Preconditioning of conjugate-gradients in observation space for 4D-VAR

S. Gratton[1] S. Gurol[2] Ph.L. Toint[3]

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

The 9th Workshop on Adjoint Model Applications in Dynamic Meteorology

Outline

4D-Var problem: Formulation

$\rightarrow$ 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:

- $x \in \mathbb{R}^n$ is the control variable
- The observations y_j and the background x_b are noisy
- $\mathcal{M}_j$ are model operators
- $\mathcal{H}_j$ are observation operators
- B is the covariance background error matrix
- R_j are covariance observation error matrices

4D-Var problem: Formulation

→ Large-scale nonlinear weighted least-squares problem:

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

Typically solved by a standard Gauss-Newton method known as Incremental 4D-Var in data assimilation community

1. Solve linearized subproblem at iteration k

$$\min_{\delta x\in\mathbb{R}^n} J(\delta x) \quad = \quad \frac{1}{2}\|\delta x - [x_b - x]\|^2_{B^{-1}} + \frac{1}{2}\|H\delta x - d\|^2_{R^{-1}}$$

Sequence of quadratic minimization problems

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

4D-Var problem: Solution

- From optimality condition

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

- The aim is to solve sequences of this linear system.
- Solution algorithms: Krylov subspace methods
- Exact solution writes:

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

Krylov subspace methods

$$\underbrace{(B^{-1} + H^T R^{-1} H)}_{A}\,\delta x = \underbrace{B^{-1}(x_b - x) + H^T R^{-1} d}_{b}$$

- Krylov subspace methods searchs for an approximate solution δx_l from a subspace $\delta x_0 + \mathcal{K}^l(A, r_0)$ where

$$\mathcal{K}^l(A, r_0) = span\left\{r_0, Ar_0, A^2 r_0, ..., A^{l-1} r_0\right\}, r_0 = b - A\delta x_0$$

- Krylov subspace methods impose the Petrov-Galerkin condition

$$r_k \perp \mathcal{L}^l(A, r_0).$$

- A is symmetric and positive definite
- $\mathcal{L}^l(A, r_0) = \mathcal{K}^l(A, r_0)$ →Lanczos, Conjugate Gradient (CG)
  →FOM (unsymmetric case for further reference)
  →minimizes $\|r_k\|_{A^{-1}}$
- $\mathcal{L}^l(A, r_0) = A\mathcal{K}^l(A, r_0)$ →MINRES
  →minimizes $\|r_k\|_2$

Krylov subspace methods

Which one to use when A is symmetric and positive definite?

- Less computational cost and memory
- Efficient preconditioning
- Efficient re-orthogonalization
- Convergence behaviour
- ...

We focus on **Lanczos and CG**

- They are implemented in the realistic applications.
- It is possible to use preconditioners. It is possible to avoid $B^{1/2}$.
- It is possible to use re-orthogonalization.
- CG is globally convergent when using the Steihaug-Toint truncated conjugate gradient trust region method

Preconditioned Lanczos algorithm ($F^{1/2}$ is not required!)

For $i = 1, 2, ..., l$

1. $w_i = (B^{-1} + H^T R^{-1} H) z_i$ → Construction of the Krylov sequence
2. $w_i = w_i - \beta_i v_{i-1}$
3. $\alpha_i = \langle w_i, z_i \rangle$ →Orthogonalization
4. $w_{i+1} = w_i - \alpha_i v_i$
5. $z_{i+1} = F w_{i+1}$ →Apply preconditioner
6. $\beta_{i+1} = \langle z_{i+1}, w_{i+1} \rangle^{1/2}$
7. $v_{i+1} = w_{i+1} / \beta_{i+1}$ →Normalization
8. $z_{i+1} = z_{i+1} / \beta_{i+1}$
9. $V = [V, v_{i+1}]$ →Orthonormal basis for Krylov subspace
10. $T_{i,i} = \alpha_i$; $T_{i+1,i} = T_{i,i+1} = \beta_{i+1}$ →Generate the tridiagonal matrix

Solution

1. $y_l = T_l^{-1} \beta_0 e_1$ →Impose the condition $r_k \perp \mathcal{K}^l(A, r_0)$
2. $\delta x_l = F V_l y_l$ →Find the approximate solution

Preconditioned CG algorithm ($F^{1/2}$ is not required!)

