Large-Scale Tikhonov Regularization of Total Least Squares

Jörg Lampe * Heinrich Voss †

Keywords: total least squares, general-form Tikhonov regularization, ill-posedness

AMS Subject Classification: 65F15, 65F22, 65F30

Abstract

The total least squares (TLS) method is a successful approach for linear problems when not only the right-hand side but the system matrix as well are contaminated by some noise. For ill-posed TLS problems regularization is necessary to stabilize the computed solution. In this paper we present a new approach for computing an approximate solution of the Tikhonov-regularized large-scale total least-squares problem. An iterative method is proposed which solves a convergent sequence of projected linear systems and thereby builds up a highly suitable search space. The focus is on efficient implementation with particular emphasis on the reuse of information.

1 Introduction

Many problems in data estimation are governed by overdetermined linear systems

$$Ax \approx b, \quad A \in \mathbb{R}^{m\times n},\ b \in \mathbb{R}^m, m \geq n. \tag{1.1}$$

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 the matrix A is an approximation of the true operator or if not only the right-hand side b but A as well are obtained by measurements, then both are contaminated by some noise.

*Germanischer Lloyd SE, D-20457 Hamburg, Germany (joerg.lampe@gl-group.com)

†Institute of Numerical Simulation, Hamburg University of Technology, D-21071 Hamburg, Germany (voss@tu-harburg.de)

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

$$\|[\Delta A, \Delta b]\|_F^2 = \text{min!} \quad \text{subject to } (A + \Delta A)x = b + \Delta b, \tag{1.2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. An overview of total least squares methods and a comprehensive list of references is contained in [19, 25, 26].

The TLS problem (1.2) can be analyzed (cf. [8, 26]) in terms of the singular value decomposition (SVD) of the augmented matrix $[A, b] = U\Sigma V^T$. A TLS solution exists if and only if the right singular subspace $\mathcal{V}_{min}$ corresponding to σ_{n+1} contains at least one vector with a nonzero last component. It is unique if it holds that $\sigma'_n > \sigma_{n+1}$ where σ'_n denotes the smallest singular value of A, and then it is given by

$$x_{TLS} = -\frac{1}{V(n+1, n+1)} V(1:n, n+1).$$

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 (cf. [5, 9]). Then least squares or total least squares methods for solving (1.1) often yield physically meaningless solutions, and regularization is necessary to stabilize the computed solution.

To regularize problem (1.2) Fierro, Golub, Hansen and O'Leary [6] suggested to filter its solution by truncating the small singular values of the TLS matrix $[A, b]$, and they proposed an iterative algorithm based on Lanczos bidiagonalization for computing truncated TLS solutions.

Another well established approach is to add a quadratic constraint to problem (1.2) yielding the regularized total least squares (RTLS) problem

$$\|[\Delta A, \Delta b]\|_F^2 = \text{min!} \quad \text{subject to } (A + \Delta A)x = b + \Delta b,\ \|Lx\| \le \delta, \tag{1.3}$$

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. Typically it holds that $\delta < \|Lx_{TLS}\|$ or even $\delta \ll \|Lx_{TLS}\|$ which indicates an active constraint. Stabilization of total least squares problems by introducing a quadratic constraint was extensively studied in [2, 7, 11, 13, 14, 15, 17, 18, 21, 23, 24].

Let $F \in \mathbb{R}^{n\times k}$ be a matrix whose columns form an orthonormal basis of the nullspace of the regularization matrix L. If it holds that

$$\sigma_{min}([AF, b]) < \sigma_{min}(AF) \tag{1.4}$$

then the solution x_{RTLS} of problem (1.3) is attained, see [1] for the proof. The weak inequality $\sigma_{min}([AF, b]) \le \sigma_{min}(AF)$ always holds by Cauchy's interlacing

theorem. Note, that an empty kernel of L directly implies that a solution is attained, e.g. this holds for all nonsingular regularization matrices.

In this paper we always assume the inequality (1.4) to hold. Then it is possible to rewrite problem (1.3) into the more tractable form

$$\frac{\|Ax-b\|^2}{1+\|x\|^2} = \min! \quad \text{subject to} \quad \|Lx\| \le \delta. \tag{1.5}$$

Closely related is the approach of Beck and Ben-Tal who adopted the Tikhonov regularization concept to stabilize the TLS solution in [1]:

$$\frac{\|Ax-b\|^2}{1+\|x\|^2} + \lambda\|Lx\|^2 = \min!. \tag{1.6}$$

By comparing the corresponding Lagrangians of problems (1.5) and (1.6) it is obvious that they yield identical solution sets, i.e., for each Tikhonov parameter $\lambda \ge 0$ there exists a corresponding value of the quadratic constraint δ. There exists a monotonic decreasing nonlinear relation that maps $\lambda \in [0,\infty)$ to $\delta \in [0, \|Lx_{TLS}\|]$, which implies that the TLS solution is attained as well. If only the RTLS solution but not the TLS solution is attained then $\lambda \in [0,\infty)$ is mapped to $\delta \in [0,\infty)$, cf. [11]. A method for solving (1.6) is proposed in [1], where in each iteration step a Cholesky decomposition has to be computed, which is prohibitive for large-scale problems.

In this paper we propose an iterative projection method which combines orthogonal projections to a sequence of generalized Krylov subspaces of increasing dimensions and Newton's method for the first order equations of (1.6). Taking advantage of the Sherman–Morrison formula the system matrices can be updated efficiently such that the essential cost of an iteration step are two matrix-vector products. Since usually a very small number of iterations steps is required for convergence the computational complexity of our method is essentially of the order of a matrix-vector product with a (general) dense matrix A.

Our paper is organized as follows. In section 2 important properties of the Tikhonov regularization for TLS problems are summarized, and several methods are reviewed for solving small sized problems. For solving large-scale problems different approaches based on orthogonal projection are proposed in section 3. The focus is on the reuse of information when building up well suited search spaces. Section 4 contains numerical examples demonstrating the efficiency of the presented methods. Concluding remarks can be found in section 5.

2 Tikhonov Regularization for TLS

The Tikhonov regularization of TLS problems is by far less intensely studied than the regularization by an additional quadratic constraint. It is an important property of the Tikhonov TLS problem (1.6) (and of the problem (1.5) as well) that in general no closed form solution for x_{RTLS} exists.

With the solution x_{RTLS} of (1.6) the corrections ΔA and Δb are given by

$$\Delta A = -\frac{(Ax_{RTLS} - b)x_{RTLS}^T}{1 + \|x_{RTLS}\|^2}$$

and

$$\Delta b = \frac{Ax_{RTLS} - b}{1 + \|x_{RTLS}\|^2}.$$

Hence it holds that

$$\|[\Delta A, \Delta b]\|_F^2 = \frac{\|Ax_{RTLS} - b\|^2}{1 + \|x_{RTLS}\|^2} =: f(x_{RTLS}),$$

and furthermore

$$f(x_{RTLS}) + \lambda \|Lx_{RTLS}\|^2 \leq \sigma_{min}^2([AF, b]),$$

as shown in [1]. The approach suggested in [1] for solving problem (1.6) is based on the reformulation into the following double minimization problem

$$\min_{\alpha \geq 1} \min_{\|x\|^2 = \alpha - 1} \frac{\|Ax - b\|^2}{\alpha} + \lambda \|Lx\|^2.$$

This can also be written as

$$\min_{\alpha \geq 1} \mathcal{G}(\alpha)$$

with

$$\mathcal{G}(\alpha) = \min_{\|x\|^2 = \alpha - 1} \frac{\|Ax - b\|^2}{\alpha} + \lambda \|Lx\|^2.$$

It has been shown that $\mathcal{G}(\alpha)$ is continuous and under a mild condition differentiable (and even unimodal in several cases). Calculating function values of $\mathcal{G}(\alpha)$ requires solving a minimization problem with a quadratic objective function and a norm equality constraint, i.e., a trust-region subproblem (TRS). In [1] an enclosing bisection algorithm is suggested by solving a sequence of TRSs. The suggested TRTLSG algorithm converges to the global minimum if the function $\mathcal{G}$ is unimodal, otherwise the bisection strategy has to be replaced by more expensive one dimensional global solvers.

