

A truncated-CG style method for symmetric generalized eigenvalue problems*

P.-A. Absil[†] C. G. Baker[‡] K. A. Gallivan[‡]

Preprint, 02-Nov-2005

Abstract

A numerical algorithm is proposed for computing an extreme eigenpair of a symmetric/positive-definite matrix pencil (A, B) . The leftmost or the rightmost eigenvalue can be targeted. Knowledge of (A, B) is only required through a routine that performs matrix-vector products. The method has excellent global convergence properties and its local rate of convergence is superlinear. It is based on a constrained truncated-CG trust-region strategy to optimize the Rayleigh quotient, in the framework of a recently-proposed trust-region scheme on Riemannian manifolds.

Key words: generalized eigenvalue problem, extreme eigenvalues, truncated conjugate gradient, Steihaug-Toint, trust-region, global convergence, superlinear convergence, matrix-free

1 Introduction

The generalized eigenvalue problem

$$Ax = \lambda Bx,$$

where A and B are $n \times n$ real symmetric matrices with B positive definite, arises in many scientific applications [Saa92]. The symmetric/positive-definite pencil (A, B) is known to admit n real eigenvalues $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{n-1} \leq \lambda_n$ with associated B -orthonormal eigenvectors v_1, \dots, v_n (see [Ste01]). We call (λ_1, v_1) and (λ_n, v_n) the *leftmost* and *rightmost* eigenpairs, respectively.

Single vector iterations [Par80, BDDR00] are the simplest methods for the eigenproblem. It is worthwhile considering them briefly, as their advantages and drawbacks are ubiquitous in eigencomputation methods. If the matrix A is positive definite, the power method

$$Bx_{k+1} = Ax_k \tau_k, \tag{1}$$

*This work was supported by the US National Science Foundation under Grants ACI0324944 and CCR9912415, and by the School of Computational Science of Florida State University through a postdoctoral fellowship. This research was initiated while the first author was a Research Fellow with the Belgian National Fund for Scientific Research (FNRS) at the University of Liège.

[†]Department of Mathematical Engineering, Université catholique de Louvain, 1348 Louvain-la-Neuve, Belgium, and Peterhouse, University of Cambridge, Cambridge CB2 1RD, UK. URL: <http://www.inma.ucl.ac.be/~absil/>

[‡]School of Computational Science, Florida State University, Tallahassee, FL 32306-4120, USA. URLs: <http://www.csit.fsu.edu/~cbaker>, [~gallivan](http://www.csit.fsu.edu/~gallivan)

where τ_k is a normalizing factor, converges to the principal eigenvector v_n of (A, B) from almost all initial points; but the rate of convergence is only linear and becomes very slow when the eigenvalues of (A, B) are not well separated. Similarly, an inverse iteration

$$(A - \mu B)x_{k+1} = Bx_k\tau_k, \quad (2)$$

with a shift μ that approximates λ_1 , converges linearly to v_1 from almost all initial conditions. A higher rate of convergence can be obtained using a feedback-like process that makes the shift depend on the current iterate. When the shift is chosen as the Rayleigh quotient

$$\mathbb{R}_0^n \rightarrow \mathbb{R} : y \mapsto \frac{y^T A y}{y^T B y}, \quad (3)$$

where \mathbb{R}_0^n denotes \mathbb{R}^n without the origin, evaluated at x_k , then a cubic rate of convergence is obtained, but global convergence is lost in the sense that the iteration converges to the “nearest” eigenvector; we refer e.g. to [Par80, BS89, ASVM04] for more details. If n is large, then only an approximate solution of (2) is sought, and the key question is to determine how crudely the solution can be approximated without tampering (too much) with the convergence of the exact iteration; for recent advances, see [SP99, GY00, SE02, vdE02, Not03, KN03].

It is natural to think of combining the individual advantages of these simple methods and obtain an iteration for which iterates are cheap to compute, convergence holds globally and the rate of convergence is superlinear. There is evidence that such a method can come from an optimization approach; indeed, for the problem of finding a minimum of a smooth cost function on the Euclidean space, the trust-region scheme proposed by Steihaug [Ste83] and Toint [Toi81], where the trust-region subproblems are approximately solved using a truncated CG inner iteration, possesses a similar combination of advantages.

It is well known (see for example [ST00]) that the leftmost and rightmost eigenvectors of (A, B) can be expressed as minimizers and maximizers of the Rayleigh quotient (3)—which thus plays the role of a cost function. More precisely, assuming that $\lambda_1 < \lambda_2$ and $\lambda_{n-1} < \lambda_n$, one has

$$\frac{v_1^T A v_1}{v_1^T B v_1} < \frac{y^T A y}{y^T B y} < \frac{v_n^T A v_n}{v_n^T B v_n}$$

for all y that are collinear with neither v_1 nor v_n . The difficulty is that the optimizers of (3) are not isolated: all the points αv_1 , $\alpha \in \mathbb{R}_0$, are minimizers, and all the points αv_n , $\alpha \in \mathbb{R}_0$, are maximizers. This is a cause of major difficulties of practical and theoretical nature; for example, applying the Newton method to the Rayleigh quotient (3) in \mathbb{R}^n yields the iteration mapping $y \mapsto 2y$, from which no information can be drawn. A remedy to this difficulty is to impose some normalization condition on y that picks typically one or two allowed points in each (or almost each) line $\{\alpha y : \alpha \in \mathbb{R}^n\}$. This was recognized in the early work of Bradbury and Fletcher [BF66] where several normalization conditions were considered (such as $\|y\|_1 = 1$, $\|y\|_2 = 1$ and $\|y\|_\infty = 1$) and a nonlinear conjugate-gradient optimization approach was proposed. For the generalized eigenproblem, we use the normalization $\|y\|_B = 1$, where $\|y\|_B := \sqrt{y^T B y}$; this particular normalization yields simplifications in the forthcoming developments. The optimization problem is thus to minimize or maximize the cost function

$$f : \{y \in \mathbb{R}^n : y^T B y = 1\} \rightarrow \mathbb{R} : y \mapsto \frac{y^T A y}{y^T B y}. \quad (4)$$

The minimizers are $\pm v_1$ and the maximizers are $\pm v_n$, i.e., the eigenvectors of (A, B) associated with the extreme eigenvalues. The remaining issue is to adapt the classical (i.e.,

in \mathbb{R}^n) Steihaug approach to the constrained minimization of f . This task was carried out in [ABG04b, ABG05] for the more general situation of a smooth cost function defined on a Riemannian manifold.

