

## EFFICIENT ALGORITHMS FOR SOLUTION OF REGULARIZED TOTAL LEAST SQUARES\*

ROSEMARY A. RENAUT<sup>†</sup> AND HONGBIN GUO<sup>†</sup>

**Abstract.** Error-contaminated systems  $Ax \approx b$ , for which  $A$  is ill-conditioned, are considered. Such systems may be solved using Tikhonov-like regularized total least squares (RTLS) methods. Golub, Hansen, and O’Leary [*SIAM J. Matrix Anal. Appl.*, 21 (1999), pp. 185–194] presented a parameter-dependent direct algorithm for the solution of the augmented Lagrange formulation for the RTLS problem, and Sima, Van Huffel, and Golub [A regularized total least squares algorithm, S. Van Huffel and P. Lemmerling, eds. Kluwer Academic Publishing, Dordrecht, 2002] have introduced a technique for solution based on a quadratic eigenvalue problem, (RTLSQEP). Guo and Renaut [Regularized total least squares based on quadratic eigenvalue problem solvers, Stanford University, Tech. Report Stanford, CA 2003] derived an eigenproblem for the RTLS which can be solved using the iterative inverse power method. Here we present an alternative derivation of the eigenproblem for constrained TLS through the augmented Lagrangian for the constrained normalized residual. This extends the analysis of the eigenproblem and leads to derivation of more efficient algorithms compared to the original formulation. Additional algorithms based on bisection search and a standard L-curve approach are presented. These algorithms vary with respect to the parameters that need to be prescribed. Numerical and convergence results supporting the different versions and contrasting with RTLSQEP are presented.

**Key words.** total least squares, regularization, ill-posedness, Rayleigh quotient iteration

**AMS subject classifications.** 65F22, 65F30

**DOI.** 10.1137/S0895479802419889

**1. Introduction.** We consider the solution of the ill-posed model dependent problem

$$Ax \approx b,$$

where  $A \in R^{m \times n}$  and  $b \in R^m$  are known, and are assumed to be error contaminated. If the matrix  $A$  is well conditioned a solution can be found using the method of total least squares (TLS)

$$(1.1) \quad \min \| [E, f] \|_F \quad \text{subject to} \quad (A + E)x = b + f,$$

where  $\| \cdot \|_F$  denotes the Frobenius norm, [4, 5, 12].

For ill-conditioned systems, Golub, Hansen, and O’Leary [3] presented and analyzed the properties of regularization for total least squares (TLS). Consistent with the formulation of the Tikhonov regularized LS problem [15, 16], the regularized TLS (RTLS) is given by

$$(1.2) \quad \min \| [E, f] \|_F \quad \text{subject to} \quad (A + E)x = b + f \quad \text{and} \quad \| Lx \| \leq \delta.$$

---

\*Received by the editors December 13, 2002; accepted for publication (in revised form) by P. C. Hansen January 29, 2004; published electronically DATE. This research was supported in part by the Arizona Alzheimer’s Disease Research Center which is funded by the Arizona Department of Health Services, and NIH grant EB 2553301. The first author is also supported by NSF grant CMG-02223 and acknowledges the support of the Technical University of Munich through the award of the John von Neumann visiting professorship in 2001–2002.

<http://www.siam.org/journals/simax/x-x/41988.html>

<sup>†</sup>Department of Mathematics and Statistics, Arizona State University, Tempe, AZ 85287-1804 (renaut@asu.edu, hb\_guo@asu.edu).

Here  $\|\cdot\|$  denotes the 2-norm,  $\delta$  is a regularization parameter, and  $L \in R^{p \times n}$  defines a (semi)norm on the solution [9, 3].

Guo and Renaut [6] obtained the solution of the RTLS problem by finding the minimum eigenpair for an augmented solution-dependent block matrix. The eigenpair is found iteratively, using inverse iteration applied to the solution-dependent matrix. Here we present a theoretical development of a convergent algorithm for determination of the minimum eigenpair. The algorithm is extended to improve efficiency by inclusion of the option to do inexact local solves and update of the constraint within the matrix formulation. Bisection search is presented because of the ability to determine precisely the number of iterations required for a given accuracy. We also provide an L-curve approach for cases in which a good estimate of the physical constraint parameter is not available.

Theoretical results which lead to the development of the algorithms are presented in section 2. The theory uses the alternative statement of the RTLS problem, namely the minimization of the normalized residual, equivalently the minimization of the Rayleigh quotient for the augmented matrix

$$(1.3) \quad M = [A, b]^T [A, b]$$

[12] subject to the addition of the constraint term for regularization of the solution. We also present results on the relationships between the Lagrange multipliers of the RTLS and the constraint parameter  $\delta$ . If any one of the set of three parameters is chosen as a free parameter, the other two are immediately specified and are monotonically related to one another. This result verifies the connection between the presented algorithms. Computational details are described in section 3, and experimental results comparing and contrasting the different approaches and comparing with the quadratic eigenvalue solver presented in [14] are described in section 4. We conclude that the eigenproblem formulation provides a powerful approach for regularized TLS solutions in practical applications.

## 2. Algorithmic development.

**2.1. Rayleigh quotient formulation.** It is well known that the solution of the TLS minimizes the sum of squared normalized residuals,

$$(2.1) \quad x_{TLS} = \operatorname{argmin}_x \phi(x) = \operatorname{argmin}_x \frac{\|Ax - b\|^2}{1 + \|x\|^2}$$

[5, 12], where  $\phi$  is the Rayleigh quotient of matrix  $M$ . This suggests an alternative formulation for regularized TLS,

$$(2.2) \quad \min_x \phi(x) \quad \text{subject to } \|Lx\| \leq \delta.$$

To distinguish the two formulations we call this the regularized Rayleigh quotient form for total least squares (RQ-RTLS). It leads to the augmented Lagrangian

$$(2.3) \quad \mathcal{L}(x, \mu) = \phi(x) + \mu(\|Lx\|^2 - \delta^2).$$

Although  $\phi(x)$  is not concave its stationary points can be characterized, which is useful in characterization of the solution of (2.3).

**LEMMA 2.1** (Fact. 1.8 in [13]). *The Rayleigh quotient of a symmetric matrix is stationary at, and only at, the eigenvectors of the matrix.*

LEMMA 2.2. *If the extreme singular values of the matrix  $[A, b]$  are simple, then  $\phi(x)$  has one unique maximum point, one unique minimum point, and  $n - 1$  saddle points.*

*Proof.* The proof follows by the observation  $\phi(x_i) = \sigma_i^2$ , where vector  $[x_i^T, -1]^T$  is the  $i$ th right singular vector of matrix  $[A, b]$  with corresponding singular value  $\sigma_i$ . The uniqueness of the maximum and minimum points is immediate. To show that all the other stationary points are saddles, it is easy to construct their neighbors and show that  $\phi(x)$  is, resp., greater and less on either side of the corresponding stationary point.  $\square$

THEOREM 2.1. *Suppose that the conditions of Lemma 2.2 are satisfied for matrix  $[A, b]$ , that  $\sigma_n > \sigma_{n+1}$  and that constraint parameter  $\delta$  is specified. Then, if the constraint is active,  $\|Lx_{RTLS}\|^2 = \delta^2$  and  $\mu > 0$ .*

*Proof.* By Lemma 2.2 and using  $\sigma_n > \sigma_{n+1}$ , the solution  $x_{TLS}$  of (2.1) is unique. If the constraint in (2.2) is active  $\|Lx_{TLS}\|^2 > \delta^2$ , and by Lemma 2.2 there is no local minimum of  $\phi$  within the set defined by the constraint  $\|Lx\|^2 < \delta^2$ . Thus, if the constraint is active,  $x_{RTLS}$  must lie on the boundary of the domain defined by  $\|Lx\|^2 \leq \delta^2$ ,

$$(2.4) \quad x_{RTLS}^T L^T L x_{RTLS} - \delta^2 = 0,$$

and the Lagrange parameter at the minimum of the Lagrangian is positive,  $\mu > 0$ .  $\square$

It is easy to see that the Kuhn–Tucker conditions for (2.2) are the same as those for (1.2). Hence we immediately obtain the following theorem for the characterization of the RTLS solution for (2.2), equivalent to that presented in [3] for the augmented Lagrangian for (1.2).

THEOREM 2.2. *The solution,  $x_{RTLS}$ , of the regularized problem (2.2), for which the constraint is active, satisfies*

$$(2.5) \quad (A^T A + \lambda_I I + \lambda_L L^T L) x_{RTLS} = A^T b,$$

$$(2.6) \quad \mu > 0, \quad x_{RTLS}^T L^T L x_{RTLS} - \delta^2 = 0,$$

where

$$(2.7) \quad \lambda_I = -\phi(x_{RTLS}),$$

$$(2.8) \quad \lambda_L = \mu(1 + \|x_{RTLS}\|^2),$$

$$(2.9) \quad \mu = -\frac{1}{\delta^2(1 + \|x_{RTLS}\|^2)} (b^T (Ax_{RTLS} - b) + \phi(x_{RTLS})).$$

*Proof.* Setting  $\nabla_x \mathcal{L}(x, \mu) = 0$ , we have

$$A^T Ax - A^T b + \mu(1 + \|x\|^2) L^T L x - \phi(x)x = 0,$$

which is (2.5) with  $\lambda_I$  and  $\lambda_L$  identified by (2.7) and (2.8), resp. Multiplying both sides by  $x^T$ , replacing  $\|Lx\|$  by  $\delta$ , and using the relationship (2.1) to rewrite  $\|Ax\|^2 - b^T Ax$  as  $(1 + \|x\|^2)\phi(x) + b^T (Ax - b)$ , we immediately obtain the expression for  $\mu$ . Moreover, (2.6) follows from Theorem 2.1.  $\square$

In [6] we observed, without proof, that this result additionally characterizes the RTLS solution in terms of an eigenpair for an augmented system. Here we present a slight modification of the result, of significant practical use, which includes the constraint condition in an alternative manner.

