Thirty-Sixth Conference of the Dutch-Flemish Numerical Analysis Communities. 5–7 October 2011, Woudschouten, Zeist, NL.

Image Deblurring with Krylov Subspace Methods

Per Christian Hansen*
Technical University of Denmark
pch@imm.dtu.dk

Abstract

Image deblurring, i.e., reconstruction of a sharper image from a blurred and noisy one, involves the solution of a large and very ill-conditioned system of linear equations, and regularization is needed in order to compute a stable solution. Krylov subspace methods are often ideally suited for this task: their iterative nature is a natural way to handle such large-scale problems, and the underlying Krylov subspace provides a convenient mechanism to regularized the problem by projecting it onto a low-dimensional "signal subspace" adapted to the particular problem. In this talk we consider the three Krylov subspace methods CGLS, MINRES, and GMRES. We describe their regularizing properties, and we discuss some computational aspects such as preconditioning and stopping criteria.

Keywords: Image deblurring, regularizing iterations, Krylov subspaces, CGLS, MINRES, GMRES, smoothing norm, weighted pseudoinverse, stopping criteria.

AMS Subject Classification: 65F22, 65F10

Image deblurring is a discrete ill-posed problem $A\,x = b$ where A represents the blurring, x^{exact} represents the exact image, and $b = A\,x^{\mathrm{exact}} + e$ represents the blurred and noisy image image. For details about this problem see, e.g., [2] and [9].

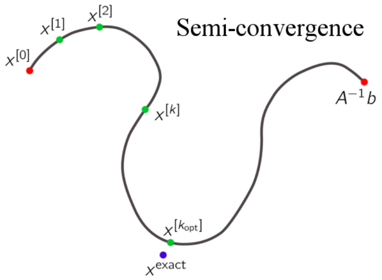


Fig. 1.

A characteristic of Krylov subspace methods applied to ill-posed problems is that they exhibit *semi-convergence*, i.e., the iterates $x^{(k)}$ tend to be better and better approximations to the exact solution x^{exact}, but at some stage they start to diverge again end instead converge towards the undesired solution $A^{-1}b$. If we can stop the iterations at $k = k_{\mathrm{opt}}$ then, in principle, we have a large-scale regularization method — see [3], [7, Ch. 5] and [13] for details.

*This work is part of the project CSI: Computational Science in Imaging, supported by grant 274-07-0065 from the Danish Research Council for Technology and Production Sciences.

Projection Methods

To motivate the Krylov subspace methods, we first briefly discuss projection methods. Often, the best way to solve an ill-posed problem is to compute an approximation to x^{exact} that effectively lies in a (low-dimensional) *signal subspace* of $\mathbb{R}^n$. For Truncated SVD, this subspace is spanned by the first k right singular vectors $v_1, \ldots, v_k$.

For large-scale problems it is infeasible, or impossible, to compute the SVD of A. But we might be able to provide, a priori, a set of basis vectors $w_1, \ldots, w_k$ that have the same overall features as the singular vectors, namely, being dominated by low-frequency components. For example, we could choose the orthonormal basis vectors of the discrete cosine transform (DCT) as our basis vectors (all computations that involve these vectors are performed by means of the FFT algorithm).

Given the matrix $W_k = (w_1, \ldots, w_k) \in \mathbb{R}^{n \times k}$, computation of the least squares solution expressed in this basis amounts to the constrained least squares problem

$$\min_x \|A\,x - b\|_2 \qquad \text{s.t.} \qquad x \in \mathcal{W}_k = \operatorname{span}\{w_1, \ldots, w_k\}. \tag{1}$$

We can reformulate the constraint in (1) as the requirement that $x = W_k\, y$, which leads to a regularized solution expressed in the more computation-friendly formulation

$$x^{(k)} = W_k\, y^{(k)}, \qquad y^{(k)} = \operatorname{argmin}_y \|(A\, W_k)\, y - b\|_2. \tag{2}$$

If k is not too large, then we can explicitly compute the matrix $A\, W_k \in \mathbb{R}^{n \times k}$ and then solve the projected least squares problem for y.

