SIAM J. Matrix Anal. Appl.
Vol. 22, No. 4, pp. 1204–1221

CHOOSING REGULARIZATION PARAMETERS IN ITERATIVE METHODS FOR ILL-POSED PROBLEMS[*]

MISHA E. KILMER[†] AND DIANNE P. O'LEARY[‡]

Abstract. Numerical solution of ill-posed problems is often accomplished by discretization (projection onto a finite dimensional subspace) followed by regularization. If the discrete problem has high dimension, though, typically we compute an approximate solution by projecting the discrete problem onto an even smaller dimensional space, via iterative methods based on Krylov subspaces. In this work we present a common framework for efficient algorithms that regularize after this second projection rather than before it. We show that determining regularization parameters based on the final projected problem rather than on the original discretization has firmer justification and often involves less computational expense. We prove some results on the approximate equivalence of this approach to other forms of regularization, and we present numerical examples.

Key words. ill-posed problems, regularization, discrepancy principle, iterative methods, L-curve, Tikhonov, truncated singular value decomposition, projection, Krylov subspace

AMS subject classifications. 65F10, 65F22

PII. S0895479899345960

1. Introduction.

Linear, discrete ill-posed problems of the form

$$\min_x \|Ax - b\|_2 \tag{1.1}$$

arise, for example, from the discretization of first-kind Fredholm integral equations and occur in a variety of applications. We shall assume that the full-rank matrix A is $m \times n$ with $m \geq n$. In discrete ill-posed problems, A is ill-conditioned and there is often no gap in the singular value spectrum. Typically, the right-hand side b contains noise due to measurement and/or approximation error. This noise, in combination with the ill-conditioning of A, means that the exact solution of (1.1) has little relationship to the noise-free solution and is worthless.

Instead, we use a *regularization* method to determine a solution that approximates the noise-free solution. We replace the original operator by a better conditioned but related one in order to diminish the effects of noise in the data. Sometimes this regularized problem is too large to solve exactly. In that case, we typically project the problem onto an even smaller dimensional space, perhaps via iterative methods based on Krylov subspaces. Sometimes this projection provides enough regularization to produce a good approximate solution, but often (see, for example, [28, 15]) additional regularization is needed.

A fundamental decision to be made in such cases is whether to regularize before or after the projection. *One subtle issue is that the regularization parameter that is optimal for the discretized problem may not be optimal for the lower-dimensional problem actually solved by the iteration*, and this leads to the research discussed in this paper.

[*] Received by the editors December 3, 1999; accepted for publication (in revised form) by P. C. Hansen November 13, 2000; published electronically April 6, 2001. This work was supported by the National Science Foundation under grants CCR 95-03126 and CCR 97-32022 and by the Army Research Office, MURI grant DAAG55-97-1-0013.

http://www.siam.org/journals/simax/22-4/34596.html

[†] Department of Mathematics, Tufts University, Medford, MA 02155 (mkilme01@tufts.edu).

[‡] Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742 (oleary@cs.umd.edu).

At first glance, there can appear to be a lot of work associated with the selection of a good regularization parameter, and many algorithms proposed in the literature are needlessly complicated, repeating a Krylov iteration multiple times. By regularizing after projection by the iterative method, so that we are regularizing the lower dimensional problem that is actually being solved, this difficulty vanishes.

The purpose of this paper is to present a common framework for parameter selection techniques applied to the problem resulting from iterative methods such as Krylov subspace techniques. We show that by determining regularization parameters based on the final projected problem rather than on the original discretization, we can better approximate the optimal parameter and reduce the cost of solution.

Our paper is organized as follows. In section 2 we survey some methods for choosing the corresponding regularization parameters. In section 3, we show how any standard parameter selection technique for the original problem can be applied instead to a projected problem obtained from an iterative method, greatly reducing the cost without much degradation in the solution. We give experimental results in section 4 and conclusions in section 5.

In the following we shall assume that $b = b_{true} + e$, where b_{true} denotes the unperturbed data vector and e denotes zero-mean white noise. We will also assume that b_{true} satisfies the discrete Picard condition; that is, the spectral coefficients of b_{true} decay faster, on average, than the singular values.

Let $\hat{U}\Sigma\hat{V}^*$ denote the singular value decomposition (SVD) of A, where the columns of $\hat{U}$ and $\hat{V}$ are the singular vectors, and the singular values are ordered as $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$. Then the solution (1.1) is given by

$$x = \sum_{i=1}^{n} \frac{\hat{u}_i^* b}{\sigma_i} \hat{v}_i = \sum_{i=1}^{n} \left(\frac{\hat{u}_i^* b_{true}}{\sigma_i} + \frac{\hat{u}_i^* e}{\sigma_i} \right) \hat{v}_i. \tag{1.2}$$

As a consequence of the white noise assumption, $|\hat{u}_i^* e|$ is roughly constant for all i, while the discrete Picard condition guarantees that $|\hat{u}_i^* b_{true}|$ decreases with i faster than σ_i does. The matrix A is ill-conditioned, so small singular values magnify the corresponding coefficients $\hat{u}_i^* e$ in the second sum, and it is this large contribution of noise that renders the exact solution x defined in (1.2) worthless. The following four classes of regularization methods try in different ways to lessen the contribution of noise. For further information on these methods, see, for example, [19, 15].

In **Tikhonov regularization**, (1.1) is replaced by

$$\min_x \|Ax - b\|_2^2 + \lambda^2 \|Lx\|_2^2, \tag{1.3}$$

where λ is a positive scalar regularization parameter, and we choose L to be the identity matrix I. Solving (1.3) is equivalent to solving

$$(A^*A + \lambda^2 I)x_\lambda = A^*b. \tag{1.4}$$

In analogy with (1.2) we have

$$x_\lambda = \sum_{i=1}^{n} \left(\frac{\sigma_i \, \hat{u}_i^* b_{true}}{\sigma_i^2 + \lambda^2} + \frac{\sigma_i \, \hat{u}_i^* e}{\sigma_i^2 + \lambda^2} \right) \hat{v}_i. \tag{1.5}$$

In **truncated SVD** we compute the regularized solution by truncating the expansion in (1.2) as

$$x_\ell = \sum_{i=1}^{\ell} \frac{\hat{u}_i^* b}{\sigma_i} \hat{v}_i. \tag{1.6}$$

Here the regularization parameter is ℓ, the number of terms retained in the sum. Rust [33] introduced a related truncation strategy, including in the sum (1.2) only those terms corresponding to a spectral coefficient $\hat{u}_i^* b$ whose magnitude is greater than or equal to some tolerance ρ, which can be regarded as the regularization parameter.

Solving (1.4) or (1.6) can be impractical if n is large, but fortunately, regularization can be achieved through **projection** onto a k-dimensional subspace; see, for example, [9]. The truncated SVD (TSVD) is one example, but projection is often achieved through the use of iterative methods such as conjugate gradients, GMRES, QMR, and other Krylov subspace methods [28, 1]. Krylov subspace algorithms tend to produce, at early iterations, solutions that resemble x_{true} more than later iterates. Therefore, the choice of the regularization parameter k, the stopping point for the iteration and the dimension of the subspace, is very important.

Another important family of regularization methods, termed **hybrid methods** [19, 15], was introduced by O'Leary and Simmons [28]. These methods combine a projection method with a direct regularization method such as TSVD or Tikhonov regularization. Since the dimension k is usually small relative to n, regularization of the restricted problem is much less expensive, but the end results can be very similar to those achieved by applying the same direct regularization technique to the original problem; see section 3.5.

2. Existing parameter selection methods.

In this section, we discuss three parameter selection techniques that have been proposed in the literature. They differ in the amount of a priori information required as well as in the decision criteria.

The discrepancy principle [26] says that if δ is the expected value of $\|e\|_2$, then the regularization parameter should be chosen so that the norm of the residual corresponding to the regularized solution x_{reg} is $\tau\delta$; that is,

$$\|Ax_{reg} - b\|_2 = \tau\delta, \tag{2.1}$$

where $\tau > 1$ is some predetermined real number. Note that as $\delta \to 0$, $x_{reg} \to x_{true}$. Other methods based on knowledge of the variance are given, for example, in [3, 13, 7].

