Proceedings of the 1993 Winter Simulation Conference
G.W. Evans, M. Mollaghasemi, E.C. Russell, W.E. Biles (eds.)

DISCRETE-EVENT SIMULATION FOR CORPORATE FINANCIAL PLANNING

Ingolf Ståhl

Stockholm School of Economics
Box 6501
S-113 83 Stockholm, Sweden

ABSTRACT

We note that discrete-event simulation is of importance for financial planning because it can at the same time handle uncertainty, the stochastic aspect, and allow us to follow a development completely over time, the dynamic aspect. It is probably of special importance for small and medium size firms in engineering and construction. We conclude that such stochastic and dynamic simulation cannot be done in a spreadsheet, but that a special language of discrete-event simulation, like GPSS, is the most suitable way of doing this. We give examples of how this can affect the teaching of finance at business schools. Finally we present a very simple program that illustrates how cash flow forecasting can be done using micro-GPSS, an easy-to-learn version of GPSS.

1. STOCHASTIC DYNAMIC SIMULATION FOR FINANCIAL PLANNING

The basic theme of this paper is that discrete-event simulation, or stochastic dynamic simulation as it can also be called, is of great importance for corporate financial planning. We are here interested both in the stochastic aspects, allowing us to deal with uncertainty, and the dynamic aspects, allowing us to follow a development over time. I shall below discuss these two aspects separately.

1.1 Stochastic simulation for financial planning

It must be stressed with great emphasis that uncertainty is a critical factor in corporate financial planning. We can just think of what answers are reasonable for the following types of critical questions:

How much will we sell next year? Is the answer: 100,000 units for certain or 80,000 - 120,000 units?

When will this customer pay? Is the answer: Within 30 days for certain or with 80 per cent probability within 40 days and with 10 per cent probability after two months?

How many DM will buy a dollar a year from now? Is the answer: 1.70 for certain or between 1.50 and 2.00?

I think it is clear that in all cases the last type of answer, indicating great uncertainty, is the more reasonable one. In fact, if all future payments of companies could be forecast with absolute certainty, all corporate debt would be as safe as government bonds. Then there would not be need for different types of financial instruments, such as e.g. convertibles, options etc., and there would in fact be no need for financial theory. Hence it is absolutely clear that uncertainty is at the core of finance.

Against this back-ground, it is indeed very strange that most simulations of the future financial position of a corporation, e.g. cash forecasts, are done using deterministic simulation, i.e. without any uncertainty, by the use of spreadsheets. Instead most financial simulation **should** be stochastic.

1.2 Dynamic simulation for financial planning

With the word dynamic simulation, set in contrast to static simulation, I imply that any point of time can, if suitable, be incorporated into the simulation, while in a static simulation only a very limited number of points of time can be taken into account. The difference can best be explained by an example. Let us look at the spreadsheet below.

Table 1: Example of cash flow forecast

Results and Cash Forecast

Month	1	2	3	4	5	6
Sales	150.00	175.00	200.00	175.00	125.00	200.00
Costs	90.00	105.00	120.00	105.00	75.00	120.00
Profits	60.00	70.00	80.00	70.00	50.00	80.00
Cash	35.00	5.00	10.00	155.00	255.00	260.00

This spreadsheet has been produced by a simple micro-GPSS simulation program for cash-forecasting, presented in section 5 below and based on a program in Ståhl (1990). The program deals with a corporation, an importer, that buys a certain machine from abroad and then resells it. It pays the foreign producer cash directly for each unit, but provides its own customers with credit. Orders arrive according to an exponential distribution, while customers' payment times vary according to an Erlang distribution.

Looking only at the end-of-the-month figures, it appears that there would be enough cash for the corporation so that there would not be any acute liquidity problems. We can even make a simple graph of the cash forecast, giving the same impression. We can call this graph static since it is only covers a small number of pre-defined time points.

