

GKB-FP: an algorithm for large-scale discrete ill-posed problems

Fermín S. Viloche Bazán · Leonardo S. Borges

Received: 24 August 2009 / Accepted: 20 May 2010
© Springer Science + Business Media B.V. 2010

Abstract We describe an algorithm for large-scale discrete ill-posed problems, called GKB-FP, which combines the Golub-Kahan bidiagonalization algorithm with Tikhonov regularization in the generated Krylov subspace, with the regularization parameter for the projected problem being chosen by the fixed-point method by Bazán (Inverse Probl. 24(3), 2008). The fixed-point method selects as regularization parameter a fixed-point of the function $\|r_\lambda\|_2/\|f_\lambda\|_2$, where f_λ is the regularized solution and r_λ is the corresponding residual. GKB-FP determines the sought fixed-point by computing a finite sequence of fixed-points of functions $\|r_\lambda^{(k)}\|_2/\|f_\lambda^{(k)}\|_2$, where $f_\lambda^{(k)}$ approximates f_λ in a k -dimensional Krylov subspace and $r_\lambda^{(k)}$ is the corresponding residual. Based on this and provided the sought fixed-point is reached, we prove that the regularized solutions $f_\lambda^{(k)}$ remain *unchanged* and therefore completely insensitive to the number of iterations. This and the performance of the method when applied to well-known test problems are illustrated numerically.

Keywords Tikhonov regularization · Large-scale problems · Ill-posed problems

Mathematics Subject Classification (2000) 15A06 · 15A23 · 15A18

Communicated by Lars Eldén.

The work of the first author was supported by CNPq, Brazil, grant 308154/2008-8. The work of the second author was supported by FAPESP, Brazil, grant 2009/52193-1.

F.S.V. Bazán (✉)
Department of Mathematics, Federal University of Santa Catarina, CEP 88040-900,
Florianópolis SC, Brazil
e-mail: fermin@mtm.ufsc.br

L.S. Borges
Department of Applied Mathematics, IMECC-UNICAMP, University of Campinas,
CP 6065, 13081-970, Campinas SP, Brazil
e-mail: lsbpls@yahoo.com

1 Introduction

We are concerned with the solution of large-scale discrete ill-posed problems of the form

$$\min_{f \in \mathbb{R}^n} \|g - Af\|_2, \quad A \in \mathbb{R}^{m \times n} \ (m \geq n), \ g \in \mathbb{R}^m, \quad (1.1)$$

where A is ill-conditioned and has singular values decaying to zero without particular gap in the singular value spectrum. These problems arise, for instance, when discretizing Fredholm integral equations of the first kind with smooth kernel. In applications the right-hand side contains noise arising from measurements or approximation error, and we assume that $g = g^{\text{exact}} + e$, where e denotes random noise, $g^{\text{exact}} = Af^{\text{exact}}$ denotes the unperturbed data and f^{exact} the noise-free solution. A consequence of this is that the naive least squares (LS) solution $f_{\text{ls}} = A^\dagger g$ (where A^\dagger denotes the pseudoinverse of A) is dominated by inaccuracies, and some sort of regularization must be used in order to compute stable approximations to the noise-free solution. Perhaps the earliest and most well-known method to deal with this class of problems is due to Tikhonov [38]. In its simplest form, Tikhonov's method amounts to replacing the least squares problem (1.1) by

$$\min_{f \in \mathbb{R}^n} \{ \|g - Af\|_2^2 + \lambda^2 \|f\|_2^2 \} \quad (1.2)$$

where $\lambda > 0$ is the regularization parameter. Solving (1.2) is equivalent to solving the *regularized* normal equations

$$(A^T A + \lambda^2 I_n) f = A^T g, \quad (1.3)$$

whose solution is $f_\lambda = (A^T A + \lambda^2 I_n)^{-1} A^T g$, where I_n is the $n \times n$ identity matrix, and the problem is how to select the parameter λ such that f_λ becomes as close as possible to the noise-free solution.

Many parameter-choice rules for Tikhonov regularization have been proposed in the literature. These can roughly be organized into two classes: rules that exploit knowledge of the norm of the error e (or some estimate), with the discrepancy principle (DP) of Morozov [31] as its most known representant, and heuristic rules that do not exploit this information. Rules of the second class include the L-curve criterion of Hansen and O'Leary [20], generalized cross-validation (GCV) of Heath, Golub and Wabba [14], weighted-GCV (W-GCV) of Chung et al. [10], and a fixed-point method (FP-method) by Bazán [1]; for a survey on Tikhonov parameter-choice rules see [21] and references therein, and for more recent contributions, see [3, 17, 18, 25, 26, 28, 37, 39]. An interesting motivation for the use of heuristics rules can be found in [12].

If the singular value decomposition (SVD) of the coefficient matrix is available, parameter-choice rules are simple to implement and several routines that exploit the SVD are now available, see, e.g., the Regularization Tools of Hansen [22]. The major drawback of SVD-based approaches is that they are unpractical for large-scale problems. Another way to construct stable approximations to the noise-free solution of (1.1) is by projections methods based on Lanczos/Arnoldi procedures. These include

CGLS and its analytical equivalent LSQR [34], GMRES [9], and other Krylov subspace methods such as MINRES [29]; it is well-known that each of these methods provides an iterative regularization algorithm as long as the iterations are stopped after the optimal number of steps, where optimality is with respect to the relative errors. Among the most well-known stopping criteria for iterative regularization are the discrepancy principle [7] and the L-curve method [19, 22]. In practice the norm of the noise $\|e\|_2$ is rarely available and so the discrepancy principle is of little use. As for the L-curve method, it is not always reliable to follow since in iterative regularization the L-curve is piecewise linear, in which case the extraction of good regularization parameters is not as simple as one might wish; for illuminating examples about this, the reader is referred to Morigi et al. [30]. Related work is also done by Frommer and Maass [13] and Golub and von Matt [16].

The difficulty in determining reliable stopping rules for Krylov subspace projection methods can be partially alleviated by combining them with an inner regularization algorithm at each iteration. This gives rise to the so-called *hybrid* methods [4, 24, 27, 28, 33]. These are attractive because the iterates *tend to stabilize* when optimal regularization parameters are chosen at each iteration. To the best of our knowledge the stabilization phenomenon was first observed by Hanke and Hansen [19, p. 302]; for an example in connection with a LSQR-Tikhonov hybrid method with the regularization parameter for the projected problem chosen by GCV, see [10, 28]. We also note that a great deal of work has been done on the determination of Tikhonov regularization parameters for large-scale problems through the so-called ribbon methodologies. These include the L-curve ribbon and corresponding curvature-ribbon, and the discrepancy-ribbon [6–8]. In the former case, the regularization parameter is determined by using approximations of the L-curve and its curvature, computed for several values of λ by partial Lanczos bidiagonalization and by the application of Gauss-type quadrature rules; a similar principle applies to the discrepancy method. However, our computational experience with the L-curve method using the *exact* L-curve and its curvature has been unsatisfactory, specially when the L-curve displays several convex L-corners, see the numerical examples in [1, 2]. For this reason our interest in the L-curve method is from the theoretical point of view only. In what follows the L-curve method will be denoted by LC and the fixed-point method by FP.

In this paper we are interested in problems for which the norm of the noise $\|e\|_2$ is not available, and concentrate on an algorithm, called GKB-FP, which can be regarded as an implementation of FP for large-scale problems. Our interest for developing GKB-FP was motivated by the excellent performance of an SVD-based implementation of FP on small problems reported in [1, 2], where FP is compared with LC, GCV and DP. The conclusion drawn in these references is that in a noisy environment FP is more robust than LC and GCV in the sense that the former determines regularization parameters more consistently than both LC and GCV. More specifically, while FP is shown to construct accurate regularized solutions, LC and GCV fail many times, this occurring mostly when the L-curve displays several convex L-corners [2].

GKB-FP combines a partial Golub-Kahan bidiagonalization (GKB) algorithm, see, e.g., [4, 15, 23] with Tikhonov regularization in the generated Krylov subspace, with the regularization parameter for the projected problem being chosen by FP. We recall that FP selects as regularization parameter a fixed-point of the

function $\|r_\lambda\|_2/\|f_\lambda\|_2$, where f_λ is the regularized solution and $r_\lambda = g - Af_\lambda$ is the corresponding residual. Practically, GKB-FP determines an approximation to the sought fixed-point by computing a finite sequence of fixed-points of functions $\|r_\lambda^{(k)}\|_2/\|f_\lambda^{(k)}\|_2$, where $f_\lambda^{(k)}$ approaches f_λ in a k -dimensional Krylov subspace and $r_\lambda^{(k)}$ is the corresponding residual.

The rest of the paper is organized as follows. Since GKB-FP relies on a combination of a partial Lanczos bidiagonalization algorithm with Tikhonov regularization in the generated Krylov subspace and properties of LSQR, these ideas are described in Sect. 2. After a brief review of FP, the main steps of our algorithm are described in Sect. 3. In Sect. 4 a convergence analysis that supports GKB-FP is presented and illustrated numerically. The paper ends with numerical results that illustrate the efficiency of GKB-FP on a variety of well-known test problems and some concluding remarks.

2 Golub–Kahan bidiagonalization and LSQR

The basis for our algorithm is the GKB algorithm. Application of $k < n$ GKB steps to A with initial vector $g/\|g\|_2$ yields three matrices: a lower bidiagonal matrix $B_k \in \mathbb{R}^{(k+1) \times k}$ and two matrices $U_{k+1} \in \mathbb{R}^{m \times (k+1)}$ and $V_k \in \mathbb{R}^{n \times k}$ with orthonormal columns, such that

$$\beta_1 U_{k+1} e_1 = g = \beta_1 u_1, \tag{2.1}$$

$$A V_k = U_{k+1} B_k, \tag{2.2}$$

$$A^T U_{k+1} = V_k B_k^T + \alpha_{k+1} v_{k+1} e_{k+1}^T, \tag{2.3}$$

where e_k denotes the k -th unit vector of appropriate dimension. LSQR produces a (finite) sequence of approximate solutions to (1.1) of the form $f^{(k)} = V_k y^{(k)}$, where $y^{(k)}$ solves the *projected* least squares problem

$$\min_{y \in \mathbb{R}^k} \|B_k y - \beta_1 e_1\|_2, \tag{2.4}$$

with

$$B_k = \begin{pmatrix} \alpha_1 & & & & & \\ \beta_2 & \alpha_2 & & & & \\ & \beta_3 & \ddots & & & \\ & & \ddots & \alpha_k & & \\ & & & \beta_{k+1} & & \end{pmatrix}. \tag{2.5}$$

In practice, $f^{(k)}$ is computed via a QR factorization of B_k which allows for an efficient updating of the LSQR iterates; the reader is referred to [34] for details. Theoretically, $f^{(k)}$ solves the problem

$$\min_{f \in \mathcal{K}_k} \|g - Af\|_2 \tag{2.6}$$

where $\mathcal{K}_k = \text{span}\{A^T g, (A^T A)A^T g, \dots, (A^T A)^{k-1}A^T g\}$, and mathematically LSQR produces the same sequence as the well-known CGLS method. A by-product of this is that while $\|f^{(k)}\|_2$ increases monotonically with k , the norms of the residual vectors $r^{(k)} = g - Af^{(k)}$ decreases monotonically with k .

