Iterative solvers for large sparse linear systems

Jocelyne Erhel - SAGE team - INRIA-RENNES

http://www.irisa.fr/sage

PARENG

Ajaccio, April 2011

Krylov iterative methods

Solve $Ax = b$, with A sparse matrix of order n

- A symmetric positive definite: Conjugate Gradient
- A symmetric indefinite
- A non symmetric: GMRES

A symmetric positive definite

One method of choice : Conjugate Gradient (CG)

- algorithm
- properties
- convergence
- preconditioning

Conjugate Gradient

Algorithm

Initialisation

choose x_0

$p_0 = r_0 = b - Ax_0$

For $k = 0, 1 \dots$

$\alpha_k = \frac{\|r_k\|^2}{(Ap_k, p_k)}$

$x_{k+1} = x_k + \alpha_k p_k$

$r_{k+1} = r_k - \alpha_k A p_k$

$\beta_{k+1} = \frac{\|r_{k+1}\|^2}{\|r_k\|^2}$

$p_{k+1} = r_{k+1} + \beta_{k+1} p_k$

End For

Properties

$(r_{k+1}, p_k) = 0$

$(r_{k+1}, r_k) = 0$

$(p_{k+1}, Ap_k) = 0$

$\|r_{k+1}\|_{A^{-1}} = \min_\alpha \|r_k - \alpha A p_{k-1}\|_{A^{-1}}$

Conjugate Gradient - properties

Orthogonality and minimisation

$$(r_k, p_i) = (r_k, r_i) = 0, \quad i \leq k-1$$
$$(p_k, Ap_i) = 0, \quad i \leq k-1$$
$$\|r_{k+1}\|_{A^{-1}} \leq \|r_k\|_{A^{-1}}$$
$$\|r_k\|_{A^{-1}} = \min_{x \in x_0 + Span(p_0,\ldots,p_{k-1})} \|b - Ax\|_{A^{-1}}$$

Krylov method

$\mathcal{K}_k(A, r_0) = Span(r_0, Ar_0, \ldots, A^{k-1}r_0)$ Krylov space

$\mathcal{K}_k(A, r_0) = Span(r_0, r_1, \ldots, r_{k-1}) = Span(p_0, p_1, \ldots, p_{k-1})$

Projection method

$x_k \in x_0 + \mathcal{K}_k(A, r_0)$ Space condition

$r_k \perp \mathcal{K}_k(A, r_0)$ Galerkin condition

Conjugate Gradient - convergence

Polynomial method

$$x_k = x_0 + Q_{k-1}(A) r_0$$
$$r_k = (I - A Q_{k-1}(A)) r_0 = P_k(A) r_0 \text{ with } P_k(0) = 1$$

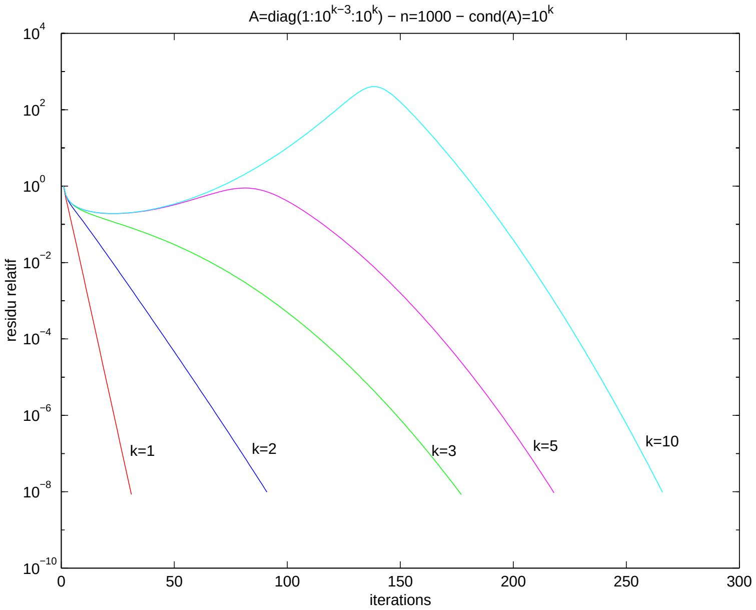
Minmax property - asymptotic convergence

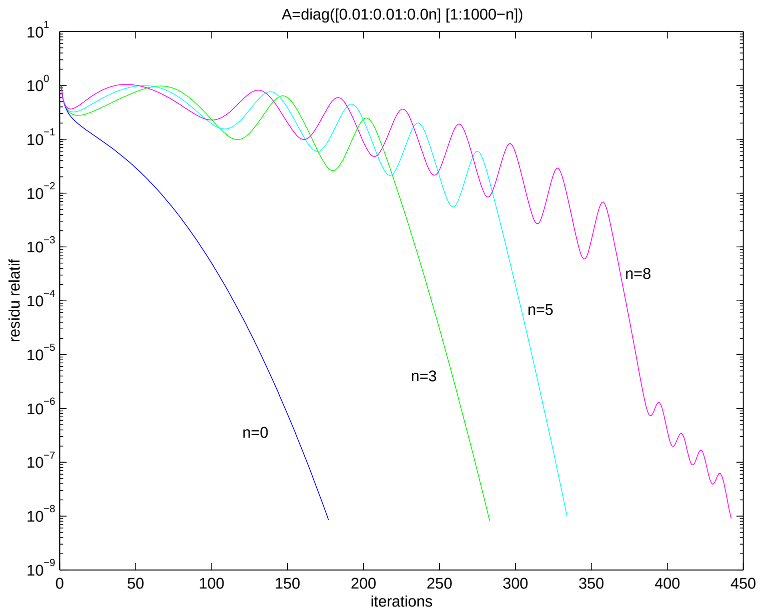
$$A = V \Delta V^{-1} \text{ with } \Delta = diag(\lambda_1, \ldots, \lambda_n)$$
$$0 < \lambda_1 \leq \ldots \leq \lambda_n \text{ and } \kappa(A) = \lambda_n / \lambda_1$$

$$\|r_k\|_{A^{-1}} \leq \|r_0\|_{A^{-1}} \max_{\{\lambda_j\}} |P_k(\lambda_j)|$$
$$\leq \|r_0\|_{A^{-1}} \min_{\{P/deg(P)=k \; P(0)=1\}} \max_{\lambda_1 \leq t \leq \lambda_n} |P(t)|$$

$$\|r_k\|_{A^{-1}} \leq 2 \|r_0\|_{A^{-1}} \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k$$

