

Maple Primer 1: Matrices in Maple

Any time you are doing linear algebra work in Maple be sure to start your document with the command `with(LinearAlgebra)` :

which loads many more additional Linear Algebra commands into Maple. Note the colon after the command and note the uppercase letters in the command. If you do not use the colon, you get a list of all the new commands that are available, see below. It is a distracting waste of space.

`with(LinearAlgebra)`

[&x, *Add, Adjoint, BackwardSubstitute, BandMatrix, Basis, BezoutMatrix, BidiagonalForm, BilinearForm, CharacteristicMatrix, CharacteristicPolynomial, Column, ColumnDimension, ColumnOperation, ColumnSpace, CompanionMatrix, ConditionNumber, ConstantMatrix, ConstantVector, Copy, CreatePermutation, CrossProduct, DeleteColumn, DeleteRow, Determinant, Diagonal, DiagonalMatrix, Dimension, Dimensions, DotProduct, EigenConditionNumbers, Eigenvalues, Eigenvectors, Equal, ForwardSubstitute, FrobeniusForm, GaussianElimination, GenerateEquations, GenerateMatrix, Generic, GetResultDataType, GetResultShape, GivensRotationMatrix, GramSchmidt, HankelMatrix, HermiteForm, HermitianTranspose, HessenbergForm, HilbertMatrix, HouseholderMatrix, IdentityMatrix, IntersectionBasis, IsDefinite, IsOrthogonal, IsSimilar, IsUnitary, JordanBlockMatrix, JordanForm, KroneckerProduct, LA_Main, LUDecomposition, LeastSquares, LinearSolve, LyapunovSolve, Map, Map2, MatrixAdd, MatrixExponential, MatrixFunction, MatrixInverse, MatrixMatrixMultiply, MatrixNorm, MatrixPower, MatrixScalarMultiply, MatrixVectorMultiply, MinimalPolynomial, Minor, Modular, Multiply, NoUserValue, Norm, Normalize, NullSpace, OuterProductMatrix, Permanent, Pivot, PopovForm, QRDecomposition, RandomMatrix, RandomVector, Rank, RationalCanonicalForm, ReducedRowEchelonForm, Row, RowDimension, RowOperation, RowSpace, ScalarMatrix, ScalarMultiply, ScalarVector, SchurForm, SingularValues, SmithForm, StronglyConnectedBlocks, SubMatrix, SubVector, SumBasis, SylvesterMatrix, SylvesterSolve, ToeplitzMatrix, Trace, Transpose, TridiagonalForm, UnitVector, VandermondeMatrix, VectorAdd, VectorAngle, VectorMatrixMultiply, VectorNorm, VectorScalarMultiply, ZeroMatrix, ZeroVector, Zip]* **(1)**

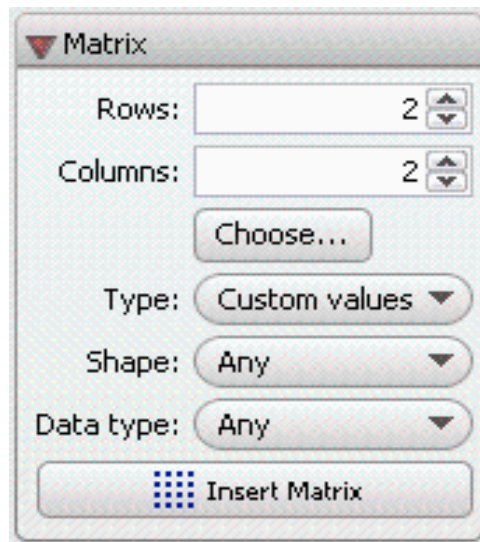
NOTE 1: Maple commands are displayed typically in *math italic font*. Notice the two buttons at the top of the Maple window. When you are doing mathematics (i.e., entering Maple commands) make sure that the 'Math' button is highlighted. The commands should appear in italic font. When typing explanatory text, make sure the 'Text' button is highlighted.

NOTE 2: To execute all of the commands in this file, go to the toolbar at the top of the page and click on **!!!** button. You will see all the commands execute consecutively. However, each command should execute after you type it and hit 'Enter' or 'Return'.

