VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Netradiční výpočty zobecněných inverzí matic

Unconventional computations of generalized inverses

2015 Jan Pacholek

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra aplikované matematiky

Zadání bakalářské práce

Student: **Jan Pacholek**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 1103R031 Výpočetní matematika

Téma: Netradiční výpočty zobecněných inverzí matic
Unconventional computations of generalized inverses

Zásady pro vypracování:

Zobecněné inverze se objevují v mnoha technických úlohách, kdy klasická inverze neexistuje. Matice jsou přitom často velmi rozsáhlé, takže je potřeba věnovat náležitou pozornost výběru vhodného výpočetního algoritmu. Cílem této práce je otestovat některé metody výpočtu zobecněných inverzí, které využívají algoritmy pro výpočet inverzí regulárních matic. Hlavní myšlenka spočívá v tom, že ze zadané singulární matice se vhodným způsobem zkonstruuje matice regulární a to taková, aby její inverze nebo část její inverze byla zobecněnou inverzí původní matice. Při výpočtech pak lze použít například Gaussovu eliminační metodu.

Generalized inverses arise in many technical problems, in which the classical inverse does not exist. Usually, the matrices are very large so that the choice of an appropriate computational algorithm is crucial. The aim of this work is to test some of the methods for computing generalized inverses that are based on algorithms for computing inverses of regular matrices. The main idea is that the new matrix is defined so that its inverse or a block of its inverse is the generalized inverse of the original matrix. In computations one can use for instance Gaussian elimination.

Seznam doporučené odborné literatury:

[1] A. Ben-Israel, T. Greville: Generalized inverses: theory and ppplications. (2nd edn) Springer, New York, 2003.

[2] R. Kučera, T. Kozubek, A. Markopoulos: On large-scale generalized inverses in solving two-by-two block linear systems. Linear Algebra and Its Applications (2012).

[3] N. Shinozaki, M. Sibuya, K. Tanabe: Numerical algorithms for the Moore-Penrose inverse of a matrix: Direct methods, Ann. Inst. Statist. Math. 24 (1972) 193-203.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. RNDr. Radek Kučera, Ph.D.**

Datum zadání: 01.09.2014

Datum odevzdání: 07.05.2015

VYSOKÁ ŠKOLA BÁŇSKÁ-TECHNICKÁ UNIVERZITA
Fakulta elektrotechniky a informatiky
* OSTRAVA *

Bouchala

doc. RNDr. Jiří Bouchala, Ph.D.
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 *Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.*

V Ostravě 7. Května 2015 .... Jan Pach

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 7. Května 2015 .... Jan Pach

Rád bych na tomto místě poděkoval Doc. RNDr. Radku Kučerovi,Ph.D., vedoucímu mé bakalářské práce, za cenné rady, připomínky, za čas strávený při konzultacích a odkazy na zdroje informací a bez něhož by tato práce nevznikla.

Abstrakt

V tomto textu se budeme zabývat inverzními maticemi a jak takové matice spočítat. Poté, pomocí podobného algoritmu, zkusíme spočítat zobecněné inverze matic, které jsou regulární. Budeme hledat způsob, kterým převedeme singulární matici na regulární, abychom mohli spočítat její inverzi a poté jak najít blok takové inverze, který bude právě onou zobecněnou inverzí. Další věcí kterou zde zmíníme je aplikace zobecněné inverze a její využití při řešení Stouksova problému.

Klíčová slova: zobecněná inverze, Moore-Penroseova pseudoinverze, Stouksův problém, inverzní matice

Abstract

This text deals with inverse matrices, where we discuss some of the ways how to calculate an inverse to a matrix. Then, using a similar algorithm, we try to calculate generalized inverses of matrices that are singular. We will find a way, how to make such matrix singular, so that it's inverse can be calculated and then how a block of this inverse is the generalized inverse. Another thing we are going to discuss is the application of generalized inverses in solving the Stokes problem.

Keywords: generalized inverse, Moore-Penrose pseudoinverse, the Stokes problem, inverse matrix

List of used abbreviations and symbols

$\mathbb{R}$	–	set of real numbers
$\mathbb{R}^n$	–	vector space of n-dimensional vectors
$\mathbb{R}^{m\times n}$	–	space of matrices of the type $m \times n$
$\mathcal{R}(A)$	–	range-space of $A \in \mathbb{R}^{m\times n}$, i.e. $\mathcal{R}(A) = \{x \in \mathbb{R}^m : x = Ay,\ y \in \mathbb{R}^n\}$
$\mathcal{N}(A)$	–	null-space of $A \in \mathbb{R}^{m\times n}$, i.e. $\mathcal{N}(A) = \{x \in \mathbb{R}^n : Ax = 0\}$
I	–	identity matrix
0	–	zero matrix, zero vector, or zero
MP	–	Moore-Penrose pseudoinverse
ChD	–	Cholesky decomposition
GChD	–	generalized Cholesky decomposition

Contents

List of Tables

List of Figures

List of Source Codes

1 Introduction

In this thesis we are going to deal with the generalized inverses of matrices. Based on some of the algorithms that are used for calculating the inverses of regular matrices, we will try to find a similar algorithm to calculate the generalized inverse. We are going to compare different types of the generalized inverses and different ways how to calculate them.

In section 2 we will briefly remind the reader what the definition of an inverse matrix is, what is a non-singular and singular matrix and show an example of calculating an inverse matrix. In section 3 we are going to define the generalized inverse, show got to calculate them using the singular value decomposition and the QR decomposition.

Section 4 deals with the Moore-Penrose (MP) pseudoinverse, which is one kind of the generalized pseudoinverse, with more strict properties. Using the null-space, we will show how to turn a singular matrix into a non-singular one and, later in the section, we will calculate an example and show how to calculate the MP pseudoinverse, using a similar algorithm to calculating regular matrices. We will also mention a method to calculate the action of MP pseudoinverse using the Cholesky decomposition, which we will later implement in MATLAB and use to compute the actual MP pseudoinverse itself, comparing it to other methods of calculations.

In the last section we will mention the application of the generalized inverse in the Stokes problem. For the sake of completness and so that the reader can see, where the generalized inverses are used, we will show how the Stokes problem, using the TFETI approximation, is converted into system of linear equations and then compare several types of the generalized inverses used to solve the system. By the end we should know, what method of calculating the generalized inverse is the most effective, requires the least iterations, and is the most precise.

2 Inverse matrix

Definition 2.1 *Let A be a square matrix. If there exists a matrix B such that*

$$AB = BA = I, \tag{1}$$

then B is called the inverse to A. A square matrix, that has an inverse, is called non-singular. If a matrix does not have any inverse, then it is called singular.

Theorem 2.1 *For any nonsingular matrix A, there is an unique inverse matrix.*

Proof. Let A be nonsingular and B, C be the inverse matrices to A such that

$$AB = I \text{ and } AC = I.$$

From that follows

$$B = BI = B(AC) = (BA)C = IC = C.$$

We have shown that $B = C$, meaning that A has the unique inverse matrix. ■

We will denote the inverse matrix to A as A^{-1}.

Example 2.1 We have the given matrix:

$$A = \begin{pmatrix} 2 & -3 & 1 \\ 1 & 2 & -1 \\ 2 & 1 & 1 \end{pmatrix}.$$

Our aim is to find it's inverse A^{-1}. To calculate the inverse, we will use the Gauss elimination with the identity matrix on the right side. By the end of the calculations, we will have our matrix A replaced by the identity matrix and on the right by the inverse (for a step-by-step solution, see Appendix A. So, the inverse to A is

$$A^{-1} = \frac{1}{12}\begin{bmatrix} 3 & 4 & 1 \\ -3 & 0 & 3 \\ 3 & 8 & -7 \end{bmatrix}.$$

To prove, that A^{-1} is truly an inverse matrix, it must satisfy (1). To do that, we would have to calculate both equations mentioned in (1).

■

3 Generalized inverses

We have defined the inverse matrix to the square, non-singular matrix in the previous section. But there exist generalized inverses to singular matrices and to matrices that are not square. We are going to talk about these generalized inverses more in this section.

Definition 3.1 *Let A be an arbitrary matrix. The generalized inverse to A is a matrix X such that*

$$AXA = A. \tag{2}$$

If $A \in \mathbb{R}^{m\times n}$, then $X \in \mathbb{R}^{n\times m}$.

Theorem 3.1 *The equality* (2) *is the neccessary and sufficient condition guaranting that*

$$x^0 = Xb$$

is the solution to any consistent linear system $Ax = b$, i.e the system with $b \in \mathcal{R}(A)$.

Proof. First, we prove the sufficiency. Let (2) hold and $b = A\hat{x}$. Then

$$Ax^0 = AXb = AXA\hat{x} = A\hat{x} = b.$$

Now we prove the necessity. Assume that $x^0 = Xb$ solves $Ax = b$, where $b = A\hat{x}$. Then

$$A\hat{x} = b = Ax^0 = AXb = AXA\hat{x}.$$

We get $(A - AXA)\hat{x} = 0$ for any $\hat{x}$. Using the unit vectors $e_1, e_2, ..., e_n$ as $\hat{x}$, we get (2). ■

Unlike inverse matrices, generalized inverses exist for any matrix, but they are not unique. We are going to discuss several generalized inverses to A.