The purpose of this paper is to apply the general Riemannian trust-region algorithm of [ABG04b, ABG05] to the minimization of the Rayleigh quotient cost function (4). This yields a numerical algorithm that automatically retains the good global and local convergence properties of the general scheme. In particular, the property of convergence to stationary points of the cost function for all initial conditions translates into convergence to eigenspaces, since the stationary points of (4) are the eigenvectors of (A, B) . Moreover, the instability of the saddle points and local maxima turns into instability of all but the leftmost eigenvector $\pm v_1$. Furthermore, similar to the classical truncated-CG-based trust-region, the Riemannian algorithm of [ABG04b, ABG05], with a suitably-chosen stopping criterion, converges super-linearly to local minimizers of the cost function; this means that the proposed algorithm converges locally superlinearly to the leftmost eigenvector $\pm v_1$. The precise statements on convergence are given in Theorem 3.1. We will also see that since the algorithm is based on CG, it only requires a routine that returns Ax and Bx given x (the algorithm is thus “matrix-free”), along with storage space for a few n -vectors and a few scalars. Therefore, the method is particularly relevant for very large-scale problems.

Since the algorithm does not assume positive definiteness of A , it can also be applied to $(-A, B)$ and compute the rightmost eigenpair of (A, B) with the same convergence properties. It is also possible to compute a few extreme eigenvectors by using a block version of the algorithm [ABG04a] or by relying on deflation techniques [Par80].

Of course, with $B = I$ the generalized eigenproblem reduces to the standard eigenproblem. However, in contrast to many methods that tackle the generalized eigenproblem by reducing it to a standard one, the proposed method deals naturally with the generalized eigenproblem; therefore, there is no interest in considering the case $B = I$ separately.

We point out that the link with the deflation-accelerated CG (DACG) algorithm of [GSF92, BGP97] is not as strong as it may seem. The DACG method minimizes the Rayleigh quotient using a nonlinear CG method, whereas the proposed algorithm uses a truncated linear CG as an inner iteration for approximately minimizing a model of the Rayleigh quotient within a trust-region. In this respect, the proposed algorithm is more closely related to inexact Newton methods. The inexact scheme not only reduces the computational load while preserving superlinear convergence, but it also yields excellent global convergence properties that the exact Newton does not possess.

The proposed algorithm does have close connections with some existing eigenvalue algorithms. In particular, it has striking similarities with a variant of the Jacobi-Davidson method as analyzed by Notay [Not02]. The methods, however, differ on important points (see Section 4.1). The proposed method also relates to the Trace Minimization method of Sameh and Wisniewski [SW82, ST00]. We believe that the model trust region concept introduced in the proposed algorithm can be combined with other existing strategies to obtain even more efficient eigenvalue algorithms; in this respect, see [ABGS05] for a combination of the proposed algorithm and Basic Tracemin within the framework of adaptive model-based methods.

The paper is organized as follows. The algorithm is derived in Section 2. Its convergence properties are studied in Section 3. Some connections with other eigenvalue methods are briefly described in Section 4. Promising numerical experiments are presented in Section 5. Conclusions are drawn in Section 6.

2 The Algorithm

The proposed method was initially derived from an algorithm for optimization on manifolds [ABG04b]. However, it can be presented with little if any reference to optimization and differential geometry, as discussed in this section. We return to the connection with the Riemannian Trust-Region method of [ABG04b] in Section 3 when we study the convergence properties of the algorithm.

Let (A, B) be a symmetric/positive-definite pencil, with (λ_1, v_1) the leftmost eigenpair. We consider the problem of computing the minimizer $\pm v_1$ of the Rayleigh quotient f (4) constrained to the set $\{y : y^T B y = 1\}$ by an iterative method evolving on $\{y : y^T B y = 1\}$. Throughout the discussion, y denotes the current iterate. Consider the function

$$\widehat{f}_y(s) := f\left(\frac{y+s}{\|y+s\|_B}\right) = \frac{(y+s)^T A (y+s)}{(y+s)^T B (y+s)}, \quad y^T B s = 0, \quad (5)$$

where s can be thought of as a correction to the current iterate y . The condition $y^T B s = 0$ imposes that s be tangent to the set $\{y : y^T B y = 1\}$. Denoting by

$$P_y = I - B y (y^T B^2 y)^{-1} y^T B \quad (6)$$

the orthogonal projector onto $\{s : y^T B s = 0\}$ and denoting by $\langle u, v \rangle = u^T v$ the inner product on the Euclidean space \mathbb{R}^n , one has

$$\begin{aligned} \widehat{f}_y(s) &= \frac{y^T A y}{y^T B y} + 2 \frac{y^T A s}{y^T B y} + \frac{1}{y^T B y} \left(s^T A s - \frac{y^T A y}{y^T B y} s^T B s \right) + O(\|s\|^3) \\ &= f(y) + \langle 2P_y A y, s \rangle + \frac{1}{2} \langle 2P_y (A - f(y)B) P_y s, s \rangle + O(\|s\|^3). \end{aligned} \quad (7)$$

Define the quadratic model

$$m_y(s) = f(y) + \langle 2P_y A y, s \rangle + \frac{1}{2} \langle \mathcal{H}_y s, s \rangle, \quad y^T B s = 0, \quad (8)$$

where $\mathcal{H}_y : \{s : y^T B s = 0\} \rightarrow \{s : y^T B s = 0\}$ is some symmetric operator. It follows from (7) that the model m_y agrees with \widehat{f}_y to order one. A popular choice is the form

$$\mathcal{H}_y = 2P_y (A - \sigma B) P_y, \quad (9)$$

where σ is a scalar. When the operator \mathcal{H}_y is chosen as the Hessian

$$\text{Hess}_y : \{s : y^T B s = 0\} \rightarrow \{s : y^T B s = 0\} : s \mapsto 2P_y (A - f(y)B) P_y s, \quad (10)$$

the model (8) becomes the *Newton model*

$$m_y^N(s) = f(y) + \langle 2P_y A y, s \rangle + \frac{1}{2} \langle 2P_y (A - f(y)B) P_y s, s \rangle, \quad y^T B s = 0, \quad (11)$$

which agrees with \widehat{f}_y to order two. We will see in the convergence analysis (Section 3) that global convergence holds with few assumptions on \mathcal{H} , while the Newton model—rather, a sufficiently accurate approximation thereof—is required for superlinear convergence to hold.

