NORGES TEKNISK-NATURVITENSKAPELIGE
UNIVERSITET

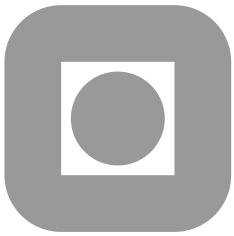
Descent Methods for Optimization on Homogeneous Manifolds

by

Elena Celledoni [1] *and Simone Fiori* [2]

Affiliations:
[1] Department of Mathematical Sciences
NTNU, Trondheim , Norway
E-mail: elenac@math.ntnu.no
[2] Dipartimento di Elettronica, Intelligenza Artificiale
e Telecomunicazioni (DEIT)
Università Politecnica delle Marche, Ancona, Italy
E-mail: fiori@deit.univpm.it

PREPRINT
NUMERICS NO. 1/2007



NORWEGIAN UNIVERSITY OF
SCIENCE AND TECHNOLOGY
TRONDHEIM, NORWAY

This report has URL
http://www.math.ntnu.no/preprint/numerics/2007/N1-2007.pdf
Address: Department of Mathematical Sciences, Norwegian University of Science and Technology, N-7491 Trondheim, Norway.

Descent Methods for Optimization on Homogeneous Manifolds

Elena Celledoni [1] *and Simone Fiori* [2]

Affiliations:

[1] Department of Mathematical Sciences
NTNU, Trondheim , Norway
E-mail: elenac@math.ntnu.no

[2] Dipartimento di Elettronica, Intelligenza Artificiale
e Telecomunicazioni (DEIT)
Università Politecnica delle Marche, Ancona, Italy
E-mail: fiori@deit.univpm.it

January 11, 2007

In this article we present a framework for line search methods for optimization on smooth homogeneous manifolds, with particular emphasis to the Lie group of real orthogonal matrices. We propose strategies of univariate descent (UVD) methods. The advantages of this approach are that the optimization problem on the manifold is broken down into one-dimensional optimization problems and that each optimization step by itself involves little computation effort.

We apply the devised method to eigen-problems as well as to independent component analysis, in order to assess its numerical features.

Keywords: Eigen-problems; Gradient-based optimization; Independent component analysis; Retractions on manifolds; Signal processing; Smooth parameters manifolds.

1 Introduction

In the present manuscript, we consider optimization problems of the type:

$$\min_{x\in\mathcal{M}} \phi(x) \quad \text{or} \quad \max_{x\in\mathcal{M}} \phi(x), \tag{1}$$

where $\mathcal{M}$ is a Lie group or an homogeneous manifold, $\phi : \mathcal{M} \to \mathbf{R}$, is a cost function to be minimized or an objective function to be maximized.

We will consider iterative methods to approximate optimum that have the form:

$$x_{k+1} = \varphi_{x_k}(\pm\alpha_k p_k),\ k = 1,\ 2,\ 3,\ \ldots, \tag{2}$$

where $T_{x_k}\mathcal{M}$ is the tangent space at x_k of $\mathcal{M}$, $\varphi_{x_k} : T_{x_k}\mathcal{M} \to \mathcal{M}$ is a retraction map, α_k a real constant scalar and $p_k \in T_{x_k}\mathcal{M}$ a search direction. Retraction maps were introduced in [33] and are particularly well-suited to describe numerical methods on manifolds [2, 7, 12, 1]. The definition of retraction will be recalled in the next section. Gradient and Newton methods on Lie groups can be included in the general format (2). We will discuss classical gradient methods on manifolds, and rational mechanics type methods previously studied in [16].

We also propose a new class of univariate descent (UVD) methods. The advantages of this approach are that the optimization problem on the manifold $\mathcal{M}$ is broken down into one-dimensional optimization problems, and that each optimization step by itself involves less burdensome computation effort than a multivariate optimization step.

The idea for these methods has been inspired by previous work on the approximation of the matrix exponential [5], [6], and related methods for the numerical approximation of differential equations on Lie groups and homogeneous manifolds [30].

As a useful case-study, we shall consider univariate descent methods on the compact Stiefel manifold.

Similar, but different approaches related to the UVD method presented here, have been previously considered in [25] and [36]. In these articles the classical Jacobi method for the

diagonalization of matrices is generalized and analyzed in a Lie algebraic setting. These approaches do not include the case of the symmetric eiganvalue problem, which can be handled instead with the framework of the present paper.

The different descent methods are compared in the numerical experiments.

We will also discuss the application of the descent methods to a variety of problems in statistical signal processing, giving a brief introduction to principal component analysis and independent component analysis.

2 Descent methods based on retractions

2.1 Retraction maps on homogeneous manifolds

A retraction is a smooth map from the tangent bundle to the manifold, $\varphi : T\mathcal{M} \to \mathcal{M}$. Denote by φ_x its restriction to the fiber $T_x\mathcal{M}$ (the tangent space at x), following [33] we assume the following conditions are satisfied:

- The map φ_x is defined in some open ball $B(0, r_x) \subset T_x\mathcal{M}$ of radius r_x about $0 \in T_x\mathcal{M}$;
- The equality $\varphi_x(v) = x$ holds if and only if $v = 0 \in T_x\mathcal{M}$;
- The equality $\varphi'_x|_0 = \mathrm{Id}_{T_x\mathcal{M}}$ holds.

Here, map $\varphi'_x : TT_x\mathcal{M} \to T\mathcal{M}$ is the tangent of the map φ_x. We may identify $TT_x\mathcal{M}$ with $T_x\mathcal{M} \times T_x\mathcal{M}$, and for each pair (u, v) we may fix the first factor, and get the map $\varphi'_x|_u : T_x\mathcal{M} \to T_{\bar{x}}\mathcal{M}$ where $\bar{x} := \varphi_x(u)$.

The case of optimization on homogeneous manifolds is of particular interest. Let $\mathcal{M}$ be a homogeneous manifold acted upon transitively by a Lie group G, with group action $\Lambda(g, x) = \Lambda_x(g)$. Let us also define:

$$\psi : \mathfrak{g} \to G, \ \rho_x := (\Lambda_x \circ \psi)'\big|_0 , \tag{3}$$

where $\mathfrak{g}$ denotes the Lie algebra associated to the Lie group G. One can prove that if it exists a linear map $\mathrm{a}_x : T_x\mathcal{M} \to \mathfrak{g}$ such that $\rho_x \circ \mathrm{a}_x = \mathrm{Id}_{T_x\mathcal{M}}$ then φ_x given by:

$$\varphi_x(v) := (\Lambda_x \circ \psi \circ \mathrm{a}_x)(v), \tag{4}$$

is a retraction, [7]. Therefore, we can construct retractions using any coordinate map from the Lie algebra $\mathfrak{g}$ to the group G, as illustrated in the Figure 1. Also, we may define the image of the tangent space under the map a_x:

$$\mathfrak{m}_x := \mathrm{a}_x(T_x\mathcal{M}) \subset \mathfrak{g}. \tag{5}$$

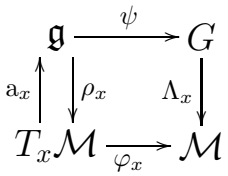


Figure 1: Construction of a retraction using a coordinate map from the algebra $\mathfrak{g}$.

The set $\mathfrak{m}_x$ is a linear subspace of the Lie algebra $\mathfrak{g}$. We will consider in the sequel different instances of the method (2) on homogeneous manifolds by either choosing search directions $p_k = \rho_{x_k}(v)$ with $v \in \mathfrak{g}$ or taking directly $p_k \in T_{x_k}\mathcal{M}$. The use of Lie group actions in the context of integration methods for differential equations on homogeneous manifolds has been introduced in [28], [29].

2.2 Univariate descent methods (UVD)

Let us consider the case of a homogeneous manifold acted upon by a group of dimension d and a basis for the Lie algebra $\mathfrak{g}$ given by E_1, E_2, ..., E_d. Assume, for the sake of simplicity, that we are solving a minimization problem on $\mathcal{M}$. The homogeneous manifold structure guarantees the transitivity of the Lie group action on $\mathcal{M}$. Assume further that the minimization problem has a solution on $\mathcal{M}$ and the minimum is attained in $x^* \in \mathcal{M}$. Given $x_0 \in \mathcal{M}$, by the transitivity of Λ we know that it exists $g \in G$ such that:

$$x^* = \Lambda(g, x_0).$$

If G is connected then any element in G can be written as a product of exponentials, namely:

$$g = \exp(v_1) \cdots \exp(v_m), \quad v_1, \ldots, v_m \in \mathfrak{g}$$

and therefore we have:

$$\begin{array}{rcl} \{v_1, \ldots, v_m\} & = & \underset{\substack{m \in \mathbf{N} \\ w_1, \ldots, w_m \in \mathfrak{g}}}{\operatorname{argmin}} \quad \phi(\Lambda_{x_0}(\exp(w_1) \cdots \exp(w_m))), \\ x^* & = & \Lambda_{x_0}(\exp(v_1) \cdots \exp(v_m)). \end{array} \tag{6}$$

If each of the exponentials $\exp(w_j)$ is in the neighborhood of the identity in G, we have:

$$\exp(w_j) = \exp(\gamma_d^j E_d) \cdots \exp(\gamma_1^j E_1),$$

with $\gamma_1^j, \ldots \gamma_d^j$, for $j = 1, \ldots, m$, being appropriate scalars, and in our examples, real numbers. This can be assumed for example when x_0 is sufficiently close to x^*. By substitution in equation (6) this leads to a reformulation of the minimization problem on $\mathcal{M}$, as minimization problem in $\mathbf{R}^{d \cdot m}$.

With this in mind, we discuss here iterative optimization methods searching for x^* by minimizing separately in turn in the direction of each basis element in $\mathfrak{g}$. By using the exponential as coordinate map and the Lie group action Λ on $\mathcal{M}$, we can obtain an algorithm of the form (2) as follows:

$$x_{k+1} = \varphi_{x_k}(-\alpha_k p_k) = \Lambda_{x_k}(\psi(-\alpha_k E_i)), \tag{7}$$

where the step size schedule α_k may be chosen according to the rule:

$$\alpha_k = \underset{\alpha \in \mathbf{R}}{\operatorname{argmin}}\, \phi(\Lambda_{x_k}(\psi(-\alpha E_i)), \tag{8}$$

with index $i \in \{1, \ldots, d\}$. The search direction is here $p_k = \rho_{x_k}(E_i)$ for $i \in \{1, \ldots, d\}$.

Assuming an ordering of the basis elements is established, we can perform a cyclic selection of the index i by taking $i = (k \bmod d) + 1$. Alternatively, we could adopt a random selection of the index i by randomly picking i within the set $\{1, \ldots, d\}$ with equal probability.

As we will see in concrete examples, it might be advantageous to chose special orderings of the basis elements in $\mathfrak{g}$. In fact, a possible convenient ordering is obtained by partitioning the basis in subsets of generators of commutative sub-algebras of maximal dimension. A complete discussion of the choice of an optimal ordering goes beyond the scope of the present article and will be addressed in future work.

Summarizing, a possible optimization procedure might thus be cast in the following framework:

set $x_0 \in \mathcal{M}$

for $k = 0,\ 1,\ 2,\ \ldots$ do

  select E_i cyclically or randomly

  find $\alpha_k = \underset{\alpha \in \mathbf{R}}{\operatorname{argmin}}\, \phi(\Lambda_{x_k}(\psi(-\alpha E_i)))$

  set $x_{k+1} = \Lambda_{x_k}(\psi(-\alpha_k E_i))$

end for

The advantages of this approach are:

- The optimization problem on $\mathcal{M}$ is broken down in (at least d) optimization problems in $\mathbf{R}$ which are allegedly easier to solve;
- Each step in the loop involves the computation of a term $\psi(-\alpha_k E_i)$ which is allegedly easy to effect.

Example 2.1 *In order to illustrate some of the features of the UVD method, we consider the minimization on a two-dimensional torus* $T^2 = S^1 \times S^1$.

Here we denote with S^1 the circle, i.e.

$$S^1 = \{g \cdot \mathbf{e}_1 \mid \mathbf{R}^2,\ g(\alpha) \in SO(2)\}, \quad g(\alpha) = \exp(\alpha E),\ \ E = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix},\ \ 0 \leq \alpha < 2\pi,$$

where $\mathbf{e}_1$ is the first canonical vector and $SO(2)$ is the commutative Lie group of planar rotations. Any element in T^2 is of the form

$$x_0 \in T^2, \quad x_0 = (g(\theta)\mathbf{e}_1, g(\varphi)\mathbf{e}_1), \quad g(\theta),\ g(\varphi) \in SO(2).$$

The Lie group acting on T^2 is $SO(2) \times SO(2)$, its corresponding Lie algebra is $\mathfrak{so}(2) \times \mathfrak{so}(2)$, has dimension $d = 2$ and basis $\{(E, O), (O, E)\}$, where O is the zero element in $\mathfrak{so}(2)$, and E is the basis element in $\mathfrak{so}(2)$. A parameterization of T^2 in $\mathbf{R}^3$ in angular coordinates, can be obtained applying the following mapping

$$(g(\theta)\mathbf{e}_1, g(\varphi)\mathbf{e}_1) \rightarrow \begin{cases} x &= (1 + \mathbf{e}_1^T g(\theta)\mathbf{e}_1) \cdot \mathbf{e}_1^T g(\varphi)\mathbf{e}_1 &= (1 + \cos(\theta))\cos(\varphi), \\ y &= (1 + \mathbf{e}_1^T g(\theta)\mathbf{e}_1) \cdot \mathbf{e}_2^T g(\varphi)\mathbf{e}_1 &= (1 + \cos(\theta))\sin(\varphi), \\ z &= \mathbf{e}_2^T g(\theta)\mathbf{e}_1 &= \sin(\theta), \end{cases}$$

with $0 \leq \theta, \varphi < 2\pi$. This is equivalent to the composition of two planar rotations and one translation in $\mathbf{R}^3$.

