Preconditioning least-squares problems using duality with an application to data assimilation

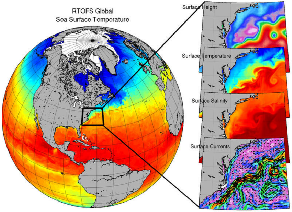
S. Gratton[1,2] S. Gürol[2] Ph.L. Toint[3]

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

7 September 2012

Outline

Examples on data assimilation systems



- **Ocean forecasting**
  → In-situ data and space-based observations are combined with the dynamical model of the ocean system in order to forecast the ocean state parameters, e.g. salinity, temperature, sea surface height.

- **Automatic landing systems on aircraft in poor weather conditions**
  → Noisy data on the aircarft's position must be combined with information on how the aircraft responds to movements of wing surfaces in order to achieve a smooth landing.

- Weather forecast, satellite orbit determination, aerosol estimation, gravity field estimation,...

Problem formulation

→ **Large-scale regularized nonlinear least-squares** problem:

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

where:

- $x \in \mathbb{R}^n$ is the vector of control variables (e.g. the model initial conditions)
- $y_j \in \mathbb{R}^m$ is the observation vector over a given time window
- x_b is the background estimate of the control vector
- $\mathcal{M}_{0,j}$ is the dynamical model predicting the state of the system at time t_j
- $\mathcal{H}_j$ maps the control vector x_j in model space to observation space
- B is an estimate of the background error covariance matrix
- R_j is an estimate of the observation error covariance matrix at time t_j

Solution algorithm

→ **Large-scale regularized nonlinear least-squares** problem:

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

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

Solution algorithm

→ **Large-scale regularized nonlinear least-squares** problem:

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

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

1. Solve the **linearized subproblem** at iteration k

$$\min_{\delta x_k \in \mathbb{R}^n} J(\delta x_k) = \frac{1}{2}||\delta x_k - x_b + x_k||^2_{B^{-1}} + \frac{1}{2}||G_k \delta x_k - d_k||^2_{R^{-1}}$$

Solution algorithm

→ **Large-scale regularized nonlinear least-squares** problem:

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

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

1. Solve the **linearized subproblem** at iteration k

$$\min_{\delta x_k \in \mathbb{R}^n} J(\delta x_k) = \tfrac{1}{2}\|\delta x_k - x_b + x_k\|^2_{B^{-1}} + \tfrac{1}{2}\|G_k \delta x_k - d_k\|^2_{R^{-1}}$$

2. Perform update $x_{k+1} = x_k + \delta x_k$

Solution with primal approach

- From optimality condition

$$\underbrace{\left(B^{-1} + {G_k}^T R^{-1} G_k\right)}_{A_k} \delta x_k = \underbrace{B^{-1}(x_b - x_k) + {G_k}^T R^{-1} d_k}_{b_k}$$

where the linear system is a **large-scale** system and A is **symmetric** and **positive-definite**.

Solution with primal approach

- From optimality condition

$$\underbrace{\left(B^{-1} + {G_k}^T R^{-1} G_k\right)}_{A_k} \delta x_k = \underbrace{B^{-1}(x_b - x_k) + {G_k}^T R^{-1} d_k}_{b_k}$$

where the linear system is a **large-scale** system and A is **symmetric** and **positive-definite**.

- **Exact solution** can be written as

$$x_b - x_k + \underbrace{\left(B^{-1} + {G_k}^T R^{-1} G_k\right)^{-1} {G_k}^T R^{-1} \left(d_k - G_k(x_b - x_k)\right)}_{\text{requires solving a linear system } \textbf{iteratively} \text{ in } \mathbb{R}^n}$$

Solution with primal approach

- From optimality condition

$$\underbrace{\left(B^{-1} + {G_k}^T R^{-1} G_k\right)}_{A_k} \delta x_k = \underbrace{B^{-1}(x_b - x_k) + {G_k}^T R^{-1} d_k}_{b_k}$$

where the linear system is a **large-scale** system and A is **symmetric** and **positive-definite**.

