Data Assimilation: concepts and algorithms
(for oceanic and atmospheric applications)

S. Gratton[1]
S. Gurol[3] Ph.L. Toint[2] J. Tshimanga[1] A. Weaver[3]

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

Joint French-Czech Workshop on Krylov Methods for Inverse Problems

Outline

What is data assimilation?

You use a kind of data assimilation scheme if you sneeze whilst driving along the motorway.
As your eyes close involuntary; you retain in your mind a picture of the road ahead and traffic nearby [background],
as well as a mental model of how the car will behave in the short time [dynamical system]
before you reopen your eyes and make a course correction [adjustment to observations].

O'Neil *et al* (2004)

Outline

Predicting the state of the atmosphere, of the ocean

The state of the atmosphere or the ocean (the system) is characterized by state variables that are classically designated as fields:

- velocity components
- pressure
- density
- temperature
- salinity

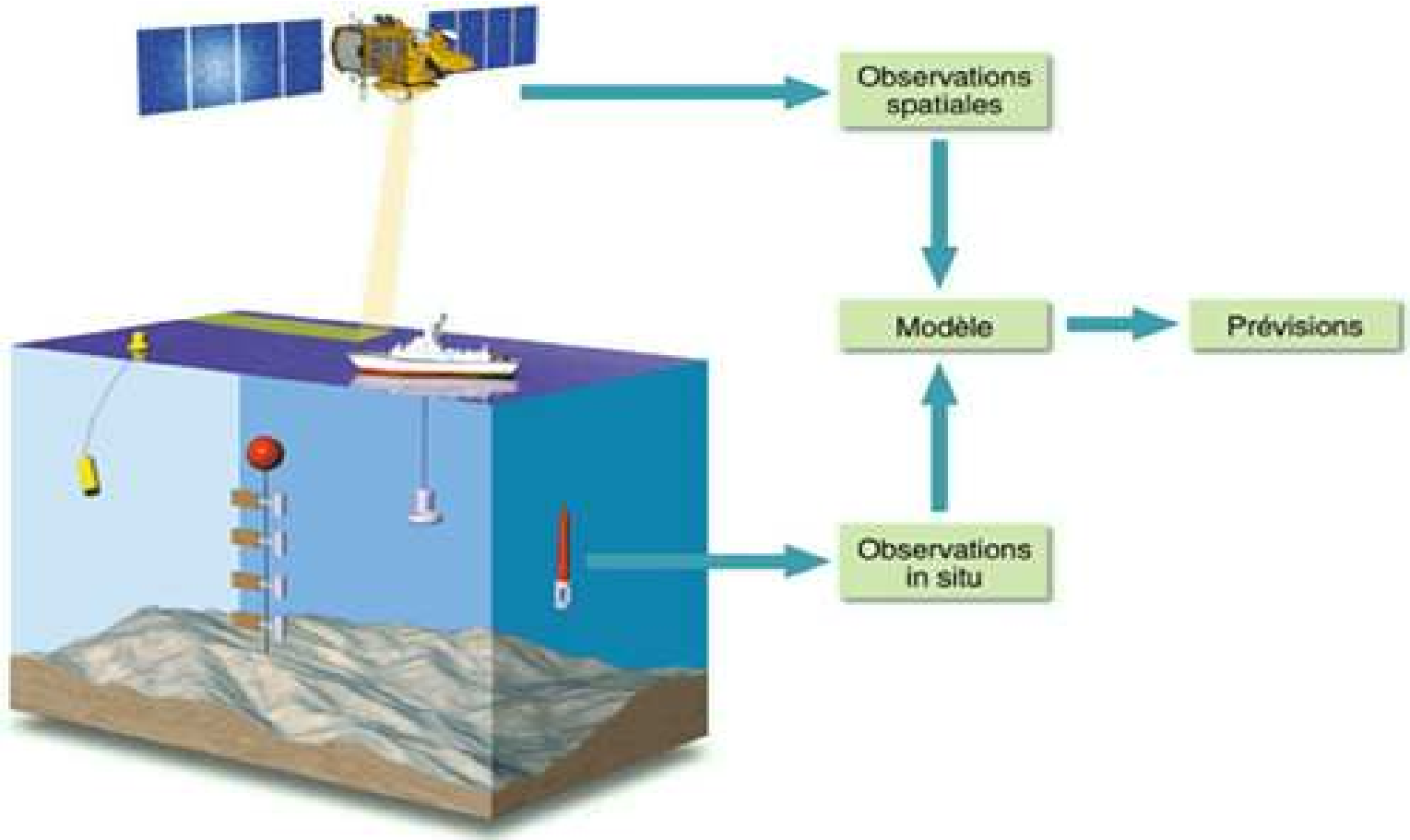
A dynamical model predicts the state of the system at a time given the state of the ocean at a earlier time. We address here this estimation problem. Applications are found in climate, meteorology, ocean,... forecasting problems. Involving large computers and nearly real-time computations.

Predicting the state of the atmosphere of the ocean

The fundamental properties of the system appear in the model as parameters:

- viscosities
- diffusivities
- rates of earth-rotation

The initial and boundary conditions necessary for integration of the dynamical model may also be regarded as parameters.


Observations spatiales
Modèle
Prévisions
Observations in situ

Optimal control problem

The fundamental problem of optimal control reads:

Definition

Find the control u (initial state parameters) out of a set of admissible controls $\mathcal{U}$ which minimizes the cost functional

$$\mathcal{J} = \int_{t_0}^{t_1} F(t, x, u) dt$$

subject to

$$\dot{x} = f(t, x, u), \text{ with } x_0 \text{ depending on } u$$

DA as an optimal control problem

Since the problem of DA is to bring the model state closer to a given set observations, this may be expressed in terms of minimizing:

$$\mathcal{J} = \int_{t_0}^{t_1} (\mathcal{H}(x) - y)^T R^{-1} (\mathcal{H}(x) - y) dt$$

subject to

$$\dot{x} = f(t, x, u)$$

or in discrete form (that we will consider for the rest)

$$\mathcal{J} = \sum_{i=0}^{N} (\mathcal{H}(\mathbf{x_i}) - \mathbf{y_i})^{\mathbf{T}} \mathbf{R^{-1}} (\mathcal{H}(\mathbf{x_i}) - \mathbf{y_i})$$

subject to

$$\mathbf{x_i} = \mathcal{M}(\mathbf{t}, \mathbf{x_0}, \mathbf{u})$$

High performance computing point of view

- The simplest instance of a DA problem is a linear least-squares problem
- Typical sizes would be for this problem 10^7 unknowns and $2 \cdot 10^7$ observations (including *a priori* information)
- The problem is not sparse
- If no particular structure taken into account, the solution of the problem on a modern ($3 \cdot 10^9$ operations/s) computer would take 200 centuries of computation by the normal equations
- In terms of memory, working with the matrix in core memory of a computer not practicable
- Therefore iterative methods are used on parallel computers

Regularization technique

If all mapping involved in the problem where linear, the data assimilation problem would often result

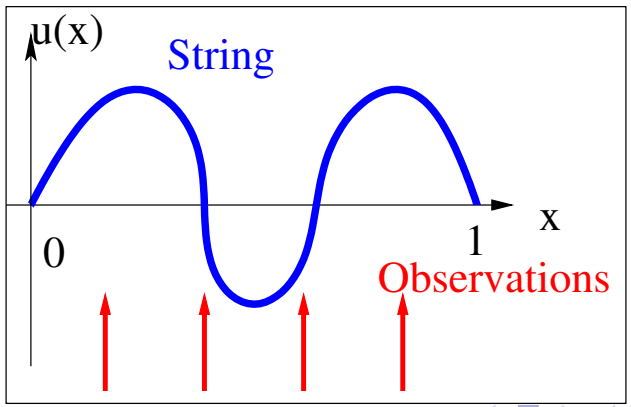
- in a linear least squares problem with more unknown than equations
- in a very ill-conditioned problem

A regularization technique is often needed. This is done using the background information

$$\mathcal{J}(\mathbf{x_0}) = \frac{\mathbf{1}}{\mathbf{2}}\|\mathbf{x_0} - \mathbf{x_b}\|^{\mathbf{2}}_{\mathbf{B^{-1}}} + \frac{\mathbf{1}}{\mathbf{2}}\sum_{\mathbf{i=0}}^{\mathbf{N}} \|\mathcal{H}_\mathbf{i}(\mathbf{x_i}) - \mathbf{y_i}\|^{\mathbf{2}}_{\mathbf{R^{-1}}}$$

A vibrating string

- We consider a vibrating string, hold fixed at both ends
- It is released with a zero initial speed, from an unknown position
- The string remains in the vertical plane
- The string is observed with a set of physical devices measuring the position string at regularly spaced points during a given time span
- We would like to make a
  forecast of the string position outside the observation time span



A vibrating string : the model

- The string position $u(x)$ is the solution of the partial differential equation

$$\begin{cases} \frac{\partial^2}{\partial t^2}u(x,t) - \frac{\partial^2}{\partial x^2}u(x,t) = 0 & in]0,1[\times]0,T[\\ u(0,t) = u(1,t) = 0, & t \in]0,T[\\ u(x,0) = u_0(x), \frac{\partial}{\partial t}u(x,0) = 0, & x \in]0,1[\end{cases}$$

- Under regularity assumptions on u_0, this system has one unique solution
- We suppose that the system is observed at times t_n
- The problem reads $\min_{u_0} \sum_{n=0}^{Nob_t} \|y_n - u(:,t_n)\|^2$
- This is an infinite dimensional linear least squares problem, that has to be discretized to be solved on a computer. Discretize then minimize.

The observations

- We consider now that the string is observed regularly in time and space. No noise, more observations than unkonwns.
- The discretized version of linear least-squares problem $\min_{u_0} \sum_{n=0}^{Nobt} \|y_n - U^n\|^2$ is solved with a conjugate gradient technique

$\longrightarrow$ `test('over')`

- Very good agreement between truth an analysis

Realistic difficult case

- In practice, observing a 3D field at all space points is out of reach
- The observations are noisy, which introduces high frequencies in the analysis
- Both effects (always) come together

$\longrightarrow$ `test('under-noisy')`

Exploiting "a priori" information

- We do not consider the previous solution acceptable, because we doubt a string might take such positions. We expect the solution to be smooth enough
- We would like to introduce the fact that the string position should not vary too much when considering points that are close in the physical space
  - purely algebraic approach, e.g. $\min_{u_0} \sum_{j=0}^{Nob_x} \frac{1}{\sigma}|u_j^0 - u_{j+1}^0|^2 + \sum_{n=0}^{Nob_t} \|y_n - U^n\|^2$
  - using a pseudo-physical smoothing process
- Sum of background (*a priori*) term and observation term

Smoothing in the discretized space with the heat equation

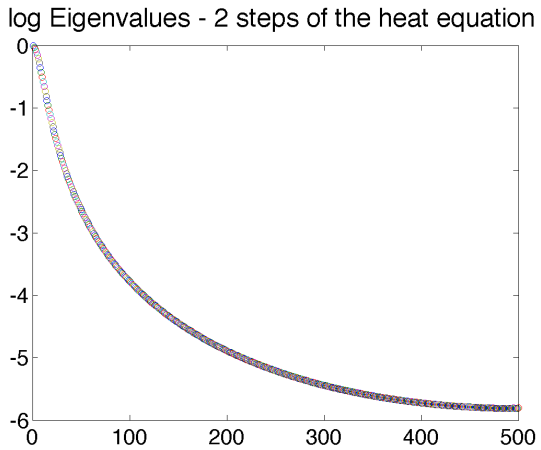
- We consider the discretized heat equation

$$\begin{cases} \frac{\partial}{\partial t}u(x,t) - \frac{\partial^2}{\partial x^2}u(x,t) = 0 & in]0,1[\times]0,T[\\ u(0,t) = u(1,t) = 0, & t \in]0,T[\\ u(x,0) = u_0(x), \frac{\partial}{\partial t}u(x,0) = 0, & x \in]0,1[\end{cases}$$

- For a given T, $u(.,T)$ is smoother than u_0, because high frequency terms get strongly damped.

