Preconditioners for Krylov solvers in data assimilation
(for oceanic and atmospheric applications)

S. Gratton[1] P. Laloyaux[2] A. Sartenaer[2]
J. Tshimanga[3] A. Weaver[3]

[1]ENSEEIHT, Toulouse, France
[2]FUNDP, Namur, Belgium
[3]CERFACS, Toulouse, France

Toulouse, February 2010

Outline

Outline

Linear systems in sequence

Let

- A: symmetric and positive definite matrix of order n
- $b_1, \ \ldots, \ b_r \in \mathbb{R}^n$: right-hand sides available in sequence

Solve in sequence:

- $Ax = b_1, \ Ax = b_2, \ldots$ by an iterative method (Krylov solvers)
- Preconditioning each system using information obtained during the solution of the previous system(s)

$\rightarrow$ Extend the idea to the case where A varies along the iterations (Gauss-Newton method – variational ocean data assimilation)

Preconditioning technique

- Solve $Ax = b_1$ and extract information info_1
- Solve $Ax = b_2$ using info_1 to precondition and extract information info_2
- Solve $Ax = b_3$ using info_2 (and possibly info_1) to precondition and extract information info_3
- ...

where info_k contains (in our case):

- Descent directions p_i
- Ritz pairs (θ_i, z_i) (approximations to eigenpairs)

produced by a conjugate gradient algorithm (or an equivalent Lanczos process)

Conjugate gradient (CG) method

$\rightarrow$ Solves $\min_{x\in\mathbb{R}^n} \frac{1}{2}x^T Ax - b^T x$ or equivalently $Ax = b$

- Given x_0, set $r_0 \leftarrow Ax_0 - b$, $p_0 \leftarrow -r_0$, $k \leftarrow 1$
- Loop on k

$$
\begin{array}{rcll}
\alpha_{k-1} & \leftarrow & \dfrac{r_{k-1}^T r_{k-1}}{{p_{k-1}}^T A p_{k-1}} & \text{Compute the step length} \\
x_k & \leftarrow & x_{k-1} + \alpha_{k-1} p_{k-1} & \text{Update the iterate} \\
r_k & \leftarrow & r_{k-1} + \alpha_{k-1} A p_{k-1} & \text{Update the residual} \\
\beta_k & \leftarrow & \dfrac{r_k^T r_k}{r_{k-1}^T r_{k-1}} & \text{Ensure } A\text{-conjugate directions} \\
p_k & \leftarrow & -r_k + \beta_k p_{k-1} & \text{Update the descent direction}
\end{array}
$$

Descent directions

- are A-conjugate:

$$p_i^T A p_j \begin{cases} > 0 & \text{if} \quad i = j \\ = 0 & \text{if} \quad i \neq j \end{cases}$$

- belong to and span the Krylov subspace:

$$\mathcal{K}(A, r_0, k) = \text{span}\{r_0, Ar_0, \ldots, A^{k-1} r_0\}$$

Lanczos method

$\rightarrow$ Related to the CG method but allows in addition to approximate eigenpairs belonging to the Krylov subspace $\mathcal{K}(A, r_0, k)$:

- Builds an orthonormal basis $Q = [q_1, \ldots, q_k]$ of $\mathcal{K}(A, r_0, k)$ where

$$q_1 = r_0 / \|r_0\|$$

- Uses the Galerkin approach, i.e., computes (θ, z) such that

$$Q^T (Az - \theta z) = 0$$

Ritz pairs

Compute (θ, z) such that

$$Q^T(Az - \theta z) = 0$$

is equivalent to compute (θ, y) such that

$$Q^T AQy = \theta y \qquad (z = Qy, \;\; y \in \mathbb{R}^k, \;\; Q^T Q = I_n)$$

