Check for updates

Extending *Maxima* Capabilities for Performing Elementary Matrix Operations

Karina F. M. Castillo-Labán(✉) and Robert Ipanaqué-Chero

Universidad Nacional de Piura, Urb. Miraflores s/n, Castilla, Piura, Peru
karina17121995@gmail.com,ripanaquec@unp.edu.pe,
http://www.unp.edu.pe/pers/ripanaque

Abstract. The elementary operations on matrices are of great importance and utility since they are applied in obtaining the so-called equivalent matrices, which are related to each other in many important attributes. In this paper we describe a new package for dealing with the elementary matrix operations of a given matrix. The package, developed by the authors in the freeware program *Maxima*, version 5.43.2, incorporates two commands which have a fairly intuitive syntax and allow the user indicating one or more elementary operations together so that they are applied to the rows (or columns) of a given matrix. Such features make it more effective and friendly than the default built-in commands in *Maxima* to perform such operations. In addition, our outputs are consistent with *Maxima*'s notation. Several illustrative examples, aimed to show the good performance of the package, are also given.

Keywords: Elementary matrix operations · Row operations · Column operations · Maxima

1 Introduction

In addition to the usual operations defined in the $\mathcal{M}_{m\times n}$ set: addition, multiplication of a scalar, multiplication, and transposition, there are operations that are performed between the elements of some rows or columns of a matrix, called elementary operations. These operations are of great importance and utility because the application of one or more of them allows obtaining certain matrices called equivalent matrices. These matrices, due to their characteristics, are related to each other in many important attributes such as: range, determinant, solution of a system of linear equations and linear combination. And, of course, when considering an augmented matrix, they allow us to obtain the inverse of said matrix [8]. Regarding the inverse of a matrix, there is a very similar method and, more importantly, a "spirit" of methodology that also uses elementary matrix operations on rows and columns, but specializes in certain types of matrices [6,7].

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12249, pp. 406–420, 2020.
https://doi.org/10.1007/978-3-030-58799-4_30

Calculating elementary matrix operations can be tedious when they are of order $m \times n$, such that $m, n > 3$. Furthermore, there is a growing concern to develop commands (in Maple [10], Mathematica [11], MatLab [9], Maxima [4], etc.), applications (in Java [1,5]) and programs (Linear Algebra Decoded [12]) specialized in automating such calculation.

What has been expressed in the previous paragraphs, the fact that *Maxima* includes four commands (with two well-marked flaws: the syntax is not intuitive and they do not allow to indicate several operations together) and the fact that the *Maxima* is distributed under the GPL [14] license has been reason enough to choose it as the programming language in which to develop a new package in *Maxima*. This package, which dealing with the elementary matrix operations of a given matrix, is described in this paper. The package, developed by the authors in the freeware program *Maxima*, version 5.43.2, incorporates two commands which have a fairly intuitive syntax and allow the user indicating one or more elementary operations together so that they are applied to the rows (or columns) of a given matrix. The outputs obtained with the built-in commands in the new package are consistent with the *Maxima* syntax. Several illustrative examples will be discussed later on in order to show the good performance of the package.

The rest of this paper is organized as follows: Sect. 2 outlines concepts related to elementary operations with the rows (columns) of a given matrix. Then, Sect. 3 describes the main standard *Maxima* tools for performing elementary matrix operations. And Sect. 4 describes the new package. Finally, the conclusions are in Sect. 5.

2 Preliminaries

When making calculations with matrices (calculation of inverse, determinant, range, etc.), the operations of the following type are especially important:

1. Swap two rows (or columns).
2. Add a multiple of another row (or column) to one row(or column).
3. Multiply a row (or column) by a non-zero scalar.

They are simple operations, which conserve certain quantities associated with the matrix in which they are operated (example: the range) or modify them in a controlled way (example: the determinant).

It will be useful to use adapted notations to describe the elementary operations that are done on rows.

1. Swap rows i and j:
$$R_i \leftrightarrow R_j$$
(remark: it will be used only for $i \neq j$).
2. Add to row i row j multiplied by scalar λ:
$$R_i + \lambda R_j \rightarrow R_i$$
(remark: it will be used only for $i \neq j$).

3. Multiply row i by $\lambda \neq 0$:

$$\lambda R_i \to R_i\,.$$

Changing the R by the C in previous list, the notations to describe the elementary operations that are done on columns are obtained.

3 Standard *Maxima* Tools for Elementary Matrix Operations

Maxima includes the `linearalgebra` package, one of whose commands is `rowop` (similarly includes the `columnop` command) [4]. The syntax for this command is:

$$\texttt{rowop}(M, i, j, \lambda). \tag{1}$$

And it works like this: if M is a matrix, it returns the resulting matrix after doing the operation:

$$R_i - \lambda \cdot R_j \to R_i\,; \tag{2}$$

if M does not have a row i or j, it returns an error message.

Example 1. Find A^{-1} to

$$A = \begin{pmatrix} 3\ 2\ 1 \\ 4\ 5\ 2 \\ 2\ 1\ 4 \end{pmatrix}\,,$$

using the method of elementary operations on rows [2] with the assistance of *Maxima*.