Definition 3.2 *The singular value decomposition to $A \in \mathbb{R}^{m\times n}$ is given as follows:*

$$A = V\hat{\Sigma}U^T,$$

where $V \in \mathbb{R}^{m\times m}$ and $U \in \mathbb{R}^{n\times n}$ are orthogonal and

$$\hat{\Sigma} = \begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{m\times n},$$

where $\Sigma = diag(\sigma_1, \sigma_2, ..., \sigma_s)$ and $\sigma_i > 0, i = 1, ..., s$ are the singular values of A.

Remark 3.1 If $s = m$ or $s = n$, then the last block row or block column in $\hat{\Sigma}$, respectively, is omitted.

Theorem 3.2 *The matrix $X = U\tilde{\Sigma}V^T$ with*

$$\tilde{\Sigma} = \begin{pmatrix} \Sigma^{-1} & K \\ L & M \end{pmatrix}$$

is the generalized inverse to A with arbitrary matrices K, L and M of an appropriate type.

Proof. Using (2), we get the following equalities:

$$AXA = V\hat{\Sigma}U^TU\tilde{\Sigma}V^TV\hat{\Sigma}U^T = V\hat{\Sigma}\tilde{\Sigma}\hat{\Sigma}U^T = A,$$

since

$$\hat{\Sigma}\tilde{\Sigma}\hat{\Sigma} = \begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \Sigma^{-1} & K \\ L & M \end{pmatrix} \begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix} = \hat{\Sigma}.$$

■

Generalized inverses may also be computed by the orthogonal (QR) decomposition of a matrix. For the sake of simplicity, we restrict ourselves to the simplest variant of the QR decomposition. The other variants are more involved, but principaly analogous. Note that the QR decomposition is the Gram-Schmidt orthogonalization process on the matrix level.

Definition 3.3 *Let $A \in \mathbb{R}^{m\times n}$ be a full column rank matix with $m \geq n$. The QR decomposition of A is given as follows:*

$$A = QR,$$

where $Q \in \mathbb{R}^{m\times n}$ is orthogonal and $R \in \mathbb{R}^{n\times n}$ is upper triangular and non-singular.

Theorem 3.3 *The matrix $X = R^{-1}Q^T$ is the generalized inverse to A.*

Proof. Recall that $Q^TQ = I$. We get

$$AXA = QRR^{-1}Q^TQR = QR = A.$$

■

4 The Moore-Penrose pseudoinverse

The Moore-Penrose (MP) pseudoinverse is a special kind of the generalized inverse. Unlike generalized inverses, the MP pseudoinverse is unique for each matrix. In this and the following sections, we are going to discuss some of the methods used to calculate the MP pseudoinverse and it's applications.

Definition 4.1 *Let $A \in \mathbb{R}^{m\times n}$ and $X \in \mathbb{R}^{n\times m}$. The matrix X is called the MP pseudoinverse to A, if the following conditions hold:*

$$AXA = A, \qquad XAX = X, \qquad (AX)^T = AX, \qquad (XA)^T = XA.$$

Theorem 4.1 *For any matrix $A \in \mathbb{R}^{m\times n}$, there exists a unique matrix X, which satisfies Definition 4.1.*

Proof. See [2]. ■

The MP pseudoinverse is denoted by A^+. The following lemma is introduced for completness of our theory.

Lemma 4.2 *Let $A \in \mathbb{R}^{m\times n}$, $M \in \mathbb{R}^{m\times l(A^T)}$, and $N \in \mathbb{R}^{n\times l(A)}$, where $l(A^T) = dim\, \mathcal{N}(A^T)$ and $l(A) = dim\, \mathcal{N}(A)$, be given and let $R_{A^T}^T M, R_A^T N$ be nonsingular, where $R_{A^T} \in \mathbb{R}^{m\times l(A^T)}$ and $R_A \in \mathbb{R}^{n\times l(A)}$ are full rank matrices whose columns span $\mathcal{N}(A^T)$ and $\mathcal{N}(A)$, respectively. Then*

$$P_M = I - M(R_{A^T}^T M)^{-1} R_{A^T}^T, \qquad P_N = I - N(R_A^T N)^{-1} R_A^T$$

are the projectors for which:

$$\begin{aligned}
\mathcal{R}(P_M) &= \mathcal{R}(A),\\
\mathcal{N}(P_M) &= \mathcal{R}(M),\\
\mathcal{N}(P_M^T) &= \mathcal{N}(A^T),\\
\mathcal{R}(P_M^T) &= \mathcal{N}(M^T),\\
\mathcal{R}(P_N) &= \mathcal{R}(A^T),\\
\mathcal{N}(P_N) &= \mathcal{R}(N),\\
\mathcal{N}(P_N^T) &= \mathcal{N}(A),\\
\mathcal{R}(P_N^T) &= \mathcal{N}(N^T).
\end{aligned}$$

Proof. See [2] ■

Theorem 4.3 *Let $M, N \in \mathbb{R}^{n\times l}$, where l is the defect of $A \in \mathbb{R}^{n\times n}$, be such that $R_{A^T}^T M, R_A^T N$ are nonsingular, where $R_{A^T}, R_A \in \mathbb{R}^{n\times l}$ are full rank matrices whose columns span the null-space of A and A^T respectively. Let us consider the following problem: find the pair $(\bar{u}, \bar{\lambda}) \in \mathbb{R}^n \times \mathbb{R}^l$ solving*

$$\begin{pmatrix} A & M \\ N^T & 0 \end{pmatrix} \begin{pmatrix} u \\ \lambda \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}, \tag{3}$$

where $b \in \mathbb{R}^n$. Then

$$\bar{u} = X_{M,N} b,$$

where $X_{M,N}$ is the generalized inverse to A, with the range-space equal to the null-space of N^T and the null-space equal to the range-space of M.

Proof. First we prove that the matrix in (3) is nonsingular. To this end we search for the solution (u_0, λ_0) to the homogeneous system:

$$Au_0 + M\lambda_0 = 0, \tag{4}$$

$$N^T u_0 = 0. \tag{5}$$

Multiplying (4) by $R_{A^T}^T$, we get $R_{A^T}^T M\lambda_0 = 0$ implying $\lambda_0 = 0$. Then (4) yields u_0 belongs to the null-space of A so that $u_0 = R_A\alpha_0$ for an appropriate $\alpha_0 \in \mathbb{R}^l$. Substituting this result into (5), we arrive at $N^T R_A \alpha_0 = 0$ that gives $\alpha_0 = 0$. Consequently we get $u_0 = 0$ so that the nonsingularity is proved.

The first equation in (3) implies $\lambda = (R_{A^T}^T M)^{-1} R_{A^T}^T b$ and then

$$A\bar{u} = (I - M(R_{A^T}^T M)^{-1} R_{A^T}^T) b = P_M b.$$

Therefore

$$\bar{u} = XP_M b + R_A \bar{\alpha}, \tag{6}$$

where X is an arbitrary generalized inverse to A and $\bar{\alpha} \in \mathbb{R}^l$. The second equation in (3) implies $\bar{\alpha} = -(N^T R_A)^{-1} N^T X P_M b$. Substituting this result in (6), we get $\bar{u} = (I - M(R_{A^T}^T M)^{-1} N^T) X P_M b = P_N^T X P_M b$. The proof is complete. ∎

Remark 4.1 If $A \in \mathbb{R}^{n\times n}$ is symmetric, i.e. $A = A^T$, then $R_{A^T} = R_A = R$. Moreover, if we put $M = N = R$, we get the MP pseudoinverse via (3). Let K denote the matrix from (3). Since K is nonsingular, one can calculate it's standard inverse K^{-1}. The block in K^{-1} corresponding to A in K is the MP pseudoinverse (see Figure 1).

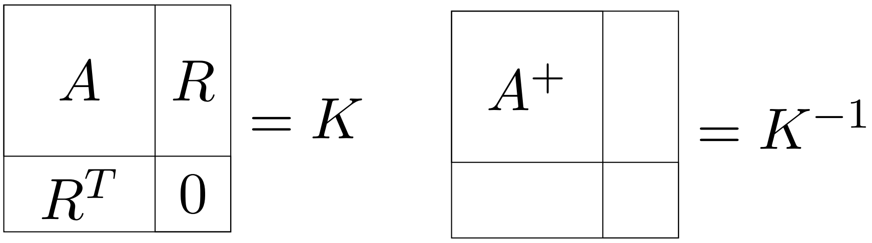

Figure 1: MP pseudoinverse in a block matrix

Example 4.1 We have the given matrix

$$A = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}.$$

Using the knowledge from previous sections, we will calculate it's MP pseudoinverse. First, we need to create the matrix K. It is easy to see, that

$$R = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix},$$

since $AR = 0$. Knowing this, we can create

$$K = \left(\begin{array}{cccc|c} 1 & -1 & 0 & 0 & 1 \\ -1 & 2 & -1 & 0 & 1 \\ 0 & -1 & 2 & -1 & 1 \\ 0 & 0 & -1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 0 \end{array}\right),$$

which gives us a nonsingular matrix. Using the Gauss elimination and extracting the respective block from the final matrix, we arrive at the following pseudoinverse (for step-by-step solution, see Appendix B):

$$A^+ = \begin{pmatrix} \frac{7}{8} & \frac{1}{8} & -\frac{3}{8} & -\frac{5}{8} \\ \frac{1}{8} & \frac{3}{8} & -\frac{1}{8} & -\frac{3}{8} \\ -\frac{3}{8} & -\frac{1}{8} & \frac{3}{8} & \frac{1}{8} \\ -\frac{5}{8} & -\frac{3}{8} & \frac{1}{8} & \frac{7}{8} \end{pmatrix},$$

which satisfies all properties of Definition 4.1. ■

4.1 Pseudoinverse based on the generalized Cholesky decomposition

In order to work with the Cholesky decomposition, we need to define positive definite and positive semi-definite matrices.

