Correspondence

Dear Sir,

Parallel Algorithms for the Iterative Solution to Linear Systems
[R. H. Barlow and D. J. Evans, *The Computer Journal*, **25**, 56 (1982)]

This paper considers systems satisfying Property A^1 for which it is possible to divide the equations into two disjoint subsets of equations wherein the update equation for any equation in one of the subsets depends only on component values in the other subset. Thus all components in a subset can be updated in parallel.

Here we wish to point out that similar subset dependency can be identified in banded linear systems. Thus given a system of size n, $AX = b$, then if the system has bandwidth $2m + 1$ it is possible to divide the components into m disjoint subsets (component i going into subset k = Remainder $(i-1, m) + 1$. The update of any component depends only on the values of components in the other subsets. Thus all components in a subset can be updated in parallel and the result is exactly the same as updating the components sequentially. Furthermore the banded nature of the system implies that the dependencies on the components in the other subsets are highly regular. Thus the update formulae for component i depends only on values of components $i - + j$ with $j < = m$. This regularity enables us to group together m consecutive components so that each group has one component from each partitioning and components in group i relate only to components in groups $i - + 1$. Up to n/m equations can be updated in parallel using p/m processors and all equations can be updated in m sequential steps. If the parallel computer is constructed of a linear array of processors with each processor connected to only its nearest neighbour on either side then all data dependencies can be handled by shifting the new set of results to the neighbouring processor on the left and right. Such limited connection structures are typical of many parallel computers having large numbers of processors.

In fact the above subset decomposition may be only one of many possibilities. Thus the dependencies defined by the iteration matrix can be represented by a graph $G(V,E)$, where nodes correspond to components and dependence is expressed through nodes being adjacent to each other. The problem of identifying independent subsets is equivalent to the graph colouring problem: thus nodes with the same colour cannot be adjacent/dependent. Of the many possible colourings, or partitioning, the maximum parallelism will be obtainable from those colourings with the minimum number of partitions. However this best partitioning for parallelism can have two defects. Firstly it may have different numbers of components in each partitioning and thus require many more processors to fully exploit this parallelism (p = max number of components in any partition) than the less parallel but regular partitioning based on the banded nature of the system. Furthermore the data communication overheads may be greater than for the bandwidth partitioning. Thus suppose we initially allocate one node or component per processor. Then n processors are required and the dependencies require new results to be made available to all m processors to the left and right. If the colouring was completely regular then we could proceed as in the banded partitioning and group l consecutive equations together with the result that each group had only one component in each partitioning and elements of a group related to other groups at most m/l distance away. However such regularity arising from these partitionings cannot be expected and any l consecutive components will not normally have l different colours. Thus some groups will contain fewer than l components and the dependence between some groups will extend to beyond the nearest m/l left or right groups.

For parallel computers of the type described above the optimization of the amount of parallelism may thus increase the overheads of exploiting parallelism.

Iterative solutions to banded and planar systems are currently under investigation on both Single Instruction Multiple Data computers (ICL DAP and CRAY 1) and Multiple Instruction Multiple Data computers (the NEPTUNE system referred to in the above paper).

Yours faithfully

R. H. BARLOW and S. MACQUEEN
DAP Support Unit
Computer Centre
Queen Mary College
Mile End Road
London E1 4NS
UK

Reference

1. Young D.M., *Iterative Solutions of Large Linear Systems*, Academic Press (1971).

Dear Sir,

Invitation to Pascal

In your November 1982 issue there appeared a review by David J. Cairns of Katzan's book *Invitation to Pascal*.[1] Unfortunately, Cairns seems to be less aware of some aspects of Pascal than Katzan, and several of his criticisms are invalid.

In particular, Cairns assumes that the Pascal language is defined by the Jensen and Wirth report.[2] However, there now exists a British Standard for Pascal (BS 6192) and an identical draft international standard (ISO/DIS 7185). (This is reproduced in Ref. 3.) Much of the 'non-standard terminology' that Cairns objects to is actually taken from these standards. Four of the 'errors of fact' that Cairns mentions would appear to be misprints. Of the remaining three, in the context of BSI Standard 6192 Cairns is partly right about two (procedure headings[4] and *var* parameters of functions[5]) and is wrong about the third (arrays[6]).

Perhaps Katzan's book is as poor as Cairns would have us believe. I have not seen it myself and cannot say. What is certain is that critics are obligated to be as accurate and up to date as the authors they criticize.

Yours faithfully

BILL FINDLAY
Computing Science Department
University of Glasgow
Glasgow G12 8QQ
UK

References

1. H. Katzan Jr, *Invitation to Pascal*, Van Nostrand Reinhold (1981).
2. K. Jensen and N. Wirth, *Pascal User Manual and Report*, 2nd ed., Springer Verlag (1975).
3. I. R. Wilson and A. M. Addyman, *A Practical Introduction to Pascal—with BS 6192*, 2nd ed., Macmillan (1982).
4. *Ibid*, p. 191.
5. *Ibid*, p. 189.
6. *Ibid*, p. 177.

Dear Sir,

Invitation to Pascal

May I, having recovered from a similar experience, extend to Dr. David Cairns every sympathy in having had to review *Invitation to Pascal* by Harry Katzan Jr. It dulls the mind and numbs the senses, no doubt accounting for the aberration in his catalogue of errors of fact. Both Jensen and Wirth (p. 39) and ISO/DIS 7185 (sections 6.4.3.2 and 6.5.3.2) state that the form

array [index1, index2] **of** component

is merely an abbreviated equivalence of

array [index1] **of array** [index2] **of** component

and hence that slicing is allowed in either case.

Yours faithfully

I. R. MACCALLUM
Department of Computer Science
University of Essex
Wivenhoe Park
Colchester CO4 3SQ
UK

Dear Sir,

Invitation to Pascal

For people like myself who were brought up on a diet of Algol 60, Pl/l and Algol 68, the