Electronic Transactions on Numerical Analysis.
Volume 28, pp. 149-167, 2008.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

A WEIGHTED-GCV METHOD FOR LANCZOS-HYBRID REGULARIZATION*

JULIANNE CHUNG†, JAMES G. NAGY†, AND DIANNE P. O'LEARY‡

In memory of Gene Golub

Abstract. Lanczos-hybrid regularization methods have been proposed as effective approaches for solving large-scale ill-posed inverse problems. Lanczos methods restrict the solution to lie in a Krylov subspace, but they are hindered by semi-convergence behavior, in that the quality of the solution first increases and then decreases. Hybrid methods apply a standard regularization technique, such as Tikhonov regularization, to the projected problem at each iteration. Thus, regularization in hybrid methods is achieved both by Krylov filtering and by appropriate choice of a regularization parameter at each iteration. In this paper we describe a weighted generalized cross validation (W-GCV) method for choosing the parameter. Using this method we demonstrate that the semi-convergence behavior of the Lanczos method can be overcome, making the solution less sensitive to the number of iterations.

Key words. generalized cross validation, ill-posed problems, iterative methods, Lanczos bidiagonalization, LSQR, regularization, Tikhonov

AMS subject classifications. 65F20, 65F30

1. Introduction. Linear systems that arise from large-scale inverse problems are typically written as

$$\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}_{\text{true}} + \boldsymbol{\varepsilon}\,, \tag{1.1}$$

where $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, and $\boldsymbol{x}_{\text{true}} \in \mathbb{R}^n$. The vector $\boldsymbol{\varepsilon} \in \mathbb{R}^m$ represents unknown perturbations in the data (such as noise). We will assume that the perturbations are independent and identically distributed with zero mean; this can often be achieved by scaling the original problem. Given $\boldsymbol{A}$ and $\boldsymbol{b}$, the aim is to compute an approximation of $\boldsymbol{x}_{\text{true}}$.

Inverse problems of the form (1.1) arise in many important applications, including image reconstruction, image deblurring, geophysics, parameter identification and inverse scattering; cf. [8, 18, 19, 32]. Typically these problems are *ill-posed*, meaning that noise in the data may give rise to significant errors in computed approximations of $\boldsymbol{x}_{\text{true}}$. The ill-posed nature of the problem is revealed by the singular values of $\boldsymbol{A}$, which decay to and cluster at 0. Thus $\boldsymbol{A}$ is severely ill-conditioned, and *regularization* is used to compute stable approximations of $\boldsymbol{x}_{\text{true}}$ [8, 15, 18, 32]. Regularization can take many forms; probably the most well known choice is Tikhonov regularization [15], which is equivalent to solving the least squares problem

$$\min_{\boldsymbol{x}} \left\| \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix} - \begin{bmatrix} \boldsymbol{A} \\ \lambda \boldsymbol{L} \end{bmatrix} \boldsymbol{x} \right\|_2 , \tag{1.2}$$

where $\boldsymbol{L}$ is a *regularization operator*, often chosen as the identity matrix or a discretization of a differentiation operator. The *regularization parameter* λ is a scalar, usually satisfying $\sigma_n \le \lambda \le \sigma_1$, where σ_n is the smallest singular value of $\boldsymbol{A}$ and σ_1 is the largest singular value of $\boldsymbol{A}$.

*Received March 7, 2007. Accepted for publication September 12, 2007. Recommended by L. Reichel. The work of the first author was supported in part by a DOE Computational Sciences Graduate Research Fellowship. The work of the second author was supported in part by NSF grant DMS-05-11454 and by an Emory University Research Committee grant. The work of the third author was supported in part by NSF Grant CCF 0514213.

†Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322, USA (`{jmchung,nagy}@mathcs.emory.edu`).

‡Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742, USA; and National Institute for Standards and Technology, Gaithersburg, MD 20899, USA (`oleary@cs.umd.edu`).

No regularization method is effective without an appropriate choice of the regularization parameter. Various techniques can be used, such as the discrepancy principle, the L-curve, and generalized cross validation (GCV) [8, 18, 32]. There are advantages and disadvantages to each of these approaches [23], especially for large-scale problems. For example, to use the discrepancy principle, it is necessary to have information about the noise. In the case of GCV, efficient implementation for Tikhonov regularization requires computing the singular value decomposition (SVD) of the matrix $\boldsymbol{A}$ [13], which may be computationally impractical for large-scale problems. Some savings can be attained by using a bidiagonalization of $\boldsymbol{A}$ [7], or the iterative technique proposed by Golub and von Matt [14], but the cost can still be prohibitive for very large matrices. In addition, the method proposed in [14] would need to be implemented carefully to avoid failure when a trial choice of parameter in the iteration is poor [5]. In the case of the L-curve, it may be necessary to solve (1.2) for several regularization parameters. This limitation can be partially alleviated by exploiting redundancies and additional information available in certain iterative methods [3, 10].

An alternative to Tikhonov regularization for large-scale problems is *iterative regularization*. In this case, an iterative method such as LSQR [30] is applied to the least squares problem,

$$\min_{\boldsymbol{x}} \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|_2. \tag{1.3}$$

When applied to ill-posed problems, iterative methods such as LSQR exhibit an interesting "semiconvergence" behavior. Specifically, the early iterations reconstruct information about the solution, while later iterations reconstruct information about the noise. This behavior can be observed (if the exact solution is known) by plotting the relative errors, $\|\boldsymbol{x}_k - \boldsymbol{x}_{\text{true}}\|_2/\|\boldsymbol{x}_{\text{true}}\|_2$, where $\boldsymbol{x}_{\text{true}}$ is the exact solution and $\boldsymbol{x}_k$ is the solution at the kth iteration. This is illustrated schematically in the left plot of Figure 1.1, where we plot the typical behavior of the relative error as the iteration proceeds. (Specific examples are detailed in later sections.) If we terminate the iteration when the error is minimized, we obtain a regularized solution. Unfortunately the exact solution $\boldsymbol{x}_{\text{true}}$ is not known in practical problems, so a plot of the relative errors cannot be used to find the optimal termination point. However, parameter selection methods such as the discrepancy principle, GCV and L-curve (see, for example, [18]) can be used to estimate this termination point. The difficulty is that these techniques are not perfect, and, as illustrated in the left plot in Figure 1.1, an imprecise estimate of the termination point can result in a solution whose relative error is significantly higher than the optimal.

The semiconvergence behavior of LSQR can be stabilized by using a *hybrid* method that combines an iterative Lanczos bidiagonalization algorithm with a direct regularization scheme, such as Tikhonov [1, 2, 4, 16, 22, 23, 25, 29] or truncated SVD. The basic idea of this approach is to project the large-scale problem onto Krylov subspaces of small (but increasing) dimension. The projected problem can be solved cheaply using any direct regularization method. The potential benefits of this approach are illustrated in the right plot of Figure 1.1. Notice that, in contrast to the behavior of the relative errors for LSQR, the hybrid approach can effectively stabilize the iteration so that an imprecise (over) estimate of the stopping iteration does not have a deleterious effect on the computed solution.

A disadvantage of the hybrid approach is that at each iteration we must choose a new regularization parameter for the projected problem. Although this is not computationally expensive, in order for the approach to be viable for practical problems, we must choose good parameters. Optimal choices for the parameter at each iteration result in convergence behavior similar to that illustrated in the right plot of Figure 1.1. However, our computational experience indicates that such optimal behavior cannot be expected when using parameter

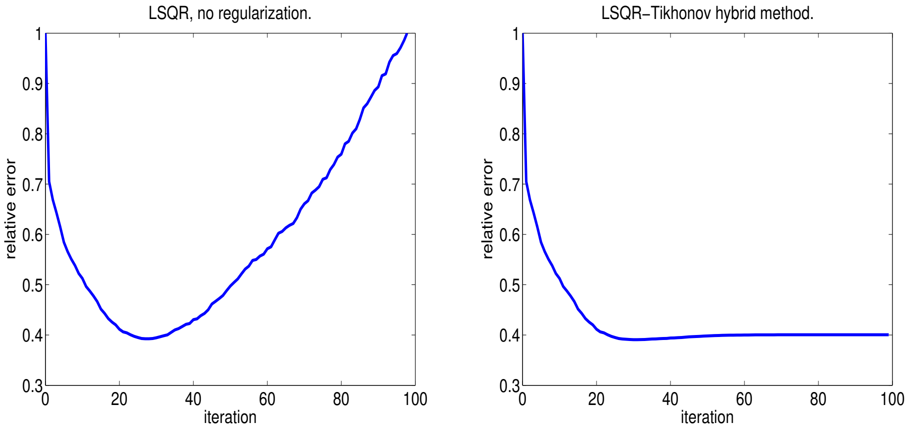

FIG. 1.1. *These plots represent relative errors, $\|\mathbf{x}_k - \mathbf{x}_{\text{true}}\|_2/\|\mathbf{x}_{\text{true}}\|_2$, where $\mathbf{x}_{\text{true}}$ is the true solution, and $\mathbf{x}_k$ is the solution at the kth iteration. The left plot illustrates semiconvergence behavior of the iterative method LSQR for an ill-posed problem; a regularized solution is computed by terminating the iteration when the relative error is small, but the error can be very large if this termination point is over-estimated. The right plot illustrates how this semiconvergence behavior can be stabilized with an iterative LSQR-Tikhonov hybrid method.*

selection methods such as the discrepancy principle, GCV and the L-curve; see also [23].

