94-81

SVD AND SIGNAL PROCESSING, III
Algorithms, Architectures and Applications
M. Moonen and B. De Moor (Editors)

THE RIEMANNIAN SINGULAR VALUE DECOMPOSITION

B.L.R. DE MOOR
ESAT-SISTA, Katholieke Universiteit Leuven
Kardinaal Mercierlaan 94
3001 Leuven, Belgium
bart.demoor@esat.kuleuven.ac.be

ABSTRACT. We define a nonlinear generalization of the SVD, which can be interpreted as a restricted SVD with Riemannian metrics in the column and row space. This so-called Riemannian SVD occurs in structured and weighted total least squares problems, for instance in the least squares approximation of a given matrix A by a rank deficient Hankel matrix B. Several algorithms to find the 'minimizing' singular triplet are suggested. This paper reveals interesting and sometimes unexplored connections between linear algebra (structured matrix problems), numerical analysis (algorithms), optimization theory, (differential) geometry and system theory (differential equations, stability, Lyapunov functions). We also point out some open problems.

KEYWORDS. (Restricted) singular value decomposition, gradient flows, differential geometry, continuous algorithms, total least squares problems, power method.

1 THE RIEMANNIAN SVD: MOTIVATION

Since the work by Eckart-Young [14], we know how to obtain the best rank deficient least squares approximation of a given matrix $A \in \mathbb{R}^{p\times q}$ of full column rank q. This approximation follows from the SVD of A by subtracting from A the rank one matrix $u.\sigma.v^T$, where (u,σ,v) is the singular triplet corresponding to the smallest singular value σ, which satisfies

$$\begin{aligned} Av &= u\sigma , & u^Tu &= 1 , \\ A^Tu &= v\sigma , & v^Tv &= 1 . \end{aligned} \quad (1)$$

Here $u \in \mathbb{R}^p$ and $v \in \mathbb{R}^q$ are the corresponding left resp. right singular vector.
When formulated as an optimization problem, we obtain

$$\min_{\substack{B \in \mathbb{R}^{p\times q} \\ y \in \mathbb{R}^q}} \|A-B\|_F^2 \quad \text{subject to} \quad \begin{array}{l} B\,y = 0 , \\ y^Ty = 1 , \end{array} \quad (2)$$

the solution of which follows from (1) as $B = A - u\sigma v^T$, $y = v$. Here, σ is the smallest singular value of A and it is easy to prove that $\min_B \|A-B\|_F^2 = \sigma^2$ if B is required to be rank deficient. This problem is also known as the Total (Linear) Least Squares (TLS) problem and has a long history [1] [2] [14] [16] [17] [20] [25] [27] [28]. The vector y describes a linear relation between the columns of the approximating matrix B, which therefore is rank deficient as required.

In this paper we consider two generalizations of the TLS problem. First of all, in the remainder of this Section, we describe how the introduction of user-given weights and/or the requirement that the approximant B is to be structured (e.g. Hankel or Toeplitz), leads to a (nonlinear) generalization of the SVD, which we call the *Riemannian SVD*. In Section 2, we show that the analysis and design of continuous time algorithms provides useful insight in many problems and how they also provide a unifying framework in which several mathematical engineering disciplines meet. In Section 3, we discuss the power method in more or less detail to illustrate this claim. We also derive some new continuous-time algorithms, that are not yet completely understood. In Section 4, we discuss some ideas to generate algorithms for the Riemannian SVD. Conclusions are presented in Section 5.

There are at least two important variations on the Total Least Squares problem (2)[1].

The first one is called the Weighted TLS problem:

$$\min_{\substack{B \in \mathbb{R}^{p\times q} \\ y \in \mathbb{R}^q}} \sum_{i=1}^{p}\sum_{j=1}^{q}(a_{ij}-b_{ij})^2 w_{ij} \quad \text{subject to} \quad \begin{array}{l} By = 0 , \\ y^ty = 1 , \end{array} \quad (3)$$

where the scalars $w_{ij} \in \mathbb{R}^+$ are user-defined weights. An example is given by chosing $w_{ij} = 1/a_{ij}^2$ in which case one minimizes (a first order approximation to) the sum of relative errors squared (instead of the sum of absolute errors squared as in (2)). Another example corresponds to the choice $w_{ij} \in \{0,1\}$ in which case some of the elements of B will be the same as corresponding elements of A (namely the ones that correspond to a weight $w_{ij} = 1$, see also the Example in Section 4.2). Yet other examples of weighted TLS problems are given in [10] [11].

The second extension consists of adding a so-called structural constraint to the optimization problem (2), in which case we have a Structured TLS problem. For instance, we could require the matrix B to be a Hankel matrix. The problem we are solving then is to approximate a given (possible structured) full column rank matrix A by a rank deficient Hankel matrix B. Rank deficient Hankel matrices in particular have important applications in systems and control theory. But in general there are many other applications where it is required to find rank deficient structured approximants [10] [11] [12] [13]. The results of this paper apply to affine matrix structures, i.e. matrices that can be written as a linear combination of a given set of fixed basis matrices. Examples are (block) Hankel, Toeplitz,

[1] The list of constraints discussed in this paper is not exhaustive. For instance, one could impose constraints on the vectors y in the null space, an example that is not treated here (see e.g. [9]).

Brownian, circulant matrices, matrices with a given zero pattern or with fixed entries, etc...

The main result, which we have derived in [10], applies to both Weighted and/or Structured TLS problems and states that these problems can be solved by obtaining the singular triplet that corresponds to the smallest singular value that satisfies:

$$\begin{array}{ll} Av = D_v u\sigma\,, & u^T D_v u = 1\,, \\ A^T u = D_u v\sigma\,, & v^T D_u v = 1\,. \end{array} \qquad (4)$$

Notice the similarity with the SVD expressions in (1). Here A is the structured data matrix that one wants to approximate by a rank deficient one. D_u and D_v are nonnegative or positive definite matrix functions of the components of the left and right singular vectors u and v. Their precise structure depends on the weights in (3) and/or the required affine structure of the rank deficient approximant B.

