Noname manuscript No.
(will be inserted by the editor)

Regularization of Large Scale Total Least Squares Problems

Heinrich Voss · Jörg Lampe

Received: date / Accepted: date

Abstract The total least squares (TLS) method is an appropriate approach for linear systems when not only the right-hand side but also the system matrix is contaminated by some noise. For ill-posed problems regularization is necessary to stabilize the computed solutions. In this presentation we discuss two approaches for regularizing large scale TLS problems. One which is based on adding a quadratic constraint and a Tikhonov type regularization concept.

1 Introduction

Many problems in data estimation are governed by over-determined linear systems

$$Ax \approx b, \quad A \in \mathbb{R}^{m\times n},\ b \in \mathbb{R}^m, m \geq n. \tag{1}$$

In the classical least squares approach the system matrix A is assumed to be free from error, and all errors are confined to the observation vector b. However, in engineering application this assumption is often unrealistic. For example, if not only the right-hand side b but A as well are obtained by measurements, then both are contaminated by some noise.

An appropriate approach to this problem is the total least squares (TLS) method which determines perturbations $\Delta A \in \mathbb{R}^{m\times n}$ to the coefficient matrix

Heinrich Voss
Institute of Numerical Simulation
Hamburg University of Technology
D-21071 Hamburg, Germany
Tel.: +49-40-428783279
Fax: +49-40-428782696
E-mail: voss@tuhh.de

Jörg Lampe
Germanischer Lloyd SE,
D-20457 Hamburg, Germany
E-mail: joerg.lampe@gl-group.com

and $\Delta b \in \mathbb{R}^m$ to the vector b such that

$$\|[\Delta A, \Delta b]\|_F^2 = \min! \quad \text{subject to } (A + \Delta A)x = b + \Delta b, \tag{2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. An overview of TLS methods and a comprehensive list of references is contained in [18].

When solving practical problems they are usually ill-conditioned, for example the discretization of ill-posed problems such as integral equations of the first kind. Then least squares or total least squares methods for solving (1) often yield physically meaningless solutions, and regularization is necessary to stabilize the computed solution.

A well established approach is to add a quadratic constraint to problem (2) yielding the regularized total least squares (RTLS) problem

$$\|[\Delta A, \Delta b]\|_F^2 = \min! \quad \text{subject to } (A + \Delta A)x = b + \Delta b,\ \|Lx\| \le \delta, \tag{3}$$

where $\|\cdot\|$ denotes the Euclidean norm, $\delta > 0$ is the quadratic constraint regularization parameter, and the regularization matrix $L \in \mathbb{R}^{p\times n}$, $p \le n$ defines a (semi-) norm on the solution through which the size of the solution is bounded or a certain degree of smoothness can be imposed [6,10,11,14,19, 20].

If the minimization problem (3) attains its solution then it can be rewritten into the more tractable form

$$f(x) := \frac{\|Ax - b\|^2}{1 + \|x\|^2} = \min! \quad \text{subject to} \quad \|Lx\| \le \delta. \tag{4}$$

Closely related is an approach which adopts Tikhonov's regularization concept to stabilize the TLS solution [2,15]:

$$f(x) + \lambda\|Lx\|^2 = \min!. \tag{5}$$

By comparing the corresponding Lagrangian functions of problems (4) and (5) it is obvious that they are equivalent in the following sense. For each Tikhonov parameter $\lambda \ge 0$ there exists a corresponding value of the quadratic constraint δ such that that the solutions of (4) and (5) are identical.

In this paper we review numerical methods for large scale regularized total least squares problems of both types (4) and (5). The first order conditions of (4) can be solved via a sequence of quadratic or linear eigenvalue problems, and the nonlinear first order condition of (5) is solved by Newton's method. To avoid the solution of large–scale eigenproblems and nonlinear systems of equation we apply iterative projection methods which allow for an excessive reuse of information from previous iteration steps. The efficiency of these approaches is evaluated with a couple of numerical examples.

2 Quadratically constrained total least squares problems

We consider the quadratically constrained TLS problem (4). We assume that the solution x_{RTLS} is attained and that the inequality is active, i.e. $\delta = \|Lx_{RTLS}\|$ (otherwise no regularization would be necessary). Under this condition Golub, Hansen and O'Leary [6] derived the following first order necessary conditions: The solution x_{RTLS} of problem (4) is a solution of the problem

$$(A^TA + \lambda_I I_n + \lambda_L L^TL)x = A^Tb, \tag{6}$$

where the parameters λ_I and λ_L are given by

$$\lambda_I = -f(x), \quad \lambda_L = \frac{1}{\delta^2}\big(b^T(b - Ax) - f(x)\big). \tag{7}$$

This condition was used in the literature in two ways to solve problem (4): In [10,20] for a chosen parameter λ_I problem (6) is solved for (x, λ_L), which yields a convergent sequence of updates for λ_I. Conversely, in [6,11,19] λ_I is chosen as a free parameter; for fixed λ_L problem (6) is solved for (x, λ_I), and then λ_L is updated in a way that the whole process converges to the solution of (4).

In either case the first order conditions require to solve a sequence of quadratic and linear eigenvalue problems, respectively. Efficient implementations recycling a great deal of information from previous iteration steps are presented in [10] for the first approach and in [11] for the latter one.

2.1 RTLS via a sequence of quadratic eigenvalue problems

The first algorithm is based on keeping the parameter λ_I fixed for one iteration step and letting $\lambda := \lambda_L$ be a free parameter. The first order optimality conditions then reads

$$B(x^k)x + \lambda L^TLx = A^Tb, \quad \|Lx\|^2 = \delta^2, \tag{8}$$

with

$$B(x^k) = A^TA - f(x^k)I, \;\; f(x^k) = -\lambda_I(x^k). \tag{9}$$

which suggests the following Algorithm 1.

