Annals of **F**uzzy **M**athematics and **I**nformatics
Volume x, No. x, (Month 201y), pp. 1–xx
ISSN: 2093–9310 (print version)
ISSN: 2287–6235 (electronic version)
http://www.afmi.or.kr

Kronecker sum decomposition and its applications

Youssouf Mezzar, Kacem Belghaba

Received 24 June 2021; Revised 24 September 2021; Accepted 4 October 2021

Abstract. We develop here, two types of Kronecker sum decompositions KSGD and KSID. We study Kronecker sum decomposition KSD, KSGD, and KSID and its algorithms. We conclude with some applications of the Kronecker sum decomposition.

2010 AMS Classification: 15A69, 65F08, 65F10, 65Y04

Keywords: Kronecker product decomposition, Kronecker sum decomposition, Isomer decomposition, Gemel decomposition, Matrix equation, Rank, Conjugate Gradient, Preconditioner incomplete Cholesky factorisation.

Corresponding Author: Kacem Belghaba (belghaba@yahoo.fr)

1. Introduction

The Kronecker product of two matrices, from its author Leopold Kronecker (1823-1891) is a very important field of linear algebra, numerical linear algebra, numerical analysis, data mining, signal processing, computer vision and others. Many properties about its trace, determinant, eigenvalues, and other decompositions have been discovered during this time, and are now part of classical linear algebra literature (See for instance [1, 2, 3, 4]).

As noted above, the Kronecker product is a particulary important tool in linear algebra and in matrix equations. By using appropriate hypotheses, it is possible to transform a matrix equation into a linear system [5] and vice-versa. The same applies to the Kronecker sum. This will be a major part of our paper.

Kronecker sum decomposition (KSD) means that a matrix T can be transformed that the Kronecker sum form of other matrices A, B, i.e. $T = A \otimes I + I \otimes B$, Kronecker sum gemel decomposition (KSGD) means the Kronecker sum in the special case $A = B$. Kronecker sum isomer decomposition (KSID) corresponds to the case $T = A \otimes I + I \otimes A^T$, where A^T is the transpose matrix of A.

Liu [6] had some interesting work about Kronecker product decomposition (KPD), Zhang and Ding [7] gave a new result about the singular value of the Kronecker product and their applications. Moreover, Jarlebring [8] studied methods for Lyapunov equations and Datta [9] dealt with the iterative methods of solving matrix equations.

In this paper, from the point of view of the inverse problem theory [6], we mainly explore the sufficient and necessary conditions and algorithms for KSD, KSGD, KSID and some of their applications to build a new preconditioner for solving linear systems using the Conjugate Gradient (CG) method, which can be converted into a Conjugate Gradient method in matrix format (CG-M) for solving matrix equations such as Lyapunov equations and Sylvester equations resulting from transforming linear systems using Kronecker sum decomposition. We note the difference in execution speed between CG and CG-M methods.

Our paper is organized as follows. We present the preliminaries on the definitions and properties of Kronecker product and sum in Section 2. We devote the Section 3 to the study of Kronecker sum decomposition with results on the sufficient and necessary conditions respectively of KSD in Section 3.1 and KSGD and KSID in Section 3.2. We present in Section 4, some applications of the algebraic theory of Kronecker previously studied. Finally, the Section 5 achieves our work.

2. Preliminaries

2.1. Definitions.

Definition 2.1 (Vec-operation [10])**.** For any $F \in \mathbb{R}^{r\times s}$, we define the vector

$$vec(F) = [f_{11}, ..., f_{r1}, f_{12}, ..., f_{r2}, ..., f_{1s}, ..., f_{rs}]^T \in \mathbb{R}^{rs},$$

where, the columns of F are stacked on top of each other.

Definition 2.2 (Kronecker product [11])**.** Let $A \in \mathbb{R}^{r\times s}$ and $B \in \mathbb{R}^{m\times n}$ be two matrices. Then the $rm \times sn$ matrix

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & ... & a_{1s}B \\ a_{21}B & a_{22}B & ... & a_{2s}B \\ \vdots & \vdots & & \vdots \\ a_{r1}B & a_{r2}B & \dots & a_{rs}B \end{bmatrix} \tag{2.1}$$

is the Kronecker product of A and B.

Remark 2.3. The Kronecker product is sometimes called a direct product or a tensor product [12].

Example 2.4. The Kronecker product of two 2×2 matrices gives a 4×4 matrix .

If $A = \begin{bmatrix} 1 & 2 \\ 3 & 7 \end{bmatrix}$ and $B = \begin{bmatrix} 4 & 6 \\ 5 & 8 \end{bmatrix}$, then

$$A \otimes B = \begin{bmatrix} 1 & 2 \\ 3 & 7 \end{bmatrix} \otimes \begin{bmatrix} 4 & 6 \\ 5 & 8 \end{bmatrix} = \begin{bmatrix} 1\times 4 & 1\times 6 & 2\times 4 & 2\times 6 \\ 1\times 5 & 1\times 8 & 2\times 5 & 2\times 8 \\ 3\times 4 & 3\times 6 & 7\times 4 & 7\times 6 \\ 3\times 5 & 3\times 8 & 7\times 5 & 7\times 8 \end{bmatrix}$$

$$= \begin{bmatrix} 4 & 6 & 8 & 12 \\ 5 & 8 & 10 & 16 \\ 12 & 18 & 28 & 42 \\ 15 & 24 & 35 & 56 \end{bmatrix}.$$

Some immediate consequences of Definition 2.2 are listed below [11].

- The Kronecker product of two matrices differs from ordinary matrix multiplication in that it is defined for any two matrices . It does not require that the number of columns in first matrix be equal to the number of rows in second matrix.

