Solving Mass Balances using Matrix Algebra

Alex Doll, P.Eng, *Alex G Doll Consulting Ltd.* http://www.agdconsulting.ca

Abstract

Matrix Algebra, also known as linear algebra, is well suited to solving material balance problems encountered in plant design and optimisation. A properly constructed matrix is not sensitive the iterations of circular calculations that can cause 'hard wired' spreadsheet mass balances to fail to properly converge and balance. This paper demonstrates how to construct equations and use matrix algebra on a typical computer spreadsheet to solve an example mass balance during a mineral processing plant design.

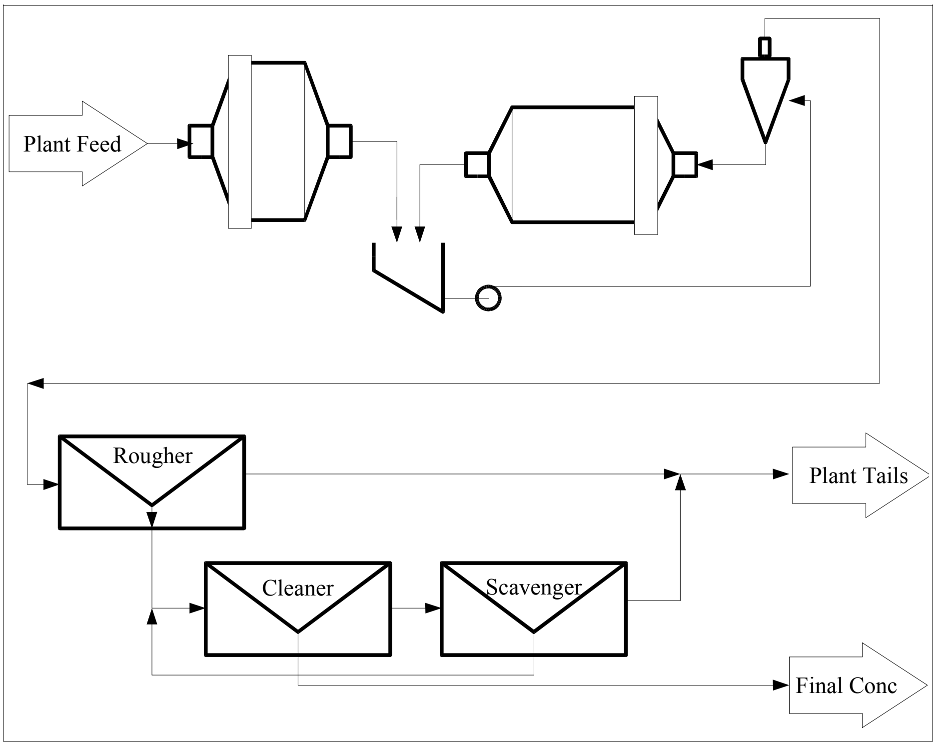
Introduction

Mass balance calculations describe an engineering problem where mass flows between unit operations and the composition of those flows are partly known and partly unknown. The purpose of the calculation is to mathematically analyse the known flows and compositions to solve for the unknown flows and compositions. Two main types of mass balances are commonly performed: design calculations and operating plant reconciliation.

The design calculation mass balance typically has few known values and many unknown values. These are typically encountered during plant design when the results of testwork and a flowsheet are the only known values. The purpose of a design mass balance is to calculate values for the unknown flows and compositions.

Operating plant reconciliation, by contrast, tends to have a large amount of data which may be contradictory. Data sources such as on-stream analysers and flowmeters produce large amounts of data, all of which are subject to random noise, calibration and sampling errors. The purpose of an operating plant reconciliation is to remove the random noise and errors to produce a single, consistent and reasonable snapshot of the state of an operating plant.

This paper will deal exclusively with the design calculation situation where there are few known values and several unknowns. No filtering and reconciliation will be performed on the known data. An example solving a copper concentrator flotation circuit is presented and the process flow diagram is given below.



The example will use the following design criteria from metallurgical testwork:

- Plant feed rate 10,000 tonnes/day of 0.5%Cu ore.
- Overall plant recovery of 90% by weight.
- Final concentrate grade of 27.5%Cu by weight.
- Rougher concentrate grade of 7%Cu by weight.

Method

The method consists of three major operations: creating a diagram of the flowsheet, deriving equations the describe the flowsheet, and using a standard personal computer spreadsheet to solve the equations.

Creating a Diagram

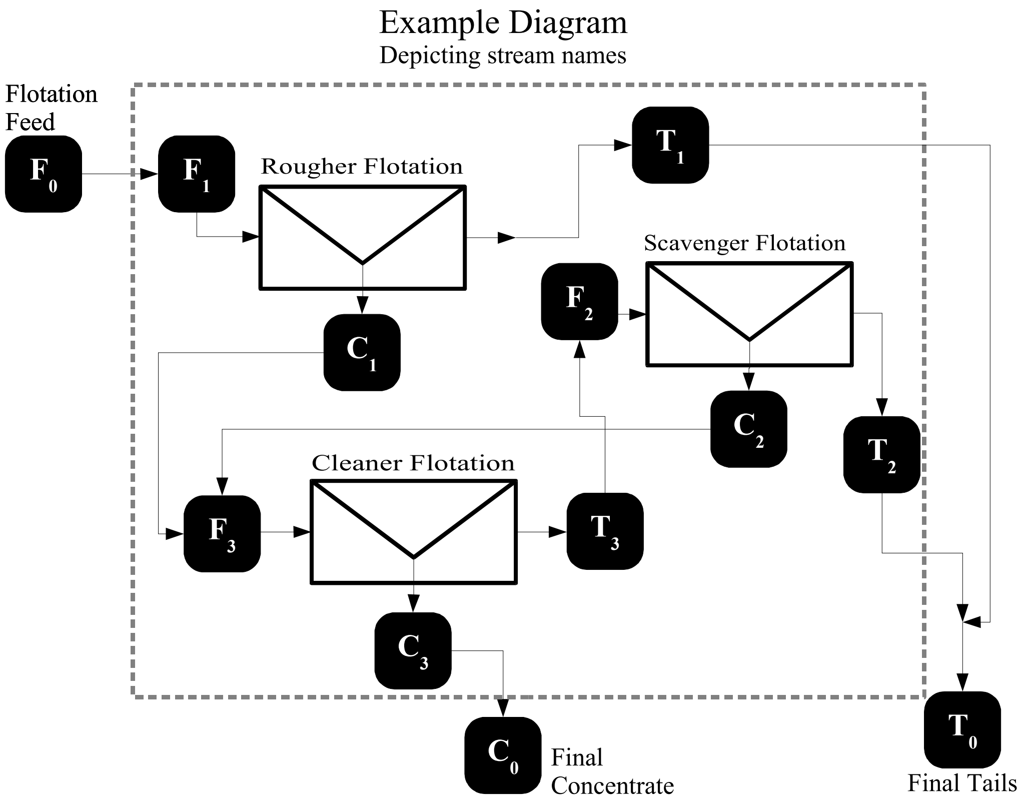
The first step in solving mass balance calculations is to create a diagram clearly depicting the positions where flows and analysis values will be calculated. Each flow position (stream) will be labelled with a unique identifier. The following notation is used:

- Fx denotes the Feed stream to unit operation 'x',
- Tx denotes the Tails stream from unit operation 'x', and
- Cx denotes the Concentrate stream from unit operation 'x'.

Where 'x' is one of:

0. Entire Flotation circuit,
1. Rougher Flotation,

2. Scavenger Flotation, or
3. Cleaner Flotation



Deriving Equations

All nodes (unit operations and connections between unit operations) can be described using formulae. This example consists of the following nodes:

Mass Balance Nodes, Unit Operations:
1. Rougher Flotation
2. Scavenger Flotation
3. Cleaner Flotation

Mass Balance Nodes, Connections:
4. Plant feed to Rougher Flotation
5. Rougher Tails to Final Tails
6. Rougher and Scavenger Concentrate to Cleaner Flotation
7. Cleaner Tails and Scavenger Tails to Final Tails
8. Cleaner Concentrate to Final Concentrate

The stream names indicated on the diagram represent the total mass flows (as t/h) of the respective streams. Expressing these unit operations connections as formulae, the first series of equations is derived governing the total mass flows around all nodes:

1. $F1 = T1 + C1$
2. $F2 = T2 + C2$
3. $F3 = T3 + C3$
4. $F0 = F1$
5. $T3 = F2$
6. $C1 + C2 = F3$
7. $T1 + T2 = T0$
8. $C3 = C0$

Now derive a second set of equations that expresses the copper balance around these same nodes. The copper flow will be expressed as ($A_Z \times Z$) where A_Z is the copper mass% in stream Z and Z is the total mass flow in stream Z:

9. $A_{F1} \times F1 = A_{T1} \times T1 + A_{C1} \times C1$
10. $A_{F2} \times F2 = A_{T2} \times T2 + A_{C2} \times C2$
11. $A_{F3} \times F3 = A_{T3} \times T3 + A_{C3} \times C3$
12. $A_{F0} \times F0 = A_{F1} \times F1$
13. $A_{T3} \times T3 = A_{F2} \times F2$
14. $A_{C1} \times C1 + A_{C2} \times C2 = A_{F3} \times F3$
15. $A_{T1} \times T1 + A_{T2} \times T2 = A_{T0} \times T0$
16. $A_{C3} \times C3 = A_{C0} \times C0$

Now identify the known values. In this example, the design criteria sets the plant feed rate $F0$ and grade A_{F0}, the final concentrate grade A_{C0}, the rougher concentrate grade A_{C1}, and the overall plant recovery. The overall recovery is actually a new formula that will be added to the equations. Thus A_{F0}, A_{C1}, A_{C0}, $F0$ are underlined.

Underline the known values in the equations and add the recovery formula:

1. $F1 = T1 + C1$
2. $F2 = T2 + C2$
3. $F3 = T3 + C3$
4. $\underline{F0} = F1$
5. $T3 = F2$
6. $C1 + C2 = F3$
7. $T1 + T2 = T0$
8. $C3 = C0$
9. $A_{F1} \times F1 = A_{T1} \times T1 + \underline{A_{C1}} \times C1$
10. $A_{F2} \times F2 = A_{T2} \times T2 + A_{C2} \times C2$
11. $A_{F3} \times F3 = A_{T3} \times T3 + A_{C3} \times C3$
12. $\underline{A_{F0}} \times \underline{F0} = A_{F1} \times F1$
13. $A_{T3} \times T3 = A_{F2} \times F2$
14. $\underline{A_{C1}} \times C1 + A_{C2} \times C2 = A_{F3} \times F3$
15. $A_{T1} \times T1 + A_{T2} \times T2 = A_{T0} \times T0$
16. $A_{C3} \times C3 = \underline{A_{C0}} \times C0$
17. $A_{T0} \times T0 = (\underline{1\text{-Recovery}}) \times \underline{A_{F0}} \times \underline{F0}$

Attempt to simplify the equations by eliminating some of the "x=y" combinations. In this example, F0 and F1 are identical and therefore one may substitute for and eliminate the other.

