An Algorithm for the Computation of the Generalized Inverse and its Implementation via MAPLE

J.Jones†, N.P.Karampetakis‡ and A.C.Pugh†

†*Department of Mathematical Sciences*
Loughborough University of Technology
Loughborough, Leics
England, U.K.

‡*Department of Mathematics*
Aristotle University of Thessaloniki
54006 *GREECE*

ABSTRACT

Given a constant non–singular matrix $A \in \mathbb{R}^{n\times n}$ (i.e. A is square and possesses a non–zero determinant) then it is well known that there exists a unique matrix $B \in \mathbb{R}^{n\times n}$ such that

$$AB = BA = I_n \tag{1}$$

where I_n denotes the $(n \times n)$ identity matrix. Here B is called the *inverse* of A and is denoted by A^{-1}. However consider the singular matrix $A \in \mathbb{R}^{n\times m}$ (i.e. A is non–square or square with zero determinant). Then although there may exist matrices $C \in \mathbb{R}^{m\times n}$ and $D \in \mathbb{R}^{m\times n}$ such that $AC = I_n$ and $DA = I_m$ there exists no matrix B that satisfies (1). (Clearly the requirement for C to exist is that A has full row rank while the existence of D requires that A has full column rank). Here C and D are termed the right and left inverses of A respectively.

Consider the system of consistent linear equations

$$Ax = b \tag{2}$$

where $A \in \mathbb{R}^{n\times m}, x \in \mathbb{R}^m$ and $b \in \mathbb{R}^n$. If A is non–singular then (2) has a *unique* solution namely

$$x = A^{-1}b \tag{3}$$

However in many cases where A is singular (i.e. A^{-1} does *not* exist) solutions of a system of linear equations as in (2) may still exist. Such solutions can be computed by introducing what is known as the *generalized inverse* of A denoted by $A^{\dagger}$.

Definition 1 *(Penrose [9])*
The generalized inverse of a constant matrix $A \in \mathbb{R}^{n\times m}$ is defined as the matrix $G \in \mathbb{R}^{m\times n}$ which satisfies the following conditions

i) $AGA = A$

ii) $GAG = G$ (4)

iii) $(AG)^{\star} = AG$

iv) $(GA)^\star = GA$

where $[\,\cdot\,]^\star$ denotes the conjugate transpose of the indicated matrix. Such a matrix G is *unique* and is denoted by $A^\dagger$. Further, such a matrix *always* exists. □

In the special case where the matrix A is square and non-singular it is clear from (4) that the generalized inverse of A is simply its inverse. i.e. $A^\dagger \equiv A^{-1}$.

The problem of obtaining such a generalized inverse has been considered by many authors [2], [4]–[9]. This is due to the large number of implications that the problem has in linear systems theory which include, for example, the solution of linear system equations, the calculation of the transfer function matrix of a multivariable system [4] and the solution of diophantine equations [5].

Recently Karampetakis [5] has extended a procedure, originally given in Decell [2] for the constant matrix case, to compute the generalized inverse of a matrix $A(s) \in \mathbb{R}[s]^{n\times m}$ (i.e. $A(s)$ is defined over the ring of polynomials in s). However, a procedure for the computation of the generalized inverse of $A(s) \in \mathbb{R}(s)^{n\times m}$ (i.e. $A(s)$ is defined over the field of rational functions) remains an open question. It is clear that for the cases when $A(s) \in \mathbb{R}[s]^{n\times m}$ and $A(s) \in \mathbb{R}(s)^{n\times m}$, the generalized inverse is defined as the unique matrix which satisfies the properties in (4). It is noted in this paper that the procedure [5] for the case of polynomial matrices equally holds for $A(s) \in \mathbb{R}(s)^{n\times m}$ when properly extended. This extended result is given below

Theorem 1 (Generalized Inverse of a Rational Matrix)

Let $A(s) \in \mathbb{R}(s)^{n\times m}$ and let $f(\lambda, s)$ denote the characteristic polynomial of $A(s)A(s)^T$ where $A(s)^T$ denotes the transpose of $A(s)$

i.e.

$$\begin{aligned} f(\lambda,s) &\stackrel{\text{def}}{=} det\left[\lambda I_n - A(s)A(s)^T\right] \\ &= (-1)^n\left(a_0(s)\lambda^n + a_1(s)\lambda^{n-1} + \cdots + a_{n-1}(s)\lambda + a_n(s)\right), \quad a_0(s) = 1 \end{aligned} \tag{5}$$

Let $a_n(s) = \ldots = a_{k+1}(s) = 0$ while $a_k(s) \neq 0$.

