# SYNCHRONIZATION-BASED CLUSTERING ON THE UNIT HYPERSPHERE

Zinaid KAPIĆ[1,2*], Aladin CRNKIĆ[2], Goran MAUŠA[1]

*Clustering on the unit hypersphere is a fundamental problem in various fields, with applications ranging from gene expression analysis to text and image classification. Traditional clustering methods are not always suitable for unit sphere data, as they do not account for the geometric structure of the sphere. We introduce a novel algorithm for clustering data represented as points on the unit sphere* $\mathbb{S}^{d-1}$. *Our method is based on the d-dimensional generalized Kuramoto model. The effectiveness of the introduced method is demonstrated on synthetic and real-world datasets. Results are compared with some of the traditional clustering methods, showing that our method achieves similar or better results in terms of clustering accuracy.*

**Keywords**: clustering, synchronization, unit-hypersphere

## 1. Introduction

Data that is directional in nature and can be represented as unit vectors on a d-dimensional sphere has numerous interesting applications [1-3]. Unit vectors are commonly used in wind data analysis [4]. Meteorologists frequently collect wind direction and speed at multiple sites and utilize unit vectors to represent and analyse wind behaviour, as well as to statistically process and make future predictions about wind patterns. The field that regularly uses unit vectors is robotics, where unit vectors represent the orientation of robot parts and robots. For example, the orientation of a robotic arm can be represented using a unit vector, which leads to a simplified control of the arm's movement [5]. Unit vectors are used in medicine as well, representing the orientation of limbs or joints during movement, enabling researchers to study and analyse the kinematics of human movement and the effects of various factors on it, such as injury or age [6].

In many of these applications, it is necessary to cluster directional data into distinct groups. Clustering is an unsupervised machine learning technique that divides data into different groups, ensuring that data points within a cluster are more similar to each other than to those in other clusters. Some common use cases of data clustering are image segmentation [7], customer segmentation [8], gene expression

[1] Faculty of Engineering, University of Rijeka, Croatia, zkapic@uniri.hr
[2] Faculty of Technical Engineering, University of Bihać, Bosnia and Herzegovina

analysis [9], and anomaly detection [10]. Clustering is an important tool for discovering hidden patterns and data relationships.

Unit vectors can also be subjected to clustering, which involves grouping them based on their direction or orientation. There are several methods for clustering unit vectors, including k-means clustering [11] and its variant, spherical k-means [12]. Spherical k-means is specifically designed for directional data clustering and uses cosine similarity as a distance measure instead of Euclidean distance. Spherical k-means has been used in medicine for breast cancer clustering [13] or for acute sinusitis classification [14]. The most common use of spherical k-means is text documents clustering [15]. Other popular techniques for grouping directional data are hierarchical clustering [16] and mixture models [17]. Hierarchical clustering creates a hierarchy of clusters based on the similarity of data points. On the other hand, mixture models are statistical models, that assume that data is produced by a combination of some well-known distributions [18, 19, 20]. Density-based clustering is a different category of clustering algorithms, with DBSCAN (Density-Based Spatial Clustering of Applications with Noise) as a well-known example [21]. DBSCAN clusters data points according to their density in relation to other points. Overall, there are a variety of techniques for clustering directional data, but the best technique depends on the specific task at hand.

We have used synchronization phenomenon for clustering in this paper. Synchronization is when two or more oscillating systems match their phases over time. Clustering based synchronization is a method for grouping data points into clusters based on their degree of synchronization [22, 23, 24]. Highly synchronized data points can be grouped, whereas weakly synchronized data points can be divided into separate clusters. The use of this kind of clustering can help find patterns and connections in data that might not be obvious from other approaches.

