Range space Krylov methods for data assimilation in meteorology and oceanography

Serge Gratton, Selime Gürol, Philippe Toint, Jean Tshimanga and Anthony Weaver

(philippe.toint@fundp.ac.be)

Namur Center for Complex Systems (naXys), University of Namur, Belgium

Lugano, May 2011

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 [observations],
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)

Predicting the state of the atmosphere, of the ocean

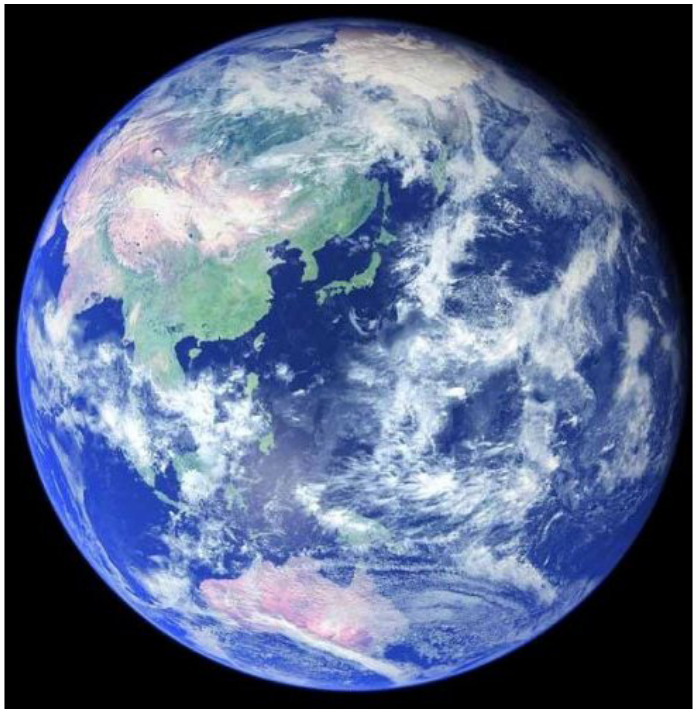
The state of the atmosphere or the ocean (the system) is characterized by state variables that are classically designated as fields:

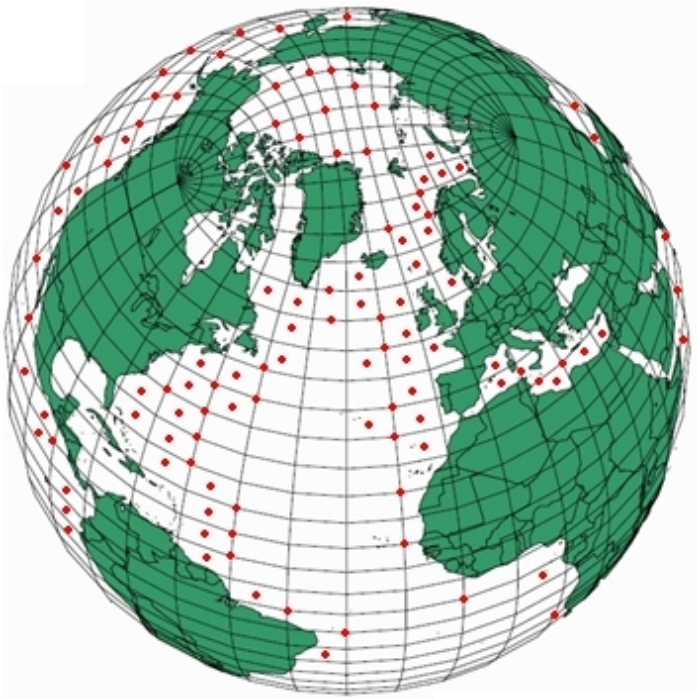
- velocity components
- pressure
- density
- temperature
- salinity

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, neutronic, hydrology, seismic,... (forecasting) problems. Involving large computers and nearly real-time computations.

