LBAS: Lanczos Bidiagonalization with Subspace Augmentation for Discrete Inverse Problems

Per Christian Hansen[1] and Kuniyoshi Abe[2]

[1]DTU Compute, Technical University of Denmark
[2]Gifu Shotoku Gakuen University, Japan

Technical Report-2017-03

Abstract

The regularizing properties of Lanczos bidiagonalization are powerful when the underlying Krylov subspace captures the dominating components of the solution. In some applications the regularized solution can be further improved 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 Lanczos bidiagonalization algorithm.

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

$$\min_x \|A\,x - b\|_2^2, \qquad A \in \mathbb{R}^{m\times n}. \tag{1}$$

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,

$$\min_x \left\{ \|A\,x - b\|_2^2 + \lambda\,\mathcal{R}(x) \right\}.$$

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.

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, cf. [9]. This leads to a projection formulation of the form

$$\min_x \|A\,x - b\|_2^2 \qquad \text{s.t.} \qquad x \in \mathcal{S}_k\ , \tag{2}$$

where the *signal subspace* $\mathcal{S}_k$ is a linear subspace of dimension k. If the basis of $\mathcal{S}_k = \text{span}\{v_1, v_2, \ldots, v_k\}$ 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:

$$\begin{aligned}
\mathcal{K}_k =& \text{span}\{A^T b, A^T A\, A^T b, (A^T A)^2 A^T b, \ldots\} \\
& \text{for the CGLS and LSQR algorithms [9], [13],} \\
\bar{\mathcal{K}}_k =& \text{span}\{b, A\, b, A^2\, b, \ldots\} \\
& \text{for the GMRES and MINRES algorithms [3], [11],} \\
\vec{\mathcal{K}}_k =& \text{span}\{A\, b, A^2 b, A^3 b, \ldots\} \\
& \text{for the RRGMRES and MR-II algorithms [2], [7],}
\end{aligned}$$

where k is the number of iterations. Depending on the application, one or more of these subspaces may be well suited to compute a good regularized solution, i.e., a good approximation that is only little sensitive to perturbations of the data, cf. [10]. Moreover, it is possible to combine the projection formulation with Tikhonov regularization; this leads to so-called hybrid methods [9].

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 in [1] and [4] 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 augmented Krylov subspace methods. See also [12].

Recently we presented an algorithm R^{3}GMRES [6] based on the range-restricted GMRES (RRGMRES) method [2] 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 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 $\mathcal{K}_k$,

$$\mathcal{S}_{p,k} = \mathcal{W}_p + \mathcal{K}_k \equiv \{y + z \,|\, y \in \mathcal{W}_p \wedge z \in \mathcal{K}_k\}, \tag{3}$$

which itself is a linear subspace.

An efficient and stable algorithm ENRICHED CGNR for this problem, based on the CGLS algorithm, was already published in [4]. In this work we present an alternative algorithm, called LBAS (Lanczas Bidiagonalization with Augmented Subspace), 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 algorithms, and our explicit use of reorthogonalization makes is numerically stable.

2 Formulation of the Algorithm

We want to solve the problem

$$\min_x \|A\,x - b\|_2^2 \qquad \text{s.t.} \qquad x \in \mathcal{W}_p + \mathcal{K}_k \ . \tag{4}$$

In principle we could use, say, a Hessenberg decomposition

$$A\,[\,W_p\,,\ A^Tb\,, A^TA\,A^Tb\,, \cdots\,,\ (A^TA)^{k-1}A^Tb\,] = V_{p+k+1}\,H_{p+k}$$

and compute the solution as

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

But we prefer to use a stable and efficient "standard" algorithm. Hence we use the Lanczos bidiagonalization algorithm to compute an orthonormal basis of $\mathcal{K}_k$, and augment it by $\mathcal{W}_p$ in each step of the algorithm. This may seem cumbersome – but the overhead is, in fact, favorably small.