Definition

If (θ_i, y_i) solves the eigenproblem above, then the pair

$$(\theta_i, z_i) \quad \text{with} \quad z_i = Qy_i$$

is called Ritz pair of A w.r.t. the considered Krylov subspace

- Ritz-vectors are orthonormal and A-conjugate

Outline

From BFGS

The BFGS updating formula (inverse Hessian):

$$H_k = \left(I_n - \frac{y_k s_k^T}{y_k^T s_k}\right)^T H_{k-1} \left(I_n - \frac{y_k s_k^T}{y_k^T s_k}\right) + \frac{s_k s_k^T}{y_k^T s_k}$$

where $s_k = x_k - x_{k-1}$ and $y_k = \nabla f(x_k) - \nabla f(x_{k-1})$

applied to $f(x) = \frac{1}{2}x^T A x - b^T x$ and with

$$s_k = x_k - x_{k-1} = \alpha_{k-1} p_{k-1} \text{ (CG step)} \quad \text{and} \quad y_k = r_k - r_{k-1} = A s_k$$

writes

$$H_k = \left(I_n - \sum_{i=1}^{k} \frac{s_i s_i^T}{s_i^T A s_i} A\right) H_0 \left(I_n - \sum_{i=1}^{k} A \frac{s_i s_i^T}{s_i^T A s_i}\right) + \sum_{i=1}^{k} \frac{s_i s_i^T}{s_i^T A s_i}$$

To a general form

Letting $S = [s_1, \ldots, s_k]$, since $S^T AS$ is diagonal,

$$H_k = \left(I_n - \sum_{i=1}^{k} \frac{s_i s_i^T}{s_i^T A s_i} A \right) H_0 \left(I_n - \sum_{i=1}^{k} A \frac{s_i s_i^T}{s_i^T A s_i} \right) + \sum_{i=1}^{k} \frac{s_i s_i^T}{s_i^T A s_i}$$

writes

$$H_k = \left[I_n - S(S^T AS)^{-1} S^T A \right] H_0 \left[I_n - AS(S^T AS)^{-1} S^T \right] + S(S^T AS)^{-1} S^T$$

$\rightarrow$ How good is this matrix when used as a preconditioner with S containing a limited number of:

- CG directions ?
- or general A-conjugate directions (such as Ritz vectors) ?
- or even any set of vectors such that $S^T AS$ is nonsingular ?

General LMP formulation

[Gratton, Sartenaer, Tshimanga, submitted to SIOPT]

Definition

- Let A and M be symmetric positive definite matrices of order n
- Let S be any n by k matrix of rank k, with $k \leq n$

The symmetric matrix:

$$H = \left[I_n - S(S^T AS)^{-1}S^T A\right] M \left[I_n - AS(S^T AS)^{-1}S^T\right] + S(S^T AS)^{-1}S^T$$

is called the Limited Memory Preconditioner (LMP)

$M \equiv$ first-level preconditioner $\quad$ $H \equiv$ second-level preconditioner

Elementary properties of the LMP

$$H = \left[I_n - S(S^TAS)^{-1}S^TA\right] M \left[I_n - AS(S^TAS)^{-1}S^T\right] + S(S^TAS)^{-1}S^T$$

Proposition

- H is *symmetric* and *positive definite*
- H is *invariant* under a *change of basis* for the *columns of* S ($S \leftarrow Z = SX$, X nonsingular)
- $H = A^{-1}$ if S is of order n ($k = n$)
- (Possibly cheap) factored form: $H = GG^T$ with

$$G = L - SR^{-1}R^{-T}S^TAL + SR^{-1}X^{-T}S^TL^{-T}$$

where

- $M = LL^T$ (L of order n)
- $S^TAS = R^TR$ (R of order k)
- $S^TL^{-T}L^{-1}S = X^TX$ (X of order k)

Connection with the existing L-BFGS form

(Let $M = I_n$)

- Using $Y = AS$ and letting $B = Y^T S = S^T AS$ we have:

$$H = \left[I_n - SB^{-1}Y^T \right] \left[I_n - YB^{-1}S^T \right] + \; SB^{-1}S^T$$

- Letting $R = \texttt{triu}(B)$ and $D = \texttt{diag}(B)$, the classical L-BFGS update reads [Gilbert, Nocedal, 1993], [Byrd, Nocedal, Schnabel, 1994]:

$$\left[I_n - SR^{-T}Y^T \right] \left[I_n - YR^{-1}S^T \right] + \; SR^{-T}DR^{-1}S^T$$

This last formula is not invariant under transformations of S

Construction cost

$$H = \left[I_n - S(S^TAS)^{-1}S^TA\right] M \left[I_n - AS(S^TAS)^{-1}S^T\right] + \; S(S^TAS)^{-1}S^T$$