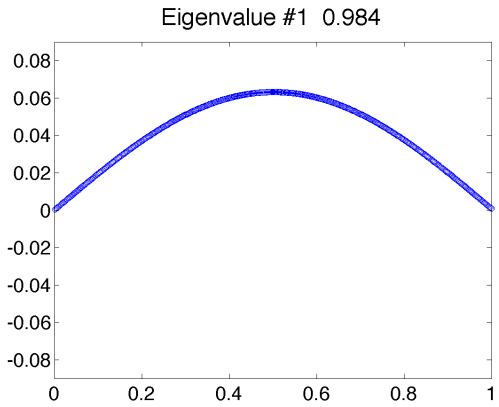
$\longrightarrow$ `simul_heat`

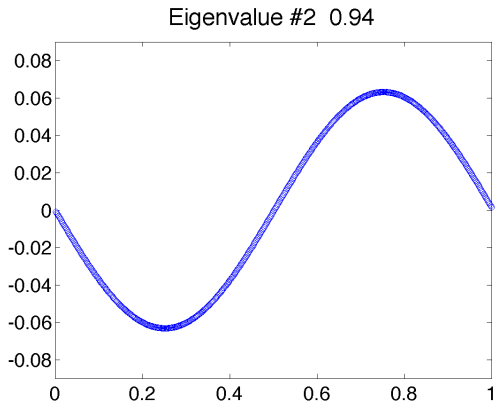
Eigenbasis of few steps in the heat equation

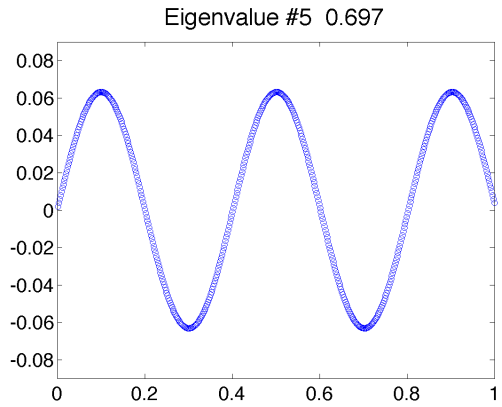


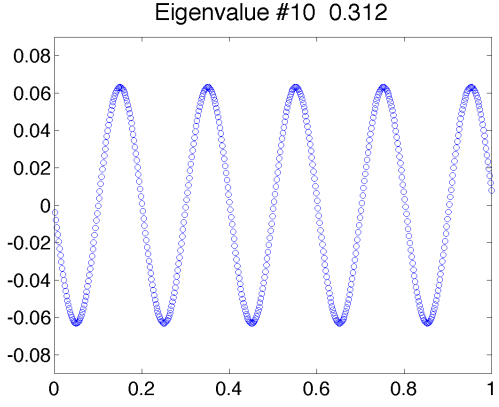
- Quickly decaying spectrum
- The resulting matrix writes $B = UDU^T$, where U is orthonormal
- The Fourier components of any u in this basis are the entries of $U^T u$

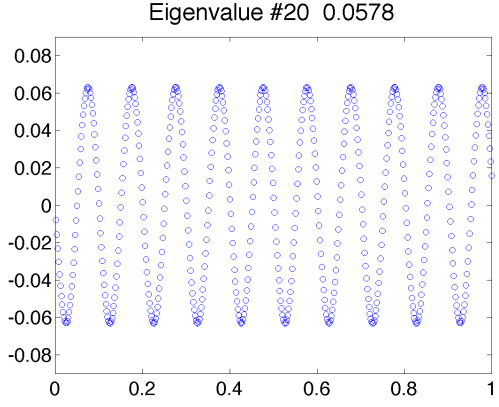
Eigenbasis of few steps in the heat equation

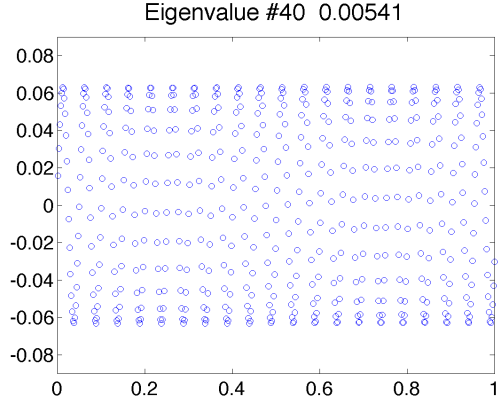












Application to the Data Assimilation problem

- A smooth vector u has most of its energy on the "largest" eigenvectors of B : $u^T B x = (Uu)^T D (Uu)$ is large
- A high-frequency vector has most of its energy on the "smallest" eigenvectors of B : $u^T B^{-1} u = (Uu)^T D^{-1} (Uu)$ is large
- We introduce the penalization of high frequencies with respect to a guess U_b, called the background : $\min_{U_0} \frac{1}{2} \|U^0 - U_b\|^2_{B^{-1}} + \frac{1}{2} \sum_{n=0}^{Nobt} \|y_n - U^n\|^2_{R^{-1}}$, where R is the covariance matrix of the observation errors

This is the 4D-Var functional

Back on the realistic difficult case

- Underdetermined case

→ `test('under-reg')`

- Noisy case

→ `test('noisy-reg')`

- Underdetermined and noisy case

→ `test('under-noisy-reg')`

Issues on background regularization

- The modelling enables to introduce a physical process to determine the background, and make the parameterization of the background error covariance matrix easy. Background matrix mat-vec in CG : another differential equation has to be solved
- In case of modeling,when a direct solution not applicable, an inner-outer iteration scheme has to be controlled
- Determining a reasonable background matrix : based on physical considerations, possibly on statistics over past assimilation periods
- Introduction of balanced relations in the background : when variables are related to each other by relations that are not accounted for in the model and not properly observed, an additional (weak) penalty term is added

Four-Dimensional Variational (4D-Var) formulation

$\longrightarrow$ 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 along the iterations

Outline

Context

We want to find the minimizer $\mathbf{x}(t_0)$ of the 4D-Var functional

$$\mathcal{J}[\mathbf{x}(t_0)] = \frac{1}{2}(\mathbf{x}(t_0) - \mathbf{x}^b)^{\mathrm{T}}\mathbf{B}^{-1}(\mathbf{x}(t_0) - \mathbf{x}^b)$$
$$+ \frac{1}{2}\sum_{j=0}^{p}(\mathcal{H}_j(\mathbf{x}(t_j)) - \mathbf{y}_j^o)^{\mathrm{T}}\mathbf{R}_j^{-1}(\mathcal{H}_j(\mathbf{x}(t_j)) - \mathbf{y}_j^o),$$

where
$\mathbf{x}(t_j) = \mathcal{M}_{j,0}(\mathbf{x}(t_0))$;
$\mathbf{B}$: background-error covariance matrix;
$\mathbf{R}_j$: observation-error covariance matrices,
$\mathcal{H}_j$: maps the model field at time t_j to the observation space.

Incremental 4D-Var Approach: algo overview

1. Transform the 4D-Var in a sequence of quadratic minimization problems
2. Increments $\delta \mathbf{x}_0^{(k)}$ are min. of functions $J^{(k)}$ defined by

$$J[\delta \mathbf{x}_0] = \frac{1}{2}\|\delta \mathbf{x}_0 - [\mathbf{x}^b - \mathbf{x}_0]\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\|\mathbf{H}\delta \mathbf{x}_0 - \mathbf{d}\|^2_{\mathbf{R}^{-1}}$$

3. Perform update

$$\mathbf{x}^{(k+1)}(t_0) = \mathbf{x}^{(k)}(t_0) + \delta \mathbf{x}_0^{(k)}.$$

Inner minimization

Minimizing

$$J[\delta \mathbf{x}_0] \quad = \quad \frac{1}{2}\|\delta \mathbf{x}_0 - [\mathbf{x}^b - \mathbf{x}_0]\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\|\mathbf{H}\delta \mathbf{x}_0 - \mathbf{d}\|^2_{\mathbf{R}^{-1}}$$

amounts to solve

$$(\mathbf{B}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H})\delta \mathbf{x}_0 = \mathbf{B}^{-1}(\mathbf{x}^b - \mathbf{x}_0) + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{d}.$$

Exact solution writes

$$\mathbf{x}^b - \mathbf{x}_0 + \left(\mathbf{B}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}\right)^{-1} \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\left(\mathbf{d} - \mathbf{H}(\mathbf{x}^b - \mathbf{x}_0)\right),$$

or equivalently (using the S-M-Woodbury formula)

$$\mathbf{x}^b - \mathbf{x}_0 + \mathbf{B}\mathbf{H}^{\mathrm{T}}\left(\mathbf{R} + \mathbf{H}\mathbf{B}\mathbf{H}^{\mathrm{T}}\right)^{-1}\left(\mathbf{d} - \mathbf{H}(\mathbf{x}^b - \mathbf{x}_0)\right).$$

Dual formulation : PSAS

1. Very popular when few observations compared to model variables. Stimulated a lot of discussion in the Ocean and Atmosphere communities
2. Relies on

$$\mathbf{x}^b - \mathbf{x}_0 + \mathbf{B}\mathbf{H}^{\mathrm{T}} \left(\mathbf{R} + \mathbf{H}\mathbf{B}\mathbf{H}^{\mathrm{T}}\right)^{-1} \left(\mathbf{d} - \mathbf{H}(\mathbf{x}^b - \mathbf{x}_0)\right)$$

3. Iteratively solve

$$\left(\mathbf{I} + \mathbf{R}^{-1}\mathbf{H}\mathbf{B}\mathbf{H}^{\mathrm{T}}\right) w = \mathbf{R}^{-1}(\mathbf{d} - \mathbf{H}(\mathbf{x}^b - \mathbf{x}_0)) \quad \text{for} \quad w$$

4. Set

$$\delta x_0 = \mathbf{x}^b - \mathbf{x}_0 + \mathbf{B}\mathbf{H}^{\mathrm{T}} w$$

Motivation : PSAS and CG-like algorithm

1. CG minimizes the Incremental 4D-Var function during its iterations. It minimizes a quadratic approximation of the non quadratic function : Gauss-Newton in the model space.
2. PSAS does not minimize the Incremental 4D-Var function during its iterations but works in the observation space.

Our goal : put the advantages of both approaches together in a Trust-Region framework, to guarantee convergence:

- Keeping the variational property, to get the so-called Cauchy decrease even when iterations are truncated.
- Being computationally efficient whenever the number of observations is significantly smaller than the size of the state vector.

Getting global convergence in the observation space !

