SIAM J. MATRIX ANAL. APPL.
Vol. 39, No. 2, pp. 611–631

A LANCZOS METHOD FOR LARGE-SCALE EXTREME LORENTZ EIGENVALUE PROBLEMS*

LEI-HONG ZHANG†, CHUNGEN SHEN‡, WEI HONG YANG§, AND JOAQUIM J. JÚDICE¶

Abstract. In this paper, we are concerned with an efficient algorithm for solving the extreme Lorentz eigenvalue problem (ELE). The Lorentz eigenvalue problem is an eigenvalue complementarity problem over the Lorentz cone, and solving ELE is equivalent to testing the Lorentz-copositivity for a given matrix. Treating ELE as a special eigenvalue problem, we propose a Lanczos-type method which mimics the Rayleigh–Ritz procedure and is suitable for large-scale and sparse problems. The numerical behavior and efficiency of the proposed method are supported by the theoretical convergence results and some preliminary numerical experiments.

Key words. eigenvalue problem, eigenvalue complementarity problem, copositivity, second-order cone problem

AMS subject classifications. 90C33, 47J20, 15A18, 90C06, 65F15

DOI. 10.1137/17M1111401

1. Introduction.

Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$, we consider the minimization problem

$$\min \left\{ q_A(\boldsymbol{x}) \triangleq \boldsymbol{x}^{\mathrm{T}} A \boldsymbol{x} \right\} \quad \text{s.t.} \quad \|\boldsymbol{x}\|_2 = 1,\ \boldsymbol{x} \in \mathbb{K}^n, \tag{1.1}$$

where

$$\mathbb{K}^n \triangleq \left\{ \boldsymbol{x} = \begin{bmatrix} \alpha \\ \boldsymbol{z} \end{bmatrix} \in \mathbb{R}^n \,:\, \|\boldsymbol{z}\|_2 \le \alpha \right\}.$$

Points in $\mathbb{K}^n$ form a cone, widely known as the *Lorentz cone*, the *second-order cone*, or the *ice-cream cone* in the literature. The minimization (1.1) is closely related to the notion of *Lorentz-copositivity* defined as follows.

DEFINITION 1.1 (see [25] and also [32, Definition 2.1]). *Let $A \in \mathbb{R}^{n\times n}$ be symmetric. We say A is Lorentz-copositive if and only if $q_A(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}} A \boldsymbol{x}$ takes only nonnegative values on $\boldsymbol{x} \in \mathbb{K}^n$.*

*Received by the editors January 12, 2017; accepted for publication (in revised form) by D. L. Boley January 17, 2018; published electronically April 5, 2018.
http://www.siam.org/journals/simax/39-2/M111140.html
Funding: The work of the first author was supported in part by the National Natural Science Foundation of China (NSFC-11671246, NSFC-91730303, NSFC-11371102), and the Basic Academic Discipline Program, the 11th five year plan of 211 Project for Shanghai University of Finance and Economics. The work of the second author was supported by the National Natural Science Foundation of China (11101281 and 11271259) and the Innovation Program of Shanghai Municipal Education Commission (12YZ172). The work of the third author was supported by the National Natural Science Foundation of China (NSFC-91730304, NSFC-11371102, NSFC-91330201). The work of the fourth author was in the scope of R&D Unit 50008, financed by the applicable financial framework (FCT/MEC through national funds and when applicable co-funded by FEDER PT2020 partnership agreement).

†School of Mathematics and Research School for Interdisciplinary Sciences, Shanghai University of Finance and Economics, Shanghai 200433, China (zhang.leihong@mail.shufe.edu.cn).
‡College of Science, University of Shanghai for Science and Technology, Shanghai 200093, China (shenchungen@gmail.com).
§School of Mathematical Sciences, Fudan University, Shanghai, 200433, People's Republic of China (whyang@fudan.edu.cn).
¶Instituto de Telecomunicações, Universidade de Coimbra - Polo II, 3030-290 Coimbra, Portugal (joaquim.judice@co.it.pt).

According to Definition 1.1, one can see that A is Lorentz-copositive whenever the global minimum, denoted as $\hat{q}$, of (1.1) is nonnegative.

Lorentz-copositivity is a kind of extension of classical *copositivity* [25, Definition 1.1], whose definition reads the same as Definition 1.1 except for replacing the Lorentz cone $\mathbb{K}^n$ by the nonnegative orthant $\mathbb{R}^n_+$. There has been a wealth of development, in both theory and implementation, of copositivity (see, e.g., [25] for a survey). It is known that there are a couple of equivalent statements for copositivity [25]; however, unfortunately, testing whether A is copositive is challenging and indeed a coNP-complete problem, meaning that testing whether A is not copositive is NP-complete [27]. On the other hand, by using the Lagrange multiplier theory, the KKT condition of minimizing $q_A(\boldsymbol{x})$ over $\mathbb{R}^n_+$ is the so-called *Pareto eigenvalue complementarity problem*, which consists of solving a pair $(\lambda, \boldsymbol{x})$ with $\lambda \in \mathbb{R}$ (also known as the *Pareto eigenvalue* [25, Definition 4.2]) and $\boldsymbol{x} \in \mathbb{R}^n \backslash \{\mathbf{0}\}$ (also known as the *Pareto eigenvector* [25, Definition 4.2]) satisfying

$$\mathbb{R}^n_+ \ni \boldsymbol{x} \perp A\boldsymbol{x} - \lambda \boldsymbol{x} \in \mathbb{R}^n_+ \quad \text{and} \quad \|\boldsymbol{x}\| = 1, \tag{1.2}$$

where $\|\cdot\|$ is some norm in $\mathbb{R}^n$. In practice, the ℓ_1-norm is used, which gives $\boldsymbol{e}^{\mathrm{T}}\boldsymbol{x} = 1$ with $\boldsymbol{e}$ a vector consisting of ones. The Pareto eigenvalue problem receives much interest in the optimization community, and there are plenty of research papers devoted to theoretical analysis, applications, and numerical algorithms [2, 6, 11, 12, 17, 19, 26, 29, 34, 44, 46, 52].

Analogously, the optimality conditions for (1.1) reads as

$$\mathbb{K}^n \ni \boldsymbol{x} \perp A\boldsymbol{x} - \lambda \boldsymbol{x} \in \mathbb{K}^n \quad \text{and} \quad \|\boldsymbol{x}\|_2 = 1 \tag{1.3}$$

for a pair $(\lambda, \boldsymbol{x})$ with $\lambda \in \mathbb{R}$. This is referred to as the *Lorentz eigenvalue problem* [19] or the *second-order cone eigenvalue complementarity problem* (SOCEiCP) [16]. A pair $(\lambda, \boldsymbol{x})$ satisfying (1.3) is called the *Lorentz eigenpair* with λ and $\boldsymbol{x}$ called a Lorentz eigenvalue and a Lorentz eigenvector, respectively. Algorithms for computing a Lorentz pair are discussed in [5] when A is symmetric and in [1, 7, 19] for the general case. These algorithms can also be applied to SOCEiCPs where $\mathbb{K}^n$ is the Cartesian product of multiple Lorentz cones.

It is clear that for a Lorentz eigenpair $(\lambda, \boldsymbol{x})$,

$$\lambda = \boldsymbol{x}^{\mathrm{T}} A \boldsymbol{x} = q_A(\boldsymbol{x}).$$

This implies that (1.1) is equivalent to finding a specific Lorentz eigenpair $(\lambda, \boldsymbol{x})$ so that λ achieves the minimum. This problem is called the extreme Lorentz eigenvalue problem (ELE), and its numerical solution is the main purpose of this paper. Indeed, we have the following.

PROPOSITION 1.1 (see [19, Proposition 4.1]). *A symmetric matrix A is Lorentz-copositive if and only if ELE has a nonnegative optimal value.*

It is important to note that the algorithms for SOCEiCP described in [1, 5, 7, 19] are designed for computing a Lorentz pair whose eigenvalue may be not the smallest. Hence these procedures cannot be applied to solve ELE. On the other hand, for the traditional eigenvalue problem there are plenty of state-of-the-art algorithms for solving both small-to-medium sized problems (e.g., the QR algorithm) and large-scale problems (e.g., methods based on the Krylov subspace techniques). In addition to the highly efficient performance numerically, elegant theoretical results have been

developed. A complete list of this ever-expanding literature is apparently hard to present, and so we only mention [3, 14, 21, 36, 42, 50] for general discussions.

Treating (1.3) as a specific eigenvalue problem, we attempt to employ the maturely developed Lanczos method for eigenvalue computations to solve (1.1). Mimicking the classical Rayleigh–Ritz (RR) procedure (see [36, section 11.3] and [14, Definition 7.1]; also see section 3.1) for the eigenvalue problem, we propose a Lanczos-type method for the Lorentz eigenvalue problem (LaLoEig). The detailed procedure is presented in section 3. This method is suitable for large-scale and sparse problems because only the matrix-vector products are involved for the matrix A, and thereby, its sparsity and special structure, if any, can be preserved. Moreover, mature and fruitful theoretical results of the Lanczos method can be used to establish the convergence of LaLoEig.

We organize the paper in the following way. In section 2, we present some preliminary results, including a basic computational procedure for ELE and a brief review of the trust-region subproblem (TRS). The LaLoEig method is proposed in section 3 with a basic introduction of the RR procedure, the Lanczos method for eigenvalue problems, and the Lanczos approach for TRS (LTRS). The convergence analysis of LaLoEig is discussed in section 4. Some preliminary numerical experiments are reported in section 5, and concluding remarks are drawn in section 6.

Notation. Throughout the paper, all vectors are column vectors and are typeset in bold. The identity matrix in $\mathbb{R}^{n\times n}$ is denoted by $I_n = [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_n]$. To simplify our presentation, we adopt MATLAB-like conventions to access the entries of vectors and matrices. For example, $A(k,\ell)$ is the (k,ℓ)th entry of A, $(k:\ell)$ stands for the set of integers from k to ℓ inclusively, and $A(k:\ell, i:\ell)$ is the submatrix of A that consists of intersections of row k to row ℓ and column i to column ℓ.

For the matrix A, $A^\dagger$ stands for the pseudoinverse of A, and A^{T} and $\mathrm{span}(A)$ denote its transpose and the range of A, respectively. The rank of A is $\mathrm{rank}(A) = r(A)$. The eigendecomposition of A is $A = V\Theta V^{\mathrm{T}}$, and its eigenvalues are represented as

$$\theta_1 = \theta_2 = \cdots = \theta_j < \theta_{j+1} \le \cdots \le \theta_n;$$

the eigenspace associated with the smallest eigenvalue θ_1 is denoted by $\mathcal{A}_1$ which is spanned by $V_1 = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_j] \in \mathbb{R}^{n\times j}$, i.e., $\mathcal{A}_1 = \mathrm{span}(V_1)$ and $j = \dim(\mathcal{A}_1)$.

