universität wien

# MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

„Copositivity detection - a preprocessing study“

verfasst von / submitted by

Dejan Kuzmanovic, BA

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2022 / Vienna 2022

| | |
|---|---|
| Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet: | UA 066 915 |
| Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet: | Masterstudium Betriebswirtschaft |
| Betreut von / Supervisor: | Univ.-Prof. Mag. Dr. Immanuel Bomze |
| Mitbetreut von / Co-Supervisor: | |

# Contents

# List of Algorithms

**Abstract**

Due to increasing interest in copositive optimization, this paper aims to observe preprocessing steps that lead to a quick answer to whether a symmetric and quadratic matrix is copositive. Insights like these can help further narrow down the problem itself based on the approximation of the optimal solution. For the case of real $n \times n$ matrices for large $n$, the preprocessing steps presented in this thesis will lead to quick results and answer the question whether or not a matrix is copositive, not copositive or there is no answer possible. The presented algorithm has a time complexity of $O(n)$ and is able to provide quick results.

**Abstract**

Aufgrund des zunehmenden Interesses an kopositiver Optimierung zielt diese Arbeit darauf ab, Vorverarbeitungsschritte zu beobachten, die zu einer schnellen Antwort auf die Frage führen, ob eine symmetrische und quadratische Matrix kopositiv ist. Erkenntnisse wie diese können helfen, das Problem selbst basierend auf der Annäherung an die optimale Lösung weiter einzugrenzen. Für den Fall reeller $n \times n$ Matrizen für große $n$ führen die in dieser Arbeit vorgestellten Vorverarbeitungsschritte zu Antworten wie, kopositiv, nicht kopositiv oder keine Antwort möglich. Der vorgestellte Algorithmus hat eine Zeitkomplexität von $O(n)$ und ist in der Lage, schnelle Ergebnisse zu liefern.

# 1 Introduction and motivation

The work on copositive optimization is closely tied to conic optimization, which is again closely linked to minimizing a linear function over a convex cone [1]. Further research widens this approach to optimizing a quadratic form over a polyhedron given in the standard form[1] .

$$min\{x^\top Qx : Ax = b, x \in \mathbb{R}^n_+\}$$

The standard definition of copositivity is based on the definition of the quadratic form. It is also true that the transformation of a vector $x$ results in a different vector $y$. We also know that $x^\top y = \sum_{i=1}^{n} x_i y_i$ corresponds to the inner product in the Euclidean space $\mathbb{R}^n$. A real symmetric matrix of order $n$ is copositive if and only if its quadratic form $x \in \mathbb{R}^n \mapsto q_A(x) = x^\top Ax$ takes only nonnegative values on the nonnegative orthant $\mathbb{R}^n_+$ [2]. Let's now focus on the cone itself. It is possible to define arbitrary cones to speak of copositivity relative to the cone of choice. The concept of copositivity relative to ice cream cones in [3] and copositivity relative to polyhedral cones in [4] shows this concept. For the sake of this paper, the observed cone will be the cone described by $\{x \in \mathbb{R}^n_+ | x_i \geq 0\}$. Based on this cone, the results will be general instead of special cases.

## 1.1 State of research

Based on [5] and [6] the natural framework for the analysis of copositivity is the $\mathbb{S}^n$ linear space of real symmetric matrices of order n. $C_n = \{A \in \mathbb{S}^n : A \text{ is copositive}\}$ is the mathematical object of interest. The important thing to note which can be directly observed from the definition of $\mathbb{S}^n$ is that $C_n$ is non-polyhedral and it cannot be expressed as the intersection of finitely many closed half-spaces. This definition is closely related to the variational problem

$$\mu(A) = min_{x \geq 0, \|x\|=1} x^\top Ax.$$

The constraint $\|x\| = 1$ simplifies the otherwise large number of scaled vectors that do not change the result we aim to find. The key to understanding this statement is to focus on the linearity property of $\mathbb{S}^n$. The linearity in $\mathbb{S}^n$ states that every matrix $A \in \mathbb{S}^n$ must obey two properties. After the transformation, $Ax = y$ for all $x \in \mathbb{R}^n_+$ all lines must remain lines, and the origin must remain fixed [7]. Based on this interpretation it is true, that if a vector $x \in \mathbb{R}^n_+$ exists with $x^\top Ax < 0$ then a scalar multiple $y$ of this vector $x$ also satisfying $y^\top Ay < 0$ must be contained in $\Omega$, where $\Omega = \{x \in \mathbb{R}^n_+ : ||x|| = 0\}$. This understanding shows that the interpretation of copositivity is more a geometrical problem than a numerical one. The interpretation of copositivity based on vector geometry and orientation and based on the definition of the variational problem, shows that there are many more fields of research in copositivity detection to be explored.

# 2 Copositivity

## 2.1 Further Definitions and Interpretations

At the beginning of this section, I will first display the general definition for copositivity for future reference. This section will have multiple definitions, remarks, and theorems that will be significant in the later work. These will help understand the algorithm used for copositivity detection in real symmetric matrices of order $i = 5, ..., 20$. The reader can find every proof or reference to the respective evidence in the source, which is stated after each definition, remark, or theorem.

**Definition 2.1** *Let $A$ be a real symmetric matrix of order $n$. One says that $A$ is copositive if its associated quadratic form $x \in \mathbb{R}^n \mapsto q_A(x) = x^\top Ax$ takes only nonnegative values on the nonnegative orthant $\mathbb{R}^n_+$ [2]. It is also defined that a copositive matrix $A$ is copositive-plus if $x \geq 0$ and $x^\top Ax = 0$ imply $Ax = 0$ [8].*

Although there will be no difference in the results in this paper, meaning a matrix is either copositive or not. It is nevertheless of some significance to point out exceptional cases of copositivity.

