = R_\alpha^{TC} + R_\alpha^A + R_\alpha^{TN} + q_\alpha^T \quad (15)$$

is in fact a system for all species α . It is solved for concentration of species in the reference phase $c_{\alpha w}$. Concentrations in other phases can then be computed through the relationship (13). In TRCHEM this system is discretized in time and in space and it is solved by *time-splitting*. The time-splitting and detailed formulation of its elements are presented in Section (3).

Remark 1: Note that in general, we can choose different reference phases for difference species and the equations obtained will be in the same form as (14). Also, we can use the concept of total concentration to get the phase-summed transport equations derived without the concept of the reference phase, see Section (9.4).

Remark 2: Note that adsorption is represented by the term R_α^A , which can be handled either by the general chemistry routine or by efficiently implemented routine for linear adsorption and

specialized routine for some non-linear adsorption. Note that the first order decay chemical reaction (including radionuclide decay) is represented by the term R_α^{TN} , which can be handled either by the general chemistry routine or by the efficiently implemented routine solved with diffusion together.

2.4.3 Special Case – Single Flowing Phase

Assume that only one phase (for example, water phase) is present. Then the specific definitions for species α are

$$\begin{aligned}\Phi_\alpha^* &= \Phi \\ u_{\alpha i}^* &= u_{wi} \\ D_{\alpha ij}^* &= (\Phi \tau d_{mol, \alpha w} + d_{trans} \|u_w\|) \delta_{ij} + (d_{long} - d_{trans}) \frac{u_{wi} u_{wj}}{\|u_w\|} \\ R_\alpha^{TC} &= \Phi R_{\alpha w}^C \\ q_\alpha^T &= q_{\alpha w}.\end{aligned}$$

Note that in the above equations we only need \mathbf{u}_w at each time which is computed by the flow model. In addition, if rock is compressible, then Φ has to be computed by the flow model at every time step.

2.4.4 Special Case – Two Flowing Phases

If two flowing phases: water and air $p = w, a$ are present, then we have

$$\begin{aligned}\Phi_\alpha^* &= \Phi (\Gamma_{\alpha a} S_a + S_w) \\ u_{\alpha i}^* &= \Gamma_{\alpha a} u_{ai} + u_{wai} \\ D_{\alpha ij}^* &= (\Phi (\tau_a S_a \Gamma_{\alpha a} d_{mol, \alpha a} + \tau_w S_w d_{mol, \alpha w}) + (d_{trans, a} \|u_a\| \Gamma_{\alpha a} + d_{trans, w} \|u_w\|)) \delta_{ij} \\ &\quad + (d_{long, a} - d_{trans, a}) \frac{u_{ai} u_{aj} \Gamma_{\alpha a}}{\|u_a\|} + (d_{long, w} - d_{trans, w}) \frac{u_{wi} u_{wj}}{\|u_w\|} \\ R_\alpha^{TC} &= \Phi (S_a R_{\alpha a}^C + S_w R_{\alpha w}^C) \\ q_\alpha^T &= (q_{\alpha a} + q_{\alpha w}).\end{aligned}$$

Note that since $S_a = 1 - S_w$, in the above equations we only need $S_w, \mathbf{u}_w, \mathbf{u}_a$ and possibly also porosity values if rock is compressible to compute Φ^*, \mathbf{u}^* .

2.5 Chemistry terms

Here we describe general chemistry routine. It is the terms R^C (and also possibly the terms R^A and/or R^N if adsorption and first order decay chemistry are put in the general chemistry routine) which appear in equation (15). We assume that the system is comprised of N_S species of which N_C are components and N_R products. When convenient, we distinguish N_R^K kinetic and N_R^Q equilibrium products ($N_R = N_R^K + N_R^Q$). We let N_M denote the number of minerals and I_M the corresponding index-set. For convenience of presentation, we use i, j for the subscript for species rather than α in this subsection.

2.5.1 Kinetic Reaction – Classical Rate Law

We allow fairly general rate-expressions for the N_R^K kinetic reactions. For each product of this kinetic reaction type, we express the kinetic rates as the difference between a *forward* and a *backward* rate

$$R_{ip}^C = k_{ip}^f \prod_{j=1}^{N_C} (c_j)^{p_{ji}} - k_{ip}^b c_{N_C+i} \quad (16)$$

where k_{ip}^f and k_{ip}^b are the *forward* and *backward* rate constants, respectively, and $p \in \mathbb{R}^{N_C \times N_R^K}$ is a matrix of powers on the components in the rate law.

2.5.2 Kinetic Reaction – Monod Style Rate Law

Monod introduced an empirical expression of the form

$$\frac{dX}{dt} = kX \frac{S}{K + S}$$

to account for the growth of microbes X on a substrate S . The constant K is commonly referred to as the “half saturation constant”. This expression is widely used to model biologically mediated reactions. We allow a generalized version of this expression. For each product of the N_R^M Monod reactions, we have

$$R_{ip}^C = k_{ip}^f \prod_{j=1}^{N_C} (c_j)^{p_{ji}} \prod_{j=1}^{N_C} \frac{c_j}{K_{ij}^{half} + c_j} - k_{ip}^b c_{N_C+i} \quad (17)$$

where K_{ij}^{half} are the half saturation constants for the component species (more than one component is allowed to participate in each reaction) and N_R^M is the number of Monod style rate expressions. When $K_{ij}^{half} = 0$ for all j the Monod expression reduces to the distance-from-equilibrium expression. Typical bioreactions are irreversible, and the backward rate constant k_{ip}^b is usually set to zero.

2.5.3 Equilibrium type (for fast reaction)

We use the standard mass-action expressions. That is,

$$c_{N_C+i} = K_i \prod_{j=1}^{N_C} (c_j)^{\hat{a}_{ji}}, \quad i = 1, \dots, N_R, \quad (18)$$

with a special case for minerals ($i \in I_M$),

$$\begin{aligned} K_i \prod_{j=1}^{N_C} (c_j)^{\hat{a}_{ji}} &= 1 & \text{if } c_{N_C+i} > 0, \\ K_i \prod_{j=1}^{N_C} (c_j)^{\hat{a}_{ji}} &\leq 1 & \text{if } c_{N_C+i} = 0, \end{aligned} \quad (19)$$

where K_i is the equilibrium constant in the reference phase for the i th reaction and \hat{a}_{ji} are stoichiometric coefficients.

2.5.4 Radionuclide Decay

Generally speaking, it includes all of the first order decay reactions. The decay rate is proportional to the concentration,

$$R_{ip}^C = -\frac{\ln 2}{t_i^{half}} c_{ip} \quad (20)$$

where, t_i^{half} is the half life time for species i .

2.5.5 Adsorption

The linear equilibrium adsorption is modeled by,

$$c_{is} = K_{is}^A c_{ir} \quad (21)$$

where, K_{is}^A is the adsorption constant for species i between solid phase s and the reference phase r .

3 Time splitting algorithm

3.1 Initial condition

The algorithm assumes that initial conditions for the flow as well as values of all concentrations are known.

The flow model is initialized using its own initialization routines. In most cases the flow model assumes hydrostatic equilibrium or that Darcy's velocities for all phases are zero with some known values of pressures or saturations at a reference depth. To determine pressures and/or saturation values at time $t = 0$, special algorithm is applied. See specific flow models for details.

The values of concentrations at $t = 0$ are read from input file.

3.2 Time stepping

Flow time step: Assume that the flow model is discretized in time so that its solutions (pressures, saturations and velocities etc.) are available at every *flow time step* $t = t^0, t^1, \dots, t^n, \dots$.

Concentration time step: Consider that at time $t = t^m$ we are given c_α^m for all species α and that we want to take a *concentration step* that is, we want to compute c_α^{m+1} at time $t = t^{m+1}$. The *concentration time step* in general is different (smaller) than the *flow time step*. We will assume that $(t^m, t^{m+1}) \subset (t^n, t^{n+1})$ and that values $u_{\alpha i}^*$ and Φ_α^* are available at the old and at the new flow time steps $t = t^n, t^{n+1}$. Since in the current implementation the transport results do not influence the flow, this is not a restriction.

Denote $\Delta t = t^{m+1} - t^m$ and set $T_\alpha = \Phi_\alpha^* c_{\alpha w}$. Direct discretization in time of system (15) gives

$$\frac{T_\alpha^{m+1} - T_\alpha^m}{\Delta t} + \nabla \cdot (\mathbf{u}^{*,m+1/2} c_{\alpha w}^m - \mathbf{D}_\alpha^{*,m} \nabla c_{\alpha w}^{m+1}) = R_\alpha^{TC,m+1/2} + R_\alpha^{TN,m+1/2} + q_\alpha^{T,m+1/2} \quad (22)$$

in which we use phase-summed properties Φ^* and velocities \mathbf{u}^* at $t \in (t^m, t^{m+1})$ computed by linear interpolation between old and new flow time steps.

Direct solution of system (22) is difficult and in case of complex chemical reactions it may be practically impossible even if some terms like advection are treated explicitly. For this reason, a time-splitting algorithm described below is used. This means that the advection, diffusion/dispersion and chemistry/radionuclide decay subproblems are solved “independently” in the

sense that they deliver intermediate values of T_α as $\bar{T}_\alpha, \hat{T}_\alpha, T_\alpha^{m+1}$ using the step-specific numerical algorithms and discretizations. Below we define the individual steps of this procedure. Some stability and consistency restrictions may apply [5].

3.3 Advection

The equation solved in advection step is formally

$$\frac{\partial \Phi_\alpha^* c_{\alpha w}}{\partial t} + \sum_i \frac{\partial u_{\alpha i}^* c_{\alpha w}}{\partial x_i} = q_\alpha^T \quad (23)$$

Currently available is first order Godunov method.

First order Godunov method: The velocities $u_{\alpha i}^{*,m+1/2}$ are computed at $t = t^{m+1/2}$ by interpolating between the values obtained for new and old flow time steps.

