An Extension of the Gauss-Jordan Procedure for Computing the Inverse of Parametric Matrices

Mahdi Dehghani Darmian[1]

Department of Mathematics, Technical and Vocational University (TVU), Tehran, Iran

Abstract

In this paper, we demonstrate that the Gauss-Jordan method can be extended to compute the inverse of parametric matrices, offering a powerful tool for solving systems of linear equations and analyzing parametric systems. Using the concepts of the Gauss-Jordan systems and linearly dependency systems for linear systems involving parameters [5, 6], we introduce the notion of an inverse matrix system for a parametric matrix. We also present an algorithm for computing an inverse system for a given parametric matrix. All mentioned algorithms have been implemented in `Maple`, and their efficiency has experimented on a diverse set of benchmark matrices.

Keywords: Parametric matrices, Gröbner system, IMS algorithm, Gauss-Jordan system, Inverse matrix system

AMS Mathematical Subject Classification [2010]: 13P10, 15A03

1 Introduction

The computation of the inverse matrix is an essential and fundamental task in the field of linear algebra. It has a wide range of applications in various fields, including engineering, physics, computer science, and statistics. The inverse matrix is a valuable tool for solving systems of linear equations and gaining insights into the underlying structures of mathematical models. Among the many techniques available for computing the inverse of a matrix, the Gauss-Jordan method is particularly notable. It is a versatile approach that is widely used due to its simplicity of implementation. To find the inverse of a $n \times n$ matrix A by this method: adjoin the identity matrix I to the right side of A, thereby producing a matrix of the form $[A|I]$ and then apply the row operations to this matrix until the left side is reduced to I. If successful, these operations will convert the right side to A^{-1}, so that the final matrix will have the form $[I|A^{-1}]$. The Gauss-Jordan method directly obtains the inverse without computing determinants or cofactors. By systematically applying row operations to the augmented matrix $[A|I]$, we can transform the left side into the identity matrix I, while the right side becomes the desired inverse matrix A^{-1}. The row operations involved in the Gauss-Jordan method include multiplying a row by a non-zero scalar, swapping rows, and adding or subtracting multiples of one row from another. By carefully performing these operations, we can eliminate all elements except

[1]Speaker

for those in the diagonal of the left side, which will become ones in the identity matrix. It is important to note that the Gauss-Jordan method may fail if the matrix is singular, meaning it does not have an inverse. In such cases, the left side of the augmented matrix will not transform into the identity matrix, indicating the absence of an inverse. Once we have successfully transformed the left side into the identity matrix, the right side of the augmented matrix will be the inverse matrix A^{-1}. This inverse matrix allows us to efficiently solve systems of linear equations, analyze transformations, and gain insights into the behavior of mathematical models. In summary, the computation of the inverse matrix using the Gauss-Jordan method is a fundamental technique in linear algebra, with widespread applications in various fields. Its simplicity and directness make it an invaluable tool for solving equations, understanding mathematical relationships, and tackling complex problems.

Example 1.1. Let us consider

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 3 \\ 1 & 0 & 8 \end{bmatrix}$$

The following process shows how does Gauss-Jordan method works.

$$[A|I] = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 5 & 3 \\ 1 & 0 & 8 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \xrightarrow[R_3 \to R_3 - R_1]{R_2 \to R_2 - 2R_1}$$

$$= \left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & -3 & -2 & 1 & 0 \\ 0 & -2 & 15 & -1 & 0 & 1 \end{array}\right] \xrightarrow{R_3 \to R_3 + 2R_2}$$

$$= \left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & -3 & -2 & 1 & 0 \\ 0 & 0 & -1 & -5 & 2 & 1 \end{array}\right] \xrightarrow{R_3 \to -R_3}$$

$$= \left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & -3 & -2 & 1 & 0 \\ 0 & 0 & 1 & 5 & -2 & -1 \end{array}\right] \xrightarrow{R_2 \to R_2 + 3R_3}$$

$$= \left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & 0 & 13 & -5 & -3 \\ 0 & 0 & 1 & 5 & -2 & -1 \end{array}\right] \xrightarrow{R_1 \to R_1 - 2R_2}$$

$$= \left[\begin{array}{ccc|ccc} 1 & 0 & 3 & -25 & 10 & 6 \\ 0 & 1 & 0 & 13 & -5 & -3 \\ 0 & 0 & 1 & 5 & -2 & -1 \end{array}\right] \xrightarrow{R_1 \to R_1 - 3R_3}$$

$$= \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & -40 & 16 & 9 \\ 0 & 1 & 0 & 13 & -5 & -3 \\ 0 & 0 & 1 & 5 & -2 & -1 \end{array}\right]$$

$$A^{-1} = \begin{bmatrix} -40 & 16 & 9 \\ 13 & -5 & -3 \\ 5 & -2 & -1 \end{bmatrix}$$

In many real-world problems, matrices with parametric entries, where the matrix elements are expressed as parameters, often arise during the modeling process. This paper focuses on exploring the application of the Gauss-Jordan method to compute the inverse of matrices with parametric entries. We demonstrate how the Gauss-Jordan method can be extended to handle matrices with parametric entries, which allows researchers to solve complex problems in various domains. The following example serves as an illustration that the usual approach for computing the inverse of matrices may not be suitable for matrices with parametric entries.