Conjugate Gradient - convergence





Preconditioned Conjugate Gradient (PCG)

symmetric positif definite preconditioning matrix M

Algorithm

Initialisation

choose x_0

$r_0 = b - Ax_0$

$z_0 = M^{-1} r_0$

$p_0 = z_0$

For $k = 0, 1 \ldots$

$\alpha_k = \frac{(r_k, z_k)}{(Ap_k, p_k)}$

$x_{k+1} = x_k + \alpha_k p_k$

$r_{k+1} = r_k - \alpha_k A p_k$

$z_{k+1} = M^{-1} r_{k+1}$

$\beta_{k+1} = \frac{(r_{k+1}, z_{k+1})}{(r_k, z_k)}$

$p_{k+1} = z_{k+1} + \beta_{k+1} p_k$

End For

PCG - dependencies

Algorithm

For $k = 0, 1 \dots$

$q_k = Ap_k$

$\alpha_k = \frac{(r_k, z_k)}{(q_k, p_k)}$

$x_{k+1} = x_k + \alpha_k p_k$

$r_{k+1} = r_k - \alpha_k A p_k$

$z_{k+1} = M^{-1} r_{k+1}$

$\beta_{k+1} = \frac{(r_{k+1}, z_{k+1})}{(r_k, z_k)}$

$p_{k+1} = z_{k+1} + \beta_{k+1} p_k$

End For

Operations

sparse matrix-vector product

scalar product

vector operation

vector operation

linear system

scalar product

vector operation

Sequence of vector and matrix operations

Parallel PCG

- Parallel sparse matrix-vector product
- Parallel preconditioning
- Parallel operations between vectors
- Synchronisations after each dot product

Some preconditionings

- diagonal or Jacobi
- SSOR or m-step SSOR
- Incomplete Cholesky IC(k) or ICT
- Approximate inverse
- Multilevel (with Schur complements)
- Schwarz
- Multigrid

Jacobi, SSOR and ILU preconditionings

decomposition $A = D + L + U$,
D diagonal, L lower triangular, U upper triangular

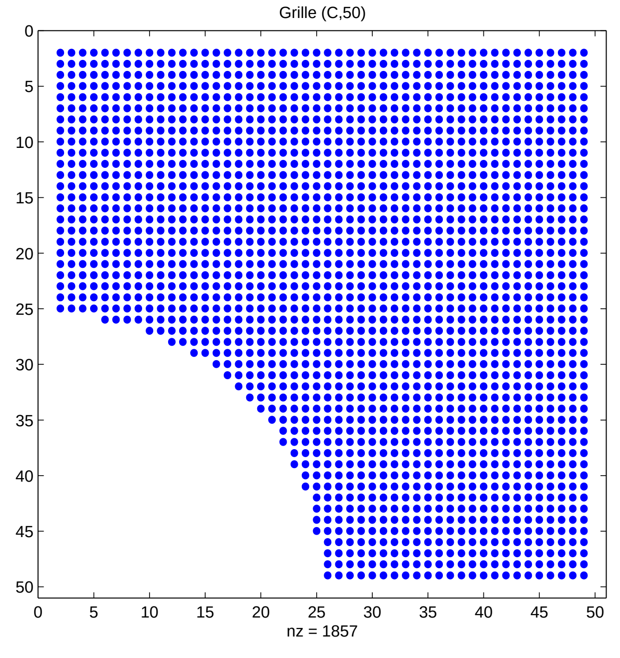
Jacobi: $M = D$, parallel but slow convergence
SSOR: $M = (D + L)D^{-1}(D + U)$, faster convergence but sequential

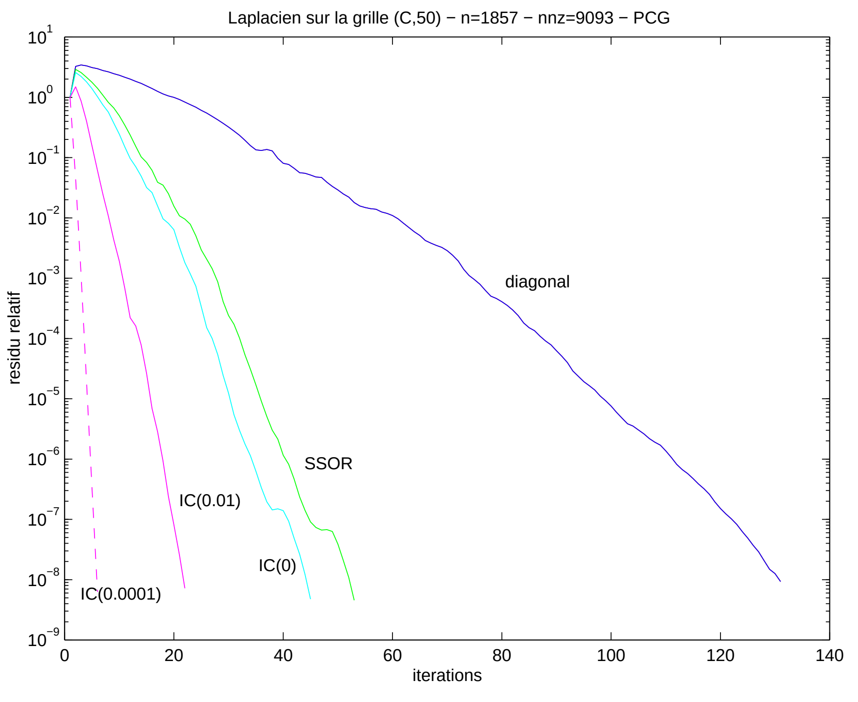
Incomplete Cholesky factorisations

$A = LDL^T + R$, various strategies to choose R
IC(0) : no fill-in
IC(k) : fill-in up to level k
ICT(α) : fill-in with threshold value α