The same substitution may be done for C0 and C3, and for T3 and F2. Furthermore, several streams will have identical analyses, allowing the following substitutions:

1. $A_{F0} = A_{F1}$
2. $A_{T3} = A_{F2}$
3. $A_{C0} = A_{C3}$

After eliminating unknowns, the simplified equations are:

1. F1 = T1 + C1
2. F2 = T2 + C2
3. F3 = T3 + C3
4. (equation eliminated)
5. (equation eliminated)
6. C1 + C2 = F3
7. T1 + T2 = T0
8. (equation eliminated)
9. $A_{F1} \times F1 = A_{T1} \times T1 + A_{C1} \times C1$
10. $A_{T3} \times T3 = A_{T2} \times T2 + A_{C2} \times C2$
11. $A_{F3} \times F3 = A_{T3} \times T3 + A_{C3} \times C3$
12. (equation eliminated)
13. (equation eliminated)
14. $A_{C1} \times C1 + A_{C2} \times C2 = A_{F3} \times F3$
15. $A_{T1} \times T1 + A_{T2} \times T2 = A_{T0} \times T0$
16. (equation eliminated)
17. $A_{T0} \times T0 = (1\text{-Recovery}) \times A_{F1} \times F1$

Check that all instances of F0, C0, F2, A_{F0}, A_{F2}, and A_{C0} have been removed from these simplified equations.

Reorganise these equations such that only the completely known terms appear on the right of the equal sign, and any terms with unknown values appear on the left.

1. T1 + C1 = F1
2. T2 + C2 - T3 = 0
3. T3 + C3 - F3 = 0
4. C1 + C2 - F3 = 0
5. T1 + T2 - T0 = 0
6. $A_{T1} \times T1 + A_{C1} \times C1 = A_{F1} \times F1$
7. $A_{T2} \times T2 + A_{C2} \times C2 - A_{T3} \times T3 = 0$
8. $A_{T3} \times T3 + A_{C3} \times C3 - A_{F3} \times F3 = 0$
9. $A_{C1} \times C1 + A_{C2} \times C2 - A_{F3} \times F3 = 0$
10. $A_{T1} \times T1 + A_{T2} \times T2 - A_{T0} \times T0 = 0$
11. $A_{T0} \times T0 = (1\text{-Recovery}) \times A_{F1} \times F1$

Several of the terms in the reorganised equations are nonlinear – they contain two unknown values multiplied together. Such terms cannot be solved using matrix algebra and, therefore, these terms must be eliminated. $A_{T0}\times T0$, $A_{T1}\times T1$, $A_{T2}\times T2$, $A_{T3}\times T3$, $A_{F3}\times F3$ and $A_{C2}\times C2$ are all nonlinear terms, whereas $\underline{A_{C1}}\times C1$ is linear because one of the values is known (and underlined).

There are two ways to resolve nonlinear terms: formula substitution and term substitution.

Formula substitution may be performed when the nonlinear term appears in only two of the equations. Express the equations in terms of the nonlinear components, and then the two equations are merge eliminating the nonlinear term. For example:

6. $A_{T1}\times T1 + \underline{A_{C1}}\times C1 = \underline{A_{F1}}\times\underline{F1}$
10. $A_{T1}\times T1 + A_{T2}\times T2 - A_{T0}\times T0 = 0$

becomes

6. $A_{T1}\times T1 = \underline{A_{F1}}\times\underline{F1} - \underline{A_{C1}}\times C1$
10. $A_{T1}\times T1 = A_{T0}\times T0 - A_{T2}\times T2$

set the two equations equal to each other and eliminate the nonlinear term. In this example, $A_{T1}\times T1 = A_{T1}\times T1$:

6. $\underline{A_{F1}}\times\underline{F1} - \underline{A_{C1}}\times C1 = A_{T0}\times T0 - A_{T2}\times T2$
10. (equation eliminated)

Another formula substitution is combination of equations 6 and 11 yielding:

6. $\underline{A_{F1}}\times\underline{F1} - \underline{A_{C1}}\times C1 = (\underline{1\text{-Recovery}})\times\underline{A_{F1}}\times\underline{F1} - A_{T2}\times T2$
11. (equation eliminated)

Terms substitution may be done when one of the unknowns of a nonlinear term only appears in that nonlinear term. In this example the nonlinear term $A_{F3}\times F3$ contains an A_{F3} value that only appears in formulae multiplied by $F3$. $F3$, by contrast, appears in formulae without A_{F3}. Because A_{F3} only appears with $F3$, this $A_{F3}\times F3$ term is a suitable candidate for term substitution. In the language of mathematics, we substituting an arbitrary new degree of freedom for the single degree of freedom belonging to A_{F3}. After the substitution there must be no instances of A_{F3} left in any equations.

Create an arbitrary new unknown set equal to the nonlinear term. Substitute this new unknown into all instances of the nonlinear term in the equations. For example:

Create new $Y_{T2}=A_{T2}\times T2$, $Y_{T3}=A_{T3}\times T3$, $Y_{F3}=A_{F3}\times F3$ and a new $Y_{C2}=A_{C2}\times C2$. Substitute these into the equations (also include the formula substitution for equations 6. and 10.).

1. $T1 + C1 = \underline{F1}$
2. $T2 + C2 - T3 = 0$
3. $T3 + C3 - F3 = 0$
4. $C1 + C2 - F3 = 0$

5. $T1 + T2 - T0 = 0$
6. $Y_{T2} - \underline{A_{C1}} \times C1 = (\underline{1\text{-Recovery}}) \times \underline{A_{F1}} \times \underline{F1} - \underline{A_{F1}} \times \underline{F1}$
7. $Y_{T2} + Y_{C2} - Y_{T3} = 0$
8. $Y_{T3} + \underline{A_{C3}} \times C3 - Y_{F3} = 0$
9. $\underline{A_{C1}} \times C1 + Y_{C2} - Y_{F3} = 0$
10. (equation eliminated)
11. (equation eliminated)

Reorganising, equations are eliminated and the known and unknown components are moved to proper sides of the equal sign. This set of equations is suitable for solving because it does not contain any nonlinear terms.

1. $T1 + C1 = \underline{F1}$
2. $T2 + C2 - T3 = 0$
3. $T3 + C3 - F3 = 0$
4. $C1 + C2 - F3 = 0$
5. $T1 + T2 - T0 = 0$
6. $\underline{A_{C1}} \times C1 - Y_{T2} = (\underline{\text{Recovery}}) \times \underline{A_{F1}} \times \underline{F1}$
7. $Y_{T2} + Y_{C2} - Y_{T3} = 0$
8. $Y_{T3} + \underline{A_{C3}} \times C3 - Y_{F3} = 0$
9. $\underline{A_{C1}} \times C1 + Y_{C2} - Y_{F3} = 0$

This completes the first attempt to mathematically describe the flowsheet and mass balance. Several checks must be performed prior to continuing to the next step. First check that all the known values underlined. Next, have all the term substitutions eliminated one unknown term from all the equations? Finally, count the number of unknown values and the number of equations; there must be more equations than unknowns in order to proceed. In this example, there are 12 unknowns (T1, C1, T2, C2, T3, C3, F3, T0, Y_{T2}, Y_{C2}, Y_{T3} and Y_{F3}) and 9 equations so it is not safe to proceed to the spreadsheet and attempt to solve the mass balance.

In order to solve the example, either more equations must be added or some unknowns must be converted to known values. "Add constraints or remove degrees of freedom" in the language of matrix mathematics. Two new equations describing the overall mass balance and overall copper balance can be added without creating any more unknowns. A third equation can be added describing the concentrate mass flow in terms of the plant recovery. These new equations are:

10. $\underline{F1} = C3 + T0$
11. $\underline{A_{F1}} \times \underline{F1} = \underline{A_{C3}} \times C3 + (A_{T0} \times T0)$
12. $\underline{A_{C3}} \times C3 = (\underline{\text{Recovery}}) \times \underline{A_{F1}} \times \underline{F1}$

Equation 11 is not a suitable addition because it contains a term that was already eliminated earlier ($A_{T0} \times T0$). Reject this equation and instead, substitute a different equation that describes the relationship between Y_{T2} and the overall cleaner copper flows.

11. $\underline{A_{C1}} \times C1 - \underline{A_{C3}} \times C3 = Y_{T2}$

The example is still not suitable for solving because there are still too few equations, and no obvious new formulae are evident from the design criteria and the flowsheet. The solution will require assumptions of either additional criteria or values for one or more of the unknowns. Reviewing the flowsheet and criteria indicates that no data exists to describe the operation of the scavenger flotation cells, so this is a reasonable place to begin adding assumptions.

Assume the concentrate grade of the scavenger A_{C2} is 3%Cu (eliminating the need for Y_{C2}). Substitute all occurrences of Y_{C2} with $\underline{A_{C2}}\times C2$ and underline the $\underline{A_{C2}}$ term.

The relation between the two tailings streams T3 and T2 is not well described either, so assume a cleaner flotation recovery of 80%. This assumption adds a new equation:

13. $Y_{T3} = (\underline{1\text{-}ClnrRec})\times Y_{F3}$

The 13 reorganised equations are:

1. $T1 + C1 = \underline{F1}$
2. $T2 + C2 - T3 = 0$
3. $T3 + C3 - F3 = 0$
4. $C1 + C2 - F3 = 0$
5. $T1 + T2 - T0 = 0$
6. $\underline{A_{C1}}\times C1 - Y_{T2} = (\underline{Recovery})\times\underline{A_{F1}}\times\underline{F1}$
7. $Y_{T2} + \underline{A_{C2}}\times C2 - Y_{T3} = 0$
8. $Y_{T3} + \underline{A_{C3}}\times C3 - Y_{F3} = 0$
9. $\underline{A_{C1}}\times C1 + \underline{A_{C2}}\times C2 - Y_{F3} = 0$
10. $C3 + T0 = \underline{F1}$
11. $\underline{A_{C1}}\times C1 - \underline{A_{C3}}\times C3 - Y_{T2} = 0$
12. $\underline{A_{C3}}\times C3 = (\underline{Recovery})\times\underline{A_{F1}}\times\underline{F1}$
13. $(\underline{1\text{-}ClnrRec})\times Y_{F3} - Y_{T3} = 0$

The 11 remaining unknowns are: (T1, C1, T2, C2, T3, C3, F3, T0, Y_{T2}, Y_{T3}, and Y_{F3})

The quantity of equations now exceeds the number of unknowns, so it is safe to proceed to the next step and program a spreadsheet to solve the mass balance.

Programming the Spreadsheet

The mass balance will be solved using the matrix algebra formula for the regression equation: $\vec{w}=(A^T \cdot A)^{-1} \cdot A^T \cdot \vec{y}$ where $\vec{w}$ is the vector of unknowns, A is the matrix of equations and $\vec{y}$ is the vector of known values[1]. The determinant of the product of $(A^T \cdot A)$ will be checked prior to inversion to ensure that it is "significantly greater than zero". This avoids the spreadsheet performing an inversion on a matrix it thinks has a determinant of 10^{-12} due to floating-point errors.