Solution 1. Let's form the matrix $(A|I)$ and stored it in the variable AI:

```
(%i1) AI:matrix([3,2,1,1,0,0],[4,5,2,0,1,0],[2,1,4,0,0,1])$
```

Now, let's do:

Operation. $\frac{1}{3}R_1 \to R_1$.

```
(%i2) AI:rowop(AI,1,1,2/3);
```

```
(%o2)
```

$$\begin{pmatrix} 1 & \frac{2}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ 4 & 5 & 2 & 0 & 1 & 0 \\ 2 & 1 & 4 & 0 & 0 & 1 \end{pmatrix}$$

Operation. $R_2 - 4F_1 \to R_2$.

```
(%i3) AI:rowop(AI,2,1,4);
```

```
(%o3)
```

$$\begin{pmatrix} 1 & \frac{2}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ 0 & \frac{7}{3} & \frac{2}{3} & -\frac{4}{3} & 1 & 0 \\ 2 & 1 & 4 & 0 & 0 & 1 \end{pmatrix}$$

Operation. $R_3 - 2F_1 \to R_3$.

```
(%i4) AI:rowop(AI,3,1,2);
```

(%o4)

$$\begin{pmatrix} 1 & \frac{2}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ 0 & \frac{7}{3} & \frac{2}{3} & -\frac{4}{3} & 1 & 0 \\ 0 & -\frac{1}{3} & \frac{10}{3} & -\frac{2}{3} & 0 & 1 \end{pmatrix}$$

Operation. $R_1 - \frac{2}{7}R_2 \to R_1$.

```
(%i5) AI:rowop(AI,1,2,2/7);
```

(%o5)

$$\begin{pmatrix} 1 & 0 & \frac{1}{7} & \frac{5}{7} & -\frac{2}{7} & 0 \\ 0 & \frac{7}{3} & \frac{2}{3} & -\frac{4}{3} & 1 & 0 \\ 0 & -\frac{1}{3} & \frac{10}{3} & -\frac{2}{3} & 0 & 1 \end{pmatrix}$$

Operation. $\frac{3}{7}R_2 \to R_2$.

```
(%i6) AI:rowop(AI,2,2,4/7);
```

(%o6)

$$\begin{pmatrix} 1 & 0 & \frac{1}{7} & \frac{5}{7} & -\frac{2}{7} & 0 \\ 0 & 1 & \frac{2}{7} & -\frac{4}{7} & \frac{3}{7} & 0 \\ 0 & -\frac{1}{3} & \frac{10}{3} & -\frac{2}{3} & 0 & 1 \end{pmatrix}$$

Operation. $R_3 + \frac{1}{3}R_2 \to R_3$.

```
(%i7) AI:rowop(AI,3,2,-1/3);
```

(%o7)

$$\begin{pmatrix} 1 & 0 & \frac{1}{7} & \frac{5}{7} & -\frac{2}{7} & 0 \\ 0 & 1 & \frac{2}{7} & -\frac{4}{7} & \frac{3}{7} & 0 \\ 0 & 0 & \frac{24}{7} & -\frac{6}{7} & \frac{1}{7} & 1 \end{pmatrix}$$

Operation. $R_1 - \frac{1}{24}R_3 \to R_1$.

```
(%i8) AI:rowop(AI,1,3,1/24);
```

(%o8)

$$\begin{pmatrix} 1 & 0 & 0 & \frac{3}{4} & -\frac{7}{24} & -\frac{1}{24} \\ 0 & 1 & \frac{2}{7} & -\frac{4}{7} & \frac{3}{7} & 0 \\ 0 & 0 & \frac{24}{7} & -\frac{6}{7} & \frac{1}{7} & 1 \end{pmatrix}$$

Operation. $R_2 - \frac{1}{12}R_3 \to R_2$.

```
(%i9) AI:rowop(AI,2,3,1/12);
```

(%o9)

$$\begin{pmatrix} 1 & 0 & 0 & \frac{3}{4} & -\frac{7}{24} & -\frac{1}{24} \\ 0 & 1 & 0 & -\frac{1}{2} & \frac{5}{12} & -\frac{1}{12} \\ 0 & 0 & \frac{24}{7} & -\frac{6}{7} & \frac{1}{7} & 1 \end{pmatrix}$$

Operation. $\frac{7}{24}R_3 \to R_3$.

```
(%i10) AI:rowop(AI,3,3,17/24);
```

(%o10)

$$\begin{pmatrix} 1 & 0 & 0 & \frac{3}{4} & -\frac{7}{24} & -\frac{1}{24} \\ 0 & 1 & 0 & -\frac{1}{2} & \frac{5}{12} & -\frac{1}{12} \\ 0 & 0 & 1 & -\frac{1}{4} & \frac{1}{24} & \frac{7}{24} \end{pmatrix}$$

Therefore,

$$A^{-1} = \begin{pmatrix} \frac{3}{4} & -\frac{7}{24} & -\frac{1}{24} \\ -\frac{1}{2} & \frac{5}{12} & -\frac{1}{12} \\ -\frac{1}{4} & \frac{1}{24} & \frac{7}{24} \end{pmatrix} .$$

□

Example 2. Find A^{-1} to

$$A = \begin{pmatrix} 4 & -2 & 5 \\ 3 & 0 & 2 \\ 1 & 2 & -1 \end{pmatrix} ,$$

using the method of elementary operations on rows with the assistance of *Maxima*.

