TAIWANESE JOURNAL OF MATHEMATICS
Vol. 14, No. 3A, pp. 885-909, June 2010
This paper is available online at http://www.tjm.nsysu.edu.tw/

SOLVING REGULARIZED TOTAL LEAST SQUARES PROBLEMS BASED ON EIGENPROBLEMS

Jörg Lampe and Heinrich Voss

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 summarize two iterative methods which are based on a sequence of eigenproblems. 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.

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 [25, 35-37].

Received May 28, 2008, revised March 5, 2009, accepted March 7, 2009.
2000 *Mathematics Subject Classification*: 65F22.
Key words and phrases: Total least squares, Regularization, Ill-posedness, Nonlinear Arnoldi method.

The name total least squares appeared only recently in the literature [15], but under the names orthogonal regression or errors-in-variables this fitting method has a long history in the statistical literature. The univariate case ($n = 1$) was already discussed in 1877 by Adcock [1]. Further historical remarks can be found in [25, 37].

The TLS problem (1.2) can be analyzed (cf. [14, 37]) in terms of the singular value decomposition (SVD) of $[A, b]$

$$[A, b] = U\Sigma V^T,\ \Sigma = \text{diag}\{\sigma_1, \ldots, \sigma_{n+1}\},\ \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{n+1}.$$

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

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

In this paper our focus is on ill-conditioned problems which arise, for example, from the discretization of ill-posed problems such as integral equations of the first kind (cf. [8, 17]). 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 [9] 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 [3] adopted the Tikhonov regularization concept to stabilize the TLS solution, i.e. they considered the problem

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

where (as in the whole paper) $\|\cdot\|$ denotes the Euclidean norm, $L \in \mathbb{R}^{p\times n}$, $p \leq n$ is a regularization matrix and $\rho > 0$ is a penalty parameter.

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

$$\|[\Delta A, \Delta b]\|_F^2 = \min! \quad \text{subject to } (A + \Delta A)x = b + \Delta b,\ \|Lx\| \leq \delta, \tag{1.4}$$

where $\delta > 0$ is a regularization parameter, and $L \in \mathbb{R}^{p\times n}$, $p \leq n$ 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 on the solution. Stabilization of total least

squares problems by introducing a quadratic constraint was extensively studied in [4, 13, 16, 19, 20, 21, 24, 26, 31, 32].

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 (1.4) the constraint $\|Lx\| \leq \delta$ holds with equality. Under this condition Golub, Hansen and O'Leary [13] derived the following first order necessary conditions: The solution x_{RTLS} of problem (1.4) is a solution of the problem

$$(A^TA + \lambda_I I_n + \lambda_L L^TL)x = A^Tb, \tag{1.5}$$

where the parameters λ_I and λ_L are given by

$$\lambda_I = -\frac{\|Ax-b\|^2}{1+\|x\|^2}, \quad \lambda_L = \frac{1}{\delta^2}\big(b^T(b-Ax) - \frac{\|Ax-b\|^2}{1+\|x\|^2}\big). \tag{1.6}$$

This condition was used in the literature in two ways to solve problem (1.4): In [13, 16, 19, 26] λ_I is chosen as a free parameter; for fixed λ_L problem (1.5) is solved for (x, λ_I), and then λ_L is updated in a way that the whole process converges to the solution of (1.4). Conversely, in [20, 21, 31, 32] for a chosen parameter λ_I problem (1.5) is solved for (x, λ_L), which yields a convergent sequence of updates for λ_I.

In either case, problem (1.5) can be solved via the solution of an eigenvalue problem. To be more specific, for the first type 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 in every iteration step. Hence, in both cases one has to solve a sequence of eigenvalue problems which converge as the methods approach the solution of (1.4). 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. Thus, the only information that can be recycled from previous iterations in these methods is the eigenvector of the preceding step that can be used as initial vector. Much more information can be exploited in general iterative projection methods such as the nonlinear Arnoldi algorithm [38] which can be started with the entire search space of the previous eigenvalue problem.

In this paper we review both types of approaches mentioned in the penultimate paragraph, and we discuss efficient implementations with particular emphasis on the reuse of information gained in the convergence history. The paper is organized as follows. In section 2 the two basic algorithms are presented and the connection

to a sequence of eigenproblems is shown. Computational considerations concerning the details of the algorithms are discussed in section 3. Section 4 contains different numerical examples, comparing effort and computation time.

2. RTLS via Sequences of Eigenproblems

In the entire paper we assume that the condition

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

holds where K is an orthonormal basis of the kernel of L which guarantees that a solution of the RTLS problem (1.4) is attained, cf. [3]. Notice that the condition is empty if the regularization matrix L is nonsingular.

Our starting point for deriving methods for solving (1.4) with the equality constraint $\|Lx\| = \delta$ are the first order necessary conditions (1.5) and (1.6). We present two different iterative approaches.

The parameters λ_I and λ_L both depend on x and make the system of equation (1.5) hardly tractable. The basic idea is to keep one of the parameters λ_I or λ_L fixed for one iteration step and to treat the other one as a free parameter. In either case the resulting system can be solved as an eigenvalue problem which is linear and quadratic, respectively.

2.1. RTLS via a sequence of quadratic eigenvalue problems

The first algorithm is based on keeping the parameter λ_I fixed for one iteration step and let $\lambda := \lambda_L$ be a free parameter. The fixed parameter is updated and initialized as suggested in (1.6)

$$\lambda_I = \lambda_I(x^k) = -\frac{\|Ax^k - b\|^2}{1 + \|x^k\|^2}. \tag{2.2}$$

The first order optimality conditions then reads

$$B(x^k)x + \lambda L^T L x = A^T b, \quad \|Lx\|^2 = \delta^2, \tag{2.3}$$

with

$$B(x^k) = A^T A - f(x^k)I, \ f(x^k) = \frac{\|Ax^k - b\|^2}{1 + \|x^k\|^2} = -\lambda_I(x^k). \tag{2.4}$$

which suggests the following Algorithm 1.

Algorithm 1. RTLSQEP

Require: Initial vector x^1.

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

$$B_k x^{k+1} + \lambda L^T L x^{k+1} = A^T b, \quad \|Lx^{k+1}\|^2 = \delta^2 \tag{2.5}$$

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

Sima, Van Huffel and Golub [32] proposed to solve (2.3) via a quadratic eigenvalue problem similarly to the approach of Golub [12] 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 (2.5) is equivalent to

$$W_k z + \lambda z := L^{-T} B_k L^{-1} z + \lambda z = L^{-T} A^T b =: h, \quad z^T z = \delta^2. \tag{2.6}$$

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

$$(W_k + \lambda I)^2 u - \delta^{-2} h h^T u = 0. \tag{2.7}$$

The choice of the rightmost eigenvalue can be motivated as the maximal Lagrange multiplier that minimizes an underlying quadratic function, cf. [10], [21].

In [20] it is proven that the rightmost eigenvalue $\hat{\lambda}$ of (2.7) 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 (2.5) 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, when the solution of (2.5) may not be unique, is discussed by Gander, Golub and von Matt, cf. [11].