The Kuramoto model is the most common mathematical model (1) for studying synchronization phenomenon [25]. It describes the dynamics of $N$ coupled oscillators as:

$$\frac{d\theta_i}{dt} = \omega_i + \frac{K}{N}\sum_{j=1}^{N} \sin(\theta_j - \theta_i), (i = 1, \dots, N), \tag{1}$$

where the oscillators are modelled by a system of coupled differential equations that describe the evolution of each oscillators' phase $\theta_i$ over time. Their interaction is controlled by the coupling strength $K$, which controls how strong oscillators influence one another and determines the degree of synchronization in the system. Notation $\omega_i$ represents an intrinsic frequency of oscillator $i$. Variants of the Kuramoto model have been applied to the study of synchronization phenomenon in complex networks [26], data clustering [27], and rotation averaging and interpolation problems [28, 29]. A generalized version of model (1) to higher dimensions is used in this paper. Each oscillator is represented as a unit vector $Q \in$

$\mathbb{R}^d$, corresponding to a point on the $(d-1)$-dimensional unit hypersphere $\mathbb{S}^{d-1}$. The dynamics of a single oscillator in this generalized setting are governed by the differential equation:

$$\dot{Q} = WQ, \tag{2}$$

where W is an antisymmetric $d \times d$ frequency matrix of the generalized oscillator. The system of coupled oscillators in this framework is described as:

$$\dot{Q}_{j,j=1,\dots,N} = \frac{K}{N}\sum_{i=1}^{N}(Q_i - \langle Q_j, Q_i\rangle Q_j) + W_j Q_j. \tag{3}$$

Equation (3) extends the classical Kuramoto model (1) from the unit circle $\mathbb{S}^1$ to the unit hypersphere $\mathbb{S}^{d-1}$. To better understand the dynamics of the system (3), we introduce order parameter:

$$R = \frac{1}{N}\sum_{j=1}^{N} Q_j.$$

Notice that $\|R\|$ (notation $\|\cdot\|$ stands for the Euclidean norm) is a real number and has a value between 0 and 1. Case $\|R\| = 1$ corresponds to the system being a completely synchronized, $Q_i = Q_j$ for all $i, j$. On the other side, $\|R\| = 0$ represents an incoherent state. In this work, we apply a variation of this generalized model for clustering tasks, taking advantage of its synchronization properties to cluster data on the unit hypersphere.

The remainder of the paper is organized as follows. In section 2, we introduce a novel approach for clustering data on the unit hyperspheres using a vector-based Kuramoto model. We also introduce a novel algorithm based on this approach. Section 3 demonstrates the performance of our algorithm through simulations and visualizations of both real-life and simulated high-dimensional circular data. Our approach is compared to some state-of-the-art algorithms. In the conclusion, we review our research and discuss possible future outlooks.

## 2. Algorithm

To introduce a clustering method for points $P_j, j = 1, \dots, N,$ belonging to the unit sphere $\mathbb{S}^{d-1}$, we consider a system of coupled differential equations that describe the evolution of the points under mutual influence. Each point follows a dynamical equation where its movement depends on the average position of all other points in the system. Setting the frequency matrix $W = 0$, (3) transforms to:

$$\dot{Q}_{j,j=1,\dots,N} = \frac{K}{N}\sum_{i=1}^{N}\big(Q_i - \langle Q_j, Q_i\rangle Q_j\big), \tag{4}$$

where $Q_j$ represents the position of each point on the hypersphere and $K$ is the coupling parameter, which we set to $K = 1$ without loss of generality. The notation $\langle\cdot,\cdot\rangle$ stands for the standard dot-product in $\mathbb{R}^d$. Initial conditions are given as $Q_j(0) = P_j, j = 1, \dots, N$.

This model preserves the unit sphere $\mathbb{S}^{d-1}$, ensuring that the evolution of each $Q_j$ remains on the sphere for all time $t$. The dynamics of the system lead to the formation of clusters, where points with similar orientations group together over time. To extract the clusters from the evolved system, we analyse the pairwise cosine distances between the points obtained at time $t = T$. Two points $Q_i$ and $Q_j$ are considered to belong to the same cluster if their cosine distance is below a predefined threshold $\epsilon$.

