LAGRANGE MULTIPLIERS AND RAYLEIGH QUOTIENT ITERATION IN CONSTRAINED TYPE EQUATIONS

DU NGUYEN
NGUYENDU@POST.HARVARD.EDU

Abstract. We generalize the Rayleigh quotient iteration to a class of functions called vector Lagrangians. Our generalized Rayleigh quotient is an expression often used as an estimate of the Lagrange multiplier in constrained optimization. We discuss two methods of solving the updating equation associated with the iteration. One method leads to a generalization of Riemannian Newton method for embedded manifolds in a Euclidean space while the other leads to a generalization of the classical Rayleigh quotient formula and its invariant subspace extension. We also obtain a Rayleigh-Chebyshev iteration with cubic convergent rate. We discuss applications of this result to linear and nonlinear eigenvalue and subspace problems as well as potential applications in optimization.

Key words. Lagrange multiplier, Rayleigh quotient, Newton-Raphson, Eigenvalue, Invariant subspace, Optimization

AMS subject classifications. 65K10, 65F10, 65F15

1. Introduction. Consider three Euclidean spaces E_{in}, E_{out}, E_L with $\dim(E_{in}) = \dim(E_{out})$. We consider a map $\boldsymbol{L} : (\boldsymbol{x}, \boldsymbol{\lambda}) \mapsto \boldsymbol{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ from $E_{in} \oplus E_L$ into E_{out} and a map $\boldsymbol{C} : \boldsymbol{x} \mapsto \boldsymbol{C}(\boldsymbol{x})$ from E_{in} to E_L. The direct sum $\mathcal{L} = \boldsymbol{L} \oplus \boldsymbol{C}$:

$$\mathcal{L}\begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{\lambda}\end{pmatrix} = \begin{pmatrix}\boldsymbol{L}(\boldsymbol{x},\boldsymbol{\lambda})\\ \boldsymbol{C}(\boldsymbol{x})\end{pmatrix} \tag{1.1}$$

is a map from $E_{in} \oplus E_L$ to $E_{out} \oplus E_L$. When the Jacobian of $\mathcal{L}$ is invertible in a domain of $E_{in} \oplus E_L$ near a root of $\mathcal{L}$, $\boldsymbol{L}$ and $\boldsymbol{C}$ have Jacobians both of full row rank. In that situation we will call $\boldsymbol{L}$ a *vector* Lagrangian and $\boldsymbol{C}$ a constraint. The equation

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = 0 \tag{1.2}$$

covers systems of equations where a number of equations in the system are dependent on a group of variables $\boldsymbol{x}$ while the remaining equations involve all variables $(\boldsymbol{x}, \boldsymbol{\lambda})$. The remaining variables are named $\boldsymbol{\lambda}$ in honor of Lagrange. The full row rank assumption for $\boldsymbol{L}$ ensures that $\boldsymbol{\lambda}$ is solvable in $\boldsymbol{x}$ (but not always explicitly). For the rest of this article we will refer to Lagrangians dropping the qualifier *vector* except when it could cause confusion, see below.

Of particular interest is the case when $\boldsymbol{L}$ is affine in $\boldsymbol{\lambda}$. We define an *explicit* Lagrangian to be a Lagrangian of the form

$$\boldsymbol{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = \boldsymbol{F}(x) - \boldsymbol{H}(x)\boldsymbol{\lambda} \tag{1.3}$$

where $\boldsymbol{F}$ is a vector function from an open set in E_{in} to E_{out} and $\boldsymbol{H}(\boldsymbol{x})$ is a linear map from E_L to E_{out} for each x. In the code we refer to Lagrangians not necessarily of explicit form as implicit Lagrangians. As this may cause confusion, we refrain from using this term in the article.

On the constraint side, from our full row rank assumption $\boldsymbol{C}$ defines a submanifold $\mathcal{M} = \boldsymbol{C}^{-1}(0)$ of codimension $\operatorname{rank}(\boldsymbol{J}_{\boldsymbol{C}}(\boldsymbol{x}))$.

This setup covers three classes of equations encountered in the literature:

- The eigenvector/invariant subspace problem:

$$\begin{aligned}\boldsymbol{F} &= A\boldsymbol{x}\\ \boldsymbol{H}(\boldsymbol{x}) &= \boldsymbol{x}\\ \boldsymbol{C}(\boldsymbol{x}) &= \frac{1}{2}(\boldsymbol{x}^T\boldsymbol{x} - \boldsymbol{I}_{E_L})\end{aligned} \tag{1.4}$$

  where $\boldsymbol{x}$ is an $n \times k$ matrix, A is an $n \times n$ matrix. In this case E_{in} and E_{out} are both (n, k) matrices and E_L is the space of symmetric $k \times k$ matrices. The case where $k = 1$ is the eigenvector problem.

- The constraint optimization problem, one of the most important problems in applied mathematics. Here $\boldsymbol{F} = \nabla f$ where f is a real value function and $\boldsymbol{H} = \boldsymbol{J}_{\boldsymbol{C}}^T$. This is the case of the classical Lagrangian multiplier equations. $E_{in} = E_{out}$ is the domain where f is defined and E_L is the target space of the restrictions $\boldsymbol{C}(\boldsymbol{x})$ on $\boldsymbol{x}$. The system (1.1) gives us the set of critical points. In this case our (*vector*) Lagrangian is the differential of the classical scalar Lagrangian $f - \boldsymbol{C}^T\boldsymbol{\lambda}$.
- The nonlinear eigenvalue problem:

$$P(\lambda)\boldsymbol{x} = 0 \tag{1.5}$$

  Here P is a matrix with polynomial entries in λ. While this is not in the form (1.1) we can impose the constraint $\boldsymbol{C}(x) = \boldsymbol{x}^T\boldsymbol{x} - 1$ (or $\boldsymbol{C}(x) = \boldsymbol{z}^T\boldsymbol{x} - 1$ for a fixed vector $\boldsymbol{z}$) This equation is not of explicit form. E_L is of dimension one and λ is a scalar. E_{in} and E_{out} are $\mathbb{R}^k$ where k is the dimension of $\boldsymbol{x}$. There is an extensive literature for this problem ([12]).

An iteration method called Rayleigh quotient iteration (RQI) is among the most powerful methods to compute eigenvalues and vectors. For a vector $\boldsymbol{v}$ the Rayleigh quotient is

$$\boldsymbol{\lambda} = \frac{\boldsymbol{v}^T A\boldsymbol{v}}{\boldsymbol{v}^T\boldsymbol{v}}$$

(see [1], algorithm 2.2. The quotient reduces to $\boldsymbol{v}^T A\boldsymbol{v}$ on the unit sphere.) The iteration computes

$$\boldsymbol{v}_{i+1} = \frac{(A - \boldsymbol{\lambda})^{-1}\boldsymbol{v}_i}{||(A - \boldsymbol{\lambda})^{-1}\boldsymbol{v}_i||}$$

which is shown to have cubic convergence in for almost all initial points if A is normal and quadratic otherwise on suitable initial points. We call the equation of form

$$(A - \boldsymbol{\lambda})\boldsymbol{x} = \boldsymbol{w}$$

a resolvent equation, as $(A - \boldsymbol{\lambda})^{-1}$ is called the resolvent in the literature. The letter $\boldsymbol{\lambda}$ used in the second example in honor of Lagrange is also often used to denote an eigenvalue, a fact we will show is a more than a happy coincident.

Let us summarize what is known about RQI at present. The fact that RQI is related to Lagrange multipliers and Newton-Raphson has been known for a long time but it is probably fair to say the relationship is not yet transparent. [8] suggested RQI is an approximation of Riemannian Newton, which is justified when ([2]) showed we can retract by a retraction more general than geodesic transport. In this case the retraction is the projection to the unit sphere. [1] gives an extension of Rayleigh quotient iteration for the invariant subspace problem where $\boldsymbol{x}$ is an $n \times k$-matrix and $\boldsymbol{\lambda}$ is a $k \times k$-matrix. The resolvent equation $(A - \boldsymbol{\lambda})\boldsymbol{v}_{i+1} = \boldsymbol{v}_i$ is replaced by a Sylvester equation. There are RQIs for nonnormal matrices and also for nonlinear eigenvector problems. But so far it seems RQI is specific to certain type of matrix problems. On the rate of convergence, it is known that we have quadratic convergence in general case, and cubic convergence for normal matrices. For nonnormal matrices we have the two-sided RQI which also has cubic convergence. However, proofs of cubic convergence seem to be specific to each instance and there is no general framework. For a general framework we would want have a generalization of $\boldsymbol{\lambda}$, of the resolvent equation and an analysis of convergent rate, including a criterion for cubic convergence.

In this paper we provide this framework by extending Rayleigh quotient iteration to all Lagrangians:
- We show that RQI is a simple variance of Newton-Raphson iteration for (1.1). When the Lagrangian is of explicit form the updating equation for $\boldsymbol{x}$ is a linear equation generalizing the resolvent equation and $\boldsymbol{\lambda}$ is calculated by an expression extending the classical Rayleigh quotient.
- When the Lagrangian is of general form, we will need to solve for $\boldsymbol{\lambda}$ in term of $\boldsymbol{x}$. If this solution is consistent (see below) we can construct a RQI. This RQI covers the case of nonlinear eigenvalues.

- In general, RQI consists of two steps:
  - Solving for a linear equation from $T\mathcal{M}$ to $\mathrm{Im}(\Pi_{\boldsymbol{L}_{\boldsymbol{\lambda}}})$ (to be defined later) where $\boldsymbol{L}_{\boldsymbol{\lambda}}$ is the partial derivative of $\boldsymbol{L}$ with respect to $\boldsymbol{\lambda}$ ($\boldsymbol{L}_{\boldsymbol{\lambda}} = -\boldsymbol{H}(\boldsymbol{x})$ in the explicit case.)
  - Retracting the solution from $T\mathcal{M}$ to the constraint manifold $\mathcal{M}$.

  If we use the Schur complement technique to solve the linear equation in the first step, the solution is a generalization of the resolvent equation. If we solve this equation by first parameterize $\mathcal{M}$, we get an extension of the Riemannian Newton iteration on $\mathcal{M}$.
- We construct a second order version of RQI called Rayleigh-Chebyshev iteration that has cubic convergence. We also give an explicit criterion for RQI to have cubic convergence. In particular, we have a new cubic convergence algorithm for eigenvectors of nonnormal matrices. We have a similar algorithm for the nonlinear eigenvalue case. This sets a novel framework to study high order iterations on manifolds.
- We provided a detailed analysis as well as open source codes for Newton-Raphson for $\mathcal{L}$ on $E_L \oplus E_{in}$ and reduce it to a form clearly showing its relationship to Riemannian Newton on $\mathcal{M}$.

In practice, E_{in}, E_{out}, E_L could be spaces of matrices or tensors. $\boldsymbol{H}$ is then a tensor, and so are the Jacobians of $\boldsymbol{F}$ and $\boldsymbol{L}$. On the theoretical side we will consider the space involves as vector, leaving the tensor related treatment to specific implementations.

2. Newton-Raphson applying to eigenvector problem. To explain the ideas involved here we look at the eigenvector problem in details. Here

$$\boldsymbol{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = A\boldsymbol{x} - \boldsymbol{x}\boldsymbol{\lambda}$$

and constraint is $\boldsymbol{C}(x) = \frac{1}{2}(\boldsymbol{x}^T\boldsymbol{x} - \boldsymbol{I})$ The Jacobian of $\mathcal{L} = \boldsymbol{L} \oplus \boldsymbol{C}$ is

$$\boldsymbol{J}_{\mathcal{L}} = \begin{pmatrix} \boldsymbol{L}_{\boldsymbol{x}}(\boldsymbol{x}) & -\boldsymbol{x} \\ \boldsymbol{x}^T & 0 \end{pmatrix}$$

where $\boldsymbol{L}_{\boldsymbol{x}}(\boldsymbol{x})(\eta) = A\eta - \eta\boldsymbol{\lambda}$. Applying the Schur complement formula we can evaluate the Newton step:

$$\begin{pmatrix} \eta \\ \delta \end{pmatrix} = \boldsymbol{J}_{\mathcal{L}}^{-1} \begin{pmatrix} -A\boldsymbol{x} + \boldsymbol{x}\boldsymbol{\lambda} \\ -\frac{1}{2}(\boldsymbol{x}^T\boldsymbol{x} - 1) \end{pmatrix}$$

With $\boldsymbol{x} = \boldsymbol{x}_i$ and $\boldsymbol{\lambda} = \boldsymbol{\lambda}_i$ and shorthand $\boldsymbol{L}_{\boldsymbol{x}}$ for $\boldsymbol{L}_{\boldsymbol{x}}(\boldsymbol{x})$ we get:

$$\boldsymbol{\lambda}_{i+1} - \boldsymbol{\lambda}_i = \delta = (\boldsymbol{x}^T\boldsymbol{L}_{\boldsymbol{x}}^{-1}\boldsymbol{x})^{-1}(\boldsymbol{x}^T\boldsymbol{L}_{\boldsymbol{x}}^{-1}(A\boldsymbol{x} - x\boldsymbol{\lambda})) - (\boldsymbol{x}^T\boldsymbol{L}_{\boldsymbol{x}}^{-1}\boldsymbol{x})^{-1}\frac{1}{2}(\boldsymbol{x}^T\boldsymbol{x} - 1)$$

$$\boldsymbol{x}_{i+1} - \boldsymbol{x}_i = \eta = -\boldsymbol{L}_{\boldsymbol{x}}^{-1}(A\boldsymbol{x} - x\boldsymbol{\lambda}) + \boldsymbol{L}_{\boldsymbol{x}}^{-1}\boldsymbol{x}\delta$$

So with $\zeta = \boldsymbol{L}_{\boldsymbol{x}}^{-1}\boldsymbol{x}$ we simplify the updating equations to:

$$\delta = \boldsymbol{\lambda}_{i+1} - \boldsymbol{\lambda}_i = (2\boldsymbol{x}^T\zeta)^{-1}(1 + \boldsymbol{x}^T\boldsymbol{x})$$

$$\eta = -\boldsymbol{x} + \zeta\delta = -\boldsymbol{x} + \zeta(2\boldsymbol{x}^T\zeta)^{-1}(1 + \boldsymbol{x}^T\boldsymbol{x})$$

From here

$$\boldsymbol{x}_{i+1} = \zeta(2\boldsymbol{x}^T\zeta)^{-1}(1 + \boldsymbol{x}_i^T x_i)$$

We see $\boldsymbol{x}_{i+1}$ is proportional to ζ, a result known from classical Rayleigh iteration. The equation for ζ is exactly the resolvent equation. However, the formula for $\boldsymbol{\lambda}_{i+1}$ is iterative. Let us link $\boldsymbol{\lambda}$ with the Rayleigh quotient. Starting with the general equation for the explicit system:

$$\boldsymbol{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = \boldsymbol{F}(\boldsymbol{x}) - \boldsymbol{H}(\boldsymbol{x})\boldsymbol{\lambda} = 0$$

From the full rank assumption $\boldsymbol{H}$ has a left inverse (for example $\boldsymbol{H}^{-} = (\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T$). We can solve for $\boldsymbol{\lambda}$:

$$\boldsymbol{\lambda} = \boldsymbol{H}^{-}\boldsymbol{F}(\boldsymbol{x}) \tag{2.1}$$

In the eigenvector case this is exactly the Rayleigh quotient. This expression appeared very early in the multiplier method literature as discussed below. Our paper started from our attempt to modify Newton-Raphson using this expression for $\boldsymbol{\lambda}$ to solve (1.1).

