TRANSACTIONS OF SOCIETY OF ACTUARIES
1975 VOL. 27

RECURSIVE DEFINITIONS OF ACTUARIAL FUNCTIONS

EDWARD J. SELIGMAN

ABSTRACT

Most mathematical problems requiring a numerical solution can be formulated in either an explicit or an implicit form. The first expresses the unknown as an explicit function of problem parameters; that is, the unknown appears alone on the left-hand side of an equation. The second formulation expresses the unknown as the root of a (usually) nonlinear equation. Very often, the implicit approach is simpler to arrive at but requires far more arithmetical computation to solve the equation. The recent dramatic increase in the speed of electronic computation now makes the implicit method quite attractive. The method is applied to several problems of actuarial interest.

THE high-speed electronic computer has produced more than the great enhancement of the speed of standard computational methods of problem-solving: it has introduced new methods and made old ones obsolete. Certainly no one who owns a pocket-size electronic calculator with trigonometric functions wants to know how to use interpolation formulas when he needs the sine or cosine of an angle which lies between tabulated values of a trigonometric table. On a larger scale, certain scientific problems are now solved numerically by the solution of linear systems of several hundred equations ("implicit" methods), where "explicit" methods had been applied laboriously with less satisfactory results.

Actuarial problems are being solved on high-speed electronic computers, but very often methods of solution based on commutation functions and standard compound interest notation fail to let the computer do as much of the work as it should. In many problems it is relatively easy to express the solution implicitly—that is, to write an equation in the unknown—and then solve the equation for the unknown analytically or numerically. We shall consider the equations generated by the recursive relations which arise in life contingencies and compound interest problems. Four examples will be considered—two compound interest problems and two life contingencies problems.

I. LOAN AMORTIZATION WITH STEP-RATE AMOUNTS OF PRINCIPAL[1]

Given is a loan to be amortized by a series of n level payments. The loan balance is segmented, with each segment having a different rate of interest. For example, the interest rate may be $1\frac{1}{2}$ per cent per month on the first \$1,000 of outstanding balance and 1 per cent per month on the excess over \$1,000. An explicit solution to this problem for the general m-segment case by Stephen G. Kellison appears in *ARCH*, 1973.5. The implicit solution, obtained from a recursive definition, is

$$l_0 = L\,, \tag{1}$$

$$l_{j+1} = l_j - [P - i(l_j)]\,, \quad j = 0, 1, \ldots, n-1\,, \tag{2}$$

$$l_n = 0\,, \tag{3}$$

where

L = Amount of original loan;

n = Number of payments;

l_j = Amount outstanding after jth payment;

$i(l_j)$ = Interest due at jth payment;

P = Level payment (to be found).

The function $i(l_j)$ defines the interest due on the outstanding amount l_j. For the previous numerical illustration,

$$i(l_j) = 0.015 l_j \qquad \text{for } l_j \le 1{,}000\,;$$

$$i(l_j) = 15 + 0.01(l_j - 1{,}000) \qquad \text{for } l_j > 1{,}000\,.$$

In general, define $i(l_j)$ to be any arbitrary function of l_j, the outstanding balance. Equations (1)–(3) then define a nonlinear equation in P which easily can be solved numerically for P (the Appendix gives a general numerical procedure for solving nonlinear equations in one unknown).

II. THE RETIREMENT INCOME POLICY

According to Jordan [2], "the policy can be regarded as a combination of a pure endowment of $1 + k$ due at the retirement age and a term insurance up to that age providing a death benefit equal to 1 or the total cash value on the policy, whichever is greater." If x is the age at issue and $x + n$ the retirement age, the amount of the pure endowment, $1 + k$, is equal to the present value at age $x + n$ of the annuity to be provided at that age. The derivation of the net premium P, payable for n years, is complicated by the fact that the total cash value is a function of P, which is not known in advance. However, the analysis can be simplified

[1] This example first appeared in *ARCH* (*Actuarial Research Clearing House* [Ann Arbor, Mich.]), 1974.1.

greatly by using a recursive formula connecting the successive yearly cash values instead of the method in Jordan which derives P explicitly using commutation functions. The recursive formula resembles those found in Jordan on pages 106–7. Assume that yearly cash values are equal to net level premium reserves.

$$ {}_0V_x = 0\,, \tag{4}$$

$$ Z = [({}_tV_x + P)(1+i) - q_{x+t}]/p_{x+t}\,, \qquad t = 0, 1, \ldots, n-1\,, \tag{5}$$

$$ {}_{t+1}V_x = Z \qquad \text{for } Z \le 1\,, \tag{6}$$

$$ {}_{t+1}V_x = ({}_tV_x + P)(1+i) \qquad \text{for } Z > 1\,, \tag{7}$$

$$ {}_nV_x = 1 + k\,, \tag{8}$$

where

P = Net yearly premium ;

${}_tV_x$ = tth-year terminal reserve;

p_{x+t} = Probability that an individual aged $x + t$ survives 1 year;

$q_{x+t} = 1 - p_{x+t}$;

i = Yearly interest rate .

