Preconditioning the Newton-Krylov method for reactive transport

Michel Kern, L. Amir, A. Taakili

INRIA Paris-Rocquencourt — Maison de la Simulation

MoMaS workshop
Reactive Transport Modeling in the Geological Sciences
Institut Henri Poincaré November 2015

Outline

1. Model and formulation
2. Elimination and preconditioning
3. Application to MoMaS benchmark

Motivations

Study algorithms for reactive transport that

1. Keep chemistry and transport codes separate
2. Allow strong numerical coupling

Outline

1. Model and formulation
2. Elimination and preconditioning
3. Application to MoMaS benchmark

Model setup

One phase reactive transport of N_s species, undergoing N_r reactions

Transport operator $Lc = \nabla.(uc - D\nabla c)$ on $\omega \subset \mathbf{R}^d, d = 1,2,3$. u Darcy velocity, D diffusion dispersion tensor.

Chemical system Aqueous and sorption reactions, all equilibrium.

$$S\begin{pmatrix} X \\ \bar{X} \end{pmatrix} = \begin{pmatrix} S_{cc} & 0 \\ S_{\bar{c}c} & S_{\bar{c}\bar{c}} \end{pmatrix} \begin{pmatrix} X \\ \bar{X} \end{pmatrix} \leftrightarrows \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

Model setup

One phase reactive transport of N_s species, undergoing N_r reactions

Transport operator $Lc = \nabla.(uc - D\nabla c)$ on $\omega \subset \mathbf{R}^d, d = 1,2,3$. u Darcy velocity, D diffusion dispersion tensor.

Chemical system Aqueous and sorption reactions, all equilibrium.

$$s\begin{pmatrix} X \\ \bar{X} \end{pmatrix} = \begin{pmatrix} S_{cc} & 0 \\ S_{\bar{c}c} & S_{\bar{c}\bar{c}} \end{pmatrix} \begin{pmatrix} X \\ \bar{X} \end{pmatrix} \leftrightarrows \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

Mass conservation

$$\begin{aligned} \phi\partial_t c \quad + Lc &= S_{cc}^T r \quad + S_{\bar{c}c}^T \bar{r}, \\ \phi\partial_t \bar{c} \quad\quad &= \quad\quad S_{\bar{c}\bar{c}}^T \bar{r}, \end{aligned}$$

Mass action laws

$$\begin{pmatrix} S_{cc} & 0 \\ S_{\bar{c}c} & S_{\bar{c}\bar{c}} \end{pmatrix} \begin{pmatrix} \log c \\ \log \bar{c} \end{pmatrix} = \begin{pmatrix} \log K \\ \log \bar{K} \end{pmatrix},$$

Elimination of equilibrium rates: coupled problem

The kernel matrix (after Saaltink et al (98))

$$U \text{ st } US^T = 0,\ U = \begin{pmatrix} U_{cc} & U_{c\bar{c}} \\ 0 & U_{\bar{c}\bar{c}} \end{pmatrix},$$

$$\begin{aligned} \phi \partial_t \left(U_{cc}\, c + U_{c\bar{c}}\, \bar{c} \right) + U_{cc}\, L c &= 0 \\ \phi \partial_t U_{\bar{c}\bar{c}}\, \bar{c} &= 0. \end{aligned}$$

Transformation

$$\begin{aligned} T &= U_{cc} c + U_{c\bar{c}} \bar{c} = T_l + T_s \\ \bar{T} &= U_{\bar{c}\bar{c}} T_s \end{aligned}$$

Coupled system

$$\begin{aligned} \phi \partial_t T_l + \phi \partial_t T_s + L T_l &= 0 \\ \phi \partial_t \bar{T} &= 0. \end{aligned}$$

Completed by mass action laws

Solving equilibrium chemistry

System of non linear equations

Mass action law

$$\hat{S} \log\left(\frac{c}{\bar{c}}\right) = \log\left(\frac{k}{\bar{k}}\right)$$

Mass conservation

$$Uc + \bar{U}\bar{c} = T \quad T \text{ known from transport}$$

Take concentration logarithms as main unknowns

Use globalized Newton's method (line search, trust region).

Solving equilibrium chemistry

System of non linear equations

Mass action law $$\hat{S} \log\left(\frac{c}{\bar{c}}\right) = \log\left(\frac{k}{\bar{k}}\right)$$

Mass conservation $U c + \bar{U}\bar{c} = T$ T known from transport

Take concentration logarithms as main unknowns

Use globalized Newton's method (line search, trust region).