Assume the cost function we want to minimize is simply the distance from a fixed plane in $\mathbf{R}^3$, say $y = 8$. This gives

$$\phi(g(\theta)\mathbf{e}_1, g(\varphi)\mathbf{e}_1) = |(1 + \cos(\theta))\sin(\varphi) - 8|, \tag{9}$$

and the minimum is attained in $\theta = 0$ and $\varphi = \pi/2$. Denote with I the identity element in $SO(2)$. Using the chosen coordinate parameterization two iterations of the univariate descent method, taking x_0 as starting value, and minimizing first in the θ coordinate, are given as follows

$$\begin{aligned} x_1 &= \Lambda_{x_0}((\exp(\alpha_1 E), I)) &= (\exp(\alpha_1 E)g(\theta_0)\mathbf{e}_1, g(\varphi_0)\mathbf{e}_1), \quad \theta_1 = \theta_0 + \alpha_1, \\ x_2 &= \Lambda_{x_1}((I, \exp(\alpha_2 E))) &= (g(\theta_1)\mathbf{e}_1, \exp(\alpha_2 E)g(\varphi_0)\mathbf{e}_1), \quad \varphi_1 = \varphi_0 + \alpha_2, \end{aligned}$$

with α_1 and α_2 the optimal rotation angles, solution of the following univariate minimization problems,

$$\begin{aligned} \alpha_1 &= \operatorname*{argmin}_{\alpha \in \mathbf{R}} \phi((\exp(\alpha E)g(\theta_0)\mathbf{e}_1, g(\varphi_0)\mathbf{e}_1)), \\ \alpha_2 &= \operatorname*{argmin}_{\alpha \in \mathbf{R}} \phi((g(\theta_1)\mathbf{e}_1, \exp(\alpha E)g(\varphi_0)\mathbf{e}_1)). \end{aligned}$$

The results of applying the method with different starting values are reported in Table 1. In this simple example the univariate descent method reaches the global minimum in two or three iterations, except for pathological starting values such as $\sin(\varphi_0) = 0$ and $\cos(\theta_0) = 0$. □

	start optimizing with θ		start optimizing with φ	
Iteration	$\sin(\varphi_0) > 0$	$-1 \leq \sin(\varphi_0) < 0$	$\cos(\theta_0) > 0$	$-1 < \cos(\theta_0) \leq 0$
1	$\theta_1 = 0$ $\varphi_1 = \varphi_0$	$\theta_1 = \pi/2$ $\varphi_1 = \varphi_0$	$\theta_1 = \theta_0$ $\varphi_1 = \pi/2$	$\theta_1 = \theta_0$ $\varphi_1 = \pi/2$
2	$\theta_2 = 0$ $\varphi_2 = \pi/2$	$\theta_2 = \pi/2$ $\varphi_2 = \pi/2$	$\theta_2 = 0$ $\varphi_2 = \pi/2$	$\theta_2 = 0$ $\varphi_2 = \pi/2$
3		$\theta_3 = 0$ $\varphi_3 = \pi/2$		

Table 1: UVD method for optimization on the torus.

The one-dimensional optimization problems to be solved at each step of the algorithm depend on the choice of action and the type of coordinates. In the present paper, we will always consider ψ to be the exponential map, and we will chose the basis E_i such that $\exp(-\alpha_k E_i)$ is easy to compute.

We list here some possible choices for the Lie group action. An obvious choice is to use the transitive Lie group action $\Lambda : G \times \mathcal{M} \to \mathcal{M}$ inducing the structure of homogeneous space on $\mathcal{M}$, however other choices are possible. Let us consider the inner automorphism of G,

$$\mathrm{I}_{\tilde{g}} : G \to G \qquad \mathrm{I}_{\tilde{g}}(g) := \tilde{g} \cdot g \cdot \tilde{g}^{-1}.$$

By composition with Λ, we obtain another Lie group action on $\mathcal{M}$, $\hat{\Lambda} : G \times \mathcal{M} \to \mathcal{M}$, such that:

$$\hat{\Lambda}(g, q) := \Lambda(\mathrm{I}_{\tilde{g}}(g), q),$$

where $\tilde{g}$ is a fixed element in G. For $\tilde{g} = I$ (with I being the identity element in the Lie group G) we obtain $\hat{\Lambda} = \Lambda$.

Example 2.2 *As a useful case-study, let us consider univariate descent methods on the Stiefel manifold.*

The compact Stiefel manifold can be represented by the set of $n \times p$ matrices with orthonormal columns:

$$\mathcal{M} = \mathrm{St}(n, p) := \{X \in \mathbf{R}^{n \times p} | X^T X = I_p\},$$

and has a homogeneous manifold structure induced by the action of the real orthogonal group of $G = O(n)$. In this case $\Lambda : O(n) \times \mathrm{St}(n, p) \to \mathrm{St}(n, p)$ is given by simple matrix-matrix multiplication, namely:

$$\Lambda(Q, X) = Q \cdot X, \quad Q \in O(n), \quad X \in \mathrm{St}(n, p).$$

Another possible choice for the action is the following. Let us consider $v_0 \in \mathbf{R}^n$, such that $v_0^T v_0 = 1$ a fixed vector, and the Householder transformation, $P_0 = I - 2v_0 v_0^T \in O(n)$, I the $n \times n$ identity matrix, and $P_0^2 = I$. We can take:

$$\hat{\Lambda}(Q, X) = \mathrm{I}_{P_0}(Q) \cdot X = P_0 Q P_0 \cdot X.$$

The dimension of the acting-group is $d = n(n-1)/2$. Let us choose as a basis of the algebra $\mathfrak{g} = \mathfrak{so}(n)$ the set of matrices:

$$\mathbf{e}_1\mathbf{e}_2^T - \mathbf{e}_2\mathbf{e}_1^T,\ \mathbf{e}_1\mathbf{e}_3^T - \mathbf{e}_3\mathbf{e}_1^T,\ \ldots,\ \mathbf{e}_{n-1}\mathbf{e}_n^T - \mathbf{e}_n\mathbf{e}_{n-1}^T, \tag{10}$$

where the $\mathbf{e}_i$'s denote the canonical basis-vectors of $\mathbf{R}^n$. As coordinate map, we select the exponential map $\psi := \exp$. Then, if we use the action $\hat{\Lambda}$, the optimization algorithm (15) reads:

$$x_{k+1} = P_0 \exp(-\alpha_k(\mathbf{e}_i\mathbf{e}_j^T - \mathbf{e}_j\mathbf{e}_i^T))P_0 \cdot x_k. \tag{11}$$

In this case, the solution of the univariate optimization problem:

$$\alpha_k = \underset{\alpha \in \mathbf{R}}{\operatorname{argmin}}\, \phi(P_0 \exp(-\alpha(\mathbf{e}_i\mathbf{e}_j^T - \mathbf{e}_j\mathbf{e}_i^T))P_0 \cdot x_k), \tag{12}$$

as well as the computation of the retraction are very simple due to the choice of basis. By setting $P_0 = I$ we the action simplifies to the case $\hat{\Lambda} = \Lambda$.

We will give details of how to solve the univariate optimization problems in the following sections.

For the purpose of random basis elements selection, it is worth to explicitly write down the relationship among the basis vectors:

$$E_{(i-1)n-i(i+1)/2+j} = \mathbf{e}_i\mathbf{e}_j^T - \mathbf{e}_j\mathbf{e}_i^T, \tag{13}$$

where it is easy to verify that $(i-1)n - i(i+1)/2 + j \in \{1, \ldots, d\}$. Therefore, a random basis matrix may be picked up uniformly from the set of basis matrices by randomly picking an index $r \in \{1, \ldots, d\}$ with equal probability for each value, by finding the largest integer $i \in \{1, \ldots, n-1\}$ such that $(i-1)(n - 1/2 - i/2) \leq r$ and by setting $j = r - (i-1)(n - 1/2 - i/2) + i$.

In order to achieve a partition of the basis elements in subsets of commuting elements we can proceed as follows. Consider the partition of the basis of $\mathfrak{so}(n)$ into the subsets corresponding to the set of indices:

$$S_l = \{(i, i+l) \quad | \quad i = 1, \ldots, n-l\}, \quad l = 1, \ldots, n-1.$$

Subdivide each of the sets S_l further into the two subsets S_l^e and S_l^o such that $S_l = S_l^e \cup S_l^o$ and $S_l^e \cap S_l^o = \emptyset$, and each consisting of indices corresponding to commuting basis elements, by taking:

$$S_l^e = \{(i, i+l) | i = 2k \cdot l + 1, \ldots, (2k+1) \cdot l, \quad k = 0, \ldots, m_1\} \cup R_l^e,$$
$$S_l^o = \{(i, i+l) | i = (2k+1) \cdot l + 1, \ldots, (2k+2) \cdot l, \quad k = 0, \ldots, m_2\} \cup R_l^o,$$

with either $n-l=(2m_1+2)\cdot l+r_1, \quad r_1<l$, $R_l^e=\emptyset$ and:

$$R_l^o=\{(i,i+l)|i=(2m_1+2)\cdot l+1,\ldots,n-l\},$$

or $n-l=(2m_2+2)\cdot l+r_2, \quad r_2<l$, $R_l^o=\emptyset$ and:

$$R_{l+1}^e=\{(i,i+l)|i=(2m_1+3)\cdot l+1,\ldots,n-l\}.$$

The above partition of the basis elements will be used in the numerical experiments. □

2.3 Gradient-based methods (GRAD)

Let $\mathcal{M}$ be a Riemannian manifold with metric $\langle\cdot,\cdot\rangle$ and $\phi:\mathcal{M}\to\mathbf{R}$ be a smooth function. The optimization method based on gradient flow (written for the minimization problem only, for the sake of easy reading) consists in setting up the differential equation on the manifold:

$$\dot{x}(t)=-\operatorname{grad}\phi\big(x(t)\big), \tag{14}$$

with appropriate initial condition $x(0)=x_0\in\mathcal{M}$. The equilibria of equation (14) are the critical points of the function ϕ. For a complete overview see for example [23] and references therein. In the above equation, the symbol $\operatorname{grad}\phi$ denotes the Riemannian gradient of the function ϕ with respect to the chosen metric. Namely, $\operatorname{grad}\phi(x)\in T_x\mathcal{M}$ and $\phi'|_x(v)=\langle\operatorname{grad}\phi(x),v\rangle$ for all $v\in T_x\mathcal{M}$. The solution of the above differential equation on $\mathcal{M}$ may be locally expressed in terms of a curve on the tangent space $T_{x_0}\mathcal{M}$ using a retraction and one has:

$$x(t)=\varphi_{x_0}(\sigma(t)).$$

By differentiating with respect to t one can find a differential equation for σ in $T_{x_0}\mathcal{M}$, which can be integrated with a forward-Euler time-stepping. The retraction is used then for mapping the result from the tangent space to the manifold, see [8] for further details. The resulting iterative method fits the format (2):

$$x_{k+1}=\varphi_{x_k}(-\alpha_k p_k)=\Lambda_{x_k}(\exp(\alpha_k a_{x_k}(p_k))),\ k=0,\ 1,\ \ldots. \tag{15}$$

where α_k is the time step of integration. Recall that a_{x_k} is a linear map $\mathrm{a}_{x_k}:T_{x_k}\mathcal{M}\to\mathfrak{g}$ such that $\rho_{x_k}\circ\mathrm{a}_{x_k}=\mathrm{Id}_{T_{x_k}\mathcal{M}}$, with $\rho_{x_k}:=(\Lambda_{x_k}\circ\exp)'|_0$.

The gradient-search direction is $p_k=\operatorname{grad}\phi(x_k)$ and we assume an appropriate starting point x_0 is being selected. The easiest choice for α_k is to take $\alpha_k=h$, $\forall k$, with h small enough positive real value. However also in this case α_k can be selected by solving the following univariate optimization problem:

$$\alpha_k=\operatorname*{argmin}_{\alpha\in\mathbf{R}}\phi(\varphi_{x_k}(-\alpha p_k)). \tag{16}$$

The solution of such univariate optimization problems has in general a higher computational complexity compared to (8).

In figure 2 we plot $-\text{grad}\phi$, the negative gradient vector field for the cost function of example 2.1. The Riemannian metric we used is,

$$\langle(\alpha_1 E\mathbf{e}_1, \beta_1 E\mathbf{e}_1), (\alpha_2 E\mathbf{e}_1, \beta_2 E\mathbf{e}_1))\rangle = \alpha_1\alpha_2 + \beta_1\beta_2,$$

and $(\alpha_1 E\mathbf{e}_1, \beta_1 E\mathbf{e}_1) \in T_{(\mathbf{e}_1,\mathbf{e}_1)}T^2$. At the point $p_0 = (g(\theta_0)\mathbf{e}_1, g(\varphi_0)\mathbf{e}_1) \in T^2 = S^1 \times S^1$ the gradient vector field can be represented by:

$$(\gamma \cdot E \cdot g(\theta_0)\mathbf{e}_1, \delta \cdot E \cdot g(\varphi_0)\mathbf{e}_1),$$

where γ and δ are real values given by,

$$\gamma = -C \cdot \sin(\theta_0)\sin(\varphi_0), \quad \delta = C \cdot (1 + \cos(\theta_0))\cos(\varphi_0),$$

and

$$C = 2((1 + \cos(\theta_0)\sin(\varphi_0) - 8).$$

The vector field $-\text{grad}\phi$ points towards the two minima of the cost function, (9). Whether the gradient method is converging to the local or global minimum depends on the initial value and on the step size schedule. For example if we fix $h = 0.01$ and consider $(\theta_0 = \pi/4 + 0.1, \varphi_0 = \frac{3}{2}\pi)$ as starting value the gradient method will converge to the local minimum $(\theta = \pi, \varphi = \frac{3}{2}\pi)$, but if we take $(\theta_0 = \pi/4 + 0.1, \varphi_0 = \frac{3}{2}\pi + 0.2)$, with the same step size, the iteration will converge to the global minimum $(\theta = 0, \varphi = \frac{1}{2}\pi)$.