Set $T_\alpha^m = \Phi_\alpha^{m,*} c_{\alpha w}^m$ and compute explicitly \bar{T}_α from

$$\frac{\bar{T}_\alpha - T_\alpha^m}{\Delta t} + \sum_i \frac{\partial}{\partial x_i} \left(u_{\alpha i}^{*,m+1/2} c_{\alpha w}^m \right) = q_\alpha^{T,m+1/2} \quad (24)$$

Intermediate values of concentrations after the advection step can be retrieved by computing $\bar{c}_{\alpha w} = \bar{T}_\alpha / \Phi_\alpha^{m+1,*}$.

3.4 Chemical Reaction

We solve the equation

$$\frac{\partial \Phi_\alpha^* c_{\alpha w}}{\partial t} = R_\alpha^{TC}$$

for kinetic reaction type (other reaction types to be discussed as follows) by ODE integration explicitly. For example, if Forward-Euler first-order scheme is used, it is solved by

$$\frac{\hat{T}_\alpha - \bar{T}_\alpha}{\Delta t} = R_\alpha^{TC}$$

Note that if we put the radionuclide decay reaction and the adsorption into the general chemistry routine, the equation to be solved is essentially

$$\frac{\partial \Phi_\alpha^* c_{\alpha w}}{\partial t} = R_\alpha^{TC} + R_\alpha^{TN} + R_\alpha^A$$

The detailed techniques available to solve the chemistry will be listed as follows.

3.4.1 Equilibrium controlled reaction

For pure equilibrium controlled reaction, we perform flash-calculation by using interior point algorithm to minimize the Gibbs free energy.

3.4.2 Kinetics type reaction

For pure kinetic reactions, we perform the standard explicit ODE integration. Currently, three options are supported, that is, the Forward Euler first-order scheme, second-order Runge-Kutta and fourth-order Runge-Kutta.

3.4.3 Mixed kinetic and equilibrium reaction system

For mixed kinetic and equilibrium reaction system, we perform a series of ODE solves and flash calculation.

3.5 Diffusion/Dispersion

We solve the equation

$$\frac{\partial \Phi_{\alpha}^* c_{\alpha w}}{\partial t} - \sum_i \frac{\partial}{\partial x_i} \left(\sum_j D_{\alpha ij}^* \frac{\partial c_{\alpha w}}{\partial x_j} \right) = 0$$

implicitly with respect to concentrations

$$\frac{T_{\alpha}^{m+1} - \hat{T}_{\alpha}}{\Delta t} - \sum_i \frac{\partial}{\partial x_i} \left(\sum_j D_{\alpha ij}^* \frac{\partial c_{\alpha w}^{m+1}}{\partial x_j} \right) = 0. \quad (25)$$

Note that dispersion tensor is time-lagged. Currently available solver for diffusion/dispersion step is one described below which can only solve the diffusion problem.

Diffusion solver: This solver is based on cell-centered finite difference method and a standard IPARS 7-point stencil. The resulting linear system is not necessarily symmetric in the case of multiple flowing phases because, in general, the saturations vary in space. The solver used is GMRES with one of many preconditioners available. Tolerance for the linear solver is set-up differently than for the solver used in the flow step.

The use of a standard IPARS 7-point stencil solver restricts i) the kind of boundary conditions that can be applied and ii) the use of general (full) tensor. In particular, because of restriction i), we cannot solve Equation (25) with Robin- or inflow- boundary conditions. On the other hand, restriction ii) makes it impossible to solve full hydrodynamic dispersion problems in particular, with significant transversal dispersivity.

3.6 Radionuclide Decay

We call it radionuclide decay, but it also includes all types of first-order decay chemical reaction. It can be handled by the general chemistry routine as mentioned above. But it is recommended that it be solved by the efficient implementation. The efficient implementation is to solve it implicitly by combining into the diagonal elements of the discrete diffusion/dispersion formulation. It is more computational efficient, and also increases stability and robustness.

3.7 Adsorption

Though the linear equilibrium type adsorption can be handled by the general chemistry routine, it is recommended that it be solved by the efficient implementation. The efficient implementation is to solve it by modifying the coefficient of the accumulation term by counting the contribution from adsorption. It is more computational efficient since it completely eliminates the flash calculation. It also increases stability and robustness.

4 General boundary conditions

Here we discuss boundary conditions or conditions specified on the boundary $\partial\Omega$ of the physical domain Ω .

The use and implementation of boundary conditions in IPARS is described in [12]. In general, specific models in IPARS can make use of the boundary conditions using framework dependent code templates which handle input and application of specific boundary conditions. For each cell adjacent to the boundary, the framework identifies which of the cell's faces belong to a boundary region specified by the user in an input file. Such faces are called “boundary faces” below.

The boundary conditions relevant to TRCHEM module are these applied to the flow model as well as those applied to transport-chemistry part. Specifically, consider equation (14). In this equation, the right hand side terms $R_\alpha^{TC} + R_\alpha^{TN} + q_\alpha^T$ are local to a cell and they are handled by the chemistry package. The interaction of TRCHEM with boundary conditions is only through the normal part of the advective and diffusive fluxes

$$(\mathbf{c}\mathbf{u} - \mathbf{D}\nabla c) \cdot \boldsymbol{\nu} \quad (26)$$

where for simplicity of notation we suppressed the subscripts and superscripts α , $*$ and where $\boldsymbol{\nu}$ is the outward unit normal to $\partial\Omega$.

The values of normal of Darcy's velocities across boundary $\mathbf{u} \cdot \boldsymbol{\nu}$ depend only on the flow boundary conditions, see Section 4.1. The additional conditions to be specified on the concentration c or on its gradient ∇c are discussed in Section 4.2 and their input in Section 4.3. The time-splitting of these conditions is defined in Section 4.4.

4.1 Flow boundary conditions

After the flow problem is solved, the flow model code postprocesses the flow data and it computes normals of the appropriate phase-summed Darcy velocities (volumetric flux) leaving or entering the boundary face(s) $\mathbf{u} \cdot \boldsymbol{\nu}$.

In flow models in IPARS, for each boundary face or for each $x \in \partial\Omega$, there are two general possibilities for boundary conditions:

1. Boundary conditions are not specified (default) or are specified as **no-flow**. In this case the Darcy velocities for each phase across the face are equal to zero $u_{pi} = 0$. This implies that the normal of phase summed velocities $\mathbf{u} \cdot \boldsymbol{\nu} = 0$ for each species.
2. Boundary conditions of various type are specified. For example, Dirichlet values of pressure or of potential can be imposed in the single phase flow model. In general, we associate with the boundary face one of the two types depending on the sign of the velocity:

inflow type $\mathbf{u} \cdot \boldsymbol{\nu} < 0$ (negative normal velocity)

outflow type $\mathbf{u} \cdot \boldsymbol{\nu} > 0$ (positive normal velocity)

One can then set $\partial\Omega = \partial\Omega_{in} \cup \partial\Omega_{out} \cup \partial\Omega_0$, where $\partial\Omega_{in}$, $\partial\Omega_{out}$ and $\partial\Omega_0$ are disjoint sets and represent the inflow, outflow and noflow parts of type boundary, respectively.

4.2 Boundary conditions in advection/diffusion

The solutions of the advection–diffusion problem (14) depend on the type of boundary conditions imposed. Traditionally, in the elliptic or parabolic theory, one distinguishes between Dirichlet, Neumann or Robin conditions but these names may be confusing in the current context as the advection-diffusion equation can be parabolic-elliptic-hyperbolic in character depending on the case. In addition, the Dirichlet / Neumann parts of the boundary in general may or may not overlap with the inflow, outflow, and no-flow boundaries what can add to the confusion and complicate the analysis.

For example, consider equation (26) on the inflow boundary $\partial\Omega_{in}$ and assume that the total of the normal advective and diffusive flux is prescribed:

$$(c\mathbf{u} - \mathbf{D}\nabla c) \cdot \boldsymbol{\nu} = c_B \mathbf{u} \cdot \boldsymbol{\nu} \quad x \in \partial\Omega_{in}. \quad (27)$$

For this condition, it is necessary to set $c_B(x), x \in \partial\Omega_{in}$. These values propagate along characteristics originating from $\partial\Omega_{in}$. What follows is that if values of c are prescribed also on $\partial\Omega_{out}$, then these should be consistent with those propagating from $\partial\Omega_{in}$. Otherwise, a boundary layer of size proportional to \mathbf{D} will arise.

As another example, one needs to specify conditions on the no-flow boundary $\partial\Omega_0$. As the advective flux drops out, one of the possibilities for the diffusive flux is to use homogeneous Neumann condition $\mathbf{D}\nabla c \cdot \boldsymbol{\nu} = 0$. However, in some environmental applications, the use of condition $c = 0$ is advised as pessimistic means to predict “maximum” of the diffusive/dispersive flux leaking out of the domain.

In addition, the spatial discretization of the advection–diffusion problem and the time–splitting algorithm may impose some stability and consistency restrictions on the boundary conditions that can be applied. In particular, in the current implementation of TRCHEM model, the application of the most general form of the inflow (Robin) condition to the diffusion part requires extensions of the linear solver which at the present time are not available. However, some special cases of Robin condition can be handled. For example, if $\mathbf{u} = 0$ then the condition (27) is reduced to $\mathbf{D}\nabla c \cdot \boldsymbol{\nu} = 0$. On the other hand, if $\mathbf{D} = 0$ or if diffusion step is skipped in the time-splitting algorithm, then the condition (27) is reduced to $c = c_B$.

4.3 Input of boundary conditions

Here we list the input flags and values for boundary conditions available in TRCHEM. Note that the code does not verify the soundness of the imposed boundary conditions but merely attempts to apply the conditions requested by the user.