Let $\mathcal{X}$ and $\mathcal{L}$ be two subspaces of $\mathbb{R}^n$ with $\chi = \dim(\mathcal{X}) \le \dim(\mathcal{L}) = \ell$, and let X and L be orthonormal basis matrices of $\mathcal{X}$ and $\mathcal{L}$, respectively. We denote by σ_i for $1 \le i \le \chi$, in ascending order, the singular values of $L^{\mathrm{T}}X$. The χ *canonical angles* $\angle_i(\mathcal{X},\mathcal{L})$ from $\mathcal{X}$ to $\mathcal{L}$, in descending order, are defined by [31]

$$0 \le \angle_i(\mathcal{X},\mathcal{L}) \triangleq \arccos\sigma_i \le \frac{\pi}{2} \quad \text{for } 1 \le i \le \chi. \tag{1.4}$$

We set

$$\angle(\mathcal{X},\mathcal{L}) \triangleq \mathrm{diag}(\angle_1(\mathcal{X},\mathcal{L}), \ldots, \angle_\chi(\mathcal{X},\mathcal{L})), \tag{1.5}$$

$$\sin\angle(\mathcal{X},\mathcal{L}) \triangleq \mathrm{diag}(\sin\angle_1(\mathcal{X},\mathcal{L}), \ldots, \sin\angle_\chi(\mathcal{X},\mathcal{L})) \tag{1.6}$$

and analogously define $\cos\angle(\mathcal{X},\mathcal{L})$ and $\tan\angle(\mathcal{X},\mathcal{L})$. When $\chi = \ell$, it is known that the distance (in the ℓ_2-norm) between $\mathcal{X}$ and $\mathcal{L}$ is (see, e.g., [21, section 2.6.3])

$$\|\sin\angle(\mathcal{X},\mathcal{L})\|_2 = \|P_{\mathcal{X}} - P_{\mathcal{L}}\|_2, \tag{1.7}$$

where $P_{\mathcal{X}}$ and $P_{\mathcal{L}}$ stand for the orthogonal projections onto $\mathcal{X}$ and $\mathcal{L}$, respectively. Also, for a given nonzero $\boldsymbol{x} \in \mathbb{R}^n$, we will use $\angle(\boldsymbol{x},\mathcal{L})$ to denote $\angle(\mathrm{span}(\boldsymbol{x}),\mathcal{L})$, and

by definition, it follows that

$$\cos\angle(\boldsymbol{x},\mathcal{L}) = \frac{\|L^{\mathrm{T}}\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2} \quad \text{and} \quad \sin\angle(\boldsymbol{x},\mathcal{L}) = \sqrt{1-\cos^2\angle(\boldsymbol{x},\mathcal{L})}.$$

For the cone $\mathbb{K}^n$, we denote its *interior* and *boundary* by

$$\mathrm{int}(\mathbb{K}^n) \triangleq \{\boldsymbol{x}\in\mathbb{K}^n : \boldsymbol{x}(1) > \|\boldsymbol{x}(2:n)\|_2\} \text{ and } \mathrm{bd}(\mathbb{K}^n) \triangleq \{\boldsymbol{x}\in\mathbb{K}^n : \boldsymbol{x}(1) = \|\boldsymbol{x}(2:n)\|_2\},$$

respectively.

2. Breaking down the ELE problem.

2.1. Preliminary results.

We begin with a geometrically obvious result about the Lorentz cone $\mathbb{K}^n$.

PROPOSITION 2.1. *Let L be an orthonormal basis for a subspace $\mathcal{L}\subseteq\mathbb{R}^n$. Then*
(i) *$\mathcal{L}\cap(\mathbb{K}^n\backslash\{\mathbf{0}\})\neq\emptyset$ if and only if $\angle(\boldsymbol{e}_1,\mathcal{L})\le\frac{\pi}{4}$;*
(ii) *$\mathcal{L}\cap\mathrm{int}(\mathbb{K}^n)\neq\emptyset$ if and only if $\angle(\boldsymbol{e}_1,\mathcal{L})<\frac{\pi}{4}$.*

Remark 2.1. For this proposition, we remark that the Lorentz cone $\mathbb{K}^n$ is a special case of the so-called revolution cone [45] defined by

$$\mathrm{Rev}(\mathbf{b},\phi) = \{\boldsymbol{x}\in\mathbb{R}^n : (\cos\phi)\|\boldsymbol{x}\|_2 \le \langle\mathbf{b},\boldsymbol{x}\rangle\}$$

associated with a given unit norm $\mathbf{b}$ and a $\phi\in[0,\frac{\pi}{2}]$. It is clear that $\mathbb{K}^n = \mathrm{Rev}(\boldsymbol{e}_1,\frac{\pi}{4})$, and Proposition 2.1 is a necessary and sufficient condition for the nontrivial intersection between $\mathbb{K}^n$ and a linear subspace $\mathcal{L}$ [45]. Also, for a given $\mathrm{Rev}(\mathbf{b},\phi)$, we can transfer any $\boldsymbol{z}\in\mathrm{Rev}(\mathbf{b},\phi)$ into $\boldsymbol{x}\in\mathbb{K}^n$ by the transformation $\boldsymbol{x} = \mathrm{diag}(1, I_{n-1}\cot\phi)H\boldsymbol{z}$, where H is the Householder transformation satisfying $H\mathbf{b}=\boldsymbol{e}_1$; therefore, our discussions in this paper can also be applied to the revolution cone.

Next, we provide some basic facts about the Lorentz eigenvalue problem. Following [46], we define the *Lorentz spectrum*

$$\sigma(A,\mathbb{K}^n) \triangleq \{\lambda\in\mathbb{R} : (\lambda,\boldsymbol{x}) \text{ is a Lorentz eigenpair of (1.3) for some } \boldsymbol{x}\}.$$

It is known that for a symmetric A, $\sigma(A,\mathbb{K}^n)$ contains a finite number of Lorentz eigenvalues [46, Corollary 4.5]. This means that we can order all the Lorentz eigenvalues as

$$\lambda_1 \le \lambda_2 \le \cdots \le \lambda_t,$$

and thus the extreme Lorentz eigenvalue $\lambda_1 = \hat{q}$ is also the global minimum of ELE (1.1). Moreover, it obviously holds that

$$\sigma(A,\mathbb{K}^n) = \sigma_{\mathrm{int}}(A,\mathbb{K}^n)\cup\sigma_{\mathrm{bd}}(A,\mathbb{K}^n),$$

where $\sigma_{\mathrm{int}}(A,\mathbb{K}^n)$ (resp., $\sigma_{\mathrm{bd}}(A,\mathbb{K}^n)$) consists of all Lorentz eigenvalues, each admitting a Lorentz eigenvector $\boldsymbol{x}\in\mathrm{int}(\mathbb{K}^n)$ (resp., $\boldsymbol{x}\in\mathrm{bd}(\mathbb{K}^n)$). We should be aware that $\sigma_{\mathrm{int}}(A,\mathbb{K}^n)$ and $\sigma_{\mathrm{bd}}(A,\mathbb{K}^n)$ are not necessarily disjoint because some Lorentz eigenvalue could possibly have Lorentz eigenvectors both in $\mathrm{int}(\mathbb{K}^n)$ and on $\mathrm{bd}(\mathbb{K}^n)$; moreover [46]

$$\lambda\in\sigma_{\mathrm{int}}(A,\mathbb{K}^n) \Longleftrightarrow \lambda \text{ is an eigenvalue of } A \text{ associated with an eigenvector in } \mathrm{int}(\mathbb{K}^n).$$

In particular, we have the following.

LEMMA 2.1. *If $(\lambda, \boldsymbol{x})$ is a Lorentz eigenpair and $\boldsymbol{x} \in \mathrm{int}(\mathbb{K}^n)$, then $(\lambda, \boldsymbol{x})$ is also an eigenpair of A.*

Proof. The assertion can be verified easily using $\boldsymbol{x}^{\mathrm{T}}(A\boldsymbol{x} - \lambda\boldsymbol{x}) = 0$, $\boldsymbol{x} \in \mathrm{int}(\mathbb{K}^n)$, $A\boldsymbol{x} - \lambda\boldsymbol{x} \in \mathbb{K}^n$, and the Moreau orthogonal decomposition theorem. ∎

2.2. Basic computational steps for ELE. If for the extreme Lorentz eigenvalue λ_1 there is a Lorentz eigenvector $\hat{\boldsymbol{x}} \in \mathrm{int}(\mathbb{K}^n)$, then $(\lambda_1, \hat{\boldsymbol{x}})$ is the eigenpair of A and $\lambda_1 = \theta_1$ is the smallest eigenvalue of A. Otherwise, any associated Lorentz eigenvector $\hat{\boldsymbol{x}}$ must be on the boundary $\hat{\boldsymbol{x}} \in \mathrm{bd}(\mathbb{K}^n)$. We point out that the latter case happens only if the eigenspace of A associated with the smallest eigenvalue θ_1 is separated from $\mathbb{K}^n$. This is shown in the following theorem.

THEOREM 2.1. *Let $\mathcal{A}_1$ be the eigenspace of A associated with the smallest eigenvalue θ_1.*

(i) *If $\mathcal{A}_1 \cap (\mathbb{K}^n \backslash \{\mathbf{0}\}) \neq \emptyset$, then $\lambda_1 = \theta_1$, and an eigenvector in $\mathbb{K}^n$ of A associated with θ_1 solves* ELE (1.1).

(ii) *If $\mathcal{A}_1 \cap (\mathbb{K}^n \backslash \{\mathbf{0}\}) = \emptyset$, then $\theta_1 < \lambda_1 \in \sigma_{\mathrm{bd}}(A, \mathbb{K}^n)$, and any Lorentz eigenvector associated with λ_1 is given by $\hat{\boldsymbol{x}} = \frac{\sqrt{2}}{2}\left[\begin{smallmatrix} 1 \\ \hat{\boldsymbol{s}} \end{smallmatrix}\right] \in \mathrm{bd}(\mathbb{K}^n)$, where $\hat{\boldsymbol{s}} \in \mathbb{R}^{n-1}$ solves the following problem:*

$$\min_{\|\boldsymbol{s}\|_2 = 1} \frac{1}{2}\boldsymbol{s}^{\mathrm{T}} H \boldsymbol{s} + \boldsymbol{s}^{\mathrm{T}}\boldsymbol{g}, \tag{2.1}$$

where

$$A = \begin{bmatrix} a_{11} & \boldsymbol{g}^{\mathrm{T}} \\ \boldsymbol{g} & H \end{bmatrix}, \quad \boldsymbol{g} \in \mathbb{R}^{n-1}, \quad H \in \mathbb{R}^{(n-1)\times(n-1)}.$$

Proof. For (i), note that

$$\theta_1 = \min_{\|\boldsymbol{x}\|_2 = 1} q_A(\boldsymbol{x}) \leq \min_{\|\boldsymbol{x}\|_2 = 1,\ \boldsymbol{x} \in \mathbb{K}^n} q_A(\boldsymbol{x}) = \lambda_1,$$

and if $\mathcal{A}_1 \cap (\mathbb{K}^n \backslash \{\mathbf{0}\}) \neq \emptyset$, any unit ℓ_2-norm vector in $\mathcal{A}_1 \cap (\mathbb{K}^n \backslash \{\mathbf{0}\})$ solves ELE (1.1).

For (ii), assume that there is a Lorentz eigenpair $(\lambda_1, \hat{\boldsymbol{x}})$ with $\hat{\boldsymbol{x}} \in \mathrm{int}(\mathbb{K}^n)$. Then by Lemma 2.1, we know that $(\lambda_1, \hat{\boldsymbol{x}})$ is also an eigenpair of A. But the assumption $\mathcal{A}_1 \cap (\mathbb{K}^n \backslash \{\mathbf{0}\}) = \emptyset$ implies that $\lambda_1 > \theta_1$ and $\boldsymbol{v}_1^{\mathrm{T}}\hat{\boldsymbol{x}} = 0$, where $\boldsymbol{v}_1 \in \mathcal{A}_1$ has unit ℓ_2-norm. Consider the point of the form $\psi\hat{\boldsymbol{x}} + \beta\boldsymbol{v}_1$. Note that for any $\beta \neq 0$ and $\psi^2 + \beta^2 = 1$,

$$\|\psi\hat{\boldsymbol{x}} + \beta\boldsymbol{v}_1\|_2 = 1 \quad \text{and} \quad q_A(\psi\hat{\boldsymbol{x}} + \beta\boldsymbol{v}_1) = \lambda_1\psi^2 + \theta_1\beta^2 < \lambda_1.$$

For sufficiently small $\beta \neq 0$, let $\psi = \sqrt{1 - \beta^2}$. Since $\hat{\boldsymbol{x}} \in \mathrm{int}(\mathbb{K}^n)$, we have $\|\psi\hat{\boldsymbol{x}} + \beta\boldsymbol{v}_1\|_2 = 1$, $\psi\hat{\boldsymbol{x}} + \beta\boldsymbol{v}_1 \in \mathbb{K}^n$, and $q_A(\psi\hat{\boldsymbol{x}} + \beta\boldsymbol{v}_1) < \lambda_1$, which contradicts that λ_1 is the minimum of ELE (1.1). Therefore, $\hat{\boldsymbol{x}} \in \mathrm{bd}(\mathbb{K}^n)$ holds and the remaining conclusion follows directly. ∎

By Proposition 2.1, it appears that the computational step for checking if $\mathcal{A}_1 \cap (\mathbb{K}^n \backslash \{\mathbf{0}\}) \neq \emptyset$ is simple. Suppose $V_1 \in \mathbb{R}^{n\times j}$ is the orthonormal basis matrix for $\mathcal{A}_1$. Then

$$\mathcal{A}_1 \cap (\mathbb{K}^n \backslash \{\mathbf{0}\}) \neq \emptyset \Longleftrightarrow \|V_1^{\mathrm{T}}\boldsymbol{e}_1\|_2^2 \geq \frac{1}{2}.$$

Theorem 2.1 together with the above equivalence provides a computational approach (Algorithm 2.1) for solving ELE (1.1). This procedure starts by checking the assumption in Theorem 2.1(i). If it does not hold, the related problem (2.1) is solved.