LSQR is also well suited for solving the ‘‘damped least squares problem’’ [34]

$$f_\lambda = \arg \min_{f \in \mathbb{R}^n} \left\| \begin{pmatrix} g \\ 0 \end{pmatrix} - \begin{pmatrix} A \\ \lambda I_n \end{pmatrix} f \right\|_2, \tag{2.7}$$

which is equivalent to (1.3), where λ is a fixed regularization parameter. In this case, the k th approximate solution is taken to be

$$f_\lambda^{(k)} = V_k y_\lambda^{(k)}, \tag{2.8}$$

where $y_\lambda^{(k)}$ solves the regularized projected problem

$$y_\lambda^{(k)} = \arg \min_{y \in \mathbb{R}^k} \left\| \begin{pmatrix} \beta_1 e_1 \\ 0 \end{pmatrix} - \begin{pmatrix} B_k \\ \lambda I_k \end{pmatrix} y \right\|_2, \tag{2.9}$$

which is solved efficiently using the QR factorization of $\begin{pmatrix} B_k \\ \lambda I_k \end{pmatrix}$,

$$Q_k \begin{pmatrix} B_k \\ \lambda I_k \end{pmatrix} = \begin{pmatrix} R_k \\ 0 \\ 0 \end{pmatrix}, \quad Q_k(\beta_1 e_1) = \begin{pmatrix} b_k \\ \bar{\varphi}_{k+1} \\ c_k \end{pmatrix}, \tag{2.10}$$

where R_k is upper bidiagonal,

$$b_k = \begin{pmatrix} \varphi_1 \\ \vdots \\ \varphi_k \end{pmatrix}, \quad c_k = \begin{pmatrix} \psi_1 \\ \vdots \\ \psi_k \end{pmatrix}, \tag{2.11}$$

and Q_k (of order $2k + 1$) is a product of two sequences of plane rotations, $Q_k = J_k G_k \cdots J_2 G_2 J_1 G_1$, where G_j and J_j are chosen to eliminate elements in positions $(k + j, j)$ and $(j + 1, j)$, $j = 1, \dots, k$, respectively. The solution $y_\lambda^{(k)}$ can then be obtained from the triangular system $R_k y = b_k$. As for $f_\lambda^{(k)}$, it can be computed using (2.8). Alternatively, as shown by Paige and Saunders [34], for fixed λ the regularized solution $f_\lambda^{(k)}$ can be computed through the updating formula

$$f_\lambda^{(k)} = f_\lambda^{(k-1)} + \varphi_k d_k, \tag{2.12}$$

where d_k , which is the last column of $D_k = V_k R_k^{-1}$, can be computed without requiring storage of V_k . Note that due to (2.1) and (2.2), the residual vector $r_\lambda^{(k)} = g - Af_\lambda^{(k)}$ and $f_\lambda^{(k)}$ satisfy

$$\|r_\lambda^{(k)}\| = \|g - Af_\lambda^{(k)}\| = \|U_{k+1}^T(\beta_1 e_1 - B_k y_\lambda^{(k)})\| = \|\beta_1 e_1 - B_k y_\lambda^{(k)}\|, \tag{2.13}$$

$$\|f_\lambda^{(k)}\| = \|y_\lambda^{(k)}\|. \tag{2.14}$$

We end the section with a technical result.

Theorem 2.1 *For fixed $\lambda > 0$ the norm of the solution $f_\lambda^{(k)}$ and corresponding norm of the residual vector $r_\lambda^{(k)}$ satisfy*

$$\|f_\lambda^{(k+1)}\| \geq \|f_\lambda^{(k)}\|, \quad \|r_\lambda^{(k+1)}\| \leq \|r_\lambda^{(k)}\|, \quad k = 0, \dots, n - 1. \quad (2.15)$$

Proof The first inequality follows from the monotonic properties of the LSQR iterates. To prove the second inequality note that like the LSQR iterate $f^{(k)}$, the regularized solution $f_\lambda^{(k)}$ satisfies

$$f_\lambda^{(k)} = \arg \min_{f \in \mathcal{K}_k} \{ \|g - Af\|^2 + \lambda^2 \|f\|^2 \}. \quad (2.16)$$

Now since $f_\lambda^{(k)} \in \mathcal{K}_{k+1}$ it follows that

$$\lambda^2 \|f_\lambda^{(k+1)}\|_2^2 + \|r_\lambda^{(k+1)}\|_2^2 \leq \lambda^2 \|f_\lambda^{(k)}\|_2^2 + \|r_\lambda^{(k)}\|_2^2,$$

which implies the second inequality in (2.15). □

3 Our algorithm

In order to describe the main ideas underlying GKB-FP, we shall start with a brief review of the fixed-point method.

3.1 Brief review of FP-method

FP can be regarded as a realization of a parameter choice rule devised by Regińska [36], who proposed as regularization parameter a local minimum of the function

$$\Psi_\mu(\lambda) = x(\lambda)y^\mu(\lambda), \quad (3.1)$$

for proper $\mu > 0$, where $y(\lambda) = \|f_\lambda\|_2^2$ and $x(\lambda) = \|g - Af_\lambda\|_2^2$. The motivation for using this rule can be supported heuristically noting that when λ is small, the squared solution norm $y(\lambda)$ gets large while $x(\lambda)$ gets small, and $\Psi_1(\lambda)$ is not minimized. Conversely, when λ is large, $y(\lambda)$ gets small while $x(\lambda)$ gets large, and once again $\Psi_1(\lambda)$ is not minimized. This suggests that the minimizer of $\Psi_1(\lambda)$ corresponds to a good balance between the size of the solution norm and the size of the residual norm. A similar result is expected when $\mu \neq 1$. Regińska proved that if the curvature of the L-curve is maximized at $\lambda = \lambda^*$, and if the tangent to the L-curve at $(\log x(\lambda^*), \log y(\lambda^*))$ has slope $-1/\mu$, then $\Psi_\mu(\lambda)$ is minimized at $\lambda = \lambda^*$. Bazán [1] investigated the properties of $\Psi_\mu(\lambda)$ and concluded that its minimizers are converging values of a sequence defined by

$$\lambda_{j+1} = \phi_\mu(\lambda_j), \quad j \geq 0, \quad \phi_\mu(\lambda) = \sqrt{\mu} \frac{\|r_\lambda\|_2}{\|f_\lambda\|_2}, \quad \lambda > 0, \quad (3.2)$$

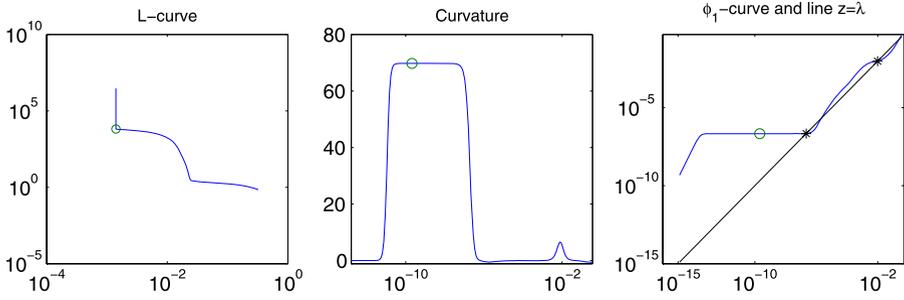


Fig. 1 Curves for test problem heat with $n = 128$. The right hand side g satisfies $\|e\|_2/\|g^{\text{exact}}\|_2 = 0.05$. Noise is generated by the *randn* Matlab function with the seed value set to zero

Table 1 Numerical results for test problem heat

	L-curve	FP
λ	0.7607e-2	0.9783e-2
RE	0.2609e-0	0.2434e-0

which gave rise to the FP-method. A little difficulty with FP is that it strongly depends on the initial guess: a wrong choice of the initial guess may lead either to a very small fixed-point of ϕ_μ or to a maximizer of Ψ_μ . Although this poses no problem when the L-curve is well-behaved (i.e., when it displays a unique and sharp convex L-corner), in which case there is a unique convex fixed-point that is easy to reach, this may not be the case when the L-curve displays several convex corners; we recall from [2] that a fixed-point of ϕ_μ is said to be *convex* when the L-curve is locally convex at such point. A multiple convex corner case is illustrated in Fig. 1 where the curvature of the L-curve and the curve of function ϕ_1 are also displayed. In this figure, the symbol “o” corresponds to the regularization parameter determined by L-curve, as done by function `l_curve` from [22], and the symbol * corresponds to a convex-fixed point of ϕ_1 . Note that in this case the parameters determined by L-curve and FP fall far away. To make matters worse, one can verify that the sharper L-corner and the smallest convex fixed-point, respectively, lead to severely undersmoothed solutions.

To circumvent the difficulties mentioned above, Bazán and Francisco [2] suggested that instead of looking for a specific L-corner such as the sharper one, for instance, one should look for the largest convex fixed-point of ϕ_μ . Regularization parameters determined by the rightmost L-corner and the largest convex-fixed point, as well as the relative errors (denoted by RE) of the corresponding regularized solutions for the test problem heat with data mentioned above are displayed in Table 1.

Further numerical results that illustrate the effectiveness of this rule can be found in [2].

We close this section by pointing out two of the main steps of FP.

- Given a proper initial guess, FP starts with $\mu = 1$ as a default value, and then proceeds with the iterates (3.2) until the largest-fixed point of ϕ_1 is reached. The choice $\mu = 1$ is because in most problems this value yields a regularization parameter that leads to a point on the L-curve near the corner of maximum curvature

(geometrically, $\mu = 1$ means that the tangent line to the L-curve at the L-corner forms an angle of 135 degrees with the horizontal axis).

- If $\mu = 1$ does not work, μ is adjusted and the iterations restart; see [1, 2] for details.

3.2 Proposed method

We start by introducing the finite sequence of functions $\phi_\mu^{(k)} : \mathbb{R}^+ \rightarrow \mathbb{R}, k = 2, \dots, n$, defined by

$$\phi_\mu^{(k)}(\lambda) = \sqrt{\mu} \frac{\|\beta_1 e_1 - B_k y_\lambda^{(k)}\|_2}{\|y_\lambda^{(k)}\|_2}, \quad \mu > 0.$$

Let the singular value decomposition of B_k be

$$B_k = P_k \begin{pmatrix} \Sigma_k \\ 0 \end{pmatrix} Q_k^T = \sum_{i=1}^k \sigma_i^{(k)} p_i q_i^T, \tag{3.3}$$

where P_k and Q_k are square orthogonal matrices, and $\Sigma_k = \text{diag}(\sigma_1^{(k)}, \dots, \sigma_k^{(k)})$ with $\sigma_1^{(k)} \geq \sigma_2^{(k)} \geq \dots \geq \sigma_k^{(k)} > 0$. Then it is immediate to verify that

$$\begin{aligned} \|y_\lambda^{(k)}\|_2^2 &= \beta_1^2 \sum_{i=1}^k \frac{[\sigma_i^{(k)}]^2 \xi_{1i}^2}{([\sigma_i^{(k)}]^2 + \lambda^2)^2}, \\ \|r_\lambda^{(k)}\|_2^2 &= \beta_1^2 \left(\sum_{i=1}^k \frac{\lambda^4 \xi_{1i}^2}{([\sigma_i^{(k)}]^2 + \lambda^2)^2} + \delta_0^{(k)2} \right), \end{aligned} \tag{3.4}$$

where we have put $\xi_{1i} = P_k(1, i)$, and where $\delta_0^{(k)}$ is the 2-norm of the incompatible part of $\beta_1 e_1$ that lies outside $\mathcal{R}(B_k)$, the column space of B_k . Straightforward computations show that

$$\phi_\mu^{(k)'}(\lambda) > 0, \quad \text{for } \lambda > 0, \tag{3.5}$$

where $'$ denotes differentiation with respect to λ , and so $\phi_\mu^{(k)}$ strictly increases with λ .