CG-like algorithm : assumptions 1

1. Suppose the CG algorithm is applied to solve the Inc-4D using a preconditioning matrix $\mathbf{F}$
2. Suppose there exists $\mathbf{G}^{m\times m}$ such that

$$\mathbf{F}\mathbf{H}^{\mathrm{T}} = \mathbf{B}\mathbf{H}^{\mathrm{T}}\mathbf{G}$$

3. For "exact" preconditioners

$$\left(\mathbf{B}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathrm{T}} = \mathbf{B}\mathbf{H}^{\mathrm{T}}\left(\mathbf{I} + \mathbf{R}^{-1}\mathbf{H}\mathbf{B}\mathbf{H}^{\mathrm{T}}\right)^{-1}$$

Preconditioned CG on Incremental 4D-Var cost function

Initialization steps

Loop: WHILE

1. $\mathbf{q}_{i-1} = \mathbf{A}\mathbf{p}_{i-1}$
2. $\alpha_{i-1} = \mathbf{r}_{i-1}^{\mathrm{T}}\mathbf{z}_{i-1} \,/\, \mathbf{q}_{i-1}^{\mathrm{T}}\mathbf{p}_{i-1}$
3. $\mathbf{v}_i = \mathbf{v}_{i-1} + \alpha_{i-1}\mathbf{p}_{i-1}$
4. $\mathbf{r}_i = \mathbf{r}_{i-1} + \alpha_{i-1}\mathbf{q}_{i-1}$
5. $\mathbf{z}_i = \mathbf{F}\mathbf{r}_i$
6. $\beta_i = \mathbf{r}_i^{\mathrm{T}}\mathbf{z}_i \,/\, \mathbf{r}_{i-1}^{\mathrm{T}}\mathbf{z}_{i-1}$
7. $\mathbf{p}_i = -\mathbf{z}_i + \beta_i\mathbf{p}_{i-1}$

Initialization steps

Loop: WHILE

1. $\mathbf{q}_{i-1} = (\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H} + \mathbf{B}^{-1})\mathbf{p}_{i-1}$
2. $\alpha_{i-1} = \mathbf{r}_{i-1}^{\mathrm{T}}\mathbf{z}_{i-1} \,/\, \mathbf{q}_{i-1}^{\mathrm{T}}\mathbf{p}_{i-1}$
3. $\mathbf{v}_i = \mathbf{v}_{i-1} + \alpha_{i-1}\mathbf{p}_{i-1}$
4. $\mathbf{r}_i = \mathbf{r}_{i-1} + \alpha_{i-1}\mathbf{q}_{i-1}$
5. $\mathbf{z}_i = \mathbf{F}\mathbf{r}_i$
6. $\beta_i = \mathbf{r}_i^{\mathrm{T}}\mathbf{z}_i \,/\, \mathbf{r}_{i-1}^{\mathrm{T}}\mathbf{z}_{i-1}$
7. $\mathbf{p}_i = -\mathbf{z}_i + \beta_i\mathbf{p}_{i-1}$

An useful observation

Theorem

Suppose that

1. $\mathbf{B}\mathbf{H}^{\mathrm{T}}\mathbf{G} = \mathbf{F}\mathbf{H}^{\mathrm{T}}$.
2. $\mathbf{v}_0 = \mathbf{x}^b - \mathbf{x}_0$.

$\longrightarrow$ *vectors* $\widehat{\mathbf{r}}_i$, $\widehat{\mathbf{p}}_i$, $\widehat{\mathbf{v}}_i$, $\widehat{\mathbf{z}}_i$ *and* $\widehat{\mathbf{q}}_i$ *such that*

$$
\begin{aligned}
\mathbf{r}_i &= \mathbf{H}^{\mathrm{T}}\widehat{\mathbf{r}}_i, \\
\mathbf{p}_i &= \mathbf{B}\mathbf{H}^{\mathrm{T}}\widehat{\mathbf{p}}_i, \\
\mathbf{v}_i &= \mathbf{v}_0 + \mathbf{B}\mathbf{H}^{\mathrm{T}}\widehat{\mathbf{v}}_i, \\
\mathbf{z}_i &= \mathbf{B}\mathbf{H}^{\mathrm{T}}\widehat{\mathbf{z}}_i, \\
\mathbf{q}_i &= \mathbf{H}^{\mathrm{T}}\widehat{\mathbf{q}}_i
\end{aligned}
$$

Preconditioned CG on Incremental 4D-Var cost function (bis)

Initialization steps

given $\mathbf{v}_0$; $\mathbf{r}_0 = (\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H} + \mathbf{B}^{-1})\mathbf{v}_0 - \mathbf{b}$, ...

Loop: WHILE

1. $\mathbf{H}^{\mathrm{T}}\widehat{\mathbf{q}}_{i-1} = \mathbf{H}^{\mathrm{T}}(\mathbf{R}^{-1}\mathbf{H}\mathbf{B}^{-1}\mathbf{H}^{\mathrm{T}} + \mathbf{I}_m)\widehat{\mathbf{p}}_{i-1}$
2. $\alpha_{i-1} = \mathbf{r}_{i-1}^{\mathrm{T}}\mathbf{z}_{i-1} \,/\, \widehat{\mathbf{q}}_{i-1}^{\mathrm{T}}\widehat{\mathbf{p}}_{i-1}$
3. $\mathbf{B}\mathbf{H}^{\mathrm{T}}\widehat{\mathbf{v}}_i = \mathbf{B}\mathbf{H}^{\mathrm{T}}(\mathbf{v}_{i-1} + \alpha_{i-1}\widehat{\mathbf{p}}_{i-1})$
4. $\mathbf{H}^{\mathrm{T}}\widehat{\mathbf{r}}_i = \mathbf{H}^{\mathrm{T}}(\mathbf{r}_{i-1} + \alpha_{i-1}\widehat{\mathbf{q}}_{i-1})$
5. $\mathbf{B}\mathbf{H}^{\mathrm{T}}\widehat{\mathbf{z}}_i = \mathbf{F}\mathbf{H}^{\mathrm{T}}\widehat{\mathbf{r}}_i = \mathbf{B}\mathbf{H}^{\mathrm{T}}\mathbf{G}\widehat{\mathbf{r}}_i$
6. $\beta_i = (\mathbf{r}_i^{\mathrm{T}}\mathbf{z}_i \,/\, \mathbf{r}_{i-1}^{\mathrm{T}}\mathbf{z}_{i-1})$
7. $\mathbf{B}\mathbf{H}^{\mathrm{T}}\widehat{\mathbf{p}}_i = \mathbf{B}\mathbf{H}^{\mathrm{T}}(-\widehat{\mathbf{z}}_i + \beta_i\widehat{\mathbf{p}}_{i-1})$

Restricted PCG (version 1)

Initialization steps

given $\mathbf{v}_0$; $\mathbf{r}_0 = (\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H} + \mathbf{B}^{-1})\mathbf{v}_0 - \mathbf{b}$, ...

Loop: WHILE

1. $\widehat{\mathbf{q}}_{i-1} = (\mathbf{I}_m + \mathbf{R}^{-1}\mathbf{H}\mathbf{B}^{-1}\mathbf{H}^{\mathrm{T}})\widehat{\mathbf{p}}_{i-1}$
2. $\alpha_{i-1} = \widehat{\mathbf{r}}_{i-1}^{\mathrm{T}}\mathbf{H}\mathbf{B}\mathbf{H}^{T}\widehat{\mathbf{z}}_{i-1} \,/\, \widehat{\mathbf{q}}_{i-1}^{\mathrm{T}}\mathbf{H}\mathbf{B}\mathbf{H}^{T}\widehat{\mathbf{p}}_{i-1}$
3. $\widehat{\mathbf{v}}_i = \widehat{\mathbf{v}}_{i-1} + \alpha_{i-1}\widehat{\mathbf{p}}_{i-1}$
4. $\widehat{\mathbf{r}}_i = \widehat{\mathbf{r}}_{i-1} + \alpha_{i-1}\widehat{\mathbf{q}}_{i-1}$
5. $\widehat{\mathbf{z}}_i = \mathbf{F}\mathbf{H}^{\mathrm{T}}\widehat{\mathbf{r}}_i = \mathbf{G}\widehat{\mathbf{r}}_i$
6. $\beta_i = \widehat{\mathbf{r}}_i^{\mathrm{T}}\mathbf{H}\mathbf{B}\mathbf{H}^{T}\widehat{\mathbf{z}}_i \,/\, \widehat{\mathbf{r}}_{i-1}^{\mathrm{T}}\mathbf{H}\mathbf{B}\mathbf{H}^{T}\widehat{\mathbf{z}}_{i-1}$
7. $\widehat{\mathbf{p}}_i = -\widehat{\mathbf{z}}_i + \beta_i\widehat{\mathbf{p}}_{i-1}$

More transformations

1. Consider $\mathbf{w}$ and $\mathbf{t}$ defined by

$$\mathbf{w}_i = \mathbf{H}\mathbf{B}\mathbf{H}^{\mathrm{T}}\widehat{\mathbf{z}}_i \quad \text{and} \quad \mathbf{t}_i = \mathbf{H}\mathbf{B}\mathbf{H}^{\mathrm{T}}\widehat{\mathbf{p}}_i$$

2. From Restricted PCG (version 1)

$$\mathbf{t}_i = \begin{cases} -\mathbf{w}_0 & \text{if } i = 0, \\ -\mathbf{w}_i + \beta_i \mathbf{t}_{i-1} & \text{if } i > 0, \end{cases}$$

3. Use these relations into Restricted PCG (version 1)
4. Transform Restricted PCG (version 1) into Restricted PCG (version 2)

Restricted PCG (version 2)

Initialization steps

Loop: WHILE

1. $\widehat{\mathbf{q}}_{i-1} = \mathbf{R}^{-1}\mathbf{t}_{i-1} + \widehat{\mathbf{p}}_{i-1}$
2. $\alpha_{i-1} = \mathbf{w}_{i-1}^{\mathrm{T}}\widehat{\mathbf{r}}_{i-1} \,/\, \widehat{\mathbf{q}}_{i-1}^{\mathrm{T}}\mathbf{t}_{i-1}$
3. $\widehat{\mathbf{v}}_i = \widehat{\mathbf{v}}_{i-1} + \alpha_{i-1}\widehat{\mathbf{p}}_{i-1}$
4. $\widehat{\mathbf{r}}_i = \widehat{\mathbf{r}}_{i-1} + \alpha_{i-1}\widehat{\mathbf{q}}_{i-1}$
5. $\widehat{\mathbf{z}}_i = \mathbf{G}\widehat{\mathbf{r}}_i$
6. $\mathbf{w}_i = \mathbf{H}\mathbf{B}\mathbf{H}^{\mathrm{T}}\widehat{\mathbf{z}}_i$
7. $\beta_i = \mathbf{w}_i^{\mathrm{T}}\widehat{\mathbf{r}}_i \,/\, \mathbf{w}_{i-1}^{\mathrm{T}}\widehat{\mathbf{r}}_{i-1}$
8. $\widehat{\mathbf{p}}_i = -\widehat{\mathbf{z}}_i + \beta_i\widehat{\mathbf{p}}_{i-1}$
9. $\mathbf{t}_i = -\mathbf{w}_i + \beta_i\mathbf{t}_{i-1}$

Comments

We summarize here the main features of RPCG:

- It amounts to solve the observation system with the right inner-product HBH^T
- It is mathematically equivalent to PCG in the sense that, in exact arithmetic, both algorithms generate exactly the same iterates.
- It contains a single occurrence of the matrix-vector products by $\mathbf{B}$, $\mathbf{H}$, $\mathbf{H^T}$ and $\mathbf{R}^{-1}$ per iteration.

