



Hybrid enriched bidiagonalization for discrete ill-posed problems

Per Christian Hansen¹ | Yiqiu Dong¹ | Kuniyoshi Abe²

¹Department of Applied Mathematics and Computer Science, Technical University of Denmark, Kgs. Lyngby, Denmark

²Faculty of Economics and Information, Gifu Shotoku Gakuen University, Gifu, Japan

Correspondence

Per Christian Hansen, Department of Applied Mathematics and Computer Science, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark.
Email: pcha@dtu.dk

Funding information

European Research Council, Grant/Award Number: 291405; National Natural Science Foundation of China, Grant/Award Number: 11701388

Summary

The regularizing properties of the Golub–Kahan bidiagonalization algorithm are powerful when the associated Krylov subspace captures the dominating components of the solution. In some applications the regularized solution can be further improved by enrichment, that is, by augmenting the Krylov subspace with a low-dimensional subspace that represents specific prior information. Inspired by earlier work on GMRES, we demonstrate how to carry these ideas over to the bidiagonalization algorithm, and we describe how to incorporate Tikhonov regularization. This leads to a hybrid iterative method where the choice of regularization parameter in each iteration also provides a stopping rule.

KEY WORDS

enriched subspaces, hybrid iterative methods, Krylov subspace methods, regularizing iterations

1 | INTRODUCTION

We are concerned with iterative Krylov subspace methods for solving large ill-conditioned systems on linear equations, arising from discretization of inverse problems, of the form

$$\min_x \|Ax - b\|_2^2, \quad A \in \mathbb{R}^{m \times n}. \quad (1)$$

Such problems arise in many areas of science and engineering,¹ and it is important to have access to efficient algorithms that can handle large-scale inverse problems. To compute a stable solution to such problems, one must incorporate prior information about the desired solution. One often chooses a variational formulation known as Tikhonov regularization,

$$\min_x \{\|Ax - b\|_2^2 + \lambda^2 \mathcal{R}(x)\}.$$

Here, $\mathcal{R}(x)$ is a regularization or smoothness term that penalizes unwanted features in the solution, and λ is a user-chosen regularization parameter.¹

Instead of enforcing smoothness conditions on the solution, one may have prior information that can be specified in the form of a low-dimensional subspace in which the solution must lie.¹ This leads to a projection formulation of the form

$$\min_x \|Ax - b\|_2^2 \quad \text{s.t.} \quad x \in S_k, \quad (2)$$

where the *signal subspace* S_k is a linear subspace of dimension k . If the basis of $S_k = \text{span}\{v_1, v_2, \dots, v_k\}$ is chosen such that it captures the main features in the solution, then this approach can be very useful. The latter approach is particularly

attractive for large-scale problems, where the signal subspace can take the form of a Krylov subspace, such as

$$\mathcal{K}_k = \text{span}\{A^T b, A^T A A^T b, (A^T A)^2 A^T b, \dots, (A^T A)^{k-1} A^T b\} \quad \text{for the CGLS and LSQR algorithms},^{1,2} \quad (3)$$

$$\bar{\mathcal{K}}_k = \text{span}\{b, Ab, A^2 b, \dots, A^{k-1} b\} \quad \text{for the GMRES and MINRES algorithms},^{3,4} \quad (4)$$

$$\vec{\mathcal{K}}_k = \text{span}\{Ab, A^2 b, A^3 b, \dots, A^k b\} \quad \text{for the RRGMRES and MR-II algorithms},^{5-8} \quad (5)$$

where k is the number of iterations; such methods are sometimes referred to as *regularizing iterations*. RRGMRES can be implemented with reorthogonalization⁵ as well as without⁶; the latter requires more memory. Since the publication of the original implementation of MR-II,⁷ it was shown⁸ how to implement this algorithm in a more stable fashion.

Depending on the application, one or more of these subspaces may be well suited to compute a good regularized solution, that is, a good approximation that is only little sensitive to perturbations of the data.⁹ Moreover, it is possible to combine the projection formulation with Tikhonov regularization; this leads to the so-called hybrid iterative methods^{1,10} where Tikhonov regularization is applied in each iteration of the algorithm to the *projected problem*, that is, the problem of small dimensions in the relevant Krylov subspace.

We can further improve the regularized solution by incorporating additional specific prior information. In this work, we assume that the solution has a significant component in a given subspace \mathcal{W}_p of dimension $p \ll k$ (e.g., chosen to represent known features in the solution). In connection with the above Krylov subspace methods, it was proposed^{11,12} to decompose the solution into a component in \mathcal{W}_p and another component in the orthogonal complement \mathcal{W}_p^\perp , which leads to the idea of *enriched (or augmented) Krylov subspace methods*. See also the work of Kuroiwa et al.¹³ for an adaptive approach.