To give one example (which happens to be a combination of a Weighted and Structured TLS problem), let us consider the approximation of a full column rank Hankel matrix $A \in \mathbb{R}^{p\times q}, p \geq q$, $\mathrm{rank}(A) = q$, by a rank deficient Hankel matrix B such that $\|A - B\|_F^2$ is minimized. In this case, the matrix D_v has the form $D_v = T_v W^{-1} T_v^T$ where

$$W = \mathrm{diag}[\,1\;2\;3\;\ldots\;\underbrace{q\;q\;\ldots\;q}_{(p-q+1)\text{ times}}\;\ldots 3\;2\;1\,]\,,$$

and T_v is a banded Toeplitz matrix (illustrated here for the case $p = 4$, $q = 3$) of the form:

$$T_v = \begin{pmatrix} v_1 & v_2 & v_3 & 0 & 0 & 0 \\ 0 & v_1 & v_2 & v_3 & 0 & 0 \\ 0 & 0 & v_1 & v_2 & v_3 & 0 \\ 0 & 0 & 0 & v_1 & v_2 & v_3 \end{pmatrix}\,.$$

The matrix D_u is constructed similarly as $D_u = T_u W^{-1} T_u^T$. Obviously, in this example, both D_u and D_v are positive definite matrices. Observe that B has disappeared from the picture, but it can be reconstructed as

$$B = A - \text{multilinear function of } (u, \sigma, v)\,,$$

(see the constructive proof in [11] for details). Observe that the modification to A is no longer a rank one matrix as is the case with the 'unstructured' TLS problem (2). Instead, the modification is a multilinear function of the 'smallest' singular triplet, the detailed formulas of which can be found in [10] [11].
We are interested in finding the *smallest* singular value in (4) because it can be shown that its square is precisely equal to the object function. For more details and properties and other examples and expressions for D_u and D_v for weighted and structured total least squares problems we refer to [10] [11] [12] [13].

In the special case that $D_u = I_p$ and $D_v = I_q$ we obtain the SVD expressions (1). In the case that D_u and D_v are fixed positive definite matrices that are independent of u and v, one obtains the so-called *Restricted* SVD, which is extensively studied in [7] together with some structured/weighted TLS problems for which it provides a solution. In the Restricted

SVD, D_u and D_v are positive (or nonnegative) definite matrices which can be associated to a certain inner product in the column and row space of A. Here in (4), D_u and D_v are also positive (nonnegative) definite, but instead of being constant, their elements are a function of the components of u and v. It turns out that we can interprete these matrices as Riemannian *metrics*, an interpretation which might be useful when developping new (continuous-time) algorithms. For this reason, we propose to call the equations in (4), the Riemannian SVD [2].

2 CONTINUOUS-TIME ALGORITHMS

As discussed in the previous section, in order to solve a (weighted and/or structured) TLS problem, we need the 'smallest' singular triplet of the (Riemannian) SVD. The calculation of the complete SVD is obviously unnecessary (and in the case of the Riemannian SVD there is even no 'complete' decomposition). For the calculation of the smallest singular value and corresponding singular vectors of a given matrix A, one could apply the *power method* or inverse iteration to the matrix A^TA (see e.g. [16] [24] [29]), which will be discussed in some more detail in the next Section.

One of the goals of this paper is to point out several interesting connections between linear algebra, optimization theory, numerical analysis, differential geometry and system theory. We also would like to summarize some recent developments by which *continuous time algorithms* for solving and analyzing numerical problems have gained considerable interest the last decade or so. Roughly speaking, a continuous time method involves a system of differential equations. The idea that a computation can be thought as a flow that starts at a certain initial state and evolves until it reaches an equilibrium point (which then is the desired result of the computation) is a natural one when one thinks about iterative algorithms and even more, about recent developments in natural information processing systems related to artificial neural networks [3]. There are several reasons why the study of continuous time algorithms is important. Continuous time methods can provide new additional insights with respect to and shed light upon existing discrete time iterative or recursive algorithms (such as e.g. in [22], where convergence properties of recursive algorithms are analysed via an associated differential equation). In contrast to the local properties of some discrete time methods, the continuous time approach often offers a global picture. Even if they are not particularly competitive with discrete time algorithms, the combination of parallelism and analog implementation does seem promising for some continuous-time algorithms. In many cases, continuous-time algorithms provide an alternative or sometimes even better understanding of discrete-time versions (e.g. in optimization, see the continuous-time version of interior point techniques in [15, p.126], or in numerical analysis, the self-similar

[2] This name is slightly misleading in the sense that we do NOT want to suggest that there is a complete decomposition with $\min(p, q)$ different singular triplets, which are mutual 'independent' (they are 'orthogonal') and which can for instance be added together to give an additive decomposition of the matrix A (the dyadic decomposition). There might be several solutions to (4) (for some examples, there is only one), but since each of these solutions goes with a different matrix D_u and D_v, it is not exaclty clear how these solutions relate to each other, let alone that they would add together in one way or another to obtain the matrix A.

[3] Specifically for neural nets and SVD, we refer to e.g. [3] [4] [21] [23] (and the references in these papers).

iso-spectral (calculating the eigenvalue decomposition) or self-equivalent (singular value decomposition) matrix differential flows, see the article by Helmke [19] in this Volume and its rather complete list of references). The reader should realize that it is not our intention to claim that these continuous-time algorithms are in any sense competitive with classical algorithms from e.g. numerical analysis. Yet, there are examples in which we have only continuous-time solutions and for which the discrete-time iterative counterpart has not yet been derived.