**Remark 2.1** *If $A$ is copositive, so is any positive multiple of $A$, any positive principal permutation of $A$, and any principal sub-matrix of $A$ [8].*

**Theorem 2.1** *$A = A^\top \in \mathbb{S}^n$ is not copositive if and only if it contains a non-singular principal sub-matrix $D$ such that a column of $D^{-1} \leq 0$ [8].*

**Theorem 2.2** *Let $A$ be a symmetric matrix of order $n$ such that each principal sub-matrix of order $(n-1)$ is copositive. Then*

$$A \text{ is copositive} \iff det(A) \geq 0 \text{ or } adjA \text{ contains a negative entry. [9, 10]}$$

**Theorem 2.3** *Let $A$ be a symmetric matrix of order $n$ such that each principal sub-matrix of order $(n-1)$ is copositive. Then*

$$A \text{ is not copositive} \iff A^{-1} \text{ exists and is non-positive entrywise. [11]}$$

In a later section of this paper, I will propose an algorithm that underlines the need for theorem 2.1, 2.2 and 2.3. At this point, it is possible to approach a better understanding of what copositivity means in general. As briefly discussed in section 1, we can think of the $Ax$ part of $x^\top Ax$ as a linear transformation of $x$, since $A \in \mathbb{S}^n$. In this regard a vector $x \in \mathbb{R}^n_+$ is transformed into another vector $y \in \mathbb{R}^n$. Since the only real information we have about the matrix $A$ is that it is symmetrical and real of order n, it is not possible to determine in which part of $\mathbb{R}^n$ the vector $y$ will land. From now on, the cone that is equal to the domain space of $x \in \mathbb{R}^n_+$ will be denoted as $\Gamma$ [2].

## 2.2 Span and copositivity

There is a good definition for copositivity relative to $\Gamma$, which leads to the question of what to do with it? It is vital to think of matrices as linear transformations in this context [12]. We can state that the most interesting linear transformation regarding this subject is the one that provides a vector $y = Ax$, which has at least one negative component.

**Proposition 2.1** *The vector $y = Ax$ with only positive components implies that the vector $x$ cannot be a violating vector in terms of copositivity of the matrix $A$ such that*

$$x^\top Ax < 0$$

.

For simplicity, imagine a two-dimensional area in a two-dimensional Cartesian coordinate system. The linear transformation A is applied to the entire area $A(\Gamma)$. Therefore, regarding copositivity, the only parts of $A(\Gamma)$ which are of significance are the ones that yield vectors $y \in A(\Gamma)$ with at least one negative component. Again it becomes apparent that it is not the vector $x \in \mathbb{R}^n_+$ itself, which is violating for the matrix $A$ but the orientation of the linear transformation $Ax = y$.

## 2.3 Copositivity and vector orientation

When discussing vector orientations regarding other vectors, we must mention the angle between two vectors at all costs. The angle between two vectors is defined as follows:

**Definition 2.2** *Let $a \in \mathbb{R}^n$ and $b \in \mathbb{R}^n$ be two vectors with real components, then the dot product of the two vectors is defined as*

$$a^\top b = \|a\|\|b\| \cos(\phi)$$

*or in other words, the dot product describes the length of the projection of $a$ onto $b$ times the length of $b$. [13]*

If now denoted as $x^\top Ax = x^\top y = \|x\|\|y\| \cos(\phi)$ it can be seen that the only part which can lead to a negative result and therefore certify that $x$ is a violating vector according to its definition is the $\cos(\phi)$ term. Since the dot product is equipped with the commutative property[13], the same is also true for

$$y^\top x = \|y\|\|x\| \cos(\phi).$$

As the angle $\phi$ stays the same, all the focus can shift to the term $\cos(\phi)$.

**Remark 2.2** *Remember that $\cos(\phi)$ is $2\pi$ periodic and $\cos(\phi) > 0$ for $0 < \phi < \pi/2$, $\cos(\phi) = 0$ for $\phi = \pi/2$ and $\cos(\phi) < 0$ for $\pi/2 < \phi < 3\pi/2$.[13]*

Therefore the only interesting vectors $y = Ax$ and $x$ are the ones whose angle is $\pi/2 < \phi < 3\pi/2$ where $\phi$ denotes the angle between $y = Ax$ and $x$. Since $x \in \mathbb{R}^n_+$ has only positive components by definition, a vector $x$ can only be a violating vector for the matrix $A$ if and only if $A$ rotates $x$ by $\pi/2 < \phi < 3\pi/2$. In other words a matrix is copositive if and only if its linear transformation of any vector $x \in \mathbb{R}^n_+$ does not rotate the vector by an angle $\phi$ which is $\pi/2 < \phi < 3\pi/2$.

## 2.4 Copositive or positive-semidefinite

Let's again look at the definition of copositivity and compare it with positive semi-definiteness. Compare definition 2.1 with the following statement.

**Definition 2.3** *A real symmetric matrix $B$ is said to be positive semi definite if $x^\top Bx \geq 0$ for all $x \in \mathbb{R}^n$.*

We can observe that the only difference between the two definitions is the domain set of the vector $x$, which implies that remark 2.2 still holds even in this case, and there is no need to compare those two cases further. It only remains to show that this approach is valid for all domain sets $\mathbb{R}^n_+$ with $n \in \mathbb{N}/\{0\}$. A span of two vectors is the set of all possible linear combinations of such two vectors in their two-dimensional plane. Based on this understanding, it holds that the angle between two vectors is always calculated in the same way as shown above.

## 2.5 Approaches to copositivity detection

The causes of non-copositivity in matrices have been discussed in the previous pages. Yet, this approach still needs a procedure that eliminates the possibility of a matrix being copositive. In the earlier sections, thoughts and considerations of vector orientation in the domain set of $\mathbb{R}^n_+$ are the cause for the copositivity of a matrix. Quickly checking if a matrix can be copositive or not is to observe the same domain set on all of its points and transform it under the linear transformation described by the matrix. To understand what happens to the domain set of $x$ under the linear transformation, we need to calculate the determinant of the matrix representing the linear transformation [12]. To illustrate this, observe the copositivity test of Cottle-Habetler-Lemke in [9].