If $\operatorname{rank}(L) = p < n$ let $L^T L = U S U^T$ be the spectral decomposition of $L^T L$. Then (2.5) is equivalent to

$$\Big((AU)^T (AU) - f(x^k) I \Big) y + \lambda S y = (AU)^T b, \quad y^T S y = \delta^2, \tag{2.8}$$

with $y = U^T x^{k+1}$. Partitioning the matrices and vectors in (2.8) in block form

$$(AU)^T (AU) = \begin{pmatrix} T_1 & T_2 \\ T_2^T & T_4 \end{pmatrix}, \; S = \begin{pmatrix} S_1 & 0 \\ 0 & 0 \end{pmatrix}, \; (AU)^T b = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}, \; y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \tag{2.9}$$

where the leading blocks have dimension p, one gets

$$\begin{pmatrix} T_1 - f(x^k) I_p & T_2 \\ T_2^T & T_4 - f(x^k) I_{n-p} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} + \lambda \begin{pmatrix} S_1 y_1 \\ 0 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}. \tag{2.10}$$

Solving the second component for y_2

$$y_2 = (T_4 - f(x^k) I_{n-p})^{-1} (c_2 - T_2^T y_1),$$

and substituting in the first component one gets

$$\begin{aligned}\Big(T_1 - f(x^k)I_p - T_2(T_4 - f(x^k)I_{n-p})^{-1}T_2^T\Big)y_1 + \lambda S_1 y_1 \\ = (c_1 - T_2(T_4 - f(x^k)I_{n-p})^{-1}c_2).\end{aligned} \tag{2.11}$$

Hence, problem (2.8) is equivalent to the quadratic eigenvalue problem (2.7), where

$$W_k = S_1^{-1/2}\Big(T_1 - f(x^k)I_p - T_2(T_4 - f(x^k)I_{n-p})^{-1}T_2^T\Big)S_1^{-1/2}, \tag{2.12}$$

$$h_k = S_1^{-1/2}\Big(c_1 - T_2(T_4 - f(x^k)I_{n-p})^{-1}c_2\Big). \tag{2.13}$$

If (λ, u) is the eigenpair corresponding to the rightmost eigenvalue and u is normalized such that $u^T h_k = \delta^2$, and $z = (W_k + \lambda I)u$, then the solution of (2.5) is recovered by $x^{k+1} = Uy$ where

$$y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} S_1^{-1/2}z \\ (T_4 - f(x^k)I_{n-p})^{-1}(c_2 - T_2^T S_1^{-1/2}z) \end{pmatrix}. \tag{2.14}$$

When the constraint at the solution of (1.4) is active, i.e. if $\|Lx^*\|^2 = \delta^2$, then the following global convergence result holds, cf. [21].

Theorem 2.1. *Any limit point x^* of the sequence $\{x^k\}$ constructed by Algorithm 1 is a global minimizer of the optimization problem*

$$f(x) := \frac{\|Ax - b\|^2}{1 + \|x\|^2} = \min! \quad \textit{subject to } \|Lx\|^2 = \delta^2. \tag{2.15}$$

Remark 2.2. Sima et al. [32] proved the weaker convergence result, that every limit point of $\{x^k\}$ satisfies the first order conditions (1.5) and (1.6). ■

Remark 2.3. Beck and Teboulle [5] considered the minimization problem

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

which is equivalent to (1.4). They proved (even for a more general rational objective function) global convergence for Algorithm 1 with inequality constraint in (2.5). Notice however, that problem (2.5) with inequality constraint can not be solved via a quadratic eigenvalue problem, but requires a spectral decomposition of a matrix of dimension p in every iteration step (cf. [4]), and is therefore much more expensive. ■

Remark 2.4. The transformation of (2.3) to the quadratic eigenproblem (2.7) seems to be very costly if L is not invertible. Notice however, that typical regularization matrices are discrete versions of 1D first or second order derivatives

$$\hat{L} = \begin{pmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \end{pmatrix} \in \mathbb{R}^{(n-1)\times n} \text{ or } \tilde{L} = \begin{pmatrix} -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \end{pmatrix} \in \mathbb{R}^{(n-2)\times n}$$

for which the spectral decomposition is explicitly known, and matrix-vector products Uw can be evaluated cheaply by the discrete cosine transform. Likewise, for discrete Fredholm integral equations with 2D or 3D domain one can take advantage of the same technique combined with Kronecker representations of L. Moreover, the smoothing properties of $\hat{L}$ and $\tilde{L}$ are not deteriorated significantly if they are replaced by nonsingular versions like (cf. [6])

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

where the diagonal element $\alpha > 0$ is small, and

$$\tilde{L}_\alpha = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{pmatrix} \text{ or } \tilde{L}_\alpha = \begin{pmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}$$

with $\alpha \in \{1, 2\}$. Which one of these modifications is chosen depends on the behaviour of the solution of (1.4) close to the boundary. ■

2.2. RTLS via a sequence of linear eigenvalue problems

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

The following version of the first order optimality conditions was proved by Renaut and Guo in [26].

Theorem 2.5. *The solution x_{RTLS} of the RTLS problem* (1.4) *subject to the active constraint satisfies the augmented eigenvalue problem*

$$B(\lambda_L(x_{RTLS})) \begin{pmatrix} x_{RTLS} \\ -1 \end{pmatrix} = -\lambda_I(x_{RTLS}) \begin{pmatrix} x_{RTLS} \\ -1 \end{pmatrix}, \tag{2.16}$$

with

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

and λ_L *and* λ_I *as given in* (1.6)*.*

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* (1.6)*, then* $\hat{x}$ *satisfies* (1.5)*, and* $\hat{\lambda} = -f(\hat{x})$*.*

This condition suggested Algorithm 2 called RTLSEVP for obvious reasons.

Algorithm 2. RTLSEVP

Require: Initial guess $\lambda_L^0 > 0$ and $B_0 = B(\lambda_L^0)$

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

$$B_{k-1}y^k = \lambda y^k \tag{2.17}$$

for eigenpair (y^k, λ) corresponding to the smallest λ

3. Scale y^k such that $y^k = \begin{pmatrix} x^k \\ -1 \end{pmatrix}$
4. Update $\lambda_L^k = \lambda_L(x^k)$ and $B_k = B(\lambda_L^k)$
5. **end for**

The choice of the smallest eigenvalue is motivated by the fact that we are aiming at $\lambda = -\lambda_I$ (cf. (2.16)), and by the first order conditions (1.6) it holds that $-\lambda_I = f(x) = \frac{\|Ax-b\|^2}{1+\|x\|^2}$ is the function to be minimized.

The straightforward idea in [16] to update λ_L in line 4 with (1.6), i.e.

$$\lambda_L^{k+1} = \frac{1}{\delta^2}\Big(b^T(b - Ax^{k+1}) - \frac{\|Ax^{k+1} - b\|^2}{1 + \|x^{k+1}\|^2}\Big) \tag{2.18}$$

does not lead in general to a convergent algorithm.

