Krylov space approximate Kalman filtering

Johnathan M. Bardsley[†], Marylesa Howard[†], and Albert Parker[*]

[†] *Department of Mathematical Sciences, University of Montana, Missoula, Montana, 59812, USA.*
[*] *Center for Biofilm Engineering, Montana State University, Bozeman, Montana, 59717, USA.*

SUMMARY

The Kalman filter is a technique for estimating a time-varying state given a dynamical model for, and indirect measurements of, the state. It is used, for example, on the control problems associated with a variety of navigation systems. Even in the case of nonlinear state and/or measurement models, standard implementations require only linear algebra. However, for sufficiently large-scale problems, such as arise in weather forecasting and oceanography, the matrix inversion and storage requirements of the Kalman filter are prohibitive, and hence, approximations must be made. In this paper, we describe how the conjugate gradient iteration can be used within the Kalman filter for quadratic minimization, as well as for obtaining low-rank, low-storage approximations of the covariance and inverse-covariance matrices required for its implementation. The approach requires that we exploit the connection between the conjugate gradient and Lanczos iterations.

KEY WORDS: Krylov subspace methods, Kalman filter, data assimilation, inverse problems.

1. Introduction

In 1960, the Kalman filter (KF) was introduced by R. E. Kalman [17], as a statistically optimal method for recursively estimating a time varying state, given a dynamical model, as well as indirect observations, of the state.

The filter has been used extensively in application areas such as autonomous and assisted navigation. We are interested here in its use on large-scale examples, such as numerical weather forecasting, where standard formulations of KF are computationally infeasible to implement.

Several variants of KF, and its nonlinear extension the extended Kalman filter (EKF) [42], propose to improve efficiency by projecting the state space onto a low dimensional subspace; see, for example, [6], [9], [14], [39]. This "reduced rank" approach is effective provided the state is well-represented on the subspace throughout the time window of the observations. However, since the subspace is typically fixed in time, the dynamics of the system are often not correctly captured [11].

Contract/grant sponsor: National Science Foundation, Division of Mathematical Sciences; contract/grant number: 0915107

Another approach is to recast the filtering problem in variational form. In weather forecasting, for example, 3D-Var relies on a variational formulation of the Kalman filter [23], whereas the current state of the art is 4D-Var (see [11, 31]), which utilizes a variational formulation of an initial value estimation problem [12, 20, 23], and has been shown to be identical to a Kalman smoother when the model is assumed to be perfect [22].

The quadratic minimization problems required for implementation of 3D- and 4D-Var are large-scale (10^4-10^7 unknowns), and so efficient numerical optimization methods are needed. Similar to the "reduced rank" methods mentioned above, the partial orthogonal decomposition is used in [7] to reduce the dimensionality of the 4D-Var minimization problem. A more standard approach is to implement a preconditioned conjugate gradient method [10, 13, 28, 40, 41, 43].

In this paper we propose the use of conjugate gradient (CG) for quadratic minimization, as well as for the computation of "low rank" approximations of covariance and inverse-covariance matrices. It is well-known that when CG is applied to the minimization of

$$\phi(\mathbf{x}) = \frac{1}{2}\langle \mathbf{A}\mathbf{x}, \mathbf{x}\rangle - \langle \mathbf{b}, \mathbf{x}\rangle,$$

where $\mathbf{A}$ is symmetric, positive definite, and $n \times n$, and $\mathbf{x}$ and $\mathbf{b}$ are $n \times 1$, the minimizer, $\mathbf{A}^{-1}\mathbf{b}$ is obtained in at most n steps. Lesser known is the fact that CG can be used to efficiently build low rank approximations of both $\mathbf{A}^{-1}$ and $\mathbf{A}$ using the close connection between the CG and Lanczos iterations. Using such low rank approximations, we are able to obtain computationally efficient, low storage implementations of KF, and of a variational formulation of KF, which we call VKF, and which is similar to 3D-Var. The CG-Lanczos connection is also exploited in [40], where it is used instead to build preconditioners for CG iterations within 4D-Var.

Our overall approach is similar to that of [1, 2], the difference being that we use CG, rather than limited memory BFGS (LBFGS) [28], for quadratic minimization, as well for constructing low storage covariance and inverse-covariance approximations.

The problem of using CG to build covariance and inverse-covariance approximations has been studied extensively by Schneider and Willsky [34, 35]. These authors even suggest the use of their ideas in the context of the Kalman filter applied to an oceanography problem [36], but provide no mathematical detail regarding their implementation. In this paper, we introduce two different approximate Kalman filters, each of which employs CG for both quadratic minimization as well as for obtaining low rank covariance and inverse-covariance approximations.

The paper is organized as follows. In Section 2, we present the Kalman filter (KF) and variation Kalman filter (VKF), and provide a general outline of the approximate filters, which we denote CG-KF and CG-VKF. In Section 3, we provide details on how to use CG/Lanczos to compute low rank approximations of covariance and inverse-covariance matrices, and provide a rigorous analysis of these approximations in the appendix in Section 6. CG-KF and CG-VKF are tested on two numerical examples in Section 4, and we provide conclusions in Section 5.

2. Kalman Filtering Methods

We begin our mathematical discussion by considering the following coupled system of discrete, linear, stochastic difference equations

$$\mathbf{x}_k = \mathbf{M}_k \mathbf{x}_{k-1} + \boldsymbol{\varepsilon}_k^p, \tag{1}$$
$$\mathbf{y}_k = \mathbf{K}_k \mathbf{x}_k + \boldsymbol{\varepsilon}_k^o. \tag{2}$$

In the first equation, $\mathbf{x}_k$ denotes the $n \times 1$ state vector of the system at time k; $\mathbf{M}_k$ is the $n \times n$ linear evolution operator; and $\boldsymbol{\varepsilon}_k^p$ is a $n \times 1$ random vector representing the prediction error and is assumed to characterize errors in the model and in the corresponding numerical approximations. In the second equation, $\mathbf{y}_k$ denotes the $m \times 1$ observed data vector; $\mathbf{K}_k$ is the $m \times n$ linear observation operator; and $\boldsymbol{\varepsilon}_k^o$ is an $m \times 1$ random vector representing the observation error. The error terms are assumed to be independent and Normally distributed, with zero mean and with covariance matrixes $\mathbf{C}_{\boldsymbol{\varepsilon}_k^p}$ and $\mathbf{C}_{\boldsymbol{\varepsilon}_k^o}$, respectively.

The task is to estimate the state $\mathbf{x}_k$ and its error covariance $\mathbf{C}_k$ at time point k given $\mathbf{y}_k$, $\mathbf{K}_k$, $\boldsymbol{\varepsilon}_k^o$, $\mathbf{M}_k$, $\boldsymbol{\varepsilon}_k^p$ and estimates $\mathbf{x}_{k-1}^{est}$ and $\mathbf{C}_{k-1}^{est}$ of the state and covariance at time point $k-1$.

The classical Kalman filter (KF) is the standard approach taken for such problems. It is optimal in the sense that it yields minimum variance estimator [17] and has the following form.

Algorithm 1 (KF): *Select initial guess* $\mathbf{x}_0^{est}$ *and covariance* $\mathbf{C}_0^{est}$*, and set* $k = 1$*.*

1. *Compute the evolution model estimate and covariance:*
   (a) *Compute* $\mathbf{x}_k^p = \mathbf{M}_k \mathbf{x}_{k-1}^{est}$*;*
   (b) *Compute* $\mathbf{C}_k^p = \mathbf{M}_k \mathbf{C}_{k-1}^{est} \mathbf{M}_k^{\mathrm{T}} + \mathbf{C}_{\boldsymbol{\varepsilon}_k^p}$*.*
2. *Compute Kalman filter estimate and covariance:*
   (a) *Compute the Kalman Gain* $\mathbf{G}_k = \mathbf{C}_k^p \mathbf{K}_k^{\mathrm{T}} (\mathbf{K}_k \mathbf{C}_k^p \mathbf{K}_k^{\mathrm{T}} + \mathbf{C}_{\boldsymbol{\varepsilon}_k^o})^{-1}$*;*
   (b) *Compute the Kalman filter estimate* $\mathbf{x}_k^{est} = \mathbf{x}_k^p + \mathbf{G}_k(\mathbf{y}_k - \mathbf{K}_k \mathbf{x}_k^p)$*;*
   (c) *Compute the estimate covariance* $\mathbf{C}_k^{est} = \mathbf{C}_k^p - \mathbf{G}_k \mathbf{K}_k \mathbf{C}_k^p$*.*
3. *Update* $k := k + 1$ *and return to Step 1.*

An equivalent variational formulation of KF follows from a sequential application of Bayes' Theorem. To see this, we recall Bayes' formula

$$p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y}) \propto p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x}) p_{\mathbf{x}}(\mathbf{x}), \tag{3}$$

where $\mathbf{x}$ is the vector of unknowns, $\mathbf{y}$ the measurements, $p_{\mathbf{x}}$ denotes the prior density, and $p_{\mathbf{y}|\mathbf{x}}$ is the density of the likelihood function. The maximum a posterior (MAP) estimate is obtained by maximizing (3). Equivalently, one can minimize

$$\ell(\mathbf{x}|\mathbf{y}) = -\log p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x}) - \log p_{\mathbf{x}}(\mathbf{x}). \tag{4}$$

For the linear model (2) at time k with Normally distributed error, the function ℓ assumes the form

$$\ell(\mathbf{x}|\mathbf{y}_k) = \frac{1}{2}(\mathbf{y}_k - \mathbf{K}_k \mathbf{x})^{\mathrm{T}} \mathbf{C}_{\boldsymbol{\varepsilon}_k^o}^{-1} (\mathbf{y}_k - \mathbf{K}_k \mathbf{x}) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_k^p)^{\mathrm{T}} (\mathbf{C}_k^p)^{-1} (\mathbf{x} - \mathbf{x}_k^p), \tag{5}$$

where $\mathbf{C}_{\boldsymbol{\varepsilon}_k^o}$ and $\mathbf{C}_k^p$ are the covariance matrices of the measurement noise $\boldsymbol{\varepsilon}_k^o$ and of the prior $\mathbf{x}_k^p$, respectively.

