



TECHNICAL UNIVERSITY OF CRETE

DEPARTMENT OF ELECTRONIC & COMPUTER ENGINEERING

SPARSE RANK-DEFICIENT VARIANCE MAXIMIZATION

DIPLOMA THESIS

BY

MEGASTHENIS M. ASTERIS

SUPERVISED BY:

ASSISTANT PROF. GEORGE N. KARYSTINOS

CHANIA, JULY 26, 2010

Contents

	List of Figures	iv
I	Introduction	1
II	Problem Statement	2
III	Exhaustive search solution - The direct solution	3
IV	Rank-1 case - A trivial case	6
V	Maximization of a Rank-deficient Quadratic Form with a Real Vector Argument under a cardinality constraint	8
VI	Algorithmic Developments	22
	VI.1 Rank-1 Case	23
	VI.2 Rank-2 Case	23
	VI.2.A Parallel Implementation	24
	VI.2.B Serial Implementation	26
	VI.2.C On the complexity of the two implementations	30
	VI.3 Rank-3 Case	32
	VI.3.A Parallel Implementation	34
	VI.3.B Serial Implementation	40

VI.3.C	On the complexity of the two implementations	48
VII	Conclusions	51
Appendix		54
A	Proof of Proposition 1	54

List of Figures

1	Example: Rank-2, Cells and regions in the Φ^1 field.	12
2	Example: Rank-3, $S(\cdot; \cdot)$ hypersurfaces.	16
3	Example: Rank-3, Partition of the Φ^2 plane.	21
4	Rank-2: Intersection of two curves - Ambiguity Resolution.	26
5	Rank-2: Serial scanning of the Φ^1 field.	27
6	Rank-2: Serial Algorithm Execution Instance.	29
7	Rank-2: Algorithm Complexity	31
8	Rank-2: Points examined, Regions, Distinct I -sets.	32
9	Rank-3: Hypersurfaces described by the elements of $ \mathbf{Vc}(\phi_{1:2}) $ and their intersection.	33
10	Rank-3: Intersection of two hypersurfaces.	35
11	Rank-3: Intersection point - Ambiguity resolution.	39
12	Rank-3: Serial, At intersection point - Special Case	44
13	Rank-3: Serial, At intersection point - Case A	45
14	Rank-3: Serial, At intersection point - Case B	46
15	Rank-3: Serial, At intersection point - Non-Existent Case.	47

16	Rank-3: Serial Algorithm Execution Instance.	48
17	Rank-3: Algorithm Complexity	50
18	Rank-3: Points examined, Regions, Distinct I -sets.	50

Abstract

We consider the problem of the maximization of a positive (semi)definite quadratic form that consists of matrix parameter and a sparse vector argument. The complexity of such an optimization problem is determined by the characteristics of the matrix parameter, as well as the alphabet of the vector argument. If, for example, the elements of the vector argument belong to the real alphabet and there is no sparsity constraint, the optimal solution equals the maximum-eigenvalue eigenvector of the matrix parameter and is obtained in polynomial time.

Complexity increases, however, if an additional, cardinality constraint (limitation on the maximum number of nonzero elements) is imposed on the vector argument, since one would have to determine the indices of nonzero elements in the optimal solution. Once the optimal index-set for the nonzero elements of the vector argument is determined, the optimization problem is reduced to a lower-dimensionality instance of the previous unconstrained problem.

In the present work we seek a way to exploit rank deficiency of the matrix parameter, in order to determine a, worst-case, polynomial - in terms of rank - size set of candidate index-sets. By introducing auxiliary spherical coordinates, we show that the multidimensional space is partitioned into a polynomial-size set of regions, each associated with one index-set. Therefore, we can collect all candidate index-sets and numerically determine the optimal index-set among them, avoiding exhaustive search among all possible index-sets. Finally, we implement serial and parallel algorithms for the efficient collection of the candidate index-sets when the rank of the quadratic form equals 2 or 3.

I Introduction

The maximization of a positive (semi)definite quadratic form that consists of a matrix parameter and a vector argument is a common design problem. The complexity of such an optimization depends both on the characteristics of the matrix parameter (whose rank determines the rank of the quadratic form) as well as the constraints imposed on the vector argument. For example, if the elements of the vector argument belong to the real alphabet and only a norm constraint is imposed on the same vector, then the quadratic form is maximized by the principal eigenvector of the matrix parameter (appropriately scaled to meet the norm constraint) and the solution of the maximization problem is trivial.

Additional constraints may dramatically increase the complexity of the optimization problem. In this work, we confront the case where, in addition to the norm constraint, a cardinality constraint is imposed on the vector argument, i.e. a limitation on the number of its nonzero elements. In other words, we are seeking a sparse solution for the quadratic maximization problem. In order to gain some intuition into the complexity of the new problem, one can see that if N is the dimensionality of the vector argument and K is the (maximum) number of nonzero elements allowed in the solution, there are $\binom{N}{K}$ possible index-sets for the nonzero elements of the vector argument. Once the optimal index-set is determined, the maximization problem is once again trivial, since it is equivalent to a lower-dimension quadratic form maximization problem, free of cardinality constraints. Determining the optimal index-set for the nonzero elements of the vector argument, however, is not that trivial and exhaustive search among all possible index-sets can be prohibitive even for relatively small values of N and K . In fact, the problem in the general case is NP-Hard and therefore most research focuses on the relaxation of the problem through replacement of the hard cardinality constraint and seeks for approximate solutions. In [6] the hard cardinality was relaxed to obtain a convex approximation, solved using semi-definite programming. In [5] an alternative approach was pursued, using greedy search and branch-and-bound methods to solve small instances of problem exactly and get good solutions for larger ones.

In this work we focus on determining the optimal solution and show that if the matrix parameter is rank deficient, then exhaustive search among all candidate index-sets can be avoided. In fact we show that size of the set of all candidate index-sets can be polynomially, in terms of rank, bounded. In the following we present the theoretic developments

for the existence of this polynomial-size set of candidate index-sets. Then, we present our algorithmic developments for the efficient construction of the latter, for the cases where the rank of the quadratic form is equal to 2 or 3.

Notation: Vectors and matrices are denoted by small and capital, respectively, bold letters. For example \mathbf{x} is a vector and \mathbf{A} is a matrix. Their elements are denoted as x_i and $A_{i,j}$. Furthermore, if I is a set of indices (positive integers in the range of \mathbf{x}), \mathbf{x}_I denotes the sub-vector of \mathbf{x} containing only the elements of \mathbf{x} indexed by the elements of I in ascending order and \mathbf{A}_I denotes the sub-matrix of \mathbf{A} containing only the rows and columns of \mathbf{A} indexed by the elements of I in ascending order.¹ Finally, when it comes handy, a Matlab-like notation may be used, where $\mathbf{A}_{i:j,k:l}$ denotes the sub-matrix of \mathbf{A} that consists of the i -th up to j -th rows and k -th up to l -th columns of it.

II Problem Statement

We consider the quadratic form

$$\mathbf{x}^T \mathbf{A} \mathbf{x}, \quad (1)$$

where $\mathbf{A} \in \mathbb{R}^{N \times N}$ is a symmetric matrix and $\mathbf{x} \in \mathbb{R}^N$ is a real vector argument. Since \mathbf{A} is symmetric, it can be decomposed as

$$\mathbf{A} = \sum_{n=1}^N \lambda_n \mathbf{q}_n \mathbf{q}_n^T, \quad (2)$$

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N, \quad \|\mathbf{q}_n\| = 1, \quad \mathbf{q}_n^T \mathbf{q}_k^T = 0, \quad n \neq k, \quad n, k = 1, 2, \dots, N,$$

where λ_n and \mathbf{q}_n are its n -th eigenvalue and eigenvector, respectively. We are interested in the computation of the real, unit-length vector \mathbf{x} that has at most K nonzero elements

¹To clarify these notations, if I is a set of K positive integers, then

$$\mathbf{x}_I = \begin{pmatrix} x_{i_1} \\ x_{i_2} \\ \vdots \\ x_{i_K} \end{pmatrix} \quad \text{and} \quad \mathbf{A}_I = \begin{pmatrix} A_{i_1, i_1} & A_{i_1, i_2} & \cdots & A_{i_1, i_K} \\ A_{i_2, i_1} & A_{i_2, i_2} & \cdots & A_{i_2, i_K} \\ \vdots & & \ddots & \vdots \\ A_{i_K, i_1} & A_{i_K, i_2} & \cdots & A_{i_K, i_K} \end{pmatrix},$$

where $i_1, i_2, \dots, i_K \in I$ and $i_1 < i_2 < \dots < i_K$.

and maximizes the quadratic form

$$\mathbf{x}_{\text{opt}} \triangleq \arg \max_{\substack{\mathbf{x} \in \mathbb{R}^N \\ \|\mathbf{x}\| = 1 \\ \|\mathbf{x}\|_0 \leq K}} \mathbf{x}^T \mathbf{A} \mathbf{x}. \quad (3)$$

Without loss of generality (w.l.o.g.) we assume that $\lambda_N = 0$. Indeed, if $\lambda_N \neq 0$, then \mathbf{A} can be substituted by $\mathbf{A} - \lambda_N \mathbf{I}$ so that the quadratic forms $\mathbf{x}^T (\mathbf{A} - \lambda_N \mathbf{I}) \mathbf{x} = \mathbf{x}^T \mathbf{A} \mathbf{x} - N \lambda_N$ and $\mathbf{x}^T \mathbf{A} \mathbf{x}$ are maximized by the same vector \mathbf{x} and the minimum eigenvalue of $\mathbf{A} - \lambda_N \mathbf{I}$ equals zero. Therefore, in the following, w.l.o.g. we assume that \mathbf{A} is semidefinite positive with rank $D \leq N - 1$, i.e.:

$$\mathbf{A} = \sum_{n=1}^D \lambda_n \mathbf{q}_n \mathbf{q}_n^T, \quad \lambda_1 \geq \lambda_2 \geq \dots \lambda_D \geq 0. \quad (4)$$

Furthermore, since $\lambda_n > 0$, $n = 1, 2, \dots, D$, we can define the weighted principal component \mathbf{v}_n as:

$$\mathbf{v}_n \triangleq \sqrt{\lambda_n} \mathbf{q}_n, \quad n = 1, 2, \dots, D \quad (5)$$

and the corresponding $N \times D$ matrix

$$\mathbf{V} \triangleq [\mathbf{v}_1 \ \mathbf{v}_2 \ \dots \ \mathbf{v}_D]. \quad (6)$$

Matrix \mathbf{V} is full-rank and has the same rank D as matrix \mathbf{A} ($D \leq N - 1$). Obviously, $\mathbf{A} = \mathbf{V} \mathbf{V}^T$ and problem (3) can be written as:

$$\mathbf{x}_{\text{opt}} \triangleq \arg \max_{\substack{\mathbf{x} \in \mathbb{R}^N \\ \|\mathbf{x}\| = 1 \\ \|\mathbf{x}\|_0 \leq K}} \mathbf{x}^T \mathbf{V} \mathbf{V}^T \mathbf{x}. \quad (7)$$

III Exhaustive search solution - The direct solution

If we omit the zero-norm constraint in (3), the maximization problem is trivial: \mathbf{x}_{opt} is the principal eigenvector of \mathbf{A} , appropriately scaled to meet the norm constraint. Since λ_1 is

the largest eigenvalue of \mathbf{A} and \mathbf{q}_1 is the corresponding eigenvector as defined in (2), \mathbf{x}_{opt} is:

$$\mathbf{x}_{\text{opt}} = \mathbf{q}_1 = \frac{\mathbf{v}_1}{\|\mathbf{v}_1\|}. \quad (8)$$

The zero-norm constraint requires that \mathbf{x}_{opt} has at most K ($\leq N$) nonzero elements. The contribution of the zero elements of \mathbf{x} in the value of $\mathbf{x}^T \mathbf{A} \mathbf{x}$ is zero and, therefore, the corresponding rows and columns of matrix \mathbf{A} can be ignored.² Let I_{opt} be the set of indices of the nonzero elements of \mathbf{x}_{opt} . (Apparently, $|I_{\text{opt}}| = K$). Then,

$$\begin{aligned} \max_{\substack{\mathbf{x} \in \mathbb{R}^N \\ \|\mathbf{x}\| = 1 \\ \|\mathbf{x}\|_0 \leq K}} \mathbf{x}^T \mathbf{A} \mathbf{x} &= \max_{\substack{\mathbf{y} \in \mathbb{R}^K \\ \|\mathbf{y}\| = 1}} \mathbf{y}^T \mathbf{A}_{I_{\text{opt}}} \mathbf{y}, \end{aligned} \quad (9)$$

where $\mathbf{y} \in \mathbb{R}^K$ and $\mathbf{A}_{I_{\text{opt}}}$ is a the submatrix of \mathbf{A} containing only the rows and columns of \mathbf{A} that correspond to the nonzero elements of \mathbf{x}_{opt} . The vector \mathbf{y}_{opt} that achieves the maximum value of $\mathbf{y}^T \mathbf{A}_{I_{\text{opt}}} \mathbf{y}$ is:

$$\mathbf{y}_{\text{opt}} = \frac{(\mathbf{a}_{I_{\text{opt}}})_1}{\|(\mathbf{a}_{I_{\text{opt}}})_1\|}, \quad (10)$$

where $(\mathbf{a}_{I_{\text{opt}}})_1$ is the weighted principal eigenvector of matrix $\mathbf{A}_{I_{\text{opt}}}$. Once \mathbf{y}_{opt} is determined, the optimal sparse solution of the original problem, \mathbf{x}_{opt} , can be obtained by expanding vector \mathbf{y}_{opt} and inserting zeros at the appropriate positions:

$$\mathbf{x}_{\text{opt}} \in \mathbb{R}^N, \quad \begin{cases} \mathbf{x}_{\text{opt}_{I_{\text{opt}}}} = \mathbf{y}_{\text{opt}} & (\in \mathbb{R}^K), \\ \mathbf{x}_{\text{opt}_{\overline{I_{\text{opt}}}}} = \mathbf{0}_{(N-K) \times 1} & (\in \mathbb{R}^{N-K}) \end{cases}, \quad (11)$$

where $\overline{I_{\text{opt}}} = \{i : i \in \{1, \dots, N\} - I_{\text{opt}}\}$ i.e. $\overline{I_{\text{opt}}}$ is a $(N - K)$ -size subset of $\{1, \dots, N\}$ containing all elements that do not belong to I_{opt} .

²When the i -th element of $\mathbf{x} \in \mathbb{R}^N$ is 0, then $\|\mathbf{x}^T \mathbf{A} \mathbf{x}\|$ equals $\|\hat{\mathbf{x}}^T \hat{\mathbf{A}} \hat{\mathbf{x}}\|$ where $\hat{\mathbf{x}}$ is the $(N - 1) \times 1$ vector that derives from \mathbf{x} if we omit the i -th element and $\hat{\mathbf{A}}$ is the $(N - 1) \times (N - 1)$ matrix that derives from \mathbf{A} if we omit its i -th row and i -th column of the latter:

$$\begin{pmatrix} \vdots \\ x_{i-1} \\ 0 \\ x_{i+1} \\ \vdots \end{pmatrix}^T \begin{pmatrix} \ddots & \vdots & & \vdots & \\ \cdots & A_{i-1,i-1} & A_{i-1,i} & A_{i-1,i+1} & \cdots \\ \cdots & A_{i,i-1} & A_{i,i} & A_{i,i+1} & \cdots \\ \cdots & A_{i+1,i-1} & A_{i+1,i} & A_{i+1,i+1} & \cdots \\ \vdots & \vdots & \vdots & \ddots & \end{pmatrix} \begin{pmatrix} \vdots \\ x_{i-1} \\ 0 \\ x_{i+1} \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ x_{i-1} \\ x_{i+1} \\ \vdots \end{pmatrix}^T \begin{pmatrix} \ddots & \vdots & \vdots & \\ \cdots & A_{i-1,i-1} & A_{i-1,i+1} & \cdots \\ \cdots & A_{i+1,i-1} & A_{i+1,i+1} & \cdots \\ \vdots & \vdots & \ddots & \end{pmatrix} \begin{pmatrix} \vdots \\ x_{i-1} \\ x_{i+1} \\ \vdots \end{pmatrix}.$$

From the above, it is obvious that the calculation of \mathbf{x}_{opt} , given the optimal set of indices of its nonzero elements, is trivial. Therefore, the problem is shifted to determining the optimal index-set I_{opt} . The most obvious way for discovering I_{opt} is exhaustive search among all possible index-sets. In other words, one may calculate the value of

$$m(I) \triangleq \max_{\substack{\mathbf{y} \in \mathbb{R}^K \\ \|\mathbf{y}\| = 1}} \mathbf{y}^T \mathbf{A}_I \mathbf{y}, \quad (12)$$

for every possible index-set, I , and then determine the optimal set I_{opt} as that set for which the maximum value in (12) is achieved:

$$I_{\text{opt}} = \arg \max_I \{m(I)\}, \quad (13)$$

Since I is a set of K elements and \mathbf{x} belongs to \mathbb{R}^N , there are $\binom{N}{K}$ possible index-sets for the nonzero elements of \mathbf{x}_{opt} . The total complexity of the exhaustive method is therefore:

$$\mathcal{C}_{\text{exh}} = \binom{N}{K} \times \mathcal{C}_{\text{eigv in } \mathbb{R}^{K \times K}}, \quad (14)$$

where $\mathcal{C}_{\text{eigv in } \mathbb{R}^{K \times K}}$ is the cost of calculating the principal eigenvalue of a $K \times K$ matrix. Note that for each set I we are only interested in the calculation of the principal eigenvalue of \mathbf{A}_I since the latter corresponds to the maximum value achieved by the quadratic form for a given set I , under the unit-norm constraint. The principal eigenvector only has to be calculated for $\mathbf{A}_{I_{\text{opt}}}$ in order to construct \mathbf{x}_{opt} .

In this work, we seek a way to reduce \mathcal{C}_{exh} by avoiding the exhaustive search among all possible index-sets. In the next sections, we will show that if matrix \mathbf{A} is rank deficient, then we can exploit rank-deficiency to reduce the number of candidate index-sets. Specifically, when $\mathcal{R}(\mathbf{A}) = D$, we introduce $D - 1$ auxiliary spherical coordinates and show that there exists a set $\mathcal{I}(\mathbf{V}_{N \times D})$ of candidate index-sets, whose size is polynomially, in terms of rank, upper-bounded and contains the optimal index-set, I_{opt} . We also present a fully developed algorithm for the construction of $\mathcal{I}(\mathbf{V}_{N \times D})$ when the rank equals 2 or 3. Before we proceed, we present the special, widely known case of $\mathcal{R}(A) = 1$. Although the rank-1 solution is rather trivial and cannot apply directly to higher-rank cases, its simplicity will be a strong motivation for seeking a solution other than exhaustive search for higher ranks as well.

IV Rank-1 case - A trivial case

In the case where the input matrix \mathbf{A} is of rank-1, it can be written as $\mathbf{A} = \mathbf{v}\mathbf{v}^T$ (i.e. matrix \mathbf{V} reduces to a vector in \mathbb{R}^N). Our optimization problem becomes:

$$\begin{aligned} \mathbf{x}_{\text{opt}} &\triangleq \arg \max_{\substack{\mathbf{x} \in \mathbb{R}^N \\ \|\mathbf{x}\| = 1 \\ \|\mathbf{x}\|_0 \leq K}} \mathbf{x}^T \mathbf{v}\mathbf{v}^T \mathbf{x} \\ &= \arg \max_{\substack{\mathbf{x} \in \mathbb{R}^N \\ \|\mathbf{x}\| = 1 \\ \|\mathbf{x}\|_0 \leq K}} \|\mathbf{v}^T \mathbf{x}\|. \end{aligned} \quad (15)$$

In order to incorporate the concept of the index-set I into the maximization problem, we can write:

$$\max_{\substack{\mathbf{x} \in \mathbb{R}^N \\ \|\mathbf{x}\| = 1 \\ \|\mathbf{x}\|_0 \leq K}} \|\mathbf{v}^T \mathbf{x}\| = \max_{\substack{\mathbf{x} \in \mathbb{R}^N \\ \|\mathbf{x}\| = 1 \\ \|\mathbf{x}\|_0 \leq K}} |\mathbf{v}^T \mathbf{x}| = \underbrace{\left(\max_I \max_{\substack{\sum_{i \in I} x_i^2 = 1 \\ x_i = 0 \ \forall i \notin I}} \left| \sum_{n=1}^N x_n v_n \right| \right)}_{Q(I)},$$

where I is the set the K indices of the nonzero elements of \mathbf{x} . For any given index-set I , we have:

$$\begin{aligned} Q(I) &= \max_{\substack{\sum_{i \in I} x_i^2 = 1 \\ x_i = 0 \ \forall i \notin I}} \left| \sum_{n=1}^N x_n v_n \right| \\ &= \max_{\sum_{i \in I} x_i^2 = 1} \left| \sum_{i \in I} (x_i \cdot v_i) + \sum_{i \notin I} (0 \cdot v_i) \right| \\ &= \max_{\sum_{i \in I} x_i^2 = 1} \left| \sum_{i \in I} (x_i \cdot v_i) \right| \\ &= \max_{\|\mathbf{x}_I\| = 1} |\mathbf{v}_I^T \mathbf{x}_I| \\ &\stackrel{(a)}{=} \|\mathbf{v}_I\|, \end{aligned}$$

where (a) derives from Cauchy-Swartz inequality in which equality is achieved for:

$$\mathbf{x}_I = \frac{\mathbf{v}_I}{\|\mathbf{v}_I\|}. \quad (16)$$

The optimal index-set, I_{opt} , is, by definition, the index-set for which the maximum value of $Q(I)$ is achieved:

$$\begin{aligned}
I_{\text{opt}} &= \arg \max_I Q(I) \\
&= \arg \max_I \|\mathbf{v}_I\| \\
&= \arg \max_I \sqrt{\sum_{i \in I} v_i^2} \\
&= \arg \max_I \sum_{i \in I} |v_i|. \tag{17}
\end{aligned}$$

From (17) we conclude that the optimal index-set for the nonzero elements of \mathbf{x}_{opt} is the set of indices of the K largest elements of $|\mathbf{v}|$. Therefore, in the case $\mathcal{R}(\mathbf{A}) = 1$, not only can exhaustive search among the $\binom{N}{K}$ candidate vectors be avoided, but determining the optimal solution reduces to:

1. Sorting the absolute values of the elements of \mathbf{v} and selecting the indices of the K largest elements of the sorted vector as the I_{opt} set (can be done in $\mathcal{O}(N \log N)$).
2. Constructing $\mathbf{x}_{\text{opt}} \in \mathbb{R}^N$ (can be done in $\mathcal{O}(N)$):

$$\forall i \in I_{\text{opt}} : x_{\text{opt}_i} = \frac{v_i}{\|\mathbf{v}_{I_{\text{opt}}}\|} \quad \& \quad \forall i \in \overline{I_{\text{opt}}} : x_{\text{opt}_i} = 0$$

In fact, in step 1, sorting the elements of $|\mathbf{v}|$ is unnecessary, since we are only interested in the set of K largest elements and not their ordering. Towards this end, we can utilize existing algorithms [4] that locate the K -th order element in an unsorted array in $\mathcal{O}(N)$ and then determine the $K - 1$ elements smaller than the K -th order element, again in $\mathcal{O}(N)$. Thus, the total complexity of the rank-1 solution declines even further.