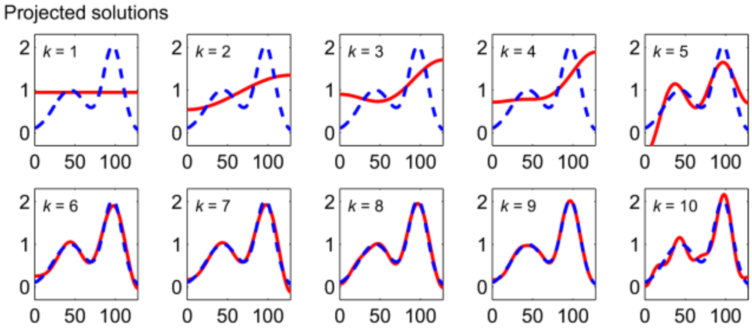


Fig. 2.

Figure 2 illustrates the use of the DCT projection method applied to the `shaw` test problem from Regularization Tools [8]. We show the solutions computed by projecting the original problem onto the basis vectors $w_1, \ldots, w_k$ for $k = 1, 2, \ldots, 10$; the projection onto a low-dimensional subspace indeed has a regularizing effect on the solution, and $k_{\mathrm{opt}} = 9$ seems to be the optimal dimension of the projection subspace.

Regularizing Krylov-Subspace Iterations

The advantage of the projection approach is that the operations involving the basis vectors can often be performed fast and without the need to explicitly compute and store the basis. At the same time, the fixed basis is also the main disadvantage of this approach; the basis vectors are not adapted to the particular problem. To our rescue comes the *Krylov subspace* associated with A and b, defined as:

$$\mathcal{K}_k \equiv \operatorname{span}\{A^T b, (A^T A)\, A^T b, (A^T A)^2 A^T b, \ldots, (A^T A)^{k-1} A^T b\}. \tag{3}$$

The dimension of this subspace is at most k. The vectors $(A^T A)^j A^T b$ become increasingly richer in the direction of the principal eigenvector of $A^T A$, making them unsuited for numerical computations; but the subspace $\mathcal{K}_k$ itself carries important information about the problem, and it is well suited to play the role of the subspace $\mathcal{W}_k$ in the projection method (1). We just need a better representation of its basis.

In principle we could apply the MGS algorithm to the vectors $(A^T A)^j A^T b$, but the well-known Lanczos bidiagonalization process, applied to the matrix A with starting vector b, produces precisely these vectors! Specifically, after k iterations the bidiagonalization algorithm has produced the matrices $W_k \in \mathbb{R}^{n\times k}$ and $U_{k+1} \in \mathbb{R}^{m\times(k+1)}$ and a lower bidiagonal matrix $B_k \in \mathbb{R}^{(k+1)\times k}$ such that

$$U_{k+1}^T A\, W_k = B_k \qquad \text{for} \qquad k = 1, 2, \ldots\ . \tag{4}$$

Strictly speaking, this is only true if the computations are done in infinite precision; but we refrain from discussing the finite-precision aspects here. The largest singular values of B_k converge to the larger singular values of A, and therefore the Lanczos algorithm is typically used for SVD computations for large sparse or structured matrices. Here we use it to produce the orthonormal basis vectors for the Krylov subspace $\mathcal{K}_k$.

The Lanczos process is closely related to the classical method of conjugate gradients (CG) for solving a system of linear equations with a symmetric positive definite coefficient matrix. In our case, this system is the so-called normal equations $A^T A\, x = A^T b$ associated with the un-regularized least squares problem $\min_x \|A\, x - b\|_2$, and one can show that the solution $x^{(k)}$ obtained after applying k steps of the CG algorithm to the normal equations $A^T A\, x = A^T b$, with the zero starting vector, is precisely the solution to the projected problem (1) with $\mathcal{W}_k = \mathcal{K}_k$ (in infinite precision). The key to the efficiency is that the CG algorithm computes this solution without the need to orthonormalize and store all the basis vectors $w_1, w_2, \ldots$ explicitly; only a few auxiliary vectors are needed. In the face of rounding errors, the convergence slows down but without deteriorating the accuracy of the solutions.