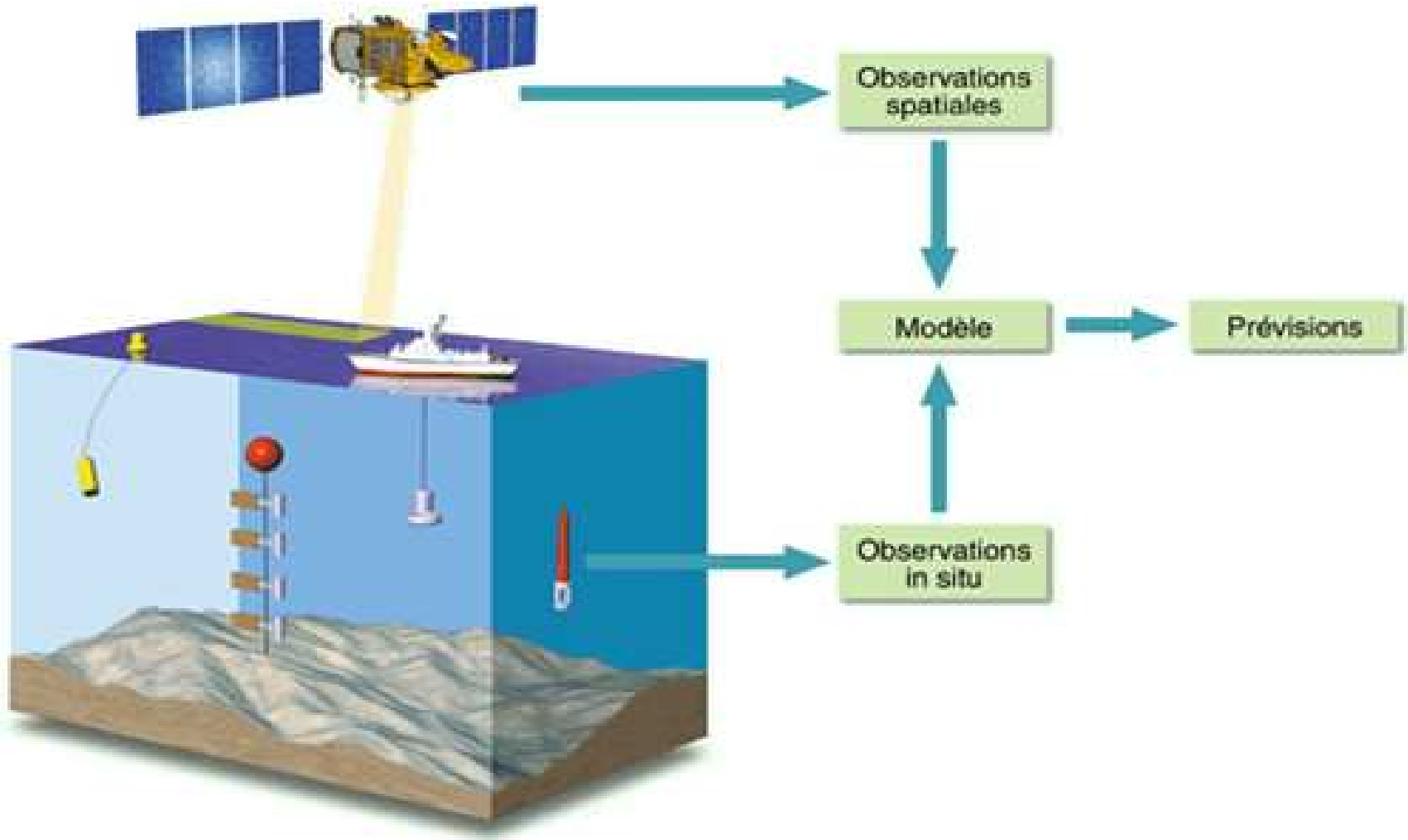
Predicting the state of the atmosphere of the ocean

Data: temperature, wind, pressure, ... everywhere and at all times!





Data collection



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

- Typical sizes would be for this problem 10^8 unknowns and 10^7 observations (Rabier, MTO)
- 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
- Furthermore, maintaining good parallel performance is just vital for 4D-Var, wrt stochastic methods. MOMA chantier : CNES, MTO, CERFACS, IRIT, IMT

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

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

Four-Dimensional Variational (4D-Var) formulation

$\rightarrow$ 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

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:

- Linearize 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 $\tilde{x}$

- Solve 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

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 discussions e.g. in the Ocean and Atmosphere communities (cfr P. Gauthier)
2. Relies on