V Maximization of a Rank-deficient Quadratic Form with a Real Vector Argument under a cardinality constraint

In the general case, \mathbf{V} is a $N \times D$ matrix. Since $\mathbf{x}^T \mathbf{V} \mathbf{V}^T \mathbf{x} = (\mathbf{V}^T \mathbf{x})^T \mathbf{V}^T \mathbf{x} = \|\mathbf{V}^T \mathbf{x}\|^2$, optimization problem (7) can be written as³:

$$\mathbf{x}_{\text{opt}} = \arg \max_{\substack{\mathbf{x} \in \mathbb{R}^N \\ \|\mathbf{x}\| = 1 \\ \|\mathbf{x}\|_0 \leq K}} \|\mathbf{V}^T \mathbf{x}\|. \quad (18)$$

W.l.o.g. we assume that each row of \mathbf{V} has at least one nonzero element, i.e. $\mathbf{V}_{n,1:2} \neq \mathbf{0}_{1 \times 2}$. Indeed, if there exists an index $n \in \{1, \dots, N\}$ such that $\mathbf{V}_{n,1:2} = \mathbf{0}_{1 \times 2}$ then independently of the value of the corresponding element x_n of \mathbf{x} , the contribution of this row to the value of $\|\mathbf{V}^T \mathbf{x}\|^2$ will be zero. So there is no point in ‘‘spending’’ in x_n a weight that could be distributed to other elements of \mathbf{x} : we can ignore the n -th row of \mathbf{V} , replace \mathbf{V} by $\mathbf{V}_{\{1, \dots, n-1, n+1, \dots, N\}, :}$ and, hence, reduce the problem size from N to $N - 1$. In the final solution \mathbf{x}_{opt} , x_n will of course be set to zero. In addition, we assume that no two rows of \mathbf{V} have the same absolute value in their first element, i.e. $|V_{i,1}| \neq |V_{j,1}|, \forall i, j \in \{1, \dots, N\}$.

For our subsequent developments, we introduce the spherical coordinates $\phi_1 \in (-\pi, \pi]$, $\phi_2, \dots, \phi_{D-1} \in (-\frac{\pi}{2}, \frac{\pi}{2}]$ and define the spherical coordinate vector

$$\boldsymbol{\phi}_{i:j} \triangleq [\phi_i, \phi_{i+1}, \dots, \phi_j]^T \quad (19)$$

and the hyperpolar vector

³In the formation of (18) we have deliberately ignored the power of $\|\mathbf{V}^T \mathbf{x}\|$ since \mathbf{x}_{opt} defined in (3) maximizes also the square root of the quadratic form.

$$\mathbf{c}(\phi_{1:D-1}) \triangleq \begin{bmatrix} \sin \phi_1 \\ \cos \phi_1 \sin \phi_2 \\ \cos \phi_1 \cos \phi_2 \sin \phi_3 \\ \vdots \\ \cos \phi_1 \cos \phi_2 \dots \sin \phi_{D-1} \\ \cos \phi_1 \cos \phi_2 \dots \cos \phi_{D-1} \end{bmatrix}. \quad (20)$$

From Cauchy–Schwartz inequality we know that for any vector $\mathbf{a} \in \mathbb{R}^D$:

$$\mathbf{a}^T \mathbf{c}(\phi_{1:D-1}) \leq \|\mathbf{a}\| \underbrace{\|\mathbf{c}(\phi_{1:D-1})\|}_{=1}, \quad (21)$$

with equality if and only if $\phi_{1:D-1}$ is the vector of spherical coordinates of \mathbf{a} . Substituting vector \mathbf{a} with vector $(\mathbf{V}^T \mathbf{x}) \in \mathbb{R}^D$ in (21) we get:

$$\begin{aligned} (\mathbf{V}^T \mathbf{x})^T \mathbf{c}(\phi_{1:D-1}) &\leq \|\mathbf{V}^T \mathbf{x}\| \underbrace{\|\mathbf{c}(\phi_{1:D-1})\|}_{=1} \\ \Rightarrow \mathbf{x}^T \mathbf{V} \mathbf{c}(\phi_{1:D-1}) &\leq \|\mathbf{V}^T \mathbf{x}\|. \end{aligned} \quad (22)$$

The quantity to be maximized appears in the right hand of the inequality. In addition, we have already stated that equality in (22) can be achieved, if and only if $\mathbf{V}^T \mathbf{x}$ is parallel to $\mathbf{c}(\phi_{1:D-1})$. Since the choice of $\phi_{1:D-1}$ is arbitrary, for any \mathbf{x} we can assign a value to $\phi_{1:D-1}$ such that $\mathbf{c}(\phi_{1:D-1})$ is co-linear with $\mathbf{V}^T \mathbf{x}$ and we can, thus, achieve equality for any \mathbf{x} . Based on this observation, in order to find \mathbf{x} that maximizes $\|\mathbf{V}^T \mathbf{x}\|$, we can equivalently try to find the (\mathbf{x}, ϕ) pair that maximizes $(\mathbf{V}^T \mathbf{x})^T \mathbf{c}(\phi_{1:D-1})$. The advantage of the transition from the original problem to the equivalent problem of maximizing $(\mathbf{V}^T \mathbf{x})^T \mathbf{c}(\phi_{1:D-1})$ over \mathbf{x} and ϕ will become obvious later.

According to the previous paragraph, we have the following, critical for our subsequent developments, equality:

$$\begin{aligned} \max_{\substack{\mathbf{x} \in \mathbb{R}^N \\ \|\mathbf{x}\| = 1 \\ \|\mathbf{x}\|_0 \leq K}} \|\mathbf{V}^T \mathbf{x}\| &= \max_{\substack{\mathbf{x} \in \mathbb{R}^N \\ \|\mathbf{x}\| = 1 \\ \|\mathbf{x}\|_0 \leq K}} \max_{\phi_{1:D-1} \in (-\pi, \pi] \times (-\frac{\pi}{2}, \frac{\pi}{2}]} \mathbf{x}^T \mathbf{V} \mathbf{c}(\phi_{1:D-1}). \end{aligned} \quad (23)$$

We interchange the maximizations in (23) to obtain the equivalent problem

$$\max_{\substack{\mathbf{x} \in \mathbb{R}^N \\ \|\mathbf{x}\| = 1 \\ \|\mathbf{x}\|_0 \leq K}} \|\mathbf{V}^T \mathbf{x}\| = \max_{\phi_{1:D} \in (-\pi, \pi] \times (-\frac{\pi}{2}, \frac{\pi}{2}]^{D-2}} \left(\max_{\substack{\mathbf{x} \in \mathbb{R}^N \\ \|\mathbf{x}\| = 1 \\ \|\mathbf{x}\|_0 \leq K}} \mathbf{x}^T \underbrace{\mathbf{V} \mathbf{c}(\phi_{1:D-1})}_{\mathbf{v}'(\phi_{1:D-1})} \right). \quad (24)$$

For a given point $\phi_{1:D-1} \in (-\pi, \pi] \times (-\frac{\pi}{2}, \frac{\pi}{2}]^{D-2}$, $\mathbf{V} \mathbf{c}(\phi_{1:D-1})$ is a fixed vector and the internal maximization problem

$$\max_{\substack{\mathbf{x} \in \mathbb{R}^N \\ \|\mathbf{x}\| = 1 \\ \|\mathbf{x}\|_0 \leq K}} \mathbf{x}^T \mathbf{V} \mathbf{c}(\phi_{1:D-1}) = \max_{\substack{\mathbf{x} \in \mathbb{R}^N \\ \|\mathbf{x}\| = 1 \\ \|\mathbf{x}\|_0 \leq K}} \mathbf{x}^T \mathbf{v}'(\phi_{1:D-1}) \quad (25)$$

is a rank-1 case problem! So for any given point $\phi_{1:D-1}$ we can determine the optimal index-set $I(\phi_{1:D-1})$ of the nonzero elements of \mathbf{x} based on the criteria developed at (17): $I(\phi_{1:D-1})$ contains the indices of the K largest elements of vector $|\mathbf{v}'(\phi_{1:D-1})| = |\mathbf{V} \mathbf{c}(\phi_{1:D-1})|$.

To gain some intuition into the purpose of inserting the second variable $\phi_{1:D-1}$, notice that every element of $\pm \mathbf{V} \mathbf{c}(\phi_{1:D-1})$ is actually a continuous function of $\phi_{1:D-1}$, a D -dimensional hypersurface and so are the elements of $|\mathbf{V} \mathbf{c}(\phi_{1:D-1})|$.

$$\pm \mathbf{V} \mathbf{c}(\phi_{1:D-1}) = \begin{pmatrix} \pm V_{1,1} \sin \phi_1 \pm \sum_{d=2}^{D-1} V_{1,d} \prod_{i=1}^{d-1} \cos \phi_i \sin \phi_d \pm V_{1,D} \prod_{i=1}^{D-1} \cos \phi_i \\ \vdots \\ \pm V_{N,1} \sin \phi_1 \pm \sum_{d=2}^{D-1} V_{N,d} \prod_{i=1}^{d-1} \cos \phi_i \sin \phi_d \pm V_{N,D} \prod_{i=1}^{D-1} \cos \phi_i \end{pmatrix}. \quad (26)$$

When we sort the elements of $|\mathbf{V} \mathbf{c}(\phi_{1:D-1})|$ at a given point $\phi_{1:D-1}$, we actually sort the hypersurfaces at point $\phi_{1:D-1}$ according to their magnitude. The key observation in our algorithm, is that due to the continuity of the hypersurfaces in the Φ^{D-1} hypercube, we expect that in an area “around” $\phi_{1:D-1}$ the hypersurfaces will retain their magnitude-sorting. So we expect the formation of cells in the Φ^{D-1} hypercube, within which the magnitude-sorting of the hypersurfaces will remain unaltered, irrespectively of whether the magnitude of each hypersurface changes. Moreover, even if the sorting of the hypersurfaces changes at some point around $\phi_{1:D-1}$ it is possible that the I does not change. So we expect the

formation of regions in the Φ^{D-1} hypercube which expand over more than one cells and within which the I -set remains unaltered, even if the sorting of the hypersurfaces changes. If we can efficiently determine all these cells (or even better regions) and obtain the corresponding I -sets, then the set of all candidate index-sets may be significantly smaller than the set of all $\binom{N}{K}$ possible index-sets. Once all the candidate I -sets have been collected, I_{opt} and \mathbf{x}_{opt} will be determined through exhaustive search among the candidate sets.

■ ————— ■ Illustration Example 1 ■ ————— ■

To illustrate such a partition of the Φ^{D-1} hypercube, we use a simple example from the case $D = \mathcal{R}(V) = 2$ and we set $N = 4$ and $K = 2$. In our case, the auxiliary vector of spherical coordinates consists of only $D - 1 = 1$ element, ϕ_1 and the rows of $|\mathbf{Vc}(\phi_1)|$ correspond to two-dimensional hypersurfaces (curves). In Figure 1 we plot the 4 curves that originate from the 4 rows of $|\mathbf{Vc}(\phi_1)|$. We observe that there are cells (intervals) in the Φ^1 field within which the sorting of the curves does not change. The borders of cells are denoted by vertical dashed lines at points $\hat{\phi}_1$ where any two curves intersect. Highlighted vertical dashed lines denote the end of regions, i.e. of areas in the Φ^1 field, within which the I -set does not change, even if the sorting of the curves does. In addition, note that two independent regions, R_1 and R_3 are associated with the same index-set. Therefore, although the number of cells given N and D is well determined, the regions can (and will actually always) be fewer and the number of independent I -sets can be even lower.

In our simple example where $N = 4$, $D = 2$, $K = 2$, there are in total $\binom{4}{2} = 6$ possible index-sets. The algorithm created 12 cells included in 4 regions which are associated with 3 independent candidate I -sets. Due to the small value of N , the number of cells exceeds the total number of possible index-sets, but this is not true for greater values of N and K . Despite this, our small-scale example allows us to observe the formation of regions and, most importantly, that the size of candidate index-sets can be reduced. Our objective is to efficiently identify these candidate index-sets since the optimal index-set I_{opt} lies among them.

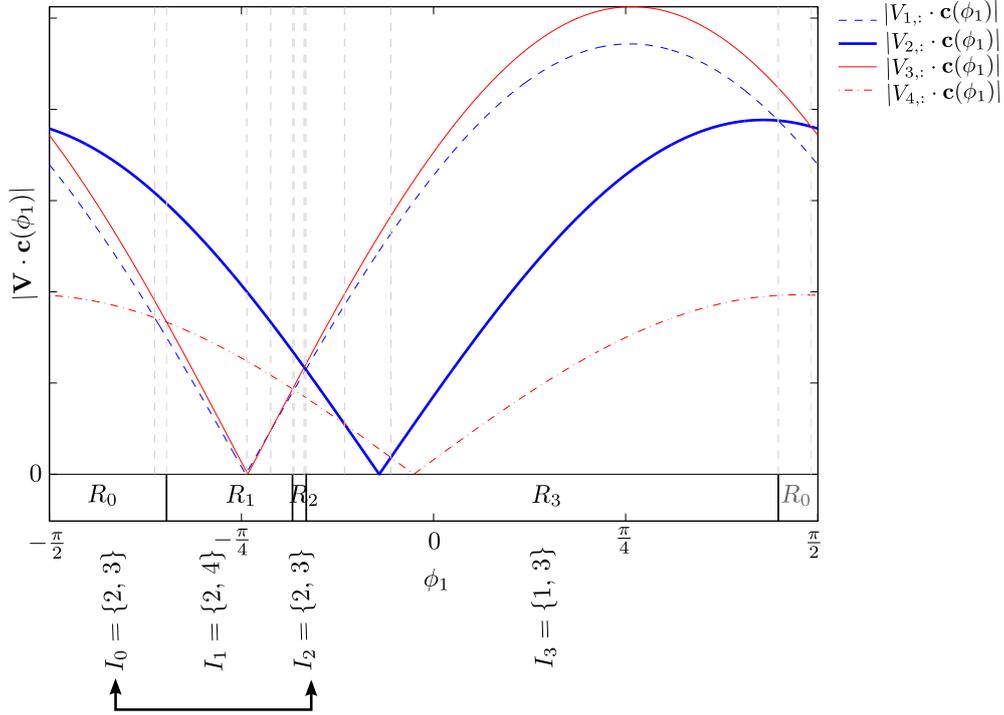


Figure 1: Example: Rank-2, Cells and regions in the Φ^1 field.

In (25) we observed that at a given point $\phi_{1:D-1}$ the maximization problem resembles the rank-1 case and consequently, the I -set at $\phi_{1:D-1}$ consists of the indices of the K largest elements of $|\mathbf{Vc}(\phi_{1:D-1})|$. Motivated by this observation, we define a *labeling function* $I(\cdot)$ that maps a point $\phi_{1:D-1}$ to an index-set I :

$$I(\mathbf{V}_{N \times D}; \phi_{1:D-1}) \triangleq \arg \max_I \sum_{i \in I} |(\mathbf{Vc}(\phi_{1:D-1}))_i|. \quad (27)$$

For simplicity, one can think of $I(\mathbf{V}_{N \times D}; \phi_{1:D-1})$ as a function that sorts the elements of $|\mathbf{Vc}(\phi_{1:D-1})|$ and then returns the set of indices of the K largest elements. In terms of complexity however, we underline that $I(\mathbf{V}_{N \times D}; \phi_{1:D-1})$ does not sort the elements of $|\mathbf{Vc}(\phi_{1:D-1})|$ and operates in $\mathcal{O}(N)$. The index-set can be returned in many equivalent forms, e.g. as set of K elements, or as a binary $N \times 1$ vector in which the n -th place is equal to 1 if index n is included in the I -set and 0 otherwise. Using the labeling function, for the given $N \times D$ matrix \mathbf{V} , each point $\phi_{1:D-1} \in (-\pi, \pi] \times (-\frac{\pi}{2}, \frac{\pi}{2}]^{D-2}$ is mapped to

a candidate index-set and the optimal index-set I_{opt} belongs to

$$\bigcup_{\phi_{1:D-1} \in (-\pi, \pi] \times (-\frac{\pi}{2}, \frac{\pi}{2}]^{D-2}} I(\mathbf{V}_{N \times D}; \phi_{1:D-1}). \quad (28)$$

We note that $\mathbf{Vc}([\phi_1, \phi_{2:D-1}]) = -\mathbf{Vc}([\phi_1 + \pi, \phi_{2:D-1}]) \Rightarrow |\mathbf{Vc}([\phi_1, \phi_{2:D-1}])| = |\mathbf{Vc}([\phi_1 + \pi, \phi_{2:D-1}])|$ for any $\mathbf{V}_{N \times D}$ and $\phi_{1:D-1} \in (-\pi, \pi] \times (-\frac{\pi}{2}, \frac{\pi}{2}]^{D-2}$. So, the sorting of the elements of $|\mathbf{Vc}(\phi_{1:D-1})|$, is the same at points $[\phi_1, \phi_{2:D-1}]$ and $[\phi_1 + \pi, \phi_{2:D-1}]$ and therefore, $I(\mathbf{V}_{N \times D}; [\phi_1, \phi_{2:D-1}]) = I(\mathbf{V}_{N \times D}; [\phi_1 + \pi, \phi_{2:D-1}])$. Since ϕ_1 and $\phi_1 + \pi$ result in the same candidate index-set, we can ignore the values of $\phi_1 \in (-\pi, -\frac{\pi}{2}] \cup (\frac{\pi}{2}, \pi]$ and rewrite the optimization problem as:

$$\max_{\phi_{1:D-1} \in \Phi^{D-1}} \max_I \sum_{i \in I} |(\mathbf{Vc}(\phi_{1:D-1}))_i|, \quad \Phi \triangleq (-\frac{\pi}{2}, \frac{\pi}{2}]. \quad (29)$$

Finally, we collect all candidate index-sets into set

$$\begin{aligned} \mathcal{I}(\mathbf{V}_{N \times D}) &\triangleq \bigcup_{\phi_{1:D-1} \in \Phi^{D-1}} \{I(\mathbf{V}_{N \times D}; \phi_{1:D-1})\} \\ &= \left\{ \hat{I} \subseteq \{1, \dots, N\} : |\hat{I}| = K \text{ and } \exists \phi_{1:D-1} \in \Phi^{D-1} \text{ such} \right. \\ &\quad \left. \text{that } \hat{I} = I(\mathbf{V}_{N \times D}; \phi_{1:D-1}) \right\} \\ &\subset \left\{ \hat{I} \subseteq \{1, \dots, N\} : |\hat{I}| = K \right\} \end{aligned} \quad (30)$$

and observe that the optimal index-set I_{opt} belongs to $\mathcal{I}(\mathbf{V}_{N \times D})$. In the following we:

- (i) show that $|\mathcal{I}(\mathbf{V}_{N \times D})| \leq \sum_{d=0}^{D-1} \binom{N^2 - N - 1}{d}$ and
- (ii) develop an algorithm for the construction of $\mathcal{I}(\mathbf{V}_{N \times D})$.

