Efficient determination of the Hyperparameter in Regularized Total Least Squares Problems

Jörg Lampe * Heinrich Voss †

January 28, 2009

Keywords: total least squares, regularization, ill-posedness, Nonlinear Arnoldi method, L-curve

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

Abstract

The total least squares (TLS) method is a successful approach for linear problems if both the system matrix and the right hand side are contaminated by some noise. For ill-posed TLS problems regularization is necessary to stabilize the computed solution. In this paper we suggest the use of the L-curve for the determination of the regularization parameter. The focus is on efficient implementation with particular emphasis on the reuse of information gained during the convergence history.

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 only available by measurements or if A is an idealized approximation of the true operator then both the matrix A and the right hand side b are contaminated by some noise.

*Institute of Numerical Simulation, Hamburg University of Technology, D-21071 Hamburg, Germany (joerg.lampe@tu-harburg.de). The work was supported by Philips Research Germany and the German Federal Ministry of Education and Research (BMBF) under grant number 13N9079.

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

An appropriate approach to this problem often 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 [27, 33].

The TLS problem (1.2) can be analyzed (cf. [14, 33]) 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 $\sigma'_n > \sigma_{n+1}$ where σ'_n denotes the smallest singular value of A, and it is then given by

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

Usually the problems to solve are ill-conditioned, for example the discretization of ill-posed problems such as integral equations of the first kind (cf. [7, 18]). 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 [8] 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.

Beck and Ben-Tal adopted the Tikhonov regularization concept to stabilize the TLS solution in [3].

Closely related to Tikhonov regularization is the well established approach 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 (as in the whole paper) $\|\cdot\|$ denotes the Euclidean norm, $\delta > 0$ is the regularization parameter, and the regularization matrix $L \in \mathbb{R}^{s\times n}$, $s \le n$ 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. Stabilization of total least squares problems by introducing a quadratic constraint was extensively studied in [4, 12, 15, 21, 22, 23, 26, 29, 31, 32].

The topic of this paper is the determination of a suitable value of the hyperparameter δ if no previous knowledge about its size is available.

For the determination of the regularization paramter in least squares problems there exist several well-known methods. For example the discrepancy principle [28], generalized cross validation [13], information criteria [1, 30] or the L-curve [17, 18, 20]. In principle it is possible to extend all of these methods to the RTLS case. Here we particularly investigate the L-curve method for problem (1.3).

The paper is organized as follows. In Section 2, two algorithms for solving the RTLS problem when the regularization parameter is known are briefly summarized. An efficient implementation of the algorithms is described in Section 3, togehter with a straightforward approach to build these methods into an L-curve approach for determining the hyperparameter. The focus is set on the reuse of information during the process to set up the L-curve at discrete points. Section 4 contains numerical examples, demonstrating the efficiency of the presented approach.

2 Solving the RTLS problem

Here we shortly summarize two methods for solving the RTLS problem (1.3) for a fixed value δ.

In the entire paper we assume that the following condition holds

$$\sigma_{min}([AK, b]) < \sigma_{min}(AK), \tag{2.1}$$

where K is an orthonormal basis of the kernel of L, which guarantees that a solution of the RTLS problem (1.3) is attained, cf. [3]. Notice that the condition is empty if the regularization matrix L is regular.

It is usually assumed that the regularization parameter $\delta > 0$ is less than $\|Lx_{TLS}\|$, where x_{TLS} denotes the solution of the total least squares problem (1.2) (otherwise no regularization would be necessary). Then at the optimal solution of (1.3) the constraint $\|Lx\| \leq \delta$ holds with equality. Under this condition Golub, Hansen and O'Leary [12] derived the following first order necessary conditions: The solution x_{RTLS} of problem (1.3) is a solution of the problem

$$(A^T A + \lambda_I I_n + \lambda_L L^T L)x = A^T b, \tag{2.2}$$

where the parameters λ_I and λ_L are given by

$$\lambda_I = -\frac{\|Ax - b\|^2}{1 + \|x\|^2}, \quad \lambda_L = \frac{1}{\delta^2}\big(b^T(b - Ax) - \frac{\|Ax - b\|^2}{1 + \|x\|^2}\big). \tag{2.3}$$

The parameters λ_I and λ_L both depend on x and make the system of equations (2.2) hardly tractable. The conditions (2.2) and (2.3) have been used in literature in two ways to solve problem (1.3): In [12, 15, 22, 29] λ_I is chosen as a free parameter; for fixed λ_L problem (2.2) is solved for (x, λ_I), and then λ_L is updated in a way that the whole process converges to the solution of (1.3). Conversely, in [21, 23, 31, 32] for a chosen parameter λ_I problem (2.2) is solved for (x, λ_L), which yields a convergent sequence of updates for λ_I.

In both cases problem (2.2) can be solved via the solution of an eigenvalue problem. In the first case one has to determine in every iteration step the eigenvector of a symmetric matrix corresponding to its smallest eigenvalue, and in the latter approach one has to find the rightmost eigenvalue and corresponding eigenvector of a quadratic eigenproblem. Hence, in both cases one has to solve a sequence of eigenvalue problems which converges as the methods approach the

solution of (1.3). This suggests, that when solving one of these eigenvalue problems one should reuse as much information as possible from previous iteration steps.

Typically, the occurring eigenproblems are solved by inverse iteration, Rayleigh quotient iteration, implicitly restarted Lanczos or second order Krylov subspace solvers. In these methods the only information that can be recycled from previous iterations is the eigenvector of the preceding step that can be used as initial vector.

Much more information can be exploited in general iterative projection methods such as the Nonlinear Arnoldi algorithm [34] which can be started with the entire search space of the previous eigenvalue problems. Details are described in Subsection 3.1.