Example 2.5. Here is an example of a Kronecker product of a 2×3 matrix and a 3×3 matrix

$$\begin{bmatrix} 2 & 3 & 1 \\ 0 & 5 & 4 \end{bmatrix} \otimes \begin{bmatrix} 2 & 4 & 1 \\ 5 & 0 & 7 \\ 3 & 2 & 2 \end{bmatrix} = \begin{bmatrix} 4 & 8 & 2 & 6 & 12 & 3 & 2 & 4 & 1 \\ 10 & 0 & 14 & 15 & 0 & 21 & 5 & 0 & 7 \\ 6 & 4 & 4 & 9 & 6 & 6 & 3 & 2 & 2 \\ 0 & 0 & 0 & 10 & 20 & 5 & 8 & 16 & 4 \\ 0 & 0 & 0 & 25 & 0 & 35 & 20 & 0 & 28 \\ 0 & 0 & 0 & 15 & 10 & 10 & 12 & 8 & 8 \end{bmatrix}.$$

- The dimensions of $A \otimes B$ and $B \otimes A$ are equal.
  If A is $r \times s$ matrix and B is $m \times n$ matrix, then both $A \otimes B$ and $B \otimes A$ have rm rows and sn columns. However, Kronecker product is not commutative.

Example 2.6. Let A and B be the following 2×2 matrices:

$$A = \begin{bmatrix} 2 & 1 \\ 5 & 4 \end{bmatrix} \qquad \textit{and} \qquad B = \begin{bmatrix} 2 & 0 \\ 1 & 5 \end{bmatrix}.$$

Then

$$A \otimes B = \begin{bmatrix} 4 & 0 & 2 & 0 \\ 2 & 10 & 1 & 5 \\ 10 & 0 & 8 & 0 \\ 5 & 25 & 4 & 20 \end{bmatrix}, \quad \textit{while} \quad B \otimes A = \begin{bmatrix} 4 & 2 & 0 & 0 \\ 10 & 8 & 0 & 0 \\ 2 & 1 & 10 & 5 \\ 5 & 4 & 25 & 20 \end{bmatrix}.$$

Remark 2.7. In matlab, the vec-operation can be computed with `b=B(:)` and the inverse operation can be computed with `B=reshape(b,n,length(b)/n)`, such that n is the row number of matrix B. The Kronecker product is implemented in `kron(A,B)`.

Definition 2.8 (Kronecker sum [13])**.** Let $A \in \mathbb{R}^{r\times r}$ and $B \in \mathbb{R}^{s\times s}$. We denote by $A \oplus B$ the Kronecker sum of A and B defined by:

$$A \oplus B = (A \otimes I_s) + (I_r \otimes B), \tag{2.2}$$

where $A \oplus B$ is an $rs \times rs$ matrix, and I_r, I_s represent the identity matrices of order r, s respectively.

Note that the Kronecker sum definition may have other formulations in the literature. Notably, Amoia et al. [14] as well as Graham [15] use the above definition, while Horn and Johnson [16] use $A \oplus B = (I_m \otimes A) + (B \otimes I_n)$. In our case, we will work with the Amoia-De Micheli's version of the Kronecker sum.

Example 2.9.
(1) Let $A = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \\ 1 & 1 & 4 \end{bmatrix}$, $I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ and $B = \begin{bmatrix} 2 & 1 \\ 2 & 3 \end{bmatrix}$, $I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Then we have

$$A \oplus B = (A \otimes I_2) + (I_3 \otimes B) = \begin{bmatrix} 1 & 0 & 2 & 0 & 3 & 0 \\ 0 & 1 & 0 & 2 & 0 & 3 \\ 3 & 0 & 2 & 0 & 1 & 0 \\ 0 & 3 & 0 & 2 & 0 & 1 \\ 1 & 0 & 1 & 0 & 4 & 0 \\ 0 & 1 & 0 & 1 & 0 & 4 \end{bmatrix} + \begin{bmatrix} 2 & 1 & 0 & 0 & 0 & 0 \\ 2 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 2 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 2 & 3 \end{bmatrix}$$

$$= \begin{bmatrix} 3 & 1 & 2 & 0 & 3 & 0 \\ 2 & 4 & 0 & 2 & 0 & 3 \\ 3 & 0 & 4 & 1 & 1 & 0 \\ 0 & 3 & 2 & 5 & 0 & 1 \\ 1 & 0 & 1 & 0 & 6 & 1 \\ 0 & 1 & 0 & 1 & 2 & 7 \end{bmatrix}.$$

(2) Let $A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$ and $I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. Then we get

$$A \oplus A = A \otimes I + I \otimes A = \begin{bmatrix} 2I & -I & 0 \\ -I & 2I & -I \\ 0 & -I & 2I \end{bmatrix} + \begin{bmatrix} A & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & A \end{bmatrix}$$

$$= \left[\begin{array}{ccc|ccc|ccc} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 \\ \hline -1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\ \hline 0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 \end{array}\right].$$

In general, $A \oplus B \neq B \oplus A$.

Example 2.10. Let A and B be the following 2×2 matrices:

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } B = \begin{bmatrix} 2 & 3 \\ 1 & 5 \end{bmatrix}.$$

Then

$$A \oplus B = A \otimes I_2 + I_2 \otimes B = \begin{bmatrix} 3 & 3 & 0 & 0 \\ 1 & 6 & 0 & 0 \\ 0 & 0 & 3 & 3 \\ 0 & 0 & 1 & 6 \end{bmatrix},$$

while

$$B \oplus A = B \otimes I_2 + I_2 \otimes A = \begin{bmatrix} 3 & 0 & 3 & 0 \\ 0 & 3 & 0 & 3 \\ 1 & 0 & 6 & 0 \\ 0 & 1 & 0 & 6 \end{bmatrix}.$$

2.2. **Properties.**

2.2.1. **Basic properties.**
We state some basic facts about Kronecker product [11, 17], which are easily derived from Definition 2.2.