In this paper we consider a Lanczos-hybrid method, using Tikhonov regularization, with the regularization parameter for the projected problem chosen by GCV. We show that GCV has a strong tendency to over-estimate the regularization parameter, but that a weighted-GCV (W-GCV) method can be very effective.

An outline for the rest of the paper is as follows. In section 2 we review Tikhonov regularization, the GCV method, and SVD based implementations. In section 3 we describe the Lanczos-hybrid method, with Tikhonov regularization for the projected problem. We illustrate the deficiencies of using GCV with that method in section 4. We show that although it is efficient, it generally provides a parameter estimate that is too large. This can seriously degrade the overall convergence behavior. In section 5, we describe the W-GCV method and show how it is related to the standard GCV. Numerical experiments are provided in section 6 that illustrate the effectiveness of the W-GCV method on the test problems (introduced in section 4.1), and some concluding remarks are given in section 7.

2. Tikhonov regularization and GCV. To establish notation used in the paper, we briefly review Tikhonov regularization and GCV. In particular, we show that by using the SVD of the matrix $\boldsymbol{A}$, we can recast the Tikhonov problem as a filtering method. In addition, the SVD allows us to put the GCV function into a computationally convenient form. Although this SVD approach is impractical for large-scale problems, it is both an extremely useful tool for problems of small dimension and an important component of the Lanczos-hybrid method.

Tikhonov regularization requires solving the minimization problem given in (1.2). For ease of notation, we take $\boldsymbol{L}$ to be the identity matrix. Let $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ denote the SVD of $\boldsymbol{A}$, where the columns $\boldsymbol{u}_i$ of $\boldsymbol{U}$ and $\boldsymbol{v}_i$ of $\boldsymbol{V}$ contain, respectively, the left and right singular vectors of $\boldsymbol{A}$, and $\boldsymbol{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n)$ is a diagonal matrix containing the singular values of $\boldsymbol{A}$, with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$. Replacing $\boldsymbol{A}$ by its SVD and performing a little algebraic manipulation, we obtain the Tikhonov regularized solution

$$\boldsymbol{x}_\lambda = \sum_{i=1}^{n} \phi_i \frac{\boldsymbol{u}_i^T \boldsymbol{b}}{\sigma_i} \boldsymbol{v}_i, \tag{2.1}$$

where $\phi_i = \dfrac{\sigma_i^2}{\sigma_i^2 + \lambda^2} \in [0, 1]$ are the Tikhonov filter factors; cf. [18]. Note that choosing $\lambda = 0$ corresponds to $\phi_i = 1$ for all i, which in turn gives the solution to (1.3). The regularization parameter, λ, plays a crucial role in the quality of the solution. For example, if λ is too large, the filter factors damp (or, equivalently, filter out) too many of the components in the SVD expansion (2.1), and the corresponding solution is *over-smoothed*. On the other hand, if λ is too small, the filter factors damp too few components, and the corresponding solution is *under-smoothed*.

As mentioned in the introduction, a variety of parameter choice methods can be used to determine λ. We choose to use GCV, which is a predictive statistics-based method that does not require a priori estimates of the error norm. The basic idea of GCV is that a good choice of λ should predict missing values of the data. That is, if an arbitrary element of the observed data is left out, then the corresponding regularized solution should be able to predict the missing observation fairly well [18]. We leave out each data value b_j in turn and seek the value of λ that minimizes the prediction errors, measured by the GCV function

$$G_{A,\,b}(\lambda) = \frac{n\|(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}_\lambda^\dagger)\boldsymbol{b}\|_2^2}{\left(\mathrm{trace}(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}_\lambda^\dagger)\right)^2}\,, \tag{2.2}$$

where $\boldsymbol{A}_\lambda^\dagger = (\boldsymbol{A}^T\boldsymbol{A} + \lambda^2\boldsymbol{I})^{-1}\boldsymbol{A}^T$ represents the pseudo-inverse of $\begin{bmatrix} \boldsymbol{A} \\ \lambda\boldsymbol{I} \end{bmatrix}$, and gives the regularized solution, $\boldsymbol{x}_\lambda = \boldsymbol{A}_\lambda^\dagger\boldsymbol{b}$. Replacing $\boldsymbol{A}$ with its SVD, (2.2) can be rewritten as

$$G_{A,\,b}(\lambda) = \frac{n\left(\displaystyle\sum_{i=1}^{n}\left(\frac{\lambda^2\boldsymbol{u}_i^T\boldsymbol{b}}{\sigma_i^2 + \lambda^2}\right)^2 + \sum_{i=n+1}^{m}(\boldsymbol{u}_i^T\boldsymbol{b})^2\right)}{\left((m-n) + \displaystyle\sum_{i=1}^{n}\frac{\lambda^2}{\sigma_i^2 + \lambda^2}\right)^2}\,, \tag{2.3}$$

which is a computationally convenient form to evaluate, thus making GCV easily used with standard minimization algorithms.

3. Lanczos-hybrid methods. Using GCV to determine the Tikhonov regularization parameter can be quite effective, but the minimization function (2.3) requires that the SVD of the matrix $\boldsymbol{A}$ be computed, and this is not feasible when $\boldsymbol{A}$ is too big. This leads us to Lanczos-hybrid methods, which make computing the SVD of the operator feasible by projecting the problem onto a subspace of small dimension. As described in section 1 and illustrated in Figure 1.1, hybrid methods can be an effective way to stabilize the semiconvergent behavior that is characteristic of iterative methods like LSQR when applied to ill-posed problems. Using an iterative method like Lanczos bidiagonalization (LBD) in combination with a direct method like Tikhonov regularization on the projected problem, we can hope to efficiently solve large-scale, ill-posed inverse problems. In this section, we provide some background on the Lanczos-hybrid methods.

Given a matrix $\boldsymbol{A}$ and vector $\boldsymbol{b}$, LBD is an iterative scheme that computes the decomposition

$$\boldsymbol{W}^T\boldsymbol{A}\boldsymbol{Y} = \boldsymbol{B}\,,$$

where $\boldsymbol{W}$ and $\boldsymbol{Y}$ are orthonormal matrices, and $\boldsymbol{B}$ is a lower bidiagonal matrix. The kth iteration of LBD computes the kth columns of $\boldsymbol{Y}$ and $\boldsymbol{B}$, and the $(k+1)$st column of $\boldsymbol{W}$.

Specifically, at iteration k, for $k = 1, \ldots, n$, we have an $m \times (k+1)$ matrix $\boldsymbol{W}_k$, an $n \times k$ matrix $\boldsymbol{Y}_k$, an $n \times 1$ vector $\boldsymbol{y}_{k+1}$, and a $(k+1) \times k$ bidiagonal matrix $\boldsymbol{B}_k$ such that

$$\boldsymbol{A}^T \boldsymbol{W}_k = \boldsymbol{Y}_k \boldsymbol{B}_k^T + \alpha_{k+1} \boldsymbol{y}_{k+1} \boldsymbol{e}_{k+1}^T, \tag{3.1}$$

$$\boldsymbol{A} \boldsymbol{Y}_k = \boldsymbol{W}_k \boldsymbol{B}_k, \tag{3.2}$$

where $\boldsymbol{e}_{k+1}$ denotes the last column of the identity matrix of dimension $(k+1)$ and α_{k+1} will be the $(k+1)$st diagonal entry of $\boldsymbol{B}_{k+1}$. Matrices $\boldsymbol{W}_k$ and $\boldsymbol{Y}_k$ have orthonormal columns, and the first column of $\boldsymbol{W}_k$ is $\boldsymbol{b}/\|\boldsymbol{b}\|$.

Given these relations, we approximate the least squares problem

$$\min_{\boldsymbol{x}} ||\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}||_2$$

by the *projected* LS problem

$$\begin{aligned} \min_{\boldsymbol{x} \in R(\boldsymbol{Y}_k)} ||\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}||_2 &= \min_{\boldsymbol{f}} ||\boldsymbol{W}_k^T \boldsymbol{b} - \boldsymbol{B}_k \boldsymbol{f}||_2 \\ &= \min_{\boldsymbol{f}} ||\beta \boldsymbol{e}_1 - \boldsymbol{B}_k \boldsymbol{f}||_2 \,, \end{aligned} \tag{3.3}$$

where $\beta = \|\boldsymbol{b}\|$, and choose our approximate solution as $\boldsymbol{x}_k = \boldsymbol{Y}_k \boldsymbol{f}$. Thus each iteration of the LBD method requires solving a least squares problem involving a bidiagonal matrix $\boldsymbol{B}_k$. Implementations of LBD iterative methods such as LSQR do not explicitly form the matrices $\boldsymbol{W}_k$, $\boldsymbol{Y}_k$, and $\boldsymbol{B}_k$ when solving well-conditioned problems. Instead, efficient updating of the solution is used, and only a few vectors are stored [30]. For ill-conditioned problems, though, the matrices are often stored so that regularization can be applied.