Algorithm 1 RTLSQEP

Require: Initial vector x^1.

1: **for** $k = 1, 2, \ldots$ until convergence **do**

2: With $B_k := B(x^k)$ solve

$$B_k x^{k+1} + \lambda L^TLx^{k+1} = A^Tb, \quad \|Lx^{k+1}\|^2 = \delta^2 \tag{10}$$

for (x^{k+1}, λ) corresponding to the largest $\lambda \in \mathbb{R}$

3: **end for**

It was shown in [12] that this algorithm converges in the following sense: any limit point x^* of the sequence $\{x^k\}$ constructed by Algorithm 1 is a global minimizer of the minimization problem

$$f(x) = \frac{\|Ax-b\|^2}{1+\|x\|^2} \quad \text{subject to } \|Lx\|^2 = \delta^2.$$

Sima, Van Huffel and Golub [20] proposed to solve (10) for fixed k via a quadratic eigenvalue problem. This motivates the name RTLSQEP of the algorithm.

If L is square and nonsingular, then with $z = Lx^{k+1}$ problem (10) is equivalent to

$$W_k z + \lambda z := L^{-T} B_k L^{-1} z + \lambda z = L^{-T} A^T b =: h, \quad z^T z = \delta^2. \tag{11}$$

Assuming that $W_k + \lambda I$ is positive definite, and denoting $u := (W_k + \lambda I)^{-2} h$, one gets $h^T u = z^T z = \delta^2$, and $h = \delta^{-2} h h^T u$ yields that $(W_k + \lambda I)^2 u = h$ is equivalent to the quadratic eigenvalue problem

$$Q_k(\lambda) u := (W_k + \lambda I)^2 u - \delta^{-2} h h^T u = 0. \tag{12}$$

If the regularization matrix L is not quadratic the problem (8) has to be reduced to the range of L to obtain the quadratic eigenproblem of the same type as (11), cf. [10,20].

In [10] $W_k + \hat{\lambda} I$ is shown to be positive semidefinite. We are only considering the generic case of $W_k + \hat{\lambda} I$ being positive definite. In this case the solution of the original problem (10) is recovered from $z = (W_k + \hat{\lambda} I) u$, and $x^{k+1} = L^{-1} z$ where u is an eigenvector corresponding to $\hat{\lambda}$ which is scaled such that $h^T u = \delta^2$. The case that $W_k + \hat{\lambda} I \geq 0$ is singular means that the solution of (12) may not be unique, which is discussed by Gander, Golub and von Matt [5,9].

The transformation to the quadratic eigenproblem (12) seems to be very costly because of the inverse of L. Notice however, that typical regularization matrices are discrete versions of 1D first (or second) order derivatives

$$L = \begin{bmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \end{bmatrix} \in \mathbb{R}^{(n-1)\times n}. \tag{13}$$

Smoothing properties of L are not deteriorated significantly if they are replaced by regular versions like (cf. [3])

$$\hat{L} := \begin{bmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \\ & & & \varepsilon \end{bmatrix} \quad \text{or} \quad \hat{L} := \begin{bmatrix} \varepsilon & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix}$$

with some small diagonal element $\varepsilon > 0$. Which one of these modifications is chosen depends on the behavior of the solution of (4) close to the boundary. It

is not necessary to calculate the inverse of L explicitly since the presented algorithms touches only L^{-1} by matrix-vector multiplications. Solving a system with $\hat{L}$ is cheap, i.e. an $\mathcal{O}(n)$-operation.

If it holds that $\operatorname{rank}(L) < n$ and one is not willing to perturb L, a basis of the range and kernel are needed. In [10,20] it is explained how to obtain a similar expression for W_k in the quadratic eigenvalue problem (12). For the L in (13) the spectral decomposition is explicitly known. For solutions on a 2D or 3D domain one can take advantage of Kronecker product representations of L and its decomposition, cf. [14].

An obvious approach for solving the quadratic eigenvalue problem (12) at the k-th iteration step of Algorithm 1 is linearization, i.e. solving the linear eigenproblem

$$\begin{bmatrix} -2W_k & -W_k^2 + \delta^{-2} h_k h_k^T \\ I & 0 \end{bmatrix} \begin{bmatrix} \lambda u \\ u \end{bmatrix} = \lambda \begin{bmatrix} \lambda u \\ u \end{bmatrix}, \tag{14}$$

and choosing the maximal real eigenvalue, and the corresponding u–part of the eigenvector, which is an eigenvector of (12).

This approach is reasonable if the dimension n of problem (1) is small. For larger n it is not efficient to determine the entire spectrum of (12). In this case one could apply the implicitly restarted Arnoldi method implemented in ARPACK [16] (and included in MATLAB as function `eigs`) to determine the rightmost eigenvalue and corresponding eigenvector of (12). However, it is a drawback of linearization that symmetry properties of the quadratic problem are destroyed.

More efficient for large–scale problems are the second order Arnoldi reduction (SOAR) introduced by Bai and Su [1] or a Krylov subspace projection method for quadratic eigenvalue problems proposed by Li and Ye [17]. The paper [14] contains implementation details for these methods and a couple of numerical examples demonstrating their efficiency.