- Boundary condition flag BOUND_TYPE_D. Two values are allowed at the current time:

BOUND_TYPE_D=0 (Default) which means “zero normal diffusive flux” or

$$\mathbf{D}\nabla c \cdot \boldsymbol{\nu} = 0.$$

BOUND_TYPE_D=2 which means: prescribed Dirichlet value in diffusion equation.

Note that this flag divides the boundary $\partial\Omega$ in two parts: $\partial\Omega_{D=0}$ and $\partial\Omega_{D=2}$.

- Prescribed value $c = c_B$. This value can be used by both advection and by the diffusion parts. It is input as a time-dependent quantity using IPARS framework table input capability. For example, for species number 2 in boundary region 5, the specification is

```
TRBOUND2(5) Block $ IODINE
  Interpolation Linear
  Extrapolation Constant
  Data 0. 1. , 100. 0.5
EndBlock
```

This condition is time dependent. In the algorithm, when solving concentration step $t^m \rightarrow t^{m+1}$, we use $c_B = c_B(t^{m+1})$. In other words, it is the value provided by table lookup at $t = t^{m+1}$.

In general, see IPARS manual [2] for more information on table input. If the table for some region is not input but it is required by the algorithm then the default value used is one of $c_B = 0$.

In the next section we provide interpretation of these input values in the advection and in the diffusion parts as well as their meaning for the full (un-split) problem.

4.4 Splitting of boundary conditions

Since we use the time splitting technique to solve the equation (14), the boundary conditions must be split between advection part and between dispersion/diffusion part.

As mentioned before, in some cases the splitting will require solving the diffusion/dispersion equation with a Robin-type condition which we cannot handle at the present time.

4.4.1 Boundary conditions in advection part

The boundary conditions in the advection part are determined by the flow velocity \mathbf{u} . Since the advection equation is a discrete hyperbolic problem, these conditions are only meaningful on the inflow boundary $\partial\Omega_{in}$.

$$\begin{aligned} \frac{\partial \Phi_{\alpha}^* c_{\alpha w}}{\partial t} + \sum_i \frac{\partial u_{\alpha i}^* c_{\alpha w}}{\partial x_i} &= q_{\alpha}^T \quad x \in \Omega \\ c_{\alpha w} &= c_{\alpha w, B} \quad x \in \partial\Omega_{in} \end{aligned}$$

4.4.2 Boundary conditions in diffusion/dispersion part

The dispersion/diffusion equation with boundary condition is as follows.

$$\begin{aligned} \frac{\partial \Phi_{\alpha}^* c_{\alpha w}}{\partial t} - \sum_i \frac{\partial}{\partial x_i} \left(\sum_j D_{\alpha ij}^* \frac{\partial c_{\alpha w}}{\partial x_j} \right) &= 0 \quad x \in \Omega \\ \mathbf{D} \nabla c_{\alpha w} \cdot \boldsymbol{\nu} &= 0 \quad x \in \partial\Omega_{D=0} \\ c &= c_B \quad x \in \partial\Omega_{D=2} \end{aligned}$$

4.4.3 Summary of boundary conditions

As explained above, the boundary conditions for the un-split advection/diffusion problem (15) are defined on intersections of $\partial\Omega_{in}$, $\partial\Omega_{out}$, $\partial\Omega_0$ with $\partial\Omega_{D=0}$, $\partial\Omega_{D=2}$. They are summarized in tables 1 and 2.

bdary part	type of condition	definition	req.on \mathbf{u}	value	flag
$\partial\Omega_{out} \cap \partial\Omega_{D=0}$ $\cup \partial\Omega_0 \cap \partial\Omega_{D=0}$	classical no- or outflow condition	$\mathbf{D} \nabla c \cdot \boldsymbol{\nu} = 0$	$\mathbf{u} \cdot \boldsymbol{\nu} \geq 0$	-	0
$\partial\Omega_{out} \cap \partial\Omega_{D=2}$	Dirichlet on no- or outflow boundary	$c = c_B$	$\mathbf{u} \cdot \boldsymbol{\nu} \geq 0$	c_B	2
$\partial\Omega_{in} \cap \partial\Omega_{D=2}$	classical inflow condition	$(c\mathbf{u} - \mathbf{D} \nabla c) \cdot \boldsymbol{\nu} = c_B \mathbf{u}$	$\mathbf{u} \cdot \boldsymbol{\nu} < 0$	c_B	2
$\partial\Omega_{in} \cap \partial\Omega_{D=0}$	inflow with no diffusion	$\mathbf{D} \nabla c \cdot \boldsymbol{\nu} = 0$	$\mathbf{u} \cdot \boldsymbol{\nu} < 0$	c_B	0

Table 1: Available boundary conditions in TRCHEM

bdary part	cond. in ADVECTION	cond.in DIFFUSION	behavior
$\partial\Omega_{out} \cap \partial\Omega_{D=0}$ $\cup \partial\Omega_0 \cap \partial\Omega_{D=0}$	outgoing advective flux= $c\mathbf{u}$	$\mathbf{D}\nabla c \cdot \nu = 0$	advective outflow
$\partial\Omega_{out} \cap \partial\Omega_{D=2}$	outgoing diffusive flux= $\mathbf{D}\nabla C \cdot \nu$	$c = c_B$	boundary layer
$\partial\Omega_{in} \cap \partial\Omega_{D=2}$	incoming advective flux= $c_B\mathbf{u}$	-	can't handle (Robin)
$\partial\Omega_{in} \cap \partial\Omega_{D=0}$	incoming advective flux= $c_B\mathbf{u}$	$\mathbf{D}\nabla c \cdot \nu = 0$	special case of the above

Table 2: Splitting of boundary conditions in TRCHEM

4.5 Output related to boundary conditions

The flow models report the fluxes across the boundary faces if the flag BNDOUT is specified.

In addition, if the flag REPORT_BD_FLUX is used, then TRCHEM reports the accumulated net amount of species flowing across boundary for each individual boundary region (unit in lbmol) as well as balance of species in each individual rock subdomain(unit in lbmol),

5 Wells

The IPARS framework handles input of well position, geometry etc. and flow data as well as output of well mass flow rates. In the TRCHEM model, the wells are represented by the source-sink term q_α^T in the equation (14). More specifically, the wells are handled in the advection subproblem in equation (23). Note that if advection step is not performed, then the wells or source terms described here are not active.

In this section we define these terms and describe the input and output related to wells.

5.1 Flow wells

Most of the flow models in IPARS handle general type wells in at least semi-implicit fashion and with the use of Peaceman correction, see [11, 3]. The wells can be either bottom-hole-pressure (BHP) specified, or (mass) rate specified. Each flow model has its own well models.

In general, the flow models compute the well mass rates per element which appear in the discrete form of equation (1) after it is integrated over element e as

$$q_{p,e} = \int_e q_p(x) dx = \rho_{p,e} Q_{p,e}$$

When added over all well elements, the total mass rate for well w

$$q_p^w = \sum_{e \in w} q_{p,e} = \sum_{e \in w} \rho_{p,e} Q_{p,e}$$

is computed and reported. Each well w in a flow model has an associated positive integer flag $\text{KINDWELL}(w)$ which denotes the type (injection or production, of BHP or rate specified).

Injection wells have nonnegative rates for at least one phase p and zero for all the others. Production wells have nonpositive rates for all phases p .

For the needs of TRCHEM, each associated flow model must deliver the volumetric flow rates $Q_{p,e}$ for each phase and for each well element or grid cell e . These are obtained from the interaction of well/flow models by the following equation (see [2] for details).

$$Q_{p,e} = G_e L_e K_e \frac{k_p|_e}{\mu_p|_e} (P_{WB,e} - \overline{P}_{p,e})$$

Note that G_e is the geometric factor including the Peaceman correction, and that $K_e, k_p|_e, \mu_p|_e$ are absolute permeability, relative permeability and viscosity in the cell e , respectively. Pressure difference is computed between wellbore pressure $P_{WB,e}$ and phase pressure at the center of well element $\bar{P}_{p,e}$ which are computed from phase pressure $P_p|_e$.

5.2 TRCHEM wells

In the TRCHEM model, the wells are either associated with flow wells or not. There are three possibilities: i) production flow wells for which $Q_{p,e} \leq 0, \forall p, e$, ii) injection flow wells for which $Q_{p,e} \geq 0, \forall p, e$ and iii) wells not associated with any flow model for which $Q_{p,e} = 0, \forall p, e$ and which are called “pure transport wells” in this manual.

The sources/sinks of species α in phase p are given by $q_{\alpha p}$ which represents the mass rate per unit volume of phase. In discrete form (after integration over element e) they are given by $q_{p\alpha,e} = \int_e q_{p\alpha}(x)dx$. How $q_{p\alpha,e}$ is defined and computed depends on the type of a given well w :

- i) if well w is a sink in flow model then production rate of species α in produced phase p is

$$q_{p,\alpha e} = c_{\alpha p} Q_{p,e} \quad Q_{p,e} \leq 0, \forall e \in w, \quad (28)$$

- ii) if well w is a source in flow model then injection rate of species α in injected phase p is

$$q_{p,\alpha e} = \bar{c}_{\alpha p} Q_{p,e} \quad Q_{p,e} \geq 0, \forall e \in w, \quad (29)$$

- iii) if well w is not present in flow model then injection rate of species α when phase volume is not injected for any phase p is

$$q_{p,\alpha e} = \bar{Q}_{\alpha p} \quad Q_{p,e} = 0, \forall e \in w. \quad (30)$$

Note that $c_{\alpha p}$ is an unknown quantity but that $\bar{c}_{\alpha p}, \bar{Q}_{\alpha p}$ must come from well data defined in input file.