Role of chemical model

Given total T, split into

Liquid $T_l = Uc$,

Solid $T_s = \bar{U}\bar{c}$

total concentrations

Result of chemical problem

$$T_s = \Psi_C(T)$$

A note on solving the equilibrium system

Chemical system

$$S \log c = \log K, \quad Uc = T$$

Using QR factorization of $S^T = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R_1 \\ 0 \end{pmatrix}$ gives $U = Q_2$.

Write $z = \log c = Q_1 z_1 + Q_2 z_2$, mass action law gives $R_1^T z_1 = \log K$.
Non linear system becomes

$$Q_2^T \exp\left(Q_1 z_1 + Q_2 z_2\right) = T$$

Only unknown is z_2 (size $N_s - N_r$). Recover primary and secondary species !
Jacobian is

$$J_c = Q_2^T C Q_2, \text{ with } C = \mathrm{diag}(c) = \mathrm{diag}(\exp(z)).$$

Source of ill-conditioning is C (cf. J. Carrayrou).

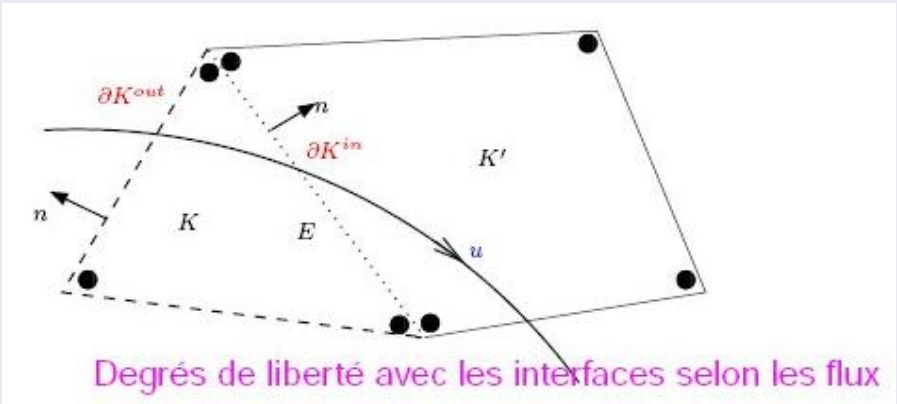
Solution of transport by operator splitting

See Chavent-Jaffré, Ackerer et al., Putti et al., Arbogast et al., ...

Advection step

Explicit, finite volumes

- Locally mass conservative
- Allows unstructured mesh
- CFL condition: use sub-time-steps



Dispersion step

Mixed finite elements (implicit)

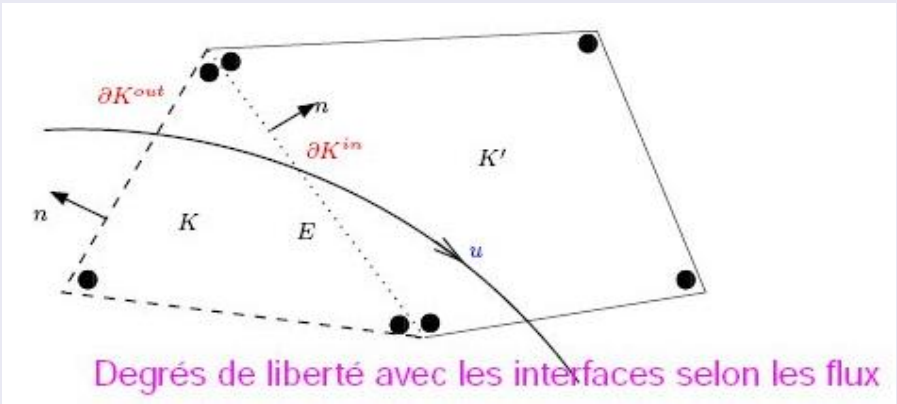
Solution of transport by operator splitting

See Chavent-Jaffré, Ackerer et al., Putti et al., Arbogast et al., ...

Advection step

Explicit, finite volumes

- Locally mass conservative
- Allows unstructured mesh
- CFL condition: use sub-time-steps



Dispersion step

Mixed finite elements (implicit)

Condense transport solver, one time step

$$\underbrace{(M + \Delta t\, L)}_{A} C^{n+1} = M C^n + \Delta t\, f^n \Leftrightarrow C^{n+1} = \Psi_T(f^n, C^n)$$

Outline

1. Model and formulation
2. Elimination and preconditioning
3. Application to MoMaS benchmark

Solution strategies

Fixed point (aka SI) Yeh-Tripathi, Carrayrou et al., Lagneau et al.

- $+$ easy to program, code reuse
- $-$ not robust, small time steps

Solution strategies

Fixed point (aka SI) Yeh-Tripathi, Carrayrou et al., Lagneau et al.