In Algorithm 1 we have to solve a sequence of quadratic eigenproblems where due to the convergence of the sequence $\{f(x_k)\}$ the matrices

$$W_k = L^{-T}(A^T A + f(x_k) I)L^{-1} = L^{-T} A^T A L^{-1} + f(x_k) L^{-T} L^{-1} \tag{15}$$

converge as well. This suggests to reuse as much information from previous steps as possible when solving $Q_k(\lambda)u = 0$.

The only degree of freedom in the Krylov methods mentioned above is the initial vector. Thus, the only information that can be recycled from previous iterations in these methods is the eigenvector of the preceding step that can be used as initial vector. Much more information can be exploited in general iterative projection methods such as the nonlinear Arnoldi algorithm [21] which can be started with the entire search space of the previous eigenvalue problem. Notice, that the Nonlinear Arnoldi method may be applied to more general nonlinear eigenvalue problem $T(\lambda)x = 0$.

Algorithm 2 Nonlinear Arnoldi

Require: Initial basis V, $V^TV = I$
1: Find rightmost eigenvalue μ of $V^TT(\mu)V\tilde{u} = 0$ and corresponding eigenvector $\tilde{u}$
2: Determine preconditioner $P \approx T(\sigma)^{-1}$, σ close to wanted eigenvalue
3: Set $u = V\tilde{u}$, $r = Q_k(\mu)u$
4: **while** $\|r\|/\|u\| > \epsilon_r$ **do**
5: $v = Pr$
6: $v = v - VV^Tv$
7: $\tilde{v} = v/\|v\|$, $V = [V, \tilde{v}]$
8: Find rightmost eigenvalue μ of $V^TT(\mu)V\tilde{u} = 0$ and corr. eigenvector $\tilde{u}$
9: Set $u = V\tilde{u}$, $r = Q_k(\mu)u$
10: **end while**

Some comments on an efficient implementation of RTLSQEP with the nonlinear Arnoldi solver are in order.

- A suitable initial basis V of Algorithm 2 for the first quadratic eigenvalue problem (12) can be determined by a small number of Lanczos steps applied to the linear eigenproblem $W_1z = \lambda z$ with a random vector r_0 because this is cheaper than executing the nonlinear Arnoldi method.
- Since the dimensions of the projected problems are small they can be solved by linearization and a dense eigensolver like the QR algorithm.
- In our numerical examples it turned out that we obtained fast convergence without preconditioning, so we simply set $P = I$.
- The representation of $W_k = L^{-T}(A^TA + f(x_k)I)L^{-1}$ demonstrates that the projected eigenvalue problem

$$V^TQ_k(\mu)V\tilde{u} = ((W_k + \mu I)V)^T((W_k + \mu I)V)\tilde{u} - \delta^{-2}(h^TV)^T(h^TV)\tilde{u} = 0$$

  can be determined efficiently if the matrices $L^{-T}A^TAL^{-1}V$, $L^{-T}L^{-1}V$ and h^TV are known. These can be updated cheaply by appending in every iteration step of the nonlinear Arnoldi method one column and component to the current matrices and vector, respectively.

The considerations above demonstrate that due to the reuse of the entire search space it is rather inexpensive to provide $V^TQ_k(\lambda)V$ if $V^TQ_{k-1}(\lambda)V$ is known. This suggests early updates, i.e. to leave the inner loop of the nonlinear Arnoldi method for determining the rightmost eigenpair long before convergence.

It turned out that while reducing the residual of the approximated rightmost eigenpair of a quadratic eigenproblem in step k by a factor 100 (instead of solving it to full accuracy), sufficient new information is added to the search space V. So the stopping criterion in line 4 of Algorithm 2 is replaced by $\|r\|/\|r_0\| > 0.01$ with the initial residual r_0 calculated in line 3. This approach leads to more outer iterations but overall to less inner iterations.

The early update variant reduces the overall computation time substantially when compared to the standard version. Implementing early update strategies in the Krylov-type algorithms destroyed the convergence of the overall process.

2.2 RTLS via a sequence of linear eigenvalue problems

The second algorithm is based on keeping the parameter λ_L fixed for one iteration step and letting $\lambda := -\lambda_I$ be a free parameter.

The following version of the first order optimality conditions was proved by Renaut and Guo in [19].

Theorem 1 *The solution x_{RTLS} of the RTLS problem (3) subject to the active constraint satisfies the augmented eigenvalue problem*

$$B(\lambda_L(x_{RTLS}))\begin{bmatrix} x_{RTLS} \\ -1 \end{bmatrix} = -\lambda_I(x_{RTLS})\begin{bmatrix} x_{RTLS} \\ -1 \end{bmatrix}, \tag{16}$$

with

$$B(\lambda_L) := [A,b]^T[A,b] + \lambda_L\, diag\{L^TL, -\delta^2\}$$

and λ_L and λ_I as given in (7).

This condition suggested Algorithm 3 called RTLSEVP for obvious reasons.

Algorithm 3 RTLSEVP

Require: Initial guess $\lambda_L^0 > 0$ and $B_0 = B(\lambda_L^0)$
1: **for** $k = 1, 2, \ldots$ until convergence **do**
2: Solve

$$B_{k-1}y^k = \lambda y^k \tag{17}$$

for eigenpair (y^k, λ) corresponding to the smallest λ
3: Scale y^k such that $y^k = \begin{bmatrix} x^k \\ -1 \end{bmatrix}$
4: Update $\lambda_L^k = \lambda_L(x^k)$ and $B_k = B(\lambda_L^k)$
5: **end for**

The choice of the smallest eigenvalue is motivated by the fact that we are aiming at $\lambda = -\lambda_I$ (cf. (16)), and by the first order conditions (5) it holds that

$$-\lambda_I = f(x) = \frac{\|Ax-b\|^2}{1+\|x\|^2}$$

is the function to be minimized.