Remark 2.1 *In [12] an efficient method for solving large-scale trust-region subproblems has been presented that is based on recycling previously gained information by the use of the Nonlinear Arnoldi method [27, 28]. The idea of reusing as much information as possible during solving a sequence of converging trust-region subproblems can be directly adapted. Employing the Nonlinear Arnoldi as TRS solver within the TRTLSG algorithm with bisection search (and for a modified version with a global minimizer for $\mathcal{G}(\alpha)$ as well) will substantially speed up the computations.*

Let us now consider several Newton approaches for solving the Tikhonov TLS problem. To derive the first-order optimality conditions of (1.6), its derivative with respect to x is set equal to zero:

$$\frac{(2A^TAx - 2A^Tb)(1+\|x\|^2) - 2x\|Ax-b\|^2}{(1+\|x\|^2)^2} + 2\lambda L^TLx = 0.$$

This can be recast to

$$\tilde{q}(x) := \left(A^TA + \lambda(1+\|x\|^2)L^TL - f(x)I\right)x - A^Tb = 0. \tag{2.1}$$

From the set of solutions of (2.1) the minimizer of $f(x) = \|Ax-b\|^2/(1+\|x\|^2)$ is the solution x_{RTLS} of (1.6). A straightforward idea is to apply Newton's method to $\tilde{q}(x)$. After some calculations the Jacobian is given by

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

Let us assume that $\tilde{q}(x)$ is two times continuously differentiable in a neighborhood of x_{RTLS} and let $\tilde{J}(x_{RTLS})$ be a regular matrix. When starting with a vector $x^0 \in \mathbb{R}^n$ close to x_{RTLS} the Newton iteration

$$x^{k+1} = x^k - \tilde{J}(x^k)^{-1} \cdot \tilde{q}(x^k) \tag{2.2}$$

indeed converges quadratically to the RTLS solution.

A second idea is to introduce the new parameter

$$\lambda_L := \lambda(1+\|x\|^2) \tag{2.3}$$

which simplifies equation (2.1) to

$$q(x) := (A^TA + \lambda_L L^TL - f(x)I)x - A^Tb = 0. \tag{2.4}$$

The meaning of a fixed value λ_L is not directly related to a fixed Tikhonov parameter λ. Only after having solved problem (1.6) the corresponding Tikhonov parameter λ can be determined by (2.3), i.e., $\lambda = \lambda_L/(1+\|x_{RTLS}\|^2)$.

Remark 2.2 *When comparing* (2.4) *to Tikhonov regularization for least squares problems, the additional term $-f(x_{RTLS})I$ has a deregularizing effect. In [7] it has been pointed out that a positive definite matrix $\lambda_L L^TL - f(x_{RTLS})I$ indicates the RTLS solution to correspond to the solution of the Tikhonov regularized least squares problem $\|Ax-b\|^2 + \|\tilde{L}x\|^2$ with $\tilde{L} = (\lambda_L L^TL - f(x_{RTLS})I)^{1/2}$. But since the matrix $\lambda_L L^TL - f(x_{RTLS})I$ is already not positive definite when L is rectangular (with $p < n$), there is no equivalent interpretation in most cases. Note that often even the matrix $(A^TA + \lambda_L L^TL - f(x_{RTLS})I)$ is indefinite! In this paper we assume the matrix $(A^TA + \lambda_L L^TL - f(x_{RTLS})I)$ to be regular, such that $x_{RTLS} = (A^TA + \lambda_L L^TL - f(x_{RTLS})I)^{-1}A^Tb$ yields the unique RTLS solution. An investigation on nonunique RTLS solutions can be found in [11].*

An advantage of the formulation (2.4) is the slightly simpler Jacobian

$$J(x) = A^T A + \lambda_L L^T L - f(x)I - 2x\frac{x^T A^T A - b^T A - f(x)x^T}{1 + \|x\|^2} \tag{2.5}$$

for the Newton scheme

$$x^{k+1} = x^k - J(x^k)^{-1} \cdot q(x^k). \tag{2.6}$$

The convergence is again locally quadratic. A drawback of these approaches are the high costs due to solving a sequence of varying linear systems. When keeping the Jacobians $\tilde{J}(x^0)$ and $J(x^0)$ respectively fixed, i.e., only one decomposition has to be computed, the convergence drops to linear within a simplified Newton scheme.

There exists an interesting connection between the Newton iteration on $q(x)$ and an algorithm proposed in [11, 23] denoted as 'Iterative Refinement for the RTLS solution' (Algorithm 4.1 in [23]). The essential idea is to keep the value $f(x)$ within the expression for $q(x)$ in (2.4) fixed for one step, and then to use the solution of the resulting linear system to update $f(x)$. This leads to the following iteration procedure

$$x^{k+1} = (A^T A + \lambda_L L^T L - f(x^k)I)^{-1} A^T b. \tag{2.7}$$

Iteration (2.7) is similar to Newton's method applied to $q(x)$, i.e., now using the slightly modified Jacobian $\hat{J}(x) = A^T A + \lambda_L L^T L - f(x)I$. The difference between $\hat{J}(x)$ and $J(x)$ from (2.5) is only a rank-1 matrix. With $\hat{J}(x)$ replacing $J(x)$ in the iteration scheme (2.6) we obtain the Newton-like iteration

$$x^{k+1} = x^k - \hat{J}(x^k)^{-1} \cdot q(x^k) = x^k - \hat{J}(x^k)^{-1} \left(\hat{J}(x^k)x^k - A^T b \right) = \hat{J}(x^k)^{-1} A^T b \tag{2.8}$$

which generates an identical sequence $\{x^k\}$ as iteration (2.7). Let us investigate iteration (2.8) by means of fixed point analysis. For the Newton iteration (2.6) it is known that the solution x_{RTLS} is an attractive fixed point of the function

$$\phi(x) := x - J(x)^{-1} \cdot q(x),$$

under the mild conditions that $J(x_{RTLS})$ is regular and the derivative $q'(x)$ is Lipschitz continuous in a neighborhood of x_{RTLS}. But for the Newton-like iteration (2.8) the fixed point x_{RTLS} does not have to be attractive. When investigating the derivative of the fixed point function

$$\hat{\phi}(x) := x - \hat{J}(x)^{-1} \cdot q(x)$$

at the RTLS solution x_{RTLS}, i.e., it holds that $q(x_{RTLS}) = 0$, we obtain:

$$\begin{aligned} \hat{\phi}'(x_{RTLS}) &= I - \hat{J}(x_{RTLS})^{-1} J(x_{RTLS}) \\ &= \hat{J}(x_{RTLS})^{-1} x_{RTLS} \cdot 2\frac{x_{RTLS}^T A^T A - b^T A - f(x_{RTLS}) x_{RTLS}^T}{1 + \|x_{RTLS}\|^2}. \end{aligned}$$

A sufficient condition for the (local) linear convergence of the iteration (2.8) is that x_{RTLS} is an attractive fixed point, i.e., by Ostrowski's theorem it has to hold that the spectral radius ρ of the Jacobian of the fixed point function at the fixed point x_{RTLS} is less than one:

$$\rho\left(\hat{\phi}'(x_{RTLS})\right) < 1.$$

Note that in the generic case it holds that $\rho\left(\hat{\phi}'(x_{RTLS})\right) \ll 1$. But Example 2.4 shows that this condition does not always hold.

