On time-parallel preconditioning for the state formulation of incremental weak constraint 4D-Var

Received: 20 May 2021 | Revised: 23 July 2021 | Accepted: 26 July 2021

DOI: 10.1002/qj.4140

Quarterly Journal of the Royal Meteorological Society RMetS

RESEARCH ARTICLE

On time-parallel preconditioning for the state formulation of incremental weak constraint 4D-Var

Ieva Daužickaitė[1] | **Amos S. Lawless**[1,2] | **Jennifer A. Scott**[1,3] | **Peter Jan van Leeuwen**[1,4]

[1]School of Mathematical, Physical and Computational Sciences, University of Reading, Reading, UK

[2]National Centre for Earth Observation, Reading, UK

[3]Scientific Computing Department, STFC Rutherford Appleton Laboratory, Didcot, UK

[4]Department of Atmospheric Science, Colorado State University, Fort Collins, Colorado

Correspondence

I. Daužickaitė, Department of Mathematics and Statistics, Pepper Lane, Whiteknights, Reading RG6 6AX, UK.
Email: i.dauzickaite@pgr.reading.ac.uk

Funding information

UK Engineering and Physical Sciences Research Council: EP/L016613/1, European Research Council CUNDA project: 694509, NERC National Centre for Earth Observation

Abstract

Using a high degree of parallelism is essential for the efficient performance of data assimilation. The state formulation of the incremental weak constraint four-dimensional variational data assimilation method allows parallel calculations in the time dimension. In this approach, the solution is approximated by minimising a series of quadratic cost functions using the conjugate gradient method. To use this method in practice, effective preconditioning strategies that maintain the potential for parallel calculations are needed. We examine approximations to the control variable transform (CVT) technique when the latter is beneficial. The new strategy employs a randomised singular value decomposition and retains the potential for parallelism in the time domain. Numerical results for the Lorenz '96 model show that this approach accelerates the minimisation in the first few iterations, with better results when CVT performs well.

KEYWORDS

conjugate gradients, data assimilation, preconditioning, randomised methods, sparse symmetric positive definite systems, time-parallel 4D-Var, weak constraint 4D-Var

1 | INTRODUCTION

The ever increasing resolution of weather models enhances the importance of parallelisation in data assimilation. Higher potential for parallel computations can be achieved by using suitable data assimilation methods. The state formulation of the weak constraint 4D-Var method, which allows for the model error, is such a method. In its incremental version, a series of quadratic cost functions is minimised via solving a series of linear systems containing the Hessian of the linearised cost function. These are solved with the conjugate gradient (CG) method (e.g., Saad 2003), where the most computationally expensive part is integrating the tangent linear model and its adjoint. It has been shown that these calculations can be parallelised in the time dimension (Fisher and Gürol 2017).

However, CG needs preconditioning for fast convergence. Efficient preconditioning for the state formulation of incremental weak constraint 4D-Var, which also preserves the potential for parallel-in-time calculations, is still an open question. By analogy with the standard preconditioning technique (also known as a control variable transform or first-level preconditioning) used in strong constraint 4D-Var, Fisher and Gürol (2017) suggested using approximations of the tangent linear model. Their search for a suitable approximation was unsuccessful. Our investigation in this paper reveals that preconditioning using the exact tangent linear model can be detrimental to the minimisation in some cases. We focus on approximations in the case when using the exact tangent linear model works well.

In the light of the growing popularity of randomised methods and examples of their use in data assimilation (Bousserez *et al.*, 2020, Daužickaitė *et al.*, 2021), we propose using a randomised singular value decomposition (RSVD; Halko *et al.*, 2011) to approximate the tangent linear model. RSVD is a block method that is easy to parallelise in the sense that it requires calculating matrix products with blocks of vectors. Because Lawless *et al.* (2008) showed that it is important to take into account the information on the background errors when using model reduction techniques in data assimilation, we also examine an approach where we approximate the tangent linear model in interaction with the background- and model-error covariance matrices.

We formulate the incremental weak constraint 4D-Var problem and discuss its preconditioning in Section 2. Our idea for randomised preconditioning is presented in Section 3. Numerical experiments exploring preconditioning using the exact tangent linear model and its low-rank approximation obtained using RSVD are presented in Section 4 and we summarize our findings and suggest future directions in Section 5.

2 | INCREMENTAL WEAK CONSTRAINT 4D-VAR

In data assimilation, the prior estimate of a model trajectory is combined with observations over a time window to obtain an improved estimate of the state (analysis) $\mathbf{x}_0^a, \mathbf{x}_1^a, \dots, \mathbf{x}_N^a$ at times $t_0, t_1, \dots, t_N$. The prior estimate of the state at t_0 is called the background and is denoted by $\mathbf{x}^b \in \mathbb{R}^n$ and the observations at time t_i are denoted by $\mathbf{y}_i \in \mathbb{R}^{p_i}$. The state variables $\mathbf{x}_i$ are mapped to the observation space using an observation operator $\mathcal{H}_i$. The nonlinear dynamical model $\mathcal{M}_i$ describes the state evolution from time t_i to t_{i+1}. It is assumed that the background, observations and model have Gaussian errors with zero mean and covariance matrices $\mathbf{B} \in \mathbb{R}^{n \times n}$, $\mathbf{R}_i \in \mathbb{R}^{p_i \times p_i}$, and $\mathbf{Q}_i \in \mathbb{R}^{n \times n}$, respectively. We assume that the observation and model errors are uncorrelated in time.

In the state formulation of weak constraint 4D-Var, the analysis is the minimiser of the nonlinear cost function

$$J(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_N) = \frac{1}{2}\|\mathbf{x}_0 - \mathbf{x}^b\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\sum_{i=0}^{N}\|\mathbf{y}_i - \mathcal{H}_i(\mathbf{x}_i)\|^2_{\mathbf{R}_i^{-1}} + \frac{1}{2}\sum_{i=0}^{N-1}\|\mathbf{x}_{i+1} - \mathcal{M}_i(\mathbf{x}_i)\|^2_{\mathbf{Q}_{i+1}^{-1}}, \tag{1}$$

where $\|\mathbf{a}\|^2_{\mathbf{A}} = \mathbf{a}^{\mathrm{T}}\mathbf{A}\mathbf{a}$ (Trémolet (2006)).

