Solving Partial Differential Algebraic Equations and reactive transport models

Jocelyne Erhel
SAGE team, INRIA, RENNES

co-authors
Souhila Sabit (SAGE team, INRIA, Rennes, France)
Caroline de Dieuleveult (Mines ParisTech, Fontainebleau, France)

Pareng, Pecs, Hungary, March 2013
1 Introduction
1 Introduction
2 Physical model
1 Introduction
2 Physical model
3 DAE global approach
4 Numerical experiment
5 Conclusion
Water resources

Pictures: Yves Chaux, Rennes, France
Diagram: http://www.ec.gc.ca/water/f_main.html
Reactive transport modeling

- Coupling transport by advection-dispersion with geochemistry
- System of Partial Differential Algebraic equations
- Model with thermodynamic equilibrium
- Method of lines: first discretize in space then in time ⇒ DAE system
- Explicit scheme (SNIA): decoupling but stability restrictions
- Implicit scheme (Global): stability but nonlinear coupled system

Our method: global approach GDAE


- C. de Dieuleveult, J. Erhel, M. Kern; A global strategy for solving reactive transport equations; Journal of Computational Physics, 2009.
Mass action laws

Aqueous reactions

\[ x_i(c) = K_{ci} \prod_{j=1}^{N_c} c_j^{S_{ij}}, \quad i = 1, \ldots N_x \]  

(1)
Mass action laws

Aqueous reactions

\[ x_i(c) = K_{ci} \prod_{j=1}^{N_c} c_j^{S_{ij}}, \quad i = 1, \ldots N_x \]  

(1)

Sorption reactions

\[ y_i(c, s) = K_{si} \prod_{j=1}^{N_c} c_j^{A_{ij}} \prod_{j=1}^{N_s} s_j^{B_{ij}}, \quad i = 1, \ldots N_y, \]  

(2)
Mass action laws

**Aqueous reactions**

$$x_i(c) = K_{ci} \prod_{j=1}^{N_c} c_j^{S_{ij}}, \quad i = 1, \ldots N_x$$  \hspace{1cm} (1)

**Sorption reactions**

$$y_i(c, s) = K_{si} \prod_{j=1}^{N_c} c_j^{A_{ij}} \prod_{j=1}^{N_s} s_j^{B_{ij}}, \quad i = 1, \ldots N_y,$$  \hspace{1cm} (2)

**Precipitation reactions**

$$\Pi_i(c) = K_{pi} \prod_{j=1}^{N_c} c_j^{E_{ij}}, \quad i = 1, \ldots N_p$$  \hspace{1cm} (3)
Mass conservation laws

Chemical variables and functions

\[ X = \begin{pmatrix} c \\ s \\ p \end{pmatrix}, \quad \Phi(X) = \begin{pmatrix} c + S^T x(c) + A^T y(c, s) + E^T p \\ s + B^T y(c, s) \\ \Pi(c) \end{pmatrix} \] (4)
Mass conservation laws

Chemical variables and functions

\[ X = \begin{pmatrix} c \\ s \\ p \end{pmatrix}, \quad \Phi(X) = \begin{pmatrix} c + S^T x(c) + A^T y(c, s) + E^T p \\ s + B^T y(c, s) \\ \Pi(c) \end{pmatrix} \tag{4} \]

Chemical model

\[ \begin{cases} 
\Phi(X) = \begin{pmatrix} T \\ W \\ 1 \end{pmatrix}, \\
\begin{align*}
c &\geq 0, \\
s &\geq 0, \\
p &> 0.
\end{align*} 
\end{cases} \tag{5} \]
Transport model

Advection-Dispersion operator

\[ \mathcal{L}(u) = \nabla \cdot (vu - D \nabla u) \]

\[ D = d_m I + \alpha_T \|v\| I + (\alpha_L - \alpha_T) \frac{vv^T}{\|v\|} \]
Transport model

Advection-Dispersion operator

\[ \mathcal{L}(u) = \nabla \cdot (\nu u - D \nabla u) \]

\[ D = d_m I + \alpha_T \|v\| I + (\alpha_L - \alpha_T) \frac{vv^T}{\|v\|} \]