Example 1.2. Let us consider the following parametric matrix with entries in $\mathbb{K}[a,b,c]$.

$$A = \begin{bmatrix} -b & 1 & a+1 \\ 0 & c & b+1 \\ -1 & 3+c & 1 \end{bmatrix}$$

The matrix inverse of A computed by the function `LinearAlgebra:-MatrixInverse(A)` of `Maple` is equal to

$$A^{-1} = \begin{bmatrix} -\frac{bc+3\,b+3}{b^2c+ac+3\,b^2+2\,b+c-1} & \frac{ac+3\,a+c+2}{b^2c+ac+3\,b^2+2\,b+c-1} & -\frac{ac-b+c-1}{b^2c+ac+3\,b^2+2\,b+c-1} \\ -\frac{b+1}{b^2c+ac+3\,b^2+2\,b+c-1} & \frac{a-b+1}{b^2c+ac+3\,b^2+2\,b+c-1} & \frac{(b+1)b}{b^2c+ac+3\,b^2+2\,b+c-1} \\ \frac{c}{b^2c+ac+3\,b^2+2\,b+c-1} & \frac{bc+3\,b-1}{b^2c+ac+3\,b^2+2\,b+c-1} & -\frac{bc}{b^2c+ac+3\,b^2+2\,b+c-1} \end{bmatrix}$$

A^{-1} is defined when $\Delta = b^2c + ac + 3b^2 + 2b + c - 1 \neq 0$. For instance if $a = 1$, $b = 0$, and $c = 0$ then A and A^{-1} are as follow:

$$A|_{\{a=1,b=0,c=0\}} = \begin{bmatrix} 0 & 1 & 2 \\ 0 & 0 & 1 \\ -1 & 3 & 1 \end{bmatrix}$$

$$A^{-1}|_{\{a=1,b=0,c=0\}} = \begin{bmatrix} 3 & -5 & -1 \\ 1 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

However, A^{-1} undefined for the values $a = 1, b = \frac{1}{3}$ and $c = 0$, since $\Delta = b^2c + ac + 3b^2 + 2b + c - 1 = 0$ and $A|_{\{a=1,b=\frac{1}{3},c=0\}}$ is the following Matrix with rank 2.

$$A|_{\{a=-2,b=-1,c=0\}} = \begin{bmatrix} -1/3 & 1 & 2 \\ 0 & 0 & 4/3 \\ -1 & 3 & 1 \end{bmatrix}$$

In such cases, the traditional approach of computing the inverse matrix using standard methods may not be suitable. The presence of parameters in the matrix complicates the calculations and requires a different approach. By incorporating the Gauss-Jordan method, we can effectively handle matrices with parametric entries and obtain the inverse matrix. The Gauss-Jordan method involves transforming the given matrix into a reduced row-echelon form through a series of elementary row operations. These operations include swapping rows, multiplying rows by constants, and adding multiples of one row to another. By performing these operations systematically, we can manipulate the matrix to obtain its inverse. Extending the Gauss-Jordan method to matrices with parametric entries requires careful consideration of the parameter values. We need to analyze the possible values of the parameters and their impact on the matrix operations. By treating the parameters as variables, we can perform the necessary calculations and derive the inverse matrix in terms of these variables. This approach allows us to obtain a general expression for the inverse matrix, which encompasses all possible parameter values. It provides a more comprehensive understanding of the matrix's properties and enables us to solve a wide range of problems that involve matrices with parametric entries. In conclusion, the application of the Gauss-Jordan method to matrices with parametric entries is a valuable tool for solving complex problems across various domains. It facilitates the computation of the inverse matrix, which is essential for analyzing systems of linear equations and understanding the underlying structures of mathematical models. By incorporating this method, researchers and practitioners can tackle real-world problems more effectively and gain deeper insights into the behavior of parametric matrices.

In fact, one may be interested in identifying all possible parametric relations between a, b, and c where A is invertible or singular. With more details, we want to partition $\mathbb{V}(\Delta)$ and also $\mathbb{C}^3 \setminus \mathbb{V}(\Delta)$ into subvarieties. For this purpose, we introduce the inverse matrix system for parametric matrices and present an algorithm for computing it. In doing so, we can use two ideas; *Gröbner systems* and *Gauss-Jordan systems* based on *linearly dependency systems*.

