TOTAL
COMMITTED TO BETTER ENERGY

LINEAR SOLVERS FOR RESERVOIR SIMULATION

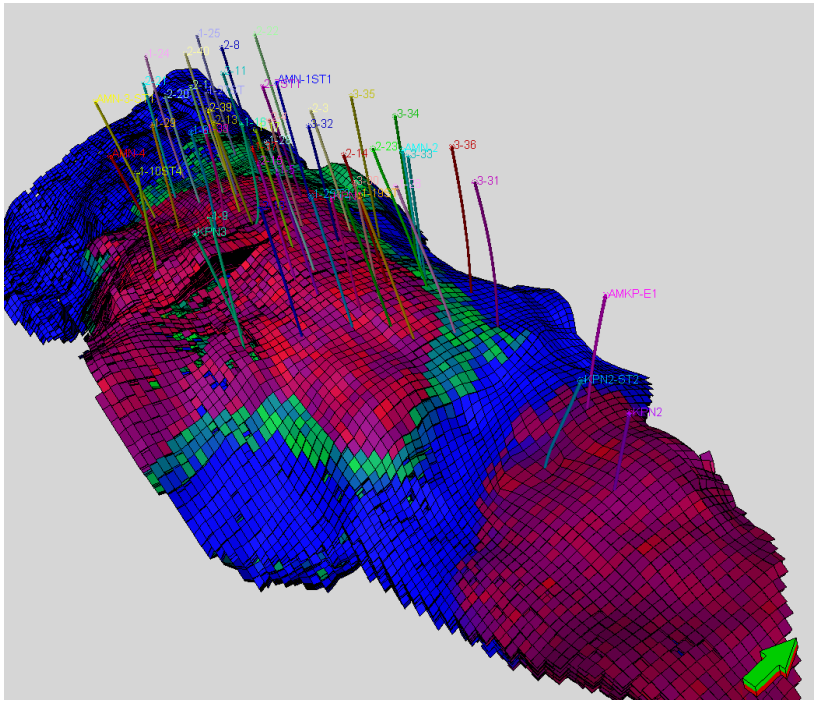
CERFACS
Serge Gratton, Pavel Jiranek, Xavier Vasseur
INRIA-LJLL
Hussam Al Daas, Laura Grigori
TOTAL E&P
Pascal Hénon

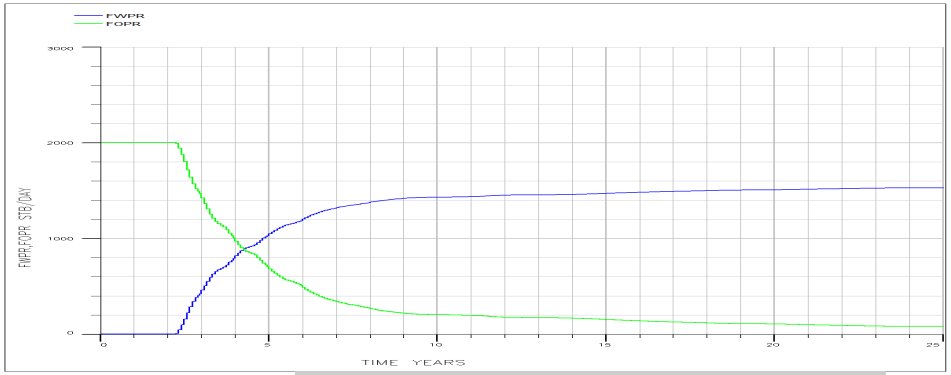
SUMMARY

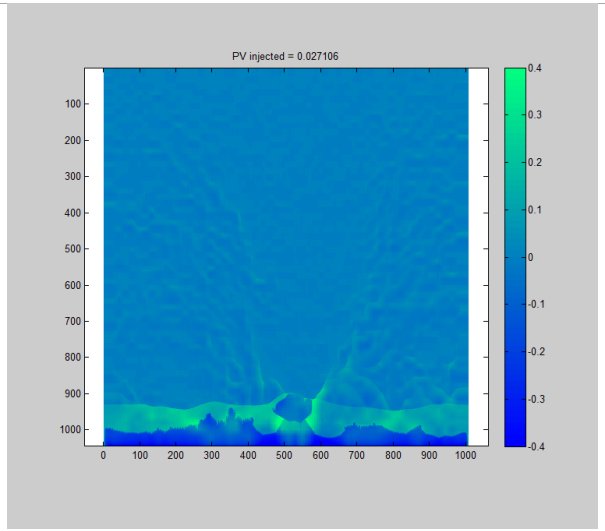
- Introduction
- Works on AMG for reservoir (collaboration with CERFACS)
- PhD 2015-2017: Enlarge GMRES (INRIA: H. Al Daas, L. Grigori)
- Conclusions and Prospects

INTRODUCTION

RESERVOIR SIMULATION : PURPOSES







- Estimation of Recovery Factor, Production plateau, EOR, modelling physic
- We have a limited knowledge of underground properties (fault, kr, ..)
- A big part of the work consists in « history matching » : needs many runs

DYNAMIC SIMULATOR ➔ MAIN EQUATIONS TO BE SOLVED

Mathematical equations

Continuity Equation (= Material Balance)

$$\frac{\partial m_c}{\partial t} = -\mathrm{div}(\rho_p \vec{V}_p) + q_p$$

Darcy's Law

$$\vec{V}_p = -\underline{\underline{K}} \frac{kr_p}{\mu_p}\left(\vec{\nabla} P_p - \rho_p g \vec{z}\right)$$

