
Using the Mathcad Solver to Teach Portfolio Optimisation

Alejandro Hazera, DBA CPA

Associate Professor of Accounting and International Business, University of Rhode Island

Abstract

This paper introduces a Mathcad program for teaching optimisation. The program, which involves a portfolio of equity shares, is less code oriented than GAMS and reinforces mathematical notation to a greater extent than the Excel solver. Also, the program adjusts automatically for alternative portfolios (other than the one presented) without the need for further programming. Thus, students can concentrate on examining alternative portfolios with little need to change parameters consistently.

Introduction

Several years ago two articles were published in *CHEER* which debated the use of alternative software programs for teaching economic optimisation. Estelles *et al.* (1996) suggested that GAMS (General Algebraic Modeling System) could be used to teach both linear and non-linear optimisation. In a response article, MacDonald (1996) argued that the Excel solver was more familiar to students than GAMS and that Excel required less file manipulation and programming code, which might distract students from the fundamentals of the optimisation problem under consideration. In the years since the publication of these articles, computer mathematics packages (including Mathematica, Maple, Matlab and Mathcad) have gained widespread acceptance as teaching tools for mathematics, engineering and economics. These packages are generally thought to offer students an efficient method for teaching the mathematical fundamentals underlying the model being studied. The purpose of this paper is to demonstrate a program that uses the Mathcad solver to teach optimisation to advanced undergraduate students in a manner which provides students with the opportunity to

learn the basics of optimisation models while also reinforcing mathematical notation commonly used in optimisation problems. The example involves a portfolio of equity shares. The program described in this paper is a full 'start to finish' program that commences with the calculation of the portfolio's holding period returns and finishes with the formulation of the minimum variance set. Also, while the portfolio considered consists of three stocks, the program will automatically adjust for larger numbers of shares.

Stock prices and returns

The data used for the example, which are provided below in the matrix labelled 'Prices' in Figure 1, are the annual closing prices for the Intel Corporation (INTC; col. 0), Cysco Systems (CSCO; col. 1) and Duke Energy (DUK; col. 2) for the period 1990–2002. The Intel Corporation and Cysco Systems are well-known NASDAQ companies while Duke Energy is a large energy company. The companies were chosen to construct a portfolio with two highly correlated stocks (INTC and CSCO) and one stock (DUK) that has little correlation with the other securities in the portfolio.

The first step in calculating the securities' returns is to define two indices that reference the rows (years) and columns (stocks) in 'Prices'. These variables, labelled i and j , are defined in Mathcad notation as follows:

$i := 1..rows(Prices) - 1$

$j := 0..cols(Prices) - 1$

	0	1	2
0	1.17	0.16	9.1
1	1.49	0.46	11.01
2	2.65	1.09	11.96
3	3.79	1.79	14.7
4	3.92	1.95	13.9
5	6.99	4.15	18.14
6	16.17	7.07	18.47
7	17.38	9.29	23.15
8	29.37	23.2	27.81
9	40.85	53.56	22.64
10	29.88	38.25	39.83
11	31.34	18.11	37.7
12	15.57	13.1	19.54

Figure 1.

The variable i represents the year-ends in the portfolio and is a range of numbers that starts at '1' and ends at the number of rows (13) in the matrix 'Prices' shown above less 1 (i.e. 12). The variable j references the stocks and is a range of numbers that starts at '0' and ends at the number of columns (3) in the matrix 'Prices' less 1 (i.e. 2).¹ Each of these ranges can be verified as follows:

$i =$	
1	
2	
3	
4	
5	
6	
7	
8	
9	
	$j =$
10	1
11	2
12	3

These indices provide a point of reference for calculating the holding period returns. Assuming (discrete) annual compounding, the returns may be calculated on the basis of the following formula:

$$\text{Returns}_{i-1,j} := \frac{\text{Prices}_{i,j} - \text{Prices}_{i-1,j}}{\text{Prices}_{i-1,j}}$$

The numerator is the difference between the i, j cell in the matrix 'Prices' less the $i - 1, j$ cell in the matrix 'Prices'. The denominator is the $i - 1, j$ cell in the matrix 'Prices'. The formula results in the array of returns shown in Figure 2. An example of the formula's results can be provided by examining the return in the $i = 0, j = 0$ cell (.2735043) in the matrix 'Returns'. This return is calculated by dividing the difference between the $i = 1, j = 0$ cell in the matrix 'Prices' (1.49) and the $i = 0, j = 0$ cell in the matrix 'Prices' (1.17) by the $i = 0, j = 0$ cell in the matrix 'Prices' (1.17).