To enforce convergence Renaut and Guo [26] proposed to determine a value θ such that the eigenvector $(x_\theta^T, -1)^T$ of $B(\theta)$ corresponding to the smallest eigenvalue of $B(\theta)$ satisfies the constraint $\|Lx_\theta\|^2 = \delta^2$, i.e. find a non-negative root $\hat{\theta}$ of the real function

$$g(\theta) := \frac{\|Lx_\theta\|^2 - \delta^2}{1 + \|x_\theta\|^2}. \tag{2.19}$$

Then the corresponding eigenvector $(x_{\hat{\theta}}^T, -1)^T$ is a solution of (2.16).

Unfortunately the last component of an eigenvector corresponding to the smallest eigenvalue of $B(\theta)$ need not be different from zero, and then $g(\theta)$ is not necessarily defined. To fill this gap the following generalization has been made in [19]:

Definition 2.6. Let $\mathcal{E}(\theta)$ denote the eigenspace of $B(\theta)$ corresponding to its smallest eigenvalue. Then

$$g(\theta) := \min_{y \in \mathcal{E}(\theta)} \frac{y^T N y}{y^T y} = \min_{(x^T, x_{n+1})^T \in \mathcal{E}(\theta)} \frac{\|Lx\|^2 - \delta^2 x_{n+1}^2}{\|x\|^2 + x_{n+1}^2} \tag{2.20}$$

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

This extends the definition of g to the case of eigenvectors with zero last components. The following theorem was proven in [19].

Theorem 2.7. *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 2.7 demonstrates that if $\hat{\theta}$ is a positive root of g, then $x := -y(1 : n)/y_{n+1}$ solves the RTLS problem (1.4) where y denotes an eigenvector of $B(\hat{\theta})$ corresponding to its smallest eigenvalue.

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

$$\frac{d\sigma_{n+1}(B(\theta))}{d\theta}\Big|_{\theta=\tilde{\theta}} = g(\tilde{\theta}). \tag{2.21}$$

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

Remark 2.9. Notice that g is not necessarily continuous. If the multiplicity of the smallest eigenvalue of $B(\theta)$ is greater than 1 for some θ_0, then g may have a jump discontinuity at θ_0. It may 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 (1.4). See [19] how to construct a solution in this case. Here we assume a unique solution, which is the generic case. ∎

2.3. Trust Region Subproblems

When solving constrained optimization problems by trust region methods a typical subproblem consists of minimizing a quadratic function subject to a quadratic constraint

$$\min \Psi(x) = \min \frac{1}{2} x^T H x + g^T x \qquad \text{subject to} \quad \|x\| \leq \Delta \tag{2.22}$$

with given $H = H^T \in \mathbb{R}^{n\times n}$, $x, g \in \mathbb{R}^n$ and $\Delta > 0$, which is called (Large scale) trust-region-subproblem (LSTRS) in [27, 28, 29, 30, 34].

The following characterization of a solution was proved in [33]

Lemma 2.10. *A feasible vector x^* is a solution to* (2.22) *with corresponding Lagrange multiplier λ^* if and only if x^*, λ^* satisfy $(H-\lambda^* I)x^* = -g$ with $H-\lambda^* I$ being positive semidefinite, $\lambda^* \leq 0$, and $\lambda^*(\Delta - \|x^*\|) = 0$.*

The connection to regularization problems is obvious when choosing $H = A^T A$ and $g = -A^T b$. Then problem (2.22) is equivalent to the regularized least squares problem (RLS) with a quadratic constraint

$$\|Ax - b\|^2 = \min! \quad \text{subject to } \|x\|^2 \leq \Delta^2. \tag{2.23}$$

For the special choice of the regularization matrix $L = I$, the RTLS problem (1.4) reduces to the RLS problem (2.23) if $\delta^2 \leq \|x_{LS}\|^2$ holds, cf. [13].

Remark 2.11. Lemma 2.10 implies that the general solution of (2.22) is

$$x = -(H - \lambda^* I)^\dagger g + z, \quad \text{with } z \in \mathcal{N}(H - \lambda^* I).$$

If the matrix $H - \lambda^* I \geq 0$ is nonsingular, then the solution is unique. The case that $H - \lambda^* I$ is singular is called hard case [28]. It indicates a (nearly) nonunique solution. ∎

Rojas, Santos and Sorensen [27, 28, 29, 30, 34] suggested to solve (2.22) via a sequence of linear eigenvalue problems:

$$B_\alpha \begin{pmatrix} 1 \\ x \end{pmatrix} = \lambda \begin{pmatrix} 1 \\ x \end{pmatrix} \tag{2.24}$$

with $B_\alpha = B(\alpha) = \begin{pmatrix} \alpha & g^T \\ g & H \end{pmatrix}$, which is equivalent to

$$\alpha - \lambda = -g^T x \quad \text{and} \quad (H - \lambda I)x = -g. \tag{2.25}$$

To fulfill the definiteness requirement of Lemma 2.10, λ is chosen as the smallest eigenvalue of (2.24). α is determined iteratively using a sequence of rational interpolations of the secular equation

$$\phi(\lambda) \equiv g^T (H - \lambda I)^\dagger g = -g^T x \tag{2.26}$$

where α is controlled such that

$$\phi'(\lambda) = g^T \left((H - \lambda I)^\dagger \right)^2 g = x^T x = \Delta^2. \tag{2.27}$$

This is essentially the same idea as determining the root of the monotonic function $g(\theta)$ from section 2.2.

The problem class LSTRS is more general than RLS problems, because the matrix H can also be indefinite and g does not have to lie in the rowspace of A, but it is straightforward to extend Algorithm 2 to solve the LSTRS problem (2.22) as long as the constraint is active. So the function $g(\theta)$ could also be used in this context.

Gander, Golub and von Matt [11] proved that the Lagrange equations of the LSTRS problem with equality constraint are equivalent to the quadratic eigenvalue problem

$$(H + \lambda I)^2 x - \Delta^{-2} g g^T x = 0 \tag{2.28}$$

considered in Section 2.1. Hence, if the quadratic constraint $\|x\|^2 - \Delta^2 = 0$ is active, then problem (2.22) can be solved via one quadratic eigenvalue problem using the techniques to be discussed in Section 3.1.

3. Computational Considerations

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 resp. linear eigenvalue problems. We give hints and advices how to save matrix vector multiplications (MatVecs) and keep the algorithms free of matrix-matrix products.

3.1. RTLSQEP - Algorithm 1

In the following subsections we discuss different approaches for solving the sequence of quadratic eigenvalue problems (2.7). It is important to note that (for not

too small dimensions) efficient methods are iterative projections methods where in each step the underlying problem (2.7) is projected to a search space $\mathcal{V} = \text{span}\{V\}$ which is expanded until the approximation by the solution of the projected problem

$$V^T\Big((W_k + \lambda I)^2 - \delta^{-2} h_k h_k^T\Big) V u = 0$$

is sufficiently accurate. Expanding the subspace by some vector v obviously requires only to append a new vector $W_k v$ and a new component $h_k^T v$ to the current projected matrix $W_k V$ and vector $h_k^T V$, respectively. Hence, one does not need the explicit matrix W_k in these algorithms but only a procedure to evaluate $W_k v$ for a given vector v.