In the next few sections we discuss a few set-ups needed to state and prove our theorems.

3. Higher derivatives as tensors. The reader can consult [3] for this section. We use slightly different notations in this paper. Recall we can use tensors to denote linear maps between two vector spaces each represented as matrix or tensor. If the domain space is of shape s_1 and the range space is of shape s_2 then a map between them could be represented as a tensor of shape (s_1, s_2). The map sending a tensor η to the tensor $T\eta$ formed by contracting to the right is the linear map represented by T.

If $\boldsymbol{F}$ is a map between two vector spaces $V = \mathbb{R}^n$ and $W = \mathbb{R}^m$ then its Jacobian $\boldsymbol{J_F}$ is a map in the space $L(V, W)$ of linear maps between V and W and is represented by an $m \times n$-matrix.

A second derivative is a linear map between V and $L(V, W) \cong V^*\otimes W$ or an element of $L(V, L(V, W)) \cong V^* \otimes V^* \otimes W$ and can be represented as $m \times n \times n$ tensor. We will denote this map as well as this tensor as $\boldsymbol{J}_F^{(2)}$. In general, we will denote the $l - th$ derivatives as $\boldsymbol{J_F}^{(l)}$ and this is an element of $L(V(L(V, \cdots L(V, W)))$ (with l copies of V and one copy of W. We can represent it as a tensor of size $m \times n \cdots \times n$.

For l vectors $\eta_1, \eta_2, \dots, \eta_l$ consider the tuple

$$[\eta_1, \eta_2, \dots, \eta_l]$$

we can define

$$T[\eta_1\eta_2\dots\eta_l] = (\cdots((T\eta_1)\eta_2)\dots)\eta_l$$

which is the repeated contraction of η_i. We write

$$\eta^{[l]} = [\eta, \eta, \cdots, \eta]$$

l times. With this notation we can write the Taylor series expansion up to order l around $\boldsymbol{v}$ as:

$$\boldsymbol{F}(\boldsymbol{v}) + \boldsymbol{J_F}(\boldsymbol{v})(\boldsymbol{h}) + \frac{1}{2}\boldsymbol{J_F}(\boldsymbol{v})(\boldsymbol{h}^{[2]}) + \cdots \frac{1}{l!}\boldsymbol{J_F}^{(l)}(\boldsymbol{v})(\boldsymbol{h}^{[l]})$$

where $\boldsymbol{h} = \boldsymbol{x} - \boldsymbol{v}$.

To summarize, there are two maps related to higher derivatives. The map $\boldsymbol{x} \mapsto \boldsymbol{J_F}^{(l)}(\boldsymbol{x})$ from V to $V^* \otimes \cdots \otimes V^* \otimes W$ is generally nonlinear resulting in a tensor. For a fixed $\boldsymbol{x}$, that tensor gives a multilinear map acting on the tangent space which is embedded in E_{in}, sending $\boldsymbol{h}$ to $\boldsymbol{J_F}^{(l)}(\boldsymbol{v})(\boldsymbol{h}^{[l]})$. In code, we need two functions for these two maps. The second map is in general just tensor contraction however depending on problem a custom implementation may be useful.

4. Retractions. Consider a submanifold $\mathcal{M}$ of $\mathbb{R}^n$ of class C^k. Recall the definitions of retractions from [2]:

- A first order retraction R is a map from $T\mathcal{M}$ to $\mathcal{M}$ around a point $\bar{x}$ if there exists a neighborhood $\mathcal{U}$ of $(\bar{x}, 0)$ in $T\mathcal{M}$ such that:
  1. $\mathcal{U} \subset \mathrm{dom}(R)$ and the restriction $R : \mathcal{U} \to \mathcal{M}$ is of class C^{k-1}.
  2. $R(x, 0) = 0$ for all $(u, 0) \in \mathcal{U}$
  3. $\boldsymbol{J}_R(x, .) = Id_{T\mathcal{M}}(x) \in \mathcal{U}$
- A second order retraction on $\mathcal{M}$ is a first order retraction on $\mathcal{M}$ that satisfies for all $(x, u) \in T\mathcal{M}$,

$$\frac{d^2}{dt^2}R(x, tu)|_{t=0} \in N_{\mathcal{M}}(x) \tag{4.1}$$

$N_{\mathcal{M}}(x)$ is the normal space at x. The exponent map is a second order retraction. It is shown in that paper that projection and orthographic projections are second order retractions. The following is clear:

PROPOSITION 4.1. *If $\boldsymbol{r}$ is a retraction on M then $\boldsymbol{r} \times Id_{E_L}$ is a retraction on $M \times E_L$, if $\boldsymbol{r}$ is a retraction of second order then $\boldsymbol{r} \times Id_{E_L}$ is a retraction of second order.*

By Id_V we mean the identity map on the space V. From this proposition and the result of [2]), we can retract intermediate iteration points to $M \times E_L$, as a result $\boldsymbol{x}_i$ can be made elements of $\mathcal{M}$ while $\boldsymbol{\lambda}_i$ is unchanged. Following the common literature, we call a Newton-Raphson iteration on $E_{in} \oplus E_L$ with no retraction to $\mathcal{M}$ infeasible start iteration. When there is retraction to $\mathcal{M}$, we have a feasible start iteration. We note Lagrange multiplier with feasible improvement has already been studied, e.g. in [10].

5. Newton-Raphson method for Lagrangians. We first examine the Newton-Raphson iteration. The Jacobian of $\mathcal{L}$ is

$$\boldsymbol{J}_{\mathcal{L}}(\boldsymbol{x}, \boldsymbol{\lambda}) = \begin{pmatrix} \boldsymbol{L}_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{\lambda}) & \boldsymbol{L}_{\boldsymbol{\lambda}}(\boldsymbol{x}, \boldsymbol{\lambda}) \\ \boldsymbol{J}_{\boldsymbol{C}}(\boldsymbol{x}) & 0 \end{pmatrix} \tag{5.1}$$

Instead of writing the full $h(\boldsymbol{x}, \boldsymbol{\lambda})$ we sometime shorthand to h to save space in equations, this applies to $h = \boldsymbol{L}, \boldsymbol{L}_{\boldsymbol{x}} \boldsymbol{L}_{\boldsymbol{\lambda}}, \boldsymbol{J}_{\boldsymbol{C}}$ for example. Newton-Raphson iteration in the framework of constraint optimization has been studied by many authors in the literature, including [9], [10], [11], [17], [18]. We will make the connection between $\boldsymbol{L}_{\boldsymbol{x}}$ with the resolvent equation more explicit, and show $\boldsymbol{\lambda}_i$ converges to the Rayleigh quotient expression. We transform the updating equations to a format closer to one derived from Riemannian Newton optimization which motivates the RQI in the next section.

To invert this Jacobian without inverting the whole matrix, we describe two approaches. One approach would focus on first parameterize η using the equation $\boldsymbol{J}_{\boldsymbol{C}}(\boldsymbol{x})\eta = -\boldsymbol{C}(\boldsymbol{x})$. If the tangent space has an explicit description this would help, otherwise it is parameterized by the null space of $\boldsymbol{J}_{\boldsymbol{C}}$ ($\eta = \eta_0 + Z\alpha$ where Z is a basis of the null space and α's are the new parameters). This will help reduce the number of variables to $\dim(E_{in}) - \dim(E_L)$. Substitute back to the first equation we get a system of $\dim(E_{out})$ equations and $\dim(E_{in})$ variables ($\dim(E_{in}) - \dim(E_L)$ from α and $\dim(E_L)$ from δ) which we can solve. This works in the most general case, including the case where $\boldsymbol{L}_{\boldsymbol{x}}$ is not invertible and is closely related to the Riemannian Newton approach on the embedded manifold $\mathcal{M}$. We present a second approach, already investigated in [17], [18] where we assume $\boldsymbol{L}_{\boldsymbol{x}}$ is invertible. This approach reveals some interesting relations mentioned above. As we use the Schur complement formula in this second approach, we will call this solution the Schur form. The approach to parameterize η first is called the tangent form. In a sense, the classical Rayleigh iteration and its extension in [1] is a Schur form solution for a modified Newton-Raphson equation.

The Schur complement with respect to the top block is $-\boldsymbol{J}_{\boldsymbol{C}} \boldsymbol{L}_{\boldsymbol{x}}^{-1} \boldsymbol{L}_{\boldsymbol{\lambda}}$ evaluated at $(\boldsymbol{x}, \boldsymbol{\lambda})$ and the inverse of the Jacobian applied on (a, b) is

$$\boldsymbol{J}_{\mathcal{L}}^{-1} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \boldsymbol{L}_{\boldsymbol{x}}^{-1} a - \boldsymbol{L}_{\boldsymbol{x}}^{-1} \boldsymbol{L}_{\boldsymbol{\lambda}} [(\boldsymbol{J}_{\boldsymbol{C}} \boldsymbol{L}_{\boldsymbol{x}}^{-1} \boldsymbol{L}_{\boldsymbol{\lambda}})^{-1} \boldsymbol{J}_{\boldsymbol{C}} \boldsymbol{L}_{\boldsymbol{x}}^{-1} a - (\boldsymbol{J}_{\boldsymbol{C}} \boldsymbol{L}_{\boldsymbol{x}}^{-1} \boldsymbol{L}_{\boldsymbol{\lambda}})^{-1} b] \\ (\boldsymbol{J}_{\boldsymbol{C}} \boldsymbol{L}_{\boldsymbol{x}}^{-1} \boldsymbol{L}_{\boldsymbol{\lambda}})^{-1} \boldsymbol{J}_{\boldsymbol{C}} \boldsymbol{L}_{\boldsymbol{x}}^{-1} a - (\boldsymbol{J}_{\boldsymbol{C}} \boldsymbol{L}_{\boldsymbol{x}}^{-1} \boldsymbol{L}_{\boldsymbol{\lambda}})^{-1} b \end{pmatrix} \tag{5.2}$$

evaluated at $(\boldsymbol{x}, \boldsymbol{\lambda})$. With $a = -\boldsymbol{L}(\boldsymbol{x})$ and $b = -\boldsymbol{C}(\boldsymbol{x})$ the Newton step is $\big(\eta, \delta\big)$ with

$$\delta = -(\boldsymbol{J}_{\boldsymbol{C}} \boldsymbol{L}_{\boldsymbol{x}}^{-1} \boldsymbol{L}_{\boldsymbol{\lambda}})^{-1} \boldsymbol{J}_{\boldsymbol{C}} \boldsymbol{L}_{\boldsymbol{x}}^{-1} \boldsymbol{L}(\boldsymbol{x}) + (\boldsymbol{J}_{\boldsymbol{C}} \boldsymbol{L}_{\boldsymbol{x}}^{-1} \boldsymbol{L}_{\boldsymbol{\lambda}})^{-1} \boldsymbol{C}(\boldsymbol{x}) \tag{5.3}$$

$$\eta = -\boldsymbol{L}_{\boldsymbol{x}}^{-1} \boldsymbol{L}(\boldsymbol{x}) - \boldsymbol{L}_{\boldsymbol{x}}^{-1} \boldsymbol{L}_{\boldsymbol{\lambda}} \delta \tag{5.4}$$

On a feasible starting point $\boldsymbol{C}(\boldsymbol{x}) = 0$, we thus have:

$$\delta = -(\boldsymbol{J}_{\boldsymbol{C}} \boldsymbol{L}_{\boldsymbol{x}}^{-1} \boldsymbol{L}_{\boldsymbol{\lambda}})^{-1} \boldsymbol{J}_{\boldsymbol{C}} \boldsymbol{L}_{\boldsymbol{x}}^{-1} \boldsymbol{L}(\boldsymbol{x}) \tag{5.5}$$

We also note $\boldsymbol{J}_{\boldsymbol{C}} \eta = 0$. For nonlinear eigenvalue problem the above process is the Nonlinear inverse iteration in the literature ([14], [12]). We will review this in subsection 7.4. Note that Schur complement

Algorithm 5.1 Newton-Raphson with constrained iterations for general Lagrangian.

Set $feasible$ to True or False
Initialize $\boldsymbol{x}_0$ and $\boldsymbol{\lambda}_0$
$i \leftarrow 0$
$\zeta \leftarrow SMALL\_NUMBER$
$err \leftarrow BIG\_NUMBER$
while not Terminal[1] (i, ζ, err) **do**
  Solve for ζ in $\boldsymbol{L}_{\boldsymbol{x}}(\boldsymbol{x}_i, \boldsymbol{\lambda}_i)\zeta = -\boldsymbol{L}_\lambda(\boldsymbol{x}_i)$
  Solve for ν in $\boldsymbol{L}_{\boldsymbol{x}}(\boldsymbol{x}_i, \boldsymbol{\lambda}_i)\nu = \boldsymbol{L}(\boldsymbol{x}_i)$
  Compute $\delta \leftarrow (\boldsymbol{J}_{\boldsymbol{C}}(\boldsymbol{x}_i)\zeta)^{-1}[-\boldsymbol{C}(\boldsymbol{x}_i) + \boldsymbol{J}_{\boldsymbol{C}}(\boldsymbol{x}_i)\nu]$
  Compute $\boldsymbol{\lambda}_{i+1} \leftarrow \boldsymbol{\lambda}_i + \delta$
  Compute $\eta \leftarrow -\nu + \zeta\delta$
  if $feasible$ **then**
    Compute $\boldsymbol{x}_{i+1} \leftarrow \boldsymbol{r}(\boldsymbol{x}_i, \eta)$ ($\boldsymbol{r}$ is a retraction.)
  else
    Compute $\boldsymbol{x}_{i+1} \leftarrow \boldsymbol{x}_i + \eta$
  end if
  $i \leftarrow i + 1$
  $err \leftarrow \boldsymbol{L}(\boldsymbol{x}_{i+1}, \boldsymbol{\lambda}_{i+1})$
end while

is widely use in equality constraint optimization, for example chapter 10 of [6] has essentially the above calculation. We summarize the steps in Algorithm 5.1.