- + easy to program, code reuse
- − not robust, small time steps

Direct substitution (DSA) Lichtner et al., Saaltink et al., Steefel et al.

- + accurate, robust,
- − difficult to code, large non-linear system

Solution strategies

Fixed point (aka SI) Yeh-Tripathi, Carrayrou et al., Lagneau et al.

- + easy to program, code reuse
- − not robust, small time steps

Direct substitution (DSA) Lichtner et al., Saaltink et al., Steefel et al.

- + accurate, robust,
- − difficult to code, large non-linear system

DAE formulation Erhel and de Dieuleveult

- + use quality DAE software, accurate
- − expensive

Solution strategies

Fixed point (aka SI) Yeh-Tripathi, Carrayrou et al., Lagneau et al.

- + easy to program, code reuse
- − not robust, small time steps

Direct substitution (DSA) Lichtner et al., Saaltink et al., Steefel et al.

- + accurate, robust,
- − difficult to code, large non-linear system

DAE formulation Erhel and de Dieuleveult

- + use quality DAE software, accurate
- − expensive

Reduction technique Knabner et al.

- + Efficient, accurate,
- − difficult to code

Solution strategies

Fixed point (aka SI) Yeh-Tripathi, Carrayrou et al., Lagneau et al.

- + easy to program, code reuse
- − not robust, small time steps

Direct substitution (DSA) Lichtner et al., Saaltink et al., Steefel et al.

- + accurate, robust,
- − difficult to code, large non-linear system

DAE formulation Erhel and de Dieuleveult

- + use quality DAE software, accurate
- − expensive

Reduction technique Knabner et al.

- + Efficient, accurate,
- − difficult to code

(Try to) use best of SI and DS

A global method from the fixed-point formulation (1)

Discrete non-linear system

$$\begin{cases} A T_l^{n+1} + M T_s^{n+1} = \underbrace{M(T_l^n + T_s^n)}_{b^n} \\ T_s{}^{n+1} - \Psi_C(T_l{}^{n+1} + T_s{}^{n+1}) = 0 \end{cases}$$

A global method from the fixed-point formulation (1)

Discrete non-linear system

$$\begin{cases} A T_l^{n+1} + M T_s^{n+1} = \underbrace{M(T_l^n + T_s^n)}_{b^n} & \text{uncoupled} \\ T_s{}^{n+1} - \Psi_C(T_l{}^{n+1} + T_s{}^{n+1}) = 0 \end{cases}$$

Elimination of T_l

$$F_2(T_s) = T_s{}^{n+1} - \Psi_C\left(T_s{}^{n+1} - A^{-1} M T_s^{n+1} + A^{-1} b^n\right) = 0$$

Can be solved by block Gauss-Seidel or by Newton's method

A global method from the fixed-point formulation (1)

Discrete non-linear system

$$\begin{cases} A T_l^{n+1} + M T_s^{n+1} = \underbrace{M(T_l^n + T_s^n)}_{b^n} & \text{uncoupled} \\ T_s{}^{n+1} - \Psi_C(T_l{}^{n+1} + T_s{}^{n+1}) = 0 \end{cases}$$

Elimination of T_l

$$F_2(T_s) = T_s{}^{n+1} - \Psi_C\left(T_s{}^{n+1} - A^{-1} M T_s^{n+1} + A^{-1} b^n\right) = 0$$

Can be solved by block Gauss-Seidel or by Newton's method

Residual computation

1. Apply Ψ_T : solve transport for each species,
2. Apply Ψ_C: solve chemistry for each grid cell.

A global method from the fixed-point formulation (2)

- + Non-intrusive approach
- + Precipitation can be included
- − One chemical equilibrium solve for each function evaluation

A global method from the fixed-point formulation (2)

- + Non-intrusive approach
- + Precipitation can be included
- − One chemical equilibrium solve for each function evaluation

Solution by Newton-Krylov method

- Solve the linear system by an iterative method (GMRES)
- Requires only jacobian matrix by vector products, Jacobian not stored
- Keep transport and chemistry as black boxes
- Used for CFD, shallow water, radiative transfer(Keyes, Knoll, JCP 04), and for reactive transport (Hammond et al., Adv. Wat. Res. 05)

Inexact Newton framework

Approximation of the Newton's direction $\|f'(x_k)d + f(x_k)\| \leq \eta_k \|f(x_k)\|$
adaptive choice of the forcing term η_k (Kelley, Eisenstat and Walker)

L. Amir, MK (Comp. Geosci. 09)

Jacobian computations

Coupled formulation

Jacobian with block structure

$$J = \begin{pmatrix} A & M \\ -D & I - D \end{pmatrix}$$

D is Jacobian of chemistry operator (cf. Knabner et al.: resolution function), computed implicitly

Inverting Jacobian ; solve transport

Elimination of T_l

Jacobian of F_2 : $J_2 = I - D + DA^{-1}M$.

