# *An Algorithm For Finding the Optimal Embedding of a Symmetric Matrix into the Set of Diagonal Matrices*

Borsdorf, Rüdiger

2012

ISSN 1749-9097

# AN ALGORITHM FOR FINDING THE OPTIMAL EMBEDDING OF A SYMMETRIC MATRIX INTO THE SET OF DIAGONAL MATRICES*

RÜDIGER BORSDORF[†]

**Abstract.** We investigate two two-sided optimization problems that have their application in atomic chemistry and whose matrix of unknowns $Y \in \mathbb{R}^{n\times p}$ $(n \geq p)$ lies in the Stiefel manifold. We propose an analytic optimal solution of the first problem, and show that an optimal solution of the second problem can be found by solving a convex quadratic programming problem with box constraints and $p$ unknowns. We prove that the latter problem can be solved by the active-set method in at most $2p$ iterations. Subsequently, we analyze the set of the optimal solutions of both problems, which is of the form of $\mathcal{C} = \{Y \in \mathbb{R}^{n\times p} : Y^TY = I_p, Y^T\varLambda Y = \varDelta\}$ for $\varLambda$ and $\varDelta$ diagonal and we address the problem how an arbitrary smooth function over $\mathcal{C}$ can be minimized. We find that a slight modification of $\mathcal{C}$ is a Riemannian manifold for which geometric objects can be derived that are required to make an optimization over this manifold possible. By using these geometric tools we propose then an augmented Lagrangian-based algorithm that minimizes an arbitrary smooth function over $\mathcal{C}$ and guarantees global convergence to a stationary point. Latter is shown by investigating when the LICQ (Linear Independence Constraint Qualification) is satisfied. The algorithm can be used to select a particular solution out of the set $\mathcal{C}$ by posing a new optimization problem. Finally we compare this algorithm numerically with a similar algorithm that, however, does not apply these geometric tools and that is to our knowledge not guaranteed to converge. Our results show that our algorithm yields a significantly better performance.

**Key words.** matrix embedding, augmented Lagrangian method, active-set method, Stiefel manifold, Grassmannian manifold, optimization over Riemannian manifolds, orthogonality constraints

**AMS subject classifications.** 65F30, 90C30, 53B20

## 1. Introduction.

This work is motivated by two problems in atomic chemistry [18], [20]. The aim is to determine localized atomic orbitals that satisfy certain properties and come from a large precomputed density operator. In the first problem these atomic orbitals should reproduce occupation numbers that are closest to prescribed values whereas in the second problem they should only reproduce the number of electrons. This leads to two mathematical problems that involve minimizing an objective function over the Stiefel manifold, whose solutions are non-unique. We will start by introducing these two problems and we will see that their solutions can be described by elements of the Stiefel manifold that embed a symmetric matrix into the set of diagonal matrices. We further investigate this set of optimal solutions called $\mathcal{C}$ and we will see that a slight modification of this set is a Riemannian manifold. We develop all necessary geometric objects to be able to apply optimization routines like the nonlinear conjugate gradient (CG) method for Riemannian manifolds that have recently been proposed. We propose then to use an augmented Lagrangian-based algorithm to impose those constraints that we had removed to prove that the remaining set is a Riemannian manifold. The resulting algorithm can be used to find one particular solution out of $\mathcal{C}$ by posing a new optimization problem whose solution is the actual point of interest. By analyzing the LICQ we will show that this Lagrangian-based algorithm is guaranteed to converge to a stationary point.

We proceed as follows. In the next section we introduce the first problem and

[*] Version of August 23, 2012.

[†] School of Mathematics, University of Manchester, Manchester, M13 9PL, UK (boru@hrz.tu-chemnitz.de. This author was supported by EPSRC, the School of Mathematics at the University of Manchester, and the Numerical Algorithms Group Ltd.

show how a solution can analytically be obtained. As the minimum value can be derived [9] by exploiting the structure of the stationary points we only need to find a point on the Stiefel manifold that attains this value. In section 3 we introduce the second problem and show by means of the derivations arising in the first problem that the second problem is equivalent to a convex quadratic programming problem. We use the active-set method to solve this problem and show that it will converge in at most $2p$ iterations to an optimal solution despite the lack of strict convexity of the objective function. Since, in general, both problems are non-unique we discuss in section 4 how one can optimize over the set of optimal solutions. We modify the constraint set by removing $p$ constraints and optimize over the remaining set that we show is a Riemannian manifold. To make an optimization over this manifold possible we develop all geometric objects needed. This yields a new algorithm that we introduce in the following section 5 and whose convergence we show in section 6. Finally we test the performance of this algorithm in section 7 numerically.

Let $N \in \mathbb{R}^{n\times n}$ be a positive definite symmetric matrix, describing a block of the density operator centered at a certain atom. Let further $p \le n$ and $T = \operatorname{diag}(t_1, t_2, \dots, t_p) \in \mathbb{R}^{p\times p}$ be a diagonal matrix whose diagonal elements are the occupation numbers of $p$ atomic orbitals. Therefore the trace of $T$ is the number of electrons contributed by that atom. We define

$$\langle A, B\rangle := \operatorname{trace}(B^T A) \tag{1.1}$$

as our inner product in $\mathbb{R}^{n\times p}$ and the corresponding norm is the Frobenius norm $||A||_F^2 := \langle A, A\rangle$. Further let $\mathcal{D}(s)$ be the set of diagonal matrices in $\mathbb{R}^{s\times s}$ with the diagonal elements in increasing order.

## 2. Problem 1.

To find a minimal set of localized orbitals that have occupation numbers closest to the prescribed diagonal elements of $T$ we seek a solution $Y_*$ of

$$\min_{Y^TY=I_p,\, Y\in\mathbb{R}^{n\times p}} ||Y^TNY - T||_F^2, \tag{2.1}$$

where $I_p$ denotes the identity in $\mathbb{R}^{p\times p}$. The columns of $Y_*$ are then the atomic orbitals expanded in the auxiliary basis. Without loss of generality let the diagonal elements of $T$ be in increasing order, i.e. $t_1 \le t_2 \le \cdots \le t_p$.

### 2.1. The Optimal Function Value.

Now the aim is to find a solution of (2.1). As the constraint set of (2.1) is the Stiefel manifold $\mathsf{St}(n,p) := \{Y \in \mathbb{R}^{n\times p} : Y^TY = I_p\}$ we can formulate the optimality conditions for (2.1) [9] and find that the eigenvalues $\delta_1^*, \dots, \delta_p^*$ of $Y_*^TNY_*$ at the minima are given by

$$\delta_i^* = \begin{cases} t_i & \text{if } t_i \in (\lambda_i, \lambda_{i-p+n}) \\ \lambda_i & \text{if } t_i \le \lambda_i \\ \lambda_{i-p+n} & \text{otherwise} \end{cases} \tag{2.2}$$

for all $i = 1, \dots, p$ where $\lambda_1, \dots, \lambda_n$ with $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$ are the eigenvalues of $N$. Hence the optimal function value $f(Y_*)$ is

$$\sum_{i=1}^{p} \bigl(\max\{0, \lambda_i - t_i, t_i - \lambda_{i-p+n}\}\bigr)^2.$$

Now we need to compute $Y_*$ that realizes these eigenvalues.

**2.2. Construction of Arrowhead Matrix With Prescribed Eigenspectrum.** Let us first consider the special case when $N = \operatorname{diag}(n_1, \ldots, n_n) \in \mathcal{D}(n)$ and $\varDelta = \operatorname{diag}(\delta_1, \ldots, \delta_{n-1}) \in \mathcal{D}(n-1)$ satisfy $n_1 \leq \delta_1 \leq n_2 \leq \cdots \leq \delta_{n-1} \leq n_n$. We say $N$ *interlaces* $\varDelta$. Then by the next theorem we can construct an arrowhead matrix $A$ such that the $(n-1)$th principal minor of $A$ is $\varDelta$ and the eigenvalues of $A$ are the diagonal elements of $N$. Hence, with $A = VNV^T$ the spectral decomposition of $A$ we can set $Y := V^T[I_{n-1}\ 0]^T$ and obtain a matrix $Y$ with orthonormal columns such that $Y^T NY = \varDelta$.

Theorem 2.1. *[23, Theorem 1] or [16, Theorem 4.3.10]. Let $\lambda_1, \ldots, \lambda_m$ interlace $\theta_1, \ldots, \theta_{m-1}$, i.e.*

$$\lambda_1 \leq \theta_1 \leq \lambda_2 \leq \cdots \leq \lambda_{m-1} \leq \theta_{m-1} \leq \lambda_m$$

*for $m > 1$. Further for $k \in \mathbb{N}$ let the vectors*

$$v_1 = \begin{pmatrix} 1 \\ v_1(2) \end{pmatrix}, v_2 = \begin{pmatrix} v_1(2)+1 \\ v_2(2) \end{pmatrix}, \ldots, v_k = \begin{pmatrix} v_{k-1}(2)+1 \\ m-1 \end{pmatrix} \in \mathbb{N}^2$$

*be chosen such that for all $i = 1, \ldots, k$*

$$\lambda_{v_i(1)} \leq \theta_{v_i(1)} = \lambda_{v_i(1)+1} = \cdots = \lambda_{v_i(2)} = \theta_{v_i(2)} \leq \lambda_{v_i(2)+1}$$

*with $v_i(2) - v_i(1)$ maximal. Then there exist $c_1, \ldots, c_{m-1} \in \mathbb{R}$ such that the symmetric arrowhead matrix*

$$A = \begin{pmatrix} \theta_1 & & & c_1 \\ & \ddots & & \vdots \\ & & \theta_{m-1} & c_{m-1} \\ c_1 & \ldots & c_{m-1} & \sum_{i=1}^{m-1}(\lambda_i - \theta_i) + \lambda_m \end{pmatrix}$$

*has the eigenvalues $\lambda_1, \ldots, \lambda_m$. For the values of $c_1, \ldots, c_{m-1}$ it holds that for $i = 1, \ldots, k$*

$$c_{v_i(1)}^2 + \ldots + c_{v_i(2)}^2 = -\frac{\varPi_{j<=v_i(1)}(\theta_{v_i(1)} - \lambda_j)\varPi_{j>v_i(2)}(\theta_{v_i(1)} - \lambda_j)}{\varPi_{j<v_i(1)}(\theta_{v_i(1)} - \theta_j)\varPi_{j>v_i(2)}(\theta_{v_i(1)} - \theta_j)} \geq 0.$$

Note that the condition of $v_i(2) - v_i(1)$ to be maximal ensures that the vectors $v_1, \ldots, v_k$ are uniquely determined. Also if $v_i(1) = v_i(2)$ then $c_{v_i(1)}^2$ is uniquely determined. Further if $\lambda_{v_i(1)} = \theta_{v_i(1)}$ or $\theta_{v_i(2)} = \lambda_{v_i(2)+1}$ then all the $c_{v_i(1)}, \ldots, c_{v_i(2)}$ are zero.

**2.3. Construction of a Solution of Problem 1.** The idea is now to generalize the procedure in section 2.2 to be able to compute for general $N$ and $\varDelta$ a $Y$ such that $Y^T NY$ has the eigenvalues (2.2). After diagonalizing $N$ with $N = P\varLambda P^T$ we will see that we apply two permutation matrices $U$ and $Q$ to $\varLambda$ and $T$, respectively such that we obtain smaller diagonal matrices $\varLambda_i$ and $T_j$ for $i = 1, \ldots, q+1$ and $j = 1, \ldots, q$, satisfying

(i). $\varLambda_i$ interlaces $T_i$ for $i = 1, \ldots, q$,

(ii). $U^T \varLambda U = \operatorname{diag}(\varLambda_1, \ldots, \varLambda_q, \varLambda_{q+1})$ and

(iii). $Q^T TQ = \operatorname{diag}(T_1, \ldots, T_q)$, respectively.