Now for given $\lambda_0^{(k)}$ consider the sequence

$$\lambda_{j+1}^{(k)} = \phi_1^{(k)}(\lambda_j^{(k)}), \quad j \geq 0, \tag{3.6}$$

and assume it converges to a fixed point $\lambda^{(k)*}$ of $\phi_1^{(k)}$; when this is true and the incompatible part of g that lies outside $\mathcal{R}(A)$ is nonzero, i.e., $\delta_0^{(n)} \neq 0$, we shall prove that the function ϕ_1 of the original problem always has a fixed-point λ^* that minimizes Ψ_1 , that the sequence $\lambda^{(k)*}$ converges to λ^* in at most n steps, and that λ^* is the largest convex fixed-point of ϕ_1 .

GKB-FP constructs an approximation to λ^* using a nonincreasing finite sequence of fixed-points $\lambda^{(k)*}$. This requires solving the projected problem (2.9) for several

Table 2 GKB-FP

Input: $A, g, p > 1, k_{\max}, \epsilon.$

Output: Regularized solution $f_{\lambda^*}^{(k)}$

1. Apply p GKB steps to A with starting vector b and form the matrix $B_p.$
2. Set $k = p.$ Compute the fixed point $\lambda^{(k)*}$ of $\phi_1^{(k)}$ and set $\lambda_0 = \lambda^{(k)*}, \lambda_{\text{old}} = \lambda_0, k \leftarrow k + 1.$
3. Perform one more GKB step and compute the fixed point $\lambda^{(k)*}$ of $\phi_1^{(k)}$ taking λ_0 as starting value. Set $\lambda_{\text{old}} = \lambda_0, \lambda_0 = \lambda^{(k)*}.$
4. **If** stopping criterion is satisfied **do**
 $\lambda^* = \lambda_{\text{old}}$
else do
 $k \leftarrow k + 1$
 Go to step 3.
end if
5. Compute the regularized solution $f_{\lambda^*}^{(k)}$

values of λ for fixed k (but increasing), and our proposal is to do this following the ideas of the LSQR algorithm as described in the previous section. The main steps of GKB-FP are summarized in Table 2.

We refer to each GKB iteration as an outer iteration and to iterations during fixed-point computations for each k as inner iterations. Computation of fixed-points requires for each k , the evaluation of $\phi_1^{(k)}$ for several values of λ and for increasing values of k . The number of evaluations depends on the desired precision and our experience is that the largest number of evaluations corresponds to step 2. Obviously, such an evaluation requires computing both $\|r_\lambda^{(k)}\|_2$ and $\|y_\lambda^{(k)}\|_2$. This can be made by using the QR factorization of the augmented matrix in (2.10) and the residual norm of the regularized projected problem (2.9) which we denote here by $\|\bar{r}_\lambda^{(k)}\|_2$. In fact, since by (2.9) and (2.10) we have

$$\|r_\lambda^{(k)}\|_2^2 + \lambda^2 \|y_\lambda^{(k)}\|_2^2 = \|\bar{r}_\lambda^{(k)}\|_2^2 = |\bar{\varphi}_{k+1}|^2 + \|c_k\|_2^2,$$

then it follows that

$$\|r_\lambda^{(k)}\|_2^2 = |\bar{\varphi}_{k+1}|^2 + \|c_k\|_2^2 - \lambda^2 \|y_\lambda^{(k)}\|_2^2. \tag{3.7}$$

This shows that the evaluation of $\phi_1^{(k)}(\lambda)$ for each λ can be made approximately in $\mathcal{O}(k)$ arithmetic operations, which is the cost of solving the regularized projected problem (2.9), see Eldén [11] and Paige and Saunders [35].

There are probably many ways of introducing savings for GKB-FP to be efficient and competitive. We mention two ways. The first one, already incorporated at step 3, is by taking $\lambda^{(k)*}$ as initial guess for computing the fixed-point of $\phi_1^{(k+1)}$. This is not only computationally practical, but also theoretically correct since, as commented above, the sequence of fixed points $\lambda^{(k)*}$ is a nonincreasing one; we shall return to this point later. The second one is by choosing a tolerance parameter for stopping

computation of fixed-points not too stringent in order to keep the number of evaluations of $\phi_1^{(k)}$ small. Our computational experience is that from $k = p + 1$ on, the number of evaluations of $\phi_1^{(k)}$ remains within moderate numbers such as 2 or 3, as we shall illustrate in the numerical results section.

Thus, if we assume that a good approximation to the largest fixed-point of ϕ_1 is reached at step k^* , the computation of f_λ via (2.8) roughly requires the evaluation of k^* matrix-vector products with each one of the matrices A and A^T , the computation of $(k^* - p + 1)$ fixed-points, and the evaluation one matrix-vector product with the matrix V_{k^*} . Since the computation of each fixed-point, from step $k = p + 1$ on, requires the evaluation of $\phi_1^{(k)}$ very few times, each being at the cost of $\mathcal{O}(k)$ arithmetic operations, the overall cost of the computation of f_λ should not exceed significantly the cost of the k^* Lanczos iterations, as long as k^* remains within moderate bounds. Taking as basis this discussion, the analysis of the cost of f_λ via the updating formula (2.12) is straightforward.

3.2.1 Stopping criteria

We first note that because the Lanczos process captures the dominant part of the singular spectrum of the coefficient matrix in a relatively small number of steps, $\phi_1^{(k)}(\lambda)$ should approximate $\phi_1(\lambda)$ relatively well, at least for λ in some interval $[a_k, \sigma_1^{(k)}]$ with a_k larger than $\sigma_k^{(k)}$ and close to some singular value dominant of A already captured at step k . Let λ^* denote the sought fixed-point associated with the large-scale problem. Then, if for certain k , one has λ^* close to but smaller than a_k , it is reasonable that the sequence of fixed-points $\lambda^{(k)*}$ stabilize near λ^* in a few additional GKB steps. This heuristic is supported by the fact that $\phi_1^{(k)}(\lambda)$ approximates $\phi_1(\lambda)$ from above and that the sequence of fixed-points $\lambda^{(k)*}$ is nonincreasing. These properties shall be proved and numerically illustrated in the next section. Based on this we choose to stop the iterations when the relative change of consecutive fixed points is small,

$$|\lambda^{(k+1)*} - \lambda^{(k)*}| < \epsilon_1 |\lambda^{(k)*}|, \quad (3.8)$$

where ϵ_1 is a small tolerance parameter. A disadvantage of the stopping criterion (3.8) is that it can delay convergence when the sequence $\lambda^{(k)*}$ decreases very slowly. To circumvent this difficulty we introduce another stopping criterion defined by

$$|\lambda^{(k+1)*} - \lambda^{(k)*}| < \epsilon_2 |\lambda^{(0)*}|, \quad (3.9)$$

where $\lambda^{(0)*}$ is the fixed-point computed at step 2 and ϵ_2 is another tolerance parameter. The stopping criterion used by GKB-FP is to accept as regularization parameter of the large scale-problem, the first fixed-point $\lambda^{(k)*}$ satisfying either (3.8) or (3.9).

4 Convergence analysis

It is important to stress that because we shall deal with mathematical properties of our algorithm, in this section we shall assume exact arithmetic. We start by discussing the

question about existence of fixed-points of the function $\phi_1^{(k)}$. The key idea is to exploit an existing close relationship between the L-curve of the projected problem and the function $\phi_1^{(k)}$. Let the L-curve in log-log scale of the projected problem be denoted by $\mathcal{L}^{(k)}$, i.e.,

$$\mathcal{L}^{(k)}(\lambda) = \{(a, b) \mid a = \log \|r_\lambda^{(k)}\|^2, b = \log \|y_\lambda^{(k)}\|^2, \lambda > 0\},$$

and let the slope of $\mathcal{L}^{(k)}$ be denoted by $m_L^{(k)}$. It is straightforward to verify that (see [1])

$$m_L^{(k)}(\lambda) = -\frac{[\phi_1^{(k)}(\lambda)]^2}{\lambda^2}.$$

Usually, $m_L^{(k)}(\lambda) \approx -1$ for λ near the L-corner and $|m_L^{(k)}(\lambda)|$ is small for λ associated with the flat part of the L-curve. We are going to show that for the function $\phi_1^{(k)}$ to have convex fixed-points it is required that the associated L-curve $\mathcal{L}^{(k)}(\lambda)$ have a flat part with $|m_L^{(k)}(\lambda)|$ sufficiently small. Before proceeding we recall from [2] that a fixed-point $\lambda^{(k)*}$ of $\phi_1^{(k)}$ is said to be a *convex fixed-point* (resp. *concave*) if $\mathcal{L}^{(k)}$ is locally convex (resp. concave) in a vicinity of $\lambda^{(k)*}$.

The theorem below provides conditions in terms of $m_L^{(k)}(\lambda)$ that guarantee existence of convex fixed-points of $\phi_1^{(k)}$.

Theorem 4.1 *Assume $\beta_{k+1} \neq 0, k \geq 2$. A sufficient condition for $\phi_1^{(k)}$ to have a convex fixed-point, and hence, a concave fixed-point, in the interval $(0, \sigma_1^{(k)})$, is that $|m_L^{(k)}(\lambda)| < 1$ in some interval $I \subset (0, \sigma_1^{(k)})$.*

Proof Introduce the function $h : \mathbb{R}^+ \rightarrow \mathbb{R}$ defined by $h(\lambda) = \|r_\lambda^{(k)}\|^2 - \lambda^2 \|y_\lambda^{(k)}\|^2$. It is clear that h is continuous on \mathbb{R}^+ and that $\bar{\lambda}$ is a fixed-point of $\phi_1^{(k)}$ if and only if $\bar{\lambda}$ is a zero of h . Note that the assumption $\beta_{k+1} \neq 0$ means that $\beta_1 e_1$ does not belong to $\mathcal{R}(B_k)$ (i.e., the system $B_k y = \beta_1 e_1$ in (2.4) is inconsistent) and so $\delta_0^{(k)} > 0$. This implies that $\|r_\lambda^{(k)}\|^2 \rightarrow \delta_0^{(k)} > 0$ as $\lambda \rightarrow 0^+$, and therefore $h(\lambda) \rightarrow \delta_0^{(k)} > 0$ as $\lambda \rightarrow 0^+$. Invoking Lemma 1 in [1] it follows that

$$[\phi_1^{(k)}(\lambda)]^2 > \lambda^2 \quad \text{for } \lambda > \sigma_1^{(k)}$$

and hence $h(\lambda) > 0$ for $\lambda > \sigma_1^{(k)}$. These inequalities together with assumption $|m_L^{(k)}(\lambda)| < 1$ imply that $h(\lambda)$ changes sign at least twice. This guarantees that h has at least two zeros which are, respectively, a convex and a concave fixed-point of $\phi_1^{(k)}$. □

The following theorem shows how the function ϕ_1 of the original problem relates to the sequence of functions $\phi_1^{(k)}$ of the projected problems.