The straightforward idea in [7] to update λ_L in line 4 with (7), i.e.

$$\lambda_L^{k+1} = \frac{1}{\delta^2}\Big(b^T(b - Ax^{k+1}) - \frac{\|Ax^{k+1}-b\|^2}{1+\|x^{k+1}\|^2}\Big) \tag{18}$$

does not lead in general to a convergent algorithm.

To enforce convergence Renaut and Guo [19] proposed to determine a value θ such that the eigenvector $(x_\theta^T, -1)^T$ of $B(\theta)$ corresponding to the smallest eigenvalue of $B(\theta)$ satisfies the constraint $\|Lx_\theta\|^2 = \delta^2$, i.e. find a non-negative root $\hat{\theta}$ of the real function

$$g(\theta) := \frac{\|Lx_\theta\|^2 - \delta^2}{1+\|x_\theta\|^2}. \tag{19}$$

Then the corresponding eigenvector $(x_{\hat\theta}^T, -1)^T$ is a solution of (4).

Solving the first order conditions in this way assumes that the last component of the eigenvector can be scaled to be -1. But the last component of an eigenvector corresponding to the smallest eigenvalue of $B(\theta)$ need not be different from zero, and then $g(\theta)$ is not necessarily defined. To fill this gap the following generalization was introduced in [11]:

Definition 1 Let $\mathcal{E}(\theta)$ denote the eigenspace of $B(\theta)$ corresponding to its smallest eigenvalue. Then

$$g(\theta) := \min_{y\in\mathcal{E}(\theta)} \frac{y^TNy}{y^Ty} = \min_{(x^T,x_{n+1})^T\in\mathcal{E}(\theta)} \frac{\|Lx\|^2-\delta^2x_{n+1}^2}{\|x\|^2+x_{n+1}^2} \tag{20}$$

is the minimal eigenvalue of the projection of N onto $\mathcal{E}(\theta)$.

This extends the definition of g to the case of eigenvectors with zero last components. The modified function g has the following properties which were proven in [11].

Theorem 2 *The function $g : [0,\infty) \to \mathbb{R}$ has the following properties:*

(i) *If $\sigma_{\min}([A,b]) < \sigma_{\min}(A)$ then $g(0) > 0$*
(ii) *$\lim_{\theta\to\infty} g(\theta) = -\delta^2$*
(iii) *If the smallest eigenvalue of $B(\theta_0)$ is simple, then g is continuous at θ_0*
(iv) *g is monotonically not increasing on $[0,\infty)$*
(v) *Let $g(\hat\theta) = 0$ and let $y \in \mathcal{E}(\hat\theta)$ such that $g(\hat\theta) = y^TNy/\|y\|^2$. Then the last component of y is different from 0.*
(vi) *g has at most one root.*

Theorem 2 demonstrates that if $\hat\theta$ is a positive root of g, then $x := -y(1:n)/y_{n+1}$ solves the RTLS problem (4) where y denotes an eigenvector of $B(\hat\theta)$ corresponding to its smallest eigenvalue.

REMARK 2.8. If the smallest singular value $\sigma_{n+1}(\tilde\theta)$ of $B(\tilde\theta)$ is simple, then it follows from the differentiability of $\sigma_{n+1}(\theta)$ and its corresponding right singular vector that

$$\frac{d\sigma_{n+1}(B(\theta))}{d\theta}\Big|_{\theta=\tilde\theta} = g(\tilde\theta). \tag{21}$$

Hence, searching the root of $g(\theta)$ can be interpreted as searching the maximum of the minimal singular values of $B(\theta)$ with respect to θ. □

REMARK 2.9. Notice that g is not necessarily continuous. If the multiplicity of the smallest eigenvalue of $B(\theta)$ is greater than 1 for some θ_0, then g may have a jump discontinuity at θ_0. It may even happen that g does not have a root, but it jumps below zero at some θ_0. This indicates a nonunique solution of problem (4). See [11] how to construct a solution in this case. Here we assume a unique solution, which is the generic case. □

To determine the minimum eigenvalue and corresponding eigenvector of

$$B(\theta_k)y = \big(M + \theta_k N\big)y = \lambda y \tag{22}$$

Renaut and Guo [19] proposed the Rayleigh quotient iteration initialized by the eigenvector found in the preceding iteration step. Hence, one uses information from the previous step, but an obvious drawback of this method is the fact that each iteration step requires $\mathcal{O}(n^3)$ operations providing the LU factorization of $B(\theta_k)$.

Similar to the approach in RTLSQEP the entire information gathered in previous iteration steps can be employed solving (22) via the nonlinear Arnoldi Algorithm 2 with thick starts applied to

$$T_k(\mu)u = (M + \theta_k N - \mu I)u = 0$$

This time in lines 1 and 8 we aim at the minimum eigenvalue of $T_k(\mu)$.

The projected problem

$$V^T T_k(\mu) V\tilde{u} = \big(([A,b]V)^T([A,b]V) + \theta_k V^T N V - \mu I\big)\tilde{u} = 0 \tag{23}$$

can be updated efficiently in both cases, if the search space is expanded by a new vector and if the iteration counter k is increased (i.e. a new θ_k is chosen).

The explicit form of the matrices M and N is not needed. If a new vector v is added to the search space span$\{V\}$, the matrices $A_V := [A,b]V$ and $L_V := LV(1:n,:)$ are refreshed appending the new column $A_v := [A,b]v$ and $L_v := Lv(1:n)$, respectively, and the projected matrix $M_V := V^T MV$ has to be augmented by the new last column $c_v := (A_V^T A_v; A_v^T A_v)$ and last row c_v^T. Since the update of Lv is usually very cheap (cf. Remark 2.4) the main cost for determining the projected problem is essentially 1 matrix–vector multiplication.

