engineering journals
Redefining Knowledge

uniform distribution theory

DOI: 10.66833/UDT-2026-0004
Unif. Distrib. Theory 21 (2026), no.1, 1-16

On the tensor approximation of Watts–Strogatz networks

J. Alberto Conejero[1,*], Antonio Falcó[2] and María Mora-Jiménez[3]

[1]Instituto Universitario de Matemática Pura y Aplicada. Universitat Politécnica de València. València. E46022. Spain.
[2]ESI International Chair CEUUCH Departamento de Matemáticas Física y Ciencias Tecnológicas. Universidad Cardenal Herrera. CEU CEU. Universities Alfara del Patriarca. 46115 - València. Spain.
[3]EDEM Escuela de Empresarios Muelle de la Aduana sn. Poblados Marítimos. 46024 - València. Spain.

ABSTRACT. Small-world networks are characterized by the existence, on average, of shortest paths between any arbitrary pair of nodes with only a few edges. In order to preserve the local clustering while permitting the existence of the small-world phenomenon, Watts and Strogatz introduced their celebrated network model (Nature, 1998) exemplifying what they observed in different types of networks, such as the neural network of the *C. elegans*, the power grid network, or the collaboration network in cinema.

As the number of data increases, the networks and matrices that model it also do so, which makes it increasingly expensive to manipulate them. Recent work has shown the efficiency of tensor-based structures when performing, for example, matrix products, reducing the number of operations performed.

In the present work, we want to approximate the representative matrices of the Watts–Strogatz networks using tensor methods and compare the accuracy and the computational cost involved in operating with the original matrices and the matrices written in the approximate tensor form.

2020 Mathematics Subject Classification: 11B83, 11D61
Keywords: regular graphs, small world networks, Watts-Strogatz networks, tensor decomposition and approximations, eigenvalues and eigenvectors.

1 Introduction

Nowadays, networks are widely used to study and model different situations in real life, so they are a tool that gains importance in manipulating large data sets. For example, they have been used to analyze connections in a social network [1][2][3], the spreading of information in a network [4] to determine optimal transport routes [5][6], and even to analyze medical images [7] or discover new drugs from chemical data [8][9].

As the size of the matrices or vectors involved in these problems increases, most of the mechanisms we use to solve them (matrix decompositions, iterative methods, optimization algorithms, etc.) lose efficiency. This effect is known as the *curse of the dimensionality problem*. In the current situation, where massive data analysis is increasingly present, we need to search for and implement techniques that allow us to manipulate them. This is where tensor structures come into play since their use reduces and speeds up the number of operations to be carried out, as it has been recently seen [10]. Other works that illustrate the goodness of these structures in working with high-dimensional problems are [11][12][13].

One of the most intriguing properties in networks is the *small world phenomenon*, firstly conjectured by Milgram [14]. A small-world network is a type of graph where most of the nodes are not neighbors of each other, but a relatively small edge path can connect two randomly chosen nodes [15][16]. In this work, we analyze the tensor decompositions of the Laplacian matrices of small-world networks in the sense of Watts and Strogatz, and compare the results obtained with the original Laplacian matrices to determine if this approach is helpful to study these networks.

The Laplacian matrix of a network is obtained as the difference between the degree and the adjacency matrix of the network. These matrices are commonly used in spectral graph theory to study the properties of the graph, in relation to its characteristic polynomial, and for representing diffusion processes on networks. They are also used to study some particular diffusion problems, as seen in [17]. Watts–Strogatz networks are constructed from a given k-regular graph, where some edges are rewired to connect a different pair of nodes with a certain probability p. For low values of p, the Laplacian matrices of these networks have a structure close to a $k+1$ diagonal matrix.

Motivated by the good properties of tensor products [18][19], we can look for a tensor decomposition of a given Laplacian matrix $\mathcal{L}$ in the form of

$$\mathcal{L} \approx A_1 \otimes \mathrm{id}_{n_2} \otimes \cdots \otimes \mathrm{id}_{n_d} + \mathrm{id}_{n_1} \otimes A_2 \otimes \cdots \otimes \mathrm{id}_{n_d} + \cdots + \mathrm{id}_{n_1} \otimes \mathrm{id}_{n_2} \otimes \cdots \otimes A_d, \qquad (1)$$

where $A_i \in \mathbb{R}^{n_i \times n_i}$ with $N = n_1 \ldots n_d$. As shown in [20], we can define an algorithm that provides the best tensor approximation in form (1) of any square matrix. In particular, for the Laplacian matrices of graphs with N nodes, this decomposition is an interesting approximation of $\mathcal{L}$, when calculating their eigenvalues or eigenvectors from algorithms that use tensors, and that, therefore, are more efficient in calculation times. So, we want to study

the algorithm's performance in [20] over Laplacian matrices of Watts–Strogatz networks. In particular, we want to study how close the obtained tensor decomposition is to the Laplacian matrices of the k-regular network used for generating the Watts–Strogatz network. In order to measure how close are the initial and the tensor decomposed networks, we compare the distribution of eigenvalues and eigenvectors of their corresponding Laplacian matrices [21].

Laplacian eigenvalues and eigenvectors are relevant to multiple aspects of complex network structures, like spanning trees [22], community structure [23] or algebraic connectivity [24]. Moreover, they represent some significant physical or chemical properties of networks, and help to understand the relations between their topology and the dynamics. For example, in the framework of generalized Gaussian structures, the dynamics of polymer networks are fully described through the Laplacian eigenvectors and eigenvalues [25][26].

The paper is divided as follows: In Section 2, we briefly review the construction of Watts–Strogatz networks from k-regular networks. Then, in Section 3, we recall some concepts about tensors and tensor decompositions, and we will describe the methodology used for obtaining tensor decompositions in order to obtain approximations of the Laplacian matrices of Watts–Strogatz networks generated from circular 2-regular and 4-regular networks. Finally, we show our results and discuss them in Sections 4 and 5, respectively.