(1) $\mu \otimes A = A \otimes \mu = \mu A$, for any scalar μ.
(2) $(\lambda A) \otimes (\mu B) = \lambda\mu(A \otimes B)$, for any scalars λ and μ.
(3) $(A + B) \otimes C = (A \otimes C) + (B \otimes C)$, if A and B are the same size.
(4) $A \otimes (B + C) = (A \otimes B) + (A \otimes C)$, if A and B are the same size.
(5) $A \otimes (B \otimes C) = (A \otimes B) \otimes C$.
(6) $(A \otimes B)^T = A^T \otimes B^T$.

Theorem 2.11 ([7])**.** *Let A and B any two square matrices.*
(1) *If A and B are invertible matrices, then $A \otimes B$ is invertible and $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$.*
(2) *If A and B are normal matrices, then $A \otimes B$ is a normal matrix.*
(3) *If A and B are unitary (orthogonal) matrices, then $A \otimes B$ is a unitary (orthogonal) matrix.*

Theorem 2.12 (The mixed product rule [7])**.** *Suppose that A, B, C, D are rectangular matrices whose dimensions make it possible to define the products AC and BD. Then*

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD). \tag{2.3}$$

Theorem 2.13 ([11, 10])**.** *For $m, n \in \mathbb{N}$, consider the square matrices $A \in \mathbb{R}^{m,m}$ with eigenpairs $(\alpha_i; x_i), i = 1, ...m$, and $B \in \mathbb{R}^{n,n}$ with eigenpairs $(\beta_j; y_j),\ j = 1, ...n$.*
(1) *If $A^T = A$ and $B^T = B$, then $(A \otimes B)^T = A \otimes B$ and $(A \oplus B)^T = A \oplus B$.*
(2)$(A \otimes B)(x_i \otimes y_j) = (\alpha_i\beta_j)(x_i \otimes y_j), i = 1, ...m,\ j = 1, ...n.$
(3) $(A \oplus B)(x_i \otimes y_j) = (\alpha_i + \beta_j)(x_i \otimes y_j), i = 1, ...m,\ j = 1, ...n.$
(4) *If one of A, B is positive definite and the other positive semi definite, then $A \oplus B$ is positive definite.*
(5) *If A, B and X are matrices in $\mathbb{R}^{n\times n}$, then $BXA^T = (A \otimes B)vec(X)$.*

Proof. (1) By Basic properties 2.2.1 (6), $(A \otimes B)^T = A^T \otimes B^T = A \otimes B$. Moreover, $(A \oplus B)^T = (A \otimes I + I \otimes B)^T = (A \otimes I)^T + (I \otimes B)^T = (A \otimes I) + (I \otimes B) = A \oplus B$.
(2) For all $i,\ j$, we have

$$(A \otimes B)(x_i \otimes y_j) = (Ax_i) \otimes (By_j) = (\alpha_i x_i) \otimes (\beta_j y_j) = (\alpha_i\beta_j)(x_i \otimes y_j).$$

(3) We have $(A \otimes I)(x_i \otimes y_j) = \alpha_i(x_i \otimes y_j)$ and $(I \otimes B)(x_i \otimes y_j) = \beta_j(x_i \otimes y_j)$. Moreover, for $i = 1, ..., m,\ j = 1, ..., n$, we get

$$\begin{aligned}(A \oplus B)(x_i \otimes y_j) &= A \otimes I + I \otimes B)(x_i \otimes y_j)\\ &= (A \otimes I)(x_i \otimes y_j) + (I \otimes B)(x_i \otimes y_j)\\ &= \alpha_i(x_i \otimes y_j) + \beta_j(x_i \otimes y_j)\\ &= (\alpha_i + \beta_j)(x_i \otimes y_j).\end{aligned}$$

(4) By (1), the eigenvalues $\alpha_i + \beta_j$ are positive because for all $i,\ j$, both α_i and β_j are non negative and one of them is positive. Then $A \oplus B$ is positive definite.

Partition $X$and A by columns as follows : $X = [x_1, ..., x_n]$ and $A = [a_1, ..., a_n]$. Then

$$\begin{aligned}(A \otimes B)vec(X) &= \begin{bmatrix} a_{11}B & ... & a_{1n}B \\ : & \ddots & : \\ a_{n1}B & \dots & a_{nn}B \end{bmatrix} \begin{bmatrix} x_1 \\ : \\ x_n \end{bmatrix}\\ &= B\left[\sum_j a_{1j}x_j, ..., \sum_j a_{nj}x_j\right]\\ &= B\,[Xa_1, ..., Xa_n]\\ &= BXA^T.\end{aligned}$$

□

Theorem 2.13 (5) is extremely important in solving the matrix equations [7]. For more properties on Kronecker products and sum see [16] and [10].

3. Kronecker sum decomposition

Initially, we assume that all matrices T, A, B used below have the appropriate size.

3.1. Sufficient and necessary conditions of KSD.

First, we give the sufficient and necessary conditions of Kronecker sum decomposition.

