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

Oberwolfach, November 2012

Our questions

What are we going to discuss in this talk?

- How can we exploit the lower dimensionality of the observation space when using Krylov methods for solving the Incremental 4DVar method for data assimilation?
- Can we maintain good convergence guarantee in doing so ?
- Can the resulting algorithm be made efficient?
  (matrix vector products, preconditioning)
- Can it be made stable?

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

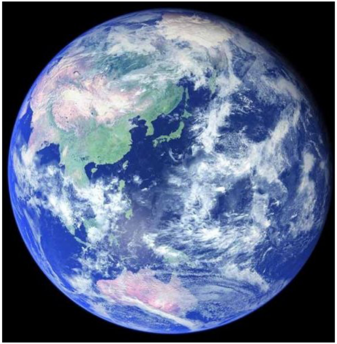
The state of the atmosphere or the ocean (the system) is characterized by state variables that are classically designated as fields:

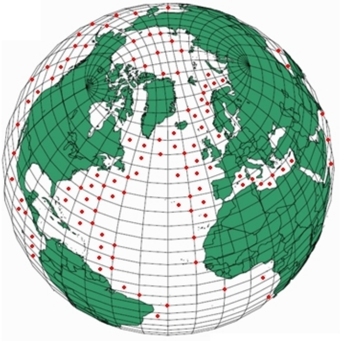
- velocity components
- pressure
- density
- temperature
- salinity

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.

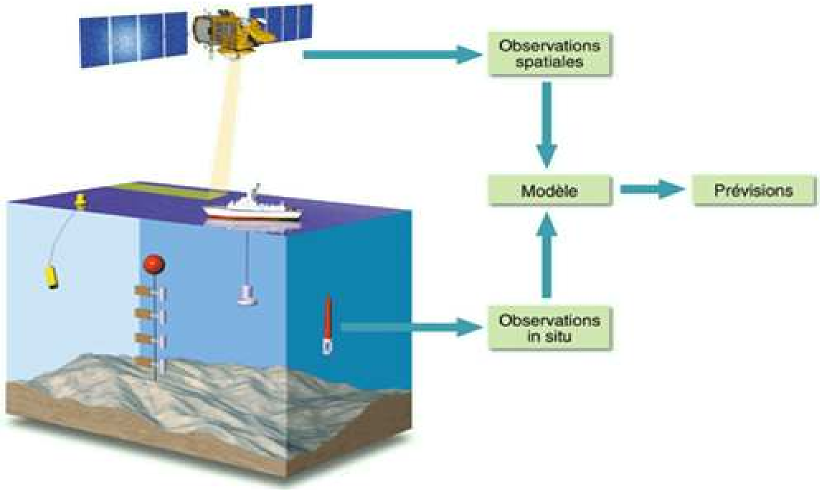
Predicting the state of the atmosphere of the ocean

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 (considered for the rest of the talk)

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

subject to

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

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.

Regularization technique

If all mappings involved in the problem were linear, the data assimilation problem would 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 introducing the background information $\mathbf{x}_0$

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

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

Rewrite the problem as:

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

Incremental 4D-Var = $\boxed{\text{inexact/truncated Gauss-Newton algorithm}}$

- Linearize ρ around the current iterate $\tilde{x}$ and solve

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

  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

Incremental 4D-Var Approach: algo overview

1. Transform the 4D-Var in a sequence of quadratic minimization problems
2. Increments $\delta x_0^{(k)}$ are (approximate) minimizers

$$J_k(\delta x_0) \;=\; \frac{1}{2}\|\delta x_0 - [x_b - x_0]\|^2_{B^{-1}} + \frac{1}{2}\|H\delta x_0 - d\|^2_{R^{-1}}$$

3. Perform the update

$$x_0^{(k+1)} = x_0^{(k)} + \delta x_0^{(k)}.$$

Inner minimization

Ignoring superscripts, minimizing

$$J(\delta x_0) = \frac{1}{2}\|\delta x_0 - [x_b - x_0]\|^2_{B^{-1}} + \frac{1}{2}\|H\delta x_0 - d\|^2_{R^{-1}}$$

amounts to iteratively solve

$$(B^{-1} + H^T R^{-1} H)\delta x_0 = B^{-1}(x_b - x_0) + H^T R^{-1} d.$$

whose exact solution is

$$x_b - x_0 + \left(B^{-1} + H^T R^{-1} H\right)^{-1} H^T R^{-1} \left(d - H(x_b - x_0)\right),$$