Mathematically, it is expressed as

$$d(Q_i, Q_j) = 1 - \frac{Q_i \cdot Q_j}{\|Q_i\|\|Q_j\|} < \epsilon.$$

From this similarity measure, we construct an adjacency matrix $A$ where:

$$A_{ij} = \begin{cases} 1, d(Q_i, Q_j) < \epsilon, \\ 0, otherwise. \end{cases}$$

The final clusters are extracted as connected components of the graph represented by $A$. We explain our clustering method in detail as follows:

1. **Initialize**
   a. Input $N$ points $P_j \in S^{d-1}, j = 1, \dots, N$.
   b. Set parameters: time step $\delta$, clustering threshold $\epsilon$, and $\nu$.
2. **Solve the dynamical system**
   a. Integrate the system of equations until stopping criteria $|\|R(t+\delta)\| - \|R(t)\|| < \nu$ is satisfied.
   b. Take the obtained time $T = t$ and compute $Q_j(T)$.
3. **Construct adjacency matrix**
   a. Compute pairwise cosine distances between all points
   b. Define adjacency matrix $A$ based on threshold $\epsilon$.
4. **Extract clusters**
   a. Identify connected components of the graph represented by $A$.
5. **Return clusters**
   a. Output the set of clusters $C_1, C_2, \dots, C_k$ corresponding to groups of points with high mutual similarity.

This method enables the automatic clustering of points on the unit hypersphere based on their emergent dynamics. A classical fourth-order Runge-Kutta method is used for solving systems of ODE's. All simulations were performed using R (version 4.4.2), employing built-in ODE solvers from the Runge-Kutta family. Visualizations were created using Wolfram Mathematica (version 13.1).

## 3. Simulations

In this section, we will evaluate the effectiveness of our clustering algorithm. To achieve this, we test its performance on various synthetic and real-world datasets and compare the results with the spkmeans (Spherical K-Means Clustering) [30] and movMF (Mixtures of von Mises-Fisher Distributions) [31] algorithms. These methods were chosen because they are widely used and specifically designed for hyperspherical data. Unlike spkmeans and movMF, our algorithm does not require the number of clusters to be specified in advance, making it practical for unsupervised settings. The experiments are implemented in R, and clustering effectiveness is measured using macro-recall, macro-precision, Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI). Macro-recall represents an average recall over all classes and is calculated as:

$$Macro - recall = \frac{1}{C}\sum_{i=1}^{C}\frac{TP_i}{TP_i+FN_i},$$

where $C$ represents a number of classes, and $TP_i$ and $FN_i$ are true positives and false negatives for class $i$, respectively. Macro-precision represents an average precision over all classes:

$$Macro - precision = \frac{1}{C}\sum_{i=1}^{C}\frac{TP_i}{TP_i+FP_i},$$

where $FP_i$ is a false positive for class $i$. Macro-recall and macro precision metrics are useful when dealing with imbalanced datasets. While macro metrics focus on class-wise accuracy, NMI and ARI measure overall agreement between predicted clusters and true labels regardless of label order. Range for NMI is from 0 to 1, with 1 indicating perfect agreement. Range for ARI is from -1 to 1, where 1 means perfect match and 0 is random labelling.

### 3.1 Synthetic datasets

We will utilize random data generated from the von Mises-Fischer distribution [19], a probability distribution over unit vectors on a sphere, to test the performance of our clustering algorithm. For these purposes, we have created two random datasets Dat_1 and Dat_2, one with 150 three-dimensional unit vectors, and the other one with 200 four-dimensional unit vectors.

In the dataset Dat_1, we have three clusters of 50 data points generated from von Mises-Fischer distribution, where the first cluster has a mean direction of $\mu_1 = (1,0,0)$, the second cluster has a mean direction of $\mu_2 = (0,1,0)$, and the third cluster has a mean direction of $\mu_3 = (0,0,1)$. The concentration parameter is fixed for all clusters and is set to $\kappa = 20$.