2.1 Sequence of quadratic eigenvalue problems

The first algorithm is based on keeping the parameter λ_I fixed for one iteration step and let $\lambda := \lambda_L$ be a free parameter. The fixed parameter is updated and initialized as suggested in (2.3)

$$\lambda_I = \lambda_I(x^k) = -\frac{\|Ax^k - b\|^2}{1 + \|x^k\|^2}. \tag{2.4}$$

The first order optimality conditions then read

$$B(x^k)x + \lambda L^T L x = A^T b, \quad \|Lx\|^2 = \delta^2, \tag{2.5}$$

with

$$B(x^k) = A^T A - f(x^k)I, \ f(x^k) = \frac{\|Ax^k - b\|^2}{1 + \|x^k\|^2} = -\lambda_I(x^k). \tag{2.6}$$

which suggests the following Algorithm 2.1.

Algorithm 2.1 RTLSQEP

1: Require Initial vector x^1.
2: **for** $k = 1, 2, \ldots$ until convergence **do**
3: With $B_k := B(x^k)$ solve

$$B_k x^{k+1} + \lambda L^T L x^{k+1} = A^T b, \quad \|Lx^{k+1}\|^2 = \delta^2 \tag{2.7}$$

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

Sima, Van Huffel and Golub [32] proposed to solve (2.5) via a quadratic eigenvalue problem similarly to the approach of Golub [11] for regularized least squares problems. This motivates the name RTLSQEP of the algorithm.

If L is square and nonsingular, then with $z = Lx^{k+1}$ problem (2.7) is equivalent to

$$W_k z + \lambda z := L^{-T} B_k L^{-1} z + \lambda z = L^{-T} A^T b =: h, \quad z^T z = \delta^2. \tag{2.8}$$

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

$$(W_k + \lambda I)^2 u - \delta^{-2} h h^T u = 0. \tag{2.9}$$

The choice of the rightmost eigenvalue can be motivated as the maximal Lagrange multiplier that minimizes an underlying quadratic function, cf. [9], [23].

In [21] it is proven that the rightmost eigenvalue $\hat{\lambda}$ of (2.9) is real and that $W_k + \hat{\lambda} I$ is positive semidefinite. We are only considering the generic case of $W_k + \hat{\lambda} I$ being positive definite. In this case the solution of the original problem (2.7) is recovered from $z = (W_k + \hat{\lambda} I) u$, and $x^{k+1} = L^{-1} z$ where u is an eigenvector corresponding to $\hat{\lambda}$ which is scaled such that $h^T u = \delta^2$. The case that $W_k + \hat{\lambda} I \geq 0$ is singular means the solution of (2.7) may not be unique, which is discussed by Gander, Golub and von Matt in [10].

When the constraint at the solution of (1.3) is active, i.e. if $\|Lx^*\|^2 = \delta^2$, then the following global convergence result holds.

Theorem 2.1 *Any limit point x^* of the sequence $\{x^k\}$ constructed by Algorithm 2.1 is a global minimizer of the optimization problem*

$$f(x) := \frac{\|Ax - b\|^2}{1 + \|x\|^2} = \min! \quad \textit{subject to } \|Lx\|^2 = \delta^2. \tag{2.10}$$

Proof: cf. [23].

■

The transformation of (2.5) to the quadratic eigenproblem (2.9) seems to be very costly because of the inverse of L. Typical regularization matrices are discrete versions of 1D first (or second) order derivatives

$$L = \begin{pmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \end{pmatrix} \in \mathbb{R}^{(n-1)\times n}. \tag{2.11}$$

Smoothing properties of L are not deteriorated significantly if they are replaced by regular versions like (cf. [5])

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

with some small diagonal element $\varepsilon > 0$. Which one of these modifications is chosen depends on the behaviour of the solution of (1.3) close to the boundary. It is not necessary to calculate the inverse of L explicitely since the presented algorithms will only touch L^{-1} by matrix-vector multiplications. Solving a system with $\hat{L}$ is cheap, i.e. an $\mathcal{O}(n)$-operation.

Remark 2.2 *If it holds that rank$(L) < n$ and one is not willing to perturb L, a basis of the range and kernel are needed. In [32] it is explained how to obtain a similar expression for W_k in* (2.8) *and the quadratic eigenvalue problem* (2.9)*. For the L in* (2.11) *the spectral decomposition is explicitly known. For solutions on a 2D or 3D domain one can take advantage of Kronecker product representations of L and its decomposition, cf. [24].*

2.2 Sequence of linear eigenvalue problems

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

The following version of the first order optimality conditions was proved by Renaut and Guo in [29].

Theorem 2.3 *The solution x_{RTLS} of the RTLS problem* (1.3) *subject to the active constraint satisfies the augmented eigenvalue problem*

$$B(\lambda_L(x_{RTLS}))\begin{pmatrix} x_{RTLS} \\ -1 \end{pmatrix} = -\lambda_I(x_{RTLS})\begin{pmatrix} x_{RTLS} \\ -1 \end{pmatrix}, \tag{2.12}$$

with

$$B(\lambda_L) = M + \lambda_L N, \quad M := [A, b]^T[A, b], \quad N := \begin{pmatrix} L^T L & 0 \\ 0 & -\delta^2 \end{pmatrix}$$

and λ_L and λ_I as given in (2.3)*.*

Conversely, if $((\hat{x}^T, -1)^T, -\hat{\lambda})$ is an eigenpair of $B(\lambda_L(\hat{x}))$ where $\lambda_L(\hat{x})$ is recovered according to (2.3)*, then $\hat{x}$ satisfies* (2.2)*, and $\hat{\lambda} = -f(\hat{x})$.*

This condition suggested Algorithm 2.2 called RTLSEVP for obvious reasons.