When deriving phase summed equations, the above well terms are summed over the phases. In particular, for elements of a production well, the total over all phases is

$$q_{\alpha e}^* = c_{\alpha r}|_e \left(Q_{r,e} + \sum_{r \neq p} \Gamma_{\alpha p} Q_{p,e} \right).$$

For (flow) injection well, the total over all phases is

$$q_{\alpha e}^* = \bar{c}_{\alpha r} \left(Q_{r,e} + \sum_{r \neq p} \Gamma_{\alpha p} Q_{p,e} \right). \quad (31)$$

For pure transport well it is

$$q_{\alpha e}^* = \bar{Q}_{\alpha r} \left(1 + \sum_{r \neq p} \Gamma_{\alpha p} \right). \quad (32)$$

For this last quantity, we define the total injection rate per unit volume as

$$\bar{Q}_{\alpha}^* = \frac{1}{|e|} q_{\alpha e}^* \quad (33)$$

Finally, we allow for the two quantities: concentration in the injected reference phase $\bar{c}_{\alpha r}$ and total injection rate \bar{Q}_{α}^* to be specified as input data. see next Section.

For each well w , the total mass rate per well is

$$q_{\alpha}^{w,*} = \sum_{e \in w} q_{\alpha e}^*. \quad (34)$$

For time discretization, see next Section.

5.3 Input for wells

The well specification in input file for TRCHEM can be given for any species and for any injection well. Specification for a production well will have no meaning. Recall that in IPARS input file the type of flow well is defined with parameter $KINDWELL(w)$ which takes positive values for both injection and production wells. To define a well that is a pure transport well, one needs to specify the value of $KINDWELL(w)=-1$.

The injection parameters are represented in the input by time dependent tables beginning with keyword $TRXWELL\alpha(w)$, for species α and well number w . The values given are interpreted in two possible ways depending on whether the well w is active in a flow model or not.

- $KINDWELL(w) > 0$ and well w is an injection well: We define $\bar{c}_{\alpha r} = TRXWELL\alpha(w)$ as the concentration in the reference phase in the injected volume. More precisely, equation (31) defines the total amount of species injected. In particular, if the reference phase is not present in injected volume, then the value $\bar{c}_{\alpha r}\Gamma_{\alpha p}$ represents the concentration in injected phase p .
- $KINDWELL(w) = -1$: We define $\bar{Q}_{\alpha}^* = TRXWELL\alpha(w)$ as the total mass of the injected species α per unit volume.

If $TRXWELL\alpha(w)$ is not specified (given) in the input, then the default value used is zero with interpretation that component α is not present in the injected volume at well w .

We also note that in general, CFL condition applies to production wells in the sense that we cannot produce more mass from a grid-block than what is contained in it. For the same stability reasons, the quantity \bar{Q}_{α}^* should be positive.

Finally, a note on time discretization of well terms. IPARS table lookup provides requested rates or BHP specification at the new flow time step $t = t^{n+1}$ and this is the value used to solve the flow problem by most flow models. This rule is equivalent to using right rectangle rule for time integration of well terms. Therefore, if the requested mass rate (or BHP) is constant over a time interval nontrivially including flow time step (t^n, t^{n+1}) , then the change of time step will not change the solution. This is not true for general table input data and multiphase flow models.

As a consequence in TRCHEM model, in the time-independent case $q_{p,e}(t) = const; t^n \leq t \leq t^{n+1}$, the volumetric rate provided after the flow time step $Q_{p,e}$ can be assumed **constant** over the flow time step $(t^n, t^{n+1}]$. In general, for consistence with velocities, the phase-summed well flow rates are time-averaged in exactly the same way as the velocities. The input quantities are however computed by table lookup at (new) concentration step $t = t^{m+1}$. For example, $\bar{c}_{\alpha r} = \bar{c}_{\alpha r}(t^{m+1})$.

For higher degree of consistency and higher order splines in table input, one needs an extension of table lookup routines by integration routines. These are not available currently under IPARS.

5.3.1 Example

In this example, well 1 is a flow model well and well 3 is a pure transport well. For concentration time steps between $t = 0$ and $t = 1$., the concentration of injected species is calculated by (linear) interpolation between 0 and 1.. For $t \geq 1$, it is assumed constant and equal 2.0.

```

$$$$$$$
WELLNAME(1) = "WELL 1, water injection"
KINDWELL(1) = 1
WELLTOP(1 TO 3,1,1) =    0  5. 5.
WELLBOTTOM(1 TO 3,1,1) = 4  5. 5.
WELLPQ(1) Block
    Interpolation Linear
    Extrapolation Constant
    Data 0.  510.
EndBlock

TRXWELL2(1) Block                                $ species 2 injected in water
    Interpolation Linear
    Extrapolation Constant
    Data
0.  1.0
1.  2.0
EndBlock

$$$$$$$$$
WELLNAME(3) = "WELL 3, pure transport source"
KINDWELL(3) = -1
WELLTOP(1 TO 3,1,3) =    0  50. 50.
WELLBOTTOM(1 TO 3,1,3) = 2  50. 50.

TRXWELL1(3) Block                                $ species 1 injected by itself
    Interpolation Linear
    Extrapolation Constant
    Data 0.  0.2
EndBlock

```

5.4 Output for wells

IPARS framework delivers, if requested, the well output for each well in the standard output file and/or in a special “well” file. These contain information about instantaneous injection/production mass rates for phases and components, water/oil ratio etc. If a well is temporarily inactive, its injection or production rate is reported as 0.

In addition to the flow model output, for all wells that are active in TRCHEM, there is output describing total injection and production rate $q_{\alpha}^{w,*}[lbmol/day]$ for each species α and for each well w as defined in equation (34). Because of a potentially large number of species, only one value per well per species is printed; the sign of value denotes whether this is an injection (+) or production (-) type of rate.

In addition we note that since concentration step does not in general coincide with flow step, it may happen that the well type will change from injection to production or vice versa between two flow time steps. In consequence, volumetric rates obtained for a given concentration step by interpolation between flow steps may appear positive or negative.

6 Getting the Source Code and Executable

Currently TRCHEM model is a part of CVS distribution of IPARSv2. In this section we describe how to get the source code and how to create an IPARSv2 executable for TRCHEM that would include the currently available flow models. The logic here is different than the one used to create an executable for a single model. However, it is somewhat similar to the one used by MultiModel (MMODEL) option in IPARSv2.

6.1 Source code

The current location of the CVS repository for IPARSv2 in TICAM is

```
/ws/csm/SOURCE/IPARSv2
```

The members of **wheeler** group can access it and check it out using `cvs checkout IPARSv2`. For specific commands on your machine or for how to use CVS or how to access the repository, contact CSM.

6.2 How to build a TRCHEM executable:

In general, instructions in IPARS manual [2] or in `*/IPARSv2/make/modular/README` should be followed. The build process for TRCHEM requires some additional modifications described below.

In the following, we show selected lines from `unix.mak` and `ipars.siz` necessary for TRCHEM. For brevity, we did not include lines that are commented out.

6.2.1 Makefile

In `unix.mak` (or your Makefile) un-comment all the flow model and solver components that you want to be included in your executable. Make sure to un-comment also line with `trchem.mak`. Example is as follows.

```
----- Sample Makefile
# unix.mak - Makefile Executive for IPARS on Unix

# Copy this file to ipars/work/makefile then edit it to uncomment the
# appropriate lines

# make                Builds production program
# make clean          Deletes work files

##### Unix/Dos Controls #####

# Define the slash for file names
S=/

# Define the target file name
EXENAM=ipars

# Define the object file extension
O=.o

# Define the copy instruction
```

```

COPY=cp

##### Misc #####

default:$(EXENAM)

SIZE=./size
SIZDAT=ipars.siz
SETSIZE=echo $(SIZDAT) $$ > ech
SETSIZE1=echo $(SIZDAT) $< $$ > ech
SIZEIT=$(SIZE) < ech

MAKDIR=..$(S)make$(S)modular$(S)

WORK=.

COPYIT=$(COPY) $? $(WORK)
COPYIT1=$(COPY) $< $(WORK)

.SUFFIXES:
.SUFFIXES: .o .f .c .C .cpp .h .df .dc .dh .dC .dcpp .in .obj

##### Framework include files #####

##### select one of fraemwork files
include $(MAKDIR)frame_t.mak
##### select one for parallel interface or NONE for single processor machine
# include $(MAKDIR)parall_c.mak

##### Linear Solver #####

include $(MAKDIR)gmres.mak
# include $(MAKDIR)ygmres.mak

##### Multi-block #####
### use mortar.mak to use files from /u11/NDAGH, ../MBLib, /mblk
#
# include $(MAKDIR)/mortar.mak
#
### OR
### use mace.mak to use files from ../MACE
# include $(MAKDIR)/mace.mak

##### Graphics #####

include $(MAKDIR)visual.mak

##### Physical Model include files

include $(MAKDIR)hydroi.mak
include $(MAKDIR)singlei.mak
# include $(MAKDIR)air.mak
include $(MAKDIR)trchem.mak

```



```
##### Combine object/lib files #####

OBJS = $(FRAMEOBJ) $(MORTAROBJ) $(MODELOBJ) $(SOLVEOBJ) $(GRAPHOBJ) $(PARALOBJ)

LIBS = $(FRAMELIB) $(MORTARLIB) $(MODELLIB) $(SOLVELIB) $(GRAPHLIB) $(PARALLIB)

##### Machine and Compiler include file (one only) #####
# This include file must be the last one

include $(MAKDIR)lnx.mak
# include $(MAKDIR)beo.mak
```

