Enlarged Krylov methods applied to industrial problems

H. Al-Daas, O. Tissot

Project advisor
L. Grigori

INRIA-ALPINES

25 August 2016

Outline

Challenge in getting scalable iterative linear solvers

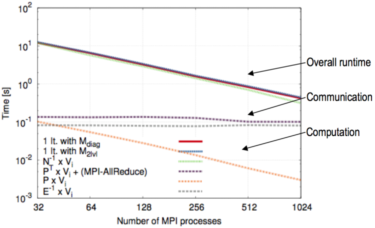
- A Krylov solver finds x_{k+1} from $x_0 + \mathcal{K}_{k+1}(A, r_0)$ where

$$\mathcal{K}_{k+1}(A, r_0) = span\{r_0, Ar_0, A^2 r_0, ..., A^k r_0\},$$

  such that the Petrov-Galerkin condition $b - Ax_{k+1} \perp \mathscr{L}_{k+1}$ is satisfied.
- Does a sequence of k SpMVs to get vectors $[x_1, ..., x_k]$
- Finds best solution x_{k+1} as linear combination of $[x_1, ..., x_k]$

Typically, each iteration requires

- Sparse matrix vector product
  $\rightarrow$ point-to-point communication
- Dot products for orthogonalization
  $\rightarrow$ global communication



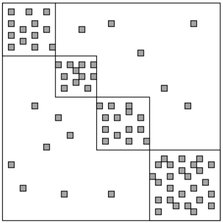
Map making, with R. Stompor, M. Szydlarski
Results obtained on Hopper, Cray XE6, NERSC

CEMRACS project: Enlak

- A way to improve the actual performances of Krylov solvers is to change numerics in order to
  - reduce overall communication
  - increase arithmetic intensity
- Goal of the project: test Enlarged Krylov methods [Grigori et al., 2014] on industrial problems.
- Participants: Hussam Al-Daas (Inria), Yvan Fournier (EDF), Laura Grigori (Inria), Pascal Hénon (Total), Philippe Ricoux (Total), Olivier Tissot (Inria).

Enlarged Krylov methods [Grigori et al., 2014]

- Partition the matrix into t domains
- Split the residual r_k 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

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

- Search for the solution of the system $Ax = b$ in $\mathscr{K}_{t,k+1}(A, r_0)$

Properties of enlarged Krylov subspaces

- The Krylov subspace $\mathcal{K}_{k+1}(A, r_0)$ is a subset of the enlarged one

$$\mathcal{K}_{k+1}(A, r_0) \subset \mathscr{K}_{t,k+1}(A, r_0)$$

- For all $k < k_{max}$ the dimensions of $\mathscr{K}_{t,k}$ and $\mathscr{K}_{t,k+1}$ are stricltly increasing by some number i_k and i_{k+1} respectively, where

$$t \geq i_k \geq i_{k+1} \geq 1.$$

- The enlarged subspaces are increasing subspaces, yet bounded.

$$\mathscr{K}_{t,1}(A, r_0) \subsetneq ... \subsetneq \mathscr{K}_{t,k_{max}-1}(A, r_0) \subsetneq \mathscr{K}_{t,k_{max}}(A, r_0) = \mathscr{K}_{t,k_{max}+q}(A, r_0), \forall q > 0$$

Outline

Introduction

Enlarged Krylov methods

Numerical experiments

Enlarged Conjugate Gradient[1]

Algorithm 1 Classic CG

$r_0 = b - Ax_0$
$p_1 = \frac{r_0}{\sqrt{r_0^t A r_0}}$
while $||r_{k-1}||_2 > \varepsilon ||b||_2$ **do**
    $\alpha_k = p_k^t r_{k-1}$
    $x_k = x_{k-1} + p_k \alpha_k$
    $r_k = r_{k-1} - A p_k \alpha_k$
    $p_{k+1} = r_k - p_k (p_k^t A r_k)$
    $p_{k+1} = \frac{p_{k+1}}{\sqrt{p_{k+1}^t A p_{k+1}}}$
end while

