DISCUSSION OF PAPER PUBLISHED IN VOLUME LV

AN ACTUARIAL NOTE ON
ACTUARIAL NOTATION

JEFFREY T. LANGE
VOLUME LV, PAGE 196

DISCUSSION BY LEWIS H. ROBERTS

A paper on actuarial notation in casualty and property insurance is welcome because it forces us to take stock of the basic language of our profession — an important subject that we usually ignore because of the press of more immediate problems. Compact and consistent notation is valuable not only because of its role in communication but also because it assists us in developing our bits and pieces of actuarial knowledge into a coherent and systematic science. In the words of Ernst Mach: "Strange as it may sound, the power of mathematics rests on its evasion of all unnecessary thought and on its wonderful saving of mental operations."[1]

It goes without saying that actuarial notation should be a tool for avoiding confusion, not causing it, yet it has sometimes seemed to this reviewer that the notation used by casualty actuaries worked more in the latter direction. Attempts to read some of the most important theoretical material in our literature are often stymied by difficulty with unique and complex notation. If the authors could have used a familiar, standardized notation, their contributions would have been much more vividly understood and appreciated by their colleagues and by students. In actuarial science, as in other branches of mathematics, "the medium is the message."

Notwithstanding the suggestion by the Committee on Terms, Definitions and Symbols,[2] it may not be the youth of casualty actuarial science that has prevented a stable notation, but at least two other reasons. One is its breadth of scope. It embraces not only the life functions — as in workmen's compensation and accident and health insurance — but mathematical statistics and other fields of applied mathematics. It is natural for notation

---

[1] E. T. Bell, *Men of Mathematics*, Simon and Schuster, New York, 1961.
[2] *PCAS* Vol. I, p. 76 and Vol. II, pp. 163, 317, and 497.

in casualty actuarial calculations to adopt the conventions of these disciplines but these conventions leave much to individual preference or whim.

A second reason is that few analogues of the complicated standard functions characteristic of life insurance exist in casualty insurance. We do have a variety of elements that can be put together in different ways to form "pure premiums," "loss ratios," etc., but the algebraic need for the compactness and expressiveness of life notation is weak in most routine calculations.

The author's discussion of notational problems in relation to computer programming is timely and well considered. I take one very small exception to his remarks in noting that at least literary Algol admits use of small letters as well as capitals. The only Algol compiler used by our staff, however, permits only capitals.

Adding to what Lange has said, I would urge that any committee appointed to work on notation for programming include, or at least consult with, a professional programmer in order to avoid unnecessary pitfalls. Options that may seem trivial from a formal or mathematical standpoint in a programming language can make a great deal of difference in core memory requirements and running time. To take one of the author's examples, AT5A2 (X, N1, N2) is preferable to AT5A (2, X, N1, N2) because each additional index or argument materially increases computing time. The more information that can be contained in the name itself, the faster a program will run. That this is not a trivial difference can be seen from the following table based on IBM 1130 execution times:

Designation of Variable	Example	Additional Microseconds Relative to an Unsubscripted Variable
No subscripts	A	0
Constant subscript	A(3)	25
One variable subscript	A(K)	280
Two variable subscripts	A(K,M)	390
Three variable subscripts	A(I,J,K)	530

In a one-shot program this is meaningless, of course, but on long production runs such differences can add up to hours.

In conclusion, I suggest that one of the strongest arguments for standardizing notation lies in computer applications, where it could aid materially toward improving the accuracy and speed of programming.