Theorem 3.1 (KSD)**.** *An arbitrary square matrix $T \in \mathbb{R}^{nm \times nm}$,*

$$T = \begin{bmatrix} T_{11} & ... & T_{1n} \\ : & \ddots & : \\ T_{n1} & \dots & T_{nn} \end{bmatrix} \text{ with } T_{ij} \in \mathbb{R}^{m \times m}(i = 1, ..., n, j = 1, ..., n),$$

may be decomposed to the form:

$$T = A \otimes I + I \otimes B$$

$$\Leftrightarrow \begin{cases} \text{if we take } B = T_{11}\ ,\ \widetilde{T} = T - (I_n \otimes B) \ \text{ and} \\ C = \{vec(\widetilde{T}_{11}), vec(\widetilde{T}_{12}), ..., vec(\widetilde{T}_{1n}), ..., vec(\widetilde{T}_{nn})\}, \ \text{ then} \\ rank\{C, vec(I_m)\} = 1, \ \text{and denote } vec(A)^T = \text{the first line of the matrix } C \end{cases}$$

with $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{m \times m}$, m, n are some certain integers.

Corollary 3.2. *An arbitrary square matrix* $T \in \mathbb{R}^{nm \times nm}$,

$$T = \begin{bmatrix} T_{11} & ... & T_{1n} \\ \vdots & \ddots & \vdots \\ T_{n1} & \dots & T_{nn} \end{bmatrix} \text{ with } T_{ij} \in \mathbb{R}^{m \times m} (i = 1, ..., n, j = 1, ..., n),$$

may be decomposed to the form:

$$T = A \otimes I + I \otimes B \Rightarrow rank\{vec(T_{11}), vec(T_{12}), ..., vec(T_{1n}), ..., vec(T_{nn})\} \leq 2$$

with $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{m \times m}$, m, n *are integers.*

Proof. Suppose $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{m \times m}$ and let I_n, I_m be the identity matrices of size n, m respectively. Then $A \otimes I_m$, $I_n \otimes B$ and $A \oplus B$ are the block matrices of size $nm \times nm$, where

$$A \otimes I = \begin{bmatrix} T^A_{11} & ... & T^A_{1n} \\ \vdots & \ddots & \vdots \\ T^A_{n1} & \dots & T^A_{nn} \end{bmatrix},\ I \otimes B = \begin{bmatrix} T^B_{11} & ... & T^B_{1n} \\ \vdots & \ddots & \vdots \\ T^B_{n1} & \dots & T^B_{nn} \end{bmatrix},\ A \oplus B = \begin{bmatrix} T_{11} & ... & T_{1n} \\ \vdots & \ddots & \vdots \\ T_{n1} & \dots & T_{nn} \end{bmatrix},$$

with $T_{ij}, T^A_{ij}, T^B_{ij} \in \mathbb{R}^{m \times m}$, and $T_{ij} = T^A_{ij} + T^B_{ij} (i = 1, ..., n, j = 1, ..., n)$, where

$$\begin{aligned} & rank\{vec(T_{11}), ..., vec(T_{nn})\} \\ & = rank\{vec(T^A_{11} + T^B_{11}), ..., vec(T^A_{nn} + T^B_{nn})\} \\ & = rank\{vec(T^A_{11}) + vec(T^B_{11}), ..., vec(T^A_{nn}) + vec(T^B_{nn})\} \\ & = rank\{\{vec(T^A_{11}), ..., vec(T^A_{nn})\} + \{vec(T^B_{11}), ..., vec(T^B_{nn})\}\} \\ & \leq rank\{vec(T^A_{11}), ..., vec(T^A_{nn})\} + rank\{vec(T^B_{11}), ..., vec(T^B_{nn})\} \\ & \leq 1 + 1 = 2. \end{aligned}$$

□

This can be demonstrated by using the following two theorems.

Theorem 3.3 (KPD [6])**.** *An arbitrary matrix* $T \in \mathbb{R}^{nm \times nm}$,

$$T = \begin{bmatrix} T_{11} & ... & T_{1n} \\ \vdots & \ddots & \vdots \\ T_{n1} & \dots & T_{nn} \end{bmatrix} \text{ with } T_{ij} \in \mathbb{R}^{m \times m} (i = 1, ..., n, j = 1, ..., n),$$

can be decomposed to the form:

$$T = A \otimes B \Leftrightarrow rank\{vec(T_{11}), vec(T_{12}), ..., vec(T_{1n}), ..., vec(T_{nn})\} = 1$$

with $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{m \times m}$, m, n *are integers.*

Theorem 3.4 ([13])**.** *Let* $C, D \in \mathbb{R}^{n \times n}$. *Then*

$$0 \leq rank(C + D) \leq rank(C) + rank(D).$$

Remark 3.5. In general, the KSD of a matrix is not unique because of

$$\begin{aligned}(A+\lambda I)\oplus(B+\mu I) &= (A+\lambda I)\otimes I + I\otimes(B+\mu I)\\ &= (A\otimes I)+\lambda(I\otimes I)+(I\otimes B)+\mu(I\otimes I)\\ &= (A\otimes I)+(I\otimes B)+(\lambda+\mu)(I\otimes I)\\ &= (A\otimes I)+(I\otimes B)\\ &= A\oplus B\end{aligned}$$

with $\lambda+\mu=0$ for arbitrary matrices A, B and constants λ, μ.

Algorithm 1: Kronecker sum decomposition (KSD)

step 1:	input T, n, m, verify the size of T is $nm \times nm$
step 2:	define $B = T_{11}$
step 3:	calculate $\widetilde{T} = T - (I_n \otimes B)$
step 4:	define $\widetilde{T}_{ij}, i = 1, ..., n, j = 1, ..., n$
step 5:	calculate $C = \{vec(\widetilde{T}_{11}), vec(\widetilde{T}_{12}), ..., vec(\widetilde{T}_{1n}), ..., vec(\widetilde{T}_{nn})\}$
step 6:	if $rank\{C, vec(I_n)\} = 1$ go to step 7
	else output "can not decompose"; end
step 7:	define A which $vec(A)'$ =the first line of the matrix C; output A, B; end

3.2. **Sufficient and necessary conditions of KSGD and KSID.**
As we pointed out in the introduction, Fuxiang Liu [6] developed two new kinds of Kronecker decompositions, Kronecker product gemel decomposition(KPGD) and Kronecker product isomer decomposition (KPID). We will try to adapt these to Kronecker sum gemel decomposition (KSGD) and Kronecker sum isomer decomposition (KSID).