We begin by observing that the labeling function which determines the I -set at a point $\phi_{1:D-1}$, is based on pair-wise comparisons of the elements of $\mathbf{Vc}(\phi_{1:D-1})$. Besides, even sorting can be considered as a procedure associated with pairwise comparisons. Now, recall that each element of $|\mathbf{V}_{N \times D} \mathbf{c}(\phi_{1:D-1})|$ is a continuous function of $\phi_{1:D-1}$, a D -dimensional hypersurface, and any point $\phi_{1:D-1}$ is mapped to an index-set I which is determined by comparing the magnitudes of these hypersurfaces at $\phi_{1:D-1}$. Due to the continuity of hypersurfaces, the index-set I does not change in the ‘‘neighborhood’’ of

$\phi_{1:D-1}$. A necessary condition for the I set to change is two of the hypersurfaces to change their magnitude ordering. The switching occurs over the intersection of two hypersurfaces. Note, that this is, of course, not a sufficient condition for the I -set to change, since the two hypersurfaces that intersect can correspond to indices that are both outside or inside the I -set. In such a case, although the ordering of the hypersurfaces changes, the index-set I does not. At the intersection of two hypersurfaces, say those that originate from rows i and j of \mathbf{V} , we have:

$$\left| \left(\mathbf{Vc}(\phi_{1:D-1}) \right)_i \right| = \left| \left(\mathbf{Vc}(\phi_{1:D-1}) \right)_j \right|, \quad i, j \in \{1, \dots, N\}. \quad (31)$$

Equality (31) yields the following two equations:

$$\begin{aligned} & \left(\mathbf{Vc}(\phi_{1:D-1}) \right)_i = + \left(\mathbf{Vc}(\phi_{1:D-1}) \right)_j \\ \Rightarrow & V_{i,:} \mathbf{c}(\phi_{1:D-1}) = + V_{j,:} \mathbf{c}(\phi_{1:D-1}) \\ \Rightarrow & (V_{i,:} - V_{j,:}) \mathbf{c}(\phi_{1:D-1}) = 0 \end{aligned} \quad (32)$$

$$\Rightarrow \phi_1 = \tan^{-1} \left(- \frac{(V_{i,2:D} - V_{j,2:D})^T \mathbf{c}(\phi_{2:D-1})}{(V_{i,1} - V_{j,1})} \right) \quad (33)$$

and

$$\begin{aligned} & \left(\mathbf{Vc}(\phi_{1:D-1}) \right)_i = - \left(\mathbf{Vc}(\phi_{1:D-1}) \right)_j \\ \Rightarrow & V_{i,:} \mathbf{c}(\phi_{1:D-1}) = - V_{j,:} \mathbf{c}(\phi_{1:D-1}) \\ \Rightarrow & (V_{i,:} + V_{j,:}) \mathbf{c}(\phi_{1:D-1}) = 0 \end{aligned} \quad (34)$$

$$\Rightarrow \phi_1 = \tan^{-1} \left(- \frac{(V_{i,2:D} + V_{j,2:D})^T \mathbf{c}(\phi_{2:D-1})}{(V_{i,1} + V_{j,1})} \right). \quad (35)$$

Each one of equations (33) and (35) determines a $(D - 1)$ -dimensional hypersurface that partitions the $(D - 1)$ -dimensional hypercube Φ^{D-1} into two regions. More specifically:

1. Function $\phi_1 = \tan^{-1} \left(- \frac{(V_{i,2:D} - V_{j,2:D})^T \mathbf{c}(\phi_{2:D-1})}{(V_{i,1} - V_{j,1})} \right)$ determines a hypersurface $S(V_{i,:}; V_{j,:})^4$ which partitions Φ^{D-1} into two regions: one where $V_{i,:} \mathbf{c}(\phi_{1:D-1}) > V_{j,:} \mathbf{c}(\phi_{1:D-1})$ and one where $V_{i,:} \mathbf{c}(\phi_{1:D-1}) < V_{j,:} \mathbf{c}(\phi_{1:D-1})$

⁴From equation (35), it is obvious that hypersurface $S(-V_{i,:}; V_{j,:})$ coincides with $S(V_{i,:}; -V_{j,:})$ and $S(-V_{i,:}; -V_{j,:})$ coincides with $S(V_{i,:}; V_{j,:})$. Therefore, hypersurfaces $S(-V_{i,:}; V_{j,:})$ and $S(-V_{i,:}; -V_{j,:})$ do not have to be considered separately.

2. Function $\phi_1 = \tan^{-1} \left(-\frac{(V_{i,2:D}+V_{j,2:D})^T \mathbf{c}(\phi_{2:D-1})}{(V_{i,1}+V_{j,1})} \right)$ determines a hypersurface $S(V_{i,:}; -V_{j,:})^4$ which partitions Φ^{D-1} into two regions: one where $V_{i,:} \mathbf{c}(\phi_{1:D-1}) > -V_{j,:} \mathbf{c}(\phi_{1:D-1})$ and one where $V_{i,:} \mathbf{c}(\phi_{1:D-1}) < -V_{j,:} \mathbf{c}(\phi_{1:D-1})$

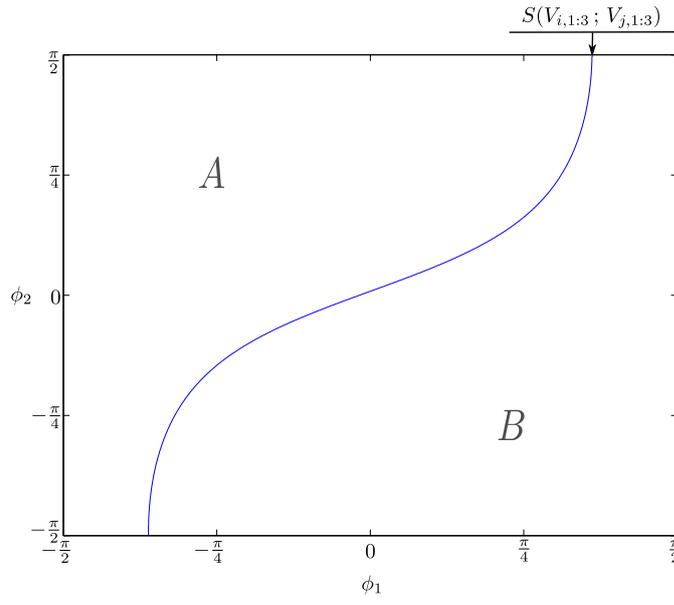
Apparently, both hypersurfaces $S(V_{i,:}; V_{j,:})$ and $S(V_{i,:}; -V_{j,:})$ together, partition Φ^{D-1} into four regions, in two of which $|V_{i,:} \mathbf{c}(\phi_{1:D-1})| > |V_{j,:} \mathbf{c}(\phi_{1:D-1})|$ while in the other two $|V_{i,:} \mathbf{c}(\phi_{1:D-1})| < |V_{j,:} \mathbf{c}(\phi_{1:D-1})|$

The two hypersurfaces, $S(V_{i,:}; V_{j,:})$ and $S(V_{i,:}; -V_{j,:})$ originate from a pair of matrix \mathbf{V} rows. Since the N rows of the $N \times D$ matrix $\mathbf{V}_{N \times D}$ can be combined in $\binom{N}{2} = \frac{N^2-N}{2}$ pairs, and each pair yields two hypersurfaces, matrix \mathbf{V} is associated with a total of $2 \cdot \binom{N}{2} = N^2 - N$ hypersurfaces, which partition the hypercube Φ^{D-1} into L cells C_1, C_2, \dots, C_L such that $\bigcup_{l=1}^L C_l = \Phi^{D-1}$, $C_l \cap C_m = \emptyset$ if $l \neq m$ and each cell C_l corresponds to an index-set I . Of course more than one cells may correspond to the same index-set. Our objective is to efficiently identify these candidate index-sets, since one of them is the optimal index-set.

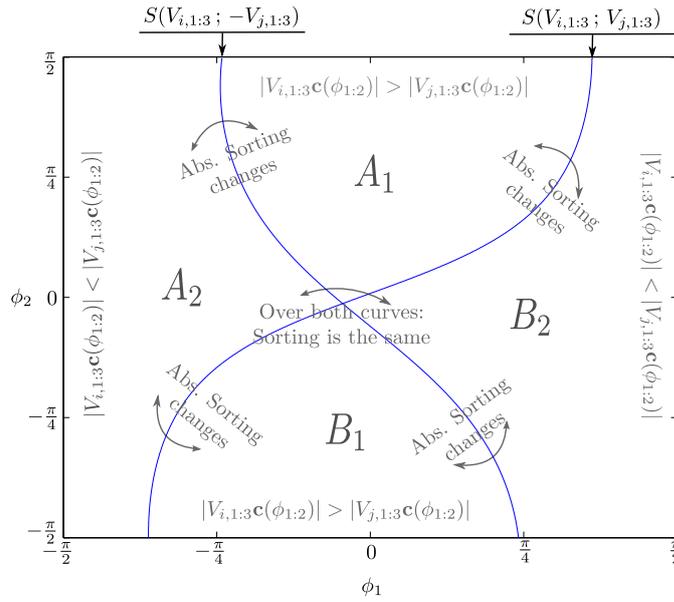
■ ————— ■ Illustration Example 2 ————— ■

To illustrate such a partition, we will refer to a rank-3 example, where each row of matrix \mathbf{V} originates a three-dimensional surface. Hypersurfaces $S(V_{i,:}; V_{j,:})$ and $S(V_{i,:}; -V_{j,:})$, which are based on the intersection of two three-dimensional surfaces, are two-dimensional curves in the Φ^2 plane. In Figure 2a, we plot the curve $S(V_{i,:}; -V_{j,:}) = \{(\phi_1, \phi_2) : \phi_1 = \tan^{-1} \left(-\frac{(V_{i,2:3}-V_{j,2:3})^T \mathbf{c}(\phi_2)}{(V_{i,1}-V_{j,1})} \right), \phi_1, \phi_2 \in (-\frac{\pi}{2}, \frac{\pi}{2}]\}$ and see that the curve partitions the two dimensional plane Φ^2 into two regions A and B . If $V_{i,1:3} \mathbf{c}(\phi_{1:2}) > V_{j,1:3} \mathbf{c}(\phi_{1:2})$ inside A , then $V_{i,1:3} \mathbf{c}(\phi_{1:2}) < V_{j,1:3} \mathbf{c}(\phi_{1:2})$ inside B and vice-versa. Note, however, that the sorting of $|V_{i,1:3} \mathbf{c}(\phi_{1:2})|$ and $|V_{j,1:3} \mathbf{c}(\phi_{1:2})|$ is not fixed in the interior of regions A and B .

In Figure 2b, we plot both curves $S(V_{i,:}; -V_{j,:})$ and $S(V_{i,:}; V_{j,:})$. Each partitions the two dimensional plane Φ^2 into two regions, for a total of four regions, A_1, A_2, B_1 and B_2 . In the interior of each one of the four areas, the sorting of $|V_{i,:} \mathbf{c}(\phi_{1:2})|$ and $|V_{j,:} \mathbf{c}(\phi_{1:2})|$ is stable. The significant part is that when we move over $S(V_{i,:}; -V_{j,:})$ or $S(V_{i,:}; V_{j,:})$ the sorting of $|V_{i,:} \mathbf{c}(\phi_{1:2})|$ and $|V_{j,:} \mathbf{c}(\phi_{1:2})|$ changes, while when moving over both curves, the initial sorting is restored.



(a) $S(V_{i,1:3}; V_{j,1:3})$



(b) $S(V_{i,1:3}; V_{j,1:3})$ and $S(V_{i,1:3}; -V_{j,1:3})$

Figure 2: Example: Rank-3, $S(\cdot; \cdot)$ hypersurfaces.

Several properties of the resulting partition that are very important for our subsequent

developments are presented in the following proposition. The proof is provided in the Appendix.

Proposition 1 *Let $\mathbf{V} \in \mathbb{R}^{N \times D}$ be a rank- D matrix and $|V_{n,1}| \neq |V_{m,1}|$, $n, m = 1, 2, \dots, N$, $n \neq m$. The following hold true:*

Part a *Each set of $D - 1$ hypersurfaces of the form $S(V_{i,:}; \pm V_{j,:})$ has either a single intersection or uncountably many intersections in Φ_{D-1} .*

Part b *For any $\phi_1, \phi_2, \dots, \phi_{D-1} \in \Phi$,*

$$(i) \quad |\mathbf{V}_{N \times D} \mathbf{c}([\phi_{1:D-2}, \frac{\pi}{2}])| = |\mathbf{V}_{N \times (D-1)} \mathbf{c}(\phi_{1:D-2})|,$$

$$(ii) \quad |\mathbf{V}_{N \times D} \mathbf{c}([\phi_{1:D-2}, -\frac{\pi}{2}])| = |\mathbf{V}_{N \times D} \mathbf{c}([-\phi_{1:D-2}, \frac{\pi}{2}])|,$$

$$(iii) \quad |\mathbf{V}_{N \times D} \mathbf{c}([\phi_{1:D-3}, \frac{\pi}{2}, \phi_{D-1}])| = |\mathbf{V}_{N \times (D-2)} \mathbf{c}(\phi_{1:D-3})|,$$

$$(iv) \quad |\mathbf{V}_{N \times D} \mathbf{c}([\phi_{1:D-3}, -\frac{\pi}{2}, \phi_{D-1}])| = |\mathbf{V}_{N \times D} \mathbf{c}([-\phi_{1:D-3}, \frac{\pi}{2}, \phi'_{D-1}])|, \forall \phi'_{D-1} \in \Phi,$$

$$(v) \quad |\mathbf{V}_{N \times D} \mathbf{c}([\phi_{1:D-3}, \pm \frac{\pi}{2}, \phi_{D-1}])| = |\mathbf{V}_{N \times D} \mathbf{c}([\phi_{1:D-3}, \pm \frac{\pi}{2}, \phi'_{D-1}])|, \forall \phi'_{D-1} \in \Phi.$$

(For Proof, see Appendix.) □

For convenience, in the following we use a pair $\{i, j\}$ to denote the rows of matrix $\mathbf{V}_{N \times D}$, that originate hypersurface $S(\frac{i}{|i|} V_{|i|,:}; \frac{j}{|j|} V_{|j|,:})$. Moreover, we allow i and j to be negative in order to encapsulate the information about the sign with which each row participates in the generation of hypersurface S , i.e.:

$$\{i, j\} \mapsto S(\frac{i}{|i|} V_{|i|,:}; \frac{j}{|j|} V_{|j|,:}), \quad i, j \in \{-N, \dots, -1, 1, \dots, N\}, \quad |i| \neq |j|. \quad (36)$$

Let $P_{D-1} \triangleq \{\{p_{1,1}, p_{1,2}\}, \{p_{2,1}, p_{2,2}\}, \dots, \{p_{D-1,1}, p_{D-1,2}\}\}$ denote a set of $D - 1$ such pairs where $p_{*,1}, p_{*,2} \in -N, \dots, -1, 1, \dots, N$ and $|p_{*,1}| \neq |p_{*,2}|$. Each pair $\{p_{*,1}, p_{*,2}\}$ corresponds to a hypersurface $S(\frac{p_{*,1}}{|p_{*,1}|} V_{|p_{*,1}|,:}; \frac{p_{*,2}}{|p_{*,2}|} V_{|p_{*,2}|,:})$ and $\phi(\mathbf{V}_{N \times D}; P_{D-1}) \in \Phi^{D-1}$ is the vector of spherical coordinates of the intersection of the $D - 1$ hypersurfaces.

If $\phi(\mathbf{V}_{N \times D}; P_{D-1})$ is uniquely determined according to Proposition 1 Part a, then it “leads” a cell, say $C(\mathbf{V}_{N \times D}; P_{D-1})$, associated with an index-set $I(\mathbf{V}_{N \times D}; P_{D-1})$ in the sense that $I(\mathbf{V}_{N \times D}; P_{D-1}) = I(\mathbf{V}_{N \times D}; \phi_{1:D-1})$ for all $\phi_{1:D-1} \in C(\mathbf{V}_{N \times D}; P_{D-1})$ and $\phi(\mathbf{V}_{N \times D}; P_{D-1})$ is the single point of $C(\mathbf{V}_{N \times D}; P_{D-1})$ for which ϕ_{D-1} is minimized. In other words, the I -set associated with all points $\phi_{1:D-1}$ in the interior of the cell is the same as the I -set at the leading vertex. In fact, the actual sorting of $|\mathbf{Vc}(\phi_{1:D-1})|$ for all points in the interior of the cell is the same as the sorting at the leading vertex, and the I -set may characterize a greater area that includes many cells. For the moment, let's just say that each cell is associated with an I -set. We collect all index-sets into

$$\mathcal{I}(\mathbf{V}_{N \times D}) \triangleq \bigcup_{P_{D-1}} I(\mathbf{V}_{N \times D}; P_{D-1}) \quad (37)$$

and observe that $\mathcal{I}(\mathbf{V}_{N \times D})$ can only be a subset of the set of all possible $\binom{N}{K}$ index-sets. In addition, since the cells are defined by a leading vertex, and a vertex is determined by a P_{D-1} -set (the intersection of $D - 1$ hypersurfaces) we conclude that there are at most $\binom{N^2 - N}{D-1}$ cells⁵. However, we note that $\binom{N^2 - N}{D-1}$ is only a rough upper bound on the number of cells, which would equal the number of intersecting points if every set of $D - 1$ hypersurfaces had a distinct intersection point. For $D \geq 3$, although an intersection point is determined by $D - 1$ hypersurfaces and all possible combinations of $D - 1$ hypersurfaces are considered, there are many cases in which more than $D - 1$ hypersurfaces intersect at the same point. This leads to examining the same intersection point multiple times. To see this, recall that each hypersurface originates from two rows of matrix \mathbf{V} . Now let's consider any two of the $D - 1$ intersecting hypersurfaces: $S(\binom{i}{|i|} V_{|i|,:}; \binom{j}{|j|} V_{|j|,:})$ and $S(\binom{k}{|k|} V_{|k|,:}; \binom{l}{|l|} V_{|l|,:})$ where $|i| \neq |j|$ and $|k| \neq |l|$. By the definition of each one of these two hypersurfaces, at $\phi_{1:D-1}$ we have $|V_{|i|,:} \mathbf{c}(\phi_{1:D-1})| = |V_{|j|,:} \mathbf{c}(\phi_{1:D-1})|$ and $|V_{|k|,:} \mathbf{c}(\phi_{1:D-1})| = |V_{|l|,:} \mathbf{c}(\phi_{1:D-1})|$.

- If $|i| \notin \{|k|, |l|\}$ and $|j| \notin \{|k|, |l|\}$, no further equality is implied. Of course, there is a possibility all 4 D -dimensional hypersurfaces $\binom{i}{|i|} V_{|i|,:} \mathbf{c}(\phi)$, $\binom{j}{|j|} V_{|j|,:} \mathbf{c}(\phi)$, $\binom{k}{|k|} V_{|k|,:} \mathbf{c}(\phi)$ and $\binom{l}{|l|} V_{|l|,:} \mathbf{c}(\phi)$ intersect at $\phi_{1:D-1}$ but the probability of such a case is practically zero and it is, by no means, implied by the fact that $S(\binom{i}{|i|} V_{|i|,:}; \binom{j}{|j|} V_{|j|,:})$

⁵We have already shown that matrix $\mathbf{V}_{N \times D}$ is associated with $N^2 - N$ hypersurfaces. Of course, when combining these hypersurfaces in sets of $D - 1$ hypersurfaces, there are $\binom{N^2 - N}{D-1}$ possible combinations.

and $S(\frac{k}{|k|}V_{|k|,:}; \frac{l}{|l|}V_{|l|,:})$ intersect at $\phi_{1:D-1}$. Therefore, for simplicity, we ignore this case.

- However, if $|i| \in \{|k|, |l|\}$ or $|j| \in \{|k|, |l|\}$, then we know that one more hypersurface also goes through the intersection point: Assume, for example, that $|i| = |k|$. Then, at point $\phi_{1:D-1}$ we have: $|V_{|i|,:}\mathbf{c}(\phi_{1:D-1})| = |V_{|j|,:}\mathbf{c}(\phi_{1:D-1})|$, $|V_{|i|,:}\mathbf{c}(\phi_{1:D-1})| = |V_{|l|,:}\mathbf{c}(\phi_{1:D-1})|$ and, therefore, a third equality, $|V_{|j|,:}\mathbf{c}(\phi_{1:D-1})| = |V_{|l|,:}\mathbf{c}(\phi_{1:D-1})|$, is deduced. The last equality, implies that hypersurface $S(\frac{j}{|j|}V_{|j|,:}; \frac{l}{|l|}V_{|l|,:})$, where $*$ = $\frac{k}{i}$ denotes only the proper sign, also goes through the intersection point. The fact that more than $D - 1$, say M , hypersurfaces intersect at the same point, along with the fact that we blindly examine all possible combinations of $D - 1$ hypersurfaces, will force us to examine the same intersection point $\binom{M}{D-1}$ times. However, there are at most $\binom{M}{D-1}$ cells originating at that point. Illustrating Example 3 shows, for instance, that in the case of rank-3 cells are less than the number of considered intersections.

Up to now, we have seen that the number of cells is only upper bounded by the number of examined intersection points. We also have to note, though, that many cells are associated with the same index-set. So, finally, we expect that the candidate index-sets contained in $\mathcal{I}(\mathbf{V}_{N \times D})$ will be by far less than $\binom{N^2-N}{D-1}$. However, the problem's complexity is also associated with the number of examined intersection points. Interestingly, the case of rank-3, which has been fully solved, has allowed us to see that not all intersection points have to be examined and has led to significant reduction in the number of examined intersection points, which seems promising for the cases of even higher ranks. More details are given in Section VI.3, where we provide the algorithmic developments for the rank-3 case.

We finally have to note that there exist cells that are not associated with an intersection-vertex:

1. Such cells are those that contain uncountably many points of the form $\phi_{1:D-1} = [\phi_{1:D-2}, -\frac{\pi}{2}]$. In Figure 3 which presents a simple rank-3 example, such cells lie at the bottom of Φ^2 plane, like $C_{(ii)}$. However, according to Proposition 1 Part (b.ii), every such cell can be ignored since there exists another cell that contains points of the form $\phi'_{1:D-1} = [-\phi_{1:D-2}, \frac{\pi}{2}]$, is associated with the same index-set, and is "led"

by an intersection-vertex, unless the initial cell contains a point with $\phi_{D-2} = \pm\frac{\pi}{2}$ as Proposition 1 Part(b.v) states.

