LARGE-SCALE DUAL REGULARIZED TOTAL LEAST SQUARES

JÖRG LAMPE * AND HEINRICH VOSS †

Dedicated to Lothar Reichel on the occasion of his 60th birthday.

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 dual 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.

Key words. total least squares, regularization, ill-posedness, generalized eigenproblem

AMS subject classifications. 65F15, 65F22, 65F30

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

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

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 [23, 28, 29].

The TLS problem (1.2) can be analyzed (cf. [7, 29]) 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 Fredholm integral equations of the first kind (cf. [3, 8]). 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 [4] suggested to filter its solution by truncating the small singular values of the TLS matrix $[A, b]$, and they proposed an

*Germanischer Lloyd SE, D-20457 Hamburg, Germany (joerg.lampe@gl-group.com)
†Institute of Mathematics, Hamburg University of Technology, D-21071 Hamburg, Germany (voss@tuhh.de)

iterative algorithm based on Lanczos bidiagonalization for computing approximate 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

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

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, 6, 11, 13, 14, 15, 16, 22, 24, 25, 26].

If the regularization matrix L is nonsingular, then the solution x_{RTLS} of problem (1.3) is attained. For the more general case of a singular L its existence is guaranteed, if

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

where $F \in \mathbb{R}^{n\times k}$ is a matrix the columns of which form an orthonormal basis of the nullspace of L (cf. [1]).

Assuming inequality (1.4) to hold it is possible to rewrite problem (1.3) into the more tractable form

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

Related to the RTLS problem is the approach of the dual RTLS that has been introduced and investigated in [20, 22, 27]. The dual RTLS (DRTLS) problem is given by

$$\|Lx\| = \min! \quad \text{subject to} \quad (A + \Delta A)x = b + \Delta b, \ \|\Delta b\| \le h_b, \ \|\Delta A\|_F \le h_A, \tag{1.6}$$

where suitable bounds for the noise levels h_b and h_A are assumed to be known. It can be shown that in case the two constraints $\|\Delta b\| \le h_b$ and $\|\Delta A\|_F \le h_A$ are active, the DRTLS problem (1.6) can be reformulated into

$$\|Lx\| = \min! \quad \text{subject to} \quad \|Ax - b\| = h_b + h_A\|x\|. \tag{1.7}$$

Note that due to the two constraint parameters, h_b and h_A, the solution set of the dual RTLS problem is larger than the one of the RTLS problem with only one constraint parameter δ. For every RTLS problem there exists a corresponding dual RTLS problem with an identical solution, but this does not hold vice versa.

In this paper we propose an iterative projection method which combines orthogonal projections to a sequence of generalized Krylov subspaces of increasing dimensions and a one-dimensional root-finding method for the iterative solution of the first order equations of (1.6). Taking advantage of the eigenvalue decomposition of the projected problem, the root-finding can be performed efficiently such that the essential cost of an iteration step are two matrix-vector products. Since usually a very small number of iteration steps is required for convergence the computational complexity of our method is essentially of the order of a matrix-vector product with the matrix A.

The paper is organized as follows. In section 2 basic properties of the dual RTLS problem are summarized, the connection to the RTLS problem is presented and two methods for solving small sized problems are investigated. 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. Dual regularized total least squares. In subsection 2.1 important basic properties of the dual RTLS problem are summarized and connections to related problems are regarded, especially the connection to the RTLS problem (1.3). In subsection 2.2 existing methods for solving small sized dual RTLS problems (1.6) are reviewed, difficulties are discussed and a refined method is proposed.

2.1. Dual RTLS basics. The literature about dual regularized total least squares (DRTLS) is very rare, and they are by far less intensely studied than the RTLS problem (1.3). The origin of the DRTLS probably goes back to Golub who analyzed in [5] the dual regularized least squares problem

$$\|x\| = \min! \quad \text{subject to} \quad \|Ax - b\| = h_b \tag{2.1}$$

assuming an active constraint, i.e., $h_b < \|Ax_{LS} - b\|$ with $x_{LS} = A^{\dagger}b$ as the least squares solution. The results also hold true for the non-standard case $L \neq I$

$$\|Lx\| = \min! \quad \text{subject to} \quad \|Ax - b\| = h_b. \tag{2.2}$$

In [5] an approach with a quadratic eigenvalue problem is presented from which the solution of (2.1) can be obtained. The dual regularized least squares problem (2.2) is exactly the dual RTLS problem with $h_A = 0$, i.e., with no error in the system matrix A.

In the following we review some facts about the dual RTLS problem.

THEOREM 2.1. *[21] If the two constraints $\|\Delta b\| \leq h_b$ and $\|\Delta A\| \leq h_A$ of the dual RTLS problem (1.6) are active, then its solution $x = x_{DRTLS}$ satisfies the equation*

$$(A^T A + \alpha L^T L + \beta I)x = A^T b \tag{2.3}$$

with the parameters α, β solving

$$\|Ax(\alpha, \beta) - b\| = h_b + h_A\|x(\alpha, \beta)\|, \quad \beta = -\frac{h_A(h_b + h_A\|x(\alpha, \beta)\|)}{\|x(\alpha, \beta)\|} \tag{2.4}$$

where $x(\alpha, \beta)$ is the solution of (2.3) for fixed α and β.

In this paper we throughout assume active inequality constraints of the dual RTLS problem, and we mainly focus on the first order necessary conditions (2.3) and (2.4).

REMARK 2.2. *In [19] a related problem is considered, i.e., the generalized discrepancy principle for Tikhonov regularization. The corresponding problem reads:*

$$\|Ax(\alpha) - b\|^2 + \alpha\|Lx(\alpha)\|^2 = \min! \tag{2.5}$$

with the value of α chosen such that it holds

$$\|Ax(\alpha) - b\| = h_b + h_A\|x(\alpha)\|. \tag{2.6}$$

Note that this problem is much easier than the dual RTLS problem. A globally convergent algorithm can be found in [19].

By comparing the solution of the RTLS problem (1.3) and of the dual RTLS problem (1.6), assuming active constraints in either case, basic differences of the two problems can be revealed: using the RTLS solution x_{RTLS} the corresponding corrections of system matrix and right-hand side are given by

$$\Delta A_{RTLS} = \frac{(b - Ax_{RTLS})x_{RTLS}^T}{1 + \|x_{RTLS}\|^2} \quad \text{and} \quad \Delta b_{RTLS} = \frac{Ax_{RTLS} - b}{1 + \|x_{RTLS}\|^2}, \tag{2.7}$$

whereas the corrections for the dual RTLS problem are given by

$$\Delta A_{dRTLS} = h_A \frac{(b - Ax_{dRTLS})x_{dRTLS}^T}{\|(b - Ax_{dRTLS})x_{dRTLS}^T\|_F} \quad \text{and} \quad \Delta b_{dRTLS} = h_b \frac{Ax_{dRTLS} - b}{\|Ax_{dRTLS} - b\|}, \tag{2.8}$$

with x_{dRTLS} as the dual RTLS solution. Notice, that the corrections for the system matrices of the two problems are always of rank one. Identical corrections occur iff it holds $x_{dRTLS} = x_{RTLS}$ and

$$h_A = \frac{\|x_{RTLS}\|\|b - Ax_{RTLS}\|}{1 + \|x_{RTLS}\|^2} \quad \text{and} \quad h_b = \frac{\|Ax_{RTLS} - b\|}{1 + \|x_{RTLS}\|^2}. \tag{2.9}$$

In this case the value for β from (2.4) can also be expressed as

$$\beta = -\frac{h_A(h_b + h_A\|x_{RTLS}\|)}{\|x_{RTLS}\|} = -\frac{\|Ax_{RTLS} - b\|^2}{1 + \|x_{RTLS}\|^2}. \tag{2.10}$$

By the first order conditions the solution x_{RTLS} of problem (1.3) is a solution of the problem

$$(A^TA + \lambda_I I_n + \lambda_L L^T L)x = A^T b, \tag{2.11}$$

where the parameters λ_I and λ_L are given by

$$\lambda_I = -\frac{\|Ax - b\|^2}{1 + \|x\|^2}, \quad \lambda_L = \frac{1}{\delta^2}\big(b^T(b - Ax) - \frac{\|Ax - b\|^2}{1 + \|x\|^2}\big). \tag{2.12}$$

Identical solutions for the RTLS and the dual RTLS problem can be constructed by using the solution x_{RTLS} of the RTLS problem to determine values for the corrections h_A and h_b as stated in (2.9). Then for the solution it holds $\beta = \lambda_I$ (cf. (2.10)) and $\alpha = \lambda_L$, i.e.,

$$\alpha = \frac{1}{\delta^2}\big(b^T(b - Ax_{RTLS}) - \frac{\|Ax_{RTLS} - b\|^2}{1 + \|x_{RTLS}\|^2}\big) \tag{2.13}$$

with $\delta = \|Lx_{RTLS}\|$. This does not hold the other way round, i.e., with the solution x_{dRTLS} of a dual RTLS problem at hand it is in general not possible to construct a corresponding RTLS problem, since the parameter δ cannot be adjusted such that the two parameters of the dual RTLS problem are matched.

REMARK 2.3. *Probably each dual RTLS problem (1.6) is equivalent to a special weighted RTLS problem*

$$\|[\Delta A, \Delta b]\|_F^2 = \min! \quad \textit{subject to} \quad (A + \Delta A)\gamma x = \gamma b + \Delta b, \ \|Lx\| \le \delta, \tag{2.14}$$

with the scaling factor γ. This is beyond the scope of this paper.

2.2. Solving the Dual RTLS problems. Although the formulation (1.7) of the dual RTLS problem looks tractable, this is generally not the case. In [22] suitable algorithms are proposed for special cases of the DRTLS problem, i.e., when it holds $h_A = 0$ or $h_b = 0$ or $L = I$, where the DRTLS problem degenerates to an easier problem. In [27] an algorithm for the general case dual RTLS problem formulation (2.3) and (2.4) is suggested. This idea has been worked out as a special two parameter fixed-point iteration in [20] and [21] where a couple of numerical examples can be found. Note that these methods for solving the dual RTLS problem require the solution of a sequence of linear system of equations, which means that complexity and effort are much higher compared to existing algorithms for solving the related RTLS problem (1.3), cf. [11, 13, 14, 15, 16]. In the following inconsistencies of the two DRTLS methods are investigated and a refined method is worked out.

Let us review the DRTLS algorithm from [27] for computing the dual RTLS solution; it will serve as the basis for the methods developed later in this paper.