For the preconditioner in line 3 it is appropriate to chose $P \approx N^{-1}$ which according to Remark 2.4 usually can be implemented very cheaply and can be kept constant throughout the whole algorithm.

Assuming that g is continuous and strictly monotonically decreasing Renaut and Guo [19] derived the update

$$\theta_{k+1} = \theta_k + \frac{\theta_k}{\delta^2} g(\theta_k) \tag{24}$$

for solving $g(\theta) = 0$ where $g(\theta)$ is defined in (19), and at step k, $(x_{\theta_k}^T, -1)$ is the eigenvector of $B(\theta_k)$ corresponding to its minimal eigenvalue. Since this sequence usually does not converge, an additional backtracking was included, i.e. the update was modified to

$$\theta_{k+1} = \theta_k + \iota \frac{\theta_k}{\delta^2} g(\theta_k) \tag{25}$$

where $\iota \in (0,1]$ was bisected until the sign condition $g(\theta_k)g(\theta_{k+1}) \geq 0$ was satisfied. However, this safeguarding hampers the convergence of the method considerably.

We propose a root-finding algorithm taking into account the typical shape of $g(\theta)$. g has at most one root $\hat{\theta}$, and exactly one root in the generic case, which we assume. Left of $\hat{\theta}$ the slope of g is often very steep, while right of $\hat{\theta}$ it

is approaching its limit $-\delta^2$ quite quickly. This makes it difficult to determine $\hat{\theta}$ because for an initial value less than $\hat{\theta}$ Newton's method is extremely slow, and if the initial value exceeds $\hat{\theta}$ then usually Newton's method yields a negative iterate.

We approximate the root of g based on rational interpolation of g^{-1} (if it exists) which has a known pole at $\theta = -\delta^2$. Assume that we are given three pairs $(\theta_j, g(\theta_j))$, $j = 1, 2, 3$ with

$$\theta_1 < \theta_2 < \theta_3 \quad \text{and} \quad g(\theta_1) > 0 > g(\theta_3). \tag{26}$$

We determine the rational interpolation

$$h(\gamma) = \frac{p(\gamma)}{\gamma + \delta^2}, \text{ where } p \text{ is a polynomial of degree 2,}$$

and p is chosen such that $h(g(\theta_j)) = \theta_j$, $j = 1, 2, 3$, and we evaluate $\theta_4 = h(0)$. In exact arithmetic $\theta_4 \in (\theta_1, \theta_3)$, and we replace θ_1 or θ_3 by θ_4 such that the new triple satisfies (26).

The coefficients of p are obtained from a 3 by 3 linear system which may become very badly conditioned, especially close to the root. We therefore use Chebyshev polynomials transformed to the interval $[g(\theta_3), g(\theta_1)]$ as a basis for representing p.

If g is strictly monotonically decreasing in $[\theta_1, \theta_3]$ then h is a rational interpolation of $g^{-1} : [g(\theta_3), g(\theta_1)] \to \mathbb{R}$.

Due to nonexistence of the inverse g^{-1} on $[g(\theta_3), g(\theta_1)]$ or due to rounding errors very close to the root $\hat{\theta}$, it may happen that θ_4 is not contained in the interval (θ_1, θ_3). In this case we perform a bisection step such that the interval definitely still contains the root of g. If $g(\theta_2) > 0$, then we replace θ_1 by $\theta_1 = (\theta_2 + \theta_3)/2$, otherwise θ_3 is exchanged by $\theta_3 = (\theta_1 + \theta_2)/2$.

To initialize Algorithm 3 we determine three values θ_i such that not all $g(\theta_i)$ have the same sign. Given $\theta_1 > 0$ we multiply either by 0.01 or 100 depending on the sign of $g(\theta_1)$ and obtain after very few steps an interval that contains the root of g.

If a discontinuity at or close to the root is encountered, then a very small $\epsilon_\theta = \theta_3 - \theta_1$ appears with relatively large $g(\theta_1) - g(\theta_3)$. In this case we terminate the iteration and determine the solution as described in [11].

The evaluation of $g(\theta)$ can be performed efficiently by using the stored matrix $LV(1:n,:)$ to determine $\|Lu\|^2 = (LV(1:n,:)\tilde{u})^T(LV(1:n,:)\tilde{u})$ in much less than a matrix–vector multiplication.

3 Tikhonov type regularization

For the Tikhonov regularization problem (5) Beck and Ben–Tal [2] proposed an algorithm where in each iteration step a Cholesky decomposition has to be computed, which is prohibitive for large-scale problems. We present a method which solves the first order conditions of (5) via an iterative projection method.