Algorithm 2.2 RTLSEVP

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

$$B_{k-1} y^k = \lambda y^k \tag{2.13}$$

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

The choice of the smallest eigenvalue is motivated by the fact that we are aiming at $\lambda = -\lambda_I$ (cf. (2.12)), and by the first order conditions (2.3) it holds that $-\lambda_I = f(x) = \frac{\|Ax-b\|^2}{1+\|x\|^2}$ is the function to be minimized.

The straightforward idea in [15] to update λ_L in line 5 with (2.3), i.e.

$$\lambda_L^{k+1} = \frac{1}{\delta^2}\Big(b^T(b - Ax^{k+1}) - \frac{\|Ax^{k+1} - b\|^2}{1 + \|x^{k+1}\|^2}\Big) \tag{2.14}$$

does not lead in general to a convergent algorithm.

To enforce convergence Renaut and Guo [29] proposed to determine a value θ such that the eigenvector $(x_\theta^T, -1)^T$ of $B(\theta)$ corresponding to the smallest eigenvalue of $B(\theta)$ satisfies the constraint $\|Lx_\theta\|^2 = \delta^2$.

The following function has been introduced in [22]:

Definition 2.4 *Let $\mathcal{E}(\theta)$ denote the eigenspace of $B(\theta)$ corresponding to its smallest eigenvalue. Then*

$$g(\theta) := \min_{y \in \mathcal{E}(\theta)} \frac{y^T N y}{y^T y} = \min_{(x^T, x_{n+1})^T \in \mathcal{E}(\theta)} \frac{\|Lx\|^2 - \delta^2 x_{n+1}^2}{\|x\|^2 + x_{n+1}^2} \tag{2.15}$$

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

The following properties of this function have been proven in [22].

Theorem 2.5 *The function $g : [0, \infty) \to \mathbb{R}$ has the following properties:*

(i) *If $\sigma_{\min}([A, b]) < \sigma_{\min}(A)$ then $g(0) > 0$*

(ii) $\lim_{\theta \to \infty} g(\theta) = -\delta^2$

(iii) *If the smallest eigenvalue of $B(\theta_0)$ is simple, then g is continuous at θ_0*

(iv) *g is monotonically not increasing on $[0, \infty)$*

(v) *Let $g(\hat{\theta}) = 0$ and let $y \in \mathcal{E}(\hat{\theta})$ such that $g(\hat{\theta}) = y^T N y / \|y\|^2$. Then the last component of y is different from 0.*

(vi) *g has at most one root.*

Theorem 2.5 demonstrates that if $\hat{\theta}$ is a positive root of g, then $x^* := -y(1 : n)/y_{n+1}$ solves the RTLS problem (1.3) where y denotes an eigenvector of $B(\hat{\theta})$ corresponding to its smallest eigenvalue.

Remark 2.6 *Notice that g is not necessarily continuous. If the multiplicity of the smallest eigenvalue of $B(\theta)$ is greater than 1 for some θ_0, then g may have a jump discontinuity at θ_0. It may also happen that g does not have a root, but it jumps below zero at some θ_0. This indicates a nonunique solution of problem (1.3). See [22] how to construct a solution in this case. Here we assume a unique solution, which is the generic case.*

3 Computation of the L-curve

The goal is to obtain a suitable L-curve for the quadratically constraint regularized least squares problem (1.3). Let us briefly review some facts about the L-curve in general.

The original idea of the L-curve is to balance the residual $\|Ax_\lambda - b\|$ and the size of the solution $\|x_\lambda\|$ in a least squares problem with Tikhonov regularization [17, 20]. Sometimes it is more suitable to choose the regularization matrix $L \neq I$, so that $\|x_\lambda\|$ is replaced by $\|Lx_\lambda\|$.

In [17] it was shown that if the noise vector is not too big, has zero mean and covariance matrix $\sigma_0 I$ and if the discrete Picard condition holds, cf. [16], the L-curve exhibits a corner. In the corner a suitable regularization parameter is located since it corresponds to a reasonable balancing between the residual and the size. In many examples a corner of the L-curve is visible even without fullfilling the Picard condition.

The solution of least squares problems with Tikhonov regularization is equivalent to a quadratically constraint if it holds $\delta < \|Lx_{LS}\|$ with x_{LS} as the least squares solution without regularization, cf. [12]. The difference is the sensitivity of the residual with respect to λ and δ, but both approaches deliver points $(\|Ax_i - b\|, \|Lx_i\|)$ from the same L-curve. Essentially only a reparameterization from $\lambda \in [0, \infty)$ to $\delta \in [0, \|Lx_{LS}\|]$ with some nonlinear monotonic decreasing relation of λ and δ is performed.

In regularized least squares the GSVD is a powerful tool: With this decomposition an analytical expression for the L-curve can be derived (and determining its corner is trivial). Hence efficient methods are based on an approximation of this decomposition which leads to a reasonable approximation of the L-curve.

In the RTLS case there is a big difference: No closed-form solution exists, neither with respect to a Tikhonov parameter λ nor to a quadratic constraint δ. This means that the L-curve can only be approximated by a finite number of points, i.e. for every chosen value of λ or δ one RTLS problem has to be solved.

Measuring the residual of a RTLS problem in terms of $\|Ax_\delta - b\|$ is not suitable. In [31, 33] it was shown that the generalized error $\dfrac{\|Ax_\delta - b\|^2}{1 + \|x_\delta\|^2}$ is a consistent choice. Note, that this is the quantity $f(x)$ appearing in theorem 2.1 that is going to be minimized. Hence the curve of interest is defined by

$$\left(\frac{\|Ax_\delta - b\|^2}{1 + \|x_\delta\|^2}, \|Lx_\delta\| \right) \quad \text{for values} \quad \delta \in [0, \|Lx_{TLS}\|] \tag{3.1}$$

corresponding to problem (1.3).