Generalized cross-validation (GCV) [11] does not depend on a priori knowledge about the noise variance. We find the parameter λ that minimizes the GCV functional

$$G(\lambda) = \frac{\|(I - AA_\lambda^\sharp)b\|_2^2}{(\text{trace}(I - AA_\lambda^\sharp))^2}, \tag{2.2}$$

where $A_\lambda^\sharp$ denotes the matrix that maps the right-hand side b onto the regularized solution x_λ. In Tikhonov regularization, for example, $A_\lambda^\sharp$ is $(A^*A + \lambda^2 I)^{-1}A^*$.

The L-curve, the plot of the norm of the regularized solution versus the corresponding residual norm for each of a set of regularization parameter values, was introduced by Lawson and popularized by Hansen [17, 25]. Intuitively, the best regularization parameter should lie on the corner of the L-curve, since for values higher than this, the residual increases without reducing the norm of the solution much, while for values smaller than this, the norm of the solution increases rapidly without much decrease in residual. In practice, only a few points on the L-curve are computed and the corner is located by estimating the point of maximum curvature [20].

The appropriate choice of regularization parameter—especially for projection algorithms—is a difficult problem, and each method has severe flaws. The discrepancy principle is convergent as the noise goes to zero, but it relies on information that

Table 2.1

Summary of additional flops needed to compute the regularization parameter for each of four regularization methods with various parameter selection techniques. Notation: q is the cost of multiplication of a vector by A; p is the number of discrete parameters that must be tried; k is the dimension of the projection; m and n are problem dimensions.

	Basic cost	Added cost		
		Disc.	GCV	L-curve
Tikhonov	$O(mn^2)$	$O(p(m+n))$	$O(p(n+m))$	$\mathrm{O}(p(m+n))$
TSVD	$O(mn^2)$	$O(m)$	$O(m)$	$O(m+n)$
Rust's TSVD	$O(mn^2)$	$O(m\log m)$	$O(m\log m)$	$O(m\log m)$
Projection	$O(qk)$	0	$O(q)$	$O(q)$

is often unavailable or erroneous. Even with a correct estimate of the variance, the solutions tend to be oversmoothed [21, p. 96]. (See also the discussion in section 6.1 of [17].) One noted difficulty with GCV is that G can have a very flat minimum, making it difficult to determine the optimal λ numerically [37]. The L-curve is usually more tractable numerically, but its limiting properties are nonideal. The solution estimates fail to converge to the true solution as $n \to \infty$ [38] or as the error norm goes to zero [8]. All methods that assume no knowledge of the error norm— including GCV—have this latter property [8].

For further discussion and references about parameter choice methods, see [7, 19]. The cost of these methods is tabulated in Table 2.1.

2.1. Previous work on parameter choice for hybrid methods.

At first glance, it appears that for Tikhonov regularization, multiple systems of the form (1.4) must be solved in order to evaluate candidate values of λ for the discrepancy principle or the L-curve.

Chan and Ng [5] note that the systems involve matrices $C(\lambda) = A^*A + \lambda I$, which they solve using a Galerkin projection method on a sequence of "seed" systems. Although economical in storage, this is unnecessarily expensive in time because they do not exploit the fact that for each fixed k, the Krylov subspace $\mathcal{K}_k(A^*b, C(\lambda))$ is the same for all values of λ.

Frommer and Maass [10] propose two algorithms for approximating the λ that satisfies the discrepancy principle (2.1). The first is a "truncated conjugate gradient (CG)" approach, solving k systems of the form (1.4), truncating the iterative process early for large λ, and using previous solutions as starting guesses for later problems. Like Chan and Ng, this algorithm does not exploit the redundant Krylov subspaces. In the second method, however, they update the CG iterates for all k systems simultaneously, stopping their "shifted CG" algorithm when $\|Ax_\lambda - b\|_2 \le \tau\delta$ for one of their λ values. The methods we propose in section 3 will usually require less work than the shifted CG algorithm because of less overhead.

Calvetti, Golub, and Reichel [4] use upper and lower bounds on the L-curve, generated by the matrices $C(\lambda)$ using a Lanczos bidiagonalization process, to approximate the best parameter for Tikhonov regularization before projection.

Kaufman and Neumaier [22] suggest an envelope guided conjugate gradient approach for the Tikhonov L-curve problem. Their method is necessarily somewhat more expensive than ours because they maintain nonnegativity constraints on the variables.

Substantial work has also been done on TSVD regularization of the projected problems. Björck, Grimme, and van Dooren [2] use GCV to determine the truncation point for the projected SVD. Their emphasis is on maintaining an accurate factorization when many iterations are needed, using full reorthogonalization and implicit restart strategies. O'Leary and Simmons [28] take the viewpoint that the problem should be preconditioned appropriately so that a massive number of iterations is unnecessary. That viewpoint is echoed in this current work, so we implicitly assume that the problem has been preconditioned [28] so that $A = M^{-1}\hat{A}$ and $b = M^{-1}\hat{b}$, where $\hat{A}$ and $\hat{b}$ are the original data and M is a preconditioning matrix. See [16, 27, 24, 23] for preconditioners appropriate for certain types of ill-posed problems.

3. Regularizing the projected problem.

In this section we categorize a dozen approaches to regularization of the projected problem that arise from using Krylov methods, giving enough detail to make the costs apparent and to show that the ideas are easy to program. Many Krylov methods have been proposed; for ease of exposition we focus on just one of these: the LSQR algorithm of Paige and Saunders [30].

LSQR iteratively computes a bidiagonalization related to that introduced by Golub and Kahan [12]. After k iterations, it has effectively computed three matrices: an upper-bidiagonal matrix B_k and two matrices $U_k \equiv [u_1, \ldots, u_k]$ and $V_k \equiv [v_1, \ldots, v_k]$, with orthonormal columns, related by

$$b = \beta_1 u_1 = \beta_1 U_{k+1} e_1 \,, \tag{3.1}$$

$$A V_k = U_{k+1} B_k \,, \tag{3.2}$$

$$A^T U_{k+1} = V_k B_k^T + \alpha_{k+1} v_{k+1} e_{k+1}^T \,, \tag{3.3}$$

where e_i denotes the ith unit vector.

In numeric computations, the columns of U_k and V_k can fail to be orthonormal. This has never given us convergence difficulties, but if it becomes troublesome, there are well-known techniques to handle it [31, 32, 36, 6].

Now suppose we want to solve

$$\min_{x \in \mathcal{S}} \|b - Ax\|_2, \tag{3.4}$$

where $\mathcal{S}$ denotes the k-dimensional subspace spanned by the first k vectors v_i. The solution we seek is of the form $x^{(k)} = V_k y^{(k)}$ for some vector $y^{(k)}$ of length k. Define $r^{(k)} = b - Ax^{(k)}$ to be the corresponding residual and observe that

$$\begin{aligned} r^{(k)} &= \beta_1 u_1 - A V_k y^{(k)} \\ &= U_{k+1}(\beta_1 e_1 - B_k y^{(k)}). \end{aligned}$$

Since U_{k+1} has, in exact arithmetic, orthonormal columns, the projected problem we wish to solve is

$$\min_{y^{(k)}} \|\beta_1 e_1 - B_k y^{(k)}\|_2. \tag{3.5}$$

Solving this minimization problem is mathematically equivalent to solving the normal equations involving the bidiagonal matrix

$$B_k^* B_k y^{(k)} = \beta_1 B_k^* e_1, \tag{3.6}$$

although more stable means are used in practice. Typically k is small, so reorthogonalization to combat round-off error might or might not be necessary. The matrix B_k may be ill-conditioned because some of its singular values approximate some of the small singular values of A. Therefore, solving the projected problem might not yield

Table 3.1

Summary of flops for projection plus inner regularization with various parameter selection techniques, in addition to the $O(qk)$ flops required for projection itself. Here k is the number of iterations (i.e., the size of the projection) taken and p is the number of discrete parameters that must be tried.