To give one example of a continuous-time unconstrained minimization algorithm, let us derive the continuous-time steepest descent method and prove its convergence to a local minimum. Consider the minimization of a scalar object function $f(z)$ in p variables $z \in \mathbb{R}^p$. Let its gradient with respect to the elements of z be $\nabla_z f(z)$. It is straightforward to prove that the system of differential equations

$$\dot{z}(t) = -\nabla_z f(z) \, , \; z(0) = z_0 \, , \tag{5}$$

where z_0 is a given initial state, will converge to a local minimum. This follows directly from the chain rule:

$$\frac{df(z)}{dt} = (\nabla_z f(z))^T \dot{z} = -\|\nabla_z f(z)\|^2 \, , \tag{6}$$

which implies that the time derivative of $f(z)$ is always negative, hence, as a function of time, $f(z)$ is non-increasing. This really means that the norm of the gradient is a Lyapunov function for the differential equation, proving convergence to a local minimum.

In Section 3.8 we will show how to derive continuous-time algorithms for *constrained* minimization problems, using ideas from differential geometry.

3 CONTINUOUS-TIME ALGORITHMS FOR THE EIGENVALUE PROBLEM

The symmetric eigenvalue problem is the following: Given a matrix $C = C^T \in \mathbb{R}^{p \times p}$, find at least one pair (x, λ) with $x \in \mathbb{R}^p$ and $\lambda \in \mathbb{R}$ such that

$$Cx = x\lambda \, , \; x^T x = 1 \, . \tag{7}$$

Of course, this problem is of central importance in most engineering fields and we refer to [16] [24] [29] for references and bibliography.
Here we will concentrate on power methods, both in continuous and discrete time. The continuous-time power methods are basically systems of vector differential equations. Some of these have been treated in the literature (see e.g. [6] [18] [26]). Others presented here are new. The difference with the matrix differential equations referred to in the previous section is that typically, these vector differential equations compute only one eigenvector or singular vector while the matrix differential equations upon convergence deliver all of them (i.e. the complete decomposition). Obviously, when solving (structured/weighted) TLS problems, we only need the 'minimal' triplet in (1) or (4). For the TLS problem (2) one might also (in principle) calculate the smallest eigenvalue of $A^T A$ and its corresponding eigenvector.

3.1 GEOMETRIC INTERPRETATION

Consider the two quadratic surfaces $x^T C x = 1$ and $x^T x = 1$. While the second surface is the unit sphere in p dimensions, the first surface can come in many disguises. For instance, when C is positive definite, it is an ellipsoid. In three dimensions, depending on its inertia, it can be a one-sheeted or two-sheeted hyperboloid or a (hyperbolic) cilinder. In higher dimensions, there are many possibilities, the enumeration of which is not relevant right now. In each of these cases, the vectors Cx and x are the normal vectors at x to the two surfaces. Hence, when trying to solve the symmetric eigenvalue problem, we are looking for a vector x such that the normal at x to the surface $x^T C x = 1$ is proportional to the normal at x to the unit sphere $x^T x = 1$. The constant of proportionality is precisely the eigenvalue λ.

3.2 THE EXTREME EIGENVALUES AS AN UNCONSTRAINED OPTIMIZATION PROBLEM

Consider the unconstrained optimization problem:

$$\min_{z \in \mathbb{R}^p} f(z) \quad \text{with} \quad f(z) = \frac{1}{2}(z^T C z)/(z^T z) \, . \tag{8}$$

It is straightforward to see that

$$\nabla_z f(z) = (Cz(z^T z) - z(z^T C z))/(z^T z)^2 \, , \tag{9}$$

from which it follows that the stationary points are given by

$$C(z/\sqrt{z^T z}) = (z/\sqrt{z^T z})(z^T C z)/(z^T z) \, .$$

This can only be satisfied if $x = z/\|z\|$ is an eigenvector of C with corresponding eigenvalue $x^T C x$. Obviously, the minimum will correspond to the minimal eigenvalue of C (which can be negative, zero or positive). The maximum of $f(z)$ will correspond to the maximum eigenvalue of C.

3.3 EXTREME EIGENVALUES AS A CONSTRAINED OPTIMIZATION PROBLEM

We can also formulate a constrained optimization problem as:

$$\min_{x \in \mathbb{R}^p} f(x) \;\text{ subject to }\; x^T x = 1 \;\text{ where }\; f(x) = x^T C x \, . \tag{10}$$

The Lagrangian for this constrained problem is

$$L(x, \lambda) = x^T C x + \lambda(1 - x^T x) \, ,$$

where $\lambda \in \mathbb{R}$ is a scalar Lagrange multiplier. The necessary conditions for a stationary point now follow from $\nabla_x L(x, \lambda) = 0$ and $\nabla_\lambda L(x, \lambda) = 0$, and will correspond exactly to the two equations in (7). Observe that these equations have p solutions (x, λ) while we are only interested in the minimizing one.

3.4 THE CONTINUOUS-TIME POWER METHOD

Let us now apply the continuous-time minimization idea of (6) to the unconstrained minimization problem (8). We then find the 'steepest descent' nonlinear differential equation

$$\dot{z} = -(Cz(z^Tz) - z(z^TCz))/(z^Tz)^2 \,. \tag{11}$$

An important property of this flow is that it is *isonormal*, i.e. the norm of the state vector z is constant over time: $\|z(t)\| = \|z(0)\|$, $\forall t \geq 0$. This is easy to see since $d\|z\|^2/dt = 2\, z^T\dot{z} = 0$, hence $\|z(t)\|$ is constant over time. This means that the state of the nonlinear system (11) evolves on a sphere with radius given by $\|z(0)\|$. We can assume without loss of generality that $\|z(0)\| = 1$. Hence (11) evolves on the unit sphere. Replacing z by x with $\|x\|^2 = x^Tx = 1$ we can rewrite (11)

$$\dot{x} = -Cx + x\frac{x^TCx}{x^Tx} \,. \tag{12}$$

We know about this flow that it is isonormal and that it will converge to a local minimum. Hence, (12) is a system of nonlinear differential equations that will converge to the eigenvector corresponding to the minimal eigenvalue of the matrix C. This flow can also be interpreted as a special case of Brockett's [5] double bracket flow (see [19] for this interpretation).

