



Structure and Approximation Properties of Laplacian-Like Matrices

J. Alberto Conejero, Antonio Falcó , and María Mora-Jiménez

Abstract. Many of today's problems require techniques that involve the solution of arbitrarily large systems $A\mathbf{x} = \mathbf{b}$. A popular numerical approach is the so-called Greedy Rank-One Update Algorithm, based on a particular tensor decomposition. The numerical experiments support the fact that this algorithm converges especially fast when the matrix of the linear system is Laplacian-Like. These matrices that follow the tensor structure of the Laplacian operator are formed by sums of Kronecker product of matrices following a particular pattern. Moreover, this set of matrices is not only a linear subspace it is a Lie sub-algebra of a matrix Lie Algebra. In this paper, we characterize and give the main properties of this particular class of matrices. Moreover, the above results allow us to propose an algorithm to explicitly compute the orthogonal projection onto this subspace of a given square matrix $A \in \mathbb{R}^{N \times N}$.

Mathematics Subject Classification. 35A01, 65L10, 65L12, 65L20, 65L70.

Keywords. matrix decomposition, laplacian-like matrix, high dimensional linear system, matrix lie algebra, matrix lie group.

1. Introduction

The study of linear systems is a problem that dates back to the time of the Babylonians, who used words like 'length' or 'width' to designate the unknowns without being related to measurement problems. The Greeks also solved some systems of equations, but using geometric methods [1]. Over the years, mechanisms to solve linear systems continued to be developed until the discovery of iterative methods, the practice of which began at the end of the 19th century, by the hand of the mathematician Gauss. The development of computers in

the mid-20th century prompted numerous mathematicians to delve into the study of this problem [2, 3].

Nowadays, linear systems are widely used to approach computational models in applied sciences, for example, in mechanics, after the discretization of a partial differential equation. There are, in the literature, numerous mechanisms to deal with this type of problem, such as matrix decompositions (QR decomposition, LU decomposition), iterative methods (Newton, quasi-Newton, ...), and optimization algorithms (stochastic gradient descent, alternative least squares, ...), among others, see for instance [4–6]. However, most of them lose efficiency as the size of the matrices or vectors involved increases. This effect is known as the *curse of the dimensionality problem*.

To try to solve this drawback, we can use tensor-based algorithms [7], since their use significantly reduces the number of operations that we must employ. For example, we can obtain a matrix of size 100×100 (i.e. a total of 10.000 entries), from two matrices of size 10×10 multiplied, by means the tensor product, $100 + 100 = 200$ entries [8].

Among the algorithms based on tensor products strategies [9], the Proper Generalized Decomposition (PGD) family, based on the so-called Greedy Rank-One Updated (GROU) algorithm [10, 11], is one of the most popular techniques. PGD methods can be interpreted as ‘a priori’ model reduction techniques because they provide a way for the ‘a priori’ construction of optimally reduced bases for the representation of the solution. 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 [12, 13]. The GROU procedure in the pseudocode is given in the Algorithm 1 (where \otimes denotes the Kronecker product, that is briefly introduced in Sect. 2).

Algorithm 1 Greedy Rank-One Update

```

1: procedure GROU( $\mathbf{b} \in \mathbb{R}^{n_1 \cdots n_d}, A \in \mathbb{R}^{n_1 \cdots n_d \times n_1 \cdots n_d}, \varepsilon > 0, \text{tol}, \text{rank\_max}$ )
2:    $\mathbf{r}_0 = \mathbf{b}$ 
3:    $\mathbf{x} = \mathbf{0}$ 
4:   for  $i = 0, 1, 2, \dots, \text{rank\_max}$  do
5:      $\mathbf{y} = \arg \min_{\mathbf{y}=\mathbf{y}_1 \otimes \dots \otimes \mathbf{y}_d} \|\mathbf{r}_i - A\mathbf{y}\|_2^2$ 
6:      $\mathbf{r}_{i+1} = \mathbf{r}_i - A\mathbf{y}$ 
7:      $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{y}$ 
8:     if  $\|\mathbf{r}_{i+1}\|_2 < \varepsilon$  or  $|\|\mathbf{r}_{i+1}\|_2 - \|\mathbf{r}_i\|_2| < \text{tol}$  then goto 13
9:     end if
10:  end for
11:  return  $\mathbf{u}$  and  $\|\mathbf{r}_{\text{rank\_max}}\|_2$ .
12:  break
13:  return  $\mathbf{u}$  and  $\|\mathbf{r}_{i+1}\|_2$ 
14: end procedure

```

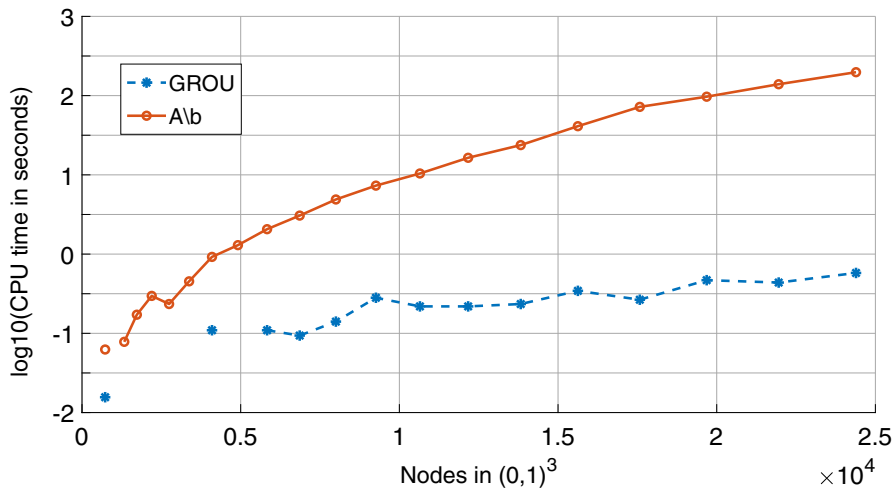


FIGURE 1. CPU time comparative to solve the discrete Poisson equation. For this numerical test, we have used a computer with the following characteristics: 11th Gen Intel(R) Core(TM) i7-11370 H @ 3.30GHz, RAM 16 GB, 64 bit operating system; and a Matlab version R2021b [14]

A good example is provided by the Poisson equation $-\Delta\phi = \mathbf{f}$. Let us consider the following problem in 3D,

$$\begin{cases} \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} + \frac{\partial^2\phi}{\partial z^2} = -\mathbf{f}(x, y, z), & \text{in } \Omega = (0, 1)^3, \\ \phi = 0 & \text{in } \partial\Omega, \end{cases} \quad (1)$$

where $\mathbf{f}(x, y, z) = 3 \cdot (2\pi)^2 \cdot \sin(2\pi x - \pi) \sin(2\pi y - \pi) \sin(2\pi z - \pi)$. This problem has a closed form solution

$$\phi(x, y, z) = \sin(2\pi x - \pi) \sin(2\pi y - \pi) \sin(2\pi z - \pi).$$

By using the Finite Element Method (see [10] for more details), we can write the Poisson equation (1) in discrete form as a linear system $A \cdot \phi_{ijk} = -\mathbf{f}_{ijk}$, where the indices i, j, k correspond to the discretization of x, y and z respectively, and A is matrix having a particular representation, called Laplacian-like (see Definition 2.2 below), that allows to solve efficiently a high dimensional linear system. In Fig. 1, we compare the CPU time employed in solving this discrete Poisson problem using the GROU Algorithm and the Matlab operator $\mathbf{x} = A \setminus \mathbf{b}$, for different numbers of nodes in $(0, 1)^3$.

So, we will use this fact to study if, for a given generic square matrix, a characterization can be stated such that we can decide whether is either Laplacian-like or not. Clearly, under a positive answer, we expect that the analysis of the associated linear system $A\mathbf{x} = \mathbf{b}$ would be simpler. This kind

of linear operator also exists in infinite dimensional vector spaces to describe evolution equations in tensor Banach spaces [15]. Its main property is that the associated dynamical system has an invariant manifold, the manifold of elementary tensors (see [16] for the details about its manifold structure).

Thus, the goal of this paper is to obtain a complete description of this linear space of matrices, showing that is, in fact, a Lie subalgebra of $\mathbb{R}^{N \times N}$, and provide an algorithm in order to obtain the best approximation to this linear space, that is, to compute explicitly is the orthogonal projection on that space.

The paper is organized as follows: in Sect. 2, we introduce the linear subspace of Laplacian-like matrices and prove that it is also a matrix Lie subalgebra associated to a particular Lie group. Then, in Sect. 3, we prove that any matrix is uniquely decomposed as the sum of a Laplacian matrix and a matrix which is the subspace generated by the identity matrix, and we show that any Laplacian matrix is a direct sum of some particular orthogonal subspaces. Section 4 is devoted, with the help of the results of the previous section, to propose an algorithm to explicitly compute the orthogonal projection onto the subspace of Laplacian-like matrices. To illustrate this result, we also give two different numerical examples: the first one on the adjacency matrix of a simple graph; and the second on the numerical solution of the Poisson equation (1) by using a Finite Difference Scheme. Finally, in Sect. 5 some conclusions and final remarks are given.