Again, we have a nonlinear equation in an unknown (P), the desired yearly premium. The equation defined by formulas (4)–(8) is easily solved numerically for P. Jordan's a, the last value of t for which cash value is less than or equal to 1, is not required.

Before proceeding to the next example, one should note that a major advantage of the recursive approach is that it permits a "dynamic" definition of the problem. One can proceed from month to month, or year to year, with the problem definition dependent on the state of one or more variables at the current time period. In the first example the formula for the interest rate is a function of the outstanding loan amount. In the second example the formula for the $(t + 1)$st terminal reserve depends on a function of the tth terminal reserve.

III. A BOND PROBLEM

The following problem appears as Example 8.3 (p. 163) of Donald [1]: "A bond of 1250 is redeemable at 105% by 25 equal installments of capital, the first due 6 years hence. Interest, which is payable half-yearly, is at $4\frac{1}{2}\%$ in the first year after purchase, and thereafter decreases by $\frac{1}{40}\%$ each year. What price should be paid to yield 4% per annum convertible half-yearly?"

The solution using standard compound interest techniques takes just over one page in Donald. While this problem does not have the "dynamic" definition of the first two examples, the changing interest rate makes the recursive formulation quite natural. Let

$$V_0 = R\,, \tag{9}$$

$$K_0 = 1{,}250\,, \tag{10}$$

$$Z = 1.02V_j - \tfrac{1}{2}K_j(0.045 - 0.01m/40)\,, \tag{11}$$

$$\left.\begin{aligned} V_{j+1} &= Z - 52.50 \\ K_{j+1} &= K_j - 50.00 \end{aligned}\right\} \quad \text{for } j+1 \geq 12 \text{ and } j+1 \text{ an even integer}\,, \tag{12}$$

$$\left.\begin{aligned} V_{j+1} &= Z \\ K_{j+1} &= K_j \end{aligned}\right\} \quad \text{for } j+1 < 12 \text{ or } j+1 \text{ an odd integer}\,, \tag{13}$$

$$j = 0, 1, \ldots, 59\,,$$

$$m = [j/2] \text{ (greatest integer in } j/2)\,,$$

$$K_{60} = 0\,, \tag{14}$$

where

V_j = Purchaser's investment after jth half-yearly period ;

K_j = Nominal outstanding loan after jth half-yearly period ;

R = Purchase price of bond to yield 4 per cent yearly, convertible half-yearly (to be found) .

Equations (9)–(14) define a nonlinear equation in R, giving an answer $R = 1{,}327.43$, correct to 6 significant figures.

IV. MINIMUM CASH VALUES

According to Jordan [2],

> [The computed minimum cash values] are defined by law to be adjusted premium surrender values computed on a prescribed mortality and interest basis with the extra initial expense E^1 derived from the following formula (based on a level insurance amount of $1000):
>
> (a) 40% of the adjusted premium for the policy, but the amount not to exceed $16; plus
>
> (b) 25% of the adjusted premium for the policy or of the adjusted premium for an otherwise similar ordinary life policy, whichever is less, but the amount not to exceed $10; plus
>
> (c) $20.

Suppose it is desired to compute the ordinary life adjusted premium P_x^A for age x. The amount available for cash value is $P_x^A - E^1$ for the first year, and P_x^A for all subsequent years. As noted in Jordan, the computation is complicated by the fact that E^1 is a function of P_x^A. With a recursive definition, however, P_x^A can be quickly defined implicitly.

$${}_0V_x = 0\,,$$

$$E^1 = 0.65 \min\,(P_x^A, 0.04) + 0.02\,,$$

$${}_1V_x = [({}_0V_x + P_x^A - E^1)(1+i) - q_x]/p_x\,,$$

$${}_{t+1}V_x = [({}_tV_x + P_x^A)(1+i) - q_{x+t}]/p_{x+t}\,, \qquad t = 1, 2, \ldots, \omega - x - 1\,,$$

$$({}_{\omega-x}V_x + P_x^A)(1+i) = 1\,,$$

where

${}_tV_x$ = tth-year terminal reserve ;

p_x = Probability that an individual aged x survives 1 year ;

$q_x = 1 - p_x$;

ω = Smallest x such that $p_x = 0$;

i = yearly rate of interest ;

P_x^A = Renewal net premium (to be found).