3.1.1. Linearization

An obvious approach for solving the QEP

$$T_k(\lambda)u := \Big((W_k + \lambda I)^2 - \delta^{-2} h_k h_k^T\Big) u = 0 \tag{3.1}$$

at the k-th iteration step of Algorithm 1 is linearization, i.e. solving the linear eigenproblem

$$\begin{pmatrix} -2W_k & -W_k^2 + \delta^{-2} h_k h_k^T \\ I & 0 \end{pmatrix} \begin{pmatrix} \lambda u \\ u \end{pmatrix} = \lambda \begin{pmatrix} \lambda u \\ u \end{pmatrix}, \tag{3.2}$$

and choosing the maximal real eigenvalue, and the corresponding u-part of the eigenvector, which is an eigenvector of (3.1).

This approach is reasonable if the dimension n of problem (1.4) is small. For larger n it is not efficient to determine the entire spectrum of (3.2). In this case one could apply the implicitly restarted Arnoldi method implemented in ARPACK [22] (and included in MATLAB as function `eigs`) to determine the rightmost eigenvalue and corresponding eigenvector of (3.2). However, it is a drawback of linearization that symmetry properties of the quadratic problem are destroyed.

3.1.2. A Krylov subspace-type method

Li and Ye [23] presented a Krylov subspace projection method for monic QEPs

$$(\lambda^2 I - \lambda P_1 - P_0)u = 0 \tag{3.3}$$

which does not use a linearization but works with the matrices P_1 and P_0 directly. The method has particularly favorable properties if some linear combination of P_1 and P_0 is a matrix of small rank q. Then with $\ell + q + 1$ steps of an Arnoldi-type process a matrix $Q \in \mathbb{R}^{n \times \ell+q+1}$ with orthonormal columns and two matrices $H_1 \in \mathbb{R}^{\ell+q+1 \times \ell}$ and $H_0 \in \mathbb{R}^{\ell+q+1 \times \ell}$ with lower bandwidth $q+1$ are determined such that

$$P_1 Q(:, 1:\ell) = Q(:, 1:\ell+q+1) H_1 \text{ and } P_0 Q(:, 1:\ell) = Q(:, 1:\ell+q+1) H_0. \tag{3.4}$$

Approximations to eigenpairs of the monic QEP are then obtained from its orthogonal projection onto span$\{Q(:,1:\ell)\}$. The straightforward choice of $P_1 = 2W_k$ and $P_0 = W_k^2 - \delta^{-2} h_k h_k^T$ results in the projected QEP

$$\big(\lambda^2 I - \lambda H_1(1:\ell,1:\ell) - H_0(1:\ell,1:\ell)\big)\tilde{u} = 0. \tag{3.5}$$

But for this straightforward choice of P_0 and P_1 usually no linear combination is of small rank q, and the matrices H_0 and H_1 will become full.

Applying $\ell+2$ steps of the algorithm of Li and Ye with $P_1 = W_k$ and $P_0 = h_k h_k^T$ one obtains a matrix $Q \in \mathbb{R}^{n\times\ell+2}$ (different from the one in (3.4)) with orthonormal columns such that

$$P_0 Q(:,1:\ell) = Q(:,1:\ell+2) H_0(1:\ell+2,1:\ell), \tag{3.6}$$

$$P_1 Q(:,1:\ell) = Q(:,1:\ell+2) H_1(1:\ell+2,1:\ell). \tag{3.7}$$

Hence,

$$\begin{aligned} P_1^2 Q(:,1:\ell) &= P_1 Q(:,1:\ell+2) H_1(1:\ell+2,1:\ell) \\ &= Q(:,1:\ell+4) H_1(1:\ell+4,1:\ell+2) H_1(1:\ell+2,1:\ell) \end{aligned} \tag{3.8}$$

and the orthogonal projection of problem (3.3) to $\mathcal{Q} := \text{span}\{Q(:,1:\ell)\}$ reads

$$\big(\lambda^2 I - 2\lambda H_1(1:\ell,1:\ell) - \hat{H}_0\big)\tilde{u} = 0 \tag{3.9}$$

with $\hat{H}_0(1:\ell,1:\ell) = \delta^{-2} H_0(1:\ell,1:\ell) - H_1(1:\ell+2,1:\ell)^T H_1(1:\ell+2,1:\ell)$.

As a consequence of rank$\{P_0\} = 1$ it follows that H_1 and $\hat{H}_0$ are symmetric pentadiagonal matrices, and the cost for expanding the subspace $\mathcal{Q}$ by one vector is one matrix-vector product (1 MatVec for short) and 9 level-1 operations, cf. [23].

Because we have to solve a sequence of QEPs, and W_k and h_k are converging it is favorable to use the solution vector of the preceding QEP as initial vector of the Arnoldi-type process.

The rightmost eigenvalue and corresponding eigenvector of a projected problem can be determined cheaply by linearization and a dense eigensolver since the dimensions of the projected problems are quite small.

The method is terminated if the residual $\|(W_k + \lambda_{rm})^2 u_{rm} - \delta^{-2} h h^T u_{rm}\|$ at the rightmost Ritz pair (λ_{rm}, u_{rm}) is small enough. Notice that the residual norm can be evaluated inexpensively taking advantage of (3.6), (3.7), (3.8) with a delay of 2 expansion steps. Computing the residual of (2.5) is expensive due to the back transformation to x^{k+1}. Since the residuals are not needed explicitly we recommend just to monitor the rightmost eigenvalue of the sequence of projected QEPs.

3.1.3. Second Order Arnoldi Reduction

Another approach is the Second Order Arnoldi Reduction (SOAR for short) introduced by Bai and Su [2] for solving the large scale QEP $(\lambda^2 M+\lambda D+K)u = 0$. The main idea is based on the observation that the Krylov space of the linearization

$$\begin{pmatrix} P_1 & P_0 \\ I & O \end{pmatrix} \begin{pmatrix} \lambda u \\ u \end{pmatrix} = \lambda \begin{pmatrix} \lambda u \\ u \end{pmatrix} \tag{3.10}$$

with $P_1 = -M^{-1}D, P_0 = -M^{-1}K$ and initial vector $\begin{pmatrix} r_0 \\ 0 \end{pmatrix}$ has the form

$$\mathcal{K}_\ell = \left\{ \begin{pmatrix} r_0 \\ 0 \end{pmatrix}, \begin{pmatrix} r_1 \\ r_0 \end{pmatrix}, \begin{pmatrix} r_2 \\ r_1 \end{pmatrix}, \ldots, \begin{pmatrix} r_{\ell-1} \\ r_{\ell-2} \end{pmatrix} \right\}, \tag{3.11}$$

where

$$\begin{aligned} r_1 &= P_1 r_0, \\ r_j &= P_1 r_{j-1} + P_0 r_{j-2}, \text{ for } j \geq 2. \end{aligned} \tag{3.12}$$