Assuming that \mathcal{H}_y is invertible, the model $m_y(s)$ admits one and only one stationary point s_* . In particular, if the Hessian operator (10) is invertible, then the Newton model $m_y^N(s)$ admits one and only one stationary point s_* , solution of

$$P_y A y + P_y (A - f(y) B) P_y s = 0, \quad y^T B s = 0. \quad (12)$$

Depending on whether the Hessian operator (10) is positive definite, negative definite, or neither, the stationary point s_* is a minimum, maximum, or saddle point of the Newton model $m_y^N(s)$, respectively. The “pure” Newton approach [Smi94] consists in computing the update s_* and warping this update back onto the manifold, for example as $y_+ = (y + s_*) / \|y + s_*\|_B$. This development is also presented in [WSS98] as an application of Tapia’s algorithm for constrained optimization [Tap74], and it is closely related to the rationale in [SW82, ST00] (with an fundamental difference explained in Section 4). It is also well known [Shu86, AMSV02] that this pure Newton method is equivalent to the Rayleigh quotient iteration, whose convergence behaviour is well understood [BS89]. The pure Newton approach, however, is limited by two difficulties. First, while our objective is to minimize the Rayleigh quotient (4), it is not guaranteed that the Newton iteration will converge to a minimizer; depending on the initial condition, it may converge to a saddle point (interior eigenvector) or a maximizer (rightmost eigenvector). Second, when the iterate is far away from the solution, solving the Newton equation (12) accurately is a waste of computational effort. Therefore, the Newton equation is usually solved approximately using iterative solvers. The approximate solution, however, has to be sufficiently accurate for the (superlinear) convergence of the pure algorithm to be preserved; recent related work include [SP99, GY00, SE02, vdE02, Not03, KN03].

This paper innovates by proposing an inner iteration scheme that addresses these two difficulties. The inner iteration, which directly stems from the truncated-CG trust-region method of Steihaug [Ste83], proceeds as a classical CG enhanced with a dedicated stopping criterion.

Steihaug’s approach relies on the following observations. Consider the quadratic model $m_y(s)$ given in (8) and assume for a moment that the chosen operator \mathcal{H}_y is positive-definite. Recall that CG (which can be viewed as an optimization algorithm for the quadratic model m_y [GV96]) builds a sequence $\{s_j\}$ of approximate minimizers of m_y , a sequence $\{d_j\}$ of search directions and a sequence $\{r_j\}$ of residuals. These search directions d_j are descent directions for the quadratic model $m_y(s)$ at s_j . The inner iterate s_{j+1} is the minimizer of $m_y(s)$ along the line $s_j + \alpha d_j$, hence $m_y(s_{j+1}) \leq m_y(s_j)$. Finally, $\|s_{j+1}\| > \|s_j\|$ where $\|\cdot\|$ denotes the standard 2-norm.

Steihaug proposes three termination rules which work along the following lines.

- (i) The purpose of the model $m_y(s)$ is to approximate $\hat{f}_y(s)$ by a simpler function. As such, when $\|s\|$ gets large, the model loses its ability to closely match $\hat{f}_y(s)$. Therefore, the CG process is terminated when it crosses the boundary of the *trust-region* $\{s : \|s\| \leq \Delta\}$, where Δ is the trust-region radius inherited from the previous outer iteration step, and the point $s = s_j + \tau d_j$, with $\tau > 0$ and $\|s\| = \Delta$, is returned.
- (ii) The operator \mathcal{H}_y may not be positive-definite. In particular, the Hessian (10) is positive-definite only when the current iterate y is sufficiently close to the minimizers $\pm v_1$. Consequently, it may happen that a search direction d_j is a direction of nonpositive curvature for the model $m_y(s)$, namely, $d_j^T \mathcal{H}_y d_j \leq 0$; then the minimizer of m_y along the direction d_j is at infinity. This case is considered separately in the iteration before α is computed, and the

point $s = s_j + \tau d_j$, with $\tau > 0$ and $\|s\| = \Delta$, is returned.

(iii) Finally, the CG process is terminated when $\|r_i\|/\|r_0\| \leq \xi$ for some ξ . With a view on preserving the superlinear convergence of the exact algorithm, following [CGT00, §7.5.1], we propose instead a stopping criterion of the form

$$\|r_j\| \leq \|r_0\| \min\left(\|r_0\|^\theta, \kappa\right) \quad (13)$$

for some positive constants θ and κ .

According to these termination criteria, the truncated CG process returns with an approximate minimizer \tilde{s} of $m_y(s)$ constrained to the trust-region $\{s : \|s\| \leq \Delta\}$. A complete algorithm is obtained by embedding the inner process in a trust-region framework. The decision to accept or not the update \tilde{s} and to modify the trust-region radius is based on the quotient

$$\rho = \frac{\widehat{f}_y(0) - \widehat{f}_y(\tilde{s})}{m_y(0) - m_y(\tilde{s})} \quad (14)$$

which compares the decrease predicted by the model with the decrease actually observed on \widehat{f}_y . If ρ is very small, then the model at \tilde{s} is a very inaccurate approximation of $\widehat{f}_y(\tilde{s})$: the step is rejected and the trust-region radius is reduced. If ρ is small but less dramatically so, then the step is accepted but the trust-region radius is reduced. If ρ is close to 1, then there is a good agreement between the model and the function over the step, and the trust-region radius can be expanded.

These considerations yield the following method, which is the numerical algorithm obtained when applying the Riemannian trust-region scheme presented and analyzed in [ABG04b, ABG05] to the Rayleigh quotient cost function (4), using the *retraction* (which defines how the manifold is locally unwarped onto the tangent space at the current iterate) given by $R_y(s) = (y + s)/\|y + s\|_B$.

Algorithm 1 (outer iteration – trust-region)

Data: symmetric $n \times n$ matrices A and B , with B positive definite.

Parameters: $\bar{\Delta} > 0$, $\Delta_0 \in (0, \bar{\Delta})$, and $\rho' \in (0, \frac{1}{4})$.

Input: initial iterate $x_0 \in \{y : y^T B y = 1\}$.

Output: sequence of iterates $\{x_k\}$ in $\{y : y^T B y = 1\}$.

for $k = 0, 1, 2, \dots$

- Obtain s_k using the Steihaug-Toint truncated conjugate-gradient method (Algorithm 2) to approximately solve the trust-region subproblem

$$\min_{x_k^T B s = 0} m_{x_k}(s) \quad \text{s.t. } \|s\| \leq \Delta_k, \quad (15)$$

where m is defined in (8).

- Evaluate

$$\rho_k = \frac{\widehat{f}_{x_k}(0) - \widehat{f}_{x_k}(s_k)}{m_{x_k}(0) - m_{x_k}(s_k)} \quad (16)$$

where \widehat{f} is defined in (5).

- Update the trust-region radius:
 - if** $\rho_k < \frac{1}{4}$
 $\Delta_{k+1} = \frac{1}{4}\Delta_k$
 - else if** $\rho_k > \frac{3}{4}$ **and** $\|s_k\| = \Delta_k$
 $\Delta_{k+1} = \min(2\Delta_k, \bar{\Delta})$
 - else**
 $\Delta_{k+1} = \Delta_k$.

