

# REVEALING BAND AND CIRCUMPLEX PATTERNS IN REORDERABLE MATRICES USING POLAR SORT AND FAST MULTIDIMENSIONAL PROJECTIONS

Celmar Guimarães da Silva  
*School of Technology, University of Campinas. Limeira, São Paulo, Brazil*

## ABSTRACT

Analysts may use matrix-based visualizations (such as heatmaps) to reveal patterns of a dataset with the help of reordering algorithms that permute matrix rows and columns properly. One of these algorithms is Polar Sort, a pattern-focused reordering method that uses a multidimensional projection technique – Classical MDS – to reveal Band and Circumplex patterns in reorderable matrices. Despite its good reordering results regarding the mentioned patterns, Polar sort is not scalable due to Classical MDS' asymptotic time complexity ( $O(n^3)$ ) for an input matrix with size  $n \times n$ . In this paper, we propose a new version of this algorithm, in which we replace Classical MDS with FastMap, a method with asymptotic time complexity  $O(n)$ . The new algorithm (Polar Sort with Fastmap, or PSF for short) permutes rows and columns according to their bidimensional projections and uses a barycenter-based ordering identical to Polar Sort's approach. The results of an experiment indicate that PSF maintained the output quality of Polar Sort regarding minimal span loss function, Moore stress, and circular correlation when reordering synthetic matrices. Besides, PSF's asymptotic time complexity is  $O(n \log n)$ . This complexity is coherent with our experiment results, which point out that PSF had lower execution time than other compared methods. We also show some examples in which real-world matrices reordered by PSF revealed patterns similar to Band and Circumplex.

## KEYWORDS

Reorderable Matrices, Patterns, Multidimensional Projection

## 1. INTRODUCTION

A common concern of many visualization researchers is about where to place visual items on the screen. E.g. graph-layout algorithms put nodes in positions that reveal characteristics such as intrinsic layers and hierarchies, cycles or tree structures; treemap visualizations have many

distinct ways to subdivide regions and dispose child nodes of each parent node. In this paper, our concern is about the ordering of rows and columns of a matrix visualization (a.k.a. tabular visualization), such as a heatmap or a Table Lens.

Matrix visualizations are based on the concept of reorderable matrix, i.e. a matrix whose rows and columns can be reordered without information loss. Since it was proposed by Bertin (2010) in 1967 in his first edition of the book “Semiology of Graphics”, many researchers have been proposing innovative reordering algorithms that try to reveal patterns that are easy to be visually perceived.

A relevant set of matrix reordering algorithms is called *pattern-focused algorithms* (Silva, 2020). They receive as input a matrix that hides a known pattern due to inappropriate permutations, and reorder the matrix to make that pattern evident. One of these algorithms – Polar Sort – is based on multidimensional projections of rows (and columns) in 2 dimensions (Silva, 2019), and such projections are used as a source for the new row (and column) order. Nevertheless, Polar Sort uses Classical MDS (Torgerson, 1952), a projection method with asymptotic time complexity  $O(n^3)$ , which makes it harder to scale.

To enhance Polar Sort, we searched for multidimensional projection algorithms that could replace Classical MDS, aiming to reduce Polar Sort’s time complexity and to maintain or improve the method’s output quality. In this paper, we found that FastMap is an ideal candidate for this replacement. Therefore, our hypothesis was that replacing Classical MDS by FastMap inside Polar Sort would improve Polar Sort’s execution time and would not deteriorate the quality of its output. In our experiments, the new *Polar Sort with FastMap (PSF)* was able to reveal Band and Circumplex patterns in matrices with quality similar to that of the original Polar Sort, but with improved execution time as expected.

The remainder of this paper is organized as follows. A theoretical background about reorderable matrices is presented at Section 2. Section 3 shows a brief bibliographic review about reordering algorithms; it highlights pattern-focused reordering algorithms, and mainly Polar Sort. It also discusses multidimensional projects in general, and FastMap in details. Section 4 presents PSF and a test methodology of this new proposal. Section 5 discusses the results of our evaluation procedures. Last but not least, Section 6 ends this paper and presents future work.

## 2. THEORETICAL BACKGROUND

This section presents some core concepts about *matrix reordering*. A *reorderable matrix* is a matrix whose columns (and rows) can be permuted without information loss. The state-of-the-art literature for reorderable matrices points out nine matrix patterns. Half of them are based on adjacency matrices of graphs (Behrisch et al., 2016): Bands, Block-diagonal, Off-diagonal blocks, and Line/star. They represent graph organizations and situations such as meshes, paths, clusters, bipartite graphs, and hubs. The remainder patterns, called *canonical data patterns* (Wilkinson, 2005), are based on multidimensional datasets presented as heatmaps: Equi-correlation (Equi, for short), Simplex, Block, Band and Circumplex.