The numerical solution of the above equation for P_x^A now follows routinely.

Two important things may be noted concerning the recursive solutions to the four examples. First, the recursive approach is the "natural" solution. The growth and decay of terminal reserves or outstanding loan amount are easy to follow from period to period, while the conventional solution using commutation functions and compound interest notation is often tortuous and contrived. Both methods, of course, are mathematically exact. Second, the recursive method requires much more computation than does the conventional method and, normally, should not be considered unless the analyst has access to a computer. Otherwise, conventional methods should be used. In any event, present-day technology requires that the computing facilities available become part of the solution, not merely an appended footnote.

APPENDIX

There are many ways to solve numerically a nonlinear equation in one unknown, $f(x) = 0$. The author of this paper prefers the method detailed below, for the following reasons: (1) It does not require the computation (or existence) of the derivative of $f(x)$. (2) It is very reliable; in particular, the method always converges if $f(x)$ is continuous, but continuity is not a necessary condition. (3) It is easy to program for a computer.

The method is that of "interval halving" and is based on a theorem due to Bolzano which states that, if there exist numbers a and b such that $f(a) < 0$, $f(b) > 0$, and $f(x)$ is continuous for x between a and b, then there exists a number c between a and b such that $f(c) = 0$. The estimate of c is $(a + b)/2$. We can then set up the iterative scheme

$$x_{i+1} = \tfrac{1}{2}(x_i + x_{i-1})\,, \quad i = 1, 2, \ldots,$$

where x_i and x_{i-1} are chosen so that $f(x_i)$ and $f(x_{i-1})$ have opposite signs. The iteration terminates when $|f(x_i)|$ is less than some preassigned tolerance. This method is not particularly fast, but its nearly unfailing reliability makes it preferable to more sophisticated schemes. The author believes that the small amount of computer time saved by more quickly convergent algorithms is dissipated the first time that one has to trace through a computer memory dump to find out why the algorithm failed. A suggested FORTRAN program follows:

```
C  PLOW AND PHIGH ARE CHOSEN SO THAT F(PLOW) AND
   F(PHIGH) HAVE OPPOSITE SIGNS.
   50 PNEW = (PLOW + PHIGH)/2
      FN = F(PNEW)
      IF (ABS(FN) - EPS) 40,40,30
C  EPS IS THE CONVERGENCE TOLERANCE
   30 IF (FN*F(PLOW)) 10,40,20
   10 PHIGH = PNEW
      GO TO 50
   20 PLOW = PNEW
      GO TO 50
C  ACCEPT PNEW AS THE ROOT OF F(X) = 0
   40 -------------
```

In order to start the iterative algorithm, we need two values, x_0 and x_1, such that $f(x_0)$ and $f(x_1)$ have opposite signs or, equivalently, $x_0 < c < x_1$, where c is the desired root of $f(x) = 0$. Of course, it is desirable to choose x_0 and x_1 as close to c as possible, since the iteration will then converge more quickly. With the recent advances in the speed of computer arithmetic, however, even outlandishly conservative guesses for x_0 and x_1 waste very little time. For example, less than one second of computer time was needed for 6-digit accuracy on all four examples with the guesses shown in the accompanying tabulation

Example	x_0	x_1
I.........	0.0	10,000.0
II........	0.0	1.0
III.......	0.0	10,000.0
IV........	0.0	1.0

for x_0 and x_1. A Digital Equipment Corporation PDP-6 computer was used for all examples.

REFERENCES

1. DONALD, D. W. A. *Compound Interest and Annuities-certain.* Cambridge: Cambridge University Press, 1963.
2. JORDAN, C. W., JR. *Life Contingencies.* Chicago: Society of Actuaries, 1967.

DISCUSSION OF PRECEDING PAPER

T. N. E. GREVILLE:

I would like to applaud this paper, which has substantial mathematical content but is nevertheless very practical and useful. The approach of solving complicated problems by defining the unknown quantity recursively and using the computer to solve the resulting nonlinear equation makes a great deal of sense.

The main purpose of this discussion is to make a case for using the time-honored regula falsi (or "method of false position") as an option alongside the interval-halving method in seeking the solution of the equation. As I see it, this has two advantages: (1) in most cases the regula falsi will converge faster (although I fully agree with the author that rapidity of convergence is not the paramount consideration);(2) the regula falsi will provide certain information the user might like to have that would not be available if the interval-halving method were used alone.