The Kalman filter estimate and its covariance $\mathbf{x}_k^{est}$ and $\mathbf{C}_k^{est}$ are precisely the minimizer and inverse Hessian of $\ell(\mathbf{x}|\mathbf{y}_k)$, respectively. This allows us to re-express the KF iteration in a variational form, which we denote the variational Kalman filter (VKF) [2]. We note that VKF is very similar to 3D-Var, which is used in the weather forecasting community.

Algorithm 2 (VKF): *Select initial guess* $\mathbf{x}_0^{est}$ *and covariance* $\mathbf{C}_0^{est}$*, and set* $k = 1$*.*

1. *Compute the evolution model estimate and covariance:*
   (a) *Compute* $\mathbf{x}_k^p = \mathbf{M}_k \mathbf{x}_{k-1}^{est}$*;*
   (b) *Define* $\mathbf{C}_k^p = \mathbf{M}_k \mathbf{C}_{k-1}^{est} \mathbf{M}_k^{\mathrm{T}} + \mathbf{C}_{\boldsymbol{\varepsilon}_k^p}$*;*
2. *Compute variational Kalman filter and covariance estimates:*
   (a) *Compute the minimizer* $\mathbf{x}_k^{est}$ *and inverse Hessian* $\mathbf{C}_k^{est}$ *of* $\ell(\mathbf{x}|\mathbf{y}_k) = (\mathbf{y}_k - \mathbf{K}_k\mathbf{x})^{\mathrm{T}}(\mathbf{C}_{\boldsymbol{\varepsilon}_k^o})^{-1}(\mathbf{y}_k - \mathbf{K}_k\mathbf{x}) + (\mathbf{x} - \mathbf{x}_k^p)^{\mathrm{T}}(\mathbf{C}_k^p)^{-1}(\mathbf{x} - \mathbf{x}_k^p)$ *to obtain* $\mathbf{x}_k^{est}$ *and* $\mathbf{C}_k^{est}$*;*
3. *Update* $k := k + 1$ *and return to Step 1.*

2.1. Extensions to nonlinear models

Nonlinear extensions of KF and VKF have been developed for the case when (1), (2) are replaced by

$$\mathbf{x}_k = \mathcal{M}(\mathbf{x}_{k-1}) + \boldsymbol{\varepsilon}_k^p, \tag{6}$$

$$\mathbf{y}_k = \mathcal{K}(\mathbf{x}_k) + \boldsymbol{\varepsilon}_k^o, \tag{7}$$

where $\mathcal{M}$ and $\mathcal{K}$ are possibly nonlinear functions.

The best known extension is the extended Kalman filter (EKF). EKF is obtained by the following modification of the KF algorithm: in Step 1, (i), use the nonlinear model $\mathbf{x}_k^p = \mathcal{M}(\mathbf{x}_k^{est})$ to compute the prior, but otherwise use the following linearized approximations of $\mathcal{M}$ and $\mathcal{K}$:

$$\mathbf{M}_k = \frac{\partial \mathcal{M}(\mathbf{x}_{k-1}^{est})}{\partial \mathbf{x}}, \quad \text{and} \quad \mathbf{K}_k = \frac{\partial \mathcal{K}(\mathbf{x}_k^p)}{\partial \mathbf{x}}. \tag{8}$$

Exactly the same changes can be made to VKF to incorporate nonlinear evolution and observation models. This is the approach we take for the nonlinear example in this paper.

The linearizations $\mathbf{M}_k$ and $\mathbf{K}_k$ in (8) can be computed or estimated in a number of ways. A common approach is to use finite differences. A more efficient approach – both computationally and in terms of storage – employs the adjoint and tangent linear codes defined by the numerical scheme(s) used in the solution of the evolution and/or the observation model. These codes are available in many important instances, e.g. in weather forecasting [20]. The tangent code for the evolution and observation operators computes multiplication of a vector by $\mathbf{M}_k$ and $\mathbf{K}_k$, respectively; whereas the adjoint code for the evolution and observation operators computes multiplication of a vector by $\mathbf{M}_k^T$ and $\mathbf{K}_k^T$, respectively.

2.2. Efficient KF and VKF algorithms using conjugate gradient

For large-scale problems, KF, EKF, and VKF can be prohibitively expensive to implement due to the need for the storage and inversion of large, non-sparse matrices at every iteration. To address this challenge, we advocate the use of iterative methods within the filters for both the solution of linear systems (or quadratic minimization), as well as for obtaining low storage approximations of these matrices. In [1, 2], the limited memory BFGS method was used for this purpose, while in this paper, we describe how to use the conjugate gradient method (CG) for efficient implementation of these filtering methods.

We assume that multiplication by the evolution and observation matrices $\mathbf{M}_k$ and $\mathbf{K}_k$, and their transposes, are efficient, both in terms of storage and CPU time.

The conjugate gradient method (CG) is a well-known iterative method for minimizing quadratic functions of the type

$$\phi(\mathbf{x}) = \frac{1}{2}\langle \mathbf{A}\mathbf{x}, \mathbf{x}\rangle - \langle \mathbf{b}, \mathbf{x}\rangle, \tag{9}$$

where $\mathbf{A}$ is an $n \times n$ symmetric positive definite matrix and $\mathbf{b}$ is an $n \times 1$ vector [28]. After k iterations of CG, we obtain an approximate minimizer of ϕ, which we denote $\mathbf{x}_{CG}^k$. We will show later that the CG iteration history can also be used to create an approximation $\mathbf{B}_k$ of $\mathbf{A}^{-1}$, and that both $\mathbf{x}_{CG}^k$ and $\mathbf{B}_k$ are optimal approximations over a certain Krylov subspace. Moreover, we will show how to use the connection between CG and the Lanczos iterations to efficiently and stably compute both $\mathbf{B}_k$ and its psuedo-inverse $\mathbf{B}_k^\dagger$, which approximates $\mathbf{A}$, also from CG iteration history.

In order to illustrate how CG is used within KF and VKF, we include psuedo-code for the approximate filtering methods.

Algorithm 3 (CG-KF): *Select initial guess* $\mathbf{x}_0^{est}$ *and covariance* $(\mathbf{B}_0^\#)^\dagger = \mathbf{C}_0^{est}$*, and set* $k = 1$*.*

1. *Compute the evolution model estimate and covariance:*
   (a) *Compute* $\mathbf{x}_k^p = \mathbf{M}_k \mathbf{x}_{k-1}^{est}$*;*
   (b) *Compute* $\mathbf{C}_k^p = \mathbf{M}_k (\mathbf{B}_{k-1}^\#)^\dagger \mathbf{M}_k^{\mathrm{T}} + \mathbf{C}_{\boldsymbol{\varepsilon}_k^p}$*.*
2. *Compute Kalman filter estimate and covariance:*
   (a) *Apply CG to (9), with* $\mathbf{A} = \mathbf{K}_k \mathbf{C}_k^p \mathbf{K}_k^{\mathrm{T}} + \mathbf{C}_{\boldsymbol{\varepsilon}_k^o}$ *and* $\mathbf{b} = (\mathbf{y}_k - \mathbf{K}_k \mathbf{x}_k^p)$*, to obtain* $\mathbf{x}_k^*$*, and* $\mathbf{B}_k^*$*;*
   (b) *Compute approximate Kalman filter estimate* $\mathbf{x}_k^{est} = \mathbf{x}_k^p + \mathbf{C}_k^p \mathbf{K}_k^{\mathrm{T}} \mathbf{x}_k^*$*;*
   (c) *Apply CG to (9) with* $\mathbf{A} = \mathbf{C}_k^p - \mathbf{C}_k^p \mathbf{K}_k^{\mathrm{T}} \mathbf{B}_k^* \mathbf{K}_k \mathbf{C}_k^p$ *and* $\mathbf{b} = \mathbf{v}$*, to obtain* $(\mathbf{B}_k^\#)^\dagger$*;*
3. *Update* $k - 1 := k$ *and return to Step 1.*

In step 2(c) (and also 1(b) below), we take $\mathbf{v}$ to be a random vector with entries -1 or 1 with equal probability, chosen to optimize the accuracy of covariance/inverse-covariance approximations. Another, equally effective choice, is to take $\mathbf{v}$ to be a realization from a Gaussian random vector with mean zero and identity covariance. In these steps, it is only the covariance/inverse-covariance approximation that is of interest.

For the variational Kalman filter, we make similar modifications to obtain the following approximate filtering algorithm.

Algorithm 4 (CG-VKF): *Select initial guess* $\mathbf{x}_0^{est}$ *and covariance* $\mathbf{B}_0^\# = \mathbf{C}_0^{est}$*, and set* $k = 1$*.*

1. *Compute the evolution model estimate and covariance:*
   (a) *Compute* $\mathbf{x}_k^p = \mathbf{M}_k \mathbf{x}_{k-1}^{est}$;
   (b) *Apply CG to (9) with* $\mathbf{A} = \mathbf{M}_k \mathbf{B}_{k-1}^{\#} \mathbf{M}_k^{\mathrm{T}} + \mathbf{C}_{\boldsymbol{\varepsilon}_k^p}$ *and* $\mathbf{b} = \mathbf{v}$ *to obtain* $\mathbf{B}_k^*$;
2. *Compute variational Kalman filter and covariance estimates:*
   (a) *Apply CG to the problem of minimizing* $\ell(\mathbf{x}|\mathbf{y}_k) = \frac{1}{2}(\mathbf{y}_k - \mathbf{K}_k\mathbf{x})^{\mathrm{T}}(\mathbf{C}_{\boldsymbol{\varepsilon}_k^o})^{-1}(\mathbf{y}_k - \mathbf{K}_k\mathbf{x}) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_k^p)^{\mathrm{T}}\mathbf{B}_k^*(\mathbf{x} - \mathbf{x}_k^p)$, *which has the form (9) with* $\mathbf{A} = \mathbf{K}_k^{\mathrm{T}}(\mathbf{C}_{\boldsymbol{\varepsilon}_k^o})^{-1}\mathbf{K}_k + \mathbf{B}_k^*$ *and* $\mathbf{b} = \mathbf{K}_k^T(\mathbf{C}_{\boldsymbol{\varepsilon}_k^o})^{-1}\mathbf{y}_k + \mathbf{B}_k^*\mathbf{x}_k^p$, *to obtain* $\mathbf{x}_k^{est}$ *and* $\mathbf{B}_k^{\#}$;
3. *Update* $k - 1 := k$ *and return to Step 1.*

Extensions of CG-KF and CG-VKF to the nonlinear case (6), (7), are exactly as for KF and VKF.