Algorithm 1 Dual Regularized Total Least Squares Basis Method

Require: $\varepsilon > 0, A, b, L, h_A, h_b$

Choose starting value $\beta_0 = -h_A^2$

for $i = 0, 1, \dots$ until convergence **do**

  Find pair (x_i, α_i) that solves

$$(A^T A + \beta_i I + \alpha_i L^T L)x_i = A^T b, \text{ s.t. } \|Ax_i - b\| = h_b + h_A\|x_i\| \tag{2.15}$$

  Compute $\beta_{i+1} = -\dfrac{h_A(h_b + h_A\|x_i\|)}{\|x_i\|}$

  Stop if $|\beta_{i+1} - \beta_i| < \varepsilon$

end for

Determine approximate dual RTLS solution $x_{dRTLS} = x_i$

The pseudo code of Algorithm 1 (directly taken from [27]) is not very precise, since the solution of (2.15) is nonunique in general and the correct pair has to be selected. Note that the motivation of Algorithm 1 in [27] is given by the analogy to a similar looking fixed point algorithm for the RTLS problem (1.5), with efficient implementation to be found in [11, 13, 14, 15, 16].

The method proposed in [20] is based on a model function approach for solving the minimization problem

$$\|Ax(\alpha, \beta) - b\|^2 + \alpha\|Lx(\alpha, \beta)\|^2 + \beta\|x(\alpha, \beta)\|^2 = \min! \tag{2.16}$$

subject to the constraints

$$\|Ax(\alpha, \beta) - b\| = h_b + h_A\|x(\alpha, \beta)\| \text{ and } \beta = -h_A^2 - \frac{h_A h_b}{\|x(\alpha, \beta)\|}. \tag{2.17}$$

The corresponding method for solving (2.16) with (2.17) is given below as Algorithm 2, cf. [20]. This approach is shown to work fine for a couple of numerical examples (cf. [20, 21]), but a proof for global convergence is only given for special cases, e.g. for $h_A = 0$. In [18] details about the model function approach for the more general problem of multi-parameter regularization can be found.

Algorithm 2 DRTLS Model Function Approach

Require: $\varepsilon > 0, A, b, L, h_A, h_b$

Choose starting values $\alpha_0 \geq \alpha^* \ \beta_0 = -h_A^2$
for $i = 0, 1, \dots$ until convergence **do**
  Solve $(A^TA + \beta_i I + \alpha_i L^TL)x_i = A^Tb$
  Compute $F_1 = \|Ax_i - b\|^2 + \alpha_i\|Lx_i\|^2 + \beta_i\|x_i\|^2$,
    $F_2 = \|Lx_i\|^2$, $F_3 = \|x_i\|^2$, $D = -(\|b\|^2 - F_1 - \alpha_i F_2)^2/F_3$,
    $T = (\|b\|^2 - F_1 - \alpha_i F_2)/F_3 - \beta_i$
  Update $\beta_{i+1} = -\dfrac{h_A(h_b + h_A\|x_i\|)}{\|x_i\|}$ and compute
    $$N = \|b\|^2 - h_b^2 - \frac{2h_Ah_b\sqrt{-D}}{T+\beta_{i+1}} + \frac{D(T + 2\beta_{i+1} + h_A^2)}{(T+\beta_{i+1})^2}$$
  Update $\alpha_{i+1} = 2\alpha_i^2 F_2/N$
  Stop if $|\alpha_{i+1} - \alpha_i| + |\beta_{i+1} - \beta_i| < \varepsilon$
end for
Solve $(A^TA + \beta_{i+1}I + \alpha_{i+1}L^TL)x_{dRTLS} = A^Tb$ for dual RTLS solution

The following example shows that Algorithm 2 does not necessarily converge to a solution of the dual RTLS problem (1.6).

EXAMPLE 2.4. *It is given the undisturbed problem*

$$A_{true} = \begin{pmatrix} 0.5 & -0.5 \\ 1 & 1 \\ 1 & -1 \end{pmatrix},\ b_{true} = \begin{pmatrix} 0.5 \\ 1 \\ 1 \end{pmatrix} \text{ with the solution } x_{true} = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

which is nicely scaled since the norm of b_{true} *is equal to the norm of a column of* A_{true}*, and thus* $\sqrt{2}\|b_{true}\| = \|A_{true}\|_F$*. It is considered the following noise:*

$$A_{noise} = \begin{pmatrix} 1/\sqrt{2} & 0 \\ 0 & 0 \\ \sqrt{0.14} & 0 \end{pmatrix},\ b_{noise} = \begin{pmatrix} 0.4 \\ 0 \\ -0.4 \end{pmatrix}$$

where it also holds $\sqrt{2}\|b_{noise}\| = \|A_{noise}\|_F$*. The constraints are chosen as* $h_A = \|A_{noise}\|_F = 0.8$ *and* $h_b = \|b_{noise}\| = 0.8/\sqrt{2}$ *and the regularization matrix is given by* $L = \begin{pmatrix} 2 & 0 \\ 1 & 1 \end{pmatrix}$*. When applying Algorithm 2 to this example, with* $\alpha_0 = 100 > \alpha^*$ *and* $\varepsilon = 1e-8$ *the following fixed point is reached after 19 iterations:*

$$x^* = (0.9300, 0.1781)^T \text{ with } \alpha^* = 0,\ \beta^* = -1.1179,\ \|Lx^*\| = 2.1650.$$

Since for the constraint condition (2.17) it holds $\|Ax^* - b\| - (h_b + h_A\|x^*\|) = -0.0356 \neq 0$*, this fixed point is not the solution of the dual RTLS problem.*

The solution of the example is given by

$$x_{dRTLS} = (0.7353, 0.0597)^T \text{ with } \alpha_{dRTLS} = 0.1125,\ \beta_{dRTLS} = -1.2534,$$

with $\|Lx_{dRTLS}\| = 1.6718 < \|Lx^*\|$ *and* $\|Ax_{dRTLS} - b\| - (h_b + h_A\|x_{dRTLS}\|) = 0$.

Example 2.4 shows that Algorithm 2 is not guaranteed to converge to the dual RTLS solution; hence we turn back to the basis Algorithm 1. The main difficulty of Algorithm 1 is the constraint condition in (2.15), i.e., $\|Ax-b\| = h_b + h_A\|x\|$. The task to find a pair (x, α) for a given value of β such that it holds:

$$(A^TA + \beta I + \alpha L^TL)x = A^Tb, \text{ s.t. } \|Ax - b\| = h_b + h_A\|x\|$$

can have a unique solution, more than one solution or no solution. In the following we try to shed some light on this problem.

Let us introduce the function $g(\alpha;\beta) := \|Ax(\alpha) - b\| - h_b - h_A\|x(\alpha)\|$ with $x(\alpha) = (A^TA + \beta I + \alpha L^TL)^{-1}A^Tb$ for a given fixed value of β. In analogy to the solution of RTLS problems we are looking for the rightmost root of g, i.e., the largest value of α, which should be non-negative, cf. [11, 13, 15, 26]. A suitable tool for the investigation of g is the generalized eigenvalue problem (GEVP) of the matrix pair $(A^TA + \beta I, L^TL)$. It is assumed that the regularization matrix L has full rank n, hence the GEVP is definite. Otherwise a spectral decomposition of L^TL could be employed to reduce the GEVP to the range of L, which is not worked out here.

With the decomposition $[V, D] = \texttt{eig}(A^TA + \beta I, L^TL)$ the following relations for the eigenvector matrix V and the corresponding eigenvalue matrix D hold:
With $V^TL^TLV = I$ and $V^T(A^TA + \beta I)V = D$ the matrix $(A^TA + \beta I + \alpha L^TL)$ can be expressed as $V^{-T}(D + \alpha I)V^{-1}$. Hence

$$\begin{aligned} x(\alpha) &= (A^TA + \beta I + \alpha L^TL)^{-1}A^Tb = V(D + \alpha I)^{-1}V^TA^Tb \\ &= V\text{diag}\left\{\frac{1}{d_i + \alpha}\right\}c \end{aligned} \tag{2.18}$$

with $c = V^TA^Tb$. This representation discloses the following properties of the function $g(\cdot) := g(\cdot;\beta) : \mathbb{R}_+ \to \mathbb{R}$:

- From $\lim_{\alpha\to\infty} x(\alpha) = 0$ we immediately get $\lim_{\alpha\to\infty} g(\alpha) = \|b\| - h_b > 0$, which can be assumed to be positive for reasonably posed problems.
- For $(A^TA + \beta I) > 0$ all eigenvalues d_i are positive. Hence $x(\cdot)$ is continuous on $\mathbb{R}_+$, and so is $g(\cdot)$.
- Poles of g can only occur if $A^TA + \beta I$ is indefinite. More precisely, if d_k is a negative eigenvalue of the pencil $(A^TA + \beta I, L^TL)$ and if the corresponding component c_k of $c = V^TA^Tb$ is different from 0, then $\alpha = -d_k$ is a pole of g.
- Let d_k be a simple eigenvalue with $c_k \neq 0$ and let v_k be a corresponding eigenvector. Then it holds that

$$\lim_{\alpha\to -d_k}\left(\frac{d_k+\alpha}{c_k}x(\alpha) - v_k\right) = 0,$$

  from which we get $g(\alpha) \approx f(\alpha)(\|Av_k\| - h_A\|v_k\|)$ with $f(\alpha) = |c_k/(d_k + \alpha)|$ for $\alpha \neq -d_k$ sufficiently close to $-d_k$. Hence, we obtain $\lim_{\alpha\to -d_k} g(\alpha) = +\infty$ for $\|Av_k\| > h_A\|v_k\|$ and $\lim_{\alpha\to -d_k} g(\alpha) = -\infty$ for $\|Av_k\| < h_A\|v_k\|$.
- If it holds $g(0) > 0$ and a simple negative eigenvalue d_k exists with non-vanishing component c_k and $\|Av_k\| < h_A\|v_k\|$, then g has at least two positive roots.
- If $g(0) < 0$ the function $g(\alpha)$ has at least one positive root, independent of the presence of poles.
- If a simple negative leftmost eigenvalue d_n exists, with non-vanishing component c_n and $\|Av_n\| < h_A\|v_n\|$, then it is sufficient to restrict the root-finding of $g(\alpha)$ to the interval $(-d_n, \infty)$, which contains the rightmost root of g.