Results of macro-recall, macro-precision, NMI and ARI for different clustering techniques over this dataset are given in Table 1. The proposed algorithm achieves the highest values across all metrics and clusters data into five clusters instead of the original three, with two of them being identified as outliers. The ability to detect and separate outliers demonstrates the algorithm's sensitivity to distinct patterns and anomalies within the dataset. The proposed algorithm does not know the number of clusters in advance, while the other two require it to be specified at the start.

Table 1

**Clustering report for the dataset Dat_1**

| Algorithm | Macro-recall | Macro-precision | ARI | NMI | Number of clusters |
|---|---|---|---|---|---|
| spkmeans | 0.980 | 0.980 | 0.940 | 0.911 | 3 |
| movMF | 0.980 | 0.980 | 0.940 | 0.911 | 3 |
| The Algorithm | **0.987** | **0.987** | **0.960** | **0.942** | 5 |

Fig. 1. illustrates the clustering results on synthetic dataset Dat_1 sampled from $\mathbb{S}^2$, obtained at time T=1.27.

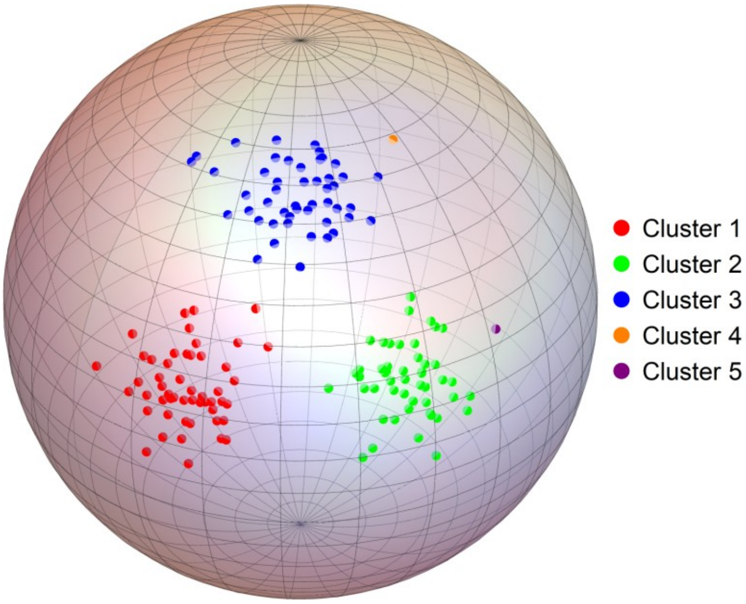

Fig. 1. Simulation results for the dataset Dat_1 with clusters obtained at time T=1.27. The algorithm identified five clusters, including two (Cluster 4 and 5) corresponding to outliers.

Dat_2 consists of 200 five-dimensional data points with 2 clusters generated with parameters $\mu_1 = (1,0,0,0,0)$, $\mu_2 = (-1,0,0,0,0)$, and $\kappa = 20$. The data points lie in five-dimensional space and therefore cannot be directly visualized.

Fig. 2. illustrates the order parameter over time, showing the synchronization process. When the order parameter reaches 1, full synchronization occurs. We select a moment before this point to examine the number of clusters.

Results presented in Table 2 are obtained at moment T = 4.6.

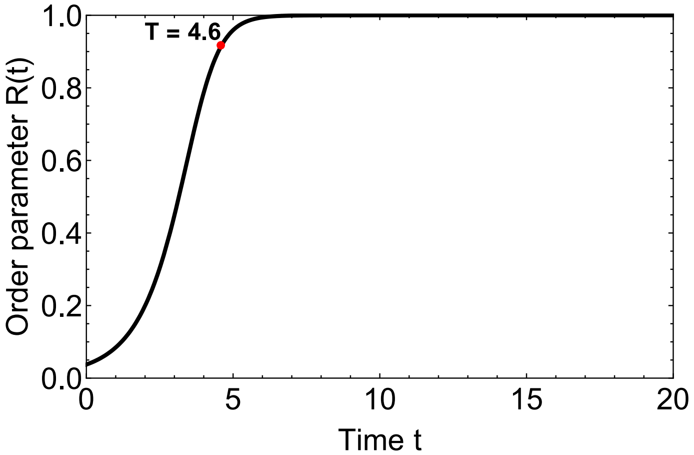

Fig. 2. Order parameter during the synchronization process for the dataset Dat_2

The moment ***T*** is chosen just before full synchronization occurs. That is a moment when the order parameter stops changing much. At that moment we find meaningful clusters. This approach is similar to the hierarchical clustering and cutting a dendrogram at the right level to find meaningful clusters before everything merges into one. Table 2 presents the macro-recall, macro-precision, NMI, and ARI results for the algorithms on the second dataset.

