‡$\frac{p}{4}$ is rounded to the nearest integer.

§$p = n - 1$, for S3 decremented by the number of immediately enclosing unaccessed levels (see above), i.e. the value of p for S3 may be smaller than that relevant to Pascal implementation. q = number of levels accessed ($< = p$).

Nothing in the above strategy is particular to S3 and it could be used for any block structured language including Pascal. Therefore we conclude that the above data for S3 are a better comparison of 2900 with the other machines.

There is almost no difference between S3 and Pascal in the overhead of access. Obviously B6700's hardware display gives the expected advantage in access to outer level data.

However, the difference in the cost of procedure call between the Pascal and S3 implementations is substantial. The number of instructions executed by 2900 S3 is closer to the number executed by B6700 than the number executed by 2900 Pascal. In some important cases 2900 appears to equal or even better B6700.

Yours faithfully,

D. K. Messham and A. Elliott

International Computers Limited
Westfields, Kidsgrove, Stoke-on-Trent ST7 1TL
24 June 1980

Erratum

We apologise to Dr Westreich that errors crept into his paper 'An efficient predictor—corrector algorithm', published in the May 1980 issue of the Journal, at the typographical stage. We are therefore reprinting the algorithm and complete references.

We take this opportunity also to publish Dr Westreich's up to date address. He is now at Department 3605, Israel Aircraft Industries, Ben-Gurion Airport, Israel.

The algorithm

Consider the system of first order differential equations

$$y' = F(t, y) \tag{1}$$

with initial value

$$y(t_0) = y_0$$

where y is an n vector to be determined and $F(t, y)$ is a continuous n vector function of t and y. To find the solution at t_f

1. Choose a step size h.
2. Find the solution y_1 to the equation at $t_1 = t_0 + h$ (say by a Runge-Kutta algorithm).
3. Set $i = 1$, $t_{i+1} = t_1 + h$, $y_0' = F(t_0, y_0)$
   $y_1' = F(t_1, y_1)$ and $I = 1$
4. Let $I = -I$
5. Use the predictor
   $$\bar{y}_{i+1} = y_{i-1} + 2h\,y_i'$$
   to obtain an approximation to the solution at $t_{i+1} = t_0 + (i + 1)h$.
6. Compute $\bar{y}_{i+1}' = F(t_{i+1}, \bar{y}_{i+1})$
7. If $I = -1$ go to step 12 otherwise go to step 8.
8. Use the corrector
   $$\bar{y}_{i+1} = y_i + \cdot 5h(y_i' + \bar{y}_{i+1}')$$
   to find an approximate solution at t_{i+1}.
9. Compute $y_{i+1}' = F(t_{i+1}, \bar{y}_{i+1})$
10. Use the corrector of step 8 again to obtain
    $$y_{i+1} = y_i + \cdot 5h\,(y_i' + y_{i+1}')$$
11. Go to step 14
12. Use the corrector
    $$y_{i+1} = y_{i-1} + h(y_{i-1}' + 4y_i' + \bar{y}_{i+1}')/3$$
13. Compute
    $$y_{i+1}' = F(t_{i+1}, y_{i+1})$$
14. If $t_{i+1} = t_f$ we stop, otherwise we let $i = i + 1$ and return to step 4.

References

Kershaw, D. (1974). Volterra Equation of the Second Kind, in *Numerical Solution of Integral Equations*, L. M. Delves and J. Walsh, eds, Clarendon Press, Oxford, pp. 140-161.

Westreich, D. and Cahlon, B. (1979). Solution of Volterra Integral Equations and Differential Equation with Continuous or Discontinuous Terms, Math. Tech. Report, Ben-Gurion University, to appear.

Book review

Computing Principles and Techniques by B. L. Vickery, 1979; 182 pages. (*Adam Hilger*, £11·95)

As the author explains, the aim of the book is to provide a basic introduction to computing principles. This would seem to be a worthwhile objective as the greatest difficulty in understanding a new subject is often to understand the jargon. What this book provides, at one level, is almost a dictionary of common computer terms. This is particularly useful, for while the author deliberately avoids the use of jargon whenever possible, the same is certainly not true of the computer world. In fact the inverse is true. Another pleasing feature of the book is the section dealing with computer arithmetic. Presumably the beginner will be vaguely aware that computers use binary arithmetic, but such terms as hexadecimal or octal may be far less clear. Examples are also given which show how numbers of one base can be converted to another. It could be argued that the average user of a large computer system may never need to know such details, but this is certainly not true of the micro or even the minicomputer user.

The next two chapters cover the first elements of programming. The topic is introduced at the level of operation codes and memory addressing and leads on to consider mnemonic programming. The subject of bit manipulation is also discussed and some very basic concepts in Boolean algebra are mentioned. This section highlights one of the fundamental weaknesses of a book of this type, i.e. the desire to introduce a whole range of topics like Boolean algebra, which can only be touched on in the most superficial way. The next section covers the basic concepts of communicating with a computer from a remote device like a terminal. The common problems involved in an operation of this type are discussed, e.g. the type of character codes which are in common use. To this end one of the appendices provides a useful comparative table for octal, decimal and ASCII.

The last three chapters which discuss the general concepts of computing cover higher level operations. Another important topic discussed in this section of the book is that of errors. Mastery of the various types of error which are bound to occur in any program is essential particularly for the beginner.

The final chapter covers certain aspects of medical computing. Although interesting, the examples chosen are somewhat specific and in practice the programming problems involved are rather complex. The reason for including the final chapter was, presumably, to justify the book's inclusion in a medical physics series. The author might have been better advised to extend the applications section considerably because the reader is likely to be interested in the medical applications of computing. Nevertheless, on balance the objective of the book is a good one and the author has produced a text which should go quite a long way towards meeting the needs of someone entering the field of medical computing.

R. I. Kitney (London)