- Update the iterate:
 - if** $\rho_k > \rho'$

$$x_{k+1} = (x_k + s_k) / \|x_k + s_k\|_B \quad (17)$$

- else**
 $x_{k+1} = x_k$.

end (for)

Algorithm 2 (inner iteration – truncated CG)

Set $s_0 = 0$, $r_0 = P_{x_k} A x_k = A x_k - B x_k (x_k^T B^2 x_k)^{-1} x_k^T B A x_k$, $\delta_0 = -r_0$;

for $j = 0, 1, 2, \dots$ *until a stopping criterion (13) is satisfied, perform the following operations, where $\langle \cdot, \cdot \rangle$ denotes the standard inner product and \mathcal{H}_{x_k} denotes the operator chosen in the model (8).*

- if** $\langle \delta_j, \mathcal{H}_{x_k} \delta_j \rangle \leq 0$

Compute τ such that $s = s_j + \tau \delta_j$ minimizes $m(s)$ in (8) and satisfies $\|s\| = \Delta$;

return s ;

Set $\alpha_j = \langle r_j, r_j \rangle / \langle \delta_j, \mathcal{H}_{x_k} \delta_j \rangle$;

Set $s_{j+1} = s_j + \alpha_j \delta_j$;

- if** $\|s_{j+1}\| \geq \Delta$

Compute $\tau \geq 0$ such that $s = s_j + \tau \delta_j$ satisfies $\|s\| = \Delta$;

return s ;

Set $r_{j+1} = r_j + \alpha_j \mathcal{H}_{x_k} \delta_j$;

Set $\beta_{j+1} = \langle r_{j+1}, r_{j+1} \rangle / \langle r_j, r_j \rangle$;

Set $\delta_{j+1} = -r_{j+1} + \beta_{j+1} \delta_j$;

end (for).

Finally, we mention that, as a CG process, the inner iteration (Algorithm 2) lends itself nicely to preconditioning. Assume that an approximation $P_y K P_y$ is available for the operator \mathcal{H}_y , where K is such that systems $K\tilde{x} = x$ can be solved efficiently. The use of $P_y K P_y$ as a preconditioner is made possible by the following result due to Olsen *et al.* [OJS90] (or see [SvdVM98]): If $y^T B K^{-1} B y \neq 0$ and $y^T B v = 0$, then the solution u of $P_y K P_y u = v$ subject to $y^T B u = 0$ is given by $u = (I - \tilde{y}(y^T B \tilde{y})^{-1} y^T B) \tilde{v}$, where \tilde{y} and \tilde{v} are solutions of $K \tilde{y} = B y$ and $K \tilde{v} = v$. If the preconditioner $P_y K P_y$ is used in the inner CG process, then the property that the length of the update vector monotonically increases becomes true in the K norm, i.e., $\|s_{j+1}\|_K > \|s_j\|_K$ (see [Ste83]). In order to preserve the property that the inner iterates never re-enter the trust-region, the trust-region is then defined as $\{s : \|s\|_K \leq \Delta\}$.

Note that some papers [SS98, vdE02] refer to preconditioning as replacing the Hessian in the correction equation (12) by some approximation. This is not what is meant here: for example, without stopping criteria, and assuming that the Hessian operator (10) is positive definite, the preconditioned CG for the Newton model (11) would compute—in exact

arithmetic—the *exact* solution of the Newton equation (12) in a finite number of steps. However, both approaches—solving exactly an inexact Newton equation (quasi-Newton approach) or solving approximately the exact Newton equation (inexact Newton approach)—are closely related [Căt05].

3 Convergence analysis

The global and local convergence properties of trust-region schemes, including the truncated CG variant of Steihaug and Toint, have been studied thoroughly in the literature; see [CGT00, NW99] and references therein. However, the method proposed in the previous section differs from a classical trust-region algorithm in order to accommodate the fact that the optimization problem is not defined on the Euclidean space but on the non-Euclidean set $\{y : y^T B y = 1\}$. In particular, the “unwarped” cost function $\widehat{f}_y(s)$ depends on the current iterate y , and the update defined by (17) is different from the classical additive update.

Fortunately, the proposed method is an application of the general Riemannian trust-region algorithm [ABG04b], whose convergence was studied in [ABG04a, ABG05]: classical assumptions were rewritten in a way that makes sense on manifolds and it was proven that the convergence results of the classical trust-region schemes are preserved, *mutatis mutandis*. This yields the following statement for the proposed algorithm.

Theorem 3.1 *Let (A, B) be an $n \times n$ symmetric/positive-definite matrix pencil with eigenvalues $\lambda_1 < \lambda_2 \leq \dots \leq \lambda_{n-1} \leq \lambda_n$ and an associated B -orthonormal basis of eigenvectors (v_1, \dots, v_n) . Let $\mathcal{S}_i = \{y : Ay = \lambda_i B y, y^T B y = 1\}$ denote the intersection of the eigenspace of (A, B) associated to λ_i with the set $\{y : y^T B y = 1\}$. Assume that the operator \mathcal{H}_y chosen in (8) is bounded over $\{y : y^T B y = 1\}$.*

(i) *Let $\{x_k\}$ be a sequence of iterates generated by Algorithm 1. Then $\{x_k\}$ converges to the eigenspace of (A, B) associated to one of its eigenvalues. That is, there exists i such that $\lim_{k \rightarrow \infty} \text{dist}(x_k, \mathcal{S}_i) = 0$.*

(ii) *Only the set $\mathcal{S}_1 = \pm v_1$ is stable. More precisely, given $i \in \{2, \dots, n\}$ and $\epsilon > 0$, there exists $x_0, \|x_0\|_B = 1$, with $\text{dist}(x_0, \mathcal{S}_i) < \epsilon$ such that the sequence $\{x_k\}$ generated by Algorithm 1 from the initial condition x_0 , converges to an \mathcal{S}_j with $\lambda_j < \lambda_i$.*

(iii) *Suppose moreover that the choice of \mathcal{H}_y in (8) is such that $\|\mathcal{H}_y - 2P_y(A - f(y)B)P_y\| \leq \beta_2 \|P_y A y\|$ for some constant β_2 , where f is the Rayleigh quotient (4) and P_y is the projector (6). Then there exists $c > 0$ such that, for all sequences $\{x_k\}$ generated by Algorithm 1 converging to \mathcal{S}_1 , there exists $K > 0$ such that for all $k > K$,*

$$\text{dist}(x_{k+1}, \mathcal{S}_1) \leq c (\text{dist}(x_k, \mathcal{S}_1))^{\min\{\theta+1, 2\}} \quad (18)$$

with $\theta > 0$ as in (13).

