economical to keep the precision of intermediate results as low as possible, i.e. 2, 3, 5, 10, 20 rather than 2, 4, 8, 16, 20 obtained by doubling each time. The required intermediate precisions are obtained from the binary representation of the precision by subtracting one, shifting right so many places and adding one again. For example: $n = 20$ so $(n - 1)$ in binary is 1 0 0 1 1

Intermediate precisions

$$1 + 1 = 2$$
$$1\ 0 + 1 = 3$$
$$1\ 0\ 0 + 1 = 5$$
$$1\ 0\ 0\ 1 + 1 = 10$$
$$1\ 0\ 0\ 1\ 1 + 1 = 20$$

General

MP arithmetic operations are slow compared with most other computer operations. It is consequently worth while carrying out some preliminary manipulations with the data in order to reduce as far as possible the number of MP operations necessary.

All the arithmetic facilities described have been coded for the IBM 7090/94 and form a MP arithmetic package which has been tested and used during the last three years. The package is available from the Quantum Chemistry Program Exchange, Indiana University (1964).

Acknowledgements

I am grateful to Professor Stanley Hagstrom and Dr. Franklin Prosser for the benefit of discussions on these routines. I wish also to thank Professor Harrison Shull for the opportunity of working with the Quantum Chemistry Group at Indiana University.

References

HOWELL, KEITH M. (1964). "Multiple Precision Arithmetic Package", Quantum Chemistry Program Exchange, Indiana University, QCPE 39.

KNUTH, DONALD E. (1962). "Evaluation of Polynomials by Computer", *Comm. A.C.M.*, Vol. 5, p. 595.

POPE, DAVID A., and STEIN, MARVIN L. (1960). "Multiple Precision Arithmetic", *Comm. A.C.M.*, Vol. 3, p. 652.

Book Review

Dynamic Programming and Modern Control Theory, by R. Bellman and R. Kalaba, 1966; 112 pages. (New York: *Academic Press*, 44s. cloth, 24s. paper)

Bellman and his colleagues have been developing the theory of dynamic programming with great ingenuity and industry during the past decade or so. From time to time they have reported progress in books in which are summarized the work of scores of papers. The first book *Dynamic programming* appeared in 1957, and was followed by *Adaptive control processes: a guided tour* in 1961, by *Applied dynamic programming* in 1962, and in 1966 by the book now under review. There has necessarily been a good deal of repetition and overlap in these books, but there has also been a most welcome increase in clarity of presentation as the series progressed. The present book gives a compact description of dynamic programming as applied to control theory, and its main novelty lies in its exceptional lucidity. Of course Bellman's style has always been felicitous in many ways—witty, free from jargon, erudite without pompousness. The apparent simplicity of dynamic programming is deceptive, however, and it offers considerable difficulties of exposition. Also, in the earlier books there was a tendency to abstractness, and to deal with examples taken from business and gambling problems which made little appeal to control engineers.

As the theory of dynamic programming has progressed, its significance has gradually changed. Originally it was thought to provide a general computational algorithm for the solution of dynamic optimization problems (in particular problems of non-linear control). This indeed was the reason for its name. Now, however, it is becoming realized that ordinary dynamic programming cannot cope with non-linear control problems involving more than, say, two-state variables, due to storage limitations in present-day computers. Bellman and Kalaba admit (page 61) that a system with, say, four-state variables will require the application of more adroit techniques than we possess at present.

Perhaps the main significance of dynamic programming is as a basic unifying concept underlying and illuminating many other theories, such as the calculus of variations, Pontryagin's maximum principle, and Huygen's principle. Bellman's brilliant achievement here has been to recognize that difficulties of rigour, involving questions of continuity and limits, are often immaterial, and can frequently be postponed simply by postulating a discrete system instead of a continuous one, and by systematic use of his principle of optimality.

The book emphasizes that dynamic programming gives a means of formulating nonlinear stochastic control problems, and even adaptive control problems; formulating, but not solving. In fact according to the undaunted final sentence in the book "it is clear . . . that there is unbounded range for imagination and talent in these areas, and that practically everything remains to be done".

A. T. FULLER