2. If $\phi_{D-2} = \pm\frac{\pi}{2}$ for a particular cell, then this cell “exists” for any $\phi_{D-1} \in (-\frac{\pi}{2}, \frac{\pi}{2}]$, implying that we can ignore ϕ_{D-1} or, say, set it to an arbitrary value ϕ'_{D-1} , set ϕ_{D-2} to $\pm\frac{\pi}{2}$ and consider cells defined on $\Phi^{D-3} \times \{\pm\frac{\pi}{2}\} \times \{\phi'_{D-1}\}$. In Figure 3 cell $C_{(v)}$ is such a cell.
3. Finally, due to Proposition 1 Part (b.iv), the cells that are defined when $\phi_{D-2} = -\frac{\pi}{2}$ are associated with the same index-sets as the cells defined when $\phi_{D-2} = \frac{\pi}{2}$. Therefore we can ignore the case $\phi_{D-2} = -\frac{\pi}{2}$, set ϕ_{D-2} to $\frac{\pi}{2}$, ignore ϕ_{D-1} and, according to Proposition 1 Part (b.iii), identify the cells that are determined by the reduced-size matrix $\mathbf{V}_{N \times D-2}$ over the hypercube Φ^{D-3} .

Hence, $\mathcal{I}_{\text{tot}}(\mathbf{V}_{N \times D}) = \mathcal{I}(\mathbf{V}_{N \times D}) \cup \mathcal{I}_{\text{tot}}(\mathbf{V}_{N \times D-2})$ and, by induction,

$$\mathcal{I}_{\text{tot}}(\mathbf{V}_{N \times d}) = \mathcal{I}(\mathbf{V}_{N \times d}) \cup \mathcal{I}_{\text{tot}}(\mathbf{V}_{N \times d-2}), \quad d = 3, 4, \dots, D,$$

which implies that

$$\begin{aligned} \mathcal{I}_{\text{tot}}(\mathbf{V}_{N \times D}) &= \mathcal{I}(\mathbf{V}_{N \times D}) \cup \mathcal{I}(\mathbf{V}_{N \times D-2}) \cup \dots \cup \mathcal{I}(\mathbf{V}_{N \times (D-2 \lfloor \frac{D-1}{2} \rfloor)}) \\ &= \bigcup_{d=0}^{\lfloor \frac{D-1}{2} \rfloor} \mathcal{I}(\mathbf{V}_{N \times (D-2d)}). \end{aligned} \quad (38)$$

As a result, the cardinality of $\mathcal{I}_{\text{tot}}(\mathbf{V}_{N \times D})$ is

$$\begin{aligned} |\mathcal{I}_{\text{tot}}(\mathbf{V}_{N \times D})| &\leq |\mathcal{I}(\mathbf{V}_{N \times D})| + |\mathcal{I}(\mathbf{V}_{N \times D-2})| + \dots + |\mathcal{I}(\mathbf{V}_{N \times (D-2 \lfloor \frac{D-1}{2} \rfloor)})| \\ &\leq \binom{N^2 - N}{D-1} + \binom{N^2 - N}{D-3} + \dots + \binom{N^2 - N}{D-1-2 \lfloor \frac{D-1}{2} \rfloor} \\ &= \sum_{d=0}^{\lfloor \frac{D-1}{2} \rfloor} \binom{N^2 - N}{D-1-2d} = \sum_{d=0}^{D-1} \binom{N^2 - N - 1}{d}. \end{aligned} \quad (39)$$

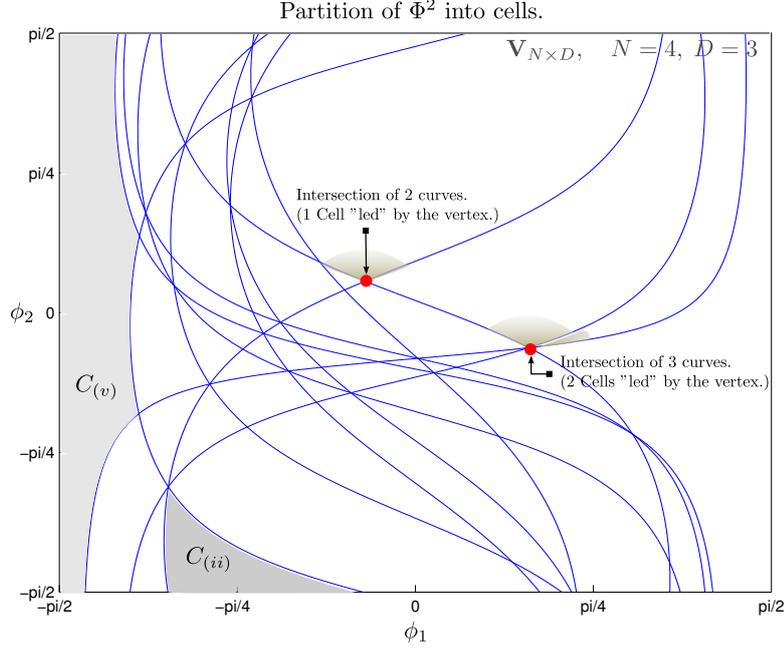


Figure 3: Example: Rank-3, Partition of the Φ^2 plane.

■ ————— ■ Illustration Example 3 ————— ■

In Figure 3 we demonstrate the partition of the Φ^2 plane, based on a random matrix $\mathbf{V}_{N \times D}$ with $N = 4$ and $D = 3$. Observe that each hypersurface (curve) intersects with all other hypersurfaces (curves) exactly once. Each intersection point can be determined as the intersection of $D - 1 = 2$ curves, irrespectively of how many curves go through it. In the case where only two curves intersect, these are of the form $S(\frac{i}{|i|}V_{|i|,:}; \frac{j}{|j|}V_{|j|,:})$ and $S(\frac{k}{|k|}V_{|k|,:}; \frac{l}{|l|}V_{|l|,:})$ with $|i| \neq |j| \neq |k| \neq |l|$. The intersection point is considered only $\binom{2}{2} = 1$ time and it “leads” exactly one cell. In the case where three curves intersect, these are of the form $S(\frac{i}{|i|}V_{|i|,:}; \frac{j}{|j|}V_{|j|,:})$, $S(\frac{i}{|i|}V_{|i|,:}; \frac{k}{|k|}V_{|k|,:})$ and $S(\frac{j}{|j|}V_{|j|,:}; \frac{k}{|k|}V_{|k|,:})$. Such intersection points are naively considered $\binom{3}{2} = 3$ times, while only two cells are “led” by them. Things get worse as D increases.

In Figure 3 we can also observe the cells at the bottom of the Φ^2 plane, at $\phi_2 = -\frac{\pi}{2}$ which are not associated with any node, such as cell $C_{(ii)}$. These type of cells can be ignored. In addition, there are two cells that are attached to the $\phi_1 = \pm\frac{\pi}{2}$ walls, like $C_{(v)}$, that “exist” for any $\phi_{D-1} = \phi_2 \in (-\frac{\pi}{2}, \frac{\pi}{2}]$. From these two cells, one can be ignored, and the other is

examined by taking the reduced-size matrix $\mathbf{V}_{N \times D-2} = \mathbf{V}_{N \times 1}$, i.e. the first column of \mathbf{V} .

To summarize the developments in this section, we have utilized $D - 1$ auxiliary spherical coordinates, partitioned the hypercube Φ^{D-1} into cells that are associated with index-sets and proved that the optimal index-set, I_{opt} , belongs to \mathcal{I}_{tot} . Therefore, the initial problem has been converted into numerical maximization of $\|\mathbf{V}^T \mathbf{x}\|$ among all index-sets $I \in \mathcal{I}_{\text{tot}}(\mathbf{V}_{N \times D})$. The cost of such an optimization is upper bounded by $\sum_{d=0}^{D-1} \binom{N^2 - N - 1}{d} \subseteq \mathcal{O}(N^{2(D-1)})$, but this is only a loose upper bound. In the next section we present a fully developed algorithm for the construction of $\mathcal{I}(\mathbf{V}_{N \times D})$ in the rank-2 and rank-3 cases.

VI Algorithmic Developments

An algorithm for the construction of $\mathcal{I}(\mathbf{V}_{N \times D})$ has been fully implemented for the cases of $D = 2$ and $D = 3$. Although the theoretic developments apply to higher rank cases as well, there are still issues to be confronted before a full algorithm is designed. Before we proceed with the description of the two low-rank cases, we can describe one last detail: how the vector of spherical coordinates $\phi(\mathbf{V}_{N \times D}; P_{d-1})$ is computed efficiently. Recall that $\phi(\mathbf{V}_{N \times D}; P_{d-1})$ represents the intersection of $S(\frac{p_{1,1}}{|p_{1,1}|} V_{|p_{1,1}|, :}; \frac{p_{1,2}}{|p_{1,2}|} V_{|p_{1,2}|, :})$, $S(\frac{p_{2,1}}{|p_{2,1}|} V_{|p_{2,1}|, :}; \frac{p_{2,2}}{|p_{2,2}|} V_{|p_{2,2}|, :})$, \dots , $S(\frac{p_{d-1,1}}{|p_{d-1,1}|} V_{|p_{d-1,1}|, :}; \frac{p_{d-1,2}}{|p_{d-1,2}|} V_{|p_{d-1,2}|, :})$, i.e. the solution of

$$\underbrace{\begin{pmatrix} \mathbf{V}_{p_{1,1}, 1:d} & -\mathbf{V}_{p_{1,2}, 1:d} \\ \mathbf{V}_{p_{2,1}, 1:d} & -\mathbf{V}_{p_{2,2}, 1:d} \\ \vdots & \vdots \\ \mathbf{V}_{p_{d-1,1}, 1:d} & -\mathbf{V}_{p_{d-1,2}, 1:d} \end{pmatrix}}_{\mathbf{B}} \mathbf{c}(\phi_{1:d-1}) = \mathbf{0}_{(d-1) \times 1}. \quad (40)$$

According to the proof of Proposition 1, Part (a), for a full-rank $(d - 1) \times d$ real matrix eq. (40) has a unique solution $\phi(\mathbf{V}_{N \times D}; P_{d-1}) \in \Phi^{d-1}$ which consist of the spherical coordinates of the zero right singular vector of \mathbf{B} . Therefore, to obtain $\phi(\mathbf{V}_{N \times D}; P_{d-1})$

we just need to compute the zero right singular vector of \mathbf{B} and calculate its spherical coordinates.

VI.1 Rank-1 Case

The construction of $\mathcal{I}(\mathbf{V}_{N \times D})$ for $D = 1$ is described in the last paragraph of Section IV.

VI.2 Rank-2 Case

In the case $\mathcal{R}(\mathbf{A}) = 2$, (or, equivalently, $D = \mathcal{R}(\mathbf{V}) = 2$), the spherical coordinate vector is reduced to a scalar, ϕ_1 and the auxiliary hyperpolar vector $\mathbf{c}(\phi_1)$ is

$$\mathbf{c}(\phi_1) \triangleq \begin{bmatrix} \sin \phi_1 \\ \cos \phi_1 \end{bmatrix}. \quad (41)$$

Every element of vector $|\mathbf{V}\mathbf{c}(\phi_1)|$ corresponds to a continuous function of ϕ_1 which describes a two-dimensional hypersurface:

$$|V_{i,1:2}\mathbf{c}(\phi_1)| = |V_{i,1} \sin \phi_1 + V_{i,2} \cos \phi_1|, \quad \forall i \in \{1, \dots, N\}. \quad (42)$$

Of course,

$$\pm V_{i,1:2}\mathbf{c}(\phi), \quad \forall i \in \{1, \dots, N\}, \quad (43)$$

are also two-dimensional hypersurfaces. The intersection of any two hypersurfaces described by (43), $S(\frac{i}{|i|}V_{|i|,:}; \frac{j}{|j|}V_{|j|,:})$ with $|i| \neq |j|$, $|i|, |j| \in \{1, \dots, N\}$ is a point which as expected, partitions the Φ^1 field into two regions. According to the theoretic developments, we would have to take a combination of $D - 1$ hypersurfaces $S(\cdot; \cdot)$ to obtain a point in Φ^{D-1} . However, now $D - 1 = 2 - 1 = 1$ and thus each hypersurface $S(\cdot; \cdot)$ is already a point.

Our goal, is the calculation of $\mathcal{I}(\mathbf{V}_{N \times 2})$. We recall, from (37), that the construction of $\mathcal{I}(\mathbf{V}_{N \times d})$ can also be fully parallelized, since the candidate index-set, $I(\mathbf{V}_{N \times d}; P_{d-1})$, can be computed *independently* for each set P_{d-1} which corresponds to a point in Φ^{d-1} . As a result, we need to present a method for the computation of $I(\mathbf{V}_{N \times 2}; P_1)$. (We

note that, according to the above, in the case of rank-2 each P_1 set consist of only one pair $\{p_1, p_2\}$, $|p_1| \neq |p_2|$, $|p_1|, |p_2| \in \{1, \dots, N\}$ which describes only one hypersurface $S(\frac{p_1}{|p_1|}V_{|p_1|,:}; \frac{p_2}{|p_2|}V_{|p_2|,:})$, since this hypersurface already is a single point).

In the following we present a method for the computation of $I(\mathbf{V}_{N \times 2}; \hat{\phi}_1)$ at each intersection point $\hat{\phi}_1$. This method, allows the parallel construction of $\mathcal{I}(\mathbf{V}_{N \times 2})$ and is, hence, referred to as “parallel” implementation. Then we will present an alternative method for the construction of $\mathcal{I}(\mathbf{V}_{N \times 2})$ that sacrifices parallelization in order to achieve a lower complexity. The second method relies on serial scanning of the Φ^1 field and is, hence, referred to as “serial” implementation.

VI.2.A Parallel Implementation

At each intersection point $\hat{\phi}_1$, we try to obtain the I -set associated by the cell “led” by $\hat{\phi}_1$ by calling the labeling function $I(\mathbf{V}_{N \times 2}; \hat{\phi}_1)$ which marks the K elements of the I -set, i.e. the indices of the K largest elements of $|\mathbf{Vc}(\hat{\phi}_1)|$. Point $\hat{\phi}_1$ is defined as the intersection $S(\frac{i}{|i|}V_{|i|,:}; \frac{j}{|j|}V_{|j|,:})$ of a pair of two-dimensional hypersurfaces. So at $\hat{\phi}_1$, we have $|V_{|i|,:}\mathbf{c}(\hat{\phi}_1)| = |V_{|j|,:}\mathbf{c}(\hat{\phi}_1)|$. If indices $|i|$ and $|j|$ are both included in the I -set, then since internal ordering of the K first elements of $|\mathbf{Vc}(\hat{\phi}_1)|$ is insignificant, the I -set has been fully determined at $\hat{\phi}_1$. If indices $|i|$ and $|j|$ have been both excluded from the I -set, then again the I -set has been fully determined at $\hat{\phi}_1$. In both previous cases, irrespectively of what the sorting of $|V_{|i|,:}\mathbf{c}(\hat{\phi}_1)|$ and $|V_{|j|,:}\mathbf{c}(\hat{\phi}_1)|$ is in the cell “led” by $\hat{\phi}_1$, we know that $|i|$ and $|j|$ will both remain inside or outside the I -set, respectively. However, a problem arises if exactly $K - 1$ elements of $|\mathbf{Vc}(\hat{\phi}_1)|$ are larger from $|V_{|i|,:}\mathbf{c}(\hat{\phi}_1)|$ and $|V_{|j|,:}\mathbf{c}(\hat{\phi}_1)|$. Then, in order to complete the I -set we have to choose one of $|i|$ and $|j|$, according to which of $|V_{|i|,:}\mathbf{c}(\hat{\phi}_1)|$ and $|V_{|j|,:}\mathbf{c}(\hat{\phi}_1)|$ is larger in the cell ”led“ by $\hat{\phi}_1$, but this cannot be determined at $\hat{\phi}_1$ since these two elements are equal at that point. Intuitively, we would like to move to $\hat{\phi}_1 + \epsilon$, $\epsilon > 0$ as depicted in Figure 4, i.e. ϵ into the cell, and check what the sorting is there. However, ϵ cannot be determined.

We have noted that any two two-dimensional hypersurfaces $\frac{i}{|i|}V_{|i|,:}\mathbf{c}(\phi_1)$ and $\frac{j}{|j|}V_{|j|,:}\mathbf{c}(\phi_1)$, with $|i| \neq |j|$ intersect at one point. So, if hypersurfaces $\frac{i}{|i|}V_{|i|,:}\mathbf{c}(\phi_1)$ and $\frac{j}{|j|}V_{|j|,:}\mathbf{c}(\phi_1)$ intersect at $\hat{\phi}_1$ this is their only intersection point, and thus in the following region they retain their sorting up to $\phi_1 = \frac{\pi}{2}$. However, we are interested in the sorting of their absolute

values rather than their actual sorting. Observing Figure 4, we see that the sorting of their absolute values can be deduced:

1. If at $\hat{\phi}_1$, the two intersecting curves have a positive value (point A), then due to continuity there is an $\epsilon > 0$ such that the curves remain positive at $\hat{\phi}_1 + \epsilon$. Thus, the greater by absolute value (i.e. the more positive one), is the curve that lies above the other at $\hat{\phi}_1 + \epsilon$ and will remain above up to $\phi_1 = \frac{\pi}{2}$ (points B and C). So, if $\frac{i}{|i|}V_{|i|,1:2}\mathbf{c}(\hat{\phi}_1) > 0$ then in the interior of the immediately following cell, and “near” the intersection point, $|V_{|i|,\cdot}\mathbf{c}(\phi_1)| > |V_{|j|,\cdot}\mathbf{c}(\phi_1)|$ if and only if $\frac{i}{|i|}V_{|i|,\cdot}\mathbf{c}(\frac{\pi}{2}) > \frac{j}{|j|}V_{|j|,\cdot}\mathbf{c}(\frac{\pi}{2})$.
2. If at $\hat{\phi}_1$, the two intersecting curves have a negative value (point A), then due to continuity there is an $\epsilon > 0$ such that the curves remain negative at $\hat{\phi}_1 + \epsilon$. Thus, the greater by absolute value (i.e. the more negative one), is the curve that lies below the other at $\hat{\phi}_1 + \epsilon$ and will remain below up to $\phi_1 = \frac{\pi}{2}$ (points B and C). So, if at $\hat{\phi}_1$, $\frac{i}{|i|}V_{|i|,1:2}\mathbf{c}(\hat{\phi}_1) < 0$ then in the interior of the immediately following cell, and “near” the intersection point, $|V_{|i|,\cdot}\mathbf{c}(\phi_1)| > |V_{|j|,\cdot}\mathbf{c}(\phi_1)|$ if and only if $\frac{i}{|i|}V_{|i|,\cdot}\mathbf{c}(\frac{\pi}{2}) < \frac{j}{|j|}V_{|j|,\cdot}\mathbf{c}(\frac{\pi}{2})$.
3. If at $\hat{\phi}_1$, $\frac{i}{|i|}V_{|i|,1:2}\mathbf{c}(\hat{\phi}_1) = \frac{j}{|j|}V_{|j|,1:2}\mathbf{c}(\hat{\phi}_1) = 0$, then also $|V_{|i|,\cdot}\mathbf{c}(\hat{\phi}_1)| = |V_{|j|,\cdot}\mathbf{c}(\hat{\phi}_1)| = 0$. At $\hat{\phi}_1$ the labeling function searches for the K largest elements of $|\mathbf{Vc}(\hat{\phi}_1)|$ which are by definition greater or equal to zero. Since $|V_{|i|,\cdot}\mathbf{c}(\hat{\phi}_1)| = |V_{|j|,\cdot}\mathbf{c}(\hat{\phi}_1)| = 0$, the probability of having other hypersurfaces of zero magnitude at $\hat{\phi}_1$ (i.e. other hypersurfaces intersecting at $\hat{\phi}_1$) is practically zero. So we assume that all other $N - 2$ hypersurfaces have a nonzero magnitude at $\hat{\phi}_1$. Therefore, an ambiguity can appear only if $K = N - 1$, when all indices that correspond to hypersurfaces of nonzero magnitude at $\hat{\phi}_1$ have been included in the I -set, and one of $|i|$ and $|j|$, which correspond to the two smallest elements of $|\mathbf{Vc}(\hat{\phi}_1)|$, also has to be included. However, we assume $K \ll N$ and we can, therefore, safely ignore this case.

Having described how ambiguities, when such appear, are resolved, we have fully described a way to calculate the I -set at any intersection point $\hat{\phi}_1$. The calculation costs $\mathcal{O}(N)$ for determining the K -th order element of $|\mathbf{V}_{N \times 2}\mathbf{c}(\hat{\phi}_1)|$ and then $\mathcal{O}(N)$ to determine the $K - 1$ elements larger than the K -th order element. Calculation ends here, unless an ambiguity appears. The discovery of an ambiguity requires checking if two indices $|i|$

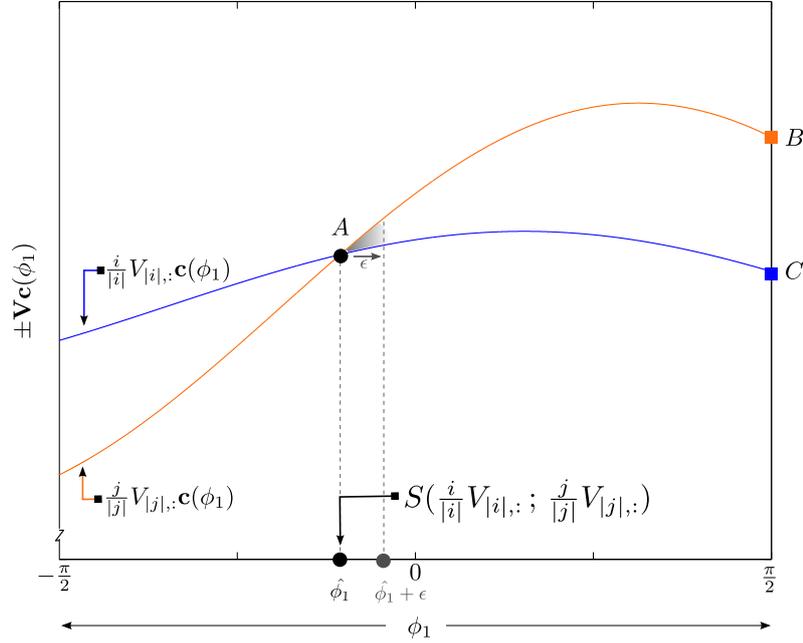


Figure 4: Rank-2: Intersection of two curves - Ambiguity Resolution.

and $|j|$ are inside the I -set and costs $\mathcal{O}(K)$ or $\mathcal{O}(D)$ depending on the implementation. Finally, if an ambiguity exists, resolving required $\mathcal{O}(D)$ calculations. So in total, we can determine the I -set at any point $\hat{\phi}_1$ in $\mathcal{O}(N)$.