The covariance matrix

The next step is to formulate the portfolio's covariance matrix. Two matrices, labelled 'Means' and 'Diff', may be used for this calculation.

		0	1	2
Returns =	0	0.2735043	1.875	0.2098901
	1	0.7785235	1.3695652	0.0862852
	2	0.4301887	0.6422018	0.229097
	3	0.0343008	0.0893855	-0.0544218
	4	0.7831633	1.1282051	0.305036
	5	1.3133047	0.7036145	0.0181918
	6	0.0748299	0.3140028	0.2533839
	7	0.6898734	1.4973089	0.2012959
	8	0.390875	1.3086207	-0.1859044
	9	-0.2685435	-0.2858476	0.7592756
	10	0.0488621	-0.5265359	-0.0534773
	11	-0.5031908	-0.2766427	-0.4816976

Figure 2.

First, the matrix labelled 'Means' is a $1 \times j$ row vector of the mean returns for the three columns (stocks) in the matrix 'Returns'.

The 'Means' matrix can be computed as follows:

$$\text{Means}_{0,j} := \text{mean}(\text{Returns}_{i,j})$$

This equation states that the j th column in the matrix 'Means' is the mean of the j th column in the matrix 'Returns'.

The following row vector of average annual returns results:

$$\text{Means} = (0.33714096 \ 0.65323985 \ 0.10724621)$$

The matrix 'Diff' is an $i \times j$ vector where each cell corresponds to a cell in the matrix 'Returns' less the mean for the corresponding column (stock) in the vector 'Means'.

The matrix 'Diff' (Figure 3) is obtained by subtracting the column entries in 'Means' from each row of the 'Returns'. The following notation is used:

$$\text{Diff}_{i-1,j} := \text{Returns}_{i-1,j} - \text{Means}_{0,j}$$

The equation states that the $i - 1, j$ cell in the matrix 'Diff' is the $i - 1, j$ cell in the matrix 'Returns' less the $0, j$ cell in the row vector 'Means'.

The matrix in Figure 3 results.

Each cell in the matrix is a stock's holding period returns less the mean return for the stock. For example, the $i = 0, j = 0$ cell for the matrix 'Diff' (-0.06363668) is the holding period return found in the $i = 0, j = 0$ cell of the matrix 'Returns' (.2735043) less the mean return found in $i = 0, j = 0$ cell of the vector 'Means' (.33714096).

The covariance matrix can be calculated by multiplying the transpose of 'Diff' by 'Diff' and dividing the resulting matrix by the number of rows in the 'Returns' matrix.

		0	1	2
Diff =	0	-0.06363668	1.22176015	0.1026439
	1	0.44138253	0.71632536	-0.02096101
	2	0.09304772	-0.01103802	0.12185078
	3	-0.30284016	-0.56385438	-0.16166798
	4	0.44602231	0.47496528	0.19778976
	5	0.97616377	0.05037461	-0.08905437
	6	-0.26231102	-0.33923702	0.14613766
	7	0.35273246	0.84406908	0.09404969
	8	0.05373409	0.65538084	-0.29315056
	9	-0.60568441	-0.9390875	0.65202941
	10	-0.28827884	-1.1797758	-0.16072348
	11	-0.84033177	-0.92988259	-0.58894382

Figure 3.

$$\text{Covar} := \frac{(\text{Diff}^T \cdot \text{Diff})}{\text{rows}(\text{Returns})}$$

$$\text{Covar} = \begin{pmatrix} 0.233608 & 0.231784 & 0.014267 \\ 0.231784 & 0.578958 & 0.021025 \\ 0.014267 & 0.021025 & 0.084415 \end{pmatrix}$$

Defining portfolio standard deviation and portfolio return

In optimising the portfolio, the objective is to minimise the portfolio standard deviation (risk) subject to a desired return and a unity constraint. Thus, it is important for students to understand the equations for calculating portfolio standard deviation and return. These equations may be exemplified by formulating an equally weighted portfolio of the three stocks. Importantly, this equally weighted portfolio also constitutes the 'guess values' for the Mathcad solver in the next section of the program. The calculation of portfolio standard deviation focuses on the covariance (Covar) matrix provided above. The first step is to define two new indices based on the rows and columns in the covariance matrix.