One mass conservation equation per constituent; in Black-oil :

- Water component in water phase
- Oil component in both oil and gas phases
- Gas component in both oil and gas phases

Pb: No analytical solution in the general case

→ **numerical solution**

→ **discretize in space (gridding)**
discretize in time (timesteps)

Numerical equation (for each cell k)

Continuity Equation + Darcy's Law for each Component

$$\frac{dM_k}{dt} = -\sum_i F_{ki} - \sum_{wells} Q_{k\_to\_wells}$$

dM_k *mass accumulated during the current timestep dt*

F_{ki} *net flow rate from cell k into all neigbouring blocks (i) during dt*

$Q_{k\ to\ wells}$ *net flow rate from cell k into wells during dt*

Non-Linear Residual

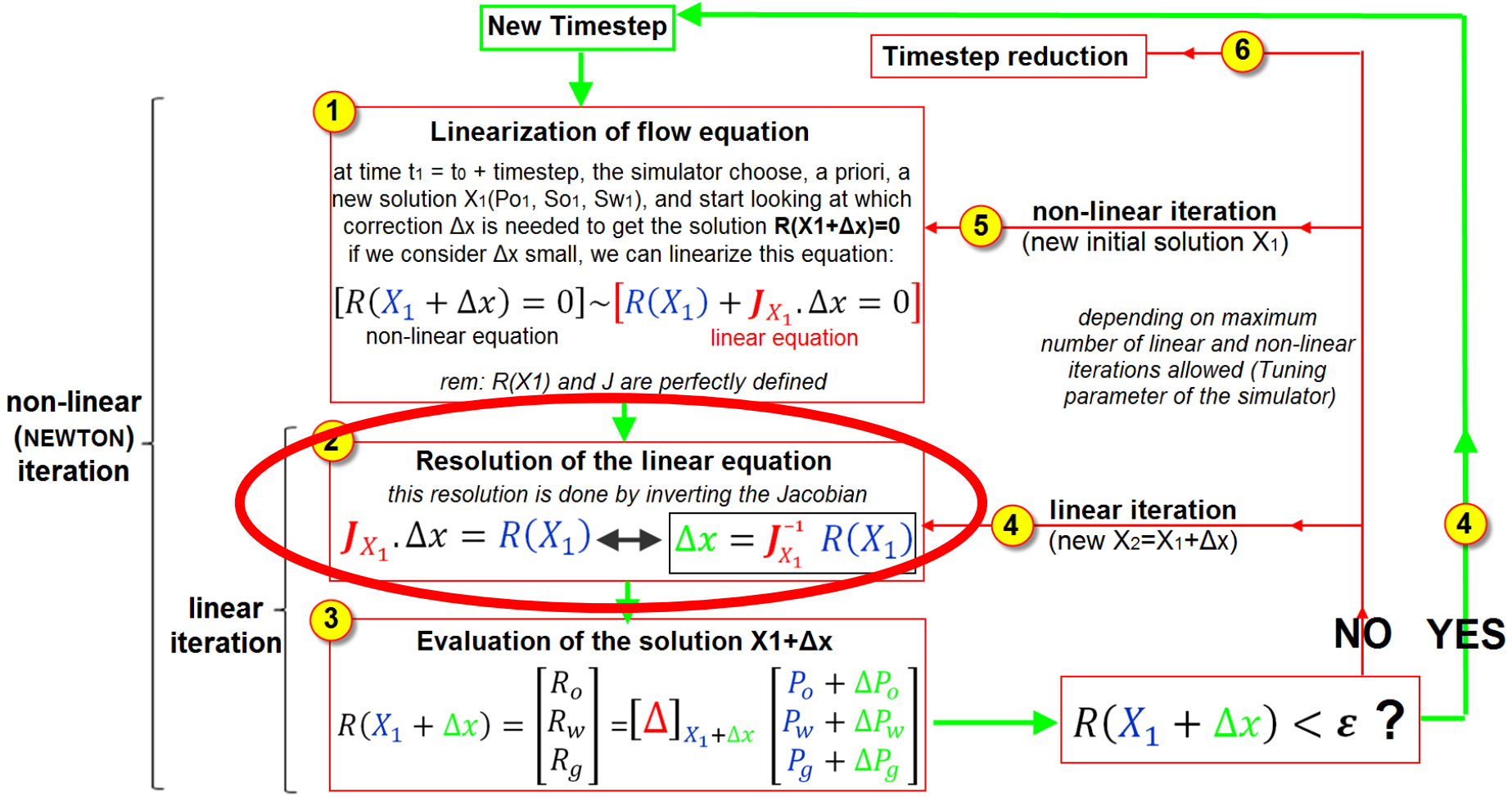
$$\left(R_{fl}\right)_{cell\_k} = \frac{dM_k}{dt} + \sum_i F_{ki} + \sum_{wells} Q_{k\_to\_wells}$$

1st Convergence criteria: $\left(R_{fl}\right)_{cell\_k} < \varepsilon$

Conclusion:

➔ For each cell and for each component, convergence is evaluated by measuring the non-linear residual $\mathbf{R_{fl}}$