Since the function $g(\alpha;\beta)$ is not guaranteed to have a root it appears suitable to replace the constraint condition in (2.15) by a corresponding minimization of $g(\alpha;\beta) := \|Ax - b\| - h_b - h_A\|x\|$ in $\mathbb{R}_+$ yielding the following revised Algorithm 3:

Algorithm 3 Dual Regularized Total Least Squares Method

Require: $\varepsilon > 0, A, b, L, h_A, h_b$

Choose starting value $\beta_0 = -h_A^2$

for $i = 0, 1, \dots$ until convergence **do**

Find pair (x_i, α_i) for rightmost $\alpha_i \geq 0$ that solves

$$(A^TA + \beta_i I + \alpha_i L^TL)x_i = A^Tb, \text{ s.t. } \min! = |g(\alpha_i; \beta_i)| \tag{2.19}$$

Compute $\beta_{i+1} = -\dfrac{h_A(h_b + h_A\|x_i\|)}{\|x_i\|}$

Stop if $|\beta_{i+1} - \beta_i| < \varepsilon$

end for

Determine approximate dual RTLS solution $x_{dRTLS} = x_i$

REMARK 2.5. *A note on the idea to extend the domain of the function $g(\alpha)$ to negative values of α, i.e., to eventually keep the root-finding instead of the minimization constraint in equation (2.19). Unfortunately, it is no principle remedy to allow negative values of α. The limit of g for negative values of α is identical to the positive limit, i.e., $g(\alpha)|_{\alpha\to-\infty} = \|b\| - h_b > 0$. Hence, it may happen that after extending $g(\alpha)$ to $\mathbb{R} \to \mathbb{R}$, only poles are present where it holds $\|Av_i\| > h_A\|v_i\|, i = 1, \dots, n$ and thus still no root of g may exist. Notice that α should be positive at the dual RTLS solution in case of active constraints.*

REMARK 2.6. *Note that the quantity $\|Lx\|$ which is to be minimized in the dual RTLS problem does not have to behave monotonic. Non-monotonic behavior may occur for the iterations of Algorithm 3, i.e., for $\|Lx_i\|, i = 0, 1, \dots,$ as well as for the function $\|Lx(\alpha)\|$ within an iteration with a fixed value of β and $x(\alpha) = (A^TA + \beta I + \alpha L^TL)^{-1}A^Tb$. This is in contrast to the quantity $f(x)$ for RTLS problems, cf. [13, 15].*

Let us apply Algorithm 3 to example 2.4. The function $g(\alpha; \beta_0)$ for the starting value $\beta_0 = -h_A^2 = -0.64$ is shown in figure 2.1. For the limit it holds $g(\alpha)|_{\alpha\to\infty} = \|b\| -$

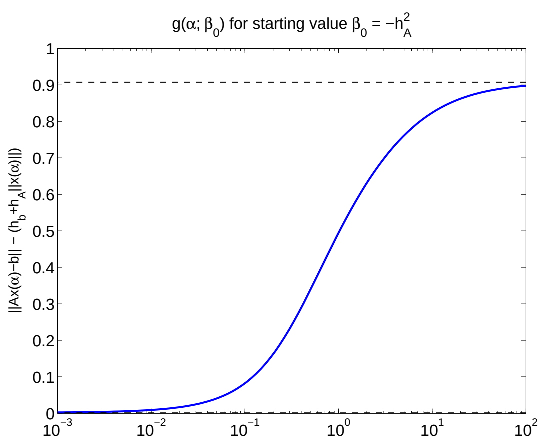


FIGURE 2.1. *Initial function $g(\alpha; \beta_0)$ for example 2.4.*

$h_b = 0.9074$ and for $g(0) = 0.0017$. The eigenvalues of $(A^TA + \beta_0 I)$ are positive and so are the eigenvalues of the matrix pair $(A^TA + \beta_0 I, L^TL)$. Hence, no poles exist for positive values of α. Furthermore, in this example no positive root exists. There do exist negative roots, i.e., the rightmost root is located at $\alpha = -0.0024$, but this is not considered any further, cf. remark 2.5. Thus, in the first iteration of Algorithm 3 the pair $(x_0, \alpha_0) = ([0.7257, 0.0909]^T, 0)$ is selected, as the minimizer of $|g(\alpha; -h_A^2)|$ for all non-negative values of α. In the following iterations the function $g(\alpha, \beta_i), i = 1, \dots$ always has a unique positive root. Machine precision 2^{-52} is reached after 5 iterations of Algorithm 3. The method of choice to find the rightmost root or to find the minimizer of $|g(\alpha)|$ respectively is discussed in section 3. Up to now any one-dimensional minimization method should be fine to solve an iteration of a small sized dual regularized total least squares problem.

REMARK 2.7. *Another interesting approach for obtaining an approximation of the dual RTLS solution is treating the constraints $h_A = \|\Delta A\|_F$ and $h_b = \|\Delta b\|$ separately. In the first stage the value h_A can be used to make the system matrix A better conditioned, e.g. by a shifted SVD, truncated SVD, shifted normal equations or most promising for large scale problems by a truncated bidiagonalization of A. In the second stage the resulting problem has to be solved, i.e., a Tikhonov least squares problem using h_b as discrepancy principle. This means with the corrected matrix $\hat{A} = A + \Delta\hat{A}$ it has to be solved:*

$$\|Lx\| = \min! \quad \textit{subject to} \quad \|\hat{A}x - b\| = h_b.$$

The first order conditions can be obtained from the derivative of the Lagrangian

$$\mathcal{L}(x, \mu) = \|Lx\|^2 + \mu(\|\hat{A}x - b\|^2 - h_b^2).$$

Setting the derivative equal to zero yields

$$(\hat{A}^T\hat{A} + \mu^{-1}L^TL)x = \hat{A}^Tb \quad \textit{subject to} \quad \|\hat{A}x - b\| = \|\Delta\hat{b}\| = h_b,$$

which is just the problem of determining the correct value μ for the Tikhonov least squares problem such that the discrepancy principle holds with equality. Hence a function

$$f(\mu) = \|\hat{A}x_\mu - b\|^2 - h_b^2 \quad \textit{with} \quad x_\mu := (\hat{A}^T\hat{A} + \mu^{-1}L^TL)^{-1}\hat{A}^Tb$$

can be introduced, where its root $\bar{\mu}$ determines the solution $x_{\bar{\mu}}$, cf. [12]. A root exists if it holds that

$$\|\mathcal{P}_{\mathcal{N}(\hat{A}^T)}b\| = \|\hat{A}x_{LS} - b\| < h_b < \|b\| \quad \textit{with} \quad x_{LS} = \hat{A}^\dagger b.$$

Note, that here this condition does not hold automatically, which may lead to difficulties. Another weak point of this approach is that none of the proposed variants in the first stage uses corrections $\Delta\hat{A}$ of small rank, although the solution dual RTLS correction matrix is of rank one, see equation (2.8).

3. Solving large DRTLS problems. 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 system in equation (2.19) in line 3 of Algorithm 3 onto search spaces of much smaller dimensions, and then only to work with the projected problems. In this paper we propose an iterative projection method that computes an approximate solution of (2.19) in a generalized Krylov subspace $\mathcal{V}$, which is then used to solve the corresponding restricted minimization problem $\min! = |g_V(\alpha_i; \beta_i)|$ with $g_V(\alpha; \beta) := \|AVy - b\| - h_b - h_A\|Vy\|$ and the columns of V as an orthonormal basis of $\mathcal{V}$. For the use of generalized Krylov subspaces in related problems see [12, 17]. The minimization of $|g_V(\alpha; \beta)|$ is in almost all practical problems equal to the determination of the rightmost root of $g_V(\alpha; \beta)$. Therefore in the following only root-finding methods are considered for solving the minimization constraint. The zero can be computed, e.g., by bracketing algorithms that enclose the rightmost root, and it turned out to be beneficial to use rational inverse interpolation (see [14, 16]). Having determined the root α_i for a value of β_i, a new value β_{i+1} is calculated. These inner iterations are carried out until the projected dual RTLS problem is solved. Only then the search space $\mathcal{V}$ is expanded by the residual of the original linear system (2.19). After expansion, a new projected DRTLS problem has to be solved, i.e., zero-finding and updating of β is repeated until convergence. The outer subspace enlargement iterations are performed until α, β or $x(\beta) = Vy(\beta)$ satisfies a stopping criterion. Since the expansion direction depends on the parameter α, the search space $\mathcal{V}$ is not a Krylov subspace. Numerical examples illustrate that the stopping criterion typically is satisfied for search spaces $\mathcal{V}$ of fairly small dimension.

The cost of enlarging the dimension of the search space by one is $\mathcal{O}(mn)$ arithmetic floating point operations, and so is the multiplication of the matrix $(A^TA + \beta I + \alpha L^TL)$ by a vector. This cost is higher than the determination of the dual RTLS solution of a projected problem. We therefore solve the complete projected DRTLS problem after each increase of $\dim(\mathcal{V})$ by one. The resulting method is given in the following Algorithm 4.

Algorithm 4 Generalized Krylov Subspace Dual RTLS Method

Require: $\varepsilon > 0, A, b, L, h_A, h_b$ and initial basis V_0, $V_0^TV_0 = I$

1: Choose starting value $\beta_0^0 = -h_A^2$
2: **for** $j = 0, 1, \dots$ until convergence **do**
3: **for** $i = 0, 1, \dots$ until convergence **do**
4: Find pair $(y(\beta_i^j), \alpha_i^j)$ for rightmost $\alpha_i^j \geq 0$ that solves

$$V_j^T(A^TA + \beta_i^j I + \alpha_i^j L^TL)V_j y(\beta_i^j) = V_j^T A^T b, \text{ s.t. } \min! = |g_{V_j}(\alpha_i^j; \beta_i^j)| \quad (3.1)$$

5: Compute $\beta_{i+1}^j = -\dfrac{h_A(h_b + h_A\|y(\beta_i^j)\|)}{\|y(\beta_i^j)\|}$
6: Stop if $|\beta_{i+1}^j - \beta_i^j|/|\beta_i^j| < \varepsilon$
7: **end for**
8: Compute $r^j = (A^TA + \beta_i^j I + \alpha_i^j L^TL)V_j y(\beta_i^j) - A^Tb$
9: Compute $\hat{r}^j = M^{-1}r^j$
10: Orthogonalize $\tilde{r}^j = (I - V_jV_j^T)\hat{r}^j$
11: Normalize $v_{\text{new}} = \tilde{r}^j/\|\tilde{r}^j\|$
12: Enlarge search space $V_{j+1} = [V_j, v_{\text{new}}]$
13: **end for**
14: Determine approximate dual RTLS solution $x_{dRTLS} = V_j y(\beta_i^j)$

Algorithm 4 iteratively adjusts the parameters α and β and builds up a search space simultaneously. Generally, "convergence" is achieved already for search spaces of fairly small dimension; see section 4. Most of the computational work is done in line 8 for computing the residual, since solving the projected dual RTLS problem in lines 3–7 is comparably inexpensive.