$$x := 0.. \text{rows}(\text{Covar}) - 1$$

$$y := 0.. \text{cols}(\text{Covar}) - 1$$

These indices may be verified as follows:

y	x
0	0
1	1
2	2

The standard deviation equation may then be exemplified by formulating the equally weighted portfolio of stocks. Define the vector 'Portweights' as:

$$\text{Portweights}_{x,0} := \frac{1}{\text{cols}(\text{Returns})}$$

This equation results in a column vector consisting of the weights in an equally weighted three-stock portfolio:

$$\text{Portweights} = \begin{pmatrix} 0.3333333 \\ 0.3333333 \\ 0.3333333 \end{pmatrix}$$

The standard deviation for this portfolio is then expressed as the following function:

$$f(\text{Portweights}) := \left[\sum_x \sum_y (\text{Portweights}^T)_{0,y} \cdot \text{Covar}_{x,y} \cdot \text{Portweights}_{x,0} \right]^{\frac{1}{2}}$$

The function results in the equally weighted portfolio's standard deviation:

$$f(\text{Portweights}) = 0.398767$$

In order to exemplify the logic of the double summation in Mathcad, students should be shown that the formula is equivalent to taking the square root of the product of the transpose of the matrix 'Portweights' (a 1×3 matrix) times

the covariance matrix (a 3×3 matrix) times the 'Portweights' (3×1) matrix:

$$\text{var} := \text{Portweights}^T \cdot \text{Covar} \cdot \text{Portweights}$$

$$\text{var} = (0.15901493)$$

$$\sqrt{\text{var}} = (0.39876676)$$

The portfolio return is calculated as:

$$\text{Portret} := \sum_{(y)} \text{Means}_{0,y} \cdot \text{Portweights}_{y,0}$$

The formula results in:

$$\text{Portret} = 0.36587567$$

Students should be shown that summation in 'Portret' is equivalent to multiplying the matrix 'Means' (a 1×3 matrix) by 'Portweights' (a 3×1 matrix).

$$\text{Means} \cdot \text{Portweights} = (0.36587567)$$

Portfolio optimisation

After reviewing the equations for portfolio standard deviation and return, the Mathcad solver may be used to find the portfolio weights of stocks that minimise the portfolio standard deviation for a specific desired return. The first step is to indicate a desired return for the portfolio (which will be .2).

$$\text{desret} := .2$$

The function to be minimised (i.e. the portfolio standard deviation) is then entered. This equation is the exact same function for the standard deviation provided in the section above.

$$f(\text{Portweights}) := \left[\sum_x \sum_y (\text{Portweights}^T)_{0,y} \cdot \text{Covar}_{x,y} \cdot \text{Portweights}_{x,0} \right]^{\frac{1}{2}}$$

A solve block is then initiated by entering the word 'Given', followed by a unity constraint and a desired return constraint.

Given

Unity constraint

$$\sum_{(y)} \text{Portweights}_{y,0} = 1$$

Desired return constraint

$$\sum_{(y)} \text{Portweights}_{y,0} \cdot \text{Means}_{0,y} = \text{desret}$$

Finally, the solve block is terminated by assigning a variable (e.g. 'optimalweights') which defines the objective function of minimising the standard deviation as defined in the function 'f(Portweights)' provided above.

$$\text{optimalweights} := \text{Minimize}(f, \text{Portweights})$$

By entering 'optimalweights', Mathcad provides the weights of the three stocks which minimise the portfolio standard deviation for the desired return:

$$\text{optimalweights} = \begin{pmatrix} 0.24636641 \\ 0.06614628 \\ 0.68748732 \end{pmatrix}$$

The portfolio's standard deviation may be calculated by entering an equation similar to the objective function provided above. In this case, however, the variable 'optimalweights' replaces 'Portfolioweights'.

