An algorithm for company valuation *

A. Christofides

Centre for Quantitative Finance, Imperial College, London SW7 2BX
Correspondence: a.christofides@imperial.ac.uk

EU IST Programme: EUROSIGNAL Project: Deliverable D3.4

May 2004

ABSTRACT

We describe the complete company valuation algorithm whose individual components were previously developed in deliverables D3.1, D3.2 and D3.3. In the description given in this report, we concentrate on the mechanics and sequence of actions that need to be performed in order to value a company, rather than on how to perform an individual action, for which the reader is referred to the earlier deliverables. This report also describes the file structures and data structures used by this algorithm and gives computational results for 10 companies chosen from the FTSE100 list. For the valuation of these companies, we have chosen 20 or so financial factors (available as time series) for which a reasonable argument can be made that they are of fundamental importance to company value. These factors defined our state space.

*Support for this work was provided by the IST Framework 5 Programme of the European Union, Contract IST2000-29405, Eurosignal Project

Although computing times are very long, company valuation is not an activity requiring instant response (contrary to simple option pricing, for example) and the algorithm is, at least from the computational point of view, practically acceptable. The results obtained show that in the large majority of cases the computed company values are much lower than the market values. These may or may not be ”errors” but the differences may at least in part be due to the fact that insufficient effort was spent by us on each company to ensure that an accurate estimate is produced of those parameters that affect real option value.

Keywords: Company valuation, Independent Components, Stochastic Modelling, Dynamic Programming, Real Options

1 Introduction

Deliverable D3.1 was concerned with the modelling of asset 'values', 'prices' or 'returns' using a suitable stochastic process and calibrating its parameters to fit the historical asset time-series. Deliverable D3.2 was concerned with the representation of the models described in D3.1 as multidimensional state-space transition graphs thus implementing in discrete time the stochastic process dynamics. Deliverable D3.3 was concerned with two issues. Firstly, because the number of 'assets' that are needed in order to value a company is large and the resulting state-space transition graph is huge, it was found necessary to try and reduce this number. Deliverable D3.3 examined methods of achieving such a reduction using techniques varying from the well-known Principal Components, to the less well-known Independent Components, to new non-linear Independent Component methods. Secondly, Deliverable D3.3 introduced a company valuation methodology, based on option pricing principles, namely the 'replication' of a company's stochastic future cash-flows using other tradable assets. The resulting dynamic programming recursions given in deliverable D3.3 are very much more complex than the corresponding recursions for options, but have been demonstrated to be do-able, albeit after a very long computing time.

The present report uses the results of the above deliverables to construct the final algorithm developed during the EUROSIGNAL project for company valuation. It consists of 3 parts.
Firstly, the complete algorithm is described in one place but with references to the earlier deliverables.
Secondly, the computational implementation is discussed in terms of the data structures used and the file structures employed for intermediate results.
Thirdly, computational results are given for the valuation of 10 FTSE100 companies on the London Stock Exchange, chosen from various sectors.

2 Algorithm description

2.1 Assumptions

To simplify the algorithm description we assume that the valuation of a company is required for ”today”, a time that we index by 0. Forward times (periods) are indexed $1, 2, 3, \ldots$. Backward (historical) times are indexed $-1, -2, -3, \ldots$.
As we see later, the main inputs to the algorithm are time series. Depending on the time series, the *period* (i.e. the actual time duration between two consecutive observations) can be one day, one month, one quarter, half a year or a year. Other frequencies of observation, for example tick-by-tick data, are not supported. The assumption is made that a month is composed of 21 working days and the number of working days in a quarter, half-year and year are computed accordingly.

2.2 Time series

The actual time series required as inputs to the algorithm are discussed in a later section. Here, we discuss some of the general issues that arise in the preprocessing of the time series stored in the EUROSIGNAL database. This preprocessing is performed by the initialization part of the algorithm and does not affect the data in the database.

2.2.1 Missing data

It is quite often the case that time series of data downloaded from any database, will have missing values on some dates. This is particularly true of time series of daily data. Some of these missing values are legitimate; for example the value of the Nikkei index on the 29 April, ”Greenery day”, is missing every year since this is a Japanese holiday, whereas it may be a working day anywhere else. Approximately 4% of daily data in financial databases is missing from these reasons. It means, however, that if the G10 economies are examined together, in over 25% of the days there is missing data for one country or another. Other data is missing because it was not recorded for technical reasons, or recorded obviously wrongly and has to be ignored.