In this paper, we have a special interest in Band and Circumplex patterns. The *Band pattern* is defined in such a way that its higher values are closer to the main diagonal, and there is a monotonical decreasing of values from this diagonal to the top-right and bottom-left corners of the matrix. More formally, Wilkinson (2005) defines a Band pattern as a matrix  $X_{n \times p}$  whose cell

REVEALING BAND AND CIRCUMPLEX PATTERNS IN REORDERABLE MATRICES USING  
POLAR SORT AND FAST MULTIDIMENSIONAL PROJECTIONS

values are  $x_{i,j} = e^{-t^2} + u_{i,j}$ , where  $t = (j/p - i/n) / b$ ,  $1 \leq i \leq n$ , and  $1 \leq j \leq p$ . The parameter  $b$  defines not only the bandwidth, but also the sharpness of the band. When  $b$  is close to 0.2, the pattern presents a thin, well-defined band. On the other hand, when  $b$  has values close to 1, the pattern shows a large band whose boundaries are smoothed. Figure 1(a-e) exemplifies these variations. In these matrices, the correlations among near columns (or rows) are high and the correlations among distant columns (or rows) are low; these correlations can be positive or negative. It is worth noting that  $u_{i,j}$  is a random error that we implemented as a salt-and-pepper error. Band patterns appear in some similarity matrices, i.e., matrices whose cells indicate a similarity measure between all pairs of elements of a set.

A *Circumplex pattern* is similar to a Band one, but considering its matrix as a torus with an endless band. Its formal definition is almost the same from Band pattern, but using  $t = \cos(\Pi (j/p - i/n))$ . As an example of this pattern, a Circumplex correlation matrix is a symmetric matrix with correlation of pairs of variables, and the set of correlations define a circular correlation structure. Figure 1(f) exemplifies this pattern.

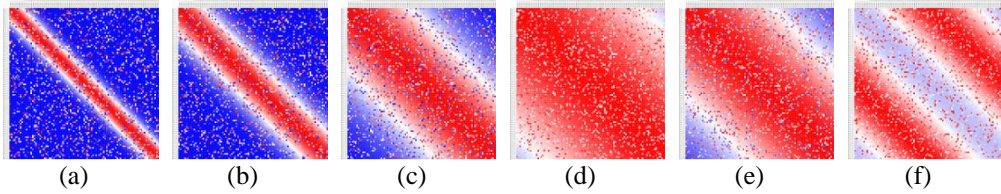


Figure 1. Patterns in matrices with size  $100 \times 100$  and noise ratio 0.2. The matrices (a) to (e) have Band patterns with the following bandwidth values: (a) 0.1; (b) 0.2; (c) 0.5; (d) 0.7; (e) 1.0. The matrix (f) presents a Circumplex pattern. Red, white and blue stands for high, average and low values, respectively

Regarding matrix nomenclature, a matrix that shows a pattern  $P$  in some permutation of its rows and columns is called a *pre- $P$  matrix*. As far as we know, Wilkinson (2005) was the first to use this terminology. In our paper, we focus on pre-Band and pre-Circumplex matrices, given that Polar Sort is able to find a permutation of these matrices such that they reveal their Band and Circumplex patterns.

The current matrix reordering literature presents some *evaluation functions* to assess the quality of a reordered matrix. We highlight Moore stress, minimal span and circular correlation.

*Moore stress* (Niermann, 2005) is a measure of local differences of cells. The stress of a cell is defined as the sum of the differences between a cell and each of its 8 neighbors. The stress of a matrix is the sum of the stresses of all its cells. In other words, given a matrix  $M_{n \times m}$ , we can define Moore stress as follows:  $Moore\_stress(M) = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} cell\_stress(M, i, j)$ , where  $cell\_stress(M, i, j) = \sum_{p=\max(i-1,0)}^{\min(i+1,n-1)} \sum_{q=\max(j-1,0)}^{\min(j+1,m-1)} |M(i, j) - M(p, q)|$ .

*Minimal span* (Wu et al., 2008) is a measure of how different all pairs of neighbor rows (or columns) of a matrix are. The difference of a pair of neighbor rows (or columns) may be calculated using a distance function such as Euclidean distance. The sum of these differences is the minimal span loss function (MSLF). In other words, we can define minimal span as follows:  $Minimal\_span_{rows}(M) = \sum_{i=0}^{n-2} d_{rows}(M, i, i+1)$ , and  $Minimal\_span_{columns}(M) = \sum_{j=0}^{m-2} d_{columns}(M, j, j+1)$ . In this work,  $d_{rows}$  and  $d_{columns}$  are the Euclidean distances of rows and of columns, respectively. They are defined as follows:  $d_{rows}(M, p, q) = \sqrt{\sum_{j=0}^{m-1} (M(p, j) - M(q, j))^2}$ , and  $d_{columns}(M, p, q) = \sqrt{\sum_{i=0}^{n-1} (M(i, p) - M(i, q))^2}$ .

Correlations between the original order of the matrix rows (or columns) and their order in the reordered matrix can also be used to measure the quality of the reordering (Wilkinson, 2005). One possible correlation measure to use is Spearman correlation (a.k.a. Spearman's Rank Correlation Coefficient). It is a measure that compares two lists of ranks (Heumann et al., 2016). Consider two lists of values –  $P = [p_1, p_2, \dots, p_n]$  and  $Q = [q_1, q_2, \dots, q_n]$  – and a function  $r(x_i)$  that represents the rank of  $x_i$  among the ordered values of the list  $X$ . Spearman correlation is defined as follows:  $R(P, Q) = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n(n^2-1)}$ , where  $d_i = R(p_i) - R(q_i)$ .  $R(P, Q)$  can return values from -1 to +1. The closer  $R$  is to +1 (or -1), the stronger the positive (or negative) correlation between  $P$  and  $Q$ .

Therefore, if we consider  $P$  as the original order of rows of a given matrix  $M$  (i.e.,  $P = [0, 1, 2, \dots, n - 1]$ ) and  $Q$  as the order of the same rows in a reordered version  $M'$  of  $M$ , then Spearman correlation can be used to assess how similar the row orders of  $M'$  and  $M$  are. The same reasoning is valid for matrix columns.