Remark 2.3 *At first sight this result is surprising, especially when comparing the Newton-like iteration to a simplified Newton iteration with the starting vector $x^0 = 0$. The Jacobian is given by*

$$J(x^0) = \hat{J}(x^0) = A^T A + \lambda_L L^T L - f(x^0) I = A^T A + \lambda_L L^T L - \|b\|^2 I,$$

which is subsequently used during the whole simplified Newton scheme, and which is also identical to the initial Jacobian approximation $\hat{J}(x^0)$ in the Newton-like iteration (2.8)*. Hence keeping this matrix fixed yields linear convergence (if $x^0 = 0$ is close enough to the solution x_{RTLS}) whereas updating the Jacobian approximations by $\hat{J}(x^k)$ can result in divergence.*

Example 2.4 *Let*

$$A = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 2 & -0.5 \\ 0 & 0 & 1.2 \end{pmatrix},\ b = \begin{pmatrix} 6 \\ -15 \\ -6 \end{pmatrix},\ L = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0.5 \end{pmatrix} \text{ and } \lambda_L = 0.7.$$

The unique Tikhonov RTLS solution is given by $x_{RTLS} \approx [1.99, -5.60, -4.39]^T$ with the value $f(x_{RTLS}) \approx 0.66$. At the solution it holds $\rho(\hat{\phi}'(x_{RTLS})) \approx 1.14 > 1$, hence for the Newton-like iteration the RTLS solution is a non-attractive fixed point. With the starting vector $x^0 = x_{RTLS} + e_r$, where e_r is a vector of normally distributed white noise with zero mean and a standard deviation such that it holds that $\frac{\|e_r\|}{\|x_{RTLS}\|} = 0.1$ (i.e., a relative error of 10% is added), the Newton iteration (2.6) *converges in 4 steps to machine precision (which is $\varepsilon = 2^{-52}$ here). A simplified Newton scheme using the Jacobian at the starting vector x^0 throughout the process converges within 14 steps, whereas iteration* (2.8) *does not converge to x_{RTLS}.*

Note that the spectral radius of the rank-1 matrix $\hat{\phi}'(x_{RTLS})$ is given by:

$$\rho\left(\hat{\phi}'(x_{RTLS})\right) = \frac{2\left(x_{RTLS}^T A^T A - b^T A - f(x_{RTLS}) x_{RTLS}^T\right) \hat{J}(x_{RTLS})^{-1} x_{RTLS}}{1 + \|x_{RTLS}\|^2}.$$

A detailed investigation on sufficient conditions with respect to the starting vector x^0 for the more general Newton-like scheme

$$x^{k+1} = x^k - K(x^k)^{-1} \cdot q(x^k) \tag{2.9}$$

with $K(x)$ as a general approximation to the Jacobian $J(x)$ has been carried out in [3, 4, 22]. Unfortunately, the sufficient conditions for the convergence of iteration (2.9) stated in Theorem 2.6 in [3] cannot be efficiently verified during the computation of a large-scale example.

3 Tikhonov TLS via Orthogonal Projection

When solving large-scale problems it is prohibitive to solve a large number of huge linear systems. A natural approach would be to project the linear systems onto search spaces of much smaller dimensions, and then only solve the projected problems. With respect to suitable search spaces an advantage of the Newton-like iteration (2.7) over Newton's method (2.6) is that the occurring sequence of linear systems can be solved efficiently by Krylov solvers due to the constant right-hand side $A^T b$ and the shift-invariance of Krylov subspaces, i.e., for all $k = 0, 1, \ldots$ it holds that

$$\mathcal{K}_\ell \left(A^T A + \lambda_L L^T L, A^T b \right) = \mathcal{K}_\ell \left(A^T A + \lambda_L L^T L - f(x^k) I, A^T b \right).$$

Thus the Krylov space $\mathcal{K}_\ell \left(A^T A + \lambda_L L^T L, A^T b \right)$ is a very suitable search space for all linear systems of the form $\hat{J}(x^k) x^{k+1} = A^T b$.

But since the underlying Newton-like iteration is not guaranteed to converge, it can hardly be assumed that any projection method based on it will do better. Thus we are only considering approximations to iteration (2.6). To make the inverse of the Jacobians more tractable it seems to be a good idea to reformulate $J(x^k)$ in terms of the Jacobian approximation $\hat{J}_k := \hat{J}(x^k)$. This can be accomplished by the Sherman-Morrison formula:
By introducing

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

and assuming

$$(v^k)^T \hat{J}_k^{-1} u^k \neq 1 \quad \text{for all} \quad k = 0, 1, \ldots$$

equation (2.6) can be reformulated into

$$\begin{aligned} x^{k+1} &= x^k - J(x^k)^{-1} q(x^k) \\ &= x^k - \left(\hat{J}_k - u^k (v^k)^T \right)^{-1} (\hat{J}_k x^k - A^T b) \\ &= x^k - \left[\hat{J}_k^{-1} + \frac{1}{1 - (v^k)^T \hat{J}_k^{-1} u^k} \hat{J}_k^{-1} u^k (v^k)^T \hat{J}_k^{-1} \right] (\hat{J}_k x^k - A^T b) \\ &= \hat{J}_k^{-1} A^T b - \frac{1}{1 - (v^k)^T \hat{J}_k^{-1} u^k} \hat{J}_k^{-1} u^k (v^k)^T (x^k - \hat{J}_k^{-1} A^T b). \end{aligned} \tag{3.1}$$

With this formulation the next Newton iterate x^{k+1} is obtained by solving two linear systems with the Jacobian approximation $\hat{J}_k$, i.e., $\hat{J}_k^{-1} A^T b$ and $\hat{J}_k^{-1} u^k$.

Note that the first term $\hat{J}_k^{-1}A^Tb$ is exactly the Newton-like iterate when performing one step with the starting vector x^k. The second term reads $c \cdot \hat{J}_k^{-1}x^k$ with some scalar $c \in \mathbb{R}$, and can be interpreted as the correction due to the Newton-like iteration. Only this second right-hand side x^k (or u^k respectively) depends on k. The system matrices $\hat{J}_k = (A^TA + \lambda_L L^TL - f(x^k)I)$ only differ by multiples of the identity. Hence for approximating the first term $\hat{J}_k^{-1}A^Tb$ a straightforward choice is to use the Krylov space $\mathcal{K}_\ell\left(A^TA + \lambda_L L^TL, A^Tb\right)$. Let us denote V_ℓ as an orthonormal basis of $\mathcal{K}_\ell(B, A^Tb)$ with $B := A^TA + \lambda_L L^TL$. A simple idea for approximating the second term $c \cdot \hat{J}_k^{-1}x^k$ is to restrict the solution of this linear system to the search space $span\{V_\ell\}$ as well, i.e.,

$$\|\hat{J}_k z - x^k\| = \text{min!}, \quad \text{subject to} \quad z \in \mathcal{K}_\ell(B, A^Tb)$$

with z given by

$$V_\ell^T \hat{J}_k V_\ell y = V_\ell^T x^k \quad \text{and} \quad z = V_\ell y. \tag{3.2}$$

In the following Algorithm 3.1 the search space $\mathcal{K}_\ell(B, A^Tb)$ is used for approximating both terms in the update equation (3.1).

Algorithm 3.1 Lanczos Tikhonov TLS Method

Require: Orthonormal basis V_0, 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 $V_k^T \hat{J}_k V_k y_1^k = V_k^T A^T b$ 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 $V_k^T \hat{J}_k V_k y_2^k = V_k^T u^k$ for y_2^k
6: Compute $x^{k+1} = V_k y_1^k - \dfrac{1}{1 - (v^k)^T V_k y_2^k} V_k y_2^k (v^k)^T (x^k - V_k y_1^k)$
7: Compute $q^{k+1} = (A^TA + \lambda_L L^TL - f(x^k)I)x^{k+1} - A^Tb$
8: Orthogonalize $\hat{r} = (I - V_k V_k^T)q^{k+1}$
9: Normalize $v_{\text{new}} = \hat{r}/\|\hat{r}\|$
10: Enlarge search space $V_{k+1} = [V_k, v_{\text{new}}]$
11: **end for**
12: Output: Approximate Tikhonov TLS solution x^{k+1}

It is possible to use different convergence criteria in Line 1:

- 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 absolute value of the last s elements of the vector $y^k := y_1^k - \dfrac{(v^k)^T(x^k - V_k y_1^k)}{1 - (v^k)^T V_k y_2^k} y_2^k$ are several orders of magnitude smaller

than the leading t elements, i.e., recent increases of the search space do not affect the computed solution significantly.

- The residual q^k from Line 7 is sufficiently small, i.e., $\|q^k\|/\|A^T b\|$ is smaller than a given tolerance.