Loss (and recovery) of orthogonality

1. The modified (G-S) orthogonalization scheme writes

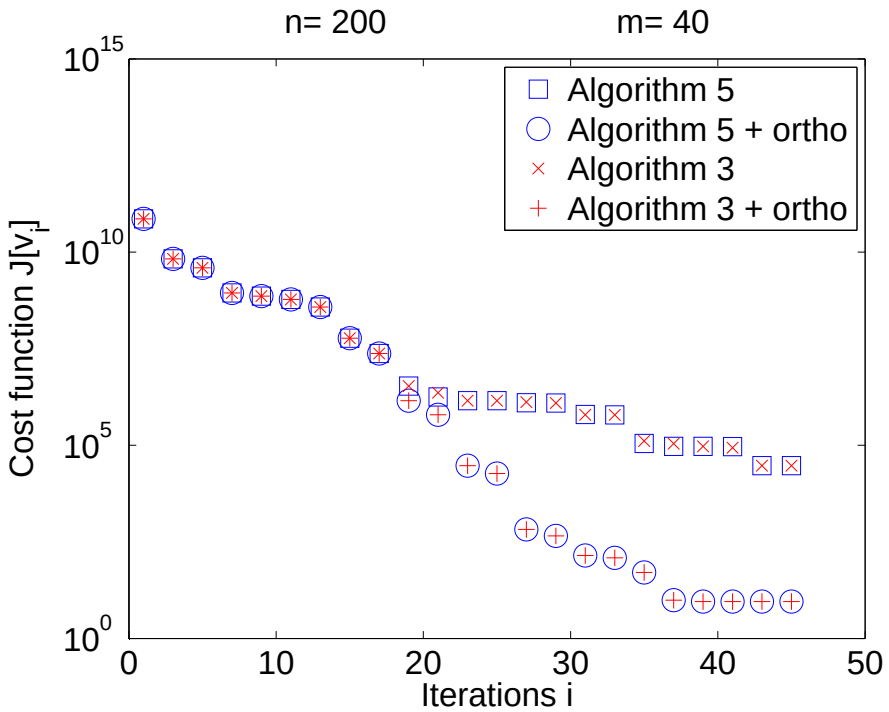
$$\mathbf{r}_i \leftarrow \prod_{j=1}^{i-1} \left(\mathbf{I}_n - \frac{\mathbf{r}_j \mathbf{r}_j^{\mathrm{T}}}{\mathbf{r}_j^{\mathrm{T}} \mathbf{F} \mathbf{r}_j} \right) \mathbf{r}_i.$$

2. We suggest the following re-orthogonalization scheme

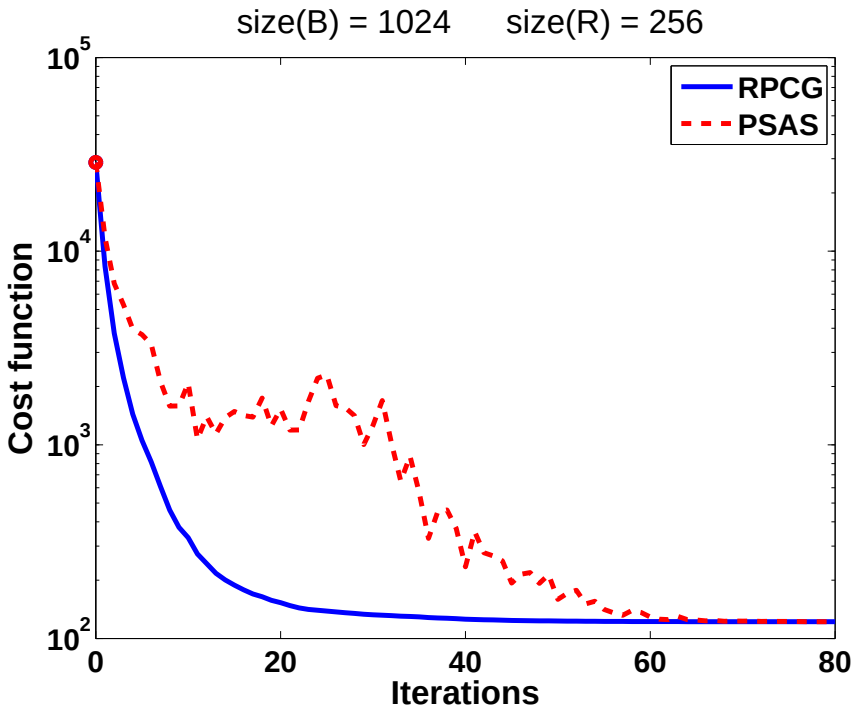
$$\hat{\mathbf{r}}_i \leftarrow \prod_{j=1}^{i-1} \left(\mathbf{I}_m - \frac{\hat{\mathbf{r}}_j \mathbf{w}_j^{\mathrm{T}}}{\hat{\mathbf{r}}_j^{\mathrm{T}} \mathbf{w}_j} \right) \hat{\mathbf{r}}_i. \qquad (1)$$

3. Note that the total number of pairs to be stored can be reduced if selective reorthogonalization is performed.

Loss (and recovery) of orthogonality : experiment



Experiments



Conclusions

- Have proposed a reformulation of the PCG for

$$(\mathbf{B}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H})\delta\mathbf{x}_0 = \mathbf{B}^{-1}(\mathbf{x}^b - \mathbf{x}_0) + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{d}$$

- The RPCG is mathematically equivalent to PCG
- Exploits the fact that all vectors lie in a subspace of $\mathbb{R}^m$
- Cheaper than CG (memory and computation)
- Some numerical experiments shown

Perpectives

Perpectives

- Behaviour in presence of round-off error
- Find efficient preconditioners $\mathbf{F}$ such that

$$\mathbf{F}\mathbf{H}^{\mathrm{T}} = \mathbf{B}\mathbf{H}^{\mathrm{T}}\mathbf{G}$$

- Implement RPCG in a real life data assimilation system : RTRA project

Towards further reduction of the cost

- We have shown that RPCG allows memory and computational cost reduction whenever the number of observation is smaller than the size of the control vector
- Similar results are possible with other Krylov methods (GMRES, FOM, ...)
- The question now is: can we reduce cost further ?
- Possible answer: inexact (cheap) matrix-vector products (truncated B^{-1}, R^{-1}, simplified models, ...)

  (Simoncini and Szyld, van den Eshop and Sleipen, Giraud, Gratton and Langou, ...)

$\rightarrow$ But, there is a need of a stable modification of RPCG.

The Arnoldi process

Define (in the full space) $A = I_n + BH^T R^{-1} H$ and set

$$K = BH^T, \qquad L = R^{-1} H$$

the successive nested Krylov subspaces generated by the sequence

$$b, \;\; (\gamma I_n + K^T L)b, \;\; (\gamma I_n + K^T L)^2 b, \;\; (\gamma I_n + K^T L)^3 b, \;\; \ldots \tag{2}$$

or, equivalently, by

$$b, \;\; (K^T L)b, \;\; (K^T L)^2 b, \;\; (K^T L)^3 b, \;\; \ldots \tag{3}$$

The Arnoldi process generates an orthonormal basis of each of the these subspaces, i.e. a set of vectors $\{v_i\}_{i=1}^{k+1}$ with $v_1 = b/\|b\|$ such that, after k steps,

$$K^T L \, V_k = V_{k+1} H_k, \tag{4}$$

where $V_k \equiv [v_1, \ldots, v_k]$ and H_k is a $(k+1) \times k$ upper-Hessenberg matrix.

Related methods: GMRES, MINRES, FOM, CG

Depending on how the matrix H_k is exploited to solve the problem we have

- The GMRES algorithm ($\equiv$ MINRES for $K^T = L$)

$$y_k = \arg\min_y \|H_k y - \beta_1 e_1\|, \qquad s_k = V_k y_k$$

- The FOM algorithm ($\equiv$ CG for $K^T = L$)

$$H_k^{\square} y = \beta_1 e_1, \qquad s_k = V_k y_k$$

  here, $H_k^{\square}$ is the leading $k \times k$ submatrix of H_k.

GMRES (FOM) use long recurrences while MINRES (CG) use short ones.
Let

$$r_k = (I + K^T K) V_k y_k - b \qquad \text{and} \qquad f_k = \frac{1}{2} y_k^T V_k^T (\gamma I + K^T K) V_k y_k - b^T V_k y_k$$

$\rightarrow$ GMRES and MINRES monotionically minimize r_k while FOM and CG monotically minimize f_k along the iterations.

Range-space GMRES and FOM (RSGMR and RSFOM)

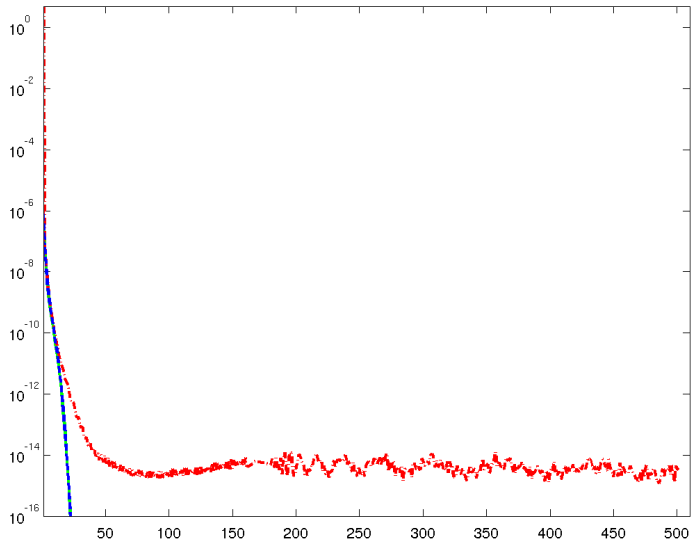
As CG may be rewritten in the observation space to yield RPCG, algorithms GMRES, MINRES and FOM may be rewritten to yields similar variants.

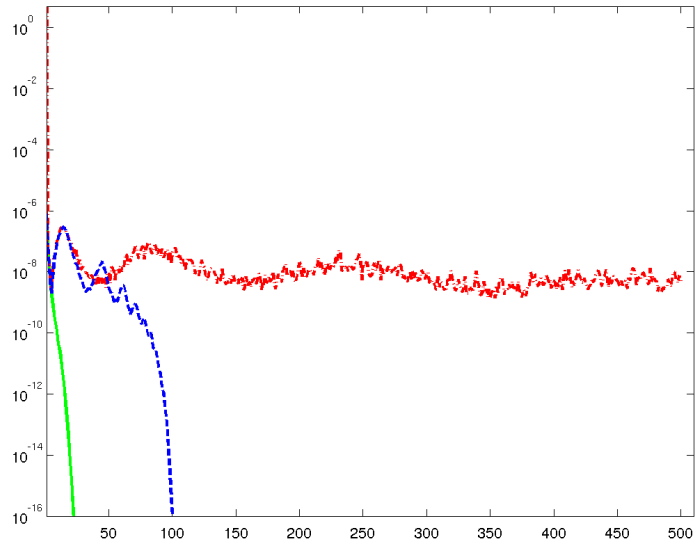
Why a range-space GMRES and FOM (RSGMR and RSFOM)?

- The FOM setting provides better accuracy and is much better suited to use inexact matrix-vector products.
- The cost of storing an orthonormal basis of the successive Krylov spaces is much lower for range-space methods than for full-space ones.

Exact and inexact products: FOM vs CG

Is CG a reasonable framework for inexact products ?





Comparing $\|r_k\|/(\|A\|\|s_*\|)$ for FOM, CG with reortho and CG for exact (left) and inexact (right) products $(\tau = 10^{-9}, \kappa \approx 10^{6})$

Stability and convergence with inexact product

We want to bound $\|r_k\|$ in the context of Arnoldi process under inexact matrix-vector products.

Some reasons to consider this question

- The inexact nature of computer arithmetic implies that such such errors are inevitable
- To allow matrix-vector products in an inexact but cheaper form

Note that

- the analysis is for GMRES but that in the context of FOM similar conclusions will hold.
- standard CG and MINRES are no longer equivalent to FOM and GMRES in the context of unsymmetric perturbations.

Two error models

Assume that each iteration i product by K, K or L is inexact, that is

$$L_i = L + E_{L,i}, \qquad K_i^T = K^T + E_{K^T,i}, \qquad \text{and} \qquad K_i = K + E_{K,i}$$

for some errors $E_{L,i}, E_{K^T,i}, \qquad$ and $\qquad E_{K,i}$. Consider two error models to describe the inaccuracy in the matrix-vector products.

- Backward:

$$\begin{aligned} \|E_{K,i+1}\| &\leq \tau_{K,i+1}\|K\|, \\ \|E_{K^T,i+1}\| &\leq \tau_{K^T,i+1}\|K\|, \\ \|E_{L,i+1}\| &\leq \tau_{L,i+1}\|L\|, \\ \|E_{K^T,*}\| &\leq \tau_*\|K\| \end{aligned}$$

- Forward:

$$\begin{aligned} \|E_{K,i+1}\, u_n\| &\leq \tau_{K,i+1}\|Ku_n\|, \\ \|E_{K^T,i+1}\, u_m\| &\leq \tau_{K^T,i+1}\|Ku_m\| \\ \|E_{L,i+1}\, u_n\| &\leq \tau_{L,i+1}\|Lu_n\| \\ \|E_{K^T,*}\, u_m\| &\leq \tau_*\|Ku_m\| \end{aligned}$$

Results for the backward error model

Define

$$q_k = H_k y_k - \beta e_1, \qquad G = \max[\|K\|, \|L\|], \qquad \omega_k = \max_{i,\ldots,k} \|\hat{v}_i\|$$

$$\kappa(K) = \text{condition number of } K$$

(... after some analysis ...)