At step k we have the decomposition

$$A\,[\,V_k\,,\ W_p\,] = \left[\,U_{k+1}\,,\ \widetilde{U}_k\,\right] \begin{bmatrix} B_k & G_k \\ 0 & F_k \end{bmatrix} \tag{5}$$

where the blue quantities are associated with the classical bidiagonalization algorithm, while the red and pink quantities are associated with the augmentation. Specifically:

- $A\,V_k = U_{k+1}B_k$ is obtained after k steps of the bidiag. 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, $u_1 = b/\|b\|_2$.
- $\widetilde{U}_k \in \mathbb{R}^{m\times p}$: $\text{range}(AW_p) = \text{range}(U_{k+1}G_k + \widetilde{U}_kF_k)$ and $\widetilde{U}_k^TU_{k+1} = 0$.
- $B_k \in \mathbb{R}^{(k+1)\times k}$ is a lower bidiagonal matrix.
- $F_k \in \mathbb{R}^{p\times p}$ and *changes in every iteration.*
- G_k is $(k+1)\times p$ and is *updated* along with B_k.

The columns of $[\,V_k\,,\ W_p\,]$ form a basis for $\mathcal{S}_{p,k}$. Now recall that

$$A\,[\,V_k\,,\ W_p\,] = \left[\,U_{k+1}\,,\ \widetilde{U}_k\,\right] \begin{bmatrix} B_k & G_k \\ 0 & F_k \end{bmatrix}\ . \tag{6}$$

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 $\text{range}(U_{k+1})$ and $\text{range}(\widetilde{U}_k)$, respectively:

$$G_k = U_{k+1}^TAW_p, \qquad F_k = \widetilde{U}_k^TAW_p\ . \tag{7}$$

Then the iterate $x^{(k)} \in \mathcal{S}_{p,k}$ is given by $x^{(k)} = [\, V_k \,,\, W_p \,]\, y^{(k)}$, where

$$y^{(k)} = \operatorname{argmin}_y \left\| \begin{bmatrix} B_k & G_k \\ 0 & F_k \end{bmatrix} y - \begin{bmatrix} U_{k+1}^T \\ \widetilde{U}_k^T \end{bmatrix} b \right\|_2^2 . \tag{8}$$

The above derivation leads to the following generic formulation:

ALGORITHMS LBAS

1. Set $U_1 = b/\|b\|_2$, $V_0 = [\,]$, $B_0 = [\,]$, $G_0 = U_1^T A W_p$, and $k = 1$.
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 AW_p with respect to U_{k+1} to obtain $\widetilde{U}_k \in \mathbb{R}^{m\times p}$.
5. Compute $F_k = \widetilde{U}_k^T A W_p \in \mathbb{R}^{p\times p}$.
6. Solve $\min_y \left\| \begin{bmatrix} B_k & G_k \\ 0 & F_k \end{bmatrix} y - \begin{bmatrix} U_{k+1}^T \\ \widetilde{U}_k^T \end{bmatrix} b \right\|_2^2$ 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 $\widetilde{U}_k$ and the small $p \times p$ matrix F_k in each step, but the dimension p of the augmentation subspace is small so this overhead is negligible.

In each step we update the orthogonal factorization:

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

$T_k^{(11)} \in \mathbb{R}^{k\times k}$ and $T_k^{(22)} \in \mathbb{R}^{p\times p}$ are upper triangular, Q 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 $\widetilde{U}_k$ is already orthogonal to U_k, hence (in principle) we can perform the update $\widetilde{U}_{k+1} = (I_m - u_{k+1} u_{k+1}^T)\, \widetilde{U}_k$, where I_m is the identity matrix of order m. For numerical stability, we must reorthogonalize the columns of V_k, U_{k+1}, and $\widetilde{U}_k$. This is an acceptable approached used in many similar algorithms, such as the algorithm HyBR [5]. The MATLAB code for LBAS is listed below.

```
function [X,rho,eta] = lbas(A,b,W,k,reorth)
%LBAS Lanczos bidiagonalization solver with augmented subspace
%
% [X,rho,eta] = lbas(A,b,W,k,reorth)
%
% A - coeffient matrix
% b - right-hand side
% W - a matrix with orthonormal columns (or a positive integer)
% k - the number of iterations
% reorth - 0: no reorthogonalization, 1: MGS ditto (default)
%
% X - matrix with k columns, each column holds an iteration vector
% rho - residual norms
% eta - solution norms
%
% If W is a positive integer p, then W has p columns corresponding
%  to the polynomials of degree 0,1,...,p-1.

% Per Christian Hansen, DTU Compute and Kuniyoshi Abe, Gifu Shotoku
% Gakuen University, Sept. 4, 2015.

if nargin < 4, error('Too few input arguments'), end
if nargin < 5, reorth = 1; end
[m,n] = size(A);
[nW,p] = size(W);
if nW == 1 && p == 1 && W > 0 && ~rem(W,1); % Set W.
    p = W;
    W = ones(n,p);
    for i = 1:p-1
        W(:,i+1) = (1:n).^i;
    end
    W = orth(W);
elseif nW ~= size(A,2);
    error('No. rows in W must equal no. columns in A');
end

% Initialize.
X = zeros(n,k);
U = zeros(m,k+1);
V = zeros(n,k);
Bk  = zeros(k+1,k);
Gk = zeros(k+1,p);
g = zeros(k+p+1,1);

% Prepare for iterations.
beta = norm(b);
u = b/beta;
U(:,1) = u;
v = zeros(n,1);
AW = A*W;
Gk(1,:) = u'*AW;
normb = beta;
g(1) = normb;
```

```
% Commence iterations.
for j = 1:k

    % Next (rightmost) column in lower bidiagonal part of matrix
    % via Lanczs bidiagonalization process.
    r = A'*u - beta*v;
    if reorth==1
        for i=1:j-1, r = r - (V(:,i)'*r)*V(:,i); end
    end
    alpha = norm(r); v = r/alpha;
    V(:,j) = v;
    Bk(j,j) = alpha;

    pp = A*v - alpha*u;
    if reorth==1
        for i=1:j, pp = pp - (U(:,i)'*pp)*U(:,i); end
    end
    beta = norm(pp); u = pp/beta;
    U(:,j+1) = u;
    Bk(j+1,j) = beta;

    % Apply stored orthog. transf. to new column of Bk.
    if j>1
        Bk(j-1:j,j) = [-si;conj(co)]*Bk(j,j);
    end

    % Determine new orthog. transf. to make Bk upper triangular.
    nu = norm(Bk(j:j+1,j));
    if nu==0 , error('Breakdown'), end
    co = Bk(j,j)/nu;
    si = -Bk(j+1,j)/nu;
    Bk(j,j) = co*Bk(j,j) - si*Bk(j+1,j);
    Bk(j+1,j) = 0;

    % Apply the orthog. transf. to updated G and to rhs.
    if j>1
        Gk(j,:) = saveG;
        g(j) = saveg;
        g(j+1:j+p) = 0;
    end
    Gk(j+1,:) = u'*AW;
    Gk(j:j+1,:) = [co,-si;si,conj(co)]*Gk(j:j+1,:);
    saveG = Gk(j+1,:);  % To be used in next iteration.
    g(j:j+1) = [co,-si;si,conj(co)]*g(j:j+1);
    saveg = g(j+1);     % Ditto.

    % Needed for bottom right block Fk.
    if j==1
        Utilde = ort(U(:,1:j+1),AW);
    else
        %Utilde = ort(U(:,j+1),Utilde);
        Utilde = ort(U(:,1:j+1),Utilde);
    end
```

```
    % QR factorization of bottom right block.
    [qq,rr] = qr([Gk(j+1,:);Utilde'*AW]);
    Gk(j+1,:) = rr(1,:);
    Fk = rr(2:end,:);
    g(j+1:j+1+p) = qq'*g(j+1:j+1+p);

    % Compute solution.
    y = [Bk(1:j+1,1:j) Gk(1:j+1,:); zeros(p,j) Fk] \ g(1:j+p+1);
    X(:,j) = [V(:,1:j),W]*y;
    % svd([Bk(1:j+1,1:j) Gk(1:j+1,:); zeros(p,j) Fk])'
    disp(y')

end

if nargout > 1
    rho = sqrt(sum(abs(A*X-repmat(b,1,k)).^2));
end
if nargout > 2
    eta = sqrt(sum(abs(X).^2));
end

% Subfunction ====================================================

function Vw = ort(V,W)
% Orthonormalize W with respect to V (remove components along V).
% Henrik Garden & Per Chr. Hansen, DTU Compute, July 30, 2013.

k = size(V,2)-1;
p = size(W,2);
for s = 1:p
    w = W(:,s);
    for i = 1:k+s
        vi = V(:,i);
        w = w-vi'*w*vi;
    end
    w = w/norm(w);
    V(:,k+s+1) = w;
end
Vw = V(:,end-p+1:end);
```

3 Numerical Examples

To illustrate the performance of our LBAS 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. We show the best solution within the iterations plus:

- the relative error $\|x_{\text{exact}} - x^{(k)}\|_2 / \|x_{\text{exact}}\|_2$,
- the relative residual norm $\|b - A\,x^{(k)}\|_2 / \|b\|_2$.