Transport of mobile species

\[ C(X) = c + S^T x(c) \quad (6) \]

\[ \omega \frac{\partial T_i}{\partial t} + \mathcal{L}(C_i) = Q_i, \quad i = 1, \ldots, N_c \quad (7) \]

with boundary and initial conditions
Transport model

Advection-Dispersion operator

\[ \mathcal{L}(u) = \nabla \cdot (\nu u - D \nabla u) \]

\[ D = d_m I + \alpha_T \|v\| I + (\alpha_L - \alpha_T) \frac{vv^T}{\|v\|} \]

Transport of mobile species

\[ C(X) = c + S^T x(c) \]  \hspace{1cm} (6)

\[ \omega \frac{\partial T_i}{\partial t} + \mathcal{L}(C_i) = Q_i, \quad i = 1, \ldots, N_c \]  \hspace{1cm} (7)

with boundary and initial conditions

Space discretization

with (for example) a finite difference method

\[ T = (T_1, \ldots, T_{N_m}) \]  \hspace{1cm} (8)
Coupling transport with chemistry

Semi-discrete reactive transport model

\[
\begin{align*}
\omega \frac{dT_i}{dt} + LC_i(X) &= Q_i + G_i, \quad i = 1, \ldots, N_c, \\
\Phi(X_j) - \begin{pmatrix} T_j \\ W_j \\ 1 \end{pmatrix} &= 0 \quad j = 1, \ldots, N_m, \\
\text{initial condition for } T,
\end{align*}
\]

(9)
Coupling transport with chemistry

Semi-discrete reactive transport model

\[
\begin{align*}
\omega \frac{dT_i}{dt} + LC_i(X) &= Q_i + G_i, \quad i = 1, \ldots, N_c, \\
\Phi(X_j) - \begin{pmatrix} T_j \\ W_j \\ 1 \end{pmatrix} &= 0 \quad j = 1, \ldots, N_m, \\
\text{initial condition for } T,
\end{align*}
\] (9)

DAE formulation

\[
\begin{align*}
\omega \frac{d\text{vec } T}{dt} + (L \otimes I)\text{vec } C(X) - \text{vec } Q - \text{vec } G &= 0, \quad i = 1, \ldots, N_c, \\
\text{vec } \Phi(X) - (I \otimes N)\text{vec } T - \text{vec } F &= 0
\end{align*}
\] (10)
DAE Global approach with substitution

Time discretization: BDF scheme

\[
\frac{d\text{vec } T}{dt} \simeq \frac{a}{\Delta t} \text{vec } T + \frac{1}{\Delta t} \text{vec } Z,
\]

\[
\begin{aligned}
\frac{a\omega}{\Delta t} \text{vec } T + (L \otimes I)\text{vec } C(X) - ... &= 0, \\
-(I \otimes N)\text{vec } T + \text{vec } \Phi(X) - ... &= 0,
\end{aligned}
\]
**DAE Global approach with substitution**

**Time discretization: BDF scheme**

\[
\frac{d}{dt} \text{vec } T \approx \frac{a}{\Delta t} \text{vec } T + \frac{1}{\Delta t} \text{vec } Z, \\
\left\{\begin{array}{l}
\frac{a\omega}{\Delta t} \text{vec } T + (L \otimes I) \text{vec } C(X) - \ldots = 0, \\
-(I \otimes N) \text{vec } T + \text{vec } \Phi(X) - \ldots = 0,
\end{array}\right.
\]

**Substitution**

\[
R(X) = \frac{\Delta t}{a\omega} (L \otimes N) \text{vec } C(X) + \text{vec } \Phi(X) - \ldots
\]

The Jacobian of \( R \) is

\[
J_R(X) = \frac{\Delta t}{a\omega} (L \otimes N) \text{diag} \left( \frac{dC}{dX}(X_j) \right) + \text{diag}(J\Phi(X_j)).
\]
**DAE Global approach with substitution**