An important property of LBD is that the singular values of $\boldsymbol{B}_k$ for small values of k tend to approximate the largest and smallest singular values of $\boldsymbol{A}$ [12]. Since the original problem is ill-posed, $\boldsymbol{B}_k$ may become very ill-conditioned. Therefore, regularization must be used to compute

$$\boldsymbol{f}_\lambda = \beta \boldsymbol{B}_{k,\lambda}^\dagger \boldsymbol{e}_1 \,,$$

as described in section 2. Notice that since the dimension of $\boldsymbol{B}_k$ is very small compared to $\boldsymbol{A}$, we can afford to use SVD-based filtering methods to solve for $\boldsymbol{f}_\lambda$ and SVD-based parameter choice methods to find λ at each iteration. O'Leary and Simmons [29] proposed using Tikhonov regularization to solve the projected problem, and Björck [1] suggested using truncated SVD (TSVD) with GCV to choose the regularization parameters. A variety of existing methods can be implemented. For a comparative study; see Kilmer and O'Leary [23]. Björck [1] also suggested using GCV as a way to determine an appropriate stopping iteration.

In the next section we illustrate how well this method works for Tikhonov regularization, using the GCV function,

$$G_{B_k,\,\beta e_1}(\lambda) = \frac{k\|(\boldsymbol{I} - \boldsymbol{B}_k \boldsymbol{B}_{k,\lambda}^\dagger)\beta \boldsymbol{e}_1\|_2^2}{\left(\text{trace}(\boldsymbol{I} - \boldsymbol{B}_k \boldsymbol{B}_{k,\lambda}^\dagger)\right)^2} \,,$$

to choose regularization parameters for (3.3) at each iteration. Note that if we define the SVD of the $(k+1) \times k$ matrix $\boldsymbol{B}_k$ as

$$\boldsymbol{B}_k = \boldsymbol{P}_k \begin{bmatrix} \boldsymbol{\Delta}_k \\ \boldsymbol{0}^T \end{bmatrix} \boldsymbol{Q}_k^T \,, \tag{3.4}$$

then $G_{B_k,\beta e_1}(\lambda)$ can be written as

$$G_{B_k,\beta e_1}(\lambda) = \frac{k\beta^2 \left(\sum_{i=1}^{k} \left(\frac{\lambda^2}{\delta_i^2 + \lambda^2} \left[\boldsymbol{P}_k^T \boldsymbol{e}_1\right]_i \right)^2 + \left(\left[\boldsymbol{P}_k^T \boldsymbol{e}_1\right]_{k+1} \right)^2 \right)}{\left(1 + \sum_{i=1}^{k} \frac{\lambda^2}{\delta_i^2 + \lambda^2} \right)^2}, \tag{3.5}$$

where $\left[\boldsymbol{P}_k^T \boldsymbol{e}_1\right]_j$ denotes the jth component of the vector $\boldsymbol{P}_k^T \boldsymbol{e}_1$, and δ_i is the ith largest singular value of $\boldsymbol{B}_k$ (i.e., the ith diagonal element of $\boldsymbol{\Delta}_k$).

4. Experimental results using Lanczos-hybrid methods and GCV. To illustrate the behavior of the GCV and W-GCV methods in Lanczos-hybrid methods, we use six problems. All computations are done in MATLAB. Data and code used in this paper can be obtained from `http://www.mathcs.emory.edu/~nagy/WGCV`.

4.1. Test problems. The first problem comes from the iterative image deblurring package, 'RestoreTools' [26]. Image deblurring has the form $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}_{\text{true}} + \boldsymbol{\varepsilon}$, where the vector $\boldsymbol{x}_{\text{true}}$ represents the true image scene, $\boldsymbol{A}$ is a matrix representing a blurring operation, and $\boldsymbol{b}$ is a vector representing the observed, blurred and noisy image. Given $\boldsymbol{A}$ and $\boldsymbol{b}$, the aim is to reconstruct an approximation of $\boldsymbol{x}_{\text{true}}$. The RestoreTools package has several data sets and tools (such as matrix construction and multiplication routines) that can be used with iterative methods. The data set we use consists of a true image of a *satellite* and a so-called point spread function (PSF) that defines the blurring operation. The matrix $\boldsymbol{A}$ is constructed from the PSF, using a matrix construction routine in RestoreTools. We then form the noise-free blurred image as $\boldsymbol{b}_{\text{true}} = \boldsymbol{A}\boldsymbol{x}_{\text{true}}$. The MATLAB instructions are:

```
>> load satellite
>> A = psfMatrix(PSF);
>> b_true = A*x_true;
```

The images have 256×256 pixels, so the vectors $\boldsymbol{b}_{\text{true}}$ and $\boldsymbol{x}_{\text{true}}$ have length $256^2 = 65,536$. The function `psfMatrix` uses an efficient data structure scheme to represent the $65,536 \times 65,536$ matrix $\boldsymbol{A}$, and the multiplication operator, *, is overloaded to allow for efficient computation of matrix-vector multiplications; see [26] for more details.

The other five test problems are taken from the 'Regularization Tools' package [17]. In each case we generate an $n \times n$ matrix $\boldsymbol{A}$, true solution vector $\boldsymbol{x}_{\text{true}}$, and (noise-free) observation vector $\boldsymbol{b}_{\text{true}}$, setting $n = 256$.

- ***Phillips*** is Phillips' "famous" test problem. $\boldsymbol{A}$, $\boldsymbol{b}$, and $\boldsymbol{x}_{\text{true}}$ are obtained by discretizing the first kind Fredholm integral equation $b(s) = \int_{-6}^{6} a(s,t)x(t)dt$, where

$$a(s,t) = \begin{cases} 1 + \cos(\frac{\pi(s-t)}{3}) & , \quad |s-t| < 3, \\ 0 & , \quad |s-t| \geq 3, \end{cases}$$
$$x(t) = \begin{cases} 1 + \cos(\frac{\pi t}{3}) & , \quad |t| < 3, \\ 0 & , \quad |t| \geq 3, \end{cases}$$
$$b(s) = (6 - |s|)\left(1 + \frac{1}{2}\cos(\frac{\pi s}{3})\right) + \frac{9}{2\pi}\sin(\frac{\pi |s|}{3}).$$

  In MATLAB, the problem can be constructed with the simple statement:

```
>> [A, b_true, x_true] = phillips(n);
```

  where `n` is the dimension of the problem.

- ***Shaw*** is a one-dimensional image restoration problem. $\boldsymbol{A}$ and $\boldsymbol{x}_{\text{true}}$ are obtained by discretizing, on the interval $-\frac{\pi}{2} \leq s, t \leq \frac{\pi}{2}$, the functions

$$a(s,t) = (\cos(s) + \cos(t)) \left(\frac{\sin(u)}{u} \right)^2, \quad u = \pi(\sin(s) + \sin(t)),$$
$$x(t) = 2\exp(-6(t-0.8)^2) + \exp(-2(t+0.5)^2),$$

  and $\boldsymbol{b}_{\text{true}} = \boldsymbol{A}\boldsymbol{x}_{\text{true}}$. The data can be constructed with the simple MATLAB statement:

```
>> [A, b_true, x_true] = shaw(n);
```

  where `n` is the dimension of the problem.
- ***Deriv2*** constructs $\boldsymbol{A}$, $\boldsymbol{b}$ and $\boldsymbol{x}_{\text{true}}$ by discretizing a first kind Fredholm integral equation, $b(s) = \int_0^1 a(s,t)x(t)dt$, $0 \leq s \leq 1$, where the kernel $a(s,t)$ is given by the Green's function for the second derivative:

$$a(s,t) = \begin{cases} s(t-1) & , \quad s < t, \\ t(s-1) & , \quad s \geq t. \end{cases}$$

  There are several choices for x and b; in this paper, we use $x(t) = t$ and $b(s) = (s^3 - s)/6$. The data can be constructed with the simple MATLAB statement:

```
>> [A, b_true, x_true] = deriv2(n);
```

  where `n` is the dimension of the problem.
- ***Baart*** constructs $\boldsymbol{A}$, $\boldsymbol{b}$ and $\boldsymbol{x}_{\text{true}}$ by discretizing the first kind Fredholm integral equation $b(s) = \int_0^\pi a(s,t)x(t)dt$, $0 \leq s \leq \frac{\pi}{2}$, where

$$a(s,t) = \exp(s\cos t),$$
$$x(t) = \sin t,$$
$$b(s) = \frac{2\sinh s}{s}.$$

  The data can be constructed with the simple MATLAB statement:

```
>> [A, b_true, x_true] = baart(n);
```

  where `n` is the dimension of the problem.
- ***Heat*** is an inverse heat equation using the Volterra integral equation of the first kind on $[0,1]$ with kernel $a(s,t) = k(s-t)$, where

$$k(t) = \frac{t^{-3/2}}{2\sqrt{\pi}} \exp\left(-\frac{1}{4t} \right).$$

  The vector $\boldsymbol{x}_{\text{true}}$ does not have a simple functional representation, but rather is constructed directly as a discrete vector; see [17] for details. The right-hand side $\boldsymbol{b}$ is produced as $\boldsymbol{b}_{\text{true}} = \boldsymbol{A}\boldsymbol{x}_{\text{true}}$. The data can be constructed with the simple MATLAB statement:

```
>> [A, b_true, x_true] = heat(n);
```

  where `n` is the dimension of the problem.