Proof. (i) Algorithm 1 is the RTR-tCG algorithm of [ABG05] applied to the Rayleigh quotient cost function (4) on the manifold $\{y : y^T B y = 1\}$ equipped with the retraction $R_y(s) = (y+s)/\|y+s\|_B$. The cost function and the retraction are smooth, the manifold is compact, the truncated CG scheme satisfies the Cauchy decrease condition, and the parameter ρ' belongs to $(0, \frac{1}{4})$. Therefore all the assumptions of [ABG05, Th. 4.4] are satisfied. Consequently, the gradient of f converges to zero, that is, $P_{x_k} A x_k \rightarrow 0$. This means that every limit point of $\{x_k\}$ is an eigenvector of (A, B) . Moreover, since the Rayleigh quotient f given in (4)

is nonincreasing over the whole sequence $\{x_k\}$, it follows that all the limit points have the same value of f . Since f evaluated at an eigenvector returns the corresponding eigenvalue, it follows that all the limit points of $\{x_k\}$ are eigenvectors of (A, B) corresponding to the same eigenvalue; that is, all limit points belong to some \mathcal{S}_i . The fact that the sequence $\{x_k\}$ converges to its limit set follows from boundedness of the manifold by a classical contradiction argument. Indeed, suppose that this is not the case; then there is an $\epsilon > 0$ and an infinite subsequence $\{x_{k_j}\}$ such that $\text{dist}(x_{k_j}, \mathcal{S}_i) > \epsilon$. Since the sequence x_{k_j} is bounded, it contains a convergent subsequence, whose limit point must belong to \mathcal{S}_i and at the same time be at a distance at least ϵ from \mathcal{S}_i , a contradiction.

(ii) It is well known (see, e.g., [ST00, AMSV02]) that the eigenvectors related to $\lambda_2, \dots, \lambda_n$ are saddle points or maxima of the Rayleigh quotient f and that the eigenvector related to λ_1 is a minimum of f . Since f is nonincreasing over the sequences generated by the algorithm, the result follows from (i).

(iii) The cost function and the retraction are smooth; the manifold is compact; $2P_yAy$ is the gradient of f at y ; $2P_y(A - f(y)B)P_y$ is the Hessian of \hat{f}_y at the origin; the assumption $\lambda_1 < \lambda_2$ ensures that $\pm v_1$ is a nondegenerate local minimum of f (the Hessian is positive definite). Consequently, all the assumptions of [ABG05, Th. 4.13] hold and the result follows. \square

Strictly speaking, $\text{dist}(u, v)$ denotes the geodesic distance on $\{y : y^T B y = 1\}$ between two points u and v , which is the length of the shortest curve on $\{y : y^T B y = 1\}$ that joins u and v . However, this distance is asymptotically equivalent to the more classical Euclidean distance $\|u - v\|$ in the embedding space \mathbb{R}^n . That is, for all u with $\|u\|_B = 1$, there exist constants c_1, c_2 and ϵ such that, for all v that satisfies $\|v\|_B = 1$ and $\|v - u\| < \epsilon$, one has $c_1\|v - u\| \leq \text{dist}(u, v) \leq c_2\|v - u\|$. Since all the statements involving “dist” in the convergence results are asymptotic, all the occurrences of dist can be replaced by the Euclidean distance.

4 Links with other methods

Not surprisingly, the proposed method relates to several Newton, CG or Krylov eigenvalue methods [ABG04a]. It can be anticipated that the strong convergence results presented in Section 3 will help understand the workings of several of these methods. In this section, we briefly consider the case of two well-known and successful methods whose workings are still the object of investigation in the literature.

4.1 Jacobi-Davidson

The proposed algorithm relates to the Simplified Jacobi-Davidson (JD) analyzed by Notay in [Not02] for the case $B = I$. In Simplified JD, the next iterate is obtained by adding to the current iterate y the computed approximate solution to the correction equation

$$(I - yy^T)(A - \sigma I)(I - yy^T)s = -(I - yy^T)Ay, \quad y^T s = 0, \quad (19)$$

where the shift σ is selected either as the Rayleigh quotient $\theta = y^T A y$ or as some ‘target’ τ less than the smallest eigenvalue λ_1 . Simplified JD is thus a JD method [SV96] without subspace acceleration. The approximate solution to the correction equation (19) is obtained using a CG iteration with a specific termination strategy [Not02, §4].

Simplified JD and the proposed algorithm are closely related. For the choice $\sigma = \theta$ and $B = I$, the exact solution to the correction equation (19) is equal to the stationary point of the Newton model (11), since (12) and (19) are identical. Under these assumptions, from a given iterate y , the inner iteration of both algorithms start generating identical sequences, because they both rely on the same CG process. The two approaches, however, differ on the use of σ and on the stopping strategy for the inner CG iteration, and this has important theoretical and practical consequences that we now describe.

The proposed algorithm terminates using a trust-region strategy combined with a residual-based condition (13). The trust-region strategy yields the global convergence properties of Theorem 3.1-(i,ii), which does not seem to have equivalents in the existing JD-related literature. The trust-region also helps avoid a waste of computational effort when the eigenvector approximations are not yet accurate. The residual-based stopping condition (13) comes directly from general concepts without any effort to adapt it to the Rayleigh quotient minimization. In contrast, the stopping criterion in [Not02] relies on a careful analysis of the relation between the norm of the residuals in the eigenvector approximations and in the residuals in the CG iterates [Not02, (12)]. However, we found in preliminary numerical experiments that the stopping criterion (13) performs better than the stopping criterion [Not02, (27-28)]. A possible explanation is that the inner CG process strives to reduce to zero the *model* residual involved in (13), and not the residual of the actual cost function computed by [Not02, (12)]. A combination of criteria based on [Not02, (12)] and on (13) may yield better results. This clearly deserves further investigation.

The other difference concerns the use of σ . The classical CG breaks down in the presence of an indefinite Hessian, and the possibility of choosing $\sigma = \tau$ provides a way out by forcing the operator in the correction equation (19) to be positive definite. In contrast, indefiniteness of the Hessian is naturally taken care of in the Steihaug trust-region CG: when a direction of negative curvature is encountered, the inner iteration hits the trust-region boundary and returns the obtained point. Therefore, the proposed algorithm can use the shift $\sigma = \theta$ throughout, which corresponds to using the Newton model given in (11). This does not mean that the trust-region approach cannot benefit from using other models. Indeed, the global convergence theory (Section 3) holds regardless of the choice of the quadratic term in the model m ; as a consequence, global convergence holds for any value of σ in (9). For local superlinear convergence to occur, however, the model Hessian \mathcal{H}_y has to be a sufficiently good approximation of the exact Hessian (see Theorem 3.1-(iii)); it is thus a good idea to select the model Hessian as the exact Hessian (10) (that is, $\sigma = \theta$) when the iteration approaches the solution. This points to an adaptive model strategy that was recently investigated in [ABGS05].

We now return to the subspace acceleration technique that was left out in Simplified JD. Much as the complete JD (i.e., with subspace acceleration) is faster in general than Simplified JD, the proposed algorithm is generally faster with a subspace acceleration enhancement. A difference is that the proposed algorithm converges globally (Theorem 3.1) without subspace acceleration, while Simplified JD benefits from subspace acceleration for global convergence purposes: selecting the leftmost Ritz pair from the acceleration subspace favors convergence to the leftmost eigenpair and makes the choice of the shift σ less critical, although no global convergence result in the form of Theorem 3.1-(i,ii) seems to be available yet.