The regula falsi uses linear interpolation between preceding estimates to arrive at the next estimate. It is based on the iterative scheme

$$x_{i+1} = \frac{x_{i-1}f(x_i) - x_i f(x_{i-1})}{f(x_i) - f(x_{i-1})} = x_i - \frac{f(x_i)}{f(x_i) - f(x_{i-1})}(x_i - x_{i-1}) \; .$$

Clearly this will "blow up" if the denominator is too small. The interval-halving method can never blow up because there is no division (except by 2); this is its great virtue.

Now, if the denominator of the regula falsi is very small (or even zero), this suggests that either (1) the $f(x)$ curve is almost horizontal in the interval or (2) the $f(x)$ curve undulates in the interval. I think the first condition is much more likely to occur in actuarial problems.

If the $f(x)$ curve is almost horizontal, the determination of the required root c is very rough, because $|f(x)|$ is less than the prescribed tolerance for every x in a sizable neighborhood of c. There is not much the user can do about this, but he may want to know that the condition exists. In this case the interval-halving method will give an answer very quickly but will give no warning that the determination is rough.

If the curve undulates, $f(x)$ may well have more than one zero in the interval, and this, too, the user may wish to know. He may want the smallest (or the largest) zero and may need to exercise some care that the one desired is the one obtained.

My proposal, then, is to start out with the regula falsi but to switch to interval halving (with an appropriate message printed out) in case the going gets rough. A suggested FORTRAN program is as follows:

```
C  PLOW AND PHIGH ARE CHOSEN SO THAT F(PLOW) AND
   F(PHIGH) HAVE OPPOSITE SIGNS
   60 DEN = F(PHIGH) - F(PLOW)
      IF (ABS(DEN) - EPS2) 70,70,80
C  IF DENOMINATOR IS TOO SMALL WE SWITCH TO
   INTERVAL HALVING
   70 PRINT MESSAGE
C  MESSAGE INFORMS USER THAT PROGRAM IS SWITCHING
   TO INTERNAL HALVING
      PNEW = (PLOW + PHIGH)/2
      GO TO 90
   80 PNEW = PHIGH - (PHIGH - PLOW)*F(PHIGH)/
   (F(PHIGH) - F(PLOW))
   90 FN = F(PNEW)
      IF (ABS(FN) - EPS) 40,40,30
C  EPS IS THE CONVERGENCE TOLERANCE
   30 IF (FN*F(PLOW)) 10,40,20
   10 PHIGH = PNEW
      GO TO 60
   20 PLOW = PNEW
      GO TO 60
C  ACCEPT PNEW AS THE ROOT OF F(X) = 0
   40 ---------
```

The author of the paper has suggested to me that the second algebraic form of the regula falsi, which is the one embodied in statement 80 of the program, often gives better accuracy on a digital computer because of the vagaries of floating-point arithmetic.

Note how this combination procedure is likely to work out in different cases.

1. In well-behaved cases the regula falsi will be used exclusively.
2. When the denominator of the regula falsi becomes small because the $f(x)$ curve is almost horizontal, the program will switch permanently to interval halving. An answer of doubtful accuracy will be produced, and the user will be warned.
3. When the denominator of the regula falsi becomes small because the curve undulates, the program is likely to oscillate between the regula falsi and interval halving. An answer will be obtained, which may be accurately determined, but if there is more than one zero of $f(x)$, it is a matter of luck which zero will be indicated, in the absence of intervention by the user.

Both the regula falsi and interval halving are discussed on page 1179 of K. Rektorys (ed.), *Survey of Applicable Mathematics* (Cambridge, Mass.: MIT Press, 1969).

STEPHEN G. KELLISON:

Mr. Seligman's excellent paper shows that many standard actuarial problems can be solved by an implicit or iterative method as well as by the more traditional explicit or analytical method. The paper is quite timely not only because of the greater computer capacity in existence today, which makes the implicit approach a practical technique for the solution of such problems, but also because of a basic change in the mathematical training which actuaries receive today.

In 1971 the Society of Actuaries Education and Examinations Committee eliminated the subject of finite differences from the examination syllabus and replaced it with numerical analysis. This paper utilizes two basic concepts from numerical analysis which were not previously on the syllabus.

The first of these is iteration, which can be characterized as a systematic refinement of the age-old technique of successive approximation. Mr. Seligman's paper utilizes the Bolzano algorithm, which is one of the standard iteration methods covered in numerical analysis. The current examination syllabus presents this algorithm and also several other standard algorithms, such as the method of successive substitutions, successive inverse interpolation, and the rapidly converging Newton-Raphson method, as well as some variations. The advantages and disadvantages of these methods are explored, including the rate of convergence and the complexity of the computation involved.

