Inexact range-space Krylov solvers for linear systems arising from inverse problems

by S. Gratton[1], Ph. L. Toint[2] and J. Tshimanga Ilunga[3]

Report 09/20 16 December 2009

[1] ENSEEIHT,
2, rue Camichel, 31000 Toulouse, France.
Email: serge.gratton@enseeiht.fr

[2] Department of Mathematics,
FUNDP-University of Namur,
61, rue de Bruxelles, B-5000 Namur, Belgium.
Email: philippe.toint@fundp.ac.be

[3] CERFACS,
42, avenue Coriolis, 31057 Toulouse, France.
Email: ilungatshimanga@yahoo.fr

Inexact range-space Krylov solvers for linear systems arising from inverse problems

S. Gratton Ph. L. Toint J. Tshimanga Ilunga

16 December 2009

Abstract

The object of this paper is twofold. Firstly, range-space variants of standard Krylov iterative solvers are introduced for unsymmetric and symmetric linear systems. These are characterized by possibly much lower storage and computational costs than their full-space counterparts, which is crucial in data assimilation applications and other inverse problems. Secondly, it is shown that the computational cost may be further reduced by using inexact matrix-vector products: formal error bounds are derived on the size of the residuals obtained under two different accuracy models, and it is shown why a model controlling forward error on the product result is often preferable to one controlling backward error on the operator. Numerical examples finally illustrate the developed concepts and methods.

Keywords: Krylov methods, linear systems, inexact matrix products, data assimilation.

1 Introduction and motivation

Inverse problems in the natural sciences and elsewhere often give rise to very large under-determined parameter identification problems. In these problems, one typically tries to identify system parameters by fitting a model's output to a number of observations which is very often much smaller than that of the parameters. The resulting under-determined fitting problem is then regularized by selecting parameter sets that are as close as possible to values known from a previous case study (see Gratton, Lawless and Nichols, 2007). As we discuss below, problems of this type often lead, possibly after preconditioning, to solving variational formulations for which one considers the iterative solution of (potentially very large) linear systems of the form

$$(\gamma I_n + K^T L)s = b, \tag{1.1}$$

where both K and L are $m \times n$ matrices (possibly identical) with $m \ll n$, I_n is the identity matrix of size n and b is a general right-hand side. The objective of this paper is to propose and analyze iterative methods whose storage requirements and linear algebra costs (beyond that of the products of vectors by K^T and L) are essentially dependent on m (at variance with standard approaches where they vary with n), and which produce monotonically decreasing values of the underlying variational objective. Moreover, their computational cost is further reduced by considering that the product of a vector by K^T or L may be only approximate (see Simoncini and Szyld, 2003, van den Eshof and Sleijpen, 2004, van den Eshof, Sleijpen and van Gijzen, 2005).

The motivation for investigating problem (1.1) and methods of the type we just mentioned finds its origin in (but is not limited by) the data assimilation problem for oceanography and weather forecasting (see Rabier, 2005, for instance). In this important application, the model is that of a discretized Navier-Stokes formulation (or a variant thereof) for describing the ocean or atmosphere and the observations are given by temperature,

current, pressure and altimetry measurements collected from satellites, ocean-bound or ground stations. The most commonly studied problem is then to determine a complete set of initial conditions from which the Navier-Stokes model can be integrated to provide predictions, with the difficulty that they are much less observations (despite today's large data collection capabilities) than variables to determine in this initial condition, thereby suggesting regularization techniques. In daily practice (2009), the number of such variables is of the order of 10^8 while the number of available observations is only of the order of 10^5 (see Rabier, 2005 again), and the ratio of the first to the second is furthermore expected to increase significantly with the advent of more complex procedures (see Trémolet, 2006*a*, 2006*b*). The previously known state from which deviation is minimized (for the purpose of regularization) is called the background state and often selected as the values of the unknown parameters at the current time derived from a model validated in the past. As explained in Gratton et al. (2007), one is then interested in solving the nonlinear least-squares problem given by

$$\min_x \tfrac{1}{2}(x-x_b)^T B^{-1}(x-x_b) + \tfrac{1}{2}\sum_{j=0}^{p} \left\{\mathcal{H}_j[x(t_j)] - y_j^o\right\}^T R_j^{-1} \left\{\mathcal{H}_j[x(t_j)] - y_j^o\right\}, \qquad (1.2)$$

where x_b is the background state, $\mathcal{H}_j$ is the operator modelling the observed quantities at time t_j y_j^o is the vector of observations, and $x(t_j)$ the state at this time, and the matrices B and R represent correlations between the background variables and the observations, respectively. Problem (1.2) is typically solved by a standard truncated Gauss-Newton method (known in the oceanography and weather-forecasting communities under the name of incremental 4D-Var (Courtier, Thépaut and Hollingsworth, 1994), an acronym for four dimensional variational assimilation). The linearized subproblem arising at iteration k of this Gauss-Newton procedure, where we have concatenated the observations and model predictions over time into a single vector (see Gratton and Tshimanga, 2009 for details) is then given by

$$\min_{s\in\mathbb{R}^n} \tfrac{1}{2}(x_k+s-x_b)^T B^{-1}(x_k+s-x_b) + \tfrac{1}{2}(Hs-d)^T(Hs-d), \qquad (1.3)$$

where x_k, x_b and s belong to $\mathbb{R}^n$ with n being the dimension of the unknown initial state, B is a positive-definite $n\times n$ symmetric matrix, $d\in\mathbb{R}^m$ is the concatenated misfit vector at x_k multiplied by the square root of the inverse of R, where $R=\mathrm{diag}(R_0,\ldots,R_p)$ is $m\times m$ and symmetric positive-definite[(1)], and H is a $m\times n$ matrix representing the concatenated linearized model also multiplied by the square root of the inverse of R. The minimization of the convex quadratic given by (1.3) would seem a reasonably well-mastered problem, if it were not for the sizes involved (recall that we aim at $n\approx 10^9$ and $m\approx 10^5$). As for smaller problems, today's standard method for the solution of (1.3) is to apply a conjugate-gradient algorithm, which may be viewed as an iterative solver for the system

$$(B^{-1}+H^TH)s = H^Td + B^{-1}(x_b-x_k). \qquad (1.4)$$

As is typical for conjugate-gradient method, preconditioning is crucial for numerical performance, and we consider here two distinct possibilities. The first technique, which is closer to current practice, is two-sided and symmetric, and gives the system

$$(I_n + B^{1/2}H^THB^{1/2})z = B^{1/2}H^Td + B^{-1/2}(x_b-x_k) \;\text{ where }\; s=B^{1/2}z. \qquad (1.5)$$

Interestingly, the cost of the products with H is so high in practical applications that every effort is made to reduce the number of such products beyond the effect of preconditioning, including the use of reorthogonalization within the considered iterative solvers, which often implies large storage requirements. As a result, the interest of maintaining a purely

(1) A more elaborate way to handle the correlation matrix R is of course used in practice, but we choose this implicit formulation here for the sake of simplicity.

symmetric formulation becomes less clear. This makes a second preconditiong technique attractive, where one uses a simple right-sided product by B, yielding the system

$$(I_n + H^T H B^{-1})z = H^T d + B^{-1}(x_b - x_k) \text{ where } s = Bz. \tag{1.6}$$

This approach furthermore opens the door to simpler techniques for handling the observation correlation matrix R than that suggested above.

It is now clear that both (1.6) and (1.5) *are examples of systems of the form (1.1)*, and that their practical solution by efficient iterative methods correspond to the declared objective of this paper. It should be stressed however that the numerically efficient algorithms for the solution of (1.1) are of interest much more broadly, as this framework covers a large class of (mostly inverse) problems considered in a variational setting and where the "action" is limited to a small subspace of the original formulation. The standard Tikhonov regularization scheme for rank-deficient or ill-posed problems also leads to systems of the form (1.1) (see Hansen, 1997).

As can be seen from the minimization context introduced, it is also highly desirable for the design of realistic termination criteria that iterates produced by the solution algorithms ensure an iteration-wise decrease of the underlying objective function in (1.3), in the symmetric case, or of the associated residual norm otherwise. Unfortunately, a simple solution consisting in using the Sherman-Morrison-Woodbury formula (see Conn, Gould and Toint, 2000, page 57, for instance) on the linear system and then applying an Krylov-space iterative method on the resulting $m \times m$ linear system (or variants thereof such as the PSAS method (Amodei, 1995, Da Silva, Pfaendtner, Guo, Sienkiewicz and Cohn, 1995, Cohn, Da Silva, Guo, Sienkiewicz and Lamich, 1998) or the equivalent "representer" technique of Bennett and Thornburn, 1992) may often be problematic (see El Akkroui, Gauthier, Pellerin and Buis, 2008, or Gratton and Tshimanga, 2009).

Our developments originate in the recent proposal by Gratton and Tshimanga (2009), where a variant of the conjugate-gradient method, called RPCG, was proposed for the exact symmetric case, i.e. the case where $L = K$ and the products by K are carried out exactly. As for the variants discussed below, RPCG's storage requirement and linear algebra costs (beyond that of the products by K and K^T) vary with m rather than n. However, and although initial numerical experiments were promising, no stability analysis was provided for this earlier method, that would cope with errors generated by computer arithmetic or by inexact matrix products. This is especially crucial since the geometric foundation of the method is to restrict operations to the *range space* of K^T, and perturbations of this matrix could therefore potentially generate significant instability. We show below that this unfortunate behaviour may indeed appear in specific cases, but we also show that the new variants proposed in this paper are, by contrast, stable for such perturbations.

The paper is organized as follows. The new range-space Krylov methods are derived for the symmetric and unsymmetric cases in Section 2. The effect of inexact matrix products on the convergence of these new algorithms is then investigated in Section 3, and the concepts are numerically illustrated in Section 4. Some conclusions and perspectives are finally presented in Section 5.