The preprocessor in our valuation algorithm fills in missing data in the time series obtained from the EUROSIGNAL database using an advanced EM algorithm [Christodoulou 2002], [Georgikopoulos 2004]. Our improved algorithm considers the fact that financial returns are fat tailed and assumes a student-t joint probability density function (also an elliptic distribution) instead of a Gaussian joint pdf. The theory of the EM algorithm extends seamlessly to this case. Also, this advanced algorithm does not make the assumption that the covariances matrix of the time series is constant. It can vary with time (but must be deterministic).

2.2.2 Change of period

This is a routine problem arising in many other applications involving time series. If two or more time series are to be analysed together (in our case, for example, we need to compute independent components of a set of time series) it is necessary for all the time series in the set to have the same period. Thus, it is necessary to be able to increase or decrease the period of a time series.

In our preprocessor, we take a very simple approach to increasing the frequency of observations (i.e. decreasing the period) from, say, monthly to, say, daily. This involves the fitting of the low-frequency observations with cubic splines [Press et al. 2002] and using interpolation from these curves to fill the higher frequency data points. Note that the precise details depend on the data represented by the time series. For example if the series are monthly sales figures, the interpolated points must be ”normalised” (i.e. divided by a number such that the sum of the daily sales equals the monthly sales) whereas if the series are monthly closing prices the interpolated points can be used directly.

The reverse procedure, namely decreasing the frequency of observations (i.e. increasing the period) from, say, daily to quarterly, is done by simply averaging over the larger period the smaller period observations. Once more the precise details depend on the data represented by the time series.

2.3 Input time series

The time series that are required as input to the company valuation algorithm are obtained from the EUROSIGNAL database and are as follows: (Note that time series of tradable assets are indicated.)

2.3.1 Macro-economic data

These time series represent the macro-economic factors that affect the company. The are clearly outside the control of the company management. The corresponding time series in the EUROSIGNAL database contain 10 years-worth of data. They time series used are:

1. GDP growth (annualised %) in the countries where the company operates.
   These time series are quarterly.

2. For each of the countries above, growth in the particular sector that the company belongs to.
   Time series are quarterly.

3. For each of the countries above, long-term interest rates.
   The 15-year swap rate. Time series are daily.

4. For each of the countries above, short-term interest rates.
   The 3-month Libor rate. Time series are daily.

5. For each of the countries above, FX rate. (Tradable)
   Time series are daily.

6. For each of the countries above, inflation rate.
   Time series are quarterly.

7. The price of oil - Brent 1-month. (Tradable)
   Time series are daily.

8. The price of any other commodity which is relevant to the company's business. (e.g. wheat, or coffee for Nestle) (Tradable)
   Time series are daily.

2.3.2 Financial assets

These are tradeable assets. The time series used are:

1. For each of the countries where the company operates, the yields on the 2-year Government bond. (Tradable)
   Time series are daily.

2. For each of the countries where the company operates, the yields on the 10-year Government bond. (Tradable)
   Time series are daily.

3. For each of the countries where the company operates, the main stock index on the relevant exchange. (Tradable)
   Time series are daily.

4. For each of the countries where the company operates, the stock index of the sector in which the company belongs. (Note: The EUROSIGNAL database does not have this information. However, it contains information on the sector to which each company belongs. The preprocessor of the company valuation algorithm automatically constructs an equally-weighted index from the stock prices of the companies in this sector *excluding* the company being valued.)
   Time constructed time series are daily.

2.3.3 Company-specific data

The time series used are:

1. Company sales.
   The time series is annual, semi-annual, quarterly and sometimes monthly.

2. Book value of equity.
   The time series is annual, semi-annual, or quarterly.

3. Dividend payments.
   The time series is semiannual.

4. R&D, (and whether it is capitalized or written off)
   The time series is semiannual.

5. EBIT (earnings before interest and taxes.)
   The time series is semiannual.

6. EBITDA (earnings before interest, taxes, depreciation and amortization.)
   The time series is semiannual.

7. Share price.
   The time series is daily.

2.4 Processing the input time series

A minimum of 18 time series is expected as input (see above) plus additional time series for whatever commodities are thought to affect the company valuation.