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

$$x_b - x_0 + BH^T \left(R + HBH^T\right)^{-1} \left(d - H(x_b - x_0)\right).$$

A first 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

$$x_b - x_0 + BH^T \left(R + HBH^T\right)^{-1} (d - H(x_b - x_0))$$

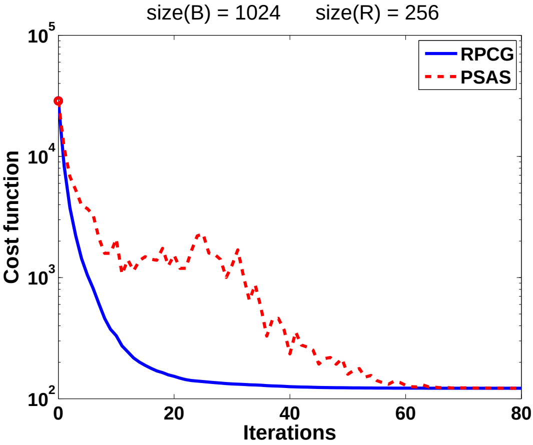
3. Iteratively solve

$$\left(I + R^{-1}HBH^T\right) w = R^{-1}(d - H(x_b - x_0)) \quad \text{for } w$$

4. Set

$$\delta x_0 = x_b - x_0 + BH^T w$$

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 : combine the advantages of both approaches:

- the variational property: enforce sufficient descent even when iterations are truncated.
- computational efficiency: work in the (dual) observation space whenever the number of observations is significantly smaller than the size of the state vector

Preserve 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 F
2. Suppose there exists $G^{m \times m}$ such that

$$F H^T = B H^T G$$

3. Ok for “exact” preconditioners:

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

Preconditioned CG on the Incremental 4D-Var

Initialization steps

Loop: WHILE

```
1. $q_{i-1} = A p_{i-1}$
2. $\alpha_{i-1} = r_{i-1}^T z_{i-1} \,/\, q_{i-1}^T p_{i-1}$
3. $v_i = v_{i-1} + \alpha_{i-1} p_{i-1}$
4. $r_i = r_{i-1} + \alpha_{i-1} q_{i-1}$
5. $z_i = F r_i$
6. $\beta_i = r_i^T z_i \,/\, r_{i-1}^T z_{i-1}$
7. $p_i = -z_i + \beta_i p_{i-1}$
```

Initialization steps

Loop: WHILE

```
1. $q_{i-1} = (H^T R^{-1} H + B^{-1}) p_{i-1}$
2. $\alpha_{i-1} = r_{i-1}^T z_{i-1} \,/\, q_{i-1}^T p_{i-1}$
3. $v_i = v_{i-1} + \alpha_{i-1} p_{i-1}$
4. $r_i = r_{i-1} + \alpha_{i-1} q_{i-1}$
5. $z_i = F r_i$
6. $\beta_i = r_i^T z_i \,/\, r_{i-1}^T z_{i-1}$
7. $p_i = -z_i + \beta_i p_{i-1}$
```

An useful observation

Suppose that

1. $BH^TG = FH^T$.
2. $v_0 \overset{\text{def}}{=} x_b - x_0$.

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

$$\begin{array}{rcl} r_i & = & H^T\widehat{r}_i, \\ p_i & = & BH^T\widehat{p}_i, \\ v_i & = & v_0 + BH^T\widehat{v}_i, \\ z_i & = & BH^T\widehat{z}_i, \\ q_i & = & H^T\widehat{q}_i \end{array}$$

Preconditioned CG on the Incremental 4D-Var (bis)

Initialization steps

given v_0; $r_0 = (H^T R^{-1} H + B^{-1}) v_0 - b, \ldots$

Loop: WHILE

1. $H^T \widehat{q}_{i-1} = H^T (R^{-1} H B^{-1} H^T + I_m) \widehat{p}_{i-1}$
2. $\alpha_{i-1} = r_{i-1}^T z_{i-1} \,/\, \widehat{q}_{i-1}^T \widehat{p}_{i-1}$
3. $BH^T \widehat{v}_i = BH^T (v_{i-1} + \alpha_{i-1} \widehat{p}_{i-1})$
4. $H^T \widehat{r}_i = H^T (r_{i-1} + \alpha_{i-1} \widehat{q}_{i-1})$
5. $BH^T \widehat{z}_i = FH^T \widehat{r}_i = BH^T G \widehat{r}_i$
6. $\beta_i = (r_i^T z_i \,/\, r_{i-1}^T z_{i-1})$
7. $BH^T \widehat{p}_i = BH^T (-\widehat{z}_i + \beta_i \widehat{p}_{i-1})$

Restricted PCG (version 1) : expensive

Initialization steps

given v_0; $r_0 = (H^T R^{-1} H + B^{-1}) v_0 - b, \ldots$

Loop: WHILE

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

More transformations

1. Consider w and t defined by

$$w_i = HBH^T \widehat{z}_i \quad \text{and} \quad t_i = HBH^T \widehat{p}_i$$

2. From Restricted PCG (version 1)

$$t_i = \begin{cases} -w_0 & \text{if } i = 0, \\ -w_i + \beta_i t_{i-1} & \text{if } i > 0, \end{cases}$$

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

Restricted PCG (version 2) : correct inner-product!