When $\boldsymbol{L}$ is explicit, $\boldsymbol{L}_{\boldsymbol{x}}$ and $\boldsymbol{L}_{\boldsymbol{\lambda}}$ are two linear maps:

$$\boldsymbol{L}_{\boldsymbol{x}}\eta = \boldsymbol{J}_{\boldsymbol{F}}(\boldsymbol{x})\eta - \boldsymbol{J}_{\boldsymbol{H}}(\boldsymbol{x})\eta\boldsymbol{\lambda}$$

$$\boldsymbol{L}_{\boldsymbol{\lambda}}\delta = -\boldsymbol{H}\delta$$

$\boldsymbol{J}_{\boldsymbol{H}}\eta\boldsymbol{\lambda}$ actually has a simpler form. Write it as

$$\sum_b\sum_c J_{abc}\eta_c\boldsymbol{\lambda}_b = \sum_c(\sum_b \boldsymbol{\lambda}_b J_{abc})\eta_c$$

we see it is $(\boldsymbol{\lambda}^T\boldsymbol{J}_{\boldsymbol{H}}^{T[12]})\eta$. Where the expression in the bracket is a tensor contraction. Here, $T[12]$ means transposing $\boldsymbol{J}_{\boldsymbol{H}}$ with respect to the first two indices.

$$\boldsymbol{L}_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{\lambda})\eta = \boldsymbol{J}_{\boldsymbol{F}}\eta - \boldsymbol{J}_{\boldsymbol{H}}\eta\boldsymbol{\lambda} = (\boldsymbol{J}_{\boldsymbol{F}} - \boldsymbol{\lambda}^T\boldsymbol{J}_{\boldsymbol{H}}^{T[12]})\eta \tag{5.6}$$

$\boldsymbol{L}_{\boldsymbol{x}}$ is a generalization of the operator $A - \boldsymbol{\lambda}I$ of eigenvalue problem. We do not need the full inverse of $\boldsymbol{L}_{\boldsymbol{x}}$ in general, but will need to solve for $\boldsymbol{L}_{\boldsymbol{x}}\eta = B$ for some matrix B in each iteration. We collect all the result thus far in Theorem 5.1.

THEOREM 5.1. *The Newton-Raphson iteration equations for $\boldsymbol{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ with constraint $\boldsymbol{C}(\boldsymbol{x}) = 0$ are*

$$\boldsymbol{\lambda}_{i+1} - \boldsymbol{\lambda}_i = \delta = (\boldsymbol{J}_{\boldsymbol{C}}\boldsymbol{L}_{\boldsymbol{x}}^{-1}\boldsymbol{L}_{\boldsymbol{\lambda}})^{-1}[\boldsymbol{C}(\boldsymbol{x}) - \boldsymbol{J}_{\boldsymbol{C}}\boldsymbol{L}_{\boldsymbol{x}}^{-1}\boldsymbol{L}(\boldsymbol{x})] \tag{5.7}$$

$$\boldsymbol{x}_{i+1} - \boldsymbol{x}_i = \eta = -\boldsymbol{L}_{\boldsymbol{x}}^{-1}(\boldsymbol{L} + \boldsymbol{L}_{\boldsymbol{\lambda}}\delta) \tag{5.8}$$

[1] *Terminal* is a function of η, err and i used to decide when to exit the iterations. Typical terminal criteria include $||err|| < max\_err$ or $||\zeta > max\_zeta||$ subjected to a max iteration count.

If $\boldsymbol{x} \in \mathcal{M}$ *then we have*

$$\boldsymbol{J_C}\eta = 0$$

In the explicit case $\boldsymbol{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = \boldsymbol{F}(\boldsymbol{x}) - \boldsymbol{H}(\boldsymbol{x})(\boldsymbol{\lambda})$ *we have*

$$\boldsymbol{L_\lambda}(\boldsymbol{x}, \boldsymbol{\lambda}) = -\boldsymbol{H}(\boldsymbol{x})$$

$$\boldsymbol{L_x} = \boldsymbol{J_F} - \boldsymbol{\lambda}\boldsymbol{J}_{\boldsymbol{H}}^{T[12]}$$

$$\boldsymbol{\lambda}_{i+1} = (\boldsymbol{J_C}\boldsymbol{L_x}^{-1}\boldsymbol{H})^{-1}[-\boldsymbol{C}(\boldsymbol{x}) + \boldsymbol{J_C}\boldsymbol{L_x}^{-1}\boldsymbol{F}(\boldsymbol{x})] \tag{5.9}$$

$$\boldsymbol{x}_{i+1} - \boldsymbol{x}_i = \eta = \boldsymbol{L_x}^{-1}(-\boldsymbol{F}(\boldsymbol{x}) + \boldsymbol{H}(\boldsymbol{x})\boldsymbol{\lambda}_{i+1}) \tag{5.10}$$

If $\boldsymbol{r}$ *is a retraction of first order we can use:*

$$\boldsymbol{x}_{i+1} = \boldsymbol{r}(\boldsymbol{x}_i, \eta)$$

Proof. We already proved the general case. The explicit case is a direct substitution of $\boldsymbol{L}$ in to (5.3). □

Algorithm 5.2 Newton-Raphson with constrained iterations for explicit Lagrangian.

Set $feasible$ to True or False
Initialize $\boldsymbol{x}_0$ and $\boldsymbol{\lambda}_0$
$i \leftarrow 0$
$\zeta \leftarrow SMALL\_NUMBER$
$err \leftarrow LARGE\_NUMBER$
while not Terminal(i, ζ, err) **do**
  Solve for ζ in $\boldsymbol{L_x}(\boldsymbol{x}_i, \boldsymbol{\lambda}_i)\zeta = \boldsymbol{H}(\boldsymbol{x}_i)$
  Solve for ν in $\boldsymbol{L_x}(\boldsymbol{x}_i, \boldsymbol{\lambda}_i)\nu = \boldsymbol{F}(\boldsymbol{x}_i)$
  Compute $\boldsymbol{\lambda}_{i+1} \leftarrow (\boldsymbol{J_C}(\boldsymbol{x}_i)\zeta)^{-1}[-\boldsymbol{C}(\boldsymbol{x}_i) + \boldsymbol{J_C}(\boldsymbol{x}_i)\nu]$
  Compute $\eta \leftarrow -\nu + \zeta\boldsymbol{\lambda}_{i+1}$
  if $feasible$ **then**
    Compute $\boldsymbol{x}_{i+1} \leftarrow \boldsymbol{r}(\boldsymbol{x}_i, \eta)$
  else
    Compute $\boldsymbol{x}_{i+1} \leftarrow \boldsymbol{x}_i + \eta$
  end if
  $i \leftarrow i + 1$
  $err \leftarrow \boldsymbol{L}(\boldsymbol{x}_{i+1}, \boldsymbol{\lambda}_{i+1})$
end while

We note the $\boldsymbol{\lambda}$ is updated first, then $\boldsymbol{\lambda}_{i+1}$ is used in equation for η:

$$\boldsymbol{L_x}\eta = -\boldsymbol{F}(\boldsymbol{x}) + \boldsymbol{H}(\boldsymbol{x})\boldsymbol{\lambda}_{i+1} \tag{5.11}$$

While we have noted before $\boldsymbol{L_x}$ is a generalization of the resolvent operator, the right-hand side of this equation is different from that of Rayleigh quotient. To compute $\boldsymbol{\lambda}_{i+1}$ and η we compute

$$\zeta = \boldsymbol{L_x}(\boldsymbol{x}_i, \boldsymbol{\lambda}_i)^{-1}\boldsymbol{H}(\boldsymbol{x}_i)$$

$$\nu = \boldsymbol{L_x}(\boldsymbol{x}_i, \boldsymbol{\lambda}_i)^{-1}\boldsymbol{F}(\boldsymbol{x}_i)$$

In the eigenvector case, ν and ζ are related:

$$\nu = \boldsymbol{x} + \zeta\boldsymbol{\lambda}$$

and we only need to solve for ζ as seen before. In the general case, we need to solve for both ν and ζ. The expression of $\boldsymbol{\lambda}_{i+1}$ is different from Rayleigh quotient.

Let us now focus on reconciling $\boldsymbol{\lambda}_{i+1}$ with Rayleigh quotient. From (5.10)

$$\boldsymbol{H}(\boldsymbol{x})\boldsymbol{\lambda}_{i+1} = \boldsymbol{L}_{\boldsymbol{x}}\eta + \boldsymbol{F}(\boldsymbol{x})$$

As before let $\boldsymbol{H}^{-}$ be a left invert to $\boldsymbol{H}$, we solve for $\boldsymbol{\lambda}_{i+1}$:

$$\boldsymbol{\lambda}_{i+1} = \boldsymbol{H}^{-}[\boldsymbol{L}_{\boldsymbol{x}}\eta + \boldsymbol{F}(\boldsymbol{x})] \tag{5.12}$$

We see if η converges to zero, $\boldsymbol{\lambda}_{i+1}$ converges to

$$\boldsymbol{\lambda}_{i+1} = \boldsymbol{H}^{-}\boldsymbol{F}(\boldsymbol{x})$$

as noted before. $\boldsymbol{H}^{-}$ may be of a more general form than $(\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T$, for example we can replace $\boldsymbol{H}^T$ with any map $\boldsymbol{H}^{\dagger}$ such that $\boldsymbol{H}^{\dagger}\boldsymbol{H}$ is invertible. This setup is relevant when we discuss the two-sided Rayleigh quotient.

We note that Gabay ([9], [10]) found the same expression for our Rayleigh quotient as an estimate of the Lagrange multiplier, together with a related quasi-Newton method. The special case of $(\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T$ appeared much earlier in the literature (for example [13]). Already in 1977, Tapia ([18]) suggested it is difficult to put a name on it without an extensive search of the literature. Lagrange multiplier approximators are widely used in constrained optimization.

Let $\Pi_{\boldsymbol{H}} = \boldsymbol{I}_n - \boldsymbol{H}\boldsymbol{H}^{-}$. This is a projection to $\operatorname{Im}\Pi_{\boldsymbol{H}}$. We note $\operatorname{Ker}(\Pi_{\boldsymbol{H}}) = \operatorname{Im}(\boldsymbol{H})$. If $\boldsymbol{H} = \boldsymbol{J}_{\boldsymbol{C}}^T$ then $\Pi_{\boldsymbol{H}}$ is a projection to the tangent space of $\mathcal{M}$ at $\boldsymbol{x}$. Substitute the expression (5.12) in (5.11) and moving η term to one side:

$$\Pi_{\boldsymbol{H}}\boldsymbol{L}_{\boldsymbol{x}}\eta = -\Pi_{\boldsymbol{H}}\boldsymbol{F}(\boldsymbol{x}) \tag{5.13}$$

This system of equations for η is of similar format to the Riemannian Newton equations. However, it is dependent on $\boldsymbol{\lambda}$ which is computed recursively. It is interesting that it involves only $\boldsymbol{H}$ here, and not $\boldsymbol{C}$. In the general case, assuming $\boldsymbol{L}_{\boldsymbol{\lambda}}^{-}$ is a left inverse of $\boldsymbol{L}_{\boldsymbol{\lambda}}$, performing a similar substitution we have with $\Pi_{\boldsymbol{L}_{\boldsymbol{\lambda}}} = \boldsymbol{I} - \boldsymbol{L}_{\boldsymbol{\lambda}}\boldsymbol{L}_{\boldsymbol{\lambda}}^{-}$

Proposition 5.2. *For equation (1.3), at a feasible point the $\boldsymbol{x}$ component Newton-Raphson update η satisfies:*

$$\boldsymbol{J}_{\boldsymbol{C}}\eta = 0$$

$$\Pi_{\boldsymbol{L}_{\boldsymbol{\lambda}}}\boldsymbol{L}_{\boldsymbol{x}}\eta = -\Pi_{\boldsymbol{L}_{\boldsymbol{\lambda}}}\boldsymbol{L}.$$

which reduces to $\Pi_{\boldsymbol{H}}\boldsymbol{L}_{\boldsymbol{x}}\eta = -\Pi_{\boldsymbol{H}}\boldsymbol{F}(\boldsymbol{x})$ in the explicit case.

For second order iteration, we note the formula (see [7]):

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \boldsymbol{J}_{\mathcal{L}}^{-1}\mathcal{L} - \frac{1}{2}\boldsymbol{J}_{\mathcal{L}}^{(2)}((\boldsymbol{J}_{\mathcal{L}}^{-1}\mathcal{L})^{[2]}) \tag{5.14}$$

for Chebyshev iteration. We choose Chebyshev over Halley to avoid another linear operator inversion. As explain before with $V = E_{in} \oplus E_L$ and $W = E_{out} \oplus E_L$, $\boldsymbol{J}_{\mathcal{L}}^{[2]}$ is an element of $L(V, L(V, W)$ and $\boldsymbol{J}_{\mathcal{L}}^{-1}\mathcal{L}$ is an element of V and the last term is a contraction of the tensor $\boldsymbol{J}_{\mathcal{L}}^{(2)}$ twice on $\boldsymbol{J}_{\mathcal{L}}^{-1}\mathcal{L}$. We note that

$$\boldsymbol{J}_{\mathcal{L}}^{(2)}\begin{pmatrix}\eta\\ \delta\end{pmatrix}^{[2]} = \begin{pmatrix}\boldsymbol{J}_{\boldsymbol{F}}^{(2)}\eta^{[2]} - (\boldsymbol{J}_{\boldsymbol{H}}^{(2)}\eta^{[2]})\boldsymbol{\lambda} - 2\boldsymbol{J}_{\boldsymbol{H}}[\eta, \delta]\\ \boldsymbol{J}_{\boldsymbol{C}}^{(2)}\eta^{[2]}\end{pmatrix}$$

We proceed to use Schur complement to evaluate the second order term to arrive at Chebyshev iteration in Algorithm 5.3. While in general this iteration may be difficult, when $\boldsymbol{F}$, $\boldsymbol{H}$ and $\boldsymbol{C}$ are at most quadratic the algorithm may be useful. In particular, we have a cubic convergent algorithm for

Algorithm 5.3 Chebyshev with constrained iterations

Set $feasible$ to True or False
Initialize $\boldsymbol{x}_0$ and $\boldsymbol{\lambda}_0 = \mathcal{R}(\boldsymbol{x})$
$i \leftarrow 0$
$\zeta \leftarrow SMALL\_NUMBER$
$err \leftarrow LARGE\_NUMBER$
while not Terminal(i, η, err) **do**
  Solve for ζ in $\boldsymbol{L_x}(\boldsymbol{x}_i, \boldsymbol{\lambda}_i)\zeta = \boldsymbol{H}(\boldsymbol{x}_i)$
  Solve for ν in $\boldsymbol{L_x}(\boldsymbol{x}_i, \boldsymbol{\lambda}_i)\nu = \boldsymbol{F}(\boldsymbol{x}_i)$
  Compute $\boldsymbol{\lambda}_{i+i} \leftarrow (\boldsymbol{J_C}(\boldsymbol{x}_i)\zeta)^{-1}[-\boldsymbol{C}(\boldsymbol{x}_i) + \boldsymbol{J_C}(\boldsymbol{x}_i)\nu]$
  Compute $\eta \leftarrow -\nu + \zeta\boldsymbol{\lambda}_{i+1}$
  Compute $l_2 \leftarrow \boldsymbol{J}^{(2)}_{\boldsymbol{F}}(\eta^{[2]}) - \boldsymbol{J}^{(2)}_{\boldsymbol{H}}(\eta^{[2]})\boldsymbol{\lambda} - 2\boldsymbol{J_H}(\eta)\delta$
  Compute $c_2 \leftarrow \boldsymbol{J}^{(2)}_{\boldsymbol{C}}(\eta^{[2]})$
  Compute $LxInvL_2 \leftarrow \boldsymbol{L}_{\boldsymbol{x}}^{-1} l_2$
  Compute $\delta_2 \leftarrow (\boldsymbol{J_C}\zeta)^{-1}(\boldsymbol{J_C}(LxInvL_2) - \boldsymbol{J}^{(2)}_{\boldsymbol{C}}(\eta^{[2]}))$
  Compute $\eta_2 \leftarrow LxInvL_2 + (\boldsymbol{J_C}\zeta)^{-1} l_2$
  Compute $\boldsymbol{\lambda}_{i+i} \leftarrow \boldsymbol{\lambda}_{i+i} - \frac{1}{2}\delta_2$
  if $feasible$ **then**
    Compute $\boldsymbol{x}_{i+1} \leftarrow \boldsymbol{r}(\boldsymbol{x}_i, \eta - \frac{1}{2}\eta_2)$
  else
    Compute $\boldsymbol{x}_{i+1} \leftarrow \boldsymbol{x}_i + \eta - \frac{1}{2}\eta_2$
  end if
  $i \leftarrow i + 1$
  $err \leftarrow \boldsymbol{L}(\boldsymbol{x}_{i+1}, \boldsymbol{\lambda}_{i+1})$
end while

eigenvectors, even when the matrix is not normal: in the Chebyshev term the only nonzero terms are $-\eta\delta$ and $\eta^T\eta$.