In order to simulate noisy data, as modeled by equation (1.1), for each test problem, we generate a noise vector $\boldsymbol{\varepsilon}$ whose entries are chosen from a normal distribution with mean 0 and variance 1, and scaled so that

$$\frac{\|\boldsymbol{\varepsilon}\|_2}{\|\boldsymbol{A}\boldsymbol{x}_{\text{true}}\|_2} = 0.1 \quad \text{(i.e., noise level = 10\%)}.$$

4.2. What goes wrong in using GCV-Lanczos?. We solved these test problems with the Lanczos-based hybrid method, using GCV to choose the Tikhonov regularization parameter λ_k at each iteration. The results are shown in Figure 4.1. In all of our examples, LSQR, which is essentially LBD with no regularization, exhibits semiconvergent behavior, as we expect. If we use 'optimal' regularization parameters at each iteration (determined using knowledge of $\boldsymbol{x}_{\rm true}$ to make the relative error in the solution as small as possible), then Lanczos-hybrid methods would be excellent at stabilizing the regularized solution, as shown with the dashed lines. However, in realistic situations, we do not know the optimal solution, so this is impossible. On the *Phillips, Shaw* and *Deriv2* problems, the performance of standard GCV, though slightly worse than optimal, is acceptable. For the other three problems, the convergence behavior for GCV is significantly worse than optimal.

A major concern is the possibility that rounding errors in the computation of the matrices $\boldsymbol{W}_k$, $\boldsymbol{Y}_k$ and $\boldsymbol{B}_k$ are causing the poor behavior. Björck, Grimme and Van Dooren [2] showed that in some cases reorthogonalization may be necessary for better performance, and Larsen [25] considered partial reorthogonalization. However, in our tests GCV still had difficulty even after reorthogonalization. Another option is to use a different regularization method such as TSVD or exponential filtering, but we found little to no improvement in the solution. In addition, we delayed regularization until after $k > k_{min}$ to wait until $\boldsymbol{B}_k$ more fully captures the ill-conditioning of $\boldsymbol{A}$, but that attempt proved futile as well.

We now know that there are good choices of the regularization parameter, so the poor behavior is caused by the suboptimal parameter chosen by GCV. In the next section we propose replacing it by a weighted-GCV method which shows much better behavior.

5. Weighted-GCV. In this section we describe a modification of the GCV function, which we call *weighted-GCV* (W-GCV), that will improve our ability to choose regularization parameters for the projected problem. We first describe the approach for Tikhonov regularization for a general linear system of equations, and then show in section 5.3 how to apply it to the projected problem.

5.1. W-GCV for Tikhonov regularization. The standard GCV method is a popular parameter choice method used in a variety of applications; however, as we have just seen, the method may not perform well for certain classes of problems. Other studies in statistical nonparametric modeling and function approximation noted that in practical applications, GCV occasionally chose Tikhonov parameters too small, thereby under-smoothing the solution [6, 9, 24, 28, 31]. To circumvent this problem, these papers use a concept that we call weighted-GCV. In contrast, we observed over-smoothing difficulties when using GCV in Lanczos-hybrid methods, which motivated us to use a different range of weights in the W-GCV method.

Instead of the Tikhonov GCV function defined in (2.2), we consider the weighted-GCV function

$$G_{_{A,\,b}}(\omega,\lambda) = \frac{n||(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}_\lambda^\dagger)\boldsymbol{b}||^2}{\left(\mathrm{trace}(\boldsymbol{I} - \omega\boldsymbol{A}\boldsymbol{A}_\lambda^\dagger)\right)^2}.$$

Notice the function's dependency on a new parameter ω in the denominator trace term. Choosing $\omega = 1$ gives the standard GCV function (2.2). If we choose $\omega > 1$, we obtain smoother solutions, while $\omega < 1$ results in less smooth solutions. The obvious question here is how to choose a good value for ω. To our knowledge, in all work using W-GCV, only experimental approaches are used to choose ω. For smoothing spline applications, Kim and Gu empirically found that standard GCV consistently produced regularization parameters that

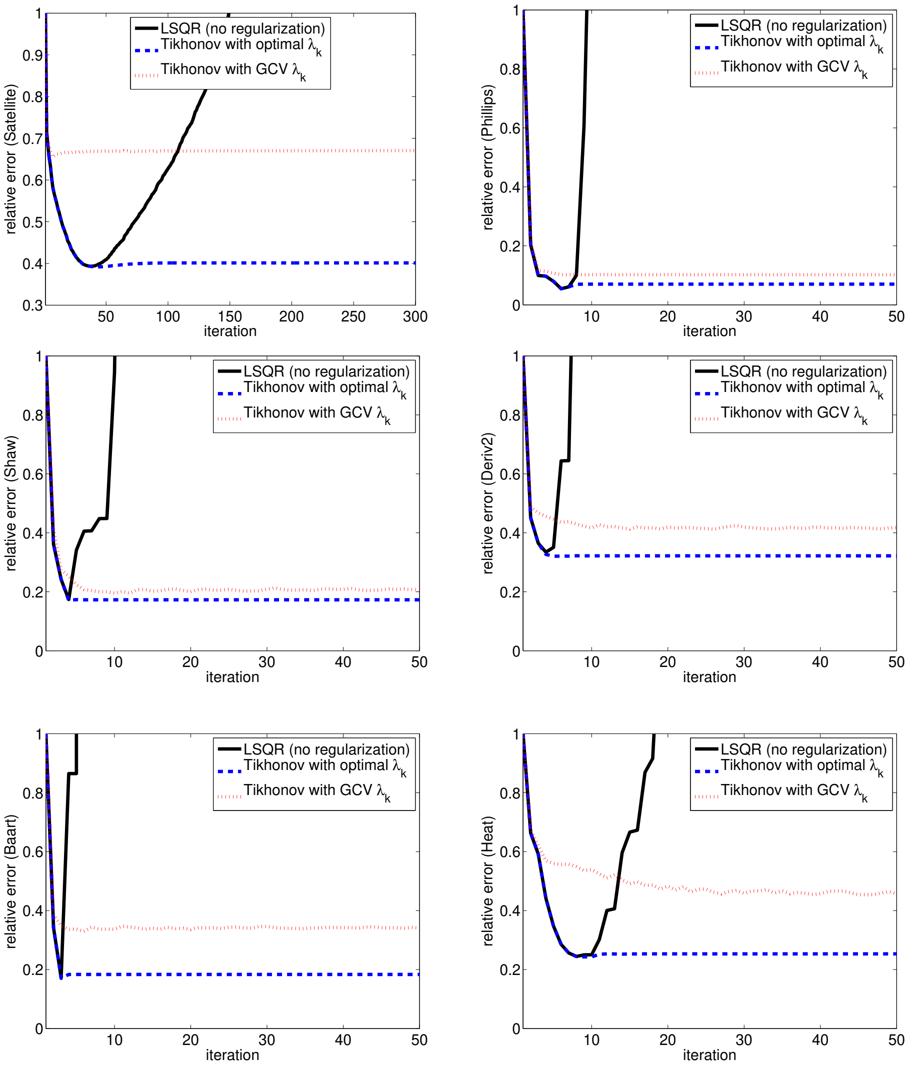

FIG. 4.1. *These plots show the relative error,* $\|\mathbf{x}_k - \mathbf{x}_{\text{true}}\|_2 / \|\mathbf{x}_{\text{true}}\|_2$*, at each iteration of LSQR and the Lanczos-hybrid method. Upper left: Satellite. Upper right: Regtools-Phillips. Middle left: Regtools-Shaw. Middle right: Regtools-Deriv2. Bottom left: Regtools-Baart. Bottom right: Regtools-Heat. The standard GCV method chooses regularization parameters that are too large at each iteration, which causes poor convergence behavior.*

were too small, while choosing ω in the range of 1.2-1.4 worked well [24]. In our problems, though, the GCV regularization parameter is chosen too large, and thus we seek a parameter ω in the range $0 < \omega \leq 1$. In addition, rather than using a user-defined parameter choice for ω as in previous papers, we propose a more automated approach that is also versatile and can be used on a variety of problems.

5.2. Interpretations of the W-GCV method. In this section, we consider the W-GCV method and look at various theoretical aspects of the method. By looking at different interpretations of the W-GCV method, we hope to shed some light on the role of the new parameter ω.