Initialization steps

Loop: WHILE

```
1. $\widehat{q}_{i-1} = R^{-1} t_{i-1} + \widehat{p}_{i-1}$
2. $\alpha_{i-1} = w_{i-1}^T \widehat{r}_{i-1} \,/\, \widehat{q}_{i-1}^T t_{i-1}$
3. $\widehat{v}_i = \widehat{v}_{i-1} + \alpha_{i-1} \widehat{p}_{i-1}$
4. $\widehat{r}_i = \widehat{r}_{i-1} + \alpha_{i-1} \widehat{q}_{i-1}$
5. $\widehat{z}_i = G \widehat{r}_i$
6. $w_i = HBH^T \widehat{z}_i$
7. $\beta_i = w_i^T \widehat{r}_i \,/\, w_{i-1}^T \widehat{r}_{i-1}$
8. $\widehat{p}_i = -\widehat{z}_i + \beta_i \widehat{p}_{i-1}$
9. $t_i = -w_i + \beta_i t_{i-1}$
```

Recovering primal-space quantities

Attention : The (final) primal increment v_i is computed using

$$v_i = v_0 + BH^T \widehat{v}_i,$$

→ one more product by BH^T !

Ok . . . but what about preconditioning?

Finding efficient preconditioners

→ Limited Memory preconditioning! (Fisher (1998), Morales and Nocedal (2000), Tshimanga 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

$$F H^T = B H^T G$$

for a given F.

G as a Quasi-Newton warm-start preconditioner

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
$q_k = (B^{-1} + H^T R^{-1} H) p_k$ and $\tau_k = 1/(q_k^T p_k)$

$\Delta F_k = F_{k+1} - F_k$ solves of the problem:

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

subject to

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

G as a Quasi-Newton Limited Memory Preconditioner

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

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

$\Delta G_k = G_{k+1} - G_k$ solves the problem:

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

subject to

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

An efficient RPCG algorithm with Quasi-Newton Preconditioner

Loop: WHILE

1. $\widehat{q}_{i-1} = R^{-1} t_{i-1} + \widehat{p}_{i-1}$
2. $\alpha_{i-1} = w_{i-1}^T \widehat{r}_{i-1} \,/\, \widehat{q}_{i-1}^T t_{i-1}$
3. $\widehat{v}_i = \widehat{v}_{i-1} + \alpha_{i-1} \widehat{p}_{i-1}$
4. $\widehat{r}_i = \widehat{r}_{i-1} + \alpha_{i-1} \widehat{q}_{i-1}$
5. $\widehat{l}_i = HBH^T \widehat{r}_i$
6. $\widehat{z}_i = G \widehat{r}_i$
7. $w_i = G^T \widehat{l}_i$
8. $\beta_i = w_i^T \widehat{r}_i \,/\, w_{i-1}^T \widehat{r}_{i-1}$
9. $\widehat{p}_i = -\widehat{z}_i + \beta_i \widehat{p}_{i-1}$
10. $t_i = -w_i + \beta_i t_{i-1}$
11. $mq_{i-1} = (l_{i-1} - l_{i-2})/\alpha_{i-1}$

Further manipulations. . .

1. Consider

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

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

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

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 $\mu_1 \leq \mu_2 \leq \ldots \leq \mu_n$, PCG algorithm with zero initial starting vector satisfies the inequality:

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

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

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

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

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

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

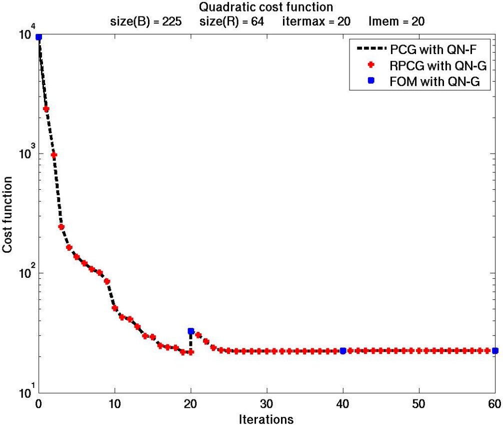
Nonlinear problems: H changes!

- When the observation operator H changes, $FH^T = BH^T G$ no longer holds.

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 (fixes are possible. . .)
- We can use the FOM algorithm (a solver for unsymmetric systems).