We can use several convergence criteria in line 2:

- Stagnation of the sequence $\{\beta^j\}$, i.e., the relative change of two consecutive values β^j at the solution of the corresponding dual RTLS problems is small: $|\beta^{j+1} - \beta^j|/|\beta^j|$ is smaller than a given tolerance.
- Stagnation of the sequence $\{\alpha^j\}$, i.e., the relative change of two consecutive values

α^j at the solution of the corresponding dual RTLS problems is small: $|\alpha^{j+1} - \alpha^j|/|\alpha^j|$ is smaller than a given tolerance.
- The relative change of two consecutive Ritz vectors $x(\beta^j) = V_j y(\beta^j)$ at the solution of projected DRTLS problems is small, i.e., $\|x(\beta^{j+1}) - x(\beta^j)\|/\|x(\beta^j)\|$ is smaller than a given tolerance.
- The absolute values of the last s elements of the vector $y(\beta^j)$ at the solution of a projected DRTLS problem are several orders of magnitude smaller than the first t elements, i.e., recent increases of the search space does not affect the computed solution significantly.
- The residual r^j from line 8 is sufficiently small, i.e., $\|r^j\|/\|A^T b\|$ is smaller than a given tolerance.

We now discuss how to efficiently determine an approximate solution of the large-scale dual RTLS problem (1.6) with Algorithm 4. 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 (MatVec) with a (general) dense matrix $A \in \mathbb{R}^{m\times n}$.

- The algorithm can be used with or without preconditioner. If no preconditioner should be used it is set $M = I$, thus line 9 can be neglected. When a preconditioner is used, it is suggested to choose $M = L^T L$ if $M > 0$ and L is sparse, and otherwise to employ a positive definite sparse approximation $M \approx L^T L$. For solving systems with M a Cholesky decomposition has to be computed once. The cost of this decomposition is less than $\mathcal{O}(mn)$, as well as solving a system with M afterwards.
- A suitable starting basis V_0 is an orthonormal basis of the Krylov space $\mathcal{K}_\ell\left(M^{-1}A^T A, M^{-1}A^T b\right)$ of small dimension, e.g. $\ell = 5$.
- The main computational cost of Algorithm 4 is building up the search space $\mathcal{V}_j$ of dimension $\ell + j$, with $\mathcal{V}_j := span\{V_j\}$. If we assume A to be unstructured and L to be sparse, the costs for determining V_j are roughly $2(\ell + j) - 1$ matrix-vector multiplications with A, i.e., one MatVec for $A^T b$ and $\ell + j - 1$ MatVecs with A and A^T, respectively. If L is dense the costs roughly double.
- An outer iteration is started with the previously determined value of β from the last iteration, i.e., $\beta_0^{j+1} := \beta_i^j, j = 0, 1, \ldots$.
- When the matrices $V_j, AV_j, A^T AV_j, L^T LV_j$ are stored and one column is appended each iteration, no additional MatVecs have to be performed.
- With the matrix $V_j \in \mathbb{R}^{n\times(\ell+j)}$ having orthonormal columns and $y = (V_j^T(A^T A + \beta_i^j I + \alpha L^T L)V_j)^{-1} V_j^T A^T b$ it can be evaluated $g_{V_j}(\alpha; \beta_i)$ as $\|AV_j y - b\| - h_b - h_A\|y\|$, where it holds $\|V_j y\| = \|y\|$.
- Instead of solving the projected linear system (3.1) several times, it is sufficient to solve the eigenproblem of the projected pencil $(V_j^T(A^T A + \beta_i^j I)V_j, V_j^T L^T LV_j)$ once for every β_i^j, which then can be used to obtain an analytic expression for $y(\alpha) = (V_j^T(A^T A + \beta_i^j I + \alpha L^T L)V_j)^{-1} V_j^T A^T b$, cf. equations (2.18) and (3.2). This enables efficient root-finding algorithms for $|g_{V_j}(\alpha_i^j; \beta_i^j)|$.
- With the vector $y^j = y(\beta_i^j)$ the residual in line 8 can be expressed as $r^j = A^T AV_j y^j + \alpha_i^j L^T LV_j y^j + \beta_i^j x(\beta_i^j) - A^T b$. Note that in exact arithmetic the direction $\bar{r} = A^T AV_j y^j + \alpha_i^j L^T LV_j y^j + \beta_i^j x(\beta_i^j)$ leads to the same new expansion v_{new}.
- For a moderate number of outer iterations $j \ll n$ the overall cost of Algorithm 4 is of the order $\mathcal{O}(mn)$.

The expansion direction of the search space in iteration j depends on the current values

α_i^j, β_i^j, see line 8. Since both parameters are not constant throughout the algorithm, the search space is no Krylov space but a generalized Krylov space, cf. [12, 17].

A few words concerning the preconditioner. Most examples in section 4 show that Algorithm 4 gives reasonable approximations to the solution x_{dRTLS} also without preconditioner, but that it is not possible to obtain a high accuracy with a moderate size of the search space. In [17] the preconditioner $M = L^T L$ or an approximation $M \approx L^T L$ has been successfully applied for solving the related Tikhonov RTLS problem, and in [14, 16] a similar preconditioner has been employed for solving the eigenproblem occurring in the RTLSEVP method of [24]. For Algorithm 4 with preconditioner convergence is typically achieved after a fairly small number of iterations.

3.1. Zero-finding methods. For practical problems the minimization constraint condition in (3.1) almost always is equal to the determination of the rightmost root of $g_{V_j}(\alpha; \beta_i^j)$. Thus, in this paper we focus on the use of efficient zero-finders, which are in need of a cheap evaluation of the constraint condition for a given pair $(y(\beta_i^j), \alpha)$. As introduced in subsection 2.2 it is beneficial for the investigation of $g_{V_j}(\alpha; \beta_i^j)$ to compute the corresponding eigendecomposition of the projected problem. It is assumed that the projected regularization matrix $V_j^T L^T L V_j$ is of full rank, which directly follows from the full rank assumption of $L^T L$, but may even hold for singular $L^T L$. An explicit expression for $y(\alpha)$ can be derived analogously to the expression for $x(\alpha)$ in equation (2.18). With the decomposition $[W, D] = \texttt{eig}(V_j^T A^T A V_j + \beta_i^j I, V_j^T L^T L V_j) = \texttt{eig}(V_j^T (A^T A + \beta_i^j I, L^T L) V_j)$ of the projected problem the following relations for the eigenvector matrix W and the corresponding eigenvalue matrix D hold:
With $W^T V_j^T L^T L V_j W = I$ and $W^T V_j^T (A^T A + \beta_i^j I) V_j W = D$ the matrix $V_j^T (A^T A + \beta_i^j I + \alpha L^T L) V_j$ can be expressed as $W^{-T}(D + \alpha I) W^{-1}$. Hence it holds

$$
\begin{aligned}
y(\alpha; \beta_i^j) &= \left(V_j^T (A^T A + \beta_i^j I + \alpha L^T L) V_j \right)^{-1} V_j^T A^T b \\
&= W (D + \alpha I)^{-1} W^T V_j^T A^T b = W \mathrm{diag} \left\{ \frac{1}{d_i + \alpha} \right\} c \qquad (3.2)
\end{aligned}
$$

with $c = W^T V_j^T A^T b$ and $V \in \mathbb{R}^{n \times (\ell + j)}$. For the function $g_{V_j}(\alpha; \beta_i^j)$ the characterization regarding poles and zeros as stated in subsection 2.2 for $g(\alpha; \beta)$ holds accordingly. So, after determining the eigenvalue decomposition in an inner iteration for an updated value of β_i^j, all evaluations of the constraint condition are then available at almost no costs.

We are in a position to discuss the design of efficient zero-finders. Newton's method is an obvious candidate. This method works well if a fairly accurate initial approximation of the rightmost zero is known. However, if our initial approximation is larger than and not very close to the desired zero, then the first Newton step is likely to give a worse approximation of the zero than the initial approximation; see figure 4.1 for a typical plot of $g(\alpha)$. The function g is seen to be flat for large values of $\alpha > 0$, and may contain several poles.

Let us review some facts about poles and zeros of $g_V(\alpha) := g_{V_j}(\alpha; \beta_i^j)$ that can be exploited for zero-finding methods.

The limit is given by $g_V(\alpha)|_{\alpha \to \infty} = \|b\| - h_b$, which is equal to the limit of the original function $g(\alpha)$, and should be positive for a reasonably posed problem where the correction of b is assumed to be smaller than the norm of the right-hand side itself. Assuming simple eigenvalues and the ordering $d_1 > \cdots > d_{m-1} > 0 > d_m > \cdots > d_{\ell+j}$, the shape of g_V can be characterized as follows:

- If no negative eigenvalue occurs $g_V(\alpha)$ has no poles for $\alpha > 0$ and nothing can be exploited.
- For every negative eigenvalue d_k, $k = m, \dots, \ell + j$, it can be evaluated $\|AV_j w_k\| - h_A\|w_k\|$, with w_k as the corresponding eigenvector, i.e., the kth column of the eigenvector matrix $W \in \mathbb{R}^{(\ell+j)\times(\ell+j)}$. If it holds $c_k \neq 0$, with c_k as the kth component of the vector $c = W^T V_j^T A^T b$, and if $\|AV_j w_k\| - h_A\|w_k\| > 0$, then the function $g_V(\alpha)$ has a pole at $\alpha = -d_k$ with $\lim_{\alpha\to -d_k} g_V(\alpha) = +\infty$. And if it holds $\|AV_j w_k\| - h_A\|w_k\| < 0$ with $c_k \neq 0$ then $g_V(\alpha)$ has a pole at $\alpha = -d_k$ with $\lim_{\alpha\to -d_k} g_V(\alpha) = -\infty$.
- The most frequent case in practical problems is the occurrence of a negative smallest eigenvalue $d_{\ell+j} < 0$ with non-vanishing component $c_{\ell+j}$ and $\|AV_j w_{\ell+j}\| < h_A\|w_{\ell+j}\|$. Then it is sufficient to restrict the root finding to the interval $(-d_{\ell+j}, \infty)$, which contains the rightmost root. This information can directly be exploited in a bracketing zero-finding algorithm.
- Otherwise the smallest negative eigenvalue corresponding to the rightmost pole of $g_V(\alpha)$ with $\lim_{\alpha\to -d_k} g_V(\alpha) = -\infty$ is determined, i.e., the smallest eigenvalue $d_k, k = m, \dots, \ell + j$ for which it holds $c_k \neq 0$ and $\|AV_j w_k\| < h_A\|w_k\|$. This rightmost pole is then used as lower bound value for a bracketing zero-finder, i.e., the interval is restricted to $(-d_k, \infty)$.