2. The Algebraic Structure of Laplacian-Like Matrices

First of all, we introduce some definitions, that will be used along this work.

Definition 2.1. Let $A \in \mathbb{R}^{M \times N}$. Then, the Fröbenius norm (or the Hilbert-Schmidt norm) is defined as

$$\|A\|_F = \sqrt{\sum_{i=1}^M \sum_{j=1}^N |a_{ij}|^2} = \sqrt{\text{tr}(A^\top A)}.$$

The Fröbenius norm is the norm induced by the trace therefore, when $N = M$, we can work with the scalar product given by $\langle A, B \rangle = \text{tr}(A^\top B)$. Let us observe that, in $\mathbb{R}^{N \times N}$,

1. $\langle A, B \rangle_{\mathbb{R}^{N \times N}} = \text{tr}(A^\top B)$
2. $\langle A, \text{id}_N \rangle_{\mathbb{R}^{N \times N}} = \text{tr}(A) = \text{tr}(A^\top)$
3. $\langle \text{id}_N, \text{id}_N \rangle_{\mathbb{R}^{N \times N}} = \|\text{id}_N\|_F^2 = N$.

Given a linear subspace $\mathcal{U} \subset \mathbb{R}^{N \times N}$ we will denote:

(a) the orthogonal complement of \mathcal{U} in $\mathbb{R}^{N \times N}$ by

$$\mathcal{U}^\perp = \{V \in \mathbb{R}^{N \times N} : \langle U, V \rangle_{\mathbb{R}^{N \times N}} = 0 \text{ for all } U \in \mathcal{U}\},$$

and,

(b) the orthogonal projection of $\mathbb{R}^{N \times N}$ on \mathcal{U} as

$$P_{\mathcal{U}}(V) := \arg \min_{U \in \mathcal{U}} \|U - V\|_F,$$

and hence

$$P_{\mathcal{U}^\perp} = \text{id}_N - P_{\mathcal{U}}.$$

Before defining a Laplacian-like matrix, we recall 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

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

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. $(A \otimes B)^\top = A^\top \otimes B^\top.$
6. $\text{tr}(A \otimes B) = \text{tr}(A)\text{tr}(B).$

From the example given in the introduction, we observe that there is a particular type of matrices to solve high-dimensional linear systems for which the GROU algorithm works particularly well: very fast convergence and also a very good approximation of the solution. These are the so-called Laplacian-Like matrices that we define below.

Definition 2.2. Given a matrix $A \in \mathbb{R}^{N \times N}$, where $N = n_1 \cdots n_d$, we say that A is a Laplacian-like matrix if there exist matrices $A_i \in \mathbb{R}^{n_i \times n_i}$ for $1 \leq i \leq d$ be such that

$$A = \sum_{i=1}^d A_i \otimes \text{id}_{[n_i]} \doteq \sum_{i=1}^d \text{id}_{n_1} \otimes \dots \otimes \text{id}_{n_{i-1}} \otimes A_i \otimes \text{id}_{n_{i+1}} \otimes \dots \otimes \text{id}_{n_d}, \tag{2}$$

where id_{n_j} is the identity matrix of size $n_j \times n_j$.

Observe that for $1 < i < d$,

$$\text{id}_{n_1} \otimes \dots \otimes \text{id}_{n_{i-1}} = \text{id}_{n_1 \dots n_{i-1}} \quad \text{and} \quad \text{id}_{n_{i+1}} \otimes \dots \otimes \text{id}_{n_d} = \text{id}_{n_{i+1} \dots n_d},$$

hence

$$A_i \otimes \text{id}_{[n_i]} = \text{id}_{n_1 \dots n_{i-1}} \otimes A_i \otimes \text{id}_{n_{i+1} \dots n_d}.$$

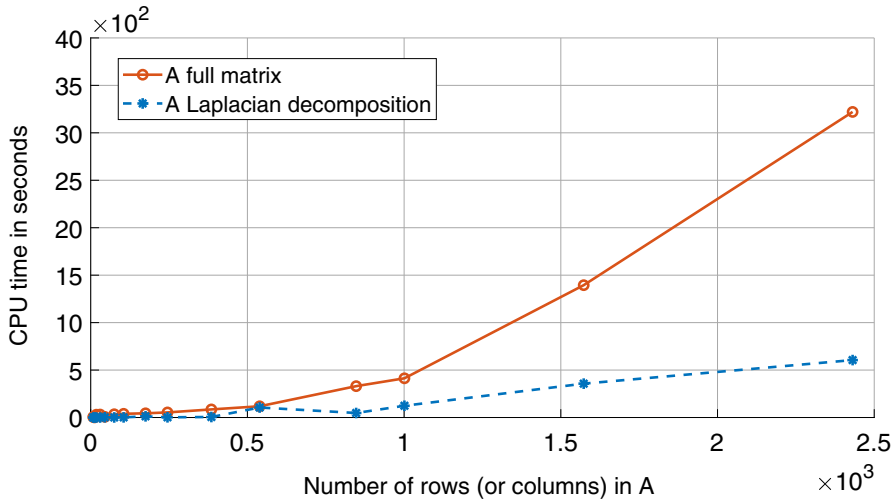


FIGURE 2. CPU time comparative to solve an $Ax = \mathbf{b}$ problem. This graph has been generated by using the following data in Algorithm 1: $\text{tol} = 2.22e - 6$; $\varepsilon = 1.0e - 06$; $\text{rank_max} = 3000$; (an $\text{iter_max} = 15$ was used to perform an ALS strategy); and the matrices have been randomly generated for each different size, in Laplacian and classical form. The characteristics of the computer used here are the same as in the case of Fig. 1

Moreover,

$$A_1 \otimes \text{id}_{[n_1]} = A_1 \otimes \text{id}_{n_2 \dots n_d} \quad \text{and} \quad A_d \otimes \text{id}_{[n_d]} = \text{id}_{n_1 \dots n_{d-1}} \otimes A_d.$$

It is not difficult to see that the set of Laplacian-like matrices is a linear subspace of $\mathbb{R}^{N \times N}$. From now on, we will denote by $\mathcal{L}(\mathbb{R}^{N \times N})$ the subspace of Laplacian-like matrices in $\mathbb{R}^{N \times N}$ for a fixed decomposition of $N = n_1 \cdots n_d$.

These matrices can be easily related to the classical Laplacian operator [17, 18] by writing:

$$\frac{\partial^2}{\partial x_i^2} = \frac{\partial^0}{\partial x_1^0} \otimes \cdots \otimes \frac{\partial^0}{\partial x_{i-1}^0} \otimes \frac{\partial^2}{\partial x_i^2} \otimes \frac{\partial^0}{\partial x_{i+1}^0} \otimes \cdots \otimes \frac{\partial^0}{\partial x_d^0}$$

and where $\frac{\partial^0}{\partial x_j^0}$ is the identity operator for functions in the variable x_j for $j \neq i$.

As the next numerical example shows, matrices written as in (2) provide very good performance of the GROU algorithm. In Fig. 2 we give a comparison of the speed of convergence to solve a linear system $Ax = \mathbf{b}$, where for each fixed size, we randomly generated two full-rank matrices: one given in the classical form and a Laplacian-like matrix. Both systems were solved following Algorithm 1.

The above results, together with the previous Poisson example given in the introduction, motivate the interest to know for a given matrix $A \in \mathbb{R}^{N \times N}$ how far it is from the linear subspace of Laplacian-like matrices. More precisely, we are interested in decomposing any matrix A as a sum of two orthogonal matrices L and L^\perp , where L is in $\mathcal{L}(\mathbb{R})$ and L^\perp in $\mathcal{L}(\mathbb{R})^\perp$. Clearly, if we obtain that $L^\perp = 0$, that is, $A \in \mathcal{L}(\mathbb{R})$, then we can solve any associated linear system by means of the GROU algorithm.

Recall that the set of matrices $\mathbb{R}^{N \times N}$ is a Lie Algebra that appears as the tangent space at the identity matrix of the linear general group $GL(\mathbb{R}^N)$, a Lie group composed by the non-singular matrices of $\mathbb{R}^{N \times N}$ (see [19]). Furthermore, the exponential map

$$\exp : \mathbb{R}^{N \times N} \longrightarrow GL(\mathbb{R}^N), \quad A \mapsto \exp(A) = \sum_{n=0}^{\infty} \frac{A^n}{n!}$$

is well-defined, however it is not surjective because $\det(\exp(A)) = e^{\text{tr}(A)} > 0$. Any linear subspace $\mathfrak{S} \subset \mathbb{R}^{N \times N}$ is a Lie-subalgebra if for all $A, B \in \mathfrak{S}$ its Lie crochet is also in \mathfrak{S} , that is, $[A, B] = AB - BA \in \mathfrak{S}$.