Look at **Example 1** on page 5 in your text. Suppose that we want to enter the augmented matrix for the system

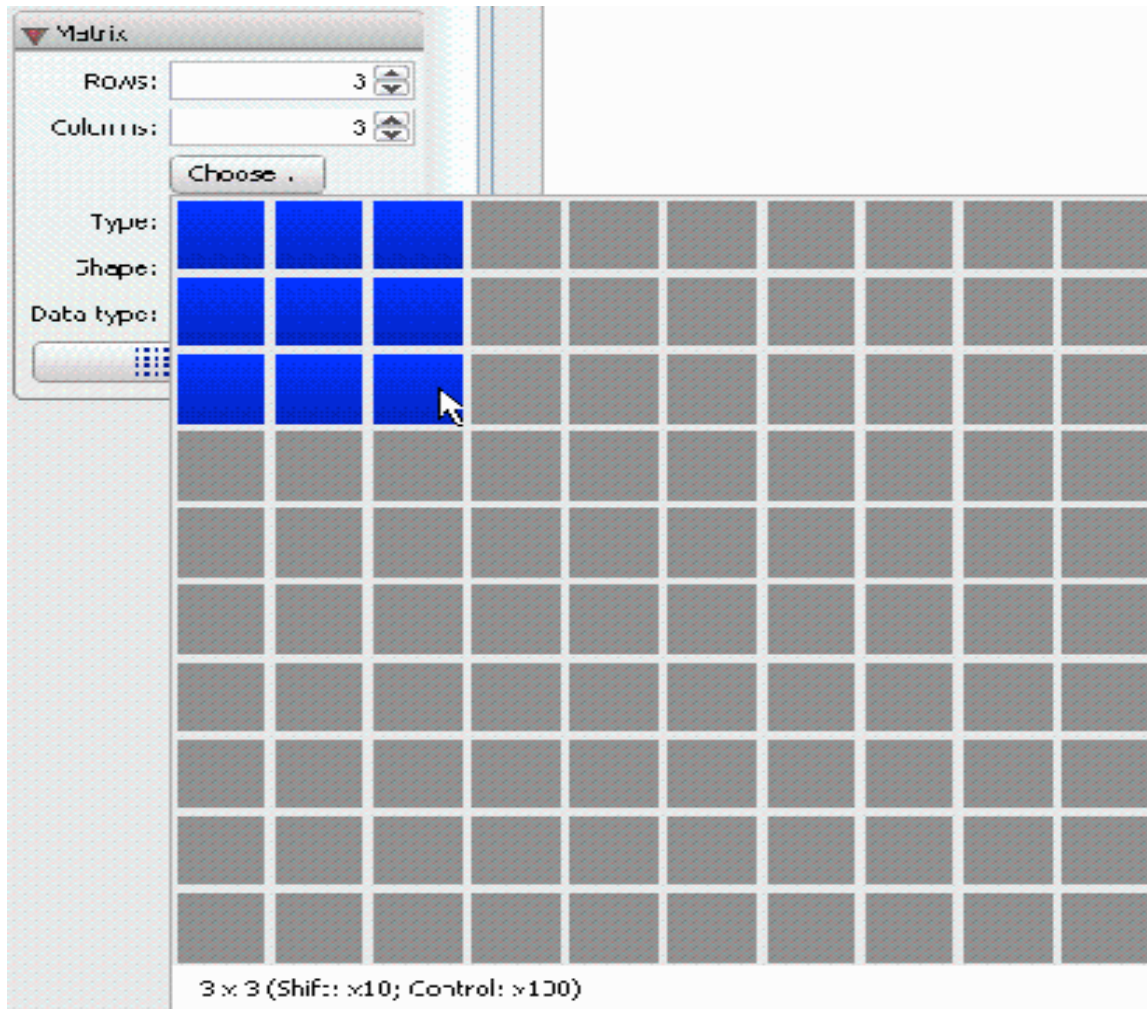
$$\begin{array}{rcl} x_1 + 2x_2 + x_3 & = & 0 \\ 2x_2 - 8x_3 & = & 8 \\ -4x_1 + 2x_2 + 9x_3 & = & -9 \end{array}$$

There are several ways to do this. The simplest is to use the **Matrix Palette** in the left column of the Maple window to interactively create a matrix:



If the palette is not open, click on the 'Matrix' button on the left. Once the Matrix palette is open, you can specify the matrix size (see below) and properties. First specify the number of rows and columns or use the 'Choose' button to specify the shape (in the figure below, a 3 x 3 matrix has been specified).

