

Randomised preconditioning for the forcing formulation of weak constraint 4D-Var

Ieva Daužickaitė

University of Reading

Amos Lawless, Jennifer Scott, Peter Jan van Leeuwen

vEGU21

EPSRC

Engineering and Physical Sciences
Research Council



**National Centre for
Earth Observation**

NATURAL ENVIRONMENT RESEARCH COUNCIL

Mathematics
of Planet Earth



**Imperial College
London**



**University of
Reading**

Objective

Explore randomised eigenvalue decomposition to construct limited memory preconditioners for the forcing formulation of incremental weak constraint four-dimensional variational data assimilation.

We will look at:

- Algorithm for randomised eigenvalue decomposition.
- Numerical example with the Lorenz 96 model.

Forcing formulation of weak constraint 4D-Var

- Minimise nonlinear cost function

$$\begin{aligned} J(x_0, \eta_1, \dots, \eta_N) = & \frac{1}{2} (x_0 - x^b)^T B^{-1} (x_0 - x^b) \\ & + \frac{1}{2} \sum_{i=0}^N (y_i - \mathcal{H}_i(x_i))^T R_i^{-1} (y_i - \mathcal{H}_i(x_i)) \\ & + \frac{1}{2} \sum_{i=1}^N \eta_i^T Q_i^{-1} \eta_i. \end{aligned}$$

- Obtain analysis $x_0, x_1, \dots, x_N$, $x_i \in \mathbb{R}^n$ using as constraint:

$$x_{i+1} = \mathcal{M}_i(x_i) + \eta_{i+1}.$$

Incremental formulation

Approximate the analysis by incrementally updating

$$\mathbf{p}^{(j)} = \mathbf{p}^{(j-1)} + \delta \mathbf{p}^{(j)},$$

where $\delta \mathbf{p}^{(j)}$ minimises the quadratic cost function

$$J^\delta(\delta \mathbf{p}^{(j)}) = \frac{1}{2} \|\delta \mathbf{p}^{(j)} - \mathbf{b}^{(j)}\|_{\mathbf{D}^{-1}}^2 + \frac{1}{2} \|\mathbf{H}^{(j)}(\mathbf{L}^{-1})^{(j)} \delta \mathbf{p}^{(j)} - \mathbf{d}^{(j)}\|_{\mathbf{R}^{-1}}^2.$$

- $\mathbf{p}^{(j)} = (x_0^{(j)T}, \eta_1^{(j)T}, \dots, \eta_N^{(j)T})^T \in \mathbb{R}^{n(N+1)}$;
- $(\mathbf{L}^{-1})^{(j)}$ includes the tangent linear model and $\mathbf{H}^{(j)}$ is the linearised observation operator;
- $\mathbf{b}^{(j)}$ consists of initial condition and model errors;
- $\mathbf{d}^{(j)}$ consists of the discrepancy between the state and the observations;
- $\mathbf{D}$ and $\mathbf{R}$ includes error covariance matrices.

Solving linear systems

- The update $\delta \mathbf{p}$ can be obtained by solving

$$\hat{\mathbf{A}} \delta \mathbf{p} = \mathbf{D}^{-1} \mathbf{b} + \mathbf{L}^{-T} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{d}$$

$$\text{where } \hat{\mathbf{A}} = \mathbf{D}^{-1} + \mathbf{L}^{-T} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \mathbf{L}^{-1}.$$

with the conjugate gradient (CG) method.

- CG convergence depends on **eigenvalue distribution** of $\hat{\mathbf{A}} \in \mathbb{R}^{n(N+1) \times n(N+1)}$.

Solving linear systems

- The update $\delta \mathbf{p}$ can be obtained by solving

$$\hat{\mathbf{A}} \delta \mathbf{p} = \mathbf{D}^{-1} \mathbf{b} + \mathbf{L}^{-T} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{d}$$
$$\text{where } \hat{\mathbf{A}} = \mathbf{D}^{-1} + \mathbf{L}^{-T} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \mathbf{L}^{-1}.$$

with the conjugate gradient (CG) method.

- CG convergence depends on **eigenvalue distribution** of $\hat{\mathbf{A}} \in \mathbb{R}^{n(N+1) \times n(N+1)}$.
- Apply first level preconditioning (control variable transform):

$$\mathbf{A} = \mathbf{I} + \mathbf{D}^{1/2} \mathbf{L}^{-T} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \mathbf{L}^{-1} \mathbf{D}^{1/2}.$$

- **A** eigenvalues:
 - ▶ $n(N+1) - q$ eigenvalues are equal to one (q is the total number of observations);
 - ▶ q eigenvalues larger than one. **[May slow down the minimisation.]**

Limited memory preconditioners (LMPs)

Preconditioning: solve $\mathbf{P}_k \mathbf{A} \delta \mathbf{p} = \mathbf{P}_k \hat{\mathbf{b}}$.