The linear space $\mathcal{L}(\mathbb{R}^{N \times N})$ is more than a linear subspace of $\mathbb{R}^{N \times N}$, it is also a Lie sub-algebra of $\mathbb{R}^{N \times N}$ as the next result shows.

Proposition 2.1. *Assume $\mathbb{R}^{N \times N}$, where where $N = n_1 \cdots n_d$. Then the following statements hold.*

- (a) *The linear subspace $\mathcal{L}(\mathbb{R}^{N \times N})$ is a Lie subalgebra of the matrix Lie algebra $\mathbb{R}^{N \times N}$.*
- (b) *The matrix group*

$$\mathfrak{L}(\mathbb{R}^{N \times N}) = \left\{ \bigotimes_{i=1}^d A_i : A_i \in GL(\mathbb{R}^{n_i}) \text{ for } 1 \leq i \leq d \right\}$$

is a Lie subgroup of $GL(\mathbb{R}^N)$.

- (c) *The exponential map $\exp : \mathcal{L}(\mathbb{R}^{N \times N}) \longrightarrow \mathfrak{L}(\mathbb{R}^{N \times N})$ is well defined and it is given by*

$$\exp \left(\sum_{i=1}^d A_i \otimes \text{id}_{n_{[i]}} \right) = \bigotimes_{i=1}^d \exp(A_i).$$

Proof. (a) To prove the first statement, take $A, B \in \mathcal{L}(\mathbb{R}^{N \times N})$. Then there exist matrices $A_i, B_i \in \mathbb{R}^{n_i \times n_i}$ for $1 \leq i \leq d$ be such that

$$A = \sum_{i=1}^d A_i \otimes \text{id}_{[n_i]} \quad \text{and} \quad B = \sum_{j=1}^d B_j \otimes \text{id}_{[n_j]}.$$

Observe, that for $i < j$

$$(A_i \otimes \text{id}_{[n_i]})(B_j \otimes \text{id}_{[n_j]}) \quad \text{and} \quad (B_j \otimes \text{id}_{[n_j]})(A_i \otimes \text{id}_{[n_i]})$$

both products are equal to

$$\text{id}_{n_1 \dots n_{i-1}} \otimes A_i \otimes \text{id}_{n_{i+1} \dots n_{j-1}} \otimes B_j \otimes \text{id}_{n_{j+1} \dots n_d}.$$

A similar expression is obtained for $i > j$. Thus,

$$[A_i \otimes \text{id}_{[n_i]}, B_j \otimes \text{id}_{[n_j]}] = 0$$

for all $i \neq j$.

On the other hand, for $i = j$ we have

$$(A_i \otimes \text{id}_{[n_i]})(B_i \otimes \text{id}_{[n_i]}) = (A_i B_i \otimes \text{id}_{[n_i]})$$

and

$$(B_i \otimes \text{id}_{[n_i]})(A_i \otimes \text{id}_{[n_i]}) = (B_i A_i \otimes \text{id}_{[n_i]}).$$

Thus,

$$[A_i \otimes \text{id}_{[n_i]}, B_i \otimes \text{id}_{[n_i]}] = (A_i B_i - B_i A_i) \otimes \text{id}_{[n_i]},$$

that is,

$$[A_i \otimes \text{id}_{[n_i]}, B_i \otimes \text{id}_{[n_i]}] = [A_i, B_i] \otimes \text{id}_{[n_i]}.$$

Here $[A_i, B_i]$ is the Lie crochet in $\mathbb{R}^{n_i \times n_i}$.

In consequence, from all said above, we conclude

$$[A, B] = \sum_{i=1}^d \sum_{j=1}^d [A_i \otimes \text{id}_{[n_i]}, B_j \otimes \text{id}_{[n_j]}] = \sum_{i=1}^d [A_i, B_i] \otimes \text{id}_{[n_i]} \in \mathcal{L}(\mathbb{R}^{N \times N}).$$

This proves that $\mathcal{L}(\mathbb{R}^{N \times N})$ is a Lie sub-algebra of $\mathbb{R}^{N \times N}$.

(b) It is not difficult to see that $\mathfrak{L}(\mathbb{R}^{N \times N})$ is a subgroup of $\text{GL}(\mathbb{R}^N)$. From Theorem 19.18 in [19], to prove that $\mathfrak{L}(\mathbb{R}^{N \times N})$ is a Lie subgroup of $\text{GL}(\mathbb{R}^N)$ we only need to show that $\mathfrak{L}(\mathbb{R}^{N \times N})$ is a closed set in $\text{GL}(\mathbb{R}^N)$. This follows from the fact that the map

$$\Phi : \text{GL}(\mathbb{R}^{n_1}) \times \dots \times \text{GL}(\mathbb{R}^{n_d}) \longrightarrow \text{GL}(\mathbb{R}^N) \quad (A_1, \dots, A_d) \mapsto \bigotimes_{i=1}^d A_i$$

is continuous. Assume that there exists a sequence, $\{A_n\}_{n \in \mathbb{N}} \subset \mathfrak{L}(\mathbb{R}^{N \times N})$ convergent to $A \in \text{GL}(\mathbb{R}^N)$. Then the sequence $\{A_n\}_{n \in \mathbb{N}}$ is bounded. Since there exists a sequence $\{(A_1^{(n)}, \dots, A_d^{(n)})\}_{n \in \mathbb{N}} \subset \text{GL}(\mathbb{R}^{n_1}) \times \dots \times \text{GL}(\mathbb{R}^{n_d})$ such that $A_n = \bigotimes_{j=1}^d A_j^{(n)}$, the sequence $\{(A_1^{(n)}, \dots, A_d^{(n)})\}_{n \in \mathbb{N}}$ is also bounded. Thus, there exists a convergent sub-sequence, also denoted by $\{(A_1^{(n)}, \dots, A_d^{(n)})\}_{n \in \mathbb{N}}$, to $(A_1, \dots, A_d) \in \text{GL}(\mathbb{R}^{n_1}) \times \dots \times \text{GL}(\mathbb{R}^{n_d})$. The continuity of Φ , implies that $A = \bigotimes_{i=1}^d A_i$. Thus $\mathfrak{L}(\mathbb{R}^{N \times N})$ is closed in $\text{GL}(\mathbb{R}^N)$, and hence a Lie subgroup.

(c) From Lemma 4.169(b) [8], the following equality

$$\exp \left(\sum_{i=1}^d \text{id}_{n_1} \otimes \dots \otimes \text{id}_{n_{i-1}} \otimes A_i \otimes \text{id}_{n_{i+1}} \otimes \dots \otimes \text{id}_{n_d} \right) = \bigotimes_{i=1}^d \exp(A_i)$$

holds. Thus, the exponential map is well defined. This ends the proof of the proposition. \square

We conclude this section describing in a more detail the structure of matrices $A \in \mathbb{R}^{N \times N}$ for which there exists $A_i \in \mathbb{R}^{n_i \times n_i}$ for $1 \leq i \leq d$ such that

$$A = \sum_{i=1}^d \text{id}_{n_1 \dots n_{i-1}} \otimes A_i \otimes \text{id}_{n_{i+1} \dots n_d}.$$

For dealing easily with Laplacian-like matrices, we introduce the following notation. For each $1 < i \leq d$ consider the integer number $n_1 n_2 \dots n_{i-1}$. Then, we will denote by

$$\begin{pmatrix} \star & \star & \cdots & \star \\ \star & \star & \cdots & \star \\ \vdots & \vdots & \ddots & \vdots \\ \star & \star & \cdots & \star \end{pmatrix}_{n_1 n_2 \dots n_{i-1} \times n_1 n_2 \dots n_{i-1}}$$

a block square matrix composed by $n_1 n_2 \dots n_{i-1} \times n_1 n_2 \dots n_{i-1}$ -blocks.