Theorem 3.6 (KPGD [6]). *An arbitrary square matrix $T \in \mathbb{R}^{n^2\times n^2}, (T \neq 0)$,*

$$T = \begin{bmatrix} T_{11} & \dots & T_{1n}\\ \vdots & \ddots & \vdots\\ T_{n1} & \dots & T_{nn}\end{bmatrix} \quad with\ T_{ij} \in \mathbb{R}^{n\times n}(i = 1, ..., n, j = 1, ..., n),$$

may be decomposed to the form:

$$T = A \otimes A$$
$$\Leftrightarrow \begin{cases} \text{there is a sub-block } T_{ij} \neq 0,\ \text{and } t^{ij}_{i,j} > 0,\ \text{where } T_{ij} = (t^{ij}_{s,t})_{s=1,\dots,n,t=1,\dots,n},\\ \text{if denotes } T_{ij}/\sqrt{t^{ij}_{i,j}} = A \text{ or } -A,\ \text{then } \forall i,j, T_{ij} = \sqrt{t^{ij}_{i,j}}A,\ \text{with } A \in \mathbb{R}^{n\times n}.\end{cases}$$

We adapt Theorem 3.6 valid for the Kronecker product to Kronecker sum by posing

$$T_{ii} - ((t^{ii}_{i,i})/2)I = A \text{ in place of } T_{ij}/\sqrt{t^{ij}_{i,j}} = A.$$

Theorem 3.7 (KSGD). *An arbitrary square matrix $T \in \mathbb{R}^{n^2 \times n^2}$,*

$$T = \begin{bmatrix} T_{11} & ... & T_{1n} \\ \vdots & \ddots & \vdots \\ T_{n1} & \dots & T_{nn} \end{bmatrix}, \ with\ T_{ij} \in \mathbb{R}^{n \times n} (i = 1, ..., n, j = 1, ..., n)$$

can be decomposed to the form:

$$T = A \otimes I + I \otimes A \Leftrightarrow \begin{cases} there\ exists\ a\ diagonal\ sub\text{-}block\ T_{ii} = (t^{ii}_{s,t})_{s=1,...,n,t=1,...,n}, \\ where\ T_{ij} = (t^{ii}_{i,j})I \qquad if \quad i \neq j, \\ and\ denote\ A = T_{ij} - ((t^{ij}_{i,j})/2)I \qquad if \quad i = j, \\ then\ for\ \forall i, T_{ii} = ((t^{ii}_{i,i})/2)I + A,\ with\ A \in \mathbb{R}^{n \times n}. \end{cases}$$

Algorithm 2: Kronecker sum gemel decomposition (KSGD)

step 1:	input T, n, verify the size of T is $n^2 \times n^2$
step 2:	define $flag = 0$, $T_{ij}, i = 1, ..., n, j = 1, ..., n$
step 3:	for $(i, j), i = 1, ..., n, j = 1, ..., n$
	define $B = T_{ij} - I * (t^{ij}_{i,j})/2$;
	if $T = (B \otimes I) + (I \otimes B)$; flag=1; break;
step 4:	if $flag = 1$; $A = B$; output A; end else output "can not gemel decompose"; end

Theorem 3.8 (KSID). *An arbitrary square matrix $T \in \mathbb{R}^{n^2 \times n^2}$,*

$$T = \begin{bmatrix} T_{11} & ... & T_{1n} \\ \vdots & \ddots & \vdots \\ T_{n1} & \dots & T_{nn} \end{bmatrix}, \ with\ T_{ij} \in \mathbb{R}^{n \times n} (i = 1, ..., n, j = 1, ..., n)$$

can be decomposed to the form:

$$T = A \otimes I + I \otimes A \Leftrightarrow \begin{cases} there\ exists\ a\ diagonal\ sub\text{-}block\ T_{ii} = (t^{ii}_{r,s})_{r=1,...,n,s=1,...,n}, \\ where\ T_{ij} = (t^{ii}_{j,i})I \qquad if \quad i \neq j, \\ and\ denote\ A' = T_{ij} - ((t^{ij}_{i,j})/2)I \qquad if \quad i = j, \\ then\ for\ \forall i, M_{ii} = ((t^{ii}_{i,i})/2)I + A',\ with\ A \in \mathbb{R}^{n \times n}. \end{cases}$$

4. Applications

4.1. **Preconditioning the Conjugate Gradient method.**

To solve large sparse linear systems,

$$Tx = f, \tag{4.1}$$

where T is a sparse symmetric positive definite matrix, we use an iterative method such as the Conjugate Gradient method (CG) [10], which we summarize in the following algorithm:
We start with $x_0, p_0 = r_0 = f - Tx_0$, the algorithm is stopped when $||r_k||_2 \,/\, ||f||_2 \leq tol$,

Algorithm 3: Conjugate Gradient (CG)

$$\begin{aligned}
d_k &= Tp_k;\\
\alpha_k &= (r_k^T r_k)/(p_k^T d_k);\\
x_{k+1} &= x_k + \alpha_k p_k;\\
r_{k+1} &= r_k - \alpha_k d_k;\\
\beta_k &= (r_{k+1}^T r_{k+1})/(r_k^T r_k);\\
p_{k+1} &= r_{k+1} + \beta_k p_k.
\end{aligned}$$

A preconditioner is a matrix M that we use to speed up the convergence of the CG method to get the Preconditioned Conjugate Gradient method (PCG) [10].