It remains to show how to construct the covariance and inverse-covariance approximations ($\mathbf{B}_k^*$ and $(\mathbf{B}_k^{\#})^{\dagger}$ in CG-KF, and $\mathbf{B}_k^*$ and $\mathbf{B}_k^{\#}$ in CG-VKF) above from CG iteration history. We do this in the next section.

3. Krylov space approximation of $\mathbf{A}$ and $\mathbf{A}^{-1}$

In this section, we provide necessary details for the implementation of CG within both CG-KF and CG-VKF. In particular, focusing on the general minimization problem (9), we present mathematical results regarding the efficient computation of both $\mathbf{B}_k^{\dagger} \approx \mathbf{A}$ and $\mathbf{B}_k \approx \mathbf{A}^{-1}$ from CG iterations. For this, we exploit the Lanczos iteration and its close connection to CG. We also present a mathematical analysis of these approximations.

3.1. Conjugate Gradient Iteration

Given a symmetric, positive definite $n \times n$ matrix $\mathbf{A}$, the CG method is an iterative algorithm for solving the linear system $\mathbf{Ax} = \mathbf{b}$ or, equivalently, for minimizing a quadratic of the form (9) [16]. In order to establish necessary notation, we present the CG iteration next [3, 26, 28].

Algorithm 5 (CG): *Given* $\mathbf{A}$, $\mathbf{b}$ *and* $\mathbf{x}^0$, *let* $\mathbf{r}^0 = \mathbf{b} - \mathbf{A}\mathbf{x}^0$, $\mathbf{p}^0 = \mathbf{r}^0$, *and* $k = 1$. *Specify some stopping tolerance* ϵ. *Iterate:*

1. $\gamma_{k-1} = \frac{\mathbf{r}^{(k-1)T}\mathbf{r}^{k-1}}{\mathbf{p}^{(k-1)T}\mathbf{A}\mathbf{p}^{k-1}}$ *is the 1-D minimzer of* ϕ *in the direction* $\mathbf{x}^{k-1} + \gamma\mathbf{p}^{k-1}$
2. $\mathbf{x}_{CG}^k = \mathbf{x}_{CG}^{k-1} + \gamma_{k-1}\mathbf{p}^{k-1}$
3. $\mathbf{r}^k = -\nabla_x\phi(\mathbf{x}_{CG}^k) = \mathbf{b} - \mathbf{A}\mathbf{x}_{CG}^k = \mathbf{r}^{k-1} - \gamma_{k-1}\mathbf{A}\mathbf{p}^{k-1}$ *is the residual*
4. $\beta_k = -\frac{\mathbf{r}^{kT}\mathbf{r}^k}{\mathbf{r}^{(k-1)T}\mathbf{r}^{k-1}}$
5. $\mathbf{p}^k = \mathbf{r}^k - \beta_k\mathbf{p}^{k-1}$ *is the next conjugate search direction.*
6. *Quit if* $||\mathbf{r}^k|| < \epsilon$. *Else set* $k := k + 1$ *and go to step 1.*

CG generates a sequence of optimizers, which at the k^{th} iteration can be written as

$$\mathbf{x}_{CG}^k = \mathbf{x}^0 + \mathbf{P}_k\boldsymbol{\gamma}^k$$

where the vector of one dimensional minimizers is $\boldsymbol{\gamma}^k = [\gamma_0, ..., \gamma_{k-1}]^T$ and $\mathbf{P}_k$ is the $n \times k$ matrix with the search directions $\{\mathbf{p}^i\}_{i=0}^{k-1}$ as columns. In exact arithmetic, the $\mathbf{A}$-norm of the error $||\mathbf{x} - \mathbf{A}^{-1}\mathbf{b}||_{\mathbf{A}}$ is minimized by $\mathbf{x} = \mathbf{x}_{CG}^k$ in $\mathbf{x}^0 + \text{span}(\mathbf{p}^0, ..., \mathbf{p}^{k-1})$, and $\mathbf{x}_{CG}^k$ converges to the solution $\mathbf{A}^{-1}\mathbf{b}$ after at most n steps.

The k vectors $\{\mathbf{p}^i\}_{i=0}^{k-1}$ are $\mathbf{A}$-conjugate, which means that $\mathbf{p}^{iT}\mathbf{A}\mathbf{p}^j = 0$ for $i \neq j$. The residuals $\{\mathbf{r}^i\}_{i=0}^{k-1}$, where $\mathbf{r}^i = \mathbf{b} - \mathbf{A}\mathbf{x}_{CG}^i$, are orthogonal. The set of search directions or the residuals are a basis for the Krylov space of dimension k,

$$\mathcal{K}^k(\mathbf{A}, \mathbf{r}^0) = \text{span}(\mathbf{r}^0, \mathbf{A}\mathbf{r}^0, \mathbf{A}^2\mathbf{r}^0, ..., \mathbf{A}^{k-1}\mathbf{r}^0).$$

Let $\mathbf{P_B}$ be the $n \times (n-k)$ matrix whose columns are the conjugate directions $\{\mathbf{p}^i\}_{i=k}^{n-1}$. Then $\mathbf{P}_n = [\mathbf{P}_k \;\; \mathbf{P}_B]$ is invertible and

$$\mathbf{D}_n = \begin{pmatrix} \mathbf{D}_k & 0 \\ 0 & \mathbf{D}_B \end{pmatrix} = \begin{pmatrix} \mathbf{P}_k^T\mathbf{A}\mathbf{P}_k & 0 \\ 0 & \mathbf{P}_B^T\mathbf{A}\mathbf{P}_B \end{pmatrix} = \mathbf{P}_n^T\mathbf{A}\mathbf{P}_n$$

is an invertible diagonal matrix, $[\mathbf{D}_n]_{ii} = ||\mathbf{p}^i||_{\mathbf{A}}^2 = \mathbf{p}^{iT}\mathbf{A}\mathbf{p}^i$. Thus

$$\mathbf{A}^{-1} = \mathbf{P}_n\mathbf{D}_n^{-1}\mathbf{P}_n^T = \mathbf{P}_k\mathbf{D}_k^{-1}\mathbf{P}_k^T + \mathbf{P}_B\mathbf{D}_B^{-1}\mathbf{P}_B^T.$$

For $k < n$, the k-rank approximation of $\mathbf{A}^{-1}$ produced by the CG algorithm is

$$\mathbf{B}_k = \mathbf{P}_k\mathbf{D}_k^{-1}\mathbf{P}_k^T. \tag{10}$$

We will show in Section 3.3 below that this approximation is optimal in the Krylov space spanned by the conjugate directions $\{\mathbf{p}^i\}$.

Remark 1. *We use (10) to define $\mathbf{B}_k$ within CG-KF, Step 2(a) and CG-VKF, Steps 1(b) and 2(a). To minimize storage requirements, $\mathbf{P}_k$ and $\mathbf{D}_k$ are saved and used to define $\mathbf{B}_k$ implicitly.*

3.2. Lanczos Iteration

The Lanczos algorithm is an iterative method for solving the eigenvalue problem for large sparse matrices [18, 19], and its performance in finite precision is well studied [26, 30, 32]. CG is equivalent to the Lanczos method for symmetric, positive definite matrices [3, 5, 8, 15, 26].

In exact arithmetic, the Lanczos algorithm determines the eigenpairs $(\lambda_i, \mathbf{w}^i)$ of a given $n \times n$ positive definite matrix $\mathbf{A}$, so that $\mathbf{A}\mathbf{w}^i = \lambda_i\mathbf{w}^i$. In order to establish necessary notation, we provide the two-term recurrence version of the Lanczos algorithm due to Paige [26].

Algorithm 6 (Lanczos): *Given an initial vector $\tilde{\mathbf{v}}^0$, let $\mathbf{v}^0 = \frac{\tilde{\mathbf{v}}^0}{||\tilde{\mathbf{v}}^0||}$, $\alpha_0 = \mathbf{v}^{0T}\mathbf{A}\mathbf{v}^0$, $\tilde{\mathbf{v}}^1 = \mathbf{A}\mathbf{v}^0 - \alpha_0\mathbf{v}^0$, and $k = 1$. Specify some stopping tolerance ϵ. Iterate:*

1. $\eta_k = ||\tilde{\mathbf{v}}^k||$. *Quit if* $\eta_k < \epsilon$.
2. $\mathbf{v}^k = \frac{\tilde{\mathbf{v}}^k}{\eta_k}$ *is a Lanczos vector.*
3. $\mathbf{u}^k = \mathbf{A}\mathbf{v}^k - \eta_k\mathbf{v}^{k-1}$
4. $\alpha_k = \mathbf{v}^{kT}\mathbf{u}^k$
5. $\tilde{\mathbf{v}}^{k+1} = \mathbf{u}^k - \alpha_k\mathbf{v}^k$
6. $k := k + 1$ *and go to step 1.*

At the k^{th} iteration, the Lanczos algorithm is equivalent to the matrix equation

$$\mathbf{A}\mathbf{V}_k = \mathbf{V}_k\mathbf{T}_k + \eta_k\mathbf{v}^k\mathbf{e}^{kT} \tag{11}$$

where $\mathbf{e}^i$ is the i^{th} column of the $k \times k$ identity $\mathbf{I}_k$, the Lanczos vector $\mathbf{v}^{i-1}$ is the i^{th} column of the $n \times k$ matrix $\mathbf{V}_k$, and the tridiagonal Lanczos matrix is

$$\mathbf{T}_k = \begin{pmatrix} \alpha_0 & \eta_1 & & & \\ \eta_1 & \alpha_1 & \eta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \eta_{k-2} & \alpha_{k-2} & \eta_{k-1} \\ & & & \eta_{k-1} & \alpha_{k-1} \end{pmatrix}.$$

Thus, if $\eta_k = 0$, then range$(\mathbf{V}_k)$ is invariant to multiplication by $\mathbf{A}$.