Let $S \leftarrow Z = SX$ (X nonsing.) with $Z^TAZ = I_n$, by invariance property:

$$H = \left(I_n - ZZ^TA\right) M \left(I_n - AZZ^T\right) + ZZ^T$$

or, if $Y = AZ$,

$$H = \left(I_n - ZY^T\right) M \left(I_n - YZ^T\right) + ZZ^T$$

$\rightarrow$ Step-by-step construction of Z (Gram-Schmidt) and Y:

- k matrix-vector products by A
- $\pm\ 3k^2n$ additional flops

Application cost

$$Hq = \left(I_n - ZY^T\right) M \left(I_n - YZ^T\right) q + ZZ^T q$$

Computation of $r = Hq$

1. $f = Z^T q$ (costs $2kn$ flops)
2. $\bar{r} = M(q - Yf)$ (costs $2kn$ flops and one product by M)
3. $r = \bar{r} - Z(Y^T \bar{r} - f)$ (costs $4kn$ flops)

$\rightarrow$ one matrix-vector product by M and $8kn$ additional flops

Eigenvalues clusterization effect of the LMP

Proposition

- *Let* $0 < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ *denote the eigenvalues of* MA
- *Let* $\mu_1, \ldots, \mu_n$ *denote the eigenvalues of* HA

Then $\{\mu_1, \ldots, \mu_n\}$ *can be split in two subsets:*

$$\begin{cases} \mu_j = 1 & \text{for } j \in \{n-k+1, \ldots, n\} \\ \lambda_j \leq \mu_j \leq \lambda_{j+k} & \text{for } j \in \{1, \ldots, n-k\} \end{cases}$$

and

$$\kappa(HA) \leq \frac{\max\{1, \lambda_n\}}{\min\{1, \lambda_1\}}$$

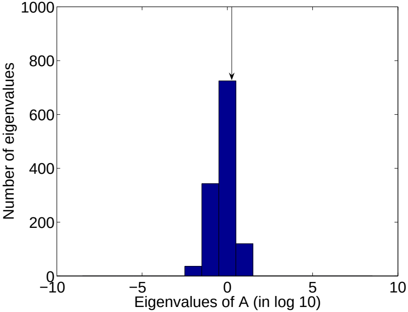
$\rightarrow$ At least k eigenvalues are clustered at 1

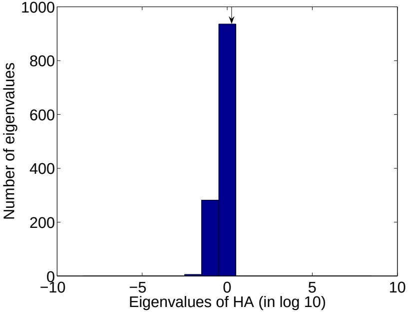
$\rightarrow$ The remaining part of the spectrum does not expand

Illustration

- $A =$ incomplete Cholesky factorization of the BCSSTK 27 matrix (Harwell-Boeing Collection)
- $n = 1224$
- $\lambda_{\min}(A) = 0.007$ and $\lambda_{\max}(A) = 36.0$
- $M = I_n$
- S constructed with 300 randomly generated vectors

$\longrightarrow$ Comparison of the eigen-distribution of A and HA





→ Part of the spectrum has been shifted to 1

→ The remaining part of the spectrum does not expand

Three particular cases

Remember that if the vectors in $S = [s_1, \ \dots, \ s_k]$ are A-conjugate, then:

$$H = \left[I_n - S(S^TAS)^{-1}S^TA\right] M \left[I_n - AS(S^TAS)^{-1}S^T\right] + \ S(S^TAS)^{-1}S^T$$

simplifies to:

$$H = \left(I_n - \sum_{i=1}^{k} \frac{s_i s_i^T}{s_i^T A s_i} A\right) M \left(I_n - \sum_{i=1}^{k} A \frac{s_i s_i^T}{s_i^T A s_i}\right) + \sum_{i=1}^{k} \frac{s_i s_i^T}{s_i^T A s_i}$$

First case: the quasi-Newton LMP

Take $s_i = p_i, \;\; i = 1, \ldots, k$, the descent directions generated by a CG method

Proposition

The LMP built with $M = H_0$ and $S = [p_1, \ldots, p_k]$ writes

$$H_k = \left(I_n - \frac{p_k p_k^T A}{p_k^T A p_k} \right) H_{k-1} \left(I_n - \frac{A p_k p_k^T}{p_k^T A p_k} \right) + \frac{p_k p_k^T}{p_k^T A p_k}$$