3.5 THE CONTINUOUS-TIME POWER METHOD FOR THE LARGEST SINGULAR VALUE

The preceeding observations can also be used to derive a system of nonlinear vector differential equations that converges to the *largest* singular value of a matrix $A \in \mathbb{R}^{p\times q}$, $p \geq q$. It suffices to choose in (12)

$$C = \begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix} \quad \text{and} \quad x = \begin{pmatrix} u \\ v \end{pmatrix} \,. \tag{13}$$

This matrix C will have q eigenvalues given by $\sigma_i(A)$, q eigenvalues equal to $-\sigma_i(A)$ and $p-q$ eigenvalues equal to 0. Hence the smallest eigenvalue is $-\sigma_{\max}(A)$. The first q components of x will go to the corresponding left singular vector u while its last q components converge to the right singular vector v.

3.6 THE 'DISCRETE-TIME' POWER METHOD

The surprising fact about the nonlinear differential equation (12) is that it can be solved analytically. Its solution is

$$x(t) = e^{-Ct}x(0)/\|e^{-Ct}x(0)\| \,, \tag{14}$$

which can be verified by direct substitution. If we consider the analytic solution at integer times $t = k = 0, 1, 2, \ldots$, we see that

$$x(k+1) = e^{-C}x(k)/\|e^{-C}x(k)\| \,,$$

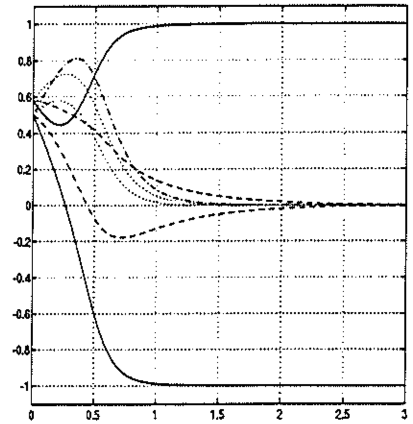

Figure 1: As a function of time, we show the convergence of the elements of the vectors $u(t)$ and $v(t)$ of the differential equation (12) with C given by (13), where A is a 4×3 'diagonal' matrix with on its diagonal the numbers $(5,3,1)$. The initial vector $z_0 = (u(0)^T\, v(0)^T)^T$ is random. This differential system converges to the eigenvector corresponding to the smallest eigenvalue of C in (13), which is $-\sigma_{\max}(A) = -5$. Hence, this flow upon convergence delivers the 'largest' singular triplet of the matrix A. The picture was generated using the Matlab numerical integration function 'ode45'.

which shows that the continuous time equation (12) interpolates the discrete time power method [4] for the matrix e^{-C}. This implies that we now have the clue to understand the *global* convergence behavior of the flow (12). Obviously, the stationary points (points where $\dot{x} = 0$) are the eigenvectors of C, but there is only one stationary point that is stable (as could be shown by linearizing (12) around all the stationary points and calculating the eigenvalues of the linearized system). It will always convergence to the eigenvector corresponding to the smallest eigenvalue of C, except when the initial vector $x(0)$ is orthogonal to the smallest eigenvector. These observations are not too surprising since the solution for the linear part in (12) (the first term of the right hand side) is $x(t) = \exp(-Ct)x(0)$. The second term is a normalization term which at each time instant projects the solution of the linear part back to the unit sphere. We can rewrite equation (12) as

$$\dot{x} = (I - \frac{xx^T}{x^Tx})Cx \,.$$

This clearly shows how $\dot{x}$ is obtained from the orthogonal projection of the vector Cx (which is the gradient of the unconstrained object function $0.5x^TCx$) onto the hyperplane which is tangent to the unit sphere at x.

[4]...which is exactly the reason why (12) is called *the continuous-time power method.*

3.7 A DIFFERENTIAL EQUATION DRIVEN BY THE RESIDUAL

Another interesting interpretation is the following: Define the residual vector $r(t) = Cx - x(x^TCx/x^Tx)$. Then the differential equation (12) reads

$$\dot{x} = -r(t) .$$

Hence, the differential equation is driven by a residual error and when $r(t) = 0$, we also have $\dot{x} = 0$ so that a zero residual results in a stationary point. Moreover, using a little known fact in numerical anaylsis [24, p.69] we can give a *backward error interpretation* as follows. Define the rank one matrix $M(t) = r(t).x(t)^T$. Then we easily see that

$$(C - M(t))x(t) = x(t)\lambda(t) \text{ with } \lambda(t) = (x(t))^TCx(t)/(x(t)^Tx(t)) .$$

The interpretation is that, at any time t, the real number $\lambda(t)$ is the *exact* eigenvalue of a *modified* matrix, namely $C - M(t)$. The norm of the modification is given by $\|M(t)\| = \|r(t)\|$, which is the norm of the residual vector and from the convergence, we know that $\|M(t)\| \to 0$ as $t \to \infty$.

3.8 DERIVATION AS A GRADIENT FLOW

Gradient flows on manifolds can be used to solve constrained optimization problems. The idea is to consider the feasible set (i.e. the set of vectors that satisfies the constraints) as a manifold, and then by picking out an appropriate Riemannian metric, determining a gradient flow.