We now discuss how to efficiently determine an approximate solution of the large-scale Tikhonov TLS problem (1.6) with Algorithm 3.1. 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 $A \in \mathbb{R}^{m \times n}$.

- A suitable starting basis V_0 is an orthonormal basis of the Krylov space $\mathcal{K}_\ell \left(B, A^T b\right)$ of small dimension, e.g. $\ell = 5$.

- The main computational cost is building up the search space $\mathcal{V}_k$ of dimension $\ell + k$, with $\mathcal{V}_k := span\{V_k\} = \mathcal{K}_{\ell+k}(B, A^T b)$. If we assume A to be unstructured and L to be sparse, the costs for determining V_k are roughly $2(\ell + k) - 1$ matrix-vector multiplications (MatVecs) with A, i.e., one MatVec for $A^T b$ and $\ell + k - 1$ MatVecs with A and A^T, respectively.

- 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. some reasonable approximation to x_{RTLS}, an additional MatVec has to be spent for computing $f(x^0)$.

- When the matrices $V_k, A^T A V_k, L^T L V_k$ are stored and one column is appended each iteration, no additional MatVecs have to be performed.

- Line 2 can be evaluated as $f(x^k) = \dfrac{x^k (A^T A V y^k) - 2 y^k V_k^T A^T b + \|b\|^2}{1 + \|y^k\|^2}$.

- For the projected right-hand side it simply holds that $V_k^T A^T b = (1, 0, \ldots, 0)^T \in \mathbb{R}^{\ell+k}$.

- It is enough to carry out one LDL^T-decomposition of the projected matrix $V_k^T \hat{J}_k V_k$, which then can be used twice to solve the systems in Line 3 and 5.

- Note that the orthogonalization in Line 8 would not be necessary if the next iterate x^{k+1} in Line 6 is approximated by $V_k y_1^k$ only. This follows directly from the property of the orthogonal projection in Line 3.

- With the vector y^k the residual in Line 7 can be expressed as $q^{k+1} = A^T A V_k y^{k+1} + \lambda_L L^T L V_k y^{k+1} - f(x^k) x^{k+1} - A^T b$. Notice that replacing q^{k+1} by the direction $B V_k(:, end)$ would yield an equivalent expansion of the search space in Line 10.

- The overall cost of Algorithm 3.1 is of the order $\mathcal{O}(mn)$.

Most examples in section 4 show that Algorithm 3.1 gives reasonable approximations to the solution x_{RTLS}, but that it is not possible to obtain a high accuracy with a moderate size of the search space.

Remark 3.1 *One idea to approximate the next Newton iterate in* (3.1) *more accurately than in Algorithm 3.1 is to take into account the second direction* $\hat{J}_k^{-1}x^k$ *more explicitly than in equation* (3.2)*. There exist several possibilities for solving* $\hat{J}_k z = x^k$ *approximately in step* k*. We could perform* m_k *steps of MINRES (cf. [20]) and then add the approximate solution* $\hat{z} \approx \hat{J}_k^{-1}x^k$ *to the current search space. A better idea is to enlarge the initial Krylov space* $\mathcal{K}_\ell(B, A^T b)$ *by the Krylov spaces* $\mathcal{K}_{m_k}(B, x^k)$*, since all MINRES solutions are included therein. A superior approach is to use all the previously gained knowledge when approximating* $\hat{J}_k^{-1}x^k$ *in step* k*, i.e., to restrict the minimization* (3.2) *to the current search space* $\mathcal{V}$*, and then to add the residual* $\hat{J}_k\hat{z} - x^k$ *to the search space.*

But it turns out that only a very good approximation of $\hat{J}_k^{-1}x^k$ *considerably speeds up convergence. In most cases it does not pay to spend much effort in a highly accurate approximation that is only used within the rank-one correction, cf. equation* (3.1)*.*

Initializing Algorithm 3.1 with a Krylov space $\mathcal{K}_\ell$ the iterates x^k are contained in a Krylov space of $A^TA+\lambda_L L^TL$, and due to the convergence properties of the Lanczos process the main contributions come from the first singular vectors of $[A; \sqrt{\lambda_L}L]$ which for small λ_L 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 use the transformation $x := L^{-1}y$ of (1.6) (for general L we had to use the A-weighted generalized inverse $L_A^\dagger$ of L) 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

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

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 3.2.

Algorithm 3.2 Generalized Krylov Subspace Tikhonov TLS Method

Require: Initial basis V_0 with $V_0^T V_0 = I$, 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 $V_k^T \hat{J}_k V_k y_1^k = V_k^T A^T b$ for y_1^k
4: Compute $u^k = 2x^k/(1 + \|x^k\|^2)$ and $v^k = A^T Ax^k - A^T b - f(x^k)x^k$
5: Solve $V_k^T \hat{J}_k V_k y_2^k = V_k^T u^k$ for y_2^k
6: Compute $x^{k+1} = V_k y_1^k - \dfrac{1}{1 - (v^k)^T V_k y_2^k} V_k y_2^k (v^k)^T (x^k - V_k y_1^k)$
7: Compute $q^{k+1} = (A^T A + \lambda_L L^T L - f(x^k) I)x^{k+1} - A^T b$
8: Compute $\tilde{r} = M^{-1} q^{k+1}$
9: Orthogonalize $\hat{r} = (I - V_k V_k^T)\tilde{r}$
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}

The difference between Algorithms 3.1 and 3.2 is the additional Line 8. Suitable convergence criteria in Line 1 can be chosen similarly to Algorithm 3.1. Basically the computational considerations that have been made for Algorithm 3.1 hold true for Algorithm 3.2 as well. Here are some additional comments.

- The main computational cost is again building up the search space $\mathcal{V}_k$. It consists of the initial basis V_0, which can be precomputed, and the k additional directions obtained during the iteration.
- A suitable initial search space is the Krylov space $\mathcal{K}_\ell \left(M^{-1}B, M^{-1}A^T b\right)$ of small dimension, e.g. $\ell = 5$.
- We assume L to be sparse. Then the cost of solving a system with M is less than $\mathcal{O}(mn)$. The costs for determining the orthonormal basis V_0 are $2\ell - 1$ MatVecs. Each subspace enlargement costs another two MatVecs, thus the overall costs for setting up V_k are $2(\ell + k) - 1$ MatVecs.
- By again storing the matrices $V_k, A^T AV_k, L^T LV_k$ and appending one column each iteration no additional MatVecs have to be performed.
- For a moderate number of iterations $k \ll n$ the overall cost of Algorithm 3.2 is of the order $\mathcal{O}(mn)$.

If the search space $span\{V_k\}$ in step k is equal to the Krylov subspace $\mathcal{K}_{\ell+k}\left(M^{-1}\hat{J}_k, M^{-1}A^T b\right)$, then this also holds true for the enlarged search space in Line 11, i.e., $span\{V_{k+1}\} = \mathcal{K}_{\ell+k+1}\left(M^{-1}\hat{J}_k, M^{-1}A^T b\right)$. Hence if the sequence $\{f(x^k)\}$ is constant and if this value has also been considered to build up the initial space, then the search spaces throughout the iteration stay Krylov

spaces. But this is not the generic case, since the $f(x^k)$ are updated in the process. We refer to the search spaces $\mathcal{V}_k$ as generalized Krylov spaces.

For Algorithm 3.2 typically convergence is achieved after a fairly small number of iterations.

Remark 3.2 *It is an interesting idea to leave a degree of freedom in calculating the next iterate x^{k+1} in Line 6 of Algorithm 3.1 or 3.2, i.e., to consider*

$$x^{k+1}(\alpha) = V(y_1^k + \alpha y_2^k)$$

instead of using the fixed value $\alpha = \dfrac{(v^k)^T(x^k - (Vy_1^k))}{1 - (v^k)^T V y_2^k}$. Hence the value of α could be used to minimize the norm of the residual

$$\min_\alpha \|q(x^{k+1}(\alpha))\| = \min_\alpha \|(A^TA + \lambda_L L^TL - f(x^k)I)x^{k+1}(\alpha) - A^Tb\|.$$

The solution of this one dimensional problem is given by

$$\alpha^* = \frac{b^TA\hat{J}_kVy_2^k - (y_1^k)^TV^T\hat{J}_k^2Vy_2^k}{(y_2^k)^TV^T\hat{J}_k^2Vy_2^k}.$$

Alternatively the value of α could be used to minimize

$$\min_\alpha \|(A^TA + \lambda_L L^TL - f(x^{k+1}(\alpha))I)x^{k+1}(\alpha) - A^Tb\|$$

which is a nonlinear minimization problem. It turned out that both ideas hardly ever improve convergence.