Definition 4.2 *A matrix $A \in \mathbb{R}^{n\times n}$ is called positive definite, if*

$$x^T Ax > 0 \quad \forall x \in \mathbb{R}^n,\, x \neq 0.$$

Definition 4.3 *A matrix $A \in \mathbb{R}^{n\times n}$ is called positive semi-definite, if*

$$x^T Ax \geq 0 \quad \forall x \in \mathbb{R}^n.$$

Now that we defined what a positive definite and positive semi-definite matrix is, we can define what the Cholesky decomposition is.

Definition 4.4 *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric, positive definite matrix. The Cholesky decomposition (ChD) of A is given as follows:*

$$A = U^T U,$$

where $U \in \mathbb{R}^{n\times n}$ is upper triangular with positive diagonal entries.

Definition 4.5 *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric, positive semi-definite, and singular matrix. The generalized Cholesky decomposition (GChD) of A is given as follows:*

$$A = U^T U, \tag{7}$$

where $U \in \mathbb{R}^{n\times n}$ is upper triangular with non-negative diagonal entries.

Let $1 \leq l < n$ be the defect of A. Then U from (7) has exactly l zeros on its diagonal and, moreover, all entries in the respective rows are zeros as well. Using an appropriate permutation, the zero diagonal entries can be located at the end of the diagonal of U. For the sake of simplicity we will use this modification of the GChD below.

If we have a symmetric positive semi-definite and singular matrix A, we can calculate it's pseudoinverse using the GChD. We suppose that

$$A = \begin{pmatrix} U_1^T & \\ U_2^T & 0 \end{pmatrix} \begin{pmatrix} U_1 & U_2 \\ & 0 \end{pmatrix} = \begin{pmatrix} U_1^T U_1 & U_1^T U_2 \\ U_2^T U_1 & U_2^T U_2 \end{pmatrix},$$

where $U_1 \in \mathbb{R}^{(n-l)\times(n-l)}$ is upper triangular and non-singular and $U_2 \in \mathbb{R}^{(n-l)\times l}$.

Lemma 4.4 *The matrix $R \in \mathbb{R}^{n\times l}$, whose columns are the linearly independent basis of $\mathcal{N}(A)$ is given as follows:*

$$R = \begin{pmatrix} -U_1^{-1} U_2 \\ I \end{pmatrix}.$$

Proof. We get

$$AR = \begin{pmatrix} U_1^T U_1 & U_1^T U_2 \\ U_2^T U_1 & U_2^T U_2 \end{pmatrix} \begin{pmatrix} -U_1^{-1} U_2 \\ I \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Furthermore,

$$R^T R = U_2 U_1^{-T} U_1^{-1} U_2 + I$$

is symmetric, positive definite and, hence, non-singular. Therefore, the column-rank of R is equal to l. ■

These results used in (3) lead to the following system:

$$\begin{pmatrix} U_1^T U_1 & U_1^T U_2 & -U_1^{-1} U_2 \\ U_2^T U_1 & U_2^T U_2 & I \\ -U_2^T U_1^{-T} & I & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \lambda \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ 0 \end{pmatrix}, \tag{8}$$

with which we are going to work with further and ultimately reach a solution to calculating the pseudoinverse.

From the third row and the second row in (8), it is easy to see that

$$x_2 = U_2^T U_1^{-T} \tag{9}$$

and

$$\lambda = b_2 - U_2^T U_1 x_1 - U_2^T U_2 x_2, \tag{10}$$

respectively. Now, if we express the first row from (8), we arrive at the following equation:

$$U_1^T U_1 x_1 + U_1^T U_2 x_2 + U_1^{-1} U_2 U_2^T U_1 x_1 + U_1^{-1} U_2 U_2^T U_2 x_2 = b_1 + U_1^{-1} U_2 b_2.$$

As a next step, we re-organize the equation to make it easier to work with

$$(U_1^T U_1 + U_1^{-1} U_2 U_2^T U_1) x_1 + (U_1^T U_2 + U_1^{-1} U_2 U_2^T U_2) x_2 = b_1 + U_1^{-1} U_2 b_2.$$

Now we substitute x_2 from (9), re-organize the equation again and after extracting x_1, we arrive at:

$$(U_1^T U_1 + U_1^{-1} U_2 U_2^T U_1) x_1 + (U_1^T U_2 + U_1^{-1} U_2 U_2^T U_2) U_2^T U_1^{-T} x_1 = b_1 + U_1^{-1} U_2 b_2,$$

which changes into

$$(U_1^T U_1 + U_1^{-1} U_2 U_2^T U_1 + U_1^T U_2 U_2^T U_1^{-T} + U_1^{-1} U_2 U_2^T U_2 U_2^T U_1^{-T}) x_1 = b_1 + U_1^{-1} U_2 b_2. \tag{11}$$

After extracting U_1^T and $U_1^{-1} U_2 U_2^T$ from (11), we receive

$$[U_1^T (U_1 + U_2 U_2^T U_1^{-T}) + U_1^{-1} U_2 U_2^T (U_1 + U_2 U_2^T U_1^{-T})] x_1 = b_1 + U_1^{-1} U_2 b_2,$$

which can be re-written as

$$(U_1^T + U_1^{-1} U_2 U_2^T)(U_1 + U_2 U_2^T U_1^{-T}) x_1 = b_1 + U_1^{-1} U_2 b_2 \tag{12}$$

making it less confusing. Last step is to simply multiply both sides of (12) with U_1 from left, which will give us this final equation that we can further work with

$$(U_1U_1^T + U_2U_2^T)(U_1U_1^T + U_2U_2^T)U_1^{-T}x_1 = U_1b_1 + U_2b_2. \tag{13}$$

To help us solve (13), we use the substitution $y_1 = (U_1U_1^T + U_2U_2^T)U_1^{-T}x_1$ and solve the following system of linear equations

$$(U_1U_1^T + U_2U_2^T)y_1 = U_1b_1 + U_2b_2, \tag{14}$$

which is easy to compute, since $U_1U_1^T + U_2U_2^T$ will give us a positive definite square matrix, meaning it can be solved by using the usual Cholesky decomposition. After we have found our y_1, we can use another substitution of $z_1 = U_1^{-T}x_1$ and solve another system of linear equations, with the same left side as before

$$(U_1U_1^T + U_2U_2^T)z_1 = y_1. \tag{15}$$

Then we need to only calculate x_1 and x_2, where

$$x_1 = U_1^T z_1,$$
$$x_2 = U_2^T z_1.$$

The process that has been described is called the action of the MP pseudoinverse matrix. We divide it into two algorithmic schemes. In the first one, we create all the necessary matrices from A.

ALGORITHMIC SCHEME 1 (PREPROCESSING):

```
Input : $A \in \mathbb{R}^{n\times n}$ symmetric, positive semi-definite, and singular.
Step 1: Compute the GChD $A = U^TU$.
Step 2: Identify in $U$ the blocks $U_1$ and $U_2$.
Step 3: Compute ChD $U_1U_1^T + U_2U_2^T = CC^T$.
Output : $U_1 \in \mathbb{R}^{(n-l)\times(n-l)}$, $U_2 \in \mathbb{R}^{(n-l)\times l}$, $C \in \mathbb{R}^{(n-l)\times(n-l)}$.
```

Now, that we have created all the necessary matrices, we can continue with the rest of the algorithm, where we get to the iterative part, which is the main purpose for creating the matrices U_1, U_2, and C beforehand.

ALGORITHMIC SCHEME 2 (ACTION):

```
Input : $U_1 \in \mathbb{R}^{(n-l)\times(n-l)}$, $U_2 \in \mathbb{R}^{(n-l)\times l}$, $C \in \mathbb{R}^{(n-l)\times(n-l)}$, $b \in \mathbb{R}^n$.
Step 1: Split $b$ such that $b = (b_1^T, b_2^T)^T$, where $b_1 \in \mathbb{R}^{n-l}$ and $b_2 \in \mathbb{R}^l$.
Step 2: $y_1 = C^{-T}(C^{-1}(U_1b_1 + U_2b_2))$.
Step 3: $z_1 = C^{-T}(C^{-1}y_1)$.
Step 4: $x = (z_1^TU_1, z_1^TU_2)^T$.
Output : $x = A^+b \in \mathbb{R}^n$.
```

Remark 4.2 The presented algorithm is a variant of the rank decomposition method (see [3]).

4.2 Implementation in MATLAB

While we implement our functions in MATLAB, we consider completely general situations including permutations.

The first thing that we need to do, is to compute U of the GChD using the LU decomposition. In order to do that, we convert A to a sparse matrix first. This prevents the permutation of rows caused by the LU decomposition. Now, that we have U, we divide each row by the square root of its diagonal entry, giving us the new form of U from GChD.