Table 2

**Clustering report for the dataset Dat_2**

| Algorithm | Macro-recall | Macro-precision | ARI | NMI | Number of clusters |
|---|---|---|---|---|---|
| spkmeans | 0.995 | 0.995 | 0.980 | 0.959 | 2 |
| movMF | **1.000** | **1.000** | **1.000** | **1.000** | 2 |
| The Algorithm | 0.980 | 0.995 | 0.980 | 0.959 | 2 |

The algorithm has demonstrated its applicability to higher dimensions. In the case of five-dimensional unit vectors, it achieved macro-precision, macro-recall, NMI, and ARI results that are competitive with the compared algorithms.

### 3.2 Real-world datasets

We can evaluate our method using real-world datasets. Real-world datasets used are household expenditure survey [32] and Iris dataset [33]. The data points are projected onto the sphere by scaling them to have unit length.

Household dataset is obtained from the R software and "HSAUR2" package. This dataset is a survey on household expenditure that has data separated into 2 clusters, men and women based on four parameters: housing, food, goods, and services. This dataset has 40 data points belonging to a dimension $\mathbb{S}^3$. Each of these $\mathbb{R}^4$ vectors are normalized to have a unit length of 1. Table 3 presents the macro-recall, macro-precision, NMI, and ARI results for the algorithms on the Household dataset.

Table 3

**Clustering report for the Household dataset**

| Algorithm | Macro-recall | Macro-precision | ARI | NMI | Number of clusters |
|---|---|---|---|---|---|
| spkmeans | 0.825 | 0.847 | 0.408 | 0.371 | 2 |
| movMF | 0.825 | 0.870 | 0.409 | 0.443 | 2 |
| The Algorithm | **0.850** | **0.885** | **0.478** | **0.510** | 2 |

Our algorithm outperformed both spkmeans and movMF algorithms in all evaluated metrics.

Iris dataset has measurements of 150 iris flowers from three different species: setosa, versicolor, and virginica. The dimensionality of this dataset is 4 and its data is represented as points on $\mathbb{S}^3$. Table 4 summarizes the macro-recall, macro-precision, NMI, and ARI results on the Iris dataset.

Table 4

**Clustering report for the Iris dataset**

| Algorithm | Macro-recall | Macro-precision | ARI | NMI | Number of clusters |
|---|---|---|---|---|---|
| spkmeans | 0.967 | **0.969** | **0.904** | **0.898** | 3 |
| movMF | 0.953 | 0.959 | 0.868 | 0.871 | 3 |
| The Algorithm | **1.0** | 0.667 | 0.568 | 0.734 | 2 |

The method identifies two clusters: one that precisely matches Iris setosa, while the other combines Iris virginica and Iris versicolor. This result aligns with expectations in unsupervised learning, as these two species cannot be easily distinguished without category labels. Both the spkmeans and movMF algorithms showed potential instability, as different runs with varying random seeds produced different results. Our simulations confirmed this behaviour, suggesting sensitivity to initialization. In contrast, our algorithm produced consistent clustering outcomes across multiple runs.