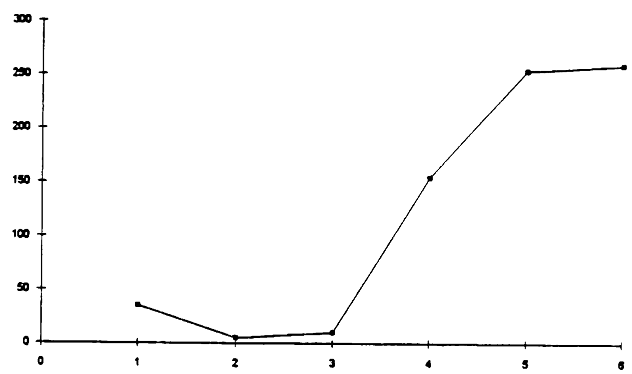

Figure 1: Static cash flow graph based on table 1

However, the GPSS program can also produce a graph of the following dynamic type that gives us a day-by-day forecast of how cash develops:

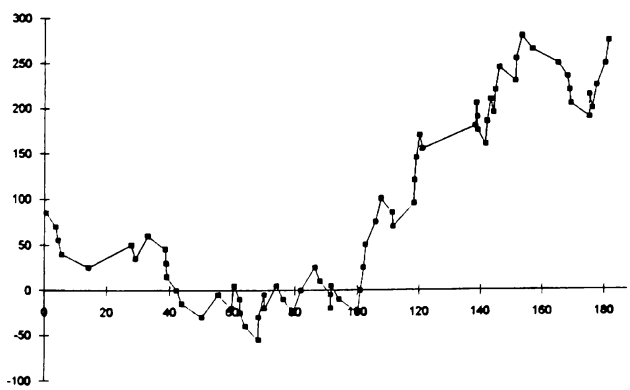

Figure 2: Dynamic cash flow graph

We see that the two graphs give completely different impressions. The great financial problems with negative cash a great many times in the future are clearly seen in figure 2. In figure 1 these problems are not perceived at all since it by chance happens that there is a cash surplus at each time-point that we regard as the end of the month. This clearly illustrates the need for a dynamic, discrete-events approach to corporate financial planning.

2. FOR WHICH CORPORATIONS IS DISCRETE-EVENT FINANCIAL SIMULATION IMPORTANT?

The discussion above gives an indication that in certain situations the use of static and/or deterministic simulation instead of discrete-event simulation can be very misleading. The question is then if anything can be said about for which cases this type of misconception is especially serious. Basing myself on some empirical data from Sweden and simulations based on this data, I put forward the hypothesis that the need for dynamic stochastic simulation for cash flow forecasts and similar types of financial planning is especially large, when the payment structure of the firm is such that a couple of hundred large payments constitute more than half of the payment flow of the company. In the case that payment times vary much, e.g. as in the example presented in section 5, there could be a significant, e.g. 25 per cent, risk that total inflow of cash during a certain month would be less than half of that expected, i.e. of what it would have been under certainty. This could for many corporations lead to financial troubles. Corporations for which 200 - 250 payments constitute more than half of the total payments are probably mostly found among small and middle size corporations in areas such as mechanical engineering and construction. For larger corporations, or for corporations with lots of small payments, like in retailing, the effects of stochastic variations in payments will balance each other and there is less of a need to follow individual payments day by day.

3. PROBLEMS WITH STOCHASTIC AND DYNAMIC SIMULATION IN A SPREADSHEET

We thus perceive the need for stochastic and dynamic simulation for corporate financial planning. The question then arises why this has to be done in a language for discrete-event simulation and not in a spreadsheet, which business people are used to. It should be stressed that the discussion below refers to ordinary spreadsheet-packages, like Excel and Lotus 1-2-3. Some of the problems mentioned below, but not all, can be remedied by the use of add-on packages like @ Risk from Palisade Corp..

