JIA 116 (1989) 663-672

USING APL FOR FINANCIAL CALCULATIONS

By D. O. Forfar, B.A., F.F.A., A.I.A., F.I.M.A., F.S.S.

[Presented at the Seminar, 'Applications of Mathematics in Insurance, Finance and Actuarial Work', sponsored by the Institute of Mathematics and Its Applications, the Institute of Actuaries, and the Faculty of Actuaries, held at the Institute of Actuaries, 6–7 July 1989.]

'L'APL est vraiment le langage qui convient à l'actuaire'
(L. Moreau, Paper to the International Congress of Actuaries, Sydney 1984)[6]

1. INTRODUCTION

The computer language APL is not as well known in the financial field as it might be and yet its structure and syntax make it very well suited to financial and actuarial calculations (as well as to mathematical and statistical calculations)—indeed it might almost be called a natural language for financial calculations.

In a contribution to the XVI ASTIN Colloquium in Liège, 1982, entitled 'Description and Comparison of various Programming Languages and their Suitability for Non-Life Actuaries', J. A. Bartola[2] concluded that APL was a very attractive programming language for actuaries. The same conclusion was reached by G. Leuven in a study of different programming languages undertaken by the Dutch Actuarial Society. A book entitled *APL, The Language and its Actuarial Applications*, by D. Stiers, M. J. Goovaerts and J. de Kerf[9] was published in 1987. This book contains a wealth of examples of applications of APL to problems in finance, general insurance, credibility theory, etc.

The special APL symbols combined with the vector, matrix and multidimensional array structure of the language make for a very powerful language which enables short programs to accomplish what in other programming languages would take much more extensive coding involving many loops. Programs (called functions in APL) can call other programs and a complex system can be built up in a modular way from simpler building blocks.

The language is an interpreted language (although compiled versions are available) and highly interactive. A great virtue of the language is that it enables quick development of short but powerful programs. Coding can be tested as work progresses since APL operates in two modes—'calculator' mode where a line of code can immediately be executed and tested with the immediacy of a pocket calculator (e.g. before insertion in the program) and 'definition' mode where the program statements are stored until the program as a whole is executed.

2. ANNUITY VALUES

An advantage of APL functions is that they can be made to look similar to standard financial symbols. For example, the standard function $a_{\overline{n}|i}$ can be evaluated by an APL function called AN where the syntax to evaluate $a_{\overline{25}|\cdot 10}$ would be AN 25 · 10. The economy of the language can be seen from the coding of the function AN which takes only one line.

```
            ∇Z←AN V
[1]         Z←+/(÷1+V[2])*⍳V[1]
            ∇
```

The symbol ιN represents the vector $(1, 2, \ldots, N)$ and we have:

$$X*\ \iota N=(X, X^2, \ldots, X^N).$$

It is easily seen (since APL evaluates expressions from right to left), that

$$(\div 1+X)*\iota N=((1+X)^{-1}, (1+X)^{-2}, \ldots, (1+X)^{-N}).$$

If A represents the vector $(a_1, a_2, \ldots, a_n)$ then

$$+/A=\sum_{i=1}^{N} a_i.$$

In the above example V is the vector (N, i) with $V[1]=N$, the term, and $V[2]=i$, the rate of interest.

Hence:

$$+/(\div 1+V[2])*\iota V[1]=\sum_{s=1}^{N}(1+i)^{-s}=a_{\overline{n}|i}.$$

It is easily seen that $Ia_{\overline{n}|i}$ can be written as

$$+/(\iota N)\times(\div 1+V[2])*\iota V[1].$$

The function can easily be generalized to calculate the value of an annuity where the rate of interest (assumed to be a one-year deposit rate) varies each year. We suppose that the vector I is equal to $(i_1, i_2, \ldots, i_N)$ and represents the rate of interest each year for N years. The value of an annuity on the basis of the above interest rates can be evaluated using additional APL symbols. If A represents $(a_1, a_2, \ldots, a_N)$ we have