Theorem

Assume the backward-error model. Then

$$\begin{aligned} \|r_k\| &\leq \sqrt{2(k+1)}\,\|q_k\| + \|K\|\omega_k \left[\tau_* \gamma \sqrt{k}\|y_k\| + 4\,G^2 \textstyle\sum_{i=1}^{k} |[y_k]_i|\,\tau_i\right] \\ &\leq \sqrt{2(k+1)}\left[\|q_k\| + \tau_{\max}\kappa(K)\,(\gamma + 4\,G^2)\|y_k\|\right]. \end{aligned}$$

where $\tau_{\max} = \max[\tau_1, \ldots, \tau_k]$.

Results for the forward error model

Theorem

Assume the forward-error model. Then

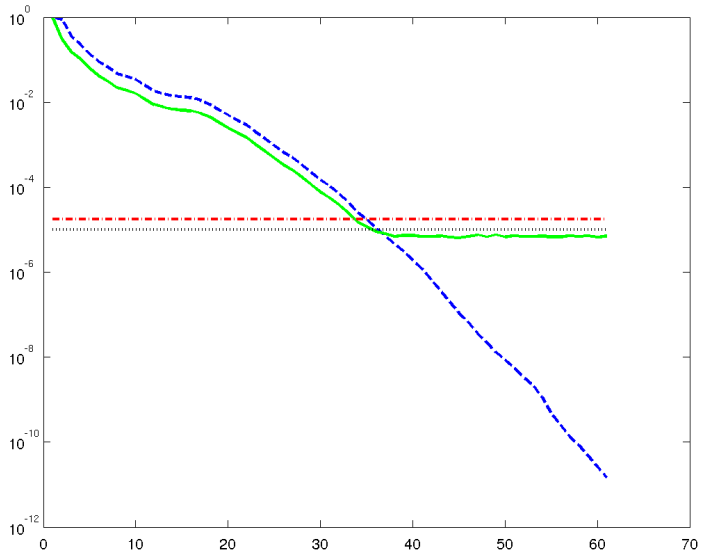
$$\begin{aligned} \|r_k\| &\leq \sqrt{2(k+1)}\,\|q_k\| + \sqrt{2}\left[\tau_* \gamma \sqrt{k}\|y_k\| + 4\,G\,\|K\| \sum_{i=1}^{k} |[y_k]_i|\,\tau_i\right] \\ &\leq \sqrt{2(k+1)}\left[\|q_k\| + \tau_{\max}\,(\gamma + 4\,G\,\|K\|)\|y_k\|\right]. \end{aligned}$$

Note in both sets of bounds:

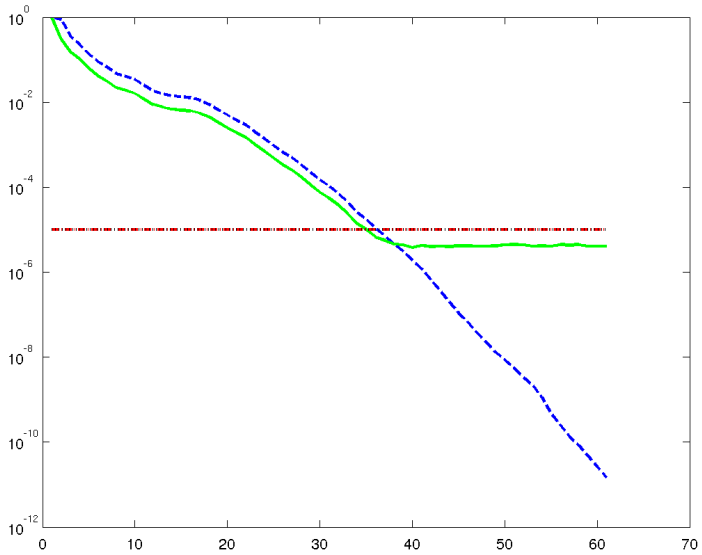
- The first of these bounds allows for variable accuray requirements
- special role of τ_*.

Error models (1)

Is the error model important? $(\epsilon = 10^{-5}, \kappa \approx 10^2)$



Backward error model

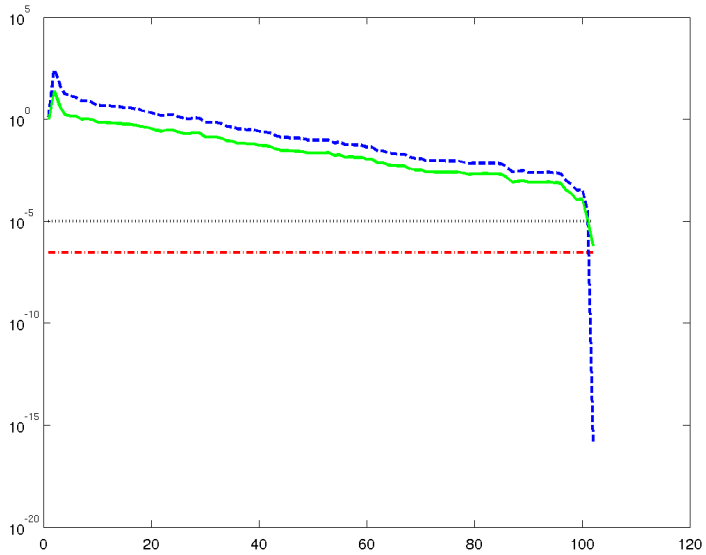


Forward error model

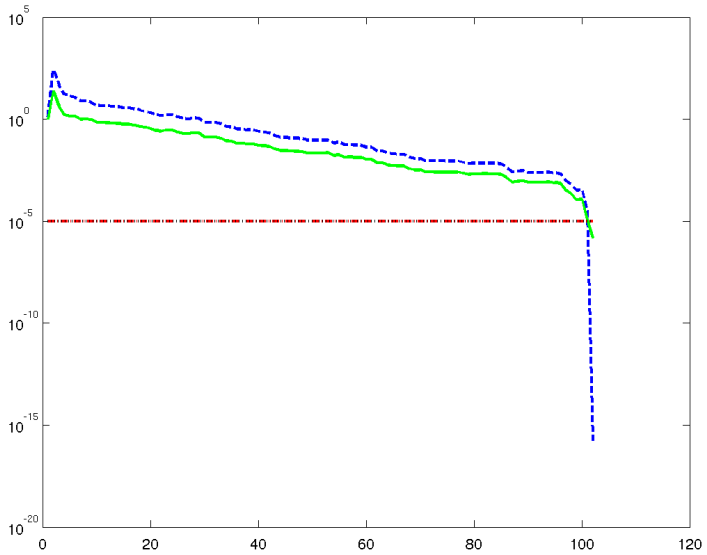
(normalized $\|r_k\|$, normalized $\|q_k\|$, accuracy threshold τ)

Error models (2)

Yes, it can definitely make the difference $(\epsilon = 10^{-5}, \kappa \approx 10^{9})$



Backward error model

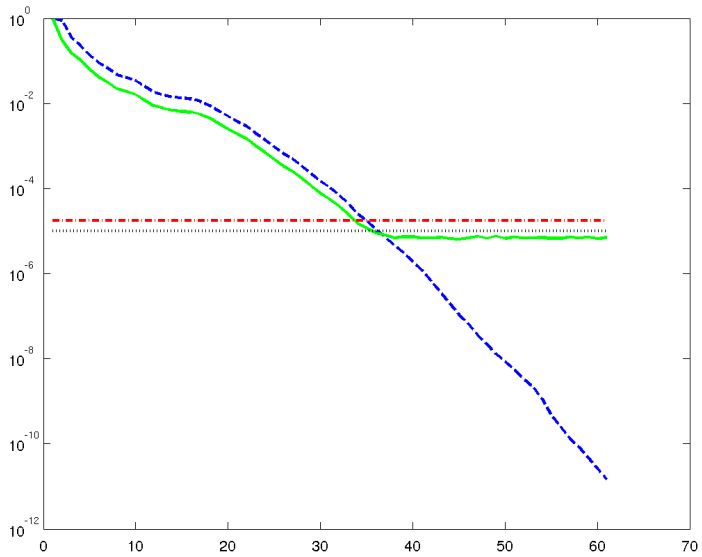


Forward error model

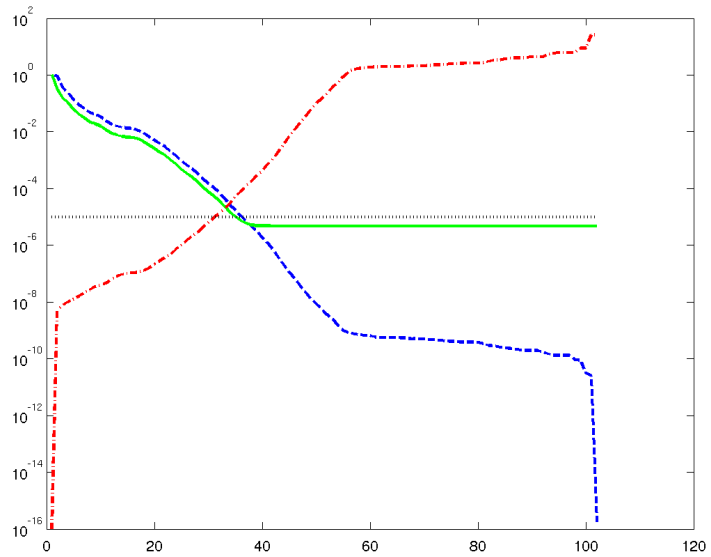
(normalized $\|r_k\|$, normalized $\|q_k\|$, accuracy threshold τ)

Fixed vs variable accuracy threshold (1)

Can we use variable accuracy thresholds efficiently? $(\epsilon = 10^{-5}, \kappa \approx 10^2)$



Fixed τ



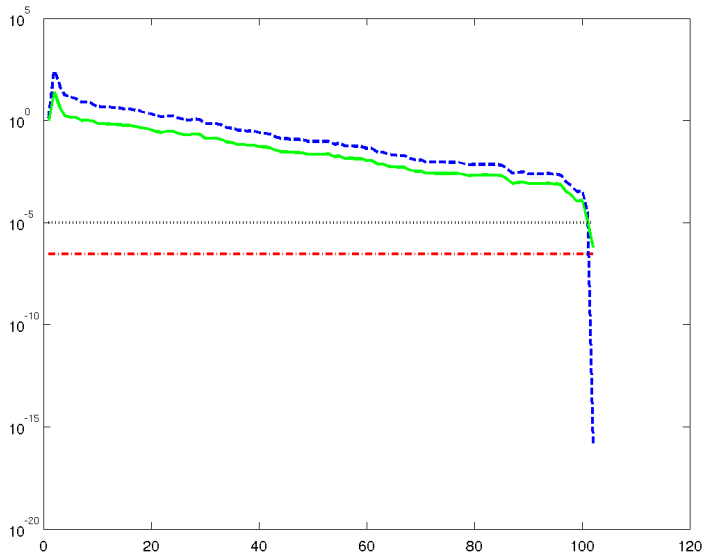
$\tau \approx 1/\|q_k\|$

(normalized $\|r_k\|$, normalized $\|q_k\|$, accuracy threshold τ)

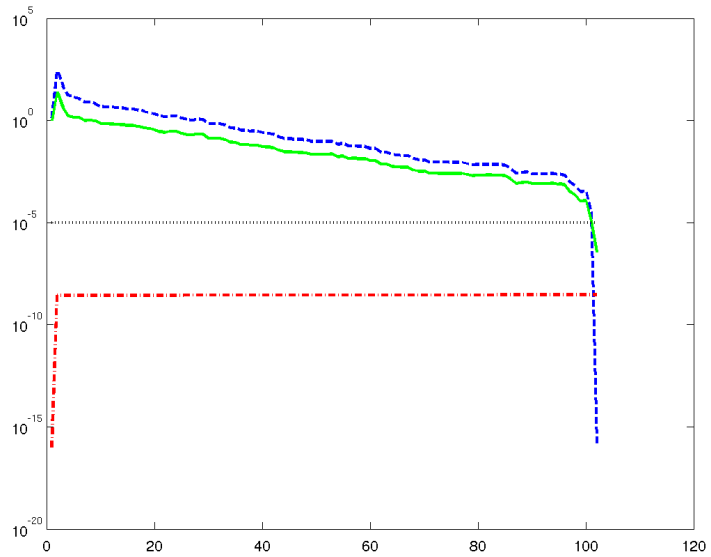
Fixed vs variable accuracy threshold (2)

Maybe..., not obvious.