The minimiser of (1) can be approximated using an inexact Gauss–Newton algorithm (Gratton *et al.*, 2007). In this incremental approach, the $(j+1)$th approximation $\mathbf{x}^{(j+1)} = \left(\mathbf{x}_0^{(j+1)\mathrm{T}}, \mathbf{x}_1^{(j+1)\mathrm{T}}, \dots, \mathbf{x}_N^{(j+1)\mathrm{T}}\right)^T \in \mathbb{R}^{(N+1)n}$ of the state is

$$\mathbf{x}^{(j+1)} = \mathbf{x}^{(j)} + \delta\mathbf{x}^{(j)}, \tag{2}$$

where the update is $\delta\mathbf{x}^{(j)} = \left(\delta\mathbf{x}_0^{(j)\mathrm{T}}, \delta\mathbf{x}_1^{(j)\mathrm{T}}, \dots, \delta\mathbf{x}_N^{(j)\mathrm{T}}\right)^T \in \mathbb{R}^{(N+1)n}$. $\mathbf{M}_i$ and $\mathbf{H}_i$ are the model and observation operators linearised at $\mathbf{x}_i$; they are known as the tangent linear model and tangent linear observation operator, respectively. We define the following matrices (following Gratton *et al.*, 2018a)

$$\mathbf{L}^{(j)} = \begin{pmatrix} \mathbf{I} & & & & \\ -\mathbf{M}_0^{(j)} & \mathbf{I} & & & \\ & -\mathbf{M}_1^{(j)} & \mathbf{I} & & \\ & & \ddots & \ddots & \\ & & & -\mathbf{M}_{N-1}^{(j)} & \mathbf{I} \end{pmatrix} \in \mathbb{R}^{(N+1)n \times (N+1)n}, \tag{3}$$

$$\mathbf{H}^{(j)} = \mathrm{diag}(\mathbf{H}_0^{(j)}, \mathbf{H}_1^{(j)}, \dots, \mathbf{H}_N^{(j)}) \in \mathbb{R}^{p \times (N+1)n}, \tag{4}$$

$$\mathbf{D} = \mathrm{diag}(\mathbf{B}, \mathbf{Q}_1, \dots, \mathbf{Q}_N) \in \mathbb{R}^{(N+1)n \times (N+1)n}, \tag{5}$$

$$\mathbf{R} = \mathrm{diag}(\mathbf{R}_0, \mathbf{R}_1, \dots, \mathbf{R}_N) \in \mathbb{R}^{p \times p}, \tag{6}$$

where $\mathbf{I} \in \mathbb{R}^{n \times n}$ is the identity matrix, diag$(\cdot)$ denotes a block diagonal matrix and $p = \Sigma_{i=0}^{N} p_i$ is the total number of observations. We use the following notation for vectors

$$\mathbf{b}^{(j)} = \begin{pmatrix} \mathbf{x}_0^{(j)} - \mathbf{x}^b \\ \mathcal{M}_0(\mathbf{x}_0^{(j)}) - \mathbf{x}_1^{(j)} \\ \vdots \\ \mathcal{M}_{N-1}(\mathbf{x}_{N-1}^{(j)}) - \mathbf{x}_N^{(j)} \end{pmatrix} \in \mathbb{R}^{(N+1)n}, \tag{7}$$

$$\mathbf{d}^{(j)} = \begin{pmatrix} \mathbf{y}_0 - \mathcal{H}_0(\mathbf{x}_0^{(j)}) \\ \mathbf{y}_1 - \mathcal{H}_1(\mathbf{x}_1^{(j)}) \\ \vdots \\ \mathbf{y}_N - \mathcal{H}_N(\mathbf{x}_N^{(j)}) \end{pmatrix} \in \mathbb{R}^{p}. \tag{8}$$

The update $\delta\mathbf{x}^{(j)}$ is the minimiser of

$$J^{\delta}(\delta\mathbf{x}^{(j)}) = \frac{1}{2}||\mathbf{L}^{(j)}\delta\mathbf{x}^{(j)} - \mathbf{b}^{(j)}||^{2}_{\mathbf{D}^{-1}} + \frac{1}{2}||\mathbf{H}^{(j)}\delta\mathbf{x}^{(j)} - \mathbf{d}^{(j)}||^{2}_{\mathbf{R}^{-1}}. \tag{9}$$

Because (9) is a quadratic cost function, $\delta\mathbf{x}^{(j)}$ can be found by solving the following large linear systems with the Hessian $\mathbf{A}^{(j)}$ of $J^{\delta}(\delta\mathbf{x}^{(j)})$:

$$\mathbf{A}^{(j)}\delta\mathbf{x}^{(j)} = (\mathbf{L}^{\mathrm{T}})^{(j)}\mathbf{D}^{-1}\mathbf{b}^{(j)} + (\mathbf{H}^{\mathrm{T}})^{(j)}\mathbf{R}^{-1}\mathbf{d}^{(j)}, \tag{10}$$

$$\text{where } \mathbf{A}^{(j)} = (\mathbf{L}^{\mathrm{T}})^{(j)}\mathbf{D}^{-1}\mathbf{L}^{(j)} + (\mathbf{H}^{\mathrm{T}})^{(j)}\mathbf{R}^{-1}\mathbf{H}^{(j)}. \tag{11}$$

It is assumed that $p \ll (N+1)n$, thus $(\mathbf{H}^{\mathrm{T}})^{(j)}\mathbf{R}^{-1}\mathbf{H}^{(j)}$ is symmetric positive semi-definite. Because $(\mathbf{L}^{\mathrm{T}})^{(j)}\mathbf{D}^{-1}\mathbf{L}^{(j)}$ is symmetric positive definite, $\mathbf{A}^{(j)} \in \mathbb{R}^{(N+1)n \times (N+1)n}$ is symmetric positive definite. Hence the method of choice for solving (10) is CG. Each iteration of CG requires one matrix–vector product with $\mathbf{A}^{(j)}$, which is expensive due to the tangent linear model and its adjoint in $\mathbf{L}^{(j)}$ and $(\mathbf{L}^{\mathrm{T}})^{(j)}$, respectively. Fisher and Gürol (2017) noted that the structure of $\mathbf{L}^{(j)}$ allows the matrix–vector products with $\mathbf{A}^{(j)}$ to be parallelised in the time dimension, that is, computation of $\mathbf{L}^{(j)}\mathbf{z}$, where $\mathbf{z} \in \mathbb{R}^{(N+1)n}$, can be parallelised for the model linearised at different times. In the rest of this paper, the superscript (j) is omitted for ease of notation.

In general, CG needs preconditioning for fast convergence. Efficient preconditioning maps the system to another system that can be solved faster and the solution of the original problem can be easily recovered from the solution of the preconditioned problem. Choosing a suitable preconditioner is highly problem-dependent. Given the possibility of parallel computations in matrix–vector products with (11), the preconditioner should keep this potential.