Projection plus –	Disc.	GCV	L-curve
Tikhonov	$O(pk)$	$O(k^3)$	$O(pk)$
TSVD	$O(k^3)$	$O(k^3)$	$O(k^3)$
Rust's	$O(k^3)$	$O(k^3)$	$O(k^3)$

Table 3.2

Summary of additional storage for each of four regularization methods under each of three parameter selection techniques. The original matrix is $m \times n$ with q nonzeros, p is the number of discrete parameters that must be tried, k iterations are used in projection, and the factorizations are assumed to take $\hat{q}$ storage.

	Basic cost	Added cost		
		Disc.	GCV	L-curve
Tikhonov	$O(\hat{q})$	$O(1)$	$O(p)$	$O(p)$
TSVD	$O(\hat{q})$	$O(1)$	$O(m)$	$O(m)$
Rust's TSVD	$O(\hat{q})$	$O(m)$	$O(m)$	$O(m)$
Projection	$O(kn)$	$O(1)$	$O(k)$	$O(k)$

Table 3.3

Summary of storage, not including storage for the matrix, for projection plus inner regularization approach and various parameter selection techniques. Here p denotes the number of discrete parameters tried. Each of these regularization methods also requires us to save the basis V or else regenerate it in order to reconstruct x.

Projection plus –	Disc.	GCV	L-curve
Tikhonov	$O(1)$	$O(p)$	$O(p)$
TSVD	$O(1)$	$O(k)$	$O(k)$
Rust's TSVD	$O(k)$	$O(k+p)$	$O(k+p)$

a good solution $y^{(k)}$, but we can use any of the methods of section 2 to regularize this projected problem; we discuss options in detail below.

If we used the algorithm GMRES [35] instead of LSQR, we would derive similar relations. Here, though, the U and V matrices are identical and the B matrix is upper Hessenberg rather than bidiagonal. Conjugate gradients would yield similar relationships.

For cost comparisons for these methods, see Tables 2.1 and 3.1. Storage comparisons are given in Tables 3.2 and 3.3.

3.1. Regularization by projection. As mentioned earlier, if we terminate the iteration after k steps, we have projected the solution onto a k-dimensional subspace and this has a regularizing effect that is sometimes sufficient. Determining the best value of k can be accomplished, for instance, by one of our three methods of parameter choice. Efficient implementation relies on LSQR recurrences for determining $\|r^{(k)}\|$ and $\|x^{(k)}\|$ cheaply, without computing either $r^{(k)}$ or $x^{(k)}$ [30, 34].

For the **discrepancy principle.** we stop the iteration for the smallest value of k for which $\|r_k\| \leq \tau\delta$.

To apply **GCV**, we note that in LSQR (see section 3.1), the operator $AA^\sharp$ is given by $U_{k+1}B_kB_k^\dagger U_{k+1}^*$, where $B_k^\dagger$ is the pseudoinverse of B_k. Thus from (2.2), the

GCV functional is [19]

$$G(k) = \frac{\|r^{(k)}\|_2^2}{(m-k)^2}.$$

We note that there are in fact two distinct definitions for $A^\sharp$ and hence two definitions for the denominator in $G(k)$; for small enough k, the two are comparable, and the definition we use here is less expensive to calculate [19, section 7.4].

To determine the **L-curve** associated with LSQR, values of $\|r^{(k)}\|_2$ and $\|x^{(k)}\|_2$ are needed for several values of k. In using this method or GCV, one must go a few iterations beyond the optimal k in order to verify the optimum [20].

3.2. Regularization by projection plus TSVD. If projection alone does not regularize, then we can compute the TSVD regularized solution to the projected problem (3.6). We need the SVD of the $(k+1) \times k$ matrix B_k. This requires $O(k^3)$ operations but can also be computed from the SVD of B_{k-1} in $O(k^2)$ operations [14].

Clearly, we still need to use some type of parameter selection technique to find a good value of $\ell(k)$. First, notice that it is easy to compute the norms of the residual and the solution resulting from retaining only the ℓ largest singular values. If ξ_{jk} is the component of e_1 in the direction of the jth left singular vector of B_k, and if γ_j is the jth singular value (ordered largest to smallest), then the residual and solution 2-norms are

$$\|r_\ell^{(k)}\| = \beta_1 \left(\sum_{j=\ell(k)+1}^{k+1} \xi_{jk}^2 \right)^{1/2} \quad \text{and} \quad \|x_\ell^{(k)}\| = \beta_1 \left(\sum_{j=1}^{\ell(k)} \left(\frac{\xi_{jk}}{\gamma_j} \right)^2 \right)^{1/2}. \tag{3.7}$$

Using this fact, we can use any of our three sample methods.

For the **discrepancy principle** we choose $\ell(k)$ to be the smallest value for which $\|r_\ell^{(k)}\| \le \tau\delta$, if such a value exists. As k increases, the number of neglected singular values will be monotonically nondecreasing (exact arithmetic).

The **GCV** functional for the kth projected problem is obtained by substituting B_k for A and $B_k^\sharp$ for $A^\sharp$, and substituting the expression of the residual in (3.7) for the numerator in (2.2):

$$G_k(\ell) = \frac{\beta_1^2 \sum_{j=\ell+1}^{k+1} \xi_{jk}^2}{(k-\ell+1)^2}.$$

We now have many **L-curves**, one for each value of k. The coordinate values in (3.7) form the discrete L-curve for a given k, from which the desired value of $\ell(k)$ can be chosen without forming the approximate solutions or residuals.

3.3. Regularization by projection plus Rust's TSVD. As in standard TSVD, to use Rust's version of TSVD for regularization of the projected problem requires computing the SVD of the $(k+1) \times k$ matrix B_k. Using the previous notation, Rust's strategy is to set

$$y_\rho^{(k)} = \sum_{j \in \mathcal{I}_\rho^{(k)}} \frac{\xi_{jk}}{\gamma_j} q_j^{(k)},$$

where $q_j^{(k)}$ are the right singular vectors of B_k and $\mathcal{I}_\rho^{(k)} = \{i < k+1 : |\xi_{ik}| > \rho\}$. We focus on three ways to determine ρ.

For the **discrepancy principle,** the norm of the residual of the regularized solution is given by $\|r_\rho^{(k)}\|_2 = \beta_1(\sum_{j\notin\mathcal{I}_\rho^{(k)}} \xi_{jk}^2)^{1/2}$. According to the discrepancy principle, we must choose ρ so that the residual is less than $\tau\delta$. In practice, this would require that the residual be evaluated by sorting the values $|\xi_{ik}|$ and adding terms in that order until the residual norm is less than $\tau\delta$.

For **GCV,** let $\mathrm{card}(\mathcal{I}_\rho^{(k)})$ denote the cardinality of the set $\mathcal{I}_\rho^{(k)}$. From (2.2), it is easy to show that the GCV functional corresponding to the projected problem for this regularization technique is given by

$$G_k(\rho) = \frac{\beta_1^2 \sum_{j\in\mathcal{I}_\rho^{(k)}} \xi_{jk}^2}{(k+1-\mathrm{card}(\mathcal{I}_\rho^{(k)}))^2}.$$

In practice, for each k we first sort the values $|\xi_{ik}|, i = 1,\ldots,k$, from smallest to largest. Then we define k discrete values ρ_j to be equal to these values with ρ_1 being the smallest. We set $\rho_0 = 0$. Note that because the values of $\rho_j, j = 1,\ldots,k$, are the sorted magnitudes of the SVD expansion coefficients, we have

$$G_k(\rho_j) = \frac{\beta_1^2(|\xi_{(k+1),k}|^2 + \sum_{i=1}^j \rho_i^2)}{(j+1)^2}, \quad j = 0,\ldots,k.$$

Finally, we take the regularization parameter to be the ρ_j for which $G_k(\rho_j)$ is a minimum.