THEOREM 2.3. *The solution  $x_{RTLS}$  of (2.2) subject to the active constraint (2.4) satisfies the augmented eigenpair problem:*

$$(2.10) \quad B(x_{RTLS}) \begin{pmatrix} x_{RTLS} \\ -1 \end{pmatrix} = -\lambda_I \begin{pmatrix} x_{RTLS} \\ -1 \end{pmatrix},$$

where the solution-dependent matrix is given by

$$(2.11) \quad B(x_{RTLS}) = M + \lambda_L(x_{RTLS}) \begin{pmatrix} L^T L & 0 \\ 0 & \alpha \end{pmatrix}, \quad \alpha = -\|Lx_{RTLS}\|^2,$$

in which  $\lambda_L(x_{RTLS})$  is given by (2.8) and (2.9).

Conversely, suppose the pair  $((\hat{x}^T, -1)^T, -\hat{\lambda})$  is an eigenpair for matrix  $\hat{B}(\hat{x})$ , where matrix  $\hat{B}$  represents the matrix  $B$  with modification in the lower right corner,  $\alpha$  replaced by  $\hat{\alpha}$ ,  $\hat{\alpha} = -\gamma$ , where  $\gamma$  can take values  $\delta^2$ , or  $\|L\hat{x}\|^2$ , and  $\lambda_L(\hat{x})$  is defined accordingly by (2.8) and (2.9). Then

1.  $\hat{x}$  satisfies (2.5),
2. the constraint is active, (2.4) is satisfied for  $\hat{x}$ , and
3. eigenvalue  $\hat{\lambda}$  is given by

$$\hat{\lambda} = -\phi(\hat{x}).$$

*Proof.* The first block equation of (2.10) comes immediately from (2.5). For the second block equation we note that by (2.7)

$$\lambda_I(1 + \|x\|^2) = -\|Ax - b\|^2,$$

but by (2.5)

$$\lambda_I\|x\|^2 = b^T Ax - \|Ax\|^2 - \lambda_L\|Lx\|^2.$$

Thus, by subtraction,

$$(2.12) \quad \lambda_I = b^T Ax + \lambda_L\|Lx\|^2 - b^T b,$$

as required. We replace  $\|Lx\|^2$  occurring in  $\alpha$  by  $\delta^2$  using the active constraint condition (2.4).

For the proof in the opposite direction, we suppose that the eigenpair  $((\hat{x}^T, -1)^T, -\hat{\lambda})$  satisfies the eigenvalue equation (2.10), with appropriate replacement of  $x_{RTLS}$  by  $\hat{x}$  and  $\lambda_I$  by  $\hat{\lambda}$ . The first block equation immediately gives (2.5). By the second block equation of the eigenvalue problem we have

$$(2.13) \quad \hat{\lambda} = b^T A\hat{x} - b^T b + \lambda_L(\hat{x})\gamma,$$

and by the inner product of the eigensystem equation with eigenvector  $(\hat{x}^T, -1)^T$  we have

$$(2.14) \quad \hat{\lambda} = -\frac{1}{\|\hat{x}\|^2 + 1}(\|A\hat{x} - b\|^2 + \lambda_L(\hat{x})(\|L\hat{x}\|^2 - \gamma)).$$

Equating these two expressions, collecting terms in  $\lambda_L$  and then using (2.8) and (2.9) we find

$$\lambda_L \left( \frac{\|L\hat{x}\|^2 - \gamma}{\|\hat{x}\|^2 + 1} + \gamma \right) = \lambda_L \delta^2.$$

Solving for  $\gamma$ , by using the fact that  $\lambda_L \neq 0$ , because it is proportional to  $\mu \neq 0$ , yields

$$(2.15) \quad \gamma = \frac{\delta^2(1 + \|\hat{x}\|^2) - \|L\hat{x}\|^2}{\|\hat{x}\|^2},$$

which is satisfied for  $\gamma = \delta^2$ , or  $\gamma = \|L\hat{x}\|^2$ , each of which also imposes the active constraint equation (2.4). When inserted back into the second expression for  $\hat{\lambda}$  this also yields

$$\hat{\lambda} = -\frac{\|A\hat{x} - b\|^2}{\|\hat{x}\|^2 + 1} - \lambda_L(\hat{x}) \frac{\|L\hat{x}\|^2 - \delta^2}{\|\hat{x}\|^2},$$

where now the second term vanishes because both of the choices for  $\gamma$  also enforce the active constraint condition. Hence  $\hat{\lambda}(\hat{x}) = \lambda_I(\hat{x})$ , as required.  $\square$

**2.2. Theoretical development.** By the definition of RTLS problem (2.2) and Theorem 2.3, the RTLS solution can be obtained through estimation of the minimum  $|\lambda_I| = \phi(x)$  which solves the augmented eigenvalue problem. Whenever the system (2.10) is satisfied, the active constraint condition is also immediately satisfied. To derive practical algorithms for the solution of the eigenproblem, we observe the similarity with the unconstrained TLS problem:  $(x_{TLS}^T, -1)^T$  is a right eigenvector for matrix  $M$  associated with its smallest eigenvalue. An algorithm based on the Rayleigh quotient iteration (RQI) for matrix (1.3) was presented by Björck, Heggernes, and Matstoms [2]. While a similar iterative approach can be implemented, there is the additional complication that the system matrix (2.11) depends on the solution  $x_{RTLS}$ , which requires consideration of the convergence properties applied to this particular situation. On the other hand, in [6], we verified numerically that inverse iteration can be used for the determination of the RTLS solution. Here we investigate the convergence properties of the approach and introduce modifications of the algorithm to improve efficiency and reliability.

To analyze the eigenproblem for (2.10), for the case in which we use  $\gamma = \delta^2$ , we introduce the parameter-dependent matrix,  $\mathbf{B}(\theta) = M + \theta N$ ,  $\theta \in R^+$ , where

$$(2.16) \quad N = \begin{pmatrix} L^T L & 0 \\ 0 & -\delta^2 \end{pmatrix}.$$

Obviously  $\mathbf{B}(\lambda_L) = B(x)$  in (2.11) if  $\lambda_L$  is given by (2.8) with  $x$  in place of  $x_{RTLS}$ . We denote the smallest eigenvalue corresponding to eigenvector  $(x_\theta^T, -1)^T$  of  $\mathbf{B}(\theta)$  by  $\varrho_{n+1}$ , and use the notation  $\mathcal{N}(A)$  for the null space of matrix  $A$ .

We also introduce the function  $g(x) = (\|Lx\|^2 - \delta^2)/(1 + \|x\|^2)$ . Then the goal of solving the augmented eigenproblem may be reformulated as follows.

**PROBLEM 2.4.** *For a constant  $\delta$ , find a  $\theta$  such that  $g(x_\theta) = 0$ .*

The following results assist with the design of an algorithm to solve this problem.

**LEMMA 2.3.** *Assuming that  $b^T A \neq 0$  and  $\mathcal{N}(A) \cap \mathcal{N}(L) = \{0\}$ , then the smallest eigenvalue of  $\mathbf{B}(\theta)$  is simple.*

*Proof.* The eigenvalue-eigenvector equation

$$\mathbf{B}(\theta) \begin{pmatrix} x_\theta \\ -1 \end{pmatrix} = \rho_\theta \begin{pmatrix} x_\theta \\ -1 \end{pmatrix},$$

yields

$$(2.17) \quad (A^T A + \theta L^T L - \rho_\theta I)x_\theta = A^T b.$$

By assumption  $A^T b \neq 0$ , so  $\rho_\theta$  is not an eigenvalue of  $A^T A + \theta L^T L$ . By the eigenvalue interlace theorem, it is thus strictly smaller than the smallest eigenvalue of  $A^T A + \theta L^T L$  and must be simple.  $\square$

LEMMA 2.4. *If  $[A, b]$  is a full rank matrix, there exists one and only one positive number, denoted by  $\theta^c$ , such that  $\mathbf{B}(\theta^c)$  is singular, and*

1. *the null eigenvalue of  $\mathbf{B}(\theta^c)$  is simple*
2. *when  $0 \leq \theta < \theta^c$ ,  $\mathbf{B}(\theta)$  is positive definite, and*
3. *when  $\theta > \theta^c$ ,  $\mathbf{B}(\theta)$  has only one negative eigenvalue, the others are positive.*

*Proof.* Because  $M$  is nonsingular,  $\mathbf{B}(\theta) = M + \theta N$  is congruent to  $C(\theta) = I + \theta X^T N X$ , where  $X$  is a nonsingular matrix. Thus  $\mathbf{B}(\theta)$  and  $C(\theta)$  have the same inertia, as do  $N$  and  $X^T N X$ . Because  $L^T L$  is nonnegative definite, we know  $C(\theta)$  is similar to

$$(2.18) \quad I + \theta \begin{pmatrix} D & & \\ & 0 & \\ & & -\omega^2 \end{pmatrix},$$

where  $D$  is a diagonal matrix with positive diagonal entries. Thus there exists only one finite real number  $\theta = \theta^c > 0$  such that the null space of  $\mathbf{B}(\theta)$  is nontrivial and the dimension of the corresponding null space is 1.