Recall 2.2. Let us observe an example from the same paper, namely:

$$A = \begin{bmatrix} 1 & \sqrt{2} \\ \sqrt{2} & 1 \end{bmatrix} A^{-1} = \begin{bmatrix} -1 & \sqrt{2} \\ \sqrt{2} & -1 \end{bmatrix}$$

Both matrices have the same determinant of $det(A) = det(A^{-1}) = -1$ although $A$ is copositive and its inverse is not. This means that not a single vector $x \in \mathbb{R}^n_+$ is transformed into another vector $y$ in the exterior of $\mathbb{R}^n_+$ and therefore the angle $\phi$ between no two vectors is $\pi/2 < \phi < 3\pi/2$. The same is not true for the inverse of $A$. If observed on its own, the basis vector in the $x$ axis direction rotates by an angle of $\phi > \pi/2$ under the transformation imposed by the matrix $A$. It, therefore, represents a violating vector for $A^{-1}$ according to the definition. It remains important to note that A does not have any more principal sub-matrices which could be positive, so theorem 2.2 fails in this example. Although interesting, theorem 2.2 is complex to handle in copositivity detection since it picks up the inversion of space and scaling of domain sets. In other words, the only possible way for a matrix $A$ to be non-copositive is for it to transform a vector $x \in \Gamma$ in such a way that the angle of $x$ and $y = Ax$ is $\pi/2 < \phi < 3\pi/2$.

## 2.6 Completely-positive matrices and applications

The difference between positive-semidefinite and copositvive matrices lies in the domain set of the vector transformed by the matrix. The difference between copositive and completely-positive matrices is the matrix itself. According to [14] completely-positive matrices can be defined as follows:

**Definition 2.4** *A matrix is completely-positive if it can be decomposed as* $A = BB^\top$, *where* $B$ *is a (not necessarily square) non-negative matrix.*

According to [14], it is easy to construct a completely-positive matrix, but it is hard to determine whether a given square matrix is completely-positive. The matrix needs to be doubly non-negative, which means both positive semidefinite and entrywise non-negative. The problem is that this condition is not always sufficient. [14] proposes the following:

**Proposition 2.2** *A is completely-positive if and only if A can be represented as a sum*

$$A = \sum_{i=1}^{k} b_i b_i^\top \ , \ b \in \mathbb{R}_+^n \ , \ i = 1, ..., k.$$

Furthermore, to establish a connection to the conic optimization problem the canonical conic program is given by:

$$min_{x \in \mathbb{R}_+^n} c^\top x$$

$$Ax - b = 0$$

$$x \geq 0$$

Since the domain set of vector $x$ for the completely-positive matrices is the entire $\mathbb{R}^n$, the cone K can be any cone in $\mathbb{R}^n$. The dual cone is defined in [15] as follows.

**Definition 2.5** *Let* $K$ *be a cone. The set*

$$K^* = \{y | x^\top y \geq 0 \ \forall x \in K\}$$

*is called the dual cone of* $K$.

As the focus of this paper lies in copositivity detection and the domain set of $x$ in $x^\top Ax$ is the non-negative orthant $\mathbb{R}^n_+$ with $|x| = 1$, the dual cone of $\mathbb{R}^n_+$ is $\mathbb{R}^n_+$ itself. A cone that abides by this property is called self-dual.

As discussed in [16] the dual cone $K^*$ of $K-$copositive matrices is defined as follows:

**Definition 2.6** *Let $K \subset \mathbb{R}^n$ be a closed convex cone and let $C_K$ be the convex cone of $K-$copositive matrices, i.e., $A \in \mathbb{S}^n$ such that $x^\top Ax \geq 0$ on $K$. Its dual cone $C^*_K \subset \mathbb{S}^n$ is the cone of $K-$completely-positive matrices.*

Furthermore [14] states:

**Definition 2.7** *The cone of completely copositive matrices $CP_n$ and the cone of copositive matrices $COP_n$ are dual cones in the space $\mathbb{S}^n$. That is $CP^*_n = COP_n$ and $COP^*_n = CP_n$.*

Studies are most often done on the conic programming problem of the form:

$$\text{Minimize } c^\top x$$

$$\text{subject to } Ax = y, x \in K \text{ [17]}$$

Where $K \subset \mathbb{R}^n$ is a pointed closed convex cone with a non-empty interior. Based on this understanding and according to [18], the conic dual optimization problem can be written as follows.

$$\sup b^\top y$$

$$\text{s.t. } c - \textstyle\sum_i y_i a_i \in K^*$$

$$y \in \mathbb{R}^n$$

Hence a feasible solution to the dual conic problem yields a lower bound to the primal conic optimization problem.

# 3 Preprocessing

In the previous section, copositiviy as an idea and concept was discussed and defined as vector orientation challenges. When talking about detection, it is often helpful to think about preprocessing itself, which could lead to quick results. This approach avoids using complex and protracted algorithms to find out if a matrix is copositive or not. Preprocessing can yield three possible results.

1. Copositive (with the reason given)
2. Not copositive
    - Violating vector
3. No-answer possible

Preprocessing aims to clarify the first two cases, and a copositivity detection algorithm for the third case.

## 3.1 Certificates

In this section, the sign tests from [19], will be discussed.

1. if $A_{ii} < 0$, then $v = e_i$ is a violating vector.

2. if $A_{ii} = 0 > A_{ij}$, then $v = (A_{jj} + 1)e_i - A_{ij}e_j$ is a violating vector.

3. if $A_{ij} \geq 0$ for all $j$, then $A$ is copositive iff $R = [\, A]_{\, i,j\neq i}$ is copositive;
   $u = [\, u_j]_{\, j\neq i}$ violating for $R \Rightarrow v = [\, 0, u] \in \mathbb{R}^n_+$ violating for A.

4. if $A_{ij} \leq 0 < A_{ii}$ for all $j \neq i$, then A is copositive iff
   $$T = [\, A_{ii}A_{jk} - A_{ij}A_{ik}]_{\, j,k\neq i} \text{ is copositive.}$$
   $w = [\, w_j]_{\, j\neq i}$ violating for $T \Rightarrow v = [-\sum_{j\neq i} A_{ij}w_j, A_{ii}w] \in \mathbb{R}^n_+$ violating for $A$.

5. if $A_{ij} < -\sqrt{A_{ii}A_{jj}} < 0$, then $v = \sqrt{A_{jj}}e_i + \sqrt{A_{ii}}e_j$ is violating.