As with standard TSVD, we now have one **L-curve** for each value of k. For fixed k, if we define the $\rho_j, j = 0,\ldots,k$, as we did for GCV above and we reorder the γ_i in the same way that the $|\xi_{ik}|$ were reordered when sorted, then we have

$$\|x_{\rho_j}^{(k)}\|_2^2 = \beta_1^2 \sum_{i=j+1}^{k} \left(\frac{\rho_i}{\gamma_i}\right)^2; \ \|r_{\rho_j}^{(k)}\|_2^2 = \beta_1^2\left(|\xi_{(k+1),k}|^2 + \sum_{i=1}^{j} \rho_i^2\right), \quad j = 0,\ldots,k.$$

When these solution and residual norms are plotted against each other as functions of ρ, the value of ρ_j corresponding to the corner is selected as the regularization parameter.

3.4. Regularization by projection plus Tikhonov.

Finally, let us consider using Tikhonov regularization to regularize the projected problem (3.5) for some integer k. Thus, for a given regularization parameter λ, we would like to solve

$$\min_y \|\beta_1 e_1 - B_k y\|_2^2 + \lambda^2\|y\|_2^2. \tag{3.8}$$

The solution $y_\lambda^{(k)}$ satisfies

$$(V_k^* A^* A V_k + \lambda^2 I) y_\lambda^{(k)} = V_k^* A^* b. \tag{3.9}$$

We need to address how to choose a suitable value of λ.

For the **discrepancy principle,** note that in exact arithmetic, we have

$$r_\lambda^{(k)} = b - A x_\lambda^{(k)} = U_{k+1}^*(\beta_1 e_1 - B_k y_\lambda^{(k)}). \tag{3.10}$$

Hence $\|B_k y_\lambda^{(k)} - \beta_1 e_1\|_2 = \|r_\lambda^{(k)}\|_2$. Therefore, to use the discrepancy principle requires that we choose λ so that $\|r_\lambda^{(k)}\|_2 \le \tau\delta$ with p discrete trial values λ_j. For a given k, we take λ to be the largest value λ_j for which $\|r_\lambda^{(k)}\|_2 < \tau\delta$, if it exists; if not, we increase k and test again.

For **GCV,** let us define $(B_k)_\lambda^\dagger$ to be the operator mapping the right-hand side of the projected problem onto the regularized solution of the projected problem:

$$(B_k)_\lambda^\dagger = (B_k^* B_k + \lambda^2 I)^{-1} B_k^*.$$

Given the SVD of B_k as above, the denominator in the GCV functional defined for the projected problem (refer to (2.2)) is

$$\left(k + 1 - \sum_{j=1}^{k} \frac{\gamma_j^2}{\gamma_j^2 + \lambda^2} \right)^2 .$$

The numerator is simply $\|r_\lambda^{(k)}\|_2^2$. For values of $k \ll n$, it is feasible to compute the singular values of B_k.

The **L-curve** is comprised of the points $(\|B_k y_\lambda^{(k)} - \beta_1 e_1\|_2, \|y_\lambda^{(k)}\|_2)$. But using (3.10) and the orthonormality of the columns of V_k, we see these points are precisely $(\|r_\lambda^{(k)}\|_2, \|x_\lambda^{(k)}\|_2)$. For p discrete values of λ, $\lambda_i, 1 \le i \le p$, the quantities $\|r_{\lambda_i}^{(k)}\|_2$ and $\|x_{\lambda_i}^{(k)}\|_2$ can be obtained by updating their respective estimates at the $(k-1)$st iteration.[1]

3.5. Correspondence between direct regularization and projection plus regularization.

In this section, we demonstrate why the projection plus regularization approaches can be expected to yield regularized solutions nearly equivalent to the direct regularization counterpart. The following theorem, a simple corollary of the invariance of Krylov sequences under shifts, establishes the desired result for the case of Tikhonov vs. projection plus Tikhonov.

THEOREM 3.1. *Fix $\lambda > 0$ and define $x_\lambda^{(k)}$ to be the kth iterate of conjugate gradients applied to the Tikhonov problem*

$$(A^* A + \lambda^2 I)x = A^* b.$$

Let $y_\lambda^{(k)}$ be the exact solution to the regularized projected problem

$$(B_k^* B_k + \lambda^2 I)y = B_k^*(\beta e_1),$$

where B_k, V_k are derived from the original problem $A^* A = A^* b$, and set $z_\lambda^{(k)} = V_k y_\lambda^{(k)}$. Then $z_\lambda^{(k)} = x_\lambda^{(k)}$.

Proof. See [15, p. 301]. □

Let us compare TSVD regularization applied to the original problem to the projection plus TSVD approach. Direct computation convinces us that the two methods compute the same regularized solution if $k = n$ and arithmetic is exact. An approximate result holds in exact arithmetic when we take k iterations, with $\ell \le k \le n$. Let the SVD of B_k be denoted by $B_k = Z_k \Gamma_k Q_k^T$, and define the $s \times \ell$ matrix $W_{s,\ell}$ as

$$W_{s,\ell} = \begin{bmatrix} I \\ 0 \end{bmatrix}.$$

[1] The technical details of the approach are found in [29, pp. 197–198], from which we obtain $\|r_\lambda^{(k)}\| = \sqrt{\|\bar{r}_\lambda^{(k)}\|^2 - \lambda^2 \|x_\lambda^{(k)}\|^2}$. The implementation details for estimating $\|x_\lambda^{(k)}\|$ and $\|\bar{r}_\lambda^{(k)}\|$ were taken from the Paige and Saunders algorithm at http://www.netlib.org/linalg/lsqr.

Then the regularized solution obtained from the TSVD regularization of the projected problem is

$$x_{reg}^{(k)} = V_k(Q_k W_{k,\ell} \Gamma_{k,1}^{-1} W_{k+1,\ell}^T Z_k^T U_k^T b),$$

where $\Gamma_{k,1}$ denotes the leading $\ell \times \ell$ principal submatrix of Γ_k. If k is taken to be sufficiently larger than ℓ so that $V_k Q_k W_{k,\ell} \approx \hat{V} W_{n,\ell}$, $W_{k+1,\ell}^T Z_k^T U_{k+1}^T \approx W_{n,\ell}^T \hat{U}^T$, and $\Gamma_{k,1} \approx \Sigma_1$ with Σ_1 the leading principal submatrix of Σ, then we expect $x_{reg}^{(k)}$ to be a good approximation to x_ℓ. This is made more precise in the following theorem.

THEOREM 3.2. *Let $k \geq \ell$ such that*

$$(V_k Q_k W_{k,\ell}) = \hat{V}_1 + E_1 \quad \text{with } \|E_1\| \leq \delta_1 \ll 1,$$

$$(U_{k+1} Z_k W_{k+1,\ell}) = \hat{U}_1 + E_2 \quad \text{with } \|E_2\| \leq \delta_2 \ll 1,$$

where $\hat{V}_1$ and $\hat{U}_1$ contain the first ℓ columns of $\hat{V}$ and $\hat{U}$, respectively. Let $D = diag(d_1, \ldots, d_\ell)$ satisfy

$$\Gamma_{k,1} = \Sigma_1 + D \quad \text{with } |d_i| \leq \delta_3 \ll 1.$$

Then

$$\|x_{reg}^{(k)} - x_\ell\| \leq \max_{1 \leq i \leq \ell} \frac{1}{\sigma_i + d_i} \left(\frac{\delta_3}{\sigma_\ell} + 3\max(\delta_1, \delta_2) \right) \|b\|.$$

Proof. Using the representations $x_\ell = \hat{V}_1 \Sigma_1^{-1} \hat{U}_1^T b$ and $x_{reg}^{(k)} = (\hat{V}_1 + E_1)\Gamma_{k,1}^{-1}(\hat{U}_1^T + E_2^T)b$, we obtain

$$\|x_{reg}^{(k)} - x_\ell\| \leq (\|\Gamma_{k,1}^{-1} - \Sigma_1^{-1}\| + \|\Gamma_{k,1}^{-1}\| \, \|E_2\| + \|E_1\| \, \|\Gamma_{k,1}^{-1}\| + \|E_1\| \, \|\Gamma_{k,1}^{-1}\| \, \|E_2\|)\|b\| ,$$

and the conclusion follows from bounding each term. □

Note that typically $\sigma_\ell \gg \sigma_n$ so that $1/\sigma_\ell$ is not too large. The bound says that the better LSQR captures the first ℓ singular values and vectors, the more we are assured the solution obtained by projection plus TSVD is close to the TSVD regularized solution to the original problem. For some results relating to the value of k necessary for the hypothesis of the theorem to hold, refer to the theory of Kaniel-Paige and Saad [31, section 12.4]. There is no universal recipe, but if k is large enough that the projected problem satisfies the discrete Picard condition, then this is some indication that the approximability property holds.