Because matrices (2.18) and  $\mathbf{B}(\theta)$  have the same inertia, we immediately obtain the other two results.  $\square$

LEMMA 2.5. *If  $b^T A \neq 0$ , and  $[A, b]$  is full rank, then*

1. *there exists a  $\lambda_L^* \in [0, \theta^c]$  which solves Problem 2.4,*
2. *the solution of Problem 2.4 is unique,*
3. *when  $\lambda_L \in (0, \lambda_L^*)$ ,  $g(x_{\lambda_L}) > 0$  and  $\lambda_L \in (\lambda_L^*, \infty)$ ,  $g(x_{\lambda_L}) < 0$ .*

*Proof.*

1. When  $\theta = 0$ ,  $\mathbf{B}(0) > 0$ . The eigenvector corresponding to the smallest eigenvalue of  $\mathbf{B}(0)$ ,  $(M)$ , is related to the TLS solution  $x_{TLS}$ ,  $g(x_{TLS}) > 0$  because the constraint is active. Moreover, for small perturbation in the matrix  $\mathbf{B}(\theta)$ , Theorem 6.3.12 in [11] yields

$$(2.19) \quad \frac{d\rho_{n+1}}{d\theta}|_{\theta=\theta_0} = g(x_{\theta_0}).$$

Thus,  $\rho_{n+1}$  increases with  $\theta$  near zero. On the other hand, by Lemma 2.4  $\rho_{n+1} = 0$  for  $\theta = \theta^c$ . Thus,  $g(x_\theta)$  must change sign in  $[0, \theta^c]$  and by continuity there must exist a number  $\lambda_L^* \in [0, \theta^c]$  such that the corresponding  $g(x_{\lambda_L^*}) = 0$ . Hence Problem 2.4 is solved.

2. We introduce notation  $x_\theta$ ,

$$(2.20) \quad x_\theta = \operatorname{argmin}_{x \in R^n} (\phi(x) + \theta g(x)).$$

Clearly, by Lemma 2.3, the smallest eigenvalue of  $\mathbf{B}(\theta)$  is simple. Suppose that vectors  $x_{\theta_1}, x_{\theta_2}$  solve (2.20) for  $\theta_1, \theta_2 > 0$ , then

$$(2.21) \quad \phi(x_{\theta_2}) + \theta_2 g(x_{\theta_2}) < \phi(x_{\theta_1}) + \theta_2 g(x_{\theta_1}),$$

$$(2.22) \quad \phi(x_{\theta_1}) + \theta_1 g(x_{\theta_1}) < \phi(x_{\theta_2}) + \theta_1 g(x_{\theta_2}).$$

Adding these inequalities yields

$$(2.23) \quad (\theta_1 - \theta_2)g(x_{\theta_1}) < (\theta_1 - \theta_2)g(x_{\theta_2}),$$

and without loss of generality assuming  $\theta_1 > \theta_2$ ,  $g(x_{\theta_1}) < g(x_{\theta_2})$ . Thus,  $g(x_\theta)$  is monotonically decreasing with respect to  $\theta$  and there exists only one  $\theta$  such that  $g(x_\theta) = 0$ .

3. The final statement follows immediately from the former two.  $\square$

REMARK 2.1. *We see from this result that there is a unique solution to our problem and that an algorithm for finding this solution should depend both on finding an update for the Lagrange parameter  $\lambda_L$  and monitoring the sign of  $g(x_{\lambda_L})$ .*

From (2.13) it is immediate that  $x_\theta$  is related to  $\theta$  by  $\theta = \frac{1}{\delta^2}(b^T(b - Ax_\theta) - \phi(x_\theta))$ . This suggests an iterative search for the  $\theta$ ,

$$(2.24) \quad \theta^{(k+1)} = \frac{1}{\delta^2}(b^T(b - Ax_{\theta^{(k)}}) - \phi(x_{\theta^{(k)}})),$$

where, at step  $k$ ,  $(x_{\theta^{(k)}}^T, -1)^T$  is the eigenvector for  $\varrho_{n+1}^{(k)}$ . On the other hand, by (2.14), we can write  $\varrho_{n+1}^{(k)} = \phi(x_{\theta^{(k)}}) + \theta^{(k)}g(x_{\theta^{(k)}})$ , which in (2.24), also using  $b^T Ax_{\theta^{(k)}} - b^T b + \delta^2 \theta^{(k)} = -\varrho_{n+1}^{(k)}$ , gives an update equation

$$(2.25) \quad \theta^{(k+1)} = \theta^{(k)} + \frac{\theta^{(k)}}{\delta^2}g(x_{\theta^{(k)}}).$$

It remains to consider whether this iteration will generate the appropriate  $\theta$  that solves Problem 2.4. We investigate the convergence properties of the update equation (2.25), but first revert to the use of the parameter  $\lambda_L$  in place of  $\theta$ . The theory presented in Lemma 2.5 suggests the use of an iteration dependent parameter  $0 < \iota^{(k)} \leq 1$  chosen such that  $g(x_{\lambda_L^{(k+1)}})$  has the same sign as  $g(x_{\lambda_L^{(0)}})$ :

$$(2.26) \quad \lambda_L^{(k+1)} = \lambda_L^{(k)} + \iota^{(k)} \frac{\lambda_L^{(k)}}{\delta^2} g(x_{\lambda_L^{(k)}}), \quad 0 < \iota^{(k)} \leq 1.$$

LEMMA 2.6. *Suppose  $\lambda_L^{(0)} > 0$ . Let sequence  $\{x_{\lambda_L^{(k)}}\}$  and  $\{\lambda_L^{(k)}\}$ ,  $k = 1, 2, \dots$ , be generated by (2.26), with parameter sequence  $0 < \iota^{(k)} \leq 1$  utilized to enforce  $g(x_{\lambda_L^{(k+1)}})g(x_{\lambda_L^{(0)}}) > 0$ .*

1.  $\lambda_L^{(k)} > 0$  for any positive integer  $k$ .
2. If  $g(x_{\lambda_L^{(0)}}) < 0$ , then sequences  $\{\lambda_L^{(k)}\}$  and  $\{\phi(x_{\lambda_L^{(k)}})\}$  decrease monotonically, while  $\{\varrho_{n+1}^{(k)}\}$  and  $\{g(x_{\lambda_L^{(k)}})\}$  increase monotonically.
3. If  $g(x_{\lambda_L^{(0)}}) > 0$ , then sequences  $\{\lambda_L^{(k)}\}$  and  $\{\phi(x_{\lambda_L^{(k)}})\}$  increase monotonically, while  $\{\varrho_{n+1}^{(k)}\}$  and  $\{g(x_{\lambda_L^{(k)}})\}$  decrease monotonically.
4. If  $g(x_{\lambda_L^{(0)}}) = 0$ ,  $\lambda_L^{(0)}$  solves Problem 2.4.

*Proof.* For ease of presentation we write  $x^{(k)}$  to indicate  $x_{\lambda_L^{(k)}}$ , assuming the dependence of the update on the  $\lambda_L^{(k)}$ .

1. Multiplying both sides of (2.26) by  $\delta^2(1 + \|x^{(k)}\|^2)$ , we obtain

$$\delta^2(1 + \|x^{(k)}\|^2)\lambda_L^{(k+1)} = \delta^2(1 - \iota^{(k)} + \|x^{(k)}\|^2)\lambda_L^{(k)} + \iota^{(k)}\lambda_L^{(k)}\|Lx^{(k)}\|^2.$$

Because  $\iota^{(k)} \leq 1$ ,  $\lambda_L^{(k+1)} > 0$  if  $\lambda_L^{(k)} > 0$ . Thus  $\lambda_L^{(0)} > 0$  ensures  $\lambda_L^{(k)} > 0$  for all  $k \geq 1$ .

2. If  $g(x^{(0)}) < 0$ , the algorithm forces  $g(x^{(k)}) < 0$  for  $k > 1$  so that also  $\lambda_L^{(k+1)} < \lambda_L^{(k)}$ . Then by (2.23)  $g(x^{(k)}) < g(x^{(k+1)}) < 0$  and combining with (2.21)  $\phi(x^{(k+1)}) < \phi(x^{(k)})$ . Moreover, because the Rayleigh–Ritz theorem also gives  $\varrho_{n+1}^{(k)} = \min_{x \in R^n} (\phi(x) + \lambda_L^{(k)} g(x))$  we have

$$\begin{aligned} \varrho_{n+1}^{(k)} &= \phi(x^{(k)}) + \lambda_L^{(k)} g(x^{(k)}) \\ &< \phi(x^{(k+1)}) + \lambda_L^{(k)} g(x^{(k+1)}) \\ &< \phi(x^{(k+1)}) + \lambda_L^{(k+1)} g(x^{(k+1)}) \\ &= \varrho_{n+1}^{(k+1)}. \end{aligned}$$

3. The proof of this case follows equivalently.

4. This is immediate from the definition of Problem 2.4.  $\square$

REMARK 2.2. For an initial  $0 < \lambda_L^{(0)} < \theta^c$ , the tendency of the generated monotonic sequence for  $\lambda_L^{(k)}$  depends on whether  $\lambda_L^{(0)} < \lambda_L^*$  or  $\lambda_L^{(0)} > \lambda_L^*$ , but in either case  $B(\lambda_L^{(k)})$  is always positive definite.

THEOREM 2.5. The iteration (2.26) with an initial  $\lambda_L^{(0)} > 0$  converges to the unique solution,  $\lambda_L^*$ , of Problem 2.4.