Since $A_i \otimes \text{id}_{n_{i+1} \dots n_d}$

$$= \begin{pmatrix} (A_i)_{1,1} \text{id}_{n_{i+1} \dots n_d} & (A_i)_{1,2} \text{id}_{n_{i+1} \dots n_d} & \cdots & (A_i)_{1,n_i} \text{id}_{n_{i+1} \dots n_d} \\ (A_i)_{2,1} \text{id}_{n_{i+1} \dots n_d} & (A_i)_{2,2} \text{id}_{n_{i+1} \dots n_d} & \cdots & (A_i)_{2,n_i} \text{id}_{n_{i+1} \dots n_d} \\ \vdots & \vdots & \ddots & \vdots \\ (A_i)_{n_i,1} \text{id}_{n_{i+1} \dots n_d} & (A_i)_{n_i,2} \text{id}_{n_{i+1} \dots n_d} & \cdots & (A_i)_{n_i,n_i} \text{id}_{n_{i+1} \dots n_d} \end{pmatrix},$$

then $\text{id}_{n_1 \dots n_{i-1}} \otimes A_i \otimes \text{id}_{n_{i+1} \dots n_d}$

$$= \begin{pmatrix} A_i \otimes \text{id}_{n_{i+1} \dots n_d} & O_i \otimes \text{id}_{n_{i+1} \dots n_d} & \cdots & O_i \otimes \text{id}_{n_{i+1} \dots n_d} \\ O_i \otimes \text{id}_{n_{i+1} \dots n_d} & A_i \otimes \text{id}_{n_{i+1} \dots n_d} & \cdots & O_i \otimes \text{id}_{n_{i+1} \dots n_d} \\ \vdots & \vdots & \ddots & \vdots \\ O_i \otimes \text{id}_{n_{i+1} \dots n_d} & O_i \otimes \text{id}_{n_{i+1} \dots n_d} & \cdots & A_i \otimes \text{id}_{n_{i+1} \dots n_d} \end{pmatrix}_{n_1 n_2 \dots n_{i-1} \times n_1 n_2 \dots n_{i-1}},$$

where O_i denotes the zero matrix in $\mathbb{R}^{n_i \times n_i}$ for $1 \leq i \leq d$. To conclude, we have the following cases

$$A_1 \otimes \text{id}_{n_2 \dots n_d} = \begin{pmatrix} (A_1)_{1,1} \text{id}_{n_2 \dots n_d} & (A_1)_{1,2} \text{id}_{n_2 \dots n_d} & \cdots & (A_1)_{1,n_1} \text{id}_{n_2 \dots n_d} \\ (A_1)_{2,1} \text{id}_{n_2 \dots n_d} & (A_1)_{2,2} \text{id}_{n_2 \dots n_d} & \cdots & (A_1)_{2,n_1} \text{id}_{n_2 \dots n_d} \\ \vdots & \vdots & \ddots & \vdots \\ (A_1)_{n_1,1} \text{id}_{n_2 \dots n_d} & (A_1)_{n_1,2} \text{id}_{n_2 \dots n_d} & \cdots & (A_1)_{n_1,n_1} \text{id}_{n_2 \dots n_d} \end{pmatrix}$$

and

$$\text{id}_{n_1 \dots n_{d-1}} \otimes A_d = \begin{pmatrix} A_d & O_d & \cdots & O_d \\ O_d & A_d & \cdots & O_d \\ \vdots & \vdots & \ddots & \vdots \\ O_d & O_d & \cdots & A_d \end{pmatrix}_{n_1 n_2 \dots n_{d-1} \times n_1 n_2 \dots n_{d-1}}.$$

We wish to point out that the above operations are widely used in quantum computing.

3. A Decomposition of the Linear Space of Laplacian-Like Matrices

We start by introducing some definitions and preliminary results needed to give an interesting decomposition of the linear space of Laplacian-like matrices. The next lemma lets us show how is the decomposition of $\mathbb{R}^{N \times N}$ as a direct sum of $\text{span}\{\text{id}_N\}$ and its orthogonal space $\text{span}\{\text{id}_N\}^\perp$, with respect the inner product $\langle A, B \rangle_{\mathbb{R}^{N \times N}} = \text{tr}(A^T B)$. From now on, we will denote by

$$\mathfrak{h}_n := \text{span}\{\text{id}_n\}^\perp = \{A \in \mathbb{R}^{n \times n} : \text{tr}(A) = 0\},$$

the linear subspace of null trace matrices in $\mathbb{R}^{n \times n}$.

Lemma 3.1. *Consider $(\mathbb{R}^{n \times n}, \|\cdot\|_F)$ as a Hilbert space. Then there exists a decomposition*

$$\mathbb{R}^{n \times n} = \text{span}\{\text{id}_n\} \oplus \mathfrak{h}_n,$$

Moreover, the orthogonal projection from $\mathbb{R}^{n \times n}$ on $\text{span}\{\text{id}_n\}$ is given by

$$P_{\text{span}\{\text{id}_n\}}(A) = \frac{\text{tr}(A)}{n} \text{id}_n,$$

and hence for each $A \in \mathbb{R}^{n \times n}$ we have the following decomposition,

$$A = \frac{\text{tr}(A)}{n} \text{id}_n + \left(A - \frac{\text{tr}(A)}{n} \text{id}_n \right),$$

where $\left(A - \frac{\text{tr}(A)}{N} \text{id}_N \right) \in \mathfrak{h}_n$.

Proof. The lemma follows from the fact that

$$P_{\text{span}\{\text{id}_n\}}(A) = \frac{\langle \text{id}_n, A \rangle_{\mathbb{R}^{n \times n}}}{\|\text{id}_n\|_F^2} \text{id}_n = \frac{\text{tr}(A)}{n} \text{id}_n,$$

is the orthogonal projection onto $\text{span}\{\text{id}_n\}$. □

Now, we consider the matrix space $\mathbb{R}^{N \times N}$ where $N = n_1 \cdots n_d$, and hence $\mathbb{R}^{N \times N} = \bigotimes_{i=1}^d \mathbb{R}^{n_i \times n_i}$ can be considered as a tensor space. A norm $\|\cdot\|$ defined over $\mathbb{R}^{N \times N}$ is called a tensor norm if and only if there exists a norm $\|\cdot\|_i$ over $\mathbb{R}^{n_i \times n_i}$ for $1 \leq i \leq d$, such that for any tensor $A = A_1 \otimes \cdots \otimes A_d \in \mathbb{R}^{N \times N}$, where $A_i \in \mathbb{R}^{n_i \times n_i}$ ($1 \leq i \leq d$), it holds

$$\|A\| = \|A_1 \otimes \cdots \otimes A_d\| = \prod_{i=1}^d \|A_i\|_i.$$

We remark that for tensors $A = A_1 \otimes \cdots \otimes A_d$ and $B = B_1 \otimes \cdots \otimes B_d$ where $A_i, B_i \in \mathbb{R}^{n_i \times n_i}$, we have

$$\begin{aligned} \langle A, B \rangle_{\mathbb{R}^{N \times N}} &= \langle A_1 \otimes \cdots \otimes A_d, B_1 \otimes \cdots \otimes B_d \rangle_{\mathbb{R}^{N \times N}} \\ &= \text{tr}((A_1 \otimes \cdots \otimes A_d)^T (B_1 \otimes \cdots \otimes B_d)) \\ &= \text{tr}((A_1^T \otimes \cdots \otimes A_d^T)(B_1 \otimes \cdots \otimes B_d)) \\ &= \text{tr}(A_1^T B_1 \otimes \cdots \otimes A_d^T B_d) \\ &= \prod_{i=1}^d \text{tr}(A_i^T B_i) = \prod_{i=1}^d \langle A_i, B_i \rangle_{\mathbb{R}^{n_i \times n_i}}, \end{aligned}$$

and hence $\|A\|_F = \sqrt{\langle A, A \rangle_{\mathbb{R}^{N \times N}}}$, is a tensor-norm. In particular, the inner product $\langle \cdot, \cdot \rangle_{\mathbb{R}^{N \times N}}$ satisfies

$$\langle \text{id}_{[n_i]} \otimes A_i, \text{id}_{[n_i]} \otimes B_i \rangle_{\mathbb{R}^{N \times N}} = \text{tr}(A_i^T B_i) \prod_{\substack{j=1 \\ j \neq i}}^d n_j, \tag{3}$$

The next result gives a first characterization of the linear space $\mathcal{L}(\mathbb{R}^{N \times N})$.

Theorem 3.1. *Let $\mathbb{R}^{N \times N}$, where $N = n_1 \cdots n_d$. Then*

$$\mathcal{L}(\mathbb{R}^{N \times N}) = \text{span}\{\text{id}_N\} \oplus \Delta, \tag{4}$$

where $\Delta = \mathfrak{h}_N \cap \mathcal{L}(\mathbb{R}^{N \times N})$. Furthermore, $\mathcal{L}(\mathbb{R}^{N \times N})^\perp$ is a subspace of \mathfrak{h}_N .

Proof. Assume that a given matrix $A \in \mathbb{R}^{N \times N}$ can be written as in (2) and denote each component in the sum representation of A by $L_i = \text{id}_{[n_i]} \otimes A_i$, where $A_i \in \mathbb{R}^{n_i \times n_i}$ for $1 \leq i \leq d$. Then $L_i \in \text{span}\{\text{id}_{[n_i]}\} \otimes \mathbb{R}^{n_i \times n_i}$ for $1 \leq i \leq d$, and in consequence,

$$\sum_{i=1}^d \text{span}\{\text{id}_{[n_i]}\} \otimes \mathbb{R}^{n_i \times n_i} = \mathcal{L}(\mathbb{R}^{N \times N}).$$

Thus, $\text{span}\{\text{id}_N\} \subset \mathcal{L}(\mathbb{R}^{N \times N})$, and, by Lemma 3.1, we have the following decomposition

$$\mathcal{L}(\mathbb{R}^{N \times N}) = \Delta \oplus \text{span}\{\text{id}_N\}. \tag{5}$$

where $\Delta = \mathfrak{h}_N \cap \mathcal{L}(\mathbb{R}^{N \times N})$. The last statement is consequence of Lemma 3.1. This ends the theorem. \square

Now, given any square matrix in $\mathbb{R}^{N \times N}$, we would like to project it onto $\mathcal{L}(\mathbb{R}^{N \times N})$ to obtain its Laplacian approximation. To compute this approximation explicitly, the following result, which is a consequence of the above theorem, will be useful.

