ANDERSON, S. F., EARLE, J. G., GOLDSCHMIDT, R. E., and POWERS, D. M. (1967). The IBM System/360 Model 91: Floating Point Execution Unit, *IBM Journal of R. and D.*, pp. 34-53.
CONTROL DATA CORPORATION (1966). 6600 Central Processor Description, Control Data Institute.
FLORES, I. (1963). *The Logic of Computer Arithmetic*, Prentice-Hall, Englewood Cliffs, N.J.
HABIBI, A., and WINTZ, P. A. (1970). Fast Multipliers, *IEEE Trans. on Computers*, Vol. C-19, No. 2, pp. 153-157.
PEZARIS, S. D. (1971). A 40-ns 17-bit by 17-bit Array Multiplier, *IEEE Trans. on Comp.*, Vol. C-20, No. 4, pp. 442-447.
PUGH, A. (1967). Application of Binary Devices and Boolean Algebra to Realisation of 3-valued Logic Circuits, *Proc. IEE*, Vol.114, pp. 225-228.
RAMAMOORTHY, C. V., and ECONOMIDES, S. C. (1969). Fast Multiplication Cellular Arrays for LSI Implementation, 1969 FJCC, Vol 35, pp. 89-98.
VRANESIC, Z. G., LEE, E. S., and SMITH, K. C. (1970). A Many-Valued Algebra for Switching Systems, *IEEE Trans. on Comp.*, Vol. C-19, No. 10, pp. 964-71.
WALLACE, C. S. (1964). A Suggestion for a Fast Multiplier, *IEEE Trans. on Elec. Comp.*, pp. 14-17.

Correspondence

To the Editor
The Computer Journal

Sir,

The generalised Euler transformation

A recent paper by Wynn (1971) discusses the generalised Euler transformation

$$\sum_{s=0}^{\infty} u_s \Rightarrow \frac{1}{1-z}\sum_{s=0}^{\infty}\left(\frac{z}{1-z}\right)^s \Delta v^s{}_0 ,$$

where $u_s = z^s v_s$. An important matter in the use of this transformation is the choice of a suitable value for z, and Wynn suggests that z be chosen as the limit as $s \to \infty$ of the ratio u_{s+1}/u_s (provided that this limit exists). In fact this does not generally lead to the best value of z.

Consider, for instance, the well-known series

$$1 - \tfrac{1}{2} + \tfrac{1}{3} - \tfrac{1}{4} + \ldots,$$

in which $u_s = (-1)^s(s+1)^{-1}$. The ratio u_{s+1}/u_s tends to -1, and putting $z = -1$ gives the transformed series

$$\tfrac{1}{2}[1 + \tfrac{1}{2}(\tfrac{1}{2}) + \tfrac{1}{3}(\tfrac{1}{2})^2 + \tfrac{1}{4}(\tfrac{1}{2})^3 + \ldots] .$$

This converges more rapidly than the original series; but we can do better still by putting $z = -\tfrac{1}{2}$, which gives

$$\tfrac{2}{3}[1 + 0 + \tfrac{1}{3}(\tfrac{1}{3})^2 + 0 + \tfrac{1}{5}(\tfrac{1}{3})^4 + \ldots] .$$

A more startling example is given by taking

$$u_s = \tfrac{1}{5}[(\tfrac{2}{5})^s + (\tfrac{4}{5})^s] .$$

In this case $u_{s+1}/u_s \to \tfrac{4}{5}$, and the transformed series with $z = \tfrac{4}{5}$ is

$$2 - 2 + 2^2 - 2^3 + 2^4 - \ldots,$$

which is divergent. Yet by taking $z = \tfrac{3}{5}$ we get the rapidly convergent series

$$1 + 0 + (\tfrac{1}{2})^2 + 0 + (\tfrac{1}{2})^4 + \ldots .$$

A theoretical method for finding the optimum value of z was given by me in an earlier paper (Scraton, 1969). This method cannot be used, however, if one knows nothing about the terms u_s except their numerical values, and as far as I am aware no computational algorithm has yet been devised for finding the optimum value of z in these circumstances.

Yours faithfully,
R. E. SCRATON

Department of Mathematics
University of Bradford
Bradford 7
14 March 1972

References

SCRATON, R. E. (1969). A note on the summation of divergent power series, *Proc. Camb. Phil. Soc.*, Vol. 66, p. 109.
WYNN, P. (1971). A note on the generalised Euler transformation, *The Computer Journal*, Vol. 14, p. 437.

To the Editor
The Computer Journal

Sir,

In his letter on high level languages in Vol. 15, No. 1, 1972 of this Journal, J. Palme gives an example of a spelling mistake which he says would be detected in ALGOL but not in FORTRAN.

Provided the incorrectly written variable did not also occur on the left hand side of an assignment statement it would be detected by a good FORTRAN compiler as an undefined variable.

Yours faithfully,
H. W. BRADLY

3 Belleville Drive
Oadby
Leicester LE2 4HA
24 March 1972

To the Editor
The Computer Journal

Sir,

G. M. Bull's article 'Dynamic debugging in BASIC' (February, 1972) was chiefly valuable in spreading the gospel about the great advantages of interpretive compilers for use in time sharing. However, his implementation contains little that is new; all his facilities except breakpoints and the trace feature have existed for several years on the Conversational Programming System (CPS) running on 360/40's and up. CPS, with a choice of PL/1 or BASIC, remains (until TSO proves otherwise) the best general purpose time sharing system available on IBM computers.

Yours faithfully,
DAVID SILBER

Reference

IBM MANUAL GH20-0758. Conversational Programming System (CPS) Terminal User's Manual.

P.S. Speaking of time sharing, does it not seem odd to anyone else that there has been no demand to form a BCS Specialist Group on Time Sharing?

Erratum

There was an error in the paper 'File design fallacies' by S. J. Waters (this *Journal*, Vol. 15, No. 1, p. 1). The formula on the 9th line of page 2 should read:

Track Hit Ratio = 1 − (1 - Record Hit Ratio) Number of Records in Track so that the multiplier should be a power.