2 Range-space Krylov methods

We start by considering the standard GMRES (Saad and Schultz, 1986) method for the system (1.1) for a possibly rank deficient matrix

$$A = \gamma I_n + K^T L.$$

Although this method is well-known, we briefly review its main features and concepts to establish a basis for later developments. The main idea of the method is to minimize the Euclidean norm of residual $(\gamma I_n + K^T L)s - b$ on the successive nested Krylov subspaces

generated by the sequence

$$b, \;\; (\gamma I_n + K^T L)b, \;\; (\gamma I_n + K^T L)^2 b, \;\; (\gamma I_n + K^T L)^3 b, \;\; \ldots \tag{2.1}$$

or, equivalently, by

$$b, \;\; (K^T L)b, \;\; (K^T L)^2 b, \;\; (K^T L)^3 b, \;\; \ldots \tag{2.2}$$

This is achieved by using the Arnoldi process (see Saad, 1996, page 154, or Kelley, 1995, page 37) to generate an orthonormal basis of each of the these subspaces, i.e. a set of vectors $\{v_i\}_{i=1}^{k+1}$ with $v_1 = b/\|b\|$ (where $\|\cdot\|$ stands for the Euclidean norm) such that, after k steps,

$$K^T L \, V_k = V_{k+1} H_k, \tag{2.3}$$

where the columns of $V_k \stackrel{\rm def}{=} [v_1, \ldots, v_k]$ form an orthonormal basis of the k-th Krylov subspace

$$\mathcal{K}_k = \mathrm{span}[\, b, \ldots, (K^T L)^{k-1} b \,],$$

and where H_k is a $(k+1) \times k$ upper-Hessenberg matrix. The linear least-squares

$$\min_y \|H_k y - \beta_1 e_1\| \tag{2.4}$$

(where the symbol e_i denotes the i-th vector of the canonical basis) is then explicitly solved (in $\mathcal{K}_k$, which is the subspace spanned by the columns of V_k), yielding the GMRES[(2)] algorithm on this page.

Algorithm 2.1: $s =$ **GMRES(** K**,** L**,** b **)**

1. Define $\beta_1 = \|b\|$ and $v_1 = b/\beta_1$.
2. For $k = 1, \ldots, m$,
   - (a) $w_k = K^T L v_k$
   - (b) for $i = 1, \ldots, k$,
     - i. $H_{i,k} = v_i^T w_k$
     - ii. $w_k \leftarrow w_k - H_{i,k} v_i$
   - (c) $H_{k,k} \leftarrow H_{k,k} + \gamma$,
   - (d) $\beta_{k+1} = H_{k+1,k} = \|w_k\|$,
   - (e) $v_{k+1} = w_k/\beta_{k+1}$,
   - (f) $y_k = \arg\min_y \|Hy - \beta_1 e_1\|$,
   - (g) if $\|Hy_k - \beta_1 e_1\| < \epsilon_r$, break.
3. Return $s = V_k y_k$.

In this statement of GMRES, one recognizes the construction of the Krylov sequence in Steps 1 and 2(a), its orthogonalization in Steps 2(b,d,e), the shift corresponding to the equivalence between (2.1) and (2.2) in Step 2(c) and the solution of the restricted linear least-squares (2.4) in Step 2(f). The convergence test of Step 2(g) is there to detect early termination, but one knows that the Krylov residual $q_k = Hy_k - \beta_1 e_1$ must decrease in norm at every iteration since q_k is nothing but the representation in $\mathcal{K}_k$ of the projection of the right-hand side onto the orthogonal to $\mathcal{K}_k$. Note that the k-th column of upper-Hessenberg matrix H is constructed at iteration k, and this allows us to drop the index

[(2)] Not to be confused with GMRES(m), a version which is restarted every m iterations.

of H indicating the iteration number from now on, since it can easily be recovered from the number of columns. Note that the matrix V_k must be stored for Steps 2(b)i and 3.

If b does not belong to the range of A, there is no reason for q_k to converge to zero. Moreover it may happen in this case that "breakdown" occurs in the very unlikely circumstance where $A^k b$ turns out to be, for some k, an exact linear combination of $\{A^i b\}_{i=1}^{k-1}$, and the vector w_k may then be identically zero at the end of Step 2(b), thereby yielding $\beta_{k+1} = 0$ and making Step 2(e) undefined. The most obvious strategy to cope with this (mostly theoretical) situation is a simple restart of the process with a slightly perturbed right-hand side. Another interesting possibility is described in Reichel and Ye (2005).

Importantly, variants of the GMRES algorithm can be derived easily. First, the solution of a linear least-squares in $\mathcal{K}_k$ of Step 2(f) may be replaced by the solution of the linear system

$$H^{\square} y_k = \beta_1 e_1 \tag{2.5}$$

where $e_1 \in \mathbb{R}^k$ and $H^{\square}$ is the leading $k \times k$ submatrix of H, in which case one replaces the condition tested in Step 2(g) by $\|H^{\square} y_k - \beta_1 e_1\| \leq \epsilon_r$. This gives a method known under the acronym of FOM (for Full Orthogonalization Method). We may also consider the symmetric case where $L = K^T$, in which case it is possible to restrict the loop of Step 2(b) to $i = \max[1, k-1]$ and $i = k$, resulting in a tridiagonal H. If y_k is derived from Step 2(f), this generates the same iterates as the MINRES (Minimum Residuals) method, while we obtain the well-known Conjugate-Gradient (CG) algorithm if (2.5) is used instead. All these variants preserve some important monotonicity properties: the norms of the Krylov residuals q_k for GMRES and MINRES and the values of the implied quadratic form

$$\tfrac{1}{2} y_k^T V_k^T (\gamma I + K^T K) V_k y_k - b^T V_k y_k \tag{2.6}$$

for FOM and CG are both decreasing (in exact arithmetic) as the iterations proceed. Unfortunately, this is not the case for the norms of the full-space residuals

$$r_k = (\gamma I + K^T K) V_k y_k - b \tag{2.7}$$

in the symmetric case case, as $\|r_k\|$ may exhibit a distinct nonmonotonic behaviour. Obviously, the comments we made regarding breakdown in GMRES also apply to its variants, but there is the additional (and equally unlikely) possibility for a breakdown in FOM where $\beta_{k+1} \neq 0$ but $H^{\square}$ turns out to be exactly singular. Fortunately, this last difficulty may be resolved by replacing the problematic FOM iteration by a (well-defined) GMRES iteration without stopping the process.

The next step follows Gratton and Tshimanga (2009) and assumes (for now) that

$$b \in \text{range}(K^T). \tag{2.8}$$

We now observe that, in this case, the sequence (2.2) may be rewritten as

$$K^T d, \;\; K^T (L K^T) d, \;\; K^T (L K^T)^2 d, \;\; K^T (L K^T)^3 d, \;\; \ldots \tag{2.9}$$

for some vector $d \in \mathbb{R}^m$, and thus deduce that the Krylov spaces in $\mathbb{R}^n$ associated with this sequence are the images by K^T of other Krylov spaces generated now in $\mathbb{R}^m$ by the sequence

$$d, \;\; (L K^T) d, \;\; (L K^T)^2 d, \;\; (L K^T)^3 d, \;\; \ldots \tag{2.10}$$

We may therefore consider an Arnoldi process based on this sequence, leading now to the relation

$$L K^T \hat{V}_k \stackrel{\text{def}}{=} L K^T \left[\; \hat{v}_1, \quad \ldots, \quad \hat{v}_k \; \right] = \hat{V}_{k+1} H \tag{2.11}$$

instead of (2.3), where we use the $\hat{x}$ notation to denote the pre-image by K^T in $\mathbb{R}^m$ of $x \in \text{range}(K^T) \subseteq \mathbb{R}^n$, i.e. $x = K^T \hat{x}$. If we now rewrite the complete GMRES algorithm in (the much smaller) $\mathbb{R}^m$, we then obtain Algorithm RSGMR0[(3)] on the next page.

[(3)] For Range-Space GMRes.

Algorithm 2.2: $s =$ **RSGMR0(** K**,** L**,** d **)**

1. Define $p_1 = K^T d, \quad \hat{z}_1 = K p_1,$
2. Set $\beta_1 = \sqrt{d^T \hat{z}_1}, \quad \hat{v}_1 = d/\beta_1 \quad \hat{z}_1 \leftarrow \hat{z}_1/\beta_1$ and $p_1 \leftarrow p_0/\beta_1$.
3. For $k = 1, \ldots, m,$
   (a) $\hat{w}_k = L p_k$
   (b) for $i = 1, \ldots, k,$
      i. $H_{i,k} = \hat{z}_i^T \hat{w}_k$
      ii. $\hat{w}_k \leftarrow \hat{w}_k - H_{i,k} \hat{v}_i$
   (c) $H_{k,k} \leftarrow H_{k,k} + \gamma,$
   (d) $p_{k+1} = K^T \hat{w}_k, \quad \hat{z}_{k+1} = K p_k, \quad \beta_{k+1} = H_{k+1,k} = \sqrt{\hat{z}_{k+1}^T \hat{w}_k},$
   (e) $\hat{v}_{k+1} \leftarrow \hat{w}_k/\beta_{k+1}, \quad \hat{z}_{k+1} \leftarrow \hat{z}_k/\beta_{k+1,}, \quad p_{k+1} \leftarrow p_k/H_{k+1,k},$
   (f) $y_k = \arg\min_y \|Hy - \beta_1 e_1\|,$
   (g) if $\|Hy_k - \beta_1 e_1\| < \epsilon_r,$ break.
4. Return $s = K^T \hat{V}_k y_k$.

Observe that we had to compute the product of p_k by K, in order to evaluate the norms (in Steps 1 and 2(d)) and the inner products (in Step 2(b.i)) in the correct metric (i.e. in $\mathbb{R}^n$). Thus we have replaced computation and storage of n-dimensional vectors by that of m-dimensional ones for the price of an additional product by K at each iteration, with the exception of p_k which remains a "large" n-dimensional vector. RSGMR0 also requires the storage of the $\hat{v}_i$ and the $\hat{z}_i$, but these are now of dimension m. Hence our comment in the introduction saying that work and storage depend essentially on m. Although the need for an additional product does not sound ideal, it may make the difference between an impractical method (where a collection of huge vectors is just too large for the computer at hand) and a more CPU-intensive but practical one.