2 Detecting invertibility or singularity via Gröbner Systems

The concepts of Gröbner systems and their algorithms were introduced by Weispfenning [15] in 1992. Gröbner systems are an extension of Gröbner bases to polynomials with parametric coefficients. Simply put, by computing the Gröbner system of a parametric ideal, we divide the parameter space into a finite set of cells. For each cell, we provide a set of polynomials. When given specific parameter values, we can determine the cell that contains these values and obtain the corresponding polynomial set, which serves as a Gröbner basis for the ideal. Gröbner systems play a crucial role in algebraic geometry and computer algebra systems. They provide a powerful tool for solving systems of polynomial equations with parametric coefficients. By analyzing the structure of the Gröbner system, we can gain valuable insights into the geometry of the solution space. The computation of Gröbner systems involves a series of algorithms that manipulate polynomials to transform the system into a more manageable form and efficiently compute the desired results. One of the key advantages of Gröbner systems is their ability to handle parameterized polynomials. This means that the coefficients of the polynomials can be expressed as variables or functions, allowing for a more flexible representation of the problem. By computing the Gröbner system of a parametric ideal, we can effectively analyze the behavior of the system over a range of parameter values. The division of the parameter space into cells is a fundamental aspect of Gröbner systems. Each cell corresponds to a region in the parameter space where the system's behavior is qualitatively different. By providing a set of polynomials for each cell, we can describe the variations in the solution space as the parameters change. When specific parameter values are given, we can determine the corresponding cell by evaluating the inequalities defined by the system. This allows us to select the appropriate polynomial set, which serves as a Gröbner basis for the ideal associated with those parameter values. Overall, Gröbner systems and their algorithms offer a powerful framework for analyzing and solving parametric polynomial systems. They provide a systematic approach to handling complex equations and enable us to explore the intricate relationships between the parameters and the solutions.

Throughout this article, we consider $\mathcal{R} = \mathbb{K}[x_1, \ldots, x_n]$ the polynomial ring in terms of $x_1, \ldots, x_n$ over a field $\mathbb{K}$. Let $\mathcal{I} = \langle f_1, \ldots, f_k \rangle \subset \mathcal{R}$ be the polynomial ideal generated by the f_i's. For any $f \in \mathcal{R}$, the *leading monomial* of f, denoted by $\mathrm{LM}_{\prec}(f)$, is the greatest monomial (w.r.t. the monomial ordering $\prec$)

appearing in f and its coefficient is the *leading coefficient* of f which denoted by $\mathrm{LC}_{\prec}(f)$. The *leading term* of f w.r.t. $\prec$ is the product $\mathrm{LT}_{\prec}(f) = \mathrm{LC}_{\prec}(f)\mathrm{LM}_{\prec}(f)$. The *leading monomial ideal* of $\mathcal{I}$ is defined to be $\mathrm{LM}_{\prec}(\mathcal{I}) = \langle \mathrm{LM}_{\prec}(f) \mid f \in \mathcal{I}\rangle$. A finite subset $\{g_1, \ldots, g_m\} \subset \mathcal{I}$ is called a *Gröbner basis* for $\mathcal{I}$ w.r.t. $\prec$ if $\mathrm{LM}_{\prec}(\mathcal{I}) = \langle \mathrm{LM}_{\prec}(g_1), \ldots, \mathrm{LM}_{\prec}(g_m)\rangle$. Gröbner bases, along with the initial algorithm to calculate them, were first introduced by Buchberger in his Ph.D. thesis under the supervision of Gröbner [1]. For further information, interested readers are directed to [2]. Using these notations, we will now recall the definition of Gröbner bases for parametric polynomial ideals, also known as Gröbner systems. In essence, Gröbner systems can be seen as an expansion of Gröbner bases for polynomial ideals over fields to polynomial ideals with parametric coefficients.

Now, we prepare to recall the definition of a Gröbner system for a parametric polynomial ideal.

Definition 2.1. Let $F = \{f_1, \ldots, f_k\} \subset \mathcal{S}$ and $\mathcal{G} = \{(N_i, W_i, G_i)\}_{i=1}^{\ell}$ be a finite triple set where $N_i, W_i \subset \mathbb{K}[\mathbf{a}]$ and $G_i \subset \mathcal{S}$ are finite for $i = 1, \ldots, \ell$. The set $\mathcal{G}$ is called a Gröbner system for $\langle F\rangle$ w.r.t. to $\prec_{\mathbf{x},\mathbf{a}}$ over $\mathcal{V} \subseteq \overline{\mathbb{K}}^m$ if for any i we have

- $\sigma(G_i)$ is a Gröbner basis of $\langle \sigma(F)\rangle$ with respect to $\prec_{\mathbf{x}}$, for any specialization $\sigma : \mathbb{K}[\mathbf{a}] \to \overline{\mathbb{K}}$ satisfying (N_i, W_i); each pair (N_i, W_i) is called a specification (parametric constraint).
- $\mathcal{V} \subseteq \bigcup_{i=1}^{\ell} \mathbb{V}(N_i) \setminus \mathbb{V}(W_i)$.

Any triple (N_i, W_i, G_i) is said a branch (segment) of the Gröbner system $\mathcal{G}$ for each $1 \leq i \leq \ell$. Moreover, if $\mathcal{V} = \overline{\mathbb{K}}^m$ then $\mathcal{G}$ is generally called a Gröbner system of F.