By construction, in exact arithmetic, the k Lanczos vectors are an orthonormal basis for $\mathcal{K}^k(\mathbf{A}, \mathbf{v}^0)$ ([26] p. 81), which shows that the tridiagonal matrix is

$$\mathbf{T}_k = \mathbf{V}_k^T\mathbf{A}\mathbf{V}_k.$$

This equation suggests the following low rank approximation of $\mathbf{A}$:

$$\mathbf{B}_k^\dagger = \mathbf{V}_k\mathbf{T}_k\mathbf{V}_k^T. \tag{12}$$

In section 3.3 we will prove that $\mathbf{V}_k\mathbf{T}_k\mathbf{V}_k^T$ is in fact the generalized inverse of $\mathbf{B}_k$ in equation (10), and that it is optimal in the Krylov space spanned by the conjugate directions.

An added benefit of decomposition (12) is that from it an efficient approximate eigenvalue decomposition for $\mathbf{A}$ can be computed. The spectral decomposition of $\mathbf{T}_k$ can be efficiently obtained via QR iteration, bisection, or divide and conquer schemes [8, 15, 30] since it is a tridiagonal matrix. The eigenvalues of $\mathbf{T}_k$, called Ritz values, are distinct,

$$\theta_1^k < ... < \theta_k^k,$$

and are the Lanczos estimates of k of the eigenvalues of $\mathbf{A}$. We express the spectral decomposition as $\mathbf{T}_k = \mathbf{Y}_k\mathbf{\Theta}_k\mathbf{Y}_k^T$, where $\mathbf{\Theta}_k = \text{diag}(\theta_1^k, ..., \theta_k^k)$ and $\mathbf{Y}_k$ has the corresponding eigenvectors $\mathbf{y}^i$ as columns. Thus, the approximate eigen-decomposition of $\mathbf{A}$ is

$$\mathbf{B}_k^\dagger = (\mathbf{V}_k\mathbf{Y}_k)\mathbf{\Theta}_k(\mathbf{V}_k\mathbf{Y}_k)^T, \tag{13}$$

with the Ritz pairs $\{(\theta_i^k, \mathbf{V}_k\mathbf{y}^i)\}$ approximating k of the eigenpairs of $\mathbf{A}$.

Remark 2. *We use covariance approximation (13) within CG-KF, Step 2(c). To minimize storage requirements, $\mathbf{V}_k\mathbf{Y}_k$ and $\mathbf{\Theta}_k$ are saved and used to define $\mathbf{B}_k$ implicitly. Our results in the next section show how $\mathbf{V}_k$ and $\mathbf{T}_k$ are obtained from CG iteration history.*

3.3. Equivalence of the CG and Lanczos Approximations

To obtain $\mathbf{V}_k$ from CG iterations, note that since the normalized CG residual vectors $\frac{\mathbf{r}^0}{||\mathbf{r}^0||_2}, \ldots, \frac{\mathbf{r}^{k-1}}{||\mathbf{r}^{k-1}||_2}$ form an orthonormal basis for $\mathcal{K}^k(\mathbf{A}, \mathbf{r}^0)$ for any k, then, if the Lanczos algorithm is initialized with $\tilde{\mathbf{v}}^0 = \mathbf{r}^0$, up to a sign change, the Lanczos vectors are normalized CG residuals. Indeed, careful inspection ([8] p. 99, [15], [26] p. 50) shows that

$$\mathbf{v}^k = (-1)^k\frac{\mathbf{r}^k}{||\mathbf{r}^k||}. \tag{14}$$

Lanczos vectors constructed in this way are called CG-Lanczos vectors in ([26] p. 195) since their behavior can be different than those from the Lanczos algorithm in finite precision.

Next, we present and prove a Lemma, which gives an important relationship between the conjugate directions and the Lanczos vectors. Define the $k \times k$ diagonal matrices

$$\mathbf{\Delta}_k = \text{diag}(||\mathbf{r}^0||, ..., ||\mathbf{r}^{k-1}||), \qquad \mathbf{N}_k = \text{diag}(1, -1, 1, ...)$$

and let $\mathbf{C}_k$ be the upper bi-diagonal matrix

$$\mathbf{C}_k = \begin{pmatrix} 1 & \beta_1 & & & \\ & \ddots & \ddots & & \\ & & & 1 & \beta_{k-1} \\ & & & & 1 \end{pmatrix},$$

where the β_i's were defined in the CG algorithm.

Lemma 3. *A QR factorization of the matrix of conjugate directions $\mathbf{P}_k$ is*

$$\mathbf{P}_k = \mathbf{V}_k \mathbf{R}_k \qquad \textit{where} \qquad \mathbf{R}_k = \mathbf{V}_k^T \mathbf{P}_k = \mathbf{N}_k \mathbf{\Delta}_k \mathbf{C}_k^{-1},$$

$\mathbf{V}_k$ is the matrix of orthonormal Lanczos vectors, and $\mathbf{R}_k$ is upper triangular. Furthermore, the columns of $\mathbf{R}_k$ are $\mathbf{T}_k$-conjugate, so that

$$\mathbf{T}_k^{-1} = \mathbf{R}_k \mathbf{D}_k^{-1} \mathbf{R}_k^T \tag{15}$$

and $\mathbf{P}_k \mathbf{D}_k^{-1} \mathbf{P}_k^T = \mathbf{V}_k \mathbf{T}_k^{-1} \mathbf{V}_k^T$.

Proof: In the CG algorithm the relation $\mathbf{r}^k = \mathbf{p}^k + \beta_k \mathbf{p}^{k-1}$ can be written as $\mathbf{V}_k = \mathbf{P}_k \mathbf{C}_k \Delta_k^{-1} \mathbf{N}_k$ which shows that range($\mathbf{P}_k$)=range($\mathbf{V}_k$). Letting $\mathbf{R}_k = \mathbf{N}_k \mathbf{\Delta}_k \mathbf{C}_k^{-1}$ yields a QR-factorization $\mathbf{P}_k = \mathbf{V}_k \mathbf{R}_k$ since $\mathbf{R}_k = \mathbf{V}_k^T \mathbf{P}_k$ is upper triangular with elements $[\mathbf{R}_k]_{ij} = (-1)^{i-1} \frac{\mathbf{r}^{(i-1)T} \mathbf{p}^{(j-1)}}{||\mathbf{r}^{i-1}||}$ which are zero for all $i > j$. Now the statement of conjugacy $\mathbf{P}_k^T \mathbf{A} \mathbf{P}_k = \mathbf{D}_k$ can be re-written as $\mathbf{R}_k^T \mathbf{V}_k^T \mathbf{A} \mathbf{V}_k \mathbf{R}_k = \mathbf{R}_k^T \mathbf{T}_k \mathbf{R}_k = \mathbf{D}_k$ and so $\mathbf{P}_k \mathbf{D}_k^{-1} \mathbf{P}_k^T = \mathbf{V}_k \mathbf{R}_k \mathbf{D}_k^{-1} \mathbf{R}_k^T \mathbf{V}_k^T = \mathbf{V}_k \mathbf{T}_k^{-1} \mathbf{V}_k^T$. □

Theorem 4. *Let $\mathbf{T}_k$ be the Lanczos tridiagonal matrix with eigenpairs $(\theta_i^k, \mathbf{y}^i)$, $\mathbf{P}_k$ be the matrix with k conjugate directions $\{\mathbf{p}^i\}$ as columns, $\mathbf{D}_k = \text{diag}(||\mathbf{p}^0||_A, ..., ||\mathbf{p}^{k-1}||_A)$, and $\mathbf{V}_k$ be the orthonormal matrix with k Lanczos vectors $\{\mathbf{v}^i\}$ as columns. The CG approximation of $\mathbf{A}^{-1}$ is*

$$\mathbf{B}_k = \mathbf{P}_k \mathbf{D}_k^{-1} \mathbf{P}_k^T = \mathbf{V}_k \mathbf{T}_k^{-1} \mathbf{V}_k^T$$

and has k non-zero eigenvalues $\{1/\theta_i^k\}$ which are the Lanczos estimates of the eigenvalues of $\mathbf{A}^{-1}$. The CG approximation of $\mathbf{A}$ is the generalized inverse

$$\mathbf{B}_k^\dagger = \mathbf{V}_k \mathbf{T}_k \mathbf{V}_k^T$$

which has k non-zero eigenvalues $\{\theta_i^k\}$ which are the Lanczos estimates of the eigenvalues of $\mathbf{A}$. The eigenvectors of $\mathbf{B}_k$ and $\mathbf{B}_k^\dagger$ are $\{\mathbf{V}_k \mathbf{y}^i\}$, the Lanczos approximations of the eigenvectors of $\mathbf{A}$.