There is no guarantee for the occurence of an L-shape of this curve. Since the Picard condition cannot be extended straightforward to the RTLS case there is the lack of an analytical tool. So the curve defined by (3.1) is used to visualize the relation between the generalized error and the size of the solution. When setting up this curve, it turned out that many examples do exhibit L-curve behaviour, which allows a reasonable balancing by picking the regularization parameter that corresponds to the corner.

In the two Subsections 3.1 and 3.2 we discuss in more detail efficient implementations of the Algorithms 1 and 2. The focus is set on dealing with the sequence of quadratic eigenvalue problems and linear eigenvalue problems, respectively.

In Subsection 3.3 a straightforward approach to set up the L-curve is suggested. For each different value of δ another RTLS problem has to be solved. But since two subsequent RTLS problems are so closely related, it is very favourable to use as much old information as possible. It turns out that the well-suited solver for a single RTLS problem introduced in 3.1 and 3.2, i.e. the Nonlinear Arnoldi method [34], fits even better to the L-curve setting.

3.1 Computation of RTLSQEP

The main computational cost of the RTLSQEP algorithm is solving a sequence of equations (2.9) via the QEPs (2.11). The rightmost eigenpair of

$$T_k(\lambda)u := \left((W_k + \lambda I)^2 - \delta^{-2} h h^T\right) u = 0 \tag{3.2}$$

at the k-th iteration step of Algorithm 2.1 could be efficiently obtained by the Krylov subspace-type method from Li and Ye in [25] or with the Second Order Arnoldi Reduction (SOAR) presented in [2]. These two methods are excellent choices when solving a single QEP. They are both started with an initial vector, and hence we start in step k with the solution u^{k-1} of the preceding step.

But since the sequence $\{f(x_k)\}$ converges, the matrices

$$W_k = L^{-T}(A^T A + f(x_k)I)L^{-1} = L^{-T}A^T A L^{-1} + f(x_k)L^{-T}L^{-1} \tag{3.3}$$

converge as well. In [21, 23, 24] it was shown that when solving a convergent sequence of QEPs, it is highly favourable to use a method that reuses more information than a single vector from the last iteration step.

This is possible with the Nonlinear Arnoldi method [34] which can be applied to much more general nonlinear eigenvalue problems $T(\lambda)u = 0$ than (3.2).

Algorithm 3.1 Nonlinear Arnoldi

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

The Nonlinear Arnoldi method allows thick starts, i.e. when solving $T_k(\lambda)u = 0$ in step k Algorithm 3.1 can be started with the orthonormal basis V that was used in the preceding step when determining the solution $u^{k-1} = V\tilde{u}$ of $V^T T_{k-1}(\lambda)V\tilde{u} = 0$, hence all search spaces are kept.

Some comments on an efficient implementation of RTLSQEP with the Nonlinear Arnoldi solver:

- The representation of W_k in (3.3) demonstrates that the projected eigenvalue problem

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

  can be determined efficiently if the matrices $L^{-T}A^T AL^{-1}V, L^{-T}L^{-1}V, V$ and $h^T V$ are known. These can be updated cheaply by appending one column and component to the current matrices and vector, respectively. Note that W_k is never needed explicitely, but only touched by matrix-vector multiplications.

- Fixed storage requirements can be met by a suitable restart strategy, cf. Subsection 3.3.

- A suitable initial basis V for the first quadratic eigenvalue problem (2.9) can be obtained by a small number, e.g. $\ell = 5 - 10$, of Lanczos steps applied to the linear eigenproblem $W_1 z = \lambda z$ with a random vector $r_0 \in \mathbb{R}^n$.

- It turned out that while reducing the residual of the approximated rightmost eigenpair of a QEP in step k by a factor 100 (instead of solving it to full accuracy), sufficient new information is added to the search space V. So the stopping criterion in line 5 of Algorithm 3.1 can be replaced by $\|r\|/\|r_k\| > 0.01$ with the current initial residual r_k calculated in line 4. This approach leads to more outer iterations but overall to less inner iterations, cf. [24].

- Since the dimension of the projected problems are small they can be solved by linearization and a dense eigensolver like MATLABs `eig`.

- In general no preconditioning is needed, so we simply set $PC = I$.

3.2 Computation of RTLSEVP

The main computational cost of the RTLSEVP algorithm is solving a sequence of linear eigenproblems (2.13). The smallest eigenpair of

$$T_k(\lambda)u := (B_k - \lambda I)u = (M + \theta_k N - \lambda I)u = 0 \tag{3.4}$$

at the k-th iteration step of Algorithm 2.2 could be obtained via Rayleigh quotient or Inverse Iteration, cf. [29]. These iterative methods have to be started

with an initial vector, hence we initialize by the eigenvector found in the preceding iteration step. This reuses information in form of a single vector. A drawback is the expensive LU-decomposition of $B(\theta_k)$. But since the sequence $\{\theta_k\}$ converges, the matrices

$$B_k = [A,b]^T[A,b] + \theta_k \begin{pmatrix} L^TL & 0 \\ 0 & -\delta^2 \end{pmatrix} \tag{3.5}$$

converge as well and again it is highly favourable to use a method that reuses much more information. Similar to the approach in RTLSQEP the entire information gathered in previous iteration steps can be employed solving (3.4) via the Nonlinear Arnoldi method 3.1, cf. [22, 24]. For RTLSEVP the lines 2 and 9 of Algorithm 3.1 have to be changed into finding the minimum eigenvalue of $V^TT_k(\mu)V$ (instead of the rightmost).