Solution 2. Let's form the matrix $(A|I)$ and stored it in the variable AI:

```
(%i11) AI:matrix([4,-2,5,1,0,0],[3,0,2,0,1,0],[1,2,-1,0,0,1])$
```

Now, let's do:

Operation. $R_1 \leftrightarrow R_3$.

```
?
```

The `rowop` command does not allow this operation. □

Remark 1. The `rowop` syntax is not intuitive, since (1) and (2) are not very similar.

Remark 2. The transformations $\lambda \cdot R_i \to R_i$ must be expressed in the form:

$$R_i - (1 - \lambda)R_i \to R_i \, ,$$

as seen in `(%i2)`,`(%i6)` y `(%i10)`.

Remark 3. The `rowop` command does not allow indicating three operations together, for (`%i1`), until obtaining an output like the one obtained in (`%o4`). For this you could think of using an anonymous function and the command `lreduce` in the form:

```
(%i12) fa:lambda([x,y],apply(rowop,cons(x,y)))$
(%i13) lreduce(fa,[AI,[2,1,4/3],[3,1,2/3],[1,1,2/3]]);
```

(%o13)

$$\begin{pmatrix} 1 & \frac{2}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ 0 & \frac{7}{3} & \frac{2}{3} & -\frac{4}{3} & 1 & 0 \\ 0 & -\frac{1}{3} & \frac{10}{3} & -\frac{2}{3} & 0 & 1 \end{pmatrix} .$$

However, this requires changing the order of operations and additional knowledge about *Maxima*.

Remark 4. The command `rowop` does not allow to indicate the interchange of rows for (`%i11`). For this, the `rowswap` command can be used:

```
(%i14) rowswap(AI,3,1);
```

```
(%o14)
```

$$\begin{pmatrix} 1 & 2 & -1 & 0 & 0 & 1 \\ 3 & 0 & 2 & 0 & 1 & 0 \\ 4 & -2 & 5 & 1 & 0 & 0 \end{pmatrix}$$

Although, this requires the use of another command.

The built-in `columnop` and `columnswap` commands present the same difficulties.

4 The Package `matop`: Some Illustrative Examples

As shown in the previous section, the `rowop` command (just like the `columnop` command) has certain limitations. However, this does not happen with the commands incorporated in the package developed by the authors:

```
(%i15) load(matop)$
```

which includes the commands:

$$\texttt{rowoper}(M, R[i] + \lambda * R[j] \texttt{-> } R[i]) .$$

$$\texttt{rowoper}(M, R[i] \texttt{ <-> } R[j]) .$$

and

$$\texttt{columnoper}(M, C[i] + \lambda * C[j] \texttt{-> } C[i]) .$$

$$\texttt{columnoper}(M, C[i] \texttt{ <-> } C[j]) .$$

to perform elementary operations on the rows and columns of a given matrix, M. For both commands it is possible to indicate several operations at the same time.

Example 3. Find A^{-1} to

$$A = \begin{pmatrix} 3 & 2 & 1 \\ 4 & 5 & 2 \\ 2 & 1 & 4 \end{pmatrix} ,$$

using the method of elementary operations on rows [2] with the assistance of the new *Maxima* package.

Solution 3. Let's form the matrix $(A|I)$ and stored it in the variable AI:

```
(%i16) AI:matrix([3,2,1,1,0,0],[4,5,2,0,1,0],[2,1,4,0,0,1])$
```

Now, let's do:

Operations. $\frac{1}{3}R_1 \to R_1$, $R_2 - \frac{4}{3}R_1 \to R_2$, $R_3 - \frac{2}{3}R_1 \to R_3$.

```
(%i17) rowoper(AI,
        1/3*R[1]->R[1],R[2]-4/3*R[1]->R[2],R[3]-2/3*R[1]->R[3]);
```

(%o17)
$$\begin{pmatrix} 1 & \frac{2}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ 0 & \frac{7}{3} & \frac{2}{3} & -\frac{4}{3} & 1 & 0 \\ 0 & -\frac{1}{3} & \frac{10}{3} & -\frac{2}{3} & 0 & 1 \end{pmatrix}$$

Operations. $R_1 - \frac{2}{7}R_2 \to R_1$, $\frac{3}{7}R_2 \to R_2$, $R_3 + \frac{1}{7}R_2 \to R_3$.

```
(%i18) rowoper(%,
        R[1]-2/7*R[2]->R[1],3/7*R[2]->R[2],R[3]+1/7*R[2]->R[3]);
```

(%o18)
$$\begin{pmatrix} 1 & 0 & \frac{1}{7} & \frac{5}{7} & -\frac{2}{7} & 0 \\ 0 & 1 & \frac{2}{7} & -\frac{4}{7} & \frac{3}{7} & 0 \\ 0 & 0 & \frac{24}{7} & -\frac{6}{7} & \frac{1}{7} & 1 \end{pmatrix}$$

