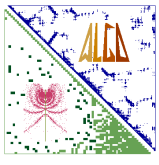
CERFACS

Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique

Differentiating the Method of Conjugate Gradients

Serge Gratton, David Titley-Peloquin, Philippe Toint, Jean Tshimanga Ilunga

Technical Report TR/PA/12/125



Publications of the Parallel Algorithms Team

http://www.cerfacs.fr/algor/publications/

Differentiating the Method of Conjugate Gradients*

Serge Gratton† David Titley-Peloquin‡ Philippe Toint§
Jean Tshimanga Ilunga¶

May 22, 2013

Abstract

The method of conjugate gradients (CG) is widely used for the iterative solution of large sparse systems of equations $Ax = b$, where $A \in \Re^{n\times n}$ is symmetric positive definite. Let x_k denote the k–th iterate of CG. This is a nonlinear differentiable function of b. In this paper we obtain expressions for J_k, the Jacobian matrix of x_k with respect to b. We use these expressions to obtain bounds on $\|J_k\|_2$, the spectral norm condition number of x_k, and discuss algorithms to compute or estimate $J_k v$ and $J_k^T v$ for a given vector v.

1 Introduction

The method of conjugate gradients (CG) of Hestenes and Stiefel [10] is widely used for the iterative solution of large sparse systems of equations $Ax = b$, where $A \in \Re^{n\times n}$ is symmetric positive definite. Let x_k denote the k–th iterate of CG. It can easily be verified that $x_k = x_k(b)$ is a nonlinear differentiable function of b. See, e.g., [27] for some effects of the nonlinearity, or the recent monograph [15] for a more general discussion on the nonlinearity of Krylov subspace methods. In this paper we obtain expressions for the Jacobian of x_k with respect to the right-hand side vector b,

$$J_k = \frac{\partial x_k}{\partial b} \in \Re^{n\times n}.$$

Our main motivation for studying this topic comes from the following problem in statistics, sometimes referred to as truncated CG regression or the partial least-squares problem; see, e.g., [3] and the references therein. Consider estimating $\bar{x} = A^{-1}\bar{b}$ from a given noisy right-hand side

*The 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'Aéronautique et l'Espace (STAE)", Toulouse, France, within the "Réseau Thématique de Recherche Avancée (RTRA)".

†IRIT-CERFACS, 42 ave Gaspard Coriolis, 31057 Toulouse, France (gratton@cerfacs.fr).

‡IRIT-ENSEEIHT, 2 rue Charles Camichel, B. P. 7122, 31071 Toulouse, France (dtitleyp@enseeiht.fr). The research of this author was supported by a postdoctoral fellowship from the Natural Sciences and Engineering Research Council (NSERC) of Canada.

§Université de Namur, 61 rue de Bruxelles, B5000 Namur, Belgium (philippe.toint@unamur.be).

¶IRIT-ENSEEIHT, 2 rue Charles Camichel, B. P. 7122, 31071 Toulouse, France (jean.tshimanga@enseeiht.fr).

$b = \bar{b} + \Delta b$, where $\Delta b \sim (0, \Sigma)$ is random noise. It is well known that the best linear unbiaised estimator of $\bar{x}$ is $x = A^{-1}b$, with

$$\bar{x} - x = A^{-1}\Delta b, \quad \text{cov}\{\bar{x} - x\} = A^{-1}\Sigma A^{-1}.$$

In practise, however, it is often unfeasible to compute x. Rather, one computes x_k, the k–th iterate of CG applied to the noisy system, with $k \ll n$. The question then becomes: what is $\text{cov}\{\bar{x} - x_k\}$? The following idea has received some recent attention in the data assimilation community and has made its way into operational data assimilation codes [20, 28]. First linearize $x_k(b) \approx x_k(\bar{b}) + J_k\Delta b$. Then, to first order,

$$\text{cov}\{\bar{x} - x_k(b)\} \approx \text{cov}\{\bar{x} - x_k(\bar{b}) - J_k\Delta b\} = \text{cov}\{J_k\Delta b\} = J_k\Sigma J_k^T.$$

The above sparked our interest in the mathematical properties of, and computations involving, the Jacobian matrix J_k.

It is usually not feasible to compute and store the (generally dense) matrix J_k. Therefore, one is usually interested in a scalar measure of the sensitivity of a computed solution, rather than the entire Jacobian matrix. For example, one quantity of interest might be the absolute condition number of x_k with respect to perturbations in b (in any chosen norm):

$$\|J_k\| = \lim_{\epsilon \to 0} \sup_{\|\Delta b\| \leq \epsilon} \frac{\|x_k(b + \Delta b) - x_k(b)\|}{\|\Delta b\|}. \tag{1}$$

See, e.g., [11, Chapter 3] for a proof of the above. We use one of our expressions for the Jacobian to obtain bounds on $\|J_k\|_2$, the spectral norm condition number of x_k. In [20, 28], the following is used as a measure of sensitivity:

$$v^T \text{cov}\{\bar{x} - x_k(b)\}v \approx v^T J_k \Sigma J_k^T v$$

for a given vector v. Hence, matrix-vector products of the form $J_k^T v$ are required. We discuss methods to compute or estimate the quantities $J_k v$ and $J_k^T v$.

There has been some related work on the sensitivity of Krylov subspace methods. Kuznetsov et. al. [1, 13] obtain expressions for the condition number of a Krylov subspace $\mathcal{K}_k(A, b)$ with respect to perturbations in A and b. Here, however, we are interested in the sensitivity not of a whole subspace but of only one vector in the space, namely, x_k. We also mention the papers of Greenbaum [6] and Strakoš [26] (see also [8]) who consider the sensitivity of CG iterates to changes in the eigenvalue distribution of A. A summary and more thorough bibliographies can be found in [15, 18, 19]. One important aim of such work is to understand how rounding errors in finite precision arithmetic affect the behaviour of the algorithm. Here our motivation is different: we are interested in applications in which b is an observation vector greatly contaminated by noise, and in which very few iterations of CG are performed. In this setting, a sensitivity analysis of x_k with respect to perturbations in b is certainly relevant.

The rest of this paper is organized as follows. In Section 2 we introduce the Lanczos [14] and CG algorithms. In Section 3 we obtain expressions for J_k, the Jacobian of x_k with respect to b. We also give bounds on the normwise relative error between J_k and A^{-1} and on the spectral norm condition number $\|J_k\|_2$. We discuss methods to compute or estimate $J_k v$ and $J_k^T v$ for a given vector v in Section 4. In Section 5 we present numerical experiments to illustrate the theory, and we conclude with a discussion in Section 6.

2 The Lanczos and Conjugate Gradients algorithms

We start by reviewing some known facts about the Lanczos algorithm and its relation to CG. Unless otherwise stated we assume exact arithmetic. The effects of rounding errors in floating point arithmetic are discussed briefly in Sections 5.2 and 6. A more thorough treatment of these topics, including implementation details, can be found, e.g., in the monographs [2, 4, 7, 12, 15, 18].

