ON THE REDUCTION OF TIKHONOV MINIMIZATION PROBLEMS AND THE CONSTRUCTION OF REGULARIZATION MATRICES

L. DYKES* AND L. REICHEL†

Dedicated to Claude Brezinski and Sebastiano Seatzu on the Occasion of Their 70th Birthdays.

Abstract. Tikhonov regularization replaces a linear discrete ill-posed problem by a penalized least-squares problem, whose solution is less sensitive to errors in the data and round-off errors introduced during the solution process. The penalty term is defined by a regularization matrix and a regularization parameter. The latter generally has to be determined during the solution process. This requires repeated solution of the penalized least-squares problem. It is therefore attractive to transform the least-squares problem to simpler form before solution. The present paper describes a transformation of the penalized least-squares problem to simpler form that is faster to compute than available transformations in the situation when the regularization matrix has linearly dependent columns and no exploitable structure. Properties of this kind of regularization matrices are discussed and their performance is illustrated.

Key words. ill-posed problem, Tikhonov regularization, regularization matrix, GSVD

1. Introduction.

We are concerned with the solution of least-squares problems

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \|A\boldsymbol{x}-\boldsymbol{b}\|, \quad A\in\mathbb{R}^{m\times n}, \quad \boldsymbol{x}\in\mathbb{R}^n, \quad \boldsymbol{b}\in\mathbb{R}^m, \quad m\geq n, \tag{1.1}$$

with a matrix whose singular values "cluster" at the origin. In particular, A is severely ill-conditioned and may be singular. This kind of minimization problems are commonly referred to as discrete ill-posed problems. They arise when discretizing ill-posed problems, such as Fredholm integral equations of the first kind with a smooth kernel. The vector $\boldsymbol{b}$ represents data and typically is contaminated by an unknown error $\boldsymbol{e}\in\mathbb{R}^m$, which may be due to measurement inaccuracies or discretization. We will refer to the vector $\boldsymbol{e}$ as "noise." Let $\hat{\boldsymbol{b}}\in\mathbb{R}^m$ denote the unknown error-free vector associated with $\boldsymbol{b}$. Thus,

$$\boldsymbol{b}=\hat{\boldsymbol{b}}+\boldsymbol{e}. \tag{1.2}$$

We will assume that a bound

$$\|\boldsymbol{e}\|\leq\varepsilon$$

is available, and that the linear system of equations with the unknown error-free right-hand side,

$$A\boldsymbol{x}=\hat{\boldsymbol{b}}, \tag{1.3}$$

is consistent. However, the methods discussed also can be applied when these conditions are violated. Throughout this paper $\|\cdot\|$ denotes the Euclidean vector norm.

Let $\hat{\boldsymbol{x}}$ denote the solution of minimal norm of the unknown consistent system (1.3). We would like to determine an approximation of $\hat{\boldsymbol{x}}$ by computing a suitable approximate solution of the available discrete ill-posed problem (1.1). It is well known

*Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA. E-mail: ldykes@math.kent.edu.

†Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA. E-mail: reichel@math.kent.edu. Research supported in part by NSF grant DMS-1115385.

that due to the severe ill-conditioning of A, the solution of minimal norm of (1.1) generally is not a meaningful approximation of $\hat{\boldsymbol{x}}$. A common approach to remedy this difficulty is to replace the least-squares problem (1.1) by a nearby problem, whose solution is less sensitive to the error $\boldsymbol{e}$. One of the most popular approaches to achieve this, known as Tikhonov regularization, replaces (1.1) by a penalized least-squares problem of the form

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \left\| \begin{bmatrix} A \\ \mu L \end{bmatrix} \boldsymbol{x} - \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix} \right\|. \tag{1.4}$$

Here $L \in \mathbb{R}^{p\times n}$ is referred to as the regularization matrix and the scalar $\mu \geq 0$ as the regularization parameter; see, e.g., Engl et al. [11] and Hansen [15] for discussions on Tikhonov regularization.

Common regularization matrices L are the identity matrix I and finite difference matrices such as

$$L := \begin{bmatrix} -1 & 2 & -1 & & & 0 \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ 0 & & & -1 & 2 & -1 \end{bmatrix} \in \mathbb{R}^{(n-2)\times n}. \tag{1.5}$$

The minimization problem (1.4) is said to be in *standard form* when $L = I$ and in *general form* otherwise. Many examples of regularization matrices can be found in [3, 4, 7, 16, 18]. Typically, p, the number of rows of L, is smaller than or equal to n, but regularization matrices with $p > n$ also find application.

The matrix L is assumed to be chosen so that

$$\mathcal{N}(A) \cap \mathcal{N}(L) = \{\boldsymbol{0}\}. \tag{1.6}$$

Then the Tikhonov minimization problem (1.4) has the unique solution

$$\boldsymbol{x}_\mu := (A^TA + \mu^2 L^T L)^{-1} A^T \boldsymbol{b} \tag{1.7}$$

for any $\mu > 0$. Here and below the superscript ^{T} denotes transposition.