$$+\backslash A=(a_1, a_1+a_2, \ldots, \sum_{i=1}^{N} a_i)$$

$$\times\backslash A=(a_1, a_1a_2, \ldots, \prod_{i=1}^{N} a_i).$$

The value of the annuity is

$$a_{\overline{n}|}=\sum_{r=1}^{N}\prod_{s=1}^{r}(1+i_s)^{-1}$$

or in APL

$$+/\times\backslash\div(1+I)$$

If the level of the annuity varies and the vector A represents the level of the annuity each year then the value of the special annuity is

$$+/A\times\ \times\backslash\div(1+I)$$

The above examples illustrate that the natural syntax of APL lends itself to financial problems.

The economy of the language can be seen by comparing the one-line APL program for the evaluation of $a_{\overline{n}|i}$, which does not contain any loops, with a BASIC program, which might look as follows:

```
10 V      =1/(1+1)
20 AN     =0
30 P      =1
40 FOR T  =1 TO N
50 P      =P*V
60 AN     =AN+P
70 NEXT T
80 PRINT AN
```

The evaluation of an annuity payable m-thly can be easily done in APL, for example the annuity $a^{(m)}_{\overline{n}|i}$ can be evaluated by the function ANM with syntax ANM n i m, e.g. $a^{(12)}_{\overline{25}|\cdot 10}$ would be expressed ANM 25 ·10 12.

```
        ∇Z←ANM V
[1]     Z←+/(÷V[3])×(÷(1+V[2])*÷V[3])*⍳V[3]×V[1]
        ∇
```

3. RANDOM FLUCTUATIONS IN INTEREST RATES

Using the symbols already described and noting that when V is a vector ρV returns the number of elements in V, a function to evaluate the mean and standard error of the elements of a vector (assumed to have two or more elements) is as follows:

```
        ∇Z←STATS V
[1]     Z←(+/V)÷⍴V
[2]     Z←Z,((+/(V-Z)*2)÷¯1+⍴V)*0.5
        ∇
```

We can show how a very short program in APL solves the following example involving random fluctuations in interest rates.

Example: The one-year deposit interest rate in future years is assumed to follow an autoregressive process of order one, namely

$$i_r = \cdot 08 + \cdot 6(i_{r-1} - \cdot 08) + \cdot 02Z$$

where Z is a standard normal variable, $N(0, 1)$.

Determine the distribution of the value of a ten-year annuity payable yearly in arrear, if the current deposit rate, i_1 is ·13 (13%).

We generate a sequence $\{i_r\}_{r=1}^{10}$ from the above equation and evaluate the annuity

$$a_{\overline{10|}} = \sum_{r=1}^{10} \prod_{s=1}^{r} (1 + i_s)^{-1}.$$

We might repeat this process say 500 times to generate a sequence of annuity values $\{a_{\overline{10|}}^t\}_{t=1}^{500}$, where t is the t^{th} simulation. The mean and standard errors of these 500 values can then be calculated.

The solution is facilitated by a function which generates a vector of values of N independent standard normal variables. The APL roll function ? is used (? N returns a random number between 1 and N) as well as the result that if $\{Y_i\}_{i=1}^{12}$ are independent random variables uniformly distributed over [0,1] then

$$Z = \sum_{i=1}^{12} (Y_i - \cdot 5)$$

is approximately normally distributed as $N(0,1)$. The symbol ρ is again used where, for example $(N, M)\,\rho A$ reshapes the vector A into a matrix of size N by M so that for example (3,2) ρA represents the matrix:

$$\begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \\ a_5 & a_6 \end{bmatrix}$$

The function STDNORM generates a vector of N independent standard normal variables,

```
      ∇STDNORM[⎕]∇
      ∇Z←STDNORM N
[1]   Z←+/¯0.5+(N,12)ρ(÷1000)×(¯1+?(12×N)ρ1001)
      ∇
```

The function to simulate the annuity values is SIMAN, with syntax SIMAN n, where n represents the term of the annuity. The problem is solved by SIMAN 10.

```
      ∇Z←SIMAN T;R;I
[1]   R←(500,1)⍴0.13
[2]   I←1
[3]  ST:R←R,[2]0.08+(0.6×(R[;I]-0.08))+0.02×STDNORM 500
[4]   →(T>I←I+1)/ST
[5]   STATS(+/×\÷1+R)
      ∇
```

Even although 5,000 individual interest rates have to be simulated the program only loops 10 times, 10 being the term of the annuity.