The second concept involved is that of a recursion formula, which relates successive functional values at finite intervals. As Mr. Seligman's paper illustrates, many standard actuarial problems can be expressed as recursion formulas. In numerical analysis a finite analogue to a differential equation is considered and is termed a "difference equation." Since differences can be broken down into combinations of functional values at finite intervals, it is immediately seen that a difference equation can also be viewed as a recursion formula. The current examination syllabus contains a systematic treatment of the solution of difference equations.

Mr. Seligman is to be congratulated on presenting a paper which is not only of significance in the solution of frequently encountered actuarial problems but is also quite timely in the education of actuaries.

QUINTIN J. MALTBY:

I hope that this paper will help dispel any lingering fears that nonanalytic solutions to compound interest, life contingency, and like problems are somehow not quite proper. I have been involved for some years in solving the unknown-yield problem and hope that the following comments will provide a proper expansion upon the concepts in the paper and also be of value to those faced with the realities of this particular problem in practice.

The recursive relation for a fixed-interest bond or mortgage is

$$V_t = V_{t-1}(1 + i) - P_t ,$$

where

V_t = Value at the end of period t;

P_t = Total payment due at the end of period t;

i = Per period yield factor;

n = Number of periods;

the object of the game is to find an i that causes V_n to equal zero.

The related analytic equation is well expressed as a polynomial in v equated to zero, namely,

$$P_n v^n + P_{n-1} v^{n-1} + \ldots + P_1 v - V_0 = 0 .$$

When $V_0 > 0$, $P_t > 0$ for all t, and $i > 0$, the functions are all very well behaved in the sense that the solution (however obtained) to the equation in v has a single real root and the dollars-and-cents V_n obtained from the recursive relation is a monotone nondecreasing function of i. The naval gunnery technique of interval halving as described in the appendix to the paper has worked very well for us under the above circumstances.

As soon as some of the restrictive conditions are lifted, however, the road to a solution becomes more tortuous. In actual practice I have allowed the P_t amounts to be negative (to represent additional mortgage advances re tax bill payments) and allowed the yield factor to go negative (re real returns during inflation). Under these relaxed conditions we are dealing analytically with a polynomial whose left-to-right graph may start either very high or very low, may have one or more maximum and/or minimum, and finally becomes monotone increasing. If more than one real solution exists, it is the highest such solution that we need in practice.

The following procedure was devised to solve for i under relaxed conditions. Some trial and error was involved before the final version was attained. It is presented only as verbal logic because the computer logic involves a great many assembler language statements. The assumptions and definitions are stated in step 1, and the logic itself starts at step 2.

1. (*a*) $V(i)$ is the result of n iterations of the recursive formula mentioned above.
   (*b*) f is a fraction, $0 < f < 1$. In practice I am using $f = 0.8$.
   (*c*) d is the least possible effective increment that can be made to i. In other words, it is 1 in the lowest-order digit of the variable.

(*d*) All i's studied are in the range $-1 < i < +1$.
(*e*) Whenever a variable value is "moved," its related function value is moved along with it.
(*f*) The variable values dealt with are H, HW, L, LW, and S. These refer to high bracket value, high working value, low bracket value, low working value, and saved valued, respectively.
(*g*) A switch is set up to tell us whether we are "on a negative slope" or "not on a negative slope."