Let us illustrate this by deriving a gradient flow to solve the constrained minimization problem (10). The set of vectors that satisfies the constraints is the unit sphere, which is known to be a manifold:

$$M = \{ x \in \mathbb{R}^p \mid x^Tx = 1 \} .$$

The tangent space at x is given by the vectors z that belong to

$$T_xM = \{ z \in \mathbb{R}^p \mid z^Tx = 0 \} .$$

The directional derivative $D_xg(z)$ is the amount by which the object function changes when moving in the direction z of a vector in the tangent space:

$$D_xg(z) = x^TCz .$$

Next, we can choose a Riemannian metric represented by a smooth matrix function $W(x)$ which is positive definite for all $x \in M$. It is well known (see e.g. [18] that, given the metric $W(x)$, the gradient ∇g can be uniquely determined from two conditions:

1 Compatibility: $D_xg(z) = z^TCx = z^TW(x)\nabla g$.

2 Tangency: $\nabla g \in T_xM \iff x^T\nabla g = 0$.

The unique solution to these two equations applied to the constrained optimization problem (10) results in the gradient given by

$$\nabla g(x) = W(x)^{-1}(Cx - x\,\frac{x^TW(x)^{-1}Cx}{x^TW(x)^{-1}x}) .$$

¿From this we obtain the gradient flow:

$$\dot{x} = -W(x)^{-1}(Cx - x\,\frac{x^TW(x)^{-1}Cx}{x^TW(x)^{-1}x}) . \quad (15)$$

It is easily seen that the stationary points of this system must be eigenvector-eigenvalues of the matrix C. Convergence is guaranteed because one can easily find a Lyapunov function (essentially the norm of the gradient) using the chain rule (see e.g. [18] for details). Observe that the norm $\|x(t)\|$ is constant for all t. This can be seen from

$$\frac{1}{2}\frac{d\|x\|^2}{dt} = x^T\dot{x} = -x^TW(x)^{-1}Cx + x^TW(x)^{-1}Cx = 0 .$$

Hence, if $\|x(0)\| = 1$, we have $\|x(t)\| = 1, \forall t > 0$. If we choose the Euclidean metric, $W(x) = I_n$, we obtain the continuous-time power method (12). An interesting open problem is how to chose the metric $W(x)$ such that for instance the convergence speed could be increased and whether a choice for $W(x)$ other than I_p leads to new iterative discrete-time algorithms.

3.9 NON-SYMMETRIC CONTINUOUS TIME POWER METHOD

So far we have only considered symmetric matrices. However most of the results still hold true *mutatis mutandis* if C is a non-symmetric matrix. For instance, the analytic solution to (12) is still given by (14) even if C is nonsymmetric. For the convergence proof, we can find a Lyapunov function using the left and right eigenvectors of the non-symmetric matrix C. Let $X \in \mathbb{R}^{p\times p}$ be the matrix of right eigenvectors of C while Y is the matrix of left eigenvectors, normalized such that

$$AX = X\Lambda , \quad Y^TX = I_p ,$$
$$A^TY = Y\Lambda , \quad XY^T = I_p .$$

For simplicity we assume that all the eigenvalues of C are real (although this is not really a restriction). The vector $y_{\min} \in \mathbb{R}^p$ is the left eigenvector corresponding to the smallest eigenvalue. It can be shown that the scalar function

$$L(x) = (x^Ty_{\min})^2 \,/\, (x^TYY^Tx)$$

is a Lyapunov function for the differential equations (12) because $\dot{L} > 0, \forall t$. Note that $x = X(Y^Tx)$, which says that the vector Y^Tx contains the components of x with respect to the basis generated by the column vectors in X (which are the right eigenvectors). The denominator is just the norm squared of this vector of components. The numerator is the component of x along the last eigenvector (last column of X), which corresponds to the smallest eigenvalue. Since $\dot{L} > 0$, this component grows larger and larger relative to all other ones, which proves convergence to the 'smallest' eigenvector.

3.10 NON-SYMMETRIC CONTINUOUS-TIME POWER METHOD FOR THE SMALLEST SINGULAR VALUE

We can exploit these insights to calculate the smallest singular triplet of a matrix $A \in \mathbb{R}^{p\times q}$ as follows. Consider the 'asymmetric' continuous-time power method:

$$\begin{pmatrix} \dot u \\ \dot v \end{pmatrix} = -\begin{pmatrix} \alpha I_p & -A \\ A^T & -\alpha I_q \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} u \\ v \end{pmatrix}\sigma\,, \qquad (16)$$

with

$$\sigma = \alpha(u^Tu - v^Tv)\;/\;(u^Tu + v^Tv)\,.$$

Here α is a user-defined real scalar. When it is chosen such that $\alpha > \sigma_{\min}(A)$, u and v will converge to the left resp. right singular vector corresponding to the smallest singular value of A. This can be understood by realizing that the $2q$ (we assume that $p \geq q$) eigenvalues of the matrix

$$\begin{pmatrix} \alpha I_p & -A \\ A^T & -\alpha I_q \end{pmatrix}$$

are $\lambda_i = \pm\sqrt{\alpha^2 - \sigma_i^2(A)}$ and that there are $p-q$ eigenvalues equal to α. Hence, if $\alpha > \sigma_i$, the corresponding eigenvalues λ_i are real, else they are pure imaginary. So if $\alpha > \sigma_{\min}$, the smallest eigenvalue is $\lambda = -\sqrt{\alpha^2 - \sigma^2_{\min}}$ to which (16) will converge. This is illustrated in Figure 2.

3.11 CHRISTIAAN'S FLOW FOR THE SMALLEST SINGULAR VALUE

The problem with the asymmetric continuous time power method (16) is that we have to know a scalar α that is an upper bound to the smallest singular value. Here we propose a new continuous-time algorithm for which we have *strong indications* that it always converges to the 'smallest' singular value, but for which there is no formal proof of convergence yet. Let $A \in \mathbb{R}^{p\times q}$ ($p \geq q$) and $\phi \in \mathbb{R}$ be a given (user-defined) scalar satisfying $0 < \phi < 1$. Consider the following system of differential equations, which we call *Christiaan's flow* [5]:

$$\begin{pmatrix} \dot u \\ \dot v \end{pmatrix} = -\begin{pmatrix} \sigma I_p & -A \\ \phi A^T & -\sigma\phi I_q \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix} \quad \text{with } \sigma = (u^TAv)/(u^Tu)\,. \qquad (17)$$

It is our experience that $(u(t), \sigma(t), v(t))$ converges to the smallest singular triplet of A. The convergence behavior can be influenced by ϕ (for instance, when $\phi \to 1$, there are many oscillations (bad for numerical integration) but convergence is quite fast in time; when $\phi \to 0$, there are no oscillations but convergence is slow in time).
There are many similarities with the continuous-time algorithms discussed so far. When we rewrite these equations as

$$\begin{aligned} \dot u &= Av - u\sigma\,, \\ \dot v &= -\phi(A^Tu - v\sigma)\,, \end{aligned} \qquad (18)$$

[5] ... after one of our PhD students Christiaan Moons who one day just tried it out and to his surpise found out that it always converges.