Some further comments on an efficient implementation:

- The projected problems

$$V^TT_k(\mu)V\tilde{u} = \big(([A,b]V)^T([A,b]V) + \theta_k V^TNV - \mu I\big)\tilde{u} = 0 \tag{3.6}$$

  can be updated easily when the matrices $[A,b]V$, $LV(1:n,:)$ and V are stored. They are refreshed on the fly appending a new column every step. The explicit form of the matrices M and N is not needed.

- It is possible to meet fixed storage requirements, by using a restart strategy, cf. Subsection 3.3.

- A suitable initial basis V for the first eigenvalue problem (2.13) can be obtained by an orthonormal basis of the Krylov subspace $\mathcal{K}_\ell(B_0, e_{n+1})$ with a small value of $\ell = 5-10$.

- It is sufficient to choose a rough preconditioner $PC \approx N^{-1}$ in line 3, which according to Remark 2.2 usually can be implemented very cheaply and can be kept constant throughout the whole algorithm.

- For the determination of the value λ_L^k in Line 5 of Algorithm 2.2 a suitable root-finding algorithm has to be applied. In [22] an enclosing algorithm for the root of the function $g(\theta)$ is presented. It is based on rational inverse interpolation and makes use of the known pole of $g^{-1}(\theta)$ at $\theta = -\delta^2$.

The computation times of RTLSQEP and RTLSEVP with the Nonlinear Arnoldi method as eigensolver are comparable. Both approaches do converge in much less matrix-vector multiplications than the dimension of the problem, so the computational complexity is of order $\mathcal{O}(mn)$.

3.3 Sequence of RTLSQEP or RTLSEVP

As described in the beginning of this Section it is only possible to approximate the L-curve (3.1) for the RTLS problem at a given set of discrete values for δ.

For each value δ_i of the set $\{\delta_1, \ldots, \delta_r\}$ another RTLS problem has to be solved. Hence we need to solve a sequence of a sequence of quadratic (cf. Subsection 3.1) or linear eigenproblems (cf. Subsection 3.2).

Let us assume the set be assorted such that $0 \leq \delta_1 < \cdots < \delta_r \leq \|Lx_{TLS}\|$ holds. If the set is not too coarse, two subsequent RTLS problems are so closely related that it will pays to reuse as much information as possible. Depending on the values of $\delta_i, i = 1, \ldots, r$ this does not only hold for two but for many more subsequent problems. This makes it even more attractive to use the Nonlinear Arnoldi method in the L-curve setting. It turned out that keeping the search space during solving the sequence of RTLS problem, instead of keeping it only for one RTLS problem, is a highly efficient choice.

The main computational cost is solving the first problem of the sequence, where a starting search space has to be built up. The following RTLS problems are often solved within a few matrix-vector multiplications due to the known search space of related problems that have already been solved.

One important algorithmic aspect is the use of restarts. When keeping all information during the sequence of RTLS problems the search space $\mathrm{span}(V)$ grows too large and the costs for solving the projected problems $V^T T_k(\lambda) V \tilde{u} = 0$ cannot be neglected any longer. To avoid this the following restart strategy is applied: A maximal dimension $p \ll n$ of the search space is set in advance, and if in the course of Algorithms 2.1 or 2.2 the dimension of V reaches p, then the Nonlinear Arnoldi method 3.1 is restarted with a subspace spanned by a small number $q < p$ of eigenvectors corresponding to the rightmost or smallest eigenvalues of $V^T T_k(\lambda) V$.

Remark 3.1 *When a restart is necessary this normally purges out information of the search space that corresponds to old δ_i, i.e. values of δ that are oders of magnitudes smaller and hence highlight different aspects of the problem. So in general a restart is not harmful at all.*

For large scale problems and a reasonable number of values $\delta_i, i = 1, \ldots, r$ (i.e. $r \approx 20 - 50$) the computation of the points of the L-curve stays an $\mathcal{O}(mn)$ operation, since only matrix-vector multiplications are performed.

4 Numerical examples

To evaluate the performance of Algorithms 2.1 and 2.2 with the Nonlinear Arnoldi method 3.1 as inner eigensolver we use large dimensional test examples from Hansen's *Regularization Tools*, [19]. 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, phillips` and `heat` (κ=1) provided the matrices A_{true}, the right hand sides b_{true} and the true solutions x_{true}, with $A_{true} x_{true} = b_{true}$. In all cases the matrices A_{true} and $[A_{true}, b_{true}]$ are ill-conditioned. The parameter $\kappa = 1$ for problem `heat` indicates a severely ill-conditioned problem.

To construct a suitable RTLS problem, the norm of b_{true} is scaled such that $\|b_{true}\| = \max_i \|A_{true}(:,i)\|$ holds, x_{true} is then scaled by the same factor. The noise added to the problem is put in relation to the average value of the elements of the augmented matrix, i.e. $aver = \sum(\sum(abs[A_{true}, b_{true}]))/(m(n+1))$. We add white noise of level $1-10\%$ to the data and obtained the system $Ax \approx b$ where $A = A_{true} + \sigma E$ and $b = b_{true} + \sigma e$, with $\sigma = aver \cdot (0.01, 0.1)$ and the elements of E and e are independent random variables with zero mean and unit variance.

To create a suitable L-curve scenario, $r = 30$ RTLS problems with values $\delta_i, i = 1, \ldots, 30$ over a range of six orders of magnitude are calculated. In the examples `baart, shaw, deriv2` and `phillips` 30 logarithmically equally spaced values from $\delta^* \cdot [1e{-}4, 1e2]$ are chosen, and for `heat` the 30 values are from $\delta^* \cdot [1e{-}4, 5e0]$, with $\delta^* = \delta_{true} = \|Lx_{true}\|$.

The numerical test were run on a PentiumR4 computer with 3.4 GHz and 8GB RAM under MATLAB R2008a.