Algorithm 2.1 Basic procedure for solving ELE (1.1)

Input: a symmetric matrix A;
Output: an extreme Lorentz eigenpair $(\lambda_1, \hat{\boldsymbol{x}})$ of A;

1: compute the orthonormal basis matrix $V_1 \in \mathbb{R}^{n\times j}$ for the eigenspace of $\mathcal{A}_1$ associated with the smallest eigenvalue θ_1 of A;
2: if $2\|V_1^{\mathrm{T}}\boldsymbol{e}_1\|_2^2 \geq 1$, then either $\hat{\boldsymbol{x}} = \frac{V_1V_1^{\mathrm{T}}\boldsymbol{e}_1}{\|V_1^{\mathrm{T}}\boldsymbol{e}_1\|_2}$ or $\hat{\boldsymbol{x}} = -\frac{V_1V_1^{\mathrm{T}}\boldsymbol{e}_1}{\|V_1^{\mathrm{T}}\boldsymbol{e}_1\|_2}$ solves (1.1) and $\lambda_1 = \theta_1$;
3: otherwise solve (2.1) for $\hat{\boldsymbol{s}}$ and $\hat{\boldsymbol{x}} = \frac{\sqrt{2}}{2}\left[\begin{smallmatrix}1\\ \hat{\boldsymbol{s}}\end{smallmatrix}\right] \in \mathrm{bd}(\mathbb{K}^n)$ and $\lambda_1 = q_A(\hat{\boldsymbol{x}})$.

We should point out that Algorithm 2.1 is only suitable for small-to-medium sized n, as it involves solving a classical eigenvalue problem. The detailed procedure of LaLoEig for large-scale problems is to be presented in section 3.

2.3. The trust-region subproblem. We next discuss the related problem (2.1) briefly. Note that for any $\nu \in \mathbb{R}$, (2.1) is equivalent to

$$\min_{\|\boldsymbol{s}\|_2=1} \frac{1}{2}\boldsymbol{s}^{\mathrm{T}}H_\nu\boldsymbol{s} + \boldsymbol{s}^{\mathrm{T}}\boldsymbol{g}, \quad \text{where} \quad H_\nu = H - \nu I_{n-1}. \tag{2.2}$$

If we choose ν so that H_ν is not positive definite, then (2.2) is equivalent to the so-called trust-region subproblem (TRS):

$$\min_{\|\boldsymbol{s}\|_2\leq 1} \frac{1}{2}\boldsymbol{s}^{\mathrm{T}}H_\nu\boldsymbol{s} + \boldsymbol{s}^{\mathrm{T}}\boldsymbol{g}. \tag{2.3}$$

The well-known optimality conditions of the global optimal solution of (2.3) due to Moré and Sorensen [33] (see also [47] and [35, Theorem 4.1]) read as follows.

LEMMA 2.2. *The vector $\hat{\boldsymbol{s}}$ is a global optimal solution of the trust-region problem (2.3) if and only if $\hat{\boldsymbol{s}}$ is feasible and there is a scalar $\hat{\varrho} \geq 0$ such that the following conditions are satisfied:*

$$(H_\nu + \hat{\varrho}I_{n-1})\hat{\boldsymbol{s}} = -\boldsymbol{g}, \quad \hat{\varrho}(1 - \|\hat{\boldsymbol{s}}\|_2) = 0, \quad H_\nu + \hat{\varrho}I_{n-1} \text{ is positive semidefinite.} \tag{2.4}$$

Let the spectral decomposition of H be

$$H = U\,\mathrm{diag}(\omega_1, \omega_2, \ldots, \omega_{n-1})U^{\mathrm{T}} \triangleq U\Omega U^{\mathrm{T}} \tag{2.5}$$

and $\mathcal{H}_1$ be the invariant subspace associated with the smallest eigenvalue $\omega_1 = \omega_2 = \cdots = \omega_p$. Thus $U_1 = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_p] \in \mathbb{R}^{(n-1)\times p}$ is an orthonormal basis matrix for $\mathcal{H}_1$ and $U = [U_1, U_2]$ and $\mathcal{H}_2 = \mathrm{span}(U_2)$.

There are two cases (e.g., [24, 33, 35]) of (2.3) to be considered:

1. The *degenerate case* [24, Lemma 2.2] (or the *hard case* [35])[1] means that

$$\boldsymbol{g} \perp \mathcal{H}_1 \ \text{ and } \ \|(H_\nu - (\omega_1 - \nu)I_{n-1})^\dagger\boldsymbol{g}\|_2 = \|(H - \omega_1 I_{n-1})^\dagger\boldsymbol{g}\|_2 \leq 1 \tag{2.6}$$

and the corresponding KKT multiplier is $\hat{\varrho} = -\omega_1 + \nu$. There are multiple global solutions $\hat{\boldsymbol{s}}$, each taking the form [24, Lemma 2.2]

$$\hat{\boldsymbol{s}} = -(H - \omega_1 I_{n-1})^\dagger\boldsymbol{g} + \tau\boldsymbol{u} \quad \forall \boldsymbol{u} \in \mathcal{H}_1 \text{ and } \|\boldsymbol{u}\|_2 = 1, \tag{2.7}$$

with

$$\tau^2 = \Delta^2 - \|(H - \omega_1 I_{n-1})^\dagger\boldsymbol{g}\|_2^2 \geq 0.$$

[1] We adopt the definitions of degenerate and nondegenerate cases of [24, Lemma 2.2] in this paper.

2. The *nondegenerate case* [24, Lemma 2.2] (or the *easy case* [35]) is the situation where (2.6) is no longer true. In this case, the corresponding KKT multiplier is $\hat{\varrho} > -\omega_1 + \nu$. The unique global solution [24, Lemma 2.2] satisfies

$$(H_\nu + \hat{\varrho} I_{n-1})\hat{\boldsymbol{s}} = -\boldsymbol{g}.$$

Because of its vital role in numerous applications, there are several algorithms for solving (2.3). Basically, these algorithms can be classified into two categories: algorithms based on matrix factorizations for small-to-medium sized dense problems (see, e.g., [33, 35]) and factorization-free algorithms for large-scale sparse problems (see, e.g., [22, 23, 24, 35, 37, 38, 39, 40, 48, 49, 51, 53]).

The Moré–Sorensen method [33] is probably the most well-known method for small-to-medium sized dense problems and is frequently embedded into procedures as a building block for solving relevant subproblems within large-scale computational problems. This is the case of the Lanczos-type method proposed in [23] (see also [9, Chapter 5]) for the large-scale TRS problem (2.3). In particular, a Lanczos method [23, section 5] for TRS (LTRS) basically follows the RR procedure (see section 3.1), whose convergence analysis was recently established in [55]. We show that LTRS can be perfectly built into the framework of our new algorithm LaLoEig because the standard Lanczos method for computing approximately the orthonormal basis matrix V_1 of $\mathcal{A}_1$ and LTRS can be nicely incorporated. The detailed procedure is to be shown in section 3.

3. Lanczos method for the Lorentz eigenvalue problem.

3.1. Rayleigh–Ritz procedure and the (block) Lanczos method. Since our new method LaLoEig is a Krylov subspace algorithm and follows the Rayleigh–Ritz (RR) procedure, we first present a brief and general explanation of the RR procedure (see [36, section 11.3] and [14, Definition 7.1]) for the symmetric eigenvalue problem $A\boldsymbol{x} = \lambda\boldsymbol{x}$. Basically, the RR procedure consists of the following three steps:

(a) seek a good subspace together with an orthonormal basis Q_k that approximates the eigenspace of A;
(b) form $T_k = Q_k^{\mathrm{T}} A Q_k$ and compute the eigenpairs $(\nu_i, \boldsymbol{r}_i)$ of T_k;
(c) form the *Ritz pairs* $(\nu_i, Q_k\boldsymbol{r}_i)$ as approximates to the eigenpairs of A.

In practice, an orthonormal basis Q_k of the subspace in (a) is commonly generated by the classical Lanczos three-term recurrence [43, Algorithm 6.15], which is a single-vector version of the Lanczos process. It is well known that, unless a certain deflating strategy is employed, the single-vector version can only find one copy of any multiple eigenvalue and also possesses slow convergence toward clustered eigenvalues (see [36, section 13.10] and more recently [31]). Since Algorithm 2.1 requires checking if $\mathcal{A}_1 \cup (\mathbb{K}^n \backslash \{\mathbf{0}\})$ is nonempty, we prefer to use the block (or the band) Lanczos process [10, 20] as it is capable of finding all copies of a multiple eigenvalue with a suitable block size; in other words, the block Lanczos process is able to achieve (or approximate) an orthonormal basis matrix for $\mathcal{A}_1$.

There are several versions of the block Lanczos process (see, e.g., [4, 36, 41]), but the simplest version [10, 20] proceeds as presented in Algorithm 3.1. Note that Algorithm 3.1 with $b = 1$ is the classical three-term-recurrence single-vector Lanczos process.

Starting from the initial orthogonal matrix $G_1 \in \mathbb{R}^{n\times b}$ with the block size $b \geq 1$ and assuming $r(Z) = b$ for $i = 1, 2, \ldots, k$ at line 8, the block Lanczos process in Algorithm 3.1 generates an orthonormal basis $Q_k \triangleq [G_1, \ldots, G_k] \in \mathbb{R}^{n\times bk}$ of the

Algorithm 3.1 Simple block Lanczos process

Given a symmetric $A \in \mathbb{R}^{n\times n}$ and an initial orthogonal matrix $G_1 \in \mathbb{R}^{n\times b}$, this generic block Lanczos process generates the Krylov subspace $\mathcal{K}_k(A, G_1)$ and the orthonormal basis $Q_k = [G_1, \ldots, G_k]$.

1: $Z = AG_1$, $A_1 = G_1^{\mathrm{T}} Z$;
2: $Z = Z - G_1 A_1$;
3: perform orthogonalization on Z to obtain $Z = G_2 B_1$, where $G_2 \in \mathbb{R}^{n\times b}$ satisfying $G_2^{\mathrm{T}} G_2 = I_b$ and $B_1 \in \mathbb{R}^{b\times b}$;
4: **for** $i = 2, \ldots, k$ **do**
5: $\quad Z = AG_i$, $A_i = G_i^{\mathrm{T}} Z$;
6: $\quad Z = Z - G_i A_i - G_{i-1} B_{i-1}^{\mathrm{T}}$;
7: $\quad$ if $Z = 0$, then *break*;
8: $\quad$ find an orthonormal basis G_{i+1} for Z so that $Z = G_{i+1} B_i$, $G_{i+1} \in \mathbb{R}^{n\times r(Z)}$;
9: **end for**

Krylov subspace

$$\mathcal{K}_k(A, G_1) = \mathrm{span}(G_1, AG_1, \ldots, A^{k-1}G_1) = \mathrm{span}(G_1) \oplus \cdots \oplus \mathrm{span}(G_k). \tag{3.1}$$

Compactly, the process yields the relationship

$$AQ_k = Q_k T_k + [0_{n\times(k-1)b}, G_{k+1}B_k], \tag{3.2}$$

where

$$T_k = Q_k^{\mathrm{T}} A Q_k = \begin{bmatrix} A_1 & B_1^{\mathrm{T}} & & & \\ B_1 & A_2 & B_2^{\mathrm{T}} & & \\ & \ddots & \ddots & \ddots & \\ & & B_{k-2} & A_{k-1} & B_{k-1}^{\mathrm{T}} \\ & & & B_{k-1} & A_k \end{bmatrix} \in \mathbb{R}^{kb\times kb} \tag{3.3}$$

is the so-called Rayleigh quotient matrix with respect to $\mathcal{K}_k(A, G_1)$ and is the projection of A onto $\mathcal{K}_k(A, G_1)$, too.