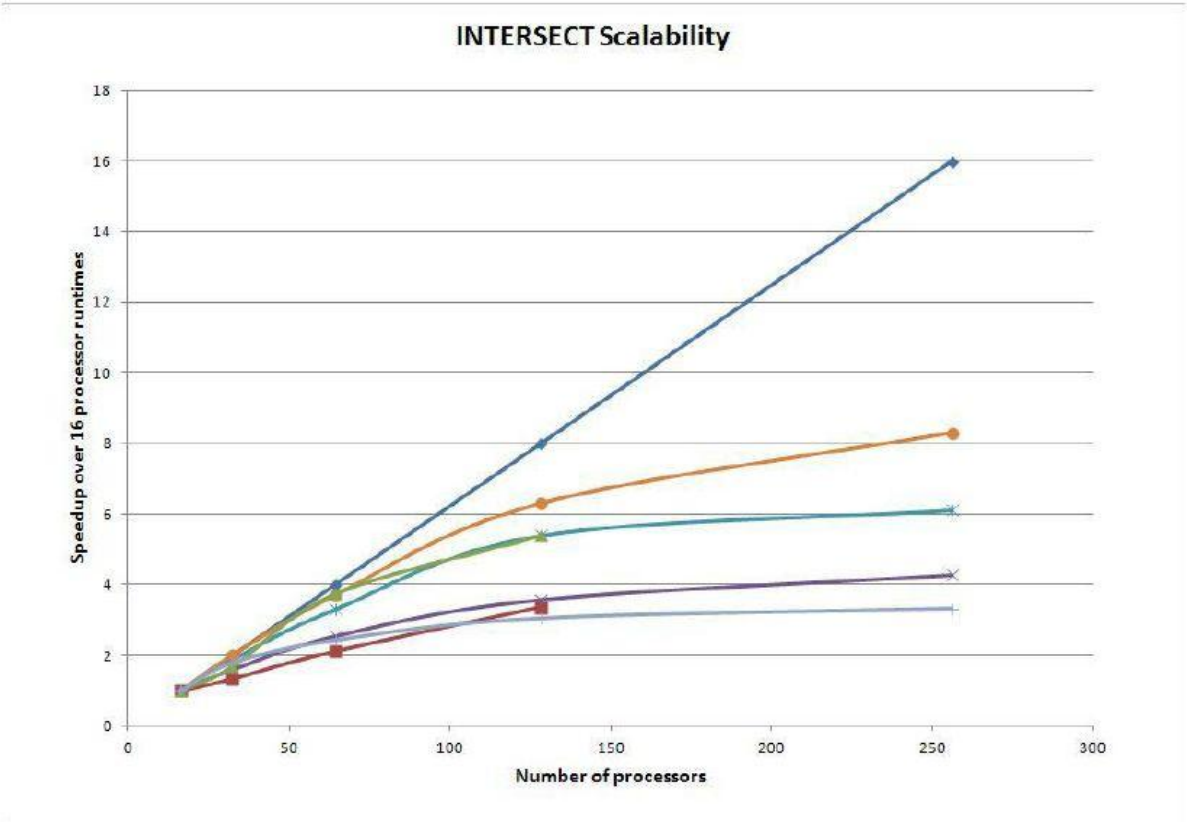
LINEAR EQUATION RESOLUTION



Conclusion: **Most of the CPU is spent on linear resolution**

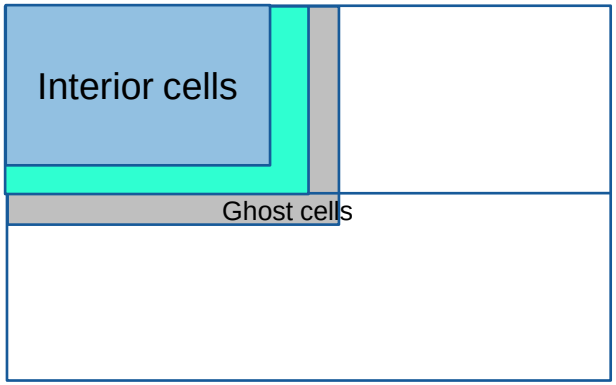
PARALLEL PERFORMANCE

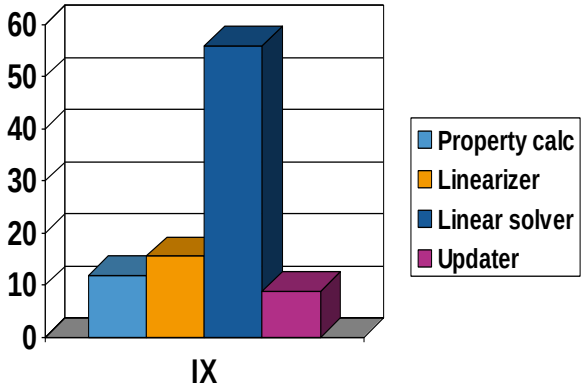
- Scalability of a few models : (upper one =13M cells BO: around 50k cells/processors)