This allows us to apply the Theorem 2.1 to the smaller diagonal matrices and thus to find a solution of (2.1). Let us now construct these permutation matrices. Let $Q$

and $U$ be two permutation matrices that are chosen such that when applied to $\Lambda$ and $T$, respectively, the eigenvalues of $\Lambda$ and diagonal elements $T$ are reordered in the following way. Let $q := \min\{p, n-p\}$. By using the floor operator $\lfloor\ \rfloor : \mathbb{R} \mapsto \mathbb{Z}$ with

$$\lfloor x \rfloor = \max_{y \in \mathbb{Z}, y \leq x} y$$

we define additionally $q$ numbers $s_i := \left\lfloor \frac{n-i}{n-p} \right\rfloor + 1$ for $i = 1, \ldots, q$. Then let

$$\begin{aligned} \Lambda_i &:= \operatorname{diag}(\lambda_i, \lambda_{i+n-p}, \ldots, \lambda_{i+(s_i-1)(n-p)}), \\ T_i &:= \operatorname{diag}(t_i, t_{i+n-p}, \ldots, t_{i+(s_i-2)(n-p)}) \end{aligned} \tag{2.3}$$

and let $\Lambda_{q+1}$ be the diagonal matrix having all the eigenvalues of $\Lambda$ on its diagonal that do not occur in (2.3). This definition of $Q$ and $U$ is well defined and as the matrices $\Lambda_i$ interlace $T_i$ for $i = 1, \ldots, q$ these permutation matrices satisfy the conditions (i)-(iii). Then by virtue of Theorem 2.1 we obtain a $Y_i$ for all $i = 1, \ldots, q$ such that $Y_i^T \Lambda_i Y_i = T_i$. Hence the matrix

$$\widehat{Y} := \operatorname{diag}(Y_1, Y_2, \ldots, Y_q) \tag{2.4}$$

solves $\widehat{Y}^T U^T \Lambda U \widehat{Y} = Q^T T Q$ and consequently a solution of $Y^T N Y = T$ is obtained by setting $Y := P U \widehat{Y} Q^T$.

## 3. Problem 2.

Now we are interested in determining a minimal set of localized orbitals that reproduce the number of electrons. Let $c \in \mathbb{N}$ be defined as this prescribed number. Then we seek a solution of

$$\min_{Y^T Y = I_p,\, Y \in \mathbb{R}^{n \times p}} (\operatorname{trace}(Y^T N Y) - c)^2. \tag{3.1}$$

### 3.1. Reformulation Into a Convex Quadratic Programming Problem.

To solve this problem we apply [13, Theorem 1], giving us a relation between the eigenvalues $\theta = (\theta_1, \ldots, \theta_p)^T$ of $Y^T N Y$ with $\theta_1 \leq \cdots \leq \theta_p$ and the eigenvalues of $N$. That is, there exists a $Y$ with orthonormal columns and $Y^T N Y = \operatorname{diag}(\theta)$ if and only if $\theta_i \in [\lambda_i, \lambda_{n-i+p}]$. We use this theorem to reformulate (3.1) into a convex quadratic programming problem with inequality constraints.

As $\operatorname{trace}(Y^T N Y) = \sum_{i=1}^p \theta_i$ we have with $c_p := c/p$ and $\mu = (\mu_1, \ldots, \mu_p)^T$ that

$$\begin{array}{rl} \min_Y & (\operatorname{trace}(Y^T \Lambda Y) - c)^2 \\ \text{s.t.} & Y^T Y = I_p \end{array} \iff \begin{array}{rl} \min_\mu & \left(\sum_{i=1}^p \mu_i\right)^2 \\ \text{s.t.} & \mu_i \in [\lambda_i - c_p, \lambda_{i+n-p} - c_p]. \end{array}$$

If we rewrite the box constraints as inequality constraints we obtain

$$\begin{array}{rrcl} \min_{\mu \in \mathbb{R}^p} & (\mu^T e)^2 & & \\ \text{s.t.} & e_i^T \mu & \geq & \lambda_i - c_p, \\ & -e_i^T \mu & \geq & -\lambda_{i+n-p} + c_p \end{array} \tag{3.2}$$

where $e \in \mathbb{R}^p$ is the vector of ones. As the feasible set of this problem is closed, convex and not empty and the objection function is convex, but not strictly convex, a solution of (3.2) exists but may not be unique. If $\mu_*$ is an optimal solution of (3.2) then to solve (3.1) it remains to determine a $Y$ such that $Y^T N Y = \operatorname{diag}(\mu + c_p e)$, which is obtained by the solution of (2.1) with $T = \operatorname{diag}(\mu + c_p e)$.

**3.2. Solving the Convex Quadratic Programming Problem.** To solve (3.2) we consider to apply the active-set method described in [22, Algorithm 16.3] and we will show that this method terminates in at most $2p$ iterations and returns an optimal solution despite the lack of strictly convexity of the objective function. First, we assume that $\lambda_i \neq \lambda_{i+n-p}$ for all $i = 1, \ldots, p$. This is no restriction as all elements of $\mu$ with $\lambda_i = \lambda_{i+n-p}$ are fixed so that the programming problem can be reduced to an equivalent programming problem that satisfies the assumption.

**3.2.1. The Active-Set Method.** The primal active-set method finds solutions of convex quadratic programming problems with linear equality and inequality constraints by iteratively solving a convex quadratic subproblem with only equality constraints. These constraints include all equality constraints and a subset of the active set $\mathcal{A}(x_k)$ at the iterate $x_k$ with the inequality constraints transformed into equality constraints. The set of the constraints considered is usually called working set $\mathcal{W}_k$ at iteration $k$. By solving the subproblem a direction $d_k$ of the overall problem is found along which the objective function is not increasing and then a step length $\alpha_k \in [0, 1]$ is maximally chosen such that the new point $x_{k+1} = x_k + \alpha_k d_k$ is feasible. If there is a blocking constraint this constraint will be added to the new working set $\mathcal{W}_{k+1}$. If the direction $d_k$ is zero and all Lagrange multipliers of the current subproblem corresponding to inequality constraints are nonnegative $x_k$ will be a global solution. If one of these multipliers is negative the corresponding constraints is removed from the working set and the iteration is continued until a global solution is found. If the convex quadratic objective function is strictly convex the active-set method converges to a unique global solution in a finite number of iterations [22, Section 16.5].

**3.2.2. Applying the Active-Set Method.** Now we are ready to apply the active-set method to (3.2). Let $\mu^k$ be the current iterate in the active-set method. Then the subproblem of (3.2) at iteration $k$ reads with $d = \mu - \mu^k$ and $a_i$ the constraint gradients of (3.2) for all $i \in \mathcal{W}_k$

$$\begin{array}{rl} \min_d & d^T ee^T d + 2d^T ee^T \mu^k \\ \text{s.t.} & a_i^T d = 0 \quad \text{ for all } i \in \mathcal{W}_k \end{array} \tag{3.3}$$

where $d$ is the direction that we are looking for. Let $r := |\mathcal{W}_k|$. The next lemma gives us an constructive optimal solution of the subproblem (3.3).

Lemma 3.1. *Let $A_k \in \mathbb{R}^{p\times r}$ be the matrix whose columns are the constraint gradients of the subproblem. From* (3.3) *we can assume that they are all of the form of $e_i$ for $i \in \{1, \ldots, p\}$. Therefore there exists a permutation matrix $P_k \in \mathbb{R}^{p\times p}$ such that $P_k^T A_k = [\ I_r \quad 0\ ]^T$. Then in MATLAB notation*

$$d_k := \begin{cases} -P_k(:, r+1:p)\frac{e^T\mu^k}{p-r}e & \text{for } r < p \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

*is an optimal solution of* (3.3). *If $r < p$ all Lagrange multipliers corresponding to the inequalities in $\mathcal{W}_k$ will be zero.*

*Proof.* The proof follows by applying the Lagrangian method to (3.3). See also [7, Lemma 4.5.3]. □

The statement of the next lemma is needed to show subsequently that the active-set method terminates at most $2p$ iterations for (3.2).

Lemma 3.2. *Let $j$ be a blocking constraint that is added to $\mathcal{W}_k$ in iteration $k$ in the active-set method for* (3.2). *Then this constraint will not be removed from $\mathcal{W}_k$*

*in the algorithm under the assumption that* (3.4) *is used for the direction in every iteration.*

*Proof.* Assume that at iteration $l > k$ the constraint $j$ is removed from $\mathcal{W}_l$. By Lemma 3.1 a constraint is only removed if $r = p$. Therefore by [22, Theorem 16.5] and the second part of the proof we have that

$$a_j^T d_{l+1} > 0. \tag{3.5}$$

Further, as $j$ was a blocking constraint at iteration $k$ it holds that $a_j^T d_k = -a_j^T P_k(:, r+1:p)\frac{e^T \mu^k}{p-r} e \leq 0$. As we minimize $(e^T \mu)^2$, $e^T \mu^k$ and $e^T \mu^{l+1}$ must have the same sign. Thus from

$$a_j^T d_{l+1} = -a_j^T P_{l+1}(:, r+1:p) e^T \mu^{l+1} e \leq 0,$$

which contradicts (3.5). □

We are ready to state our main result of this section.

Theorem 3.3. *If the directions $d_k$ are chosen as in* (3.4) *then the active-set method converges to a global solution of* (3.2) *and terminates in at most* $2p$ *iterations.*

*Proof.* We need to show that after at most $2p$ iterations the active-set method reaches a point $\mu^*$ where the convex quadratic subproblem (3.2) has the solution $d = 0$ and all Lagrange multipliers $\lambda_i$ with $i \in \mathcal{W}_k$ are nonnegative [22, Section 16.5]. Let us first assume $r < p$.

At iteration $k$, either $d_k$ is nonzero or $d_k$ is zero, which implies according to Lemma 3.1 that $\mu^k$ is an optimal solution of (3.2). In the former case a constraint will be added to $\mathcal{W}_k$ or the new direction $d_{k+1} = 0$. Latter implies, unless $r = p$, that an optimal solution is found. Therefore, for $r < p$ in every iteration one constraint is added to the working set and no constraint is removed until $r = p$ or an optimal solution has been found. Let $m$ be the number of iterations until $r = p$. Note that $m \leq p$.

Let now $r = p$. Then the solution of (3.2) is $d_k = 0$. Assume that there exists a Lagrange multiplier with $\lambda_j < 0$ for $j \in \mathcal{W}_k$ that is removed from the working set $\mathcal{W}_k$. Since in the next iteration $r = p - 1 < p$ we obtain for $d_{k+1}$ either zero and an optimal solution is found or according to [22, Theorem 16.5] a direction for $q(\cdot)$ along which the inequality $j$ is satisfied. If a blocking constraint exists then this constraint will be added to the new working set and $r = p$, otherwise an optimal solution is found. By Lemma 3.2 this procedure can happen at most $p - m$ times, requiring at most $2(p - m)$ additional iterations. Thus, in total we have at most $2(p - m) + m$ iterations and as $m$ can be zero, the algorithm terminates after at most $2p$ iterations. Note that the factor 2 results from the iteration where $r = p$ and one constraint is removed from the working set, and the subsequent iteration where a reduction of the objective function is achieved. □

## 4. Optimization over a Modified Set of Solutions.

### 4.1. Strategy to Select one Optimal Solution.

In the previous sections we solved the problems that were introduced at the beginning of section 2 and 3. However, the solutions obtained might not be unique. For (2.1) we have shown that the set of optimal solutions is equivalent to

$$\mathcal{C} := \left\{ Y \in \mathsf{St}(n,p) : Y^T \Lambda Y = \Delta \right\} \tag{4.1}$$

with $\Delta = \operatorname{diag}(\delta_1, \ldots, \delta_p)$ as defined in (2.2) and $\Lambda$ the diagonal matrix with the eigenvalues of $N$ on its diagonal. Moreover, we have seen in section 3.1 that this set also plays an important role in solving (3.1). To select a particular solution out of the set in (4.1) the idea is to pose a new optimization problem. We therefore establish a new framework in this section that allows the optimization of an arbitrary smooth function $f$ over the set (4.1). Depending on the application, this function should then be chosen such that the minimum value of $f$ is attained at the points of interest in (4.1). Our approach assumes that the diagonal elements of $\Delta$ are distinct and in increasing order.

We will first consider a set that imposes $p$ fewer constraints on $Y \in \mathsf{St}(n,p)$ than $\mathcal{C}$ but can easily been proven to be a Riemannian manifold. We will then show that all geometric objects can be developed to make an optimization over this manifold possible by using optimizing algorithms that are applicable over Riemannian manifolds [2]. Therefore to optimize over $\mathcal{C}$ it remains to impose the $p$ constraints that we have disregarded. We tackle this problem by applying the augmented Lagrangian method [6, Section 4.2] in the next section 5.