Corollary 3.1. *Assume $\mathbb{R}^{N \times N}$, with $N = n_1 \cdots n_d \in \mathbb{N}$. Then*

$$P_{\mathcal{L}(\mathbb{R}^{N \times N})} = P_{\text{span}\{\text{id}_N\}} + P_\Delta,$$

that is, for all $A \in \mathbb{R}^{N \times N}$ it holds

$$P_{\mathcal{L}(\mathbb{R}^{N \times N})}(A) = \frac{\text{tr}(A)}{N} \text{id}_N + P_\Delta(A).$$

Next, we need to characterize Δ in order to explicitly construct the orthogonal projection onto $\mathcal{L}(\mathbb{R}^{N \times N})$. From the proof of the Theorem 3.1 we see that the linear subspaces given by

$$\mathfrak{h}_N \cap (\text{span}\{\text{id}_{[n_i]}\} \otimes \mathbb{R}^{n_i \times n_i}),$$

for $1 \leq i \leq d$, are of interest to characterize Δ as the next result shows.

Theorem 3.2. *Let $\mathbb{R}^{N \times N}$ with $N = n_1 \cdots n_d \in \mathbb{N}$, and let \mathfrak{H}_i be the orthogonal complement of $\text{span}\{\text{id}_N\}$ in the linear subspace $\mathbb{R}^{n_i \times n_i} \otimes \text{span}\{\text{id}_{[n_i]}\}$ for $1 \leq i \leq d$. Then,*

$$\Delta = \bigoplus_{i=1}^d \mathfrak{H}_i. \tag{6}$$

Furthermore, a matrix A belongs to Δ if and only if it has the form

$$A = \sum_{i=1}^d A_i \otimes \text{id}_{[n_i]}, \quad \text{with } \text{tr}(A_i) = 0, \quad i = 1, \dots, d.$$

Proof. First, we take into account that $\mathbb{R}^{n_i \times n_i} \otimes \text{span}\{\text{id}_{[n_i]}\}$ a linear subspace of $\mathcal{L}(\mathbb{R}^{N \times N})$ linearly isomorphic to the matrix space $\mathbb{R}^{n_i \times n_i}$. Thus, motivated by Lemma 3.1 applied on $\mathbb{R}^{n_i \times n_i}$, we write

$$\begin{aligned} \mathbb{R}^{n_i \times n_i} \otimes \text{span}\{\text{id}_{[n_i]}\} &= (\text{span}\{\text{id}_{n_i}\} \oplus \mathfrak{h}_{n_i}) \otimes \text{span}\{\text{id}_{[n_i]}\} \\ &= (\text{span}\{\text{id}_{n_i}\} \otimes \text{span}\{\text{id}_{[n_i]}\}) \oplus (\mathfrak{h}_{n_i} \otimes \text{span}\{\text{id}_{[n_i]}\}). \end{aligned}$$

Now, we claim that $\mathfrak{h}_{n_i} \otimes \text{span}\{\text{id}_{[n_i]}\}$ it is the orthogonal complement of the linear subspace generated by the identity matrix $\text{id}_N = \text{id}_{n_i} \otimes \text{id}_{[n_i]}$ in the linear subspace $\mathbb{R}^{n_i \times n_i} \otimes \text{id}_{[n_i]}$. To prove the claim, observe that for $A_i \otimes \text{id}_{[n_i]} \in \mathfrak{h}_{n_i} \otimes \text{span}\{\text{id}_{[n_i]}\}$ ($1 \leq i \leq d$), by using (3), it holds

$$\langle A_i \otimes \text{id}_{[n_i]}, \text{id}_N \rangle_{\mathbb{R}^{N \times N}} = \langle A_i \otimes \text{id}_{[n_i]}, \text{id}_{[n_i]} \otimes \text{id}_{n_i} \rangle_{\mathbb{R}^{N \times N}} = \text{tr}(A_i) \prod_{\substack{j=1 \\ j \neq i}}^d n_j = 0,$$

because $A_i \in \mathfrak{h}_{n_i}$ and hence $\text{tr}(A_i) = 0$, for $1 \leq i \leq d$. Thus, the claim follows and hence

$$\begin{aligned} \mathfrak{H}_i &= \mathfrak{h}_{n_i} \otimes \text{span}\{\text{id}_{[n_i]}\} \\ &= \{A_i \otimes \text{id}_{[n_i]} \in \mathbb{R}^{n_i \times n_i} \otimes \text{span}\{\text{id}_{[n_i]}\} : \text{tr}(A_i) = 0\} \\ &= \{A_i \otimes \text{id}_{[n_i]} \in \mathbb{R}^{n_i \times n_i} \otimes \text{span}\{\text{id}_{[n_i]}\} : \langle A_i \otimes \text{id}_{[n_i]}, \text{id}_N \rangle_{\mathbb{R}^{N \times N}} = 0\}. \end{aligned}$$

To prove (6), we first consider $1 \leq i < j \leq d$, and take $A_k \otimes \text{id}_{[n_k]} \in \mathfrak{H}_k$ for $k = i, j$. Then the inner product satisfies

$$\begin{aligned} \langle A_i \otimes \text{id}_{[n_i]}, A_j \otimes \text{id}_{[n_j]} \rangle_{\mathbb{R}^{N \times N}} &= \text{tr} \left((A_i \otimes \text{id}_{[n_i]})^T (A_j \otimes \text{id}_{[n_j]}) \right) \\ &= \text{tr} \left((A_i^T \otimes \text{id}_{[n_i]}) (A_j \otimes \text{id}_{[n_j]}) \right) \\ &= \text{tr} \left(\text{id}_{n_1 \dots n_{i-1}} \otimes A_i^T \otimes \text{id}_{n_{i+1} \dots n_{j-1}} \otimes A_j \otimes \text{id}_{n_{j+1} \dots n_d} \right) \\ &= \prod_{\substack{\ell=1 \\ \ell \neq i, j}}^d \langle \text{id}_\ell, \text{id}_\ell \rangle_{\mathbb{R}^{n_\ell \times n_\ell}} \text{tr}(A_i) \text{tr}(A_j) = 0, \end{aligned}$$

because $\text{tr}(A_i) = \text{tr}(A_j) = 0$. The same equality holds for $j < i$. Thus, we conclude that \mathfrak{H}_i is orthogonal to \mathfrak{H}_j for all $i \neq j$. So, the subspace

$$\Delta' = \bigoplus_{i=1}^d \mathfrak{H}_i,$$

is well defined and it is a subspace of $\mathcal{L}(\mathbb{R}^{N \times N})$.

To conclude the proof (6), we will show that $\Delta' = \Delta$. Since, for each $1 \leq i \leq d$, \mathfrak{h}_i is orthogonal to $\text{span}\{\text{id}_N\}$ we have

$$\text{span}\{\text{id}_N\} \oplus \Delta' \subset \mathcal{L}(\mathbb{R}^{N \times N}).$$

To obtain the equality, take $A \in \mathcal{L}(\mathbb{R}^{N \times N})$. Then there exists $A_i \in \mathbb{R}^{n_i \times n_i}$ for $1 \leq i \leq d$ be such that

$$A = \sum_{i=1}^d A_i \otimes \text{id}_{[n_i]}.$$

From Lemma 3.1 we can write

$$A_i = \frac{\text{tr}(A_i)}{n_i} \text{id}_{n_i} + \left(A_i - \frac{\text{tr}(A_i)}{n_i} \text{id}_{n_i} \right)$$

for each $1 \leq i \leq d$. Then,

$$\begin{aligned} A &= \sum_{i=1}^d \left(\frac{\text{tr}(A_i)}{n_i} \text{id}_{n_i} + \left(A_i - \frac{\text{tr}(A_i)}{n_i} \text{id}_{n_i} \right) \right) \otimes \text{id}_{[n_i]} \\ &= \sum_{i=1}^d \frac{\text{tr}(A_i)}{n_i} \text{id}_{n_i} \otimes \text{id}_{[n_i]} + \sum_{i=1}^d \left(A_i - \frac{\text{tr}(A_i)}{n_i} \text{id}_{n_i} \right) \otimes \text{id}_{[n_i]} \\ &= \sum_{i=1}^d \frac{\text{tr}(A_i)}{n_i} (\text{id}_{n_i} \otimes \text{id}_{[n_i]}) + \sum_{i=1}^d \left(A_i - \frac{\text{tr}(A_i)}{n_i} \text{id}_{n_i} \right) \otimes \text{id}_{[n_i]} \\ &= \left(\sum_{i=1}^d \frac{\text{tr}(A_i)}{n_i} \right) \text{id}_N + \sum_{i=1}^d \left(A_i - \frac{\text{tr}(A_i)}{n_i} \text{id}_{n_i} \right) \otimes \text{id}_{[n_i]}. \end{aligned}$$

Observe that $\left(\sum_{i=1}^d \frac{\text{tr}(A_i)}{n_i}\right) \text{id}_N \in \text{span}\{\text{id}_N\}$ and

$$\sum_{i=1}^d \left(A_i - \frac{\text{tr}(A_i)}{n_i} \text{id}_{n_i}\right) \otimes \text{id}_{[n_i]} \in \Delta'.$$

Thus, $\mathcal{L}(\mathbb{R}^{N \times N}) \subset \text{span}\{\text{id}_N\} \oplus \Delta'$. In consequence $\Delta' = \Delta$, and this proves the theorem. \square

A direct consequence of the above theorem is the next corollary.