Remark 3.3 *If a suitable value of the Tikhonov parameter is not available, it is a straightforward idea to extend Algorithm 3.2 within an L-curve approach. It should pay to reuse the search space when solving a sequence of Tikhonov TLS problems corresponding to a set of values $\lambda_L^i, i = 1, \ldots, k_L$. When solving the $(j+1)$th Tikhonov TLS problem corresponding to the value λ_L^{j+1} we can use the search space $\mathcal{V}_j$ as initial basis, which just has been build up while solving the previous problem for λ_L^j. The stored matrices V_j, A^TAV_j and L^TLV_j can directly be reused. A similar approach for obtaining several values of the L-curve of the quadratically constrained TLS problem* (1.3) *has been presented in [16], where the efficiency of reusing the search space when solving a sequence of eigenvalue problems has been demonstrated.*

4 Numerical Examples

To evaluate the performance of Algorithms 3.1 and 3.2 we use large dimensional test examples from Hansen's *Regularization Tools*, cf. [10]. Most of the problems in this package are discretizations of Fredholm integral equations of the first kind, which are typically very ill-conditioned.

The MATLAB routines `baart, shaw, deriv2(2), deriv2(3), ilaplace(1), ilaplace(3), heat` (κ=1), `heat` (κ=5) and `phillips` provide square matrices $A_{\text{true}} \in \mathbb{R}^{n\times n}$, right-hand sides b_{true} and true solutions x_{true}, with $A_{\text{true}}x_{\text{true}} = b_{\text{true}}$. In all cases the matrices A_{true} and $[A_{\text{true}}, b_{\text{true}}]$ are ill-conditioned. The parameter κ for problem `heat` controls the degree of ill-posedness of the kernel: $\kappa = 1$ yields a severely ill-conditioned and $\kappa = 5$ a mildly ill-conditioned problem. The number in brackets for `deriv2` and `ilaplace` specifies the shape of the true solution, e.g. for `deriv2` the '2' corresponds to a true continuous solution which is exponential while '3' corresponds to a piecewise linear one. The right-hand side is modified correspondingly.

To construct a suitable Tikhonov TLS problem, the norm of b_{true} is scaled such that it holds $\|b_{\text{true}}\| = \max_i \|A_{\text{true}}(:, i)\|$, x_{true} is then scaled by the same factor. The noise added to the problem is put in relation to the norm of A_{true} and b_{true} respectively. Adding a white noise vector $e \in \mathbb{R}^m$ to b_{true} and a matrix $E \in \mathbb{R}^{m\times n}$ to A_{true} yields the error-contaminated problem $\bar{A}x \approx \bar{b}$ with $\bar{b} = b_{\text{true}} + e$ and $\bar{A} = A_{\text{true}} + E$. We refer to the quotient

$$\sigma := \frac{\|e\|}{\|b_{\text{true}}\|} = \frac{\|E\|_F}{\|A_{\text{true}}\|_F}$$

as the *noise level*. In the examples we consider the noise levels $\sigma = 1 \cdot 10^{-2}$ and $\sigma = 1 \cdot 10^{-3}$.

To adapt the problem to an overdetermined linear system of equations we stack two error-contaminated matrices and right-hands (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 $b \in \mathbb{R}^{2n}$. Stacked problems of this kind arise when two measurements of system matrix and right-hand side are available.

A suitable value of the regularization parameter λ_L is determined by using the implementation of the RTLSQEP method described in [13, 15, 17]. Thereby the quadratic constraint δ in (1.3) is set to $\delta = \gamma\|Lx_{\text{true}}\|$, with $\gamma \in [0.8, 1.2]$. Then the RTLS solution is computed, which yields the corresponding value of λ_L. For the large-scale examples the approximations from the RTLSQEP method serve as additional reference solution. For further comparison the results of the RTLSEVP method are given as well, details of the implementation can be found in [14, 17].

The regularization matrix L is chosen to be an approximation of the scaled discrete first order derivative operator in one space-dimension,

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

This regularization matrix has a structure that allows transformation of a regularized least squares problem to standard form. Note that for regularized total

least squares problems there exists no standard form, cf. [11]. In all examples we use the following regular approximation of L:

$$\widetilde{L} = \begin{bmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \\ & & & \varepsilon \end{bmatrix} \in \mathbb{R}^{n \times n}$$

with $\varepsilon = 0.1$. The numerical tests are carried out on an Intel Core 2 Duo T7200 computer with 2.3 GHz and 2GB RAM under MATLAB R2009a (actually our numerical examples require less than 0.5 GB RAM).

In subsection 4.1 the problem `phillips` of small size is investigated in some detail. Algorithms 3.1 and 3.2 are compared to Newton's method (2.6). Several examples from Regularization Tools of dimension 4000×2000 are considered in subsection 4.2.

4.1 Small size problem

In this subsection we investigate the convergence behavior of Algorithm 3.2. The convergence history of the relative residual norm is compared to Algorithm 3.1 and Newton's method. The system matrix $A \in \mathbb{R}^{400 \times 200}$ is obtained by using `phillips`, adding noise of the level $\sigma = 10^{-2}$ and stacking two perturbed matrices as described above. In this example we use $x^0 = x_{RTLS} + e_x$ as starting vector, with a white noise vector e_x and a noise level of $\frac{\|e_x\|}{\|x_{RTLS}\|} = 50\%$. Note that x^0 is not very close to the solution. The RTLS solution x_{RTLS} is obtained by the RTLSQEP method using the value $\delta = 1.0\|Lx_{true}\|$ as quadratic constraint. Details of the RTLSQEP implementation are omitted here, for a detailed description we refer to subsection 4.2. Together with the RTLSQEP solution x_{RTLS}, the corresponding Tikhonov parameter λ_L is determined. Different convergence histories of the Generalized Krylov Subspace Tikhonov Total Least Squares Method (GKS-TTLS) are displayed in Figure 1.

The size of the initial search space is equal to 5. Since no stopping criterion is applied, Algorithm 3.2 actually runs until $\dim(\mathcal{V}) = 200$. Since all quantities shown in Figure 1(a)-(e) quickly converge to machine precision only the first part of each convergence history is shown. In the upper left subplot of Figure 1 the convergence history of $\{f(x^k)\}$ is shown. In every iteration the dimension of the search space is increased by one. Convergence is achieved within 15 iterations, corresponding to a search space of dimension 20. In Figure 1(b) the relative change of $\{f(x^k)\}$ is displayed logarithmically, roughly reaching machine precision after 15 iterations. The Figures 1(c) and (d) show the relative change of the GKS-TTLS iterates $\{x^k\}$ and the norm of the residual $\{q(x^k)\}$, respectively. For a search space dimension of about 20, convergence is reached for these quantities too. Note that convergence does not have to be monotonically decreasing. Figure 1(e) displays logarithmically the first 50 absolute values of the entries in the coefficient vector y^{200}. This stresses the