Step 1. Fill-in missing data
The set of time series is partitioned into subsets so that all time series in a given subset have the same period. We consider each subset in turn and any missing data in any of the time series in the subset is filled-in by applying the enhanced EM algorithm described earlier.

Step 2. Setting a common period
In order to use the time series as a single set, they have to be converted to a common period. We arbitrarily chose to convert all series to monthly observations using the interpolation and averaging procedures described earlier.

Step 3. Dimensionality reduction
We need to reduce the dimensionality of the problem so that the 20 or so time series are represented by a smaller number of independent components. We used the Principal and Independent Component procedures described in deliverable D3.3 to achieve this. By experimentation we found that 7 independent components combined in a non-linear manner using a Neural Network, were sufficient for the accuracy required. Although the required accuracy is clearly a subjective

judgement, it seems to be reasonable. For example, the RMS error in the reconstruction approximation of the 19 time series used for ICI was less than 4%. The Neural Network used for this purpose used two hidden layers. The input layer had 7 neurons corresponding to the 7 independent components. The output layer had 19 neurons corresponding to the number of input time series. The first hidden layer had 5 neurons and the second hidden layer had 3 neurons. There is no theory to give guidance for these numbers; rather they are the result of a large number of experiments to gain some insight into the size of Neural Network required. The Neural Network was trained by using the non-linear optimisation algorithm BFGS [Fletcher 1987] with an improved line-search technique developed in [Charalambous 1991].

2.5 Modelling the Independent Components

Deliverables D3.1 and D3.3 described various stochastic processes that can be used to model financial asset dynamics or the dynamics of independent components derived from financial data.

Step 4. Fitting stochastic dynamics
In our final algorithm we used just two models (see deliverable D3.3) namely GBM+Jumps+GARCH and MR+Jumps+GARCH to model the independent components derived in step 3 above. We used the calibration procedure in [Krkic and Christofides, 2003] and chosen the best model (of the two models above) depending on the value of the Akaike criterion.

2.6 Constructing a state-space graph

Deliverable D3.2 described how to construct an arbitrage-free state transition graph from the stochastic modelling equations. In this case we are constructing the graph to represent the evolution of independent components (rather than assets directly) and we need to ensure that the *tradable* assets do not allow any arbitrage to exist. This is done in an exactly analogous way as described in deliverable D3.2 namely checking each possible (statistically equivalent) evolution at each vertex for the existence of arbitrage and choosing an evolution which is arbitrage-free.

At this step we also need to consider the single-path predictions of the forecasts that are the results of the macro-economic model and the results of the data-mining model. Let us call the assets modelled by the above two methods 'exogenous'. At some time t in the future the state transition graph contains a large number of vertices (states) and at each one of these vertices there is a corresponding value for each asset (derived from the values of the independent components at that vertex). For an exogenous asset, the expected value at time t is given by the predictions of the macro-economic or data-mining model. If the mean of an exogenous asset at time t computed from the vertices $v_i \in V_t$ of the state space graph is different from that of the two external models, then the values of this asset at each of the vertices V_t is increased/decreased by the same amount δ to make the two means the same. Note that if the exogenous asset is tradeable, the above adjustment may re-introduce arbitrage. This is not a shortcoming, only an indication that an 'unreasonable' prediction cannot be accommodated. In any case, in our algorithm only non-tradeable (GDP growth from the macro-economic model and company sales from data-mining) assets are exogenous so arbitrage complications do not arise.

Step 5.Constructing a state-space graph
Construct the arbitrage-free state transition graph representation of the stochastic equations of the independent components. Modify the values of the exogenously defined variables so that their expected future values agree with the macro-economic/data-mining model forecasts.

2.7 Additional inputs: Discretionary expenditure

If we ignore the additional value resulting from exercising real options that may exist, no additional inputs are required from the user. (Note that algorithmic issues such as the number of time-steps used in the state transition graph, the maximum number of vertices in this graph, etc. are already determined by experimentation and set automatically without user input.) However, if the real options are considered (and in some cases - see computational results- they make a lot of difference) then the following data is also required regarding the effects of what, in deliverable D3.3, has been called *discretionary expenditure*.