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

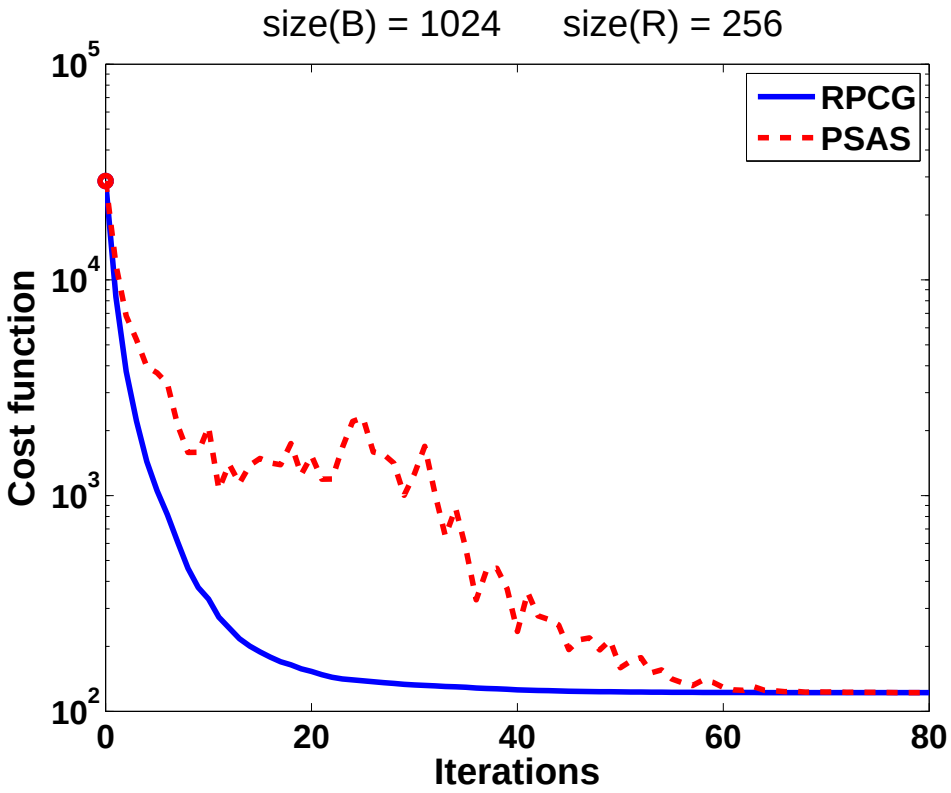
3. Iteratively solve

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

4. Set

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

Experiments



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. $\mathbf{v}_0 = \mathbf{x}^b - \mathbf{x}_0$.

$\rightarrow$ *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) : expensive

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) : right inner-product!

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. $\mathbf{t}_i = -\mathbf{w}_i + \beta_i\mathbf{t}_{i-1}$

Finding efficient preconditioners

$\rightarrow$ Limited Memory preconditioning! (Fisher (1998), Morales and Nocedal (2000), Tschimanga et al. (2008))

The idea is:

1. Formulate the limited memory Quasi-Newton matrix
2. Generate the preconditioner using the information from CG iterations.

- Want to find G that satisfies

$$FH^T = BH^TG$$

for a given F.

G as a Quasi-Newton warm-start preconditioner : Gilbert, Lemarechal, Nocedal, Byrd, Zhu

Formulation of F as a Quasi-Newton Limited Memory Preconditioner

$$F_{k+1} = (I - \tau_k p_k q_k^T) F_k (I - \tau_k q_k p_k^T) + \tau_k p_k p_k^T$$

p_k is the search direction
$\tau_k = 1/(q_k^T p_k)$
$q_k = (B^{-1} + H^T R^{-1} H) p_k$

ΔF_k defined by $\Delta F_k = F_{k+1} - F_k$, is the solution to the problem:

$$\min_{\Delta F_k} \left\| W^{1/2} \Delta F_k W^{1/2} \right\|_F$$

subject to $\Delta F_k = \Delta F_k^T, \quad F_{k+1} q_k = p_k$

Formulation for G as a Quasi-Newton Limited Memory Preconditioner

$$G_{k+1} = (I - \hat{\tau}_k \hat{p}_k (M\hat{q}_k)^T) G_k (I - \hat{\tau}_k \hat{q}_k \hat{p}_k^T M) + \hat{\tau}_k \hat{p}_k \hat{p}_k^T M$$

$M = HBH^T$, $\hat{p}_k$ is the search direction,
$\hat{q}_k = (I_m + R^{-1} HBH^T) \hat{p}_k$ and
$\hat{\tau}_k = 1/(\hat{q}_k^T HBH^T \hat{p}_k)$

ΔG_k defined by $\Delta G_k = G_{k+1} - G_k$ is the solution to the problem:

$$\min_{\Delta G_k} \left\| (WM)^{1/2} \Delta G_k (M^{-1} W)^{1/2} \right\|_F$$

subject to $M\Delta G_k = \Delta G_k^T M, \quad G_{k+1} \hat{q}_k = \hat{p}_k$

Computationally efficient RPCG algorithm using Quasi-Newton Preconditioner

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. $\widehat{\mathbf{l}}_i = \mathbf{H}\mathbf{B}\mathbf{H}^{\mathrm{T}}\widehat{\mathbf{r}}_i$
6. $\widehat{\mathbf{z}}_i = \mathbf{G}\widehat{\mathbf{r}}_i$
7. $\mathbf{w}_i = \mathbf{G}^{\mathrm{T}}\widehat{\mathbf{l}}_i$
8. $\beta_i = \mathbf{w}_i^{\mathrm{T}}\widehat{\mathbf{r}}_i / \mathbf{w}_{i-1}^{\mathrm{T}}\widehat{\mathbf{r}}_{i-1}$
9. $\widehat{\mathbf{p}}_i = -\widehat{\mathbf{z}}_i + \beta_i\widehat{\mathbf{p}}_{i-1}$
10. $\mathbf{t}_i = -\mathbf{w}_i + \beta_i\mathbf{t}_{i-1}$
11. $\mathbf{m}\mathbf{q}_{i-1} = (\mathbf{l}_{i-1} - \mathbf{l}_{i-2})/\alpha_{i-1}$