In this paper two suitable bracketing zero-finding methods are suggested. As a standard bracketing algorithm for determining the root in the interval $(-d_{\ell+j}, \infty)$, $(-d_k, \infty)$ or $[0, \infty)$ it is chosen the King method, cf. [10]. The King method is an improved version of the Pegasus method, such that after each secant step a modified step has to follow.

In a second bracketing zero-finder a suitable model function for g_V is used, cf. also [12, 14, 16]. Since the behavior at the rightmost root is not influenced much by the rightmost pole but much more by the asymptotic behavior of g_V as $\alpha \to \infty$, it is reasonable to incorporate this knowledge. Thus, we derive a zero-finder based on rational inverse interpolation, which takes this behavior into account. Consider the model function for the inverse of $g_V(\alpha)$,

$$g_V^{-1} \approx h(g) = \frac{p(g)}{g - g_\infty} \quad \text{with a polynomial} \quad p(g) = \sum_{i=0}^{k-1} a_i g^i, \tag{3.3}$$

where for the pole it holds $g_\infty = \|b\| - h_b$ which is independent of the search space $\mathcal{V}$. The degree of the polynomial can be chosen depending on the information of g_V that is to be used in each step. We propose to use three function values, i.e., $k = 3$. This choice yields a small linear systems of equations with a $k \times k$ matrix that have to be solved in each step.

Let us consider the use of three pairs $\{\alpha^i, g_V(\alpha^i)\}$, $i = 1, 2, 3$; see also [14]. Assume that the following inequalities are satisfied,

$$\alpha^1 < \alpha^2 < \alpha^3 \quad \text{and} \quad g_V(\alpha^1) > 0 > g_V(\alpha^3); \tag{3.4}$$

otherwise we renumber the α^i so that (3.4) holds.

If g_V is strictly monotonically decreasing in $[\alpha^1, \alpha^3]$, then (3.3) is a rational interpolant of $g_V^{-1} : [g_V(\alpha^3), g_V(\alpha^1)] \to \mathbb{R}$. Our next iterate is $\alpha_{\text{new}} = h(0)$, where the polynomial $p(g)$ is of degree 2. The coefficients a_0, a_1, a_2 are computed by solving the equations $h(g_V(\alpha^i)) = \alpha^i$, $i = 1, 2, 3$, which we formulate as a linear system of equations with a 3×3 matrix. In exact arithmetic, $\alpha_{\text{new}} \in (\alpha^1, \alpha^3)$, and we replace α^1 or α^3 by α_{new}, so that the new triplet satisfies (3.4).

Due to round-off errors or possible non monotonic behavior of g, the computed value α_{new} might not be contained in the interval (α^1, α^3). In this case we carry out a bisection step, so that the interval is guaranteed to still contain the zero. If we have two positive values $g_V(\alpha^i)$, then we let $\alpha^1 = (\alpha^2 + \alpha^3)/2$; in the case of two negative values $g_V(\alpha^i)$, we let $\alpha^3 = (\alpha^1 + \alpha^2)/2$.

4. Numerical examples. To evaluate the performance of Algorithm 4 we use large dimensional test examples from Hansen's *Regularization Tools*, cf. [9]. 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(1)`, `deriv2(2)`, `deriv2(3)`, `ilaplace(2)`, `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 dual RTLS problem, the norm of b_{true} is scaled such that it holds $\sqrt{n}\|b_{\text{true}}\| = \|A_{\text{true}}\|_F$, 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 $e \in \mathbb{R}^n$ to b_{true} and a matrix $E \in \mathbb{R}^{n\times n}$ 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

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

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.

Suitable values of constraint parameters are given by $h_A = \gamma\|E\|_F$ and $h_b = \gamma\|e\|$, with $\gamma \in [0.8, 1.2]$.

For the small-scale example the model function approach of Algorithm 2 as well as the refined Algorithm 3 and the iterative projection Algorithm 4 are applied, using the two proposed zero-finders.

For the large-scale examples two methods for solving the related RTLS problem are evaluated additionally for further comparison. The implementation of the RTLSQEP method is described in [13, 15, 16], and details of the RTLSEVP implementation can be found in [14, 16]. For both algorithms the value of the quadratic constraint δ in (1.3) is set to $\delta = \gamma\|Lx_{\text{true}}\|$. Please note that the dual RTLS problem and the RTLS problem have different solutions, cf. subsection 2.1.

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

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

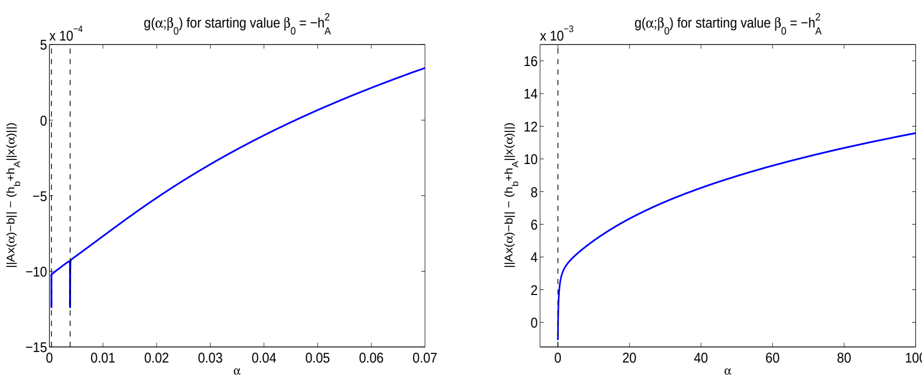
In all examples we use the following regular approximation of L:

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

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 *heat*$(\kappa=1)$ of small size is investigated in some detail. The projection Algorithm 4 is compared to the full DRTLS method described in Algorithm 3 and to the model function Algorithm 2. 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 4. The convergence history of the relative approximation error norm is compared to Algorithm 2 and the full DRTLS Algorithm 3. The system matrix $A \in \mathbb{R}^{400\times 200}$ is obtained by using *heat*$(\kappa=1)$, adding noise of the level $\sigma = 10^{-2}$ and stacking two perturbed matrices as described above. The initial value for β is given by $\beta_0 = -h_A^2 = -3.8757e-5$ and the corresponding initial function $g(\alpha;\beta_0)$ is displayed in figure 4.1.



(a) (b)

FIGURE 4.1. *Initial function $g(\alpha)$ for small size example.*

The function $g(\alpha;\beta_0)$ has 182 poles for $\alpha > 0$, with the rightmost pole at $\alpha = -d_{200} = 0.0039$ and the second rightmost pole at $-d_{199} = 0.00038$ as indicated by the dashed lines in figure 4.1. For these poles it holds $\lim_{\alpha\to -d_k} g(\alpha) = -\infty$ since $\|Av_k\| - h_A\|v_k\| < 0, k = n-1, n$. In the left subplot it can be observed that the occurrence of the poles does not influence the behavior at the zero $\alpha_0 = 0.0459$. In the right subplot the behavior for large values of α is displayed. The limit value is given by $g(\alpha;\beta_0)|_{\alpha\to\infty} = g_\infty = 0.0435$.

Figure 4.2 displays the convergence history of the Generalized Krylov Subspace Dual Regularized Total Least Squares Method (GKS DRTLS) using the preconditioner $M = \widetilde{L}^T \widetilde{L}$ for different convergence measures.

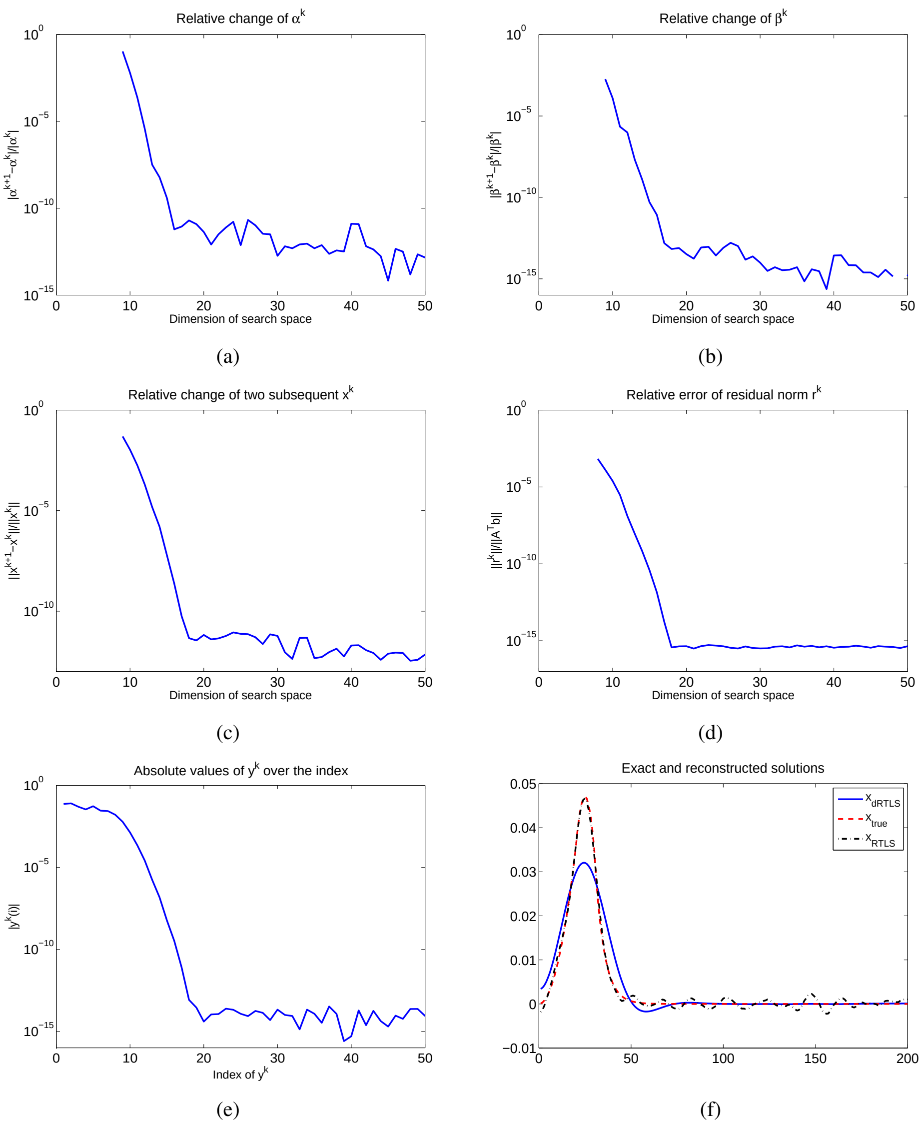
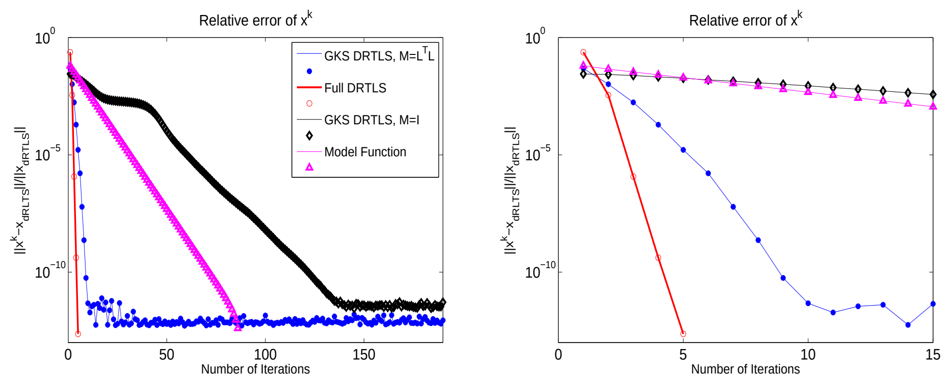