The spreadsheet used in this example is part of the OpenOffice suite which is freely available for download from http://www.openoffice.org. OpenOffice performs matrix functions similar to Microsoft Excel and the example code should be identical. Lotus 1-2-3, MathCAD, Mathematica and other maths packages use different syntax and structures to solve matrix algebra, so the spreadsheet functions used in the example will not work without modification. The overall procedure given is still appropriate to these other packages.

Matrices in OpenOffice (and Excel) are subsets of an Array structure. Array formulae return results that span more than one cell, and it is the responsibility of the engineer programming the spreadsheet to know the correct dimensions of the matrix output. Refer to a matrix algebra textbook for the details of the dimensions to be expected from the output of matrix multiplication.

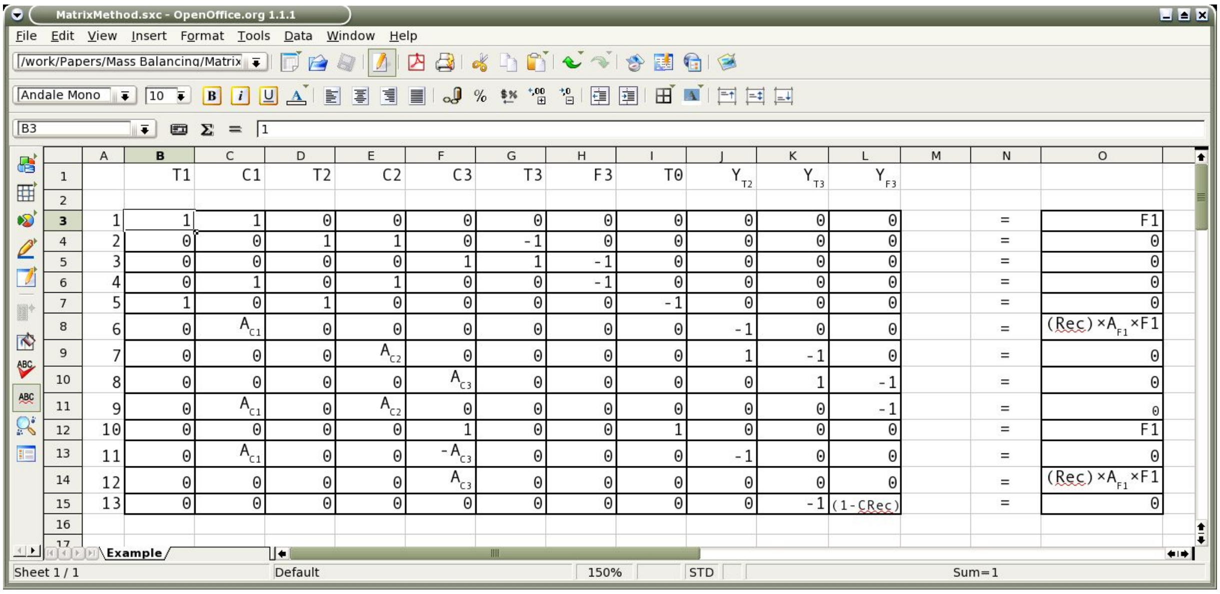
Array formulae differ from regular OpenOffice (and Excel) formulae by requiring the output range to be highlighted whilst typing the formula, and ending the formula with a "ctrl-shift-enter" keystroke rather than the usual "enter" keystroke. Refer to the spreadsheet documentation for details of "array functions".[2]

The first programming step is to create the matrix A and vector y. Expand the equations such that each equation includes all of the possible variables, with the exponent of a variable being zero if that variable is not used in an equation.

1. $1{\times}T1 + 1{\times}C1 + 0{\times}T2 + 0{\times}C2 + 0{\times}C3 + 0{\times}T3 + 0{\times}F3 + 0{\times}T0 + 0{\times}Y_{T2} + 0{\times}Y_{T3} + 0{\times}Y_{F3} = \underline{F1}$
2. $0{\times}T1 + 0{\times}C1 + 1{\times}T2 + 1{\times}C2 + 0{\times}C3 + -1{\times}T3 + 0{\times}F3 + 0{\times}T0 + 0{\times}Y_{T2} + 0{\times}Y_{T3} + 0{\times}Y_{F3} = 0$
3. $0{\times}T1 + 0{\times}C1 + 0{\times}T2 + 0{\times}C2 + 1{\times}C3 + 1{\times}T3 + -1{\times}F3 + 0{\times}T0 + 0{\times}Y_{T2} + 0{\times}Y_{T3} + 0{\times}Y_{F3} = 0$
4. $0{\times}T1 + 1{\times}C1 + 0{\times}T2 + 1{\times}C2 + 0{\times}C3 + 0{\times}T3 + -1{\times}F3 + 0{\times}T0 + 0{\times}Y_{T2} + 0{\times}Y_{T3} + 0{\times}Y_{F3} = 0$
5. $1{\times}T1 + 0{\times}C1 + 1{\times}T2 + 0{\times}C2 + 0{\times}C3 + 0{\times}T3 + 0{\times}F3 + -1{\times}T0 + 0{\times}Y_{T2} + 0{\times}Y_{T3} + 0{\times}Y_{F3} = 0$
6. $0{\times}T1 + \underline{A_{C1}}{\times}C1 + 0{\times}T2 + 0{\times}C2 + 0{\times}C3 + 0{\times}T3 + 0{\times}F3 + 0{\times}T0 + -1{\times}Y_{T2} + 0{\times}Y_{T3} + 0{\times}Y_{F3} = (\underline{Recovery}){\times}\underline{A_{F1}}{\times}\underline{F1}$
7. $0{\times}T1 + 0{\times}C1 + 0{\times}T2 + \underline{A_{C2}}{\times}C2 + 0{\times}C3 + 0{\times}T3 + 0{\times}F3 + 0{\times}T0 + 1{\times}Y_{T2} + -1{\times}Y_{T3} + 0{\times}Y_{F3} = 0$
8. $0{\times}T1 + 0{\times}C1 + 0{\times}T2 + 0{\times}C2 + \underline{A_{C3}}{\times}C3 + 0{\times}T3 + 0{\times}F3 + 0{\times}T0 + 0{\times}Y_{T2} + 1{\times}Y_{T3} + -1{\times}Y_{F3} = 0$
9. $0{\times}T1 + \underline{A_{C1}}{\times}C1 + 0{\times}T2 + \underline{A_{C2}}{\times}C2 + 0{\times}C3 + 0{\times}T3 + 0{\times}F3 + 0{\times}T0 + 0{\times}Y_{T2} + 0{\times}Y_{T3} + -1{\times}Y_{F3} = 0$
10. $0{\times}T1 + 0{\times}C1 + 0{\times}T2 + 0{\times}C2 + 1{\times}C3 + 0{\times}T3 + 0{\times}F3 + 1{\times}T0 + 0{\times}Y_{T2} + 0{\times}Y_{T3} + 0{\times}Y_{F3} = \underline{F1}$
11. $0{\times}T1 + \underline{A_{C1}}{\times}C1 + 0{\times}T2 + 0{\times}C2 - \underline{A_{C3}}{\times}C3 + 0{\times}T3 + 0{\times}F3 + 0{\times}T0 + -1{\times}Y_{T2} + 0{\times}Y_{T3} + 0{\times}Y_{F3} = 0$
12. $0{\times}T1 + 0{\times}C1 + 0{\times}T2 + 0{\times}C2 + \underline{A_{C3}}{\times}C3 + 0{\times}T3 + 0{\times}F3 + 0{\times}T0 + 0{\times}Y_{T2} + 0{\times}Y_{T3} + 0{\times}Y_{F3} = (\underline{Recovery}){\times}\underline{A_{F1}}{\times}\underline{F1}$
13. $0{\times}T1 + 0{\times}C1 + 0{\times}T2 + 0{\times}C2 + 0{\times}C3 + 0{\times}T3 + 0{\times}F3 + 0{\times}T0 + 0{\times}Y_{T2} + -1{\times}Y_{T3} + (\underline{1-ClnrRec}){\times}Y_{F3} = 0$

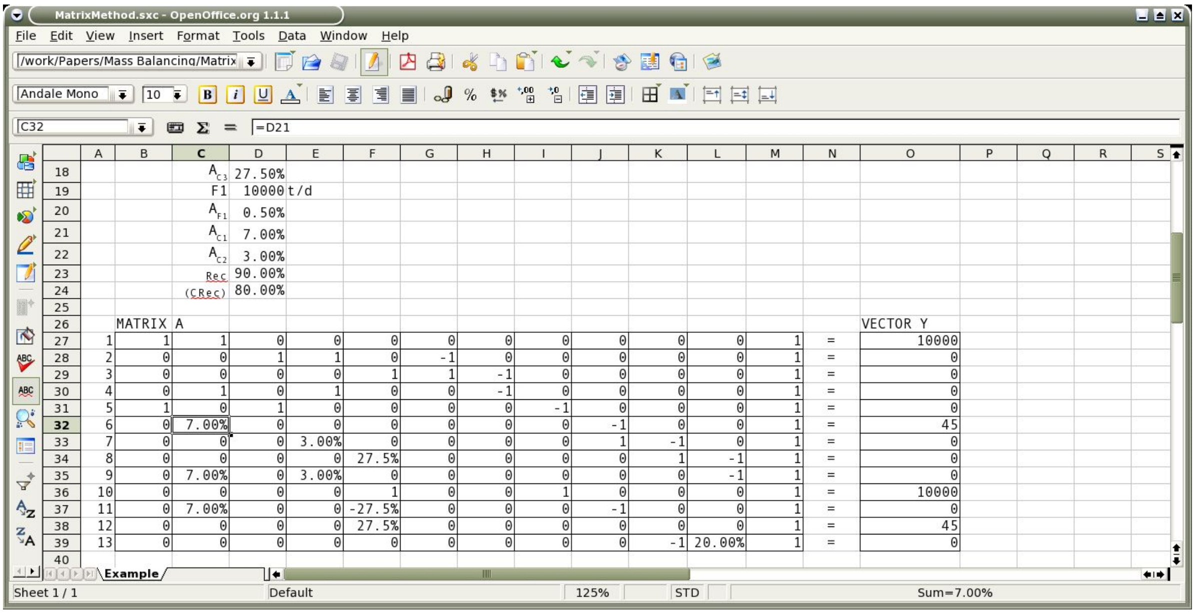
1 Derivation of regression equation: http://luna.cas.usf.edu/~mbrannic/files/regression/regma.htm
2 A good introduction on spreadsheet arrays available at http://www.cpearson.com/excel/array.htm

Equations are transferred to a matrix in the spreadsheet software. Label the top row of the matrix with the name of the unknown, label the leftmost column of the matrix with the equation number, and fill the matrix with the coefficients of the unknown values. Leave a couple of blank columns then enter the expressions from the right-hand side of the equal signs. Use the same constant notation as the derivation to avoid confusion.