We apply the discrepancy principle to determine a suitable value of μ. It prescribes that $\mu > 0$ be chosen so that the associated solution (1.7) satisfies

$$\|A\boldsymbol{x}_\mu - \boldsymbol{b}\| = \eta\varepsilon \tag{1.8}$$

for a user-supplied parameter $\eta > 1$, which usually is chosen to be fairly close to unity. Detailed discussions on the discrepancy principle can be found in [11, 15]. We can find a value $\mu > 0$ so that the associated solution (1.7) of (1.4) satisfies (1.8) by applying a zero-finder to the nonlinear function

$$\phi(\mu) := \|\boldsymbol{b} - A\boldsymbol{x}_\mu\|^2 - \eta^2\varepsilon^2, \qquad \mu > 0. \tag{1.9}$$

This function is a strictly increasing and, therefore, ϕ has a unique zero for $\eta\varepsilon$ sufficiently large; see Section 2 for details. The computation of this zero with a zero-finder generally requires the evaluation of $\phi(\mu)$ for several values of μ, and this makes it necessary to solve (1.4) for $\boldsymbol{x}_\mu$ for several values of μ. The repeated solution of (1.4) without initial simplification of the problem can be expensive when the matrices A and L are fairly large. This paper discusses methods for transforming the least-squares problem (1.4) to simpler form that allows efficient evaluation of $\phi(\mu)$ for a sequence of μ-values. We are interested in problems that are large enough to make the transformation worthwhile, but small enough to allow factorization of the matrices involved. This means typically matrices A and L of order from a few hundred to a few thousand.

While this paper focuses on the determination of μ by the discrepancy principle, we note that also many other approaches to determine μ, including the L-curve criterion, generalized cross validation, and extrapolation, require the evaluation of $\phi(\mu)$ for several values of μ; see, e.g., [5, 6, 15] for examples and discussions. The transformations discussed in this paper therefore also are attractive to apply in the context of these parameter determination methods.

To reduce the expense of repeated solution of the penalized least-squares problem (1.4), one transforms the problem to simpler form before solution. Eldén [10] introduced the A-weighted pseudoinverse of L, given by

$$L_A^\dagger := (I - (A(I - L^\dagger L))^\dagger A)L^\dagger, \tag{1.10}$$

for this purpose. Here $L^\dagger$ denotes the Moore-Penrose pseudoinverse of L. The penalized least-squares problem (1.4) can be transformed to standard form with the aid of $L_A^\dagger$. This transformation reduces the computational effort required for each evaluation of $\phi(\mu)$; see Section 2 for details.

It is attractive to use the matrix $L_A^\dagger$ when the pseudoinverse $L^\dagger$ can be determined inexpensively and $\mathcal{N}(L)$ is explicitly known. This is the case when L is given by (1.5). We are interested in the situation when L does not have a structure that allows inexpensive computation of $L^\dagger$. Section 2 discusses the computational effort required to use $L_A^\dagger$ under these circumstances.

A popular approach to simplifying (1.4) when $L_A^\dagger$ is expensive to apply is to use the generalized singular value decomposition (GSVD) of the matrix pair $\{A, L\}$,

$$A = U\mathrm{diag}[\sigma_1, \sigma_2, \ldots, \sigma_p, \underbrace{1, 1, \ldots, 1}_{n-p}]X, \tag{1.11}$$

$$L = V\Big[\mathrm{diag}[\rho_1, \rho_2, \ldots, \rho_p], \underbrace{\mathbf{0}, \mathbf{0}, \ldots, \mathbf{0}}_{n-p}\Big]X, \tag{1.12}$$

where the matrices $U \in \mathbb{R}^{m\times n}$ and $V \in \mathbb{R}^{p\times p}$ have orthonormal columns, $X \in \mathbb{R}^{n\times n}$ is nonsingular, and the scalars σ_j and ρ_j are ordered and scaled so that

$$0 \le \sigma_1 \le \sigma_2 \le \ldots \le \sigma_p \le 1, \qquad 0 \le \rho_p \le \rho_{p-1} \le \ldots \le \rho_1 \le 1,$$

and

$$\sigma_j^2 + \rho_j^2 = 1, \qquad 1 \le j \le p;$$

see, e.g., [1, 12, 13, 15] for details. Substituting the decompositions (1.11)-(1.12) into (1.4) gives an equivalent minimization problem with A and L replaced by diagonal matrices. This simplified problem can be solved and $\phi(\mu)$ can be evaluated in $\mathcal{O}(p)$ arithmetic floating point operations (flops) for every value of $\mu > 0$.

A reason why the GSVD of $\{A, L\}$ is not always used for solving (1.4) is that its computation can be quite expensive. Bai [1, Table 5.1] shows the leading term in the flop counts for several GSVD algorithms when $A, L \in \mathbb{R}^{n\times n}$. The counts for some algorithms depend on the number of sweeps, j, required. Typically, $2 \le j \le 10$.

When j is small, Paige's algorithm has the smallest leading coefficient, $(5.3+15j)n^3$, which generally is at least $35.3n^3$ and could be much larger.

It is the purpose of this paper to present a reduction method that, similarly to the GSVD, is well suited for the situation when L has linearly dependent columns and no structure that can be exploited to simplify the computation of $L^\dagger$, and is cheaper to apply than the GSVD. The method reduces the matrix pair $\{A, L\}$ to a pair of bidiagonal and diagonal matrices. It is described in Section 2, where also reduction methods presented by Eldén [9, 10] are reviewed. Operation counts for these reduction methods are compared. Section 3 discusses the construction of useful regularization matrices L that are the product of two singular matrices, and for which the computation of $L^\dagger$ is expensive. Our reduction method is well suited for these regularization matrices. Computed examples with these regularization matrices are presented in Section 4. Concluding remarks can be found in Section 5.