FIGURE 4.2. *Convergence histories for* heat(1)*, size* 400×200*.*

The size of the initial search space is equal to 8. Since no stopping criterion for the outer iterations is applied, Algorithm 4 actually runs until $\dim(\mathcal{V}) = 200$. Since all quantities shown in figure 4.2(a)-(d) quickly converge only the first part of each convergence history is shown. It can be observed that not all quantities converge to machine precision which is due to the convergence criteria used within an inner iteration. Note that for each subspace enlargement in the outer iteration a DRTLS problem of the dimension of the current search space has to be solved. For the solution of these projected DRTLS problems a zero-finder is applied, which in the following is referred to as inner iterations. For the computed $heat(1)$

example the convergence criteria have been chosen as $1e-12$ for the relative error of $\{\beta^k\}$ in the inner iterations, and also $1e-12$ for the relative error of $\{\alpha^k\}$ and for the absolute value of $g_{V_j}(\alpha^k;\beta_i^j)$ within the zero-finder. In the upper left subplot of figure 4.2 the convergence history of $\{\alpha^k\}$ is shown. In every outer iteration the dimension of the search space is increased by one. Convergence is achieved within 12 iterations, corresponding to a search space of dimension 20. In figure 4.2(b) the relative change of $\{\beta^k\}$ is displayed logarithmically, roughly reaching machine precision after 12 iterations. The figures 4.2(c) and (d) show the relative change of the GKS DRTLS iterates $\{x^k\}$, i.e., the approximate solutions $V_j y(\beta_i^j)$ obtained from the projected DRTLS problems, and the norm of the residual $\{r^k\}$, 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 4.2(e) displays logarithmically the first 50 absolute values of the entries in the coefficient vector y^{200}. This stresses the 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 4.2(f) the true solution together with the GKS-TTLS approximation x^{12} are shown. The relative error $\|x_{\text{true}} - x^{12}\|/\|x_{\text{true}}\|$ is approximately 30%. Note that the same relative error with respect to the true solution is obtained with the GKS DRTLS method without preconditioner, the full DRTLS method and the model function approach. The RTLS solution x_{RTLS} has a relative error of $\|x_{\text{true}} - x_{RTLS}\|/\|x_{\text{true}}\| = 8\%$, but it has to be stressed that this corresponds to the solution of a different problem. The dual RTLS solution does not exactly match the peak of x_{true}, but on the other hand does not show the ripples from the RTLS solution. In figure 4.3 the convergence history of the relative error norms of $\{x^k\}$ with respect to the solution x_{dRTLS} are displayed for Algorithm 4 with and without preconditioner, the model function Algorithm 2 and the full DRTLS Algorithm 3.



(a) (b)

FIGURE 4.3. *Convergence history of approximations x^k for* heat(1)*, size* 400×200*.*

In the left subplot of Figure 4.3 the whole convergence history of the approximation error norms of both GKS DRTLS iterates are shown, i.e., until $dim(\mathcal{V}) = 200$ which corresponds to 192 outer iterations. As mentioned above machine precision is not reached due to the applied convergence criteria for the inner iterations, i.e., it is reached a relative approximation error of $1e-12$. Additionally the convergence history of Algorithms 2 and 3 to the the same approximation level is shown. The right subplot is a close–up of the left one that only displays the first 15 iterations. While the full DRTLS method converges within 5 and the GKS DRTLS method with the preconditioner $M = \widetilde{L}^T\widetilde{L}$ in about 12 iterations to the required accuracy, the GKS DRTLS method without preconditioner requires 140 iterations.

This is a very typical behavior of the GKS DRTLS method without preconditioner, i.e., it is in need of a rather large search space, like here are needed 140 vectors of the $\mathbb{R}^{200}$. The model function approach was started with the initial value $\alpha_0 = 1.5\alpha^*$, with $\alpha^* = 0.04702$ as the value at the solution x_{dRTLS}. Despite the good initial value the required number of iterations was 85, where in each iteration of Algorithm 2 a different large linear system of equations has to be solved. The main effort of one iteration of the full DRTLS method 3 is computing a large eigendecomposition; such that the zero-finding problem can then be carried out at negligible costs. Hence, the costs of the full DRTLS method are much less compared to the model function approach. Note that the costs for obtaining the approximation x^{12} of the GKS DRTLS method with preconditioner are essentially only 39 MatVecs, i.e., 15 for the building up the initial space and 24 for the resulting search space $\mathcal{V} \in \mathbb{R}^{200\times20}$.

A few words concerning the zero-finders for the full DRTLS method and the GKS DRTLS Algorithm 4. We start the bracketing zero-finders by first determining values α^k, such that not all $g(\alpha^k;\beta_i)$ or $g_{V_j}(\alpha^k;\beta_i^j)$ are of the same sign. Such values can be determined by multiplying available values of the parameter α by 0.01 or 100 depending on the sign of $g(\alpha,\beta_i)$. After very few steps this gives an interval that contains a root of $g(\alpha,\beta_i)$. For the King method two values $\alpha^k, k = 1,2$ with $g(\alpha^1;\beta_i)g(\alpha^2;\beta_i) < 0$ are sufficient for initialization, while for the rational inverse interpolation three pairs $(\alpha^k, g(\alpha^k;\beta_i)), k = 1,2,3$ have to be given with not all $g(\alpha^k;\beta_i), k = 1,2,3$ having the same sign. For Algorithm 3 the initial value for α is chosen as $\alpha^1 = -1.1d_{200} = 0.0043$, with d_{200} as the smallest eigenvalue of $(A^TA + \beta_0 I, \widetilde{L}^T\widetilde{L})$. This initial guess is located slightly right from the rightmost pole, see also figure 4.1. For the GKS DRTLS method 4 no pole of $g_{V_0}(\alpha;\beta_0^0)$ for the initial search space $V_0 \in \mathbb{R}^{200\times8}$ exists, thus the initial value was set to $\alpha^1 = 1$. Note that nevertheless $g_{V_0}(\alpha;\beta_0^0)$ does have a positive root. When, subsequently, the dimension of the search space is increased, the initial value for the parameter β_0^{j+1} is set equal to the last determined value β_i^j. And the first value of the parameter α, i.e., α^1, used during initialization for the zero-finding problems $g_{V_j}(\alpha;\beta_i^j) = 0, i = 1,2,\dots$ is set equal to the last calculated value α_i^{j-1}.

Tables 4.1 and 4.2 show the number of outer and inner iterations as well as the iterations required for the zero-finder within one inner iteration for the full DRTLS method and the generalized Krylov subspace DRTLS method with and without preconditioner. In Table 4.1 the iterations required for the Algorithm 3 are compared to the inner and outer iterations of Algorithm 4 when no preconditioner is applied, i.e., with $M = I$. The King method and the rational inverse interpolation zero-finder introduced in subsection 3.1 are compared, for solving all the inner iterations. The first outer iteration of the GKS DRTLS method is treated separately since it corresponds to solving the projected DRTLS with the starting basis V_0, where no information from previous iterations can be used as initial guess for the parameters α and β. Thus, this leads to a number of 5 or 6 inner iterations, i.e., updates of β_i^0, depending on the applied zero-finder. The iterations required by the zero-finder is 6 and 7 respectively for determining the very first update of β_0^0, i.e., β_1^0. The effort for determining the subsequent values $\beta_i^0, i = 2,3,\dots$ drastically decreases, e.g. for the rational inverse interpolation zero finder determining β_2^0 requires 2 and determining β_3^0 requires only 1 iteration of the zero finder. Determining the zeros in the following 60 outer iterations only consists of $3-4$ inner iterations each time. After more than 60 outer iterations have been carried out, i.e., for the dimension of the search space it holds $\dim(\mathcal{V}_j) = 68$, typically one or two inner iterations are sufficient for solving the projected DRTLS problem. Note that a '0' in Table 4.1 for of the number of iterations of a zero-finder means that the corresponding initialization was sufficient to fulfill the convergence criteria. The King method and the rational inverse interpolation scheme perform similarly. The full DRTLS does not carry out any outer projection

TABLE 4.1
Number of iterations for Full and GKS DRTLS with $M = I$.

Zero-finder	Alg.	Outer iters	Inner iters	1st it.	2nd it.	3rd it.	ith it.
Rat.-Inv.	GKS	1	6	6	2	1	0
Rat.-Inv.	GKS	2–60	3–4	1–2	0–1	0	0
Rat.-Inv.	GKS	>60	1–3	0–1	0	0	-
Rat.-Inv.	Full	-	6	6	2	1	0
King	GKS	1	5	7	3	3	0–1
King	GKS	2–60	3–4	2–3	1–3	0–1	0–1
King	GKS	>60	1–2	0–1	0	-	-
King	Full	-	5	7	3	3	0–1

iterations and directly treats the full problem. So the meaning of inner iterations as updating the parameter β is identical for Algorithms 3 and 4.