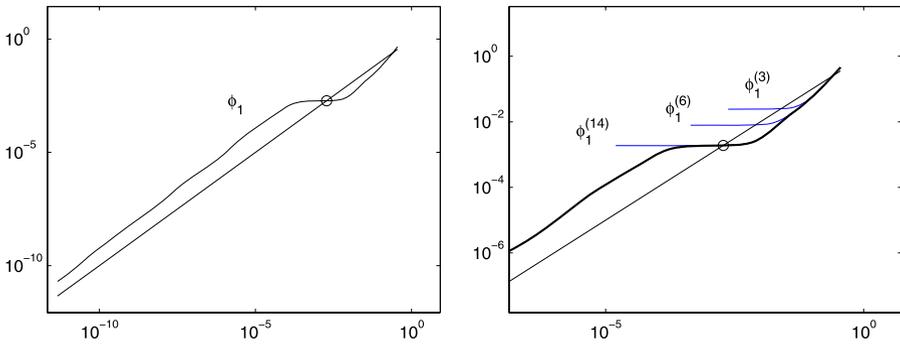


Fig. 2 Function $\phi_1(\lambda)$ corresponding to the “large” problem (1.2) (left). Some functions $\phi_1^{(k)}(\lambda)$ corresponding to the projected problem (2.4) (right). In this case we consider the test problem heat of size $n = 800$ and data satisfying $\|\epsilon\|_2/\|s^{\text{exact}}\|_2 = 0.1$. The *small circle* denotes the largest convex fixed-point of ϕ_1

Theorem 4.2 For all $\lambda > 0$ there holds

$$\phi_1^{(k+1)}(\lambda) \leq \phi_1^{(k)}(\lambda), \quad k = 2, \dots, n - 1, \tag{4.1}$$

and therefore

$$\phi_1(\lambda) \leq \phi_1^{(k)}(\lambda), \quad k = 2, \dots, n - 1. \tag{4.2}$$

Proof The first inequality is straightforward from Theorem 2.1. The second inequality follows upon taking $k = n - 1$ in (4.1). □

Perhaps the most important consequence of Theorem 4.2 is that due to the approximation properties of the Lanczos process, the sequence $\phi_1^{(k)}(\lambda)$ quickly converges to $\phi_1(\lambda)$ at least for the values of λ near the part of the singular spectrum of A that contains the largest convex fixed-point of $\phi_1(\lambda)$. This is illustrated in Fig. 2 where are depicted some functions $\phi_1^{(k)}$ as well as its limiting value $\phi_1(\lambda)$ for the heat test problem. Notice that excellent approximations to ϕ_1 are obtained in a few GKB steps.

Another consequence of Theorem 4.2 is that if the assumptions of Theorem 4.1 hold, then the sequence $\{\lambda^{(k)*}\}$ converges to a convex fixed-point of ϕ in a finite number of GKB steps. This is the subject of the following theorem.

Theorem 4.3 Provided that the assumptions of Theorem 4.1 hold, the sequence of convex fixed-points $\lambda^{(k)*}$ is nonincreasing, and if $\delta_0^{(n)} \neq 0$, then $\lambda^{(k)*}$ converges to a convex fixed-point of ϕ_1 that is closest to $\lambda^{(p)*}$ in at most n GKB steps.

Proof Let $\lambda^{(k)*}$, $k \geq p$, be a convex fixed-point of $\phi_1^{(k)}(\lambda)$. Due to Theorem 4.2 we have

$$\phi_1^{(k+1)}(\lambda^{(k)*}) \leq \phi_1^{(k)}(\lambda^{(k)*}) = \lambda^{(k)*}. \tag{4.3}$$

If $\lambda^{(k)*}$ is a fixed-point of $\phi_1^{(k+1)}$ there is nothing to prove. Assume then that $\phi_1^{(k+1)}(\lambda^{(k)*}) < \lambda^{(k)*}$, and consider the sequence $\lambda_{j+1} = \phi_1^{(k+1)}(\lambda_j)$, $j \geq 0$, with starting value $\lambda_0 = \lambda^{(k)*}$. Based on the fact that $\phi_1^{(k+1)}$ increases with λ , see (3.5), it follows that λ_j forms a non increasing sequence, and since $\delta_0^{k+1} > 0$ by assumption, it turns out that $\{\lambda_j\}$ converges to a fixed point of $\phi_1^{(k+1)}$, i.e.,

$$\lim_{j \rightarrow \infty} \lambda_j = \lambda^{(k+1)*} = \phi_1(\lambda^{(k+1)*}),$$

with $\lambda^{(k+1)*} \leq \lambda^{(k)*}$. Now notice that according to (4.3) and the continuity of $m_L^{(k+1)}(\lambda)$, it follows that $|m_L^{(k+1)}(\lambda)| < 1$ in some vicinity of $\lambda^{(k)*}$, and convexity of $\lambda^{(k+1)*}$ is a consequence of Theorem 4.1.

Finally, note that combining the assumption $\delta_0^{(n)} \neq 0$ with the property (4.2), because after n GKB steps the Krylov space equals \mathbb{R}^n , it follows that $\phi_1^{(n)}$ equals ϕ_1 and so $\{\lambda^{(k)*}\}$ must converge to the largest convex fixed-point of ϕ_1 in at most n GKB steps, and the proof concludes. \square

A remarkable property of GKB-FP is that from a certain k on, the error in the regularized solution $f_{\lambda^{(k)*}}^{(k)}$ with respect to f^{exact} remains essentially constant provided that $\lambda^{(k)*}$ is sufficiently close to λ^* . More precisely, we have the following result.

Theorem 4.4 *Assume that the largest convex fixed-point of ϕ_1 is reached in k GKB steps, i.e., $\lambda^* = \phi_1^{(k)}(\lambda^*)$. Then $f_{\lambda^*}^{(k)} = f_{\lambda^*}^{(k+1)} = \dots = f_{\lambda^*}^{(n)} = f_{\lambda^*}$ and as a consequence, the error norm $\|f^{\text{exact}} - f_{\lambda^*}^{(j)}\|_2$ remains constant for $k \leq j \leq n$.*

Proof We first note that according to the LSQR algorithm, at step k we have $f_{\lambda^*}^{(k)} = V_k y_{\lambda^*}^{(k)}$, and that $y_{\lambda^*}^{(k)}$ solves the system $R_k y = b_k$, with R_k and b_k obtained from the transformation

$$Q_k \begin{bmatrix} B_k & \beta_1 e_1 \\ \lambda^* I_k & 0 \end{bmatrix} = \begin{bmatrix} R_k & b_k \\ 0 & \bar{\varphi}_{k+1} \\ 0 & c_k \end{bmatrix}, \quad b_k = \begin{bmatrix} \varphi_1 \\ \vdots \\ \varphi_k \end{bmatrix}, \quad c_k = \begin{bmatrix} \psi_1 \\ \vdots \\ \psi_k \end{bmatrix} \tag{4.4}$$

where $Q_k \in \mathbb{R}^{(2k+1) \times (2k+1)}$ is a product of Givens rotations. It is worthwhile to mention that only three new quantities are to be computed at step $k + 1$: φ_{k+1} (replacing $\bar{\varphi}_{k+1}$), $\bar{\varphi}_{k+2}$, and ψ_{k+1} ; the quantities in b_k and c_k remain unchanged and form part of b_{k+1} and c_{k+1} , respectively. We also mention that $f_{\lambda^*}^{(k+1)}$ can be obtained through the updating formula

$$f_{\lambda^*}^{(k+1)} = f_{\lambda^*}^{(k)} + \varphi_{k+1} d_{k+1},$$

see (2.12), where d_{k+1} is the $(k + 1)$ th column of $D_{k+1} = V_{k+1} R_{k+1}^{-1}$. We shall prove that $\varphi_{k+1} = 0$. In fact, since λ^* is a fixed-point of $\phi_1^{(j)}$ for $j = k, \dots, n$ (which holds

because of Theorem 4.2), we have

$$\|\beta_1 e_1 - B_k y_{\lambda^*}^{(k)}\|_2 = \lambda^* \|f_{\lambda^*}^{(k)}\|_2, \quad \text{and} \quad \|\beta_1 e_1 - B_{k+1} y_{\lambda^*}^{(k+1)}\|_2 = \lambda^* \|f_{\lambda^*}^{(k+1)}\|_2, \tag{4.5}$$

and hence

$$\|f_{\lambda^*}^{(k+1)}\|_2 \leq \|f_{\lambda^*}^{(k)}\|_2,$$

where we have used the fact that $\|\beta_1 e_1 - B_{k+1} y_{\lambda^*}^{(k+1)}\| \leq \|\beta_1 e_1 - B_k y_{\lambda^*}^{(k)}\|$ (see (2.15)). But since the solution norms form a nondecreasing sequence (see (2.15) again), it follows that $\|f_{\lambda^*}^{(k+1)}\|_2 = \|f_{\lambda^*}^{(k)}\|_2$, and substitution of this result into (4.5) yields

$$\|\beta_1 e_1 - B_k y_{\lambda^*}^{(k)}\|_2 = \|\beta_1 e_1 - B_{k+1} y_{\lambda^*}^{(k+1)}\|_2. \tag{4.6}$$

On the other hand, using (4.4) it follows that

$$\|\beta_1 e_1 - B_k y_{\lambda^*}^{(k)}\|_2^2 = |\bar{\varphi}_{k+1}|^2 + \|c_k\|_2^2 - \lambda^* \|f_{\lambda^*}^{(k)}\|_2^2,$$

and

$$\|\beta_1 e_1 - B_{k+1} y_{\lambda^*}^{(k+1)}\|_2^2 = |\bar{\varphi}_{k+2}|^2 + \|c_{k+1}\|_2^2 - \lambda^* \|f_{\lambda^*}^{(k+1)}\|_2^2.$$

Substituting these results into (4.6) yields

$$|\bar{\varphi}_{k+1}|^2 + \|c_k\|_2^2 = |\bar{\varphi}_{k+2}|^2 + \|c_{k+1}\|_2^2. \tag{4.7}$$

But since the 2-norm of

$$Q_k \begin{bmatrix} \beta_1 e_1 \\ 0 \end{bmatrix} = \begin{bmatrix} b_k \\ \bar{\varphi}_{k+1} \\ c_k \end{bmatrix},$$

in (4.4) equals β_1 and does not depend on k , it follows that

$$\begin{aligned} &|\varphi_1|^2 + \dots + |\varphi_k|^2 + |\bar{\varphi}_{k+1}|^2 + \|c_k\|_2^2 \\ &= (|\varphi_1|^2 + \dots + |\varphi_k|^2) + |\varphi_{k+1}|^2 + |\bar{\varphi}_{k+2}|^2 + \|c_{k+1}\|_2^2. \end{aligned}$$