When $\boldsymbol{x}$ is a vector and $\boldsymbol{H}(\boldsymbol{x})$ is represented as a matrix, ζ is a matrix and $\boldsymbol{J}_C(\boldsymbol{x}_i)\zeta$ can be represented as a square matrix, so the calculation is simple. When $\boldsymbol{x}$ is a matrix, $\boldsymbol{H}(\boldsymbol{x})$ could be a higher order tensor, so ζ and $\boldsymbol{J}_C\boldsymbol{x}_i\zeta$ in general are tensors. The main difficulty of this method is in evaluating these tensors and the inverse of the Schur complement $\boldsymbol{J}_C\boldsymbol{x}_i\zeta$.

6. Rayleigh Quotient Iteration.

Motivated by [1] (we learned about [9], [10] late in our research), Proposition 5.2 and the above analysis on Lagrange multipliers, with:

$$\boldsymbol{\lambda} = \mathcal{R}(\boldsymbol{x}) := \boldsymbol{H}^{-}\boldsymbol{F}(\boldsymbol{x})$$

in the expression for $\boldsymbol{L_x}$, it is plausible that the system:

$$\begin{aligned} \Pi_{\boldsymbol{H}}\boldsymbol{L_x}\eta &= -\,\Pi_{\boldsymbol{H}}\boldsymbol{F}(\boldsymbol{x}) \\ \boldsymbol{J_C}\eta &= 0 \end{aligned} \tag{6.1}$$

would provide a generalization of RQI to vector Lagrangians. Using augmented Lagrangian technique, Gabay ([10]) proposed a quasi-Newton method with this expression of $\boldsymbol{\lambda}$ as an estimate for the Lagrange multiplier. He showed that it converges superlinearly in general. We will prove quadratic convergence here and also consider the Chebyshev version.

Similar to the Newton-Raphson case, if $\boldsymbol{L_x}$ is invertible we have a solution to this system in term of $\boldsymbol{L}_x^{-1}$. In fact, let $\nu = \boldsymbol{L}_{\boldsymbol{x}}^{-1}\boldsymbol{F}(\boldsymbol{x})$ and $\zeta = \boldsymbol{L}_{\boldsymbol{x}}^{-1}\boldsymbol{H}$ we see

$$\eta = \zeta(\boldsymbol{J_C}\zeta)^{-1}\boldsymbol{J_C}\nu - \nu \tag{6.2}$$

satisfies (6.1) by direct calculation. In essence, this is a projection of $-\nu$ to the tangent space in the direction of ζ. As before we call it the Schur form solution. Before we proceed with the theorems and the proofs, let examine the iterative process associated with equation (6.1) in two instances.

The first instance is the eigenvector problem $\boldsymbol{\lambda} = (\boldsymbol{x}^T\boldsymbol{x})^{-1}\boldsymbol{x}^T A\boldsymbol{x}$ and $\boldsymbol{L}_{\boldsymbol{x}}\eta = A\eta - \eta\boldsymbol{\lambda}$, so (6.1) is the resultant equation and a calculation similar to section 2 shows $\boldsymbol{x}_{i+1} = (A - \boldsymbol{\lambda})\boldsymbol{x}_i$. The iterative process is exactly the classical RQI if we use the projection to the sphere as the retraction.

The second instance is the constraint optimization problem: $\boldsymbol{F} = \nabla f$, $\boldsymbol{H} = \boldsymbol{J}_{\boldsymbol{C}}^T$ and $\boldsymbol{H}^- = (\boldsymbol{J}_{\boldsymbol{C}}\boldsymbol{J}_{\boldsymbol{C}}^T)^{-1}\boldsymbol{J}_{\boldsymbol{C}}$ and hence $\boldsymbol{\lambda} = (\boldsymbol{J}_{\boldsymbol{C}}\boldsymbol{J}_{\boldsymbol{C}}^T)^{-1}\boldsymbol{J}_{\boldsymbol{C}}\nabla f$. In notation of [3], the projection to the tangent space of $\mathcal{M}$ at $\boldsymbol{x}$ is simply $\Pi_{\boldsymbol{H}}$:

$$\Pi_{\boldsymbol{H}}(\boldsymbol{x}) = \boldsymbol{I}_{E_{out}} - \boldsymbol{J}_{\boldsymbol{C}}^T(\boldsymbol{J}_{\boldsymbol{C}}\boldsymbol{J}_{\boldsymbol{C}}^T)^{-1}\boldsymbol{J}_{\boldsymbol{C}} = P_{\boldsymbol{x}}$$

Hence the Riemannian gradient is simply $\Pi_{\boldsymbol{H}}(\boldsymbol{x})\boldsymbol{F}$. Note

$$\boldsymbol{L}_{\boldsymbol{x}}\eta = \boldsymbol{J}_{\boldsymbol{F}}\eta - \boldsymbol{J}_{\boldsymbol{C}}^{(2)}\eta\boldsymbol{\lambda} = \nabla^2 f\eta - \boldsymbol{J}_{\boldsymbol{C}}^{(2)}\eta(\boldsymbol{J}_{\boldsymbol{C}}\boldsymbol{J}_{\boldsymbol{C}}^T)^{-1}\boldsymbol{J}_{\boldsymbol{C}}\nabla f$$

Formula (5.15) in section 5.3 of [3] shows the Riemannian Hessian of f is:

$$\operatorname{Hess} f[\eta] = P_{\boldsymbol{x}}(D(P_{\boldsymbol{x}}\nabla f)\eta = P_{\boldsymbol{x}}D(\nabla f - \boldsymbol{J}_{\boldsymbol{C}}^T(\boldsymbol{J}_{\boldsymbol{C}}\boldsymbol{J}_{\boldsymbol{C}}^T)^{-1}\boldsymbol{J}_{\boldsymbol{C}}\nabla f)$$

Here D is the classical derivatives with respect to $\boldsymbol{x}$. Expanding the above and keep exploiting the fact that $P_{\boldsymbol{x}} = \Pi_{\boldsymbol{H}}$ annihilates terms starting with $\boldsymbol{J}_{\boldsymbol{C}}^T$

$$\begin{aligned}\operatorname{Hess} f[\eta] &= P_{\boldsymbol{x}}(D(\nabla f - \boldsymbol{J}_{\boldsymbol{C}}^T(\boldsymbol{J}_{\boldsymbol{C}}\boldsymbol{J}_{\boldsymbol{C}}^T)^{-1}\boldsymbol{J}_{\boldsymbol{C}}\nabla f)\eta \\ &= P_{\boldsymbol{x}}\nabla^2 f\eta - P_{\boldsymbol{x}}D(\boldsymbol{J}_{\boldsymbol{C}}^T(\boldsymbol{J}_{\boldsymbol{C}}\boldsymbol{J}_{\boldsymbol{C}}^T)^{-1}\boldsymbol{J}_{\boldsymbol{C}})\nabla f \\ &= P_{\boldsymbol{x}}\nabla^2 f\eta - P_{\boldsymbol{x}}\boldsymbol{J}_{\boldsymbol{C}}^{(2)}\eta\boldsymbol{\lambda} = P_{\boldsymbol{x}}(\nabla^2 f - \boldsymbol{J}_{\boldsymbol{C}}^{(2)}\eta\boldsymbol{\lambda})\end{aligned}$$

and this is exactly $\Pi_{\boldsymbol{H}}\boldsymbol{L}_{\boldsymbol{x}}\eta$. So the first equation of (6.1) is the Riemannian Newton equation for the embedded manifold $\mathcal{M}$ in this case.

We note $\Pi_{\boldsymbol{H}}\boldsymbol{L}_{\boldsymbol{x}}$ is a map from $T\mathcal{M}_{\boldsymbol{x}}$ to $\operatorname{Im}(\Pi_{\boldsymbol{H}})$, both of the same dimension $\dim(E_{in}) - \dim(E_L)$ and hence it could have an inverse. From the above analysis it is a generalization of the Riemannian Hessian, while $\Pi_{\boldsymbol{H}}\boldsymbol{L}$ is a generalization of the Riemannian gradient. As we need the Riemannian Hessian to be invertible for Riemannian Newton to work, we would need $\Pi_{\boldsymbol{H}}\boldsymbol{L}_{\boldsymbol{x}}$ to be invertible and to satisfy a smoothness condition which we will assume in the following theorem:

Theorem 6.1. *Let $\boldsymbol{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = \boldsymbol{F}(\boldsymbol{x}) - \boldsymbol{H}(x)\boldsymbol{\lambda}$ be a Lagrangian of explicit form, $\boldsymbol{H}^-$ be a left inverse of $\boldsymbol{H}$, $\mathcal{R}(x) = \boldsymbol{H}^-\boldsymbol{F}(\boldsymbol{x})$ be the generalized Rayleigh quotient and $\Pi_{\boldsymbol{H}} = \boldsymbol{I} - \boldsymbol{H}\boldsymbol{H}^-$ be the projection as above. Let $(\boldsymbol{v}, \boldsymbol{\mu})$ be a solution to the equation (1.3) ($\boldsymbol{\mu} = \mathcal{R}(\boldsymbol{v})$) and $\boldsymbol{r}$ be a first order retraction. Assume:*

- *$\boldsymbol{H}, \boldsymbol{F}$ are of class C^2.*
- *$\mathcal{R}$ is of class C^1 in a neighborhood of $(\boldsymbol{v}, \boldsymbol{\mu})$.*
- *$\Pi_{\boldsymbol{H}}(\boldsymbol{x}_i)$ is of class C^1.*
- *the map $\Pi_{\boldsymbol{H}}(\boldsymbol{x}_i)\boldsymbol{L}_{\boldsymbol{x}}(\boldsymbol{x}_i)$ from $T\mathcal{M}_{\boldsymbol{x}_i}$ to $\operatorname{Im}(\Pi_{\boldsymbol{H}}(\boldsymbol{x}_i))$ is invertible and for $\boldsymbol{x}$ in a neighborhood of $\boldsymbol{v}$:*

$$||\Pi_{\boldsymbol{L}_{\boldsymbol{\lambda}}}\boldsymbol{L}_{\boldsymbol{x}}(\boldsymbol{x}, \mathcal{R}(\boldsymbol{x}))\psi|| \geq C||\psi|| \tag{6.3}$$

then for a starting point $\boldsymbol{x}_0$ close enough to $\boldsymbol{v}$, the system

$$\begin{aligned}\Pi_{\boldsymbol{H}}\boldsymbol{L}_{\boldsymbol{x}_i}(\boldsymbol{x}_i)\eta &= -\,\Pi_{\boldsymbol{H}}\boldsymbol{F}(\boldsymbol{x}_i) \\ \boldsymbol{J}_{\boldsymbol{C}}(\boldsymbol{x}_i)\eta &= 0 \\ \boldsymbol{x}_{i+1} &= \boldsymbol{r}(\boldsymbol{x}_i, \eta)\end{aligned}$$

provides an update to an iteration converging to $(\boldsymbol{v}, \mathcal{R}(\boldsymbol{v}))$ quadratically.

If further $\boldsymbol{H}, \boldsymbol{F}$ are of class C^3, $\mathcal{R}$ is of class C^2 and $\boldsymbol{r}$ is a second order retraction, then for a starting point $\boldsymbol{x}_0$ close enough to $\boldsymbol{v}$ the Rayleigh-Chebyshev iteration with update:

$$\boldsymbol{x}_{i+1} = \boldsymbol{r}(\boldsymbol{x}_i, \eta + T(\boldsymbol{x}_i)[\eta^{[2]}])$$

where η and $T(\boldsymbol{x}_i)[\eta^{[2]}]$ are defined in the following equations:

$$\begin{gathered}\Pi_{\boldsymbol{H}} \boldsymbol{L}_{\boldsymbol{x}}(\boldsymbol{x}_i)\eta = -\Pi_{\boldsymbol{H}} \boldsymbol{F}(\boldsymbol{x}_i)\\ \Pi_{\boldsymbol{H}} \boldsymbol{L}_{\boldsymbol{x}} T(\boldsymbol{x}_i)[\eta^{[2]}] = G(\boldsymbol{x}_i)[\eta^{[2]}] := -\frac{1}{2}\boldsymbol{J}_F^{(2)}(\boldsymbol{x}_i)[\eta^{[2]}] - \boldsymbol{J}_{\boldsymbol{H}}(\boldsymbol{x}_i)[\eta]\boldsymbol{J}_{\mathcal{R}}(\boldsymbol{x}_i)[\eta] + \frac{1}{2}\boldsymbol{J}_{\boldsymbol{H}}^{(2)}(\boldsymbol{x}_i)[\eta^{[2]}]\mathcal{R}(\boldsymbol{x}_i)\\ \boldsymbol{J}_{\boldsymbol{C}}(\boldsymbol{x}_i)[\eta + T(\boldsymbol{x}_i)[\eta^{[2]}]] = 0\end{gathered} \tag{6.4}$$

converges cubically to $\boldsymbol{v}$.

If $G(\boldsymbol{v}) = 0$ and $\boldsymbol{r}$ is of second order the Rayleigh quotient iteration converges cubically.

If $\boldsymbol{L}_{\boldsymbol{x}}(\boldsymbol{x}_i)$ is invertible the Schur form solution exists:

$$\begin{gathered}\nu = \boldsymbol{L}_{\boldsymbol{x}}^{-1}(\boldsymbol{x}_i)\boldsymbol{F}(\boldsymbol{x}_i)\\ \zeta = \boldsymbol{L}_{\boldsymbol{x}}^{-1}(\boldsymbol{x}_i)\boldsymbol{H}(\boldsymbol{x}_i)\\ \eta = -\nu + \zeta(\boldsymbol{J}_{\boldsymbol{C}}(\boldsymbol{x}_i)\zeta)^{-1}\boldsymbol{J}_{\boldsymbol{C}}(\boldsymbol{x}_i)\nu\\ \eta_* = -\nu\\ \tau_* = \boldsymbol{L}_{\boldsymbol{x}}^{-1}(\boldsymbol{x}_i)\{-\frac{1}{2}\boldsymbol{J}_F^{(2)}(\boldsymbol{x}_i)[\eta_*^{[2]}] - \boldsymbol{J}_{\boldsymbol{H}}(\boldsymbol{x}_i)[\eta_*]\boldsymbol{J}_{\mathcal{R}}(\boldsymbol{x}_i)[\eta_*] + \frac{1}{2}\boldsymbol{J}_{\boldsymbol{H}}^{(2)}(\boldsymbol{x}_i)[\eta_*^{[2]}]\mathcal{R}(\boldsymbol{x}_i)\}\\ \tau = \tau_* - \zeta(\boldsymbol{J}_{\boldsymbol{C}}(\boldsymbol{x}_i)\zeta)^{-1}\boldsymbol{J}_{\boldsymbol{C}}(\boldsymbol{x}_i)[\tau_*]\end{gathered} \tag{6.5}$$

We will state and prove the theorem for the general Lagrangian case, as the notations turn out to be simpler and the estimates clearly show the relationship to Taylor series. The explicit case will follow as a corollary. Note that for Rayleigh-Chebyshev iteration, we just need τ to be in the tangent space (at $\boldsymbol{x}_i$) while the intermediate η could be chosen as η_*, which need not be in the tangent space as in (6.5).