Experiments



Comments

Th main features of RPCG:

It amounts to solving the observation system using PCG with the correct 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 $FH^T = BH^T G$ for a given F to accelerate convergence in the dual space.

The corresponding algorithm can be rearranged to contain a single occurrence of the matrix-vector products by B, H, H^T and R^{-1} per iteration.

Perpectives at this stage

- Behaviour in presence of round-off error ?
- Find other efficient preconditioners F such that

$$FH^T = BH^T G$$

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

Towards further reduction of the cost

- RPCG allows memory and computational cost reduction whenever the number of observation is smaller than the size of the control vector
- 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, ...)

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

$$b,\ (\gamma I_n + K^T L)b,\ (\gamma I_n + K^T L)^2 b,\ (\gamma I_n + K^T L)^3 b,\ \ldots \tag{3.1}$$

or, equivalently, by

$$b,\ (K^T L)b,\ (K^T L)^2 b,\ (K^T L)^3 b,\ \ldots \tag{3.2}$$

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,

$$K^T L\, V_k = V_{k+1} H_k, \tag{3.3}$$

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, we have

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

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

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

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

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

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

Let

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

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)

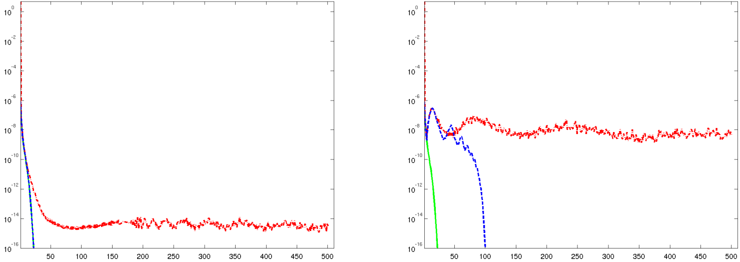
As CG may be rewritten in the observation space to yield RPCG, algorithms GMRES, MINRES and FOM may be rewritten to yields similar variants.

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

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

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

- Backward:

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

- Forward:

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

Results for the backward error model

Define

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

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

(... after some analysis ...)

Theorem

Assume the backward-error model. Then

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

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

Results for the forward error model

Theorem

Assume the forward-error model. Then

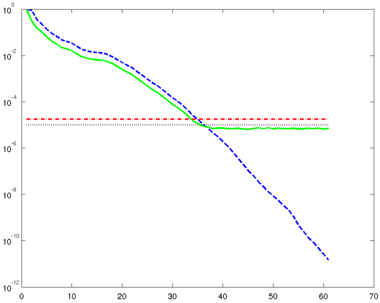
$$\begin{aligned}\|r_k\| &\le \sqrt{2(k+1)}\,\|q_k\| + \sqrt{2}\left[\tau_*\gamma\sqrt{k}\|y_k\| + 4\,G\,\|K\|\sum_{i=1}^{k}|[y_k]_i|\,\tau_i\right]\\ &\le \sqrt{2(k+1)}\left[\|q_k\| + \tau_{\max}\,(\gamma + 4\,G\,\|K\|)\|y_k\|\right].\end{aligned}$$

Note in both sets of bounds:

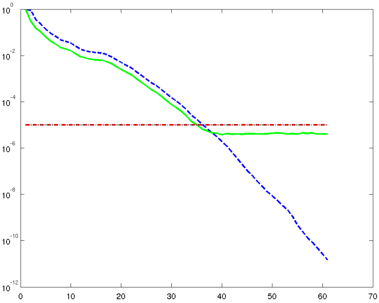
- The first of these bounds allows for variable accuray requirements
- special role of τ_*.

Error models (1)

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



Backward error model

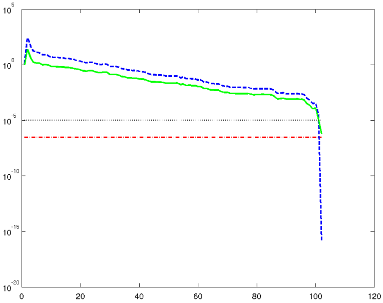


Forward error model

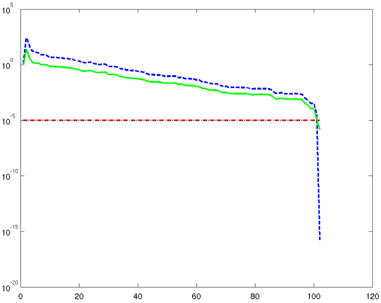
(normalized $\|r_k\|$, normalized $\|q_k\|$, accuracy threshold τ)

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)

Further conclusions

Dual space methods may be designed to gain from low rank, ie when observations are substantially fewer than states

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 !