6. $A$ is copositive if and only if
   $$A' = [\, \tfrac{A_{ij}}{\sqrt{A_{ii}A_{jj}}}] \text{ is copositive.}$$
   we have $diagA' = e$

Every certificate yields a violating vector for the matrix it is tested against, if it meets the stated requirements. Testing these requirements is an operation that must be performed for every matrix at least once. In the case of 3, 4 and 6 possibly multiple times. Case 6 produces a different matrix with $A_{ii} = 1$ and fractions as every other element. Following the definitions of the requirements comes their explanation and connection to the vector orientation that has been explored in section 2. It is helpful to understand the linear transformation being executed in every single case.

If any diagonal element is negative with $A_{ii} < 0$ it means that the corresponding canonical basis vector $e_i$ is scaled by some value $A_{ii} < 0$ in the negative direction of $e_i$ that by definition yields a violating vector. The rotation angle of this violating vector is $\pi/2 < \phi < 3\pi/2$, and therefore, according to its definition, the dot product must be negative. case 2 can be represented with a simple matrix.

$$A = \begin{bmatrix} 0 & -1 \\ -1 & 1 \end{bmatrix}$$

As the rows and columns represent the new canonical basis vectors, what happens with the domain space is clear. The $\mathbb{R}^n_+$ is rotated and stretched. Therefore only a linear combination of the canonical basis vectors can produce a violating vector. Following this rule, the violating vector for the matrix $A$ is $v = \begin{bmatrix} 1 & 1 \end{bmatrix}$.
For case 3, the results prove to be far more interesting. Assuming the matrix $A \in \mathbb{S}^n$ with $n > 2$ and $A$ contains a row with only positive entries. The domain space of a possible violating vector x is $x \in \mathbb{R}^n_+$. Since the matrix, $A$ transforms the vector $x$ into another vector y where $y \in \mathbb{R}^n$ the domain space can also be restricted to $\mathbb{R}^{(n-1)}_+$, where a certain row and column of matrix $A$ is deleted. This results in another matrix $A' \in \mathbb{S}^{(n-1)}$. In the case of $n = 3$, the domain space is restricted to the plane $\mathbb{R}^2_+$ like in the example above. As a vector $x \in \mathbb{R}^2_+$ will be transformed into another vector $y \in \mathbb{R}^2$ there is no need for the third dimension of the domain space even to be considered if $x \in \mathbb{R}^2_+$ is a violating vector for $A$.

**Remark 3.1** *Note that under case 3, it is stated that the violating vector for the original domain space is given by* $v = [\,0, u\,] \in \mathbb{R}^n_+$ *which is obviously only correct if* $i = 1$. *For this proposition to be valid, one must note that every column and row index* $i$ *deleted in the matrix* $A$ *represents the indices of the zero elements in the violating vector. The violating vector combines zeroes in all elements with a noted index and the violating vector of the final reduced matrix.*

The following matrix provides an example of remark 3.1

$$A = \begin{bmatrix} 0 & -1 & 1 \\ -1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \text{ with the violating vector } \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

This statement is true since the image space of the vector $x \in \mathbb{R}^2$ under the linear transformation, which is defined by the $(2 \times 2)$ matrix $A$ has the same dimension as the domain space of $x \in \mathbb{R}^2$. Therefore the vector $x \in \mathbb{R}^2 \subset \mathbb{R}^3$ does not care about additional dimensions since we are only interested in its orientation and not its length. therefore the angle between $x \in \mathbb{R}^2$ and $Ax \in \mathbb{R}^2$ is similar to the angle between $x \in \mathbb{R}^3$ and $Ax \in \mathbb{R}^3$. This statement is true since $Ax \in \mathbb{R}^2$ and $Ax \in \mathbb{R}^3$ are generally located in the same orthant. Therefore there must be an angle $\phi < \pi/2$ between both vectors, where $Ax \in \mathbb{R}^2$ is extended by an additional 0 element like discussed above. With this approach, we know that $x \in \mathbb{R}^2_+ \subset \mathbb{R}^3$ and that $y \in \mathbb{R}^3$ is located in another orthant of the image space. The angle between $x$ and $y$ is $\phi > \pi/2$ and therefore, the dot product must be negative. Based on the definition of basis vectors in $\mathbb{R}^n$, we know that basis vectors are pairwise orthogonal. Therefore this understanding can be expanded to general $\mathbb{R}^n_+$ domain sets. The dot product keeps its properties for $\mathbb{R}^n$.

## 3.2 No-answer possible

Six cases can produce a violating vector to summarize the previous certificate section. In the first case, the matrix has a negative diagonal element $A_{ii} < 0$ that implies that, if the matrix reduces to a $(2 \times 2)$ matrix, one of the basis vectors rotates into another quadrant. Therefore this same vector must be a violating vector. A matrix $A$ obeys the second certificate test has a zero element in the diagonal that is also larger than another element in the same row. The violating vector in this case is also clearly defined as $v = (A_{ii} + 1)e_i - A_{ij}e_j$, or simply $v = e_i - A_{ij}e_j$. The third case has been discussed in the previous section to the full extent and also provides a violating vector if the matrix has columns that only hold positive or zero elements. The fourth case is more complex in its application. In this case, the matrix needs to hold a positive diagonal element while at the same time having negative off-diagonal elements in the same row.

This can be easily seen with a negative sign matrix where $N_{ij} = 1$ if $A_{ij} < 0$ and $N_{ij} = 0$ otherwise. The column and row with the most non-zero elements are removed, and the matrix $T$ is calculated as described in the enumeration in subsection 3.1. The row and column reduction are performed as often as non-zero elements in the matrix $N$ exist. This case appears especially problematic as it does not always lead to a violating vector for the initial matrix A.

Suppose the problem starts with the matrix A that through reduction of row and column $i = 2$ and matrix operations described in case 4 becomes the matrix T:

$$A = \begin{bmatrix} 2 & -3 & 5 \\ -3 & 1 & -2 \\ 5 & -2 & 2 \end{bmatrix} \quad T = \begin{bmatrix} -7 & -1 \\ -1 & -2 \end{bmatrix}$$

A violating vector $w$ for matrix $T$ is obviously one of the two basis vectors $e_i$ (in this case let's settle for $e_1 = (1, 0)$).