Computing Jacobian: solve transport. Solve cannot use matrix form (GMRES OK)

Preconditioning

- Essential for good linear performance
- Difficult for matrix free formulation
- Block preconditioners preserve problem structure

Jacobi

$$P_{\text{Jac}} = \begin{pmatrix} A & 0 \\ 0 & I - D \end{pmatrix}$$

$$P_{\text{Jac}}^{-1} J = \begin{pmatrix} I & A^{-1}M \\ -(I-D)^{-1}D & I \end{pmatrix}$$

Gauss-Seidel

$$P_{\text{GS}} = \begin{pmatrix} A & 0 \\ -D & (I - D) \end{pmatrix}$$

$$P_{\text{GS}}^{-1} J = \begin{pmatrix} I & A^{-1}M \\ 0 & (I-D)^{-1}(I - D + DA^{-1}M) \end{pmatrix}$$

Elimination of T_l equivalent to Schur comp. of Gauss-Seidel.

Field of value analysis

GMRES convergence not determined by eigenvalues (Greenbaum et al. 96)

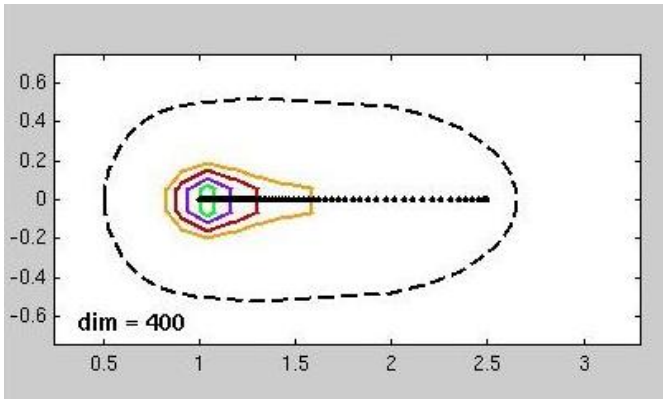
$W(A) \equiv \{x^* A x | x \in \mathbb{C}^n, \|x\| = 1\}$, convex set, contains eigenvalues of A

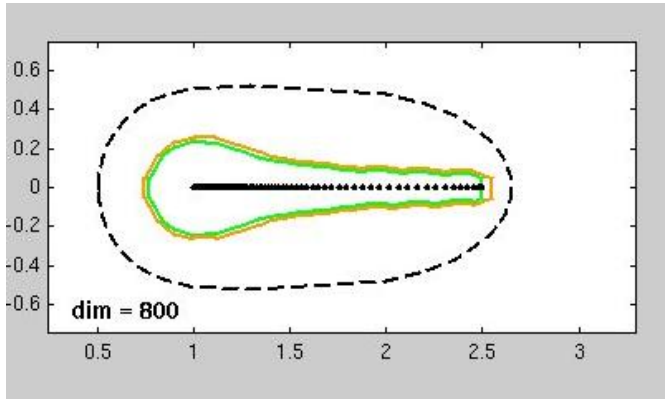
$$\|r_k\|_2 \leq 2 \min_{p \in \mathscr{P}_k^*} \max_{z \in W(A)} |p(z)| \, \|r_0\|_2.$$

Simplified system: one species with sorption

$\Lambda(P_{\text{Jac}}^{-1} J) \subset [1 - iCh, 1 + iCh]$ $\qquad$ $\Lambda(J_2) \subset [1, 1 + Ch^2]$

Bounded away from 0 independently of h.





Eingenvalues, field of values and pseudospectrum for GS preconditioning

Preconditioner performance

1D model (Matlab + Sundials), $h = 0.05$, $K_L = 1.$, $\sigma = 1.5$, and $\Delta t = 0.0135$.

Mesh dependence : constant forcing term

	h		$h/2$		$h/4$		$h/8$	
	NI	LI	NI	LI	NI	LI	NI	LI
None	3	104	3	167	3	267	3	453

Mesh dependence : adaptive forcing term

	h		$h/2$		$h/4$		$h/8$	
	NI	LI	NI	LI	NI	LI	NI	LI
None	8	42	8	76	10	105	10	177

NI: # nonlinear iters, NLI: total # linear iters.

Preconditioner performance

1D model (Matlab + Sundials), $h = 0.05$, $K_L = 1.$, $\sigma = 1.5$, and $\Delta t = 0.0135$.