Experiments implementing the variable step size selection according to (16) on this example, show no improvement of the convergence of the gradient method unless the univariate minimization problem (16) is solved very accurately. In this example the univariate descent method performs better then the gradient method, we conjecture that this is due to the commutativity of $SO(2)$.

For the case of the Stiefel manifold, example 2.2, we consider the Reimannian metric $\langle V, W\rangle := \frac{1}{2}\text{tr}(V^T W)$ with $V, W \in T_X\text{St}(n,p)$, and the gradient vector field is:

$$\text{grad}\,\phi = \left(\left(\frac{\partial\phi}{\partial X}\right)X^T - X\left(\frac{\partial\phi}{\partial X}\right)^T\right)\cdot X, \tag{17}$$

Therefore, the optimization algorithm in the tangent space formulation reads:

$$\begin{cases} G_k &= \left(\frac{\partial\phi}{\partial X}\right)\Big|_{X_k}, \\ A_k &= a_{X_k}(-\text{grad}\phi(X_k)) \;=\; -(G_k X_k^T - X_k G_k^T) \\ X_{k+1} &= \exp(hA_k)X_k, \\ k &= 1,\ 2,\ 3,\ \ldots, \end{cases} \tag{18}$$

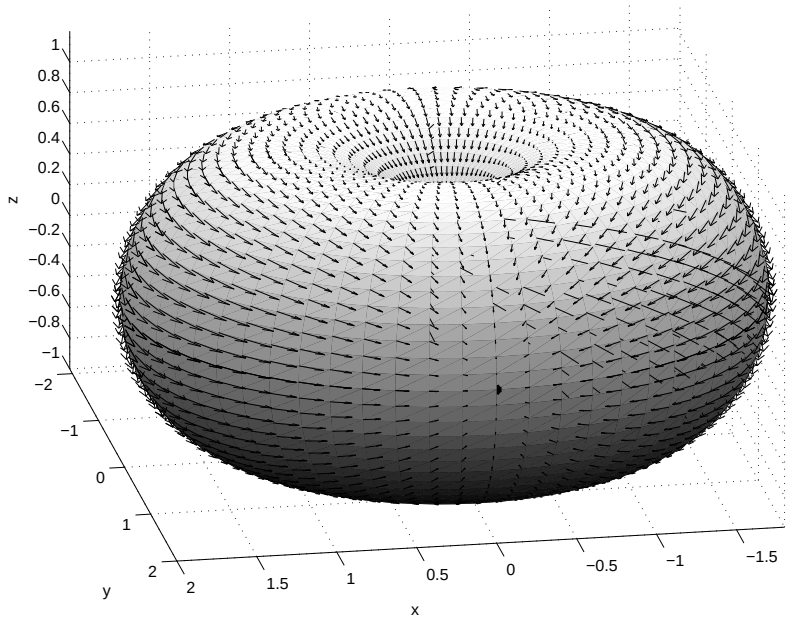


Figure 2: The gradient vector field of the cost function $\phi((g(\theta)\mathbf{e}_1, g(\varphi)\mathbf{e}_1) = (1+\mathbf{e}_1^T g(\theta)\mathbf{e}_1)\cdot \mathbf{e}_2^T g(\varphi)\mathbf{e}_1 - 8)^2$ on the torus. The vector field points towards the two minima, the global minimum is marked with a black spot in the middle of the picture.

with initial guess $X_0 \in \mathrm{St}(n,p)$ arbitrarily selected (a typical choice would be $X_0 = I_{n\times p}$) and step size $h \in \mathbf{R}$ appropriately selected.

It is worth noting that, in the above equations, the action of the orthogonal group on the Stiefel manifold may be conveniently computed by:

$$X_{k+1} = [X_k, X_k^\perp]\exp\left(-h\begin{bmatrix} C_k - C_k^T & -R_k^T \\ R_k & 0_p \end{bmatrix}\right)[X_k, X_k^\perp]^T X_k, \tag{19}$$

where matrices $C, R \in \mathbf{R}^{p\times p}$ are defined $C_k := X_k^T G_k$ and $G_k - X_k C_k =: X_k^\perp R_k$. This formula requires computing an orthogonal complement $X_k^\perp \in \mathrm{St}(n,p)$ of the matrix X_k (namely, $[X_k\ X_k^\perp] \in \mathrm{St}(n,2p)$). In this way, it is necessary to exponentiate skew-symmetric matrices of dimension $2p \times 2p$ instead of matrices of size $n \times n$. The computational cost of the exponentiation amounts to $9np^2 + np + \mathcal{O}(p^3)$ flops.

2.4 Rational-mechanics-type methods (MEC)

For the special case of optimization on the Stiefel manifold or on the orthogonal group a second class of decent algorithms which can be cast in the presented framework arises from the numerical approximation of rational-mechanics-type equations.

Let us consider $\mathcal{S}^* = \{[m_i, \pm w_i]\}$ to be a rigid system of $2p$ masses m_i with positions $w_i \in \mathbf{R}^n$, $i = 1,\ 2,\ \ldots,\ p$. The masses rotate in a viscous liquid around a mass-center. In the hypothesis that all masses are equal and unitary, namely $m_i = 1$, the equations

describing the dynamics of such system are [15, 16]:

$$\begin{cases} \dot{W} &= HW,\ P = -\mu HW, \\ \dot{H} &= \frac{1}{4}\left[(F+P)W^T - W(F+P)^T\right]. \end{cases} \tag{20}$$

The matrix W lays on the orthogonal group $O(n)$ (for $p = n$) or on the Stiefel manifold $\mathrm{St}(n,p)$ (for $p < n$) and denotes the set of masses positions, $F \in \mathbf{R}^{n\times p}$ denotes the set of active forces that make the system move, $H \in \mathfrak{so}(n)$ denotes the angular velocity tensor. The constant $\mu > 0$ denotes a viscosity parameter and $P \in \mathbf{R}^{n\times n}$ denotes the viscosity resistance. The active forces are generated by a potential energy field, namely:

$$F := -\frac{\partial U}{\partial W}, \tag{21}$$

with $U = U(W)$ denoting a potential energy function. The equilibria of the mechanical systems $\mathcal{S}^*$ coincide to the local minima of the potential U. In the optimization context, the potential energy function can be assumed as a cost function to be minimized or as an objective function to be maximized. In this case, the mechanical system state $W(t)$ asymptotically approaches the solution of the optimization problem as the system tends to *minimize* its potential energy function.

When $n \gg p$, equations (20) may be reformulated in a more convenient framework by using the tangent space $T_W\mathrm{St}(n,p)$, as shown in [4]. The tangent-space formulation of the equations is profitable if such differential equations are integrated numerically in order to gain numerical efficiency. It has been observed (see, e.g., [17, 34]) that certain classes of signal processing problems involve rectangular matrices whose p/n ratio is quite low. An integration method that takes into account such observation might achieve moderate computational complexity. In particular, equations (20) may be rewritten as:

$$\dot{W} = V,\ \dot{V} = g(V,W), \tag{22}$$

where $V \in T_W\mathrm{St}(n,p)$ is a system velocity vector and $g : T_W\mathrm{St}(n,p) \times \mathrm{St}(n,p) \to T_W\mathrm{St}(n,p)$, defined as:

$$\begin{cases} V &:= (GW^T - WG^T)W, \\ g(V,W) &:= (LW^T - WL^T)W + (GW^T - WG^T)V, \\ G &:= V - W(W^TV/2 + S), \\ L &:= F - \mu V. \end{cases} \tag{23}$$

In the above equations, the matrix $S \in \mathbf{R}^{p\times p}$ may be arbitrarily selected, for example taking $S = 0_p$. The matrix $G \in \mathbf{R}^{n\times p}$. The matrix $L \in \mathbf{R}^{n\times p}$ represents the set of effective forces, namely the active forces deflated from the viscous braking forces.

As $H(t)$ and $V(t)$ evolve on linear spaces, the differential equations describing their dynamics may be solved by a classical forward Euler time-stepping. Conversely, the dynamics

of the matrix W is approximated with a retraction based Euler method, as was done for the case of the gradient method.

The MEC optimization algorithm, written in the original Lie algebra formulation, reads:

$$\begin{cases} H_{k+1} &= H_k + h((F_k + P_k)W_k^T - W_k(F_k + P_k)^T), \\ P_k &= -\mu H_k W_k, \\ F_k &= F(W_k), \\ W_{k+1} &= \exp(hH_k)W_k, \\ k &= 1,\ 2,\ 3,\ \ldots, \end{cases} \tag{24}$$

with $W_0 \in \mathrm{St}(n,p)$ and $H_0 \in \mathfrak{so}(n)$ arbitrarily selected (a typical choice would be $W_0 = I_{n\times p}$ and $H_0 = 0_n$). Here, search direction H_k is selected in the Lie algebra as an approximation of the differential equation for H. As a consequence the computation of the action $\exp(hH_k)W_k$ involves the exponentiation of $n \times n$ matrices and requires at least $\mathcal{O}(n^2p)$ floating-point operations. In the above equations, the parameter $h \in \mathbf{R}$ denotes a constant step-size. Instead, the MEC optimization algorithm in the tangent space formulation reads:

$$\begin{cases} V_{k+1} &= V_k + hg(V_k, W_k), \\ G_k &= V_k - \frac{1}{2}W_k(W_k^T V_k), \\ W_{k+1} &= \exp(h(G_k W_k^T - W_k G_k^T))W_k, \\ k &= 1,\ 2,\ 3,\ \ldots, \end{cases} \tag{25}$$

with $W_0 \in \mathrm{St}(n,p)$ and $V_0 \in T_{W_0}\mathrm{St}(n,p)$ arbitrarily selected (a typical choice would be $W_0 = I_{n\times p}$ and $V_0 = 0_{n\times p}$). In the above equations, the parameter $h \in \mathbf{R}$ denotes again a constant step-size.

In this case, as in the previous section, we can make use of the formula (19) for the computation of the action of the orthogonal group on the Stiefel manifold. The total computational cost of one step of the algorithm amounts to $21np^2 + 6np + \mathcal{O}(p^3)$ flops.

3 Signal processing applications

The general optimization problem (1) arises in several applied fields ranging from engineering to applied physics and neurophysiology. Some specific exemplary applied topics that can be addressed under the above general settings are principal component/subspace analysis, eigenvalue and generalized eigenvalue problems, optimal linear compression, noise reduction and signal representation [11, 14, 35, 37]; neural independent component analysis, blind source separation and blind signal processing [15, 17]; minimal linear system realization from noise-injection measured data and invariant subspace computation [13, 26], linear programming and sequential quadratic programming [3, 13], analysis of natural three-dimensional movement [38], synthesis of digital filters [21], physics of bulk materials [13] and reduced-rank Wiener filtering [34].

An interesting example of a problem that can be tackled via statistical signal processing is the *cocktail-party problem*. Let us suppose two signals $x_1(t)$ and $x_2(t)$ were recorded from two different positions in a room where two speakers are talking in. Each recorded signal is a linear mixture of the voices of two speakers $s_1(t)$ and $s_2(t)$, namely:

$$x_1(t) = a_{1,1}s_1(t) + a_{1,2}s_2(t),$$
$$x_2(t) = a_{2,1}s_1(t) + a_{2,2}s_2(t),$$

where the four coefficients $a_{i,j} \in \mathbf{R}$ denote the mixing proportions. The cocktail party problem consists in estimating signals $s_1(t)$ and $s_2(t)$ from the only knowledge of their mixtures $x_1(t)$ and $x_2(t)$. The main assumption on the source signals is that $s_1(t)$ and $s_2(t)$ are statistically independent. A known method to solve this problem is to invoke a statistical signal processing technique termed Independent Component Analysis (ICA). A possible way to interpret ICA as an optimization problem is the following: After mixing, the independent signals $s_1(t)$ and $s_2(t)$ result in signals $x_1(t)$ and $x_2(t)$ that are statistically dependent, therefore, a way of estimating the independent sources is to minimize a criterion function that is able to tell how much dependent two (or more) signals are. As it can be shown, it is always possible to hypothesize that the coefficients $a_{i,j}$ possess some constraints, such as that they form a 2×2 orthogonal matrix, therefore the estimation of independent components may be cast as an optimization problem of the kind (1), where the parameter manifold is the Lie group of real orthogonal matrices. The general ICA setting is explained in section 3.2.

3.1 Application to principal component analysis

Data reduction techniques are statistical signal processing methods that aim at providing efficient representations of data. A well-known data compression technique consists in mapping an high-dimensional data space into a lower dimensional representation space by means of a linear transformation. It requires the computation of the data covariance matrix and then the application of a numerical procedure to extract its eigenvalues and the corresponding eigenvectors. Compression is then obtained by the use of the only eigenvectors associated with the most significant eigenvalues as a new basis.