PCG - comparisons

Laplacian on a Finite difference grid





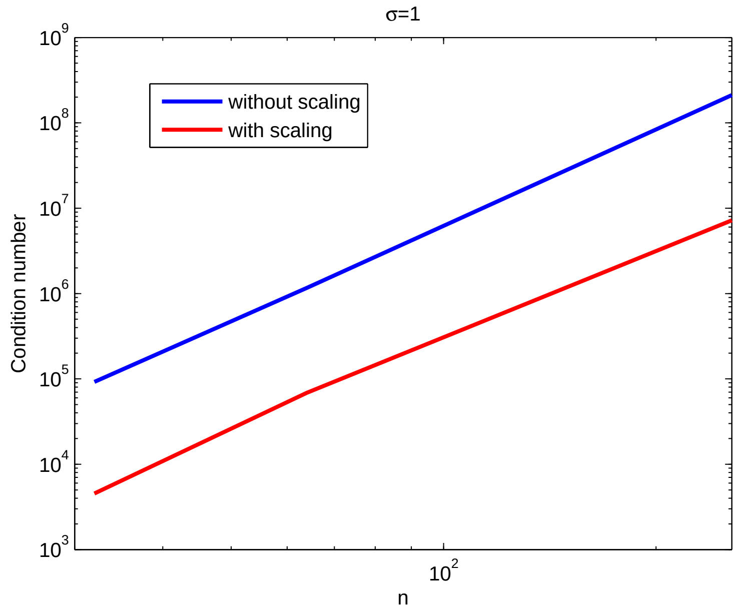
Flow equations in heterogeneous porous media