The motivation for this approach is that optimization over smooth manifolds has recently become more popular and it has been shown that for certain applications the algorithms that optimize over these manifolds can outperform the conventional state-of-the-art algorithms [1], [2], [3], [28]. Examples for the successful applications of optimization algorithms over smooth manifolds can be found in many areas of science. Image processing is one example where segmentation and registration algorithms often rely on these optimization algorithms [10], [25]. Blind source separation is another application where efficient algorithms were proposed [24]. See also [4]. Another example is the low rank nearness problems as they can be transformed into optimization problem over manifolds [21]. An extension to tensors was proposed in [17] where their algorithm achieves superlinear convergence. Other candidates are the algorithm described in [27] for multilevel optimization of rank constraint matrix problems applied to find the low rank solution of Lyapunov equations and the nearest weighted low rank correlation matrix algorithm proposed in [14]. The latter algorithm can also compute the nearest correlation matrix for an element-wise weighting of the matrix that is not supported by algorithms that rely on the projection onto the set of positive semidefinite matrices [15], [8]. A popular application is also to compute eigenvalues of a given matrix by minimizing the Rayleigh quotient over the sphere in $\mathbb{R}^n$, which is a smooth manifold [2, Section 2].

For definitions and an introduction to smooth manifold and related geometric objects we refer to [19], [7, Chapter 3] and in particular for the optimization over matrix manifolds to [12], [2]. Our notation for the geometric objects is mainly taken from [2], [7].

**4.2. Formulation as a Riemannian Manifold.** Let $D_Y := Y^T \Lambda Y$ and $d_i := (D_Y)_{ii}$ for all $i$. Note that $d_i$ depends on $Y$ but for simplicity we leave out the subscript $Y$. Further let us now define the new constraint set as

$$\mathcal{B}(n,p) = \{ Y \in \mathsf{St}(n,p) : \operatorname{offdiag}(D_Y) = 0 \text{ and } d_1 < \cdots < d_p \ \}$$

where offdiag : $\mathbb{R}^{p \times p} \mapsto \mathbb{R}^{p(p-1)}$ is the operator that stacks the off-diagonals into a long vector starting from the most upper right. Note that $\mathcal{B}(n,p)$ does not impose the constraints that the diagonal elements of $D_Y$ coincide with the diagonal elements of $\Delta$. The idea is to impose these constraints separately in our optimization routine.

Let further $\mathcal{S}^p$ be the set of symmetric and $\mathcal{K}(p)$ the set of skew-symmetric matrices in $\mathbb{R}^{p \times p}$. Now we are ready to show that $\mathcal{B}(n,p)$ is an embedded submanifold of $\mathbb{R}^{n \times p}$.

Lemma 4.1. *$\mathcal{B}(n,p)$ is an embedded submanifold of $\mathbb{R}^{n \times p}$ with dimension $np - p^2$.*

*Proof.* Let $Y \in \mathcal{B}(n,p)$. Then there exists an open neighbourhood $U_Y$ of $Y$ in $\mathbb{R}^{n \times p}$ such that the diagonal elements of $D_X := X^T \Lambda X$ are distinct for all $X \in U_Y$. Let $U = \bigcup_{Y \in \mathcal{B}(n,p)} U_Y$. As $U_Y$ is an open subset of $\mathbb{R}^{n \times p}$ $U$ is clearly an open submanifold of $\mathbb{R}^{n \times p}$ of dimension $np$.

Consider $F : U \mapsto \mathcal{S}^p \times \mathcal{S}^p_0$ with

$$F(X) = \begin{bmatrix} X^T X - I_p \\ D_X - \mathrm{diag}\,(D_X) \end{bmatrix}, \tag{4.2}$$

where $\mathcal{S}^p_0 := \{Z \in \mathcal{S}^p : \mathrm{diag}(Z) = 0\}$. Then by construction it holds that $F^{-1}(0) = \mathcal{B}(n,p)$. Now the idea is to apply Theorem [2, Proposition 3.3.3], which says that set $F^{-1}(0)$ is an embedded submanifold of $U$ if 0 is a *regular* point of $F$. See [7, Section 3.3.1] for a definition.

Let $S = \begin{bmatrix} S_1 & S_2 \end{bmatrix}^T \in \mathcal{S}^p \times \mathcal{S}^p_0$ be arbitrary and $\widehat{Z} = \frac{1}{2} X(S_1 + K)$ with $K \in \mathcal{K}(p)$ and

$$K_{ij} = \begin{cases} \frac{(D_X S_1 + S_1 D_X - 2S_2)_{ij}}{(D_X)_{jj} - (D_X)_{ii}} & \text{for } i \neq j \\ 0 & \text{otherwise.} \end{cases}$$

Then from

$$DF(X)[Z] = \begin{bmatrix} X^T Z + Z^T X \\ 2\mathrm{sym}(X^T \Lambda Z) - 2\,\mathrm{diag}\left(\mathrm{sym}\left(X^T \Lambda Z\right)\right) \end{bmatrix}$$

we have that $DF(X)[\widehat{Z}] = \begin{bmatrix} S_1 & S_2 \end{bmatrix}^T$. As the matrix $S$ was chosen arbitrarily $F$ is of full rank at all $X \in \mathcal{B}(n,p)$, which implies that 0 is a regular value of $F$. Hence, by Theorem [2, Proposition 3.3.3] $\mathcal{B}(n,p)$ is an embedded submanifold of $U$ with dimension $\dim(U) - \dim(\mathcal{S}^p_0) = np - p^2$. As $U$ covers $\mathcal{B}(n,p)$ by [2, Proposition 3.3.2] $\mathcal{B}(n,p)$ is an embedded submanifold of $\mathbb{R}^{n \times p}$. □

Note that the Riemannian manifold $\mathcal{B}(n,p)$ is bounded as each column of $Y \in \mathcal{B}(n,p)$ has 2-norm one, implying that $||Y||_F = \sqrt{p}$, but it is not closed as demonstrated by the following example. Let $\{\varepsilon_k\}_{k \geq 0}$ be a sequence with $\varepsilon_k \searrow 0$ as $k \to \infty$. Let $\Lambda = \mathrm{diag}(1,1,2)$ and $\{Y_k\}_{k \geq 0}$ be a sequence with

$$Y_k = \begin{bmatrix} 1 & 0 \\ 0 & (1 - \varepsilon_k)/s_k \\ 0 & \varepsilon_k / s_k \end{bmatrix} \text{ where } s_k = \sqrt{\varepsilon_k^2 + (1 - \varepsilon_k)^2}.$$

Then $Y_k$ is in $\mathcal{B}(3,2)$ for all $k$ as ${Y_k}^T \Lambda Y_k = \mathrm{diag}\left(1, \frac{2\varepsilon_k^2 + (1-\varepsilon_k)^2}{\varepsilon_k^2 + (1-\varepsilon_k)^2}\right)$. However $Y_* = \lim_{k \to \infty} Y_k \notin \mathcal{B}(3,2)$ as $Y_*^T \Lambda Y_* = \mathrm{diag}(1,1)$.

**4.3. Geometric Objects.** When optimizing a smooth function $f : \mathbb{R}^n \mapsto \mathbb{R}$ over $\mathbb{R}^n$ the usual procedure to find stationary points, i.e. points $x \in \mathbb{R}^n$ with $\nabla f(x) = 0$ and $\nabla f$ the derivative of $f$, is to generate a sequence with

$$x_{k+1} = x_k + \alpha_k d_k, \tag{4.3}$$

starting from a given point $x_0$. If $d_k$ is a gradient-related descent direction and $\alpha_k$ is suitably chosen one can show that the sequence converges to a stationary point.

However, if $x_k$ is on a manifold $\mathcal{M}$ then a direct generalization of this procedure is not possible as for instance $x_{k+1}$ might not be in the manifold. Let us briefly explain how this can be generalized to manifolds that are embedded in $\mathbb{R}^{n\times p}$ in a simplified manner. We refer to [19] and [2] for more details.

To generalize the conventional optimization over $\mathbb{R}^{n\times p}$ to matrix manifolds we first need to derive the tangent space $T_{x_k}\mathcal{M}$ of $\mathcal{M}$ at $x_k$. This space has the same dimension as the manifold and can be equipped with an inner product $\langle\cdot,\cdot\rangle$, allowing the generalization of the gradient of $f$ at $x_k$ that is the vector $\operatorname{grad} f \in T_{x_k}\mathcal{M}$ that satisfies

$$\langle \operatorname{grad} f, x\rangle = \nabla f^T x \quad \text{for all } x \in T_{x_k}\mathcal{M}. \tag{4.4}$$

A descent direction is then a vector $d_k \in T_{x_k}\mathcal{M}$ that satisfies $\langle d_k, \operatorname{grad} f\rangle < 0$ and a stationary point is a point $x \in \mathcal{M}$ with $\operatorname{grad} f(x) = 0$. We generalize then the iteration in (4.3) by using a smooth mapping $R_{x_k} : \mathcal{T}_{x_k}\mathcal{M} \mapsto \mathcal{M}$ called retraction that satisfies $R_{x_k}(0) = x_k$ and that $\gamma'(0) = d$ for the curve $\gamma(t) := R_{x_k}(td)$ for all $d \in T_{x_k}\mathcal{M}$. The generalization for (4.3) is then to compute $x_{k+1}$ as $x_{k+1} = R_{x_k}(\alpha_k d_k) \in \mathcal{M}$. This iterations allows to develop globally convergent algorithms over Riemannian manifolds [2, Chapter 4]. Note that a retraction is a first order approximation of the geodesic. It is often also required to compare vectors of tangent spaces at different points for instance in the nonlinear CG method [26]. Therefore one needs to transport the vector from one tangent space into another. For manifolds this is realized by concept of vector transports; see [2, Definition 8.1.1]. To transport a vector $\xi_x \in T_x\mathcal{M}$ into $T_{R_x(\eta_x)}\mathcal{M}$ for $\eta_x \in T_x\mathcal{M}$ we will use later the vector transport [2, Section 8.1.3]

$$\mathcal{T}_{\xi_x}\eta_x = P_{R_x(\eta_x)}\xi_x \tag{4.5}$$

where $P_{R_x(\eta_x)}\xi_x$ is the orthogonal projection of $\xi_x$ onto the tangent space $T_{R_x(\eta_x)}\mathcal{M}$.

**4.3.1. Tangent and Normal Space.** Let us now introduce the tangent space of $\mathcal{B}(n,p)$. We start with the definition of an operator $A : \mathbb{R}^{n\times p} \mapsto \mathcal{K}(p)$ at $Y \in \mathcal{B}(n,p)$ with

$$A_{ij}(Z) = \frac{2\operatorname{sym}(Y^T \Lambda Z)_{ij}}{d_j - d_i}$$

for $i \neq j$ and $A_{ii} = 0$ for all $i = 1,\ldots,p$.

LEMMA 4.2. *[7, Lemma 4.6.3] The tangent space $T_Y\mathcal{B}(n,p)$ of $\mathcal{B}(n,p)$ at $Y \in \mathcal{B}(n,p)$ is*

$$T_Y\mathcal{B}(n,p) = \left\{ Z = YA(Y_\perp B) + Y_\perp B : B \in \mathbb{R}^{(n-p)\times p} \text{ free} \right\}. \tag{4.6}$$

The symbol $Y_\perp$ in Lemma 4.2 denotes a matrix in $\mathbb{R}^{n\times(n-p)}$ that has orthonormal columns and are complementary to the columns of $Y$. Let us endow all tangent spaces $T_Y\mathcal{B}(n,p)$ with the inner product defined in (1.1). This allows us also to define the normal space $N_Y\mathcal{B}(n,p)$ that is of dimension $p^2$.