$$\text{Portfoliosd} := \left[\sum_x \sum_y \left(\text{optimalweights}^T \right)_{0,y} \cdot \text{Covar}_{x,y} \cdot \text{optimalweights}_{x,0} \right]^{\frac{1}{2}}$$

$$\text{Portfoliosd} = 0.26628872$$

Calculating the minimum variance set

A desirable characteristic in any solver is to be able to examine different scenarios by changing the right-hand side (RHS) value of a constraint. In the problem provided above, this may be done by examining the standard deviations and portfolio weights for different desired returns. The first step is to define a range of 'desired returns' for which the variance will be minimised. We will use a range of returns that begins at .1 and ends at 1. These 'endpoints' may be defined as follows.

$$\text{first} := .10$$

$$\text{final} := 1.00$$

A desired interval (say .1) is then specified.

$$\text{interval} := .1$$

The variable h is used to define the number of steps in the vector of returns:

$$h := \frac{(\text{final} - \text{first})}{\text{interval}}$$

$$h = 9$$

An index, i , is then created which commences at '0' and ends at 'h':

$$\text{interval} := .1$$

The range i is:

i
0
1
2
3
4
5
6
7
8
9

The desired return (desret) vector is then defined on the basis of the following linear equation:

$$\text{desret}_i := \text{first} + (i \text{ interval})$$

	0
0	0.1
1	0.2
2	0.3
3	0.4
desret = 4	0.5
5	0.6
6	0.7
7	0.8
8	0.9
9	1

This vector of desired returns is then applied to virtually the same solve block provided above. The only difference is that the subscript i is added to the variables 'desret' and 'optimalweights'.

$$f(\text{Portweights}) := \sum_x \sum_y \left(\text{Portweights}^T \right)_{0,y} \cdot \text{Covar}_{x,y} \cdot \text{Portweights}_{x,0}$$

Given

$$\sum_{(y)} \text{Portweights}_{y,0} \cdot \text{Means}_{0,y} = \text{desret}_i$$

$$\sum_{(y)} \text{Portweights}_{y,0} = 1$$

In this case the variable 'optimalweights' returns in the following matrix:

	0
0	[3,1]
1	[3,1]
2	[3,1]
3	[3,1]
optimalweights = 4	[3,1]
5	[3,1]
6	[3,1]
7	[3,1]
8	[3,1]
9	[3,1]

In Mathcad notation, each cell in 'optimalweights' incorporates a 3×1 matrix which corresponds with the portfolio weights for each of the desired returns provided above. For example, $\text{optimalweights}_{0,0}$ provides the portfolio weights for a desired return of 10% (the desired return in row 0 in the vector 'desret' above):

$$\text{optimalweights}_{0,0} = \begin{pmatrix} 0.26803448 \\ -0.12612953 \\ 0.85809505 \end{pmatrix}$$

Also, $\text{optimalweights}_{1,0}$ provides the portfolio weights for a desired return of 20% (the desired return in row 1 in the vector 'desret' above):

$$\text{optimalweights}_{1,0} = \begin{pmatrix} 0.24636641 \\ 0.06614628 \\ 0.68748732 \end{pmatrix}$$

The following equation is then used to form a matrix of variances for each of the portfolios in 'optimalweights'.

$$\text{var}_i := \text{optimalweights}_i^T \cdot \text{Covar} \cdot \text{optimalweights}_i$$

	0
0	[1,1]
1	[1,1]
2	[1,1]
3	[1,1]
4	[1,1]
5	[1,1]
6	[1,1]
7	[1,1]
8	[1,1]
9	[1,1]

In Mathcad notation, the matrix 'var' consists of ten 1 x 1 matrices. Each 1 x 1 matrix contains the portfolio variance for each desired return. The standard deviations may be obtained by taking the square root of each variance. The following notation is used:

$$\text{sd}_i := \sqrt{\text{var}_i}$$

The vector of standard deviations is:

	0
0	[1,1]
1	[1,1]
2	[1,1]
3	[1,1]
4	[1,1]
5	[1,1]
6	[1,1]
7	[1,1]
8	[1,1]
9	[1,1]

The following notation (for a new matrix 'Sd') may be used to produce the entire matrix of standard deviations:

$$\text{Sd}_i := (\text{sd}_{i,0})_{0,0}$$

	0
0	0.27292921
1	0.26628872
2	0.32993966
3	0.43398368
4	0.55619662
5	0.68694882
6	0.82217674
7	0.95998967
8	1.09941697
9	1.23991384

The portfolio standard deviations may then be graphed against the desired returns to obtain a representation of the minimum variance set (Figure 4).