Corollary 3.2. *Assume $\mathbb{R}^{N \times N}$, with $N = n_1 \cdots n_d \in \mathbb{N}$. Then*

$$P_{\mathcal{L}(\mathbb{R}^{N \times N})} = P_{\text{span}\{\text{id}_N\}} + \sum_{i=1}^d P_{\mathfrak{h}_i},$$

that is, for all $A \in \mathbb{R}^{N \times N}$ it holds

$$P_{\mathcal{L}(\mathbb{R}^{N \times N})}(A) = \frac{\text{tr}(A)}{N} \text{id}_N + \sum_{i=1}^d A_i \otimes \text{id}_{[n_i]},$$

where $A_i \in \mathbb{R}^{n_i \times n_i}$ satisfies $\text{tr}(A_i) = 0$ for $1 \leq i \leq d$.

4. A Numerical Strategy to Perform a Laplacian-Like Decomposition

Now, in this section we will study some numerical strategies in order to compute, for a given matrix $A \in \mathbb{R}^{N \times N}$, with the help of Proposition 3.1 and Theorem 3.2, its best Laplacian-like approximation. Then, we will present two numerical examples to give consistency to the results obtained. In the first example, we will work with the adjacency matrix of a simple graph of 6 nodes; with it, we intend to show, step by step, the procedure to follow to calculate the Laplacian decomposition of a square matrix. The second example will complete the study of the discrete Poisson equation described in Sect. 1; with this example, we will illustrate the goodness of the Laplacian decomposition to solve PDEs in conjunction with the GROU Algorithm. We start with the following Greedy Algorithm.

Theorem 4.1. *Let A be a matrix in $\mathbb{R}^{N \times N}$, with $N = n_1 \cdots n_d$, such that $\text{tr}(A) = 0$. Consider the following iterative procedure:*

1. Take $X_k^{(0)} = 0$ for $1 \leq k \leq d$.
2. For each $\ell \geq 1$ compute for $1 \leq i \leq d$ the matrix $U_i^{(\ell)}$ as

$$U_i^{(\ell)} = \arg \min_{U_i \in \mathfrak{h}_i} \left\| A - \sum_{k=1}^{i-1} X_k^{(\ell)} \otimes \text{id}_{[n_k]} - \xi(U_i) \otimes \text{id}_{[n_i]} - \sum_{k=i+1}^d X_k^{(\ell-1)} \otimes \text{id}_{[n_k]} \right\|,$$

where

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

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

Then

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

Proof. Recall that $P_{\Delta}(A)$ solves the problem

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

To simplify notation put $P_{\Delta}^{(\ell)}(A) = \sum_{k=1}^d X_k^{(\ell)} \otimes \text{id}_{[n_k]}$ for $\ell \geq 0$. By construction we have that

$$\|A - P_{\Delta}^{(1)}(A)\| \geq \|A - P_{\Delta}^{(2)}(A)\| \geq \dots \geq \|A - P_{\Delta}^{(\ell)}(A)\| \geq \dots \geq 0,$$

holds. Since the sequence $\{P_{\Delta}^{(\ell)}(A)\}_{\ell \in \mathbb{N}}$ is bounded, there is a convergent subsequence also denoted by $\{P_{\Delta}^{(\ell)}(A)\}_{\ell \in \mathbb{N}}$, so that

$$L_A = \lim_{\ell \rightarrow \infty} P_{\Delta}^{(\ell)}(A) \in \Delta.$$

If $L_A = P_{\Delta}(A)$, the theorem holds. Otherwise, assume that $L_A \neq P_{\Delta}(A)$ (see Fig. 3), then it is clear that

$$\|A - P_{\Delta}(A)\| \leq \|A - L_A\|.$$

Suppose that $\|A - P_{\Delta}(A)\| < \|A - L_A\|$ and let $\lambda \in (0, 1)$. Now, consider the linear combination $\lambda L_A + (1 - \lambda)P_{\Delta}(A)$. Since $L_A, P_{\Delta}(A) \in \Delta$, they can be written as

$$L_A = \sum_{i=1}^d A_i \otimes \text{id}_{[n_i]}, \quad \text{and} \quad P_{\Delta}(A) = \sum_{i=1}^d A_i^* \otimes \text{id}_{[n_i]}$$

so $\lambda L_A + (1 - \lambda)P_{\Delta}(A) = \sum_{i=1}^d \text{id}_{[n_i]} \otimes (\lambda A_i + (1 - \lambda)A_i^*) \in \Delta$. Hence,

$$\|A - P_{\Delta}(A)\| < \|A - (\lambda L_A + (1 - \lambda)P_{\Delta}(A))\| < \|A - L_A\|.$$

That is, we have found d matrices $Z_i = \lambda A_i + (1 - \lambda)A_i^*$, $i = 1, \dots, d$, such that

$$\|A - L_A\| = \left\| A - \sum_{i=1}^d A_i \otimes \text{id}_{[n_i]} \right\| > \left\| A - \sum_{i=1}^d Z_i \otimes \text{id}_{[n_i]} \right\|,$$

which is a contradiction with the definition of L_A . □

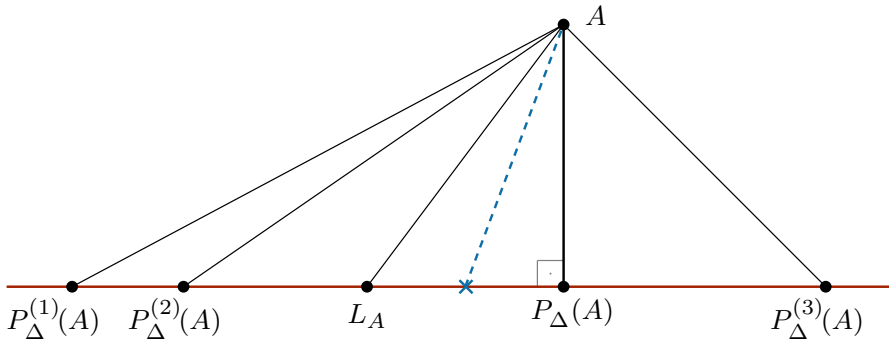


FIGURE 3. Situation described in reasoning by *Reductio ad absurdum*

The previous result allows us to describe the procedure to obtain the Laplacian approximation of a square matrix, in the form of an algorithm. We can visualize the complete algorithm in the form of pseudocode in Algorithm 2.