2. Initial extreme values of L and H are set up, $-0.999\ldots$ and $+0.999\ldots$, respectively (i.e., as $-1 + d$ and $1 - d$). $V(H)$ is found and defeat conceded forthwith if $V(H)$ is not greater than zero. $V(L)$ is also found. The switch is set to say "not on negative slope," L is moved to LW, and H is moved to HW. The purpose of steps 3–9 is to find the lowest value of H that produces $V(H) > 0$, with all higher values of H also producing $V(H) > 0$.
3. (*a*) i is set up to be the lower of $(1 - f)LW + fHW$ and $HW - d$.
   (*b*) If $i = HW - d = LW$, we route to step 8 below.
   (*c*) A value $V(i)$ is computed.
   (*d*) If $V(i) > V(HW)$, we route to step 5 below to handle the negative-slope condition.
4. Here we have $V(i) \leq V(HW)$. The logic always routes to step 3 above after the appropriate operation *a* or *b* or *c* has been completed.
   (*a*) If $V(i) < 0$, i is moved to L and LW and the switch is reset to say "not on a negative slope."
   (*b*) If $V(i) = 0$, i is moved to LW only.
   (*c*) If $V(i) > 0$, HW is moved to H and then i is moved to HW.
5. Here we find ourselves on a negative slope from step 3(*d*) above.
   (*a*) i is moved to LW.
   (*b*) If the switch says "not on a negative slope," it is changed to the new truth and i is moved to S to save this point of negative-slope detection.
6. From this point we try to find the slope at HW and then react accordingly.
   (*a*) i is set up to be the lower of $(1 - f)LW + fHW$ and $HW - d$.
   (*b*) If $i = HW - d = LW$, we route to step 7 below.
   (*c*) A value $V(i)$ is computed.
   (*d*) If $V(i) > V(HW)$, we route to step 7 below.
   (*e*) If $V(i) < V(HW)$, we have found some positive slope and thus go to step 4 above to look after things.
   (*f*) If $V(i) = V(HW)$, i is moved to HW and we reloop to 6(*a*) just above to try again.
7. At this point we know that the slope at HW is negative, when related to a just lower variable value of i. First HW is moved to LW, and then H is moved to HW. Then we reloop to step 3 above to see whether we can detect a minimum in the new working range below zero.
8. At this point the range LW to HW being studied via step 3 has closed up.
   (*a*) HW is moved to H.
   (*b*) If $V(LW) < 0$, we route to the final operations of step 13 below.

(c) If $V(LW) = 0$, L is moved to LW and we route to step 10 below.
(d) If $V(LW) > 0$, we route to step 9 just below.

9. Here we have studied a portion of the curve with a negative slope and found no crossing of the origin. Thus it can be safely ignored.
   (a) If $V(L) \geq 0$, defeat is conceded.
   (b) If $V(L) < 0$, L is moved to LW, S is moved to H and HW, the switch is set to say "not on a negative slope," and we loop back to step 3 above.
10. At this point we have found our desired H-value and now wish to find the highest L which is less than H and which gives a value $V(L) < 0$.
11. (a) i is set up to be the lower of $(1-f)LW + fHW$ and $HW - d$.
   (b) If $i = HW - d = LW$, we route to step 12 below.
   (c) A value $V(i)$ is computed.
   (d) If $V(i) = 0$, i is moved to HW and we reloop to 11(a) just above.
   (e) If $V(i) \neq 0$, i is moved to LW and we reloop to 11(a) just above.
12. If $V(LW) \geq 0$, we concede defeat.
13. At this point we have bracketing values of LW and H. A best-solution value for i is determined as the mean of LW and H. If any rounding of this value is needed, it is toward the value giving the $V(LW)$ or $V(H)$ that is closest to zero, with $V(H)$ getting the nod on equals.

LESTER R. McCRACKEN:

Mr. Seligman is to be commended for his excellent demonstrations of the successive iteration technique for solving nonlinear equations. Perhaps the portion of the appendix which points out that the iteration should terminate when the absolute value of the function is smaller than some given delta should be underlined in red. I am sure that many hours of computer time have been wasted by "programmers" attempting to iterate to a tolerance of exactly zero, which was beyond the accuracy limit of either the computer or the equation itself.

The recursive approach is not a universal solution, however. While this approach will generate a theoretically correct number for Example IV (minimum cash values), this result may not be identical with the published value. (This is because *1958 CSO Derived Values* is based on precisely 10-digit N_x and M_x values, with all calculations performed on a machine which would be treated like a ten-bank calculator.) Although the recursive program's results will differ only by pennies from the published values, they may be unacceptable for ratebook purposes, since these values might be rejected by the state insurance departments because they do not match the published amounts.

An additional caveat is needed: be sure that $f(x)$ is actually continuous rather than only apparently so. In 1968 a number of letters appeared in *The Actuary* demonstrating pseudocontinuous equations with multiple roots. I quote in part from a letter from Daniel S. Harris, published in September, 1968:

Sam's transactions were all in one share of Stock X as follows:

12/31/64: Price of Stock X = \$1,000.00; Sam buys.
12/31/65: Price of Stock X = \$3,180.00; Sam sells.
12/31/66: Price of Stock X = \$3,370.70; Trend of stock looks very good; Sam buys.
12/31/67: Price of Stock X = \$1,190.91; Looks as if bottom is dropping out; Sam panics; Sam sells.