Constructing $\mathcal{I}(V_{N \times 2})$ requires “calling” the labeling function for all $\binom{N^2-N}{D-1} = \binom{N^2-N}{2-1} = N^2 - N$ intersection points and hence the total complexity is $(N^2 - N) \times \mathcal{O}(N)$. However, note that since the calculations are independent among the intersection points, the construction can be fully parallelized.

VI.2.B Serial Implementation

The “Parallel” version of the algorithm calculates the I -set at all intersection points. However, we know that most intersection points appear in the interior of regions of fixed I -set and are only related to the reordering of the curves inside or outside the I -set. In order to avoid the unnecessary examination of many points we suggest an alternative to the “Parallel” version.

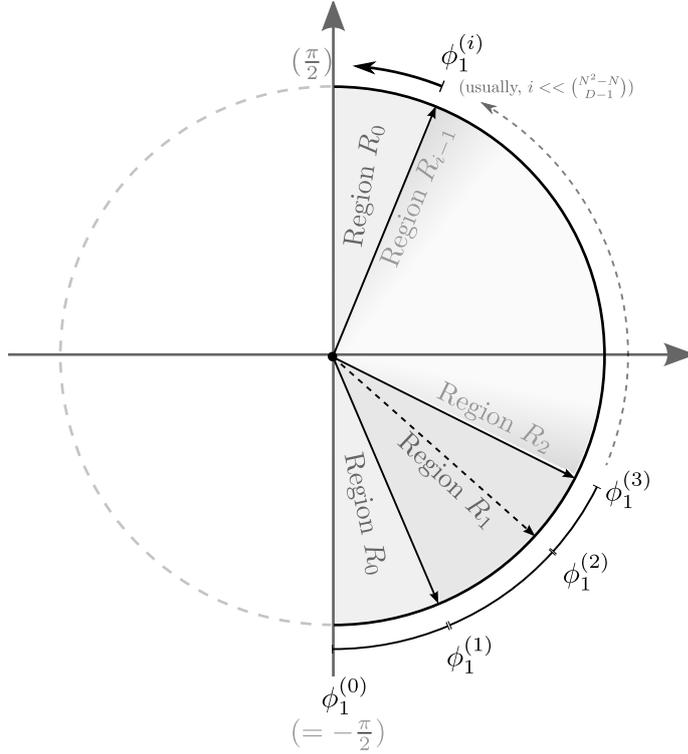


Figure 5: Rank-2: Serial scanning of the Φ^1 field.

We have seen how to calculate the I -set at a certain point ϕ'_1 in the Φ^1 field. Due to the continuity of the curves as functions of ϕ_1 we know that in an area close to ϕ'_1 the I -set will not change. The fundamental question behind the new implementation is when the I -set will change. Moving towards $\phi_1 = \frac{\pi}{2}$, curves intersect, but not all intersection are critical for determining the I -set. In order for an index $i, i \in \{1, \dots, N\}$, that is outside of the I -set at ϕ'_1 to be included in the I -set at some other point ϕ''_1 , the corresponding curve $|V_{i,:}c(\phi_1)|$ has to grow large enough to be among the K largest curves of $|\mathbf{V}c(\phi''_1)|$. In order to do that, due to continuity, curve $|V_{i,:}c(\phi_1)|$ will have to gradually reach the $(K + 1)$ -th position, intersect with a curve holding the K -th position at some point $\hat{\phi}_1$ and swap places with the latter, to obtain the K -th position. So a necessary condition for the I -set to change, in an area around ϕ'_1 is the K -th order curve to intersect. However, the K -th order curve may also intersect and switch places with the $(K - 1)$ -th curve, in which case the ordering of the curves whose indices are include in the I -set changes but the I -set itself does not. Therefore, the intersection of the K -th order curve is a necessary, but not a sufficient condition for the I to change. Even in this case, though, the intersection

is useful in order to keep track of what the new K -th order curve is, since in order to determine where the I -set actually changes, we need the intersections of the K -th curve.

The new algorithm, is as follows:

1. We call the labeling function at $\phi_1 = -\frac{\pi}{2}$ and obtain the index of the K -th order element of $|\mathbf{V}_{N \times 2} \mathbf{c}(\phi_1)|$ at $\phi_1 = -\frac{\pi}{2}$ as well as the other $K - 1$ indices included in the I -set. This is our first candidate index-set and can be immediately added to $\mathcal{I}(\mathbf{V}_{N \times 2})$.
2. We calculate the intersections of the K -th order waveform of $|\mathbf{V}_{N \times 2} \mathbf{c}(\phi_1)|$ with the other $N - 1$ waveforms described by the vector. The next point to be visited, $\phi_{1,\text{next}}$, is the intersection of $|V_{i,:} \mathbf{c}(\phi_1)|$ closest to $\phi_1 = -\frac{\pi}{2}$ in the direction towards $\phi_1 = \frac{\pi}{2}$.
3. We move at $\phi_{1,\text{next}}$ which now becomes $\phi_{1,\text{cur}}$. If $\phi_{1,\text{cur}}$ was the intersection of the current K -th order waveform, $|V_{i,:} \mathbf{c}(\phi_1)|$, with $|V_{j,:} \mathbf{c}(\phi_1)|$, $j \in \{1, \dots, N\}$, then j is the index of the new K -th order waveform. If j was previously not included in the I -set, then the latter changes and the new I -set has to be added to $\mathcal{I}(\mathbf{V}_{N \times 2})$. If j was previously included in the I -set, then no action need to be taken.
4. In any case, $|V_{j,:} \mathbf{c}(\phi_1)|$ is the new K -th order waveform. We calculate the $2(N - 1)$ intersections of $|V_{i,:} \mathbf{c}(\phi_1)|$ with the other $N - 1$ waveforms $|V_{*,:} \mathbf{c}(\phi_1)|$, where $* \in \{1, \dots, N\} - j$. The new $\phi_{1,\text{next}}$ point is the intersection of $|V_{j,:} \mathbf{c}(\phi_1)|$ closest to $\phi_{1,\text{cur}}$ in the direction towards $\phi_1 = \frac{\pi}{2}$.
5. We repeat from step 3 until $\phi_{1,\text{next}} \geq \frac{\pi}{2}$.

Note that the labeling function is only called once at the beginning of the algorithm to determine a reference I -set and mark the K -th order waveform at $\phi_1 = \frac{\pi}{2}$. Then the algorithm keeps track of the I -set by performing low cost swaps at the intersection points when necessary. Therefore, the calculation of the I -set at any visited point (apart from $\phi_1 = -\frac{\pi}{2}$) is actually limited to the modification of the I -set calculated at the previous visited point. Hence, in contrast to the “parallel” implementation, intersection points cannot be examined independently. Instead, the alternative version has to “serially” scan the Φ^1

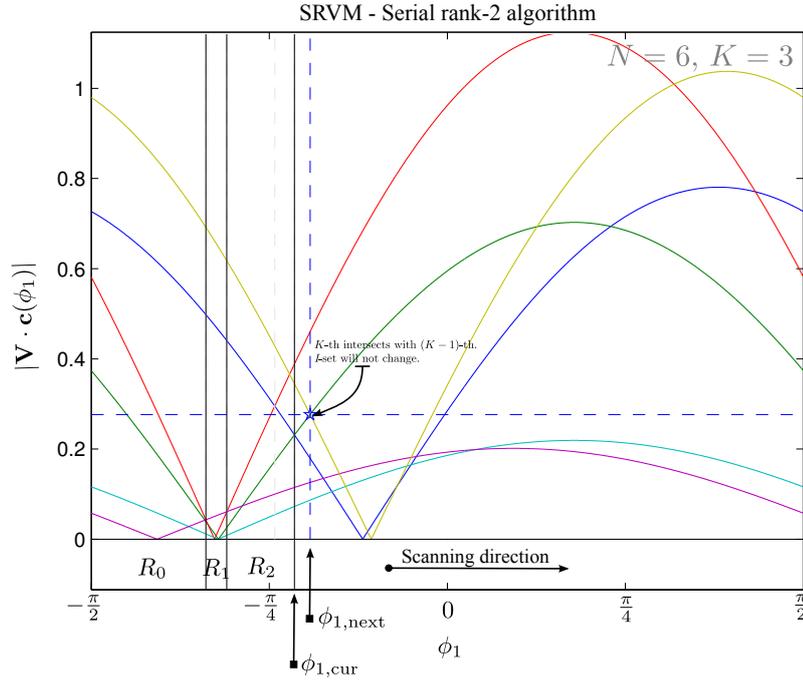


Figure 6: Rank-2: Serial Algorithm Execution Instance.

field as depicted in Figure 5 and, hence, the name of the algorithm.

Illustration Example 4

In Figure 6 we present an execution instance of the serial rank-2 algorithm on a random matrix $V_{6 \times 2}$ with $K = 3$. Starting from $\phi_1 = \frac{\pi}{2}$, all visited points are denoted by vertical lines: dashed lines denote intersection points at which the I -set did not change, while continuous black lines denote alteration of the I -set and thus the end of a region. Execution is now at $\phi_{1,cur}$. This point is the end of a region and the beginning of a new one, since a new index was inserted in the I -set. All the intersections of the new K -th order waveform are calculated and $\phi_{1,next}$ is the one closest to $\phi_{1,cur}$. At the new point, we see that the K -th order waveform intersects and swaps with the $(K - 1)$ -th order waveform and, thus, only an internal reordering of the waveforms included in the I -set will occur.

VI.2.C On the complexity of the two implementations

The fundamental motivation behind the serial algorithm was to minimize the number of examined intersection points by avoiding the unnecessary examination of points which would definitely not contribute more candidate index-sets to $\mathcal{I}(\mathbf{V}_{N \times 2})$. Unfortunately, in the parallel version, we cannot determine a-priori if a point's contribution to $\mathcal{I}(\mathbf{N} \times \mathbf{2})$, and we, therefore, have to blindly examine all intersection points. On the contrary, the serial scanning of the Φ^1 field, allows the serial version to select which points to examine according to information collected at each stage of the execution. Comparing the two alternative versions of the rank-2 case algorithm, we note the following:

- We have experimentally seen that the serial algorithm examines by far less points than the parallel version. At each visited point, the cost of calling the labeling function is dropped, since the latter is called once at the beginning of the execution. Instead, the serial version performs constant cost functions for the modification of the I -set and keeping track of the K -th waveform. In addition, in order to determine the which point to visit next, the algorithm performs $\mathcal{O}(N)$ calculations. Therefore, the cost at each visited point is still $\mathcal{O}(N)$.
- The main drawback of the serial algorithm is that it cannot be parallelized. The execution must be carried out serially since decisions at each state, depend on the previous states of the execution. On the contrary, decisions in the parallel version, are independent at each point. So the execution can be fully parallelized and utilize all available resources. In addition, depending on the implementation, the serial version can be more memory consuming.

In Figure 7 we plot the complexity curves for the two versions of the rank-2 algorithm versus the size N of the input for $K = \lceil \frac{N}{2} \rceil$. In the case of the parallel, complexity is equal to $(N^2 - N) \times \mathcal{O}(N)$. In the case of serial version, the complexity at each examined point is again $\mathcal{O}(N)$, but the number of points examined is not a deterministic function of N . Therefore, we plot an average complexity over many experimental executions. The

complexity of the parallel algorithm does not depend on K , however, in the case of the serial algorithm we have observed that the smaller the value of K , the fewer the points examined and therefore, the lower the complexity is. In Figure 8 we plot (a) the number of points examined by the parallel version, (b) the average number of points examined by the serial version, (c) the average number of regions in the Φ^1 field and (d) the number of distinct candidate index-sets (size of $\mathcal{I}(\mathbf{V}_{N \times 2})$), all versus the input size N for $K = \lceil \frac{N}{2} \rceil$. Curves (a) and (b) are associated with those of Figure 7 (scaled by N). The number of regions, is of course much smaller than the number of points examined by the two algorithms. Finally, the the number of distinct candidate vectors is orders of magnitude smaller than the number of points examined. Although the complexity of the algorithm for the construction of $\mathcal{I}(\mathbf{V}_{N \times 2})$ is determined by the latter, the the cardinality of the $\mathcal{I}(\mathbf{V}_{N \times 2})$ is also significant, since the final solution to the maximization problem will be numerically calculated with exhaustive search among the index-sets contained in $\mathcal{I}(\mathbf{V}_{N \times 2})$. Once again, note that the smaller the value of K , the fewer the regions and, consequently, the fewer the distinct candidate index-sets.

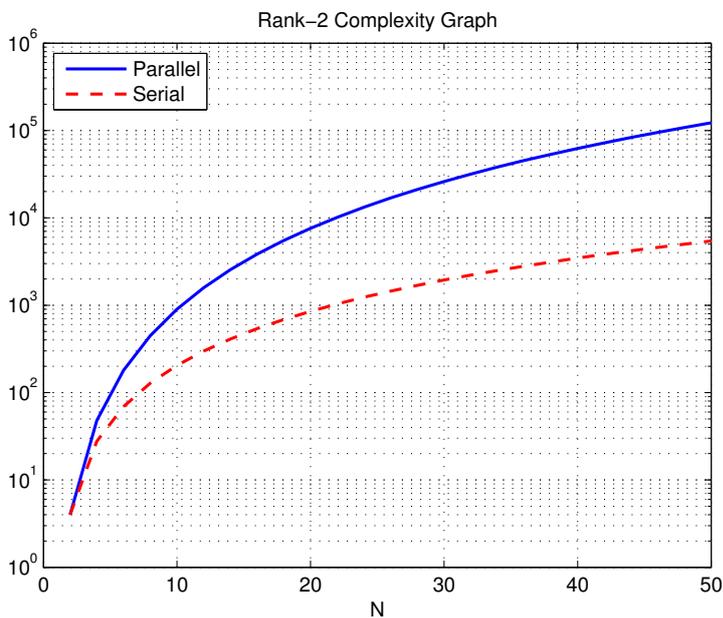


Figure 7: Rank-2: Algorithm Complexity

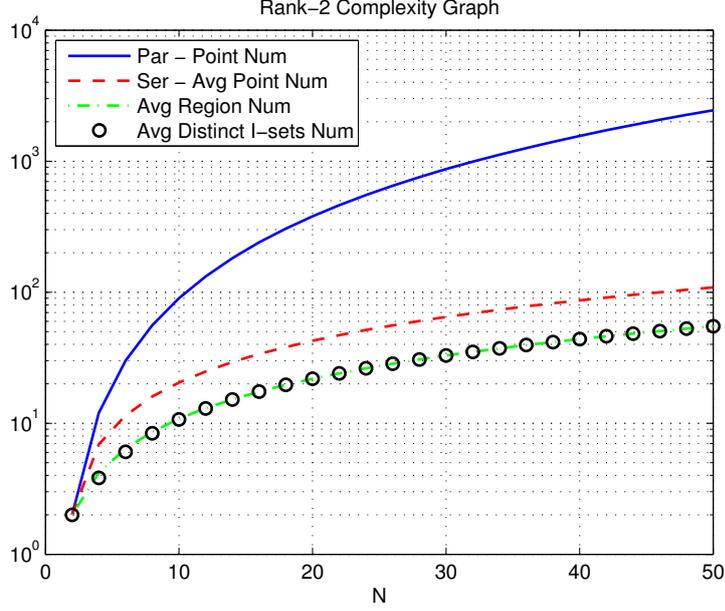


Figure 8: Rank-2: Points examined, Regions, Distinct I -sets.

VI.3 Rank-3 Case

In the case $\mathcal{R}(\mathbf{A}) = 3$ (or, equivalently, $D = \mathcal{R}(\mathbf{V}) = 3$), the spherical coordinate vector is a two-element vector, $\phi_{1:2} = [\phi_1, \phi_2]$ and the auxiliary hyperpolar vector $\mathbf{c}(\phi_{1:2})$ is

$$\mathbf{c}(\phi_{1:2}) \triangleq \begin{bmatrix} \sin \phi_1 \\ \cos \phi_1 \sin \phi_2 \\ \cos \phi_1 \cos \phi_2 \end{bmatrix}. \quad (44)$$

Every element of vector $|\mathbf{V}\mathbf{c}(\phi_{1:2})|$ corresponds to a continuous function of ϕ which describes a three-dimensional hypersurface:

$$|V_{i,1:3}\mathbf{c}(\phi_{1:2})| = |V_{i,1} \sin \phi_1 + V_{i,2} \cos \phi_1 \sin \phi_2 + V_{i,3} \cos \phi_1 \cos \phi_2|, \quad \forall i \in \{1, \dots, N\}. \quad (45)$$

Of course,

$$\pm V_{i,1:3}\mathbf{c}(\phi_{1:2}), \quad \forall i \in \{1, \dots, N\} \quad (46)$$

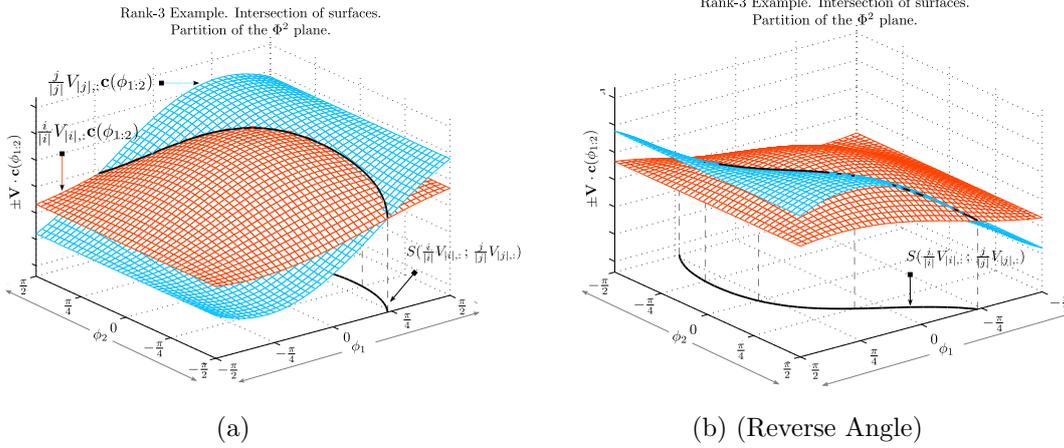


Figure 9: Rank-3: Hypersurfaces described by the elements of $|\mathbf{V}\mathbf{c}(\phi_{1:2})|$ and their intersection.

are also three-dimensional surfaces. The intersection of any two surfaces described by (46), is a curve and its projection on the Φ^2 plane, $S(\frac{i}{|i|}V_{|i|,:}; \frac{j}{|j|}V_{|j|,:})$ with $|i| \neq |j|$, $|i|, |j| \in \{1, \dots, N\}$, is a two-dimensional curve which partitions the Φ^2 plane into two regions. According to the theoretic developments, we take combinations of $D - 1 = 3 - 1 = 2$ hypersurfaces $S(\cdot; \cdot)$ to obtain points leading cells in Φ^2 .

Our goal, is the calculation of $\mathcal{I}(\mathbf{V}_{N \times 3})$. We recall, from (37), that the construction of $\mathcal{I}(\mathbf{V}_{N \times d})$ can also be fully parallelized, since the candidate index-set, $I(\mathbf{V}_{N \times d}; P_{d-1})$, can be computed *independently* for each set P_{d-1} which corresponds to a point in Φ^{d-1} . As a result, we need to present a method for the computation of $I(\mathbf{V}_{N \times 3}; P_2)$. We remind that, according to the above, in the case of rank-3 each P_2 set consist of two pairs $\{\{p_{1,1}, p_{1,2}\}, \{p_{2,1}, p_{2,2}\}\}$ where $|p_{1,1}| \neq |p_{1,2}|$, $|p_{2,1}| \neq |p_{2,2}|$ and $|p_{*,*}| \in \{1, \dots, N\}$. The two pairs describe two hypersurfaces $S(\frac{p_{1,1}}{|p_{1,1}|}V_{|p_{1,1}|,:}; \frac{p_{1,2}}{|p_{1,2}|}V_{|p_{1,2}|,:})$ and $S(\frac{p_{2,1}}{|p_{2,1}|}V_{|p_{2,1}|,:}; \frac{p_{2,2}}{|p_{2,2}|}V_{|p_{2,2}|,:})$ respectively, and therefore the P_2 set corresponds to an intersection point on the Φ^2 plane.