Algorithm 2 Laplacian Decomposition Algorithm

```

1: procedure LAP( $A^*$ , iter_max, tol)
2:    $A = A^* - (\text{tr}(A)/N)\text{id}_N$ , iter = 1, Lap = 0
3:   while iter < iter_max do
4:      $A \leftarrow A - \text{Lap}$ 
5:     for  $k = 1, 2, \dots, d$  do
6:        $P_k(A) = \text{id}_{n_1} \otimes \dots \otimes \text{id}_{n_{k-1}} \otimes X_k \otimes \text{id}_{n_{k+1}} \otimes \dots \otimes \text{id}_{n_d}$ 
7:        $X_k \leftarrow \min_{X_k} \|A - \sum_{i=1}^k P_i(A)\|$ 
8:       Lap = Lap +  $P_k(A)$ 
9:     end for
10:    if  $\|A - \text{Lap}\| < \text{tol}$  then goto 14
11:    end if
12:    iter = iter + 1
13:  end while
14:  return Lap
15: end procedure

```

4.1. Numerical Examples

4.1.1. The Adjacency Matrix of a Simple Graph. First, let us show an example in which the projection $P_{\Delta}(A)$ coincides with A and how the tensor representations is provided by the aforementioned proposed algorithm. Let us consider the simple graph $G(V, E)$, with $V = \{1, 2, \dots, 6\}$ the set of nodes and $E = \{(1, 2), (1, 4), (2, 3), (2, 5), (3, 6), (4, 5), (5, 6)\}$ the set of edges. Then, the

adjacency matrix of G is

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}.$$

We want to find a Laplacian decomposition of the matrix $A \in \mathbb{R}^{6 \times 6}$. Since $\text{tr}(A) = 0$, we can do this by following the iterative scheme given by Theorem 4.1. So, we look for $X_1 \in \mathbb{R}^{2 \times 2}$, $X_2 \in \mathbb{R}^{3 \times 3}$ matrices such that

$$P_\Delta(A) = X_1 \otimes \text{id}_{n_2} + \text{id}_{n_1} \otimes X_2,$$

where $n_1 = 2, n_2 = 3$. We proceed according to the algorithm:

1. Computing X_1 by

$$\min_{X_1} \|A - X_1 \otimes \text{id}_{n_2}\|,$$

we obtain

$$X_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

2. Computing X_2 by

$$\min_{X_2} \|A - X_1 \otimes \text{id}_{n_2} - \text{id}_{n_1} \otimes X_2\|,$$

we obtain

$$X_2 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

Since the residual value is $\|A - P_\Delta(A)\| = 0$, the matrix $A \in \Delta$. Thus, we can write it as

$$A = P_\Delta(A) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \text{id}_{n_2} + \text{id}_{n_1} \otimes \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

4.1.2. The Poisson’s Equation. Now, let us consider the Poisson’s equation (1) with homogeneous boundary condition given in Sect. 1. For each $u \in \{x, y, z\}$ we fix $h = \frac{1}{n}$, where $n \in \mathbb{N}$, and take $u_\ell = (\ell - 1)h$ for $1 \leq \ell \leq n$. Next, we consider a derivative approximation scheme given by

$$\frac{\partial^2 \psi}{\partial u^2} \approx \frac{\psi(u_{\ell+1}) - 2\psi(u_\ell) + \psi(u_{\ell-1}))}{h^2},$$

in (1) for $u \in \{x, y, z\}$. It allows to obtain a linear system written as

$$A\phi(x_i, y_j, z_k) = \mathbf{f}(x_i, y_j, z_k) \tag{7}$$

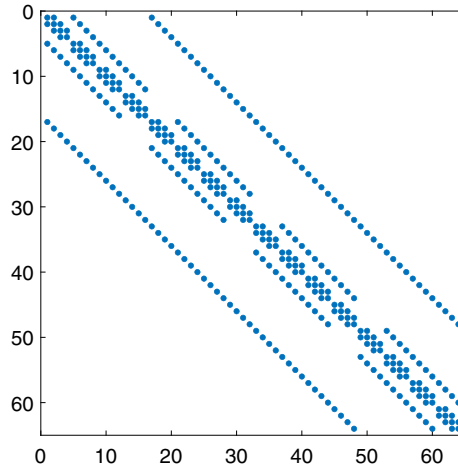


FIGURE 4. Matlab `spy(A)` representation of A given in (8) for $n = 4$

where the indices $1 \leq i, j, k \leq n$ correspond to the discretization of x, y and z respectively, and $A \in \text{GL}(\mathbb{R}^{n^3})$ is the matrix given by

$$A = \begin{pmatrix} T & -\text{id}_{n^2} & & & \\ -\text{id}_{n^2} & T & -\text{id}_{n^2} & & \\ & \ddots & \ddots & \ddots & \\ & & -\text{id}_{n^2} & T & -\text{id}_{n^2} \\ & & & -\text{id}_{n^2} & T \end{pmatrix}, \tag{8}$$

where $T \in \mathbb{R}^{n^2 \times n^2}$ is the matrix

$$T = \begin{pmatrix} D & -\text{id}_n & & & \\ -\text{id}_n & D & -\text{id}_n & & \\ & \ddots & \ddots & \ddots & \\ & & -\text{id}_n & D & -\text{id}_n \\ & & & -\text{id}_n & D \end{pmatrix}$$

with $D = \begin{pmatrix} 6 & -1 & 0 & \dots & 0 \\ -1 & 6 & -1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & -1 & 6 & -1 \\ 0 & \dots & 0 & -1 & 6 \end{pmatrix} \in \mathbb{R}^{n \times n}.$

We can visualize a representation of the sparsity of the matrix A with the `spy` command from Matlab (see Fig. 4). Since $\text{tr}(A) = 6n^3 \neq 0$, instead of looking for the Laplacian approximation of A , we will look for $\hat{A} = (A - 6 \text{id}_{n^3})$, which

has null trace. Proceeding according Algorithm 2 for sizes $n_1 = n_2 = n_3 = n$, we obtain the decomposition

$$\hat{A} = X \otimes \text{id}_n \otimes \text{id}_n + \text{id}_n \otimes X \otimes \text{id}_n + \text{id}_n \otimes \text{id}_n \otimes X,$$

where

$$X = \begin{pmatrix} 0 & -1 & 0 & \dots & 0 \\ -1 & 0 & -1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & -1 & 0 & -1 \\ 0 & \dots & 0 & -1 & 0 \end{pmatrix} \in \mathbb{R}^{n \times n},$$

and the residual of the approximation of \hat{A} is $\|\hat{A} - P_\Delta(\hat{A})\| = 0$. Thus, following Corollary 3.2, we can write the original matrix A as

$$A = 6 \text{id}_{n^3} + X \otimes \text{id}_n \otimes \text{id}_n + \text{id}_n \otimes X \otimes \text{id}_n + \text{id}_n \otimes \text{id}_n \otimes X. \tag{9}$$

Note that the first term is $6 \cdot \text{id}_{n^3} = 6 \cdot \text{id}_n \otimes \text{id}_n \otimes \text{id}_n$, and hence A can be written as

$$A = Y \otimes \text{id}_n \otimes \text{id}_n + \text{id}_n \otimes Y \otimes \text{id}_n + \text{id}_n \otimes \text{id}_n \otimes Y,$$

where

$$Y = X + 2 \text{id}_n = \begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & -1 & 2 & -1 \\ 0 & \dots & 0 & -1 & 2 \end{pmatrix}.$$

Next, we study the CPU-time used to solve the linear system (7) by using (a) the Matlab command $A \setminus b$, (b) the GROU Algorithm 1 with A in the compact form (8), and (c) the GROU Algorithm with A in Laplacian-Like form (9), previously obtained from the Laplacian Decomposition Algorithm 2.

We have used in the numerical experiments the following parameters values. For the GROU Algorithm 1: `tol` = $2.2204e - 16$; ϵ = $2.2204e - 16$; `rank_max` = 15; (an `iter_max` = 5 was used to perform an ALS strategy); and the number of nodes in $(0, 1)^3$ (that is, the number of rows or columns of the matrix A) increase from 10^3 to 35^3 . For the Laplacian Decomposition Algorithm we fixed `iter_max` = 4 and a tolerance `tol` = 10^{-5} . The characteristics of the computer are the same as we give in Sect. 1.

In the first experiment (see Fig. 5) the algorithms were implemented by means the Matlab standard environment to perform basic matrix calculations whereas we have done a second experiment to increase the size of the high dimensional matrices (see Fig. 6). To this end, the algorithms were implemented under the Matlab environment for sparse matrices, which require less CPU memory. In this second experience we consider matrices with a number of rows (or columns) in a range from 10^3 to 100^3 .

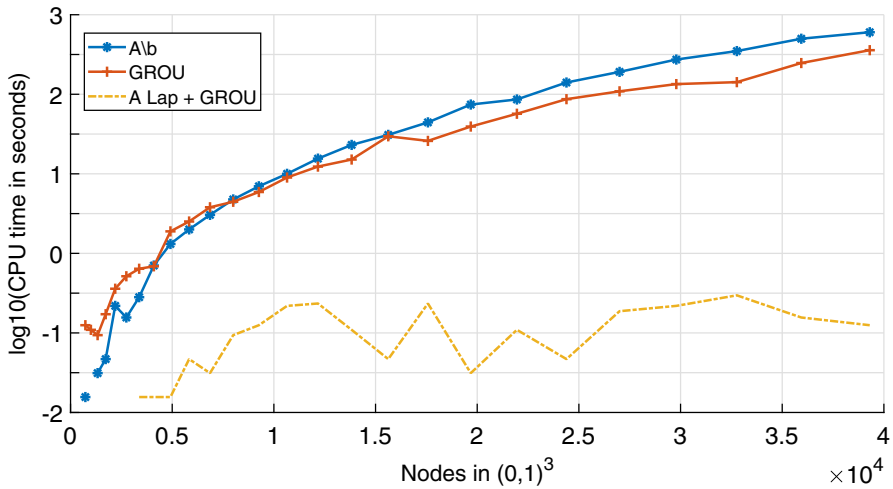


FIGURE 5. CPU Time, in seconds, employed to solve (7) by using the Matlab command $A \setminus b$, the GROU Algorithm 1, and the GROU Algorithm 1 with A written in Laplacian form, obtained from the Laplacian decomposition Algorithm 2. All algorithms were performed under the Matlab standard environment for basic matrix calculations