1. Consider a new vector l is defined as

$$l_i = HBH^T\widehat{r}_i$$

2. $\widehat{z}_i = G\widehat{r}_i$ and $w_i = HBH^T\widehat{z}_i$
3. HBH^TG is symmetric ($HFH^T = HBH^TG$)

$$w_i = HBH^TG\widehat{r}_i = G^THBH^T\widehat{r}_i = G^Tl_i$$

4. Multiply line 18 of RPCG ($\widehat{r}_i = \widehat{r}_{i-1} - \alpha_i\widehat{q}_i$) with HBH^T gives

$$HBH^T\widehat{q}_i = (l_i - l_{i-1})/\alpha_i$$

Convergence Properties

- If FA has eigenvalues $\mu_1 \leq \mu_2 \leq ... \leq \mu_n$, PCG algorithm with zero initial starting vector satisfies the inequality:

$$\|x_{k+1} - x^*\|_A \leq 2(\frac{\sqrt{\mu_n} - \sqrt{\mu_1}}{\sqrt{\mu_n} + \sqrt{\mu_1}})^k \|x^*\|_A$$

  where $A = B^{-1} + H^T R^{-1} H$

- If $G\widehat{A}$ has eigenvalues $\nu_1 \leq \nu_2 \leq ... \leq \nu_m$, RPCG with zero initial starting vector satisfies the inequality:

$$\|x_{k+1} - x^*\|_A \leq 2(\frac{\sqrt{\nu_m} - \sqrt{\nu_1}}{\sqrt{\nu_m} + \sqrt{\nu_1}})^k \|x^*\|_A$$

  where $\widehat{A} = I + R^{-1} H B H^T$

$$\|x_{k+1} - x^*\|_A \leq 2(\frac{\sqrt{\nu_m} - \sqrt{\nu_1}}{\sqrt{\nu_m} + \sqrt{\nu_1}})^k \|x^*\|_A \leq 2(\frac{\sqrt{\mu_n} - \sqrt{\mu_1}}{\sqrt{\mu_n} + \sqrt{\mu_1}})^k \|x^*\|_A$$

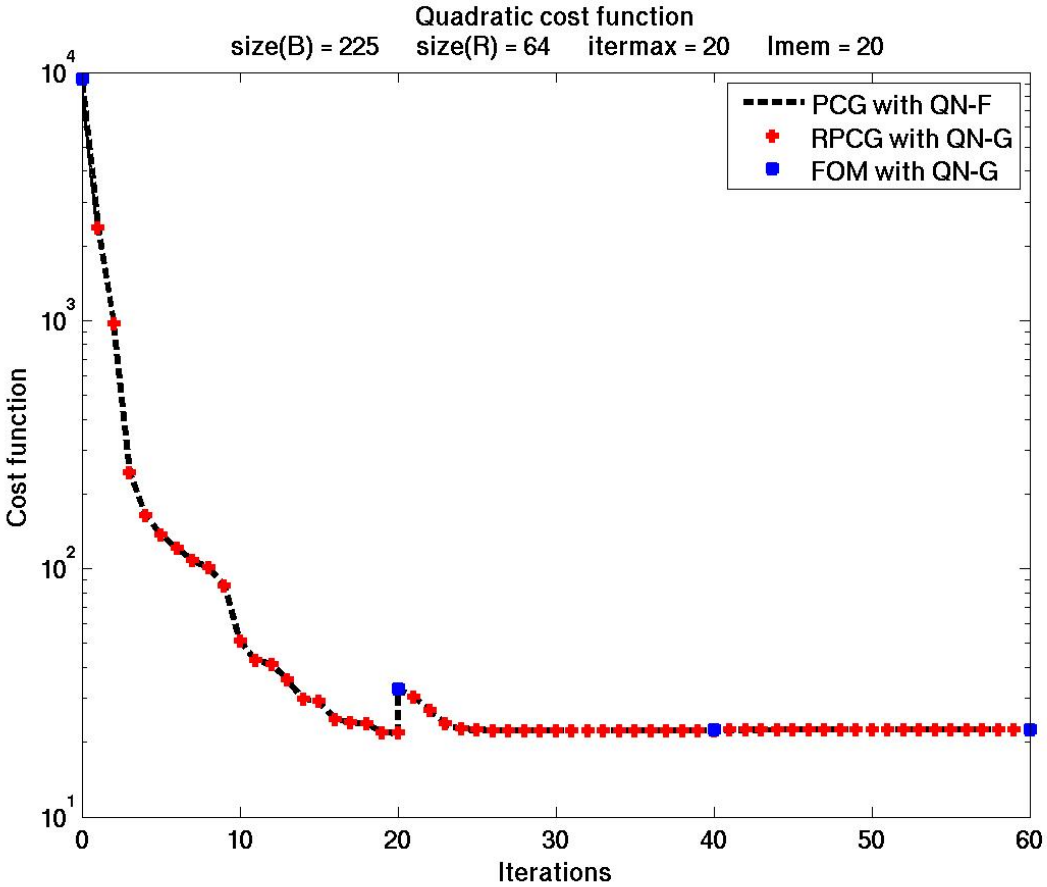
When H changes!