It is now easy again to obtain range-space variants of FOM, MINRES and CG, now denoted by the acronyms RSFOM, RSMR and RSCG, exactly in the same manner as that indicated for the full-space variants. Obviously, the additional product by K in Step 3(d) is redundant with the product by L whenever $K = L$. In fact RSCG is identical (in exact arithmetic) to the RPCG method of Gratton and Tshimanga (2009).

Observe finally that, since GMRES and RSGMR0 are mathematically equivalent, the sequence $\|q_k\|$ generated by RSGMR0 is identical to that generated by GMRES for $b = K^T d$ and therefore enjoys the same monotonicity property. The same argument ensures that the values of (2.6) generated by RSFOM and RSCG (RPCG) are monotonically decreasing since they are the same as those generated by FOM and CG.

The reader interested in the solution of symmetric problems ($L = K$) may wonder at this point at the cost of storing an orthonormal basis of the successive Krylov spaces in RSFOM, compared to the lower requirements of RSCG and RSMR. While we indicate in Section 4 that the FOM setting provides better accuracy and is much better suited to the use of inexact matrix-vector products (discussed in Section 3), we also note that this cost is much lower for range-space methods than for full-space ones. Indeed, the Krylov vectors now belong to $\mathbb{R}^m$ rather than $\mathbb{R}^n$, which may make a considerable difference in practical applications where $m \ll n$.

While we have derived Algorithms RSGMR0 under the condition (2.8), it is not difficult to avoid this assumption and allow for a right-hand side which is not the range of K^T.

This simply achieved by considering the extended system

$$(\gamma I + \overline{K}^T \overline{L})s = \overline{K}^T e_{m+1}, \quad \text{where } \overline{K} = \begin{bmatrix} K \\ b^T \end{bmatrix} \text{ and } \overline{L} = \begin{bmatrix} L \\ 0^T \end{bmatrix}, \tag{2.12}$$

since then

$$\overline{K}^T \overline{L} = K^T L \quad \text{and} \quad \overline{K}^T e_{m+1} = b. \tag{2.13}$$

It may however be inconvenient to augment K and L as in (2.12) explicitly, mainly because the product by these matrices typically result from applying some potentially complicated operator to a vector. We therefore need a version of the algorithm based on K and L rather than $\overline{K}$ and $\overline{L}$. After some tedious but elementary rewriting, in particular using the identity

$$b^T p_k = b^T [\, K^T b \,] \, \hat{w}_k = [\, \hat{u}^T \, \|b\|^2 \,] \, \hat{w}_k,$$

for $\hat{u} = Kb$, one may then reformulate the RSGMR0 algorithm in its final, more general form (dubbed RSGMR) on this page.

Algorithm 2.3: $s =$ **RSGMR(** K**,** L**,** b **)**

1. Define $\beta_1 = \|b\|, \quad p_1 = b, \quad \hat{u} = Kb, \quad \hat{z}_1 = \hat{u}/\beta_1,$ and $\hat{v}_1 = e_{m+1}/\beta_1$.
2. For $k = 1, \ldots, m+1$,
   - (a) $\hat{w}_k^T = [(Lp_k)^T \, 0], \quad \hat{w}_k \leftarrow \hat{w}_k/\beta_k,$
   - (b) for $i = 1, \ldots, k$,
     - i. $H_{i,k} = [\hat{z}_i^T \, 0] \, \hat{w}_k$
     - ii. $\hat{w}_k \leftarrow \hat{w}_k - H_{i,k}\hat{v}_i$
   - (c) $H_{k,k} \leftarrow H_{k,k} + \gamma,$
   - (d) $p_{k+1} = [K^T b] \, \hat{w}_k, \quad \hat{z}_{k+1} = Kp_{k+1}, \quad \zeta_{k+1} = [\hat{u}^T \, \beta_1^2] \, \hat{w}_k,$
   - (e) $\beta_{k+1} = H_{k+1,k} = \sqrt{[\hat{z}_{k+1}^T \; \zeta_{k+1}] \, \hat{w}_k},$
   - (f) $\hat{v}_{k+1} \leftarrow \hat{w}_k/\beta_{k+1}, \quad \hat{z}_{k+1} \leftarrow \hat{z}_k/\beta_{k+1},$
   - (g) $y_k = \arg\min_y \|Hy - \beta_1 e_1\|,$
   - (h) if $\|Hy_k - \beta_1 e_1\| < \epsilon_r$, break.
3. Return $s = [K^T b]\hat{V}_k y_k$.

In this last formulation, p_k is the only n-dimensional vector, $\hat{w}_k$ and the $\hat{v}_i$ are $(m+1)$-dimensional, u and the $\hat{z}_i$ are m-dimensional and y_k is k-dimensional. We may also see from (2.13) that the Krylov subspaces generated in $\mathbb{R}^n$ by RSGMR are identical to those generated by RSGMR0 in $\mathbb{R}^n$, and therefore that the sequence of residuals norms produced by these two algorithms are the same. Thus considering (2.12) does not alter the variational properties of the original GMRES, as was already the case for RSGMR0.

Once more, range-space variants of FOM, MINRES and CG handling general right-hand sides may be derived from Algorithm RSGMR. Because it is of special interest below and in practical data assimilation applications, we state the final range-space symmetric variant of FOM, namely Algorithm RSFOM on the next page.

Observe that, as for RSGMR, the sequence of Krylov subspaces generated by RSFOM are identical to those generated by the full-space CG (in exact arithmetic) and hence that the sequence of objective function's values (2.6) are also the same, preserving the desirable monotonicity property of full-space CG.

Algorithm 2.4: $s =$ **RSFOM(** K**,** b **)**

1. Define $\beta_1 = \|b\|, \quad u = Kb, \quad \hat{z}_1 = u/\beta_1$ and $\hat{v}_1 = e_{m+1}/\beta_1$.
2. For $k = 1, \ldots, m+1$,
   (a) $\hat{w}_k^T = [\hat{z}_k^T \, 0]$
   (b) for $i = 1, \ldots, k$,
      i. $H_{i,k} = [\hat{z}_i^T \, 0] \, \hat{w}_k$
      ii. $\hat{w}_k \leftarrow \hat{w}_k - H_{i,k}\hat{v}_i$
   (c) $H_{k,k} \leftarrow H_{k,k} + \gamma$,
   (d) $\hat{z}_{k+1} = [KK^T \, u] \, \hat{w}_{k+1}, \quad \zeta_{k+1} = [u^T \, \beta_1^2] \, \hat{w}_k$,
   (e) $\beta_{k+1} = H_{k+1,k} = \sqrt{[\hat{z}_{k+1}^T \; \zeta_{k+1}] \, \hat{w}_k}$,
   (f) $\hat{v}_{k+1} \leftarrow \hat{w}_k/\beta_{k+1}, \quad \hat{z}_{k+1} \leftarrow \hat{z}_k/\beta_{k+1}$,
   (g) solve $H^{\square} y = \beta_1 e_1$ for y_k,
   (h) if $\|Hy_k - \beta_1 e_1\| < \epsilon_r$, break.
3. Return $s = [K^T \, b]\hat{V}_k y_k$.

The numerical efficiency of our range-space algorithms may clearly be further improved (at the price of a severe loss in readability) by introducing the standard Givens rotations to solve (2.4) or (2.5). The idea is to update a QR factorization of the matrix H as the iterations proceed, factorization from which the solution of either problem can then be derived efficiently. Because our focus in this paper is not on implementation issues and also because the improvements obtained are marginal compared to the cost of the matrix products, we choose not to develop this technique further here, but refer the interested reader to Kelley (1995), Section 3.5, or to Björck, 1996, page 53, for instance.

In the quest for efficient range-space iterative method for the symmetric problem, the authors have also derived RSLSQR, a range-space variant of LSQR (Paige and Saunders, 1982), whose most elaborate formulation is given in Appendix. It is based on applying the LSQR method to the problem

$$\min_s \left\| \begin{bmatrix} I_n \\ K \end{bmatrix} s - \begin{bmatrix} 0 \\ d \end{bmatrix} \right\|_2 \tag{2.14}$$

whose normal equation is identical to (1.1) with $\gamma = 1$ and $L = K$. However, the constraint to operate on vectors in $\mathbb{R}^m$ implies that it has to use products with KK^T and therefore looses the decoupling between K and K^T which is an inherent advantage of the original (full-space) LSQR. As it then turns out, the performance of this algorithm remains disappointing on (even moderately) ill-conditioned problems, which can be explained as follows. At variance with the techniques described so far, the algorithm builds an orthonormal basis both in $\mathbb{R}^{n+m}$ (implicitly) and in $\mathbb{R}^m$. The metric in $\mathbb{R}^{n+m}$ is then of the form

$$\begin{pmatrix} I & 0 \\ 0 & KK^T \end{pmatrix}$$

and numerical difficulties arise as soon as the diagonal blocks of this metric become substantially different in size. We therefore do not consider RSLSQR further here.

This discussion has however another consequence: because range-space methods are constrained to use the products by LK^T or KK^T, their numerical performance should not be expected to compare with that of "square root" methods (like the full-space LSQR) on severely ill-conditioned problems. While this is not a strong restriction for data assimilation applications (where the condition number of $I_n + H^T H$ rarely exceeds 10^4, see

Tshimanga, Gratton, Weaver and Sartenaer, 2008), this certainly reflects that substantially cheaper algorithms also come at a price... Fortunately, we show below that the accuracy obtained with RSGMR and RSFOM remains very acceptable in many cases[(4)].

We conclude this section by a brief comparison of the computational and storage costs associated with full- and range-space GMRES and FOM. Table 2.1 reports the storage and computational costs[(5)] at iteration k of these algorithms, where the computational cost is divided in operations internal to the algorithm and the products which have to be computed (very often using external software). In this table, the terms $k(k+3)/2$ in the storage costs correspond to storing the Hessenberg matrix H and y_k, while the first terms in the internal flops counts correspond to the cost of the relevant orthogonalization process. The symbol $[sol]$ represents the cost of solving the linear least-squares (2.4) or the system (2.5) in the k-dimensional Krylov subspace.