4. Numerical results.

In this section, we present three numerical examples. All experiments were carried out using Matlab with IEEE double precision floating point arithmetic. Where noted, we made use of certain routines in Hansen's Regularization Tools [18]. Since the exact, noise-free solutions were known in these examples, we evaluated the methods using the relative, 2-norm difference between the regularized solutions and the exact solutions. When we applied Rust's method to the original problem, the ρ_i were taken to be the magnitudes of the spectral coefficients of b sorted in increasing order.

TABLE 4.1

Example 1: *comparison of* $\|x_{true} - x_{reg}\|_2/\|x_{true}\|_2$ *for each of four regularization methods on the original problem, where the regularization method was chosen using methods indicated. The parameter values selected for each method are indicated in parentheses.*

	Disc.	GCV	L-curve	Optimal
Tikhonov	(1.6E−1) 2.2E−2	(8.0E−2) 2.2E−2	(4.0E−2) 4.3E−2	(1.3E−1) 2.1E−2
TSVD	(6) 1.1E−1	(9) 1.6E−2	(10) 1.6E−2	(9) 1.6E−2
Rust's TSVD	(1.6E−2) 2.5E−2	(5.3E−5) 2.2E + 4	(1.6E−2) 2.5E−2	(1.6E−2) 2.5E−2
Projection	(5) 2.5E−2	(5) 2.5E−2	(10) 2.2E−2	(9) 2.2E−2

4.1. Example 1.

The 200×200 matrix A and true solution x_{true} for this example were generated using the function `phillips` in Hansen's Regularization Tools. We generated $b_{true} = Ax_{true}$ and then computed the noisy vector b as $b + e$, where e was generated using the Matlab `randn` function and was scaled so that the noise level, $\frac{\|e\|}{\|b_{true}\|}$, was 5×10^{-3}. The condition number of A was on the order of 4×10^7.

Table 4.1 displays the values of the regularization parameters chosen when the original problem was solved using one of the three parameter selection techniques together with one of the four regularization methods. We set $\tau\delta$ for the discrepancy principle to be 8E−2, close to the value $\|e\|_2 =$ 7.65E−2.

The last column in the table gives the value of the parameter that yielded a regularized solution with minimum relative error. Several values of λ were tested: $\log_{10} \lambda = -4, -3.9, \ldots, 0$. The relative error values for regularized solutions corresponding to the parameters are also presented in this table. The GCV and L-curve parameters for projection were determined after 15 iterations. Note that using GCV to determine a regularization parameter for Rust's TSVD resulted in an extremely noisy solution with huge error.

The corners of the L-curves for the Tikhonov, projection, and TSVD methods were determined using Hansen's `lcorner` function, with the modification that sometimes points not strictly on the portion of the curve that was L-shaped (that is, points with very large residual or very small residual) were not considered (otherwise, a false corner resulted); this was most often a concern with the TSVD method. Since the corner was so clearly defined for Rust's method but the function had trouble automatically finding the corner, the corner was picked manually.

Next, we projected using LSQR and then regularized the projected problem with one of the other three regularization methods together with one of the three parameter selection techniques. Results at iterations 10 and 25 are given in Tables 4.2 and 4.3, respectively. As before, the `lcorner` routine was used to determine the corners of the respective L-curves, with the modifications as mentioned above.

Comparing Tables 4.1 and 4.2, we observe that using either the discrepancy principle or the L-curve, 10 steps of projection plus Tikhonov gives results as good as or much better than if those techniques had been used with Tikhonov on the original problem. A similar statement can be made for projection plus Rust's TSVD when any of the 3 selection methods are used and for projection plus TSVD when the discrepancy principle is used. After 25 iterations, the errors for projection plus Tikhonov or Rust's TSVD closely resemble the errors in Table 4.1 with one exception. We note that at 25 iterations, the parameters chosen for projection plus Tikhonov by the discrepancy principle or the L-curve method and their corresponding errors are identical to those chosen for the original problem.

In fact, the L-curve, GCV, and discrepancy methods applied to the projected

TABLE 4.2

Example 1*, iteration* 10*: comparison of* $\|x_{true} - x_{reg}\|_2/\|x_{true}\|_2$ *for projection plus Tikhonov, TSVD, and Rust's TSVD. The parameter values for each method are indicated in parentheses.*

	Disc.	GCV	L-curve	Optimal
Tikhonov	(1.6E−1) 2.1E−2	(2.5E−2) 2.5E−2	(2.0E−4) 2.2E−2	(2.0E−2) 2.0E−2
TSVD	(7) 2.5E−2	(7) 2.5E−2	(10) 2.2E−2	(10) 2.2E−2
Rust's TSVD	(9.7E−3) 2.5E−2	(9.7E−3) 2.5E−2	(5.5E−4) 2.2E−2	(9.1E−3) 2.1E−2

TABLE 4.3

Example 1*, iteration* 25*: comparison of* $\|x_{true} - x_{reg}\|_2/\|x_{true}\|_2$ *for projection plus Tikhonov, TSVD, and Rust's TSVD. The parameter values are given in parentheses.*

	Disc.	GCV	L-curve	Optimal
Tikhonov	(1.6E−1) 2.2E−2	(2.0E−1) 2.3E−2	(4.0E−2) 4.3E−2	(1.3E−1) 2.1E−2
TSVD	(17) 2.5E−2	(17) 2.5E−2	(21) 2.4E−2	(19) 1.6E−2
Rust's TSVD	(2.0E−2) 2.5E−2	(2.0E−2) 2.5E−2	(1.5E−2) 2.5E−2	(1.5E−2) 2.5E−2

problem with Tikhonov regularization consistently chose the same parameter for future iterations (see Figure 4.1, for instance), and correspondingly the errors remain constant; however, the results at earlier iterations are actually better than after the parameter on the projected problem has converged to the L-curve parameter on the original. For the projection plus TSVD, both the discrepancy principle and GCV method yielded parameters for which the solutions had similar errors from one iteration to the next for at least the first 80 iterations (see the top of Figure 4.2); the L-curve behaved slightly less consistently for iterations beyond about 50. Discrepancy and GCV when applied to projection plus Rust's TSVD also gave consistent solutions for about 40 iterations, after which the GCV solutions began to grow very large in error, much like GCV applied to the original problem (refer to the bottom of Figure 4.2).

4.2. Example 2. The 3969×3969 matrix A for this example was a symmetric, block Toeplitz matrix with Toeplitz blocks formed according to $A = T \otimes T$. Here T is a symmetric, banded Toeplitz matrix with entries $T_{i,j} = t_{i-j}$; the nonzero entries in the first row were $t_k = (\sin(k/B)/(k/B))^2, 0 \le k \le 4, B = .8$. The singular values of this matrix range from 5.7 to 8.6×10^{-8} but do not decay very quickly, and the matrix has a condition number of about 7×10^7. x was obtained by stacking by columns the 63×63 image that was zero except for a rectangle with value 1 from rows 20 to 49, columns 4 to 24, and another rectangle with value .8 at rows 23 to 53, columns 29 to 52. We generated $b_{true} = Ax_{true}$ and then computed the noisy vector b as $b + e$, where e was generated using the Matlab `randn` function and was scaled so that the noise level, $\frac{\|e\|}{\|b_{true}\|}$, was 2×10^{-3}.