The Lanczos algorithm computes an orthogonal tridiagonalization of the symmetric matrix $A \in \Re^{n\times n}$ column by column, starting from an arbitrary normalized vector v_1. After k steps the algorithm produces $V_k = \begin{bmatrix} v_1, \dots v_k \end{bmatrix} \in \Re^{n\times k}$ with orthonormal columns and

$$T_k = \begin{bmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_k \\ & & \beta_k & \alpha_k \end{bmatrix}, \qquad \widetilde{T}_k = \begin{bmatrix} T_k \\ \beta_{k+1} e_k^T \end{bmatrix}, \tag{2}$$

such that

$$AV_k = V_k T_k + \beta_{k+1} v_{k+1} e_k^T = V_{k+1} \widetilde{T}_k. \tag{3}$$

(Here e_k is the k–th standard basis vector, not to be confused with the error ϵ_k below.) The columns of V_k form an orthonormal basis for the Krylov subspace

$$\mathcal{K}_k(A, v_1) = \text{span}\{v_1, Av_1, \dots, A^{k-1}v_1\}.$$

Starting from an initial guess x_0 with residual $r_0 = b - Ax_0$, CG produces a sequence of iterates satisfying

$$\begin{aligned} &x_k \in x_0 + \mathcal{K}_k(A, r_0), \\ &r_k = b - Ax_k \in A\mathcal{K}_k(A, r_0), \qquad r_k \perp \mathcal{K}_k(A, r_0). \end{aligned}$$

Define $\beta_1 = \|r_0\|_2$ and start the Lanczos algorithm with $v_1 = r_0/\beta_1$. Then the iterate x_k and residual r_k from CG can be written as

$$\begin{aligned} x_k &= x_0 + V_k T_k^{-1} \beta_1 e_1 \\ r_k &= r_0 - AV_k T_k^{-1} \beta_1 e_1 = -\beta_1 \beta_{k+1} e_k^T T_k^{-1} e_1 v_{k+1}. \end{aligned} \tag{4}$$

We will also use the fact that $\mathcal{K}_k(A, r_0) = \text{Range}(V_k) = \text{Range}(K_k)$, where

$$K_k = \begin{bmatrix} r_0, \dots, A^{k-1} r_0 \end{bmatrix} = \begin{bmatrix} A\epsilon_0, \dots, A^k \epsilon_0 \end{bmatrix} \tag{5}$$

and $\epsilon_0 = A^{-1}b - x_0$ is the initial error.

It can also be useful to think of CG in terms of polynomials:

$$x_k = x_0 + \zeta_{k-1}(A) r_0, \tag{6}$$

where $\zeta_{k-1}(A)$ is a polynomial of degree at most $k-1$. Then the error $\epsilon_k = A^{-1}b - x_k$ and the residual $r_k = b - Ax_k$ satisfy

$$\epsilon_k = \rho_k(A)\epsilon_0, \qquad r_k = \rho_k(A) r_0, \qquad \rho_k(A) = I - A\zeta_{k-1}(A). \tag{7}$$

Below we outline some properties of the polynomial ρ_k to be used in later sections. Let Π_k denote the set of polynomials of degree at most k. It is well known that

$$\|\epsilon_k\|_A = \|\rho_k(A)\epsilon_0\|_A = \min_{\substack{\rho\in\Pi_k \\ \rho(0)=1}} \|\rho(A)\epsilon_0\|_A. \tag{8}$$

Let $\mu_k^{(j)}$, $j = 1, \ldots, k$, denote the eigenvalues of T_k, known as Ritz values. These are the roots of ρ_k (see, e.g., [4, §2.4], [25, §2]) which can therefore be written in the form

$$\rho_k(\lambda) = \prod_{j=1}^{k} \left(1 - \frac{\lambda}{\mu_k^{(j)}}\right). \tag{9}$$

In the following lemma we give another characterization of ρ_k. Similar ideas are used in [5, §1] and [17, §6].

Lemma 2.1. *Let the spectral decomposition of A be*

$$A = Q\Lambda Q^T, \quad Q^{-1} = Q^T, \quad \Lambda = \operatorname{diag}(\lambda_i), \quad 0 < \lambda_1 \le \cdots \le \lambda_n,$$

and define

$$L_k = \begin{bmatrix} \lambda_1 & \ldots & \lambda_1^k \\ \vdots & & \vdots \\ \lambda_n & \ldots & \lambda_n^k \end{bmatrix}, \quad w = \Lambda^{1/2} Q^T \epsilon_0, \quad W = \operatorname{diag}(w_i), \quad e = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}. \tag{10}$$

Write the polynomial ρ_k in (7) as

$$\rho_k(\lambda) = 1 + \sum_{i=1}^{k} \tau_i \lambda^i, \qquad t_k = [\tau_1, \ldots, \tau_k]^T. \tag{11}$$

Then provided CG has not reached the exact solution $A^{-1}b$ before step k, the matrix $L_k^T W^2 L_k$ is non-singular and the vector of coefficients t_k satisfies

$$t_k = \arg\min_t \|W(e + L_k t)\|_2^2 = -(L_k^T W^2 L_k)^{-1} L_k^T W^2 e. \tag{12}$$

Proof. If the matrix $L_k^T W^2 L_k$ is singular there exists a vector $y = [\gamma_1, \ldots, \gamma_k]^T \neq 0$ such that $WL_k y = 0$. Because W is diagonal, this implies, for all indices $i \in [1, k]$, either $w_i = 0$ or

$$\gamma_1 \lambda_i + \gamma_2 \lambda_i^2 + \cdots + \gamma_k \lambda_i^k = 0.$$

Let j denote the smallest index such that $\gamma_j \neq 0$, and without loss of generality scale y such that $\gamma_j = 1$. Then either $w_i = 0$ or

$$1 + \gamma_{j+1} \lambda_i + \cdots + \gamma_k \lambda_i^{k-j} = 0.$$

In other words, there is a polynomial $\tilde{\rho}_{k-j}$ of degree at most $k - j$ such that $\tilde{\rho}_{k-j}(0) = 1$ and $\tilde{\rho}_{k-j}(\lambda_i) = 0$ for all i satisfying $w_i \neq 0$. Note from the definition of w in (10) that $w_i = 0$ implies $e_i^T Q \epsilon_0 = 0$. Thus, we have

$$\tilde{\rho}_{k-j}(A)\epsilon_0 = Q\tilde{\rho}_{k-j}(\Lambda)Q^T\epsilon_0 = 0.$$

The above and (8) implies that $\epsilon_{k-j} = 0$, i.e., $x_{k-j} = A^{-1}b$. Therefore, so long as CG has not reached the exact solution, the matrix $L_k^T W^2 L_k$ cannot be singular.

Now from (8),

$$\|\rho_k(A)\epsilon_0\|_A = \min_{\substack{\rho\in\Pi_k \\ \rho(0)=1}} \|\rho(\Lambda)w\|_2^2 = \min_{\substack{\rho\in\Pi_k \\ \rho(0)=1}} \|W\rho(\Lambda)e\|_2^2 = \min_t \|W(e + L_k t)\|_2^2.$$

In other words, the vector of coefficients t_k in (11) is the solution of a weighted linear least-squares problem:

$$t_k = \arg\min_t \|W(e + L_k t)\|_2^2 = -(L_k^T W^2 L_k)^{-1} L_k^T W^2 e.$$

□

Lemma 2.1 gives a convenient expression for the coefficients of the polynomial ρ_k in (7) in terms of the eigenvalues and eigenvectors of A and the initial error ϵ_0. We use this expression to obtain our main result in the next section.

3 Properties of the Jacobian matrix

Let x_k denote the k–th iterate of CG applied to $Ax = b$. Throughout we assume that x_k is well-defined, that is, CG has not reached the exact solution $A^{-1}b$ prior to step k.