Proof: The result follows from equation (10) and Lemma 3. □

Remark 5. *For covariance approximation (12), we obtain* $\mathbf{V}_k$ *from CG iterations via (14) and* $\mathbf{T}_k^{-1}$ *from (15) with* $\mathbf{R}_k = \mathbf{V}_k^T \mathbf{P}_k$*. Note that* $\mathbf{T}_k$ *can also be obtained from CG via the following relationships:*

$$\alpha_i = \frac{1}{\gamma_i} - \frac{\beta_i}{\gamma_{i-1}} = \frac{d_i}{||\mathbf{r}^i||_2^2} + \frac{d_{i-1}||\mathbf{r}^i||_2^2}{||\mathbf{r}^{i-1}||_2^4}, \quad \textit{and} \quad \eta_{i+1} = \frac{\sqrt{-\beta_{i+1}}}{\gamma_i} = \frac{d_i||\mathbf{r}^{i+1}||_2}{||\mathbf{r}^i||_2^3}, \tag{16}$$

where $0 \leq i < k-1$ *and the convention* $\beta_0 = 0$ *and* $\gamma_{-1} = 1$ *is used [8, 26, 33]. However, in our experience, this is a less numerically stable approach.*

We now have all of the pieces for a numerical implementation of CG-KF and CG-VKF. Thus, in the next section we present our numerical experiments. However, a rigorous theoretical analysis of the CG/Lanczos covariance and inverse covariance approximations, including Krylov subspace optimality results as well as theoretical bounds for the covariance and inverse covariance approximations, is also desirable. The interested reader can find such an analysis in the appendix in Section 6.

4. Numerical Examples

In this section, we test CG-KF and CG-VKF on two examples.

4.1. An Example with a Large-Scale Linear Evolution Model

The first example is large-dimensional and linear. We consider the following forced heat equation model

$$\frac{\partial x}{\partial t} = -\frac{\partial^2 x}{\partial u^2} - \frac{\partial^2 x}{\partial v^2} + \alpha \exp\left[-\frac{(u-2/9)^2 + (v-2/9)^2}{\sigma^2}\right], \tag{17}$$

where x is a function of u and v over the domain $\Omega = \{(u,v) \mid 0 \leq u, v \leq 1\}$ and $\alpha \geq 0$. We generate synthetic data using (17) with $\alpha > 0$ and assume that the evolution model is given by (17) with $\alpha = 0$, which gives a model bias. The problem can be made arbitrarily large-scale via a sufficiently fine spatial discretization. However, the well-behaved nature of solutions of (17) calls for further experiments with a different test case, hence our second example in the next subsection.

We discretize the model (17) using a uniform $N \times N$ computational grid and the standard finite difference schemes of both the time and spatial derivatives. This gives the time stepping equation $\mathbf{x}_{k+1} = \mathbf{M}\mathbf{x}_k + \mathbf{f}$, where $\mathbf{M} = \mathbf{I} - \Delta t \mathbf{L}$. Here $\mathbf{L}$ is given by the standard finite difference discretization of the two-dimensional Laplacian operator with homogeneous Dirichlet boundary conditions, Δt is chosen to guarantee stability, and $\mathbf{f}$ is the constant vector determined by the evaluation of the forcing term in (17) at each of the points of the computational grid. We define $\mathbf{K}_k = \mathbf{K}$ for all k in (2), where $\mathbf{K}$ is the full weighting matrix, which has the following grid representation

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}.$$

Such an observation matrix could model, for example, an array of square heat sensors on the bottom of a metal plate that have dimension $2/N \times 2/N$ with the edges aligned with the grid lines and equally spaced at $n^2/64$ locations.

We first generate synthetic data using the stochastic equations

$$\mathbf{x}_{k+1} = \mathbf{M}\mathbf{x}_k + \mathbf{f}, \tag{18}$$

$$\mathbf{y}_{k+1} = \mathbf{K}\mathbf{x}_{k+1} + N(\mathbf{0}, \sigma^2_{\text{obs}}\mathbf{I}), \tag{19}$$

with $\alpha = 3/4$ in (17) and where σ^2_{obs} is chosen so that the signal-to-noise ratio $\|\mathbf{K}\mathbf{x}_0\|^2/n^2\sigma^2_{\text{obs}}$ is 10. The initial condition used for the data generation is

$$[\mathbf{x}_0]_{ij} = \exp[-((u_i - 1/2)^2 + (v_j - 1/2)^2)],$$

where (u_i, v_j) is the ijth grid point.

For the implementation of our filtering algorithms, we assume the model

$$\mathbf{x}_{k+1} = \mathbf{M}\mathbf{x}_k + N(\mathbf{0}, \sigma^2_{\text{ev}}\mathbf{I}),$$

$$\mathbf{y}_{k+1} = \mathbf{K}\mathbf{x}_{k+1} + N(\mathbf{0}, (10 \cdot \sigma_{\text{obs}})^2\mathbf{I}),$$

with $\sigma^2_{\text{ev}} = \|\mathbf{x}_0\|^2/n^2$, and where $\mathbf{x}_0^{\text{est}} = \mathbf{1}$ and $\mathbf{C}_0^{\text{est}} = \sigma^2_{\text{ev}}\mathbf{I}$ in Step 1 of the filter. Notice that the forcing function $\mathbf{f}$ is not contained in the evolution model, which adds a bias.

In our first example, we assume a 64×64 computational grid. For the CG-VKF method, Step 2(a) is somewhat ill-posed, and hence some regularization is needed. One benefit of the VKF framework is that an additional stabilizing penalty (or regularization) term can easily be added to the cost function ℓ in Step 2 of CG-VKF. We added a term $\alpha\|\mathbf{x} - \mathbf{x}_k^a\|^2$ with $\alpha = 1/2$. Note that this is akin to replacing $\mathbf{B}_k^*$ in Step 2(a) of the CG-VKF algorithm with $\mathbf{B}_k^* + \alpha\mathbf{I}$. Note that no such natural regularization presents itself within the CG-KF iteration, however for this example none was needed.

For all implementations of CG within CG-KF and CG-VKF, we set a maximum of 200 iterations and an additional stopping tolerance of $\|\mathbf{r}_k\| < 10^{-6}$. The implementation of 150 iterations of KF took approximately 850 seconds, CG-KF and CG-VKF both took approximately 9 seconds. The cost of implementing the KF iteration is dominated by the computation of the Cholesky factorization of the matrix whose inverse appears in Step 2(a) of KF (Algorithm 1), as well as the system back-solves with this matrix required in Steps 2(b) and (c). The cost of implementing CG-KF and CG-VKF, on the other hand, is dominated by the matrix-vector multiplies needed in each CG iteration. For this example, the number of CG iterations required to meet the stopping criteria remained relatively low.

Plots of the root mean-square error $\sqrt{\frac{1}{N}\|\mathbf{x}_k - \mathbf{x}_k^{\text{true}}\|}$ for KF, CG-KF, and CG-VKF are given in Figure 1. Note that CG-VKF closely mimics the performance of KF with a two orders-of-magnitude improvement in computational speed, whereas CG-KF appears to yield better results at roughly the same computational cost as CG-VKF.

4.2. An Example with a Small-Scale, Nonlinear Evolution Model

Our second example produces chaotic, unpredictable behavior, but is not as large-scale or ill-conditioned as the last example. We consider the non-linear Lorenz'95 model introduced and analyzed in [24, 25], given by

$$\frac{\partial x^i}{\partial t} = (x^{i+1} - x^{i-2})x^{i-1} - x^i + 8, \qquad i = 1, 2, \ldots, 40, \tag{20}$$

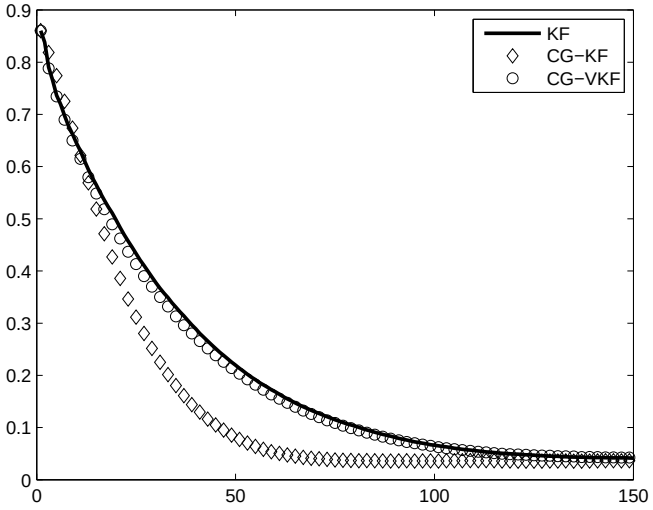

Figure 1. Root mean-square error versus iteration for KF, CG-KF, and CG-VKF.

with periodic state space variables, i.e. $x^{-1} = x^{n-1},\ x^0 = x^n$ and $x^{n+1} = x^1$. In the present tests we use the dimension $n = 40$. The model shares many characteristics with realistic atmospheric models (cf. [25]), and is often used as a test case for various weather forecasting schemes.

Next, we apply the filtering methods to the problem of estimating the state variables from data generated using the nonlinear, chaotic evolution model (20). The data was generated by integrating the model using a fourth order Runge-Kutta (RK4) method with time-step $\Delta t = 0.025$. The discussion in [25] suggests that when using (20) as a test example for weather forecasting algorithms, the characteristic time scale is such that the above Δt corresponds to 3 hours. The "truth" was generated by taking 42920 time steps of the RK4 method, i.e., 5365 days. The initial state for the data generation was $x^{20} = 8 + 0.008$ and $x^i = 8$ for all $i \neq 20$.

The observed data is then computed using this true data. In particular, after a 365 day long initial period, the true data is observed at every other time step and at the last 3 grid points in each set of 5; that is, the observation matrix is $m \times n$, with nonzero entries

$$[\mathbf{K}]_{rs} = \begin{cases} 1 & (r,s) \in \{(3j+i, 5j+i+2) \mid i = 1,2,3,\ j = 0,1,\ldots,7\}, \\ 0 & \text{otherwise.} \end{cases}$$

The observation error is simulated using Gaussian noise $N(\mathbf{0}, (0.15\,\sigma_{\text{clim}})^2\mathbf{I})$ where $\sigma_{\text{clim}} = 3.6414723$ is a standard deviation of the model state used in climatological simulations. The data generation codes are written in MATLAB and were those used in the papers [1, 2] and were originally transcribed by us from the `scilab` codes written by the author of [21].