- **Exact solution** can be written as

$$x_b - x_k + \underbrace{\left(B^{-1} + {G_k}^T R^{-1} G_k\right)^{-1} {G_k}^T R^{-1} \left(d_k - G_k(x_b - x_k)\right)}_{\text{requires solving a linear system } \textbf{iteratively} \text{ in } \mathbb{R}^n}$$

- Solution algorithm: **Preconditioned Conjugate Gradient method (PCG)**

Solution with primal approach

- From optimality condition

$$\underbrace{\left(B^{-1} + {G_k}^T R^{-1} G_k\right)}_{A_k} \delta x_k = \underbrace{B^{-1}(x_b - x_k) + {G_k}^T R^{-1} d_k}_{b_k}$$

where the linear system is a **large-scale** system and A is **symmetric** and **positive-definite**.

- **Exact solution** can be written as

$$x_b - x_k + \underbrace{\left(B^{-1} + {G_k}^T R^{-1} G_k\right)^{-1} {G_k}^T R^{-1} \left(d_k - G_k(x_b - x_k)\right)}_{\text{requires solving a linear system \textbf{iteratively} in } \mathbb{R}^n}$$

- Solution algorithm: **Preconditioned Conjugate Gradient method (PCG)**

  $\rightarrow$ Preconditioning with the quasi-Newton Limited Memory Preconditioner (Morales and Nocedal 2000) (Gratton, Sartenaer and Tshimanga 2011)

Exploiting the structure: Dual Approach

- Alternatively, the **exact solution** can be rewritten from duality theory or using Sherman-Morrison-Woodbury formula

$$x_b - x_k + BG_k^T \underbrace{(G_k B G_k^T + R)^{-1}(d_k - G_k(x_b - x_k))}_{\text{requires solving a linear system } \textbf{iteratively} \text{ in } \mathbb{R}^m}$$

Exploiting the structure: Dual Approach

- Alternatively, the **exact solution** can be rewritten from duality theory or using Sherman-Morrison-Woodbury formula

$$x_b - x_k + BG_k^T \underbrace{(G_k B G_k^T + R)^{-1}(d_k - G_k(x_b - x_k))}_{\text{requires solving a linear system } \textbf{iteratively} \text{ in } \mathbb{R}^m}$$

- If $m << n$, then performing the minimization in $\mathbb{R}^m$ can **reduce memory and computational cost**.

Exploiting the structure: Dual Approach

- Alternatively, the **exact solution** can be rewritten from duality theory or using Sherman-Morrison-Woodbury formula

$$x_b - x_k + BG_k^T \underbrace{(G_k B G_k^T + R)^{-1}(d_k - G_k(x_b - x_k))}_{\text{requires solving a linear system \textbf{iteratively} in } \mathbb{R}^m}$$

- If $m << n$, then performing the minimization in $\mathbb{R}^m$ can **reduce memory and computational cost**.

Minimization in dual space with PCG requires

1. **special inner product** $(G_k B G_k^T)$ for preserving the monotonic decrease on quadratic cost
2. **augmentation** for accommodating b_k in the right space

Dual approach

Minimization in dual space

1. **Iteratively** solve

$$(I_m + \tilde{R}^{-1}\tilde{G}_k B \tilde{G}_k{}^T)\lambda = \tilde{d}$$

2. **Set** $\delta x_k = x_b - x_k + B\tilde{G}_k{}^T\lambda$

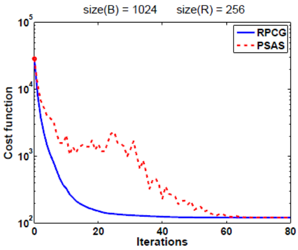
Dual approach

Minimization in dual space

1. **Iteratively** solve

$$(I_m + \tilde{R}^{-1}\tilde{G}_k B \tilde{G}_k{}^T)\lambda = \tilde{d}$$

2. **Set** $\delta x_k = x_b - x_k + B\tilde{G}_k{}^T \lambda$



- **PSAS** (Courtier 1997): Preconditioned CG (PCG) with $\tilde{R}$ inner product.
- **RPCG** (Gratton and Tshimanga 2009): PCG with $\tilde{G}_k B \tilde{G}_k^T$ inner product.
  $\rightarrow$ It generates the **same iterates** as those generated by the **primal approach.**