In particular, Principal Component Analysis (PCA) is a second-order adaptive statistical data processing technique that helps removing the second-order correlation among given random signals. Let us consider the stationary multivariate random process $x(t) \in \mathbf{R}^n$ and suppose its covariance matrix $A = E[(x - E[x])(x - E[x])]^T]$ exists bounded. If $A \in \mathbf{R}^{n \times n}$ is not diagonal, then the components of $x(t)$ are statistically correlated. Such second-order redundancy may be partially (or completely) removed by computing a linear operator $F \in \mathbf{R}^{n \times p}$ such that the new random signal defined by $y(t) := F^T(x(t) - E[x]) \in \mathbf{R}^p$ has uncorrelated components, with $p \leq n$ properly selected. Any covariance matrix is

symmetric (semi) positive-definite, therefore the operator F is known to be the matrix formed by the eigenvectors of the matrix A corresponding to its largest eigenvalues and $F \in \mathrm{St}(n,p)$ (see, e.g., the recent tutorials [11, 19]). The component signals of $y(t)$ are termed *principal components of the signal* $x(t)$, whose relevance is proportional to the corresponding eigenvalues $\sigma_i^2 = E[y_i^2]$ which are supposed here to be arranged in descending order ($\sigma_i^2 \geq \sigma_{i+1}^2$).

The data-stream $y(t)$, thus, represents a compressed version of the data stream $x(t)$. After the reduced-size data have been processed (i.e. stored, transmitted), they need to be recovered, that is, they need to be brought back to the original structure. However, the principal-component-based data-reduction technique is not lossless, thus only an approximation $\hat{x}(t) \in \mathbf{R}^n$ of the original data-stream may be recovered. As F is a tall-skinny orthogonal operator, an approximation of $x(t)$ is given by $\hat{x}(t) = Fy(t) + E[x]$: Such approximate data-stream minimizes the reconstruction error $E[||x - \hat{x}||_2^2]$, which equals $\sum_{i=n+1}^{p} \sigma_i^2$.

In the above equations, symbol $E[\cdot]$ denotes statistical expectation. For a scalar or a vector-valued random variable $x \in \mathbf{R}^n$ endowed with a probability density function $p_x : x \in \mathbf{R}^n \rightarrow p_x(x) \in \mathbf{R}$, the expectation of a function $\beta : \mathbf{R}^n \rightarrow \mathbf{R}$ is defined as:

$$E[\beta] := \int_{\mathbf{R}^n} \beta(x) p_x(x) d^n x. \tag{26}$$

Under the hypothesis that the signals whose expectation is to be computed are ergodic, the actual expectation (ensemble average) may be replaced by temporal-average on the basis of the available signals samples, namely:

$$E[\beta] \approx \frac{1}{T} \sum_{t=1}^{T} \beta(x(t)). \tag{27}$$

In the following, we shall consider the problem of PCA computation as tackled by the UVD as well as MEC optimization techniques.

3.1.1 Principal component analysis by the univariate descent method

In the context of univariate descent methods, the problem of principal component analysis reduces to the computation of eigenpairs of the symmetric positive-definite signal covariance matrix. To compute p eigenpairs of a $n \times n$ symmetric and positive-definite matrix $\tilde{A}$, we consider the maximization of the function:

$$\phi(X) = \frac{1}{2} \mathrm{trace}(X^T \tilde{A} X), \tag{28}$$

on the Stiefel manifold $\mathrm{St}(n,p) \ni X$. Let us get into the details of the application of the univariate descent algorithm (11) along with the univariate-optimization problem (12).

For the particular choice of basis in $\mathfrak{so}(n)$ the exponentials $\exp(\alpha(\mathbf{e}_i\mathbf{e}_j^T - \mathbf{e}_j\mathbf{e}_i^T))$, with $\alpha \in [0\ 2\pi]$, are Givens rotators, i.e. the matrices:

$$G_{ij}(\alpha) := I + [\mathbf{e}_i, \mathbf{e}_j] \begin{bmatrix} \cos(\alpha) - 1 & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) - 1 \end{bmatrix} [\mathbf{e}_i, \mathbf{e}_j]^T,$$

where I is the $n \times n$ identity matrix.

By denoting with $X_k \in \mathrm{St}(n,p)$ the sequence of putative solutions, the univariate-descent algorithm (11) becomes:

$$X_{k+1} = P_0 G_{i,j}(\alpha_k) P_0 \cdot X_k,\ k = 0,\ 1,\ 2,\ \ldots, \tag{29}$$

with X_0 arbitrarily selected in $\mathrm{St}(n,p)$ and $P_0 = I - 2\mathbf{v}_0\mathbf{v}_0^T$ and Householder transformation not depending on α. Each univariate-optimization problem (12) is modified accordingly so that we are looking for the appropriate angle α that maximizes the quantity $\phi(\alpha) := \phi(P_0 G_{ij} P_0(\alpha) Q)$ for a given $Q \in \mathrm{St}(n,p)$ and a selected Givens rotator $G_{ij}(\cdot)$. In the present setting, the criterion function to optimize $\phi(\alpha)$ is quadratic in the matrix $P_0 G_{ij}(\alpha) P_0 Q$ and thus it is expressible as a quadratic form in the quantities $\cos\alpha$ and $\sin\alpha$. Its optimum might thus be easily gotten numerically. In fact, in this case, it is easily found that:

$$\phi(\alpha) = a\sin^2\alpha + b\cos^2\alpha + c\sin\alpha + d\cos\alpha + e\sin\alpha\cos\alpha + f, \tag{30}$$

$$\phi'(\alpha) = (a-b)\sin(2\alpha) + e\cos(2\alpha) + c\cos\alpha - d\sin\alpha, \tag{31}$$

where a, b, c, d, e and f are coefficients depending on indices i and j and on the matrices $A = P_0\tilde{A}P_0$ and $R := P_0 Q Q^T P_0$. Such coefficients are defined as:

$$\begin{cases} a := \frac{1}{2}(A^{jj}R^{ii} - 2A^{ij}R^{ij} + A^{ii}R^{jj}), \\ b := \frac{1}{2}(A^{ii}R^{ii} + 2A^{ij}R^{ij} + A^{jj}R^{jj}), \\ c := (RA)^{ij} - (RA)^{ji} - A^{ij}(R^{ii} - R^{jj}) - R^{ij}(A^{jj} - A^{ii}), \\ d := (RA)^{ii} + (RA)^{jj} - A^{ii}R^{ii} - 2A^{ij}R^{ij} - A^{jj}R^{jj}, \\ e := A^{ij}(R^{ii} - R^{jj}) + R^{ij}(A^{jj} - A^{ii}), \\ f := \frac{1}{2}\mathrm{trace}(RA) - (RA)^{ii} - (RA)^{jj} + b. \end{cases} \tag{32}$$

(In the above equations, notation $A^{rs}(\alpha)$ denotes the $(r,s)^{\text{th}}$ entry of the matrix A.)

As it is readily seen, some terms in the expressions of the coefficients repeat, and only a few entries of matrices R and RA are actually used at each iteration, therefore it is not necessary to explicitly form the two matrix products.

The Householder matrix P_0 might be set equal to the identity or chosen in other appropriate ways. A possible choice is to take P_0 different from the identity in the first $n-1$ steps of the algorithm so to reduce $\tilde{A}$ to tridiagonal form, i.e. requiring after $n-1$ steps that

$$A = P_0^{(n-1)} \cdots P_0^{(1)} \tilde{A} P_0^{(1)} \cdots P_0^{(n-1)}$$

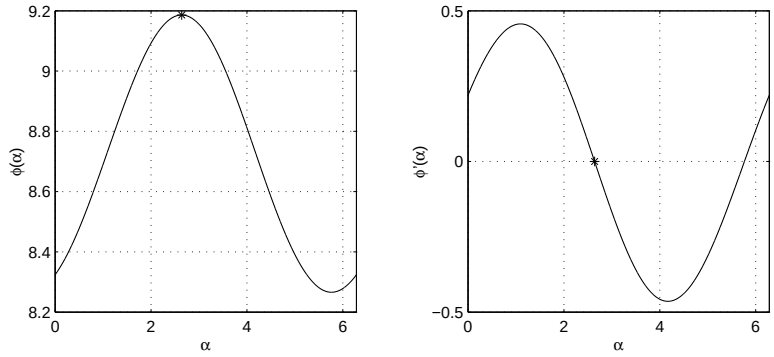


Figure 3: Shape of functions $\phi(\alpha)$ and $\phi'(\alpha)$ on a randomly generated matrix A. In this example: $n = 10$, $p = 3$ and the rotator $G_{24}(\alpha)$ was used. The matrix Q was also randomly generated in $\mathrm{St}(10, 3)$. The asterisk on the left-hand panel marks the approximate maximum, while the asterisk on the right-hand panel marks the approximate location of the corresponding zero.

is tridiagonal, see [22] for details. This operation introduces some extra computations in the algorithm, but has the advantage of leading to a simplification of the formulae (32), as well as improving the convergence of the method.

To look for the absolute maximum (or minimum) of the function $\phi(\alpha)$ we consider the following strategies:

- Find the zeros of $\phi'(\alpha)$ by transforming the problem into a complex polynomial equation of degree 4, via the substitutions $\sin(\alpha) = \frac{e^{i\alpha}-e^{-i\alpha}}{2i}$ and $\cos(\alpha) = \frac{e^{i\alpha}+e^{-i\alpha}}{2}$.

- Sample the function $\phi(\alpha)$ on equally-spaced values of the parameter α in the interval $[0\ 2\pi]$ and pick the value corresponding to the largest function value found. As the function to optimize is smooth, a good approximation of the actual maximum is expected to be reached.

- Sample the function $\phi'(\alpha)$ on equally-spaced values of the parameter α in the interval $[0\ 2\pi]$ and pick the values that are closer to 0. The found values need to be tested as they might correspond to local maxima/minima or saddle points, unless the function $\phi''(\alpha)$ is computed as well.

- The second method above, with zooming maximum search. This method consists in a loose partitioning of the interval $[0\ 2\pi]$ and in a fine repartitioning of sub-intervals around a point of interest, until no meaningful changes of ϕ are observed.

Figure 3 displays an exemplary shape of functions $\phi(\alpha)$ and $\phi'(\alpha)$ on a randomly generated matrix A. The maximum as well as the zero-crossing point where found numerically by using the second method above.

In order to assess the UVD algorithm for PCA, the matrix to be decomposed may be generated as $A = VDV^T$, where $V \in O(n)$ and $D \in \mathbf{R}^{n\times n}$ diagonal, with positive entries ordered in ascending order. In this way, a PCA $X \in \mathrm{St}(n,p)$ is such that $\phi^\star := \phi(X) = \sum_{i=1}^{p} D^{ii}$. Figure 4 displays a run on a randomly generated matrix A, both with cyclically and randomly selected algebra basis elements. In both cases, the iteration stops when the absolute mismatch $|\phi - \phi^\star|$ becomes smaller than 10^{-3} or when the number of iterations exceeds 500. As it is readily observed, in this simulation both

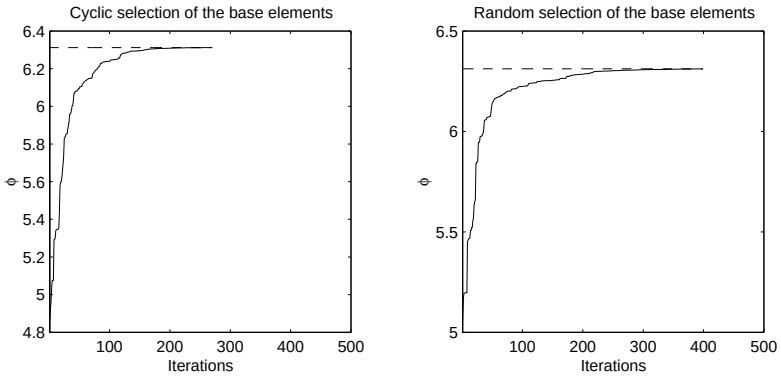


Figure 4: A run on a randomly generated matrix A. In this example: $n = 10$, $p = 3$. The matrix Q was randomly generated in $\mathrm{St}(10,3)$. The left-hand panel shows the behavior of the algorithm when the basis elements for the algebra are cyclically selected, while the right-hand panel shows the behavior of the algorithm when the basis elements for the algebra are randomly selected. Solid-line: Value of ϕ during iteration. Dashed-line: Value of $\phi^\star$. (Principal component analysis experiment.)

the UVD algorithm that makes use of cyclic basis elements selection and the UVD algorithm that makes use of random basis elements selection terminate iteration because the exit condition $|\phi - \phi^\star| < 10^{-3}$ is met. However, in the shown simulation, the first algorithm runs for 270 iterations while the second one runs through 399 iterations. While, for a given problem and with given initial conditions, the first algorithm runs on a fixed amount of iterations, the second one depends on random fluctuations in the algebra and thus it might run even faster or slower: Its velocity of convergence is thus unpredictable. Figure 5 displays the empirical distribution of the number of iterations to converge to the UVD with randomly selected algebra basis elements on 500 independent trials for the same eigenpair-computation problem. The empirical mean value is about 273 iterations with standard deviation 95. The mean convergence velocity of the UVD algorithm with randomly selected algebra basis elements, thus, equals the velocity of the UVD algorithm that makes use of cyclic basis elements selection. It should be noted, however, that in the 4.4% of the cases, the algorithm does not appear to converge within the maximum allotted

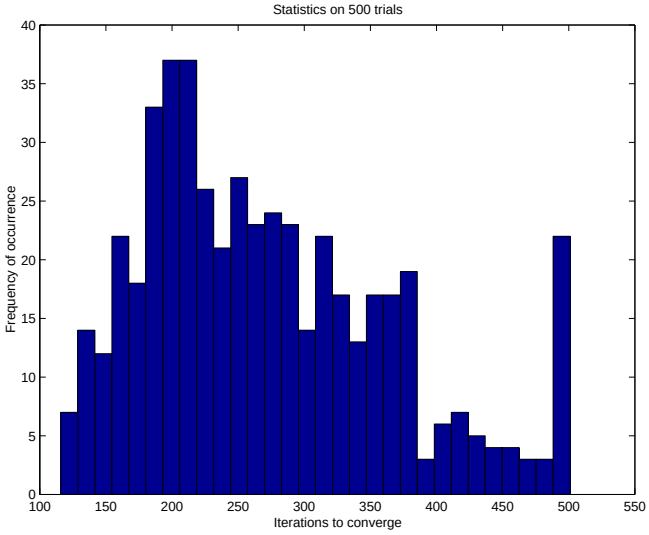


Figure 5: Empirical distribution of the number of iterations to converge of the UVD algorithm with randomly selected algebra basis elements. (Principal component analysis experiment.)

number of iterations. Because of these findings, we shall not be using the the UVD algorithm with randomly selected algebra basis elements any more for principal component analysis purpose.