The modified Gram–Schmidt process or the rank-revealing QR factorization (see, e.g., [21]) can be implemented to find an orthonormal basis G_{i+1} for Z at line 8. Suppose at the ith iteration $r(Z) < b$; then G_{i+1} at line 8 consists of $r(Z)$ columns which are obtained through removing the linearly dependent vectors in Z and orthogonalizing the remaining columns.[2] In this case, $G_{i+1} \in \mathbb{R}^{n\times r(Z)}$, but the relationship (3.2) (with $k = i$) is still valid; the number of columns of G_ℓ in the subsequent $G_{i+2}, \ldots, G_k$ is nonincreasing, and the *breakdown* happens when $Z = 0$ at line 7. We will see later that such a *breakdown* is welcome, implying that the exact solution of ELE (1.1) can be obtained.

The RR procedure with the (block) Lanczos process yields the (block) Lanczos method for solving the symmetric eigenvalue problem. In particular, suppose $b \geq j = \dim(\mathcal{A}_1)$ and an approximation of the eigenspace $\mathcal{A}_1$ associated with the extreme eigenvalue θ_1 is desired. Then we can compute the eigenpairs $(\nu_i, \boldsymbol{r}_i)$ for $i = 1, 2, \ldots, kb$ of T_k and approximate the eigenpairs of A using the Ritz pairs $(\nu_i, Q_k \boldsymbol{r}_i)$. As $kb \ll n$ holds in general, the state-of-the-art eigensolvers such as the QR algorithm can be used

[2] In MATLAB, G_{i+1} can be simply obtained via $G_{i+1} = \mathsf{orth}(Z)$.

for obtaining accurate eigenpairs of T_k. There has been a wealth of development, in both theory and implementation, of Lanczos-based methods, and we refer the reader to [14, 36] for a complete development up to 1998. More recent investigation [30, 31, 43] provides more detailed convergence analysis and shows that, under certain conditions of distribution of the eigenvalues θ_i and the choice of G_1, the first j Ritz vectors $Q_k \boldsymbol{r}_i$ for $i = 1, 2, \ldots, j$ form an accurate orthonormal basis $\widetilde{V}_1$ for the eigenspace $\mathcal{A}_1$, and the dimension $j = \dim(\mathcal{A}_1)$ can be gradually reflected as k increases.

3.2. Lanczos method for TRS (LTRS). Suppose an accurate approximation of $\mathcal{A}_1$ together with an orthonormal basis $\widetilde{V}_1$ is achieved by the block Lanczos method. Then according to Algorithm 2.1, we next check if $2\|\widetilde{V}_1^{\mathrm{T}} \boldsymbol{e}_1\|_2^2 \geq 1$. The algorithm terminates whenever this condition is fulfilled; otherwise, the TRS (2.1) needs to be solved. Fortunately, in this scenario, the information produced by the block Lanczos process can be further utilized efficiently and essentially, with no extra significant computational costs. In other words, the main computation complexity is roughly the same as for computing an accurate approximation of $\mathcal{A}_1$.

To see why this is possible, we first briefly review the Lanczos method (LTRS) for TRS (2.1) proposed in [23] (see also [9, Chapter 5]). LTRS basically follows the RR procedure, too. At the kth iteration, it generates the kth Krylov subspace

$$\mathcal{K}_k(H_\nu, \boldsymbol{g}) = \mathcal{K}_k(H, \boldsymbol{g}) = \mathrm{span}(\boldsymbol{g}, H\boldsymbol{g}, \ldots, H^{k-1}\boldsymbol{g})$$

via the Lanczos process, i.e., Algorithm 3.1 with $b = 1$. Suppose Π_k is the orthonormal basis matrix for $\mathcal{K}_k(H, \boldsymbol{g})$ and, thereby, $\Pi_k(:, 1) = \Pi_k \boldsymbol{e}_1 = \boldsymbol{g}/\|\boldsymbol{g}\|_2$. The kth approximate $\boldsymbol{s}_k$ of (2.1) is defined as the solution to

$$\min_{\boldsymbol{s} \in \mathcal{K}_k(H, \boldsymbol{g}), \|\boldsymbol{s}\|_2 \leq 1} \left\{ \frac{1}{2} \boldsymbol{s}^{\mathrm{T}} H_\nu \boldsymbol{s} + \boldsymbol{s}^{\mathrm{T}} \boldsymbol{g} \right\}. \tag{3.4}$$

Denoting $\boldsymbol{s} \in \mathcal{K}_k(H, \boldsymbol{g})$ by $\boldsymbol{s} = \Pi_k \boldsymbol{y}$ for $\boldsymbol{y} \in \mathbb{R}^k$, we know that $\boldsymbol{s}_k = \Pi_k \boldsymbol{y}_k$, where $\boldsymbol{y}_k$ is defined as the solution to the projected and reduced TRS:

$$\boldsymbol{y}_k = \operatorname*{argmin}_{\|\boldsymbol{y}\|_2 \leq 1} \left\{ \frac{1}{2} \boldsymbol{y}^{\mathrm{T}} \Pi_k^{\mathrm{T}} H_\nu \Pi_k \boldsymbol{y} + \|\boldsymbol{g}\|_2 \boldsymbol{e}_1^{\mathrm{T}} \boldsymbol{y} \right\}. \tag{3.5}$$

As the size k of the projected TRS (3.5) is small in general, sophisticated solvers based on matrix-factorization such as certain modifications of the Moré–Sorensen method [33] can be employed as the computational cost of solving (3.5) is roughly negligible. Numerical testing for LTRS indicates that it is very efficient, especially when TRS (2.1) is not close to the degenerate case; moreover, the theoretical convergence is recently established in [55].

3.3. Algorithmic framework: LaLoEig. To see how to incorporate the block Lanczos method for computing an (approximate) orthonormal basis of $\mathcal{A}_1$ in solving (2.1), we use the trick of choosing

$$G_1(:, 1) = G_1 \boldsymbol{e}_1 = \boldsymbol{e}_1 \in \mathbb{R}^n.$$

In other words, we select the initial block G_1 whose first column is $\boldsymbol{e}_1$ in Algorithm 3.1. Noting that $Q_k = [G_1, G_2, \ldots, G_k] \in \mathbb{R}^{n \times kb}$ is an orthonormal basis for $\mathcal{K}_k(A, G_1)$, such a choice of G_1 implies that

$$Q_k = [G_1, G_2, \ldots, G_k] = \begin{bmatrix} 1 & \mathbf{0}^{\mathrm{T}} \\ \mathbf{0} & \widehat{Q}_k \end{bmatrix}, \text{ with } \widehat{Q}_k^{\mathrm{T}} \widehat{Q}_k = I_{kb-1}, \tag{3.6}$$

and

$$\widehat{T}_k \triangleq T_k(2:kb, 2:kb) = \widehat{Q}_k^{\mathrm{T}} H \widehat{Q}_k \in \mathbb{R}^{(kb-1)\times(kb-1)}. \tag{3.7}$$

It is also true that by $\boldsymbol{x} = Q_k \boldsymbol{y}$, the constraint

$$\{\boldsymbol{x} : \boldsymbol{x} \in \mathcal{K}_k(A, G_1),\ \|\boldsymbol{x}\|_2 = 1 \text{ and } \boldsymbol{x} \in \mathbb{K}^n\}$$

is equivalent to $\{Q_k \boldsymbol{y} : \|\boldsymbol{y}\|_2 = 1 \text{ and } \boldsymbol{y} \in \mathbb{K}^{bk}\}$. Most importantly, we have the following theorem.

THEOREM 3.1. *Suppose the initial orthogonal block $G_1 \in \mathbb{R}^{n\times b}$ $(b \geq 1)$ satisfies $G_1(:,1) = \boldsymbol{e}_1 \in \mathbb{R}^n$ and $Q_k = [G_1, \ldots, G_k] \in \mathbb{R}^{n\times kb}$ is the orthonormal basis for $\mathcal{K}_k(A, G_1)$ generated by the block Lanczos process in Algorithm* 3.1. *Then*

$$\mathcal{K}_{k-1}(H, \boldsymbol{g}) \subseteq \mathrm{span}(\widehat{Q}_k).$$

If $b = 1$, then

$$\mathcal{K}_{k-1}(H, \boldsymbol{g}) = \mathrm{span}(\widehat{Q}_k).$$

Proof. It is clear that $\mathcal{K}_k(A, \boldsymbol{e}_1) \subseteq \mathcal{K}_k(A, G_1) = \mathrm{span}(Q_k)$. Assume that

$$W_k = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_k] \in \mathbb{R}^{n\times k}$$

is the orthonormal basis matrix for $\mathcal{K}_k(A, \boldsymbol{e}_1)$ generated by the Lanczos process starting from $\boldsymbol{e}_1$. Note that $W_k(:,1) = W_k \boldsymbol{e}_1 = \boldsymbol{w}_1 = \boldsymbol{e}_1$ and also

$$W_k = \begin{bmatrix} 1 & \mathbf{0}^{\mathrm{T}} \\ \mathbf{0} & \widehat{W}_k \end{bmatrix}.$$

We show that $\widehat{W}_k = [\hat{\boldsymbol{w}}_1, \hat{\boldsymbol{w}}_2, \ldots, \hat{\boldsymbol{w}}_{k-1}] \in \mathbb{R}^{(n-1)\times(k-1)}$ is the orthonormal basis matrix for $\mathcal{K}_{k-1}(H, \boldsymbol{g})$ generated by the Lanczos process.

To this end, we assume that the compact relation of the Lanczos process (i.e., Algorithm 3.1 with $b = 1$) applying to $\mathcal{K}_k(A, \boldsymbol{e}_1)$ starting from $\boldsymbol{e}_1$ is

$$AW_k = W_k C_k + \beta_k \boldsymbol{w}_{k+1} \boldsymbol{e}_k^{\mathrm{T}} \quad \text{and} \quad W_k^{\mathrm{T}} A W_k = C_k \in \mathbb{R}^{k\times k}, \tag{3.8}$$

where C_k is tridiagonal, $W_k^{\mathrm{T}} \boldsymbol{w}_{k+1} = \mathbf{0}$, and $\beta_k = \|A\boldsymbol{w}_k - \boldsymbol{w}_k C_k(k,k) - \boldsymbol{w}_{k-1} C_k(k-1,k)\|_2$. This relation also implies that

$$W_k(:,2) = \boldsymbol{w}_2 = \begin{bmatrix} 0 \\ \boldsymbol{g}/\|\boldsymbol{g}\|_2 \end{bmatrix}, \text{ i.e., } \hat{\boldsymbol{w}}_1 = \boldsymbol{g}/\|\boldsymbol{g}\|_2.$$

Moreover, by the structure of W_k and (3.8), the following relation also holds:

$$H\widehat{W}_k = \widehat{W}_k C_k(2:k, 2:k) + \beta_k \hat{\boldsymbol{w}}_{k+1} \boldsymbol{e}_{k-1}^{\mathrm{T}},$$

where $C_k(2:k, 2:k) = \widehat{W}_k^{\mathrm{T}} H \widehat{W}_k \in \mathbb{R}^{(k-1)\times(k-1)}$ is tridiagonal and $\hat{\boldsymbol{w}}_{k+1} = \boldsymbol{w}_{k+1}(2:k)$. As a result, $\widehat{W}_k$ is indeed the orthonormal basis matrix of $\mathcal{K}_{k-1}(H, \boldsymbol{g})$ generated by the Lanczos process, i.e., $\mathcal{K}_{k-1}(H, \boldsymbol{g}) = \mathrm{span}(\widehat{W}_k)$. On the other hand, from

$$\mathrm{span}(W_k) = \mathcal{K}_k(A, \boldsymbol{e}_1) \subseteq \mathcal{K}_k(A, G_1) = \mathrm{span}(Q_k),$$

it holds that $\mathrm{span}(\widehat{W}_k) \subseteq \mathrm{span}(\widehat{Q}_k)$, and

$$\mathcal{K}_{k-1}(H, \boldsymbol{g}) = \mathrm{span}(\widehat{W}_k) \subseteq \mathrm{span}(\widehat{Q}_k)$$

as required.