2.1 | Preconditioning

We consider an extension of the control variable transform (also known as first-level preconditioning), which is used in 3D-Var where the model evolution is omitted, and in the strong constraint formulation of 4D-Var where the model is assumed to have no error (e.g., Lorenc *et al.*, 2000, Rawlins *et al.*, 2007, Lawless 2013). The idea is to apply the preconditioner so that the first term of the preconditioned Hessian is equal to identity. Then the preconditioned Hessian is a sum of the identity matrix and a low-rank symmetric positive semi-definite matrix with rank at most p. Its smallest eigenvalue is equal to 1 and it has at most p eigenvalues that are larger than 1. The latter can impair CG convergence if they are not well separated (a more general discussion appears in, e.g., Nocedal and Wright 2006 and Liesen and Strakoš 2012).

Applying this kind of preconditioning to the state formulation of weak constraint 4D-Var requires preconditioning with $\mathbf{L}^{-1}\mathbf{D}^{1/2}$, where

$$\mathbf{L}^{-1} = \begin{pmatrix} \mathbf{I} & & & & \\ \mathbf{M}_{0,0} & \mathbf{I} & & & \\ \mathbf{M}_{0,1} & \mathbf{M}_{1,1} & \mathbf{I} & & \\ \vdots & \vdots & \ddots & \ddots & \\ \mathbf{M}_{0,N-1} & \mathbf{M}_{1,N-1} & \dots & \mathbf{M}_{N-1,N-1} & \mathbf{I} \end{pmatrix} \tag{12}$$

and $\mathbf{M}_{i,j} = \mathbf{M}_j \dots \mathbf{M}_i$ denotes the linearised model integration from time t_i to t_{j+1}. Matrix–vector products with $\mathbf{L}^{-1}$ are sequential in the time dimension, that is,

$$\begin{aligned}\mathbf{L}^{-1}\mathbf{z} = [&\mathbf{z}_0^{\mathrm{T}}, (\mathbf{M}_0\mathbf{z}_0 + \mathbf{z}_1)^{\mathrm{T}}, \{\mathbf{M}_1(\mathbf{M}_0\mathbf{z}_0 + \mathbf{z}_1) + \mathbf{z}_2\}^{T}, \dots,\\ &\{\mathbf{M}_{N-1}(\mathbf{M}_{N-2} \dots \mathbf{M}_0\mathbf{z}_0 + \mathbf{M}_{N-2} \dots \mathbf{M}_1\mathbf{z}_1\\ &+ \dots + \mathbf{z}_{N-1}) + \mathbf{z}_N\}^{T}]^{\mathrm{T}},\end{aligned}$$

where $\mathbf{z} = (\mathbf{z}_0^{\mathrm{T}}, \mathbf{z}_1^{\mathrm{T}}, \dots, \mathbf{z}_N^{\mathrm{T}})^{\mathrm{T}}$. Fisher and Gürol (2017) suggested using an approximation $\widetilde{\mathbf{L}}^{-1}$ of $\mathbf{L}^{-1}$ in the preconditioner. Then the preconditioned system to be solved is

$$\mathbf{A}^{pr}\delta\widetilde{\mathbf{x}} = \mathbf{D}^{1/2}\widetilde{\mathbf{L}}^{-\mathrm{T}}(\mathbf{L}^{\mathrm{T}}\mathbf{D}^{-1}\mathbf{b} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{d}), \tag{13}$$

$$\text{where } \mathbf{A}^{pr} = \mathbf{D}^{1/2}\widetilde{\mathbf{L}}^{-\mathrm{T}}(\mathbf{L}^{\mathrm{T}}\mathbf{D}^{-1}\mathbf{L} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H})\widetilde{\mathbf{L}}^{-1}\mathbf{D}^{1/2}, \tag{14}$$

$$\widetilde{\mathbf{L}}^{-1}\mathbf{D}^{1/2}\delta\widetilde{\mathbf{x}} = \delta\mathbf{x}. \tag{15}$$

With an appropriate choice of $\widetilde{\mathbf{L}}^{-1}$, $\mathbf{A}^{pr}$ is symmetric positive definite. $\widetilde{\mathbf{L}}^{-1}$ should be chosen so that it can be applied in parallel. Fisher and Gürol could not find a suitable approximation that would guarantee good convergence. Gratton *et al.*, (2018a,2018b) discussed using $\widetilde{\mathbf{L}}^{-1}$ where $\mathbf{M}_i$ is set to zero or to the identity matrix in (12), which may be useful if the model state does not change significantly from one time step to the next. This may be unrealistic. In the next section we propose a new approximation strategy that avoids this assumption.

3 | RANDOMISED PRECONDITIONING

Randomised methods for low-rank matrix approximations have attracted a lot of interest in recent years because they require matrix products with blocks of vectors that can be easily parallelised, and it has been shown that good approximations for matrices with rapidly decaying singular values can be obtained with high probability (e.g., Halko *et al.*, 2011, Martinsson and Tropp 2020). These methods have been explored in data assimilation when designing solvers for strong constraint 4D-Var (Bousserez *et al.*, 2020) and preconditioning for the forcing formulation of the incremental weak constraint 4D-Var (Daužickaitė *et al.*, 2021).

A low-rank approximation of $\mathbf{L}^{-1}$ cannot be used in (13), because it would make (14) low rank and thus singular. Hence, we suggest exploiting the structure of $\mathbf{L}^{-1}$ when generating the preconditioner. We write

$$\mathbf{L}^{-1} = \mathbf{I} + \mathbf{P}, \tag{16}$$

where $\mathbf{P}$ is a strictly lower triangular matrix (with 0 on the diagonal). We propose using a rank k approximation $\widetilde{\mathbf{P}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$, where k is small compared to $(N+1)n$ and $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{(N+1)n \times k}$, $\mathbf{\Sigma} \in \mathbb{R}^{k \times k}$ is a truncated singular value decomposition, that is, $\mathbf{\Sigma}$ is diagonal with approximations to the k largest singular values of $\mathbf{P}$ on the diagonal, and the columns of $\mathbf{U}$ and $\mathbf{V}$ are approximate left and right singular vectors, respectively. Then a non-singular $\widetilde{\mathbf{L}}^{-1}$ is

$$\widetilde{\mathbf{L}}^{-1} = \mathbf{I} + \widetilde{\mathbf{P}} = \mathbf{I} + \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}. \tag{17}$$

An RSVD can be used to obtain $\widetilde{\mathbf{P}}$. RSVD is essentially one iteration of a classic subspace iteration method (Gu 2015). To increase the probability of success, the randomised methods work with larger subspaces than the required rank of the approximation. This is called oversampling. Halko *et al.* (2011) indicate that setting the oversampling parameter l to 5 or 10 generally gives good results. We present the RSVD in Algorithm 1. The entries of the Gaussian random matrix are independent standard normal random variables. Note that we remove the smallest l computed singular values and the corresponding singular vectors. In this way the oversampling increases the cost of generating the preconditioner (in particular, the cost of the matrix–matrix products in steps 2 and 4), but not of its application. RSVD needs two expensive matrix–matrix products with $\mathbf{P}$ (steps 2 and 4 in Algorithm 1), where $\mathbf{P}$ is multiplied with a matrix of size $(N+1)n \times (k+l)$. For efficiency, these can be parallelised. These matrix–matrix products consist of products with $\mathbf{M}_{i,j}$. Hence, the cost of generating the preconditioner depends on the cost of integrating the tangent linear model over the assimilation window sequentially.