The concept of Gröbner system was introduced by Weispfenning in [15]. He proved that any parametric polynomial ideal has a Gröbner system [15, Proposition 3.4 and Theorem 2.7] and described an algorithm to compute it [15, Theorem 3.6]. Since then, several different algorithms have been designed to compute the Gröbner system of an ideal, each of which has advantages and faults [4, 6, 7, 8, 9, 10, 11, 12, 13, 14].

Example 2.2. Let $F = \{[ax^2 - (b-1)xy - z, xy - (b-c)y^3, 2 - (a+c)z]\} \subset \mathbb{K}[a, b, c, d, , x, y, z]$ where x, y, z are variables and a, b, c, d are parameters. We consider the monomial orderings $z \prec_{drl} y \prec_{drl} x$ and $d \prec_{drl} c \prec_{drl} b \prec_{drl} a$. Using our implementation of the PF_4 algorithm [4], we can compute a Gröbner system for $\langle F\rangle$ as follows

$$\begin{cases}
([a+c],[c,b-c], & [1]) \\
([b-c,a+c],[c], & [1]) \\
([c-1,b-1,a],[\,], & [1]) \\
([b-1,a],[c-1], & [1]) \\
([b-c,a],[c-1], & [1]) \\
([c,a],[b,b-1], & [1]) \\
([c-1,b-1],[a,a+1], & [ax^2-z, xy, -az-z+2, 2y]) \\
([-b+c],[a,a+c,c-1], & [ax^2-cxy+xy-z, xy, -az-cz+2, 2y]) \\
([\,],[a,a+c,b-c], & [ax^2-bxy+xy-z, -by^3+cy^3+xy, -az-cz+2]) \\
([a],[c,b-1,b-c], & [-bxy+xy-z, -b^2y^3+bcy^3+by^3-cy^3-z, -cz+2, \\
 & 2by^2-cxz-2cy^2, 2bx^2+2byz-2cyz-2x^2])
\end{cases}$$

The above Gröbner system has ten branches while assuming $a = 1$, $b = 1$, $c = 1$ randomly, the seventh branch corresponds to these values of parameters, and so $\{x^2 - z, xy, -2z + 2, 2y\}$ is a Gröbner basis for the ideal $\langle F\rangle \mid_{a=1,b=1,c=1}$.

One may wonder how to determine whether a parametric matrix A is invertible or singular using Gröbner systems. To do so, we can compute a Gröbner system $\{(N_i, W_i, G_i)\}_{i=1}^{\ell}$ for the corresponding parametric linear ideal $\langle \mathcal{I} \rangle$. If $|G_i| = |\mathcal{I}|$ then A is invertible as long as N_i and W_i are satisfied; otherwise, it is singular ($|\mathcal{I}|$ is the cardinality of $\mathcal{I}$). Let's illustrate this approach with a straightforward example.

Example 2.3. Consider the following parametric matrix A where a, b, c are parameters.

$$A = \begin{bmatrix} a & 1 & -1 \\ -1 & b & 1 \\ 1 & -1 & c \end{bmatrix}$$

which corresponds to the parametric linear ideal $\mathcal{I} = \langle ax + y - z, -x + by + z, x - y + cz \rangle$. A Gröbner system of $\mathcal{I}$ w.r.t. the product ordering of $z \prec_{lex} y \prec_{lex} x$ and $c \prec_{lex} b \prec_{lex} a$ by using our implementation of PFGLM algorithm [5, 6] in Maple is as follows:

$$\begin{cases} ([\,], & [a+1, c+b+a+abc], & [x-y+cz, y+ay-z-acz, cz+bz+az+abcz]), \\ ([a+1], & [b-1, c-1], & [x-y+z, by-y+2z, cz-z]), \\ ([c-1, a+1], & [b-1], & [x-y+z, by-y+2z]), \\ ([b-1, a+1], & [\,], & [x-y, z]), \\ ([c+b+a+abc], & [a+1], & [x-y+cz, y+ay-z-acz]). \end{cases}$$

This follows the following rank system of A (introduced in [6]):

$$\begin{cases} ([\,], & [a+1, c+b+a+abc], & 3) \\ ([a+1], & [b-1, c-1], & 3) \\ ([c-1, a+1], & [b-1], & 2) \\ ([b-1, a+1], & [\,], & 2) \\ ([c+b+a+abc], & [a+1], & 2). \end{cases}$$

So, we can conclude the following system for deciding the invertibility or singularity of A.

$$\begin{cases} ([\,], & [a+1, c+b+a+abc], & \text{Invertible}) \\ ([a+1], & [b-1, c-1], & \text{Invertible}) \\ ([c-1, a+1], & [b-1], & \text{Singular}) \\ ([b-1, a+1], & [\,], & \text{Singular}) \\ ([c+b+a+abc], & [a+1], & \text{Singular}). \end{cases}$$

For example, if $a = -1, b = 2$, and $c = 1$ the third branch corresponds to these values of parameters, and so $A|_{a=-1,b=2,c=1}$ is singular.