We compare combinations of the following algorithms:

- CGLS is the implementation from Regularization Tools [8].
- RRGMRES is the implementation from Regularization Tools [8].
- R^{3}GMRES is our implementation of the algorithm from [6].
- LBAS is our new algorithm.

3.1 A Large Component in Augmented Subspace

The test problem is `deriv2(n,2)` from Regularization Tools [8], with $n = 32$ and relative noise level $\eta = 10^{-5}$. The augmentation subspace is

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

For this problem we have

$$\|W_2 W_2^T x_{\text{exact}}\|_2 / \|x_{\text{exact}}\|_2 = 0.99 \ ,$$

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

and we only need to spend effort in capturing the small component of the solution in $\mathcal{W}_2^{\perp}$. This is reflected in the numerical results in Fig. 1 that demonstrate the feasibility of the methods using the augmented subspace; both methods give the same accuracy so nothing is lost in going from R^{3}GMRES to LBAS.

3.2 Capture a Discontinuity

For this test problem we use `gravity(n)` from Regularization Tools with $n = 100$, $\eta = 10^{-3}$, but the exact solution changed to include a discontinuity between elements $\ell = 50$ and $\ell + 1 = 51$. The augmentation matrix W_2 allows us to represent this discontinuity:

$$w_1 = \begin{bmatrix} \text{ones}(\ell, 1) \\ \text{zeros}(n-\ell, 1) \end{bmatrix}, \quad w_2 = \begin{bmatrix} \text{zeros}(\ell, 1) \\ \text{ones}(n-\ell, 1) \end{bmatrix}.$$

The results are shown in Fig. 2. As before, LBAS produces a solution of the same quality as R^{3}GMRES – while the CGLS and RRGMRES solutions are inferior.

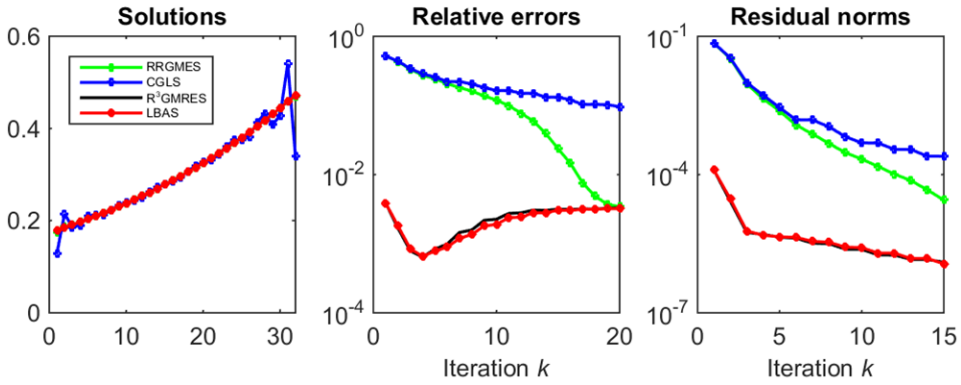

Figure 1: Results for the test problem with a large solution component in the augmentation subspace $\mathcal{W}_2$. CGLS is not able to produce a good solution due to the Krylov subspace $\mathcal{K}_k$; RRGMRES performs better but requires many iterations. R^{3}GMRES as well as our new LBAS perform equally well and much better than the former two methods.

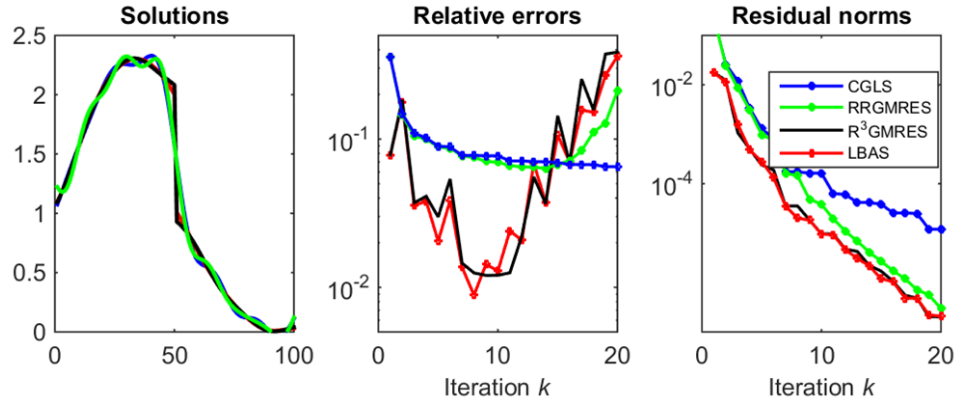

Figure 2: Numerical results for the test problem with a discontinuity in the solution. The CGLS and RRGMRES algorithms are not able to reproduce the discontinuity very well. Both R^{3}GMRES and LBAS give good results of the equal quality.