3.1 Expressions for J_k

First, we obtain the matrix of partial derivatives of t_k defined in (11) with respect to b. In the proof we use the following notation. For a differentiable matrix function

$$X : \begin{cases} \Re^n \to \Re^{p\times q}, \\ b \mapsto X(b), \end{cases}$$

$\partial_b X : \Re^n \to \Re^{p\times q}$ is the linear operator defined by

$$X(b + \Delta b) = X(b) + \partial_b X(b) \cdot \Delta b + o\left(\|\Delta b\|_2\right).$$

The matrix representation of $\partial_b X$ in the standard basis in $\Re^{pq\times n}$ is the Jacobian matrix $\partial\big(\mathrm{vec}(X)\big)/\partial b$. For a detailed introduction to matrix differential calculus we recommend [16, Chapter 5].

Lemma 3.1. *Let t_k be the vector of coefficients defined in* (11)*. In the notation of Lemma 2.1,*

$$\frac{\partial t_k}{\partial b} = -2(WL_k)^\dagger \rho_k(\Lambda)\Lambda^{-1/2}Q^T. \tag{13}$$

Proof. Recall from Lemma 2.1 that

$$t_k = \arg\min_t \|W(e + L_k t)\|_2^2 = -(L_k^T W^2 L_k)^{-1} L_k^T W^2 e, \tag{14}$$

where only W in the above right-hand side depends on b. Denote $M_k = L_k^T W^2 L_k$. Then for any $\Delta b \in \Re^n$,

$$\partial_b(M_k^{-1}) \cdot \Delta b = -M_k^{-1}(\partial_b M_k \cdot \Delta b)M_k^{-1} = -M_k^{-1} L_k^T \big(\partial_b W^2 \cdot \Delta b\big) L_k M_k^{-1}.$$

Furthermore, because W is diagonal, $\partial_b W^2 = 2W\partial_b(W)$. Therefore, from (14),

$$\begin{aligned}\partial_b t_k \cdot \Delta b &= M_k^{-1}L_k^T\big(\partial_b W^2 \cdot \Delta b\big)L_k M_k^{-1}L_k^T W^2 e - M_k^{-1}L_k^T\big(\partial_b W^2 \cdot \Delta b\big)e \\ &= 2M_k^{-1}L_k^T W(\partial_b W \cdot \Delta b)(L_k M_k^{-1}L_k^T W^2 e - e) \\ &= -2(WL_k)^\dagger(\partial_b W \cdot \Delta b)(L_k t_k + e).\end{aligned}$$

Next, noticing that $(L_k t_k + e) = \rho_k(\Lambda)e$ and using the fact that $\partial_b W \cdot \Delta b$ is diagonal, we obtain

$$\begin{aligned}\partial_b t_k \cdot \Delta b &= -2(WL_k)^\dagger\rho_k(\Lambda)(\partial_b W \cdot \Delta b)e \\ &= -2(WL_k)^\dagger\rho_k(\Lambda)(\partial_b w \cdot \Delta b) \\ &= -2(WL_k)^\dagger\rho_k(\Lambda)\Lambda^{-1/2}Q^T\partial_b b \cdot \Delta b.\end{aligned}$$

The last equality in the above follows from the fact that

$$w = \Lambda^{1/2}Q^T\epsilon_0 = \Lambda^{-1/2}Q^T b - \Lambda^{1/2}Q^T x_0,$$

see (10), so $\partial_b w = \Lambda^{-1/2}Q^T\partial_b b$. Since $\partial_b b$ is the identity operator, the above implies that $\partial t_k/\partial b$, the matrix representation of $\partial_b t_k$, is given by (13). □

Using the above Lemma we can now obtain our main result. The following theorem gives two equivalent expressions for J_k, the Jacobian of x_k with respect to b, in terms of the matrices V_k and T_k and the polynomials ρ_k and ζ_{k-1} defined in Section 2.

Theorem 3.1. *Let x_k be the k–th iterate of CG applied to $Ax = b$ starting from x_0. In the notation of Section 2,*

$$J_k = A^{-1}\big[I - \rho_k(A)\big] + 2V_k T_k^{-1}V_k^T\rho_k(A). \tag{15}$$

Equivalently,

$$J_k = 2V_k T_k^{-1}V_k^T + (I - 2V_k T_k^{-1}V_k^T A)\zeta_{k-1}(A). \tag{16}$$

Proof. Because $x_k = A^{-1}b - \epsilon_k$, we have $\frac{\partial x_k}{\partial b} = A^{-1} - \frac{\partial \epsilon_k}{\partial b}$. Then with K_k defined in (5) and t_k in (11) we obtain

$$\begin{aligned}\epsilon_k = \rho_k(A)\epsilon_0 &= \left(I + \sum_{i=1}^{k}\tau_i A^i\right)\epsilon_0, \\ \frac{\partial \epsilon_k}{\partial b} = \rho_k(A)\frac{\partial \epsilon_0}{\partial b} + \sum_{i=1}^{k}A^i\epsilon_0\frac{\partial \tau_i}{\partial b} &= \rho_k(A)A^{-1} + K_k\frac{\partial t_k}{\partial b},\end{aligned}$$

where we have used the fact that $\epsilon_0 = A^{-1}b - x_0$, so that $\frac{\partial \epsilon_0}{\partial b} = A^{-1}$. Thus,

$$\frac{\partial x_k}{\partial b} = A^{-1}\big[I - \rho_k(A)\big] - K_k\frac{\partial t_k}{\partial b}. \tag{17}$$

Note that

$$K_k = \big[A\epsilon_0, \ldots, A^k\epsilon_0\big] = Q\Lambda^{-1/2}\left[\Lambda w, \Lambda^2 w, \ldots \Lambda^k w\right] = Q\Lambda^{-1/2}WL_k. \tag{18}$$

From this and Theorem 3.1 we obtain

$$K_k \frac{\partial t_k}{\partial b} = -2Q\Lambda^{-1/2}(WL_k)(WL_k)^\dagger \rho_k(\Lambda)\Lambda^{-1/2}Q^T. \tag{19}$$

Using (18), $\text{Range}(V_k) = \text{Range}(K_k)$, and $V_k^T A V_k = T_k$, we obtain

$$(WL_k)(WL_k)^\dagger = \Lambda^{1/2}Q^T K_k (K_k^T A K_k)^{-1} K_k^T Q \Lambda^{1/2} = \Lambda^{1/2}Q^T V_k T_k^{-1} V_k^T Q \Lambda^{1/2}.$$

Thus,

$$K_k \frac{\partial t_k}{\partial b} = -2V_k T_k^{-1} V_k^T \rho_k(A).$$

We obtain (15) from the above and (17). Equation (16) follows from (15) and the relationship (7) between the polynomials ρ_k and ζ_{k-1}. □

The formula for J_k given in (15) can be used to derive relationships between J_k and A^{-1}. (See Corollaries 3.1 and 3.2 below.) The expression in (16) is useful for understanding the relationship between the condition number $\|J_k\|_2$ and $\|T_k^{-1}\|_2$, as shown in Corollary 3.3.

3.2 Relationship to A^{-1}

For the exact solution $x(b) = A^{-1}b$, the Jacobian is simply $\partial x/\partial b = A^{-1}$. Therefore, intuitively, we might expect that J_k approaches A^{-1} as k increases. However, this is not always the case. The following corollary bounds the normwise relative error between J_k and A^{-1}.