3 Computation of Inverse Matrix Systems

The computation of the inverse matrix system using the Gröbner system has two drawbacks. Firstly, if the third component of any triple is "Invertible", only the invertibility is reported without calculating the inverse matrix. Additionally, Suzuki and Sato [14] have noted that the computation of Gröbner systems of parametric linear ideals is generally slow and inefficient. This is why we employ the efficient algorithm GJS to compute the Gauss-Jordan system (without utilizing Gröbner systems) for parametric matrices. The GJS algorithm is based on the LDS and GES algorithms proposed in [6]. To begin, we restate the behavior of the LDS algorithm [6] to determine the dependency of a linear parametric polynomial on a given set of parametric polynomials using a simple example (without employing the Gröbner system). Next, we recall the definition of a Gaussian elimination form for a parametric matrix and compute it accordingly for a parametric matrix. These concepts are crucial in designing the GJS algorithm for computing the Gauss-Jordan form of a matrix with parametric entries. The LDS algorithm serves as a vital tool for solving the parametric linear dependency check problem. More precisely, let us consider $g \in \mathcal{S}$ as a parametric linear polynomial and a triple (N, W, G) such that $G \subset \mathcal{S}$ is a linear reduced Gröbner basis w.r.t. a given monomial ordering according to the constraint sets $N, W \subset \mathbb{K}[\mathbf{a}]$. The question that may arise is: how to characterize the dependency of g on the triple (N, W, G)? For this purpose, the LDS algorithm is described in [6]. The linear dependency system of the following example is computed utilizing our `Maple` implementation of the LDS algorithm.

Example 3.1. Consider $(N, W) = ([m, n], [a - 1])$ as a specification, $G = [u + v, by - cz, bx - acz]$ a linear Gröbner basis, and $g = av - bw + mz + nx - u - y$ a parametric polynomial. We fix the monomial orderings $n \prec_{lex} m \prec_{lex} d \prec_{lex} c \prec_{lex} b \prec_{lex} a$ and $w \prec_{lex} v \prec_{lex} u \prec_{lex} z \prec_{lex} y \prec_{lex} x$ on the parameters and the variables, respectively. So a linear dependency system of g on G acording to (N, W) is as follow:

$$\text{Sys} = \begin{cases} ([m,n],[c,a-1], & [false,[-1,\frac{-1}{b},0],-\frac{cz}{b}+(a+1)v-bw]), \\ ([m,n,-c],[a-1,a+1], & [false,[-1,\frac{-1}{b},0],-\frac{cz}{b}+(a+1)v-bw]), \\ ([m,n,-c,a+1],[b], & [false,[-1,\frac{-1}{b},0],-\frac{cz}{b}-bw]), \\ ([m,n,-c,a+1,b],[\ ], & [true,[-1,\frac{-1}{b},0],0]). \end{cases}$$

Using the LDS algorithm, we re-explain the Gaussian elimination form of parametric matrices (so-called Gaussian elimination systems [6, Definition 10] and also GES algorithm to compute these systems [6, Algorithm 3]).

Definition 3.2. Let A be a parametric matrix over $\mathbb{K}[\mathbf{a}] = \mathbb{K}[a_1, \ldots, a_m]$. A finite set of triples $\{(N_i, W_i, M_i)\}_{i=1}^{\ell}$ is called a *Gaussian elimination system* of A if for each specialization $\sigma : \mathbb{K}[a] \to \bar{\mathbb{K}}$, there is an i such that the following conditions hold

- The matrix $\sigma(M_i)$ is a Gaussian elimination form of $\sigma(A)$,
- $\sigma(p) = 0$ for each $p \in N_i$ and $\sigma(q) \neq 0$ for each $q \in W_i$.
- $\bigcup_{i=1}^{\ell} \mathbb{V}(N_i) \setminus \mathbb{V}(\prod_{w \in W_i} w) = \overline{\mathbb{K}}^m$

In the following, we compute the Gaussian elimination system for a parametric matrix utilizing our `Maple` implementation of the efficient GES algorithm [6, Algorithm 3].

Example 3.3. Let us consider the parametric matrix A in example 2.3. We get the following Gaussian elimination system of A using our `Maple` implementation of the above GES algorithm.

$$\left\{\begin{array}{lll}
[\,], & [a+1, c+b+a+abc], & \begin{bmatrix} 1 & -1 & c \\ 0 & a+1 & -1-ac \\ 0 & 0 & a+b+c+abc \end{bmatrix} \\
[a+1], & [b-1, c-1], & \begin{bmatrix} 1 & -1 & 1 \\ 0 & b-1 & 2 \\ 0 & 0 & c-1 \end{bmatrix} \\
[c-1, a+1], & [b-1], & \begin{bmatrix} 1 & -1 & 1 \\ 0 & b-1 & 2 \\ 0 & 0 & 0 \end{bmatrix} \\
[b-1, a+1], & [\,], & \begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \\
[c+b+a+abc], & [a+1], & \begin{bmatrix} 1 & -1 & c \\ 0 & a+1 & -1-ac \\ 0 & 0 & 0 \end{bmatrix}.
\end{array}\right.$$

A matrix is said to be in reduced row echelon form (rref) when it is in row echelon form and vectors having one entry equal to 1 and all the other entries equal to 0. The standard algorithm used to transform a matrix into a reduced row echelon form is called Gauss-Jordan elimination. However, the Gauss-Jordan elimination of a parametric matrix has not, to our knowledge, been studied in the literature. On the other hand, many problems in science and engineering can be modeled by parametric matrices, and have to be repeatedly solved for different values of parameters. So, the Gauss-Jordan of a parametric matrix may have a wide range of applications. The following example shows that the traditional approach for computing the Gauss-Jordan form of matrices may not be used for such a matrix.