4. INTRODUCTION OF MORTALITY RATES

We have not yet introduced mortality into the functions considered but this can easily be done. We suppose that

$$Q = (q_0, q_1, \ldots, q_{W-1})$$

is a vector representing mortality rates from age 0 to age $(W-1)$ where W is the limiting age of the table.

We have that

$$\begin{aligned} M\uparrow A &= (a_1, a_2, \ldots, a_M) \qquad (M \text{ positive } \leqslant N) \\ &= (a_{N-M+1}, \ldots, a_N) \qquad (M \text{ negative } \geqslant -N) \end{aligned}$$

Hence if X is the age of the life assured

$$N\uparrow(X-W)\uparrow Q = (q_x, q_{x+1}, \ldots, q_{x+n-1})$$

Hence the function $a_{x:\overline{n}|i}$ can be represented by the function AXN, with syntax AXN *x n i*.

```
        ∇ Z←AXN V
[1]     Z←+/(×\1-V[2]↑(V[1]-W)↑Q)×(÷1+V[3])*⍳V[2]
        ∇
```

This function can easily be generalized to give the value of an *m*-thly annuity with syntax AXNM *x n i m*.

```
        ∇ Z←AXNM V
[1]     Z←V[2]↑(V[1]-W)↑Q
[2]     Z←×\,⍉(V[4],V[2])⍴(1-Z)*÷V[4]
[3]     Z←+/Z×(÷V[4])×(÷(1+V[3])*÷V[4])*⍳V[2]×V[4]
        ∇
```

It is possible to calculate tables of annuity values without a loop in the program. The following program produces values of $\{a_{x+t}\}_{t=0}^{w-1-x}$ at rate of interest i, with syntax TANX x i.

```
        ∇ Z←TANX V
[1]     X←V[1],0ρI←V[2]
[2]     Z←×\1-(((W-X),(W+1-X))ρ(X-W)↑Q)[(⍳W-X)∘.≥⌽⍳W+1-X
[3]     Z←3⍕+/Z×((W-X),(W+1-X))ρ(÷1+I)*⍳W+1-X
        ∇
```

It is not difficult to build up functions which derive the office's premium rates. For example, if the office premium is given by the formula

$$P_{x:\overline{n|}}=\frac{E+1000A_{x:\overline{n|}}}{(1-k)\ddot{a}_{x:\overline{n|}}}$$

we can write functions which we shall call ADVANX and ASX to evaluate $\ddot{a}_{x:\overline{n|}}$ and $A_{x:\overline{n|}}$.

The function to derive the office premium is then PXN, with syntax PXN x n.

```
        ∇ Z←PXN V
[1]     Z←E+1000×ASX V
[2]     Z←Z÷(1-K)×ADVANX V
        ∇
```

5. PROFIT TESTING

APL programs can be used to good effect for profit testing of unit-linked products, and each item of cash flow can be held as a vector. The following example is taken from Forfar and Gupta (1986),[3] page 38.

Example: A unit-linked policy has a 5-year term and an annual premium of £200. The death benefit is £1000 or the bid value of the units if higher. The maturity value is the bid value of the units. The allocation percentage is 90% throughout. The bid price is 95% of the offer price. All premiums are invested in accumulation units under which the management charge is ·75%. On surrender of the policy the value of the accumulation units is paid less a penalty. The penalty is 4% of the total premiums outstanding under the policy. The sterling interest rate is 6·0%, the fund growth rate 8·0% and the discount rate 12·0%. Expenses are £80·00 initially and £8·00 in each subsequent year. The mortality rate is ·85% p.a., and the rate of withdrawal 5% p.a.

