The Moore-Penrose inverse of a vector: Coping with a sometimes tricky case differentiation

KARSTEN SCHMIDT

Schmalkalden University of Applied Sciences, Germany
e-mail: kschmidt@fh-sm.de

1 Introduction

For any matrix $A \in \mathbb{R}^{m \times n}$ a unique matrix $A^+ \in \mathbb{R}^{n \times m}$ exists, which satisfies the following four conditions

$$AA^+ A = A \quad (1)$$
$$A^+ A A^+ = A^+ \quad (2)$$
$$(A^+ A)' = A^+ A \quad (3)$$
$$(AA^+)' = AA^+ \quad (4)$$

Note that $A^+$ has the same dimension as $A'$, the transpose of $A$.

If $A$ is square and nonsingular, its inverse $A^{-1}$ exists. Obviously, in this case the above conditions (1) to (4) are satisfied when $A^{-1}$ is substituted for $A^+$. Hence, if $A$ is a nonsingular matrix, we have $A^+ = A^{-1}$. 
2 Computation of the Moore-Penrose inverse of a vector

The Moore-Penrose inverse of a (column) vector $a \in \mathbb{R}^n$ is given by

$$a^+ = \begin{cases} \frac{1}{a^\prime a} a^\prime & \text{if } a \neq o \\ o^\prime & \text{if } a = o \end{cases}$$  \hspace{1cm} (5)

where $o$ denotes the $(n \times 1)$ zero vector. Apparently, $a^+$ is a row vector.

Since a vector is nothing else but a matrix with only one column, it should be declared in Derive as such. The following MPIV function has no problem computing the Moore-Penrose inverse of any vector which contains numbers only,

```plaintext
MPIV(a):=
  IF DIM(a^\prime) = 1
    IF (a^\prime a)111 = 0
      0 \cdot a^\prime
    a^\prime/(a^\prime a)111
    "This is not a column vector!"
```

but might not be able to compute the Moore-Penrose inverse of a vector which has non-numeric elements. To illustrate this, we define the following set of vectors in Derive

$$a = \begin{pmatrix} 0 \\ 2 \end{pmatrix}; \quad b = \begin{pmatrix} x \\ 2 \end{pmatrix}; \quad c = \begin{pmatrix} 0 \\ x \end{pmatrix}; \quad o = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

```plaintext
#2: a = \begin{pmatrix} 0 \\ 2 \end{pmatrix}

#3: b = \begin{pmatrix} x \\ 2 \end{pmatrix}

#4: c = \begin{pmatrix} 0 \\ x \end{pmatrix}

#5: o = \begin{pmatrix} 0 \\ 0 \end{pmatrix}
```
Since $a'a = 4 \neq 0$, the Moore-Penrose inverse of $a$ is

$$a^+ = \frac{1}{a'a}a' = \frac{1}{4}(0 \quad 2) = \begin{pmatrix} 0 \\ -\frac{1}{2} \end{pmatrix}$$

As $b'b = x^2 + 4$, $b'b > 0$ for any $x \in \mathbb{R}$, such that

$$b^+ = \frac{1}{bb}b' = \frac{1}{x^2+4}(x \quad 2) = \begin{pmatrix} x \\ \frac{2}{x^2+4} \end{pmatrix}$$

As $c'c = x^2$, we have $c'c = 0$ for $x = 0$, and $c'c > 0$ otherwise. Therefore

$$c^+ = \begin{cases} \frac{1}{c'c}x' = \frac{1}{x}(0 \quad x) = \begin{pmatrix} 0 \\ \frac{1}{x} \end{pmatrix} & \text{if } x \neq 0 \\ o' & \text{if } x = 0 \end{cases}$$

Since $o'o = 0$, the Moore-Penrose inverse of $o$ is

$$o^+ = o'$$

Let us now see how the MPIV function copes with these vectors:

<table>
<thead>
<tr>
<th>Step</th>
<th>MPIV(a)</th>
<th>MPIV(b)</th>
<th>MPIV(c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>#6</td>
<td>$\begin{pmatrix} 0, &amp; \frac{1}{2} \end{pmatrix}$</td>
<td>$\begin{pmatrix} x, &amp; \frac{2}{x^2+4} \ \frac{2}{x^2+4}, &amp; \frac{2}{x^2+4} \end{pmatrix}$</td>
<td>$\begin{pmatrix} x, \frac{2}{x^2+4} \ \frac{2}{x^2+4}, \frac{2}{x^2+4} \end{pmatrix}$</td>
</tr>
<tr>
<td>#7</td>
<td></td>
<td></td>
<td>$\text{IF}(x = 0, 0, \begin{pmatrix} 0 \ x \end{pmatrix}, \begin{pmatrix} 0 \ x \end{pmatrix}, \begin{pmatrix} 0 \ x \end{pmatrix}, \begin{pmatrix} 0 \ x \end{pmatrix})^{-1}$</td>
</tr>
<tr>
<td>#8</td>
<td></td>
<td></td>
<td>$\begin{pmatrix} 1, &amp; 1 \end{pmatrix}$</td>
</tr>
<tr>
<td>#9</td>
<td></td>
<td></td>
<td>$[0, \ 0]$</td>
</tr>
</tbody>
</table>

The MPIV function finds $a^+$, $b^+$ and $o^+$, but fails to compute the Moore-Penrose inverse of $c$. 
3 Computation of the Moore-Penrose inverse of a matrix

For the computation of the Moore-Penrose inverse of a matrix we apply Greville’s method, which is an iterative algorithm that needs $n$ steps for the computation of the Moore-Penrose inverse of an $m$ by $n$ matrix. The MPI function given below calls the MPIV function in each step. Hence, the MPI function might be unable to compute the Moore-Penrose inverse of a matrix if it has at least one non-numeric element.