Example 3.4. Let

$$A = \begin{bmatrix} a-1 & 0 & c-2 & 1 \\ 2 & 0 & -1 & b-1 \\ a & b+c & 0 & -1 \end{bmatrix}$$

where a, b and c are parameters in the real numbers. The reduced row echelon form of A computed by the function `MTM:-rref(A)` of `Maple` is equal to

$$B = \begin{bmatrix} 1 & 0 & 0 & \frac{bc-2b-c+3}{a-5+2c} \\ 0 & 1 & 0 & \frac{abc-2ab-ac+4a+2c-5}{(a-5+2c)(b+c)} \\ 0 & 0 & 1 & -\frac{ab-a-b-1}{a-5+2c} \end{bmatrix}$$

which is a matrix of rank 3 for all values of parameters. However, this is undefined for the values for the values $a = 3, b = 2$ and $c = 1$, where the Gauss-Jordan form of $A|_{\{a=3,b=2,c=1\}}$ is the following Matrix with rank 2.

$$\begin{bmatrix} 1 & 0 & -\frac{1}{2} & \frac{1}{2} \\ 0 & 1 & \frac{1}{2} & -\frac{5}{6} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

In fact, one may be interested in finding what values of parameters, the Gauss-Jordan form of the above matrix is of rank 2 and for what values it is of rank 3. In doing so, we introduce the notion of the Gauss-Jordan system for a parametric matrix, and we present an algorithm for computing it.

Definition 3.5. Let A be a parametric matrix over $\mathbb{K}[a_1, \ldots, a_m]$. A triples set $\{(N_i, W_i, M_i)\}_{i=1}^{\ell}$ is called a *Gauss-Jordan system* of A if

- $\bigcup_{i=1}^{\ell} \mathbb{V}(N_i) \setminus \mathbb{V}(\prod_{w \in W_i} w) = \overline{\mathbb{K}}^m$
- for each specialization $\sigma : \mathbb{K}[\mathbf{a}] \to \bar{\mathbb{K}}$, there is an i such that $\sigma(M_i)$ is a Gauss-Jordan form of $\sigma(A)$.

Using the LDS and GES algorithms proposed in [6], we present an efficient algorithm to compute a Gauss-Jordan system for parametric matrices. Below, given an $n \times m$ parametric matrix M over $\mathbb{K}[\mathbf{a}]$, we associate a linear parametric polynomial system $F \subset \mathcal{S}$.

Algorithm 1 GJS (Gauss-Jordan System)

Require: M; a parametric matrix over $\mathbb{K}[\mathbf{a}]$
Ensure: A Gauss-Jordan system for M
  Compute a Gaussian Elimination System $\{(N_i, W_i, M_i)\}_{i=1}^{\ell}$ for M using GES algorithm [6]
  Sys:= { }
  for i **from** 1 **to** ℓ **do**
    $B_i :=$ the reduced row echelon form of M_i
    Sys :=Sys $\bigcup$ $\{(N_i, W_i, B_i)\}$
  end for
  Return(Sys)

Example 3.6. Consider the matrix A mentioned in example 2.3. Using our `Maple` implementation of the GJS algorithm we obtain the following Gauss-Jordan system of A.

$$\begin{cases} [\ ], & [a+1, c+b+a+abc], & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ [a+1], & [b-1, c-1], & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ [c-1, a+1], & [b-1], & \begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \\ [b-1, a+1], & [\ ], & \begin{bmatrix} 1 & 0 & \frac{b+1}{b-1} \\ 0 & 1 & \frac{2}{b-1} \\ 0 & 0 & 0 \end{bmatrix} \\ [c+b+a+abc], & [a+1], & \begin{bmatrix} 1 & 0 & \frac{c-1}{a+1} \\ 0 & 1 & -\frac{ac+1}{a+1} \\ 0 & 0 & 0 \end{bmatrix}. \end{cases}$$

In the following, we introduce and compute an inverse matrix system using an extended version of the Gauss-Jordan method called the Gauss-Jordan system. This method is used to handle matrices with parametric entries, which are crucial in modeling systems that are subject to change or variability. Computing the inverse of such matrices is essential for solving systems of linear equations and gaining insights into the behavior of parametric systems. Therefore, it is appropriate to apply the Gauss-Jordan method to compute the inverse of matrices with parametric entries. Similar to matrices with constant arrays, to compute the inverse of a parametric matrix $A_{n \times n}$, we begin by appending the identity matrix $I_{n \times n}$ to the right side of A, creating the matrix $[A|I]$. We then apply the GJS algorithm to this matrix to compute a Gauss-Jordan system of $[A|I]$. At the end of this procedure, we obtain a set of triples in which the right side of the resulting matrices $n \times 2n$ is A^{-1}, while the state is "Invertible". To address this, we present the IMS algorithm, which is an extension of the Gauss-Jordan method, to compute the inverse matrix system of parametric matrices. Below, we define the variable Sys as an empty set, which will ultimately represent the output inverse matrix system. It is important to note that each recorded triple in Sys has the form (N, W, Singular or A^{-1}), where (N, W) represents a pair of conditions and A^{-1} is the inverse matrix of A according to (N, W) in non-singular cases.