Mesh dependence : constant forcing term

	h		$h/2$		$h/4$		$h/8$	
	NI	LI	NI	LI	NI	LI	NI	LI
None	3	104	3	167	3	267	3	453
BGS	3	48	3	48	3	48	3	45

Mesh dependence : adaptive forcing term

	h		$h/2$		$h/4$		$h/8$	
	NI	LI	NI	LI	NI	LI	NI	LI
None	8	42	8	76	10	105	10	177

NI: # nonlinear iters, NLI: total # linear iters.

Preconditioner performance

1D model (Matlab + Sundials), $h = 0.05$, $K_L = 1.$, $\sigma = 1.5$, and $\Delta t = 0.0135$.

Mesh dependence : constant forcing term

	h		$h/2$		$h/4$		$h/8$	
	NI	LI	NI	LI	NI	LI	NI	LI
None	3	104	3	167	3	267	3	453
BGS	3	48	3	48	3	48	3	45
Elimination	3	41	3	41	3	41	3	40

Mesh dependence : adaptive forcing term

	h		$h/2$		$h/4$		$h/8$	
	NI	LI	NI	LI	NI	LI	NI	LI
None	8	42	8	76	10	105	10	177

NI: # nonlinear iters, NLI: total # linear iters.

Preconditioner performance

1D model (Matlab + Sundials), $h = 0.05$, $K_L = 1.$, $\sigma = 1.5$, and $\Delta t = 0.0135$.

Mesh dependence : constant forcing term

	h		$h/2$		$h/4$		$h/8$	
	NI	LI	NI	LI	NI	LI	NI	LI
None	3	104	3	167	3	267	3	453
BGS	3	48	3	48	3	48	3	45
Elimination	3	41	3	41	3	41	3	40

Mesh dependence : adaptive forcing term

	h		$h/2$		$h/4$		$h/8$	
	NI	LI	NI	LI	NI	LI	NI	LI
None	8	42	8	76	10	105	10	177
BGS	8	23	7	24	7	22	8	25

NI: # nonlinear iters, NLI: total # linear iters.

Preconditioner performance

1D model (Matlab + Sundials), $h = 0.05$, $K_L = 1.$, $\sigma = 1.5$, and $\Delta t = 0.0135$.

Mesh dependence : constant forcing term

	h		$h/2$		$h/4$		$h/8$	
	NI	LI	NI	LI	NI	LI	NI	LI
None	3	104	3	167	3	267	3	453
BGS	3	48	3	48	3	48	3	45
Elimination	3	41	3	41	3	41	3	40

Mesh dependence : adaptive forcing term

	h		$h/2$		$h/4$		$h/8$	
	NI	LI	NI	LI	NI	LI	NI	LI
None	8	42	8	76	10	105	10	177
BGS	8	23	7	24	7	22	8	25
Elimination	5	15	5	15	5	15	5	15

NI: # nonlinear iters, NLI: total # linear iters.