2. Reduction methods. We first review application of the A-weighted pseudoinverse of L, given by (1.10), to the reduction of the problem (1.4) to standard form. This is described by Eldén [10] as well as in [15, Section 2.3]. We discuss flop counts for the situation when $A, L \in \mathbb{R}^{n\times n}$ and $L^\dagger$ cannot be computed cheaply. Subsequently, we describe a new approach to reducing (1.4) based on a GSVD-like reduction of the matrix pair $\{A, L\}$, that is cheaper to compute than the GSVD and also is cheaper to apply than $L_A^\dagger$ when L has linearly dependent columns and no exploitable structure.

2.1. Reduction based on $L_A^\dagger$. Eldén [10] showed that the problem (1.4) is equivalent to the minimization problem in standard form,

$$\min_{\boldsymbol{y}\in\mathbb{R}^n} \left\| \begin{bmatrix} AL_A^\dagger \\ \mu I \end{bmatrix} \boldsymbol{y} - \begin{bmatrix} \bar{\boldsymbol{b}} \\ \boldsymbol{0} \end{bmatrix} \right\|, \tag{2.1}$$

where $\bar{\boldsymbol{b}} = P_{A\mathcal{N}(L)}^\perp \boldsymbol{b}$ and $P_{A\mathcal{N}(L)}^\perp$ is the orthogonal projector onto the complement of $A\mathcal{N}(L)$. Let $\boldsymbol{y}_\mu$ denote the solution of (2.1). Then the solution $\boldsymbol{x}_\mu$ of (1.4) can be computed from $\boldsymbol{y}_\mu$ via

$$\boldsymbol{x}_\mu = L_A^\dagger \boldsymbol{y}_\mu + \boldsymbol{x}_0, \tag{2.2}$$

where $\boldsymbol{x}_0 = (AP_{\mathcal{N}(L)})^\dagger \boldsymbol{b}$. When $\mathcal{N}(L)$ is of small dimension and has an explicitly known basis, $\boldsymbol{x}_0$ is inexpensive to compute. This is the case for the matrix (1.5). Eldén [10] shows that

$$\|A\boldsymbol{x}_\mu - \boldsymbol{b}\| = \|AL_A^\dagger \boldsymbol{y}_\mu - \bar{\boldsymbol{b}}\|. \tag{2.3}$$

Therefore, $\phi(\mu)$ can be evaluated using the transformed problem (2.1).

We investigate the cost of transforming the minimization problem (1.4) to (2.1) when the structure of $L \in \mathbb{R}^{p\times n}$, $p \le n$, does not allow efficient evaluation of $L^\dagger$. The main computational cost then is the computation of $L^\dagger$. Let

$$L = \widetilde{U}\Big[\widetilde{\Sigma}, \underbrace{\boldsymbol{0}, \boldsymbol{0}, \ldots, \boldsymbol{0}}_{n-p}\Big]\widetilde{V}^T \tag{2.4}$$

be the singular value decomposition (SVD) of L with $\widetilde{U} \in \mathbb{R}^{p\times p}$ and $\widetilde{V} \in \mathbb{R}^{n\times n}$ orthogonal matrices and

$$\widetilde{\Sigma} = \mathrm{diag}[\widetilde{\sigma}_1, \widetilde{\sigma}_2, \ldots, \widetilde{\sigma}_p] \in \mathbb{R}^{p\times p}.$$

The singular values $\widetilde{\sigma}_j$ are ordered according to

$$\widetilde{\sigma}_1 \geq \widetilde{\sigma}_2 \geq \ldots \geq \widetilde{\sigma}_{p-\ell} > \widetilde{\sigma}_{p-\ell+1} = \ldots = \widetilde{\sigma}_p = 0.$$

The number of vanishing singular values, ℓ, is assumed to be small and independent of p and n in our flop counts. It is clear from (1.4) that L can be replaced by

$$\widetilde{L} := \widetilde{U}^T L = \widetilde{\Sigma}\widetilde{V}^T$$

without changing the solution of the problem. Therefore, only the matrices $\widetilde{\Sigma}$ and $\widetilde{V}$ in the SVD of L have to be computed. Their computation requires about $12n^3$ flops when $p \approx n$; see [12, p. 254].

We turn to the computation of the matrix $AL_A^\dagger$. Split $\widetilde{V} = [\widetilde{\boldsymbol{v}}_1, \widetilde{\boldsymbol{v}}_2, \ldots, \widetilde{\boldsymbol{v}}_n]$ into

$$\widetilde{V}_1 = [\widetilde{\boldsymbol{v}}_1, \widetilde{\boldsymbol{v}}_2, \ldots, \widetilde{\boldsymbol{v}}_{p-\ell}], \qquad \widetilde{V}_0 = [\widetilde{\boldsymbol{v}}_{p-\ell+1}, \widetilde{\boldsymbol{v}}_{p-\ell+2}, \ldots, \widetilde{\boldsymbol{v}}_n].$$

Thus, the columns of $\widetilde{V}_0$ form an orthonormal basis for $\mathcal{N}(L)$. Compute the QR factorization

$$A\widetilde{V}_0 = \widetilde{Q}\widetilde{R},$$

where the matrix $\widetilde{Q} \in \mathbb{R}^{m\times(n+\ell-p)}$ has orthonormal columns and the matrix $\widetilde{R} \in \mathbb{R}^{(n+\ell-p)\times(n+\ell-p)}$ is upper triangular. It follows from (1.6) that $\widetilde{R}$ is nonsingular. Since

$$A(I - L^\dagger L) = A\widetilde{V}_0\widetilde{V}_0^T = \widetilde{Q}\widetilde{R}\widetilde{V}_0^T,$$

we have

$$(A(I - L^\dagger L))^\dagger = \widetilde{V}_0\widetilde{R}^{-1}\widetilde{Q}^T.$$