In Table 4.2 the number of iterations required for the full DRTLS algorithm is compared to the GKS DRTLS method when the preconditioner $M = \widetilde{L}^T\widetilde{L}$ is applied. Table 4.2 shows

TABLE 4.2
Number of iterations for Full and GKS DRTLS with $M = \widetilde{L}^T\widetilde{L}$.

Zero-finder	Alg.	Outer iters	Inner iters	1st it.	2nd it.	3rd it.	ith it.
Rat.-Inv.	GKS	1	5	6	2	1	0
Rat.-Inv.	GKS	2–8	2–5	0–3	0–1	0	0
Rat.-Inv.	GKS	>8	1	0	-	-	-
Rat.-Inv.	Full	-	6	6	2	1	0
King	GKS	1	5	6	3	2	0
King	GKS	2–8	3–4	1–4	0–3	0–1	0–1
King	GKS	>8	1	0	-	-	-
King	Full	-	5	7	3	3	0–1

a similar behavior as already observed in Table 4.1: The King method and rational inverse interpolation zero-finder perform comparably good; and the larger the iteration number of an inner iteration gets the less iterations are required by the zero-finder. In contrast to the method without preconditioner here are needed much less outer iterations for convergence. Convergence of the GKS DRTLS method corresponds to an almost instant solution of the zero finder in only one inner iteration. Note that no convergence criterion for stopping the outer iterations has been applied.

4.2. Large-scale examples. In this subsection we compare the accuracy and performance of Algorithm 4 with and without preconditioner, the RTLSQEP method from [13, 15, 16] and the RTLSEVP method from [14, 16]. Various examples from Hansen's Regularization Tools are employed to demonstrate the efficiency of the proposed Generalized Krylov Subspace Dual RTLS method. All examples are of the size 4000×2000. With a value γ from the interval $[0.8, 1.2]$ the quadratic constraint of the RTLS problem is set to $\delta = \gamma\|Lx_{\text{true}}\|$ and the constraints for the dual RTLS are set to γh_A and γh_b respectively. 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 (1.3). For all examples it computes almost identical values of $\lambda_L = \alpha$ as the RTLSQEP method. The stopping criterion for the RTLSEVP 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]. The starting search space is $\mathcal{K}_5([A,b]^T[A,b],[0,\dots,0,1]^T)$.

For the GKS-DRTLS method the dimension of the initial search space is 6 for all examples unless stated differently and the following stopping criterion is applied: The relative change of subsequent approximations for α and β in two outer iterations has to be less than 10^{-10}. For the variant without preconditioner an additional convergence criterion is applied: the dimension of the search space is not allowed to exceed 100 which corresponds to a maximum number of 94 iterations. For all examples 10 different noise realizations are computed and the averaged results can be found in Tables 4.3 and 4.4.

In Table 4.3 several problems from the Regularization Toolbox [9] are investigated with respect to under- and over-regularization for the noise level $\sigma = 1e-2$. For all problems in Table 4.3 the residual of the GKS-DRTLS method with preconditioner (denoted as 'DRTLS') converges to almost machine precision. The variant without preconditioner (denoted as 'DRTLSnp') is not very accurate, e.g. with residual norms between $0.01 - 10\%$ while using the same convergence criterion. This deficiency is also highlighted in figure 4.3. The accuracy of the RTLSQEP and RTLSEVP methods are somewhere in between, where in most examples the latter one yields more accurate approximations. In the second column the relative error of the corresponding constraint condition is given: for Algorithm 4 this is $|g(\alpha^*;\beta^*)|/(h_b + h_A\|x_{dRTLS}\|) = |(\|Ax_{dRTLS} - b\| - h_b - h_A\|x_{dRTLS}\|)|/(h_b + h_A\|x_{dRTLS}\|)$ and for the RTLS methods this is $|(\|Lx_{RTLS} - \delta)|/\delta$. The constraint condition within the DRTLS methods is fulfilled with almost machine precision, while for the used implementations of the RTLS methods this quantity varies with the underlying problem. The number of iterations for DRTLSnp is always equal to the maximum number of iterations, which is 94 in most cases. For *heat(1)* and *heat(5)* the dimension of the initial search space was increased to 8 and 10, respectively, to ensure the function $g_{V_0}(\alpha;\beta_0^0)$ to have a positive root. Note that this is not essential for Algorithm 4 if it is equipped with a minimizer of $|g_{V_j}(\alpha_i^j;\beta_i^j)|$ and not only a zero finder. In none of the examples the convergence criteria $|\alpha_i^{j+1} - \alpha_i^j|/\alpha_i^j| < 10^{-10}$ and $|\beta_i^{j+1} - \beta_i^j|/\beta_i^j| < 10^{-10}$ are reached by Algorithm 4 without preconditioner, while the variant with $M = \widetilde{L}^T\widetilde{L}$ always converged. The DRTLS and DRTLSnp algorithm increase the search space 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 DRTLSnp method the 94 iterations directly correspond to $2 \cdot (\text{MaxIters}{+}6) - 1 = 199$ MatVecs, see section 3. Similarly for the variant with preconditioner it holds the relation $2 \cdot (\text{Iters}{+}6) - 1 = \text{MatVecs}$. Thus for Algorithms 4 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 smaller than for RTLSEVP.

The CPU times in the next to last column are given in seconds. They are closely related to the number of MatVecs, since these are the most expensive operations within all four algorithms. Thus, the main part of the CPU time is required for computing the MatVecs, i.e., roughly 60% for GKS-DRTLS method without preconditioner and $80 - 90\%$ for the other three algorithms. Note that the CPU time for simply computing 100 matrix vector multiplications with $A \in \mathbb{R}^{4000\times 2000}$ is about 1.7 seconds. The DRTLS method outperforms the other

TABLE 4.3
Problems from Regularization Tools, noise level $\sigma = 1e-2$.

Problem factor γ	Method	$\frac{\|r^j\|}{\|A^T b\|}$	Constr.	Iters	Mat-Vecs	CPU time	$\frac{\|x-x_{true}\|}{\|x_{true}\|}$
shaw	DRTLS	1.4e-13	2.6e-13	3.0	17.0	0.32	3.4e-1
$\gamma = 1.2$	DRTLSnp	7.7e-02	1.0e-12	94.0	199.0	6.47	2.6e-1
	RTLSQEP	4.0e-07	3.8e-05	6.7	104.3	3.01	1.2e-1
	RTLSEVP	1.3e-12	1.1e-02	4.0	54.2	0.92	1.2e-1
baart	DRTLS	5.7e-11	5.0e-15	1.9	14.8	0.31	2.1e-1
$\gamma = 1.1$	DRTLSnp	1.1e-01	5.4e-14	94.0	199.0	6.10	1.9e-1
	RTLSQEP	2.8e-06	3.0e-02	6.3	100.7	2.82	1.3e-1
	RTLSEVP	1.7e-12	1.8e-02	2.0	40.8	0.77	1.2e-1
phillips	DRTLS	1.5e-11	6.4e-15	3.4	17.8	0.33	1.0e-1
$\gamma = 1.1$	DRTLSnp	3.8e-03	5.3e-14	94.0	199.0	5.94	1.0e-1
	RTLSQEP	8.1e-05	7.1e-01	9.5	141.9	1.88	7.9e-2
	RTLSEVP	2.4e-12	1.3e-02	2.6	62.4	1.15	6.1e-2
heat(1)	DRTLS	1.7e-11	1.5e-13	7.3	29.6	0.58	3.1e-1
$\gamma = 1.0$	DRTLSnp	1.2e-03	2.2e-14	92.0	199.0	6.30	3.1e-1
	RTLSQEP	7.6e-07	6.4e-06	17.8	212.4	3.67	6.5e-2
	RTLSEVP	5.2e-11	1.6e-06	5.1	78.0	1.48	6.5e-2
heat(5)	DRTLS	1.5e-08	9.8e-15	14.0	47.0	0.87	8.9e-2
$\gamma = 1.0$	DRTLSnp	9.5e-04	2.5e-13	90.0	199.0	5.89	8.9e-2
	RTLSQEP	2.5e-04	6.1e-04	23.4	212.4	4.14	8.3e-3
	RTLSEVP	1.7e-04	8.6e-04	3.5	76.6	1.30	6.6e-3
deriv2(1)	DRTLS	1.1e-13	2.1e-13	3.0	17.0	0.32	3.3e-1
$\gamma = 1.0$	DRTLSnp	3.3e-02	1.8e-13	94.0	199.0	6.22	3.4e-1
	RTLSQEP	9.3e-07	1.7e-04	15.6	194.6	3.50	1.1e-1
	RTLSEVP	9.8e-13	1.4e-09	5.1	77.0	1.39	1.1e-1
deriv2(2)	DRTLS	2.2e-13	6.3e-13	3.0	17.0	0.34	2.9e-1
$\gamma = 0.9$	DRTLSnp	3.5e-02	6.3e-14	94.0	199.0	6.66	3.0e-1
	RTLSQEP	7.6e-07	1.9e-04	5.1	101.1	1.89	9.0e-2
	RTLSEVP	5.9e-14	1.2e-08	6.1	78.6	1.43	9.0e-2
deriv2(3)	DRTLS	1.6e-13	5.5e-13	3.0	17.0	0.36	2.0e-1
$\gamma = 0.9$	DRTLSnp	1.4e-01	2.6e-12	94.0	199.0	6.51	1.4e-1
	RTLSQEP	1.1e-07	2.3e-09	3.0	54.8	1.02	5.1e-2
	RTLSEVP	2.3e-13	2.8e-10	5.4	67.2	1.22	5.1e-2
ilaplace(2)	DRTLS	4.7e-12	5.5e-14	5.0	21.0	0.43	3.4e-1
$\gamma = 0.8$	DRTLSnp	8.8e-03	2.2e-13	94.0	199.0	6.30	7.9e-1
	RTLSQEP	2.3e-07	9.8e-07	4.0	79.4	1.44	4.2e-1
	RTLSEVP	3.5e-12	5.5e-03	1.4	46.8	0.84	4.1e-1
ilaplace(3)	DRTLS	9.3e-13	3.7e-11	17.7	46.4	0.98	3.9e-1
$\gamma = 0.8$	DRTLSnp	3.8e-04	1.3e-14	94.0	199.0	6.29	2.6e-1
	RTLSQEP	1.1e-06	2.0e-09	5.0	84.0	1.51	2.6e-1
	RTLSEVP	1.3e-11	2.9e-08	3.0	48.6	0.87	2.6e-1

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 DRTLS methods are solving a different problem compared to the RTLS methods. The smallest relative errors are obtained with $\gamma = 1$. Values of γ larger than 1 corresponds to a certain degree of under-regularization whereas $\gamma < 1$ corresponds to over-regularization.

