

# **EFFICIENT SOLUTION OF MULTICOMPONENT TRANSPORT SYSTEMS**

*G. (Kumar) Mahinthakumar  
Center for Computational Sciences  
Oak Ridge National Laboratory*

*Caroline Tebes—Stevens, USEPA, Athens, GA*

*Albert J. Valocchi, UIUC, Urbana, IL*

*Faisal Saied, NCSA, Urbana, IL*

*March 25, 1999*

## OUTLINE

---

### ● Background

- Nonlinear solution strategies
- Reactive multicomponent transport system
- Features of our transport code

### ● Implementation Details

- Sequential Iterative Algorithm (SIA)
- Modified SIA's

### ● Test problem description

### ● Convergence Results

- Influence of reaction rates
- Influence of problem size

### ● Conclusions

# NON LINEAR SOLUTION STRATEGIES FOR COUPLED KINETIC REACTION – TRANSPORT SYSTEMS

---

## Operator Splitting

- most commonly employed.
- the reaction system is separately solved as a coupled ODE system.
- no convergence problems (no iterations!).
- can suffer from accuracy problems if the reaction and transport time scales are different (e.g. fast reactions).
  - modifications available to improve accuracy (e.g. Strang's splitting)
- modular implementation simplifies adding user defined reactions.

## Sequential Iterative Algorithm

- more accurate than operator splitting for fast kinetic reactions.
- can suffer from convergence problems
- modifications to standard SIA may alleviate some of these problems.

## Full Newton–Raphson

- robust for most systems
- may not be computationally feasible for large coupled systems (huge memory and computational requirements)

## MAIN FEATURES OF TRANSPORT CODE

---

- **Multicomponent 3D advection – dispersion equation with first order kinetic sorption, bioremediation and reversible kinetic (or ion exchange) reactions.**
- **Galerkin finite – elements with 8 – node hexahedral elements.**
- **Non – linearity handled by SIA iterations.**
- **Matrix solution performed by iterative Krylov solvers. Options available to use BICGSTAB, GMRES(m), ORTHOMIN(k), and CGS.**
- **Parallelized for distributed memory architectures using 2D domain decomposition and MPI message passing.**
- **Tested on a variety of parallel architectures including the Intel Paragon, SGI Origin 2000, Cray T3E, Convex Exemplar, and IBM SP [PP99].**

# TRANSPORT EQUATIONS

$$\frac{\partial C_i}{\partial t} = \nabla \cdot (\mathbf{D} \cdot \nabla C_i) - \nabla \cdot (C_i \mathbf{v}) + \frac{q}{\theta} (C_T - C_{0i}) + R_i \quad i = 1, 2, 3, \dots, nc$$

**MULTICOMPONENT  
TRANSPORT**

$$D_{ij} = \alpha_L |\mathbf{v}| \delta_{ij} + (\alpha_L - \alpha_T) \frac{v_i v_j}{|\mathbf{v}|} + D_m \quad i, j = x, y, z$$

**DISPERSION  
TENSOR**

$$R_i = E_i + B_i (\lambda_i + \lambda_{wi}) C_T - \frac{\rho}{\theta} \frac{\partial S_i}{\partial t} \quad i = 1, 2, 3, \dots, nc$$

**REACTIONS**  
(reversible, biodegradation, decay, sorption)

$$E_i = (\eta_i - \eta'_i) \left[ k_f \prod_{j=1}^{j=nc} C_j^{\eta_j} - k_r \prod_{k=1}^{k=nc} C_k^{\eta'_k} \right] \quad i = 1, 2, 3, \dots, nc$$

**REVERSIBLE  
ION EXCHANGE**

$$B_i = -\mu_{\max} F_i X \prod_{j=1}^{j=nc} f_{ji} \left( \frac{C_j}{K_j + C_j} \right) \quad i = 1, 2, 3, \dots, nc$$