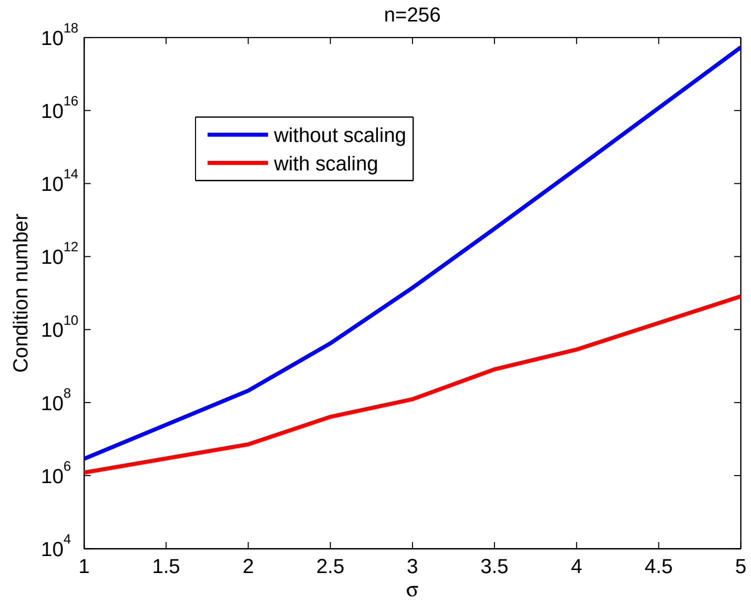
Flow equations in a 2D domain

Log-normal permeability random field with exponential covariance

Finite Volume with a structured mesh

Estimated condition number



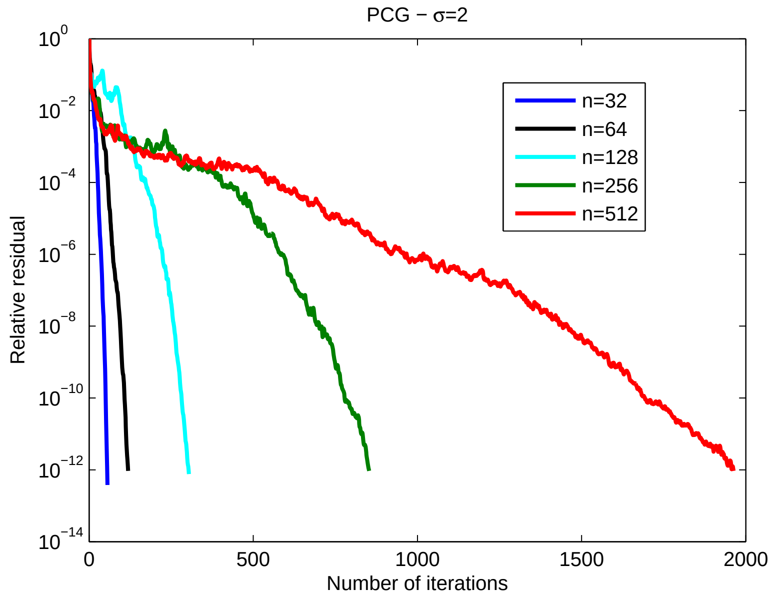


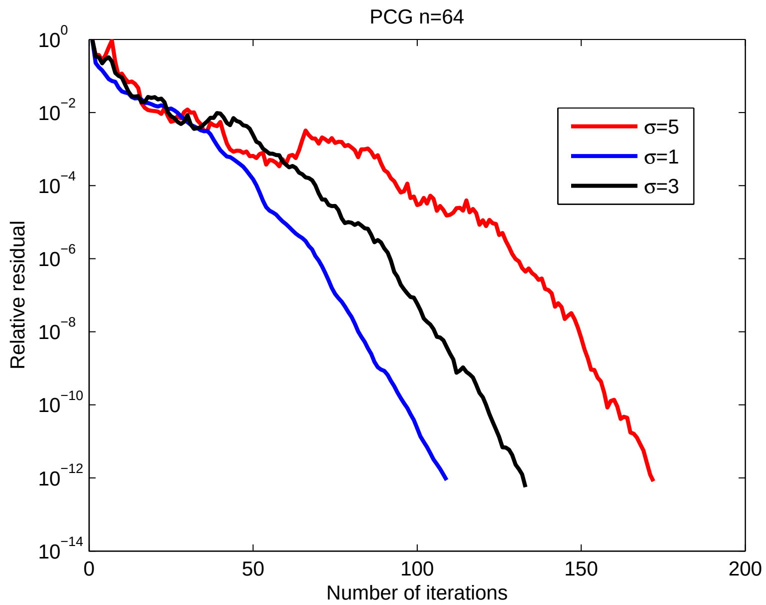
MICAS ANR-funded project

common work with Anthony Beaudoin and Jean-Raynald de Dreuzy

PCG for heterogeneous porous media

IC(0) preconditioner

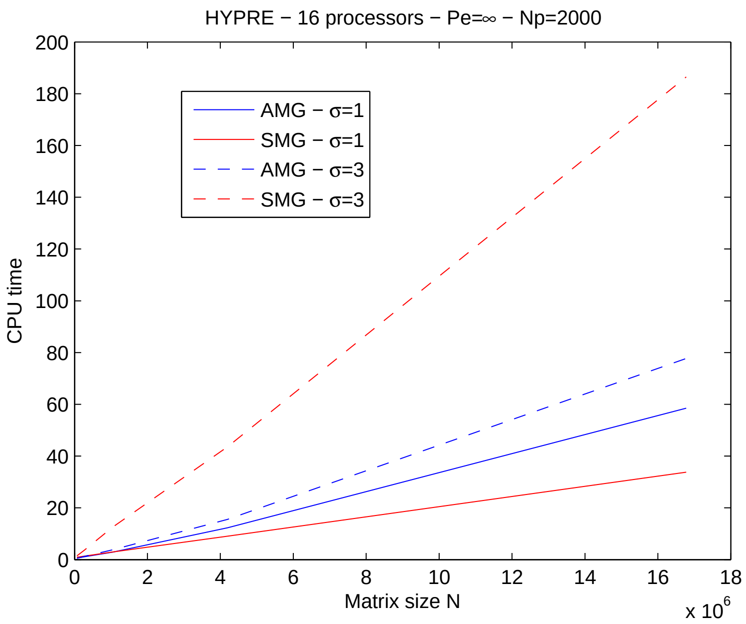


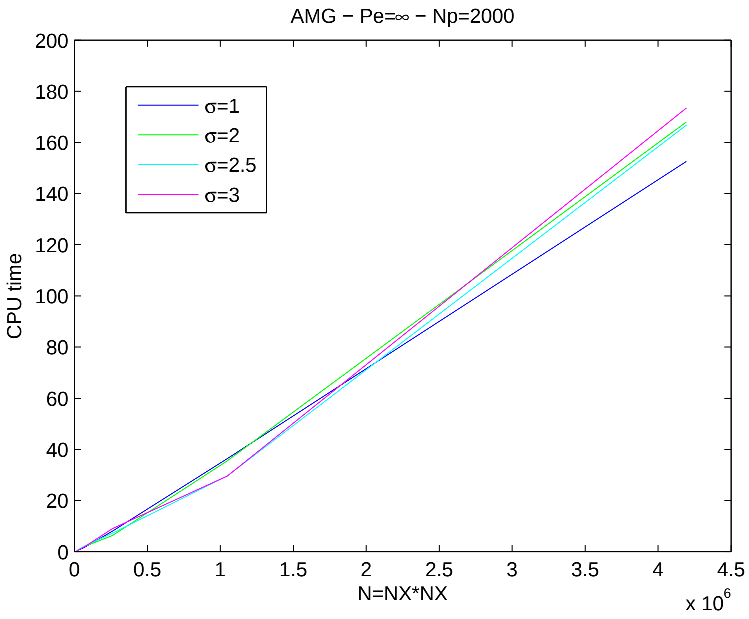


Multigrid for heterogeneous porous media

Geometric Multigrid SMG and Algebraic multigrid AMG

Library Hypre





Flow equations in fractured media

Flow equations in a 3D network of fractures

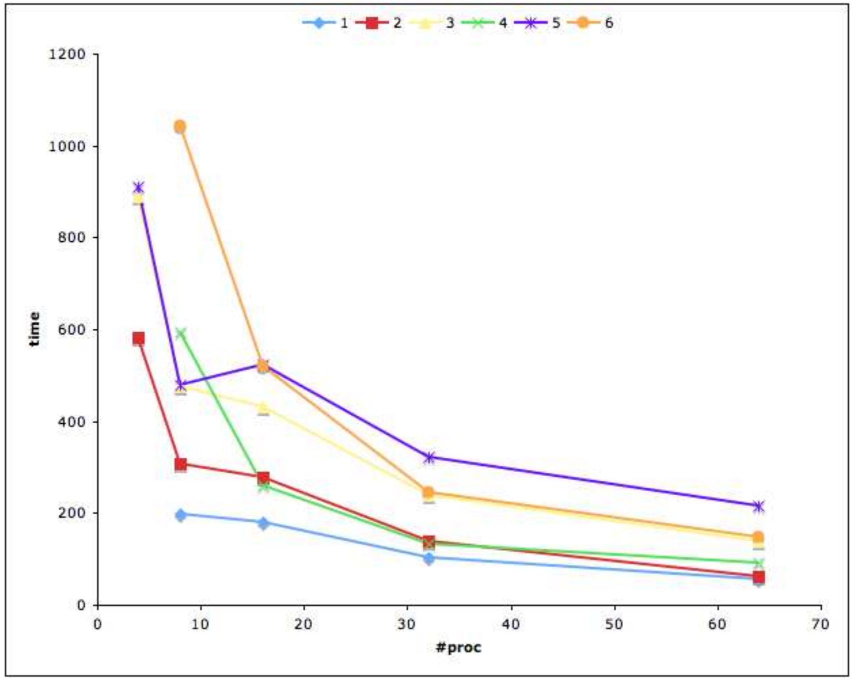
Continuity of pressure and flux through the fracture intersections