In Table 2 we present a comparison of the performance of three differernt UVD methods, in the case $p = n - 1$. The considered methods are UVD with cyclic selection of the basis elements, UVD (toral) with ordering of the basis elements in maximal commutative subalgebras (as described at the end of subsection 2.2), and UVD (Householder) based on the use of Householder transformations to achieve a tridiagonal form and with cyclic selection of the basis elements. We report the CPU times and the number of Givens rotations required by the methods to achieve convergence. The iteration stops when the error is smaller than 10^{-3}. The results indicate the superiority of UVD (toral). These results are still preliminary and must be further investigated. In the next sections only the method UVD (cyclic) will be used in the experiments.

3.1.2 Principal component analysis by the rational-mechanics-type method

In order to employ the MEC method to computing the p largest eigenvalues of a symmetric positive-definite $n \times n$ matrix A, it is necessary to properly choose a potential energy function for the abstract mechanical system. In this case, the potential energy function is:

$$U(W) = -\frac{1}{2}\text{trace}(W^T AW), \tag{33}$$

which gives rise to the forcing term $F = AW$.

Experiment	Method	CPU time	Rotations
$n = 10$, $p = 9$	UVD (cyclic)	0.0596	166
$n = 10$, $p = 9$	UVD (toral)	0.0386	109
$n = 10$, $p = 9$	UVD (Householder)	0.0572	114
$n = 20$, $p = 19$	UVD (cyclic)	0.1805	417
$n = 20$, $p = 19$	UVD (toral)	0.0575	157
$n = 20$, $p = 19$	UVD (Householder)	0.6114	1285
$n = 30$, $p = 29$	UVD (cyclic)	0.2617	681
$n = 30$, $p = 29$	UVD (toral)	0.1070	264
$n = 30$, $p = 29$	UVD (Householder)	0.5061	954

Table 2: Comparison of different UVD methods: UVD (cyclic), UVD (toral), UVD (Householder). The iteration stops when the error is less then 10^{-3}. (Principal component analysis: $n \times n$ random symmetric matrix, $p = n - 1$ number of computed eigenpairs.)

The matrix A was generated as $A = VDV^T$, where $V \in O(n)$ and $D \in \mathbf{R}^{n\times n}$ diagonal, with positive entries ordered in ascending order. In this way, a PCA $W \in \mathrm{St}(n, p)$ is such that $U^\star := U(W) = -\frac{1}{2}\sum_{i=1}^{p} D^{ii}$. Figure 6 displays a run on a randomly generated matrix A, both with MEC in the Lie-algebra and in the tangent-space formulations. In both cases, iteration stops when the absolute mismatch $|U - U^\star|$ becomes smaller than 10^{-3} or when the number of iterations exceeds 120. Algorithms' parameters were chosen as: $h = 0.1$ and $\mu = 1$. As it is readily observed, both MEC versions terminate iteration because the exiting condition $|U - U^\star| < 10^{-3}$ is met.

3.1.3 Comparisons on principal component analysis problem

We may compare the performances of the UVD method based on cyclic selection of basis elements with the performances of the MEC method (both in the algebra-based and on the tangent-space-based versions) as well as the performances of the GRAD method on principal component analysis with varying problem sizes. As performance indices, we may consider the total floating points operations requires by the algorithms to run and the discrepancy between the final value of the criterion function and the optimal value of the criterion function. Algorithms' parameters were chosen as: $h = 0.1$ and $\mu = 1$. The termination criterion for all the algorithms is that the absolute error becomes smaller than 10^{-3}. The obtained results are summarized in the Table 3, which illustrate the numerical behavior of the four considered algorithms in terms of slops, absolute estimation error and number of iterations to converge. For the UVD method, the total number of iterations

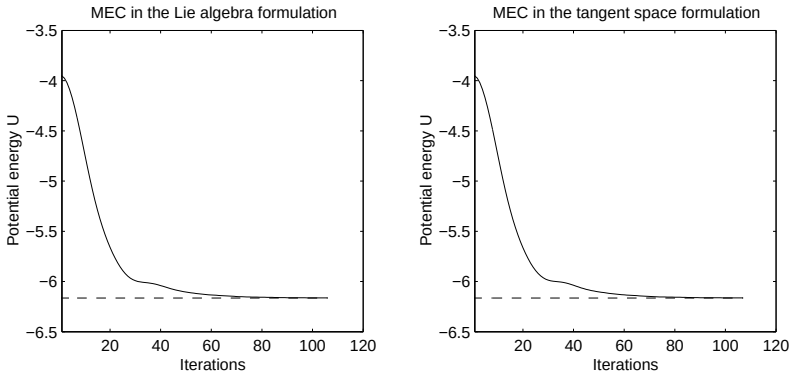


Figure 6: A run on a randomly generated matrix A. In this example: $n = 10$, $p = 3$. The left-hand panel shows the behavior of the MEC algorithm in the Lie-algebra formulation, while the right-hand panel shows the behavior of the MEC algorithm in the tangent-space formulation. Solid-line: Value of potential U during iteration. Dashed-line: Value of $U^\star$. (Principal component analysis experiment.)

was divided by the basis dimension (d) in order to show a fair comparison: Every step of the other three methods is equivalent to d steps of the UVD method. Figure 7 shows a run on a random matrix A of the four considered algorithms for $n = 10$ and $p = 8$, while Figure 8 graphically shows a computational-complexity/performance comparison for the same matrix A. It is worth recalling that the adaptation parameter of the gradient method must be chosen *ad hoc* to get convergence, while in UVD method there is no adaptation parameter. The results show that the UVD method is competitive with the other methods in terms of global computational complexity, for low-dimensional PCA problems and when $p < n$, while for large-dimensional problems the MEC algorithms may be more advantageous than UVD and GRAD algorithms.

3.2 Application to independent component analysis

Independent component analysis is a well-established statistical signal/data processing technique that aims at decomposing a set of multivariate random signals into a basis of statistically independent streams with the minimal loss of information content. The main recognized purposes of ICA are:

- *Linear blind source separation*: In this case the aim is to recover a number of statistically independent signals from their unknown linear mixtures, under simple consistency conditions. The typical consistency conditions on the unknown sources are: 1) each source signal is an independent identically distributed (IID) stationary random process; 2) the source signals are statistically independent at any time; 3) at most one among the source signals has Gaussian distribution. These requirements ensure the existence and uniqueness of the solution to the blind separation problem (but for

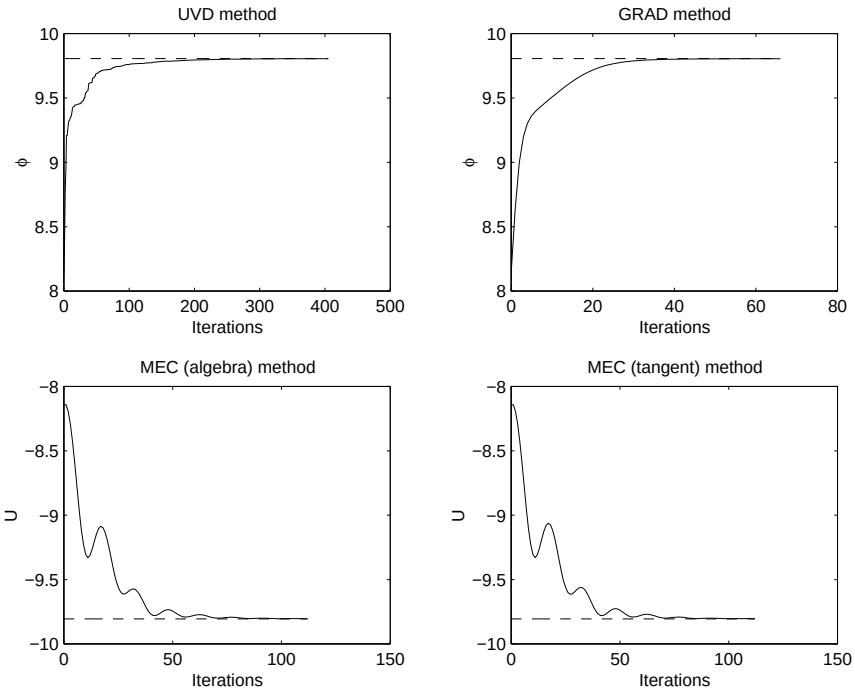


Figure 7: A run of the GRAD, UVD, MEC (algebra) and MEC (tangent) algorithms considered for comparison purpose, on a $n = 10$, $p = 8$ problem. Solid lines: Values of ϕ and U. Dashed lines: Values of $\phi^\star$ and $U^\star$. (Principal component analysis experiment.)

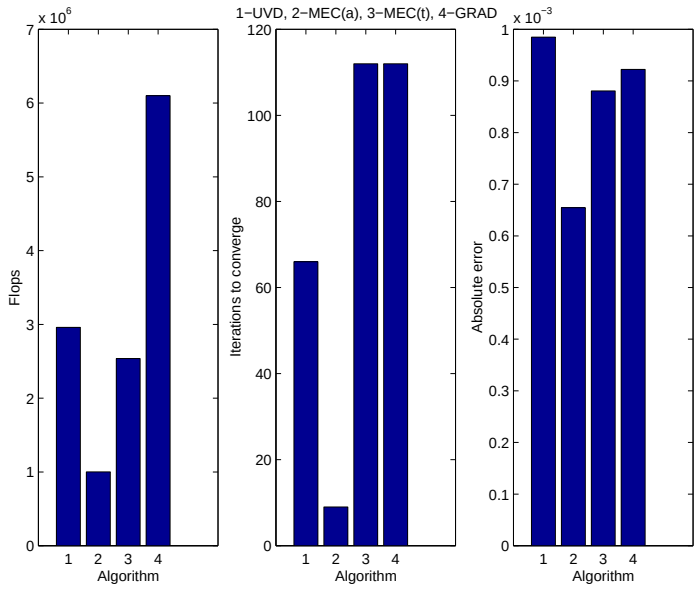


Figure 8: Computational-complexity/performance comparison of GRAD, UVD, MEC (algebra) and MEC (tangent) on a $n = 10$, $p = 8$ problem. (Principal component analysis experiment.)

Experiment	Method	Flops	Error	Iterations
$n = 10$, $p = 8$	GRAD	1,882,936	0.9771×10^{-3}	42
$n = 10$, $p = 8$	UVD (cyclic)	444,600	0.7341×10^{-3}	4
$n = 10$, $p = 8$	MEC (algebra)	2,016,264	0.9744×10^{-3}	89
$n = 10$, $p = 8$	MEC (tangent)	5,337,714	0.9168×10^{-3}	98
$n = 20$, $p = 16$	GRAD	$17,614,650$	0.9694×10^{-3}	50
$n = 20$, $p = 16$	UVD (cyclic)	16,295,160	0.1856×10^{-3}	6
$n = 20$, $p = 16$	MEC (algebra)	$17,662,442$	0.9016×10^{-3}	106
$n = 20$, $p = 16$	MEC (tangent)	$45,216,610$	0.9449×10^{-3}	106
$n = 30$, $p = 4$	GRAD	4,338,915	0.9729×10^{-3}	215
$n = 30$, $p = 4$	UVD (cyclic)	21,083,580	0.8941×10^{-3}	6
$n = 30$, $p = 4$	MEC (algebra)	100,224,060	0.9808×10^{-3}	151
$n = 30$, $p = 4$	MEC (tangent)	6,772,741	0.9571×10^{-3}	151
$n = 30$, $p = 24$	GRAD	171,366,945	0.9713×10^{-3}	145
$n = 30$, $p = 24$	UVD (cyclic)	178,672,770	0.3468×10^{-3}	9
$n = 30$, $p = 24$	MEC (algebra)	82,517,198	0.9247×10^{-3}	151
$n = 30$, $p = 24$	MEC (tangent)	215,866,643	0.9313×10^{-3}	151

Table 3: Comparison of UVD method based on cyclic selection of basis elements, MEC method (both in the algebra-based and on the tangent-space-based versions) and GRAD method. (Principal component analysis: $n \times n$ random symmetric matrix, p number of computed eigenpairs.)

ordering, sign, and power scaling);

- *Data representation and visualization*: High-dimensionality data/signals are difficult to handle and to visualize, but often contain significant redundancies. Such observation suggests the usefulness of designing signal processing algorithms that would be capable of finding a suitable lower-dimensionality representation of the signals at hand by reducing their statistical dependencies. In this context, independent component analysis technique may discover a linear projection of the data into a low-dimensional basis of statistically independent signals, that carry on no mutual information, thus providing a parsimonious maximally-informative representation of the original data.