*Proof.* By Lemma 2.6  $\{\lambda_L^{(k)}\}$  is monotonic and by Lemma 2.5, statement 3, it is bounded by  $\lambda_L^*$ . Thus it converges. Suppose that it converges to the limit point  $\tilde{\lambda}_L$ , then this limit point should satisfy (2.26) in the limit, and  $g(x_{\tilde{\lambda}_L}) = 0$ . But now Problem 2.4 has a unique solution and thus  $\tilde{\lambda}_L \equiv \lambda_L^*$ .  $\square$

**2.3. Algorithms.** The theoretical results justify the basic algorithm for the solution  $x_{RTLS}$  of (1.2) which uses exact determination of the smallest eigenvalue for each update of the Lagrange parameter  $\lambda_L$  with Rayleigh quotient iteration.

ALGORITHM 1 (EXACT RTLS: Alternating iteration on  $\lambda_L$  and  $x$ ). For given  $\delta$  and initial guess  $\lambda_L^{(0)} > 0$  calculate the eigenpair determined by  $(\varrho_{n+1}^{(0)}, x^{(0)})$ . Set  $k = 0$ . Update  $\lambda_L^{(k)}$  and  $x^{(k)}$  until convergence.

1. While not converged

**Do**

- (a)  $\iota^{(k)} = 1$

- (b) **Inner Iteration:** Until sign condition is satisfied **Do:**

- i. Update  $\lambda_L^{(k+1)}$  by (2.26).

- ii. Calculate the smallest eigenvalue,  $\varrho_{n+1}^{(k)}$ , and the corresponding eigenvector,  $[x^{(k+1)}, -1]$ , of matrix  $\mathbf{B}(\lambda_L^{(k)})$ .

- iii. If sign condition  $g(x^{(k+1)})g(x^{(0)}) > 0$  is not satisfied, set  $\iota^{(k)} = \iota^{(k)}/2$  else **Break**.

**End Do**

- (c) Test for convergence. If converged **Break** else  $k = k + 1$ .

**End Do.**

2.  $x_{RTLS} = x^{(k)}$ .

At the inner iteration in Algorithm 1 we find the minimum eigenvalue using an application of the approach presented in [2], based on cubically convergent Rayleigh quotient iteration for the constant matrix  $B(\lambda_L^{(k)})$ . Block Gaussian elimination is used to improve the efficiency. Specifically, for fixed  $\lambda_L$  we iterate over  $j$  such that at



iteration  $j$  we wish to find the vector  $y^{(k,j+1)} = ((x^{(k,j+1)})^T, -1)^T$  such that

$$(2.27) \quad \mathbf{B}(\lambda_L^{(k)})y^{(k,j+1)} = \beta_{(k,j)}y^{(k,j)},$$

$$(2.28) \quad \mathbf{B}(\lambda_L^{(k)}) = \begin{pmatrix} J^{(k,j)} & A^T b \\ b^T A & \eta_{(k,j)} \end{pmatrix},$$

$$(2.29) \quad J^{(k,j)} = A^T A + \lambda_L^{(k)} L^T L - \rho_{(k,j)} I_n \quad \eta_{(k,j)} = b^T b - \lambda_L^{(k)} \delta^2 - \rho_{(k,j)},$$

where  $\rho_{(k,j)}$  is the RQI shift. Here we use the double index  $(k, j)$  to indicate that the inner iteration to find the eigenvalue is over index  $j$  as compared to the outer iteration for  $\lambda_L$  which is over  $k$ . Having made this distinction, we now assume the dependence on  $k$  whenever iteration  $j$  is denoted. We suppose that the matrix  $J^{(j)}$  is positive definite, certainly the case without shift by assumption on the initial choice of  $\lambda_L^{(0)}$ , so that we can apply block Gauss elimination

$$\begin{pmatrix} J^{(j)} & A^T b \\ 0 & \tau_j \end{pmatrix} \begin{pmatrix} x^{(j+1)} \\ -1 \end{pmatrix} = \beta_j \begin{pmatrix} x^{(j)} \\ -(z^{(j)})^T x^{(j)} - 1 \end{pmatrix},$$

where

$$(2.30) \quad \tau_j = \eta_j - b^T A z^{(j)}$$

and  $x^{(j+1)} = z^{(j)} + \beta_j u^{(j)}$ . Here  $z^{(j)}$  and  $u^{(j)}$  solve the systems

$$(2.31) \quad J^{(j)} z^{(j)} = A^T b,$$

$$(2.32) \quad J^{(j)} u^{(j)} = x^{(j)},$$

and the scaling parameter is given by

$$(2.33) \quad \beta_j = \tau_j / ((z^{(j)})^T x^{(j)} + 1).$$

**REMARK 2.3.** *The algorithm as presented to match the theoretical results requires precise determination of the smallest eigenvalue for each  $\lambda_L$ . However, an inexact determination, particularly in early stages of the iteration, may increase efficiency by reducing the total number of iterations. Moreover, the key requirement of the convergence result is that the update  $x_{\theta(k+1)}$  is such that the sign property for function  $g$  is maintained. Thus, suppose that we do not solve the eigenproblem exactly for each  $\lambda_L^{(k)}$ , but that instead an approximate eigenpair is found,  $(\tilde{\rho}_{n+1}^{(k)}, \tilde{x}_{\theta(k+1)})$ , for which  $g(\tilde{x}_{\theta(k+1)})$  maintains the sign condition, then the iteration will still converge. This leads to modification of Algorithm 1 based on inexact update for the eigenvalue.*

**ALGORITHM 2 (INEXACT RTLS: Alternating iteration on  $\lambda_L$  and  $x$ ).** *Implement the exact algorithm but initialized with  $0 < \lambda_L^{(0)} < \theta^c$  chosen such that the initial matrix  $B(\lambda_L^{(0)})$  is positive definite. At each iteration do not search for the exact eigenpair for each  $k$ , rather use inverse iteration and seek satisfactory  $x^{(k,J_k)}$ , such that  $g(x^{(k,J_k)})g(x^{(0)}) > 0$ . If this condition is satisfied for  $j = J_k$ , assign  $x^{(k)} = x^{(k,J_k)}$  and update  $\lambda_L^{(k+1)}$ . The initial vector for each inner inverse iteration is  $x^{(k,0)} = x^{(k-1,J_{k-1})}$ .*

**REMARK 2.4.** *It is immediate to see from the convergence theory that if the requirement on the sign of  $g$  is relaxed a divergent sequence can result. It was this version of Algorithm 2 that was implemented in [6]. In particular, without the condition on the sign of  $g$ , each inner iteration to calculate  $x^{(k)}$  uses just one step,  $j = 1$ , and the matrix  $B(\lambda_L^{(k)})$  is updated each step.*

REMARK 2.5. In Algorithms 1 and 2 we assume that  $\gamma = \delta^2$ , where  $\lambda_L^{(0)}$  is required. While the theory does not immediately follow, these algorithms can be modified to use  $\gamma = \|Lx^{(k,j)}\|^2$ , where initial solution  $x^{(0)}$  or  $x^{(0,0)}$  is required. This modification introduces new versions of both algorithms, which we denote by 1.2 and 2.2, resp., reserving the notation 1.1 and 2.1 for the former versions. If blow-up does not occur, which we demonstrate through our numerical experiments is seldom the case, we find that the iteration converges much more quickly.

REMARK 2.6. While the exact determination of the smallest eigenvalue at any step will be made efficient if the shift of the Rayleigh quotient iteration is utilized, it should be clear that it is not, in general, desirable to use the shift when we pose the problem in the inexact form, for which we want to change  $\lambda_L$  efficiently to get to the RTLS solution, rather than to find each intermediate eigenvalue precisely. Thus, in general, given an initial choice of  $\lambda_L$  such that  $\mathbf{B}$  is positive definite, the block matrix  $J^{(j)}$  without shift is guaranteed positive definite.

**2.4. Interdependence of parameters.** In the preceding algorithms we assume that the physical parameter  $\delta$  is known a priori, which may not always be the case. Hence we need to understand the relationship between  $\delta$  and the other parameters  $\lambda_L$  and  $\lambda_I$  in order to determine an algorithm for which  $\delta$  is not provided.

Consistent with earlier notation, we distinguish the solution of the RTLS problem via the  $\delta$ -specified algorithm as  $x_\delta$ . Moreover, we use  $J(\lambda_L) = A^T A - \phi(x_{\lambda_L})I + \lambda_L L^T L$  and  $s(x_{\lambda_L}) = A^T A x_{\lambda_L} - A^T b - \phi(x_{\lambda_L})x_{\lambda_L}$  for which  $J(\lambda_L)x_{\lambda_L} = A^T b$  and  $s(x_{\lambda_L}) = -\lambda_L L^T L x_{\lambda_L}$ .

THEOREM 2.6. Suppose matrix  $J(\lambda_L)$  is positive definite and  $\lambda_L > 0$ ; then

1.  $\frac{d\phi(x_{\lambda_L})}{d\lambda_L} > 0$ ,  $\phi(x_{\lambda_L})$  is monotonically increasing with respect to  $\lambda_L$ , and
2.  $\frac{d(\|Lx_{\lambda_L}\|^2)}{d\lambda_L} < 0$ ,  $\|Lx_{\lambda_L}\|^2$  is monotonically decreasing with respect to  $\lambda_L$ .