3.1 Stochastic simulation in a spreadsheet

First of all it should be noted that it is difficult to carry out experiments of stochastic simulation in a correct way in a spreadsheet. For a correct experimental design it is important to keep as much as possible under control so that the differences in the results between runs on two different decision variables as far as possible are due to differences in the decision variables. If we e.g. want to compare two different order quantities, it is important that the effect on cash flow does not depend on the receipt of differently many orders. We want the same order arrival pattern to occur in the simulation regardless of whether we buy 10 units or 20 units each time. In order assure this, we want to use the same random number stream for order arrivals in both cases. Since we are likely to have also other random variables, e.g. for delivery times, we cannot allow these other random variables to take random numbers from the same stream that we use for order arrivals. Hence we need several random number streams for correct stochastic simulation experiments. The most well-known spreadsheet packages, like Excel, have, however, only one single random number stream. Furthermore, in most major spreadsheet packages it is very difficult even to replicate the same random number stream (Schellhaas, 1993).

Among other limitations as regards the use of spreadsheets for stochastic simulation we note that in order to generate a sequence of random numbers one usually needs to use a macro. Furthermore, in contrast to modern simulation languages, there are generally no built in statistical distributions, such as the normal and the exponential distribution, from which one can draw samples. One has to write a macro to do this. Likewise it is, in comparison with many simulation languages, difficult to sample from a distribution given by a set of empirical data.

3.2 Dynamic simulation in a spreadsheet

We next turn to the case when we have dynamic, but deterministic simulation, e.g. when payments always come after a fixed number of days, say 30. The problem is here only that of the number of columns needed. For the realistic case that we want to follow payments day by day, but not hour by hour, we would in order to follow cash flows over a year require 365 columns. This is much, but feasible, though we loose in terms of overview.

3.3 Stochastic AND dynamic simulation in a spreadsheet

The big problem for a spreadsheet, which is fundamental and can not be remedied by any add-on package, arises, however, when we want to do both stochastic **and** dynamic simulation. The root of the problem is that while a language like GPSS is "forward directed", the spreadsheet is "backward directed". In GPSS we can, e.g. by the block ADVANCE FN$RLNG3*45, schedule the payment of a sales transaction made on e.g. January 3 to come after a sampled time of e.g. 57 days, i.e. on March 1. In a spreadsheet we have to write in the cell, denoting the day of repayment, from which **earlier** day it shall take the sales transaction that on this later day leads to a payment. This is virtually impossible to do in a logically correct way when payment time is a stochastic variable.

Let us in order to prove this point first look at a very simple example. We assume that payments come after either one or two months. We locate all sales and payments to the first of each month. Let p_j be a random value, being either 1 or 0, determining whether or not the payment at the start of month j refers to sales 2 months earlier. We would now in the cell of March 1 have p_3*sales(Jan. 1)+(1-p_3)*sales(Feb. 1) and in the cell of April 1 p_4*sales(Feb. 1) + (1-p_4)*sales(March.1). The logical problem is that we might very well sample out $p_3=0$ and $p_4=1$. This would imply that the sales of Feb. 1 would be paid for on both March 1 and April 1, i.e. an illogical double payment. We could also sample $p_3=1$ and $p_4=0$, implying that the sales on Feb.1 would never lead to a payment, since March 1 would point to Jan 1. and April 1 to March 1.

This problems is due to the fact that we, by having to write the formula in the cell of the spreadsheet, restrict ourselves to using **constants** when referring to the cells in the matrix of the spreadsheet. In the example with March 1 above, we do the same thing as writing in a GPL: M(pRow,3):=p_3*M(sRow,1)+(1-p_3)*M(sRow,2), where M is the spreadsheet matrix, pRow the row of payments and sRow the row of sales. We would like to be able to write M(pRow, RandomColumn):= expression, but this is not possible, since we have to put the formula in a specific cell with a given column.