THEOREM 6.2. *Let $\boldsymbol{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ be a Lagrangian, $\boldsymbol{L}_{\boldsymbol{x}}, \boldsymbol{L}_{\boldsymbol{\lambda}}$ be its partial derivatives with respect to $\boldsymbol{x}$ and $\boldsymbol{\lambda}$. Let $(\boldsymbol{v}, \boldsymbol{\mu})$ be a solution for the system (1.1). Assuming*

- *$\boldsymbol{L}$ is of class C^2.*
- *$\mathcal{R}$ is a function of class C^1 from a neighborhood of $\boldsymbol{v}$ to E_L such that $\mathcal{R}(\boldsymbol{v}) = \boldsymbol{\mu}$.*
- *$\boldsymbol{L}_{\boldsymbol{\lambda}}{}^-$ is a left inverse of $\boldsymbol{L}_{\boldsymbol{\lambda}}$ of class C^1 and $\Pi_{\boldsymbol{L}_{\boldsymbol{\lambda}}} = \boldsymbol{I} - \boldsymbol{L}_{\boldsymbol{\lambda}}\boldsymbol{L}_{\boldsymbol{\lambda}}{}^-$.*
- *$\Pi_{\boldsymbol{L}_{\boldsymbol{\lambda}}}\boldsymbol{L}_{\boldsymbol{x}}(\boldsymbol{x})$ is invertible (as a map from $T\mathcal{M} = \mathrm{Ker}(\boldsymbol{J}_{\boldsymbol{C}}(\boldsymbol{x}))$ to $\mathrm{Im}(\Pi_{\boldsymbol{L}_{\boldsymbol{\lambda}}}(\boldsymbol{x}))$) such that*

$$||\Pi_{\boldsymbol{L}_{\boldsymbol{\lambda}}}\boldsymbol{L}_{\boldsymbol{x}}(\boldsymbol{x}, \mathcal{R}(\boldsymbol{x}))\psi|| \geq C||\psi|| \tag{6.6}$$

in a neighborhood of $\boldsymbol{v}$.

The generalized Rayleigh quotient iteration $\boldsymbol{x}_{i+1} = \boldsymbol{r}(\boldsymbol{x}_i, \eta)$ with

$$\begin{gathered}\boldsymbol{\lambda} = \mathcal{R}(\boldsymbol{x}_i)\\ \Pi_{\boldsymbol{L}_{\boldsymbol{\lambda}}}\boldsymbol{L}_{\boldsymbol{x}}(\boldsymbol{x}_i, \mathcal{R}(\boldsymbol{x}_i))\eta = -\Pi_{\boldsymbol{L}_{\boldsymbol{\lambda}}}\boldsymbol{L}(\boldsymbol{x}_i, \mathcal{R}(\boldsymbol{x}_i))\\ \boldsymbol{J}_{\boldsymbol{C}}(\boldsymbol{x}_i)\eta = 0\end{gathered} \tag{6.7}$$

together with a first order retraction converges quadratically to $(\boldsymbol{v}, \boldsymbol{\mu})$. If further $\boldsymbol{L}$ is of class C^3 and $\mathcal{R}$ if of class C^2 then the Rayleigh-Chebyshev iterative process $\boldsymbol{x}_{i+1} = \boldsymbol{r}(x_i, \tau)$ with $\tau = \eta_* + T(\eta_*^{[2]})$ constructed to satisfy:

$$\begin{gathered}\boldsymbol{\lambda} = \mathcal{R}(\boldsymbol{x}_i)\\ \Pi_{\boldsymbol{L}_{\boldsymbol{\lambda}}}\boldsymbol{L}_{\boldsymbol{x}}(\boldsymbol{x}_i)\eta_* = -\Pi_{\boldsymbol{L}_{\boldsymbol{\lambda}}}\boldsymbol{L}(\boldsymbol{x}_i, \mathcal{R}(\boldsymbol{x}_i))\\ \Pi_{\boldsymbol{L}_{\boldsymbol{\lambda}}}\boldsymbol{L}_{\boldsymbol{x}}(\boldsymbol{x}_i)T(\boldsymbol{x}_i)[\eta_*^{[2]}] = G(\boldsymbol{x}_i)[\eta_*^{[2]}] := -\frac{1}{2}\boldsymbol{L}_{\boldsymbol{x}\boldsymbol{x}}(\boldsymbol{x}_i)[\eta_*^{[2]}] - \boldsymbol{L}_{\boldsymbol{x}\boldsymbol{\lambda}}(\boldsymbol{x}_i)[\eta_*, \boldsymbol{J}_{\mathcal{R}}[\eta_*]] - \frac{1}{2}\boldsymbol{J}_{\boldsymbol{\lambda}\boldsymbol{\lambda}}(\boldsymbol{x}_i)[(\boldsymbol{J}_{\mathcal{R}}[\eta_*]^{[2]})]\\ \boldsymbol{J}_{\boldsymbol{C}}(\boldsymbol{x}_i)[\eta_* + T(\boldsymbol{x}_i)[\eta_*^{[2]}]] = 0\end{gathered} \tag{6.8}$$

together with a second order retraction converges cubically. If $G(\boldsymbol{v}) = 0$ and $\boldsymbol{r}$ is of second order the generalized RQI converges cubically. When $\boldsymbol{L_x}$ is invertible the Schur form of the solution exists with:

$$(6.9)\quad \begin{aligned}\boldsymbol{\lambda} &= \mathcal{R}(\boldsymbol{x}_i)\zeta = -\boldsymbol{L_x}^{-1}(\boldsymbol{x}_i, \boldsymbol{\lambda})\boldsymbol{L_\lambda}(\boldsymbol{x}_i, \boldsymbol{\lambda})\\ \nu &= \boldsymbol{L_x}^{-1}(\boldsymbol{x}_i, \boldsymbol{\lambda})\boldsymbol{L}(\boldsymbol{x}_i, \boldsymbol{\lambda})\\ \boldsymbol{\lambda}_* &= (\boldsymbol{J_C}(\boldsymbol{x}_i)\zeta)^{-1}\boldsymbol{J_C}(\boldsymbol{x}_i)\nu\\ \eta &= -\nu + \zeta\boldsymbol{\lambda}_*\\ \eta_* &= -\nu\\ \tau_* &= \boldsymbol{L_x}^{-1}G(\boldsymbol{x}_i)[\eta_*^{[2]}]\\ \tau &= \tau_* - \zeta(\boldsymbol{J_C}(\boldsymbol{x}_i)\zeta)^{-1}\boldsymbol{J_C}(\boldsymbol{x}_i)[\tau_*]\end{aligned}$$

In existing Rayleigh quotient literature convergence proofs are usually discussed in term of distance of $\boldsymbol{\lambda}$ to eigenvalues. (6.6) replaces the distance to eigenvalues discussion. If B is a linear operator and B has bounded inverse then $||B^{-1}\psi|| \leq ||B^{-1}||_{op}||\psi||$ and

$$||B^{-1}(B\psi)|| \leq ||B^{-1}||_{op}||B\psi||$$

and hence $||B\psi|| \geq \frac{1}{||B^{-1}||_{op}}||\psi||$. A continuous family of operators $B(\boldsymbol{x})$ in a bounded neighborhood of $\boldsymbol{v}$ would have $\frac{1}{||B^{-1}(\boldsymbol{x})||_{op}}$ locally bounded by the same constant C. So (6.6) only requires the inverse of $\Pi_{\boldsymbol{L_\lambda}}\boldsymbol{L_x}$ to be locally bounded. When $B = \Pi_{\boldsymbol{L_\lambda}}(A - \boldsymbol{\lambda I})$, bounds on B can be discussed in term of distance from $\boldsymbol{\lambda}$ to eigenvalues. This condition allows us to translate an estimate of $||\Pi_{\boldsymbol{L_\lambda}}\boldsymbol{L_x}(\boldsymbol{x}, \mathcal{R}(\boldsymbol{x}))(\boldsymbol{x} - \boldsymbol{v})||$ to an estimate of convergent rate of $||\boldsymbol{x} - \boldsymbol{v}||$.

When $\boldsymbol{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ is explicit, we can take $\mathcal{R}(x) = \boldsymbol{H}^-(x)\boldsymbol{F}(x)$. When it is not explicit, $\mathcal{R}$ is most often given implicitly as a solution to a system

$$\mathcal{N}(\boldsymbol{x}, \boldsymbol{\lambda}) = 0$$

where $\mathcal{N}$ is a map from $E_{in} \oplus E_L$ to E_L with continuous derivatives up to degree two such that if $(\boldsymbol{x}, \boldsymbol{\lambda})$ satisfy the general Lagrangian system $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = 0$, then $\mathcal{N}(\boldsymbol{x}, \boldsymbol{\lambda}) = 0$. Assuming $\boldsymbol{J}_{\mathcal{N}}$ is of rank $\dim E_L$ in a neighborhood of $(\boldsymbol{v}, \boldsymbol{\mu})$ and $\mathcal{R}$ is the implicit function solution. Then $\mathcal{N}(\boldsymbol{x}, \mathcal{R}(\boldsymbol{x})) = 0$ for all $\boldsymbol{x}$ in a neighborhood of $\boldsymbol{v}$ and hence:

$$(6.10)\quad \boldsymbol{J}_{\mathcal{R}} = -\mathcal{N}_{\boldsymbol{\lambda}}^{-1}\mathcal{N}_{\boldsymbol{x}}$$

In section on nonlinear eigenvalue problem below we give a concrete example how this is done.

Proof. We have been trying to show evidence that Rayleigh and Rayleigh-Chebyshev iterations are modified forms of Newton-Raphson and Chebyshev iterations for $\mathcal{L}$. As expected our proofs will involve Taylor series expansion of $\mathcal{L}$ to different degrees. We first look at the Rayleigh case where Taylor expansion to degree 1 gives

$$\begin{aligned}0 = \boldsymbol{L}(\boldsymbol{v}, \boldsymbol{\mu}) = \boldsymbol{L}(\boldsymbol{x}_i, \mathcal{R}(\boldsymbol{x}_i)) + \boldsymbol{L_x}(\boldsymbol{x}_i, \mathcal{R}(\boldsymbol{x}_i))(\boldsymbol{v} - \boldsymbol{x}_i) + \boldsymbol{L_\lambda}(\boldsymbol{x}_i, \mathcal{R}(\boldsymbol{x}_i))(\boldsymbol{\mu} - \mathcal{R}(\boldsymbol{x}_i)) +\\ O(||\begin{pmatrix}\boldsymbol{x}_i\\ \boldsymbol{\mu}\end{pmatrix} - \begin{pmatrix}\boldsymbol{v}\\ \mathcal{R}(\boldsymbol{x}_i)\end{pmatrix}||^2))\end{aligned}$$

By the C^1 assumption of $\mathcal{R}$ and note $\boldsymbol{\mu} = \mathcal{R}(\boldsymbol{v})$ we have $||\boldsymbol{\mu} - \mathcal{R}(x_i)|| \leq C_1||\boldsymbol{v} - \boldsymbol{x}_i||$. So the last term above is $O(||\boldsymbol{v} - \boldsymbol{x}_i||^2)$. Applying $\Pi_{\boldsymbol{L_\lambda}}(\boldsymbol{x}_i)$ to both sides and rearrange the terms:

$$\Pi_{\boldsymbol{L_\lambda}}\boldsymbol{L_x}(\boldsymbol{x}_i, \mathcal{R}(\boldsymbol{x}_i))(\boldsymbol{x}_i - \boldsymbol{v}) = \Pi_{\boldsymbol{L_\lambda}}\boldsymbol{L}(\boldsymbol{x}_i, \mathcal{R}(\boldsymbol{x}_i)) + O(||\boldsymbol{v} - \boldsymbol{x}_i||^2)$$

Note that we use $\Pi_{\boldsymbol{L_\lambda}}(\boldsymbol{x}_i)\boldsymbol{L_\lambda} = 0$ in the above, plus continuous derivative condition of $\Pi_{\boldsymbol{L_\lambda}}$ for the last term. Expressing $(\boldsymbol{x}_{i+1} - \boldsymbol{v}) = \boldsymbol{x}_i - \boldsymbol{v} + \eta + O(||\eta||^2)$ using the first order retraction and apply $\Pi_{\boldsymbol{L_\lambda}}\boldsymbol{L_x}(\boldsymbol{x}_i, \mathcal{R}(\boldsymbol{x}_i))$ to both sides

$$(6.11)\quad \begin{aligned}\Pi_{\boldsymbol{L_\lambda}}\boldsymbol{L_x}(\boldsymbol{x}_i, \mathcal{R}(\boldsymbol{x}_i))(\boldsymbol{x}_{i+1} - \boldsymbol{v}) = \Pi_{\boldsymbol{L_\lambda}}\boldsymbol{L}(\boldsymbol{x}_i, \mathcal{R}(\boldsymbol{x}_i)) + \Pi_{\boldsymbol{L_\lambda}}\boldsymbol{L_x}(\boldsymbol{x}_i, \mathcal{R}(\boldsymbol{x}_i))[\eta] +\\ +O(||\boldsymbol{v} - \boldsymbol{x}_i||^2) + O(||\eta||^2)\end{aligned}$$

Choose η satisfying:

$$\Pi_{\boldsymbol{L_\lambda}} \boldsymbol{L_x} \eta = -\Pi_{\boldsymbol{L_\lambda}} \boldsymbol{L}(\boldsymbol{x}_i, \mathcal{R}(\boldsymbol{x}_i))$$

The right-hand side is $O||\boldsymbol{x}_i - \boldsymbol{v}||$, so (6.6) shows $O(||\eta||^2)$ is also $O(||\boldsymbol{x}_i - \boldsymbol{v}||^2)$. Finally:

$$||\boldsymbol{x}_{i+1} - \boldsymbol{v}|| \leq \frac{1}{C} ||\Pi_{\boldsymbol{L_\lambda}} \boldsymbol{L_x}(\boldsymbol{x})(\boldsymbol{x}_{i+1} - \boldsymbol{v})|| = O||\boldsymbol{x}_i - \boldsymbol{v}||^2.$$

(The first estimate follows from (6.6) and the second is from (6.11) and the choice of η). This proves quadratic convergence of RQI.