However, when comparing a reordered pre-Circumplex matrix and its original circumplex pattern, it is recommended to use *circular correlation* (Rocha et al., 2017) instead of Spearman correlation to consider all shifted permutations of this matrix. Circular correlation can be defined as the highest possible absolute Spearman correlation between rows (or columns) of a matrix  $A$  and their respective ordering in all possible shifted versions of rows (or columns) of a matrix  $B$ . (Silva, 2020).

### 3. RELATED WORK

Since the creation of the concept of reorderable matrices by Bertin, a multitude of matrix reordering algorithms was developed (Behrisch et al., 2016; Liiv, 2010). Silva (2020) pointed out a particular class of this type of algorithms, called *pattern-focused matrix reordering methods*. These methods focus on revealing a set of one or more patterns in a reorderable matrix when these patterns are hidden in the input matrix due to inappropriate permutations of its rows or columns. Table 1 summarizes the current set of pattern-focused methods (to the best of our knowledge). In this table,  $n$  is the size of a  $n \times n$  matrix, and  $k$  is a configuration parameter of the Block pattern.

In this table, we can observe that three algorithms deal with Circumplex pattern (Polar Sort, Hybrid Sort, and Circumplex Sort), and two deal with Band pattern (Polar Sort and Hybrid Sort). Hybrid Sort uses a matrix classifier to classify an input matrix as a pre-Band, pre-Circumplex, pre-Equi, pre-Simplex, or pre-Block matrix. According to this classification, the method uses FVS (Silva et al., 2017) to reorder pre-Simplex or pre-Equi matrices, Polar Sort to reorder pre-Band or pre-Circumplex matrices, and Block Reordering (Santos et al., 2016) to reorder pre-Block matrices. It is worth noting that Polar Sort produced results with better quality than those produced by Circumplex Sort when reordering pre-Circumplex matrices (Silva, 2019); this fact justifies the use of Polar Sort (and not Circumplex Sort) within Hybrid Sort, despite the lower time complexity of Circumplex Sort.

REVEALING BAND AND CIRCUMPLEX PATTERNS IN REORDERABLE MATRICES USING  
POLAR SORT AND FAST MULTIDIMENSIONAL PROJECTIONS

Table 1. Pattern-focused reordering methods

Method	Expected input matrices	Patterns revealed in the output	Asymptotic time complexity
<i>Circumplex Sort</i> (Rocha et al., 2017)	Pre-Circumplex	Circumplex	$O(n^2)$
<i>FVS</i> (Silva et al., 2017)	Pre-Simplex or pre-Equi	Simplex or Equi	$O(n^2)$
<i>Block Reordering</i> (Santos et al., 2016)	Pre-Block	Block	$O(kn^2)$
<i>Polar Sort</i> (Silva, 2019)	Pre-Band or pre-Circumplex	Band or Circumplex	$O(n^3)$
<i>Hybrid Sort</i> (Silva, 2020)	Pre-Band, pre-Simplex, pre-Equi, pre-Circumplex or pre-Block	Band, Simplex, Equi, Circumplex or Block	$O(n^3)$

We can classify Polar Sort as a *dimensional reduction* algorithm – one of a set of seven classes of matrix reordering algorithms surveyed by Behrisch et al. (2016). This class of algorithms reorders rows (or columns) of a  $m \times n$  matrix according to multidimensional projections of these rows (or columns). (All the following statements about rows are also applied to columns, replacing  $m$  by  $n$  and vice-versa as needed). A row may be considered a multidimensional point in  $\mathfrak{R}^n$ . Some dimensional projection methods project these  $m$   $n$ -dimensional rows in  $\mathfrak{R}^k$ ,  $k \ll n$ . Usually  $k=1$  or  $k=2$ . When  $k=1$ , the rows are projected into a line, and their relative order may be replicated as the order of the rows in the matrix. Wilkinson (2005) used  $k=1$  with interactive MDS and SVD. When  $k=2$ , the rows are projected into a plane. In the particular case of Band and Circumplex matrices, this projection has the format of an arc or a circle, respectively, when they are projected by Classical MDS (Silva, 2019).

Polar Sort uses  $k=2$  and Classical MDS (Torgerson, 1952) to create such arcs or circles. After that, it calculates the barycenter of the arc or circle, and uses it as the center of a polar coordinate system. The rows are ordered according to the angle of their respective points in this coordinate system, starting close to the biggest gap between neighbor points.

Polar Sort’s main concern is its  $O(n^3)$  asymptotic time complexity, derived from the time complexity  $O(n^3)$  of the Classical MDS method for projecting a set of  $n$  input objects. The lack of scalability of this projection method motivated the search for a suitable and scalable multidimensional projection method to replace Classical MDS inside Polar Sort. A recent survey about multidimensional projection methods from Nonato and Aupetit (2019) pointed out the existence of 27 “core methods” and a variety of variants of them. Classical MDS has one of the worst time complexities according to this survey. The fastest dimensional projection method presented by Nonato and Aupetit is FastMap (Faloutsos & Lin, 1995). Its asymptotic time complexity is  $O(nk)$ , where  $n$  is the number of  $d$ -dimensional objects considered for the projection, and  $k$  is the dimensionality of the “target space”, i. e., the projection space. Nonato and Aupetit (2019) indicate that this time complexity is  $O(n)$ , considering  $k$  as a constant.

