much utility for data compression. For example the element GEOR may be isolated as a step in the formation of GEORGE. After that the frequency of GEOR will fall to zero or close to zero because it rarely or never occurs except as a part of GEORGE. Removal of such elements from the dictionary with corresponding adjustments to the data structure should improve the performance of the program.

If binary reference numbers are used then there is an advantage in ensuring that the size of the dictionary corresponds exactly with a power of two, otherwise part of the coding capacity of the reference numbers will be wasted. Alternatively a third improvement may be used, namely the introduction of Huffmann (1952) codes instead of simple reference numbers. The principle here, of course, is that shorter codes are used for the more frequent elements with a consequent saving in space. Note that the frequencies to be used in calculating Huffmann codes are not the frequencies recorded when elements are first formed but the frequencies of elements recorded after the dictionary is complete.

The method is suitable for any data which contains character strings which are repeated at irregular intervals. If however the data contains variable length sequences of identical contiguous characters or identical groups of characters then such sequences are probably best handled by run-length coding. For example, with data containing many strings of blanks of variable length MK10 would wastefully build up a whole set of elements corresponding to the various strings encountered. If all such strings were each converted to a single blank, perhaps with a record of the number of repetitions, then MK10 may operate on this pretreated text quite satisfactorily. Where repeating groups of characters occur then run-length coding of the groups should be applied after MK10 has identified the groups. Using a combination of a process like MK10 with a form of run-length coding it should be possible to extract the redundancies from most types of data.

6. Conclusion

This program seems to provide an effective way of developing an 'alphabet' of frequent character strings which, when identified in a text and replaced by shorter reference numbers or Huffman codes, allow a worthwhile compression of natural language data. The frequency information gathered by the program is exploited to minimise the amount of pattern matching required during segmentation so that encoding is economical. Once a data structure has been built up it can be used to encode any text having a structure similar to the original sample. Reconstituting a text from a string of reference numbers presents no problem. There seems no reason in principle why the set of strings obtained should not also be used as key 'words' in a document retrieval system as considered by Clare *et al.* (1972) and by Schuegraf and Heaps (1972).

References

Clare, A. C., Cook, E. M. and Lynch, M. F. (1972). The identification of variable-length, equifrequent character strings in a natural language data base, *The Computer Journal*, Vol. 15, pp. 259-262.

Edwards, E. (1969). *Information Transmission*, London: Chapman & Hall.

Huffman, D. A. (1952). A method for the construction of minimum-redundancy codes, *Proc. IERE*, Vol. 40, pp. 1698-1101.

Garner, W. R. and Carson, D. H. (1960). A multivariate solution of the redundancy of printed English, *Psychol. Rep.*, Vol. 6, pp. 123-141

Lynch, M. F. (1973). Compression of bibliographic files using an adaptation of run-length coding, *Info. Stor. Retr*, Vol. 9, pp. 207-214.

Olivier, D. C. (1968). Stochastic grammars and language acquisition mechanisms, Unpublished Ph.D. thesis, Harvard University.

Schuegraf, E. J. and Heaps, H. S. (1973). Selection of equi-frequent word fragments for information retrieval, *Info. Stor. Retr.*, Vol. 9, pp. 697-711.

Schuegraf, E. J. and Heaps, H. S. (1974). A comparison of algorithms for data base compression by use of fragments as language elements, *Info. Stor. Retr.*, Vol. 10, pp. 309-319.

Wolff, J. G. (1975). An algorithm for the segmentation of an artificial language analogue, *Br. J. Psychol.*, Vol. 66, pp. 79-90.

Wolff, J. G. (1977). The discovery of segments in natural language, *British Journal of Psychology*, Vol. 68, pp. 97-106.

Book review

Matrix Computation for Engineers and Scientists, by A. Jennings, 1977; 330 pages. (*Wiley*, £10·50)

This is a valuable book by one who has made significant contributions to the field over the last dozen years. It combines the practical approach of the engineer, together with plenty of real problems from which matrix computations arise, with a proper regard for the mathematical and computational niceties.

After an initial chapter which reviews basic matrix algebra, the book falls naturally into two parts, the solution of systems of linear equations and the solution of linear eigenvalue problems. An important feature is that each starts with a chapter on typical problems. These are drawn mainly from engineering, rather than science in general, and serve to show how important is the formulation of a problem in matrix terms before its solution begins: alternative formulations may well lead to matrices of differing sizes and quite different properties. In view of the wide use of the finite element method nowadays, one could perhaps have expected a less specialised and ad hoc interpretation of the technique as well as some indication of the more general form of matrices arising from it and a further reference to it under vibration problems.

The chapters on solution of equations cover storage schemes and matrix multiplication, elimination methods, sparse matrix elimination and iterative methods. The treatment of sparsity is particularly thorough and all these chapters show a close concern for the details of implementing each method on a computer. There are plenty of simple numerical examples worked out in detail and central pieces of coding are given in both FORTRAN and ALGOL. The eigenvalue chapters continue in the same style but without the coding, covering transformation methods, Sturm sequence methods and vector iterative methods, including simultaneous iteration for several eigenvalues and vectors.

The detail and authority with which certain topics are treated will make the book of interest to experienced numerical analysts and computer scientists, though the book is clearly aimed primarily at scientists and engineers. These will note with pleasure that the author does not hesitate to compare methods, listing their advantages and disadvantages and making judgements on which are the best for particular classes of problems. No more mathematics is used than is absolutely necessary: for instance, no use is made of matrix, as distinct from vector, norms and ideas of linear independence and rank are only briefly mentioned. Perhaps surprisingly, there is no mention of linear programming.

K. W. Morton (Reading)