Algorithm 1. Randomised singular value decomposition (RSVD)

```
Input: matrix $\mathbf{A} \in \mathbb{R}^{s \times s}$, target rank $k$, an oversampling parameter $l$
Output: orthogonal $\mathbf{U} \in \mathbb{R}^{s \times k}$ and $\mathbf{V} \in \mathbb{R}^{s \times k}$ whose columns are approximations to left and right singular vectors of $\mathbf{A}$, respectively, and diagonal $\mathbf{\Sigma} \in \mathbb{R}^{k \times k}$ with approximations to the largest singular values of $\mathbf{A}$.
1: Form a Gaussian random matrix $\mathbf{G} \in \mathbb{R}^{s \times (k+l)}$.
2: Form a sample matrix $\mathbf{Y} = \mathbf{A}\mathbf{G} \in \mathbb{R}^{s \times (k+l)}$.
3: Orthonormalize the columns of $\mathbf{Y}$ to obtain orthonormal $\mathbf{Z} \in \mathbb{R}^{s \times (k+l)}$.
4: Form $\mathbf{K} = \mathbf{Z}^{\mathrm{T}}\mathbf{A} \in \mathbb{R}^{(k+l) \times s}$.
5: Form SVD of $\mathbf{K} : \mathbf{K} = \hat{\mathbf{U}}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$, where $\hat{\mathbf{U}}, \mathbf{\Sigma} \in \mathbb{R}^{(k+l) \times (k+l)}, \mathbf{V} \in \mathbb{R}^{s \times (k+l)}$.
6: Remove last $l$ columns and rows of $\mathbf{\Sigma}$, so that $\mathbf{\Sigma} \in \mathbb{R}^{k \times k}$.
7: Remove last $l$ columns of $\hat{\mathbf{U}}$ and $\mathbf{V}$, so that $\hat{\mathbf{U}} \in \mathbb{R}^{(k+l) \times k}, \mathbf{V} \in \mathbb{R}^{s \times k}$.
8: Form $\mathbf{U} = \mathbf{Z}\hat{\mathbf{U}} \in \mathbb{R}^{s \times k}$.
```

Lawless *et al.* (2008) showed that including the background-error covariance matrix when using model reduction methods may lead to better results. Hence, we also explore using an approximation of

$$\mathbf{S} = \mathbf{L}^{-1}\mathbf{D}^{1/2} = \begin{pmatrix} \mathbf{B}^{1/2} & & & & \\ \mathbf{M}_{0,0}\mathbf{B}^{1/2} & \mathbf{Q}_1^{1/2} & & & \\ \mathbf{M}_{0,1}\mathbf{B}^{1/2} & \mathbf{M}_{1,1}\mathbf{Q}_1^{1/2} & \mathbf{Q}_2^{1/2} & & \\ \vdots & \vdots & \ddots & \ddots & \\ \mathbf{M}_{0,N-1}\mathbf{B}^{1/2} & \mathbf{M}_{1,N-1}\mathbf{Q}_1^{1/2} & \dots & \mathbf{M}_{N-1,N-1}\mathbf{Q}_{N-1}^{1/2} & \mathbf{Q}_N^{1/2} \end{pmatrix}. \tag{18}$$

As when approximating $\mathbf{L}^{-1}$, we write

$$\mathbf{S} = \mathbf{D}^{1/2} + \mathbf{W}, \tag{19}$$

where $\mathbf{W}$ is strictly lower triangular. An approximation $\widetilde{\mathbf{S}} = \mathbf{D}^{1/2} + \widetilde{\mathbf{W}}$ can be obtained by using RSVD to generate a low-rank approximation of $\mathbf{W}$. The system to be solved is (13) with $\widetilde{\mathbf{L}}^{-1}\mathbf{D}^{1/2}$ replaced by $\widetilde{\mathbf{S}}$.

4 | NUMERICAL RESULTS

We test preconditioning using $\mathbf{L}^{-1}$ and the approximations $\widetilde{\mathbf{L}}^{-1}$ and $\widetilde{\mathbf{S}}$ in (13) numerically. Preconditioning using the exact $\mathbf{L}^{-1}$ is considered so that we understand when preconditioning using $\widetilde{\mathbf{L}}^{-1}$ or $\widetilde{\mathbf{S}}$ may be effective, but this is not regarded as a practical approach when parallelisation in the time dimension is desired. Identical twin experiments are performed. The background state $\mathbf{x}^{\mathrm{b}}$ is generated by adding random, Gaussian noise with covariance $\mathbf{B}$ to $\mathbf{x}_0^{\mathrm{t}}$, where $\mathbf{x}_i^{\mathrm{t}}$ is the reference state at time t_i. We use direct