	GMRES	RSGMR
storage	$n(k+1)+k(k+3)/2$	$n+(2m+1)k+k(k+3)/2$
internal flops	$4nk+3n+[sol]$	$4mk+7m+[sol]$
products by	K^T, L	K^T, K, L
	FOM	RSFOM
storage	$n(k+1)+k(k+3)/2$	$(2m+1)k+k(k+3)/2$
internal flops	$4nk+3n+[sol]$	$4mk+6m+[sol]$
products by	K^T, K	K^T, K

Table 2.1: Comparative storage, floating point operations and matrix vector products at iteration k for full- and range-space Krylov solvers

We also give in Table 2.2 the initialization and termination computational costs, assuming that termination occurs at iteration k.

	GMRES, FOM	RSGMR, RSFOM
initialization	$3n$	$2n+2m+\text{prod}(K)$
termination (k)	$2nk$	$2(m+1)k+\text{prod}(K^T)$

Table 2.2: Comparative initialization and termination computational costs for full- and range-space Krylov solvers

The comparative advantages and drawbacks of the range-space methods appear clearly in the above tables: range-space methods are preferable when $m \ll n$ and, for the unsymmetric case, when the cost of an additional product by K per iteration is not prohibitive compared to that of reduced storage. Note that this last restriction does not apply to the symmetric case.

3 Stability and convergence with inexact products

After deriving RSGMR and RSFOM, we now propose an analysis of the behaviour of these algorithms in the case where the products by K^T and L are performed inexactly. There are two main reasons to consider this question. First of all, the inexact nature of computer arithmetic implies that such errors on the products are inevitable. As a consequence, the ideal version of the methods considered so far may be considerably perturbed because the computed quantities may no longer be restricted the the range of the (ideal but unknown) K^T. It is therefore important to verify that this deviation from our assumption does note cause numerical instability. The second reason to consider inexact matrix products

(4) Extensive numerical experiments not reported here show that the best achievable accuracy for RSGMR and RSFOM is comparable to that obtained with normal-equations approaches.

(5) We ignore the constants in this evaluation.

is clearly to allow them in an inexact but cheaper form. This is especially crucial in the context of data assimilation as discussed in Section 1 where the products constitute by far the most costly part of the computation. In particular, inexact products allow us to consider, in this context, inexact solves with B^{-1} or R^{-1} as well as the use of less expensive, degraded versions of the operators H and H^T. Similar considerations, albeit in different contexts, were also used for motivating the analysis proposed by van den Eshof and Sleijpen (2004) and by Simoncini and Szyld (2003) for the full-space versions of iterative Krylov solvers.

Consider RSGMR0 and assume that the first products (corresponding to the scaling of d) are already performed inexactly, that is

$$\hat{v}_1 = d/\beta_1 = d/\sqrt{d^T(K + E_{K,1})(K^T + E_{K^T,1})d} \tag{3.1}$$

for some error matrices $E_{K^T,0}$ and $E_{K,0}$. Assume also that each subsequent product by K^T, K or L is inexact in the sense that, at iteration i,

$$L_i = L + E_{L,i+1}, \quad K_i^T = K^T + E_{K^T,i+1} \text{ and } K_i = K + E_{K,i+1},$$

for some errors $E_{L,i+1}$, $E_{K_i^T+1}$ and $E_{K,i+1}$. If we terminate at iteration k, the solution $s_k = s$ is finally computed using an inexact product with K^T in the formula

$$s_k = (K^T + E_{K^T,*})V_k y_k, \tag{3.2}$$

for some error $E_{K^T,*}$.

We now propose two different models for describing the inaccuracy in the matrix-vector products. In the first model, which we call the *backward-error model*, we assume that

$$\|E_{K,i+1}\| \le \tau_{K,i+1}\|K\|, \text{ and } \|E_{K^T,i+1}\| \le \tau_{K^T,i+1}\|K\| \text{ for } i = 0, \ldots, k, \tag{3.3}$$

$$\|E_{L,i+1}\| \le \tau_{L,i+1}\|L\| \text{ for } i = 1, \ldots, k$$

and

$$\|E_{K^T,*}\| \le \tau_*\|K\|$$

for some tolerances $\tau_{K,i+1}$, $\tau_{K^T,i+1}$, $\tau_{L,i+1}$ and τ_* belonging to the interval $[0, 1)$.

The second error model for inexact products, called the *forward-error model* is stronger and replaces bounds on the errors on the operators by bounds on the errors on the vector resulting from the application of the operator. In this model, we replace the above bounds by

$$\|E_{K,i+1}\, u_n\| \le \tau_{K,i+1}\|Ku_n\|, \text{ and } \|E_{K^T,i+1}\, u_m\| \le \tau_{K^T,i+1}\|Ku_m\| \text{ for } i = 0, \ldots, k,$$
$$\|E_{L,i+1}\, u_n\| \le \tau_{L,i+1}\|Lu_n\| \text{ for } i = 1, \ldots, k \tag{3.4}$$

and

$$\|E_{K^T,*}\, u_m\| \le \tau_*\|Ku_m\|$$

where u_n and u_m are vectors of dimension n and m, respectively, to which the operators K or L (for u_n) or K^T (for u_m) are applied.

Which error model is preferable is unclear in general and might depend on context. While the backward-error approach is more widespread in the literature (it used by van den Eshof and Sleijpen, 2004, and Simoncini and Szyld, 2003), the forward-error approach may be judged more realistic in situations where monitoring the output of a complex process for a specific input is feasible, but impossible or too expensive for all possible inputs.

Our aim is then to bound $\|r_k\|$, the norm of the true residual at iteration k, where

$$r_k = (\gamma I_n + K^T L)s_k - K^T d,$$

if possible by quantities which can be obtained or estimated in the course of the computation. We first conduct our analysis under the assumption that no breakdown occurs, that is

$$\hat{w}_i \neq 0 \quad \text{for} \quad i = 1, \ldots, k, \tag{3.5}$$

(where $\hat{w}_i$ is considered at the end of the normalization, that is after Step 2(b)), but will comment on the situation where this condition fails at the end of the section.

We start by analyzing the perturbed Arnoldi process and prove a useful bound on the residual norm, irrespective of the error model considered.

Lemma 3.1 *Under the above assumptions, one has that*

$$\|r_k\| \leq \|Q_k\| \, \|Hy_k - \beta_1 e_1\| + \gamma \|E_{K^T,*} \hat{V}_k y_k\| + \|K\| \sum_{i=1}^{k} |[y_k]_i| \, \|(LK_*^T \hat{v}_i - L_i K_i^T)\hat{v}_i\| \tag{3.6}$$

where $Q_k \stackrel{\rm def}{=} K^T \hat{V}_{k+1}$.

Proof. By constuction, RSGMR0 ensures that the Arnoldi relation (2.11) holds with perturbed matrices, that is

$$\left[(\gamma I_n + L_1 K_1^T)\hat{v}_1, \ldots, (\gamma I_n + L_k K_k^T)\hat{v}_k\right] = \hat{V}_{k+1} H$$

and therefore

$$K^T \left[(\gamma I_n + L_1 K_1^T)\hat{v}_1, \ldots, (\gamma I_n + L_k K_k^T)\hat{v}_k\right] y_k = K^T \hat{V}_{k+1} H y_k. \tag{3.7}$$

Observe now that, because of (3.1),

$$K^T d = \beta_1 K^T \hat{v}_1 = \beta_1 K^T \hat{V}_{k+1} e_1.$$

Substracting this quantity from both sides of (3.7), we obtain that

$$K^T \left[(\gamma I_n + L_1 K_1^T)\hat{v}_1, \ldots, (\gamma I_n + L_k K_k^T)\hat{v}_k\right] y_k - K^T d = K^T \hat{V}_{k+1}(Hy_k - \beta_1 e_1).$$

Remembering now that s_k is given by (3.2), we see that

$$\begin{aligned}
\|r_k\| &= \|K^T L K_*^T \hat{V}_k y_k - K^T d + \gamma K_*^T \hat{V}_k y_k\| \\
&= \|K^T L K_*^T \hat{V}_k y_k + K^T \hat{V}_{k+1}(Hy_k - \beta_1 e_1) \\
&\qquad -K^T \left[(\gamma I_n + L_1 K_1^T)\hat{v}_1, \ldots, (\gamma I_n + L_k K_k^T)\hat{v}_k\right] y_k + \gamma K_*^T \hat{V}_k y_k\| \\
&= \|K^T L K_*^T \hat{V}_k y_k + Q_k(Hy_k - \beta_1 e_1) + \gamma(K_*^T - K^T)\hat{V}_k y_k \\
&\qquad -K^T \left[L_1 K_1^T \hat{v}_1, \ldots, L_k K_k^T \hat{v}_k\right] y_k\| \\
&\leq \|Q_k\| \, \|Hy_k - \beta_1 e_1\| + \gamma \|E_{K^T,*} \hat{V}_k y_k\| \\
&\qquad +\|K\| \, \|LK_*^T \hat{V}_k y_k - \left[L_1 K_1^T \hat{v}_1, \ldots, L_k K_k^T \hat{v}_k\right] y_k\|
\end{aligned} \tag{3.8}$$

But we have, using the triangle inequality, that

$$\|LK_*^T \hat{V}_k y_k - \left[L_1 K_1^T \hat{v}_1, \ldots, L_k K_k^T \hat{v}_k\right] y_k\| \leq \sum_{i=1}^{k} |[y_k]_i| \, \|LK_*^T \hat{v}_i - L_i K_i^T \hat{v}_i\|$$

and (3.6) follows. □

After this general result, we now focus on the backward-error model.

Lemma 3.2 *Under the above assumptions and assuming the backward-error model, one has that*

$$\|r_k\| \leq \|Q_k\| \, \|Hy_k - \beta_1 e_1\| + \|K\| \, \pi_k \left[\tau_{K^T,*} \gamma \sqrt{k} \|y_k\| + 4\, G^2 \sum_{i=1}^{k} |[y_k]_i| \, \tau_i \right], \tag{3.9}$$

where $\pi_k \stackrel{\rm def}{=} \max_{i=1,\ldots,k} \|\hat{v}_i\|$, $G \stackrel{\rm def}{=} \max[\|K\|, \|L\|]$ *and* $\tau_i \stackrel{\rm def}{=} \max[\tau_*, \tau_{K^T,i}, \tau_{K,i}, \tau_{L,i}]$.