To insert a matrix, click the 'Insert Matrix' button.



In our case we need a 3 x 4 matrix. After inserting the matrix:

1. Enter the values of the entries. To move to the next entry placeholder, press **Tab**.
2. After specifying all entries, press **Enter (Return)**.

That's what I have done below. After hitting return, the computer echoes what you have typed.

$$\begin{bmatrix} 1 & -2 & 1 & 0 \\ 0 & 2 & -8 & 8 \\ -4 & 5 & 9 & -9 \end{bmatrix}$$

$$\begin{bmatrix} 1 & -2 & 1 & 0 \\ 0 & 2 & -8 & 8 \\ -4 & 5 & 9 & -9 \end{bmatrix} \quad (2)$$

It is often useful to give the matrix a name so that we can refer to it in subsequent calculations. Traditionally mathematicians use uppercase letters to denote matrices. Let's use A. We use the symbol := to indicate that we are defining the matrix A to be the matrix above. There's no need to retype the matrix, you can cut and paste the matrix from where you first entered it. Better yet, simply insert A:= in front of what you entered above and hit enter again.

$$A := \begin{bmatrix} 1 & -2 & 1 & 0 \\ 0 & 2 & -8 & 8 \\ -4 & 5 & 9 & -2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & -2 & 1 & 0 \\ 0 & 2 & -8 & 8 \\ -4 & 5 & 9 & -2 \end{bmatrix} \quad (3)$$

Notice that A is the augmented matrix in Example 1 on page 5. Now we are ready to do some row reduction using Maple commands.

Performing Elementary Row Operations on a Matrix

Here's the command that we will be using: **RowOperation(A, row(s), s)**

The parameters or inputs into this command determine what row operation is performed. Specifically, the **parameters** or **inputs** are

- A** which is the matrix to be changed
- row(s)** a list of one or two row numbers, either **i** or **[i, j]**, depending on whether one or two rows are used in the operation.
- s** an optional argument representing the multiplier of the row (if any)

Description

- The specific operation that is performed is determined by the **parameters** that are used in the calling sequence.

REPLACEMENT: ADDING A MULTIPLE OF ONE ROW TO ANOTHER:

RowOperation(A, [i,j], s) produces a matrix which has the same entries as A except that **row i** is replaced by **row i + s*row j**. This operation is used to add (subtract) a multiple of one row to another. **NOTE** the order of i and j in the brackets!

We will do the row reduction just as in **Example 1** on pages 5 and 6 in the text. In the example below, B is the matrix formed by adding to row 3 of A the multiple 4 times row 1 of A.

$$B := \text{RowOperation}(A, [3, 1], 4)$$

$$\begin{bmatrix} 1 & -2 & 1 & 0 \\ 0 & 2 & -8 & 8 \\ 0 & -3 & 13 & -2 \end{bmatrix} \quad (1.1)$$

Notice that the result matches the text on page 5. The next step is to do scaling in row 2.

SCALING: MULTIPLYING A ROW BY A CONSTANT

RowOperation(A, k, s) produces a matrix which has the same entries as A except that the **k**th row is multiplied by **s**. Of course we want to work with matrix B above. In the example below, C is the matrix where row 2 of B is multiplied by 1/2.

$$C := \text{RowOperation}\left(B, 2, \frac{1}{2}\right)$$
$$\begin{bmatrix} 1 & -2 & 1 & 0 \\ 0 & 1 & -4 & 4 \\ 0 & -3 & 13 & -2 \end{bmatrix} \quad (1.2)$$

This is the matrix in the text. Before we continue with the row reduction in the text, let's introduce the last row operation:

INTERCHANGE: SWITCHING TWO ROWS

RowOperation(A, [i, j]) produces a matrix which has the same entries as A except that rows **i** and **j** are interchanged. Note the square brackets around **i** and **j**.