Table 4.4 contains the results of the problems considered in Table 4.3 but now with the noise level reduced to $\sigma = 1e-3$. The results are similar to Table 4.3. The GKS-DRTLS with preconditioner outperforms DRTLSnp, RTLSQEP and RTLSEVP in all examples, i.e., the relative residual is computed to almost machine precision within a search space of fairly small dimension. For the examples *heat(1)* and *heat(5)* the dimension of the initial search space was now increased to 12 and 16 and for both *ilaplace* examples to 9, to ensure the function $g_{V_0}(\alpha; \beta_0^0)$ to have a positive root. Note that for problem *heat(5)* with the noise level $\sigma = 1e-3$ the DRTLSnp method converges for several noise realizations to the required accuracy, whereas for all other examples the maximum number of iterations is reached. For most examples the number of MatVecs of Algorithm 4 with $M = \widetilde{L}^T\widetilde{L}$ is often only about $10-50\%$ of the MatVecs required for the RTLSQEP and RTLSEVP method. The DRTLSnp 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 4.4 indicates again suitable computed approximations for all algorithms.

5. Conclusions. A new method based on orthogonal projection for solving dual regularized total least squares problems is presented. The proposed iterative method solves a convergent sequence of projected two parameter linear systems with minimization constraint. 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.

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 (2006), pp. 98–118.

[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 (2006), pp. 425–445.

[3] H. ENGL, M. HANKE, AND A. NEUBAUER, *Regularization of Inverse Problems*, Kluwer, Dordrecht, The Netherlands, 1996.

[4] 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 (1997), pp. 1223–1241.

[5] G.H. GOLUB, *Some Modified Matrix Eigenvalue Problems*, SIAM Review, 15 (1973), pp. 318–334.

[6] G.H. GOLUB, P.C. HANSEN, AND D.P. O'LEARY, *Tikhonov regularization and total least squares*, SIAM J. Matrix Anal. Appl., 21 (1999), pp. 185–194.

[7] G.H. GOLUB AND C.F. VAN LOAN, *Matrix Computations*, The John Hopkins University Press, Baltimore and London, 3rd ed., 1996.

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

[9] P.C. HANSEN, *Regularization tools version 4.0 for Matlab 7.3*, Numer. Alg., 46 (2007), pp. 189–194.

[10] R.F. KING, *An improved Pegasus-method for root finding.*, BIT, 13 (1973), pp. 423–427.

[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, L. REICHEL, AND H. VOSS, *Large-Scale Tikhonov Regularization via Reduction by Orthogonal Projection*, Linear Alg. Appl., 436 (2012), pp. 2845–2865.

TABLE 4.4
Problems from Regularization Tools, noise level $\sigma = 1e-3$.

Problem factor γ	Method	$\frac{\|r^j\|}{\|A^Tb\|}$	Constr.	Iters	Mat-Vecs	CPU time	$\frac{\|x-x_{true}\|}{\|x_{true}\|}$
shaw	DRTLS	2.5e-14	3.9e-13	3.0	17.0	0.32	2.3e-1
$\gamma = 1.2$	DRTLSnp	6.1e-04	2.4e-12	94.0	199.0	6.12	2.1e-1
	RTLSQEP	4.3e-08	6.5e-08	15.2	183.4	1.51	9.6e-2
	RTLSEVP	5.3e-13	1.2e-01	1.0	44.6	0.81	5.7e-2
baart	DRTLS	8.7e-14	1.2e-12	1.9	14.8	0.28	9.5e-2
$\gamma = 1.1$	DRTLSnp	2.3e-03	1.8e-12	94.0	199.0	6.06	1.2e-1
	RTLSQEP	8.2e-08	3.9e-07	11.2	150.6	1.61	1.1e-1
	RTLSEVP	1.5e-12	1.1e-01	1.0	31.4	0.54	7.9e-2
phillips	DRTLS	1.3e-13	1.1e-13	4.8	22.6	0.43	2.7e-2
$\gamma = 1.1$	DRTLSnp	1.6e-04	1.5e-14	94.0	199.0	6.04	2.7e-2
	RTLSQEP	2.3e-08	1.1e-06	20.0	228.2	4.01	3.8e-2
	RTLSEVP	4.5e-10	3.3e-02	1.0	62.0	1.12	3.8e-2
heat(1)	DRTLS	8.7e-13	4.1e-12	11.1	45.2	0.81	1.1e-1
$\gamma = 1.0$	DRTLSnp	4.7e-06	2.1e-13	88.0	199.0	5.91	1.1e-1
	RTLSQEP	8.0e-09	2.3e-10	23.4	261.6	4.68	2.7e-2
	RTLSEVP	5.7e-09	4.7e-02	3.2	87.0	1.42	4.9e-2
heat(5)	DRTLS	7.3e-09	2.3e-12	23.8	78.6	1.59	1.3e-2
$\gamma = 1.0$	DRTLSnp	1.4e-06	9.8e-12	82.1	195.2	5.36	1.3e-2
	RTLSQEP	1.1e-03	4.3e-02	24.0	301.4	5.78	2.1e-2
	RTLSEVP	1.4e-05	9.6e-04	2.0	78.0	1.29	2.1e-3
deriv2(1)	DRTLS	3.3e-14	2.2e-13	7.0	25.0	0.47	1.7e-1
$\gamma = 1.0$	DRTLSnp	4.6e-04	3.6e-14	94.0	199.0	5.97	1.9e-1
	RTLSQEP	3.5e-08	3.9e-07	22.2	238.8	3.91	5.3e-2
	RTLSEVP	3.5e-11	3.2e-05	5.3	84.6	1.42	4.9e-2
deriv2(2)	DRTLS	1.8e-14	2.7e-13	7.0	25.0	0.50	1.5e-1
$\gamma = 0.9$	DRTLSnp	5.1e-04	7.1e-14	94.0	199.0	6.41	1.7e-2
	RTLSQEP	4.9e-08	2.3e-06	18.8	217.4	4.22	4.2e-2
	RTLSEVP	7.4e-12	5.9e-06	4.5	80.6	1.48	4.2e-2
deriv2(3)	DRTLS	1.0e-13	6.4e-14	4.0	19.0	0.37	7.3e-2
$\gamma = 0.9$	DRTLSnp	2.0e-03	2.6e-13	94.0	199.0	5.99	4.5e-2
	RTLSQEP	4.0e-09	3.8e-09	3.0	52.0	0.96	4.9e-2
	RTLSEVP	1.4e-13	2.1e-10	5.0	63.2	1.15	4.9e-2
ilaplace(2)	DRTLS	2.5e-13	2.5e-13	4.9	26.8	0.51	3.8e-1
$\gamma = 0.8$	DRTLSnp	2.4e-04	2.2e-13	91.0	199.0	6.01	7.7e-1
	RTLSQEP	2.6e-08	1.5e-07	9.2	128.6	2.39	4.1e-1
	RTLSEVP	4.5e-13	1.4e-03	1.0	44.6	0.80	4.1e-1
ilaplace(3)	DRTLS	1.3e-13	1.2e-12	12.7	42.4	0.85	1.4e-1
$\gamma = 0.8$	DRTLSnp	4.9e-06	1.8e-13	91.0	199.0	6.05	1.0e-1
	RTLSQEP	7.5e-07	2.2e-09	5.0	83.2	1.50	2.5e-1
	RTLSEVP	3.7e-11	2.8e-08	3.0	45.0	0.81	2.5e-1

[13] J. LAMPE AND H. VOSS, *On a quadratic eigenproblem occurring in regularized total least squares*, Comput. Stat. Data Anal., 52 (2007), pp. 1090–1102.

[14] J. LAMPE AND H. VOSS, *A fast algorithm for solving regularized total least squares problems*, Electron. Trans. Numer. Anal., 31 (2008), pp. 12–24.

[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 (2008), pp. 55–66.

[16] J. LAMPE AND H. VOSS, *Solving Regularized Total Least Squares Problems Based on Eigenproblems*, Taiwanese J. Math., 14 (2010), pp. 885–909.

[17] J. LAMPE AND H. VOSS, *Large-Scale Tikhonov Regularization of Total Least Squares*, J. Comput. Appl. Math., 238 (2013), pp. 95–108.

[18] S. LU AND S.V. PEREVERZEV, *Multi-parameter regularization and its numerical realization*, Numerische Mathematik, 118 (2011), pp. 1–31.

[19] S. LU, S.V. PEREVERZEV, Y. SHAO, AND U. TAUTENHAHN, *On the generalized discrepancy principle for Tikhonov regularization in Hilbert scales*, J. Integ. Equ. Appl., 22 (2010), pp. 483–517.

[20] S. LU, S.V. PEREVERZEV, AND U. TAUTENHAHN, *Dual regularized total least squares and multi-parameter regularization*, J. Comput. Appl. Math., 8 (2008), pp. 253–262.

[21] S. LU, S.V. PEREVERZEV, AND U. TAUTENHAHN, *A model function method in total least squares*, Applicable Analysis, 89 (2010), pp. 1693–1703.

[22] S. LU, S.V. PEREVERZEV, AND U. TAUTENHAHN, *Regularized total least squares: computational aspects and error bounds*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 918–941.

[23] I. MARKOVSKY AND S. VAN HUFFEL, *Overview of total least squares methods*, Signal Processing, 87 (2007), pp. 2283 – 2302.

[24] R. RENAUT AND H. GUO, *Efficient algorithms for solution of regularized total least squares*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 457–476.

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

[26] D.M. SIMA, S. VAN HUFFEL, AND G.H. GOLUB, *Regularized total least squares based on quadratic eigenvalue problem solvers*, BIT, 44 (2004), pp. 793–812.

[27] U. TAUTENHAHN, *Regularization of linear ill-posed problems with noisy right hand side and noisy operator*, Inverse and Ill-posed Problems, 16 (2008), pp. 507–523.

[28] S. VAN HUFFEL AND P. LEMMERLING, *Total Least Squares and Errors-in-Variables Modeling: Analysis, Algorithms and Applications*, Kluwer, Dordrecht, 2002.

[29] S. VAN HUFFEL AND J. VANDEVALLE, *The Total Least Squares Problems: Computational Aspects and Analysis*, vol. 9 of Frontiers in Applied Mathematics, SIAM, Philadelphia, 1991.