The function PT produces an analysis of the sterling fund in the form outlined in Table 13 of the above reference. The syntax is PT p a n, where p is the annual premium, a the allocation and n the term.

```
      ∇PT V
[1]   VCOSTALLOC←V[2]×0.95×(VP←(N←V[3])⍴V[1])
[2]   VNOTALLOC←VP-VCOSTALLOC
[3]   VGR←(GR←1.08×1-0.0075)*¯1+⍳N
[4]   VFUND←GR×VGR×+\VCOSTALLOC÷VGR
[5]   VMANCH←(0.0075÷1-0.0075)×VFUND
[6]   VEXP←80,(N-1)⍴8
[7]   VINT←0.06×VNOTALLOC-VEXP
[8]   VMORT←0.0085×(0⌈1000-VFUND)
[9]   VLAPCOST←0.05×0.04×VP×⌽¯1+⍳N
[10]  VCASHFL←VNOTALLOC+VINT+VMANCH+VLAPCOST-(VEXP+VMORT)
[11]  VSURV←1,×\(N-1)⍴(1-0.05+0.0085)
[12]  NPV←+/(÷1.12*⍳N)×VPROF←VCASHFL×VSURV
[13]  '    YEAR      PRM       INT       MAN      SURR      EXP      MORT  CASHFL   PROFIT'
[14]  ' '
[15]  8 2⍕⍉(9,N)⍴(⍳N),VNOTALLOC,VINT,VMANCH,VLAPCOST,(-VEXP,VMORT),VCASHFL,
      VPROF
[16]  ' '
[17]  '       NPV  ',2⍕NPV
      ∇
```

6. PRICING OF OPTIONS

A simple APL function can be written for pricing a call option according to the Black–Scholes equation. It is first necessary to write a function which calculates the probability that a standard normal variable is less than a given value x. It is known that a good approximation to the area under the normal curve between a given value x and ∞, is given by

$$\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} (\cdot4361836\, t - \cdot1201676\, t^2 + \cdot9372980 t^3)$$

$$t = (1 + \cdot33267x)^{-1}$$

Accordingly the following function gives the probability that a standard normal variable will be less than or equal to x,

```
      ∇Z←NORMAL X;A;T
[1]   A←(0.4361836×T)+(¯0.1201676×T*2)+(0.937298×(T←÷1+0.33267×|X)*3)
[2]   A←A×(÷(2×○1)*0.5)×*¯0.5×X*2
[3]   Z←⍎4⍕((X≥0)×(1-A))+((X<0)×(A))
      ∇
```

The Black–Scholes equation for the price of a call option is

$$P\cdot\mathrm{NORMAL}\ (d_1) - E\, e^{-\delta T}\cdot\mathrm{NORMAL}\ (d_2)$$

where

P = current share price
E = exercise price of the option
T = time to expiry of the option
δ = risk free rate of return
σ = a measure of the volatility of the share price

$$d_1 = \frac{(\ln P/E) + (\delta + \frac{\sigma^2}{2})T}{\sigma\sqrt{T}}$$

$$d_2 = \frac{(\ln P/E) + (\delta - \frac{\sigma^2}{2})T}{\sigma\sqrt{T}}$$

The function CALL evaluates the price of a call option according to the syntax CALL *p e t*, where *p* is the share price, *e* the exercise price and *t* the time to expiry.

```
      ∇Z←CALL V;P;E;T;D1;D2
[1]   P←V[1],0⍴E←V[2],0⍴T←V[3]
[2]   D1←(÷SIGMA×T*0.5)×(⍟P÷E)+(DELTA+0.5×SIGMA*2)×T
[3]   D2←(÷SIGMA×T*0.5)×(⍟P÷E)+(DELTA-0.5×SIGMA*2)×T
[4]   Z←(P×NORMAL D1)-(E×*-DELTA×T)×(NORMAL D2)
      ∇
```