In a given iteration  $r$ ,  $1 \leq r \leq k$ , FastMap calculates the  $r^{\text{th}}$  coordinate of the  $n$  objects in the projection space. Consider that the input of this iteration has  $\alpha$ -dimensional objects, where  $\alpha = k + 1 - r$ . At the start of an iteration, the algorithm heuristically selects two objects (called “pivot objects”,  $O_a$  and  $O_b$ ) whose distance is maximal in the dataset, and considers the line  $p$  that intersects them. Each of the  $n$  objects is projected on  $p$ . Given an object  $O_i$  and its projection  $O_i'$ , the distance between  $O_a$  and  $O_i'$  defines the coordinate of this object in the

$r^{\text{th}}$  dimension of the projected space. All the objects are projected on a hyperplane  $H$  that is perpendicular to  $p$ . Given that  $H$  is a hyperplane inside an  $\alpha$ -dimensional space,  $H$  has  $\alpha-1$  dimensions. The next iteration of the method starts using the  $(\alpha-1)$ -dimensional objects projected on  $H$  as the input objects to be projected on a  $(k-1)$ -dimensional space. After the  $k$  iterations, all  $n$  objects have  $k$  coordinates, one for each dimension of the projection space.

Figure 2 exemplifies one iteration of FastMap for four 3D points ( $O_a$ ,  $O_b$ ,  $O_1$ , and  $O_2$ ).  $O_a$  and  $O_b$  are the pivot objects in this case.  $O_1'$  and  $O_2'$  are projections of  $O_1$  and  $O_2$  on the line  $p$ , respectively.  $P_1$  and  $P_2$  are the projections of  $O_1$  and  $O_2$  on  $H$ , respectively. In this case, the projections of  $O_b$  and  $O_a$  on  $H$  coincide at  $O_a$ .

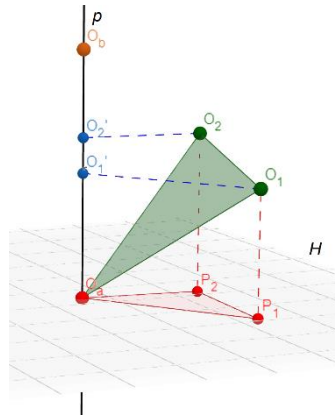


Figure 2. Example of one iteration of FastMap whose input was a set of four 3D objects ( $O_a$ ,  $O_b$ ,  $O_1$ , and  $O_2$ ). The blue dashed line segments are perpendicular to the line  $p$ . The red dashed line segments and  $p$  are perpendicular to the hyperplane  $H$  (which, in this case, is a plane). (We painted the red and green triangles only as a visual aid for readers)

## 4. METHODOLOGY

Aiming to reduce Polar Sort's asymptotic time complexity, we proposed the following methodology:

- 1) Search for a multidimensional projection method with low time complexity.
- 2) Find an implementation of the selected method.
- 3) Create a new version of the Polar Sort method, replacing Classical MDS by the method found in Step 1.
- 4) Test the ability of the new version of Polar Sort to reveal Band and Circumplex patterns in pre-Band and pre-Circumplex input matrices, respectively, and compare this method with the original Polar Sort and other related reordering methods.

According to the literature review, we chose FastMap (Faloutsos & Lin, 1995) as the fastest multidimensional projection algorithm among the ones reported by Nonato and Aupetit (2019). It has the lowest asymptotic time complexity among the surveyed algorithms:  $O(kn)$ . In our case, we want to project points (which represent rows or columns) on a plane, therefore  $k=2$ . In other words, we can consider this complexity as  $O(n)$ .

The Projection Explorer tool (Paulovich et al., 2007) provides a Java implementation of FastMap. We adopted it, and inserted Projection Explorer as a library of the Java-based tool Matrix Reordering Analyzer (MRA) (Silva et al., 2014), which was used to run our evaluation procedures.

The new version of the Polar Sort method with FastMap (**PSF** for short) was created based on the original Polar Sort, in which we replaced Classical MDS by FastMap. Therefore, the PSF method does the following steps to reorder an  $n \times n$  input matrix  $M$ :

- *Step 1.* Project the rows of  $M$  in  $\mathcal{R}^2$ , using FastMap ( $O(n)$ ).
- *Step 2.* Convert the Cartesian coordinate system into a new polar coordinate system whose center is the barycenter of the projected points ( $O(n)$ ).
- *Step 3.* Sort the points by their angular coordinates ( $O(n \log n)$ ). Find the pair of adjacent points with maximum angular distance between them ( $O(n)$ ). The ordered list of points must start with one point of this pair.
- *Step 4.* Replicate this order for the matrix rows ( $O(n)$ ).
- *Step 5.* Repeat steps 1 to 4 for columns instead of rows ( $O(n \log n)$ ).

Therefore, the asymptotic time complexity of PSF is  $O(n \log n)$ , which is a good improvement when compared to the  $O(n^3)$  from its predecessor.