6.2.2 Size file

In `ipars.siz` un-comment all the model and solver sizefiles (important: this is different than for MultiModel). For the moment only one common choice of a solver for all flow models is possible. The available associations are listed in file `mmodel.txt`.

```
----- sample ipars.siz
$      ipars.siz - Size Data Executive for all Machines and Models

$  Keep this file as simple as possible.  Most size variable and file names
$  should be set in include files.

$  INCLUDE FILES MARKED WITH AN * DO NOT CURRENTLY EXIST

Target_Directory
./

$ ##### Unix/Dos control (select one) #####

Replace_Symbols

$ PCSLASH YES          $ Convert file names to PC format
PCSLASH NO             $ Do not convert file names to PC format

$ ##### Machine and message system (select one) #####

Include_File

../size/modular/lnx.siz  $ Linux
$ ../size/modular/beo.siz  $ Linux with MPI messages

$ ##### Framework (select one) #####

Include_File

../size/modular/frame1.siz $ 30x105x100, 1 Processor, 20 Wells, 3 Blocks
$ ../size/modular/frame36.siz $ 40x200x200, 36 Processors, 40 Wells, 10 Blocks

$ ##### Physical Model (select one) #####

$ Turn off all models then turn on one model in an include file
```

Replace_Symbols

```
HYDROI   C          $ Exclude hydrology-implicit model
HYDROE   C          $ Exclude hydrology-impes model
BLACKI   C          $ Exclude black-oil implicit
BLACKP   C          $ Exclude black-oil impes
COMP     C          $ Exclude compositional
CHEM     C          $ Exclude chemical-flood
MMODEL   C          $ Exclude multi model
NOMMOD   ""         $ Exclude multi model = not MMODEL
SINGLEEE  C          $ Exclude single-phase sequential flow model
AIR       C          $ Exclude air-water simulation model
SINGLEEI  C          $ Exclude single-phase implicit flow model
TRCHEM   C          $ Exclude transport-chemistry multimodel
DGIMPES  C
DGMISC   C
```

Include_File

```
../size/modular/hydroi.siz  $ Hydrology Implicit Model
../size/modular/singlei.siz  $ Single Phase Model
$ ../size/modular/air.siz     $ Air-Water Model
../size/modular/trchem.siz   $ transport-chemistry multimodel

$ ##### Graphics #####

$ Turn off all graphics code then optionally turn on one or more
$ capabilities in an include file(s)
```

Replace_Symbols

```
VISUAL   C $ disable Tecplot output
DISCOVER  C $ disable connection with DISCOVER
```

Include_File

```
../size/modular/visual.siz  $ TECPLOT visualization          *

$ ##### Multi-block Capability #####

$ Turn off multi-block code then optionally turn on the capability
$ in an include file(s)
```

Replace_Symbols

```
MORTAR   C          $ Exclude mortar code
DUAL     C          $ Exclude dual code
IPARSv3  C          $ Exclude code which is only relevant in IPARSv3
$ and which is not excluded by other keywords
```

Include_File

```
$ ../size/modular/mortar.siz  $ Multi-block mortar code
```

```

$ ##### Linear solver (select one) #####

$ Turn off all linear solvers then turn on one in an include file

Replace_Symbols

TICAMA  C          $ Remove gmres linear solver
YTICAMA C          $ Remove ygmres linear solver
LSOR    C          $ Remove line sor linear solver
PCG     C          $ Remove pcg linear solver

Include_File

../size/modular/ticama.siz  $ Gmres linear solver
$ ../size/modular/yticama.siz $ Gmres linear solver, additions by YVassilevski

```

6.3 Running the code

Follow instructions in IPARS manual [2]: use command line or create file IPARS.IN. Use MPI specific commands for running in parallel.

7 Input

The section below describes only the keywords specific to the TRCHEM model. Please refer to the relevant IPARS documentation [2] on other parts of the input file.

7.1 General Remarks

A few types of Keywords used by this model are listed as follows.

1. Flag type: A flag variable may not be followed by data; only the presence or absence of the variable will be noted by the input processor.
2. Real number (or numeric) type. The variable name must be followed by one numeric entry.
3. Character string: The variable name must be followed by character string entry in quote marks, e.g. "HYDROLOGY_IMPLICIT".
4. Logical: The variable name must be followed by one logical data entry.
5. Array of above type: The variable name must be followed by one or more data entries. See the file "IPARSv2/doc/reader.txt" for detail. In this manual, array of any above type mean 1-dimensional array except indicated otherwise.

Remarks on the units in the input file is as follows.

1. $[ft^3]$ should be written as "cu-ft" in the input file.
2. $[lb - mol]$ should be written as "lbM" in the input file.

7.2 TRCHEM Control Parameters

1. `FLOW_MODEL` (character string): Choose one of “`SINGLE_PHASE_IMPLICIT`” and “`HYDROLOGY_IMPLICIT`”. Other flowing models are not supported so far. This keyword must be specified. No default value is available. Example is as follows.

```
TRANSPORT_CHEMISTRY  
FLOW_MODEL = "HYDROLOGY_IMPLICIT"
```

2. `SKIP_FLOW` (flag): It is used in the case of time independent flow problems (for example: single incompressible phase flow with time-independent boundary and wells flow rates). It is user's responsibility to ensure that the use of the flag makes sense. Execution of flow steps and possibly of some postprocessing routines can be skipped. The default value is “false”. Example is as follows.

```
SKIP_FLOW $ use this or do not skip flow but
```

3. `NO_CFL_CHECK` (flag): It is used to force to disable the CFL check. The default value is “false”. Example is as follows.

```
NO_CFL_CHECK
```

4. `REPORT_BD_FLUX` (flag): Enable the report of the accumulated net amount of species flowing across boundary for each individual boundary region (unit in lbmol) and the report of balance of species in each individual rock subdomain (unit in lbmol) from which the information of flux across interface between different rock domain might be derived. The default value is “false”. Example is as follows.

```
REPORT_BD_FLUX
```

5. `REPORT_BAL_ALL` (flag): Enable the balance report (in [lbmol]) and well output of all species. Balance and well information are reported for only the first species by default. The summed balance for all species in [lb] is always reported. The default value is “false”. Example is as follows.

```
REPORT_BAL_ALL
```

6. `NO_ADVECTION` (flag): Disable advection, which might be used for: 1) debug purpose for chemistry or dispersion; 2) cases for stationary flowing phases (limited interest though). The default value is “false”. Example is as follows.

```
NO_ADVECTION
```

7. `NO_DISPERSION` (flag): Disable dispersion/diffusion, which might be used for: 1) cases in which dispersion/diffusion is small or not interesting; 2) debug purpose. The default value is “false”. Example is as follows.

```
NO_DISPERSION
```

8. NSPECIES (integer): The number of total chemical species considered in transport-chemistry model. The default value is 1. The default value is "false". Example is as follows.

```
NSPECIES = 2
```

9. N_CONC_STEPS (integer): The number of concentration steps per flow step. The default value is 1. The default value is "false". Example is as follows.

```
N_CONC_STEPS = 1
```

10. NAME_SPECIES (array of character strings): The names of all species. The default value is ("CONC1_", "CONC2_", "CONC3_", ...). The default value is "false". Example is as follows.

```
NAME_SPECIES(1) = "PLUTONIUM"
NAME_SPECIES(2) = "IODINE"
```

11. NROCKTYPE (integer): The total number of rock types. The default value is 1. Example is as follows.

```
NROCKTYPE = 4
```

12. CONC1_, CONC2_, ... (grid array of real numbers): The initial concentration for each species. Please note the keywords CONC1_, CONC2_, ... are the default names of the species. If the user defined non-default species' name, the above keywords need to be modified accordingly. Note that the name of species must be followed by the faultblock number according to the usual rule adopted in IPARS for input of grid arrays. The default units are $[lbmol/ft^3]$. The default value is 0.0 for each entry. Example is as follows.

```
PLUTONIUM3(,,) = 0.          $ plutonium in faultblock 3
CONC2_1(,,) = 0             $ species # 2 in faultblock 1
```

7.3 Parameters for Advection/Diffusion/Dispersion

1. EQ_PARTITIONING_COEFF (2-D array of real numbers): The equilibrium partitioning coefficients (dimensionless) for each species in each phase with regard to the reference phase. The default value is 1.0 for each entry. Example is as follows.

```
EQ_PARTITIONING_COEFF(2,1) = 1.0    $ for species 1 in phase 2
```

2. MOLECULAR_DIFFUSIVITY (3-D array of real numbers): The molecular effective diffusivity for each species in each rock type and in each phase. The default unit is $[ft^2/day]$. The default value is 0.0 (no molecular diffusion). Example is as follows.

```
MOLECULAR_DIFFUSIVITY(3,2,1) = 1.474508E-4
      $ for species 2 in phase 1 and rock type 3
```

3. LONGITUDINAL_DISPERSIVITY (2-D array of real numbers): The longitudinal dispersivity for each phase in each rock type. The default unit is $[ft]$. The default value is 0.0 (no longitudinal dispersion). Example is as follows.

```
LONGITUDINAL_DISPERSIVITY(2,1) = 1.0
$ phase 1 in rock type 2
```

Note that the dispersion is not calculated by the present code.

4. TRANSVERSE_DISPERSIVITY (2-D array of real numbers): The transverse dispersivity for each phase in each rock type. The default unit is $[ft]$. The default value is 0.0 (no transverse dispersion). Example is as follows.

```
TRANSVERSE_DISPERSIVITY(2,1) = 0.2
$ phase 1 in rock type 2
```

Note that the dispersion is not calculated by the present code.