In Algorithm 2.1 the outer iteration for one RTLS problem was terminated if the $f(x_k)$ has converged, i.e. two subsequent values do not differ relatively by more than 0.1%. The starting search space for all examples was an orthonormal basis of the Krylov space $\mathcal{K}_{10}(L^{-T}A^TAL^{-1}, (1, \ldots, 1)^T)$. When the dimension of the search space V exeeds the value $p = 60$, the Nonlinear Arnoldi method is restarted with $q = 10$ eigenvectors corresponding to the rightmost eigenvalues. For the RTLSQEP algorithm two regularization matrices have been tested: The 1D discrete first order derivative operator L from equation (2.11) and a slightly disturbed variant $\hat{L}$ with $\varepsilon = 0.1$ which is denoted by 'b' in comparison to the unperturbed L denoted by 'a' in the Tables 1 and 2. If not the regular matrix $\hat{L}$ is used, one needs to solve systems with the range of L, cf. Remark 2.2. This can be done efficiently in less than $\mathcal{O}(n^2)$ by using the discrete cosine transform.

Algorithm 2.2 was terminated if the $g(\theta_k)$ is sufficiently close to zero, i.e. less than $10^{-3}\delta_i^2/(1 + \|x_k\|^2)$. It was used the enclosing algorithm from [22], where in the starting phase one has to find three pairs $\theta_i, g(\theta_i)$ with the $g(\theta_i)$ not all having the same sign. One step of the enclosing algorithm is counted as one (outer) iteration, note that performing such an outer iteration costs much less than a matrix-vector multiplication. The starting search space for all examples with RTLSEVP was an orthonormal basis of the Krylov space $\mathcal{K}_{10}(M, e_{n+1})$ with e_{n+1} as last unit vector. When the dimension of the search space V exeeds the value $p = 80$ the Nonlinear Arnoldi method is restarted with $q = 10$ eigenvectors corresponding to the smallest eigevalues. Here a preconditioner is useful and was calculated with UMFPACK [6], i.e. MATLABs $[L, U, P, Q] = lu(\tilde{N})$, with a slightly perturbed $\tilde{N} \approx N$ to make it nonsingular.

Tables 1 and 2 contain the results of regularization problems averaged over 10 random simulations, differing only in the additional noise that was put onto A_{true} and b_{true} with noise levels 1% and 10% respectively. The dimension of the matrix $A \in \mathbb{R}^{m \times n}$ is either $m = 2n = 2000$ or $m = 2n = 4000$. Originally the examples are quadratic, here simply two system matrices are put on top of each other.

In Table 1 the following shortcuts are used: 'QEP' denotes the RTLSQEP

Algorithm 2.1 and 'EVP' denotes the RTLSEVP Algorithm 2.2. The use of the regularization matrices L and $\hat{L}$ are indicated by an additional 'a' and 'b' respectively.

'MVs' is the number of matrix-vector multiplications (MatVecs) for setting up the whole L-curve consisting of $r = 30$ points. 'Iter' is the number of iterations. For RTLSQEP one step of the Nonlinear Arnoldi, i.e. one inner iteration, was counted as one iteration, whereas for RTLSEVP it means one evaluation of $g(\theta)$, which is an outer iteration. RTLSQEP has much less outer iterations, but one outer iteration needs much more MatVecs than in RTLSEVP, hence it was more interesting to compare these values. 'Restarts' is the average number of restarts that are performed. $\frac{|\|Lx_i\|-\delta_i|}{\delta_i}$ denotes the average relative constraint violation, i.e. with $r = 30$ problems for one L-curve $\frac{1}{r}\sum_{i=1}^{r} \frac{|\|Lx_i\| - \delta_i|}{\delta_i}$, where the x_i are the computed solutions of the corresponding RTLS problem with constraint equal to δ_i. In the last two columns the relative error of the corner of the L-curve and the point $(f^*, \delta^*) = (\frac{\|Ax_{true}-b\|^2}{1+\|x_{true}\|^2}, \|Lx_{true}\|)$ is given. The values δ_C and f_C are determined by a corner finding routine from P.C. Hansen, cf. [19]. This routine chooses the pair $(f(x_i), \delta_i)$ that most likely corresponds to the corner of an underlying L-curve. Note, that by picking the δ_C from the set of given values for δ_i it was impossible by construction to hit the value δ^*. An asterix behind the relative error of δ_C indicates that the best possible δ_i was picked in all 10 random examples.

For most examples one needs about 400-600 MatVecs with the matrix A for setting up one L-curve, independent of the problem size. The three different algorithms 'QEPa', 'QEPb' and 'EVPa' have a comparable number of MatVecs, but in most examples the 'QEPa' needs a little less. These MatVecs of the algorithms are the main costs, determining the corner after having calculated the points on the L-curve is neglectable. For RTLSQEP one inner iteration roughly costs 4 MatVecs, cf. [24], which can be seen in the tables. The cost of an inner iteration of RTLSEVP is about the half, i.e. 2 MatVecs. The number of inner iterations of 'QEP' and outer iterations of 'EVP' are similar, hence RTLSEVP roughly performes two inner iterations within one outer iteration, which leads two many more inner iterations when compared to RTLSQEP.

The number of restarts for the 'QEP' algorithms is between 1-2, compared to 3-4 restarts for 'EVPa'. When using the exact regularization matrix L in 'QEPa' and 'EVPa' the relative constraint violation is smaller than compared to 'QEPb'. In general these small relative errors indicates a reliable determination of the complete L-curve. The last two columns indicate that the L-curve approach was suitable for the examples. The determined corner is a good approximation to the corresponding unperturbed data. When using the exact regularization matrix L one more likely hits the best possible δ_i.