As mentioned in section 2, the standard GCV method is a "leave-one-out" prediction method. In fact, in leaving out the jth observation, the derivation seeks to minimize the prediction error $(b_j - [\boldsymbol{Ax}]_j)^2$, when $\boldsymbol{x}$ is the minimizer of

$$\sum_{i=1, i\neq j}^{m} (b_i - [\boldsymbol{Ax}]_i)^2 + \lambda^2 ||\boldsymbol{x}||_2^2.$$

If we define the $m \times m$ matrix

$$\boldsymbol{E}_j = \mathrm{diag}(1, 1, ...1, 0, 1, ...1),$$

where 0 is the j^{th} entry, then the above minimization is equivalent to

$$\min_{\boldsymbol{x}} ||\boldsymbol{E}_j(\boldsymbol{b} - \boldsymbol{Ax})||_2^2 + \lambda^2 ||\boldsymbol{x}||_2^2.$$

We can derive the W-GCV method in a similar manner, but we instead use a *weighted* "leave-one-out" philosophy. More specifically, consider the case $0 < \omega < 1$. Then define the matrix

$$\boldsymbol{F}_j = \mathrm{diag}(1, 1, ...1, \sqrt{1-\omega}, 1, ...1),$$

where $\sqrt{1-\omega}$ is the j^{th} diagonal entry of $\boldsymbol{F}_j$. By using the W-GCV method, we seek a solution to the minimization problem,

$$\min_{\boldsymbol{x}} ||\boldsymbol{F}_j(\boldsymbol{b} - \boldsymbol{Ax})||_2^2 + \lambda^2 ||\boldsymbol{x}||_2^2.$$

In this problem, the jth observation is still present but has been down-weighted by the factor $\sqrt{1-\omega}$; thus it is completely left out when $\omega = 1$. A derivation of the W-GCV method follows immediately from the derivation of the GCV method found in [11].

By introducing a new parameter in the trace term of the GCV function, we not only introduce a new weighted prediction approach, but also change the interpretation of the function we are minimizing. We consider the special case of Tikhonov regularization and look at how the GCV function is altered algebraically and graphically with the new parameter. Using the SVD expansion of $\boldsymbol{A}$, it can be shown that the trace term in the standard GCV function is given by

$$\mathrm{trace}(\boldsymbol{I} - \boldsymbol{AA}_\lambda^\dagger) = \sum_{i=1}^{n} \frac{\lambda^2}{\sigma_i^2 + \lambda^2} + (m-n).$$

In contrast, the trace term for the W-GCV function is given by:

$$\begin{aligned}\mathrm{trace}(\boldsymbol{I} - \omega\boldsymbol{AA}_\lambda^\dagger) &= \sum_{i=1}^{n} \frac{(1-\omega)\sigma_i^2 + \lambda^2}{\sigma_i^2 + \lambda^2} + (m-n) \\ &= \sum_{i=1}^{n} (1-\omega)\phi_i + \sum_{i=1}^{n} \frac{\lambda^2}{\sigma_i^2 + \lambda^2} + (m-n).\end{aligned}$$

Thus, if $\omega < 1$ then we are adding a multiple of the sum of the filter factors to the original trace term, and if $\omega > 1$ we are subtracting a multiple. The graph of the GCV function also undergoes changes as ω is changed from 1. The denominator becomes zero for some value of $\omega > 1$, so the W-GCV function has a pole. Fortunately, in our case, $0 < \omega \leq 1$. Note that larger values of ω result in larger computed regularization parameters, and smaller values of ω result in smaller values of λ.

ETNA
Kent State University
etna@mcs.kent.edu

5.3. W-GCV for the bidiagonal system. In the previous subsection we discussed W-GCV in the context of Tikhonov regularization on the original (full) system of equations involving $\boldsymbol{A}$ and $\boldsymbol{b}$. This allowed us to provide a general description, but our aim is to apply W-GCV to choosing regularization parameters for the projected problem, (3.3). In this specific case, the W-GCV function has the form

$$G_{B_k,\,\beta e_1}(\omega,\lambda) = \frac{k\|(\boldsymbol{I}-\boldsymbol{B}_k\boldsymbol{B}_{k,\lambda}^{\dagger})\beta\boldsymbol{e}_1\|_2^2}{\left(\mathrm{trace}(\boldsymbol{I}-\omega\boldsymbol{B}_k\boldsymbol{B}_{k,\lambda}^{\dagger})\right)^2}$$

$$= \frac{k\beta^2\left(\sum_{i=1}^{k}\left(\frac{\lambda^2}{\delta_i^2+\lambda^2}\left[\boldsymbol{P}_k^T\boldsymbol{e}_1\right]_i\right)^2+\left(\left[\boldsymbol{P}_k^T\boldsymbol{e}_1\right]_{k+1}\right)^2\right)}{\left(1+\sum_{i=1}^{k}\frac{(1-\omega)\delta_i^2+\lambda^2}{\delta_i^2+\lambda^2}\right)^2},$$

where, using the notation introduced in (3.5), $\boldsymbol{P}_k$ is an orthogonal matrix containing the left singular vectors of $\boldsymbol{B}_k$, δ_i is the ith largest singular value of $\boldsymbol{B}_k$, and $\boldsymbol{W}_k^T\boldsymbol{b} = \beta\boldsymbol{e}_1$ with $\beta = \|\boldsymbol{b}\|$. Note that this reduces to the expression in (3.5) when $\omega = 1$.

5.4. Choosing ω. For many ill-posed problems, a good value of ω is crucial for the success of Lanczos-hybrid methods. In this section we consider how different values of ω may affect convergence behavior and present an adaptive approach for finding a good value for ω.

Consider the test problem *Heat*, whose convergence graph with Tikhonov regularization and the standard GCV method is given in the bottom right corner of Figure 4.1. To illustrate the effects of using the W-GCV function with Lanczos-hybrid methods, we present results using the same fixed value of ω at all steps of the iteration. The results are shown in Figure 5.1.

For this particular example, it is evident that $\omega = 0.2$ is a good value for the new parameter. However, finding a good ω in this way is not possible since the true solution is generally not available. Hence, we introduce an automated, adaptive approach that in our experience produces adequate results.

Recall from section 3 that at each iteration of the Lanczos-hybrid method, we solve the projected LS problem (3.3) using Tikhonov regularization. Since the early iterations of LBD do not capture the ill-conditioning of the problem, we expect that little or no regularization is needed to solve the projected LS problem. Let $\lambda_{k,opt}$ denote the optimal regularization parameter at the k^{th} iteration. Then, we can assume that for small k, $\lambda_{k,opt}$ should satisfy

$$0 \le \lambda_{k,opt} \le \sigma_{min}\left(\boldsymbol{B}_k\right),$$

where σ_{min} denotes the smallest singular value of the matrix. If at iteration k, we assume that we know $\lambda_{k,opt}$, then we can find ω by minimizing the GCV function with respect to ω. That is, solving

$$\frac{\partial}{\partial\lambda}\left(G_{B_k,\,\beta e_1}(\omega,\lambda)\right)\Big|_{\lambda=\lambda_{k,opt}} = 0.$$

Since we do not know $\lambda_{k,opt}$, we instead find $\hat{\omega}_k$ corresponding to $\lambda_{k,opt} = \sigma_{min}\left(\boldsymbol{B}_k\right)$. In later iterations, this approach fails because $\sigma_{min}\left(\boldsymbol{B}_k\right)$ becomes nearly zero due to ill-conditioning. For these iterations, a better approach is to adaptively take

$$\omega_k = \text{mean}\left\{\hat{\omega}_1, \hat{\omega}_2, \ldots, \hat{\omega}_k\right\}.$$