The next step is to identify the blocks U_1 and U_2. To identify the block matrix U_1, we can use the indeces of non-zero values on the diagonal of U, while using the indeces of zero values to identify the block matrix U_2, respectively. The last step of our function is to compute the ChD of $U_1U_1^T + U_2U_2^T$.

```
function [U1,U2,C,ind] = preprocessing (A)
%Step1
A = sparse(A);
[L,U] = lu(A,0); d = diag(U);
for i = 1:numel(d)
    if abs(d(i)) <= 1e−7
        d(i) = 1/sqrt(d(i));
    else d(i) = 0;
    end
end
U = diag(d)*U;
%Step2
ind = diag(U)~= 0;
U1 = U(ind,ind);
U2 = U(ind,~ind);
%Step3
C = chol(U1*U1'+U2*U2','upper');
```

Code 1: The generalized Cholesky decomposition - MATLAB function

Now, all we have to do is to split the vector b appropriately and solve the systems of linear equations.

```
function x = action(U1,U2,C,ind,b)
%Step1
b1 = b(ind); b2 = b(~ind);
%Step2
y1 = C\((U1*b1+U2*b2)'/C)';
%Step3
z1 = C\(y1'/C)';
%Step4
x1 = (z1'*U1)'; x2 = (z1'*U2)';
x(ind) = x1; x(~ind) = x2;
```

Code 2: The action of the MP pseudoinverse - MATLAB function

4.3 Numerical experiments

The following table shows the accuracy of our MP pseudoinverse calculated by Code 1 and Code 2. The MP pseudoinverses are assembled by our functions applied on the columns of the identity matrix. The first column of the tables represents the size of matrix A, which was randomly generated. The remaining columns represent the accuracy for each condition of the MP pseudoinverse mentioned in Definiton 4.1.

From the first table we can see that the accuracy of our calculations is satisfying even for bigger matrices. We need to mention that our calculations are affected by the round errors, which cause inaccuracies while checking the fulfillment of the MP pseudoinverse defining equalities.

Size of A	$AA^+A = A$	$A^+AA^+ = A^+$	$(AA^+)^T = AA^+$	$(A^+A)^T = A^+A$
5	2.04e-14	4.34e-13	2.68e-14	2.60e-14
10	5.84e-15	2.99e-14	5.51e-15	4.99e-15
20	8.43e-16	6.81e-17	1.92e-16	1.89e-16
40	1.47e-14	6.17e-15	3.02e-15	3.05e-15
80	4.23e-14	1.53e-15	3.62e-15	3.62e-15
160	9.69e-14	7.01e-16	1.97e-15	1.97e-15
320	9.60e-13	1.53e-14	2.32e-14	2.32e-14
640	1.56e-11	6.65e-15	1.01e-13	1.01e-13
1280	2.72e-13	2.09e-16	4.93e-15	4.93e-15
2560	1.31e-11	9.52e-15	2.02e-13	2.02e-13
5120	7.62e-12	1.05e-16	1.01e-14	1.01e-14

Table 1: Accuracy of our function

The next table represents the same tests as Table 1, only this time, we have used MATLAB's function **pinv** to calculate the MP pseudoinverse.

Size of A	$AA^{+}A = A$	$A^{+}AA^{+} = A^{+}$	$(AA^{+})^{T} = AA^{+}$	$(A^{+}A)^{T} = A^{+}A$
5	4.95e-16	1.27e-14	2.83e-15	1.73e-15
10	1.76e-15	1.31e-15	1.71e-15	5.26e-16
20	3.04e-16	2.80e-17	1.07e-16	1.27e-16
40	3.55e-15	5.88e-16	5.90e-16	5.01e-16
80	4.04e-15	2.40e-17	1.48e-16	2.49e-16
160	1.18e-14	2.17e-17	1.53e-16	1.44e-16
320	3.21e-13	2.37e-16	1.10e-15	6.70e-16
640	1.37e-15	3.62e-20	1.12e-17	7.16e-18
1280	6.35e-14	2.30e-19	2.95e-17	2.25e-17
2560	5.31e-13	1.40e-18	8.81e-17	5.80e-17
5120	1.59e-13	4.99e-20	1.06e-17	1.19e-17

Table 2: Accuracy of MATLAB's function

If we compare both tables, we can see that MATLAB's function offers a bit more accurate results.

5 Application of the Moore-Penrose pseudoinverse

The action of pseudoinverse is used and needed in some of the engineering problems. In this section, we are going to discuss the Stokes problem.

5.1 Formulation of the problem

Let Ω be a bounded domain in $\mathbb{R}^2$ with a sufficiently smooth boundary $\partial\Omega$. We consider a Stokes flow in Ω given by the equations:

$$-\nu\Delta\mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega, \tag{16}$$

$$\nabla\mathbf{u} = 0 \quad \text{in } \Omega, \tag{17}$$

where $\mathbf{u} = (u_1, u_2)$ is the vector function describing flow velocity, p is the scalar pressure function, $\mathbf{f} = (f_1, f_2)$ describes forces acting on the fluid, and $\nu > 0$ is a kinematic viscosity. We assume that the Dirichlet boundary condition is given on $\gamma = \partial\Omega$:

$$\mathbf{u} = \mathbf{u}_D \quad \text{on } \gamma. \tag{18}$$

It is well-known that the velocity is always unique and the pressure is only unique up to a hydrostatic constant. To guarantee the uniqueness, we will also satisfy the zero pressure average condition on the matrix level:

$$\int_\Omega p \, \mathrm{d}x = 0. \tag{19}$$

In next sections we introduce the finite element approximation based on P1-bubble/P1 finite elements and the algebraic problem arising from the TFETI domain decomposition method.

5.2 Standard finite element approximation

We consider the finite element approximation based on P1-bubble/P1 FEM [4]. The algebraic problem arising from the finite element approximation reads as follows:

$$\begin{gathered} \textit{Find } \ (u,p) \in \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \textit{ such that} \\ \begin{pmatrix} A & B^T \\ B & -E \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} b \\ c \end{pmatrix}, \end{gathered} \tag{20}$$

where $A \in \mathbb{R}^{n_u \times n_u}$ is the symmetric, positive definite stiffness matrix, $B \in \mathbb{R}^{n_p \times n_u}$ is the diveregence matrix, $E \in \mathbb{R}^{n_p \times n_p}$ is the stabilization matrix, $b \in \mathbb{R}^{n_u}$, $c \in \mathbb{R}^{n_p}$. We use the following notation: n_p is the number of nodes, n_d is the number of nodes with the Dirichlet boundary condition, and $n_u = 2n_p - 2n_d$ is the number of velocity components corresponding to nodes belonging to the interior of Ω.

To quarantee the uniqueness of the solution of (20), we need one additional condition. We prefer two choices.

(a) Prescribing zero pressure at the last node. The last row in B, E, c, and the last column in E are omitted. We arrive formally at the linear system with the same 2-by-2 block structure as (20), but now with the non-singular matrix. After computing the solution component $\hat{p} \in \mathbb{R}^{n_p-1}$, we get the solution to (20) as $p = (\hat{p}^T, 0)^T$.

(b) The zero pressure average condition. The algebraic representation of (19) reads as follows:

$$a^T p = 0, \tag{21}$$

where $a \in \mathbb{R}^{n_p}$. We arrive formally at the linear system with the same 2-by-2 block structure as (20).

5.3 Discretizations with the TFETI

We consider the domain Ω divided into s non-empty, non-overlaping subdomains Ω_k so that

$$\overline{\Omega} = \bigcup_{k=1}^{s} \overline{\Omega}_k.$$

The decomposition of the problem onto subdomains leads to the block diagonal structure of the stiffness matrix A, the divergence matrix B, and the stabilization matrix E as follows:

$$\begin{aligned} A &= diag(A_1, \ldots, A_s) \in \mathbb{R}^{2n_p \times 2n_p}, \\ B &= diag(B_1, \ldots, B_s) \in \mathbb{R}^{n_p \times 2n_p}, \\ E &= diag(E_1, \ldots, E_s) \in \mathbb{R}^{n_p \times n_p}, \end{aligned}$$

where $A_k \in \mathbb{R}^{2n_{pk} \times 2n_{pk}}$, $B_k \in \mathbb{R}^{n_{pk} \times 2n_{pk}}$, $E_k \in \mathbb{R}^{n_{pk} \times n_{pk}}$, $1 \le k \le s$. It follows easily that B_k has full row-rank, E_k is symmetric, positive definite, and A_k is symmetric, positive semidefinite, and singular. Note that n_{pk} is the number of nodes belonging to $\overline{\Omega}_k$ and $n_p = \sum_{k=1}^{s} n_{pk}$ is the total number of nodes. The analogous block structure exhibits also the right hand side vectors:

$$\begin{aligned} b &= (b_1^T, \ldots, b_s^T)^T \in \mathbb{R}^{2n_p}, \\ c &= (c_1^T, \ldots, c_s^T)^T \in \mathbb{R}^{n_p}, \end{aligned}$$

where $b_k \in \mathbb{R}^{2n_{pk}}$, $c_k \in \mathbb{R}^{n_{pk}}$, $1 \le k \le s$. The continuity of the velocity and the pressure components along interfaces of subdomains is enforced by the gluing matrix:

$$B_g = (B_{g1}, \ldots, B_{gs}) \in \mathbb{R}^{m_g \times n_p},$$

where $B_{gk} \in \mathbb{R}^{m_g \times n_{pk}}$, $1 \le k \le s$, and m_g is the number of nodes, in which the continuity is required (without redundancy). The Dirichlet boundary condition for the velocity is ensured by the matrix:

$$B_d = (B_{d1}, \ldots, B_{ds}) \in \mathbb{R}^{m_d \times n_p},$$

where $B_{dk} \in \mathbb{R}^{m_d \times n_{pk}}$, $1 \leq k \leq s$, and m_d is the number of nodes, in which the Dirichlet boundary condition given by $u_D \in \mathbb{R}^{m_d}$ is required (without redundancy). One additional condition guarantees the uniqueness of the pressure solution component. We use the same two choices as in the previous case, both realized by one Lagrange multiplier.