Using (4.7) into this last equation we conclude that $\varphi_{k+1} = 0$, as claimed. □

In practice the condition $\lambda^* = \phi_1^{(k)}(\lambda^*)$ in Theorem 4.4 may not be satisfied after some number of GKB steps. For this, it seems useful to bound the error $\|f_{\lambda^{(k)*}}^{(k)} - f^{\text{exact}}\|_2$ and to assess the behavior of the bound as the iterations proceed. This can be addressed as follows. For each GKB step consider the (finite) sequence of positive numbers ϵ_k defined by

$$(\lambda^{(k)*2} - \lambda^{*2}) = \epsilon_k \lambda^{*2}. \tag{4.8}$$

Let the regularized solution of the large problem corresponding to $\lambda^{(k)*}$, $f_{\lambda^{(k)*}}^{(k)}$, be decomposed as

$$f_{\lambda^{(k)*}}^{(k)} = f_{\lambda^{(k)*}}^{(k)} + \eta_k. \tag{4.9}$$

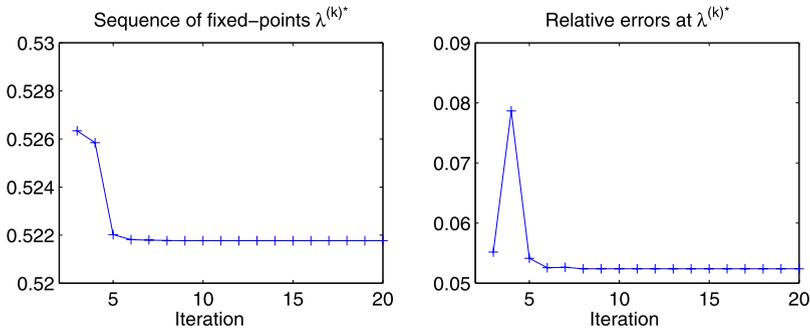


Fig. 3 Verification of Theorem 4.4 using test problem phillips of size $n = 1024$ with $\|e\|_2 = 0.05\|g^{\text{exact}}\|_2$. Note that the relative error stabilizes at the same time as the sequence $\lambda^{(k)*}$ does

Then from the triangular inequality and the decomposition above we have that

$$\|f_{\lambda^{(k)*}}^{(k)} - f^{\text{exact}}\|_2 \leq \|f^{\text{exact}} - f_{\lambda^*}\|_2 + \|f_{\lambda^*} - f_{\lambda^{(k)*}}\|_2 + \|\eta_k\|_2. \tag{4.10}$$

The second term on the right hand side depends on the closeness between $\lambda^{(k)*}$ and λ^* and can readily be bounded using the regularized normal equations and the fact that $\lambda^{(k)*}$ approximates λ^* from above. More specifically, it is easy to prove that

$$\|f_{\lambda^{(k)*}} - f_{\lambda^*}\|_2 \leq \epsilon_k \|f_{\lambda^*}\|_2.$$

Using this inequality in (4.10) we can deduce that

$$\frac{\|f_{\lambda^{(k)*}}^{(k)} - f^{\text{exact}}\|_2}{\|f^{\text{exact}}\|_2} \leq \frac{\|f^{\text{exact}} - f_{\lambda^*}\|_2}{\|f^{\text{exact}}\|_2} + \epsilon_k \frac{\|f_{\lambda^*}\|_2}{\|f^{\text{exact}}\|_2} + \frac{\|\eta_k\|_2}{\|f^{\text{exact}}\|_2}. \tag{4.11}$$

Hence, since both ϵ_k and $\|\eta_k\|_2$ approximate zero as the number of GKB steps grows, which is easy to check from (4.8) and (4.9), it follows that the bound in (4.11) approximates the relative error in $f_{\lambda^*}^*$ from above. This suggests that for ϵ_k small enough the absolute error in $f_{\lambda^{(k)*}}^{(k)}$ should stabilize close to $\|f^{\text{exact}} - f_{\lambda^*}\|_2$, which is the error in $f_{\lambda^{(k)*}}^{(k)}$ if λ^* is reached in k GKB steps; see Theorem 4.4.

Our computational experience is that both the regularization parameter $\lambda^{(k)*}$ and the corresponding regularized solution stabilize in a relatively small number of steps. This is illustrated in Fig. 3. The spike in this figure simply means that the relative error in $f_{\lambda^{(k)*}}^{(k)}$ is unpredictable at the beginning of the GKB iterations even if the regularization parameters are nonincreasing. Further numerical results are presented in the next section.

The stabilization phenomenon illustrated above was first noticed by Hanke and Hansen [19, p. 302] and more recently numerically demonstrated by Hansen [21, pp. 172–173] in connection with a Lanczos-based method combined with TSVD. A similar numerical illustration can also be found in [10]. In all cases however, no mathematical proof is provided. Our contribution here is a theoretical proof of the stabilization phenomenon for GKB-FP.

We emphasize that the theoretical results described in this section hold in exact arithmetic and that in practice, numerical difficulties may arise due to the fact that the Lanczos vectors (the columns of U_k and V_k) lose orthogonality. The cure is to use partial or complete reorthogonalization but extra work is needed. We note, however, that, if the number of GKB steps remains within reasonable bounds, numerical experience shows that the extra work spent with reorthogonalization is not substantial, as we illustrate in the numerical results section. For a discussion on the use of reorthogonalization in connection with a Lanczos-based hybrid algorithm for ill-posed problems, see [5]. A thorough discussion on the issue can be found in Hansen's Book [21, Sect. 6.5, p. 157ff.]; the reader is also referred to the references therein.

Finally, it is worth remarking that the analysis of this section can also be carried out for GKB-FP using Tikhonov regularization in general form. In this case, the bidiagonalization step must be replaced by a joint bidiagonalization step, as done e.g., in [27], which allows one to construct regularized approximate solutions using suitable Krylov subspaces. Implementation details of GKB-FP in this more general version, as well as the corresponding convergence analysis are the subject of future research.

5 Numerical results

The purpose of this section is to illustrate the performance of GKB-FP when applied to typical linear discrete-ill posed problems of moderate to large dimension. For this purpose we choose eight test problems from the Regularization Tools [22] and one problem from the iterative image deblurring package RestoreTools [32]. In each case we generated triples $A, f^{\text{exact}}, g^{\text{exact}}$ so that $Af^{\text{exact}} = g^{\text{exact}}$ and then simulated distinct noisy vectors $g, g = g^{\text{exact}} + e$, where e was generated by the Matlab `randn` function with the seed value set to zero. In our implementation of GKB-FP, the regularized solution f_λ is computed in some cases via (2.8) and in others via the updating formula (2.12); all computations were carried out using Matlab with about 16 significant decimal digits on a PC.

5.1 Test problems from Hansen's toolbox

We consider the test problems

- | | | | |
|-----------------------|---------------|--------------|------------|
| (a) foxgood, | (b) heat, | (c) shaw, | (d) baart, |
| (e) deriv2(example1), | (f) phillips, | (g) gravity, | (h) tomo. |

These involve $n \times n$ linear systems with severely ill-conditioned coefficient matrices with both g^{exact} and f^{exact} known. For the 7 first problems we choose $n = 1200$ and for the eighth one we take $n = 35$, which leads to a linear system of 1225 variables, with the observation that for problem tomo we always use the same data matrix (we recall that for this problem the data matrix depends on random numbers). In all cases the GKB procedure was implemented with complete reorthogonalization; we use noisy vectors g with six distinct noise levels:

$$\|g^{\text{exact}} - g\|_2 / \|g^{\text{exact}}\|_2 = 10^{-6}, 10^{-5}, \dots, 10^{-2}, \text{ and } 5 \times 10^{-2}.$$

5.1.1 Behavior of selected regularization parameter at each GKB iteration

In order to give an idea of what can be expected from GKB-FP in terms of accuracy and speed, we shall compare the regularization parameters that are selected at each GKB step by FP, L-curve, GCV and W-GCV, with the optimal regularization parameter of the large problem. The latter is denoted by λ_{OP} and determined by an exhaustive search using the SVD of the coefficient matrix; a very brief description of W-GCV is given in the next section; see Chung et al. [10] for details.

The result of the comparison for four problems with noise level of 1% is displayed in Fig. 4, where for convenience the optimal regularization parameter is displayed as a constant at each step. Observe that only a few parameters selected by W-GCV are displayed. This is because these parameters are the only to be returned by HyBR, a routine from RestoreTools which implements W-GCV and decides when to stop the GKB procedure. Observe that with the exception of FP, L-curve, GCV and W-GCV select parameters that are not monotonic.

Observe also that the regularization parameters selected by L-curve and GCV may stabilize too late compared with FP, which explains why GKB-FP will not be compared with hybrid methods that use GCV or L-curve to choose regularization param-

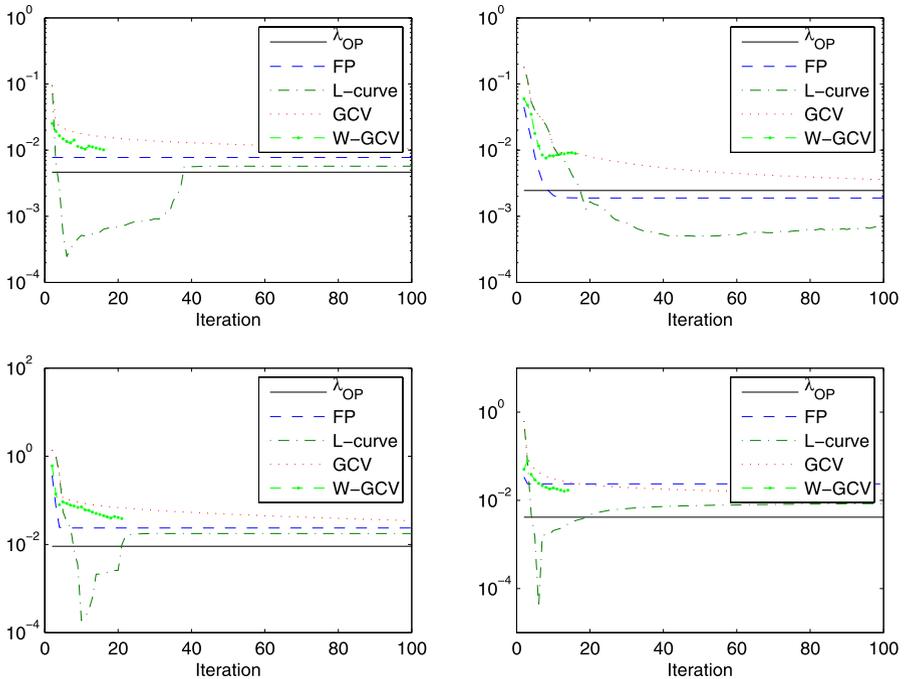


Fig. 4 Regularization parameters selected at each GKB iterations, $2 \leq k \leq 100$, for foxgood (top left), heat (top right), shaw (bottom left), and baart (bottom right). For completeness the optimal regularization parameter of the original problem is denoted here by λ_{OP} and plotted as a constant function along the iterations

ters for the projected problem. A comparison of GKB-FP with a hybrid method that uses W-GCV as selection parameter method at each GKB step is done in Sect. 5.2.