7. A STATISTICAL EXAMPLE INVOLVING SIMULATION

As a statistical exercise it is interesting to calculate the distribution of high card points (Ace = 4, King = 3, Queen = 2, Jack = 1) in a hand at bridge. The number of high card points can vary between 0 and 37. The following function deals *n* independent hands and returns the distribution and cumulative distribution of high card points (see Gilman and Rose, 1974).[4] The syntax is HCP *n*. It is the fact that the syntax of APL enables the function to be so short which is interesting.

```
      ∇Z←HCP N;I;T
[1]    I←1,0⍴T←⍳0
[2]   ST:T←T,+/(⍳4)×+/4 4⍴+⌿(13?52)∘.=⍳16
[3]    →(N≥I←I+1)/ST
[4]    I←0,0⍴A←⍳0
[5]   ST1:A←A,(+/I=T)
[6]    →(37≥I←I+1)/ST1
[7]    Z←3 38⍴(¯1+⍳38),A,+\A
      ∇
```

8. LEAST SQUARES ESTIMATES

This example illustrates the matrix handling capabilities of APL by showing how a short program in APL can calculate the polynomial best fitting a set of data points and return the values of the polynomial at the points and the least squares value at the minimum point.

If A and B are matrices then

$$A+\cdot\times B$$

represents the matrix product of A and B. The symbol ⌹B represents the matrix inverse of B. The decode function $\perp$ operates such that

$$x\perp a_n \quad a_{n-1}\ldots a_0$$

represents the value of

$$a_0+a_1x+a_2x^2\ldots a_nx^n$$

and hence derives the value of the polynomial at the point x.

We suppose we have a data set $\{y_i, x_i\}_{i=1}^{n}$ and we wish to derive the best fitting polynomial of order m, where $m\leqslant n$. We suppose the polynomial is

$$p(x)=a_0+a_1x\ldots a_mx^m.$$

The parameters to minimize D^2 where

$$D^2=\sum_{i=1}^{n}(y_i-p(x_i))^2$$

are given by the solution to the normal equations, namely

$$YM(M^TM)^{-1}$$

where M is the matrix,

$$\begin{bmatrix} 1 & x_1 & x_1^2 \ldots & x_1^{m-1} \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ 1 & x_n & x_n^2 \ldots & x_n^{m-1} \end{bmatrix}$$

The syntax is $m \;\; y_1\ldots y_n$ FIT $x_1\ldots x_n$, where m is the order of the polynomial to be fitted.

```
      ∇Z←Y FIT X;A;B
[1]    'PARAMETERS      ',6⍕A←((1↓Y)+.×M)+.×⌹(⍉M)+.×M←X∘.*¯1+⍳(⍴X)⌊(1↑Y)
[2]    'FITTED VALUES   ',6⍕B←,(((⍴X),1)⍴X)⊥((1↑Y),1)⍴⌽A
[3]    'SUM OF SQUARES ',6⍕+/(B-(1↓Y))*2
      ∇
```

9. CONCLUSION

The above examples are only a few of the many which could be given to illustrate the power and usefulness of APL. The interested reader is referred to the references for further information.

REFERENCES

(1) ANSCOMBE, F. (1981). *Computing in Statistical Science through APL*, Springer-Verlag.

(2) BARTOLA, J. A. (1982) XVI ASTIN Colloquium, Liège.

(3) FORFAR, D. O. & GUPTA, A. K. (1986) The Mathematics of Profit Testing for Conventional and Unit Linked Business, Institute of Actuaries Study Note.

(4) GILMAN, L. & ROSE, A. J. (1974) *APL, An Interactive Approach* (2nd edition), John Wiley.

(5) LEUVEN, H. Report to the Dutch Actuarial Society.

(6) MOREAU, L. (1984) *Transactions of the International Congress of Actuaries*, Sydney.

(7) NEUBURGER, E. (1984) Der Barwert als Zahl einem Zahlensystem mit verallgemeinerter Basis, DGVM, Band XVI, Heft 3.

(8) OYAL, A. (1985) Vektorielle Barwerte und Deckungskapitale in einer APL, Darstellung, DGVM, Band XVII, Heft 2.

(9) STIERS, D., GOOVAERTS, M. J. & DE KERF, J. (1987) *APL, The Language and its Actuarial Applications*, North Holland.