In deliverable D3.3 we considered (and gave a Dynamic Programming recursion for) discretionary expenditure such as advertising, where for each additional unit of this expenditure there is a corresponding increase in sales and earnings from the next period onwards. In that case we need an expression for this relationship and we assume a linear relationship obtained from performing linear regression on past data of the company. The **slope** and **intercept** parameters then define this relationship.

An alternative type of discretionary expenditure is R&D. In this case R&D either succeeds in delivering a blockbuster product (as in the case of a major new drug for a pharmaceutical company), or fails to do so. In the latter case the company sales/earnings etc. remain the same at the existing vertices of the state transition graph (other than, of course, the money wasted on R&D which must be subtracted from the earnings). Success in R&D is represented by an additional arc (with a small transition probability) leading from the vertex where the R&D has succeeded to a corresponding vertex in a ”higher/parallel” graph where the company sales/profits are increased by some multiplier. Figure 1 illustrates the concept. The concept can be easily generalised so that success does not arise suddenly in a single step but is a result of a multistage process (as for example in the case of drug approval). In that case a third, fourth, etc. parallel graphs are defined with transitions only between a graph at layer ℓ and one at layer $\ell + 1$. Sales, earnings etc. can remain the same at vertices of all graphs other than the topmost graph that represent final R&D success.

In our case we only considered a single-stage R&D success/failure with a given budget that can be spent all in one go (or not at all) at any vertex of the graph. Again only two pieces of data need to be specified:
(1) The **probability of success**. (All other transition probabilities are adjusted downwards accordingly and equally.)
(2) The **multiplier factor** for company sales and earnings in case of success.

The values of the probability parameter can be estimated from examining large R&D projects in the sector the company belongs to. The effect of a successful

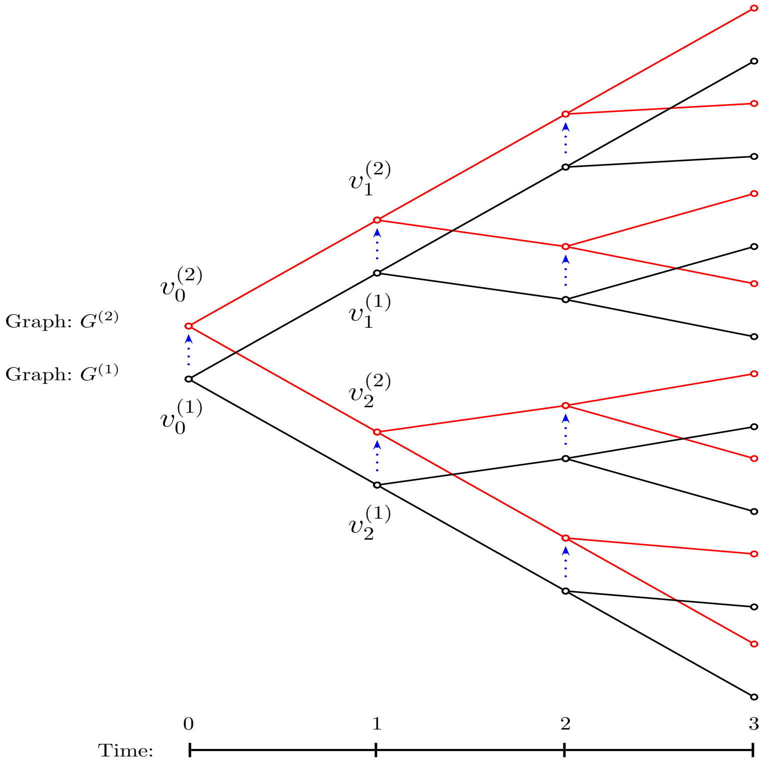

Figure 1: Effect of successful R&D expenditure

R&D project has to be estimated by regression. In deliverable D3.3 the DP recursion for valuation with discretionary expenditure of the ”advertising” type is given. A similar DP recursion applies for ”R&D” type expenditure.

2.8 Computing cash flows

Normal cash flows (with no discretionary expenditure) is computed as:

Step 6. Computing cash flows

At each vertex of the state transition graph we reconstruct from the independent components the company data at that state. We can then compute the earnings of the company at that vertex. Since the effects of R&D are fully accounted for by a possible increase in the cash flows, R&D cannot be capitalised as well; hence we have to subtract R&D expenditure from the book value if it has been capitalised.