**Proposition 3.1** *Based on these propositions, a symmetrical real matrix $A$ is not copositive if this procedure can find a violating vector based on one of the six criteria. $A$ is copositive if no principal sub-matrix of second-order transforms the basis vectors of $\mathbb{R}^2$ into another quadrant other than $\mathbb{R}^2_+$. No answer is possible otherwise. Remember that these propositions are connected with the preprocessing part of matrices and not with the copositive optimization.*

The final certificate that is being used to determine if a matrix is copositive or not is the case 6, where the matrix is transformed into $A' = [\frac{A_{ij}}{\sqrt{A_{ii}A_{jj}}}]$ where all the diagonal elements are one elements. This matrix is used again to test for the previously described certificates.

## 3.3 Additional certificates

The following lemma in [20] a non-copositive matrix is similar to this paper's earlier parts.

**Lemma 3.1** *Let $X \in \mathbb{S}^n$ such that either $n = 1$ or $X_{[1:n]/i} \in COP^{(n-1)}$ for all $i \in [1 : n]$. Then the following are equivalent:*

1. $X \notin COP^n$
2. *$X$ is non-singular and $-X^{-1} \in N^n$ where $N = \mathbb{S}^n \cap \mathbb{R}_+^{n \times n}$*
3. $\forall b \in \mathbb{R}_+^n, \exists u \in \mathbb{R}_+^n$ *with* $Xu = -b$
4. $\exists u \in \mathbb{R}_+^n$ *such that* $Xu = -1_n$
5. $\exists u \in \mathbb{R}_+^n$ *such that* $-Xu \in \mathbb{R}_{++}^n$

Lemma 3.1 statement 2 is equivalent to theorem 2.3 in [21]. Lemma 3.1 statement 2 suggests that the negative of the inverse of $X$ is symmetric with only positive entries. This statement is according to [21] only true for matrices in which all of the $(n-1)$ principal sub-matrices are copositive. A geometrical interpretation of lemma 3.1 provides further insight into this situation. Since we know that the columns of matrix $A$ are the new basis vectors according to which the vector $x \in \mathbb{R}_+^m$ will be transformed. The following example shows what happens to the vector, for simplicity reasons, the matrix $A \in \mathbb{S}^2$ with

$$A = \begin{bmatrix} -2 & 1 \\ 1 & -2 \end{bmatrix} A^{-1} = 1/3 * \begin{bmatrix} -2 & -1 \\ -1 & -2 \end{bmatrix}$$

A is not copositive since the copositivity test 1. yields a violating vector. Although the determinant of $A$ is positive, it is because $A$ describes a rotation of $\mathbb{R}_+^n$ by an angle $\phi$ with $\pi/2 \leq \phi \leq 3\pi/2$. Therefore the transformation, which inverts this rotation, must be a matrix with non-positive components.

# 4 Preprocessing algorithm

This section will provide an algorithm and its explanation and results for testing copositivity in real symmetric matrices as described in the previous sections of this paper. The algorithm aims to deliver quick results with answers specified in section 3 and is swift in eliminating matrices that fall into the earlier categories. The cases which yield a no-answer possible are the cases that require further judgment. The algorithm is written in $C\#$ [22] under .NET Framework 4.8 and will be available as an attachment to this paper and a plain text document with 100000 examples.

## 4.1 The input matrix

The first challenge when dealing with algorithms that use matrices as input is the input itself. Simple tests with small matrices are done with the already discussed examples, but for the actual input, the matrix has to obey some rules, most notably:

1. $A \in \mathbb{S}^n$.

2. order of $A$ is $5 \leq n \leq 20$

For this, I have first created a matrix of a random order $A$ is $5 \leq n \leq 20$ and filled it with random elements which are given by

**Algorithm 1** Create symmetric input matrix

**Require:** $5 \leq n \leq 20$

$A_{ij} \leftarrow random(0, 60) - random(0, 20)$

$n \leftarrow random(5, 20)$

$dim(A) = n$

**for** $i, j$ **do**

  **if** $j > i$ **then**

    $A_{j,i} = A_{i,j}$

  **else if** $j \leq i$ **then**

    $A_{i,j} = A_{i,j}$

  **end if**

**end for**

**return** A

Algorithm 1 is the root of every other operation in the rest of the code. Every time the complete preprocessing algorithm executes until the end, algorithm 1 is rerun to ensure that a different input matrix is generated by following the same rules. One certificate has to be checked before testing for other certificates. Every function requires that there are no negative diagonal elements. Hence if there is a negative diagonal element, the algorithm stops execution immediately and returns a violating vector.

## 4.2 Preprocessing

The next test performed can be found in subsection 3.1 under index 2.

**Algorithm 2** Check for zero diagonal elements

**Require:** $A \leftarrow$ Algorithm 1 (initial)

**Require:** $dim(v) = n$

**for** $\forall i$ **do**

**if** $A_{ii} < 0$ **then return** $v = e_i$

**end if**

**end for**

**for** i,j **do**

**if** $A_{i,i} = 0$ & $A_{j,i} < 0$ **then**

$v = (A_{jj} + 1)e_i - A_{ij}e_j$

**end if**

**end for**

**if** $v$ is not empty **then return** v

**else if** $v$ is empty **then**

**if** Step $A_{i,j} \geq 0 : \forall\ j$ **then**

**if** $i \neq empty$ **then** $A \leftarrow$ RemoveRow(i)

**if** $v$ is violating for $A$ **then** $u_i = v_i$ & $u_j = 0$ **return** u

**else if** $i = empty$ & $v = empty$ **then** Algorithm 3

**else if** $v = empty$ & $i \neq empty$ **then** Repeat Step

**end if**

**end if**

**end if**

**end if**

**Algorithm 3** Remove Row i

**Require:** $index \leftarrow$ Algorithm 2

**Require:** $A \leftarrow$ Algorithm 1 (initial)

**for** $\forall i, j$ **do**

**if** $i \neq index$ & $j \neq index$ **then** $A_{i-1j-1} = A_{ij}$ **return** Algorithm 4

**end if**

**end for**

Algorithm 4 corresponds to case 4 from subsection 3.1. The main objective here is first to verify that matrix $A$ holds one row where the diagonal element is strictly positive with $A_{ii} > 0$. In contrast, the off-diagonal elements are given by $A_{ij} \leq 0$. As already discussed, this method's result depends on the matrix and does not always provide a violating vector, as shown in subsection 3.2.