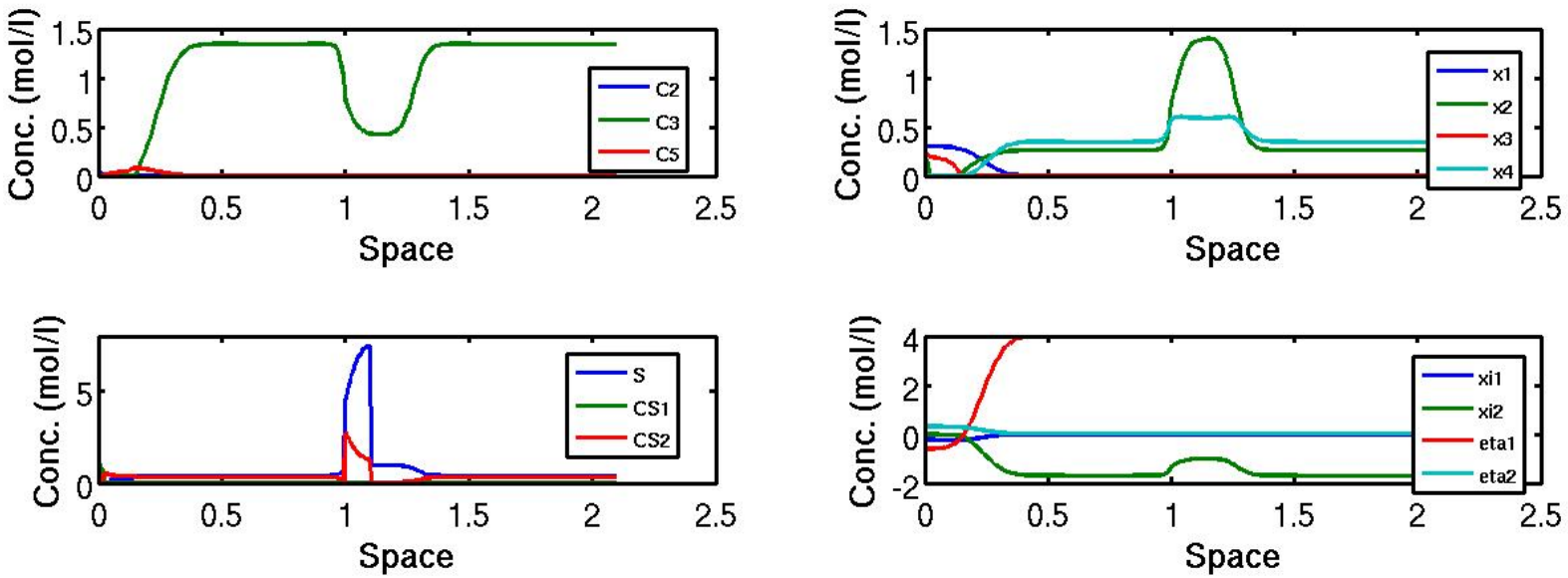
Outline

1. Model and formulation
2. Elimination and preconditioning
3. Application to MoMaS benchmark

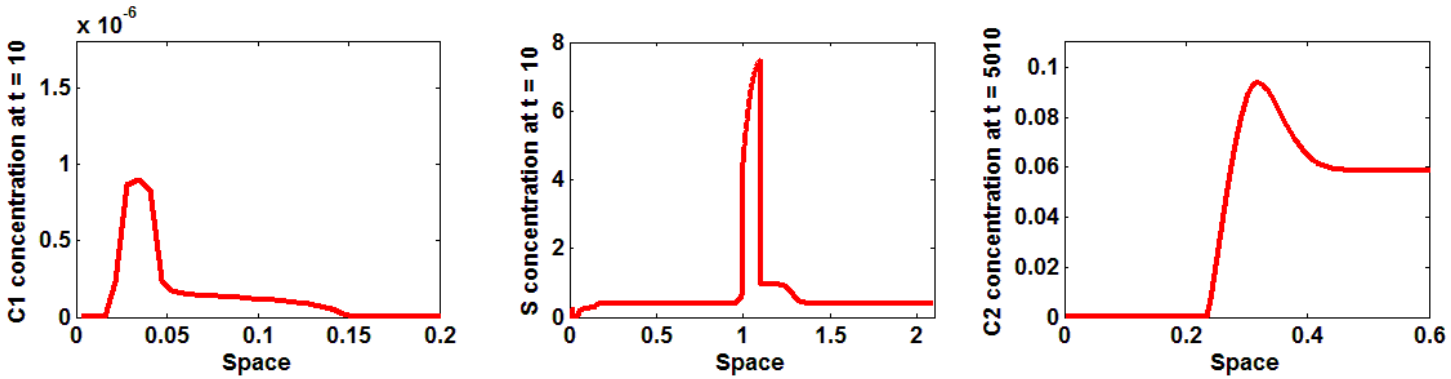
MoMaS benchmark (easy, 1D)

- 9 liquid, 3 sorbed primary species.
- Huge variation in equilibrium constants, large stoichiometric coeffs.
- Long simulation time
- Set-up by J. Carrayrou, MK, P. Knabner (Comp. Geo. 2009)