We conclude with a remark concerning the projector (6) appearing in the model (8) and in the Hessian operator (10). This is an orthogonal projector, in contrast to the oblique projector on which JD for $B \neq I$ usually relies (see e.g. [SvdV00]). The choice of an orthogonal projection comes from geometric considerations: by definition, the gradient $P_y A y$ appearing

in (8) must belong to the tangent space to the manifold, and the Hessian operator (10) must be from the tangent space into itself. Consequently, the projector (6) must be into the tangent space of $\{y : y^T B y = 1\}$, which is $(By)^\perp$. This has favorable consequences: orthogonal projectors are numerically safer, and the resulting operator $2P_y(A - f(y)B)P_y$ maps the space $(By)^\perp$ to itself, which is not the case in the usual JD formulation, thus making the use of a preconditioner unavoidable if a Krylov solver is used.

4.2 Tracemin

Sameh and Wisniewski [SW82] and Sameh and Tong [ST00] proposed and analyzed a trace minimization (Tracemin) algorithm for computing a few (p) minor eigenpairs of a symmetric positive definite matrix pencil (A, B) . For simplicity, we consider the algorithm for the case $p = 1$; block versions of Algorithm 1-2 and the Tracemin algorithm will be considered elsewhere. The basic Tracemin method can be viewed as using a particular choice of the model (8):

$$\begin{aligned} m_y^{\text{TM}}(s) &= y^T A y + 2y^T A s + s^T A s \\ &= y^T A y + \langle 2P A y, s \rangle + \frac{1}{2} \langle 2P A P s, s \rangle, \quad y^T B s = 0. \end{aligned} \tag{20}$$

Comparing with the Newton model (11), we see that there is a “missing term” in the second-order part. This indicates why the simple Tracemin method does not reach superlinear convergence. On the other hand, assuming that A is also positive definite, the model $m_y^{\text{TM}}(s)$ has an interesting beneficial feature: the exact minimizer s_\star of (20) satisfies $\widehat{f}_y(s_\star) \leq \widehat{f}_y(0)$, i.e.,

$$\frac{(y + s_\star)^T}{\|y + s_\star\|_B} A \frac{(y + s_\star)}{\|y + s_\star\|_B} \leq \frac{y^T}{\|y\|_B} A \frac{y}{\|y\|_B},$$

and moreover, if CG is used to compute s_\star , then the above inequality is satisfied by all intermediate iterates of the CG process [ST00, Lemma 3.2]. Therefore, the basic Tracemin method is in fact a descent method for the Rayleigh quotient that is robust with respect to inexact solves.

To improve the speed of convergence of the iteration, Sameh and Wisniewski [SW82] and Sameh and Tong [ST00] proposed a dynamic shift technique that appears to be effective in practice but whose workings are not yet rigorously understood. The results of this paper may shed some light on this issue, since the “missing term” in (20) is simply a Rayleigh quotient shift.

5 Numerical experiments

In this section, we report on preliminary numerical experiments that show the strong potential of Algorithm 1-2 as a competitive method for computing extreme eigenpairs of symmetric/positive-definite matrix pencils.

The first set of experiments was conducted to illustrate the convergence properties presented in Section 3. The matrices A and B were chosen from random distributions and the initial condition x_0 was chosen from a normal distribution and B -normalized. More than 10^4 such experiments were conducted and convergence to the leftmost eigenvector v_1 was systematically observed. The θ parameter in the inner stopping criterion (13) was set to $\theta = 1.0$,

and the observed results were compatible with the (at least) quadratic convergence proven in Section 3. In fact, due to the symmetry of the problem, it can be argued that the rate of convergence is actually $\min\{1 + \theta, 3\}$, and this is supported by the numerical experiments. We refer to [ABG04a] for details.

A second set of experiments was conducted to compare Algorithm 1-2 with the matrix-free Krylov subspace method for the generalized eigenproblem proposed by Golub and Ye [GY02, Alg. 1] (referred to as the *GY method*). Note that the use of preconditioners is not considered here. These preliminary experiments were conducted on matrices of moderate size ($n = 100$); since the proposed algorithm is matrix-free, it is suitable for dealing with very-large-scale problems, but the influence of finite-precision arithmetic deserves further theoretical and numerical investigation.

In each experiment, a symmetric positive-definite matrix A was generated with specific eigenvalues. The symmetric positive definite matrix B was chosen as $B = SS^T + 1000I$, where S was a square matrix with elements chosen from a standard normal distribution. This choice allowed the eigenvalue distribution of the pencil to be essentially determined via A , while testing the ability of the method to operate on a non-trivial B . For each generated problem (A, B) , the proposed method was applied using the Newton model (11) and three different values of the θ parameter from criterion (13): $\theta = 0.5$, $\theta = 1.0$, and $\theta = 1.5$. The GY method was allowed to form a basis of size $m = 6$. This number was chosen so that both of the algorithms were allowed an equal amount of memory. The distance to the solution is measured by computing the angle between the current iterate and the leftmost eigenvector of the pencil.

Figure 1(a) shows the results of the first test, where the gap between the leftmost two eigenvalues is small ($\frac{\lambda_2(A,B) - \lambda_1(A,B)}{\lambda_{100}(A,B) - \lambda_1(A,B)} \approx .009$)(Figure 1(b)). The superlinear convergence of the proposed algorithm is clearly seen. Moreover, we see that in terms of the number of matrix-vector multiplications (which can be considered as a consistent measure of the computational cost of both algorithms), the proposed method outperforms the GY method, even for mild accuracy requirements.

Figure 1(c) shows the results of a second test, where the gap between the leftmost two eigenvalues was much larger ($\frac{\lambda_2(A,B) - \lambda_1(A,B)}{\lambda_{100}(A,B) - \lambda_1(A,B)} \approx .47$)(Figure 1(d)). The numerical performance, in term of matrix-vector multiplications, has improved for both algorithms. While the GY method experienced greater improvement in performance due to the larger gap, the proposed method performed comparably well.

Also note that while there is some variation in the performance of the proposed method for different values of the stopping parameter θ , the performance is not dramatically sensitive to this parameter. This is important, because it suggests that the choice of θ be more easily made than is often the case with parameter-based methods in the literature in order to provide adequate performance of the algorithm across varying matrices A and B .

Note that the GY method has been shown to yield faster convergence when a preconditioner is used; future experiments will consider the relative performance of the preconditioned GY method against a preconditioned version of the proposed method.