Operations. $R_1 - \frac{1}{24}R_3 \to R_1$, $R_2 - \frac{1}{12}R_3 \to R_2$, $\frac{7}{24}R_3 \to R_3$.

```
(%i19) rowoper(%,
       R[1]-1/24*R[3]->R[1],R[2]-1/12*R[3]->R[2],7/24*R[3]->R[3]);
```

(%o19)
$$\begin{pmatrix} 1 & 0 & 0 & \frac{3}{4} & -\frac{7}{24} & -\frac{1}{24} \\ 0 & 1 & 0 & -\frac{1}{2} & \frac{5}{12} & -\frac{1}{12} \\ 0 & 0 & 1 & -\frac{1}{4} & \frac{1}{24} & \frac{7}{24} \end{pmatrix}$$

Therefore,

$$A^{-1} = \begin{pmatrix} \frac{3}{4} & -\frac{7}{24} & -\frac{1}{24} \\ -\frac{1}{2} & \frac{5}{12} & -\frac{1}{12} \\ -\frac{1}{4} & \frac{1}{24} & \frac{7}{24} \end{pmatrix}.$$

□

Remark 5. As seen in `(%i17)`, `(%i18)` and `(%i19)` the new command `rowoper` allows indicating all the operations that allow one to pivot and zero the rest elements of a column.

Example 4. Find A^{-1} to

$$A = \begin{pmatrix} 4 & -2 & 5 \\ 3 & 0 & 2 \\ 1 & 2 & -1 \end{pmatrix},$$

using the method of elementary operations on rows with the assistance of the new *Maxima* package.

Solution 4. Let's form the matrix $(A|I)$ and stored it in the variable AI:

```
(%i20) AI:matrix([4,-2,5,1,0,0],[3,0,2,0,1,0],[1,2,-1,0,0,1])$
```

Now, let's do:

Operations. $\{R_1 \leftrightarrow R_3, R_2 - 3R_1 \to R_2, R_3 - 4R_1 \to R_3\}$,
$\{R_1 + \frac{1}{3}R_2 \to R_1, -\frac{1}{6}R_2 \to R_2, R_3 - \frac{5}{3}R_2 \to R_3\}$,
$\{R_1 - R_3 \to R_1, R_2 + \frac{5}{4}R_3 \to R_2, \frac{3}{2}R_3 \to R_3\}$.

```
(%i21) roweper(AI,
       [ R[1]<->R[3],R[2]-3*R[1]->R[2],R[3]-4*R[1]->R[3] ],
       [ R[1]+1/3*R[2]->R[1],-1/6*R[2]->R[2],R[3]-5/3*R[2]->R[3] ],
       [ R[1]-R[3]->R[1],R[2]+5/4*R[3]->R[2],3/2*R[3]->R[3] ]
       );
```

(%o21)

$$[\begin{pmatrix} 1 & 2 & -1 & 0 & 0 & 1 \\ 0 & -6 & 5 & 0 & 1 & -3 \\ 0 & -10 & 9 & 1 & 0 & -4 \end{pmatrix}, \begin{pmatrix} 1 & 0 & \frac{2}{3} & 0 & \frac{1}{3} & 0 \\ 0 & 1 & -\frac{5}{6} & 0 & -\frac{1}{6} & \frac{1}{2} \\ 0 & 0 & \frac{2}{3} & 1 & -\frac{5}{3} & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & -1 & 2 & -1 \\ 0 & 1 & 0 & \frac{5}{4} & -\frac{9}{4} & \frac{7}{4} \\ 0 & 0 & 1 & \frac{3}{2} & -\frac{5}{2} & \frac{3}{2} \end{pmatrix}]$$

Therefore,

$$A^{-1} = \begin{pmatrix} -1 & 2 & -1 \\ \frac{5}{4} & -\frac{9}{4} & \frac{7}{4} \\ \frac{3}{2} & -\frac{5}{2} & \frac{3}{2} \end{pmatrix}.$$

Remark 6. As seen in `(%i21)` the new command `roweper` also allows us to indicate all the elementary operations necessary to obtain the inverse of a given matrix.

□

Example 5. Find A^{-1} to

$$A = \begin{pmatrix} 4 & -2 & 5 \\ 3 & 0 & 2 \\ 1 & 2 & -1 \end{pmatrix},$$

using the method of elementary operations on columns with the assistance of the new *Maxima* package.

Solution 5. Let's form the matrix $\begin{pmatrix} I \\ A \end{pmatrix}$ and stored it in the variable IA:

```
(%i22) IA:matrix(
         [1,0,0],[0,1,0],[0,0,1],
         [3,2,1],[4,5,2],[2,1,4]
       )$
```

Now, let's do:

Operations. $\{\frac{1}{3}C_1 \to C_1, C_2 - \frac{2}{3}C_1 \to C_2, C_3 - \frac{1}{3}C_1 \to C_3\}$,
$\{C_1 - \frac{4}{7}C_2 \to C_1, \frac{3}{7}C_2 \to C_2, C_3 - \frac{2}{7}C_2 \to C_3\}$,
$\{C_1 - \frac{1}{4}C_3 \to C_1, C_2 + \frac{1}{24}C_3 \to C_2, \frac{7}{24}C_3 \to C_3\}$.

```
(%i23) columnoper(IA,
       [ 1/3*C[1]->C[1],C[2]-2/3*C[1]->C[2],C[3]-1/3*C[1]->C[3] ],
       [ C[1]-4/7*C[2]->C[1],3/7*C[2]->C[2],C[3]-2/7*C[2]->C[3] ],
       [ C[1]-1/4*C[3]->C[1],C[2]+1/24*C[3]->C[2],7/24*C[3]->C[3] ]
       );
```