LEMMA 4.3. *[7, Lemma 4.6.4] The normal space of $\mathcal{B}(n,p)$ at $Y$ is given by*

$$N_Y\mathcal{B}(n,p) = \Big\{ Z \in \mathbb{R}^{n\times p} : Z = \Lambda Y \operatorname{sym}(C) - Y\operatorname{sym}(CD_Y) + YT) \\ \text{for } C, T \in \mathbb{R}^{p\times p} \text{ with } C_{ii} = 0 \text{ and } T \text{ diagonal} \Big\}.$$

**4.3.2. Projection onto Tangent and Normal Space.** Not only for computing the vector transport in (4.5) but also for determining the gradient in (4.4) the projection onto the tangent space $T_Y\mathcal{B}(n,p)$ is needed. As $T_Y\mathcal{B}(n,p)$ is of dimension $p(n-p)$ the projection of an element $Z \in \mathbb{R}^{n\times p}$ onto this space generally requires to solve a linear system of dimension $p(n-p)$. Assuming $p \ll n$ it is significantly less expensive to compute the projection onto $N_Y\mathcal{B}(n,p)$ at $Y$ instead and subtract it from $Z$ as this involves solving only a linear system of dimension $p^2$. Therefore we devote yourselves to this projection. It will turn out that solving a sparse linear system of dimension $q := p(p-1)/2$ will be sufficient.

Let $Q : \{X \in \mathbb{R}^{p\times p} : \operatorname{diag}(X) = 0\} \mapsto \mathbb{R}^{n\times p}$ be an operator with

$$Q(C) = \Lambda Y(C^T + C) - Y(CD_Y + D_Y C^T). \tag{4.7}$$

Then to find the projection of $Z$ onto $N_Y\mathcal{B}(n,p)$ we need to determine the element $Z_n \in N_Y\mathcal{B}(n,p)$ that satisfies

$$\langle Z - Z_n, Q(e_i e_j^T)\rangle = 0 \quad \text{for all } i \neq j \text{ and } \langle Z - Z_n, Y e_i e_i^T\rangle = 0 \quad \text{for all } i = 1,\dots,p.$$

Let $\mathcal{H} \in \mathbb{R}^{p\times p\times p\times p}$ be a tensor and $B \in \mathbb{R}^{p\times p}$ with

$$\mathcal{H}_{i,j,k,l} = \begin{cases} \langle Q(e_k e_l), Q(e_i e_j^T)\rangle & \text{for } i \neq j, k \neq l \\ \langle Q(e_k e_l^T), Y e_i e_j^T\rangle & \text{for } i = j, k \neq l \\ \langle Y e_k e_l^T, Q(e_i e_j^T)\rangle & \text{for } i \neq j, k = l \\ \langle Y e_k e_l^T, Y e_i e_j^T\rangle & \text{for } i = j, k = l \end{cases}, \quad b_{ij} = \begin{cases} \langle Z, Q(e_i e_j^T)\rangle & \text{for } i \neq j \\ \langle Z, Y e_j e_i^T\rangle & \text{for } i = j. \end{cases} \tag{4.8}$$

Then with $H \in \mathbb{R}^{p^2\times p^2}$ being the unfolding of the tensor $\mathcal{H}$ in mode 1 and 2 along the rows and mode 3 and 4 along the columns and $b \in \mathbb{R}^{p^2}$ the mode 1 unfolding of $B$ the linear system that needs to be solved to compute $Z_n$ is $Hz = b$. The vector $z \in \mathbb{R}^{p^2}$ is related to $Z_n$ as follows. Let $C \in \mathbb{R}^{p\times p}, T \in \mathbb{R}^{p\times p}$ be defined as $C := \sum_{i\neq j} z_{((j-1)p+i)} e_i e_j^T$ and $T := \sum_{i=1}^{p} z_{((i-1)p+i)} e_i e_i^T$ then $Z_n = Q\left(C\right) + YT$. Note that only the right-hand side $b$ depends on $Z$. Hence, a multiple projecting onto the same normal space is not of much higher cost than a single projection. Then by noticing that for $k \neq l$

$$\begin{aligned} \mathcal{H}(i,j,k,l) - \mathcal{H}(i,j,l,k) &= \operatorname{trace}\left((e_k e_l^T + e_l e_k^T)(d_l - d_k)(d_j - d_i) e_i e_j^T\right) \\ &= \begin{cases} (d_l - d_k)^2 & \text{for } j = l, i = k \\ -(d_l - d_k)^2 & \text{for } j = k, i = l \\ 0 & \text{otherwise,} \end{cases} \end{aligned}$$

and $k = l$

$$\mathcal{H}(i,j,k,l) = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases}$$

we can transform $Hz = b$ into an equivalent system $\widetilde{H}\widetilde{z} = \widetilde{b}$ with

$$\widetilde{z} = \begin{bmatrix} \widetilde{z}_1 \\ \widetilde{z}_2 \end{bmatrix}, \ \widetilde{b} = \begin{bmatrix} \widetilde{b}_1 \\ \widetilde{b}_2 \end{bmatrix} \in \mathbb{R}^{p^2}, \text{ and } \widetilde{H} = P_c L^{-1} H L P_r = \begin{bmatrix} S & \widetilde{H}_1 \\ 0 & \widetilde{H}_2 \end{bmatrix} \in \mathbb{R}^{p^2\times p^2}, \tag{4.9}$$

where $P_c, P_r$ are permutation matrices, $S \in \mathbb{R}^{(q+p)\times(q+p)}$ diagonal and $\widetilde{H}_1 \in \mathbb{R}^{(q+p)\times q}$, $\widetilde{H}_2 \in \mathbb{R}^{q\times q}$. The matrix $L \in \mathbb{R}^{p^2\times p^2}$ is an invertible lower triangular matrix that corresponds to subtracting the $((k-1)p+l)$st column of $H$ from the $((l-1)p+k)$st column for all $k > l$. Similarly, $L^{-1}$ corresponds to adding the $((i-1)p+j)$st row of $H$ to the $((j-1)p+i))$st row for $j > i$. These rows and columns operations yield together with the permutation matrices $P_c, P_r$ the diagonal matrix $S$ in the upper left corner in (4.9). Hence, the major cost is to solve $\widetilde{H}_2\widetilde{z}_2 = \widetilde{b}_2$, which is of order $q$. Fortunately, $\widetilde{H}_2$ has additional structure shown by the next lemma that we might be able to exploit when solving the linear system.

LEMMA 4.4. *The matrix $\widetilde{H}_2$ is symmetric and sparse for $p$ large whereas the ratio $R$ of the number of zeros to the total number of elements in $\widetilde{H}_2$ is*

$$R \geq 1 - \frac{4}{p-1} + \frac{6}{p(p-1)}, \quad p \neq 1.$$

*Proof.* The proof can be found in [7, Lemma 4.6.6]. □

**4.3.3. A Retraction.** It remains to define a retraction on $\mathcal{B}(n,p)$. Let $\widehat{R}_Y(H)$ be a retraction on the Stiefel manifold $\mathsf{St}(n,p)$ at $Y \in \mathcal{B}(n,p)$ with $H \in T_Y\mathcal{B}(n,p) \subset T_Y\mathsf{St}(n,p)$. Let $R_Y : T_Y\mathcal{B}(n,p) \mapsto \mathcal{B}(n,p)$ be a map with

$$R_Y(H) = \widehat{R}_Y(H)P, \tag{4.10}$$

where $P\Theta P^T$ is the spectral decomposition of $F(H) := \widehat{R}_Y(H)^T \Lambda \widehat{R}_Y(H)$ with the diagonal elements of $\Theta$ in increasing order. Then by the next lemma $R_Y(H)$ is a retraction on $\mathcal{B}(n,p)$ at $Y$.

LEMMA 4.5. *The map $R_Y(H)$ defined in* (4.10) *is a retraction on $\mathcal{B}(n,p)$.*

*Proof.* We need to check the conditions for $R_Y(H)$ to be a retraction. We mentioned these conditions at the beginning of section 4.3. For a definition of a retraction see [2, Definition 4.1.1].

For $H$ in the neighbourhood of $0_Y \in T_Y\mathcal{B}(n,p)$ $R_Y(H)$ is clearly smooth as the diagonal elements of $F(0_Y)$ are distinct. Furthermore as $F(0_Y)$ is diagonal we have that $P = I_p$ for $H = 0_Y$ and

$$R_Y(0_Y) = Y.$$

Let us now consider the curve $R_Y(tH) = \widehat{R}(tH)P(t)$, which exists for all $t$ sufficiently small [11, Section 2.2], where $P(t)$ is the orthogonal matrix that diagonalizes $F(tH)$. Then

$$\left.\frac{d}{dt}R_Y(tH)\right|_{t=0} = HI_p + Y\left.\frac{d}{dt}P(t)\right|_{t=0}. \tag{4.11}$$

From [11, Section 2.2] we obtain that $\left.\frac{d}{dt}P(t)\right|_{t=0} = P(0)T$ with $T$ skew-symmetric and

$$T_{ij} = \frac{\left(P(0)^T \frac{d}{dt}\left(F(tH)\right)\Big|_{t=0} P(0)\right)_{ij}}{d_j - d_i} \quad \text{for } i \neq j, i = 1, \ldots, p,$$

where $d_i = (Y^T\Lambda Y)_{ii}$. As $\operatorname{offdiag}\left(\frac{d}{dt}\left(F(tH)\right)\Big|_{t=0}\right) = \operatorname{offdiag}\left(H^T\Lambda Y + Y^T\Lambda H\right) = 0$ and $T$ skew-symmetric we have that $T = 0$. This implies that the left-hand side of (4.11) is $H$. Therefore all conditions for $R_Y(H)$ to be a retraction are satisfied. □

**4.4. Connection to Grassmannian Manifold.** Let us now briefly investigate the connection of $\mathcal{B}(n,p)$ with the Grassmannian manifold $\mathsf{Gr}(n,p)$, which is a quotient manifold and can be described as the set of all $p$-dimensional subspaces of $\mathbb{R}^n$. It can be defined as the collection of all equivalent classes

$$[Y] := \{YQ : Q \in \mathsf{O}(p)\}$$

for $Y \in \mathsf{St}(n,p)$. See [7, Section 3.8.2], [2], [12]. This manifold has the same dimension $p^2$ as $\mathcal{B}(n,p)$ and its horizontal space is $H_Y = \{Y_\perp B : B \in \mathbb{R}^{(n-p)\times p}\}$. It is easy to see that each $Y \in \mathcal{B}(n,p)$ corresponds to exactly one element in $\mathsf{Gr}(n,p)$ and is a representative of the equivalence class. Similarly, there exists a bijective mapping $h_Y(Z) : T_Y\mathcal{B}(n,p) \mapsto H_Y$ with $h_Y(Z) = (I - YY^T)Z$ and $h_Y^{-1}(Z) = Y[A_{ij}(Z)]_{ij} + Z$ for all $Y \in \mathcal{B}(n,p)$. However, not for all elements in $\mathsf{Gr}(n,p)$ exist a corresponding element in $\mathcal{B}(n,p)$. Let $Y$ be a representative of an element in $\mathsf{Gr}(n,p)$ such that $Y^T \Lambda Y$ has an eigenvalue with a multiplicity greater than one. Then all elements in $[Y]$ have this property and therefore there exists no element in $\mathcal{B}(n,p)$ that represents $[Y]$ and hence, there is also no corresponding element to $Y$ in $\mathcal{B}(n,p)$. Thus, $\mathcal{B}(n,p)$ is isomorphic to an open submanifold of $\mathsf{Gr}(n,p)$ that has the same dimension.

**5. The Algorithm.** Now we have developed all necessary tools to apply first-order optimization algorithms over manifolds like the nonlinear CG method [7, Algorithm 3.9.1] to optimize a smooth function $f$ over $\mathcal{B}(n,p)$.

However, the aim is to optimize $f$ over $\mathcal{C}$ as defined in (4.1). Therefore, in order to incorporate the $p$ constraints of $\mathcal{C}$ that are disregarded in $\mathcal{B}(n,p)$ we are interested in solving

$$\begin{array}{lll} \min_{Y\in\mathcal{B}(n,p)} & f(Y) & \\ \text{s.t.} & c_i(Y) = 0 & \text{for all } i = 1,\dots,p, \end{array} \tag{5.1}$$

where $c_i(Y)$ are the $p$ equality constraints with

$$c_i(Y) = d_i - \delta_i \quad \text{for all } i = 1,\dots,p.$$

and $d_i = y_i^T \Lambda y_i$. The symbol $y_i$ denotes the $i$th columns of $Y$. In the following we will consider to optimize the function $f(Y)$ for $\varepsilon > 0$ over

$$\mathcal{B}^C(\varepsilon) := \{Y \in \mathcal{B}(n,p) : d_{i+1} - d_i \geq \varepsilon \text{ for all } i = 1,\dots,p-1\}$$

subject to the constraints $c_i(Y) = 0$ since $\mathcal{B}^C(\varepsilon)$ is compact and for $\varepsilon$ small this problem is equivalent to (5.1).