Mixed Hybrid Finite Element method with an unstructured mesh

Conforming mesh at intersections

PCG preconditioned by AMG (Boomer-AMG from HYPRE)

MICAS project;
PhD thesis of Baptiste Poirriez;
SAGE team, INRIA, Rennes



A non symmetric

More difficult ...

Not possible to get
a short recurrence together with
a minimisation of the residual

- Minimisation : GMRES
- Short recurrence : Bi-Conjugate Gradient, QMR, etc
- Preconditioning

Arnoldi and GMRES

Arnoldi process

$Span(V_k) = \mathcal{K}_k(A, v_1)$ Basis of Krylov space

$V_k^T V_k = I$ Orthonormal system

$AV_k = V_k H_k + h_{k+1,k} v_{k+1} e_k^T$ H_kHessenberg matrix

GMRES algorithm

$r_0 = \|r_0\|_2 v_1 = \beta v_1$ Krylov space

$x_k = x_0 + V_k y \;\; y \in \mathbb{R}^k$ Space condition

$r_k \perp AV_k$ Galerkin condition

$$r_k = r_0 - AV_k y = V_{k+1}(\beta e_1 - \overline{H}_k y) \quad \overline{H}_k = \begin{pmatrix} H_k \\ h_{k+1,k} e_k^T \end{pmatrix}$$

$$(AV_k)^T r_k = \overline{H}_k^T (\beta e_1 - \overline{H}_k y)$$

Solve $\min_{y \in \mathbb{R}^k} \|\beta e_1 - \overline{H}_k y\|_2$ Galerkin condition

GMRES - Convergence

Polynomial method

$$x_k = x_0 + Q_{k-1}(A) r_0$$
$$r_k = (I - AQ_{k-1}(A)) r_0 = P_k(A) r_0 \text{ avec } P_k(0) = 1$$

Minmax property

IF $A = U \Delta U^{-1}$ with $\Delta = diag(\lambda_1, \ldots, \lambda_n)$

$$\|r_k\|_2 \leq \|r_0\|_2 \kappa(U) \min_{\{P/deg(P)=k\ P(0)=1\}} \max_{1 \leq j \leq n} |P(\lambda_j)|$$

GMRES - restarting

Memory space and complexity

k itérations

Arnoldi : $O(k \times nz) + O(k^2 \times n)$ operations

Least-squares : negligible

Storage of $k + 3$ vectors of length n

Restarted GMRES(m)