For the application of EKF and VKF, we employ the coupled system

$$\mathbf{x}_{k+1} = \mathcal{M}(\mathbf{x}_k) + N(\mathbf{0}, (0.05\,\sigma_{\text{clim}})^2\mathbf{I}), \tag{21}$$

$$\mathbf{y}_{k+1} = \mathbf{K}\mathbf{x}_{k+1} + N(\mathbf{0}, (\sigma_{\text{clim}})^2\mathbf{I}), \tag{22}$$

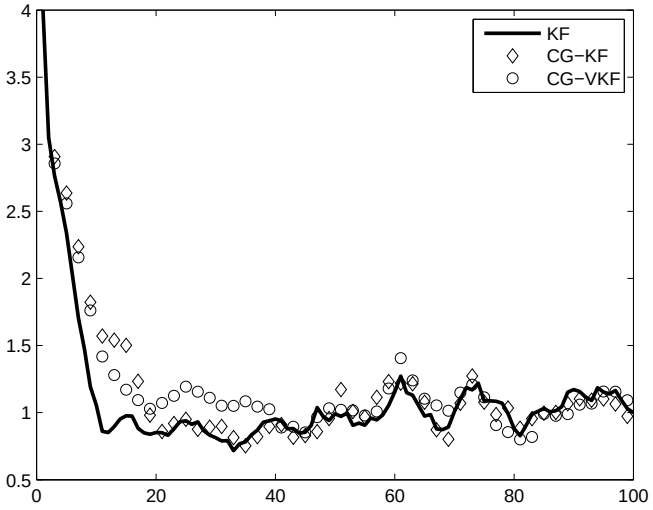

Figure 2. Root mean-square error versus iteration for KF, CG-KF, and CG-VKF.

where $\mathcal{M}(\mathbf{x}_k)$ is obtained by taking two steps of the RK4 method applied to (20) from $\mathbf{x}_k$ with time-step 0.025. We note that this coincides with the data generation scheme, if the noise term is removed and the above initial condition is used. Due to the fact that $\mathcal{M}$ is a nonlinear function, EKF must be used (see equations (6) and (7). Since $\mathcal{K} := \mathbf{K}$ in (7) is linear, $\mathbf{K}_k = \mathbf{K}$ for all k in (8). However a linearization of the nonlinear evolution function $\mathcal{M}$ is required. The computation of $\mathbf{M}_k$ in (8) is performed by a routine in one of the `scilab` codes mentioned above, adopted for our use in MATLAB.

The initial guesses for the Kalman filter iterations were $\mathbf{x}_0 = \mathbf{1}$ and $\mathbf{C}_0 = \mathbf{I}$. In all implementations of CG within CG-KF and CG-VKF, a maximum number of iterations of 50 was allowed, and iterations were stopped once $\|\mathbf{r}_k\| < 10^{-6}$. Note that since the dimensionality of this problem is only 40, the finite precision issues of CG have less of an impact. Thus no regularization via the truncation of CG iterations was needed for CG-KF. However, regularization was still needed for CG-VKF; we added at penalty of the form $\alpha\|\mathbf{x} - \mathbf{x}_k^a\|$ to ℓ in Step 2 of CG-VKF, this time with $\alpha = 5$.

The results are given in Figure 2, where it is readily seen that CG-KF and CG-VKF, track closely with the KF estimates after a sufficient number of iterations. We note that different noise realizations were used in each case, so a part of the difference between the methods in the early iterations is likely due to this difference.

5. Conclusions

For large-scale examples, such as arise in weather forecasting and oceanography, the Kalman filter (KF) can be prohibitively expensive to implement due to the fact that it requires the

solution of large linear systems and the storage of large matrices. In this paper, we show how the conjugate gradient (CG) algorithm can be used for both the approximate solution of large linear systems, as well as for obtaining low rank and low storage approximations of matrices within KF.

It is well-known that CG is very efficient for approximately solving large, symmetric positive definite linear systems. Lesser known is that fact that its iteration history can be used to cheaply obtain approximations of the inverse of the coefficient matrix of the system, as well as of the coefficient matrix itself using the CG/Lanzcos connection. In k CG iterations, this inverse approximation has rank k and requires the storage of only k n-vectors, as well as a $k \times k$ diagonal matrix. To obtain the approximation of the coefficient matrix itself, the $k \times k$ tridiagonal Lanzcos matrix $\mathbf{T}_k$ must be diagonalized, which can be efficiently done for k of small-to-moderate size. Multiplication by these approximate matrices is also efficient.

More specifically, our implementation of CG-VKF made use only of approximation (10), whereas CG-KF used (10), as well as (12), which required that we exploit the connection between CG and the Lanczos algorithm. A proof of the equivalence of (10) and (12) was provided and the results there were used both in our numerical calculations of Lanczos-from-CG, as well as in the theoretical analysis of the covariance approximations contained in the appendix.

The resulting algorithms, which we have denoted the CG Kalman filter (CG-KF) and CG variational Kalaman filter (CG-VKF), are much more efficient to implement than KF for the two numerical examples considered here – one large-scale and linear and the other medium-scale and nonlinear – and provide results that are comparable.

6. Appendix: Analysis of CG and Lanczos Covariance Approximations

In this section, we address several important questions regarding the CG/Lanczos covariance approximations in a mathematically rigorous fashion. We begin with an analysis of the Lanczos approximation (12), and show that approximation (12) of $\mathbf{A}$ is optimal in a certain Krylov subspace. We then provide theoretical bounds for the covariance approximations, and end with a discussion of the effect of finite precision arithmetic.

6.1. Analysis of Lanczos Approximation

Order the eigenvalues of $\mathbf{A}$ by

$$\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n \quad \text{or} \quad \lambda_{-1} \geq \lambda_{-2} \geq ... \geq \lambda_{-n},$$

and let $\mathbf{w}^i$ be the corresponding eigenvectors. Similarly order the distinct eigenvalues of $\mathbf{T}_k$ by either

$$\theta_1^k < ... < \theta_k^k \quad \text{or} \quad \theta_{-1}^k > ... > \theta_{-k}^k$$

with corresponding eigenvectors denoted by $\mathbf{y}^i$.

The fact that $\mathbf{T}_k = \mathbf{V}_k^T \mathbf{A} \mathbf{V}_k$ shows that the Lanczos algorithm is a Raleigh-Ritz process, with $\mathbf{T}_k$ the minimizer of

$$\rho(\boldsymbol{\zeta}|\mathbf{V}_k) := ||\mathbf{A}\mathbf{V}_k - \mathbf{V}_k\boldsymbol{\zeta}||_2$$

[30]. In other words,

$$\min_{\boldsymbol{\zeta}\in\mathbb{R}^{k\times k}} \rho(\boldsymbol{\zeta}|\mathbf{V}_k) = ||(\mathbf{A}-\mathbf{V}_k\mathbf{T}_k\mathbf{V}_k^T)\mathbf{V}_k||_2 = ||\mathbf{A}-\mathbf{V}_k\mathbf{T}_k\mathbf{V}_k^T||_2$$

shows that $\mathbf{B}_k^\dagger = \mathbf{V}_k\mathbf{T}_k\mathbf{V}_k^T$ is the best rank k approximation of $\mathbf{A}$ in range($\mathbf{V}_k$), the k dimensional Krylov space also spanned by the CG conjugate directions.

To see how the Ritz pairs $\{\theta_i^k, \mathbf{y}^i)\}$ "converge" to the eigenpairs of $\mathbf{A}$, consider that in exact arithmetic when the eigenvalues of $\mathbf{A}$ are distinct, after n iterations the Lanczos algorithm yields $\mathbf{T}_n = \mathbf{V}_n^T\mathbf{A}\mathbf{V}_n$, which shows that $\mathbf{A}$ and $\mathbf{T}_n$ are similar. Thus if $(\theta_i^n, \mathbf{y}^i)$ is an eigenpair of $\mathbf{T}_n$, then $(\lambda_i = \theta_i^n, \mathbf{w}^i = \mathbf{V}_n\mathbf{y}^i)$ is an eigenpair of $\mathbf{A}$. When $k < n$, for any $i \le k$,

$$\theta_i^k \to \lambda_i \quad \text{and} \quad \theta_{-i}^k \to \lambda_{-i}$$

strictly monotonically as $k \to n$ [26].

There is a simple way to monitor convergence of the Ritz pairs. Multiplying equation (11) on the right by $\mathbf{y}^i$ shows that $\rho(\theta_i^k|\mathbf{V}_k\mathbf{y}^i)$ is

$$||\mathbf{A}\mathbf{V}_k\mathbf{y}^i - \theta_i^k\mathbf{V}_k\mathbf{y}^i||_2 = \eta_k|y_k^i|, \tag{23}$$

where y_k^i is the last component of the eigenvector $\mathbf{y}^i$. This shows that $\eta_k|y_k^i| \approx 0$ signals convergence of $(\theta_i^k, \mathbf{V}_k\mathbf{y}^i)$ to $(\lambda_i, \mathbf{w}^i)$ ([26] p. 9; [30] p. 260). In particular, $\eta_k \approx 0$ suggests that many Ritz pairs converge simultaneously, which is observed in practice [26]. Comparing (16) and (23) shows that $||\mathbf{r}^k||_2 = 0$ corresponds to CG finding a solution to $\mathbf{A}\mathbf{x} = \mathbf{b}$ at the same time that all k Ritz pairs converge. On the other hand, it also shows that some Ritz pairs can (and usually do) converge first [26]. The point that we wish to stress here is that by the time CG converges, Ritz pairs have also converged, and so the Ritz vectors span a k dimensional eigenspace of $\mathbf{A}$. In other words, the matrix $\mathbf{B}_k^\dagger$ approximates $\mathbf{A}$ well in these same eigenspaces.