- When the observation operator H changes, $FH^T = BH^TG$ is not satisfied.

Solution: To re-generate G by using the recent HBH^T

- It is costly!
- We can approximate HBH^T (Using quasi-Newton formula!) and use this information to re-generate G. This is computationally efficient, but the system matrix is not symmetric with respect to the approximated inner product.
- We can use FOM algorithm which is a solver for unsymmetric systems.

Experiments



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 is possible to find G that satisfies $FH^T = BH^TG$ for a given F to accelerate convergence in dual space.
- 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 : CONGRAD vs M1QN3

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

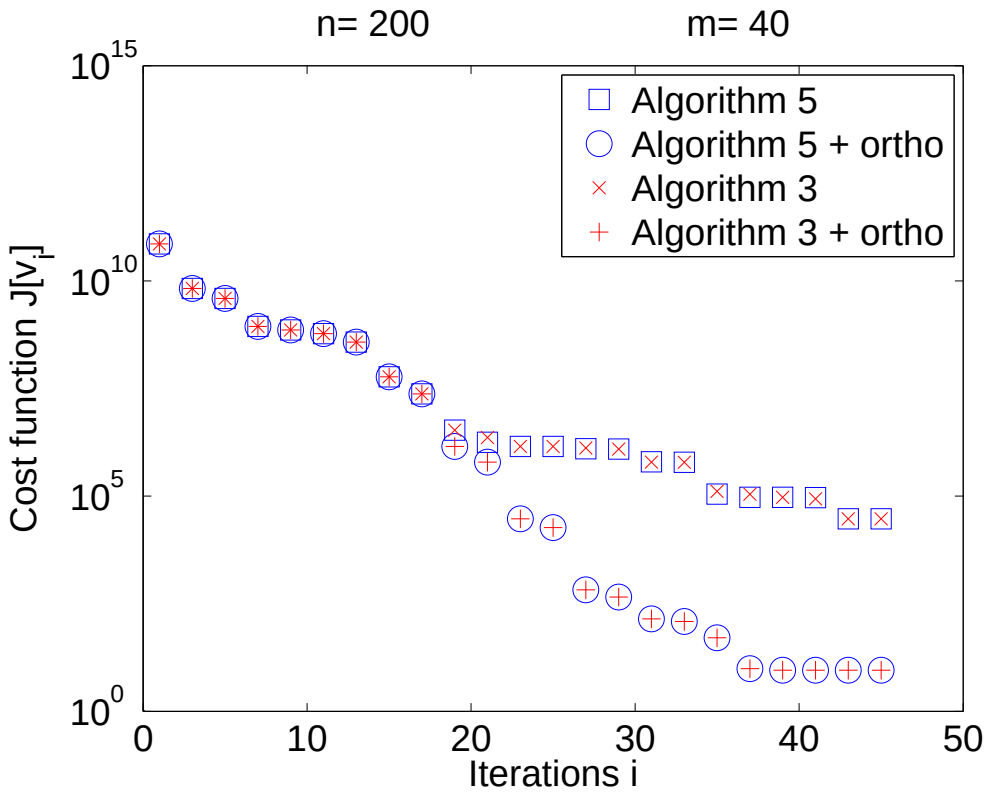
$$\mathbf{r}_i \leftarrow \prod_{j=1}^{i-1} \left(\mathbf{I}_n - \frac{\mathbf{r}_j \mathbf{r}_j^{\mathrm{T}}}{\mathbf{r}_j^{\mathrm{T}} \mathbf{F} \mathbf{r}_j} \right) \mathbf{r}_i .$$

2. We suggest the following re-orthogonalization scheme

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

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



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 another 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 monotonically minimize r_k while FOM and CG monotically minimize f_k along the iterations.