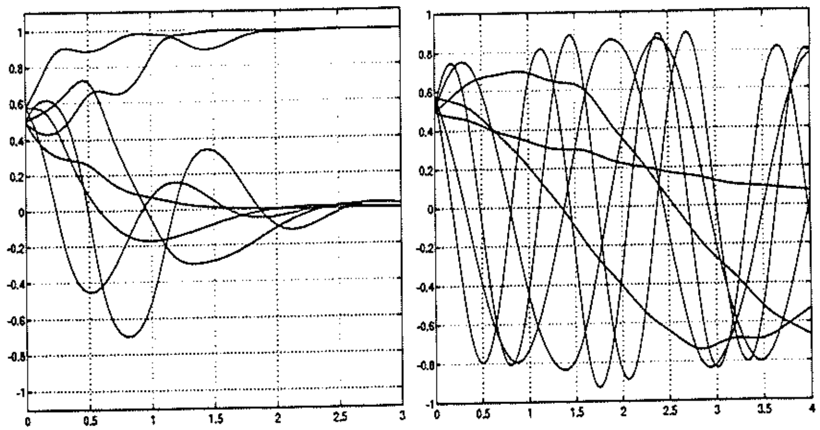

Figure 2: Convergence as a function of time of the components of $u(t)$ and $v(t)$ of the asymmetric continuous-time power method (16) for the same matrix A and the same random initial vectors $u(0)$ and $v(0)$ as in the previous Figure. The left picture shows the behavior for $\alpha = 2$, which is larger than the smallest singular value and therefore converges. The right picture shows the dynamic behavior for $\alpha = 0.5$. Now, all the eigenvalues of the system matrix are pure imaginary. Therefore, there is no convergence but instead, there is an oscillatory regime.

we easily see from (1) that both equations are 'driven' by the (scaled) residual error (see Section 3.7). Another intriguing connection is seen by rewriting (17) as

$$\begin{pmatrix} \dot u \\ \dot v \end{pmatrix} = \begin{pmatrix} -I & 0 \\ 0 & \phi I \end{pmatrix}[-\begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} u \\ v \end{pmatrix}\sigma]$$

which compares very well to (15), except that the metric here is indefinite!
As for a formal convergence proof, there are the following facts: It is readily verified that the singular triplets of A are the stationary points of the flow. When we linearize the system around these stationary points, it is readily verified that they are all unstable, except for the stationary point corresponding to the 'smallest' singular triplet of A. However, at this moment of writing, we do not have a formal proof of convergence (for instance a Lyapunov function that proves stability) [6].

4 ALGORITHMS FOR THE RIEMANNIAN SVD

In this section, we'll try to use the ideas presented in the previous section to come up with continuous time algorithms for the Riemannian SVD (4) and hence for structured/weighted total least squares problems. The ideas presented here may be premature but on the other

[6] ...and as we have done before, we offer a chique dinner in the most exquis restaurant of Leuven if somebody solves our problem.

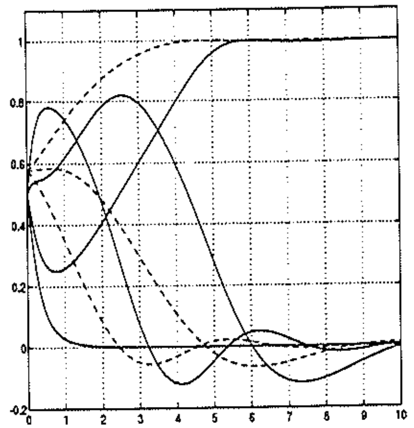

Figure 3: Example of the convergence behavior of Christiaan's flow (17) for the same matrix A and initial vectors $u(0)$ and $v(0)$ as in the first Figure. The full lines are the components of u, the dashed ones those of v. Both vectors converge asymptotically to the left and right singular vectors of A corresponding to the smallest singular value.

hand, they offer intriguing and challenging perspectives, which is the reason why we mention them. It should be noted that we have derived a heuristic iterative discrete-time algorithm for the Riemannian SVD in [12], the basic inspiration of which is the classical power method. While this algorithm works well in most cases, there is no formal proof of convergence nor is there any guarantee that it will convergence to (an at least local) minimum.
Therefore, we try to go via these continuous-time algorithms to find an approach which would be *guaranteed* to *converge* to a local minimum. So far we have not succeeded in doing so, but we hope that the elements presented in this section provide enough new material to convince the reader of the usefulness of the presented approach.

4.1 AN OPTIMIZATION PROBLEM

Let us try to solve (see the equations (4):

$$\sigma = \min_{u,v} \frac{u^T A v}{u^T D_v u} = \min_{u,v} \frac{v^T A^T u}{v^T D_u v} .$$

A nice property which can often be used in manipulating formulas in this framework, is that for every vector u and v, we always have $u^T D_v u = v^T D_u v$ (see [12] for a proof).
The fact that D_u is independent of v and D_v is independent of u allows us to apply the continous-time algorithm (5) and derive the system of differential equations

$$\begin{aligned} \dot{u} &= -\frac{d\sigma}{du} = \frac{1}{u^T D_v u}(-Av + D_v u \frac{u^T A v}{u^T D_v u}) , \\ \dot{v} &= -\frac{d\sigma}{dv} = \frac{1}{v^T D_u v}(-A^T u + D_u v \frac{v^T A^T u}{v^T D_u v}) . \end{aligned}$$

While this system will converge to a local minimum, it is not this local minimum that solves our structured/weighted TLS problem. Indeed, to see why, it suffices to consider the special case where $D_v = I_p$ and $D_u = I_q$ in which case we recover the system (12)-(13) which converges to $-\sigma_{\max}(A)$ and the corresponding singular vectors (and NOT $\sigma_{\min}(A)$!).