We start with $x_0, r_0 = f - Tx_0, p_0 = s_0 = Mr_0$, the algorithm is stopped when $||r_k||_2 \,/\, ||f||_2 \leq tol$,

Algorithm 4: Preconditioned Conjugate Gradient (PCG)

$$\begin{aligned}
d_k &= Tp_k;\\
\alpha_k &= (s_k^T r_k)/(p_k^T d_k);\\
x_{k+1} &= x_k + \alpha_k p_k;\\
r_{k+1} &= r_k - \alpha_k d_k;\\
s_{k+1} &= s_k - \alpha_k M d_k;\\
\beta_k &= (s_{k+1}^T r_{k+1})/(s_k^T r_k);\\
p_{k+1} &= s_{k+1} + \beta_k p_k.
\end{aligned}$$

4.1.1. **Numerical example.**
Consider an open Ω and a function f continuous on Ω. The Poisson problem is to find a function of two variables $\phi(x, y)$ defined on Ω which verifies the following two relations:

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = f(x, y) \qquad \text{on} \quad \Omega, \tag{4.2}$$

and boundary conditions

$$\phi(x,y) = 0 \quad \text{on} \quad \partial\Omega.$$

The finite difference approximation of the Poisson problem (4.2) can be written as follows:

$$4w_{i,j} - w_{i+1,j} - w_{i-1,j} - w_{i,j+1} - w_{i,j-1} = -h^2 f(x_i, y_j), \tag{4.3}$$

where w_{ij} is an approximation of $\phi(x_i, y_j)$, $x_i = ih$ and $y_j = jh$, $0 \le i, j \le m+1$.

The system (4.3) can be written as the following linear system [18]:

$$Tx = f, \quad T \in \mathbb{R}^{n\times n}, \quad f \in \mathbb{R}^n, \quad n = m^2, \tag{4.4}$$

where

$$T = \begin{bmatrix} D & -I & 0 & 0 & 0 & \dots & 0 \\ -I & D & -I & 0 & 0 & \dots & 0 \\ 0 & -I & D & -I & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & -I & D & -I & 0 \\ 0 & \dots & \dots & 0 & -I & D & -I \\ 0 & \dots & \dots & \dots & 0 & -I & D \end{bmatrix} \tag{4.5}$$

such I is the identity matrix of size $m \times m$, and D is also an $m \times m$ matrix, where

$$D = \begin{bmatrix} 4 & -1 & 0 & 0 & 0 & \dots & 0 \\ -1 & 4 & -1 & 0 & 0 & \dots & 0 \\ 0 & -1 & 4 & -1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & -1 & 4 & -1 & 0 \\ 0 & \dots & \dots & 0 & -1 & 4 & -1 \\ 0 & \dots & \dots & \dots & 0 & -1 & 4 \end{bmatrix}$$

and

$$x = \begin{bmatrix} \phi_1, & \phi_2, & \phi_3, & \dots, & \phi_n \end{bmatrix}^T, \; f = \begin{bmatrix} f_1, & f_2, & f_3, & \dots, & f_n \end{bmatrix}^T.$$

For $m = 3$, the system (4.4) becomes

$$\begin{bmatrix} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \end{bmatrix} \tag{4.6}$$

The matrix T may be decomposed as KSGD ($T = A \otimes I + I \otimes A$) such that I is the $m \times m$ identity matrix and $A \in \mathbb{R}^{m\times m}$, where

$$A = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & \dots & 0 \\ -1 & 2 & -1 & 0 & 0 & \dots & 0 \\ 0 & -1 & 2 & -1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & -1 & 2 & -1 & 0 \\ 0 & \dots & \dots & 0 & -1 & 2 & -1 \\ 0 & \dots & \dots & \dots & 0 & -1 & 2 \end{bmatrix}$$

It is clear that the matrix T is symmetric positive definite [10], so the linear system (4.4) can be solved by the Conjugate Gradient method.

The following table summarizes the results of solving the Poisson problem using the CG method and PCG method by taking different preconditioners as incomplete Cholesky preconditioner **IC(0)** and Incomplete Cholesky with threshold dropping **ICT(1e-3)** preconditioner, and finally our preconditioner **IC-K** defined by $M = L \otimes I + I \otimes L$, where $A = LL'$ is the Incomplete Cholesky factorisation of the matrix A. This is done using Matlab database [19].

Matrix size (n)	**PCG with tol=1e-8**				
	Prec	# Iterations	P-time	It-time	Tot-time
40 000	No-P	369	0	0.9822	0.9822
	IC-K	63	0.0203	0.3244	0.3447
	IC(0)	139	0.0051	0.7240	0.7291
	ICT(1e-3)	35	0.0341	0.2832	0.3174
160 000	No-P	734	0	9.1999	9.1999
	IC-K	95	0.0926	2.2571	2.3497
	IC(0)	274	0.0263	6.3183	6.3446
	ICT(1e-3)	66	0.1206	2.3659	2.4865
360 000	No-P	1105	0	31.4799	31.4799
	IC-K	119	0.2008	6.5616	6.7624
	IC(0)	401	0.0520	22.8266	22.8786
	ICT(1e-3)	98	0.3789	9.5338	9.9127
640 000	No-P	1479	0	80.2185	80.2185
	IC-K	140	0.3864	15.9863	16.3727
	IC(0)	533	0.0889	64.2007	64.2896
	ICT (1e-3)	125	1.3558	23.5370	24.8927
1 000 000	No-P	/	/	/	/
	IC-K	159	0.5161	22.9225	23.4386
	IC(0)	666	0.1277	115.8678	115.9954
	ICT(1e-3)	143	1.0350	39.8834	40.9184