Substitution this expression into (1.10) and replacing the trailing matrix $L^\dagger$ by $\widetilde{L}^\dagger$ yields

$$\begin{aligned} L_A^\dagger &= (I - \widetilde{V}_0\widetilde{R}^{-1}\widetilde{Q}^T A)\widetilde{V}\widetilde{\Sigma}^\dagger, \\ AL_A^\dagger &= (I - \widetilde{Q}\widetilde{Q}^T)A\widetilde{V}\widetilde{\Sigma}^\dagger. \end{aligned}$$

The dominating work for forming $AL_A^\dagger$ is, besides the computation of the matrices $\widetilde{\Sigma}$ and $\widetilde{V}$, the evaluation of the matrix-matrix product $A\widetilde{V}$, which requires $2n^3$ flops when $m \approx n$. We note that the product $(I - \widetilde{Q}\widetilde{Q}^T)(A\widetilde{V}\widetilde{\Sigma}^\dagger)$ can be evaluated in only $\mathcal{O}(n^2)$ flops since the number of columns of $\widetilde{Q}$ is independent of n and p (and is small). Thus, the flop count for determining $AL_A^\dagger$ is about $14n^3$. In addition, Householder bidiagonalization of $AL_A^\dagger$, without explicitly forming products of the Householder matrices involved, requires about $2\frac{2}{3}n^3$ flops, see [12, p. 252], giving a total count of about $16\frac{2}{3}n^3$ flops.

The reduction of the matrix in Tikhonov regularization problems in standard form to bidiagonal form, instead of to diagonal form via the SVD, was first proposed by Eldén [9], where further details on the computations can be found. The determination of $AL_A^\dagger$ is suggested in [10]. Thus, the outlined scheme combines the approaches in [9, 10]. The reduction described in [9] requires more computational effort when adapted to the situation of the present paper.

We remark that when the computations are carried out in MATLAB using the function svd, all matrices in the SVD of L (2.4) are computed. This increases the operation count for computing the SVD of L to about $22n^3$ flops, see [12, p. 254], and the total number of flops required by the reduction method to about $26\frac{2}{3}n^3$, which is 60% larger than for the bidiagonalization method described above.

Having reduced $AL_A^\dagger$ to bidiagonal form, the function $\phi(\mu)$ can be evaluated in only $\mathcal{O}(n)$ flops for each value of μ by using the relation (2.3). Having determined a zero $\mu > 0$ of $\phi(\mu)$, we compute $\boldsymbol{x}_\mu$ from (2.2). The latter computations can be carried out in $\mathcal{O}(n^2)$ flops if $\mathcal{N}(L)$ is of small dimension (independent of n) and has an explicitly known basis. It follows that the dominating cost for the solution of (1.4) by the method of this subsection is the $16\frac{2}{3}n^3$ flops required for the formation and reduction of $AL_A^\dagger$.

2.2. A GSVD-like reduction. This subsections describes an approach for reducing the matrix pair $\{A, L\}$ to a pair of simpler matrices that is cheaper than computing the GSVD and is well suited for use in Tikhonov regularization. Our initial computations are identical with those of several algorithms for computing the GSVD of $\{A, L\}$. A thorough discussion of such algorithms is provided by Bai [1]; see also [2]. All flop counts are for the case when $m = n = p$.

The first step of many GSVD algorithms is to compute the QR factorization

$$\begin{bmatrix} A \\ L \end{bmatrix} = QR, \quad Q = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}, \quad Q_1 \in \mathbb{R}^{m\times n}, \quad Q_2 \in \mathbb{R}^{p\times n}, \quad R \in \mathbb{R}^{n\times n}, \tag{2.5}$$

where Q has orthonormal columns and R is upper triangular. It follows from (1.6) that R is nonsingular. The computation of the QR factorization with Householder matrices requires about $3\frac{1}{3}n^3$ flops; see [12, Section 5.2.1]. This yields Q in factored form. We will need elements of the matrix Q_1, but not of Q_2. Given Q in factored form, the entries of Q_1 can be determined in about $3\frac{1}{3}n^3$ flops by computing them from bottom up.

Many GSVD algorithms proceed by determining the SVD of Q_1. We will instead reduce Q_1 to bidiagonal by using Householder matrices, because this is cheaper. Let

$$Q_1 = U_1 B_1 V_1^T,$$

where $B_1 \in \mathbb{R}^{m\times n}$ is upper bidiagonal, and $U_1 \in \mathbb{R}^{m\times m}$ and $V_1 \in \mathbb{R}^{n\times n}$ are orthogonal. The evaluation of this decomposition, with U_1 and V_1 represented in factored form, can be carried out in $2\frac{2}{3}n^3$ flops when $m \approx n$; see [12, Section 5.4.3]. We obtain the decomposition

$$Q = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} = \begin{bmatrix} U_1 & 0 \\ 0 & I_p \end{bmatrix} \begin{bmatrix} B_1 \\ Q_2 V_1 \end{bmatrix} V_1^T, \tag{2.6}$$

where I_p denotes the $p \times p$ identity matrix. The computation of the decomposition (2.6) requires about $9\frac{1}{3}n^3$ flops (for the evaluation of the QR factorization (2.5), the formation of the matrix Q_1, and Householder bidiagonalization of Q_1). This is the dominating work for the reduction method of this subsection.