2 Regular networks and the Watts–Strogatz networks

Let us consider a non-directed simple connected network $G = (V, E)$, where $V = \{v_1, \dots, v_n\}$ is the set of nodes, and E is the set of edges. We recall that the adjacency matrix $\mathcal{A}$ of G is defined as $\mathcal{A}_{ij} = 1$ if $(v_i, v_j) \in E$ and 0 elsewhere. Since the network is simple, all elements in the diagonal satisfy $\mathcal{A}_{ii} = 0$, and since the network is non-directed $\mathcal{A}_{ij} = \mathcal{A}_{ji}$ for all $1 \leq i, j \leq n$.

The degree of a vertex v_i, denoted by $\delta(v_i)$, is defined as the number of edges incident on it. We can define the *degree matrix* $\mathcal{D}$ of G as a diagonal matrix where $\mathcal{D}_{ii} = \delta(v_i)$ for all $1 \leq i \leq n$. Then, the *Laplacian matrix* $\mathcal{L}$ is defined as $\mathcal{L} = \mathcal{D} - \mathcal{A}$. Since $\mathcal{A}$ is symmetric and $\mathcal{D}$ is diagonal, we have that $\mathcal{L}$ is symmetric, too. Laplacian matrices of simple networks satisfy that the sum of all the elements in a row (or column) is equal to 0. Therefore, $\lambda = 0$ is always an eigenvalue (with associated eigenvector $u^\top = (1, \dots, 1)$), of the Laplacian matrix. On the other hand, for every node $v_i \in V$, we can define the state on this node at instant t as $u_i(t)$. So, we can express the diffusion between adjacent nodes as the *Abstract Cauchy Problem* on ℓ_2 defined on (2).

$$\begin{cases} \dot{u}_i(t) = D \sum_{(i,j)\in E} \big(u_j(t) - u_i(t)\big), i = 1, \dots, n \\ u_i(0) = u_0(i), u_0(i) \in \mathbb{R} \end{cases} \tag{2}$$

where D is the diffusion coefficient. If $u = (u_i(t))_{i=1}^n \in \ell_2$, where $u = (u_i(t))_{i=1}^n \in \ell_2$ is the number of nodes, this diffusion equation can be stated as $\mathrm{d}u(t)/dt = -D\mathcal{L}u(t)$, in terms of the Laplacian matrix of the network.

In the sequel, we deal with circular networks, in which we represent the nodes presented over a circumference and link them with their neighbors. We recall that a network is k-regular if $\delta(v_i) = k$ for all $1 \leq i \leq n$. So as to, a circular network C_{2n} is a 2-regular network with n vertices, $n \geq 3$. Up to isomorphism, there is a unique representation of C_{2n}, with the following Laplacian matrix:

$$\mathcal{L} = \begin{pmatrix} 2 & -1 & 0 & \dots & 0 & -1 \\ -1 & 2 & -1 & \dots & 0 & 0 \\ 0 & -1 & 2 & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & -1 & 2 & -1 \\ -1 & 0 & \dots & 0 & -1 & 2 \end{pmatrix}. \tag{3}$$

We can also consider 4-regular circular networks, denoted by C_{4n}, with n nodes, with $n \geq 5$. Here, a node is connected to the next two nodes on the right and the next two on the left until both cycles are completed (doubly circular graphs), see Figure 1.

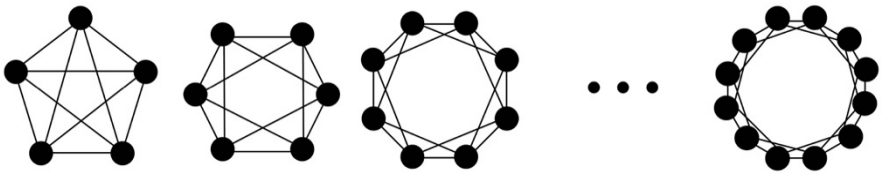

Figure 1. Examples of 4-regular graphs: $C_{4,5}, C_{4,6}, C_{4,7}$, and $C_{4,12}$.

The Laplacian matrix of 4-regular circular network has 4 in all the elements in the main diagonal, and -1 in the two diagonals above and below the main one, as we can see below

$$\mathcal{L} = \begin{pmatrix} 4 & -1 & -1 & 0 & \dots & -1 & -1 \\ -1 & 4 & -1 & -1 & \dots & 0 & -1 \\ -1 & -1 & 4 & -1 & \dots & 0 & 0 \\ 0 & -1 & -1 & 4 & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & -1 & 4 & -1 & -1 \\ -1 & 0 & \dots & -1 & -1 & 4 & -1 \\ -1 & -1 & \dots & 0 & -1 & -1 & 4 \end{pmatrix}. \tag{4}$$

Watts and Strogatz introduced randomness into k-regular graphs to construct networks that preserve a high local clustering coefficient while reducing the average shortest length path between any pair of nodes to resemble the small world phenomenon. They set an algorithm in order to construct these networks. First, we start with a circular $2k$-regular network of n nodes, denoted by $C_{2k,n}$, where each node is connected with its closest 2k neighbor nodes. Then, we set a probability p, and for every existing edge, we decide whether to rewire a different pair of nodes, with probability p, or to leave it as it is.

The case $p = 0$ returns the original $2k$-regular network $C_{2k,n}$. As we increase the value of p, we introduce randomness in the connectivity while maintaining the average

degree equal to $2k$. In the case of $p = 1$, the generated network is completely random. We can appreciate how the network changes while increasing the value of p in Figure 2.