When $b = 1$, it is easy to see that $\mathrm{span}(\widehat{W}_k) = \mathcal{K}_{k-1}(H, \boldsymbol{g}) = \mathrm{span}(\widehat{Q}_k)$. ∎

Algorithm 3.2 Algorithmic framework of LaLoEig for solving ELE (1.1)

Input: a symmetric A;

Output: an (approximate) extreme Lorentz eigenpair $(q_A(\hat{\boldsymbol{x}}_k), \hat{\boldsymbol{x}}_k)$ of $(\lambda_1, \hat{\boldsymbol{x}})$;

1: choose an initial block $G_1 \in \mathbb{R}^{n\times b}$ $(b \geq 1)$ satisfying $G_1^{\mathrm{T}} G_1 = I_b$ and $G_1(:,1) = \boldsymbol{e}_1$;

2: apply the block Lanczos method to get the (approximate) smallest eigenvalue $\tilde{\theta}_1$ and an (approximate) orthonormal basis $\widetilde{V}_1 \in \mathbb{R}^{n\times j}$ of the eigenspace $\mathcal{A}_1$. Let Q_k be the orthonormal basis generated by the block Lanczos process (Algorithm 3.1) for $\mathcal{K}_k(A, G_1)$ and satisfy (3.2);

3: if $2\|\widetilde{V}_1^{\mathrm{T}}\boldsymbol{e}_1\|_2^2 \geq 1$, then either $\hat{\boldsymbol{x}}_k = \frac{\widetilde{V}_1\widetilde{V}_1^{\mathrm{T}}\boldsymbol{e}_1}{\|\widetilde{V}_1^{\mathrm{T}}\boldsymbol{e}_1\|_2}$ or $\hat{\boldsymbol{x}}_k = -\frac{\widetilde{V}_1\widetilde{V}_1^{\mathrm{T}}\boldsymbol{e}_1}{\|\widetilde{V}_1^{\mathrm{T}}\boldsymbol{e}_1\|_2}$ solves (approximately) (1.1) and $\lambda_1 \approx \tilde{\theta}_1$;

4: otherwise solve

$$\boldsymbol{y}_k = \operatorname*{argmin}_{\|\boldsymbol{y}\|_2=1,\ \boldsymbol{y}\in\mathbb{R}^{kb-1}} \left\{ \frac{1}{2}\boldsymbol{y}^{\mathrm{T}}\widehat{T}_k\boldsymbol{y} + \|\boldsymbol{g}\|_2\boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{y} \right\}, \tag{3.9}$$

with $\widehat{T}_k = T_k(2:kb, 2:kb)$, and set $\hat{\boldsymbol{x}}_k = \frac{\sqrt{2}}{2}\begin{bmatrix} 1 \\ \widehat{Q}_k\boldsymbol{y}_k \end{bmatrix} \in \mathrm{bd}(\mathbb{K}^n)$ and $\lambda_1 \approx q_A(\hat{\boldsymbol{x}}_k)$.

Algorithm 3.3 Algorithmic framework of LaLoEig(k) for solving ELE (1.1)

Input: A symmetric matrix A;

Output: The best approximate extreme Lorentz eigenpair $(\lambda_1^{(k)}, \hat{\boldsymbol{x}}^{(k)})$ over $\mathcal{K}_k(A, G_1)$;

1: choose an initial block $G_1 \in \mathbb{R}^{n\times b}$ $(b \geq 1)$ satisfying $G_1^{\mathrm{T}} G_1 = I_b$ and $G_1(:,1) = \boldsymbol{e}_1$;

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

3: compute an orthonormal basis Q_k of $\mathcal{K}_k(A, G_1)$ and the block tridiagonal $T_k = Q_k^{\mathrm{T}} A Q_k \in \mathbb{R}^{kb\times kb}$ given in (3.3) by the block Lanczos process;

4: solve the projected smaller size ELE problem:

$$\boldsymbol{y}_k \triangleq \operatorname*{argmin}_{\|\boldsymbol{y}\|_2=1,\ \boldsymbol{y}\in\mathbb{K}^{kb}} q_{T_k}(\boldsymbol{y}); \tag{3.10}$$

5: set $\hat{\boldsymbol{x}}^{(k)} = Q_k\boldsymbol{y}_k$ and $\lambda_1^{(k)} = q_A(\hat{\boldsymbol{x}}^{(k)}) = q_{T_k}(\boldsymbol{y}_k)$;

6: **end for**

With Theorem 3.1, we are now in a position to present the algorithmic framework of LaLoEig. It basically follows Algorithm 2.1 but with the efficient treatments for Steps 1 and 4.

Remark 3.1. There are several remarks for Algorithm 3.2.

1. The choice of the block size b in general should be larger than the dimension $j = \dim(\mathcal{A}_1)$; however, as j is unknown a priori, a relatively big b can be initialized and the block Lanczos method can give the information of the dimension of $\mathcal{A}_1$. In practice, $b = 2$ or 3 works well. Also, one can employ the adaptive block Lanczos algorithm [54], which adaptively chooses the block size b according to clustering of Ritz values, and can find an approximation of $\mathcal{A}_1$.
2. A large Lanczos step k delivers an accurate $\tilde{\theta}_1 = \nu_1$, and an accurate solution to (2.1) as well. Under certain conditions, we see in section 4 that a *breakdown* (i.e., $Z = 0$ at line 7 of Algorithm 3.1) in the Lanczos process not only

implies that $\tilde{\theta}_1 = \theta_1$ exactly, but also ensures that $Q_k \boldsymbol{y}_k$ solves (2.1) exactly, too. There are mature convergence results for the block Lanczos method for the eigenvalue problem which provide certain criteria for the choice of k. Moreover, a recent convergence analysis made for LTRS in [55] enables us to present a detailed analysis for the accuracy of both the eigenspace $\mathcal{A}_1$ (in step 2 of Algorithm 3.2) and the solution (2.1) (in step 5 of Algorithm 3.2) in section 4.

3. The minimization problem in step 5 represents the projected and reduced problem of (2.1) onto $\mathrm{span}(\widehat{Q}_k)$ via the relation $\boldsymbol{s} = \widehat{Q}_k \boldsymbol{y}$. The projected problem is stated as

$$\boldsymbol{y}_k = \underset{\|\boldsymbol{y}\|_2=1,\ \boldsymbol{y}\in\mathbb{R}^{kb-1}}{\operatorname{argmin}} \left\{ \frac{1}{2}\boldsymbol{y}^{\mathrm{T}} \widehat{Q}_k^{\mathrm{T}} H \widehat{Q}_k \boldsymbol{y} + \boldsymbol{y}^{\mathrm{T}} \widehat{Q}_k^{\mathrm{T}} \boldsymbol{g} \right\},$$

and reformulating it into the form (3.9) follows from (3.6), (3.7), and $\widehat{Q}_k^{\mathrm{T}} \boldsymbol{g} = \|\boldsymbol{g}\|_2 \boldsymbol{e}_1$.

4. As described in section 2.3, problem (3.9) can be rewritten as a TRS problem. Note that $kb - 1 \ll n$ in general, and we then can solve (3.9) using some sophisticated TRS solver.

3.4. LaLoEig(k): A variant of LaLoEig. It is worth mentioning that Algorithm 3.2 alternatively can be implemented as a *projection method*, which iteratively produces a block Krylov subspace $\mathcal{K}_k(A, G_1)$, and then projects the original ELE (1.1) onto $\mathcal{K}_k(A, G_1)$ to formulate a much smaller size ELE. Solving the projected ELE, one then has the kth iteration, which is indeed the best approximation of ELE (1.1) over $\mathcal{K}_k(A, G_1)$. This alternative version is denoted by LaLoEig(k) and summarized in Algorithm 3.3. This version indicates more clearly why our method indeed follows the RR procedure and is an extension of the (block) Lanczos method for the symmetric eigenvalue problem.

4. Convergence analysis. LaLoEig Algorithm 3.2 basically consists of two procedures, i.e., the block Lanczos method for the eigenvalue problem and the Lanczos method for TRS. Next, we provide an analysis on the accuracy for these procedures. Since the behavior of both procedures is already known, we can use this knowledge to shed light on the numerical performance of LaLoEig Algorithm 3.2 and also LaLoEig(k) Algorithm 3.3.

For characterizing the convergence of the Lanczos method for either the eigenvalue problem or TRS, the Chebyshev polynomials play an important role. The kth Chebyshev polynomial of first kind is given by

$$\begin{aligned} \mathscr{T}_k(t) &= \cos(k \arccos t) && \text{for } |t| \le 1 \\ &= \frac{1}{2}\left[\left(t + \sqrt{t^2-1}\right)^k + \left(t + \sqrt{t^2-1}\right)^{-k} \right] && \text{for } |t| \ge 1. \end{aligned}$$

Because of its numerous nice properties, the Chebyshev polynomial plays a critical role in numerical analysis and computations. A distinctive property of $\mathscr{T}_k(t)$ says that $|\mathscr{T}_k(t)| \le 1$ for $|t| \le 1$ and $|\mathscr{T}_k(t)|$ grows extremely fast for $|t| > 1$. A result due to Chebyshev himself (see [8, p. 65]) says that if $p(t)$ is a polynomial of degree no bigger than k and $|p(t)| \le 1$ for $-1 \le t \le 1$, then $|p(t)| \le |\mathscr{T}_k(t)|$ for any t outside $[-1, 1]$.

Let us begin with the block Lanczos method for the eigenvalue problem based upon the recent convergence analysis presented in [31]. Assume that

$$\mathrm{rank}(G_1^{\mathrm{T}} V_1) = \dim(\mathcal{A}_1) = j \le b. \tag{4.1}$$

Define an orthogonal projection P_b onto the eigenspace associated with the first smallest b eigenvalues of A:

$$P_b = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_b][\boldsymbol{v}_1, \ldots, \boldsymbol{v}_b]^{\mathrm{T}}.$$

The assumption (4.1) ensures that there exists a matrix $X_0 \in \mathbb{R}^{n\times j}$ such that [31]

$$\mathrm{span}(X_0) \subseteq \mathrm{span}(G_1) \quad \text{and} \quad P_b X_0 = V_1.$$

With these settings and applying directly [31, Theorem 4.1], we have the following.

THEOREM 4.1. *Under the assumption* (4.1)*, suppose* $G_i \in \mathbb{R}^{n\times b}$ *for* $i = 1, 2, \ldots, k$ *and let* $T_k = Q_k^{\mathrm{T}} A Q_k \in \mathbb{R}^{kb\times kb}$ *be given by* (3.3) *with the orthonormal basis* Q_k *of the Krylov subspace* $\mathcal{K}_k(A, G_1)$*. Let* $(\nu_i, \boldsymbol{r}_i)$ *for* $i = 1, 2, \ldots, kb$ *be the eigenpairs of* T_k *and* $\widetilde{V}_1 = Q_k[\boldsymbol{r}_1, \ldots, \boldsymbol{r}_j] \in \mathbb{R}^{n\times j}$ *be an approximate orthonormal basis for* $\mathcal{A}_1$*. Then*

$$\varepsilon_k \triangleq \|\sin\angle(\mathcal{A}_1, \mathrm{span}(\widetilde{V}_1))\|_2 \le \frac{\varphi\|\tan\angle(\mathcal{A}_1, \mathrm{span}(X_0))\|_2}{\mathscr{T}_{k-1}(1+2\kappa)}, \tag{4.2}$$

where $\angle(\mathcal{A}_1, \mathrm{span}(\widetilde{V}_1))$ *and* $\angle(\mathcal{A}_1, \mathrm{span}(X_0))$ *are defined according to* (1.5)*,*

$$\varphi = 1 + \frac{c}{\nu_{j+1} - \theta_1}\|P_b A(I_n - P_b)\|_2, \quad \kappa = \frac{\theta_{j+1} - \theta_1}{\theta_n - \theta_{j+1}},$$

and c *is a constant between* 1 *and* $\pi/2$*.*

Theorem 4.1 provides a priori bounds and reveals the convergence behavior of the block Lanczos method for achieving an approximate eigenspace $\mathrm{span}(\widetilde{V}_1)$ of $\mathcal{A}_1$. The dominant factor is the Chebyshev polynomial $\mathscr{T}_{k-1}(1+2\kappa)$, which grows extremely fast for a big κ (see [31] for more discussions and numerical examples). Not indicated in Theorem 4.1, we also point out that when the breakdown in the Lanczos process Algorithm 3.1 occurs, it implies that the exact eigenspace $\mathcal{A}_1$ is found [14, 20, 31].