Recently, we have presented an algorithm R³GMRES¹⁴ based on the range-restricted GMRES (RRGMRES) method⁵ and the corresponding Krylov subspace $\vec{\mathcal{K}}_k$. In the present work, we consider a similar approach based on the LSQR method and the underlying Golub–Kahan bidiagonalization algorithm,¹⁵ and the corresponding Krylov subspace \mathcal{K}_k . Specifically, we compute regularized solutions in a signal subspace $\mathcal{S}_{p,k}$ that is the direct sum of the two subspaces \mathcal{W}_p and \mathcal{K}_k ,

$$\mathcal{S}_{p,k} = \mathcal{W}_p + \mathcal{K}_k \equiv \{y + z \mid y \in \mathcal{W}_p \wedge z \in \mathcal{K}_k\}, \quad (6)$$

which itself is a linear subspace.

As shown by many numerical examples,^{11–14} the inclusion of a well-chosen enrichment subspace \mathcal{W}_p can dramatically improve the regularized solution. However, we want to emphasize that this subspace should be chosen with care, based on prior knowledge of the solution; if it is chosen incorrectly (i.e., it includes undesired components), then any enriched Krylov subspace method can give very incorrect results.

An efficient and stable algorithm, ENRICHED CGNR for this problem, based on the CGLS algorithm, was published in the work of Calvetti et al.¹² Here, we present an alternative algorithm that takes its basis in the LSQR algorithm and the underlying bidiagonalization process. Due to this formulation, our algorithm lends itself easily to extensions to hybrid iterative algorithms where an additional Tikhonov regularization term is incorporated, and where it is possible to efficiently compute the corresponding regularization parameter in each iteration. The latter also provides a stopping rule for the iterations, when the regularization parameter stabilizes.

Our paper is organized as follows. We first present the basic enriched bidiagonalization algorithm and discuss how to implement it. Then, we introduce Tikhonov regularization, which leads to the hybrid version of the algorithm, and again, we discuss implementation aspects. We conclude with a few illustrative numerical examples. Throughout, I_q denotes an identity matrix of order q .

2 | FORMULATION OF THE BASIC ALGORITHM

We want to solve the problem

$$\min_x \|Ax - b\|_2^2 \quad \text{s.t.} \quad x \in \mathcal{W}_p + \mathcal{K}_k. \quad (7)$$

In principle, we could use, say, a Hessenberg decomposition

$$A [W_p, A^T b, A^T A A^T b, \dots, (A^T A)^{k-1} A^T b] = V_{p+k+1} H_{p+k}$$

and compute the solution as

$$\begin{aligned} x^{(k)} &= [W_p, A^T b, A^T A A^T b, \dots, (A^T A)^{k-1} A^T b] y^{(k)}, \\ y^{(k)} &= \operatorname{argmin}_y \|H_{p+1} y - V_{p+k+1}^T b\|_2^2. \end{aligned}$$

However, we prefer to use a stable and efficient “standard” algorithm. Following the ideas behind the R³GMRES algorithm¹⁴ for square matrices, we use the bidiagonalization algorithm to compute an orthonormal basis of \mathcal{K}_k , and augment it by \mathcal{W}_p in each iteration of the algorithm. This may seem cumbersome, but the overhead is, in fact, favorably small.

At iteration k , we have the decomposition

$$A[V_k, W_p] = \begin{bmatrix} U_{k+1}, \tilde{U}_k \end{bmatrix} \begin{bmatrix} B_k & G_k \\ 0 & F_k \end{bmatrix}, \quad (8)$$

where V_k , U_{k+1} and B_k are associated with the classical bidiagonalization algorithm. The remaining matrices are associated with the enrichment/augmentation. Specifically,

- $A V_k = U_{k+1} B_k$ is obtained after k iterations of the bidiagonalization process;
- $V_k \in \mathbb{R}^{n \times k}$ has orthonormal columns that span \mathcal{K}_k ;
- $U_{k+1} \in \mathbb{R}^{m \times (k+1)}$ has orthonormal columns, and $u_1 = b/\|b\|_2$;
- $B_k \in \mathbb{R}^{(k+1) \times k}$ is a lower bidiagonal matrix;
- $\tilde{U}_k \in \mathbb{R}^{m \times p}$ satisfies $\operatorname{range}(AW_p) = \operatorname{range}(U_{k+1}G_k + \tilde{U}_kF_k)$ and $\tilde{U}_k^T U_{k+1} = 0$;
- $F_k \in \mathbb{R}^{p \times p}$ and *changes in every iteration*;
- G_k is $(k+1) \times p$ and is *updated* associated with U_{k+1} .