5. BOUND_TYPE_D (array of integers): The type of surface boundary condition for transport problem is set by the subscripted keyword, BOUND_TYPE_D, for which three are three possible values: i) value = 0: outflow/no-flow; ii) value = 2: Dirichlet condition. Subscript on BOUND_TYPE_D is 1..NBND_REG. The default value is 0 (i.e. outflow/no-flow) for each entry. Example is as follows.

```
BOUND_TYPE_D(1 TO 2) = 0          $ outflow/no-flow
BOUND_TYPE_D(3 TO 6) = 2          $ Dirichlet condition
```

6. TRBOUND (array of table block): The concentration as a function of time for each boundary region. The default units are $[lbmol/ft^3]$ for the concentration and $[day]$ for the time. No default value is available. Example is as follows.

```
TRBOUND1(3) Block    $ for species 1 (PLUTONIUM) in boundary 3
  Interpolation Linear
  Extrapolation Constant
  Data 0. 0.
EndBlock
```

7. TRXWELL (array of table block): It specifies the source rate of a pure transport well as a function of time for each well. (Pure transport well means the well that has source term for the transport but does not influence the flow at all.) Notice that you must declare it as a pure transport well by letting $KINDWELL(i) = -1$ so that flow part do not see the wells. The default units are $[lbmol/(ft^3 * day)]$ for the concentration and $[day]$ for the time. No default value is available. Example is as follows.

```
TRXWELL1(1) Block    $ for species 1 (PLUTONIUM) in well 1
  Interpolation Linear
  Extrapolation Constant
  Data
    3.6500000E05    1.2667984E-55
    7.8110000E05    5.7521294E-52
    8.2490000E05    9.9747909E-49
EndBlock
```

8. TMP_LSOL_TOL , TMP_LSOL_ATOL, TMP_LSOL_ABSFLAG, TMP_LSOL_ITMAX (various types). These parameters allow to specify linear solver parameters that are applied to diffusion step only. If absent, the values applied to the solver in flow step are applied. These are either default or they are specified in the input file by using flags LSOL_TOL etc...

7.4 Parameters for Chemistry

1. N_COMPONENTS (integer): The number of components. The default value is 1. Example is as follows.

```
N_COMPONENTS = 2
```

2. N_PRODUCTS (integer): The number of products. The default value is 0. Example is as follows.

```
N_PRODUCTS = 2
```

3. ODE_ALG_TYPE (character string): Choose one from "RUNGE_KUTTA_4", "RUNGE_KUTTA_2" and "EULER". The default value is "RUNGE_KUTTA_4". Example is as follows.

```
ODE_ALG_TYPE = "RUNGE_KUTTA_2"
```

4. N_RXNSTEPS (integer): The number of reaction steps per transport step. The default value is 1. Example is as follows.

```
N_RXNSTEPS = 2
```

5. MOLECULAR_WEIGHT (array of real numbers): The molecular weight for each species. The default unit is $[lb/lbM]$. The default value is 18.0 lb/lbM for each species. Example is as follows.

```
MOLECULAR_WEIGHT(1) = 242    $ for species 1 (PLUTONIUM)
```

6. USE_RADIODECAY (flag): Enables efficient implementation of first order decay chemistry in diffusion/dispersion step. The default value is "false". Example is as follows.

```
USE_RADIODECAY
```

7. HALF_LIFE_TIME (array of real numbers): The half life time for the specific first order decay chemistry. The default unit is $[day]$. The default value is 1.0D99 day for each entry (essentially for no-reaction). Example is as follows.

```
HALF_LIFE_TIME(1) = 1.3724E8 $in days
```

```
HALF_LIFE_TIME(2) = 5.7305E9 $in days
```

8. CHEMICAL_REACTION_TYPES (array of character strings): The reaction type for each product. The types of "MONOD", "EQUILIBRIUM" and "MASS_ACTION" are supported. The default value is "unknow_rxn_type". Example is as follows.

CHEMICAL_REACTION_TYPES(1) = "EQUILIBRIUM"

9. STOICH (2-D array of integers): The formula coefficients for chemical reaction for each product. The default value is 0 for each entry. Example is as follows.

STOICH(2,) = 0.0 0.0 10.87 62.5 0.0 -0.5
 \$ for product 2

10. RATE_LAW_POWERES (2-D array of integers): The rate-law powers for each component in the reaction for each product. The default value is 0 for each entry. Example is as follows.

RATE_LAW_POWERES(2,) = 0.0 0.0 0.0 0.0 0.0 1.0
 \$ for product 2

11. PK_FORWARD (2-D array of real numbers): Log 10 forward rate-constants for each reaction in each phase. The default value is 0 for each entry. Example is as follows.

PK_FORWARD(1,2) = 6.16 \$ for product 2 in phase 1

12. PK_BACKWARD (2-D array of real numbers): Log 10 backward rate-constants for each reaction in each phase. The default value is 0 for each entry. Example is as follows.

PK_BACKWARD(1,2) = -99.0 \$ for product 2 in phase 1

13. PK_EQUILIBRIUM (array of real numbers): Equilibrium constants for each reaction. No default value is available. Example is as follows.

PK_EQUILIBRIUM(1) = 1.0 \$ for product 1

14. USE_LINEAR_ADSORPTION (flag): Enable the linear equilibrium adsorption chemistry, solved by combining it in advection-diffusion/dispersion part. The default value is "false". Example is as follows.

USE_LINEAR_ADSORPTION

15. EFF_SORP_CAPACITY (2-D array of real numbers): (If flag USE_LINEAR_ADSORPTION is present) the effective adsorption capacity for each species in each rock type, defined to be the product of Henry's law constant and the adsorption capacity [dimensionless]. Notice that $dx*dy*dz*EFF_SORP_CAPACITY*conc$ means the amount (in lbmol) of the species adsorbed in the cell (dx, dy and dz are cells size). The default value is 0.0 (no adsorption). Example is as follows.

EFF_SORP_CAPACITY(2,1) = 19999.9 \$ for species 1 in rock type 2

16. PHASE_TYPE (array of character strings): The phase type for each species, which can be either "FLOWING" or "STATIONARY". The default value is "FLOWING" for each entry. Example is as follows.

PHASE_TYPE(1) = "FLOWING" \$ for species 1

17. COMP_CHARGE (array of real numbers): The intrinsic charge for each component (we treat it as dimensionless quantity). The default value is 0 for each entry. Example is as follows.

```
COMP_CHARGE(1) = 0           $ for species 1
```

18. HALF_SAT_CONST (2-D array of real numbers): The half-saturation constants in Mond kinetics for each component in each reaction. The default unit is $[lbmol/cu - ft]$. The default value is 0 for each entry. Example is as follows.

```
HALF_SAT_CONST(,2) = 0.0 0.0 5.356E-3 8.850E-5 0.0 0.0
                    $ for product 2
```

8 Implementation

The TRCHEM module is implemented *on top* of existing models in IPARSv2. In other words, it can be used in addition to another model (flow model). In this sense, it is similar to MultiModel (MMODEL) concept in IPARSv2 in which the modules of code for different models and solvers are combined in one executable. Note however, that in MMODEL the different models are assigned to different *faultblocks* whereas in TRCHEM, the flow model and the reactive transport models are executed in the same faultblocks.

Currently, the flow modules and reactive transport modules are executed sequentially in time. The solution obtained in the flow step is postprocessed for the needs of TRCHEM and TRCHEM-specific modules are called. However, there is no feedback from TRCHEM to flow model.

Most variables are specific to either the flow model or to the TRCHEM model. Some variables used for postprocessing are shared by the flow models and by the TRCHEM model and their pointers are kept in header files available to both flow model and TRCHEM code.

8.1 Memory management and variables

Concentration variables are visible in two ways by the framework. First, they are members of the array CONC of pointer N_CONC which holds all species together. Secondly, they are available through individual pointers or members N_CONCARR. This duality allows to make all species available at once to chemistry routines and individual species available to advection routines which speeds up memory access.

8.2 Flow model postprocessing

The main postprocessing routines are called by the flow model only if TRCHEM model is active (this is achieved with the use of keyword \$TRCHEM).

For example in HYDROI model, the water saturation and the Darcy's velocities for both water and oil phases are computed and made available to the TRCHEM model. For each component, a phase-summed capacity Φ_{α}^* is computed using water saturation and the phase partitioning coefficients. Likewise, a phase-summed velocity is computed.

In addition, boundary fluxes and well rates are computed. These computations are done extra for the needs of TRCHEM model as it needs volumetric rather than mass quantities which are computed by flow model(s) by default.

In some cases, the quantities delivered by the flow model are specific to a time step (flow time step). In other cases, they are averaged over time interval between the (flow) time steps.

8.3 Implementation of Godunov method

Consider a fully discrete counterpart of equation (24). Suppress subscripts α, w and superscript $*$ and use subscript ijk to denote grid cell (gridblock). Note: in the code, concentrations are numbered $CONC(i, j, k) = c_{ijk}, i = 1, 2, \dots, I$. Each component of velocity $\mathbf{u} = (u^x, u^y, u^z)$ is a separate grid array VELX, VELY, VELZ. Velocities are numbered by edges. For example, $VELX(i, j, k) \equiv u_{i-1/2, jk}^x, i = 1, 2, \dots, I + 1$.

Godunov procedure for solution of (24) consists of three steps:

1. Compute $T_{ijk}^m = \Phi_{ijk}^m c_{ijk}^m$

2. Advect: solve

$$\frac{\bar{T}_{ijk} - T_{ijk}^m}{\Delta t} + \nabla \cdot (\mathbf{u}c^m)_{ijk} = q^T \quad (35)$$

In fact, this means explicitly

$$\bar{T}_{ijk} = T_{ijk}^m + \Delta t q^T - \frac{\Delta t}{\Delta x_{i+1/2}} u_{i+1/2, jk}^x c_{i+1/2, jk}^m + \frac{\Delta t}{\Delta x_{i-1/2}} u_{i-1/2, jk}^x c_{i-1/2, jk}^m + \quad (36)$$

$$\text{similar terms in yz or jk direction} \quad (37)$$

where $c_{i+/-1/2, jk}^m$ is the up-winded value defined as

$$c_{i+1/2, jk}^m = \begin{cases} c_{ijk}^m & u_{i+1/2, jk}^x \geq 0 \\ c_{i+1, jk}^m & u_{i+1/2, jk}^x < 0 \end{cases}$$

3. Compute

$$\bar{c}_{ijk} = \frac{\bar{T}_{ijk}}{\Phi_{ijk}^m}$$

Note: in the implementation, Equation (35) is solved in conservative form. In other words, it corresponds to the Equation (24) after integration over cell ijk . In particular, the cell volumes $V_{ijk} = \Delta x_i \Delta y_j \Delta z_k$ are lumped in the computation of Φ_{ijk}^m . Also, the areas of faces (edges) are lumped in the computation of volumetric Darcy's velocities. For example, $\Delta y_j \Delta z_k$ enters $u_{i-1/2, jk}^x$.