(a) Prescribing zero pressure at the last node. We consider the vector $a := e_{n_p} \in \mathbb{R}^{n_p}$ containing one at the position of the last entry and zeros elsewhere.

(b) The zero pressure average condition. It is realized by the block vector $a := (a_1^T, \ldots, a_s^T)^T \in \mathbb{R}^{n_p}$, where $a_k \in \mathbb{R}^{n_{kp}}$, $1 \leq k \leq s$.

The algebraic representation in both cases is the same:

$$a^T p = 0.$$

The algebraic problem arising from the TFETI approximation reads as follows:

$$\textit{Find } (u, p, \lambda_u, \lambda_p) \in \mathbb{R}^{2n_p} \times \mathbb{R}^{n_p} \times \mathbb{R}^{m_u} \times \mathbb{R}^{m_p} \textit{ such that}$$

$$\begin{pmatrix} A & B^T & B_u^T & 0 \\ B & -E & 0 & B_p^T \\ B_u & 0 & 0 & 0 \\ 0 & B_p & 0 & 0 \end{pmatrix} \begin{pmatrix} u \\ p \\ \lambda_u \\ \lambda_p \end{pmatrix} = \begin{pmatrix} b \\ c \\ u_D \\ 0 \end{pmatrix}, \tag{22}$$

where $m_u = 2m_g + 2m_d$, $m_p = m_g + 1$, and

$$B_u = \begin{pmatrix} B_g & 0 \\ 0 & B_g \\ B_d & 0 \\ 0 & B_d \end{pmatrix} \in \mathbb{R}^{m_u \times 2n_p}, \quad B_p = \begin{pmatrix} B_g \\ a^T \end{pmatrix} \in \mathbb{R}^{m_p \times n_p}.$$

5.4 The solver based on the velocity-pressure elimination

We consider in (22) an appropriate permutation leading to the following block partition of this system :

$$\begin{pmatrix} K & B_K^T \\ B_K & 0 \end{pmatrix} \begin{pmatrix} \bar{u} \\ \bar{\lambda} \end{pmatrix} = \begin{pmatrix} \bar{b} \\ \bar{u}_D \end{pmatrix}, \tag{23}$$

where K takes the form $K = \mathit{diag}(K_1, \ldots, K_s)$ with

$$K_k := \begin{pmatrix} A_k & B_k^T \\ B_k & -E_k \end{pmatrix}, \quad 1 \leq k \leq s.$$

The block $B_K \in \mathbb{R}^{(m_u+m_p) \times 3n_p}$, the vector $b \in \mathbb{R}^{3n_p}$, and the unknown $\bar{u} \in \mathbb{R}^{3n_p}$ are given by the permutations of the blocks from B_u, B_p, from b, c, and from u, p, respectively, and $\bar{u}_D = (u_D^T, 0^T)^T \in \mathbb{R}^{m_u+m_p}$. The defect of each K_k is 2 and the matrix whose columns are the basis to the null-space of K_k is given by

$$\bar{R}_k = \begin{pmatrix} e & 0 \\ 0 & e \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{3n_{pk} \times 2}, \quad 1 \leq k \leq s,$$

where $e \in \mathbb{R}^{n_{pk}}$ is the vector of all ones. The matrix whose columns are basis to the null-space of K reads as

$$\bar{R} = diag(\bar{R}_1, \dots, \bar{R}_s) \in \mathbb{R}^{3n_p \times 2s}.$$

The first equation in (23) is satisfied, when:

$$\bar{b} - B_K^T \bar{\lambda} \in \mathcal{R}(K) \tag{24}$$

and

$$\bar{u} = K^+(\bar{b} - B_K^T \bar{\lambda}) + \bar{R}\bar{\alpha} \tag{25}$$

for an appropriate $\bar{\alpha} \in \mathbb{R}^{2s}$. If we substitute (25) into the second equation in (23), we arrive at

$$-B_K K^+ B_K^T \bar{\lambda} + B_K \bar{R} \bar{\alpha} = \bar{u}_D - B_K K^+ \bar{b}. \tag{26}$$

Moreover, (24) is equivalent to

$$\bar{R}^T(\bar{b} - B_K^T \bar{\lambda}) = 0. \tag{27}$$

We have shown that $(\bar{\lambda}, \bar{\alpha}) \in \mathbb{R}^{m_u + m_p} \times \mathbb{R}^{2s}$ satisfies the dual system:

$$\begin{pmatrix} F & G^T \\ G & 0 \end{pmatrix} \begin{pmatrix} \bar{\lambda} \\ \bar{\alpha} \end{pmatrix} = \begin{pmatrix} \bar{d} \\ \bar{e} \end{pmatrix}, \tag{28}$$

where $F = B_K K^+ B_K^T$, $G = -\bar{R}^T B_K^T$, $\bar{d} = B_K K^+ \bar{b} - \bar{u}_D$, and $\bar{e} = -\bar{R}^T \bar{b}$. To solve this system, we use the orthogonal projector onto $\mathcal{N}(G)$ given by:

$$P_G = I - G^T(GG^T)^{-1}G.$$

Applying this projector on the first equation in (28), we obtain that $\bar{\lambda}$ satisfies

$$P_G F \bar{\lambda} = P_G d, \quad G\bar{\lambda} = \bar{e}. \tag{29}$$

To arrange (29) as one equation on the vector space $\mathcal{N}(G)$, we decompose the solution $\bar{\lambda}$ into $\bar{\lambda}_{\mathcal{N}} \in \mathcal{N}(G)$ and $\bar{\lambda}_{\mathcal{R}} \in \mathcal{R}(G^T)$ so that

$$\bar{\lambda} = \bar{\lambda}_{\mathcal{N}} + \bar{\lambda}_{\mathcal{R}}. \tag{30}$$

Since $\bar{\lambda}_{\mathcal{R}}$ is easily available via

$$\bar{\lambda}_{\mathcal{R}} = G^T(GG^T)^{-1}\bar{e},$$

we still have to show how to get $\bar{\lambda}_{\mathcal{N}}$. If we substitute (30) into (29), we can see that $\bar{\lambda}_{\mathcal{N}}$ satisfies:

$$P_G F \bar{\lambda}_{\mathcal{N}} = P_G(\bar{d} - F\bar{\lambda}_{\mathcal{R}}). \tag{31}$$

Knowing $\bar{\lambda}$, the solution component $\bar{\alpha}$ is given by

$$\bar{\alpha} = (GG^T)^{-1}G(\bar{d} - F\bar{\lambda}).$$

We summarize our results in the algorithmic scheme:

ALGORITHMIC SCHEME:

(1.a) Assemble $G = -\bar{R}^T B_K^T$, $H = (GG^T)^{-1}$, $\bar{d} = B_K K^+ \bar{b} - \bar{u}_D$, $\bar{e} = -\bar{R}^T \bar{b}$.
(1.b) Assemble $\bar{\lambda}_{\mathcal{R}} = G^T (GG^T)^{-1} \bar{e}$.
(1.c) Assemble $\tilde{d} = P_G(\bar{d} - F\bar{\lambda}_{\mathcal{R}})$.
(1.d) Compute $\bar{\lambda}_{\mathcal{N}}$ by solving $P_G F \bar{\lambda}_{\mathcal{N}} = \tilde{d}$ on $\mathcal{N}(G)$.
(1.e) Assemble $\bar{\lambda} = \bar{\lambda}_{\mathcal{N}} + \bar{\lambda}_{\mathcal{R}}$.
(2) Assemble $\bar{\alpha} = HG(\bar{d} - F\bar{\lambda})$.
(3) Assemble $\bar{u} = K^+(\bar{b} - B_K^T \bar{\lambda}) + \bar{R}\bar{\alpha}$.

The matrices F and P_G may not be assembled explicitly. Their actions on $\bar{\lambda}$ can be evaluated successively as it is indicated by parentheses on the right hand-sides of

$$F\lambda = B_K(K^+(B^T\bar{\lambda})) \quad \text{and} \quad P_G\bar{\lambda} = \bar{\lambda} - G(H(G^T\bar{\lambda})).$$

The actions of K^+ will be discussed below.

5.4.1 The MP pseudoinverse

For each of the matrices K_k, we compute the LU decomposition of

$$M_k := \begin{pmatrix} K_k & \bar{R}_k \\ \bar{R}_k^T & 0 \end{pmatrix}$$

so that

$$P_k M_k = L_k U_k$$

for $1 \leq k \leq s$. This computation is the preprocessing step.