Different from the least squares problem the first order condition for (5) is a nonlinear system of equations. After some simplification one gets

$$q(x) := (A^TA + \mu L^TL - f(x)I)x - A^Tb = 0, \quad \text{with } \mu := (1 + \|x\|^2)\lambda. \tag{27}$$

Newton's method then reads

$$x^{k+1} = x^k - J(x^k)^{-1}q(x^k)$$

with the Jacobi matrix

$$J(x) = A^TA + \mu L^TL - f(x)I - 2x\frac{x^TA^TA - b^TA - f(x)x^T}{1 + \|x\|^2}. \tag{28}$$

Taking advantage of the fact that J is a rank-one modification of

$$\hat{J}(x) := A^TA + \mu L^TL - f(x)I$$

and using the Sherman–Morrison formula Newton's method obtains the following form:

$$x^{k+1} = \hat{J}_k^{-1}A^Tb - \frac{1}{1 - (v^k)^T\hat{J}_k^{-1}u^k}\hat{J}_k^{-1}u^k(v^k)^T(x^k - \hat{J}_k^{-1}A^Tb), \tag{29}$$

where

$$u^k := 2x^k/(1 + \|x^k\|^2) \quad \text{and} \quad v^k := A^TAx^k - A^Tb - f(x^k)x^k.$$

Hence, in every iteration step we have to solve two linear systems with system matrix $\hat{J}(x^k)$, namely $\hat{J}_kz = A^Tb$ and $\hat{J}_kw = u^k$.

To avoid the solution of the large scale linear systems with varying matrices $\hat{J}_k$ we combine Newton's method with an iterative projection method similar to our approach for solving the eigenvalue problems in Section 2.

Assume that $\mathcal{V}$ is an ansatz space of small dimension k, and let the columns of $V \in \mathbb{R}^{k\times n}$ form an orthonormal basis of $\mathcal{V}$. We replace $z = \hat{J}_k^{-1}A^Tb$ with Vy_1^k where y_1^k solves the linear system $V^T\hat{J}_kVy_1^k = A^Tb$, and $w = \hat{J}_k^{-1}u^k$ is replaced with Vy_2^k where y_2^k solves the projected problem $V^T\hat{J}_kVy_2^k = u^k$.

If

$$x_{k+1} = V_ky_1^k - \frac{1}{1 - (v^k)^TV_ky_2^k}V_ky_2^k(v^k)^T(x^k - V_ky_1^k)$$

does not satisfy a prescribed accuracy requirement, then $\mathcal{V}$ is expanded with the residual

$$q(x^{k+1}) = (A^TA + \mu L^TL - f(x^k)I)x^{k+1} - A^Tb$$

and the step is repeated until convergence.

Initializing the iterative projection method with a Krylov space $\mathcal{V} = \mathcal{K}_\ell(A^TA + \mu L^TL, A^Tb)$ the iterates x^k are contained in a Krylov space of $A^TA + \mu L^TL$, and due to the convergence properties of the Lanczos process the main contributions come from the first singular vectors of $[A; \sqrt{\mu}L]$ which

for small μ are close to the first right singular vectors of A. It is common knowledge that these vectors are not always appropriate basis vectors for a regularized solution, and it may be advantageous to apply the regularization with a general regularization matrix L implicitly.

To this end we assume that L is nonsingular and we use the transformation $x := L^{-1}y$ of (5) (for general L we had to use the A-weighted generalized inverse $L_A^\dagger$, cf. [4]) which yields

$$\frac{\|AL^{-1}y - b\|^2}{1 + \|L^{-1}y\|^2} + \lambda\|y\|^2 = \min!.$$

Transforming the first order conditions back and multiplying from the left with L^{-1} one gets

$$(L^TL)^{-1}(A^TAx + \mu L^TLx - f(x)x - A^Tb) = 0.$$

This equation suggests to precondition the expansion of the search space with L^TL or an approximation $M \approx L^TL$ thereof which yields Algorithm 4.

Algorithm 4 Tikhonov TLS Method

Require: Initial basis V_0 with $V_0^TV_0 = I$, starting vector x^0
1: **for** $k = 0, 1, \ldots$ until convergence **do**
2: Compute $f(x^k) = \|Ax^k - b\|^2/(1 + \|x^k\|^2)$
3: Solve $V_k^T\hat{J}_kV_ky_1^k = V_k^TA^Tb$ for y_1^k
4: Compute $u^k = 2x^k/(1 + \|x^k\|^2)$ and $v^k = A^TAx^k - A^Tb - f(x^k)x^k$
5: Solve $V_k^T\hat{J}_kV_ky_2^k = V_k^Tu^k$ for y_2^k
6: Compute $x^{k+1} = V_ky_1^k - \frac{1}{1-(v^k)^TV_ky_2^k}V_ky_2^k(v^k)^T(x^k - V_ky_1^k)$
7: Compute $q^{k+1} = (A^TA + \mu L^TL - f(x^k)I)x^{k+1} - A^Tb$
8: Compute $\tilde{r} = M^{-1}q^{k+1}$
9: Orthogonalize $\hat{r} = (I - V_kV_k^T)\tilde{r}$
10: Normalize $v_{\text{new}} = \hat{r}/\|\hat{r}\|$
11: Enlarge search space $V_{k+1} = [V_k, v_{\text{new}}]$
12: **end for**
13: Output: Approximate Tikhonov TLS solution x^{k+1}

Possible stopping criteria in Line 1 are the following ones (or a combination thereof):

- Stagnation of the sequence $\{f(x^k)\}$, i.e., the relative change of two consecutive values of $f(x^k)$ is small: $|f(x^{k+1}) - f(x^k)|/f(x^k)$ is smaller than a given tolerance.
- The relative change of two consecutive Ritz vectors x^k is small, i.e., $\|x^{k+1} - x^k\|/\|x^k\|$ is smaller than a given tolerance.
- The relative residual q^k from Line 7 is sufficiently small, i.e., $\|q^k\|/\|A^Tb\|$ is smaller than a given tolerance.

Some remarks how to efficiently determine an approximate solution of the large-scale Tikhonov TLS problem (5) with Algorithm 1 are in order. For

large-scale problems matrix valued operations are prohibitive, thus our aim is to carry out the algorithm with a computational complexity of $\mathcal{O}(mn)$, i.e., of the order of a matrix-vector product with a (general) dense matrix $A \in \mathbb{R}^{m\times n}$.