Corollary 3.1. *In the notation of Theorem 3.1,*

$$\frac{\|J_k - A^{-1}\|_2}{\|A^{-1}\|_2} \le 3\|\rho_k(A)\|_2. \tag{20}$$

Furthermore, if $\|\rho_k(A)\|_2 < 1$,

$$\frac{|r_k^T J_k r_k|}{1 + \|\rho_k(A)\|_2} \le \|\epsilon_k\|_A^2 \le \frac{|r_k^T J_k r_k|}{1 - \|\rho_k(A)\|_2}. \tag{21}$$

Proof. From (15) we have

$$J_k - A^{-1} = \big(-A^{-1} + 2V_k T_k^{-1} V_k^T \big)\rho_k(A).$$

The relationship (20) follows by takings norms and using the fact that $\|V_k T_k^{-1} V_k^T\|_2 = \|T_k^{-1}\|_2 \le \|A^{-1}\|_2$. Recall from Section 2 that $r_k^T V_k = 0$. Therefore,

$$\|\epsilon_k\|_A^2 = r_k^T A^{-1} r_k = r_k^T \big(J_k + A^{-1}\rho_k(A)\big) r_k = r_k^T J_k r_k + r_k^T A^{-1/2}\rho_k(A)A^{-1/2} r_k,$$

from which the inequalities in (21) follow. □

We can interpret Corollary 3.1 as follows. If $\|\rho_k(A)\|_2 \ll 1$ then the Jacobian J_k is close to A^{-1} and the energy norm of the error is close to $(r_k^T J_k r_k)^{1/2}$. From (9),

$$\|\rho_k(A)\|_2 = \max_i \prod_{j=1}^{k} \left| 1 - \frac{\lambda_i}{\mu_k^{(j)}} \right|.$$

Therefore, if all eigenvalues of A are well approximated by a Ritz value, $\|\rho_k(A)\|_2 \ll 1$. Alternatively, from the characterization of ρ_k in (8),

$$\|\rho_k(A)\|_2 \geq \frac{\|\rho_k(A)\epsilon_0\|_*}{\|\epsilon_0\|_*} = \frac{\|\epsilon_k\|_*}{\|\epsilon_0\|_*},$$

where $\|\cdot\|_*$ can denote either the A-norm, the 2-norm, or the A^2-norm, i.e., the residual 2-norm. If ϵ_0 has a significant component along the eigenvector of A corresponding to the largest in magnitude eigenvalue of $\rho_k(A)$, the above lower bound is a reasonable approximation. If this is the case, and if $\|\epsilon_k\|_*/\|\epsilon_0\|_* \ll 1$, then $\|\rho_k(A)\|_2 \ll 1$. Of course, it is certainly possible to construct examples in which $\|\rho_k(A)\|_2 \gg 1$. One such example is given in Section 5.

Because $\|\rho_k(A)\|_2$ can be much larger than 1, and because the quantity $J_k r_k$ is expensive to compute (see Section 4), it is doubtful whether (21) can be used as a reliable and efficient way to estimate the energy norm of the error. Nevertheless, we report (21) as it gives one way to characterize the relationship between J_k and A^{-1}.

Note from (8) and (15) that $\rho_k(A) = 0$ implies both $x_k = A^{-1}b$ and $J_k = A^{-1}$. However, it may be the case that $J_k \neq A^{-1}$ when $x_k = A^{-1}b$ if $\epsilon_k = \rho_k(A)\epsilon_0 = 0$ but $\rho_k(A) \neq 0$. In other words, if for a specific b we have obtained the exact solution in $k < n$ steps, it does not necessarily follow that $x_k(b) = A^{-1}b$ for all b. Hence, we cannot conclude that $\partial x_k/\partial b = A^{-1}$. The following corollary gives an expression for J_k when CG reaches the exact solution, that is, when for a specific b we have $x_k = A^{-1}b$.

Corollary 3.2. *Suppose that CG reaches the exact solution at step k. Then*

$$J_k = A^{-1}\big[I - \rho_k(A)\big] = A^{-1} - Q \begin{bmatrix} 0 & \\ & \Lambda_2^{-1}\rho_k(\Lambda_2) \end{bmatrix} Q^T, \tag{22}$$

where the spectral decomposition of A is

$$A = Q\Lambda Q^T = [Q_1, Q_2] \begin{bmatrix} \Lambda_1 & \\ & \Lambda_2 \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix},$$

and Λ_1 consists of those eigenvalues of A to which a Ritz value has converged. If the exact solution is only obtained at step n, then $J_n = A^{-1}$.

Proof. Let $T_k = \bar{Q}_k \bar{\Lambda}_k \bar{Q}_k^T$ be the spectral decomposition of T_k. It is known that $x_k = A^{-1}b$ implies $AV_k = V_k T_k$ in (3), so that $Q_1 = V_k \bar{Q}_k$, $\Lambda_1 = \bar{\Lambda}_k$, and $\rho_k(\Lambda_1) = \rho_k(T_k) = 0$. Thus in (15) we have

$$V_k^T \rho_k(A) = V_k^T [V_k \bar{Q}_k, Q_2] \begin{bmatrix} \rho_k(\Lambda_1) & \\ & \rho_k(\Lambda_2) \end{bmatrix} Q^T = [\bar{Q}_k, 0] \begin{bmatrix} 0 & \\ & \rho_k(\Lambda_2) \end{bmatrix} Q^T = 0,$$

from which (22) follows. If this only occurs at step n, the block $\Lambda_1 = \bar{\Lambda}_n$ has dimension n and the block Λ_2 is inexistant. In other words, $\rho_n(A) = 0$ and thus $J_n = A^{-1}$. □

3.3 The condition number of x_k

Recall from the introduction that a useful scalar measure of the sensitivity of x_k to perturbations in b is its spectral norm condition number, $\|J_k\|_2$. From Theorem 3.1 we can obtain bounds on $\|J_k\|_2$, as shown below.

Corollary 3.3. *In the notation of Theorem 3.1,*

$$\|T_k^{-1}e_1\|_2 \leq \|J_k\|_2 \leq 2\|T_k^{-1}\|_2 + \big(1 + 2\|A\|_2\|T_k^{-1}\|_2\big)\|\zeta_{k-1}(A)\|_2.$$

Proof. For the lower bound, from (15) and (7) we have

$$\|J_k\|_2 \geq \frac{\|J_k r_0\|_2}{\|r_0\|_2} = \frac{\big\|\zeta_{k-1}(A)r_0 + 2V_k T_k^{-1} V_k^T \rho_k(A) r_0\big\|_2}{\|r_0\|_2}.$$

Recall that $V_k^T \rho_k(A) r_0 = V_k^T r_k = 0$. Furthermore, using (6), (4), and the fact that $\beta_1 = \|r_0\|_2$ we have

$$\frac{\|J_k r_0\|_2}{\|r_0\|_2} = \frac{\|\zeta_{k-1}(A)r_0\|_2}{\|r_0\|_2} = \|T_k^{-1}e_1\|_2. \tag{23}$$

The upper bound is an immediate consequence of (16). □

If r_0 has a significant component along the eigenvector of A corresponding to the largest eigenvalue of $\zeta_{k-1}(A)$, then from (23) $\|\zeta_{k-1}(A)\|_2 \approx \|T_k^{-1}e_1\|_2$. If this is the case, Corollary 3.3 shows that both the lower bound and upper bound on the spectral norm of the Jacobian depend only on terms involving T_k^{-1} (as opposed to A^{-1}). In such cases, the spectral norm condition number of x_k is essentially determined by the reciprocal of the smallest Ritz value. Of course, this reasoning does not hold in the worst case, and it may happen that $\|J_k\|_2 \gg \|T_k^{-1}\|_2$. Numerical examples are given in Section 5.