$\rightarrow$ Amounts to the preconditioner proposed by Morales and Nocedal, 2000

Second case: the spectral-LMP

Take $s_i = v_i, \;\; i = 1, \ldots, k$, eigenvectors of A (associated to λ_i)

Proposition

The LMP built with $M = I_n$ and $S = [v_1, \ldots, v_k]$ writes

$$H = \prod_{i=1}^{k} \left[I_n - \left(1 - \frac{1}{\lambda_i} \right) v_i v_i^T \right]$$

- → Amounts to the preconditioner proposed by Fisher, 1998
- → Daily used in operational data assimilation systems but with Ritz pairs (θ_i, z_i) to approximate eigenpairs (λ_i, v_i):

$$\tilde{H} = \prod_{i=1}^{k} \left[I_n - \left(1 - \frac{1}{\theta_i} \right) z_i z_i^T \right]$$

- → "Inexact spectral-LMP" (!!! no more a member of our class of LMP !!!)

Third case: the Ritz-LMP

Take $s_i = z_i$, $i = 1, \dots, k$, Ritz vectors of A (associated to θ_i)

Proposition

The LMP built with $M = I_n$ and $S = [z_1, \dots, z_k]$ writes

$$H = \prod_{i=1}^{k} \left[I_n - \left(1 - \frac{1}{\theta_i} - \omega_i^2 \right) z_i z_i^T - \omega_i (z_i q_{k+1}^T + q_{k+1} z_i^T) \right]$$

where q_{k+1} is a Lanczos vector and

$$|\omega_i| = \frac{\|A z_i - \theta_i z_i\|}{\theta_i}$$

for $i = 1, \dots, k$

$\longrightarrow$ New preconditioner !!!

Inexact spectral-LMP versus Ritz-LMP

Compare

$$\tilde{H} = \prod_{i=1}^{k} \left[I_n - \left(1 - \frac{1}{\theta_i} \right) z_i z_i^T \right]$$

with

$$H = \prod_{i=1}^{k} \left[I_n - \left(1 - \frac{1}{\theta_i} - \omega_i^2 \right) z_i z_i^T - \omega_i (z_i q_{k+1}^T + q_{k+1} z_i^T) \right]$$

$\rightarrow$ The Ritz-LMP is an enriched version of the inexact spectral-LMP

$\rightarrow$ The Ritz-LMP uses only one more vector than the inexact spectral-LMP

Proposition

$$\|\tilde{H} - H\|_2 \leq k \left(\max_i(\omega_i^2) + \max_i(|\ \omega_i|) \right)$$

where

$$|\omega_i| = \frac{\|Az_i - \theta_i z_i\|}{\theta_i}$$

for $i = 1, \ldots, k$

$\longrightarrow$ The smaller the error in the Ritz pairs, the closer the inexact spectral-LMP to the Ritz-LMP

Quasi-Newton LMP versus Ritz-LMP

Compare

$$H_k = \left(I_n - \frac{p_k p_k^T A}{p_k^T A p_k} \right) H_{k-1} \left(I_n - \frac{A p_k p_k^T}{p_k^T A p_k} \right) + \frac{p_k p_k^T}{p_k^T A p_k}$$

with

$$H = \prod_{i=1}^{k} \left[I_n - \left(1 - \frac{1}{\theta_i} - \omega_i^2 \right) z_i z_i^T - \omega_i (z_i q_{k+1}^T + q_{k+1} z_i^T) \right]$$

→ The quasi-Newton LMP is about twice more expensive in storage than the Ritz-LMP

- CG descent directions and Ritz vectors span the same Krylov subspace

- The LMP H is invariant under transformations of S

Proposition

The Ritz-LMP and the quasi-Newton LMP are analytically equivalent when they are constructed with all available information (descent directions or Ritz vectors) from a CG-like method run on a same matrix

Comparison of the three particular cases:

- Quasi-Newton LMP
- Inexact spectral-LMP
- Ritz-LMP

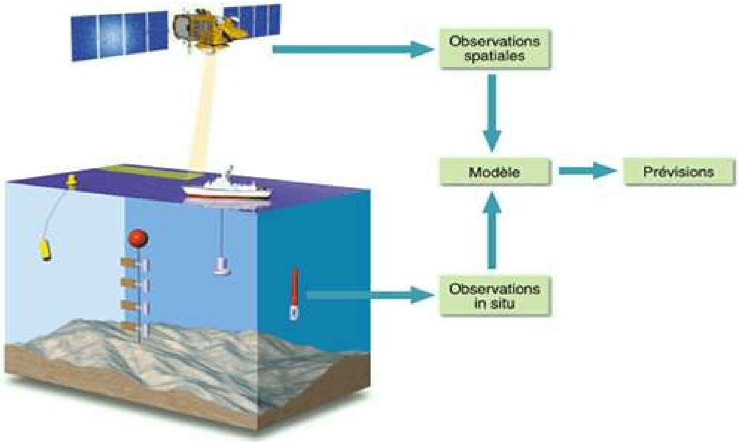
on a realistic data assimilation system

Outline

1. General framework
2. A class of Limited Memory Preconditioners (LMP)
3. Application to variational ocean data assimilation
4. Further improvements

Motivation

- Numerical forecast is performed by integrating PDE describing the model of evolution of the system of interest (atmosphere, ocean, etc.)
- An important part of forecast systems is data assimilation which combines observations and model equations to produces the “best” initial condition
- Data assimilation belongs to the class of nonlinear least-squares problems
- Our interest: improve some optimization software involved in data assimilation in oceanography


Observations spatiales
Modèle
Prévisions
Observations in situ

Four-Dimensional Variational (4D-Var) formulation

→ Very large-scale nonlinear weighted least-squares problem:

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2}||x - x_b||^2_{B^{-1}} + \frac{1}{2}\sum_{j=0}^{N} ||\mathcal{H}_j(\mathcal{M}_j(x)) - y_j||^2_{R_j^{-1}}$$

where:

- Size of real (operational) problems: $x, x_b \in \mathbb{R}^{10^6}$, $y_j \in \mathbb{R}^{10^5}$
- The observations y_j and the background x_b are noisy
- $\mathcal{M}_j$ are model operators (nonlinear)
- $\mathcal{H}_j$ are observation operators (nonlinear)
- B is the covariance background error matrix
- R_j are covariance observation error matrices

Incremental 4D-Var

Let rewrite the problem as:

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2} ||\rho(x)||_2^2$$

Incremental 4D-Var is an inexact/truncated Gauss-Newton algorithm:

- It linearizes ρ around the current iterate $\tilde{x}$ and solves

  $$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|\rho(\tilde{x}) + J(\tilde{x})(x - \tilde{x})\|_2^2$$

  where $J(\tilde{x})$ is the Jacobian of $\rho(x)$ at $\tilde{x}$

- It thus solves a sequence of linear systems (normal equations)

  $$J^T(\tilde{x}) J(\tilde{x})(x - \tilde{x}) = -J^T(\tilde{x}) \rho(\tilde{x})$$

  where the matrix is symmetric positive definite and varies

First-level preconditioner

$$\left(f(x) = \frac{1}{2}||\rho(x)||_2^2 = \frac{1}{2}||x - x_b||_{B^{-1}}^2 + \frac{1}{2}\sum_{j=0}^{N}||\mathcal{H}_j(\mathcal{M}_j(x)) - y_j||_{R_j^{-1}}^2 \right)$$

At the background x_b:

$$J^T(x_b)J(x_b) = B^{-1} + \sum_{j=0}^{N}\mathbf{M}_j^T\mathbf{H}_j^T R_j^{-1}\mathbf{H}_j\mathbf{M}_j$$

Choosing $M = B^{1/2}(B^{1/2})^T$ as first-level preconditioner yields:

$$(B^{1/2})^T J^T(x_b)J(x_b)B^{1/2} = I_n + \sum_{j=0}^{N}(B^{1/2})^T\mathbf{M}_j^T\mathbf{H}_j^T R_j^{-1}\mathbf{H}_j\mathbf{M}_j B^{1/2} \quad (= A_0)$$