We propose to use the augmented Lagrangian method for solving

$$\begin{array}{lll} \min_{Y\in\mathcal{B}^C(\varepsilon)} & f(Y) & \\ \text{s.t.} & c_i(Y) = 0 & \text{for all } i = 1,\dots,p, \end{array} \tag{5.2}$$

which can be stated as follows. Let us first define the augmented Lagrangian function of (5.2), that is

$$G_{\mu,\theta}(Y) = f(Y) - \sum_{i=1}^{p} \theta_i c_i(Y) + \frac{\mu}{2}\sum_{i=1}^{p} c_i(Y)^2 \tag{5.3}$$

where $Y \in \mathcal{B}^C(\varepsilon)$, $\theta \in \mathbb{R}^p$ are the Lagrange multipliers and $\mu > 0$ is the penalty parameter. Let $\theta^0 \in \mathbb{R}^p$ be the initial estimate of the Lagrange multipliers and

$\mu_0 > 0$. Then the augmented Lagrangian method is to determine at the $k$th iteration

$$Y_{k+1} \in \operatorname*{argmin}_{Y \in \mathcal{B}^C(\varepsilon)} G_{\mu_k, \theta^k}(Y) \tag{5.4}$$

and, according to some rules [22, Algorithm 17.4], to update the Lagrange multipliers by

$$\theta_i^{k+1} := \theta_i^k - \mu_k c_i(Y_{k+1}) \quad \text{or} \quad \theta_i^{k+1} := \theta_i^k$$

and to update the penalty parameter by

$$\mu_{k+1} := \mu_k \quad \text{or} \quad \mu_{k+1} > \mu_k.$$

We use the nonlinear CG method [7, Algorithm 3.9.1] to solve (5.4). However, when applying the geometric optimization tools to find a local minimum of (5.4) we need to make sure that the generated iterates in the nonlinear CG method lie in $\mathcal{B}^C(\varepsilon)$. Another problem is that in order to show convergence we require that the LICQ is satisfied at the limit point of a subsequence of $\{Y_k\}_{k \geq 0}$. Let us therefore first define the LICQ on a Riemannian manifold and then investigate when it holds for our problem.

Definition 5.1. *Let $\mathcal{M}$ be a Riemannian manifold and let $c_i(Y) = 0$ for $i = 1, \ldots, p$ be $p$ equality constraints with $Y \in \mathcal{M}$. Then the* LICQ *on $\mathcal{M}$ at $Y \in \mathcal{M}$ holds if* $\operatorname{grad} c_1(Y), \ldots, \operatorname{grad} c_p(Y) \in T_Y\mathcal{M}$ *are linearly independent.* The next lemma characterizes the points of (5.2) at which the LICQ holds.

Lemma 5.2. *Let $c_i(Y) := y_i^T \Lambda y_i - \delta_i$ for $i = 1, \ldots, p$ and $Y \in \mathcal{B}(n, p)$. Then the LICQ holds at $Y$ iff $y_i$ is not an eigenvector of $\Lambda$ for all $i$.*

*Proof.* By the definition of $\operatorname{grad} c_i(Y)$ it holds for all $Z(B) = YA(Y_\perp Y_\perp^T B) + Y_\perp Y_\perp^T B \in T_Y\mathcal{B}(n,p)$ and $B \in \mathbb{R}^{n \times (n-p)}$ with $\nabla c_i(Y) = 2[0_{1\times(i-1)}, \Lambda y_i, 0_{1\times(p-i)}]$ that

$$\begin{aligned}
&\sum_{i=1}^p x_i \langle \operatorname{grad} c_i(Y), Z(B) \rangle \\
=&\sum_{i=1}^p x_i \, \langle \nabla c_i(Y), Z(B) \rangle \\
=&\sum_{i=1}^p x_i \left( \underbrace{\langle \nabla c_i(Y), YA(Y_\perp Y_\perp^T B) \rangle}_{=0} + \langle \nabla c_i(Y), Y_\perp Y_\perp^T B \rangle \right) \\
=&\sum_{i=1}^p x_i \langle 2[0_{1\times(i-1)}, \Lambda y_i, 0_{1\times(p-i)}], Y_\perp Y_\perp^T B \rangle = 2 \sum_{i=1}^p x_i y_i^T \Lambda (I_n - YY^T) b_i \\
=&2 \sum_{i=1}^p x_i y_i^T (\Lambda - d_i I_n) b_i.
\end{aligned} \tag{5.5}$$

Now assume that $y_i$ is not an eigenvector of $\Lambda$ for all $i$. Then $y_i^T(\Lambda - d_i I_n) \neq 0$ and for $\widehat{B} \in \mathbb{R}^{n \times (n-p)}$ with $\widehat{B}_i := (\Lambda - d_i I_n) y_i x_i$ it follows that $\left\langle \sum_{i=1}^p x_i \operatorname{grad} c_i(Y), Z(\widehat{B}) \right\rangle = 0$ iff $x_i = 0$ for all $i$. Hence, the LICQ is satisfied at $Y$. Conversely, assume that the LICQ is satisfied at $Y$ and that $y_j$ is an eigenvector of $\Lambda$ for $j \in \{1, \ldots, p\}$. Then it follows that $y_j^T(\Lambda - d_j I_n) b_j = 0$ for all $B \in \mathbb{R}^{n \times (n-p)}$. Hence, (5.5) implies for all $x_i = 0$ with $i \neq j$ and $x_j \neq 0$ that $\langle \sum_{i=1}^{p} x_i \operatorname{grad} c_i(Y), Z(B) \rangle = 0$ for all $B \in \mathbb{R}^{n\times(n-p)}$, which contradicts our assumption that the LICQ is satisfied at $Y$. $\square$

Note that from the proof of Lemma 5.2 it follows that all $\operatorname{grad} c_i(Y)$ are linearly independent where $y_i$ is not an eigenvector of $\Lambda$. Furthermore, if $y_i$ is an eigenvector of $\Lambda$ then $\operatorname{grad} c_i(Y) = 0$. Hence, it may occur during the iteration that the iterates $Y_k$ do not move away from a point at which the constraints are not satisfied, even for arbitrarily large $\mu$. We have observed this in our numerical tests. Therefore if a column $i$ of $Y$ is an eigenvector of $\Lambda$ our idea is to replace this column by a vector $\hat{y}$ that lies in $\operatorname{span}\{[\bar{Y}\ \Lambda\bar{Y}]_\perp\}$ with $\bar{Y} = [y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_p]$ such that $\hat{y}$ is an eigenvector of $\Lambda$ and $d_{i-1} + \varepsilon \leq \hat{y}^T \Lambda \hat{y} \leq d_{i+1} - \varepsilon$. If, however, $c_i(Y) = 0$ we do not need to modify the $i$th column of $Y$ as the constraint $i$ is satisfied. We will see that these replacements will lead to an algorithm that generates a convergent subsequence whose limit point is a stationary point of (5.2). Moreover, if in the nonlinear CG method a step size cannot be taken due to the new iterate being outside of $\mathcal{B}^C(\varepsilon)$ we will also apply this replacement strategy. To find the vector $\hat{y}$ we will use the algorithm that we introduced in section 2 to solve (2.1). We state this replacement strategy in Algorithm 5.1.

**Algorithm 5.1** Algorithm that implements the replacement strategy

```
Require: $Y, \Lambda, \Delta = (\delta_1, \dots, \delta_p), \varepsilon, n, p$.
1  $\mathcal{I} = \emptyset$
2  repeat
3    ds_changed = false
4    for $i = 1 : p$ do
5      if  $((y_i, d_i)$ is an eigenpair of $\Lambda$ and $|c_i(Y)| > 0)$
           or $(i \neq 1$ and $d_i - d_{i-1} < \varepsilon)$ or $(i \neq p$ and $d_{i+1} - d_i < \varepsilon)$ then
6          $d_i := \begin{cases} \max\{\delta_i, (d_{i-1} + d_i)/2\} & \text{for } d_i > \delta_i \wedge (i \neq 1 \wedge d_{i-1} > \delta_i - \varepsilon) \\ \min\{\delta_i, (d_{i+1} + d_i)/2\} & \text{for } d_i < \delta_i \wedge (i \neq p \wedge d_{i+1} < \delta_i + \varepsilon) \\ \delta_i & \text{otherwise.} \end{cases}$
7          ds_changed = true
8          $\mathcal{I} = \mathcal{I} \cup \{i\}$
9      end if
10   end for
11 until ds_changed = false
12 if $\mathcal{I} \neq \emptyset$ then
13   In MATLAB notation set $\mathcal{I}^C := \{1, \dots, p\} \setminus \mathcal{I}$ and $Z := \left[\Lambda Y\left(:, \mathcal{I}^C\right)\ Y\right]_\perp$ and
     $Q := [Z\ Y(:, \mathcal{I})]$.
14   Solve (2.1) with $N = Q^T \Lambda Q$ and $T = \operatorname{diag}(d_j)$ for $j \in \mathcal{I}$. Set solution to $X_*$.
15   $Y(:, \mathcal{I}) := Q X_*$
16   if $Y \notin \mathcal{B}^C(\varepsilon)$ or $\exists\ y_i$ that is an eigenvector of $\Lambda$ with $|c_i(Y)| > 0)$ then
17     Solve (2.1) with $N = \Lambda$ and $T = \Delta$. Set solution to $Y$.
18   end if
19 end if
20 return $Y$
```

At first on line 6 of Algorithm 5.1 the value of $d_i$ will be changed if it is close to $d_{i-1}$ or $d_{i+1}$, respectively, meaning that $Y \notin \mathcal{B}^C(\varepsilon)$. The value will also be modified if $(y_i, d_i)$ is an eigenvector of $\Lambda$ due to the reasons explained above. This procedure continues until the if-clause on line 5 is false for all $i$. If $\min_{i\in\{2,\ldots,p\}} \delta_i - \delta_{i-1} > \varepsilon$ then it is clear that Algorithm 5.1 will leave the loop on line 2 in a finite number of iterations. If changes have been made to the $d_i$s the new columns $Y(:,\mathcal{I})$ are determined on line 14 by solving (2.1) as in section 2. If $X_*$ attains the function value of zero in (2.1) on line 14, i.e. $\operatorname{diag}(X_*^T N X_*) = \operatorname{diag}(d_j)$, then $Y$ on line 16 will be in $\mathcal{B}^C(\varepsilon)$ and for all columns of $Y$ that are eigenvectors of $\Lambda$ the corresponding equality constraints will satisfy $c_i(Y) = 0$. If the latter conditions are not fulfilled the trivial solution discussed in section 2 is returned on line 17, which is feasible, however, does not depend on the input parameter $Y$. Hence, this algorithm returns a point in $\mathcal{B}^C(\varepsilon)$ at which all equality constraints that are not satisfied have gradients that are linearly independent. The next lemma shows that under some assumptions the conditions for the if-clause on line 16 are always false. Hence, Algorithm 5.1 does not return the trivial solution from section 2.