Initialisation

choix de x_0

Until convergence

k itérations de GMRES

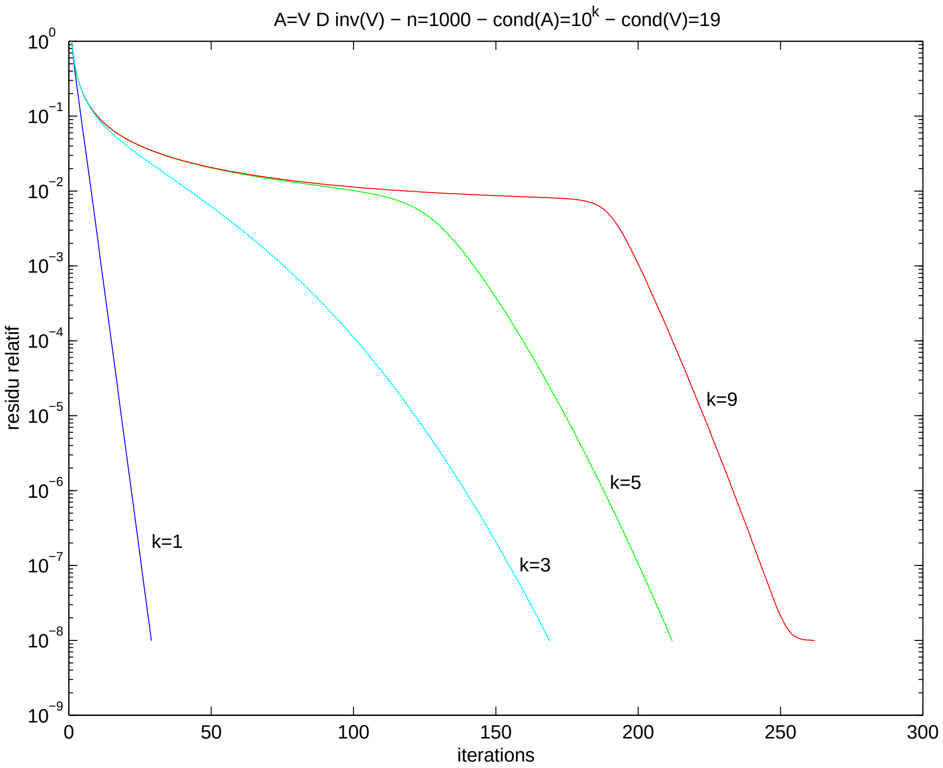
$x_m = x_0 + V_m y$

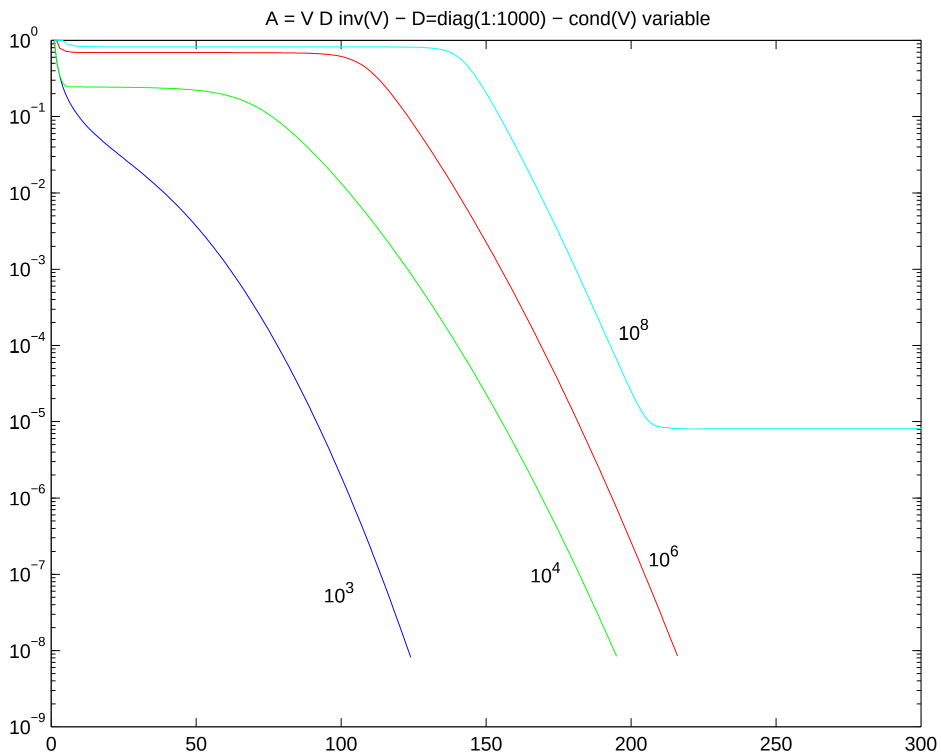
$x_0 = x_m$

End Until

Risk of stagnation

GMRES - examples of convergence





Some Preconditionings

- Jacobi, m-step SOR
- Incomplete Gauss ILU(k) or ILUT
- Approximate inverse
- Multilevel
- Multigrid
- Schwarz

GMRES - algorithm

Initialisation
choose x_0
$r_0 = b - Ax_0$

Until convergence Do
Arnoldi process
  $v_1 = \frac{r_0}{\|r_0\|_2}$
  For $j = 1, m$
    $w = M^{-1}Av_j$
    For $i = 1, j$
      $h_{ij} = v_i^T w$
      $w = w - h_{ij}v_i$
    End For
    $h_{j+1,j} = \|w\|_2$
    $v_{j+1} = w/h_{j+1,j}$
Givens rotations
    $\overline{H}_j = Q_j R_j$
    compute $\|r_j\|_2$
    convergence test
  End For

Least-squares problem
  compute y_m solution of $\min_y (\|r_0\|_2 e_1 - \overline{H}_m y)$
  $x_m = x_0 + V_m y_m$
  $r_m = b - Ax_m$
  convergence test
Restarting
  $x_0 = x_m$
  $r_0 = r_m$
End Do

Parallel Preconditioned GMRES

With Arnoldi process: $M^{-1}AV_m = V_{m+1}\bar{H}_m$
Data dependencies at each operation $w = w - h_{ij}v_i$

With Newton polynomials

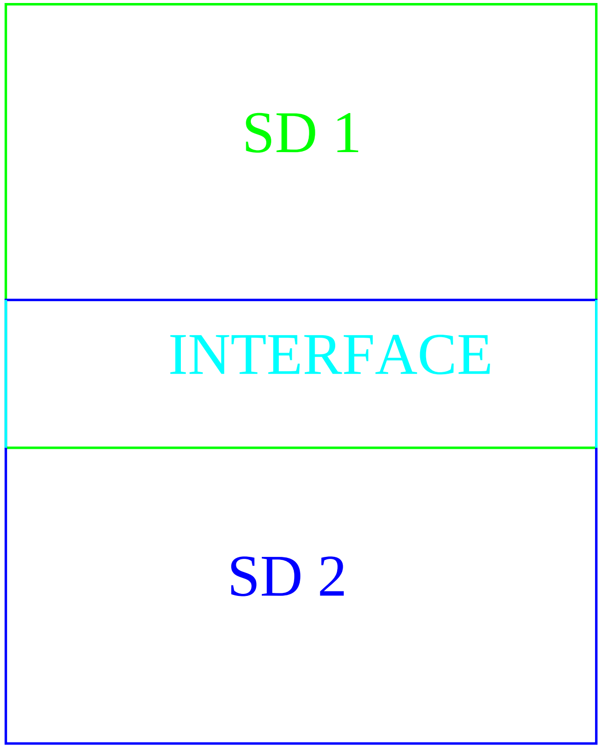
1. Parallel basis computation: $M^{-1}AW_m = W_{m+1}\bar{T}_m$
   - $\sigma_{j+1}w_{j+1} = (M^{-1}A - \lambda_{j+1}I)w_j - \delta_{j+1}w_{j-1}$
   - Scalars σ_j, λ_j and δ_j chosen for stability
   - $M^{-1}AW_m = W_{m+1}\bar{T}_m$