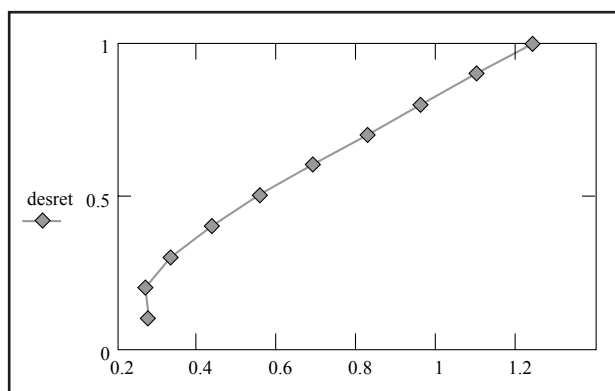


Figure 4. Minimum variance set.

At this point, students may use either the initial portfolio optimisation program or the extension of the program to explore allocation scenarios between the three assets. For example, the equally weighted portfolio discussed above provided a return of .36587 and a standard deviation of .398767. Students should use the program to see whether a different allocation can provide the same return with a lower standard deviation. In fact, an allocation of .21042, .38509 and .44049 provides the same return but a standard deviation of .395659. This same scenario may be illustrated graphically on the minimum variance set.

Relative limitations and advantages

While I am an enthusiastic user of Mathcad for teaching purposes, relative to Excel, the program is generally unfamiliar to students. As a result, it takes students a bit more time to become accustomed to methods for inputting variables and data as well as understanding the program's logic (e.g. solve blocks). Consequently, the program may be best for students who have a familiarity with optimisation in a spreadsheet environment (see, for example, Benninga, 1998), but who wish to extend their knowledge of optimisation to computer mathematics packages.

Once students are familiar with the problem, however, Mathcad provides a great deal of flexibility which enhances students' ability to modify the optimisation problem. For example, simple short sales may be disallowed simply by adding non-negativity constraints. Also, as shown above, the solver may be 'called up' to examine different scenarios without the need for simple but repetitive programming code (for example, Benninga, 1998). Finally, alternative portfolios of securities (with any number of securities) may be examined simply by reading new prices into the initial ('Prices') matrix describe above. The program will automatically adjust for many aspects of the portfolio, including weights, risk, return and the minimum variance set. For the more adventurous, it is even possible to formulate a full program that solves the same optimisation problem through lagrange multipliers, and which shows the marginal changes in risk. Such a program was used to verify the optimal weights provided by the solver-based program described in this paper.

Conclusion

Mathcad provides a third alternative to optimisation problems. Since Mathcad is less 'code oriented' than GAMS, it allows students to formulate optimisation problems in a manner that is closer to the notation found in most investment texts. Also, since Mathcad allows the direct formulation of equations and relies less on 'checking' of options in dialog boxes, it provides greater flexibility in allowing the students directly to formulate and test the results of models based on the fundamental mathematics discussed in texts. Finally, Mathcad allows models to be formulated in many different ways.

Given that it requires a fundamental knowledge of the mathematics involved in an optimisation problem, the use of Mathcad is probably most effective for advanced undergraduate or graduate courses. However, knowledge of Mathcad or other mathematical computer programs will provide students with a familiarity of optimisation in the context of the increasingly important world of computer mathematics packages.

Notes

- ¹ By default, Mathcad assigns the '0' as the first column or row of matrices. These settings may be changed by the user.

References

- Benninga, S. (1998) *Financial Modeling* (2nd edn), Cambridge, MA: MIT Press.
- Estelles, T. C., Arce, M. M. and Garrido, R. S. (1996) 'Economic optimisation with GAMS', *Computers in Higher Education Economics Review*, 10(2), pp. 2-7.
- MacDonald, Z. (1996) 'Economic optimisation, an Excel alternative to Estelles *et al*'s GAMS approach', *Computers in Higher Education Economic Review*, 10(3), pp. 2-5.

Contact details

Alejandro Hazera, DBA CPA
Associate Professor of Accounting and International Business
College of Business Administration
University of Rhode Island
Kingston RI 02881
USA

Tel: (401) 874-4332
Fax: (401) 874-4312
Email: Hazera@uriacc.uri.edu