Source : Schlumberger : SPE ACTE, Oct 31th 2011

SCALABILITY BOTTLENECK IN RESERVOIR SIMULATION



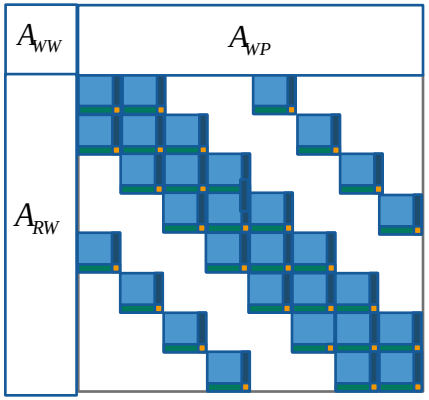


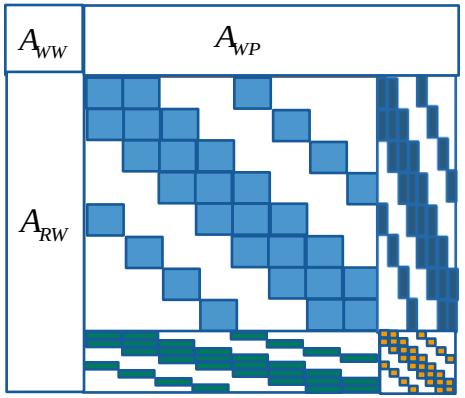
- Distributed memory framework (MPI)
- Load balancing : work per reservoir cell varies during the simulation.
- Linear solver method CPR-AMG is scalable with number of unknowns (weak scalability) but poorly scalable with number of processors (strong scalability)

LINEAR SOLVER : CONSTRAINT PRESSURE RESIDUAL

- The Constraint Pressure Residual (CPR) solver : this solver specific to reservoir is a two stages method (John Wallis and co. SPE 1985) :

$$\begin{pmatrix} A_{WW} & A_{WR} \\ A_{RW} & A_{RR} \end{pmatrix} \qquad P^t.A_{RR}.P = \begin{pmatrix} A_{SS} & A_{SP} \\ A_{PS} & A_{PP} \end{pmatrix} \qquad \begin{pmatrix} A_{WW} & A_{WS} & A_{WP} \\ A_{SW} & A_{SS} & A_{SP} \\ A_{PW} & A_{PS} & A_{PP} \end{pmatrix}$$





$$M_{CPR}^{-1} = M_2^{-1}(I - A.M_1^{-1}) + M_1^{-1}$$

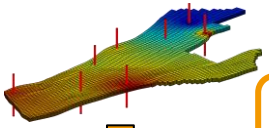
- M_1 : **1er stage is global : find approximate pressure** (near-elliptic problem : AMG)
- M_2 : **2nd stage is applied on the A_{RR} system** (eg BILU(0)) : block preconditioner

IMPROVING AMG FOR RESERVOIR SIMULATION

RESERVOIR SIMULATION: ONE ITERATION OF CPR-AMG

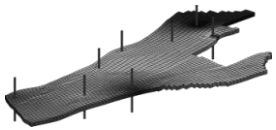
The critical performance bottleneck in reservoir is the pressure solver part

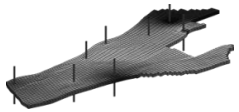
$$M_{CPR}^{-1} = M_2^{-1}(I - A.M_1^{-1}) + M_1^{-1}$$

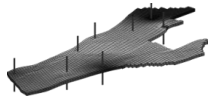


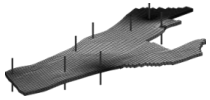
Pressure pseudo-decoupling

CPR

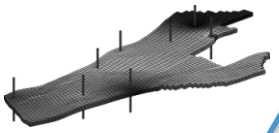


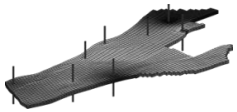


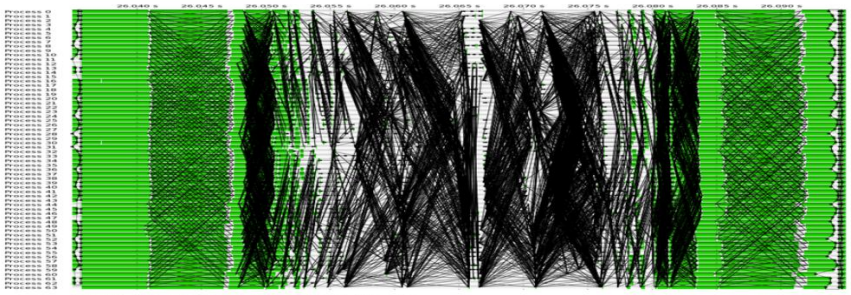


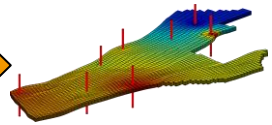


AMG update (different grid levels)