How shall this problem be handled? Of course one can by going into complicated macros, different for different spreadsheets, do as in a GPL. We can only notice that in the spreadsheets we have looked at this is quite cumbersome. It appears to be a lot easier for the person, who has worked in spreadsheets without learning the use of macros, to learn enough micro-GPSS to do the simulation the discrete-event way than to learn macros and do it the macro-way.

If one would be restricted to spreadsheets without use of macros, one could possibly attempt to do the combined dynamic-stochastic simulation the following way: We restrict the analysis to the simple case that all sales on one day would be paid back on the same later day. One could then for each day, when a sale takes place, sample the number of days of credit and, on the basis of this day of sales, calculate the day of payment and put this in a cell. For each possible day of payment

Table 2: Part of spreadsheet attempting dynamic stochastic simulation

	A	B	C	D	E
1	Day	1	2	3	4
2	Sales	=60+RAND()*40	=60+RAND()*40	=60+RAND()*40	=60+RAND()*40
3	Payment	=SUM(B7:B36)	=SUM(C7:C36)	=SUM(D7:D36)	=SUM(E7:E36)
4					
5	Credit time	=1+INT(RAND()*30)	=1+INT(RAND()*30)	=1+INT(RAND()*30)	=1+INT(RAND()*30)
6	Payment day	=B1+B5	=C1+C5	=D1+D5	=E1+E5
7	Pot. paym 1		=IF(B6=C1,B2,0)	=IF(B6=D1,B2,0)	=IF(B6=E1,B2,0)
8	Pot. paym 2			=IF(C6=D1,C2,0)	=IF(C6=E1,C2,0)
9	Pot. paym 3				=IF(C6=E1,D2,0)

y, one would then have a list of potential payments, such that if the day of payment of the sales made on day x is this day y, then the payment is the sales amount on day x, else it is 0. The actual payment of the day is then the sum of all such potential payments. For the case of March 1 in the simple example above, we would have two such potential payments:

1. IF (Payment day of Jan 1 = March 1 THEN sales(Jan 1) ELSE 0) and
2. IF (Payment day of Feb 1 = March 1 THEN sales(Feb 1) ELSE 0).

In this way there is logical consistency, since the system becomes forward directed. We will however get an enormous problem of matrix size, even for the simple case when all sales on a specific day are supposed to be paid on the same future day. In order to illustrate the problem, we study in table 2 above a simple example of an attempt to do dynamic stochastic simulation in EXCEL. We here assume that credit time can take any integer value between 1 and 30 days, i.e. some pay already the next day, some after two days and some after a month. The columns represent days as seen in the row 1. We assume that the company starts its activities on day 1. In row 2 we sample a sales amount for each day, namely a value varying between 60 and 100. In row 3 we sum up the 30 potential payments given in rows 7 - 36 of the same column. On day 1 there can not be any payment, but on day 2 there can be a payment, if the payment day of the sales on day 1 (B6) is day 2 (C1). On day 3 there are two potential payments, the sales on days 1 and 2. Already for this extremely simple example we have quite a complicated matrix, which we would have seen if we had included all 30 days.

The problem of matrix size becomes even more acute if we assume that payments can be delayed for possibly up to a year. If we then want to follow payments day-by-day, simulation in a spreadsheet would require at least a 365 x 365 matrix just for this cash forecast. One can hence safely say that dynamic stochastic simulation is not possible in a spreadsheet. With macros as cumbersome to learn and use as a GPL in this case, an easy-to-learn discrete-events simulation language appears to be the best alternative, as will be exemplified further in section 5.

4. IMPLICATIONS FOR BUSINESS SCHOOLS

The ideas presented above has had implications on how corporate financial planning is taught in Sweden, both at the Stockholm School of Economics (SSE) and Stockholm University (SU). At the SSE, GPSS is taught in a course on Corporate financial planning. This is a third year course and it is taken by around 250 students of the 300 students that enter the SSE annually. The GPSS part of this course is 12 hours and examination is done in the form of a student project, roughly 100 blocks in size, regarding some kind of cash flow forecast and involving decisions on purchasing, inventory and borrowing policy. This project can be regarded as a more advanced version of the program in section 5. Some of these student project programs are presented in Ståhl (1993b). It should in this connection be mentioned that we with this course do not only cater to students who are likely to work in the medium size companies in engineering and construction mentioned in section 2, but also to students who in some kind of consulting capacity, e.g. as bankers or CPAs, will deal with such corporations.