In the example below rows 2 and 3 of C are switched.

$$X := \text{RowOperation}(C, [2, 3])$$
$$\begin{bmatrix} 1 & -2 & 1 & 0 \\ 0 & -3 & 13 & -2 \\ 0 & 1 & -4 & 4 \end{bmatrix} \quad (4)$$

Of course, we did not need to switch the rows to do the reduction, so let's switch them back. Notice the same command can be used to do this. We will redefine C as X with rows 2 and 3 switched.

$$C := \text{RowOperation}(X, [2, 3])$$
$$\begin{bmatrix} 1 & -2 & 1 & 0 \\ 0 & 1 & -4 & 4 \\ 0 & -3 & 13 & -2 \end{bmatrix} \quad (5)$$

Summary: The effect of the row operation command is determined by its input parameters.

RowOperation(A, k, s) produces a matrix which has the same entries as A except that the **k**th row is multiplied by **s**.

RowOperation(A, [i,j], s) produces a matrix which has the same entries as A except that **row i** is replaced by **row i + s*row j**.

RowOperation(A, [i, j]) produces a matrix which switches rows **i** and **j**.

PROBLEM: Finish reducing the matrix A to reduced row echelon form using the three operations.

Using other commands to define matrices and vectors

To define a matrix using the **Matrix** command:

Specify all element values as a list of lists that define the element values row-wise. Lists are enclosed in square brackets and commas are used to separate elements of the list. For example, to create a 2 x 3 matrix with 1 2 3 in the first row and 4 5 6 in the second row, we will need a list of two lists --- the two lists are the first row [1,2,3] and the second row [4,5,6] which are then put in a list, i.e. inside square brackets.

$$K := \text{Matrix}([[1, 2, 3], [4, 5, 6]])$$
$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \quad (6)$$

Column vectors (usually denoted with lower case letters) are easier, simply use angle brackets and commas:

$$z := \langle 7, 8 \rangle$$

$$\begin{bmatrix} 7 \\ 8 \end{bmatrix} \tag{7}$$

Building Augmented Matrices from Coefficient Matrices and Column Vectors

We can build an augmented matrix from the coefficient matrix A of a system and the column vector b of constants in the system:

First create the matrix A using the matrix palette (or use the method above). Let's say it is 3 x 3:

$$A := \begin{bmatrix} 27 & 99 & 92 \\ 8 & 29 & -31 \\ 69 & 44 & 67 \end{bmatrix}$$

$$\begin{bmatrix} 27 & 99 & 92 \\ 8 & 29 & -31 \\ 69 & 44 & 67 \end{bmatrix} \tag{8}$$

Now create the column vector b (a 3 x 1 matrix) as usual (or use the method above):

$$b := \begin{bmatrix} -32 \\ -74 \\ -4 \end{bmatrix}$$

$$\begin{bmatrix} -32 \\ -74 \\ -4 \end{bmatrix} \tag{9}$$

To create the augmented matrix, use pointed brackets and a vertical bar (vertical bars indicate new columns):

$$Aug := \langle A|b \rangle$$

$$\begin{bmatrix} 27 & 99 & 92 & -32 \\ 8 & 29 & -31 & -74 \\ 69 & 44 & 67 & -4 \end{bmatrix} \tag{10}$$

Why bother with this method? Because if we now want to use the same coefficient matrix but a different set of constants in the column, we do not need to retype the entire matrix. Suppose we want to solve

$$C := \left\langle A \left| \begin{bmatrix} -76 \\ -72 \\ -2 \end{bmatrix} \right. \right\rangle$$

$$\begin{bmatrix} 27 & 99 & 92 & -76 \\ 8 & 29 & -31 & -72 \\ 69 & 44 & 67 & -2 \end{bmatrix} \tag{11}$$

Notice we did not first need to define the new column vector, we can just slap A together with a column vector of our choice.