Table 1: Problems from Regularization Tools - 1% noise

Problem $size$(A)	Solver	MVs(Iter)	Restarts	$\frac{\lvert\lVert Lx_i\rVert-\delta_i\rvert}{\delta_i}$	$\frac{\lvert\delta_C-\delta^*\rvert}{\delta^*}$	$\frac{\lvert f_C-f^*\rvert}{f^*}$
baart	QEPa	400 (82)	1.0	1e-5	0.35	4e-3
2000×1000	QEPb	545 (106)	1.7	9e-2	0.15*	2e-3
	EVPa	525 (117)	3.0	8e-5	0.35	4e-3
baart	QEPa	382 (77)	1.0	5e-8	0.37	2e-3
4000×2000	QEPb	494 (96)	1.4	9e-2	0.17	1e-3
	EVPa	517 (116)	3.0	8e-5	0.37	2e-3
shaw	QEPa	396 (78)	1.1	3e-5	0.15*	7e-2
2000×1000	QEPb	608 (120)	1.6	2e-2	0.15*	7e-2
	EVPa	542 (118)	3.0	8e-5	0.15*	7e-2
shaw	QEPa	357 (70)	1.0	1e-5	0.15*	6e-2
4000×2000	QEPb	569 (112)	1.9	4e-2	0.15*	7e-2
	EVPa	537 (118)	3.0	9e-5	0.15*	6e-2
deriv2	QEPa	580 (126)	1.8	1e-2	0.15*	1e-1
2000×1000	QEPb	571 (112)	1.6	1e-2	0.37	9e-3
	EVPa	544 (116)	3.0	3e-3	0.15*	1e-1
deriv2	QEPa	547 (115)	1.7	2e-2	0.15*	1e-1
4000×2000	QEPb	614 (121)	1.6	1e-1	0.38	5e-3
	EVPa	640 (158)	3.8	6e-3	0.15*	1e-1
phillips	QEPa	476 (94)	1.1	3e-4	0.37	1e-2
2000×1000	QEPb	656 (130)	1.8	3e-3	0.88	1e-2
	EVPa	607 (137)	3.3	2e-4	0.37	1e-2
phillips	QEPa	488 (96)	1.1	2e-1	0.38	7e-3
4000×2000	QEPb	696 (136)	1.9	2e-1	0.79	7e-3
	EVPa	623 (143)	3.4	1e-3	1.04	8e-3
heat (κ=1)	QEPa	501 (104)	1.0	2e-2	0.12*	2e-2
2000×1000	QEPb	701 (146)	1.9	7e-2	0.28	2e-3
	EVPa	599 (149)	3.6	4e-3	0.12*	2e-2
heat (κ=1)	QEPa	550 (114)	1.5	2e-2	0.12*	6e-3
4000×2000	QEPb	660 (138)	1.7	5e-2	0.28	4e-2
	EVPa	528 (116)	3.0	3e-3	0.12*	1e-2

Table 2: Problems from Regularization Tools - 10% noise

Problem $size$(A)	Solver	MVs(Iter)	Restarts	$\frac{\lvert\lVert Lx_i\rVert-\delta_i\rvert}{\delta_i}$	$\frac{\lvert\delta_C-\delta^*\rvert}{\delta^*}$	$\frac{\lvert f_C-f^*\rvert}{f^*}$
baart	QEPa	514 (114)	1.1	6e-8	0.80	5e-2
2000×1000	QEPb	356 (68)	1.0	3e-4	0.15*	2e-3
	EVPa	532 (119)	3.0	3e-4	0.80	5e-2
baart	QEPa	468 (102)	1.0	2e-9	0.80	6e-2
4000×2000	QEPb	369 (72)	1.1	2e-3	0.15*	7e-4
	EVPa	527 (115)	3.0	8e-5	0.80	6e-2
shaw	QEPa	440 (94)	1.1	2e-6	0.67	3e-2
2000×1000	QEPb	394 (77)	1.1	4e-4	0.47	1e-2
	EVPa	546 (119)	3.0	4e-4	0.67	3e-2
shaw	QEPa	367 (74)	1.0	4e-7	0.67	2e-2
4000×2000	QEPb	392 (77)	1.1	2e-3	0.47	9e-3
	EVPa	536 (115)	3.0	7e-5	0.67	2e-2
deriv2	QEPa	441 (98)	1.0	1e-6	0.15*	9e-4
2000×1000	QEPb	379 (76)	1.0	6e-5	0.15*	1e-2
	EVPa	545 (121)	3.1	8e-5	0.15*	9e-4
deriv2	QEPa	432 (93)	1.0	5e-6	0.15*	7e-4
4000×2000	QEPb	383 (75)	1.0	2e-4	0.15*	3e-2
	EVPa	542 (117)	3.0	9e-5	0.15*	7e-4
phillips	QEPa	431 (92)	1.0	3e-5	0.21	4e-1
2000×1000	QEPb	417 (86)	1.0	6e-4	0.21	4e-1
	EVPa	558 (119)	3.0	2e-4	0.21	4e-1
phillips	QEPa	401 (84)	1.0	3e-5	0.24	3e-1
4000×2000	QEPb	421 (85)	1.0	1e-3	0.24	3e-1
	EVPa	583 (129)	3.1	2e-4	0.24	3e-1
heat (κ=1)	QEPa	455 (98)	1.0	1e-2	0.23	2e-2
2000×1000	QEPb	461 (99)	1.1	2e-3	0.23	2e-2
	EVPa	581 (133)	3.3	2e-2	0.23	2e-2
heat (κ=1)	QEPa	452 (96)	1.0	6e-6	0.23	2e-2
4000×2000	QEPb	517 (113)	1.4	1e-2	0.23	2e-2
	EVPa	594 (136)	3.5	7e-3	0.23	2e-2

In Table 2 the examples are calculated again, now with the noise level increased to 10%. For most examples the three algorithms still require about 400-600 MatVecs with the matrix A for setting up the L-curve, independent of the problem size. The other values are also comparable to those from Table 1. This indicates that the approach is not very sensitive to the noise level.