Spectral LMP [Tshimanga et al., 2008] is constructed with eigenvalues λ_i and eigenvectors $\mathbf{v}_i$

$$\mathbf{P}_k = \mathbf{I} - \sum_{i=1}^k (1 - \lambda_i^{-1}) \mathbf{v}_i \mathbf{v}_i^T.$$

- $\mathbf{P}_k$ maps eigenvalues λ_i to one;
- For large systems, $\mathbf{P}_k$ is constructed with approximations of λ_i and $\mathbf{v}_i$;
- Estimates of λ_i and $\mathbf{v}_i$ can be obtained cheaply **after** solving $\mathbf{A} \delta \mathbf{p} = \hat{\mathbf{b}}$;
- Estimates of λ_i and $\mathbf{v}_i$ of $\mathbf{A}$ in one inner loop are used to precondition the consecutive inner loop. **[This will only be useful when $\mathbf{A}$ does not change significantly in these inner loops.]**

Randomised methods for low rank approximations

- Give accurate approximation with high probability when the eigenvalues/singular values decrease rapidly [Halko et al., 2011];
- Easy to implement;
- Block methods (easy to parallelise);
- Can be used to develop solvers in data assimilation [Bousserez et al., 2020].

Randomised eigenvalue decomposition (REVD):

- Estimates of λ_i and v_i of $\mathbf{A}$ can be used to **precondition the same inner loop**.

Randomised eigenvalue decomposition (REVD)

Input: symmetric matrix $\mathbf{A} \in \mathbb{R}^{n_A \times n_A}$, target rank k , an oversampling parameter r

Output: orthogonal $\mathbf{U} \in \mathbb{R}^{n_A \times k}$ with approximations to eigenvectors of $\mathbf{A}$ as its columns, and diagonal $\mathbf{\Theta} \in \mathbb{R}^{k \times k}$ with approximations to the largest eigenvalues of $\mathbf{A}$ on the diagonal

Randomised eigenvalue decomposition (REVD)

Input: symmetric matrix $\mathbf{A} \in \mathbb{R}^{n_A \times n_A}$, target rank k , an oversampling parameter r

Output: orthogonal $\mathbf{U} \in \mathbb{R}^{n_A \times k}$ with approximations to eigenvectors of $\mathbf{A}$ as its columns, and diagonal $\mathbf{\Theta} \in \mathbb{R}^{k \times k}$ with approximations to the largest eigenvalues of $\mathbf{A}$ on the diagonal

- 1: Form a Gaussian random matrix $\mathbf{G} \in \mathbb{R}^{n_A \times (k+r)}$
- 2: Form a sample matrix $\mathbf{Y} = \mathbf{A}\mathbf{G}$
- 3: Orthonormalize the columns of $\mathbf{Y}$ to obtain orthonormal $\mathbf{Z}$

$$\begin{array}{l} \text{Compute } \mathbf{Z}: \\ \mathbf{A} \approx \mathbf{Z}\mathbf{Z}^T\mathbf{A}\mathbf{Z}\mathbf{Z}^T \end{array}$$

Randomised eigenvalue decomposition (REVD)

Input: symmetric matrix $\mathbf{A} \in \mathbb{R}^{n_A \times n_A}$, target rank k , an oversampling parameter r

Output: orthogonal $\mathbf{U} \in \mathbb{R}^{n_A \times k}$ with approximations to eigenvectors of $\mathbf{A}$ as its columns, and diagonal $\mathbf{\Theta} \in \mathbb{R}^{k \times k}$ with approximations to the largest eigenvalues of $\mathbf{A}$ on the diagonal

- 1: Form a Gaussian random matrix $\mathbf{G} \in \mathbb{R}^{n_A \times (k+r)}$
- 2: Form a sample matrix $\mathbf{Y} = \mathbf{A}\mathbf{G}$
- 3: Orthonormalize the columns of $\mathbf{Y}$ to obtain orthonormal $\mathbf{Z}$
- 4: Form $\mathbf{K} = \mathbf{Z}^T \mathbf{A} \mathbf{Z}$
- 5: Form EVD of $\mathbf{K}$: $\mathbf{K} = \mathbf{W} \mathbf{\Theta} \mathbf{W}^T$
- 6: Truncate $\mathbf{\Theta}$ and $\mathbf{W}$
- 7: Form $\mathbf{U} = \mathbf{Z} \mathbf{W}$.