We generated our discrete λ_i using $\log_{10} \lambda = -4, -4.9, \ldots, 0$. The norm of the noise vector was 3.66E−1, so we took $\tau\delta = 4.00\text{E}{-1}$ for the discrepancy principle.

In this example, when no preconditioning was used, it took 90 iterations for LSQR to reach a minimum relative error of 7.93E−2. Likewise, the dimension k of the projected problem had to be at least 90 to obtain good results with the projection-plus-regularization approaches and even larger for the parameter selection techniques to work well on the projected problem. Therefore, for the projection based techniques, we chose to work with a left preconditioned system (refer to the discussion at the end of section 2.1). Our preconditioner was chosen as in [23] where the parameter defining the preconditioner was taken to be $m = 2080$. Results for right preconditioning were

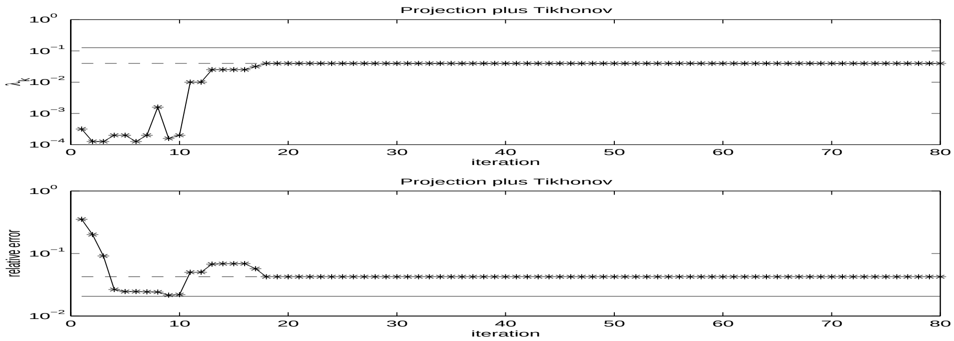

FIG. 4.1. *Example* 1. *Top: λ_k as selected by L-curve method; bottom: relative error for corresponding solution. The solid line indicates the optimal value on the original problem, and the dashed line indicates value selected by L-curve on the original problem.*

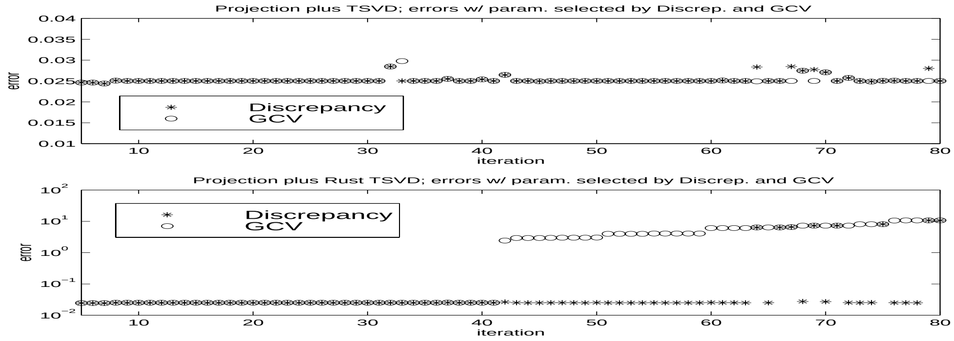

FIG. 4.2. *Example* 1. *Relative error between computed and exact solutions for projection plus TSVD (top) and projection plus Rust's TSVD (bottom) when the parameters for the projected problem are selected by either the discrepancy principle (*) or GCV method (o).*

similar, although the errors were not quite as small. On other examples, though, we found that right preconditioning by this type of preconditioner was only effective in certain instances, even when left preconditioning was effective.[2]

The results of the resulting regularization for the original problem parameters are given in Table 4.4. We note that GCV with Rust's TSVD was ineffective. Also, after 50 iterations on the left preconditioned system, the GCV functional for projection was still decreasing, so the value in Table 4.4 corresponds to the value after 50 iterations. The L-curve parameter in the table was determined after 20 iterations.

Although we projected using LSQR, we note that since the matrix and preconditioner were symmetric, we could have used MINRES as in [23]. The results in each case at iterations $10, 20$, and 40 are given in Tables 4.5, 4.6, and 4.7, respectively, and we summarize results up to 60 iterations in the discussion below.

Again, we used the `lcorner` routine to determine the corners of the respective L-curves, with the modification that for 20 iterations and beyond for TSVD, we first removed points on the curve with residual norm greater than 10 to avoid detecting a false corner.

[2] In the language of [23], right preconditioning worked well only when K was a very good approximation to C so that right preconditioning did not mix noise into early iterates; left preconditioning was not nearly as sensitive to the approximation on the transition and noise subspaces.

Table 4.4

Example 2*: comparison of* $\|x_{true} - x_{reg}\|_2/\|x_{true}\|_2$ *for each of four regularization methods on the original problem. The parameter values are given in parentheses. The projection was performed on a left preconditioned system.*

	Disc.	GCV	L-curve	Optimal
Tikhonov	(1.6E−1) 8.5E−2	(5.0E−2) 8.0E−2	(3.2E−3) 5.3E−1	(6.3E−2) 7.8E−2
TSVD	(2073) 9.9E−2	(2534) 8.1E−2	(1509) 1.2E−1	(2521) 8.0E−2
Rust's TSVD	(2.1E−2) 7.6E−2	(9.2E−2) 4.0E + 3	(1.6E−2) 2.3E−1	(2.0E−2) 7.6E−2
Projection	(2) 9.7E−2	(50+) 2.7E−1	(13) 8.3E−2	(8) 7.9E−2

Table 4.5

Example 2*, iteration* 10*: comparison of* $\|x_{true} - x_{reg}\|_2/\|x_{true}\|_2$ *for projection plus Tikhonov, TSVD, and Rust's TSVD. Parameter values are given in parentheses.*

	Disc.	GCV	L-curve	Optimal
Tikhonov	(7.9E−2) 7.9E−2	(6.3E−2) 7.9E−2	(2.0E−4) 7.9E−2	(5.0E−2) 7.9E−2
TSVD	(6) 9.9E−2	(6) 7.9E−2	(8) 9.8E−2	(10) 7.9E−2
Rust's TSVD	(2.2E−1) 8.5E−2	(2.6E−1) 9.9E−2	(2.3E−1) 9.9E−2	(3.9E−4) 7.9E−2

Table 4.6

Example 2*, iteration* 20*: comparison of* $\|x_{true} - x_{reg}\|_2/\|x_{true}\|_2$ *for projection plus Tikhonov, TSVD, and Rust's TSVD. Parameter values are given in parentheses.*

	Disc.	GCV	L-curve	Optimal
Tikhonov	(7.9E−2) 7.9E−2	(6.3E−2) 7.8E−2	(2.0E−4) 1.1E−1	(6.3E−2) 7.8E−2
TSVD	(12) 9.9E−2	(12) 9.9E−2	(19) 8.3E−2	(19) 8.3E−2
Rust's TSVD	(1.6E−1) 9.5E−1	(7.9E−2) 1.1E−1	(4.6E−2) 1.1E−1	(1.3E−1) 8.3E−2

Table 4.7

Example 2*, iteration* 40*: comparison of* $\|x_{true} - x_{reg}\|_2/\|x_{true}\|_2$ *for projection plus Tikhonov, TSVD, and Rust's TSVD. Parameter values are given in parentheses.*

	Disc.	GCV	L-curve	Optimal
Tikhonov	(7.9E−2) 7.9E−2	(6.3E−2) 7.8E−2	(2.0E−1) 2.3E−1	(6.3E−2) 7.9E−2
TSVD	(24) 9.9E−2	(24) 9.9E−2	(28) 9.9E−2	(38) 8.3E−2
Rust's TSVD	(1.5E−1) 9.2E−2	(5.8E−2) 2.3E−1	(1.6E−1) 9.2E−2	(1.5E−1) 9.2E−2

Discrepancy and GCV consistently chose the same regularization parameter and hence gave the same error for projection plus Tikhonov for 10 to 60 iterations. From the tables, we see that these are not the same parameters as those chosen when applied to the original problem and that, in fact, the solutions for projection plus Tikhonov have smaller error. The errors for the solutions obtained using any of the 3 parameter selection methods applied to find ℓ for projection plus TSVD were also consistent for 10 to 60 iterations, as alluded to in the tables. Figure 4.3 shows the errors from iterations 5 to 60 for projection plus Tikhonov and projection plus TSVD when GCV is used. For Rust's TSVD, the L-curve and discrepancy rules are fairly consistent at picking parameters that give solutions with similar error from iteration to iteration. We note that GCV for Rust's TSVD picked parameters giving solutions with reasonably small errors, even though GCV for Rust's TSVD on the original problem failed, giving a solution with huge error. A similar statement can be made for the L-curve with projection plus Tikhonov.