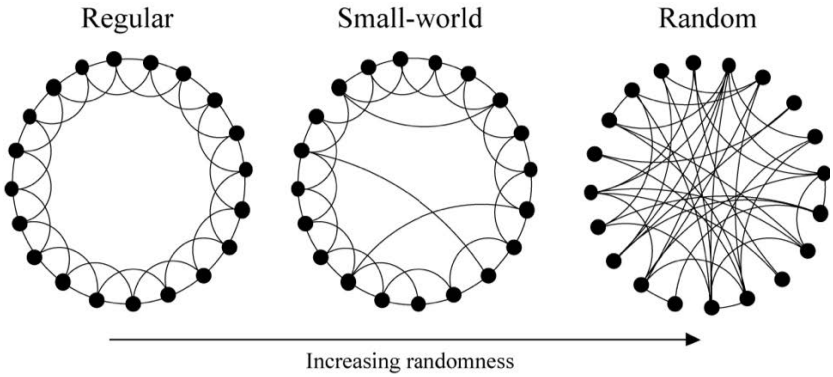

Figure 2. Watts–Strogatz networks constructed from a circular $C_{4,n}$ network for different values of p: $p = 0$ (left), $0 < p < 1$ (center), and $p = 1$ (right) [27].

3 The best tensor based decomposition

First of all, let's remember that the *Kronecker product* of two matrices $A \in \mathbb{R}^{N_1 \times M_1}$, $B \in \mathbb{R}^{N_2 \times M_2}$ is defined by

$$A \otimes B = \begin{pmatrix} A_{1,1}B & A_{1,2}B & \dots & A_{1,M_1}B \\ A_{2,1}B & A_{2,2}B & \dots & A_{2,M_1}B \\ \vdots & \vdots & \ddots & \vdots \\ A_{N_1,1}B & A_{N_1,2}B & \dots & A_{N_1,M_1}B \end{pmatrix} \in \mathbb{R}^{N_1 N_2 \times M_1 M_2}.$$

And, also, some of the well-known properties of the Kronecker product are:

1) $A \otimes (B \otimes C) = (A \otimes B) \otimes C$..
2) $(A + B) \otimes C = (A \otimes C) + (B \otimes C)$.
3) $AB \otimes CD = (A \otimes C)(B \otimes D)$.
4) $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$.
5) $(A \otimes B)^{\mathsf{T}} = A^{\mathsf{T}} \otimes B^{\mathsf{T}}$.
6) $\mathrm{tr}(A \otimes B) = \mathrm{tr}(A) \cdot \mathrm{tr}(B)$.

One of the most popular techniques among the algorithms based on tensor products strategies [28] is the Proper Generalized Decomposition (PGD) family, based on the so-called Greedy Rank-One Updated (GROU) algorithm [29][30]. In particular, they impose a separation of variables to approximate the exact solution of a problem without knowing, in principle, the functions involved in this decomposition [31].

There is a particular type of matrices to solve high-dimensional linear systems for which the GROU algorithm works particularly well, those that are written in the form

$$A = \sum_{i=1}^{d} A_i \otimes \mathrm{id}_{[n_i]} \doteq \sum_{i=1}^{d} \mathrm{id}_{n_1} \otimes \cdots \otimes \mathrm{id}_{n_{i-1}} \otimes A_i \otimes \mathrm{id}_{n_{i+1}} \otimes \cdots \otimes \mathrm{id}_{n_d}, \qquad (5)$$

where $A \in \mathbb{R}^{N\times N}$ with $N = n_1 \ldots n_d$, $A_i \in \mathbb{R}^{n_i \times n_i}$ for $1 \le i \le d$, and id_{n_j} is the identity matrix of size $n_j \times n_j$. These matrices are called Laplacian-like matrices, since they can be easily related to the classical Laplacian operator [32][33]. However, to avoid falling into a nomenclature ambiguity, we will refer to this structure as the tensor decomposition (of rank d) of a matrix.

So, we are interested in trying to achieve a structure similar to (5), to study the matrices of large-dimensional graphs. To do this, we will use the following theorem (Theorem 7 in [20]), which describes the Greedy Algorithm that calculates the best tensor decomposition (of rank d) of a given square matrix A.

Theorem 1. *Let A be a matrix in $\mathbb{R}^{N\times N}$, with $N = n_1 \cdots n_d$ such that $tr(A) = 0$ and let*

$$\Delta = \left\{A \in \mathbb{R}^{N\times N} : A = \sum_{i=1}^{d} A_i \otimes \mathrm{id}_{[n_i]}, \text{ with } \mathrm{tr}(A_i) = 0, i = 1, \ldots, d\right\}.$$

If we consider the following iterative procedure:

1. *Take $X_k^{(0)} = 0$ for $1 \le k \le d$*

2. *For each $\ell \ge 1$ compute for $1 \le i \le d$ the matrix $U_i^{(\ell)}$ as*

$$U_i^{(\ell)} = \arg \min_{\substack{U_i \in \mathbb{R}^{n_i \times n_i} \\ \mathrm{tr}(U_i)=0}} \left\| A - \sum_{k=1}^{i-1} X_k^{(\ell)} \otimes \mathrm{id}_{[n_k]} - \xi(U_i) \otimes \mathrm{id}_{[n_i]} - \sum_{k=i+1}^{d} X_k^{(\ell-1)} \otimes \mathrm{id}_{[n_k]} \right\|,$$

where

$$\xi(U_i) = X_i^{(\ell-1)} + U_i,$$

and put $X_i^{(\ell)} = X_i^{(\ell-1)} + U_i^{(\ell)}$, then

$$\lim_{\ell\to\infty} \sum_{k=1}^{d} X_k^{(\ell)} \otimes \mathrm{id}_{[n_k]} = P_\Delta(A),$$

where $P_\Delta(A)$ solves the problem

$$\min_{A^* \in \Delta} \|A - A^*\|.$$

The algorithm described in the previous theorem can be written in the form of the pseudocode Algorithm 1.

3.1 Tensor decomposition of Watts-Strogatz networks

We study the tensor decomposition of Watts–Strogatz networks for low values of p. The Laplacian matrices of these networks will be very close to the Laplacian matrix of the associated circular k-regular network, with most of their non-null elements grouped in the main diagonals of the matrix.

We will consider the families of 2-regular and 4-regular networks, and we will modify them to get Watts–Strogatz networks for small probability values p, ranging between 0.05 and 0.2. Then, we will compute the Laplacian tensor-based approximation, with canonical rank d, as given in (1). With these numerical experiments, we want to determine whether, given a Watts–Strogatz network, its Laplacian matrix decomposition, written in the form (1), is a good approximation of the Laplacian matrix of the network and, if writing the matrix in tensor form, this speeds up the computations involving it, for example, in the computation of eigenvalues and eigenvectors.