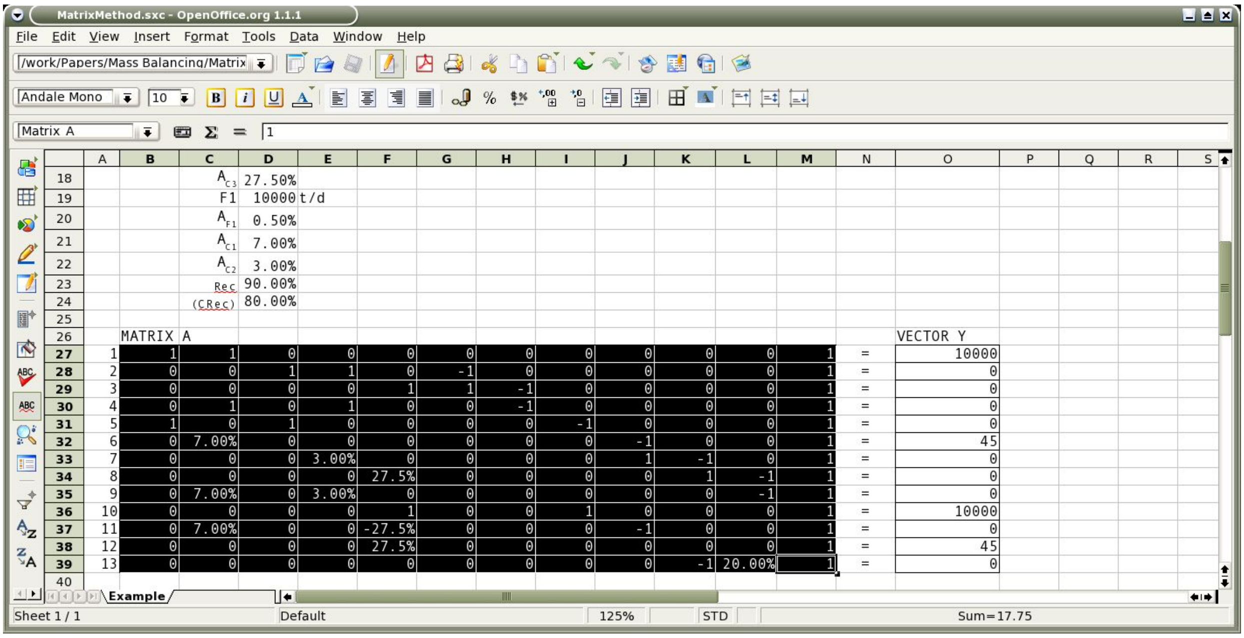
	T1	C1	T2	C2	C3	T3	F3	T0	Y_{T2}	Y_{T3}	Y_{F3}		
1	1	1	0	0	0	0	0	0	0	0	0	=	F1
2	0	0	1	1	0	-1	0	0	0	0	0	=	0
3	0	0	0	0	1	1	-1	0	0	0	0	=	0
4	0	1	0	1	0	0	-1	0	0	0	0	=	0
5	1	0	1	0	0	0	0	-1	0	0	0	=	0
6	0	A_{C1}	0	0	0	0	0	0	-1	0	0	=	$(Rec) \times A_{F1} \times F1$
7	0	0	0	A_{C2}	0	0	0	0	1	-1	0	=	0
8	0	0	0	0	A_{C3}	0	0	0	0	1	-1	=	0
9	0	A_{C1}	0	A_{C2}	0	0	0	0	0	0	-1	=	0
10	0	0	0	0	1	0	0	1	0	0	0	=	F1
11	0	A_{C1}	0	0	$-A_{C3}$	0	0	0	-1	0	0	=	0
12	0	0	0	0	A_{C3}	0	0	0	0	0	0	=	$(Rec) \times A_{F1} \times F1$
13	0	0	0	0	0	0	0	0	0	-1	(1-CRec)	=	0

Enter a block of info below the matrix containing the known values. Copy the matrix and vector below the info block and calculate the known values using the info block values.



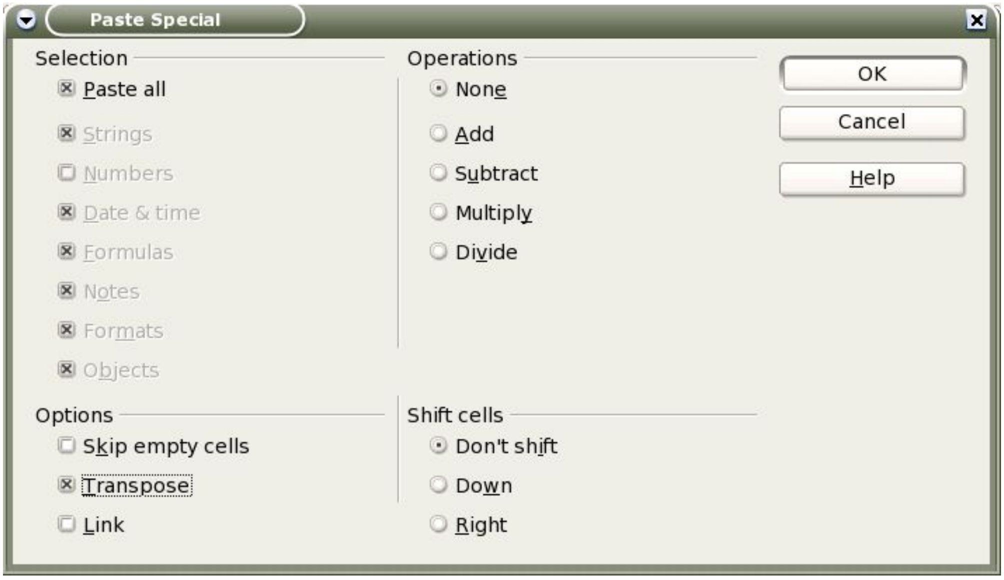
The regression procedure requires an 'error' column with the value “1” be added to the matrix A. This new unknown value, “e” provides the method with a degree of freedom to fit the equations together. More on the derivation of the regression procedure is available on the Internet or in a statistics or mathematics textbook.

Optional step, create “range names” for each of the matrix and vector. Name the vector range “Matrix_A” and the vector range “Vector_Y”.



This screenshot shows the matrix with error column selected in the spreadsheet. The range name is displayed in one of the boxes in the top-left of the window.

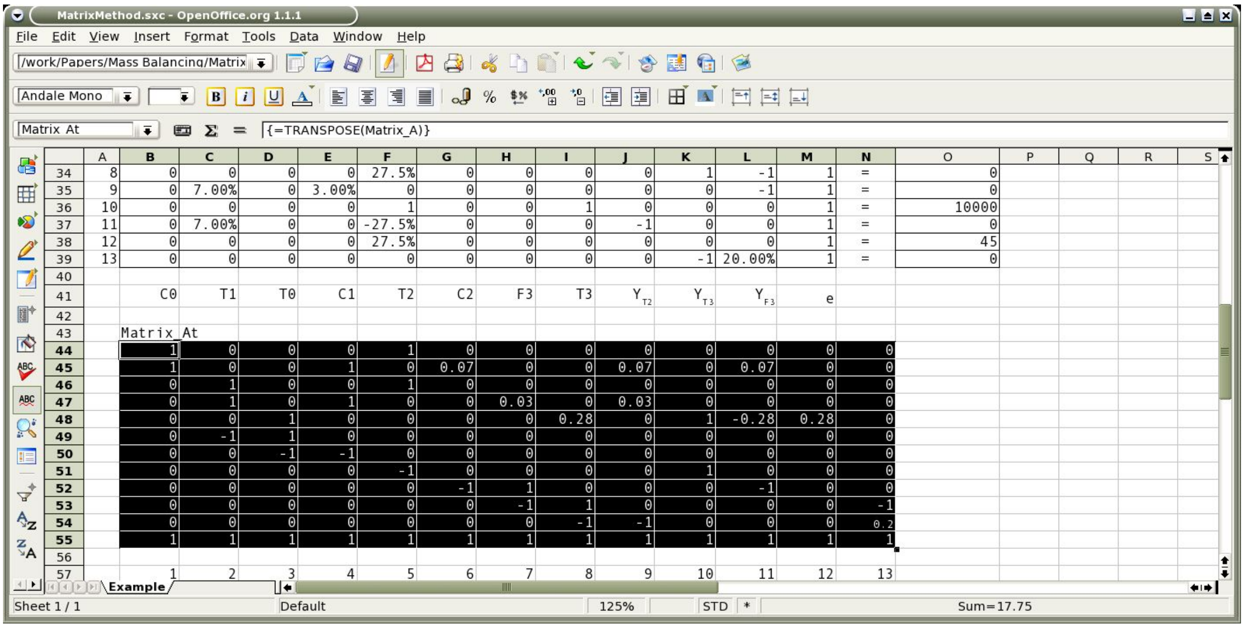
The next step in the procedure is to create a transpose of matrix A. This transpose will contain as many columns as Matrix_A has rows, and as many rows as Matrix_A has columns. A shortcut to get a properly sized matrix transpose is to copy the matrix, and paste-special and select “Transpose”.



After the paste, the range selected will be the correct size for the matrix transpose.

If the optional range name is set, then type the formula “=transpose(Matrix_A)” and press Ctrl-Shift-Enter on the keyboard. The formula without a range name will depend on where the matrix is positioned on the spreadsheet; in this example the alternate formula is “=transpose (B27:M13)” followed by the Ctrl-Shift-Enter keystroke. The formula will appear with {curly braces} if it is correctly composed.

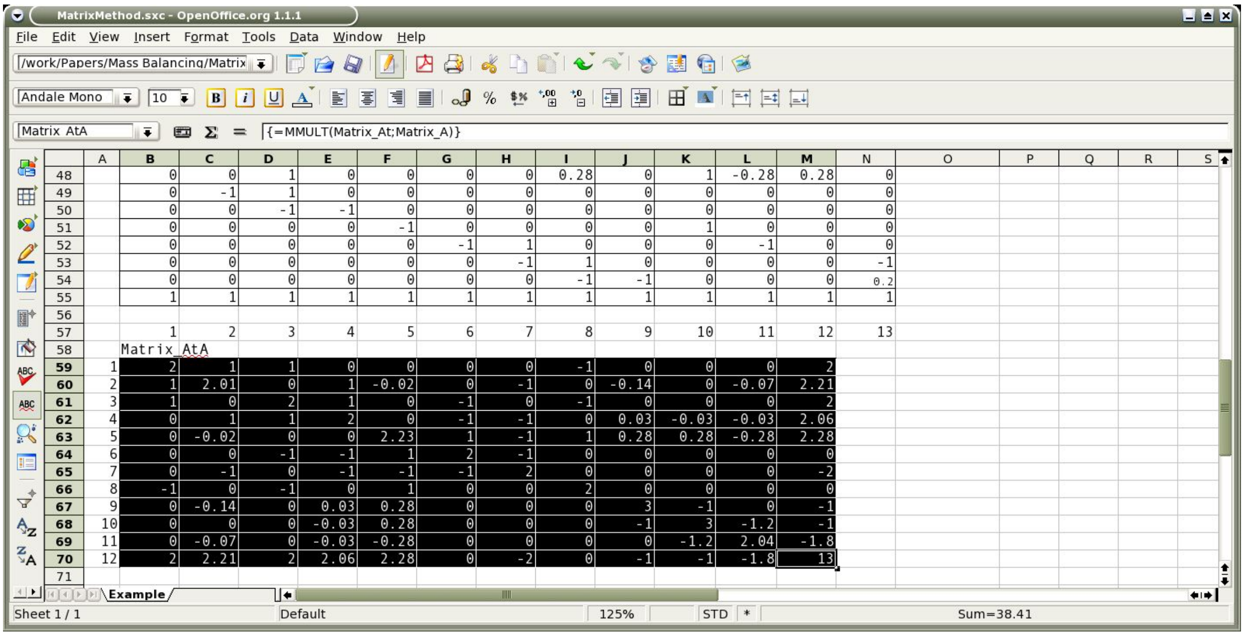
Optional, name this transposed matrix “Matrix_At”.