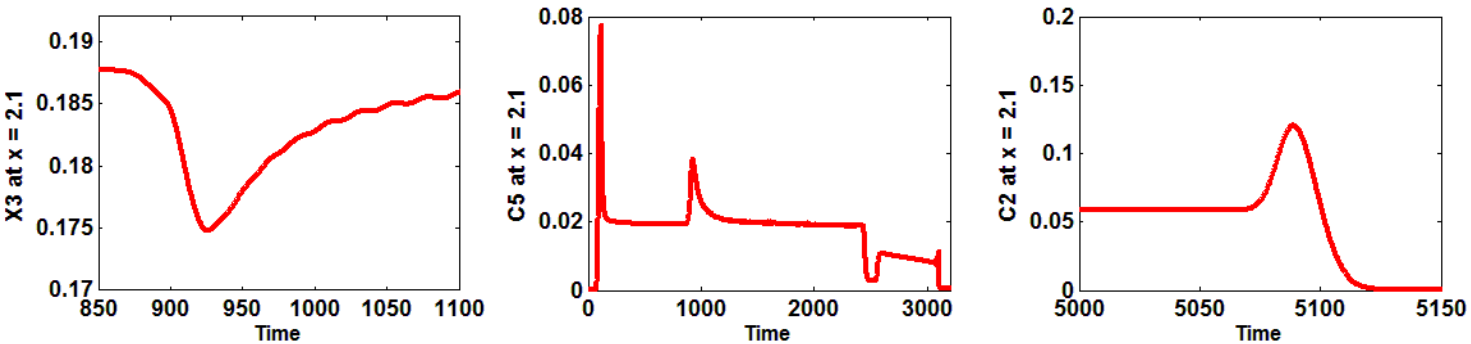
Species concentrations, $t = 10$



Concentration profiles



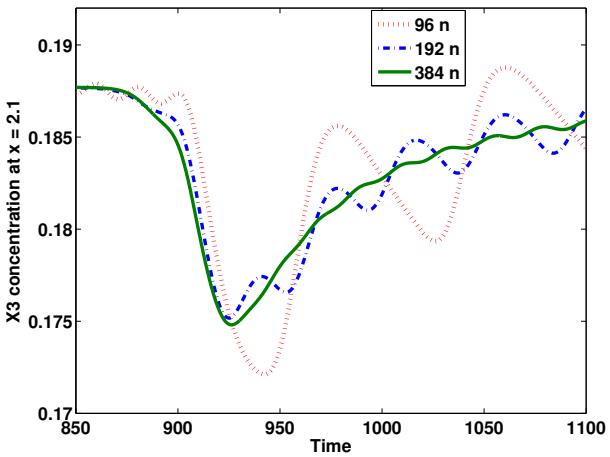
Left: C_1 at $t = 10$, middle S at $t = 10$, right: C_2 at $t = 5010$

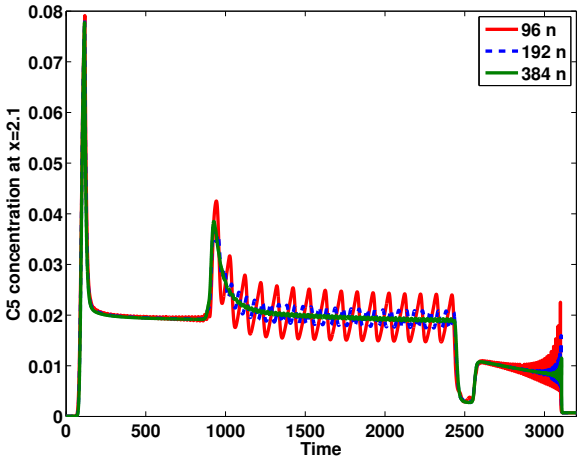


Concentrations at right end of the domain as a function of time.