The most stable way to implement the CG algorithm for the normal equations is known as the *CGLS algorithm*. This algorithm requires one multiplication with A and one multiplication with A^T per iteration. Except for a normalization and perhaps a sign, the "search vectors" of the CGLS algorithm are equal to the orthonormal basis vectors for the Krylov subspace. The solution norm $\|x^{(k)}\|_2$ and residual norm $\|A\, x^{(k)} - b\|_2$ are monotonic functions of k,

$$\|x^{(k)}\|_2 \geq \|x^{(k-1)}\|_2, \qquad \|A\, x^{(k)} - b\|_2 \leq \|A\, x^{(k-1)} - b\|_2, \qquad k = 1, 2, \ldots$$

allowing us to use the L-curve method as a stopping criterion for this method.

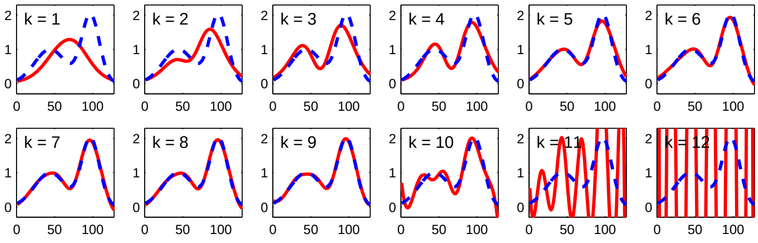


Fig. 3.

To illustrate the regularizing effects of the CGLS algorithm, we applied this algorithm to the same test problem as above. Figure 3 shows the CGLS iterates $x^{(k)}$ for $k = 1, 2, \ldots, 12$. We see that the Krylov subspace $\mathcal{K}_k$ provides a good basis for this problem, and the best approximation to x^{exact} is achieved after $k_{\text{opt}} = 9$ iterations.

CGLS Focuses on the Significant Components

The Truncated SVD method, by definition, includes all the first k SVD components in the regularized solution x_k; all these SVD components are included whether they are needed or not. On the other hand, in the projection method we are free to only includes those basis vectors that are needed.

This important feature carries over to the CGLS method, due to the fact that its basis vectors are those of the Krylov subspace, and not the singular vectors. Recall that the Krylov subspace $\mathcal{K}_k$ (3) is defined in terms of both the matrix A and the right-hand side b (while the SVD depends on A only). The Krylov subspace $\mathcal{K}_k$ will adapt itself in an optimal way to the specific right-hand side – while the SVD basis is optimal if no information about b is given.

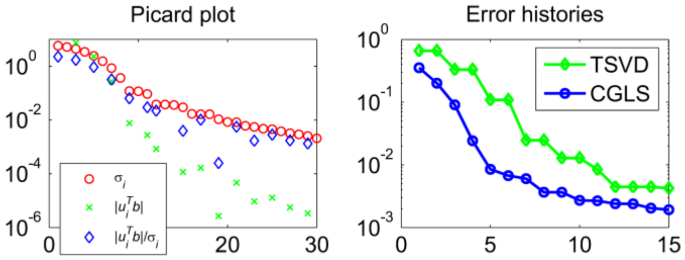


Fig. 4.

To illustrate this feature of the Krylov subspace we applied the CGLS algorithm to the `phillips` test problem from Regularization Tools. Due to the symmetries in both the matrix and the exact solution, many of the SVD coefficients of the exact solution are zero. The Picard plot in the left part of Fig. 4 reveals that about every second SVD component is zero. For this reason, the TSVD method with truncation parameter k includes about twice the amount of components in x_k that are necessary, because half of them are so small. This is reflected in the error histories for the TSVD solutions: about every second increment of k leaves the error unchanged, when a small SVD component is included.