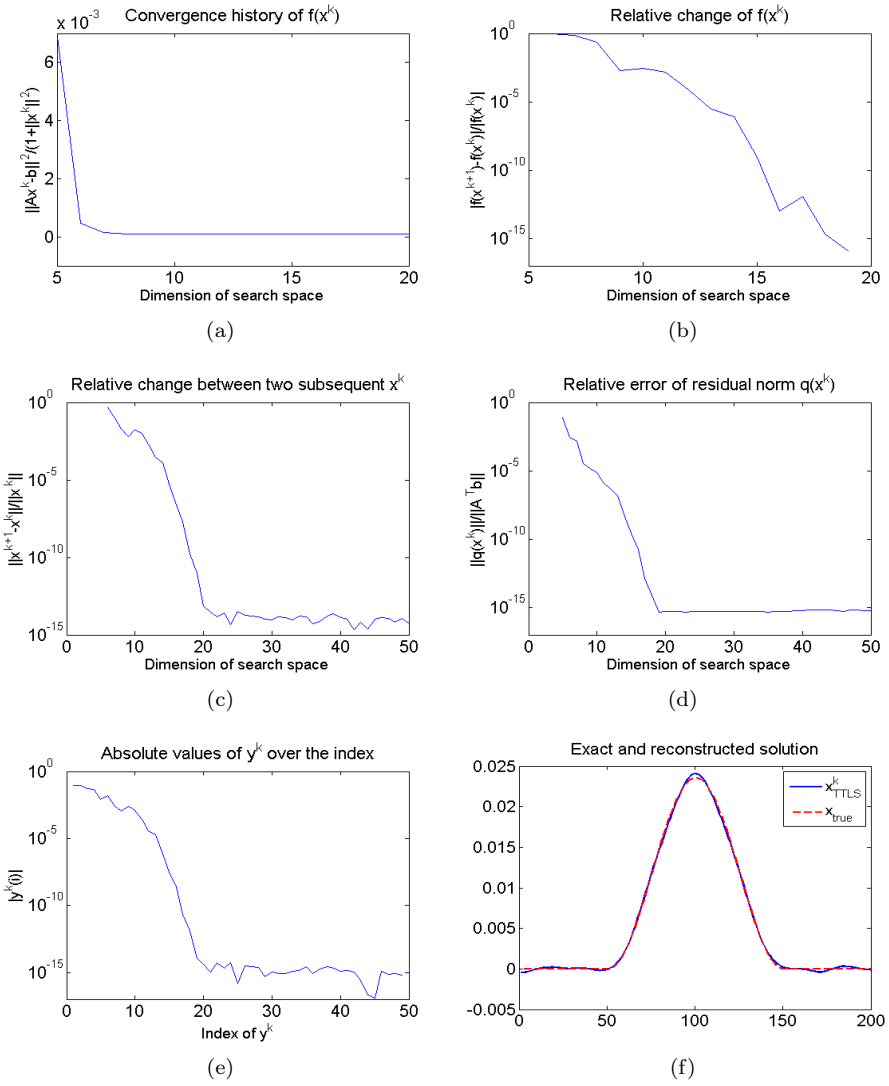
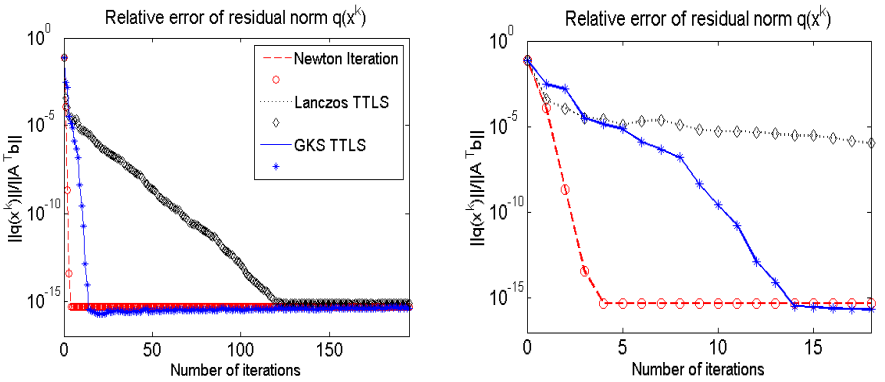


Figure 1: Convergence histories for *phillips*, size 400×200

quality of the first 20 columns of the basis V of the search space. The coefficients corresponding to basis vectors with a column number larger than 20 are basically zero, i.e., around machine precision. In Figure 1(f) the true solution together with the GKS-TTLS approximation x^{15} are shown. The relative error $\|x_{\text{true}} - x^{15}\|/\|x_{\text{true}}\|$ is approximately 2%. Note that the same relative error with respect to the true solution is obtained with Newton's method and the Lanczos TTLS method when a huge search space is used. In Figure 2 the convergence history of $\{q(x^k)\}$ for Algorithm 3.1, Algorithm 3.2 and Newton's method (2.6) are displayed.



(a) (b)

Figure 2: Convergence history of residual norms for *phillips*, size 400×200

In the left subplot of Figure 2 the whole convergence history of the relative residual norms are shown, i.e., until $dim(\mathcal{V}) = 200$ which corresponds to 195 iterations. The right subplot is a close–up of the left one that only displays the first 18 iterations. While Newton's method converges within 4 and the GKS-TTLS method within 14 iterations to machine precision, the Lanczos TTLS method requires 120 iterations. This is a very typical behavior of the Lanczos TTLS method. It is in need of a rather large search space, like here are needed 125 vectors of the $\mathbb{R}^{200}$.

4.2 Large-scale problems

In this subsection we compare the accuracy and performance of Algorithm 3.1, Algorithm 3.2, the RTLSQEP method from [13, 15, 17] and the RTLSEVP method from [14, 17]. Various examples from Hansen's Regularization Tools are employed to demonstrate the efficiency of the proposed GKS-TTLS method. All examples are of the size 4000×2000. The initial vector is chosen as $x^0 = 0$ in all cases, which turns out to be sufficient to converge to the RTLS solutions. The value of λ_L is computed again via the RTLSQEP method. With a value γ from the interval $[0.8, 1.2]$ the quadratic constraint is set to $\delta = \gamma\|Lx_{\text{true}}\|$. The

stopping criterion for the RTLSQEP method is chosen as the relative change of two subsequent values of $f(x^k)$ to be less than 10^{-6}. The initial space is $\mathcal{K}_7(L^{-T}A^TAL^{-1}, A^Tb)$. The RTLSEVP method also solves the quadratically constrained TLS problem (1.5). For all examples it computes almost identical values of λ_L as the RTLSQEP method. The stopping criterion for the RTLS-EVP method is chosen as the residual norm of the first order condition to be less than 10^{-8}, which has also been proposed in [14]. And the starting search space is $\mathcal{K}_5([A,b]^T[A,b],[0,\ldots,0,1]^T)$. For the GKS-TTLS method the size of the initial search space is 5 for all examples and the following stopping criterion is applied: The relative change of two subsequent approximations x^k has to be less than 10^{-12}. For the Lanczos TTLS method an additional convergence criterion is applied, i.e., the dimension of the search space is not allowed to exceed 100 which corresponds to a maximum number of 95 iterations. For all examples 10 different noise realizations are computed and the averaged results can be found in Tables 1 and 2.

In Table 1 the problem *phillips* is investigated with respect to different noise levels, under- and over-regularization. For all problems in Table 1 the residual of the Generalized Krylov Subspace Tikhonov Total Least Squares Method converges to machine precision. The Lanczos TTLS method (L-TTLS) is not very accurate, e.g. with residual norms around 2% for the value $\gamma = 0.9$ while using the same convergence criterion as in Algorithm 3.2. This deficiency is also highlighted in Figure 2. The accuracy of the RTLSQEP and RTLSEVP methods are somewhere in between, where in most examples the latter one yields more accurate approximations. The number of iterations for L-TTLS is always equal to 95, which is the maximum number of iterations. So in none of the examples the convergence criterion $\|x^{k+1} - x^k\|/\|x^k\| < 10^{-12}$ is reached with Algorithm 3.1. The number of iterations is much smaller for RTLSQEP and RTLSEVP compared to GKS-TTLS. This is no surprise since the latter one increases the search space only by one vector every iteration, whereas the RTLSQEP and RTLSEVP methods may add several new vectors in one iteration. More interesting is the number of overall matrix-vector multiplications (MatVecs). For the L-TTLS method the 95 iterations directly correspond to $2 \cdot (\text{MaxIters}+5) - 1 = 199$ MatVecs, see section 3. Similarly for GKS-TTLS it holds the relation $2 \cdot (\text{Iters}+5) - 1 = \text{MatVecs}$. Thus for Algorithms 3.1 and 3.2 the dimension of the search space is the size of the initial space plus the number of iterations. For the RTLSQEP method we are in need of four MatVecs to increase the size of the search space by one, whereas the RTLSEVP method requires only two MatVecs. Hence despite the large number of MatVecs required for RTLSQEP the dimension of the search space often is the smallest of the four algorithms. The GKS-TTLS method outperforms the other three algorithms, i.e., in almost all cases the highest accuracy is obtained with the smallest number of MatVecs. In the final column the relative error with respect to the true solution x_{true} can be found. This quantity shows that all methods yield reasonable approximations, but this is no suitable value for comparison since the purpose of the methods is only to minimize the norm of the residual $q(x)$. The smallest relative errors are obtained with the smallest noise level and $\gamma = 1$. Values of