For this purpose, we have analyzed networks of different sizes, from 100 to 2000 nodes per graph. For each network, we have conducted the following study, which consists of two parts.

First part: Analysis of the k-regular network

1) We start with a 2 (or 4) regular circular network and we obtain its Laplacian matrix $\mathcal{L}$.
2) We compute the tensor-based approximation of $\mathcal{L}$, and we denote it by $\mathcal{L}_{\otimes}$. This approximation is obtained using the algorithm described in [20], which is summarized in the pseudocode of Algorithm 1, using a maximum of 50 iterations and a tolerance of $2.22e-4$.
3) To determine how good such an approximation is, the resulting residual will be computed as $\text{res} = \text{norm}(\mathcal{L} - \mathcal{L}_{\otimes})/\text{norm}(\mathcal{L})$.
4) Lastly, we study the relationship between eigenvalues and eigenvectors of $\mathcal{L}$ and $\mathcal{L}_{\otimes}$, and compare the time to obtain them in each case using the command eigs. To measure the relative distance between the eigenvalues obtained, we will use the indicator $\text{dist}_{\lambda} = \text{norm}(\lambda - \lambda_{\otimes})/\text{norm}(\lambda)$; and we will measure the computational time of λ and $\lambda_{\otimes}$ with the tic-toc instruction implemented in Matlab.

It is worth mentioning that each network has a different decomposition depending on the prime factor decomposition of the network size N. Thus, if $N = n_1 \cdots n_d$, we will look for matrices $A_i \in \mathbb{R}^{n_i \times n_i}$. We will call the vector $[n_1, \ldots, n_d]$ as *the vector of sizes.*

For the sake of completeness, we present the pseudocode of the algorithm used for this decomposition, which has been described in [20].

Algorithm 1. Laplacian decomposition Algorithm

procedure LAP(A^*, iter_max, tol)
  $A = A^* - (\mathrm{tr}(A)/N)\mathrm{id}_N$, iter $= 1$, Lap $= 0$
  while iter $<$ iter_max **do**
    $A \leftarrow A -$ Lap
    for $k = 1,2,\dots,d$ **do**
      $P_k(A) = \mathrm{id}_{n_1} \otimes \cdots \otimes \mathrm{id}_{n_{k-1}} \otimes X_k \otimes \mathrm{id}_{n_{k+1}} \otimes \cdots \otimes \mathrm{id}_{n_d}$
      $X_k \leftarrow \min_{X_k} \left\| A - \sum_{i=1}^{k} P_i(A) \right\|$
      Lap $=$ Lap $+ P_k(A)$
    end for
    if $\|A -$ Lap $\| <$ tol **then goto** 14
    end if
    iter $=$ iter $+ 1$
  end while
  return Lap
end procedure

In this first part, there is no randomness introduced, so the tensor approximation $\mathcal{L}_{\otimes}$ of $\mathcal{L}$, for a concrete graph G is always the same.

Second part: Analysis of the Watts–Strogatz networks

1) We take the 2 (or 4) regular circular networks considered in the first part.
2) For each one of these networks, name it G, we convert it into different Watts–Strogatz networks G'_{p_i}, using four different probabilities: $p_1 = 0.05$, $p_2 = 0.1$, $p_3 = 0.15$, and $p_4 = 0.2$.
3) Once fixed p_i and for every edge $e \in E$, we generate a random r_e number from a uniform distribution on (0,1). This number r_e will be used for deciding if we rewire the nodes connected by e or not, using the following rule.
   - If $r_e < p_i$, then the edge e is rewired, connecting a new pair of nodes chosen randomly among all of them that were not already connected.
   - If $p_i \leq r_e$, the edge is maintained as it is.

4) We repeat the process described in the first part for each network G'_{p_i}. We first obtain the associated Laplacian matrices $\mathcal{L}'_{p_i}$, and then we compute the respective approximate Laplacians $\mathcal{L}'_{\otimes,p_i}$.
5) Finally, we obtain the eigenvalues and eigenvectors of matrices $\mathcal{L}'_{p_i}$ and $\mathcal{L}'_{\otimes,p_i}$, and we measure the difference between the eigenvectors and eigenvalues, as described at the end of the first part.

To make a more robust and consistent analysis in this second part, in order to avoid randomness effects, we repeat 1000 times this scheme. The charts and results in Section 4 will show the averaged values obtained from the 1000 repetitions.

4 Results analysis

We will now show and comment on the results of the experiment described in Section 3.1. The experiments were carried out on a computer with the following characteristics: 11th Gen Intel(R) Core(TM) i7-11370H @ 3.30GHz, RAM 16,0 GB, 64 bit operating system; and a Matlab version R2021b 0.

First, we compare the Laplacian matrices of the 2 and 4-regular graphs and their tensor approximations. In Figure 3, we compare the computation time taken to compute the eigenvalues in the starting $\mathcal{L}$ Laplacian matrices and their approximate $\mathcal{L}_{\otimes}$ matrices in tensor form. As the number of nodes increases, the computation time for the original matrices increases much faster than the times required by the tensor-based approximate matrices.