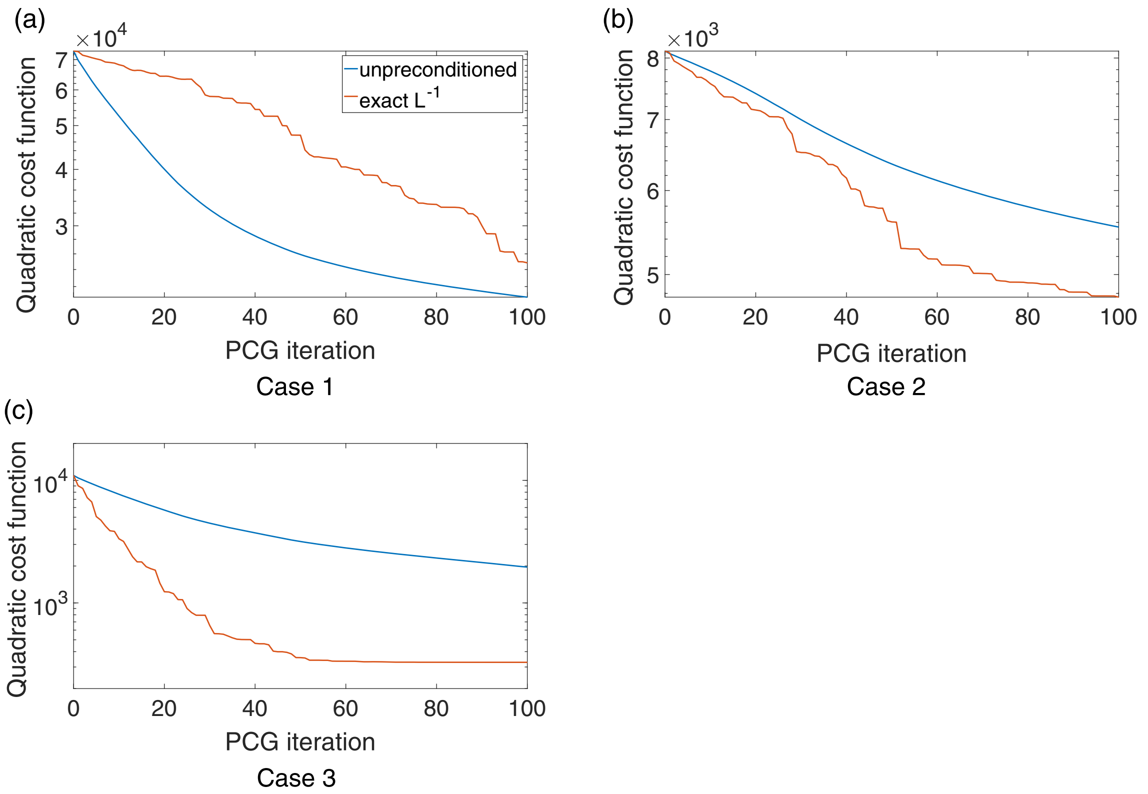

FIGURE 1 The values of the quadratic cost function at every PCG iteration when using no preconditioner and when preconditioning using exact $\mathbf{L}^{-1}$. Values of σ_o and the number of observations p for cases 1, 2 and 3 are given in the text [Colour figure can be viewed at wileyonlinelibrary.com]

observations that are obtained by adding random, Gaussian noise with covariance $\mathbf{R}_i$ to $\mathcal{H}_i(\mathbf{x}_i^t)$.

The nonlinear Lorenz '96 model (Lorenz 1996) is used, where the dynamics of $\mathbf{x}_i = (X^1, \dots, X^n)^{\mathrm{T}}$ are described by a set of n coupled ODEs:

$$\frac{\mathrm{d}X^j}{\mathrm{d}t} = -X^{j-2}X^{j-1} + X^{j-1}X^{j+1} - X^j + F, \qquad (20)$$

with conditions $X^{-1} = X^{n-1}, X^0 = X^n$ and $X^{n+1} = X^1$ and $F = 8$. We use a fourth-order Runge–Kutta scheme (Butcher 1987). We consider the system with $n = 100$ and $N = 149$, so $\mathbf{A}^{pr}$ is a 15,000 × 15,000 matrix. The time step is set to $\Delta t = 2.5 \times 10^{-2}$ and the grid point distance is $\Delta X = 1/n$.

The covariance matrices are $\mathbf{B} = 0.2^2\mathbf{C}_b$, $\mathbf{Q}_i = 0.05^2\mathbf{C}_q$, where $\mathbf{C}_b$ is a second-order autoregressive (SOAR; Daley 1993) matrix and $\mathbf{C}_q$ a Laplacian (Johnson *et al.*, 2005) correlation matrix with length-scales $2\Delta X$ and $0.75\Delta X$, respectively. We consider $\mathbf{R}_i = \sigma_o^2\mathbf{I}$ and vary σ_o.

The computations are performed with Matlab R2019b and the linear systems are solved with the Matlab preconditioned conjugate gradient (PCG) implementation *pcg*.

4.1 | Preconditioning with exact $\mathbf{L}^{-1}$

We have noticed that the effectiveness of the exact preconditioner $\mathbf{L}^{-1}$ depends on how much of the system is observed and the interaction between the model and observation errors. There are observations at every 10th time step, ensuring that there are observations at the final time. We consider the following cases regarding the observation error variance σ_o and the total number of observations p:

1. $\sigma_o = 1.5 \times 10^{-1}$, $p = 300$ (observing 2% of the system);
2. $\sigma_o = 4.5 \times 10^{-1}$, $p = 300$;
3. $\sigma_o = 1.5 \times 10^{-1}$, $p = 60$ (observing 0.4% of the system).

In Figure 1, we show that preconditioning using $\mathbf{L}^{-1}$ is not useful in case 1 but can be effective if the observation error variance is increased while keeping the same number of observations (case 2), or if the number of observations is reduced while σ_o is unchanged (case 3).

Note that we compare the value of the quadratic cost function at every PCG iteration without taking into account the cost of the computation, which can be evaluated in terms of runtime or energy consumption and depends on how much parallelism can be achieved (e.g.,

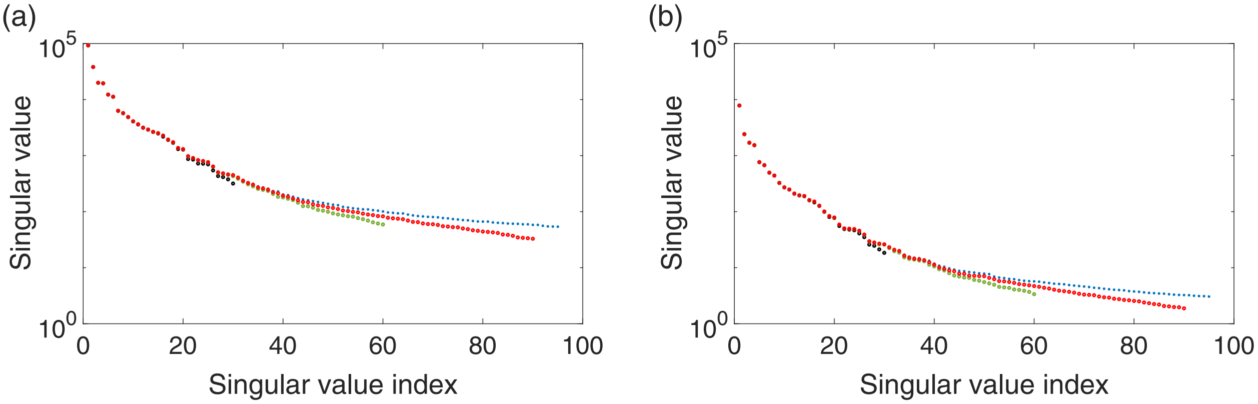

FIGURE 2 Largest singular values of (a) **P** (blue) and (b) **W** (blue) and their approximations given by RSVD when using rank $k = 30$ (black), $k = 60$ (green) and $k = 90$ (red). The largest singular values and their approximations coincide [Colour figure can be viewed at wileyonlinelibrary.com]

Carson and Strakoš 2020). If matrix–vector products with **L** can be parallelised, then PCG iterations when solving the unpreconditioned system can be performed faster than with preconditioning. Then, in terms of the runtime, preconditioning in case 1 can be even worse than indicated by comparing the quadratic cost function at every PCG iteration. In the same manner, preconditioning using exact $\mathbf{L}^{-1}$ in cases 2 and 3 may not be as effective as displayed. In the following section, we test preconditioning using $\widetilde{\mathbf{L}}^{-1}$ and $\widetilde{\mathbf{S}}$ in cases 2 and 3.