The columns of $[V_k, W_p]$ form a basis for $S_{p,k}$. The matrices $G_k \in \mathbb{R}^{(k+1) \times p}$ and $F_k \in \mathbb{R}^{p \times p}$ are composed of the coefficients of AW_p with respect to basis of $\operatorname{range}(U_{k+1})$ and $\operatorname{range}(\tilde{U}_k)$, respectively:

$$G_k = U_{k+1}^T AW_p, \quad F_k = \tilde{U}_k^T AW_p. \quad (9)$$

Then, the iterate $x^{(k)} \in S_{p,k}$ is given by $x^{(k)} = [V_k, W_p] y^{(k)}$, where $y^{(k)}$ solves a least squares problem:

$$y^{(k)} = \operatorname{argmin}_y \left\| \begin{bmatrix} B_k & G_k \\ 0 & F_k \end{bmatrix} y - \begin{bmatrix} U_{k+1}^T \\ \tilde{U}_k^T \end{bmatrix} b \right\|_2^2. \quad (10)$$

The above derivation leads to the following generic formulation.

BASIC ENRICHED BIDIAGONALIZATION

1. Set $U_1 = b/\|b\|_2$, $V_0 = []$, $B_0 = []$, $G_0 = U_1^T AW_p$, $k = 1$.
2. Use the bidiagonalization process to obtain v_k and u_{k+1} such that $A V_k = U_{k+1} B_k$,
where $V_k = [V_{k-1}, v_k]$, $U_{k+1} = [U_k, u_{k+1}]$, $B_k = \begin{bmatrix} B_{k-1} & 0 \\ \times & \times \end{bmatrix}$.
3. Compute $G_k = \begin{bmatrix} G_{k-1} \\ u_{k+1}^T AW_p \end{bmatrix} \in \mathbb{R}^{(k+1) \times p}$.
4. Orthonormalize AW_p with respect to U_{k+1} to obtain $\tilde{U}_k \in \mathbb{R}^{m \times p}$.
5. Compute $F_k = \tilde{U}_k^T AW_p \in \mathbb{R}^{p \times p}$.
6. Solve $\min_y \left\| \begin{bmatrix} B_k & G_k \\ 0 & F_k \end{bmatrix} y - \begin{bmatrix} U_{k+1}^T \\ \tilde{U}_k^T \end{bmatrix} b \right\|_2^2$ to obtain $y^{(k)}$.
7. Then, $x^{(k)} = [V_k, W_p] y^{(k)}$.
8. Stop, or set $k := k + 1$ and return to Step 2.

We note that we need to recompute the skinny $m \times p$ matrix \tilde{U}_k and the small $p \times p$ matrix F_k in each iteration, but the dimension p of the augmentation subspace is small so this overhead is negligible.

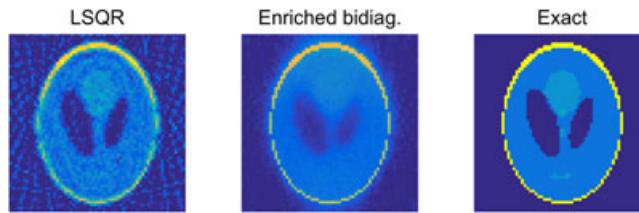


FIGURE 1 The Shepp–Logan head phantom and two reconstructions in which negative pixels are truncated to 0. The LSQR reconstruction shows the usual line artifacts associated with few-projection computed tomography problems. On the other hand, the reconstruction computed with the enriched bidiagonalization method (as well as enriched CGNR) has much fewer artifacts

For efficiency reasons when solving the least squares problem in Step 6, in each iteration, we update the orthogonal factorization:

$$\begin{bmatrix} B_k & G_k \\ 0 & F_k \end{bmatrix} = Q \begin{bmatrix} T_k^{(11)} & T_k^{(12)} \\ 0 & T_k^{(22)} \\ 0 & 0 \end{bmatrix},$$

where $T_k^{(11)} \in \mathbb{R}^{k \times k}$ and $T_k^{(22)} \in \mathbb{R}^{p \times p}$ are upper triangular, and $Q \in \mathbb{R}^{(k+p+1) \times (k+p+1)}$ is orthogonal. We update $T_k^{(11)}$ via Givens rotations that are also applied to G_k and $U_{k+1}^T b$. The matrix \tilde{U}_k is already orthogonal to U_k ; hence (in principle), we can perform the update $\tilde{U}_{k+1} = (I_m - u_{k+1} u_{k+1}^T) \tilde{U}_k$. For numerical stability, we must reorthogonalize the columns of V_k , U_{k+1} , and \tilde{U}_k . This is an acceptable approach used in many similar algorithms, such as the algorithm HyBR.¹⁶

We emphasize that the iterates produced by our algorithm are mathematically identical to those produced by the enriched CGNR method.¹²

To illustrate the power of incorporating additional information via the augmentation subspace \mathcal{W}_p , we consider a parallel-beam X-ray computed tomography (CT) problem generated by means of the function `paralleltom` from the software package AIR TOOLS II.¹⁷ In this example, the 64×64 test image is the Shepp–Logan head phantom and our prior information is that the image has a particular oval structure that represents the skull. Hence, we can choose binary basis vectors of \mathcal{W}_p that represent this structure.