3.3 Fix Boundary Conditions

This test problem is based on the discretization of a first-kind Fredholm integral equation:

$$\int_0^\pi t\, \exp(-s\, t^2)\, f(t)\, dt = g(s), \quad 0 \le s \le \pi,$$

and we use

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

The problem is discretized in such a way that the matrix A corresponds to zero boundary conditions. The matrix $\mathcal{W}_2$ is chosen such that it compensates for the incorrect boundary conditions implicit in the matrix A, by allowing the regularized solutions to have nonzero values and nonzero derivatives at the endpoints. We consider both a square matrix A obtained with $m = n = 32$ and a rectangular matrix obtained with $m = 64$ and $n = 32$. The results are shown in Fig. 3; here LBAS outperforms the other methods.

3.4 Compute the Spectrum of X-Ray Source

The spectrum of an X-ray source (where accelerated electrons hit an anode) consists of a continuous spectrum superimposed with line spectra. We know the frequencies of the line spectral, so we can easily incorporate this information through the augmentation subspace and thus estimate the combined spectrum from measured data.[1] We experiment with two choices:

- $W_{\text{delta}} \rightarrow$ two delta functions at the right frequencies,
- $W_{\text{Gauss}} \rightarrow$ two narrow Gauss functions at the right frequencies.

In Fig. 4 we see that LBAS is very capable of computing a good approximation to the exact spectrum, especially with the two narrow Gauss functions in the augmented subspace.

4 Conclusion

We considered how to implement an algorithm LBAS, based on Lanczos bidiagonalization, that augments the CGLS Krylov subspace with a user-defined subspace of low dimension that captures desired features of the solution. We formulate the algorithm and demonstrate how to implement it efficiently. Numerical examples demonstrate the feasibility of our algorithm and its advantage over related algorithms.

[1] We than Prof. Jan Sijbers from University of Antwerp for inspiration to this example.