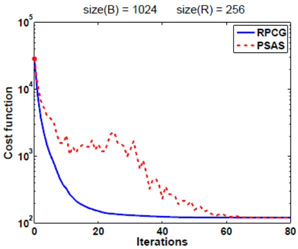
Dual approach

Minimization in dual space

1. **Iteratively** solve

$$(I_m + \tilde{R}^{-1}\tilde{G}_k B \tilde{G}_k{}^T)\lambda = \tilde{d}$$

2. **Set** $\delta x_k = x_b - x_k + B\tilde{G}_k{}^T\lambda$



- **PSAS** (Courtier 1997): Preconditioned CG (PCG) with $\tilde{R}$ inner product.
- **RPCG** (Gratton and Tshimanga 2009): PCG with $\tilde{G}_k B \tilde{G}_k^T$ inner product.
  → It generates the **same iterates** as those generated by the **primal approach.**

→ Cost function: $J(\delta x_k) = \frac{1}{2}\|\delta x_k - x_b + x_k\|^2_{B^{-1}} + \frac{1}{2}\|G_k \delta x_k - d_k\|^2_{R^{-1}}$

Preconditioning in dual space

- It is **possible to maintain** the one-to-one correspondance between primal and dual iterates, under the assumption that

$$F_{k-1}\tilde{G}_k^T = B\tilde{G}_k^T\tilde{K}_{k-1}$$

where F_{k-1} is a preconditioner for a primal solver and $\tilde{K}_{k-1}$ is a preconditioner for a dual solver (Gratton and Tshimanga 2009).

- The preconditioner $\tilde{K}_{k-1}$ needs to be **symmetric** in $\tilde{G}_k B \tilde{G}_k^T$ inner product.

- For **linear case**, Gratton, Gurol and Toint (2012) derive the quasi-Newton LMP in dual space which generates **mathematically equivalent iterates** to those of primal approach.

The quasi-Newton LMP in dual space (Linear case)

- The **quasi-Newton LMP**: The descent directions p_i, $i = 1, ..., l$ generated by a CG method are used.

The quasi-Newton LMP in dual space (Linear case)

- The **quasi-Newton LMP**: The descent directions p_i, $i = 1, ..., l$ generated by a CG method are used.

$$F_i = \left(I_n - \frac{p_i p_i^T A}{p_i^T A p_i} \right) F_{i-1} \left(I_n - \frac{A p_i p_i^T}{p_i^T A p_i} \right) + \frac{p_i p_i^T}{p_i^T A p_i},$$

$\rightarrow F = F_l$.

The quasi-Newton LMP in dual space (Linear case)

- The **quasi-Newton LMP**: The descent directions p_i, $i = 1, ..., l$ generated by a CG method are used.

$$F_i = \left(I_n - \frac{p_i p_i^T A}{p_i^T A p_i} \right) F_{i-1} \left(I_n - \frac{A p_i p_i^T}{p_i^T A p_i} \right) + \frac{p_i p_i^T}{p_i^T A p_i},$$

$\rightarrow F = F_l$.