At the department of Business Administration of Stockholm University, which is quite separate from the SSE, micro-GPSS is also used in a course in Financial planning. This part of the course is similar to the mentioned course given at SSE, but differs in that it spends only 8 hours on micro-GPSS and has 40 students. The course ends with the study of programs similar to that in section 5.

5. A micro-GPSS programs for cash flow forecasts

I shall finally present a small program for cash flow forecasting, written in micro-GPSS. It deals with the simple situation mentioned in section 1.2. Micro-GPSS is a stream-lined version of GPSS, aimed at being very easy to learn, use and understand. The whole language is presented in Ståhl (1990) and the pedagogical ideas behind it in Ståhl (1993). By presenting this program, I hope to give some substance to my claim that discrete-event simulation is a suitable method for this type of problems.

```
    simulate
    let       x$price=25 ! Machines are sold for 25
    let       x$cost=15  ! Machines are bought for 15
    let       x$cash=100 ! Initial cash 100
sales function  x$price,c5
10,310/20,112/30,60/40,40/50,30
* Customer segment
    generate  fn$xpdis*365/fn$sales
* An order comes on average every 365/fn$sales day
* following an exponential distribution
    let+      x$sal,x$price
* Total sales increase by price of sold unit
    let+      x$tcost,x$cost
* Total costs increase by price of bought unit
    let-      x$cash,x$cost
* Cash decreases by cost = price of bought unit
    help      fprin,c1,x$cash ! Measure cash for graph
    advance   45*fn$rlng3
* Customer pay after 45 days on average, but payment
* time sampled from Erlang  with 3 drawings
    let+      x$cash,x$price
* Cash is increased by price when customer pays
    help      fprin,c1,x$cash ! Measure cash for graph
    terminate
* Report segment
    generate  365/12
*  A report is produced at end of each month
    let+      x$num,1    ! Increase number of month
    let       x$prof=x$sal-x$tcost ! Profits=sales-costs
* Write data into spreadsheet matrix
    help      decimal,1
    help      matin,1,x$num,x$num ! Month
    help      matin,2,x$num,x$sal ! Total sales
    help      matin,3,x$num,x$tcost ! Total costs
    help      matin,4,x$num,x$prof ! Profits
    help      matin,5,x$num,x$cash ! Cash
    let       x$sal=0    ! Set sales of new month to 0
    let       x$tcost=0
* Set total costs of new month to 0
    terminate 1
    start     6,np ! Terminate after 6 months
    end
```

Micro-GPSS code of simple cash flow program

The comments (after ! or the initial *) makes the program virtually self-documenting, but some external comments might still be helpful. We start the program by determining the value of the sales price and the cost of purchasing a unit from the manufacturer. We also set the initial cash of the importer. These are three values that one might want to experiment with. We also establish a demand function, where annual sales are defined by five pairs of data as a function of price. Thus if price is e.g. 30 then 60 units will be sold.

In the customer segment, the block GENERATE FN$XPDIS*365/FN$SALES generates each specific order. FN$SALES is annual sales in units and 365/FN$SALES is hence the average distance in time between two orders. (If we e.g. sell 36 units annually there would be an average of 10 days between each order.) The actual time between two orders is sampled from an exponential distribution by multiplying this average value by FN$XPDIS, an exponential distribution with the average 1. When a new order arrives we increase total revenues by the sales price of the sold unit and total cost of goods sold by the unit cost = purchasing price of the unit. Since the importer pays the manufacturer in cash, his cash is decreased by the unit cost. We then put the current values of time (C1) and of cash into a file to be used for the later automatic generation of a graph. (The HELP FPRIN command refers to a subroutine already built into micro-GPSS.)