The entire information on $\mathcal{K}_\ell$ is therefore contained in the second order Krylov space

$$\mathcal{G}_\ell(P_1, P_0; r_0) = \text{span}\{r_0, r_1, \ldots, r_{\ell-1}\}. \tag{3.13}$$

Bai and Su [2] presented an Arnoldi type algorithm based on the two term recurrence (3.12) for computing an orthonormal basis $Q_\ell \in \mathbb{R}^{n\times\ell}$ of $\mathcal{G}_\ell(P_1, P_0; r_0)$. The orthogonal projection of the QEP (3.1) onto $\mathcal{G}_\ell(P_1, P_0; r_0)$ is the structure-preserving variant of projecting the linearized problem (3.10) onto $\mathcal{K}_\ell$ from (3.11). The SOAR approach has the same approximation quality, but outperforms the Arnoldi method applied to the linearized problem.

Since the QEPs (3.1) are monic there is no need to perform a LU-decomposition of the matrix $M = I$ and the matrices $P_1 = -2W_k$ and $P_0 = -W_k^2 + \delta^{-2} h_k h_k^T$ are directly available.

The current second order Krylov space $\mathcal{G}_\ell(P_1, P_0; r_0)$ is expanded by $\tilde{q} := P_1 q_\ell + P_0 p_\ell$, where $p_\ell = Q_\ell s_\ell$ is some vector $p_\ell \in \text{span}\{Q_\ell\}$. Orthogonalization yields the direction of the new basis element

$$\begin{aligned} q_{\ell+1} &= (I - Q_\ell Q_\ell^T)(P_1 q_\ell + P_0 p_\ell) \\ &= (I - Q_\ell Q_\ell^T)(-2W_k q_\ell - W_k^2 Q_\ell s_\ell + \delta^{-2} h_k h_k^T Q_\ell s_\ell) \end{aligned}$$

where $W_k Q_\ell s_\ell$ can be updated from the previous step. Hence, expanding the search space $\mathcal{G}_\ell(P_1, P_0; r_0)$ requires 2 MatVecs.

A single step of the SOAR methods costs essentially twice as much as the one of the Krylov-type method in section 3.1.2. On the other hand, SOAR builds up a 2nd order Krylov space which has better approximation properties than the search space of the method of Li and Ye, however, this gain usually is not enough to balance the higher cost.

We now present a variant of the SOAR method that reduces the cost per expansion to 1 MatVec. This approach approximates the second order Krylov space $\mathcal{G}_\ell(P_1, P_0; r_0)$ by $\mathcal{G}_\ell(P_1, \tilde{P}_0; r_0)$ with $\tilde{P}_0 = \delta^{-2} h_k h_k^T$. In this case the current search space is expanded by the vector

$$\begin{aligned}\hat{q}_{\ell+1} &= (I - \hat{Q}_\ell \hat{Q}_\ell^T)(-2W_k \hat{q}_\ell + \delta^{-2} h_k h_k^T \hat{Q}_\ell \hat{s}_\ell) \\ &= (I - \hat{Q}_\ell \hat{Q}_\ell^T)(-2W_k \hat{q}_\ell - \hat{Q}_\ell (W_k \hat{Q}_\ell)^T (W_k \hat{Q}_\ell) \hat{s}_\ell + \delta^{-2} h_k h_k^T \hat{Q}_\ell \hat{s}_\ell),\end{aligned}$$

and the approximate eigenpairs are obtained from the projected problem

$$\Big(\lambda^2 I + 2\lambda \hat{Q}_\ell^T (W_k \hat{Q}_\ell) + (W_k \hat{Q}_\ell)^T (W_k \hat{Q}_\ell) - \delta^{-2} (\hat{Q}_\ell^T h_k)(\hat{Q}_\ell^T h_k)^T \Big) z = 0.$$

The search spaces of this variant and of the original SOAR coincide when the starting vector is chosen to be $r_0 = h_k$. Then

$$\mathcal{G}_\ell(-2W_k, -W_k^2 + \delta^{-2} h_k h_k^T; h_k) = \mathcal{G}_\ell(-2W_k, \delta^{-2} h_k h_k^T; h_k) = \mathcal{K}_\ell(W_k; h_k), \tag{3.14}$$

and the second order Krylov spaces $\mathcal{G}_\ell$ both reduce to the usual Krylov space. However, this choice is often not appropriate. For an arbitrary r_0 it holds that

$$\begin{aligned}\mathcal{G}_\ell(-2W_k, -W_k^2 + \delta^{-2} h_k h_k^T; r_0) &\subseteq \big[\mathcal{K}_\ell(W_k; r_0) \cup \mathcal{K}_{\ell-2}(W_k; h_k)\big] \\ \mathcal{G}_\ell(-2W_k, \delta^{-2} h_k h_k^T; r_0) &\subseteq \big[\mathcal{K}_\ell(W_k; r_0) \cup \mathcal{K}_{\ell-2}(W_k; h_k)\big].\end{aligned}$$

3.1.4. Nonlinear Arnoldi method

A further method for solving the QEP $T_k(\lambda)u = 0$ from (3.1) is the nonlinear Arnoldi method [38] which applies to much more general nonlinear eigenvalue problems $T(\lambda)u = 0$ than (2.7).

In Algorithm 1 a sequence of quadratic eigenvalue problems has to be solved, and the convergence of the matrices and vectors

$$W_k = C - f(x^k)\tilde{S} - D(T_4 - f(x^k) I_{n-p})^{-1} D^T, \tag{3.15}$$

$$h_k = g_1 - D(T_4 - f(x^k) I_{n-p})^{-1} c_2. \tag{3.16}$$

with $C = S_1^{-1/2} T_1 S_1^{-1/2}$, $\tilde{S} = S_1^{-1}$, $D = S_1^{-1/2} T_2$ and $g_1 = S_1^{-1/2} c_1$ (cf. (2.12) and (2.13)) suggest to reuse information from the previous steps when solving problem (3.1) in step k, cf. [20, 21].

Algorithm 3. Nonlinear Arnoldi

Require: Initial basis V, $V^TV = I$

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

For the two Krylov subspace methods in subsections 3.1.2 and 3.1.3 the only degree of freedom is the choice of the initial vector, and we therefore start in step k with the solution u^{k-1} of the preceding step. The nonlinear Arnoldi method allows thick starts, i.e. when solving $T_k(\lambda)u = 0$ in step k Algorithm 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 $V^TT_{k-1}(\lambda)V\tilde{u} = 0$.