We use 22 projections of 91 pixels, so the matrix is $2,002 \times 4,096$ and the problem is underdetermined. We add 1% Gaussian noise to the data. We then solve the CT reconstruction problem by means of LSQR and the basic enriched bidiagonalization algorithm. The results are shown in Figure 1 where we see that incorporation of the specific information about the skull gives a much improved reconstruction, compared to that from the LSQR algorithm.

3 | FORMULATION OF THE HYBRID ALGORITHM

For some inverse problems, the Krylov subspace or the enriched subspace may not provide enough regularization, and an additional regularization term $\lambda^2 \mathcal{R}(x)$ is needed. Here, we consider the case where $\mathcal{R}(x) = \|x\|_2^2$ and the corresponding problem

$$\min_x \left\{ \|Ax - b\|_2^2 + \lambda^2 \|x\|_2^2 \right\} \quad \text{s.t.} \quad x \in S_{p,k}. \quad (11)$$

After reformulating (11) as

$$\min_x \left\{ \left\| \begin{bmatrix} A \\ \lambda I \end{bmatrix} x - \begin{bmatrix} b \\ 0 \end{bmatrix} \right\|_2^2 \right\} \quad \text{s.t.} \quad x \in S_{p,k}, \quad (12)$$

both the enriched CGNR algorithm and our enriched bidiagonalization algorithm can handle this problem efficiently by adding a Tikhonov regularization term to the projected problem in each iteration. The details on how such a term is added to the LSQR algorithm can be found in section 6.4 and fig. 6.10 of the work of Hansen.¹

A further advantage of adding a regularization term to the projected problem is that, in some case, we can choose a different regularization parameter λ_k in each iteration. Moreover, because the projected problems have much smaller dimensions than the original problem (1), we are free to use a variety of parameter-choice rules, including those that require the computation of the SVD of the projected matrix. This is often referred to as a *hybrid iterative method* and it is the basis of the HyBR algorithm¹⁶ that applies this idea to the LSQR algorithm, using weighted generalized cross validation

to choose the regularization parameter. Similar hybrid iterative algorithms based on (flexible) GMRES are implemented in the IR TOOLS package.¹⁸

The way that Tikhonov regularization is introduced in the enriched CGNR algorithm¹² does not allow the regularization parameter to change in each iteration. On the other hand, because our algorithm explicitly builds on the bidiagonalization that also underlies LSQR, we can indeed introduce an iteration-dependent regularization parameter λ_k . Specifically, we obtain the following algorithm where PCR denotes the chosen parameter-choice rule.

HYBRID ENRICHED BIDIAGONALIZATION

1. Set $U_1 = b/\|b\|_2$, $V_0 = []$, $B_0 = []$, $G_0 = U_1^T A W_p$, $k = 1$.
2. Use the bidiagonalization process to obtain v_k and u_{k+1} such that $A V_k = U_{k+1} B_k$,
where $V_k = [V_{k-1}, v_k]$, $U_{k+1} = [U_k, u_{k+1}]$, $B_k = \begin{bmatrix} B_{k-1} & 0 \\ 0 & \times \end{bmatrix}$.
3. Compute $G_k = \begin{bmatrix} G_{k-1} \\ u_{k+1}^T A W_p \end{bmatrix} \in \mathbb{R}^{(k+1) \times p}$.
4. Orthonormalize $A W_p$ with respect to U_{k+1} to obtain $\tilde{U}_k \in \mathbb{R}^{m \times p}$.
5. Compute $F_k = \tilde{U}_k^T A W_p \in \mathbb{R}^{p \times p}$.
6. Use PCR to choose λ_k and compute $y_{\lambda_k}^{(k)}$ by solving the regularized problem $\min_y \left\| \begin{bmatrix} B_k & G_k \\ 0 & F_k \\ \lambda_k V_k & \lambda_k W_p \end{bmatrix} y - \begin{bmatrix} U_{k+1}^T \\ \tilde{U}_k^T \\ 0 \end{bmatrix} b \right\|_2^2$.
7. Then, $x^{(k)} = [V_k, W_p] y_{\lambda_k}^{(k)}$.
8. Stop, or set $k := k + 1$ and return to Step 2.

Following the HyBR algorithm¹⁶ as well as the hybrid iterative algorithms in IR TOOLS,¹⁸ we stop the iterations when the regularization parameter λ_k stabilizes.