Figure 1 shows the convergence history of the Nonlinear Arnoldi method for setting up the L-curve for the example *deriv2* with dimension 2000×1000 and 1% noise. Algorithm 2.1 with an exact regularization matrix L has been used. The dashed red lines indicate the solution of a RTLS problem after the corresponding number of inner iterations. The blue stars are the relative residuals of the

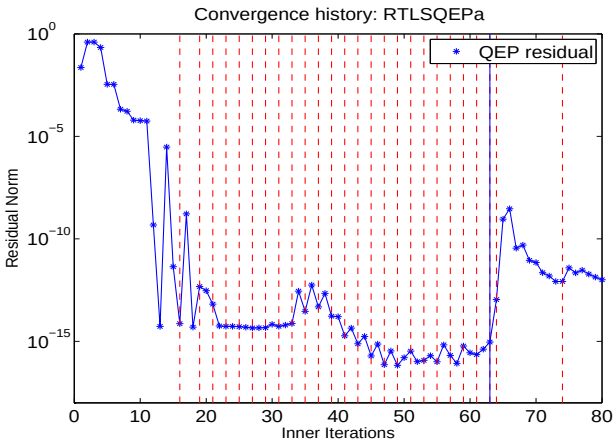

Figure 1: Convergence history of L-curve for *deriv2*

quadratic eigenvalueproblems, where 'relative' means with respect to the norm of the current rank-one matrix $\delta_i^{-2} hh^T$. The first 10 inner iterations are needed to build up the starting search space and after another 6 iterations 2 QEPs are solved, which are sufficient to solve the first RTLS problem with δ_1. The search space at this moment contains such good information that subsequent problems are solved within 2-3 inner iterations. After the restart which is indicated by the straight blue line, the next problems needed some more iterations until the search space contains again enough good information.

The RTLSQEP algorithm with the unperturbed regularization matrix turned out to be slightly superior to the RTLSEVP algorithm, at least in the chosen examples. All approaches do converge in much less MatVecs than the dimension of the problem, so the computational complexity is of order $\mathcal{O}(mn)$.

5 Conclusions

In many cases the L-curve is a suitable tool to determine the hyperparameter δ for the RTLS problem with quadratic constraint. Two efficient algorithms for evaluating the L-curve at discrete points have been presented.

For solving a single RTLS problem the RTLSQEP Algorithm 2.1 and the RTLSEVP Algorithm 2.2 are well-suited choices when combined with the Nonlinear Arnoldi Algorithm 3.1. Setting up the L-curve requires solving a sequence of RTLS problems, where the advantage of the Nonlinear Arnoldi method is even more apparent: The high reuse of information, not only within one RTLS problem but throughout the whole sequence.

For large scale problems with huge dimension $m \times n$ and a reasonable number of points on the L-curve, i.e. $r \ll n$, the overall costs are of order $\mathcal{O}(mn)$.

References

[1] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716 – 723, 1974.

[2] Z. Bai and Y. Su. SOAR: A second order Arnoldi method for the solution of the quadratic eigenvalue problem. *SIAM J. Matrix Anal. Appl.*, 26:640 – 659, 2005.

[3] 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.

[4] 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.

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

[6] T.A. Davis and I.S. Duff. Algorithm 832: UMFPACK, an unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software*, 30:196 – 199, 2004.

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

[8] 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.

[9] W. Gander. Least squares with a quadratic constraint. *Numer. Math.*, 36:291 – 307, 1981.

[10] W. Gander, G.H. Golub, and U. von Matt. A constrained eigenvalue problem. *Lin.Alg.Appl.*, 114–115:815 – 839, 1989.

[11] G.H. Golub. Some modified matrix eigenvalue problems. *SIAM Review*, 15:318 – 334, 1973.

[12] 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.

[13] G.H. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21:215 – 223, 1979.

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

[15] H. Guo and R.A. Renaut. A regularized total least squares algorithm. In S. Van Huffel and P. Lemmerling, editors, *Total Least Squares and Errors-in-Variable Modelling*, pages 57 – 66, Dodrecht, The Netherlands, 2002. Kluwer Academic Publisher.

[16] P.C. Hansen. The discrete picard condition for discrete ill-posed problems. *BIT*, 30:658 – 672, 1990.

[17] P.C. Hansen. Analysis of discrete ill-posed problems by means of the L-curve. *SIAM Review*, 34(4):561 – 580, 1992.

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

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

[20] P.C. Hansen and D.P. O'Leary. The use of the L-curve in regularization of discrete ill-posed problems. *SIAM J. Scien. Comp.*, 14:1487 – 1503, 1993.

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

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

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

[24] J. Lampe and H. Voss. Solving regularized total least squares problems based on eigenproblems. Technical report, Institute of Numerical Simulation, Hamburg University of Technology, 2008. Submitted to Taiwanese Journal of Mathematics.

[25] R.-C. Li and Q. Ye. A Krylov subspace method for quadratic matrix polynomials with application to constrained least squares problems. *SIAM J. Matrix Anal. Appl.*, 25:405 – 428, 2003.

[26] S. Lu, S.V. Pereverzyev, and U. Tautenhahn. Regularized total least squares: computational aspects and error bounds. Technical Report 2007-30, Johann Radon Institute for Computational and Applied Mathematics; Austrian Academy of Sciences, Linz, Austria, 2007.

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

[28] V.A. Morozov. On the solution of functional equations by the method of regularization. *Soviet. Math. Dokl.*, 7:414 – 417, 1966.

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

[30] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461 – 464, 1978.

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

[32] 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.

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

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