To compute actions of the MP pseudoinverse K^+ on a vector x, we use the loop over all subdomains.

```
Input: $x = (x_1^T, \ldots, x_s^T)^T$
for $k = 1 : s$
    $\tilde{x}_k = (x_k^T, 0, 0)^T$
    $\tilde{y}_k = U_k^{-1}(L_k^{-1}(P_k\tilde{x}_k))$
    $y_k = \tilde{y}_k(1 : \text{end} - 2)$
end
Ouput: $y = (y_1^T, \ldots, y_s^T)^T$
```

5.4.2 The generalised inverse based on the regularization

For each of the matrices K_k, we compute the LU decomposition of of the regularized matrix $M_k := K_k$ redefined as follows:

$$M_k(1,1) := M_k(1,1) + 1, \quad M_k(n_{pk}+1, n_{pk}+1) := M_k(n_{pk}+1, n_{pk}+1) + 1$$

so that

$$P_k M_k = L_k U_k$$

for $1 \leq k \leq s$. This computation is the preprocessing step.

To compute actions of the generalized inverse K^+ on a vector x, we use the loop over all subdomains again.

```
Input: $x = (x_1^T, \ldots, x_s^T)^T$
for $k = 1 : s$
    $y_k = U_k^{-1}(L_k^{-1}(P_k x_k))$
end
Ouput: $y = (y_1^T, \ldots, y_s^T)^T$
```

5.5 Numerical experiments with the generalized inverses

In this section we present our experiments with the generalized inverses to M_k introduced in Subsection 5.4.1 and 5.4.2. The tables then represent the division of the domain Ω onto Ω_k and the accuracy of one calculated MP pseudoinverse matrix from the domain.

5.5.1 The first example

On the figures below, we can observe the structure of the matrices M_k, L_k, U_k, and P_k from Subsection 5.4.1.

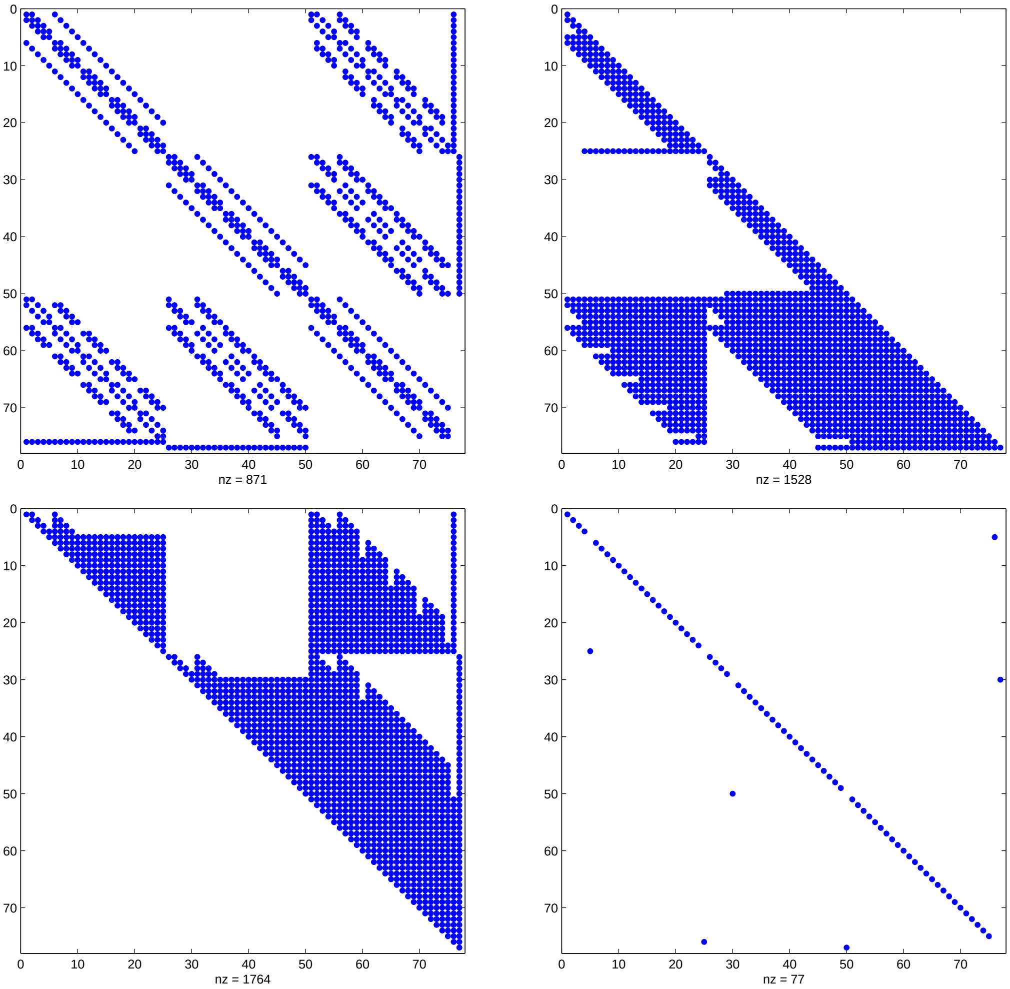

Figure 2: The matrix M_k, L_k, U_k, and P_k (without regularization)

The first table represents the accuracy of the MP pseudoinverse matrix calculated using the block strategy represented by Figure 1.

Division	$AA^+A = A$	$A^+AA^+ = A^+$	$(AA^+)^T = AA^+$	$(A^+A)^T = A^+A$
4x4	5.19e-18	2.61e-16	1.53e-17	3.72e-17
8x8	2.05e-18	1.69e-16	8.64e-18	3.14e-17
12x12	1.65e-18	2.42e-16	6.30e-18	4.19e-17
16x16	1.11e-18	1.98e-16	4.67e-18	3.44e-17
20x20	8.95e-19	2.91e-16	3.75e-18	3.85e-17
24x24	7.52e-19	2.56e-16	3.82e-18	4.43e-17
28x28	6.59e-19	2.53e-16	2.83e-18	4.06e-17
32x32	5.68e-19	2.05e-16	2.55e-18	3.76e-17
36x36	5.26e-19	2.31e-16	2.23e-18	3.96e-17
40x40	4.70e-19	2.12e-16	1.97e-18	4.11e-17

Table 3: Accuracy of the MP pseudoinverse (using the block strategy)

In the second table, we have assembled the MP pseudoinverse using the idea presented in Subsection 5.4.1 onto columns of the identity matrix (for one M_k).

Division	$AA^+A = A$	$A^+AA^+ = A^+$	$(AA^+)^T = AA^+$	$(A^+A)^T = A^+A$
4x4	8.55e-18	2.37e-15	2.86e-17	2.24e-16
8x8	1.77e-17	3.68e-15	3.22e-17	4.95e-16
12x12	3.85e-17	4.48e-15	4.03e-17	5.04e-16
16x16	6.17e-17	7.77e-15	4.63e-17	1.19e-15
20x20	9.23e-17	6.71e-15	6.01e-17	7.67e-16
24x24	1.53e-16	7.72e-15	1.05e-16	9.78e-16
28x28	1.87e-16	1.01e-14	9.63e-17	1.75e-15
32x32	2.50e-16	1.33e-14	1.24e-16	2.28e-15
36x36	3.25e-16	1.68e-14	1.54e-16	2.35e-15
40x40	3.64e-16	1.65e-14	1.83e-16	1.50e-15

Table 4: Accuracy of the MP pseudoinverse (using the GChD)

As we can see from the tables, the first computation gives us more stable and accurate results even with bigger matrices. If we look at the second table, we can notice that the accuracies decrease as we increase the size of the matrix, but are still acceptable.

5.5.2 The second example

In the next example, we have used the idea for the generalised inverse based on the regularization. This idea might save us some time, but the resulting inverse matrix will not be the MP pseudoinverse, but only a generalized inverse.

Once again, we will present the structures of the matrices M_k, L_k, U_k, and P_k in the following figures. We can notice, how the structure of M_k is similar to the one in Example 5.5.1, except for being a little smaller. Another difference easily noticable is the change of the permutation matrix P_k, which in this case is the identity matrix.