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

- A suitable initial basis V of Algorithm 1 for the first quadratic eigenvalue problem (2.7) was determined by a small number ($\ell = 6$, e.g.) of Lanczos steps applied to the linear eigenproblem $W_1z = \lambda z$ with a random vector $r_0 \in \mathbb{R}^p$ because this is cheaper than executing the nonlinear Arnoldi method.
- Since the dimensions of the projected problems are small they can be solved by linearization and a dense eigensolver like the QR algorithm.
- In our numerical examples it turned out that we obtained fast convergence without preconditioning, so we simply set $PC = I$.
- The representation of W_k and h_k in (3.15) and (3.16) demonstrates that the projected eigenvalue problem

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

  can be determined efficiently if the matrices CV, SV, D^TV and g_1^TV are known. These can be updated cheaply by appending in every iteration step of the nonlinear Arnoldi method one column and component to the current matrices and vector, respectively. Since the number $n - p$ of columns of D is very small the cost is essentially one MatVec $C\tilde{v}$.

- The determination of the residual $r = T_k(\mu)u$ costs another MatVec with W_k. Due to $T_k(\mu)u = (W_k+\mu I)(W_k+\mu I)V\tilde{u} - \delta^{-2}h_k h_k^T V\tilde{u}$ we can make use of the stored matrices CV, SV, D^TV and g_1^TV to obtain $(W_k+\mu I)V\tilde{u}$ and $h_k^TV\tilde{u}$ easily, but one further multiplication with $(W_k+\mu I)$ has to be executed. So, one iteration step of the nonlinear Arnoldi method roughly costs 2 MatVecs.

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

The cost of an outer iteration in Algorithm 1, namely to obtain updates of W_kV and h_k^TV from the preceding matrices $W_{k-1}V$ and h_{k-1}^TV (cf. (3.15) and (3.16)) is only a fourth of the cost of one inner iteration in the nonlinear Arnoldi method. Evaluating $f(x^k)$ costs 1 or 2 MatVecs (dependent on the structure of U) and the cost of an inner iteration are 2 MatVecs with W_k, that is 4 resp. 8 MatVecs.

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 4 of Algorithm 3 is replaced by $\|r\|/\|r_0\| > 0.01$ with the initial residual r_0 calculated in line 3. This approach leads to more outer iterations but overall to less inner iterations.

The early update variant reduces the overall computation time substantially when compared to the standard version. Implementing early update strategies in the Krylov-type algorithms of Subsections 3.1.2 and 3.1.3 destroyed the convergence of the overall process.

3.2. RTLSEVP - Algorithm 2

It is a common demand of both approaches, RTLSQEP and RTLSEVP that one has to solve a converging sequence of eigenvalue problems which again suggests to reuse information gained in previous iteration steps. An advantage of RTLSEVP over RTLSQEP is the fact that for $p < n$ (i.e. if L is not a nonsingular matrix) we do not have to reduce the first order conditions (2.5) to the range of L and hence do not need a spectral decomposition of L^TL although this can be implemented inexpensively in many important cases (cf. Remark 2.4).

3.2.1. Solving the sequence of linear eigenproblems

Renaut and Guo [26] proposed to determine the minimum eigenvalue of

$$B(\theta_k)y = \big(M + \theta_k N\big)y = \lambda y \tag{3.17}$$

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

Similar to the approach in RTLSQEP the entire information gathered in previous iteration steps can be employed solving (3.17) via the nonlinear Arnoldi Algorithm 3 with thick starts applied to

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

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

The projected problem

$$V^T T_k(\mu) V\tilde{u} = \big(([A,b]V)^T([A,b]V) + \theta_k V^T N V - \mu I\big)\tilde{u} = 0 \tag{3.18}$$

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

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

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

The evaluation of the residual

$$r = T_k(\mu)V\tilde{u} = [A,b]^T([A,b]V)\tilde{u} + \theta_k N V\tilde{u} - \mu u$$

in line 3 and 9 costs another MatVec with $[A,b]^T$. Hence, one inner iteration step of the nonlinear Arnoldi method costs 2 MatVecs resp. 4 MatVecs in the case of an unstructured and full regularizarion matrix L. These are half the cost of an inner iteration step of the nonlinear Arnoldi applied in the RTLSQEP algorithm, cf. 2.11.

3.2.2. Root-finding algorithm

Assuming that g is continuous and strictly monotonically decreasing Renaut and Guo [26] derived the update

$$\theta_{k+1} = \theta_k + \frac{\theta_k}{\delta^2} g(\theta_k) \tag{3.19}$$

for solving $g(\theta) = 0$ where $g(\theta)$ is defined in (2.19), and at step k, $(x_{\theta_k}^T, -1)$ is the eigenvector of $B(\theta_k)$ corresponding to its minimal eigenvalue. Since this sequence usually does not converge, an additional backtracking was included, i.e. the update was modified to

$$\theta_{k+1} = \theta_k + \iota \frac{\theta_k}{\delta^2} g(\theta_k) \tag{3.20}$$

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

We propose a root-finding algorithm taking into account the typical shape of $g(\theta)$ which is shown in Figure 3.1. Left of its root $\hat{\theta}$ the slope of g is often very steep, while right of $\hat{\theta}$ it is approaching its limit $-\delta^2$ quite quickly. This makes it difficult to determine $\hat{\theta}$.

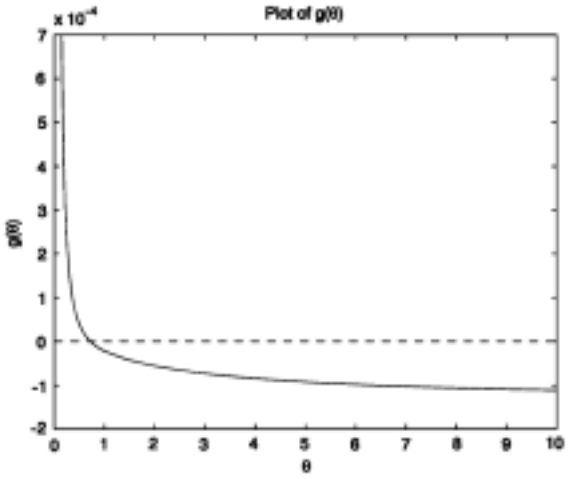

Fig. 3.1. Plot of a typical function $g(\theta)$.

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

$$\theta_1 < \theta_2 < \theta_3 \quad \text{and} \quad g(\theta_1) > 0 > g(\theta_3). \tag{3.21}$$

We determine the rational interpolation

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

and p is chosen such that $h(g(\theta_j)) = \theta_j$, $j = 1, 2, 3$, and we evaluate $\theta_4 = h(0)$. In exact arithmetic $\theta_4 \in (\theta_1, \theta_3)$, and we replace θ_1 or θ_3 by θ_4 such that the new triple satisfies (3.21).

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

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

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

To initialize Algorithm 2 we determine three values θ_i such that not all $g(\theta_i)$ have the same sign. Given $\theta_1 > 0$ we multiply either by 0.01 or 100 depending on the sign of $g(\theta_1)$ and obtain after very few steps an interval that contains the root of g.

If a discontinuity at or close to the root is encountered, then a very small $\epsilon_\theta = \theta_3 - \theta_1$ appears with relatively large $g(\theta_1) - g(\theta_3)$. In this case we terminate the iteration and determine the solution as described in [19].