The Krylov subspace, on the other hand, is constructed from the starting vector $A^T b$ which has the SVD expansion

$$A^T b = A^T A\, x^{\text{exact}} + A^T e = \sum_{i=1}^{n} \sigma_i^2 \, (v_i^T x^{\text{exact}})\, v_i + \sum_{i=1}^{n} \sigma_i \, (u_i^T e)\, v_i .$$

Hence, about half of the SVD coefficients of this vector are solely due to the noise components, and therefore these components are small (they are zero for the ideal noise-free problem). Consequently, the Krylov space $\mathcal{K}_k$ is dominated by precisely those singular vectors that contribute most the the solution – and only the needed SVD directions are well represented in the Krylov subspace. Hence, the error decreases much more rapidly for the CGLS solution than the TSVD solution, see the right part of Fig. 4. We also have better suppression of the noise, because the unnecessary pure-noise TSVD components are avoided in the CGLS solution.

Other Krylov Methods – MR-II and RRGMRES

It is natural to seek iterative regularization methods based on other Krylov subspaces. For example, the matrix A may be symmetric in which case there is obviously no need for its transpose — or A may represent the discretization of a linear operator for which it is difficult or inconvenient to write a black-box function for multiplication with A^T. Hence there is an interest in the algorithms MINRES and GMRES, designed for symmetric and nonsymmetric square matrices respectively; both are based on the Krylov subspace span$\{b, A\,b, A^2\,b, \ldots, A^{k-1}\,b\}$. Unfortunately this subspace has a big disadvantage: it includes the noisy right-hand side $b = A\,x^{\text{exact}} + e$ and thus the noise component e. This means that the solution, obtained as a linear combination of the vectors $b, A\,b, A^2\,b, \ldots$, is likely to include a large noise component.

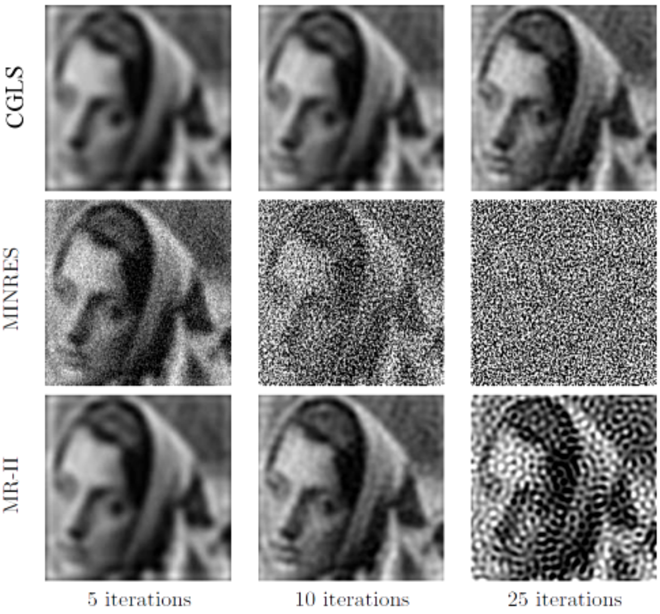


Fig. 5.

The solution to this problem is to work with the "shifted" Krylov subspace that starts with the vector $A\,b$, i.e.,

$$\vec{\mathcal{K}}_k = \mathrm{span}\{A\,b, A^2\,b, \ldots, A^k\,b\}. \tag{5}$$

The advantage of this subspace is that the noise component is now multiplied with A which has a smoothing effect and thus dampens the high-frequency components of the noise. The associated iterative algorithms based on $\vec{\mathcal{K}}_k$ are called MR-II [4] and RRGMRES [1]; see Fig. 5 for an illustrate of the advantage of MR-II over MINRES.