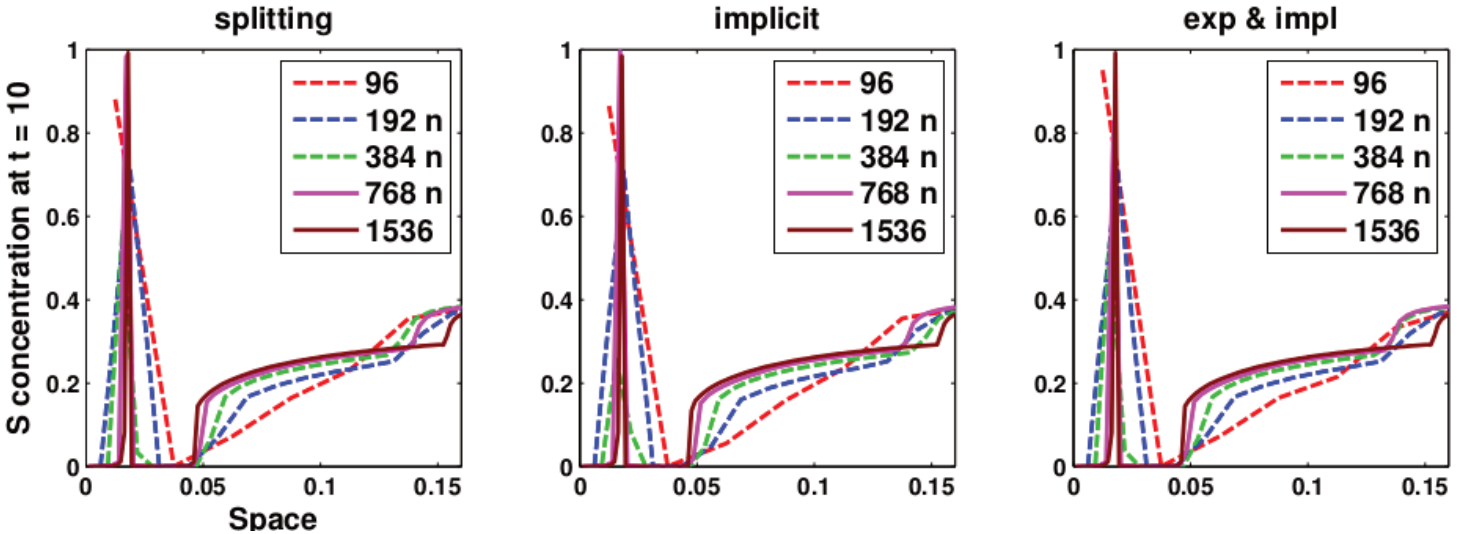
Left: $X3$, middle: $C5$, right: $C2$

Accuracy, spatial discretization



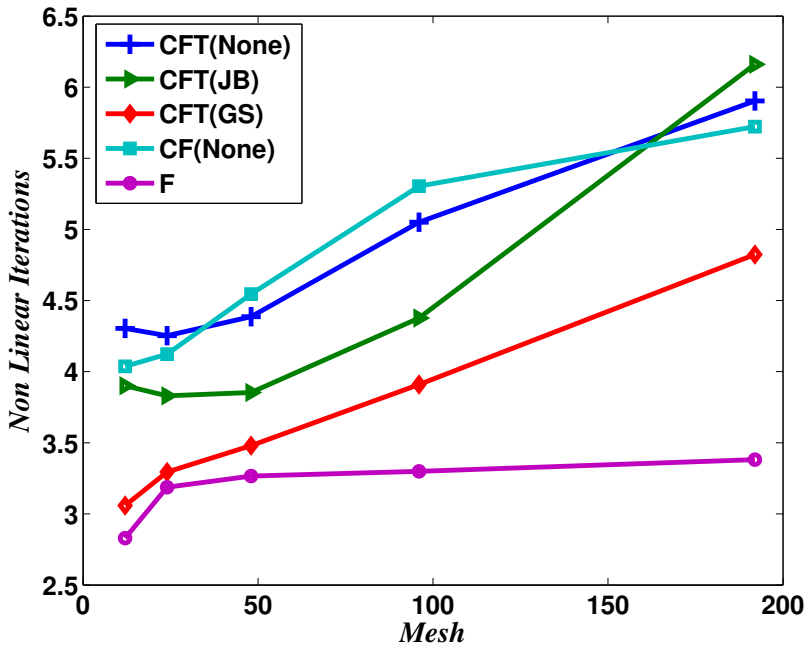


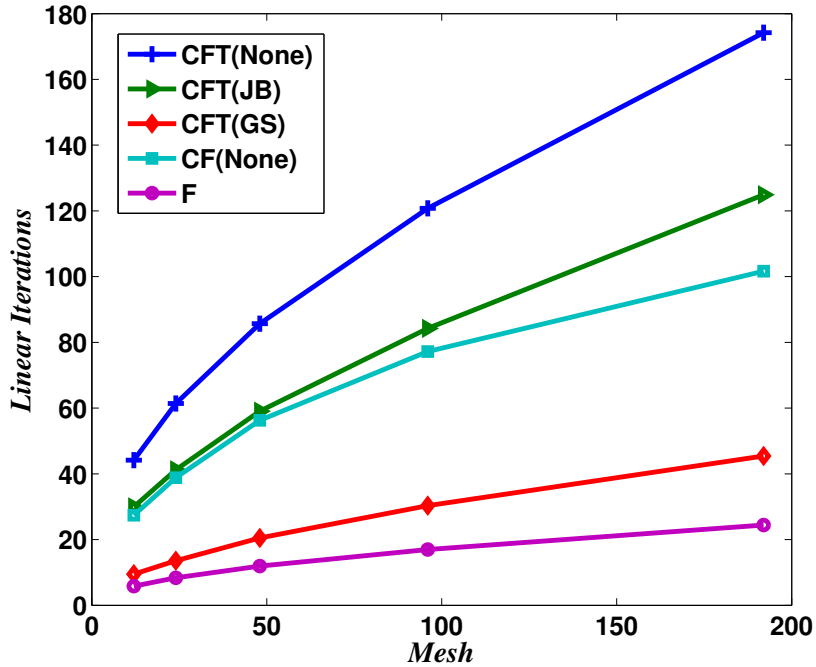
Elution curves: left X_3, right C_5



Concentration of species S at $t = 10$, different schemes

Comparison of preconditioning strategies





Left: Non-linear iterations, right: linear iterations