- Similar to Algorithms 1 and 3 the Tikhonov TLS methods allows for a massive reuse of information from previous iteration steps. Assume that the matrices V_k, A^TAV_k, L^TLV_k are stored. Then neglecting multiplications with L and L^T and solves with M (due to the structure of a typical regularization matrix L and a typical preconditioner M) the essential cost in every iteration step is only two matrix-vector products with dense matrices A and A^T for extending A^TAV_k. With these matrices $f(x^k)$ in Line 1 can be evaluated as

$$f(x^k) = \frac{1}{1+\|y^k\|^2}\left((x^k)^T(A^TAy^k) - 2(y^k)^TV_k^T(A^Tb) + \|b\|^2\right),$$

  and q^{k+1} in Line 7 can be determined according to

$$q^{k+1} = (A^TAV_k)y^{k+1} + \mu(L^TLV_k)y^{k+1} - f(x^k)x^{k+1} - A^Tb.$$

- A suitable initial basis V_0 is an orthonormal basis of the Krylov space $\mathcal{K}_\ell\left(M^{-1}(A^TA+\mu L^TL), M^{-1}A^Tb\right)$ of small dimension, e.g. $\ell = 5$.
- Typically the initial vector $x^0 = 0$ is sufficiently close to the RTLS solution. In this case it holds that $f(x^0) = \|b\|^2$. If a general starting vector is used, e.g. the solution of the projected problem with initial space V_0 or some other reasonable approximation to x_{RTLS}, an additional matrix–vector product has to be spent for computing $f(x^0)$.
- It is enough to carry out one LDL^T-decomposition of the projected matrix $V_k^T\hat{J}_kV_k$, which then can be used twice to solve the linear systems in Lines 3 and 5.
- Since the the number of iteration steps until convergence is usually very small compared to the dimension n, the overall cost of Algorithm 1 is of the order $\mathcal{O}(mn)$.

4 Numerical Examples

To evaluate the performance of Algorithms 1, 3, and 4 we use large scale test examples from Hansen's *Regularization Tools* [8] which contains MATLAB routines providing square matrices $A_{\text{true}} \in \mathbb{R}^{n\times n}$, right-hand sides b_{true} and true solutions x_{true}, with $A_{\text{true}}x_{\text{true}} = b_{\text{true}}$. All examples originate from discretizations of Fredholm equations of the first kind and the matrices A_{true} and $[A_{\text{true}}, b_{\text{true}}]$ are ill-conditioned.

To adapt the problem to an overdetermined linear system of equations we stack two error-contaminated matrices and right-hand sides (with different noise realizations), i.e.,

$$A = \begin{bmatrix}\bar{A}\\ \bar{A}\end{bmatrix}, \qquad b = \begin{bmatrix}\bar{b}\\ \bar{b}\end{bmatrix},$$

with the resulting matrix $A \in \mathbb{R}^{2n\times n}$ and $b \in \mathbb{R}^{2n}$.

The regularization matrix L is chosen to be the nonsingular approximation to the scaled discrete first order derivative operator in one space-dimension, i.e. L in (13) with an additional last row $(0,\dots,0,0.1)$. The numerical tests are carried out on an Intel Core 2 Duo T7200 computer with 2.3 GHz and 2 GB RAM under MATLAB R2009a (actually our numerical examples require less than 0.5 GB RAM).

Table 1 contains the results for six problems of dimension $n = 2000$ from the Regularization Toolbox for the three methods mentioned before. The underlying problems are all discrete versions of Fredholm's integral equations of the first kind. For problem *heat* the parameter κ controls degree of ill-posedness.

Table 1 shows the relative residuals, number of iterations and of matrix-vector products, and the relative error averaged over 10 examples of the same type with different realizations of the error. σ denotes the noise level.

The examples demonstrate that the Tikhonov method outperforms the two methods based on eigenvalue solvers. For most of the examples the number of matrix-vector products of Algorithm 4 is about only $50-75\%$ of the ones required for the RTSLQEP and RTLSEVP methods. This is reconfirmed by many more examples in [15]. The relative errors in the last column of Table 1 indicate that all three methods yield reliable numerical solutions.

Table 1 Problems from Regularization Tools

Problem	Method	$\frac{\|q(x^k)\|}{\|A^Tb\|}$	Iters	MatVecs	$\frac{\|x-x_{true}\|}{\|x_{true}\|}$
phillips	TTLS	8.5e-16	8.0	25.0	8.9e-2
$\sigma = 1e-3$	RTLSQEP	5.7e-11	3.0	42.0	8.9e-2
	RTLSEVP	7.1e-13	4.0	47.6	8.9e-2
baart	TTLS	2.3e-15	10.1	29.2	1.5e-1
$\sigma = 1e-3$	RTLSQEP	1.0e-07	15.7	182.1	1.4e-1
	RTLSEVP	4.1e-10	7.8	45.6	1.5e-1
shaw	TTLS	9.6e-16	8.3	25.6	7.0e-2
$\sigma = 1e-3$	RTLSQEP	3.7e-09	4.1	76.1	7.0e-2
	RTLSEVP	2.6e-10	3.0	39.0	7.0e-2
deriv2	TTLS	1.2e-15	10.0	29.0	4.9e-2
$\sigma = 1e-3$	RTLSQEP	2.3e-09	3.1	52.3	4.9e-2
	RTLSEVP	2.6e-12	5.0	67.0	4.9e-2
heat(κ=1)	TTLS	8.4e-16	19.9	48.8	1.5e-1
$\kappa = 1e-2$	RTLSQEP	4.1e-08	3.8	89.6	1.5e-1
	RTLSEVP	3.2e-11	4.1	67.2	1.5e-1
heat(κ=5)	TTLS	1.4e-13	25.0	59.0	1.1e-1
$\sigma = 1e-3$	RTLSQEP	6.1e-07	4.6	105.2	1.1e-1
	RTLSEVP	9.8e-11	4.0	65.0	1.1e-1