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

Remark 3.1. In section 2.3 another problem class was mentioned, where a sequence of linear eigenproblems (2.24) has to be solved. In [28] the implicitly restarted Lanczos method (IRLM) is proposed as eigensolver, that can make use of the solution vector of the preceding eigenproblem as starting vector for the next iteration. A straightforward idea is to use the nonlinear Arnoldi in this context as well, making use of all previous information by thick starts. Another idea is to use the function $g(\theta)$ defined in (2.20) for updating the parameter α, notice that $\theta = \frac{b^T b-\alpha}{1+\Delta^2}$ holds. Then the proposed enclosing root-finding algorithm could be applied as well.

Remark 3.2. Let θ_c be the value where $B(\theta)$ gets indefinite. The root of g is definitely contained in the interval $[0, \theta_c]$. If for all θ_k holds $\theta_k \in [0, \theta_c]$, then $B(\theta_k) > 0$ and the sequence of EVPs can be interpreted of a sequence of TLS problems. A TLS problem (1.2) is just the determination of the smallest eigenpair of a positive definite matrix.

4. Numerical Examples

To evaluate the performance of Algorithms 1 and 2 for large dimensions we use a 1D and a 2D test example from Hansen's *Regularization Tools*, [18].Two

functions *heat(1)* and *tomo*, which are both discretizations of integral equations, are used to generate matrices $A_{\text{true}} \in \mathbb{R}^{m\times n}$, right hand sides $b_{\text{true}} \in \mathbb{R}^m$ and solutions $x_{\text{true}} \in \mathbb{R}^n$ such that

$$A_{\text{true}} x_{\text{true}} = b_{\text{true}}.$$

In all cases the matrices A_{true} and $[A_{\text{true}}, b_{\text{true}}]$ are ill-conditioned.

To construct a suitable RTLS problem, the norm of b_{true} is scaled such that $\|b_{true}\|_2 = \max_i \|A_{true}(:,i)\|_2$ holds. x_{true} is 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, $aver = \sum(\sum(abs[A_{true}, b_{true}]))/(m(n+1))$. We add white noise of level 1-10% to the data and obtained the systems $Ax \approx b$ where $A = A_{\text{true}} + \sigma E$ and $b = b_{\text{true}} + \sigma e$, with $\sigma = aver \cdot (0.01 \ldots 0.1)$ and the elements of E and e are independent random variables with zero mean and unit variance.

The numerical test were run on a PentiumR4 computer with 3.4 GHz and 8GB RAM under MATLAB R2007b. Tables 1 and 2 contain the CPU times in seconds averaged over 100 random simulations. for the 1D problem the dimensions $n = 1000$, $n = 2000$, $n = 4000$ are chosen and for the 2D problem $n = 900$, $n = 1600$, $n = 2500$ which correspond to a solution on a 30x30, 40x40 and 50x50 grid. In all examples the number of the rows of A is twice the number of the columns, i.e. $m = 2n$. The noise levels are 1% and 10% for both examples.

For the 1D problem *heat(1)* the regularization matrix $\hat{L} \in \mathbb{R}^{n-1\times n}$ approximates the first order derivative, cf. Remark 2.4. The δ_1 is chosen to be $\delta_1 = 0.8\|\hat{L}x_{\text{true}}\|$. It was also tested a slightly disturbed variant $\hat{L}_\alpha$ with $\alpha = 0.1$ which is denoted by 'b' in comparison to the unperturbed $\hat{L}$ denoted by 'a' in the Table 1.

For the 2D example *tomo* a discrete version of the 2D first order derivative operator $\hat{L}$ is used. Here again a slightly disturbed variant to make $\hat{L}_\alpha$ have full rank is denoted by 'b' whereas the original $\hat{L}$ is denoted by 'a' in Table 2 below. The δ_2 is chosen to be $\delta_2 = 0.3\|\hat{L}x_{\text{true}}\|$.

In Algorithm 1 the outer iteration was terminated if the $f(x_k)$ has converged, i.e. two subsequent values do not differ relatively by more than 0.1%. If not a regular matrix $\hat{L}$ is used, the matrix U is needed, but never set up explicitely. Performing matrix vector multiplication with U can be done efficiently in less than $\mathcal{O}(n^2)$ by using either the discrete cosine transform in the 1D case or the Kronecker-product structure in the 2D case. The quadratic eigenproblems are solved by the presented solvers from subsections 3.1.2, 3.1.3 and 3.1.4. From subsection 3.1.3 it is taken the variant of the SOAR method that needs less MatVecs.

Algorithm 2 is terminated if the $g(\theta_k)$ is sufficiently close to zero. i.e. less than 10^{-10}. A preconditioner is only needed in RTLSEVP and was calculated with UMFPACK [7], i.e. MATLAB's $[L, U, P, Q] = lu(N)$, with a slightly perturbed N to make it nonsingular. The eigenproblems are solved by the nonlinear Arnoldi

method according to section 3.2.1, and the root-finding algorithm from section 3.2.2 is applied.

In Table 1 the following shortcuts are used: 'LY' denotes the Krylov subspace-type method from Li and Ye, 'SO' the Second Order Arnoldi Reduction and 'NLA' denotes the nonlinear Arnoldi method. Algorithms 1 and 2 are denoted by '1' and '2' and the regularization matrices $\hat{L}$ in the cases 'a' and 'b' are unperturbed and perturbed respectively.

Table 1. Example *heat(1)*, average CPU time in seconds

$noise$	n	LY 1a	LY 1b	SO 1a	SO 1b	NLA 1a	NLA 1b	NLA 2a
1%	1000	0.53	0.47	0.52	0.63	0.35	0.36	0.19
	2000	1.28	1.19	1.13	1.02	1.08	0.99	0.60
	4000	4.94	4.68	4.37	3.78	4.21	3.88	2.65
10%	1000	0.55	0.46	0.48	0.45	0.36	0.32	0.19
	2000	1.37	1.18	1.19	0.99	1.07	0.98	0.61
	4000	4.95	4.67	4.31	3.73	4.17	3.92	2.54

Table 2. Example *tomo*, average CPU time in seconds

$noise$	n	Li&Ye 1a	SOAR 1a	NL Arn. 1a	NL Arn. 1b	NL Arn. 2a
1%	30x30	0.77	1.01	1.02	1.24	0.20
	40x40	2.62	2.55	2.07	2.81	0.54
	50x50	6.93	6.44	4.78	6.03	3.86
10%	30x30	0.77	1.02	1.00	1.23	0.21
	40x40	2.63	2.56	2.02	2.88	0.56
	50x50	6.89	6.38	4.80	5.98	3.83

When using the RTLSQEP for problem *heat*(1) roughly 100 MatVecs are performed in about 3 outer iterations. This is the case for all tested eigensolvers, both noise levels and different problem sizes. A matrix vector multiplication is the most expensive operation within the algorithms, so the computation times are about equal. Algorithm 2 only needs approximately 50 MatVecs and this results in half the time.

