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 , ..., 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:

$\int a_{1,1}$	<i>a</i> _{1,2}		<i>a</i> _{1,N}	$\begin{bmatrix} x_1 \end{bmatrix}$		$\begin{bmatrix} b_1 \end{bmatrix}$	
a _{2,1}	<i>a</i> _{2,2}	•••	<i>a</i> _{2,N}	<i>x</i> ₂		b_2	
:	÷	••.	:	:	=	:	
$\lfloor a_{N,1} \rfloor$	$a_{\scriptscriptstyle N,2}$		$a_{N,N}$	$[x_N]$		b_N	

In matrix form the equation can be represented as:

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

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 A^{-1} exists. One condition for the existence of the inverse matrix is that the determinant of the original matrix, |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:

- **MINIVERSE** 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.
- You may want to set aside a set of contiguous cells for A⁻¹, N rows by N columns. You would use MINIVERSE to do this.
- 5. Set aside a set of contiguous cells for the solution vector \mathbf{x} , N rows by 1 column. You would use **MMULT** to do this.

Lets consider the set of equations:

-0.5	0	1	0	1]	$\begin{bmatrix} x_1 \end{bmatrix}$		$\begin{bmatrix} 1 \end{bmatrix}$
0	-3	1.75	2	0	<i>x</i> ₂		2
-2	-1	0.5	1	1	<i>x</i> ₃	=	1
2.7	-3	0.5	0	.5	<i>x</i> ₄		-1
4	0	4.3	0	0	x_{5}		4

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

- The values for the coefficient matrix are in cells F2:J6. The coefficients for the righthand 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.

C) 🖻	日	ķ	🖻 🕯	, ⊳.	ĝ↓ ⊼↓	$f_x \Sigma$	0 🎬 -	55	· -		L	÷			00_E	ExcelSi	nultane
Fil	le	Home	e	Insert	Page Layout	Formu	las Data	a Revier	N		Viev	v		Dev	/eloper	Asp	en Pro	perties
F8		Ŧ		× •	f _x	=MDETERN	VI(F2:J6)											
	АВ	С	D	E F	G	н	1	J	к	LΜ	N	0	Ρ	Q	R	S	т	
1														L				
2				-0.5	0	1	0	1			x ₁				1			
3				0	-3	1.75	2	0			x ₂				2			
4				-2	-1	0.5	1	1			x ₃		=	L	1			
5				2.7	-3	0.5	0	0.5			x ₄				-1			
6				4	0	4.3	0	0			X ₅				4			
7									П					F				
8	det(A)	=	6.985]													
9																		
10				-4.00143	-1.84681	3.693629	0.615605	1.181102										
11				-3.76807	-1.81603	3.632069	0.272011	1.161417										
12	A ⁻¹		=	3.722262	1.717967	-3.43593	-0.57266	-0.86614										
13	_			-8.90909	-3.72727	8.454545	0.909091	2.5										
14			_	-4.72298	-2.64137	5.282749	0.880458	1.456693	Ц					_				
15										-		-		┢				
16											x			L	0.107373			
17											x ₂				0.605583			
18											X3		=		0.830351			
19											x ₄				1.181818			
20											x ₅				0.223336			
21																		

The next image shows how MINVERSE can be called to determine the inverse matrix \mathbf{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 <u>Ctrl-Shift-Enter</u>, not just *Enter*.

	Ľ	-	H	Đ	2	<u>[</u> à, 5		ĝ↓ ⊼↓	$f_x \Sigma$) 🎬 - I	58		4	L				00_	ExcelSim
	ile		lon	ne		Insert	Page Layout	Formu	las Data	n Reviev	v		View)ev	eloper	As	oen Prop
F1	.0			-	:	× v	f _x {	=MINVERS	E(F2:J6)}										
	A	в	С	D	E	F	G	Н	1	J	к	LM	N	0	Ρ	Q	R	S	т
1					L						_	_		_				Ц	
2						-0.5	0	1	0	1			X1				1		
3						0	-3	1.75	2	0			x ₂				2		
4						-2	-1	0.5	1	1			x ₃		=		1		
5						2.7	-3	0.5	0	0.5			x ₄				-1		
6						4	0	4.3	0	0			X ₅				4		
7																		Π	
8		det(A)		=		6.985													
9											_								
10						-4.00143	-1.84681	3.693629	0.615605	1.181102									
11						-3.76807	-1.81603	3.632069	0.272011	1.161417		_							
12		A ⁻¹		=		3.722262	1.717967	-3.43593	-0.57266	-0.86614									
13						-8.90909	-3.72727	8.454545	0.909091	2.5		_							
14					L	-4.72298	-2.64137	5.282749	0.880458	1.456693		_							
15												÷	-	H				Ц	
16													X ₁				0.107373		
17													x ₂				0.605583		
18													X ₃		=		0.830351		
19													x ₄				1.181818		
20													х ₅				0.223336		
21																			

Ľ	נ 🛋	H	I F	8	<u>[</u>]()	• @ -	£↓ ∡↓	$f_x \Sigma$	0 🇱 -	57	•		L	Ŧ			00	_ExcelSi
Fi	ile	Ho	ne		Insert I	Page Layout	: Formu	las Data	a Review	w		View			Dev	eloper	As	pen Pro
R1	6		Ŧ		×v	f _x	{= MMULT(F10:J14, R	2:R6)}									
	А В	C	D	E	F	G	н	1	J	KI	LM	N	0	Ρ	Q	R	S	т
1																		
2					-0.5	0	1	0	1		Γ	x ₁				1		
3					0	-3	1.75	2	0			x ₂				2		
4					-2	-1	0.5	1	1			X3		=		1		
5					2.7	-3	0.5	0	0.5			X4				-1		
6					4	0	4.3	0	0			X5				4		
7		-		Н						۲	Ч	-	۲		Н		۲	
8	det(/	۹)	=		6.985													
9																		
10				Γ	-4.00143	-1.84681	3.693629	0.615605	1.181102									
11					-3.76807	-1.81603	3.632069	0.272011	1.161417									
12	A ⁻¹		=		3.722262	1.717967	-3.43593	-0.57266	-0.86614									
13					-8.90909	-3.72727	8.454545	0.909091	2.5									
14				L	-4.72298	-2.64137	5.282749	0.880458	1.456693									
15																		
16												x ₁				0.107373		
17												x ₂				0.605583		
18												x ₃		=		0.830351		
19												x4				1.181818		
20										Ħ		X5				0.223336		
21		-								Ħ	H	-	۲		H			

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.

[נ	6	88	<u>a</u> •	ר< <u>י</u> יל ייל	ĝ↓ ĭ↓	$f_x \Sigma$	0		58 -	- 🖡 =	
Fi	ile		Home	Insert	Page Layou	t Formi	ılas Dat	ta	Review	N	View	Developer
01	.0		• :	×	f _x	{= MMULT	(MINVERS	E(C	2:G6), O	2:06	;)}	
	Α	в	с	D	E	F	G	нт	JKL	MN	0	P Q
1									_			_
2			-0.5	0	1	0	1		x1		1	
3			0	-3	1.75	2	0		x ₂		2	
4			-2	-1	0.5	1	1		X3	=	1	
5			2.7	-3	0.5	0	0.5		×4		-1	
6			4	0	4.3	0	0		x ₅		4	
7								Π				
8									det(A)	=	6.985	
9												-
10									X ₁		0.107373	
11									x2		0.605583	
12									X3	=	0.830351	
13									x ₄		1.181818	
14									X5		0.223336	
15												1
16												