A generalization of the algorithm as defined by equations (7) *et seq.* is now needed for the case in which $\boldsymbol{\delta x}$ is not prescribed by equation (7).

An argument similar to that of Section 3 leads to the replacement of (9) by

$$\boldsymbol{D}^{(i)} = \frac{(\boldsymbol{f}^{(i+1)} - \boldsymbol{f}^{(i)} - \boldsymbol{J}^{(i)}\boldsymbol{\delta x}^{(i)})(\boldsymbol{z}^{(i)})^T}{(\boldsymbol{z}^{(i)})^T\boldsymbol{\delta x}^{(i)}} \qquad (44)$$

which reduces to (9) in the usual case. In practice the use of (44) for the calculation of $\boldsymbol{D}^{(i)}$ in every case is recommended.

The values of C_m^k and Δ_k were monitored for all the experiments of Section 12. From these it would seem that the simplest procedure likely to give consistent results is to test C_m^k only, and reject the step if $|C_m^k| < \rho_0$. ρ_0 might be 10^{-4}. Larger values of ρ_0 may delay convergence considerably.

A more general method, which has a much larger domain of convergence, may be formed by imposing a success criterion. The usual criterion employed is the minimization of f^2.

It is ensured that each step gives rise to an improvement (i.e. reduces f^2) by multiplying the step by a suitable scalar in those cases where the direct application of the algorithm does not give rise to an improvement. The imposition of such a criterion ensures convergence over a large domain but does not impair the final convergence rate.

Acknowledgements

The author wishes to express his thanks to Imperial Chemical Industries Limited for permission to publish this paper, to his colleagues Mr. I. Gray and Dr. H. H. Robertson for their constant advice and encouragement, and to the referee for his constructive criticisms.

References

Bittner, L. (1959). "Eine Verallgemeinerung des Sekantenverfahrens (regula falsi) zur näherungsweisen Berechnung der Nullstellen eines nichtlinearen Gleichungssystems," *Wissen. Zeit. der Technischen Hochschule Dresden*, Vol. 9, p. 325.

Gantmacher, F. R. (1959). *Applications of the Theory of Matrices*, New York: Interscience Publishers Inc.

Todd, J. (1962) (Ed.). *A Survey of Numerical Analysis*, New York: McGraw-Hill Book Co.

Tornheim, L. (1964). "Convergence of Multipoint Iterative Methods," *J. Assoc. Comp. Mach.*, Vol. 11, p. 210.

Varga, R. S. (1962). *Matrix Iterative Analysis*, London: Prentice-Hall International.

Wolfe, P. (1959). "The Secant Method for Simultaneous Non-linear Equations," *Comm. Assoc. Comp. Mach.*, Vol. 2, p. 12.

To the Editor,

The Computer Journal.

"An impossible program"

Dear Sir,

I do not know whose leg Mr. Strachey is pulling (this *Journal*, January 1965, p. 313); but if each letter in refutation of his proof adds to some private tally for his amusement, then I am happy to amuse him. May I offer three independent refutations?

(i) He defines a function *T*[*R*]. Any subsequent "proof" that *T* cannot exist is then idle; the function exists by definition.

(ii) If *T* does not exist, then *P* does not exist, since *T* is essentially involved in the statement of *P*. So *P* is not a program. So *P* is not an acceptable argument for *T*.

(iii) If one accepts Mr. Strachey's reasoning up to the point "In each case *T*[*P*] has exactly the wrong value", the appropriate deduction is not "this contradiction shows that the function *T* cannot exist" but "this contradiction shows that either the function *T* does not exist or that *P* is not a program". Since the non-existence of *T* itself implies that *P* is not a program, the most that can be concluded is that in any event *P* is not a program.

I am, of course, being careful not to claim that Mr. Strachey's initial assertion (that it is impossible to write a program which can examine any other program and tell, in every case, if it will terminate or get into a closed loop when it is run) is false. But what is manifest is that his proof of the far stronger assertion (that *T*[*R*] does not exist) is invalid: both in its final step (see (iii) above) and in its assumption that a set of statements in CPL—or any other language—necessarily constitutes a program. (If anybody doubts my counter assertion that *P* is not a program, let him try compiling *P* in—any—machine language!)

Yours faithfully,
H. G. ApSimon.

22 Stafford Court,
London, W.8.
18 February 1965.