Sam figures his yield:

$$1{,}000 + 3{,}370.70v^2 = 3{,}180v + 1{,}190.91v^3 .$$

Multiplying by $(1+i)^3/1{,}000$ and transposing,

$$(1+i)^3 - 3.180(1+i)^2 + 3.3707(1+i) - 1.19091 = 0 .$$

Factoring and solving,

$$[(1+i) - 1.05][(1+i) - 1.06][(1+i) - 1.07] = 0 .$$

Thus $i = 5$, 6, or 7 per cent.

Sam's broker is still trying to explain to him that these things can happen when there is an adjustment in the market.

The reader can observe that there is an obvious discontinuity in this example between December 31, 1965, and December 31, 1966, since Sam was out of the market.

Unfortunately, there is a practical extension to the problem set forth in the Harris letter. If you were asked to calculate the effective yield of a portion of an investment fund (including unrealized capital gains and losses) over a market such as that experienced from 1971 through the first quarter of 1975, the equation might either (1) not converge at all or (2) converge only if the iterative fund values were allowed to stray outside their normal domain at intermediate points. It would seem that the proper procedure would be to calculate yields for several intervals where the performance was fairly stable and then compute a geometric mean of the $1 + i$ values calculated.)

Despite the above-mentioned possibilities for error in application of the technique, the recursive approach is a valuable weapon in the actuary's arsenal. In fact, it may be the only conceptually easy approach when the formulas are complex and the limiting conditions themselves are recursive.

THOMAS P. TIERNEY:

Mr. Seligman has written a very fine paper and one which raises quite topically the question as to how an actuary in 1975 should go about solving computer-based problems (and those that are not automated but ought to be). The answer, as shown by his four examples, will usually be to adopt an approach that minimizes human effort and transfers as much work as possible to the computer. Translated into machine specifications, this means that the problem-solver should be striving to express a problem solution in its simplest possible form. Although it is true that this approach usually will result in an increase in machine computation costs, it is also true that it should result in a sizable decrease in overall costs. This is true because simplicity-based increases in computer production expenses are almost always more than offset by concomitant decreases in machine testing and personnel costs. Production costs typically run at less than 20 per cent of a total research and development budget, and experience has shown that simplification efforts aimed at the remaining 80 per cent usually have the larger payoff.

Unfortunately, this emphasis on conceptual simplicity is not as much a part of today's data-processing world as it should be. Undoubtedly, part of the reason for this is that the machine costs are much more noticeable and measurable and so they tend to be the focus of our efficiency efforts. However, there are two other reasons which are probably just as significant:

1. *The automating of manual procedures.*—This is related to Mr. Seligman's comments about "methods . . . [that] fail to let the computer do as much of the work as it should," and it refers to the structuring of a computer process in a pattern which parallels very closely the corresponding manual process. This almost always turns out to be a very bad approach, since men and computers differ in that what is easy for one is usually quite difficult for the other and vice versa. The machines should complement rather than emulate the analyst. There are literally millions of examples of this tendency to make the machine act like a man, and they include such things as performing annual recursive accumulations with tricky once-a-year modal adjustments when a monthly accumulation would have been simpler, and the use of complicated commutation functions that are dripping with prescripts, postscripts, subscripts, and the like, when a simple summation would have been much more appropriate. The villain in these cases is occasionally the lazy analyst who doesn't feel like making a proper design effort, but more often than not it is just an unawareness of what these big, underworked data-crunching machines can really do.

2. *The so-called programmer ego problem.*—"Programmer ego" is a term which began to appear in data-processing literature a few years ago; it refers to the tendency of systems designers to "hotdog" it, or to be tricky, confusing, and intricate in their formulation of problem solutions. Now, while this fanciness may show off one's skills and provide in a perverted sort of a way that all-important ego gratification, it is almost always counterproductive in the long run. The actuarial student should take particular note, lest he be misled by the earlier examinations into thinking that the conceptually brilliant solution

requiring only minimal computation is always the preferred way. Of course, in an examination environment it usually is, but there the goal is the testing of one's aptitude and skills and not the solution of problems per se. Back in the real world the preferred solution oftentimes will mean much more of a brute-force approach.

The science of computer systems design is still in its infancy (it is about where engineering was in the pyramid days), but it is developing very rapidly, and today's actuary has to keep pace. He will have to learn a lot more about computer capacity, which should not be too difficult, and he will have to approach the machines more humbly.

MATT B. TUCKER:

The choice of subjects for this paper is a very interesting one. The recursive approach to the solution of actuarial problems is a very powerful and flexible tool. These comments should further reinforce those attributes.