6 Conclusion and future work

We have proposed a “matrix-free” method for computing the leftmost eigenpair of a symmetric/positive-definite matrix pencil (A, B) . The algorithm stems from a method of optimization on Rie-

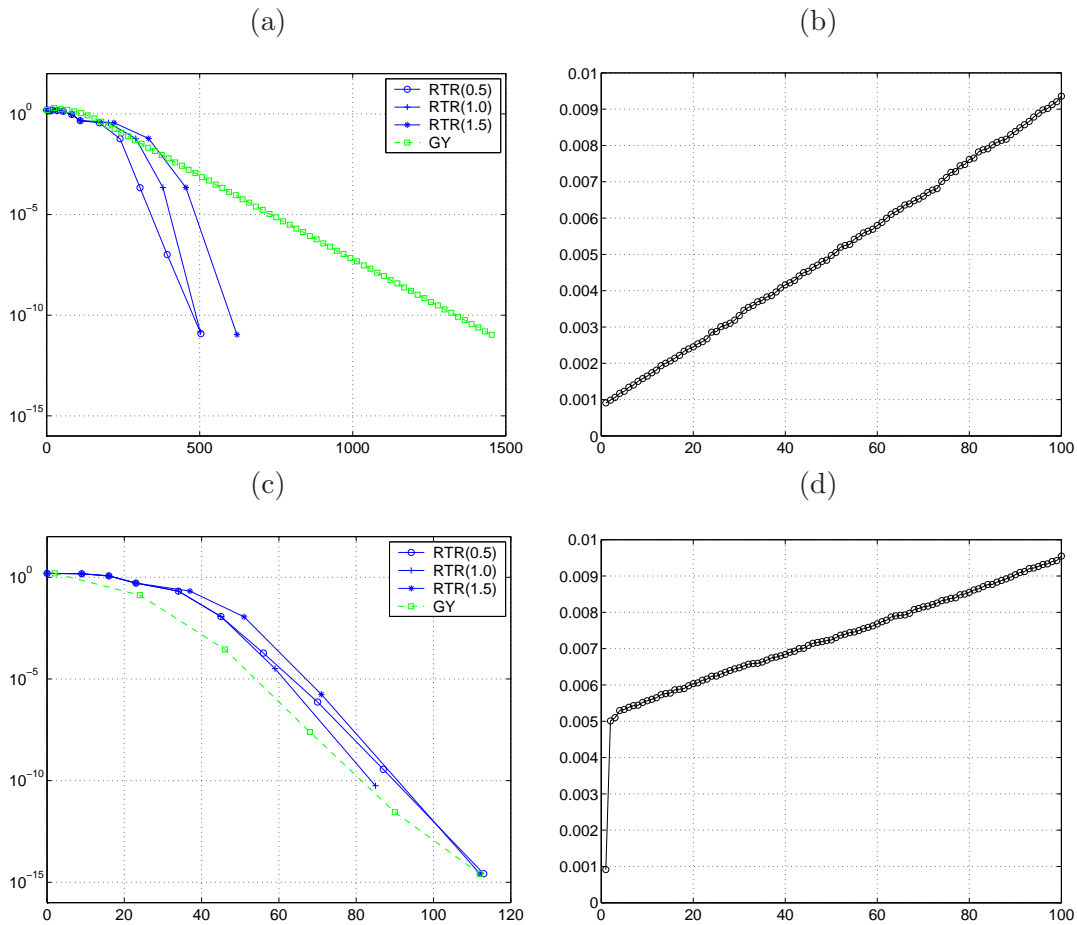


Figure 1: Numerical efficiency of Algorithm 1-2 (with $\theta = 0.5, 1.0$ and 1.5 in (13)) and the Krylov subspace method of [GY02, Alg. 1]. (a,c) plots the distance to the solution versus the number of matrix-vector products by A and B . (b,d) illustrates the spectrum of the pencil $A - \lambda B$.

mannian manifolds [ABG04b, ABG05], from which it inherits good and well-understood local and global convergence properties (Theorem 3.1). It employs a trust-region strategy where the trust-region subproblems are solved approximately using a truncated conjugate-gradient method. The algorithm can be applied to $(-A, B)$ to compute the eigenvector corresponding to the rightmost eigenvalue (A, B) . The algorithm relates to the Jacobi-Davidson method [FSvdV98] and the trace minimization method [SW82, ST00]. In particular, it is closely related to a variant of the Jacobi-Davidson method analyzed by Notay [Not02], from which it notably differs by the trust-region-based inner stopping criterion that avoids a waste of computational effort and yields global convergence properties. Numerical experiments show that the proposed method is able to outperform a recently-proposed [GY02] Krylov subspace method for the generalized eigenproblem.

The current form of the proposed method is simply a direct application of the Riemannian trust-region method of [ABG04b, ABG05] to the eigenproblem, but even in this simple form it demonstrates promising numerical results and sheds light on the behaviour of other well-

known methods. In an upcoming paper, we report on improvements to the method that take into account properties specific to the eigenproblem and compare the resulting algorithms with several other methods. We will also report on a block version of the algorithm obtained by applying the Riemannian trust-region method on the Grassmann manifold to the trace minimization problem associated with the symmetric generalized eigenvalue problem.

Acknowledgements

Thanks to Alan Edelman, Paul Van Dooren and Gerard Sleijpen for useful discussions. We also thank the anonymous referees for their insightful comments.

References

- [ABG04a] P.-A. Absil, C. G. Baker, and K. A. Gallivan, *Trust-region methods on Riemannian manifolds*, Tech. Report FSU-CSIT-04-13, School of Computational Science, Florida State University, July 2004.
- [ABG04b] ———, *Trust-region methods on Riemannian manifolds with applications in numerical linear algebra*, Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems (MTNS2004), Leuven, Belgium, 5–9 July 2004, 2004.
- [ABG05] ———, *Trust-region methods on Riemannian manifolds*, accessible from <http://www.inma.ucl.ac.be/~absil/>, submitted, March 2005.
- [ABGS05] P.-A. Absil, C. G. Baker, K. A. Gallivan, and A. Sameh, *Adaptive model trust region methods for generalized eigenvalue problems*, International Conference on Computational Science (Vaidy S. Sunderam, Geert Dick van Albada, and et al. Peter M. A. Slot, eds.), Lecture Notes in Computer Science, vol. 3514, Springer-Verlag, 2005, pp. 33–41.
- [AMSV02] P.-A. Absil, R. Mahony, R. Sepulchre, and P. Van Dooren, *A Grassmann-Rayleigh quotient iteration for computing invariant subspaces*, SIAM Rev. **44** (2002), no. 1, 57–73 (electronic).
- [ASVM04] P.-A. Absil, R. Sepulchre, P. Van Dooren, and R. Mahony, *Cubically convergent iterations for invariant subspace computation*, SIAM J. Matrix. Anal. Appl. **26** (2004), no. 1, 70–96.
- [BDDR00] Zhaojun Bai, James Demmel, Jack Dongarra, and Axel Ruhe (eds.), *Templates for the solution of algebraic eigenvalue problems*, Software, Environments, and Tools, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000, A practical guide.
- [BF66] W. W. Bradbury and R. Fletcher, *New iterative methods for solution of the eigenproblem*, Numer. Math. **9** (1966), 259–267.
- [BGP97] Luca Bergamaschi, Giuseppe Gambolati, and Giorgio Pini, *Asymptotic convergence of conjugate gradient methods for the partial symmetric eigenproblem*, Numer. Linear Algebra Appl. **4** (1997), no. 2, 69–84.
- [BS89] S. Batterson and J. Smillie, *The dynamics of Rayleigh quotient iteration*, SIAM J. Numer. Anal. **26** (1989), no. 3, 624–636.
- [Căt05] Emil Cătiņaș, *The inexact, inexact perturbed, and quasi-Newton methods are equivalent models*, Math. Comp. **74** (2005), no. 249, 291–301 (electronic).
- [CGT00] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Trust-region methods*, MPS/SIAM Series on Optimization, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, and Mathematical Programming Society (MPS), Philadelphia, PA, 2000.
- [FSvdV98] Diederik R. Fokkema, Gerard L. G. Sleijpen, and Henk A. van der Vorst, *Jacobi-Davidson style QR and QZ algorithms for the reduction of matrix pencils*, SIAM J. Sci. Comput. **20** (1998), no. 1, 94–125 (electronic).