With the help of Lemma 4.1 (see, e.g., [28, Theorem 3.2]), Theorem 4.1 also implies that LaLoEig in Algorithm 3.2 is able to identify the two different situations in Algorithm 2.1. Furthermore, it offers an approximate solution for each one of the cases.

LEMMA 4.1. *For any unit norm* $\boldsymbol{x}$*, we have*

$$\left|\sin\angle(\boldsymbol{x}, \mathcal{A}_1) - \sin\angle(\boldsymbol{x}, \mathrm{span}(\widetilde{V}_1))\right| \le \|\sin\angle(\mathcal{A}_1, \mathrm{span}(\widetilde{V}_1))\|_2. \tag{4.3}$$

Remark 4.1. We now are able to describe the two situations that may occur in LaLoEig Algorithm 3.2.

(i) $\mathcal{A}_1 \cap \mathrm{int}(\mathbb{K}^n) \neq \emptyset$. In this case, by Proposition 2.1, $\sin\angle(\boldsymbol{e}_1, \mathcal{A}_1) < \sqrt{2}/2$. Therefore, under the assumption (4.1), we know by Lemma 4.1 that

$$\sin\angle(\boldsymbol{e}_1, \mathrm{span}(\widetilde{V}_1)) \le \varepsilon_k + \sin\angle(\boldsymbol{e}_1, \mathcal{A}_1),$$

where ε_k is given by (4.2). After k Lanczos steps satisfying

$$\sin\angle(\boldsymbol{e}_1, \mathrm{span}(\widetilde{V}_1)) \le \varepsilon_k + \sin\angle(\boldsymbol{e}_1, \mathcal{A}_1) \le \frac{\sqrt{2}}{2}, \tag{4.4}$$

and by Proposition 2.1 again, LaLoEig Algorithm 3.2 identifies that the solution $\hat{\boldsymbol{x}}$ of ELE (1.1) is characterized by Theorem 2.1(i). Moreover, LaLoEig produces the Ritz vector $Q_k\boldsymbol{r}_1$ or $-Q_k\boldsymbol{r}_1$ (the one in $\mathbb{K}^n$) and the Ritz value ν_1 as the approximates for $\hat{\boldsymbol{x}}$ and $q_A(\hat{\boldsymbol{x}})$, respectively.

(ii) $\mathcal{A}_1 \cap (\mathbb{K}^n \backslash \{\mathbf{0}\}) = \emptyset$. In this case, by Proposition 2.1, $\sin\angle(\boldsymbol{e}_1, \mathcal{A}_1) > \sqrt{2}/2$. Therefore, under the assumption (4.1), we know by Lemma 4.1 that

$$\sin\angle(\boldsymbol{e}_1, \mathrm{span}(\widetilde{V}_1)) \geq \sin\angle(\boldsymbol{e}_1, \mathcal{A}_1) - \varepsilon_k.$$

After k Lanczos steps satisfying

$$\sin\angle(\boldsymbol{e}_1, \mathrm{span}(\widetilde{V}_1)) \geq \sin\angle(\boldsymbol{e}_1, \mathcal{A}_1) - \varepsilon_k > \frac{\sqrt{2}}{2}, \tag{4.5}$$

LaLoEig Algorithm 3.2 identifies that the solution $\hat{\boldsymbol{x}}$ of ELE (1.1) is characterized by Theorem 2.1(ii). In this case, the LTRS method is consequently called to yield an approximate solution of ELE (1.1), whose analysis of the accuracy is summarized in Theorem 4.2.

THEOREM 4.2. *Under the assumptions of Theorem* 4.1 *and additionally* $G_1(:,1) = \boldsymbol{e}_1$, *let* $\hat{\boldsymbol{x}}_k = \frac{\sqrt{2}}{2}\left[\begin{smallmatrix} 1 \\ \widehat{Q}_k \boldsymbol{y}_k \end{smallmatrix}\right]$, *where* $\boldsymbol{y}_k$ *is the solution to* (3.9), *which is the nondegenerate case, and* $\widehat{Q}_k = Q_k(2:n, 2:kb)$ *is defined in* (3.6). *If* $\mathcal{A}_1 \cap (\mathbb{K}^n \backslash \{\mathbf{0}\}) = \emptyset$, *then*

$$0 \leq q_A(\hat{\boldsymbol{x}}_k) - q_A(\hat{\boldsymbol{x}}) \leq 2\|H + \hat{\varrho} I_{n-1}\|_2 \zeta_k^2, \tag{4.6}$$

$$\|\hat{\boldsymbol{x}}_k - \hat{\boldsymbol{x}}\|_2 \leq 2\sqrt{\varkappa}\,\zeta_k, \tag{4.7}$$

where $\hat{\varrho}$ *is the Lagrangian multiplier of* (2.2) *given in Lemma* 2.2,

$$\zeta_k = \min\left\{ \frac{2\|\boldsymbol{g}\|_2 \epsilon_k^{\mathrm{ra}}(\eta)}{\omega_{n-1} - \omega_1}, \frac{1}{\mathscr{T}_{k-1}(\eta)} \right\}, \tag{4.8}$$

$$\epsilon_k^{\mathrm{ra}}(\eta) = \frac{(\eta + \sqrt{\eta^2 - 1}\,)^{2-k}}{\eta^2 - 1}, \quad \eta = \frac{\varkappa + 1}{\varkappa - 1}, \quad \varkappa = \frac{\omega_{n-1} + \hat{\varrho}}{\omega_1 + \hat{\varrho}},$$

and $\omega_1 \leq \cdots \leq \omega_{n-1}$ *are the eigenvalues of* H.

Proof. By Theorem 2.1, the condition $\mathcal{A}_1 \cap (\mathbb{K}^n \backslash \{\mathbf{0}\}) = \emptyset$ implies that the solution $\hat{\boldsymbol{x}}$ of ELE (1.1) is on the boundary of $\mathbb{K}^n$, i.e., $\hat{\boldsymbol{x}} = \frac{\sqrt{2}}{2}\left[\begin{smallmatrix} 1 \\ \hat{\boldsymbol{s}} \end{smallmatrix}\right]$, where $\hat{\boldsymbol{s}}$ is the solution to (2.1) or, equivalently, (2.2). Recall that the block Lanczos process generates $\mathcal{K}_k(A, G_1)$ and Theorem 3.1 shows $\mathcal{K}_{k-1}(H, \boldsymbol{g}) \subseteq \mathrm{span}(\widehat{Q}_k)$. Also, we have mentioned in Remark 3.1 that $\boldsymbol{s}_k = \widehat{Q}_k \boldsymbol{y}_k$ satisfies

$$\boldsymbol{s}_k = \underset{\|\boldsymbol{s}\|_2 = 1,\ \boldsymbol{s} \in \mathrm{span}(\widehat{Q}_k)}{\mathrm{argmin}} \left\{ f(\boldsymbol{s}) \triangleq \frac{1}{2}\boldsymbol{s}^{\mathrm{T}} H \boldsymbol{s} + \boldsymbol{s}^{\mathrm{T}} \boldsymbol{g} \right\}.$$

Therefore,

$$f(\boldsymbol{s}_k) \leq \min_{\|\boldsymbol{s}\|_2 = 1,\ \boldsymbol{s} \in \mathcal{K}_{k-1}(H, \boldsymbol{g})} f(\boldsymbol{s}). \tag{4.9}$$

Let

$$\tilde{\boldsymbol{s}}_k = \underset{\|\boldsymbol{s}\|_2 = 1,\ \boldsymbol{s} \in \mathcal{K}_{k-1}(H, \boldsymbol{g})}{\mathrm{argmin}} f(\boldsymbol{s}).$$

So $\tilde{\boldsymbol{s}}_k$ is indeed the approximate solution to (2.2) obtained from LTRS in the $(k-1)$th iteration, and the convergence result of [55] for LTRS can be applied to get

$$0 \leq f(\tilde{\boldsymbol{s}}_k) - f(\hat{\boldsymbol{s}}) \leq 2\|H + \hat{\varrho} I_{n-1}\|_2 \zeta_k^2,$$

where ζ_k is defined by (4.8). With $\hat{\boldsymbol{x}}_k = \frac{\sqrt{2}}{2}\begin{bmatrix} 1 \\ \widehat{Q}_k \boldsymbol{y}_k \end{bmatrix} = \frac{\sqrt{2}}{2}\begin{bmatrix} 1 \\ \boldsymbol{s}_k \end{bmatrix}$, the inequality (4.6) follows due to

$$0 \le q_A(\boldsymbol{x}_k) - q_A(\hat{\boldsymbol{x}}) = f(\boldsymbol{s}_k) - f(\hat{\boldsymbol{s}}) \le f(\tilde{\boldsymbol{s}}_k) - f(\hat{\boldsymbol{s}}) \le 2\|H + \hat{\varrho} I_{n-1}\|_2 \zeta_k^2.$$

The a priori upper bound (4.7) follows directly from [55]. This completes the proof.□

It is worth pointing out that the above convergence analysis established for LaLoEig Algorithm 3.2 can also be applied to LaLoEig(k) Algorithm 3.3. In particular, these results reveal that after k steps with k satisfying either (4.4) or (4.5), the two possible situations in (3.10) coincide with those in (1.1), and the accuracy of the corresponding approximate solution is well characterized. The breakdown situation in the Lanczos process Algorithm 3.1 is not revealed in bounds of Theorem 4.2 which are a priori. Fortunately, it has been shown [23, 55] that when the breakdown occurs, $\boldsymbol{s}_k = \widehat{Q}_k \boldsymbol{y}_k$ is an exact solution to the associated TRS (2.1) and hence $\hat{\boldsymbol{x}}_k = \frac{\sqrt{2}}{2}\begin{bmatrix} 1 \\ \boldsymbol{s}_k \end{bmatrix}$ is an exact solution to ELE (1.1).

5. Numerical experiments. In this section, we provide preliminary numerical experiments of Algorithm 3.3 in a MATLAB environment (version 7.11, R2010b). For evaluation of its performance, we also report numerical results of Algorithm 2.1 with the MATLAB build-in routine `eigs` in step 1 and sophisticated TRS solvers in step 2. In particular, we choose two TRS solvers for comparison. The first one is a semidefinite programming primal-dual method [18] (denoted by FW), and the other is LSTRS proposed in [39] and based on a formulation of TRS as a parameterized eigenvalue problem. In the MATLAB environment, both FW[3] and LSTRS[4] are available on the internet. All our tests were conducted on a PC under the Windows 7 (64bit) system with Intel Core i5-3230M CPU (2.6 GHz) and 4 GB memory.

We describe briefly the parameters used in each algorithm. For Algorithm 3.3, we set

$$k = 100,\ b = 2,\ G_1(:,1) = \boldsymbol{e}_1,\ G_1(:,2:b) = \texttt{randn}(n, b-1)$$

and used QR factorization to orthogonalize the columns of G_1. In the block Lanczos process (see Algorithm 3.1), we chose the rank-revealing QR factorization [21] to get the orthogonal matrix G_{i+1}, and the modified Gram–Schmidt process was applied to reorthogonalize Z and Q_i. For the projected ELE problem (3.10), we invoked the MATLAB build-in routine `trust`,[5] which is stable and suitable for the small-size TRS. As for FW and LSTRS in step 2 of Algorithm 2.1, we modified some subroutines and parameters so that all solvers can compute a solution within roughly the same accuracy. In particular, we extended FW so that it can accept a matrix-vector multiplication routine instead of the Hessian matrix, and set the duality gap tolerance as $dgaptol = 10^{-8}$. For the eigensolver called inside LSTRS, the MATLAB build-in routines `eig` and `eigs` are employed for dense matrices and sparse matrices, respectively, where default options are used.

To compare the accuracy of solutions computed by different solvers, we adopted the error of the related KKT system (1.3) as a measure of the quality of the computed solution. Specially, the total error E_{total} consists of three parts: $E_{\boldsymbol{x}}$ corresponding to

[3] FW is available at http://www.math.uwaterloo.ca/~hwolkowi/henry/software/trustreg.d/.

[4] LSTRS is available at http://ta.twi.tudelft.nl/wagm/users/rojas/lstrs.html.