In the following we will present such a method for the computation of $I(\mathbf{V}_{N \times 3}; \hat{\phi}_{1:2})$ at each intersection point $\hat{\phi}_{1:2}$. This method, allows the parallel construction of $\mathcal{I}(\mathbf{V}_{N \times 3})$ and is, hence, referred to as “parallel” implementation. Then we will present an alternative method for the construction of $\mathcal{I}(\mathbf{V}_{N \times 3})$ that sacrifices parallelization in order to achieve a lower complexity like the case of rank-2. The latter is based on serial scanning of the

Φ^2 -plane and is, hence, referred to as “serial” implementation.

VI.3.A Parallel Implementation

At each intersection point $\hat{\phi}_{1:2}$, we try to obtain the I -set associated with the cell originating at $\hat{\phi}_{1:2}$ by calling the labeling function $I(\mathbf{V}_{N \times 3}; \hat{\phi}_1)$ which marks the K elements of the I -set, i.e. the indices of the K largest elements of $|\mathbf{Vc}(\hat{\phi}_{1:2})|$. Point $\hat{\phi}_{1:2}$ is defined as the intersection of two two-dimensional hypersurfaces $S(\frac{i}{|i|}V_{|i|,:}; \frac{j}{|j|}V_{|j|,:})$ and $S(\frac{k}{|k|}V_{|k|,:}; \frac{l}{|l|}V_{|l|,:})$, as depicted in Figure 10. According to the theoretic developments, there are in total $\binom{N^2-N}{D-1} = \binom{N^2-N}{2}$ intersection points, which can all be blindly examined. However, it turns out that in the case of rank-3, we can significantly reduce the number of points examined. In the following, we will first describe which are the points of interest in the Φ^2 plane and we will then proceed with the process of determining the I -set at each point.

As noted, in the case of rank-3, $\frac{i}{|i|}\mathbf{V}_{|i|}\mathbf{c}(\phi_{1:2})$, $i \in \{-N, \dots, -1, 1, \dots, N\}$ corresponds to a three-dimensional surface. The projection of the intersection of any two such surfaces $\frac{i}{|i|}\mathbf{V}_{|i|}\mathbf{c}(\phi_{1:2})$ and $\frac{j}{|j|}\mathbf{V}_{|j|}\mathbf{c}(\phi_{1:2})$ with $|i| \neq |j|$ on the Φ^2 -plane, $S(\frac{i}{|i|}V_{|i|,:}; \frac{j}{|j|}V_{|j|,:})$, is a two-dimensional curve. Any two such curves intersect at one point $\hat{\phi}_{1:2}$. There are $\binom{N^2-N}{D-1} = \binom{N^2-N}{2}$ intersection points, leading cells within which the sorting of $|\mathbf{Vc}(\phi_{1:2})|$ is fixed. More than one cells in the same neighborhood may correspond to the same I -set and then they form a “region”. We would like to avoid the examination of all intersection points leading cells that belong to the same region, since all these points are associated with the same I -set and cannot contribute any extra candidate index-set to $\mathcal{I}(\mathbf{V}_{N \times 3})$. We observed that among the cells that form a region, one cell will be the souther cell and its leading vertex can be considered as the leading vertex of the whole region. In addition, the two borders of this cell will also be the region’s borders “near” its leading vertex. The key observation, is the following: “near” the leading vertex of a region, the surfaces of $|\mathbf{Vc}(\phi_{1:2})|$ have a certain sorting, and a certain surface is the K -th order surface. If $|i|$ is the index of this K -th order surface, both left and right borders of the region “near” the leading vertex, have to be of the form $S(\frac{i}{|i|}V_{|i|,:}; \frac{*}{|*|}V_{|*|,:})$ where $|*|$ is not included in the I -set of a region. Had $|*|$ been included in the I -set, then over $S(\frac{i}{|i|}V_{|i|,:}; \frac{*}{|*|}V_{|*|,:})$ only an internal swapping of surfaces would occur and the curve would not be a region-border.

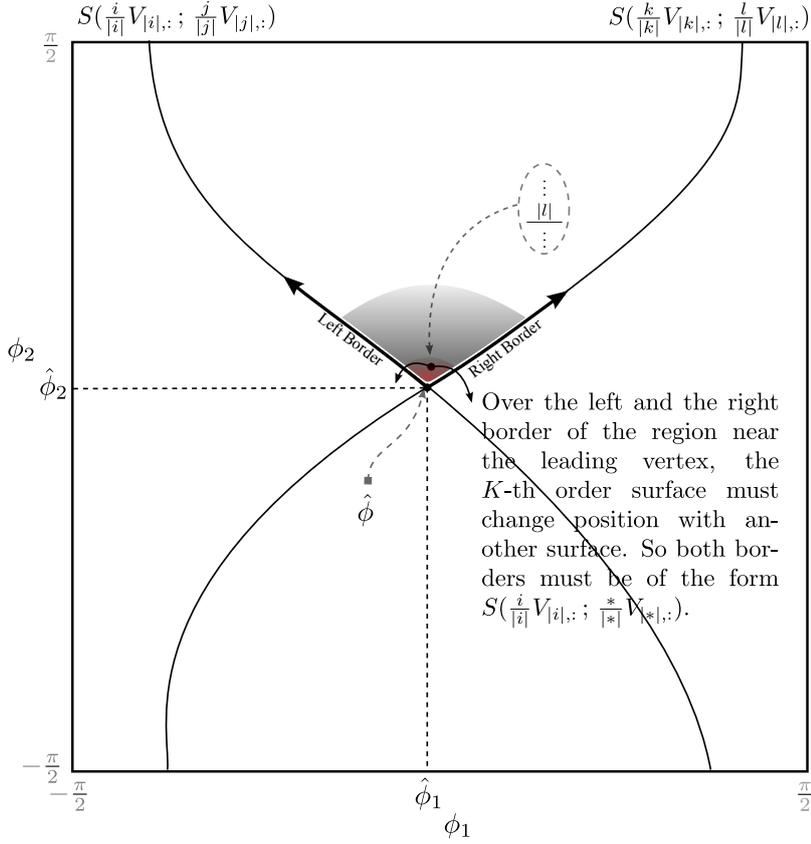


Figure 10: Rank-3: Intersection of two hypersurfaces.

Hence, we expect that “leading-vertices” of regions, are points where both intersecting curves are of the form $S(\frac{i}{|i|} V_{|i|, :}; \frac{*}{|*|} V_{|*|, :})$ ⁶, where $|i|$ is the index of the K -th order surface in the interior of the region, “near” the leading vertex.

Therefore, we can only examine intersection points of this form, which are potential region-leading vertices. Although it is possible that not all such points lead a region, based on the previous observation we avoid the examination of point where the intersecting curves are of the form $S(\frac{i}{|i|} V_{|i|, :}; \frac{j}{|j|} V_{|j|, :})$ and $S(\frac{k}{|k|} V_{|k|, :}; \frac{l}{|l|} V_{|l|, :})$, with $|i| \neq |j| \neq |k| \neq |l|$. Such intersection points lie in the interior of regions, and can, therefore, contribute nothing more to $\mathcal{I}(\mathbf{V}_{N \times 3})$. We also note the following:

When two curves $S(\frac{i}{|i|} V_{|i|, :}; \frac{j}{|j|} V_{|j|, :})$ and $S(\frac{i}{|i|} V_{|i|, :}; \frac{k}{|k|} V_{|k|, :})$ intersect at one point $\hat{\phi}_{1,2}$,

⁶Note that the two borders may also be of the form $S(\frac{i}{|i|} V_{|i|, :}; \frac{*}{|*|} V_{|*|, :})$ and $S(\frac{-i}{|-i|} V_{|-i|, :}; \frac{*}{|*|} V_{|*|, :})$, but since $S(\frac{-i}{|-i|} V_{|-i|, :}; \frac{*}{|*|} V_{|*|, :}) = S(\frac{i}{|i|} V_{|i|, :}; \frac{-*}{|-*|} V_{|-*|, :})$, we write only the general case for simplicity.

then a third curve $S(\frac{j}{|j|}V_{|j|,:}; \frac{k}{|k|}V_{|k|,:})$ also intersects with the previous two at the same point, unless $k = -j$ in which case no third curve is implied. Hence,

- If $k = -j$, then at the intersection point $\hat{\phi}_{1:2}$ we would have $\frac{i}{|i|}V_{|i|,:}\mathbf{c}(\hat{\phi}_{1:2}) = \frac{j}{|j|}V_{|j|,:}\mathbf{c}(\hat{\phi}_{1:2}) = -\frac{j}{|j|}V_{|j|,:}\mathbf{c}(\hat{\phi}_{1:2})$ where the last equality implies that $|\frac{i}{|i|}V_{|i|,:}\mathbf{c}(\hat{\phi}_{1:2})| = |\frac{j}{|j|}V_{|j|,:}\mathbf{c}(\hat{\phi}_{1:2})| = 0$. Since the labeling function searched for the K largest elements of $|\mathbf{Vc}(\hat{\phi}_{1:2})|$ when called at the intersection point, zero-valued surfaces would never be chosen among the K largest surfaces, unless $K \geq N - 1$. Since, we assume that $K \ll N$, we can ignore this case.
- According to the previous observation, we are only interested in intersection points where $|i| \neq |j| \neq |k|$. Every set of 3 indices gives 4 such intersection point. For example, if indices 1, 2 and 3 give:

$$\begin{aligned} +V_{1,:}\mathbf{c}(\hat{\phi}_{1:2}) &= +V_{2,:}\mathbf{c}(\hat{\phi}_{1:2}) = +V_{3,:}\mathbf{c}(\hat{\phi}_{1:2}) \\ -V_{1,:}\mathbf{c}(\hat{\phi}_{1:2}) &= +V_{2,:}\mathbf{c}(\hat{\phi}_{1:2}) = +V_{3,:}\mathbf{c}(\hat{\phi}_{1:2}) \\ +V_{1,:}\mathbf{c}(\hat{\phi}_{1:2}) &= -V_{2,:}\mathbf{c}(\hat{\phi}_{1:2}) = +V_{3,:}\mathbf{c}(\hat{\phi}_{1:2}) \\ +V_{1,:}\mathbf{c}(\hat{\phi}_{1:2}) &= +V_{2,:}\mathbf{c}(\hat{\phi}_{1:2}) = -V_{3,:}\mathbf{c}(\hat{\phi}_{1:2}) \end{aligned}$$

To summarize the above, we have to examine 4 intersection points for each subset of $\{1, \dots, N\}$ of size 3. There are $\binom{N}{3}$ such subsets, so we will eventually examine $4\binom{N}{3}$ intersection points. These, for sufficiently large N , are significantly fewer points compared to the $\binom{N^2-N}{D-1} = \binom{N^2-N}{2}$ we would blindly examine.

Recall, that in the case of rank-2, we had to examine all intersection points, which were equal to $\binom{N^2-N}{D-1} \stackrel{D=2}{=} N^2 - N$. In rank-3 we reduced the number of “interesting” intersection points down to $4\binom{N}{3}$. We observe that:

- $D = 2$: $N^2 - N = \binom{N}{2} 2^1 = \binom{N}{D} 2^{D-1}$
- $D = 3$: $4\binom{N}{3} = \binom{N}{3} 2^2 = \binom{N}{D} 2^{D-1}$

Therefore, in both rank-2 and rank-3 cases, we have to examine $\binom{N}{D} 2^{D-1}$ points. This is probably a good hint on discovering the points we need to consider in higher rank cases! Furthermore, this number is a new upper bound for the size of $\mathcal{I}(\mathbf{V}_{N \times D})$, $D = 2, 3$.

Up to now we have determined the points that need to be examined and it remains to describe how we obtain the I -set associated with each point $\hat{\phi}_{1:2}$. The first step is, of course, to call the labeling function at $\hat{\phi}_{1:2}$. However, at $\hat{\phi}_{1:2}$ we have $|V_{|i|,:\mathbf{c}(\hat{\phi}_1)}| = |V_{|j|,:\mathbf{c}(\hat{\phi}_1)}| = |V_{|k|,:\mathbf{c}(\hat{\phi}_1)}|$ where $|i|$, $|j|$ and $|k|$ are the indices of the surfaces whose pairwise intersection-curves all meet at $\hat{\phi}_{1:2}$. So if indices $|i|$, $|j|$ and $|k|$ are all included in the I -set, then since internal ordering of the K first elements of $|\mathbf{Vc}(\hat{\phi}_{1:2})|$ is insignificant, the I -set has been fully determined at $\hat{\phi}_{1:2}$. Similarly, if all three indices have been excluded from the I -set, then again the I -set has been fully determined at $\hat{\phi}_{1:2}$ and no further action is required. However, a problem arises if any of the three indices is included in the I -set while others are excluded or vice-versa. Then, we need to determine the “ordering” of $|V_{|i|,:\mathbf{c}(\phi_{1:2})}|$, $|V_{|j|,:\mathbf{c}(\phi_{1:2})}|$ and $|V_{|k|,:\mathbf{c}(\phi_{1:2})}|$ in the interior of the cell determine which indices should be included in the I -set. This information cannot be deduced at $\hat{\phi}_{1:2}$ though.

Let’s see how ambiguity is resolved. Take, for example, the $\{|i|, |k|\}$ pair: we want to determine which one of $|V_{|i|,:\mathbf{c}(\phi_{1:2})}|$ and $|V_{|k|,:\mathbf{c}(\phi_{1:2})}|$ is greater in the interior of the region. Observe, referring to Figure 10 that surfaces $\frac{i}{|i|}V_{|i|,:\mathbf{c}(\phi_{1:2})}$ and $\frac{k}{|k|}V_{|k|,:\mathbf{c}(\phi_{1:2})}$ only intersect along $S(\frac{i}{|i|}V_{|i|,:\mathbf{c}(\phi_{1:2})}; \frac{k}{|k|}V_{|k|,:\mathbf{c}(\phi_{1:2})})$. So in the interior of the region, i.e. towards the other two curves that intersect at $\hat{\phi}_{1:2}$, the two surfaces $\frac{i}{|i|}V_{|i|,:\mathbf{c}(\phi_{1:2})}$ and $\frac{k}{|k|}V_{|k|,:\mathbf{c}(\phi_{1:2})}$ retain their sorting. So for example, in Figure 10, if $\frac{i}{|i|}V_{|i|,:\mathbf{c}(\phi_{1:2})} > \frac{k}{|k|}V_{|k|,:\mathbf{c}(\phi_{1:2})}$ in the interior of the region “near” the intersection point at the side of the curve $S(\frac{i}{|i|}V_{|i|,:\mathbf{c}(\phi_{1:2})}; \frac{j}{|j|}V_{|j|,:\mathbf{c}(\phi_{1:2})})$, then

$\frac{i}{|i|}V_{|i|,:}\mathbf{c}(\phi_{1:2}) > \frac{k}{|k|}V_{|k|,:}\mathbf{c}(\phi_{1:2})$ for all points of the cell “near” $S(\frac{i}{|i|}V_{|i|,:}; \frac{j}{|j|}V_{|j|,:})$, up to point A . So we can check the relationship of $\frac{i}{|i|}V_{|i|,:}\mathbf{c}(\phi_{1:2})$ and $\frac{k}{|k|}V_{|k|,:}\mathbf{c}(\phi_{1:2})$ at point A . Like in the case of rank-2, utilizing the information about the sign of the value of the surfaces $\frac{i}{|i|}V_{|i|,:}\mathbf{c}(\phi_{1:2})$ and $\frac{k}{|k|}V_{|k|,:}\mathbf{c}(\phi_{1:2})$ at the intersection point, as well as their sorting at A , we can deduce the sorting of their absolute values “near” the intersection point at the side of $S(\frac{i}{|i|}V_{|i|,:}; \frac{j}{|j|}V_{|j|,:})$:

1. If $\frac{i}{|i|}V_{|i|,1:3}\mathbf{c}(\hat{\phi}_{1:2}) > 0$ then in the interior of the cell “near” the intersection point at the side of $S(\frac{i}{|i|}V_{|i|,:}; \frac{j}{|j|}V_{|j|,:})$, $|V_{|i|,1:3}\mathbf{c}(\phi_{1:2})| > |V_{|k|,:}\mathbf{c}(\phi_{1:2})|$ if and only if $\frac{i}{|i|}V_{|i|,1:3}\mathbf{c}(\phi_A) > \frac{k}{|k|}V_{|k|,:}\mathbf{c}(\phi_A)$.
2. If $\frac{i}{|i|}V_{|i|,1:3}\mathbf{c}(\hat{\phi}_{1:2}) < 0$ then in the interior of the cell “near” the intersection point at the side of $S(\frac{i}{|i|}V_{|i|,:}; \frac{j}{|j|}V_{|j|,:})$, $|V_{|i|,1:3}\mathbf{c}(\phi_{1:2})| > |V_{|k|,:}\mathbf{c}(\phi_{1:2})|$ if and only if $\frac{i}{|i|}V_{|i|,1:3}\mathbf{c}(\phi_A) < \frac{k}{|k|}V_{|k|,:}\mathbf{c}(\phi_A)$.
3. If $\frac{i}{|i|}V_{|i|,1:3}\mathbf{c}(\hat{\phi}_{1:2}) = 0$, we can be sure that an ambiguity cannot appear, for reasons that have been thoroughly explained above as well as in section VI.2, for the Rank-2 case.

In the same way, we can determine the ordering for the $\{|j|, |k|\}$ pair at point A . So we know two binary relationships at A which are also valid for the interior of the cell, “near” the intersection point at the side of $S(\frac{i}{|i|}V_{|i|,:}; \frac{j}{|j|}V_{|j|,:})$. However, at A we cannot decide about the relationship of $|V_{|i|,1:3}\mathbf{c}(\hat{\phi}_{1:2})|$ and $|V_{|j|,:}\mathbf{c}(\hat{\phi}_{1:2})|$ since we are on the intersection of these three-dimensional surfaces. This decision can be taken on the same way, at one of the points B and C .

Since we do not know the relative position of the three curves (unless we compare points A , B and C), we can visit all points A , B and C and at each determine the binary relationships that can be determined at that point, regarding the ambiguities caused by the other two curves. Then we will have $3 \cdot 2 = 6$ binary “ordering” relationships. According to these relationships we can finally determine which indices of $|i|$, $|j|$ and $|k|$ have to be included in the I -set. If conflicting relationships appear, we can create all the candidate index-sets implied by the binary ordering-relationships and add the all to $\mathcal{I}(\mathbf{V}_{N \times 3})$. In fact the points that are leading vector of regions, will only give one candidate index-set, i.e. the curve in the middle will denote a rearrangement of surfaces either in or out of the I -set. If two

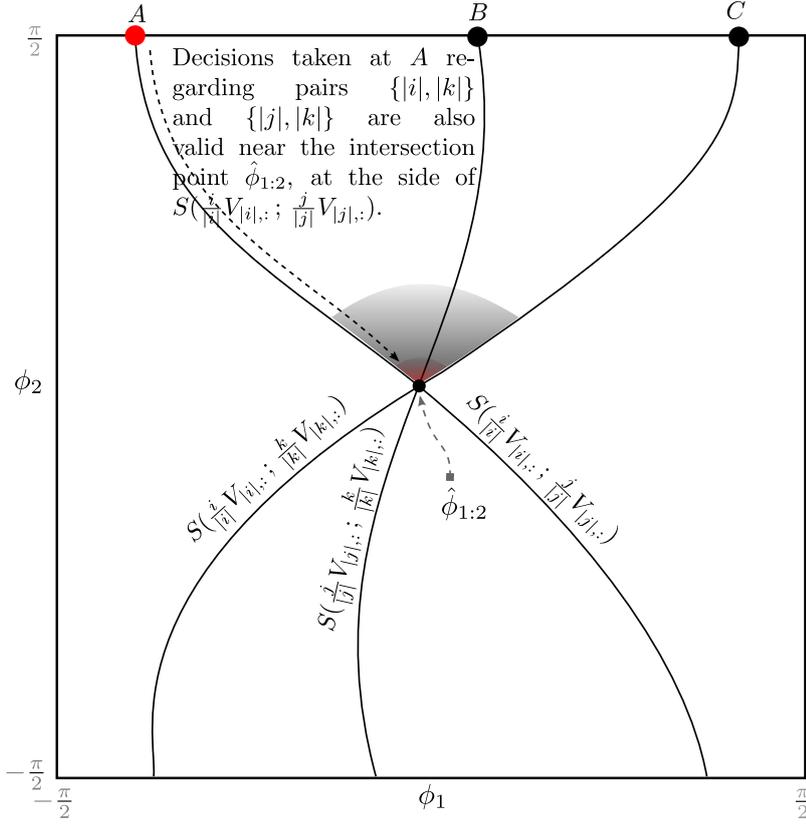


Figure 11: Rank-3: Intersection point - Ambiguity resolution.

candidate index-sets are obtained, it means that two regions meet at the intersection point and lie at the two sides of the middle curve. Then instead of adding both to $\mathcal{I}(\mathbf{V}_{N \times 3})$, we can ignore them since the same index-sets will be obtained at the leading vertices of these two regions.

We now have fully described how the I -set is determined at an intersection point $\hat{\phi}_{1:2}$. Determining the K -th order element of $|\mathbf{V}_{N \times 3} \mathbf{c}(\hat{\phi}_{1:2})|$ as well as the $K - 1$ elements larger than the K -th order element costs $\mathcal{O}(N)$. Discovering an ambiguity requires checking if the three indices $|i|$, $|j|$ and $|k|$ are all inside or all outside of the I -set, while its resolution requires $\mathcal{O}(1)$ calculations. In total, we can determine the I -set at any point $\hat{\phi}_{1:2}$ in $\mathcal{O}(N)$. Constructing $\mathcal{I}(\mathbf{V}_{N \times 3})$ requires calling the labeling function for all $4 \binom{N}{3}$ intersection points. Therefore, the total complexity of the rank-3 Parallel version is $4 \binom{N}{3} \mathcal{O}(N)$. Note, however, that since the calculation of the I occurs independently at each point, the construction of $\mathcal{I}(\mathbf{V}_{N \times 3})$ can be fully parallelized.