To solve the regularized problem in Step 6 efficiently, we already described how to update a QR factorization of the top 2×2 block matrix. To treat the large bottom block $\lambda_k[V_k, W_p] \in \mathbb{R}^{n \times (k+p)}$ efficiently, we multiply from the left with the orthogonal matrix $[V_k, V_o]^T$, where the columns of V_o are chosen to make the full matrix orthogonal. The 2-norm is unaffected by this transformation, and the bottom block then takes the form

$$[V_k, V_o]^T \lambda_k [V_k, W_p] = \lambda_k \begin{bmatrix} I_k & V_k^T W_p \\ 0 & V_o^T W_p \end{bmatrix}, \quad V_k^T W_p = \begin{bmatrix} V_{k-1}^T W_p \\ v_k^T W_p \end{bmatrix},$$

where we used that $V_k = [V_{k-1}, v_k]$ and $W_p^T v_k$ is a vector of length p . Because the matrix V_o is not explicitly available, we consider instead the Cholesky factorization of the symmetric and positive definite $p \times p$ matrix

$$(V_o^T W_p)^T V_o^T W_p = W_p^T V_o V_o^T W_p = W_p^T (I_n - V_k V_k^T) W_p = R_k^T R_k,$$

where $R_k \in \mathbb{R}^{p \times p}$ is the Cholesky factor. It follows immediately that

$$R_k^T R_k = W_p^T (I_n - V_{k-1} V_{k-1}^T) W_p - (W_p^T v_k) (W_p^T v_k)^T = R_{k-1}^T R_{k-1} - (W_p^T v_k) (W_p^T v_k)^T.$$

Hence, we can compute R_k from R_{k-1} using techniques that *downdate* a Cholesky factor due to a rank-one change,¹⁹ and the bottom block is reduced to a small matrix of size $(k+p) \times (k+p)$. We emphasize that downdating is always potentially unstable and needs to be implemented carefully for numerical stability reasons.^{19,20}

4 | NUMERICAL EXAMPLES

To illustrate the performance of our hybrid enriched bidiagonalization (HEB) algorithm, we use the following approach.

1. Generate a noise-free system: $A x_{\text{exact}} = b_{\text{exact}}$.
2. Add noise: $b = b_{\text{exact}} + e$ where e is a random vector of Gaussian white noise scaled such that $\|e\|_2/\|b_{\text{exact}}\|_2 = \eta$.
3. Show the best solution within the iterations, the relative error $\|x_{\text{exact}} - x^{(k)}\|_2/\|x_{\text{exact}}\|_2$, and the residual norm $\|b - Ax^{(k)}\|_2$.

We compare the following three algorithms.

- LSQR is the implementation from REGULARIZATION TOOLS.²¹
- HEB with a fixed regularization parameter λ chosen by the user.
- HEB with the regularization parameter λ_k chosen by applying generalized cross validation (GCV) to the projected problem in each iteration.

4.1 | A large component in augmented subspace

The first test problem is `deriv2(n, 2)` from REGULARIZATION TOOLS²¹ with $n = 32$ and relative noise level $\eta = 10^{-6}$. The augmentation subspace is

$$\mathcal{W}_2 = \text{span}\{w_1, w_2\}, \quad w_1 = (1, 1, \dots, 1)^T, \quad w_2 = (1, 2, \dots, n)^T. \quad (13)$$

For this problem, we have

$$\|W_2 W_2^T x_{\text{exact}}\|_2 / \|x_{\text{exact}}\|_2 = 0.99, \quad \|(I_n - W_2 W_2^T) x_{\text{exact}}\|_2 / \|x_{\text{exact}}\|_2 = 0.035,$$

and we only need to spend effort in capturing the small component of the solution in \mathcal{W}_2^\perp . However, as shown in the numerical results in Figure 2, the augmentation approach is crucial for computing a feasible reconstruction without the artificial oscillations toward the ends produced by LSQR.

If we know a good value of the regularization parameter λ , then HEB can produce a good reconstruction. Without such prior knowledge, however, HEB produces less accurate reconstructions as demonstrated with the choice $\lambda = 10^{-5}$, which is too large. On the other hand, if we use GCV to adaptively choose λ_k in each iteration, then this parameter settles at approximately 10^{-7} after six iterations, which is where our strategy will stop the iterations (we continue them, to illustrate that λ_k and the errors stay essentially unchanged).