- [GSF92] Giuseppe Gambolati, Flavio Sartoretto, and Paolo Florian, *An orthogonal accelerated deflation technique for large symmetric eigenproblems*, Comput. Methods Appl. Mech. Engrg. **94** (1992), no. 1, 13–23.
- [GV96] G. H. Golub and C. F. Van Loan, *Matrix computations, third edition*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, 1996.
- [GY00] Gene H. Golub and Qiang Ye, *Inexact inverse iteration for generalized eigenvalue problems*, BIT **40** (2000), no. 4, 671–684.
- [GY02] G. H. Golub and Q. Ye, *An inverse free preconditioned Krylov subspace method for symmetric generalized eigenvalue problems*, SIAM J. Sci. Comput. **24** (2002), no. 1, 312–334 (electronic).
- [KN03] Andrew V. Knyazev and Klaus Neymeyr, *A geometric theory for preconditioned inverse iteration. III. A short and sharp convergence estimate for generalized eigenvalue problems*, Linear Algebra Appl. **358** (2003), 95–114, Special issue on accurate solution of eigenvalue problems (Hagen, 2000).
- [Not02] Y. Notay, *Combination of Jacobi-Davidson and conjugate gradients for the partial symmetric eigenproblem*, Numer. Linear Algebra Appl. **9** (2002), no. 1, 21–44.
- [Not03] ———, *Convergence analysis of inexact Rayleigh quotient iteration*, SIAM J. Matrix Anal. Appl. **24** (2003), no. 1, 627–644.
- [NW99] J. Nocedal and S. J. Wright, *Numerical optimization*, Springer Series in Operations Research, Springer-Verlag, New York, 1999.
- [OJS90] J. Olsen, P. Jørgensen, and J. Simons, *Passing the one-billion limit in full configuration-interaction (FCI) calculations*, Chemical Physics Letters **169** (1990), 463–472.
- [Par80] B. N. Parlett, *The symmetric eigenvalue problem*, Prentice-Hall, Inc., Englewood Cliffs, N.J. 07632, 1980, republished by SIAM, Philadelphia, 1998.
- [Saa92] Y. Saad, *Numerical methods for large eigenvalue problems*, Manchester University Press, Manchester, 1992.
- [SE02] Valeria Simoncini and Lars Eldén, *Inexact Rayleigh quotient-type methods for eigenvalue computations*, BIT **42** (2002), no. 1, 159–182.
- [Shu86] M. Shub, *Some remarks on dynamical systems and numerical analysis*, Proc. VII ELAM. (L. Lara-Carrero and J. Lewowicz, eds.), Equinoccio, U. Simón Bolívar, Caracas, 1986, pp. 69–92.
- [Smi94] Steven T. Smith, *Optimization techniques on Riemannian manifolds*, Hamiltonian and gradient flows, algorithms and control, Fields Inst. Commun., vol. 3, Amer. Math. Soc., Providence, RI, 1994, pp. 113–136.
- [SP99] P. Smit and M. H. C. Paardekooper, *The effects of inexact solvers in algorithms for symmetric eigenvalue problems*, Linear Algebra Appl. **287** (1999), no. 1-3, 337–357, Special issue celebrating the 60th birthday of Ludwig Elsner.
- [SS98] Andreas Stathopoulos and Yousef Saad, *Restarting techniques for the (Jacobi-)Davidson symmetric eigenvalue methods*, Electron. Trans. Numer. Anal. **7** (1998), 163–181 (electronic), Large scale eigenvalue problems (Argonne, IL, 1997).
- [ST00] A. Sameh and Z. Tong, *The trace minimization method for the symmetric generalized eigenvalue problem*, J. Comput. Appl. Math. **123** (2000), 155–175.
- [Ste83] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal. **20** (1983), 626–637.
- [Ste01] G. W. Stewart, *Matrix algorithms, vol II: Eigensystems*, Society for Industrial and Applied Mathematics, Philadelphia, 2001.
- [SV96] G. L. G. Sleijpen and H. A. Van der Vorst, *A Jacobi-Davidson iteration method for linear eigenvalue problems*, SIAM J. Matrix Anal. Appl. **17** (1996), no. 2, 401–425.
- [SvdV00] G. Sleijpen and H. van der Vorst, *Jacobi-Davidson methods*, Templates for the Solution of Algebraic Eigenvalue Problems (Zhaojun Bai, James Demmel, Jack Dongarra, Axel Ruhe, and Henk van der Vorst, eds.), SIAM, Philadelphia, 2000, pp. 123–127.

- [SvdVM98] G. L. Sleijpen, H. A. van der Vorst, and E. Meijerink, *Efficient expansion of subspaces in the jacobi-davidson method for standard and generalized eigenproblems*, Electron. Trans. Numer. Anal. **7** (1998), 75–89.
- [SW82] A. H. Sameh and J. A. Wisniewski, *A trace minimization algorithm for the generalized eigenvalue problem*, SIAM J. Numer. Anal. **19** (1982), no. 6, 1243–1259.
- [Tap74] R. A. Tapia, *Newton’s method for optimization problems with equality constraints*, SIAM J. Numer. Anal. **11** (1974), 874–886.
- [Toi81] Ph. L. Toint, *Towards an efficient sparsity exploiting Newton method for minimization*, Sparse Matrices and Their Uses (I. S. Duff, ed.), Academic Press, London, 1981, pp. 57–88.
- [vdE02] Jasper van den Eshof, *The convergence of Jacobi-Davidson iterations for Hermitian eigenproblems*, Numer. Linear Algebra Appl. **9** (2002), no. 2, 163–179.
- [WSS98] Kesheng Wu, Yousef Saad, and Andreas Stathopoulos, *Inexact Newton preconditioning techniques for large symmetric eigenvalue problems*, Electron. Trans. Numer. Anal. **7** (1998), 202–214 (electronic), Large scale eigenvalue problems (Argonne, IL, 1997).