The performance and the regularizing properties of MR-II and RRGMRES are carefully studied in [12], where details can be found. To summarize:

- The absence of the vector b in the Krylov subspace $\vec{\mathcal{K}}_k$ (5) is essential for the use of MR-II and RRGMRES for discrete inverse problems.
- MR-II is a spectral filtering method. Negative eigenvalues of A do not inhibit the regularizing effect, but they can deteriorate the convergence rate.
- RRGMRES mixes the SVD components in each iteration and thus it does not provide a filtered SVD solution. The method works well if the mixing is weak (e.g, if A is nearly symmetric), or if the Krylov basis vectors are well suited for the problem. RRGMRES fails to produce regularized solutions when the mixing is strong, or when A has an unfavorable null space.

In conclusion, MR-II is a competitive alternative to CGLS for symmetric indefinite matrices, while for nonsymmetric matrices there is no transpose-free iterative regularization algorithm that can be used as a black-box method, in the same way as CGLS.

Noise Propagation in the Krylov Subspace Methods

To study how the noise in the data propagates to the solutions, we consider how CGLS, GMRES and RRGMRES treat the wanted signal contents from the exact component $b^{\mathrm{exact}} = A\,x^{\mathrm{exact}}$ as well as the contents from the noise component e. These results are from [11]. The CGLS solution can be written in terms of the bidiagonalization of A

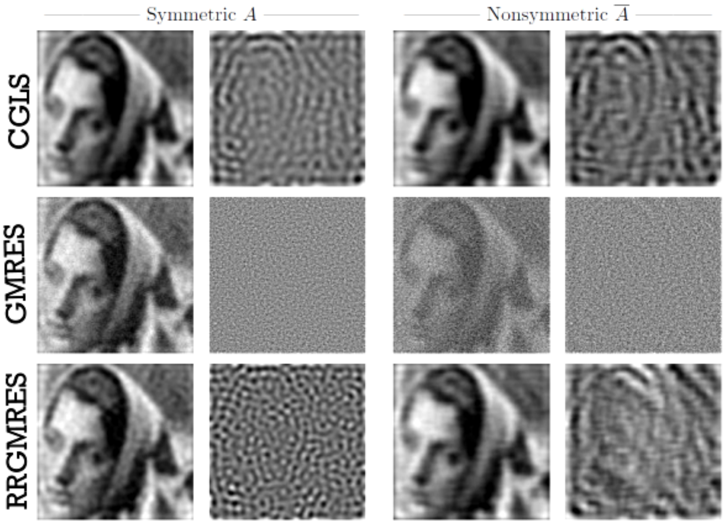


Fig. 6.

as $x^{(k)} = W_k\, B_k^\dagger\, U_{k+1}^T b$, and hence it can be split into the signal and noise components $x^{(k)} = x_{b^{\rm exact}}^{(k)} + x_e^{(k)}$ with

$$x_{b^{\rm exact}}^{(k)} = W_k\, B_k^\dagger\, U_{k+1}^T b^{\rm exact}, \qquad x_e^{(k)} = W_k\, B_k^\dagger\, U_{k+1}^T e.$$

Similarly for the GMRES and RRGMRES methods we have the splitting

$$\hat{x}^{(k)} = \hat{x}_{b^{\rm exact}}^{(k)} + \hat{x}_e^{(k)} \qquad \text{and} \qquad \tilde{x}^{(k)} = \tilde{x}_{b^{\rm exact}}^{(k)} + \tilde{x}_e^{(k)}.$$

Note that the Krylov subspace $\mathcal{K}_k$ is generated by b (and not $b^{\rm exact}$), and therefore the component $x_{b^{\rm exact}}^{(k)}$ differs from the CGLS solution produced with $b^{\rm exact}$ as starting vector. The same is true for GMRES and RRGMRES. This situation, where the signal component depends on the noise in the data, is unique for regularizing iterations (due to the dependence of the Krylov subspace on $b = b^{\rm exact} + e$).