For the Chebyshev case we expand the series to second order. We denote $\boldsymbol{L_{xx}}, \boldsymbol{L_{x\lambda}}, \boldsymbol{L_{\lambda x}}, \boldsymbol{L_{\lambda\lambda}}$ to be tensors representing higher order the partial derivatives of $\boldsymbol{L}$. Put $(\boldsymbol{x}_i, \mathcal{R}(\boldsymbol{x}_i)) = \hat{\boldsymbol{x}}_i$ we have

$$\begin{aligned} 0 = \boldsymbol{L}(\boldsymbol{v}, \boldsymbol{\mu}) &= \boldsymbol{L}(\hat{\boldsymbol{x}}_i) + \boldsymbol{L_x}(\hat{\boldsymbol{x}}_i)(\boldsymbol{v} - \boldsymbol{x}_i) + \boldsymbol{L_\lambda}(\hat{\boldsymbol{x}}_i)(\boldsymbol{\mu} - \mathcal{R}(\boldsymbol{x}_i)) \\ &+ \frac{1}{2} \boldsymbol{L}_{xx}(\hat{\boldsymbol{x}}_i)(\boldsymbol{v} - \boldsymbol{x}_i)^{[2]} + \boldsymbol{L_{x\lambda}}(\hat{\boldsymbol{x}}_i)[(\boldsymbol{v} - \boldsymbol{x}_i)(\boldsymbol{\mu} - \mathcal{R}(\boldsymbol{x}_i))] + \\ &\frac{1}{2} \boldsymbol{L_{\lambda\lambda}}(\hat{\boldsymbol{x}}_i)(\boldsymbol{\mu} - \mathcal{R}(\boldsymbol{x}_i))^{[2]} + O(||\boldsymbol{v} - \boldsymbol{x}_i||^3) \end{aligned}$$

The residual is $O(||\boldsymbol{v} - \boldsymbol{x}_i||^3)$ in the above because we apply the bound $||\mu - \mathcal{R}(\boldsymbol{x}_i)|| \leq C_1 ||\boldsymbol{v} - \boldsymbol{x}_i||$ similar to the Rayleigh case. Again, move the term $\boldsymbol{L_x}(\hat{\boldsymbol{x}}_i)(\boldsymbol{v} - \boldsymbol{x}_i)$ to the left-hand side then apply $\Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i)$:

$$\begin{aligned} \Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i) \boldsymbol{L_x}(\hat{\boldsymbol{x}}_i)(\boldsymbol{x}_i - \boldsymbol{v}) &= \Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i) \boldsymbol{L}(\hat{\boldsymbol{x}}_i) + \frac{1}{2} \Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i) \boldsymbol{L}_{xx}(\hat{\boldsymbol{x}}_i)(\boldsymbol{v} - \boldsymbol{x}_i)^{[2]} + \\ \Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i) \boldsymbol{L_{x\lambda}}(\hat{\boldsymbol{x}}_i)[(\boldsymbol{v} - \boldsymbol{x}_i)(\mu - \mathcal{R}(\boldsymbol{x}_i))] &+ \frac{1}{2} \Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i) \boldsymbol{L_{\lambda\lambda}}(\hat{\boldsymbol{x}}_i)(\mu - \mathcal{R}(\boldsymbol{x}_i))^{[2]} + O(||\boldsymbol{v} - \boldsymbol{x}_i||^3) \end{aligned} \tag{6.12}$$

We will choose the next iteration $\boldsymbol{x}_{i+1}$ by choosing η_* and $T[\eta_*]$:

$$\boldsymbol{x}_{i+1} = \boldsymbol{r}(\boldsymbol{x}_i, \tau) = \boldsymbol{r}(\boldsymbol{x}_i, \eta_* + T[\eta_*^{[2]}]) = \boldsymbol{x}_i + \eta_* + T[\eta_*^{[2]}] + O(||\tau||^3)$$

Using the two expressions below, substitute in (6.12):

$$\boldsymbol{x}_i - \boldsymbol{v} = (\boldsymbol{x}_{i+1} - \boldsymbol{v}) - \tau - O(||\tau||^3)$$

$$\boldsymbol{\mu} - \mathcal{R}(\boldsymbol{x}_i) = (\boldsymbol{\mu} - \mathcal{R}(\boldsymbol{x}_{i+1})) + \boldsymbol{J}_{\mathcal{R}}(\hat{\boldsymbol{x}}_i)\tau + O(||\tau||^2)$$

Expand and collect the terms with a factor of $\boldsymbol{v} - \boldsymbol{x}_{i+1}$ or $(\boldsymbol{\mu} - \mathcal{R}(\boldsymbol{x}_{i+1}))$ to a group A, while leaving the terms with only τ factors, together with any cubic term in the expression we get

$$\begin{aligned} \Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i) \boldsymbol{L_x}(\hat{\boldsymbol{x}}_i)(\boldsymbol{x}_{i+1} - \boldsymbol{v}) - \Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i) \boldsymbol{L_x}(\hat{\boldsymbol{x}}_i)\eta_* - \Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i) \boldsymbol{L_x}(\hat{\boldsymbol{x}}_i) T[\eta_*^{[2]}] = A + \Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i) \boldsymbol{L}(\hat{\boldsymbol{x}}_i) + \\ \frac{1}{2} \Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i) \boldsymbol{L}_{xx}(\hat{\boldsymbol{x}}_i)\tau^{[2]} + \Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i) \boldsymbol{L_{x\lambda}}(\hat{\boldsymbol{x}}_i)[\tau \boldsymbol{J}_{\mathcal{R}} \tau] + \frac{1}{2} \Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i) \boldsymbol{L_{\lambda\lambda}}(\hat{\boldsymbol{x}}_i)[(\boldsymbol{J}_{\mathcal{R}}\tau)^{[2]}] \\ + O(||\boldsymbol{v} - \boldsymbol{x}_i||^3) + O(||\tau||^3) \end{aligned} \tag{6.13}$$

We do not need to list the terms of A explicitly. It is sufficient to know that A is sum of terms of total order two in term of τ or $(\boldsymbol{v} - \boldsymbol{x}_{i+1})$ or $(\boldsymbol{\mu} - \mathcal{R}(\boldsymbol{x}_{i+1}))$ and containing at least one term of the last two forms. The continuous derivative assumption shows

$$||A|| \leq D ||\boldsymbol{x}_{i+1} - \boldsymbol{v}|| \times ||\boldsymbol{x}_i - \boldsymbol{v}||$$

for some constant D near $(\boldsymbol{v}, \boldsymbol{\mu})$. If $\tau = O(||\boldsymbol{x}_i - \boldsymbol{v}||)$ (which will turn out to be the case), $||\Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i) \boldsymbol{L_x}(\hat{\boldsymbol{x}}_i)(\boldsymbol{x}_{i+1} - \boldsymbol{v}) - A||$ is dominated by $\Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i) \boldsymbol{L_x}(\hat{\boldsymbol{x}}_i)(\boldsymbol{x}_{i+1} - \boldsymbol{v})$, therefore:

$$||\Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i) \boldsymbol{L_x}(\hat{\boldsymbol{x}}_i)(\boldsymbol{x}_{i+1} - \boldsymbol{v}) - A|| \leq C_2 ||\boldsymbol{x}_{i+1} - \boldsymbol{v}||$$

with $||C_2|| > 0$, hence we get an estimate for $\boldsymbol{x}_{i+1} - \boldsymbol{v}$. To eliminate the first order terms we choose η_* such that

$$\Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i)\boldsymbol{L_x}(\hat{\boldsymbol{x}}_i)\eta_* + \Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i)\boldsymbol{L}(\hat{\boldsymbol{x}}_i) = 0$$

To eliminate the second order terms we need:

$$\Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i)T\eta_*^{[2]} + \frac{1}{2}\Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i)\boldsymbol{L}_{xx}(\hat{\boldsymbol{x}}_i)(\eta_*) + \Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i)\boldsymbol{L_{x\lambda}}(\hat{\boldsymbol{x}}_i)[\eta_*\boldsymbol{J}_{\mathcal{R}}\eta_*] + \frac{1}{2}\Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i)\boldsymbol{L_{\lambda\lambda}}(\hat{\boldsymbol{x}}_i)(\boldsymbol{J}_{\mathcal{R}}\eta_*)^{[2]} = 0.$$

With this choice, $\tau = O(||\boldsymbol{x}_i - \boldsymbol{v}||)$. From our argument on the dominance of $\Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i)(\boldsymbol{x}_{i+1} - \boldsymbol{v})$ over A cubic convergence follows. The Schur form solution is just a simple substitution and check. If $G(\boldsymbol{v}) = 0$ and instead of the Chebyshev step, we only apply RQI step then instead of (6.13) we have

$$\begin{aligned}\Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i)(\boldsymbol{x}_{i+1} - \boldsymbol{v}) = A + \Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i)\boldsymbol{L}(\hat{\boldsymbol{x}}_i) + \frac{1}{2}\Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i)\boldsymbol{L}_{xx}(\hat{\boldsymbol{x}}_i)\eta^{[2]} +& \\ \Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i)\boldsymbol{L_{x\lambda}}(\hat{\boldsymbol{x}}_i)[\eta\boldsymbol{J}_{\mathcal{R}}\eta] + \frac{1}{2}\Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i)\boldsymbol{L_{\lambda\lambda}}(\hat{\boldsymbol{x}}_i)(\boldsymbol{J}_{\mathcal{R}}\eta)^{[2]} +& \\ \Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i)\boldsymbol{L_x}(\hat{\boldsymbol{x}}_i)\eta + O(||\boldsymbol{v} - \boldsymbol{x}_i||^3) + O(||\eta||^3)& \\ = A + \Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i)\boldsymbol{L}(\hat{\boldsymbol{x}}_i) + \Pi_{\boldsymbol{L_\lambda}}(\hat{\boldsymbol{x}}_i)\boldsymbol{L_x}(\hat{\boldsymbol{x}}_i)\eta - G(\hat{\boldsymbol{x}}_i)\eta^{[2]} + O(||\boldsymbol{v} - \boldsymbol{x}_i||^3) + O(||\eta||^3)&\end{aligned} \tag{6.14}$$

Since G has continuous derivative and $G(\boldsymbol{v}) = 0$:

$$||G(\hat{\boldsymbol{x}}_i)\eta^{[2]}|| = ||(G(\hat{\boldsymbol{x}}_i) - G(\boldsymbol{v}))\eta^{[2]}|| \leq C_G||\hat{\boldsymbol{x}}_i - \boldsymbol{v}||(||\eta||^2)$$

for some constant C_G and from here cubic convergence of RQI follows.

For the Schur form of the Chebyshev step, while we can project both η_* and $T[\eta_*^{[2]}]$ to the tangent space, it we save one step by projecting only τ_*. With this, we can choose $\eta_* = -\nu$ or $\eta_* = -\nu + \eta\boldsymbol{\lambda}_i$. For the explicit case, we just need to substitute $\boldsymbol{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = \boldsymbol{F}(\boldsymbol{x}) - \boldsymbol{H}(\boldsymbol{x})\boldsymbol{\lambda}$ and simplify the algebraic expressions. ∎

The theorem shows there much freedom in choosing $\mathcal{R}$. The main requirement is consistency: if $(\boldsymbol{v}, \boldsymbol{\mu})$ is a solution then $\mathcal{R}(\boldsymbol{v}) = \mu$. So even for the explicit case $\mathcal{R}$ does not need to come from a pseudo inverse. We numerically tested this general RQI with $\boldsymbol{L}$ is given by a (tensor) Taylor series up to degree 3 and constraint given by products of spheres. For each sphere a Rayleigh functional is given by $p_i(\boldsymbol{x})\boldsymbol{L}(\boldsymbol{x}, \boldsymbol{\lambda})$, where p_i is the projection to coordinate of the i^{th} sphere. We get convergence as expected. We summarize the discussion on the constraint optimization case earlier in the following proposition:

Proposition 6.3. *For a constrained optimization problem when $\boldsymbol{F} = \nabla f, \boldsymbol{H} = \boldsymbol{J}_{\boldsymbol{C}}^T$ and $\boldsymbol{H}^- = (\boldsymbol{J_C}\boldsymbol{J}_{\boldsymbol{C}}^T)^{-1}\boldsymbol{J_C}$ we have $\boldsymbol{\lambda} = \mathcal{R}(\boldsymbol{x}) = (\boldsymbol{J_C}\boldsymbol{J}_{\boldsymbol{C}}^T)^{-1}\boldsymbol{J_C}\nabla f$, $\Pi_{\boldsymbol{H}}\boldsymbol{F}(\boldsymbol{x})$ is the projection to $T\mathcal{M}_{\boldsymbol{x}}$, $\Pi_{\boldsymbol{H}}\boldsymbol{F}(\boldsymbol{x})$ is the Riemannian gradient and $\Pi_{\boldsymbol{H}}(\boldsymbol{x})\boldsymbol{L_x}(\boldsymbol{x})$ is the Riemannian Hessian of the embedded manifold. The RQI iteration equation is the Riemannian Newton equation on $\mathcal{M}$ and the Rayleigh-Chebyshev equation is a second order iteration on the same manifold.*

If not using Schur form, we can solve the system directly as a linear system from $T\mathcal{M}_{\boldsymbol{x}_i}$ to $\mathrm{Im}(\Pi_{\boldsymbol{L_\lambda}})$. This is essentially an extension of the Riemannian Newton approach. We note an important feature in the proof is the use of $\Pi_{\boldsymbol{L_\lambda}}$ to eliminate the term $\boldsymbol{L_\lambda}(\hat{\boldsymbol{x}}_i)(\mathcal{R}(\boldsymbol{x}_i) - \mathcal{R}(\boldsymbol{v})$. To solve the updating equations, as mentioned we can parameterize η using $\boldsymbol{J_C}\eta = 0$ or use the Schur form. For the first case since we may not have a convenient matrix for $\mathrm{Im}(\Pi_{\boldsymbol{L_\lambda}})$, a strategy could be to start with a basis for $\mathrm{Ker}(\boldsymbol{J_C})$ and map it by $\Pi_{\boldsymbol{L_\lambda}}\boldsymbol{L_x}$ to a basis of $\mathrm{Im}(\Pi_{\boldsymbol{L_\lambda}})$. Representing $\Pi_{\boldsymbol{L_\lambda}}\boldsymbol{L}(\boldsymbol{x}_i)$ in this basis we can solve the updating equation. If we look at the Schur form, beside solving $\boldsymbol{L_x}$ for ν and ζ, we also need to compute $\boldsymbol{\lambda}_* = (\boldsymbol{J_C}\zeta)^{-1}\boldsymbol{J_C}\nu$. A natural question is if we can use $\mathcal{R}(\boldsymbol{x})$ instead of $\boldsymbol{\lambda}_*$. The resulting $-\nu + \zeta\mathcal{R}(\boldsymbol{x}_i)$ is no longer on the tangent space, but would be close so if the retraction is extended near $T\mathcal{M}$ in E_{in}, the iterative process could converge. [10] showed a quasi-Newton method based on this Rayleigh quotient converges superlinearly in general. We analyzed the use of this approximation of $\boldsymbol{\lambda}_*$ in the linear constraint example below.