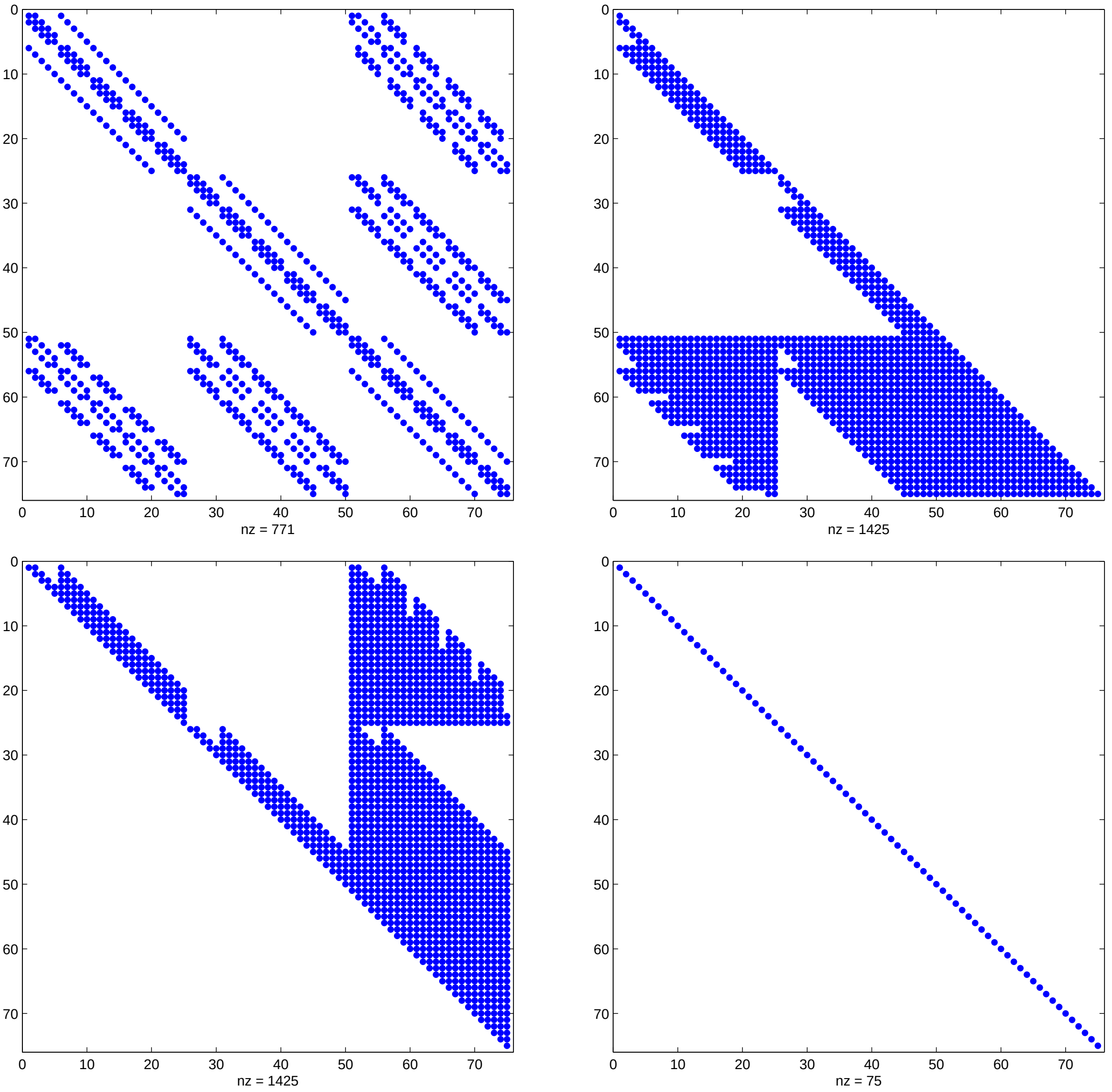

Figure 3: The matrix M_k, L_{M_k}, U_{M_k}, and P_{M_k} (with regularization)

Division	$AXA = A$	$XAX = X$	$(AX)^T = AX$	$(XA)^T = XA$
4x4	5.83e-17	5.03e-02	9.86e-03	9.86e-03
8x8	5.05e-17	4.56e-02	5.04e-03	5.04e-03
12x12	6.52e-17	4.33e-02	3.32e-03	3.32e-03
16x16	5.55e-17	4.19e-02	2.46e-03	2.46e-03
20x20	1.17e-16	4.10e-02	1.95e-03	1.95e-03
24x24	1.00e-16	4.03e-02	1.61e-03	1.61e-03
28x28	8.78e-17	3.99e-02	1.38e-03	1.38e-03
32x32	5.44e-17	3.96e-02	1.20e-03	1.20e-03
36x36	9.06e-17	3.93e-02	1.06e-03	1.06e-03
40x40	1.04e-16	3.91e-02	9.54e-04	9.54e-04

Table 5: Accuracy of the generalized inverse

And as we can see from the table, the matrix is indeed a generalized inverse as it satisfies only the first from the MP pseudoinverse conditions. Yet, we are going to continue using this algorithm in future experiments, as it might yield interesting results compared to the MP pseudoinverse.

5.6 Numerical experiments with iteration methods

Now, that we have analyzed the accuracies of our pseudoinverses and generalized matrices, we can continue and try solving the Stokes problem based on the algorithm in Subsection 5.4. We will no longer calculate the solution for just one Ω_k domain, but for all. We will observe the overall effectivness of calculations with regard to the H/h ratio, where H is the diameter of the largest subdomain Ω_k and h is the stepsize of the finite element mesh.

5.6.1 The first example

To solve the linear systems, we are going to use two iteration methods. One of them is BiCGSTAB [5] algorithm that is used for the numerical solution of non-symmetric linear systems. In the Tables 6 and 7 we can see how many iterations does it take to solve the linear system and the accuracy of the calculations (the relative norm of the residual vector). The difference between Table 6 and Table 7 is in the used type of the generalized inverse. In Table 6 we have used the MP pseudoinverse to solve the system. For Table 7 we used the generalized inverse based on the regularization.

Ratio	$\frac{H}{h} = 2$	$\frac{H}{h} = 4$	$\frac{H}{h} = 6$	$\frac{H}{h} = 8$	$\frac{H}{h} = 10$
$Nx = 2$	123, 5.9e-4	909, 9.4e-4	1957, 9.5e-4	2087, 9.1e-4	3863, 9.4e-4
$Nx = 4$	261, 8.8e-4	1327, 9.4e-4	1419, 9.0e-4	2571, 9.4e-4	2023, 9.8e-4
$Nx = 6$	345, 7.7e-4	1897, 8.7e-4	2185, 9.7e-4	1903, 9.8e-4	2381, 9.6e-4
$Nx = 8$	401, 9.7e-4	2081, 9.6e-4	4463, 9.4e-4	1929, -	3261, 9.8e-4
$Nx = 10$	693, 9.5e-4	3693, 9.8e-4	3715, 9.7e-4	5961, 9.0e-4	7519, 9.7e-4

Table 6: Efficiency of BiCGSTAB (the MP pseudoinverse)

Ratio	$\frac{H}{h} = 2$	$\frac{H}{h} = 4$	$\frac{H}{h} = 6$	$\frac{H}{h} = 8$	$\frac{H}{h} = 10$
$Nx = 2$	145, 7.4e-4	877, 8.1e-4	1741, 9.8e-4	1773, 9.7e-4	3119, 9.6e-4
$Nx = 4$	245, 8.0e-4	1027, 7.8e-4	1749, 9.8e-4	3573, 9.2e-4	1727, 9.8e-4
$Nx = 6$	401, 9.2e-4	2577, 9.2e-4	2207, 8.8e-4	2563, 9.4e-4	2667, 9.3e-4
$Nx = 8$	383, 9.7e-4	2307, 9.5e-4	3263, 9.8e-4	3781, 9.3e-4	3509, 9.5e-4
$Nx = 10$	485, 8.2e-4	3811, 9.8e-4	4759, -	7349, 9.8e-4	6355, 9.0e-4

Table 7: Efficiency of BiCGSTAB (the generalized inverse)

As we can see from the tables, BiCGSTAB algorithm requires a lot of iterations, but offers a quite precise results. As far as the type of generalized inverse goes, it does not seem to have much effect on the results. Since this algorithm is an iterative method for indefinite matrices and since it is known to cause breakdowns, it is unable to give us results for all our calculations, hence some of the accuracies are missing.

5.6.2 The second example

As the second iteration method we are going to use the conjugate gradient method [6]. We are experimenting with this method, as it is designed for use with symmetric, positive definite matrices, whereas we are working with indefinite matrices. What we will try to do, is to use this method and see if it allows us to calculate the systems of linear equations and if it does, then in how many iterations will it complete the calculations.

Ratio	$\frac{H}{h} = 2$	$\frac{H}{h} = 4$	$\frac{H}{h} = 6$	$\frac{H}{h} = 8$	$\frac{H}{h} = 10$
$Nx = 2$	97, 7.2e-4	301, 9.2e-4	373, 9.5e-4	463, 9.9e-4	485, 9.8e-4
$Nx = 4$	125, 8.6e-4	319, 5.0e-4	373, 6.8e-4	385, 7.9e-4	349, 8.8e-4
$Nx = 6$	157, 7.9e-4	303, 8.4e-4	325, 8.5e-4	353, 7.6e-4	315, 9.8e-4
$Nx = 8$	163, 8.3e-4	319, 9.0e-4	333, 8.6e-4	309, 9.3e-4	323, 7.7e-4
$Nx = 10$	175, 8.5e-4	315, 7.9e-4	319, 9.0e-4	343, 5.3e-4	283, 9.4e-4

Table 8: Efficiency of CGM (the MP pseudoinverse)

Ratio	$\frac{H}{h} = 2$	$\frac{H}{h} = 4$	$\frac{H}{h} = 6$	$\frac{H}{h} = 8$	$\frac{H}{h} = 10$
$Nx = 2$	97, 6.6e-4	305, 6.2e-4	399, 6.0e-4	463, 8.9e-4	485, 9.9e-4
$Nx = 4$	125, 9.1e-4	321, 5.3e-4	373, 6.9e-4	385, 7.7e-4	349, 8.8e-4
$Nx = 6$	159, 9.0e-4	303, 8.5e-4	325, 8.5e-4	353, 7.6e-4	315, 9.8e-4
$Nx = 8$	167, 9.5e-4	319, 7.7e-4	333, 9.5e-4	309, 9.3e-4	323, 7.7e-4
$Nx = 10$	189, 9.3e-4	315, 8.1e-4	319, 9.0e-4	343, 5.3e-4	283, 9.4e-4

Table 9: Efficiency of CGM (the generalized inverse)

As we can see from the residual norms, which are below the level of the terminating tolerance, the CGM can be used to solve the systems of linear equations. We can see that the CGM is undoubtebly more efficient and reliable than the BiCGSTAB method. Not only does it require far less iterations, but gave us results for each calculation as well. More than that, we have made an interesting discovery, where the number of iterations for both types of the generalized inverses are very similar, with only little differences.

