

Use of *Excel* to Solve Simultaneous Algebraic Equations

A previous white paper showed how Cramer's rule provides a methodology to solve systems of simultaneous algebraic equations. This is fairly convenient when dealing with systems of 2 or 3 equations and doing the calculations by hand. If one wanted to use Excel this is not necessarily the most convenient way to deal with the equations (since separate matrices would have to be set up for each matrix to represent the numerator in Cramer's Rule). Instead it would be more convenient to use *Excel's* intrinsic formulas for matrix operations.

Matrix Formulation for Systems of N Equations

Let's assume we have a system of N simultaneous algebraic equations with N unknowns x_1, x_2, \dots, x_N where the equations are in the form:

$$a_{1,1}x_1 + a_{1,2}x_2 + \dots + a_{1,N}x_N = b_1$$

$$a_{2,1}x_1 + a_{2,2}x_2 + \dots + a_{2,N}x_N = b_2$$

and so forth to:

$$a_{N,1}x_1 + a_{N,2}x_2 + \dots + a_{N,N}x_N = b_N$$

This system of equations can be put into matrix form as:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,N} \\ a_{2,1} & a_{2,2} & \dots & a_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N,1} & a_{N,2} & \dots & a_{N,N} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix}$$

In matrix form the equation can be represented as:

$$\mathbf{Ax} = \mathbf{b}$$

and the representation of the solution is almost trivial:

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}.$$

Note that this is valid only if the equations are independent, so the inverse matrix \mathbf{A}^{-1} exists. One condition for the existence of the inverse matrix is that the determinant of the original matrix, $|\mathbf{A}|$, is non-zero.

Even though the representation of the solution is trivial the actual calculations that must be performed are not.

When one has a complex set of calculations to be performed it is natural to try to use *Excel*. It turns out that *Excel* has built in functions that can make the calculations easier:

- **MINVERSE** to determine the inverse of an NxN matrix.
- **MMULT** to multiply two matrices, the first have the same number of columns as the second has rows.
- **MDETERM** to calculate the determinant of an NxN matrix.

The procedure could be done as follows:

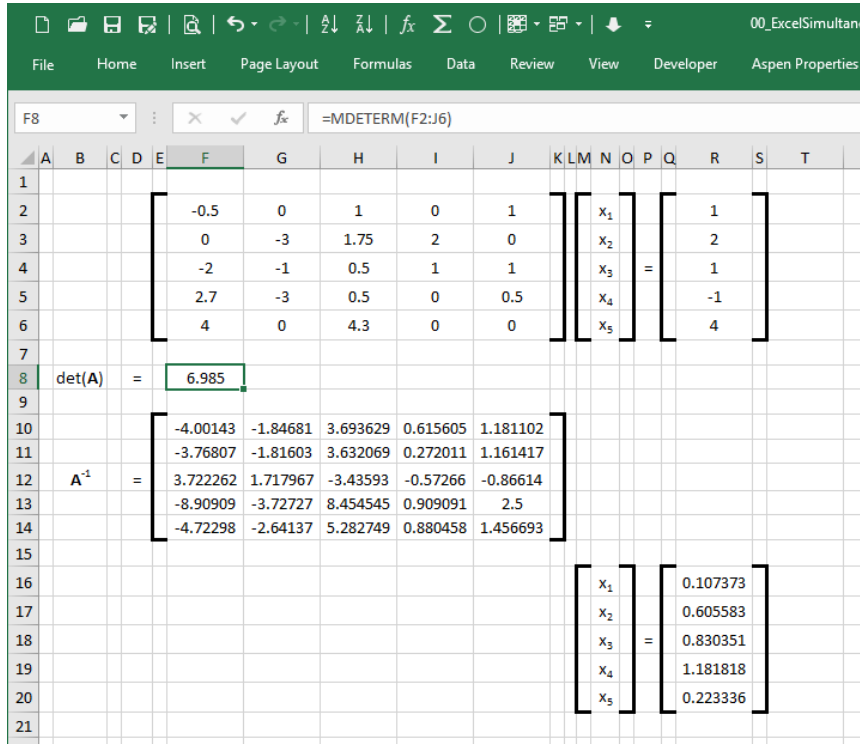
1. Set up the coefficient matrix **A** in a set of contiguous cells, N rows by N columns.
2. Set up the right-hand side vector **b** in a set of contiguous cells, N rows by 1 column.
3. You may want to use **MDETERM** to calculate the determinant of matrix **A** to ensure a solution to the equations should exist.
4. You may want to set aside a set of contiguous cells for \mathbf{A}^{-1} , N rows by N columns. You would use **MINVERSE** to do this.
5. Set aside a set of contiguous cells for the solution vector **x**, N rows by 1 column. You would use **MMULT** to do this.