As mentioned in the introduction to this section, the classical example used to informally explain the blind separation problem is the "cocktail party" scenario. An example that helps clarifying the concept low-dimensional data representation might arise in the analysis of vibrating machines: A rotating machine is equipped with a number of accelerometers that

measure its vibration intensities. We recognize that the recorded accelerometer signals are originated by many vibrating parts, whose number exceeds the number of accelerometers, which in turn might exceed the number of reasonable signals we can process for fault testing. By seeking for a compact representation of the measured data by the ICA technique, the obtained basis of independent signals capture the information on the complex vibrations of a small number of virtual oscillators whose linear superposition generate the observed data.

In symbols, the involved quantities in the signal model are:

- The unknown source signal stream $s(t) := [s_1(t),\ \ldots,\ s_p(t)]^T$;
- The unknown mixing matrix $A \in \mathbf{R}^{n\times p}$;
- The observable signal stream $x(t) := [x_1(t),\ \ldots,\ x_n(t)]^T$ given by $x(t) = As(t)$.

Typically, it holds $n > p$, namely, the number of observations exceeds the number of actual sources. Also, a typical assumption is that the source signals are spatially white, that means $E[ss^T] = I_p$. The aim of independent component analysis is to find estimates $y(t)$ of signals in $s(t)$ by constructing a de-mixing matrix $W \in \mathbf{R}^{n\times p}$ and by computing $y(t) := W^T x(t)$. As the involved quantities are unknown, the only way to tackle the problem at hand is to invoke statistical signal processing methods, that is, to formulate the problem of finding a de-mixing matrix W as an optimization problem on homogeneous manifolds.

The geometrical structure of the parameter space in ICA comes from a signal pre-processing step termed *signal whitening*, that is operated on the observable signal $x(t) \to \tilde{x}(t) \in \mathbf{R}^p$, in such a way that the components of the signal $\tilde{x}(t)$ are uncorrelated and have variances equal to 1, namely $E[\tilde{x}\tilde{x}^T] = I_p$ [10]. This also means that redundant observations are eliminated and the ICA problem is brought back to the smallest dimension p. A possible way to pre-whiten the signals and to remove the redundant dimensions in the data is to use eigenvalue decomposition, namely to compute $E[xx^T] = VDV^T$, with $V \in \mathrm{St}(n,p)$ and $D \in \mathbf{R}^{p\times p}$ diagonal invertible. Then pre-whitening is obtained as:

$$\tilde{x}(t) := D^{-\frac{1}{2}} V^T x(t). \tag{34}$$

Pre-whitening the observable signals implies $\tilde{x}(t) = D^{-1/2}V^T As(t) =: \tilde{A}s(t)$, where $E[\tilde{x}\tilde{x}^T] = \tilde{A}E[ss^T]\tilde{A}^T = \tilde{A}\tilde{A}^T = I_p$. Consequently, any ICA problem may be reduced to an orthogonal ICA problem, where the mixing matrix is orthogonal and, as a consequence, the de-mixing matrix should be orthogonal as well. After observable signal pre-whitening, the de-mixing matrix may be searched for such as it solves the optimization problem:

$$\max_{W\in O(p)} \phi(W). \tag{35}$$

In the ICA context, the criterion function $\phi(W)$ is termed *discriminant constrast* [10]. The contrast $\phi(W)$ actually depends on the statistics of the signals obtained by the separation rule $y(t) := W^T \tilde{x}(t)$.

As explained, after pre-whitening, the number of projected-observations in the signal $\tilde{x}(t)$ equals the number of sources. However, in some applications it is known that not all the source signals are useful, therefore, it is sensible to analyze only a few of them. This may be the case in antenna array processing (e.g., in synthetic aperture radar (SAR) imagery processing, [18]) or in non-destructive evaluation of metallic slabs by exploiting the eddy-current-phenomenon (ECT-NDE testing, [20]). In these cases, if we denote by $\overline{p} \ll p$ the actual number of independent components that are sought for, the appropriate way to cast optimization problem for ICA is:

$$\max_{W \in \mathrm{St}(n,\overline{p})} \phi(W), \text{ with } \overline{p} \ll p. \tag{36}$$

Typically, in SAR as well as in ECT-NDE applications, we set $\overline{p} = 1$.

The optimization principles (35) and (36) may be presented within a unified framework, namely:

$$\max_{W \in \mathrm{St}(p,\overline{p})} \phi(W), \text{ with } \overline{p} \leq p, \; y(t) := W^T \tilde{x}(t). \tag{37}$$

The matrix $\Pi := W^T D^{-1/2} V^T A$ is termed *separation product* and at optimum is should be a scaled version of a permutation matrix, so that $s(t) \approx y(t) = W\tilde{x}(t)$, as the basic theory of ICA ensures optimizing contrast functions yields independent components up to possible reordering, scaling and sign switching. The definition of the separation product Π allows us to define an optimization performance index, termed *inter-channel interference* (ICI), as:

$$\mathrm{ICI} := \frac{1}{p} \|\Pi\Pi^T - \mathrm{diag}(\Pi\Pi^T)\|_{\mathrm{F}}, \tag{38}$$

where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius matrix norm.

3.2.1 Independent component analysis by non-Gaussianity optimization

As a possible principle for reconstruction, the maximization or minimization of non-Gaussianity is viable. It is based on the notion that the sum of independent random variables has distribution closer to Gaussian than the distributions of the original random variables. A measure of non-Gaussianity is the kurtosis, defined for a scalar signal $z \in \mathbf{R}$ as:

$$\mathrm{kurt}(z) := E[z^4] - 3E^2[z^2]. \tag{39}$$

If the random signal z has unitary variance, then the kurtosis computes as $\mathrm{kurt}(z) = E[z^4] - 3$. Maximizing or minimizing kurtosis is thus a possible way of estimating independent components from their linear mixtures [10, 27]. For unit-variance components, this amounts

at maximizing or minimizing the fourth-order moment of the reconstructed signals. This method is readily applicable when:

- All the p source signals in a mixture $x(t)$ are super-Gaussian (namely, all the sources have positive kurtosis), possibly with the exception of one source signal, which is Gaussian (namely, it has zero kurtosis): In this case, the contrast function may be defined as $\phi(W) := \frac{1}{4}\sum_{i=1}^{p} \operatorname{kurt}(w_i^T \tilde{x})$;
- All the source signals in a mixture are sub-Gaussian (namely, all the sources have negative kurtosis), possibly with the exception of one source signal, which is Gaussian: In this case, the contrast function may be defined as $\phi(W) := -\frac{1}{4}\sum_{i=1}^{p} \operatorname{kurt}(w_i^T \tilde{x})$;

Both the contrast functions above are to be maximized under the constraint $W \in \mathrm{St}(p, \overline{p})$. In the context of independent component analysis by non-Gaussianity optimization, both full-rank analysis (namely, $\overline{p} = p$) and reduced-rank analysis (namely, $\overline{p} \ll p$) are of use. In the latter case, the *most kurtotic* components are extracted from the observations (namely, the components exhibiting the largest absolute kurtosis are extracted from the observed mixture).

3.2.2 Non-negative independent component analysis

We now add the technical hypothesis that the sources do not have degenerate (i.e., point-mass-like) joint probability density function. Under this hypothesis, an interesting variant of standard ICA may be invoked when the additional knowledge on the non-negativity of the source signals is considered. In some signal processing setting, it is known *a priori* that the source signals to be recovered have non-negative values [31, 32]. This is the case, for instance, in image processing, where the values of the luminance or the intensity of the color in the proper channel are normally expressed by non-negative integer values. Another interesting potential application is spectral un-mixing in remote sensing [24]. In hyper-spectral imagery, pixels are a mixture of more than one distinct substance. In fact, this may happen if the spatial resolution of a sensor is so low that diverse materials can occupy a single pixel, as well as when distinct materials are combined into a homogeneous mixture. Spectral de-mixing is the procedure with which the measured spectrum is decomposed into a set of component spectra and a set of corresponding abundances, that indicate the proportion of each component present in the pixels. The theoretical foundations of so-termed *non-negative independent component analysis* were stated in [31].

Under the hypotheses motivated in [31], a way to perform non-negative independent component analysis is to construct a cost function $\phi(W)$ that is identically zero if and only if the components of $y(t)$ are non-negative with probability 1. The criterion function

proposed in [32] gives thus rise to the contrast:

$$\phi(W) := -\frac{1}{2}E[\|\tilde{x} - W\rho(W^T\tilde{x})\|_2^2], \tag{40}$$

where $\|\cdot\|_2$ denotes the standard L_2 vector norm and the function $\rho : \mathbf{R} \to \mathbf{R}_0^+$ is defined as:

$$\rho(u) := \begin{cases} u, \text{ if } u \geq 0, \\ 0, \text{ otherwise,} \end{cases} \tag{41}$$

and it is supposed to act component-wise on vectors. From the definition (40), it is clear that when all the network output signals have positive values, it result $\phi = 0$, otherwise $\phi \neq 0$. In this case, performing ICA may thus be accomplished by maximizing the criterion function ϕ over the orthogonal group of matrices.

3.2.3 Independent component analysis by the univariate descent method

In the case of independent component analysis by kurtosis optimization, it is necessary to express the criterion function $\phi(W)$ in closed form for univariate optimization purpose. Likewise for the PCA case, we identify the initial point with W_0 and hypothesize to move to the point $G_{ij}(\alpha)W_0$. Therefore, the optimal value of α such that $\phi(\alpha) := \phi(G_{ij}(\alpha)W_0)$ is maximized/minimize is to be sought for.

Let us define the signal $y_0(t) := W_0^T\tilde{x}(t) \in \mathbf{R}^p$, whose components are hereafter denoted by $y_{0,r}(t)$. Straightforward computations show that:

$$\phi(\alpha) = a(\sin^4\alpha + \cos^4\alpha) + b\sin(4\alpha) + c\sin^2(2\alpha) + d, \tag{42}$$

where coefficients a, b, c and d were found to be:

$$\begin{cases} a &:= \pm\frac{1}{4}(E[y_{0,i}^4] + E[y_{0,j}^4]), \\ b &:= \pm\frac{1}{4}(E[y_{0,i}^3 y_{0,j}] - E[y_{0,i}y_{0,j}^3]), \\ c &:= \pm\frac{3}{4}E[y_{0,i}^2 y_{0,j}^2], \\ d &:= \pm\frac{1}{4}\sum_{r\neq i,j}\operatorname{kurt}(y_{0,r}) \mp \frac{3}{2}. \end{cases} \tag{43}$$

The sign depends on whether the kurtoses of the sources is known to be positive or negative. The constant term d in the expression (42) is quite burdensome to compute and does not affect the maximization of the contrast function $\phi(\alpha)$: Its computation may therefore be dispensed of.

Figure 9 shows four grey-level pictures, taken as exemplary source signals, along with their kurtoses, which are all negative. The sources were mixed by a random matrix A of size 4×4 and were then subjected to pre-whitening: Figure 10 shows the shape of the contrast function $\phi(\alpha) = \phi(G_{ij}(\alpha)W_0)$ as in (42) for $W_0 = I_4$, for two pairs of indices (i, j) in the ranges $i \in \{1,\ 2,\ 3\}$ and $j \in \{i+1,\ \ldots,\ 4\}$.

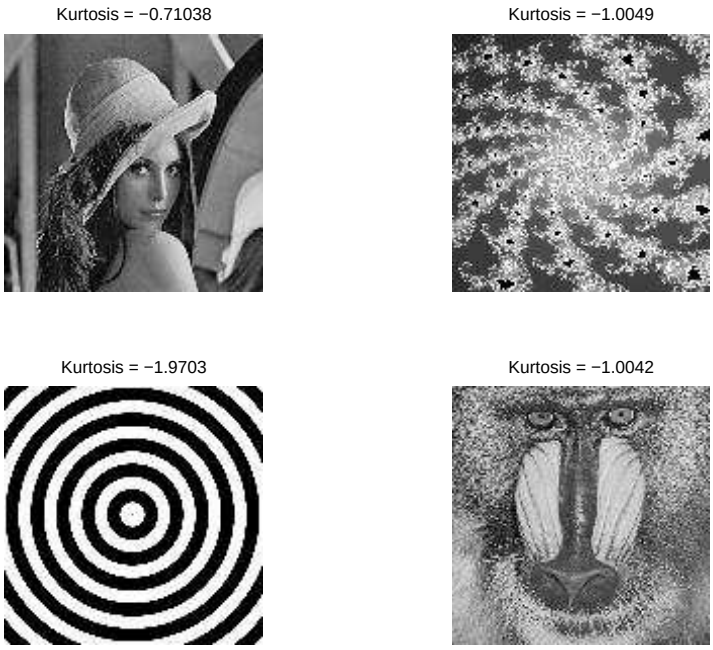


Figure 9: Four grey-level 128×128 pixels pictures along with their kurtoses. If regarded as source signals, these are sub-Gaussian sources. (Kurtosis-based independent component analysis experiment.)