4.2 CHRISTIAAN'S FLOW FOR THE RIEMANNIAN SVD

Although we are completely driving on heuristics here, it turns out that the following generalization of Christiaan's flow (17) works very well to find the minimal singular triplet of the Riemannian SVD (4).

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = -\begin{pmatrix} \sigma D_v & -A \\ \phi A^T & -\sigma\phi D_u \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} \quad \text{with } \sigma = (u^T A v)/(u^T u) , \qquad (19)$$

with $\phi \in \mathbb{R}$ a user-defined number satisfying $0 < \phi < 1$. The only difference between (17) and (19) is the introduction of the positive definite metric matrices D_u and D_v. We have no idea whatsoever about possible convergence properties (that are formally provable), except for the fact that in a stationary point, the necessary conditions (4) are satisfied.

Yet, our numerical experience is that this system of nonlinear differential equations converges to a local minimum of the object function. As an example, let $A \in \mathbb{R}^{6\times 5}$ be a given matrix (we took a random matrix), that will be approximated in Frobeniusnorm by a rank deficient matrix B, by not modifying all of its elements but only those elements that are 'flagged' by a '1' in the following matrix

$$V = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix} \; ; \text{ Let } W = \begin{pmatrix} \infty & 1 & \infty & \infty & \infty \\ 1 & \infty & 1 & \infty & 1 \\ \infty & \infty & 1 & 1 & \infty \\ 1 & 1 & 1 & \infty & 1 \\ \infty & 1 & 1 & 1 & 1 \\ \infty & \infty & 1 & 1 & 1 \end{pmatrix}$$

be the elementwise inverse of V. The matrix W contains the weights as in (3) and the element ∞ means that we impose an infinite weight on the modification of the corresponding element in A (which implies that it will not be modified and that the corresponding element in B will be equal to that in A). It can be shown [11] that the metric matrices D_u and D_v for this flow are diagonal matrices given by

$$D_v = \operatorname{diag}(V \begin{pmatrix} v_1^2 \\ v_2^2 \\ v_3^2 \\ v_4^2 \\ v_5^2 \end{pmatrix}) , \quad D_u = \operatorname{diag}(V^T \begin{pmatrix} u_1^2 \\ u_2^2 \\ u_3^2 \\ u_4^2 \\ u_5^2 \\ u_6^2 \end{pmatrix}) .$$

Here v_i and u_i denote the i-th component of v, resp. u and u_i^2 and v_i^2 are their squares. As an initial vector for Christiaan's flow (19) we took $[u(0)^T \; v(0)^T] = [e_6^T/\sqrt{6} \;\; e_5^T/\sqrt{5}]$ where $e_k \in \mathbb{R}^k$ is a vector with all ones as its components. The resulting behavior as a function of

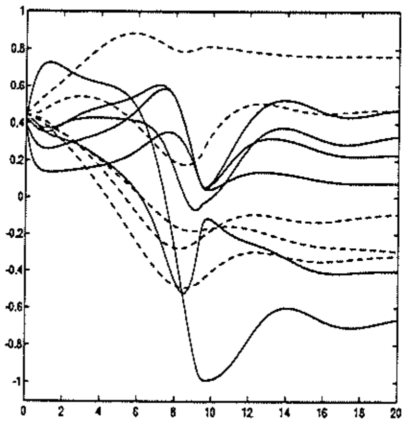

Figure 4: Vectors $u(t)$ and $v(t)$ as a function of time for Christiaan's flow (19), which solves the problem of least squares approximation of a given matrix A by a rank deficient one, while not all of its elements can be modified as specified by the elements in the matrix V which belong to $\{0,1\}$. The vector differential equation converges to the same solution as the one provided by the discrete-time algorithm of [11], which on its turn is inspired by the discrete-time power method.

time is shown in Figure 4.

5 CONCLUSIONS

In this paper, we have discussed how weighted and/or structured total least squares problems lead to a nonlinear generalization of the SVD, which we have called the Riemannian SVD. Next, we have derived several interesting interpretations of the continuous-time power method (geometric, unconstrained and constrained optimization, gradient flow, residual driven differential equation). We have also discussed Christiaan's flow, which is a set of nonlinear differential equations that seems to converge to the 'smallest' singular triplet, both of the SVD and the Riemannian SVD. As of now, there is however no formal proof of convergence.

There are interesting analogies between the SVD and the Riemannian SVD, which are enumerated in the following table:
There are still several open problems, which all together form a complete research program: Proofs of convergence of some of the algorithms we have introduced here, discretization (e.g. based on geodesic approximations (see e.g. [26])), decreasing the rank with more than 1 while preserving the structure, etc .... Also in the applications there are several interesting sideways that could be followed. For instance, in [8] we have shown how our approach leads to an H_2-model reduction algorithm, while in [12] [13] we show how the structured total least squares problem with a 'double' Hankel matrix of noisy inputs and outputs is the

	'Euclidean'	'Riemannian'
Problem	TLS (Eckart-Young)	Weighted/Structured TLS
Applications	Linear relations, noisy data	Hankel matrices Model reduction, Dynamic TLS Relative error TLS Maximum likelihood
Decomposition	SVD of A 'Identity' metric	Riemannian SVD of A Positive definite metric D_u, D_v
Approximation B	Rank 1 modification	Multilinear matrix function
Algorithms	Power method Christiaan's flow	Power method Christiaan's flow

same as the L_2-optimal so-called errors-in-variables problem from system identification.