Initialization

- $r_0 = A\delta x_0 - b,\ z_0 = Fr_0,\ p_0 = z_0$

For $i = 0, 1, ...$

1. $q_i = (B^{-1} + H^T R^{-1} H)p_i$
2. $\alpha_i = < r_i, z_i > / < q_i, p_i >$ →Compute the step-length
3. $\delta x_{i+1} = \delta x_i + \alpha_i p_i$ →Update the iterate
4. $r_{i+1} = r_i - \alpha_i q_i$ →Update the residual
5. $r_{i+1} = r_{i+1} - RZ^T r_{i+1}$ →Re-orthogonalization
6. $z_{i+1} = Fr_{i+1}$ →Update the preconditioned residual
7. $\beta_i = < r_{i+1}, z_{i+1} > / < r_i, z_i >$ →Ensure A-conjugate directions
8. $R = [R, r/\beta_i]$ →Re-orthogonalization
9. $Z = [Z, z/\beta_i]$ →Re-orthogonalization
10. $p_{i+1} = z_{i+1} + \beta_i p_i$ →Update the descent direction

Alternative Methods: Dual Approaches

Can we reduce the computational cost?

Dual Approaches!

While using the dual approaches is it possible to

- keep the convergence behaviour?
- apply efficient preconditioners?
- apply re-orthogonalization?

Dual Approach

- Exact solution writes

$$x_b - x + \underbrace{\left(B^{-1} + H^T R^{-1} H\right)^{-1} H^T R^{-1} (d - H(x_b - x))}_{\delta v \in \mathbf{R}^n, n \approx 10^7}$$

or equivalently using the Sherman-Morrison-Woodbury formula or duality theory

$$x_b - x + BH^T \underbrace{(R^{-1} HBH^T + I)^{-1} R^{-1} (d - H(x_b - x))}_{\lambda \in \mathbf{R}^m, m \approx 10^5}$$

- Performing inner minimization in $\mathbf{R}^m$ hopefully reduces memory and computational cost !

Dual approach

Minimization in dual space

1. Iteratively solve

$$(I_m + R^{-1}HBH^T)\lambda = R^{-1}(d - H(x_b - x))$$

2. Set $\delta x = x_b - x + BH^T\lambda$

- PSAS algorithm (Courtier 1997): PCG on this linear system with R inner product
- RPCG algorithm (Gratton and Tschimanga 2009): PCG on this linear system with HBH^T inner product
- RLanczos algorithm: Lanczos on this linear system with HBH^T inner product

Dual Approach: RPCG and PSAS algorithm

Initialization

$\lambda_0 = 0$, $\widehat{r}_0 = R^{-1}(d - H(x_b - x))$,
$\widehat{z}_0 = G\widehat{r}_0$, $\widehat{p}_1 = \widehat{z}_0$, $k = 1$

Loop on k

1. $\widehat{q}_i = \widehat{A}\widehat{p}_i$
2. $\alpha_i = < \widehat{r}_{i-1}, \widehat{z}_{i-1} >_M \; / < \widehat{q}_i, \widehat{p}_i >_M$
3. $\lambda_i = \lambda_{i-1} + \alpha_i \widehat{p}_i$
4. $\widehat{r}_i = \widehat{r}_{i-1} - \alpha_i \widehat{q}_i$
5. $\beta_i = < \widehat{r}_{i-1}, \widehat{z}_{i-1} >_M \; / < \widehat{r}_{i-2}, \widehat{z}_{i-2} >_M$
6. $\widehat{z}_i = G\widehat{r}_i$
7. $\widehat{p}_i = \widehat{z}_{i-1} + \beta_i \widehat{p}_{i-1}$

- $\widehat{A} = R^{-1}HBH^T + I_m$
- G is the preconditioner.
- M is the inner-product.
- PSAS Algorithm: $M = R$ cheap matvec
- RPCG Algorithm: $M = HBH^T$ expensive matvec (model integration is required)
- G should be symmetric w.r.t. to M