- The corresponding quasi-Newton preconditioner in dual space is given as

$$K_i = \left(I_m - \frac{\widehat{p}_i \widehat{p}_i^T \widehat{A} C}{\widehat{p}_i^T \widehat{A} C \widehat{p}_i} \right) K_{i-1} \left(I_m - \frac{\widehat{A} \widehat{p}_i \widehat{p}_i^T C}{\widehat{p}_i^T \widehat{A} C \widehat{p}_i} \right) + \frac{\widehat{p}_i \widehat{p}_i^T C}{\widehat{p}_i^T \widehat{A} C \widehat{p}_i}$$

where $i = 1, ..., l$, $C = GBG^T$, $\widehat{A} = I_m + R^{-1}GBG^T$ and $\widehat{p}_i$ is the search direction.
$\rightarrow K = K_l$.

The quasi-Newton LMP in dual space (Linear case)

- The **quasi-Newton LMP**: The descent directions p_i, $i = 1, ..., l$ generated by a CG method are used.

$$F_i = \left(I_n - \frac{p_i p_i^T A}{p_i^T A p_i} \right) F_{i-1} \left(I_n - \frac{A p_i p_i^T}{p_i^T A p_i} \right) + \frac{p_i p_i^T}{p_i^T A p_i},$$

$\rightarrow F = F_l$.

- The corresponding quasi-Newton preconditioner in dual space is given as

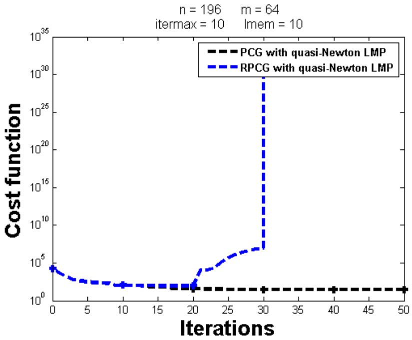
$$K_i = \left(I_m - \frac{\widehat{p}_i \widehat{p}_i^T \widehat{A} C}{\widehat{p}_i^T \widehat{A} C \widehat{p}_i} \right) K_{i-1} \left(I_m - \frac{\widehat{A} \widehat{p}_i \widehat{p}_i^T C}{\widehat{p}_i^T \widehat{A} C \widehat{p}_i} \right) + \frac{\widehat{p}_i \widehat{p}_i^T C}{\widehat{p}_i^T \widehat{A} C \widehat{p}_i}$$

where $i = 1, ..., l$, $C = GBG^T$, $\widehat{A} = I_m + R^{-1} GBG^T$ and $\widehat{p}_i$ is the search direction.
$\rightarrow K = K_l$.

$\rightarrow$ This preconditioner satisfies the relation: $FG^T = BG^T K$ and it is symmetric in the C inner product (Gratton, Gurol and Toint 2012).

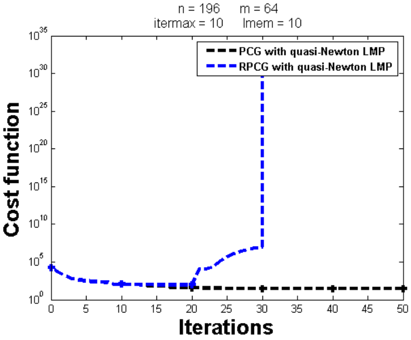
The quasi-Newton LMP in dual space (Nonlinear case)

- For **nonlinear case**, inheriting the previous preconditioner may not be possible!



The quasi-Newton LMP in dual space (Nonlinear case)

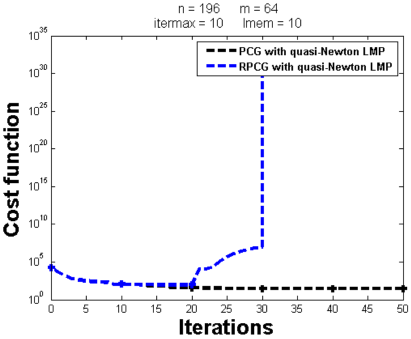
- For **nonlinear case**, inheriting the previous preconditioner may not be possible!



- Why?

The quasi-Newton LMP in dual space (Nonlinear case)

- For **nonlinear case**, inheriting the previous preconditioner may not be possible!