5 Conclusions

Three methods for solving regularized total least squares problems are discussed two of which require the solution of a sequence of (linear or quadratic) eigenvalue problems and the third one needs the solution of a sequence of linear problems. In either case it is highly advantageous to combine the method with an iterative projection method and to reuse information gathered in previous iteration steps. Several examples demonstrate that all three methods require fairly small dimensions of the ansatz spaces and are qualified to solve large-scale regularized total least squares problems efficiently.

In this paper we assumed that the regularization parameters λ, δ and μ, respectively is given. In practical problems this parameter has to be determined by some method like the L-curve criterion, the discrepancy principle, generalized cross validation, or information criteria. All of these approaches require to solve the RTLS problem repeatedly for many different values of the regularization parameter. Notice however, that the eigenvalue problems in Section 2 and the first order condition in Section 3 depend in a very simple way on the respective regularization parameter such that the matrices characterizing the projected problem for one parameter can be reused directly when changing the parameter. For the quadratically constrained TLS problem and the L-curve criterion the details were worked out in [13] demonstrating the high efficiency of reusing search spaces when solving the sequence of eigenvalue problems.

References

1. Bai, Z., Su, Y.: SOAR: A second order Arnoldi method for the solution of the quadratic eigenvalue problem. SIAM J. Matrix Anal. Appl. **26**, 640 – 659 (2005)
2. Beck, A., Ben-Tal, A.: On the solution of the Tikhonov regularization of the total least squares problem. SIAM J. Optim. **17**, 98 – 118 (2006)
3. Calvetti, D., Reichel, L., Shuibi, A.: Invertible smoothing preconditioners for linear discrete ill-posed problems. Appl. Numer. Math. **54**, 135 – 149 (2005)
4. Eldén, L.: A wighted pseudoinverse, generalized singular values, and constrained least squares problems. BIT **22**, 487 – 502 (1982)
5. Gander, W., Golub, G., von Matt, U.: A constrained eigenvalue problem. Linear Algebra Appl. **114–115**, 815 – 839 (1989)
6. Golub, G., Hansen, P., O'Leary, D.: Tikhonov regularization and total least squares. SIAM J. Matrix Anal. Appl. **21**, 185 – 194 (1999)
7. Guo, H., Renaut, R.: A regularized total least squares algorithm. In: S. Van Huffel, P. Lemmerling (eds.) Total Least Squares and Errors-in-Variable Modelling, pp. 57 – 66. Kluwer Academic Publisher, Dodrecht, The Netherlands (2002)
8. Hansen, P.: Regularization tools version 4.0 for Matlab 7.3. Numer. Alg. **46**, 189 – 194 (2007)
9. Lampe, J.: Solving regularized total leasr squares problems based on eigenproblems. Ph.D. thesis, Institute of Numerical Simulation, Hamburg University of Technology (2010)
10. Lampe, J., Voss, H.: On a quadratic eigenproblem occuring in regularized total least squares. Comput. Stat. Data Anal. **52/2**, 1090 – 1102 (2007)
11. Lampe, J., Voss, H.: A fast algorithm for solving regularized total least squares problems. Electr. Trans. Numer. Anal. **31**, 12 – 24 (2008)
12. Lampe, J., Voss, H.: Global convergence of RTLSQEP: a solver of regularized total least squares problems via quadratic eigenproblems. Math. Modelling Anal. **13**, 55 – 66 (2008)

13. Lampe, J., Voss, H.: Efficient determination of the hyperparameter in regularized total least squares problems. Tech. rep., Institute of Numerical Simulation, Hamburg University of Technology (2009). DOI 10.1016/j.apnum.2010.06.005. To appear in Appl. Numer. Math.
14. Lampe, J., Voss, H.: Solving regularized total least squares problems based on eigenproblems. Taiwanese J. Math. **14**, 885 – 909 (2010)
15. Lampe, J., Voss, H.: Large-scale Tikhonov regularization of total least squares. Tech. Rep. 151, Institute of Numerical Simulation, Hamburg University of Technology (2011). Submitted to J. Comput. Appl. Math.
16. Lehoucq, R., Sorensen, D., Yang, C.: ARPACK Users' Guide. Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods. SIAM, Philadelphia (1998)
17. Li, R.C., Ye, Q.: A Krylov subspace method for quadratic matrix polynomials with application to constrained least squares problems. SIAM J. Matrix Anal. Appl. **25**, 405 – 428 (2003)
18. Markovsky, I., Van Huffel, S.: Overview of total least squares methods. Signal Processing **87**, 2283 – 2302 (2007)
19. Renaut, R., Guo, H.: Efficient algorithms for solution of regularized total least squares. SIAM J. Matrix Anal. Appl. **26**, 457 – 476 (2005)
20. Sima, D., Van Huffel, S., Golub, G.: Regularized total least squares based on quadratic eigenvalue problem solvers. BIT Numerical Mathematics **44**, 793 – 812 (2004)
21. Voss, H.: An Arnoldi method for nonlinear eigenvalue problems. BIT Numerical Mathematics **44**, 387 – 401 (2004)