Then the generalized inverse $A(s)^\dagger \in \mathbb{R}(s)^{m\times n}$ of $A(s)$ is given by

$$A(s)^\dagger = -a_k(s)^{-1}A(s)^T\left[\left(A(s)A(s)^T\right)^{k-1} + a_1(s)\left(A(s)A(s)^T\right)^{k-2} + \cdots + a_{k-1}(s)I\right] \tag{6}$$

If $k = 0$ $\left(\text{i.e. } a_n(s) = \ldots = a_1(s) = 0\right)$ then $A(s)^\dagger = 0_{m,n}$, the $(m \times n)$ zero matrix. □

In an analogous way to that presented in [2] a recursive algorithm can be produced to form the generalized inverse of $A(s)$ as given in (6). This is a direct consequence of the proof of Theorem 1 and a modification to Leverrier's method [see 3] and is given below.

Algorithm 1(Algorithm for the Computation of the Generalized Inverse)

Step 1:

Let $A(s) \in \mathbb{R}(s)^{n\times m}$. Form the following sequences $\left[A_0(s), A_1(s), \ldots, A_n(s)\right]$, $\left[a_0(s), a_1(s), \ldots, a_n(s)\right]$, and $\left[B_0(s), B_1(s), \ldots, B_n(s)\right]$ which are constructed in the following recursive way.

$$\begin{array}{lll}
A_0(s) = 0 & a_0(s) = 1 & B_0(s) = I_n \\
A_1(s) = \left(A(s)A(s)^T\right) B_0(s) & a_1(s) = -\dfrac{trace\,(A_1(s))}{1} & B_1(s) = A_1(s) + a_1(s)I_n \\
\vdots & \vdots & \vdots \\
A_n(s) = \left(A(s)A(s)^T\right) B_{n-1}(s) & a_n(s) = -\dfrac{trace\,(A_n(s))}{n} & B_n(s) = A_n(s) + a_n(s)I_n
\end{array} \tag{7}$$

Step 2:

Let $a_n(s) = \ldots = a_{k+1}(s) = 0$ while $a_k(s) \neq 0$. Then the generalized inverse $A(s)^\dagger$ of $A(s)$ is given by

$$A(s)^\dagger = -a_k(s)^{-1} A(s)^T B_{k-1}(s) \tag{8}$$

If $k = 0$ $\Big($ i.e. $a_n(s) = \ldots = a_1(s) = 0\Big)$ then $A(s)^\dagger \stackrel{\text{def}}{=} 0_{m,n}$, the $(m \times n)$ zero matrix. □

The algorithm obtained for generating the generalized inverse of a matrix $A(s) \in \mathbb{R}(s)^{n\times m}$ is very computationally attractive for several reasons.

i) It is a simple recursion where each of the 3 sequences are updated at each step. The storage requirement can also be reduced by noting that the sequence $\left[A_0(s), A_1(s), \ldots A_n(s)\right]$ is not needed in the final result (8). Therefore, only the storage space for the first matrix A_0 is required. This is then updated at each successive step for $A_1, A_2, \ldots$. This saving would become more significant as n increased.

ii) There is no inversion of any of the matrices involved and therefore the algorithm can be considered stable in this respect. The only inversion required is of a rational function term $a_k(s)$. If $a_k(s) = 0 \Leftrightarrow A(s)^\dagger = 0$ and therefore we do not encounter the possibility of dividing by zero in the algorithm.

iii) The dimension of the matrix sequences involved, i.e. the dimensions of $\left[A_0(s), A_1(s), \ldots, A_n(s)\right]$, $\left[B_0(s), B_1(s), \ldots, B_n(s)\right]$, remain fixed throughout and do not increase at any step. Therefore no computational problems arise when n is decreed large.

iv) The algorithm equally holds when $A(s)$ is polynomial or indeed constant. Thus the proposed computer procedure covers all forms of $A(s)$.

v) The same algorithm may be used for n-variable rational matrices[6].