5.1.2 Performance of GKB-FP

In our computations the solution f_λ was computed by using (2.8), and fixed-point computations started with $p = 5$ taking $\lambda_0^{(5)} = 10^{-4}$ as initial guess. Tolerance parameters ϵ_1, ϵ_2 in (3.8), (3.9) were set to 10^{-4} and, in order to accelerate convergence, computation of fixed-points for each $k > p$ was implemented at low precision with $\sqrt{\epsilon_1}$ as tolerance parameter. To illustrate the performance of GKB-FP on the above test problems we ran 50 realizations and then computed average values of regularization parameters, average relative errors, as well as other informative quantities. For comparison purposes, we also compute results associated with the optimal regularization parameter.

In order to illustrate the quality of the computed solutions for distinct noise levels and distinct test problems in a simple and comprehensive way, the attained errors for the first four test problems are displayed in Fig. 5.

For a complete description of the results, computed quantities are labeled according to the following list.

Symbol	Description
NL	Noise level in g .
$\bar{\lambda}_{FP}$	Average value of regularization parameters computed by GKB-FP.
$\bar{\lambda}_{OP}$	Average value of optimal regularization parameters.
\bar{E}	Average relative error in f_λ .
\bar{OE}	Average relative error in $f_{\lambda_{OP}}$.
ρ	Ratio \bar{E}/\bar{OE} .
std_λ	Standard deviation of computed regularization parameters.
k_M	Maximum stopping iteration of 50 runs.
k_ϕ	Maximum number of evaluations of $\phi_1^{(k)}$ along inner iterations.

The results for the first five noise levels are organized in Tables 3, 4, 5, 6, and 7.

As we can observe, the most important features of the SVD-based implementation described in [1, 2], namely the consistent determination of regularization parameters

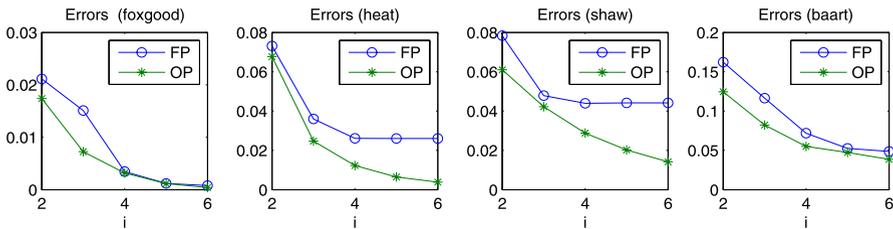


Fig. 5 Relative errors versus $NL = 10^{-i}, i = 2, \dots, 6$, for the first four test problems of the list for noise data such that $\|b - b^{\text{exact}}\|_2 = NL\|b^{\text{exact}}\|_2$. Errors associated with the optimal regularization parameter are denoted here by OP

Table 3 Numerical results for noise level NL = 0.0001%

	foxgood	heat	shaw	baart
$\bar{\lambda}_{FP}$	0.7755e-6	0.2856e-4	0.4106e-4	0.2308e-5
$\bar{\lambda}_{OP}$	0.4624e-4	0.9045e-5	0.7660e-5	0.2568e-5
\bar{E}	0.8025e-3	0.2605e-1	0.4414e-1	0.4869e-1
\overline{OE}	0.4263e-3	0.3886e-2	0.1418e-1	0.3863e-1
ρ	2.2512	6.7129	3.7022	1.3311
std_{λ}	0.1763e-8	0.2679e-8	0.6402e-7	0.3223e-8
$k_M(k_{\phi})$	5(3)	19(5)	8(5)	6(4)
	deriv2	phillips	gravity	tomo
$\bar{\lambda}_{FP}$	0.5440e-5	0.4382e-3	0.2914e-4	0.3499e-1
$\bar{\lambda}_{OP}$	0.1050e-5	0.1333e-2	0.2399e-3	0.7020e-4
\bar{E}	0.1255	0.8476e-2	0.5373e-2	0.1269
\overline{OE}	0.4362e-1	0.4318e-3	0.9342e-3	0.4821e-2
ρ	2.8778	19.8021	6.12394	26.4382
std_{λ}	0.1793e-8	0.4195e-5	0.1092e-6	0.3292e-6
$k_M(k_{\phi})$	17(4)	8(4)	11(4)	47(6)

Table 4 Numerical results for noise level NL = 0.001%

	foxgood	heat	shaw	baart
$\bar{\lambda}_{FP}$	0.7734e-5	0.2862e-4	0.4703e-4	0.2310e-4
$\bar{\lambda}_{OP}$	0.1473e-3	0.3960e-4	0.3345e-4	0.5497e-5
\bar{E}	0.1197e-2	0.2605e-1	0.4415e-1	0.5262e-1
\overline{OE}	0.1172e-2	0.6445e-2	0.2019e-1	0.4731e-1
ρ	1.0385	4.0586	2.3672	1.1384
std_{λ}	0.9569e-8	0.2681e-7	0.5556e-6	0.3553e-7
$k_M(k_{\phi})$	5(3)	19(5)	8(5)	5(4)
	deriv2	phillips	gravity	tomo
$\bar{\lambda}_{FP}$	0.5497e-5	0.2233e-2	0.9833e-4	0.3499e-1
$\bar{\lambda}_{OP}$	0.4156e-2	0.1273e-1	0.1134e-2	0.1104e-2
\bar{E}	0.1255	0.2086e-1	0.7660e-2	0.1269
\overline{OE}	0.6572e-1	0.1097e-2	0.2024e-2	0.7070e-2
ρ	1.9106	19.0954	4.0100	18.6534
std_{λ}	0.1780e-7	0.9037e-4	0.9131e-6	0.4133e-5
$k_M(k_{\phi})$	17(4)	8(4)	10(4)	47(6)

(as explained by the small standard deviation) and the accuracy of computed solutions, are preserved in GKB-FP. Note that in general both the maximum dimension of the Krylov subspace (as described by k_M) in which the computed solution lives and the maximum number of evaluations of function $\phi_1^{(k)}$ (as described by the value of k_{ϕ}) are relatively small.

Table 5 Numerical results for noise level $NL = 0.01\%$

	foxgood	heat	shaw	baart
$\bar{\lambda}_{FP}$	0.7737e-4	0.3413e-4	0.2362e-3	0.2322e-3
$\bar{\lambda}_{OP}$	0.5675e-3	0.2145e-3	0.3297e-3	0.8667e-4
\bar{E}	0.3467e-2	0.2610e-1	0.4396e-1	0.7187e-1
\overline{OE}	0.3193e-2	0.1230e-1	0.2881e-1	0.5510e-1
ρ	1.3316	2.1311	1.5919	1.3132
std_{λ}	0.9183e-7	0.2328e-6	0.1080e-5	0.3160e-6
$k_M(k_{\phi})$	5(3)	19(5)	8(5)	5(4)
	deriv2	phillips	gravity	tomo
$\bar{\lambda}_{FP}$	0.1037e-4	0.2744e-2	0.5938e-3	0.3500e-1
$\bar{\lambda}_{OP}$	0.2453e-4	0.1273e-1	0.4545e-2	0.5835e-2
\bar{E}	0.1310	0.2430e-1	0.7758e-2	0.1270
\overline{OE}	0.9659e-1	0.2794e-2	0.4237e-2	0.2005e-1
ρ	1.3575	8.8564	1.9525	6.4924
std_{λ}	0.1237e-6	0.1628e-4	0.1958e-5	0.4346e-4
$k_M(k_{\phi})$	16(4)	5(4)	10(4)	46(5)

Table 6 Numerical results for noise level $NL = 0.1\%$

	foxgood	heat	shaw	baart
$\bar{\lambda}_{FP}$	0.7737e-3	0.1945e-3	0.2330e-2	0.2332e-2
$\bar{\lambda}_{OP}$	0.2182e-2	0.7385e-3	0.3606e-2	0.8914e-3
\bar{E}	0.1511e-1	0.3600e-1	0.4787e-1	0.1165
\overline{OE}	0.7249e-2	0.2472e-1	0.4221e-1	0.8220e-1
ρ	2.2130	1.4761	1.1574	1.6036
std_{λ}	0.8981e-6	0.1096e-5	0.3990e-5	0.3206e-5
$k_M(k_{\phi})$	5(3)	17(5)	8(5)	5(4)
	deriv2	phillips	gravity	tomo
$\bar{\lambda}_{FP}$	0.8109e-4	0.5533e-2	0.5896e-2	0.3560e-1
$\bar{\lambda}_{OP}$	0.1131e-3	0.3944e-1	0.2265e-1	0.5682e-1
\bar{E}	0.1541	0.2430e-1	0.1138e-1	0.1275
\overline{OE}	0.1423	0.7389e-2	0.9769e-2	0.6232e-1
ρ	1.0843	3.3664	1.2356	2.0516
std_{λ}	0.3718e-6	0.7794e-4	0.1181e-4	0.4248e-3
$k_M(k_{\phi})$	13(4)	6(4)	9(4)	46(6)

Since the computed errors for $NL = 0.05$ (i.e. 5% noise) remained within bounds that do not exceed 30% (except for heat, baart and tomo which are known to be difficult), the numerical results for this noise level are not reported here. Note that the number of GKB steps decreases as the noise level grows. This is in accordance with the regularizing effects of Krylov methods in that at high noise levels more singular

Table 7 Numerical results for noise level NL = 1%

	foxgood	heat	shaw	baart
$\bar{\lambda}_{FP}$	0.7753e-2	0.1907e-2	0.2355e-1	0.2375e-1
$\bar{\lambda}_{OP}$	0.6688e-2	0.2251e-2	0.1155e-1	0.6247e-2
\bar{E}	0.2113e-1	0.7311e-1	0.7834e-1	0.1622
\overline{OE}	0.1740e-1	0.6767e-1	0.6099e-1	0.1244
ρ	1.3724	1.0877	1.3021	1.4735
std_{λ}	0.9208e-5	0.6817e-5	0.3919e-4	0.5568e-4
$k_M(k_{\phi})$	4(3)	14(5)	7(5)	4(4)
	deriv2	phillips	gravity	tomo
$\bar{\lambda}_{FP}$	0.8301e-3	0.5089e-1	0.5897e-1	0.7630e-1
$\bar{\lambda}_{OP}$	0.5285e-3	0.1376	0.9389e-1	0.4799
\bar{E}	0.2208	0.2453e-1	0.2431e-1	0.1712
\overline{OE}	0.2087	0.1986e-1	0.2203e-1	0.1658
ρ	1.0604	1.2908	1.1190	1.0320
std_{λ}	0.2462e-5	0.1124e-3	0.1093e-3	0.3942e-2
$k_M(k_{\phi})$	10(4)	7(4)	9(4)	46(6)

values are swamped and thus the iteration must be stopped earlier (see, e.g., Hanke and Hansen [19]).

Finally, observing the computed relative errors for the various noise levels and the dimension of the Krylov subspace being used, we conclude that GKB-FP is very effective for realistic noise levels (e.g., 0.1% and 0.01%) and therefore competitive .