*Proof.* Differentiating  $J(\lambda_L)x_{\lambda_L} = A^T b$  with respect to  $\lambda_L$  yields

$$J(\lambda_L) \frac{dx_{\lambda_L}}{d\lambda_L} = \left( \frac{d\phi(x_{\lambda_L})}{d\lambda_L} I - L^T L \right) x_{\lambda_L}.$$

Now

$$\begin{aligned} \frac{d\phi(x_{\lambda_L})}{d\lambda_L} &= (\nabla_{x_{\lambda_L}} \phi(x_{\lambda_L}))^T \frac{dx_{\lambda_L}}{d\lambda_L} \\ &= \frac{2}{1 + \|x_{\lambda_L}\|^2} s^T(x_{\lambda_L}) \frac{dx_{\lambda_L}}{d\lambda_L} \\ (2.34) \quad &= -\frac{2\lambda_L x_{\lambda_L}^T L^T L}{1 + \|x_{\lambda_L}\|^2} \frac{dx_{\lambda_L}}{d\lambda_L}. \end{aligned}$$

Rearranging yields

$$\frac{d\phi(x_{\lambda_L})}{d\lambda_L} \left( \frac{1 + \|x_{\lambda_L}\|^2}{2} \right) = \lambda_L x_{\lambda_L}^T L^T L \left( \frac{d\phi(x_{\lambda_L})}{d\lambda_L} J(\lambda_L)^{-1} x_{\lambda_L} - J(\lambda_L)^{-1} L^T L x_{\lambda_L} \right).$$

Hence

$$\frac{d\phi(x_{\lambda_L})}{d\lambda_L} \left( \frac{1 + \|x_{\lambda_L}\|^2}{2} + \lambda_L x_{\lambda_L}^T L^T L J(\lambda_L)^{-1} x_{\lambda_L} \right) = \lambda_L x_{\lambda_L}^T L^T L J(\lambda_L)^{-1} L^T L x_{\lambda_L} > 0$$

by assumptions on  $J(\lambda_L)$  and  $\lambda_L$ , and the first statement follows immediately.

On the other hand,

$$\frac{d(\|Lx_{\lambda_L}\|^2)}{d\lambda_L} = 2x_{\lambda_L}^T L^T L \frac{dx_{\lambda_L}}{d\lambda_L},$$

which, after substitution in (2.34), gives

$$\frac{d\phi(x_{\lambda_L})}{d\lambda_L} = -\frac{\lambda_L}{(1 + \|x_{\lambda_L}\|^2)} \frac{d(\|Lx_{\lambda_L}\|^2)}{d\lambda_L},$$

and the second statement follows.  $\square$

These results justify the introduction of alternative algorithms.

**2.4.1. Bisection search.** Because of the direct monotonic relationship between parameters  $\delta = \|Lx_{\lambda_L}\|$  and  $\lambda_L$ , we can use a standard bisection search technique on parameter  $\lambda_L$  to obtain an update mechanism for  $\lambda_L$ . With this approach the number of solves for each  $\lambda_L$  is determined by the precision required and the initial interval for bisection and is thus of most use in situations for which we know that the class of problems is difficult to solve. This gives the following algorithm, for which details are standard.

**ALGORITHM 3** (RTLS: Bisection search on  $\lambda_L$ ). *Given  $\delta$ , a search tolerance **TOL** on the active constraint,  $|\|Lx_{\lambda_L}\| - \delta| \leq \mathbf{TOL}$ , and two initial choices of  $\lambda_L$  for which  $g(x_{\lambda_L})$  are of different signs, do bisection until the tolerance is satisfied. At each iteration estimate solution  $\hat{x}_{\lambda_L}$  by Algorithm 2 with  $\gamma$  updated each step, namely Algorithm 2.2 except  $\lambda_L$  is fixed.*

**2.4.2. An L-curve algorithm.** The earlier algorithms assume a priori information to designate  $\delta$  which may not be available. We consider instead, then, an approach based on the use of the L-curve [8, 10] to give a  $\delta$ -independent algorithm for the formulation

$$(2.35) \quad \min_x \phi(x) + \mu \|Lx\|^2.$$

Here the positive regularization parameter  $\mu$  controls how much weight is given to the penalty function  $\|Lx\|^2$  as compared to the Rayleigh quotient  $\phi(x)$ . Necessary conditions for a minimum of (2.35) are the same as for (2.2) except for (2.9). If the constraint is active, the solution satisfies

$$(A^T A + \lambda_I I + \mu(1 + \|x\|^2)L^T L)x = A^T b,$$

where  $\lambda_I = -\phi(x)$ . Substituting  $\lambda_L = \mu(1 + \|x\|^2)$ , we once again obtain (2.5):

$$(A^T A + \lambda_I I + \lambda_L L^T L)x = A^T b.$$

For each fixed parameter  $\lambda_L$ , the solution  $x_{\lambda_L}$  is equivalent to the  $x_{RTLS}$  solution obtained with constraint parameter  $\delta = \|Lx_{\lambda_L}\|$ . Hence we need to determine  $\lambda_L$  so that it simultaneously gives a small Rayleigh quotient  $\phi(x_{\lambda_L})$  and a moderate value of the penalty term  $\|Lx_{\lambda_L}\|^2$ . We use the L-curve method which was designed for the Tikhonov regularized LS problem [8, 10] for the *log-log* scale plot of  $\phi(x_{\lambda_L})$  versus  $\|Lx_{\lambda_L}\|^2$ .

**ALGORITHM 4** (RTLS: L-curve). *Given a discrete set of values for  $\lambda_L$  on an interval  $[a, b]$ , find RTLS solutions  $x_{\lambda_L}$ . Generate the L-curve and pick the lower left corner point of the curve to generate  $x_{RTLS}$ .*

1. **For**  $\lambda_L$  over a discrete set.

**Inner iteration** For fixed  $\lambda_L$ , calculate RTLS solution  $x_{\lambda_L}$  by alternatively updating  $x_{\lambda_L}$  by solving (2.5) and  $\lambda_I$  through (2.7) until the inner iteration has converged.

**End For**

2. Plot on *log-log* scale the pairs  $\phi(x_{\lambda_L})$  versus  $\|Lx_{\lambda_L}\|^2$ .
3. Find the lower left corner point of the L-curve, the corresponding parameter  $\lambda_L$ , and solution  $x_{\lambda_L} = x_{RTLS}$ .

REMARK 2.7. To carry out the final step of the algorithm we could use “Algorithm FindCorner” in [10].

### 3. Computational considerations.

**3.1. Termination criteria.** In the inner iterations for the Rayleigh quotient or inverse iteration, where also the system matrix may depend on the current update if  $\gamma$  is updated each step, we test convergence on the residual  $r^{(j)} = B(x^{(j)})\bar{y}^{(j)} + \lambda_I(x^{(j)})\bar{y}^{(j)}$ , where  $\bar{y}^{(j)}$  is  $y^{(j)}$  normalized. It is easy to verify that

$$(B(x^{(j)}) + C^{(j)})\bar{y}^{(j)} = -\lambda_I(x^{(j)})\bar{y}^{(j)},$$

where  $C^{(j)} = -r^{(j)}(\bar{y}^{(j)})^T$ . Let  $\epsilon$  represent machine accuracy and  $c$  be a quite mild function of degree  $n+1$ , then the best accuracy we can expect to achieve is  $\|C^{(j)}\|/\|B(x^{(j)})\| \leq c\epsilon$  [17, Chap. 5, sect. 58]. Hence

$$\|C^{(j)}\| = \|r^{(j)}\| \leq c\epsilon\|B(x^{(j)})\| \approx c\epsilon\|A, b\|^2.$$

Since  $B(x^{(j)})$  is a symmetric matrix, the accuracy of  $\lambda_I(x^{(j)})$  is also approximately  $c\epsilon\|A, b\|^2$ . This suggests using  $|\lambda_I^{(j)} - \lambda_I^{(j-1)}|/|\lambda_I^{(j)}| < \mathbf{TOL}$  as stopping criterium, where  $\mathbf{TOL}$  is a tolerance.  $\frac{\|r^{(j)}\|}{|\lambda_I^{(j)}|} < \mathbf{TOL}$  may also be used as termination criterium.

When  $\delta$  is known we may directly use  $\|Lx^{(j)} - \delta\|$  as stopping criterium. Also  $\|r^{(j)}\|$  is a measurement of the distance of  $x^{(j)}$  to the boundary (2.4). In fact, by the Cauchy–Schwarz inequality,  $\|\bar{y}^{(j)}\| = 1$ , and using, from (2.9),  $\mu$  as a function of  $x$ ,

$$(3.1) \quad \begin{aligned} \|r^{(j)}\| &\geq |(\bar{y}^{(j)})^T (B(x^{(j)}) + \lambda_I(x^{(j)})I)\bar{y}^{(j)}| \\ &= |\mu(x^{(j)})(\|Lx^{(j)}\|^2 - \delta^2)|. \end{aligned}$$

Thus, the residual  $\|r^{(j)}\|$  also provides an upper estimate for the violation of the constraint condition (2.4) and, if  $\bar{y}^{(j)}$  is sufficiently close to an eigenvector of  $B(x^{(j)})$ , then the inequality in (3.1) is close to an equality. Since we solve the eigenproblem for  $B$  and need to find  $-\lambda_I$ , we would expect  $(B + \lambda_I^{(j)}I)\bar{y}^{(j)}$  becomes zero if  $(-\lambda_I^{(j)}, \bar{y}^{(j)})^T$  is an eigenpair for  $B$ .

**3.2. The generalized SVD (GSVD) of  $[A, L]$ .** All of the presented algorithms depend on the efficiency of solving systems with coefficient matrix  $A^T A + \lambda_L L^T L$ , or the shifted version  $A^T A + \lambda_L L^T L - \rho_k I$ . Here we focus on the derivation of an efficient algorithm for the solution of systems with system matrix,  $J$ , without shift. Notice that, without loss of generality, we drop the dependence on iteration  $(k, j)$ , and consider the solution of the system

$$(3.2) \quad (A^T A + \lambda_L L^T L)w = f.$$

While different approaches can be considered for (3.2), we also note the similarity of (3.2) with the system to be solved in Tikhonov regularization of the least squares

TABLE 3.1  
Algorithmic summary and comparison.