The orthonormality of the columns of Q yields

$$I_n = B_1^T B_1 + V_1^T Q_2^T Q_2 V_1. \tag{2.7}$$

Substituting (2.5) and (2.6) into (1.4) gives the least-squares problem

$$\min_{\boldsymbol{y}\in\mathbb{R}^n} \left\| \begin{bmatrix} B_1 \\ \mu Q_2 V_1 \end{bmatrix} \boldsymbol{y} - \begin{bmatrix} U_1^T \boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix} \right\|, \quad \boldsymbol{y} = V_1^T R \boldsymbol{x}.$$

Using (2.7), the associated normal equations can be expressed as

$$\left(\mu^2 I_n + (1-\mu^2) B_1^T B_1\right) \boldsymbol{y} = B_1^T U_1^T \boldsymbol{b}. \tag{2.8}$$

It is generally advantageous, because it may yield a computed approximate solution of higher accuracy, to solve a least-squares problem rather than the associated normal equations. We therefore calculate the solution $\boldsymbol{y}_\mu$ of (2.8) by solving the least-squares problem

$$\min_{\boldsymbol{y}\in\mathbb{R}^n} \left\| \begin{bmatrix} \delta B_1 \\ \mu I_n \end{bmatrix} \boldsymbol{y} - \begin{bmatrix} \frac{1}{\delta} U_1^T \boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix} \right\|, \quad \delta := \sqrt{1-\mu^2}. \tag{2.9}$$

Eldén [9] described an algorithm for solving this kind of least-squares problem in $\mathcal{O}(n)$ flops by applying a judiciously chosen sequence of Givens rotations. In typical applications, the desired value of the regularization parameter μ is much smaller than unity. If this is not the case for the problem at hand, then (1.1) can be rescaled to achieve this. Therefore, δ is about unity and can be computed with high relative accuracy. We remark that the matrix U_1 does not have to be stored; the factors of U_1^T can be applied to $\boldsymbol{b}$ as soon as they are computed during bidiagonalization of Q_1.

The function (1.9) can be expressed as

$$\phi(\mu) = \|B_1 \boldsymbol{y}_\mu - U_1^T \boldsymbol{b}\|^2 - \eta^2 \varepsilon^2, \tag{2.10}$$

where $\boldsymbol{y}_\mu$ solves (2.9). The evaluation of $\phi(\mu)$ requires the solution of (2.9). Therefore the evaluation for each value of $\mu > 0$ can be carried out in only $\mathcal{O}(n)$ flops. Having determined a zero of (2.10), and the associated solution $\boldsymbol{y}_\mu$ of (2.9), the corresponding solution of (1.4) is given by

$$\boldsymbol{x}_\mu = R^{-1} V_1 \boldsymbol{y}_\mu. \tag{2.11}$$

We conclude this section with a discussion of some properties of the function (2.10) that are of interest when choosing a zero-finder. In particular, we will see that it may be more convenient to compute the zero of the function

$$\psi(\nu) := \phi(1/\sqrt{\nu}) - \eta^2 \varepsilon^2 \tag{2.12}$$

than of ϕ.

PROPOSITION 2.1. *Assume that $B_1^T U_1^T \boldsymbol{b} \neq \boldsymbol{0}$. Then the function (2.12) is strictly decreasing and convex for $\nu \geq 1$. With a suitable scaling of (1.1), the function ψ has a unique zero, which is larger than unity.*

Proof. The normal equations (2.8) yield

$$\boldsymbol{y}_\mu = (\mu^2 I_n + (1-\mu^2) B_1^T B_1)^{-1} B_1^T U_1^T \boldsymbol{b}.$$

Substituting this expression into (2.10) and using the SVD of B_1 shows that ψ is convex and decreasing.

Multiplying A, $\boldsymbol{b}$, and μ in (1.4) by a positive scalar does not change the solution $\boldsymbol{x}_\mu$. By choosing this positive scalar to be arbitrarily small, we can secure that the regularization parameter $\mu > 0$ that satisfies (1.8) is arbitrarily small. Therefore, the zero of ψ can be made arbitrarily large by rescaling of (1.1). $\square$

The above proposition suggests that it may be convenient to apply the secant method to determine the zero $\breve{\nu}$ of ψ with initial iterates $1 \leq \nu_0 < \nu_1 < \breve{\nu}$. Successive iterates ν_j, $j = 2, 3, \ldots$, then converge monotonically to $\breve{\nu}$.

In summary, the computation of the matrices in (2.9) requires about $9\frac{1}{3}n^3$ flops, the evaluation of $\phi(\mu)$ can then be done in only $\mathcal{O}(n)$ flops for each value of μ, and having determined a value that satisfies the discrepancy principle, we can compute the associated solution (2.11) in $\mathcal{O}(n^2)$ flops.

We would like to emphasize that the flop counts of this and the previous subsections are for the situation when L does not have an exploitable structure and does not have an explicitly known null space of low dimension. The following section provides examples of such regularization matrices. When L is banded with a small bandwidth and has an explicitly known null space of low dimension, it may be possible to speed up the computations significantly. For instance, when L is given by (1.5), the matrix $AL_A^\dagger$ can be formed in only $\mathcal{O}(n^2)$ flops.