8.4 Developer's notes: TRCHEM wells

Note that framework and flow models assume MXHISD types of input/output for every well. This quantity is increased in `trchem.siz` by $2 * \text{MXNSPEC}$ to allow input of concentration and output of rates for both injection and production wells. The structure of the preprocessor requires then that `trchem.siz` be included after flow model specific files are included. It is assumed that well output of type 1..MXFHIS may be used by the flow models and that MXFHIS+1...MXHISD may be used for inj./prod. of species.

The table number for well per species is numbered as `NTRXTAB(numwell, nspec)`. If well data is not specified for some species, it is considered zero and all values expected from this well data are set to zero.

9 Appendix

9.1 The molecular diffusion tensor

One of three contributions to the mass flux density J_{api} comes from molecular diffusion.

$$J_{api}^{\text{diff}} = -\Phi S_p \sum_j D_{apij}^{\text{diff}} \frac{\partial c_{ap}}{\partial x_j}$$

The tensor D_{apij}^{diff} is usually assumed to be diagonal and it can be calculated by,

$$D_{apij}^{\text{diff}} = \tau_p d_{mol,ap} \delta_{ij}$$

Where τ_p is the tortuosity coefficient for p phase. A theoretical analysis [8] suggests the tortuosity coefficient τ_p to be of the form

$$\tau_p = \Phi^{1/3} S_p^{7/3}$$

The molecular diffusion coefficient $d_{mol,ap}$ depends on the concentrations in general. It is a weak function in many cases, thus we assume $d_{mol,ap}$ to be a constant, which depends only the type of phase and type of species.

9.2 The mechanical dispersion tensor

9.2.1 Derivation for single flowing phase

For the convenience of expression, we use Einstein summation notation in this subsection. From the theoretical analysis and experimental study [6], it is found that, for single flowing phase, the dispersion tensor D_{ij} can be computed by

$$D_{ij} = a_{ijkl} \frac{v_k v_l}{\|v\|} f(Pe_\alpha, r) \quad (38)$$

where, a_{ijkl} is the dispersivity to be determined; v is the pore velocity (Darcy's velocity over porosity) and v_k is its components in the direction x_k ; the parameters $r = l/\Delta$ represents the ratio between characteristic lengths, in the direction of the flow and normal to it, within a pore; $Pe_\alpha = \frac{V}{D_\alpha/\Delta}$ expresses the ratio between the rates of transport by advection and by diffusion; and $f(Pe_\alpha, r) = \frac{Pe_\alpha}{Pe_\alpha + 1 + r}$; the norm $\|\cdot\|$ is the usual Euclidean norm. Fortunately, for most cases occurred in practice, $Pe_\alpha \gg 1$, so $f(Pe_\alpha, r) \approx 1$, so we assume $f = 1$ in our model.

The dispersivity a_{ijkl} is a fourth rank tensor, has 81 components in a three-dimensional space. For non-isotropic porous media the form of the dispersivity tensor a_{ijkl} , thus the form of dispersion tensor D_{ij} , becomes very complicated and the number of independent parameters becomes so great that calibration is infeasible. Therefore dispersion is usually assumed to be isotropic, although this may not be strictly so, see [9]. One can show by pure mathematics that the general isotropic fourth rank tensor Q_{ijkl} must have following form.

$$Q_{ijkl} = Q_1 \delta_{ij} \delta_{kl} + Q_2 \delta_{ik} \delta_{jl} + Q_3 \delta_{il} \delta_{jk}$$

Thus our isotropic dispersivity tensor a_{ijkl} must have the form $a_{ijkl} = a_1 \delta_{ij} \delta_{kl} + a_2 \delta_{ik} \delta_{jl} + a_3 \delta_{il} \delta_{jk}$. Substitute a_{ijkl} into the equation ((38)) (let $f = 1$), we have

$$D_{ij} = a_1 \delta_{ij} \|v\| + (a_2 + a_3) \frac{v_i v_j}{\|v\|} \quad (39)$$

Consider the special case in which the coordinate system is oriented with the flow field in the x_1 direction such that $v_1 = v$ and $v_2 = v_3 = 0$. Then above equation for dispersion tensor becomes

$$D = \begin{bmatrix} (a_1 + a_2 + a_3)v & 0 & 0 \\ 0 & a_1 v & 0 \\ 0 & 0 & a_1 v \end{bmatrix}$$

Above form give us the physical meaning that $(a_1 + a_2 + a_3)v$ is the dispersion coefficient in the flow direction and $(a_1 v)$ transverse to the direction of flow. Thus we define $d_{long} = (a_1 + a_2 + a_3)$ and $d_{trans} = a_1$. The equation ((39)) becomes

$$D_{ij} = d_{trans} \|v\| \delta_{ij} + (d_{long} - d_{trans}) \frac{v_i v_j}{\|v\|}$$

Substituting $v = u/\Phi$ into above equation (u is the Darcy's velocity and v is the pore velocity), we have,

$$\Phi D_{ij} = d_{trans} \|u\| \delta_{ij} + (d_{long} - d_{trans}) \frac{u_i u_j}{\|u\|}$$

9.2.2 Derivation for multiple flowing phases

Now let's consider the multiple flowing phases. We define the pore velocity for phase p as $v_p = \frac{u_p}{S_p \Phi}$, where u_p and S_p are the Darcy's velocity and the saturation for phase p , respectively. Using the argument similar as above, except we ignore the interaction between phases, we can have the expression for the dispersion tensor D_{pij}

$$D_{pij} = d_{trans} \|v_p\| \delta_{ij} + (d_{long} - d_{trans}) \frac{v_{pi} v_{pj}}{\|v_p\|}$$

Substituting $v_p = u_p / (S_p \Phi)$ into above equation, we have,

$$\Phi S_p D_{pij} = d_{trans} \|u_p\| \delta_{ij} + (d_{long} - d_{trans}) \frac{u_{pi} u_{pj}}{\|u_p\|}$$

The parameters d_{trans} and d_{long} depends on the microscopic configuration of the phase p within REV (Representative Elementary Volume) [6]. If the interaction between flowing phases is completely ignored and the flowing behavior of each phase is assumed to be similar, the parameters d_{trans} and d_{long} will be expected to be the same for all flowing phases. However, the interaction between flowing phases exists in the real world and the microscopic configuration of wetting phase will be quite different from the microscopic configuration of non-wetting phase, the parameters d_{trans} and d_{long} should be phase-dependent. To make our model more generic, we consider d_{trans} and d_{long} differently for each flowing phase p , denote them as $d_{trans,p}$ and $d_{long,p}$, although d_{trans} and d_{long} are considered to be same values for all phases in some literature (such as [9]). We like to point out that this won't significantly increase computational and memory cost anyway because the number of flowing phases usually is only 2 or 3.

9.2.3 The longitudinal dispersivity d_{long} and the transverse dispersivity d_{trans}

Both of the longitudinal dispersivity and the transverse dispersivity have a length dimension, and both are related to the length l that characterizes the microscopic configuration of the phase within the REV. It is shown that longitudinal dispersivity d_{long} should be the order of magnitude of a pore size and d_{trans} is 8 to 24 times smaller than d_{long} [6]. It is pointed out that both dispersivities are often observed to exhibit apparent scale dependence, the longitudinal dispersivity typically is found to range from 0.01 to 0.1 of the mean travel distance for distances less than 1 km, and the transverse dispersivity is typically in the range of 0.1 to 0.3 times the longitudinal dispersivity [9]. Like the porosity, the longitudinal dispersivity d_{long} and the transverse dispersivity d_{trans} is function of rock type and could be heterogeneous when the rock properties is heterogeneous. Thus in this model, we consider d_{long} and d_{trans} to be grid arrays, which might vary with position, but are constant with time. For multiple flowing phases, we consider $d_{long,p}$ and $d_{trans,p}$ to be grid arrays, which might vary with type of phase and with position, but are constant with time.

9.3 Alternative for overall algorithm

We have one alternative for the overall algorithm. The advantage for this algorithm is that the resulting linear system in the dispersion/diffusion subproblem is symmetric, and thus the preconditioned conjugate gradient method can be used. The disadvantage is that we have assumed the flowing phases is incompressible in this algorithm. Because we want also to solve problem for compressible flowing phases, we do not use this algorithm.

In this algorithm, we solve for $c_{\alpha w}$ in each splitting step.

Recall that the phase-summed transport equation for species α and the flow equation are as follows respectively,

$$\frac{\partial \Phi_{\alpha}^* c_{\alpha w}}{\partial t} + \sum_i \frac{\partial u_{\alpha i}^* c_{\alpha w}}{\partial x_i} - \sum_i \frac{\partial}{\partial x_i} \left(\sum_j D_{\alpha ij}^* \frac{\partial c_{\alpha w}}{\partial x_j} \right) = R_{\alpha}^{TC} + R_{\alpha}^{TN} + q_{\alpha}^T$$

$$\Phi \frac{\partial \rho_p S_p}{\partial t} + \sum_i \frac{\partial \rho_p u_{pi}}{\partial x_i} = \gamma_p$$

Assuming that the density derivative terms to be of second-order importance yields

$$\begin{aligned} \frac{\partial \Phi_{\alpha}^*}{\partial t} + \sum_i \frac{\partial u_{\alpha i}^*}{\partial x_i} &= \sum_p \Gamma_{\alpha p} \left(\Phi \frac{\partial S_p}{\partial t} + \sum_i \frac{\partial u_{pi}}{\partial x_i} \right) \\ &= \sum_p \Gamma_{\alpha p} \left(\frac{\gamma_p}{\rho_p} \right) \end{aligned}$$

If we define

$$\gamma_{\alpha}^* = \sum_p \Gamma_{\alpha p} \left(\frac{\gamma_p}{\rho_p} \right)$$

The phase-summed transport equation becomes,

$$\Phi_{\alpha}^* \frac{\partial c_{\alpha w}}{\partial t} + \sum_i u_{\alpha i}^* \frac{\partial c_{\alpha w}}{\partial x_i} - \sum_i \frac{\partial}{\partial x_i} \left(\sum_j D_{\alpha ij}^* \frac{\partial c_{\alpha w}}{\partial x_j} \right) + \gamma_{\alpha}^* c_{\alpha w} = R_{\alpha}^{TC} + R_{\alpha}^{TN} + q_{\alpha}^T$$

where the term $\gamma_\alpha^* c_{\alpha w}$ comes from the total interphase mass transfer.