To illustrate this, we define the following four matrices in Derive

\[
A = \begin{pmatrix} 1 & 0 \\ 2 & 0 \end{pmatrix}; \quad B = \begin{pmatrix} x & 0 \\ 2 & 0 \end{pmatrix}; \quad C = \begin{pmatrix} 1 & 0 \\ 2 & x \end{pmatrix}; \quad D = \begin{pmatrix} x & 0 \\ 0 & 0 \end{pmatrix}
\]
The MPI function finds $A^+$ and $B^+$, but fails to compute the Moore-Penrose inverse of $C$ and $D$.

Note that

$$ C^+ = \begin{cases} 
\begin{pmatrix} 1 & 0 \\ -\frac{2}{x} & \frac{1}{x} \end{pmatrix} & \text{if } x \neq 0 \\
\begin{pmatrix} \frac{1}{5} & \frac{2}{5} \\ 0 & 0 \end{pmatrix} & \text{if } x = 0
\end{cases} \quad \text{and} \quad D^+ = \begin{cases} 
\begin{pmatrix} \frac{1}{x} & 0 \\ 0 & 0 \end{pmatrix} & \text{if } x \neq 0 \\
\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \text{if } x = 0
\end{cases} $$
4 A way out

Let us consider the rank of the four vectors from section 2:

\[ r(a) = 1, \quad r(b) = 1 \quad \text{for any } x \in \mathbb{R}, \quad r(c) = \begin{cases} 1 & \text{if } x \neq 0 \\ 0 & \text{if } x = 0 \end{cases}, \quad r(o) = 0 \]

Obviously, the rank of \( c \) depends on \( x \). But in Derive

\[
\begin{align*}
#10: & \quad \text{RANK}(a) = 1 \\
#11: & \quad \text{RANK}(b) = 1 \\
#12: & \quad \text{RANK}(c) = 1 \\
#13: & \quad \text{RANK}(o) = 0
\end{align*}
\]

we get \( r(c) = 1 \), i.e. \( \text{Derive} \) does not make a case differentiation. Apparently, the single \( x \) value, which turns \( c \) into a zero vector, is neglected.

Let us now see how the MPIV0 function copes with the four vectors:

\[
\begin{align*}
\text{MPIV0(a)} & \quad = \begin{bmatrix} 0, & \frac{1}{2} \end{bmatrix} \\
\text{MPIV0(b)} & \quad = \begin{bmatrix} \frac{x}{2x + 4}, & \frac{2}{2x + 4} \end{bmatrix} \\
\text{MPIV0(c)} & \quad = \begin{bmatrix} 0, & \frac{1}{x} \end{bmatrix} \\
\text{MPIV0(o)} & \quad = \begin{bmatrix} [?, & ?] \end{bmatrix}
\end{align*}
\]

The MPIV0 function computes the Moore-Penrose inverse of \( c \) for \( x \neq 0 \). The special case \( x = 0 \) is ignored. However, it is unable to compute the Moore-Penrose inverse of \( o \).
Let us now consider the rank of the four matrices from section 3:

\[ r(A) = 1, \quad r(B) = 1 \text{ for any } x \in \mathbb{R}, \quad r(C) = \begin{cases} 2 & \text{if } x \neq 0 \\
1 & \text{if } x = 0 \end{cases}, \quad r(D) = \begin{cases} 1 & \text{if } x \neq 0 \\
0 & \text{if } x = 0 \end{cases} \]

Obviously, the value of \( x \) is crucial as to whether the second column of \( C \), and the first column of \( D \), are zero vectors or not. But in Derive

| #11: | RANK(A) = 1 |
| #12: | RANK(B) = 1 |
| #13: | RANK(C) = 2 |
| #14: | RANK(D) = 1 |

we get \( r(C) = 2 \), i.e. Derive again does not make a case differentiation. Apparently, the single \( x \) value, which turns the second column of \( C \), and the first column of \( D \), into zero vectors, is neglected.
The **MPI0** function is identical to the **MPI** function in section 3, except that it calls the **MPIV0** function instead of the **MPIV** function in lines 3 and 11. Let us now see how the **MPI0** function copes with the four matrices:

| #15:                     | **MPI0(A)** = \[
|                          | \begin{bmatrix}
|                          | \text{?} & \text{?} \\
|                          | \text{?} & \text{?} \\
|                          | \end{bmatrix}
| #16:                     | **MPI0(B)** = \[
|                          | \begin{bmatrix}
|                          | 1 & 0 \\
|                          | \text{?} & \text{?} \\
|                          | \end{bmatrix}
| #17:                     | **MPI0(C)** = \[
|                          | \begin{bmatrix}
|                          | \frac{1}{x} & \frac{1}{x} \\
|                          | - \frac{2}{x} & 1 \\
|                          | \end{bmatrix}
| #18:                     | **MPI0(D)** = \[
|                          | \begin{bmatrix}
|                          | \text{?} & \text{?} \\
|                          | \text{?} & \text{?} \\
|                          | \end{bmatrix}
|                          |

The **MPI0** function computes the Moore-Penrose inverse of **C** for \( x \neq 0 \). The special case \( x = 0 \) is ignored. However, it is unable to compute the Moore-Penrose inverses of **A**, **B**, and **D**.

Note that **C** is a nonsingular matrix for \( x \neq 0 \) such that \( C^+ = C^{-1} \). Computing the inverse of **C** in *Derive* generates the same matrix as the **MPI0** function, i.e. *Derive* is consistent in terms of disregarding the special case \( x = 0 \).

| #19:                     | \( C^{-1} = \[
|                          | \begin{bmatrix}
|                          | 1 & 0 \\
|                          | - \frac{2}{x} & \frac{1}{x} \\
|                          | \end{bmatrix}\)