3. Regularization matrices. The matrix A in discrete ill-posed problems (1.1) typically has many "tiny" singular values. The problems then are numerically underdetermined. The purpose of the regularization matrix L is to define a least-squares problem (1.4) that for $\mu > 0$ that satisfies (1.8) has a solution $\boldsymbol{x}_\mu$ that is close to the solution $\hat{\boldsymbol{x}}$ of (1.3). In particular, L should be chosen so that important features of $\hat{\boldsymbol{x}}$ are not damped. One way to achieve this is to choose a regularization matrix L with suitable $\mathcal{N}(L)$, because components of $\boldsymbol{x}_\mu$ in $\mathcal{N}(L)$ are not damped by L. We remark that an appropriate choice of L may enhance the quality of the computed approximation of $\hat{\boldsymbol{x}}$ significantly; see Section 4 as well as [3, 7, 16, 18] for examples.

The following proposition shows how "designer regularization matrices," whose null space contains a desired subspace, can be constructed from standard regularization matrices such as (1.5).

PROPOSITION 3.1. *Let the matrix* $W = [\boldsymbol{w}_1, \boldsymbol{w}_2, \dots, \boldsymbol{w}_\ell] \in \mathbb{R}^{n\times\ell}$, $1 \leq \ell < n$, *have orthonormal columns. Let* $L' \in \mathbb{R}^{p\times n}$ *and define*

$$L := L'(I - WW^T). \tag{3.1}$$

Then

$$\mathcal{N}(L) = \mathcal{R}(W) \cup \mathcal{N}(L'_{\mathcal{R}(W)^\perp}), \tag{3.2}$$

where $\mathcal{R}(W)$ *denotes the range of* W *and* $L'_{\mathcal{R}(W)^\perp}$ *is the restriction of* L' *to the orthogonal complement of* $\mathcal{R}(W)$.

Proof. The expression (3.2) is an immediate consequence of the definition (3.1). □

We remark that generally there are no simple useful expressions for the Moore-Penrose pseudoinverse of the product of two singular matrices, such as (3.1); see, e.g., [8, Section 1.4] for a discussion.

THEOREM 3.2. *Let the matrices* L' *and* W *be defined as in Proposition 3.1. Then (3.1) is the matrix closest to* L' *in the Frobenius norm with* $\mathcal{R}(W) \subset \mathcal{N}(L)$.

Proof. In order for the vector $\boldsymbol{v} \in \mathbb{R}^n$ to live in the null space of a matrix $M \in \mathbb{R}^{p\times n}$, each row of M has to be orthogonal to $\boldsymbol{v}$. Let $\boldsymbol{r}^T$ be a row of the matrix L'. The orthogonal projector $P_W^\perp := I - WW^T$ maps $\boldsymbol{r}^T$ to the closest row that is orthogonal to $\mathcal{R}(W)$, i.e.,

$$\|\boldsymbol{r}^T P_W^\perp - \boldsymbol{r}^T\| = \min_{\substack{\tilde{\boldsymbol{r}}^T W = \boldsymbol{0}^T \\ \tilde{\boldsymbol{r}} \in \mathbb{R}^n}} \|\tilde{\boldsymbol{r}}^T - \boldsymbol{r}^T\|.$$

Application to every row of L' shows that

$$\|L'P_W^\perp - L'\|_F = \min_{\substack{MW=0\\ M\in\mathbb{R}^{p\times n}}} \|M - L'\|_F,$$

where $\|N\|_F := \sqrt{\sum_{i,j} n_{i,j}^2}$ denotes the Frobenius norm of the matrix $N = [n_{i,j}] \in \mathbb{R}^{p\times n}$. $\square$

Example 3.1. Let n be even and assume that the desired solution $\hat{\boldsymbol{x}}$ of (1.3) is known to have large slowly oscillatory and linear components. Let L' be the regularization matrix (1.5) and define the discretization of the function $\gamma\cos(t)$ on $[-\pi,\pi]$,

$$\boldsymbol{w} := \gamma[\cos(-t_{n/2}), \cos(-t_{n/2-1}), \ldots, \cos(-t_1), \cos(t_1), \ldots, \cos(t_{n/2-1}), \cos(t_{n/2})]^T,$$

where $t_j = \frac{2j-1}{n}\pi$, $j = \pm 1, \pm 2, \ldots, \pm n/2$, and the constant γ is chosen so that $\boldsymbol{w}$ is of unit length. Let L be given by (3.1) with $W := \boldsymbol{w}$. Then $\mathcal{N}(L) = \mathcal{N}(L') \cup \mathrm{span}\{\boldsymbol{w}\}$. The performance of the regularization matrix L is illustrated in Section 4. The role of the matrix (1.5) is to provide damping of high-frequency components of the computed solution without damping the linear component; we have

$$\mathcal{N}(L') = \mathrm{span}\left\{ \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \\ \vdots \\ n \end{bmatrix} \right\}.$$

The factor $I - WW^T$ avoids damping of a slowly oscillatory component. $\square$

4. Computed examples. All computations were carried out on a desktop computer using MATLAB with about 16 significant decimal digits. We first comment on the relevance of the flop counts reported in Sections 1 and 2 for the different solution methods. The leading term of the flop counts were determined under the assumption that the matrices A and L are of size about $n \times n$. We recall that the leading terms for the methods of Subsections 2.1 and 2.2 are $16\frac{2}{3}n^3$ and $9\frac{1}{3}n^3$, respectively, and for the GSVD generally at least $35.3n^3$. These terms dominate the flop count when n is sufficiently large. It is interesting to investigate whether it is possible to choose n large enough on a standard desktop computer so that the leading terms of the flop counts, indeed, dominate. To shed light on this question, we timed the computation of the GSVD of matrix pairs $\{A, L\}$ with $A, L \in \mathbb{R}^{n\times n}$ for $n = 500, 1000, 1500, \ldots, 4000$, using the MATLAB function `gsvd` and the MATLAB timing commands `tic` and `toc`. We found that i) the required execution time as a function of n can be approximated well by a cubic monomial, and ii) the computations were rapid enough to be feasible. We therefore expect the method of Subsection 2.2, whose flop count has the smallest leading coefficient, to be competitive for practical problems of moderately large size.