$\rightarrow$ Large amount of eigenvalues already clustered at 1

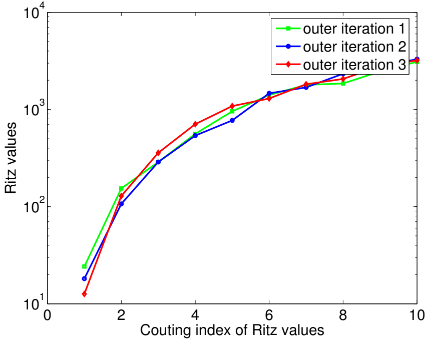
The framework

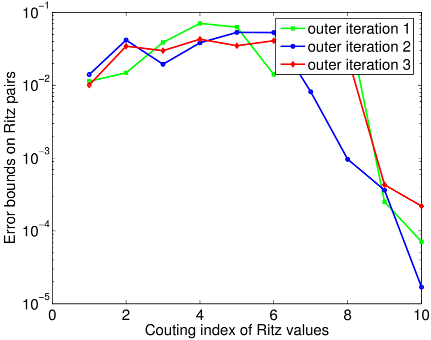
[Tshimanga, Gratton, Weaver, Sartenaer, QJRMS, 2007]

- A realistic outer/inner loop configuration is considered:
  - 3 outer loops of Gauss-Newton (linearization)
  - 10 inner loops of conjugate gradient (on each of the 3 systems)
- The performance is measured by the value of the quadratic cost function
- The convergence of Ritz pairs is measured by the backward errors:

$$\frac{\|Az_i - \theta_i z_i\|}{\|A\|\|z_i\|}$$

Unpreconditioned runs





→ The Ritz values for the three matrices are close together

→ The extremal Ritz pairs have the smallest backward errors (better approx.)

Preconditioned runs

We consider the three forms:

- Quasi-Newton LMP
- Inexact spectral-LMP
- Ritz-LMP

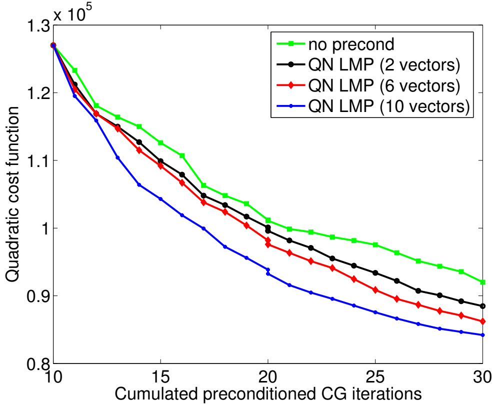
In order to

- Analyse, for each, the effect of increasing the number of vectors in S (second and third systems)
- Compare their performance (second system)

To this aim, an unpreconditioned conjugate gradient is run on the first system to produce 10 vectors from which 2, 6 and 10 relevant ones are selected:

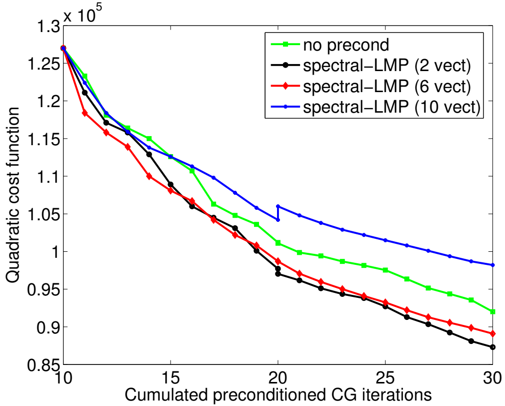
- Ritz-vectors are selected according to their convergence
- Descent directions are selected as the latest ones