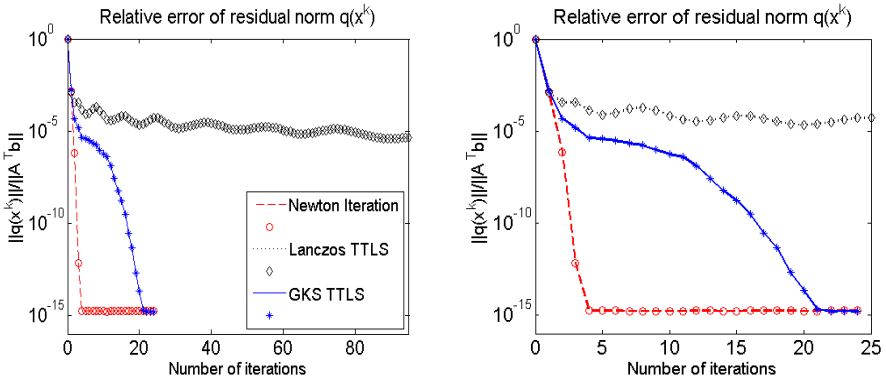
Table 1: Problem *phillips*, size 4000×2000

Problem *noise level* factor γ	Method	$\frac{\|q(x^k)\|}{\|A^T b\|}$	Iters	MatVecs	$\frac{\|x - x_{true}\|}{\|x_{true}\|}$
phillips	GKS-TTLS	8.7e-16	8.0	25.0	8.9e-2
$1e-2$	L-TTLS	1.9e-02	95.0	199.0	1.0e-1
$\gamma = 0.9$	RTLSQEP	5.7e-11	3.0	42.0	8.9e-2
	RTLSEVP	6.4e-13	4.0	47.6	8.9e-2
phillips	GKS-TTLS	7.2e-16	15.9	40.8	1.8e-2
$1e-2$	L-TTLS	6.4e-06	95.0	199.0	1.6e-2
$\gamma = 1.0$	RTLSQEP	1.8e-08	3.0	75.2	1.8e-2
	RTLSEVP	3.9e-08	4.4	60.4	1.8e-2
phillips	GKS-TTLS	7.1e-16	22.6	54.2	6.3e-2
$1e-2$	L-TTLS	6.2e-07	95.0	199.0	6.2e-2
$\gamma = 1.1$	RTLSQEP	5.1e-07	7.0	119.6	6.3e-2
	RTLSEVP	9.3e-08	3.0	65.0	6.2e-2
phillips	GKS-TTLS	8.5e-16	8.0	25.0	8.9e-2
$1e-3$	L-TTLS	1.7e-02	95.0	199.0	1.0e-1
$\gamma = 0.9$	RTLSQEP	5.7e-11	3.0	42.0	8.9e-2
	RTLSEVP	7.1e-13	4.0	47.6	8.9e-2
phillips	GKS-TTLS	7.1e-16	20.9	50.8	6.3e-3
$1e-3$	L-TTLS	1.2e-07	95.0	199.0	6.2e-3
$\gamma = 1.0$	RTLSQEP	2.3e-08	4.2	88.8	6.3e-3
	RTLSEVP	1.9e-08	2.1	60.6	6.3e-3
phillips	GKS-TTLS	7.7e-16	42.0	93.0	4.1e-2
$1e-3$	L-TTLS	3.0e-09	95.0	199.0	4.1e-2
$\gamma = 1.1$	RTLSQEP	1.8e-08	23.1	244.9	4.1e-2
	RTLSEVP	1.5e-12	4.2	73.1	4.1e-2

γ larger than 1 corresponds to a certain degree of under-regularization whereas $\gamma < 1$ corresponds to over-regularization. Here the value $\gamma = 1.1$ leads to a larger number of required MatVecs for Algorithm 3.2 and RTLSQEP. The noise level does not significantly change the effort of the methods.

Table 2 contains the results for a lot of different problems from the Regularization Toolbox. The results are similar to Table 1. The GKS-TTLS method outperforms L-TTLS, RTLSQEP and RTLSEVP in all examples, i.e., the relative residual is computed to machine precision within a search space of fairly small dimension. For most examples the number of MatVecs of Algorithm 3.2 is about $50 - 75\%$ of the MatVecs required for the RTLSQEP and RTLSEVP method. The Lanczos TTLS method is clearly inferior to the other three methods in terms of accuracy and number of MatVecs. The relative error in the last column of Table 2 indicates again suitable computed approximations for all algorithms.

We now pick one example in more detail: Problem *deriv2(2)* of dimension 4000×2000, with noise level 1% and $\gamma = 0.9$. The RTLSQEP method is used to compute the corresponding value of λ_L. In Figure 3 the convergence history of the relative residual norm is displayed. The GKS-TTLS and L-TTLS methods are compared to Newton's method. The starting vector is chosen



(a) (b)

Figure 3: Convergence history of residual norms for *deriv2(2)*, size 4000×2000

to $x^0 = 0$ for all three methods. A short note on computation time. For Newton's method (2.6) it is necessary to set up the Jacobians $J(x^k)$ from (2.5) explicitly. Evaluating the multiplication $A^T A$ takes 1.5s, computing a LDL^T-decomposition of a matrix $J(x^k)$ takes 0.75s, whereas the whole iteration process of GKS-TTLS is finished within 0.5s. For larger examples the computation time scales cubically for Newton's method, but only quadratically for GKS-TTLS, L-TTLS, RTLSQEP and RTLSEVP.

Convergence to machine precision is reached after four iterations with New-

Table 2: Problems from Regularization Tools

Problem *noise level* factor γ	Method	$\frac{\|q(x^k)\|}{\|A^T b\|}$	Iters	MatVecs	$\frac{\|x-x_{true}\|}{\|x_{true}\|}$
baart	GKS-TTLS	2.3e-15	10.1	29.2	1.5e-1
$1e-3$	L-TTLS	1.9e-08	95.0	199.0	1.1e-1
$\gamma = 1.2$	RTLSQEP	1.0e-07	15.7	182.1	1.4e-1
	RTLSEVP	4.1e-10	7.8	45.6	1.5e-1
baart	GKS-TTLS	1.8e-15	15.4	39.8	1.2e-1
$1e-2$	L-TTLS	2.9e-06	95.0	199.0	1.3e-1
$\gamma = 1.1$	RTLSQEP	8.5e-06	10.0	137.8	1.2e-1
	RTLSEVP	7.3e-08	8.6	49.4	1.2e-1
shaw	GKS-TTLS	1.1e-15	10.8	30.6	5.4e-2
$1e-2$	L-TTLS	2.1e-06	95.0	199.0	5.0e-2
$\gamma = 1.0$	RTLSQEP	2.5e-07	3.3	74.9	5.4e-2
	RTLSEVP	1.6e-08	5.0	47.2	5.4e-2
shaw	GKS-TTLS	9.6e-16	8.3	25.6	7.0e-2
$1e-3$	L-TTLS	3.5e-05	95.0	199.0	7.9e-2
$\gamma = 0.9$	RTLSQEP	3.7e-09	4.1	76.1	7.0e-2
	RTLSEVP	2.6e-10	3.0	39.0	7.0e-2
deriv2(2)	GKS-TTLS	8.3e-16	24.6	58.2	9.1e-2
$1e-2$	L-TTLS	3.3e-06	95.0	199.0	9.9e-2
$\gamma = 0.9$	RTLSQEP	3.5e-07	4.7	93.5	9.1e-2
	RTLSEVP	6.4e-10	6.2	79.2	9.1e-2
deriv2(3)	GKS-TTLS	1.2e-15	10.0	29.0	4.9e-2
$1e-3$	L-TTLS	2.0e-04	95.0	199.0	2.9e-2
$\gamma = 0.9$	RTLSQEP	2.3e-09	3.1	52.3	4.9e-2
	RTLSEVP	2.6e-12	5.0	67.0	4.9e-2
ilaplace(1)	GKS-TTLS	2.1e-15	11.0	31.0	1.6e-1
$1e-3$	L-TTLS	1.9e-03	95.0	199.0	1.4e-1
$\gamma = 0.8$	RTLSQEP	5.2e-08	6.0	87.0	1.6e-1
	RTLSEVP	1.4e-10	4.0	61.0	1.6e-1
ilaplace(3)	GKS-TTLS	9.3e-16	13.3	35.6	2.7e-1
$1e-2$	L-TTLS	2.0e-04	95.0	199.0	2.1e-1
$\gamma = 0.8$	RTLSQEP	6.7e-08	7.6	106.6	2.7e-1
	RTLSEVP	7.2e-10	3.0	55.0	2.7e-1
heat(κ=1)	GKS-TTLS	8.4e-16	19.9	48.8	1.5e-1
$1e-2$	L-TTLS	2.2e-04	95.0	199.0	1.6e-1
$\gamma = 0.8$	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)	GKS-TTLS	1.4e-13	25.0	59.0	1.1e-1
$1e-3$	L-TTLS	9.7e-03	95.0	199.0	1.1e-1
$\gamma = 0.8$	RTLSQEP	6.1e-07	4.6	105.2	1.1e-1
	RTLSEVP	9.8e-11	4.0	65.0	1.1e-1