Algorithm 6.1 Rayleigh quotient iteration for explicit Lagrangians in Schur form

Initialize $\boldsymbol{x}_0$
$i \leftarrow 0$
$\zeta \leftarrow SMALL\_NUMBER$
$err \leftarrow LARGE\_NUMBER$
while not Terminal(i, ζ, err) **do**
  Compute $\boldsymbol{\lambda}_i = \mathcal{R}(\boldsymbol{x}_i) = \boldsymbol{H}^{-}(\boldsymbol{x}_i)\boldsymbol{F}(\boldsymbol{x}_i)$
  Solve for ζ in $\boldsymbol{L}_{\boldsymbol{x}}(\boldsymbol{x}_i, \boldsymbol{\lambda}_i)\zeta = \boldsymbol{H}(\boldsymbol{x}_i)$
  Solve for ν in $\boldsymbol{L}_{\boldsymbol{x}}(\boldsymbol{x}_i, \boldsymbol{\lambda}_i)\nu = \boldsymbol{F}(\boldsymbol{x}_i)$
  Compute $\boldsymbol{\lambda}_* = (\boldsymbol{J}_{\boldsymbol{C}}\zeta)^{-1}\boldsymbol{J}_{\boldsymbol{C}}(\nu)$
  Compute $\eta \leftarrow -\nu + \zeta\boldsymbol{\lambda}_*$
  Compute $\boldsymbol{x}_{i+1} \leftarrow \boldsymbol{r}(\boldsymbol{x}_i, \eta)$
  Compute $err \leftarrow \boldsymbol{L}(\boldsymbol{x}_{i+1}, \boldsymbol{\lambda}_i)$
  $i \leftarrow i + 1$
end while

Algorithm 6.2 Rayleigh-Chebyshev iteration for explicit Lagrangians in Schur form

Initialize $\boldsymbol{x}_0$
$i \leftarrow 0$
$\zeta = SMALL\_NUMBER$
while not Terminal(i, η, err) **do**
  Compute $\boldsymbol{\lambda}_i = \mathcal{R}(\boldsymbol{x}_i) = \boldsymbol{H}^{-}(\boldsymbol{x}_i)\boldsymbol{F}(\boldsymbol{x}_i)$
  Solve for ζ in $\boldsymbol{L}_{\boldsymbol{x}}(\boldsymbol{x}_i, \boldsymbol{\lambda}_i)\zeta = \boldsymbol{H}(\boldsymbol{x}_i)$
  Solve for ν in $\boldsymbol{L}_{\boldsymbol{x}}(\boldsymbol{x}_i, \boldsymbol{\lambda}_i)\nu = \boldsymbol{F}(\boldsymbol{x}_i)$
  Compute $\eta_* \leftarrow -\nu + \zeta\boldsymbol{\lambda}_i$
  Compute $T\eta_*^{[2]}$ as solution to $\boldsymbol{L}_{\boldsymbol{x}}T\eta_*^{[2]} = -[\frac{1}{2}\boldsymbol{J}_F^{(2)}\eta_*^{[2]} - (\boldsymbol{J}_{\boldsymbol{H}}\eta_*)(\boldsymbol{J}_{\mathcal{R}}\eta_*) - \frac{1}{2}\boldsymbol{J}_{\boldsymbol{H}}^{(2)}\eta_*^{[2]}\mathcal{R}]$
  Compute $\tau_* = \eta_* + T\eta_*^{[2]}$
  Compute $\tau = \tau_* - \boldsymbol{J}_{\boldsymbol{C}}(\boldsymbol{J}_{\boldsymbol{C}}\zeta)^{-1}\boldsymbol{J}_{\boldsymbol{C}}(\tau_*)$
  Compute $\boldsymbol{x}_{i+1} \leftarrow \boldsymbol{r}(\boldsymbol{x}_i, \tau)$, $\boldsymbol{r}$ is the retraction
  Compute $err \leftarrow \boldsymbol{L}(\boldsymbol{x}_{i+1}, \boldsymbol{\lambda}_i)$ $i \leftarrow i + 1$
end while

7. Examples. The python code for these examples could be found at [15]. The code is set up for a general framework but for matrix or higher tensor the user needs to provide his own method. We consulted [5] and [19]. To call the functions the user needs to specify the constraint in a constraint object and the function as well as its partial derivatives in a Lagrangian object. The solver is mainly of Schur form, but we also show the tangent form for Stiefel manifold. We also need a solver for $(\boldsymbol{J}_{\boldsymbol{C}}\zeta)\boldsymbol{J}_{\boldsymbol{C}}\nu$ in some cases. For left inverse we use $\boldsymbol{H}^{-} = (\boldsymbol{H}^{\dagger}\boldsymbol{H})^{-1}\boldsymbol{H}^{\dagger}$ for $\boldsymbol{H}^{\dagger}$ such that $(\boldsymbol{H}^{\dagger}\boldsymbol{H})$ is invertible. We use different choices of $\boldsymbol{H}^{\dagger}$ in our examples. Our aim is to verify the result, so we have not spent much effort to optimize the code.

7.1. Optimization on embedded manifolds. As mentioned in this case $\boldsymbol{H} = \boldsymbol{J}_{\boldsymbol{C}}^T$ and the Rayleigh iteration equation is exactly the Riemannian Newton equation. RQI provides a way to find critical points. As in the unconstrained case, a critical point finding method needs to be used with a gradient method to find local optimal points. There is an extensive literature on quasi-Newton method in constrained optimization, including Sequential Quadratic Programming method (SQP) where an approximation of $\boldsymbol{L}_{\boldsymbol{x}}$ is used in iterations. [6], chapter 10 has a number of examples using Schur form. As explained there, when the Schur form is sparse this algorithm becomes valuable.

7.2. Eigenvectors. First we consider the eigenvector problem under the quadratic constraint:

$$\frac{1}{2}(\boldsymbol{x}^T\boldsymbol{x} - 1) = 0$$

We apply all explicit Lagrangian algorithms listed here. We obtained convergent for all cases as expected with appropriate initial points. Care need to be taken in determine when $\boldsymbol{L}_x$ becomes singular as terminal condition. We already looked at Newton-Raphson for eigenvector in section 2 and have showed $\boldsymbol{x}_{i+1}$ is proportional to ζ. However, the second derivative

$$\begin{pmatrix} -2\eta\delta \\ \eta^T\eta \end{pmatrix}$$

is not dependent on A, so we do not have cubic convergence even for normal matrices. Applying (5.2), the Chebyshev adjustment involves one more matrix inversion $(A - \boldsymbol{\lambda I})^{-1}\eta$ and is simple to calculate.

We have showed our RQI is the classical RQI in this case. With Rayleigh quotient $(\boldsymbol{x}^T\boldsymbol{x})^{-1}\boldsymbol{x}^TA\boldsymbol{x}$

$$\boldsymbol{J}_{\mathcal{R}}(\boldsymbol{x})(\eta) = -(\boldsymbol{x}^T\eta + \eta^T\boldsymbol{x})\boldsymbol{x}^TA\boldsymbol{x} + \eta^TA\boldsymbol{x} + \boldsymbol{x}^TA\eta$$

When A is normal $\boldsymbol{J}_{\mathcal{R}}(\boldsymbol{v}) = 0$ hence $G(\boldsymbol{v}) = 0$, therefore we have cubic convergence.

For the nonnormal case the Chebyshev term is again proportional to $(A - \lambda\boldsymbol{I})^{-1}\eta$. We note we use the same $A - \lambda\boldsymbol{I}$ in the second step so intermediate data, for example LU factorization could be reused. It would be interesting to study global convergence property of the Rayleigh-Chebyshev algorithm.

Using the same $\boldsymbol{H}(\boldsymbol{x}) = \boldsymbol{x}$ but imposing the constraint:

$$\boldsymbol{z}^T\boldsymbol{x} = 1$$

instead of the quadratic constraint. For the left inverse we will use $\boldsymbol{H}^- = (\boldsymbol{H}^\dagger\boldsymbol{H})^{-1}\boldsymbol{H}^\dagger$. We can still use $\boldsymbol{H}^\dagger(\boldsymbol{x}) = \boldsymbol{x}^T$. The Rayleigh quotient will still be $(\boldsymbol{x}^T\boldsymbol{x})^{-1}(\boldsymbol{x}^TA\boldsymbol{x})$. In the code we also tested the case where $\boldsymbol{H}^\dagger(\boldsymbol{x}) = \boldsymbol{z}^T$, which gives the Rayleigh quotient $(\boldsymbol{z}^T\boldsymbol{x})^{-1}(\boldsymbol{z}^TA\boldsymbol{x})$. We note $\boldsymbol{J}_{\mathcal{R}}(\boldsymbol{x})(\eta) = \boldsymbol{z}^TA(\boldsymbol{I} - \boldsymbol{x}\boldsymbol{z}^T)\eta$ on the constraint manifold. A consequence is $\boldsymbol{J}_{\mathcal{R}}(\boldsymbol{v}) = 0$ if $\boldsymbol{z}^T$ is a left eigenvector. So in that case the Rayleigh quotient iteration converges cubically for the corresponding right eigenvector. This may not be of practical use but may be an interesting note in view of the two-sided Rayleigh quotient method below. The Rayleigh-Chebyshev iteration requires solving $(A - \mathcal{R}(\boldsymbol{x})\boldsymbol{I})^{-1}\eta$ as before.

7.3. Two-sided Rayleigh quotient. This algorithm by Ostrowski [16] has cubic convergent rate even for nonnormal matrix A. We show this is an example of our Lagrangian approach. Let

$$\boldsymbol{x} = \begin{pmatrix} u \\ v \end{pmatrix}$$

$$\boldsymbol{\lambda} = (\lambda, \mu)$$

We thus define E_{in}, E_{out} to be $\mathbb{R}^{2n}$ while E_L is $\mathbb{R}^2$.

$$\boldsymbol{F}(\boldsymbol{x}) = \begin{pmatrix} 0 & A^T \\ A & 0 \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} A^Tv \\ Au \end{pmatrix}$$

$$\boldsymbol{H}(\boldsymbol{x})(\boldsymbol{\lambda}) = \begin{pmatrix} v\lambda \\ u\mu \end{pmatrix}$$

$$\boldsymbol{C}(\boldsymbol{x}) = \begin{pmatrix} v^Tv - 1 \\ u^Tu - 1 \end{pmatrix}$$

We use again $\boldsymbol{H}^- = (\boldsymbol{H}^\dagger\boldsymbol{H})^{-1}\boldsymbol{H}^\dagger$ with $\boldsymbol{H}^\dagger$ is a map from E_{in} to E_L defined by $\boldsymbol{H}^\dagger\begin{pmatrix} a \\ b \end{pmatrix} = (u^Ta, v^Tb)$. We recover Ostrowski's algorithm from:

$$(\boldsymbol{H}^\dagger\boldsymbol{H}) = \begin{pmatrix} u^Tv & 0 \\ 0 & v^Tu \end{pmatrix}$$

$$\mathcal{R}(\boldsymbol{x}) = ((u^T A^T v)/(u^T v), (v^T A u)/(v^T u)).$$

While $\mathcal{R}$ has two components, they are in fact identical. Cubic convergent rate is a consequence of $\boldsymbol{J}_{\mathcal{R}}(\boldsymbol{v}) = 0$ at the eigenvector which we can verify directly. See [4] for an invariant subspace version.

7.4. General Lagrangian: Nonlinear eigenvalue problem. Newton-Raphson method was applied to nonlinear eigenvalue problem (also called λ-matrix problem) in [12]. With $\boldsymbol{\lambda}$ is a scalar, the equation has the form

$$\boldsymbol{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = P(\boldsymbol{\lambda})\boldsymbol{x}$$

Applying Algorithm 5.1 we have $\boldsymbol{L}_{\boldsymbol{x}} = P(\boldsymbol{\lambda})$, $\boldsymbol{L}_{\boldsymbol{\lambda}} = P'(\boldsymbol{\lambda})\boldsymbol{x}$ and the Schur complement is either $\boldsymbol{x}^T P^{-1}(\boldsymbol{\lambda})P'(\boldsymbol{\lambda})\boldsymbol{x}$ for the quadratic constraint or $\boldsymbol{z}^T P^{-1}(\boldsymbol{\lambda})P'(\boldsymbol{\lambda})\boldsymbol{x}$ for the linear constraint and reduces to algorithm 4.7 in the above citation. To apply Theorem 6.2, we need a Rayleigh quotient, which we define to be the $\boldsymbol{\lambda}$ satisfying of the system

$$\mathcal{N}(\boldsymbol{x}, \boldsymbol{\lambda}) = \boldsymbol{x}^H P(\boldsymbol{\lambda})\boldsymbol{x}$$

so

$$\boldsymbol{J}_{\mathcal{R}}(\boldsymbol{x}) = -P_{\boldsymbol{\lambda}}^{-1}\boldsymbol{x}^H(P + P^H)$$

In particular if P is normal $\boldsymbol{J}_{\mathcal{R}}(\boldsymbol{v}) = 0$ for any nonlinear eigenvector. The RQI update reduces to algorithm 4.9 in [12]. For non symmetric case we can define $\boldsymbol{x} = \begin{pmatrix} u \\ v \end{pmatrix}$ as before and

$$\hat{P} = \begin{pmatrix} 0 & P^H \\ P & 0 \end{pmatrix}$$

Take $\mathcal{N}(\boldsymbol{x}, \boldsymbol{\lambda}) = v^H P u + u^H P^H v$ we recover the nonlinear two-sided RQI with cubic convergence. For the quadratic eigenvalue problem, the Rayleigh-Chebyshev algorithm could be competitive as we only need to solve for $\boldsymbol{\lambda}$ and inverting $P_{\boldsymbol{\lambda}}$ once.

7.5. Vector Lagrangian with various constraints. We tested the algorithms with two nonlinear constraints. If $\boldsymbol{x}$ is if of size n and $\boldsymbol{C}$ consists of k constraints, we assume that it has been solved for the first $n-k$ variables, so $\boldsymbol{x}[n-k+i] = c_i(\boldsymbol{x}[0:n-k])$. Locally any constraint could be transformed to this form. We use the orthographic retraction, which is simpler in this case. The constraint functions we tested are of form:

$$x[n_f] = x[0:n_f] + \sin(x[0:n_f])$$
$$x[n_f+1] = x[0:n_f] + \cos(x[0:n_f])$$

We take $\boldsymbol{H}$ to be either a constant function, or a quadratic function. For $\boldsymbol{F}$ we take $\boldsymbol{F}(\boldsymbol{x}) = A\boldsymbol{x}$ or $\boldsymbol{F}(\boldsymbol{x}) = A\boldsymbol{x} + \sin(B\boldsymbol{x})$ for some square matrices A and B. The difficulty is with choosing the initial point, otherwise the algorithms converge sufficiently fast.

7.6. An experiment with linear constraints. We consider the case $E_{in} = E_{out} = \mathbb{R}^n, E_L = \mathbb{R}^p$ and $b \in \mathbb{R}^p$ with the constraint $\boldsymbol{C}(\boldsymbol{x}) = C\boldsymbol{x} - b$ and $\boldsymbol{H}(\boldsymbol{x}) = \boldsymbol{J}_{\boldsymbol{C}}^T = C^T$. Here C is a $p \times n$ matrix. As $\boldsymbol{J}_{\boldsymbol{H}}$ is zero $\boldsymbol{L}_{\boldsymbol{x}} = \boldsymbol{J}_{\boldsymbol{F}}$. The Rayleigh quotient is $\mathcal{R}(\boldsymbol{x}) = (C^T C)^{-1} C\boldsymbol{F}(\boldsymbol{x})$. However only $\boldsymbol{\lambda}_* = (C\zeta)^{-1}(C\nu)$ appear in the iteration, so we have a situation that has been analyzed in [6].