The next block ADVANCE 45*FN$RLNG3 represents the actual credit time. FN$RLNG3 samples from the Erlang distribution, giving the average of three independent samples from the negative exponential distribution with the average 1. (The Erlang distribution is a distribution available in micro-GPSS, but not in other GPSS versions.) The average credit time is 45 days, but some buyers pay within a shorter time and some pay considerably later. The Erlang distribution with parameters 2, 3 or 4 appears to provide realistic approximations of payment behavior, at least as regards customers of some medium size Swedish firms.

After the expiration of this credit time, the customer will pay for the product, increasing the importer's cash by the price of the product. We again record the time and value of cash for the graph. With TERMINATE the sales transaction it then finished.

We next turn to the report segment. Here we start each report by updating the number of the month. (X$NUM is automatically set to 0 at the start of the simulation.) We next calculate the profit of the month as total sales - total costs. In the next lines we write the current value of the month, total sales, total costs, profits and cash into the cells, with rows 1 - 5 respectively and with the column of the month, of an internal matrix. This matrix is then at the end of the simulation automatically written into a file. The line HELP DECIMAL 1 insures that the numbers of the months on line 1 are printed without decimals. If we have called the program e.g. CASHSIM, this automatically produced file will have the name CASHSIM.TXT. This file will then contain table 1 above, provided we have also supplied a "frame file" CASHSIM.FRM with text as follows:

Table 3: Frame for text in the output file

```
*       Results and Cash Forecast

Month

Sales
Costs
-----
Profits
=====
Cash
```

Finally in the report segment, we set total sales and total costs to 0, since we want these values only to refer to the present month. START 6 implies that we end the simulation after six months, without standard GPSS output because of the NoPrint. END terminates the GPSS program.

6. FINAL COMMENTS

Judging from the great many bankruptcies in the last couple of years in Northern Europe it appears that cash flow forecasting models of the type presented above have strong relevance for a great number of smaller and medium size corporations, e.g. in the engineering industry. Models that allow for dealing with uncertainty and for following development closely over time are needed. At the same time it is important that the models are simple to learn and use. In this respect micro-GPSS, which allows the user to write useful models already after one day (Ståhl 1992), appears to provide the best alternative.

REFERENCES

Schellhaas, K.-U. 1993. Problems and prospects of spreadsheet-based simulation models in business. In *Modelling and Simulation 1993*, ed. A. Pave. Lyon: SCS.

Ståhl, I. 1990. *Introduction to Simulation with GPSS: On the PC, Macintosh and VAX.* Hemel Hempstead, UK: Prentice Hall.

Ståhl, I. 1992. *Simulation in One Day with micro-GPSS - a Short Tutorial* (incl. diskette with demo-version of micro-GPSS and 32 program examples and available free of charge from the author). Stockholm: SSE.

Ståhl, I. 1993. Principles behind the Design of an Easy-to-learn Simulation Language. In *Simulation Applications in Business, Management and MIS,* ed. Roberts, R.S. and S. Monroe. San Diego: SCS.

Ståhl, I. 1993b. *Some micro-GPSS programs for corporate financial planning.* EFI Research Paper, Stockholm.

AUTHOR BIOGRAPHY

INGOLF STÅHL is Professor at the Stockholm School of Economics, Stockholm, and has a chair in Computer Based Applications of Economic Theory. He was visiting Professor, Hofstra University, N.Y., 1983-1985 and leader of research project on inter-active simulation at the International Institute for Applied Systems Analysis, Vienna, 1989-1982.. He has taught GPSS for over fifteen years at universities and colleges in Sweden and the USA: He has on the basis of this experience led the development of the micro-GPSS system. He is also consultant in simulation to Swedish banks and industry.