Algorithm 2 IMS (Inverse Matrix System)

Require: $A_{n\times n}$; a parametric matrix over $\mathbb{K}[\mathbf{a}]$
Ensure: A inverse matrix system for A
  $M_{n\times 2n} := [A_{n\times n}|I_{n\times n}]$
  $\{(N_i, W_i, M_i)\}_{i=1}^{\ell}$:= a Gauss-Jordan System for M using GJS algorithm
  Sys:= { }
  for i **from** 1 **to** ℓ **do**
    L_i := The left sub-matrix of M with order n
    R_i := The right sub-matrix of M with order n
    if $L_i = I_{n\times n}$ **then**
      Sys :=Sys $\bigcup$ $\{(N_i, W_i, R_i)\}$
    else
      Sys :=Sys $\bigcup$ $\{(N_i, W_i, \text{"Singular"})\}$
    end if
  end for
  Return(Sys)

Theorem 3.7. *The* IMS *algorithm terminates after a finite number of steps and accurately computes inverse matrix systems.*

Proof. The algorithm's termination is guaranteed by the finiteness of the GJS algorithm. The correctness of the algorithm is ensured by the correctness of the LDS algorithm and the reliability of the Gauss-Jordan method for computing inverse matrices with constant arrays. Specifically, the algorithm applies the Gauss-Jordan method to compute the inverse of matrices with parametric entries. It examines the parameter space using the LDS algorithm and determines the invertibility or singularity of matrices based on the parametric constraints. If the left sub-matrix M is equal to the identity matrix I, then the triple (N_i, W_i, R_i) is added to the global variable Sys, where R_i represents the right sub-matrix M. On the other hand, if the left sub-matrix M is not equal to the identity matrix I, the triple $(N_i, W_i, \text{"Singular"})$ is added to the global variable Sys, indicating that A is singular based on the corresponding conditions pair (N_i, W_i). □

The ability to compute the inverse of parametric matrices (inverse matrix systems) provides a valuable tool for solving systems of linear equations that involve parametric coefficients. This, in turn, opens up a wide range of possibilities for analyzing and predicting the behavior of systems that have varying parameters. In fields such as engineering, physics, and economics, there are numerous practical examples where matrices with parametric entries and their inverses play a critical role. These examples serve as a testament to the versatility and applicability of the Gauss-Jordan method in parametric modeling, as it effectively addresses real-world challenges and provides meaningful insights. We illustrate the behavior of this algorithm with a simple example.

Example 3.8. Let us consider the parametric matrix

$$A = \begin{bmatrix} b-1 & 2 & a \\ 1 & -n+1 & b+1 \\ 0 & cm & -1 \end{bmatrix}$$

where a, b, c, m, and n are parameters. The following inverse matrix system of A, computed using our `Maple` implementation of the IMS algorithm, consists of 8 branches. The computation time required is 0.37 seconds, and it utilizes 0.06 gigabytes of memory. These results, including timing and memory usage, were obtained on a personal computer running `Windows 10 (64 bits)` with `Ryzen 6800` and `8 GB RAM`.

Parametric Constraints	Inverse Matrix System (IMS)
$N_1 = [\,], W_1 = [b-1, bn-b-n+3,$ $(-b^2+a+1)cm+bn-b-n+3]$	$\begin{bmatrix} \frac{n-bcm-cm-1}{(-b^2+a+1)cm+bn-b-n+3} & \frac{acm+2}{(-b^2+a+1)cm+bn-b-n+3} & \frac{an-a+2b+2}{(-b^2+a+1)cm+bn-b-n+3} \\ \frac{1}{(-b^2+a+1)cm+bn-b-n+3} & \frac{1-b}{(-b^2+a+1)cm+bn-b-n+3} & \frac{-b^2+a+1}{(-b^2+a+1)cm+bn-b-n+3} \\ \frac{cm}{(-b^2+a+1)cm+bn-b-n+3} & \frac{cm(1-b)}{(-b^2+a+1)cm+bn-b-n+3} & \frac{-bn+b+n-3}{(-b^2+a+1)cm+bn-b-n+3} \end{bmatrix}$
$N_2 = [bn-b-n+3]$, $W_2 = [c, m, b-1, -b^2+a+1]$	$\begin{bmatrix} -\frac{b^2cm-cm+2}{cm(-b^2+a+1)(b-1)} & \frac{acm+2}{cm(-b^2+a+1)} & -2\frac{1}{cm(b-1)} \\ \frac{1}{cm(-b^2+a+1)} & -\frac{b-1}{cm(-b^2+a+1)} & \frac{1}{cm} \\ (-b^2+a+1)^{-1} & -\frac{b-1}{-b^2+a+1} & 0 \end{bmatrix}$
$N_3 = [b-1], W_3 = [acm+2]$	$\begin{bmatrix} -\frac{2cm-n+1}{acm+2} & 1 & \frac{an-a+4}{acm+2} \\ (acm+2)^{-1} & 0 & \frac{a}{acm+2} \\ \frac{cm}{acm+2} & 0 & -2(acm+2)^{-1} \end{bmatrix}$
$N_4 = [b-1, acm+2], W_4 = [c, m]$	Singular
$N_5 = [cm, bn-b-n+3, -b^2+a+1]$, $W_5 = [b-1]$	Singular
$N_6 = [bn-b-n+3, -b^2+a+1]$, $W_6 = [c, m, b-1]$	Singular
$N_7 = [cm, bn-b-n+3]$, $W_7 = [b-1, -b^2+a+1]$	Singular
$N_8 = [b^2cm-acm-bn-cm+b+n-3]$, $W_8 = [c, m, b-1, bn-b-n+3]$	Singular