Range-space GMRES and FOM (RSGMR and RSFOM)

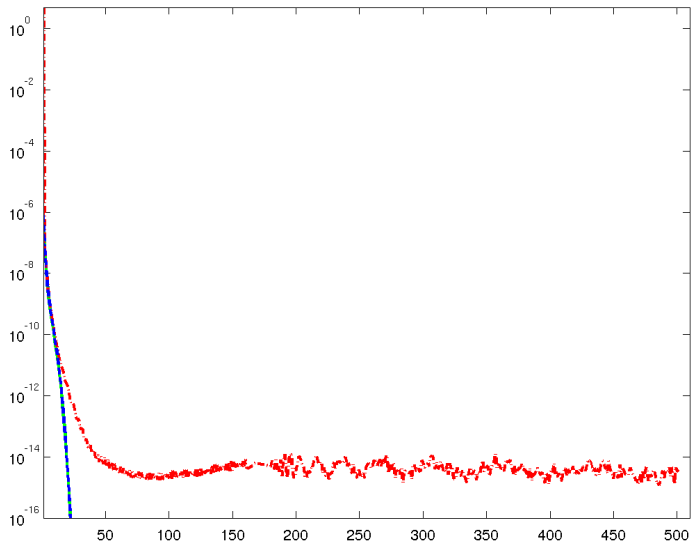
As CG may be rewritten in the observation space to yield RPCG, algorithms GMRES, MINRES and FOM may be rewritten to yields similar variants.

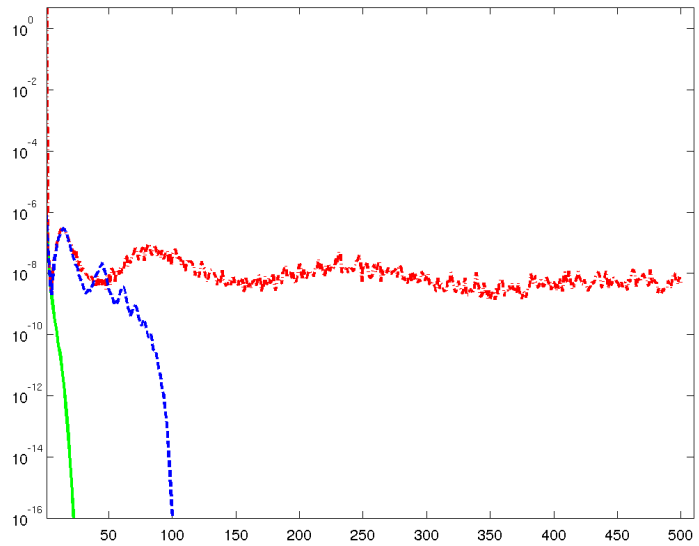
Why a range-space GMRES and FOM (RSGMR and RSFOM)?

- 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 $E_{L,i}, E_{K^T,i}, \quad$ and $\quad E_{K,i}$. 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

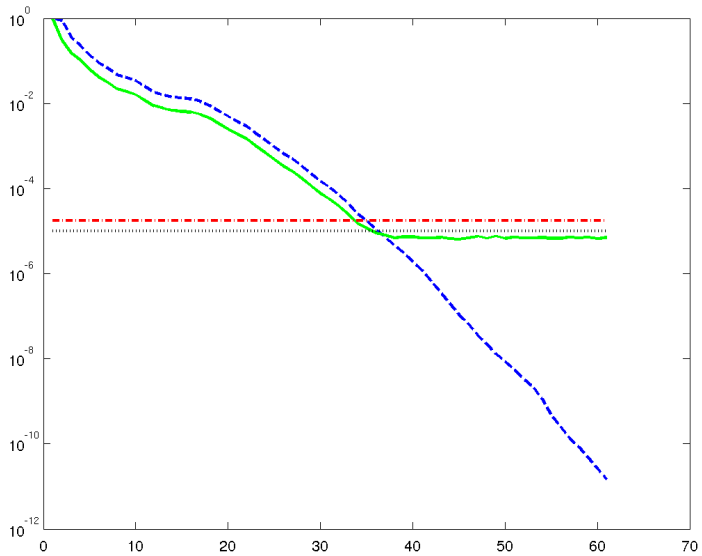
$$\begin{aligned} \|r_k\| &\leq \sqrt{2(k+1)}\,\|q_k\| + \sqrt{2}\left[\tau_* \gamma \sqrt{k}\|y_k\| + 4\,G\,\|K\| \sum_{i=1}^{k} |[y_k]_i|\,\tau_i\right] \\ &\leq \sqrt{2(k+1)}\left[\|q_k\| + \tau_{\max}\,(\gamma + 4\,G\,\|K\|)\|y_k\|\right]. \end{aligned}$$

Note in both sets of bounds:

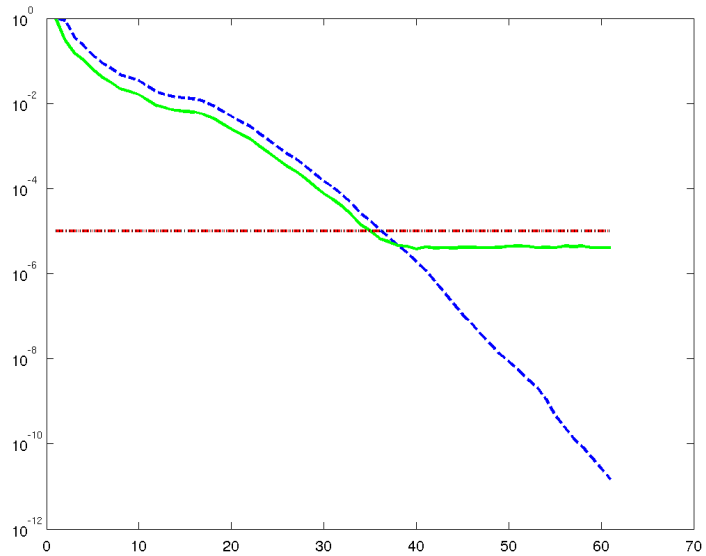
- The first of these bounds allows for variable accuray requirements
- special role of τ_*.

Error models (1)

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



Backward error model

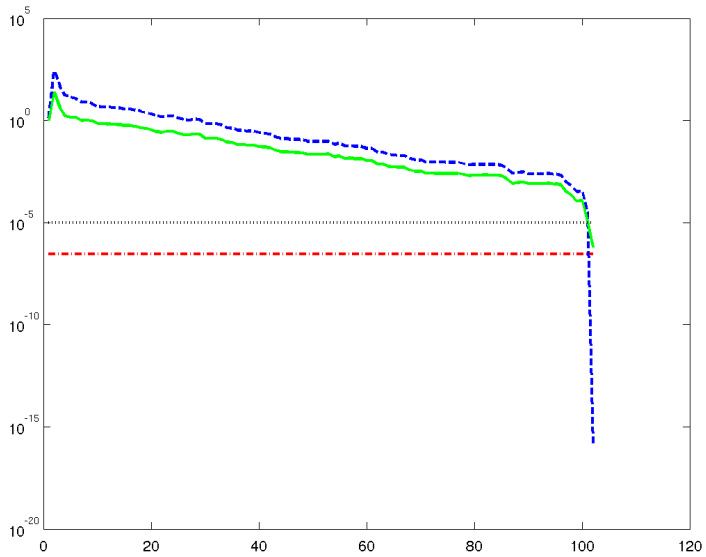


Forward error model

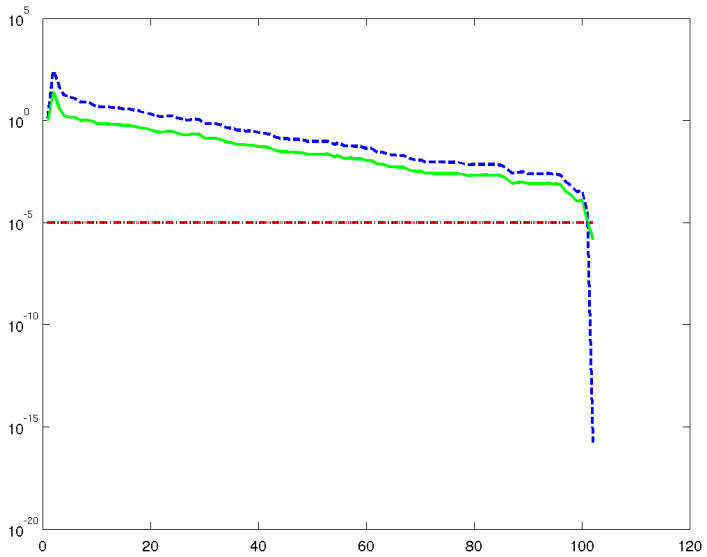
(normalized $\|r_k\|$, normalized $\|q_k\|$, accuracy threshold τ)

Error models (2)