**Algorithm 4** Positivity test and copositivity test 4

**for** $\forall i, j, k$ **do**

**if** $A_{ii} > 0$ & $A_{ij} \leq 0$ & $i \neq j$ **then** $T_{jk} = A_{jk} + A_{ik}$

**for** $\forall i$ **do**

**if** $T_{ii} < 0$ **then** $w = e_i$

**end if**

**if** $A_{ij} = -1$ & $T_{jk} < 0$ **then** $w = (2 + A_{ik})e_j - T_{jk}e_k$

**end if**

**if** $T_{ij} < -\sqrt{T_{ii}T_{jj}} < 0$ **then** $w = \sqrt{T_{jj}}e_i + \sqrt{T_{ii}}e_j$

**end if**

**end for**

**end if**

**if** $w \neq empty$ **then return** $v_{k=i} = -\sum_{j \neq i} A_{ij}w_j$ & $v_{k \neq i} = [\, A_{ii}w]$

**end if**

**if** $v = empty$ **then return** SpectralPr. & DiagonalDominance & Algorithm 5

**end if**

**end for**

The final part of the main algorithm has to do with the case 6 from subsection 3.1, where the entire input matrix is transformed according to $A' = [\frac{A_{ij}}{\sqrt{A_{ii}A_{jj}}}]$.

**Algorithm 5** Copositivity test 6

**Require:** $A \leftarrow$ Algorithm 1 (initial)

**Require:** $A' = [\frac{A_{ij}}{\sqrt{A_{ii}A_{jj}}}]$

**for** $i, j$ **do**

**if** $A'_{i,i} = 0$ & $A'_{j,i} < 0$ **then**

$v = (A'_{ii} + 1)e_i - A'_{ij}e_j$

**end if**

**if** $A'_{i,j} \geq 0 : \forall\ j$ & $i \neq empty$ **then** $A' \leftarrow$ Algorithm 2

**end if**

**if** $A_{ii} > 0$ & $A_{ij} \leq 0$ & $i \neq j$ **then return** Algorithm 4

**if** $w \neq empty$ **then return** $v_{k=i} = -\sum_{j \neq i} A_{ij}w_j$ & $v_{k \neq i} = [A_{ii}w]$

**end if**

**end if**

**end for**

Since A is from now on normalized by algorithm 5, it only has to be proven that $A_{ij} < -1$ since $A_{ii} = 1 : \forall i$. Therefore a violating vector is not $v = \sqrt{A_{jj}}e_i + \sqrt{A_{ii}}e_j$ but the reduced form $v = e_i + e_j$.

**Algorithm 6** Check negative square roots of diagonal elements

**Require:** $A \leftarrow A'$ (Algorithm 5)

**for** $\forall i, j$ **do**

**if** $A_{ij} < -1$ **then** $v = e_i + e_j$

**end if**

**if** $v \neq empty$ **then return** $v$

**end if**

**if** $v = empty$ **then return** null

**end if**

**end for**

The last part of the main algorithm ends with the multiplication of every matrix element by the product of the square root of the corresponding diagonal elements, which in the case of diagonal element results in $A'_{ii} = 1$. This new matrix is then regarded as the new input matrix and all of the previous operations are repeated with $A' = [\frac{A_{ij}}{\sqrt{A_{ii}A_{jj}}}]$.

Remember, remark 2.1 states that if A is copositive, so is any positive multiple of A, any positive principal permutation of A, and any principal sub-matrix of A [8]. Since a matrix that describes a linear transformation takes the entire cone of input vectors $x \in \mathbb{R}^n_+$ and either widen it or narrows it down, depending on the linear transformation [12], every positive multiple of that same linear transformation does the same. The exciting part, which is also observed by the algorithm, is the statement that any principal sub-matrix of A must also be copositive if A is copositive. Algorithm 3 yields that same result, where rows and columns are removed if a condition is met. Even when removing any row and the corresponding column from that matrix, it will stay copositive nonetheless. The circumstances that can be solved by applying the discussed calculations are still not sufficient to find an answer in every case. Therefore the algorithm uses theorem 2.1, 2.2 and 2.3 to find matrices that have yielded no result until this point in the algorithm.

## 4.3 Spectral preprocessing

Since in some cases, the algorithm returns a no-answer possible result or tries to categorize a matrix as copositive based on the criteria specified above. It is also necessary to check for hard proof for the copositivity of the observed matrix. Based on the findings in [7] and [8] it is known that a positive-semidefinite matrix is also copositive by definition. In [19] the eigenpairs $\lambda_i, u_i$ with $u_i^\top u_i = 1$ are used to find a proof that a matrix is indeed positive-semidefinite with the following lemma:

**Lemma 4.1** *Denote by* $\lambda_1 \leq ... \leq \lambda_k \leq \lambda_{k+1} \leq ... \leq \lambda_n$ *the ordered eigenvalues of the matrix A.*

1. *If* $\lambda_1 \geq 0$ *then A is positive-semidefinite and hence copositive.*
2. *If* $\lambda_1 < -\lambda_n$ *then at least one of the two vectors* $u_1^+$ *or* $u_1^-$ *is violating.*
3. *If* $\lambda_1 = -\lambda_n$ *and if there is no eigenvector of* $\lambda_n$ *in* $\mathbb{R}^n_+$*, then at least one of the two vectors* $u_1^+$ *or* $u_1^-$ *is violating.*

Although examples, where case 2 and 3 are triggered are harder to find on purpose, the first case can quickly test the algorithm's accuracy.

$$\text{Let A} = \begin{bmatrix} 7 & 2 \\ 2 & 1 \end{bmatrix}$$

Since I am using the Math.Net library to compute the eigenvalues and eigenvectors, the eigenvalues are denoted in a complex number format. This part of the algorithm is triggered after every row reduction. The algorithm yields the following results: $\lambda_1 = 0.394449$ and $\lambda_2 = 7.60555$. The columns of the matrix U are the eigenvectors of the matrix A.

$$\text{U} = \begin{bmatrix} 0.289784 & 0.957092 \\ -0.957092 & 0.289784 \end{bmatrix}$$

Since $\lambda_1 \geq 0$ the matrix A is positive-semidefinite and therefore copositive. This part of the algorithm is executed on every input matrix, and its results are discussed in subsection 5.2.