5.2 Image restoration from RestoreTools: test problem satellite

To test our algorithm on a large-scale problem we choose a deblurring problem from the RestoreTools by Nagy and co-workers [32], where the matrix A plays the role of blur operator. The noisy free data is a “satellite” image in space of 256×256 pixels and stored in a vector $f^{exact} \in \mathbb{R}^{65536}$. The blur operator is thus of size 65536×65536 and the blurred, but noisy free, image associated with f^{exact} is $g^{exact} = A f^{exact}$. The “right-hand” side g , i.e., the blurred and noisy image, is thus $g = g^{exact} + e$ where e is generated by the Matlab function `randn` with the seed value set to zero. The goal here is to compare the performance of our method against a hybrid Lanczos-based method by Chung et al. [10] that selects the regularization parameter by W-GCV at each GKB step. W-GCV was designed so as to overcome well-known difficulties of standard GCV. The basis for W-GCV is the weighted-GCV function

$$G_{A,g}(\omega, \lambda) = \frac{n \|(I - AA_{\lambda}^{\dagger})g\|_2^2}{(\text{trace}(I - \omega AA_{\lambda}^{\dagger}))^2}, \tag{5.1}$$

where $A_{\lambda}^{\dagger} = (A^T A + \lambda^2 I)^{-1} A^T$ and ω is a positive parameter to be determined. In the hybrid method we are interested in, the choice of the regularization parameter is based on the W-GCV function associated with the projected problem (2.4) and on an

Table 8 Results for noise level NL = 0.1%

	FP_p^\perp	W-GCV $_p^\perp$	FP_p	W-GCV $_p$
$\bar{\lambda}$	0.15850e-3	0.3161e-2	0.6590e-3	0.1032e-1
std_λ	0.2834e-4	0.4470e-3	0.6784e-3	0.2219e-2
\bar{E}	0.2820	0.2787	0.2823	0.2822
T_M	40.3537	21.5221	59.2169	46.3883
$k_M(k_\phi)$	54(3)	38(-)	90(3)	90(-)
	FP^\perp	W-GCV $^\perp$	FP	W-GCV
$\bar{\lambda}$	0.1153e-2	0.4476e-2	0.1218e-2	0.4798e-2
std_λ	0.8082e-6	0.5482e-5	0.2207e-4	0.2851e-4
\bar{E}	0.3095	0.2867	0.2999	0.2957
T_M	55.6790	54.8228	56.5264	69.9674
$k_M(k_\phi)$	81(3)	84(-)	103(3)	128(-)

adaptive choice of ω . For simplicity, from here on this hybrid method will be referred to as W-GCV.

Except for quantities associated with λ_{OP} , which is difficult to determine in large-scale problems, we compute the same quantities as in the previous section considering 50 runs as well. W-GCV-based solutions are computed by using the routine HyBR from RestoreTools taking advantage of particular features that allow efficient computation of matrix-vector products; the reader is referred to Chung et al. [10] and Nagy et al. [32] for details. As the satellite problem is a much larger problem than those tested before, to accelerate convergence we also compute W-GCV-based solutions using a preconditioned technique available in RestoreTools, and the same preconditioner is also used to compute GKB-FP-based solutions; for details of the preconditioner, the reader is referred to [32] and references therein. Summarizing, we use four versions of W-GCV and four versions of GKB-FP. The W-GCV versions include one with preconditioning denoted by W-GCV $_p$, one with preconditioning plus reorthonormalization denoted by W-GCV $_p^\perp$, one without preconditioning but with reorthonormalization denoted by W-GCV $^\perp$, and one without reorthonormalization and without preconditioning denoted simply by W-GCV. The GKB-FP versions consider the same variants as W-GCV; for simplicity they are denoted by FP_p^\perp , FP_p , FP^\perp and FP. When GKB-FP is used without reorthonormalization, the solution f_λ is computed by using the updating formula (2.12). For this problem we consider vectors g with three distinct noise levels:

$$\|g^{\text{exact}} - g\|_2 / \|g^{\text{exact}}\|_2 = 0.001, 0.01, \text{ and } 0.05.$$

In order to better evaluate the performance of both GKB-FP and W-GCV, in addition to the quantities computed in the experiment of the previous section, we also compute the maximum time spent by the algorithms along the runs which we denote by T_M (measured in seconds). Computed quantities are summarized in Tables 8, 9, and 10; quantities that help the interpretation of results appear in boldface.

We first note that in the low noise case all versions of the methods behave similarly, with a slight advantage in favor of the preconditioned versions for which the relative

Table 9 Results for noise level NL = 1%

	FP_p^\perp	$W-GCV_p^\perp$	FP_p	$W-GCV_p$
$\bar{\lambda}$	0.53440e-2	0.1876e-1	0.5345e-2	0.1876e-1
std_λ	0.5290e-4	0.7887e-3	0.5172e-4	0.7923e-3
\bar{E}	0.2984	0.2984	0.2984	0.2984
T_M	8.5525	7.7174	8.3127	6.8226
$k_M(k_\phi)$	12(2)	11(-)	12(2)	11(-)
	FP^\perp	$W-GCV^\perp$	FP	$W-GCV$
$\bar{\lambda}$	0.5730e-2	0.3119e-1	0.6353e-2	0.3119e-2
std_λ	0.3800e-5	0.7439e-4	0.3230e-4	0.6430e-4
\bar{E}	0.3266	0.3971	0.3201	0.3975
T_M	41.5397	20.6554	39.3188	21.3298
$k_M(k_\phi)$	62(3)	11(-)	74(3)	45(-)

Table 10 Results for noise level NL = 5%

	FP_p^\perp	$W-GCV_p^\perp$	FP_p	$W-GCV_p$
$\bar{\lambda}$	0.27690e-1	0.5487e-1	0.2769e-1	0.5487e-1
std_λ	0.1730e-3	0.5837e-3	0.1730e-3	0.5837e-3
\bar{E}	0.3773	0.3772	0.3773	0.3772
T_M	3.7735	3.840	3.7823	3.8044
$k_M(k_\phi)$	5(2)	4(-)	5(2)	4(-)
	FP^\perp	$W-GCV^\perp$	FP	$W-GCV$
$\bar{\lambda}$	0.3221e-1	0.1010	0.3250e-1	0.8735e-1
std_λ	0.2443e-4	0.5895e-1	0.2707e-4	0.6835e-1
\bar{E}	0.4065	0.4864	0.4032	0.4864
T_M	24.6873	72.7300	25.6511	162.9850
$k_M(k_\phi)$	41(3)	101(-)	49(3)	234(-)

error approaches 28%, see Table 8. Regarding these results, note that although the relative error is much larger than the noise to signal ratio (0.1%), the computed solutions yield images that resemble the true and noise free image relatively well. This is shown in Fig. 6 where we have depicted some results of the first run. Unfortunately, the behavior of the methods changes when the noise level is increased: while the accuracy of W-GCV starts to deteriorate more or less significantly in the higher noise cases, the quality of solutions produced by the other tested versions (as illustrated by the average relative errors) remain comparable, see Tables 9 and 10.

We continue our analysis of the results by examining the computed regularization parameters. The conclusion in this case is that W-GCV determines parameters that can vary very much from one run to another (as explained by the values of std_λ), while GKB-FP yields parameters (and corresponding solutions) with small variations. A consequence of this fact is that the time spent by the GKB-FP versions (as illustrated by the time spent by the methods) does not vary so much from a run to

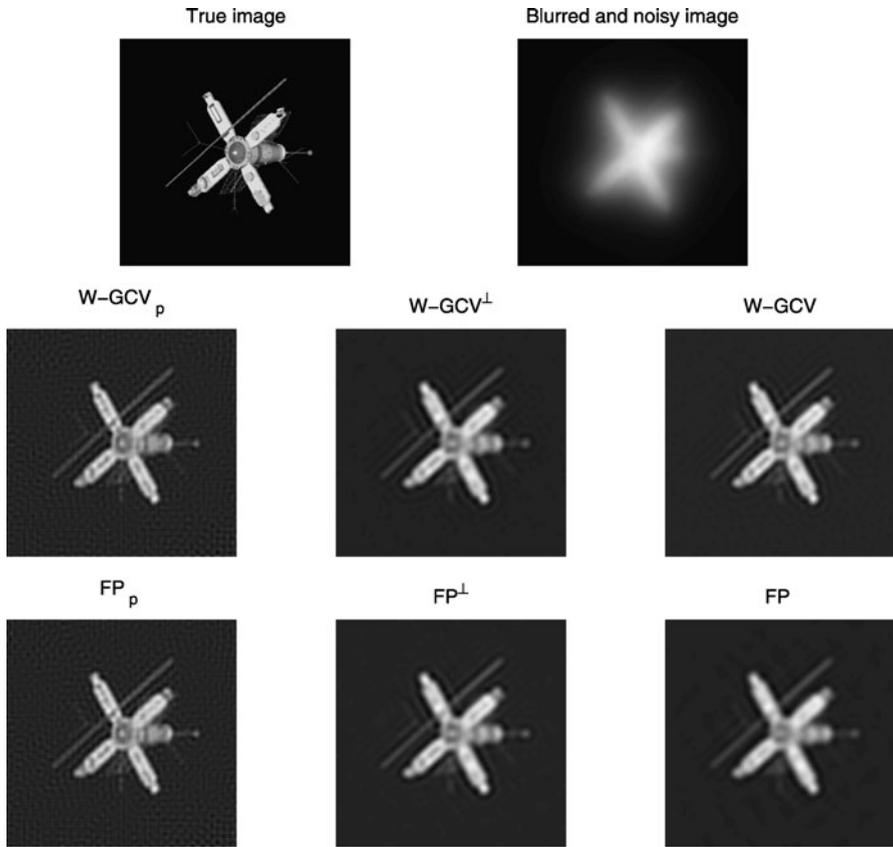


Fig. 6 Satellite data and reconstructed images for noise level 0.1% obtained in the first run

another, whereas the time spent by the W-GCV versions can vary from small, which also holds for GKB-FP, to too large when compared to the time spent by GKB-FP. A similar comment applies to the dimension of the Krylov subspace being used.

Note that at high noise levels the subspace dimension, the computational work and the quality of computed solutions by the preconditioned versions of both W-GCV and GKB-FP are all comparable. However, the performance of the GKB-FP versions without preconditioning in terms of accuracy is superior, see Tables 9 and 10, and Fig. 7.

Another observation in connection with GKB-FP, which is in accordance with regularization theory, is that the number of outer iterations in GKB-FP that can reliably be performed decreases when the noise level increases, and a similar comment applies to the time spent by GKB-FP. However, this no longer holds for k_M and T_M in connection with W-GCV; compare the values of k_M and T_M in Tables 9 and 10.