We remark that the execution time not only is affected by the flop count, but also by the programming style, whether the code is interpreted or a compiled version is executed, the computer architecture, in particular communication speed and the amount of fast computer memory available, as well as the number of CPUs. Since we presently do not have carefully designed MATLAB codes available for all methods in our comparison, we refrain from reporting timings for the methods described in Section 2.

The following example compares the computed approximate solutions of (1.1) determined by Tikhonov regularization with the regularization matrices of Example 3.1,

Regularization matrix	$\|\boldsymbol{x}_\mu - \hat{\boldsymbol{x}}\|/\|\hat{\boldsymbol{x}}\|$
Example 3.1	$3.26 \cdot 10^{-3}$
(1.5)	$8.76 \cdot 10^{-3}$
I	$2.02 \cdot 10^{-2}$

TABLE 4.1

Example 4.1. Errors in approximate solutions computed by Tikhonov regularization with several regularization matrices for the noise-level $\alpha = 1 \cdot 10^{-3}$.

(1.5), and $L = I$. The "noise-vector" $\boldsymbol{e}$ in (1.2) has normally distributed pseudorandom entries with mean zero and is normalized to correspond to a chosen noise-level

$$\alpha := \frac{\|\boldsymbol{e}\|}{\|\hat{\boldsymbol{b}}\|}.$$

Here $\hat{\boldsymbol{b}}$ denotes the noise-free right-hand side vector in (1.3).

Example 4.1. Consider the Fredholm integral equation of the first kind

$$\int_{-6}^{6} \kappa(t,s)x(s)ds = g(t), \qquad -6 \le t \le 6, \tag{4.1}$$

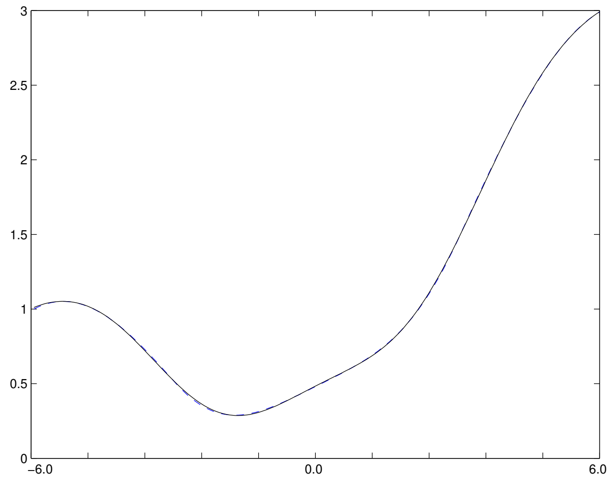
with the kernel and solution given by

$$\kappa(t,s) := x(t-s),$$
$$x(s) := \begin{cases} 1 + \cos(\frac{\pi}{3}s), & \text{if } |s| < 3, \\ 0, & \text{otherwise.} \end{cases}$$

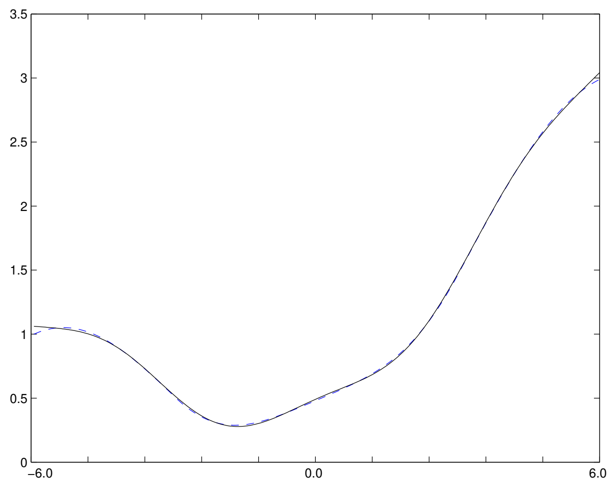
The right-hand side $g(t)$ is defined by (4.1). This integral equation is discussed by Phillips [17]. We use the MATLAB code phillips in [14] to determine a discretization $A \in \mathbb{R}^{200\times200}$ by a Galerkin method with orthonormal box functions. The code phillips also provides a discretization of a scaled solution, $\boldsymbol{x}_0 \in \mathbb{R}^{200}$. We add a discretization of the function $1 + s/6 + \cos(2\pi(1 + s/6))$, $-6 < s < 6$, to the vector $\boldsymbol{x}_0$ to obtain a slowly oscillatory and increasing solution $\hat{\boldsymbol{x}}$. The noise-free right-hand side is given by $\hat{\boldsymbol{b}} = A\hat{\boldsymbol{x}}$. We generate a noise-vectors $\boldsymbol{e}$ corresponding to the noise-level $\alpha = 1 \cdot 10^{-3}$ and obtain the contaminated vector $\boldsymbol{b}$ in (1.1) from (1.2). The regularization parameter μ is determined by the discrepancy principle with the coefficient η in (1.8) set to 1.01.