Proof. We first obtain, using the triangle inequality and the definition of π_k, that

$$\begin{aligned} \textstyle\sum_{i=1}^k |[y_k]_i| \, \|(LK_*^T - L_i K_i^T)\hat{v}_i\| &\leq \textstyle\sum_{i=1}^k |[y_k]_i| \, \|LK_*^T - L_i K_i^T\| \, \|\hat{v}_i\| \\ &\leq \pi_k \textstyle\sum_{i=1}^k |[y_k]_i| \, \|LK_*^T - L_i K_i^T\|. \end{aligned}$$

Again using the triangle inequality and (3.3), we now deduce that

$$\begin{aligned} \|LK_*^T - L_i K_i^T\| &= \|LE_{K^T,*} - LE_{K^T,i} - E_{L,i}K^T - E_{L,i}E_{K^T,i}\| \\ &\leq 3\tau_i G^2 + \tau_i^2 G^2 \\ &\leq 4\tau_i G^2, \end{aligned} \tag{3.10}$$

where we used the bound $\tau_i \leq 1$ to derive the last inequality, and therefore that

$$\sum_{i=1}^{k} |[y_k]_i| \, \|(LK_*^T - L_i K_i^T)\hat{v}_i\| \leq 4\pi_k G^2 \sum_{i=1}^{k} |[y_k]_i| \, \tau_i.$$

Substituting this bound in (3.6) and using the inequality

$$\|E_{K^T,*}\hat{V}_k y_k\| \leq \tau_* \|K\| \pi_k \sqrt{k} \|y_k\| \leq \tau_i \|K\| \pi_k \sqrt{k} \|y_k\|$$

then gives (3.9). □

Observe that the proof of this lemma does not use τ_K and that the error on the products by K does not explicitly appear in the bound (3.9), but is present implicitly as the quantities $\|Q_k\|$ and π_k crucially depend on the metric KK^T and therefore, in our case, on these errors $E_{K,i}$ $(i = 1, \ldots, k)$. We now bound these quantities, provided the error remains sufficiently small compared to the condition number of K defined by

$$\kappa(K) \stackrel{\rm def}{=} \frac{\sigma_{\max}(K)}{\sigma^0_{\min}(K)}$$

where $\sigma_{\max}(K)$ is the largest singular value of K and $\sigma^0_{\min}(K)$ is the smallest of the strictly positive ones (Björck, 1996, page 28).

Lemma 3.3 *Suppose that the backward-error model holds, that (3.5) is satisfied and that* τ *is sufficiently small to ensure that, for all* i*,*

$$\tau_i \kappa(K) < \tfrac{1}{6}. \tag{3.11}$$

Then

$$\|Q_k\| \leq \sqrt{2(k+1)} \quad \textit{and} \quad \pi_k \leq \frac{\sqrt{2}}{\sigma^0_{\min}(K)}. \tag{3.12}$$

Proof. From the Cauchy-Schwarz inequality, the fact that $\hat{v}_i \in \text{range}(K^T)$ by construction and (3.3), we first verify that

$$\frac{|\hat{v}_i^T K E_{K^T,i} \hat{v}_i|}{\hat{v}_i^T K K^T \hat{v}_i} \leq \frac{\|E_{K^T,i}\hat{v}_i\|}{\|K^T \hat{v}_i\|} \leq \frac{\tau_{K^T,i}\|K\| \, \|\hat{v}_i\|}{\sigma^0_{\min}(K)\|\hat{v}_i\|} \leq \tau_i \, \kappa(K). \tag{3.13}$$

Similarly,

$$\frac{|\hat{v}_i^T E_{K,i} K^T \hat{v}_i|}{\hat{v}_i^T K K^T \hat{v}_i} \leq \tau_i \kappa(K) \quad \text{and} \quad \frac{|\hat{v}_i^T E_{K,i} E_{K^T,i} \hat{v}_i|}{\hat{v}_i^T K K^T \hat{v}_i} \leq \tau_i^2 \kappa(K)^2. \tag{3.14}$$

Thus we deduce from (3.11), (3.13) and (3.14) that

$$\left| \frac{\hat{v}_i^T (K E_{K^T,i} + E_{K,i} K^T + E_{K,i} E_{K^T,i}) \hat{v}_i}{\hat{v}_i^T K K^T \hat{v}_i} \right| \leq 3\tau_i \kappa(K) \leq \frac{1}{2}. \tag{3.15}$$

As a consequence, since clearly

$$\hat{v}_i^T (K+E_{K,i})(K^T+E_{K^T,i})\hat{v}_i = \left[1 + \frac{\hat{v}_i^T (K E_{K^T,i} + E_{K,i} K^T + E_{K,i} E_{K^T,i}) \hat{v}_i}{\hat{v}_i^T K K^T \hat{v}_i}\right] \hat{v}_i^T K K^T \hat{v}_i,$$

we thus obtain that

$$\frac{1}{2}\|K^T \hat{v}_i\|^2 \leq \hat{v}_i^T (K + E_{K,i})(K^T + E_{K^T,i})\hat{v}_i \leq \frac{3}{2}\|K^T \hat{v}_i\|^2 \tag{3.16}$$

and the normalization (with respect to the inexact metric $K_i K_i^T$) performed by the algorithm to ensure that

$$\hat{v}_i^T (K + E_{K,i})(K^T + E_{K^T,i})\hat{v}_i = 1 \tag{3.17}$$

is legal because of (3.5) and because $\hat{v}_i$ belongs to the range of K^T by construction. Inserting (3.17) in (3.16), we then deduce that

$$\frac{2}{3} < \hat{v}_i^T K K^T \hat{v}_i < 2. \tag{3.18}$$

which in turn implies that

$$\|Q_k\| = \|[K^T \hat{v}_1, \ldots, K^T \hat{v}_{k+1}]\| \leq \sqrt{2(k+1)}.$$

as requested. We finally prove the second part of (3.12) by observing that, for $i = 1, \ldots, k$,

$$\|\hat{v}_i\|^2 = \frac{\hat{v}_i^T \hat{v}_i}{\hat{v}_i^T K K^T \hat{v}_i} \hat{v}_i^T K K^T \hat{v}_i \leq \frac{2}{\sigma_{\min}^0(K)^2}$$

where we used (3.18) to deduce the last inequality. □

While the bound on $\|Q_k\|$ given by (3.12) is formally correct and conceptually tight, it is often very pessimistic in practice, as it does not take the typical random nature of the error into account. In particular, it is not unusual for the residual error to be independent of the factor $\sqrt{k+1}$ in the bound expression.

We may then combine the two above lemmas and obtain a final set of bounds on $\|r_k\|$ for the case where the backward-error model is considered.

Theorem 3.4 *Suppose that the backward-error model holds. Then*

$$\begin{aligned} \|r_k\| &\leq \sqrt{2(k+1)}\,\|Hy_k - \beta_1 e_1\| + \|K\|\pi_k \left[\tau_* \gamma \sqrt{k}\|y_k\| + 4\,G^2 \textstyle\sum_{i=1}^k |[y_k]_i|\,\tau_i\right] \\ &\leq \sqrt{2(k+1)}\,\|Hy_k - \beta_1 e_1\| + \tau_{\max}\sqrt{k}\|K\|\pi_k(\gamma + 4\,G^2)\|y_k\| \\ &\leq \sqrt{2(k+1)}\big[\|Hy_k - \beta_1 e_1\| + \tau_{\max}\kappa(K)\,(\gamma + 4\,G^2)\|y_k\|\big]. \end{aligned} \tag{3.19}$$

where $\tau_{\max} \stackrel{\rm def}{=} \max[\tau_1, \ldots, \tau_k]$.

The first and sharpest of these three bounds allows the consideration of levels of inexactness in the products that vary from iteration to iteration, in the spirit of Simoncini and Szyld (2003) and van den Eshof and Sleijpen (2004). It only involves computable quantities, as $\|Hy_k - \beta_1 e_1\|$ (the norm of the residual in $\mathcal{K}_k$), $\|y_k\|$ and π_k can be recurred within the RSGMR0 iterations in $O(\max[m, k^2])$ operations[(6)]. The second bound is most interesting in the case where a constant bound on inexactness is selected. The third bound is similar, but does not requires the computation of π_k, at the cost of a (experimentally often severe) overestimation.

We observe the specific role of τ_* in the error bound, as it appears multiplied by $\|y_k\|$ rather than by one of the $|[y_k]_i|$, a phenomenon specific to the range-space setting. The final product by K^T to produce s in Step 4 should therefore be computed with a potentially higher accuracy than any of the preceding products if the overall error bound is to be preserved.

It is also interesting to note that there is little to be gained by controlling the errors of the products of K, K^T and L differently, as the error-bound analysis show their effects to be intertwined (see e.g. (3.15)). This is also born out in our numerical experiments (see Section 4).

Because our analysis is based on the (in this case inexact) Arnoldi process, it also applies to the FOM context and similar conclusions therefore hold for the inexact variant of RSFOM. However, the standard CG and MINRES methods are no longer equivalent to FOM and GMRES in the context of unsymmetric perturbations (because the latter methods consider an underlying symmetric tridiagonal matrix H), and the bounds (3.19) therefore do not extend as such to these methods.

We now turn to deriving similar bounds for the case where the forward-error model holds. We first derive the equivalent to Lemma 3.3.