4.2 | Preconditioning with randomised low rank approximation

We generate $\widetilde{\mathbf{L}}^{-1}$ and $\widetilde{\mathbf{S}}$ by using rank $k \in \{30, 60, 90\}$ approximations of **P** and **W** in (16) and (19), respectively. The oversampling parameter is set to $l = 5$. We found that using $l = 10$ or $l = 15$ does not make a significant difference to the results (not shown). RSVD produces high-quality approximations of the singular values of both **P** and **W**. The largest singular values and their approximations are shown in Figure 2, where the same random seed is used to generate the random matrix **G** for all k values. The matrices **P** and **W** do not depend on whether case 2 or 3 is considered, because the cases differ in the observation terms. In each case, we run the RSVD algorithm one hundred times with different Gaussian matrices **G** and solve the systems with the resulting preconditioners. The spread is illustrated in Figure 3 for $\widetilde{\mathbf{S}}$. In both cases, the variation in the values of the cost function is small during the early iterations. This shows that our results are not very sensitive to the choice of **G** and, in practice, it is only necessary to run the RSVD algorithm once.

The means of the quadratic cost function in cases 2 and 3 are shown in Figure 4. Higher-rank approximations in both cases and using $\widetilde{\mathbf{S}}$ in case 3 result in faster minimisation. Notice that, in the first few iterations of PCG, preconditioning gives the same improvement regardless of the rank of approximation and whether $\widetilde{\mathbf{L}}^{-1}$ or $\widetilde{\mathbf{S}}$ is used. Preconditioning is more useful in case 3, which has fewer observations. The approximations used to generate $\widetilde{\mathbf{L}}^{-1}$ and $\widetilde{\mathbf{S}}$ are very low rank compared to the size of the system and there is a good improvement over the unpreconditioned case when the number of observations is low, especially in the beginning of the iterative process, which is the most relevant in practical settings. In the case with more observations (case 2), the randomised preconditioning is useful if a small number of PCG iterations is run. Since in an operational context we run only a small number of iterations, we are more likely to be in this regime. In cases 2 and 3, using exact $\mathbf{L}^{-1}$ results in a modest (case 2) and a rapid (case 3) decrease of the cost function in the first PCG iterations (Figure 1). Our proposed preconditioners replicate such behaviour and if larger k is used then the performance of exact $\mathbf{L}^{-1}$ is followed for more PCG iterations. In case 3, the quadratic cost function value is reduced by a factor of 2 after five PCG iterations when using exact $\mathbf{L}^{-1}$ in the preconditioner; the same result is obtained after eight ($k = 30$) and six ($k = 60$ and $k = 90$) PCG iterations using $\widetilde{\mathbf{L}}^{-1}$, and six ($k = 30$) and five ($k = 60$ and $k = 90$) PCG iterations using $\widetilde{\mathbf{S}}$. In case 2, the quadratic cost function is reduced only by a factor of 1.7 in 100 PCG iterations when preconditioning with the exact $\mathbf{L}^{-1}$. When using our preconditioners, the values of the quadratic cost function after 100 PCG iterations are larger than when using exact $\mathbf{L}^{-1}$ or no preconditioning. This can be addressed by using a larger rank approximation, computational resources permitting.