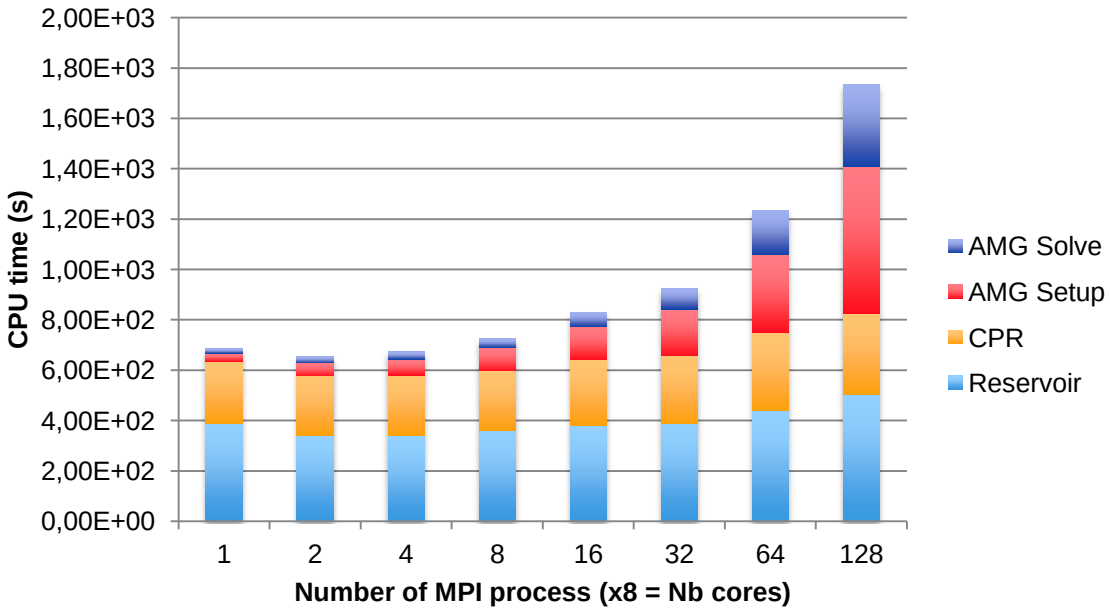




Level	#Rows	#Nz	$s_{\min}$	$s_{\max}$	s_{avg}
1	191'102'976	1'335'730'176	4	7	7.0
2	65'691'648	458'721'792	4	7	7.0
3	32'514'048	483'720'192	6	15	14.9
4	15'096'050	230'760'782	6	19	15.3
5	6'235'899	364'876'769	14	105	58.5
6	1'193'641	134'568'815	25	179	112.7
7	121'332	14'660'566	24	192	120.8
8	12'912	1'211'004	20	156	93.8
9	1'480	89'468	9	103	60.5
10	126	1'698	2	28	13.5

SCALABILITY IN A RESERVOIR SIMULATION

- Example on a one million grid blocks BO model (SPE10) in TPP
- The figure represents TCPU*number of procs (perfect == same length)
- The scalability bottleneck is clearly in AMG (setup + solve)

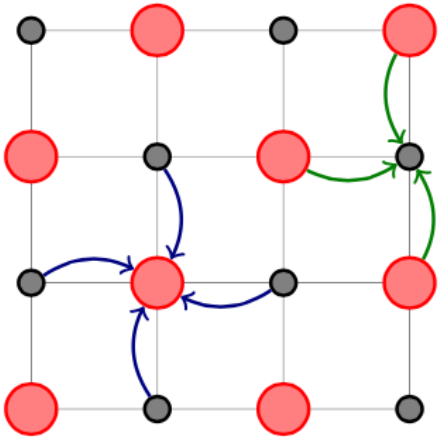


PMG: HYBRID AGGREGATION/CLASSIC COARSENING

- Pavel Jiranek : post-doc CERFACS 2012 – 2014 then 15 month in CSTJF until 11-30-2015
- Hybrid strategy Aggregation/RS is more efficient than standard RS strategy
- PMG package relies on MPI/Thread parallelization
- Ref: « Reducing complexity of algebraic multigrid by aggregation », in Num. Lin. Algebra. with applications
- AMG based on Aggregation is efficient for the two first levels

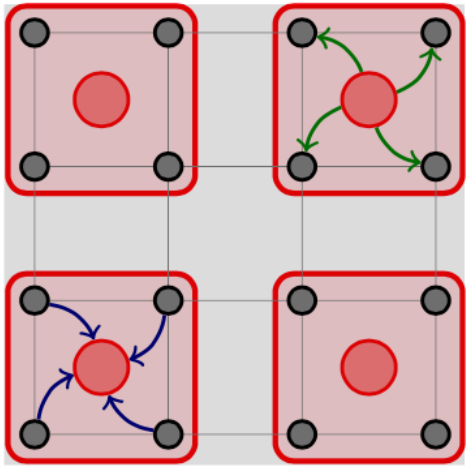
Classic coarsening

$$-(\mathbf{A})_{ij} \geq \theta \max_{k \neq i}[-(\mathbf{A})_{ik}] \quad \text{for some } \theta \in [0, 1].$$



Aggregation coarsening

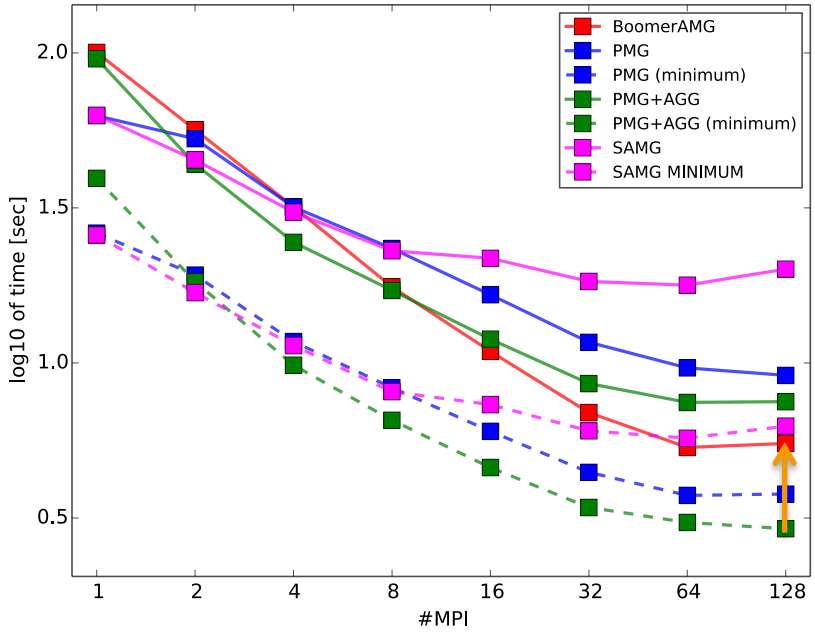
$$|(\mathbf{A})_{ij}| \geq \theta \sqrt{|(\mathbf{A})_{ii}(\mathbf{A})_{jj}|} \quad \text{for some } \theta \geq 0.$$