We executed an evaluation experiment through MRA to discover how PSF performance can be compared to other similar algorithms when reordering pre-Band and pre-Circumplex synthetic matrices. Our experiment compared four algorithms to PSF: Classical MDS (with  $k=1$ ), SVD (with  $k=1$ ), FastMap (with  $k=1$ ), and Polar Sort; in this case,  $k$  is the dimensionality of the target space of the multidimensional projections. The first three algorithms are multidimensional projection algorithms used to reorder rows and columns of the input matrices. We defined  $k=1$  for both Classical MDS and SVD as used by Wilkinson (2005). We inserted FastMap as a direct concurrent of Classical MDS and SVD, in equal projection configurations (i.e.,  $k=1$ ). Polar Sort and PSF, by definition, use  $k=2$  for their multidimensional projection algorithms – Classical MDS and FastMap, respectively. The experiment settings are as follows:

- Patterns to be tested: Band (with bandwidth  $b=0.1, 0.2, 0.5,$  and  $1.0$ ), and Circumplex;
- Matrix sizes:  $250 \times 250, 500 \times 500, 750 \times 750,$  and  $1000 \times 1000$ ;
- Noise levels (i.e., the proportion of cells whose content is replaced by random values):  $0.01, 0.05, 0.1, 0.3,$  and  $0.5$ ;
- Functions for assessing the output matrices: minimal span loss function and Moore stress, for pre-Band matrices; circular correlation for pre-Circumplex matrices.

For each combination of pattern, matrix size and noise level, we created 50 matrices, which were randomly permuted. After permutation, they were reordered by the selected reordering algorithms and assessed by evaluation functions.

We also selected real-world examples of matrices to be reordered with PSF, aiming to illustrate its utility. The next section shows these examples and the results of our experiments.

## 5. RESULTS AND DISCUSSION

Figure 3 summarizes the main results related to our experiment about synthetic data. We selected 6 charts that represent the overall dataset of results, due to space limitations. We also

omitted results related to column reordering due to their similarity to the results of row reordering. In the figure, “Cl.MDS” stands for Classical MDS.

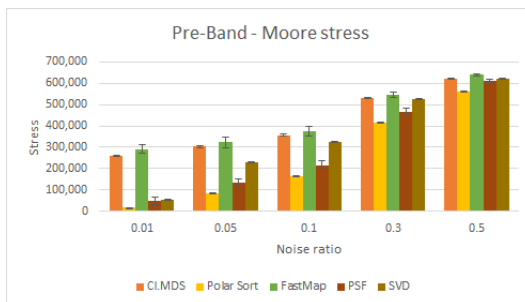
Regarding the results about pre-Band matrices (Figure 3(a-d)), we did the following observations:

- a) The Moore stress chart at Figure 3(a) shows that the average stress tends to be lower for Polar Sort and PSF results for  $b=0.1$  (we also noted the same for  $b=0.2$ ). For  $b=0.5$  (Figure 3(b)), the average stresses of each tested algorithm are very similar each other when considering the same noise level. These results are independent of the matrix size.
- b) It is possible to perceive the same pattern from (a) in the minimum span charts (whose representative charts are Figure 3(c) and Figure 3(d)). Besides, we noted in the overall results that SVD, Polar Sort and PSF have very similar minimum span measures.
- c) The execution time chart (Figure 3(e)) highlights the performance of FastMap and PSF, which was an expected result due to the low time complexity discussed before. For matrices with size  $1000 \times 1000$ , FastMap and PSF spent approximately 3 seconds, while their competitors spent hundreds of seconds. For matrices with size  $250 \times 250$  (not shown in Figure 3), the execution time of FastMap and PSF is in the order of hundredths of a second, while the time spent by their competitors is in the order of seconds.

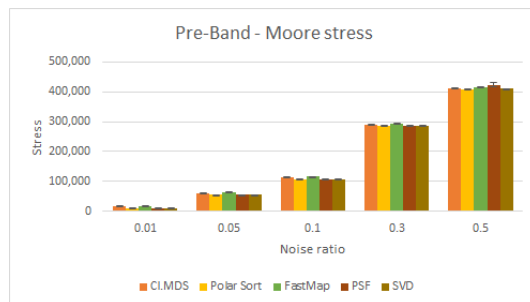
The previous comments enable us to state that PSF tends to be the best choice that combines the criteria of quality and better execution time to unveil a Band pattern in a pre-Band matrix, when compared to the other tested algorithms.

We also highlight the following results related to pre-Circumplex matrices (Figure 3(f-h)):

- a) In the circular correlation charts (represented by Figure 3(f)), Polar Sort and PSF have average correlations close to 1 (which is the desired value). The average values of the concurrent algorithms are near 0.5, and they also have a big standard deviation. This entire situation happens regardless the noise level and the matrix size.
- b) The execution time charts for pre-Circumplex matrices (Figure 3(g,h)), similarly to those from pre-Band matrices, indicate that FastMap and PSF were the fastest algorithms among the tested ones. As an example, their execution times were in the order of seconds or hundredths of seconds, respectively, for matrices with size  $1000 \times 1000$  (Figure 3(g)) and  $250 \times 250$  (Figure 3(h)). These times are two orders of magnitude lower than the times of their competitors.



(a)  $b=0.1$ , size  $1000 \times 1000$



(b)  $b=0.5$ , size  $1000 \times 1000$



REVEALING BAND AND CIRCUMPLEX PATTERNS IN REORDERABLE MATRICES USING POLAR SORT AND FAST MULTIDIMENSIONAL PROJECTIONS



Figure 3. Experiment results related to pre-Band (a-e) and pre-Circumplex (f-h) matrices

We concluded from the experiment results that the best choice to reorder pre-Circumplex matrices, among the tested algorithms, is also PSF, given its execution time and output quality.

For the sake of completeness, we note that our experiment run into a high-performance cluster that has 160 cores Power 755 3.3 GHz, 8 cores Power 720 3.0 GHz, and 24 cores Intel X5650 2.6 GHz. However, we cannot define exactly which processor executed the jobs.