Additionally after the definition of the input matrix and after each row reduction step, the algorithm will check for diagonal dominance by setting all the positive entries to zero and calculating the difference between the diagonal entry and the sum of all negative off-diagonal entries. Diagonal dominance implies that the input matrix is positive semidefinite and therefore will return a flag, that will mark the matrix as copositive.

## 4.4 Testing

In this subsection, the testing part of the algorithm is discussed briefly. It is needed to indicate if a copositivity method yields a violating vector when executed correctly and if the input matrix meets a criterion for the certificates in subsection 3.1. Based on types and type conversions, it is not always possible to straightforward combine different elements in the algorithm. These parts can be found in the code itself and will not be discussed in detail in this paper.

**Algorithm 7** Vector Matrix Multiplication

**Require:** $A \neq empty$

**Require:** $v \neq empty$

**Require:** $result \neq empty$

**for** $\forall i, j$ **do**

$result_i = [\sum_j A_{ij} v_j]$

**end for**

**return** DotProduct(v,result) = Algorithm 8

**Algorithm 8** Dot Product

**for** $\forall i$ **do**

product $= \sum_i v_i * result_i$

**end for**

**if** $product < 0$ **then return** Matrix is not copositive

**end if**

**if** $product \geq 0$ **then return** No violating vector found

**end if**

The testing part of the algorithm is invoked every time another part of the algorithm yields a violating vector. Every violating vector is stored in a separate list accessed at the end of every process to retrieve the count of violating vectors produced for a given input matrix. As long as some operation is realizable, the algorithm will keep executing it until it can find a result or until the resulting matrix does not meet any of the criteria statet in 3.1.

# 5 Examples and Results

In this section I will present my results and some examples from other papers for comparison. Also, as an attachment to this paper, there will be three text documents with 100000 example matrices that have been tested by my proposed algorithm, that also notes the total number of copositive, non-copositive, and no-answer cases.

## 5.1 Examples

The first matrix which will be inserted into the algorithm is the matrix

$$\mathrm{A} = \begin{bmatrix} 2 & -3 & 5 \\ -3 & 1 & -2 \\ 5 & -2 & 2 \end{bmatrix}$$

With the help of the slightly changed algorithm the result is the following series of calculations:

$$\mathrm{N} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \mathrm{T} = \begin{bmatrix} -7 & -1 \\ -1 & -2 \end{bmatrix} \quad \mathrm{w} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Based on the criterion $v_{k=i} = -\sum_{j \neq i} A_{ij} w_j$ & $v_{k \neq i} = [A_{ii} w]$, the violating vector for the matrix A is given by: $v = (1, 3, 0)$ with $v^\top A v = -7$ where $i = 2$.

The next example is the matrix:

$$\mathrm{A} = \begin{bmatrix} 0 & -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Since $A_{00} = 0 > A_{01}$ this example falls under the case 2 and 3. The matrix is first reduced by the first non-negative row and column it can find once, while the algorithm tests the other certificates before trying the reduction step again. After the first reduction the algorithm yields the violating vector $v = (2, 1, 0, 0, 0)$ with the result $v^\top Av = -3$.

The next matrix which is inserted into the algorithm is

$$\mathrm{A} = \begin{bmatrix} 1 & -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 \end{bmatrix}$$

also known as the Horn matrix [23]. Based on [23] this matrix is copositive but cannot be decomposed into the form $A = A_1 + A_2$ where $A \in C_n = P_n + N_n$. The algorithm yields a no-answer result in every case. Since there is no preprocessing step, that would reduce number of rows or columns it is not possible to determine the copositivity of the matrix in this algorithm. The final example which I will display is the following matrix:

$$\mathrm{A} = \begin{bmatrix} 32 & 5 & 9 & 29 & 43 \\ 5 & 0 & 21 & -12 & 40 \\ 9 & 21 & 13 & 45 & 14 \\ 29 & -12 & 45 & 20 & 20 \\ 43 & 40 & 14 & 20 & 50 \end{bmatrix}$$

This matrix falls under Algorithm 2, where if $A_{ii} = 0 > A_{ij}$, then $v = (A_{jj}+1)e_i - A_{ij}e_j$. The violating vector for this case is given by $v = (0, 21, 0, 12, 0)$ with the result:

$$v^\top Av = -3168$$

Other much larger matrices fall under other certificates but will not be displayed here since they would require a large amount of space. Since the attachment consists of many more examples with violating vectors resulting from one or more of the cases stated in subsection 3.1, the reader can find other interesting examples.