Lets consider the set of equations:

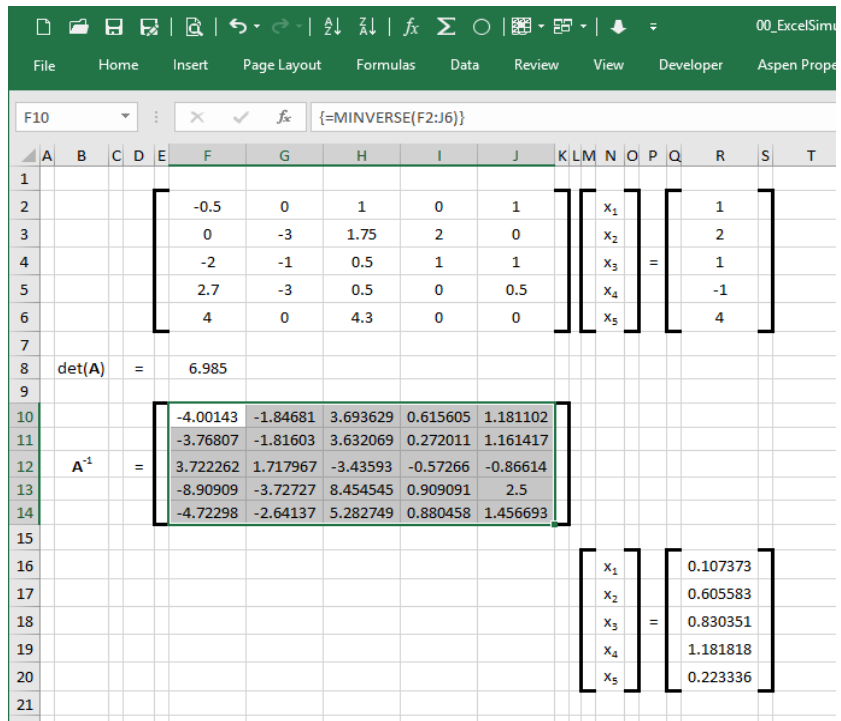
$$\begin{bmatrix} -0.5 & 0 & 1 & 0 & 1 \\ 0 & -3 & 1.75 & 2 & 0 \\ -2 & -1 & 0.5 & 1 & 1 \\ 2.7 & -3 & 0.5 & 0 & .5 \\ 4 & 0 & 4.3 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \\ -1 \\ 4 \end{bmatrix}$$

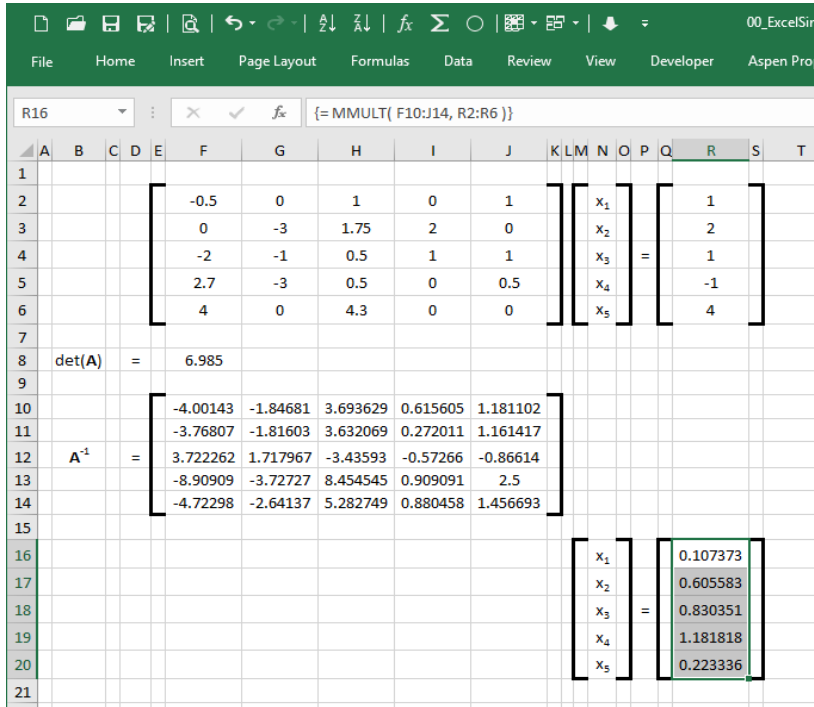
The following image shows a set up to use Excel to find the solution vector **x**. Notice the following about the setup:

- The values for the coefficient matrix are in cells F2:J6. The coefficients for the right-hand side vector are in cells R2:R6.
- There are borders used to imply matrices (such as E2:E6) but these are not necessary to perform the calculations.
- The calculation for the determinant is in F8 & shown in the formula bar.



The next image shows how MINVERSE can be called to determine the inverse matrix A^{-1} . Note that the cells F10:J14 are selected & all of the calculated values for the array are distributed. Notice the braces (“{” & “}”) around the formula in the formula bar? This denotes that this is an array formula & the individual results are to be distributed across the cells. You get this by entering the formula & pressing **Ctrl-Shift-Enter**, not just *Enter*.





Finally, we can see how the **MMULT** function is used for the final solution vector. Again, notice that this is an array function & you need to highlight cells for all values, enter the function, & then press *Ctrl-Shift-Enter* (not just *Enter*).

One final image shows a very compact way to determine the solution vector without showing the actual inverse matrix. Now the **MMULT** function has the **MINVERSE** function embedded in it.