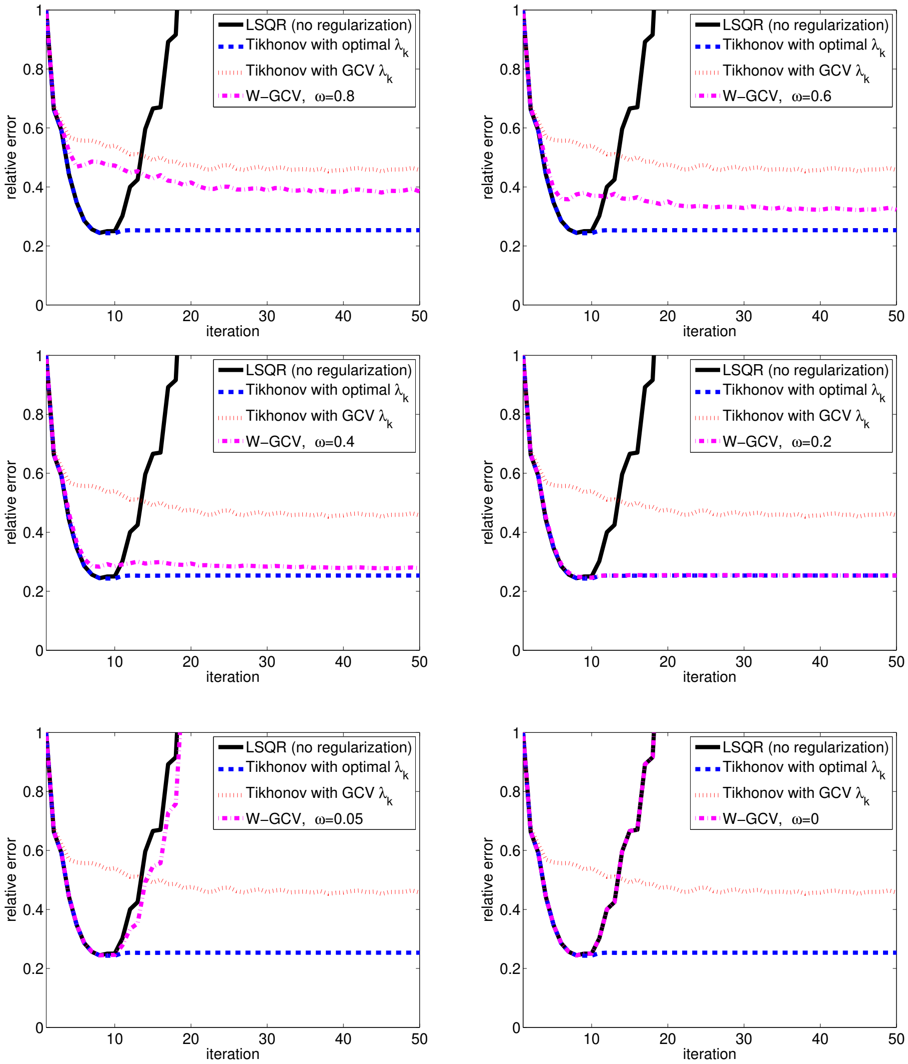

FIG. 5.1. *This figure shows plots of the relative errors,* $\|\mathbf{x}_k - \mathbf{x}_{\text{true}}\|_2/\|\mathbf{x}_{\text{true}}\|_2$, *for LSQR and the Lanczos-hybrid method for the Heat example from Regtools. The various plots show how the convergence behavior changes when regularization parameters are chosen using the W-GCV method with different values of* ω. *Note that* $\omega = 1$ *is equivalent to using standard GCV, and* $\omega = 0$ *is equivalent to using no regularization.*

By averaging the previously computed ω values, we are essentially using the earlier well-conditioned components of our problem to help stabilize the harmful effects of the smaller singular values. There are two disadvantages to this approach. First, it over-smooths the solutions at early iterations, since it uses a rather large value of λ for a well-conditioned problem. Since these solutions are discarded, this is not a significant difficulty. Second, it undersmooths values for large k, so semiconvergence will eventually reappear. However, in practice we will also be using a method like GCV to choose a stopping iteration, so k will not be allowed to grow too large; this is discussed in the following subsection.

5.5. Stopping criteria for LBD. The next practical issue to consider is an approach to determine an appropriate point at which to stop the iteration. Björck [1] suggested using GCV for this purpose, when TSVD is used to solve the projected problem. However, Björck, Grimme and Van Dooren [2] showed that modifications of the algorithm were needed to make the approach effective for practical problems. Specifically, they proposed a fairly complicated scheme based on implicitly restarting the iterations.

In this section we describe a similar approach for Tikhonov regularization, but we do not need implicit restarts. We begin by defining the computed solution at each iteration of the Lanczos hybrid method as

$$\boldsymbol{x}_k = \boldsymbol{Y}_k \boldsymbol{f}_{\lambda_k} = \boldsymbol{Y}_k(\boldsymbol{B}_k^T\boldsymbol{B}_k + \lambda_k^2\boldsymbol{I})^{-1}\boldsymbol{B}_k^T\boldsymbol{W}_k^T\boldsymbol{b} \equiv \boldsymbol{A}_k^{\dagger}\boldsymbol{b}\,. \tag{5.1}$$

Using the basic idea of GCV, we would like to determine a stopping iteration, k, that minimizes

$$\widehat{G}(k) = \frac{n\|(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}_k^{\dagger})\boldsymbol{b}\|_2^2}{\left(\mathrm{trace}(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}_k^{\dagger})\right)^2}\,. \tag{5.2}$$

Using (5.1) and (3.2), the numerator of equation (5.2) can be written as

$$n\|(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}_k^{\dagger})\boldsymbol{b}\|_2^2 = n\|(\boldsymbol{I} - \boldsymbol{B}_k(\boldsymbol{B}_k^T\boldsymbol{B}_k + \lambda_k^2\boldsymbol{I})^{-1}\boldsymbol{B}_k^T)\beta\boldsymbol{e}_1\|_2^2\,.$$

If we now replace $\boldsymbol{B}_k$ with its SVD (3.4), we obtain

$$\begin{aligned} n\|(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}_k^{\dagger})\boldsymbol{b}\|_2^2 &= n\beta^2 \left\| \begin{bmatrix} \frac{\lambda_k^2}{\delta_1^2+\lambda_k^2} & & & \\ & \ddots & & \\ & & \frac{\lambda_k^2}{\delta_k^2+\lambda_k^2} & \\ & & & 1 \end{bmatrix} \boldsymbol{P}_k^T\boldsymbol{e}_1 \right\|_2^2 \\ &= n\beta^2 \left(\sum_{i=1}^{k} \left(\frac{\lambda_k^2}{\delta_i^2 + \lambda_k^2} \left[\boldsymbol{P}_k^T\boldsymbol{e}_1\right]_i \right)^2 + \left(\left[\boldsymbol{P}_k^T\boldsymbol{e}_1\right]_{k+1} \right)^2 \right). \end{aligned} \tag{5.3}$$

Similarly, the denominator of equation (5.2) can be written as

$$\left(\mathrm{trace}\left(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}_k^{\dagger}\right)\right)^2 = \left((m-k) + \sum_{i=1}^{k} \frac{\lambda_k^2}{\delta_i^2 + \lambda_k^2}\right)^2. \tag{5.4}$$

Thus, combining (5.3) and (5.4), equation (5.2) can be written as

$$\widehat{G}(k) = \frac{n\beta^2 \left(\sum_{i=1}^{k} \left(\frac{\lambda_k^2}{\delta_i^2 + \lambda_k^2} \left[\boldsymbol{P}_k^T\boldsymbol{e}_1\right]_i \right)^2 + \left(\left[\boldsymbol{P}_k^T\boldsymbol{e}_1\right]_{k+1} \right)^2 \right)}{\left((m-k) + \sum_{i=1}^{k} \frac{\lambda_k^2}{\delta_i^2 + \lambda_k^2}\right)^2}\,. \tag{5.5}$$

This is the form of $\widehat{G}(k)$ that we use to determine a stopping iteration in our implementations. The numerator is n/k times the numerator in (3.5) for $G_{A,b}(\lambda_k)$, and the denominator differs only in its first term.

ETNA
Kent State University
etna@mcs.kent.edu

In the ideal situation where the convergence behavior of the Lanczos-hybrid method is perfectly stabilized, we expect λ_k to converge to a fixed value corresponding to an appropriate regularization parameter for the original problem (1.2). In this case the values of $\widehat{G}(k)$ converge to a fixed value. Therefore, we choose to terminate the iterations when these values change very little,

$$\left| \frac{\widehat{G}(k+1) - \widehat{G}(k)}{\widehat{G}(1)} \right| < \text{tol}\,,$$

for some prescribed tolerance.

However, as remarked in the previous subsection, it may be impossible to completely stabilize the iterations for realistic problems, resulting in slight semiconvergent behavior of the iterations. In this case, the GCV values $\widehat{G}(k)$ will begin to increase. Thus, we implement a second stopping criteria to stop at iteration k_0 satisfying

$$k_0 = \operatorname*{argmin}_k \widehat{G}(k)\,.$$

In the next section we present some numerical experiments with an implementation of the Lanczos-hybrid method that uses this approach.

6. Numerical results.

We now illustrate the effectiveness of using the W-GCV method in Lanczos-hybrid methods with Tikhonov regularization.

6.1. Results on various test problems.

We implement the adaptive method presented in section 5.4 for choosing ω and provide numerical results for each of the test problems. The resulting convergence curves are displayed in Figure 6.1.

In all of the test problems, choosing ω adaptively provides nearly optimal convergence behavior. The results for the *Phillips* and *Shaw* problems are excellent with the adaptive W-GCV approach. The *Satellite, Baart* and *Heat* examples exhibit a slowed convergence compared to Tikhonov with the optimal regularization parameter but achieve much better results than with the standard GCV. This slowed convergence is due to the fact that at the early iterations the projected problem is well conditioned and W-GCV produces a solution that is too smooth. At later iterations, when more small singular value information is captured in the bidiagonalization process, better ω, and hence λ, parameters are found, and the W-GCV parameter choice is close to optimal. In addition, W-GCV avoids the early stagnation behavior that GCV exhibits.

It should be noted that when LBD takes many iterations, preconditioning could be used to accelerate convergence. It is interesting to note that the *Deriv2* example converges, but eventually exhibits small signs of semiconvergent behavior. Nevertheless, the results are still better than the standard GCV, and, moreover, if combined with the stopping criteria described in the previous section, the results are quite good. To illustrate, in Table 6.1 we report the iteration at which our code detected a minimum of $\widehat{G}(k)$.

TABLE 6.1
Results of using $\widehat{G}(k)$ to determine a stopping iteration. The numbers reported in this table are the iteration index at which our Lanczos-hybrid code detected a minimum of $\widehat{G}(k)$.