The stabilization phenomenon attributed to GKB-FP is illustrated in Fig. 8, where are displayed both the regularization parameters and error histories for GKB-FP and LSQR. Finally, observe that the maximum number of evaluations of $\phi_1^{(k)}$ (as indicated

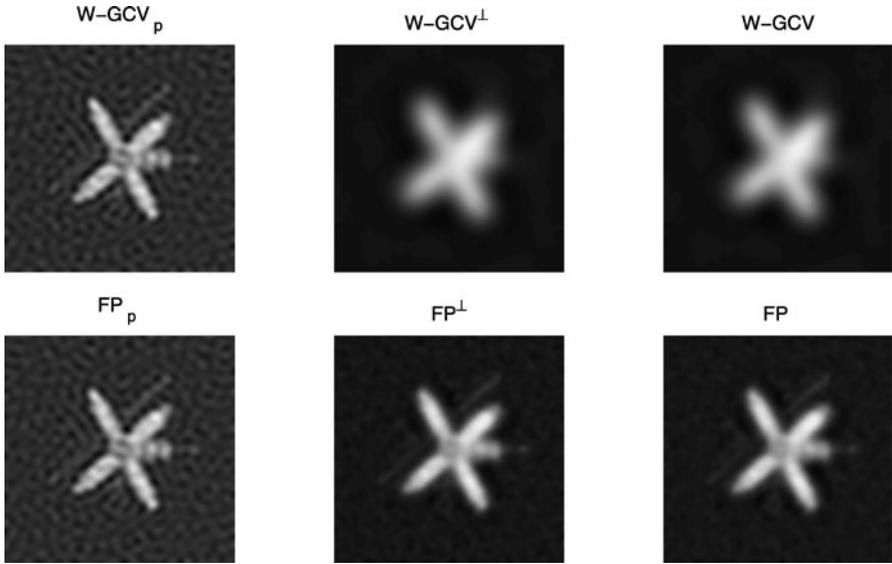


Fig. 7 Reconstructed images for noise level 5% obtained in the last run

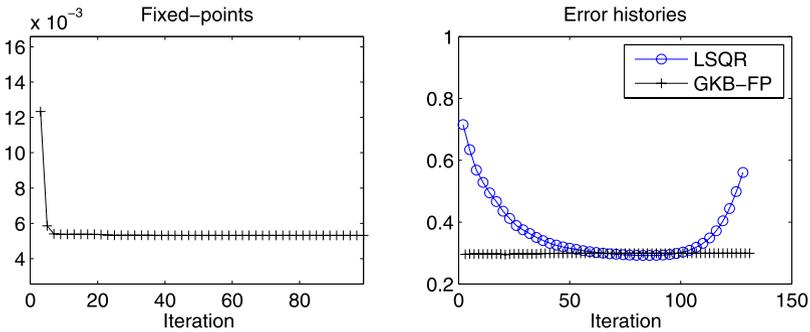


Fig. 8 Stabilization of regularization parameter determined by GKB-FP and relative error for GKB-FP and LSQR. Displayed results correspond to FP^\perp for noise level of 1%

by k_ϕ) along inner iterations does not exceed 3; this confirms what we commented in Sect. 3 and indicates again that GKB-FP is competitive.

Based on the theory of the previous section and the analysis above we conclude that theoretically GKB-FP is much more robust than W-GCV, that the preconditioned versions of W-GCV for the tested problems perform similarly as their GKB-FP counterparts, and that GKB-FP can in some circumstances yield more accurate regularized solutions than those produced by W-GCV (see Tables 9 and 10 again).

6 Conclusions

We have reviewed the fixed-point algorithm for Tikhonov regularization of Bazán [1] and discussed/analyzed an implementation of it for large scale problems, which resulted in a new method called GKB-FP. The method combines Lanczos iterations with Tikhonov regularization in the generated Krylov subspace with the regularization parameter for the projected problem chosen by the FP-algorithm. Additionally, we compared the performance of GKB-FP against a hybrid Lanczos-based method that selects the regularization parameter by W-GCV at each GKB step, and concluded that in terms of accuracy, consistency, and computational effort, the preconditioned versions of the latter are as robust as their GKB-FP counterparts, with the observation that the GKB-FP can in some circumstances produce more accurate results than W-GCV. Furthermore, we proved that GKB-FP is able to *completely stabilize* the iterations, a property already seen in the literature in connection with hybrid methods but not theoretically proved so far.

It is clear that continued experience with GKB-FP is necessary to fully assess its potential. In particular, experience is needed with large problems from distinct application areas requiring the use of general-form Tikhonov regularization. This is the subject of ongoing research.

Acknowledgements The authors wish to thank J. Chung for providing information on both the stopping criteria for W-GCV and the code that implements W-GCV. Thanks also go to the reviewers for helpful suggestions that improved the presentation of this work.

References

1. Bazán, F.S.V.: Fixed-point iterations in determining the Tikhonov regularization parameter. *Inverse Probl.* **24**, 035001 (2008)
2. Bazán, F.S.V., Francisco, J.B.: An improved fixed-point algorithm for determining a Tikhonov regularization parameter. *Inverse Probl.* **25**, 045007 (2009)
3. Belge, M., Kilmer, M.E., Miller, E.L.: Efficient determination of multiple regularization parameters in a generalized L-curve framework. *Inverse Probl.* **18**, 1161–1183 (2002)
4. Björck Å.: A bidiagonalization algorithm for solving ill-posed systems of linear equations. *BIT Numer. Math.* **28**, 659–670 (1988)
5. Björck Å., Grimme, E., Van Dooren, P.: An implicit shift bidiagonalization algorithm for ill-posed systems of linear equations. *BIT Numer. Math.* **34**, 510–534 (1994)
6. Calvetti, D., Golub, G.H., Reichel, L.: Estimation of the L-curve via Lanczos bidiagonalization. *BIT Numer. Math.* **39**, 603–619 (1999)
7. Calvetti, D., Reichel, L.: Tikhonov regularization of large scale problems. *BIT Numer. Math.* **43**, 263–283 (2003)
8. Calvetti, D., Reichel, L., Shuibi, A.: L-curve and curvature bounds for Tikhonov regularization. *Numer. Algorithms* **35**, 301–314 (2004)
9. Calvetti, D., Lewis, B., Reichel, L.: On the regularizing properties of the GMRES method. *Numer. Math.* **91**, 605–625 (2002)
10. Chung, J., Nagy, J.G., O’Leary, D.P.: A weighted-GCV method for Lanczos-hybrid regularization. *Electron. Trans. Numer. Anal.* **28**, 149–167 (2008)
11. Eldén, L.: Algorithms for the regularization of ill-conditioned least squares problems. *BIT Numer. Math.* **17**, 134–145 (1977)
12. Engl, H.W., Hanke, M., Neubauer, A.: *Regularization of Inverse Problems. Mathematics and Its Applications*, vol. 375. Kluwer Academic, Dordrecht (1996)
13. Frommer, A., Maass, P.: Fast CG-based methods for Tikhonov–Phillips regularization. *SIAM J. Sci. Comput.* **20**, 1831–1850 (1999)

14. Golub, G.H., Heath, M., Wahba, G.: Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics* **21**, 215–222 (1979)
15. Golub, G.H., Kahan, W.: Calculating the singular values and pseudo-inverse of a matrix. *SIAM J. Numer. Anal. Ser. B* **2**, 205–224 (1965)
16. Golub, G.H., von Matt, U.: Tikhonov regularization for large scale problems. In: Golub, G.H., Lui, S.H., Luk, F., Plemmons, R.J. (eds.) *Workshop on Scientific Computing*, pp. 3–26. Springer, New York (1997)
17. Hämarik, U., Raus, T.: On the choice of the regularization parameter in ill-posed problems with approximately given noise level of data. *J. Inverse Ill-Posed Probl.* **14**(3), 251–266 (2006)
18. Hämarik, U., Palm, R., Raus, T.: Use of extrapolation in regularization methods. *J. Inverse Ill-Posed Probl.* **15**(3), 277–294 (2007)
19. Hanke, M., Hansen, P.C.: Regularization methods for large-scale problems. *Surv. Math. Ind.* **3**, 253–315 (1993)
20. Hansen, P.C., O’Leary, D.P.: The use of the L-curve in the regularization of discrete ill-posed problems. *SIAM J. Sci. Comput.* **14**, 1487–1503 (1993)
21. Hansen, P.C.: *Rank-Deficient and Discrete Ill-Posed Problems*. SIAM, Philadelphia (1998)
22. Hansen, P.C.: Regularization tools: A MATLAB package for analysis and solution of discrete ill-posed problems. *Numer. Algorithms* **6**, 1–35 (1994)
23. Hnětynková, I., Plešinger, M., Strakoš, Z.: The regularization effect of the Golub-Kahan iterative bidiagonalization and revealing the noise level. *BIT Numer. Math.* **49**, 669–696 (2009)
24. Jiang, M., Xia, L., Shou, G., Liu, F., Crozier, S.: Two hybrid regularization frameworks for solving the electrocardiography inverse problem. *Phys. Med. Biol.* **53**, 5151–5164 (2008)
25. Johnston, P.R., Gulrajani, R.M.: An analysis of the zero-crossing method for choosing regularization parameter. *SIAM J. Sci. Comput.* **24**(2), 428–442 (2002)
26. Krawczy-Stando, D., Rudnicki, M.: Regularization parameter selection in discrete ill-posed problems—the use of the U-curve. *Int. J. Appl. Math. Comput. Sci.* **17**(2), 157–164 (2007)
27. Kilmer, M.E., Hansen, P.C., Español, M.L.: A projected-based approach to general-form Tikhonov regularization. *SIAM J. Sci. Comput.* **29**(1), 315–330 (2007)
28. Kilmer, M.E., O’Leary, D.P.: Choosing regularization parameters in iterative methods for ill-posed problems. *SIAM J. Matrix Anal. Appl.* **22**, 1204–1221 (2001)
29. Kilmer, M.E., Stewart, G.W.: Iterative regularization and MINRES. *SIAM J. Matrix Anal. Appl.* **21**, 613–628 (1999)
30. Morigi, S., Reichel, L., Sgallari, F., Zama, F.: Iterative methods for ill-posed problems and semiconvergent sequences. *J. Comput. Appl. Math.* **193**(1), 157–167 (2006)
31. Morozov, V.A.: *Regularization Methods for Solving Incorrectly Posed Problems*. Springer, New York (1984)
32. Nagy, J.G., Palmer, K., Perrone, L.: Iterative methods for image deblurring: a MATLAB object oriented approach. *Numer. Algorithms* **36**, 73–93 (2004)
33. O’Leary, D.P., Simmons, J.A.: A bidiagonalization-regularization procedure for large scale discretizations of ill-posed problems. *SIAM J. Sci. Stat. Comput.* **2**, 474–489 (1981)
34. Paige, C.C., Saunders, M.A.: LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Softw.* **8**(1), 43–71 (1982)
35. Paige, C.C., Saunders, M.A.: Algorithm 583. LSQR: Sparse linear equations and least squares problems. *ACM Trans. Math. Softw.* **8**(2), 195–209 (1982)
36. Regińska, T.: A regularization parameter in discrete ill-posed problems. *SIAM J. Sci. Comput.* **3**, 740–749 (1996)
37. Rust, B.W., O’Leary, D.P.: Residual periodograms for choosing regularization parameters for ill-posed problems. *Inverse Probl.* **24**, 034005 (2008)
38. Tikhonov, A.N.: Solution of incorrectly formulated problems and the regularization method. *Sov. Math. Dokl.* **4**, 1035–1038 (1963)
39. Zibetti, M.V.W., Bazán, F.S.V., Mayer, J.: Determining the regularization parameters for super-resolution problems. *Signal Process.* **88**, 2890–2901 (2008)