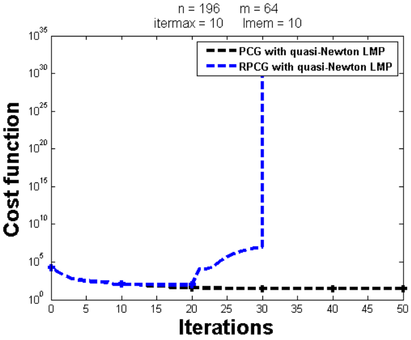


- Why?

  → The **augmented matrix** $\tilde{G}_k$ **changes** along the outer iterations (k) because of the nonlinearity and varying augmentation term.

The quasi-Newton LMP in dual space (Nonlinear case)

- For **nonlinear case**, inheriting the previous preconditioner may not be possible!



- Why?

  → The **augmented matrix** $\tilde{G}_k$ **changes** along the outer iterations (k) because of the nonlinearity and varying augmentation term.

  → This causes **loss of symmetry** for the preconditioner $\tilde{K}_{k-1}$ with respect to the current inner product $\tilde{G}_k B \tilde{G}_k^T$

The quasi-Newton LMP in dual space (Nonlinear case)

Solution:

- **Re-generate** the preconditioner using the current inner product.

The quasi-Newton LMP in dual space (Nonlinear case)

Solution:

- **Re-generate** the preconditioner using the current inner product.

$$K_i = \left(I_m - \frac{\widehat{p}_i \widehat{p}_i^T \widehat{A} C}{\widehat{p}_i^T \widehat{A} C \widehat{p}_i} \right) K_{i-1} \left(I_m - \frac{\widehat{A} \widehat{p}_i \widehat{p}_i^T C}{\widehat{p}_i^T \widehat{A} C \widehat{p}_i} \right) + \frac{\widehat{p}_i \widehat{p}_i^T C}{\widehat{p}_i^T \widehat{A} C \widehat{p}_i}$$

where $i = 1, ..., l$, $C = GBG^T$, $\widehat{A} = I_m + R^{-1} GBG^T$ and $\widehat{p}_i$ is the search direction.

The quasi-Newton LMP in dual space (Nonlinear case)

Solution:

- **Re-generate** the preconditioner using the current inner product.

$$K_i = \left(I_m - \frac{\widehat{p}_i \widehat{p}_i^T \widehat{A} C}{\widehat{p}_i^T \widehat{A} C \widehat{p}_i} \right) K_{i-1} \left(I_m - \frac{\widehat{A} \widehat{p}_i \widehat{p}_i^T C}{\widehat{p}_i^T \widehat{A} C \widehat{p}_i} \right) + \frac{\widehat{p}_i \widehat{p}_i^T C}{\widehat{p}_i^T \widehat{A} C \widehat{p}_i}$$

where $i = 1, ..., l$, $C = GBG^T$, $\widehat{A} = I_m + R^{-1}GBG^T$ and $\widehat{p}_i$ is the search direction.

$\rightarrow$ It is **costly** for large-scale problems.

The quasi-Newton LMP in dual space (Nonlinear case)

Solution:

- **Re-generate** the preconditioner using the current inner product.

$$K_i = \left(I_m - \frac{\widehat{p}_i \widehat{p}_i^T \widehat{A} C}{\widehat{p}_i^T \widehat{A} C \widehat{p}_i} \right) K_{i-1} \left(I_m - \frac{\widehat{A} \widehat{p}_i \widehat{p}_i^T C}{\widehat{p}_i^T \widehat{A} C \widehat{p}_i} \right) + \frac{\widehat{p}_i \widehat{p}_i^T C}{\widehat{p}_i^T \widehat{A} C \widehat{p}_i}$$

  where $i = 1, ..., l$, $C = GBG^T$, $\widehat{A} = I_m + R^{-1} GBG^T$ and $\widehat{p}_i$ is the search direction.

  $\rightarrow$ It is **costly** for large-scale problems.

- **Define a criterion** on whether we precondition the system or not by using a measure on symmetry.