4 Computing matrix-vector products

In most practical applications, due to memory limitations it is clearly not possible to explicitly compute and store the entire matrix $J_k \in \Re^{n\times n}$. Recall from the introduction that quantities of interest are $\|J_k\|_2$ as well as $v^T J_k \Sigma J_k^T v$ for given Σ and v. In this section we briefly review how matrix-vector products (matvecs) $J_k v$ and $J_k^T v$ can be computed or estimated using automatic differentiation techniques. (See for example [21, §7.2] or [9] for an introduction to automatic differentiation.) This can be used to estimate the spectral norm and Frobenius norm condition numbers of x_k, as follows: for any v of unit length we have a lower bound $\|J_k\|_2 \geq \|J_k v\|_2$, while if $v \sim (0, I)$ then $\|J_k v\|_2^2$ is an unbiased estimator of $\|J_k\|_F^2$.

Algorithm 1 defines the standard CG iterations, presented essentially as in [10]. We can recast one full iteration of CG in terms of the action of operators. First we replace the scalars μ_{k-1} and

Algorithm 1 The standard CG iterations

1: Given A, b, and x_0
2: $r_0 = b - Ax_0$
3: $p_0 = r_0$
4: $k = 1$
5: **while** stopping criterion not satisfied **do**
6: $\quad \mu_{k-1} = r_{k-1}^T r_{k-1} \,/\, p_{k-1}^T A p_{k-1}$
7: $\quad x_k = x_{k-1} + \mu_{k-1} p_{k-1}$
8: $\quad r_k = r_{k-1} - \mu_{k-1} A p_{k-1}$
9: $\quad \nu_k = r_k^T r_k \,/\, r_{k-1}^T r_{k-1}$
10: $\quad p_k = r_k + \nu_k p_{k-1}$
11: $\quad k = k + 1$
12: **end while**
13: **end**

ν_k by their corresponding expressions

$$\begin{aligned}
\mu_{k-1} &= \frac{r_{k-1}^T r_{k-1}}{p_{k-1}^T A p_{k-1}}, \\
\nu_k &= \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}} = \frac{(r_{k-1} - \mu_{k-1} A p_{k-1})^T (r_{k-1} - \mu_{k-1} A p_{k-1})}{r_{k-1}^T r_{k-1}} \\
&= 1 - 2\frac{p_{k-1}^T A r_{k-1}}{p_{k-1}^T A p_{k-1}} + \frac{(r_{k-1}^T r_{k-1})(p_{k-1}^T A^2 p_{k-1})}{(p_{k-1}^T A p_{k-1})^2}.
\end{aligned}$$

Define the vector $z_k = [x_k^T, r_k^T, p_k^T]^T \in \Re^{3n}$. For the initial step (lines 2–4) we have

$$z_0 = F_0(b) = \begin{bmatrix} x_0 \\ b - Ax_0 \\ b - Ax_0 \end{bmatrix}, \qquad \frac{\partial z_0}{\partial b} = \begin{bmatrix} 0 \\ I_n \\ I_n \end{bmatrix}.$$

At step k of CG (lines 6–8) we have

$$z_k = F\left(z_{k-1}\right) = \left[\begin{array}{l} x + \frac{r^T r}{p^T A p} p \\ r - \frac{r^T r}{p^T A p} A p \\ r - \frac{r^T r}{p^T A p} A p + \left(1 - 2\frac{p^T A r}{p^T A p} + \frac{(r^T r)(p^T A^2 p)}{(p^T A p)^2}\right) p \end{array} \right]_{k-1}.$$

Differentiating the above, we obtain

$$\frac{\partial z_k}{\partial z_{k-1}} = \left[\begin{array}{lll} I_n & \frac{2}{\theta} p r^T & \mu I_n - 2\frac{\mu}{\theta} p q^T \\ 0_n & I_n - \frac{2}{\theta} q r^T & -\mu A + 2\frac{\mu}{\theta} q q^T \\ 0_n & I_n - \frac{2}{\theta} q r^T - \frac{2}{\theta} p q^T + \frac{2\tau}{\theta^2} p r^T & G \end{array} \right]_{k-1}$$

where $q = Ap$, $\mu = \frac{r^T r}{p^T q}$, $\theta = p^T q$, $\tau = q^T q$, and

$$G = -\mu A + \frac{2\mu}{\theta} q q^T + \left(1 - \frac{2q^T r}{\theta} + \frac{\mu\tau}{\theta}\right) I_n \\ - \frac{2}{\theta} p r^T A + \frac{4q^T r}{\theta^2} p q^T + \frac{2\mu}{\theta} p q^T A - \frac{4\mu\tau}{\theta^2} p q^T.$$

Applying the chain rule to successive iterations we have

$$J_k = \frac{\partial x_k}{\partial b} = [I_n, 0, 0] \frac{\partial z_k}{\partial b} = [I_n, 0, 0] \left(\frac{\partial z_k}{\partial z_{k-1}}\right) \cdots \left(\frac{\partial z_1}{\partial z_0}\right) \left(\frac{\partial z_0}{\partial b}\right).$$

For any vector v, one can compute

$$J_k v = [I_n, 0, 0] \left(\frac{\partial z_k}{\partial z_{k-1}}\right) \cdots \left(\frac{\partial z_1}{\partial z_0}\right) \left(\frac{\partial z_0}{\partial b}\right) v$$

on the fly by updating $v \leftarrow \left(\frac{\partial z_k}{\partial z_{k-1}}\right) v$ at step k of CG. This is known as the "forward" or "direct" mode in automatic differentiation. The cost of this operation is one matvec with A as well as $\sim 18n$ flops for each update step. In comparison, one step of CG requires one matvec with A and $\sim 10n$ flops. Additionally, the quantities $(\partial *_k / \partial b)v$ must be stored, where $*_k$ denotes every variable in Algorithm 1. Thus, acquiring sensitivity information in the form of $J_k v$ has a very significant, but not prohibitive, computational cost, which some users are willing to pay [20, 28].

Computing

$$J_k^T v = \left(\frac{\partial z_0}{\partial b}\right)^T \left(\frac{\partial z_1}{\partial z_0}\right)^T \cdots \left(\frac{\partial z_k}{\partial z_{k-1}}\right)^T [I_n, 0, 0]^T v,$$

cannot be done on the fly, since

$$\left(\frac{\partial z_i}{\partial z_{i-1}}\right)^T \cdots \left(\frac{\partial z_k}{\partial z_{k-1}}\right)^T [I_n, 0, 0]^T v,$$

is not know at step $i < k$ of CG. One has to store (or recompute) all the quantities involved at every step in CG and loop through the algorithm in reverse order a posteriori. This is known as the "reverse" or "adjoint" mode.

We have attempted to find more efficient methods to compute or estimate $J_k v$ and/or $J_k^T v$ using the recurrences for the polynomials $\zeta_{k-1}(A)$ and the formula (16) given in Section 3. In our experience, however, the most accurate method of computing matvecs with J_k is using the automatic differentiation techniques summarized above. Numerical examples are given in the next section.

5 Numerical experiments

We provide some numerical experiments merely to illustrate the theory developed in the previous sections. For real-world, large-scale data assimilation applications in which the linearization has been incorporated for sensitivity analyses, we refer to [20, 28].