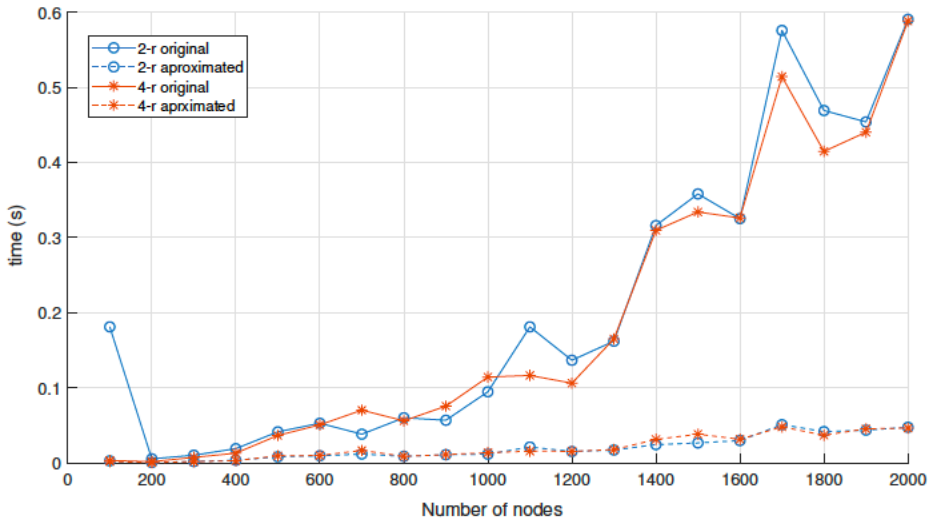

Figure 3. Eigenvalue calculation time

We also compare the eigenvalues of each of the matrices $\mathcal{L}$ and $\mathcal{L}_{\otimes}$, measuring the norm of their difference.

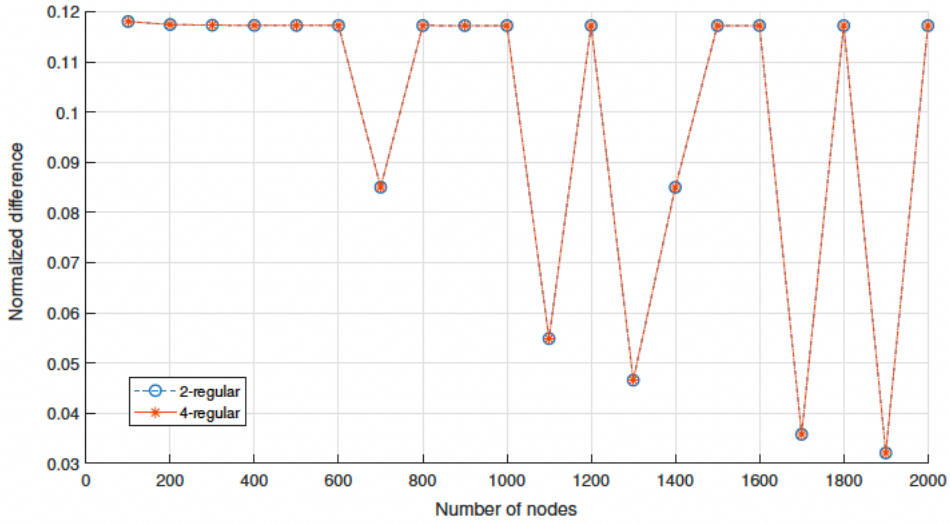

Figure 4. Distance (norm) between eigenvalues

In Figure 4, we see that the results oscillate between the values 0.03 and 0.12. So that, the eigenvalues of the $\mathcal{L}_{\otimes}$ approximate pretty well the eigenvalues of the original matrix $\mathcal{L}$. We observe that the distance between the eigenvalues of the Laplacian matrix and its approximation depends on the network size but not on being a 2-regular or a 4-regular network.

Also, noteworthy that there are cases where the approximations are much better than in the rest. This occurs for 700, 1100, 1300, 1700, and 1900 nodes, where prime numbers $7, 11, 13, 17$, and 19 are involved in the factorization of the network size, which yields vectors of sizes of shorter lengths. For example, if we look at $N = 1700$, the vector of sizes obtained from the factorization of N will be $\text{factor}(N) = [2,2,5,5,17]$; which is shorter than the vector associated to $N = 1800$, $\text{factor}(N) = [2,2,2,3,3,5,5]$. This fact illustrates the importance of the number (and which) prime factors appear in the decomposition of the network size into prime factors.

In the second part, we study the approximations obtained of the Watts–Strogatz Laplacian matrices. In this case, we group the figures into two different groups: in Figure 5 and Figure 6, we show the computation times of eigenvectors of the original Laplacian matrices and those written in tensor form for the different probabilities $p_1, \ldots, p_4$.

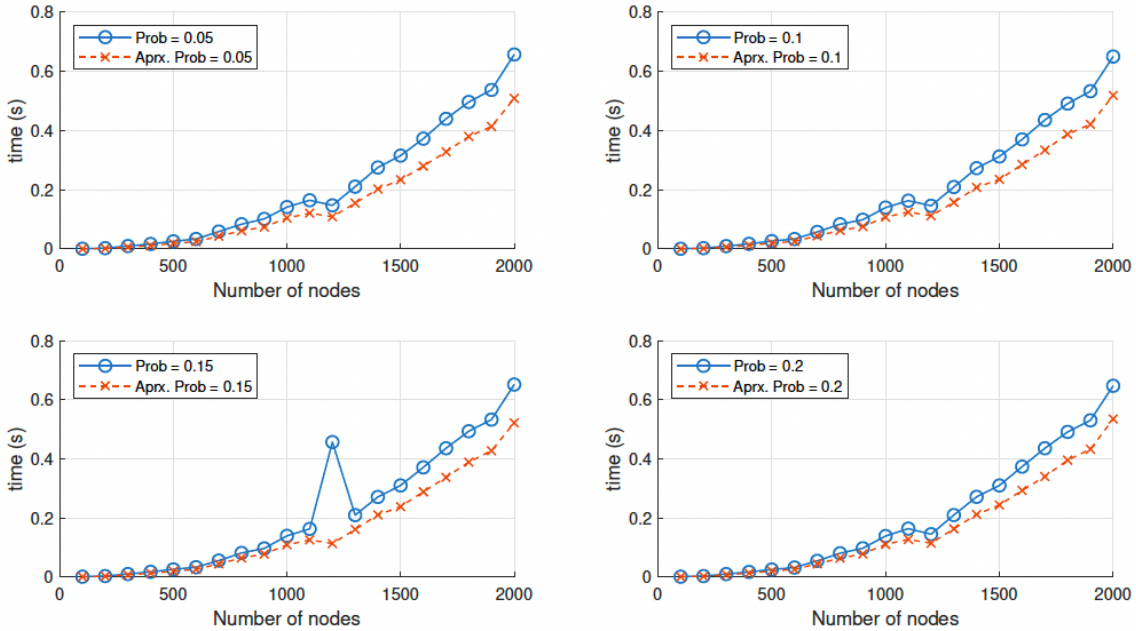

Figure 5. Mean time spent computing eigenvectors in Watts–Strogatz networks from 2-RG

Again, we see that we need less time to calculate eigenvalues and eigenvectors of the tensor approximate matrices than for the original Laplacian ones. Besides, the time increases as long as the network size does.