  $\rightarrow$ Sensitive to the threshold value.

The quasi-Newton LMP in dual space (Nonlinear case)

Solution:

- **Re-generate** the preconditioner using the current inner product.

$$K_i = \left(I_m - \frac{\widehat{p}_i \widehat{p}_i^T \widehat{A} C}{\widehat{p}_i^T \widehat{A} C \widehat{p}_i} \right) K_{i-1} \left(I_m - \frac{\widehat{A} \widehat{p}_i \widehat{p}_i^T C}{\widehat{p}_i^T \widehat{A} C \widehat{p}_i} \right) + \frac{\widehat{p}_i \widehat{p}_i^T C}{\widehat{p}_i^T \widehat{A} C \widehat{p}_i}$$

  where $i = 1, ..., l$, $C = GBG^T$, $\widehat{A} = I_m + R^{-1} GBG^T$ and $\widehat{p}_i$ is the search direction.

  $\rightarrow$ It is **costly** for large-scale problems.

- **Define a criterion** on whether we precondition the system or not by using a measure on symmetry.

  $\rightarrow$ Sensitive to the threshold value.

Objective:

- A robust algorithm that handles this sensitivity.
- A globally convergent algorithm.

The Gauss-Newton method with a trust-region strategy

- Global convergence can be ensured by inserting the Gauss-Newton strategy in a trust region framework.

- Trust-region method simply solves the following problem at iteration k:

$$\min_{\delta x_k \in \mathbb{R}^n} J(\delta x_k) = \frac{1}{2}\|\delta x_k - x_b + x_k\|^2_{B^{-1}} + \frac{1}{2}\|G_k \delta x_k - d_k\|^2_{R^{-1}}$$

subject to $\|\delta x_k\|_{F_k^{-1}} \leq \Delta_k$ (primal approach)

where Δ_k is the trust region radius.

Trust-region in dual space

- The preconditioner $\tilde{K}_{k-1}$ that is inherited from previous iteration may **not be symmetric in the current inner product** and may **not be positive-definite in the full dual space** (merely in one direction).

Trust-region in dual space

- The preconditioner $\tilde{K}_{k-1}$ that is inherited from previous iteration may **not be symmetric in the current inner product** and may **not be positive-definite in the full dual space** (merely in one direction).

- We need to adapt the strategy in the trust region algorithm.

Trust-region in dual space

- The preconditioner $\tilde{K}_{k-1}$ that is inherited from previous iteration may **not be symmetric in the current inner product** and may **not be positive-definite in the full dual space** (merely in one direction).

- We need to adapt the strategy in the trust region algorithm.

Step calculation with a **flexible Steihaug-Toint PCG algorithm**

1. Check the positive-definiteness along the steepest descent direction.
2. Compute the Cauchy step by using the **Steihaug-Toint truncated PCG**
3. Compute the step beyond the Cauchy step with the**augmented RPCG** algorithm (ignoring symmetry problem)

Trust-region in dual space

- The preconditioner $\tilde{K}_{k-1}$ that is inherited from previous iteration may **not be symmetric in the current inner product** and may **not be positive-definite in the full dual space** (merely in one direction).
- We need to adapt the strategy in the trust region algorithm.

Step calculation with a **flexible Steihaug-Toint PCG algorithm**

1. Check the positive-definiteness along the steepest descent direction.
2. Compute the Cauchy step by using the **Steihaug-Toint truncated PCG**
3. Compute the step beyond the Cauchy step with the**augmented RPCG** algorithm (ignoring symmetry problem)

Trust-region in dual space

Flexible trust region algorithm

1. Initialization
2. Compute the step by the **flexible Steihaug-Toint PCG algorithm**
3. **Accept the step beyond the Cauchy step** if

$$f(y_k) < f(x_k^C)$$

4. Accept the trial point according to the ratio of achieved to predicted reduction
5. Update the trust region

→ The global convergence can be proved!