Lemma 3.5 *Suppose that the forward-error model holds, that (3.5) is satisfied and that τ verifies*

$$\tau < \tfrac{1}{6}. \tag{3.20}$$

Then

$$\|K^T \hat{v}_i\| \leq \sqrt{2} \quad \textit{and} \quad \|Q_k\| \leq \sqrt{2(k+1)}. \tag{3.21}$$

Proof. Inspired by (3.13), we verify that

$$\frac{|\hat{v}_i^T K E_{K^T,i} \hat{v}_i|}{\hat{v}_i^T K K^T \hat{v}_i} \leq \frac{\|E_{K^T,i}\hat{v}_i\|}{\|K^T\hat{v}_i\|} \leq \frac{\tau_{K^T,i}\|K^T\hat{v}_i\|}{\|K^T\hat{v}_i\|} = \tau \tag{3.22}$$

where we successively used the Cauchy-Schwarz inequality and (3.4). Similarly,

$$\frac{|\hat{v}_i^T E_{K,i} K^T \hat{v}_i|}{\hat{v}_i^T K K^T \hat{v}_i} \leq \tau \quad \text{and} \quad \frac{|\hat{v}_i^T E_{K,i} E_{K^T,i} \hat{v}_i|}{\hat{v}_i^T K K^T \hat{v}_i} \leq \tau^2. \tag{3.23}$$

Thus we deduce from (3.20), (3.22) and (3.23) that

$$\left| \frac{\hat{v}_i^T (K E_{K^T,i} + E_{K,i} K^T + E_{K,i} E_{K^T,i}) \hat{v}_i}{\hat{v}_i^T K K^T \hat{v}_i} \right| \leq 3\tau \leq \frac{1}{2}.$$

We then continue the proof as in Lemma 3.3 and deduce the first bound of (3.21) instead of (3.18). The second part of (3.21) then follows exactly as in Lemma 3.3. □

We may then use the first of these bounds in a result similar in spirit to Lemma 3.2.

(6) Assuming the use of Givens rotations to compute y_k.

Theorem 3.6 *Under the above assumptions and assuming the forward-error model, one has that*

$$\begin{aligned}\|r_k\| &\leq \sqrt{2(k+1)}\,\|Hy_k - \beta_1 e_1\| + \sqrt{2}\left[\tau_*\gamma\sqrt{k}\|y_k\| + 4\,G\,\|K\|\sum_{i=1}^{k}|[y_k]_i|\,\tau_i\right] \\ &\leq \sqrt{2(k+1)}\Big[\|Hy_k - \beta_1 e_1\| + \tau_{\max}\,(\gamma + 4\,G\,\|K\|)\|y_k\|\Big].\end{aligned} \tag{3.24}$$

Proof. We note that

$$\|LE_{K^T,*}\hat{v}_i\| \leq \tau_*\,\|L\|\,\|K^T\hat{v}_i\| \leq \sqrt{2}\,\tau_*\,\|L\|, \leq \sqrt{2}\,\tau_i\,\|L\|,$$

where we used (3.4) qnd the first part of (3.21). Similarly,

$$\|LE_{K^T,i}\hat{v}_i\| \leq \sqrt{2}\,\tau_{K^T,i}\|L\| \leq \sqrt{2}\,\tau_i\|L\|.$$

We also derive in the same manner that

$$\|E_{L,i}K^T\hat{v}_i\| \leq \tau_{L,i}\|LK^T\hat{v}_i\| \leq \tau_{L,i}\|L\|\,\|K^T\hat{v}_i\| \leq \sqrt{2}\,\tau_i\|L\|$$

and finally that

$$\|E_{L,i}E_{K^T,i}\hat{v}_i\| \leq \tau_{L,i}\|LE_{K^T,i}\hat{v}_i\| \leq \tau_{L,i}\|L\|\,\tau_{K^T,i}\|K^T\hat{v}_i\| \leq \sqrt{2}\,\tau_i\|L\|.$$

Combining these bounds, we deduce that

$$\begin{aligned}\|(LK_*^T - L_iK_i^T)\hat{v}_i\| &= \|LE_{K^T,*}\hat{v}_i - LE_{K^T,i}\hat{v}_i - E_{L,i}K^T\hat{v}_i - E_{L,i}E_{K^T,i}\hat{v}_i\| \\ &\leq 4\sqrt{2}\,\tau_i\,G.\end{aligned}$$

Substituting this last bound in (3.6) and using

$$\|E_{K^T,*}\hat{V}_k y_k\| \leq \tau_*\,\|K^T\hat{V}_k\|\,\|y_k\| \leq \tau_*\,\sqrt{2k}\,\|y_k\|$$

then yields the first inequality in (3.24), from which the second easily follows using the second part of (3.21). □

A comparison of Theorems 3.4 and 3.6 shows that the second is considerably stronger, in that the conditioning of K or the factor $\|K\|\pi_k$ no longer appear. As a consequence, possibly significantly more inexact products are possible in the forward-error model than in the backward-error one, with the same final accuracy requirement.

We observe that breakdown may obviously occur in the inexact case just as with exact products, and can be solved using the same strategies. In the context of inexact products, a simpler method also consists in recomputing the last matrix-vector product with a marginally different accuracy threshold, which then corresponds to a slightly different error in the inexact product. This suppresses the breakdown (with probability one) and allows the algorithm to proceed.

We finally note that, because it uses exact arithmetic, our analysis applies without modification to mathematically equivalent variants of GMRES, including classical or iterated Gram-Schmidt procedures, as well as Householder variants.

4 Numerical illustrations

We now illustrate some of the concepts and motivation numerically. Our first example is in the symmetric case and aims at motivating the consideration of FOM besides CG for inexact products. It has been noted in van den Eshof and Sleijpen (2004) and Simoncini and Szyld (2003) that managing inexact products may be more difficult for CG than for

FOM, mostly because the convergence of $\|Hy_k - \beta e_1\|$ is uncertain in the first case, while it is ensured in the second by the full orthogonalisation property. We now provide a simple example where this phenomenon happens. We consider A to be a symmetric tridiagonal matrix of dimension $n = 100$, whose main diagonal is given by n entries whose logarithms are equally spaced between -1 and 5, and whose off-diagonal contain values equally spaced between 0.08 and 1.5. This matrix is safely positive definite, with its minimum eigenvalue around 0.05, and moderatly ill-conditioned ($\kappa(A) \approx 10^6$). The right-hand side of the system is then given by e_1, the first vector of the canonical basis (note that $\|b\| = 1$). Figure 4.1 shows the convergence of the normalized Krylov residual $\|q_k\|/(\|A\| \, \|s_*\|)$ as a function of k, where s_* is the true solution of the system. The left-most graph shows the convergence of the CG normalized residual as the top-most, mostly horizontal dotted curve and that of the FOM normalized residual as the left-most, nearly vertical continuous one, in the case where products by $K = L$ are exact. The convergence of both methods is reasonable in this case, although that of FOM is faster and less subject to rounding noise. The graph on the right shows the same quantities for the case where products are inexact in the sense that, at iteration $k > 2$, a small ($\tau_i = \tau = 10^{-9}$) multiple of the normalized Krylov basis vector computed at iteration $k - 2$ is added to the product vector. This graph clearly demonstrates that inexact products barely affect FOM, but may have a very negative effect on CG, for which the normalized residual is never decreased below the level τ (we only illustrate the first 500 iterations).

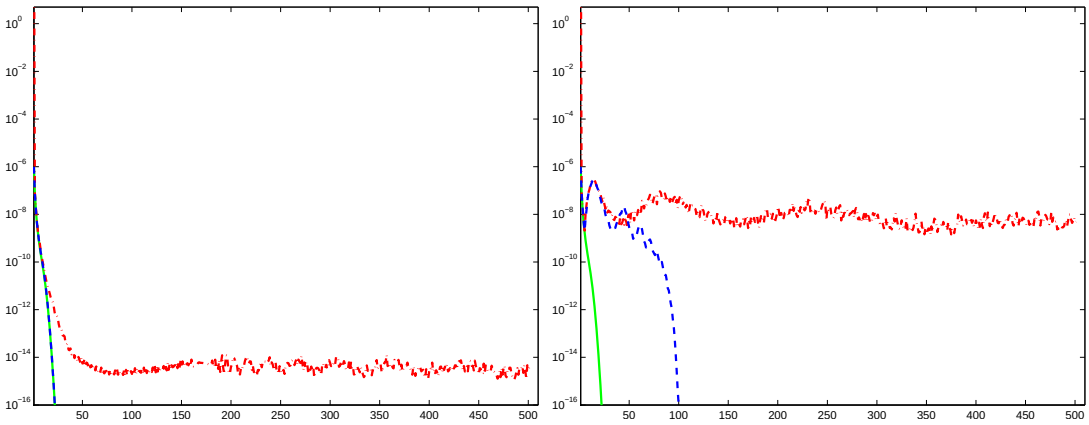

Figure 4.1: The effect of inexact products on FOM, CG with reorthogonalization and CG (exact products on the left, inexact products on the right) on computed residual norm as a function of k

However, one should remember that FOM has higher memory requirements than CG due to its full orthogonalization procedure, making the previous comparison somewhat biased. For a more balanced view, it is therefore natural to consider applying a variant of CG where reorthogonalization is performed at every iteration (thereby requiring similar storage) to the same problem. The result of this test is illustrated by the middle dashed curve in both graphs of Figure 4.1. While this variant of CG is nearly undistinguishable form FOM on our example when products are computed exactly, significant differences reappear for inexact products. It is clear in this case that reorthogonalization improves convergence (which now happens in exactly n steps) for CG, but the performance of FOM remains markedly superior. Other extensive tests (not reported here) indicate that, although not always as extremely as in the example presented above, FOM globally outperforms CG with reorthogonalization, which itself is very often more efficient than the standard CG. This is especially true when inexact products are considered.