$(\epsilon = 10^{-5}, \kappa \approx 10^2)$



Fixed τ



$\tau \approx 1/\|q_k\|$

(normalized $\|r_k\|$, normalized $\|q_k\|$, accuracy threshold τ)

Conclusions

- Range space methods may be designed to gain from low rank
- Further gains may be obtained from inexact products
- Formal bounds on the residual norm are available in this context
- Forward error modelling gives more flexibility than backward
- True application: a real challenge (but we are working on it!)

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 $\min_{x \in \mathbb{R}^n} \frac{1}{2} x^T A x - b^T x$ 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

$$\alpha_{k-1} \leftarrow \frac{r_{k-1}^T r_{k-1}}{p_{k-1}{}^T A p_{k-1}}$$ Compute the step length

$$x_k \leftarrow x_{k-1} + \alpha_{k-1} p_{k-1}$$ Update the iterate

$$r_k \leftarrow r_{k-1} + \alpha_{k-1} A p_{k-1}$$ Update the residual

$$\beta_k \leftarrow \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}}$$ Ensure A-conjugate directions

$$p_k \leftarrow -r_k + \beta_k p_{k-1}$$ Update the descent direction

Elementary properties of the LMP

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

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*

$$G = L \; - \; SR^{-1}R^{-T}S^T AL \; + \; SR^{-1}X^{-T}S^T L^{-T}$$

where

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

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

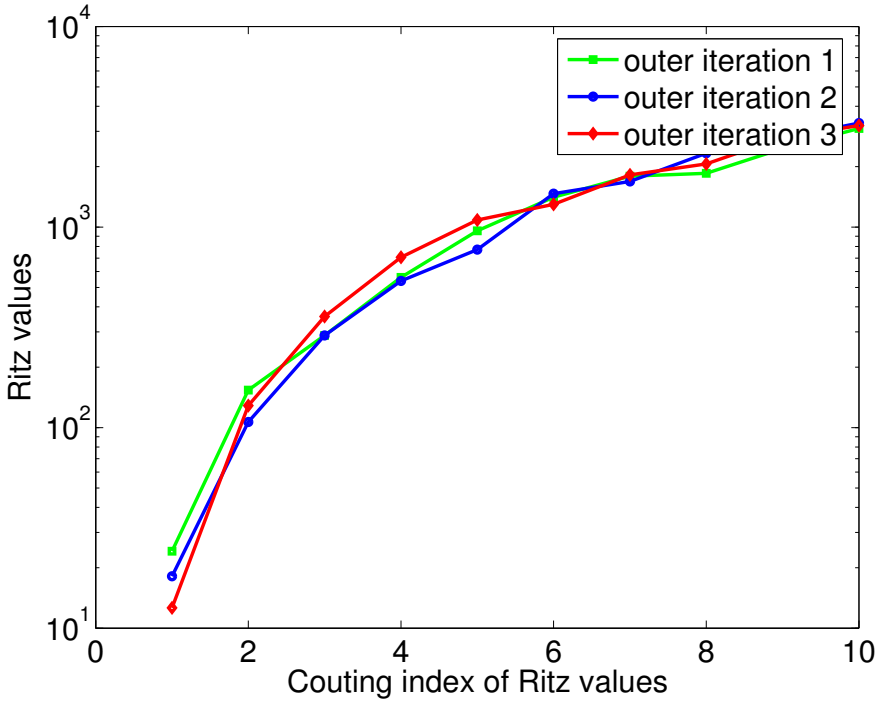
The framework

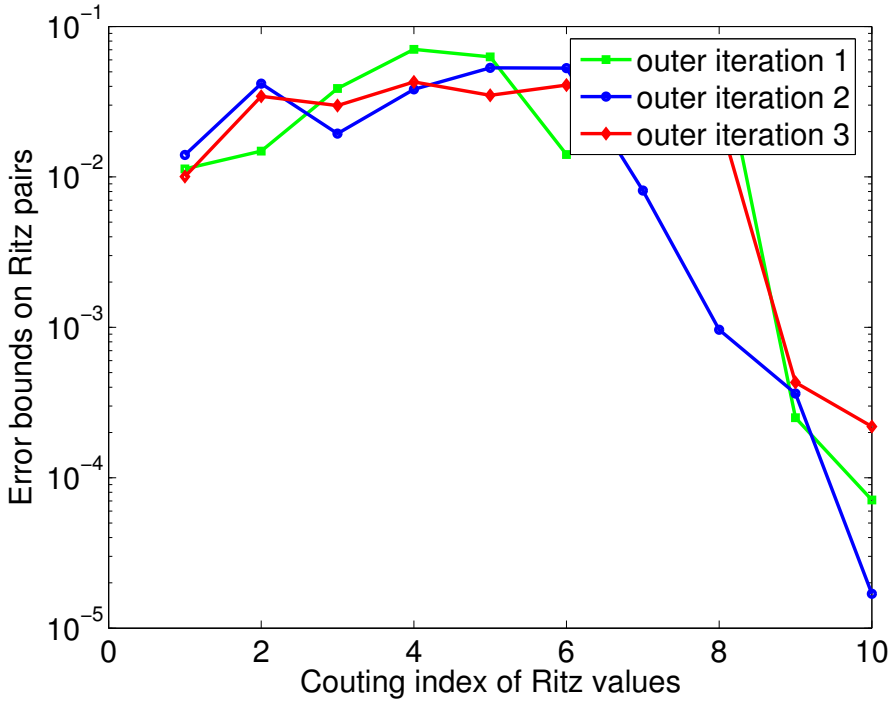
[Tshimanga, Gratton, Weaver, Sartenaer, QJRMS, 2007]

- System with 10^7 degrees of freedom
- 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

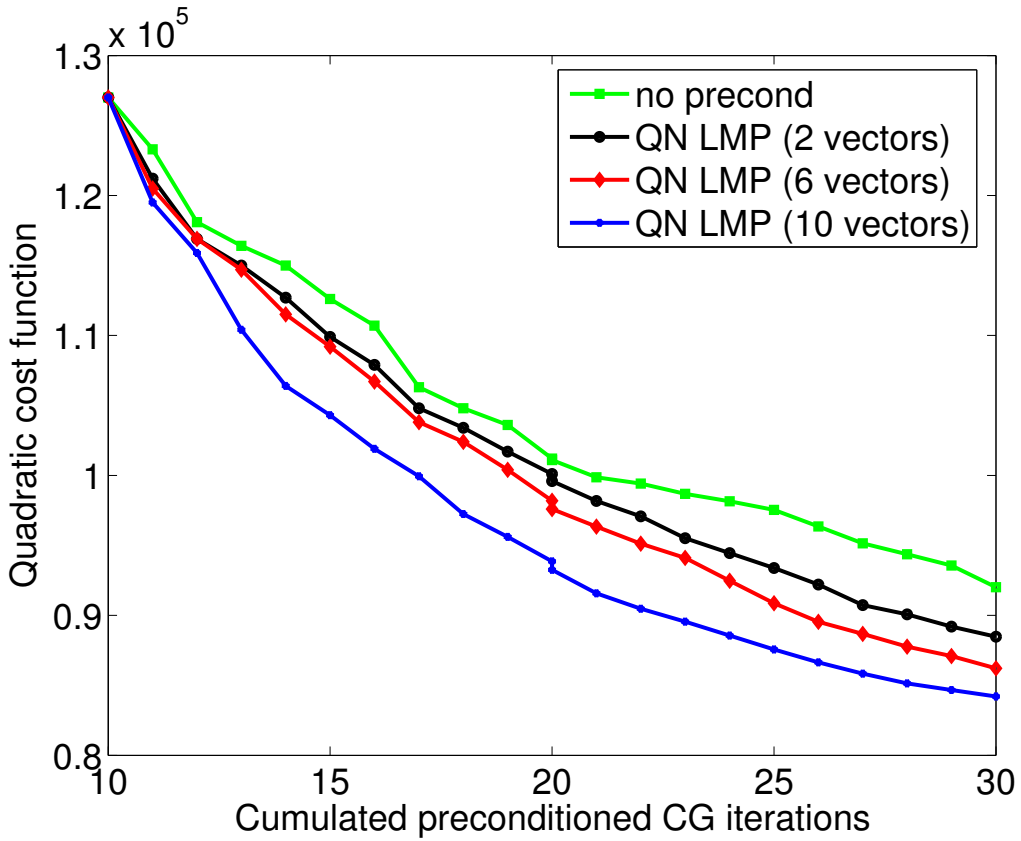
In order to

- 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:

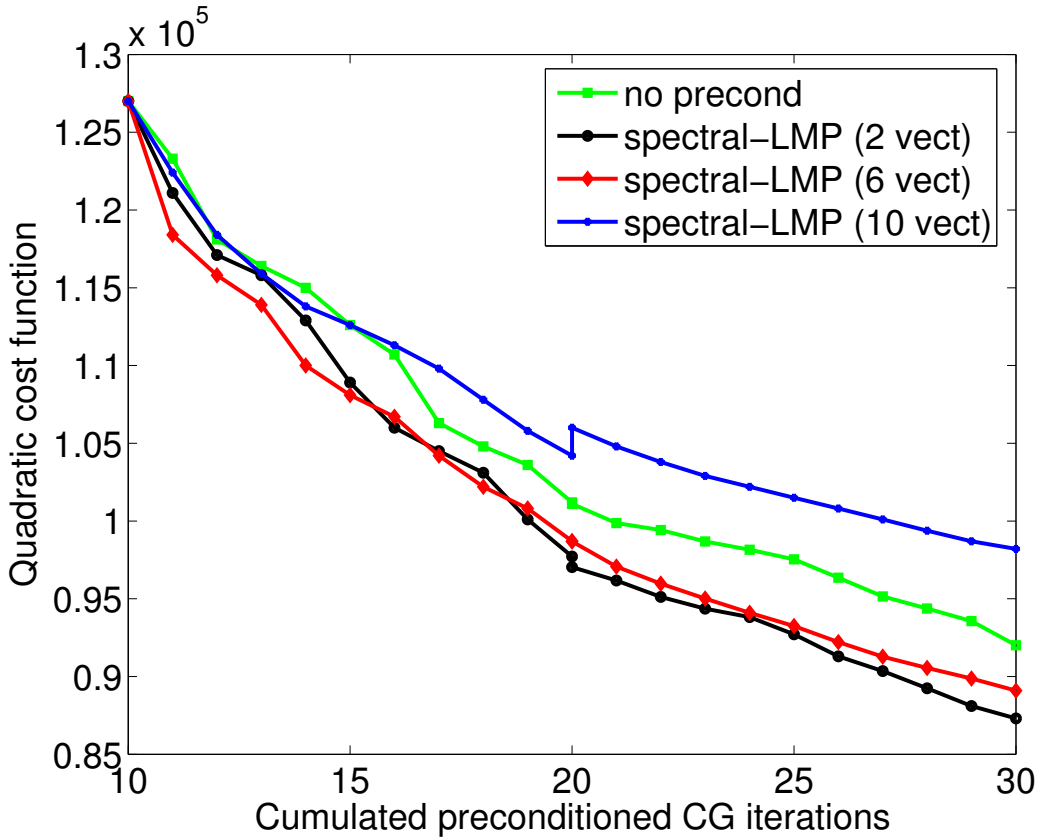
- Ritz-vectors are selected according to their convergence
- Descent directions are selected as the latest ones