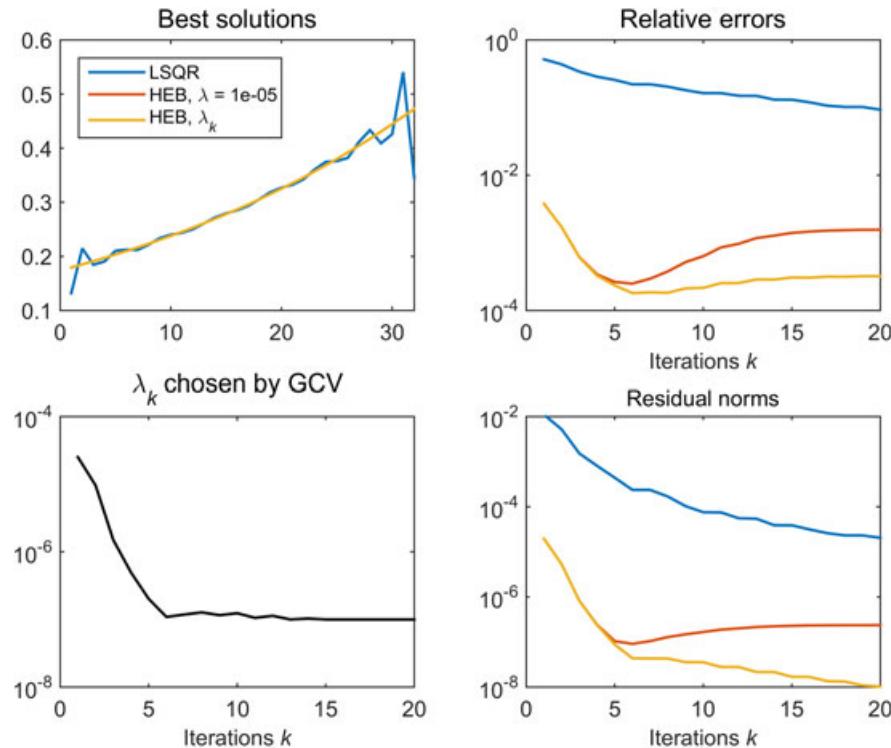


FIGURE 2 Results for the test problem `deriv2(n, 2)`. LSQR is not able to produce a good solution with the Krylov subspace \mathcal{K}_k , HEB performs much better due to the subspace augmentation, and generalized cross validation (GCV) successfully computes a very good regularization parameter λ_k that stagnates (as it should) after six iterations. The exact solution is indistinguishable from the HEB solution

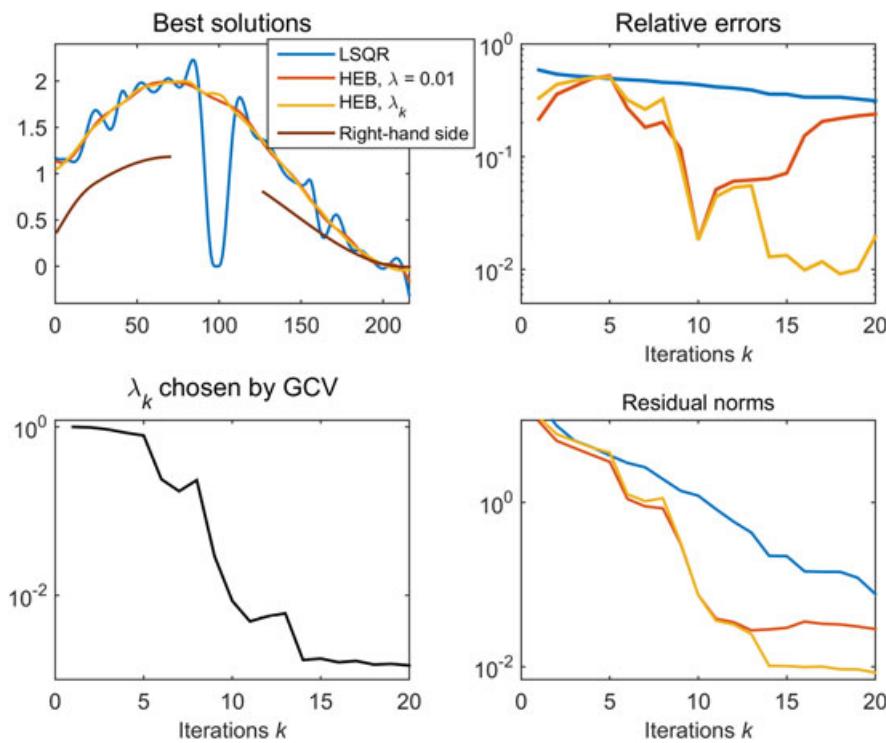


FIGURE 3 Numerical results for the test problem with combined deconvolution and inpainting. The top left plot shows the best reconstructions with the three methods, as well as the right-hand side with 56 “missing” values in the middle. The LSQR solution has many unwanted oscillations and a large error in the middle. The other two methods can produce good reconstructions, but only HEB with λ_k chosen by generalized cross validation (GCV) needs no prior knowledge about the noise level or the regularization parameter

4.2 | Combined deconvolution and inpainting

The second test problem is a 1D problem that combines deconvolution with inpainting. We first create an $n \times n$ Toeplitz matrix A_{full} (with $n = 216$), which is similar to the one from the test problem `phillips` from REGULARIZATION TOOLS, except that we change the point spread function to have half the width. Then, we remove rows 71–126 of this matrix to obtain (using MATLAB notation) $A = A_{\text{full}}([1 : 70, 127 : 216], :) \in \mathbb{R}^{160 \times 216}$. The exact solution x_{exact} has elements $\sin(1.5\pi i/n) + \cos(0.1\pi i/n)$, and the exact right-hand side is $b_{\text{exact}} = Ax_{\text{exact}}$ (think of b_{exact} as the blurred version of x_{exact} with the middle 56 elements removed; cf. Figure 3). We then add Gaussian noise with relative noise level $\eta = 10^{-4}$.