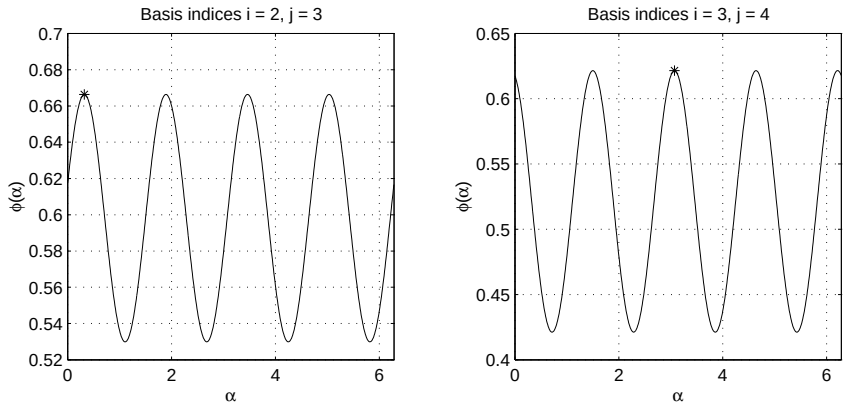


Figure 10: Shape of the contrast function (42) for $W_0 = I_4$ for two pairs of indices (i, j), after the sources of Figure 9 were mixed by a 4×4 random matrix A and after pre-whitening. The asterisks denote the maximum values of the function (42) computed numerically. (Kurtosis-based independent component analysis problem.)

In the case of non-negative independent component analysis, the contrast function cannot be easily written in closed form due to the presence of the non-linear function ρ in it. It is first worth noting that, thanks to the condition $W \in O(p)$, the contrast function (40) may be readily rewritten in terms of the signal $y(t) = W^T \tilde{x}(t)$ as $\phi(W) = -\frac{1}{2}E[\|y - \rho(y)\|_2^2]$. This observation turns out to be profitable when writing the explicit expression of the criterion $\phi(\alpha) = \phi(G_{ij}(\alpha)W_0)$. By defining again the auxiliary signal $y_0(t) := W_0^T \tilde{x}(t) \in \mathbf{R}^p$, we get:

$$\begin{aligned}\phi(\alpha) &= \frac{1}{2}\sum_{r\neq i,j} E[y_{0,r}\rho(-y_{0,r})] + \\ &\quad \frac{1}{2}E[(y_{0,i}\cos\alpha + y_{0,j}\sin\alpha)\rho(-y_{0,i}\cos\alpha - y_{0,j}\sin\alpha)] + \\ &\quad \frac{1}{2}E[(-y_{0,i}\sin\alpha + y_{0,j}\cos\alpha)\rho(y_{0,i}\sin\alpha + y_{0,j}\cos\alpha)]. \qquad (44)\end{aligned}$$

The constant term $\sum_{r\neq i,j} E[y_{0,r}\rho(-y_{0,r})]$ in the expression (44) is burdensome to compute and does not affect the maximization of the contrast function $\phi(\alpha)$, therefore, its computation may be dispensed of, in practice. Therefore, we may define the lifted NNICA criterion and compute its first order derivative:

$$\begin{aligned}\phi_\ell(\alpha) &:= \frac{1}{2}E[(y_{0,i}\cos\alpha + y_{0,j}\sin\alpha)\rho(-y_{0,i}\cos\alpha - y_{0,j}\sin\alpha)] + \\ &\quad \frac{1}{2}E[(-y_{0,i}\sin\alpha + y_{0,j}\cos\alpha)\rho(y_{0,i}\sin\alpha - y_{0,j}\cos\alpha)], \qquad (45)\\ \phi'_\ell(\alpha) &= E[(-y_{0,i}\sin\alpha + y_{0,j}\cos\alpha)\rho(-y_{0,i}\cos\alpha - y_{0,j}\sin\alpha)] + \\ &\quad -E[(y_{0,i}\cos\alpha + y_{0,j}\sin\alpha)\rho(y_{0,i}\sin\alpha - y_{0,j}\cos\alpha)]. \qquad (46)\end{aligned}$$

As an exemplary instance, Figure 11 shows the shape of the contrast function $\phi(\alpha) = \phi(G_{ij}(\alpha)W_0)$ as in (45) for $W_0 = I_4$, for two pairs of indices (i, j). The Figure also shows the shape of the derivative $\phi'(\alpha)$ as computed in (46).

The strategy of choice to compute the maximum of the univariate lifted contrast function for every pair of indices (i, j) in a numerically cheap way is:

1. Sample *coarsely* the univariate contrast function (45) in the interval $[0\ 2\pi]$. Such first sampling has the purpose of determining a value of the angle α close to the point of maximum;

2. Use a numerical algorithm to find the zero of function (46) close to the value of the angle α determined with the previous operation (e.g., by a MATLAB's built-in root-finder primitive).

As an example of behavior of the UVD method in the context of ICA, Figure 12 shows the result of a run of UVD method to optimize the non-negative ICA criterion as well as the kurtosis-based criterion. The algorithms converge steadily. As it might be expected,

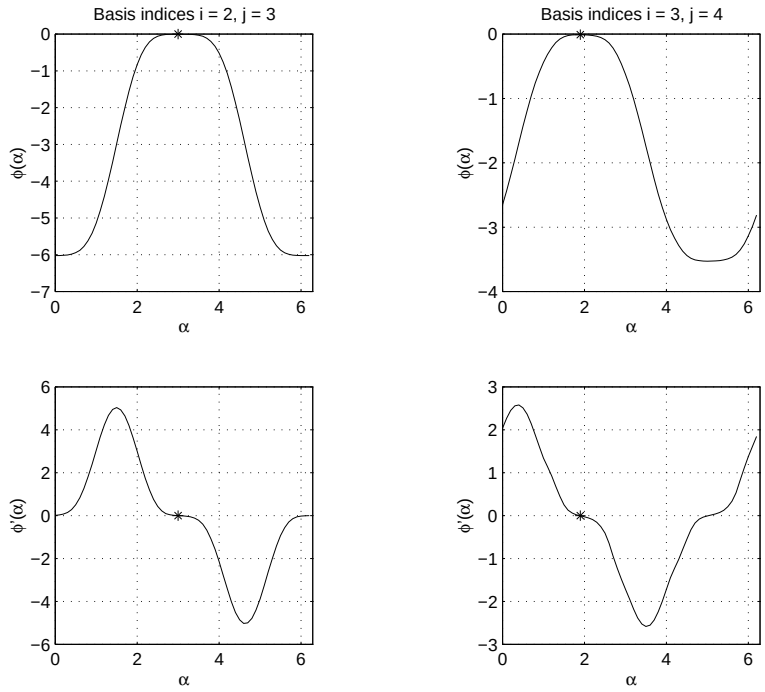


Figure 11: Shape of the lifted contrast function (45) and its derivative derivative (46) for $W_0 = I_4$ for two pairs of indices (i, j), after the sources of Figure 9 were mixed by a 4×4 random matrix A and after pre-whitening. The asterisks denote the maximum values of the function (45) computed numerically. (Non-negative independent component analysis problem.)

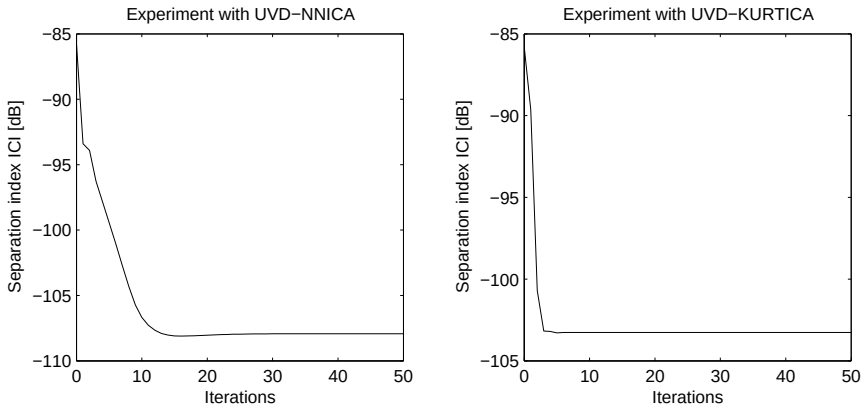


Figure 12: Result of a run of UVD method to optimize the non-negative ICA criterion (UVD-NNICA) as well as the kurtosis-based criterion (UVD-KURTICA).

the method based on non-negativity minimization exhibits better component analysis capability, as it is based on a *a priori* knowledge about the sources, while this is not the case for the kurtosis-based method. However, the latter method exhibits quicker convergence.

3.2.4 Independent component analysis by the rational-mechanics-type method

In order to formulate a kurtosis-based ICA or a NNICA procedure within the rational-mechanics settings, it is necessary to compute the expression of the active forces in these two cases.

In the case of independent component analysis by kurtosis optimization, it is worth recalling that a contrast function is intended to be maximized, while the mechanical system's potential energy minimizes according to system's dynamics. Therefore, according to the cases considered in the section 3.2.1:

- All the p source signals in a mixture $x(t)$ are super-Gaussian (possibly with the exception of a source signal which is allowed to be Gaussian): In this case, the potential energy function may be selected as $U(W) := -\frac{1}{4}(\sum_{i=1}^{p} E[(w_i^T \tilde{x})^4] - 3p)$ and thus the active-force-matrix writes thus $F = E[\tilde{x}(\tilde{x}^T W)^3]$, where $(\cdot)^3$ is meant to act component-wise;

- All the p source signals in a mixture $x(t)$ are sub-Gaussian (possibly with the exception of a source signal which is allowed to be Gaussian): In this case, the potential energy function may be selected as $U(W) := \frac{1}{4}(\sum_{i=1}^{p} E[(w_i^T \tilde{x})^4] - 3p)$ and the forcing term writes thus $F = -E[\tilde{x}(\tilde{x}^T W)^3]$.

In the case of non-negative independent component analysis, according to the contents of section 3.2.2, the potential energy function should be chosen as $U(W) := \frac{1}{2}E[\|\tilde{x} - W\rho(W^T\tilde{x})\|_2^2]$, which gives rise to the forcing term $F = -E[\tilde{x}\rho(\tilde{x}^T W)]$.

As an example of behavior of the MEC method in the context of ICA, Figure 13 shows the result of a run of MEC method to optimize the non-negative ICA criterion as well as the kurtosis-based criterion. The MEC algorithm is considered again in both Lie-algebra and tangent-space formulation. In the case of kurtosis-based ICA, the learning parameters were chosen as $h = 0.1$ and $\mu = 1$. In such case, the algorithms converge steadily. In the non-negative ICA case, the learning parameters were chosen as $h = 0.5$ and $\mu = 1$. Despite a larger learning step size was selected, the algorithm converges very slowly in this case.

3.2.5 Comparisons on independent component analysis problem

In the present section, comparative results among the GRAD, UVD and MECs algorithms are illustrated and discussed separately for the non-negative ICA and the kurtosis-based ICA. The algorithms were implemented in MATLAB 6.5, which does not implement flops

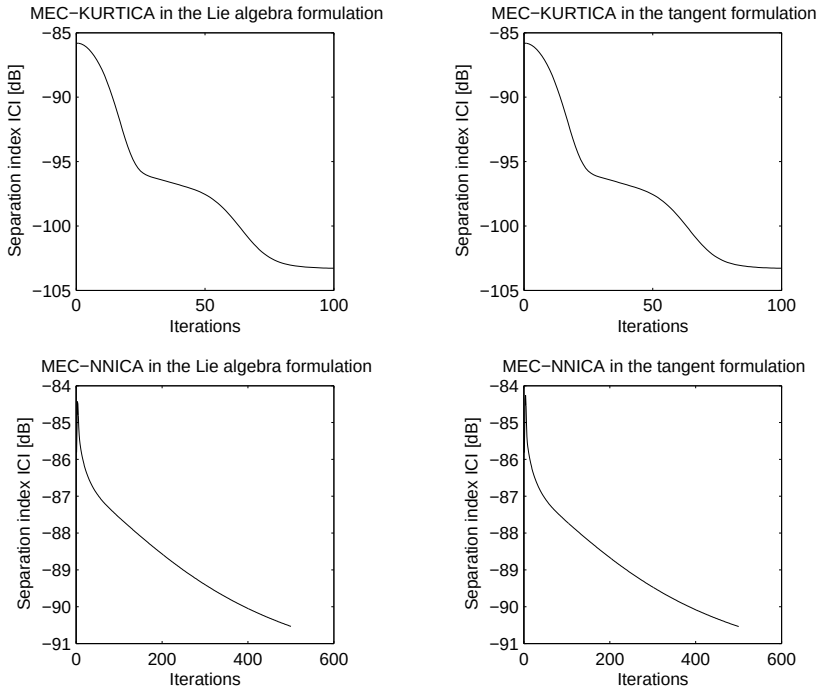


Figure 13: Result of a run of MEC method to optimize the non-negative ICA criterion (MEC-NNICA) as well as the kurtosis-based criterion (MEC-KURTICA). The MEC algorithm is tested in both Lie-algebra and tangent-space formulation.

count, therefore, the computational complexity of the developed algorithms was presented in terms of CPU run-time. As shown in the previous sections, when the ICA algorithms converge properly they reach values of the ICI index below the -100 dB, therefore, as exit criterion for the iteration we choose the threshold of -100 dB, while a maximum number of 100 iterations was allowed for each algorithm.