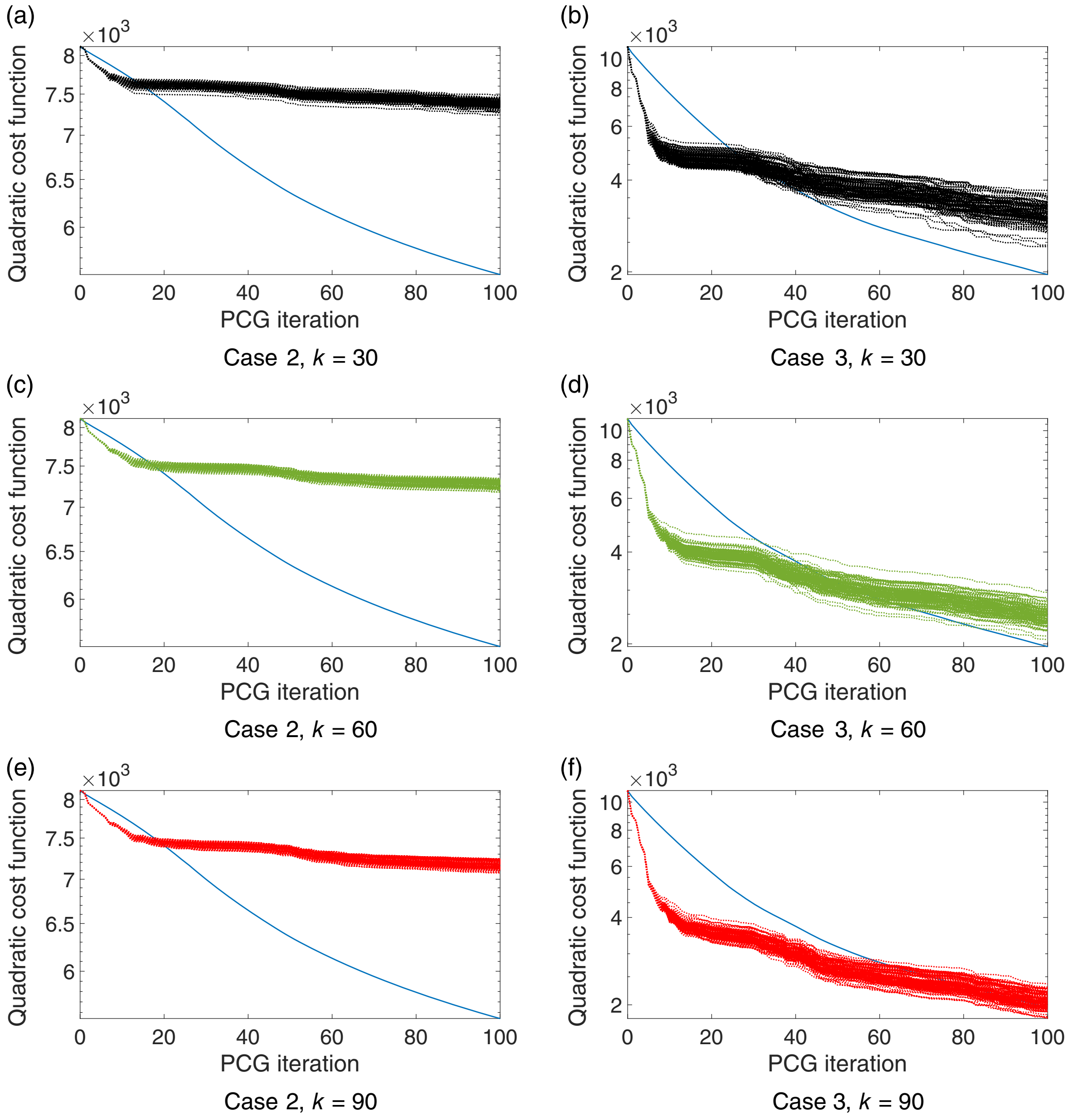

FIGURE 3 Values of the quadratic cost function at every PCG iteration when using no preconditioner (blue solid line) and preconditioning using $\widetilde{\mathbf{S}}$ (dotted lines) which are constructed using rank $k \in \{30, 60, 90\}$ approximation. One hundred realisations of the randomised preconditioner are shown. Values of σ_o and the number of observations p in cases 2 and 3 are given in the text [Colour figure can be viewed at wileyonlinelibrary.com]

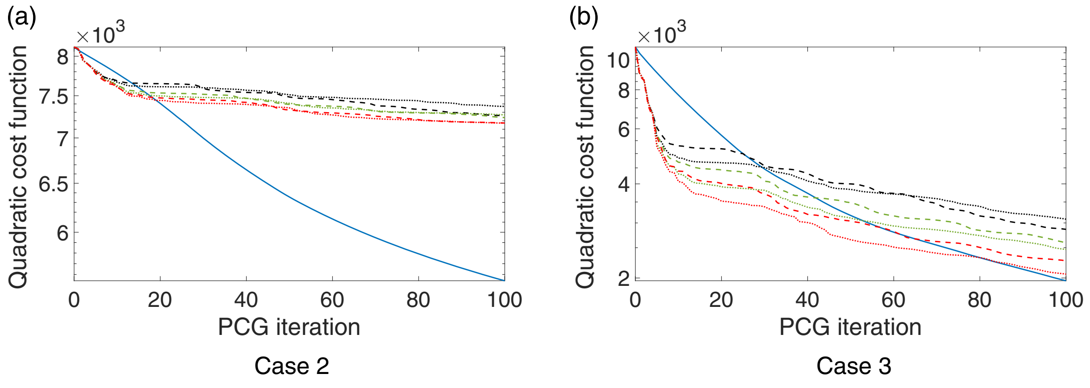

FIGURE 4 Mean values (over one hundred realisations) of the quadratic cost function at every PCG iteration when using no preconditioner (blue solid) and when preconditioning using $\widetilde{\mathbf{L}}^{-1}$ (dashed) and $\widetilde{\mathbf{S}}$ (dotted) that are constructed using rank $k = 30$ (black), $k = 60$ (green) and $k = 90$ (red) approximations. Values of σ_o and the number of observations p in cases 2 and 3 are given in the text [Colour figure can be viewed at wileyonlinelibrary.com]

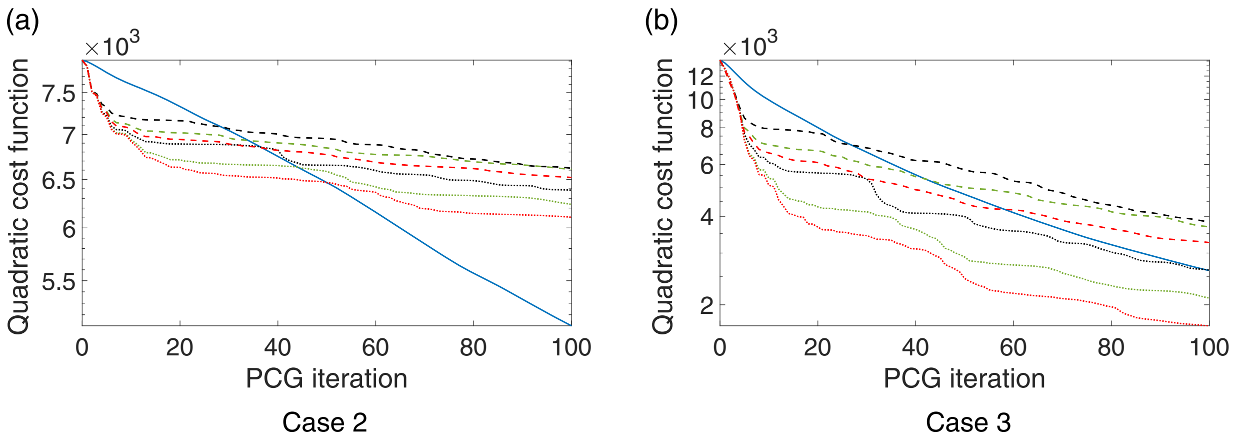

FIGURE 5 As Figure 4, but the model error covariance matrix is $\mathbf{Q}_i = 0.1^2\mathbf{C}_q$ and $\mathbf{C}_q$ has length-scale $2\Delta X$ [Colour figure can be viewed at wileyonlinelibrary.com]

4.2.1 | Large model error

We explore how the preconditioning using approximations of $\mathbf{L}^{-1}$ and $\mathbf{L}^{-1}\mathbf{D}^{1/2}$ compare when the model error is large. The numerical experiments are performed using the same set-up as before, but now we set $\mathbf{Q}_i = 0.1^2\mathbf{C}_q$ and $\mathbf{C}_q$ has length-scale $2\Delta X$. The means over one hundred runs are presented in Figure 5. There is a clear separation between the minimisation using $\widetilde{\mathbf{L}}^{-1}$ and $\widetilde{\mathbf{S}}$ in the preconditioner after the first few PCG iterations, with the latter resulting in faster minimisation. Notice that the preconditioning using both approximations remains useful for more PCG iterations than in the set-up with a smaller model error. This can be expected because the increase of length-scales of $\mathbf{Q}_i$ has a detrimental effect on the conditioning of the unpreconditioned Hessian (e.g., Chapter 6 of El-Said 2015) and hence preconditioning can be more efficient.