**Time discretization: BDF scheme**

\[
\frac{d \text{vec } T}{dt} \approx \frac{a}{\Delta t} \text{vec } T + \frac{1}{\Delta t} \text{vec } Z,
\]

\[
\begin{align*}
\frac{a \omega}{\Delta t} \text{vec } T + (L \otimes I) \text{vec } C(X) - \ldots &= 0, \\
-(I \otimes N) \text{vec } T + \text{vec } \Phi(X) - \ldots &= 0,
\end{align*}
\]

**Substitution**

\[
R(X) = \frac{\Delta t}{a \omega} (L \otimes N) \text{vec } C(X) + \text{vec } \Phi(X) - \ldots
\]

The Jacobian of \( R \) is

\[
J_R(X) = \frac{\Delta t}{a \omega} (L \otimes N) \text{diag} \left( \frac{dC}{dX}(X_j) \right) + \text{diag}(J_\Phi(X_j)).
\]

**Nonlinear system**

\[
R(X) = 0
\]

solved with Newton method
GRT3D software

Transport modules

- The velocity $\nu$ is computed with MODFLOW
- The transport operator $L$ is computed with MT3D
GRT3D software

Transport modules

- The velocity $v$ is computed with MODFLOW
- The transport operator $L$ is computed with MT3D

Chemistry modules

- The functions $\Phi(X)$ and $C(X)$
- The derivatives $J_\Phi(X)$ and $dC(X)/dX$
## GRT3D software

### Transport modules
- The velocity $\nu$ is computed with MODFLOW
- The transport operator $L$ is computed with MT3D

### Chemistry modules
- The functions $\Phi(X)$ and $C(X)$
- The derivatives $J_\Phi(X)$ and $dC(X)/dX$

### Coupling modules
- The function $R(X)$
- The derivative $J_R(X)$
- The DAE solver IDA in SUNDIALS using Newton-LU method
- The sparse linear solver UMFPACK
Versions of GRT3D

First version with logarithms
No elimination of $T$ and $C$
Versions of GRT3D

First version with logarithms
No elimination of $T$ and $C$

Reduced version with logarithms
Elimination of $T$ and $C$ in the linearized equations
Versions of GRT3D

First version with logarithms
No elimination of $T$ and $C$

Reduced version with logarithms
Elimination of $T$ and $C$ in the linearized equations

Optimized version without logarithms
Elimination of $T$ and $C$ in the linearized equations
Andra qualification test

Injection of alcaline water $NaOH$ in a porous medium with quartz $SiO_2$

\[
\begin{align*}
\epsilon &= 1. \\
\nu &= \begin{pmatrix} 5.7 \times 10^{-7} \\ 0. \end{pmatrix} \text{ m.s}^{-1} \\
\alpha_L &= 0.2 \text{ m} \\
\alpha_T &= 0.05 \text{ m} \\
T &= 30 \text{ days} \\
\text{no flux boundary conditions}
\end{align*}
\]
Chemistry conditions

Chemistry equations

\[ \begin{align*}
H_2O & \leftrightarrow H^+ + OH^- & K_1 = 10^{-14} \\
H_4SiO_4 & \leftrightarrow H_3SiO_4^- + H^+ & K_2 = 10^{-9.8} \\
H_4SiO_4 & \leftrightarrow SiO_2(s) + 2H_2O & K_3 = 10^{3.6}
\end{align*} \] (11)
Chemistry conditions

Chemistry equations

\[
\begin{align*}
H_2O & \leftrightarrow H^+ + OH^- & K_1 = 10^{-14} \\
H_4SiO_4 & \leftrightarrow H_3SiO_4^- + H^+ & K_2 = 10^{-9.8} \\
H_4SiO_4 & \leftrightarrow SiO_2(s) + 2H_2O & K_3 = 10^{3.6}
\end{align*}
\]

(11)

Stoichiometric coefficients

<table>
<thead>
<tr>
<th></th>
<th>(Na^+)</th>
<th>(OH^-)</th>
<th>(H_4SiO_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(H^+)</td>
<td>0</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>(H_3SiO_4^-)</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SiO2</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Chemistry conditions