## 4. Conclusion

In this paper, we have introduced a new algorithm for performing directional data clustering. This algorithm belongs to the group of synchronization clustering algorithms because it is based on the extension of the classical Kuramoto model to the unit hypersphere. We demonstrated the effectiveness of our algorithm on both real-world and synthetic datasets, showing that it achieves comparable or superior clustering accuracy compared to established techniques such as spherical k-means and the movMF algorithm. The algorithm showed effectiveness in identifying outliers inside data. Given that the algorithm does not require defining the number of clusters in advance and autonomously discovers the structure of groups, it belongs to the category of unsupervised algorithms. This feature makes it more practical in real problems where in most cases the number of groups is not known in advance. However, its reliance on numerically solving differential equations introduces computational cost, especially for large datasets. As part of future work, we plan to extend the evaluation to larger datasets, improve computational cost, and investigate the performance on other non-Euclidean manifolds.

## REFERENCES

[1] *K. V. Mardia, P.E. Jupp*, Directional statistics, Vol. **2**, Wiley, New York, 1999.

[2] *N. I. Fisher*, Statistical analysis of circular data, Cambridge University Press, 1993.

[3] *A. Pawsey, E. García-Portugués,* Recent advances in directional statistics, TEST, Vol. **30**, 2021, pp. 1-58, DOI: 10.1007/s11749-021-00759-x

[4] *U. Lund*, Cluster analysis for directional data, Commun. Stat. Simul. Comput., Vol. **28**, 1999, pp. 1001–1009, DOI: 10.1080/03610919908813589

[5] *D. Zhang, X. Jin, H. Su*, A perspective on attitude control issues and techniques, 2022.

[6] *D. Rancourt, L-P. Rivest, J. Asselin*, Using orientation statistics to investigate variations in human kinematics, J. R. Stat. Soc. Ser. C, Vol. **49**, 2000, pp. 81–94.

[7] *H. Mittal, A. C. Pandey, M. Saraswat, S. Kumar, R. Pal, G. Modwel*, A comprehensive survey of image segmentation: clustering methods, performance parameters, and benchmark datasets, Multimed. Tools Appl., Vol. **81**, 2022, pp. 35001–35026, DOI: 10.1007/s11042-021-10594-9

[8] *T. Kansal, S. Bahuguna, V. Singh, T. Choudhury*, Customer segmentation using K-means clustering, Proc. Int. Conf. CTEMS, IEEE, Belgaum, India, 2018, pp. 135–139, DOI: 10.1109/CTEMS.2018.8769171

[9] *J. Oyelade, I. Isewon, F. Oladipupo, O. Aromolaran, E. Uwoghiren, F. Ameh, M. Achas, E. Adebiyi*, Clustering algorithms: their application to gene expression data, Bioinform. Biol. Insights, Vol. **10**, 2016, DOI: 10.4137/BBI.S38316

[10] *R. A. Ariyaluran Habeeb, F. Nasaruddin, A. Gani, M. A. Amanullah, I. Abaker Targio Hashem, E. Ahmed, M. Imran*, Clustering-based real-time anomaly detection—A breakthrough in big data technologies, Trans. Emerg. Telecommun. Technol., Vol. **33**, 2022, DOI: 10.1002/ett.3647

[11] *J.B. MacQueen*, Some methods for classification and analysis of multivariate observations, Proc. 5th Berkeley Symp. Math. Stat. Probab., Vol. **1**, 1967, pp. 281–297.

[12] *A. Ng, M. Jordan, Y. Weiss*, On spectral clustering: analysis and an algorithm, Adv. Neural Inf. Process. Syst., Vol. **14**, 2001.

[13] *Z. Rustam, F.A. Leudityara*, Breast cancer clustering using modified spherical K-means, J. Phys. Conf. Ser., Vol. **1490**, 2020, DOI: 10.1088/1742-6596/1490/1/012028

[14] *Arfiani, Z. Rustam, J. Pandelaki, A. Siahaan*, Kernel spherical K-means and support vector machine for acute sinusitis classification, IOP Conf. Ser. Mater. Sci. Eng., Vol. **546**, 2019, DOI: 10.1088/1757-899X/546/5/052011

[15] *I.S. Dhillon, Modha D S*, Concept decompositions for large sparse text data using clustering, *Mach. Learn.*, Vol. **42**, 2001, pp. 143–175, DOI: 10.1023/A:1007612920971

[16] *F. Murtagh, P. Contreras*, Algorithms for hierarchical clustering: an overview, *WIREs Data Min. Knowl. Discov.*, Vol. **2**, 2012, pp. 86–97, DOI: 10.1002/widm.53

[17] *M. Golzy, M. Markatou, A. Shivram*, Algorithms for clustering on the sphere: advances & applications, *Proc. World Congr. Eng. Comput. Sci.*, Vol. **1**, 2016, pp. 420–425.