BLAS 1&2 operations

\# messages per iteration

O(1) from SpMV +

O(log P) from dot prod + norm

Algorithm 2 EK-CG

$R_0 = T(b - Ax_0)$
$P_1 =$ A-orthonormalize(R_0)
while $|| \sum_{i=1}^{t} R_k^{(i)} ||_2 < \varepsilon ||b||_2$ **do**
    $\alpha_k = P_k^t R_{k-1}$ $\triangleright\ t \times t$
    $X_k = X_{k-1} + P_k \alpha_k$ $\triangleright\ n \times t$
    $R_k = R_{k-1} - A P_k \alpha_k$ $\triangleright\ n \times t$
    $P_{k+1} = AP_k - P_k (P_k^t A A P_k) - P_{k-1} (P_{k-1}^t A A P_k)$ $\triangleright\ n \times t$
    $P_{k+1} =$ A-orthonormalize(P_{k+1})
end while
$x = \sum_{i=1}^{t} X_k^{(i)}$ $\triangleright\ n \times 1$

BLAS 3 operations

\# messages per iteration

O(1) from SpMV +

O(log P) from BCGS + A-ortho

[1] Paper in preparation L.Grigori and OT.

Enlarged GMRES[2]

Algorithm 3 GMRES

$r_0 = b - Ax_0$
for $i = 1$ to k **do**
    $p = Ap_{i-1}$
    $\tilde{p}_i \leftarrow \text{BCGS}\ (\tilde{p}_i, p_0, \ldots, p_{i-1})$
    $p_i = \tilde{p}_i / \|\tilde{p}_i\|_2$
    update H
end for
solve LSQ problem with H

BLAS 1&2 operations

\# messages per iteration
O(1) from SpMV +
O(log P) from BCGS + norm

Algorithm 4 Enlarged GMRES

$r_0 = b - Ax_0,\ R_0 = T(r_0)$
for $i = 1$ to k **do**
    $\tilde{P}_i = AP_{i-1}$
    $\tilde{P}_i \leftarrow \text{BCGS}\ (\tilde{P}_i, P_0, \ldots, P_{i-1})$
    $[P_i, L] = TSQR(\tilde{P}_i)$
    update H
end for
solve LSQ problem with H

BLAS 3 operations

\# messages per iteration
O(1) from SpMV +
O(log P) from BCGS + TSQR

[2] Paper in preparation H. Al Daas, L. Grigori, P. Hénon

Dynamic reduction of search directions

In Krylov subspace methods we have the following relation at iteration k

$$\tilde{P}_{k+1}\alpha_{k+1} = (I - V_k V_k^\top)R_k$$

where $\tilde{P}_{k+1}$ are the new search directions, and

$$\alpha_{k+1} = \tilde{P}_{k+1}^\top R_k$$

So $Rank(\alpha_{k+1}) = Rank(R_k)$ by construction of the Krylov basis.
To select only adding-value search directions we write the truncated SVD decomposition:

$$\alpha_{k+1} \approx U_{k+1}^{+}\Sigma_{k+1}^{+}W_{k+1}^{+}$$

The new search directions are given by the relation:

$$P_{k+1} = \tilde{P}_{k+1}U_{k+1}^{+}$$

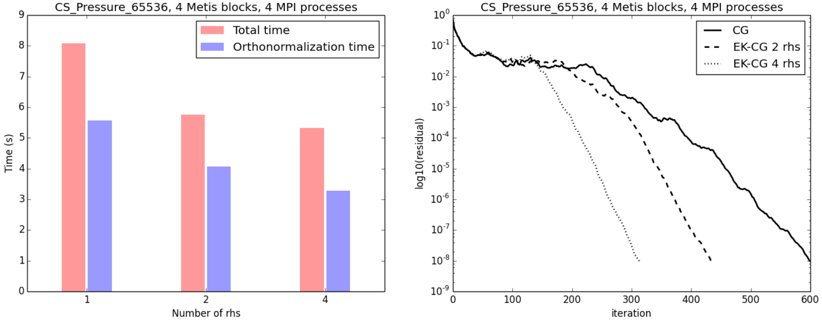
.

Outline

Work done

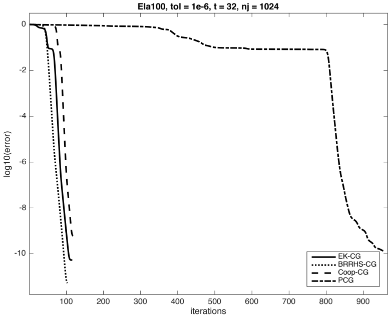
- Implementation of the methods in C/MPI
  1. We started from a homemade library written by S. Cayrols for sparse matrices.
  2. We added dense matrices in that library.
  3. We linked with MKL for the local linear algebra part.
  4. We implemented 3 different Block CG-like methods (EK, Coop and BRRHS) and 2 different A-orthonormalization algorithm (Orthomin and Orthodir).
  5. We almost finalized implementing Block GMRES (difficulties in updating block Hessenberg matrix).
- Validation of the CG prototype.
- Test on EDF matrices (coming from *Code_Saturne*).
- Numerical experiments of EGMRES with CPR on Matlab (reservoir simulation).