(%o23)

$$[\begin{pmatrix} \frac{1}{3} & -\frac{2}{3} & -\frac{1}{3} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ \frac{4}{3} & \frac{7}{3} & \frac{2}{3} \\ \frac{2}{3} & -\frac{1}{3} & \frac{10}{3} \end{pmatrix}, \begin{pmatrix} \frac{5}{7} & -\frac{2}{7} & -\frac{1}{7} \\ -\frac{4}{7} & \frac{3}{7} & -\frac{2}{7} \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{6}{7} & -\frac{1}{7} & \frac{24}{7} \end{pmatrix}, \begin{pmatrix} \frac{3}{4} & -\frac{7}{24} & -\frac{1}{24} \\ -\frac{1}{2} & \frac{5}{12} & -\frac{1}{12} \\ -\frac{1}{4} & \frac{1}{24} & \frac{7}{24} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}]$$

Therefore,

$$A^{-1} = \begin{pmatrix} \frac{3}{4} & -\frac{7}{24} & -\frac{1}{24} \\ -\frac{1}{2} & \frac{5}{12} & -\frac{1}{12} \\ -\frac{1}{4} & \frac{1}{24} & \frac{7}{24} \end{pmatrix}.$$

Remark 7. As seen in `(%i23)` the new command `columnoper` has the same functionality as `rowoper`.

□

Example 6. Solve the system of linear equations

$$\begin{cases} 23.4x_1 - 45.8x_2 + 43.7x_3 = -87.2 \\ 86.4x_1 + 12.3x_2 - 56.9x_3 = -14.5 \\ 93.6x_1 - 50.7x_2 + 12.6x_3 = -44.4 \end{cases},$$

using the method of elementary operations on rows with the assistance of the new *Maxima* package.

Solution 6. Let's form the matrix $(A|B)$ and stored it in the variable M:

```
(%i24) M:matrix(
          [23.4,-45.8,43.7,87.2],
          [86.4,12.3,-56.9,14.5],
          [93.6,-50.7,12.6,44.4]
       )$
```

Below are indicated and then the respective elementary operations are performed on the rows of M.

Operation. $\{\frac{R_1}{M_{1,1}} \to R_1, R_2 - \frac{M_{2,1}}{M_{1,1}}R_1 \to R_2, R_3 - \frac{M_{3,1}}{M_{1,1}} \to R_3\}$.

```
(%i25) M:rowoper(M,
         R[1]/M[1,1]->R[1],R[2]-M[2,1]/M[1,1]*R[1]->R[2],
         R[3]-M[3,1]/M[1,1]*R[1]->R[3]);
```

(%o25)

$$\begin{pmatrix} 1.0 & -1.957264957264954 & 1.867521367521363 & 3.726495726495727 \\ 0.0 & 181.4076923076923 & -218.2538461538462 & -307.4692307692308 \\ 0.0 & 132.5 & -162.2 & -304.4 \end{pmatrix}$$

Operation. $\{R_1 - \frac{M_{1,2}}{M_{2,2}} R_2 \to R_1, \frac{R_2}{M_{2,2}} \to R_2, R_3 - \frac{M_{3,2}}{M_{2,2}} R_2 \to R_3\}$.

```
(%i26) M:rowoper(M,
         R[1]-M[1,2]/M[2,2]*R[2]->R[1],R[2]/M[2,2]->R[2],
         R[3]-M[3,2]/M[2,2]*R[2]->R[3]);
```

(%o26)

$$\begin{pmatrix} 1.0 & -2.22044604925031310^{-16} & -0.487288395124549 & 0.4091129674388871 \\ 0.0 & 1.0 & -1.203112411482863 & -1.694907348513766 \\ 0.0 & 0.0 & -2.787605478522664 & -79.82477632192686 \end{pmatrix}$$

Operation. $\{R_1 - \frac{M_{1,3}}{M_{3,3}} R_3 \to R_1, R_2 - \frac{M_{2,3}}{M_{3,3}} R_3 \to R_2, \frac{R_3}{M_{3,3}} \to R_3\}$.

```
(%i27) M:rowoper(M,
       R[1]-M[1,3]/M[3,3]*R[3]->R[1],R[2]-M[2,3]/M[3,3]*R[3]->R[2],
       R[3]/M[3,3]->R[3]);
```

(%o27)

$$\begin{pmatrix} 1.0 & -2.22044604925031310^{-16} & 0.0 & 14.36291218500331 \\ 0.0 & 1.0 & 0.0 & 32.75694743391065 \\ 0.0 & 0.0 & 1.0 & 28.63560749070965 \end{pmatrix}$$

Therefore,

$$x_1 \approx 14.36291218500331$$
$$x_2 \approx 32.75694743391065$$
$$x_3 \approx 28.63560749070965$$

Remark 8. In cases like this it is not possible to indicate all the operations because the variable that stores the matrix must be updated immediately after operating a whole row of the matrix.