[18] *A. Banerjee, I. Dhillon, J. Ghosh, S. Sra*, Expectation maximization for clustering on hyperspheres, Technical Report, 2003.

[19] *A. Banerjee, I. S. Dhillon, J. Ghosh, S. Sra, G. Ridgeway*, Clustering on the unit hypersphere using von Mises-Fisher distributions, *J. Mach. Learn. Res.*, Vol. **6**, 2005.

[20] *H. D. Nguyen*, A novel algorithm for clustering of data on the unit sphere via mixture models, *arXiv preprint*, arXiv:1709.04611, 2017.

[21] *S. U. Rehman, S. Asghar, S. Fong, S. Sarasvady*, DBSCAN: past, present and future, *Proc. 5th Int. Conf. ICADIWT*, IEEE, Bangalore, India, 2014, pp. 232–238, DOI: 10.1109/ICADIWT.2014.6814687

[22] *J. Shao, X. He, C. Böhm, Q. Yang, C. Plant*, Synchronization-inspired partitioning and hierarchical clustering, *IEEE Trans. Knowl. Data Eng.*, Vol. **25**, 2013, pp. 893–905, DOI: 10.1109/TKDE.2012.32

[23] *C. Böhm, C. Plant, J. Shao, Q. Yang*, Clustering by synchronization, *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, Washington DC, USA, ACM, 2010, pp. 583–592, DOI. 10.1145/1835804.1835879

[24] *X. Chen*, A fast synchronization clustering algorithm, *arXiv preprint*, arXiv:1407.7449, 2014.

[25] *Y. Kuramoto*, Self-entrainment of a population of coupled non-linear oscillators, Int. Symp. Math. Probl. Theor. Phys., 1975, pp. 420–422, DOI: 10.1007/bfb0013365.

[26] *A. Arenas, A. Díaz-Guilera, J. Kurths, Y. Moreno, C. Zhou*, Synchronization in complex networks, Phys. Rep., Vol. **469**, 2008, pp. 93–153, DOI: 10.1016/j.physrep.2008.09.002.

[27] *A. Crnkić, V. Jaćimović*, Data clustering based on quantum synchronization, Natural Computing, Vol. **18**, 2018, pp. 907-911, DOI: 10.1007/s11047-018-9720-z

[28] *Z. Kapić, A. Crnkić, V. Jaćimović, N. Mijajlović,* A new dynamical model for solving rotation averaging problem, 20th International Symposium INFOTEH-JAHORINA (INFOTEH), 2021, DOI: 10.1109/INFOTEH51037.2021.9400663

[29] *Z. Kapić, A. Crnkić,* Interpolation on the unit hypersphere using the n-dimensional generalized Kuramoto model, Vol. **1298**, No. 1, 2023, pp. 12-22, DOI: 10.1088/1757-899X/1298/1/012022

[30] *K. Hornik, I. Feinerer, M. Kober, C. Buchta,* Spherical k-Means Clustering, Journal of Statistical Software, Vol. **50**, 2012, DOI: 10.18637/jss.v050.i10

[31] *K. Hornik, B. Grün,* movMF: An R Package for Fitting Mixtures of von Mises-Fischer Distributions, Journal of Statistical Software, Vol. **58**, 2014, DOI: 10.18637/jss.v058.i10

[32] *B. Everitt, T. Hothorn,* A handbook of statistical analyses using R, Chapman and Hall/CRC, 2009

[33] *R. A. Fischer,* The use of multiple measurements in taxonomic problems, Annals of eugenics, Vol. **7**, No. 2, 1936, pp. 179-188