Quasi-Newton LMP



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

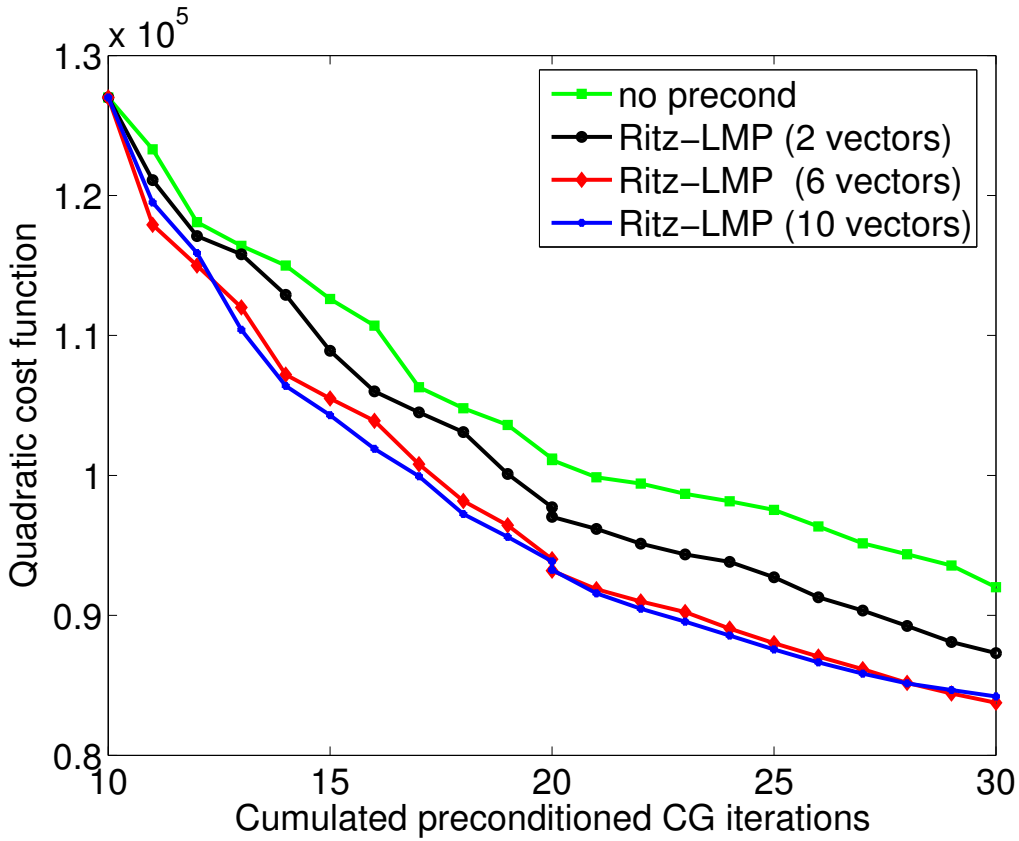
Inexact spectral-LMP



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

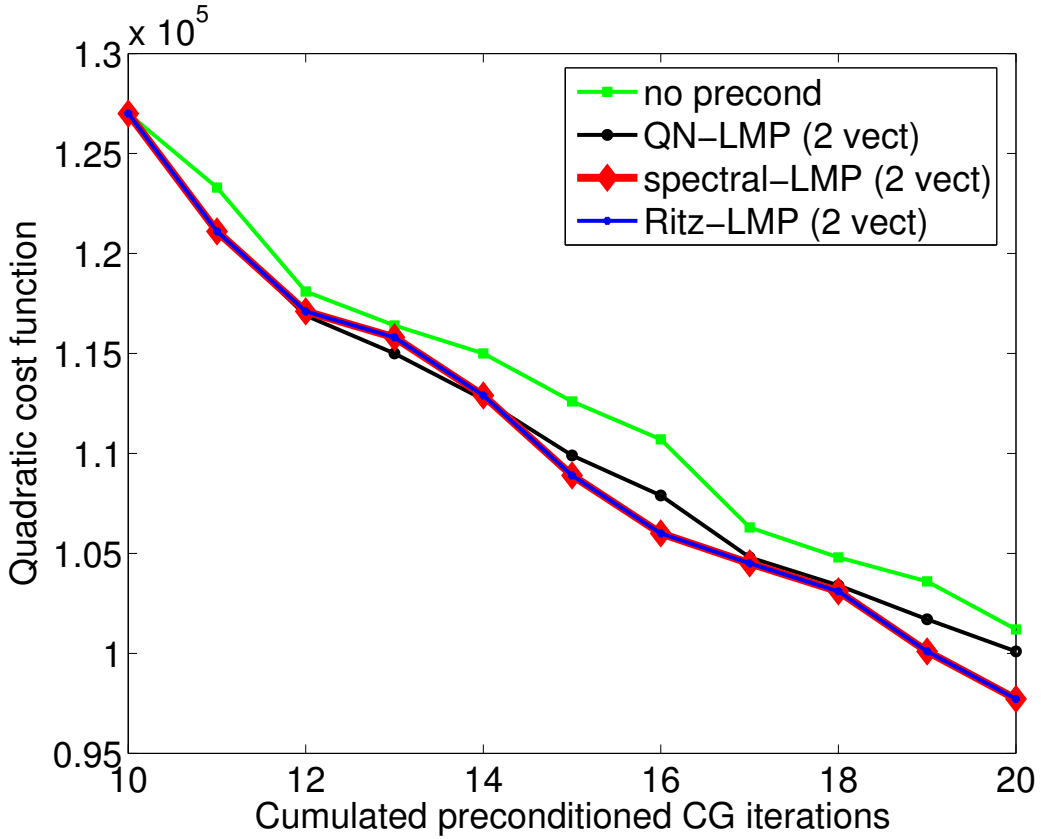
(Ritz pairs may be bad eigenpairs approximation)

Ritz-LMP



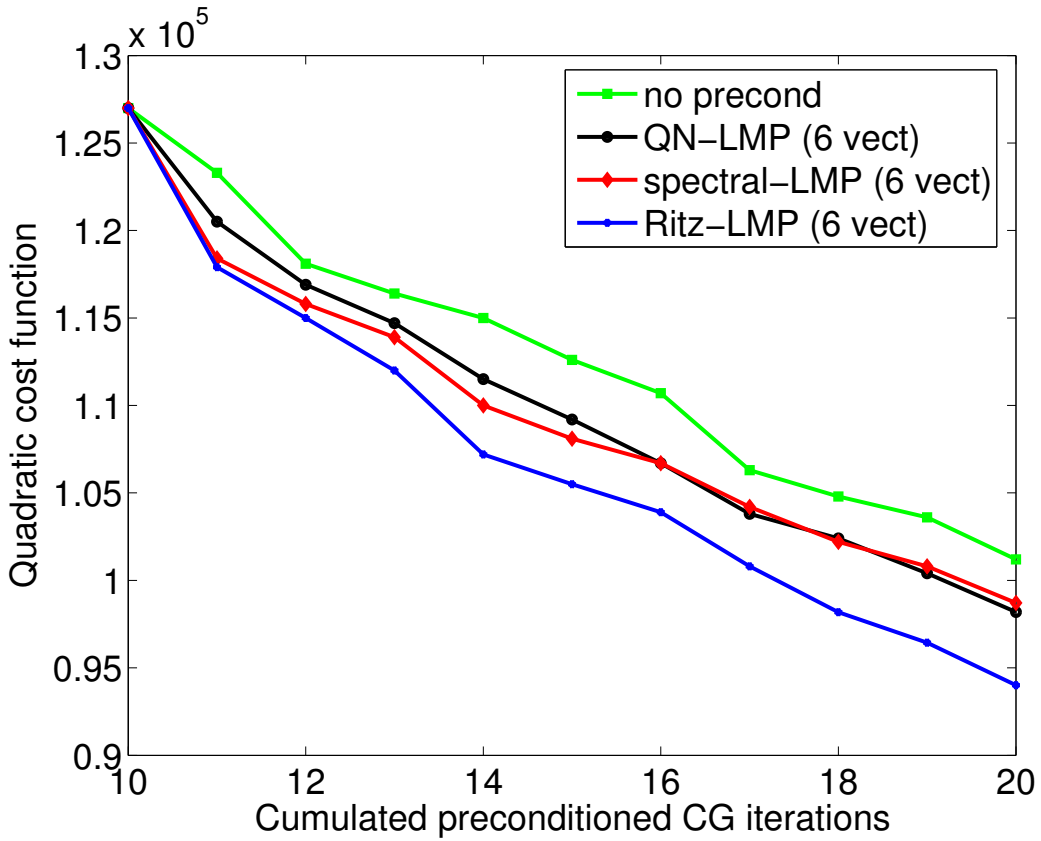
$\longrightarrow$ Positive and faster impact of an increase in the number of vectors in S

Ranking LMP (2 vectors)



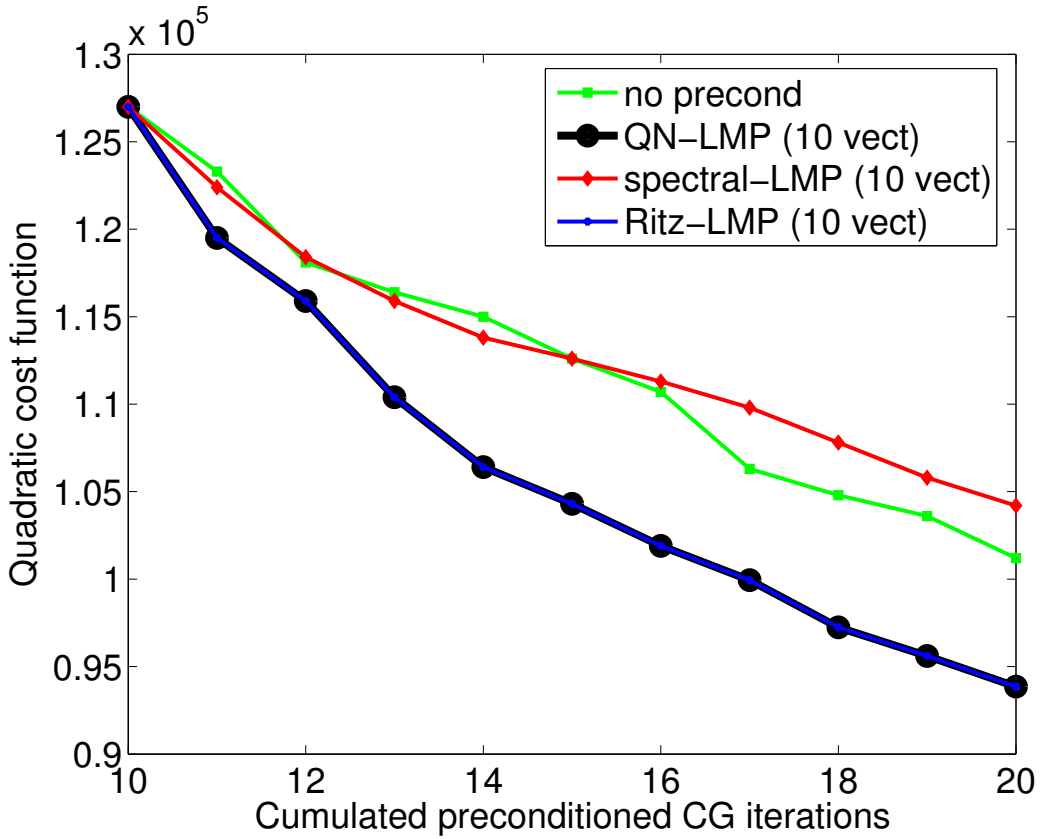
$\longrightarrow$ Inexact spectral-LMP $\equiv$ Ritz-LMP – Quasi-Newton LMP is worse

Ranking LMP (6 vectors)



$\longrightarrow$ Ritz-LMP is the best – Inexact spectral-LMP deteriorates

Ranking LMP (10 vectors)



$\longrightarrow$ Quasi-Newton LMP $\equiv$ Ritz-LMP

$\longrightarrow$ Inexact spectral-LMP even worse than no preconditioning

What about the first system (A_0)?