RESULTS: PMG VS BOOMERAMG VS SAMG

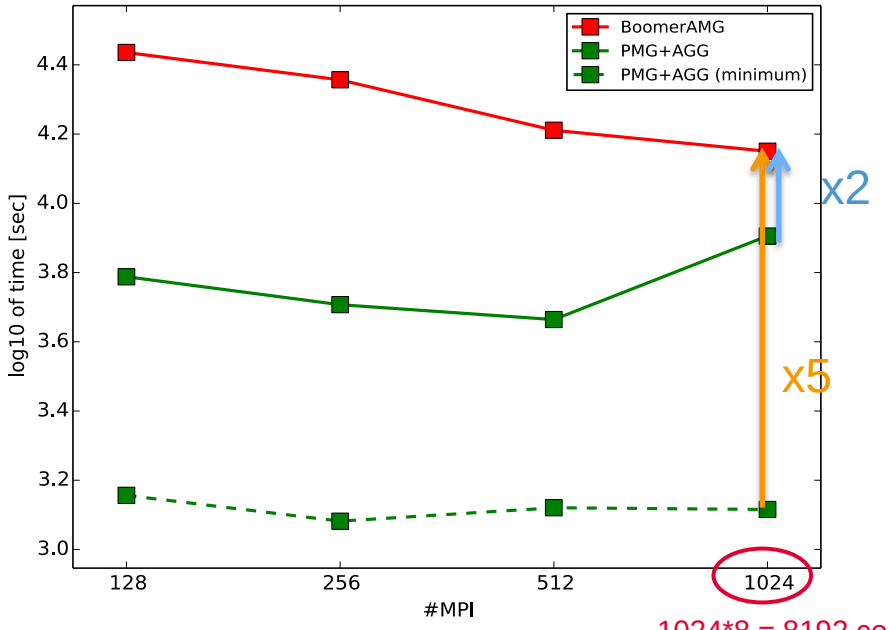
- BoomerAMG from Lawrence Livermore (DOE).
- SAMG from Fraunhofer Institute.
- Minimum setup consists in keeping same grid interpolators for several linear system (we limit to the same time step)
- On small-medium cases (< 1’000’000 cells); no significant improvement over BoomerAMG for total simulation time.

AMG CPU Time



RESULTS: PMG VS BOOMERAMG (MPI+THREAD)

- Viscous fingering study.
- Tuned parameter for BoomerAMG: aggressive coarsening
- 60’000’000 cells model. Test up to 8192 cores (1024 processors of PANGEA1).
- The speed-up is more important (first layers of AMG represents more computations)



ENLARGED KRYLOV METHOD

Slides courtesy of Laura Grigori and Hussam Al Daas

GMRES: SCALABILITY LIMITATION

Algorithm 2 GMRES

1: $r_0 = b - Ax_0$, $v_1 = \frac{r_0}{\|r_0\|_2}$
2: **for** $j = 1 : m$ **do**
3: Matrix-vector multiplication
4: $w = Av_j$
5: Orthogonolization
6: **for** $i = 1 : j - 1$ **do**
7: $h_{i,j} = v_i^H w$
8: $w = w - h_{i,j} v_i$
9: **end for**
10: **if** $\|w\|_2 \neq 0$ **then**
11: $v_j = \frac{w}{\|w\|_2}$
12: **else**
13: Exact breakdown occured. Break
14: **end if**
15: **end for**
16: $H = (h_{i,j})_{i,j}$, $V = \{v_1, \cdots, v_m\}$
17: Solve the least squared problem $y_m = argmin\|Hy - \|r_0\|_2 e_1\|_2$, where e_1 is the first vector of the canonical basis
18: $x_m = x_0 + Vy_m$

ENLARGE KRYLOV METHOD (L. GRIGORI ET AL 2014)

- Partition the matrix into t domains
- split the residual r_{k-1} into t vectors corresponding to the t domains,

$$r_0 \rightarrow T(r_0) = \begin{bmatrix} * & 0 & & 0 \\ \vdots & \vdots & & \vdots \\ * & 0 & & 0 \\ 0 & * & & 0 \\ \vdots & \vdots & & \vdots \\ 0 & * & & 0 \\ & & \ddots & \\ 0 & 0 & & * \\ \vdots & \vdots & & \vdots \\ 0 & 0 & & * \end{bmatrix}$$

- generate t new basis vectors, obtain an enlarged Krylov subspace

$$\mathscr{K}_{t,k}(A, r_0) \quad = \quad span\{T_s(r_0), AT_s(r_0), A^2 T_s(r0), ..., A^{k-1} T_s(r_0)\}$$