Where we can get the unknown γ_p ? By the nature of physics, γ_p should be computed from p coupled transport equations (one for each phase) and γ_p will be given to the flow equation. But that mean we have to solve the p coupled transport equation instead of one single phase-summed equation. But in many real environment problem, γ_p is negligible such as in hydrology model and air-water model. In black-oil model, we can get γ_p (non-zero for gas and oil phases) from the black-oil model, which contain a simplified transport equations.

9.3.1 Advection

We solve the equation

$$\Phi_\alpha^* \frac{\partial c_{\alpha w}}{\partial t} + \sum_i u_{\alpha i}^* \frac{\partial c_{\alpha w}}{\partial x_i} = q_\alpha^T$$

Using the Godunov Methods, we solve for $c_{\alpha w}$ by

$$\Phi_\alpha^* \frac{\bar{c}_{\alpha w} - c_{\alpha w}^m}{\Delta t} + \sum_i u_{\alpha i}^* \frac{\partial c_{\alpha w}^m}{\partial x_i} = q_\alpha^T$$

9.3.2 Radionuclide decay

We solve the equation

$$\Phi_\alpha^* \frac{\partial c_{\alpha w}}{\partial t} = R_\alpha^{TN}$$

explicitly by

$$\Phi_\alpha^* \frac{\tilde{c}_{\alpha w} - \bar{c}_{\alpha w}}{\Delta t} = R_\alpha^{TN}$$

9.3.3 Chemical Reaction

We solve the equation

$$\Phi_\alpha^* \frac{\partial c_{\alpha w}}{\partial t} = R_\alpha^{TC} - \gamma_\alpha^* c_{\alpha w}$$

explicitly by

$$\Phi_\alpha^* \frac{\hat{c}_{\alpha w} - \tilde{c}_{\alpha w}}{\Delta t} = R_\alpha^{TC} - \gamma_\alpha^* \tilde{c}_{\alpha w}$$

9.3.4 Diffusion/Dispersion

We solve the equation

$$\Phi_\alpha^* \frac{\partial c_{\alpha w}}{\partial t} - \sum_i \frac{\partial}{\partial x_i} \left(\sum_j D_{\alpha ij}^* \frac{\partial c_{\alpha w}}{\partial x_j} \right) = 0$$

implicitly by

$$\frac{c_{\alpha w}^{m+1} - \hat{c}_{\alpha w}}{\Delta t} - \sum_i \frac{\partial}{\partial x_i} \left(\sum_j D_{\alpha ij}^* \frac{\partial c_{\alpha w}^{m+1}}{\partial x_j} \right) = 0$$

The cell-centered finite difference is used. Fortunately, the resulting linear system is symmetric. Thus the preconditioned conjugate gradient method will be used as in ParSSim [4] to solve the equation.

9.4 Phase-summed transport equations based on total concentration

In many interesting physical cases, some species will have very large or very small partitioning coefficients. This will result in numerical difficulty if we choose the wrong reference phase. There are two methods to solve this problem. One method is to choose individual reference phase for each individual species. For example, in water-air system, for species of oxygen, we might want to choose the air as the reference phase, but for the species of salt, we might want to choose the water as the reference phase. One of the disadvantage for this method is that we need more book-keeping routines.

Another method is the method of total concentration, which might be used in the future version of this model. The total concentration C_α for species α is defined as

$$C_\alpha = \sum_p S_p c_{\alpha p}$$

Substituting the partitioning relationship into above equation, we have,

$$C_\alpha = \sum_p S_p \Gamma_{\alpha p} c_{\alpha r}$$

Thus, the concentration in each phase can be calculated by following flash equation.

$$c_{\alpha p} = \frac{\Gamma_{\alpha p}}{\sum_q S_q \Gamma_{\alpha q}} C_\alpha$$

The phase-summed transport equation for species α in flowing phases will become,

$$\frac{\partial \Phi C_\alpha}{\partial t} + \sum_i \frac{\partial u_{\alpha i}^* C_\alpha}{\partial x_i} - \sum_i \frac{\partial}{\partial x_i} \left(\sum_j D_{\alpha ij}^* \frac{\partial C_\alpha}{\partial x_j} \right) = R_\alpha^{TC} + R_\alpha^{TN} + q_\alpha^T$$

where,

$$\begin{aligned} \Phi &= (\text{porosity}) \\ u_{\alpha i}^* &= \frac{\sum_p \Gamma_{\alpha p} u_{pi}}{\sum_p S_p \Gamma_{\alpha p}} \\ D_{\alpha ij}^* &= \frac{\Phi \sum_p S_p \Gamma_{\alpha p} D_{\alpha pij}}{\sum_p S_p \Gamma_{\alpha p}} = \left(\frac{\Phi \sum_p \tau_p S_p \Gamma_{\alpha p} d_{mol, \alpha p}}{\sum_p S_p \Gamma_{\alpha p}} + \frac{\sum_p d_{trans, p} \|u_p\| \Gamma_{\alpha p}}{\sum_p S_p \Gamma_{\alpha p}} \right) \delta_{ij} \\ &\quad + \frac{1}{\sum_p S_p \Gamma_{\alpha p}} \sum_p \frac{(d_{long, p} - d_{trans, p}) u_{pi} u_{pj} \Gamma_{\alpha p}}{\|u_p\|} \\ R_\alpha^{TC} &= \Phi \sum_p S_p R_{\alpha p}^C \\ R_\alpha^{TN} &= \Phi \sum_p S_p R_{\alpha p}^N \\ q_\alpha^T &= \sum_p q_{\alpha p} \end{aligned}$$

and R_α^{TC} and R_α^{TN} will be the function of $c_{\alpha w}$, $\alpha = 1, 2, \dots$ and S_p , $p = 1, 2, \dots$. Of course, R_α^{TC} and R_α^{TN} will also depends on Φ and $\Gamma_{\alpha p}$, which is considered to be constant with time in this work. (Φ might vary slightly with time if the rock is slightly compressible)

References

- [1] *IPARS: Integrated Parallel Accurate Reservoir Simulator, CVS distribution of IPARSv2*. Center for Subsurface Modeling, TICAM, University of Texas at Austin.
- [2] *IPARS user's manual (draft)*. CSM, TICAM, UT-Austin, Oct 8, 2000
- [3] *IPARS: Integrated Parallel Accurate Reservoir Simulator, CVS distribution of IPARSv2/doc/wells.txt*. Center for Subsurface Modeling, TICAM, University of Texas at Austin.
- [4] T. Arbogast. *User's Guide to ParSSim1: The Parallel Subsurface Simulator, Single Phase*, TICAM report 98-13, 1998
- [5] T. Arbogast, C. N. Dawson and M. F. Wheeler. A parallel algorithm for two phase multicomponent contaminant transport. *Applications of mathematics*, 40(3): 163-174, 1995.
- [6] J. Bear and Y. Bachmat. *Introduction to modeling of transport phenomena in porous media*. Kluwer Academic Publishers, The Netherlands, 1990.
- [7] Qin Lu, Malgorzata Peszynska and Xiuli Gai, *Implicit Black-oil Model in IPARS Framework*. TICAM report 01-33, Texas Institute for Computational and Applied Mathematics, The University of Texas at Austin, Austin, TX, 2001.
- [8] R. J. Millington and J. P. Quirk. Permeability of porous media. *Nature London*, 183: 387-388, 1959
- [9] J. C. Parker. Multiphase flow and transport in porous media. *Reviews of Geophysics*, 27(3): 311-328, 1989.
- [10] M. Parashar, J. A. Wheeler, G. Pope, K. Wang, and P. Wang. A new generation EOS compositional reservoir simulator. part II: Framework and multiprocessing. In *Fourteenth SPE Symposium on Reservoir Simulation, Dalas, Texas*, pages 31-38. Society of Petroleum Engineers, June 1997.
- [11] D. W. Peaceman. *Fundamentals of numerical reservoir simulation*. Elsevier Scientific Publishing Company, Amsterdam-Oxford-New York, first edition, 1977.
- [12] Malgorzata Peszynska, Eleanor Jenkins, and Mary F. Wheeler. Boundary conditions for fully implicit two-phase flow model. submitted.
- [13] Malgorzata Peszynska, Steven Bryant, Shuyu Sun, Tara LaForce, and Shawn Snider. Modeling of Couplex1 case with IPARS TRCHEM module. TICAM report 01-31, Texas Institute for Computational and Applied Mathematics, The University of Texas at Austin, Austin, TX, 2001.
- [14] T. F. Russell and M. F. Wheeler. Finite element and finite difference methods for continuous flows in porous media. In R. E. Ewing, editor, *The Mathematics of Reservoir Simulation*, pages 35-106. SIAM, Philadelphia, 1983.
- [15] P. Wang, I. Yotov, M. F. Wheeler, T. Arbogast, C. N. Dawson, M. Parashar, and K. Sepehrnoori. A new generation EOS compositional reservoir simulator. part I: Formulation and discretization. In *Fourteenth SPE Symposium on Reservoir Simulation, Dalas, Texas*, pages 55-64. Society of Petroleum Engineers, June 1997.