The next step is $(A^T \cdot A)$ the multiplication[3] of the transposed matrix by matrix A. Count the number of columns in Matrix_A. Label a square area with this number of columns and rows. This example consists of 12 columns in Matrix_A, so the matrix multiplication result will be a 12x12 matrix. Select the output range, type “=mmult(Matrix_At;Matrix_A)”, and press “ctrl-shift-enter”. The semi-colon in this formula may need to be replaced with a comma depending upon the nationality your computer is configured to.

3 Definition of multiplication of matrices: http://www.mathwords.com/m/matrix_multiplication.htm

Name this new range "Matrix_AtA".



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
48		0	0	1	0	0	0	0	0.28	0	1	-0.28	0.28	0
49		0	-1	1	0	0	0	0	0	0	0	0	0	0
50		0	0	-1	-1	0	0	0	0	0	0	0	0	0
51		0	0	0	0	-1	0	0	0	0	1	0	0	0
52		0	0	0	0	0	-1	1	0	0	0	-1	0	0
53		0	0	0	0	0	0	-1	1	0	0	0	0	-1
54		0	0	0	0	0	0	0	-1	-1	0	0	0	0.2
55		1	1	1	1	1	1	1	1	1	1	1	1	1
56														
57		1	2	3	4	5	6	7	8	9	10	11	12	13
58		Matrix_AtA												
59	1	2	1	1	0	0	0	0	-1	0	0	0	2	
60	2	1	2.01	0	1	-0.02	0	-1	0	-0.14	0	-0.07	2.21	
61	3	1	0	2	1	0	-1	0	-1	0	0	0	2	
62	4	0	1	1	2	0	-1	-1	0	0.03	-0.03	-0.03	2.06	
63	5	0	-0.02	0	0	2.23	1	-1	1	0.28	0.28	-0.28	2.28	
64	6	0	0	-1	-1	1	2	-1	0	0	0	0	0	
65	7	0	-1	0	-1	-1	-1	2	0	0	0	0	-2	
66	8	-1	0	-1	0	1	0	0	2	0	0	0	0	
67	9	0	-0.14	0	0.03	0.28	0	0	0	3	-1	0	-1	
68	10	0	0	0	-0.03	0.28	0	0	0	-1	3	-1.2	-1	
69	11	0	-0.07	0	-0.03	-0.28	0	0	0	0	-1.2	2.04	-1.8	
70	12	2	2.21	2	2.06	2.28	0	-2	0	-1	-1	-1.8	13	
71														

Check the determinant[4] of Matrix_AtA. The matrix inversion step will only succeed if the determinant is significantly greater than zero. Select a cell to the right of Matrix_AtA and enter the formula "=mdeterm(Matrix_AtA)" and press enter (not "ctrl-shift-enter", just enter).

The example has a determinant of zero, and therefore cannot be solved yet. In the language of mathematicians, "the problem contains too many degrees of freedom relative to the constraints". Even though the matrix contains more equations (constraints) than variables (degrees of freedom), some of the equations are duplicates of the same underlying constraint. Again, the language of mathematicians says that this situation consists of "linearly dependent equations". The example's equations 1 through 5 can be considered to describe the same constraint as equation 10 (it is possible to reorganise equations 1 through 5 to obtain equation 10).

It is necessary to stop programming and return to the derivation to add more equations or remove some unknowns. Any equations added should be "linearly independent" of the existing equations.

Assume a concentration ratio of 4:1 for the scavenger flotation stage (the mass of concentrate is one-quarter the mass of the cleaner tailing). This adds one additional design criteria resulting in a new equation:

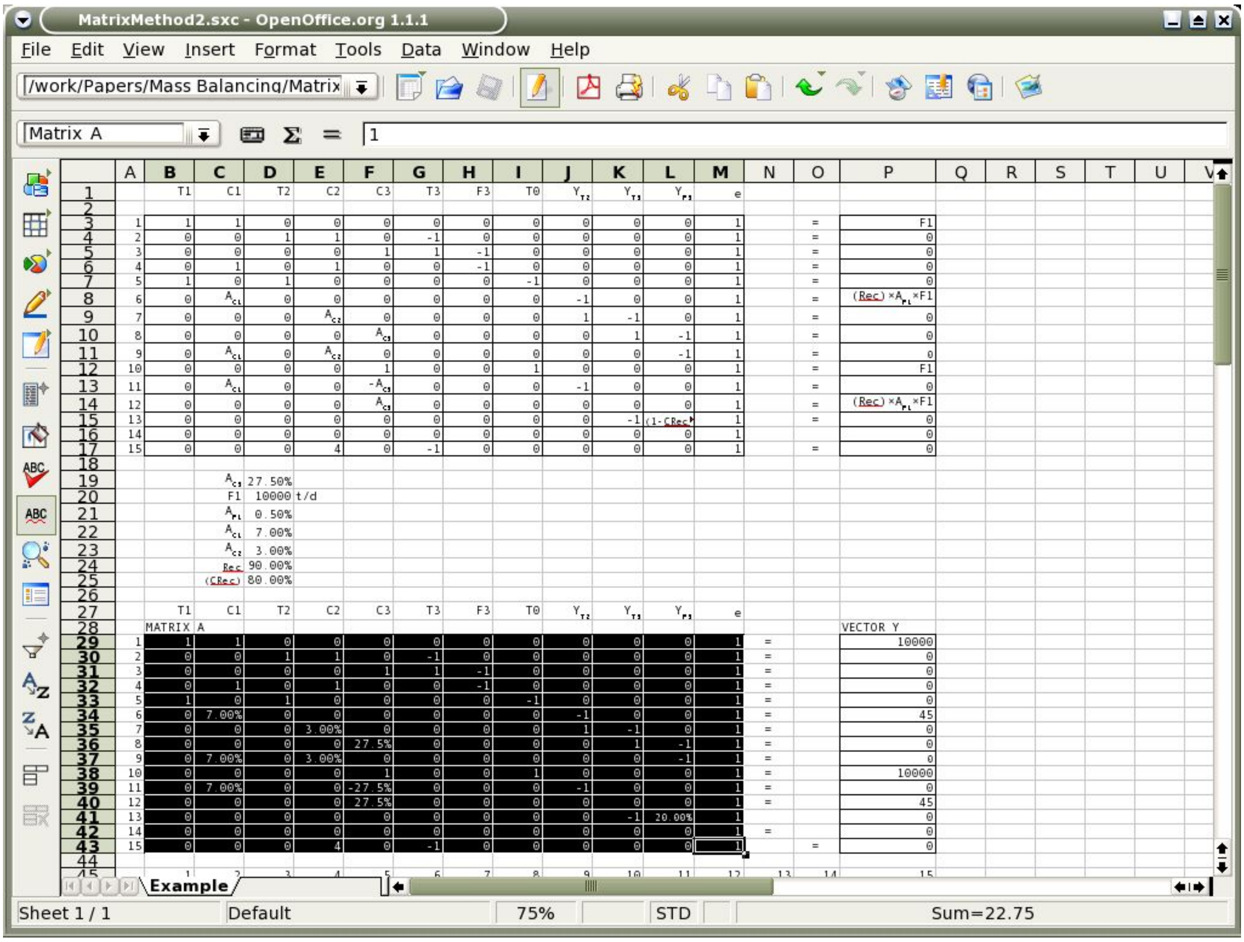
14. 4×C2 - T3 = 0

Start a fresh spreadsheet file and create a new matrix and transpose using the method above. It is strongly recommended that the new equation not be appended to the existing spreadsheet because of the way Array functions and range names misbehave when columns and rows are

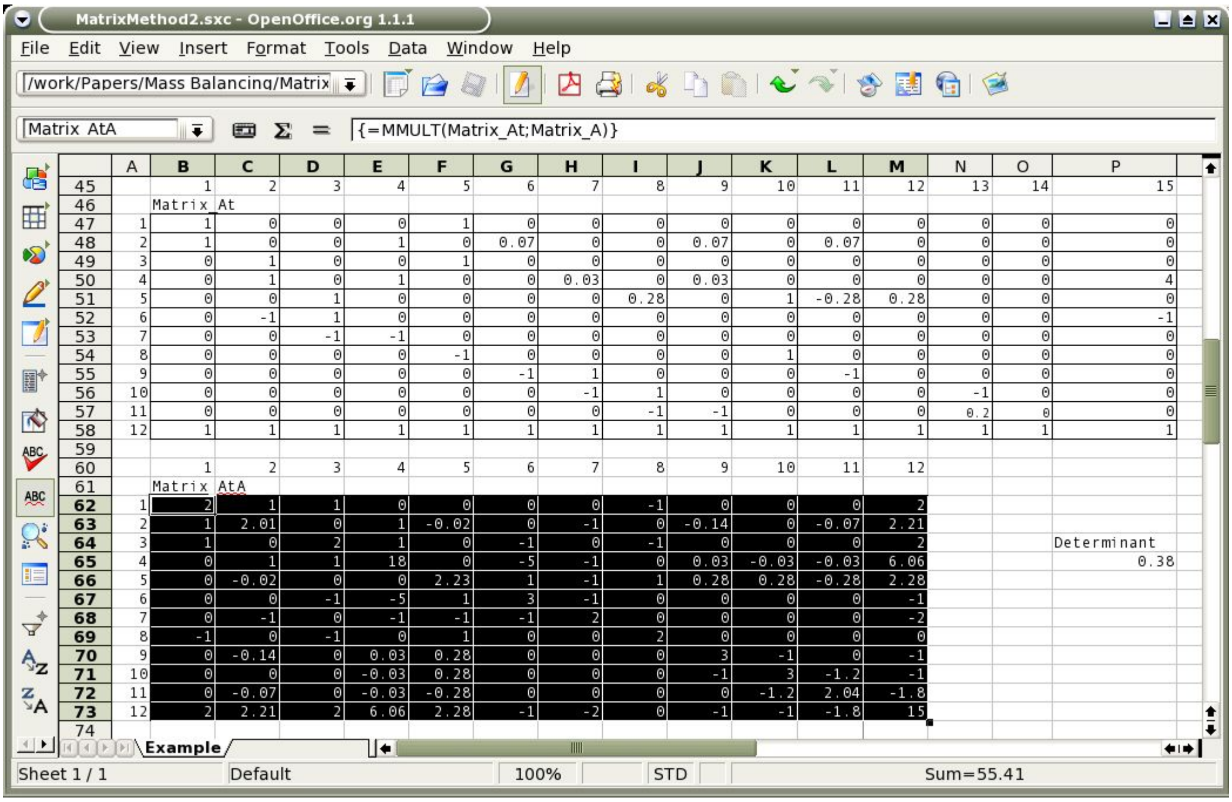
4 Definition of determinant: http://www.mathwords.com/d/determinant.htm

inserted into existing spreadsheets. It is safer to start over than to modify the existing spreadsheet.