- search for the solution of the system $Ax = b$ in $\mathscr{K}_{t,k}(A, r_0)$

ENLARGED GMRES: ALGORITHM

- GMRES: find x in $span\{r_0, Ar_0, ..., A^k r_0\}$ minimizing $||Ax - b||_2$
- Enlarged GMRES: find x in $span\{T(r_0), AT(r_0), ..., A^k T(r_0)\}$ minimizing $||Ax - b||_2$

GMRES

1: **for** $i = 1$ to k **do**
2: $w = Av_{i-1}$
3: block CGS $(w, v_0, \ldots v_{i-1})$
4: update v_i, H
5: **end for**
6: solve LSQ problem with H

Enlarged GMRES

1: $r_0 = Ax_0 - b$, $R_0 = T(r_0)$
2: **for** $i = 1$ to k **do**
3: $W_i = AV_{i-1}$
4: $\tilde{W}_i \leftarrow$ block CGS $(W_k, V_0, \ldots V_{i-1})$
5: $[V_i, R] = TSQR(\tilde{W}_i)$
6: update H
7: **end for**
8: solve LSQ problem with H

- TSQR = Tall Skinny QR

ENLARGED GMRES: DETAILS

The method can be seen as solving t systems, $AX = T(r_0)$.

Detection of systems that converged:

- At iteration k, detect $AX(:, j) = T(r0)(:, j)$
- Add only a subset of relevant vectors to the basis
- Eigenvalues and eigenvectors are well approximated when convergence is detected

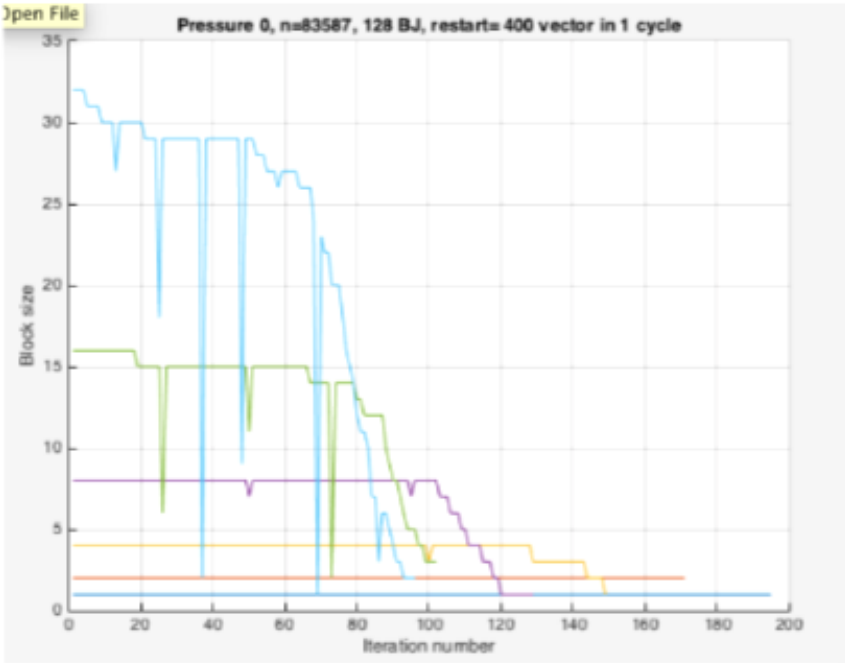
Restarted enlarged GMRES + deflation

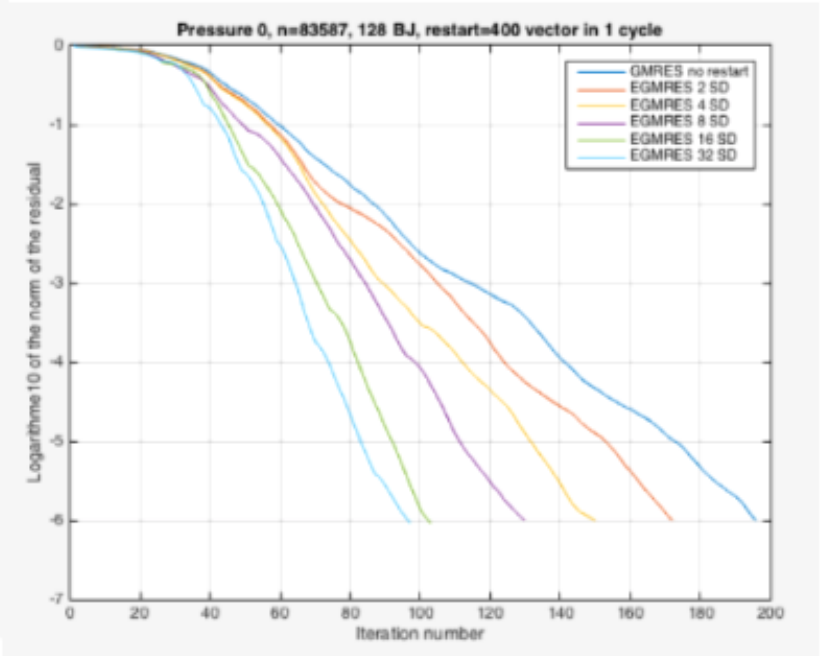
- Enlarged GMRES is restarted when the dimension of the basis becomes large with respect to the memory available
- Deflation recovers the most important information of the enlarged Krylov subspace from previous iterations before restart

ENLARGED GMRES: EXPERIMENTS (1/2)

Number of vectors added to the enlarged subspace per iteration.