[5] The built-in MATLAB routine `trust` is available in MATLAB 7.0 (R14). `trust` can be used for small- to medium-size trust-region subproblems because it solves the related secular equation in which the full eigendecomposition of the coefficient matrix is computed.

the (relative) feasibility of $\boldsymbol{x} \in \mathbb{K}^n$, $E_{\boldsymbol{y}}$ corresponding to the (relative) feasibility of

$$\boldsymbol{y} = \frac{A\boldsymbol{x} - \lambda \boldsymbol{x}}{\|A\boldsymbol{x} - \lambda \boldsymbol{x}\|_2} \in \mathbb{K}^n, \tag{5.1}$$

and E_c corresponding to the complementarity $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} = 0$, where $\boldsymbol{x}$ is a computed solution and $\lambda = \boldsymbol{x}^{\mathrm{T}} A \boldsymbol{x}$; that is,

$$\begin{aligned} E_{total} &= E_{\boldsymbol{x}} + E_{\boldsymbol{y}} + E_c \\ &= \max(0, |\boldsymbol{x}(1)| - \|\boldsymbol{x}(2:n)\|_2) + \max(0, |\boldsymbol{y}(1)| - \|\boldsymbol{y}(2:n)\|_2) + |\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}|. \end{aligned}$$

We remark that the vector $\boldsymbol{y}$ (5.1) in $E_{\boldsymbol{y}}$ is so defined because the solutions of our test problems fall in the case (ii) of Theorem 2.1. The situation (i) of Theorem 2.1 is of little interest to us in the numerical testing because it is exactly the traditional extreme eigenvalue problem and does not reflect the specific feature of ELE (3.10).

5.1. Performance on random dense matrices. We first test random matrices generated by the MATLAB build-in function `randn` for two specific types of dense matrices:

(I) $H = G + G^{\mathrm{T}}$, where $G =$ `randn`(n),
(II) $H = GG^{\mathrm{T}} - I_n$, where $G =$ `randn`(n),

with the size n varying from 1000 to 3000. We remark that in type (II), the matrix $-I_n$ is added to GG^{T} so that the resulting H is not positive definite.

In our numerical testing, for each dimension n of types (I) and (II), we generated 10 random cases. All algorithms solved these problems successfully, and we present the average performance of the CPU time in seconds and the accuracy for each method. The detailed numerical results are reported in Table 5.1, where $t(s)$ stands for the executing CPU time in seconds.

TABLE 5.1
Numerical results on random dense matrices.

n	$H = G + G^{\mathrm{T}}$						$H = GG^{\mathrm{T}} - I_n$					
	LaLoEig		FW		LSTRS		LaLoEig		FW		LSTRS	
	E_{total}	t(s)	E_{total}	t(s)	E_{total}	t(s)	E_{total}	t(s)	E_{total}	t(s)	E_{total}	t(s)
1000	2.24E-13	0.2	7.41E-12	1.04	3.76E-05	9.6	4.67E-13	0.2	3.05E-11	8.3	2.94E-05	1.6
1200	4.59E-13	0.3	4.11E-12	1.36	1.53E-05	17.8	2.16E-12	0.3	9.85E-12	11.5	3.69E-05	2.8
1400	1.75E-12	0.3	9.10E-12	1.88	2.34E-05	32.2	7.02E-13	0.3	3.52E-11	15.2	3.59E-05	43.9
1600	6.65E-12	0.4	5.39E-12	2.47	1.54E-05	44.4	1.00E-12	0.4	2.07E-11	19.0	5.45E-05	66.2
1800	7.22E-12	0.4	8.09E-12	3.15	1.15E-05	66.5	1.35E-12	0.4	2.76E-11	23.4	4.10E-05	86.2
2000	9.40E-12	0.5	1.00E-11	3.90	1.11E-05	95.5	7.22E-13	0.5	1.36E-11	29.1	4.22E-05	120.7
2200	6.56E-12	0.6	6.21E-12	5.11	1.06E-05	117.8	4.22E-13	0.6	3.00E-11	34.3	2.75E-05	166.1
2400	6.61E-12	0.7	1.72E-11	5.79	4.00E-05	164.7	4.91E-13	0.7	4.80E-11	40.7	3.27E-05	451.8
2600	4.35E-12	0.8	6.50E-12	7.31	1.41E-05	215.3	1.76E-12	0.8	2.89E-11	48.3	5.04E-05	262.4
2800	5.52E-12	0.8	2.71E-12	8.08	9.16E-06	253.5	1.18E-12	0.9	1.46E-11	54.2	3.12E-05	323.8
3000	4.34E-12	0.9	7.55E-12	9.10	1.44E-05	355.3	8.23E-13	0.9	3.01E-11	60.9	4.53E-05	390.5

The numerical results displayed in Table 5.1 indicate that both FW and LSTRS converge much faster for the problems of type (I) than for those of type (II). The reason behind this performance is that the condition numbers (approximately $O(10^6)$) of matrices of type (II) are larger than those (approximately $O(10^3)$) of matrices of type (I). On the contrary, there is no significant increase of CPU time for our algorithm LaLoEig. Overall, Table 5.1 shows that our algorithm is efficient for ELE (1.1) and outperforms the other two solvers in terms of the accuracy of the computed solution and computational costs.

5.2. Performance on sparse matrices. In this subsection, we test the performance of our algorithm on some symmetric sparse problems taken from the University of Florida sparse matrix collection [13] with no particular preference in the selections. Table 5.2 gives 20 test matrices and their corresponding characteristics, where n is the size of the matrix, nnz is the number of nonzero entries, nnz/n^2 is the sparsity, and *cond* is the condition number obtained by the MATLAB routine `condest`.

TABLE 5.2
Test sparse matrices.

Matrix	n	nnz	Sparsity(%)	*cond*	Kind
Dubcova1	16129	253009	0.10%	2.62E+03	2D/3D problem
EX6	6545	295680	0.69%	1.79E+18	combinatorial problem
OPF_3754	15435	141478	0.06%	2.99E+09	power network
PGPgiantcompo	10680	48632	0.04%	6.55E+04	undirected multigraph
Pres_Poisson	14822	715804	0.33%	3.20E+06	computational fluid dynamics
Si5H12	19896	738598	0.19%	3.00E+04	theoretical/quantum chemistry
ca-AstroPh	18772	396160	0.11%	6.55E+04	undirected graph
ca-HepPh	12008	237010	0.16%	6.55E+04	undirected graph
cvxqp3	17500	114962	0.04%	2.17E+16	optimization problem
flowmeter0	9669	67391	0.07%	2.71E+07	model reduction problem
fv1	9604	85264	0.09%	1.28E+01	2D/3D problem
fxm3_6	5026	94026	0.37%	6.55E+04	optimization problem
man_5976	5976	225046	0.63%	6.55E+04	structural problem
nd3k	9000	3279690	4.05%	5.95E+07	2D/3D problem
nemeth01	9506	725054	0.80%	3.80E+02	theoretical/quantum chemistry
net25	9520	401200	0.44%	6.55E+04	optimization problem
rajat06	10922	46983	0.04%	2.23E+05	circuit simulation problem
ramage02	16830	2866352	1.01%	6.55E+04	computational fluid dynamics
stokes64s	12546	140034	0.09%	1.20E+18	computational fluid dynamics
t2dah	11445	176117	0.13%	1.26E+17	model reduction problem

TABLE 5.3
Numerical results on sparse matrices.

Matrix	LaLoEig		FW		LSTRS	
	E_{total}	t(s)	E_{total}	t(s)	E_{total}	t(s)
Dubcova1	4.91E-12	1.2	6.89E-14	2.9	4.17E-06	1.8
EX6	4.11E-13	0.6	3.24E-14	3.7	1.09E-07	2.7
OPF_3754	9.84E-16	1.1	9.42E-11	0.5	*	*
PGPgiantcompo	2.22E-16	0.8	2.73E-14	0.5	1.16E-04	0.3
Pres_Poisson	9.58E-06	1.2	4.52E-01	453.3	1.13E-04	58.7
Si5H12	1.67E-15	1.6	5.41E-11	4.7	5.84E-09	5.1
ca-AstroPh	3.87E-09	1.4	8.79E-14	2.4	1.59E-04	0.9
ca-HepPh	1.39E-14	0.9	3.01E-14	0.7	1.47E-07	0.6
cvxqp3	3.96E-13	1.2	1.14E-12	112.0	9.77E-06	41.0
flowmeter0	2.71E-15	0.8	3.28E-10	0.2	3.20E-05	0.2
fv1	4.74E-10	0.7	1.54E-14	4.9	9.60E-06	1.0
fxm3_6	3.33E-16	0.4	1.38E-12	3.6	2.00E-07	3.4
man_5976	2.22E-12	0.6	6.25E-11	2.5	6.56E-05	0.8
nd3k	3.62E-12	1.7	8.30E-08	217.4	6.77E-05	91.1
nemeth01	1.05E-10	0.9	3.52E-01	130.4	2.05E-04	29.2
net25	6.85E-11	0.9	7.37E-14	0.6	2.65E-07	0.3
rajat06	2.15E-14	0.8	1.24E-11	1.2	7.77E-05	1.0
ramage02	8.42E-11	2.0	1.19E-14	5.7	7.55E+00	5.6
stokes64s	5.50E-09	1.0	3.06E-01	216.4	4.40E-06	22.5
t2dah	2.16E-15	0.9	4.90E-14	0.2	*	*

The performance of the three algorithms for this set of test problems is summarized in Table 5.3, where "$*$" means that LSTRS fails due to the call of `eigs` within

the iteration. The numerical results displayed in this table are consistent with the conclusions drawn from Table 5.1. In particular, Algorithm 3.3 generally has a stable performance in terms of the speed and the accuracy of the computed solution.

A much clearer demonstration of the results in Table 5.1 is through the performance profiles proposed by Dolan and Moré [15]. In particular, suppose that ϑ denotes one of the three tested algorithms, and Ω stands for the set consisting of 20 problems listed in Table 5.2. In terms of the executing CPU time, for a particular algorithm ϑ and a test problem $\varpi \in \Omega$, we can compute $\varsigma = \log_2(\frac{t(\vartheta,\varpi)}{\text{best}t(\varpi)})$, where $t(\vartheta, \varpi)$ represents the CPU time that the algorithm ϑ uses for solving the problem ϖ and "best $t(\varpi)$" means the smallest CPU time among the three algorithms. Note that the value ς implies that for the test problem ϖ, the solver ϑ is roughly at worst 2^{ς} times slower than the best in terms of executing CPU time. In the left figure in Figure 5.1, we plot the curve

$$y_\vartheta(x) = \frac{1}{20} \times \text{size}\left\{\varpi \in \Omega : \log_2\left(\frac{t(\vartheta,\varpi)}{\text{best } t(\varpi)}\right) \le x\right\}$$

with respect to x for three algorithms. Analogously, the right figure in Figure 5.1 is the performance profile for the accuracy. Both performance profiles demonstrate the efficiency of the algorithm LaLoEig on the test problems in Table 5.2.

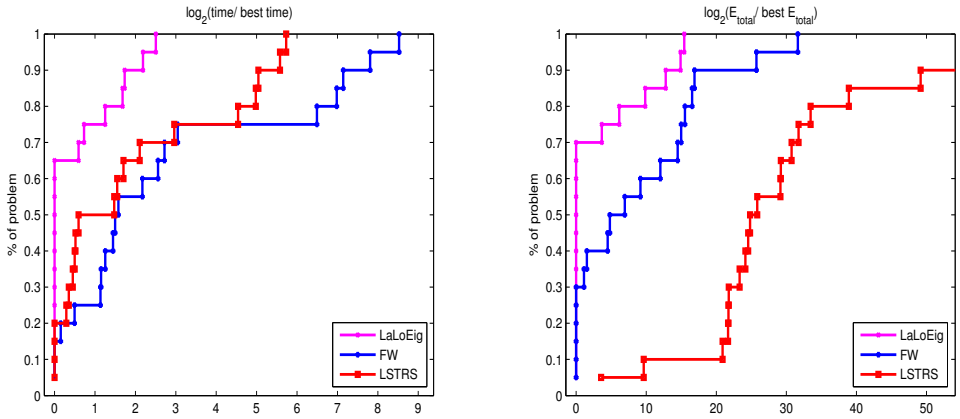

FIG. 5.1. *Performance profile for* CPU time *(left) and the accuracy (right).*

6. Concluding remarks. In this paper, we have numerically treated a special Lorentz eigenvalue problem, namely solving the extreme Lorentz eigenvalue problem (ELE). This problem is intimately related to that of testing the Lorentz-copositivity of the given matrix A. We developed our method by first breaking down ELE into two basic computational procedures: the extreme eigenvalue problem of A and the trust-region subproblem, both of which can be tackled within a Rayleigh–Ritz framework; our numerical scheme then effectively takes advantage of the choice of the initial vector in the Lanczos process and finds an approximate solution of ELE using a single Krylov subspace. The convergence behavior is discussed in theory, and preliminary numerical results on dense and sparse matrices are reported and show the efficiency for solving ELE.