ton's method and after 21 iterations for GKS-TTLS. The L-TTLS method reaches a relative residual of $4 \cdot 10^{-6}$ after 95 iterations, which is the maximum number of iterations and corresponds to a search space of dimension 100. The number of iterations for reaching the stopping criterion $\|x^{k+1} - x^k\|/\|x^k\| < 10^{-12}$ is 24 for Algorithm 3.2, which is close to the average value given in Table 2. Thus the stopping criterion in this case is quite strong, when compared to convergence of the residual. For Algorithm 3.1 the criterion is not reached within the maximum number of iterations, and the attained residual norm is also in accordance with Table 2. The solution of Newton's method and the approximation $x^{21}_{\text{GKS-TTLS}}$ after 21 iterations of GKS-TTLS are almost identical, i.e., it holds that $\|x_{\text{Newton}} - x^{21}_{\text{GKS-TTLS}}\|/\|x_{\text{Newton}}\| = 5.8e - 12$.

5 Conclusions

A new method based on orthogonal projection for solving Tikhonov-regularized total least-squares problems is presented. While approximating Newton iterates, the proposed iterative method solves a convergent sequence of projected linear systems. Due to convergence of this sequence it turns out highly advantageous to reuse the information gathered while solving one system for the solution of the next. Several numerical examples demonstrate that the computed search space is highly suitable. Typically search spaces of fairly small dimension are sufficient. It will be interesting to investigate the straightforward extension of the proposed method to compute points of the L-curve for determining a suitable value of the Tikhonov parameter.

References

[1] A. Beck and A. Ben-Tal. On the solution of the Tikhonov regularization of the total least squares problem. *SIAM J. Optim.*, 17:98 – 118, 2006.

[2] A. Beck, A. Ben-Tal, and M. Teboulle. Finding a global optimal solution for a quadratically constrained fractional quadratic problem with applications to the regularized total least squares problem. *SIAM J. Matrix Anal. Appl.*, 28:425 – 445, 2006.

[3] P. Deuflhard. *Newton Methods for Nonlinear Problems.* Springer, Berlin, 2004.

[4] P. Deuflhard and G. Heindl. Affine invariant convergence theorems for newton's method and extensions to related methods. *SIAM J. Numer. Anal.*, 16(1):1 – 10, 1979.

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

[6] R.D. Fierro, G.H. Golub, P.C. Hansen, and D.P. O'Leary. Regularization by truncated total least squares. *SIAM J. Sci. Comput.*, 18:1223 – 1241, 1997.

[7] G.H. Golub, P.C. Hansen, and D.P. O'Leary. Tikhonov regularization and total least squares. *SIAM J. Matrix Anal. Appl.*, 21:185 – 194, 1999.

[8] G.H. Golub and C.F. Van Loan. *Matrix Computations*. The John Hopkins University Press, Baltimore and London, 3rd edition, 1996.

[9] P.C. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. SIAM, Philadelphia, 1998.

[10] P.C. Hansen. Regularization tools version 4.0 for Matlab 7.3. *Numer. Algorithms*, 46:189 – 194, 2007.

[11] J. Lampe. *Solving Regularized Total Least Squares Problems Based on Eigenproblems*. PhD thesis, Hamburg University of Technology, Institute of Numerical Simulation, 2010.

[12] J. Lampe, M. Rojas, D.C. Sorensen, and H. Voss. Accelerating the LSTRS algorithm. *SIAM J. Sci. Comput.*, 33:175 – 194, 2011.

[13] J. Lampe and H. Voss. On a quadratic eigenproblem occurring in regularized total least squares. *Comput. Stat. Data Anal.*, 52(2):1090 – 1102, 2007.

[14] J. Lampe and H. Voss. A fast algorithm for solving regularized total least squares problems. *Electron. Trans. Numer. Anal.*, 31:12 – 24, 2008.

[15] J. Lampe and H. Voss. Global convergence of RTLSQEP: a solver of regularized total least squares problems via quadratic eigenproblems. *Math. Model. Anal.*, 13:55 – 66, 2008.

[16] J. Lampe and H. Voss. Efficient determination of the hyperparameter in regularized total least squares problems. Technical report, Institute of Numerical Simulation, Hamburg University of Technology, 2009. Appl. Numer. Math. Accepted.

[17] J. Lampe and H. Voss. Solving regularized total least squares problems based on eigenproblems. *Taiwanese J. Math.*, 14:885 – 909, 2010.

[18] S. Lu, S.V. Pereverzyev, and U. Tautenhahn. Regularized total least squares: computational aspects and error bounds. *SIAM J. Matrix Anal. Appl.*, 31:918 – 941, 2009.

[19] I. Markovsky and S. Van Huffel. Overview of total least squares methods. *Signal Processing*, 87:2283 – 2302, 2007.

[20] C.C. Paige and M.A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM J. Numer. Anal.*, 12:617 – 629, 1975.

[21] R.A. Renaut and H. Guo. Efficient algorithms for solution of regularized total least squares. *SIAM J. Matrix Anal. Appl.*, 26(2):457 – 476, 2005.

[22] H. Schwetlick. *Numerische Lösung nichtlinearer Gleichungen*. Oldenbourg R., München, 1979.

[23] D.M. Sima. *Regularization Techniques in Model Fitting and Parameter Estimation*. PhD thesis, Katholieke Universiteit Leuven, Leuven, Belgium, 2006.

[24] D.M. Sima, S. Van Huffel, and G.H. Golub. Regularized total least squares based on quadratic eigenvalue problem solvers. *BIT Numerical Mathematics*, 44:793 – 812, 2004.

[25] S. Van Huffel and P. Lemmerling, editors. *Total Least Squares and Errors-in-Variables Modeling: Analysis, Algorithms and Applications*. Kluwer, Dordrecht, 2002.

[26] S. Van Huffel and J. Vandewalle. *The Total Least Squares Problems: Computational Aspects and Analysis*, volume 9 of *Frontiers in Applied Mathematics*. SIAM, Philadelphia, 1991.

[27] H. Voss. An Arnoldi method for nonlinear symmetric eigenvalue problems. Williamsburg, 2003. Proc. SIAM Conf. Applied Linear Algebra.

[28] H. Voss. An Arnoldi method for nonlinear eigenvalue problems. *BIT Numerical Mathematics*, 44:387 – 401, 2004.