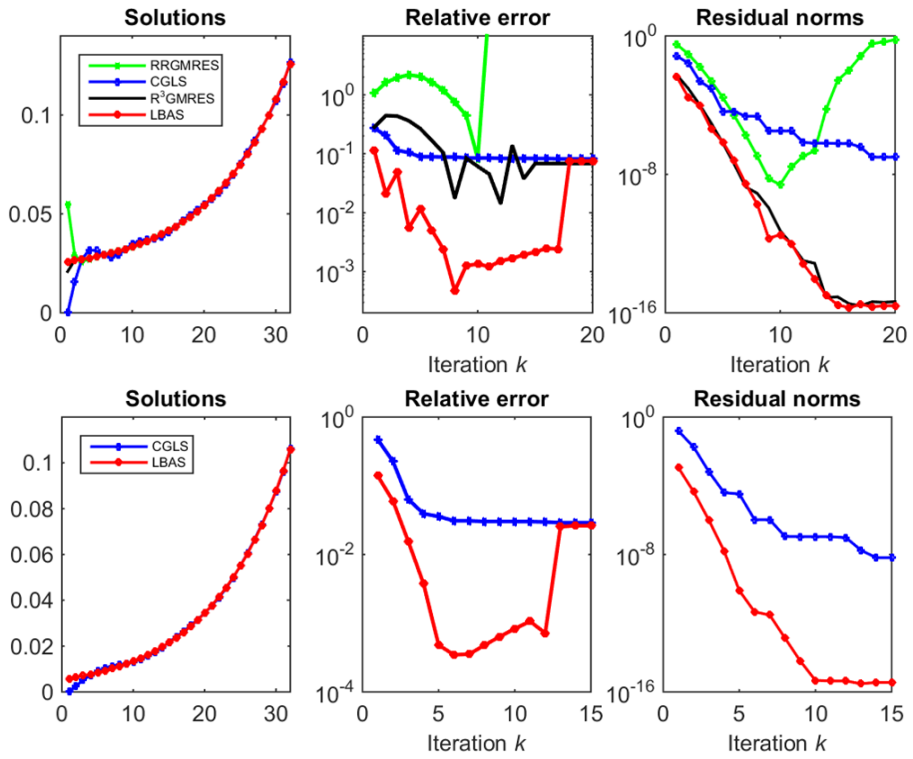

Figure 3: Test problem where the augmentation subspace compensates for incorrect boundary conditions in the matrix A. Top: square matrix with $m = n = 32$. Bottom: rectangular matrix with $m = 64$ and $n = 32$. In both cases LBAS gives very good solutions.

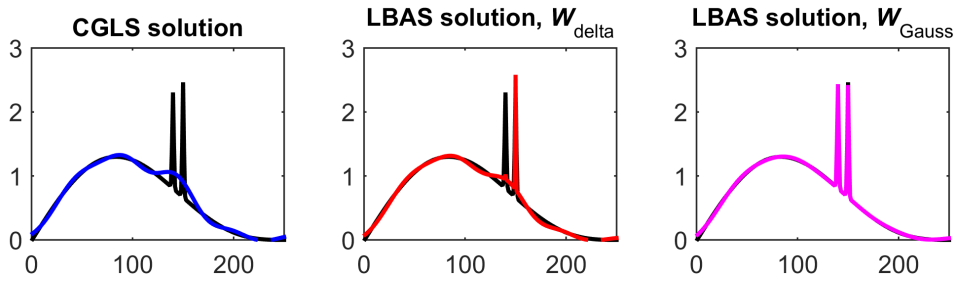

Figure 4: Reconstruction of a X-ray source's spectrum consisting of a continuous spectrum superimposed with line spectra. By representing the line spectra explicitly in the augmented subspace we are able to compute a good reconstruction.

References

[1] J. Baglama and L. Reichel, *Augmented GMRES-type methods*, Num. Lin. Alg. Appl., 14 (2007), pp. 337–350.

[2] D. Calvetti, B. Lewis, and L. Reichel, *GMRES-type methods for inconsistens systems*, Lin. Alg. Appl, 316 (2000), pp. 157–169.

[3] D. Calvetti, B. Lewis, and L. Reichel, *On the regularizing properties of the GMRES method*, Numer. Math, 91 (2002), pp. 605–625.

[4] D. Calvetti, L. Reichel, and A. Shuibi, *Enriched Krylov subspace methods for ill-posed problems*, Lin. Alg. Appl., 362 (2003), pp. 257–273.

[5] J. Chung, J. G. Nagy, and D. P. O'Leary, *A weighted-GCV method for Lanczos-hybrid regularization*, Electronic Transactions on Numerical Analysis, 28 (2008), pp. 149–167.

[6] Y. Dong, H. Garde, and P. C. Hansen, *R^3GMRES: including prior information in GMRES-type methods for discrete inverse problems*, Electronic Trans. Numerical Analysis, 42 (2014), pp. 136–146

[7] M. Hanke, *Conjugate Gradient Type Methods for Ill-Posed Problems*, Pitman Research Notes in Mathematics 327, Longman, Harlow, UK, 1995.

[8] P. C. Hansen, *Regularization Tools Version 4.0 for Matlab 7.3*, Numer. Algo., 46 (2007), pp. 189–194.

[9] P. C. Hansen, *Discrete Inverse Problems: Insight and Algorithms*, SIAM, Philadelphia, 2010.

[10] T. K. Jensen and P. C. Hansen, *Iterative regularization with minimum-residual methods*, BIT, 47 (2007), pp. 103–120.

[11] M. E. Kilmer and G. W. Stewart, *Iterative Regularization and MINRES*, SIAM J. Matrix Anal. Appl., 21 (1999), pp. 613–628.

[12] N. Kuroiwa and T. Nodera, *The adaptive augmented GMRES method for solving ill-posed problems*; in G. N. Mercer and A. J. Roberts (Eds.), *Proceedings of the 14th Biennial Computational Techniques and Applications Conference, CTAC-2008*, ANZIAM J., 50 (2008), pp. C654–C667.

[13] A. van der Sluis and H. A. van der Vorst, *SIRT- and CG-type methods for the iterative solution of sparse linear least-squares problems*, Lin. Alg. Appl., 130 (1990), pp. 257–302.