Quasi-Newton LMP



$\rightarrow$ Positive impact of an increase in the number of vectors in S

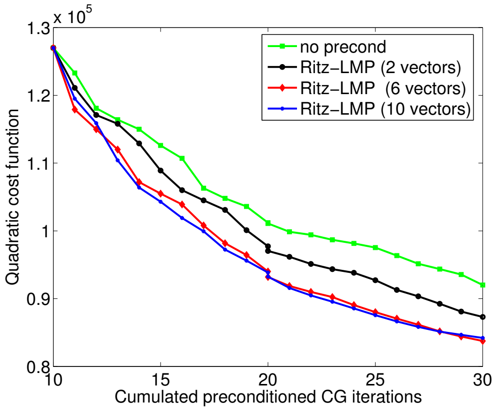
Inexact spectral-LMP



$\longrightarrow$ Negative impact of an increase in the number of vectors in S

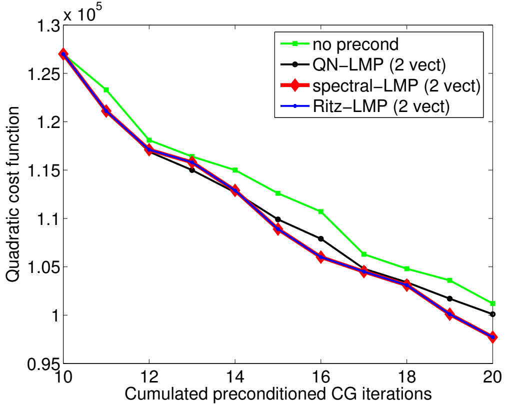
(Ritz pairs may be bad eigenpairs approximation)

Ritz-LMP



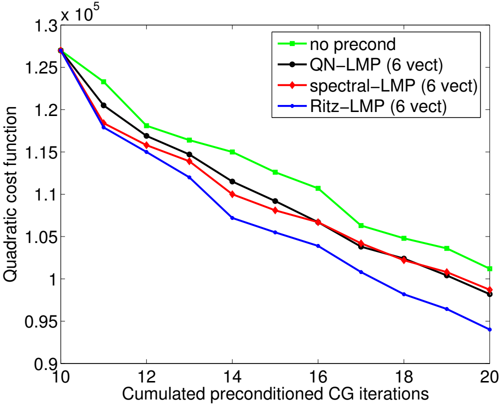
$\rightarrow$ Positive and faster impact of an increase in the number of vectors in S

Ranking LMP (2 vectors)



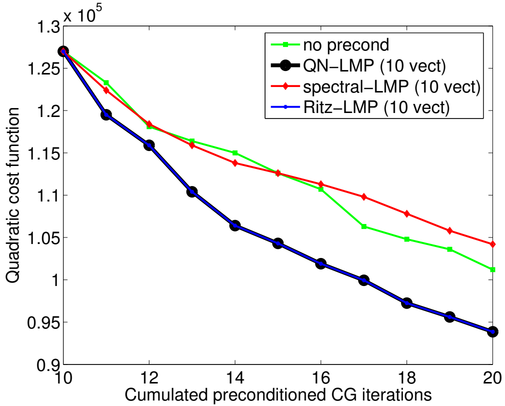
$\longrightarrow$ Inexact spectral-LMP $\equiv$ Ritz-LMP – Quasi-Newton LMP is worse

Ranking LMP (6 vectors)



→ Ritz-LMP is the best – Inexact spectral-LMP deteriorates

Ranking LMP (10 vectors)



→ Quasi-Newton LMP ≡ Ritz-LMP

→ Inexact spectral-LMP even worse than no preconditioning

Outline

What about the first system (A_0)?

[Gratton, Laloyaux, Sartenaer, Tshimanga, in preparation]

- Appropriate starting point for CG
- $\rightsquigarrow$ LMP again!

$\longrightarrow$ Illustration on a one-dimensional shallow water model

One-dimensional shallow water model

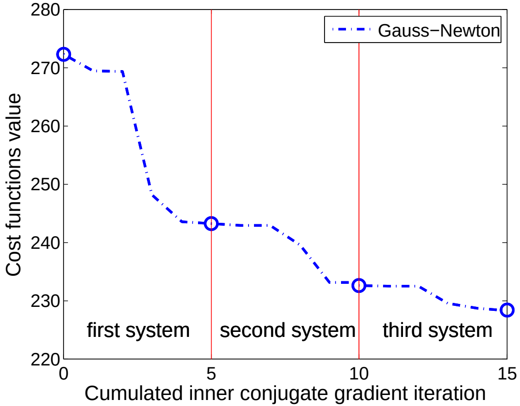
→ Estimate the velocity and geopotential of a fluid flow over an obstacle:

- 1D-grid with 250 mesh-points
- x, x_b (background) $\in \mathbb{R}^{500}$
- y_j (observations) $\in \mathbb{R}^{80}$

→ Outer/inner loop configuration:

- 3 outer loops of Gauss-Newton (linearization)
- 5 inner loops of conjugate gradient (on each of the 3 systems)

Gauss-Newton (with $x_0^0 = x_b$)



$\longrightarrow$ Computational cost dominated by 15 matrix-vector products

Improving the starting point x_0^0

Physical considerations:

- The ocean and the atmosphere exhibit an attractor
- Most of the variability can be explained in the "attractor subspace" (of low dimension r)

$\rightarrow$ Minimize first in this subspace (of basis L)

Empirical Orthogonal Functions (EOFs)

Construction of L:

- Let $\underline{x}^1, \ldots, \underline{x}^p \in \mathbb{R}^n$ be a set of state vectors $(p = 200)$
- Build $C = \frac{1}{p-1} \sum_{i=1}^{p} (\underline{x}^i - \bar{x})(\underline{x}^i - \bar{x})^T$
- Compute the eigenvectors of C (EOFs)
- Store r eigenvectors corresponding to the largest eigenvalues

$\longrightarrow$ Already used in the reduced Kalman filters (SEEK filter)

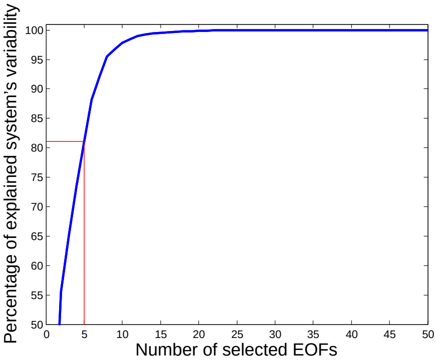
Choice for r

Select r such that:

$$\frac{\sum_{i=1}^{r} \lambda_i}{\sum_{i=1}^{n} \lambda_i} \geq 0.8$$

$(\lambda_i \searrow)$

For the shallow water model



$\longrightarrow$ The five first EOFs are computed $(r = 5)$

Ritz-Galerkin starting point

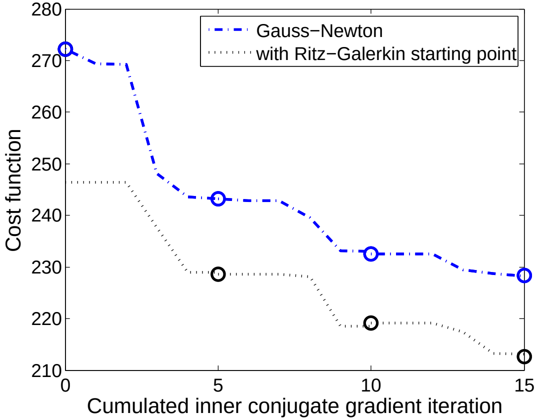
The solution of the first system in the subspace spanned by L:

$$x_0^0 = x_b + L(L^T A_0 L)^{-1} L^T b_0$$

- is called the Ritz-Galerkin starting point
- is used as starting point in the CG for the first system $(A_0 x = b_0)$

$\longrightarrow$ Computational cost dominated by $r = 5$ matrix-vector products

First improvement

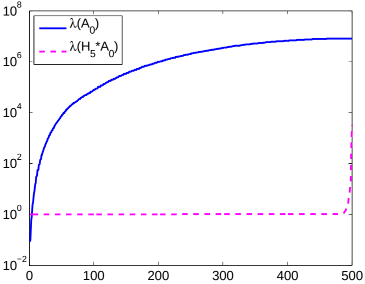


LMP again!

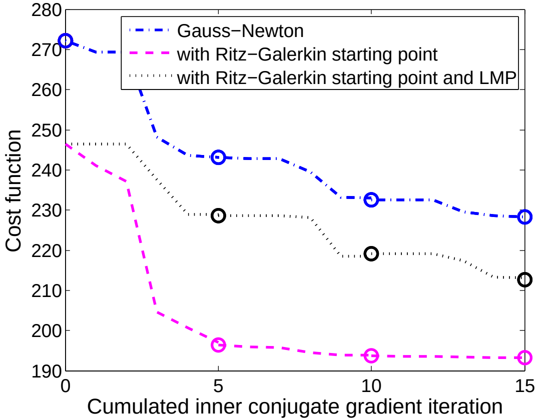
$$H = \left[I_n - S(S^T A_0 S)^{-1} S^T A_0\right] M \left[I_n - A_0 S(S^T A_0 S)^{-1} S^T\right] + S(S^T A_0 S)^{-1} S^T$$

- with $S = L$ (r EOFs)
- for free ($A_0 L$ known)

- better clustering
- better condition number



Second improvement



(Same H for the 3 systems)

Thank you for your attention !

Thank you Serge !