Chemistry equations

\[ H_2O \quad \leftrightarrow \quad H^+ + OH^- \quad K_1 = 10^{-14} \]
\[ H_4SiO_4 \quad \leftrightarrow \quad H_3SiO_4^- + H^+ \quad K_2 = 10^{-9.8} \]
\[ H_4SiO_4 \quad \leftrightarrow \quad SiO_2(s) + 2H_2O \quad K_3 = 10^{3.6} \] (11)

Stoichiometric coefficients

<table>
<thead>
<tr>
<th></th>
<th>Na$^+$</th>
<th>OH$^-$</th>
<th>H4SiO4</th>
</tr>
</thead>
<tbody>
<tr>
<td>H$^+$</td>
<td>0</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>H3SiO4$^-$</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SiO2</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Initial values

<table>
<thead>
<tr>
<th></th>
<th>Na$^+$</th>
<th>OH$^-$</th>
<th>SiO2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outside $M$</td>
<td>0</td>
<td>0</td>
<td>10.</td>
</tr>
<tr>
<td>At $M$</td>
<td>$10^{-2}$</td>
<td>$10^{-2}$</td>
<td>10.</td>
</tr>
</tbody>
</table>
## Accuracy results

### Accuracy of computed pH

\[ E_{H^+} = \| \tilde{x}_{H^+} - x_{H^+} \| \]

<table>
<thead>
<tr>
<th>Mesh</th>
<th>first GRT3D</th>
<th>reduced GRT3D</th>
<th>optimized GRT3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>21x14</td>
<td>1.333005E-11</td>
<td>1.591450E-11</td>
<td>8.040057E-11</td>
</tr>
<tr>
<td>41x28</td>
<td>2.489791E-09</td>
<td>2.489787E-09</td>
<td>8.113751E-11</td>
</tr>
<tr>
<td>81x56</td>
<td>7.640456E-09</td>
<td>7.640825E-09</td>
<td>3.055914E-10</td>
</tr>
<tr>
<td>71x101</td>
<td>7.747011E-09</td>
<td>7.746415E-09</td>
<td>4.161827E-10</td>
</tr>
<tr>
<td>161x112</td>
<td>7.9736E-09</td>
<td>7.9738E-09</td>
<td>2.6672E-10</td>
</tr>
<tr>
<td>322x224</td>
<td>-</td>
<td>3.0871E-09</td>
<td>4.3067E-10</td>
</tr>
</tbody>
</table>
### CPU time of GRT3D

Computations done on two six-core Intel Xeon processors X5690, with a frequency of 3.46 GHz and 4GB of RAM per core.

<table>
<thead>
<tr>
<th>Mesh</th>
<th>first GRT3D</th>
<th>reduced GRT3D</th>
<th>optimized GRT3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>21x14</td>
<td>18 s</td>
<td>4 s</td>
<td>2 s</td>
</tr>
<tr>
<td>48x28</td>
<td>1 min 36 s</td>
<td>21 s</td>
<td>8s</td>
</tr>
<tr>
<td>81x56</td>
<td>6 min 33 s</td>
<td>1 min 53 s</td>
<td>50 s</td>
</tr>
<tr>
<td>71x101</td>
<td>11 min 55 s</td>
<td>3 min 28 s</td>
<td>1 min 21 s</td>
</tr>
<tr>
<td>161x112</td>
<td>32 min 43 s</td>
<td>16 min 30 s</td>
<td>4 min 32 s</td>
</tr>
<tr>
<td>322x224</td>
<td>-</td>
<td>1 h 52 min</td>
<td>37 min 38 s</td>
</tr>
</tbody>
</table>
Comparison results

Performance/Accuracy results

Comparison with an analytical solution

\[ E_{Na^+} = \left[ \frac{1}{N_m N_t} \sum_{n,j} (\tilde{x}_{Na^+}(m_j, t_n) - x_{Na^+}(m_j, t_n)) \right]^{1/2} \]
Summary

- The DAE global approach (implicit scheme and Newton method) is very efficient.
- Substitution is very efficient.
- The model without logarithms is more efficient than with logarithms.
Summary

- The DAE global approach (implicit scheme and Newton method) is very efficient
- Substitution is very efficient
- The model without logarithms is more efficient than with logarithms

Future work

- Precipitation-dissolution with vanishing $p$
- Parallel computations