Subsequently, in the paper, we present a computer procedure based on Algorithm 1 above. The program is written in the symbolic computational language MAPLE [1]. Using MAPLE enables the user to implement any of the "built in" procedures inherent in it predominantly, in this case, the linear algebra package `linalg` which contains a collection of procedures for matrix manipulation. This results in further simplification of the program code required.

As an application of the generalized inverse procedure given above consider the solution of the equation

$$A(s)X(s)B(s) = C(s) \tag{9}$$

where $A(s) \in \mathbb{R}(s)^{n\times m}$, $X(s) \in \mathbb{R}(s)^{m\times k}$, $B(s) \in \mathbb{R}(s)^{k\times l}$ and $C(s) \in \mathbb{R}(s)^{n\times l}$. The case where $A(s), X(s), B(s), C(s)$ are constant was considered by [9] and this is extended to the case (9)

Theorem 2 (Solution of the equation $A(s)X(s)B(s) = C(s)$)

(9) has a solution iff

$$A(s)A(s)^{\dagger}C(s)B(s)^{\dagger}B(s) = C(s) \tag{10}$$

in which case all the solutions are given by

$$X(s) = A(s)^{\dagger}C(s)B(s)^{\dagger} + Y(s) - A(s)^{\dagger}A(s)Y(s)B(s)B(s)^{\dagger} \tag{11}$$

where $A(s)^{\dagger}, B(s)^{\dagger}$ are the generalized inverses of $A(s)$ and $B(s)$ respectively and $Y(s)$ is arbitrary to within having the dimension of $X(s)$. □

In the paper a MAPLE implementation of this result, utilizing Algorithm 1, is given which provides a solution to (11) whenever (10) is satisfied.

Consider the case when $B(s) = I_k, l = k$. i.e. (9) reduces to

$$A(s)X(s) = C(s) \tag{12}$$

For this equation there exists a procedure `linsolve` inherent within MAPLE to compute such a solution. The procedure presented in this paper when applied to (12) contains the complete free parameter set in the solution $X(s)$ while the `linsolve` procedure generally gives an underparametrized solution. However, an advantage of the `linsolve` procedure is that it can be used independently of the `linalg` package in contrast to our procedure which is strongly dependent on it. These issues will be addressed futher in the paper.

In summary, we extend an existing algorithm for the construction of a generalized inverse of a non–regular polynomial matrix $A(s) \in \mathbb{R}[s]^{n\times m}$ to that of the rational case. This clearly widens the number of applications of the generalized inverse in linear systems theory. This algorithm has been implemented in the symbolic computational language MAPLE. As a direct application we consider the solution of the rational equation $A(s)X(s)B(s) = C(s)$ which has also been implemented in MAPLE. In particular the solution of the equation $A(s)X(s) = C(s)$ via our procedure can be favorably compared to the existing procedure `linsolve` inherent within MAPLE.

References

[1] Char, B.W., Geddes, K.G., Gonnet, G.H. and Watt, S.M. (1991). *Maple V Language Reference Manual.* Springer-Verlag.

[2] Decell, H.P. (1965). An Application of the Cayley–Hamilton Theorem to generalized matrix Inversion. *SIAM Review.* **7** 526–528.

[3] Faddeev, D.K. and Faddeeva, V.N. (1963). Computational Methods of Linear Algebra. *W.H.Freeman Co., San Francisco.* 260–265.

[4] Fragulis, G., Mertzios, B.G. and Vardulakis, A.I.G. (1991). Computation of the Inverse of a Polynomial Matrix and Evaluation of its Laurent Expansion. *Int.J.Control.* **53** 431–443.

[5] Karampetakis, N.P. (1995). Computation of the Generalized Inverse of a Polynomial Matrix and Applications. To appear in Linear Algebra and Applications.

[6] Karampetakis, N.P. (1995). Generalized Inverses of Two Variable Polynomial Matrices and Applications. *Mathematical Sciences Report No. A247, Loughborough University of Tech.*

[7] Lovass Nagy, V., Miller, V. and Powers, D. (1976). Research Note: On the Application of Matrix Generalized Inverses to the Construction of Inverse Systems. *Int.J.Control.* **24** 733–739.

[8] Mertzios, B.G. and Lewis, F.L. (1989). Fundamental Matrix of Singular Systems. *Circuit Systems & Signal Processing* **8** 341–355.

[9] Penrose, R. (1955). A Generalized Inverse for Matrices. *Proc. Cambridge Philos. Soc.* **51** 406–413.