6 Conclusion

The aim of this thesis was to test several ways of calculating the generalized inverses based on the algorithms used for calculating the inverses of non-singular matrices. We have shown how to transform a singular matrix into non-singular one and which block of the inverse of the non-singular matrix is the generalized inverse of the original matrix. More than that, we presented two more ways of computing a generalized inverse. We have found out that using the block strategy offers the most accurate results and, by the end of the thesis, we have solved systems of linear equations arising from the Stokes problem using two types of the generalized inverses.

Jan Pacholek

7 References

[1] Kučera R., Kozubek T., Markopoulos A., Machálková J., *Numerical Linear Algebra with Applications: 677 - 699,* Wiley Online Library, John Wiley & Sons, Ltd., 18. October 2011.

[2] Kučera R., Kozubek T., Markopoulos A., *Linear Algebra and its Applications 438: 3011 - 3029*, Elsevier, Elsevier Inc., 4. January 2013.

[3] Ben-Israel A., Greville T., *Generalized Inverses: Theory and Applications*, New York: Wiley, 1974.

[4] Koko J., *Vectorized Matlab codes for the Stokes problem with P1-bubble/P1 finite element*, at: `www.isima.fr/~jkoko/Codes.html`, 2012.

[5] Van der Vorst H. A., *Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Non Symmetric Linear Systems*, SIAM Journal on Scientific and Statistical Computing 13 (2): 631 - 644, March 1992

[6] Golub J., Van Loan C. F., *Matrix Computations, Third Edition*, Baltimore, Johns Hopkins University Press, 15. October 1996

[7] Dostál Z., Vondrák V., *Lineární algebra*, at: `mi21.vsb.cz/sites/mi21.vsb.cz/files/unit/linearni_algebra.pdf`, 24.4.2012

A Inverse matrix

Here is the solution for the example calculated in Section 2:

$$\left[\begin{array}{ccc|ccc} 2 & -3 & 1 & 1 & 0 & 0 \\ 1 & 2 & -1 & 0 & 1 & 0 \\ 2 & 1 & 1 & 0 & 0 & 1 \end{array}\right] \begin{array}{c} \\ 2r_2 - r_1 \rightarrow \\ -r_1 \end{array} \left[\begin{array}{ccc|ccc} 2 & -3 & 1 & 1 & 0 & 0 \\ 0 & 7 & -3 & -1 & 2 & 0 \\ 0 & 4 & 0 & -1 & 0 & 1 \end{array}\right] \begin{array}{r} \\ \rightarrow \\ -r_2 \end{array}$$

$$\left[\begin{array}{ccc|ccc} 2 & -3 & 1 & 1 & 0 & 0 \\ 0 & 4 & 0 & -1 & 0 & 1 \\ 0 & 7 & -3 & -1 & 2 & 0 \end{array}\right] \begin{array}{r} \\ \rightarrow \\ 4r_3 - 7r_2 \end{array} \left[\begin{array}{ccc|ccc} 2 & -3 & 1 & 1 & 0 & 0 \\ 0 & 4 & 0 & -1 & 0 & 1 \\ 0 & 0 & -12 & 3 & 8 & -7 \end{array}\right] \begin{array}{l} 4r_1 + 3r_2 \\ \\ \\ \end{array}$$

$$\rightarrow \left[\begin{array}{ccc|ccc} 8 & 0 & 4 & 1 & 0 & 3 \\ 0 & 4 & 0 & -1 & 0 & 1 \\ 0 & 0 & -12 & 3 & 8 & -7 \end{array}\right] \begin{array}{r} 3r_1 + r_3 \\ \rightarrow \\ \\ \end{array} \left[\begin{array}{ccc|ccc} 24 & 0 & 0 & 6 & 8 & 2 \\ 0 & 4 & 0 & -1 & 0 & 1 \\ 0 & 0 & -12 & 3 & 8 & -7 \end{array}\right]$$

B The Moore-Penrose pseudoinverse

Here is the solution for the example calculated in Section 4:

$$\left[\begin{array}{ccccc|ccccc} 1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \begin{array}{l} \\ +r_1 \\ \qquad \rightarrow \\ \\ -r_1 \end{array}$$

$$\left[\begin{array}{ccccc|ccccc} 1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 2 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 2 & 1 & -1 & -1 & -1 & 0 & 0 & 0 & 1 \end{array}\right] \begin{array}{l} \\ \\ +r_2 \rightarrow \\ \\ -2r_2 \end{array}$$

$$\left[\begin{array}{ccccc|ccccc} 1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 2 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 3 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 3 & 1 & -5 & -3 & -2 & 0 & 0 & 1 \end{array}\right] \begin{array}{l} \\ \\ \qquad \rightarrow \\ +r_3 \\ -3r_3 \end{array}$$

$$\left[\begin{array}{ccccc|ccccc} 1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 2 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 3 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 4 & -14 & -6 & -5 & -3 & 0 & 1 \end{array}\right] \begin{array}{l} \\ \\ \rightarrow \\ \\ \\ \end{array}$$

$$\left[\begin{array}{ccccc|ccccc} 1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 2 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 3 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 4 & -14 & -6 & -5 & -3 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 \end{array}\right] \begin{array}{l} \\ \\ \qquad \rightarrow \\ +14r_5 \\ \\ \end{array}$$

$$\left[\begin{array}{ccccc|ccccc} 1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 2 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 3 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & -\frac{5}{2} & -\frac{3}{2} & \frac{1}{2} & \frac{7}{2} & 1 \\ 0 & 0 & 0 & 0 & 1 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 \end{array}\right] \begin{array}{l} \\ \\ \quad \rightarrow \\ \frac{r_4}{4} \\ \\ \end{array}$$

$$\left[\begin{array}{ccccc|ccccc} 1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 2 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 3 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -\frac{5}{8} & -\frac{3}{8} & \frac{1}{8} & \frac{7}{8} & \frac{1}{4} \\ 0 & 0 & 0 & 0 & 1 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 \end{array}\right] \begin{array}{l} -r_5 \\ -2r_5 \\ -3r_5 \rightarrow \\ \\ \\ \end{array}$$

$$\left[\begin{array}{ccccc|ccccc} 1 & -1 & 0 & 0 & 0 & \frac{3}{4} & -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & 0 \\ 0 & 1 & -1 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 0 \\ 0 & 0 & 1 & -1 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & -\frac{3}{4} & 0 \\ 0 & 0 & 0 & 1 & 0 & -\frac{5}{8} & -\frac{3}{8} & \frac{1}{8} & \frac{7}{8} & \frac{1}{4} \\ 0 & 0 & 0 & 0 & 1 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 \end{array}\right] \begin{array}{l} \\ \\ +r_4 \rightarrow \\ \\ \\ \end{array}$$

$$\left[\begin{array}{ccccc|ccccc} 1 & -1 & 0 & 0 & 0 & \frac{3}{4} & -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & 0 \\ 0 & 1 & -1 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 0 \\ 0 & 0 & 1 & 0 & 0 & -\frac{3}{8} & -\frac{1}{8} & \frac{3}{8} & \frac{1}{8} & \frac{1}{4} \\ 0 & 0 & 0 & 1 & 0 & -\frac{5}{8} & -\frac{3}{8} & \frac{1}{8} & \frac{7}{8} & \frac{1}{4} \\ 0 & 0 & 0 & 0 & 1 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 \end{array}\right]\begin{array}{l} \\ +r_3 \\ \\ \\ \\ \end{array} \rightarrow$$

$$\left[\begin{array}{ccccc|ccccc} 1 & -1 & 0 & 0 & 0 & \frac{3}{4} & -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & 0 \\ 0 & 1 & 0 & 0 & 0 & \frac{1}{8} & \frac{3}{8} & -\frac{1}{8} & -\frac{3}{8} & \frac{1}{4} \\ 0 & 0 & 1 & 0 & 0 & -\frac{3}{8} & -\frac{1}{8} & \frac{3}{8} & \frac{1}{8} & \frac{1}{4} \\ 0 & 0 & 0 & 1 & 0 & -\frac{5}{8} & -\frac{3}{8} & \frac{1}{8} & \frac{7}{8} & \frac{1}{4} \\ 0 & 0 & 0 & 0 & 1 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 \end{array}\right]\begin{array}{l} +r_2 \\ \\ \\ \\ \\ \end{array} \rightarrow$$

$$\left[\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 0 & 0 & \frac{7}{8} & \frac{1}{8} & -\frac{3}{8} & -\frac{5}{8} & \frac{1}{4} \\ 0 & 1 & 0 & 0 & 0 & \frac{1}{8} & \frac{3}{8} & -\frac{1}{8} & -\frac{3}{8} & \frac{1}{4} \\ 0 & 0 & 1 & 0 & 0 & -\frac{3}{8} & -\frac{1}{8} & \frac{3}{8} & \frac{1}{8} & \frac{1}{4} \\ 0 & 0 & 0 & 1 & 0 & -\frac{5}{8} & -\frac{3}{8} & \frac{1}{8} & \frac{7}{8} & \frac{1}{4} \\ 0 & 0 & 0 & 0 & 1 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 \end{array}\right]$$