Note that although only earnings are computed for the vertices of the graph, it does not mean that the rest of the company data is therefore irrelevant. Indeed this additional information may be vital. The independent components from which earnings are computed are affected by *all* time series mentioned earlier and are computed so as represent well all the company data. A completely different model would result if only company earnings were considered for the company data.

Increases due to discretionary expenditure are taken care of automatically by the DP recursion (See step 7.) using the two parameters mentioned in the previous section.

2.9 The DP recursion

The state transition graph with company cash flows occurring at every vertex and transition probabilities on every arc is now complete. The valuation recursion now needs to be solved. **Step 7: Solving the DP**
Solve the company valuation DP recursion as given in deliverable D3.3. The answer to this Dynamic Program is the value of the company.

3 The data structures used in the algorithm

The algorithm is written in C++ and makes extensive use of object orientation. It is a well-documented and maintainable code that runs equally well in Linux or windows. Its only user interface is the one provided by the EUROSIGNAL website. Simple data structures were used in all stages of the algorithm.

3.1 File structures

All files are stored as comma separated (sequential) files (csv files). This means that all files (from the large files containing daily time series to very small files containing, for example, the arc weights of the fully trained neural network) can be fully formatted, viewed and edited in Excel. Also, for the files containing time series, the first two rows are reserved for titles, so data start from the 3rd row. For these files the first column contains the date (in any format) of the observation. In the interest of consistency, even for the files that do not contain time series, the first two rows and first column are left blank.

Data read from the database are immediately written onto file in the csv format mentioned above. This means that the algorithm can also be executed in a stand-alone batch mode if supplied with the required time series in an Excel worksheet.

A large number of intermediate files are generated by default. These include files containing the time series with a common period and filled missing values, the Principal and independent components time series, the trained neural network, the calibrated parameters of the chosen stochastic processes, the full state transition graph, and the final answers including some sensitivity results that are obtained with no additional computational effort during the solution of the DP recursion.

3.2 The state transition graph structure

The data structure used for the state transition graph is the ”forward star” structure. This is composed of three lists.

The first list is an arc list, as long as the number of arcs. It groups together all arcs emanating out of vertex 1, followed by all the arcs emanating out of vertex 2, etc. It stores in each arc position the number of the vertex which is the terminal vertex of that arc.

The second list is also an arc list. A property of the arc (for example its transition probability) is held in the corresponding arc position.

The third list is a vertex list, as long as the number of vertices. In a vertex position it stores a pointer pointing to the arc position of the first arc emanating from that vertex.

For example, if the pointer of vertex i points to arc position A and the pointer of vertex $i+1$ points to arc position B, then all arcs $A, A+1, A+2, \ldots, B-1$ emanate from vertex i. If the entry in the first and second arc lists for arc A, say, are j, and q respectively, then arc A is the arc (i, j) and its transition probability is q.

The forward star is a simple graph data structure that is suitable only for problems where the graph will be constructed and will not be modified thereafter. This is the case with our existing algorithm.

3.3 The DP tables

The DP tables for the recursive function $f_i(w, a)$ can be implemented as 3-dimensional tables. This, however, is very wasteful in storage since at any one time only the vertices $v_j \in V_j^{out}$ are needed to compute $f_i(w, a)$. Thus the values $f_i(w, a)$ need to be held for every (w, a) not for all vertices, but only for vertices at one time period (actually slightly more).

4 Computational results

We computed the value of 10 companies taken from the FTSE100 list. Before performing each valuation we estimated the two parameters required for the discretionary expenditure. We did not expend sufficient effort to get good values for these estimates which may partly explain the ”errors” between our computed company values and the market values.

The results are shown in table 1. In almost every case our company valuation is well below the market value of these companies.

The computing times shown in the table are hours:minutes:seconds on a 3.2 Ghz Intel Pentium 4 processor running windows XP and using the Visual C++ Version 7.0 compiler.