Acknowledgment

I genuinely appreciate the valuable comments and suggestions provided by the reviewers, as they have greatly contributed to enhancing the manuscript's quality.

References

[1] Buchberger, B. Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal. Translation from the German. *J. Symb. Comput. 41*, 3-4 (2006), 475–511.

[2] Cox, D., Little, J., and O'Shea, D. *Ideals, varieties, and algorithms. An introduction to computational algebraic geometry and commutative algebra. 3rd ed.* New York, NY: Springer, 2007.

[3] Dehghani Darmian, M. Gauss-Jordan Systems. In *Proceedings of the 4th Conference on Computational Algebra, Computational Number Theory, and Applications, CACNA 2023, Kashan, Iran, July 4–6, 2023.*

[4] Dehghani Darmian, M., and Hashemi, A. Generalization of the F_4 Algorithm to Parametric Polynomial Ideals. In *Proceedings of the 7th International Conference on Combinatorics, Cryptography, Computer Science and Computation, Tehran, Iran 2022, I4C 2022, Tehran, Iran, November 16–17, 2022.*

[5] Dehghani Darmian, M., and Hashemi, A. Zero-dimensional Parametric Gröbner Bases Conversion. In *Proceedings of the 7th International Conference on Combinatorics, Cryptography, Computer Science and Computation, Tehran, Iran 2022, I4C 2022, Tehran, Iran, November 16–17, 2022.*

[6] Dehghani Darmian, M., and Hashemi, A. Parametric fglm algorithm. *J. Symb. Comput. 82* (2017), 38–56.

[7] Dehghani Darmian, M., Hashemi, A., and Montes, A. Erratum to "A new algorithm for discussing Gröbner bases with parameters" [J. Symbolic Comput. 33 (1-2) (2002) 183-208]. *J. Symb. Comput. 46*, 10 (2011), 1187–1188.

[8] Hashemi, A., Dehghani Darmian, M., and Barkhordar, M. Gröbner systems conversion. *Math. Comput. Sci. 11*, 1 (2017), 61–77.

[9] Hashemi, A., M.-Alizadeh, B., and Dehghani Darmian, M. Computing comprehensive Gröbner systems: a comparison of two methods. *Comput. Sci. J. Mold. 25*, 3(75) (2017), 278–302.

[10] Kapur, D., Sun, Y., and Wang, D. A new algorithm for computing comprehensive Gröbner systems. In *Proceedings of the 35th international symposium on symbolic and algebraic computation, ISSAC 2010, Munich, Germany, July 25–28, 2010.* New York, NY: Association for Computing Machinery (ACM), 2010, pp. 29–36.

[11] Montes, A. A new algorithm for discussing Gröbner bases with parameters. *J. Symb. Comput. 33*, 2 (2002), 183–208.

[12] Montes, A., and Wibmer, M. Gröbner bases for polynomial systems with parameters. *J. Symb. Comput. 45*, 12 (2010), 1391–1425.

[13] Nabeshima, K. A speed-up of the algorithm for computing comprehensive Gröbner systems. In *Proceedings of the 2007 international symposium on symbolic and algebraic computation, ISSAC 2007, Waterloo, ON, Canada, July 29–August 1, 2007*. New York, NY: Association for Computing Machinery (ACM), 2007, pp. 299–306.

[14] Suzuki, A., and Sato, Y. A simple algorithm to compute comprehensive Gröbner bases using Gröbner bases. In *Proceedings of the international symposium on symbolic and algebraic computation, ISSAC 06, Genova, Italy, July 9–12, 2006*. New York, NY: ACM Press, 2006, pp. 326–331.

[15] Weispfenning, V. Comprehensive Gröbner bases. *J. Symb. Comput. 14*, 1 (1992), 1–29.

e-mail: `m.dehghanidarmian@gmail.com`
e-mail: `m.dehghanidarmian@ipm.ir`