Our second numerical illustration aims at showing effect of using different strategies for allowing inexactness in the products in applying RSGMR to the system (1.1). The simplest technique is to decide of a product accuracy threshold in view of the desired

final accuracy on the normalized true system residual, the latter being given (assuming termination at iteration k) by

$$\frac{\|r_k\|}{\|A\|\,\|s_*\|}, \tag{4.1}$$

where s_* is again the true system's solution. In the results presented below, we have chosen this threshold τ according to the formula

$$\tau_{BEM} = \frac{40\epsilon}{\sqrt{2(m+1)}\,\kappa(K)} \quad \text{and} \quad \tau_{FEM} = \epsilon,$$

where BEM and FEM refer to the backward-error model and the forward-error model, respectively. The formula for τ_{BEM} is inspired by the bound (3.19) (using the empirical observations that $\gamma + 4G^2 \approx 4G^2$ and $\|y_k\| = O(\|s_*\|)$ and a factor 10 to counteract the loosesness of the bound). That for τ_{FEM} is directly derived from (3.24). The RSGMR algorithm using these two strategies and associated error models are applied to an example with $\gamma = 1$, $n = 1000$ and $m = 100$ whose matrices K and L are chosen randomly with m nonzero singular values whose logarithms are equally spaced between 0.1 and 0.3. The requested accuracy is chosen as $\epsilon = 10^{-5}$. The results are shown in Figure 4.2, in which the true normalized residual norms (4.1) are represented by a solid line, normalized Krylov residual norms

$$\frac{\|q_k\|}{\|H\|\,\|y_k\|}$$

by a dashed line, final accuracy requirement by an horizontal dotted line and accuracy threshold τ at iteration k by a dashed-dotted line. The left graph shows the effect of applying the backward-error model (with $\tau_k = \tau_{BEM}$) and the left graph that of applying the forward-error model (with $\tau_k = \tau_{FEM}$).

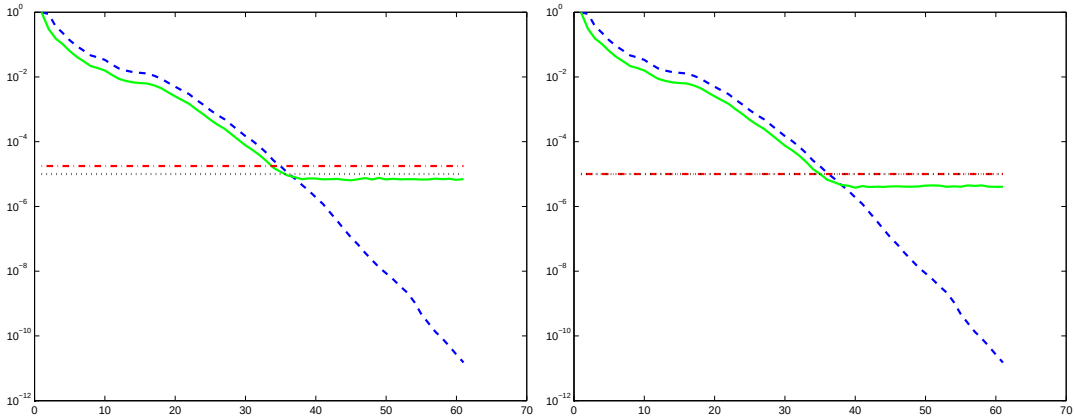

Figure 4.2: Accuracy threshold, normalized true and Krylov residual norms as a function of k when using the backward-error (left) and forward-error (right) accuracy models for the products by K, K^T and L ($\sigma_{\min}(A) \approx 2 \times 10^{-2}$, $\sigma_{\max}(A) \approx 4$)

The differences between the two error strategies are small in this case. Being also interested in variable accuracy thresholds strategies, we also applied a technique recommended by Simoncini and Szyld (2003) (and adapted to (4.1)): we ran the same example with the choice

$$\tau_i = \tau_{SS} = \frac{\sigma_{\min}(A)}{m}\,\frac{\epsilon\,\|s_*\|}{\|q_i\|} \quad \text{and} \quad \tau_* = \tau_{BEM} \tag{4.2}$$

within the backward-error accuracy model, which is consistent with assumptions used in this reference. The result of this experiment is shown in Figure 4.3.
We see in this case a profile of the accuracy threshold similar to those reported by Simoncini and Szyld, but note that the variable theshold is significantly below the thesholds

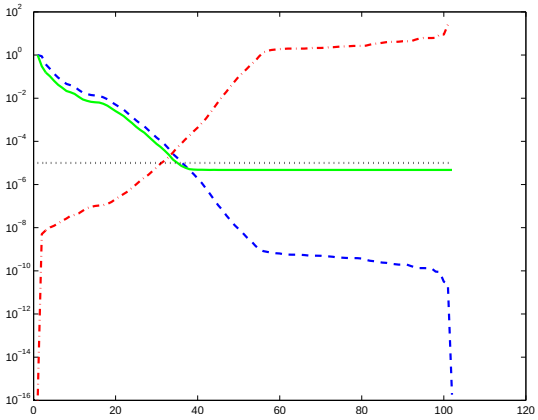

Figure 4.3: Accuracy threshold, normalized true and Krylov residual norms as a function of k when using the variable backward accuracy requirements described by (4.2) for the products by K, K^T and L ($\sigma_{\min}(A) \approx 2 \times 10^{-2}$, $\sigma_{\max}(A) \approx 4$)

illustrated in Figure 4.2 practically up to the point where the maximal accuracy has been reached for the true residual.

We now apply the same algorithm on the same example modified so that the nonzero singular values of K and L have their logarithm equally spaced between 1 and 3 (instead of 0.1 and 0.3), significantly affecting both $\kappa(K)$ and $\|K\|$. The results are shown in Figure 4.4.
As expected the accuracy threshold for the backward-error model decreases (to compensate the change in conditioning and $\|K\|$), but the forward-error-model threshold remains unchanged. Unfortunately, the variable strategy now picks up a too severe accuracy threshold because of the smaller $\sigma_{\min}(A)$ and only allows a small error on the products, thereby illustrating the difficulties of designing a robust and efficient variable accuracy scheme.

We finally conclude our numerical illustrations by verifying the claim made above that manipulating the accuracy thresholds on K and L differently does not affect the algorithm much. To this aim, we return to our second test case (for which the behaviour of RSGMR is shown in Figure 4.2), and run the algorithm first allowing inexact products with L only, and then with K and K^T only. Figure 4.5 presents the results of these two runs (using the backward-error model), and one checks that they do not differ significantly from the left graph of Figure 4.2 (except maybe a marginally lower final true residual when the product with K^T is exact, probably resulting from a better accuracy in recovering the final s).

5 Conclusions and perspectives

Motivated by applications in inverse problems and, more specifically, by data assimilation for oceanography and wheather forecasting, we have introduced range-space variants of GMRES, FOM, CG and MINRES. These variants are characterized by significantly lower storage requirements and inner computational costs than their full-space versions, at the cost of an additional matrix vector product per iteration in the unsymmetric case.

With the aim of reducing the computational burden further, we have also considered how strategies involving inexact matrix-vector products could be applied in the new methods, and have distinguished two distinct models to describe the inexactness allowed in these products. The first of this model considers the backward error on the matrix itself, while the second only controls the forward error on the vector resulting from the product. Formal error bounds on the true system residual are derived in both cases, indicating that the second strategy might allow looser accuracy requirements than the first.

We have finally provided numerical illustrations confirming these findings. Further-

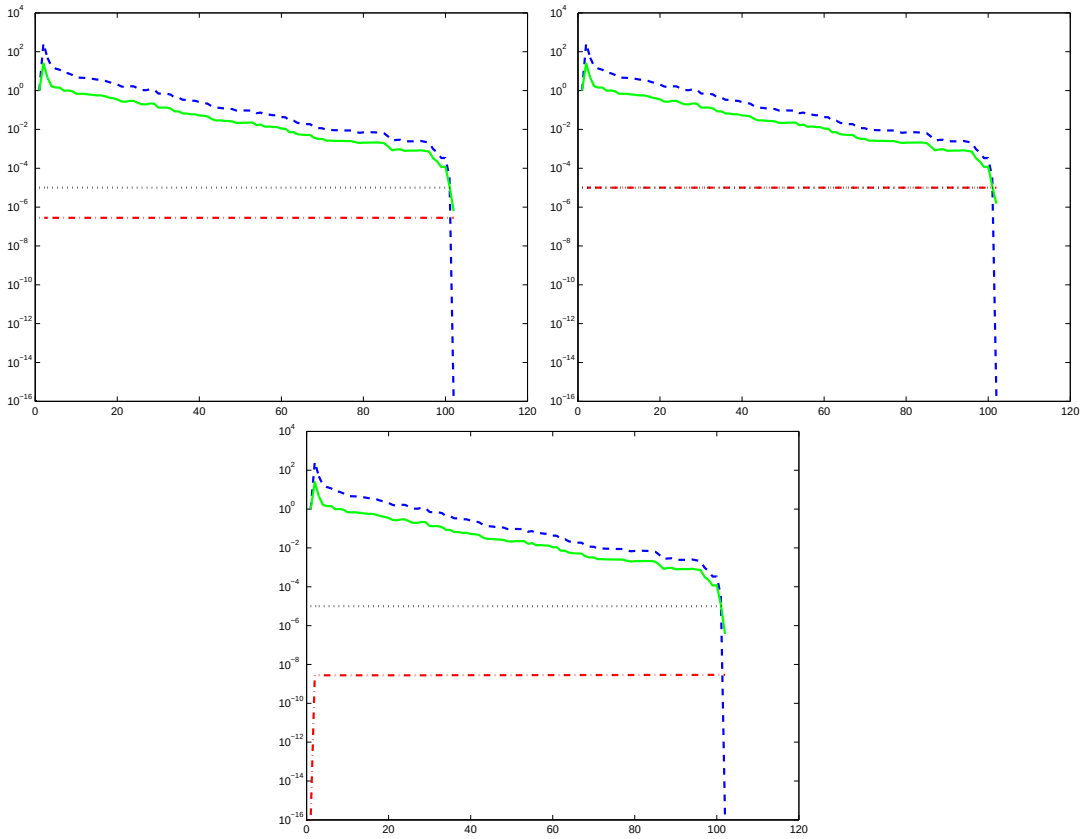

Figure 4.4: Accuracy threshold, normalized true and Krylov residual norms as a function of k when using the backward-error (top-left) and forward-error (top-right) accuracy models and the variable model (bottom) for the products by K, K^T and L ($\sigma_{\min}(A) \approx 2 \times 10^{-4}$, $\sigma_{\max}(A) \approx 5 \times 10^5$)

more, numerical evidence is also presented that indicates that FOM-like methods might be preferable to CG when products are computed inexactly, and that variable accuracy strategies may be difficult to design.