This experiment revealed that PSF can produce fast and good quality results when reordering pre-Band and pre-Circumplex matrices with the tested sizes. Due to its  $O(n \log n)$  asymptotic time complexity, PSF tends to be faster than the remainder tested algorithms even for matrices with size greater than  $1000 \times 1000$ ; however, we have no proof that the output quality will remain so good for these untested situations as it was in our tests. It is worth to consider that

PSF could replace Polar Sort inside Hybrid Sort, which will also reduce Hybrid Sort’s time complexity to  $O(n \log n)$ .

We selected three real-world datasets to exemplify the utility of PSF. The first example presents how PSF can unveil a pattern that is similar to a Circumplex pattern. Figure 4 shows a matrix of the correlation between pairs of 12 mood-related adjectives (Browne, 1992), in three distinct permutations: (a) the matrix in its original order; (b) a scrambled version of it; and (c) the results of using PSF to order the scrambled matrix. The color scheme is such that reddish and bluish cells stands for high and low correlations, respectively. Note that (a) and (c) have the same circular list of adjectives in their rows (and columns), but in reverse order and starting at a distinct adjective.

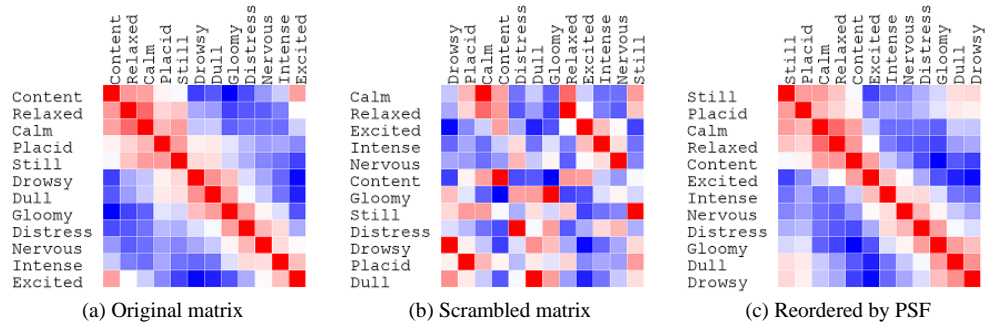


Figure 4. Using PSF to reorder a pre-Circumplex matrix.

The second example (Figure 5) is related to distances between cities. It is based on a table of travel distances between cities in Portugal (<https://portugal.com/portugal/travel-tips/distances>, accessed in April 1, 2021). We converted this table into a table of how close these cities are to each other. Each distance measure  $d(x,y)$  was converted to a proximity measure  $p(x,y)=1 - d(x,y) / \max d$ , where  $\max d$  is the greatest distance between any pair of cities in the dataset. Figure 5(a) presents a heatmap whose cities follow the same order provided in the original matrix. Figure 5(b) shows how PSF sorted this heatmap, which is close to a Band pattern. A map with these cities is available at Figure 6(a). Note that the order provided by PSF follows approximately the latitudes of the cities, from north to south, as presented at Figure 6(b).

REVEALING BAND AND CIRCUMPLEX PATTERNS IN REORDERABLE MATRICES USING POLAR SORT AND FAST MULTIDIMENSIONAL PROJECTIONS

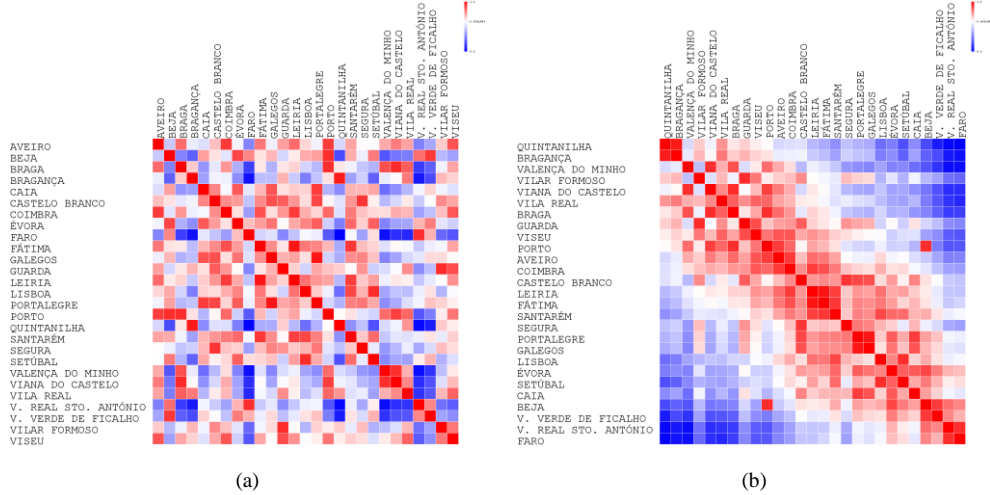


Figure 5. (a) Matrix about proximities between cities in Portugal; the red-white-blue color scale represents how close (red) or distant (blue) two cities are. (b) The same matrix, reordered by PSF