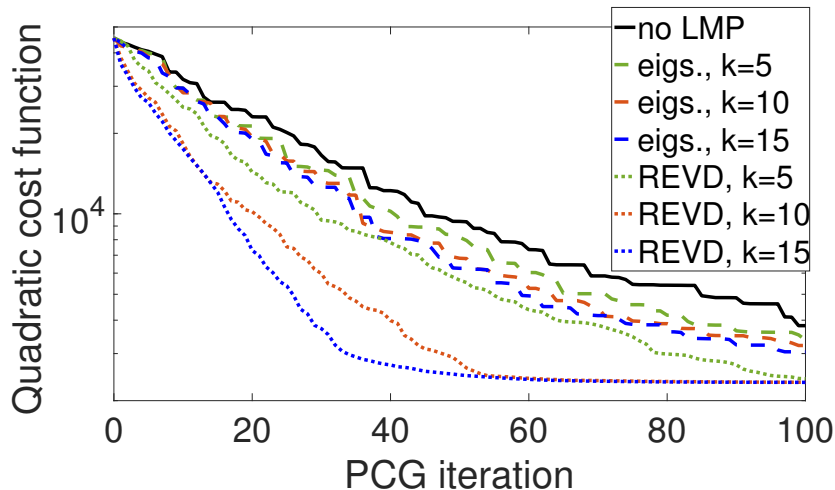
Compute $\mathbf{Z}$:
$$\mathbf{A} \approx \mathbf{Z} \mathbf{Z}^T \mathbf{A} \mathbf{Z} \mathbf{Z}^T$$

Use $\mathbf{Z}$ for standard EVD

Numerical example. Setup

- Identical twin experiments;
- Lorenz 96 model with $n = 80$ variables, grid point distance $\Delta X = 1/n$, $N = 150$, $\Delta t = 0.025$;
- $B_i = 0.2^2 C_b$, $Q_i = 0.1^2 C_q$, C_b and C_q are Laplacian and SOAR matrices, respectively, with length scales $2\Delta X$, $R_i = 0.15^2 I$;
- Direct observations, $q = 120$;
- First inner loop is solved without preconditioning;
- Preconditioning in the second inner loop:
 - ▶ use eigenvalue and eigenvector estimates computed with **eigs in the first loop (deterministic LMP)**;
 - ▶ run **REVD in the second loop** fifty times with different G and report the mean values of $J^\delta(\delta \mathbf{p}^{(2)})$ at every iteration (**randomised LMP**);
- Performed with Matlab (part of code written by A. El-Said).

Randomised vs. deterministic (eigs) LMPs



Summary

- We looked at second level preconditioning for the forcing formulation of the incremental weak constraint 4D-Var;
- Randomised methods for low rank approximations can be used to construct LMPs independently of the previous loops;
- Numerical example shows:
 - ▶ Randomised preconditioners can be more effective than currently used techniques;
 - ▶ Using more vectors to construct randomised LMPs results in faster minimisation (may not happen in the first CG iterations).

Further work

- Test other randomised methods (see [Daužickaitė et al., 2021]);
- Explore the sensitivities to the number of observations q (see [Daužickaitė et al., 2021]);
- Test randomised preconditioning for large and more realistic systems.

References

 Bousserez, N., Guerrette, J. J., and Henze, D. K. (2020).

Enhanced parallelization of the incremental 4D-Var data assimilation algorithm using the Randomized Incremental Optimal Technique.

Quarterly Journal of the Royal Meteorological Society, 146(728):1351 – 1371.

 Daužickaitė, I., Lawless, A. S., Scott, J. A., and van Leeuwen, P. J. (2021).

Randomised preconditioning for the forcing formulation of weak constraint 4D-Var.

Available on <https://arxiv.org/abs/2101.07249>.

 Halko, N., Martinsson, P., and Tropp, J. (2011).

Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions.

SIAM Review, 53(2):217 – 288.

 Tshimanga, J., Gratton, S., Weaver, A. T., and Sartenaer, A. (2008).

Limited-memory preconditioners, with application to incremental four-dimensional variational data assimilation.

Quarterly Journal of the Royal Meteorological Society, 134(632):751 – 769