2. Parallel QR factorization: $W_{m+1} = V_{m+1}R_{m+1}$

$$M^{-1}AV_m = V_{m+1}\underbrace{R_{m+1}\bar{T}_m R_m^{-1}}_{\bar{H}_m}$$

Schwarz preconditionings

Domain partition into subdomains with overlapping



p subdomains Ω_i
R_i restriction of Ω in Ω_i
R_i^T extension of Ω_i in Ω
A in Ω_i : $A_i = R_i^T A R_i$
M_i preconditioning of A_i

Additive Schwarz preconditioning

Solving $Mz = r$

Algorithm

$r_1 = R_1 r$
$M_1 z_1 = r_1$
$z^{(1)} = R_1^T z_1$
$r_2 = R_2 r$
$M_2 z_2 = r_2$
$z^{(2)} = R_2^T z_2$
$z = z^{(1)} + z^{(2)}$

Data dependencies

subdomain 2 in parallel with subdomain 1

global addition in z

$$M^{-1} = R_1^T M_1^{-1} R_1 + R_2^T M_2^{-1} R_2$$

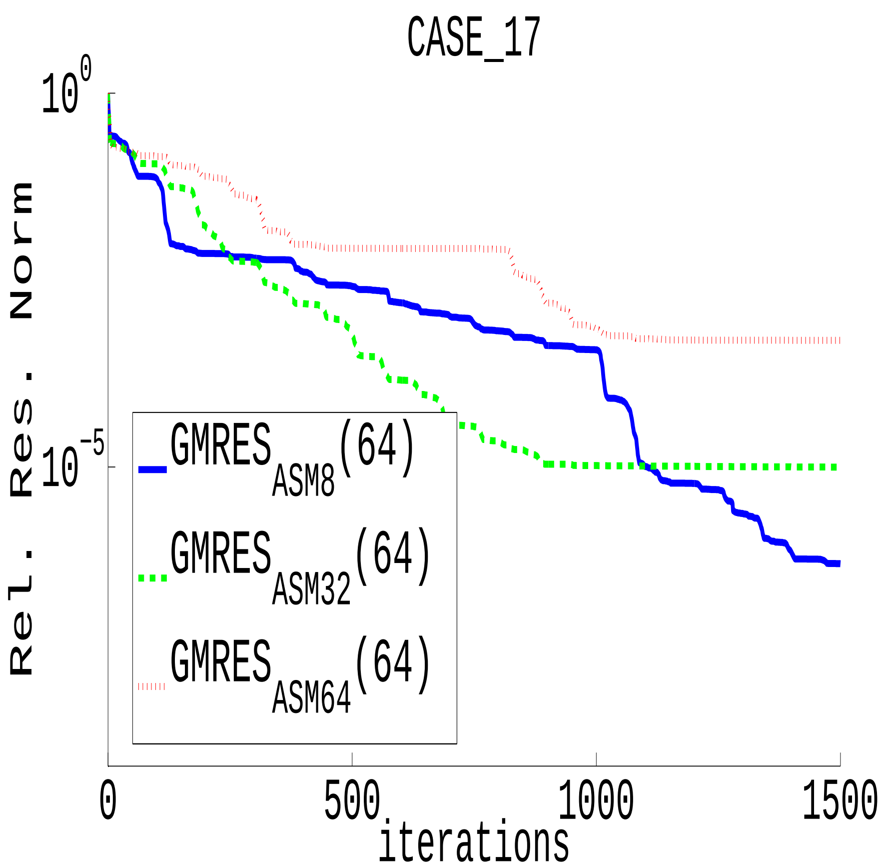
CFD computations:GMRES with Additive Schwarz

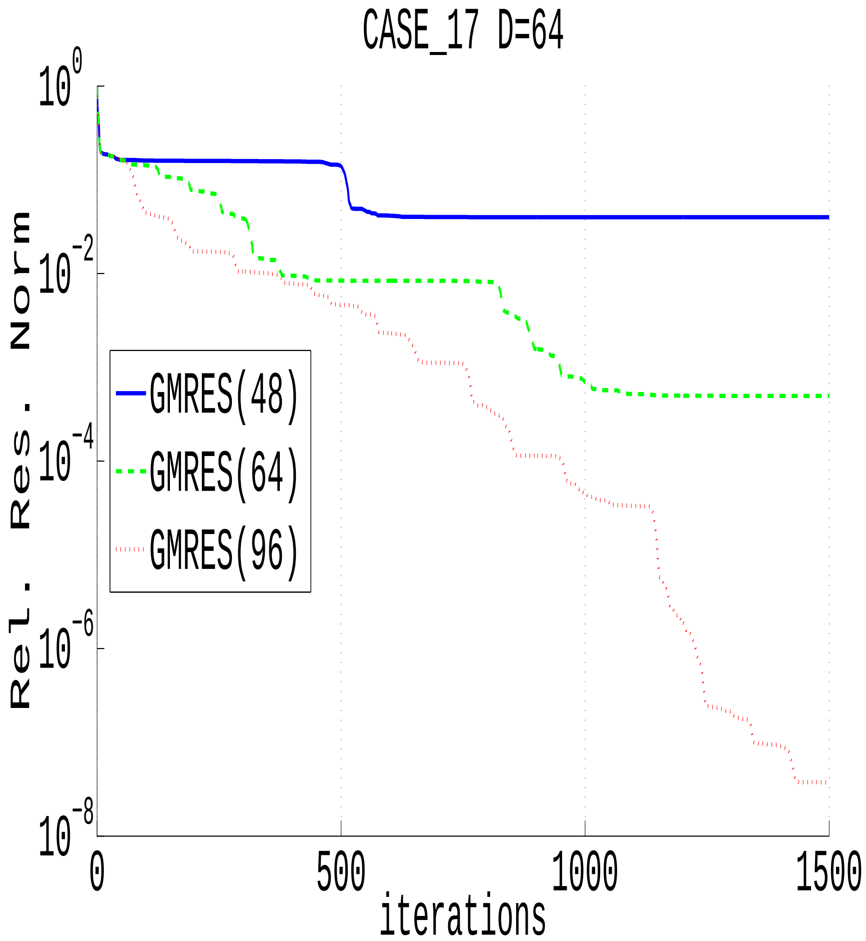
Matrix CASE_17 : n = 381,689; nz = 37,464,962

- Multigrid does not converge;
- GMRES(m) : restarted GMRES; cycle of m iterations;
- Preconditioning by ILU or multigrid converges slowly;
- Preconditioning by Restricted Additive Schwarz; D Subdomains ; MUMPS in subdomains.