	Algorithm *.1	Algorithm *.2	Bisection	L-curve
$\delta$	given	given	given	unknown
$x^{(0)}$	random	required	required	required
$\lambda_L^{(0)}$	given	derived from $x^{(0)}$	derived from $x^{(0)}$	derived from $x^{(0)}$
subalg.	No	No	Algorithm 2.2	Algorithm 2.2

problem. Hence we should use the algorithms which have been demonstrated as successful for regularized LS. Moreover, we can safely assume that matrix  $L$ , which in our examples is a low order derivative operator, is well conditioned. In particular, the smallest nonzero singular values of the first and second order derivative operators are of order  $n^{-1}$  and  $n^{-2}$ , resp., and their null spaces are spanned by very *smooth* vectors. Thus, if  $\lambda_L$  is not too small, matrix  $A^T A + \lambda_L L^T L$  is well posed for a large class of matrices  $A$ , and the GSVD of the matrix pair  $[A, L]$  can be calculated with a stable numerical method [7]. This approach, used to solve (3.2) [7], also motivates use of algorithms without shift. Using the algorithm of Bai and Demmel [1], the calculation of the GSVD for matrix pair  $[A, L]$  requires  $2m^2n + 15n^3$  flops (the coefficient of  $n^3$  depends on the number of iterations required). Given the GSVD the solution of each equation (3.2) costs just  $8n^2$  flops.

**3.3. Summary of the algorithms.** In the preceding sections we have presented several different algorithmic approaches for the solution of the given RTLS problem. We now summarize these algorithms with respect to the initialization requirements and the subalgorithm that is used to solve an eigenproblem with fixed parameter  $\lambda_L$ . We list the requirements in Table 3.1, where Algorithm \*.1 and Algorithm \*.2 represent versions  $\gamma = \delta^2$  and  $\gamma = \|Lx^{(k)}\|^2$ , resp.

**4. Numerical experiments.** To test the given algorithms we mainly use test examples taken from Hansen's *Regularization Tools* [9]. Three functions, *ilaplace*, *phillips*, and *shaw*, are used to generate matrices  $A$ , right-hand sides  $b$ , and solutions  $x^\sharp$  so that  $Ax^\sharp = b$  is satisfied. In all cases, the data are scaled so that  $\|A\|_F = \|Ax^\sharp\|_2 = 1$ , and a 5% Gaussian perturbation is added to both coefficient matrix and right-hand side. For *ilaplace* and *shaw* matrix  $A$  has size  $65 \times 64$ , and for *phillips* the matrix is  $64 \times 64$ , [3, 9]. We let operator  $L \in R^{(n-1) \times n}$  approximate the first-derivative operator. For algorithms in which  $\delta$  is specified, we choose  $\delta = 0.9\|Lx^\sharp\|$ . In all tests we chose tolerance **TOL** =  $10^{-4}$ , and we denote the estimated solution of each algorithm by  $x_{est}$ .

In the results we report the relative error with respect to  $x^\sharp$ . On the other hand, we know the solutions should converge to  $x_{RTLS}$ , which is the solution of the equation subject to constraint. Thus we may expect that evaluation compared to  $x^\sharp$  is limited for a single test, and that it is the speed with which a converged solution satisfying the constraint is achieved, which is important. Thus in Test 4.3 we repeat tests over 100 perturbations for each experiment and report the average results for each case, except for the L-curve in which we present results of one sample perturbation. We measure the speed with respect to the numbers of system solves of type (3.2) that are required, hence providing a comparison between algorithms. To give the total cost of each test, we add the cost for the GSVD and the iterations, i.e.,  $2m^2n + 15n^3 + K \cdot 8n^2$  flops, where  $K$  is the number of solves and report the number of megaflops. In each case we initialize the iteration with  $\lambda_L^{(0)} = 0.1$ , and for Algorithm 2.2 with  $x^{(0,0)} = x_{RLS}$  obtained with regularization parameter  $\lambda = .001$ .

For the test of the L-curve algorithm we pick 20 equally spaced points, with respect to the log scale, on the interval  $[a, b] = [1.0e - 6, 0.1]$ . For any choice of  $\lambda_L$  we stop the inner iteration if convergence is not achieved in 15 steps. If the curve has a clear L-shape, 20 points are sufficient to identify the corner because more points are located near the corner than at other places on the curve.

*Test 4.1* (evaluation of inexact and exact algorithms). Here we demonstrate the impact of use of the exact solve by Algorithm 1 as compared to the inexact approach in Algorithm 2, in which for the inexact solve we search only for a new update which satisfies the sign condition. We find that primarily  $J_k = 1$ ; namely the inexact solve mostly uses one step of inverse iteration prior to update of parameter  $\lambda_L$ . The long-term and short-term convergence history for  $-\lambda_L^{(k)}$  is illustrated in Figure 4.1. In these tests we do not update  $\gamma$  occurring in  $\mathbf{B}$ , but fix  $\gamma = \delta^2$ . This test is thus a true comparison for the convergence theory for inexact solve in place of the exact solve. Clearly, the costs for the inexact solve are less than the total cost for determination of the exact eigenpair each outer iteration but the total impact depends on the algorithm used for the exact solve. Thus we do not report relative costs in each case. We observe that over the long term there is no detrimental impact on the convergence behavior, even though we see that at the early iterations the solutions obtained are not exactly the same. Clearly inexact solve produces an alternative update in the early steps without being deleterious for ultimate convergence.

*Test 4.2* (evaluation of inclusion of RQ shift). We now consider the impact of the use of the shift for improving the convergence of the inexact algorithm, Algorithm 2. In Figure 4.2 we show the lack of impact on the convergence of inclusion of the shift for the inner iteration of Algorithm 2. The top and bottom three figures are associated with Algorithms 2.1 and 2.2, resp. We illustrate three cases, the first without any shift, the second in which we shift at all steps, and the third in which, consistent with the RQI for the TLS problem introduced by [2], we shift after the first step. We

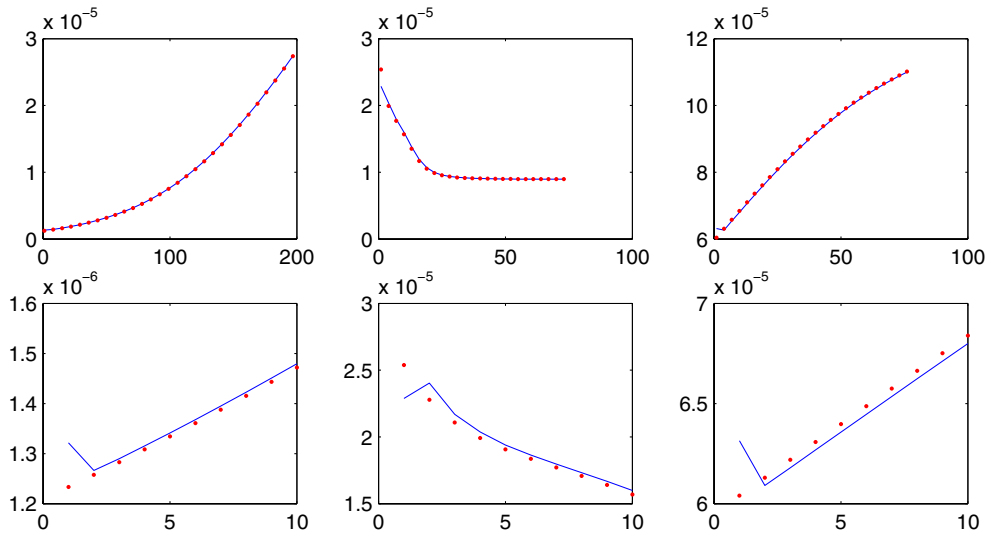


FIG. 4.1. The figures compare the convergence history for  $-\lambda_L^{(k)}$  for the exact compared to inexact algorithms. The dotted and dashed lines show the convergence for the exact and inexact algorithms, resp. The first row shows the whole convergence history while the second row shows the first 10 steps. From left to right, examples ilaplace, shaw, and phillips, resp.

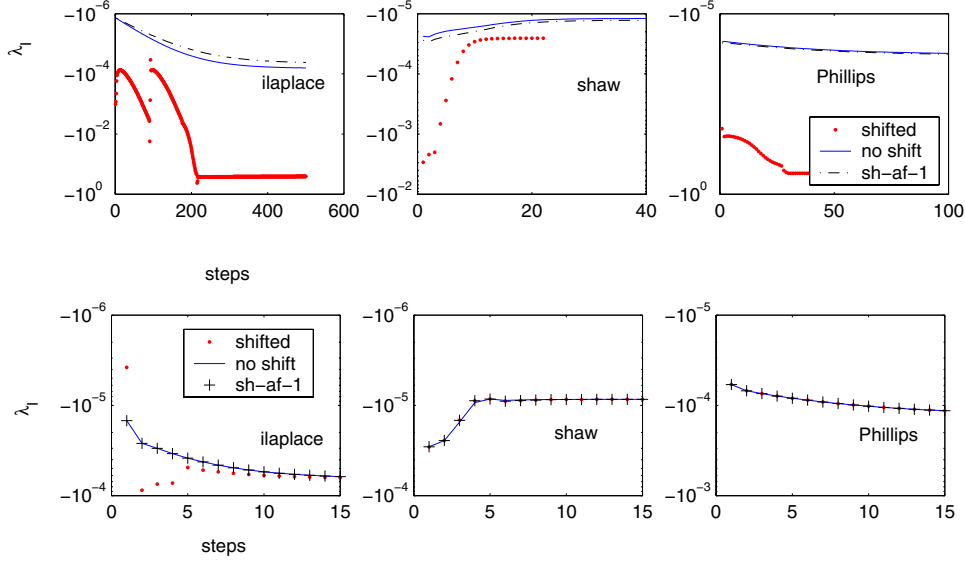


FIG. 4.2. The figures show the convergence history of  $-\lambda_I$  for Algorithms 2.1 and 2.2, top and bottom, resp., with the shift added at different stages in the iteration process, i.e., shifted for all steps (“shifted”), no shift at all (“no shift”), and the shift added after the first step (“sh-af-1”). From left to right, examples ilaplace, shaw, and phillips, resp.

note the different line type used for the case when the shift is added after the first step between the upper and lower figures. This is deliberate because, in addition, Algorithm 2.2 converges with far fewer iterations (compare the scales on the x-axes) and so the use of + also for Algorithm 2.1, which requires many iterations, would mask all the other results in these figures.