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



Backward error model

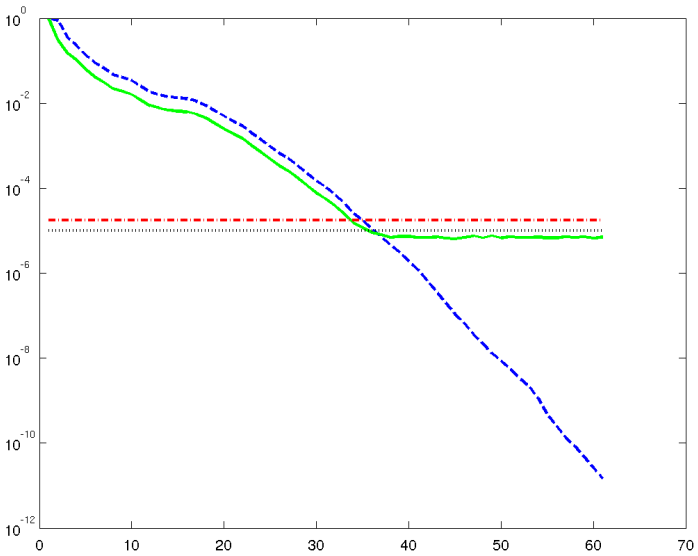


Forward error model

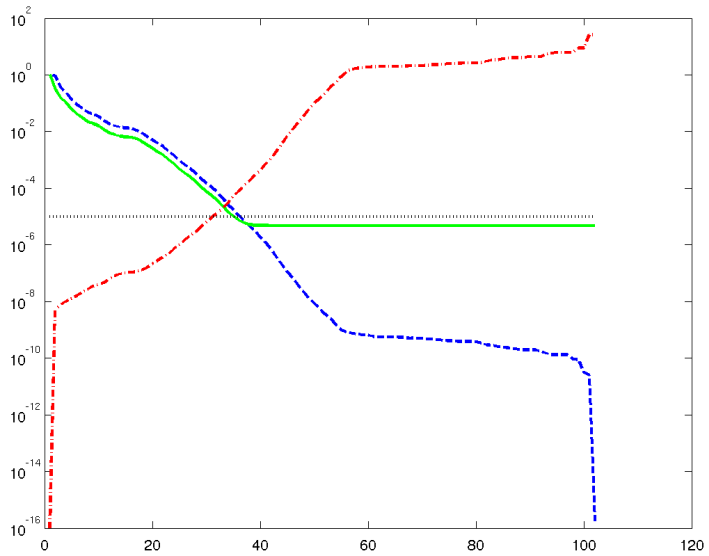
(normalized $\|r_k\|$, normalized $\|q_k\|$, accuracy threshold τ)

Fixed vs variable accuracy threshold (1)

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



Fixed τ

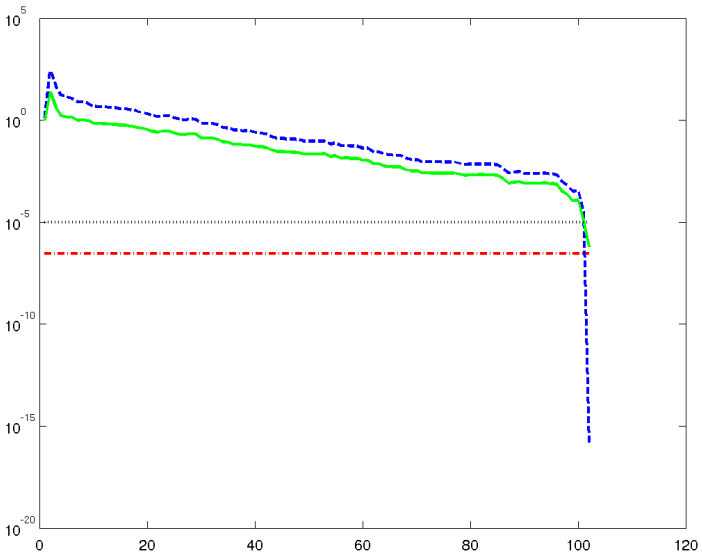


$\tau \approx 1/\|q_k\|$

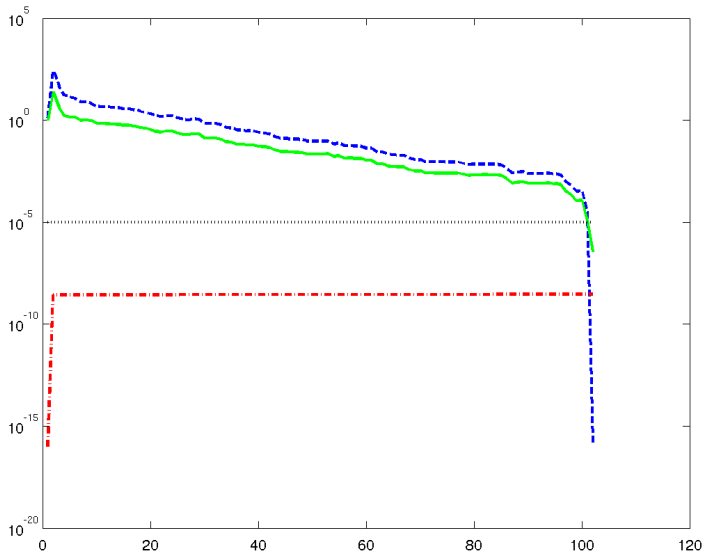
(normalized $\|r_k\|$, normalized $\|q_k\|$, accuracy threshold τ)

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!)

Thank you for your attention !