The eigenvalues that are best approximated at iteration k are the extreme ones and the well separated ones [26, 30, 32, 38]. This is because if θ_i^k is closest to the eigenvalue λ^* of $\mathbf{A}$ and the gap,

$$\xi = \min_{\lambda_j \neq \lambda^*} |\theta_i^k - \lambda_j|,$$

between θ_i^k and the other eigenvalues of $\mathbf{A}$ is large, then there is the appealing bound

$$|\theta_i^k - \lambda^*| \le \frac{(\eta_k y_k^i)^2}{\xi}. \tag{24}$$

The eigenvectors corresponding to the eigenvalues which are sufficiently separated are also approximated well,

$$|\sin(\angle(\mathbf{w}^*, \mathbf{V}_k\mathbf{y}^i))| \le \frac{\eta_k|y_k^i|}{\xi}, \tag{25}$$

where $\mathbf{w}^*$ is the eigenvector corresponding to λ^*. If a Ritz value θ_i^k converges to a cluster of eigenvalues λ_j and λ_{j+1} of $\mathbf{A}$, then it is arbitrary which eigenvector is to be approximated by the associated Ritz vector $\mathbf{V}_k\mathbf{y}^i$, corresponding to ξ getting small and the bounds getting large. It turns out that the Ritz vector "splits the difference." This is the same ambiguity which occurs when $\lambda_j = \lambda_{j+1}$ is a repeated eigenvalue of $\mathbf{A}$ [4, 30].

Remark 6. *Theorem 4 and equations (24) and (25) show that* $\mathbf{B}_k^\dagger$ *approximates* $\mathbf{A}$ *(and* $\mathbf{B}_k$ *approximates* $\mathbf{A}^{-1}$*) in the eigenspaces corresponding to the extreme and well separated eigenvalues of* $\mathbf{A}$*.*

6.2. Theoretical bounds for the covariance approximations

In this section, the L_2 norm is used to quantify how the covariance approximation $\mathbf{B}_k$ differs from the desired covariance $\mathbf{A}^{-1}$ when $k < n$. In particular, the norm of the error in covariance estimation, $||\mathbf{A}^{-1}-\mathbf{B}_k||_2$, is at least as large as $1/\lambda_{k+1}$, the largest eigenvalue of $\mathbf{A}^{-1}$ not being estimated by the Lanczos method, and it can get as large as this eigenvalue plus the error in the Lanczos Ritz pairs. Similarly, the norm of the error in covariance estimation, $||\mathbf{A}-\mathbf{B}_k^\dagger||_2$, is at least as large as $\lambda_{-(k+1)}$, the largest eigenvalue of $\mathbf{A}$ not being estimated by the Lanczos method, and it can get as large as this eigenvalue plus the error in the Lanczos Ritz pairs.

Let $\mathbf{\Lambda}$ be a diagonal matrix such that $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, ..., \lambda_n)$. Let $\mathbf{W}_k$ have as columns the k eigenvectors $\{\mathbf{w}^i\}_{i=1}^k$ of $\mathbf{A}$ that correspond to $\lambda_1 \leq ... \leq \lambda_k$, and let $\mathbf{W}_{-k}$ have as columns the k eigenvectors $\{\mathbf{w}^i\}_{i=n-k+1}^n$ of $\mathbf{A}$ that correspond to $\lambda_{-k} \leq ... \leq \lambda_{-1}$. Then the matrix whose columns are all of the eigenvectors of $\mathbf{A}$ can be written as $\mathbf{W} = [\mathbf{W}_k \;\; \mathbf{W}_B] = [\mathbf{W}_{-B} \;\; \mathbf{W}_{-k}]$, and similarly $\mathbf{\Lambda} = [\mathbf{\Lambda}_k \;\; \mathbf{\Lambda}_B] = [\mathbf{\Lambda}_{-B} \;\; \mathbf{\Lambda}_{-k}]$ so that

$$\mathbf{A}^{-1} = \mathbf{W}_k\mathbf{\Lambda}_k^{-1}\mathbf{W}_k^T + \mathbf{W}_B\mathbf{\Lambda}_B^{-1}\mathbf{W}_B^T = \mathbf{W}_{-B}\mathbf{\Lambda}_{-B}^{-1}\mathbf{W}_{-B}^T + \mathbf{W}_{-k}\mathbf{\Lambda}_{-k}^{-1}\mathbf{W}_{-k}^T.$$

By Remark 6, if the well separated eigenvalues of $\mathbf{A}$ are the *small* ones and the rest of the spectrum is relatively large, then

$$\mathbf{B}_k \;\; = \;\; \mathbf{P}_k\mathbf{D}_k^{-1}\mathbf{P}_k^T = \mathbf{V}_k\mathbf{Y}_k\mathbf{\Theta}_k^{-1}\mathbf{Y}_k^T\mathbf{V}_k^T \approx \mathbf{W}_k\mathbf{\Lambda}_k^{-1}\mathbf{W}_k^T,$$

and so the bounds we will give in (26) and (27) below show rigorously when

$$||\mathbf{A}^{-1} - \mathbf{B}_k||_2 = ||\mathbf{P}_B\mathbf{D}_B^{-1}\mathbf{P}_B^T||_2 \approx ||\mathbf{W}_B\mathbf{\Lambda}_B^{-1}\mathbf{W}_B^T||_2$$

is small. Note that equations (26) and (27) do not depend on the conjugacy of $\{\mathbf{p}^i\}$.

Similarly, if the well separated eigenvalues of $\mathbf{A}$ are the *large* ones and the rest of the spectrum is relatively small, then $\mathbf{B}_k \approx \mathbf{W}_{-k}\mathbf{\Lambda}_{-k}^{-1}\mathbf{W}_{-k}^T$, and now

$$\mathbf{B}_k^\dagger \;\; = \;\; \mathbf{V}_k\mathbf{T}_k\mathbf{V}_k^T \approx \mathbf{W}_{-k}\mathbf{\Lambda}_{-k}\mathbf{W}_{-k}^T.$$

The bounds in (28) and (29) below show rigorously when

$$||\mathbf{A} - \mathbf{B}_k^\dagger||_2 \approx ||\mathbf{W}_{-B}\mathbf{\Lambda}_{-B}\mathbf{W}_{-B}^T||_2$$

is small, (28) and (29) do not depend on conjugacy being maintained.

6.2.1. Theoretical bounds for $||\mathbf{A}^{-1} - \mathbf{B}_k||_2$ A lower bound on the covariance error is given by an application of Weyl's Theorem [30],

$$\frac{1}{\lambda_{k+1}} \leq ||\mathbf{A}^{-1} - \mathbf{B}_k||_2 \tag{26}$$

which is a lower bound of $||\mathbf{A}^{-1} - \mathbf{M}||$ for any k-rank $n \times n$ matrix M. Another application of Weyl's Theorem shows that the largest "harmonic Ritz error" [27, 29, 37] also gives a lower bound

$$\max_{1\leq i\leq k}\left(\frac{1}{\lambda_i} - \frac{1}{\theta_i^k}\right) \leq ||\mathbf{A}^{-1} - \mathbf{B}_k||_2,$$

which is tight in many numerical examples.

An upper bound is found by

$$\begin{aligned} ||\mathbf{A}^{-1} - \mathbf{B}_k||_2 &\leq ||\mathbf{A}^{-1} - \mathbf{W}_k\mathbf{\Lambda}_k^{-1}\mathbf{W}_k^T||_2 + ||\mathbf{W}_k\mathbf{\Lambda}_k^{-1}\mathbf{W}_k^T - \mathbf{B}_k||_2 \\ &= \frac{1}{\lambda_{k+1}} + ||\mathbf{W}_k\mathbf{\Lambda}_k^{-1}\mathbf{W}_k^T - \mathbf{V}_k\mathbf{T}_k^{-1}\mathbf{V}_k^T||_2. \end{aligned} \tag{27}$$

This and (26) show that the norm of the error in covariance estimation is at least as large as the largest eigenvalue not being estimated by the Lanczos method, and it can get as large as this eigenvalue plus the error in the Lanczos Ritz pairs.

6.2.2. Theoretical bounds for $||\mathbf{A} - \mathbf{B}_k^\dagger||_2$ Weyl's Theorem gives the lower bounds

$$\lambda_{-(k+1)} \leq ||\mathbf{A} - \mathbf{B}_k^\dagger||_2. \tag{28}$$

and

$$\max_{1\leq i\leq k} \left(\lambda_{-i} - \theta_{-i}^k\right) \leq ||\mathbf{A} - \mathbf{B}_k^\dagger||_2$$

The triangle inequality gives the upper bound

$$||\mathbf{A} - \mathbf{B}_k^\dagger||_2 \leq \lambda_{-(k+1)} + ||\mathbf{W}_{-k}\mathbf{\Lambda}_{-k}\mathbf{W}_{-k}^T - \mathbf{V}_k\mathbf{Y}\mathbf{\Theta}\mathbf{Y}_k^T\mathbf{V}_k^T||_2. \tag{29}$$

This and (28) show that the norm of the error in covariance estimation is at least as large as the largest eigenvalue not being estimated by the Lanczos method, and it can get as large as this eigenvalue plus the error in the Lanczos estimates.

6.3. The effect of finite precision

CG is remarkably robust in finite precision. In fact, if the eigenvalues of $\mathbf{A}$ are in k distinct clusters, then CG tends to find an approximate solution to $\mathbf{Ax} = \mathbf{b}$ after only k iterations [28, 38]. As long as "local orthogonality" is maintained, then, unless $\mathbf{A}$ has eigenvalues on the order of machine precision or has a large condition number, convergence $\mathbf{x}_{CG}^k \to \mathbf{A}^{-1}\mathbf{b}$ and a small residual $\mathbf{r}^k$ is guaranteed [26].