In the 2D problem *tomo* Algorithm 1 roughly needs 200-300 MatVecs, due to a lot of outer iterations. From the different quadratic eigensolvers the nonlinear Arnoldi with the unperturbed regularization matrix $\hat{L}$ (i.e. 'NL Arn. 1a') is the best choice. The computation time is much less when using the RTLSEVP algorithm with about 60 MatVecs for the smaller problems and about 150 MatVecs for the 50x50 example.

The RTLSEVP turned out to be superior to the RTLSQEP algorithm, at least in the chosen examples. Both approaches do converge in much less MatVecs than the

dimension of the problem, so the computational complexity of both approaches is of order $\mathcal{O}(n^2)$.

5. Conclusions

The RTLSQEP algorithm for solving the RTLS problem (1.4) is very efficient when combined with iterative projection methods in the inner loop like the Li/Ye method, SOAR or the nonlinear Arnoldi method. The RTLSEVP algorithm is also very efficient when combined with the nonlinear Arnoldi in the inner loop and a suitable root-finding algorithm for $g(\theta) = 0$ based on a rational inverse interpolation. The computational complexity of the proposed approaches is kept at the order of $\mathcal{O}(n^2)$. We present a detailed description of an efficient implementation of the different parts of the algorithms.

Acknowledgment

The work of Jörg Lampe was supported by the German Federal Ministry of Education and Research (Bundesministerium für Bildung und Forschung – BMBF) under grant number 13N9079.

References

1. R. Adcock, Note on the method of least squares, *The Analyst*, **4** (1877), 183-184.
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** (2005), 540-659.
3. A. Beck and A. Ben-Tal, On the solution of the Tikhonov regularization of the total least squares problem, *SIAM J. Optim.*, **17** (2006), 98-118.
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** (2006), 425-445.
5. A. Beck and M. Teboulle, A convex optimization approach for minimizing the ratio of indefinite quadratic functions over an ellipsoid, *Math. Progr.*, **118** (2009), 13-35.
6. D. Calvetti, L. Reichel and A. Shuibi, Invertible smoothing preconditioners for linear discrete ill-posed problems, *Appl. Numer. Math.*, **54** (2005), 135-149.
7. T. Davis and I. Duff, Algorithm 832: UMFPACK, an unsymmetric-pattern multifrontal method, *ACM Transactions on Mathematical Software*, **30** (2004), 196-199.
8. H. Engl, M. Hanke and A. Neubauer, *Regularization of Inverse Problems*, Kluwer, Dodrecht, The Netherlands, 1996.
9. R. Fierro, G. Golub, P. Hansen and D. O'Leary, Regularization by truncated total least squares, *SIAM J. Sci. Comput.*, **18** (1997), 1223-1241.

10. W. Gander, Least squares with a quadratic constraint, *Numer. Math.*, **36** (1981), 291-307.

11. W. Gander, G. Golub and U. von Matt, A constrained eigenvalue problem, *Lin. Alg. Appl.*, **114-115** (1989), 815-839.

12. G. Golub, Some modified matrix eigenvalue problems, *SIAM Review*, **15** (1973), 318-334.

13. G. Golub, P. Hansen and D. O'Leary, Tikhonov regularization and total least squares, *SIAM J. Matrix Anal. Appl.*, **21** (1999), 185-194.

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

15. G. Golub and C. Van Loan, An analysis of the total least squares problem, *SIAM J. Numer. Anal.*, **17** (1980), 883-893.

16. H. Guo and R. Renaut, A regularized total least squares algorithm, in: *Total Least Squares and Errors-in-Variable Modelling*, S. Van Huffel and P. Lemmerling, eds., Dodrecht, The Netherlands, 2002, Kluwer Academic Publisher, pp. 57-66.

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

18. ——, Regularization tools version 4.0 for Matlab 7.3, *Numer. Alg.*, **46** (2007), 189-194.

19. J. Lampe and H. Voss, A fast algorithm for solving regularized total least squares problems, *Electr. Trans. Numer Anal.*, **31** (2008), 12-24.

20. ——, On a quadratic eigenproblem occuring in regularized total least squares, *Comput. Stat. Data Anal.*, **52(2)** (2007), 1090-1102.

21. ——, Global convergence of RTLSQEP: a solver of regularized total least squares problems via quadratic eigenproblems, *Math. Modelling Anal.*, **13** (2008), 55-66.

22. R. Lehoucq, D. Sorensen and C. Yang, *ARPACK Users' Guide. Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, SIAM, Philadelphia, 1998.

23. 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** (2003), 405-428.

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

25. I. Markovsky and S. Van Huffel, Overview of total least squares methods, *Signal Processing*, **87** (2007), 2283-2302.

26. R. Renaut and H. Guo, Efficient algorithms for solution of regularized total least squares, *SIAM J. Matrix Anal. Appl.*, **26** (2005), 457-476.

27. M. Rojas, Regularization of large-scale ill-posed least squares problems, Tech. Report Report 96-32, Dept. Comput. Appl. Math., Rice University, Nouston, 1996, *Internat. J. Comput. Math.*, to appear.

28. M. Rojas, S. Santos and D. Sorensen, A new matrix-free algorithm for the large-scale trust-region subproblem, *SIAM J. Optim.*, **11** (2000), 611-646.

29. ——, LSTRS: MATLAB software for large-scale trust-region subproblems and regularization, *ACM Trans. Math. Software*, **34** (2007), Article 11.

30. M. Rojas and D. Sorensen, A trust-region approach for the regularization of large-scale discrete forms of ill-posed problems, *SIAM J. Sci. Comput.*, **23** (2002), 1843-1861.

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

32. D. Sima, S. Van Huffel and G. Golub, Regularized total least squares based on quadratic eigenvalue problem solvers, *BIT Numerical Mathematics*, **44** (2004), 793-812.

33. D. Sorensen, Newton's method with a model trust-region modification, *SIAM J. Numer. Anal.*, **19** (1982), 409-426.

34. ——, Minimization of a large-scale quadratic function subject to a spherical constraint, *SIAM J. Optim.*, **7** (1997), 141-161.

35. S. Van Huffel, ed., *Recent Advances in Total Least Squares Techniques and Errors-in Variables Modeling: Proceedings of the Second International Workshop on Total Least Squares Techniques and Errors-in Variables Modeling*, SIAM, Philadelphia, 1997.

36. S. Van Huffel and P. Lemmerling, eds., *Total Least Squares and Errors-in-Variables Modeling: Analysis, Algorithms and Applications*, Kluwer, Dordrecht, Boston, London, 2002.

37. 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.

38. H. Voss, An Arnoldi method for nonlinear eigenvalue problems, *BIT Numerical Mathematics*, **44** (2004), 387-401.

Jörg Lampe and Heinrich Voss
Institute of Numerical Simulation,
Hamburg University of Technology,
D-21071 Hamburg,
Germany
E-mail: joerg.lampe@tu-harburg.de
voss@tu-harburg.de