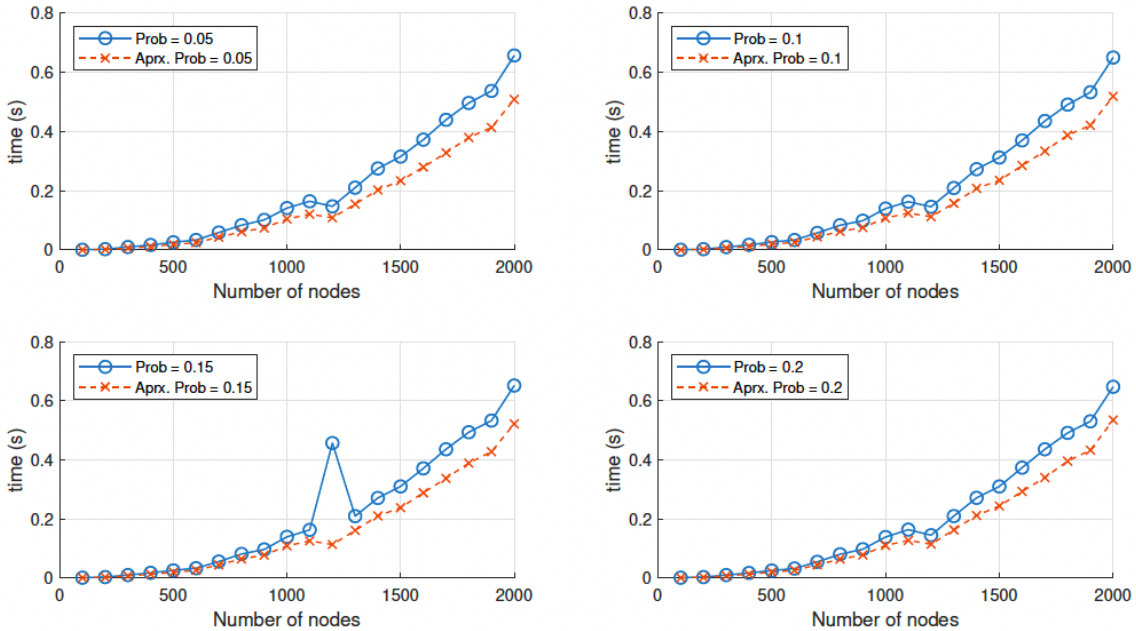

Figure 6. Mean time spent computing eigenvectors in Watts–Strogatz networks from 4-RG

Finally, in Figure 7 and Figure 8, we measure the difference between the eigenvalues of the original Laplacian matrices and those from the tensor-based approximations.

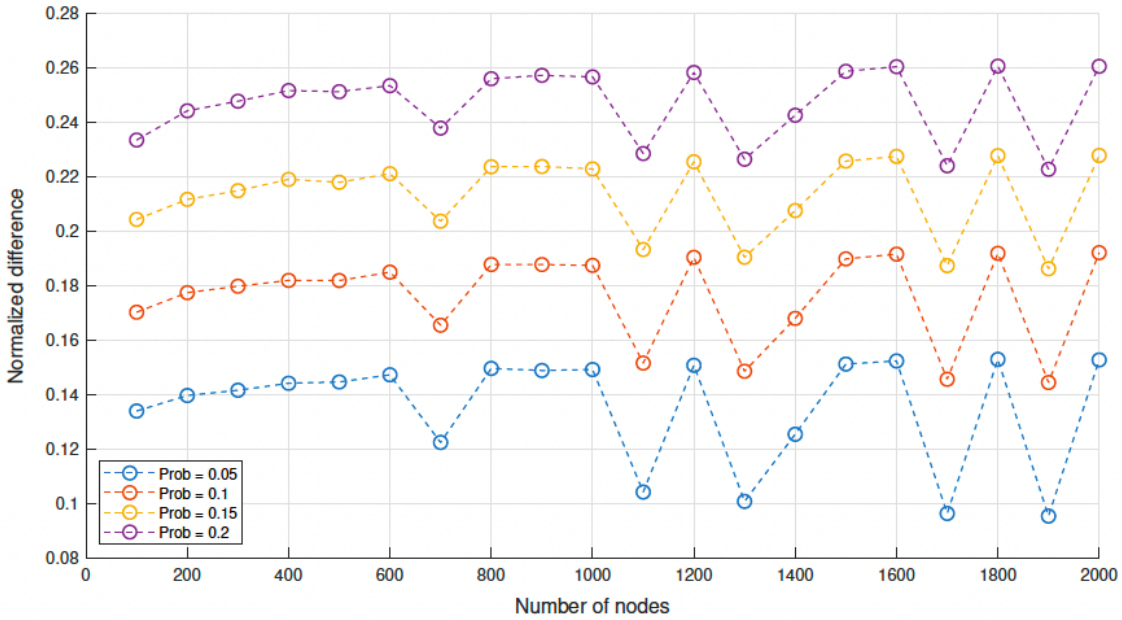

Figure 7. Difference (in norm) between the eigenvalues obtained in Watts–Strogatz networks from 2-regular graphs.