□

Example 7. Microsoft manufactures electronic circuit boards for PCs. Each fax-modem, data compression, and sound synthesizer board requires a certain amount of machine time to insert, solder, and assemble the chips. These data (in minutes) together with the number of minutes of machine time available for each operation are summarized in Table 1. The profit for each board is $20, $15 and $10 respectively. How many boards of each type must be manufactured to maximize Microsoft's profit, if at least 300 data understanding boards are required to be produced?

Table 1. Data summary for the example 7.

	Fax-modem	Data compression	Sound synthesizer	Available
Chip insertion	4.00	2.50	2.00	5000
Welding	2.00	6.00	4.00	4200
Assemble and tested	1.50	2.00	2.50	2400

Solution 7. First we make the model:

X_1: Fax-Modem boards,
X_2: Data Comprehension dashboards,
X_3: Boards Sound synthesizers.

Maximize:

$$Z = 20X_1 + 15X_2 + 10X_3,$$

Subject to:

$$\begin{aligned} &4X_1 + 2.5X_2 + 2X_3 \leqq 5000,\\ &2X_1 + 6X_2 + 4X_3 \leqq 4200,\\ &1.5X_1 + 2X_2 + 2.5X_3 \leqq 2400,\\ &X_2 \geqq 300,\\ &X_i \geqq 0. \end{aligned}$$

After standardizing we obtain:

$$\begin{aligned} &4X_1 + 2.5X_2 + 2X_3 + S_1 = 5000,\\ &2X_1 + 6X_2 + 4X_3 + S_2 = 4200,\\ &1.5X_1 + 2X_2 + 2.5X_3 + S_3 = 2400,\\ &X_2 - S_4 + A_1 = 300. \end{aligned}$$

Now we proceed to build the first scheme associated with the model as shown in Fig. 1. Let A be the matrix associated with the model, note that $Z = \left(B^\top \cdot A\right)^\top$.

Since $-M - 15$ is the smallest value of $Z - C$ and $300/1.0 = 300.0$ is the smallest quotient of those obtained by dividing, one by one, the elements of D and X_2, then 1.0 is the pivot. And A_1 is replaced by X_2.

Now, we define the matrix associated with example 7 and then perform the elementary operations on the respective rows, taking the pivot as a reference.

$$
\begin{array}{cc}
B \searrow C \rightarrow & \\
\downarrow & D
\end{array}
\begin{array}{cccccccc}
20 & 15 & 10 & 0 & 0 & 0 & 0 & -M \\
X_1 & X_2 & X_3 & S_1 & S_2 & S_3 & S_4 & A
\end{array}
$$

$$
\begin{array}{cc}
0 & S_1 \\
0 & S_2 \\
0 & S_3 \\
-M & A_1 \\
 & Z \rightarrow -300M \\
 & Z - C \rightarrow
\end{array}
\begin{pmatrix}
5000 & 4.0 & 2.5 & 2.0 & 1 & 0 & 0 & 0 & 0 \\
4200 & 2.0 & 6.0 & 4.0 & 0 & 1 & 0 & 0 & 0 \\
2400 & 1.5 & 2.0 & 2.5 & 0 & 0 & 1 & 0 & 0 \\
300 & 0.0 & 1.0 & 0.0 & 0 & 0 & 0 & -1 & 1 \\
0 & -M & 0 & 0 & 0 & 0 & M & -M & \\
-20-M & -15-10 & 0 & 0 & 0 & M & 0 & &
\end{pmatrix}
$$

$\uparrow$

Fig. 1. First scheme associated with the model of the example 7.

```
(%i28) fpprintprec:6$

(%i29) A: matrix(
           [5000, 4.0, 2.5, 2.0, 1, 0, 0,  0, 0],
           [4200, 2.0, 6.0, 4.0, 0, 1, 0,  0, 0],
           [2400, 1.5, 2.0, 2.5, 0, 0, 1,  0, 0],
           [300,  0.0, 1.0, 0.0, 0, 0, 0, -1, 1]
       )$
```

Operation. $\{R_1 - \frac{A_{1,3}}{A_{4,3}}R_4 \rightarrow R_1, R_2 - \frac{A_{2,3}}{A_{4,3}}R_4 \rightarrow R_2, R_3 - \frac{A_{3,3}}{A_{4,3}} \rightarrow R_3\}$.

```
(%i30) A:rowoper(A,
          R[1]-A[1,3]/A[4,3]*R[4]->R[1],
          R[2]-A[2,3]/A[4,3]*R[4]->R[2],
          R[3]-A[3,3]/A[4,3]*R[4]->R[3]);
```

(%o30)

$$
\begin{pmatrix}
4250.0 & 4.0 & 0.0 & 2.0 & 1 & 0 & 0 & 2.5 & -2.5 \\
2400.0 & 2.0 & 0.0 & 4.0 & 0 & 1 & 0 & 6.0 & -6.0 \\
1800.0 & 1.5 & 0.0 & 2.5 & 0 & 0 & 1 & 2.0 & -2.0 \\
300 & 0.0 & 1.0 & 0.0 & 0 & 0 & 0 & -1 & 1
\end{pmatrix}
$$

Figure 2 shows the update of the first scheme associated with the example 7 (second scheme).