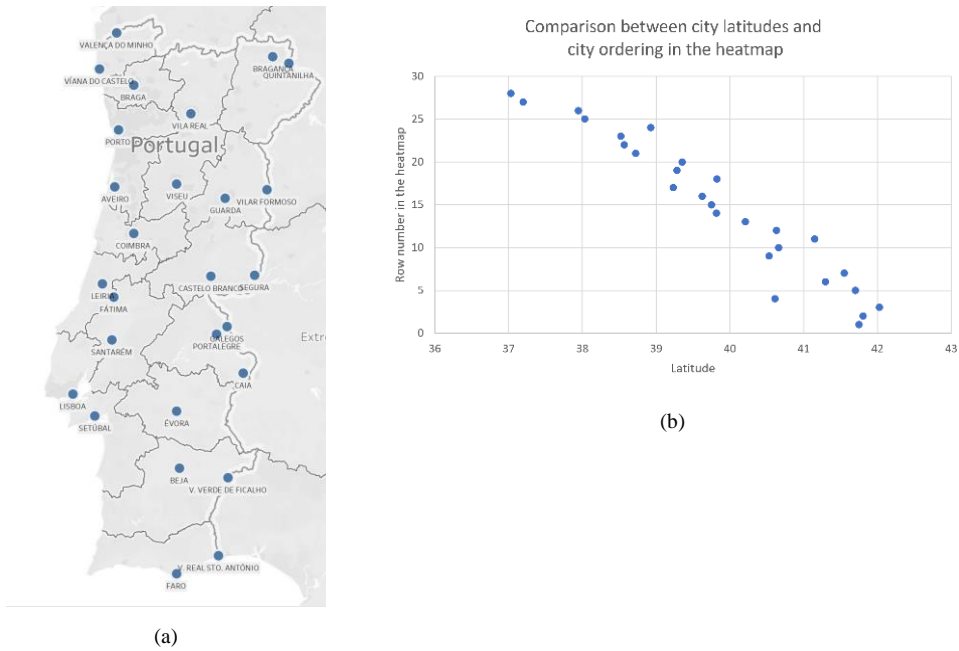
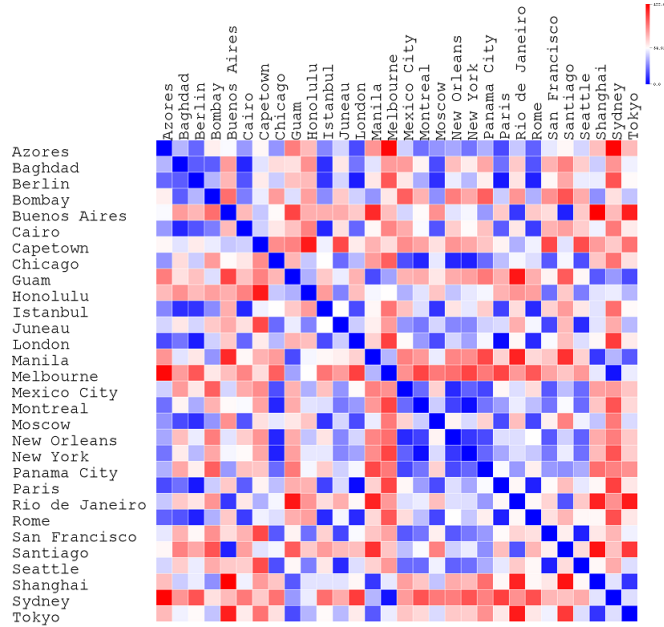
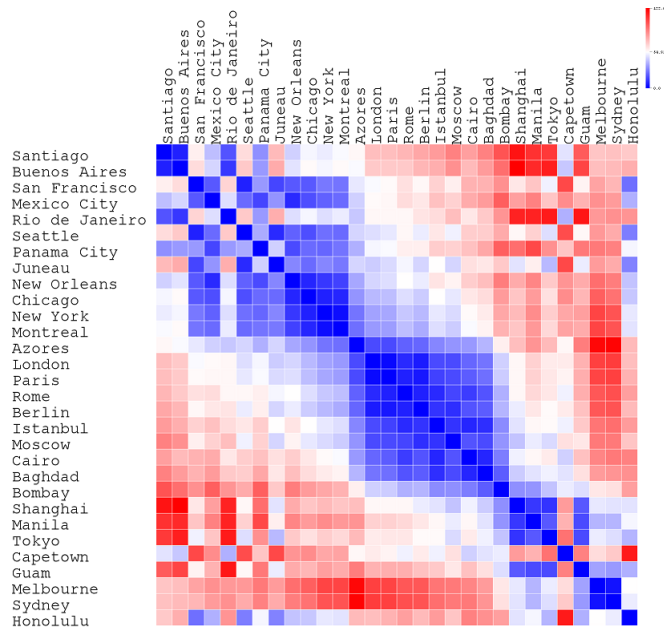


Figure 6. (a) A map of Portugal with the cities cited in Figure 5. (b) A scatterplot showing the relationship between the order of rows in the reordered matrix at Figure 5(b) and the latitude of the Portuguese cities mentioned. (Sources of latitudes: Wikipedia webpages of each city, with the support of <https://geohack.toolforge.org/>; and Citipedia.info ([https://www.citipedia.info/city/general/Portugal\\_Marvão\\_Galegos\\_id\\_2267758](https://www.citipedia.info/city/general/Portugal_Marvão_Galegos_id_2267758)). Map generated by Tableau Desktop ([www.tableau.com](http://www.tableau.com))



(a)



(b)

Figure 7. A heatmap about airline distances between 30 cities. The red-white-blue color scale represents how close (red) or distant (blue) two cities are. (a) Cities alphabetically sorted. (b) Cities sorted according to the output of PSF