It is clear that adding the shift at every step can cause Algorithm 2.1 to converge to an eigenpair which is not the smallest as defined by the eigenvalue. This possibility exists because the algorithm is initialized with a random vector which then generates a bad initial RQ shift. This problem is avoided if the shift occurs only after one step of inverse iteration, and the results are almost the same as without shift for all steps, both approaches converge to the RTLS solution. On the other hand, Algorithm 2.2 always converges to the RTLS solution, and shift does not cause any significant difference in the convergence history of  $\lambda_I$  after the first few steps of the iteration.

In summary, adding the shift makes no positive contribution to the convergence, contrary to the case for RQI for the TLS problem. Moreover, with inclusion of the shift we cannot take advantage of the calculation of the GSVD for the augmented matrix  $[A, L]$ . Thus our results demonstrate no reason to include the shift in (2.29), which also further justifies our assumption that matrix  $J$  remains positive definite throughout the iteration.

*Test 4.3* (comparison of the algorithms). Here we emphasize the improvement due to setting  $\gamma = \|Lx^{(k,j)}\|^2$  in the right corner of  $B(x^{(k,j)})$  as compared with  $\gamma = \delta^2$ . Details of average results for all four algorithms, over 100 perturbations, are provided in Tables 4.1–4.3, and the solutions of one sample perturbation are illustrated in Figure 4.3. In the tables, the relative error reported is the average relative error of  $x_{est}$  to  $x^\#$ .

TABLE 4.1  
Average solutions for *ilaplace*, (Test 4.3).

Algorithm	$\lambda_L$	$\lambda_I$	K	cost mflops	relerr
1.1 $\gamma = \delta^2$	6.6382e-01	-5.6144e-05	799	NA	6.75e-02
1.2 $\gamma = \ Lx^{(k,j)}\ ^2$	6.6433e-01	-5.6228e-05	54.4	NA	6.75e-02
2.1 $\gamma = \delta^2$	6.6382e-01	-5.6144e-05	799	31.0	6.75e-02
2.2 $\gamma = \ Lx^{(k,j)}\ ^2$	6.6433e-01	-5.6228e-05	54.2	6.2	6.75e-02
Bisection	6.6231e-01	-5.5937e-05	100.6	7.8	6.79e-02
L-curve (sample)	5.6234e-4	-3.0242e-8	161	9.8	1.7470e-2

TABLE 4.2  
Average solutions for *shaw*, (Test 4.3).

Algorithm	$\lambda_L$	$\lambda_I$	K	cost mflops	relerr
1.1 $\gamma = \delta^2$	3.1329e-04	-9.9912e-06	98.1	NA	9.13e-02
1.2 $\gamma = \ Lx^{(k,j)}\ ^2$	3.1296e-04	-9.9907e-06	21.0	NA	9.12e-02
2.1 $\gamma = \delta^2$	3.1310e-04	-9.9895e-06	99.4	7.7	9.11e-02
2.2 $\gamma = \ Lx^{(k,j)}\ ^2$	2.9090e-04	-9.8089e-06	25.8	5.3	9.51e-02
Bisection	2.7939e-04	-1.0033e-05	81.5	7.1	9.46e-02
L-curve (sample)	1.7783e-4	-1.0094e-5	158	9.7	1.0478e-1

TABLE 4.3  
Average solutions for *phillips*, (Test 4.3).

Algorithm	$\lambda_L$	$\lambda_I$	K	cost mflops	relerr
1.1 $\gamma = \delta^2$	1.8454e-01	-1.3426e-04	368.6	NA	9.05e-02
1.2 $\gamma = \ Lx^{(k,j)}\ ^2$	1.8454e-01	-1.3426e-04	71.5	NA	9.05e-02
2.1 $\gamma = \delta^2$	1.8454e-01	-1.3426e-04	369.0	17.0	9.05e-02
2.2 $\gamma = \ Lx^{(k,j)}\ ^2$	1.8454e-01	-1.3426e-04	71.6	6.8	9.05e-02
Bisection	1.8502e-01	-1.3460e-04	66.3	6.6	9.05e-02
L-curve (sample)	5.6234e-4	-4.4807e-6	119	8.36	5.1365e-2

We note that the total numbers of outer iterations for Algorithm 1 and Algorithm 2 are comparable, thus again demonstrating the benefit of the use of the inexact solve for each outer iteration. Again we do not report the costs of the exact solve, denoted in the tables by NA, which depends on the chosen algorithm and is certainly not optimal if inverse iteration is used. Given the lack of benefit of the use of exact solve, we chose not to investigate the most efficient technique for its solution. In all cases, we see a dramatic decrease in the total number of steps required to reach convergence for Algorithm 2.2 as compared to Algorithm 2.1. While the solutions are different in all cases, because of the dependence on the specific converged value for  $\lambda_L$ , all solutions other than those obtained by the L-curve algorithm are qualitatively similar, see the figures on the left of Figure 4.3. We note that example *shaw* does not give a good L-shape and thus it is hard to determine the optimal  $\lambda_L$ .

*Test 4.4* (comparison with solution based on the quadratic eigenvalue problem (QEP) [14]). To compare the approach with that using the QEP we compare Algorithm 2.1 with the QEP again over 100 cases, each with the random 5% perturbation. For both algorithms we adopt the stopping rule  $\|x^{(k+1)} - x^{(k)}\|/\|x^{(k+1)}\| < \mathbf{TOL}$  used in [14], and the QEP program is written exactly as stated for **rtlsqep** in [14]. We use a random initial solution  $x^{(0)}$  and matrix-vector multiplication to avoid matrix multiplication. Algorithm 2.1 is initialized in each case with  $\lambda_L = 0.1$ . In some situations—for example, *ilaplace*—this generates an apparently *zero* cost solution. Actually this corresponds to a one step iteration to convergence because the



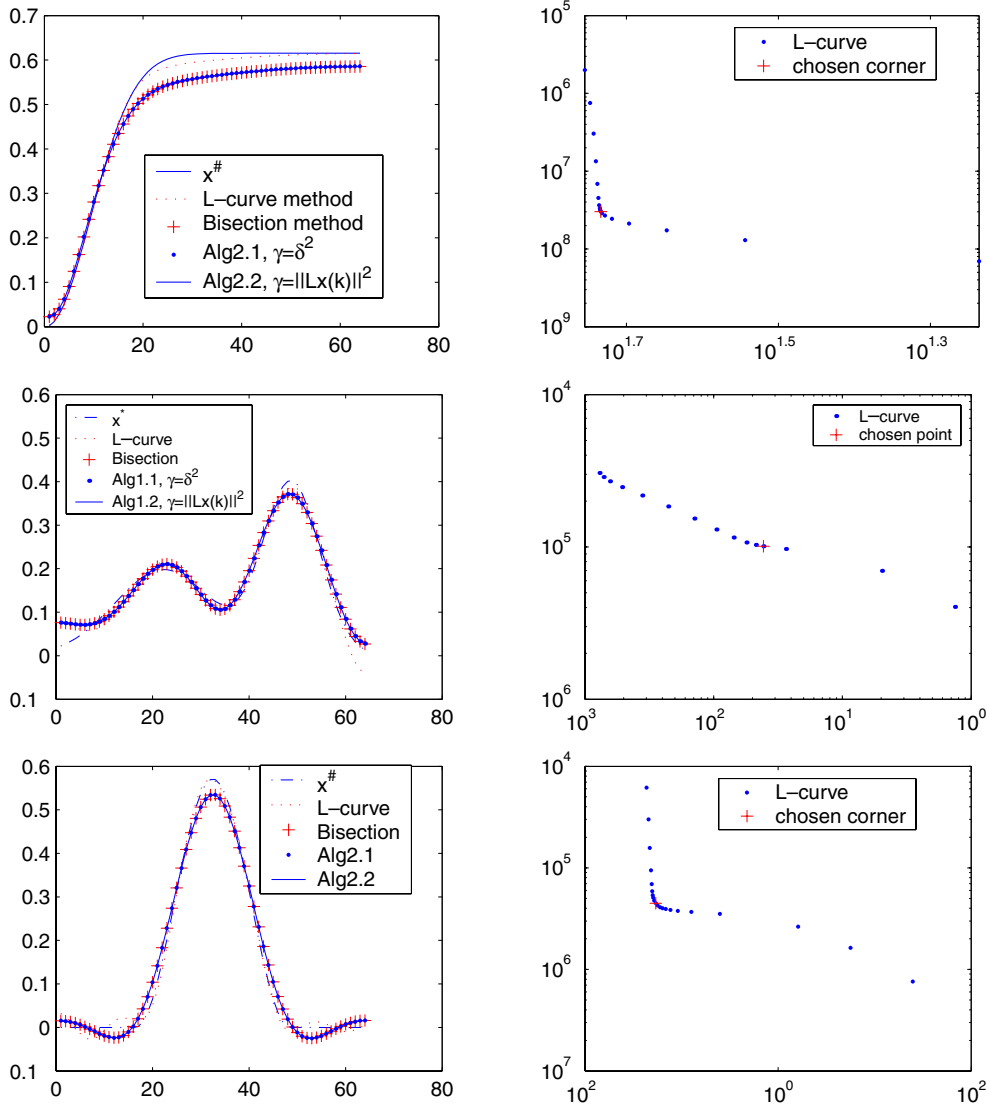


FIG. 4.3. From top to bottom, examples *ilaplace*, *phillips*, and *shaw*, resp. Solutions are indicated on the left and the L-curve on the right.

initialization  $\lambda_L = 0.1$  presents an almost perfect estimation to the regularization parameter  $\lambda_L$ , which would *never* occur for use of random  $x^{(0)}$  with QEP.