LIBRAERO ANR-funded project;
Ph-D thesis of Désiré Nuentsa Wakam; SAGE team, INRIA, Rennes

GMRES(m) with Restricted Additive Schwarz





GMRES with deflation

Preconditioning with approximate invariant subspace

$U = [u_1 \dots u_k]$ basis of the subspace associated with k smallest Ritz values

$\bar{M}^{-1} \equiv I_n + U(|\lambda_n|T^{-1} - I_k)U^T, \quad T = U^T A U$

GMRES(m) applied to $AM^{-1}\bar{M}^{-1}$

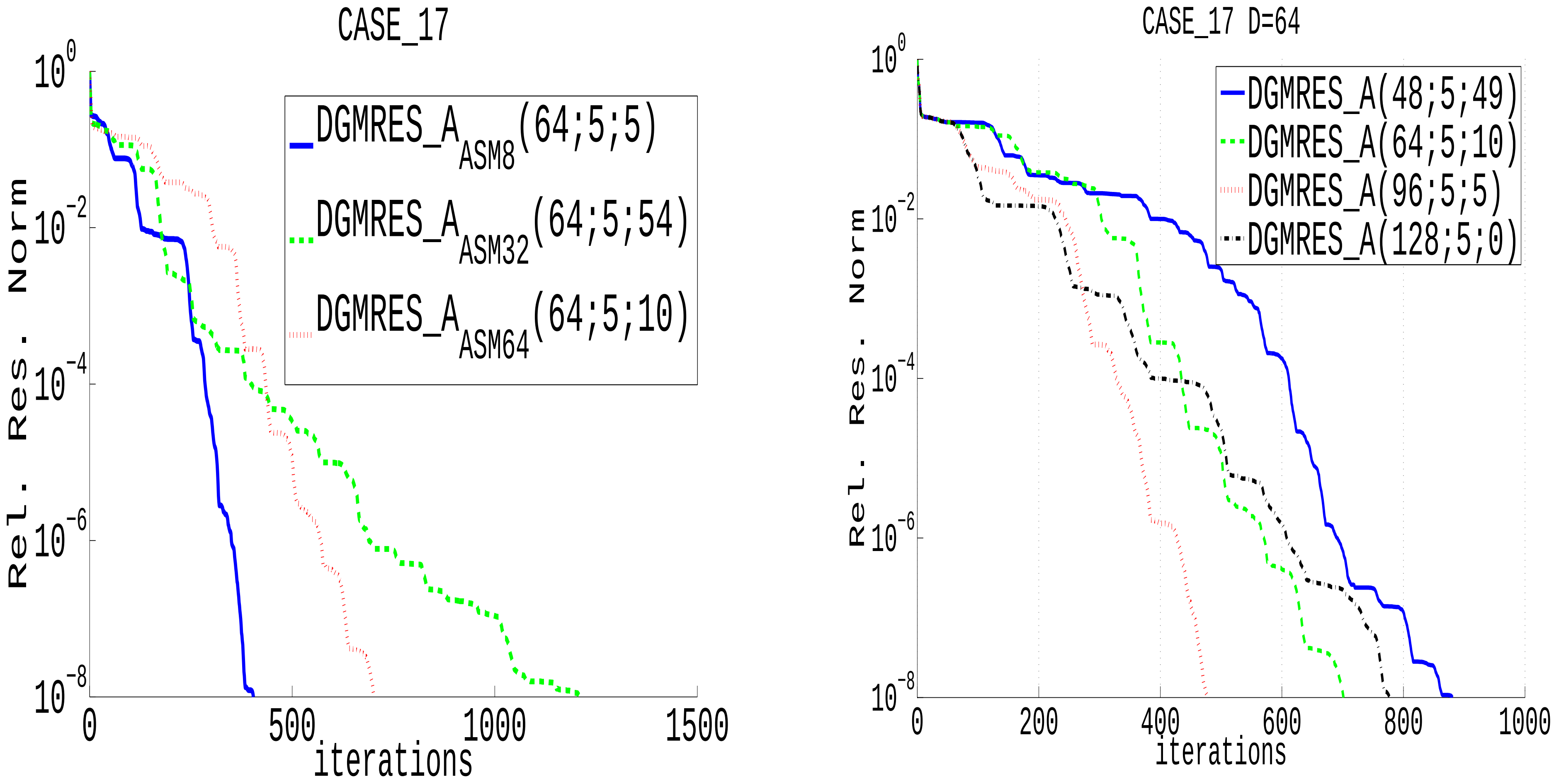
Adaptive deflation

Increase the invariant subspace at each restart

test the rate of convergence and increase if necessary

Deflated GMRES for CFD problems

Matrix CASE_17; GMRES(m) with Restricted Additive Schwarz using D subdomains
DGMRES(m,k,max): k eigenvalues deflated up to max



Conclusion

- Preconditioned Krylov methods efficient in general cases
- Multigrid preconditioning efficient for large problems
- Schwarz preconditioning also efficient
- Schwarz preconditioning can be combined with Aitken acceleration
- Deflation can be combined with any preconditioning

DGMRES: now implemented in Petsc

DefGPREMS: Deflated GMRES Preconditioned by Multiplicative Schwarz