LEMMA 5.3. *Let $r := |\mathcal{I}| > 0$ with $\mathcal{I}$ as defined in Algorithm 5.1. If all diagonal elements of $\Delta$ satisfy $\delta_i \in [\lambda_{2(p-r)+i}, \lambda_{n-2(p-r)-p+i}] \supset [\lambda_{3p-2}, \lambda_{n-3p+3}]$ for $i = 1,\ldots,p$ then the optimal solution of* (2.1) *on line 14 in Algorithm 5.1 attains the function value zero.*

*Proof.* Let $m \geq n-2(p-r)$ be the number of columns of $Q$ in Algorithm 5.1. For $T = \operatorname{diag}(t_1,\ldots,t_r) = \operatorname{diag}(d_i)$ and $i \in \mathcal{I}$ it is enough to show that for $j \in \{1,\ldots,r\}$ arbitrary $t_j \in [\mu_j, \mu_{m+j-r}]$ where $\mu_1,\ldots,\mu_m$ are the eigenvalues of $Q^T\Lambda Q$ on line 14. Let $\sigma : \{1,\ldots,r\} \mapsto \mathcal{I}$ be defined as the map that satisfies $t_k = d_{\sigma(k)}$ for all $k = 1,\ldots,r$. Now without loss of generality we can assume that $d_{\sigma(j)} \leq \delta_{\sigma(j)}$. Then from Algorithm 5.1 on line 6 it follows that $y_{\sigma(j)}^T \Lambda y_{\sigma(j)} \leq d_{\sigma(j)} = t_j \leq \delta_{\sigma(j)}$. Since $\mu_j \leq y_{\sigma(j)}^T \Lambda y_{\sigma(j)}$ we obtain $\mu_j \leq t_j$. From our assumption it holds that $\delta_{\sigma(j)} \leq \lambda_{n-2(p-r)-p+\sigma(j)} \leq \lambda_{m-r+j} \leq \mu_{m-r+j}$ as $\mu_{m-r+j} \in [\lambda_{m-r+j}, \lambda_{n-r+j}]$ and $\sigma(j) \leq p-r+j$. □

Therefore if the assumption of Lemma 5.3 is satisfied the major cost of Algorithm 5.1 is to compute once the eigenvalues of $Q^T\Lambda Q$ and only if needed. Now we are ready to state Algorithm 5.2 that minimizes an arbitrary smooth function over $\mathcal{C}$. In the next section we will show the convergence of a subsequence of the iterates generated by this algorithm to a stationary point of (5.2).

## 6. Convergence.

In order to show convergence of Algorithm 5.2 we need the following lemma, showing that for $\mu_k$ large enough the iterate that is returned by Algorithm 5.1 on line 9 in Algorithm 5.2 will attain a smaller function value in $G_{\mu_k,\theta^k}(\cdot)$ than $\widehat{Y}$ under the assumption that the iterate has been altered by Algorithm 5.1.

LEMMA 6.1. *Let $\gamma > 0$, $\varepsilon > 0$, and $\widehat{Y}$ be the iterate in iteration $k$ that is returned by the nonlinear CG on line 3 in Algorithm 5.2. Further assume that $\widehat{Y} \notin \mathcal{B}^C(\varepsilon/2)$ or it exists an index $s$ with $(\hat{y}_s, d_s)$ an eigenpair of $\Lambda$ and $|c_s(\widehat{Y})| > 0$. Let $\widetilde{Y}$ be the point that is returned by Algorithm 5.1 on line 9. Then there exists a $\bar{\mu} > 0$ such that for $\mu_k \geq \bar{\mu}$ it holds that*

$$G_{\mu_k,\theta^k}(\widehat{Y}) - G_{\mu_k,\theta^k}(\widetilde{Y}) \geq \gamma.$$

*Proof.* Let

$$\alpha := \begin{cases} \min\left\{\frac{\varepsilon}{4}, \eta\right\} & \text{if there exists an index } s, \\ \frac{\varepsilon}{4} & \text{otherwise,} \end{cases}$$

**Algorithm 5.2** Algorithm to solve (5.2) based on [22, Framework 17.3] of the augmented Lagrangian method.

```
Require: smooth function $f$ defined on $\mathcal{B}(n,p)$, $N$, $\mathit{\Delta}$, $n, p, \varepsilon$.
Generate a starting point $Y_0 \in \mathcal{B}(n,p)$, set initial values for $\mu_0 > 0$, $\theta_0$, $\gamma > 0$, tolerance $\tau$, initial tolerance $\tau_0$, and constraint violation tolerance $\nu$. $k = 0$.
while $|| \operatorname{grad} G_{\mu_k,\theta^k}(Y_k)||_F \geq \tau$ or $||c(Y_k)||_2 \geq \nu$ do
  Find an approximate minimizer of $G_{\mu_k,\theta^k}(\cdot)$ in $\mathcal{B}^C(\varepsilon/2)$, starting at $Y_k$, by applying the nonlinear CG-method in [7, Algorithm 3.9.1] with a modified Armijo-backtracking strategy: if the new iterate $Y_k^i$ that yields sufficient descent lies outside of $\mathcal{B}^C(\varepsilon/2)$ then, if needed, the step size is reduced such that $Y_k^i \in \mathcal{B}(n,p)$ and $Y_k^i \notin \mathcal{B}^C(\varepsilon/2)$. The iterate $Y_k^i$ in the nonlinear CG-method is returned. If the latter does not occur terminate when $|| \operatorname{grad} G_{\mu_k,\theta^k}(Y_k^i)||_F \leq \tau_k$. Set the returned point to $\widehat{Y}$.
  if $\widehat{Y} \in \mathcal{B}^C(\varepsilon/2)$ then
    $\hat{\varepsilon} \leftarrow \varepsilon/2$
  else
    $\hat{\varepsilon} \leftarrow \varepsilon$
  end if
  Apply Algorithm 5.1 with $Y = \widehat{Y}$, $\mathit{\Delta} = \mathit{\Delta}$, $\varepsilon = \hat{\varepsilon}$; set returned point to $Y_{k+1}$.
  if $\widehat{Y} = Y_{k+1}$ then
    $\theta^{k+1} = \theta^k - \mu_k c(Y_k)$.
    Select tolerance $\tau_{k+1} \leq \tau_k$.
  end if
  if $|| \operatorname{grad} G_{\mu_k,\theta^k}(Y_{k+1})||_F > \tau_k$ and $G_{\mu_k,\theta^k}(\widehat{Y}) - G_{\mu_k,\theta^k}(Y_{k+1}) > \gamma$ then
    $\mu_{k+1} \leftarrow \mu_k$
  else
    Select $\mu_{k+1} > \mu_k$.
  end if
  $k \leftarrow k + 1$
end while
return $Y_k$
```

where $\eta := \min\{|\lambda_j - \delta_i| : |\lambda_j - \delta_i| > 0 \text{ and } i \in \{1,\ldots,p\}, j \in \{1,\ldots,n\}\}$. Then $\alpha$ is a lower bound for the minimal change in modulus in one of the constraint functions $c_j(\cdot)$ when going from $\widehat{Y}$ to $\widetilde{Y}$. To explain more in detail, if $\widehat{Y} \notin \mathcal{B}^C(\varepsilon/2)$ then after applying Algorithm 5.1 $\widetilde{Y} \in \mathcal{B}^C(\varepsilon)$ therefore the minimal change in one of the constraint function must be at least $\varepsilon/4$. On the other hand, if there exists an index $s$ and $\widehat{Y} \in \mathcal{B}^C(\varepsilon/2)$ either the change in $c_s(\cdot)$ is greater than or equal to $\varepsilon/4$ or $d_s$ in Algorithm 5.1 will be set to $\delta_s$ where $|d_s - \delta_s| \geq \eta$ as $c_s(Y) > 0$. Hence, it exists an index $j$ so that $|c_j(\widetilde{Y}) - c_j(\widehat{Y})| =: \alpha_j \geq \alpha > 0$, which implies that $c_j(\widetilde{Y}) = c_j(\widehat{Y}) - \operatorname{sgn}(c_j(\widehat{Y}))\alpha_j$. Note that $\alpha > 0$ is a constant that is independent of the iteration $k$ and the point $\widehat{Y}$. As $f(Y) - \sum_{i=1}^{p} \theta_i^k c_i(Y)$ is smooth there exists a

Lipschitz constant $L$ with $|f(\widehat{Y}) - f(\widetilde{Y})| \leq L||\widehat{Y} - \widetilde{Y}||_F$ and we have

$$\begin{aligned} G_{\mu_k,\theta^k}(\widehat{Y}) - G_{\mu_k,\theta^k}(\widetilde{Y}) &= f(\widehat{Y}) - f(\widetilde{Y}) - \sum_{i=1}^{p} \theta_i^k (c_i(\widehat{Y}) - c_i(\widetilde{Y})) \\ &\quad + \frac{\mu_k}{2} \sum_{i \neq j} \underbrace{\left(c_i(\widehat{Y})^2 - c_i(\widetilde{Y})^2\right)}_{\geq 0} + \frac{\mu_k}{2} \left(c_j(\widehat{Y})^2 - c_j(\widetilde{Y})^2\right) \\ &\geq -L||\widehat{Y} - \widetilde{Y}||_F + \frac{\mu_k}{2} \left(2\text{sgn}(c_j(\widehat{Y}))\alpha_j c_j(\widehat{Y}) - \alpha_j^2\right) \\ &\geq -L\sqrt{p} + \frac{\mu_k}{2}\alpha_j^2 \geq -L\sqrt{p} + \frac{\mu_k}{2}\alpha^2. \end{aligned}$$

Hence, the claim holds for $\mu_k \geq 2\frac{L\sqrt{p}+\gamma}{\alpha^2}$. □

Now we can state our convergence result, which is based on [5, Proposition 2.3].

Theorem 6.2. *Let $\varepsilon > 0$ and let $\Lambda \in \mathbb{R}^{n\times n}$ and $\Delta \in \mathbb{R}^{p\times p}$ be diagonal and suppose there exists a $Y \in \mathcal{B}^C(\delta)$ with $Y^T\Lambda Y = \Delta$ and $\delta > \varepsilon$. Let further $f$ be a bounded real smooth function over $\mathcal{B}(n,p)$. If the Lagrange multipliers $\{\theta^k\}_{k\geq 0}$ in Algorithm 5.2 are bounded, the sequence $\{\mu_k\}_{k\geq 0}$ satisfies $\mu_k \to \infty$, and for $\{\tau_k\}_{k\geq 0}$ holds that $\tau_k \geq 0$ for all $k$ with $\tau_k \to 0$, then a subsequence of the iterates generated by Algorithm 5.2 converges to a stationary point of* (5.2)*. Hence, for $\nu > 0$ this algorithm terminates.*

*Proof.* Let $\{Y_k\}_{k\geq 0}$ be the sequence generated by Algorithm 5.2 with $Y_k \in \mathcal{B}(\varepsilon/2)$. If on line 3 of Algorithm 5.2 the nonlinear CG fails to find a point $Y_k^i$ with $||\operatorname{grad} G_{\mu_k,\theta^k}(Y_k^i)||_F \leq \tau_k$ then on line 10 $Y_k \neq \widehat{Y}$ so that the Lagrange multipliers $\theta^k$ and the tolerance parameter $\tau_k$ will not be altered. Hence, either $\operatorname{grad} G_{\mu_k,\theta^k}(Y_k)$ will become zero as $k$ further increases or $\mu_k$ will be large enough so that for all further $k$ it holds that $G_{\mu_k,\theta^k}(Y_{k-1}) - G_{\mu_k,\theta^k}(Y_k) \geq \gamma$. As in the latter case $\mu_k$ is not further increased, $G_{\mu_k,\theta^k}(\cdot)$ is bounded from below, and $\mathcal{B}^C(\varepsilon/2)$ compact, we must have for $k$ large enough

$$||\operatorname{grad} G_{\mu_k,\theta^k}(Y_k)||_F \leq \tau_k.$$

Therefore, there exists a subsequence $\{l\}_{l\geq 0}$ of $\{k\}_{k\geq 0}$, for which it holds that for all $l$ in this sequence

$$||\operatorname{grad} G_{\mu_l,\theta^l}(Y_l)||_F \leq \tau_l \text{ and } 0 < \mu_l < \mu_{l+1} \text{ with } \mu_l \to \infty.$$

As $\mathcal{B}^C(\varepsilon/2)$ is bounded there exists a subsequence $\{Y_s\}_{s\geq 0}$ of $\{Y_l\}_{l\geq 0}$ that converges to a point $Y_*$. From Lemma 5.2 and Algorithm 5.1 it must hold that at this point the constraint gradients $\operatorname{grad} c_i(Y_*)$ with $c_i(Y_*) \neq 0$ are linearly independent and all other constraint gradient are zero. Let $\mathcal{J} := \{i : c_i(Y_*) \neq 0\}$ and $\widehat{c}(Y) := [c_i(Y)]_{i\in\mathcal{J}}$. We define for all $s$

$$\widehat{\theta}^s := [\theta_i^s]_{i\in\mathcal{J}} - \mu_s \widehat{c}(Y_s).$$