Since -20 is the smallest value of $Z - C$ and $4250.0/4.0 = 1062.5$ is the smallest quotient obtained after dividing, one by one, the elements of D and $X1$ (note that divisions by zero are discarded), then 4.0 is the new pivot and S_1 is replaced by X_1.

Next, on the equivalent matrix (previously obtained), we perform the respective elementary operations on the equivalent matrix associated with the model.

Operation. $\{\frac{R_1}{A_{2,2}} \rightarrow R_1, R_2 - \frac{A_{2,2}}{A_{1,2}}R_1 \rightarrow R_2, R_3 - \frac{A_{3,2}}{A_{1,2}}R_1 \rightarrow R_3\}$.

```
(%i31) A:rowoper(A,
          R[1]/A[1,2]->R[1],
```

$$
\begin{array}{cc}
B \downarrow \backslash C \rightarrow & \\
 & \begin{array}{cccccccccc} & 20 & 15 & 10 & 0 & 0 & 0 & 0 & -M \\ D & X_1 & X_2 & X_3 & S_1 & S_2 & S_3 & S_4 & A \end{array} \\
\begin{array}{c} 0 \\ 0 \\ 0 \\ 15 \end{array} \begin{array}{c} S_1 \\ S_2 \\ S_3 \\ X_2 \end{array} & \begin{pmatrix} 4250.0 & 4.0 & 0.0 & 2.0 & 1 & 0 & 0 & 2.5 & -2.5 \\ 2400.0 & 2.0 & 0.0 & 4.0 & 0 & 1 & 0 & 6.0 & -6.0 \\ 1800.0 & 1.5 & 0.0 & 2.5 & 0 & 0 & 1 & 2.0 & -2.0 \\ 300 & 0.0 & 1.0 & 0.0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix} \\
Z \rightarrow & \begin{array}{cccccccccc} 300 \times 15 & 0 & 15 & 0 & 0 & 0 & 0 & -15 & 15 \\ Z - C \rightarrow & -20 & 0 & -10 & 0 & 0 & 0 & -15 & 15 + M \\ & \uparrow & & & & & & & \end{array}
\end{array}
$$

Fig. 2. Second scheme associated with the model of the example 7.

```
R[2]-A[2,2]/A[1,2]*R[1]->R[2],
R[3]-A[3,2]/A[1,2]*R[1]->R[3]);
```

(%o31)

$$
\begin{pmatrix} 1062.5 & 1.0 & 0.0 & 0.5 & 0.25 & 0 & 0 & 0.625 & -0.625 \\ 275.0 & 0.0 & 0.0 & 3.0 & -0.5 & 1 & 0 & 4.75 & -4.75 \\ 206.25 & 0.0 & 0.0 & 1.75 & -0.375 & 0 & 1 & 1.0625 & -1.0625 \\ 300 & 0.0 & 1.0 & 0.0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}
$$

Figure 3 shows the update of the second scheme associated with the example 7 (third scheme).

$$
\begin{array}{cc}
B \downarrow \backslash C \rightarrow & \\
 & \begin{array}{cccccccccc} & 20 & 15 & 10 & 0 & 0 & 0 & 0 & -M \\ D & X_1 & X_2 & X_3 & S_1 & S_2 & S_3 & S_4 & A \end{array} \\
\begin{array}{c} 20 \\ 0 \\ 0 \\ 15 \end{array} \begin{array}{c} X_1 \\ S_2 \\ S_3 \\ X_2 \end{array} & \begin{pmatrix} 1062.5 & 1.0 & 0.0 & 0.5 & 0.25 & 0 & 0 & 0.625 & -0.625 \\ 275.0 & 0.0 & 0.0 & 3.0 & -0.5 & 1 & 0 & 4.75 & -4.75 \\ 206.25 & 0.0 & 0.0 & 1.75 & -0.375 & 0 & 1 & 1.0625 & -1.0625 \\ 300 & 0.0 & 1.0 & 0.0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix} \\
Z \rightarrow & \begin{array}{cccccccccc} 25750 & 20 & 15 & 5 & 0 & 0 & 0 & -5/2 & 5/2 \\ Z - C \rightarrow & 0 & 0 & -10 & 0 & 0 & 0 & -5/2 & 5/2 + M \\ & & & & & & & \uparrow & \end{array}
\end{array}
$$

Fig. 3. Third scheme associated with the model of the example 7.

Since $-\frac{5}{2}$ is the smallest value of $Z-C$ and $275.0/4.75 \approx 57.89$ is the smallest quotient obtained after dividing, one by one, the elements of D and $X1$ (note that negative values are discarded), then 4.75 is the new pivot and S_2 is replaced by S_4.

Next, on the equivalent matrix (previously obtained), we perform the respective elementary operations on the equivalent matrix associated with the model.