The first set of results pertains to the non-negative ICA case. Figure 14 illustrates the results obtained on a mixture of the four digital images (as explained in section) by running the algorithms GRAD (with step-size $h = 0.5$), UVD, MEC in the Lie-algebra formulation (with parameters $h = 0.5$ and $\mu = 1$) and MEC in the tangent-space formulation (with parameters $h = 0.5$ and $\mu = 1$). The results illustrated in the Figure show that, among the four considered algorithms, the UVD is the only one that converges within 100 iterations (actually, it converges in 6 iterations). The simulations however show that the UVD method is the most expensive one among the four considered methods, due to the needed zero-finding sub-procedure.

The second set of results pertains to the kurtosis-based ICA case. The Figure 15 illustrates the results obtained by running the algorithms GRAD (with step-size $h = 0.5$), UVD, MEC in the Lie-algebra formulation (with parameters $h = 0.5$ and $\mu = 1$) and MEC in the tangent-space formulation (with parameters $h = 0.5$ and $\mu = 1$). The obtained nu-

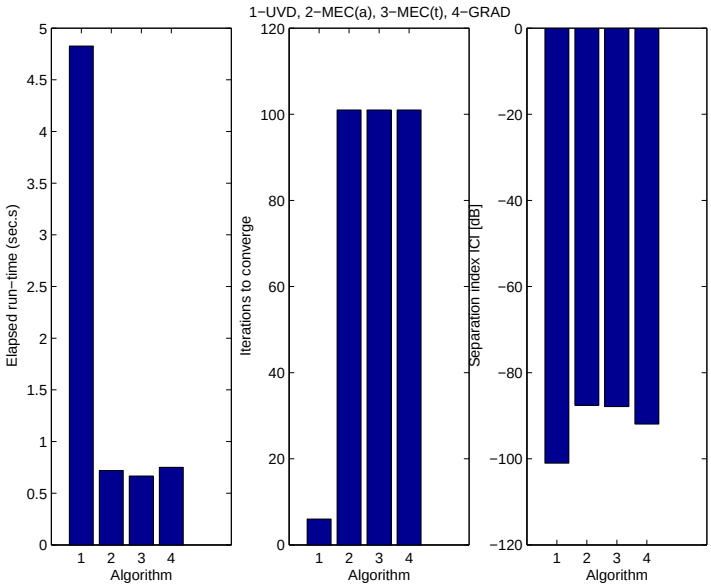


Figure 14: Computational-complexity/performance comparison of GRAD, UVD, MEC (algebra) and MEC (tangent). (Non-negative independent component analysis.)

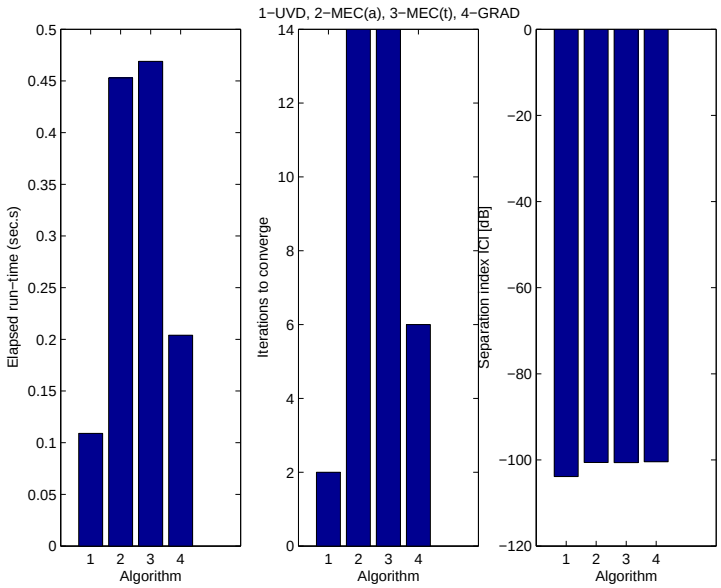


Figure 15: Computational-complexity/performance comparison of GRAD, UVD, MEC (algebra) and MEC (tangent). (Kurtosis-based independent component analysis experiment.)

merical results show that all the four considered algorithms converge within 100 iterations. The simulations show that the UVD method is the most convenient one among the four considered algorithms as well as the fastest one in terms of iterations to converge.

4 Conclusions

In this paper we have presented a framework for descent methods on homogeneous manifolds based on the use of retraction maps and Lie group actions. Within this framework we presented a new class of univariate descent methods (UVD) optimizing in turn in the direction of each of the basis elements of the Lie algebra. The methods require a very moderate computational burden per step and show good performance in eigenvalue problems and signal processing applications. The study of these methods is still preliminary and deserves further attention.

Gradient type methods on manifolds can be also included in the considered framework and are compared to the univariate descent methods. We showed that in some cases it might be convenient to formulate the descent methods using tangent space parametrisations as this might lead to considerable gain in the computational complexity. Retraction maps are particularly well-suited for this task.

The methods discussed in the paper have been applied to a variety of problems in statistical signal processing.

Acknowledgment

The first author of this paper wishes to thank J. P. Dedieu and J. C. Yakoubson for the pleasant stay in Luminy during May 2005 where the topics of this paper were first discussed. We also wish to thank L. Lopez for encouraging and supporting our work and the use of geometric methods in statistical signal processing. The present work was conceived while the second author was visiting the Department of Mathematical Sciences at the Faculty of Information Technology, Mathematics and Electrical Engineering of the Norwegian University of Science and Technology (Trondheim, Norway) in February-March 2005.

The preparation of this manuscript was consistently advanced while the second author was visiting the Laboratory for Signal and Image Processing (SIP) at the Department of Electrical and Electronic Engineering of the Tokyo University of Agriculture and Technology (Koganei-shi, Tokyo, Japan) and the Mathematical Neuroscience Unit of Brain Science Institute (BSI) at RIKEN (Wako-shi, Saitama, Japan) during summer 2006. The author wishes to gratefully thank the SIP Laboratory director, Prof. Toshihisa Tanaka and the MNS Unit director, Prof. Shun-ichi Amari, for making this fruitful visit be possible.

References

[1] P.-A. Absil, C. G. Baker, K. A. Gallivan, *Trust-region methods on Riemannian manifolds*, to appear on J. of FOCM, 2006.

[2] R. Adler, J.-P. Dedieu, J.Y. Margulies, M. Martens and M. Shub, *Newton's method on Riemannian manifolds and a geometric model for the human spine*, IMA J. of Numerical Analysis, Vol. 22, No. 3, 359 – 390, 2002.

[3] R.W. Brockett, *Dynamical systems that sort lists, diagonalize matrices and solve linear programming problems*, Linear Algebra and Its Applications, Vol. 146, pp. 79 – 91, 1991.

[4] E. Celledoni and S. Fiori, *Neural learning by geometric integration of reduced 'rigid-body' equations*, Journal of Computational and Applied Mathematics (JCAM), Vol. 172, No. 2, pp. 247 – 269, 2004.

[5] E. Celledoni and A. Iserles, *Approximating the exponential form of a Lie algebra to a Lie group*, Mathematics of Computation, Vol. 69, pp. 1457 – 1480, 2000.

[6] E. Celledoni and A. Iserles, *Methods for the approximation of the matrix exponential in a Lie- algebraic setting*, IMA J. Numer. Anal., Vol. 21, pp. 463–488, 2001.

[7] E. Celledoni and B. Owren, *On the implementation of Lie group methods on the Stiefel manifold*, Numerical Algorithms, Vol. 32, pp. 163 – 183, 2003.

[8] E. Celledoni and B. Owren, *A class of intrinsic schemes for orthogonal integration*, SIAM Journal on Numerical Analysis, Vol. 21, pp. 463 – 488, 2001.

[9] M.T. Chu and K.R. Drissel, *The projected gradient method for least squares matrix approximations with spectral constraints*, SIAM Journal of Numerical Analysis, Vol. 27, pp. 1050 – 1060, 1990.

[10] P. Comon, *Independent component analysis, a new concept ?*, Signal Processing, Vol. 36, pp. 287 – 314, 1994

[11] S. Costa and S. Fiori, *Image compression using principal component neural networks*, Image and Vision Computing Journal (special issue on "Artificial Neural Network for Image Analysis and Computer Vision"), Vol. 19, No. 9-10, pp. 649 – 668, 2001.

[12] J.-P. Dedieu and D. Nowicki, *Symplectic methods for the approximation of the exponential map and the Newton iteration on Riemannian submanifolds*, Journal of Complexity, Vol. 21, No. 4, pp. 487 – 501, 2005.

[13] A. Edelman, T.A. Arias and S.T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM Journal on Matrix Analysis Applications, Vol. 20, No. 2, pp. 303 – 353, 1998.

[14] Y. Ephraim and L. Van Trees, *A signal subspace approach for speech enhancement*, IEEE Transactions on Speech and Audio Processing, Vol. 3, No. 4, pp. 251 – 266, 1995.

[15] S. Fiori, *A theory for learning by weight flow on Stiefel-Grassman manifold*, Neural Computation, Vol. 13, No. 7, pp. 1625 – 1647, 2001.

[16] S. Fiori, *A theory for learning based on rigid bodies dynamics*, IEEE Transactions on Neural Networks, Vol. 13, No. 3, pp. 521 – 531, 2002.

[17] S. Fiori, *Unsupervised neural learning on Lie group*, International Journal of Neural Systems, Vol. 12, No.s 3 & 4, pp. 219 – 246, 2002.

[18] S. Fiori, *Overview of independent component analysis technique with an application to synthetic aperture radar (SAR) imagery processing*, Neural Networks (Special Issue on "Neural Networks for Analysis of Complex Scientific Data: Astronomy, Geology and Geophysics"), Vol. 16, No. 3-4, pp. 453 – 467, 2003.

[19] S. Fiori, *Non-linear complex-valued extensions of Hebbian learning: An essay*, Neural Computation, Vol. 17, No. 4, pp. 779 – 838, 2005.

[20] E. Frulloni and S. Fiori, *Eddy-current-based non-destructive evaluation data quality enhancement through independent component analysis*, Scientific Bulletin of the Academic Computer Center in Gdansk ("TASK Quarterly" Journal), Vol. 8, No. 3, pp. 359 – 375, 2004.

[21] K. Gao, M.O. Ahmed and M.N. Swamy, *A constrained anti-Hebbian learning algorithm for total least-squares estimation with applications to adaptive FIR and IIR filtering*, IEEE Transactions on Circuits and Systems – Part II, Vol. 41, No. 11, pp. 718 – 729, 1994.

[22] G.H. Golub and C.F. Van Loan, *Matrix Computations*, The John Hopkins University Press, 1996.

[23] U. Helmke and J. B. Moore, Optimization and dynamical systems. Communications and Control Engineering Series. Springer-Verlag London, Ltd., London, 1994.

[24] N. Keshava and J.F. Mustard, *Spectral unmixing*, IEEE Signal Processing Magazine, Vol. 19, No. 1, pp. 44 – 57, 2002.

[25] M. Kleinsteuber, U. Helmke, K. Hüper *Jacobi's algorithm on compact Lie algebras*, SIAM J. Matrix Anal. Appl. , Vol. 26, pp. 42–69, 2004.

[26] B.C. Moore, *Principal component analysis in linear systems: Controllability, observability and model reduction*, IEEE Transactions on Automatic Control, Vol. AC-26, No. 1, pp. 17 – 31, 1981.

[27] E. Moreau and O. Macchi, *Higher order contrasts for self-adaptive source separation*, International Journal of Adaptive Control and Signal Processing, Vol. 10, No. 1, 19 – 46, 1996.

[28] H. Munthe-Kaas and A. Zanna, *Numerical integration of differential equations on homogeneous manifolds*, Foundations of computational mathematics, (Rio de Janeiro 1997), pp. 305–315, Springer, Berlin, 1997.

[29] H. Munthe-Kaas *High order Runge-Kutta methods on manifolds*, Appl. Numer. Math., Vol. 29, pp. 115–127, 1999.

[30] , B. Owren and A. Marthinsen, *Integration methods based on canonical coordinates of the second kind*, Numer. Math. 87, pp. 763–790, 2001.

[31] M.D. Plumbley, *Conditions for non-negative independent component analysis*, IEEE Signal processing Letters, Vol. 9, No. 6, pp. 177 – 180, 2002.

[32] M.D. Plumbley, *Algorithms for nonnegative independent component analysis*, IEEE Transactions on Neural Networks, Vol. 14, No. 3, pp. 534 – 543, 2003.

[33] M. Shub, *Some remarks on dynamical systems and numerical analysis*, in Dynamical systems and partial differential equations (L. Lara-Carrero and J. Lewowicz editors, VII ELAM), pp. 62–92, 1986.

[34] T. Tanaka and S. Fiori, *Simultaneous tracking of the best basis in reduced-rank wiener filter*, International Conference on Acoustics, Speech and Signal Processing (IEEE-ICASSP, Toulouse, France), Vol. III, pp. 548 – 551, 2006.

[35] L. Xu, E. Oja and C.Y. Suen, *Modified Hebbian learning for curve and surface fitting*, Neural Networks, Vol. 5, pp. 393 – 407, 1992.

[36] N. J. Wildberger, *Diagonalization in compact Lie algebras and a new proof of a theorem of Kostant*, Proc. Amer. Math. Soc., Vol. 119, pp. 649–655, 1993.

[37] B. Yang, *Projection approximation subspace tracking*, IEEE Transactions on Signal Processing, Vol. 43, No. 1, pp. 1247 – 1252, 1995.

[38] K. Zhang and T.J. Sejnowski, *A theory of geometric constraints on neural activity for natural three-dimensional movement*, Journal of Neuroscience, Vol. 19, No. 8, pp. 3122 – 3145, 1999.