Acknowledgements

The author is a Senior Research Associate of the Belgian National Fund for Scientific Research. This work was supported by grants from the Federal Ministry of Scientific Policy (DWTC, with grants IUAP/PAI-17 (Modelling and Control of Dynamical Systems), IUAP/PAI-50 (Automation in Design and Production)), the Flemish NFWO-project no. G.0292.95 (Matrix algorithms for adaptive signal processing systems, identification and control) and the SIMONET (System Identification and Modelling Network) supported by the Human Capital and Mobility Program of the European Commision.

I would sincerely like to thank Christiaan Moons, Jeroen Dehaene and Johan Suykens for many lively discussions as well as Thomas De Moor for letting me use his balloon in my attempts to illustrate the equivalence principle of general relativity in Gene Golub's car.

References

[1] Adcock R.J. *Note on the method of least squares.* The Analyst, Vol IV, no.6, Nov. 1877, pp.183-184.

[2] Adcock R.J. *A problem in least squares.* The Analyst, March 1878, Vol V, no.2, pp.53-54.

[3] Baldi P., Hornik K. *Neural networks and principal component analysis: Learning from examples without local minima.* Neural Networks, Vol.2, pp.53-58, 1989.

[4] Bourlard H., Kamp Y. *Auto-association by multilayer perceptrons and singular value decomposition.* Biol. Cybern., 59, pp.291-294, 1988.

[5] Brockett R.W. *Dynamical systems that sort lists and solve linear programming problems.* Linear Algebra and its Applications, 146, pp.79-91, 1991.

[6] M.T. Chu, *On the continuous realization of iterative processes*, SIAM Review, 30, 3, pp.375-387, September 1988.

[7] De Moor B., Golub G.H. *The restricted singular value decomposition: properties and applications.* Siam Journal on Matrix Analysis and Applications, Vol.12, no.3, July 1991, pp.401-425.

[8] De Moor B., Van Overschee P., Schelfhout G. *H_2-model reduction for SISO systems.* Proc. of the 12th World Congress International Federation of Automatic Control, Sydney, Australia, July 18-23 1993, Vol. II pp.227-230. [7]

[9] De Moor B., David J. *Total linear least squares and the algebraic Riccati equation.* System & Control Letters, Volume 18, 5, pp. 329-337, May 1992 [8].

[10] De Moor B. *Structured total least squares and L_2 approximation problems.* Special issue of Linear Algebra and its Applications, on Numerical Linear Algebra Methods in Control, Signals and Systems (eds: Van Dooren, Ammar, Nichols, Mehrmann), Volume 188-189, July 1993, pp.163-207.

[11] De Moor B. *Total least squares for affinely structured matrices and the noisy realization problem* . IEEE Transactions on Signal Processing, Vol.42, no.11, November 1994.

[12] De Moor B. *Dynamic Total Linear Least Squares.* SYSID '94, Proc. of the 10th IFAC Symposium of System Identification, 4-6 July 1994, Copenhagen, Denmark.

[13] De Moor B., Roorda B. *L_2-optimal linear system identification: Dynamic total least squares for SISO systems.* ESAT-SISTA TR 1994-53, Department of Electrical Engineering, Katholieke Universiteit Leuven, Belgium. Accepted for publication in the Proc. of 33rd IEEE CDC, Florida, December 1994.

[14] Eckart C., Young G. *The approximation of one matrix by another of lower rank.* Psychometrika, 1, pp.211-218, 1936.

[15] Fiacco A.V., McCormick G.P. *Nonlinear programming; Sequential unconstrained minimization techniques.* SIAM Classics in Applied Mathematics, 1990.

[16] Golub G.H., Van Loan C. *Matrix Computations.* Johns Hopkins University Press, Baltimore 1989 (2nd edition).

[17] Golub G.H., Van Loan C.F. *An analysis of the total least squares problem.* Siam J. of Numer. Anal., Vol. 17, no.6, December 1980.

[18] Helmke U., Moore J.B. *Optimization and dynamical systems.* CCES, Springer Verlag, London, 1994.

[19] Helmke U. *Isospectral matrix flows for numerical analysis.* This volume.

[7] Reprinted in the Automatic Control World Congress 1993 5-Volume Set, Volume 1: Theory.

[8] Forms also the basic material of a Chapter in "Peter Lancaster, Leiba Rodman. *Algebraic Riccati Equations.* Oxford University Press, 1994.

[20] Householder A.S., Young G. *Matrix approximation and latent roots.* Americ. Math. Monthly, 45, pp.165-171, 1938.

[21] Kung S.Y., Diamantaras K.I., Taur J.S. *Neural networks for extracting pure/constrained/oriented principal components.* Proc. of the 2nd International Workshop on SVD and Signal Processing, June 1990.

[22] Ljung L. *Analysis of recursive stochastic algorithms.* IEEE Transactions on Automatic Control, Vol.AC-22, no.4., August 1977, pp.551-575.

[23] Oja E. *A simplified neuron model as a principal component analyzer.* J. Math. Biology, 15, pp.267-273, 1982.

[24] Parlett B. *The symmetric eigenvalue problem.* Prentice Hall, Englewood Cliffs, NJ, 1980.

[25] Pearson K. *On lines and planes of closest fit to systems of points in space.* Phil. Mag., 2, 6-th series, pp.559-572.

[26] S.T. Smith, *Geometric Optimization Methods for Adaptive Filtering*, PhD Thesis, Harvard University, May 1993 Cambridge, Massachusetts, USA.

[27] Van Huffel S., Vandewalle J. *The Total Least Squares Problem: Computational Aspects and Analysis.* Frontiers in Applied Mathematics 9, SIAM, Philadelphia, 300 pp., 1991.

[28] Young G. *Matrix approximation and subspace fitting.* Psychometrika, vol.2, no.1, March 1937, pp.21-25.

[29] Wilkinson J. *The algebraic eigenvalue problem.* New York, Oxford University Press, 1965.