Trust-region in dual space

Flexible trust region algorithm

1. Initialization
2. Compute the step by the **flexible Steihaug-Toint PCG algorithm**
3. **Accept the step beyond the Cauchy step** if

$$f(y_k) < f(x_k^C)$$

4. Accept the trial point according to the ratio of achieved to predicted reduction
5. Update the trust region

→ The global convergence can be proved!

→ This approach is similar to the approach that computes the magical step proposed by (Conn, Gould, Toint 2000).

Numerical experiment on heat equation (1/3)

The dynamical model is considered to be the nonlinear heat equation defined by

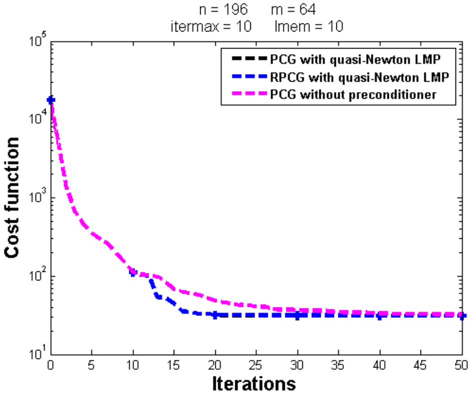
$$\frac{\delta x}{\delta t} - \frac{\delta^2 x}{\delta u^2} - \frac{\delta^2 x}{\delta v^2} + f[x] = 0 \text{ in } \Omega \times (0, \infty)$$
$$x[u, v, t] = 0 \text{ on } \delta\Omega \times (0, \infty)$$

where the temperature variable $x[u, v, t]$ depend on both time t and position given by spatial coordinates u and v. The function $f[x]$ is defined by

$$f[x] = exp[\eta x]$$

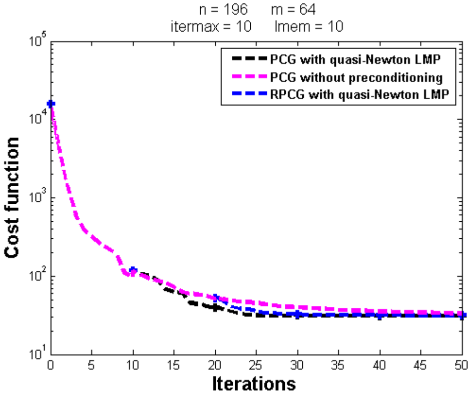
Numerical experiment on heat equation (2/3)

$$f[x] = exp[\eta x] \quad \eta = 2$$



Numerical experiment on heat equation (3/3)

$$f[x] = exp[\eta x] \quad \eta = 4.2$$



Conclusions

- Dual approach is interesting whenever $m << n$.
- It requires to introduce a special inner product and an augmentation.
- A robust and globally convergent algorithm is possible by using a flexible Steihaug-Toint trust-region algorithm in case of preconditioning in nonlinear case.

Conclusions

- Dual approach is interesting whenever $m << n$.
- It requires to introduce a special inner product and an augmentation.
- A robust and globally convergent algorithm is possible by using a flexible Steihaug-Toint trust-region algorithm in case of preconditioning in nonlinear case.
- We implemented RPCG algorithm into two operational data assimilation systems: ROMS (Regional Ocean Model System) and NEMOVAR (Variational data assimilation system for NEMO global ocean model)

Conclusions

- Dual approach is interesting whenever $m << n$.
- It requires to introduce a special inner product and an augmentation.
- A robust and globally convergent algorithm is possible by using a flexible Steihaug-Toint trust-region algorithm in case of preconditioning in nonlinear case.
- We implemented RPCG algorithm into two operational data assimilation systems: ROMS (Regional Ocean Model System) and NEMOVAR (Variational data assimilation system for NEMO global ocean model)

Future Work

- Implement the proposed algorithm including preconditioners into the ocean data assimilation systems NEMOVAR and ROMS.

Thank you for your attention !