Convergence for t varying between 2 and 32.



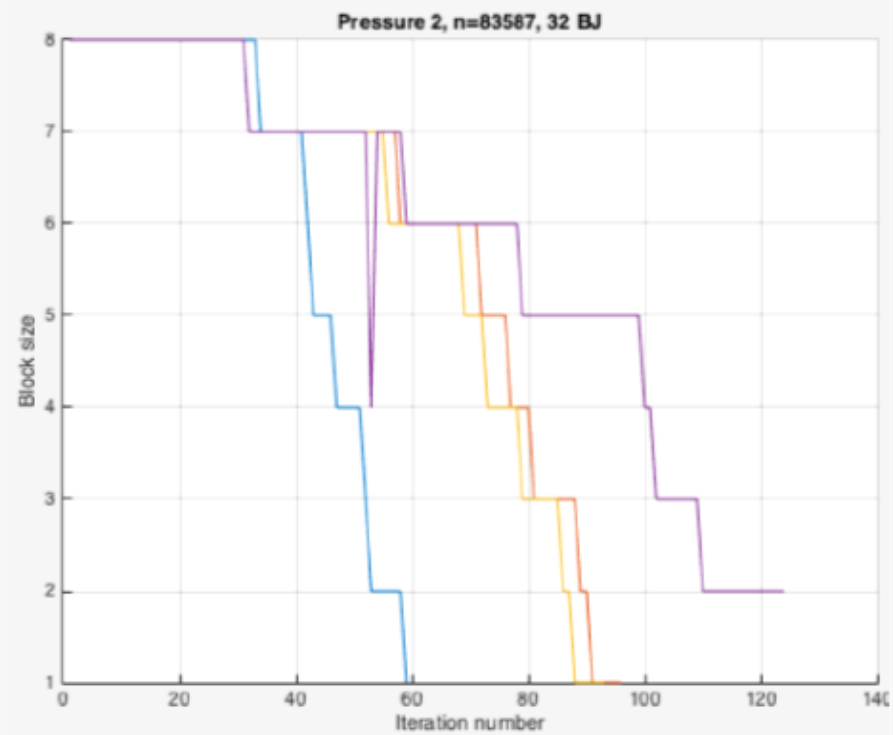


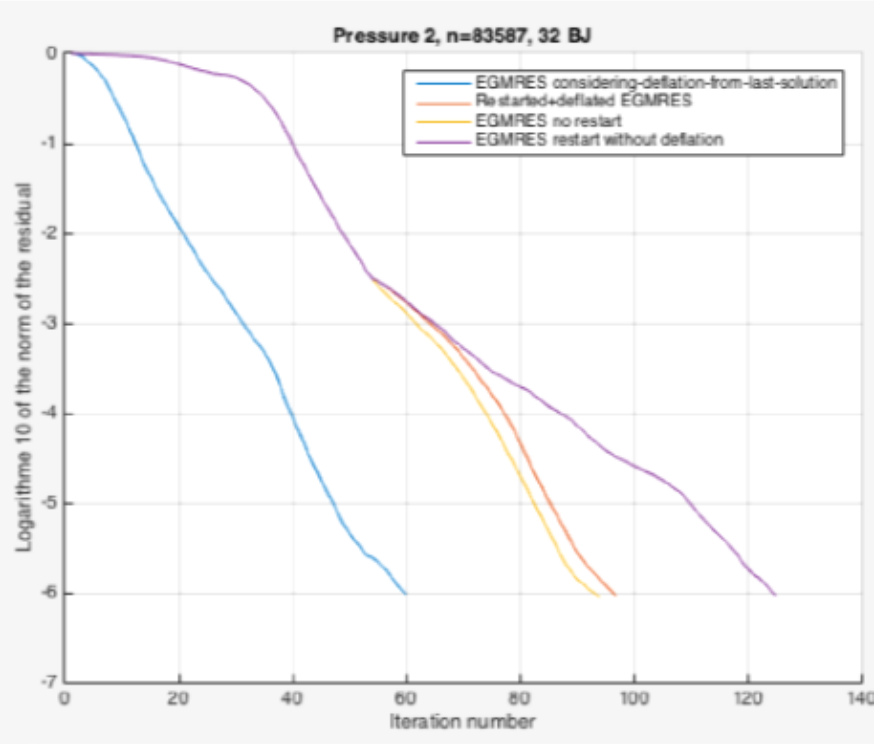
- Pressure matrix from reservoir modelling (Total), 83587 unknowns.
- Preconditioner: block Jacobi with 128 diagonal blocks.
- Restart when the dimension of the basis is 400.

ENLARGED GMRES: EXPERIMENTS (2/2)

Number of vectors added to the enlarged subspace per iteration.

Subsequent solves.





Method:

1. Enlarged GMRES (restart + deflation) used to solve $Ax = b_1$
2. Solve $Ax = b_2$ by using estimated eigenvalues and eigenvectors from previous solve through deflation

CONCLUSION & PROSPECTS

CONCLUSION & PROSPECTS

- Works on AMG: some improvements for large models in parallel have been obtained by an hybrid coarsening strategy.

- Parallel implementation (C++) of EGMRES method and integration in a reservoir simulator

- Experiment EGMRES for pressure solve (with BJ precond) in a reservoir simulator