5.1 Behaviour of J_k and $\|J_k\|_2$

First we illustrate the relationship between J_k and A^{-1} from Corollary 3.1, as well as the relationship between the condition number $\|J_k\|_2$ and $\|T_k^{-1}\|_2$ from Corollary 3.3, as k increases.

In these small examples we explicitly compute and store J_k using the automatic differentiation techniques described in Section 4. To simulate exact arithmetic we run CG with double reorthogonalization (of the residual vectors). When we reorthogonalize in CG we also differentiate the reorthogonalization steps of the algorithm, i.e., the step

$$r_k \leftarrow r_k - \frac{r_j r_j^T}{r_j^T r_j} r_k, \qquad j = 1, \dots k-1,$$

leads to the update

$$\frac{\partial r_k}{\partial b} \leftarrow \frac{\partial r_k}{\partial b} - \frac{r_j r_j^T}{r_j^T r_j}\frac{\partial r_k}{\partial b} - \frac{r_j^T r_k}{r_j^T r_j}\frac{\partial r_j}{\partial b} - \frac{r_j r_k^T}{r_j^T r_j}\frac{\partial r_j}{\partial b} + 2\frac{r_j^T r_k}{(r_j^T r_j)^2} r_j r_j^T \frac{\partial r_j}{\partial b}.$$

We use the known relationships

$$\alpha_1 = \frac{1}{\mu_0}, \quad \alpha_i = \frac{1}{\mu_{i-1}} + \frac{\nu_{i-1}}{\mu_{i-2}}, \quad \beta_i = \frac{\sqrt{\nu_{i-1}}}{\mu_{i-2}}, \quad i = 2, 3, \dots, k$$

to obtain the entries of T_k, from which we can then easily compute $\|T_k^{-1}\|_2$.

Example 1 is meant to illustrate extreme, but highly unlikely, behaviour of $\|J_k\|_2$. In this example A is a diagonal 64×64 matrix with 32 eigenvalues equally spaced in $[10^{-1}, 10^0]$ and 32 eigenvalues equally spaced in $[10^1, 10^2]$. The right-hand side vector is $b = \begin{bmatrix}1, \dots 1, 0, \dots, 0\end{bmatrix}^T$ and the iteration is started with $x_0 = 0$. Because r_0 is orthogonal to all eigenvectors corresponding to the eigenvalues in $[10^1, 10^2]$, the Lanczos algorithm fails to compute any Ritz value in this interval. Consequently, for all k,

$$\begin{aligned} \|\rho_k(\Lambda)\|_2 \geq |\rho_k(\lambda_n)| &= \prod_{j=1}^{k} \left|1 - \frac{\lambda_n}{\mu_k^{(j)}}\right| \geq \prod_{j=1}^{k} \left(\frac{10^2}{10^0} - 1\right) \approx 10^{2k}, \\ \|\zeta_{k-1}(\Lambda)\|_2 \geq |\zeta_{k-1}(\lambda_n)| &= \frac{1}{\lambda_n}|1 - \rho_k(\lambda_n)| \approx 10^{2k-2}. \end{aligned} \tag{24}$$

In other words, the condition number of x_k grows very quickly as k increases, and the iterates are soon highly sensitive to perturbations in b. This is illustrated in Figure 1.

The extreme behaviour in the above example is a trivial consequence resulting from a very specific choice of b. Examples 2a and 2b below illustrate more typical behaviour of $\|J_k\|_2$ and the relative error $\eta_k = \|J_k - A^{-1}\|_2 / \|A^{-1}\|_2$. By more typical we mean a situation in which ϵ_0 is *not* orthogonal to eigenvectors of A corresponding to large eigenvalues. In each case A is formed via its spectral decomposition. The matrix of eigenvectors is the Q factor in the QR decomposition of a random matrix, $b = e$, and $x_0 = 0$. In example 2a the eigenvalues of A are logarithmically equally spaced between 10^{-4} and 1, while in example 2b there are $n-1$ eigenvalues of A linearly equally spaced between 1 and 10 with one extra eigenvalue 10^{-7}.

Results are plotted in Figure 2. In each case, $\|J_k\|_2$ behaves essentially as $\|T_k^{-1}\|_2$, which can be much smaller than $\|A^{-1}\|_2$ in the early iterations. In example 2a, η_k behaves essentially as $\|\epsilon_k\|_A / \|\epsilon_0\|_A$. In example 2b, η_k is bounded above by $\|\epsilon_k\|_A / \|\epsilon_0\|_A$ until it reaches its maximum attainable accuracy.

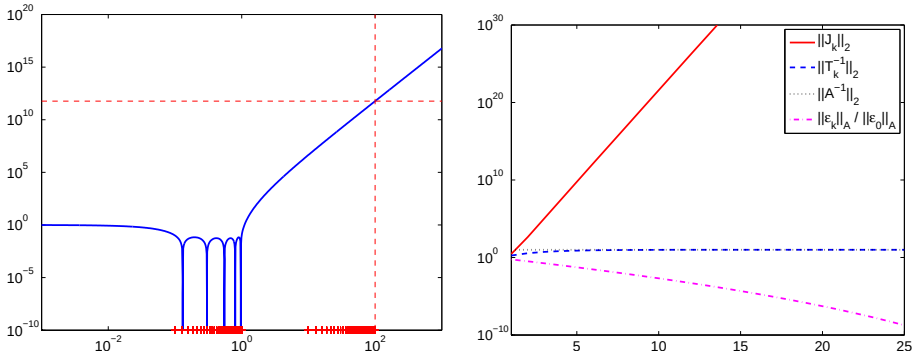


Figure 1: EXAMPLE 1. Left: At a fixed iteration $k = 5$, the magnitude of the polynomial $|\rho_5(\lambda)|$ plotted versus λ. The crosses on the horizontal axis represent the eigenvalues of A. The horizontal line is $\|\rho_5(A)\|_2 = \max_i |\rho_5(\lambda_i)| = |\rho_5(\lambda_n)|$ and the vertical line shows the corresponding eigenvalue λ_n. Right: the condition number $\|J_k\|_2$ plotted versus k. As predicted by (24), $\|J_k\|_2$ increases with a slope close to 2 on the semilog plot.

5.2 Effects of finite precision arithmetic

It is well known that properties of CG derived assuming exact arithmetic, in particular, orthogonality of the residual vectors, no longer hold when the algorithm is run in finite precision arithmetic. This can greatly affect the behaviour of the algorithm when it is run in finite precision arithmetic.

The relationships in Section 3 were derived using the connection between the CG and Lanczos algorithms, assuming exact arithmetic. Here we wish to verify to what extent these relationships continue to hold when the algorithm is run in floating point arithmetic. We repeat the tests used to produce Figure 2, but without any reorthogonalization.

Results are given in Figure 3. When no reorthogonalization is used, the relative error between A^{-1} and J_k (as computed using automatic differentiation) can oscillate as k increases, particularly when $k \geq n$. (Recall, however, that we are usually interested in stopping the method after $k \ll n$ iterations.) In both examples, even for large k, the spectral norm condition number $\|J_k\|_2$ still behaves similarly to $\|T_k^{-1}\|_2$, modulo some slight oscillations.

5.3 Validity of a first-order analysis

As discussed in the introduction, CG is a highly nonlinear algorithm. Therefore, it is reasonable to question whether the linearization

$$x_k(b + \Delta b) \approx x_k(b) + J_k \Delta b \tag{25}$$

gives a meaningful estimate. From Taylor's theorem, this certainly must be the case for sufficiently small $\|\Delta b\|_2$. Here we provide numerical experiments to investigate how small is "sufficiently small".