The new file will look like this:



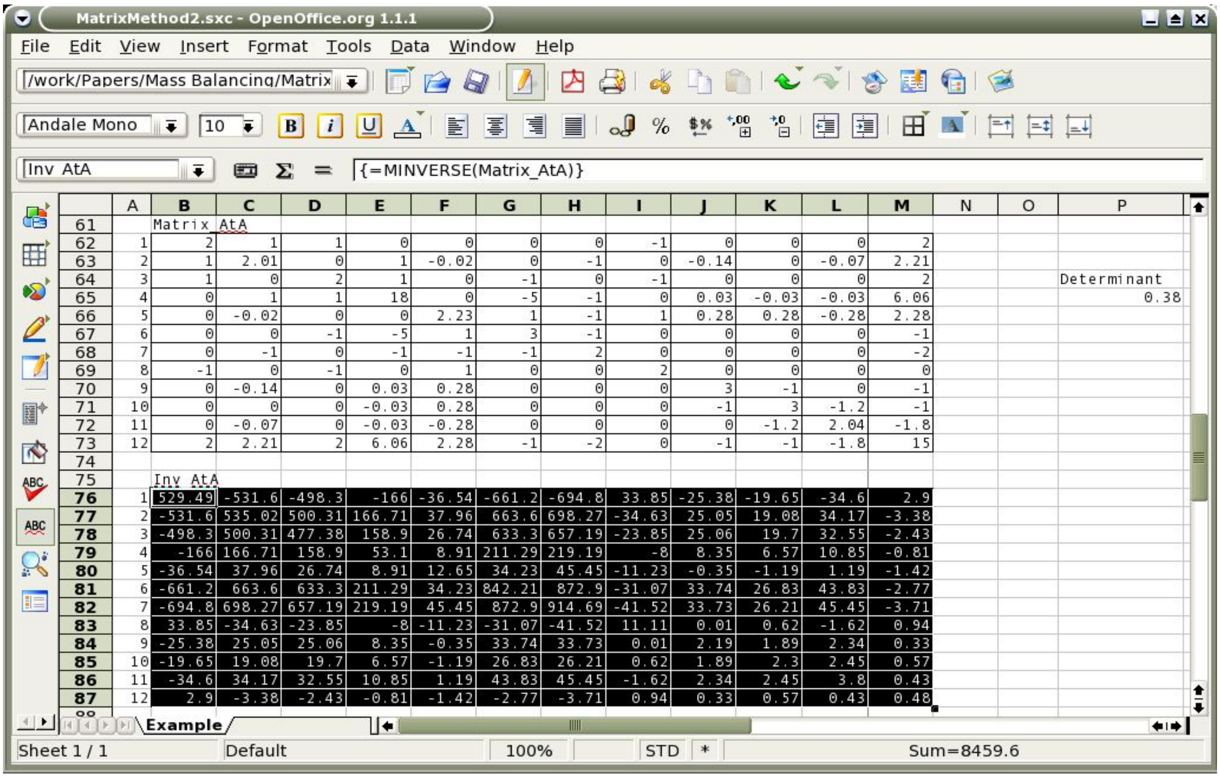
Perform the matrix multiplication and check the determinant.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
45		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
46		Matrix_At														
47	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
48	2	1	0	0	1	0	0.07	0	0	0.07	0	0.07	0	0	0	0
49	3	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
50	4	0	1	0	1	0	0	0.03	0	0.03	0	0	0	0	0	4
51	5	0	0	1	0	0	0	0	0.28	0	1	-0.28	0.28	0	0	0
52	6	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	-1
53	7	0	0	-1	-1	0	0	0	0	0	0	0	0	0	0	0
54	8	0	0	0	0	-1	0	0	0	0	1	0	0	0	0	0
55	9	0	0	0	0	0	-1	1	0	0	0	-1	0	0	0	0
56	10	0	0	0	0	0	0	-1	1	0	0	0	0	-1	0	0
57	11	0	0	0	0	0	0	0	-1	-1	0	0	0	0.2	0	0
58	12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
59																
60		1	2	3	4	5	6	7	8	9	10	11	12			
61		Matrix AtA														
62	1	2	1	1	0	0	0	0	-1	0	0	0	2			
63	2	1	2.01	0	1	-0.02	0	-1	0	-0.14	0	-0.07	2.21			
64	3	1	0	2	1	0	-1	0	-1	0	0	0	2			Determinant
65	4	0	1	1	18	0	-5	-1	0	0.03	-0.03	-0.03	6.06			0.38
66	5	0	-0.02	0	0	2.23	1	-1	1	0.28	0.28	-0.28	2.28			
67	6	0	0	-1	-5	1	3	-1	0	0	0	0	-1			
68	7	0	-1	0	-1	-1	-1	2	0	0	0	0	-2			
69	8	-1	0	-1	0	1	0	0	2	0	0	0	0			
70	9	0	-0.14	0	0.03	0.28	0	0	0	3	-1	0	-1			
71	10	0	0	0	-0.03	0.28	0	0	0	-1	3	-1.2	-1			
72	11	0	-0.07	0	-0.03	-0.28	0	0	0	0	-1.2	2.04	-1.8			
73	12	2	2.21	2	6.06	2.28	-1	-2	0	-1	-1	-1.8	15			
74																

The determinant is significantly greater than zero, so the example may now be solved using the regression method. It contains sufficient constraints (linearly independent equations) to solve the unknowns.

Invert[5] the square Matrix_AtA $(A^T \cdot A)^{-1}$ using the =MInverse() array function (the dimensions of the inverted matrix will be the same as the dimensions of Matrix_AtA). Name the inverted matrix "Inv_AtA".



Matrix_AtA

	B	C	D	E	F	G	H	I	J	K	L	M
1	2	1	1	0	0	0	0	-1	0	0	0	2
2	1	2.01	0	1	-0.02	0	-1	0	-0.14	0	-0.07	2.21
3	1	0	2	1	0	-1	0	-1	0	0	0	2
4	0	1	1	18	0	-5	-1	0	0.03	-0.03	-0.03	6.06
5	0	-0.02	0	0	2.23	1	-1	1	0.28	0.28	-0.28	2.28
6	0	0	-1	-5	1	3	-1	0	0	0	0	-1
7	0	-1	0	-1	-1	-1	2	0	0	0	0	-2
8	-1	0	-1	0	1	0	0	2	0	0	0	0
9	0	-0.14	0	0.03	0.28	0	0	0	3	-1	0	-1
10	0	0	0	-0.03	0.28	0	0	0	-1	3	-1.2	-1
11	0	-0.07	0	-0.03	-0.28	0	0	0	0	-1.2	2.04	-1.8
12	2	2.21	2	6.06	2.28	-1	-2	0	-1	-1	-1.8	15