Recursive functions in actuarial literature are actually fairly old. One of the most familiar is Hoskins' asset share formula. The use of recursive expressions for calculations was not widespread in earlier years because of the laborious computations that were necessary to arrive at a solution. In many cases simplifying assumptions were made so that an explicit solution was found, thereby reducing the computational requirements. The advent of computers provided us with a tool that removed the computational difficulties. We did not recognize this very quickly. We were accustomed to commutation function expressions (which were devised to ease computations by hand) and oftentimes applied our old thinking to computer applications. I first encountered the recursive approach to actuarial problems in a system developed by IBM around 1960: '62CFO (an administrative maintenance system for life insurance policies—"consolidated functions ordinary") included some programs to calculate reserves and minimum cash values for the valuation and nonforfeiture calculation portion of that system. I believe the idea of recursive formulas was suggested by Karl Manchester, who was helping with the actuarial aspects of that system. The calculation programs used the interval-halving approach to solve for premiums and reserves.

Since that time I have used this approach fairly extensively. Some improvements can be made in the speed of convergence over the interval-halving approach, but the improvements are less reliable. The use of an approach called "regula falsi" improves the speed, but care must be taken in using this approach. I have used other approaches for specific problems or programs (approximate annuities to compute the "next premium guess"). The two following problems, which were solved using recursive functions on a computer, indicate some of the power of the recursive approach: (1) A plan of insurance in which the death benefit is return of cash value plus $1,000 for the first twenty years, with a level death benefit after twenty years of $1,000 plus the twentieth-year cash value. Cash values and reserves were computed for this plan. (2) A retirement income policy where the death benefit after the *a* year was not equal to the reserves or cash values. This was a continuous functions case where the immediate payment of claims caused a death cost above the return of cash value. There are many other actuarial problems that readily lend themselves to recursive approaches (graded reserves and cash values, varying interest rates).

I have encountered some technical problems that might be noted:

1. The q_x-l_x-d_x problem. In order to reproduce published premiums (which were computed using commutation functions), a q_x must be calculated to a larger number of decimal places than is published. This is needed because d_x/l_x (unrounded) $\neq q_x$. Commutation functions cause d_x/l_x to be used in calculations.
2. If floating-point numbers are not used, then great care must be taken in the size of numbers used. This was a significant problem on older computers.
3. Improved convergence becomes a problem when more complex examples are involved. For example: to calculate a plan with minimum cash values for five years, graded to Commissioners Reserve Valuation Method reserves at fifteen years and graded to net level reserves at twenty years, requires the computation of (1) adjusted premium for whole life, (2) renewal premium for FPT on a twenty-payment life, (3) minimum cash values for the plan, (4) CRVM reserves for the plan, (5) net level reserves for the plan, and (6) graded values for the periods (*a*) fifth through fifteenth year and (*b*) fifteenth through twentieth year. Each of the above requires complete iterations for solution and probably five-decimal accuracy in the premiums.

I foresee a tremendous expansion of this approach in the future. The following are just two items for future application: (1) cases in which the recursive expression involves variables that may be stochastic in nature and (2) solutions to problems in which the concern is not only for a mean expected value but also for variance analysis. The power and flexibility of recursive functions as applied to actuarial science will become more apparent in the future.

(AUTHOR'S REVIEW OF DISCUSSION)

EDWARD J. SELIGMAN:

I thank Dr. Greville, Professor Kellison, and Messrs. Maltby, McCracken, Tierney, and Tucker for their discussions of my paper.

Dr. Greville's alternate scheme for solving the equation $f(x) = 0$ by starting with regula falsi, and switching to half-interval in case $f(x)$ is nearly horizontal near the root, is a very effective one. When I began the computations appearing in my paper, I used regula falsi to solve the equations. I encountered the problem of a nearly horizontal $f(x)$ in one of the examples and immediately switched to the slower but always reliable half-interval method. In retrospect, Dr. Greville's combination would have retained the faster convergence of regula falsi with no sacrifice of reliability.

I also wish to thank Dr. Greville for calling my attention to an error which appeared in the galley proof version of my paper.

With respect to Mr. McCracken's discussion, the existence of multiple rates of return is due not to a discontinuity in $f(x)$ but rather to the fact that multiple rates of return do, in fact, exist. This phenomenon has been documented in the literature.[1] I believe that a method for calculating a unique figure of merit for the performance of an investment over any period is needed, although Mr. McCracken's suggested procedure is too dependent on subjective considerations.

[1] See William H. Jean, "On Multiple Rates of Return," *Journal of Finance*, March, 1968, and Seymour Kaplan, "A Note on a Method for Precisely Determining the Uniqueness or Nonuniqueness of the Internal Rate of Return for a Proposed Investment," *Journal of Industrial Engineering*, January-February, 1965.