The third example is about city distances around the world. It comprises 30 international cities and their pairwise airline distance. This dataset, called HA30, is available at <https://people.sc.fsu.edu/~jburkardt/datasets/cities/cities.html> (accessed in September 1<sup>st</sup>, 2021). Figure 7(a) shows a heatmap that shows this data with the cities alphabetically sorted. Figure 7(b) is a reordered version of the same heatmap, according to PSF output. The blue regions at the center of the heatmap indicates groups of cities that are close to each other, e.g. London, Paris, and Rome, or Chicago, New York, and Montreal. The cells with an intense red indicate the most distant pairs of cities, such as Rio de Janeiro and Tokyo. The dataset seems to have a kind of reversed Band pattern, i.e., a matrix whose band has the lower values of the matrix. However, the last heatmap column about Honolulu and its partial similarity with both Sydney and Santiago columns suggest that this matrix may have a Circumplex-like pattern (with noise). This pattern is very coherent with the nature of the dataset, which comprises cities around all the globe.

## 6. CONCLUSION AND FUTURE WORK

In this paper, we proposed PSF, a new version of the matrix reordering algorithm Polar Sort that uses the multidimensional projection algorithm FastMap. In the same way as Polar Sort, the goal of PSF is to reveal Band and Circumplex patterns by reordering appropriately rows and columns of pre-Band and pre-Circumplex matrices. PSF's time complexity ( $O(n \log n)$ ) surpasses the  $O(n^3)$  time complexity of its antecessor without reducing the quality reordered matrices, as we noted in the results of experiments with synthetic data. We also presented the potential of this method for revealing information in real-world data.

Future work includes studying other matrix patterns, such as the ones defined by Behrisch et al. (2016), and discovering or proposing fast reordering algorithms that are able to reveal these patterns.

## ACKNOWLEDGEMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001. We also thank the grant #2396/21 from FAEPEX/PRP/Unicamp.

## REFERENCES

- Bertin, J., 2010. *Semiology of Graphics: Diagrams, Networks, Maps*. USA: Esri Press.
- Behrisch, M. et al., 2016. Matrix reordering methods for table and network visualization. *Computer Graphics Forum*, 35(3), pp. 693–716.
- Browne, M. W. 1992. Circumplex models for correlation matrices. *Psychometrika*, 57(4), pp. 469–497.
- Faloutsos, C. and Lin, K., 1995. FastMap: A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets. *SIGMOD Conference*, pp. 163-174.
- Heumann, C., et al., 2016. *Introduction to Statistics and Data Analysis*. Springer.

- Liiv, I., 2010. Seriation and matrix reordering methods: An historical overview, *Statistical Analysis and Data Mining*, 3(2), pp. 70–91.
- Niermann, S., 2005. Optimizing the ordering of tables with evolutionary computation. *The American Statistician*, Vol. 59, No. 1, pp. 41–46.
- Nonato, L. G. and Aupetit, M., 2019. Multidimensional Projection for Visual Analytics: Linking Techniques with Distortions, Tasks, and Layout Enrichment. *IEEE Transactions on Visualization and Computer Graphics* 25, pp. 2650–2673.
- Paulovich, F. et al., 2007. The Projection Explorer: A Flexible Tool for Projection-based Multidimensional Visualization. *Proc. of SIBGRAPI 2007 - 20<sup>th</sup> Brazilian Symposium on Computer Graphics and Image Processing*, pp. 27–36.
- Rocha, M. M. N. et al., 2017. Circumplex Sort: a two-phase method for revealing circumplex data patterns in reorderable matrices. *Proc. Int. Conf. Computer Graphics, Visualization, Computer Vision and Image Processing 2017 and Big Data Analytics, Data Mining and Computational Intelligence 2017*. Lisbon, Portugal, pp. 181–188.
- Santos, A. M. et al., 2016, Block Reordering: Um algoritmo para evidenciar padrão Block em matrizes reordenáveis. *Workshop of Undergraduate Works (WUW) in the 29th Conference on Graphics, Patterns and Images (SIBGRAPI'16)*, São José dos Campos, Brazil.
- Silva, C. G. et al., 2014. PQR sort: using PQR trees for binary matrix reorganization. *Journal of the Brazilian Computer Society*, Vol. 20, Issue 1, Article 3.
- Silva, C. G., 2019. Polar Sort: combining multidimensional scaling and polar coordinates for matrix reordering. *Proc. Int. Conf. Interfaces and Human Computer Interaction 2019; Game and Entertainment Technologies 2019; and Computer Graphics, Visualization, Computer Vision and Image Processing 2019*, Porto, Portugal, pp. 239–243.
- Silva, C. G., 2020. Hybrid Sort - a pattern-focused matrix reordering approach based on classification. *Proc. Int. Conf. on Computer Graphics, Visualization, Computer Vision and Image Processing 2020, Big Data Analytics, Data Mining and Computational Intelligence 2020, and Theory and Practice in Modern Computing 2020*, online, pp. 35–43.
- Silva, C. G. et al., 2017. A fast feature vector approach for revealing simplex and equi-correlation data patterns in reorderable matrices. *Information Visualization*, 16(4), pp. 261–274.
- Torgerson, W. S., 1952. Multidimensional scaling: I. theory and method. *Psychometrika*, Vol. 17, No. 4, pp. 401–419.
- Wilkinson, L., 2005. *The Grammar of Graphics*. Springer Science & Business Media. New York, USA.
- Wu, H.-M. et al., 2008. Matrix Visualization. *Handbook of Data Visualization*. Springer, pp. 681–708.