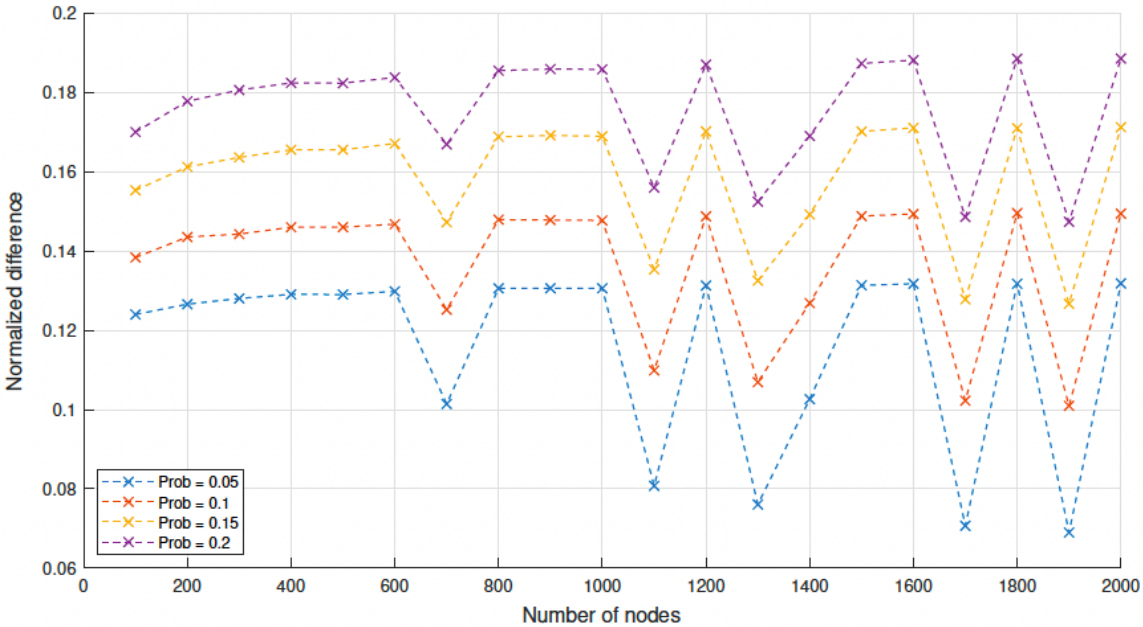

Figure 8. Difference (in norm) between the eigenvalues obtained in Watts–Strogatz networks from 4-regular graphs.

The norms of the difference take small values, less than 0.26 for Watts–Strogatz networks obtained from 2-regular networks, and less than 0.19 for those obtained from 4-regular networks. As the probability increases, we obtain worse approximations, since networks become more *random* and less *regular*, which results in a structure that is harder

to approximate in tensor form. It is also worth mentioning that the prime factorization of the matrix size is relevant, as the algorithm works in blocks. For instance, if we compute the sum of the cubes of the prime factors for $n = 1000,1100,\ldots,2000$, we find that the sum for $1100, 1300, 1700$, and 1900 is greater than for the rest. This observation is consistent with the performance plots, where the execution time for the other dimensions is higher than for those ones. On the other hand, the accuracy of the decomposition may improve for those values of n that have a dominant factor in their product representation. Therefore, a trade-off between decomposition accuracy and execution time must be considered.

Nevertheless, we remark the importance of the vectors of sizes and which prime numbers are involved in the factorization, as the spikes in the graph appeared similarly for each network size.

5 Conclusions

In this work, we have used tensor decompositions of the form (1) to approximate Laplacian matrices of 2 and 4-regular networks, and of some Watts–Strogatz networks obtained from these regular networks.

By introducing the randomness in Watts–Strogatz networks, we lose the regularity in the structure of the Laplacian matrices, which is reflected in the approximations that we obtained: as the randomness increases, the matrices in tensor form approximate worse the original Laplacian matrices, although we still obtain interesting results. Based on the results obtained, we can affirm that this technique allows us to work with tensor approximations of the different networks, by reducing the costs of time spent in computations, as it is the case for computing eigenvectors and eigenvalues.

It may be interesting to carry out similar experiments, using regular networks of higher degree $(6,8,\ldots)$ to see if the 'repetition of patterns' is still maintained in the resulting graphs. It is also interesting to ask what would happen if we modified the *vector of sizes*, that is, the number of matrices that we use in each decomposition and what would be the optimal size of the matrices in that case. For example, if we only use two matrices, are the results better if the matrices have similar sizes, or if, on the contrary, they have the smallest and largest possible size, respectively? Therefore, there are still interesting studies to be carried out to estimate the goodness of the use of tensor decompositions.

Acknowledgments

JAC acknowledges funding from grant PID2021-124618NB-C21 funded by MCIN/AEI/ 10.13039/501100011033 and by "ERDF A way of making Europe", by the "European Union". MMJ is supported by the Generalitat Valenciana and the European Social Found under Grant [number ACIF/2020/269)]; AF is supported by Ministerio de

Ciencia, Innovación y Universidades under Grant [number RTI2018-093521-B-C32]; Universidad CEU Cardenal Herrera under Grant [number INDI20/13].

REFERENCES

[1] L. Freeman, The development of social network analysis, A study in the sociology of science 1 (2004) 159–167.

[2] M. Newman, D. Watts, S. Strogatz, Random graph models of social networks, Proc. Natl. Acad. Sci. USA 99 (2002) 2566–2572. doi:10.1073/pnas.012582999.

[3] W. Jiang, G. Wang, M. Bhuiyan, J. Wu, Understanding graph-based trust evaluation in online social networks: Methodologies and challenges, ACM Comput. Surv. 49 (2016). doi:10.1145/2906151.

[4] R. Pastor-Satorras, A. Vespignani, Epidemic spreading in scale-free networks, Phys. Rev. Let. 86 (2001) 3200.

[5] R. Żochowska, P., Soczówka, Analysis of selected transportation network structures based on graph measures, Scientific Journal of Silesian University of Technology. Series Transport 98 (2018) 223–233. doi:10.20858/sjsutst.2018.98.21.

[6] M. Kanchana, K. Kavitha, A comparative study of transportation problem by graph theoretical algorithms, Advances in Mathematics: Scientific Journal 9 (2020) 6251–6260. doi:10.37418/amsj.9.8.92.