5 | CONCLUSIONS

We have considered preconditioning for the state formulation of incremental weak constraint 4D-Var, which closely follows the control variable transform (first-level preconditioning) strategy for the strong constraint formulation. We have shown that such preconditioning may not be useful even when using the exact $\mathbf{L}^{-1}$, which also makes the matrix–vector products with the Hessian sequential in the time dimension. In the cases where such preconditioning is useful, a good preconditioner can be obtained by using randomised singular value decompositions to approximate $\mathbf{L}^{-1}$ or $\mathbf{L}^{-1}\mathbf{D}^{1/2}$. These preconditioners are cheap to compute and apply and do allow for parallelization in the time dimension. They can improve the solution of the exact inner loop problem, resulting in a greater reduction of the quadratic cost function in the same number of iterations compared to using no preconditioning or obtaining the same quadratic cost function value in fewer iterations. The effect of the accuracy of the inner loop solution on the analysis has been studied by, for example, Lawless and Nichols (2006).

Our results call for caution when designing preconditioning approaches that focus on approximating $\mathbf{L}^{-1}$, especially when the number of observations is high. In practical NWP settings, around 1% of the system is observed, hence approximating $\mathbf{L}^{-1}$ may be useful. Using randomised approximations of $\mathbf{L}^{-1}$ or $\mathbf{L}^{-1}\mathbf{D}^{1/2}$ should be tested using large and more realistic systems, where meaningful evaluations of the runtime and energy consumption can be obtained. A more detailed investigation on when preconditioning with $\mathbf{L}^{-1}$ gives good results would also be useful.

AUTHOR CONTRIBUTIONS

Ieva Daužickaitė: conceptualization, formal analysis, investigation, methodology, software, validation, visualization, writing-original draft, writing-review & editing. **Amos S. Lawless:** investigation; methodology; supervision; validation; writing – review and editing. **Jennifer A. Scott:** funding acquisition; investigation; methodology; project administration; resources; supervision; validation; writing – review and editing. **Peter Jan van Leeuwen:** investigation; methodology; supervision; validation; writing – review and editing.

ACKNOWLEDGEMENTS

We thank Dr. Adam El-Said for his code for the weak constraint 4D-Var assimilation system. We are also grateful for the helpful comments by two anonymous reviewers. The work was supported by the UK Engineering and Physical Sciences Research Council grant no. EP/L016613/1, the European Research Council CUNDA project grant no. 694509 and the NERC National Centre for Earth Observation.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

ORCID

Ieva Daužickaitė https://orcid.org/0000-0002-1285-1764
Amos S. Lawless https://orcid.org/0000-0002-3016-6568
Jennifer A. Scott https://orcid.org/0000-0003-2130-1091

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, 1351–1371.

Butcher, J.C. (1987) *The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods*. Chichester, UK: Wiley.

Carson, E. and Strakoš, Z. (2020) On the cost of iterative computations. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 378. https://doi.org/10.1098/rsta.2019.0050.

Daley, R. (1993) *Atmospheric Data Analysis*. 2, Cambridge, UK: Cambridge University Press.

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. https://arxiv.org/abs/2101.07249.

El-Said, A. (2015). Conditioning of the weak-constraint variational data assimilation problem for numerical weather prediction. PhD thesis, Department of Mathematics and Statistics, University of Reading, UK.

Fisher, M. and Gürol, S. (2017) Parallelisation in the time dimension of four-dimensional variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 143, 1136–1147.

Gratton, S., Lawless, A.S. and Nichols, N.K. (2007) Approximate Gauss–Newton methods for nonlinear least squares problems. *SIAM Journal on Optimization*, 18(1), 106–132.

Gratton, S., Gürol, S., Simon, E. and Toint, P.L. (2018a) A note on preconditioning weighted linear least-squares, with consequences for weakly constrained variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 144, 934–940.

Gratton, S., Gürol, S., Simon, E. and Toint, P.L. (2018b) Guaranteeing the convergence of the saddle formulation for weakly constrained 4D-Var data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 144, 2592–2602.

Gu, M. (2015) Subspace iteration randomization and singular value problems. *SIAM Journal on Scientific Computing*, 37(3), A1139–A1173.

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.

Johnson, C., Hoskins, B.J. and Nichols, N.K. (2005) A singular vector perspective of 4D-Var: filtering and interpolation. *Quarterly Journal of the Royal Meteorological Society*, 131, 1–19.

Lawless, A.S. (2013). Variational data assimilation for very large environmental problems. In M.J.P. Cullen, M.A. Freitag, S. Kindermann, and R. Scheichl (Eds.), *Large Scale Inverse Problems: Computational Methods and Applications in the Earth Sciences*, pp. 55–90. Berlin: De Gruyter.

Lawless, A.S. and Nichols, N.K. (2006) Inner-loop stopping criteria for incremental four-dimensional variational data assimilation. *Monthly Weather Review*, 134(11), 3425–3435.

Lawless, A.S., Nichols, N.K., Boess, C. and Bunse-Gerstner, A. (2008) Using model reduction methods within incremental 4D-Var. *Monthly Weather Review*, 136, 1511–1522.

Liesen, J. and Strakoš, Z. (2012) *Krylov Subspace Methods: Principles and Analysis*. Numerical Mathematics and Scientific Computation, Oxford, UK: Oxford University Press.

Lorenc, A.C., Ballard, S.P., Bell, R.S., Ingleby, N.B., Andrews, P.L.F., Barker, D.M., Bray, J.R., Clayton, A.M., Dalby, T., Li, D., Payne, T.J. and Saunders, F.W. (2000) The Met Office global three-dimensional variational data assimilation scheme. *Quarterly Journal of the Royal Meteorological Society*, 126, 2991–3012.

Lorenz, E.N. (1996). Predictability – a problem partly solved, in Proceedings of Seminar on Predictability, 4–8 September 1995. Reading, UK: ECMWF.

Martinsson, P.G. and Tropp, J.A. (2020) Randomized numerical linear algebra: foundations and algorithms. *Acta Numerica*, 29, 403–572.

Nocedal, J. and Wright, S. (2006) *Numerical Optimization* (2nd ed.). New York, NY: Springer.

Rawlins, F., Ballard, S.P., Bovis, K.J., Clayton, A.M., Li, D., Inverarity, G.W., Lorenc, A.C. and Payne, T.J. (2007) The Met Office global four-dimensional variational data assimilation scheme. *Quarterly Journal of the Royal Meteorological Society*, 133, 347–362.

Saad, Y. (2003) *Iterative Methods for Sparse Linear Systems* (2nd ed.). Philadelphia, PA: SIAM.

Trémolet, Y. (2006) Accounting for an imperfect model in 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, 132, 2483–2504.

How to cite this article: Daužickaitė, I., Lawless, A.S., Scott, J.A. & van Leeuwen, P.J. (2021) On time-parallel preconditioning for the state formulation of incremental weak constraint 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, 1–9. Available from: https://doi.org/10.1002/qj.4140