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

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

One-dimensional shallow water model

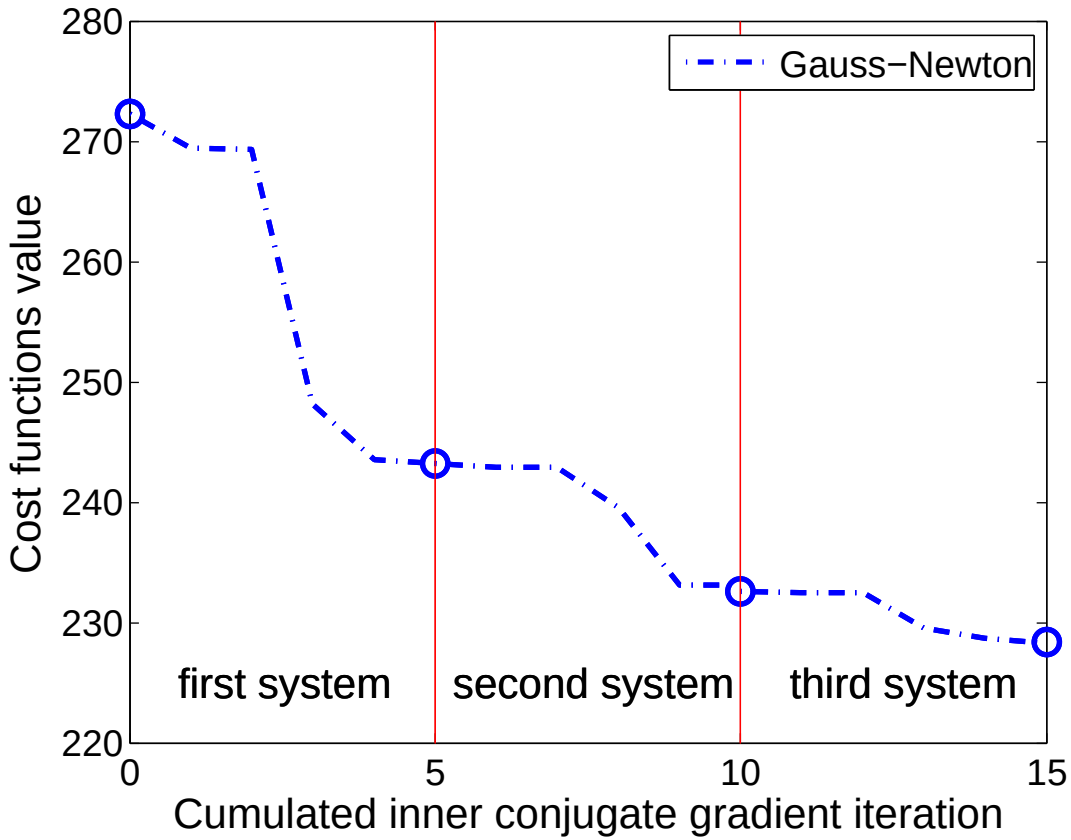
$\longrightarrow$ Estimate the velocity and geopotential of a fluid flow over an obstacle:

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

$\longrightarrow$ 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)

$\longrightarrow$ 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

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

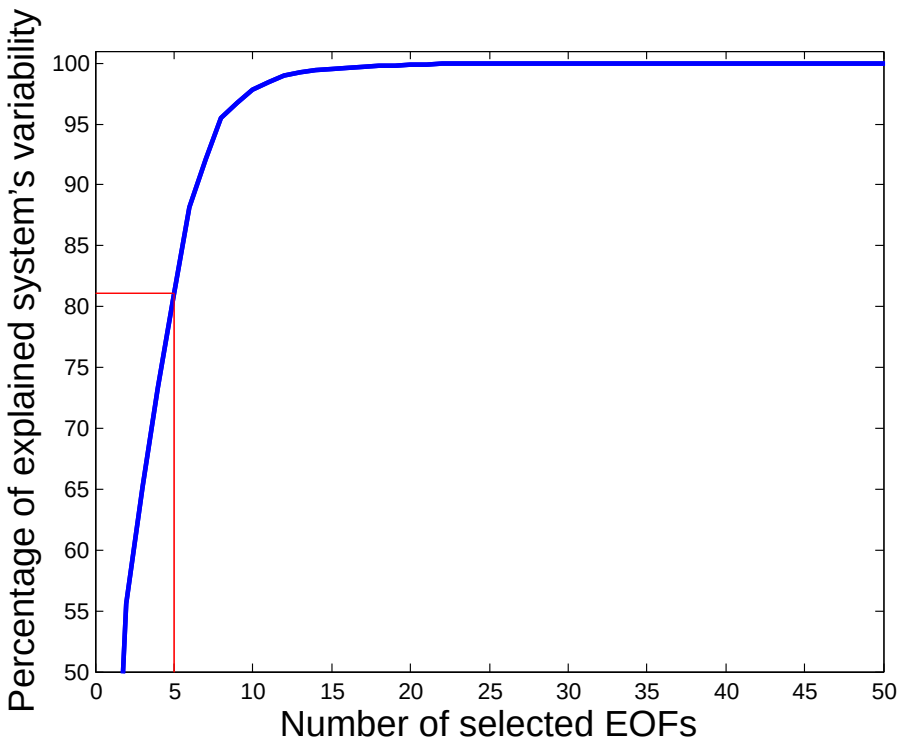
Choice for r

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



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

Ritz-Galerkin starting point

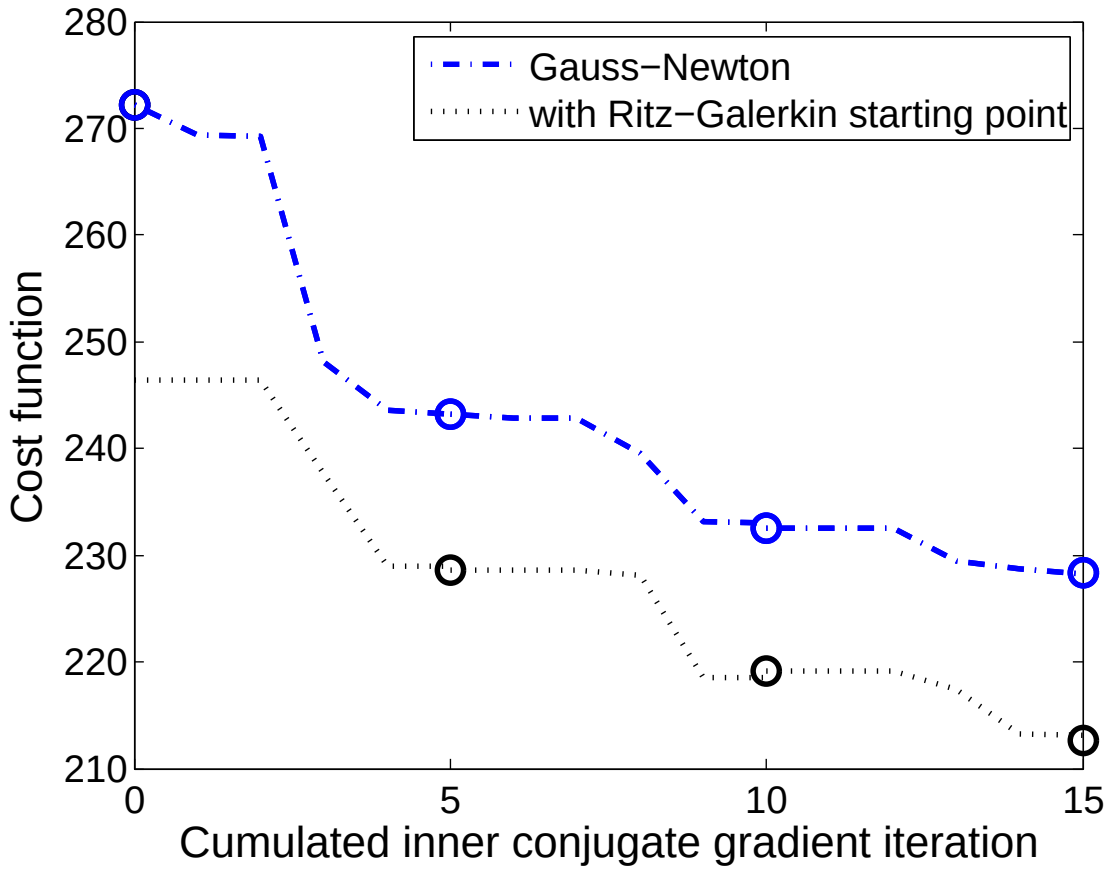
The solution of the first system in the subspace spanned by L:

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

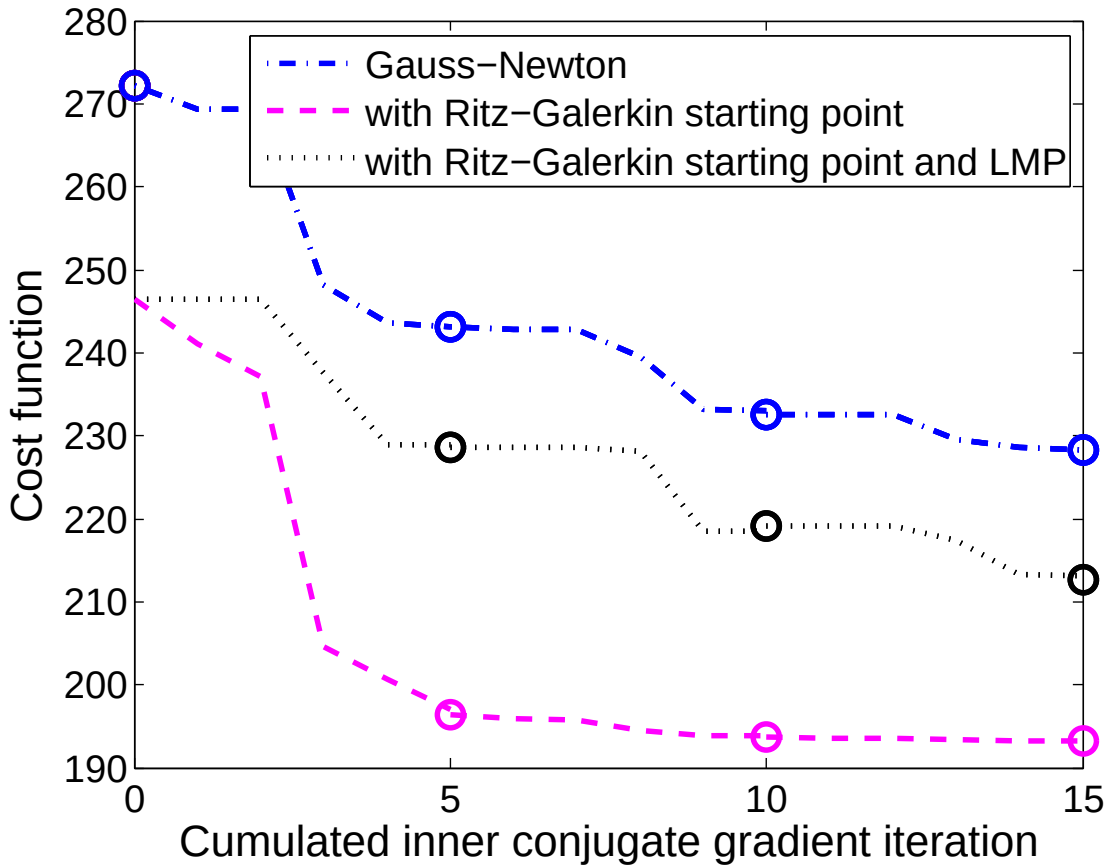
- is called the Ritz-Galerkin starting point
- is used as starting point in the CG for the first system $(A_0 x = b_0)$

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

First improvement



Second improvement



(Same H for the 3 systems)

Thank you for your attention !