Problem	*Satellite*	*Phillips*	*Shaw*	*Deriv2*	*Baart*	*Heat*
Stopping Iteration	197	18	23	20	9	21

Comparing the results in Table 6.1 with the convergence history plots shown in Figure 6.1, we see that our approach to choosing a stopping iteration is very effective. Although

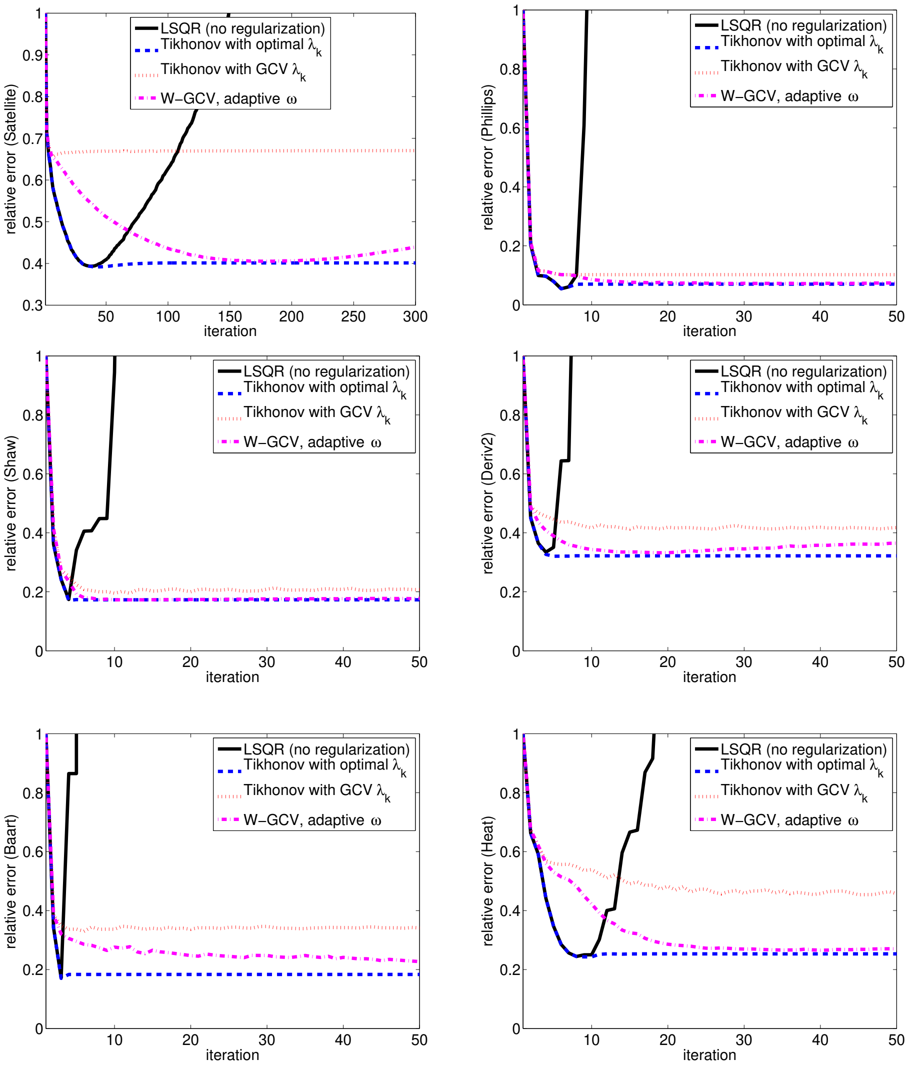

FIG. 6.1. *These plots show the relative error,* $\|\mathbf{x}_k - \mathbf{x}_{\text{true}}\|_2/\|\mathbf{x}_{\text{true}}\|_2$, *at each iteration of LSQR and the Lanczos-hybrid method. Upper left: Satellite. Upper right: Regtools-Phillips. Middle left: Regtools-Shaw. Middle right: Regtools-Deriv2. Bottom left: Regtools-Baart. Bottom right: Regtools-Heat. The standard GCV method chooses regularization parameters that are too large at each iteration, which cause poor convergence behavior. However, the W-GCV method, with our adaptive approach to choose* ω *produces near optimal convergence behavior.*

the scheme does not perform as well on the *Baart* example, the results are still quite good considering the difficulty of this problem. (Observe that with no regularization, semiconvergence happens very quickly, and we should therefore expect difficulties in stabilizing the iterations.) These results show that our W-GCV method performs better than standard GCV, and that we are able to determine an appropriate stopping iteration on a wide class of problems.

We also remark that the *Satellite* example is a much larger problem than the other examples, and so more iterations are needed. However, the Lanczos-hybrid method can easily incorporate standard preconditioning techniques to accelerate convergence. For the *Satellite*

ETNA
Kent State University
etna@mcs.kent.edu

image deblurring example, we used a Kronecker product based preconditioner [20, 21, 27] implemented in RestoreTools [26]. In this case, the Lanczos-hybrid method, with W-GCV, detects a minimum of $\widehat{G}(k)$ in only 54 iterations. The corresponding solution has relative error 0.4001, which is actually slightly lower than the relative error 0.4061 achieved at iteration 197 when using no preconditioning.

6.2. Effect of noise on ω. We now consider how the choice of ω depends on the amount of noise in the data. In particular, we report on numerical results for the test problems described in section 4.1 with three different noise levels:

$$\frac{\|\varepsilon\|_2}{\|\boldsymbol{A}\boldsymbol{x}_{\text{true}}\|_2} = 0.1, 0.01, \text{ and } 0.001.$$

Thus these problems have 10%, 1% and 0.1% noise levels respectively. Some of the results reported in previous sections for 10% noise are repeated here for comparison purposes.

Recall that because standard GCV computes regularization parameters that are too large, we should choose $0 < \omega \leq 1$ in W-GCV. Generally we observe that the over-smoothing caused by standard GCV is more pronounced for larger noise levels. Therefore large noise levels typically need smaller values of ω, while small noise levels need larger values of ω. Our next experiments were designed to see how far the "optimal" value of ω differs from the GCV value $\omega = 1$. The results are shown in Table 6.2, which displays ω values that allow W-GCV to compute near-optimal regularization parameters at each iteration of the Lanczos-hybrid method. For example, in Figure 5.1 we see that for 10% noise, $\omega = 0.2$ produces near optimal convergence behavior for the *Heat* problem, and thus this value appears in the first row, last column of table.

TABLE 6.2
Values of ω (found experimentally) that produce optimal convergence behavior of the Lanczos-hybrid method for different noise levels. Figure 6.2 shows how these values perform on the Baart and Heat examples.

	Satellite	*Phillips*	*Shaw*	*Deriv2*	*Baart*	*Heat*
Noise Level	ω_{opt}	ω_{opt}	ω_{opt}	ω_{opt}	ω_{opt}	ω_{opt}
10 %	0.40	0.20	0.05	0.10	0.01	0.20
1%	0.50	0.40	0.05	0.20	0.05	0.40
0.1%	0.80	0.50	0.10	0.60	0.10	0.80

The results reported in Table 6.2 were found experimentally. We see clearly from this table that optimal values of ω depend on the noise level (increasing with decreasing noise level), as well as with the problem. However, more work is needed to better understand these relationships.

Figure 6.2 shows how our adaptive approach to choosing ω compares to the optimal values on two of the test problems (*Baart* and *Heat*) and for the various noise levels. These two test problems are representative of the convergence behavior we observe with the other test problems. We see that if a good choice of ω can be found, W-GCV is very effective (much more so than GCV) at choosing regularization parameters, and thus at stabilizing the convergence behavior, especially for high noise levels. Moreover, although we do not yet have a scheme that chooses the optimal value of ω, these results show that our adaptive approach produces good results on a wide class of problems, and for various noise levels.

7. Concluding remarks. In this paper, we have considered using a weighted-GCV method in Lanczos-hybrid methods for solving large scale ill-posed problems. The W-GCV