## 5.2 Final results and findings

As already discussed earlier, the input for the algorithm was on 100'000 randomly generated real symmetric matrices $A \in \mathbb{S}^n$ with $n \in [5, 20]$. Each time a matrix meets one of the criteria from subsection 3.1 or an algorithm from section 4 yields a non-empty result, a violating vector is found with $v^\top Av < 0$. Based on the random nature of the matrices, it is often possible that a matrix meets multiple criteria at once and therefore has various violating vectors, of which one is denoted in the notCopositiveResults.txt file to reduce redundancy. The total number of violating vectors and results are presented below and at the bottom of the notCopositiveResults.txt file.

| Copositivity Results | | | |
|---|---|---|---|
| Certificate | Copositive | Not Copositive | no-answer |
| | 5988 | 91162 | 2850 |
| Case 1 | | 85006 | |
| Case 2 | | 1458 | |
| Case 3 | | 474 | |
| Case 4 | | 0 | |
| Case 5 | | 4224 | |
| Spectral Pre. | | 0 | |

The number of results for cases like 4 and the spectral preprocessing case appears to have no results. This is the case because the given matrix must also fall under another certificate checked first and displayed in the NotCopositiveResults.txt file. I have described the functionality in subsection 4.3 for better understanding.

| Rows Deleted | |
|---|---|
| Case 3 | Case 4 |
| 129365 | 13798 |

# 6 Conclusion

To sum up the results of my work, Every cycle of the algorithm alters 2 flags that are members of every input matrix. One is the copositivity of type boolean and the other is an array of integers that denotes the violating vector. The algorithms for Case 3 and 4 can set the values for the copositivity flag and also return a violating vector. The three possible outcomes are:

1. Copositive if no violating vector can be found as well as if Case 3 and 4 from subsection 3.1 can set the copositivity flag of a matrix to true. This is the case if the reduced matrix is flagged as copositive either by diagonal dominance or spectral preprocessing.

2. Not copositive if the algorithm can find a violating vector.

3. no-answer if no violating vector can be found and the copositivity flag is set to false.

In this case, the algorithm can sort out input matrices that are not positive semidefinite, nor is there a violating vector found.

| Copositivity Results | | |
|---|---|---|
| Copositive | Not Copositive | no-answer |
| 5988 | 91162 | 2850 |

A unique violating vector is displayed next to the matrix in the notCopositiveResults.txt file for every matrix. These examples require no further attention. It is also important to note that I have written each of the algorithms as a stand-alone method, that can be excluded, included, and expanded when needed without breaking the program. In the future, the algorithm can also be extended by other methods if this should be required. The results in the copositiveResults.txt file have been marked as copositive while the results in the noAnswerResults.txt file are not solvable by this algorithm. I have included a percentage of deleted rows and columns by Case 3 and 4 as well as the total number of deleted rows at the end of the output files. The matrices are sorted by size in all output files.

# 7 References


[1] Immanuel M. Bomze. Copositive optimization – Recent developments and applications. *European Journal of Operational Research*, 216(3):509–520, 2012.

[2] Immanuel M Bomze, Mirjam Dür, Etienne De Klerk, Cornelis Roos, Arie J Quist, and Tamás Terlaky. On copositive programming and standard quadratic optimization problems. *Journal of Global Optimization*, 18(4):301–320, 2000.

[3] Miroslav Fiedler. Positivity with respect to the round cone. *Matematický časopis*, 24(2):155–159, 1974.

[4] D.H. Martin and D.H. Jacobson. Copositive matrices and definiteness of quadratic forms subject to homogeneous linear inequality constraints. *Linear Algebra and its Applications*, 35:227–258, 1981.

[5] Marshall Hall Jr. Combinatorial theory, blaisdell publ. *Co., Waltham Mass*, 1967.

[6] Kh D Ikramov and NV Savel'Eva. Conditionally definite matrices. *Journal of Mathematical Sciences*, 98(1):1–50, 2000.

[7] Tilo Arens, Frank Hettlich, Christian Karpfinger, Ulrich Kockelkorn, Klaus Lichtenegger, and Hellmuth Stachel. *Mathematik*. Spektrum Akademischer Verlag, 1. aufl. 2008 edition, 2 2008.

[8] Hannu Väliaho. Criteria for copositive matrices. *Linear Algebra and its Applications*, 81:19–34, 1986.

[9] Richard W Cottle, GJ Habetler, and Carlton E Lemke. Quadratic forms semi-definite over convex cones. Technical report, STANFORD UNIV CALIF OPERATIONS RESEARCH HOUSE, 1967.

[10] Richard W Cottle, GJ Habetler, and CE Lemke. On classes of copositive matrices. *Linear Algebra and Its Applications*, 3(3):295–310, 1970.

[11] Karl-Peter Hadeler. On copositive matrices. *Linear Algebra and its Applications*, 49:79–89, 1983.

[12] S. Bosch. *Lineare Algebra*. Springer-Lehrbuch. Springer, 2003.

[13] L. Papula. *Mathematische Formelsammlung: Für Ingenieure und Naturwissenschaftler*. Springer Fachmedien Wiesbaden, 2017.

[14] A. Berman and N. Shaked-monderer. *Completely Positive Matrices*. World Scientific Publishing Company, 2003.

[15] François Glineur. Conic optimization: An elegant framework for convex optimization, 2001.

[16] Jean Bernard Lasserre. New approximations for the cone of copositive matrices and its dual, 2012.

[17] Akihiro Tanaka and Akiko Yoshise. An lp-based algorithm to test copositivity. *Pacific Journal of Optimization*, 11:101–120, 01 2015.

[18] Julia Witzel. *Some aspects of the optimization over the copositive and the completely positive cone*. doctoralthesis, Universität Trier, 2014.

[19] Immanuel M. Bomze and Gabriele Eichfelder. Copositivity detection by difference-of-convex decomposition and $\omega$-subdivision. *Math. Program.*, 138(1-2):365–400, 2013.

[20] Peter J.C. Dickinson. A new certificate for copositivity. *Linear Algebra and its Applications*, 569:15–37, 2019.

[21] K.P. Hadeler. On copositive matrices. *Linear Algebra and its Applications*, 49:79–89, 1983.

[22] M.J. Price. *C# 9 and .NET 5 – Modern Cross-Platform Development - Fifth Edition: Build intelligent apps, websites, and services with Blazor, ASP.NET Core, and Entity Framework Core using Visual Studio Code*. Packt Publishing, 2020.

[23] Palahenedi Hewage Diananda. On non-negative forms in real variables some or all of which are non-negative. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 58, pages 17–25. Cambridge University Press, 1962.

# 8 Links

1. Results: https://www.dropbox.com/scl/fo/ps7mlsgrlivrrgp2g1xyk/h?dl=0&rlkey=yshpabeqsersi9e64jm6ttp5u

2. Code: https://github.com/Inferno29/CopositivityDetectionAlgorithm