J. Fluid Mech. (1993), vol. 254, pp. 722–723

REVIEW

Parallel Computational Fluid Dynamics: Implementation and Results. Edited by H. D. Simon. The MIT Press, 1992. 345 pp. £40.50.

Recent developments in computer architecture are fast changing the face of computational fluid dynamics. Whereas past advances were typically driven by improved modelling of fluid phenomena, better understanding of theoretical fluid dynamics and greater algorithmic ingenuity, the new name of the game is to exploit exceedingly fast and elaborate parallel computing devices.

At a first glance, parallel computation is an intellectual cop-out. Thus, if your present combination of physical and mathematical understanding, numerical analysis and computer resources are unequal to their task, just wait for Intel or The Thinking Machines Corporation to release their next generation of super-number-crunchers and (as long as your financial backing is secure) your problems will be over. This impression, needless to say, is false. Sophisticated use of parallel computers opens up a whole range of new problems and challenges and calls for a great deal of intellectual input – in a new direction.

The present volume is a worthwhile introduction to many of these problems. Its main *leitmotif* is the emphasis on computer architecture. To a generation of computer users who have been weaned on the 'von Neumann machine', computers differ only in speed, not in structure. However, combining many processors to operate in unison has obvious – and non-trivial – implications. Firstly, the cost of communication becomes as significant as the cost of computation. The expense of passing messages among the processors is, clearly, dependent on their connectedness and 'architecture'. Some of the current favourites are ring, lattice and – strongly emphasized in the present volume – hypercube configurations.

The need to perform a large number of similar tasks in parallel and to pass messages in an economical and restrictive manner require a new approach to the very concept of a numerical algorithm and often change the designation of a 'good' computational method. Classical, well-known numerical methods, all developed in the ante-parallel days, are typically suboptimal in view of the new rules of the game. Inasmuch as the main effort in the early days of parallel computing (the prehistoric early Eighties) has been expanded on harnessing old computational favourites to a new setting, the present emphasis is on novel approaches that overtly exploit parallelism. Possibly the most important technique, extensively mentioned in the volume, is domain decomposition, whereby a 'difficult' geometry is decomposed into 'simple' chunks, a PDE solved separately in each of these portions (needless to say, in parallel) and the outcome 'glued together' to encompass the original geometry. Conceptually, it is an example of the 'divide and conquer' approach, ubiquitous in modern parallel computation. Another algorithmic aspect of crucial importance is how to exploit parallel, vector or systolic structure to speed up basic linear algebra operations, from vector-matrix products to the solution of sparse linear systems. There is much useful material on this subject in this book, in particular on iterative solution of linear systems. An interested reader may also turn to the extensive review of parallel linear algebra by Demmel, Heath and van der Vorst in *Acta Numerica 1993*.

An up-and-coming topic, absent from the volume, is the question of visualization. The sheer volume of data produced in parallel computing often obscures, rather than helping to elucidate, the underlying behaviour of a scientific model. The problem becomes more acute when interested parties include a very broad range of specialities, from hardware technicians to mathematicians to engineers – and often to media, decision makers and members of the public – each with their own terminology and outlook. Hence the growing importance of modern techniques of representation and visualization of data.

Various papers in this book approach parallel CFD from three generic points of view – the phenomenological, the algorithmic and the computer-scientific. This is a most welcome feature, since – as should be obvious by now – successful parallel computation is an interdisciplinary effort. The terminology is, however, firmly within the traditions of fluid dynamics and computer jargon is, thankfully, kept at bay.

New generations of hardware and algorithmic advances revolutionize parallel CFD almost by the year. Hence, almost any exposition of the subject is past its sell-by-date even before it reaches the printers. One often encounters the sentiment of 'why learn parallel computing if everything is anyway about to change again soon?' This sentiment is misplaced. Developments are not random – they follow common principles and build upon former developments. Time spent learning the parallel craft is time well spent and the present volume is a good starting point.

A. Iserles