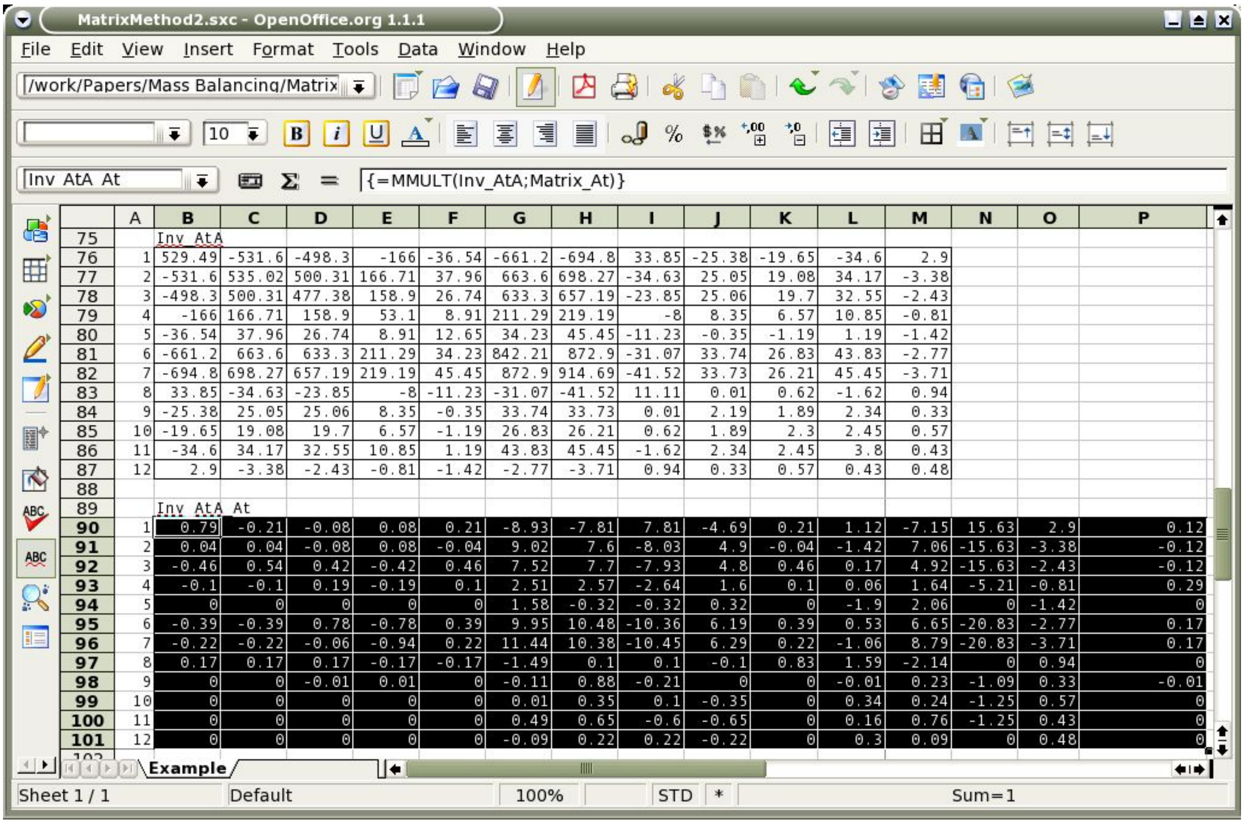
Determinant 0.38

Inv_AtA

	B	C	D	E	F	G	H	I	J	K	L	M
1	529.49	-531.6	-498.3	-166	-36.54	-661.2	-694.8	33.85	-25.38	-19.65	-34.6	2.9
2	-531.6	535.02	500.31	166.71	37.96	663.6	698.27	-34.63	25.05	19.08	34.17	-3.38
3	-498.3	500.31	477.38	158.9	26.74	633.3	657.19	-23.85	25.06	19.7	32.55	-2.43
4	-166	166.71	158.9	53.1	8.91	211.29	219.19	-8	8.35	6.57	10.85	-0.81
5	-36.54	37.96	26.74	8.91	12.65	34.23	45.45	-11.23	-0.35	-1.19	1.19	-1.42
6	-661.2	663.6	633.3	211.29	34.23	842.21	872.9	-31.07	33.74	26.83	43.83	-2.77
7	-694.8	698.27	657.19	219.19	45.45	872.9	914.69	-41.52	33.73	26.21	45.45	-3.71
8	33.85	-34.63	-23.85	-8	-11.23	-31.07	-41.52	11.11	0.01	0.62	-1.62	0.94
9	-25.38	25.05	25.06	8.35	-0.35	33.74	33.73	0.01	2.19	1.89	2.34	0.33
10	-19.65	19.08	19.7	6.57	-1.19	26.83	26.21	0.62	1.89	2.3	2.45	0.57
11	-34.6	34.17	32.55	10.85	1.19	43.83	45.45	-1.62	2.34	2.45	3.8	0.43
12	2.9	-3.38	-2.43	-0.81	-1.42	-2.77	-3.71	0.94	0.33	0.57	0.43	0.48

5 Definition of matrix inversion: http://www.mathwords.com/i/inverse_of_a_matrix.htm

Multiply the inverted matrix by the transposed matrix $(A^T \cdot A)^{-1} \cdot A^T$. Name the resulting matrix Inv_AtA_At. The dimensions of this matrix will be the same as the transpose Matrix_At.

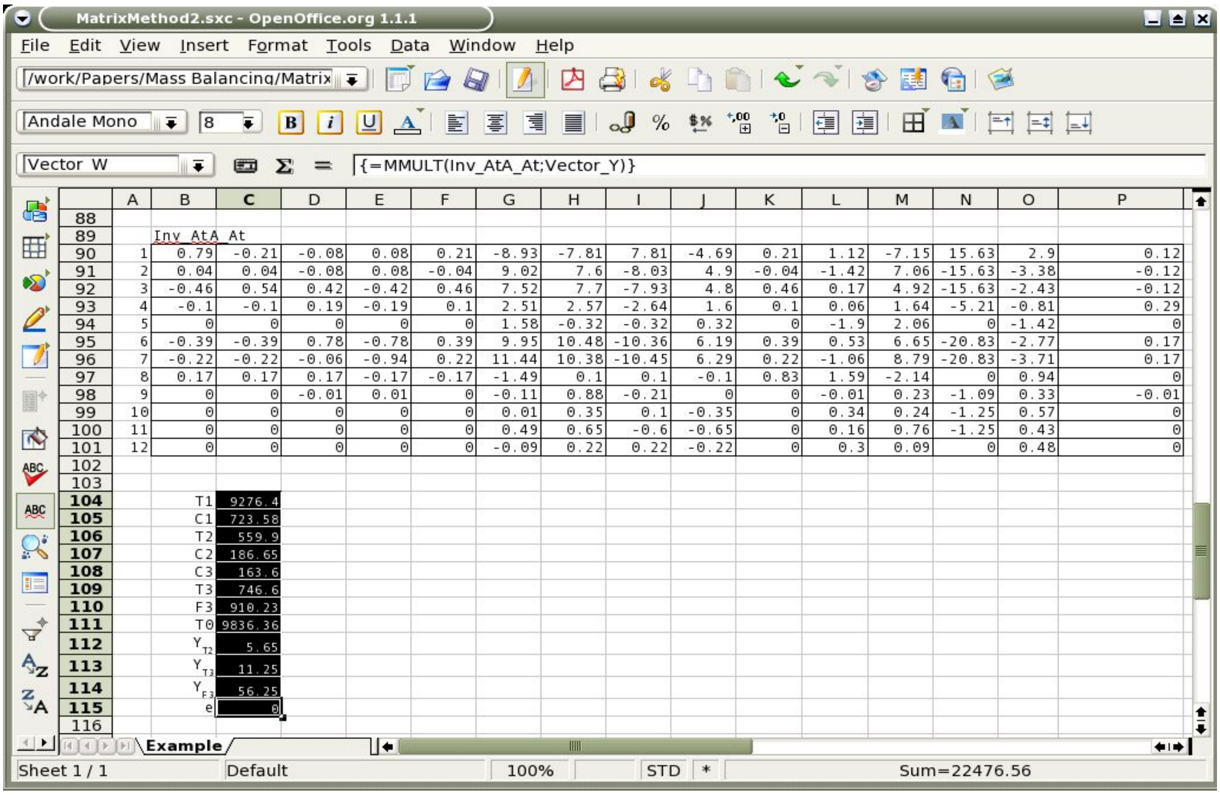


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
75		Inv_AtA														
76	1	529.49	-531.6	-498.3	-166	-36.54	-661.2	-694.8	33.85	-25.38	-19.65	-34.6	2.9			
77	2	-531.6	535.02	500.31	166.71	37.96	663.6	698.27	-34.63	25.05	19.08	34.17	-3.38			
78	3	-498.3	500.31	477.38	158.9	26.74	633.3	657.19	-23.85	25.06	19.7	32.55	-2.43			
79	4	-166	166.71	158.9	53.1	8.91	211.29	219.19	-8	8.35	6.57	10.85	-0.81			
80	5	-36.54	37.96	26.74	8.91	12.65	34.23	45.45	-11.23	-0.35	-1.19	1.19	-1.42			
81	6	-661.2	663.6	633.3	211.29	34.23	842.21	872.9	-31.07	33.74	26.83	43.83	-2.77			
82	7	-694.8	698.27	657.19	219.19	45.45	872.9	914.69	-41.52	33.73	26.21	45.45	-3.71			
83	8	33.85	-34.63	-23.85	-8	-11.23	-31.07	-41.52	11.11	0.01	0.62	-1.62	0.94			
84	9	-25.38	25.05	25.06	8.35	-0.35	33.74	33.73	0.01	2.19	1.89	2.34	0.33			
85	10	-19.65	19.08	19.7	6.57	-1.19	26.83	26.21	0.62	1.89	2.3	2.45	0.57			
86	11	-34.6	34.17	32.55	10.85	1.19	43.83	45.45	-1.62	2.34	2.45	3.8	0.43			
87	12	2.9	-3.38	-2.43	-0.81	-1.42	-2.77	-3.71	0.94	0.33	0.57	0.43	0.48			
88																
89		Inv_AtA_At														
90	1	0.79	-0.21	-0.08	0.08	0.21	-8.93	-7.81	7.81	-4.69	0.21	1.12	-7.15	15.63	2.9	0.12
91	2	0.04	0.04	-0.08	0.08	-0.04	9.02	7.6	-8.03	4.9	-0.04	-1.42	7.06	-15.63	-3.38	-0.12
92	3	-0.46	0.54	0.42	-0.42	0.46	7.52	7.7	-7.93	4.8	0.46	0.17	4.92	-15.63	-2.43	-0.12
93	4	-0.1	-0.1	0.19	-0.19	0.1	2.51	2.57	-2.64	1.6	0.1	0.06	1.64	-5.21	-0.81	0.29
94	5	0	0	0	0	0	1.58	-0.32	-0.32	0.32	0	-1.9	2.06	0	-1.42	0
95	6	-0.39	-0.39	0.78	-0.78	0.39	9.95	10.48	-10.36	6.19	0.39	0.53	6.65	-20.83	-2.77	0.17
96	7	-0.22	-0.22	-0.06	-0.94	0.22	11.44	10.38	-10.45	6.29	0.22	-1.06	8.79	-20.83	-3.71	0.17
97	8	0.17	0.17	0.17	-0.17	-0.17	-1.49	0.1	0.1	-0.1	0.83	1.59	-2.14	0	0.94	0
98	9	0	0	-0.01	0.01	0	-0.11	0.88	-0.21	0	0	-0.01	0.23	-1.09	0.33	-0.01
99	10	0	0	0	0	0	0.01	0.35	0.1	-0.35	0	0.34	0.24	-1.25	0.57	0
100	11	0	0	0	0	0	0.49	0.65	-0.6	-0.65	0	0.16	0.76	-1.25	0.43	0
101	12	0	0	0	0	0	-0.09	0.22	0.22	-0.22	0	0.3	0.09	0	0.48	0

Finally, multiply the previous matrix by the original vector Y. This is the final step in the programming of the regression equation $\vec{w}=(A^T \cdot A)^{-1} \cdot A^T \cdot \vec{y}$. The dimensions of the output will be a single column with as many rows as there are unknowns (including the 'e' term).

Copy and paste-special, transpose the labels for the unknowns from above Matrix_A to a location at the bottom of the spreadsheet (this gives the list of the names of the unknowns, in order. C0, T1, T0, and so on).