The numerical results are shown in Figure 3, using the augmentation subspace

$$\mathcal{W}_3 = \text{span}\{w_1, w_2, w_3\}, \quad w_1 = (1, 1, \dots, 1)^T, \quad w_2 = (1, 2, \dots, n)^T, \quad w_3 = (1, 4, \dots, n^2)^T. \quad (14)$$

The LSQR methods seek to approximate a minimum-norm solution, and therefore, even the best solution has a large error in the middle where the method seeks to introduce small elements. The best HEB solutions are much better because the augmentation subspace W_3 in (14) provides basis vectors that—together with the Tikhonov regularization—are able to “fill the gap” in the middle of the solution. The HEB method with a fixed λ depends on a good a priori choice; otherwise, many values have to be tried. The HEB method with λ_k chosen by GCV works very well, and no prior knowledge about the noise level or the regularization parameter is required.

4.3 | 2D example with combined image deblurring and inpainting

The final example illustrates that our new algorithm also works for larger problems. This test problem is a 2D version of the previous one, and it combines image deblurring and inpainting. The image is $N \times N$ with pixels given by $\sin(\pi i/(N-1)) \sin(\pi j/(N-1))$ for $i, j = 0, 1, 2, \dots, N-1$. The sparse matrix that represents the blurring is first generated by the function `blur` from REGULARIZATION TOOLS. To create missing data, we then remove those rows of the matrix that corresponds to a 16×16 region in the middle of the image; this produces a matrix A of dimensions $(N^2 - 256) \times N^2$. Finally we compute the noise-free data as $b = Ax$ and add Gaussian noise e such that $\|e\|_2/\|b\|_2 = 0.002$. We use $N = 80$.

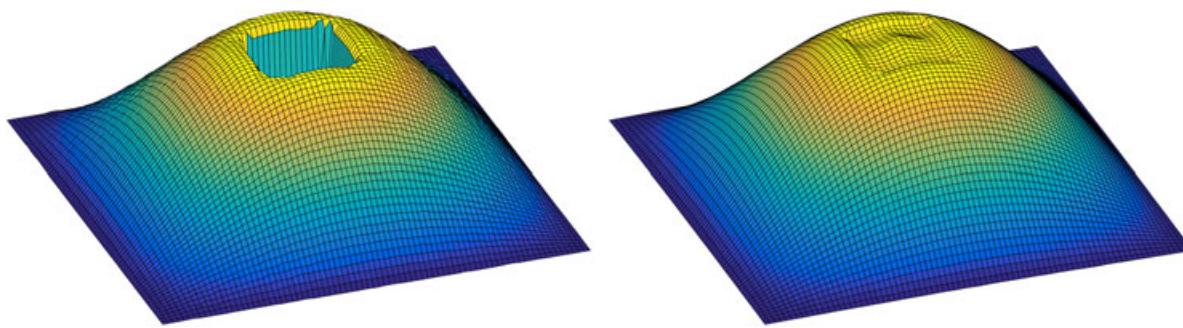


FIGURE 4 (Left) the best LSQR solution in which the central inpainted region has small pixel values, leaving a “hole” in the reconstruction. (Right) the best HEB solution with a fairly smooth inpainting of the central region. We note artifacts at the border of the inpaint region caused by the standard-form regularization term $\lambda_k^2 \|x\|_2^2$

In order to produce a smooth fill in the central part of the image with missing data, we use a four-dimensional augmentation subspace \mathcal{W}_4 whose basis vectors are vectorized versions of four simple arrays (using some MATLAB notation):

$$w_1 = \text{vec}(\text{ones}(N, N)), \quad w_2 = \text{vec}(\text{ones}(N, 1) * (1 : N)), \quad w_3 = \text{vec}((1 : N)' * \text{ones}(1 : N)), \quad w_4 = \text{vec}((1 : N)' * (1 : N)).$$

We use GCV to determine the regularization parameter in each iteration.

Figure 4 compares the best LSQR solution with the best HEB solution described above. We see that the LSQR solution, which uses an standard (unaugmented) Krylov subspace, is not able to produce smooth inpainting; it leads to a smooth reconstruction but with a central region with small pixel values. The HEB solution, on the other hand, inpaints the missing pixels in a smooth fashion as dictated by our augmentation subspace. We notice artifacts at the border of the inpaint region; they are caused by the standard-form regularization term $\lambda_k^2 \|x\|_2^2$ in (11), which does not ensure smoothness (that will require the implementation of a Laplacian smoothing term, which is outside the scope of this paper).