Company	Computed value		Computing time		Market value
	excl. RO	incl. RO	excl. RO	incl. RO	
BAE Sys	133.5	133.5	1:06:33	3:11:09	196.25
Barclays	384.1	390.2	0:54:04	5:43:25	491.00
HSBC	646.0	660.6	0:58:53	5:28:22	794.00
ICI	174.4	201.0	1:10:45	6:08:02	199.75
Unilever	382.2	382.2	0:55:13	2:55:23	521.00
Reuters	270.7	291.8	1:03:09	4:42:28	340.75
BP	354.8	431.3	0:49:44	6:28:12	487.25
AstraZen	1470.9	1921.4	0:57:36	4:55:43	2626.00
mmO2	48.6	59.3	1:10:44	4:40:19	92.50
Vodafone	112.8	145.6	1:00:39	6:26:51	135.25

Table 1: Computational results from the valuation of ten companies from the FTSE100 on 11 May 2004.

References

[Akaike 1973] Akaike H (1973). *Information theory and an extension of the maximum likelihood principle*. In Petrov B, Csaki F (Eds), Second international symposium on information theory. Academia Kiado, Budapest, Hungary, pp 267-281.

[Black and Scholes 1973] Black F. and Scholes M. (1973). *The pricing of options and corporate liabilities*. J. of Political Economy, vol. 81, pp 637-659.

[Bollerslev 1986] Bollerslev T. (1986). *Generalised autoregressive conditional heteroskedasticity*. J. of Econometrics, vol. 31, pp 307-327.

[Campbell, Lo and MacKinlay 1997] Campbell JY, Lo AW and MacKinlay AC (1997). *The econometrics of financial markets*. Princeton University Press, Princeton, USA.

[Charalambous 1991] Charalambous C (1991). *A conjugate gradient algorithm for the efficient training of neural networks*. IEE Proc., Part G, 1991.

[Christodoulou 2002] Christodoulou M (2002). *Algorithms for risk management*. PhD Thesis, London University (Imperial College), London, UK.

[Christofides, Christofides and Christofides 2003] Christofides S., Christofides A. and Christofides N. (2003). *Graph representations for financial dynamics*. Report, Centre for Quantitative Finance, Imperial College, UK.

[Cont 2000] Cont R. (2000). *Empirical properties of asset returns*. Quantitative Finance, vol. 1, pp 223-236.

[Cox, Ingersoll and Ross 1985-b] Cox J.C. Ingersoll J.E. and Ross S.A. (1985-b). *An intertemporal general equilibrium model of asset prices*. Econometrica, vol. 53, pp363-384.

[Cox, Ingersoll and Ross 1985-a] Cox J.C. Ingersoll J.E. and Ross S.A. (1985-a). *A theory of the term structure of interest rates*. Econometrica, vol. 53, pp 385-407.

[Duffie 1996] Duffie (1996). *Dynamic asset pricing theory*. Princeton University Press, second ed., Princeton, USA.

[Erb et al. 1994] Erb B., Campbell H.R. and Tadas V.E. (1994). *Forecasting international equity correlations*. Financial Analysts Journal, Nov-Dec, pp 32-44.

[Fletcher 1987] Fletcher R (1987). *Practical methods of optimisation - 2nd ed.*. Wiley, New York, USA.

[Georgikopoulos 2004] Georgikopoulos N (2004). *Extensions of the EM algorithm*. MPhil Thesis, London University (Imperial College), London, UK.

[Kon 1984] Kon S.J. (1984). *Models of stock returns: A comparison*. J. of Finance, vol. XXXIX, pp 147-165.

[Krkic and Christofides, 2003] Krkic M. and Christofides N. (2003), *The calibration of stochastic models*, Research Report CQF-03-12, Centre for Quantitative Finance, Imperial College, London UK.

[LeBaron 2001] LeBaron B. (2001). *A builder's guide to agent-based financial markets*. Quantitative Finance, vol. 1, pp254-262.

[Nelson 1990] Nelson D B (1990). *ARCH models as diffusion approximations*. J. of Econometrics, vol. 45, pp 7-38.

[Press et al. 2002] Press HT, Teukolsky SA, Vetterling WT and Flannery BP (2002). *Numerical recipes in C++*. Cambridge University Press, second ed., Cambridge, UK.

[Vasicek 1977] Vasicek O.A. (1977). *An equilibrium characterization of the term structure* Journal of Financial Economics, vol. 5, pp 177-188.