We did a little more analysis to understand the effect of replacing $\boldsymbol{\lambda}_*$ with $\boldsymbol{\lambda} = \mathcal{R}(\boldsymbol{x})$ in the Schur form solution $\eta = -\nu + \zeta\boldsymbol{\lambda}_*$. Recall the projection of x_0 to $\mathcal{M}$ in this case is

$$\Pi(\boldsymbol{x}) = \Pi_{\boldsymbol{H}}(\boldsymbol{x}) = \boldsymbol{I}_n - C^T(CC^T)^{-1}C$$

If $f(x) = \frac{1}{2}x^T A x + d^T x$ then $\boldsymbol{F}(x) = \nabla f = Ax + d$ and $\boldsymbol{L}(\boldsymbol{x}) = A\boldsymbol{x} - C^T\boldsymbol{\lambda}$. Here A is a positive definite matrix. The closed form solution to the constrained system:

$$A\boldsymbol{x} + d - C^T\boldsymbol{\lambda} = 0$$
$$C\boldsymbol{x} = b$$

is

$$\boldsymbol{x} = A^{-1}C^T(CAC^T)^{-1}b + A^{-1}C^T(CA^{-1}C^T)^{-1}CA^{-1}d.$$

If we isolate the tangent space in the Riemannian Newton method, we arrive at the exact solution in one step. The Schur form with

$$\eta = -\nu + \zeta\boldsymbol{\lambda}_*$$

also gives the exact solution after one iteration. If we use $\boldsymbol{\lambda}$ as a proxy for $\boldsymbol{\lambda}_*$, the Rayleigh quotient iteration with the retraction Π above gives an iterative process

$$\boldsymbol{x}_{i+1} = C^T(CC^T)^{-1}b - \Pi A^{-1}Ad - (\Pi A^{-1}\Pi A - \Pi)\boldsymbol{x}_i$$

When this series converges, it does converge linearly to the closed form solution, a fact not hard to establish but nontrivial at first look. This example shows $\boldsymbol{\lambda}_*$ is needed when we want quadratic convergence.

7.7. RQI on Stiefel manifolds. The constraint for Stiefel manifolds is $\frac{1}{2}(\boldsymbol{x}^T\boldsymbol{x} - \boldsymbol{I}_p)$. We will focus on the case $\boldsymbol{H} = \boldsymbol{J}_{\boldsymbol{C}}^T$. Here $E_{in} = E_{out} = M_{n,p}$ and E_L is the space of $p \times p$ symmetric matrices. The tangent space is defined by the equation:

$$\boldsymbol{J}_{\boldsymbol{C}}(\eta) = \frac{1}{2}(\boldsymbol{x}^T\eta + \eta^T\boldsymbol{x}).$$

So it could be represented as an average of right and left multiplication tensors. Its conjugate $\boldsymbol{J}_{\boldsymbol{C}}^T$ is the map $\gamma \mapsto \boldsymbol{x}\gamma$. So the Rayleigh quotient turns out to be:

$$\mathcal{R}(\boldsymbol{x}) = \frac{1}{2}(x^T\boldsymbol{F}(\boldsymbol{x}) + \boldsymbol{F}(\boldsymbol{x})^T\boldsymbol{x})$$

and the projection $\Pi_{\boldsymbol{H}}$ is

$$\xi \mapsto \xi - \frac{1}{2}\boldsymbol{x}\boldsymbol{x}^T\xi - \frac{1}{2}\boldsymbol{x}\xi^T\boldsymbol{x}$$

as seen in [3]. We tested the explicit Lagrangian with $\boldsymbol{F}(x) = A\boldsymbol{x} + b$ where A is an (n, p, n, p) tensor and b is a (n, p) matrix. For the Schur form, we need to compute $\zeta = \boldsymbol{L}_{\boldsymbol{x}}^{-1}\boldsymbol{J}_{\boldsymbol{C}}^T$ and $\nu = \boldsymbol{L}_{\boldsymbol{x}}^{-1}\boldsymbol{F}(\boldsymbol{x})$. We note ζ is an $(n, p, p(p+1)/2)$ tensor in this case. We form a (sparse) matrix formed by concatenating the vectorized $\boldsymbol{F}(\boldsymbol{x})$ and $\boldsymbol{J}_{\boldsymbol{C}}^T$ (represented as a tensor reshaped as an $(np, p(p+1)/2)$ matrix). That way we can solve for ν and η in the same step. $\boldsymbol{\lambda}_* = (\boldsymbol{J}_{\boldsymbol{C}}\zeta)^{-1}\boldsymbol{J}_{\boldsymbol{C}}\nu$ is a $(p(p+1)/2, p(p+1)/2)$ matrix, so $\zeta\boldsymbol{\lambda}_*$ is an (n, p) matrix. The Schur form requires solving a larger system with dimension np instead of $np - p(p+1)/2$ of the Riemannian Newton method, but if the codimension of the Stiefel manifold is not too big the Schur form could be a useful alternative. We also tested the solution in tangent form.

8. RQI on Grassmann manifolds. Functions on Grassmann manifolds could be considered as function on fixed rank matrices equivariant under right multiplication by invertible matrices, or on Stiefel manifolds equivariant under the orthogonal group. Here we assume $\boldsymbol{H}$ to be the right multiplication by $\boldsymbol{x}^T$. The orthogonal group O_p acts on E_{in}, E_{out} and E_L, generating vector fields on these spaces. Invariance under the action of O_p allows us to identify the tangent space of the Grassmann manifold with the space of $n \times p$ matrices η with $x^T\eta = 0$. The action on E_{out} define a subspace of $\mathrm{Im}(\Pi_{\boldsymbol{H}})$ orthogonal to the vector fields generated by the action. We call the projection to this space Π_G, which turns out to be $(\boldsymbol{I} - \boldsymbol{x}\boldsymbol{x}^T)$. We arrive at the equations:

$$\begin{aligned} \boldsymbol{L}_{\boldsymbol{x}}\eta &= \boldsymbol{J}_{\boldsymbol{F}}\eta - \eta\boldsymbol{x}^T\boldsymbol{F}(\boldsymbol{x}) \\ \Pi_G\boldsymbol{L}_{\boldsymbol{x}}\eta &= -\Pi_G\boldsymbol{F} \\ \boldsymbol{x}^T\eta &= 0 \end{aligned} \tag{8.1}$$

We can try to solve it in Schur form. $\zeta = \boldsymbol{L}_{\boldsymbol{x}}^{-1}\boldsymbol{H}$ is now a tensor. In general, it could not be expressed as a matrix multiplication. This is because we do not always have $\boldsymbol{J}_{\boldsymbol{F}}(\boldsymbol{x})(\psi\gamma) = (\boldsymbol{J}_{\boldsymbol{F}}(\boldsymbol{x})\psi)\gamma$ for $n \times p$ matrix ψ and $p \times p$ matrix γ. But when this is the case the Schur form is simpler:

$$\begin{aligned} \zeta &= \boldsymbol{L}_{\boldsymbol{x}}^{-1}\boldsymbol{x} \\ \nu &= \boldsymbol{L}_{\boldsymbol{x}}^{-1}\boldsymbol{F}(x) \\ \eta &= -\nu + \zeta(\boldsymbol{x}^T\zeta)^{-1}(\boldsymbol{x}^T\nu) \end{aligned} \tag{8.2}$$

which we can verify directly, using the associativity mentioned above. An important case is the sphere, as γ is then a scalar. Another important case is invariant subspace:

$$\begin{aligned} A\boldsymbol{x} - \boldsymbol{x}\Lambda &= 0 \\ \frac{1}{2}(\boldsymbol{x}^T\boldsymbol{x} - \boldsymbol{I}_p) &= 0 \end{aligned} \tag{8.3}$$

with $\boldsymbol{F}(\boldsymbol{x}) = A\boldsymbol{x}$. The Rayleigh quotient is $\mathcal{R}(\boldsymbol{x}) = \boldsymbol{x}^T A^T \boldsymbol{x}$. Now if $z = \boldsymbol{x} + \zeta\boldsymbol{\lambda}$ then

$$Az - z\boldsymbol{\lambda} = A\boldsymbol{x}$$

so we can take $\nu = \boldsymbol{x} + \boldsymbol{x}\boldsymbol{\lambda}$, hence:

$$\eta = -\boldsymbol{x} + \zeta(\boldsymbol{\lambda}_* - \mathcal{R}(x))$$

where $\boldsymbol{\lambda}_* = (\boldsymbol{x}^T\zeta)^{-1}(\boldsymbol{x}^T\nu)$. Thus $x_{i+1} = \zeta(\boldsymbol{\lambda}_* - \mathcal{R}(x))$ is in the same space spanned by ζ. So the result of [1] is the Schur form of the Rayleigh quotient algorithm. We are not sure that these are the only examples where ζ is representable as a matrix. Conceptually we can formulate a Rayleigh-Chebyshev iteration for Grassmannians, but we have not worked out the details.

We consider another example, finding critical point for the function:

$$\frac{1}{2}\operatorname{Tr}(\boldsymbol{x}^T L\boldsymbol{x}) + \frac{\alpha}{4}\rho(\boldsymbol{x})^T L^{-1}\rho(\boldsymbol{x})$$

$$\rho(\boldsymbol{x}) = \operatorname{diag}(\boldsymbol{x}\boldsymbol{x}^T)$$

with $\boldsymbol{x}^T\boldsymbol{x} = \boldsymbol{I}_p$ ([20], [5]). We get the gradient

$$\boldsymbol{F}(x) = L\boldsymbol{x} + \alpha\operatorname{diag}(L^{-1}\rho(\boldsymbol{x}))\boldsymbol{x} \tag{8.4}$$

and we use the $GL(p)$ equivariant form

$$\rho(\boldsymbol{x}) = \operatorname{diag}(\boldsymbol{x}(\boldsymbol{x}\boldsymbol{x}^T)^{-1}\boldsymbol{x}^T) \tag{8.5}$$

$\boldsymbol{F}$ is equivariant under the action of right multiplication. $\boldsymbol{J}_{\boldsymbol{F}}\eta$ is

$$\boldsymbol{J}_{\boldsymbol{F}}\eta = A\eta + 2\alpha\operatorname{diag}(L^{-1}\operatorname{diag}((\boldsymbol{I} - \boldsymbol{x}\boldsymbol{x}^T)\eta\boldsymbol{x}^T))\boldsymbol{x} + \alpha\operatorname{diag}(L^{-1}\rho(\boldsymbol{x}))\eta$$

Because of the middle term, $\boldsymbol{J}_{\boldsymbol{F}}$, ζ cannot be represented as a matrix multiplication. In the code we computed ζ as a tensor and found critical points as expected.

9. Discussion.

As high order method on manifolds is a natural next step after Newton method, we provide here the such iterations on embedded manifolds in the Rayleigh-Chebyshev method. We expect it could be extended to all manifolds with a connection. While we have mostly provided an analysis of embedded manifolds in this paper, a similar analysis for quotient manifolds may be fruitful and may yield results that could be applicable to quotients of submanifolds, for example to flag manifolds.

REFERENCES

[1] P. Absil, R. Mahony, R. Sepulchre, and P. Van Dooren, *A grassmann–rayleigh quotient iteration for computing invariant subspaces*, SIAM Rev., 44 (2002), pp. 57–73, https://doi.org/10.1137/S0036144500378648.

[2] P. Absil and J. Malick, *Projection-like retractions on matrix manifolds*, SIAM J. Optim, 22 (2012), pp. 135–158, https://doi.org/10.1137/100802529.

[3] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, Princeton, NJ, USA, 2007.

[4] P.-A. Absil and P. Van Dooren, *Two-sided grassmann-rayleigh quotient iteration*, Numer. Math., 114 (2010), pp. 549–571, https://doi.org/10.1007/s00211-009-0266-y.

[5] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, *Manopt, a Matlab toolbox for optimization on manifolds*, Journal of Machine Learning Research, 15 (2014), pp. 1455–1459, http://www.manopt.org.

[6] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004, https://doi.org/10.1017/CBO9780511804441.

[7] V. Candela and A. Marquina, *Recurrence relations for rational cubic methods ii: The chebyshev method*, Computing, 45 (1990), pp. 355–367, https://doi.org/10.1007/BF02238803.

[8] A. Edelman, T. A. Arias, and S. T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 303–353, https://doi.org/10.1137/S0895479895290954.

[9] D. Gabay, *Minimizing a differentiable function over a differential manifold*, Journal of Optimization Theory and Applications, 37 (1982), pp. 177–219, https://doi.org/10.1007/BF00934767.

[10] D. Gabay, *Reduced quasi-Newton methods with feasibility improvement for nonlinearly constrained optimization*, Springer Berlin Heidelberg, Berlin, Heidelberg, 1982, pp. 18–44, https://doi.org/10.1007/BFb0120946.

[11] D. Gabay and D. Luenberger, *Efficiently converging minimization methods based on the reduced gradient*, SIAM J. Control Optim., 14 (1976), pp. 42–61, https://doi.org/10.1137/0314004.

[12] S. Güttel and F. Tisseur, *The nonlinear eigenvalue problem*, Acta Numer., 26 (2017), p. 1–94, https://doi.org/10.1017/S0962492917000034.

[13] M. R. Hestenes, *Multiplier and gradient methods*, J. Optim. Theory Appl., 4 (1969), pp. 303–320, https://doi.org/10.1007/BF00927673.

[14] P. Lancaster, *Some applications of the newton-raphson method to non-linear matrix problems*, Proc. Roy. Soc. London. Ser. A., 271 (1963), pp. 324–331, https://doi.org/10.1098/rspa.1963.0021.

[15] D. Nguyen, *Project lagrange_rayleigh*. https://github.com/dnguyend/lagrange_rayleigh, 2019.

[16] A. M. Ostrowski, *On the convergence of the rayleigh quotient iteration for the computation of the characteristic roots and vectors. iii*, Arch. Ration. Mech. Anal., 3 (1959), pp. 325–340, https://doi.org/10.1007/BF00284184.

[17] M. J. D. Powell, *Algorithms for nonlinear constraints that use lagrangian functions*, Math. Program., 14 (1978), pp. 224–248, https://doi.org/10.1007/BF01588967.

[18] R. A. Tapia, *Diagonalized multiplier methods and quasi-newton methods for constrained optimization*, J. Optim. Theory Appl., 22 (1977), pp. 135–194, https://doi.org/10.1007/BF00933161.

[19] J. Townsend, N. Koep, and S. Weichwald, *Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation*, J. Mach. Learn. Res., 17 (2016), pp. 1–5, http://jmlr.org/papers/v17/16-177.html.

[20] Z. Zhao, Z. Bai, and X. Jin, *A riemannian newton algorithm for nonlinear eigenvalue problems*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 752–774, https://doi.org/10.1137/140967994.