VI.3.B Serial Implementation

In the rank-2 case, we serially scanned the Φ^1 field from $\phi_1 = -\frac{\pi}{2}$ to $\phi_1 = \frac{\pi}{2}$, trying to determine the regions of fixed I -set. Regions were intervals on the ϕ_1 -axis and their borders were single points. At any point ϕ_1' there was one active region associated with a certain I -set and the point to be visited next was determined by the intersections of the K -th order element of vector $|\mathbf{Vc}(\phi_1)|$ which was a two-dimensional curve. The K -th order curve could, of course, change in the interior of a region.

The rank-3 case is more complex: elements of $|\mathbf{Vc}(\phi_{1:2})|$ are surfaces, and the projection of their intersections on the Φ^2 plane are two dimensional curves. Regions of fixed I -set expand over one or more cells and their borders are parts of these two dimensional curves. The K -th order surface can vary in the interior of the region, but the I -set does not change. In the serial version of rank-3, we scan the Φ^2 plane, moving a horizontal line parallel to the ϕ_1 -axis from $\phi_2 = -\frac{\pi}{2}$ to $\phi_2 = \frac{\pi}{2}$. At any value of ϕ_2 , the horizontal scanning-line intersects with more than one regions. So, contrary to the rank-2 case, the serial scanner of the Φ^2 plane must keep (in parallel) track of the evolution of more than one regions. The “region” as an entity can be sufficiently described by (a) its I -set, (b) its left and right borders and (c) the two indices of the K -th order surfaces “near” the left and right borders respectively. The region borders are two-dimensional curves of the form $S(\frac{i}{|i|}V_{|i|,:}; \frac{j}{|j|}V_{|j|,:})$, $|i|, |j| \in \{1, \dots, N\}$. Obviously, if $S(\frac{i}{|i|}V_{|i|,:}; \frac{j}{|j|}V_{|j|,:})$ is the left border of a region, one of $|i|, |j|$ is the index of the K -th order surface of this region “near” its left border, while the other is the index of the K -th order surface near the right border of the adjacent region.

To keep track of the region, we need to know when its borders change, which is equivalent to the modification of the left or right K -th order surface. If $|i|$ is the index of the K -th order surface “near” a border, then this border is of the form $S(\frac{i}{|i|}V_{|i|,:}; \frac{*}{|*|}V_{|*|,:})$. Moving along this border, towards $\phi_2 = \frac{\pi}{2}$, remaining in interior of a region, the K -th order surface will eventually swap with another surface, over a curve of the form $S(\frac{i}{|i|}V_{|i|,:}; \frac{*}{|*|}V_{|*|,:})$. Over the new curve, the region may change or not, depending on the new K -th order surface and whether its index was previously included in the I -set. In any case however, the K -th order surface will change and the corresponding border will also be modified. Therefore, modifications of regions occur at points where both intersecting curves are

of the form $S(\frac{i}{|i|}V_{|i|,:}; \frac{*}{|*|}V_{*|,:})$ where $|i|$ is the index of the K -th order surface “near” the border. This was expected, since these were the points also examined by the parallel version. There, we explained that each such point is the intersection of three curves, unless $j = -k$. Contrary to the parallel version, the serial one must also handle this case in order to keep track of the regions’ evolution. Finally, we note that due to the fact that adjacent regions share a common border, we can keep track only of the left border of each region. This way we also examine the right border of the region on its left.

The suggested algorithm is as follows:

1. We begin, by running a serial rank-2 algorithm on matrix $[V_{:,1} - V_{:,2}]$ which is a concatenation of the first column of \mathbf{V} and the opposite of its second column. This is equivalent to identifying the regions formed at the bottom of the Φ^2 plane, at $\phi_2 = -\frac{\pi}{2}$. In Figure 16 which depicts an execution instance of the serial rank-3 algorithm, the red markers at the bottom plane, show the region border-points determined by the rank-2 serial algorithm. If curves $|i|$ and $|j|$ intersect at a marker, then this implies that the projection of the intersection of surfaces $|i|$ and $|j|$ in the rank-3 problem, $S(\frac{i}{|i|}V_{|i|,:}; \frac{j}{|j|}V_{|j|,:})$, originates from this marker and is the border of a region near the bottom of the Φ^2 plane. Therefore, we obtain a list of active regions at $\phi_2 = -\frac{\pi}{2}$. Each region is described by its I -set, its two borders, and the indices of the K -th order surface near the left and right borders respectively. Note that the last region, that contains uncountably many points of the form $(\frac{\pi}{2}, \phi_2)$ is considered the same region as the one that contains uncountably many points of the form $(-\frac{\pi}{2}, \phi_2)$. The algorithm will gradually scan the Φ^2 plane, keeping track of the evolution of all active regions.
2. The ϕ_2 value to be visited next, $\phi_{2,\text{next}}$, is the level at which the active region list has to be modified (region removal, insertion or modification). Modifications occur at the intersection points of borders, as described above. For each one of the active regions, we calculate the $2(N-1)$ intersections of its left-border $S(\frac{i}{|i|}V_{|i|,:}; \frac{j}{|j|}V_{|j|,:})$, with all other curves of the form $S(\frac{i}{|i|}V_{|i|,:}; \frac{*}{|*|}V_{*|,:})$ where $|*| \notin \{|i|, |j|\}$ and $|i|$ is the index of K -th order surface of the region “near” the left border. Among all these points, we mark the intersection point whose ϕ_2 coordinate is closest to $\phi_{2,\text{cur}}$ value. This is the nearest modification point, and no alteration occurs on the regions between the current level, $\phi_{2,\text{cur}}$ and the next level $\phi_{2,\text{next}}$.

3. Having determined the $\phi_{2,\text{next}}$ value, we must determine the kind of modification that occurs in the intersection point of interest and then update the active region list accordingly. The examination of an intersection point is thoroughly described in the next paragraph. Every time a new border appears to any of the cells, its intersections must be calculated to be taken under consideration in the next iteration for the determination of the new $\phi_{2,\text{next}}$ value.
4. $\phi_{2,\text{next}}$ is now $\phi_{2,\text{cur}}$. We repeat from step 2 until $\phi_{2,\text{next}} \geq \frac{\pi}{2}$.

It remains to describe, how the regions are modified at an intersection point. In the following description, the border that intersects at the chosen intersection point $(*, \phi_{2,\text{next}})$ is referred to as the *Reference Border*. The region on its right, i.e. the region whose left-border is the *Reference Border*, is referred to as the *Current Region*. The regions on its left and right, are the *Previous* and *Next* regions respectively. The curve intersecting the *Reference Border* is the *Intersecting Curve*, while the third curve that did not explicitly participate in determining the intersection point, but we know that goes through it, is the *Induced Curve*. We note that, if more than two borders intersect at the chosen intersection point, we choose the left-most border to be the *Reference-Border* and of course the corresponding region as the *Current Region*. This choice will simplify the examination of the intersection point. Finally, in the illustrating Figures 12 to 15, $d(l, m)$ will stand for the *Reference Border*, with $|l|$ being the index of the K -th order surface of the *Current Region* near the *Reference Border*. $d(l, n)$ stands for the *Intersecting curve* and $d(m, n)$ is the *Induced Curve*.

At the intersection point, we first check if $n = -m$:

- If yes, we are in the special case depicted in Figure 12. The case is referred to as special, since it is the only case in which only two curves intersect at the intersection point. The two curves are the borders of the *Current Region*. These borders are swapped and the right border of the *Previous Region* as well as the left border of the *Next Region* are modified accordingly. Although the left and right borders of the *Current Region* intersect, and we could thus consider that the *Current Region* is terminated and should be removed from the active-region list, we observe that the region actually continues above the intersection point (*I*-set does not change).

- If not, then the intersection point is the intersection of three curves, $d(l, m)$, $d(l, n)$ and $d(m, n)$. The actions taken, depend on the following cases. We first check whether the I -set changes over the *Intersecting curve*, $d(l, n)$, “near” the intersection point. In other words, we check whether $|n|$ which swaps places with $|i|$, was not a member of the I -set of the *Current Region*.

A If yes, then region changes over the *Intersecting Curve*, $d(l, n)$. There two sub-cases, depending on whether the $d(l, n)$ is the right border of the *Current Region* or not:

A.1 If yes, we are in sub-case *A.1*, depicted in Figure 13a. The *Current Region* is terminated since its left and right borders intersect. According to the rearrangements of the sorting of $|Vc(\phi)|$ we derive that the *Induced curve*, $d(m, n)$, can only be in the interior of the *Current Region*. Above the intersection point, the *Previous* and *Next* regions meet at their new common border: $d(m, n)$.

A.2 If no, we are in sub-case *A.2*, depicted in Figure 13b. In this case, we observe that the *Intersecting Curve*, $d(l, n)$, can only come from the left of the *Reference-Border*, and will become the new left-border of the *Current Region* after the intersection point. Also, the induced curve $d(n, m)$ becomes the new right-border of the *Previous Region*. A new region appears between the *Current* and the *Previous region*.

B If no, i.e. if $|n|$ is already in the I -set and the index-set of the *Current Region* does not change over the *Intersecting curve*, $d(l, n)$, near the intersection point. In this case, the *Intersecting curve* can only come from the right of the *Reference-Border* (moving towards the intersection point). This is justified in Figure 15 and is based on the fact the the algorithm always selects as the *Reference Border* the left-most border among all borders intersecting at the intersection point of interest. Two sub-cases arise, depending whether the *Induced Curve* is the right-border of the *Current Region* or not.

B.1 If yes, we are in sub-case *B.1*, depicted in Figure 14a. The *Current Region* is terminated, since its two borders intersect. The *Previous* and *Next Regions* meet at their new common border, $d(l, n)$.

B.2 If no, we are in sub-case *B.2*, depicted in Figure 14b. The *Induced Curve*

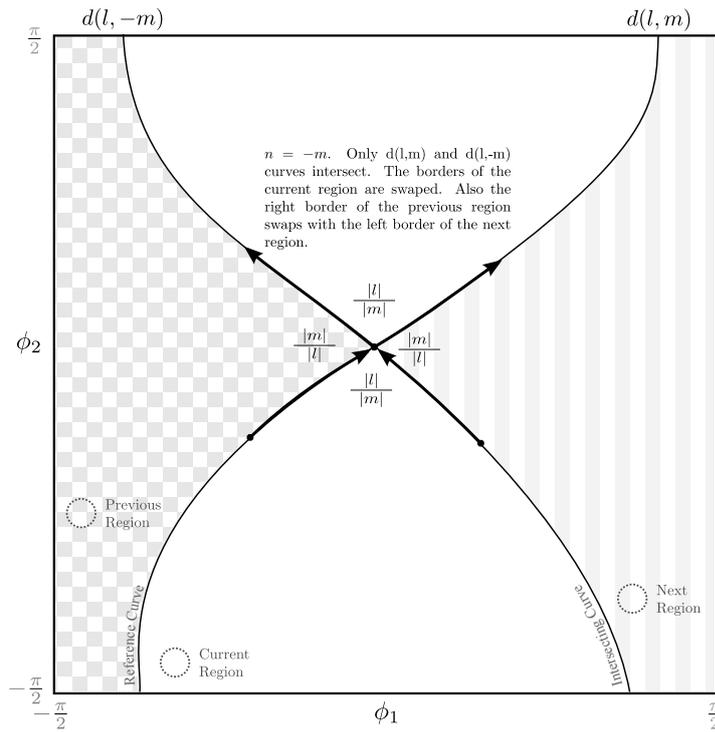
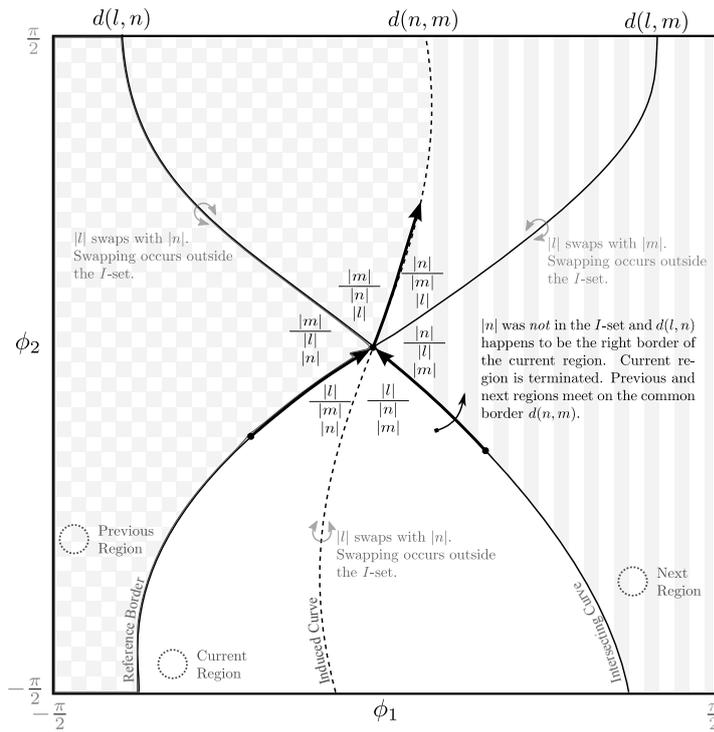
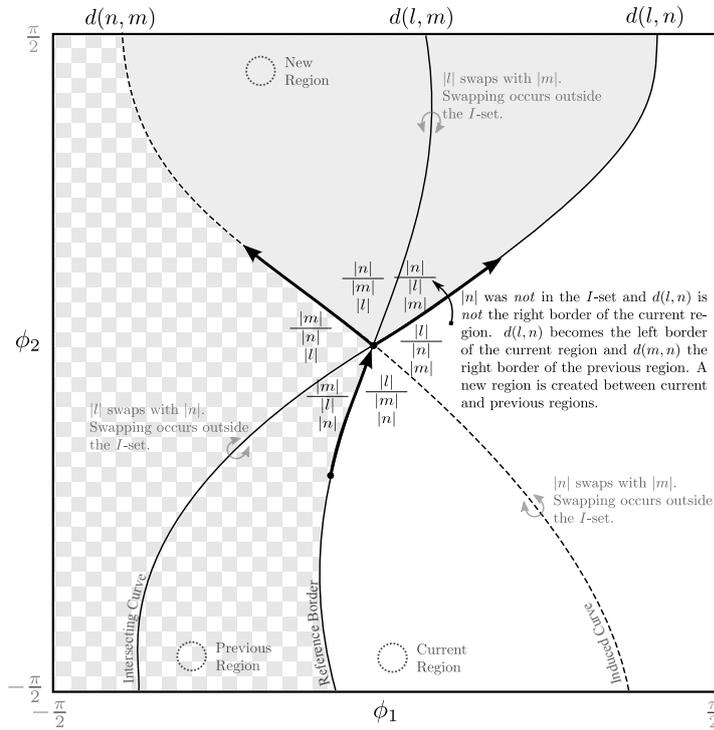


Figure 12: Rank-3: Serial, At intersection point - Special Case

can only be on the left of the *Reference-Border* and it cannot, of course, be the right border of the *Current Region*. After the intersection point, it becomes the new left border of the *Current Region* and $d(l, n)$ becomes the new right border of the *Previous Region*. A new cell appears between the modified *Previous* and *Current Regions*.

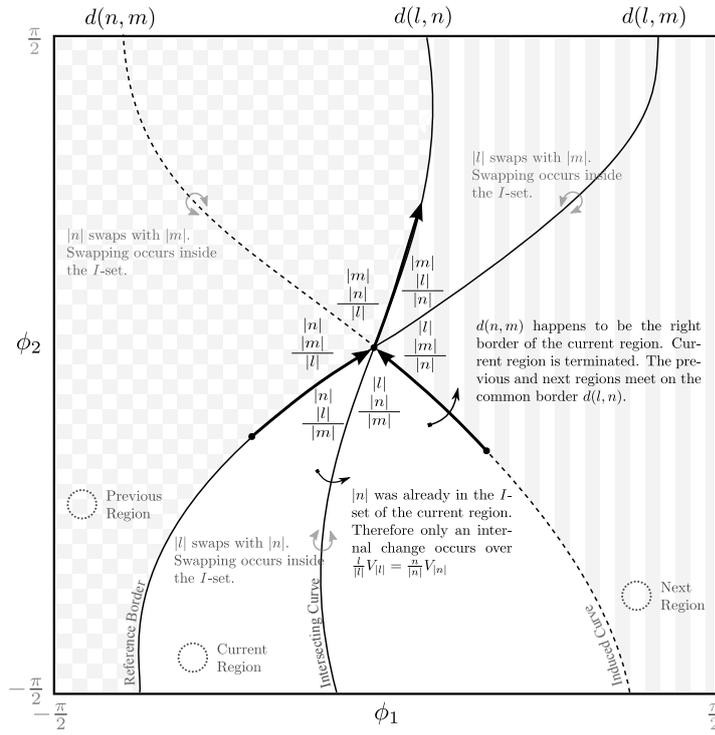


(a) Sub-Case A.1

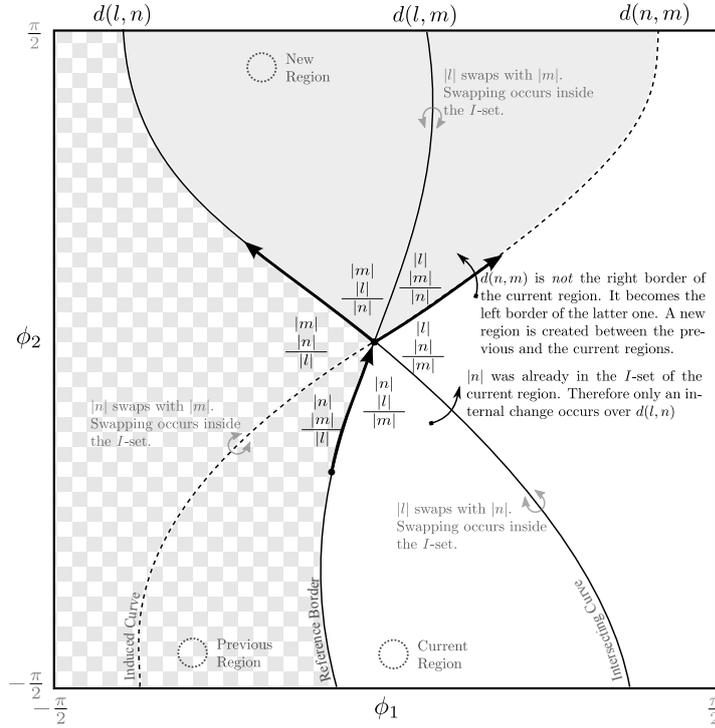


(b) Sub-Case A.2

Figure 13: Rank-3: Serial, At intersection point - Case A



(a) Sub-Case B.1



(b) Sub-Case B.2

Figure 14: Rank-3: Serial, At intersection point - Case B

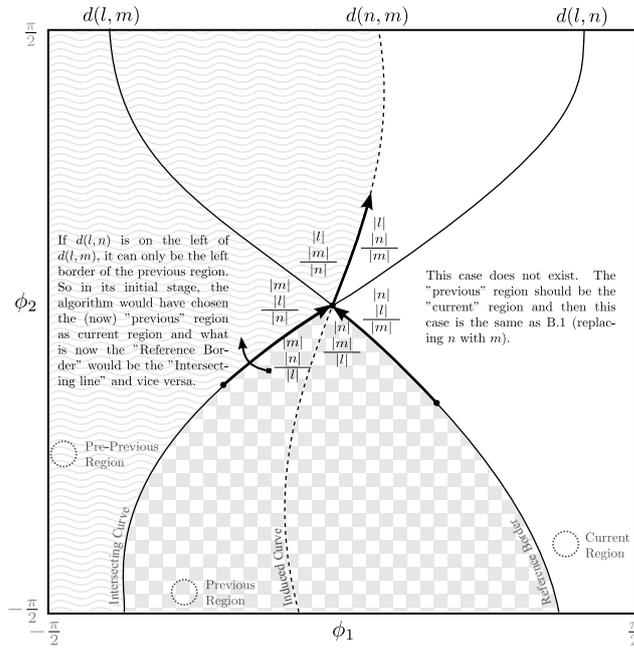


Figure 15: Rank-3: Serial, At intersection point - Non-Existent Case.

■ ————— ■ Illustration Example 5 ————— ■

In Figure 16 we present an execution instance of the rank-3 serial algorithm on random matrix $V_{4 \times 3}$, with $K = 2$. Execution starts by running a rank-2 serial algorithm on matrix $[V_{:,1} - V_{:,2}]$ which determines the beginning of the regions, at the bottom of the Φ^2 plane (red markers). We proceed by scanning the Φ^2 plane towards $\phi_2 = \frac{\pi}{2}$, keeping track of the evolution of all regions. Dashed horizontal lines, mark the visited ϕ_2 values, i.e. the ϕ_2 values at which an intersection point for one (or more) of the borders was examined.