Then we have that

$$\begin{aligned} \operatorname{grad} G_{\mu_s,\theta^s}(Y_s) &= \operatorname{grad} f(Y_s) - \sum_{i=1}^{s} (\theta_i^s - \mu_s c_i(Y_s)) \operatorname{grad} c_i(Y_s) \\ &= \operatorname{grad} f(Y_s) - \sum_{i\in\mathcal{J}} \operatorname{grad} \widehat{c}_i(Y_s)\widehat{\theta}_i^s + \zeta_s, \end{aligned}$$

with $\zeta_s \to 0$, and for all $s$ such that $\operatorname{grad} \widehat{c}(Y_s) := [\operatorname{vec}(\operatorname{grad} c_i(Y_s))]_{i \in \mathcal{J}} \in \mathbb{R}^{np \times |\mathcal{J}|}$ has full column rank

$$\widehat{\theta}^s = \left(\operatorname{grad} \widehat{c}(Y_s)^T \operatorname{grad} \widehat{c}(Y_s)\right)^{-1} \operatorname{grad} \widehat{c}(Y_s)^T (\operatorname{grad} f(Y_s) + \zeta_s - \operatorname{grad} G_{\mu_s, \theta^s}(Y_s)).$$

As $\operatorname{grad} G_{\mu_s,\theta^s}(Y_s) \to 0$, it follows that $\widehat{\theta}^s \to \widehat{\theta}_*$ with

$$\widehat{\theta}_* = \left(\operatorname{grad} \widehat{c}(Y_*)^T \operatorname{grad} \widehat{c}(Y_*)\right)^{-1} \operatorname{grad} \widehat{c}(Y_*)^T \operatorname{grad} f(Y_*).$$

Since $\theta^s$ is bounded and $\{[\theta_i^s]_{i \in \mathcal{J}} - \mu_s \widehat{c}(Y_s)\}_{s \geq 0} \to \widehat{\theta}_*$, it follows that $\{\mu_s \widehat{c}(Y_s)\}_{s \geq 0}$ is bounded and hence as $\mu_s \to \infty$ $\operatorname{grad} f(Y_*) = \operatorname{grad} c(Y_*)\theta_*$ and $c(Y_*) = 0$. □

**7. Numerical Tests.** As the initial eigenvalue decomposition of $N$ is the major cost to compute the optimal solution of problem (2.1) and the active-set method for problem (3.1) converges in at most $2p$ iterations we do not expect to gain further insight in the performance of the corresponding algorithms by applying them to test examples. Therefore we focus in this section on investigating the performance of Algorithm 5.2. For our tests we use a machine with 32 AMD Opteron(TM) Processors (2999MHz) and 256GB RAM, on Linux 64bit in MATLAB R2010a. Let us now specify our test examples.

**7.1. Test Problem and Test Matrices.** As we currently do not have an objective function available that can be used in our application in atomic chemistry for (5.1) we use only a test function, which is a convex quadratic function with randomly generated coefficients. Let $f(Y) := \langle Y, AY \rangle + \langle B, Y \rangle$ be a function from $\mathcal{B}(n,p) \mapsto \mathbb{R}$ with $A \in \mathbb{R}^{n \times n}$, and $B \in \mathbb{R}^{n \times p}$. To generate $A$ and $B$ we use the MATLAB function `rand` where we ensure that $A$ is symmetric positive semidefinite by runnig `A=rand(n,n); A=A'*A/2`. In order to apply Algorithm 5.2 to (5.1) we also need test matrices for $\Delta$ and $\Lambda$. We look at two different classes whereas the first is more for demonstrating purposes.

- **ldchem**: A. Sax provided us with a small example for $N \in \mathbb{R}^{11 \times 11}$ in (2.1). If we prescribe two orbitals with occupation numbers 1.5 and 0.1 then from (2.2) we find that $\Delta$ is $\Delta = \operatorname{diag}(0.1, 1.5)$. For more details on this test example see [7, Section 4.7.3]
- **ldrand**: For the second class we randomly generate a symmetric matrix $N \in \mathbb{R}^{n \times n}$ and a diagonal matrix $T \in \mathcal{D}(p)$, respectively by means of the MATLAB commands `N=rand(n,n);N=N+N';` and `diag(sort(rand(p,1)*p));`, respectively. Then we set, accordingly to (2.2), $\Delta$ to the closest diagonal matrix that is embeddable in $N$. If one diagonal element of $\Delta$ is within the range of 0.01 of another we repeat the process of generating $\Delta$.

For our starting matrix $Y_0$ in Algorithm 5.2 we randomly generate a matrix $Y \in \mathsf{St}(n,p)$ by applying `rand` again and computing the $Q$-factor of the randomly generated matrix by means of `qr`. Thereafter we set $Y_0 = YP$ where $P \in \mathsf{O}(p)$ computed by `eig` diagonalizes $Y^T \Lambda Y$ with the diagonal elements increasing. If the distance between two diagonal elements is less than 0.01 we repeat the procedure, making sure that $Y_0 \in \mathcal{B}(n,p)$.

**7.2. Numerical Methods.** In addition to Algorithm 5.2 we use the following algorithm for comparing purposes. We will refer to this algorithm as ALS. Let $c_{ij} : \mathbb{R}^{n \times p} \mapsto \mathbb{R}$ be defined as $c_{ij}(Y) = (Y^T \Lambda Y)_{ij}$ for $i > j$ and $c_{ii}(Y) = (Y^T \Lambda Y)_{ii} - D_{ii}$ for $i = j$ and $i,j = 1, \ldots, p$. We reformulate (5.1) as

$$\begin{array}{lll} \min_{Y \in \mathsf{St}(n,p)} & f(Y) & \\ \text{s.t.} & c_{ij}(Y) = 0 & i,j = 1,\ldots,p \text{ and } i \geq j \end{array} \tag{7.1}$$

and apply, similarly to Algorithm 5.2, the augmented Lagrangian method [22, Algorithm 17.4] to (7.1). The inner problem is then to minimize the following augmented Lagrangian function

$$G_{\mu,\theta}(Y) = f(Y) - \sum_{i \geq j} \theta_{ij} c_{ij}(Y) + \frac{\mu}{2} \sum_{i \geq j} c_{ij}(Y)^2 \tag{7.2}$$

over the Stiefel manifold. In (7.2) $\theta_{ij}$ are the Lagrange multipliers for $i \geq j$. To minimize $G_{\mu,\theta}(Y)$ we use again the nonlinear CG method.

**7.3. Chosen Specifications in Tested Algorithms.** We now list the most important specifications that we used in both methods. Most often we refer to Algorithm 5.2 in this context as we choose the parameters for ALS analogously. For more details on these configurations including the reasons why we have chosen them as specified we refer to [7, Section 4.7.4].

- We use $\varepsilon = 0.01$, $\tau = np^2 10^{-5}$ , $\theta_0 = 0$, $\gamma = 10^{-10}||\varLambda||_F$, $\nu = 10^{-4}np$, and $\mu_0 = 10$ and if $\mu_k$ needs to be enlarged we increase it by a factor of 2.
- The parameter $\tau_k$ in Algorithm 5.2 is chosen as follows. We start with $\tau_0 := np\mu_0$ and tighten it by setting $\tau_{k+1} := \max\{\tau, \tau_k/\mu_k\}$ if the current violation satisfies $||c(Y_k)||_2 \leq \nu_k$, and otherwise, by setting $\tau_{k+1} := np/\mu_k$.
- The parameter $\nu_k$ is determined by $\nu_0 := 1/\mu_0^{0.1}$ and $\nu_{k+1} := \nu_k/\mu_k^{0.9}$ if $||c(Y_k)||_2 \leq \nu_k$, and otherwise, $\nu_{k+1} := 1/\mu_k^{0.1}$. These specifications are based on [22, Algorithm 17.4].
- We limit the number of iterations in the augmented Lagrangian method to 100.
- In the nonlinear CG method we set the maximal number of iterations to $150,000$ and use a conventional Armijo-backtracking procedure where we compute a guess for the initial step length as follows. Let $Y_k^i$ be the current iterate and $\xi$ be the current direction then we take the largest solution of

$$\min_{t \in (0,1]} G_{\mu_k,\theta_k}(Y_k^i + t\xi_{Y_i}), \tag{7.3}$$

  as our initial step length.
- For the retraction on the Stiefel manifold in ALS we use the $Q$-factor as described in [2, Example 4.1.3]. For Algorithm 5.2 we use the retraction that we introduced in Lemma 4.5 and choose for $\widehat{R}_Y(H)$ again the $Q$-factor.
- For the vector transport we apply for both algorithms the one proposed in [7, Eq. (3.10)] that uses the projection onto the tangent space. To compute the latter for Algorithm 5.2 we will use the approach discussed in Section 4.3 and will solve the linear system by applying `chol` provided by MATLAB to compute the Cholesky factorization as we observed in our numerical tests that the coefficient matrix is often strictly diagonal dominant. If it fails we will use the routine `ldl` in MATLAB for the $LDL^T$ decomposition. We also tried to exploit the sparsity property of the coefficient matrix in Lemma 4.4 but our tests have shown that for the matrix sizes tested no benefit can be gained by using sparse linear solvers [7, Section 4.7.4].

**7.4. Test 1.** Our first test is more for demonstrating purposes. We first generate a convex quadratic function $f(Y)$ and apply then Algorithm 5.2 to minimize it for the matrices $\varLambda$ and $\varDelta$ of type **ldchem**. We generate 100 different starting points $Y_0$

TABLE 7.1
*Output of Algorithm 5.2 for test matrices* **ldchem**.

| | **Test 1a** | **Test 1b** |
|---|---|---|
| Outer iterations | 22 | 24 |
| Total number of iterations in CG | 145 | 34 |
| Total number of backtracking steps in CG | 3 | 3 |
| $\operatorname{grad} G_{\mu_*,\theta_*}(Y_*)$ | 1.258e-7 | 1.4e-13 |
| $\mu_*$ | 80 | 80 |
| Computational time in seconds | 1 | 0.4 |
| $f(Y_0)$ | 10.96 | 0 |
| $f(Y_*)$ | -0.78321 | 0 |
| $f(Y_{\text{ana}})$ | 1.21393 | 0 |
| Constraint violation $\|\|c(Y_*)\|\|_2$ | 3.8e-10 | 9.15e-16 |
| Calls of Algorithm 5.1 where $D$ was modified | 0 | 1 |
| Number of independent gradients $\operatorname{grad} c_i(Y_*)$ | 2 | 2 |

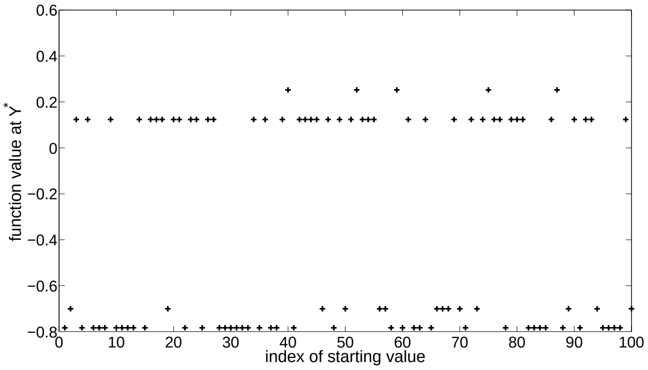

FIG. 7.1. *Function value at final iterate of Algorithm 5.2 for different starting values.*

and apply our algorithm starting for each of them. In Figure 7.1 we plot the function value at the final iterate for all 100 different starting values and we present in the second column of Table 7.1 the output of Algorithm 5.2 for the point with smallest function value where we use for this test a violation tolerance of $\nu = 10^{-10}$ and a tolerance of $\tau = 4.4 \times 10^{-8}$. Note that we call $Y_*$ the final iterate of Algorithm 5.2, $\mu_*$ the corresponding penalty parameter, and $\theta_*$ the Lagrange multipliers. We denote the point that we discussed in section 2 and that can be analytically computed by $Y_{\text{ana}}$. In Figure 7.1 we see that we converge to different points when we use different starting values so that we could apply Algorithm 5.2 to find different stationary points. In Table 7.1 we see that the optimization was successful as the function value has been reduced and the gradient norm is small. Notice that the parameter $\mu_*$ is relatively small. Interesting is that during the iteration there was no need in Algorithm 5.1 to modify the $d_i$s. Therefore, let us apply this algorithm for $f(Y) = 0$ and for the starting point $Y_0 = [I_p \ 0]^T$. As at this point the LICQ does not hold, the $\operatorname{grad} G_{\mu_k,\theta_k}(Y)$ is zero, and the constraints are not satisfied. Our results are present in the last column of Table 7.1. We observe that due to the modification in Algorithm 5.1 the Algorithm 5.2 overcomes the point $Y_0$ where the LICQ does not hold and does not break down. Let us now look at the performance of Algorithm 5.2 for larger matrix sizes.

