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

S. Gratton\textsuperscript{1,2} S. Gürol\textsuperscript{2} Ph.L. Toint\textsuperscript{3}

\textsuperscript{1}ENSEEIHT, Toulouse, France
\textsuperscript{2}CERFACS, Toulouse, France
\textsuperscript{3}FUNDP, Namur, Belgium

7 September 2012
Outline

1. Introduction
2. Preconditioning in dual space
3. Globally convergent algorithm in dual space
4. Numerical Results
5. Conclusions
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 aircraft’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:

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

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:

\[
\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2} \|x - x_b\|_B^{-1} + \frac{1}{2} \sum_{j=0}^{N} \| H_j(M_{0,j}(x)) - y_j \|_{R_j^{-1}}^2
\]

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:

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

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

1. Solve the linearized subproblem at iteration \( k \)

\[
\min_{\delta x_k \in \mathbb{R}^n} J(\delta x_k) = \frac{1}{2} \| \delta x_k - x_b + x_k \|_B^{-1}^2 + \frac{1}{2} \| G_k \delta x_k - d_k \|_{R^{-1}}^2
\]
Solution algorithm

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

\[
\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2} ||x - x_b||^2_{B^{-1}} + \frac{1}{2} \sum_{j=0}^{N} \left( \frac{1}{2} ||H_j(M_{0,j}(x)) - y_j||^2_{R_j^{-1}} \right)
\]

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

1. Solve the **linearized subproblem** at iteration \(k\)

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

2. Perform update \(x_{k+1} = x_k + \delta x_k\)
Solution with primal approach

- From optimality condition

\[
\left( B^{-1} + G_k^T R^{-1} G_k \right) A_k \delta x_k = B^{-1} (x_b - x_k) + G_k^T R^{-1} d_k
\]

where the linear system is a **large-scale** system and \( A \) is **symmetric** and **positive-definite**.
Solution with primal approach

- From optimality condition

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

where the linear system is a large-scale system and \( A \) is symmetric and positive-definite.

- Exact solution can be written as

\[
x_b - x_k + \left( B^{-1} + G_k^T R^{-1} G_k \right)^{-1} G_k^T R^{-1} (d_k - G_k (x_b - x_k))
\]

requires solving a linear system **iteratively** in \( \mathbb{R}^n \)
Solution with primal approach

- From optimality condition

\[
\begin{align*}
\underbrace{(B^{-1} + G_k^T R^{-1} G_k)}_{A_k} \delta x_k &= B^{-1} (x_b - x_k) + G_k^T R^{-1} d_k \\
\end{align*}
\]

where the linear system is a large-scale system and \( A \) is symmetric and positive-definite.

- **Exact solution** can be written as

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

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

- From optimality condition

\[
\left( B^{-1} + G_k^T R^{-1} G_k \right) \delta x_k = B^{-1}(x_b - x_k) + G_k^T R^{-1} d_k
\]

where the linear system is a large-scale system and \( A \) is symmetric and positive-definite.

- Exact solution can be written as

\[
x_b - x_k + \left( B^{-1} + G_k^T R^{-1} G_k \right)^{-1} G_k^T R^{-1} (d_k - G_k(x_b - x_k))
\]

requires solving a linear system iteratively in \( \mathbb{R}^n \).

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

  → Preconditioning with the quasi-Newton Limited Memory Preconditioner (Morales and Nocedal 2000) (Gratton, Sartenaer and Tshimanga 2011)
Alternatively, the exact solution can be rewritten from duality theory or using Sherman-Morrison-Woodbury formula

\[
x_b - x_k + BG_k^T (G_k BG_k^T + R)^{-1} (d_k - G_k (x_b - x_k))
\]

requires solving a linear system iteratively in \( \mathbb{R}^m \).
Alternatively, the **exact solution** can be rewritten from duality theory or using Sherman-Morrison-Woodbury formula

\[
x_b - x_k + BG_k^T (G_k BG_k^T + R)^{-1} (d_k - G_k (x_b - x_k))
\]

requires solving a linear system **iteratively** 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

\[ x_b - x_k + B G_k^T \left( G_k B G_k^T + R \right)^{-1} (d_k - G_k (x_b - x_k)) \]

requires solving a linear system *iteratively* in \( \mathbb{R}^m \).

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

**Minimiation 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 \(\delta x_k = x_b - x_k + B \tilde{G}_k^T \lambda\)
Introduction

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**.
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.

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

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

$\rightarrow F = F_l.$
The quasi-Newton LMP: The descent directions $p_i, i = 1, \ldots, l$ generated by a CG method are used.

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

$\rightarrow F = F_l.$

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

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

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

$\rightarrow K = K_l.$
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

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

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

$\rightarrow K = K_l$.

This preconditioner satisfies the relation: $FG^T = BG^T K$ and it is symmetric in the $C$ inner product (Gratton, Gürol and Toint 2012).
For **nonlinear case**, inheriting the previous preconditioner may not be possible!

![Graph showing cost function vs iterations](image)
For **nonlinear case**, inheriting the previous preconditioner may not be possible!

Why?
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.

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

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

It is costly for large-scale problems. Define a criterion on whether we precondition the system or not by using a measure on symmetry. Sensitive to the threshold value. Objective: A robust algorithm that handles this sensitivity. A globally convergent algorithm.
The quasi-Newton LMP in dual space (Nonlinear case)

**Solution:**

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

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

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

→ 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.

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

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

→ It is **costly** for large-scale problems.

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

→ Sensitive to the threshold value.
Preconditioning in dual space

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

Solution:

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

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

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

→ It is **costly** for large-scale problems.

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

→ Sensitive to the threshold value.

Objective:

- A robust algorithm that handles this sensitivity.
- A globally convergent algorithm.
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 $k$:

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

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

where $\Delta_k$ is the trust region radius.
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).
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.
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)
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 Results

Numerical experiment on heat equation (1/3)

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

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

\[
x[u, v, t] = 0 \text{ on } \partial \Omega \times (0, \infty)
\]

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)

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

\[ n = 196 \quad m = 64 \quad \text{itermax} = 10 \quad \text{Imem} = 10 \]
Numerical experiment on heat equation (3/3)

\[ f(x) = \exp[\eta x] \quad \eta = 4.2 \]

\[ n = 196 \quad m = 64 \quad \text{itermax} = 10 \quad \text{Imem} = 10 \]
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 !