Execution is now at $\phi_{2,cur}$. There are four active regions (the region on the right, attached to $\phi_1 = \frac{\pi}{2}$ is the region on the left, attached to $\phi_1 = -\frac{\pi}{2}$). We examine the intersections of the left-border of each region, and choose the next ϕ_2 value, $\phi_{2,next}$, based on the first point at which one or more of the regions are modified. We observe that two border-curves, the left-borders of regions A and B , both share the nearest intersection point. According to the algorithm, the left-most of the intersecting borders, the left border of region A is selected as the *Reference Border* and region A is, thus, the *Current Region*. The intersection point will be examined according to the cases developed in the previous paragraph and the necessary

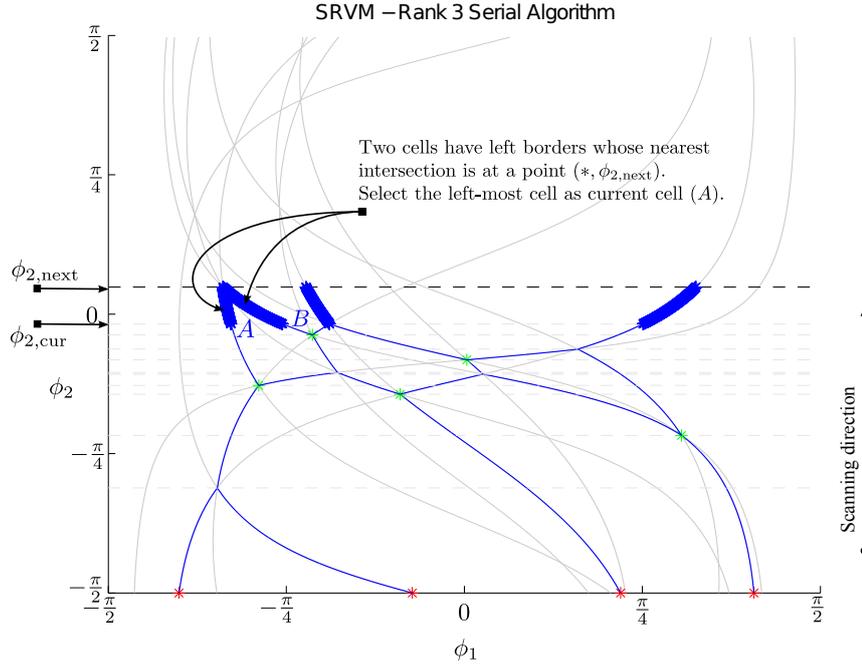


Figure 16: Rank-3: Serial Algorithm Execution Instance.

modifications will then be applied to the active region list.

VI.3.C On the complexity of the two implementations

The parallel and serial version of the algorithm have been thoroughly compared in the rank-2 case. The same observations apply in the rank-3 case.

In Figure 17 we plot the complexity curves for the two versions of the rank-3 algorithm versus the size N of the input for $K = \lceil \frac{N}{2} \rceil$. In the case of the parallel, complexity is equal to $\binom{N^2-N}{2} \times \mathcal{O}(N)$. In the case of serial version, the complexity at each examined point is again $\mathcal{O}(N)$, but the number of points examined is not a deterministic function of N . Therefore, we plot an average complexity over many experimental executions. The complexity of the parallel algorithm does not depend on K , however, in the case of the

serial algorithm we have observed that the smaller the value of K , the fewer the points examined and therefore, the lower the complexity is.

In Figure 18 we plot (a) the number of points determined by all pair-wise intersections of curves, (b) the number of points examined by the parallel version, (c) the average number of points examined by the serial version, (d) the average number of regions in the Φ^2 plane and (e) the number of distinct candidate index-sets (size of $\mathcal{I}(\mathbf{V}_{N \times 3})$), all versus the input size N for $K = \lceil \frac{N}{2} \rceil$. From curves (a) and (b) we see that the number of intersection points examined by the rank-3 Parallel algorithm is significantly smaller than the number of points we would examine by taking all pair combinations of curves ($\binom{N}{D} 2^{D-1} \ll \binom{N^2-N}{D-1}$, $D = 3$). Curves (b) and (c) are associated with those of Figure 17 (scaled by N). The relationship of the complexity of the two versions is the same as that in the rank-2 case. Finally, curves (d) and (e) allow us to see that the number of regions and distinct I -sets is much smaller than the number of points examined by the two algorithms. We note that the number of regions and distinct candidate index-sets is not fixed N , D and K . In addition, in the average sense, it seems to depend on K : the smaller the value of K , the fewer the regions and distinct candidates. Finally, we once again underline that, although the complexity of the algorithm for the construction of $\mathcal{I}(\mathbf{V}_{N \times 3})$ is mainly determined by the number of points examined, the cardinality of the $\mathcal{I}(\mathbf{V}_{N \times 3})$ is also significant, since the final solution of the maximization problem will be numerically calculated with exhaustive search among the index-sets contained in $\mathcal{I}(\mathbf{V}_{N \times 3})$.

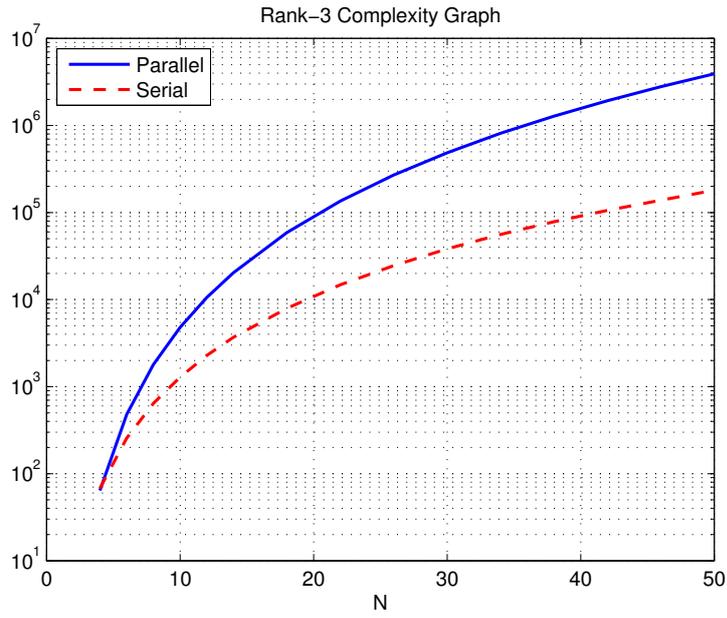


Figure 17: Rank-3: Algorithm Complexity

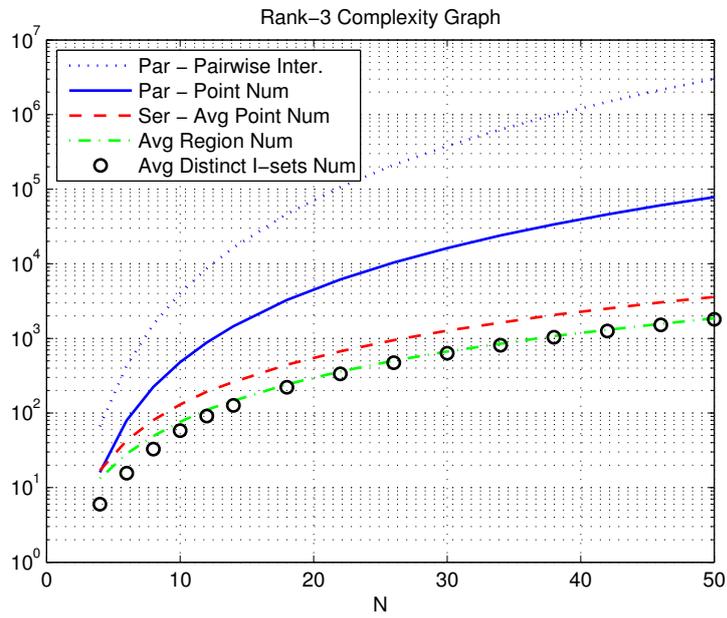


Figure 18: Rank-3: Points examined, Regions, Distinct I -sets.

VII Conclusions

We considered the problem of identifying the index-set of the nonzero elements of the vector that maximizes a rank-deficient quadratic form. We introduced auxiliary spherical coordinates and proved that there exists a set of candidate index-sets whose size is polynomially bounded, in terms of rank, and contains the optimal index-set, i.e. the index-set of the nonzero elements of the optimal solution of the quadratic form. Finally, we developed (parallel and serial implementations of) an algorithm that computes the collection of the candidate index-sets in polynomial time, for the cases where the rank of the quadratic form equals 2 or 3.

Appendix

A Proof of Proposition 1

Part a

Consider $P_{D-1} = \{\{p_{1,1}, p_{1,2}\}, \{p_{2,1}, p_{2,2}\}, \dots, \{p_{D-1,1}, p_{D-1,2}\}\}$ and the $D-1$ hypersurfaces $S(\frac{p_{1,1}}{|p_{1,1}|}V_{|p_{1,1}|,:}; \frac{p_{1,2}}{|p_{1,2}|}V_{|p_{1,2}|,:}), \dots, S(\frac{p_{D-1,1}}{|p_{D-1,1}|}V_{|p_{D-1,1}|,:}; \frac{p_{D-1,2}}{|p_{D-1,2}|}V_{|p_{D-1,2}|,:})$ that correspond to $D-1$ pairs of rows of $\mathbf{V}_{N \times D}$. Since each hypersurface $S(\frac{p_{*,1}}{|p_{*,1}|}V_{|p_{*,1}|,:}; \frac{p_{*,2}}{|p_{*,2}|}V_{|p_{*,2}|,:})$ is described by the equation $(\frac{p_{*,1}}{|p_{*,1}|}V_{|p_{*,1}|,:} - \frac{p_{*,2}}{|p_{*,2}|}V_{|p_{*,2}|,:})\mathbf{c}(\phi_{1:D-1}) = 0$, $\{p_{*,1}, p_{*,2}\} \in P_{D-1}$, their intersection will satisfy the system of equations

$$\underbrace{\begin{cases} (\frac{p_{1,1}}{|p_{1,1}|}V_{|p_{1,1}|,:} - \frac{p_{1,2}}{|p_{1,2}|}V_{|p_{1,2}|,:})\mathbf{c}(\phi_{1:D-1}) & = 0 \\ (\frac{p_{2,1}}{|p_{2,1}|}V_{|p_{2,1}|,:} - \frac{p_{2,2}}{|p_{2,2}|}V_{|p_{2,2}|,:})\mathbf{c}(\phi_{1:D-1}) & = 0 \\ \vdots & \\ (\frac{p_{D-1,1}}{|p_{D-1,1}|}V_{|p_{D-1,1}|,:} - \frac{p_{D-1,2}}{|p_{D-1,2}|}V_{|p_{D-1,2}|,:})\mathbf{c}(\phi_{1:D-1}) & = 0 \end{cases}}_{\mathbf{V}_P \mathbf{c}(\phi_{1:D-1}) = \mathbf{0}_{(D-1) \times 1}} \quad (47)$$

The above system is rewritten as $\mathbf{V}_P \mathbf{c}(\phi_{1:D-1}) = \mathbf{0}_{(D-1) \times 1}$. Therefore, the solution $\phi_{1:D-1}$ is such that $\mathbf{c}(\phi_{1:D-1})$ belongs to the null space of \mathbf{V}_P which is denoted by $\mathcal{N}(\mathbf{V}_P)$ and has dimension greater than or equal to one, since $\mathcal{R}(\mathbf{V}_{P,1:D}) \leq D-1$. Let $\mathbf{V}_{P,1:D} = \tilde{\mathbf{U}}_{(D-1) \times (D-1)} \mathbf{\Lambda}_{(D-1) \times D} \mathbf{U}_{D \times D}^T$ be the singular value decomposition of $\mathbf{V}_{P,1:D}$, where $\tilde{\mathbf{U}}$ and \mathbf{U}^T are orthogonal matrices, $\mathbf{\Lambda} = [\text{diag}(\lambda_1, \lambda_2, \dots, \lambda_{D-1}) \ \mathbf{0}_{(D-1) \times 1}]$ and w.l.o.g. $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{D-1} \geq 0$.

We consider two cases.

1. If $\lambda_{D-1} > 0$, then $\mathcal{N}(\mathbf{V}_{P,1:D}) = \{\alpha \mathbf{U}_{1:D,D} : \alpha \in \mathbb{R}\}$, which implies that $\mathbf{c}(\phi_{1:D-1}) = \frac{\mathbf{U}_{1:D,D}}{\|\mathbf{U}_{1:D,D}\|}$. Since we require $\phi_1 \in (-\frac{\pi}{2}, \frac{\pi}{2}]$, only one solution $\pm \frac{\mathbf{U}_{1:D,D}}{\|\mathbf{U}_{1:D,D}\|}$ is valid and the spherical coordinate vector we look for is uniquely determined by the spherical coordinates of $\frac{\mathbf{U}_{1:D,D}}{\|\mathbf{U}_{1:D,D}\|}$ or $-\frac{\mathbf{U}_{1:D,D}}{\|\mathbf{U}_{1:D,D}\|}$.
2. if $\lambda_{D-1} = 0$, then $\dim \mathcal{N}(\mathbf{V}_{P,1:D}) \geq 2$ which implies that there are uncountably many solutions for $\mathbf{c}(\phi_{1:D-1})$ that satisfy the requirement $\phi_1 \in (-\frac{\pi}{2}, \frac{\pi}{2}]$.

Part b

(i)

$$\begin{aligned}
& |\mathbf{V}_{N \times D} \mathbf{c}([\boldsymbol{\phi}_{1:D-2}, \frac{\pi}{2}])| \\
&= |\mathbf{V}_{N \times D} \begin{pmatrix} \sin \phi_1 \\ \cos \phi_1 \sin \phi_2 \\ \vdots \\ \cos \phi_1 \cdots \cos \phi_{D-2} \sin \frac{\pi}{2} \\ \cos \phi_1 \cdots \cos \phi_{D-2} \cos \frac{\pi}{2} \end{pmatrix}| = |\mathbf{V}_{N \times D} \begin{pmatrix} \sin \phi_1 \\ \cos \phi_1 \sin \phi_2 \\ \vdots \\ \cos \phi_1 \cdots \cos \phi_{D-2} \end{pmatrix}| \\
&= |\mathbf{V}_{N \times (D-1)} \mathbf{c}(\boldsymbol{\phi}_{1:D-2})|
\end{aligned}$$

(ii)

$$\begin{aligned}
& |\mathbf{V}_{N \times D} \mathbf{c}([\boldsymbol{\phi}_{1:D-2}, -\frac{\pi}{2}])| \\
&= |\mathbf{V}_{N \times D} \begin{pmatrix} \sin \phi_1 \\ \cos \phi_1 \sin \phi_2 \\ \vdots \\ \cos \phi_1 \cdots \cos \phi_{D-2} \sin -\frac{\pi}{2} \\ \cos \phi_1 \cdots \cos \phi_{D-2} \cos -\frac{\pi}{2} \end{pmatrix}| = |\mathbf{V}_{N \times D} \begin{pmatrix} -\sin -\phi_1 \\ -\cos -\phi_1 \sin -\phi_2 \\ \vdots \\ -\cos -\phi_1 \cdots \cos -\phi_{D-2} \sin \frac{\pi}{2} \\ -\cos -\phi_1 \cdots \cos -\phi_{D-2} \cos \frac{\pi}{2} \end{pmatrix}| \\
&= |-\mathbf{V}_{N \times D} \mathbf{c}([-\boldsymbol{\phi}_{1:D-2}, \frac{\pi}{2}])| = |\mathbf{V}_{N \times D} \mathbf{c}([-\boldsymbol{\phi}_{1:D-2}, \frac{\pi}{2}])|
\end{aligned}$$

(iii)

$$\begin{aligned} & |\mathbf{V}_{N \times D} \mathbf{c}([\phi_{1:D-3}, \frac{\pi}{2}, \phi_{D-1}])| \\ &= |\mathbf{V}_{N \times D} \begin{pmatrix} \sin \phi_1 \\ \cos \phi_1 \sin \phi_2 \\ \vdots \\ \cos \phi_1 \cdots \cos \phi_{D-3} \sin \frac{\pi}{2} \\ \cos \phi_1 \cdots \cos \phi_{D-3} \cos \frac{\pi}{2} \sin \phi_{D-1} \\ \cos \phi_1 \cdots \cos \phi_{D-3} \cos \frac{\pi}{2} \cos \phi_{D-1} \end{pmatrix}| = |\mathbf{V}_{N \times D} \begin{pmatrix} \sin \phi_1 \\ \cos \phi_1 \sin \phi_2 \\ \vdots \\ \cos \phi_1 \cdots \cos \phi_{D-3} \end{pmatrix}| \\ &= |\mathbf{V}_{N \times (D-2)} \mathbf{c}(\phi_{1:D-3})| \end{aligned}$$

(iv)

$$\begin{aligned} & |\mathbf{V}_{N \times D} \mathbf{c}([\phi_{1:D-3}, -\frac{\pi}{2}, \phi_{D-1}])| \\ &= |\mathbf{V}_{N \times D} \begin{pmatrix} \sin \phi_1 \\ \cos \phi_1 \sin \phi_2 \\ \vdots \\ \cos \phi_1 \cdots \cos \phi_{D-3} \sin -\frac{\pi}{2} \\ \cos \phi_1 \cdots \cos \phi_{D-3} \cos -\frac{\pi}{2} \sin \phi_{D-1} \\ \cos \phi_1 \cdots \cos \phi_{D-3} \cos -\frac{\pi}{2} \cos \phi_{D-1} \end{pmatrix}| = |\mathbf{V}_{N \times D} \begin{pmatrix} -\sin -\phi_1 \\ -\cos -\phi_1 \sin -\phi_2 \\ \vdots \\ -\cos -\phi_1 \cdots \cos -\phi_{D-3} \end{pmatrix}| \\ &= |\mathbf{V}_{N \times D} \begin{pmatrix} -\sin -\phi_1 \\ -\cos -\phi_1 \sin -\phi_2 \\ \vdots \\ -\cos -\phi_1 \cdots \cos -\phi_{D-3} \sin \frac{\pi}{2} \\ -\cos -\phi_1 \cdots \cos -\phi_{D-3} \cos \frac{\pi}{2} \sin \phi'_{D-1} \\ -\cos -\phi_1 \cdots \cos -\phi_{D-3} \cos \frac{\pi}{2} \cos \phi'_{D-1} \end{pmatrix}| = |-\mathbf{V}_{N \times D} \mathbf{c}([-\phi_{1:D-3}, \frac{\pi}{2}, \phi'_{D-1}])| \\ &= |\mathbf{V}_{N \times D} \mathbf{c}([-\phi_{1:D-3}, \frac{\pi}{2}, \phi'_{D-1}])| \end{aligned}$$

(v)

$$\begin{aligned} & |\mathbf{V}_{N \times D} \mathbf{c}([\phi_{1:D-3}, \pm \frac{\pi}{2}, \phi_{D-1}])| \\ &= |\mathbf{V}_{N \times D} \begin{pmatrix} \sin \phi_1 \\ \cos \phi_1 \sin \phi_2 \\ \vdots \\ \cos \phi_1 \cdots \cos \phi_{D-3} \sin \pm \frac{\pi}{2} \\ \cos \phi_1 \cdots \cos \phi_{D-3} \cos \pm \frac{\pi}{2} \sin \phi_{D-1} \\ \cos \phi_1 \cdots \cos \phi_{D-3} \cos \pm \frac{\pi}{2} \cos \phi_{D-1} \end{pmatrix}| = |\mathbf{V}_{N \times (D-2)} \begin{pmatrix} \sin \phi_1 \\ \cos \phi_1 \sin \phi_2 \\ \vdots \\ \pm \cos \phi_1 \cdots \cos \phi_{D-3} \end{pmatrix}| \\ &= |\mathbf{V}_{N \times D} \begin{pmatrix} \sin \phi_1 \\ \cos \phi_1 \sin \phi_2 \\ \vdots \\ \cos \phi_1 \cdots \cos \phi_{D-3} \sin \pm \frac{\pi}{2} \\ \cos \phi_1 \cdots \cos \phi_{D-3} \cos \pm \frac{\pi}{2} \sin \phi'_{D-1} \\ \cos \phi_1 \cdots \cos \phi_{D-3} \cos \pm \frac{\pi}{2} \cos \phi'_{D-1} \end{pmatrix}| = |\mathbf{V}_{N \times D} \mathbf{c}([\phi_{1:D-3}, \pm \frac{\pi}{2}, \phi'_{D-1}])| \end{aligned}$$

Bibliography

- [1] G. N. Karystinos and D. A. Pados, “*Rank-2-optimal adaptive design of binary spreading codes,*” IEEE Trans. Inform. Theory, vol. 53, pp. 3075-3080, Sept. 2007.
- [2] G. N. Karystinos and A. P. Liavas, “*Efficient computation of the binary vector that maximizes a rank-3 quadratic form,*” 2006 Allerton Conf. Commun., Control, and Computing, Allerton House, Monticello, IL, Sept. 2006, pp. 1286-1291.
- [3] G. N. Karystinos and A. P. Liavas, “*Efficient computation of the binary vector that Maximizes a Rank-deficient Quadratic Form,*” IEEE Trans. Inform. Theory, vol. 56, pp. 3581-3593, July 2010.
- [4] T.H Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, “*Introduction to Algorithms,*” 2nd ed. MIT Press 2001, Part II, Chapter 9.
- [5] B. Moghaddam, Y. Weiss and S. Avidan, “*Spectral Bounds for Sparse PCA: Exact and Greedy Algorithms,*” Advances in Neural Information Processing Systems 18 (NIPS), MIT Press, 2006.
- [6] A. D’Aspremont, L. El Ghaoui, M. I. Jordan and G. R. G. Lanckriet, “*A direct formulation for sparse PCA using semidefinite programming,*”, SIAM Review, vol. 49, no. 3, pp. 434–448, 2007.
- [7] B. K. Sriperumbudur, D. A. Torres and G. R. G. Lanckriet, “*Sparse Eigen Methods by D.C. Programming*”, Proceedings of the 24 th International Conference on Machine Learning, Corvallis, OR, 2007.
- [8] T. Jolliffe, N.T. Trendafilov and M. Uddin, “*A modified principal component technique based on the LASSO,*”, Journal of Computational and Graphical Statistics, vol 12, pp. 531–547, 2003.