The authors are well aware that a complete numerical evaluation of the new range-space methods is necessary and that a number of issues raised in this paper merit further development. These include, in particular, the design of an efficient and robust variable accuracy scheme, clear stopping rules based on the formal error estimates, additional short recurrences techniques beyond CG and MINRES, and several other implementations issues such as the selection of the most suitable GMRES formulation amongst Gram-Schmidt and Householder variants. The extension of the forward-error model to derive tighter residual bounds for full-space methods may also be of interest. These topics are the object of ongoing research.

It is of course especially worthwhile to apply the new methods in the context of the motivating inverse problems in data assimilation, and also to specialize them further to gracefully handle the correlation matrices B and R. Inexact products are especially appealing in this context, where iterative solutions of linear systems involving these matrices may be truncated, or variable-fidelity techniques (like multigrid or simplified physics) exploited to alter the accuracy of the underlying models.

Acknowledgements. This research presented in this paper was conducted with the support of the "Assimilation de Données pour la Terre, l'Atmosphère et l'Océan (ADTAO)" project, funded by the Fondation "Sciences et Technologies pour l'Aéoronautique et l'Espace (STAE)", Toulouse, France, within the "Réseau Thématique de Recherche Avancée (RTRA)".

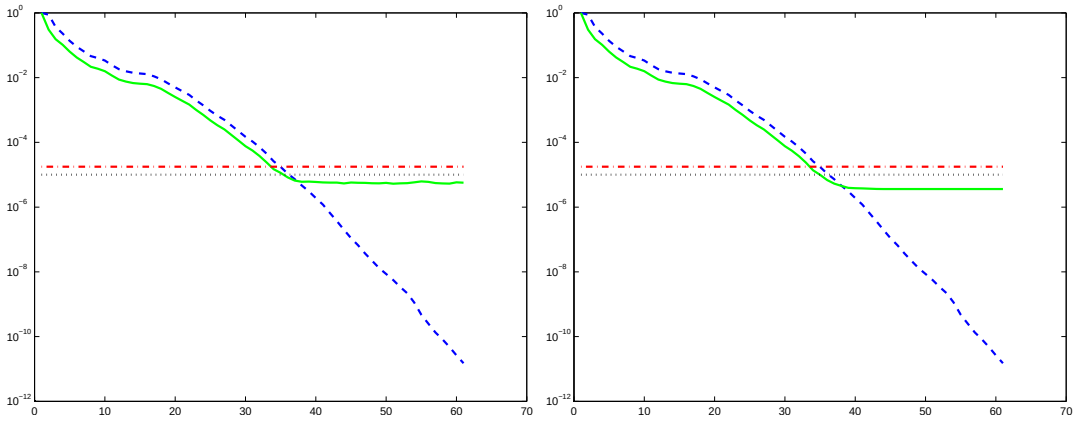

Figure 4.5: Impact of different accuracy thresholds for the products by K and L: no error on products by L on left and no error on products by K and K^T on the right

References

L. Amodei. Solution approchée pour un problème d'assimilation de donnéees météorologiques avec prise en compte de l'erreur modèle. *C. R. Acad. Sci., Paris*, **321**, 1087–1094, 1995.

A. Bennett and M. Thornburn. The generalized inverse of a non-linear quasi-geostrophic ocean circulation model. *J. Phys. Oceanogr.*, **3**, 213–230, 1992.

Å. Björck. *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, USA, 1996.

S. Cohn, A. Da Silva, J. Guo, M. Sienkiewicz, and D. Lamich. Assessing the effects of data selection with the DAO physical-space statistical analysis system. *Mon. Weather. Rev.*, **126**, 2913–2926, 1998.

A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-Region Methods*. Number 01 *in* 'MPS-SIAM Series on Optimization'. SIAM, Philadelphia, USA, 2000.

Ph. Courtier, J.-N. Thépaut, and A. Hollingsworth. A strategy for operational implementation of 4D-Var using an incremental approach. *Q. J. R. Meteorol. Soc.*, **120**, 1367–1388, 1994.

A. Da Silva, J. Pfaendtner, J. Guo, M. Sienkiewicz, and S. Cohn. Assessing the effects of data selection with the DAO physical-space statistical analysis system. *in* 'Proceedings of the 2nd WMO Symposium on assimilation of observations in meteorology and oceanography', pp. 273–278, Geneva, 1995. World Meteorological Organization.

A. El Akkroui, P. Gauthier, S. Pellerin, and S. Buis. Intercomparison of the primal and dual formulations of variational data assimilation. *Q. J. R. Meteorol. Soc.*, **134**(633), 1015–1025, 2008.

S. Gratton and J. Tshimanga. An observation-space formulation of variational assimilation using a restricted preconditioned conjugate-gradient algorithm. *Q. J. R. Meteorol. Soc.*, **135**, 1573–1585, 2009.

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

P. C. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. SIAM, Philadelphia, USA, 1997.

C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. Frontiers in Applied Mathematics. SIAM, Philadelphia, USA, 1995.

C. C. Paige and M. A. Saunders. LSQR: an algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, **8**, 43–71, 1982.

F. Rabier. Overview of global data assimilation developments in numerical weather-prediction centres. *Q. J. R. Meteorol. Soc.*, **131**, 3215–3233, 2005.

L. Reichel and Q. Ye. Breakdown-free GMRES for singular systems. *SIAM Journal on Matrix Analysis and Applications*, **26**(4), 1001–1021, 2005.

Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, Boston, USA, 1996.

Y. Saad and M. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, **7**, 856–869, 1986.

V. Simoncini and D. Szyld. Theory of inexact Krylov subspace methods and applications to scientific computing. *SIAM Journal on Scientific Computing*, **25**(2), 454–477, 2003.

Y. Trémolet. Accounting for an imperfect model in 4D-Var. *Q. J. R. Meteorol. Soc.*, **132**, 2483–2504, 2006*a*.

Y. Trémolet. Model error estimation in 4D-Var. *Q. J. R. Meteorol. Soc.*, **133**, 1267–1280, 2006*b*.

J. Tshimanga, S. Gratton, A. Weaver, and A. Sartenaer. Limited-memory preconditioners with application to incremental four-dimensional variational data assimilation. *Q. J. R. Meteor. Soc.*, **134**, 751–769, 2008.

J. van den Eshof and G. L. G. Sleijpen. Inexact Krylov subspace methods for linear systems. *SIAM Journal on Matrix Analysis and Applications*, **26**(1), 125–153, 2004.

J. van den Eshof, G. L. G. Sleijpen, and M. B. van Gijzen. Relaxation strategies for nested Krylov methods. *Journal of Computational and Applied Mathematics*, **177**, 347–365, 2005.

Appendix: A range-space variant of LSQR

The range-space variant of LSQR is derived for the problem (2.14) and is stated as Algorithm RSLSQR below. In this description, the GIVROT$(\bar{\rho}_k, \beta_{k+1})$ algorithm computes the parameters of a Givens rotation such that

$$\begin{pmatrix} \kappa_k & \sigma_k \\ -\sigma_k & \kappa_k \end{pmatrix} \begin{pmatrix} \bar{\rho}_k \\ \beta_{K+1} \end{pmatrix} = \begin{pmatrix} \rho_k \\ 0 \end{pmatrix},$$

(see Björck, 1996, page 54). The numerical difficulty reported at the end of Section 2 arises in the computation of β_{k+1} at Step 5(a), and is not specific to the formula used (several variants have been tried with worse results).

Algorithm .1: $s =$ **RSLSQR(** K**,** d **)**

1. Define $\beta_1 = \|d\|, \quad \hat{h}_1 = d/\beta_1, \quad \hat{v}_1 = \hat{h}_1, \quad \hat{s}_1 = 0, \quad \hat{z}_1 = KK^T\hat{v}_1,$
2. $\alpha_1 = \sqrt{\hat{z}_1^T\hat{v}_1}, \quad \hat{v}_1 \leftarrow \hat{v}_1/\alpha_1, \quad \hat{z}_1 \leftarrow \hat{z}_1/\alpha_1,$
3. $\hat{w}_1 = \hat{v}_1, \quad \bar{\phi}_1 = \beta_1, \quad \bar{\rho}_1 = \alpha_1,$
4. For $k = 1, \ldots, m,$
   (a) $\beta_{k+1} = \sqrt{\alpha_k^2 + \hat{z}_k^T(\hat{v}_k + \hat{z}_k) - 2\alpha_k\hat{z}_k^T\hat{h}_k},$
   (b) $\hat{h}_{k+1} = (\hat{v}_k + \hat{z}_k - \alpha_k\hat{h}_k)/\beta_{k+1}, \quad \hat{v}_{k+1} = \hat{h}_{k+1} - \beta_{k+1}\hat{v}_k,$
   (c) $\hat{z}_{k+1} = KK^T\hat{v}_{k+1},$
   (d) $\alpha_{k+1} = \sqrt{\hat{z}_{k+1}^T\hat{v}_{k+1}}, \quad \hat{v}_{k+1} \leftarrow \hat{v}_{k+1}/\alpha_{k+1}, \quad \hat{z}_{k+1} \leftarrow \hat{z}_{k+1}/\alpha_{k+1},$
   (e) $[\kappa_k, \sigma_k, \rho_k] = \mathrm{GIVROT}(\bar{\rho}_k, \beta_{k+1}),$
   (f) $\theta_k = \sigma_k\alpha_{k+1}, \quad \bar{\rho}_{k+1} = \kappa_k\alpha_{k+1}, \quad \phi_{k+1} = \kappa_k\bar{\phi}_k, \quad \bar{\phi}_{k+1} = -\sigma_k\bar{\phi}_k,$
   (g) $\hat{s}_k = \hat{s}_{k-1} + (\phi_k/\rho_k)\hat{w}_k \quad \hat{w}_{k+1} = \hat{v}_k - (\theta_k/\rho_k)\hat{w}_k$
5. Return $s = K^T\hat{s}_k.$