TABLE 1. *Comparison of the preconditioners, IC-K, IC(0) and ICT(1e-3) for solving the Poisson problem with Preconditioned Conjugate Gradient method (PCG).*

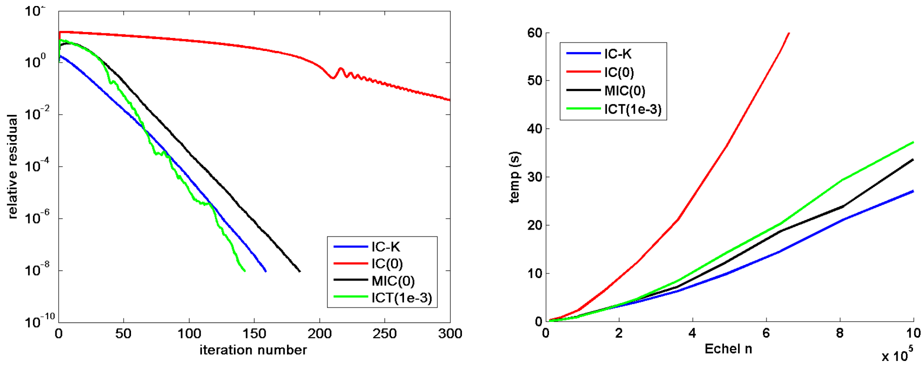

FIGURE 1. *Relative residual versus iteration number and execution time for PCG with variant incomplete Cholesky factorization preconditioners.*

4.2. **Transform linear system into matrix equation.**

4.2.1. **Sylvester equation.**

Definition 4.1. Let $A, B, F \in \mathbb{R}^{n\times n}$ be square matrices. Sylvester matrix equation is a matrix equation in the form:

$$BX + XA^T = F \tag{4.7}$$

in which the matrix $X \in \mathbb{R}^{n\times n}$ is the unknown.

Property 4.2. *If the matrix $T \in \mathbb{R}^{n^2\times n^2}$, may be decomposed as KSD ($T = A \otimes I + I \otimes B$), such that $A, B \in \mathbb{R}^{n\times n}$, then*

$$Tx = f \Leftrightarrow BX + XA^T = F,$$

where X, $F \in \mathbb{R}^{n\times n}$ and $x = vec(X), f = vec(F)$.

Proof. This follows immediately from Theorem 2.13 (5).

$$\begin{aligned} Tx = f \quad &\Leftrightarrow \quad (A \otimes I + I \otimes B)x = f \\ &\Leftrightarrow \quad (A \otimes I + I \otimes B)vec(X) = vec(F) \\ &\Leftrightarrow \quad (A \otimes I)vec(X) + (I \otimes B)vec(X) = vec(F) \\ &\Leftrightarrow \quad (IXA^T + BXI) = F \\ &\Leftrightarrow \quad BX + XA^T = F. \end{aligned}$$

□

4.2.2. **Lyapunov equation.**

Definition 4.3 ([8])**.** Lyapunov matrix equation is a particular case of the Sylvester matrix equation when B equals A. For square matrices $A, F \in \mathbb{R}^{n\times n}$, this matrix is written in the following form:

$$AX + XA^T = F. \tag{4.8}$$

Property 4.4. *If $T \in \mathbb{R}^{n^2 \times n^2}$ can be decomposed as KSGD ($T = A \otimes I + I \otimes A$) such that $A \in \mathbb{R}^{n \times n}$, then*

$$Tx = f \Leftrightarrow AX + XA^T = F,$$

where $X,\ F \in \mathbb{R}^{n \times n}$ and $x = vec(X), f = vec(F)$.

4.3. **The Conjugate Gradient algorithm in matrix format.**
According to Property 4.4 the linear system (4.4) becomes the following Lyapunov equation:

$$AX + XA^T = F, \tag{4.9}$$

where $x = vec(X)$ and $f = vec(F)$.

It is advantageous to use a matrix equation formulation [10] to use the conjugate gradient algorithm if n large. We define the matrices $X; R; P; F; T \in \mathbb{R}^{m \times m}$ by

$x = vec(X), r = vec(R), p = vec(P), d = vec(D)$ and $f = vec(F)$. Then

$$Tx = f \Leftrightarrow AX + XA = F$$

and $d = Tp \Leftrightarrow D = AP + PA$. This leads to the following algorithm.
We start with $X_0, P_0 = R_0 = F - (AX_0 + X_0A)$, the iteration is stopped when

$$\sqrt{\left(\sum_{i=1}^{m}\sum_{j=1}^{m}(R_{ij})_k * (R_{ij})_k\right) \Big/ \left(\sum_{i=1}^{m}\sum_{j=1}^{m}(F_{ij})_k * (F_{ij})_k\right)} \leq tol$$

Algorithm 5: Conjugate Gradient in matrix format (CG-M)

$$\begin{aligned}
D_k &= AP_k + P_kA; \\
\alpha_k &= \left(\sum_{i=1}^{m}\sum_{j=1}^{m}(R_{ij})_k * (R_{ij})_k\right) \Big/ \left(\sum_{i=1}^{m}\sum_{j=1}^{m}(P_{ij})_k * (D_{ij})_k\right); \\
X_{k+1} &= X_k + \alpha_k P_k; \\
R_{k+1} &= R_k - \alpha_k D_k; \\
\beta_k &= \left(\sum_{i=1}^{m}\sum_{j=1}^{m}(R_{ij})_{k+1} * (R_{ij})_{k+1}\right) \Big/ \left(\sum_{i=1}^{m}\sum_{j=1}^{m}(R_{ij})_k * (R_{ij})_k\right); \\
P_{k+1} &= R_{k+1} + \beta_k P_k.
\end{aligned}$$

Example 4.5. We try to solve the linear system (4.4) with the two algorithms CG and CG-M for various m.
The following table shows the difference in speed between CG and CG-M. We use $tol = 1e-8,\ f = [1, 1, ..., 1]^T$ and $x_0 = [0, 0, ..., 0]^T$.