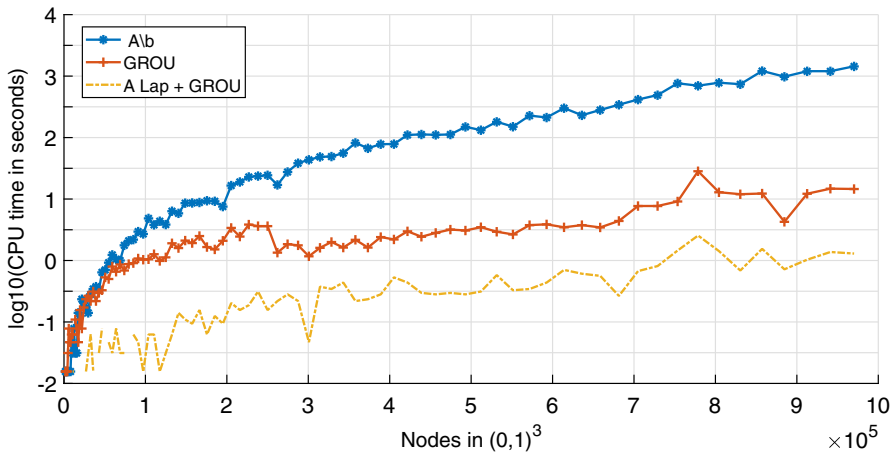


FIGURE 6. CPU Time, in second, employed to solve (7) by using the Matlab command $A \setminus b$, the GROU Algorithm 1, and the GROU Algorithm 1 with A written in Laplacian form, obtained from the Laplacian decomposition Algorithm 2. All algorithms were performed under the Matlab environment for matrices in sparse form

In both figures we observe how, for high-dimensional matrices, the GROU Algorithm 1 improves the CPU time of Matlab's command $A \setminus b$. But certainly the Laplacian Decomposition Algorithm 2 combined with the GROU Algorithm 1 significantly reduces the CPU time of the two previous methods.

5. Conclusions

We have presented a result to approximate a generic square matrix by its Laplacian form, and thus decompose it as the sum of two linearly independent matrices. This decomposition is motivated by the fact that tensor algorithms are more efficient when working with Laplacian matrices. We have also described the procedure to perform this approximation in the form of an algorithm and illustrated how it works on some basic examples.

With the proposed algorithm, we may provide an alternative way to solve linear systems, especially interesting if we combine algorithms 1 and 2, as shown in the second example presented above. Due to its structure, this matrix decomposition can be interesting for studying various types of matrices, such as sparse matrices, matrices resulting from the discretization of a PDE, adjacency matrices of simple graphs, and others. We will explore the computational benefits of this approach in future works.

Acknowledgements

This work was supported by the Generalitat Valenciana and the European Social Found under Grant [number ACIF/2020/269]; Ministerio de Ciencia, Innovación y Universidades under Grant [number RTI2018-093521-B-C32]; Universidad CEU Cardenal Herrera under Grant [number INDI22/15].

Author contributions All authors equally contributed and read and approved the final manuscript.

Funding Information The research was supported by grants from the Universidad CEU Cardenal Herrera, the Ministerio de Ciencia, Innovación y Universidades and the Generalitat Valenciana.

Declarations

Conflict of interest The authors declare no potential conflict of interest.

Open Access. This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated

otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- [1] Luzardo, D., Peña P., A.J.: Historia del Álgebra Lineal hasta los Albores del Siglo XX. *Divulgaciones Matemáticas* 14(2), 153–170 (2006)
- [2] Saad, Y.: Iterative methods for linear systems of equations: A brief historical journey. In: Brenner, S.C., Shparlinski, I.E., Shu, C.-W., Szyld, D.B. (eds.) *75 Years of Mathematics of Computation. Contemporary Mathematics*, vol. 754, pp. 197–215. American Mathematical Society, (2020). <https://doi.org/10.1090/conm/754/15141>
- [3] Saad, Y., Van der Vorst, H.A.: Iterative solution of linear systems in the 20th century. *J. Comput. Appl. Math.* **123**(1), 1–33 (2000). [https://doi.org/10.1016/S0377-0427\(00\)00412-X](https://doi.org/10.1016/S0377-0427(00)00412-X)
- [4] Leiserson, C.E., Rivest, R.L., Cormen, T.H., Stein, C.: *Introduction to Algorithms*, 3rd edn. The MIT press (2009)
- [5] Strang, G.: *Linear Algebra and Its Applications*. Thomson, Brooks/Cole, Belmont, CA (2006)
- [6] Golub, G.H., Van Loan, C.F.: *Matrix Computations*. JHU Press (2013)
- [7] Nouy, A.: Chapter 4: Low-Rank Methods for High-Dimensional Approximation and Model Order Reduction., pp. 171–226. Society for Industrial and Applied Mathematics (2017). <https://doi.org/10.1137/1.9781611974829.ch4>
- [8] Hackbusch, W.: *Tensor Spaces and Numerical Tensor Calculus (Second Edition)*. Springer Series in Computational Mathematics, p. 605. Springer (2019)
- [9] Simoncini, V.: Numerical solution of a class of third order tensor linear equations. *Boll. Unione. Mat. Ital.* **13**, 429–439 (2020). <https://doi.org/10.1007/s40574-020-00247-4>
- [10] Ammar, A., Chinesta, F., Falcó, A.: On the convergence of a Greedy Rank-One Update algorithm for a class of linear systems. *Arch. Comput. Methods Eng.* **17**(4), 473–486 (2010). <https://doi.org/10.1007/s11831-010-9048-z>
- [11] Georgieva, I., Hofreither, C.: Greedy low-rank approximation in Tucker format of solutions of tensor linear systems. *J. Comput. Appl. Math.* **358**, 206–220 (2019). <https://doi.org/10.1016/j.cam.2019.03.002>
- [12] Falcó, A., Nouy, A.: Proper generalized decomposition for nonlinear convex problems in tensor Banach spaces. *Numer. Math.* **121**, 503–530 (2012). <https://doi.org/10.1007/s00211-011-0437-5>
- [13] Quesada, C., Xu, G., González, D., Alfaro, I., Leygue, A., Visonneau, M., Cueto, E., Chinesta, F.: Un método de descomposición propia generalizada para operadores diferenciales de alto orden. *Rev. Int. Metod. Numer.* **31**(3), 188–197 (2015). <https://doi.org/10.1016/j.rimmi.2014.09.001>
- [14] MATLAB: Version R2021b. The MathWorks Inc., Natick, Massachusetts (2021)

- [15] Falcó, A.: Tensor Formats Based on Subspaces are Positively Invariant Sets for Laplacian-Like Dynamical Systems. In: Abdulle, A., Deparis, S., Kressner, D., Nobile, F., Picasso, M. (eds.) Numerical Mathematics and Advanced Applications - ENUMATH 2013, pp. 315–323. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-10705-9_31
- [16] Falcó, A., Hackbusch, W., Nouy, A.: On the Dirac–Frenkel variational principle on tensor banach spaces. *Found Comput Math* **19**(1), 159–204 (2019). <https://doi.org/10.1007/s10208-018-9381-4>
- [17] Hackbusch, W., Khoromskij, B., Sauter, S., Tyrtysnikov, E.: Use of tensor formats in elliptic eigenvalue problems. *Numer. Linear Algebra Appl.* **19**, 133–151 (2012). <https://doi.org/10.1002/nla.793>
- [18] Heidel, G., Khoromskaia, V., Khoromskij, B.N., Schulz, V.: Tensor product method for fast solution of optimal control problems with fractional multidimensional Laplacian in constraints. *J. Comput. Phys.* **424**, 109865 (2021). <https://doi.org/10.1016/j.jcp.2020.109865>
- [19] Gallier, J., Quaintance, J.: *Differential Geometry and Lie Groups*. Springer (2020)

J. Alberto Conejero and María Mora-Jiménez
Instituto Universitario de Matemática Pura y Aplicada
Universitat Politècnica de València
Camí de Vera
46022 Valencia
Spain
e-mail: aconejero@mat.upv.es;
mamoji7@doctor.upv.es

Antonio Falcó
ESI International Chair@CEU-UCH, Departamento de Matemáticas,
Física y Ciencias Tecnológicas
Universidad Cardenal Herrera-CEU, CEU Universities
San Bartolomé 55, Alfara del Patriarca
46115 Valencia
Spain
e-mail: afalco@uchceu.es

Received: September 27, 2022.

Accepted: June 27, 2023.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.