Table 4.1 shows the regularization matrix of Example 3.1 to yield the best approximations of $\hat{\boldsymbol{x}}$; the error achieved is smaller than for the other regularization matrices. Figure 4.1 displays the computed solutions determined with the regularization matrices of Table 4.1. The regularization matrix of Example 3.1 can be seen to yield better boundary behavior of the computed solution than the other regularization matrices. $\square$

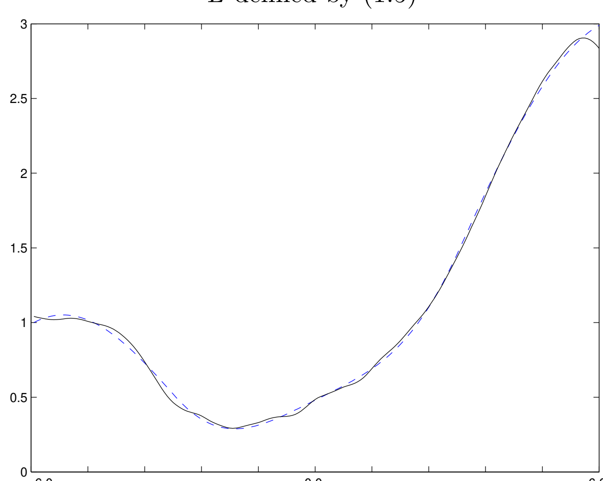
5. Conclusion and extension. The reduction of Tikhonov minimization problems (1.4) to simpler forms is investigated and a new approach that is well suited for the situation when it is expensive to determine the Moore-Penrose generalized inverse of the regularization matrix is proposed. This approach allows considerable flexibility in the choice of regularization matrix. Computations illustrate that this flexibility can help us determine regularization matrices that give more accurate approximations of the desired solution than standard regularization matrices, such as (1.5) and the identity matrix.



L defined in Example 3.1



L defined by (1.5)



$L = I$

FIG. 4.1. *Example 4.1: Approximate solutions* $\boldsymbol{x}_\mu$ *(continuous graphs) computed by Tikhonov regularization with different regularization matrices and noise-level* $\alpha = 1 \cdot 10^{-3}$*. The dashed graphs depict the desired solution* $\hat{\boldsymbol{x}}$*.*

Acknowledgement. We would like to thank a referee for comments.

REFERENCES

[1] Z. Bai, *The CSD, GSVD, their applications and computation*, IMA preprint 958, Institute for Mathematics and its Applications, University of Minnesota, Minneapolis, MN, 1992.

[2] Z. Bai and J. W. Demmel, *Computing the generalized singular value decomposition*, SIAM J. Sci. Comput., 14 (1993), pp. 1464–1486.

[3] C. Brezinski, M. Redivo-Zaglia, G. Rodriguez, and S. Seatzu, *Extrapolation techniques for ill-conditioned linear systems*, Numer. Math., 81 (1998), pp. 1–29.

[4] C. Brezinski, M. Redivo-Zaglia, G. Rodriguez, and S. Seatzu, *Multi-parameter regularization techniques for ill-conditioned linear systems*, Numer. Math., 94 (2003) 203–228.

[5] C. Brezinski, G. Rodriguez, and S. Seatzu, *Error estimates for linear systems with application to regularization*, Numer. Algorithms, 49 (2008), pp. 85–104.

[6] C. Brezinski, G. Rodriguez, and S. Seatzu, *Error estimates for the regularization of least squares problems*, Numer. Algorithms, 51 (2009), pp. 61–76.

[7] D. Calvetti, L. Reichel, and A. Shuibi, *Invertible smoothing preconditioners for linear discrete ill-posed problems*, Appl. Numer. Math., 54 (2005), pp. 135–149.

[8] S. L. Campbell and C. D. Meyer, *Generalized Inverses of Linear Transformations*, Dover, Mineola, 1991.

[9] L. Eldén, *Algorithms for the regularization of ill-conditioned least squares problems*, BIT, 17 (1977), pp. 134–145.

[10] L. Eldén, *A weighted pseudoinverse, generalized singular values, and constrained least squares problems*, BIT, 22 (1982), pp. 487–501.

[11] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of Inverse Problems*, Kluwer, Dordrecht, 1996.

[12] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, 1996.

[13] P. C. Hansen, *Regularization, GSVD and truncated GSVD*, BIT, 29 (1989), pp. 491–504.

[14] P. C. Hansen, *Regularization tools version 4.0 for Matlab 7.3*, Numer. Algorithms, 46 (2007), pp. 189–194.

[15] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems*, SIAM, Philadelphia, 1998.

[16] S. Morigi, L. Reichel, and F. Sgallari, *Orthogonal projection regularization operators*, Numer. Algorithms, 44 (2007), pp. 99–114.

[17] D. L. Phillips, *A technique for the numerical solution of certain integral equations of the first kind*, J. ACM, 9 (1962), pp. 84–97.

[18] L. Reichel and Q. Ye, *Simple square smoothing regularization operators*. Electron. Trans. Numer. Anal., 33 (2009), pp. 63–83.