Dual Approach: Precond. RLanczos algorithm and PSAS

For $i = 1, 2, ..., l$

1. $\widehat{w}_i = (I + R^{-1}HBH^T)z_i$
2. $\widehat{w}_i = \widehat{w}_i - \beta_i \widehat{v}_{i-1}$
3. $\alpha_i = \langle \widehat{w}_i, z_i \rangle_M$
4. $\widehat{w}_{i+1} = \widehat{w}_i - \alpha_i \widehat{v}_i$
5. $\widehat{z}_{i+1} = G\widehat{w}_{i+1}$
6. $\beta_{i+1} = \langle \widehat{z}_{i+1}, \widehat{w}_{i+1} \rangle_M^{1/2}$
7. $\widehat{v}_{i+1} = \widehat{w}_{i+1} / \beta_{i+1}$
8. $\widehat{z}_{i+1} = \widehat{z}_{i+1} / \beta_{i+1}$
9. $\widehat{V} = [\widehat{V}, \widehat{v}_{i+1}]$
10. $T_{i,i} = \alpha_i$; $T_{i+1,i} = T_{i,i+1} = \beta_{i+1}$

Solution

1. $y_l = T_l^{-1} \beta_0 e_1$
2. $\delta x_l = BH^T V_l y_l$

- $M = HBH^T$
- G is the preconditioner
- When $M = R$, the iterates are mathematically equivalent to that of PSAS

Dual approach: Primal equivalent algorithms

- Assume that

$$r_0 \in \text{range}(H^T)$$

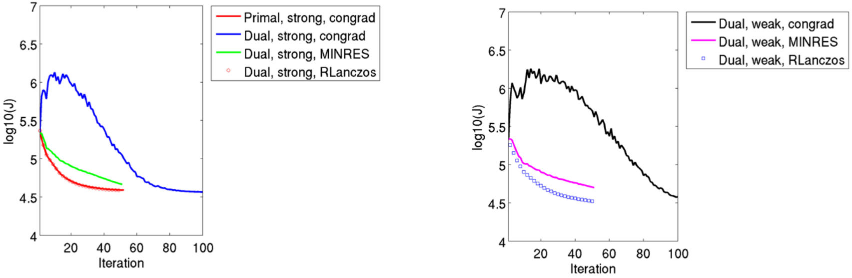
$$FH^T = BH^T G$$

  where F is the preconditioner in primal space and G is the preconditioner in dual space.

- Rlanczos, RPCG, CONGRAD, **PCG and Lanczos method in the primal space** are mathematically equivalent to each other.
- MINRES is not equivalent, it minimizes $\|r_k\|_2$!

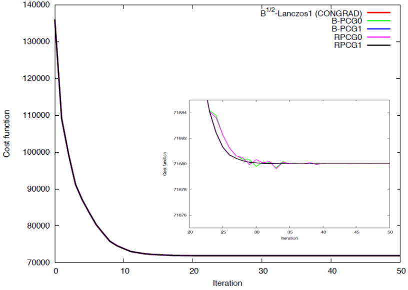
Results for ROMS

- Observations: SST (Sea Surface Temperature) and SSH(Sea Surface Height) observations from satellites. Sub-surface hydrographic observations from floats.
- Number of observations (m): 10^5
- Number of state variables (n): 10^6 for strong constraint and 10^7 for weak constraint.
- Computation: 64 CPUs



Results for 3D-VAR FGAT NEMOVAR

- Observations: Temperature, unbalanced salinity, unbalanced sea surface height
- Number of observations (m): 2×10^5
- Number of state variables (n): 8×10^6
- Computation: 8 processors are used



Second level preconditioning

- Use an approximation of the Hessian of the quadratic problem $\rightarrow$ Limited Memory preconditioning (Fisher (1998), Morales and Nocedal (2000), Tschimanga, Gratton, Sartenaer, Weaver (2008))

The idea is:

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

- For equivalence with the primal method, find G that satisfies

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

for a given F

- For now, assume that H is not changing for each outer loop.