Matrix size (m)	**CG and CGM with tol=1e-8**					
	# Iterations		Norm $\|\|Ax-b\|\|$		Time	
	CG	CG-M	CG	CG-M	CG	CG-M
40 000	369	369	1.9968e-06	6.6725e-07	0.4446	0.3459
160 000	734	734	3.8562e-06	1.1138e-06	4.7613	4.8170
360 000	1105	1105	5.9286e-06	1.6579e-06	15.9729	17.3569
640 000	1479	1479	7.7670e-06	2.1074e-06	44.2815	43.1813
1 000 000	1853	1853	9.8535e-06	2.6705e-06	94.3168	87.0503

TABLE 2. *The number of iterations and the time of execution for solving the Poisson problem on an $m \times m$ grid for various m* by CG method and CG-M method

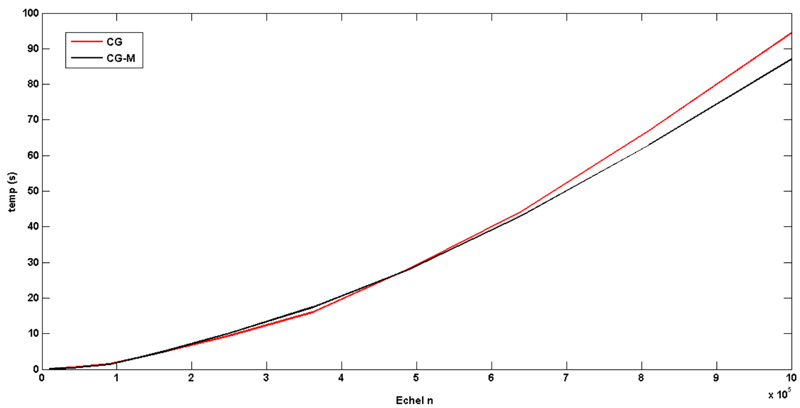

FIGURE 2. *Comparison of CG and CG-M methods in terms of execution time.*

5. CONCLUSION

Defining the Kronecker sum decomposition of matrices can be very useful for speeding up the solution of linear systems $Tx = f$ either by creating preconditioners or converting them into matrix equations.

6. ACKNOWLEDGEMENTS.

The authors wish to warmly thank the referees for all their useful suggestions.

REFERENCES

[1] C. F. Van Loan, The ubiquitous Kronecker product, Journal of Computational and Applied Mathematics 123 (2000) 85–100.

[2] S. Liu and G. Trenkler, Hadamard, Khatri-Rao, Kronecker And Other Matrix Products, International Journal of Information and Systems Sciences 4 (1) (2008) 160–177.

[3] P. A. Regaliat and S. K. Mitra, Kronecker products, unitary matrices and signal processing applications, Industrial and Applied Mathematics 31 (4) (1989) 586–613.
[4] D. S. Bernstein, Matrix Mathematics, Theory, Facts and Formulas, 2^{nd} edition, Princeton University Press 2009.
[5] E. M. Sadek, Méthodes itératives pour la résolution d'équations matricielles, Modélisation et simulation, Université du Littoral Côte d'Opale 2015.
[6] F. Liu, New Kronecker product decompositions and its applications, International Journal of Engineering and Science 1 (11) (2012) 25–30.
[7] H. Zhang and F. Ding, On the Kronecker products and their applications, Journal of Applied Mathematics 2013 (2013) Article ID 296185, 8 pages.
[8] E. Jarlebring, Numerical Methods for Lyapunov Equations, KTH Royal Institute of Technology 2017.
[9] B. N. Datta, Numerical Methods For Linear Control Systems Design And Analysis, Elsevier Academic Press 2003.
[10] T. Lyche, Numerical Linear Algebra and Matrix Factorizations, Texts in Computational Science and Engineering, Springer 2020.
[11] S. Banerjee and A. Roy, Linear algebra and matrix analysis for statistics, Texts in Statistical Science 2014.
[12] W. H. Steeb, Matrix Calculus and Kronecker Product with Applications and C++ Programs, International School of Scientific Computing 1997.
[13] A. J. Laub, Matrix Analysis for Scientists and Engineers, Industrial and App Math, SIAM Philadelphia 2005.
[14] V. Amoia, G. De Micheli and M. Santomauro, Computer-oriented formulation of transition-rate matrices via Kronecker algebra, IEEE Transactions on Reliability 30 (1981) 123–132.
[15] A. Graham, Kronecker Products and Matrix Calculus With Applications, Ellis Horwood Limited, Chichester 1981.
[16] R. A. Horn and C. R. Johnson, Topics in Matrix Analysis, Cambridge University Press, Cambridge 1991.
[17] Â. Björck, Numerical Methods in Matrix Computations, Texts in Applied Mathematics, Volume 59 2015.
[18] E. Kreyszig, Advanced Engineering Mathematics, John Wiley & Sons, Inc. 2011.
[19] MATLAB User's Guide, Version 8.1.0.604, The Math Works (R2013a).

Youssouf Mezzar (youcefmezzar@gmail.com)
Université Oran1, Laboratory of mathematics and its applications (LAMAP), P.O. Box 1524, Oran 31000, Algeria

Kacem Belghaba(belghaba@yahoo.fr)
Université Oran1, Laboratory of mathematics and its applications (LAMAP), P.O. Box 1524, Oran 31000, Algeria