Figure 6 illustrates the splitting for CGLS, GMRES and RRGMRES after 10 iterations, for two test problems with a symmetric and a non-symmetric matrix A for an isotropic and a non-isotropic point-spread function, respectively. We see how the noise propagates very differently in the three methods, due to the differences in the associated Krylov subspaces.

The CGLS algorithm produces low-frequent ringing effects in the signal component $x_{b^{\rm exact}}^{(k)}$ for both the symmetric and the nonsymmetric coefficient matrix. The noise component $x_e^{(k)}$ consists of bandpass-filtered noise in the form of freckles, and the shape of the freckles depends on the shape of the point-spread function. It is interesting to see how both the ringing in the signal component and the freckles in the noise component are correlated with the contours of the image, caused by the specific Krylov subspace.

GMRES propagate a white-noise component in the signal component $\hat{x}_{b^{\rm exact}}^{(k)}$, caused by the explicit presence of the noise in the basis vectors for the Krylov subspace. The white-noise component is particularly pronounced in the GMRES signal component.

The RRGMRES signal component behaves much like the CGLS signal component, except that it tends to carry more details after the same number of iterations. The noise components resemble those of the CGLS method. For the symmetric matrix, the freckles are smaller in diameter than for CGLS, and they are more visible in the signal component. For the nonsymmetric matrix, both components are quite similar to the CGLS components.

Subspace Preconditioning

Subspace preconditioning is a technique to improve the reconstructions by working in a modified subspace. To motivate this, it is well known that it is often advantageous to use Tikhonov regularization in the general form

$$\min_x \left\{ \|A\,x - b\|_2^2 + \lambda^2\, \|L\,x\|_2^2 \right\} \ , \tag{6}$$

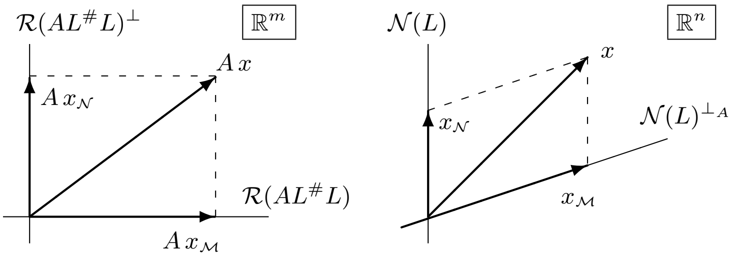
where L is a carefully chosen matrix (see [7] for more details). This problem can be turned into an equivalent problem in standard form

$$\min_{\bar{x}} \|\overline{A}\,\bar{x} - b\|_2^2 + \lambda^2 \|\bar{x}\|_2^2 \quad \text{with} \quad \overline{A} = A\,L^{\#} \quad \text{and} \quad x = L^{\#}\bar{x} + x_{\mathcal{N}} \tag{7}$$

where $L^{\#}$ is the oblique pseudoinverse of L and $x_{\mathcal{N}}$ is the component of the solution in $\mathcal{N}(L)$. If we apply CGLS to the standard-form problem (7) then the iterates, when transformed back to the original setting, lie in the affine space

$$\text{span}\{MA^Tb, (MA^TA)\,MA^Tb, (MA^TA)^2MA^Tb, \ldots, (MA^TA)^{k-1}MA^Tb\} + x_{\mathcal{N}} \ ,$$

where $M = L^{\#}(L^{\#})^T$. Hence we can think of L as a preconditioner for CGLS, with the purpose of providing a better suited subspace. The Krylov subspace methods are implemented in such as way that $\overline{A}$ is never formed explicitly; see [5] and [7] for implementation details.