For a Lanczos eigensolver in finite precision, when the i^{th} Ritz pair $(\theta_i^k, \mathbf{V}_k\mathbf{y}^i)$ converges at iteration k, then (23) shows that $\eta_k|\mathbf{y}_k^i| \approx 0$. Unfortunately,

$$\mathbf{v}^{kT}\mathbf{V}_k\mathbf{y}^i \leq \frac{\epsilon||\mathbf{A}||}{\eta_k|\mathbf{y}_k^i|} \tag{30}$$

is also true [15, 26, 30], where ϵ is machine precision. Thus, in the face of finite precision, the newest Lanczos vector $\mathbf{v}^k$ loses orthogonality with the others when some Ritz pair has converged to an eigenvalue of $\mathbf{A}$, and the unwanted component of $\mathbf{v}^k$ is in the direction of the converged Ritz vector $\mathbf{V}_k\mathbf{y}^i$. Now equation (14) explains why CG experiences loss of orthogonality of the residuals and a corresponding loss of conjugacy in the search directions. Equations (23), (16) and (30) show that loss of conjugacy can happen at the same time as CG converges, but it usually happens before. The upside is that, by the time CG converges, Lanczos eigenpair estimates have already converged, and so $\mathbf{B}_k$ is a good approximation to $\mathbf{A}^{-1}$ in the corresponding eigenspaces (Theorem 4).

As iterations continue past convergence of some of the Lanczos Ritz pairs and the corresponding loss of orthogonality, "ghost eigenvalues" of $\mathbf{T}_k$ appear [8, 15]. These new eigenvalues of $\mathbf{T}_k$ estimate eigenvalues of $\mathbf{A}$ which have already been estimated by earlier Ritz values. In other words, $\mathbf{T}_k$ has clustered eigenvalues near an isolated eigenvalue of $\mathbf{A}$. Equation (30) partially explains this phenomenon, showing that the newer Ritz vectors leak back into eigenspaces spanned by the previous Ritz vectors. Meurant [26] calls into question the validity of considering the Ritz pairs after loss of orthogonality, in part since there is no proof of the "Lanczos Phenomenon" [8], which hopes that for each distinct eigenvalue of $\mathbf{A}$, there exists a k such that $\mathbf{T}_k$ has the same eigenvalue. Using CG somewhat alleviates this issue, since CG stops when the residual gets too small.

REFERENCES

1. Auvinen H, Bardsley JM, Haario H, Kauranne T. Large-scale Kalman filtering using the limited memory BFGS method. *Electronic Transactions in Numerical Analysis* 2010; **35**(9), 217-233.
2. Auvinen H, Bardsley JM, Haario H, Kauranne T. The variational Kalman filter and an efficient implementation using limited memory BFGS. *International Journal for Numerical Methods in Fluids* 2010; published online, DOI: 10.1002/fld.2153.
3. Axelsson O. *Iterative Solution Methods*. Cambridge University Press, 1996.
4. Bai Z, Demmel J, Ruhe A, van der Vorst H. *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*. SIAM, 2000.
5. Brandts J, van der Vorst H. The Convergence of Krylov methods and Ritz values. In *Conjugate gradient algorithms and finite element methods* Krizek M, Neittaanmaki P, Glowinski R, Korotov S (eds.). Springer, 2004; 47–68.
6. Cane MA, Miller RN, Tang B, Hackert EC, Busalacchi AJ. Mapping tropical Pacific sea level: data assimilation via reduced state Kalman filter. *J. Geophys. Res.* 1996; **101**, 599-617.
7. Cao Y, Zhu J, Navon IM, Luo Z. A reduced order approach to four-dimensional variational data assimilation using proper orthogonal decomposition. *Int. J. Numerical Methods in Fluids* 2007; **53**(10), 1571-1583.
8. Cullum JK, Willoughby RA. *Lanczos Algorithms for Large Symmetric Eigenvalue Computations*. Birkhauser, 1985.
9. Dee DP. Simplification of the Kalman filter for meteorological data assimilation. *Quart. J. Roy. Meteor. Soc.* 1990; **117**, 365-384.
10. Fisher M. Minimization algorithms for variational data assimilation. *Proceedings of the ECMWF Seminar on Recent Developments in Numerical Methods for Atmospheric Modelling, Reading, England* 7-11 September 1998; 364-385.
11. Fisher M, Andersson E. Developments in 4D-Var and Kalman filtering. *ECMWF Technical Memorandum 347, European Center for Medium Range Weather Forecasting* 2001.
12. Fisher M, Leutbecher M, Kelley G. On the equivalence between Kalman smoothing and weak-constraint four-dimensional variational data assimilation. *Q. J. R. Meterol. Soc.* 2005; **131**, 3235-3246.
13. Fisher M, Nocedal J, Trémolet Y, Wright SJ. Data assimilation in weather forecasting: a case study in PDE-constrained optimization. *Optim. Eng.* 2009; **10**(3), 1389-4420.
14. Gejadze IY, Le Dimet FX, Shutyaev V. On analysis error covariances in variational data assimilation. *SIAM J. Sci. Comput.* 2008; **30**, 1847-1874.
15. Golub GH, Van Loan CF. *Matrix Computations, 3rd edition*. The Johns Hopkins University Press, 1996.
16. Hestenes MR, Stiefel E. Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Standards* 1952; **49**, 409-436.
17. Kalman RE. A new approach to linear filtering and prediction problems. *Transactions of the ASME – Journal of Basic Engineering (Series D)* 1960; **82**, 35-45.
18. Lanczos C. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Res. Nat. Bur. Standards* 1950; **45**, 255-282.
19. Lanczos C. Solutions of linear equations by minimized iterations. *J. Res. Nat. Bur. Standards* 1952; **49**, 33-53.
20. LeDimet FX, Talagrand O. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus A series,* 1986; **38**, 97-110.

21. Leutbecher M. A data assimilation tutorial based on the Lorenz-95 system. *European Centre for Medium-Range Weather Forecasts Web Tutorial*. www.ecmwf.int/newsevents/training/lecture_notes/pdf_files/ASSIM/Tutorial.pdf.
22. Li Z, Navon M. Optimality of variational data assimilation and its relationship with the Kalman filter and smoother. *Q. J. R. Meteorol. Soc.* 2008; **127**, 661-683.
23. Lorenc AC. Modelling of error covariances by 4D-Var data assimilation. *Q. J. R. Meterol. Soc.* 2003; **129**, 3167-3182.
24. Lorenz EN. Predictability: a problem partly solved. *Proc. Seminar on Predicability, European Center for Medium Range Weather Forecasting, Reading, Berkshire, UK* 1996; **1**, 1-18.
25. Lorenz EN, Emanuel KA. Optimal sites for supplementary weather observations: simulation with a small model. *Journal of Atmospheric Science* 1998; 399-414.
26. Meurant G. *The Lanzcos and conjugate gradient algorithms*. SIAM, Philadelphia, 2006.
27. Morgan RB. Computing Interior Eigenvalues of Large Matrices. *Linear Algebra and Its Applications* 1991; **154-156**, 289-309.
28. Nocedal J, Wright S, *Numerical Optimization*, Springer 1999.
29. Paige C, Parlett BN, van der Vosrst HA. Approximate solutions and eigenvalue bounds from krylov subspaces. *Numerical Linear Algebra with Applications* 1995; **2**(2), 115-133.
30. Parlett BN. *Symmetric Eigenvalue Problem*, Prentice Hall, 2980.
31. Rabier F, Järvinen H, Klinker E, Mahfouf JF, Simmons A. The ECMWF operational implementation of four-dimensional variational assimilation. Part I: experimental results with simplified physics. *Q. J. R. Meteorol. Soc.* 2000; **126**, 1143-1170.
32. Saad Y. *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, 1992.
33. Scales JA. On the use of conjugate gradient to calculate the eigenvalues and singular values of large, sparse matrices. *Geophysical Journal* 1989; **97**, 179-183.
34. Schneider MK, Willsky AS. Krylov subspace estimation. *SIAM J. Sci. Comput.* 2000; **22**(5), 1840-1864.
35. Schneider MK, Willsky AS. A Krylov subspace method for covariance approximation and simulation of random processes and fields. *Multidimensional Syst. Signal Process.* 2003; **14**(4), 295–318.
36. Schneider MK, Willsky AS. Krylov subspace algorithms for space-time oceanography data assimilation. *IEEE International Geoscience and Remote Sensing Symposium* 2000; **2**, 727-729.
37. Sleijpen G, van der Eshof J. Accurate approximations to eigenpairs using the harmonic Rayleigh-Ritz method. *Preprint 1184, Department of Mathematics, University of Utrecht*, April 2001.
38. Sleijpen GLC, Van Der Sluis A. Further results on the convergence behavior of conjugate-gradients and Ritz values. *Linear Algebra and Its Applications* 1996; **246**, 233-278.
39. Tian X, Xie Z, Dai A. An ensemble-based explicit four-dimensional variational assimilation method. *J. Geophys. Res.* 2008; **113**(D21124), 1-13.
40. Tshimanga J, Gratton S, Weaver AT, Sartenaer A. Limited-memory preconditioners, with application to incremental four-dimensional variational data assimilation. *Q. J. R. Meteorol. Soc.* 2008; **134**, 751-769.
41. Veersé F, Auroux D, Fisher M. Limited-memory BFGS diagonal predonditioners for a data assimilation problem in meteorology. *Optimization and Engineering* 2000; **1**, 323-339.
42. Welch G, Bishop G. An introduction to the Kalman filter. *Technical Report 95-041, University of North Carolina at Chapel Hill, Department of Computer Science*, 1995.
43. Yang W, Navon IM, Courtier P. A new Hessian preconditioning method applied to variational data assimilation experiments using NASA general circulation models. *Monthly Weather Rev.* 1996; **124**, 1000-1017.