ETNA
Kent State University
etna@mcs.kent.edu

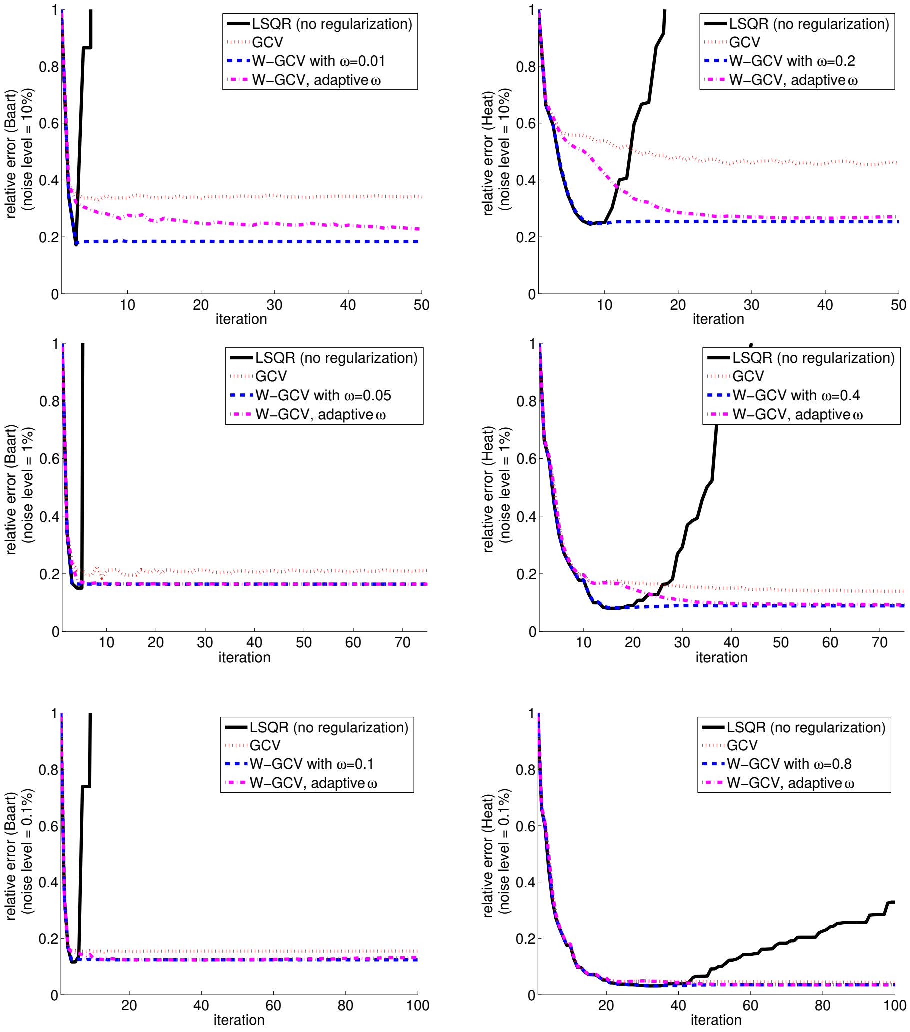

FIG. 6.2. *These plots show the relative error,* $\|\mathbf{x}_k - \mathbf{x}_{\text{true}}\|_2/\|\mathbf{x}_{\text{true}}\|_2$, *at each iteration of LSQR and the Lanczos-hybrid method. The plots on the left correspond to the Baart example, for three different noise levels. The plots on the right correspond to the Heat example. Optimal choices of* ω *(found experimentally) produce optimal convergence behavior, and our adaptive approach to choose* ω *produces near optimal convergence behavior. It can be observed that standard GCV is ineffective for moderate to high levels of noise.*

method requires choosing yet another parameter, so we proposed and implemented an adaptive, automatic approach for choosing this parameter. We demonstrated through a variety of test problems that our approach was effective in stabilizing semiconvergence behavior.

The MATLAB implementations used to generate the results presented in this paper can be obtained from `http://www.mathcs.emory.edu/~nagy/WGCV`.

Several open questions remain. With the ability to obtain near optimal solutions, Lanczos-hybrid methods should have a significant impact on many applications. Recently, Kilmer, Hansen and Español [22] suggested a projection-based algorithm that can be implemented for more general regularization operators. We can treat this iterative method as a hybrid method and apply W-GCV. In addition, we would like to see how well W-GCV works in combination with truncated SVD and Lanczos-hybrid methods. Finally, work remains to be done on alternative ways to determine the new parameter in the W-GCV method.

Acknowledgements. We are grateful to Kevin J. Coakley and Bert W. Rust for helpful comments on the manuscript.

The views expressed here are those of the authors alone, not necessarily those of NIST or NSF. Certain commercial products are identified in order to specify adequately experimental procedures. In no case does such identification imply recommendation or endorsement by NIST, nor does it imply that the items identified are necessarily the best available for the purpose.

REFERENCES

[1] Å. BJÖRCK, *A bidiagonalization algorithm for solving large and sparse ill-posed systems of linear equations*, BIT, 28 (1988), pp. 659–670.
[2] Å. BJÖRCK, E. GRIMME, AND P. VAN DOOREN, *An implicit shift bidiagonalization algorithm for ill-posed systems of linear equations*, BIT, 34 (1994), pp. 510–534.
[3] D. CALVETTI, G. H. GOLUB, AND L. REICHEL, *Estimation of the L-curve via Lanczos bidiagonalization*, BIT, 39 (1999), pp. 603–619.
[4] D. CALVETTI AND L. REICHEL, *Tikhonov regularization of large scale problems*, BIT, 43 (2003), pp. 263–283.
[5] ———, *Tikhonov regularization with a solution constraint*, SIAM J. Sci. Comput., 26 (2004), pp. 224–239.
[6] D. CUMMINS, T. FILLOON, AND D. NYCHKA, *Confidence intervlas for non-parametric curve estimates: Toward more uniform pointwise coverage*, J. Amer. Statist. Assoc., 96 (2001), pp. 233–246.
[7] L. ELDÉN, *Algorithms for the regularization of ill-conditioned least squares problems*, BIT, 17 (1977), pp. 134–145.
[8] H. W. ENGL, M. HANKE, AND A. NEUBAUER, *Regularization of Inverse Problems*, Kluwer Academic Publishers, Dordrecht, 2000.
[9] J. FRIEDMAN AND B. SILVERMAN, *Flexible parsimonious smoothing and additive modeling*, Technometrics, 31 (1989), pp. 3–21.
[10] A. FROMMER AND P. MAASS, *Fast CG-based methods for Tikhonov-Phillips regularization*, SIAM J. Sci. Comput., 20 (1999), pp. 1831–1850.
[11] G. H. GOLUB, M. HEATH, AND G. WAHBA, *Generalized cross-validation as a method for choosing a good ridge parameter*, Technometrics, 21 (1979), pp. 215–223.
[12] G. H. GOLUB, F. T. LUK, AND M. L. OVERTON, *A block Lanczos method for computing the singular values and corresponding singular vectors of a matrix*, ACM Trans. Math. Software, 7 (1981), pp. 149–169.
[13] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd edition, Johns Hopkins University Press, Baltimore, MD, 1996.
[14] G. H. GOLUB AND U. VON MATT, *Quadratically constrained least squares and quadratic problems*, Numer. Math., 59 (1991), pp. 561–580.
[15] C. W. GROETSCH, *The Theory of Tikhonov Regularization for Fredholm Integral Equations of the First Kind*, Pitman, Boston, 1984.
[16] M. HANKE, *On Lanczos based methods for the regularization of discrete ill-posed problems*, BIT, 41 (2001), pp. 1008–1018.
[17] P. C. HANSEN, *Regularization tools: A MATLAB package for analysis and solution of discrete ill-posed problems*, Numer. Algorithms, 6 (1994), pp. 1–35.
[18] ———, *Rank-Deficient and Discrete Ill-Posed Problems*, SIAM, Philadelphia, PA, 1997.
[19] P. C. HANSEN, J. G. NAGY, AND D. P. O'LEARY, *Deblurring Images: Matrices, Spectra and Filtering*, SIAM, Philadelphia, PA, 2006.
[20] J. KAMM AND J. G. NAGY, *Kronecker product and SVD approximations in image restoration*, Linear Algebra Appl., 284 (1998), pp. 177–192.
[21] ———, *Optimal Kronecker product approximation of block Toeplitz matrices*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 155–172.
[22] M. E. KILMER, P. C. HANSEN, AND M. I. ESPAÑOL, *A projection-based approach to general-form Tikhonov regularization*, SIAM J. Sci. Comput., 29 (2007), pp. 315–330.
[23] M. E. KILMER AND D. P. O'LEARY, *Choosing regularization parameters in iterative methods for ill-posed problems*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 1204–1221.

ETNA
Kent State University
etna@mcs.kent.edu

[24] Y. KIM AND C. GU, *Smoothing spline Gaussian regression: More scalable computation via efficient approximation*, J. Roy. Statist. Soc. B, 66 (2004), pp. 337–356.
[25] R. M. LARSEN, *Lanczos Bidiagonalization with Partial Reorthogonalization*, PhD thesis, Deptartment of Computer Science, University of Aarhus, Denmark, 1998.
[26] J. NAGY, K. PALMER, AND L. PERRONE, *Iterative methods for image deblurring: A MATLAB object oriented approach*, Numer. Algorithms, 36 (2004), pp. 73–93.
[27] J. G. NAGY, M. K. NG, AND L. PERRONE, *Kronecker product approximation for image restoration with reflexive boundary conditions*, SIAM J. Matrix Anal. Appl., 25 (2004), pp. 829–841.
[28] D. NYCHKA, B. BAILEY, S. ELLNER, P. HAALAND, AND M. O'CONNEL, *FUNFITS: Data analysis and statistical tools for estimating functions*, in Case Studies in Environmental Statistics, Springer-Verlag, New York, 1998, pp. 159–179.
[29] D. P. O'LEARY AND J. A. SIMMONS, *A bidiagonalization-regularization procedure for large scale discretizations of ill-posed problems*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 474–489.
[30] C. C. PAIGE AND M. A. SAUNDERS, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software, 8 (1982), pp. 43–71.
[31] R. VIO, P. MA, W. ZHONG, J. NAGY, L. TENORIO, AND W. WAMSTEKER, *Estimation of regularization parameters in multiple-image deblurring*, Astronom. and Astrophys., 423 (2004), pp. 1179–1186.
[32] C. R. VOGEL, *Computational Methods for Inverse Problems*, SIAM, Philadelphia, PA, 2002.