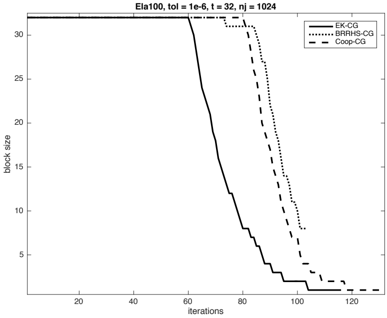
Numerical experiments: EK-CG



- C/MPI prototype on laptop (4 cores)
- Matrix of size 65 536 with 439 552 non zero elements coming from *Code_Saturne*

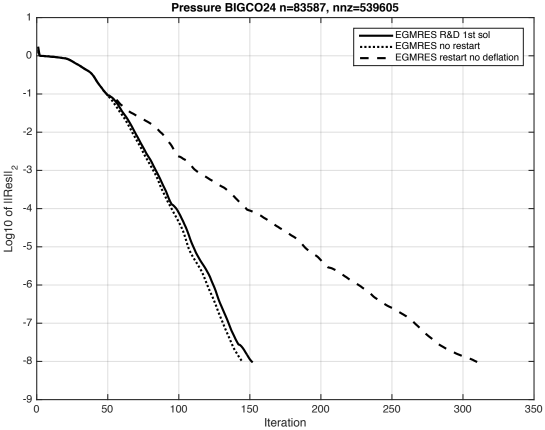
Numerical experiments: EK-CG size reduction





- Matlab results
- Elasticity matrix of size 36 663 with 1 231 497 non zero elements
- Block Jacobi preconditioner

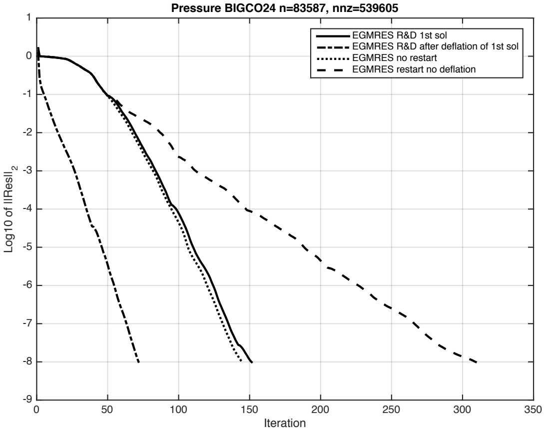
Numerical experiments: EGMRES



Method:

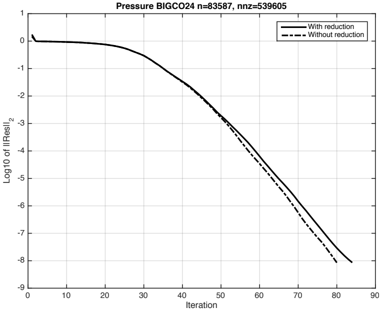
- Solve $Ax = b1$ with EGMRES R&D with max subspace dimension $= 400$.
- Solve $Ax = b1$ with EGMRES no restart.

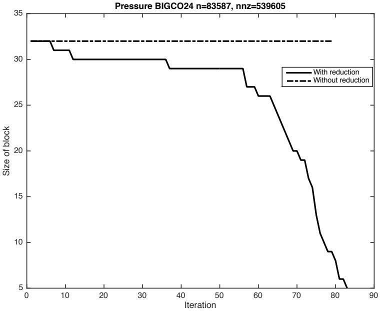
Numerical experiments: EGMRES



- Solve $Ax = b1$ with EGMRES R&D with max subspace dimension $= 400$.
- Solve $Ax = b2$ with EGMRES R&D using deflation info from the last solution.

Numerical experiments: EGMRES size reduction





Ongoing work

- Finalize EGMRES code.
- Add preconditioner.
- Dynamically decrease the number of search directions during the iterations.
- Deflate eigenvalues when solving the same system with different rhs each given in a time.
- Test on bigger matrices.

Thank you for your attention!

Questions?

References (1)

Grigori, L., Moufawad, S., and Nataf, F. (2014).
Enlarged Krylov Subspace Conjugate Gradient Methods for Reducing Communication.
Technical Report 8597, INRIA.