G as a Quasi-Newton warm-start preconditioner

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

Formulation for 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,
$\widehat{\tau}_k = 1/(\widehat{q}_k^T HBH^T \widehat{p}_k)$
$\widehat{q}_k = (I_m + R^{-1} HBH^T) \widehat{p}_k$

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. $\widehat{\lambda}_i = \widehat{\lambda}_{i-1} + \alpha_{i-1}\widehat{\mathbf{p}}_{i-1}$
4. $\widehat{\mathbf{r}}_i = \widehat{\mathbf{r}}_{i-1} - \alpha_{i-1}\widehat{\mathbf{q}}_{i-1}$
5. $\widehat{\mathbf{l}}_i = \mathbf{HBH}^{\mathrm{T}}\widehat{\mathbf{r}}_i$
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. $\mathbf{mq}_{i-1} = (\mathbf{l}_{i-1} - \mathbf{l}_{i-2})/\alpha_{i-1}$

---

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 the expression $\widehat{r}_i = \widehat{r}_{i-1} - \alpha_i\widehat{q}_i$ with HBH^T gives

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

Convergence Properties

- If FA has eigenvalues $\mu_1 \le \mu_2 \le ... \le \mu_n$, PCG algorithm satisfies the inequality:

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

- If $G\widehat{A}$ has eigenvalues $\nu_1 \le \nu_2 \le ... \le \nu_m$, RPCG satisfies the inequality:

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

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

- Same iterates, but tighter bound on convergence rate with the dual approach
- Improvement of tightness can be arbitrarily large on purposely chosen problems

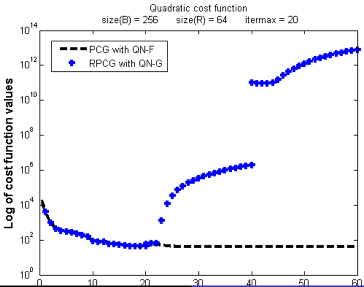
When H changes!

- When H changes in nonlinear iterations, $FH^T = BH^TG$ is not satisfied. The preconditioner is not symmetric anymore wrt HBH^T and perturbed CG is in trouble.

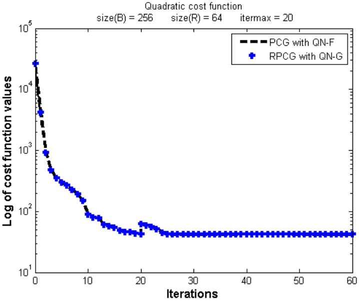
Expression for G

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

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



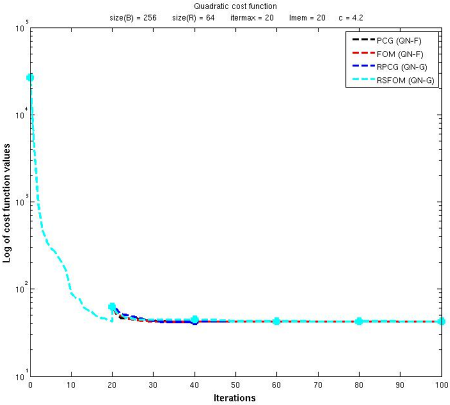
- Straigthforward approach: re-generate G by using the recent p_k and HBH^T: costs one matvec per preconditioning pair



- Accept to handle non symmetry : use FOM algorithm

Solutions (2/2)

- Use FOM with Quasi-Newton preconditioner G where approximated M is used.
- Approximation with the Davidon Fletcher Powell (DFP) formula.



Conclusions

- We have dual space methods RPCG and RLanczos that generate the same iterates as PCG and Lanczos in primal space
- $B^{1/2}$ operator is not required with the proposed primal solvers for B preconditioning
- RPCG and RLanczos were implemented in realistics systems: NEMOVAR thanks to Anthony Weaver and Andrea Piacentini, ROMS thanks to Andy Moore.
- Preconditioning is possible: find G such that $FH^T = BH^TG$

Thank you for your attention !