Since the loop is executed M times, the saving becomes even more significant.

6. Conclusions

An algorithm for variable delay in an arithmetic expression has been presented, and its use in a scheme for recognising parallel processable code in computer programs has been described. The implications of the variable delay method and its apparent value of discovering hidden parallelism have been illustrated through analysis of an example program.

The algorithm in its present form will not handle expressions that contain array or function references. However, only minor modifications are necessary to incorporate this feature. This and the value of the method will be the object of further research in this area.

Acknowledgement

The author expresses his gratitude to Professor C. V. Ramamoorthy and his colleagues for their encouragement and assistance.

References

BAER, J. L. (1972). *A Survey on Multiprocessing*, TR-72-05-01, Computer Science Group, University of Washington, Seattle, Washington.

BERNSTEIN, A. J. (1966). Analysis of Programs for Parallel Processing, *IEEE Trans.*, EC-15, pp. 757-763.

BINGHAM, H. W., FISHER, D. A., and REIGEL, E. W. (1967). *Automatic Detection of Parallelism in Computer Programs*, TR-ECOM-02463-4, Burroughs Corp., TR-67-4, AD 622 274.

CARNAHAN, B., LUTHER, H. A., and WILKES, J. O. (1969). *Applied Numerical Methods*. New York: John Wiley & Sons, Inc.

CONTROL DATA 6400/6500/6600 Computer Systems Reference Manual (1969). Control Data Corporation, St. Paul, Minnesota.

LORIN, H. (1972). *Parallelism in Hardware and Software: Real and Apparent Concurrency*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc.

STEVENS, W. L. (1971). *An Automated Method for the Editing, Construction and Graphical Display of Activity Networks*. (Ph.D. Thesis), Texas A. & M. University, Department of Industrial Engineering, College Station, Texas.

RAMAMOORTHY, C. V., and GONZALEZ, M. J. (1969*a*). *Recognition and Representation of Parallel Processable Code in Computer Programs*, N71-11333, The University of Texas, Austin, Texas.

RAMAMOORTHY, C. V., and GONZALEZ, M. J. (1969*b*). Survey of Techniques for Recognizing Parallel Processable Streams in Computer Programs, *Proc. FJCC, AFIPS*, Vol. 35, pp. 1-15.

RUSSELL, E. C., and BAER, J. L. (1970). Preparation and Evaluation of Computer Programs for Parallel Processing Systems, in Hobbs, L. C., Thesis, D. J., Titus, H., Trimble, J. and Highberg, I. (Eds.), *Parallel Processor Systems, Technologies, and Applications*. New York: Spartan Books, pp. 375-416.

VOLANSKY, S. A. (1970). Graph Model Analysis and Implementation of Computational Sequences. (Ph.D. Thesis), University of California at Los Angeles, Department of Computer Science.

WARSHALL, S. (1962). A Theorem on Boolean Matrices, *JACM*, Vol. 9, pp. 11-12.

Book reviews

Computational Methods in Ordinary Differential Equations, by J. D. Lambert, 1972; 278 pages. (*John Wiley and Sons Ltd.*, £5·50)

The title is slightly misleading, as the book is almost entirely concerned with initial-value problems; there are just five pages at the end concerned with the boundary-value case. Dr. Lambert takes a middle road between the rocky highlands of abstract numerical analysis and the muddy ground of routine computation, where the algorithms grow. As he states in his preface, few theorems are stated and even fewer are proved, but there is a great deal of careful analysis, and references are given where necessary.

A large part of the book is concerned with a careful discussion of linear multi-step methods, both from the theoretical and the practical standpoint. This section contains an admirably thorough treatment of the various types of instability which may arise. There follows a chapter dealing with Runge-Kutta methods, leading on to discussions of hybrid formulae and extrapolation methods. The whole of this part of the book deals only with the solution of a single first-order differential equation.

Chapter 8 then extends the methods to deal with systems of simultaneous equations, and in particular to stiff systems. This chapter gives the impression of being rather hurried, especially by comparison with the detailed arguments of the earlier part. Here the author does not seem to me to give sufficient weight to the practical problems which arise. In practice a single equation does not appear very often. Systems of two or three equations are more usual, and twenty or more are all too common. The very practical problem of implementing an implicit method for such a system is dismissed rather briefly in this chapter. The discussion of stiff systems is also rather brief, but this is a field which is still being vigorously tilled, and Dr. Lambert mentions a number of recent advances.

The publication of this book follows closely on the appearance of C. W. Gear's *Numerical Initial Value Problems in Ordinary Differential Equations*.* The two books cover so closely the same ground that a review will involve a comparison, which is worth making explicit. Both bring the reader up to date with recent work; both have good bibliographies, and it is interesting that each of them has a substantial number of references which do not appear in the other. Gear gives a much more formal treatment, in the form of lemmas, theorems and proofs, where Lambert gives a discussion with references. At the other end of the spectrum, Gear gives some complete FORTRAN programs, and thus tries to cater for the practical user of algorithms as well as the pure numerical analyst. Lambert, however, keeps to the middle road, and seems to me much more successful in providing the practical man with the explanation and information which he needs. In particular, Lambert gives more than just a survey of the vast collection of methods, with their derivation. When he believes that Method A is better than Method B, he is quite ready to say so.

D. F. MAYERS (Oxford)

*Reviewed in this *Journal* (Volume 15, number 2, May 1972, page 155).

Computer Science: Projects and Study Problems, by Alexandra I. Forsythe, Elliott I. Organick, and Robert P. Plummer, 1973; 292 pages. (*John Wiley and Sons Ltd.*, £2·85)

This book is a companion to the main texts and language supplements of a series that has developed over the last few years. This latest offering contains the specification of thirteen programming projects with hints and background discussion. These are of varying complexity and between them offer a thorough grounding in the practice of the programming art. In addition there are a large number of smaller exercises with examples and discussion on more detailed topics.

The major flaw in this otherwise valuable collection is the continual cross-referencing to the main texts—this being most pronounced in the study problems which are linked by chapter and section. This makes it less useful to those who are not hooked on the series, which is a pity. Rather worse, it gives those who are hooked one more excuse for not looking at the world outside. Most parents will be familiar with the problems of weaning their children away from the Famous Five, and I greatly fear that Alexandra I. Forsythe (who seems the most plausible candidate) is set to become the Enid Blyton of the programming schools.

C. M. REEVES (Leeds)