5 | CONCLUSION

We demonstrated how to augment the bidiagonalization algorithm underlying LSQR such that it computes solutions in an enriched subspace consisting of the standard Krylov subspace plus a low-dimensional linear subspace that represents additional prior information about the solution. We also demonstrated how to add standard-form Tikhonov regularization, thus arriving at a hybrid enriched bidiagonalization algorithm that can choose the regularization parameter adaptively in each iteration. We discussed implementation issues and gave a few illustrative numerical examples.

ACKNOWLEDGEMENTS

This work was supported by the European Research Council under Grant 291405 (HD-Tomo) and by the National Natural Science Foundation of China under Grant 11701388.

ORCID

Per Christian Hansen <https://orcid.org/0000-0002-7333-7216>

Yiqiu Dong <https://orcid.org/0000-0001-8363-9448>

Kuniyoshi Abe <https://orcid.org/0000-0001-5910-0127>

REFERENCES

1. Hansen PC. Discrete inverse problems: Insight and algorithms. Philadelphia, PA: SIAM; 2010.
2. van der Sluis A, van der Vorst HA. SIRT- and CG-type methods for the iterative solution of sparse linear least-squares problems. Linear Algebra Appl. 1990;130:257–303.
3. Calvetti D, Lewis B, Reichel L. On the regularizing properties of the GMRES method. Numer Math. 2002;91:605–625.
4. Kilmer ME, Stewart GW. Iterative regularization and MINRES. SIAM J Matrix Anal Appl. 1999;21:613–628.
5. Calvetti D, Lewis B, Reichel L. GMRES-type methods for inconsistent systems. Linear Algebra Appl. 2000;316:157–169.

6. Neuman A, Reichel L, Sadok H. Implementations of range restricted iterative methods for linear discrete ill-posed problems. *Linear Algebra Appl.* 2012;436:3974–3990.
7. Hanke M. Conjugate gradient type methods for ill-posed problems. Harlow, UK: Longman; 1995. Pitman research notes in mathematics, No. 327.
8. Dykes L, Marcellan F, Reichel L. The structure of iterative methods for symmetric linear discrete ill-posed problems. *BIT Numer Math.* 2014;54:129–145.
9. Jensen TK, Hansen PC. Iterative regularization with minimum-residual methods. *BIT Numer Math.* 2007;47:103–120.
10. Gazzola S, Novati P, Russo MR. On Krylov projection methods and Tikhonov regularization. *Electron Trans Numer Anal.* 2015;44:83–123.
11. Baglama J, Reichel L. Augmented GMRES-type methods. *Numer Linear Algebra Appl.* 2007;14:337–350.
12. Calvetti D, Reichel L, Shuibi A. Enriched Krylov subspace methods for ill-posed problems. *Linear Algebra Appl.* 2003;362:257–273.
13. Kuroiwa N, Nodera T. The adaptive augmented GMRES method for solving ill-posed problems. *ANZIAM J.* 2008;50:654–667.
14. Dong Y, Garde H, Hansen PC. R³GMRES: including prior information in GMRES-type methods for discrete inverse problems. *Electron Trans Numer Anal.* 2014;42:136–146.
15. Paige CC, Saunders MA. LSQR: an algorithm for sparse linear equations and sparse least squares. *ACM Trans Math Softw.* 1982;8:43–71.
16. Chung J, Nagy JG, O'Leary DP. A weighted-GCV method for Lanczos-hybrid regularization. *Electron Trans Numer Anal.* 2008;28:149–167.
17. Hansen PC, Jørgensen JS. AIR Tools II: algebraic iterative reconstruction methods, improved implementation. *Numer Algo.* 2018;79:107–137. <https://doi.org/10.1007/s11075-017-0430-x>
18. Gazzola S, Hansen PC, Nagy JG. IR tools: a MATLAB package of iterative regularization methods and large-scale test problems. *Numer Algo.* 2018; <https://doi.org/10.1007/s11075-018-0570-7>
19. Golub GH, Van Loan CF. Matrix computations. 4th ed. Baltimore, MD: Johns Hopkins University Press; 2013.
20. Pan C-T. A perturbation analysis of the problem of downdating a Cholesky factorization. *Linear Algebra Appl.* 1993;183:103–115.
21. Hansen PC. Regularization tools version 4.0 for Matlab 7.3. *Numer Algo.* 2007;46:189–194.

How to cite this article: Hansen PC, Dong Y, Abe K. Hybrid enriched bidiagonalization for discrete ill-posed problems. *Numer Linear Algebra Appl.* 2019;e2230. <https://doi.org/10.1002/nla.2230>