Following [6], we can use a simple geometric argument to explain why we need to use the oblique pseudoinverse $L^{\#}$ in (7). The idea is to split the solution into two components $x = x_{\mathcal{M}} + x_{\mathcal{N}}$. The second component lies in the null space of L, i.e., $x_{\mathcal{N}} \in \mathcal{N}(L)$, while the first component $x_{\mathcal{M}}$ is A^TA–orthogonal to $x_{\mathcal{N}}$, i.e., it lies in a subspace $\mathcal{N}(L)^{\perp_A}$. This corresponds to an *oblique* splitting of the subspace $\mathbb{R}^n$.

Given this splitting, it follows immediately that the vector $A\,x = A\,x_{\mathcal{M}} + A\,x_{\mathcal{N}}$ splits into two *orthogonal* components, and thus the Tikhonov problem reduces to two independent problems for $x_{\mathcal{M}}$ and $x_{\mathcal{N}}$, respectively:

$$\min_{x_{\mathcal{M}}} \|A\,x_{\mathcal{M}} - b\|_2^2 + \lambda^2 \|x_{\mathcal{M}}\|_2^2 \qquad \text{and} \qquad \min_{x_{\mathcal{N}}} \|A\,x_{\mathcal{N}} - b\|_2^2.$$

The corresponding two subspaces in $\mathbb{R}^m$ are $\mathcal{R}(A\,L^{\#}L)$ and its orthogonal complement. Moreover, $x_{\mathcal{M}} = L^{\#}L\,x$ such that $A\,x_{\mathcal{M}} = (A\,L^{\#})\,(L\,x)$ leading to the standard-form problem (7); we refer to [6] for details.

It can be shown that the matrix $L^{\#}L$ is the oblique projector on $\mathcal{N}(L)^{\perp_A}$ along $\mathcal{N}(L)$, and that $\overline{A} = A\,L^{\#}$ is the corresponding oblique pseudoinverse. How to efficiently implement the computations with $L^{\#}$ is described in, e.g., [5] and [7].

An alternative formulation of the standard-form transformation suited for the algorithms MR-II and RRGMRES is described in [10].

References

[1] D. Calvetti, B. Lewis, and L. Reichel, *GMRES-type methods for inconsistent systems*, Linear Algebra Appl., 316 (2000), pp. 157-169.

[2] J. Chung, S. Knepper, and J. Nagy, *Large-Scale Inverse Problems in Imaging;* in O. Scherzer (ed.) *Handbook of Mathematical Methods in Imaging*, Springer, Berline, 2011, pp. 43–86.

[3] J. Chung, J. Nagy, and D. P. O'Leary, *A weighted GCV Method for Lanczos hybrid regularization*, Elec. Trans. Numer. Anal., 28 (2008), pp. 149–167.

[4] M. Hanke, *Conjugate Gradient Type Methods for Ill-Posed Problems*, Longman Scientific & Technical, Essex, 1995.

[5] M. Hanke and P. C. Hansen, *Regularization methods for large-scale problems*, Surveys Math. Indust., 3 (1993), pp. 253–315.

[6] P. C. Hansen, *Oblique projections and standard-form transformations for discrete inverse problems*, Numer. Lin. Alg. Appl., to appear.

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

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

[9] P. C. Hansen, J. G. Nagy, and D. P. O'Leary, *Deblurring Images: Matrices, Spectra, and Filtering*, SIAM, Philadelphia, 2006 (130 pages).

[10] P. C. Hansen and T. K. Jensen, *Smoothing-norm preconditioning for regularizing minimum-residual methods*, SIAM J. Matrix Anal. Appl., 29 (2006), pp. 1–14.

[11] P. C. Hansen and T. K. Jensen, *Noise propagation in regularizing iterations for image deblurring*, Electronic Transactions on Numerical Analysis, 31 (2008), pp. 204–220.

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

[13] J. G. Nagy, K. P. Lee, and L. Perrone, *Iterative methods for image deblurring: A Matlab object oriented approach*, Numer. Algo., 36 (2004), pp. 73–93. Software available from www.mathcs.emory.edu/∼nagy/RestoreTools.