Summarizing, we observe two phenomena. First, the parameters selected to regularize the projected problem can be different from those chosen on the original problem but still yield solutions of better or comparable error. Second, as this and the previous

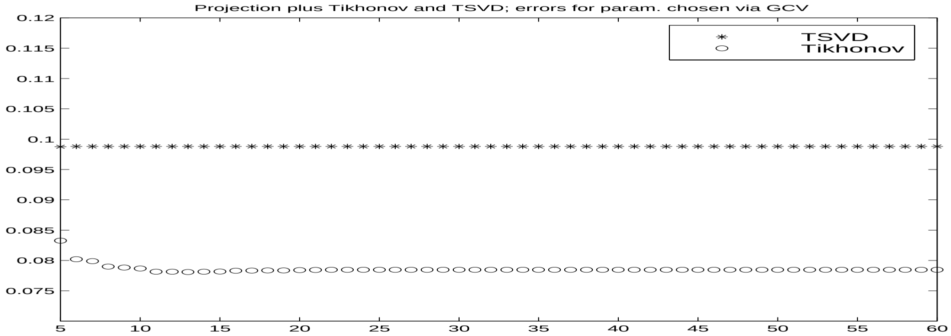

Fig. 4.3. *Example* 2*: Errors for projection plus Tikhonov (*) and projection plus TSVD (o) when the regularization parameter for the projected problem was given by GCV.*

Table 4.8

Example 3: *comparison of* $\|x_{true} - x_{reg}\|_2 / \|x_{true}\|_2$ *for each of the 4 regularization methods on the original problem. Parameter values are given in parentheses. Those for GCV and the L-curve are those selected after* 30 *iterations.*

	Disc.	GCV	L-curve	Optimal
Tikhonov	(1.0) 3.9E−1	(1.3) 4.0E−1	(5.0E−1) 4.1E−1	(7.9E−1) 3.9E−1
TSVD	(232) 4.2E−1	(400) 1.3E+4	(261) 4.0E−1	(241) 4.0E−1
Rust's TSVD	(3.0E−1) 7.4E+2	(1.8E−1) 1.2E+4	(3.7) 4.6E−1	(3.9E−1) 4.6E−1
Projection	(9) 4.0E−1	(23) 4.3E−1	(16) 4.0E−1	(12) 3.9E−1

example show, loss of orthogonality does not seem to hamper the parameter selection process, at least not for a reasonable number of iterations. This may be due to the fact that the parameter selection methods are applied directly to the projected problem: for example, the denominator of our GCV function for projection plus TSVD is different from the denominator of the GCV function given in [2, (3.8)].

4.3. Example 3. Our final example is from the field of computed tomography. In this example, the true vector x corresponded to the 20×20 image created with the `phantom.m` function. The matrix A was the corresponding 561×400 Radon transform matrix where it is understood that the data was taken at angles from 0 to 179 degrees in increments of 11 degrees. The matrix itself was computed (albeit naively) in Matlab column by column using successive applications of `radon.m` on images of point sources. The singular values fall off very slowly at first (the first 260 of the 400 singular values range between 18 and about 1) after which they fall off rapidly, resulting in a condition number for A of about 10^7.

Since the norm of the noise vector was about 3.44, we took the tolerance for the discrepancy principle to be 3. The discrete values λ_i used for Tikhonov regularization were 51 evenly log-spaced points between 10^{-4} and 10^1. The results computed using discrepancy, GCV, and L-curve methods for Tikhonov, TSVD, Rust's TSVD, and projection on the original problem are given in Table 4.8.

Table 4.9 gives the results after 10 iterations of LSQR. Notice that the errors for the projection plus Tikhonov solutions via GCV and L-curve are slightly better than the corresponding error for Tikhonov without projection at only 10 iterations. Also interesting is the fact that at 10 iterations the discrepancy and GCV methods for projection plus Rust's TSVD give solutions with reasonable errors, whereas these techniques give solutions with very large errors when applied to the original problem.

TABLE 4.9

Example 3, *iteration* 10: *comparison of* $\|x_{true} - x_{reg}\|_2/\|x_{true}\|_2$ *for projection plus Tikhonov, TSVD, and Rust's TSVD. Parameter values are given in parentheses.*

	Disc.	GCV	L-curve	Optimal
Tikhonov	(1.0) 4.0E−1	(2.2) 4.0E−1	(1.6E−4) 3.9E−1	(4.0E−1) 4.0E−1
TSVD	(10) 3.9E−1	(1) 8.6E−1	(5) 8.3E−1	(10) 3.9E−1
Rust's TSVD	(1.0) 3.9E−1	(1.5) 4.0E−1	(2.2) 4.0E−1	(0.0) 3.9E−1

TABLE 4.10

Example 3, *iteration* 40: *comparison of* $\|x_{true} - x_{reg}\|_2/\|x_{true}\|_2$ *for projection plus Tikhonov, TSVD, and Rust's TSVD. Parameter values are given in parentheses.*

	Disc.	GCV	L-curve	Optimal
Tikhonov	(1.0) 3.9E−1	(1.2) 4.1E−1	(5.0E−1) 4.1E−1	(7.9E−1) 3.9E−1
TSVD	(37) 4.0E−1	(15) 7.8E−1	(39) 4.1E−1	(38) 4.0E−1
Rust's TSVD	(6.0E−1) 4.2E−1	(1.2) 4.1E−1	(2.7E−1) 4.1E−1	(6.6E−1) 4.0E−1

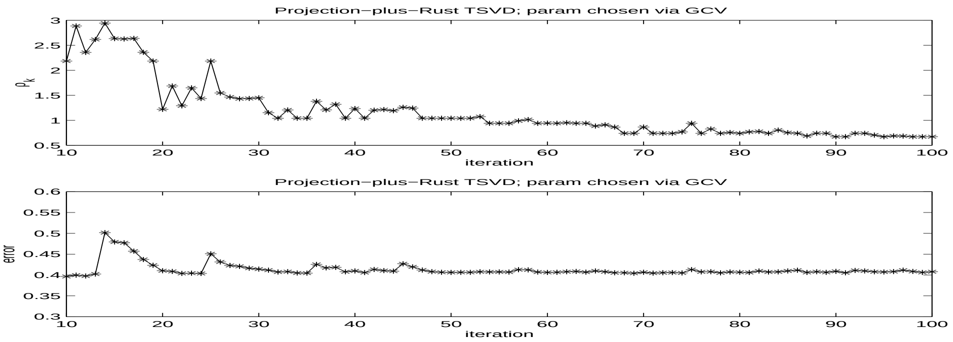

FIG. 4.4. *Example* 3. *Top: Value of* ρ_k *selected by GCV for projection plus Rust's TSVD; Bottom: Relative error of the corresponding solutions.*

Table 4.10 shows the parameters and the errors after 40 iterations. From these results, we see that the L-curve for projection plus Tikhonov eventually gives the same regularization parameter and same solution error as when applied to the larger problem, and we observed this to be true for several iterations beyond 40. Again, we see that discrepancy and GCV used with projection plus Rust's TSVD is effective, whereas they are ineffective when used on the original problem; we observed this behavior well beyond 40 iterations (see Figure 4.4).