We compute $x_k(b)$ and $x_k(b + \Delta b)$ for $k = 5$. We then compare the relative error $\|x_k(b + \Delta b) - x_k(b)\|_2 / \|\Delta b\|_2$ with the condition number $\|J_k\|_2$. (See (1) and (25).) We perform tests with perturbations Δb scaled to obtain various values of $\|\Delta b\|_2 / \|b\|_2 \in [10^{-14}, 1]$. For Δb we pick both

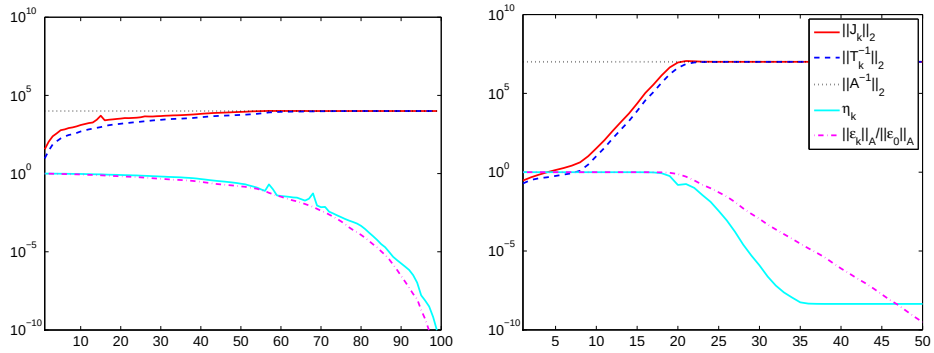


Figure 2: The condition number $\|J_k\|_2$ and relative error $\eta_k = \|J_k - A^{-1}\|_2/\|A^{-1}\|_2$ versus k for example 2a (left) and 2b (right) with double reorthogonalization.

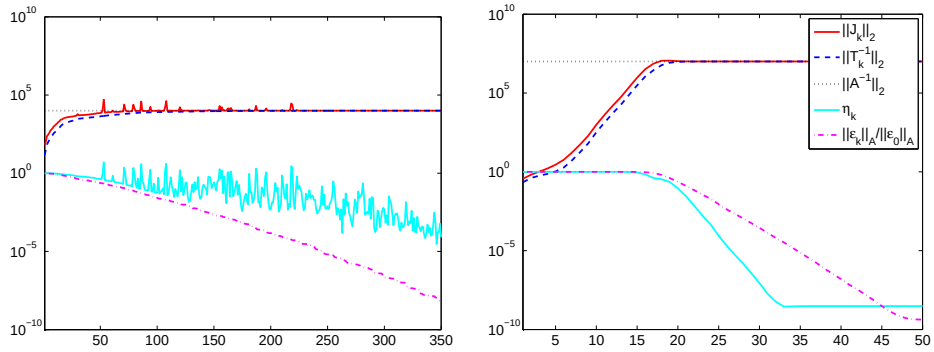


Figure 3: The condition number $\|J_k\|_2$ and relative error $\eta_k = \|J_k - A^{-1}\|_2/\|A^{-1}\|_2$ versus k for example 2a (left) and 2b (right) without any reorthogonalization. (Compare to Figure 2.)

(i) a random vector, and (ii) $\Delta b = v^*$, the right singular vector of J_k corresponding to its largest singular value, so that to first order

$$\frac{\|x_k(b + \Delta b) - x_k(b)\|_2}{\|\Delta b\|_2} \approx \frac{\|J_k \Delta b\|_2}{\|\Delta b\|_2} = \|J_k\|_2.$$

We use examples 1, 2a, and 2b described above, as well as example 3, a five-point finite difference discretization of the Laplacian on a 20×20 regular grid on $[-1, 1] \times [-1, 1]$. The resulting matrix A has dimension $n = 324$, b is set to $b = e$, and $x_0 = 0$.

Results are plotted in Figure 4. In example 1, the first-order analysis is descriptive of the true sensitivity of x_k only for tiny perturbations, for which $\|\Delta b\|_2/\|b\|_2$ is close to the unit roundoff. In this (pathological) example, CG is extremely nonlinear in b. In the remaining examples, however, the relative error is roughly constant with $\|\Delta b\|_2$, even when $\|\Delta b\|_2/\|b\|_2$ is much larger than the unit roundoff. (For $\|\Delta b\|_2/\|b\|_2$ less than roughly 10^{-1}, 10^{-2}, and 10^{-4} in examples 2a, 2b, and 3, respectively.) This indicates that, on these test problems, the first-order analysis is descriptive even

for fairly large values of $\|\Delta b\|_2/\|b\|_2$. These results were produced using double reorthogonalization. We have verified that the same phenomenon still holds when no reorthgonalization is performed, for $k = 5$ as well as with different choices of k.

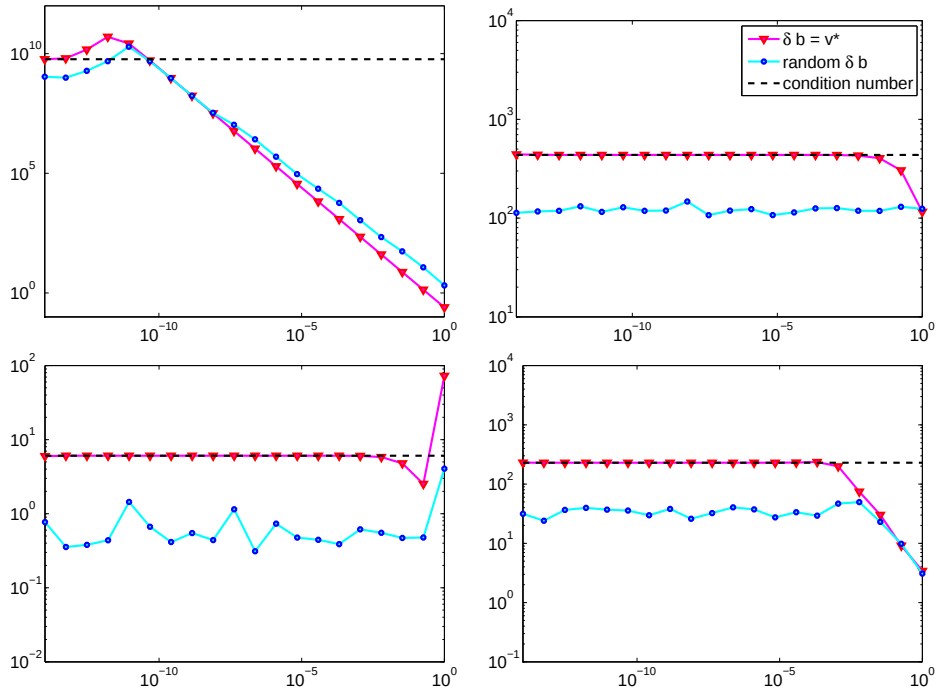


Figure 4: Relative error $\|x_k(b + \Delta b) - x_k(b)\|_2/\|\Delta b\|_2$ versus $\|\Delta b\|_2/\|b\|_2$ at iteration $k = 5$ for examples 1 (top left), 2a (top right), 2b (bottom left) and 3 (bottom right).

6 Discussion

We have performed a first-order perturbation analysis for CG iterates. Our results quantify, to first order, how sensitive the iterates are to perturbations in the right-hand side vector. In Theorem 3.1 we obtained an expression for the Jacobian of x_k in CG in terms of the matrices V_k and T_k from the Lanczos algorithm and the polynomials ρ_k and ζ_{k-1} in (7). We used this result to obtain bounds on the normwise relative error between J_k and A^{-1}, as well as bounds on the spectral norm condition number of x_k.