The results are summarized in Figure 4.4, in which the distribution of relative errors of  $x_{est}$  to  $x^\#$  (two top rows of figures),  $-\lambda_I(x_{est})$  (middle two rows of figures), and the CPU costs in seconds are shown (last two rows of figures). In each case the figures are organized with results for Algorithm 2.1 first, followed by those for QEP, and with, from left to right, examples *ilaplace*, *shaw*, and *phillips*, resp.

For *ilaplace* Algorithm 2.1 has generally smaller error but is a little more expensive than QEP, while the results with *shaw* are similar but Algorithm 2.1 is cheaper, and *phillips* outperforms the QEP in all measures.

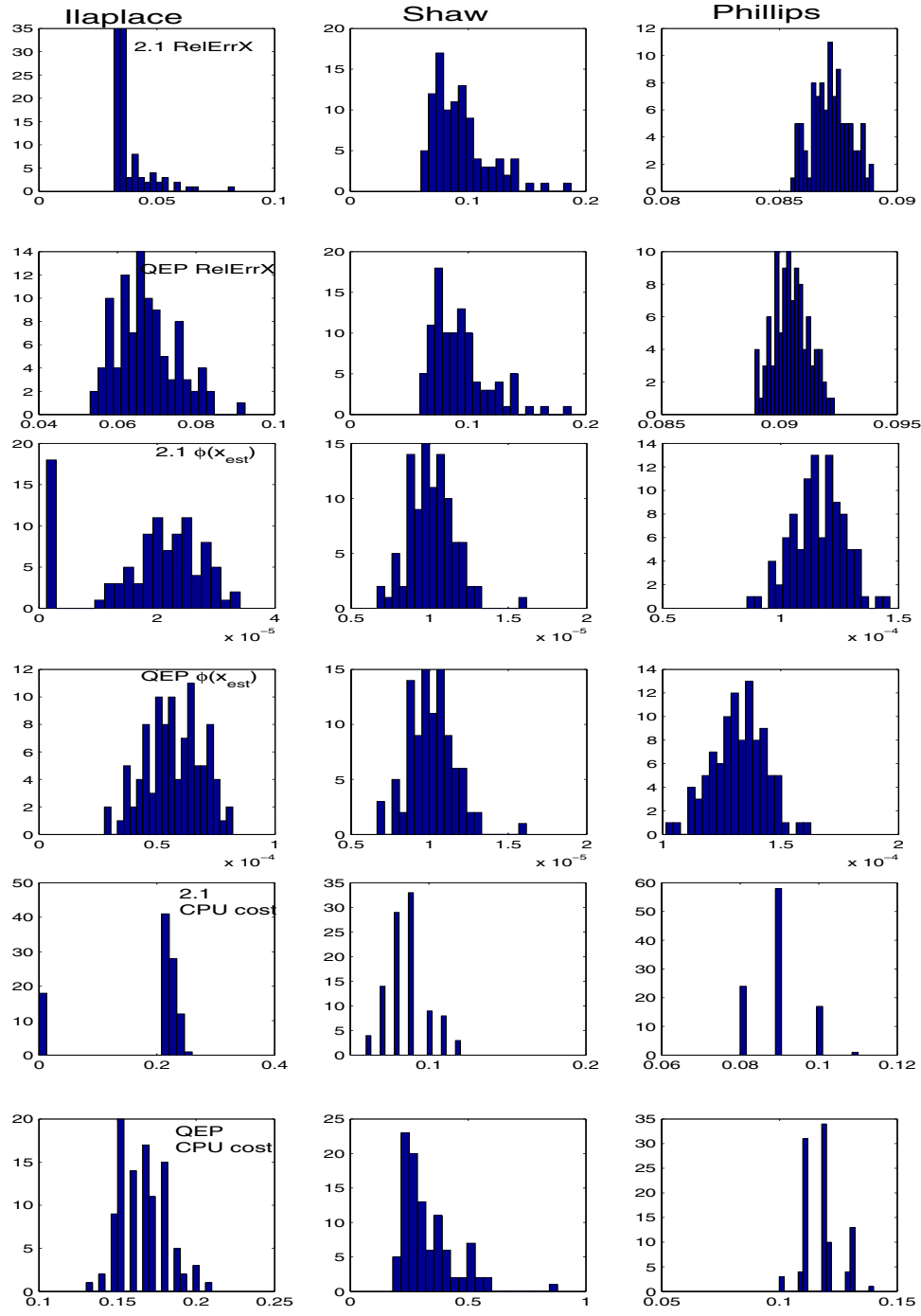


FIG. 4.4. Comparison of Algorithm 2.1 and the QEP algorithm, Test 4.4.

**5. Conclusions.** We have demonstrated new algorithms for the solution of the regularized total least squares problem. These algorithms employ the relationship between the RTLS solution and the eigensolution of an augmented matrix. The results are summarized as follows:

1. In the case that a good estimate on the constraint condition for the solution is available, an efficient approach uses inverse iteration for the solution of the eigenproblem combined with the GSVD for solution of the systems arising at each inverse iteration.
2. If a good estimate for the constraint parameter is available, but the algorithm is shown to converge slowly for a given class of problems, bisection search may be used to predict the total number of outer steps required for a given desired accuracy.
3. For cases without constraint information, the L-curve approach used for regularized LS has been adapted for regularized TLS.

Numerical experiments have been presented which verify all algorithms and we conclude with the following.

1. Algorithm 2.2 provides an efficient and practical approach for the solution of the RTLS problem in which a good estimate of the physical parameter is provided.
2. Otherwise, if blowup occurs, bisection search may yield a better solution satisfying the constraint condition.
3. If no constraint information is provided, the L-curve technique can be successfully employed.
4. Algorithm 2.1 performs better than QEP for all of our tests.

In all cases we have demonstrated a constructive and practical approach for the solution of regularized TLS problems.

**Acknowledgment.** The authors gratefully acknowledge the comments of three anonymous referees who suggested that we seek a proof of the convergence of our basic algorithm, which ultimately led to our improvement of the reliability of the solution technique.

#### REFERENCES

- [1] Z. BAI AND J. DEMMEL, *Computing the generalized singular value decomposition*, SIAM J. Sci. Comput., 14 (1993), pp. 1464–1486.
- [2] A. BJÖRCK, P. HEGGERNES, AND P. MATSTOMS, *Methods for large scale total least squares problems*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 413–429.
- [3] G. H. GOLUB, P. C. HANSEN, AND D. P. O’LEARY, *Tikhonov regularization and total least squares*, SIAM J. Matrix Anal. Appl., 21 (1999), pp. 185–194.
- [4] G. H. GOLUB AND C. VAN LOAN, *An analysis of the total least squares problem*, SIAM J. Numer. Anal., 17 (1980), pp. 883–893.
- [5] G. H. GOLUB AND C. VAN LOAN, *Matrix Computations*, 3rd ed., Johns Hopkins Press, Baltimore, 1996.
- [6] H. GUO AND R. A. RENAUT, *A regularized total least squares algorithm*, in Total Least Squares and Errors-in-Variables Modeling: Analysis, Algorithms and Applications, S. Van Huffel and P. Lemmerling, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002, pp. 57–66.
- [7] P. C. HANSEN, *Regularization, GSVD and truncated GSVD*, BIT, 29 (1989), pp. 491–504.
- [8] P. C. HANSEN, *Analysis of discrete ill-posed problems by means of the L-curve*, SIAM Rev., 34 (1992), pp. 561–580.
- [9] P. C. HANSEN, *Regularization tools: A Matlab package for analysis and solution of discrete ill-posed problems*, Numer. Algorithms, 6 (1994), pp. 1–35.

- [10] 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.
- [11] R. A. HORN AND C. R. JOHNSON, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.
- [12] S. VAN HUFFEL AND J. VANDEWALLE, *The Total Least Squares Problem: Computational Aspects and Analysis*, SIAM, Philadelphia, 1991.
- [13] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.
- [14] D. SIMA, S. VAN HUFFEL, AND G. H. GOLUB, *Regularized Total Least Squares Based on Quadratic Eigenvalue Problem Solvers*, Tech. Report SCCM-03-03, SCCM, Stanford University, Stanford, CA, 2003.
- [15] A. N. TIKHONOV, *Solution of incorrectly formulated problems and the regularization method*, Soviet Math. Dokl., 4 (1963), pp. 1035–1038.
- [16] A. N. TIKHONOV AND V. Y. ARSENIN, *Solution of Ill-Posed Problems*, John Wiley & Sons, New York, 1977.
- [17] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, UK, 1965.