**BIODEGRADATION**

$$\frac{\partial X}{\partial t} = \mu_{\max} X Y \prod_{i=1}^{i=nc} g_i \left( \frac{C_i}{K_i + C_i} \right) - bX$$

**BIOMASS  
GROWTH AND DECAY**

$$\frac{\partial S_i}{\partial t} = k_{ri} (K_{di} C_T - S_i) - (\lambda_i + \lambda_{si}) S_i \quad i = 1, 2, 3, \dots, nc$$

**SORPTION**



## SEQUENTIAL ITERATIVE ALGORITHM (SIA)

---

- Reactive transport equation:

$$\frac{\partial C_i}{\partial t} + L(C_i) = R_i \quad i = 1, 2, 3, \dots, nc$$

- Standard SIA (SIA-0):

$$\frac{C_i^{n+1,p+1} - C_i^n}{\Delta t} + L\left(C_i^{n+1,p+1}\right) = R_i^{n+1,p} \quad \begin{array}{l} n = \text{time level} \\ p = \text{iteration level} \end{array}$$

set  $R_i^{n+1,0} = R_i^n$  and iterate until  $|C_i^{n+1,p+1} - C_i^{n+1,p}|_{\infty} < \epsilon$

- Above example is shown for fully implicit time stepping but in our results we used Crank–Nicolson time stepping.
- We compute non-linear reaction term  $R_i$  with trial solution from the previous iteration ( $C_i^{n+1,p}$ ).
- Left side contributes to both 'matrix' and 'rhs' of the global matrix system and for SIA-0, right side contributes only to the 'rhs'.
- A full matrix solve is required for each transporting component in each iteration.
- More details in [Yeh and Tripathi, 1989; Kinzelbach et al., 1991] WRR articles.

## MODIFICATIONS TO STANDARD SIA

---

### ● SIA-1

$$R_i = R_i^p + \left[ \frac{\partial R_i}{\partial C_i} \right]^p (C_i^{p+1} - C_i^p)$$

- $R_i$  is approximated using truncated first-order Taylor-Series.
- Time level (n+1) dropped for convenience above.
- Second term on the right contributes to the diagonal of the matrix (assuming lumped formulation). Other terms contribute to the rhs.
- Full details given in Ph.D. thesis of Caroline Tebes-Stevens [UIUC, September 1998] and [Tebes-Stevens et al., *Journal of Hydrology*, August 1998]

## MODIFICATIONS TO STANDARD SIA (CONTD.)

---

### ● SIA-0'

$$R_i = (R'_i)^p (C_i^{p+1}) + (R'_i)^p$$

- A simple modification of the SIA-0 algorithm.
- $R_i$  is divided into 2 portions: factorizable and unfactorizable.
- The first term on the right contributes to the diagonal of the matrix (semi-implicit). Other terms contribute to the rhs.
- No derivative computation required!
- For some reactions either the factorizable or unfactorizable portion may not exist
  - good if factorizable portion exists and unfactorizable portion does not exist
  - reduces to standard SIA-0 if factorizable portion does not exist

### ● SIA-1'

$$R_i = (R'_i)^p (C_i^{p+1}) + (R'_i)^p + \left[ \frac{\partial R_i}{\partial C_i} \right]^p (C_i^{p+1} - C_i^p)$$

- improves implicitness of SIA-1 by replacing the first term by SIA-0' approximation

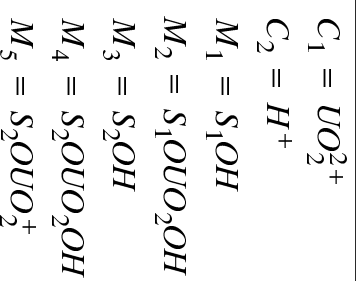
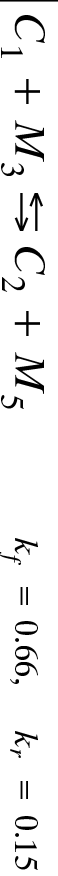
## TEST PROBLEM