In our experience, automatic differentiation seems to be the most reliable way to compute $J_k v$ and $J_k^T v$. The cost of obtaining such sensitivity information is certainly significant. So far we have not found a way to compute $J_k v$ or $J_k^T v$ very accurately without performing k extra matvecs with A. A cheaper way to estimate J_k is $J_k \approx V_k T_k^{-1} V_k^T$. This is particularly efficient in data assimilation applications in which the overwhelming cost of the computation is that of matvecs with A. However, it may not give an accurate estimate of J_k (besides the norm, as demonstrated in Corollary 3.3). Whether or not better estimates can be obtained remains an open question.

In deriving Theorem 3.1 we have analyzed the Lanczos and CG algorithms assuming exact arithmetic. In finite precision the relationship (3) no longer holds. Let $V_k^{(c)}$, $T_k^{(c)}$, and $\widetilde{T}_k^{(c)}$ denote the matrices in (3) obtained when the algorithm is run in floating point arithmetic with machine unit roundoff u. Paige [22] showed that, under some mild assumptions,

$$AV_k^{(c)} = V_{k+1}^{(c)}\widetilde{T}_k^{(c)} + F_k, \qquad \|F_k\|_2 \leq \sqrt{k}(7\|A\|_2 + n\||A|\|_2)u + \mathcal{O}(u^2).$$

In the above, the columns of $V_k^{(c)}$ can quickly lose their orthogonality and even their linear independence. Paige [23, 24] has recently shown that there exists a matrix Q_k with orthonormal columns and $q_1 = [v_1^T, 0]^T$ such that

$$\left(\begin{bmatrix} A & \\ & T_k^{(c)} \end{bmatrix} + H_k\right) Q_k = Q_{k+1}\widetilde{T}_k^{(c)}, \qquad \|H_k\|_2 \leq u\|A\|_2 + \mathcal{O}(u^2).$$

In other words, the computed T_k is the tridiagonal matrix produced by the exact Lanczos process applied to a small perturbation of an augmented matrix $\mathrm{diag}(A, T_k^{(c)})$ started with the augmented vector $[v_1^T, 0]^T$. Paige called the above the augmented backward stability of the Lanczos algorithm. In the future we intend to use this result to analyze the true sensitivity of the Lanczos and CG algorithms implemented in floating point arithmetic. (By this we do not mean the difference between x_k and $\tilde{x}_k$, the iterates computed in exact and floating point arithmetic, respectively, but rather between $\tilde{x}_k(b)$ and $\tilde{x}_k(b + \Delta b)$.)

We are working on extending the ideas presented in this manuscript to minimum-residual and other polynomial-based iterative methods. For CG and other methods we can also consider other derivatives such as $\frac{\partial x_k}{\partial x_0}$, $\frac{\partial \|r_k\|_2}{\partial r_0}$, etc. It may also be possible to find sparse approximations of these derivatives. Another interesting and very challenging problem which we have not considered here is the computation of the derivative of x_k with respect to elements of A.

References

[1] J.-F. Carpraux, S. K. Godunov, and S. V. Kuznetsov. Condition number of the Krylov bases and subspaces. *Linear Algebra and its Applications*, 248(1):137–160, 1996.

[2] J. W. Demmel. *Applied numerical linear algebra*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.

[3] L. Eldén. Partial least-squares vs. Lanczos bidiagonalization—I: analysis of a projection method for multiple regression. *Computational Statistics & Data Analysis*, 46:11–31, 2004.

[4] B. Fischer. *Polynomial Based Iteration Methods for Symmetric Linear Systems*. SIAM, Philadelphia, USA, 2011.

[5] A. Greenbaum. Comparison of splittings used with the conjugate gradient algorithm. *Numerische Mathematik*, 33(1):181–194, 1979.

[6] A. Greenbaum. Behavior of slightly perturbed Lanczos and conjugate gradient recurrences. *Linear Algebra and its Applications*, 113(1):7–63, 1989.

[7] A. Greenbaum. *Iterative Methods for Solving Linear Systems*. SIAM, Philadelphia, USA, 1997.

[8] A. Greenbaum and Z. Strakoš. Predicting the behaviour of finite precision Lanczos and conjugate gradient computations. *SIAM Journal on Matrix Analysis and Applications*, 13(1):121–137, 1992.

[9] A. Griewank and G. Corliss. *Automatic Differentiation of Algorithms: Theory, Implementation and Application*. SIAM, Philadelphia, USA, 1991.

[10] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of the National Bureau of Standards*, 49:409–436, 1952.

[11] Nicholas J. Higham. *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008.

[12] A. S. Householder. *The Theory of Matrices in Numerical Analysis*. Dover, New-York, 1964.

[13] S. V. Kuznetsov. Perturbation bounds of the Krylov bases and associated Hessenberg forms. *Linear Algebra and its Applications*, 265(1):1–28, 1997.

[14] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of research of the National Bureau of Standards B*, 45:225–280, 1950.

[15] J. Liesen and Z. Strakoš. *Krylov Subspace Methods: Principles and Analysis*. Oxford University Press, Oxford, UK, 2012.

[16] J.R. Magnus and H. Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. Wiley Series in Probability and Statistics. Wiley, 1999.

[17] G. Meurant. *Computer Solution of Large Linear Systems*. North Holland, Amsterdam, 1999.

[18] G. Meurant. *The Lanczos and Conjugate Gradient Algorithms: From Theory to Finite Precision Computations*. SIAM, Philadelphia, USA, 2006.

[19] G. Meurant and Z. Strakoš. The Lanczos and conjugate gradient algorithms in finite precision arithmetic. *Acta Numerica*, 15:471–542, 2006.

[20] A. M. Moore, H. G. Arango, and G. Broquet. Estimates of analysis and forecast errors derived from the adjoing of 4D-Var. *Submitted for publication*, 2012.

[21] J. Nocedal and S. J. Wright. *Numerical Optimization*. Series in Operations Research. Springer Verlag, Heidelberg, Berlin, New York, 1999.

[22] C. C. Paige. Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix. *Journal of the Institute of Mathematics and its Applications*, 18:341–349, 1976.

[23] C. C. Paige. A useful form of unitary matrix obtained from any sequence of unit 2-norm n-vectors. *SIAM Journal on Matrix Analysis and Applications*, 31(2):565–583, 2009.

[24] C. C. Paige. An augmented stability result for the lanczos hermitian matrix tridiagonalization process. *SIAM Journal on Matrix Analysis and Applications*, 31(5):2347–2359, 2010.

[25] C. C. Paige, B. Parlett, and H. van der Vorst. Approximate solutions and eigenvalue bounds from Krylov subspaces. *Numerical Linear Algebra with Applications*, 2(2):115–133, 1995.

[26] Z. Strakoš. On the real convergence rate of the conjugate gradient method. *Linear Algebra and its Applications*, 154-156:535–549, 1991.

[27] A. J. Wathen and T. Rees. Chebyshev semi-iteration in preconditioning for problems including the mass matrix. *Electronic Transactions on Numerical Analysis*, 34:125.

[28] Y. Zhu and R. Gelaro. Observation sensitivity calculations using the adjoint of the Gridpoint Statistical Interpolation (GSI) analysis system. *Monthly Weather Review*, (136):335–351, 2008.