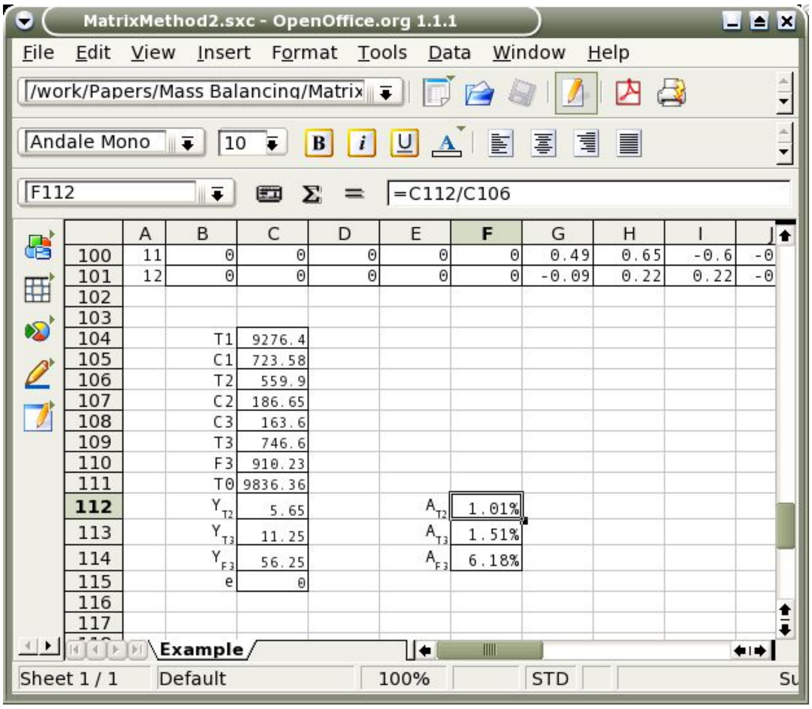
Perform the final matrix multiply =MMULT(Inv_AtA_At;Vector_Y) to create the output vector. Name this range Vector_W.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
88																
89		Inv_AtA_At														
90	1	0.79	-0.21	-0.08	0.08	0.21	-8.93	-7.81	7.81	-4.69	0.21	1.12	-7.15	15.63	2.9	0.12
91	2	0.04	0.04	-0.08	0.08	-0.04	9.02	7.6	-8.03	4.9	-0.04	-1.42	7.06	-15.63	-3.38	-0.12
92	3	-0.46	0.54	0.42	-0.42	0.46	7.52	7.7	-7.93	4.8	0.46	0.17	4.92	-15.63	-2.43	-0.12
93	4	-0.1	-0.1	0.19	-0.19	0.1	2.51	2.57	-2.64	1.6	0.1	0.06	1.64	-5.21	-0.81	0.29
94	5	0	0	0	0	0	1.58	-0.32	-0.32	0.32	0	-1.9	2.06	0	-1.42	0
95	6	-0.39	-0.39	0.78	-0.78	0.39	9.95	10.48	-10.36	6.19	0.39	0.53	6.65	-20.83	-2.77	0.17
96	7	-0.22	-0.22	-0.06	-0.94	0.22	11.44	10.38	-10.45	6.29	0.22	-1.06	8.79	-20.83	-3.71	0.17
97	8	0.17	0.17	0.17	-0.17	-0.17	-1.49	0.1	0.1	-0.1	0.83	1.59	-2.14	0	0.94	0
98	9	0	0	-0.01	0.01	0	-0.11	0.88	-0.21	0	0	-0.01	0.23	-1.09	0.33	-0.01
99	10	0	0	0	0	0	0.01	0.35	0.1	-0.35	0	0.34	0.24	-1.25	0.57	0
100	11	0	0	0	0	0	0.49	0.65	-0.6	-0.65	0	0.16	0.76	-1.25	0.43	0
101	12	0	0	0	0	0	-0.09	0.22	0.22	-0.22	0	0.3	0.09	0	0.48	0
102																
103																

	B	C
104	T1	9276.4
105	C1	723.58
106	T2	559.9
107	C2	186.65
108	C3	163.6
109	T3	746.6
110	F3	910.23
111	T0	9836.36
112	Y_{T2}	5.65
113	Y_{T3}	11.25
114	Y_{F3}	56.25
115	e	0
116		

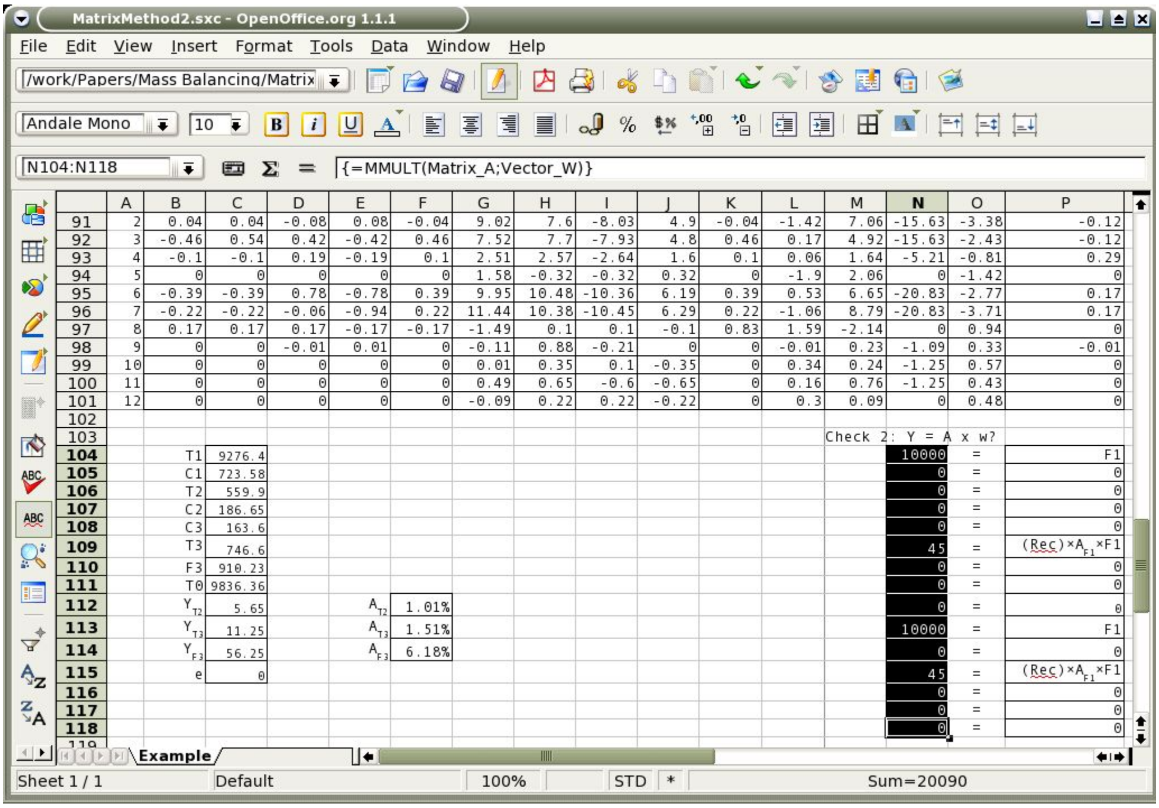
Unravel the 'substituted' unknowns Y_{T2}, Y_{T3} and Y_{F3} by adding fomulae to calculate the assay copper values in these streams.



Checking Results

The calculation is now done, and it is time to check the results to ensure no errors have appeared in the calculation. The first check is to review the results and confirm they are reasonable. In this example, the assays and flows all appear to be roughly the right order of magnitude.

The second check is the equivalent to "plotting the regression line with the raw data points". Run the regression in reverse to see that $\vec{y}=A\cdot\vec{w}$ actually gives us a Vector Y that looks like the Vector Y at the beginning of the spreadsheet.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
91	2	0.04	0.04	-0.08	0.08	-0.04	9.02	7.6	-8.03	4.9	-0.04	-1.42	7.06	-15.63	-3.38	-0.12
92	3	-0.46	0.54	0.42	-0.42	0.46	7.52	7.7	-7.93	4.8	0.46	0.17	4.92	-15.63	-2.43	-0.12
93	4	-0.1	-0.1	0.19	-0.19	0.1	2.51	2.57	-2.64	1.6	0.1	0.06	1.64	-5.21	-0.81	0.29
94	5	0	0	0	0	0	1.58	-0.32	-0.32	0.32	0	-1.9	2.06	0	-1.42	0
95	6	-0.39	-0.39	0.78	-0.78	0.39	9.95	10.48	-10.36	6.19	0.39	0.53	6.65	-20.83	-2.77	0.17
96	7	-0.22	-0.22	-0.06	-0.94	0.22	11.44	10.38	-10.45	6.29	0.22	-1.06	8.79	-20.83	-3.71	0.17
97	8	0.17	0.17	0.17	-0.17	-0.17	-1.49	0.1	0.1	-0.1	0.83	1.59	-2.14	0	0.94	0
98	9	0	0	-0.01	0.01	0	-0.11	0.88	-0.21	0	0	-0.01	0.23	-1.09	0.33	-0.01
99	10	0	0	0	0	0	0.01	0.35	0.1	-0.35	0	0.34	0.24	-1.25	0.57	0
100	11	0	0	0	0	0	0.49	0.65	-0.6	-0.65	0	0.16	0.76	-1.25	0.43	0
101	12	0	0	0	0	0	-0.09	0.22	0.22	-0.22	0	0.3	0.09	0	0.48	0
102																
103													Check 2: Y = A x w?			
104		T1	9276.4											10000	=	F1
105		C1	723.58											0	=	0
106		T2	559.9											0	=	0
107		C2	186.65											0	=	0
108		C3	163.6											0	=	0
109		T3	746.6											45	=	(Rec)×A_{F1}×F1
110		F3	910.23											0	=	0
111		T0	9836.36											0	=	0
112		Y_{T2}	5.65		A_{T2}	1.01%								0	=	0
113		Y_{T3}	11.25		A_{T3}	1.51%								10000	=	F1
114		Y_{F3}	56.25		A_{F3}	6.18%								0	=	0
115		e	0											45	=	(Rec)×A_{F1}×F1
116														0	=	0
117														0	=	0
118														0	=	0

The example returns the sample values in this "Check vector Y" as the original Vector Y, so the calculation is successful.

The third check is highly recommended, although arduous. Spreadsheet packages are notorious for having poorly programmed higher maths functions and the results must be considered suspect until verified by software that uses a different calculation algorithm. Repeat the calculation in another program -- and look closely at the residual error ('e') and the determinant in the second program. If the residual error and determinant exactly match the values in the first program, then the test is inconclusive and must be repeated with a different program. (The second program likely uses the same "software code library[6]" as the first to perform the

6 Software developers commonly construct new software out of older "code libraries" of common functions. These building blocks speed the development of new programs, but any errors or simplifications existing in the code library will also appear in the final software. If a second program is built from the same code library as the first, then the two of them will share the same errors and simplifications contained in the library.

mathematics. Errors in that library will not be evident when comparing the two programs).

In the example, OpenOffice calculated the residual error is $-4.23 \bullet 10^{-12}$ and the determinant as 0.3847219. The same calculation in Excel results in an error of $-1.795 \bullet 10^{-11}$ and a determinant of 0.3849219. The same again, in Lotus 1-2-3 yields error of $4.88 \bullet 10^{-15}$ (but 1-2-3 does not contain a function for matrix determinants). All three programs provide the same material balance results, but do so with "insignificant" differences in the residual error and determinant. Therefore the three programs use different code libraries to perform matrix math, yet return the same material balance results. Therefore this check is successful: it determined that there are no errors in the underlying code library providing a false positive result.

Normalisation, et al.

Matrices are supposed to be "normalised" as part of the regression procedure. The method presented in this paper skips that step due to the system not being overly constrained. Normalisation is needed when doing regression on a large pool of "noisy" data to ensure that all noise is weighted the same when fitting coefficients.

The check that normalisation is not necessary is in the residual error term "e" in Vector W. If the residual error is orders of magnitude smaller than the smallest value in the data set, then normalisation is unnecessary. If the data set is over constrained and the residual error is the same order of magnitude as the data, then refer to a maths textbook and apply the normalising procedure it describes.

In this example, the residual error is on the order of 10^{-12}, whereas the smallest value is on the order of 10^{-2}. The error is significantly smaller than the data, so normalisation is not required.

Conclusion

Application of a "regression equation" matrix algebra approach to material balance calculations, together with spreadsheet software, solve design calculations.

The example determines a material balance around the flotation circuit where the results are displayed in the output vector and the "unravelled unknowns".