Our development for ELE in this paper is an example of applying the Krylov subspace and Rayleigh–Ritz framework to solve the cone-constrained eigenvalue problem.

Since the Krylov subspace–type method usually is efficient for finding a specific set of eigenpairs, two of our future topics include developing Lanczos methods for Lorentz eigenvalue problems over multiple Lorentz cones [16] and for Lorentz quadratic eigenvalue problems [5].

Acknowledgments. The authors are grateful to Associate Editor Daniel Boley and the referees for their careful reading and very useful comments and suggestions, all of which led to a significant improvement of the paper.

REFERENCES

[1] S. Adly and H. Rammal, *A new method for solving second-order cone eigenvalue complementarity problems*, J. Optim. Theory Appl., 165 (2015), pp. 563–585, https://doi.org/10.1007/s10957-014-0645-0.

[2] S. Adly and A. Seeger, *A nonsmooth algorithm for cone-constrained eigenvalue problems*, Comput. Optim. Appl., 49 (2011), pp. 299–318, https://doi.org/10.1007/s10589-009-9297-7.

[3] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, eds., *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, Software Environ. Tools 11, SIAM, Philadelphia, 2000, https://doi.org/10.1137/1.9780898719581.

[4] Z. Bai and R. W. Freund, *A symmetric band Lanczos process based on coupled recurrences and some applications*, SIAM J. Sci. Comput., 23 (2001), pp. 542–562, https://doi.org/10.1137/S1064827500371773.

[5] C. P. Brás, M. Fukushima, A. N. Iusem, and J. J. Júdice, *On the quadratic eigenvalue complementarity problem over a general convex cone*, Appl. Math. Comput., 271 (2015), pp. 594–608, https://doi.org/10.1016/j.amc.2015.09.014.

[6] C. P. Brás, M. Fukushima, J. J. Júdice, and S. S. Rosa, *Variational inequality formulation of the asymmetric eigenvalue complementarity problem and its solution by means of gap functions*, Pacific J. Optim., 8 (2011), pp. 197–215.

[7] J.-S. Chen and S. H. Pan, *Semismooth Newton methods for the cone spectrum of linear transformations relative to Lorentz cones*, Linear Nonlinear Anal., 1 (2015), pp. 13–36.

[8] E. W. Cheney, *Introduction to Approximation Theory*, 2nd ed., Chelsea Publishing Company, New York, 1982.

[9] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust-Region Methods*, MOS–SIAM Ser. Optim. 1, SIAM, Philadelphia, 2000, https://doi.org/10.1137/1.9780898719857.

[10] J. K. Cullum and W. E. Donath, *A block Lanczos algorithm for computing the q algebraically largest eigenvalues and a corresponding eigenspace of large, sparse, real symmetric matrices*, in Decision and Control Including the 13th Symposium on Adaptive Processes, IEEE, Washington, DC, 1974, pp. 505–509.

[11] A. Pinto da Costa, J. A. C. Martins, I. N. Figueiredo, and J. J. Júdice, *The directional instability problem in systems with frictional contacts*, Comput. Methods Appl. Mech. Engrg., 193 (2004), pp. 357–384, https://doi.org/10.1016/j.cma.2003.09.013.

[12] A. Pinto da Costa and A. Seeger, *Cone-constrained eigenvalue problems: Theory and algorithms*, Comput. Optim. Appl., 45 (2010), pp. 25–57, https://doi.org/10.1007/s10589-008-9167-8.

[13] T. Davis and Y. Hu, *The University of Florida sparse matrix collection*, ACM Trans. Math. Software, 38 (2011), pp. 1–25, https://doi.org/10.1145/2049662.2049663.

[14] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997, https://doi.org/10.1137/1.9781611971446.

[15] E. D. Dolan and J. Moré, *Benchmarking optimization software with performance profiles*, Math. Program., 91 (2002), pp. 201–213, https://doi.org/10.1007/s101070100263.

[16] L. M. Fernandes, M. Fukushima, J. J. Júdice, and H. D. Sherali, *The second-order cone eigenvalue complementarity problem*, Optim. Methods Softw., 31 (2016), pp. 24–52, https://doi.org/10.1080/10556788.2015.1040156.

[17] L. M. Fernandes, J. J. Júdice, H. D. Sherali, and M. Fukushima, *On the computation of all eigenvalues for the eigenvalue complementarity problem*, J. Global Optim., 59 (2014), pp. 307–326, https://doi.org/10.1007/s10898-014-0165-3.

[18] C. Fortin and H. Wolkowicz, *The trust region subproblem and semidefinite programming*, Optim. Methods Softw., 19 (2004), pp. 41–67, https://doi.org/10.1080/10556780410001647186.

[19] P. Gajardo and A. Seeger, *Solving inverse cone-constrained eigenvalue problems*, Numer. Math., 123 (2013), pp. 309–331.
[20] G. H. Golub and R. R. Underwood, *The block Lanczos method for computing eigenvalues*, in Mathematical Software III, J. R. Rice, ed., Academic Press, New York, 1977, pp. 361–377.
[21] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
[22] G. H. Golub and U. von Matt, *Quadratically constrained least squares and quadratic problems*, Numer. Math., 59 (1991), pp. 561–580, https://doi.org/10.1007/BF01385796.
[23] N. I. M. Gould, S. Lucidi, M. Roma, and P. L. Toint, *Solving the trust-region subproblem using the Lanczos method*, SIAM J. Optim., 9 (1999), pp. 504–525, https://doi.org/10.1137/S1052623497322735.
[24] W. W. Hager, *Minimizing a quadratic over a sphere*, SIAM J. Optim., 12 (2001), pp. 188–208, https://doi.org/10.1137/S1052623499356071.
[25] J.-B. Hiriart-Urruty and A. Seeger, *A variational approach to copositive matrices*, SIAM Rev., 52 (2010), pp. 593–629, https://doi.org/10.1137/090750391.
[26] J. Júdice, M. Raydan, S. S. Rosa, and S. A. Santos, *On the solution of the symmetric eigenvalue complementarity problem by the spectral projected gradient algorithm*, Numer. Algorithms, 47 (2008), pp. 391–407, https://doi.org/10.1090/S0025-5718-09-02258-3.
[27] N. Kabadi and K. G. Murty, *Some NP-complete problems in quadratic and nonlinear programming*, Math. Program., 39 (1987), pp. 117–129, https://doi.org/10.1007/BF02592948.
[28] A. V. Knyazev and M. E. Argentati, *Majorization for changes in angles between subspaces, Ritz values, and graph Laplacian spectra*, SIAM J. Matrix Anal. Appl., 29 (2006), pp. 15–32, https://doi.org/10.1137/060649070.
[29] D. Kressner, M. M. Pandur, and M. Shao, *An indefinite variant of LOBPCG for definite matrix pencils*, Numer. Algorithms, 66 (2014), pp. 681–703, https://doi.org/10.1007/s11075-013-9754-3.
[30] R.-C. Li, *Sharpness in rates of convergence for symmetric Lanczos method*, Math. Comp., 79 (2010), pp. 419–435, https://doi.org/10.1090/S0025-5718-09-02258-3.
[31] R.-C. Li and L.-H. Zhang, *Convergence of block Lanczos method for eigenvalue clusters*, Numer. Math., 131 (2015), pp. 83–113, https://doi.org/10.1007/s00211-014-0681-6.
[32] R. Loewy, *Positive operators on the n-dimensional ice cream cone*, J. Math. Anal. Appl., 49 (1975), pp. 375–392.
[33] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 553–572, https://doi.org/10.1137/0904038.
[34] Y. S. Niu, T. Pham Dinh, H. A. Le Thi, and J. J. Júdice, *Efficient DC programming approaches for the asymmetric eigenvalue complementarity problem*, Optim. Methods Softw., 28 (2013), pp. 812–829, https://doi.org/10.1080/10556788.2011.645543.
[35] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd ed., Springer, New York, 2006, https://doi.org/10.1007/b98874.
[36] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Classics Appl. Math. 20, SIAM, Philadelphia, 1998, https://doi.org/10.1137/1.9781611971163.
[37] R. Rendl and H. Wolkowicz, *A semidefinite framework for trust region subproblems with applications to large scale minimization*, Math. Program., 77 (1997), pp. 273–299, https://doi.org/10.1007/BF02614438.
[38] M. Rojas, S. A. Santos, and D. C. Sorensen, *A new matrix-free algorithm for the large-scale trust-region subproblem*, SIAM J. Optim., 11 (2000), pp. 611–646, https://doi.org/10.1137/S105262349928887X.
[39] M. Rojas, S. A. Santos, and D. C. Sorensen, *Algorithm* 873: *LSTRS: MATLAB software for large-scale trust-region subproblems and regularization*, ACM Trans. Math. Software, 34 (2008), pp. 1–28, https://doi.org/10.1145/1326548.1326553.
[40] M. Rojas and D. C. Sorensen, *A trust-region approach to the regularization of large-scale discrete forms of ill-posed problems*, SIAM J. Sci. Comput., 23 (2002), pp. 1842–1860, https://doi.org/10.1137/S1064827500378167.
[41] A. Ruhe, *Implementation aspects of band Lanczos algorithms for computation of eigenvalues of large sparse symmetric matrices*, Math. Comp., 33 (1979), pp. 680–687.
[42] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, Manchester, UK, 1992.
[43] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003, https://doi.org/10.1137/1.9780898718003.
[44] A. Seeger, *Eigenvalue analysis of equilibrium processes defined by linear complementarity conditions*, Linear Algebra Appl., 292 (1999), pp. 1–14, https://doi.org/10.1016/S0024-3795(99)00004-X.

[45] A. Seeger and D. Sossa, *Critical angles between two convex cones* II. *Special cases*, TOP, 24 (2016), pp. 66–87, https://doi.org/10.1007/s11750-015-0382-z.
[46] A. Seeger and M. Torki, *On eigenvalues induced by a cone constraint*, Linear Algebra Appl., 372 (2003), pp. 181–206, https://doi.org/10.1016/S0024-3795(03)00553-6.
[47] D. C. Sorensen, *Newton's method with a model trust region modification*, SIAM J. Numer. Anal., 19 (1982), pp. 409–426, https://doi.org/10.1137/0719026.
[48] D. C. Sorensen, *Minimization of a large-scale quadratic function subject to a spherical constraint*, SIAM J. Optim., 7 (1997), pp. 141–161, https://doi.org/10.1137/S1052623494274374.
[49] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637, https://doi.org/10.1137/0720042.
[50] G. W. Stewart, *Matrix Algorithms, Vol.* II: *Eigensystems*, SIAM, Philadelphia, 2001, https://doi.org/10.1137/1.9780898718058.
[51] A. Tarantola, *Inverse Problem Theory*, Elsevier, Amsterdam, 1987.
[52] H. A. L. Thi, M. Moeini, T. Pham Dinh, and J. Júdice, *A DC programming approach for solving the symmetric eigenvalue complementarity problem*, Comput. Optim. Appl., 51 (2012), pp. 1097–1117, https://doi.org/10.1007/s10589-010-9388-5.
[53] P. L. Toint, *Towards an efficient sparsity exploiting Newton method for minimization*, in Sparse Matrices and Their Uses, I. S. Duff, ed., Academic Press, London, 1981, pp. 57–88.
[54] Q. Ye, *An adaptive block Lanczos algorithm*, Numer. Algorithms, 12 (1996), pp. 97–110, https://doi.org/10.1007/BF02141743.
[55] L.-H. Zhang, C. Shen, and R.-C. Li, *On the generalized Lanczos trust-region method*, SIAM J. Optim., 27 (2017), pp. 2110–2142, https://doi.org/10.1137/16M1095056.