5. Conclusions.

In this work we have given a common framework for methods based on regularizing a projected problem. We have shown that determining regularization parameters based on the final projected problem rather than on the original discretization has firmer mathematical justification and often involves less computational expense. We presented results that in fact the regularized solution obtained by backprojecting the TSVD or Tikhonov solution to the projected problem is almost equivalent to applying TSVD or Tikhonov to the original problem, where "almost" depends on the size of k. The examples indicate the practicality of the method and illustrate that our regularized solutions are usually as good as those computed using the original system, and they can be computed in a fraction of the time, using a fraction of the storage. We note that similar approaches are valid using other Krylov subspace methods for computing the projected problem.

In this work, we did not address potential problems from loss of orthogonality as the iterations progress. In this discussion, we did, however, assume that either k was naturally very small compared to n or that preconditioning had been applied to enforce this condition. Possibly for this reason, we found that for modest k, round-off did not appear to degrade either the LSQR estimates of the residual and solution norms or the computed regularized solution in the following sense: the regularization parameters chosen via the projection-regularization and the corresponding regularized solutions were comparable to those chosen and generated for the original discretized problem. Another possible reason for the success of our approach is that we chose parameters for the projected problem directly, rather than for the backprojected, larger problem. In our experiments, we found that the parameters selected usually leveled out after a few iterations. The stagnation of the parameters themselves may suggest when k is large enough.

For the Tikhonov approach in this paper, we have assumed that the regularization operator L was the identity or was related to the preconditioning operator; this allowed us to efficiently compute $\|r_\lambda^{(k)}\|$ and $\|x_\lambda^{(k)}\|$ for multiple values of λ efficiently for each k. If L is not the identity but is invertible, we can first implicitly transform the problem to "standard form" [19]. With $\bar{A} = AL^{-1}$, $\bar{x} = Lx$, we can solve the equivalent system

$$\min_{\bar{x}} = \|\bar{A}\bar{x} - b\|_2^2 + \lambda^2 \|\bar{x}\|_2^2.$$

Then the projection plus regularization schemes may be applied to this transformed problem. Clearly the projection based schemes will be useful as long as solving systems involving L can be done efficiently.

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. V. Dooren, *An implicit shift bidiagonalization algorithm for ill-posed systems*, BIT, 34 (1994), pp. 510–534.

[3] P. Blomgren and T. F. Chan, *Modular Solvers for Constrained Image Restoration Problems*, Tech. Report, Mathematics Department, UCLA, Los Angeles, 1999.

[4] D. Calvetti, G. Golub, and L. Reichel, *Estimation of the L-curve via Lanczos bidiagonalization*, BIT, 39 (1999), pp. 603–619.

[5] T. Chan and M. Ng, *Galerkin projection method for solving multiple linear systems*, SIAM J. Sci. Comput., 21 (1999), pp. 836–850.

[6] J. Cullum and R. A. Willoughby, *Lanczos and the computation in specified intervals of the spectrum of large, sparse real symmetric matrices*, in Sparse Matrix Proceedings 1978, I. S. Duff and G. W. Stewart, eds., SIAM, Philadelphia, 1979, pp. 220–255.

[7] L. Desbat and D. Girard, *The "minimum reconstruction error" choice of regularization parameters: Some more efficient methods and their application to deconvolution problems*, SIAM J. Sci. Comput., 16 (1995), pp. 1387–1403.

[8] H. W. Engl and W. Grever, *Using the L-curve for determining optimal regularization parameters*, Numer. Math., 69 (1994), pp. 25–31.

[9] H. E. Fleming, *Equivalence of regularization and truncated iteration in the solution of ill-posed image reconstruction problems*, Linear Algebra Appl., 130 (1990), pp. 133–150.

[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. 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. Golub and W. Kahan, *Calculating the singular values and pseudo-inverse of a matrix*, J. Soc. Indust. Appl. Math. Ser. B Numer. Anal., 2 (1965), pp. 205–224.

[13] W. Groetsch, *Theory of Tikhonov Regularization for Fredholm Equations of the First Kind*, Pitman, Boston, 1984.

[14] M. Gu and S. Eisenstat, *A Stable and Fast Algorithm for Updating the Singular Value Decomposition*, Tech. Report RR-939, Department of Computer Science, Yale University, New Haven, 1993.

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

[16] M. Hanke, J. Nagy, and R. Plemmons, *Preconditioned iterative regularization for ill-posed problems*, in Numerical Linear Algebra and Scientific Computing, L. Reichel, A. Ruttan, and R. S. Varga, eds. 1993, pp. 141–163.

[17] P. C. Hansen, *Analysis of discrete ill-posed problems by means of the L-curve*, SIAM Rev., 34 (1992), pp. 561–580.

[18] P. C. Hansen, *Regularization tools: A Matlab package for analysis and solution of discrete ill-posed problems*, Numer. Algorithms, 6 (1994), pp. 1–35.

[19] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems. Numerical Aspects of Linear Inversion,* SIAM Monogr. Math. Model Comput., SIAM, Philadelphia, 1998.

[20] P. C. Hansen and D. P. O'Leary, *The use of the L-curve in the regularization of discrete ill-posed problems*, SIAM J. Sci. Comput., 14 (1993), pp. 1487–1503.

[21] B. Hofmann, *Regularization for Applied Inverse and Ill-Posed Problems*, Teubner-Texte Mathe. 85, Teubner, Leipzig, 1986.

[22] L. Kaufman and A. Neumaier, *Regularization of ill-posed problems by envelope guided conjugate gradients*, J. Comput. Graph. Statist., 6 (1997), pp. 451–463.

[23] M. Kilmer, *Regularization of ill-posed problems using (symmetric) Cauchy-like preconditioners*, in Proceedings of the SPIE Annual Meeting, Advanced Signal Processing Algorithms, Architectures, and Implementations VIII, 1998, SPIE, San Diego, CA, pp. 381–392.

[24] M. Kilmer and D. P. O'Leary, *Pivoted Cauchy-like preconditioners for regularized solution of ill-posed problems*, SIAM J. Sci. Stat. Comput., 21 (1999), pp. 88–110.

[25] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, NJ, 1974.

[26] V. A. Morozov, *On the solution of functional equations by the method of regularization*, Soviet Math. Dokl., 7 (1966), pp. 414–417.

[27] J. Nagy, R. Plemmons, and T. Torgersen, *Iterative image restoration using approximate inverse preconditioning*, IEEE Trans. Image Process., 5 (96), pp. 1151–1163.

[28] D. P. O'Leary and J. A. Simmons, *A bidiagonalization-regularization procedure for large scale discretization of ill-posed problems*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 474–489.

[29] C. C. Paige and M. A. Saunders, *Algorithm* 583*, LSQR: Sparse linear equations and least squares problems*, ACM Trans. Math. Software, 8 (1982), pp. 195–209.

[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] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[32] B. N. Parlett and D. S. Scott, *The Lanczos algorithm with selective orthogonalization*, Math. Comp., 33 (1979), pp. 217–238.

[33] B. W. Rust, *Truncating the Singular Value Decomposition for Ill-Posed Problems*, Tech. Report NISTIR 6131, Mathematical and Computational Sciences Division, National Institute of Standards and Technology, Gaithersburg, MD, 1998.

[34] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Boston, 1996.

[35] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

[36] H. D. Simon, *Analysis of the symmetric Lanczos algorithm with reorthogonalization methods*, Linear Algebra Appl., 61 (1984), pp. 101–131.

[37] J. M. Varah, *Pitfalls in the numerical solution of linear ill-posed problems*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 164–176.

[38] 1993, pp. 141–163. C. R. Vogel, *Non-convergence of the L-curve regularization parameter selection method*, Inverse Problems, 12 (1996), pp. 535–547.