---

● **Modified version of the Uranium–Quartz example of [Kohler et al., WRR, 1996] also in [Tebes–Stevens, Ph.D. thesis, 1998]**

● **The original problem used equilibrium speciation reactions**

● **3 reversible kinetic reactions involving 2 aqueous ( $C_i$ ) and 5 immobile ( $M_j$ ) components**



$$R(C_1) = (K_{f_1}C_1M_1 - K_{r_1}C_2^2M_2) + (K_{f_2}C_1M_3 - K_{r_2}C_2^2M_4) + (K_{f_3}C_1M_3 - K_{r_3}C_2M_5)$$

$$R(C_2) = -2(K_{f_1}C_1M_1 - K_{r_1}C_2^2M_2) - 2(K_{f_2}C_1M_3 - K_{r_2}C_2^2M_4) - (K_{f_3}C_1M_3 - K_{r_3}C_2M_5)$$

## CONVERGENCE TESTS

---

- Two types of tests conducted
- First test consisted of increasing the difficulty level of the problem by scaling  $k_f$  and  $k_r$  by factors of 1, 2, 5, and 10
  - 1D column flow ( $v=12.6$  cm/hr).
  - Same flow and dispersion parameters as Kohler et al., 1996.
  - The problem size was fixed at  $421 \times 3 \times 3$ .
  - Third type injection boundary condition used for the aqueous components.
  - Solved on a single processor SGI.
- Second test consisted of using the original  $k_f$  and  $k_r$  but inducing 3D flow and increasing the problem size from  $51 \times 11 \times 11$  to  $401 \times 41 \times 41$ 
  - Flow fields generated by solving the steady state saturated flow equation with K-field generated by turning bands.
  - Mean flow in the longitudinal x-direction.
  - Solved on the Intel Paragon XPS/150 using 4 to 64 processors.
- Both tests used Crank–Nicolson time stepping and the BICGSTAB solver.

## INFLUENCE OF REACTION RATES

---

- The original forward and reverse reaction rates for each of the 3 reactions were scaled by 1, 2, 5, 10.
- Total number of iterations are reported below for 100 time steps.
- Maximum number of iterations per time step is limited to 100 with a relative tolerance of  $10^{-8}$
- In the beginning of the computations negative concentrations occurred for some components!
- **damping option had to be turned on for convergence!** (not good)

Factor	SIA-0	SIA-0'	SIA-1	SIA-1'
1	1431	850	842	791
2	DNC	1220	1293	1070
5	DNC	DNC	7728	5190
10	DNC	DNC	DNC	DNC

DNC = Did Not Converge

- CPU time per iteration is slightly higher (10%) for the SIA-1 methods
- None of the methods converged for factor = 10 !

## INFLUENCE OF PROBLEM SIZE

---

- Heterogeneous 3D flow field.
- Problem size increased gradually.
- Maximum number of iterations per time step is limited to 100 with a relative tolerance of  $10^{-8}$

Problem Size	SIA-0	SIA-0'	SIA-1	SIA-1'
51x11x11	2210	1091	970	875
101x21x21	2191	1080	956	820
201x41x41	2219	1109	981	866
401x41x41	2315	1125	992	892

- Problem size has minimal impact on all SIA methods.
- The semi-implicit methods (SIA-0', SIA-1, SIA-1') have even smaller variation than SIA-0.

## CONCLUSIONS

---

- Slight modifications of the SIA algorithm can improve convergence for certain problems.
- Convergence is severely affected by fast reaction rates.
  - Our tests with other mildly nonlinear reactions (e.g. bioremediation) indicated that SIA-0' is sufficient and comparable in performance to SIA-1.
- Problem size had minimal impact on convergence rates for all SIA algorithms
- In terms of convergence performance the following order can be generally stated:
  - SIA-1' > SIA-1 > SIA-0' > SIA-0.
- For fast reaction rates none of the SIA methods converged!
  - Problems may be unrealistic.
- Options to use either 'operator splitting' or 'SIA' is recommended for production codes.