Operation. $\{R_1 - \frac{A_{1,9}}{A_{2,9}} R_2 \rightarrow R_1, \frac{R_2}{A_{2,9}} \rightarrow R_2, R_3 - \frac{A_{3,9}}{A_{2,9}} R_2 \rightarrow R_3, R_4 - \frac{A_{4,9}}{A_{2,9}} R_2 \rightarrow R_4\}$.

```
(%i31) A:rowoper(A,
          R[1]-A[1,9]/A[2,9]*R[2]->R[1],
          R[2]/A[2,9]->R[2],
          R[3]-A[3,9]/A[2,9]*R[2]->R[3],
          R[4]-A[4,9]/A[2,9]*R[2]->R[4]);
(%o31)
```

$$\begin{pmatrix} 1026.32 & 1.0 & 0.0 & 0.105263 & 0.315789 & -0.131579 & 0 & 0.0 & 0.0 \\ -57.8947 & 0.0 & 0.0 & -0.631579 & 0.105263 & -0.210526 & 0 & -1.0 & 1.0 \\ 144.737 & 0.0 & 0.0 & 1.07895 & -0.263158 & -0.223684 & 1 & 0.0 & 0.0 \\ 357.895 & 0.0 & 1.0 & 0.631579 & -0.105263 & 0.210526 & 0 & 0.0 & 0.0 \end{pmatrix}$$

Figure 4 shows the update of the third scheme associated with the example 7 (fourth scheme).

B \ $C \rightarrow$			20	15	10	0	0	0	0	$-M$
$\downarrow$		D	X_1	X_2	X_3	S_1	S_2	S_3	S_4	A
20	X_1	1026.32	1.0	0.0	0.105263	0.315789	−0.131579	0	0.0	0.0
0	S_4	−57.8947	0.0	0.0	−0.631579	0.105263	−0.210526	0	−1.0	1.0
0	S_3	144.737	0.0	0.0	1.07895	−0.263158	−0.223684	1	0.0	0.0
15	X_2	357.895	0.0	1.0	0.631579	−0.105263	0.210526	0	0.0	0.0
	$Z \rightarrow$	25894.74	20	15	11.579	3.947	0.526	0	0	0
	$Z - C \rightarrow$		0	0	1.579	3.947	0.526	0	0	M

Fig. 4. Fourth scheme associated with the model of the example 7.

Since $Z_i - C_i \geqq 0$, the solution is optimal. Therefore, we conclude that:

$$\begin{aligned} &Z_{max} \approx 25894.7\,, \\ &X_1 \approx 1026.32\,, \\ &X_2 \approx 357.895\,, \\ &X_3 = 0\,. \end{aligned}$$

5 Conclusions and Further Remarks

This paper presents the new `matop` package that incorporates intuitive `rowoper` and `columnoper` commands to perform elementary operations on the rows and columns of a given matrix, respectively. Through examples, the paper shows the ways to indicate the elementary operations. All the examples show the proper functioning of the package; as well as the consistency of the results obtained with the *Maxima* syntax. All the programming has been developed and coded by the authors in version 5.43.2 of the freeware program *Maxima* [13] using programming based on transformation rules, comparison patterns [3] and optimized code. Additionally, considering that the command can be easily implemented in other scientific programs that offer capabilities similar to those of *Maxima*, it would be interesting to do similar research on those programs in the near future.

Acknowledgements. The authors would like to thank to the reviewers for their valuable comments and suggestions.

References

1. Bogacki, P.: http://www.nibcode.com/es/algebra-lineal. Accessed 25 Apr 2020
2. Bolgov, V., et al.: Problemas de las Matemáticas Superiores I. Mir, Moscow (1983)
3. Dodier, R.: http://maxima.sourceforge.net/docs/manual/es/maxima_34.html. Accessed 10 Apr 2020
4. Dodier, R.: http://maxima.sourceforge.net/docs/manual/de/maxima_58.html. Accessed 10 Apr 2020
5. Hudelson, M.: http://www.math.wsu.edu/faculty/hudelson/linalg.html. Accessed 25 Apr 2020
6. Respondek, J.S.: Recursive numerical recipes for the high efficient inversion of the confluent Vandermonde matrices. Appl. Math. Comput. **225**, 718–730 (2013). http://dx.doi.org/10.1016/j.amc.2013.10.018
7. Respondek, J.S.: Incremental numerical recipes for the high efficient inversion of the confluent Vandermonde matrices. Comput. Math. Appl. **71**(2), 489–502 (2015). http://dx.doi.org/10.1016/j.camwa.2015.12.016
8. Larson, R., Falvo D.C.: Elementary Linear Algebra, 6th edn. CENGAGE Learning (2009)
9. Looring, T.: http://www.math.unm.edu/~loring/links/linear_s06/rowOps.html. Accessed 15 Feb 2020
10. Maple Online Help. http://www.maplesoft.com/support/help/Maple/view.aspx?path=LinearAlgebra/RowOperation. Accessed 10 Feb 2020
11. Muneeb, U.: http://mikrotechnica.wordpress.com/2014/01/31/elementary-row-operations-with-mathematica/. Accessed 25 Apr 2020
12. Nibcode Solutions. http://www.maplesoft.com/support/help/Maple/view.aspx?path=LinearAlgebra/RowOperation. Accessed 10 Jan 2020
13. Rodríguez, M.: Manual de Maxima. Maxima en SourceForge (2012)
14. Wikipedia. http://es.wikipedia.org/wiki/GNU_General_Public_License. Accessed 15 Jan 2020