TABLE 7.2
*Results for the randomly generated matrices $\Lambda$ and $D$.*

| | | Algorithm 5.2 | | | | | ALS | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\overline{\mathbf{f_0}}$ | $\overline{\mathbf{t}}$ | $\overline{\mathbf{it}}$ | $\overline{\mathbf{fv}}$ | $\overline{\mathbf{gradn}}$ | $\overline{\mathbf{crp}}$ | $\overline{\mathbf{t}}$ | $\overline{\mathbf{it}}$ | $\overline{\mathbf{fv}}$ | $\overline{\mathbf{gradn}}$ |
| | | | | | $n = 50$ | | | | | |
| $p = 10$ | 325 | 119 | 2480 | -9.49 | 0.04 | 1.4 | 368 | 9553 | -9.42 | 0.05 |
| $p = 20$ | 344 | 785 | 9615 | 53.24 | 0.20 | 0 | 8465 | 2.1e5 | 54.55 | 0.20 |
| $p = 30$ | 397 | 580 | 3376 | 127.29 | 0.41 | 0.2 | 5004 | 1.2e5 | 124.26 | 0.43 |
| $p = 40$ | 1.2e3 | 875 | 2242 | 163.45 | 0.71 | 0.2 | 6183 | 1.4e5 | 181.54 | 0.77 |
| | | | | | $n = 150$ | | | | | |
| $p = 10$ | 2.8e3 | 181 | 3536 | -12.53 | 0.14 | 1.4 | 145 | 3533 | -12.57 | 0.13 |
| $p = 20$ | 2.9e3 | 377 | 4312 | -26.91 | 0.53 | 1 | 1121 | 2.6e4 | -27.16 | 0.53 |
| $p = 30$ | 3.0e3 | 433 | 2581 | -32.22 | 1.28 | 0 | 1981 | 4.3e4 | -32.10 | 1.32 |
| $p = 40$ | 4.9e3 | 1121 | 2913 | -12.54 | 2.09 | 0 | 1947 | 3.7e4 | -12.53 | 2.13 |
| | | | | | $n = 250$ | | | | | |
| $p = 10$ | 7.8e3 | 318 | 5856 | -7.63 | 0.25 | 2 | 209 | 4877 | -7.58 | 0.23 |
| $p = 20$ | 8.0e3 | 706 | 7762 | -23.52 | 0.89 | 2.2 | 871 | 1.8e4 | -23.32 | 0.88 |
| $p = 30$ | 8.1e3 | 1067 | 6012 | -46.19 | 2.19 | 1.4 | 3093 | 5.4e4 | -46.57 | 2.23 |
| $p = 40$ | 11e4 | 1228 | 2854 | -58.53 | 3.51 | 0.2 | 3832 | 5.7e4 | -59.08 | 3.73 |

**7.5. Test 2.** We compare the performance of Algorithm 5.2 with ALS. For this reason we first generate randomly 5 instances of $\Lambda$ and $T$ of type **ldrand** and coefficient matrices for the convex quadratic function $f(Y)$ for $n = 50, 100, \ldots, 300$ and $p = 10, \ldots, 40$. Then we apply Algorithm 5.2 and ALS to our problem (5.1) with these matrices. A selection of our results is shown in Table 7.2 where we use the following abbreviations:

- $\overline{\mathbf{t}}$: mean computational time in seconds to compute the final iterate $Y_*$,
- $\overline{\mathbf{it}}$: mean total number of iterations in the nonlinear CG method,
- $\overline{\mathbf{fv}}$: mean function value at $Y_*$,
- $\overline{\mathbf{gradn}}$: mean $||\operatorname{grad} f(Y_*)||_F$,
- $\overline{\mathbf{crp}}$: mean number of calls of Algorithm 5.1.

We see in Table 7.2 that in most tests Algorithm 5.2 outperforms ALS in terms of the computational time. The main reason is that the Algorithm 5.2 needs much fewer iterations in the nonlinear CG method to satisfy the stopping criterion. We do not report the number of the outer iterations as this number does not vary much with $n$ or $p$ and is on average in the range of 5 to 13. The penalty parameter lies for both algorithms on average between 80 and 4000, which is of moderate size.

We also observe that the cost per iteration is more expensive in Algorithm 5.2 than in ALS and the relative difference is increasing with $p$. An explanation is clearly that the cost to compute the projection onto the normal space of $\mathcal{B}(n, p)$ is of order $\mathcal{O}(p^6)$. To demonstrate this more illustratively we plot the fraction of the time taken to compute the projection to the total time in Figure 7.2 for $n = 200$ and $p = 5, 10, \ldots, 50$. For $p = 50$ approximately 81% of the runtime of the code is spent to compute projection onto $T_Y\mathcal{B}(n, p)$.

## 8. Conclusion.

Motivated by an application in atomic chemistry we addressed the problem of embedding a symmetric matrix into the set of diagonal matrices. We started with investigating two two-sided optimization problems and showed that they generally do not have unique optimal solutions. We proposed algorithms to find optimal solutions of either problems whose major cost are a few eigenvalue decompositions.

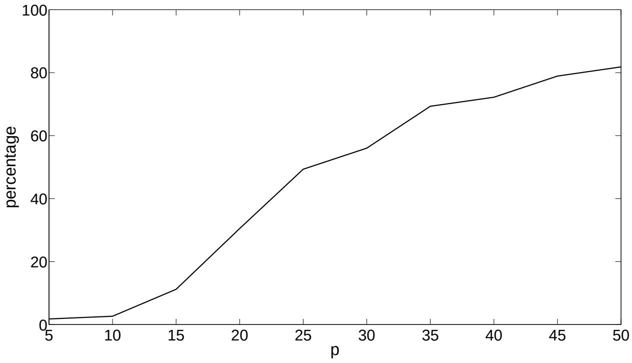

FIG. 7.2. *Ratio of time spent on computing the projection to total time.*

Then we analyzed the optimal set $\mathcal{C}$ of solutions of the first problem and gained deeper theoretical understanding of this set by analyzing the geometric properties of a similar set. We proposed an augmented Lagrangian-based algorithm that can optimize a smooth objective function over $\mathcal{C}$ by using geometric tools and showed convergence to a stationary point. The latter property can generally not be guaranteed for the augmented Lagrangian method ALS that we used for our comparison. Moreover, our numerical tests demonstrated a convincing performance of our new algorithm as it outperformed the latter approach. Surprisingly, this even holds for moderate sized $p$, although the computation of the projection onto $T_Y\mathcal{B}(n,p)$, which is required by the new algorithm, is of $\mathcal{O}(p^6)$ and hence, expensive. We conclude therefore that for moderate sized $p$ this new algorithm is a good choice for optimizing a smooth function over $\mathcal{C}$ as it guarantees global convergence and performs well.

**Acknowledgements.** We thank Prof. Alexander Sax for attracting us to this interesting topic and providing us with the details related to the application in atomic chemistry. Many thanks go also to Prof. Higham for many helpful comments on this subject and to Prof. Absil for directing us to [9] and pointing out the connection of $\mathcal{B}(n,p)$ to the Grassmannian manifold.

## REFERENCES

[1] P.-A. Absil, C. G. Baker, and K. A. Gallivan, *Trust-region methods on Riemannian manifolds*, Found. Comput. Math., 7 (2007), pp. 303–330.

[2] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, 2008.

[3] P.-A. 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.

[4] S. I. Amari, T. P. Chen, and A. Cichocki, *Nonholonomic orthogonal learning algorithms for blind source separation*, Neural Computation, 12 (2000), pp. 1463–1484.

[5] D. P. Bertsekas, *Constrained optimization and Lagrange multiplier methods*, vol. 1 of Computer science and applied mathematics, Academic Press, 1982.

[6] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, USA, 1999.

[7] R. Borsdorf, *Structured Nearness Matrix Problems: Theory and Algorithms*, PhD thesis, The University of Manchester, Manchester, UK, 2012. Available at `http://eprints.ma.man.ac.uk/1841/`.

[8] R. Borsdorf and N. J. Higham, *A preconditioned Newton algorithm for the nearest correlation matrix*, IMA J. Numer. Anal., 30 (2010), pp. 94–107.

[9] T. P. Cason, P.-A. Absil, and P. Van Dooren, *Comparing two matrices by means of isometric projections*, in Numerical Linear Algebra in Signals, Systems and Control, S. P.; Chan R. H.; Olshevsky V.; Routray A. Van Dooren, P.; Bhattacharyya, ed., vol. 80 of Lecture Notes in Electrical Engineering, Springer-Verlag, 2011, pp. 77–93.

[10] A. Del Bue, M. Stosic, M. Dodig, and J. Xavier, *2D-3D registration of deformable shapes with manifold projection*, in Proceedings of the 16th IEEE international conference on Image processing, IEEE Press, 2009, pp. 1057–1060.

[11] L. Dieci and T. Eirola, *On smooth decompositions of matrices*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 800–819.

[12] A. Edelman, T. A. Arias, and S. T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 303–353.

[13] K. Fan and G. Pall, *Imbedding conditions for Hermitian and normal matrices*, Canad. J. Math., 9 (1957), pp. 298–304.

[14] I. Grubišić and R. Pietersz, *Efficient rank reduction of correlation matrices*, Linear Algebra Appl., 422 (2007), pp. 629–653.

[15] N. J. Higham, *Computing the nearest correlation matrix—A problem from finance*, IMA J. Numer. Anal., 22 (2002), pp. 329–343.

[16] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 1991.

[17] M. Ishteva, P.-A. Absil, S. Van Huffel, and L. De Lathauwer, *Best low multilinear rank approximation of higher-order tensors, based on the Riemannian trust-region scheme*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 115–135.

[18] J. Ivanic, G. J. Atchity, and K. Ruedenberg, *Intrinsic local constituents of molecular electronic wave functions. I. Exact representation of the density matrix in terms of chemically deformed and oriented atomic minimal basis set orbitals*, Theoretical Chemistry Accounts: Theory, Computation, and Modeling (Theoretica Chimica Acta), 120 (2008), pp. 281–294.

[19] J. M. Lee, *Introduction to Smooth Manifolds*, Springer-Verlag, 2003.

[20] W. C. Lu, C. Z. Wang, M. W. Schmidt, L. Bytautas, K. M. Ho, and K. Ruedenberg, *Molecule intrinsic minimal basis sets. I. Exact resolution of ab initio optimized molecular orbitals in terms of deformed atomic minimal-basis orbitals*, The Journal of Chemical Physics, 120 (2004), pp. 2629–2637.

[21] J. H. Manton, R. Mahony, and Y. Hua, *The geometry of weighted low-rank approximations*, IEEE Trans. Signal Processing, 51 (2003), pp. 500–514.

[22] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, 2nd ed., 2006.

[23] B. N. Parlett and G. Strang, *Matrices with prescribed Ritz values*, Linear Algebra and its Applications, 428 (2008), pp. 1725–1739.

[24] K. Rahbar and J. Reilly, *Geometric optimization methods for blind source separation of signals*, in Second International Workshop on Independent Component Analysis and Blind Signal Separation (ICA 2000), Helsinki, Finland, June 2000.

[25] A. Shaji, S. Chandran, and D. Suter, *Manifold optimisation for motion factorisation*, in Pattern Recognition, 2008. ICPR 2008. 19th International Conference on, IEEE, 2009, pp. 1–4.

[26] S. T. Smith, *Optimization techniques on Riemannian manifolds*, in Hamiltonian and gradient flows, algorithms and control, vol. 3 of Fields Inst. Commun., Amer. Math. Soc., Providence, RI, 1994, pp. 113–136.

[27] B. Vandereycken, *Riemannian and multilevel optimization for rank-constrained matrix problems*, PhD thesis, Katholieke Universiteit Leuven, 2010.

[28] W. H. Yang and L. H. Zhang, *Optimality conditions of the nonlinear programming on Riemannian manifolds*, (2011). `http://www.optimization-online.org/DB_FILE/2011/08/3124.pdf`.