[7] X. Chen, L. Pan, A survey of graph cuts/graph search based medical image segmentation, IEEE Rev. Biomed. Eng. 11 (2018) 112–124. doi:10.1109/RBME.2018.2798701.

[8] M. Ay, K.-I. Goh, M. Cusick, A.-L. Barabasi, M. Vidal, et al, Drug–target network, Nat. Biotechnol. 25 (2007) 1119–1127.

[9] F. MacLean, Knowledge graphs and their applications in drug discovery, Expert Opinion Drug. Discov. 16 (2021) 1–13. doi:10.1080/17460441.2021.1910673.

[10] A. Fawzi, M. Balog, A. Huang, T. Hubert, B. Romera-Paredes, M. Barekatain, A. Novikov, F. Ruiz, J. Schrittwieser, G. Swirszcz, D. Silver, D. Hassabis, P. Kohli, Discovering faster matrix multiplication algorithms with reinforcement learning, Nature 610 (2022) 47–53. doi:10.1038/s41586-022-05172-4.

[11] W. Hackbusch, B. Khoromskij, S. Sauter, E. Tyrtyshnikov, Use of tensor formats in elliptic eigenvalue problems, Numer. Lin. Algebra Appl. 19 (2012) 133–151.

[12] G. Heidel, V. Khoromskaia, B. Khoromskij, V. Schulz, Tensor product method for fast solution of optimal control problems with fractional multidimensional Laplacian in constraints, J. Comput. Phys. 424 (2021) 109865.

[13] C. Quesada, G. Xu, D. González, I. Alfaro, A. Leygue, M. Visonneau, E. Cueto, F. Chinesta, Un método de descomposición propia generalizada para operadores diferenciales de alto orden, Rev. Int. Metod. Numer. 31 (2015) 188–197.

[14] S. Milgram, The small world problem, Psychol. Today 2 (1967) 60–67.

[15] A.-L. Barabàsi, Network science, Cambridge University Press, 2016. URL: http://barabasi.com/networksciencebook/.

[16] D. Watts, Six degrees: The science of a connected age, W. W. Norton & Company, 2003.

[17] J. Banasiak, M. Mokhtar-Kharroubi, Evolutionary Equations with Applications in Natural Sciences, Springer, 2015. doi:10.1007/978-3-319-11322-7.

[18] W. Hackbusch, Tensor Spaces and Numerical Tensor Calculus (Second Edition), Springer Series in Computational Mathematics, Springer Cham, 2019.

[19] V. Simoncini, Numerical solution of a class of third order tensor linear equations, Boll. Unione. Mat. Ital. 13 (2020) 429–439.

[20] J.A. Conejero, A. Falcó, and M. Mora-Jiménez. Structure and approximation properties of Laplacian-like matrice. Results Math. 78 (2023) 184. doi: 10.1007/s00025-023-01960-0.

[21] H. Liu, M. Dolgushev, Y. Qi, Z. Zhang, Laplacian spectra of a class of small-world networks and their applications, Sci. Rep. 5 (2015) 1–7. doi:10.1038/srep09024.

[22] R. Dobrin, P. M. Duxbury, Minimum spanning trees on random networks, Phys. Rev. Lett. 86 (2001) 5076–5079. doi:10.1103/PhysRevLett.86.5076.

[23] M. E. J. Newman, Finding community structure in networks using the eigenvectors of matrices, Phys. Rev. E 74 (2006) 036104. doi:10.1103/PhysRevE.74.036104.

[24] R.B. Bapat, Graphs and matrices, volume 27, Springer, 2010.

[25] C. Liu, Y. Pan, J. Li, L. Dai, The normalized laplacians on both two iterated constructions associated with graph and their applications, J. Appl. Math. Phys. 8 (2020) 838–860. doi:10.4236/jamp.2020.85066.

[26] P. Nakkirt, The eigenvalue distribution of the Watts-Strogatz random graph, arXiv preprint arXiv:2009.00332 (2020).

[27] D. Watts, S. Strogatz, Collective dynamics of 'small-world' networks, Nature 393 (1998) 440–442. doi:10.1038/30918

[28] V. Simoncini, Numerical solution of a class of third order tensor linear equations, Boll. Unione. Mat. Ital. 13 (2020) 429–439.

[29] A. Ammar, F. Chinesta, A. Falcó, On the convergence of a Greedy Rank-One Update algorithm for a class of linear systems, Arch. Comput. Methods Eng. 17 (2010) 473–486.

[30] I. Georgieva, C. Hofreither, Greedy low-rank approximation in Tucker format of solutions of tensor linear systems, J. Comput. Appl. Math. 358 (2019) 206–220.

[31] A. Falcó, A. Nouy, Proper generalized decomposition for nonlinear convex problems in tensor Banach spaces, Numer. Math. 121 (2012) 503–530.

[32] W. Hackbusch, B. Khoromskij, S. Sauter, E. Tyrtyshnikov, Use of tensor formats in elliptic eigenvalue problems, Numer. Lin. Algebra Appl. 19 (2012) 133–151.

[33] G. Heidel, V. Khoromskaia, B. Khoromskij, V. Schulz, Tensor product method for fast solution of optimal control problems with fractional multidimensional Laplacian in constraints, J. Comput. Phys. 424 (2021) 109865.

MATLAB, version R2021b, The MathWorks Inc., Natick, Massachusetts, 2021.

Received October 22, 2025
Accepted January 1, 2026
Published June 1, 2026

J. Alberto Conejero

Instituto Universitario de Matemática Pura y Aplicada Universitat Politécnica de València

E-mail:

Antonio Falcó

ESI International Chair CEUUCH Departamento de Matemáticas Física y Ciencias Tecnológicas. Universidad Cardenal Herrera. CEU CEU. Universities Alfara del Patriarca. 46115 - València. Spain

María Mora-Jiménez

EDEM Escuela de Empresarios Muelle de la Aduana sn. Poblados Marítimos. 46024 -València. Spain